ROYAUME DU MAROC
Université Mohammed V - RABAT -
جامعة محمد الخامس – الرباط –
Faculté des sciences
كلية العلوم
CENTRE D'ETUDES DOCTORALES - SCIENCES ET TECHNOLOGIES

N° d'ordre 3313

THESE

En vue de l'obtention du : ***DOCTORAT***

Structure de Recherche : Laboratoire Mathématiques, Informatique et Applications – Sécurité de l'Information
Discipline : Informatique
Spécialité : Informatique, Agent Mobile, Sécurité informatique.

Présentée et soutenue le : 15/07/2020 par :

Sophia ALAMI KAMOURI

Modèles de service à base d'agents mobiles pour l'amélioration de la communication et de la sécurité dans un environnement intelligent

JURY

Souad EL BERNOUSSI	PES	Faculté des Sciences, Université Mohammed V de Rabat	Présidente
Ghizlane ORHANOU	PH	Faculté des Sciences, Université Mohammed V de Rabat	Directrice de thèse
Said ELHAJJI	PES	Faculté des Sciences, Université Mohammed V de Rabat	Co- directeur de thèse
Redouane BENAINI	PH	Faculté des Sciences, Université Mohammed V de Rabat	Rapporteur/Examinateur
Jalal LAASSIRI	PH	Faculté des Sciences, Université Ibn Tofail, Kénitra	Rapporteur/Examinateur
Hicham BENSAID	PH	Institut National des Postes et Télécommunications, Rabat	Examinateur
Kaouthar CHETIOUI	PA	ENSA Fès, Université Mohamed Ben Abdellah	Invitée

Année Universitaire : 2019-2020

Dédicaces

À ma très chère mère :
Fatima LAHBOUB

À la personne qui m'a tout donné sans compter, la meilleure de toutes les mamans
Aucun hommage ne saurait transmettre à sa juste valeur l'amour que je porte pour toi.
Source inépuisable de tendresse, de patience et de sacrifice
Ta prière et ta bénédiction m'ont été d'un grand soutien tout au long de ma vie.
Tu n'as pas cessé de me soutenir et de m'encourager, j'espère ne jamais te décevoir.
Je te dédie cette thèse qui concrétise ton rêve le plus cher
et qui n'est le fruit que de tes conseils et encouragements.
Puisse Dieu tout puissant te protéger du mal, te procurer longue vie, santé et bonheur
afin que je puisse te rendre un minimum de ce que tu as fait pour moi.

A mon très cher père :
Mouhcen ALAMI KAMOURI

Tu as toujours été pour moi un exemple de père respectueux, honnête
Aucune dédicace ne saurait exprimer le respect, l'amour et l'estime que j'ai pour toi.
Tu as su m'entourer d'attention et m'inculquer les valeurs nobles de la vie.
Que Dieu te préserve afin que tu demeures le flambeau illuminant mon chemin.

Remerciements

Ce travail a été réalisé au sein du laboratoire Mathématiques, Informatique et Applications - Sécurité de l'Information (LabMIA - SI), à la Faculté des Sciences de Rabat, à l'Université Mohammed V de Rabat sous la direction des Professeurs Ghizlane ORHANOU et Said ELHAJJI.

Le mérite d'une thèse appartient certes à l'auteur, mais également à son directeur qui l'encadre, je tiens à exprimer toute ma reconnaissance et mes chaleureux remerciements à ma directrice de thèse, Madame Ghizlane ORHANOU, Professeur Habilité à la Faculté des Sciences, Université Mohammed V de Rabat. Je la remercie pour sa disponibilité, pour sa patience et pour le partage de connaissance qui m'a été de grande utilité sans oublier son soutien moral.

Je remercie également mon co-directeur de thèse, Monsieur Said ELHAJJI, Professeur de l'Enseignement supérieur à la Faculté des sciences, Université Mohammed V de Rabat, pour le temps qu'il a consacré pour répondre à mes innombrables questions et pour ses conseils avisés. C'est certes avec joie et fierté que je dépose aujourd'hui ma thèse, mais aussi avec un brin de nostalgie que je termine ce programme de doctorante.

J'adresse mes sincères remerciements à la présidente du Jury, Madame Souad EL BERNOUSSI, Professeur de l'Enseignement supérieur à la Faculté des sciences, Université Mohammed V de Rabat. C'est pour moi un grand honneur de vous voir présider ma soutenance de thèse. Je vous exprime mon estime respect.

J'exprime ma gratitude à Monsieur Redouane BENAINI, Professeur Habilité à la Faculté des Sciences, Université Mohammed V de Rabat, de l'honneur qu'il m'a fait en acceptant d'être rapporteur de cette thèse et pour le temps consacré à la lecture de mon travail.

Je tiens à témoigner toute ma reconnaissance à Monsieur Jalal LAASSIRI, Professeur Habilité à la Faculté des Sciences, Université Ibn Tofail de Kénitra de s'être rendu disponible pour ma soutenance et d'avoir accepté la fonction de rapporteur pour juger mon travail. Veuillez accepter l'expression de mon profond respect et reconnaissance.

Je remercie Monsieur Hicham BENSAID, Professeur Habilité à l'Institut National des Postes et Télécommunications - Rabat,d'avoir accepté de se joindre à ce jury comme examinateur, de l'honneur qu'il m'a fait en acceptant d'être rapporteur de cette thèse et pour le temps consacré à la lecture de mon travail.

Je tiens aussi à remercier Madame Kaouthar CHETIOUI, Professeur Assistant à École Nationale des Sciences Appliquées de Fès, Université Mohamed ben Abdellah pour l'intérêt qu'elle a manifesté en participant en qualité de membre invité à ce jury.

À tous ces intervenants, je présente mes remerciements, mon respect et ma gratitude.

Je tiens à remercier aussi tous les professeurs du Laboratoire Mathématiques, Informatique et Applications - Sécurité de l'information (LabMIA - SI) et mes collègues chercheurs pour leur soutien et l'ambiance du travail au sein du laboratoire.

Enfin, je ne peux passer outre ma reconnaissance envers mes parents, ma famille et mes amis. Leur présence, leur écoute, leur confiance en moi et leur soutien constant m'assurent des bases solides me permettant de persévérer.

Un grand merci à vous tous.

Résumé

La technologie d'agent mobile est une technologie prometteuse pour les systèmes distribués grâce aux différents avantages et qualités qu'elle apporte. Dans notre travail, nous nous intéressons à intégrer le modèle d'agents mobiles dans les environnements intelligents particulièrement dans les soins de santé. Nous proposons deux modèles de service à base d'agents mobiles, pour ambulance intelligente, pour la transmission des données en temps quasi-réel, offrant ainsi plus de flexibilité et d'efficacité pour le service médical. Nous nous focalisons, ensuite, sur l'aspect sécurité des systèmes à agents mobiles, nous proposons deux mécanismes de sécurité : la trace cryptographique pour assurer l'intégrité de l'agent mobile et l'authentification de son origine, le mécanisme d'agent SOS pour assurer la protection de l'agent léger lors de sa migration contre les hôtes malveillants et aussi contre les attaques de déni de service (DOS). Par ailleurs, nous menons une étude sur l'association de la technologie d'agent mobile avec l'internet des objets (IoT) dans les soins de santé intelligents. Ensuite, nous présentons une amélioration d'une architecture existante de soins de santé intelligents typique en intégrant les agents mobiles afin d'assurer la confidentialité des données transmises à travers le réseau et la disponibilité du système.

Mots clés : Agent mobile, ambulance intelligente, trace cryptographique, agent SOS, sécurité.

Abstract

Mobile agent technology is a promising technology for distributed systems thanks to the various advantages and qualities it brings. In our work, we are interested in integrating the mobile agent model in intelligent environments, particularly in healthcare. We offer two models of service based on mobile agents, for an intelligent ambulance, for transmission data in near real time, thus offering more flexibility and efficiency for the medical service. We then focus on the security aspect of mobile agent systems, we propose two security mechanisms : the cryptographic trace to ensure the integrity of the mobile agent and the authentication of its origin, the SOS agent mechanism to ensure the protection of the light agent during its migration against malicious hosts and also against denial of service (DOS) attacks. Otherwise, we are conducting a study on the association of mobile agent technology with the Internet of Things (IoT) in smart healthcare. Next, we present an improvement of a typical architecture of intelligent healthcare, where we integrate mobile agents to ensure data confidentiality and system availability.

Key words : Mobile agent, smart ambulance, cryptographic trace, SOS agent, security.

Table des matières

Table des figures

Liste des tableaux

Liste des acronymes

A

ACL	Agent Communication Language
AID	Agent IDentifier
AMS	Agent Management System
API	Application Programming Interface

C

CEF	Computing with Encrypted Fonctions
CORBA	Common Object Request Broker Architecture
CSI	Computer Security Institute

D

DF	Directory Facilitator
DMZ	Demilitarized Zone
DOS	Disk Operating System
DVR	Digital Video Recorder

E

ETSI	European Telecommunications Standards Institute

F

FIPA	Foundation for Intelligence Physical Agents
FIPA-OS	FIPA- Open Source
FISMA	Federal Information Security Modernization Act

H

HCS	Hospital Central Server
HIPAA	health Insurance and Portability Accountability Act
HTTP	HyperText Transfer Protocol

I

IBM International Business Machines
IDL Interface Definition Language
IoT Internet of Things
IP Internet Protocol
ISO International Organization of Standardization

J
JADE Java Agent Development Framework
JRE Java Runtime Environment

K
KQML Knowledge Query and Manipulation Language

L
LW LightWeight
LA Local Agent
LAN Local Area Network

M
MASIF Mobile Agent System Interoperability Facilities specifications
MEITCA Mitsubishi Electric Information Technology Center America
MITM Man In The Middle

O
OMG Object Management Group
ORB Object Request Broker

P
PA Primary Agent
PAN Personal Area Network
PDA Personal Digital Assistant
P2P Peer to Peer

R
RFID Radio Frequency IDentification
RMA Remote Management Agent
RMI Remote Method Invocation
RPC Remote Procedure Call
RSH Remote SHell

S
SA Shadow Agent
SAEPP Smart Ambulance European Procurers Platforms
SIC Secure-Image Controller
SIM Secure IMage
SMA Systèmes Multi-Agents
SSH Secure SHell

T

TCP / IP	Transmission Control Protocol / Internet Protocol
TIC	Technologies de l'Information et de la Communication

W

WAN	Wide Area Network
WBAN	Wireless Body Area Network

Introduction Générale

Avec la propagation rapide de l'internet en plus des progrès réalisés dans le domaine des réseaux informatiques et l'émergence de l'informatique omniprésente, les systèmes distribués sont de plus en plus répandus puisqu'ils permettent à plusieurs systèmes informatiques de fonctionner ensemble en coordonnant et communiquant de manière efficace comme une seule unité dans un réseau. Les systèmes distribués sont plus puissants et plus rapides que les systèmes à ordinateur unique et leur utilisation apporte un certain nombre d'avantages pour les tendances technologiques actuelles comme les systèmes intelligents, l'internet des objets, etc, en raison de la quantité des informations à accéder. Le modèle de communication le plus utilisé dans les systèmes distribués est le modèle traditionnel Client / Serveur, mais ce dernier présente plusieurs inconvénients notamment la consommation de la bande passante qui provoque des retards de réponse, d'où la nécessité d'un nouveau modèle de communication capable d'apporter plus d'avantages et de flexibilité aux systèmes distribués. Pour surmonter cela, les systèmes multi- agents et le modèle d'agents mobiles en particulier sont les plus appropriés aux systèmes distribués.

Dans un monde hautement connectés, une quantité énorme de données est traitée par un nombre indéfini d'utilisateurs quotidiennement. Les systèmes multi-agents jouent un rôle important pour mettre en œuvre des systèmes distribués qui nécessitent l'autonomie de leurs entités, ils tolèrent l'interopérabilité des ressources et le partage de connaissances.

L'intégration des agents mobiles [1] dans un système distribué fournit divers avantages, spécialement lors de la récupération d'information. Le modèle d'agents mobiles répond aux exigences et besoins des nouvelles technologies et celles des applications distribuées, puisqu'il offre des caractéristiques uniques, une flexibilité et plus d'avantages que le modèle traditionnel Client / Serveur.

La technologie des agents mobiles représente un domaine de recherche en plein effervescence. Il s'agit d'un moyen idéal pour pouvoir transmettre les données tout en économisant la bande passante contrairement au modèle Client / Serveur, puisqu'il est capable de déplacer le code au lieu de déplacer de grandes quantités de données. L'agent mobile est une entité logicielle qui agit au nom d'un utilisateur. Il est capable de se déplacer à travers le réseau de manière autonome afin de se rapprocher des ressources distantes pour accéder aux données localement et ne déplacer que les informations nécessaires. Il a aussi la capacité de coopérer et d'interagir avec d'autres agents afin de partager des données. Cela réduit le trafic réseau et améliore la transmission des données.

Adopter la technologie d'agents mobiles augmente les performances des applications distri-

buées, puisqu'elle offre une meilleure utilisation des ressources, réduit la latence du réseau et la consommation de la bande passante. Migrer vers l'information au lieu de transférer et déplacer un grand volume d'information offre ainsi un gain au niveau de l'échange de données. L'agent mobile est autonome et tolère les pannes, la déconnexion qui survient lors de la communication entre différents sites n'a pas d'effet sur le fonctionnement du système à base d'agents mobiles puisqu'il termine sa tâche et revient une fois la connexion est rétablie.

La technologie d'agent mobile est utilisée dans plusieurs domaines tels que la recherche sur le web, le commerce électronique, l'environnement intelligent, la surveillance des soins de santé intelligents, etc. Dans notre thèse nous nous sommes focalisés sur les progrès du secteur de la médecine et la santé, spécialement le domaine des soins de santé intelligents. Le développement de nouvelles technologies informatiques a donné naissance à la télémédecine et aux soins de santé intelligents, afin d'offrir plus de liberté dans la vie quotidienne des patients (spécialement les personnes âgées et les personnes ayant une maladie chronique) tout en continuant à être surveiller à distance tranquillement.

Aujourd'hui, les soins de santé intelligents fournissent des systèmes qui s'adaptent au personnel médical et aussi aux patients. Le fait d'associer la technologie d'agent mobile et l'internet des objets aux systèmes de soins de santé permet de résoudre plusieurs problèmes qui dans le secteur de la médecine traditionnelle étaient compliqués, puisque ces nouveaux systèmes sont utilisés pour surveiller le patient à distance, diagnostiquer son état, l'ausculter et lui prescrire des médicaments sans être amené à se déplacer jusqu'à l'hôpital selon le cas.

Malgré les avantages apportés par l'intégration du modèle d'agents mobiles dans les systèmes distribués, cette technologie a également des inconvénients notables, particulièrement quand il s'agit de l'aspect de la sécurité. La mobilité de l'agent est un atout mais en même temps pose un problème de sécurité, car au cours de son déplacement, il peut être la cible de plusieurs attaques puisqu'il est exécuté sur d'autres plateformes différentes de celle qui l'a créé.

Nous nous intéressons dans cette thèse à la technologie d'agents mobiles et au système à base d'agents mobiles. Nous présentons cette technologie en montrant ses avantages et ses inconvénients. Ensuite, nous proposons un modèle à base d'agents mobiles pour la transmission des données dans le domaine des soins de santé intelligents qui est l'ambulance intelligente. Nous implémentons aussi cette solution afin de montrer sa faisabilité. Puis, nous nous penchons sur l'aspect de la sécurité qui est primordial dans un système à base d'agents mobiles en proposant deux mécanismes de sécurité afin de garantir la confidentialité des données, la protection de l'intégrité du système, la non-répudiation et l'authentification de l'origine, en plus de contrer les attaques DOS et faire face aux hôtes malveillants. Nous décrivons aussi les défis de sécurité dans le domaine de soins de santé intelligents, ces défis consistent en la disponibilité du système et la confidentialité des données lors du traitement. Pour cela, nous proposons une amélioration de la couche réseau d'une architecture de soins de santé intelligents typique. Cette architecture a pour objectif d'améliorer la communication entre un patient utilisant l'application des soins de santé et l'hôpital et aussi d'assurer la disponibilité du système et la confidentialité des données en intégrant la technologie d'agents mobiles.

Ce rapport est organisé comme suit :

Le premier chapitre introduit la technologie d'agents mobiles dans les systèmes distribués. Nous présentons les différents modèles de communications les plus utilisés dans les applications distribuées avec leurs avantages et leurs inconvénients. Puis nous comparons entre ces différents

modèles et le modèle agent mobile afin de le mettre en exergue en montrant les avantages et inconvénients de chaque modèle. Nous décrivons le concept d'agents mobiles, ces composants, caractéristiques et ces qualités, en plus des différents types de mobilité d'un agent. Nous définissons les normes de standardisation, puis la manière dont fonctionne la technologie d'agents mobiles à savoir sa mise en œuvre, la manière de communiquer, de coordonner et coopérer entre agents mobiles, le langage de communication et les services d'exécution (S. Alami-Kamouri et al [1]).

Le deuxième chapitre illustre certains domaines d'applications de la technologie d'agents mobiles les plus connus, tels que la recherche de l'information sur le web, le commerce électronique et les environnements intelligents comme le domaine des soins de santé sur lequel on s'est basé dans notre étude. Nous présentons notre proposition qui est le déploiement d'un modèle à base d'agents mobiles pour la transmission des données pour une ambulance intelligente, nous décrivons les types d'agents mobiles utilisés dans cette approche. Nous décrivons aussi les différents types de plateformes existantes pour la mise en place d'un système à base d'agents mobiles avec le choix de la plateforme JADE qui est utilisé pour la mise en œuvre du modèle proposé afin de montrer sa validité et sa faisabilité (S. Alami-Kamouri et al [2, 3]).

Dans le troisième chapitre, nous passons à l'aspect de la sécurité qui est primordial dans chaque système à base d'agents mobiles. Nous commençons par discuter les objectifs de la sécurité des systèmes informatiques. Puis nous décrivons les problématiques de sécurité des agents mobiles, à savoir les menaces de sécurité d'un système à base d'agents mobiles, les exigences de la sécurité des agents mobiles, en plus des différents types d'attaques qui visent la disponibilité, la confidentialité et l'intégrité les systèmes à base d'agents mobiles. Nous discutons des approches existantes dans le domaine de la sécurité des systèmes à base d'agents mobiles, puis nous présentons notre modèle de sécurité. Notre approche est bi-dimensionnelle, puisque nous proposons deux mécanismes de sécurité, à savoir la trace cryptographique afin de garantir l'intégrité de l'agent mobile et l'authentification de l'origine lors de son déplacement d'un environnement à un autre. En plus de l'agent SOS afin d'assurer la protection de l'agent contre les hôtes malveillants et les attaques par déni de service et garantir en même temps la disponibilité du système. Ensuite, nous implémentons le mécanisme agent SOS en simulant des attaques DOS et voir si l'agent SOS peut détecter les hôtes malveillants (S. Alami-Kamouri et al [4]).

Dans le quatrième chapitre, nous faisons une analyse dans le domaine de soins de santé intelligents et nous discutons les défis de sécurité qui touchent ce domaine. Nous présentons le concept de soins de santé intelligents. Puis nous montrons l'utilisation de l'internet des objets et les systèmes multi-agents dans les soins de santé et le plus qu'ils apportent à ce dernier. Nous décrivons les défis de sécurité des soins de santé intelligents, en précisant les exigences et les menaces de la sécurité, en plus des services de sécurité dans un environnement intelligent. Nous discutons aussi deux lois adoptées pour la protection de la vie privée des patients et qui doivent être appliquées dans les plateformes dédiées au domaine médical. Puis, nous présentons une architecture de soins de santé intelligents typique et nous examinons les problématiques de cette dernière. Ensuite, nous proposons une amélioration de cette architecture en y intégrant les agents mobiles pour la transmission des données afin de garantir la disponibilité du système et la confidentialité des données transmises.

A travers cette thèse, nous nous sommes intéressés à la technologie d'agents mobiles pour la transmission des données vue les avantages qu'elle apporte au système l'utilisant, nous l'avons associé au domaine des soins de santé et à l'IoT où beaucoup de données sont partagées de

manière régulière. Nous nous sommes focalisés aussi sur l'aspect sécurité en proposant des solutions pour système à base d'agents mobiles. L'intégration du modèle agent mobile avec l'IoT nous a permis d'améliorer l'architecture de soins de santé intelligents et d'ouvrir une nouvelle vision sur la protection de la vie privée des utilisateurs pour les travaux de recherche à venir.

Chapitre 1

Les agents mobiles dans les systèmes distribués : état de l'art

De nos jours, Internet a donné accès à une quantité d'information répartie sur plusieurs systèmes du réseau. L'évolution des réseaux informatiques à grande échelle a donné naissance au domaine de l'informatique distribuée où l'ensemble des ressources disponibles ne se trouvent pas sur la même machine mais sur un ensemble d'ordinateurs indépendants connectés et qui communiquent entre eux via un réseau. La recherche d'information, l'échange des données et la répartition des tâches dans ces systèmes distribués nécessitent la coopération et l'interaction entre plusieurs et différentes entités à travers le réseau. Pour cela, nous nous intéressons dans ce chapitre à la technologie des agents mobiles qui est une solution prometteuse facilitant la mise en œuvre et la communication des applications dans les systèmes distribués [10].

1.1 Introduction

Au début, les systèmes informatiques étaient des entités isolées ne communiquant qu'avec des opérateurs humains. Aujourd'hui, les systèmes informatiques sont devenus interconnectés et sont mis en réseau dans de grands systèmes distribués. La tendance de l'informatique actuelle est d'avoir des systèmes omniprésents, interconnectés et intelligents. Le développement de paradigmes logiciels capables d'exploiter le potentiel de tels systèmes informatiques constitue un défi. Pour cela, les systèmes multi-agents semblent une bonne technologie pour le développement de systèmes distribués autonomes et intelligents. L'idée d'un système multi-agents est simple ; ce type de système est composé de plusieurs agents qui communiquent entre eux et interagissent les uns avec les autres pour échanger des informations et des données. Dans le cas général, ces agents agissent au nom de l'utilisateur [1, 10, 11].

Actuellement, les applications deviennent de plus en plus diverses et complexes, la technologie des agents mobiles a fait preuve de flexibilité dans le développement et la gestion des applications distribuées. Un agent est une entité logicielle capable de se déplacer d'un environnement à un autre afin de se rapprocher des ressources distantes pour réussir la tâche qui lui a été confiée.

Il existe plusieurs types d'agents. Dans notre recherche, nous nous penchons sur le type d'agent mobile dont la mobilité joue un grand rôle dans un système distribué, puisqu'elle lui

permet de migrer et de se déplacer d'un environnement à un autre et d'interagir avec d'autres agents .

La technologie des agents mobiles est un concept qui a connu une croissance éminente tant en matière de techniques que d'applications, tels que : l'informatique mobile, le commerce électronique, les formations en ligne, le travail coopératif, la gestion du réseau et les télécommunications. La puissance des agents mobiles pour la résolution des problèmes complexes réside dans leur autonomie, leur mobilité, leur tolérance aux pannes, leur charge réseau réduite, ce qui les aident à atteindre leurs objectifs de manière flexible en utilisant l'interaction et la communication avec d'autres agents dans le réseau [10, 11, 12].

Dans ce chapitre, nous allons étudier le concept des agents mobiles, ses caractéristiques et ses spécificités. Nous allons commencer par discuter des modèles de communications les plus utilisés dans les systèmes distribués. Ensuite, nous allons présenter le concept de l'agent mobile et la manière dont la communication se déroule entre agents mobiles, l'architecture de ce dernier et ses caractéristiques générales en mettant l'accent sur le plus qu'il ajoute et qui le différencie vis-à-vis des autres modèles. Nous allons aussi décrire les efforts de standardisation son fonctionnement.

1.2 Évolution des modèles de communication

Il existe plusieurs modèles de communication pour la création des systèmes et applications distribués. L'utilisation de ces modèles diffère selon l'application à laquelle ils seront dédiés. Dans cette section, nous allons décrire et présenter les modèles les plus utilisés pour mettre en œuvre une application distribuée, montrer les diverses insuffisances de chaque modèle, pour ensuite mettre en exergue la technologie des agents mobiles.

1.2.1 Modèle Client / Serveur

La plupart des applications qui impliquent des communications sur un réseau utilise le paradigme traditionnel Client / Serveur. Ce modèle est basé sur la communication entre deux entités, l'une nommée client et l'autre nommée serveur. Il consiste à établir une connexion entre des machines clientes qui font partie du réseau comme le montre la figure 1.1 et un serveur qui leur fournit des services. Ils sont connectés via des canaux de communications où les requêtes sont envoyées sur l'ensemble du réseau. Le client et le serveur sont généralement localisés sur deux machines distinctes mais parfois ils peuvent être sur la même machine [9, 13, 14].

Le modèle Client / Serveur se compose de quatre entités principales [9] :

- **le client** : processus qui demande l'exécution d'une opération à un autre processus en envoyant un message, contenant l'opération à exécuter sous forme de requête et attend un message en retour.
- **le serveur** : processus qui accomplit une opération sur la demande du client et lui retourne la réponse de sa requête. Il tourne en permanence et peut répondre à plusieurs clients en même temps.

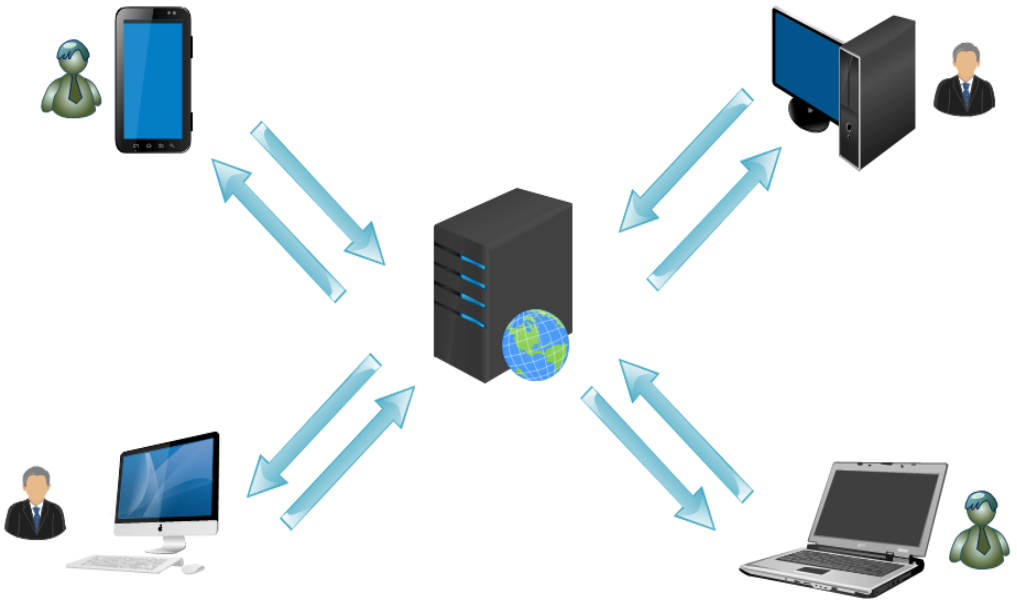

Fig. 1.1 – Architecture Client / Serveur

- **la requête** : c'est le message envoyé et transmis par le client au serveur, contenant l'opération à exécuter au nom du client.
- **la réponse** : c'est le message transmis par le serveur au client après avoir exécuté l'opération de ce dernier, ce message contient la réponse à la requête.

Fonctionnement du système Client / Serveur :

Comme le montre la figure 1.2, la communication du modèle Client / Serveur se déroule comme suit :

- Le client émet une requête vers le serveur en indiquant l'adresse IP de ce dernier et le port qui désigne un service particulier.
- Le serveur reçoit la demande et répond grâce à l'adresse IP et le port client de la machine cliente.
- Le client réceptionne les résultats délivrés par le serveur.

Avantages du modèle Client / Serveur :

On ne peut pas dire que le modèle Client / Serveur est un modèle de communication parfait, surtout pour les systèmes distribués mais cela dépend dans quel cas il est utilisé. Il est recommandé pour les réseaux qui nécessitent un grand niveau de fiabilité. Ses principaux avantages sont :

- **ressources centralisées** : le serveur gère les ressources communes à tous les utilisateurs. Étant donné qu'il est au centre du réseau, toutes les données sont centralisées sur un seul serveur, ce qui donne un contrôle de sécurité simplifié.

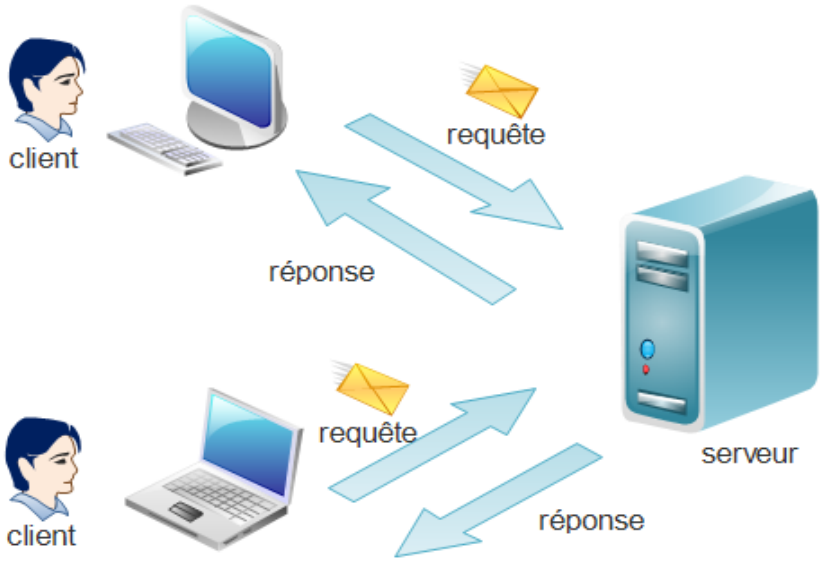

Fig. 1.2 – Communication Client / Serveur

- **meilleure sécurité** : un faible nombre de points d'entrée pour accéder aux données.
- **réseau évolutif** : l'architecture Client / Serveur donne la possibilité d'ajouter ou de supprimer des clients et même des serveurs sans brouiller et perturber le fonctionnement du réseau.

Malgré ses avantages, le modèle Client / Serveur a des inconvénients et le maillon faible de ce modèle tourne autour du serveur, car tout le modèle est architecturé autour de ce dernier.

Inconvénients du modèle Client / Serveur :

Lorsqu'un client a besoin d'un service particulier, il envoie généralement un message de requête au serveur contenant le service requis. Si le client a besoin d'un service qu'un serveur particulier ne possède pas, il doit trouver un serveur capable de répondre à la demande en envoyant plusieurs messages à d'autres serveurs.

Si plusieurs clients veulent communiquer avec le serveur au même moment, ce serveur risque de ne pas supporter la charge.

Cette approche traditionnelle Client / Serveur est coûteuse et peu fiable lorsque de nombreux messages doivent être envoyés entre le client et le serveur, c'est à dire, lorsque l'application commence à consommer beaucoup de bande passante du réseau. Dans ce cas, la communication augmente le trafic réseau et provoque des retards de réponse [9, 13, 14].

1.2.2 Modèle d'évaluation à distance

Le modèle d'évaluation à distance nommé aussi l'approche de l'envoi du savoir-faire ou du code permet le transfert du contrôle et des données entre les différentes fonctions d'une

application. Ce paradigme appartient au domaine de la mobilité du code. Dans ce modèle, le client dispose du code ou du savoir-faire propre de la tâche à réaliser, les ressources et l'unité d'exécution se trouvent sur le serveur. Le modèle d'évaluation à distance comme le montre la figure 1.3 s'applique dans le cas où le client envoie son savoir-faire ou son code à exécuter à un nœud distant du réseau tel un serveur. Dans ce cas, le serveur utilise ses propres ressources pour exécuter le code reçu. Une fois l'exécution du savoir-faire est terminée, il envoie les résultats au client. On prend comme exemple utilisant l'évaluation à distance la commande du système Unix RSH (Remote SHell) qui autorise l'exécution d'un script sur une machine distante [9, 10, 13].

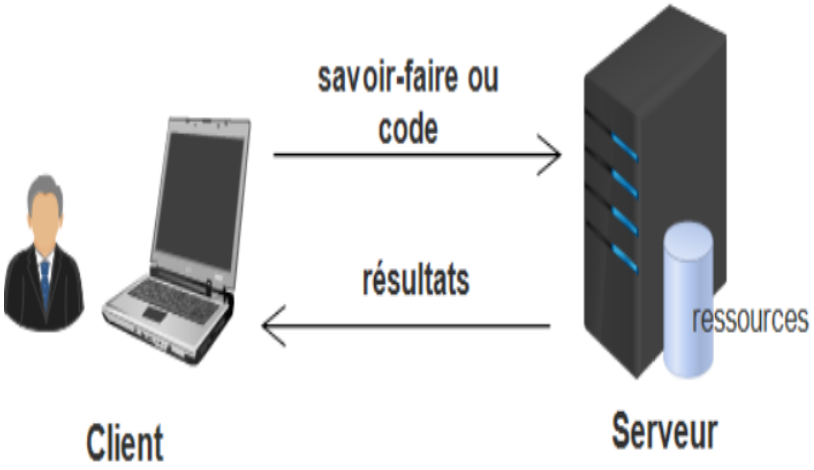

Fig. 1.3 – Architecture du modèle d'évaluation à distance

La différence entre le modèle traditionnel Client / Serveur et le modèle d'évaluation à distance est la mobilité du code, puisque ce dernier permet la mobilité du savoir-faire pour se rapprocher des ressources distantes et s'exécute sans avoir une connexion permanente entre le client et le serveur.

Avantages du modèle d'évaluation à distance :

Les atouts de ce modèle est sa capacité à collecter des informations détaillées sur l'utilisateur dans des contextes d'utilisation réels. Ceci est très utile dans des contextes où il est difficile d'installer un évaluateur pour enregistrer ou observer directement la session. En plus, cette centralisation a pour but d'accorder aux clients la possibilité de réaliser l'évaluation dans leurs environnements familiers, ce qui aide à assurer plus de sécurité grâce au comportement naturel des utilisateurs.

Inconvénients du modèle d'évaluation à distance :

Malgré les atouts de ce modèle, l'évaluation à distance présente certaines limites du côté de la détection des conditions environnementales dans lesquelles se déploient et évoluent la session. Sans oublier que la collecte des informations et le suivi des interactions et des comportements de l'utilisateur deviennent très difficiles dans le cas des applications dotées de fonctionnalités limitées (comme les appareils mobiles), puisqu'ils infligent des contraintes sur les types de techniques à utiliser.

1.2.3 Modèle code à la demande

Dans le modèle code à la demande nommé aussi l'approche de récupération du savoir-faire ou du code, le client dispose d'un ensemble de ressources importantes pour accomplir sa tâche, mais ne dispose pas du code ou du savoir-faire. Le client est dans ce cas dans l'obligation de coopérer avec le serveur distant pour récupérer le code nécessaire afin d'exécuter sa mission [10, 11]. Le serveur transmet le code au client et l'exécution va être effectuée sur la machine cliente tel qu'il est illustré sur la figure 1.4.

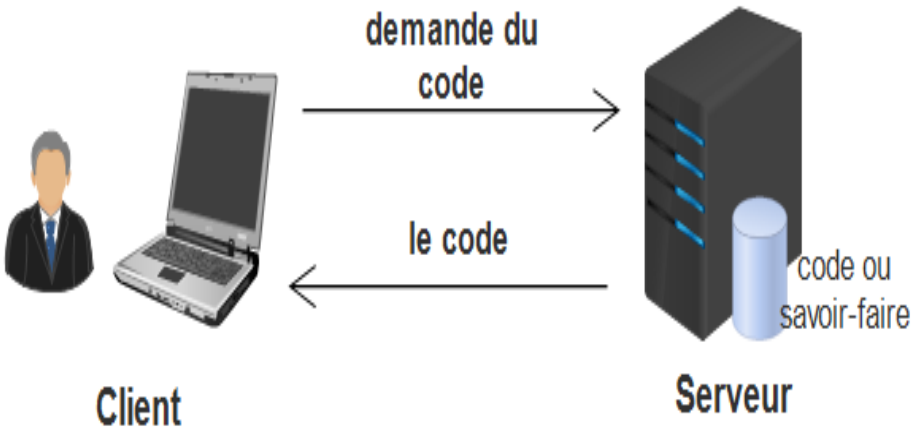

Fig. 1.4 – Architecture du modèle code à la demande

La différence entre le modèle précédent évaluation à distance et le modèle code à la demande se trouve dans les rôles du client et du serveur, puisque leurs rôles seront permutés.

Avantages du modèle code à la demande :

Ce modèle est très utilisé dans les services web pour ses avantages, puisqu'il permet d'élargir les fonctionnalités du site client en exécutant le code sous forme de script ou d'applet et réduit le nombre de fonctionnalités à pré-implémenter. Il peut également s'appliquer aux services et aux consommateurs de services. Par exemple, la conception de services peut permettre aux serveurs de reporter dynamiquement certaines parties de la logique aux programmes clients de service. Cette approche qui consiste à ralentir l'exécution du code du côté client est justifiable lorsque la logique du service peut être exécutée par le consommateur de manière plus efficace [15].

Inconvénients du modèle code à la demande :

L'inconvénient majeur de l'utilisation du modèle code à la demande est de réduire la visibilité de l'API (Application Programming Interface) sous-jacente, ce type de flexibilité n'est pas supporté par toutes les API [15].

1.2.4 Modèle de migration du processus

Ce modèle de migration du processus est utilisé pour équilibrer la charge dans un réseau local, comme le montre la figure 1.5. Un processus qui n'est pas terminé peut finir son exécution sur un autre site en déplaçant le savoir-faire (le code), les données en plus de l'état d'exécution. On parle ici d'une migration forte à un site distant (serveur) afin d'avoir un équilibrage de charge et ainsi chaque site supporte la même charge. Un processus est personnalisé par un savoir-faire ou du code qui décrit son comportement et par les données et l'état d'exécution de l'application qui l'intègre. Lorsqu'un processus énonce ces besoins en ressources, le système va positionner le processus demandeur sur la machine qui répond le mieux à ses besoins. Pour ce modèle, on parle de migration réactive, puisque la migration du processus est initiée par le système [10].

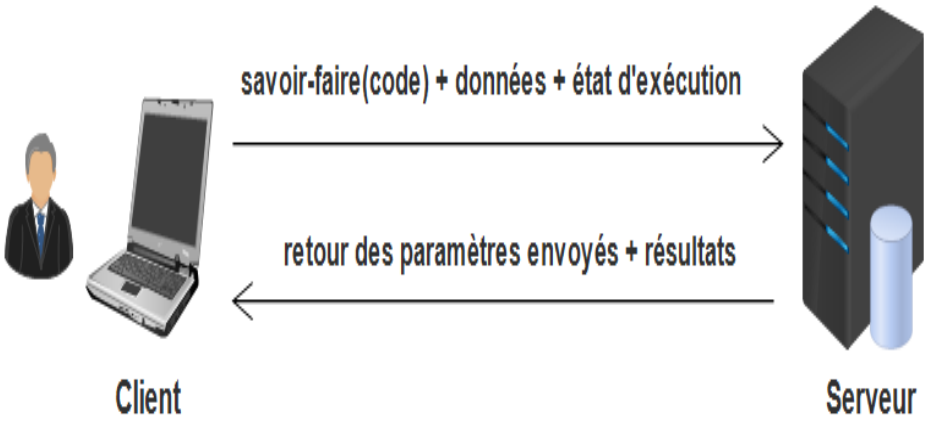

Fig. 1.5 – Architecture du modèle de migration de processus

Avantages du modèle de la migration du processus :

La migration de processus peut être appliquée afin d'améliorer les performances d'un système distribué dans un certain nombre de domaines. Elle permet d'équilibrer la charge dynamique dans un système distribué, puisque la charge de traitement des différents hôtes varie souvent de manière significative. Les processus peuvent être déplacés depuis des hôtes qui ont une charge relativement plus élevée vers des hôtes qui ont une charge relativement plus faible afin de répartir la charge de travail plus équitablement à travers le système. Elle permet aussi de réduire le trafic réseau, puisqu'un petit processus peut être migré vers le site d'une ressource qui ne peut pas être déplacée ou une ressource dont le mouvement entraînerait plus de trafic que la migration du processus (exemple : une grande base de données) [16].

Inconvénients du modèle de la migration du processus :

Malgré que la migration du processus est utilisée pour minimiser la charge dans un réseau, des inconvénients persistent. Ce modèle est typiquement implanté au niveau du système d'exploitation, le défi majeur de la migration du processus est de transférer l'état interne du

processus qui se trouve dans le noyau du système d'exploitation, en plus de la difficulté à transférer les ressources dans un système hétérogène [16].

1.3 Concept de l'agent mobile

Après avoir présenté les différents modèles de communications existants utilisés dans les applications distribuées en discutant leurs avantages et de leurs inconvénients, dans cette section, nous allons définir le modèle d'agent mobile et montrer ce qui le différencie des autres modèles étudiés dans la section précédente. Nous allons étudier ses composants, caractéristiques et qualités, en plus des types de mobilité de l'agent et son architecture, puis sa normalisation technique.

1.3.1 Définition d'un agent mobile

Dans le domaine de la recherche scientifique, il existe plusieurs définitions de l'agent mobile. D'une part des chercheurs considèrent un agent comme étant un logiciel et d'une autre part, des chercheurs considèrent un agent comme une entité capable d'agir à la place d'un être humain [17, 18, 19, 20, 21, 22, 23, 24].

Tout d'abord, la définition que nous allons aborder est celle d'un agent stationnaire donné par Ferber [25] puis nous allons montrer comment ce concept d'agent a évolué vers la mobilité.

D'après Ferber [25], un agent stationnaire est une entité physique ou virtuelle qui possède les propriétés suivantes :

- capable d'agir dans un environnement,
- capable de communiquer directement avec d'autres agents,
- capable de se mouvoir par un ensemble de tendances,
- possède des ressources propres,
- capable de percevoir (mais de manière limitée) son environnement,
- possède des compétences et offre des services,
- peut éventuellement se reproduire,
- son comportement tend à satisfaire ses objectifs, en tenant compte des ressources et des compétences dont il dispose et en fonction de sa perception, de ses représentations et des communications qu'il reçoit.

Pour rendre ce concept plus adéquat aux besoins des réseaux et à l'informatique nomade, l'aspect de la mobilité s'est alors associé à l'approche agent pour définir un nouveau concept nommé "agent mobile". Ce type d'agent est destiné à la mise en œuvre d'applications dont les performances varient en fonction de la disponibilité et de la qualité des services et des ressources, ainsi que du volume des données échangées [1].

D'ici, on déduit qu'un agent mobile est un agent qui accomplit les propriétés et les tâches d'un agent stationnaire [10] et grâce à sa mobilité, il devient capable de se déplacer d'une machine à une autre et d'un environnement à un autre pour se rapprocher des ressources distantes et réussir sa mission comme le montre la figure 1.6.

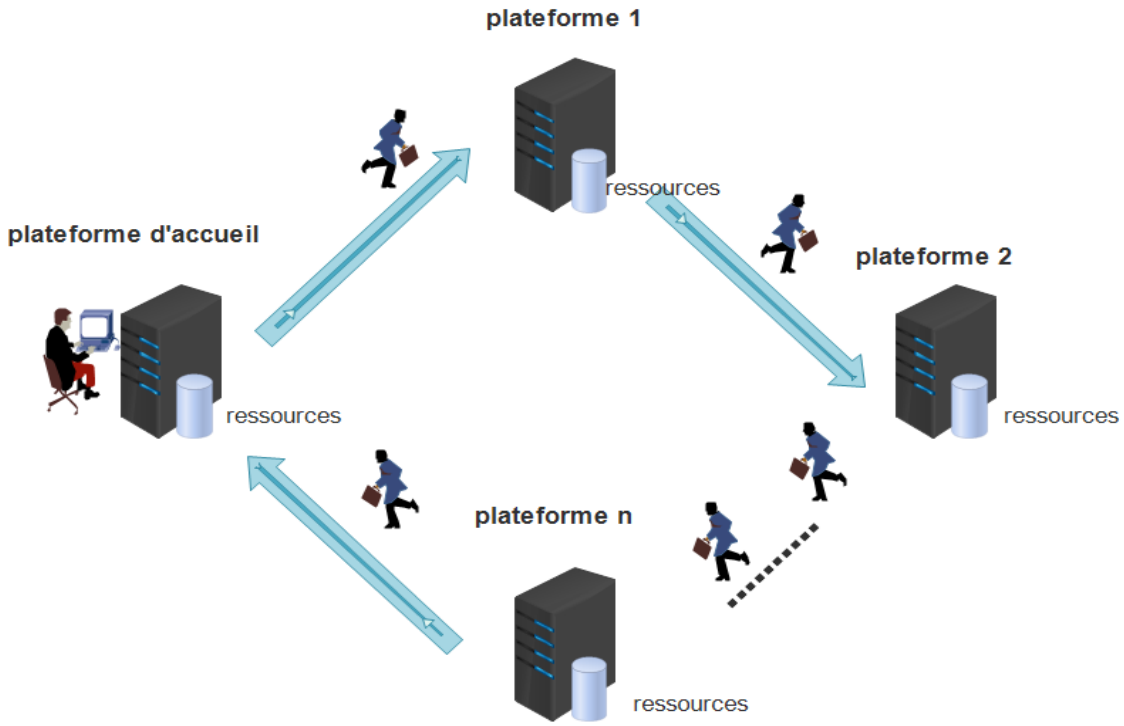

Fig. 1.6 – Architecture de l'agent mobile

L'utilisation des agents mobiles dans différents domaines donne lieu à plusieurs définitions, mais dans le contexte du présent travail on trouve :

Définition 1 : Un programme qui peut migrer d'un ordinateur vers un autre ordinateur au sein d'un réseau hétérogène. Le programme choisit quand et où migrer. Il peut suspendre son exécution à un moment arbitraire, transporter son code vers une autre machine et reprendre l'exécution sur une nouvelle machine [17].

Définition 2 : Un agent qui peut se déplacer entre différentes plateformes (hôtes) à différents moments tandis que l'agent stationnaire réside en permanence sur une seule plateforme (hôte) [20].

Définition 3 : Une entité logicielle mobile définie par un comportement, une autonomie, un aspect social au niveau de la communication, une réactivité, un degré de sécurité et une capacité à mémoriser l'information [22].

Définition 4 : Un agent mobile n'est pas lié au système où il commence son exécution, il a la capacité unique de se déplacer d'un système dans un réseau à un autre. La capacité de migrer permet à un agent mobile de se déplacer vers un système contenant un objet avec lequel l'agent souhaite interagir, puis de profiter du même hôte ou réseau que l'objet [14].

De ces divers définitions décrites ci-dessus, on peut déduire qu'un agent mobile est une entité logicielle autonome, capable de prendre des décisions et de se déplacer d'un environnement à un autre afin de coopérer et d'interagir avec d'autres agents pour réussir la tâche qui lui a été dédiée par l'utilisateur.

1.3.2 Composants, caractéristiques et qualités d'un agent mobile

a - Composants d'un agent mobile :

Un agent mobile contient les trois composants suivants [1] :

- **Zone du code** : cette partie contient une référence à tous les fragments de code pouvant être appelés lors de l'exécution de l'agent. Cette zone de code inclut deux références à des fragments appartenant à l'agent, à savoir un code spécifiant le comportement du composant d'application utilisé par l'agent et des références à des classes externes pouvant faire partie d'un lieu, d'un système ou d'une région, telles qu'un code de procédure qui implémente les services du système.
- **État d'exécution** : cette partie contient toutes les informations relatives à l'évolution de l'agent mobile qui lui permettent de reprendre ses activités après avoir migré vers un nouvel hôte.
- **Espace des données** : cette partie contient des références à des ressources externes accessibles par l'agent.

b - Caractéristiques d'un agent mobile :

Grâce à sa mobilité, un agent mobile a la capacité de migrer et de se déplacer d'un système à un autre afin d'interagir avec d'autres agents. Pour cela, le modèle agent mobile procure de nombreux avantages dans la construction d'un système distribué. Dans cette partie, nous allons décrire sept caractéristiques selon Lange et Oshima [26] qui motivent à l'utilisation des agents mobiles [14] :

- **Ils s'exécutent de manière autonome et asynchrone** : un agent mobile se déplace d'un système à un autre pour se rapprocher des informations nécessaires afin de réussir à accomplir sa tâche. Il est capable de prendre des décisions afin d'améliorer l'exécution de sa mission. L'agent mobile résout le problème des tâches qui nécessitent une connexion permanente entre appareils mobiles et réseau fixe, car une fois les agents sont envoyés, ils deviennent indépendants du processus qui les a créé et commencent à fonctionner de manière autonome et asynchrone. Un agent mobile a la capacité aussi de se cloner pour pouvoir s'exécuter sur différents systèmes en parallèle.
- **Ils s'adaptent dynamiquement** : les agents mobiles peuvent détecter leur espace d'exécution et réagissent de manière autonome aux changements. Un agent mobile doit être capable de communiquer avec les autres agents du système, agents locaux ou agents distants, afin d'échanger des informations et bénéficier de leur savoir-faire.
- **Ils réduisent la charge du réseau** : les systèmes distribués reposent souvent sur des protocoles de communication impliquant de multiples interactions pour accomplir une tâche donnée. Le résultat est l'augmentation du trafic sur le réseau. De nombreuses études [12] ont reconnu que les agents mobiles permettent aux utilisateurs d'organiser une conversation et de l'envoyer à un hôte de destination où les interactions ont lieu localement, ce qui signifie que le transfert des agents d'une machine à une autre à travers le réseau minimise et annule les communications distantes entre les clients et les serveurs. Les agents mobiles sont également utiles pour réduire le flux des données brutes sur le réseau. Lorsque de très grands volumes de données sont stockés sur des hôtes distants,

ces données doivent être traitées dans leur localité plutôt que transférées sur le réseau. La devise du traitement des données par agent mobile est simple : déplacer le calcul vers les données plutôt que les données vers le calcul.

- **Ils surmontent la latence du réseau** : les systèmes temps réel sont critiques, ils doivent réagir en temps réel aux modifications de leurs environnements. Le contrôle de tel systèmes via un réseau d'usine, par exemple, de taille importante implique des latences importantes. Pour les systèmes critiques en temps réel, ces latences ne sont pas acceptables. Les agents mobiles offrent une solution, car ils peuvent être envoyés depuis un contrôleur central pour agir localement et exécuter directement ses instructions.
- **Ils encapsulent des protocoles** : lorsque les données sont échangées dans un système distribué, chaque hôte possède le code qui implémente les protocoles nécessaires pour coder correctement les données sortantes et interpréter les données entrantes. Toutefois, à mesure que les protocoles évoluent pour répondre aux nouvelles exigences d'efficacité ou de sécurité, il est difficile, voire impossible, de mettre à niveau le code de protocole correctement. En conséquence, les protocoles deviennent souvent un problème hérité. Les agents mobiles peuvent, dans ce cas, se déplacer vers des hôtes distants pour établir des canaux basés sur des protocoles propriétaires.
- **Ils sont naturellement hétérogènes** : l'informatique en réseau est fondamentalement hétérogène, souvent du point de vue matériel et logiciel. Parce que les agents mobiles sont généralement indépendants de l'ordinateur et du transport (ne dépendent que de leurs environnements d'exécution), ils offrent les conditions optimales pour une intégration système transparente.
- **Ils sont robustes et tolérants aux pannes** : la capacité des agents mobiles à réagir de manière dynamique aux situations et événements défavorables facilite la création de systèmes distribués robustes et tolérants aux pannes. Si un hôte est en cours d'arrêt, tous les agents qui s'exécutent sur cette machine sont avertis. En plus, ils ont le temps de repartir et de poursuivre leurs opérations sur un autre hôte du réseau.

c - Qualités d'un agent mobile :

En plus des caractéristiques détaillées ci-dessus, la puissance d'un agent mobile résulte dans sa mobilité qui lui donne les qualités suivantes [27] :

- **Efficacité** : l'agent mobile est capable de migrer d'une machine à une autre. Il se déplace pour accéder localement aux données et effectue le traitement en local et ne déplace que les données utiles, ce qui réduit et diminue le trafic réseau.
- **Persistance** : une fois l'agent mobile est lancé, il devient indépendant du système qui l'a créé. Même si le système initial tombe en panne, l'agent mobile n'est pas affecté. La migration de l'agent et sa capacité de se déplacer entre les nœuds des réseaux joue un rôle très important pour collecter les informations nécessaires et atteindre le plus de ressources distantes possibles. Cette particularité est très utile pour les utilisateurs ayant des ordinateurs et des téléphones mobiles, puisqu'un utilisateur peut se connecter, initier l'agent avec la tâche à accomplir, se déconnecter et ensuite contrôler le progrès de l'agent de manière discontinue.
- **Communication "point à point" (P2P - peer to peer)** : un agent mobile est considéré comme une entité paritaire et peut ainsi choisir la position la plus appropriée

à ses besoins actuels, tantôt des clients tantôt des serveurs.

- **Tolérance aux fautes** : lorsqu'un serveur tombe en panne en cours d'exécution d'une requête, il est difficile pour un client de revenir à son état initial afin de se synchroniser à nouveau avec le serveur, contrairement à un agent mobile qui n'a pas besoin de maintenir des connexions permanentes, grâce à l'état d'exécution qui fait partie de ses composants et qui contient toutes les informations relatives à son évolution.

1.3.3 Mobilité de l'agent

La mobilité est une propriété très importante dans le modèle d'agent mobile, puisqu'elle lui permet de migrer et de se déplacer d'un hôte à un autre à travers le réseau. La mobilité d'une application se traduit par l'interruption de l'exécution de l'application sur le site source. Lors de l'utilisation d'un agent mobile dans une application répartie, on trouve deux types de mobilité, la mobilité faible et la mobilité forte comme le montre la figure 1.7.

Fig. 1.7 – Mobilité d'un agent mobile

Mobilité faible :

La mobilité faible [9, 13] contient deux éléments, le code ou le savoir-faire et les données courantes. Ce type de mobilité a pour but de transférer l'exécution d'une application de la machine source vers la machine destination, en passant par l'interruption de l'exécution de l'application sur le site source. Une fois le code et les données courantes sont sur le site de destination, l'application mobile reprend son exécution depuis le début avec les valeurs de ses données mises à jour. Les agents mobiles utilisant le degré de mobilité faible ne conservent pas les données traitées et les actions exécutées précédemment.

Mobilité forte :

Ce type de mobilité contient les éléments de la mobilité faible qui sont le code ou le savoir-faire et les données courantes, en plus de l'état d'exécution de l'application qui joue le rôle de la reconnaissance. Une fois qu'une application se déplace d'un site source vers un site de destination, elle peut reprendre son exécution à partir du point où elle a été interrompue sur le site de départ, car le code, les données utilisées et l'état courant de l'exécution de l'application du site source sont transférés vers le site de destination [9, 13].

Les agents mobiles avec une mobilité forte ont la capacité d'accéder et traiter les données à partir des éléments du réseau mais peuvent aussi rassembler de l'information et la préserver lors de la migration. Cette fonction permet la mise en œuvre des tâches plus complexes dans lesquelles les opérations de l'agent dépendent des données recueillies chez des hôtes visités précédemment. Dans la gestion du réseau, les agents mobiles avec une mobilité forte sont les plus adaptés pour les tâches de configuration, les tâches intensives de données impliquant le regroupement de données d'éléments du réseau fortement distribués et l'analyse de ces données durant l'exécution.

L'inconvénient majeur d'une mobilité forte est la taille de l'agent puisqu'il doit garder les informations précédentes lors de la migration de l'agent du site source au site de destination, donc la taille de l'agent varie en fonction de la tâche qu'il doit accomplir et la quantité d'information qu'il déplace.

La différence entre un agent avec un degré de mobilité faible et un agent avec un degré de mobilité forte se fait lors de la migration de l'agent et après la migration puisque :

- dans le cas d'une mobilité faible, une fois l'agent migre vers le site ou l'hôte de destination il est relancé à nouveau et les valeurs de ses variables sont restaurées. Son exécution commence à partir du début ou à partir d'une procédure spécifique.
- par contre, dans le cas d'une mobilité forte, non seulement le code et les données sont déplacés mais aussi l'ensemble de l'état d'exécution pour pouvoir relancer l'exécution de l'agent à partir du point où il a été stoppé avant sa migration.

Après avoir défini la technologie d'agent mobile, ses qualités, ses composants et les degrés de mobilité, nous allons discuter, à présent, l'architecture du modèle d'agent mobile. Après, nous allons nous intéresser aux efforts de standardisation du modèle d'agent mobile.

Malgré les nombreux avantages du modèle agent mobile, la mobilité de l'agent présente certains dangers, particulièrement côté sécurité. Lors de sa migration, l'agent est confronté à des environnements et des plateformes différentes de la plateforme initiale qui peuvent nuire à son fonctionnement si des précautions de sécurité adéquates ne sont pas mises en place. Nous allons détailler cet aspect de la sécurité par la suite dans le chapitre 3. Dans ce qui suit, nous allons nous intéresser aux normes dédiées aux systèmes à base d'agents mobiles afin de garantir l'interopérabilité des systèmes.

1.3.4 Normalisation technique

Pour assurer le bon fonctionnement des agents mobiles, plusieurs plateformes d'exécution ont été créées et commercialisées. Certes, ces plateformes permettent de concevoir et de créer

un agent mobile, elles sont aussi capables de réceptionner et d'activer le code d'un agent et sa migration vers une autre plateforme. Mais, le problème qui se pose est que l'agent mobile ne peut pas être exécuté dans une plateforme qui exécute un système différent de sa plateforme initiale. Cette variété est due à plusieurs variables, telles que : la structure de l'agent et le domaine d'utilisation de ce dernier.

Pour cela, les chercheurs étaient dans l'obligation d'établir une standardisation des fonctionnalités et des concepts de plateformes où l'agent va être exécuté et en même temps lui assurer un niveau d'interopérabilité [1] [28] afin qu'il puisse communiquer et interagir avec d'autres plateformes et systèmes existants sans restriction. Dans ce cadre, deux normes se sont imposées : la norme MASIF [29, 31] et la norme FIPA [30, 31].

a - La norme MASIF (Mobile Agent System Interoperability Facilities specifications)

Depuis l'apparition de l'OMG (Object Management Group) [2] [32], des plateformes d'agents mobiles ont été développées, construites sur différents systèmes d'exploitation, basées sur différents langages de programmation et technologies. Même de nouveaux langages ont été réalisés, exclusivement conçus pour le support des agents mobiles. Cependant, des tendances communes peuvent être remarquées : les langages de programmation basés sur des interprètes comme Java constituent la base de la plupart des plateformes d'agents d'aujourd'hui, et plusieurs approches sont associées à l'intégration d'agents mobiles et d'intergiciels [3] [33] basés sur RPC (Remote Procedure Call) qui est un protocole réseau permettant de faire des appels de procédure sur un ordinateur distant à l'aide d'un serveur d'applications, comme CORBA (Common Object Request Broker Architecture) [4] [34].

Plusieurs exigences fondamentales ont été identifiées en raison des expériences acquises au cours des activités de recherche et développement. Ces exigences couvrent les sujets suivants :

- support de gestion,
- support de sécurité,
- support de mobilité,
- support d'identification unique,
- support de transaction,
- support de communication

En raison de ces exigences, l'OMG a lancé une demande de proposition de norme pour les agents mobiles en novembre 1995. La soumission correspondante du MASIF a été adoptée par l'OMG en février 1998 [35].

1. terme informatique qui signifie la capacité de permettre à divers produit ou système à fonctionner avec d'autres systèmes existants ou du futur sans restriction d'accès ou de mise en œuvre.
2. le groupe OMG est un consortium américain à but non lucratif et qui a pour objectif de standardiser et promouvoir le modèle objet sous toutes les formes.
3. ou middleware, est un logiciel tiers qui permet de créer un réseau d'échange d'information entre différentes applications informatiques.
4. CORBA est une norme définie par des constructeurs de matériaux informatiques et des éditeurs de logiciels regroupés au sein d'OMG.

À l'origine MAF est une norme pour les systèmes d'agents mobiles adoptée comme technologie OMG. MASIF traite les interfaces entre les systèmes d'agents, et non entre les applications d'agents et le système d'agents et définit aussi des paramètres dans le profil de l'agent pour spécifier ses exigences sur le système de l'agent récepteur. Cette norme est composée d'un ensemble de définitions d'interfaces fournissant une interface interopérable pour les systèmes d'agents mobiles. Afin de résoudre les problèmes concernant l'interopérabilité, les interfaces ont été définies au niveau du système de l'agent plutôt qu'au niveau de l'agent. MASIF s'est concentrée sur la définition d'un cadre conceptuel, de services et d'interfaces pour l'interopérabilité basée sur CORBA entre des plateformes d'agents mobiles hétérogènes. MASIF aborde aussi de manière approfondie la sécurité.

L'idée derrière la norme MASIF est d'atteindre un certain degré d'interopérabilité entre les plateformes d'agents mobiles de différents fabricants sans imposer de modifications radicales des plateformes. MASIF n'est pas destinée à construire la base d'une nouvelle plateforme d'agent mobile. Au lieu de cela, les spécifications fournies doivent être utilisées comme complément aux systèmes déjà existants [35].

MASIF ne standardise pas les opérations d'agent local comme la sérialisation / désérialisation [5] et l'exécution de l'agent, mais standardise plutôt :

- **la gestion des agents et des systèmes d'agents différents** : en définissant des opérations suivantes de manière standard : la création d'un agent, la suspension d'un agent, la reprise de l'agent et la fin de la mission d'un agent.
- **le transfert de l'agent** : la migration et le déplacement d'un agent entre différents systèmes d'agents, ce qui permet la création d'un environnement hétérogène.
- **le nommage des agents et des systèmes d'agents** : en normalisant la sémantique et la syntaxe des noms des systèmes d'agents et des agents pour pouvoir s'identifier.
- **la syntaxe du type de système et de l'emplacement d'un agent** : ces aspects sont standardisés pour que les systèmes d'agents puissent se localiser. Le transfert d'agent ne peut aboutir que si le type de système d'agent peut prendre en charge l'agent.

MASIF accorde à un système d'agents d'apercevoir les exigences de l'agent sur son système, et fournit les fonctionnalités requises pour le premier niveau d'interopérabilité à savoir le transport des informations sur l'agent.

Une plateforme d'agent compatible MASIF est accessible via deux interfaces standardisées spécifiées au moyen du langage de définition d'interface CORBA : MAFAgentSystem et MAFFinder. Ces interfaces fournissent des opérations fondamentales pour la gestion des agents, leur suivi et leur transport. Il s'agit d'une architecture logicielle pour le développement des composants qui sont assemblés pour construire des applications complètes. Ses deux interfaces sont définit comme suit [36] :

- MAFAgent System : cette interface a pour rôle d'assurer et de garantir le transfert de l'agent, lors de son envoi et de sa réception. Elle définit aussi les opérations pour la gestion du cycle de vie des agents telles que : la création de l'agent, sa suspension, sa reprise et la fin de sa mission.

5. le concept de sérialisataion et désérialisation est utilisé chaque fois que les données relatives aux objets doivent être envoyées d'une application à une autre. La sérialisation est utilisée pour exporter les données d'applications dans un fichier.

- MAFFinder : cette interface se charge de l'enregistrement et de la localisation des agents, des places et lieux d'agents et des systèmes d'agents.

La norme MASIF s'appuie sur des services CORBA à savoir :

- service de nommage CORBA : lie les noms à des objets CORBA. Les applications utilisent ce type de service pour rechercher un objet à partir de son nom.
- service de cycle de vie CORBA : définit et présente les services et les conventions de la gestion des objets CORBA qui peuvent être créés, copiés, supprimés, déplacés par ce service. Puisque le transfert de l'état de l'agent est obligatoire, le service de cycle de vie doit être associé au service d'externalisation.
- service d'externalisation (sérialisation) : procure un mécanisme normalisé d'enregistrement de l'état d'un objet sur un flux de données et de sa restauration à partir de ce dernier.
- service de sécurité : ce service est composé de service de sécurité prioritaire, qui incluent des fonctionnalités d'authentification et de contrôle d'accès, en plus des services de sécurité conformes au niveau de la norme CSI (Computer Security Institute). Les agents mobiles et les systèmes d'agents disposent de plusieurs politiques de sécurité selon leurs activités, en fonction de l'authenticité des parties qui interagissent entre elles, l'autorité et la classe de l'agent et d'autres facteurs.

La norme MASIF [37] couvre un ensemble minimal de fonctionnalités car elle est conçue comme un complément aux plateformes d'agents existantes plutôt qu'une base de systèmes complètement nouveaux. La fonctionnalité d'une plateforme compatible MASIF est accessible via les interfaces IDL [6] (Interface Definition Language). Ces interfaces fournissent, entre autres, des méthodes de gestion (c'est-à-dire la création, la suspension, la reprise et la résiliation), de transport et de suivi des agents.

b - La norme FIPA (Foundation for Intelligent Physical Agents)

En 1996 en Suisse, FIPA [37] qui est une organisation internationale de normalisation de la société IEE Computer Society a été créée afin de promouvoir l'interopérabilité des agents et des services qu'ils peuvent représenter. Elle a été prévue afin de réaliser des spécifications de normes logicielles pour les agents et les systèmes d'agents pour pouvoir communiquer et coopérer, fonctionner normalement et de manière efficace. FIPA est une technologie générique de base pour différents domaines d'applications où les développeurs peuvent créer des systèmes à base d'agents avec un niveau d'interopérabilité élevé.

La norme FIPA décrit un modèle de base d'une plateforme d'agents. Ce standard mentionne quelques agents obligatoires pour la gestion de la plateforme avec le rôle de chacun, spécifie aussi le langage de communication, le contenu et l'ontologie du langage des systèmes multi-agents. FIPA a réalisé de nombreux documents et publications qui décrivent les spécifications qui permettent aux agents et aux systèmes d'agents de communiquer et d'interagir entre eux, on trouve FIPA97 [38, 39], FIPA98 [36] et FIPA2000 [40] qui est la plus récentes.

En général, les spécifications de la norme FIPA, comme le montre la figure 1.8, sont déter-

6. un langage voué à la définition de l'interface de composants logiciels qui permet de faire communiquer des modules implémentés dans des langages différents ou déployés à travers un réseau sur des systèmes hétérogènes.

minées selon cinq catégories qui sont décrites ci-dessous :

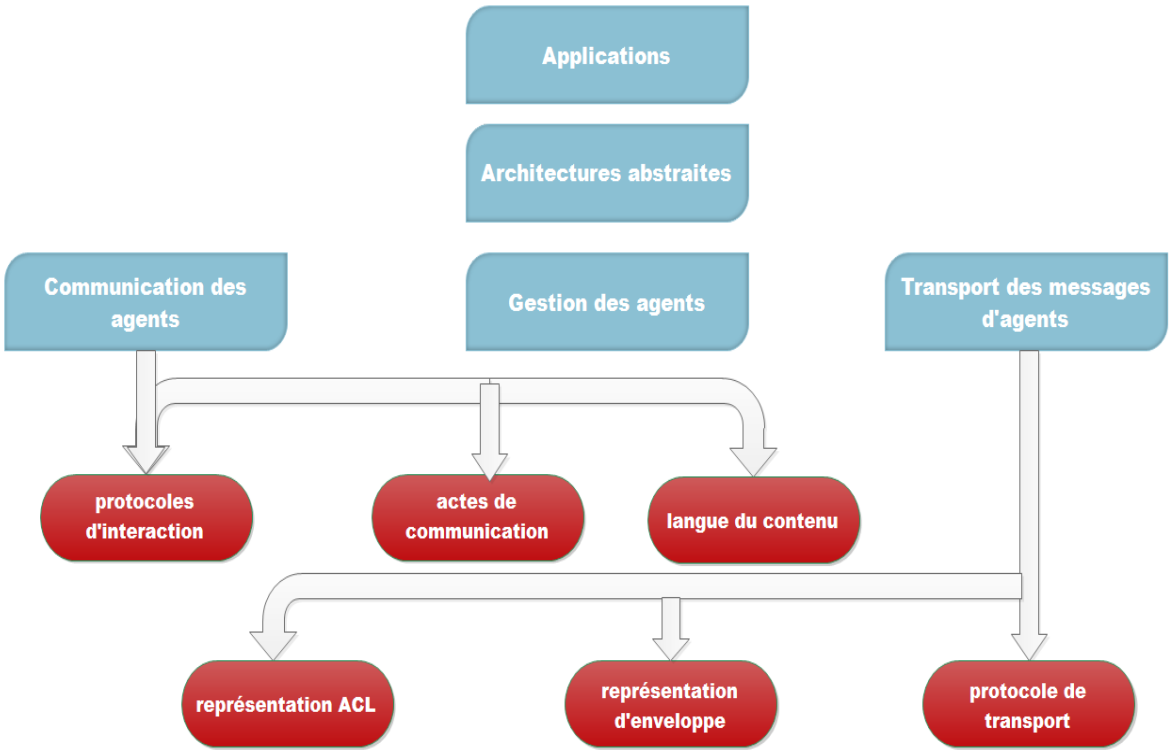

Fig. 1.8 – Spécifications définies par la norme FIPA

- **applications** : définit des exemples de domaines d'applications sur lesquels des agents FIPA peuvent être déployés.
- **architectures abstraites** : définit les entités abstraites nécessaires au développement d'un environnement d'agents.
- **gestion des agents** : concerne la gestion et le contrôle des agents entre eux et dans les plateformes.
- **communication des agents** : définit des messages de communication ACL (Agent Communication Language), des protocoles d'échange de messages, etc.
- **transport des messages d'agents** : définit la représentation et la transmission des messages à travers différents protocoles du réseau.

Selon [41], la norme FIPA se répartie en deux types de spécifications :

- **spécification de type formatif** : ce type représente le comportement externe de l'agent et garantit l'interopérabilité avec les autres sous-systèmes spécifiés de la norme FIPA.
- **spécification de type informatif** : ce type constitue un guide pour l'industrie qui s'intéresse à appliquer le standard FIPA.

Structure d'un message FIPA échangé entre deux agents : La figure 1.9 est un exemple montrant la transmission d'un message FIPA entre deux agents A et B.

Tout d'abord, l'agent A débute par créer le corps du message, nécessaire pour définir l'échange d'information qui sera fait, comme montré sur la figure 1.9. Il comporte toutes les

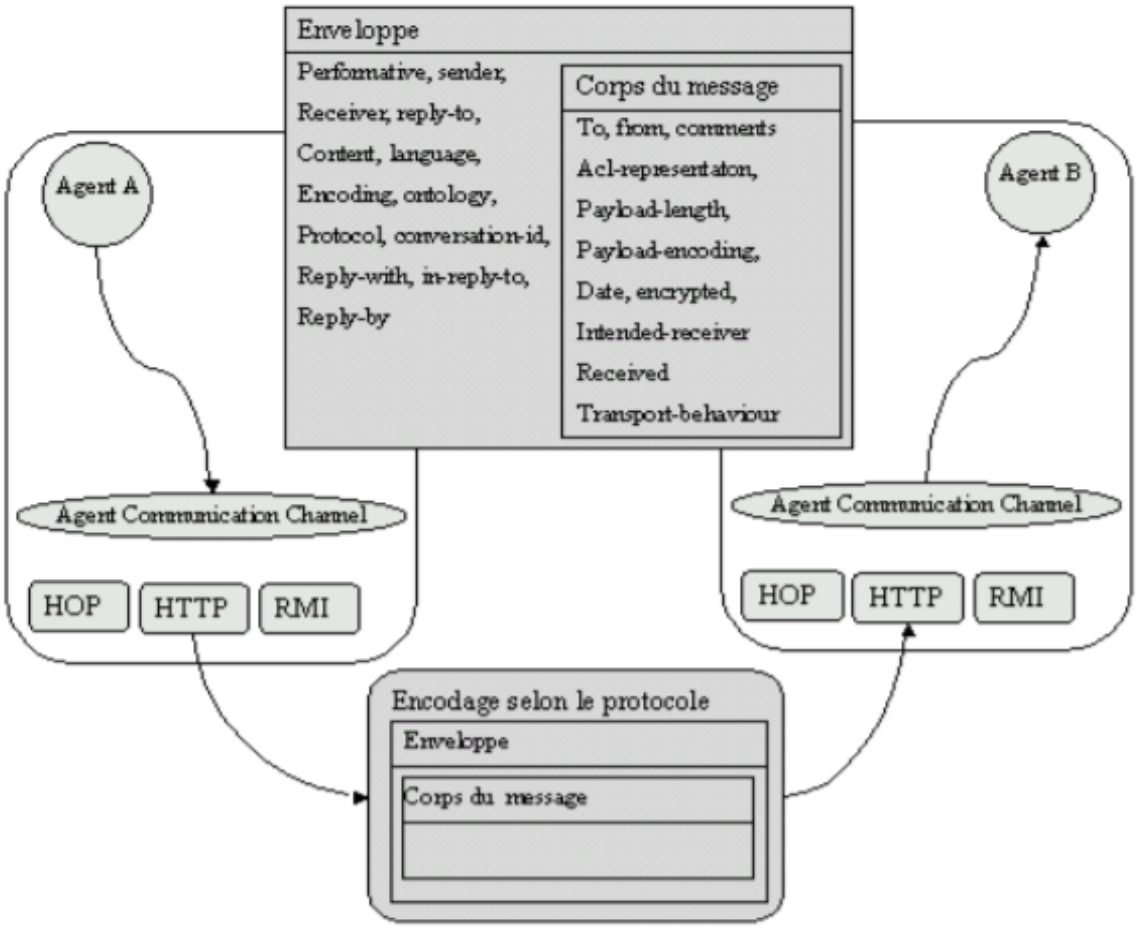

Fig. 1.9 – Envoi et réception de message entre deux agents de FIPA [42]

informations relatives à l'agent émetteur et l'agent récepteur. L'enveloppe rassemble les informations concernant les protocoles de transport, encodage, etc. Ensuite, l'agent émetteur A délègue l'envoi du message au Message Transport System qui, selon le protocole utilisé par l'agent émetteur (ici, HTTP) choisit Message Transport Provider pour pouvoir communiquer avec la plateforme de l'agent destinataire B.

La dernière version de FIPA (FIPA2000) [40] s'intéresse surtout à la spécification des exigences et des besoins en plus des technologies qui accordent à l'agent de tirer avantage et profiter de la mobilité. Dans ce cadre, les efforts de la norme FIPA se sont focalisés sur la spécification de ce qui suit :

- **Les protocoles de mobilité** : cette spécification réunit un ensemble de protocoles pour répondre à tous types et formes de mobilité. Ils sont nécessaires pour réaliser les opérations de déplacement nécessaires notamment à la migration, au clonage (créer une copie de l'agent pour pouvoir s'exécuter en parallèle sur plusieurs plateformes) et à l'invocation de l'agent (créer un agent en invoquant et faisant appel aux protocoles d'appel de l'agent).
- **Le cycle de vie de la mobilité d'agent** : cette spécification est une extension du cycle de vie de l'agent stationnaire qui se caractérise par l'ajout d'un nouvel état nommé "Transit" et deux nouvelles actions nommées "Move" et "Execute" pour pouvoir entrer et quitter l'état. La figure 1.10 montre le cycle de vie d'un agent défini par FIPA.
  — Au début, l'agent est créé mais il n'est pas encore enregistré afin de pouvoir commu-

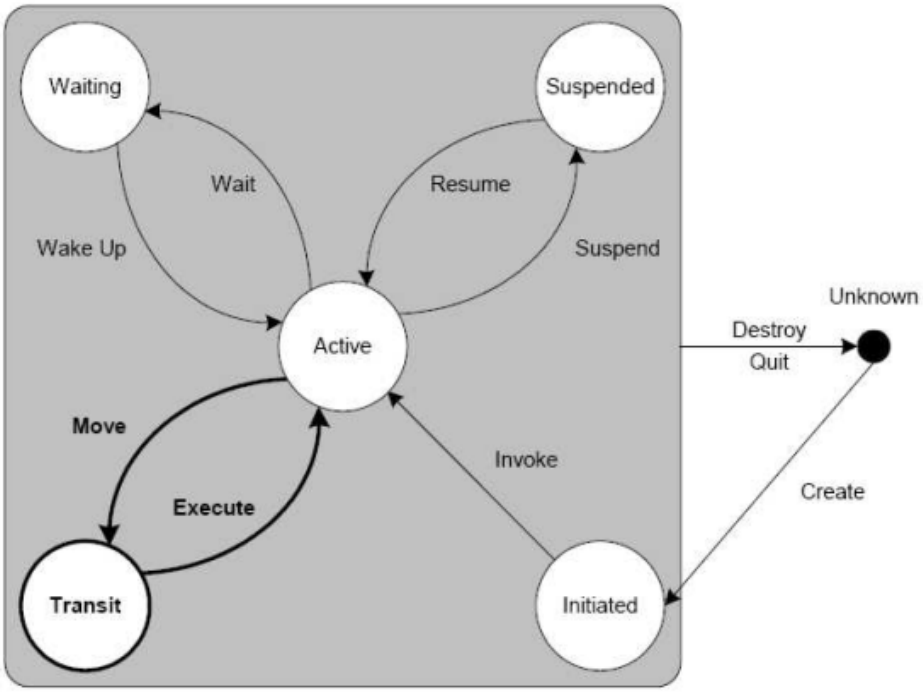

Fig. 1.10 – Cycle de vie d'un agent mobile défini par FIPA [41]

niquer avec d'autres agents ou d'autres plateformes.

— Une fois l'agent est enregistré, il devient capable de passer d'un comportement à un autre. Donc, il est actif et fonctionne normalement. Ici, l'agent peut être soit en état de suspension, soit en état d'attente, soit en état de transit.

- **L'ontologie de la mobilité d'agent** : l'ontologie est déterminée pour définir une sémantique et un vocabulaire propre pour le contenu des messages partagés entre les agents. Cette spécification fournit des extensions de l'ontologie FIPA-Agent-Management composée d'un ensemble de cadres caractérisant des classes d'objets et des fonctions pour la prise en charge de la mobilité. Chaque objet comprend de nombreux paramètres, chaque paramètre est caractérisé par une description textuelle qui détaille la sémantique de ce paramètre, une présence qui permet de connaitre si le paramètre est obligatoire ou facultatif et un type de la valeur du paramètre (entier, url, word, etc), en plus d'une liste comprenant les valeurs supportées par ce paramètre.

Pour assurer l'interopérabilité, la norme FIPA a abordé plusieurs spécifications et standards logiciels, mais elle n'a pas touché le côté sécurité dans le cas de la mobilité dans ses spécifications ni dans ses implémentations (FIPA-OS : FIPA-Open Source). Pour cela, plusieurs recherches ont été faites pour tâcher de concevoir des architectures sécurisées. Les auteurs dans [43] ont proposé une architecture accordant l'implémentation de deux services, un service qui traite la sécurité lors de la communication pour contrer l'attaque par écoute des plateformes externes, en plus d'un service pour la sécurité lors de l'exécution pour assurer la protection contre les accès non autorisés aux ressources et aux plateformes d'agents.

De nos jours, la norme MASIF et la norme FIPA sont appliquées dans le développement de plusieurs plateformes d'agents mobiles pour assurer la migration des agents. La norme MASIF fournit une représentation du concept d'un agent stationnaire, agent mobile, système de l'agent, en plus de l'état et l'emplacement de l'agent. La norme FIPA décrit les protocoles de la mobilité

et le concept du cycle de vie d'un agent mobile.

1.4 Fonctionnement d'un agent mobile

Le principe de fonctionnement d'un agent mobile repose sur sa capacité à se déplacer à travers le réseau, il ne reste pas sur la machine qui l'a déployée, en plus de son autonomie et sa capacité à communiquer avec d'autres agents. Dans cette section, nous allons décrire les étapes par lesquelles un agent mobile doit passer. Ensuite, nous allons discuter la manière utilisée par les agents mobiles pour communiquer entre eux, ainsi que les services requis pour l'exécution de l'agent.

1.4.1 Fonctionnalités d'un agent mobile

Pour qu'un agent mobile puisse fonctionner normalement, on doit prendre en considération les étapes suivantes :

- **création et initialisation d'un agent mobile** : la création d'un agent mobile consiste à lui déterminer un identifiant unique AID (Agent IDentifier) sur la plateforme, composé d'un nom de l'agent, du nom de la plateforme (nom du domaine ou l'adresse IP), ainsi que le numéro du port, pour pouvoir ensuite lui définir des méthodes. Une fois l'agent créé et initialisé selon les informations administrées par l'utilisateur, il doit passer au mode actif (déclencher l'exécution du code) pour pouvoir se déplacer et migrer d'une plateforme à une autre et pouvoir ainsi communiquer et interagir avec les autres agents.

  Pour la plateforme JADE par exemple (on va trouver plus de détails sur cette plateforme dans le chapitre 2, section 2.4), l'agent doit passer au mode actif pour pouvoir s'enregistrer auprès de l'agent AMS (Agent Management System) afin d'accéder aux fonctionnalités proposées par JADE.

  La création d'un agent mobile n'est pas liée à la mobilité et accepte soit une création locale, soit une création et exécution à la demande, soit une création et exécution à distance.

- **migration de l'agent** : grâce à sa capacité de se déplacer à travers le réseau, l'agent mobile s'approche des ressources distantes afin de pouvoir coopérer et interagir en local avec d'autres agents en vue d'accomplir sa tâche pour laquelle il a été créé. Lors de sa migration vers une nouvelle destination, l'agent peut entrer dans l'état de transit : ici, le système conserve les messages pour les envoyer à l'agent une fois arrivé à sa nouvelle destination. Comme nous l'avons expliqué précédemment, la migration d'un agent se fait soit selon une migration faible (l'exécution de l'agent est réinitialisée puisqu'il est transféré uniquement qu'avec son code et ses données) soit selon une migration forte (l'agent est transféré avec, en plus de son code et de ses données, de l'état de son exécution qui lui permet de reprendre son exécution à partir du point où il a été avant son déplacement).

  L'agent se déplace d'une machine à une autre afin de coopérer et d'interagir avec d'autres agents. Ce déplacement peut être volontaire suite à une décision prise par l'agent lui-même ou un déplacement envisagé avec un itinéraire à suivre. Ces deux types de dépla-

cements conduisent à une politique de migration, divisée en deux classes [44] : migration ciblée et migration libre.

— la migration ciblée : cette classe de migration est appliquée lorsqu'un agent désire collaborer et interagir avec d'autres agents ou avec un site distant qui sont déterminés et identifiés, mais leur localisation est indécise. Dans ce cas, l'agent essaie de se rapprocher le plus possible de cette plateforme cible suivant les informations collectées. La migration ciblée est adoptée par les applications et systèmes métiers, qui y recourent pour tirer profit des services strictement identifiés.

— la migration libre : cette deuxième classe est employée lorsqu'un agent se déplace d'un site distant à un autre et cherche l'opportunité d'interagir et de coopérer avec les agents se trouvant sur ce site distant. La migration libre est appliquée pour les environnements utilisant la gestion du contexte, où les composants changent constamment.

Une fois que l'agent mobile passe à l'état de migration, l'agent prend l'une des caractéristiques qui suivent :

agent actif / suspendu / en attente : l'agent est autorisé à exécuter les tâches qui lui sont administrées (sous formes de comportements), s'il est en état actif. Une fois l'agent est suspendu, ce qui signifie qu'il est en arrêt d'exécution, dans ce cas, aucun comportement ne sera exécuté. Si l'agent est en attente, cela signifie qu'il est bloqué et ne va reprendre son exécution que lorsque certaines conditions vont être accomplies.

clonage de l'agent : l'agent a la capacité de se cloner et de se multiplier afin de s'exécuter en parallèle sur plusieurs systèmes à la fois.

- **terminaison ou fin de l'agent** : cette étape est appelée une fois que l'agent termine les tâches qui lui ont été dédiées. Sur JADE, une fois l'agent a définitivement terminé son exécution il est supprimé. Le thread interne a terminé son exécution et l'agent n'est plus enregistré auprès de l'agent AMS.

Donc, une fois un utilisateur demande un service à une application qui est développée à base d'agents mobiles, l'application envoie les informations à propos du service voulu à la plateforme d'exécution des agents via une API (Application Programming Interface). Une fois l'agent qui est créé et qui est en mode actif reçoit ces informations, celles-ci sont envoyées sous forme de requête d'un serveur à un autre afin de trouver les informations nécessaires. Lors de son exécution, l'agent peut se déplacer d'une machine à une autre afin de récupérer les informations requises. Une fois sa mission est terminée, il passe ses informations collectées à l'application cliente d'origine.

1.4.2 Interactions entre agents mobiles

Dans la partie qui suit, nous allons discuter de la manière dont les agents communiquent entre eux, coordonnent et coopèrent afin de réussir leur tâche.

a - Communication entre agents :

La composante clé d'un système basé sur les agents mobiles est la communication. Les agents doivent impérativement être capables de communiquer avec les utilisateurs, avec les

ressources distantes et entre eux afin d'interagir, de coopérer, de négocier pour réussir des tâches spécifiques selon les besoins de l'utilisateur.

Pour cela, on peut classifier la communication entre agents en trois catégories [45] :

- **communication en local** : ce type de communication est réalisé seulement entre agents locaux pour assurer la transmission des messages sans interruption.
- **communication déléguée** : ce type enrichit la communication en mettant en œuvre des agents intermédiaires de type mobile ou statique selon les exigences, grâce auxquelles les agents peuvent discuter entre eux.
- **communication expressive** : ce type de communication est basé sur l'utilisation de l'ontologie définie par la norme FIPA vue précédemment, afin de déterminer la sémantique de l'échange et assurer ainsi une bonne compréhension entre agents lors du dialogue.

Selon [46], la communication entre agents est établie de deux manières :

- soit en appliquant des mécanismes qui admettent une communication directe, communication synchrone ou asynchrone.
- soit en utilisant des mécanismes de communication indirecte, en envoyant un message à un intermédiaire qui doit le transmettre au destinataire.

En plus de ces caractéristiques, la communication nécessite des protocoles au niveau de la couche transport du modèle TCP/IP, un langage de communication (langage commun entre agents pour pouvoir communiquer d'une manière flexible), protocole d'interaction (pour assurer l'interopérabilité, puisque l'agent migre d'une plateforme à une autre différente de celle qui l'a créée).

b - Coordination entre agents

La coordination est un processus dans lequel les agents s'engagent à assurer qu'une communauté d'agents individuels fonctionne de façon homogène et cohérente. De nombreuses raisons font que la coordination entre agents doit s'appliquer. Parmi ces raisons, on trouve :

- les objets des agents peuvent engendrer des interactions entre les actions des agents,
- les objets des agents peuvent être interdépendants,
- les agents peuvent avoir des capacités différentes,
- les objets des agents peuvent être obtenus rapidement si plusieurs agents agissent sur chacun d'eux.

c - Coopération entre agents

En plus de la communication et la coordination, s'ajoute la politique de coopération entre agents, afin d'assurer le déroulement normal des interactions à savoir : arriver à bout de leur chemin, sans interruptions prématurées (bien sûr, sauf dans le cas de panne ou erreur de système d'exploitation). Pour pouvoir réaliser et gérer la coopération entre agents tout en gardant le

contrôle sur l'exécution de l'agent, on distingue deux types de coopération qui sont appliqués sur les agents :

- la coopération directe : est une séquence d'appels de méthodes proposées par l'environnement d'exécution et qui est contrôlée et gérée par un mécanisme de coordination présenté au niveau applicatif. Elle est utilisée lors d'une interaction en local et une communication directe et entre agents. Les auteurs du travail [44] ont proposé le déploiement de cette coopération sous forme d'un tableau blanc anonyme effectué par un agent tiers. Ce tableau est implémenté à fur et à mesure que des pairs d'agents coopèrent et vont déposer le service effectué par chacun d'eux, ils peuvent aussi récupérer des informations contribuant à leur requête. Il faut que le système donne un coup de main pour que les agents puissent s'identifier entre eux et pour pouvoir présenter des coopérations avec les coordinations possibles comme le montre la figure 1.11.

  Donc, l'environnement doit être capable d'offrir un service sous forme d'annuaire local auquel l'agent peut y accéder de n'importe quelle plateforme lors de son départ, à l'instar du service de nommage défini par CORBA.

  Avantage : lors de la migration des agents, au lieu de traverser tous les nœuds du réseau pour collecter les informations recherchées, la coopération directe vient pour affiner la recherche et éviter le déplacement inutile des agents. Ce type de coopération (directe) ordonne à suivre les critères asynchrone et de localisation.

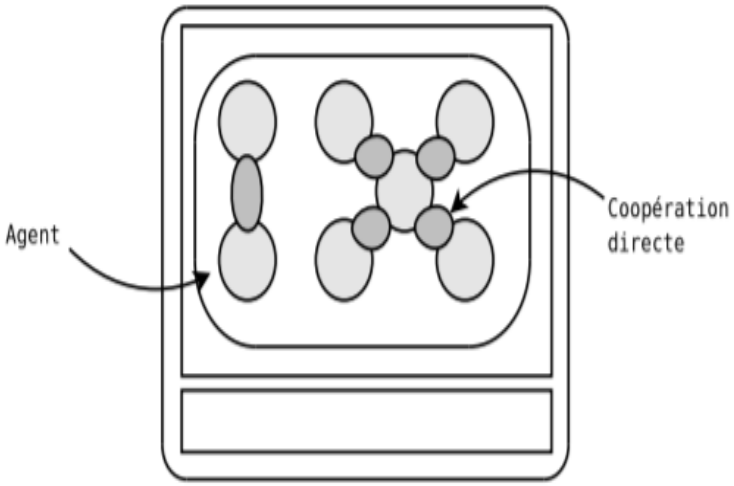

Fig. 1.11 – Coopération directe d'agents [44]

- la coopération indirecte : contrairement à la coopération directe qui est basée sur une communication en local, ici la communication est déléguée à des agents nommés agents intermédiaires qui sont considérés comme étant le support de cette communication lors de la migration de ces agents. Nous prenons comme exemple deux agents A et B qui souhaitent entamer une coopération indirecte, comme le montre la figure 1.12. L'agent B est l'agent qui va prendre le rôle d'intermédiaire entre l'agent A et l'agent C en migrant d'un site à un autre pour collaborer.
  Avantage : des fois, accomplir une coopération directe s'avère difficile sur des sites où les agents se déplacent plusieurs fois. L'agent n'arrive pas à trouver son partenaire pour pouvoir collaborer avec lui (changement d'environnement et de localisation). Pour cela, la coopération indirecte est mise en place. L'agent intermédiaire lors de sa migration d'un nœud à un autre à travers le réseau, laisse sur chacun les informations et les données collectées et qui sont évaluées comme étant des données essentielles.

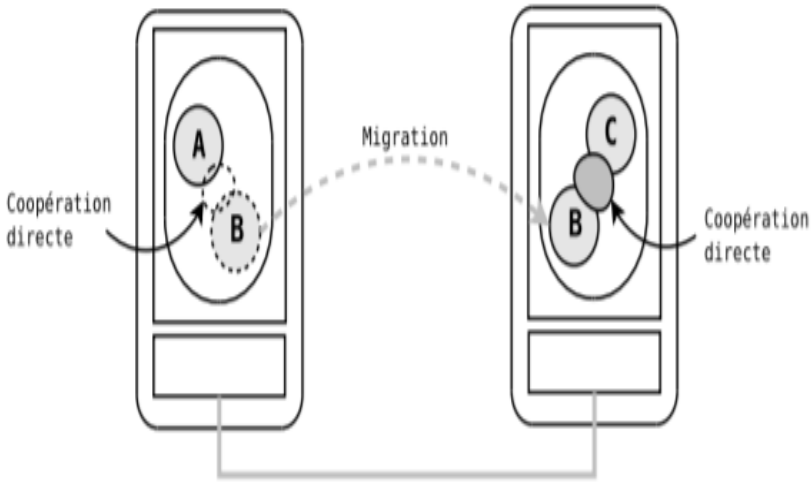

Fig. 1.12 – Coopération indirecte d'agents [44]

On ajoute que la coopération s'annule une fois que le lien de communication est rompu ou un des tiers s'est résilié.

d - Langage de communication

Les agents adoptent des langages de communication particuliers et spéciaux, fondés sur la théorie de l'acte de parole et qui admettent une séparation entre les actes de communication et le langage du contenu. A présent, le langage de communication le plus appliqué est **FIPA ACL** [30]. Ce langage dispose de contenu et de gestion des conversations par des protocoles d'interaction pré-établis. Le tableau 1.1 comprend les actions du langage FIPA-ACL avec la description de chacune.

Le langage FIPA-ACL est un langage similaire au langage KQML (Knowledge Query and Manipulation Language). Ce dernier est un langage et un protocole de communication de haut niveau qui supporte l'interopérabilité entre agents, orienté message pour la transmission et l'échange d'information. Le langage FIPA-ACL, comme son nom l'indique, est créé par le standard FIPA afin que les agents puissent communiquer entre eux. Il a une syntaxe similaire au langage KQML et il est caractérisé en définissant deux ensembles qui sont les suivants [47] :

- un ensemble de 21 actes de communication primitifs. Ces actes de base peuvent être combinés avec d'autres actes pour déterminer de nouveaux actes.
- un ensemble de messages prédéfinis qui sont approuvés et faciles à comprendre par chaque agent.

Ces 21 actes de communication sont reformés sous formes de 5 groupes qui définissent le langage FIPA-ACL :

- groupe pour le passage d'information et contient : inform*, inform-if (macro act), inform-ref (macro act), confirm*, disconfirm*
- groupe pour la réquisition d'information et contient : query-if, query-ref, subscribe
- groupe pour la négociation et comporte : accept-proposal, cfp, propose, reject-proposal

Action (performatif)	Description
sender	émetteur du message
receiver	destinataire du message
reply-to	participant à l'acte de communication
content	contenu du message (information transportée par l'action)
language	langage dans lequel le contenu est représenté
encoding	décrit le mode d'encodage du contenu du message
ontology	nom de l'ontologie utilisé pour donner un sens aux termes utilisés dans le contenu
protocol	contrôle la conversation
conversation-id	identificateur de la conversation
reply-with	identificateur unique du message, en vue d'une référence ultérieure
in-reply-to	référence à un message auquel l'agent est en train de répondre (préciser par l'attribut reply-with de l'émetteur)
reply-by	impose un délai pour la réponse

Tab. 1.1 – Langage FIPA-ACL [30]

- groupe ayant pour rôle la distribution des tâches ou l'exécution d'une action et comporte : request, request-when, request-whenever, agree, cancel, refuse
- groupe pour la manipulation des erreurs et contient : failure, not-understood.

La différence entre le langage KQML et FIPA-ACL est dans le choix et les définitions employées pour chaque langage afin de définir les états d'agents ainsi que dans la manière dont les agents sont gérés : en langage KQML, il traite ces tâches en tant que propositions de premier ordre ; en langage FIPA-ACL il aborde ces tâches en tant qu'actions rattachées aux actions.

Après avoir discuté la mise en œuvre d'un agent mobile et les différentes étapes du déroulement d'un agent mobile, nous présentons dans ce qui suit les services nécessaires afin d'exécuter un agent mobile.

1.4.3 Services d'exécution

Chaque environnement à base d'agents mobiles doit être capable d'accorder des services de base pour l'exécution de ces agents qui consistent en leur déploiement et leur mise en place à savoir : la création, la communication, la migration, l'exécution dans un environnement différent de celui où il a été créé. En plus de ces services de base, d'autres services s'ajoutent à eux afin d'assurer l'exécution sur des systèmes capables d'héberger et d'accepter les agents. Parmi ces services requis, on trouve :

a - Service de nommage

Afin que la communication entre agents puisse aboutir, ils sont contraints d'avoir un nom unique. Généralement, pour les systèmes à base d'agents mobiles, le nom de l'agent est créé à partir du nom de la machine où il est créé ou à partir de son adresse IP, d'un identifiant unique ou d'un numéro de port.

Le service de nommage comporte des conteneurs appelés NamingContext plus des objets qui sont affiliés au nom et aux références des composants du Serveur (nommé servant). Les NamingContext sont des objets CORBA et sont instaurés en arbre de nommage. La figure 1.13 illustre un exemple de l'arbre de nommage.

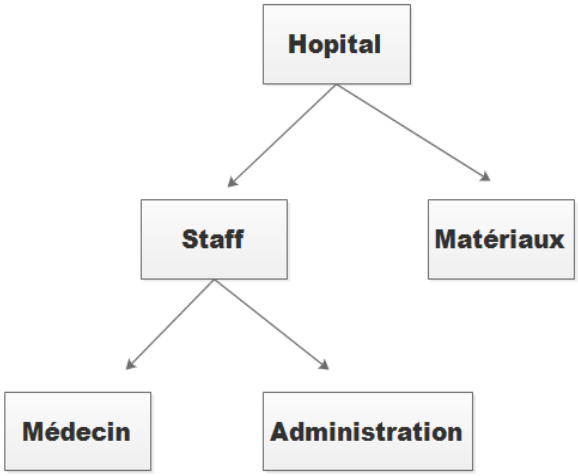

Fig. 1.13 – Exemple d'arbre de nommage CORBA

Sur cette figure, le nom va être sous la forme suivante : Hopital.nc/Staff.nc/Médecin.service. Comme on peut le remarquer, le nom est caractérisé par nom.type et du séparateur /.

b - Service de localisation

Contrairement aux systèmes traditionnels, la communication dans un environnement à base d'agents mobiles nécessite de localiser le positionnement de l'agent afin d'interagir avec lui vue que grâce à sa mobilité, il se déplace d'une machine à une autre en changeant le lieu de son exécution. Le service de localisation est primordial dans un système à base d'agents mobiles, pour que les agents puissent communiquer n'importe où et n'importe quand. Il est présenté sous forme de serveur de noms qui est composé soit par la localisation courante de l'agent soit par des informations qui nous approchent de sa localisation.

Des chercheurs ont mis au point quatre axes nécessaires pour la localisation de l'agent, et qui sont définis comme suit [48] :

- la mise à jour de l'agent sur la machine qui l'a créé, soit le site d'origine : chaque fois que l'agent se déplace d'une machine à une autre, le serveur de noms qui se trouve sur le site d'origine doit être mis à jour. Pour s'assurer que la mise à jour a été faite, le nombre de communications avec le serveur d'origine doit être égal au nombre de déplacements exécuté par l'agent.

- l'enregistrement : Chaque déplacement réalisé par l'agent est enregistré et défini au sein d'un serveur de noms centralisé et qui se trouve sur une machine autre que celle qui l'a créé. Le nombre de communications doit être égal au nombre de déplacements exécuté par l'agent plus la communication faite de la machine origine vers le serveur de noms.
- la recherche : le serveur de noms cherche la localisation d'un agent suivant un itinéraire bien défini. Pour cela on admet que l'itinéraire de l'agent est préalablement connu. Le nombre de communication peut différer de un (1) jusqu'au nombre de l'itinéraire étudié.

- la poursuite : localiser l'agent en surveillant un lien de poursuite. Le nombre de communications doit être au maximum égal au nombre de nœuds visités par l'agent.

c - Service de sécurité

Même si l'utilisation des agents mobiles apportent plusieurs avantages aux systèmes distribués, la mobilité de l'agent peut facilement devenir destructrice. Lors du déplacement et de la migration de l'agent d'une plateforme à une autre, il est dans l'obligation de coopérer et d'interagir avec d'autres environnements d'exécution différents de celui qui l'a créé.

La sécurité joue un rôle primordial. Pour cela, les systèmes à base d'agents mobiles doivent assurer des mécanismes de sécurité afin de garantir le bon fonctionnement de l'agent au niveau des plateformes visitées. La sécurité des systèmes à base d'agents mobiles vise à assurer la sécurité de l'agent contre un environnement d'exécution malveillant, assurer la sécurité entre agents, assurer la sécurité de l'environnement d'exécution contre un agent malveillant. Cette partie va être détaillée dans le chapitre 3 sur la sécurité des agents mobiles.

d - Service de sérialisation/ désérialisation

La sérialisation d'un agent mobile est un processus servant à encoder l'agent en suite d'octets, puis envoyer cette suite à travers le réseau. Notons que cette suite d'octets est obligatoire pour la sauvegarde. Ensuite, vient le processus de désérialisation permettant de décoder cette suite d'octets reçue afin d'en sortir une copie conforme de l'état de l'agent.

e - Service de traçabilité

Un agent mobile comme son nom l'indique est un agent capable de se déplacer d'un nœud à un autre en quête d'information et qui s'exécute de manière asynchrone. Pour cela, garder une trace sur l'itinéraire de l'agent et de ses déplacements effectués est considéré comme primordial pour pouvoir suivre son état d'exécution. Même si les agents ont la caractéristique d'être autonome, puisqu'ils peuvent changer l'ordre de l'itinéraire de leurs chemins par rapport au dysfonctionnement d'un nœud dans le réseau et prendre des décisions pour réussir leurs tâches, la traçabilité permet de grader sous contrôle le déroulement de leur fonctionnement.

f - Service de tolérance aux pannes

Un agent mobile s'exécute sur plusieurs machines, ce qui lui fait courir des risques pouvant altérer son fonctionnement normal et peut disparaître soudainement. Les systèmes à base d'agents mobiles doivent être dans l'obligation de fournir aux agents un environnement d'exécution tolérant les pannes et les fautes.

1.5 Conclusion

Dans ce chapitre, nous nous sommes focalisés sur la présentation et la description de la technologie agent mobile. Nous avons commencé par présenter les modèles de communication existants et les plus utilisés dans les systèmes distribués tout en invoquant leurs avantages et leurs inconvénients, afin de pouvoir mettre, par la suite, en exergue notre choix du travail avec la technologie d'agent mobile.

Ensuite, nous avons décrit le concept d'agent mobile en définissant ce dernier, ces composants, les caractéristiques et les qualités qui ont fait de l'agent mobile une technologie émergente répandue. Puis, nous avons discuté le rôle de la mobilité qui est le point far de ce paradigme, en plus de son architecture. Ensuite, nous avons montré les efforts de standardisation, les normes qui ont été dédiées à l'utilisation des environnements à base d'agents mobiles afin de garantir l'interopérabilité et la fiabilité.

Ensuite, nous nous sommes focalisés sur les éléments primordiaux pour le bon déroulement et l'exécution de l'agent. Nous avons commencé par expliquer son fonctionnement, puis la manière dont les agents communiquent et coordonnent entre eux afin de coopérer et de réaliser leurs tâches. Nous avons décrit ensuite les services nécessaires à l'exécution d'un agent mobile.

Le chapitre suivant va être sur l'intégration des agents mobiles dans les environnements intelligents, spécialement le domaine de soins de santé.

Chapitre 2

Proposition d'un modèle de service à base d'agents mobiles pour la transmission des données d'une ambulance intelligente

L'évolution de l'internet a permis au monde entier de diffuser, d'accéder et de partager une grande quantité d'information et de données. Cette large distribution de données implique un trafic accru sur le réseau, d'où la nécessité de nouvelles technologies et de nouveaux paradigmes. Ces dernières années, la technologie agent mobile a éveillé l'attention dans le monde de la recherche. Elle est considérée comme étant une technologie prometteuse pour le déploiement et la mise en place des applications distribuées. L'utilisation des agents mobiles satisfait le besoin des applications suscitant la récupération d'informations distribuées en rapprochant les données d'exécution des données à traiter ; l'agent mobile se déplace à travers le réseau afin de se rapprocher des ressources distances, s'exécuter sur les machines hôtes et ne revenir qu'avec les informations demandées.

Après la discussion entamée dans le chapitre précédent sur le concept de la technologie d'agent mobile, son évolution et en comparant ce modèle avec les modèles les plus utilisés. Dans ce chapitre, nous allons commencer par une partie qui présente les différents domaines utilisant les agents mobiles. Puis, nous allons donner des exemples dans le secteur médical, nous allons nous focaliser sur le domaine de soins de santé (healthcare). Ensuite, nous allons présenter notre proposition qui est un modèle de service pour les ambulances connectés basée sur le paradigme d'agents mobiles [2, 3].

2.1 Introduction

Les innovations technologiques actuelles telles que l'internet des objets, les systèmes intelligents, l'analyse des données augmentées et l'automatisation des environnements ont mené à la recherche de modèles et de paradigmes capables d'envoyer et de recevoir les données entre plusieurs entités de manière flexible, et de réaliser un traitement continu et en temps réel.

La technologie d'agents mobiles est une technologie prometteuse qui a réussit à se propager dans le domaine des systèmes distribués grâce à ses nombreux avantages et caractéristiques uniques. Le modèle d'agents mobiles répond aux besoins des nouvelles technologies telles que les

systèmes intelligents et des systèmes distribués, puisqu'il offre plus de flexibilité et d'avantages que le modèle traditionnel Client / Serveur. Il existe plusieurs raisons d'utiliser les agents mobiles, parmi lesquelles ils permettent de réduire la charge du réseau, surmonter la latence du réseau, encapsuler des protocoles, fonctionner de manière autonome et asynchrone, s'adapter dynamiquement et tolérer les pannes. En plus de ces caractéristiques, l'agent mobile a la capacité de migrer entre différents nœuds d'un réseau afin de s'approcher des informations nécessaires pour réussir sa mission et prendre des décisions sans la participation directe de l'utilisateur.

Plusieurs secteurs et domaines ont bénéficié de l'approche agent mobile, à savoir : les environnements intelligents pour suivre les utilisateurs d'applications lorsqu'ils se déplacent dans différents espaces intelligents, le domaine de la santé afin de gérer les données médicales et de surveiller les patients à distance en l'occurrence la télémédecine, domaine du commerce en automatisant plusieurs étapes dans le processus d'achat, etc.

Dans ce chapitre, nous allons commencer par présenter quelques domaines utilisant la technologie d'agents mobiles. Ensuite, nous allons nous concentrer sur l'utilisation des agents mobiles dans le domaine de soins de santé, précisément sur le nouveau concept de l'ambulance intelligente. Dans ce contexte, nous présentons notre proposition de service de modèle d'agent mobile dans une ambulance intelligente. Il s'agit d'une nouvelle approche qui utilise le paradigme des agents mobiles. Chaque agent mobile aura comme rôle de visiter le site serveur cible de l'application afin de collecter des informations pour son client, ce qui va permettre à ce dernier d'interagir localement avec le serveur, et donc de réduire le trafic sur le réseau en ne transmettant que les données utiles.

2.2 Domaines d'application de la technologie d'agents mobiles

Actuellement, les applications qui recourent à l'utilisation des agents mobiles sont des applications à forte intensité de données et qui nécessitent la communication avec des services distants. La technologie d'agents mobiles offre un nouveau modèle de conception pour les applications distribuées par rapport au modèle traditionnel Client / Serveur. Au lieu d'appliquer le concept de requête et réponse de ce dernier, l'agent mobile se déplace sur le réseau en migrant d'un nœud à un autre afin d'atteindre les services à exécuter. Le modèle à base d'agent mobile prend en charge les opérations déconnectées. Chaque nœud est l'équivalent d'un serveur dans le réseau. Pour cela, un grand nombre de nouvelles applications se développent autour de ce type de modèle.

Dans cette section, nous allons présenter quelques domaines connus qui ont tiré profit des qualités et avantages que procure la technologie d'agents mobiles.

2.2.1 Recherche d'information sur le web

Une des applications les plus importantes dans le domaine des agents mobiles est la recherche de l'information sur le web. L'utilisation croissante du web a touché quasiment tous les secteurs. Plusieurs sociétés sont passées du monde physique au monde virtuel en créant et

développant leur propre interface web, afin d'exposer leurs services et leurs produits, ce qui permet à l'utilisateur de chercher les bons plans sans avoir à se déplacer et perdre du temps.

Un moteur de recherche est un ensemble de programmes capables de lire et de consulter toutes les pages web, de créer un index des informations qu'il rencontre et les comparer selon la demande imposée par l'utilisateur, ensuite il transmet à ce dernier les résultats trouvés [49]. Le moteur de recherche joue le rôle d'intermédiaire entre les utilisateurs et les documents du web.

Contrairement aux modèles traditionnels où toutes les pages du web sont acheminées vers les moteurs de recherche ce qui engendre un trafic accru et inutile, l'intégration des agents mobiles dans le web permet d'affiner la recherche tout en économisant le trafic réseau. La technologie d'agents mobiles apporte plusieurs avantages, notamment le concept du code mobile qui constitue une architecture souple et dynamique, capable de migrer à travers le réseau d'une machine à une autre et interagit avec d'autres agents localement. Les agents mobiles peuvent se déplacer d'un site web à un autre tout en effectuant des transactions et en collectant des informations nécessaires.

Plusieurs études ont montré l'efficacité de l'intégration de la technologie d'agents mobiles dans la recherche d'information sur le web. Prenons comme exemple un utilisateur à la recherche d'un hôtel sur le net. L'utilisateur interfère avec de nombreux hôtels dans le but de trouver un bon plan selon les critères voulus. Le fonctionnement de manière traditionnelle se déroule comme suit :

- Les informations collectées sont expédiées sur la machine de l'utilisateur à travers le réseau.
- Une fois ces informations reçues, l'utilisateur réalise un tri sur les offres selon l'offre qui lui convient le plus.
- Lorsque l'utilisateur choisit l'offre qui lui correspond, les autres offres deviennent inutiles.

Donc, dans le cadre de diminuer les échanges inutiles à travers le réseau, l'intégration du modèle agent mobile devient envisageable. Le fonctionnement dans ce cas se produit comme suit :

- L'utilisateur crée et déploie un agent mobile qui a pour mission la recherche d'un hôtel selon les critères souhaités par l'utilisateur, en se déplaçant à travers différents sites afin de réussir sa mission.
- Grâce à sa capacité de migrer vers un site et de s'exécuter localement, l'utilisation de la technologie d'agents mobiles libère la charge réseau.

2.2.2 Commerce électronique

L'apparition des applications du commerce électronique a donné le jour à une multitude de modèles commerciaux centrés sur le web. Le commerce électronique établit des affaires d'achats et de ventes sur des réseaux interconnectés, il fournit aussi des services aux clients en plus des transactions d'informations avec d'autres organisations (inter) ou au sein de la même organisation (intra).

Le commerce électronique ou le e-commerce accentue les processus commerciaux, aide à

minimiser les coûts et rehausse la valeur commerciale. Pour cela, de nouvelles méthodes sont fournies afin de prendre en charge le commerce en ligne et de procurer des applications rentables et flexibles. Parmi les méthodes les plus adaptées dans ce domaine, on trouve les agents mobiles, qui constituent un atout majeur pour la création des systèmes et applications efficaces de commerce électronique.

La majorité des systèmes et applications de commerce électronique nécessitent des échanges d'informations de manière continue et un traitement de données intense. Comparé au modèle Client / Serveur, comme on a vu dans le chapitre précédent, les agents mobiles accordent une coopération et une interaction de manière rapide et efficace avec les services distants en plus de l'économie de la bande passante du réseau.

Comme nous le savons tous, une transaction commerciale a besoin d'un accès en temps réel à des ressources distantes et peut être même une négociation entre agents. Différents agents auront des objectifs différents et mettront en œuvre et appliqueront différentes stratégies pour atteindre ces objectifs. La technologie des agents mobiles est une solution très attrayante à ce genre de problème.

Plusieurs travaux appliquent les agents mobiles dans le commerce électronique et cela ne date pas d'aujourd'hui, puisqu'on trouve :

- un marché virtuel en ligne pour l'achat et la vente des biens nommé Kasbah [50], où l'utilisateur crée un agent autonome pour vendre et acheter des produits en son nom. L'utilisateur fournit à l'agent des critères et indications tels que : la quantité, le prix, le type de marchandise à vendre ou à acheter, etc. Ensuite, il lance l'agent dans ce marché. On y trouve deux types d'agents : les agents vendeurs qui ont pour rôle d'exposer leurs offres dans un tableau, puis les agents acheteurs dont le rôle consiste à filtrer les offres disponibles selon les critères imposés par l'utilisateur et passent ensuite à l'étape de négociation de l'offre sous le nom de son utilisateur. Ce marché virtuel assure le fonctionnement continu malgré la déconnexion de l'utilisateur.
- le papier [51] présente à partir d'une analyse des problèmes du commerce électronique traditionnel, la conception du e-commerce en se basant sur les agents mobiles. Les auteurs ont montré qu'un système de commerce électronique basé sur la technologie d'agents mobiles est un système intelligent, réparti et coopérant entre agents. Les clients sont aptes à réaliser des recherches et des achats en ligne, tandis que les commerçants gèrent leurs produits et boutiques en ligne. Cela est exécuté selon la répartition des rôles d'agents selon les besoins et aussi la définition de tâches spécifiques.

Avec ces avancées technologiques, un nouveau modèle de commerce électronique a fait son apparition nommé le M-commerce. Le M-commerce, plus précisément commerce Mobile est un type de commerce électronique où les opérations de transactions ventes et achats sont effectuées par le biais des appareils mobiles à savoir, des téléphones mobiles ou des assistants personnels (PDA : Personal Digital Assistant).

2.2.3 Environnement intelligent : soins de santé

De nos jours, les tendances technologiques actuelles comme les systèmes intelligents et l'Internet des objets, ont conduit à la recherche de nouveaux modèles permettant d'envoyer et de recevoir des données en plus des informations nécessaires en temps réel tout en diminuant la

charge du réseau. La technologie des agents mobiles a gagné du terrain même dans les systèmes intelligents. Dans cette partie, nous allons nous intéresser spécialement à la télémédecine et aux soins de santé intelligents qui constituent le vif de notre sujet, particulièrement l'intégration des agents mobiles dans les ambulances intelligentes.

Le secteur de la santé est un secteur primordial dans la société, car la qualité de vie des citoyens en dépend directement. Le secteur de la santé est non seulement largement distribué et fragmenté, mais il présente également une grande diversité et une forte autonomie locale. La programmation des agents mobiles est l'un des paradigmes le plus répandu pour le développement des applications sur Internet. Grâce à leur capacité de se déplacer, de coopérer et de s'adapter, ils sont déployés dans une variété d'applications dans le domaine de la santé, à savoir : l'aide à la décision, l'accès à des sources de données distribuées, ou la coordination d'activités des soins de santé, la gestion de données médicales, la récupération d'informations médicales, sans oublier la sécurité des données médicales.

Dans [52], les auteurs ont discuté l'importance de déployer des agents mobiles pour le développement des applications de soins de santé en ligne, dans le but d'échange de données et la surveillance à distance des patients n'importe où et n'importe quand. Selon les auteurs, la technologie d'agents mobiles est une technologie prometteuse qui grâce à sa flexibilité et sa capacité de coopérer et d'interagir entre agents, elle permet d'atteindre et de réussir la tâche requise avec un certain degré d'automatisation, ce qui correspond au nouvel environnement de soins de santé.

Afin de surmonter les problèmes d'interopérabilité dans des environnements hétérogènes, les auteurs dans ce papier [53] ont travaillé sur la conception et le développement d'un système d'information distribué basé sur l'utilisation d'une plateforme d'agents mobiles pour la surveillance fœtale automatisée en temps réel.

Pour pouvoir communiquer avec les patients à distance, la télémédecine a choisi d'utiliser des agents mobiles fonctionnant sur des réseaux et qui sont capables de se déplacer d'un serveur à un autre afin d'obtenir le résultat souhaité. A cet égard, les auteurs de [54] ont discuté la construction d'une télémédecine sécurisée basée sur l'architecture du réseau P2P en intégrant les agents mobiles.

Dans cette section, nous avons présenté différents domaines utilisant la technologie d'agents mobiles afin de bénéficier de ses nombreux avantages qu'elle apporte au sein du réseau, à savoir la diminution des coûts de communication sur un réseau à faible débit, la réduction de la charge réseau, etc.

Dans la prochaine section, nous allons présenter notre travail de recherche qui consiste à introduire la technologie d'agents mobiles dans les ambulances intelligentes afin de répondre efficacement au traitement, à la transmission de données et aux réponses rapides aux soins d'urgence. Notre approche proposée est un modèle de service d'agents mobiles capable de diagnostiquer l'état du patient une fois dans l'ambulance afin de lui administrer des soins d'urgence et lui trouver un hôpital approprié.

2.3 Proposition d'un modèle à base d'agents mobiles pour ambulance intelligente

Les améliorations apportées au service d'ambulance se focalisent sur la gestion des aspects information et documentation des incidents médicaux et des soins aux patients. Notre proposition consiste à déployer et à mettre en place un modèle de service d'agents mobiles capable de diagnostiquer l'état du patient et de trouver un hôpital approprié en utilisant l'interaction et la coopération entre ambulance et hôpital [2, 3].

2.3.1 Concept d'ambulance intelligente - ambulance du futur

a - L'objectif de l'ambulance intelligente

L'ambulance du futur est une ambulance intelligente, capable d'agir rapidement et d'arriver à l'heure chez les malades. Son rôle est de faire un diagnostic immédiat car les premières minutes dans certains cas cruciaux pour le patient, peuvent être décisives. Avec l'ambulance intelligente, nous ne sommes plus obligés d'amener des personnes à l'hôpital si elles n'en ont pas vraiment besoin.

La particularité de cette ambulance est sa capacité à réagir rapidement en cas d'urgence, à envoyer et à recevoir des informations sur le patient en contactant les médecins de l'hôpital pour un diagnostic plus approfondi et même pour un service de soins plus approprié. Pour cela, ces ambulances intelligentes devraient être équipées de plusieurs capteurs et puces RFID (Radio Frequency IDentification) afin de se conformer à la vision réglementaire, à la transmission et à la diffusion des informations.

Lorsque le patient est dans l'ambulance, l'infirmière prend ses paramètres de santé à savoir la fréquence cardiaque, la température actuelle, la pression artérielle, le niveau de sang, etc, qui seront envoyés à l'hôpital. Ces paramètres seront affichés dans l'unité de l'hôpital sur un ordinateur auquel les infirmières locales auront accès.

L'aspect le plus crucial de l'ambulance intelligente est sa capacité à envoyer et à recevoir des données entre les entités concernées dans les plus brefs délais. À partir de là, nous avons eu l'idée d'utiliser et d'appliquer le modèle des agents mobiles afin de proposer un modèle de communication efficace et flexible basé sur les agents mobiles.

b - Travaux connexes

À ce jour, il n'y a pas de normalisation des spécifications d'ambulances intelligentes. Dans cette partie, nous allons présenter divers travaux et idées existants sur l'ambulance intelligente.

Parmi ces travaux, on trouve le projet SAEPP (Smart Ambulance European Procurers Platforms) [55]. Ce projet est réalisé par un groupe de chercheurs dans différents domaines des hôpitaux et des soins de santé en collaboration avec d'autres organisations. Ils ont mis en place

un prototype d'ambulance équipé de nouvelles technologies permettant de réduire le nombre de patients transportés à l'hôpital en les soignant sur place.

De nos jours, le nombre de voitures en circulation a augmenté, et cela cause des problèmes d'embouteillage particulièrement durant les heures de pointes. Dans ces conditions, une ambulance ne peut pas arriver à temps pour fournir les premiers soins au patient et l'amener à l'hôpital dans les plus brefs délais. Pour surmonter cette situation, le papier [56] décrit un système qui a pour but de contrôler le signal de trafic. Les auteurs ont mis au point un algorithme permettant de localiser l'ambulance et de lui permettre d'atteindre rapidement l'hôpital en ayant accès à la programmation des feux de circulation.

Concernant l'exemple du projet Smart Pods [57], il a pour objectif de comprendre les modèles actuels des soins d'urgence et de fournir aux praticiens en soins d'urgence les équipements nécessaires, les véhicules et l'espace dont ils ont besoin afin d'effectuer une évaluation et un traitement plus efficace sur place, réduisant ainsi le nombre de patients admis à l'hôpital.

Des études ont mené à mettre en place un système multi-agents de coordination d'ambulances pour les services médicaux d'urgence. Les chercheurs dans [58] traitent la mise en point d'un système multi-agents permettant de gérer l'organisation répartie des ressources à Gérone afin de relever le défi lancé par l'administration des grands hôpitaux. Gérone est une ville au nord de l'Espagne qui a mis au point un système informatique facilitant la coordination entre les ambulances. Ce système sélectionne une ambulance entre plusieurs ambulances qui arriveront d'abord chez le patient, lui prodiguera les premiers soins et le transportera dans le centre médical approprié en coordonnant plusieurs ambulances et en tenant compte des tâches à accomplir ainsi que l'heure estimée d'arrivée de l'ambulance.

Après avoir illustré et présenté certains systèmes existants utilisant la technologie d'agents mobiles en télémédecine, dans la section qui suit, nous allons décrire notre modèle proposé pour les ambulances intelligentes et qui aidera à lever le voile sur la transmission de données en temps réel.

2.3.2 Utilisation des agents mobiles dans l'ambulance intelligente

Afin d'assurer les échanges entre l'ambulance intelligente et l'hôpital de manière efficace, flexible, nous choisissons pour notre proposition d'appliquer la technologie d'agents mobiles pour ses nombreux avantages qu'elle offre. Il existe plusieurs types d'agents mobiles, mais dans notre cas, nous allons adopter les types d'agents suivants :

- **L'agent léger** : comme son nom l'indique, c'est un agent de petite taille, capable de se déplacer très rapidement en raison du temps de transmission très court vu les données qu'il transporte. En plus de cela, l'agent léger va servir aussi d'agent intermédiaire entre l'agent local et l'agent lourd, en raison de sa bande passante à faible coût.
- **L'agent local** : cet agent va agir en local dans la plateforme initiale qui l'a créée pour le modèle proposé.
- **L'agent lourd** : cet agent est nommé lourd en raison de la taille du code exécutable et de celle de données transportées. L'agent lourd exécute une tâche qui demande de longues périodes de traitement, telle que la recherche dans la base de données des patients, qui contient toutes les informations nécessaires au sujet des patients.

Dans notre travail, nos objectifs sont divisés en deux parties : les objectifs immédiats et les objectifs une fois que le patient arrive à l'hôpital.

Les objectifs immédiats en route vers l'hôpital, consistent à :

- connaître l'hôpital auquel le patient est associé grâce au nom du patient pour y diriger directement l'ambulance.
- connaître les soins d'urgence à administrer au patient selon son état actuel, dans l'ambulance avant son arrivée à l'hôpital, en fonction de ses conditions antérieures et des informations collectées à partir de son dossier médical.

Les objectifs une fois arrivé à l'hôpital comportent :

- préparer à l'avance la salle de réanimation et le personnel du service adéquat.
- mettre en place un dispositif adéquat selon l'état déclaré en avance du patient.

Nous allons présenter deux modèles de services, le premier (figure 2.1 et figure 2.2) concerne le cas où le patient a des antécédents médicaux dans un hôpital approprié, le second cas (figure 2.3 et figure 2.4) concerne un patient qui n'est pas rattaché à un hôpital approprié.

2.3.3 Premier cas d'utilisation - patient avec antécédents médicaux

Pour le premier cas, nous allons présenter et détailler le modèle de service, le patient est rattaché à un hôpital où il a des antécédents médicaux.

Comme on peut voir sur la figure 2.1, il fonctionne selon les étapes suivants :

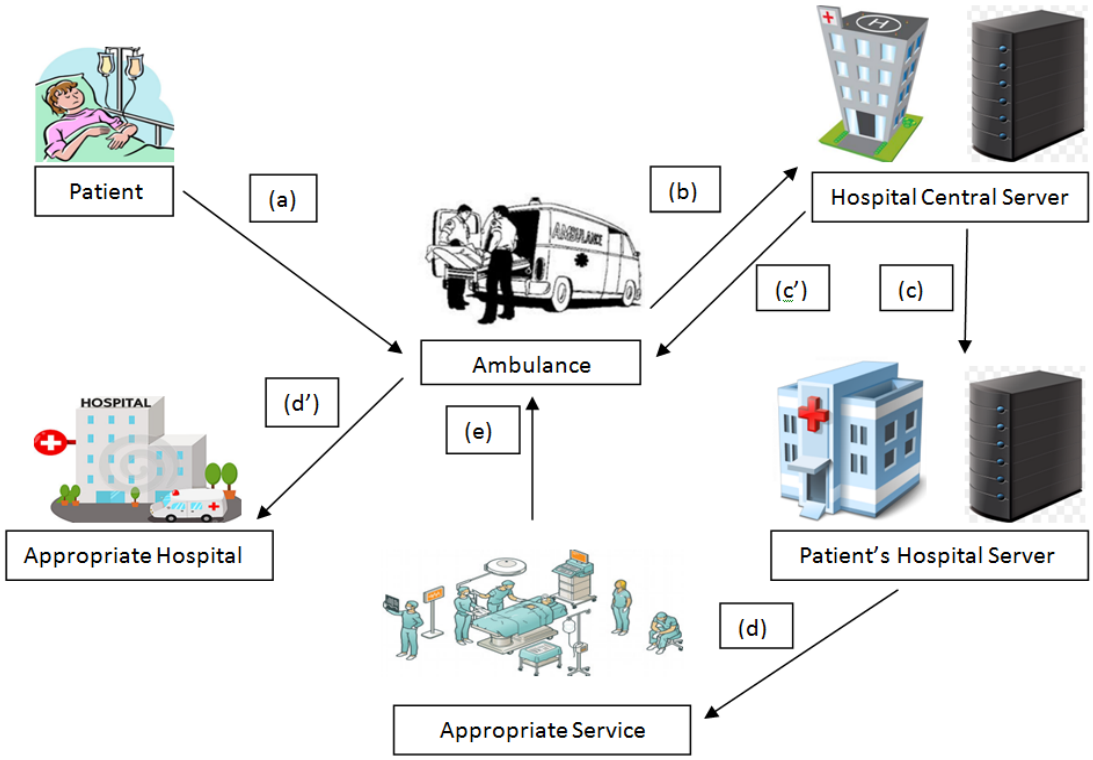

Fig. 2.1 – Architecture proposée pour le premier modèle de service

- (a) Une fois que le patient est pris en charge par l'ambulance, l'infirmière l'identifie et prend ses paramètres de santé à savoir : le rythme cardiaque, la température corporelle, la pression artérielle, le niveau de sang.
- (b) Ces données sont envoyées au serveur central des hôpitaux afin de voir si le patient est rattaché à un hôpital ou non.
- (c') Le serveur central des hôpitaux envoie à l'ambulance le nom de l'hôpital où le patient est rattaché.
- (c) Le serveur central des hôpitaux envoie à l'hôpital approprié le nom du patient.
- (d) Le service approprié recherche le dossier médical du patient et en informe le personnel adéquat.
- (d') L'ambulance conduit le patient à l'hôpital approprié.
- (e) Le service approprié envoie à l'infirmière qui se trouve dans l'ambulance les premiers soins à administrer au patient en attendant son arrivée.

Description du modèle proposé en intégrant les agents mobiles :

Le diagramme présenté sur la figure 2.2 montre le rôle que joue le modèle d'agents mobiles dans la transmission des données, afin d'obtenir une réponse précise dans les brefs délais.

- (1) Lorsque le patient est dans l'ambulance, l'infirmière prend note son nom et son état actuel à savoir les paramètres de santé tels que la fréquence cardiaque, la température corporelle, la pression artérielle, le niveau de sang.
- (2) Ces données prises par l'infirmière sont enregistrées dans l'agent local de l'ambulance.
- (3) L'agent léger (LW) récupère le nom du patient et ses données actuelles.
- (4) Et grâce à sa capacité de migration, l'agent léger migre vers le serveur central des hôpitaux, où sont stockées les données sur chaque patient, à savoir son nom et l'hôpital auquel il est rattaché.
- (5) L'agent léger recherche dans la base de données si le nom du patient est déjà enregistré auprès d'un hôpital approprié.

Une fois trouvé et grâce à la capacité de l'agent mobile à se cloner afin de s'exécuter en parallèle sur plusieurs machines, l'agent va se cloner pour pouvoir effectuer deux tâches en même temps :

- (6') Le nom de l'hôpital auquel le patient est rattaché est envoyé à l'ambulance,
- (7') Une fois l'agent léger arrive à l'ambulance, le nom de l'hôpital approprié est enregistré dans l'agent local de celle-ci pour y amener le patient directement.
- (6) La 2 ème tâche : l'agent léger cloné migre vers l'hôpital du patient afin de rechercher le service approprié lui permettant d'obtenir le dossier médical du patient.
- (7) L'agent léger va questionner les agents lourds qui vont rechercher l'information voulue, à savoir le service approprié à chaque patient.
- (8) L'agent migre vers le service approprié avec l'état actuel du patient.

- (9) L'agent léger recherche le dossier médical du patient.
- (10) L'agent léger migre vers l'ambulance avec les recommandations de soins immédiats pour le patient en attendant son arrivée à l'hôpital.
- (11) Préparation du personnel approprié pour accueillir le patient une fois arrivé à l'hôpital.

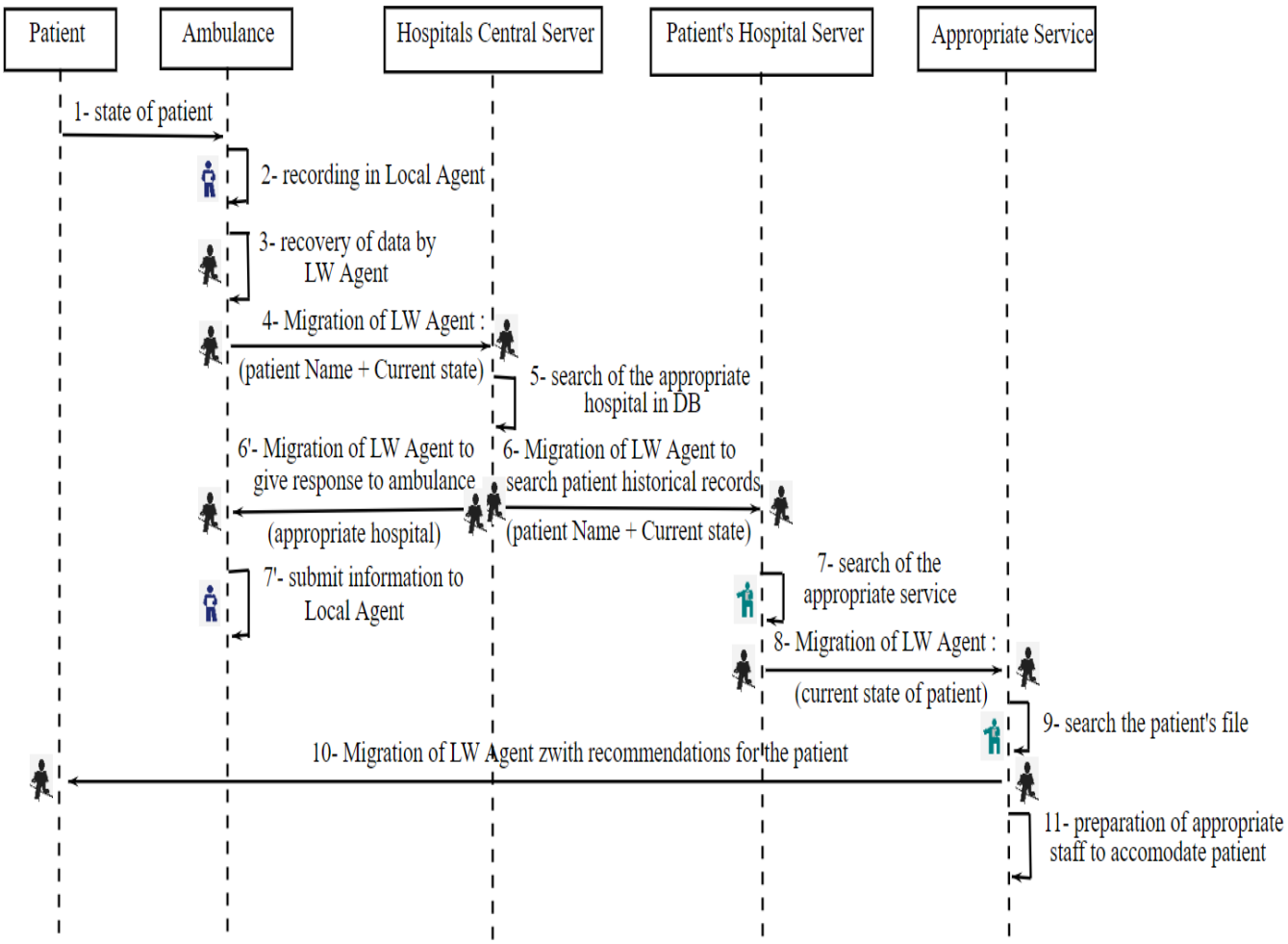

Fig. 2.2 – Diagramme de transmission des données à base d'agents mobiles pour le 1er cas

2.3.4 Deuxième cas d'utilisation - patient sans antécédents médicaux

Dans le cas où le patient n'a pas d'antécédents médicaux, le modèle de service est illustré sur la figure 2.3 :

- (a) Lorsque le patient est dans l'ambulance, l'infirmière prend ses paramètres de santé tels que : la fréquence cardiaque, la pression artérielle, la température du corps et le niveau de sang.
- (b) Ces paramètres sont envoyés au serveur central des hôpitaux afin de savoir si le patient est rattaché ou non à un hôpital particulier.
- (c) Le serveur central des hôpitaux constate que le patient n'a pas d'antécédents médicaux. Il conseille donc à l'ambulance, à l'aide des données reçues, de l'envoyer à un hôpital approprié.

- (d) Le serveur central des hôpitaux envoie les données du patient à l'hôpital indiqué en fonction de l'état actuel.
- (e) L'hôpital indiqué déclare l'arrivée du patient dans quelques instants et prépare le service de réanimation.

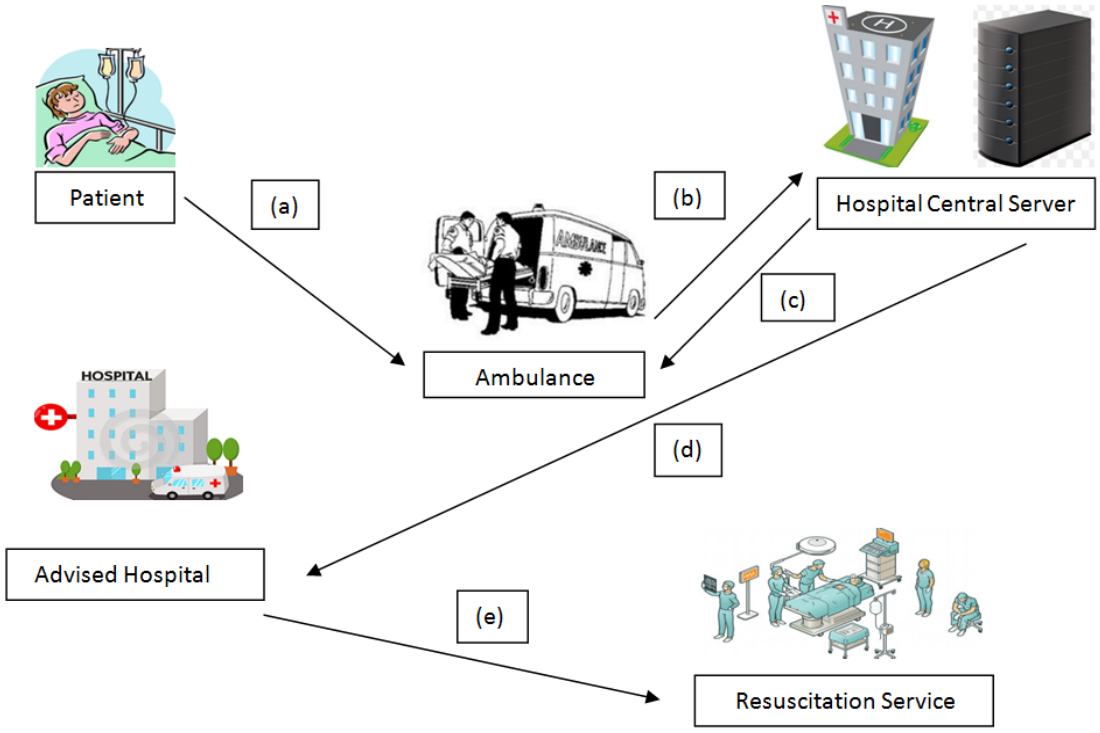

Fig. 2.3 – Architecture proposée pour le deuxième modèle de service

Dans le cas où le patient n'a pas d'antécédents médicaux, le diagramme de transmission des données devient comme le montre la figure 2.4.

- (1) Une fois que le patient est dans l'ambulance, l'infirmière prend le nom et l'état actuel du patient (l'état actuel désigne les paramètres de soins de santé).
- (2) Ces données, le nom du patient et son état actuel, prises par l'infirmière sont enregistrées dans l'agent local de l'ambulance.
- (3) Ces données sont récupérées par un agent léger.
- (4) L'agent léger se déplace vers le serveur central des hôpitaux pour rechercher des informations sur ce patient.
- (5) L'agent léger recherche dans la base de données de l'hôpital central et ne trouve pas d'information sur le patient.

Dans l'étape qui suit, l'agent léger va se cloner pour pouvoir en parallèle exécuter les tâches suivantes :

- (6) L'agent léger renvoie la réponse à l'ambulance : le patient n'est pas rattaché à un hôpital et, selon son cas, il est informé de l'hôpital le plus proche et le plus adéquat.
- (6') L'agent léger cloné migre vers l'hôpital choisi afin d'informer l'équipe médicale de l'arrivée d'un patient et leur transmettre les informations sur ce dernier.

- (7) Ensuite, ces informations sont enregistrées dans l'agent local.

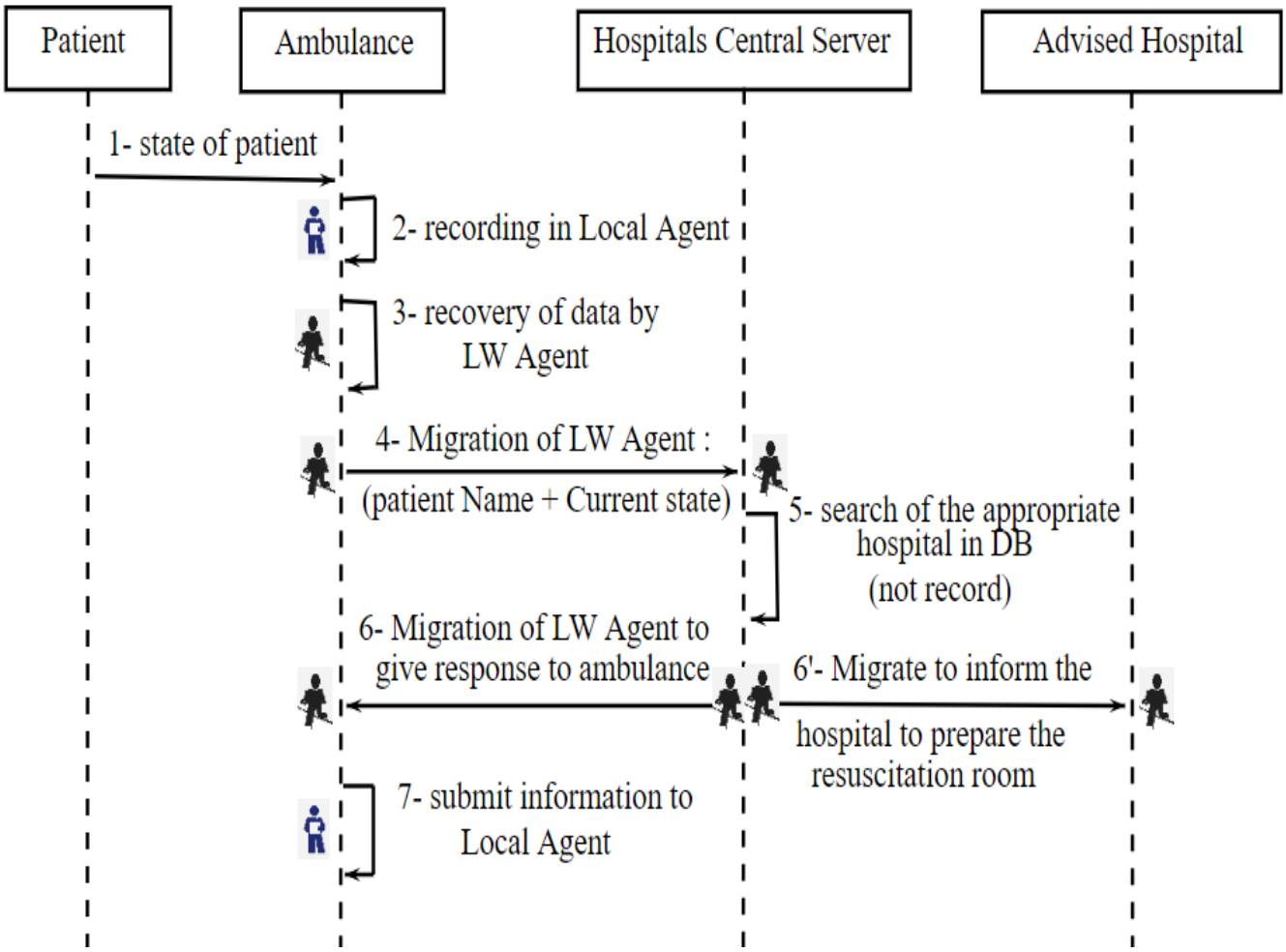

Fig. 2.4 – Diagramme de transmission des données à base d'agents mobiles pour le 2ème cas

2.4 Implémentation de notre proposition sur JADE

Dans cette partie, nous présentons la mise en œuvre de notre approche proposée afin de montrer le rôle et le fonctionnement de l'agent mobile en plus de la faisabilité et la validité de notre modèle. Mais tout d'abord, nous allons commencer par une brève présentation des plateformes existantes d'exploitation des agents mobiles.

2.4.1 Plateformes d'exploitation des agents mobiles

Depuis l'année 1990, plusieurs plateformes ont été déployées pour mettre en place des architectures et des applications basées sur les agents mobiles. Certaines plateformes se sont dissipées, d'autres se sont étendues et devenues les plus appliquées dans divers travaux en plus de leur exploitation dans plusieurs applications à base d'agents mobiles. Dans cette partie, nous allons présenter les plateformes les plus utilisées dans le domaine d'agent mobile et faire une comparaison entre elles selon plusieurs critères (le langage de programmation, le système d'exploitation, le type et protocole de communication, etc).

a - Plateforme Voyager :

Voyager est une plateforme Java à base d'un agent Java ORB (Object Request Broker) [14, 59] développée par la société ObjectSpace en 1997. En combinant les techniques traditionnelles et l'informatique distribuée à base d'agents mobiles, elle fournit au programmeur une flexibilité pour la création des systèmes et applications réparties. Voyager facilite la gestion des communications à distance des protocoles traditionnels CORBA et RMI (Remote Method Invocation). Cette plateforme adopte la syntaxe régulière de Java pour la création des objets distants et de les déplacer d'une application à une autre, localise les agents de manière transparente et leur envoie des messages même en cours de déplacement. Elle procure aussi des fonctionnalités pour générer le code et l'agent mobile. Voyager prend en charge l'architecture traditionnelle Client / Serveur et l'architecture basée sur les agents mobiles.

Inconvénient : la plateforme Voyager n'est pas disponible en version gratuite, c'est un produit commercial. La traçabilité des agents grâce au mécanisme de transmission utilisé est inefficace puisqu'il doit parcourir toute la chaine pour localiser l'agent, et si un seul lien est suspendu, l'agent devient inaccessible.

b - Plateforme Concordia :

Concordia [14, 59] est une infrastructure développée au sein de l'institut de la technologie et de l'information américain (MEITCA) de Mitsubishi, ayant pour but de gérer et de développer des applications à base d'agents mobiles capables de se déplacer à travers le réseau pour atteindre les informations voulues n'importe quand et n'importe où. Le mécanisme de mobilité de l'agent sur cette plateforme prévoit la transmission des informations sur l'état d'exécution de l'agent, en déterminant son emplacement lors de sa migration, les missions accomplies, sa prochaine destination et ce qu'il doit encore effectuer. Côté sécurité, Concordia assure la protection des agents contre la falsification et les ressources contre les accès non autorisés. Elle utilise aussi le chiffrement lors de la migration de l'agent.

Inconvénient : Le problème qui se pose pour un système composé de plusieurs agents réside dans le temps nécessaire pour le stockage persistant des informations de tous les agents utilisés. En plus, si plusieurs agents sont créés puis supprimés, la synchronisation du service de noms fourni par le composant Service Naming entre tous les serveurs Concordia s'avèrent difficile.

c - Plateforme TACOMA :

TACOMA [59, 60] est une plateforme qui supporte les agents programmés en C, C++, Python, Perl, ML, Visual Basic. Elle est aussi adaptée aux nouveaux systèmes d'exploitation Windows 95, NT, CE depuis 2009. Dans TACOMA, un agent est un fragment de code pouvant être installé et exécuté sur un ordinateur distant. Elle prend en charge la manière dont les agents peuvent résoudre les problèmes traités par le modèle Client / Serveur.

Inconvénient : le seul moyen de déplacer l'état d'un agent d'un processeur à un autre consiste à stocker explicitement cet état dans un ou plusieurs dossiers. Les programmeurs de TACOMA doivent savoir quel état capturer et déplacer ; ils doivent programmer ces actions explicitement.

d - Plateforme Aglet

Aglet [59, 61] est une plateforme d'agent mobile développée par IBM en 1997 en langage Java au sein d'un laboratoire. Elle a été maintenu par la communauté open source depuis 2001. Elle se réfère aussi aux spécifications de la norme MASIF. Aglet adopte pour ces agents des proxies pour envoyer des messages et communiquer avec des agents distants, elle est composée de thread unique pour les agents. Les deux types de messages synchrone et asynchrone sont pris en charge. La plateforme Aglet est simple à manipuler et dispose de plusieurs fonctionnalités de sécurité à base de Java API et aussi d'un contrôleur que les développeurs peuvent personnaliser par leurs propres méthodes afin d'assurer la confidentialité des données.

Inconvénient majeur de la plateforme Aglet demeure dans le fait que les proxies fournis ne sont pas dynamiques, ils ne peuvent pas être utilisés après le déplacement d'agent. L'utilisateur est donc dans l'obligation de se procurer lui-même un proxy mis à jour avant de l'appliquer. Aussi, le problème réside dans le fait d'attribuer à chaque agent un thread, l'exécution de longue tâche peut poser problème. Aglet ne prend pas en charge les appels distants aux agents.

e - Plateforme JADE :

JADE [45, 62] (Java Agent DEvelopment framework) est une plateforme logicielle open source écrite en langage Java, développée par le groupe Telecom Italia en collaboration avec l'université de Parme Italie et distribuée par Telecom Italia Lab (Tilab) pour faciliter la mise en place et le déploiement des systèmes multi-agents et des applications conformes aux spécifications FIPA. JADE assure l'interopérabilité, même si les machines ne disposent pas du même système d'exploitation. Cette plateforme propose aux programmeurs d'applications des fonctionnalités prêtes à être utilisées, des interfaces abstraites pour les tâches personnelles dépendantes de l'application. En plus, elle est orientée vers la programmation objet dans des environnements hétérogènes distribués.

JADE comprend les éléments suivants :

- Environnement runtime : c'est l'endroit où les agents JADE demeurent. Pour pouvoir lancé des agents, l'environnement runtime doit être lancé, sinon il ne peuvent pas être exécutés.
- Bibliothèques de classes : bibliothèques prêtes à être utiliser, peuvent être adoptée par les programmeurs directement ou en les personnalisant pour développer les agents.
- Outils graphiques : utilisés pour faciliter la gestion et la surveillance des plateformes d'agents, ainsi que les activités en cours d'exécution des agents.

La plateforme JADE est la plus utilisée par les chercheurs et développeurs car elle offre un système entièrement distribué et hétérogène, une conformité absolue aux spécifications FIPA, un transport efficace des messages asynchrones via une API transparente pour la localisation, une implémentation de pages jaunes, une gestion simple et efficace du cycle de vie des agents, une prise en charge de la mobilité des agents, un ensemble d'outils graphiques pour la surveillance, une prise en charge des ontologies et des langages de contenu.

Plateformes et Caractéristiques	Langage de programmation	Système d'exploitation	Disponibilité de téléchargement	Type de communication	Protocole de communication	Composants	Interface graphique	Mécanisme de sécurité
VOYAGER	Java, C, C++, Net	Unix, Windows	payant (version d'essai)	synchrone, asynchrone	TCP / IP	Serveurs, Agents	Non	Oui (de base)
CONCORDIA	Java	tous types de systèmes	Open source	synchrone, asynchrone	TCP / IP	Machine virtuelle, Serveurs, Agents	quelques uns	Oui
TACOMA	C, C++, Python, Perl, ML, Scheme, Visual basic	Unix et nouveaux systèmes Windows	Open source	synchrone, asynchrone	HTTP, RPC	Agents, Dossiers, Port document, Armoire	quelques uns	Oui
AGLET	Java	tous types de systèmes avec installation de JRE	Open source d'IBM	synchrone, asynchrone	ATP, RMI, CORBA	Contexte, Agents de base (aglets), Tahiti	quelques uns	Oui (de base)
JADE	Java	tous types de systèmes avec installation de JRE	Open source	synchrone, asynchrone	HTTP, IIOP, HTTPS, RMI	Conteneur, Conteneur principal, Plateforme, Agents AMS/ MTS/ DF	Oui	Oui

Tab. 2.1 – Tableau comparatif des plateformes dédiées aux systèmes multi-agents

Dans nos travaux de recherche, nous avons utilisé la plateforme JADE pour l'implémentation des modèles proposés car elle est la plus répandue dans le domaine des systèmes distribués à base d'agents mobiles, en plus, de ses nombreuses caractéristiques intéressantes qui simplifient la communication entre agents et la rendent plus flexible.

2.4.2 Déploiement d'agents mobiles sur JADE

Nous avons développé notre application avec la plateforme JADE (Java Agent Development Framework) comme nous l'avons vu précédemment. JADE est un middleware permettant aux développeurs de construire des systèmes multi-agents. JADE utilise le langage Java et propose de nombreux packages Java, ainsi que des applications de programmeurs à la fois des fonctionnalités prêtes à l'emploi et des interfaces abstraites pour tâches personnelles en fonction de l'application [45, 62].

L'utilisation de Java dans la plateforme JADE est un atout, puisque Java fournit :

- la possibilité d'exécuter plusieurs processus légers (threads) dans un environnement Java qui s'exécute dans un processus Unix standard. Cela permet de gérer l'exécution de plusieurs agents dans le même environnement.
- la possibilité de charger des classes dynamiques qui peuvent être locales comme distantes. Donc un serveur qui accueille un agent peut charger dynamiquement les classes qui composent le programme de l'agent.
- le mécanisme de sérialisation fournit par Java permet le transfert des instances d'un environnement à un autre.

Un agent de la plateforme JADE est conforme à la norme FIPA. Il a un cycle de vie, il a un ou plusieurs comportements (de type behaviours) qui désignent les opérations à exécuter, communique avec les messages de type ACL (Agent Communication Language) et fournit des services. L'agent est globalement identifié par un AID (nom unique).

La plateforme d'agent JADE est composée de conteneurs d'agents pouvant être répartis sur le réseau. Les agents mobiles vivent dans des conteneurs, processus Java fournissant tous les services nécessaires à l'hébergement et à l'exécution des agents.

L'élément primordial d'une plateforme JADE est le conteneur "Container" où les agents sont hébergés. C'est un processus Java produisant l'exécution JADE ainsi que les services fondamentaux à l'hébergement et l'exécution des agents. L'ensemble de ces conteneurs en mode actif constituent une plateforme. Un agent est composé de différents et plusieurs comportements qui sont ajoutés de manière dynamique ; chaque comportement définit la tâche qu'un agent doit achever et il est implémenté dans la méthode "behaviour object".

Lors de la création d'une plateforme à base d'agents mobiles sur JADE, on trouve un conteneur principal "MainContainer" déjà créé sur la plateforme et qui doit être lancé en premier, après les autres conteneurs s'enregistrent à fur et à mesure et se connectent au conteneur principal pour exécuter les agents JADE. Ce conteneur principal contient trois agents qui sont obligatoires comme le montre la figure 2.5.

- l'agent AMS (Agent Management System) : est un agent qui exerce un contrôle de supervision sur l'accès et l'utilisation de la plateforme d'agents. Chaque agent doit s'inscrire

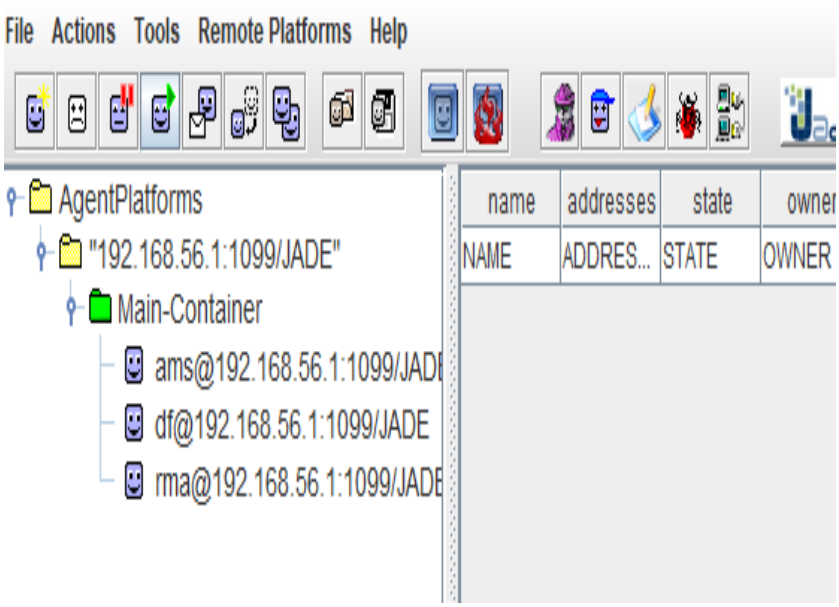

Fig. 2.5 – Conteneur principal de la plateforme JADE

auprès de l'agent AMS afin obtenir un identifiant unique nommé AID (Agent IDentifier) valide. Il est activé par défaut une fois que le conteneur principal est démarré.

- l'agent RMA (Remote Management Agent) : est un agent implémentant la console de gestion JADE elle même. Toutes les actions et opérations effectuées sur l'interface graphique sont achevées par l'agent RMA.
- l'agent DF (Directory Facilitator) : consacré aux pages jaunes, les agents souhaitants annoncer leur service s'enregistrent auprès du DF. Les agents en visite peuvent alors demander au DF de rechercher des agents qui fournissent les services qu'ils désirent. Il est activé par défaut une fois que le conteneur principal est démarré.

2.4.3 Description de l'implémentation de notre modèle

Les principaux composants de l'architecture proposée sont détaillés ci-dessous :

Le nœud d'ambulance :

C'est le nœud le plus important de notre architecture, sous la forme d'un terminal, qui fournit une interface graphique pour interagir avec l'application. Ici, l'infirmière aura comme rôle de déterminer les paramètres suivants :

- La saisie du nom du patient et son statut actuel.
- Lancement de l'agent mobile pour apporter les informations nécessaires.
- Attendre en retour le nom de l'hôpital du patient, en plus des premiers soins à administrer selon le cas du patient.

Le nœud de l'hôpital central :

- Une fois les données récupérées, l'agent mobile léger passera de l'ambulance au serveur central des hôpitaux (abréviation HCS) afin d'effectuer une recherche dans la base de données de l'hôpital.
- Une fois le nom du patient trouvé, cette réponse sera envoyée à l'ambulance afin qu'elle emprunte le chemin qui mène à l'hôpital approprié et simultanément à l'hôpital concerné.

Le nœud de l'hôpital approprié au patient :

- Une fois le message reçu, l'agent lourd va rechercher dans la base de données du patient ses dossiers médicaux.
- Ensuite, préparer l'arrivée du patient et envoyer les premiers secours à l'ambulance.

La figure 2.6 illustre la plateforme de l'application avec les différents nœuds définis sur JADE en tant que conteneurs. Chaque conteneur contient les agents faisant partie de ce nœud.

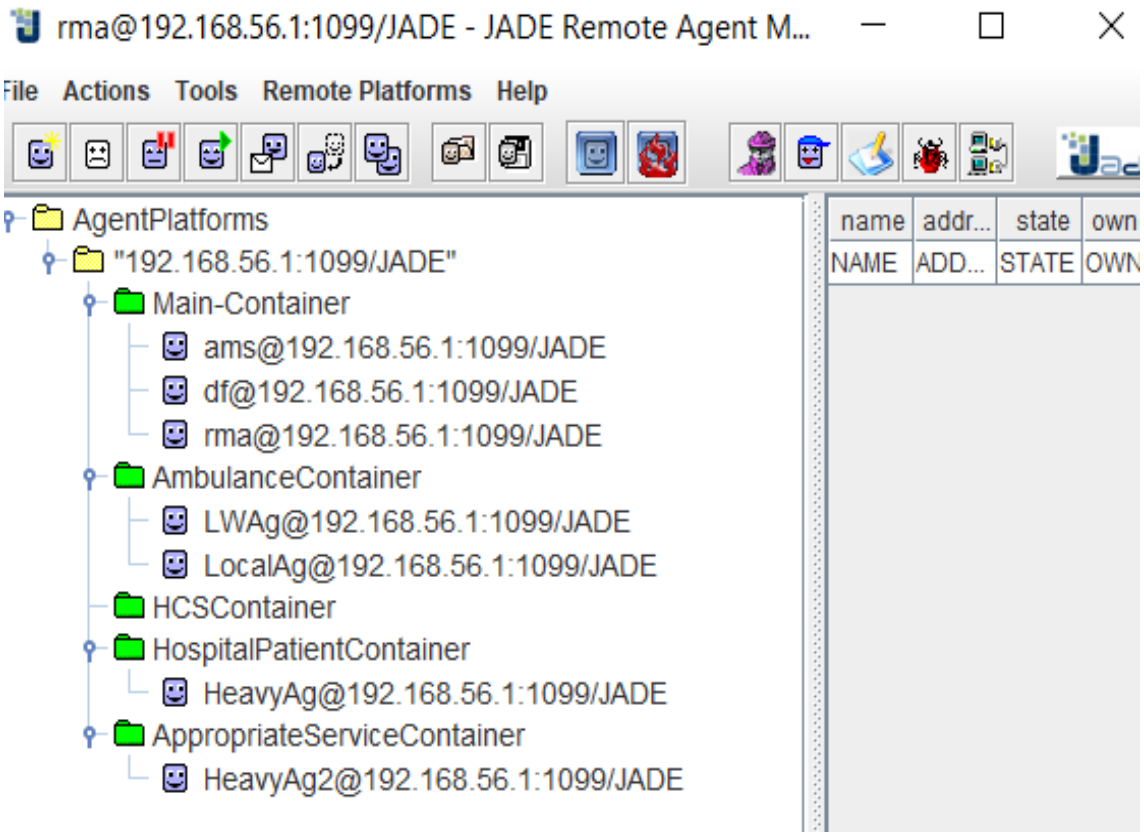

Fig. 2.6 – Plateforme de notre application

Lorsque les données sont saisies par l'infirmière sous forme de message ACL comme le montre la figure 2.7, ces données sont stockées dans l'agent local comme nous le voyons sur la figure 2.8 et sont ensuite récupérées par l'agent léger.

La migration de l'agent léger du nœud de l'ambulance vers le nœud de l'hôpital central HCS est effectuée une fois les données extraites de l'agent local, comme illustré à la figure 2.9.

Une fois que l'agent léger s'est déplacé dans le serveur de l'hôpital central HCS, il cherche dans la base de données le nom de l'hôpital approprié au patient selon les informations obtenues. Une fois trouvé, il envoie les informations à l'agent local de l'hôpital et, en même temps, à l'agent local de l'ambulance. La figure 2.10 montre le message reçu par ce dernier.

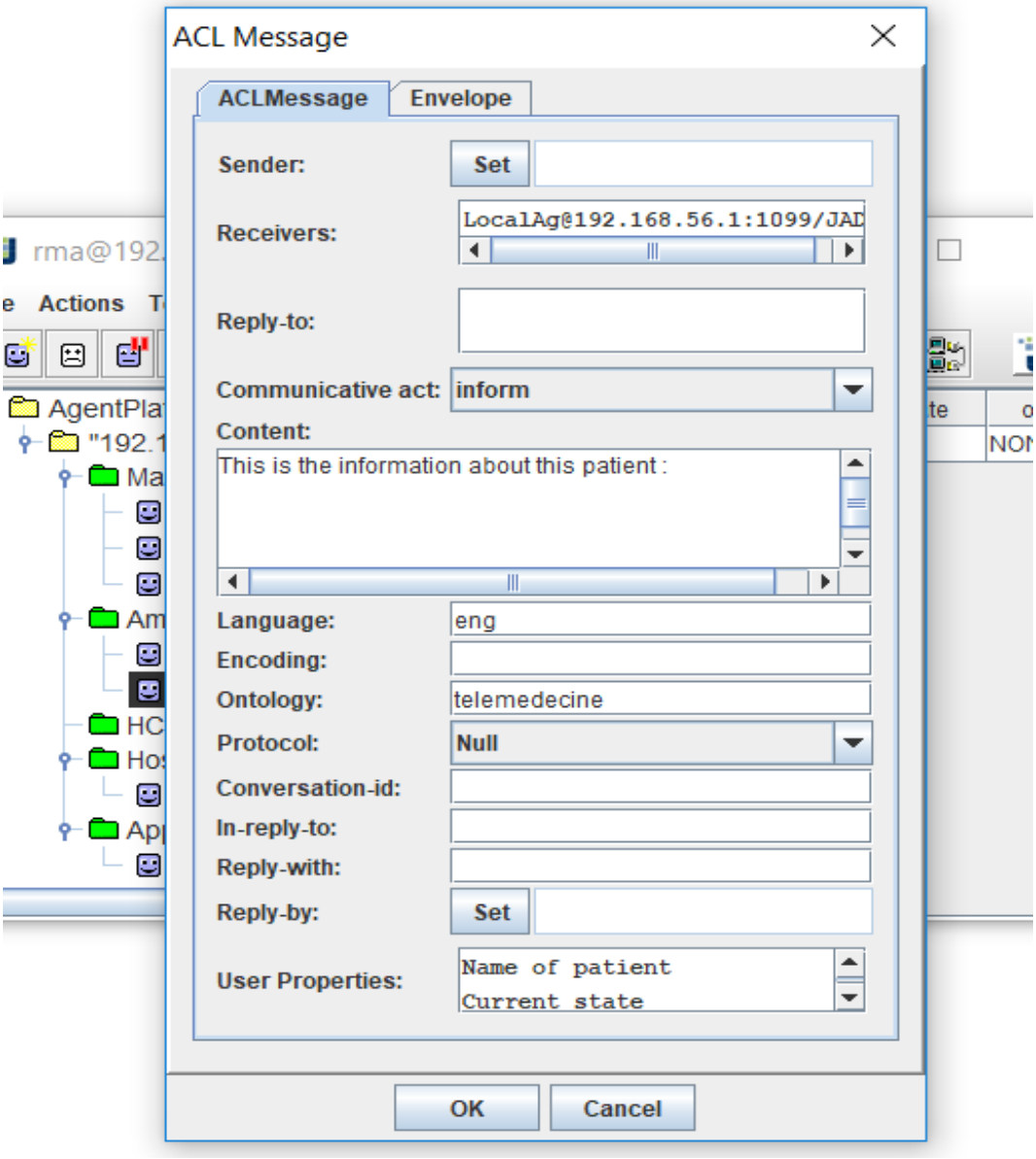

Fig. 2.7 – Données saisies par l'infirmière

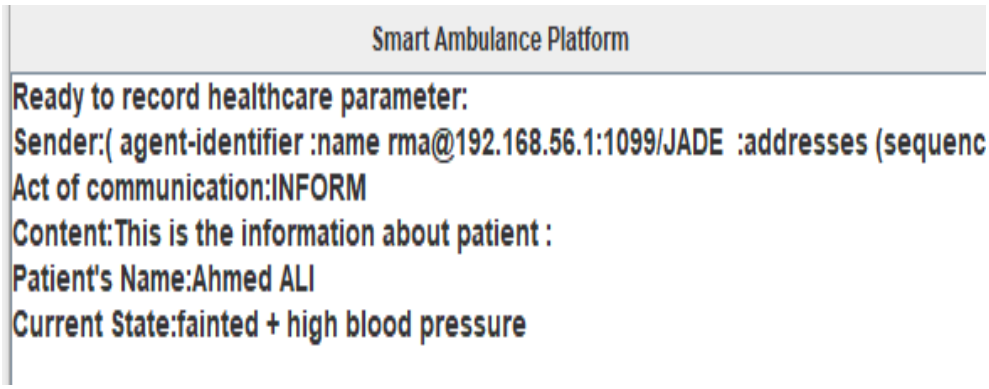

Fig. 2.8 – Enregistrement des données dans l'agent local

La figure 2.11 illustre la communication entre des agents situés à différents endroits, afin de montrer que l'interaction et la coopération entre agents ont bien abouti.

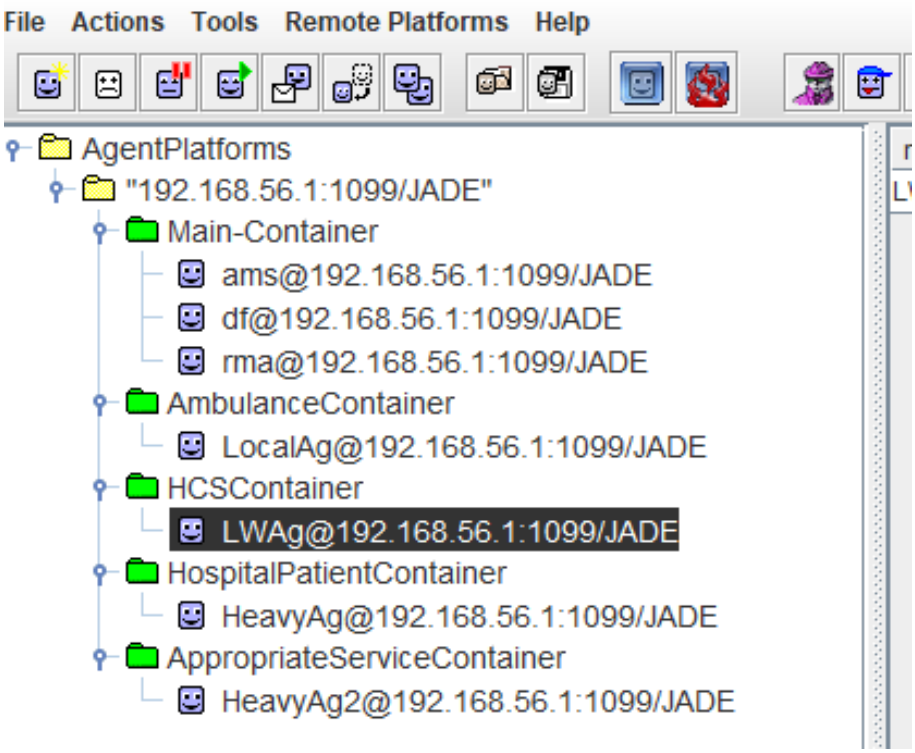

Fig. 2.9 – Migration de l'agent léger

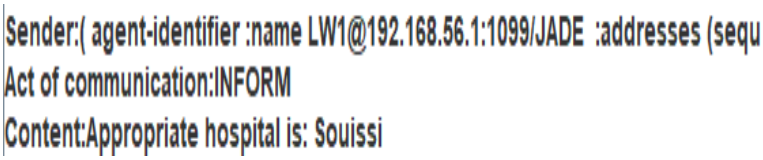

Fig. 2.10 – Données reçues par l'agent local de l'ambulance

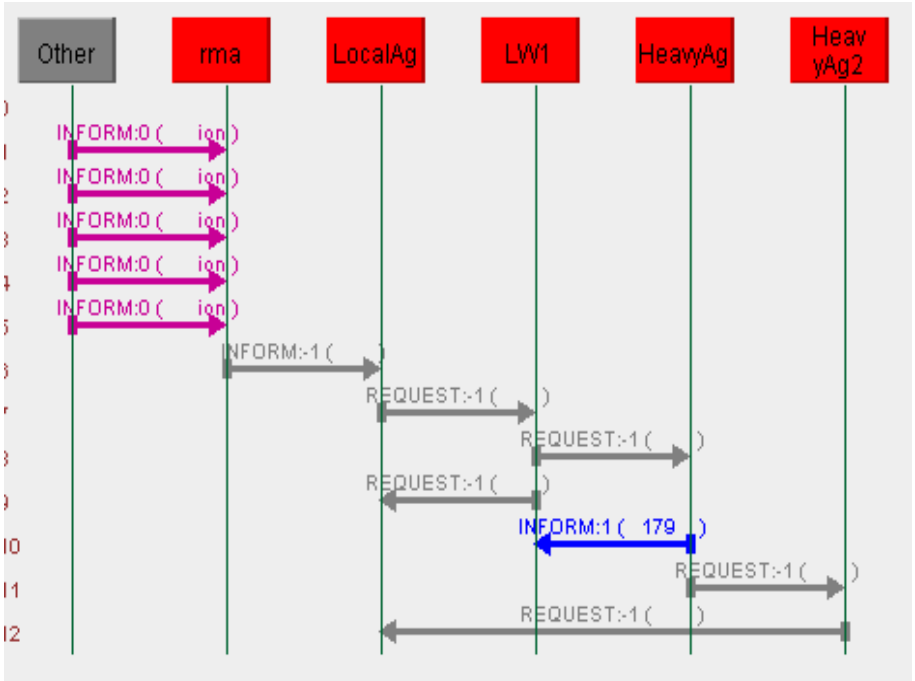

Fig. 2.11 – Le flux de trafic entre agents

2.5 Conclusion

Dans ce chapitre, nous avons intégré la technologie des agents mobiles dans l'ambulance intelligente, puisque le paradigme d'agents mobiles est très utile dans les environnements de ce type et répond aux exigences des applications distribuées. Son principal avantage est sa capacité à permettre aux programmes de passer d'un hôte à l'autre afin d'accomplir une tâche donnée ou de ne transférer que les données utiles.

Notre travail se focalise sur l'ambulance du futur et en particulier sur sa capacité à diagnostiquer et à communiquer l'état du patient à l'hôpital pour recevoir rapidement les recommandations du médecin. Une fois que l'état du patient est enregistré auprès de l'agent local, l'agent léger récupère ses données et migre vers l'hôpital central pour rechercher l'hôpital approprié. Deux cas ont été étudiés : le cas où le patient est rattaché à un hôpital et le second quand il ne l'est pas. Lors de l'obtention des informations nécessaires, l'agent LW est cloné. Le premier continue sa migration vers l'hôpital et le clone revient avec l'adresse de l'hôpital approprié à l'agent local qui se trouve dans l'ambulance intelligente.

Nous avons implémenté nos modèles dans la plateforme JADE pour tester la migration de l'agent léger d'une plateforme à une autre et mettre en œuvre ses communications avec les agents locaux. Grâce à l'utilisation des messages ACL et du flux de trafic, nous avons pu donner une idée précise du fonctionnement de l'ensemble du système d'agents mobiles utilisé par nos modèles proposés.

Dans le chapitre suivant, nous nous sommes intéressés davantage à l'aspect sécurité du modèle à base d'agent mobile.

Chapitre 3

Proposition d'un modèle de sécurité d'un système à agents mobiles

Les progrès achevés dans le domaine des systèmes informatiques et d'Internet évoluent considérablement au fil des années et la sécurité des données manipulées est de plus en plus au centre des préoccupations. Ainsi, avec l'émergence, de nos jours, des réseaux distribués, hétérogènes et de grande taille, les chercheurs et les experts en informatique se précipitent à la recherche de solutions afin d'assurer la sécurité des systèmes informatiques. La technologie d'agents mobiles se voit prometteuse dans le domaine des systèmes distribués, puisqu'elle offre plusieurs avantages et une flexibilité accrue dans le développement des applications. Il existe plusieurs raisons qui font que l'utilisation des agents mobiles est la plus adaptée pour ces types de systèmes. En effet, ils réduisent la charge du réseau, ils surmontent la latence du réseau, ils encapsulent les protocoles, ils s'exécutent de manière asynchrone et autonome, ils s'adaptent dynamiquement, ils sont naturellement hétérogènes et robustes, en plus, ils tolèrent les pannes [63].

Malgré ces nombreux avantages pratiques qui font de la technologie d'agents mobiles un moyen efficace pour réduire les problèmes répartis complexes, la gestion de la sécurité des systèmes d'agents mobiles reste un défi majeur qui dépasse la capacité des méthodes de sécurité traditionnelles. Pour cela, il est primordial de mettre en place des techniques de sécurité efficaces et d'adopter des approches qui ont pour but de créer et déployer un système d'agents mobiles flexible et robuste contre les attaques visant à nuire au système.

Nous nous focalisons dans cette partie sur l'aspect sécurité des systèmes à base d'agents mobiles, en présentons les objectifs de la sécurité et les différentes problématiques qui visent la sécurité des agents mobiles. Puis les travaux connexes sur cet axe seront présentés. Ensuite nous proposons notre approche bidimensionnelle constituée de deux mécanismes de sécurité [4].

3.1 Introduction

L'évolution des systèmes informatiques vers les systèmes distribués a fait en sorte qu'une machine peut coopérer et interagir avec d'autres machines à travers le réseau avec une connexion temporaire et non pas une connexion permanente. La technologie d'agents mobiles représente une grande vague d'innovation et de développement dans les systèmes informatiques distribués.

La flexibilité des agents mobiles, leur autonomie, leur mobilité, leur adaptabilité sur le réseau les rendent plus puissants dans la résolution des problèmes complexes. Cependant, la mobilité et l'autonomie des agents posent des problèmes de sécurité dans les environnements distribués. Comme on l'a expliqué précédemment, un agent mobile est un code logiciel qui se déplace d'une plateforme à une autre. Lorsqu'un client confie une mission à un agent, ce dernier se déplace dans le réseau afin d'accéder aux services localement et achève sa tâche pour le compte de son client.

Lorsqu'un agent se déplace, il est essentiel de s'assurer qu'il sera exécuté correctement en toute sécurité sur le nouveau système visité. De même, il est essentiel de rassurer le système récepteur sur le fait qu'il n'y aura aucun risque à accueillir et héberger un nouvel agent. Un agent mobile peut être la cible de plusieurs types d'attaques car il migre d'une machine à une autre à travers le réseau, cette machine peut être malveillante.

Les agents mobiles peuvent être des outils puissants, mais ils peuvent aussi devenir destructeurs si les précautions de sécurité adéquates ne sont pas mises en place. Nous sommes donc face à un dilemme : Comment pouvons-nous fournir simultanément des mécanismes de sécurité efficaces pour les agents mobiles et les hôtes fréquentés par ces agents mobiles. Un agent mobile peut être exécuté dans un environnement qui peut ne pas être contrôlé par l'expéditeur de l'agent, donc le serveur ne peut pas garantir la confidentialité et l'intégrité de l'agent. Un agent qui migre peut devenir agent malveillant, donc le serveur a besoin d'un type de mécanisme d'évaluation de l'état afin de vérifier l'exactitude de toutes les transitions d'états de l'agent, ou il doit former des relations de confiance (en utilisant la signature numérique) avec d'autres serveurs.

Ainsi, afin d'assurer la sécurité d'un système à base d'agents mobiles, on trouve dans la littérature deux tendances de recherche : la proposition de divers techniques ayant pour but de contrecarrer les attaques, la spécification d'un ensemble de règlements déterminant la manière dont les informations et les ressources au sein d'un système informatique doivent être protégées. L'ensemble de ces règlements forme la politique de sécurité d'un système informatique. Au niveau des systèmes informatiques basés sur les agents mobiles, il est nécessaire de déployer une structure de sécurité flexible, capable de s'adapter aux changements dynamiques selon les besoins de la sécurité des agents mobiles et aussi selon leurs systèmes d'exécution.

Dans ce chapitre, nous nous concentrons spécialement sur la sécurité des agents mobiles. Nous présentons les objectifs de sécurité des systèmes informatiques et spécialement des systèmes à base d'agents mobiles. Nous abordons les menaces et les exigences en plus des problèmes liés à la sécurité des agents mobiles. Ensuite, nous présentons les différents types de mécanismes de sécurité utilisés dans les travaux connexes. Puis, nous proposons notre travail réalisé pour la sécurité et la protection des systèmes informatiques à base d'agents mobiles qui repose sur une approche en matière de confidentialité des données, de protection de l'intégrité du système, de non-répudiation et d'authentification de l'origine. Nous nous intéressons également à la sécurité des agents mobiles contre les hôtes malveillants et contre les attaques DOS qui visent la disponibilité du système.

3.2 Objectifs de la sécurité des systèmes informatiques

La sécurité des systèmes informatiques est primordiale pour faire face aux attaques et aux menaces qui les ciblent. La sécurité a comme objectifs d'éviter la modification en plus de la divulgation non-autorisée des données, et de contrecarrer l'exploitation non-autorisée de ressources du système à protéger.

3.2.1 Les principaux services de sécurité

La Norme internationale ISO (International Organization for Standardization) 7498-2 donne une description générale des éléments généraux d'architecture pour l'aspect de la sécurité afin de réaliser une communication sûre entre les systèmes[64], elle définit quatorze services de sécurité qui sont regroupés en 5 parties principales : [65, 66, 67, 68]

- **Authentification** : assurer l'authentification dans un système consiste à prouver que l'identité des différentes entités doit pouvoir être vérifiées, et que les informations reçues sont conformes à celles fournies. Il faut assurer aussi que les deux entités communicantes sont bien ce qu'elles affirment être. De plus, le service d'authentification doit montrer que la connexion ne peut être brouillée par une troisième entité essayant de se faire passer pour un des deux correspondants.
- **Contrôle d'accès ou autorisation** : est un composant essentiel de la sécurité des données. Il permet de contrôler l'accès aux ressources en déterminant qui peut accéder aux informations, puisque l'accès à certaines ressources doit être restreint à des entités autorisées ; autrement dit : qui ou quoi peut afficher ou utiliser des ressources dans un système. L'objectif principal du contrôle d'accès est de fournir des mécanismes de sécurité afin de minimiser le risque d'accès non autorisé dans un environnement informatique, les politiques de contrôle d'accès assurent la protection des informations confidentielles. Les objectifs du service de contrôle d'accès rejoignent ceux de la disponibilité.
- **Confidentialité** : assurer la confidentialité des données contre toute divulgation non autorisée, consiste à faire en sorte que les informations restent secrètes et que seules les personnes autorisées y aient accès. Son objectif est de protéger les données sensibles contre tout accès non autorisé ; en d'autres termes, empêcher que les informations secrètes ou sensibles ne parviennent à la mauvaise personne, tout en garantissant que seules les personnes autorisées peuvent l'obtenir.
- **Intégrité** : permet de contrecarrer les menaces actives et d'éviter l'altération, la corruption et la destruction des données dans le réseau de manière non autorisée, Elle reste un domaine très large couvrant à la fois les modifications, les moyens de modifications mais également la cohérence des données après la modification. Les données ne doivent pas être modifiées en transit et des mesures doivent être prises afin de garantir que les données ne peuvent pas être modifiées par des personnes non autorisées (voire dans le cas de la violation de la confidentialité). Les mécanismes d'intégrité gardent les données fiables en protégeant les données du système contre les changements intentionnels ou accidentels, Ils ont 3 objectifs : empêcher les utilisateurs non autorisés d'apporter des modifications aux données ou aux programmes, empêcher les utilisateurs autorisés de faire des modifications inappropriées ou non autorisées, maintenir la cohérence interne et externe des données et des programmes.

- **Non-répudiation** : assurer que lorsqu'une information est transmise entre deux entités, l'émetteur ne doit pas pouvoir nier avoir envoyé l'information et le destinataire l'avoir reçue. Chaque partie possède ainsi la preuve de l'existence de la transaction, donc ni l'un ni l'autre ne peut nier ultérieurement avoir traité les données. Ce service prend l'une des formes suivantes ou les deux : la non-répudiation avec preuve de l'origine qui assure la protection contre toute tentative de l'expéditeur de nier et le fait qu'il a envoyé des données ou leur contenu. La non-répudiation avec preuve de la remise qui assure contre toute tentative ultérieure du destinataire de nier le fait d'avoir reçu les données ou leur contenu. La non-répudiation garantit que l'expéditeur des données reçoit une preuve de livraison et que le destinataire reçoit une preuve de l'identité de l'expéditeur, de sorte que ni l'un ni l'autre ne peut nier ultérieurement avoir traité les données. De plus, ce concept peut s'appliquer à n'importe quelle activité, pas seulement à l'envoi et à la réception de données ; dans un sens plus général, c'est un mécanisme pour prouver qu'une activité a été réalisée et par qui. La non-répudiation comprend généralement les services d'authentification, d'audit / journalisation et de cryptographie. Une application courante de ce service serait la signature numérique de messages électroniques pour prouver que le message reçu a bien été envoyé par le prétendu expéditeur. La non-répudiation et le contrôle d'accès sont la plupart du temps mis en œuvre ensemble car ils partagent plusieurs composants communs.

3.2.2 Autres services de sécurité

En plus des services de sécurité mentionnés ci-dessus, on trouve les objectifs fondamentaux suivants : [66, 69, 70, 71, 72]

- **Disponibilité** : l'objectif principal de la disponibilité est que le système doit être opérationnel et les informations disponibles au moment et à l'endroit où elles sont nécessaires. La disponibilité est maintenue lorsque les informations sont disponibles au moment où les utilisateurs autorisés doivent y accéder et lorsque tous les composants du système d'information fonctionnent correctement. Si un problème surgit dans le système, il peut rendre l'accès à l'information impossible, ainsi l'information devient indisponible.
- **Confinement** : ce service de sécurité est complémentaire au service de la confidentialité pour le bon usage des informations. Le confinement garantit qu'un sujet n'arrive pas à divulguer volontairement le contenu des objets auxquels il a accès à quelqu'un qui n'a pas le droit d'y accéder.
- **Identification** : ce service intervient avant celui de l'authentification. Il repose sur la manière d'établir l'identité de l'utilisateur. Ce dernier adopte un identifiant qui peut être sous forme d'un login, d'un nom d'utilisateur, etc, qui l'identifie et qui lui est attribué individuellement puisqu'il est unique comme identifiant.
- **Traçabilité** : la traçabilité des données permet de retrouver les opérations effectuées par une entité en conservant les traces de l'information. Le rôle de la traçabilité est primordial vu qu'elle permet d'être sûr que les services de confidentialité, de disponibilité et d'intégrité sont respectés. La traçabilité d'une information définit le fait de savoir d'où elle vient, par où elle est passée et où elle a abouti.
- **Responsabilité** : ce service requiert que les administrateurs et les utilisateurs sont tenus en tant que responsables des comportements qui peuvent causer un problème à la sécurité des informations.

- **Assurance** : approuve que les services de sécurité, à savoir la confidentialité, l'intégrité, la disponibilité et la responsabilité, sont réalisés convenablement. Ceux qui incluent une fonctionnalité qui marche normalement, une protection approuvable à faire contre les erreurs involontaires faites soit par les utilisateurs soit par les logiciels, et une résistance suffisante au contournement intentionnel.

3.3 Problématiques de sécurité des agents mobiles

3.3.1 Menaces sur un système à base d'agents mobiles

Certes, la technologie d'agents mobiles apporte aux applications et aux systèmes distribués une flexibilité et une facilité, mais elle présente également un nombre de défis et de problèmes particulièrement dans le domaine de la sécurité. Un agent mobile peut être la cible de plusieurs types d'attaques car une fois créé, il se déplace d'un environnement d'exécution à un autre à travers le réseau afin d'accomplir sa tâche. La plateforme initiale qui l'a créé est la plus fiable pour un agent. Mais lorsqu'il migre vers un autre environnement d'exécution différent de celui qui l'a créé, ce dernier peut compromettre la sécurité puisqu'il prend le contrôle total sur l'agent (code, données et état d'exécution). On trouve aussi le cas d'un agent malveillant, une fois il est exécuté sur une plateforme il peut nuire à celle-ci. C'est pourquoi, la sécurité des systèmes à base d'agents mobiles doit viser la sécurité entre agents, la sécurité entre un agent et la plateforme d'exécution, la sécurité entre plateformes et la sécurité entre un tiers et une plateforme ou un agent.

Ainsi, on distingue quatre catégories principales [73] de menaces visant la sécurité des agents mobiles qui sont présentées ci-dessous et illustrées sur la figure 3.1.

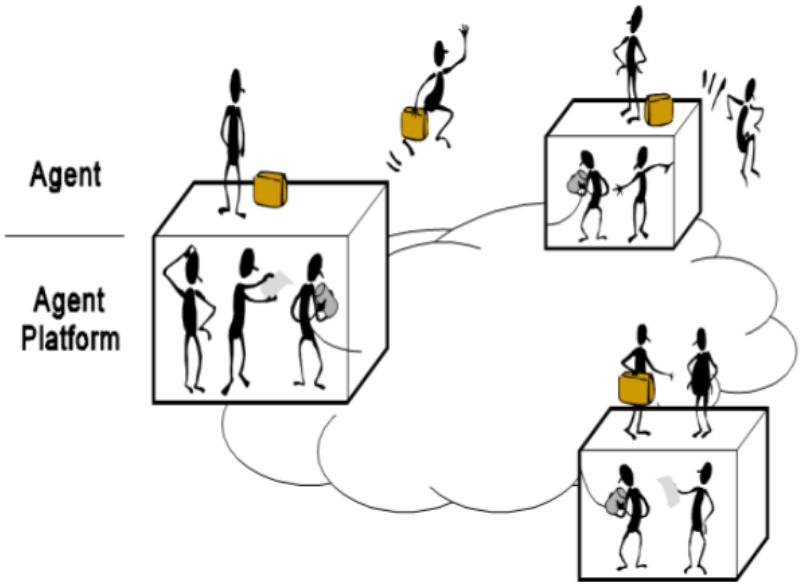

Fig. 3.1 – Les menaces de sécurité d'un système à base d'agents mobiles [73]

- **Agent contre la plateforme d'agents** : cette catégorie englobe toutes les menaces où les agents exploitent les faiblesses de sécurité de la plateforme d'agents ou lancent des attaques contre la plateforme d'agents. Un agent qui s'exécute sur une plateforme a deux formes d'attaques principales. La première est d'obtenir un accès non autorisé à

des informations situées sur la plateforme d'agents. La seconde est d'employer son accès autorisé de manière inattendue afin de perturber le fonctionnement de la plateforme. L'accès non autorisé peut se réaliser simplement suite à l'absence des mécanismes de contrôle d'accès adéquats sur la plateforme ou en se faisant passer pour un agent approuvé par la plateforme. Une fois l'accès à la plateforme est réalisé, les informations peuvent être partagées avec cet agent et divulguées ou modifiées.

- **Plateforme d'agents contre agents** : cette catégorie représente toutes les menaces où les plateformes compromettent la sécurité des agents. Une plateforme d'agents qui va recevoir un agent peut facilement isoler et attaquer ce dernier en extrayant des informations, en modifiant ou corrompant son code ou son état, en refusant les services demandés ou simplement en les réinitialisant ou en les terminant complètement. Une plateforme malveillante peut infecter un agent en répondant de manière inadéquate ou fausse aux demandes d'informations ou de services ou de manière à retarder l'agent afin que sa tâche ne soit plus faisable dans le délai demandé. Cette catégorie comprend les attaques DOS, espionnage a et masquage.
- **Agents contre agents** : cette catégorie englobe l'ensemble des menaces où les agents exploitent les faiblesses de sécurité d'autres agents ou lancent des attaques contre d'autres agents. Un agent peut cibler un autre agent en appliquant plusieurs approches générales. Celles-ci incluent des actions visant à falsifier des transactions, à écouter des conversations ou à interférer avec l'activité d'un agent. Un agent attaquant peut répondre de manière fausse aux demandes directes qu'il reçoit d'une cible ou même nier qu'une transaction légitime a eu lieu.
- **Un tiers contre un agent ou une plateforme d'agents** : cette catégorie expose les menaces dans lesquelles des entités externes, autres agents ou plateformes, menacent la sécurité des agents et des plateformes internes. Même en supposant que ces derniers aient de bons comportements, d'autres entités peuvent tenter des actions afin de perturber ou nuire le cadre fonctionnel des agents.

3.3.2 Exigences de sécurité des agents mobiles

En règle générale, un système à base d'agents mobiles sécurisé doit atteindre et maintenir les objectifs de sécurité suivants : [64, 65, 72]

- **Authentification** : c'est le processus de vérification de l'identifiant d'un utilisateur, d'un périphérique ou d'une entité avant de lui autoriser l'accès aux ressources d'un système afin de l'empêcher de falsifier ou de masquer des informations. Un agent mobile doit s'authentifier auprès de chaque système visité basé sur la technologie d'agents mobiles. Ainsi, un système d'agents est en mesure de décider s'il s'agit d'un agent approuvé ou non. Et vice versa, l'agent mobile doit pouvoir authentifier le système d'agents mobiles accueillant.
- **Contrôle d'accès ou autorisation** : c'est un processus qui définit soit l'acceptation ou le refus d'une demande émanant d'un utilisateur, d'un programme ou d'une entité après confirmation de l'authentification.
- **Confidentialité** : impose que lors des échanges dans un système, les données doivent être protégées contre la divulgation non autorisée et que seules les entités autorisées puissent y accéder. L'agent mobile et le système d'agents doivent protéger leurs informations privées contre les accès non autorisés.

- **Anonymat** : les politiques de sécurité de la plateforme d'agents mobiles et ses exigences en matière d'audit doivent être soigneusement pesées par rapport aux attentes des agents mobiles en matière de confidentialité. Cependant, la plateforme doit garder anonyme l'identité de l'agent par rapport aux autres agents.
- **Disponibilité** : comporte la disponibilité des données et la disponibilité des services d'un agent mobile afin que des utilisateurs légitimes puissent accéder aux données et aux systèmes en temps voulu, sans oublier les attaques DOS qui peuvent nuire à un système et le rendre indisponible. Cette propriété garantit l'accessibilité aux ressources et/ou aux services tant qu'il s'agit d'un agent autorisé.
- **Intégrité** : cette propriété est divisée en deux parties : l'intégrité des données et l'intégrité du système. L'intégrité des données signifie que les données ne doivent pas être altérées ou détruites de manière non autorisée pour pouvoir maintenir la cohérence. L'intégrité du système signifie qu'un système doit être à l'abri des manipulations non autorisées. Dans le contexte des agents mobiles, l'itinéraire de l'agent est une donnée qui nécessite une protection contre toute forme de modification. La plateforme d'agents mobiles doit protéger les agents contre les modifications non autorisées de leur code, de leur statut d'exécution et de leurs données, et aussi garantir que seuls des agents ou des processus autorisés modifient les données partagées.
- **Non-répudiation** : signifie que chaque utilisateur et entité ne doit pas nier la communication effectuée ultérieurement. Pour ce faire, des échanges de communications importants doivent être enregistrés afin d'éviter qu'une partie ne nie avoir traité une transaction.
- **Équité** : cette propriété signifie qu'aucune partie ne peut avoir d'avantage sur les autres. Ainsi, des mécanismes sont nécessaires pour assurer une interaction équitable de la plateforme d'agents mobiles dans l'échange électronique.

Dans notre travail de recherche, nous nous sommes intéressés surtout sur les systèmes à bases d'agents mobiles.

3.3.3 Différents types d'attaques sur les systèmes à base d'agents mobiles

Une attaque informatique est une suite d'actions qui a pour intention de nuire à l'un des services de sécurité exposés dans la partie où nous avons discuté des objectifs de la sécurité des systèmes informatiques. On peut classifier les attaques les plus courantes contre les systèmes à base d'agents mobiles en trois parties :

- des attaques qui visent la disponibilité d'un système,
- des attaques qui visent la confidentialité des données,
- des attaques qui visent l'intégrité du système.

a - Les attaques visant la disponibilité d'un système basé sur les agents mobiles :

Les attaques par déni de service [70] connues sous l'acronyme DoS (Disk Operating System) ciblent à rendre un système indisponible pendant un certain moment en surchargeant les

ressources disponibles afin d'empêcher les utilisateurs de manier et d'exploiter les services d'un système. Ils peuvent provoquer :

- une indisponibilité ou une perte d'un service de façon temporaire.
- une destruction des fichiers d'un système informatique.
- une augmentation du trafic réseau afin de désactiver un nœud du réseau.

Les attaques par déni de service sont répandues entre la plateforme et l'agent et vice versa. Lorsqu'une plateforme reçoit un agent, elle doit exécuter ce dernier et l'aider à accomplir sa tâche tout en lui fournissant les ressources nécessaires et en respectant son fonctionnement. Dans le cas où la plateforme est malveillante, elle peut ignorer les demandes de l'agent et lui causer des retards, refuser ou terminer son exécution sans message de retour. Ici, un agent devient non réactif puisqu'il reçoit des tâches à accomplir mais il est incapable de les résoudre et se bloque. Ce type d'attaques ne visent pas à voler des données mais visent à rendre un système indisponible un certain moment afin de provoquer des pertes couteuses au système cible (matériel et budgétaire).

b - Les attaques visant la confidentialité d'un système basé sur les agents mobiles :

Ce type d'attaques cible à démasquer le contenu des informations sensibles et d'écouter la communication qui se déroule entre les entités afin de tirer profit de cette dernière. On y trouve :

- **attaque par écoute** : est une attaque passive où les communications secrètes d'un agent mobile sont surveillées et interceptées par un tiers afin d'exploiter les informations qui circulent. L'attaque par écoute est appliquée lorsqu'un agent s'exécute sur une plateforme malveillante. Cette dernière peut déduire le code, les données ou le contrôle de flux et les identifiants des agents avec lesquels elle communique à partir des informations données par l'agent lors de son exécution. Cette forme d'attaque est souvent difficile à prévenir et à détecter [70, 74].
- **attaque d'agrégation de données** [70] : elle permet à un attaquant de déduire et de différencier entre les informations classifiées et non classifiées. Dans le cas où un attaquant renifle la clé de chiffrement ou vol un mot de passe, il se procure un accès non autorisé au système ce qui nuit au service de sécurité qui est la confidentialité et commence à suivre les communications qui ne lui sont pas dédiées en tant qu'un utilisateur légitime. L'attaque d'agrégation de données se sert de la technologie de diffusion utilisée dans la plupart des réseaux.
- **attaque par masquerade** : ce type d'attaque est provoqué lorsqu'une plateforme prétend être une autre plateforme que l'agent doit visiter au cours de son parcours. Ici, l'agent s'exécute sur cette plateforme malveillante et dans ce cas, elle peut lire les informations qui ne lui sont pas dédiées et modifier le code de l'agent.

c - Les attaques visant l'intégrité d'un système basé sur les agents mobiles :

Dans ce type d'attaques, l'attaquant tente de modifier les informations de la communication qui se déroule entre les entités ou les données d'un système. Parmi les divers formes d'attaques

on trouve [70] :

- **attaque de l'homme au milieu (MITM)** : où un attaquant est capable de lire et de modifier des messages entre deux entités sans qu'elles ne s'en doutent. L'attaquant renifle les paquets du réseau, les modifie et les réinsère ensuite dans le réseau [75]. De telles attaques se produisent principalement dans les systèmes à clé publique où l'introduction de clés signées par des tiers de confiance peut aider à concevoir un mécanisme pour réaliser de telles attaques.
- **attaque par altération** : un agent mobile est exécuté sur plusieurs plateformes différentes lors de son itinéraire. Un hôte malveillant est capable de modifier le contenu d'un agent et au lieu d'exécuter les tâches qui lui ont été dédiées, il se verra en train d'exécuter des tâches différentes. Cependant, à chaque fois qu'il visite une nouvelle plateforme, si elle est aussi malveillante, de nouvelles modifications vont être faites sur l'agent. L'altération peut être détectée en faisant en sorte que l'auteur original signe le code de l'agent. Cette détection devient dans ce cas difficile pour les agents qui visitent plusieurs plateformes.

Après avoir présenté les différents types d'attaques visant les systèmes à base d'agents mobiles, nous discutons dans la partie qui suit divers approches existantes afin d'assurer la protection des agents mobiles.

3.3.4 Approches existantes pour la protection des agents mobiles

La technologie des agents mobiles présente plusieurs avantages à savoir sa capacité de se déplacer et de migrer d'un environnement à un autre pour se rapprocher des ressources distantes. Un agent mobile a aussi la capacité de se cloner pour pouvoir fonctionner en parallèle sur plusieurs systèmes, mais également de communiquer avec d'autres agents afin de partager leur connaissance et leur expertise. Malgré ses avantages, si l'aspect de sécurité n'est pas pris en compte, cette technologie peut poser problème.

En effet, la plateforme d'agents à l'origine ou à la création d'un agent mobile est nommée la plateforme d'origine, elle est généralement la plus fiable et la plus sécurisée pour un agent. Une fois que l'agent mobile migre vers un autre environnement, ce dernier est nommé un environnement hôte et prend le contrôle total du code et de l'état d'exécution de l'agent mobile. Il est donc difficile d'assurer la protection des agents mobiles contre des hôtes malveillants puisque dans ce cas ils peuvent être exposés à plusieurs menaces de sécurité.

Les agents mobiles sont soumis à de nombreuses menaces lorsqu'ils se déplacent et s'exécutent dans un autre environnement contrôlé par une autre plateforme d'agents mobiles différente de celle qui l'a créée. C'est pourquoi différentes exigences de sécurité doivent être générées dans un système afin d'atteindre les objectifs de sécurité.

Dans cette partie, nous décrivons certains mécanismes de sécurité et solutions proposées dans la littérature afin d'assurer la protection des agents mobiles.

a - Sécurisation des systèmes basés sur des agents mobiles contre des hôtes malveillants

Dans l'article [76], les auteurs discutent d'une façon générale la sécurité des agents mobiles contre des hôtes malveillants. Ce travail fournit une solution contre les attaques DOS lancées par un hôte malveillant qui bloque un agent mobile visiteur et l'empêche de poursuivre son itinéraire.

Cette approche utilise deux agents mobiles : un agent primaire noté PA (Primary Agent) et un agent ombre nommé SA (Shadow Agent). Le mécanisme utilise un accusé de réception et de synchronisation afin de s'assurer qu'un agent mobile a visité un hôte dans son itinéraire et s'est rendu en toute sécurité au suivant. Un hôte est considéré non bloquant s'il permet au PA de poursuivre sa tâche et de partir en toute sécurité vers l'hôte suivant. Le SA soupçonne une action malveillante s'il ne reçoit pas d'accusé de réception dans un délai approprié, après quoi il demande l'aide de l'hôte principal pour identifier l'hôte malveillant et prend des mesures correctives. Lorsque l'hôte local identifie l'hôte malveillant, il envoie une nouvelle instance de l'autorité de sécurité à un hôte sécurisé afin de répondre à une association de sécurité qui contient une copie des données collectées. SA rechargera les données collectées dans le nouveau PA. Ce dernier poursuivra son chemin en ignorant l'hôte malveillant. Dans ce travail, le problème qui se pose est que le SA, une fois migré vers une autre plateforme, peut devenir aussi une cible d'attaque.

b - Utilisation du mécanisme Secure-Image pour protéger les agents mobiles contre des hôtes malveillants (SIM)

Le mécanisme de Secure-Image (SIM) proposé dans l'article [77] vise à assurer la protection des agents mobiles contre les hôtes malveillants, les attaques d'espionnage et d'altération. Le mécanisme SIM fonctionne comme suit : un agent mobile de SIM migre d'un hôte à un autre en suivant un itinéraire afin d'effectuer sa tâche. L'agent mobile est chiffré lorsqu'il passe d'un hôte à un autre pour assurer la protection dans les canaux de communication. L'agent mobile est déchiffré lorsqu'il arrive chez l'hôte. Par conséquent, certaines parties de l'agent mobile peuvent rencontrer des problèmes de sécurité si l'hôte n'est pas approuvé. Le mécanisme SIM intervient afin de protéger toutes les parties de l'agent mobile dans des hôtes non approuvés. SIM génère une image sécurisée pour l'agent mobile avant son arrivée sur des hôtes classés comme hôtes non fiables. Si l'hôte suivant dans l'itinéraire des agents mobiles n'est pas approuvé, l'agent visite le contrôleur SIC (Secure-Image Controller) proche qui génère une image sécurisée de l'agent et l'envoie à l'hôte non approuvé. Cela protège l'agent d'origine des hôtes malveillants en visite.

Le point faible de cette solution est que les hôtes approuvés et non approuvés doivent être connus, ce qui n'est pas toujours le cas dans les systèmes distribués.

c - Sur la sécurité du code mobile

Sander et al dans [78] proposent des fonctionnalités chiffrées à l'aide de la méthode informatique non interactive, avec CEF (Computing with Encrypted Fonctions) comme solution créée afin de répondre aux exigences de sécurité du code mobile. Le but de cette proposition est de chiffrer des fonctions afin que leur transformation puisse à nouveau être implémentée en tant que programmes. Le programme résultant consistera en instructions sous forme de texte clair

qu'un processeur comprendra, mais il ne sera pas en mesure de comprendre la fonction du programme. Bien que certains résultats théoriques liés au CEF aient été produits avec des données chiffrées, ces résultats semblent peu pratiques en ce qui concerne leur faisabilité informatique ainsi que leur interactivité.

d - Sécurité Blackbox limitée dans le temps : protection des agents mobiles contre les hôtes malveillants

Dans l'article [79], Hohl introduit la sécurité BlackBox afin d'assurer la protection du code mobile contre les hôtes malvaillants en générant du code exécutable à partir d'une spécification donnée d'agent. Ce code généré est exécuté par l'hôte en tant que boîte noire, c'est à dire que l'hôte de l'agent ne peut ni le modifier ni le lire, il ne peut que l'exécuter. L'utilisation de Blackbox peut être un risque majeur pour l'hôte qui va l'exécuter, car ce dernier ne dispose pas d'informations suffisantes sur ce qu'il reçoit.

e- Extension de la traçabilité des exécutions pour la sécurité du code mobile

Dans l'article [80], Tan et al ont pour objectif de protéger le code de l'agent mobile contre l'attaque DOS et l'attaque à falsification de l'état causés par des hôtes malveillants. Ils ont proposé une approche basée sur l'extension du mécanisme de trace cryptographique. Cette approche implique un serveur nommé serveur de vérification tiers, de confiance, qui entreprend le processus de vérification des traces pour le compte du propriétaire de l'agent. Lorsqu'un propriétaire d'agent lance un agent mobile sur une plateforme hôte, il crée une copie du code et de l'état de l'agent et les transmet à un serveur de vérification désigné par la plateforme hôte.

Pendant que l'hôte exécute l'agent, il crée simultanément une trace de cette exécution. À la demande de la migration, l'hôte transmet cette trace et l'état final de l'agent au serveur de vérification désigné, ce qui garantit la validité de la séquence d'exécution. Une fois que ce serveur reçoit une copie de l'agent, il sera informé de l'identité de la plateforme exécutant l'agent réel.

Il peut ainsi mettre en œuvre un mécanisme permettant de s'assurer qu'une trace de l'exécution parvient de l'hôte requis dans un délai raisonnable. Cela fournit un moyen de se protéger contre certaines formes d'attaques DOS. La préoccupation des auteurs vis-à-vis de cette approche est de s'assurer que les traces, le code et le statut de l'agent se propagent de manière sécurisée dans leur système, et que les traces sont correctement associées aux agents correspondants. Parmi les inconvénients de ce travail, il y a le coût élevé de la cryptographie, car les chercheurs doivent essayer de trouver des moyens de réduire le coût de la cryptographie du protocole utilisé sans compromettre les propriétés de sécurité.

f - Exigences et modèle de sécurité pour l'authentification de l'agent mobile

Les auteurs de l'article [81] se sont focalisés sur la sécurité du système d'agents mobiles, spécialement la sécurité de la communication qui se déroule dans un système à base d'agents mobiles. Leur travail repose sur une approche d'authentification adoptée pour assurer la sécurité

de l'agent en utilisant des techniques cryptographiques pour assurer une communication cryptée sécurisée en utilisant le protocole SSL / TLS (Secure Sockets Layer/ Transport Layer Security).

L'approche proposée est basée sur l'authentification par signature numérique, qui est une méthode alternative afin de s'identifier auprès d'un serveur de connexion, au lieu de taper le mot de passe. Dans le cas étudié par les auteurs, il s'agit d'identifier l'agent auprès de la plateforme ou du serveur visité sur la base du protocole ssl / tls.

Cette approche est composée des signatures numériques et une infrastructure à clé publique partagée avec le protocole de communication tls. Les auteurs supposent que l'agent fait confiance à d'autres agents qui viennent d'un système approuvé. L'agent doit être signé à chaque fois qu'il est envoyé vers une plateforme, ce qui vérifie la non-répudiation et l'intégrité de l'agent.

Le tableau 3.1 ci-dessous est une comparaison entre les solutions existantes concernant les mécanismes de sécurité utilisés et les objectifs de sécurité qu'ils ont atteints. On voit bien qu'aucune des solutions ne garantit à la fois la confidentialité, l'intégrité, la non-répudiation et la disponibilité.

Article	**Mécanisme de sécurité**	**Confidentialité**	**Intégrité**	**Non-répudiation**	**Disponibilité**
[76]	Reconnaissance, minuteur, agent PA et SA	Non	Non	Non	Oui
[77]	Image sécurisée (SIM)	Oui	Oui	Non	Oui
[78]	Calcul avec fonctions de chiffrement	Oui	Oui	Non	Non
[79]	Blackbox	Oui	Oui	Non	Oui
[80]	Trace cryptographique	Oui	Oui	Non	Non
[81]	signature numérique, protocole SSL/TLS	Non	Oui	Oui	Non

Tab. 3.1 – Tableau comparatif des mécanismes de sécurité existants

Après avoir discuté les travaux liés à la sécurité des agents mobiles, leurs avantages et inconvénients, nous présentons dans la section suivante notre proposition pour la sécurité des agents mobiles et de leurs environnements afin d'assurer l'intégrité du système, la confidentialité des données et la non-répudiation, en plus de la protection de l'agent contre les plateformes malveillantes et les attaques DOS pour assurer la disponibilité du système.

3.4 Proposition d'un nouveau modèle de sécurité des systèmes à base d'agents mobiles

Notre solution se focalise sur la protection des agents contre les attaques d'agents malveillants, les plateformes malveillantes et l'assurance de l'intégrité du système. Nous proposons

donc une approche bidimensionnelle pour traiter les menaces de sécurité dans notre système. Notre proposition utilise deux mécanismes de sécurité : la trace cryptographique et l'agent SOS.

- Nous avons adopté la trace cryptographique pour assurer l'intégrité de l'agent mobile et l'authentification de l'origine lors de sa migration d'une plateforme à une autre. Une fois que l'AM (Agent Mobile) est arrivé, la plateforme visitée pourrait procéder à la vérification de la trace cryptographique encapsulée dans son code afin de l'authentifier et récupérer le message.
- Nous avons aussi proposé le modèle d'agent SOS qui vise à protéger l'agent contre les hôtes malveillants et les attaques DOS, puis à assurer la disponibilité du système. Nous avons choisi le nom SOS de cet agent pour montrer qu'il sera déclenché comme signal de détresse et demander de l'aide. Son rôle est de veiller à la sécurité de l'agent mobile en surveillant ses déplacements sur les plateformes visitées à l'aide d'un temporisateur. Si l'agent remplit sa mission avant la fin du délai imparti, il doit envoyer un accusé de réception à l'agent SOS avant de passer à la plateforme suivante, confirmant qu'il est sécurisé et que sa mission a été accomplie avec succès.

  Notre agent SOS, contrairement aux modèles existants, reste dans la plateforme de base, qui est la plus sécurisée. Si le délai est dépassé et qu'aucun message n'a été reçu de l'agent mobile, l'agent SOS identifie la plateforme réellement visitée comme malveillante et l'ajoute à sa liste noire, puis envoie un nouvel agent avec les données accumulées afin de continuer la mission de l'agent précédent.

3.4.1 Proposition du mécanisme de la trace cryptographique

La sécurité de l'agent mobile et de son code est primordiale, car les agents malveillants peuvent tenter d'obtenir un accès non autorisé à l'hôte, ou les hôtes malveillants peuvent extraire des informations confidentielles exportées par l'agent et les utiliser par la suite. La trace cryptographique consiste en une séquence d'identificateurs et d'informations signés par la clé secrète de la plateforme qui envoie le message. Elle est composée d'une séquence de paires (n,s), où n représente des identificateurs uniques et s est la signature.

La signature de la plateforme n'est nécessaire que pour les informations qui dépendent des interactions avec l'environnement informatique. Pour les informations qui ne reposent que sur les valeurs des variables internes, une signature n'est pas requise et, par conséquent, est omise [83].

La trace cryptographique est l'une des méthodes qui assurent une sécurité et une transmission de données efficaces. Nous utilisons la trace cryptographique pour que l'agent local qui a créé et envoyé l'agent léger (LW) puisse suivre ses mouvements et ses itinéraires. En gardant cette trace, nous connaîtrons le chemin parcouru par ce dernier et nous nous assurerons qu'il n'y a pas d'usurpation d'identité d'agent.

Dans notre proposition, nous supposons que chaque hôte possède une clé privée notée Ks et une clé publique notée Kp qui seront utilisées lors du chiffrement et du déchiffrement des messages et signatures. Comme le montre la figure 3.2, notre agent LW migre de la plateforme initiale H0 vers la plateforme H1, puis vers H2 jusqu'à arriver à Hn pour accomplir sa mission et revient à la plateforme d'origine H0 qui l'a créé [4].

Le message encapsulé par l'agent mobile se compose d'un code à exécuter en plus des

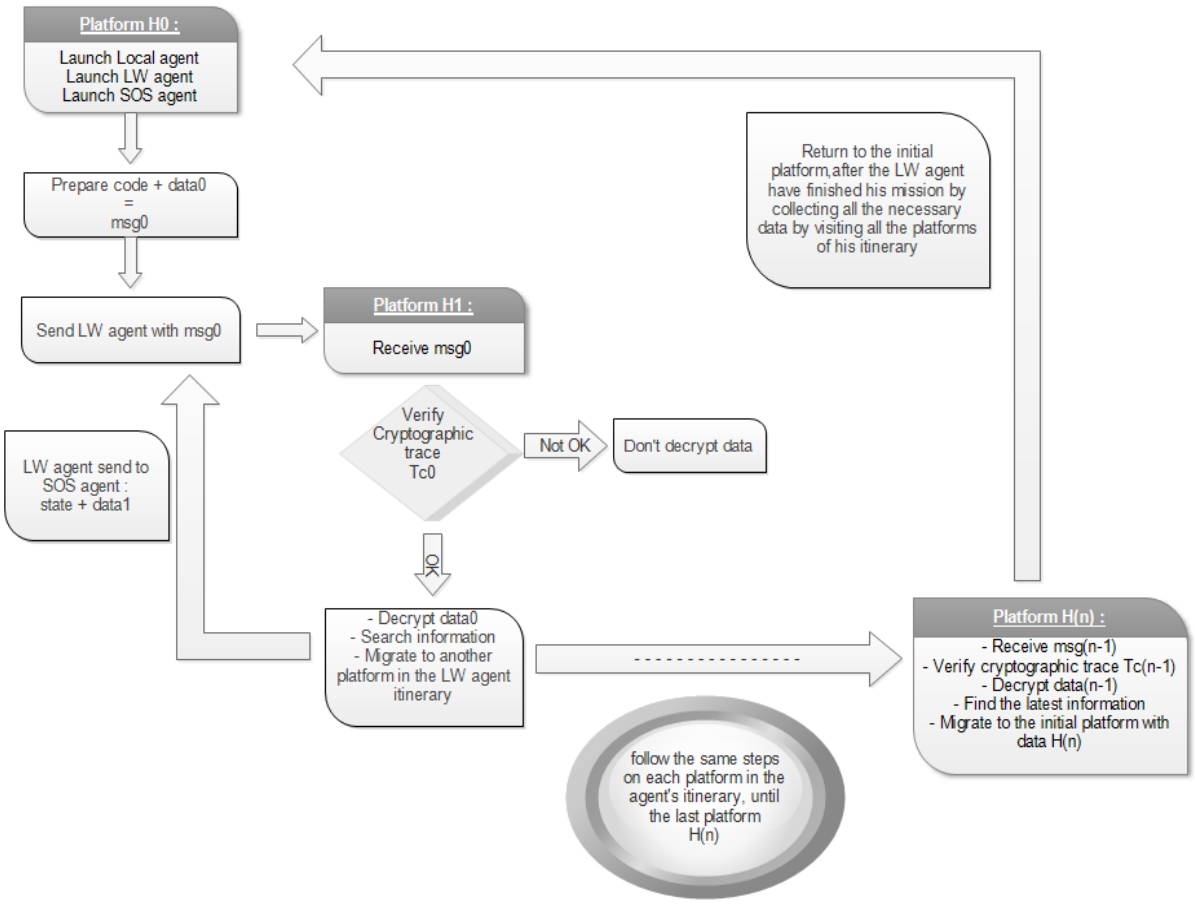

Fig. 3.2 – Schéma explicatif de la transmission des données de l'agent LW [4]

identifiants, de la trace cryptographique et du hachage pour assurer la non-répudiation et la protection de l'intégrité.

Une fois que l'agent LW est lancé et a reçu sa mission, il migre de **la plateforme** H_0 vers **la plateforme** H_1 avec le message suivant :

$msg_0 =$ id H_0 , @IP H_0 , @IP H_1 , $Hash_0$, Tc_0 , $Encrypt_{Kp1}(data_0)$

Le message envoyé msg_0 contient les paramètres suivants :

- id H_0 : l'identifiant de la plateforme émettrice.
- @IP H_0 : l'adresse IP de la plateforme émettrice.
- @IP H_1 : l'adresse IP de la plateforme réceptrice.

Ces trois premières entités spécifient que le message provient de H_0 et est dirigé vers H_1.

- $Hash_0 = Hash_{Ks0}$(idH_0, id LW agent) : le hachage de l'identifiant unique de l'agent LW et l'identifiant de la plateforme initiale H_0 avec la clé secrète H_0. $Hash_0$ est envoyé dans chaque message, afin d'être vérifié à la fin de l'itinéraire de l'agent par la plateforme initiale pour montrer que c'est lui même l'agent LW qui a été déployé au début de la mission. $Hash_0$ est également utilisé pour calculer la trace cryptographique.
- $Tc_0 = Signé_{Ks0}$(id H_0, @IP H_0, @ IP H_1, $Hash_0$) : la trace cryptographique qui est le signé de l'id et de l'adresse IP de H_0, l'adresse IP de H_1 et $Hash_0$ avec la clé secrète de H_0. Cette trace cryptographique montre que ce message provient de la plateforme H_0 et est destiné à la plateforme H_1. Une fois que H_1 vérifie que ce message provient de la

plateforme H_0 et qu'il est destiné à H_1 grâce à cette trace cryptographique, la plateforme procède au déchiffrement des données et à l'exécution du code agent.

- $Encrypt_{Kp1}$ $(data_0)$: les données envoyées dans le message sont chiffrées avec la clé publique de la plateforme H_1. Ces données seront déchiffrées par la clé privée de la plateforme qui a reçu le message, ici H_1.

Au sein de la plateforme H_1 :

- Une fois que la plateforme H_1 reçoit le message msg_0, elle procède à la vérification de la trace cryptographique afin de connaître la source du message et s'il est destiné à cette plateforme ou non.
- La trace cryptographique Tc_0 comme nous l'avons vu ci-dessus est le signe de l'identifiant et de l'adresse IP de H_0 et l'adresse IP de H_1 et du hachage avec la clé secrète de H_0. Donc pour la vérification, la plateforme H_1 déchiffre cette trace avec la clé publique de H_0 pour s'assurer de l'origine et de la destination du message.
- Lorsque la vérification de la trace cryptographique est concluante, la plateforme traite les données chiffrées envoyées dans le msg_0. La plateforme déchiffre $Encrypt_{Kp1}$ $(data_0)$ avec sa clé privée puis l'agent accomplit sa mission sur cette plateforme.
- Une fois le traitement des données terminé, l'agent LW enverra le résultat des données traitées $(data_1)$ à l'agent SOS avec le message "aller au suivant", comme nous le verrons dans la deuxième partie de notre approche (sous-section 3.4.2).

L'agent LW migre de la plateforme H_1 vers H_2 avec le message msg_1 contenant les paramètres suivants :

msg_1 = id H_1, @IP H_1, @IP H_2, $Hash_0$, Tc_1, Tc_0, $Encrypt_{Kp2}(data_1)$

- Les trois premières entités identifiant H_1, @IP H_1, @IP H_2 spécifient que le message provient de H_1 et est dirigé vers H_2.
- $Hash_0$ comme expliqué ci-dessus, il est envoyé dans chaque message pour être vérifié à la fin par la plateforme d'origine.
- $Encrypt_{Kp2}$ $(data_1)$: les données sont chiffrées avec la clé publique de la plateforme H_2. Ces données seront déchiffrées par la clé privée de la plateforme H_2 qui recevra le message.
- La trace cryptographique $Tc_1 = Signed_{Ks1}$ (idH_1, @IP H_1, @ IP H_2, $Hash_0$, Tc_0) est vérifiée par la plateforme qui reçoit le message.
- La trace cryptographique Tc_0 est ajoutée à la fois dans Tc_1 et msg_1. Elle est ajoutée à la signature H_1 pour garantir l'intégrité du système et elle est ajoutée à msg_1 pour permettre à H_2 d'effectuer la vérification de la signature.

Pour généraliser ce processus, nous présentons le cas des plateformes H_i et H_{i+1} où $0 < i \leq n$.

Entre la plateforme H_i et H_{i+1} :

Les mêmes étapes décrites ci-dessus sont suivies jusqu'à la plateforme H_i, où le message est le suivant :

msg_i = id H_i, @IP H_i, @IP H_{i+1}, $Hash_0$, Tc_i, Tc_{i-1}, $Encrypt_{Kpi+1}(data_i)$ et la trace cryptographique est comme suit :

$Tc_i = Signed_{Ksi}(\text{id}H_i, \text{@IP } H_i, \text{@IP } H_{i+1}, Hash_0, Tc_{i-1})$

Entre la plateforme H_n et H_0 :

Lorsque l'agent LW termine sa mission, il revient sur la plateforme initiale avec le résultat de l'exécution.

H_n envoie le message suivant msg_n à H_0 :

$msg_n = \text{id } H_n, \text{@IP } H_n, \text{@IP } H_0, Hash_0, Tc_n, Tc_{n-1}, Encrypt_{Kp0}(data_n)$

Au sein de la plateforme H_0 :

- La plateforme H_0 vérifie la trace cryptographique Tc_n pour connaître l'émetteur du message et le destinataire, de la même manière expliquée précédemment :
  $Tc_n = Signed_{Ks0}(\text{id}H_n, \text{@IP } H_n, \text{@IP } H_0, Hash_0, Tc_{n-1})$
- Ensuite, la deuxième étape consiste à vérifier le $Hash_0$. Ce paramètre est inclus dans chaque message de l'itinéraire de l'agent qui est vérifié une fois que la plateforme initiale reçoit le message final pour garantir l'intégrité du système.
  - $Hash_0 = Hash_{Ks0}(\text{id}H_0, \text{id LW agent})$: le hachage de l'identifiant unique de l'agent LW et l'identifiant de la plateforme initiale H_0 avec la clé secrète H_0.
  - Lorsque la plateforme reçoit le $Hash_0$, qui est une valeur de 32 bits ou une valeur de 64 bits selon la fonction de hachage utilisée, elle calcule celle attendue à partir de l'identifiant de la plateforme d'accueil et de l'identifiant de l'agent LW à l'aide de sa clé secrète. Ensuite, elle compare la valeur reçue avec celle calculée.

    Si les deux valeurs sont identiques, cela veut dire que les informations et le code envoyés n'ont pas été changés ou modifiés tout au long de l'itinéraire de l'agent.
- Ensuite, H_0 déchiffre les données reçues $Encrypt_{Kp0}$ $(data_n)$.

Après avoir présenté la méthode proposée pour la trace cryptographique pour garantir l'intégrité de l'agent mobile et l'authentification de l'origine, dans la sous-section suivante, nous présenterons un nouveau mécanisme pour suivre la migration de l'agent afin d'éviter les attaques DOS et d'assurer la disponibilité du système et le bon fonctionnement de l'agent.

3.4.2 Le mécanisme d'agent SOS

Dans un système multi-agents typique, chaque plateforme utilise deux agents spéciaux : un agent local (LA- Local Agent) et un agent léger (LW- LightWeight). L'agent local crée et attribue des missions à l'agent léger. Ce dernier (LW) migre de la plateforme d'origine H_0 vers d'autres plateformes H = { H_1, H_2 ... H_N } comme vu précédemment, et recherche les informations souhaitées en fonction des missions assignées. Une fois la tâche terminée, l'agent LW revient à la plateforme d'accueil. Pendant son voyage, un ou plusieurs hôtes dans l'itinéraire spécifié pourraient être malveillants et bloqueraient l'agent LW.

Même si nous avons proposé le mécanisme de trace cryptographique pour garantir l'intégrité des données, connaître l'itinéraire de l'agent LW et être sûr qu'il n'y a pas d'usurpation d'identité d'agent, nous devons toujours garantir la disponibilité de l'agent mobile en évitant les attaques DOS.

En effet, le mécanisme proposé dans cette sous-section se focalise sur la détection des attaques par déni de service sur les agents LW lorsqu'ils migrent pour effectuer une tâche assignée. Précisément, il utilise un nouvel agent appelé agent SOS qui utilise le même concept d'accusé de réception et de délai d'attente que celui utilisé dans l'agent SA pour surveiller les mouvements de l'agent PA dans le papier [76]. La principale différence est qu'au lieu de rester derrière l'itinéraire de l'agent PA d'une ou deux étapes, notre agent SOS restera sur la plateforme d'accueil, la plateforme la plus sécurisée pour un agent mobile, et surveillera les mouvements du LW. En utilisant cette approche, l'agent SOS ne sera pas une cible d'attaque comme ce fut le cas avec l'agent ombre (SA).

Lorsque l'agent local (LA) attribue une tâche à l'agent léger (LW) :

1. l'agent LW commence son itinéraire (H_i, i = 0 ... N) tandis que l'agent SOS restera dans l'hôte d'origine H_0.
2. Lorsque l'agent LW termine sa tâche dans H_i, il envoie un accusé de réception à l'agent SOS avec le résultat de son exécution dans H_i, $Encrypt_{Kp0}$ ($data_i$), les données accumulées. L'idée est que l'agent LW envoie un accusé de réception et le résultat de sa mission à chaque fois qu'il termine sa tâche sur un nouvel hôte.
3. Ensuite, l'agent passera à la plateforme suivante H_{i+1}.
4. Si l'agent SOS reçoit l'accusé de réception, nous pouvons supposer que H_i est considéré comme non bloquante, ce qui signifie que la tâche a été exécutée en toute sécurité et ainsi l'agent peut passer à l'hôte suivant H_{i+1}. Dans ce cas, l'agent SOS recalcule le temporisateur en fonction de l'emplacement du prochain hôte afin de surveiller le nouvel hôte visité et attend à nouveau le nouvel acquittement avec les nouvelles données accumulées.
5. Par contre, si le temporisateur spécifié utilisé par l'agent SOS expire, et puisque nous savons déjà que H_{i-1} est considéré comme non bloquant car nous avons déjà reçu l'accusé de réception précédent (en utilisant la même logique), alors nous sommes sûr que l'hôte H_i est malveillant et des mesures correctives doivent être prises.

Avec cette approche, l'agent SOS peut identifier et mettre sur la liste noire tous les hôtes malveillants. En d'autres termes, lorsque l'hôte d'origine lance une nouvelle instance de l'agent léger LW_{new}, ce dernier reçoit les données accumulées de l'agent SOS et migre directement vers le dernier hôte non bloquant visité H_{i-1} pour continuer sa mission. Mais cette fois, il ignorera l'hôte malveillant en fonction de sa liste noire, l'hôte H_i, et reprendra sa mission dans la partie restante de l'itinéraire en commençant par H_{i+1}.

Il est important de noter que puisque l'agent SOS reste dans l'hôte d'accueil (l'hôte le plus sécurisé), l'intégrité des données accumulées est protégée de toute modification ou ingénierie inverse [82]. De plus, nous avons utilisé une interprétation différente du concept d'accusé de réception et de délai d'expiration : l'agent SOS utilise un temporisateur T pour vérifier si l'agent LW a exécuté sa tâche en toute sécurité dans l'hôte actuel avant de migrer vers le suivant au lieu de vérifier si l'agent PA sera bloqué par l'un des deux hôtes de destination avant d'effectuer la tâche [76]. Cette différence offre l'avantage d'identifier et de mettre sur liste noire tout hôte malveillant détecté.

La figure 3.3 montre le fonctionnement de l'agent SOS, les étapes sont les suivants :

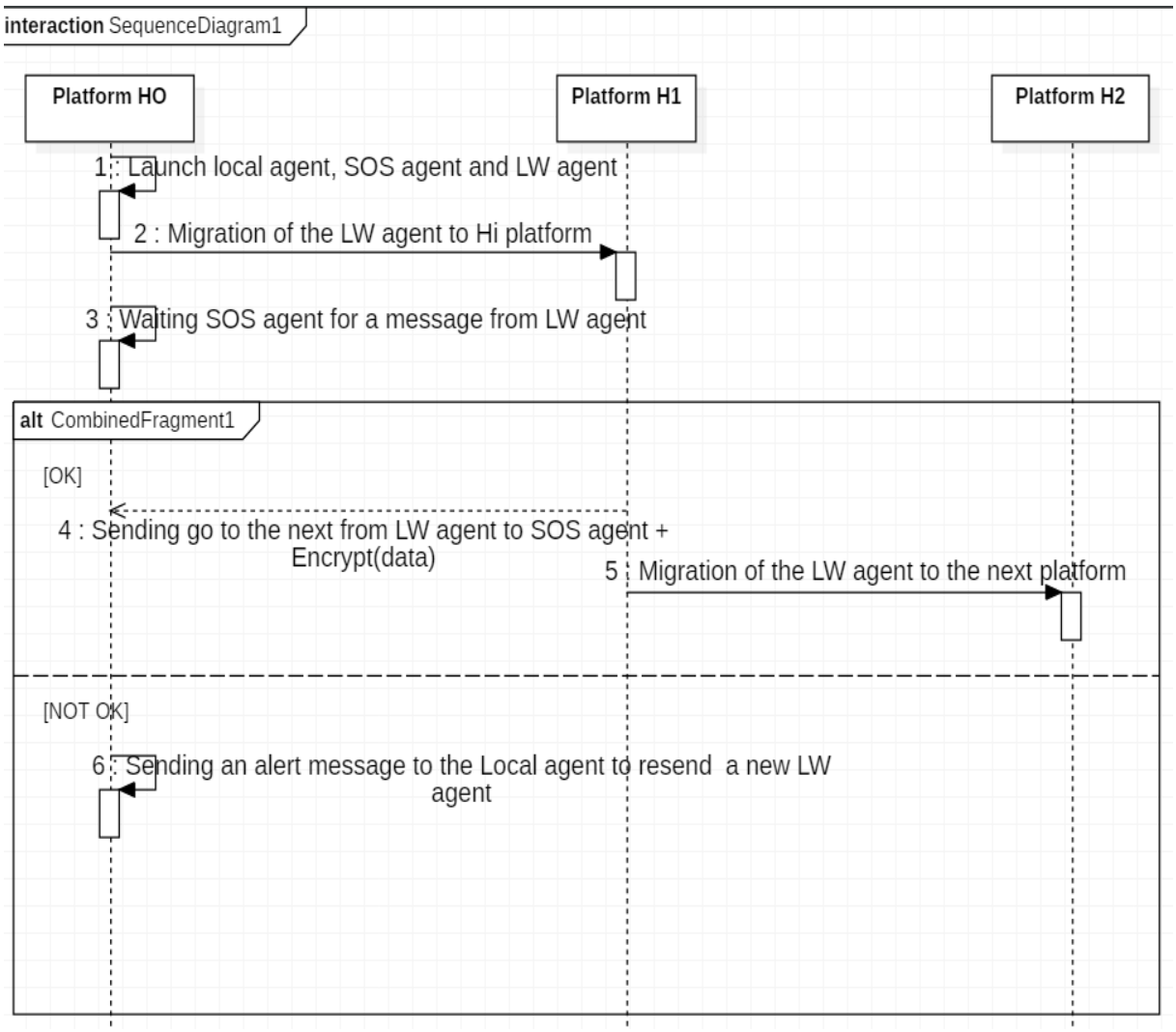

Fig. 3.3 – Interaction entre l'agent SOS et l'agent LW [4]

1. Lancement d'un agent local, d'un agent LW et d'un agent SOS, chacun avec sa mission.
2. L'agent LW migre vers la plateforme Hi (i=0...n) pour exécuter sa tâche.
3. L'agent SOS attend le message de l'agent LW selon le délai donné par le temporisateur T. Après cette étape, deux scénarios sont possibles :
4. Cas 1 : Si l'agent SOS reçoit le message "Aller à la suivante" de l'agent LW en plus des données chiffrées, cela signifie que sa mission s'est bien déroulée chez l'hôte actuel. L'agent SOS recalcule le temporisateur et attend à nouveau son message.
5. Une fois que l'agent LW envoie "Aller à la suivante" à l'agent SOS, il migre vers la plateforme suivante pour continuer sa tâche, etc.
6. Cas 2 : si le temporisateur s'est écoulé et que l'agent LW n'a pas envoyé de message à l'agent SOS, l'agent SOS envoie une alerte à l'agent local pour déployer un nouvel agent LW. Si une réponse tardive de LW vient après l'écoulement de T, elle sera ignorée.

Dans la section suivante, nous décrivons la mise en œuvre et analysons en détail notre approche d'agent SOS.

3.4.3 Mise en œuvre de l'approche d'agent SOS

Le but de l'approche d'agent SOS est de détecter l'attaque par déni de service par un hôte malveillant qui bloque un agent mobile en visite et l'empêche de poursuivre son itinéraire. Nous avons implémenté notre approche proposée en utilisant Java Agent DEvelopment Framework (la plate-forme JADE 4.5.0) [84].

Au cours de la mise en œuvre, de nombreux scénarios ont été testés à partir du cas le plus trivial où tous les hôtes sont considérés comme non bloquants pour en final se trouver avec plusieurs hôtes malveillants. Dans tous ces scénarios, l'agent SOS a pu détecter tous les hôtes malveillants simulés, permettant à l'agent léger LW de les ignorer tous et de ne visiter que ceux qui ne bloquent pas. Cela donne confiance dans la validité et montre la faisabilité de l'approche proposée.

a - Simulation d'une attaque SOS Afin de simuler une attaque DOS effectuée par un hôte malveillant, nous avons utilisé des messages ACL (Agent Communication Language) avec des réponses et des demandes spécifiques échangées entre l'agent léger LW et un agent local LA de chaque hôte visité.

- Lorsque l'agent léger LW migre vers un hôte H_i pour exécuter une tâche affectée, il envoie une demande "Bonjour" à l'agent local de H_i.
- Si l'hôte H_i est malveillant, son agent local répondra par un message ACL avec le contenu "malveillant", puis l'agent léger LW se terminera simplement. Dans un scénario réel, l'agent léger LW se terminera en raison d'une action malveillante de l'hôte malveillant.
- Si l'hôte H_i est amical, l'agent local de Hi enverra un message ACL "amical". Dans ce cas, l'agent LW s'exécutera normalement et reprendra son itinéraire par la suite.

  C'est le rôle de l'agent SOS de détecter chaque hôte malveillant dans l'environnement.

b - Étude de cas Dans ce qui suit, nous prenons un cas d'étude afin d'expliquer le fonctionnement de l'approche SOS. Dans cet exemple, l'agent LW se déplace dans un itinéraire de huit hôtes simulés H_1 à H_8. Comme le montre la figure 3.4, les hôtes H_2, H_3, H_5 et H_7 sont des hôtes malveillants tandis que les autres sont des hôtes non bloquants.

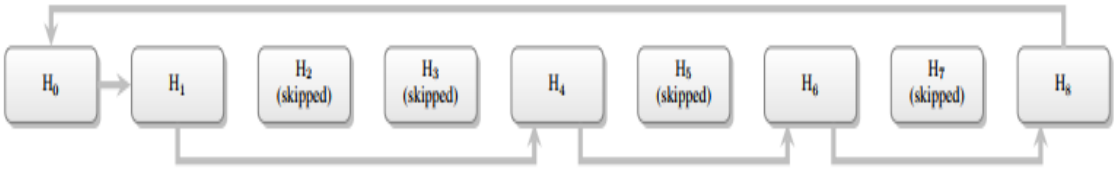

Fig. 3.4 – Simulation de l'environnement [4]

Dans cet exemple, le scénario suivant est simulé comme suit :

- Dans l'hôte H_0, l'agent SOS et l'agent local LA sont initialement créés.
- L'agent local LA déploie un nouvel agent mobile léger LW avec une tâche spécifique à exécuter et quelques informations supplémentaires (par exemple l'itinéraire). L'agent SOS reste toujours dans l'hôte H_0 et l'agent LW se déplace à la visite de l'hôte H_1.

- L'agent SOS attend un accusé de réception de l'agent LW qui l'envoie lorsqu'il termine son exécution dans la plateforme H_1 et avant sa migration vers la prochaine plateforme de son itinéraire qui est H_2. L'agent SOS utilise un temporisateur T qui est calculé en fonction de l'emplacement de l'agent LW et du temps d'exécution de la tâche affectée. Le temps T doit être suffisamment long pour recevoir un accusé de réception s'il n'y a aucune action malveillante.
- Une fois que l'agent LW a terminé sa tâche, il envoie un accusé de réception à l'agent SOS et passe à l'hôte suivant H_2.
- Si l'agent SOS reçoit l'accusé de réception avant l'expiration du temporisateur T, cela signifie que l'hôte H_1 n'est pas un hôte malveillant et que l'agent LW se déplace vers la plateforme H_2. L'agent SOS recalcule le temporisateur T en conséquence.
- L'hôte H_2 étant malveillant, il bloquera l'agent LW et mettra fin à son exécution. Par conséquent, l'agent LW ne peut pas envoyer l'accusé de réception.
- Lorsque le temporisateur T expire, l'agent SOS décide que L'hôte H_2 a bloqué l'agent LW. Dans ce cas, il enverra une demande à l'agent local indiquant que l'agent LW est bloqué dans L'hôte H_2 et que le dernier hôte non bloquant visité est L'hôte H_1.
- À ce stade, l'hôte H_0 va créer une nouvelle instance d'agent LW avec le nouvel itinéraire H_1, H_3 ... H_8 en ignorant l'hôte malveillant H_2.
- Lorsque la nouvelle instance de l'agent LW arrive au dernier hôte non bloquant visité, qui est H_1 selon les données accumulées, l'agent LW renverra l'accusé de réception à SOS indiquant que l'hôte H_1 n'est pas bloquant et qu'il se déplace vers H_3. Dans ce cas, l'agent SOS définira son temporisateur T pour qu'il soit prêt à recevoir l'accusé de réception suivant lorsque l'agent LW terminera sa tâche en H_3.
- L'hôte H_3 est malveillant, il bloquera donc l'agent LW. Le même scénario va continuer à détecter et d'ignorer chaque hôte malveillant jusqu'à ce que l'agent LW atteigne finalement l'hôte H_8 et revienne à l'hôte H_0.

Notre approche proposée a le principal avantage d'identifier exactement un hôte malveillant potentiel à l'aide de temporisateur et de reconnaissances. Le fonctionnement de l'agent SOS se déroule comme le montre l'algorithme 1 ci dessous.

L'agent local LA de l'hôte H_0 gère le déploiement des agents légers LW. Il utilise les étapes indiquées dans l'algorithme 2.

Algorithm 1 Algorithme général du fonctionnement de l'agent SOS

Données : $Listenoire, Listeblanche, Itinéraire$
- Lancer l'interface utilisateur de l'agent SOS
// Tous les paramètres sont vides.
Si le bouton 'Déployer LW' est enfoncé alors,
// L'agent SOS enverra dans la demande l'itinéraire H_1, ... H_n à utiliser.
- Envoyer une demande à l'agent local
// Après le déploiement, l'agent léger LW va migrer de manière autonome vers le premier hôte H_1.
- Démarrer le temporisateur T et attendre 'Aller à la prochaine : id du dernier conteneur' comme accusé de réception
// Attendre la confirmation avant l'expiration du délai T.
Si l'accusé de réception est reçu alors,
// Ajouter le dernier hôte non bloquant visité à la liste blanche.
- Mettre à jour la liste blanche.
- Redémarrer le temporisateur.
Fin
Si le délai de T expire alors,
- Mettre à jour la liste noire.
- Envoyer une demande à l'agent local.
// La demande contient la liste noire mise à jour, la liste blanche mise à jour, l'itinéraire avec le dernier hôte non bloquant visité.
- Redémarrer le temporisateur.
Fin
Fin

Algorithm 2 Algorithme général du fonctionnement de l'agent local dans l'hôte H_0

Data : $Listenoire, Listeblanche, Itinéraire$
- Attendre une demande de l'agent SOS.
Si la demande est reçue alors,
- Mettre à jour les données.
- Déployer l'agent léger LW avec de nouvelles informations.
// Dans le premier déploiement (lorsqu'aucune donnée n'est accumulée), l'agent LW va commencer à partir de l'hôte H_1.
Fin

Nous avons personnalisé notre agent SOS avec une interface utilisateur graphique pour surveiller son comportement. Cette interface graphique fournit les informations suivantes (voir figure 3.5) :

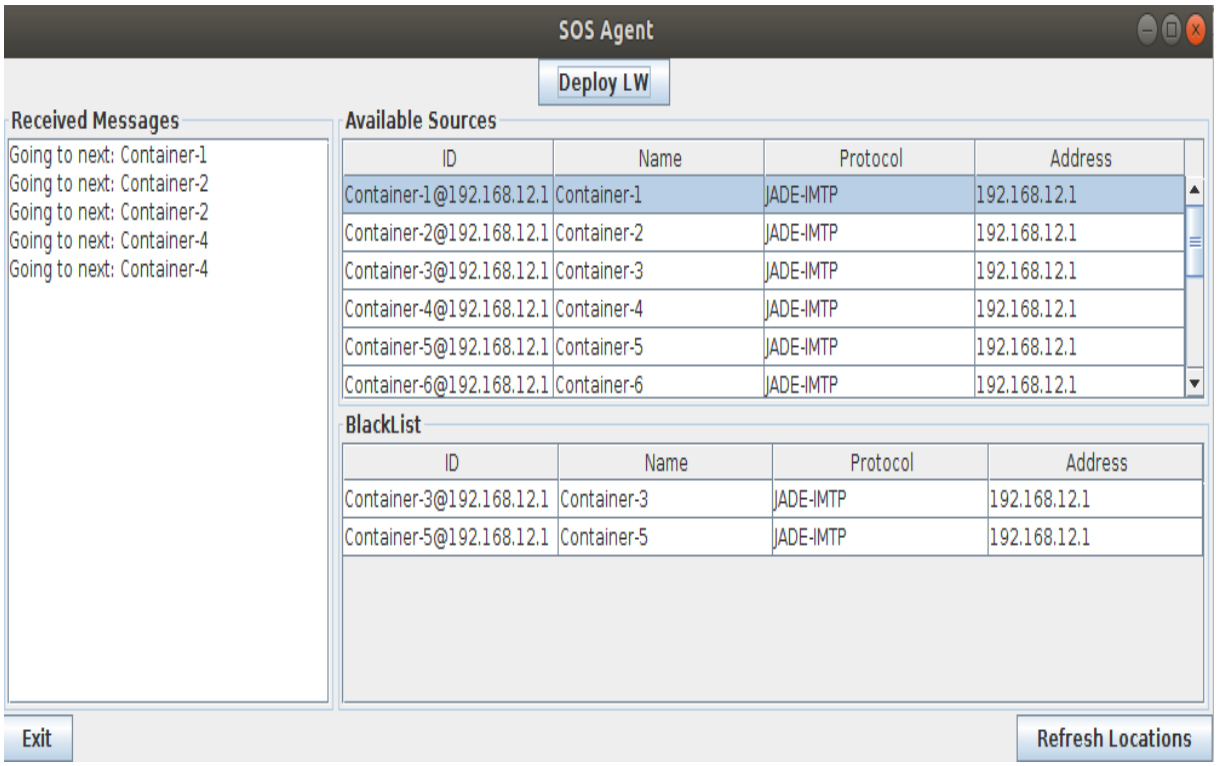

Fig. 3.5 – Interface de l'agent SOS [4]

- Toutes les sources / hôtes disponibles dans l'environnement.
- Une liste noire contenant la liste des hôtes malveillants détectés jusqu'à présent.
- Le contenu des messages ACL envoyés par l'agent léger LW (dans ce cas, c'est : "Aller à la suivante" : l'id du dernier hôte visité).
- Un bouton pour déployer l'agent LW : lorsque ce bouton est enfoncé, l'agent SOS enverra une demande à l'agent local LA de l'hôte H_0 pour déployer un nouvel agent LW. Ce bouton n'est pressé qu'une seule fois.

Dans ce cas d'étude, l'agent léger LW simulé a réussi à visiter tous les hôtes non bloquants et à ignorer tous les hôtes malveillants. L'agent léger LW a commencé son voyage à partir de l'hôte d'origine H_0 et y est retourné avec succès, comme il est illustré sur la figure 3.6.

```
Value of Dst is: null
Agent: LW0 is ready !
LW0 is now moving elsewhere.
Good hosts:[Container-1]
R.I.P Agent.
Malicious hosts: [Container-2]
Value of Dst is: Container-1
Agent: LW1 is ready !
LW1 is now moving elsewhere.
Good hosts:[Container-1]
R.I.P Agent.
Malicious hosts: [Container-2, Container-3]
Value of Dst is: Container-1
Agent: LW2 is ready !
LW2 is now moving elsewhere.
Good hosts:[Container-1]
Good hosts:[Container-1, Container-4]
R.I.P Agent.
Malicious hosts: [Container-2, Container-3, Container-5]
Value of Dst is: Container-4
Agent: LW3 is ready !
LW3 is now moving elsewhere.
Good hosts:[Container-1, Container-4]
Good hosts:[Container-1, Container-4, Container-6]
R.I.P Agent.
Malicious hosts: [Container-2, Container-3, Container-5, Container-7
Value of Dst is: Container-6
Agent: LW4 is ready !
LW4 is now moving elsewhere.
Good hosts:[Container-1, Container-4, Container-6]
Good hosts:[Container-1, Container-4, Container-6, Container-8]
Lightweight agent returned safely
```

Fig. 3.6 – Console de l'agent SOS

La figure 3.6 montre également que l'agent SOS garde la trace des hôtes malveillants détectés jusqu'à présent ainsi que la liste des hôtes non bloquants (étiquetés "bons hôtes"). Ainsi, chaque fois qu'un agent léger LW est interrompu par un hôte malveillant, l'agent SOS va déployer un nouvel agent LW avec les informations accumulées comme décrit précédemment pour aider le nouvel agent à ignorer et éviter l'hôte malveillant.

3.5 Conclusion

Dans ce chapitre, nous nous sommes concentrés sur l'aspect de la sécurité des agents mobiles lors de la communication et de la migration vers d'autres hôtes afin de se rapprocher des ressources distantes. Notre objectif principal était donc de fournir une sécurité optimale aux systèmes basés sur des agents mobiles. Nous avons décrit les menaces et les exigences de sécurité auxquelles est confrontée la technologie des agents mobiles. Nous avons également présenté certains travaux connexes et souligné certains de leurs avantages et inconvénients.

Ensuite, nous avons présenté notre approche proposée dans laquelle nous avons traité les exigences de sécurité particulièrement la protection des agents malveillants contre les attaques d'agents malveillants, les plateformes malveillantes et l'intégrité du système. Pour améliorer le profil de sécurité, nous avons pris en compte des attaques comme les attaques DOS. Nous avons adopté l'utilisation de la trace cryptographique pour garantir l'intégrité du système lors de la migration de l'agent et pour être sûr qu'il n'y a pas eu de vol d'identité.

Dans notre approche de la sécurité, nous avons créé également un agent appelé agent SOS qui est lancé dans la plateforme d'origine afin de surveiller l'agent léger LW lors de son déplacement vers un autre hôte. L'agent SOS utilise des accusés de réception et une période d'attente pour détecter les attaques DOS et les contourner. Cette approche a été mise en œuvre et testée dans une étude de cas.

Comme nous pouvons le voir, les mécanismes proposés dans notre approche garantissent :

- L'intégrité du système en plus de l'authentification de l'origine grâce à l'utilisation de la trace cryptographique qui contient des informations sur l'émetteur et le récepteur du message.
- La non-répudiation qui est garantie par la signature.
- La confidentialité des données à l'aide d'un chiffrement asymétrique.
- La disponibilité du système et la protection contre les attaques DOS grâce au mécanisme d'agent SOS proposé.

Sur le tableau suivant 3.2, nous donnons une comparaison entre les modèles existants et notre approche proposée.

Article	**Mécanisme de sécurité**	**Confidentialité**	**Intégrité**	**Non-répudiation**	**Disponibilité**
[76]	Reconnaissance, minuteur, agent PA et SA	Non	Non	Non	Oui
[77]	Image sécurisée (SIM)	Oui	Oui	Non	Oui
[78]	Calcul avec fonctions de chiffrement	Oui	Oui	Non	Non
[79]	Blackbox	Oui	Oui	Non	Oui
[80]	Trace cryptographique	Oui	Oui	Non	Non
[81]	signature numérique, protocole SSL/TLS	Non	Oui	Oui	Non
Notre proposition	Agent SOS, trace cryptographique	Oui	Oui	Oui	Oui

Tab. 3.2 – Tableau comparatif entre les mécanismes de sécurité existants et notre proposition

Dans le chapitre qui suit, nous nous intéressons aux soins de santé intelligents qui utilisent aussi le paradigme des agents mobiles, en plus d'autres technologies récentes.

Chapitre 4

Amélioration d'une architecture de soins de santé intelligents basée sur l'intégration de l'IoT et les agents mobiles

Aujourd'hui, on parle beaucoup des environnements intelligents et en particulier des villes intelligentes. Cette transformation d'une ville traditionnelle en une ville intelligente vise à offrir aux citoyens un mode de vie amélioré, sain et de qualité. Pour cela, plusieurs secteurs ont mis en place des concepts innovants avec de nouvelles technologies informatiques et de communication pour créer cet écosystème de ville intelligente (Smart City), comme le secteur de la santé qui est un secteur vital pour les citoyens d'une ville et qui est au cœur de notre sujet.

Dans ce chapitre, nous visons à donner un aperçu général du concept de soins de santé intelligents et le rôle de l'Internet des objets (Internet of things IoT) dans ce dernier. Nous nous sommes penchés spécialement sur les défis de sécurité présentés par les soins de santé intelligents puisque dans notre étude nous avons choisi comme domaine Smart healthcare. Puis, nous présentons une architecture existante de soins de santé intelligents typique, nous discutons les problématiques posées par cette architecture et nous proposons une amélioration en y intégrant la communication à base d'agents mobiles.

4.1 Introduction

Nous vivons dans un monde défini par l'urbanisation et l'ubiquité numérique, où les connexions mobiles à large bande sont plus nombreuses que les fixes, les machines dominent un nouvel Internet des objets et plus de personnes vivent dans les villes qu'à la campagne. Dans la transformation des villes traditionnelles en villes intelligentes, il y a une tendance croissante dans le monde entier vers les infrastructures dynamiques intelligentes qui fournissent aux citoyens de nouveaux services qui peuvent changer en mieux leur qualité de vie et imprégner les caractères d'efficacité. Concernant cela, un défi majeur et qui est primordial est de connaître la manière d'accorder aux villes et aux citoyens le meilleur moyen pour améliorer la détection des données en ce qui concerne divers facteurs différents.

Le concept de Smart City s'inscrit dans une tendance générale qui est l'avènement l'Internet des objets. Les appareils et périphériques deviennent aptes à contrôler, quadriller leur environ-

nement, transmettre l'état des appareils utilisés et optimiser la consommation d'énergie afin de répondre convenablement aux besoins des citoyens, des entreprises et des institutions dans différents domaines.

Les projets de ville intelligente les plus fréquents comprennent les maisons intelligentes, les systèmes de transport intelligents, l'éclairage intelligent, les administrations intelligents, les hôpitaux intelligents, etc. Ces nouvelles technologies sont basées essentiellement sur la collecte des données en plus de la manière dont celles-ci sont analysées par les capteurs. Elles procurent des solutions innovantes et économiques au nombre croissant de défis auxquels sont confrontés les citoyens et les gouvernements.

Dans ce chapitre, nous nous concentrons sur le secteur de la santé qui est un secteur vital et sur son intégration dans la ville intelligente en tant que domaine de soins de santé intelligent. En fait, l'amélioration de l'efficacité des infrastructures de santé est l'un des objectifs les plus difficile de la société intelligente. À l'époque actuelle, les procédures de gestion et de suivi des patients sont dans la plupart du temps réalisées manuellement par des infirmières ou des aides soignantes. Pour cela, les avancées technologiques récentes amènent au développement des systèmes intelligents afin d'améliorer les soins de santé, à savoir l'identification et le suivi des personnes automatiquement et à distance, la surveillance des paramètres des patients en temps réel et n'importe où.

Néanmoins, malgré les innombrables avantages qu'apporte la ville intelligente, de nombreux défis demeurent persistants lors du déploiement et la mise en place d'un environnement intelligent, en raison des besoins spécifiques de la ville et des interprétations multiples des concepts de déploiement, c'est-à-dire les défis technologiques, l'interopérabilité entre systèmes, le manque de confiance des citoyens qui sont considérés comme des utilisateurs bénéficiant des services que la ville intelligente va leur procurer, les données en plus de la sécurité numérique.

L'objectif principal de ce chapitre est d'étudier le concept des soins de santé dans la ville intelligente, l'utilisation de l'Internet des objets et du système multi-agents (SMA) dans les soins de santé intelligents, puis d'identifier et discuter les principaux problèmes et défis de la sécurité. Le reste de ce chapitre est organisé comme suit : nous abordons le concept de santé intelligente et nous commençons par définir la ville intelligente, l'expansion de la santé intelligente, puis les exigences de la santé intelligente. Ensuite, nous discutons l'utilisation et le déploiement de l'Internet des objets et des systèmes multi-agents dans les soins de santé intelligents. Enfin, nous présentons une architecture de soins de santé intelligents typique et nous montrons les problématiques de cette dernière. Nous proposons une amélioration de cette architecture qui repose sur la couche réseau en intégrant le modèle d'agents mobiles afin d'assurer une communication efficace en temps quasi-réel et assurer la disponibilité du système.

4.2 Concept de soins de santé intelligents

4.2.1 Caractéristiques d'une ville intelligente

Jusqu'à ce moment là, il n'y a pas d'explication définitive ou globale de la ville intelligente, mais on trouve des définitions dans la littérature, les plus courantes sont :

- La ville intelligente signifie la combinaison des technologies de l'information et de la communication afin d'améliorer la qualité et la performance des services urbains comme l'énergie, le transport et les services publics pour minimiser la consommation des ressources, les déchets et les coûts globaux [85].
- La ville intelligente offre de meilleurs services urbains et une meilleure qualité de vie avec des infrastructures connectées et intelligentes pour les résidents et les visiteurs [86].

- La ville intelligente consiste à offrir de meilleures conditions de vie aux citoyens, tout en la rendant plus durable, résistante et vivable. La technologie est l'épine dorsale d'une ville intelligente. Une ville peut être définie comme intelligente lorsque les investissements dans le capital humain et social, les infrastructures de communications traditionnelles (transports) et modernes (TIC) alimentent un développement économique durable et une qualité de vie élevée, avec une gestion avisée des ressources naturelles, grâce à une action et un engagement participatif [87].
- Les villes intelligentes sont des villes fortement fondées sur les technologies de l'information et de la communication qui investissent dans le capital humain et social pour améliorer la qualité de vie de leurs citoyens en favorisant la croissance économique, la gouvernance participative, la gestion rationnelle des ressources, la durabilité et une mobilité efficace, tout en garantissant l'intimité et la sécurité des citoyens [88].

Par conséquent, une ville intelligente est une ville qui cherche à fournir un environnement sain et une meilleure qualité de vie aux citoyens dans différents domaines et à réagir rapidement et plus efficacement aux besoins des citoyens. Elle rassemble la technologie, le gouvernement et la société pour permettre les fonctionnalités suivantes : environnement intelligent, économie intelligente, mobilité intelligente, vie intelligente et gouvernance intelligente.

Lorsque nous investissons dans le capital humain et social, ainsi que dans les infrastructures traditionnelles (transports) et modernes (TIC) créant un développement économique durable et une qualité de vie élevée grâce à la gestion intelligente des ressources naturelles par la gouvernance et les citoyens, nous appelons cette ville Smart City.

Une ville devient "intelligente" lorsque toutes les parties de ses services d'infrastructure et du gouvernement sont connectées et optimisées numériquement. Comme le montre la figure 4.1, une ville intelligente inclut l'intelligence dans différents secteurs et domaines tels que :

- mobilité intelligente : rendre le système de transport plus flexible et mieux adapté aux besoins des citoyens.
- maison intelligente : également appelée domotique, facilite la vie quotidienne à la maison. La maison intelligente offre aux propriétaires une sécurité, un confort, une commodité et une efficacité énergétique en leur permettant de contrôler les appareils intelligents en partageant les données d'utilisation des consommateurs entre eux et en automatisant les actions en fonction des préférences des propriétaires.
- société intelligente : est un ensemble de systèmes qui expriment un ensemble de valeurs conventionnellement établies. Tout découle de ces valeurs. Il s'agit d'un domaine dans lequel les dirigeants et les citoyens prennent des décisions fondées sur des données qui permettent d'améliorer constamment les résultats en matière de prospérité économique, de bien-être social, de durabilité environnementale et de bonne gouvernance.
- soin intelligent : système de dossier de santé électronique pour contrôler et prévenir les

Fig. 4.1 – Une ville intelligente [89]

maladies et suivre les patients partout et à tout moment.

4.2.2 Expansion et exigences des soins de santé intelligents

Ces dernières années, le secteur de la santé a connu des progrès et un développement dans ses différents domaines : équipements, méthodes de soins et de sauvetage, etc. Mais avec l'augmentation du nombre de citoyens en plus des maladies, il devient essentiel de trouver des solutions pour réagir et prendre soin des citoyens rapidement, efficacement, partout et à tout moment. Pour cela, l'adoption de nouvelles technologies informatiques a permis un changement radical du secteur de la santé.

Avec ces nouvelles technologies informatiques, le secteur de la santé est passé du modèle traditionnel consistant en un médecin examinant un patient individuel pour générer un diagnostic et à recommander un traitement au nouveau modèle en utilisant de nouvelles technologies qui permettent au médecin d'extraire en temps réel les antécédents du patient, connaître son historique médical et suivre son état.

Nous pouvons classer les soins de santé intelligents en 4 catégories, comme le montre la figure 4.2 :

- Dispositifs et technologies de connectivité : cette catégorie comprend les dispositifs intégrant des capteurs, des actionneurs, des micro-contrôleurs qui aident à la surveillance et à l'observation à distance en détectant les signaux chimiques, biologiques et physiques du patient. Il existe plusieurs types de capteurs et les plus utilisés dans le monde des soins de santé intelligents sont : capteur de température, capteur de sang, capteur de pouls et de tension, capteur cardiaque. Leur rôle est de recueillir des informations et de les renvoyer dans une base de données centrale.

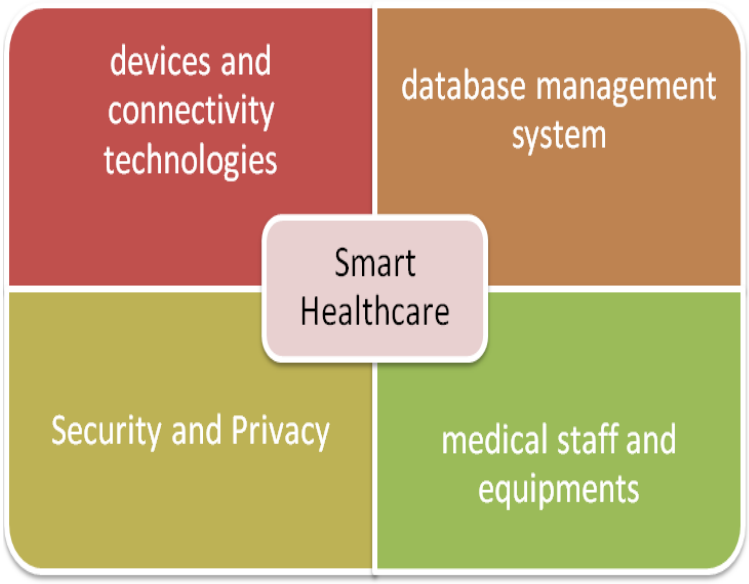

Fig. 4.2 – Exigences des soins de santé intelligents

Les technologies de connectivité sont primordiales dans les applications Smart Healthcare. Ces technologies sans fil telles que Wi-Fi, Zigbee, 6LowPan aident à surveiller l'état de santé à distance via l'Internet des objets.

- Système de gestion de base de données : afin de pouvoir suivre la santé d'un patient, on doit avoir son historique médical en plus des informations le concernant. Ce système de gestion des bases de données permet de créer et gérer des utilisateurs, en plus de mettre à jour les données.
- Sécurité et confidentialité : la confidentialité est l'un des plus grands défis des soins de santé intelligents et des villes intelligentes. Les données personnelles sont considérées comme des données sensibles, les gens s'inquiètent de la façon dont leurs données sont utilisées par d'autres parties. La recherche sur les questions de confidentialité suggère que les gens évaluent le but pour lequel les données sont utilisées et les avantages que ces données peuvent apporter.
- Personnel médical et équipements : une solution de gestion des actifs médicaux est nécessaire pour permettre au personnel de localiser rapidement les équipements, de surveiller les patients, de se charger de la maintenance des équipements.

Après avoir présenté le concept de la ville intelligente et l'intégration de l'intelligence dans le domaine de soins de santé, nous analysons dans la partie qui suit l'utilisation de deux technologies qui jouent un rôle primordial dans les soins de santé intelligents qui sont les systèmes multi-agents et l'IoT.

4.3 Utilisation de l'IoT et SMA dans les soins intelligents de santé

Dans cette section, nous présentons les différents domaines de soins intelligents de santé et donnons des exemples de travaux intégrant l'IoT et les SMA dans ces soins de santé.

4.3.1 Domaines d'applications dans les soins de santé

Tout d'abord, il faut savoir que lorsque nous parlons de soins intelligents de santé nous insinuons les réseaux WBAN qui sont des réseaux sans fil pour le corps (WBAN - Wireless Body Area Network) [90]. Ce sont les éléments de base de la surveillance des soins de santé communautaires. Le suivi des soins de santé communautaires aide à créer un réseau autour de la communauté locale. Les réseaux WBAN multiples constituent un réseau de santé communautaire et les réseaux de santé communautaires multiples constituent un réseau coopératif. WBAN a pour rôle de faciliter et améliorer la qualité du soins et de la surveillance médicale à distance.

La figure 4.3 montre les applications des soins de santé intelligents, qui vont de la surveillance de la condition physique à la surveillance des signes vitaux dans les hôpitaux.

Fig. 4.3 – Domaines d'application dans les soins de santé intelligents [90]

- Un suivi de la forme grâce à une montre intelligente, ainsi que des paramètres tels que le nombre de calories brûlées, les mesures prises.
- Surveillance à distance des patients en fonction de leurs maladies en suivant leur pression, pouls, température, etc.
- Les applications de gestion de l'environnement aident à établir la communication entre l'hôpital et le patient, à commencer les premiers soins dans l'ambulance en contactant l'hôpital et à informer le personnel de l'arrivée du patient.

4.3.2 L'IoT dans les soins de santé intelligents

Définition de l'Internet des objets :

L'Internet des objets, abrégé en IoT, est le nom donné à Internet qui relie le monde réel ou physique à un monde numérique. L'IoT, c'est quand deux objets connectés via Internet

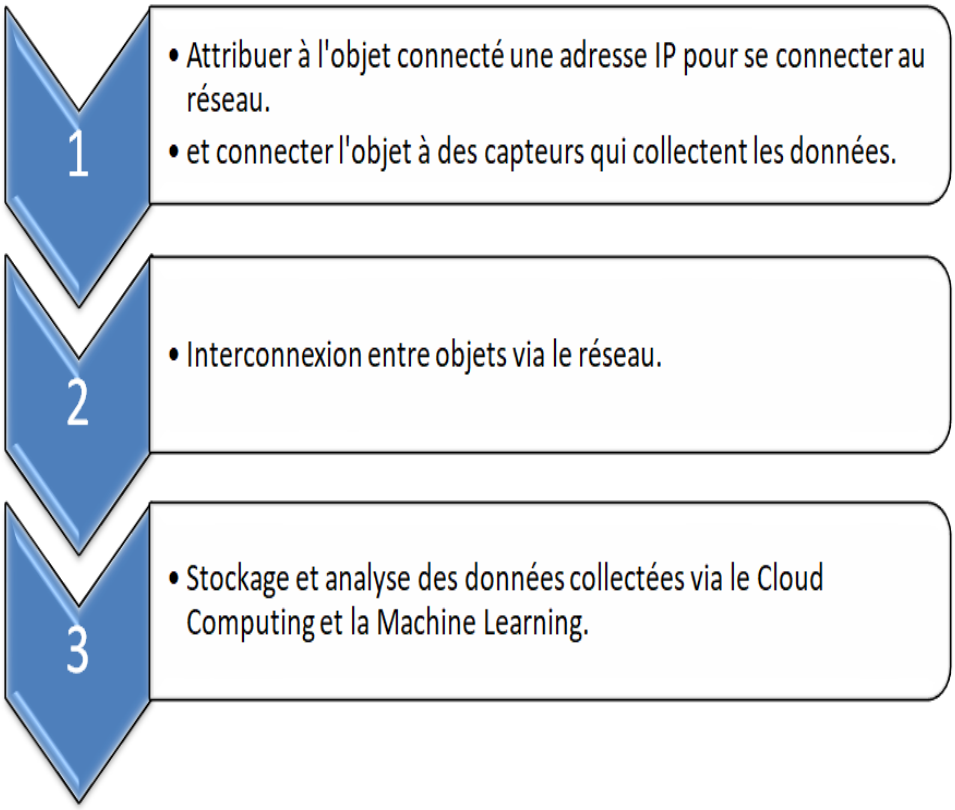

Fig. 4.4 – Fonctionnement des objets connectés

commencent à se parler pour nous faciliter la vie. La figure 4.4 montre le fonctionnement des objets connectés.

Lorsque nous parlons d'IoT, nous parlons d'un objet connecté. Un objet connecté est un objet physique qui remplit une ou plusieurs fonctions dans le monde réel, mais une fois connecté au monde virtuel de l'informatique, il devient un objet contrôlé à distance afin de remplir des tâches qui lui sont dédiées. Ainsi pour qu'un objet soit connecté, il doit avoir un échange entre le monde physique et le monde numérique. Parmi les fonctions principales d'un objet connecté, nous trouvons : collecte d'information provenant de son environnement, déclenchement d'une action en fonction des informations captées et transmises.

Technologie de communication :

L'Internet des objets est l'Internet qui transmet les données collectées par les objets connectés. Parmi les éléments nécessaires de ces objets connectés on trouve le protocole de communication qui est utilisé afin d'échanger les données et les informations entre différents appareils.

Les développeurs travaillant sur des produits et des systèmes pour l'IoT ont un choix déconcertant d'options de connectivité. Mais selon l'application, des facteurs spécifiques tels que les exigences en matière de données, les problèmes de sécurité aident à déterminer la ou les technologies à utiliser parmi les choix existants.

Le tableau 4.1 compare les principales technologies de communication utilisées dans l'IoT en fonction de leur standard, fréquence, portée et vitesse de transmission. Il résume les principales

différences entre les protocoles de communications les plus répandus. Chaque protocole est basé sur une norme et fournit un certain débit de données. Nous remarquons que le WI-FI fournit un débit plus élevé comparé aux autres protocoles. Le tableau décrit aussi chaque protocole avec sa plage de communication, et sa puissance de transmission.

L'IoT dans les soins de santé intelligents :

Pour les soins de santé, l'IoT représente un environnement émergent qui changera la façon dont les technologies de l'information et de la communication sont utilisées et qui exercera probablement une variété de fonctions, y compris le diagnostic, la surveillance, le traitement et la vie assistée.

Les architectures basées sur l'IoT sont utilisées pour collecter des informations médicales auprès de l'utilisateur. L'Internet des objets fonctionne comme un pont entre le médecin et le patient en fournissant un accès à distance, ce qui peut aider le médecin à surveiller en permanence le patient et à donner une consultation à distance. Combinant des capteurs, des actionneurs, des micro contrôleurs, des processeurs et l'informatique en nuage, l'IoT aide à obtenir des résultats précis et rend les soins de santé accessibles à tous et partout. L'utilisation de l'IoT dans les soins de santé a conduit les chercheurs du monde entier à concevoir des cadres et des technologies prometteuses qui peuvent fournir une assistance médicale à tout le monde [90].

Dans ce qui suit, nous présentons des travaux connexes qui montrent l'intégration de l'IoT dans les soins intelligents de santé.

a - Caractéristiques de conception d'un système de santé intelligent en tant qu'application IoT

Dans l'article [91], les auteurs présentent "IoT Healthcare" pour la collecte de données afin de fournir aux patients un système de soins de santé offrant un concept d'hôpital intelligent pouvant servir n'importe qui, n'importe quand et n'importe où. Pour atteindre ce système intelligent, l'IoT est suggéré de composer l'architecture du Healthcare IoT comme illustré sur la figure 4.5.

Les données médicales sont collectées par des capteurs pilotés par l'IoT qui collectent des données de surveillance en temps réel à partir de capteurs intelligents. Les données sont collectés via un réseau mobile et intelligent, puis analysées sur le Cloud Computing, qui fournit des ressources de stockage flexibles, fiables et puissantes qui prennent en charge l'informatique à grande échelle via la virtualisation, l'intégration dynamique des données et la combinaison de plusieurs sources de données.

L'inconvénient qui se pose lors de la création d'un concept adéquat d'hôpital intelligent (prototype) est que la technologie et les outils informatiques se développent de manière considérable, puisque le monde technologique ne cessent d'évoluer et de se diversifier .

Protocol	Standard	Frequency	Range	Transmission speeds
Bluetooth	Bluetooth basic specification 4.2	2,4 GHz(ISM)	50-150 m(Smart/BLE)	1Mbit/s (Smart/BLE)
Zigbee	Zigbee 3.0 based on IEEE802.15.4e	2,4 GHz	10-100m	250 Kbit/s
Z-Wave	Z-Wave Alliance ZAD12837/ITU-TG9959	900 MHz(ISM)	30m	9.6/40/100 Kbit/s
6LowPAN	RFC6282	Bluetooth Smart(2.4 GHz), Zigbee or Low Power RF(Sub-GHz)	N/A	N/A
Thread	based on IEE802.15.4 and 6LowPAN	2.4 GHz(ISM)	N/A	N/A
WI-FI	based on 802.11n	2.4 GHz and 5GHz bands	about 50 m	600 Mbit/s max
Cellular technology	GSM/GPRS (2G), UMTS(3G), LTE(4G)	900/1800/1900/2100 MHz	35 Km max for GSM, 200 Km max for HSPA	35-170 Kbit/s(GPRS), 120-384 Kbit/s(EDGE), 384 Kbit/s-2 Mbit/s(UMTS), 600 Kbit/s-10 Mbit/s(HSPA), 3-10 Mbit/s(LTE)
NFC	ISO/CEI18000-3	13,56 MHz(ISM)	10cm	100-420 Kbit/s
Sigfox	Sigfox	900 Mhz	30-50 Km(rural environments), 3-10 Km(urban environments)	10-1000 bit/s
Neul	Neul	900 MHz(ISM), 458 MHz(UK), 470-790 MHz(White space)	10 Km	a few bit/s to 100 Kbit/s
LORaWAN	LORaWAN	variable	15 Km(rural environments), 2-5 Km(urban environments)	0,3-50 Kbit/s

Tab. 4.1 – Tableau comparatif entre différents protocoles de communication

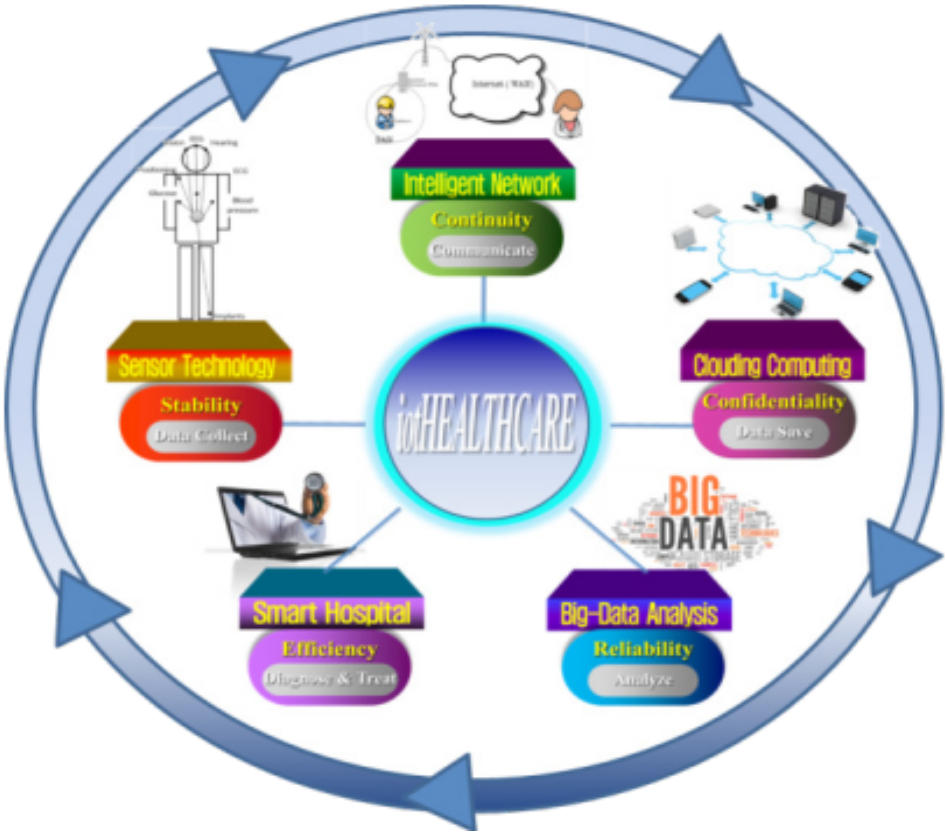

Fig. 4.5 – Architecture logique de IoT Healthcare [91]

b - Mise en œuvre d'un système de soins de santé intelligent à base d'IoT

L'objectif principal de l'article [92] est d'intégrer une architecture sensible à l'IoT pour améliorer les systèmes de soins de santé intelligents pour la surveillance environnementale automatique de l'hôpital (température, humidité, lumière ambiante, etc.) et de la santé des patients (c.-à-d. température et fréquence cardiaque).

Le système proposé par les auteurs collecte en temps réel les conditions environnementales et les paramètres physiologiques des patients et les transmet à un centre de contrôle. Avec la vision de l'IoT à l'esprit, une infrastructure réseau complexe reposant sur les paradigmes CoAP [7] (Constrained Application Protocol), 6LoWPAN [8] (IPv6 Low Power wireless Area Networks) et REST [9] (REpresentational State Transfer) a été mise en œuvre afin de permettre l'interopérabilité entre les UHF RFID Gen2 (Ultra High Frequency Radio Frequency IDentification), le réseau de capteurs sans fil (WSN - Wireless Sensor Network) et les technologies mobiles intelligentes.

Le système d'hôpital intelligent conçu a été mis en œuvre selon l'architecture qui se compose

7. c'est un protocole de transfert Web optimisé pour les périphériques utilisés dans les réseaux de capteurs sans fil pour former l'IoT

8. 6LOWPAN défini les mécanismes d'encapsulation et de compression d'entêtes permettant aux paquets IPv6 d'être envoyés ou reçus via le protocole de communication IEEE 802.15.4 (protocole de communication destiné aux réseaux sans fil de la famille WPAN pour leur faible consommation, de leur faible portée et du faible débit des dispositifs appliquant ce protocole)

9. REST est un style d'architecture logicielle définissant un ensemble de contraintes à appliquer afin de créer des services web assurant l'interopérabilité entre les systèmes informatiques sur Internet

de trois parties principales : WSN amélioré RFID (Radio Frequency IDentification) [10], la passerelle intelligente IoT, les interfaces utilisateur pour la visualisation et la gestion des données. La même interface permet aux médecins disposant de privilèges spécifiques d'accéder à la fois aux données en temps réel et historiques des patients. Ces informations peuvent également être gérées à distance par le personnel médical via une application logicielle mobile.

Les résultats obtenus prouvent la pertinence du système proposé par les auteurs, puisqu'il permet non seulement l'identification et le suivi des patients, du personnel infirmier et des appareils biomédicaux dans les hôpitaux et les instituts de soins infirmiers, mais également pour fournir une surveillance à distance efficace des patients et une gestion immédiate des urgences.

c - Internet des objets pour le cyberhealthacre (IoT4C) : diffusion de l'information, interopérabilité des systèmes et sécurité

Le but de cet article [93] est d'examiner les moyens d'améliorer les soins de santé en les impliquant dans l'IoT, en développant des techniques efficaces de surveillance en temps réel et en améliorant la prestation de services de santé publique. Le système proposé dans cette recherche est un système IoT4C qui vise à connecter plusieurs plateformes, applications et systèmes pour échanger différents types de données et également améliorer les aspects de santé en ligne tels que :

- aspects sécurité des communications de données et confidentialité des systèmes pour assurer la sécurité des données et les rendre accessibles uniquement aux personnes autorisées.
- aspect normalisation et interopérabilité des systèmes pour rendre les données accessibles et échangées partout et à tout moment dans les différents systèmes de santé.

Dans cette partie, nous présentons différents travaux incluant l'internet des objets dans les soins de santé pour améliorer les services médicaux afin d'offrir aux patients des systèmes capables d'être utilisés n'importe où et n'importe quand. Dans la partie qui suit, nous décrivons des travaux connexes utilisant les systèmes multi-agents dans les soins de santé intelligents.

4.3.3 Systèmes multi-agents dans les soins de santé intelligents

Dans cette partie, nous discutons des systèmes de soins intelligents de santé à base de systèmes multi-agents. Ces SMA sont composés d'agents autonomes et intelligents qui coopèrent et interagissent entre eux afin d'exécuter les tâches qui lui sont administrées.

Systèmes multi-agents pour la cybersanté et télémédecine

Dans l'article [94], les auteurs discutent de l'importance apportée lors de l'utilisation des agents mobiles dans le développement des applications en e-santé pour l'échange de données des patients et la surveillance à distance. Selon les auteurs, la technologie des agents mobiles

10. RFID est une technologie pour mémoriser et récupérer des données à distance en utilisant des rayonnement électromagnétiques

est une technologie prometteuse avec sa capacité flexible de coordination et d'interaction qui permet aux agents mobiles de coopérer pour atteindre des objectifs communs de ressources et de tâches et de permettre un certain degré d'automatisation, ce qui correspond au nouvel environnement dans lequel les soins de santé est fourni.

Intégration du paradigme agent mobile pour les nouveaux modèles de service dans l'ambulance intelligente

Les auteurs de l'article [3] ont introduit la technologie des agents mobiles dans les ambulances intelligentes pour répondre efficacement au traitement, à la transmission de données et aux interventions rapides en cas d'urgence.

Nous avons proposé dans [3] représenté dans le chapitre 2, section 2.3 un modèle de service d'agent mobile capable de diagnostiquer l'état du patient une fois dans l'ambulance pour administrer les soins d'urgence, trouver un hôpital approprié (nous avons distingué deux cas, le cas où le patient a des antécédents médicaux et attaché à un hôpital et le cas où le patient n'a pas d'antécédents médicaux) en utilisant l'interaction et la coopération entre l'hôpital et l'ambulance. La mise en œuvre des deux cas est faite pour montrer le rôle et la fonction du modèle proposé.

Un système d'information d'agents mobiles pour une surveillance fœtale omniprésente

L'article [95] présente la conception et le développement d'un système d'information distribué basé sur une plateforme mobile multi-agents pour une surveillance fœtale automatisée en temps réel. Pour surmonter les problèmes d'interopérabilité et d'ouverture dans des environnements hétérogènes, l'utilisation d'agents mobiles et le déploiement d'une plateforme dans l'environnement JADE est la solution utilisée.

Avec l'évolution des technologies informatiques, le domaine de soins de santé a vu une migration des soins de santé traditionnels vers des soins de santé intelligents. Au lieu d'attendre les fils d'attente dans les hôpitaux ou les cabinets de médecins, grâce à l'intégration de l'IoT et des SMA dans les systèmes de soins de santé les patients sont capables d'être suivis à distance n'importe quand et n'importe et en se servant de leur historique médical afin de diagnostiquer leur état et définir ce qui est mieux pour eux sans être dans l'obligation de se déplacer à chaque fois.

Ces avantages apportent une certaine flexibilité aux patients et aux domaines de soins de santé, mais le problème qui se pose est l'aspect sécurité. Les données de plusieurs personnes sont collectées et partagées, donc comment garantir la confidentialité de ces données puisqu'elles font partie de la vie privée des personnes en plus d'assurer qu'elles ne seront pas modifiées ou utilisées pour des fins méconnues.

4.4 Défis de la sécurité des soins de santé intelligents

Avec le développement d'applications des soins de santé intelligentes basées sur les nouvelles technologies telles que l'Internet des objets et les systèmes multi-agents, la préoccupation principale reste la protection de la vie privée des utilisateurs des systèmes et des applications servant le domaine de soins de santé, car les données personnelles seront partagées et gérées par différents systèmes de santé. Bien que ces applications aident à fournir de meilleurs soins de santé aux citoyens, elles sont vulnérables aux menaces. Les appareils de santé sont mobiles, ce qui permet à l'utilisateur de se connecter à différents réseaux, tels que les réseaux domestiques, les réseaux de bureau et les réseaux publics. Cela augmente le risque d'attaques sur l'appareil et l'accès aux informations personnelles.

Dans ce qui suit, nous levons le voile sur le côté de la sécurité dans les soins de santé intelligents. Nous présentons les exigences de la sécurité dans ce domaine, ensuite nous discutons des menaces visant les soins de santé intelligents, ainsi que les services de sécurité.

4.4.1 Exigences de sécurité des systèmes de soins intelligents

Les réseaux de santé contiennent des informations personnelles qui peuvent être facilement modifiées lors du transfert et de l'échange de données entre différentes plateformes. Pour cela, la protection des systèmes de santé est primordiale.

Comme le montre la figure 4.6, la sécurité du système informatique vise à garantir :

Fig. 4.6 – Exigences de la sécurité

- **authentification de l'utilisateur** : l'utilisateur doit prouver son identité afin d'accéder aux ressources et données sécurisées. Il existe plusieurs méthodes d'authentification disponibles aujourd'hui selon le niveau d'assurance, mais la plus utilisée est l'authentification forte qui combine plusieurs facteurs pour rendre la tâche plus compliquée pour un attaquant. Dans les soins de santé intelligents, une authentification au moins à deux niveaux doit être mise en œuvre pour garantir l'identité de l'homologue.
- **confidentialité des données** : est une exigence de sécurité clé dans les soins de santé intelligents. Les données qui incluent des informations privées sur l'utilisateur, doivent être partagées uniquement avec les utilisateurs autorisés. Seuls les utilisateurs autorisés doivent avoir accès aux services ou aux ressources.
- **disponibilité du système** : garantir l'accès à l'information ou à un service aux personnes autorisées souvent en temps réel. Un système d'information doit être capable de fournir les résultats voulus par l'utilisateur au bon moment. Dans le domaine de soins de santé, un médecin ou un infirmier doit avoir la possibilité de rafraichir (ajouter et modifier) les données dans le domaine de la santé.
- **intégrité** : indique si les données n'ont pas été modifiées lors de la communication. L'intégrité doit être maintenue dans le système de santé, garantissant aux utilisateurs que les données transmises et reçues ne sont ni altérées ni compromises. Si le périphérique inter-connecté est compromis, le système de sécurité doit s'assurer qu'il n'y a pas d'attaque sur les informations ou le périphérique dans le réseau de soins de santé. Les appareils interconnectés doivent être auto-réparateurs dans une certaine mesure, ce qui garantit que si un appareil tombe en panne, il a un impact minimal dans le domaine de la santé.

4.4.2 Menaces pesant sur les systèmes de soins de santé

Plusieurs menaces de sécurité sont identifiées dans les soins de santé basés sur l'IoT et basés sur les SMA, telles que interception des données des capteurs lors du transfert vers une application de santé sur le système, indisponibilité des services au niveau de l'utilisateur (problème d'utilisation) ou au niveau du serveur (attaques par déni de service), vol d'identité en usurpant l'adresse IP ou l'adresse MAC de l'appareil.

Parmi les attaques de piratage ciblant les soins de santé à base d'IoT et les appareils de santé, on trouve :

- attaque de Scott Erven et Mark Collao [96] sur l'accès à distance à près de 70000 systèmes médicaux appartenant à une grande agence américaine, dont 21 anesthésies, 488 cardiologies, 67 systèmes de médecine nucléaire, 133 systèmes de perfusion et 31 stimulateurs cardiaques. Ces deux chercheurs en sécurité ont utilisé un moteur de recherche Shodan qui peut trouver des appareils IoT connectés à Internet.

  Ils ont créé des pots de miel, qui sont des serveurs spéciaux qui apparaissent comme des dispositifs médicaux. Ces appareils contenaient de fausses données médicales et de réelles vulnérabilités, mais ils avaient également un composant de journalisation. Lorsque les chercheurs ont examiné les journaux collectés par ces pots de miel, ils ont constaté que les attaquants avaient réussi à s'authentifier via SSH sur ces faux dispositifs médicaux.

  Ils ont également découvert que dans la plupart des cas, les attaquants visant à infecter une machine parmi celles connectées à leurs réseaux de zombies. Si les pirates découvrent

que les appareils pourraient les conduire vers d'autres serveurs avec des informations plus sensibles, ils n'hésiteraient pas à mener une attaque plus sophistiquée pour obtenir ces informations précieuses. Ils pourraient également utiliser les appareils pour propager des logiciels malveillants dangereux dans une infrastructure informatique d'hôpital.

- Des attaques par déni de service visant les appareils IoT : il existe plusieurs méthodes pour lancer de telles attaques. Certaines méthodes consistent à utiliser des botnets malveillants, tels que le botnet Mirai capable de créer des botnets à partir des produits IoT voir capteurs.

  Contrairement aux autres réseaux de zombies qui dépendent des machines, les logiciels malveillants Mirai infectent également les appareils connectés à Internet, tels que les caméras et les DVR [11] (Digital Video Recorder), avec des noms d'utilisateur et des mots de passe par défaut. Les pirates ont profité du code source du logiciel malveillant Mirai, après avoir lancé une attaque par déni de service massive contre le site Web de cybercriminalité du journaliste Brian Krebs.

  Depuis que le code source de Mirai a été publié, les pirates ont commencé à développer de nouvelles variantes de logiciels malveillants. Le nombre total d'appareils IoT infectés par le malware Mirai a atteint 493 000 en Octobre 2016, contre 213 000 bots avant la divulgation du code source [97].

- Billy Rios, un chercheur en sécurité qui aide le Département de la sécurité intérieure des États-Unis a prouvé qu'il pouvait administrer à distance une dose létale de médicaments via une pompe à insuline pour patients. Lui et son collègue ont pu déterminer les mots de passe après avoir acquis des logiciels intégrés et des manuels techniques auprès de plusieurs fournisseurs. Il a également réussi à pirater des mots de passe préprogrammés à partir de centaines d'appareils [98].

- Medjacking ou le piratage de dispositifs médicaux : l'infiltration extérieure est une menace pour les dispositifs médicaux et, plus important encore, pour les patients qui en dépendent. Les appareils sont généralement piratés afin que les attaquants puissent accéder à de plus grands systèmes médicaux et voler des informations de santé protégées [98].

Après avoir décrit les exigences de sécurité et les menaces, dans la partie suivante, nous discutons certaines techniques pour assurer l'authentification, la confidentialité des données dans un environnement intelligent.

4.4.3 Services de sécurité existants dans un environnement intelligent

a - Système d'authentification

Le système de protection d'accès est primordial dans les réseaux IoT et en particulier pour les soins de santé à distance afin d'assurer un échange de données fiable. Actuellement, il n'y a pas de norme industrielle reconnue par les fabricants d'appareils IoT qui doit être suivie. L'ETSI [12] (European Telecommunications Standards Institute) vient de publier une norme dans le domaine, le TS 103 645 pour les consommateurs des objets connectés [99].

11. enregistreur de vidéo surveillance.
12. Institut européen des normes de télécommunications est une organisation à but non lucratif basé en France, elle est responsable de la normalisation des technologies de l'information et de la communication et coopère

De nos jours, en dehors des mécanismes de sécurité standards tels que mot de passe, modèles de contrôle d'accès, les fonctions les plus utilisées dans les applications basées sur l'IoT sont des fonctions biologiques telles que l'empreinte digitale, l'iris, la voix, le visage. Ces éléments biologiques sont uniques à chaque personne. Ils sont considérés comme une alternative au mot de passe et à d'autres identifiants pour lever le doute sur l'identité de la personne. Et pour un système qui doit être plus robuste, il est possible de combiner plusieurs fonctions biologiques.

Le fonctionnement d'une authentification biométrique [100] se déroule comme suit :

- Selon le type d'analyse biométrique utilisée, une image ou un son, la capture d'analyse biométrique est essentielle (éliminer et extraire les entités incohérentes).
- Ensuite, vient l'étape d'inscription où une image numérique générée est enregistrée dans un fichier numérique. Ce fichier sera stocké dans une base de données.
- Ensuite, à travers un lecteur vient l'étape de vérification. L'utilisateur se soumet à une phase de vérification dont il sera précisé s'il dispose ou non d'un droit d'accès.

Bien que l'authentification biométrique soit considérée comme une sécurité renforcée, car elle est unique à une personne. le problème se pose lorsqu'un tiers collecte ces données sensibles, cela peut affecter la vie privée et la liberté de l'individu.

b - Confidentialité des données

Différentes enquêtes internationales ont montré que les données des personnes, en particulier les données médicales, financières et civiques, doivent rester confidentielles car elles sont considérées comme des données très sensibles, contrairement aux données sur la personne comme la nationalité, l'âge, etc. Nous comprenons que la confidentialité des données concernant les citoyens dépend du type de données, la finalité et le service pour lesquels ces données sont utilisées.

Dans les soins de santé intelligents, la confidentialité se caractérise par :

- la confidentialité de l'identité d'une personne, en protégeant ses données personnelles.
- la confidentialité de la mobilité d'une personne, en sécurisant les coordonnées de localisation.
- la confidentialité des communications : protéger les canaux de communication contre les écoutes.
- la confidentialité des transactions, protection des demandes et des réponses contre la surveillance.

Nous présentons ci-dessous les mécanismes de sécurité les plus utilisés pour garantir la confidentialité des données.

- Le chiffrement des données est la méthode la plus utilisée pour assurer la protection des données personnelles et confidentielles. En chiffrant les données, nous rendons le message incompréhensible, sauf pour celui qui détient les clés de chiffrement / déchiffrement. Le problème avec le mécanisme de sécurité basé sur la cryptographie est la complexité de la réponse lorsqu'un système est composé de plusieurs nœuds en plus du coût élevé de

avec ses équivalents américains.

son déploiement. Dans ce contexte, le chiffrement homomorphe est devenu un domaine en évolution rapide et l'un des plus étudiés dans les réseaux IoT du côté de la sécurité.

Dans le domaine de la santé, l'ouvrage [101] utilise le chiffrement homomorphe pour sécuriser les données personnelles des patients sur les réseaux. Lorsque l'IoT utilise des nœuds partagés, cette architecture devient vulnérable aux attaques, car un nœud malveillant peut se faire passer pour légitime et voler les identités des utilisateurs ou produire des résultats erronés. Le chiffrement homomorphe joue un rôle important dans ce contexte, car il aide le réseau à identifier et à isoler les nœuds malveillants visant à accéder aux réseaux.

Bien que l'intégration du chiffrement homomorphe dans les réseaux IoT améliore la sécurité de ces architectures, les coûts de calcul peuvent limiter l'application de cette méthode.

- La blockchain est un mécanisme de sécurité utilisé pour garantir la confidentialité des données dans l'IoT. L'avantage de la blockchain est sa capacité à stocker des données de manière distribuée et non centralisée. Les auteurs de [102] ont développé une nouvelle architecture de système de santé basée sur une chaîne de blocs qui permet aux patients de partager et de gérer leurs informations de santé personnelles tout en préservant la confidentialité de leurs données. Ils ont utilisé le mécanisme de blockchain pour s'assurer que les données médicales ne seront pas modifiées en interne ou en externe (ni par les médecins ni par un tiers).

  Dans l'application hospitalière, lorsqu'un patient visite un médecin pour un test sanguin, il chiffre les données avec une clé et envoie au médecin les données chiffrées et la clé. Le médecin utilise cette clé pour déchiffrer le message. Ici, l'hôpital qui gère l'application ne suivra que la trace de l'acte médical et non le résultat.

c - Sécurité, confidentialité et anonymat

Dans le domaine des soins de santé intelligents, la protection de la vie privée et la sécurisation des infrastructures sont un défi incontournable que le milieu de la recherche a encore du mal à relever. La sécurité et la protection de la vie privée sont essentielles dans presque tous les aspects de notre vie. Cependant, dans le contexte d'une ville intelligente, c'est encore plus important du fait que les informations recueillies sont très personnelles. À partir des données collectées dans un système de santé intelligent, il serait possible de déduire les habitudes des citoyens, leur statut social et les informations de santé.

Parmi les types de protection de la vie privée, des techniques d'exploration de données qui sont à l'étude, pourraient offrir une assurance aux habitants des villes intelligentes :

- Les données peuvent être camouflées lors de leur collecte en perturbant les données. Les données collectées peuvent être perturbées en ajoutant (ou en multipliant) du bruit aux données d'origine. Le bruit a une distribution statistique connue et le résultat d'une perturbation de l'utilisation peut être partagé publiquement. [103].
- dans le cas où les données collectées originales seront divulguées à des tiers, un objectif du data mining préservant la confidentialité est de garantir l'anonymat des données. Cela fait référence à l'assurance que les attributs identifiables pour un utilisateur donné ne peuvent pas être distingués d'au moins k-1 autres utilisateurs.

  L'anonymat peut être atteint grâce à des techniques qui incluent la suppression d'attri-

buts sensibles, l'augmentation de la diversité des attributs sensibles ou l'ajout de données synthétiques pour masquer les valeurs réelles, permettant ainsi aux données sensibles de se cacher [103].

Dans le cas des données mobiles, une alternative à l'attribution d'un identifiant constant unique à chaque utilisateur consiste à changer périodiquement les identifiants. Cette modification rend le suivi des utilisateurs dans le temps et l'espace difficile. Un moment idéal pour changer l'identifiant est lorsqu'un utilisateur entre dans un espace avec au moins k-1 autres utilisateurs afin que les anciens et les nouveaux identifiants ne puissent pas être facilement liés. [103]

Dans cette partie, nous avons présenté les exigences en plus des menaces qui visent les soins de santé intelligents. Ensuite, nous avons discuté des différents types d'attaques faites sur les périphériques et services de soins de santé intelligents. En plus, nous avons illustré les mécanismes de sécurité primordiaux et les plus utilisés afin de faire face à ces menaces. Dans la partie qui suit, nous proposons une amélioration de l'architecture typique de soins de santé intelligents proposée par S. Ahmed et al dans [109] pour garantir la vie privée des utilisateurs des applications concernant la télémédecine.

4.5 Amélioration de l'architecture de soins de santé garantissant la vie privée des patients

Les soins de santé associés à l'internet des objets permettent de pratiquer la médecine préventive et visent à fournir aux patients et aux praticiens des services de surveillance à distance afin de minimiser le temps d'attente aux hôpitaux en plus du suivi, le diagnostic et le traitement en temps quasi-réel des états des patients, spécialement atteints de maladies chroniques. Certes cela aide à offrir de meilleurs services aux patients, mais le problème réside dans le risque de fuite des données qui touche la vie privée des utilisateurs.

Pour cela, la législation des États-Unis a imposé des lois qui doivent être appliquées lors de la mise en œuvre d'architectures et de plateformes dédiées au domaine médical afin d'assurer la protection des informations de santé des patients et du personnel médical qui sont créées, reçues, conservées ou transmises via la voie électronique.

Dans ce qui suit, nous allons présenter deux lois qui s'inscrivent dans ce cadre. Puis, nous allons décrire une architecture typique existante de soins de santé intelligents que nous allons améliorer par la suite en introduisant un système multi-agents pour la communication entre les différentes entités. Ce système permettra d'assurer à la fois la confidentialité et la disponibilité du système de soins de santé.

4.5.1 Lois HIPAA et FISMA

a - La loi HIPAA

La loi HIPPA (Health Insurance and Portability Accountability Act) [108, 109] a été adoptée en 1996 afin de protéger la vie privée des patients. Elle est constituée de plusieurs règles, les principales sont :

- Le patient a la possibilité de choisir comment ses informations de santé sont échangées individuellement ;
- Le patient reçoit toutes les informations nécessaires spécifiques à ses informations de santé ;
- Le patient est clairement informé du contexte dans lequel les informations seront échangées et dans quel but ;
- Le patient a le droit d'imposer certaines restrictions à ses dossiers.

La loi HIPAA impose des restrictions aux praticiens de la santé et aux systèmes d'information en place. Les praticiens de la santé reçoivent des directives claires sur la manière dont ces informations doivent être traitées ou partagées. Cela comprend des instructions claires pour s'assurer que les informations d'identification individuelles utilisées pour accéder aux systèmes de santé sont partagées et destinées au seul but du professionnel de la santé. Cependant, ces derniers ne comprennent pas la nature des risques encourus lors du stockage des informations au format numérique et sont généralement bercés dans un faux sentiment de sécurité compte tenu du processus d'autorisation mis en place. Les praticiens des technologies de l'information sont chargés de veiller à la manière dont les données sont stockées et transmises. Ces dernières doivent être transmises de manière sûre [109].

b - La loi FISMA

La loi HIPAA a été renforcée par la loi fédérale de la gestion de la sécurité de l'information FISMA (Federal Information Security Modernization Act) [110] en 2002. Elle a pour but de suivre différents systèmes d'information et de mettre en place les contrôles de sécurité appropriés. La loi FISMA a été ajoutée à la loi HIPAA pour obliger les organismes de santé à mettre en place des mesures pour assurer la confidentialité et la disponibilité des données des patients. FISMA exige une surveillance continue pour les organismes afin de s'assurer que les contrôles de sécurité continuent de fonctionner comme requis.

Après avoir discuté des attentes vis-à-vis des systèmes de soins de santé afin d'accroître la confiance des utilisateurs des applications de santé (patients et personnel médical). Nous présentons l'architecture de soins de santé existante avec ses composants, nous examinons aussi les problématiques de cette architecture. Ensuite, nous proposons une amélioration pour l'architecture de soins de santé intelligents typique tout en garantissant la sécurité des informations médicales selon les lois établies.

4.5.2 Description de l'architecture de soins de santé existante

Dans la section précédente (4.4), nous avons discuté les défis de sécurité des soins de santé intelligents et nous avons présenté les exigences et les menaces qui visent ces plateformes de santé, en plus des services de sécurité existants. Dans le domaine des soins de santé intelligents, le défi majeur consiste en la confidentialité des données des patients. Lors du traitement et le partage des données, ces dernières peuvent être la cible de plusieurs attaques.

Les auteurs de [109] ont présenté un environnement de soins de santé intelligents typique, comme le montre la figure 4.7. La loi FISMA considère que la clé d'une organisation de soins

de santé est la couche réseau. La couche réseau construit la voie de communication pour le transport des données des utilisateurs.

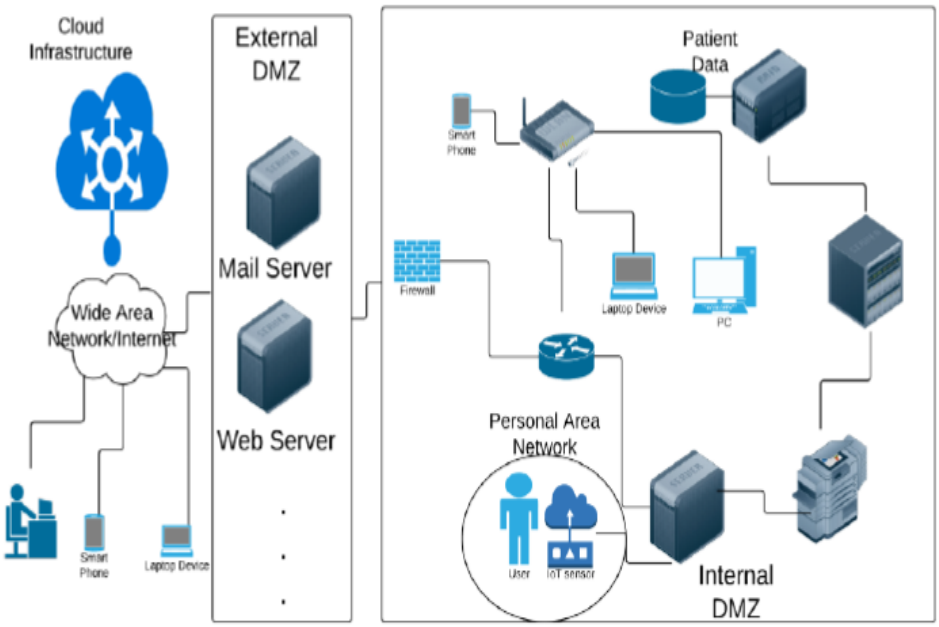

Fig. 4.7 – Environnement de soins de santé intelligents typique [109]

La figure 4.7 montre l'architecture proposée par les auteurs pour un environnement de soins intelligents et qui assure la sécurité des données en utilisant, entre autres, des DMZ (interne et externe). Cette architecture est composée des éléments suivants :

- réseau local LAN (Local Area Network) : un réseau de périphériques connectés existant dans une zone limitée, en plus d'une zone démilitarisée DMZ (Demilitarized zone) interne. Cette zone permet uniquement aux personnes autorisées d'accéder aux ressources numériques protégées, c'est-à-dire elle permet de filtrer les accès. En plus, il y a une DMZ externe qui permet aux membres à l'extérieur du réseau LAN d'une organisation de se connecter et d'accéder à d'autres ressources via Internet (ces services sont placés dans la DMZ externe), même le serveur Web qui héberge le site Web officiel d'une organisation est également placé dans la DMZ externe.
- réseau personnel PAN (Personal Area Network) : est un réseau informatique organisé autour d'une personne individuelle. Il désigne l'interconnexion entre des appareils et équipements informatiques dans la portée de l'espace privé d'une personne, généralement cet espace est limité entre 10 à 20 mètres de la personne ou de l'appareil. Le réseau PAN se compose généralement d'un téléphone portable, d'un ordinateur mobile, d'un assistant numérique personnel. Il est utilisé afin de permettre la communication entre ces appareils à travers des liens filiaires ou sans fil (Bluetooth / Wifi).
- réseau étendu WAN (Wide Area Network) : réseau de télécommunications capable de couvrir une zone géographique très vaste, utilisé pour connecter les réseaux locaux et les clients à Internet.

Problématiques

Les applications médicales exigent qu'une infrastructure réseau capable de prendre en charge les applications médicales soit extrêmement fiable, sécurisée et performante à un niveau élevé. L'utilisation du réseau local LAN traditionnel limite l'efficacité d'un réseau hospitalier puisqu'il ne résout pas les problèmes de suivi et de navigation et les problèmes nécessitant le déploiement d'appareils mobiles.

De plus, l'aspect sécurité est essentiel pour les applications liées au suivi de la santé des patients. Plusieurs systèmes de suivi nécessitent une communication sans fil entre les appareils mobiles intelligents et des appareils informatiques fixes. Lorsqu'un nœud mobile envoie un signal, il se rend vulnérable à tout attaquant susceptible d'écouter passivement la communication sans fil. La gestion de la sécurité comprend aussi la protection de la vie privée des patients et du personnel médical, en plus de la protection du réseau contre les attaques malveillantes et les pannes.

Il ne faut aussi pas oublier que dans le domaine de soins de santé intelligents, certains cas de patients nécessitent des soins intensifs et doivent être suivis en temps réel. Pour cela, il faut que l'architecture proposée par les auteurs soit capable à faire face aux attaques par déni de service qui risque de surcharger le système par des requêtes afin de le rendre indisponible.

Dans l'architecture proposée par les auteurs comme environnement des soins de santé intelligents typique, nous constatons que l'aspect "intelligent" est plus au moins absent. Pour cela, nous nous focalisons sur la partie qui rend l'environnement intelligent, c'est-à-dire, l'IoT et la communication entre les parties tierces.

Les menaces de sécurité les plus répandues visant un système des soins de santé visent surtout la vie privée de l'utilisateur et la disponibilité du système. Notre amélioration de l'architecture de soins de santé intelligents typique repose sur la couche réseau. Elle consiste à intégrer la technologie d'agents mobiles afin d'assurer la transmission des données en temps quasi-réel tout en veillant à la disponibilité du système en plus de l'intégrité et la confidentialité des données pour répondre aux problématiques présentées ci-dessus.

Dans ce qui suit, nous proposons un scénario afin de décrire la communication qui se déroule entre un patient se trouvant chez lui qui veut interagir avec l'hôpital. Tout d'abord, nous allons présenter les types de réseaux utilisés dans notre scénario qui vise à partager et protéger les informations médicales entre différentes entités de l'hôpital dans un réseau externe. Et afin d'assurer la confidentialité, l'authentification et l'intégrité des données transmises, nous intégrons notre solution d'agent SOS présentée dans le chapitre 3.

4.5.3 Proposition de l'architecture améliorée de soins de santé intelligents

L'objectif de cette architecture est de construire une infrastructure réseau filière et sans fil innovante pour les soins de santé intelligents. Elle vise à améliorer la communication entre patients et personnel médical (médecin, infirmier) en permettant d'une façon continue la collecte, l'interprétation et la gestion des données des patients en temps réel. Nous présentons le cas où le patient se connecte de chez lui ou d'un endroit se trouvant à l'extérieur de l'hôpital.

La figure 4.8 montre notre architecture améliorée afin qu'un patient puisse interagir avec le

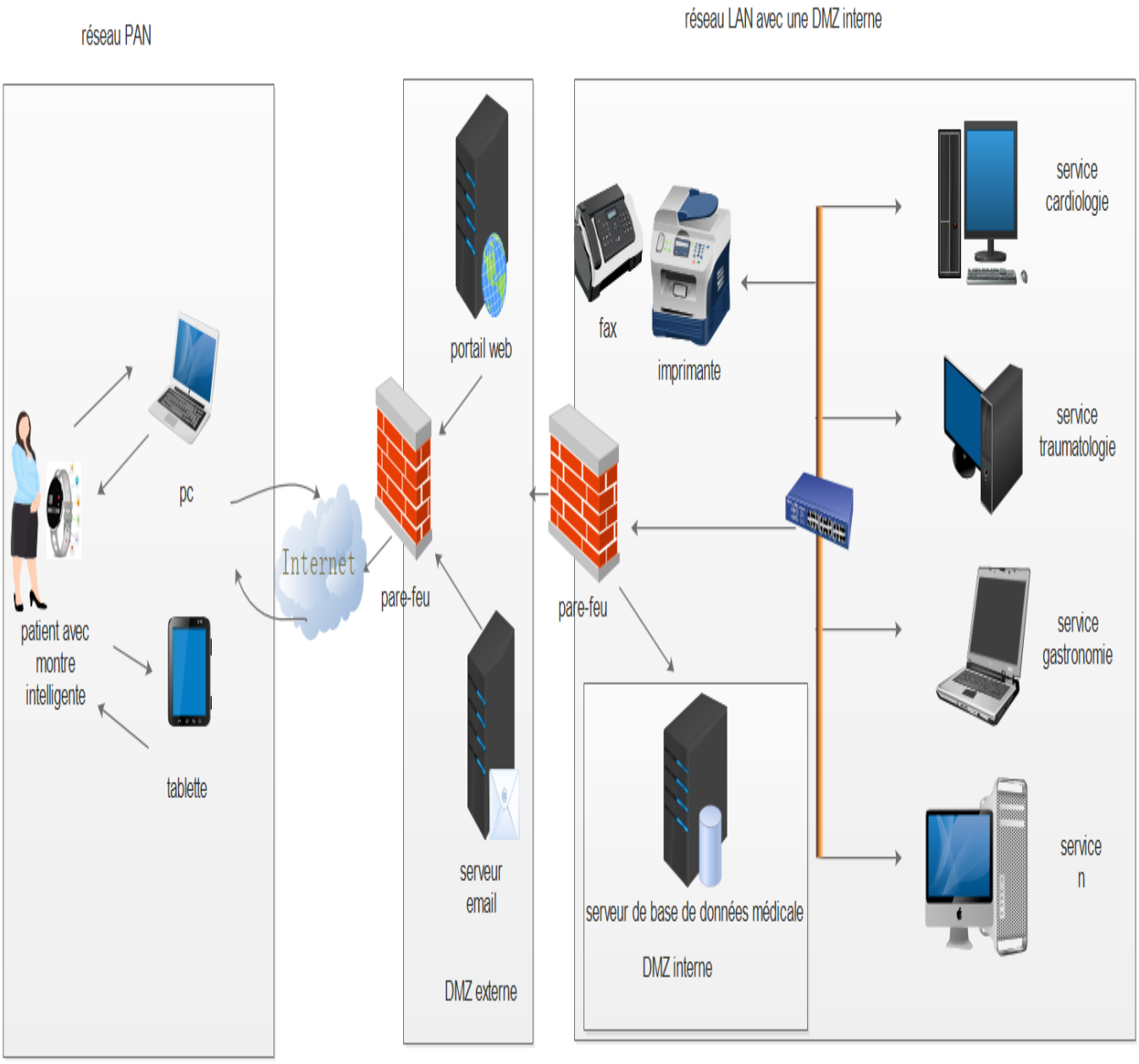

Fig. 4.8 – Architecture intelligente proposée

serveur médical n'importe quand et n'importe où.

Le réseau personnel PAN est constitué de :

- une montre intelligente : appareil qui est porté sur le poignet du patient et qui collecte ses paramètres de santé tels que : la température, le pouls artériel, la tension, etc.
- un téléphone portable, ordinateur ou tablette : les données collectées par la montre intelligente sont transmises via Bluetooth ou WIFI à l'ordinateur, téléphone portable ou tablette personnelle du patient dont la tâche est de fournir ces données collectées du patient au centre de surveillance via un portail web dédié.

Le réseau local LAN est constitué de différents serveurs médicaux des différents services connectés en interne, en plus de la zone démilitarisée interne qui abrite le serveur de base de données médicale globale.

La DMZ externe contient le serveur du portail web qui a pour rôle de relier le réseau PAN au serveur de base de données médicale se trouvant au niveau de la DMZ interne, en plus d'autres serveurs accessibles de l'extérieur.

Par ailleurs, le personnel médical, selon le service auquel il appartient (service cardiologie, service traumatologie, service gastronomie, etc) et selon le rôle administré à chacun d'eux (médecin, infirmier, administration, etc) affichent ou modifient les données des patients rattachés à leurs services.

Les données de chaque patient sont envoyées au serveur de base de données médicale local au fur et à mesure qu'elles sont générées par les appareils intelligents de son PAN.

Les membres du personnel médical peuvent récupérer les données du patient enregistrées sur le serveur de base de données médicale protégé dans la DMZ interne selon les autorisations appropriées à chaque membre.

Afin de rendre cette architecture plus flexible tout en assurant un partage d'information efficace et rapide entre les patients et le personnel médical, nous intégrons le modèle de communication basé sur les agents mobiles afin de gérer les données des patients en temps quasi-réel et en toute sécurité. L'efficacité de la technologie des agents mobiles est due à son autonomie, sa capacité d'adaptation et sa capacité à interagir et coopérer avec d'autres agents.

Description du diagramme

Dans notre proposition, nous utilisons quatre types d'agents mobiles :

- **agent local** : agit en local dans la plateforme qui l'a créé. L'agent local lance l'agent léger et l'agent SOS, chacun d'eux a une mission à réaliser comme il est décrit ci-dessous. Ici, nous disposons d'un agent local sur la plateforme de l'appareil personnel du patient nommé "LA1" et un autre agent local se trouvant sur la plateforme du portail web nommé "LA2".
- **agent léger** :est capable de se déplacer très rapidement en raison de sa petite taille afin de collecter les informations nécessaires. L'agent léger sert d'agent intermédiaire entre l'agent local et l'agent lourd. Dans notre cas, nous déployons un agent léger au niveau de la plateforme de l'appareil personnel du patient nommé "LW1", un autre agent léger déployé au niveau de la plateforme du serveur de base de données médicale nommé "LW2".
- **agent lourd** : exécute une tâche qui demande parfois un lent traitement et des autorisations d'accès, comme la recherche dans la base de données des patients, qui contient toutes les informations nécessaires au sujet des patients. Ici, nous déployons un agent lourd au niveau de la plateforme du serveur de base de données médicale nommé "AgL".

- **agent SOS** : est un agent proposé par S.alami-kamouri et al dans [4]. Son rôle est de veiller à la sécurité de l'agent mobile léger, en surveillant ses déplacements lors de sa migration d'une plateforme à une autre en utilisant un temporisateur T. Si l'agent mobile remplit sa mission dans le délai T, il envoie un accusé de réception avec les résultats de sa mission chiffrés. Si le délai est dépassé, l'agent SOS demande à l'agent local de lancer

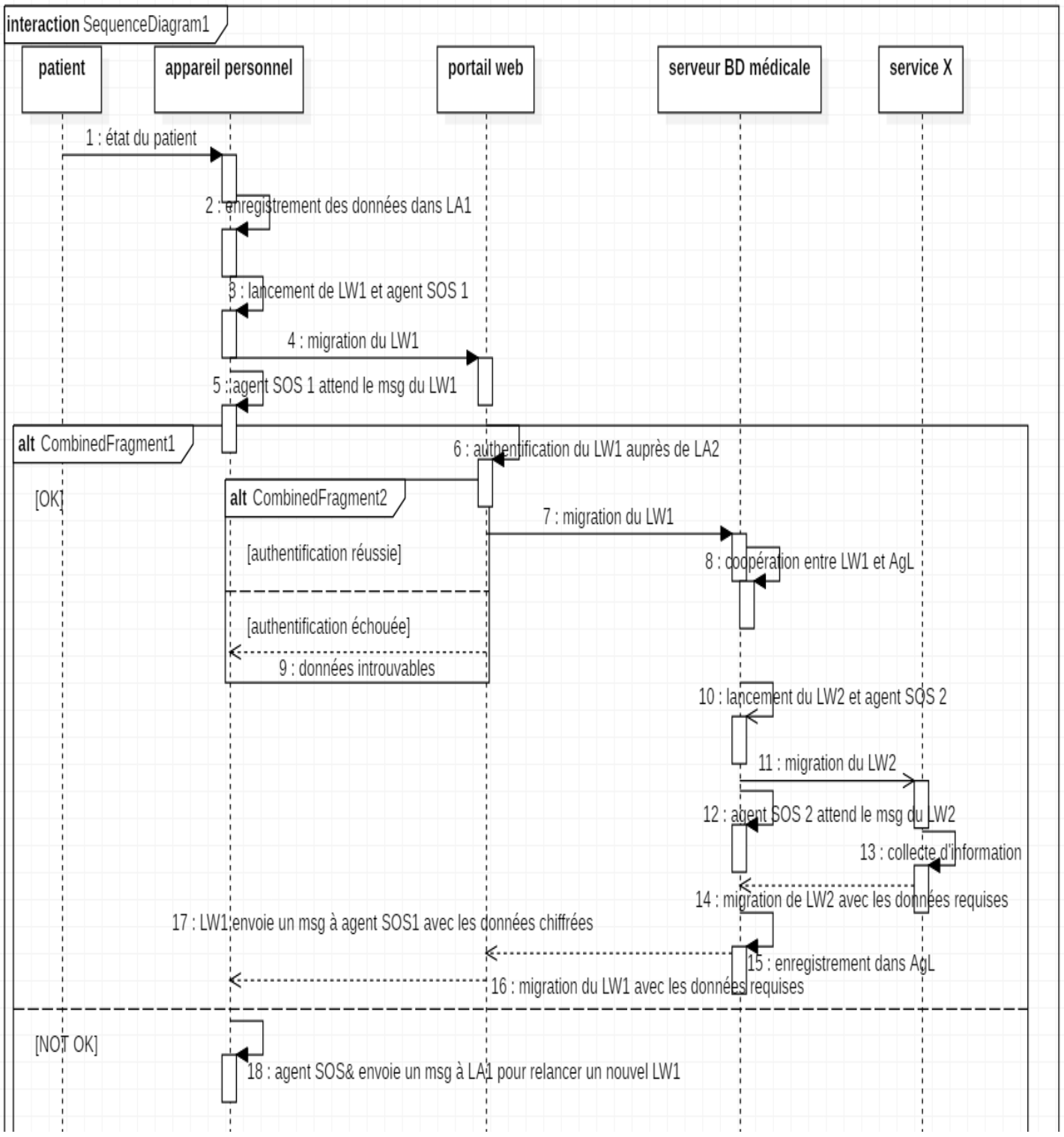

Fig. 4.9 – Diagramme de séquence pour la transmission des données à base d'agents mobiles

un nouvel agent léger et identifie cette plateforme comme étant malveillante et la dépose dans une liste noire. Dans notre cas, nous déployons un agent SOS sur la plateforme de l'appareil personnel du patient nommé "agent SOS 1" et un autre agent SOS déployé sur la plateforme du serveur de base de données médicale nommé "agent SOS 2".

Le diagramme (figure 4.9) explique le rôle du modèle d'agent mobile pour la transmission de données afin d'obtenir une réponse précise en temps quasi-réel. Cette transmission est établie entre un patient utilisant l'application des soins de santé et l'hôpital.

- Dans un premier lieu, les paramètres de santé du patient sont collectées par un appareil intelligent, dans notre cas une montre intelligente comme le montre la figure 4.8. Ensuite, ces données collectées par la montre intelligente sont transmises à l'appareil personnel du patient (ordinateur portable, téléphone, tablette).
- Une fois ces données sont transmises à l'appareil personnel du patient, elle sont enregistrées dans l'agent local "LA1". Puis, l'agent local crée un agent léger "LW1" et agent SOS "agent SOS1".
- L'agent local "LA1" attribue à l'agent léger "LW1" la mission à accomplir. Dès l'instant où LW1 migre vers le portail web, l'agent SOS "agent SOS1" lance un temporisateur T et attend le retour du "LW1" dans ce délai T.
- Lorsque le "LW1" arrive sur le portail web, cet agent léger va s'authentifier auprès de l'agent local "LA2" qui se trouve dans ce portail web.

Dans cette étape, on distingue entre deux cas : si "LA2" ne trouve pas l'identifiant du "LW1" dans sa liste, cela signifie que l'agent léger n'est pas autorisé à accéder au portail web, donc l'authentification a échoué. Sinon, dans le cas où l'authentification du "LW1" s'est bien déroulée :

- L'agent local "LA2" permet au "LW1" d'accéder au portail et de migrer vers le serveur de base de données médicale.
- Une fois l'agent "LW1" arrive au serveur de base de données, il entre en communication avec l'agent lourd "AgL" qui se trouve dans cette plateforme. Et en même temps, "LW1" envoie un accusé de réception à l'agent SOS 1.
- L'agent lourd récupère les informations reçues par le "LW1", les insère dans la base de données médicale. Puis, "AgL" crée un agent léger "LW2" et un agent SOS "agent SOS 2".
- Ensuite, l'agent lourd envoie le "LW2" au service approprié X avec la mission à accomplir. En même temps, "agent SOS 2" lance le temporisateur T afin de veiller sur le "LW2". Si le délai T est dépassé sans avoir de réponse du "LW2", l'agent SOS 2 envoie une alerte à "AgL" et identifie la plateforme d'accueil actuelle comme étant malveillante. L'agent lourd envoie alors un nouvel agent léger. Sinon :
- "LW2" migre vers le serveur de base de données médicale avec les informations collectées du service approprié.
- "LW2" envoie un accusé de réception à l'agent SOS 2 avec les informations collectées chiffrées. Ces informations sont enregistrées grâce à l'agent lourd sur le serveur de base de données médicale.
- Puis, "LW1" récupère les données requises de l'agent lourd et migre vers l'appareil personnel du patient en passant par le portail web.
- Une fois arrivé à la plateforme initiale, "LW1" envoie un accusé de réception à "agent SOS 1" avec les données chiffrées. Ces données sont traitées par l'agent local qui se trouve dans l'appareil personnel du patient.

Dans le cas où le délai T du "LW1" est dépassé, l'agent SOS 1 envoie une alerte à l'agent local "LA1" afin de déployer un nouvel agent léger. Même si l'agent SOS reçoit une réponse tardive de l'agent léger, elle sera refusée.

Ces types d'agents mobiles, et particulièrement l'agent SOS, permet d'assurer la disponibilité

du système qui est primordiale dans un système de soins de santé, surtout lorsque ce système assure le suivi des patients n'importe quand et n'importe où dans un temps quasi réel. Le chiffrement des données transmises par l'agent mobile permet d'assurer la confidentialité et la protection de la vie privée des utilisateurs de telles applications.

Nous avons déployé ces agents dans le réseau PAN et aussi dans le réseau local, car si nous prenons comme exemple l'attaque par déni de service qui vise la disponibilité du système, ce type d'attaque peut provenir de l'intérieur comme de l'extérieur. Ce qui s'avère coûteux pour les plateformes des soins de santé, car elles ont comme objectif de faire un suivi en temps réel pour les patients et spécialement pour les patients atteints de maladies chroniques ou en soins intensifs.

4.6 Conclusion

Les progrès des technologies informatiques et l'utilisation de l'IoT et des systèmes multi-agents dans le contexte de la santé intelligente ont amélioré l'accès à des informations précises sur les patients et ont permis de les diagnostiquer à distance et de surveiller rigoureusement leur état de santé. Cela présente des avantages considérables en terme d'accès aux soins mais pose également des défis complexes en terme de sécurité des données et de l'information.

Ce chapitre examine les recherches actuelles afin de fournir des informations sur les villes intelligentes et plus spécifiquement sur les soins de santé intelligents. Nous donnons également un aperçu de l'utilisation de l'Internet des objets et des systèmes multi-agents pour améliorer la cybersanté, la télémédecine et les services de santé intelligents.

Cependant, ces technologies et la mobilité offerte par les solutions sans fil présentent de nombreux défis de sécurité. C'est pourquoi, nous nous sommes également intéressés à l'étude des problèmes de sécurité auxquels est confronté le secteur des soins de santé intelligents, des menaces et des attaques auxquelles les soins de santé intelligents sont vulnérables et enfin des solutions existantes pour assurer la sécurité des données des patients. Ensuite, nous avons intégré notre solution proposée dans le chapitre 3 dans l'architecture de soins de santé typique proposée par les auteurs [109], afin d'assurer la disponibilité du système de soins de santé intelligents et la transmission des données en temps quasi-réel.

Conclusion générale

Au cours de ces dernières années, les technologies informatiques ont révolutionné le monde, ils ont changé la vie quotidienne des personnes de tout âge. Ces nouvelles technologies que se soit des appareils, des applications ont mis à la portée de main des utilisateurs des informations et des techniques utiles qui ont aidé à améliorer la vie du quotidien et la rendre meilleure et plus facile à gérer.

Afin de servir le nombre immense d'utilisateurs de ces nouvelles technologies et répondre à leurs besoins et exigences en temps réel, des protocoles de communications doivent être utilisés de manière à rendre la communication entre différentes entités à travers le réseau plus flexible et rapide. Le modèle à base d'agents mobiles est le plus adéquats pour ce type de technologie puisqu'il offre plus de flexibilité au système distribué, en plus de sa capacité de migrer pour traiter les informations localement et ne déplacer que l'information recherchée.

En plus d'apporter une facilité dans le quotidien des personnes, le partage de ces informations doit être sécurisé. Le problème primordial qui se pose dans ces nouvelles technologies et que la communication et les données qui circulent lors du transfert et du partage doivent rester confidentielles, la vie privée des utilisateurs est un enjeu important dans un monde numérique.

Notre thèse consiste à trouver des solutions afin de garantir une communication entre différentes entités à travers le réseau de manière flexible et en quasi-réel, en plus d'assurer la vie privée des utilisateurs. Notre étude tourne autour du modèle de communication agent mobile, afin de transmettre les informations entre différentes entités en temps quasi réel tout en diminuant la charge du réseau, sécuriser les systèmes à base d'agents mobiles. En plus de son intégration dans les systèmes de soins de santé intelligents.

Notre travail de recherche s'articule autour de quatre axes :

- le premier axe a pour but de montrer notre choix pour le modèle agents mobiles, en le comparant aux modèles de communication existant. Décrire ses avantages et ses inconvénients, en plus de son fonctionnement.
- le deuxième axe tourne autour de l'intégration de la technologie d'agent mobile dans le domaine de soins de santé intelligents, qui consiste en une proposition d'un modèle de service à base d'agents mobiles pour la transmission des données d'une ambulance intelligente.

- le troisième axe concerne l'aspect sécurité, en proposant deux mécanismes de sécurité pour les systèmes basés sur le modèle des agents mobiles.
- le quatrième axe s'articule sur l'association des agents mobiles et de l'internet des objets dans les environnements intelligents et spécialement dans les soins de santé intelligents, tout en évoquant les défis de sécurité. Nous proposons une amélioration d'architecture de soins de santé intelligents en intégrant la technologie d'agents mobiles et IoT afin de la rendre plus flexible et efficace.

Dans notre thèse, nous avons présenté les contributions suivantes :

- une amélioration apportée au service de l'ambulance intelligente, qui consiste au déploiement et la mise en place d'une solution intégrant le modèle des agents mobiles afin de diagnostiquer et communiquer l'état du patient une fois dans l'ambulance à l'hôpital afin de recevoir les recommandations du médecin avant d'arrivée à l'hôpital et en même temps préparer le service approprié et le personnel médical en temps quasi-réel. Nous avons proposé deux cas d'utilisation, le premier cas concerne les patients qui ont un antécédent médical et le deuxième cas concerne les patients sans antécédent médical. Nous avons aussi implémenté notre proposition pour montrer le fonctionnement du système d'agents mobiles utilisés et sa faisabilité.
- proposition d'un nouveau modèle de sécurité pour les systèmes à base d'agents mobiles. Comme l'aspect de sécurité reste un défi majeur pour la technologie d'agent mobile, nous avons proposé une approche bidimensionnelle basée sur deux mécanismes de sécurité :
  - la trace cryptographique : ce mécanisme est adopté pour garantir l'intégrité de l'agent mobile et l'authentification de l'origine afin de s'assurer que ce message est bien dédié a cette plateforme en plus de connaitre la plateforme émettrice.
  - le modèle d'agent SOS : ce mécanisme est proposé pour assurer la protection de l'agent lors de son déplacement contre les plateformes malveillantes, pour faire face aux attaques DOS et assurer la disponibilité du système.
- Nous nous sommes aussi focalisés sur l'association de la technologie d'agent mobile et l'internet des objets (IoT) dans les soins de santé intelligents et les défis de sécurité. Nous nous sommes intéressés à l'amélioration d'une architecture de soins de santé intelligents typique en intégrant la technologie d'agents mobiles afin d'assurer la disponibilité du système et la confidentialité des données lors du traitement et le partage des données.

Bibliographie

[1] S. Alami-kamouri, G. Orhanou and S. Elhajji, "Overview of mobile agent and security", International Conference on Engineering and MIS(ICEMIS 2016), 2016.

[2] S. Alami-kamouri, G. Orhanou and S. Elhajji, "Mobile Agent Service model for Smart Ambulance", Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST,2nd EAI International Conference on ICT Infrastructures and Services for Smart Cities (IISSC2017), Brindisi, Italy, April, 2017.

[3] S. Alami-kamouri, G. Orhanou and S. Elhajji, "Mobile Agent Paradigm Integration for New Service Models in Smart Ambulance", International Journal of Innovative Technology and Exploring Engineering (IJITEE), vol. 8(6), pp. 503-509, 2019.

[4] S. Alami-kamouri, N. Moukafih, G. Orhanou and S. Elhajji, "Mobile Agent Security based on Cryptographic Trace and SOS Agent mechanisms", Journal of Communications (JCM), vol. 15(3), pp. 221-230, 2020.

[5] S. Alami-kamouri, G. Orhanou and S. Elhajji, "Improvement of a smart healthcare architecture based on the integration of IoT and mobile agents", en cours.

[6] M. Espinilla, J. Liu and J.M. Garcia-Chamizo, "Recents Advancements in Ubiquitous Computing", Journal of Ambient Intelligence and Humanized Computing, vol.8(4), pp. 467-468, 2017.

[7] J. Lonchamp, "Introduction aux systèmes informatiques - Architectures, composants, mise en œuvre", https ://www.dunod.com/sites/default/files/atoms/files/9782100759446/Feuilletage.pdf, Dunod 2017, [Online]. [Accessed : February, 2020].

[8] Q. Larson, "A thorough introduction to distributed systems", https ://www.freecodecamp.org/news/a-thorough-introduction-to-distributed-systems-3b91562c9b3c/, April 2018, [Online]. [Accessed : February, 2020].

[9] M. A. Ibarhim, "Agents mobiles natifs pour systèmes embarqués", thèse de doctorat spécialité génie électrique, université de Sherbrooke, 2014.

[10] M. Loulou, "Approche formelle pour la spécification, la vérification et l'imposition des politiques de sécurité dynamiques dans les systèmes à base d'agents mobiles", thèse de doctorat en co-tutelle avec l'université de Bordeaux1, Université de Sfax, 2017.

[11] N. Ben Seghir and O. Kazar, "Une architecture basée agents mobiles pour la recherche d'information dans des sources hétérogènes et réparties", Proceedings of the 2nd Conférence Internationale sur l'Informatique et ses Applications (CIIA'09), January 2009.

[12] R.S. Gray, D. Kotz, G. Cybenko and D. Rus, "Mobile Agents : motivations and state of the art", Handbook of Agent Technology, 2002.

[13] H. Idrissi, "Contributions à la sécurité des systèmes d'agents mobiles", thèse de Doctorat, Université Mohammed V de Rabat, Faculté des sciences, Rabat, 2016.

[14] D.B. Lange, "Mobile Objects and Mobile Agents : The future of Distributed Computing ?", 12th European Conference on Object-Oriented Programming Brussels, Belgium, August 2002.

[15] H. Subramanian and P. Raj, "Hands-On RESTful API Design Patterns and Best Practices", Book Packt, January 2019.

[16] M. Hitchens, "Process Migration and Stability", Basser Dept. of Computer Science, University of Sydney, Australia.

[17] M. Flores-Badillo and E. Lopez-Mellado, "Workflow Management Based on Mobile Agent Technology", Encyclopedia of Artificial Intelligence, 2009.

[18] A. Patel, "Concept of Mobile Agent based electronic market place- Safety Measures", Encyclopedia of E-Business Development and Management in the Global Economy, 2010.

[19] N. Jailani, A. Patel, M. Mukhtar, S. Abdullah and Y. Yahya, "Concept of an Agent based Electronic Marketplace", Encyclopedia of E-Business Development and Management in the Global Economy, 2010.

[20] Z. Kotulski and A. Zwierko, "Security of Mobile Code", Handbook of Research on Wireless Security, 2008.

[21] A. Poggi and M. Tomaiuolo, "Concepts and Technologies", Handbook of Research on Mobility and Computing : Evolving Technologies and Ubiquitous Impacts, 2011.

[22] K. Karoui, "Interaction between Mobile Agents and Web Services", Encyclopedia of Multimedia Technology and Networking, Second Edition, 2009.

[23] B. Zhou, Q. Shi and M. Merabti, "A Novel Intrusion Detection System for Smart Space", Handbook of Research on Computational Forensics, Digital Crime and Investigation : Methods and Solutions, 2010.

[24] O. Urra, S. Ilarri, R. Trillo aand E. Mena, "Mobile Agents for a Mobile World", Handbook of Research on Innovations in Systems and Software Engineering, 2015.

[25] J. Ferber, "Les Systemes multi-agents : vers une intelligence collective", InterEditions, 1995.

[26] D. B. Lange and M. Oshima, "Seven good reasons for mobile agents", Communication. ACM, vol. 42, 88-89, 1999.

[27] Z. Maamar, "Apercu général sur la technologie des agents mobiles", RIST, vol. 8, n1, 1998.

[28] JDN, https ://www.journaldunet.fr/web-tech/dictionnaire-de-l-iot/1208123-interoperabilite-une-capacite-essentielle-pour-l-iot/, [Online]. [Accessed : February, 2020].

[29] D. Milojicic, M. Breugst, I. Busse, J. Campbell, S. Covaci, B. Friedman, K. Kosaka, D. Lange, K. Ono, M. Oshima, C. Tham, S. Virdhagriswaran and J. White, "MASIF : the OMG Mobile Agent System Interoperability Facility", Personal and Ubiquitous Computing 2(2), 117-129, 1998.

[30] Foundation for intelligent Physical Agents, FIPA Agent Management Support for Mobility Specification, document number dc00087c, Technical report, Geneva, Switzerland, May 2002.

[31] N. Islam, G.A. Mallah and Z.A. Shaikh, "FIPA and MASIF standards : a comparative study and strategies for integration", Proceedings of the 2010 National Software Engineering Conference NSEC'10, pp. 1-6, October 2010.

[32] https ://www.omg.org/about/index.htm, [Online]. [Accessed : February, 2020].

[33] https ://azure.microsoft.com/fr-fr/overview/what-is-middleware/, [Online]. [Accessed : February, 2020].

[34] D. Acremann, G. Moujeard and L. Rousset, "Développer avec CORBA en Java et C++", Edition Campus Press, 2000.

[35] https ://fr.slideshare.net/VasundharaGhose/mobile-agent-in-mobile-computing, [Online]. [Accessed : January, 2020].

[36] Foundation for Intelligent Physical Agents. FIPA Specification : Human Agent Interaction, 1998, http ://www.fipa.org/, [Online]. [Accessed : December, 2019].

[37] S. Poslad and P. Charlton, "Standardizing Agent Interoperability : the FIPA Approach", Book Multi-Agent Systems and Applications : 9th ECCAI Advanced Course, ACAI 2001 and Agent Link's 3rd European Agent Systems Summer School, pp. 98-117, July 2001.

[38] Foundation for Intelligent Physical Agents. FIPA 97 Specification part 1 : Agent Management, October 1997, http ://www.fipa.org/, [Online]. [Accessed : December, 2019].

[39] Foundation for Intelligent Physical Agents. FIPA 97 Specification part 2 : Agent Communication Language, November 1997, http ://www.fipa.org/, [Online]. [Accessed : December, 2019].

[40] Foundation for Intelligent Physical Agents. FIPA Specification, FIPA 2000 and beyond, 2000, http ://www.fipa.org/, [Online]. [Accessed : December, 2019].

[41] O. Boissier, Cours SMA-DEA-CCSA Multi-agent systems, MAS platforms, SMA/SIMMO, EMNS Ecole des Mines de Saint Etienne, 2001.

[42] B. Benmammar, Z. Jrad, F. Krief and N. Mbarek, "Dynamique de l'environnement : Scénarios, simulations et maquette", HAL Id : hal-00659969, 2012.

[43] M. Zhang, A. Karmouch and R. Impey, "Towarads a secure agent platform based on FIPA", In Proceedings of the third International Workshop on Mobile Agents for Telecommunication Applications (MATA 01), vol. 2164 of LNCS, pp. 277-289, Montreal, Canada, 2001.

[44] C. Cubat Dit Cros, "Agents mobiles coopérants, pour les environnements dynamiques", thèse de doctorat, Institut National Polytechnique de Toulouse, Decembre 2005.

[45] JAva DEvelopment framework (JADE), https ://jade.tilab.com/, [Online]. [Accessed : September, 2019].

[46] S. El Falou, "Programmation répartie, optimisation par agent mobile", thèse de doctorat, Université CAEN/ Basse-Normandie, Novembre 2006.

[47] B. Espinasse, "Communication et langage de communication dans les SMA", https ://pageperso.lis-lab.fr/bernard.espinasse/Supports/SMA/SMA-BE4-2012-4p.pdf, [Online]. [Accessed on september 2019].

[48] D.S. Milojicicc, W. LaForge and D. Chauhan, "Mobile Objects and Agents (MOA)", Proceedings of the fourth Conference on Object-Oriented Technologies and Systems (COOTS), 1998.

[49] N. Singhal, A. Dixit, R.P. Agarwal and A.K. Sharma, "A study of Mobile Agent Platforms for Distributed Web Crawling", International Journal of Advances Engineering Science and Technology (IJAEST), vol. 1(2), pp. 111-121, 2018.

[50] A.Chavez and P. Maes, "Kasbah : an agent marketplace for buying and selling goods", Proceedings of the First International Conference on the Practical Application of Intelligent Agents and multi-agents technology, London, UK, pp. 75-90, April 1996.

[51] L. Che and X.P. Yang, "Research and Application of Mobile Agent in E-commerce System", Applied Mechanics and Materials, vol. 519(520), pp. 458-461, 2014.

[52] F. Bergenti, A. Poggi and M. Tomaiuolo, "Multi-Agent Systems for E-health and Telemedicine", Encyclopedia of E-health and Telemedicine, 2016.

[53] C.J. Su and T.W. Chu, "A mobile Multi-agent information system for ubiquitous fetal monitoring", International Journal of Environmental Research and Public Health, vol. 11(1), pp. 600-625, 2014.

[54] W.S. Hsu and J.I. Pan, "Secure Mobile Agent for Telemedecine Based on P2P Networks", Journal of Medical System, 2013.

[55] https ://cordis.europa.eu/project/id/644329/fr, [Online]. [Accessed : February, 2020].

[56] G. Beri, P. Ganjare ; A. Gate, A. Channawar and V. Gaikwad, "Intelligent Ambulance traffic control", Special Issue on International Journal of Electrical, Electonics and Computer Systems, for the 3rd National Conference on Advancements in Communication, Computing and Electonics Technology (ACCET), vol. 2, pp. 74-80, 2016.

[57] S. Hignett, A. Jonesand J. Benger, "Portable and mobile clinical pods to suuport the delivery of community based urgent care", International Conference on Inclusive Design, London, UK, 2015.

[58] B. Lopez, B. Innocenti and D. Busquets, "A Multi-agent System for Coordinating Ambulances for Emergency Medical Services", IEEE Intelligent System, vol. 23(5), pp. 50-57, October 2008.

[59] R. Trillo, S. Ilarri and E. Mena, "Comparison and performance evaluation of mobile agent platforms", Third International Conference on Autonomic and Autonomous Systems (ICAS'07), June 2007.

[60] Tromso and Cornell, "Moving Agents", http ://www.tacoma.cs.uit.no/index.html.[Online]. [Accessed : January, 2020].

[61] D.B. Lange and M. Oshima, "Mobile agents with Java : The Aglet API", World Wide Web Internet and Web Information Systems, vol. 1(3), pp. 111-121, September 1998.

[62] F.L. Bellifimine, G. Caire, D. Greenwood, "Developping Multi-Agent Systems with JADE", Journal Developing Multi-agent systems with JADE, pp.1-286, 2007.

[63] D.Schoder, T.Eymann, "Technical opinion : The real challenges of mobile agents", communications of the ACM, vol. 43(6), pp. 111-112, June 2000.

[64] ISO 7498-2, Systèmes de traitement de l'information - Interconnexion de systèmes ouverts - Modèle de références de base, partie 2 : Architecture de sécurité, 1989.

[65] K. Richards and S. Shea, https ://searchsecurity.techtarget.com/definition/access-control ?amp=1, [Online]. [Accessed : January, 2020].

[66] M. Haughn and S. Gibilisco, "Confidentiality, integrity and availability (CIA triad)", https ://www.google.com/amp/s/whatis.techtarget.com/definition/confidentiality-integrity-and-availability-CIA3famp=1, [Online]. [Accessed : January, 2020].

[67] J. Breithaupt and M. S. Merkov, "Information Security Principles of Success", book Information Security : Principles and Practices, 2nd Edition, July 2014.

[68] E. Wheeler, "Security Controls and Services", in book Security Risk Management, pp. 127-146, 2011.

[69] Strasbourg Academy, https ://ssi.ac-strasbourg.fr/bonnes-pratiques/recommandations/lidentification-et-lauthentification/ [Accessed : December 2019]

[70] P. Dadhich, K. Dutta, M.C. Govil, "Security issues in mobile agents", International Journal of Computer Applications, vol. 11(4), pp. 1-7, 2010.

[71] N.Karnik, "Security in Mobile Agent systems", PhD. Dissertation, Department of Computer Science, University of Minnesota, October 1998.

[72] A. Henderson, "The CIA Triad : Confidentiality, Integrity, Availability", http ://panmore.com/the-cia-triad-confidentiality-integrity-availability, [Online]. [Accessed : January, 2020].

[73] A.M. Ngereki, A.M. Kahonge, "A multi faceted approach to mobile agent security", International Journal of Computer Applications, vol. 120(21), pp. 20-26, 2015.

[74] B. Amro, "Mobile Agent Systems, Recent Security threats and Counter Measures", International Journal of Computer Science Issues, vol. 11(2), pp. 146-151, 2014.

[75] D.M. Chess, "Security Issues in Mobile Code Systems", Lecture Notes in Computer Science LNCS book series, vol. 1419, pp. 1-14, June 1998.

[76] M.A. Madkour, F.E. Eassa, A.M. Ali and N.U. Qayyum, "Securing Mobile -Agent- Based Systems Against Malicious Hosts", World Applied Sciences Journal, vol. 29(2), pp. 287-297, 2014.

[77] T.M. Ahmed, "Using Secure-Image Mechanism to Protect Mobile Agent against malicious hosts", International Scholarly and Scientific Research and Innovation, vol. 3(11), pp. 364-369, 2009.

[78] M. Hafeeda and B. Bhargava, "On Mobile Code Security", Center of Education and Research in Information Assurance and Security, 2001.

[79] F. Hohl, "Time Limited Blackbox Security : protecting mobile agent from malicious hosts", Mobile Agents and Security, Lecture Notes in Computer Science 1419, Springer-Verlag, pp. 92-113, Berlin, 1998.

[80] H.K. Tan and L.Moreau, "Extending execution tracing for mobile code security", Second International Workshop on Security of Mobile Muti-agent Systems, Italy, 2002.

[81] S. Hanaoui, J. Laassiri and Y. Bergui, "Security requirements and model for mobile agent authentication", Smart network inspired paradigm approaches in IoT applications, pp. 179-189, July 2019.

[82] M. Popa, "Binary Code Disassembly for Reverse Engineering", Journal of Mobile, Embedded and Distributed Systems (JMEDS), vol. 4(4), pp. 233-248, 2012.

[83] G. Vigna, "Cryptographic traces for mobile agents", proceeding Mobile Agents and Security, pp. 137-153, Springer-Verlag, London, UK, 1998.

[84] F.Bellifemine, G. Caire and D. Greenwood, "Developing multiagent systems with JADE", Chichester, England Hoboken, NJ : John Wiley, 2007.

[85] https ://www.techopedia.com/definition/31494/smart-city, "What is a Smart City ?", [Online], [Accessed : August, 2019].

[86] W.D. Eggers and J. Skowron, "Forces of change : Smart cities", part of Deloitte series on Smart cities, Copyright 2018 Deloitte Development LLC, [Online], [Accessed : August, 2019].

[87] M. Choudhary, "Technology - The backbone of a smart cities", May 2018, [Online]. Available : https ://www.geospatialworld.net/article/technlogythe-backbone-of-a-smart-city/. [Accessed : August, 2019].

[88] P.A. Prez-Martnez, A. Martnez-Ballest and A. Solanas, "Privacy in Smart Cities A case Study of Smart Public Parking", in Proceedings of the 3rd International Conference on Pervasive Embedded Computing and Communication Systems, PECCS 2013, pp. 55-59, January 2013.

[89] O. Krakovetskyi, "How to make a smart cities smart ?", in devrain intelligent solutions and services, March 2017. [Online]. Available : https ://devrain.com/posts/how-to-make-cities-smart. [Accessed : August, 2019].

[90] P. Sundaravadivel, E. Kougianos, S.P Mohanty and M.K. Ganapathiraju, "Everything You Wanted to Know about Smart Health Care : Evaluating the Different Technologies and Components of the Internet of Things for Better Health", in IEEE Consumer Electronics Magazine , vol. 7(1), pp. 18-28, 2018.

[91] J.S. Jeong, O. Han and Y.Y. You, "A Design Characteristics of Smart Healthcare System as the IoT Application", Indian Journal of Science and Technology, vol. 9(37), pp. 37-44, 2016.

[92] S.Sivagami, D. Revathy and L. Nithyabharathi, "Smart health care system implemented using IoT", International Journal of Contemporary Research in Computer Science and Technology (IJCRCST), vol.2(3), pp. 641-646, 2016.

[93] C. Kakoko Lubamba, "Internet of Things for Cyberhealthcare (IoT4C) : Information Dissemination, Systems Interoperability and Security", in Workshop on Open Source Solutions for the Internet of Things, July 2017.

[94] F. Bergenti, A. Poggi and M. Tomaiuolo, "Multi-Agent Systems for E-Health and Telemedicine", in Encyclopedia of E-Health and Telemedicine, Hershey : PA, IGI Global, pp. 688-699, 2016.

[95] C.J. Su and T.W. Chu, "A Mobile multi-agent information system for ubiquitous foetal monitoring", International Journal of Environmental Research and Public Health, vol. 11(1), pp. 600-625, 2014.

[96] D. Pauli, "Thousands of directly hackable hospital devices exposed online", September 2015. [Online]. Available : https ://www.webcitation.org/6cTqrpe1w. [Accessed : August, 2019].

[97] K. Michael, "Security and Privacy Issues with IoT in Healthcare", October 2016. [Online]. Available : http ://www.pcworld.com/article/3132571/hackerscreate-more-iotbotnets-with-mirai-source-code.html, 18 [Accessed : August, 2019]

[98] A. Chacko and T. Hayajneh, "Security and Privacy Issues with IoT in Healthcare", EAI Endorsed Transactions on Pervasive Health and Technology, vol. 4, July 2018.

[99] C. Boyer, "ETSI releases first globally applicable standard forconsumer IoT security", February 2019. [Online]. Available : https ://www.etsi.org/newsroom/press-releases/1549-2019-02-etsireleases-first-globally-applicable-standard-for-consumer-iot-security, [Accessed : September, 2019].

[100] Gemalto a Thales Company, "Biometrics : authentication and Identification (definition, trends, use cases, laws and latest news)- 2020 review", 2020 https ://www.gemalto.com/govt/inspired/biometrics, [Online]. [Accessed : January, 2020].

[101] S. Hussain Talpur, Z. Alam Bhuiyan and G. Wang, "Shared-node IoT network architecture with ubiquitous homomorphic encryption for healthcare monitoring", International Journal of Embedded Systems, vol. 7(1), pp. 43-54, 2015

[102] X. Yue, H. Wang, D. Jin, M. Li and W. Jiang, "Healthcare Data Gateways : Found Healthcare Intelligence on Blockchain with Novel Privacy Risk Control", Journal of Medical Systems, vol. 40(10), pp.1-8, 2016.

[103] D.J. Cook, G. Duncan, G. Sprint and R.L. Fritz, "Using smart city technology to make healthcare smarter", in Proceedings of the IEEE, vol. 106(4), pp. 1-15, January 2018.

[104] T. Bayer and C. Reich, "Security of Mobile Agents in Distributed Java Agent Development Framework (JADE) platforms", The Twelfth International Conference on Systems, held in Venice, April 2017.

[105] N. Bouchemal and R.Maamri, "CAPMA : Clone agent to protect mobile agents in dynamic environments", International Conference on Advanced Aspects of Software Engineering (ICAASE), 2016.

[106] R. H. chowhan and P. Dayya, "Itinerary and mobile code patterns for emerging mobile agent systems in large scale distributed environments", International Journal of Computer Sciences and Engineering, vol 5(6), 2018.

[107] R. H. chowhan, "Mobile agent programming paradigm and its application scenarios", International Journal of Current Microbiology and Applied Sciences, vol. 7(5), 2018.

[108] https ://www.govinfo.gov/content/pkg/PLAW-113publ283/pdf/PLAW-113publ283.pdf, [Online]. [Accessed : March, 2020].

[109] S. Ahmed and A. Rajput, "Threats to Patients' Privacy in Smart Healthcare Environment", Innovation in Health Informatics A Smart healthcare Primer, vol. 16(1), pp. 375-394, 2019.

[110] https ://csrc.nist.gov/topics/laws-and-regulations/laws/fisma, [Online]. [Accessed : March, 2020].