

---

# Sécurisation d'environnements CORBA : Le cas des coupes-feux pour les accès Internet

Bruno TRAVERSON  
EDF Division R&D

---

© DIVISION RECHERCHE ET DÉVELOPPEMENT



Cet exposé est basé sur une expérimentation menée à la division R&D d'EDF, l'expérimentation SPEC (Sécurisation de Plates-formes et Environnements CORBA) démarrée en 98. Cette étude consiste à comparer plusieurs outils de sécurité associés aux offres logicielles autour de produits CORBA. En 98, les évaluations ont porté sur des services de sécurité applicative tels que *OrbixSecurity* et *DAIS Security*. En 99, l'étude se focalise sur des services de sécurité réseau et notamment les coupe-feux pour les accès *Internet*. Des tests ont été réalisés sur le produit *OrbixWonderwall* de la société IONA.

# Sommaire

---

- Introduction
- Contexte et problèmes à résoudre
- Éléments de réponse
- Conclusion et perspectives

---

© DIVISION RECHERCHE ET DÉVELOPPEMENT



Cet exposé est composé de deux parties principales. La première partie introduit les concepts utilisés dans l'exposé et les problèmes à résoudre. La deuxième partie présente des éléments de réponse : la spécification standard CORBA/Firewall de l'OMG et un retour d'expérience sur le produit *OrbixWonderwall* de la société *IONA technologies*.

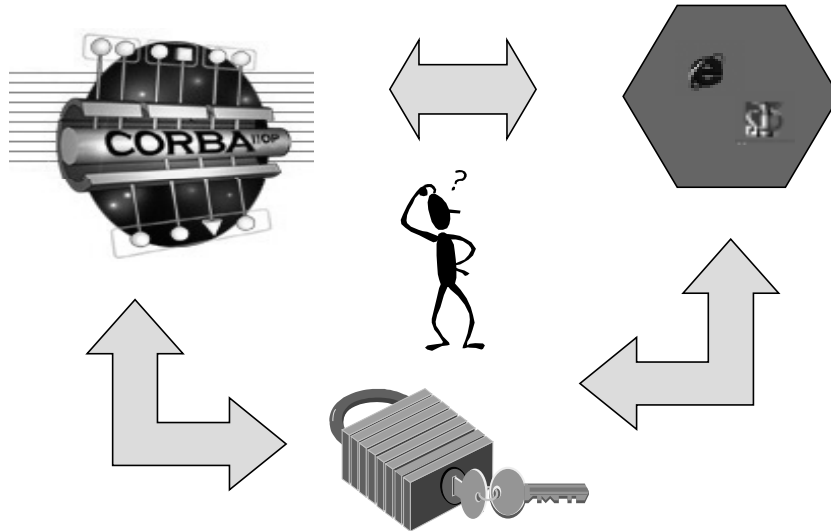
# Sommaire

---

- Introduction
- Contexte et problèmes à résoudre
- Éléments de réponse
- Conclusion et perspectives

Passons à la partie « Introduction ».

# Introduction



© DIVISION RECHERCHE ET DÉVELOPPEMENT



Les éléments à considérer sont : l'environnement CORBA, les accès Internet et la sécurité.

Pris deux à deux, les thèmes d'étude deviennent : les accès Internet à un environnement CORBA, la sécurisation des environnements CORBA, la sécurité sur Internet.

Pris tous les trois, le problème à résoudre se formule de la façon suivante : comment accéder via Internet à un environnement CORBA de façon sécurisée.

# Sommaire

---

- Introduction
- Contexte et problèmes à résoudre
  - L'environnement CORBA
  - Le couplage Internet/CORBA
  - La sécurisation d'environnements CORBA
- Éléments de réponse
- Conclusion et perspectives

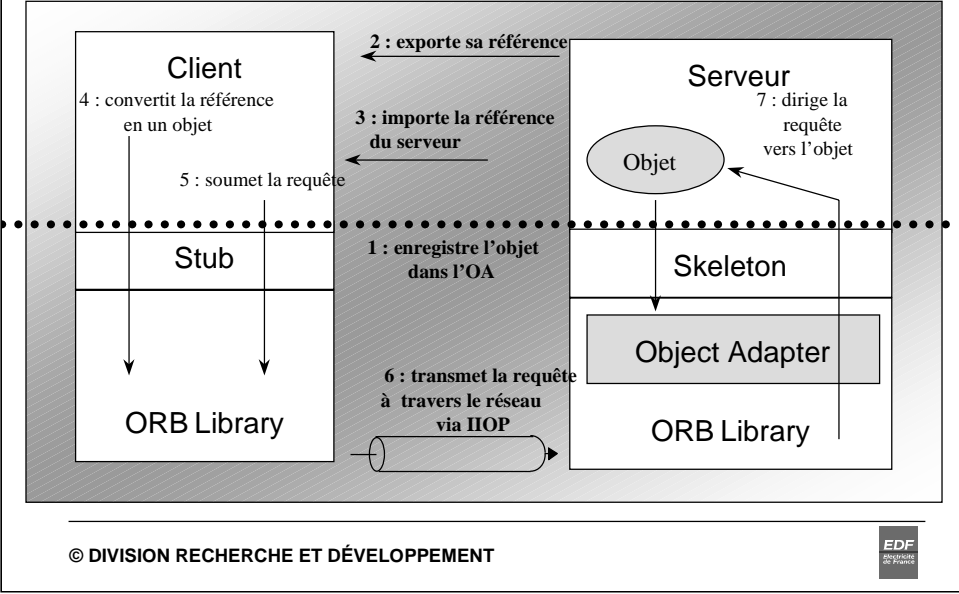
Cela nous mène à la deuxième partie. En partant de l'angle CORBA, nous allons tout d'abord examiner chacun des axes qui en part: le couplage Internet/CORBA et le la sécurisation d'environnements CORBA. Puis, nous formulerons une liste de problèmes à résoudre.

# L'environnement CORBA

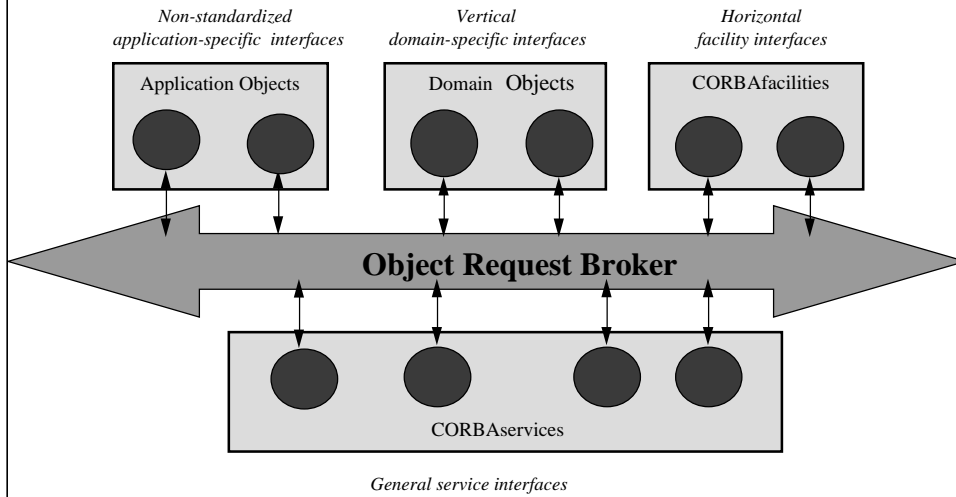
---

- CORBA (Common Object Request Broker Architecture) est une spécification de l'OMG (Object Management Group)
  - Vise à définir un système qui garantit deux propriétés fondamentales, la portabilité et l'interopérabilité, en milieu hétérogène.
  - Plusieurs offres commerciales: ORB (Object Request Broker), OTM (Object Transaction Monitor), AS (Application Server).

# Le modèle CORBA



# Le modèle OMA



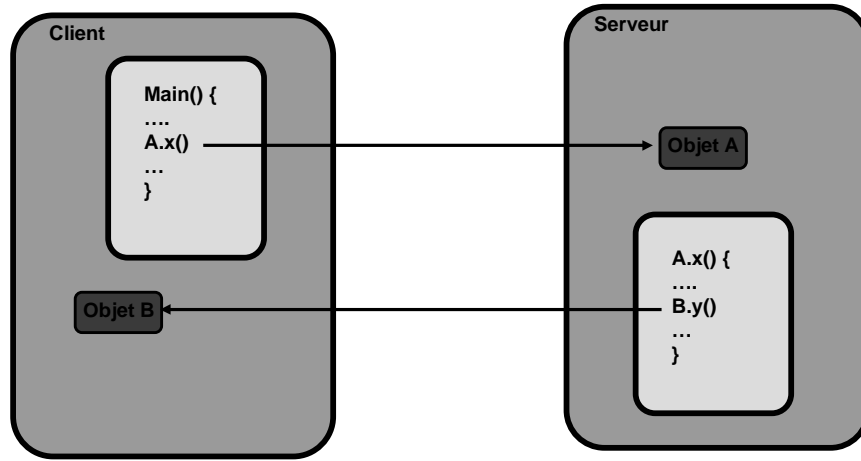
# Les références d'objet

---

Version	AdHost	NoPort	ObjId	Components
---------	--------	--------	-------	------------

↙ Adresse IP ou nom DNS de la machine qui héberge le serveur

# Les appels en retour



© DIVISION RECHERCHE ET DÉVELOPPEMENT

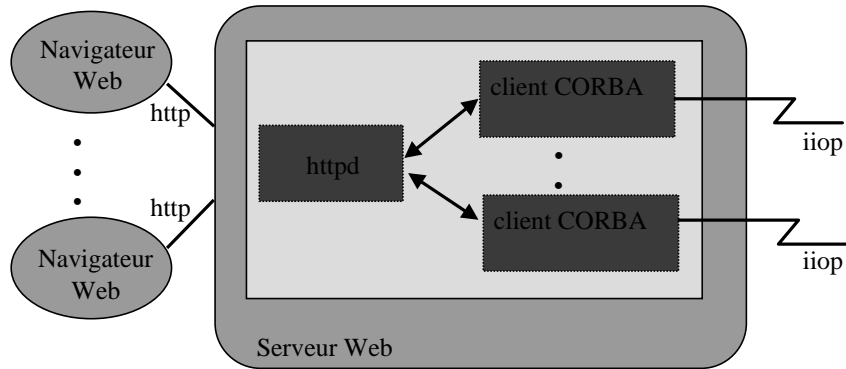


# Le couplage Internet/CORBA

---

- Enjeux
  - client léger
  - commerce électronique
- Architectures d'intégration
  - utilisation de servlets Java
  - utilisation d'applets Java

# Utilisation de servlets Java

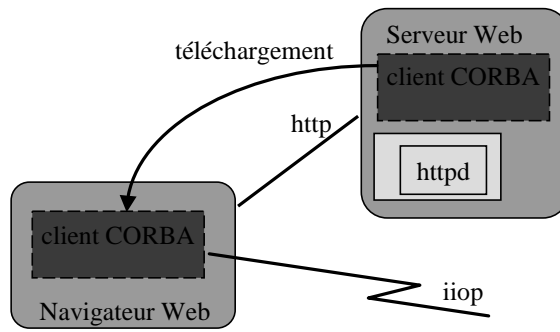


© DIVISION RECHERCHE ET DÉVELOPPEMENT



# Utilisation d'applets Java

---



## La sécurisation d'environnements CORBA

---

- **Spécification d'un service de sécurité CORBASEC**
  - identification/authentification,
  - contrôle d'accès,
  - confidentialité/intégrité,
  - non répudiation/audit,
  - définition de politiques sécuritaires.
- **Mises en œuvre KERBEROS ou SESAME**

---

© DIVISION RECHERCHE ET DÉVELOPPEMENT



L'identification permet d'attribuer une identité aux utilisateurs humains ou aux objets de l'application répartie. L'authentification permet de vérifier que ceux-ci sont bien ceux qu'ils prétendent être au moment de la communication. Utilisateurs humains et objets sont souvent regroupés sous le terme de *principal*.

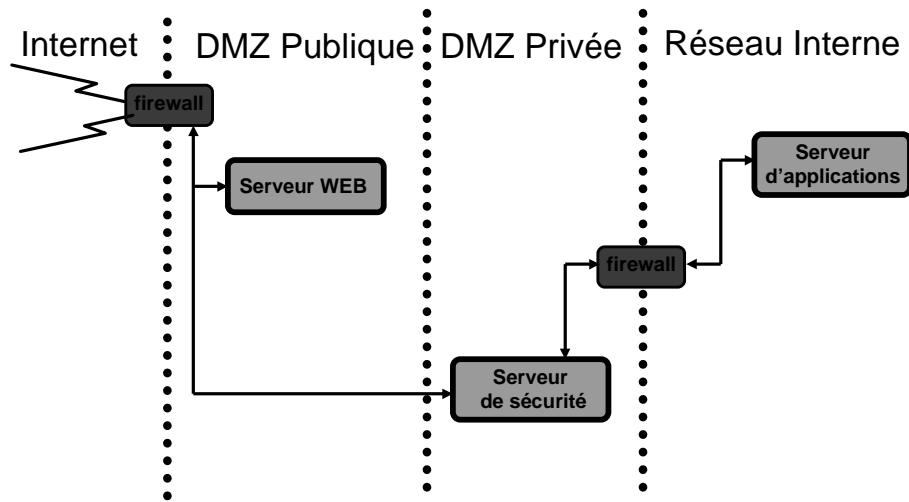
Le contrôle d'accès permet de décider si un utilisateur ou un objet demandeur peut accéder à un objet cible. La décision peut s'effectuer aussi bien sur les caractéristiques du demandeur que celles de l'objet cible. Ces caractéristiques sont appelées attributs qui peuvent être de deux types: identité ou privilège.

La confidentialité et l'intégrité s'appliquent en général aux communications entre objets et permet d'assurer que, respectivement, les messages échangés ne peuvent être ni lus (confidentialité) ni modifiés (intégrité) pendant leur transfert.

La non-répudiation fournit une garantie de preuve irréfutable de la participation d'un *principal* (cf. note 1) à une action aussi bien au niveau de la source que de la cible de l'action. La fonction d'audit doit permettre la traçabilité des actions.

La définition de politiques de sécurité permet de construire un domaine partageant des règles et des techniques communes à une communauté - par exemple une entreprise - et, éventuellement, des règles applicables entre communautés. Par exemple, la délégation d'identité ou de privilège permet la transmission ou non des attributs d'un *principal* vers un autre *principal*.

# Architecture des accès Internet



Inspiré du livre blanc d'Ifatec « Architectures objets Web/distribués »

© DIVISION RECHERCHE ET DÉVELOPPEMENT



## Problèmes à résoudre

---

- Transparence à la localisation
- Protocole IIOP
- Accès multi-sites dans le cas des applets
- Appels en retour

---

© DIVISION RECHERCHE ET DÉVELOPPEMENT



- Transparence à la localisation : service de base de CORBA alors que les coupe-feux sont basés sur des contraintes topologiques de localisation.
- Protocole IIOP: protocole utilisé dans CORBA mais généralement pas reconnu dans un coupe-feu classique.
- Accès multi-sites dans le cas des applets : accéder à d'autres sites que celui qui héberge le serveur http qui a permis de rapatrier l'applet.
- Appels en retour : callback ouvre une connexion sortante.

# Sommaire

---

- Introduction
- Contexte et problèmes à résoudre
- **Eléments de réponse**
  - La spécification CORBA/Firewall
  - Un exemple de réalisation
- Conclusion et perspectives

Passons à la partie « Eléments de réponse » avec une présentation de la spécification « CORBA/Firewall » et un retour d'expérience sur le produit OrbixWonderwall.

# La spécification CORBA/Firewall

---

- Spécification d'un coupe-feu capable de filtrer les requêtes IIOP
- Fonctions :
  - Contrôler les accès extérieurs aux serveurs CORBA internes,
  - Supporter IIOP (en plus de FTP et de HTTP),
  - Effectuer éventuellement un contrôle d'accès au niveau objet voire au niveau méthode,
  - Supporter des interfaces de gestion et de configuration.

---

© DIVISION RECHERCHE ET DÉVELOPPEMENT



La spécification CORBASEC, service de sécurité applicative défini par l'OMG, n'aborde pas l'utilisation des coupe-feux pour les accès Internet qui est traité dans un document à part appelé CORBA/Firewall et qui est encore au stade de soumission [OMG 98c]. Ce manque s'explique par le fait, qu'au niveau du bus CORBA, un objet cible n'a accès qu'aux *credentials* du client, ce qui ne permet pas de connaître sa localisation. Or, la technique des coupe-feux (ou *firewall*) est essentiellement basée sur des contraintes topologiques de localisation.

La spécification CORBA/Firewall comble ce manque en spécifiant un coupe-feu capable de filtrer des requêtes IIOP et ainsi d'interdire les accès extérieurs (par *Internet* notamment) à des objets CORBA à usage purement interne à l'entreprise.

Les fonctions que doivent remplir le coupe-feu sont les suivantes :

- Fournir/refuser les accès extérieurs à des serveurs CORBA internes,
- Traiter IIOP comme un protocole ordinaire (comme HTTP ou FTP, par exemple),
- Effectuer éventuellement un contrôle d'accès au niveau de l'objet, de la méthode,
- Supporter des interfaces IDL pour sa gestion et sa configuration.

Il existe deux sortes de coupe-feux selon qu'ils se situent au niveau transport ou au niveau applicatif.

## Les coupe-feux de niveau transport

---

- Permettre un contrôle d'accès au niveau transport
- TCP Proxy
  - substituer dans l'IOR du serveur l'adresse et le port du coupe-feu (proxification d'IOR)
- SOCKS Proxy
  - utiliser le protocole SOCKSv5 pour établir un canal de communication entre client et serveur par le biais d'un proxy.

Pour un coupe-feu de type transport, le contrôle d'accès ne peut s'effectuer qu'au niveau de l'adresse et du port de la source. Une première variété de coupe-feu de niveau transport, appelée TCP proxy, permet de substituer dans l'IOR du serveur l'adresse et le port du coupe-feu à la place de ceux du serveur : cette technique est appelée **proxification d'IOR**. Une deuxième variété de coupe-feu de niveau transport, appelée SOCKS proxy, utilise le protocole SOCKSv5 (IETF RFC 1928) qui a été conçu pour établir un canal de données par le biais d'un proxy entre un client et un serveur communiquant au-dessus de TCP ou d'UDP. Le proxy créé par SOCKS est transparent aux deux parties. SOCKS supporte également la négociation de plusieurs méthodes d'authentification (user/password, Kerberos ou SSL). Le serveur n'est pas modifié et le client a juste besoin d'être édité avec la bibliothèque SOCKS client.

## Les coupe-feux de niveau application

- Permettre un contrôle d'accès au niveau application
- GIOP Proxy
  - mode « normal »: le proxy est un intermédiaire de confiance qui peut lire les messages IIOP échangés.
  - Mode « passthrough »: le proxy ne fait que transmettre les messages, le contrôle d'accès ne se fait qu'à la connexion.

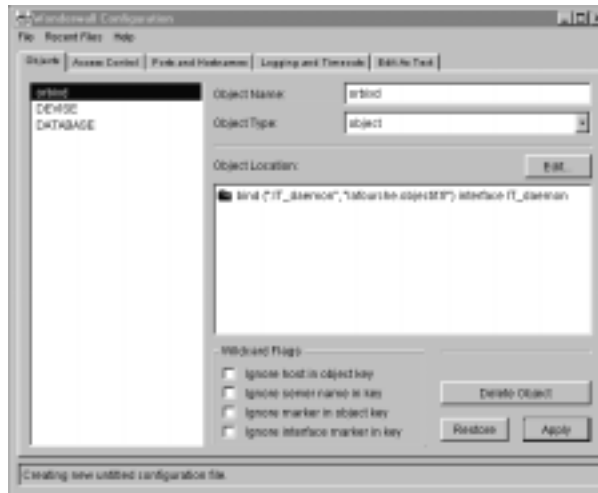
Les coupe-feux de type applicatif, appelés GIOP proxy, permettent une analyse des trames IIOP et donc un contrôle au niveau applicatif. Le contrôle d'accès peut s'effectuer sur l'objet cible ou sur une opération particulière d'un objet donné. Il existe deux modes de connexion pour ce genre de coupe-feu. Le mode " normal " est le mode où le proxy joue le rôle d'intermédiaire entre le client et le serveur : il joue le rôle d'un serveur pour le client et de client pour le serveur. Il doit donc être capable de lire et autorisé à analyser les messages IIOP échangés entre eux. Cela peut poser des problèmes de sécurité si le coupe-feu n'est pas un intermédiaire de confiance. De plus, le client et le serveur peuvent utiliser un algorithme d'authentification ou de chiffrement inconnu du proxy. Ces deux problèmes ont conduit un deuxième mode de connexion, appelé " passthrough ", où le coupe-feu ne fait que transmettre les messages IIOP entre client et serveur sans les analyser : il n'est soit pas capable soit pas autorisé à analyser le trafic entre client et serveur) ; Ce mode de connexion se rapproche d'un fonctionnement de coupe-feu de niveau transport bien qu'il fonctionne au niveau objet : on peut avoir un contrôle d'accès au niveau objet au moment de la connexion puis un le flot entre les deux entités n'est plus analysé.

# Un exemple de réalisation

---

- OrbixWonderwall™ de la société IONA.
- Produits:
  - Coupe-feu « GIOP Proxy »: iioproxy,
  - Fenêtre d'administration.
- Plusieurs modes d'utilisation

## Fenêtre d'administration du *iioproxy*



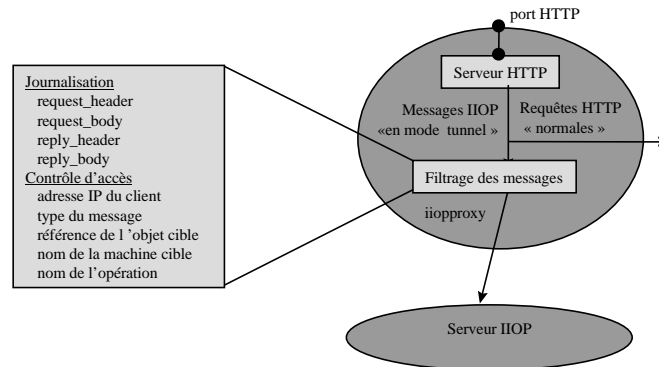
© DIVISION RECHERCHE ET DÉVELOPPEMENT



La fenêtre principale de l'outil de configuration est composée de cinq panneaux.

- Le panneau d'objets traite tous les objets dont les requêtes passeraient par le coupe-feu.
- Le panneau de contrôle d'accès permet d'éditer la liste des règles de sécurité à appliquer aux objets.
- Le panneau de ports et noms de machine permet de configurer le coupe-feu pour ses fonctions de serveur Web et de proxy IIOP.
- Le panneau de journalisation et des dépassements de délai, propose différentes options sur le contenu du fichier d'audit et sur les seuils de dépassement de délai.
- Le dernier panneau permet d'éditer toute la configuration de l'agent à l'aide d'un petit éditeur de texte intégré.

# Messages IIOP en mode « tunnel »

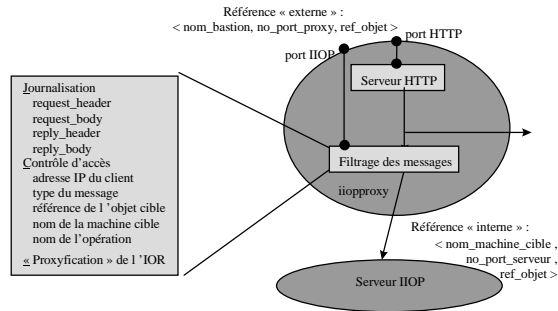


© DIVISION RECHERCHE ET DÉVELOPPEMENT



L'agent *iioproxy*, vu de l'extérieur, se comporte comme un serveur Web et reçoit les messages IIOP encapsulés dans des requêtes HTTP. Il peut alors les analyser pour effectuer un contrôle d'accès sur les différentes zones des messages ou les journaliser.

# Messages IIOP en mode « passerelle »

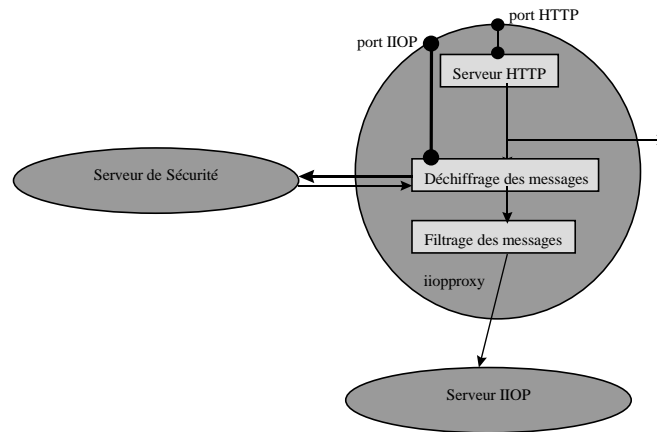


© DIVISION RECHERCHE ET DÉVELOPPEMENT



L'agent *iioproxy* se comporte comme une passerelle des messages IIOP en s'appuyant sur une technique de proxyfication d'IIOR. Il analyse les références d'objets CORBA se trouvant dans les messages pour les transformer et les transmettre aux serveurs d'application.

# Chiffrage des communications externes

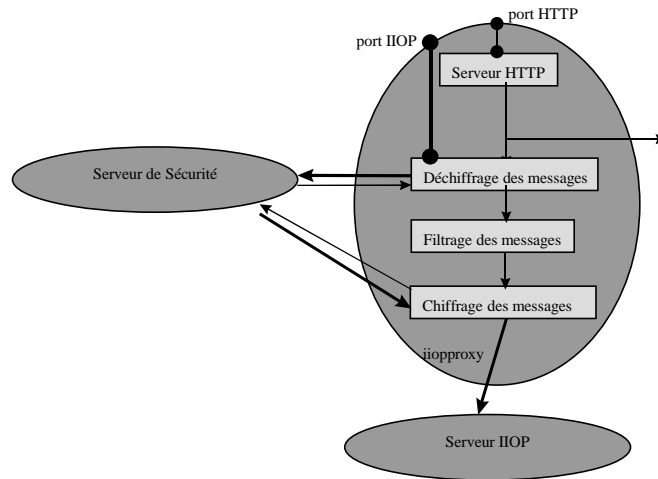


© DIVISION RECHERCHE ET DÉVELOPPEMENT



Ce niveau permet l'utilisation d'un service de sécurité externe à l'agent *iioproxy* qui est capable de déchiffrer les messages envoyés par les clients externes. Ce serveur doit être déclaré dans le fichier de configuration *iioproxy.cf* de l'agent.

# Chiffrement des communications internes



© DIVISION RECHERCHE ET DÉVELOPPEMENT



Le service de sécurité externe à l'agent *iioproxy*, vu dans la section précédente, peut également être utilisé pour chiffrer les messages envoyés aux serveurs d'application internes.

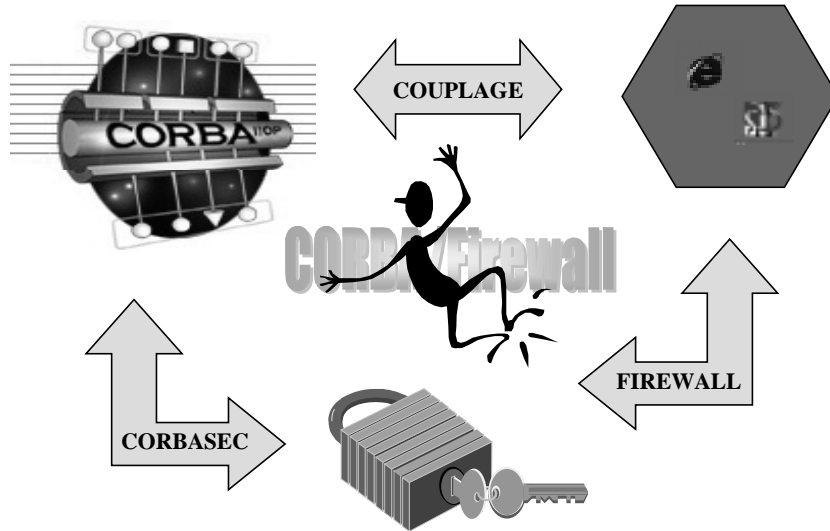
# Sommaire

---

- Introduction
- Contexte et problèmes à résoudre
- Éléments de réponse
- Conclusion et perspectives

Passons à la partie « Conclusion et perspectives ».

# Conclusion



© DIVISION RECHERCHE ET DÉVELOPPEMENT



Nous avons vu que des solutions existent pour chacun des axes et que la spécification CORBA/Firewall apporte des éléments de réponse pour l'intégration des trois éléments.

# Résolution des problèmes

---

- **Transparence à la localisation**
- **Protocole IIOP**
  - => définir des coupe-feux capables de filtrer les requêtes IIOP
- **Accès multi-sites dans le cas des applets**
  - => proxification d'IOR
- **Appels en retour**
  - => modification de IIOP (version 1.2 symétrique)

# Perspectives

---

- **Sécurisation de niveau réseau**
  - SSL : Secure Socket Layer
- **Sécurisation de niveau application**
  - CORBASEC v2
- **Sécurisation dans les modèles de composants**
  - CCM, EJB, COM+

---

© DIVISION RECHERCHE ET DÉVELOPPEMENT



Le service de sécurité SSL (Secure Socket Layer) apporte l'authentification, la confidentialité et l'intégrité des communications réseaux. L'authentification permet aux applications de vérifier l'identité d'autres applications. La confidentialité assure que les données transmises entre les applications ne sont pas capturées ou comprises par un intermédiaire. L'intégrité permet aux applications de détecter les substitutions ou les modifications effectuées durant le transport de données sur le réseau.