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

---

Université Kasdi Merbah – Ouargla
Faculté des Nouvelles Technologies de l'Information et de la Communication
Département d'Informatique et des Technologies de l'Information

Mémoire

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

Option : Technologie de l'Information et de Communication

Une approche agent mobile pour la sécurité de système industriel distribué

Par

OUENDJEN Mohamed El Fateh

Devant le jury :

Dr. Laalam Fatima Zohra,	Maître de conférences A	Université de Ouargla	Président
Dr. KAZAR Okba,	Professeur	Université de Biskra	Rapporteur
Dr. Bennoui Hammadi,	Maître de conférences A	Université de Biskra	Examinateur
Dr. Terissa Labib Sadek,	Maître de conférences A	Université de Biskra	Examinateur

À mes très chers parents…

À mon épouse et mes fils Anes et Lokmane…

À mes frères et mes sœurs…

Je dédie ce travail.

Remerciement

Je tiens à remercier tout d'abord, le Professeur Mr. KAZAR Okba, de m'avoir proposé un tel intéressant sujet, pour son encadrement avec patience, ses précieux conseils, sa disponibilité et son soutien tout au long de ce travail.

Je tiens à remercie également les membres du jury d'avoir accepté d'évaluer mon travail.

Mes remerciements à tous ceux qui m'ont aidé de près ou de loin…

Résumé

L'accroissement de l'interconnexion des systèmes informatiques et l'évolution des réseaux à grande échelle, les a rendus vulnérables aux différents types d'attaques. Sécuriser ces systèmes est devenu un enjeu essentiel ; une faille de sécurité dans un tel système peut causer de sérieux dommages. La sécurité des systèmes et réseaux peut se faire via deux approches ; Une approche préventive, qui consiste à protéger les données et les ressources contre des attaques potentielles. Cependant, il est impossible d'avoir un système complètement sûr. Pour pallier ce problème, nous nous focalisons, dans le cadre de ce mémoire, sur la deuxième approche de sécurité qui est l'approche de détection. Cette dernière consiste à contrôler en permanence le comportement du système, en travaillant en arrière-plan, pour qu'elle puisse détecter les attaques le plutôt possible afin de réagir rapidement et éviter de graves dégâts. Cette approche représente un mécanisme particulier de gestion de sécurité qui est la détection d'intrusions, qui a prouvé être un mécanisme efficace pour détecter les attaques. La technologie d'agents mobiles s'impose comme un concept efficace et bien adapté pour répondre aux nouveaux besoins de gestion de sécurité et de faire une détection d'intrusion dans les systèmes fortement distribués, grâce à leurs propriétés de réduire la charge dans le réseau, d'exécuter d'une façon asynchrone et autonome, où ces agents peuvent exécuter un traitement local sur les données collectés. Ces propriétés justifient notre choix d'utiliser le concept d'agents mobiles pour la gestion de la sécurité des systèmes industriels distribués par une approche de détection d'intrusion (AM-SID).

Mots clés : système multi-agents, agent mobile, sécurité, détection d'intrusion, système distribué, système industriel.

Table des matières

Liste des figures

Introduction générale

Les systèmes informatiques sont aujourd'hui des outils essentiels pour le bon fonctionnement de la majorité des entreprises. Ils sont utilisés dans différents secteurs comme la banque, le commerce, la médecine et encore le domaine de l'industrie. L'évolution des réseaux à grande échelle et l'accroissement de l'interconnexion de ces systèmes, les a rendus par conséquent, vulnérables aux différents types d'attaques. Sécuriser ces systèmes est devenu un enjeu essentiel ; une faille de sécurité dans un système peut causer de sérieux dommages.

La sécurité des systèmes et réseaux peut se faire via une approche préventive, qui consiste à protéger les données et les ressources contre des attaques potentielles. Par exemple, l'authentification par mot de passe, le cryptage des données ou l'utilisation du pare-feu. Cependant, protéger contre tous les types attaques, semble quelque peu irréel. En fait, il est impossible d'avoir un système complètement sûr. Pour pallier ce problème, nous nous focalisons, dans le cadre de ce mémoire, sur la deuxième approche de sécurité qui est « l'approche de détection ». Cette dernière consiste à contrôler en permanence le comportement du système ou réseau, en travaillant en arrière-plan, pour qu'elle puisse détecter les attaques le plutôt possible afin de réagir rapidement et éviter de graves dégâts. Cette approche représente un mécanisme particulier de gestion de sécurité qui est la détection d'intrusions, qui a prouvé être un mécanisme efficace pour détecter les attaques.

Les systèmes de détection d'intrusions classiques ou centralisés, dans laquelle le traitement des données collectés se fait au niveau d'une unité centrale, ne peuvent pas détecter les attaques actuelles, qui sont devenues beaucoup plus complexes et sophistiqués, particulièrement les attaques distribués. Ces systèmes souffrent ainsi, de problème de gestion des grandes masses d'information et de traitement du trafic dans les réseaux, due à la complexité croissante de ceux-ci.

La technologie d'agents mobiles s'impose comme un concept efficace et bien adapté pour répondre aux nouveaux besoins de gestion de sécurité et de faire une détection d'intrusion, grâce à leurs propriétés de réduire la charge dans le réseau, d'exécuter d'une façon asynchrone et autonome, où ces agents peuvent exécuter un traitement local sur les données collectés. Ces propriétés justifient notre choix d'utiliser le concept d'agents mobiles pour la gestion de la sécurité des systèmes industriels distribués par une approche de détection d'intrusion.

Un agent mobile est une entité autonome ayant une propriété mobile la permettant de se déplacer avec son code, d'un hôte à l'autre afin d'effectuer des traitements. Cet agent s'adapte aux changements de l'environnement, communique et coopère avec d'autres agents. C'est un paradigme alternatif à l'approche client-serveur et de plus en plus utilisé dans les systèmes distribués.

Notre mémoire se compose de quatre chapitres organisés comme suite :

Le premier chapitre présente les systèmes industriels distribués ainsi que les problèmes et les défis de ce domaine.

Le second chapitre est consacré à la présentation de la technologie des agents mobiles et leurs problèmes de sécurité. Nous décrivons en premier lieu, la notion d'agent, Systèmes Multi-agents et agents mobiles, ensuite nous détaillons l'aspect sécuritaire de cette technologie. Nous achevons ce chapitre par mentionner quelques travaux ayants traité les systèmes de détection d'intrusion à base d'agents mobiles.

Au niveau du troisième chapitre, nous exposons la conception de notre approche de détection d'intrusion basée agents mobiles pour un système industriel distribué.

Dans le quatrième chapitre nous allons étudier et implémenter un cas bien précis, sur lequel sont projetés les principaux aspects de notre approche de sécurité.

Nous terminons ce mémoire par une conclusion générale suivi par des perspectives suggérées.

Chapitre I. Les systèmes industriels distribués

I.1 Introduction

Les activités industrielles (Vrignat, 2009) font presque quotidiennement les grands titres des actualités avec leurs conséquences d'incidents, d'accidents ou d'événements catastrophiques. En effet, le zéro défaut ou le risque zéro n'existe malheureusement pas pour les activités industrielles à cause de l'occurrence de défaillances humaines ou matérielles. Pour réduire les risques à un niveau le plus faible possible et acceptable, des méthodes, des techniques et des outils scientifiques ont été développés dès le début du 20ème siècle pour évaluer les risques potentiels, prévoir l'occurrence des défaillances et tenter de minimiser les conséquences des situations catastrophiques, lorsqu'elles se produisent. Dans la plupart des systèmes industriels, une demande croissante est apparue en matière de remplacement des politiques de sécurité et de maintenance curative par des mécanismes de sécurité préventive. Cette mutation nécessite des moyens technologiques ainsi que la connaissance de techniques d'analyse appropriées.

Ce chapitre est organisé comme suit : en premier nous allons décrire d'une manière générale les systèmes industriels distribués ainsi que leur évolution, par la suite on va parler sur le mécanisme de pilotage (ou de conduite) de ces systèmes, et après nous allons voir comment surveiller et contrôler les activités industriels par l'utilisation des systèmes informatiques. Ensuite on passe à l'automatisation de ces systèmes grâce aux automates programmables industriels (API) et en fin nous terminerons ce chapitre par une conclusion.

I.2 Systèmes de production

I.2.1 Définition

Un système de production (comme présenté dans la Figure I.1) est un système à caractère industriel qui regroupe des éléments matériels ou immatériels (machines, ressources humaines, énergie, logiciels, techniques, …) conçu pour transformer, sous certaines conditions, un flux de matières d'œuvre (matière premières) en un autre flux de produits finis (biens ou services) de valeur augmenté.

L'objectif principal de n'importe quel système de production est d'exposer sur le marché le produit le plus compétitif possible pour l'entreprise.

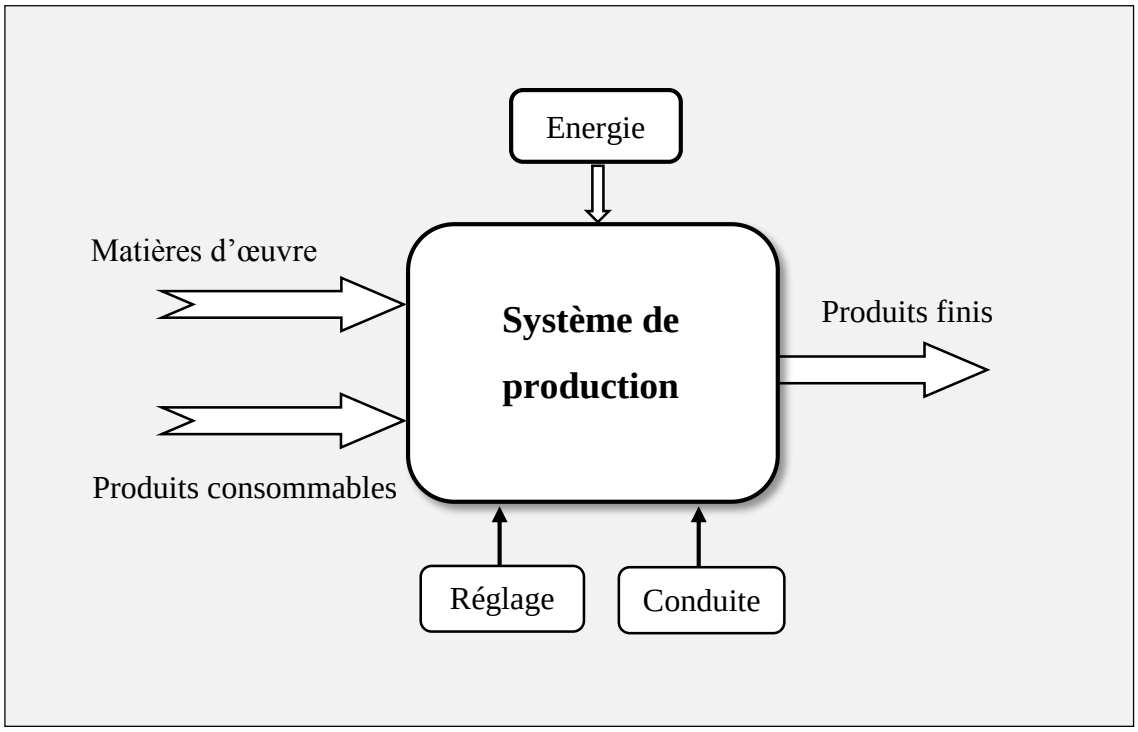

Figure I.1 - Système de production

Nous définissons ci-dessous quelques concepts essentiels liés aux systèmes de productions industriels :

- Les matières d'œuvres : l'objectif du système est de transformer les matières d'œuvre (état initial) en produits finis (état final). Ces dernières se présentent sous plusieurs formes, par exemple des matières premières, de l'énergie, des informations, …etc.

- Le procédé : c'est la partie opératoire qui exécute les processus industriels. d'une autre manière c'est la partie qui transforme les matières d'œuvres en produits finis.

- La conduite (ou pilotage) **:** le rôle de la conduite dans un système de production est de piloter, guider et élaborer les ordres nécessaires au procédé pour satisfaire les objectifs de la production.

I.2.2 Evolution des systèmes de production

Les différentes entreprises de productions actuelles sont le produit de l'évolution et des changements qu'ont connus les systèmes de productions tout au long des siècles passés. Partant de la production artisanale et arrivant aux systèmes flexibles. Les systèmes de production ont connu plusieurs développements favorisés par des évolutions techniques, technologiques et économiques. Une liste non exhaustive de ces évolutions comprend (Soltani, 2007):

- La révolution industrielle qui a permis l'utilisation des différents types d'énergie (vapeur, charbon, électricité, pétrole) et l'apparition des machines industrielles ;

- La révolution scientifique concrétisée par la propagation de l'outil informatique, et son introduction dans tous les domaines (communication, fabrication, formation, gestion …etc.) ;

- La mutation du contexte économique d'une économie ordinaire vers une économie de marché. Ceci à imposer les entreprises industrielles de passer d'une production locale caractérisée par des séries de grande quantité de produits standard à une production internationale caractérisée par des petites et moyennes séries de produits personnalisés.

I.3 L'informatique industrielle

L'informatique industrielle, est comme son nom l'indique, concerne l'utilisation de l'outil informatique pour la fabrication de produits industriels, du bureau d'études (conception assistée par ordinateur) à leur production (fabrication assistée par ordinateur, automatique, robotique) en passant par la logistique, la gestion des stocks, …etc. Elle regroupe les programmes dont les variables représentent des grandeurs physiques comme, la température d'une cuve, l'état d'un capteur ou la position d'un bras robotique.

L'Informatique Industrielle (Calvet, 2011) établit des concepts, spécifie des modèles, élabore des méthodes, développe des outils en vue de la conception et de la réalisation matérielle et logicielle de systèmes informatisés de commande.

En résumé, on peut dire que dans une entreprise industrielle, l'informatique représente son système nerveux.

I.3.1 Domaines d'application

L'informatique industrielle est aujourd'hui au cœur de tous les secteurs industriels grâce à sa capacité d'accélérer, d'automatiser, de conduire et de contrôler les activités industriels. Nous citons quelques domaines d'application de cette discipline :

- L'industrie automobile par l'utilisation de robots industriels pour effectuer l'assemblage et la peinture des carrosseries.
- Robotique.
- Télécommunications et réseaux.
- Météo (mesures et acquisition des données météorologiques, ...).
- Militaire (radars, systèmes de guidage de missiles, ...).
- Paiement pour carte bancaire.

I.4 La conduite des systèmes industriels

Généralement les systèmes industriels sont des systèmes complexes et distribués. Le pilotage ou la conduite de ces systèmes est une activité difficile à réaliser et dépasse les compétences des humains. Nous entendons le pilotage d'un système industriel l'ensemble des tâches suivantes : l'installation du système, la configuration, la commande, la surveillance et la supervision.

I.4.1 La commande

La commande d'un système industriel consiste (Combacau, 2000) à exécuter un ensemble d'opérations au système en fixant des consignes de fonctionnement en réponse à des ordres d'exécution (Figure I.2). Il peut s'agir de réaliser :

- une séquence d'opérations constituant une gamme de fabrication dans le but de fabriquer un produit en réponse à une demande d'un client,
- une séquence d'actions corrective destinée à rendre au système de production toute ou partie des fonctionnalités requises pour assurer sa mission. La perte d'une partie des fonctionnalités initialement disponibles faisant suite à l'occurrence d'une défaillance,
- des actions prioritaires et souvent prédéfinies sur le procédé dans le but d'assurer la sécurité de l'installation et du personnel.
- des opérations de test, de réglage, de nettoyage permettant de garantir que le système de production pourra continuer d'assurer sa mission.

La commande regroupe toutes les fonctions qui agissent directement sur les actionneurs du procédé. On y retrouve naturellement ce que l'on a coutume d'appeler :

- le fonctionnement en l'absence de défaillance,
- reprise ou gestion des modes,
- les traitements d'urgence,
- une partie de la maintenance corrective.

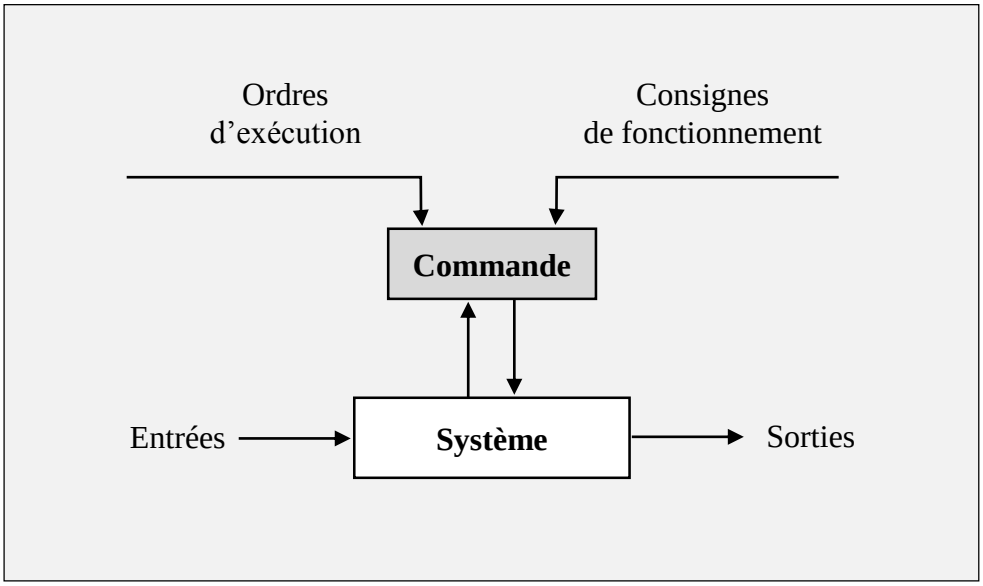

Figure I.2 - Commande d'un système industriel

I.4.2 La surveillance

La surveillance est une activité en permanence dont le but est de collecter en temps réel toutes les informations du système industriel, d'indiquer les anomalies de son comportement, de reconstituer son état réel et faire toutes les inférences nécessaires pour produire les données utilisées :

- pour dresser des historiques de fonctionnement,
- le cas échéant, pour mettre en œuvre un processus de traitement de défaillance.

La surveillance a donc un rôle passif vis-à-vis du système de commande et du procédé, parce que son rôle est limitée aux fonctions qui collectent des informations, les archivent, font des inférences, etc. sans agir réellement ni sur le procédé ni sur la commande.

I.4.3 La supervision

La supervision consiste (Combacau, 2000) à contrôler et surveiller l'exécution d'une opération ou d'un travail effectué par d'autres sans rentrer dans les détails de cette exécution. La supervision détermine les modes de fonctionnement normal et anormal du système supervisé, à partir d'un modèle et des évènements qu'il reçut (Figure I.3).

- En fonctionnement normal, son rôle est surtout de prendre en temps réel les dernières décisions correspondant aux degrés de liberté exigés par la flexibilité décisionnelle. Pour cela elle est amenée à faire de l'ordonnancement temps réel, de l'optimisation, à modifier en ligne la commande et à gérer le passage d'un algorithme de surveillance à l'autre.

- En présence de défaillance, la supervision va prendre toutes les décisions nécessaires pour le retour vers un fonctionnement normal. Après avoir déterminé un nouveau fonctionnement, Il peut s'agir de choisir une solution curative, d'effectuer des ré-ordonnancements locaux, de prendre en compte la stratégie de surveillance de l'entreprise, de déclencher des procédures d'urgence, etc.

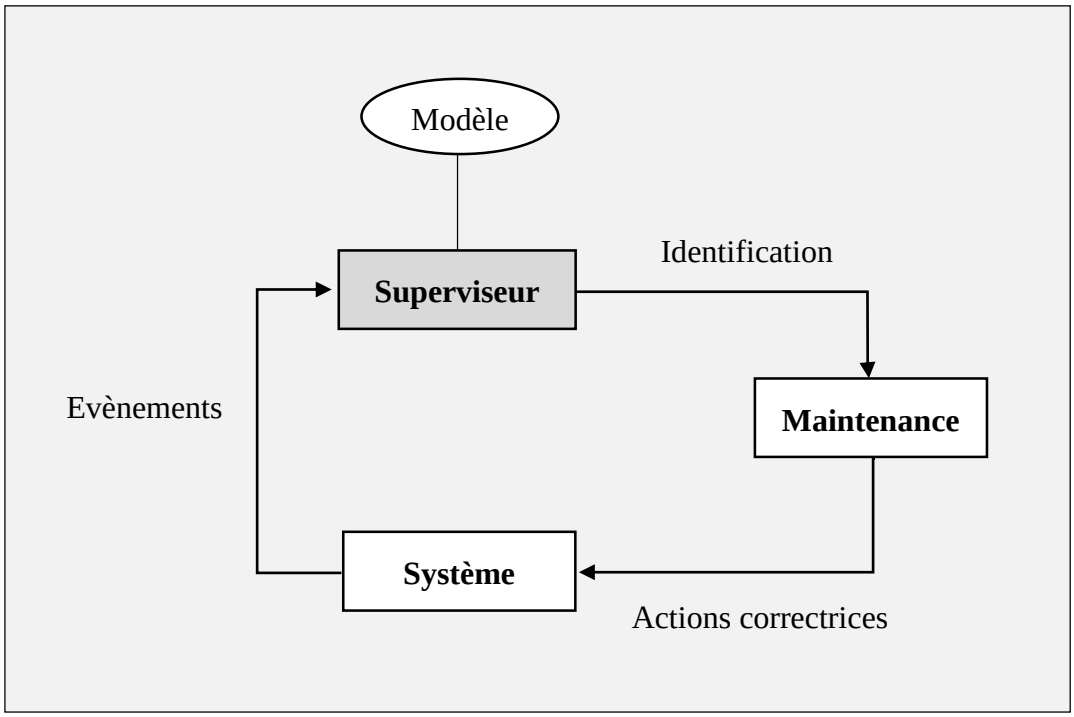

Figure I.3 - Système industriel supervisé

I.5 Système de contrôle industriel

Les systèmes de contrôle industriels permettent de surveiller, de contrôler et d'assurer le bon fonctionnement des processus industriels. Ces systèmes sont utilisés pour soutenir et automatiser plusieurs secteurs dans l'industrie, notamment les secteurs de l'énergie (comme la distribution de gaz et d'électricité, le raffinage du pétrole), le transport (réseau ferroviaire), traitement de l'eau, la fabrication, la santé, …etc. En effet, un système industriel ne peut exister longtemps sans contrôle et sans surveillance.

De manière générale, il est considéré que le contrôle de processus industriel intègre quatre grands champs d'application et de recherche (Babus, 2008):

- L'identification de l'ensemble des variables permettant de détecter le plus rapidement et le plus facilement les fautes intervenues au cours du fonctionnement d'un processus.
- la détection des fautes, qui consiste à mettre clairement en évidence l'apparition de la faute à l'aide de l'ensemble de variables retenues à l'étape d'identification.
- le diagnostic de processus, qui consiste à établir la nature de la faute détectée à l'étape de détection.
- la reconfiguration du système, qui suppose à ramener le processus dans des conditions opérationnelles nominales ou de le modifier afin de réduire, voire annuler, l'impact de la faute sur la qualité du produit fabriqué.

I.5.1 Fonctionnalités d'un système de contrôle

Les systèmes de contrôle industriels fonctionnent en arrière plans et en temps réel avec les systèmes commandés, ils doivent en permanence assurer les fonctions suivantes (Benoudina, 2009) :

- La mesure : l''état du système est déterminé par la mesure des grandeurs sur lesquelles s'effectue la commande (température, tension, vitesse…etc.) grâce aux différents capteurs, tout changement de ces grandeurs doit être pris en compte, c'est la propriété de sensibilité.

- L'affichage : un système de contrôle industriel doit afficher aux exploitants les états de tous les composants du système et les grandeurs mesurées pour que ces opérateurs soient plus proches du système et leur intervention très rapide.

- La régulation : toutes les mesures effectuées sont transmises vers les actionneurs et les automates programmables industriels. Une comparaison aura lieu pour calculer l'écart entre ces valeurs et les valeurs souhaitées. S'il y'a un simple décalage, un programme s'exécute permettant d'envoyer des signaux de réglage. S'il y'a un problème empêchant le réglage, une alarme se déclenche appelant l'intervention humaine.

I.6 Composants d'un système informatique industriel

Les systèmes informatiques utilisés dans le domaine industriels (que ce soit le système de conduite, de contrôle, de surveillance, …etc.) sont constitués de composants matériels et logiciels. Nous allons décrire dans cette section les principaux composants matériels.

I.6.1 Capteurs

Les capteurs (ou encore *sondes* et *senseurs*) sont « les sens » d'un système informatique industriel. Ils sont des composants physiques qui servent à obtenir des mesures sur les phénomènes physiques afin de les rendre exploitable par le système de contrôle. En d'autre terme, les capteurs transforment, comme illustrer sur la Figure I.4, les grandeurs physiques en gradeurs qui seront plus facile à utiliser, par exemple une tension électriques. Cette tension électrique sera ensuite numérisée pour l'exploiter dans un système informatique industriel.

Les mesures fournies par les capteurs doivent être précises, car ces mesures sont très sensibles pour le fonctionnement et la sécurité du système industriel.

Il existe plusieurs types de capteurs, dont les plus principaux sont ceux qui mesurent la température, la pression, la force, le débit, le niveau.

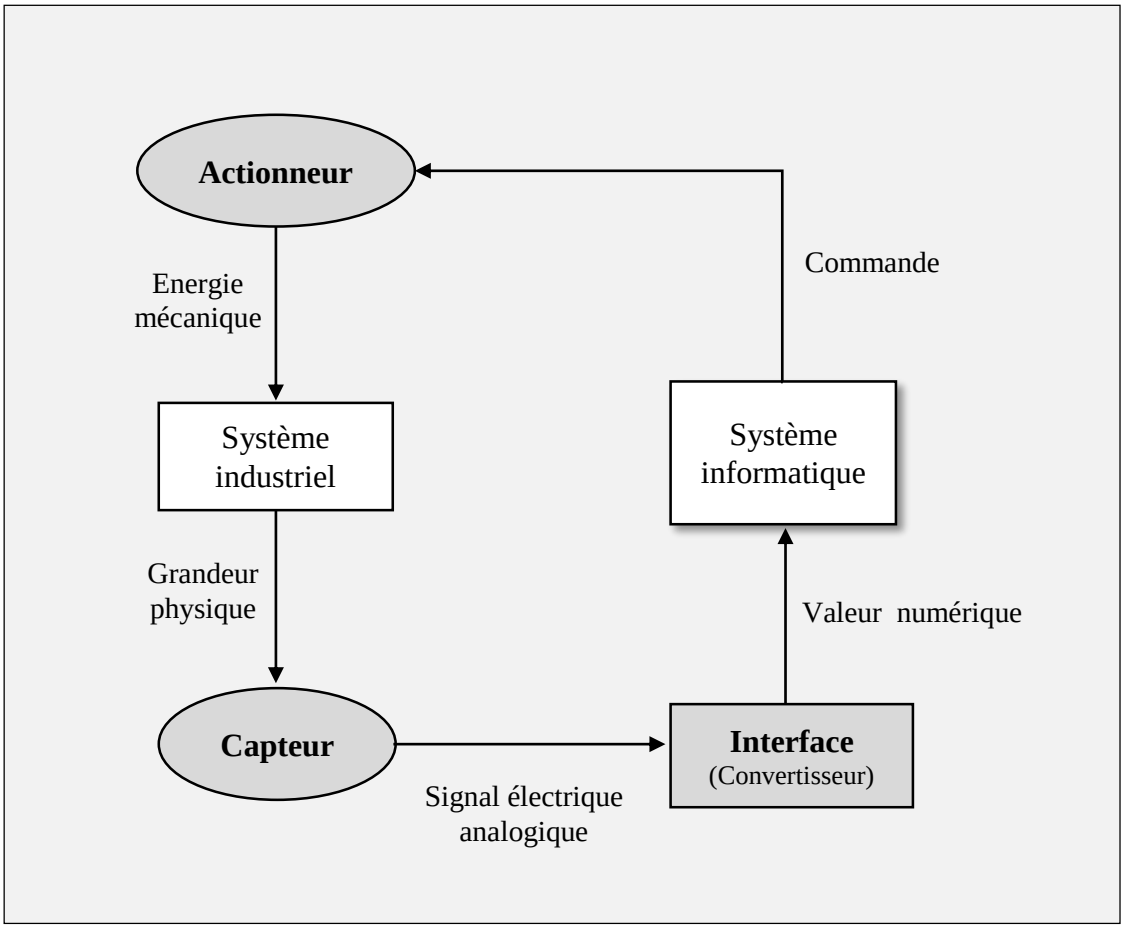

Figure I.4 – le rôle du capteur et de l'actionneur

I.6.2 Actionneurs

Les actionneurs sont « les muscles » d'un système informatique industriel. Ils ont pour rôle d'exécuter les ordres du système informatique et d'assurer l'évolution des processus industriels dans le sens souhaité. Les actionneurs reçoivent les commandes provenant du système informatique et les transforment en énergie mécanique au niveau du système physique piloté ou contrôlé (Figure I.4).

Il existe plusieurs types d'actionneurs, ils peuvent être des électrovannes, des moteurs, des relais, …etc.

I.6.3 Interfaces

Les interfaces assure le dialogue entre les capteurs et le système informatique d'un côté, et entre celui-ci et les actionneurs de l'autre côté. Autrement dit, elles permettent de traduire le langage du système informatique en langage machine et inversement. Comme montré sur la Figure I.4, l'interface « Convertisseur » sert à convertir le signal électrique analogique provenant du capteur en valeur numérique exploitable par le système informatique industriel.

I.6.4 Equipements de communication

Généralement, les entreprises industrielles comportent de milliers de capteurs et d'actionneurs répartis sur plusieurs régions. Pour assurer le bon fonctionnement du système et la coordination de ses différents éléments, Il est nécessaire d'avoir des équipements spécialisés pour assurer la transmission et l'échange des informations en toute sécurité et fiabilité et ce parfois nécessairement en temps réel.

Les systèmes industriels actuels font communiquer leurs équipements par l'utilisation des réseaux locaux industriels. Ces dernières ont pour objectif de faire dialoguer de nombreux et divers équipements, en intégrant une caractéristique importante, celle de fournir des services contraints par le temps.

I.7 L'automatisation des systèmes industriels

Un système industriel (Bergougnoux, 2005) est dit automatisé s'il exécute de manière autonome et automatique un cycle de travail préétabli qui se décompose en séquences et en étapes. Comme montrée dans la Figure I.5, les systèmes automatisés industriels possèdent une structure de base identique. Ils sont composés de deux parties principales reliées entre elles, plus ou moins complexes :

- une partie opérative.
- une partie commande (ou système de contrôle et de commande).

A partir de cette définition, nous pouvons déduire qu'un système industriel est dit automatisé lorsque son exécution, du début jusqu'à la fin, se fait sans intervention humaine et d'une manière répétitive.

I.7.1 Description des parties

I.7.1.1 La partie opérative

C'est la partie qui exécute le travail. Autrement dit, la partie physique du système. Elle reçoit les ordres de la partie commande et les exécute, ensuite elle transmet les comptes rendu à la partie commande.

La partie opérative comporte les éléments du procédé, c'est à dire (Bergougnoux, 2005):

- des capteurs qui informent la partie commande de l´exécution du travail. Par exemple, on va trouver des capteurs mécaniques, pneumatiques, électriques ou magnétiques montés sur les vérins. Le rôle des capteurs (ou détecteurs) est donc de contrôler, mesurer, surveiller et informer la partie commande sur l'´évolution du système.
- des pré-actionneurs (distributeurs, contacteurs) qui reçoivent des ordres de la partie commande ;
- des actionneurs (vérins, moteurs, vannes) qui ont pour rôle d'exécuter ces ordres. Ils transforment l'énergie pneumatique (air comprimé), hydraulique (huile sous pression) ou électrique en énergie mécanique ;

I.7.1.2 La partie commande

La partie commande gère et contrôle le déroulement des opérations à réaliser selon son programme. Elle gère aussi les consignes données par l'opérateur et les comptes rendu reçus de la partie opérative. Elle reçoit ces informations en provenance des capteurs de la partie opérative, et adresse les ordres vers cette même partie à travers les actionneurs.

La Figure I.5 montre la structure d'un système industriel automatisé ainsi que les interactions entre ses composants.

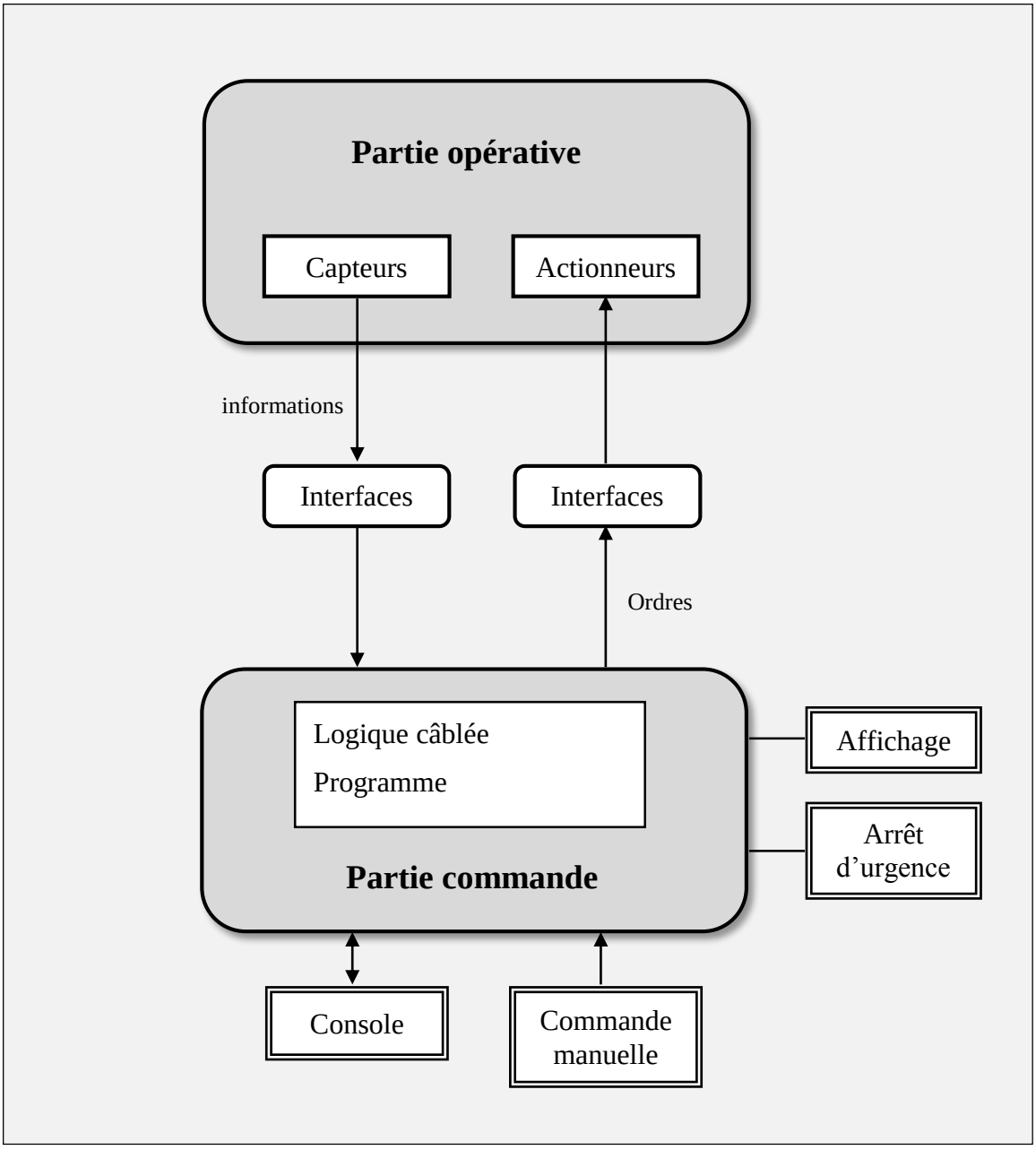

Figure I.5 – structure d'un système industriel automatisé

I.7.2 Les automates programmables industriels (API)

Les Automates Programmables Industriels (Abidi, 2014) sont apparus aux Etats-Unis à la fin des années soixante, à la demande des industries de l'automobile américaine pour développer des chaînes de fabrication automatisées qui pourraient suivre l'évolution des techniques et des modèles fabriqués.

L'automate programmable industriel est une machine électronique programmable, destiné à piloter en temps réel des procédés industriels. Elle réalise des fonctions d'automatisme pour assurer la commande de pré-actionneurs et d'actionneurs à partir d'informations logique, analogique ou numérique.

I.8 Conclusion

Les systèmes industriels actuels sont devenus de de plus en plus complexes et leurs constituants deviennent de plus en plus hétérogènes et décentralisés, cela est dû aux développements de nouvelles techniques de communication ainsi que les distances éloignées entre leurs composants ou entités. Il est donc primordial d'assurer la sécurité de ces systèmes par de nouveaux programmes informatiques capables de prendre en considération ces évolutions.

Comme nous allons le voir dans la suite de ce mémoire, les systèmes multi-agents et précisément les agents mobiles sont les mieux adaptés, pour sécuriser et surveiller les systèmes industriels complexes et distribués.

Chapitre II. Agents mobiles & sécurité

II.1 Introduction

Depuis les dernières décennies ont été marquées que l'informatique devient de plus en plus distribuée. Cependant, la programmation de telle application distribuée demeure difficile. La plupart de ces applications sont basées sur le modèle client/serveur ; souvent mal adaptée à l'aspect dynamique et à la diversité de tels systèmes. Le paradigme « Agent Mobile » semble une alternative intéressante à l'architecture client-serveur. L'idée est de remplacer l'envoi des données vers le l'hôte distant par la mobilité d'un agent qui peut se déplacer (avec son code, son état d'exécution et ses données propres), de son propre initiative, d'une machine à une autre et communique avec d'autres agents ou accède aux ressources distantes.

Le principale avantage de l'utilisation des agents mobiles est la mobilité de ces derniers qui permet de réaliser des interactions asynchrones et autonome ainsi que de réduire le coût de communication dans le réseau. Néanmoins, ce paradigme mis en évidence un sérieux problème de sécurité qui ne cesse pas de freiner son expansion.

Dans ce chapitre nous étudierons le problème de sécurité dans les systèmes d'agents mobiles. Nous commençons en premier lieu, par présenter brièvement la notion d'agent et les systèmes multi-agents. Ensuite, nous détaillons le paradigme d'agent mobile ainsi que l'aspect sécuritaire dans cette technologie. Nous achevons ce chapitre par mentionner quelques travaux ayants traité ce problème.

II.2 La technologie d'agent

II.2.1 Définition

Dans la littérature, on trouve plusieurs définitions d'agents. Elles se ressemblent mais diffèrent selon le type d'application pour lequel l'agent est conçu.

D'après Jacques Ferber (Ferber, 1995) un agent est une entité physique ou virtuelle :

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

(Wooldridge & Jennings, 1995) ont proposé la définition suivante pour un agent : « Un agent est un système informatique, *situé* dans un environnement, et qui agit d'une façon *autonome* et *flexible* pour atteindre les objectifs pour lesquels il a été conçu ».

II.2.2 Caractéristiques d'un agent

A partir de la définition précédente (proposée par Wooldridge & Jennings) on peut donc conclure les principaux caractéristiques d'un agent (Jarras & Chaib-draa, 2002) :

- *situé:* l'agent est capable d'agir sur son environnement à partir des entrées sensorielles qu'il reçoit de ce même environnement. Exemples: systèmes de contrôle de processus, systèmes embarqués, etc ;
- *autonome:* l'agent est capable d'agir sans l'intervention d'un tiers (humain ou agent) et contrôle ses propres actions ainsi que son état interne ;
- *flexible:* l'agent dans ce cas est:
  - *capable de répondre à temps:* l'agent doit être capable de percevoir son environnement et élaborer une réponse dans les temps requis ;
  - *proactif:* l'agent doit exhiber un comportement proactif et opportuniste, tout en étant capable de prendre l'initiative au "bon" moment ;
  - *social:* l'agent doit être capable d'interagir avec les autres agents (logiciels et humains) quand la situation l'exige afin de compléter ses tâches ou aider ces agents à accomplir les leurs.

En pratique, dépendamment des domaines considérés, certaines propriétés sont plus importantes que d'autres, il peut même s'avérer que pour certaines types d'applications, des propriétés additionnelles soient requises (tel que la mobilité, l'intelligence, etc…).

II.2.3 Classification des agents

Selon la granularité des agents, deux grandes classes peuvent être distinguées : les agents cognitifs et les agents réactifs.

II.2.3.1 Agents cognitifs

Selon (Ferber, 1995), La société d'agents cognitifs est composée d'un petit nombre d'agents "intelligent". Chaque agent dispose d'une base de connaissance comprenant l'ensemble des informations et des savoir-faire nécessaires à la réalisation de sa tâche et à la gestion des interactions avec les autres agents et avec son environnement. On dit aussi que les agents sont "intentionnels", c'est-à-dire qu'ils possèdent des buts et des plans explicites leur permettant d'accomplir leurs buts.

On parle ici d'agents intelligents ou rationnels qui sont capables à eux seuls de résoudre des problèmes complexes. Ils peuvent raisonner et prendre des décisions en utilisant leurs expériences qu'ils ont acquises en s'appuyant sur ses bases de connaissances (Figure II.1).

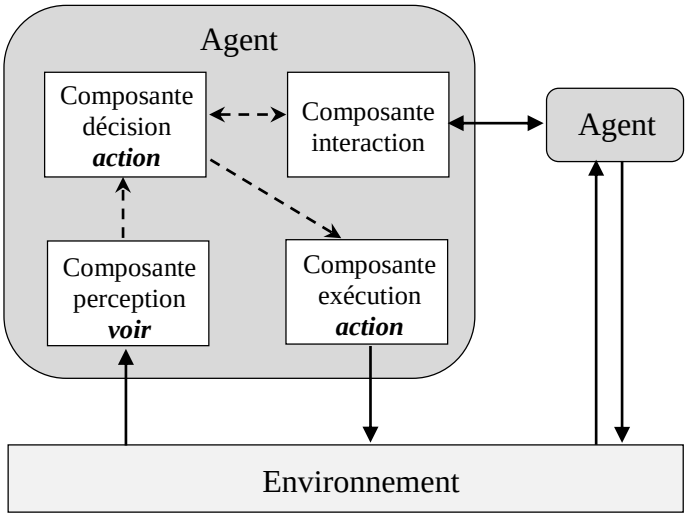

Figure II.1 - Structure d'un agent cognitif

II.2.3.2 Agents réactifs

Par opposition aux précédents, les agents réactifs sont des agents simples qualifiés non intelligents et ne possèdent pas de représentation de leur environnement. Leur comportement est basé sur le principe de « Stimulus-Réponse », c'est-à-dire, ils répondent d'une manière opportune aux changements de leurs environnements.

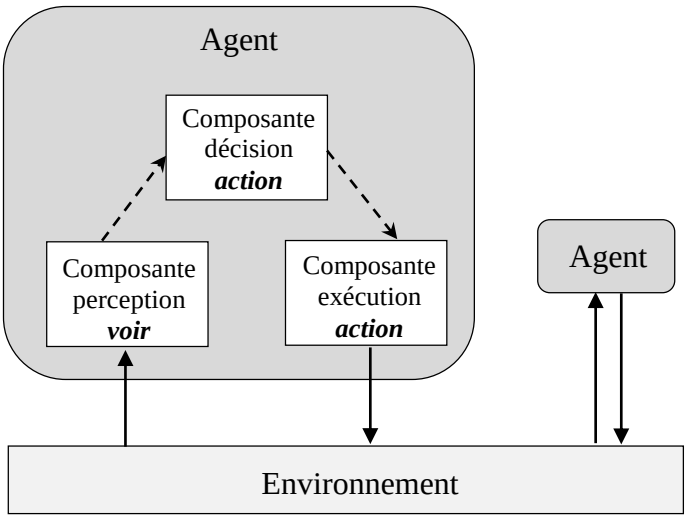

Figure II.2 - Structure d'un agent réactif

Les systèmes basés sur des agents réactifs sont caractérisés par le nombre important des agents qui sont capables ensemble de produire des actions complexes et coordonnées. C'est le cas de la fourmilière, où l'ensemble des fourmis sont capables d'accomplir des tâches extrêmement complexes, par contre chaque fourmi prise séparément n'a pas de but global et ne possède que d'une représentation partiale de l'environnement.

Le tableau suivant montre une comparaison entre les agents cognitifs et réactifs (Labidi & Lejouad, 1993) :

Systèmes d'agents cognitifs	**Système d'agents réactifs**
Représentation explicite de l'environnement	Pas de représentation explicite
Peut tenir compte de son passé	Pas de mémoire de son historique
Agents complexes	Fonctionnement stimulus / réponse
Petit nombre d'agents	Grand nombre d'agents

Tableau II.1 - Les agents cognitifs vs réactifs

II.3 Système multi-agents

À cause de leur approche principalement centralisée, les agents cognitifs rencontrent des difficultés pour résoudre les problèmes complexes et distribués, ce qui les a poussés à changer leur fonctionnement en distribuant l'intelligence dans différents éléments communiquant entre eux (Sansonnet, 2004). C'est de cette idée que viennent les systèmes multi-agents.

II.3.1 Définition

La plupart des auteurs sont défini un système multi-agents (SMA) comme « un ensemble organisé d'agents » qui communiquent et collaborent pour atteindre leur but commun.

Jacques Ferber (Ferber, 1995) propose une définition plus précise d'un SMA :

On appelle système multi-agent ou SMA, un système composé des éléments suivants :

- *Un environnement E*, c'est à dire un espace disposant généralement d'une métrique.

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

La figure II.4 (Ferber, 1995) donne une représentation d'un système multi-agents.

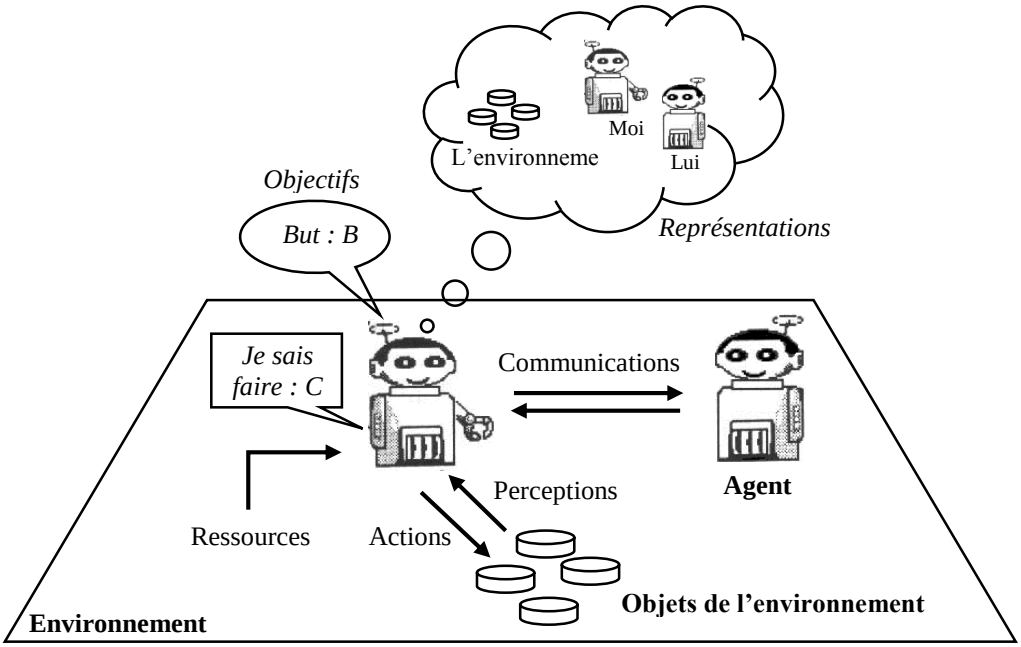

Figure II.3 - Représentation d'un système Multi-Agents

II.3.2 Propriétés des systèmes Multi-Agents

Un système Multi-Agent est caractérisé par (Jennings, Sycara, & Wooldridge, 1998) :

- Chaque agent possède des informations incomplètes ou des capacités de résoudre de problèmes limitées, donc chaque agent à un point de vue partiel ;
- Il n'y a pas de contrôle global du système ;
- Les données sont décentralisées ;
- Le calcul est asynchrone.

II.4 Agents Mobiles

II.4.1 Définition

Contrairement à un agent situé qui s'exécute localement pendant son cycle de vie, un agent mobile est capable de se déplacer au cours de son exécution dans le réseau, d'un site à un autre pour accéder à des données (ou des ressources) ou à la demande d'un client (autre agent ou humain). Il se déplace avec ses données propres et son code, ainsi que avec son état d'exécution. C'est un paradigme de plus en plus utilisé dans les systèmes distribués.

Lorsqu'un client donne une mission à un agent, ce dernier se déplace dans le réseau en accédant localement aux services offerts par les hôtes du réseau. On peut distinguer trois phases (Perret, 1997):

1. l'activation de l'agent mobile avec la description de sa mission ;
2. l'exécution de la mission par l'agent qui se déplace pour accéder aux services ;
3. la récupération éventuelle des résultats de l'agent mobile.

II.4.2 Types de mobilité

La mobilité (ou la migration) d'un agent mobile peut s'effectuer selon deux types :

- Mobilité faible : La mobilité faible, ne permet de transférer avec l'agent que son code et ses données, il n'y a pas de migration de l'état d'exécution. L'agent redémarre son exécution au début sur l'hôte de destination (l'exécution de l'agent est réinitialisée).

- Mobilité forte : La mobilité forte, où l'agent en entier (données, code et état d'exécution) migre vers les différents hôtes. Dans ce cas, l'agent avant d'être migré, il suspend son état d'exécution. Une fois arrivé à sa destination, il reprend son exécution au point précédent.

II.4.3 Agents mobiles et client-serveur

Le modèle client-serveur est certainement le plus utilisé pour le développement des applications réparties. Selon ce modèle les échanges entre le client et le serveur se font par envoi des messages de façon interactive et synchrone (Figure II.4).

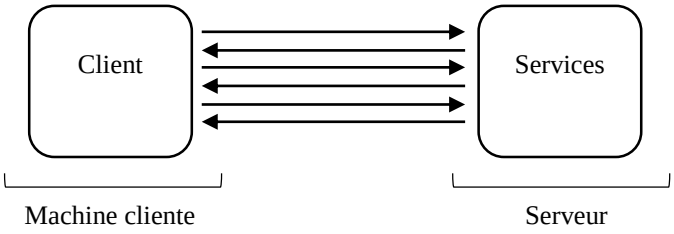

Figure II.4 - Modèle Client-Serveur

En effet, l'inconvénient majeur de ce modèle c'est que tous les composants du système (ressource nécessaire pour l'exécution, code à exécuter et état d'exécution du code) sont stationnaires de manière qu'ils ne peuvent interagir que d'une manière distante ce qui augmente le trafic sur le réseau et exige une connexion permanente entre le client et le serveur (LOULOU, 2010).

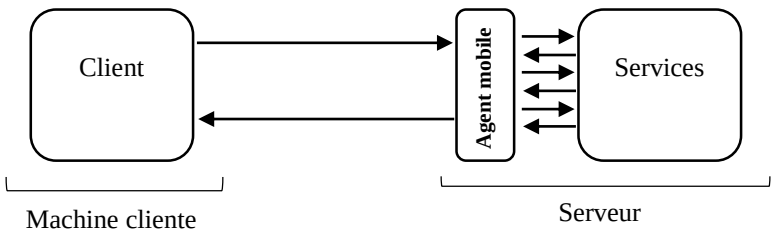

Figure II.5 - Modèle Agent Mobile

Par contre, dans la technologie d'agents mobile, un agent se déplace d'une manière autonome à travers le réseau, d'une machine à une autre, pour exécuter des tâches d'une manière asynchrone (Figure II.5). La mobilité de l'agent permet de décharger le client et déléguer l'exécution de certaines tâches sur d'autres machines du réseau. Ce qui permet de dire que ce paradigme est mieux adapté que le modèle client-serveur surtout sur des réseaux à grande échelle.

II.4.4 La standardisation

L'émergence de nombreuses plates-formes d'exécution des agents mobiles a rendu essentielle de proposer une standardisation des différents concepts et fonctionnalités communs des plateformes d'agents mobiles dans le but d'assurer un haut niveau d'interopérabilité.

Au temps présent, il existe deux normes principales (CUBAT, 2005):

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

II.5 Besoins de sécurité

Dans le cadre de la sécurité des systèmes à base d'agents mobiles on trouve principalement deux classes différentes. La première classe concerne la sécurité des agents mobiles et la deuxième s'occupe de la sécurisation des hôtes qui les hébergent et les exécutent. Nous allons étudier ce problème dans ce qui va suivre, mais en premier lieu, nous allons développer les besoins de sécurité requis dans cette technologie, tel que la confidentialité, l'authenticité, l'intégrité et la disponibilité.

- **Confidentialité :** la confidentialité assure que l'information stockée dans un hôte ou transmise par un agent mobile n'est pas accessible à ceux qui n'en ont pas l'autorisation. Les hôtes, ainsi que les agents mobiles doivent protéger leurs informations privées contre les accès non autorisés.
- **Authenticité :** l'authentification garantit la reconnaissance sûre de l'identité. Elle consiste à vérifier que l'identité d'une entité est comme déclarée. En fait, un agent mobile doit s'authentifier sur chaque système d'agents visité et par conséquent, un système d'agents est alors capable de décider s'il s'agit d'un agent de confiance. Dans le même ordre d'idée, l'agent mobile doit être capable d'authentifier le système d'agents.
- **Intégrité :** l'intégrité des données assure que les données n'ont pas été modifiées par une entité non autorisée. La plateforme doit protéger le code, l'état, et les données de ses agents contre des modifications non autorisées, et d'un autre côté, assurer que seuls les agents autorisés emportent des modifications aux informations qu'elle expose.
- **Disponibilité** *:* la disponibilité assure d'une part que les ressources sont accessibles par ceux qui en ont besoin tant qu'il s'agit des entités autorisées. D'autre part, cette propriété assure l'utilisation convenable et non abusive des services et/ou des ressources du système.

II.6 Sécurité des hôtes

II.6.1 Les attaques contre les hôtes

Les attaques contre les hôtes peuvent prendre différents types. On distingue principalement les types d'attaques suivants :

- **Mascarade :** Ce type d'attaque se produit lorsqu'un agent masque son identité et se sert de l'identité d'un autre agent pour le but d'obtenir des privilèges dont il n'a pas le droit ou de faire endosser la responsabilité de certaines actions à un autre agent.
- **Déni de service :** Le déni de service a lieu quand un agent consomme ou corrompre les ressources de l'hôte de destination, comme l'espace disque, le processeur ou la bande passante de la connexion réseau. Ce type d'attaque peut être intentionnel ou non (erreur de programmation). Étant donné que le code de l'agent mobile est écrit dans un hôte autre que celui où il entraine d'exécuter son code, il peut contenir du code malicieux.
- **Accès non autorisé :** Les agents mobiles malveillants tentent d'accéder à des ressources et des services de l'hôte sans qu'ils possèdent les autorisations adéquates. Afin de contrecarrer cette attaque, la plate-forme doit avoir une politique de sécurité précisant les règles d'accès applicables aux différents agents, ainsi qu'un mécanisme pour appliquer cette politique (Alfalayleh & Brankovic, 2005).
- **Répudiation :** La Répudiation se produit quand un agent arrive à causer une rupture de la communication qui a eu lieu entre les éléments du système. La répudiation que ce soit causer d'une manière accidentelle ou planifiée, peut mener à de sérieux problèmes qui ne peuvent pas être résolus facilement.

II.6.2 La protection des hôtes

La protection des hôtes contre les attaques est un problème qui est aujourd'hui bien maîtrisé. Beaucoup de solutions sont proposées et voici les techniques les plus connues (Jansen & Karygiannis, 1999; Pierre, 2011; Loureiro, Molva, & Roudier, 2000; Rubin & Geer, 1998; Fong, 1998):

- Carré de sable
- Signature du Code

- Code avec preuve
- Authentification des agents
- Contrôle d'accès et d'autorisation
- Vérification du code
- Estimation d'état
- Historique des hôtes
- Technique de limitation
- Journal d'audit

II.6.2.1 Carré de sable

L'idée principale de la technique de carré de sable (Sandbox) consiste à limiter l'accès d'un programme au système d'exploitation sous-jacent. En d'autre terme, le code est exécuté dans une sorte de « carré de sable ». L'accès à l'environnement est restreint en termes de ressources et de privilèges ; et par conséquent, les risques sont limités. Pour ce faire, on va confiner le code du programme dans un environnement dont les accès au système d'exploitation sont limités et contrôlés (Figure II.6). Étant donné que les ressources du code à exécuter dans le carré de sable sont limitées, celui-ci ne peut avoir que des fonctionnalités réduites.

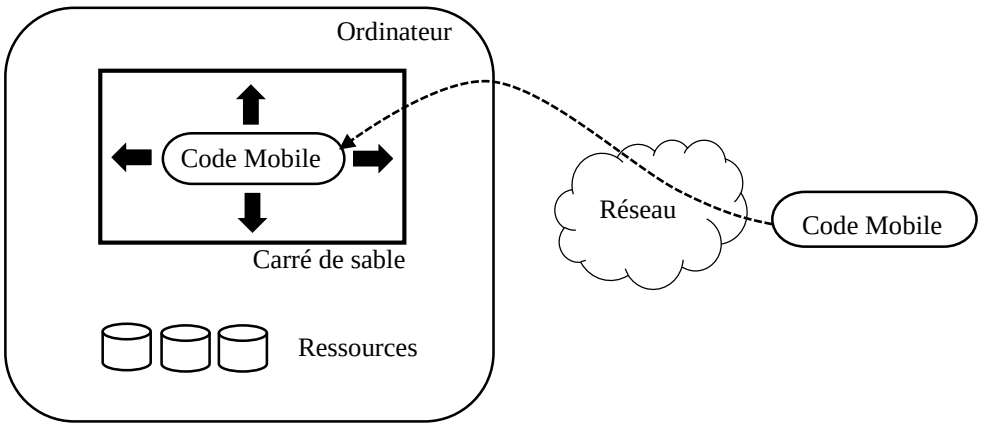

Figure II.6 - Carré de sable

II.6.2.2 Signature du Code

La signature numérique du code permet de confirmer l'authenticité du code de l'agent mobile, de son origine et de son intégrité. Le créateur de l'agent mobile va signer numériquement son code lors de sa création afin qu'il puisse s'identifier pendant sa migration sur les différents hôtes. Si sa signature est valide, on peut être sûr de l'hôte d'origine du code et que celui-ci n'a pas subi de modification (Figure II.7).

La signature numérique a été utilisée par Microsoft dans ses contrôleurs ActiveX et aussi dans le JDK Java 1.1. Ce modèle bénéficie de la disponibilité de l'infrastructure à clé publique, puisque le certificat contenant l'identité de l'entité et sa clé publique peut être facilement vérifié et localisé.

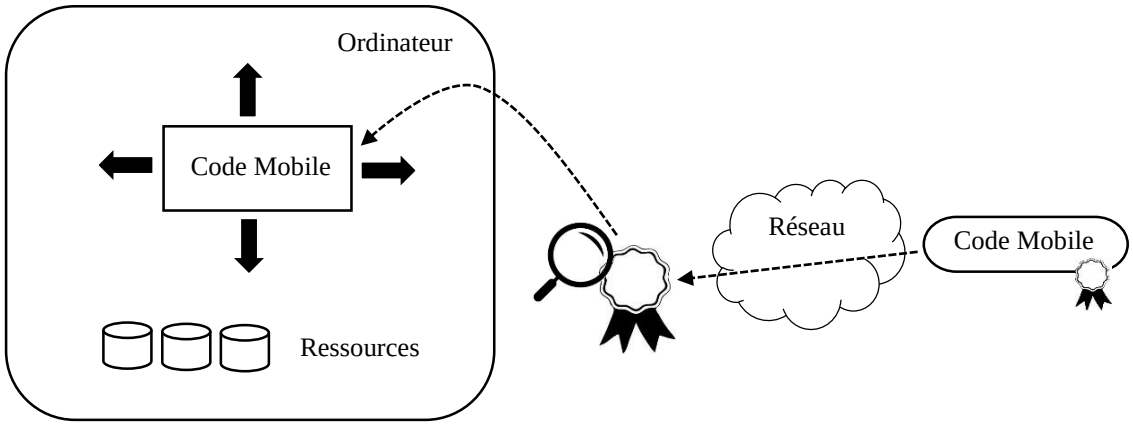

Figure II.7 - Signature numérique du code

L'inconvénient de cette méthode est qu'elle ne garantit pas le comportement de l'agent mobile lors de son exécution, même si l'intégrité de son code est prouvée. C'est-à-dire, si le code de l'agent est signé, il possède un accès complet à toutes les ressources de l'hôte de destination et il peut exécuter des actions nuisibles à ce dernier, sinon (le code de l'agent n'est pas signé) l'agent ne sera jamais exécuté.

II.7 Sécurité des agents mobiles

Les méthodes de sécurité traditionnelles ne suffisent pas à protéger efficacement l'agent mobile. Puisque l'hôte qui exécute un agent mobile, il a un plein pouvoir sur lui ; Du moment qu'il l'héberge, il peut facilement accéder et changer ses données ainsi que son code. Par conséquent, les problèmes de sécurité des agents mobiles sont différents et beaucoup plus de ceux rencontrés dans les systèmes distribués.

En effet, il existe trois volets de sécurité d'un agent mobile :

- La sécurité de l'agent contre un autre agent malveillant ;
- La sécurité de l'agent contre un hôte malveillant ;
- La sécurité de l'agent contre le réseau de communication.

II.7.1 Les différentes attaques contre un agent mobile

On distingue les attaques suivantes contre un agent mobile :

- **Mascarade :** La mascarade se produit lorsqu'un hôte malicieux masque son identité pour tromper l'agent mobile et s'identifier comme sa vraie destination. De cette façon, l'hôte peut obtenir des informations sensibles contenues dans l'agent mobile. La mascarade est généralement suivie d'autres types d'attaques comme l'espionnage des données.
- **Déni de service :** Lors d'un déni de service (Pierre, 2011), la plate-forme refuse d'effectuer les services demandés par l'agent. Ainsi, elle peut ignorer les demandes de service, introduire des délais inacceptables pour des tâches critiques, ne pas exécuter le code de l'agent ou bien le terminer sans avertissement. D'autres agents qui attendent la réponse de cet agent seront en inter-blocage (deadlock). Par exemple, si une plate-forme ne donne pas d'accès réseau à un agent mobile, il ne parvient pas à se déplacer.
- **Espionnage :** Dans ce type d'attaque, un hôte malveillant interprète l'agent mobile en analysant son comportement afin d'extraire ses données sensibles (par exemple l'identité de l'hôte d'origine, le prix d'un produit, les caractéristiques techniques…). Ceci est généralement utilisé lorsque le code et les données de l'agent mobile sont cryptés.

- **Altération :** Une altération a lieu (Alfalayleh & Brankovic, 2005) lorsqu'un hôte malveillant modifie le contenue de l'agent mobile (ses données, son code et/ou son état d'exécution). La modification du comportement de l'agent mobile peut l'entraîner à exécuter des actions nuisibles à d'autres hôtes, y compris l'hôte d'origine.

II.7.2 La protection des agents mobiles

Alors que les contre-mesures (Pierre, 2011) visant à protéger les plates-formes s'inspirent des systèmes conventionnels en employant des méthodes préventives, les techniques de protection des agents font plutôt de la détection. C'est que l'agent est complètement dépendant de la plate-forme sur laquelle il s'exécute et ne peut, par lui-même, empêcher une attaque. Par contre, on peut lui donner les moyens de la détecter ou de la rendre moins dangereuse.

La protection des agents mobiles est un problème très difficile à résoudre et reste aujourd'hui un champ de recherche ouvert. Supposons qu'un hôte envoie son agent (code, données et état) sur le premier hôte de destination. Lors de son migration, Il n'y aura plus de vérification sur le comportement de l'agent.

Les techniques de protections des agents mobiles les plus connues sont : (Jansen & Karygiannis, 1999; Pierre, 2011; Loureiro, Molva, & Roudier, 2000; Rubin & Geer, 1998; Fong, 1998) :

- Enregistrement d'itinéraires avec des agents coopérants
- Traces cryptographiques
- Génération des clés à partir de l'environnement
- Calcul de fonctions cryptographiques
- Boite noire limitée dans le temps
- Encapsulation de résultats partiels
- Protection de l'agent par la tolérance aux fautes
- Marquage du code
- Tiers de confiance

II.7.2.1 Traces cryptographiques

Cette technique (Vigna, 1997) a été proposée pour la détection de toute exécution anormale d'un agent mobile après son retour à partir d'un fichier appelé " *trace "*. Cette trace contient l'historique de l'exécution de l'agent sur les différentes hôtes visités, ce qui permet de détecter facilement l'hôte malicieux. Après le retour de l'agent mobile à son hôte d'origine, et quand ce dernier a un doute à la bonne exécution de son agent, il le ré-exécute grâce au traces et, en cas d'erreur, il peut identifier l'hôte coupable de mauvaise exécution de l'agent.

Les traces sont des paires (*n*, *s*) où :

n : identifiant d'une ligne de code ;
s : signature de la ligne de code.

Par exemple :

Code : 3. read (x) où : 3 est l'identifiant de la ligne du code.

Trace : (3, x = 9)

Il y a des inconvénients de cette approche telles que :

- Consommation de ressources (temps C.P.U.) ;
- La taille du trace est potentiellement grande ;
- Pour détecter une attaque, il faut attendre jusqu'au la fin de l'exécution de l'agent mobile ;
- Dans le cas de 'Multi-Thread' (un agent a plusieurs fils), cette technique est difficile à utiliser.

II.7.2.2 Calcul de fonctions cryptographiques

Le calcul de fonctions cryptographiques est proposé par (Sander & Tschudin, 1998), où la protection de l'agent mobile est basée sur l'exécution des fonctions cryptées. Il s'agit d'exécuter sur l'hôte de destination un programme encrypté sans qu'il soit capable de le décrypter. L'agent mobile est en sorte de boîte noire pour l'hôte sur lequel il s'exécute.

Par exemple : « Anes a un algorithme qui calcule la fonction *f*. Hamza a une entrée *x* et veut calculer *f(x)* pour Anes, mais il veut que Hamza n'apprenne rien à propos de *f*. ». La méthode est la suivante :

(1) Anes encrypte *f*.

(2) Anes crée un programme *P(E(f))* qui implémente *E(f)*.

(3) Anes envoie *P(E(f))* à Hamza.

(4) Hamza exécute *P(E(f))* à *x*.

(5) Hamza envoie *P(E(f))(x)* à Anes.

(6) Anes décrypte *P(E(f))(x)* et obtient *f(x)*.

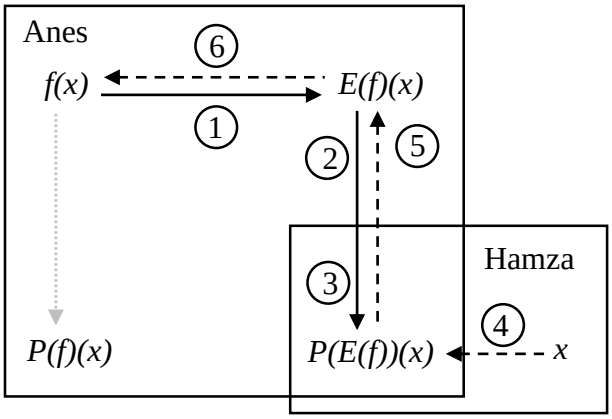

Figure II.8 - Calcul de fonctions cryptographiques

L'inconvénient de cette technique est qu'elle est valide uniquement pour les fonctions rationnelles et polynomiales.

II.8 Travaux connexes

Nous allons aborder, dans ce qui suit, une revue sur quelques travaux qui sont conçus pour gérer la sécurité des systèmes distribués par une approche de détection d'intrusion à base d'agents.

II.8.1 L'approche DIDS

L'approche DIDS « Distributed Intrusion Detection System » (Snapp, 1991), a été l'une des premières approches de détection d'intrusions à base d'agents. Dans cette approche chaque agent est chargé de collecter les données de l'hôte surveillé et les envoyer à l'unité centrale qui fait l'analyse de ces données. Tous les agents utilisés dans cette approche sont stationnaires dans tous les hôtes surveillés.

L'approche DIDS se compose de trois entités (Figure II.9) :

(1) L'agent Hôte (Host Monitor) : il collecte les données de l'hôte surveillé. Cet agent fait une première analyse simple sur les données pour extraire les informations pertinentes puis les transmet à l'unité centrale d'analyse « DIDS Director ».

(2) L'agent LAN (LAN Monitor) : il est chargé de collecter les informations réseaux en surveillant le trafic sur le LAN et reporte à l'unité centrale « DIDS Director » les comportements anormaux.

(3) Le gestionnaire DIDS « DIDS Director » : il joue le rôle le plus important dans l'architecture, il analyse les rapports transmises par les agents « LAN Monitor » et « Host Monitor » pour détecter les attaques probables.

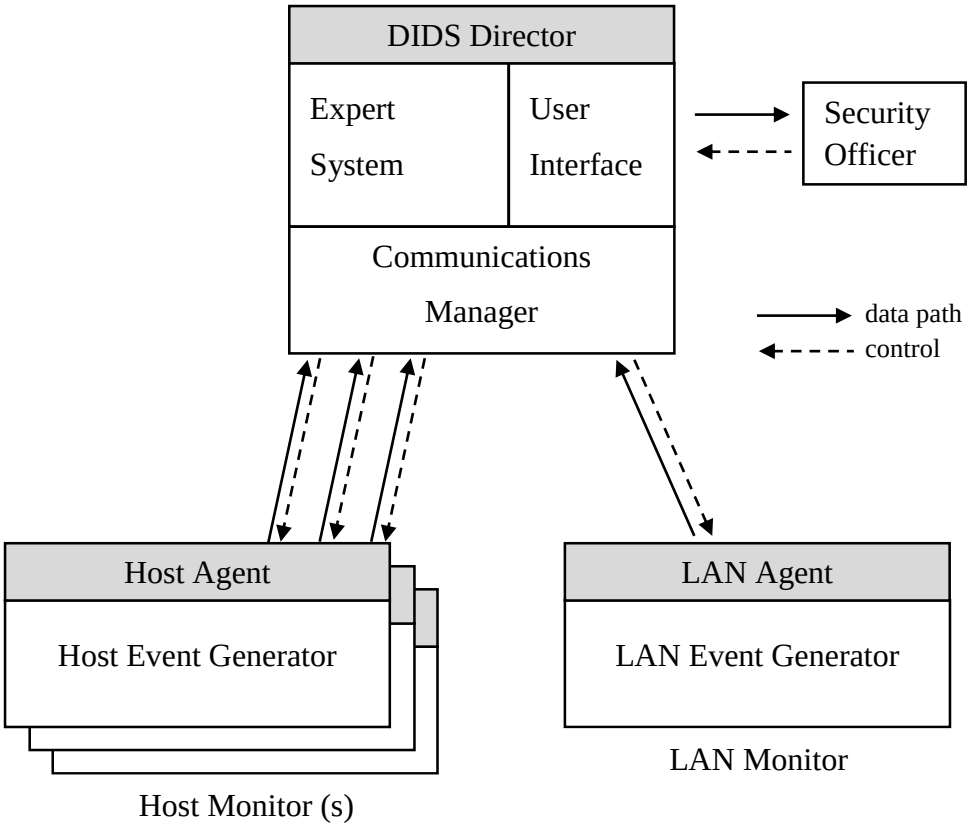

Figure II.9 - Architecture de l'approche DIDS

II.8.2 L'approche AAFID

L'approche AAFID (Autonomous Agent For Intrusion Detection) a été proposée par (Balasubramaniyan, Garcia-Fernandez, Isacoff, Spafford, & Zamboni, 1998) à l'université de Purdue. C'est une approche de détection d'intrusions multicouche avec des agents stationnaires et autonomes organisés sous forme d'une architecture hiérarchique. Chaque couche de la hiérarchie s'occupe de la détection des attaques spécifiques.

Les deux figures ci-après (Figure II.10, Figure II.11) représentent l'architecture physique et logique de l'approche AAFID et montrent qu'elle est composée de trois entités essentielles, qui sont :

(1) Les agents : le rôle d'un agent est de surveiller certains aspects d'un hôte, ensuite il reporte au « transceiver » de cet hôte, les informations liées aux activités suspicieuses. un agent ne peut pas générer des alarmes et ne communique pas avec d'autres agents. un ou plusieurs agents peuvent surveiller le même hôte.

(2) Les transmetteurs (transceivers) : sont l'interface externe de communication de chaque hôte. il doit y avoir obligatoirement un pour chaque hôte surveillé. Les « transceivers » ont deux rôles, contrôle et traitement des données.

- Le contrôle : ils lancent et arrêtent les agents s'exécutant sur ses hôtes. Ils gardent des traces de ces agents et répondent aux instructions provenant de ses moniteurs en fournissant les informations nécessaires ou d'effectuer les opérations appropriées.
- Le traitement des données : ils reçoivent et analysent les rapports générés par les agents s'exécutant sur ses hôtes, ensuite envoient les résultats soit à d'autres agents, soit au moniteur approprié.

(3) Les moniteurs (Monitor) : sont les entités de plus haut niveau dans l'architecture AAFID. Ils ont aussi un rôle de contrôle et un rôle de traitement de données de la même manière que les « transceivers ». La différence principale entre les deux entités est que le moniteur peut contrôler des entités (transceivers et autre moniteurs) qui s'exécutent sur plusieurs hôtes différents alors que le transceiver ne contrôle que les agents locaux.

Dans leur rôle de contrôle, les moniteurs reçoivent les instructions provenant d'autres moniteurs et contrôlent des transceivers et autres moniteurs.
Dans l'autre rôle (traitement de données), les moniteurs reçoivent les informations de tous les transceivers qu'ils contrôlent, ensuite font des corrélations de plus haut niveau et détectent les événements qui impliquent plusieurs hôtes. Les moniteurs ont la capacité de détecter les événements indécelable par les transceivers. De plus, les moniteurs communiquent avec l'interface utilisateur et fournir un accès au système AAFID.

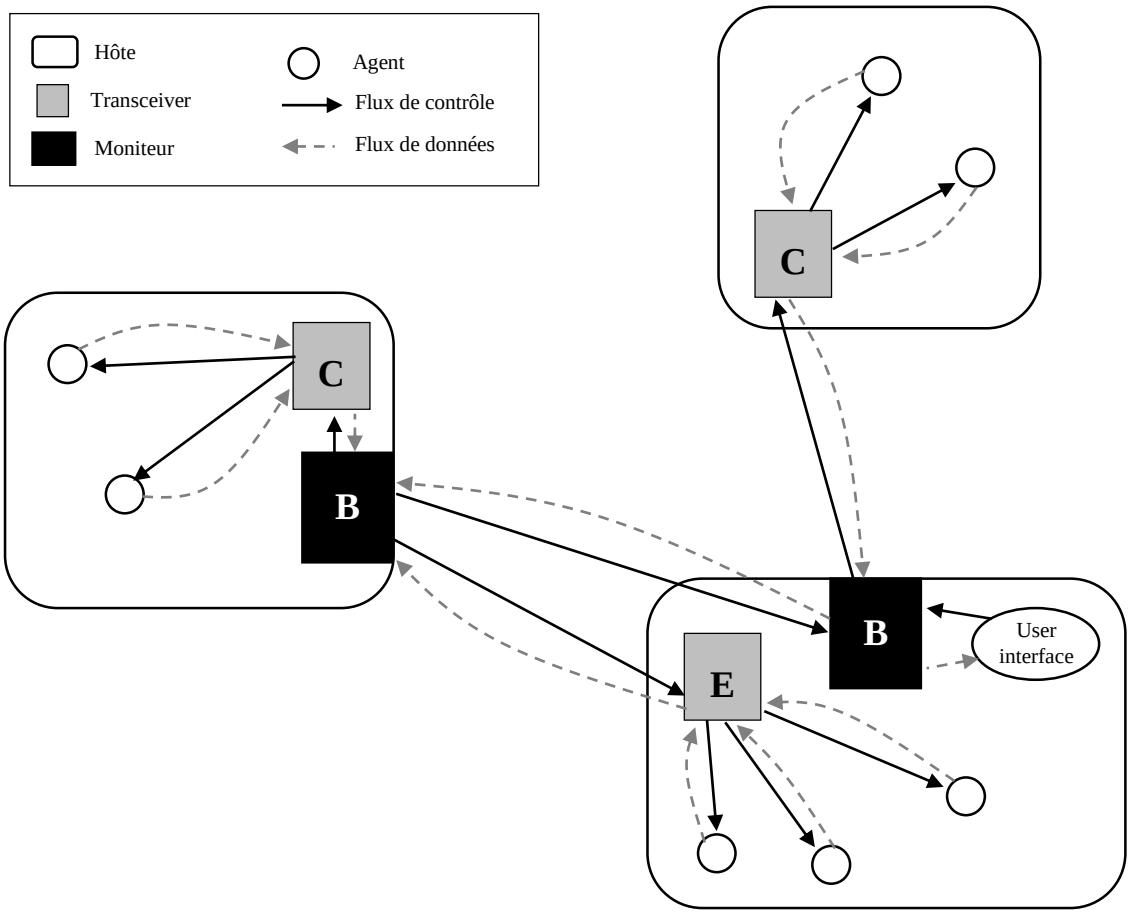

Figure II.10 - Architecture physique de l'approche AAFID

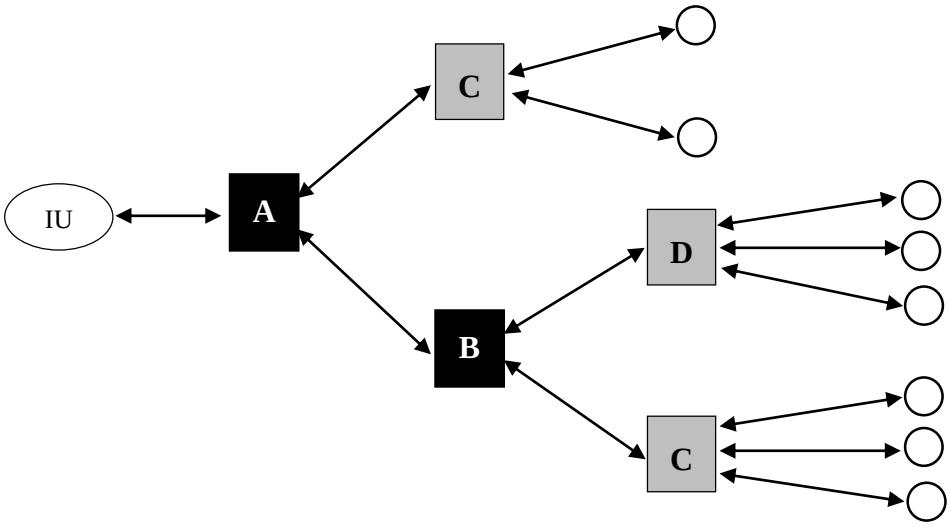

Figure II.11 - Architecture logique de l'approche AAFID

La mobilité des agents était absente dans l'approche AAFID. Après quelque années, (Skaita & Mourlin, 2002) ont proposé une nouvelle approche appelée AAFID 3 en ajoutant la mobilité aux agents.

II.8.3 L'approche MADIDF

(Ye, Bai, Zhang, & Ye, 2008) ont proposé une architecture complètement distribuée (MADIDF - Mobile Agents Distributed Intrusion Detection Framework by Using) basée agents mobiles pour la détection d'intrusion. L'approche utilise à la fois des agents stationnaires et des agents mobiles. Elle est composée de six agents :

- Quatre agents stationnaires : agent moniteur, agent analyseur, agent exécutif et agent manager.
- Deux agents mobiles : agent Retrieval et agent Result.

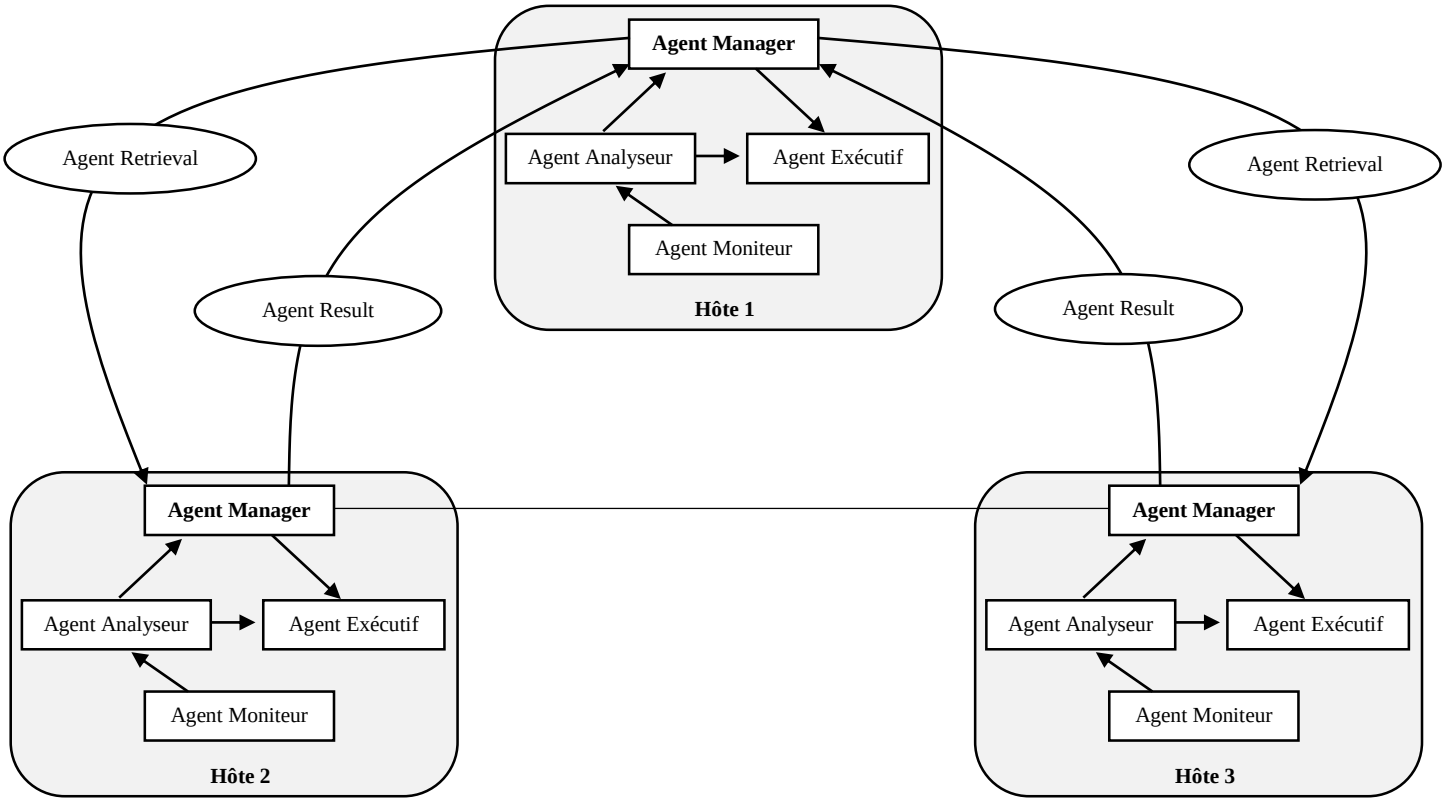

Figure II.12 - Architecture de l'approche MADIDF

Les quatre agents stationnaires s'exécutent sur chaque hôte surveillé. La Figure II.12 démontre l'architecture de l'approche MADIDF. Nous décrivons le rôle de chaque agent dans cette architecture :

(1) L'agent Moniteur : le rôle de cet agent est de surveiller les activités locales de l'hôte qui l'exécute. Ces activités concernent à la fois, le trafic réseau et l'audit du système. Quand une activité suspecte est détectée par cet Agent, il collecte et prétraite les informations pertinentes, puis les envois à l'Agent Analyseur.

(2) L'agent Analyseur **:** L'agent analyseur intègre et analyse les informations reçues de l'agent de moniteur. Dans l'approche MADIDF, chaque hôte dans le réseau a une base de connaissance locale qui contient des informations critiques, tel que les signatures d'attaque, les modèles d'intrusion, la taille des fichiers dans le système, …etc. Si l'agent analyseur confirme la présence d'une intrusion ou d'une attaque, il envoie une notification à l'agent exécutif pour prendre toutes les mesures nécessaires. Si l'agent analyseur suspecte d'une attaque, il demande à l'agent Manager de sauvegarder l'activité suspecte.

(3) L'agent Exécutif **:** son rôle est d'exécuter des tâches en fonction de la notification de l'agent analyseur. Ces tâches comprennent la restauration de fichiers corrompus, empêchant la connexion réseau, et ainsi de suite.

(4) L'agent Manager **:** cet agent collecte des informations et détecte les attaques à partir de plusieurs hôtes du réseau, alors que l'agent Moniteur collecte seulement les informations de son hôte. L'agent Manager gère les processus de récupération des informations contenues dans différents hôtes en utilisant les deux agents mobiles « Retrieval » et « Result ».

(5) L'agent Retrieval **:** il se déplace vers les autres hôtes distants et demande aux agents Analyseurs de ces hôtes de vérifier s'il existe des données similaires d'une même activité suspicieuse. Il existe quatre principaux types d'informations que l'agent de récupération doit maintenir, qui sont l'adresse IP de source d'où l'hôte d'origine distribue cet agent de récupération, des personnages de l'incident, la recherche Agent Identifier (RAID), et temps de vie (TTL). L'agent Retrieval sera supprimé lorsque la valeur du TTL (son durée de vie) atteint zéro ou il n'y a plus d'hôte à parcourir.

(6) L'agent Result **:** cet agent est dispatché avec un rapport de résultat vers un autre hôte, qui a été sollicité ce rapport par l'agent Retrieval (hôte initiateur). ensuite, l'agent Manager de l'hôte initiateur examine le rapport reçu pour prendre des décisions finales.

Le diagramme de séquence représenté dans la Figure II.13 décrit le fonctionnement de l'approche MADIDF, où :

1. **monitor ()** : surveille les activités locales.
2. **report ()** : rapporte à l'Agent Analyseur les informations liées à l'activité suspecte se déroulant dans l'hôte surveillé.
3. **analyse ()** : analyse les informations réçues et décide s'il ya une intrusion ou attaque en se basant sur la base de connaissances locale.
4. **find_intrusion ()** : informe l'Agent Exécutif de l'existence d'une attaque ou intrusion.
5. **handle_intrusion ()** : prendre les mesures nécessaires contre l'attaque ou l'intrusion.
6. **request_Retrieval ()** : demande à l'Agent Manager de vérifier une activité suspecte.
7. **generate () :** crée l'Agent Retrieval pour demander la collecte des informations liées à l'activité suspecte sur les différents hôtes distants.
8. **visit ()** : se déplace vers l'hôte distant.
9. **generate ()** : crée l'Agent Result
10. **report ()** : rapporte à l'Agent Manager initiateur (celui qui envoie l'Agent Retrieval) les informations liées à l'activité suspecte.
11. **broadcast ()** : diffuse le résultat de détection de l'attaque à d'autres hôtes du réseau, et informe l'Agent Exécutif local de prendre des mesures contre l'attaque.

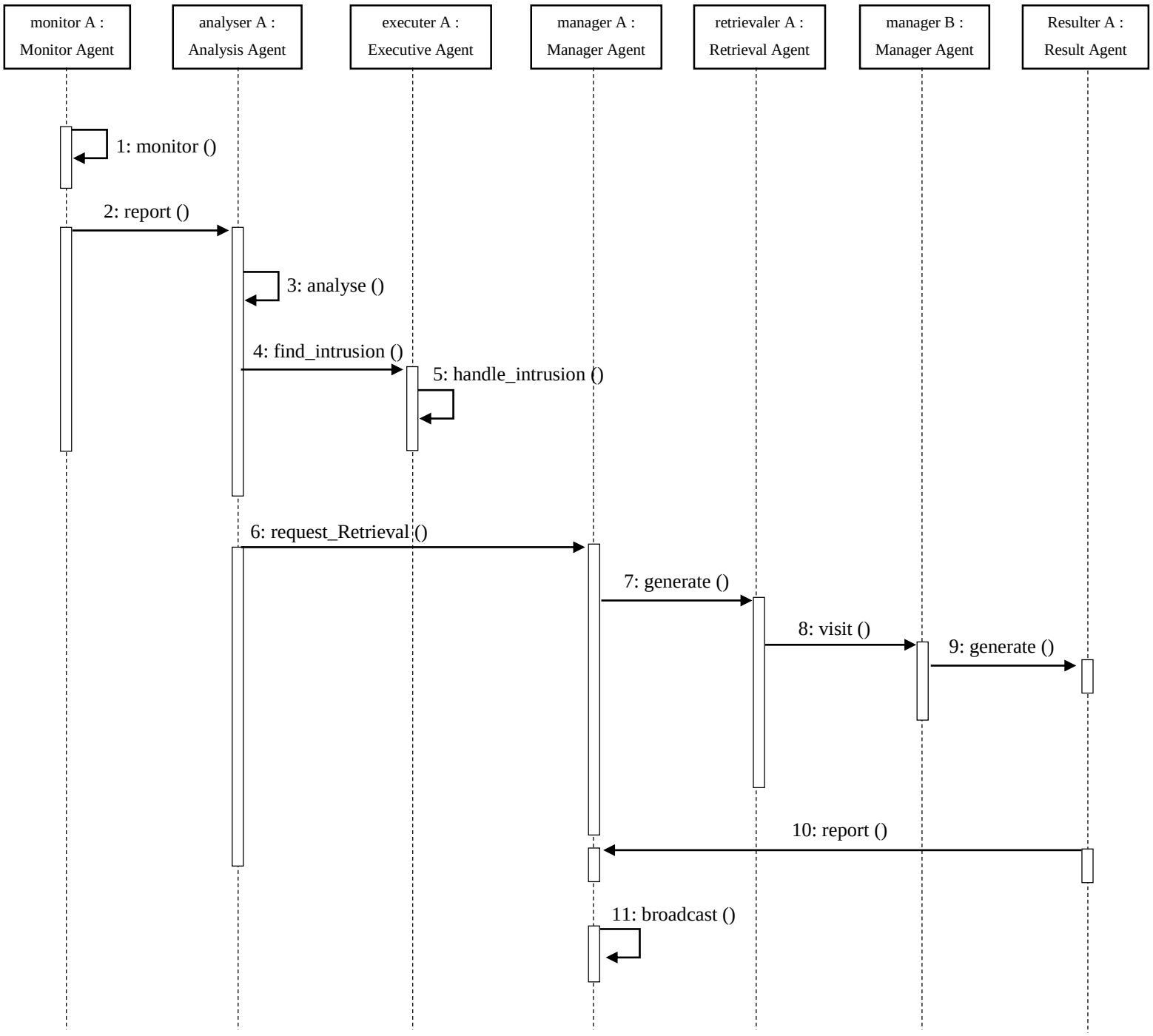

Figure II.13 - Diagramme de séquence général de l'approche MADIDF

II.8.4 Synthèse

Dans cette section nous allons faire une synthèse sur les travaux décrits précédemment dans laquelle on discutera les avantages et les inconvénients de chacun.

La structure de l'architecture DIDS est la même que celle des architectures centralisées monolithiques, la seule différence est l'utilisation des agents pour collecter les données. L'inconvénient majeur de cette approche est la dépendance totale à l'unité centrale « DIDS Director », dans laquelle cette dernière reçoit et analyse toutes les données collectées par les agents « Host Monitor » et « LAN Monitor » de chaque hôte surveillé. Si « DIDS Director » est tombé en panne, tout le système sera bloqué (Figure II.14).

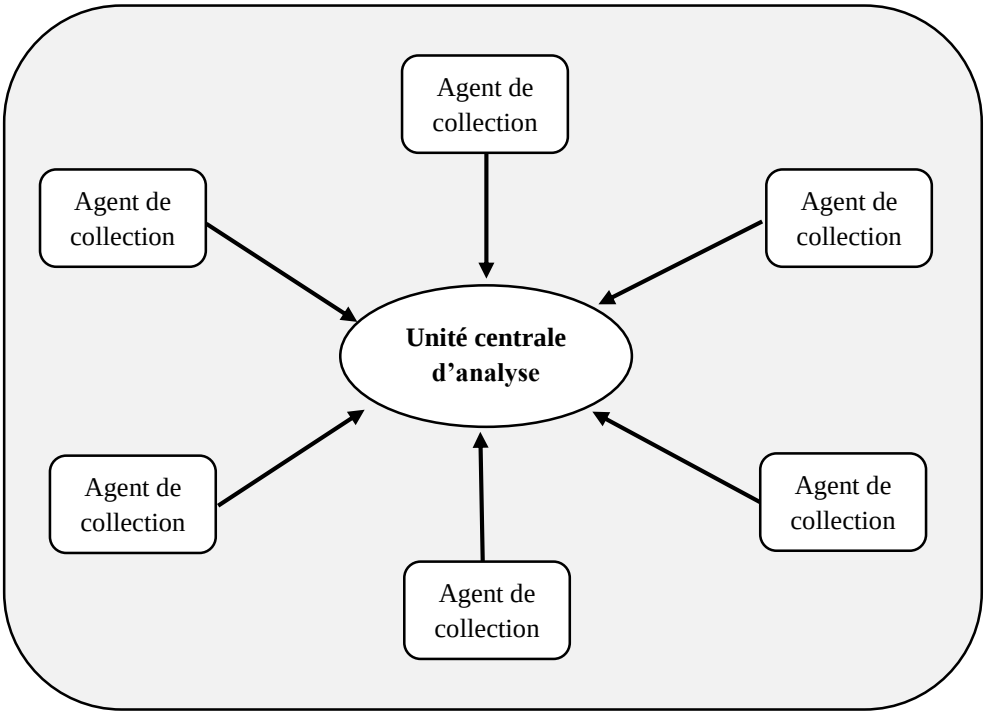

Figure II.14 - Architecture de détection d'intrusions centralisée

L'approche DIDS est préconisée pour surveiller un réseau local, mais à cause de sa nature centralisée, n'est pas une bonne solution dans les cas des réseaux à grande échelle où la communication avec l'unité centrale « DIDS Director » peut congestionner le réseau. L'approche AAFID traite ce problème, elle propose une architecture distribuée (au lieu monolithique) où plusieurs entités opèrent de façon coopérative pour la détection d'intrusion.

L'approche AAFID est représentée sous la forme d'un arbre (Figure II.15), où les données collectées parcourent les nœuds de l'arbre. A chaque niveau, ces données peuvent être traitées pour diminuer la quantité d'informations échangées dans le système. Cette approche est tolérante aux faute et extensible, mais les moniteurs représente le point faible dans l'approche. Si l'un des moniteurs s'arrête de fonctionner, tous les transceivers qu'il contrôle ne peuvent pas envoyer les informations nécessaires pour la détection. En plus l'architecture hiérarchique est difficile à mettre en place, ce qui représente une autre faiblesse.

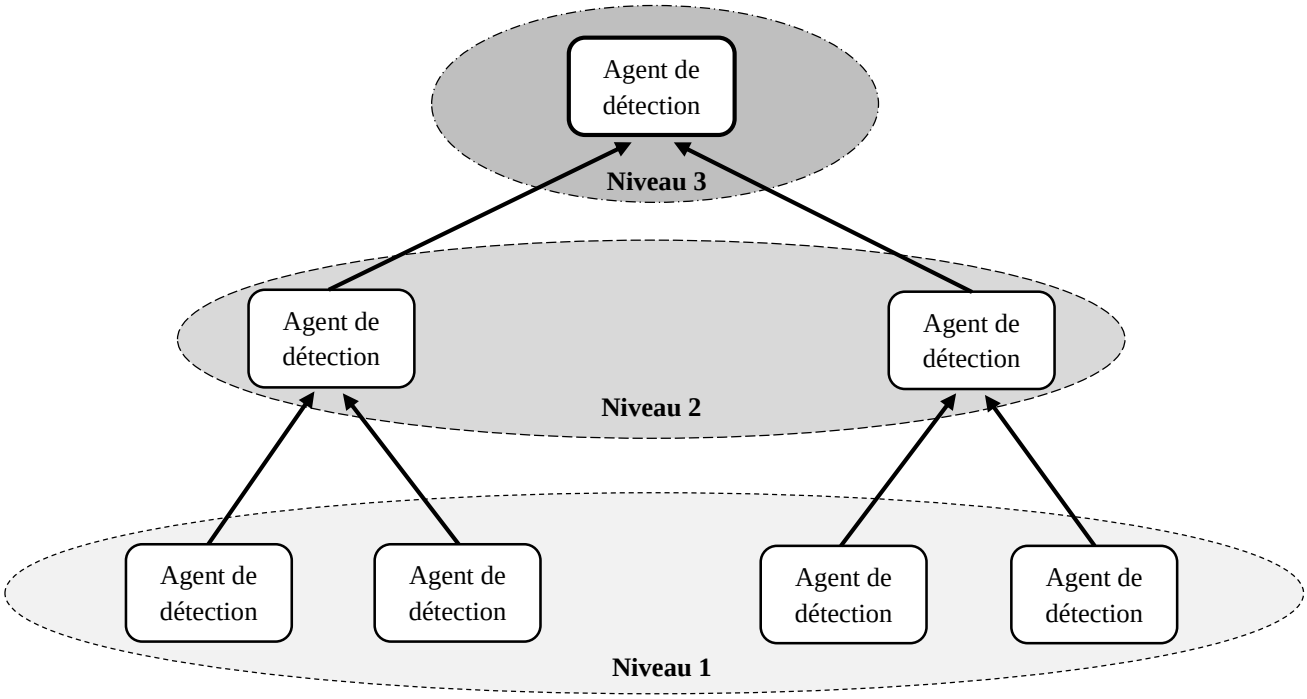

Figure II.15 - Architecture de détection d'intrusions hiérarchique

L'architecture MADIDF est complètement distribuée (Figure II.16) en raison d'utilisation des agents mobile. Ces agents sont capables de se déplacer vers les hôtes à surveiller pour collecter et analyser les données, ce qui réduit la charge sur le réseau et offre plus de sécurité et de flexibilité dans le système. Pour ces avantages nous allons adopter technologie d'agents mobiles pour concevoir notre approche de sécurité.

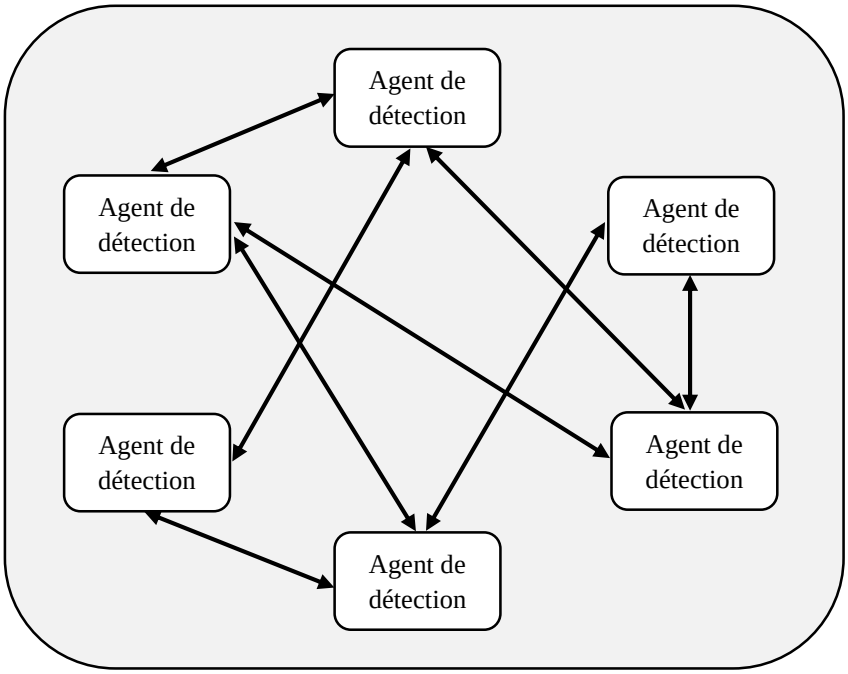

Figure II.16 - Architecture de détection d'intrusions complètement distribuée

L'inconvénient de l'approche MADIDF est qu'elle confie la tâche d'analyse et de traitement des données aux agents stationnaires (l'agent Analyseur et l'agent Manager) en se basant sur plusieurs bases de connaissances locales au niveau de chaque hôte surveillé, ce qui augmente potentiellement les messages d'échanges pour actualisé chaque bases de connaissance et par conséquent congestionner le réseau. Nous allons soulever ce problème dans notre conception (le chapitre ci-après) en confiant la tâche d'analyse des données aux agents mobiles.

II.9 Conclusion

Nous avons présenté au niveau de ce chapitre une vision générale sur les agents et les systèmes multi-agents. Ensuite nous avons décrit en détail la technologie d'agents mobiles. Par la suite, nous avons abordé le problème de sécurité dans les systèmes à base d'agents mobiles en mettant l'accent sur les différents types d'attaques possibles provenant des hôtes ou bien d'autres agents mobiles malveillants.

Nous avons vu ainsi, les travaux de recherche qui utilisent les agents mobiles pour améliorer les performances des systèmes de détections d'intrusions, avantages et lacunes de chacun. Ces approches montrent que les agents mobiles sont considérés comme un concept bien adapté pour les systèmes de détections d'intrusions grâce à leurs propriétés de réduire la charge sur le réseau, d'exécuter d'une façon asynchrone et autonome et de s'adapter dynamiquement aux environnements d'exécutions. Ces propriétés justifient le choix d'utiliser le concept d'agents mobiles dans les systèmes de détections d'intrusions fortement distribués et complexes.

Dans le chapitre suivant, nous allons proposer une approche de détection d'intrusion à base d'agents mobile.

Chapitre III. Conception d'une approche de sécurité basée agents mobile

III.1 Introduction

Nous avons vu précédemment l'avantage de l'utilisation des systèmes agents mobiles et leurs applications dans le domaine de sécurité et plus précisément dans les systèmes de détection d'intrusion. Dans le présent chapitre nous proposons une approche hybride distribuée (appelée AM-SID) à base d'agents mobiles pour la détection d'intrusion dans un système industriel distribué.

III.2 Caractéristiques de notre système de détection d'intrusion

Pour décrire d'une manière évidente les caractéristiques de notre système de détection d'intrusion proposé, nous les présentons dans la Figure III.1 (les caractéristiques choisies dans notre système sont présentés en gris).

III.2.1 Méthode de détection

La méthode de détection choisie dans la conception de notre système est la méthode de « détection par scénario », puisque dans les systèmes industriels distribués nous nous intéressons à la détection des attaques qui sont déjà maitrisées et bien déterminées.

L'autre méthode de détection (comportementale) a la possibilité de détecter les attaques inconnues, mais ce principe de détection n'est pas prouvé exact, ce qui rend cette méthode non adéquate dans notre cas où l'exactitude est un facteur important dans le domaine industriel. En

plus, cette méthode déclenche beaucoup de fausses alarmes (une fausse alarme arrive quand le système déclenche une alerte alors qu'il n'y a pas d'attaque).

Notons que dans notre architecture proposée (Figure III.3) la détection des intrusions est confiée aux deux agents mobiles analyseurs suivants :

- Agent Analyseur-NIDS : analyse le trafic réseau.
- Agent Analyseur-HIDS : analyse les fichiers d'audit système.

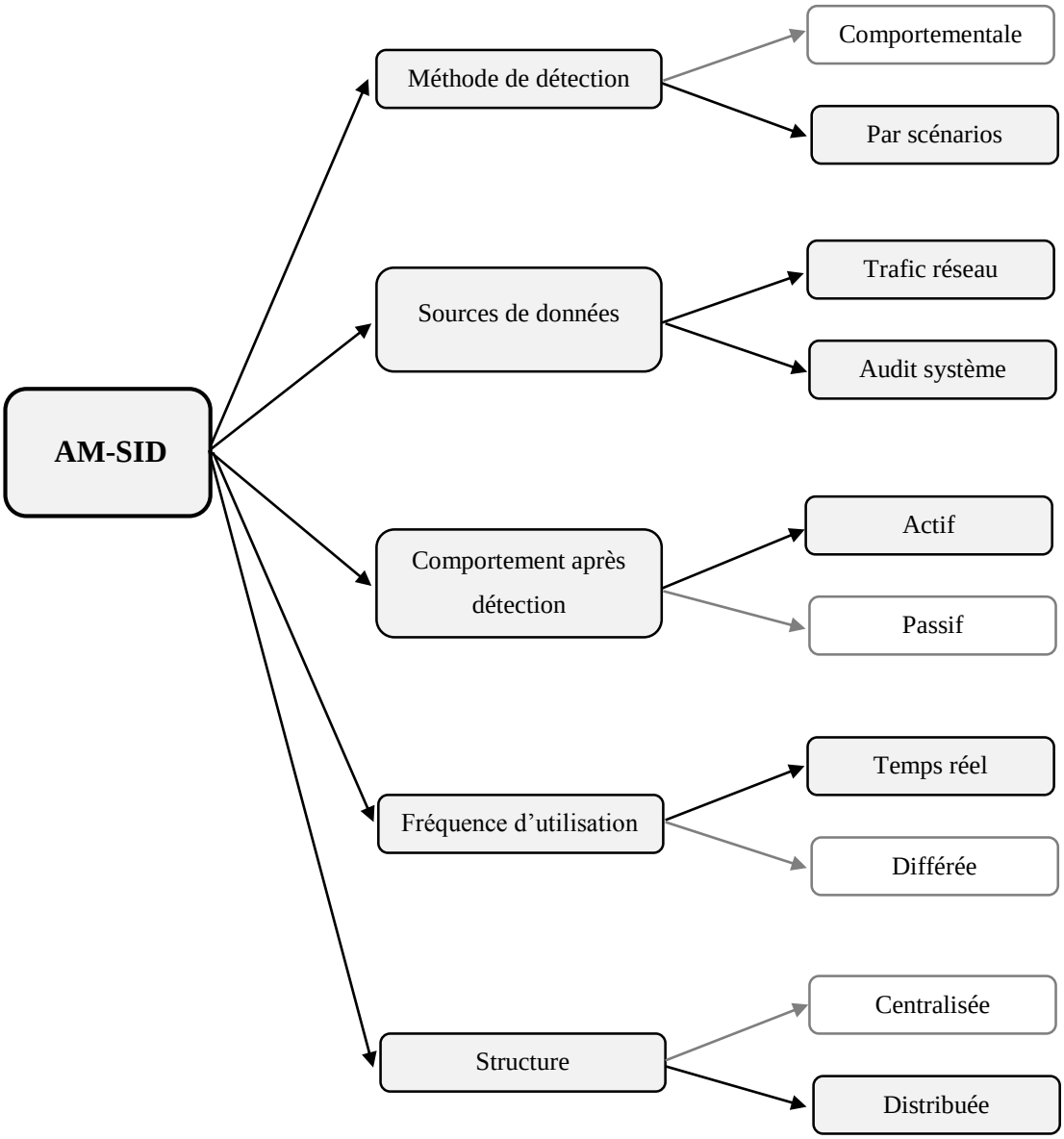

Figure III.1 – Caractéristiques de notre approche (AM-SID)

III.2.2 Sources de données

Les sources de données représentent l'interface entre le système surveillé et le système de détection d'intrusion. Ils jouent un rôle très important dans le mécanisme de détection.

Pour minimiser le maximum possible des risques dans notre système industriel, nous avons proposé une approche de détection hybride qui utilise les deux types de sources de données qui sont : NIDS (trafic réseau) et HIDS (fichiers d'audit du système surveillé).

Dans notre architecture proposée (Figure III.3) la collecte des informations est confiée aux deux agents stationnaires suivants :

- Collecteur-NIDS : il collecte et formate les paquets réseaux.
- Collecteur-HIDS : qui collecte et prépare les fichiers d'audits du système surveillé.

III.2.3 Comportement après détection

Les systèmes industriels distribués nécessitent souvent des réactions et des contre-mesures automatiques après une faille dans le système ou après la détection d'une attaque. Par exemple, régulation automatique de température, arrêter une machine ou déconnecter un utilisateur illégitime. C'est pour ça que notre approche se base sur un comportement actif après la détection d'une attaque, mais dans des cas spéciaux la réaction peut se limiter à une alerte d'avertissement.

Au niveau de notre architecture proposée (Figure III.3) c'est l'agent Actionneur qui est chargé de cet aspect.

III.2.4 Fréquence d'utilisation

Dans un système industriel une attaque non détecter au bon moment, peut lui provoquer des dégâts fatales de point de vues matériels et logiciels. Une détection rapide de cette attaque est nécessaire pour la contrer dans les brefs délais. Pour cette raison nous avons adopté dans notre approche une surveillance en temps réel en utilisant des agents mobiles qui se déplacent d'un hôte à l'autre en permanence et sans cesse pour détecter des attaques et des intrusions le plus tôt possible.

III.3 Technique de recherche de motif

Les techniques de détection d'intrusions représentent le noyau d'un système de détection d'intrusion, parce que le processus de détection consomme la majorité du temps d'exécution.

Dans notre système nous utilisons pour l'analyse des données la technique de « recherche de motif » (ou *Pattern matching* en Anglais) qui appartient aux techniques basées sur les signatures. Ces techniques représentent les attaques sous forme de signatures dans une base de connaissance, ensuite ces signatures seront comparées avec les données collectées. S'il y a des correspondances, une alerte est déclenchée.

La technique de *recherche de motif* permet à notre système de détection d'intrusion de trouver les informations à chercher le plus rapidement possible. Cette technique est représentée dans la Figure III.2.

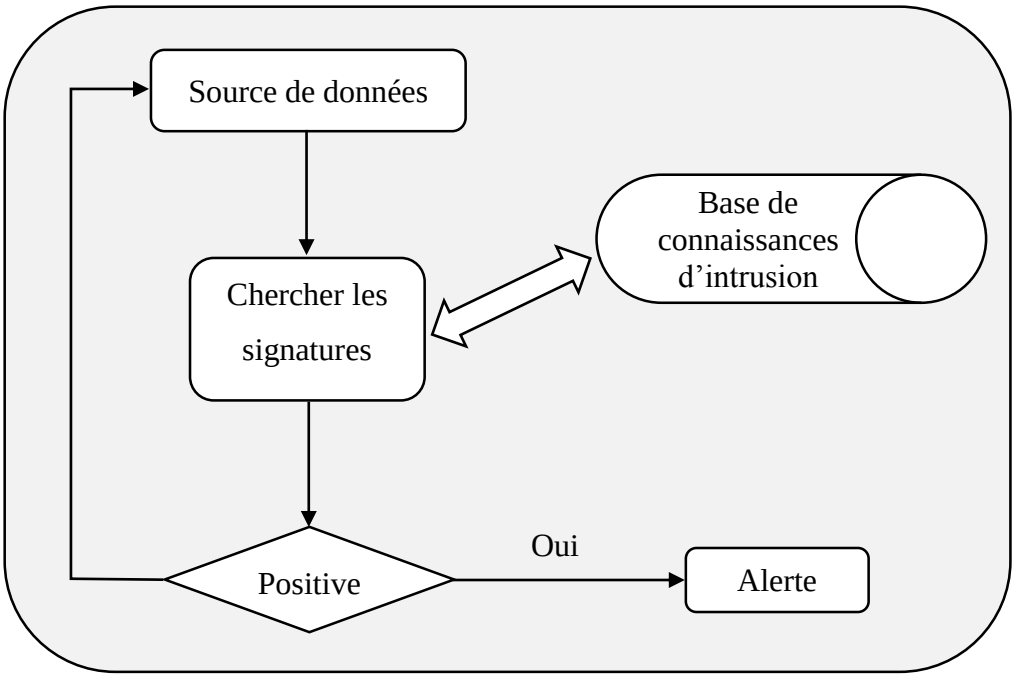

Figure III.2 - Organigramme de la technique de Recherche de Motif

Il existe plusieurs algorithmes de recherche de motif comme l'algorithme de *Boyer-Moore* (BM) qui est intéressant et plus efficace. Nous utilisons cet algorithme pour analyser les données courantes du système.

III.4 Architecture globale de l'approche AM-SID

La Figure III.3 présente l'architecture générale de notre approche de sécurité à base d'agents mobiles pour la détection d'intrusion dans un système industriel distribué (AM-SID). L'architecture utilise à la fois des agents stationnaires et des agents mobiles :

- Agents stationnaires : Agent Administrateur, Agent de Crise, Agent Actionneur, Agent Collecteur-NIDS et Agent Collecteur-HIDS.
- Agents mobiles : Agent Analyseur-NIDS et Agent Analyseur-HIDS.

Notre approche utilise, comme indiqué dans la Figure III.3, deux bases de données : BD_NIDS et BD_HIDS. La première contient les signatures d'attaques du réseau et l'autre contient les signatures d'attaques de l'hôte. Ces bases de données contiennent aussi, les réactions et les mesures à prendre dans le cas de détection d'une attaque.

Le capteur (ou Sniffer) est utilisé pour surveiller et capturer le trafic sur le réseau et de reporter à l'Agent Collecteur-NIDS les informations liées aux réseaux.

La communication entre les différents agents est asynchrone, où un agent envoie un message et continue son travail. Nous décrivons en détail les différents agents utilisés dans notre architecture.

III.4.1 Agent Administrateur

Comme son nom l'indique, c'est l'agent maître du système. Parmi ses rôles, nous pouvons citer :

- Création et destruction des différents agents,
- Interaction avec les différents agents,
- Mise à jour des deux bases de données,
- Gestion des alarmes,
- Détection des connexions.

Le diagramme de classe correspondant à cet agent est représenté dans la Figure III.4

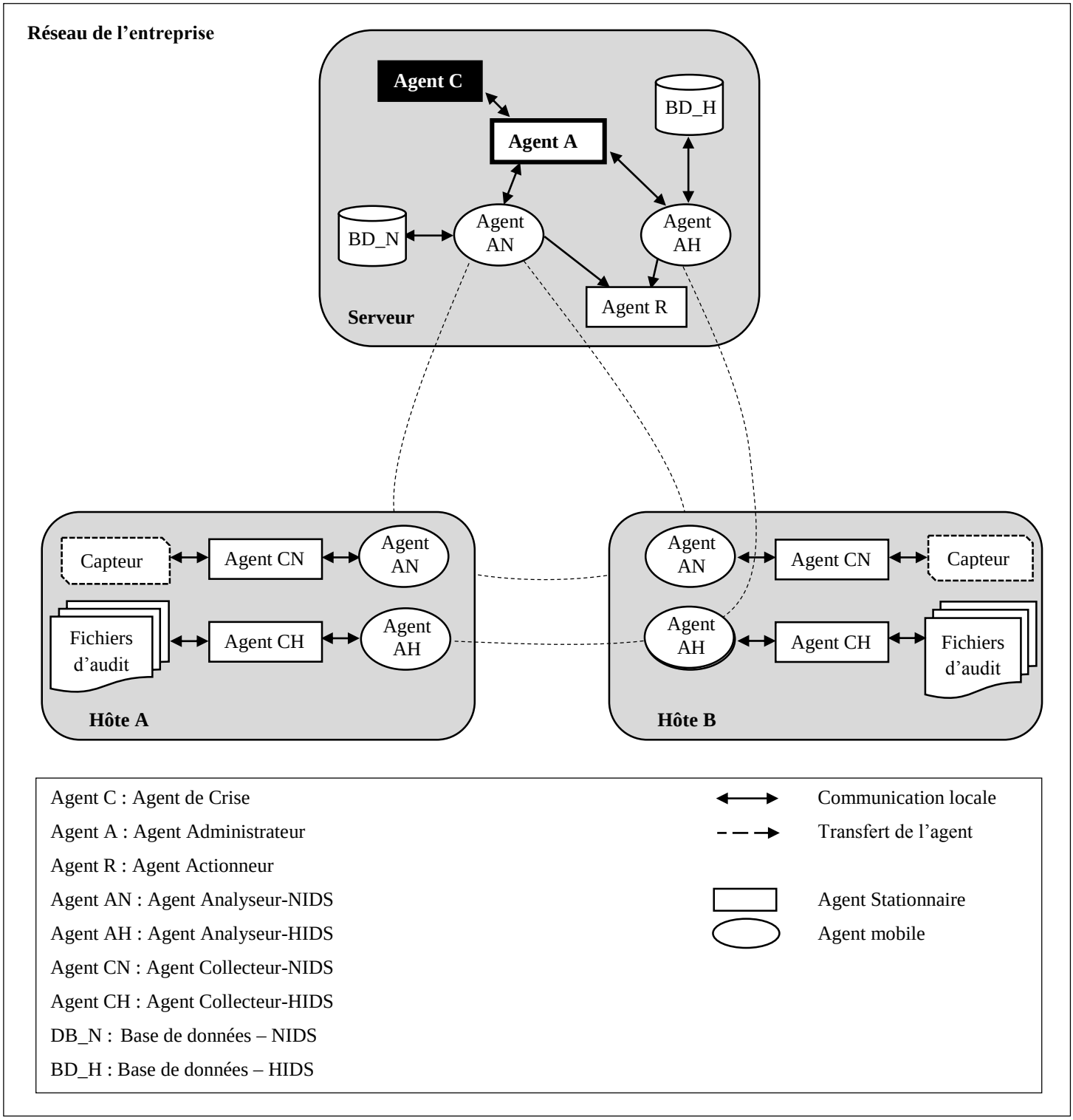

Figure III.3 - Architecture générale de l'approche AM-SID

III.4.2 Agent de Crise

Le rôle de l'agent de Crise est d'assurer un service minimum et de réinitialiser le système en cas de crise. Si l'agent Administrateur tombe en panne, l'agent de Crise réagit immédiatement et crée un nouvel agent Administrateur pour récupérer toutes les informations du premier agent Administrateur.

Le diagramme de classe de cet agent est représenté sur la Figure III.5.

III.4.3 Agents Collecteurs

Pour la collecte des informations et les préparer pour l'analyse, notre système utilise deux agents stationnaires collecteurs pour chaque hôte surveillé, Agent Collecteur-NIDS et Agent Collecteur-HIDS. Le premier agent collecte les paquets réseaux capturés par le Sniffer (ou Capteur) et l'autre agent chargé de collecter les fichiers d'audits de l'hôte surveillé.

Les diagrammes de classes correspondants à ces agents sont illustrés dans les Figure III.6 et Figure III.7

III.4.4 Agents Analyseurs

Les deux agents analyseurs (Analyseur-NIDS et Analyseur-HIDS) sont des agents mobiles qui peuvent se déplacer entre les hôtes ; ils analysent les données pour détecter les tentatives d'attaques. Ces agents décident s'il y a une attaque ou intrusion en se basant sur les deux bases de données utilisées dans notre approche.

Le premier agent analyse les données collectées par l'agent Collecteur-NIDS en utilisant la base de données des signatures d'attaques BD_NIDS. L'autre agent (Analyseur-HIDS) analyse les données collectées par l'agent Collecteur-HIDS en utilisant la base de données des signatures d'attaques BD_HIDS.

Pour détecter les intrusions, les agents analyseurs utilisent (comme nous avons vu précédemment) la technique de recherche de motif (Pattern matching) pour analyser le trafic réseaux et les fichiers d'audit du système. Si l'un des agents détecte une attaque, il envoie un message d'alerte à l'agent Actionneur.

Les diagrammes de classes correspondants aux deux agents Analyseur-NIDS et Analyseur-HIDS sont représentés successivement dans les Figure III.8 et Figure III.9.

III.4.5 Agent Actionneur

Quand l'un des agents analyseurs détecte une attaque, il envoie un message d'alerte à l'agent Actionneur qui prend les réactions et les mesures nécessaires pour contrer l'attaque détectée. Cet agent intervient généralement après la détection d'une attaque. Si l'attaque n'a pas pu être contrée, cet agent déclenche une alarme appelant l'intervention humaine.

Le diagramme de classe de cet agent est illustré dans la Figure III.10.

III.4.6 Classe Utilisateur

La classe Utilisateur est ajoutée pour l'authentification des utilisateurs lorsqu'ils se connectent. Elle représente l'officier de sécurité du système.

Cette classe est représentée dans la Figure III.11.

III.5 Modélisation de notre approche

Le langage de modélisation unifié UML (Agent Unified Modelling) est insuffisant pour modéliser les systèmes multi-agents et agents mobiles, c'est pour cette raison qu'OMG (Object Management Group) et FIPA (Foundation for Intelligent Physical Agents) ont proposés AUML (Agent Unified Modelling Language) comme une extension du standard UML pour la modélisation des systèmes à base d'agents.

III.5.1 Diagrammes de classes

Les diagrammes de classes présentent la structure interne en fournissant une représentation abstraite des agents du système. Ces diagrammes contiennent les attributs de l'agent, ses actions, ses méthodes et les messages entrants et sortants.

Dans ce qui suit, nous allons décrire le diagramme de classe correspond à chaque agent dans notre système ainsi que le diagramme de classe de la classe Utilisateur. Voici ces diagrammes de classes :

III.5.1.1 Agent administrateur

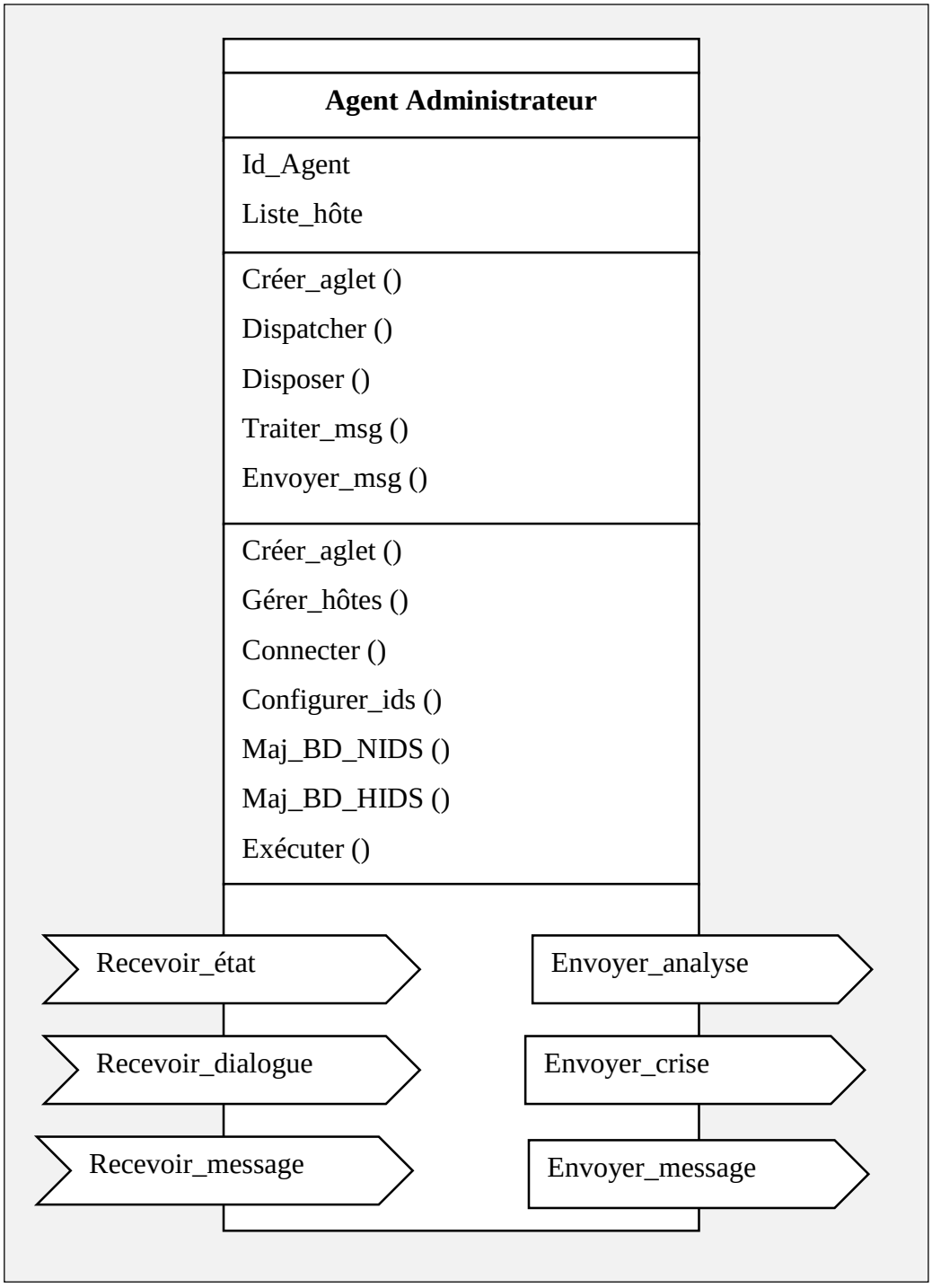

Figure III.4 - Diagramme de classe de l'Agent Administrateur

III.5.1.2 Agent de Crise

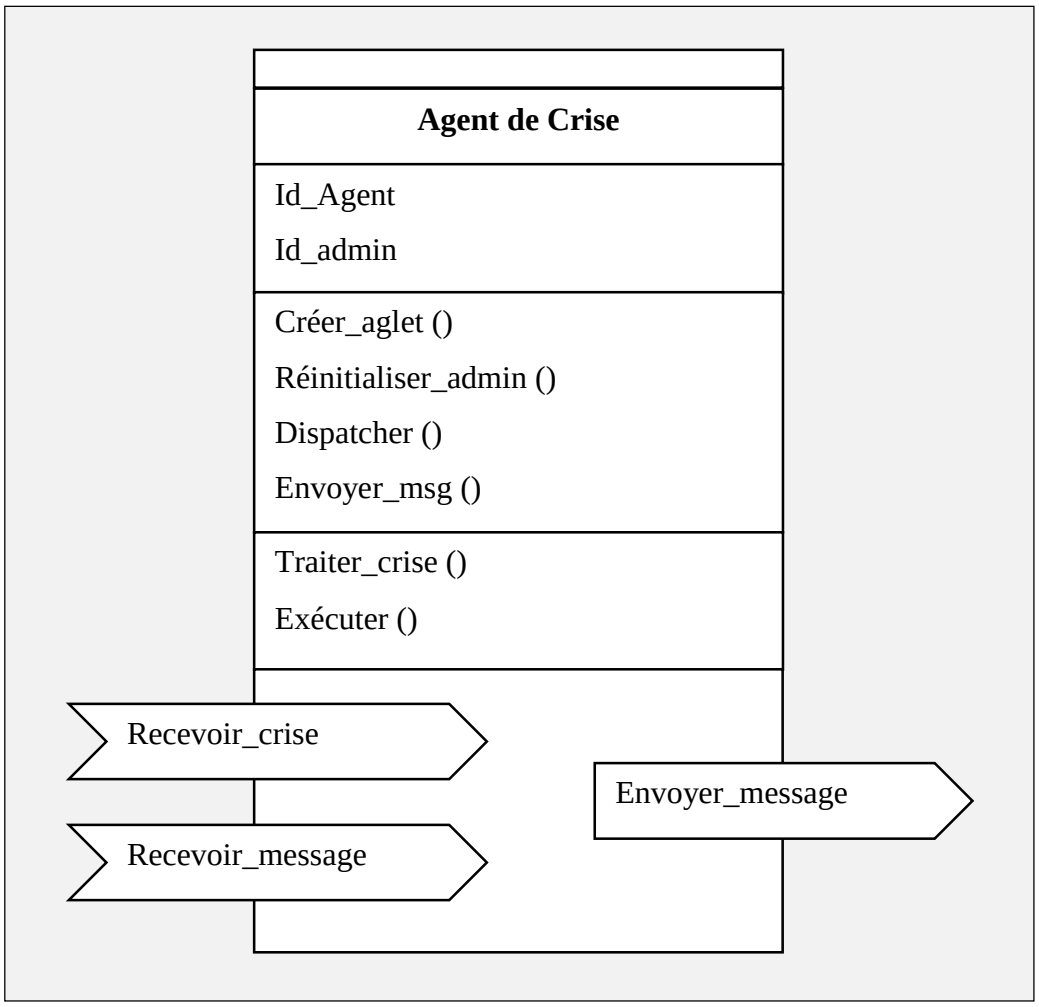

Figure III.5 - Diagramme de classe de l'Agent de Crise

III.5.1.3 Agent Collecteur-NIDS

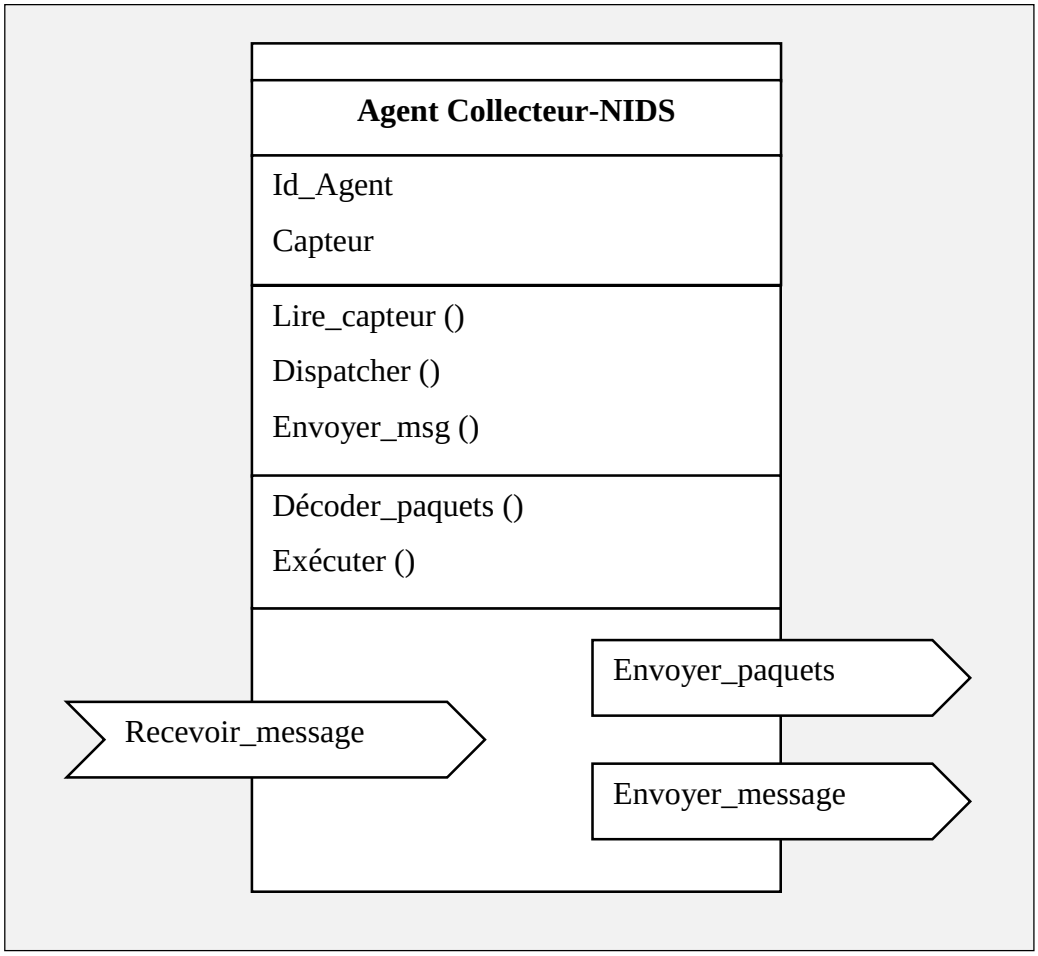

Figure III.6 - Diagramme de classe de l'Agent Collecteur-NIDS

III.5.1.4 Agent Collecteur-HIDS

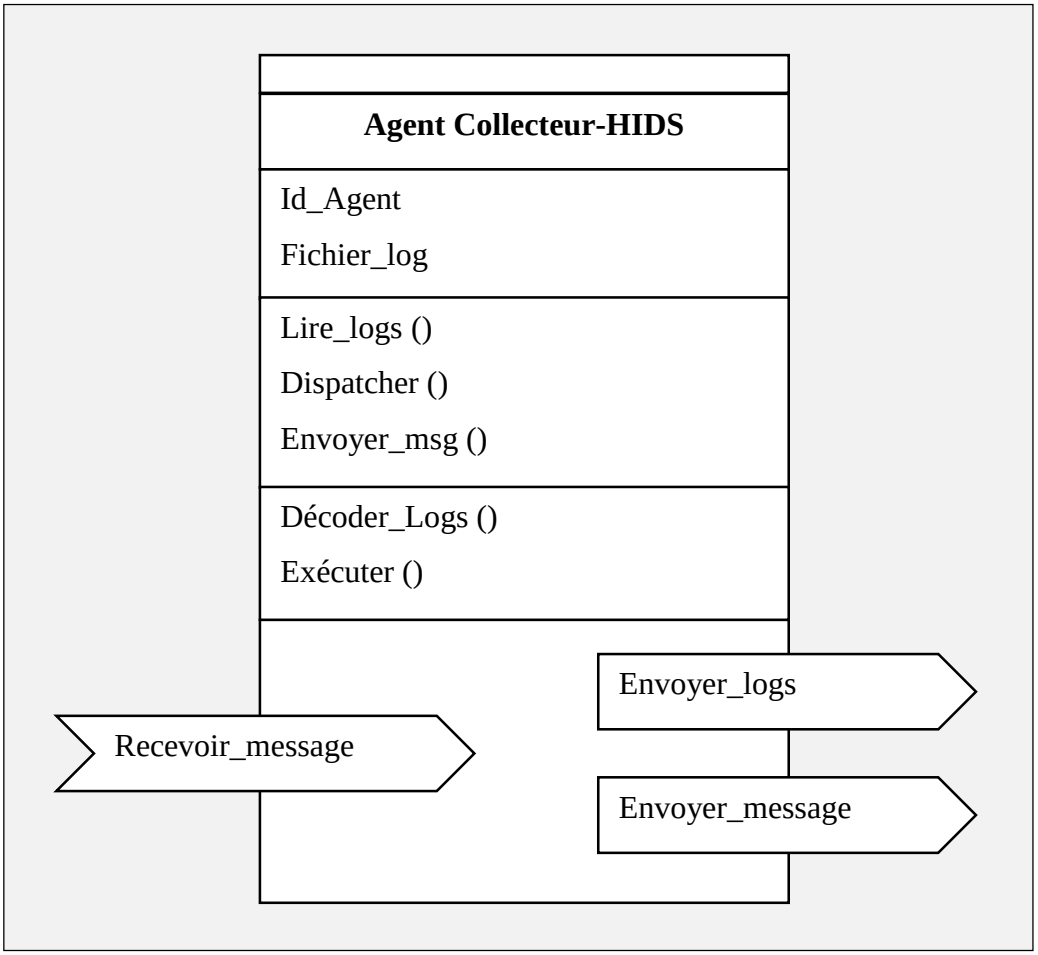

Figure III.7 - Diagramme de classe de l'Agent Collecteur-HIDS

III.5.1.5 Agent Analyseur-NIDS

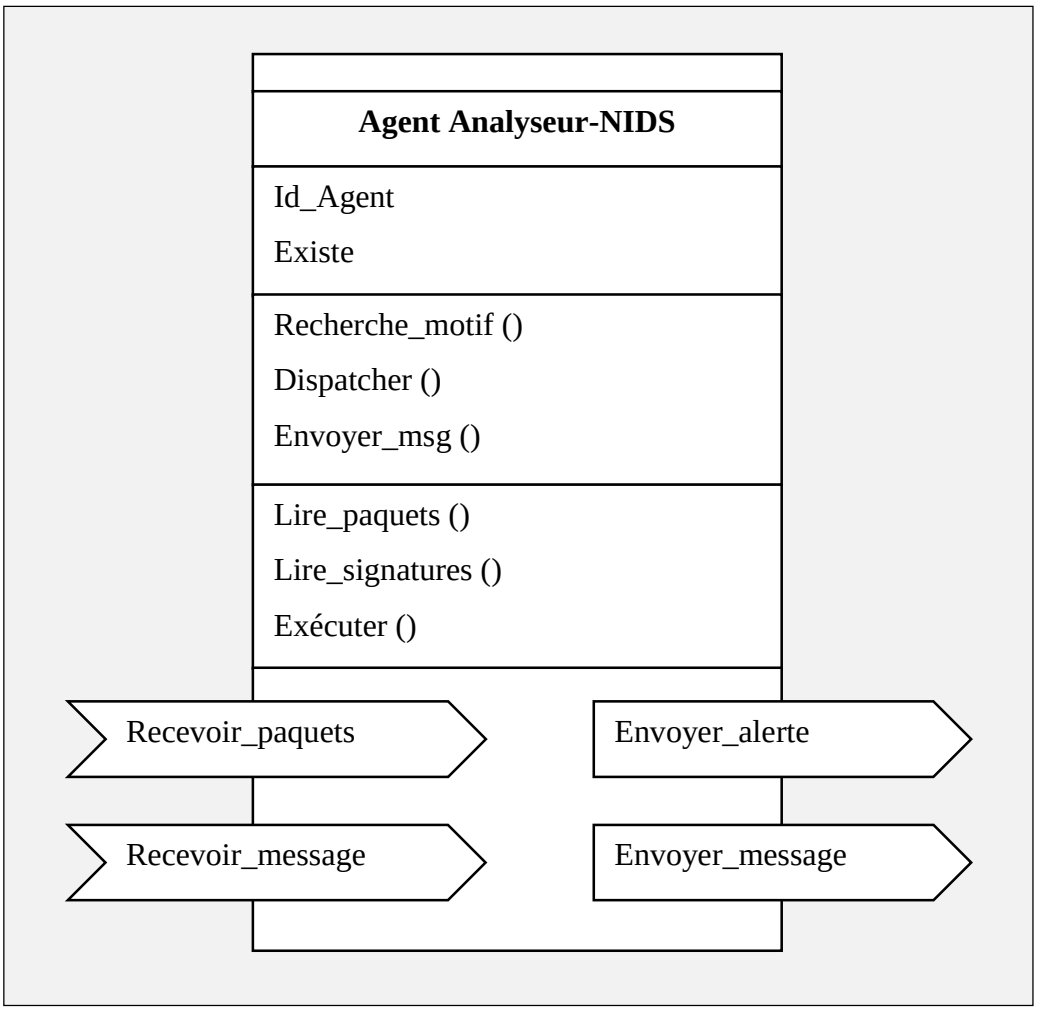

Figure III.8 - Diagramme de classe de l'Agent Analyseur-NIDS

III.5.1.6 Agent Analyseur-HIDS

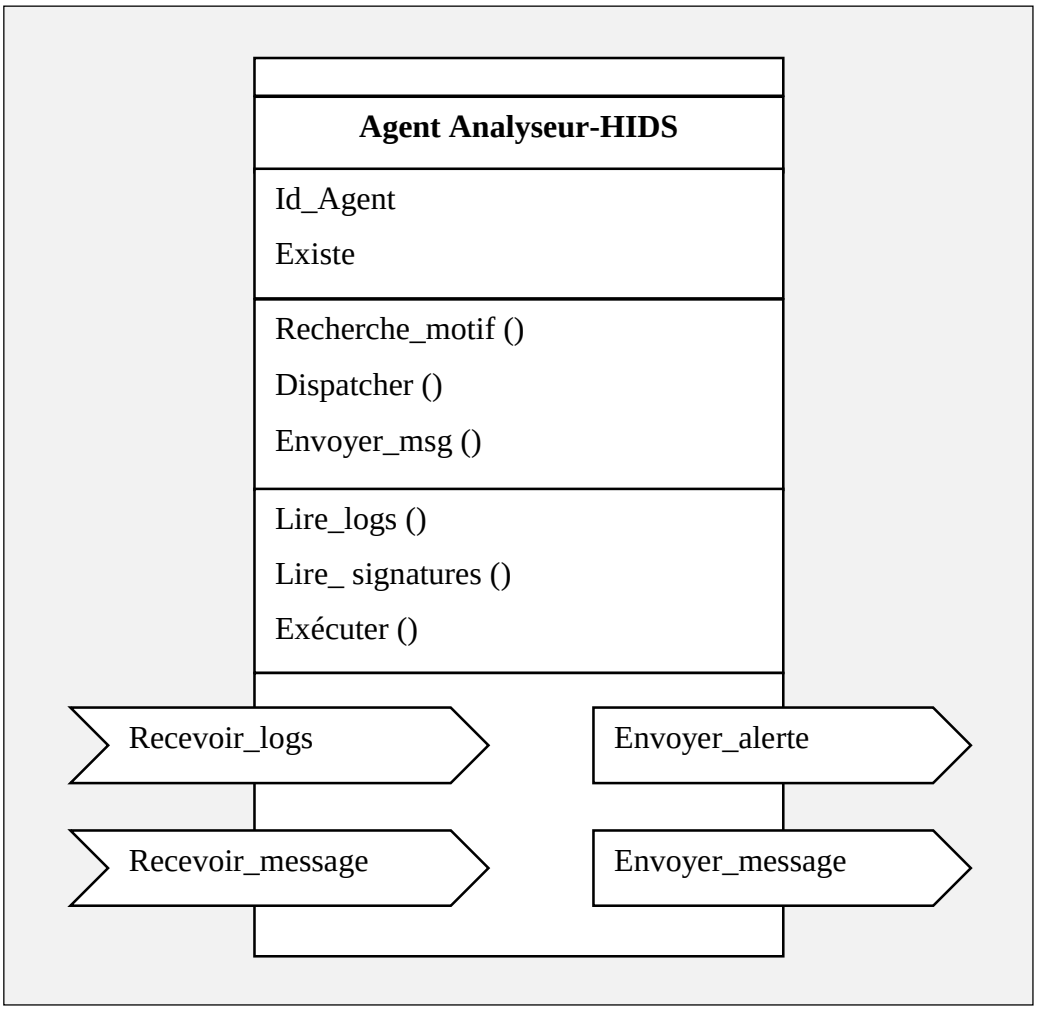

Figure III.9 - Diagramme de classe de l'Agent Analyseur-HIDS

III.5.1.7 Agent Actionneur

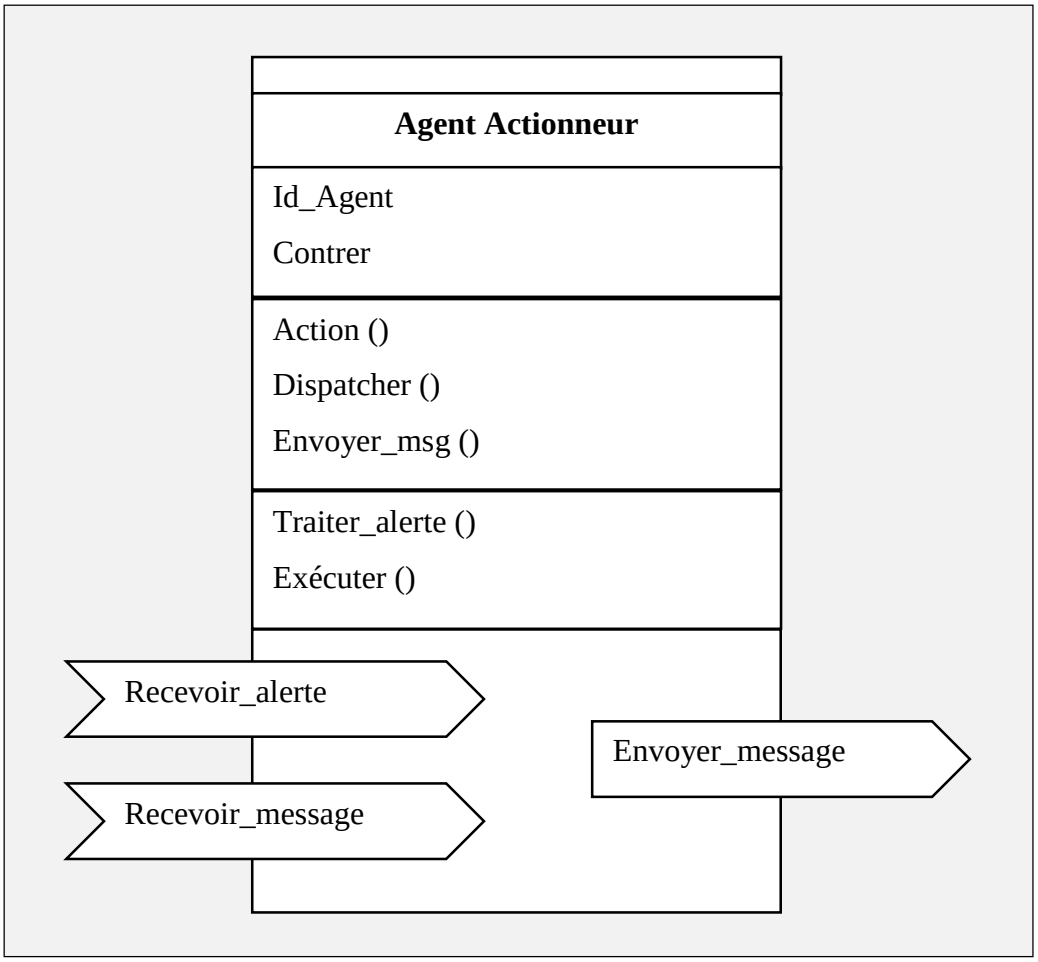

Figure III.10 - Diagramme de classe de l'Agent Actionneur

III.5.1.8 Classe Utilisateur

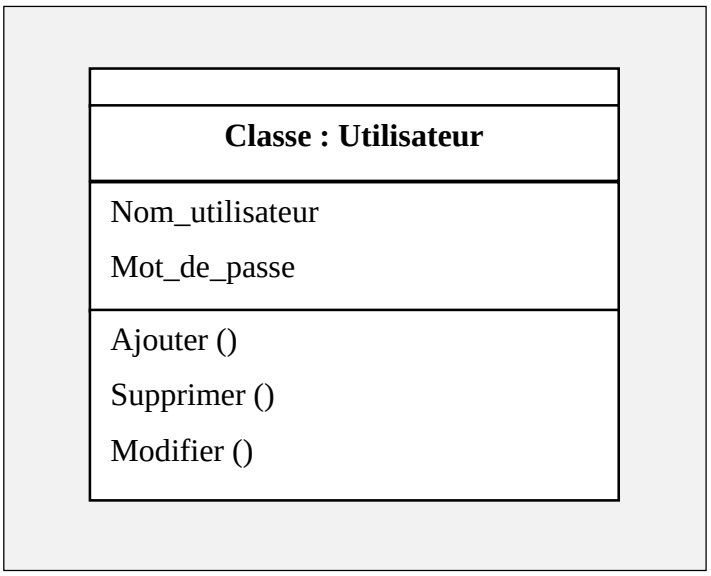

Figure III.11 – La classe Utilisateur

III.5.1.9 Diagramme de classe globale

La Figure III.12 présente le diagramme de globale de notre système. Ce diagramme montre les associations entre les différentes classes utilisées.

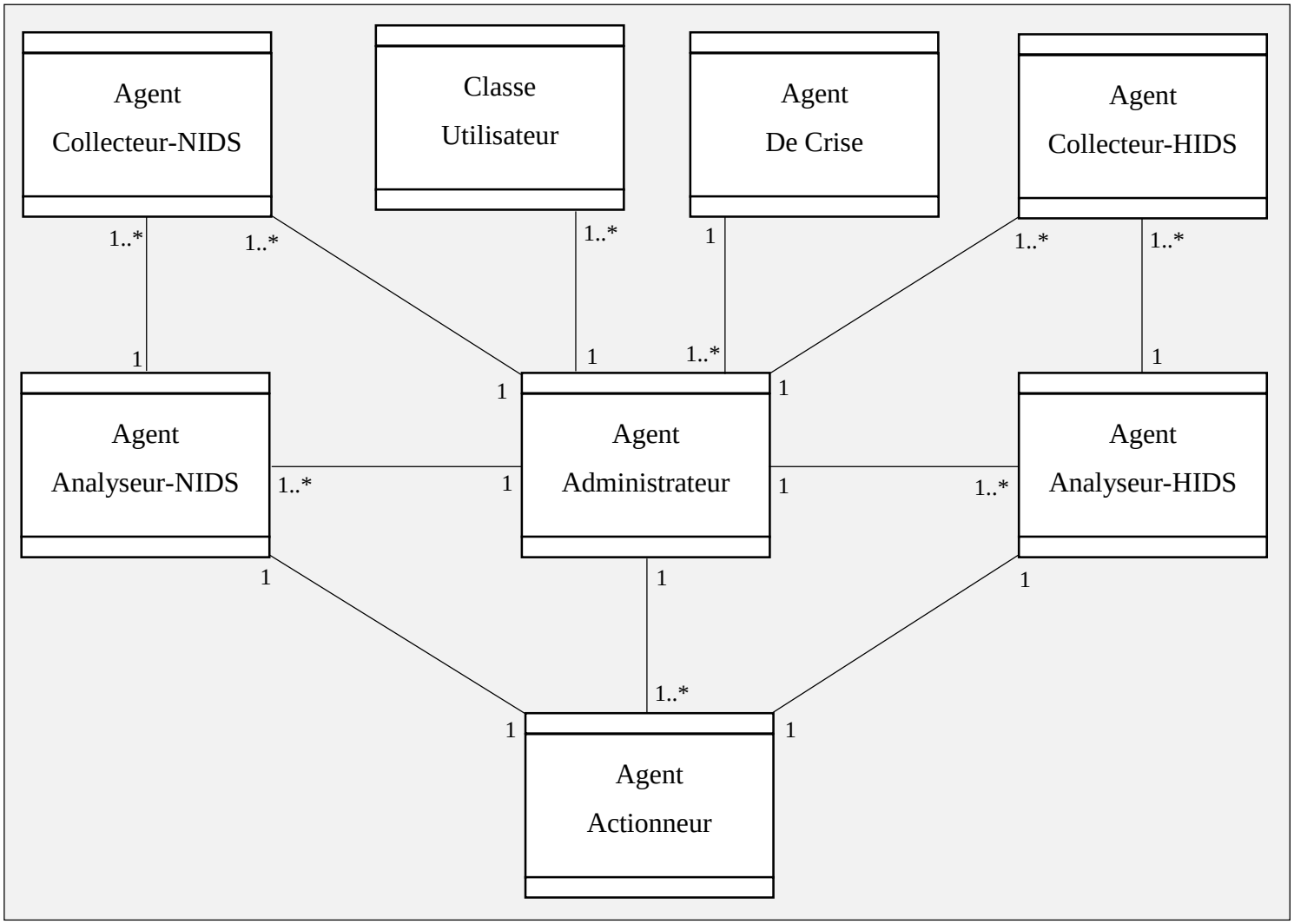

Figure III.12 - Diagramme de classe du système

III.5.2 Diagrammes de séquences

Les diagrammes de séquences montrent explicitement les interactions pouvant intervenir entres les différents agents du système en précisant la chronologie de ces interactions. Nous présentons ci-dessous les principaux diagrammes de séquences de notre système.

III.5.2.1 Diagramme de séquence de collecte des informations

Les étapes ci-après expliquent le diagramme de séquence de collecte des informations illustré dans la Figure III.13 :

- Un utilisateur demande de se connecter (*Connexion*).
- Si l'utilisateur est connu, il se connecte avec son mot de passe (*Connecter ()*).
- Après la connexion de l'utilisateur, l'agent administrateur récupère l'adresse IP de l'utilisateur et met à jour la liste des hôtes connectés (*Gérer_hôtes ()*).
- L'agent Administrateur demande (*Demande_collecter*) aux agents collecteurs (Collecteur-NIDS et Collecteur-HIDS) de commencer la collecte des informations (paquets et fichiers d'audit) concernant la station détectée.
- L'agent Collecteur-NIDS récupère les paquets capturés par le Sniffer (*Lire_capteur ()*) et les prétraite (*Décoder_paquets ()*) pour l'analyse.
- L'agent Collecteur-HIDS récupère les fichiers d'audit (*Lire_logs ()*) et les prétraite (*décoder_logs ()*) pour l'analyse.
- Les deux agents Collecteurs informent (*Réponse_collecter ()*) l'agent Administrateur que la collecte des informations est terminée.

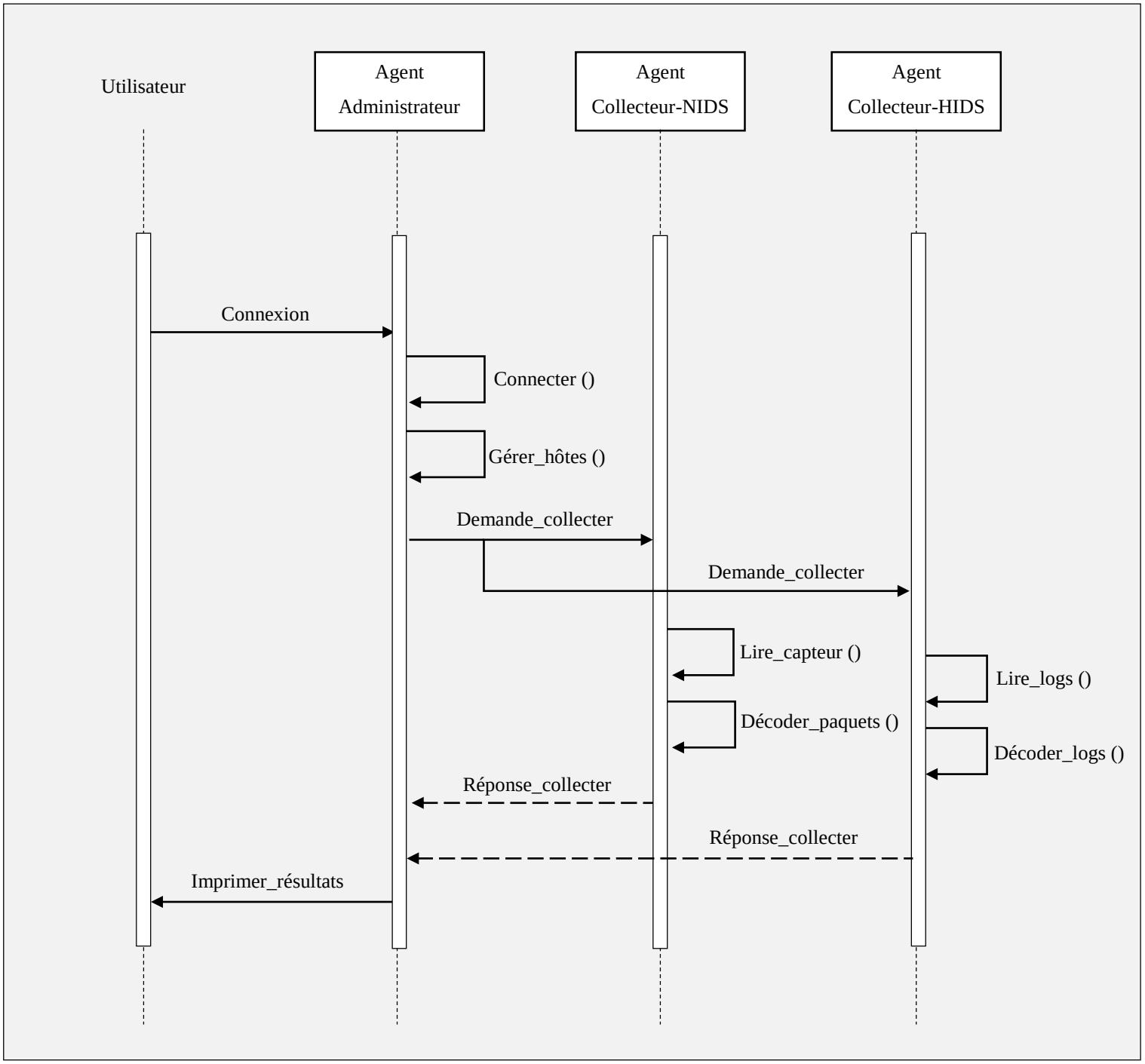

Figure III.13 - Diagramme de séquence de collecte des informations

III.5.2.2 Diagramme de séquence d'analyse et de détection d'intrusion

Les étapes ci-dessous décrivent le diagramme de séquence d'analyse et de détection d'intrusion représenté dans la Figure III.14 :

- Un utilisateur demande de se connecter (*Connexion*).
- Si l'utilisateur est connu, il se connecte avec son mot de passe (*Connecter ()*).
- Après la connexion de l'utilisateur, l'agent administrateur récupère l'adresse IP de l'utilisateur et met à jour la liste des hôtes connectés (*Gérer_hôtes ()*).
- L'agent administrateur demande (*Demande_Collecte*) aux agents mobiles Analyseurs (Analyseur-NIDS et Analyseur-HIDS) de commencer l'analyse des informations (paquets et fichiers d'audit) collectées par les agents Collecteurs (Collecteur-NIDS et Collecteur-HIDS).
- L'agent Analyseur-NIDS lit les paquets (*Lire_paquets ()*) collectés par l'agent Collecteur-NIDS et les compare (*Recherche_motif ()*) avec la liste des signatures d'attaque qui se trouvent au niveau de la base de données BD_NIDS (*lire_signatures ()*).
- L'agent Analyseur-HIDS lit les fichiers d'audit (*Lire_logs ()*) collectés par l'agent Collecteur-HIDS et les compare (*Recherche_motif ()*) avec la liste des signatures d'attaque qui se trouvent au niveau de la base de données BD_HIDS (*lire_signatures ()*).
- Lorsque l'un des deux agents Analyseurs détecte une attaque, un message d'alerte est envoyé à l'agent Actionneur (*Envoyer_alerte*).
- L'agent actionneur traite les messages d'alertes envoyés par les agents Analyseurs (*Traite_alerte ()*).

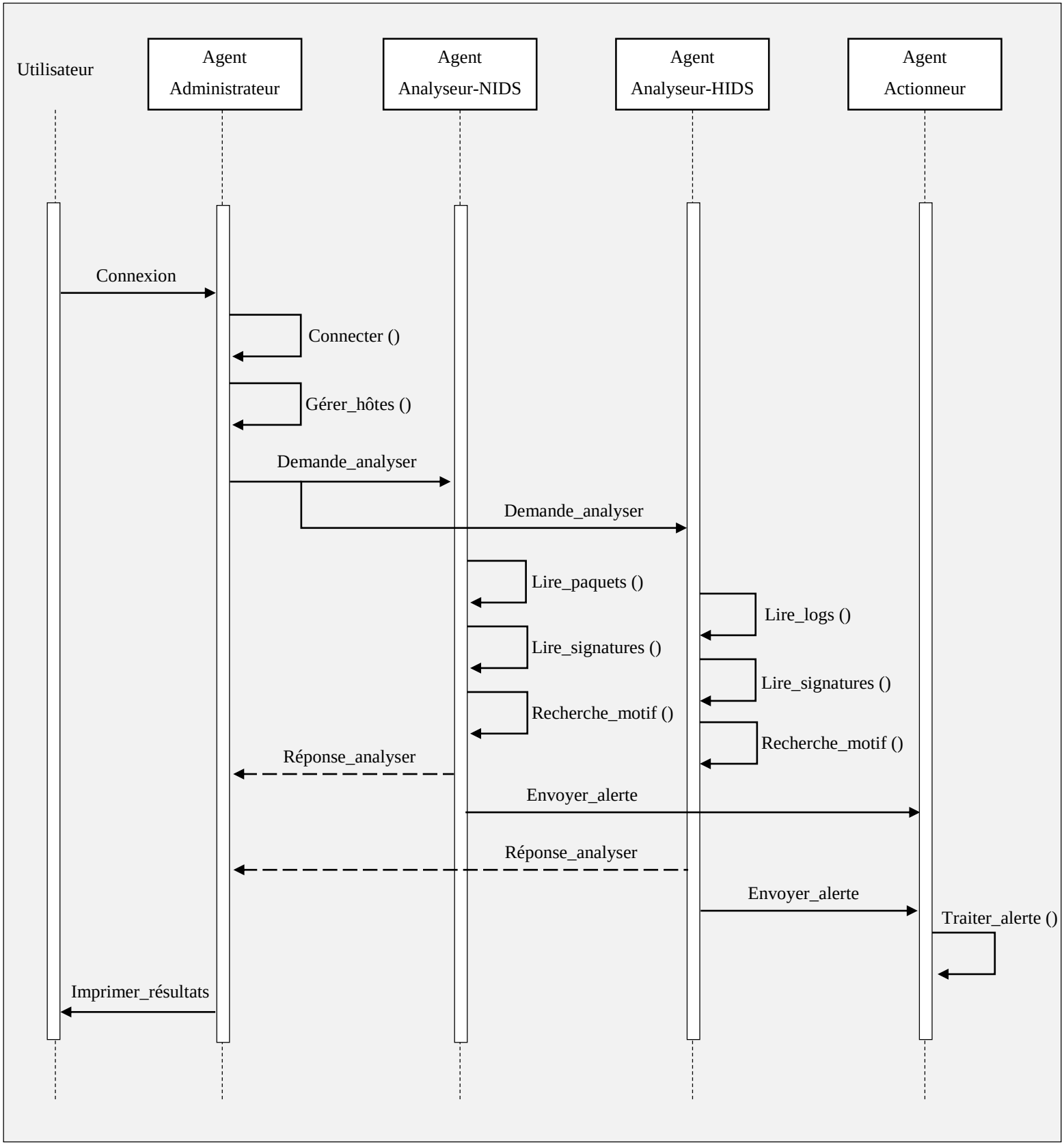

Figure III.14 - Diagramme de séquence d'analyse et de détection d'intrusion

III.5.2.3 Diagramme de séquence de réinitialisation de l'agent Administrateur

Les étapes ci-après décrivent le diagramme de séquence de réinitialisation de l'agent Administrateur (le diagramme est représenté dans la Figure III.15) :

- En cas de problème lié à l'état de l'agent Administrateur, ce dernier envoie un message de crise avec son état à l'agent de Crise (*Envoyer_crise*).
- L'agent de Crise traite l'urgence (*Traiter_crise ()*), puis réinitialise l'agent Administrateur (*Réinitialiser_admin ()*).
- L'agent de Crise envoie ensuite l'état de l'agent Administrateur (*Envoyer_état*).

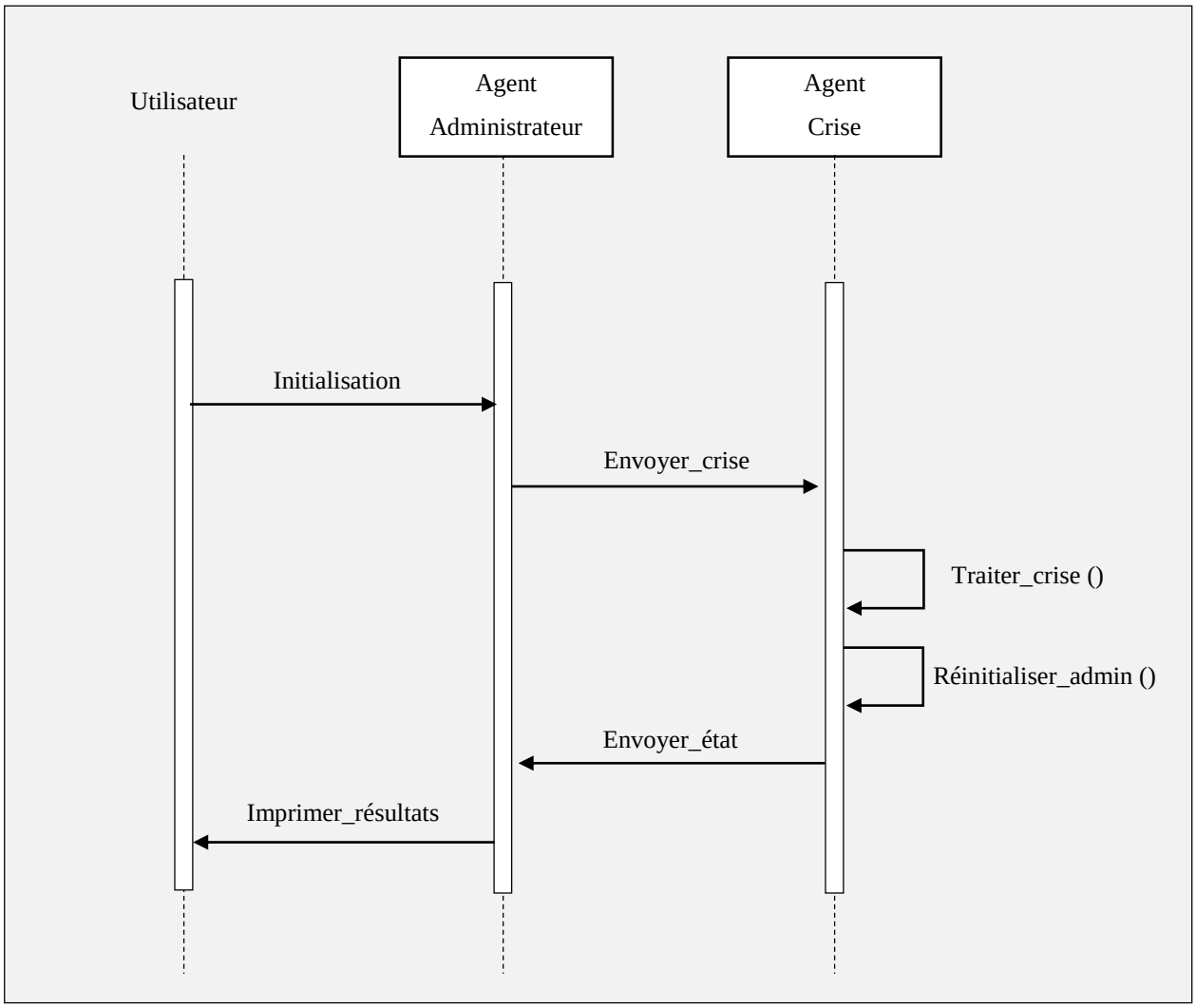

Figure III.15 - Diagramme de séquence de réinitialisation de l'agent Administrateur

III.6 Conclusion

Dans ce chapitre nous avons abordé la conception de notre approche de sécurité basée agent mobile (AM-SID), où on a vu l'architecture générale du système ainsi que les détails de chaque composant. Par la suite nous avons modélisé notre système avec le standard de modélisation « AUML ».

La phase de conception étant terminée, nous allons valider par la suite, notre approche par l'implémenter d'une application logicielle qui étude un cas de notre système. Enfin, nous démontrons quelques GUI (interfaces graphiques) de cette application.

Chapitre IV. Etude de cas et validation

IV.1 Introduction

Après une étude détaillée sur les systèmes industriels distribués et la technologie d'agents mobiles, et après avoir expliqué en détail notre approche à base d'agents mobile (AM-SID) pour sécuriser ces systèmes, nous sommes arrivés à la phase finale de notre travail qui consiste à la mise en place de notre application.

Dans ce chapitre nous allons implémenter les différents agents proposés dans notre approche et tous les modules nécessaires au bon fonctionnement du système.

IV.2 Environnement de travail

IV.2.1 Outils de développement

IV.2.1.1 Le langage Java

Pour la mise en œuvre de notre système, nous allons choisir le langage de programmation orienté objet « Java » développé par Sun Microsystems. Ce langage a réussi à intéresser beaucoup de développeurs à travers le monde. En effet, Java est un langage multiplateforme disposant d'une machine virtuelle appelée JVM (Java Virtual Machine) lui permettant de s'exécuter sur n'importe quelle machine. Java est capable de tourner aussi bien sur un PC que sur un MAC, sur un téléphone ou encore sur une carte à puce.

Pour programmer avec java dans notre application, nous utilisons la version 8.0.250 du JDK (Java Development Kit).

IV.2.1.2 L'IDE NetBeans

NetBeans (Web-NetBeans, 2014) est un projet open source ayant un succès et une base d'utilisateur très large, une communauté en croissance constante, et près 100 partenaires mondiaux et des centaines de milliers d'utilisateur à travers le monde. Sun Microsystems a fondé le projet open source NetBeans en Juin 2000 et continue d'être le sponsor principal du projet.

L'EDI NetBeans est un environnement de développement et un outil pour les programmeurs pour écrire, compiler, déboguer et déployer des programmes. Il est écrit en Java mais peut supporter n'importe quel langage de programmation. Il y a également un grand nombre de modules pour étendre l'EDI NetBeans. L'EDI NetBeans est un produit gratuit, sans aucune restriction quant à son usage.

Dans notre implémentation nous allons utiliser la version actuelle de l'IDE NetBeans (NetBeans IDE 8.0.1).

IV.2.1.3 La plateforme Aglet

Aglet (Agent et Applet) est la plateforme choisi dans le cadre de ce mémoire pour implémenter notre approche de détection d'intrusion. Cette plateforme est basée sur java et a proposé pour le développement des agents mobiles. Elle a été développée par une équipe de chercheurs du laboratoire de recherche d'IBM à Tokyo au début 1995 ; son but est de fournir une plate-forme uniforme pour les agents mobiles dans un environnement hétérogène tel que celui de l'Internet. La plateforme Aglets a un serveur d'agent appelé « Tahiti ».

Les aglets sont des objets java mobiles qui peuvent se déplacer d'un hôte à un autre. Ainsi, un aglet peut stopper son exécution, se déplacer vers un hôte distant et continuer son exécution dans le nouvel environnement.

IV.2.1.3.1 Architecture de la plateforme Aglet

Les principaux éléments de la plateforme Aglets sont les suivants (Bettahar, 2003):

- Aglet : est un objet mobile de Java qui se déplace et visite les différents hôtes d'un réseau. Un aglet est autonome puisqu'il peut reprendre son exécution dès son arrivée à destination et réactif car il peut répondre (réagir) à des événements de son environnement.
- Proxy : est le représentant d'un aglet. Il sert de bouclier à l'aglet contre l'accès direct à ses méthodes publiques. Le proxy fournit également la transparence à l'emplacement pour l'aglet. C'est-à-dire qu'il peut cacher le vrai emplacement de l'aglet.

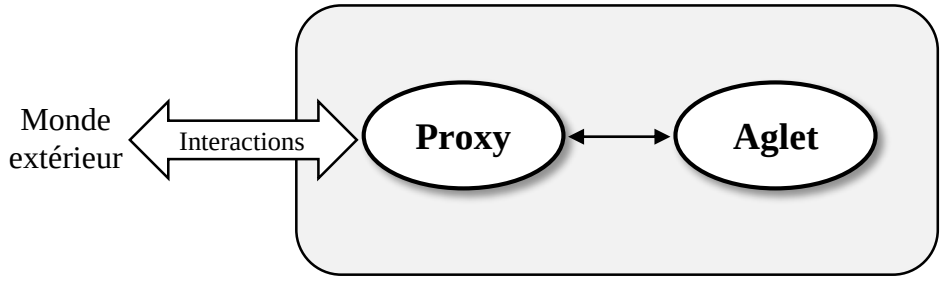

Figure IV.1 - Relation entre un Aglet et son Proxy

- Contexte : représente l'environnement d'exécution de l'aglet. Il fournit des moyens pour mettre à jour et contrôler des aglets dans un environnement uniforme d'exécution.

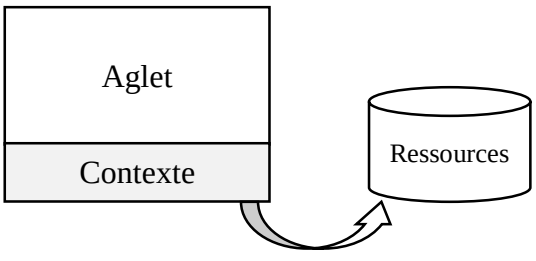

Figure IV.2 - Evolution d'un Aglet dans un contexte

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

IV.2.1.3.2 Cycle de Vie d'un Aglet

Ci-dessous nous définissons les principales opérations affectant la vie d'un aglet, à savoir, la création, le clonage, la déportation, la récupération, la désactivation, l'activation et la destruction (voir la Figure IV.3) :

- Création : la création d'un Aglet se fait dans un contexte par l'appel de la méthode *createAglet()*. Un Identifiant unique est assigné. L'initialisation et l'exécution de l'aglet commence immédiatement.

- Clonage : c'est la création d'une copie identique de l'Aglet d'origine dans le même contexte. La seule différence est qu'un autre identificateur est attribué au nouveau Aglet cloné. A noter que les processus ne peuvent pas être clonés.
- Déportation (dispatching) : comme décrit dans la Figure IV.4, la déportation consiste à transférer l'Aglet à partir de son contexte vers un autre, où il reprend son exécution. On dit que l'aglet a été poussé vers un nouveau contexte.

- Récupération (retraction) : l'Aglet déporté est récupéré dans son contexte d'origine.

- Activation et Désactivation : la désactivation d'un aglet est l'interruption temporaire de son exécution et le stockage de son état sur un support secondaire.

- Libération ou destruction : fin de vie de l'aglet et son retrait du contexte.

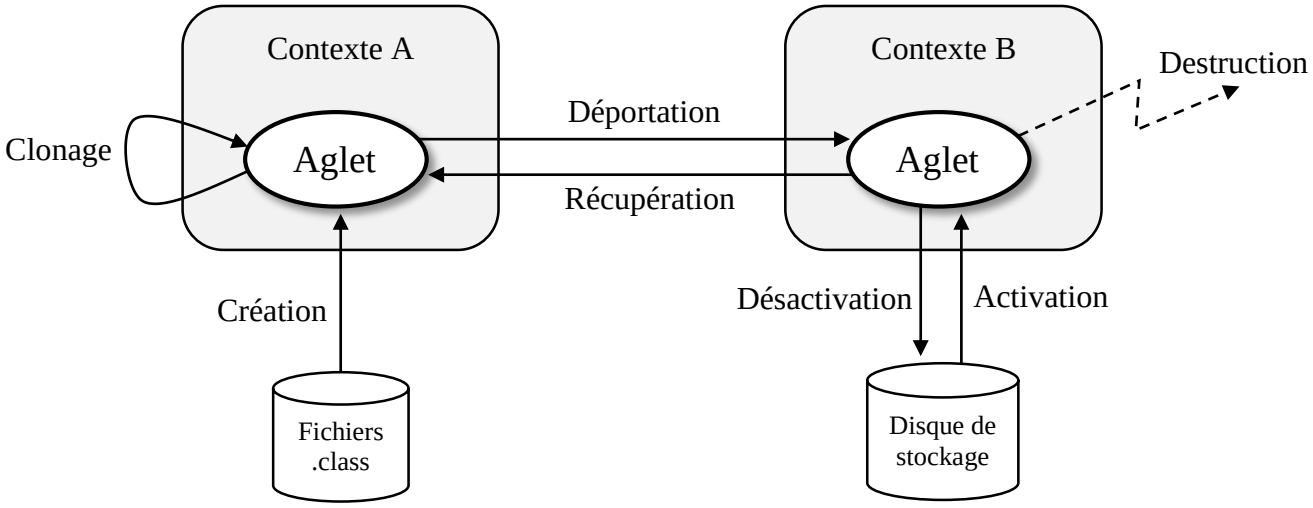

Figure IV.3 - Modèle du cycle de vie d'un Aglet

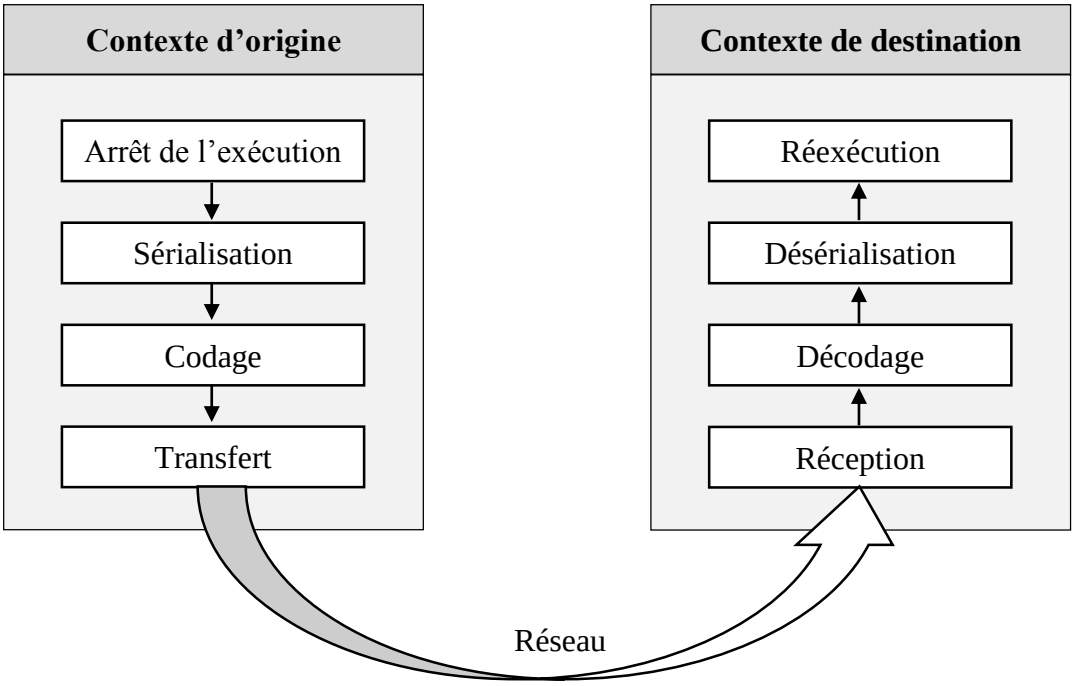

Figure IV.4 - Transfert d'un Aglet

IV.2.1.3.3 Serveur d'Aglets

Le serveur d'Aglets que nous allons choisir dans notre implémentation est le serveur Tahiti, qui est un gestionnaire d'agents visuel. Tahiti (Bettahar, 2003) utilise une interface graphique unique pour suivre et contrôler l'exécution des aglets. Il est possible en utilisant le glisser déposer de faire communiquer deux agents ou de les faire migrer vers un site particulier. Tahiti dispose d'un gestionnaire de sécurité paramétrable qui détecte toute opération non autorisée et empêche l'agent de la réaliser.

IV.2.2 Système d'exploitation

Le système de détection d'intrusions que nous allons implémenter tourne sur n'importe quel système d'exploitation. Autrement dit, est un système multiplateforme. Toutefois, il nécessaire d'installer la plateforme d'Aglets, et ce pour exécuter les agents mobiles.

IV.2.3 Matériel

Notre approche « AM-SID » que nous allons implémenter nécessite :

- Un réseau local (LAN), qui contient au minimum deux (02) machines.
- Un serveur d'agent « Aglets » (Serveur Tahiti) installé sur chaque machine du réseau.

IV.3 Présentation de l'étude de cas

Afin de validé notre approche de sécurité proposée « AM-SID », nous allons l'appliquer, pour étude de cas, sur le système de paiement carte bancaire en se basant sur les distributeurs automatiques de billets.

IV.3.1 Les distributeurs automatiques de billets

Pour que les clients d'une institution financière puissent retirer de l'argent liquide facilement et en permanence, les distributeurs automatiques de billets offrent une possibilité de retrait d'argent sans intervention du personnel de l'institution et ce 24 heures sur 24.

Un Distributeur Automatique de Billets (DAB) ou Guichet Automatique Bancaire (GAB), comme le montre la Figure IV.5, est un appareil électronique et électromécanique de télécommunication informatisé permettant aux clients d'effectuer différentes transactions financière dans un lieu public, notamment, le retrait d'espèces, la consultation de solde, la commande de chéquiers, …etc. L'homologue en anglais est Automated Teller Machine (ATM).

Le client insère une carte en plastique munie d'une bande magnétique ou d'une puce contenant les données nécessaires à l'identification du client. Pour demander l'accès à son compte, le client saisit un code de quatre chiffres. Si le code est erroné plusieurs fois de suite, la machine retire la carte dans le but d'éviter tout risque de fraude.

IV.3.1.1 Matériels et logiciels

Les DAB sont composés de nombreuses pièces, lui permettant bien sûr de remplir sa fonction, mais aussi de faciliter la tâche à l'utilisateur et garantir la sécurité de l'argent et des données de l'utilisateur.

Au niveau matériel (Web-01, 2014), le DAB dispose (voir Figure IV.5) :

- Un écran et un clavier numérique, pour permettre au client de rentrer son code et choisir la somme à retirer.
- Un lecteur de carte afin d'insérer la carte bancaire, et permettant, grâce à la puce de cette carte, de vérifier sa validité.
- Un coffre, dans lequel sont présentes plusieurs cassettes selon la valeur des billets.

Figure IV.5 - Distributeur Automatique de Billets (DAB)

Les DAB utilisent aujourd'hui des composants similaires à ceux d'un PC (carte mère, CPU, disque dur), la partie logicielle est basée sur un système d'exploitation comme Linux, ou Windows.

IV.3.1.2 Sécurité

Afin de sécuriser le coffre d'un DAB, il est parfois munie de certaines sécurités, tels des gaz incapacitants, ou de l'encre indélébile, systèmes qui s'enclenchent en cas d'ouverture forcée du coffre. Ce renforcement de sécurité a donc permis l'installation de DAB dans un plus grand nombre de lieux. De plus, une caméra de sécurité est souvent présente autour du distributeur.

IV.3.1.3 Réseaux

Les DAB sont connectés à un réseau bancaire (grâce au modem intégré dans le DAB), permettant à la machine de vérifier des informations comme le code personnel, la validité de la carte, l'identité de la personne (censée posséder la carte) et le solde restant sur son compte en banque. Le processeur du DAB assure le chiffrement des données utilisateurs, qui ne sont bien sûr pas stockées sur le disque dur du DAB, mais directement envoyées au serveur bancaire pour comparaison.

IV.3.1.4 Risques et attaques

Malgré une sécurité matérielle et logicielle plutôt renforcée, des risques et des attaques subsistent. Il existe certaines failles de sécurité au niveau logiciel, permettant à un pirate expérimenté de contrôler la machine à distance ou de retirer l'argent comme bon lui semble. Il existe aussi des vols en ligne, par exemple avec des sites de ventes frauduleux, sur lesquels le code saisi par l'utilisateur sera récupéré immédiatement par le voleur.

IV.4 Réalisation de notre système

IV.4.1 Implémentation des agents

Notre système est constitué de sept (07) agents qui coopèrent pour la détection d'intrusion. Ces agents ont implémenté (comme les montre la Figure IV.6) en utilisant le langage Java et la plateforme Aglet. L'Agent Administrateur, Agent de Crise, Agent Actionneur, Agent Collecteur-NIDS et Agent Collecteur-HIDS sont des agents stationnaires. Tandis que l'agent Analyseur-NIDS et agent Analyseur-HIDS sont des agents mobiles.

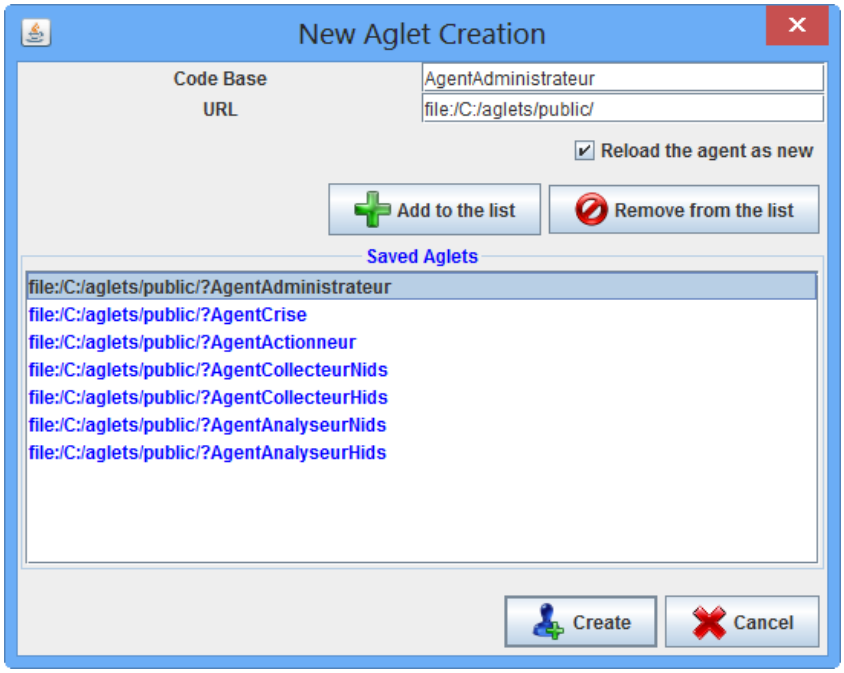

Figure IV.6 - Implémentation des différents agents utilisés dans notre système

Nous avons installé aussi le serveur Tahiti au niveau de chaque DAB pour offrir une gestion visuelles des agents. Tahiti utilise une interface graphique unique pour suivre et contrôler l'exécution des agents (ou aglets). Il est possible en utilisant le glisser déposer de faire communiquer deux agents ou de les faire migrer vers un site particulier. Tahiti dispose d'un gestionnaire de sécurité paramétrable qui détecte toute opération non autorisée et empêche l'agent de la réaliser.

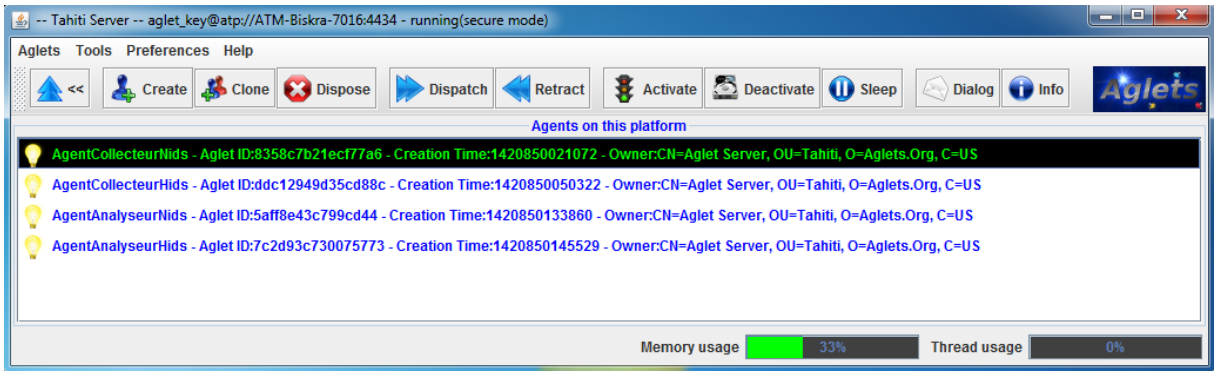

Figure IV.7 - Agents exécutés sur le distributeur ATM-Biskra-7016 (interface Tahiti)

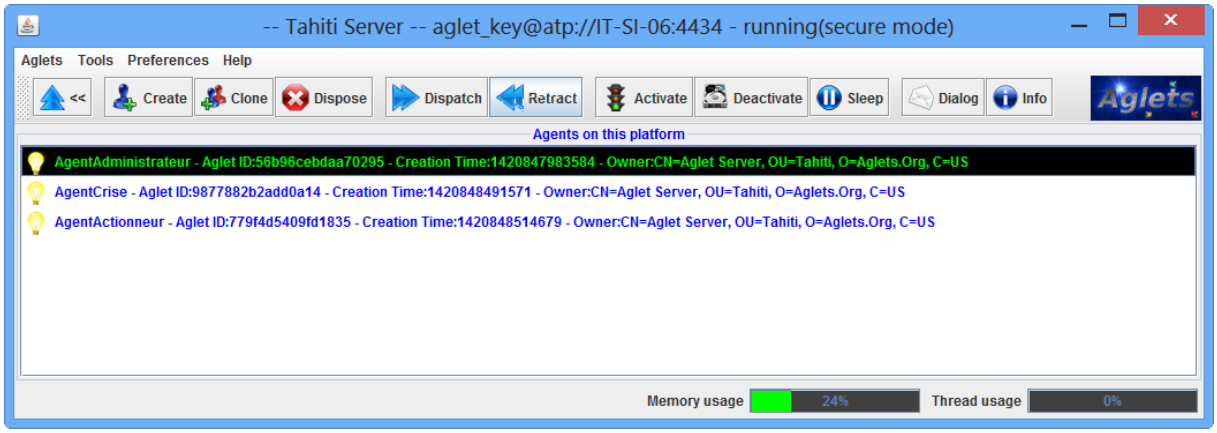

Figure IV.8 - Agents exécutés sur le Serveur (interface Tahiti)

IV.4.2 Administration du système

Pour administrer notre système, nous utilisons à la fois l'application que nous avons développée « Administration AM-SID » ainsi que le serveur Tahiti. Ce dernier est installé au niveau de chaque DAB, tandis que notre application développée est installée seulement au niveau du serveur et elle permet de gérer à distance :

- les agents actifs au niveau de chaque DAB ;
- les attaques détectées ;
- la liste et l'état des différents DAB connectés.

Voir les figures : Figure IV.9, Figure IV.10 et Figure IV.11

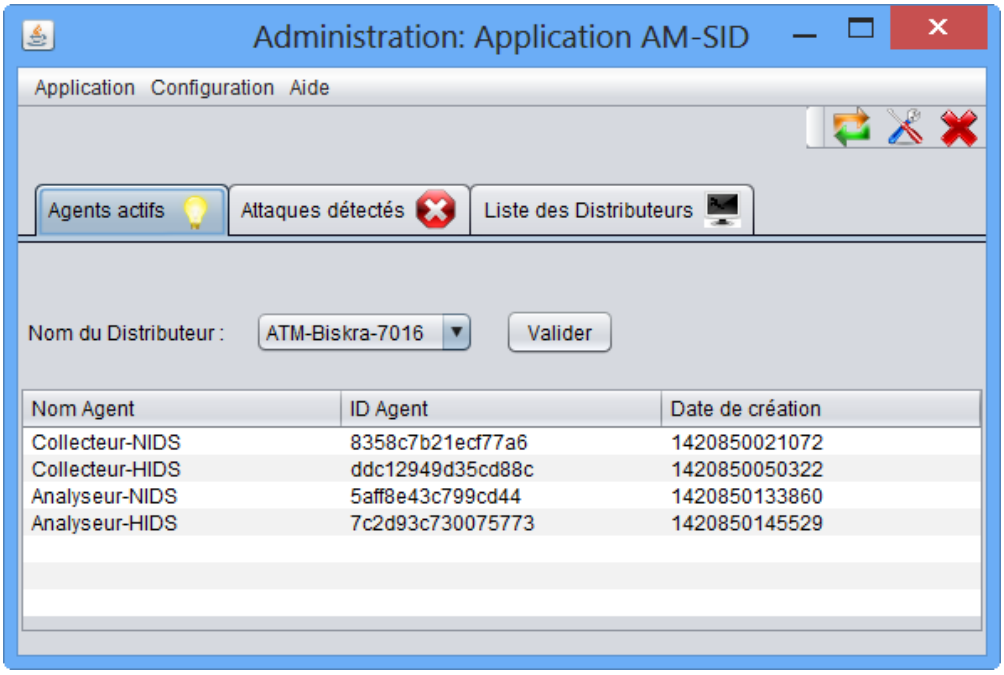

Figure IV.9 - Agents exécutés sur le distributeur ATM-Biskra-7016

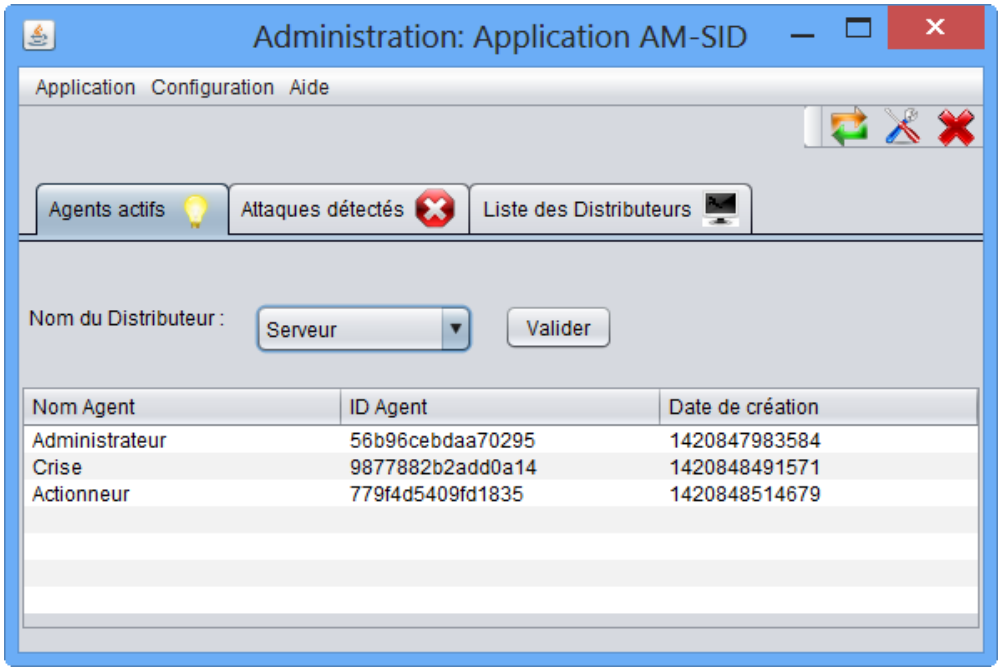

Figure IV.10 - Agents exécutés sur le Serveur

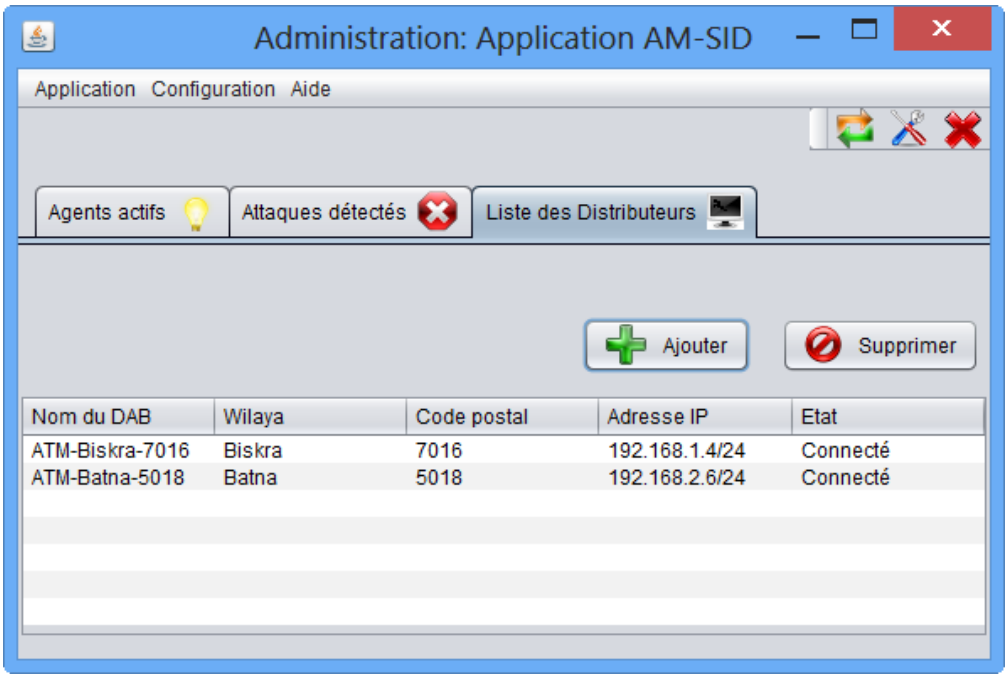

Figure IV.11 - Liste des DABs dans le système

IV.4.3 Simulation d'une attaque

Pour évaluer et valider notre système de sécurité AM-SID, nous allons simuler une attaque au niveau d'un distributeur automatique de billets. En effet, il existe plusieurs types d'attaque. Parmi les plus répondus on trouve l'attaque de type « TCP SYN flooding » qu'elle est également connue sous le nom « SYN attack ».

L'objectif de l'attaque TCP SYN flooding (Boudaoud, 2002) est de rendre indisponible un service TCP offert sur une machine. Le principe de Cette attaque est de créer des connexions TCP semi-ouvertes sur la machine cible afin de remplir la file d'attente où sont stockées les demandes d'ouverture de connexions. L'attaquant envoie un grand nombre de requêtes SYN à la machine cible et remplace son adresse source avec l'adresse d'une machine indisponible ou inexistante afin que les réponses SYN/ACK ne soient jamais reçues et que donc les messages ACK ne soient jamais générés, ce qui signifie que la file d'attente restera pleine.

Les conséquences de cette attaque est que toutes les requêtes arrivant sur le port TCP cible seront ignorées et de ce fait le service fourni sur ce port sera indisponible. Dans certains cas, la machine peut aussi devenir indisponible.

Nous simulerons cette attaque en utilisant l'outil « HPING » (Boukhlouf & Kazar, 2012) qui est capable d'envoyer des paquets TCP / IP personnalisés pour les hôtes du réseau. Cet outil est utilisé comme suite :

```
# hping –S –i u10 –p 80 –a @IP_MACHINE_USURPEE @IP_MACHINE_CIBLE
```

Où :

- *S :* demande le positionnement du flag.
- *i :* permet de préciser l'intervalle de temps (micro secondes) entre deux envois.
- *p :* précise le port destination.
- *MACHINE_USURPEE :* adresse source.
- *MACHINE_CIBLE :* adresse de destination.

IV.4.4 Détection de l'attaque

Dans le cas où l'agent Analyseur détecte une attaque sur un DAB distant, il envoie une alerte à l'agent Actionneur. Ce dernier prendra les mesures nécessaires pour contrer l'attaque. Ensuite il met à jour la base de données des signatures d'attaques et affiche le message ci-dessous en précisant le type de l'attaque ainsi que les propriétés du DAB attaqué (nom, Adresse IP, Wilaya, …).

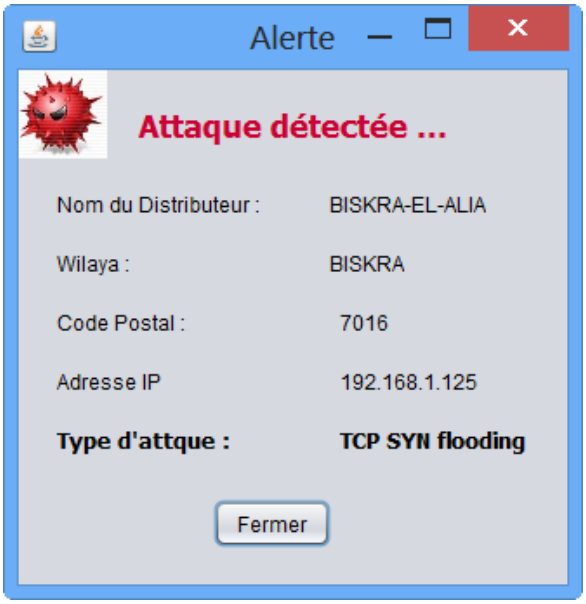

Figure IV.12 - Détection de l'attaque TCP SYN Flooding.

IV.5 Conclusion

Durant ce chapitre nous avons proposé une étude de cas pour l'implémentation et la mise en œuvre de notre approche AM-SID tout en concentrant sur la mobilité des agents. Nous avons utilisé pour l'implémentation le langage Java avec la plateforme Aglets. Cette dernière permet d'un côté de développer des agents, et d'un autre côté d'être utilisée comme serveur d'agents mobiles. L'inconvénient majeur de la plateforme Aglets est qu'il faut l'installer sur chaque hôte dans le système.

Conclusion générale

Dans ce mémoire nous nous sommes focalisé sur le problème de gestion de la sécurité et plus précisément la détection d'intrusions dans les systèmes industriels distribués, où nous avons proposé une approche hybride de détection d'intrusion à base d'agents mobiles « AM-SID ».

Les agents mobiles sont capables de se déplacer vers les hôtes à surveiller pour collecter et analyser les informations, ce qui réduit la charge sur le réseau et diminue la quantité d'informations échangées dans les systèmes complexes et complètement distribués.

Nous avons implémenté notre approche en basant sur la plateforme Aglets pour la création et la distribution des différents agents mobiles. Les Aglets sont des objets mobiles Java qui peuvent se déplacer d'un hôte à un autre.

Ce projet nous a été très bénéfique, car il nous a permis de découvrir les systèmes industriels distribués, la sécurité informatique, les systèmes de détection d'intrusion, la technologie d'agents mobile et la plateforme Aglets. Ainsi, nous avons enrichi nos connaissances sur les réseaux.

En perspective la question de protection des agents mobiles eux-mêmes est un problème majeur, puisque les méthodes de sécurité actuelles ne suffisent pas à protéger efficacement l'agent mobile. En effet, l'hôte qui exécute un agent mobile, il a un plein pouvoir sur lui ; du moment qu'il l'héberge, il peut facilement accéder et changer ses données ainsi que son code. Par conséquent, les problèmes de sécurité des agents mobiles sont différents et beaucoup plus de ceux rencontrés dans les systèmes distribués. Ce problème aura traité dans un futur travail.

Bibliographie

Abidi, H. (2014). *Cours pour l'enseignement technologique.* http://www.technologuepro.com/.

Alfalayleh, M., & Brankovic, L. (2005). *An overview of security issues and techniques in mobile agents.* Australia: The School of Electrical Engineering and Computer Science, The University of Newcastle.

Babus, F. (2008). *Contrôle de processus industriels complexes et instables par le biais des techniques statistiques et automatiques, Thèse de doctorat.* Université d'Angers.

Balasubramaniyan, J. S., Garcia-Fernandez, J. O., Isacoff, D., Spafford, E., & Zamboni, D. (1998). *An architecture for intrusion detection using autonomous agents.* In Proceedings of the 14th IEEE computer security applications conference.

Benoudina, L. (2009). *Modélisation et simulation basées multi-agents du contrôle de processus industriels - Thèse de Magister.* Université 20 Aout 1955 - Skikda.

Bergougnoux, L. (2005). *A.P.I. - Automates Programmables Industriels.* Revue technique. Université de polytechnique SIIC de Marseille.

Boudaoud, K., & Nobelis, N. (2006). *Apprentissage de nouvelles attaques avec un modèle de Case-Based Reasoning.* Laboratoire I3S-CNRS, Université de Nice Sophia-Antipolis.

Calvet, J.-L. (2011). France: http://www.eea.ups-tlse.fr/Pres/Fr_Pres.htm.

Combacau, M. -B.-C.-K. (2000). *Systèmes de Production Sûrs de Fonctionnement- Réflexions sur la Terminologie Surveillance - Supervision.* http://homepages.laas.fr/combacau/SPSF/sursup.html.

CUBAT, D. C. (2005). *Agents Mobiles Coopérants pour les Environnements Dynamiques, Thèse de Doctorat.* Toulouse, France: l'Institut National Polytechnique de Toulouse.

Ferber, J. (1995). *Les Systèmes multi-agents : Vers une intelligence collective.* InterEditions.

Fong, P. W. (1998). *Viewer's Discretion : Host Security in Mobile Code Systems. Technical Report SFU CMPT TR 1998-19, School of Computing Science, Simon Fraser University*. Burnaby, B.C., Canada.

Jansen, W., & Karygiannis, T. (1999). *Mobile Agent Security,*. National Institute of Standards and Technology Computer, NIST Special Publication 800-19.

Jarras, I., & Chaib-draa, B. (2002). *Aperçu sur les systèmes multiagents - Série scientifique.* Montréal: CIRANO.

Jennings, N. R., Sycara, K., & Wooldridge, M. (1998). *A Roadmap of Agent Research and Development - Journal: Autonomous Agents and Multi-Agent Systems.* Boston, USA: Kluwer Academic Publishers.

Labidi, S., & Lejouad, W. (1993). *De l'Intelligence Artificielle Distribuée aux Systèmes Multi-Agents* . INRIA - N°2004.

LOULOU, A. M. (2010). *Approche Formelle pour la Spécification, la Vérification et le Déploiement des Politiques de Sécurité Dynamiques dans les Systèmes à base d'Agents Mobiles, Thèse de Doctorat, Université de Bordeaux 1 et Université de Sfax.*

Loureiro, S., Molva, R., & Roudier, Y. (2000). *Mobile Code Security, In Proceedings of ISYPAR 2000.* Toulouse.

Perret, S. (1997). *Agents mobiles pour l'accès nomade à l'information répartie dans les réseaux de grande envergure, Thèse de doctorat de l'Université Joseph Fourier - Grenoble I.*

Peter, B., & Wilhelm, R. (2005). *Mobile agents : basic concepts, mobility models, and the Tracy toolkit.* San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Pierre, S. (2011). *Réseaux et systèmes informatiques mobiles: Fondements, architectures et applications.* Montréal: Presses internationales Polytechnique.

Rubin, A. D., & Geer, D. E. (1998). *Mobile Code Security. IEEE Internet Computing, 2(6) :30–34.* New Jersey, USA.

Sander, T., & Tschudin, C. F. (1998). *Protecting Mobile Agents Against Malicious Hosts. in G. Vigna (ed.), Mobile Agent Security. International Computer Science Institute.* USA.

Sansonnet, J.-P. (2004). *Introduction aux systèmes multi-agents, Cours SMA en ligne du Master recherche de Paris XI.*

Skaita, J., & Mourlin, F. (2002). *A mobile approach for the intrusion detection.* University of Paris XII, Val-de-Marne.

Snapp, S. R.-B.-D.-G. (1991). *DIDS (Distributed Intrusion Detection System) - Motivation, Architecture, and An Early Prototype.* University of California, Davis: In Proceedings of the 14th National Computer Security Conference.

Soltani, L. (2007). *Les Systèmes Multi-Agent pour le Contrôle de Production - Thèse de Magister*. Université Hadj Lakhdar - Batna.

Vigna, G. (1997). *Protecting Mobile Agents through Tracing. In Proceedings of the 3rd ECOOP Workshop on Mobile Object Systems, Jyvalskyla.* Finland.

Vrignat, P. -A.-D.-K. (2009). *Evaluation et organisation des activités opérationnelles en maintenance dans le cadre de processus industriels.* Colloque International Francophone, Evaluation des Performances et Maîtrise des Risques Technologiques pour les Systèmes Industriels et Energétiques, LE HAVRE, France.

Wayne, J. (2000). *Countermeasures for Mobile Agent Security.* Computer Communications.

Wooldridge, M., & Jennings, N. (1995). *Intelligent agents: Theory and practice.* The Knowledge Engineering Review, 10(2):115–152.

Yahiaoui, S. -I. (2005). *Conception et Réalisation d'un Système de Détection d'Intrusion - Mémoire de fin d'étude, en vue d'obtention du diplôme d'ingénieur d'état en Informatique.* Université Mouloud Mammeri Tizi-Ouzou.

Ye, D., Bai, Q., Zhang, M., & Ye, Z. (2008). *P2P Distributed Intrusion Detections by Using Mobile Agents.* In Proceedings of the 7th National Computer Security Conference.