Vers une collecte pertinente des informations de sécurité sur Internet

M. Camus[1], L. Auroux[1] & D. Gousseau[1]

1: L.E.R.I.A, Laboratoire Epitech de Recherche en Informatique Appliquée
24 rue Pasteur 94270 Le Kremlin Bicêtre, France
{camus_m,auroux_l,gousse_d}@anciens.epitech.net

Résumé

De nos jours, il est difficile pour un administrateur système de collecter des informations de sécurité. En effet, le flux de données présent sur la toile mondiale est trop important, ce qui rend le traitement des informations de sécurité trop long et trop complexe. De ce fait, un outil de récupération d'information de sécurité sur internet est utile pour toute personne ayant besoin de recevoir les informations relatives à la sécurité. Cet outil est composé d'un système multi-agents autonome distribué, grâce au protocole CORBA, dont le rôle est de collecter des informations en s'assurant de l'authenticité et de la pertinence de celles-ci. De plus, un tri automatique est effectué pour que l'utilisateur n'aie plus à traiter que les données de sécurité qui l'intéressent comme par exemple ne plus recevoir que les informations sur les failles de sécurité relative à ses propres équipements.

1. Introduction

1.1. Contexte

Aujourd'hui tous les internautes ont accès à un grand nombre d'informations relatives à la sécurité informatique sur internet. Par exemple une recherche internet sur les mots "portail sécurité internet" donne entre 10 000 et 33 000 réponses (voila.fr et altavista 10 000 réponses, google 33 500 réponses) de même une recherche sur les mots "mailing-list securite informatique" donne entre 2 000 et 5 000 réponses sur ces mêmes moteurs. Ce (trop) grand nombre de sources d'informations pose la question de savoir comment sélectionner celles-ci pour être sûr d'avoir rapidemment un aperçu de toutes les vulnérabilités, bugs et nouveautés pertinentes.

De plus si le nombre de sources d'informations est important, le nombre d'informations émis par les différentes sources par besoin d'exhaustivité est lui aussi très important. Par exemple le CERT [1] a recensé 2437 vulnérabilitées en 2001 et publié 341 notes de sécurité la même année. Le problème est d'obtenir des informations utiles et vérifiées. Utiles parce que toutes les alertes ne concernent pas tout le monde et qu'il y a un risque de ne plus lire les informations si il n'y a aucune possibilité de les filtrer. Et vérifiées parce que les fausses alertes de sécurité (faux virus ou faux bugs) sont de plus en plus courantes. Le nombre de sites traitant des "canulars" ou "hoax" est de plus en plus important (par exemple www.hoaxbuster.com depuis 2000 est un des premier site français traitant de ce sujet), et les portails sur la sécurité informatique sont aujourd'hui obligés de tenir compte de ce phénomène.

L'objectif de ce projet est d'apporter un outil de veille sécurité qui permet de collecter les informations relatives a la sécurité, d'authentifier l'information collectée (niveau de confiance en l'information), et de filtrer les informations pour ne présenter que celles qui sont pertinentes pour l'utilisateur.

[1] Computer Emergency Response Team

1.2. Composition du projet

Cet outil de veille sécurité s'appuie essentiellement sur un système multi-agents autonomes possédant une base de connaissance privée et commune pour éviter la redondance de connaissances. Ainsi les agents pourront prendre des décisions sur tels types d'informations ou de problèmes. Ces agents sont de 3 types : agent local, agent central, et agent collecteur.

Agent local : Il gère l'authentification de l'utilisateur et l'affichage des informations. En fait ce n'est qu'une interface utilisateur adaptable à d'autres systèmes d'informations.

Agent central : Il authentifie les agents locaux (l'utilisateur), il gère le profil de l'utilisateur, et alimente l'agent local avec les informations collectées si nécessaire.

Agents collecteurs : Ces sont les agents qui collectent les informations sur internet et les donnent à l'agent central. Ils peuvent être de différents types :

- agent mailing-list, cet agent s'abonne à des mailing-list et sélectionne les informations pour alimenter l'agent central.
- agent newsgroup, il fait la même chose que l'agent mail, mais pour les newsgroups.
- agent web-portail, cet agent collecte les informations sur les portails de sécurité.
- agents générique, cet agent est une interface directe entre les fournisseurs d'informations de sécurité, et l'agent central.

2. Fonctionnement du système multi-agents autonomes

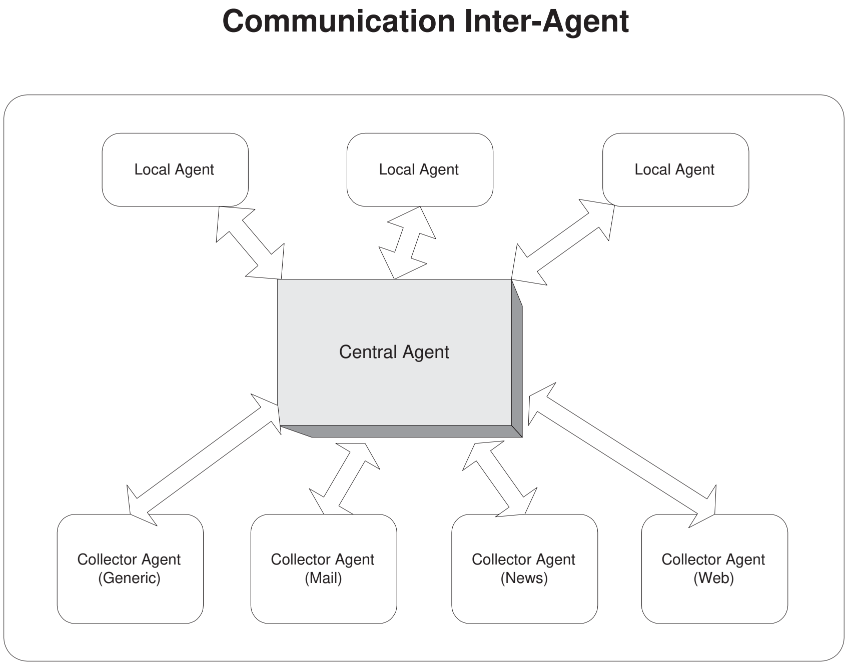
Celui-ci est composé de différentes entités bien distinctes. Chacune de ces entités aura un rôle bien spécifique à jouer lors de la communication des informations qui auront été collectées par les agents collecteurs. L'agent local, lui, est une interface utilisateur avec laquelle celui-ci va se mettre en relation avec l'agent central. Le nombre d'agents locaux par utilisateur n'est pas limité, cela permet une configuration de recherche poussée.

Les agents locaux ne correspondent pas directement avec les agents collecteurs. En effet, un agent central, qui est unique, joue le rôle d'un intermédiaire pour filtrer les informations en fonction des profils des utilisateurs.

2.1. L'environnement de communication

Comme le montre le schéma ci-dessous, la communication entre les différents agents se fait dans les deux sens. C'est à dire qu'une capacité de lecture et d'écriture est donnée à tous les agents qui pourront, comme ceci, agir directement entre eux. La communication entre agents de même nature est impossible. Chaque agent de nature locale ou collecteur est obligé de passer par l'agent central pour lire les caractéristiques des utilisateurs et écrire les informations collectées.

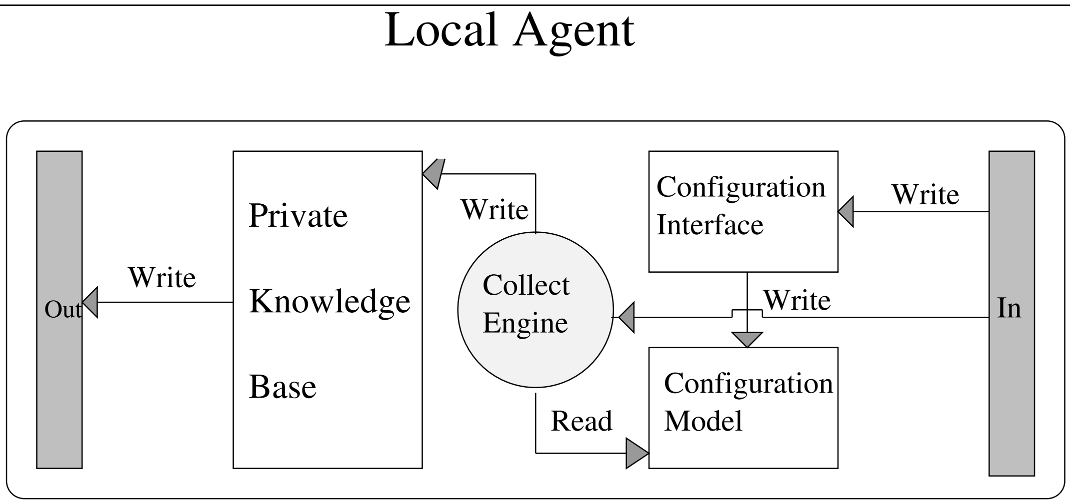
L'agent central récupère les informations collectées par les agents collecteurs pour les rediriger vers les agents locaux correspondant au profil des informations.



2.2. L'agent local

Comme le montre la figure ci-dessous, l'agent local se compose de six éléments bien distincts ayant un système de communication strict. Cinq de ces éléments gravitent autour du cœur de l'agent : le moteur de collecte.

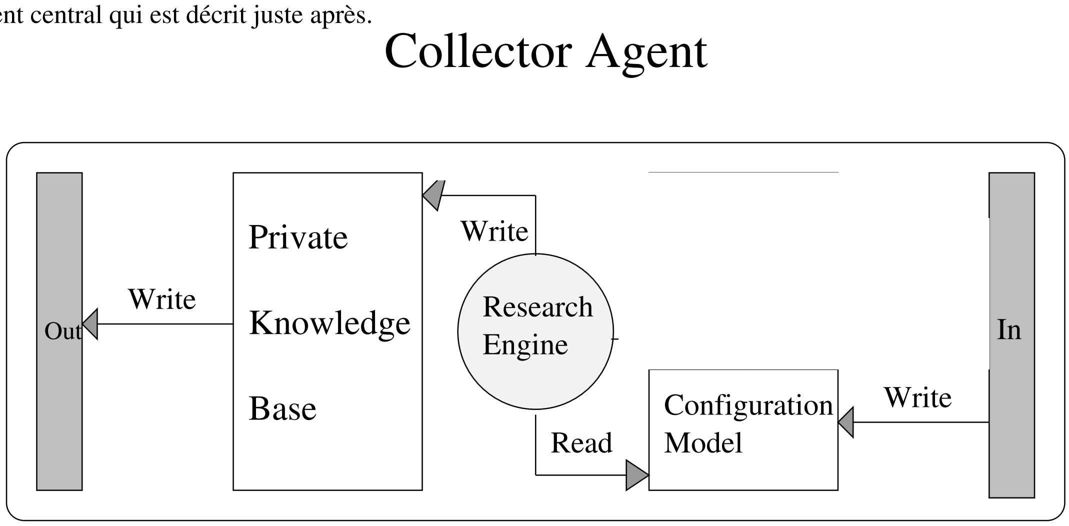
Ce moteur va regrouper les informations arrivant de l'extérieur directement dans la base de connaissance de l'agent, tout en prennant connaissance de la configuration de base de celui-ci. La Base de connaissances et le modèle de configuration sont aussi des éléments important de l'agent puisque la base va stocker les données désirées, le modèle et le profil de l'utilisateur. Bien sûr, les trois autres éléments ont aussi leur importance, l'interface de configuration sert à récupérer le profil utilisateur de l'agent pour l'écrire dans le modèle de configuration. Quand à l'entrée et la sortie, elles donnent bien entendu la possibilité à l'agent de communiquer avec l'extèrieur.



2.3. L'agent collecteur

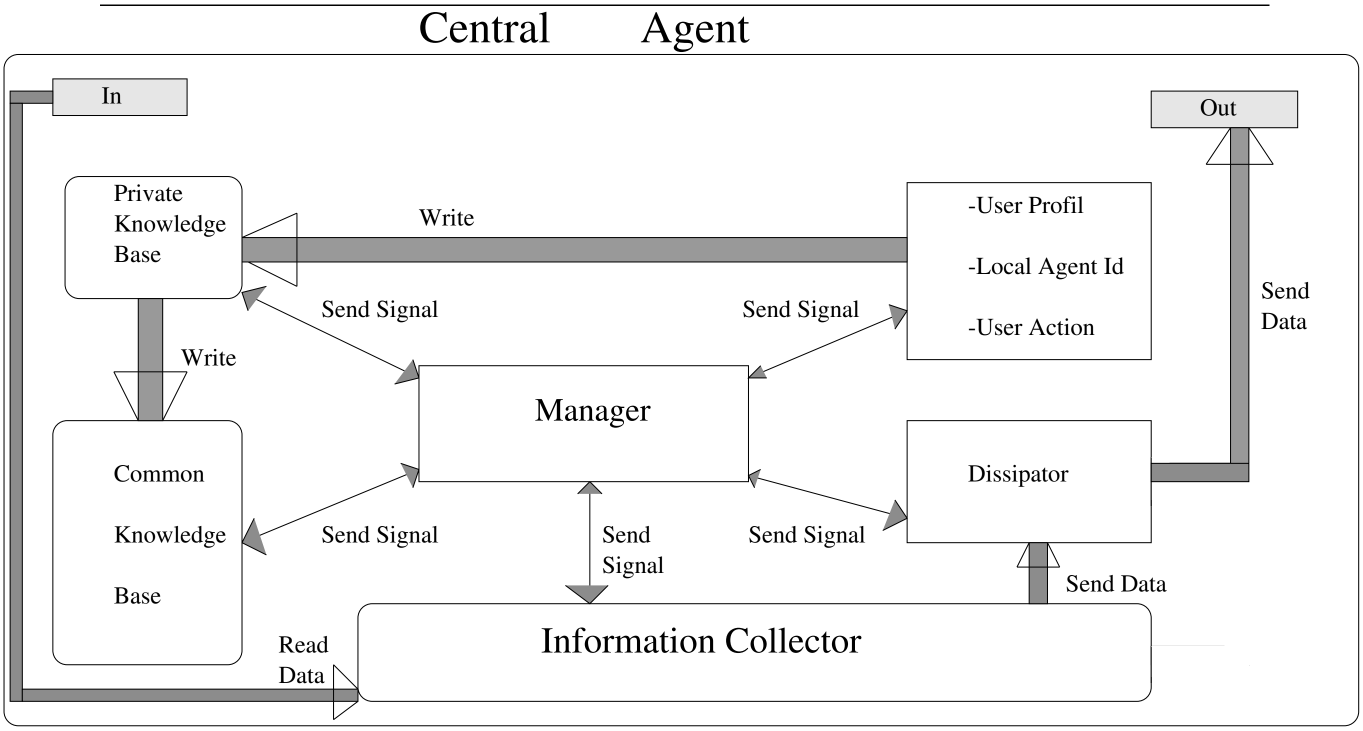
Cet agent possède une structure qui ressemble à l'agent local à quelques exceptions près. Celles-ci se distinguent au niveau de la communication des éléments, du cœur de l'agent et de l'interface utilisateur.

En effet, le cœur de l'agent est un moteur de recherche qui va effectuer celle-ci en tenant compte du modèle de configuration précédement récupéré de l'agent central. D'où l'absence d'une interface de communication, car l'utilisateur n'agit pas directement sur cet agent, mais passe par l'intermédiaire de l'agent central. Les données récupérées seront bien entendu inscrites dans la base de connaissance pour ensuite être envoyées à l'agent central qui est décrit juste après.



2.4. L'agent central

L'agent central est l'entité la plus importante du système. C'est lui qui dirige et met en relation les agents locaux et les agents collecteurs. c'est aussi lui qui possède la base de connaissance commune à tous les agents du système. Comme le montre le schéma ci-dessous, l'agent central possède une entrée et une sortie. Chacune de ces connexions possède plusieurs flux qui donnent la possiblité à l'agent de communiquer en lecture-écriture avec plusieurs agents.



L'agent central est composé de plusieurs éléments qui ont un rôle bien précis pour la communication inter-agent :

– une entrée,
– une sortie,
– une base de connaissance privée,
– une base de connaissance commune,
– un collecteur d'information interne,
– un dissipateur d'information,
– une interface utilisateur,
– un manager

Tous ces éléments peuvent communiquer entre eux. Cette communication peut se faire de deux manières :

– une communication directe,
– Une communication par intermédiaire

La communication directe se fait, soit en lecture, soit en écriture, elle ne peut pas se faire en half-duplex. La communication par intermédiaire, quant à elle, se fait par des IPC[2]. La communication directe est utilisée par l'entrée,la base de connaissance privée, le collecteur d'information, l'interface utilisateur, et le dissipateur. Pour les signaux, ceux-ci sont utilisés par le manager, la base de connaissance privée, la base de connaissance commune, le collecteur d'information, le dissipateur et l'interface utilisateur.

Tous ces éléments, qui forment l'agent central, ont la charge d'exécuter certaines actions délibérément séparées. Voici le rôle de chaque élément :

– **L'entrée** est chargé de faire circuler et de synchroniser les flux entrants pour le compte du collecteur d'information. Elle gère une file d'attente de type FIFO (First Input First Output), le temps que les informations soient traitées.
– **La sortie** est chargée de faire circuler et de synchroniser les flux sortants pour le compte du dissipateur. Cet élément gère aussi une file d'attente de type FIFO pour que les informations soient redistribuées.

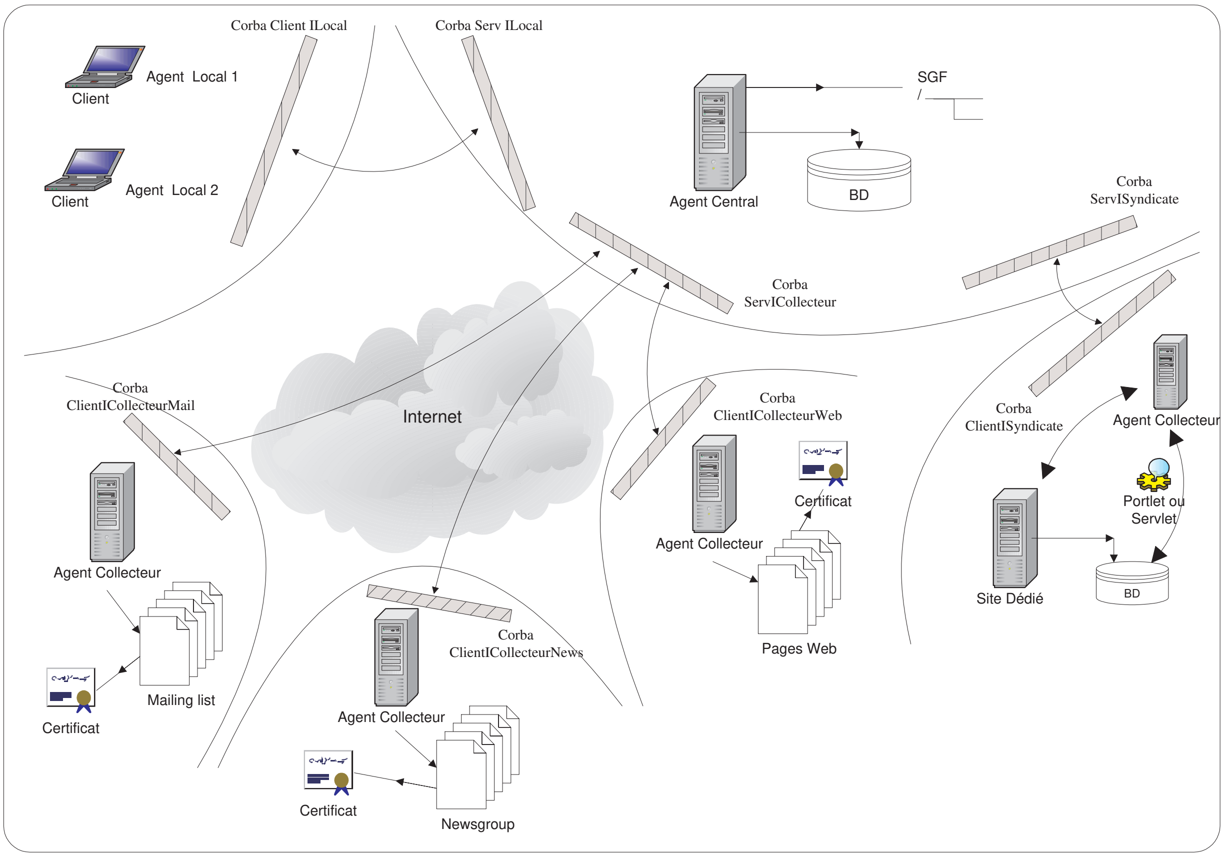
[2]Inter Process Communication

- **La base de connaissance privée** récupère des informations concernant les comptes des utilisateurs et les actions à mener pour ceux-ci, ainsi que la configuration pour tous les agents locaux et les agents collecteurs, décrite par les utilisateurs.
- **La base de connaissance commune** regroupe toutes les informations communes à tous les agents locaux et les agents collecteurs, ce qui évite la redondance d'information et augmente la capacité de stockage de la base de connaissance privée des agents locaux et collecteurs.
- **Le collecteur d'information interne** se charge de récupérer les informations des agents collecteurs en gérant une file d'attente par taille d'information, c'est à dire que quand le collecteur possède un nombre suffisant d'informations, il passe ces dernières au dissipateur.
- **Le dissipateur** gère la répartition des informations vers les agents locaux. Il traite les différents profils des utilisateurs pour envoyer les informations aux agents locaux correspondant.
- **L'interface utilisateur** sert lors de la configuration des comptes des utilisateurs et quand ceux-ci décrivent leur profil au niveau des agents locaux et des agents collecteurs. Cette interface écrit ensuite les données dans la base privée de l'agent central.
- **Le manager**, comme son nom l'indique, gère tous les éléments composant l'agent central. C'est lui qui joue le rôle de l'intermédiaire pour la communication par signaux. Un système de questions-réponses fonctionne entre lui et les autres éléments. Par exemple, il interroge la base de connaissance privée et commune pour que le dissipateur puisse trier les informations, ou il interroge ces deux même bases pour le compte de l'interface utilisateur, ce qui permet à cette dernière d'accepter ou non une configuration selon les éléments qui sont déjà connus.

L'agent central est donc le cœur du sytème multi-agents autonomes. C'est lui qui garde les informations critiques et qui permet aux utilisateurs d'avoir une communication indirecte avec les agents collecteurs afin de modifier leur configuration.

3. Spécification du protocole de communication entre les agents

Les langages multi-agents comme KQML (Knowledge Query and Manipulation Language) ou ACL (Agent Communication Language) ne proposent pas de système de localisation des agents ce qui rend l'utilisation d'une plateforme comme jade (Java Agent DEvelopment Framework) nécéssaire. la manière la plus adéquate pour implémenter cette architecture multi-agent, vu les besoins fonctionnels du projet, est d'utiliser les capacités de communication LAN-WAN de Corba.



En effet, Corba permet, comme indiqué dans l'exemple schématisé ci-dessus, une architecture distribuée.

Pour répondre à toutes les fonctionnalitées précédement exposées nous avons identifié différentes interfaces corba possible. Les noms indiqués sont à titre d'exemple. Nous indiquons aussi le nom probable de certaines methodes Corba et certaines contraintes d'implémentation pour une meilleur compréhension générale.

3.1. Agent central

L'agent central est composé de différents servants [3] corba enregistrés auprés de l'adaptateur d'objet corba [4]. L'agent central est composé du servant ILocalAgent de l'agent local et du servant ICentralAgent de l'agent central.

Outre l'interface corba, l'agent central doit répondre aux fonctionnalitées suivantes :

- Les agents locaux doivent être connus de l'agent central pour être authentifiés.
- Les agents collecteurs doivent être listés au niveau de l'agent central,pour qu'ils puissent s'inscrire auprés d'eux. Cela permet de partager un même agent collecteur pour plusieurs agents centraux, et ainsi

[3] interface serveur de l'objet distribué
[4] service d'enregistrement des objets distribué

d'introduire le concept de collecteurs publics dans une architecture répartie sur Internet.
– L'agent central liste tous les agents locaux connectés pour leur pousser l'information.
– Pour optimiser la recherche, l'agent central doit générer un filtre commun à tous les profils des agents locaux. Il le fournira aux agents collecteurs, afin que ceux-ci puissent filtrer une première fois l'information. Ainsi, l'agent central ne recevra que les informations utiles.

3.2. Agent local (interface ILocalAgent)

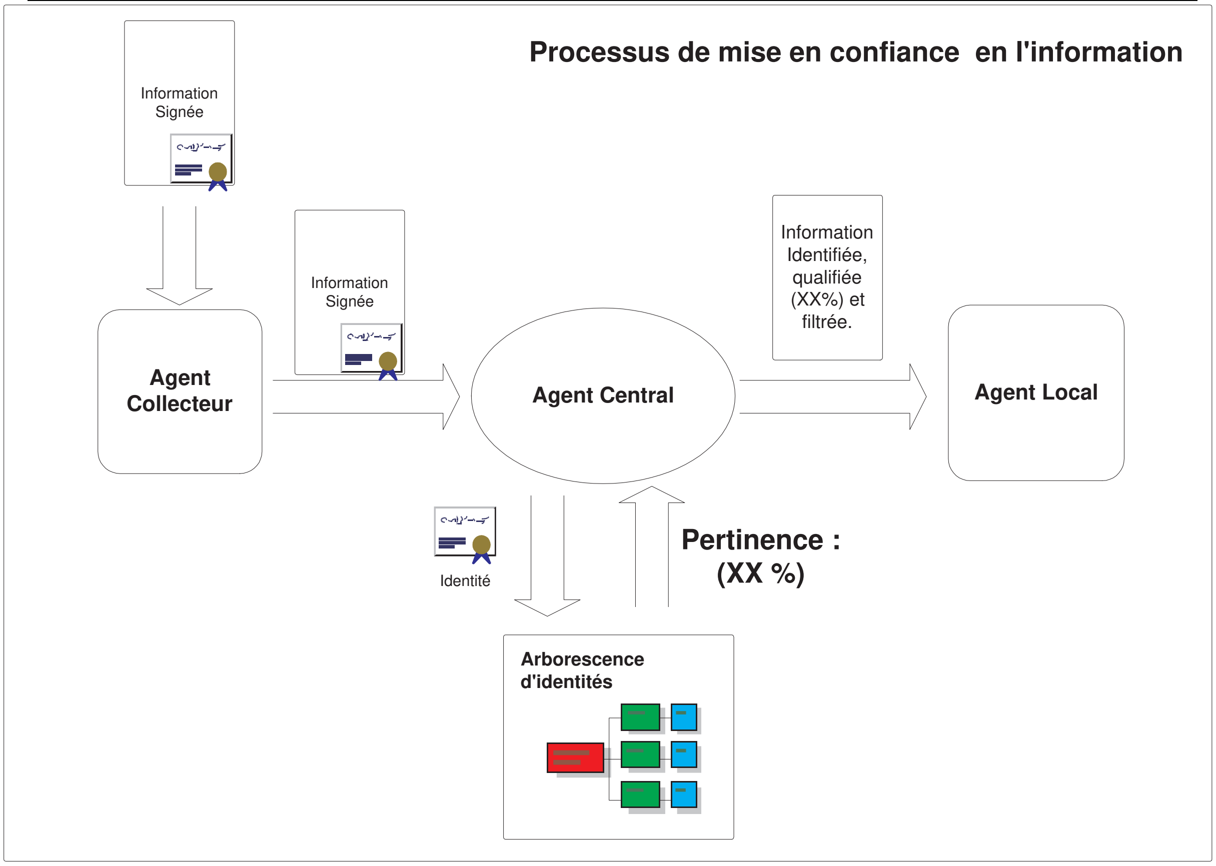
Le servant ILocalAgent est présent sur l'agent central, l'agent local contient la partie cliente du servant et accéde aux méthodes suivantes :
– L'agent local doit s'authentifier avant toute chose auprès de l'agent central qu'il connait. Il recevra un contexte de connexion utile pour la gestion de la session, lors de l'invoquation de méthode future. Il doit être possible de détecter si une erreur est survenue.
```
Context         connect(in Key key) raises(EGeneral);
```
– Il doit pouvoir administrer son profil, donc recevoir celui sauvegardé auprès de l'agent central, et le modifier.
```
Profile     GetProfile() raises (EGeneral) context ("ctx");
void        SetProfile(in Profile prof) raises (EGeneral) context ("ctx");
```
– Il doit créer une interface spécifique (IPush) pour que l'agent central puisse lui pousser l'information. L'information doit être accompagnée de la clef d'authentification de l'agent central.
```
void        PushInfo(in Key key, in ListInfo list);
```

3.3. Agent collecteur (interface ICollector)

Le servant ICollector est présent sur l'agent collecteur. IL met à disposition de l'agent central les méthodes suivantes :
– Il inscrit les agents centraux dans sa base de connaissance. Pour cela, il utilise la même méthode que les agents locaux.
```
Context         connect(in Key key) raises(EGeneral);
```
– Il gére les profils des agents centraux.
```
Profile     GetProfile() raises (EGeneral) context (\"ctx\");
void        SetProfile(in Profile prof) raises (EGeneral) context (\"ctx\");
```
– Il récupére et envoie l'information sur l'agent central en fonction du filtre qu'il doit utiliser pour cet agent. Pour cela il utilise la seule méthode de l'interface ICentralAgent.
```
void        PushInfo(in Key key,
                     in KindOf kind,
                     in Buffer data,
                     in TrustInfo trust) raises (EGeneral) context (\"ctx\");
```

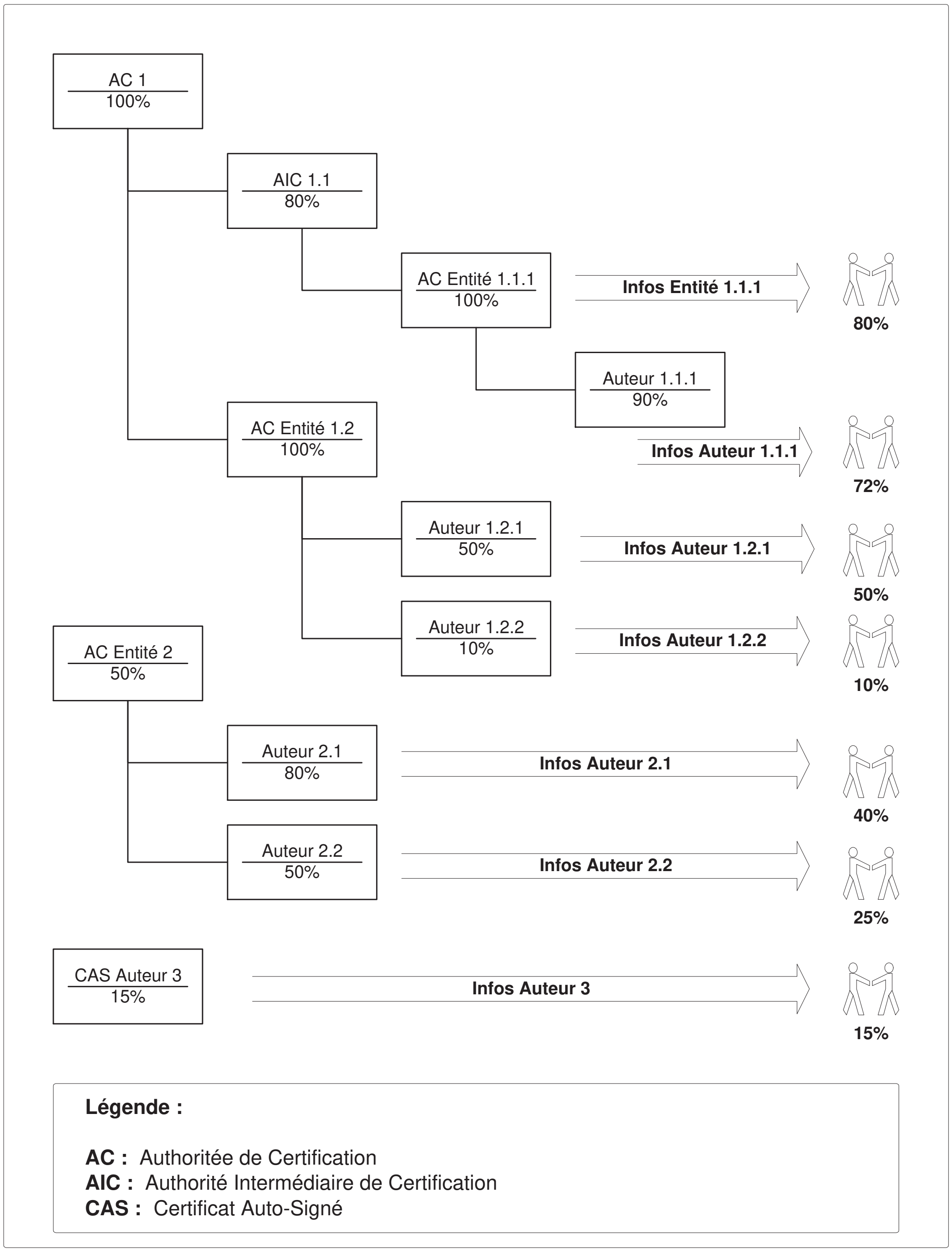
4. Système de confiance en l'information

Pour que l'utilisateur puisse filtrer et quantifier la pertinence de l'information en fonction de l'émetteur et de l'organisation auquel appartient l'émetteur, nous nous appuyons sur l'architecture des certificats X509. Une connexion de l'arborescence des identités sur une infrastructure PKI permet la vérification de la validité des certificats. Comme le montre le schéma ci-dessous, le processus de qualification de la pertinence de l'information utilise une arborescence d'identités des émetteurs.



L'utilisation d'une arborescence d'identités, permet à l'utilisateur, à travers son profil d'agent local, de donner des poids aux nœuds et branches de l'arbre, comme présenté dans le schéma ci-dessous. Chaque profil local possède ses propres poids sur cles nœuds de l'arbre de sorte que chaque utilisateur puisse filtrer les informations comme il le souhaite. Le poids des nœuds s'hérite de père en fils pour simuler la confiance que l'utilisateur possède envers une organisation ou une entité.

Exemple d'un profil de confiance



Dans le cas où le collecteur reçoit une informations non signée, l'agent central insère l'émetteur dans l'arborscence comme racine (il n'y a pas d'héritage possible) en utilisant les informations qui sont disponibles (adresse email, nom du site internet, émetteur de la news, etc...). Cela permet quand même à l'utilisateur de donner un poids à la source d'information.

5. Conclusion

Grâce à ce système d'agents communicants, le tri de l'information est automatique. Bien sûr, l'utilisateur de ce système a tout de même une configuration à effectuer, mais c'est minime par rapport au travail effectué par les méthodes de recherches classique. De plus, cette configuration ne se fait qu'une seule fois : un système de notation des news reçues permet à l'agent de réorienter le profil utilisateur. Par la suite, une mise à jour peut être effectuée pour réorienter le filtrage.

Cet outil est un atout quand on sait que la plupart des administrateurs système passent beaucoup de temps à trier les informations de sécurité qu'ils ont reçu par mail ou par forums. De plus, les systèmes de mailing-list ou newsgroup ne proposent pas de tri en fonction de l'identité de l'informateur. La méthode multi-agents est un gain de temps pour les personnes qui ont un besoin d'informations fraiches, utiles et vérifiées comme les administrateurs systèmes et réseaux, mais aussi pour les auditeurs et les laboratoires de veille technologique. La quantité et la qualité du flux d'information internet ne doit pas être un frein pour la diffusion et la crédibilité de l'information, il est donc nécessaire de mettre en place les outils qui traitent le maximum d'informations mais ne présentent que celles qui intéressent l'utilisateur.

Références

[1] G.M.P. O'Hare et N.R. Jennings. *Foundations of Distributed Artificial Intelligence*. John Wiley & Sons, Inc., 1996, New York, USA.

[2] G. Weiss, éditeur. *Multiagent Systems : A Modern Approach to Distributed Artificial Intelligence*. MIT Press, 2000.

[3] N. El Kadhi et M. Ben Ahmed. Multi agent global securing system (M.A.G.SE.S). *IASTED International Conference on Computer Systems and Applications (CSA'98)*, Irbid, Jordanie, Mars 1998.

[4] D. Benech, T. Desprats, et Y. Raynaud. A KQML-CORBA based architecture for intelligent agents communication in cooperative service and network management. *IFIP/IEEE international Conference on Management of Multimedia Networks and Services'97*. Montréal, Canada, juillet 1997.

[5] R. Khosla et T. Dillon. *Engineering Intelligent Hybrid Multi-Agent Systems*. Kluwer Academic Publishers, 1997.

[6] J. Feghhi, J. Feghhi, et P. Williams. *Digital Certificates : Applied Internet Security*. Addison Wesley, 1998.

[7] C. Gransart, P. Merle, et J.-M. Geib. *CORBA : des concepts à la pratique*. Dunod, seconde édition, octobre 1999.

[8] L.N. Foner. A security architecture for multi-agent matchmaking. *Second International Conference on Multi-Agent Systems (ICMAS'96)*, décembre 1996.

[9] Peter Stone. *Layered Learning in Multiagent Systems*. MIT Press, mars 2000.

[10] T. Finin, R. Fritzson, D. McKay, et R. McEntire. KQML as an agent communication language. *Proceedings of the Third International Conference on Information and Knowledge Management (CIKM'94)*, ACM Press, novembre 1994.

[11] M. Henning et S. Vinoski. *Advanced CORBA Programming with C++*. Addison Wesley, 1999.

[12] P.H. Winston. *Artificial intelligence*. Addison Wesley, seconde édition, 1984.