

Rendus des étudiants

MIDDLEWARE

Sommaire

1. ALBERT Frédéric albert.frederic@orange.fr Middlewares.....	2
2. ALI Loubna loubnaali@yahoo.fr Middleware ?	4
3. BENAHMED Mustapha <mustapha.benahmed@laposte.net> Synthèse d'informations tirées d'Internet sur la notion de Middleware	6
4. CHERIF Fethi cfethi@yahoo.fr Middleware.....	7
5. ELKOSANTINI Sabeur skossentini@yahoo.fr Middleware.....	9
6. GRANDGIRARD Emilie egrandgir@9online.fr MIDDLEWARE.....	10
7. GRUAT-LA-FORME France-Anne France-anne.Gruat-la-forme@insa-lyon.fr Les middlewares	12
8. HOFFMANN Patrick hoffmannp@free.fr Middleware.....	14
9. HUERTA Abraham Eneida <n_huerta70@hotmail.com> Middleware	17
10. JABER Mayyad mayyad_j@hotmail.com Middleware.....	18
11. REYMONDON Francis <freymondon@yahoo.fr> Middleware : intergiciel ou les « logiciels du milieu ».....	20
12. SAEZ Hector <jaaipayofr@yahoo.fr> Middleware	21

ALBERT Frédéric
albert.frederic@orange.fr
Middleware

Middleware

Couche logicielle permettant des échanges de données entre des clients et des Mainframes. Il existe deux sortes de Middlewares : les middlewares transactionnels (tels les ORB) et les middlewares asynchrones (tels les messageries interapplicatives et les outils EAI)

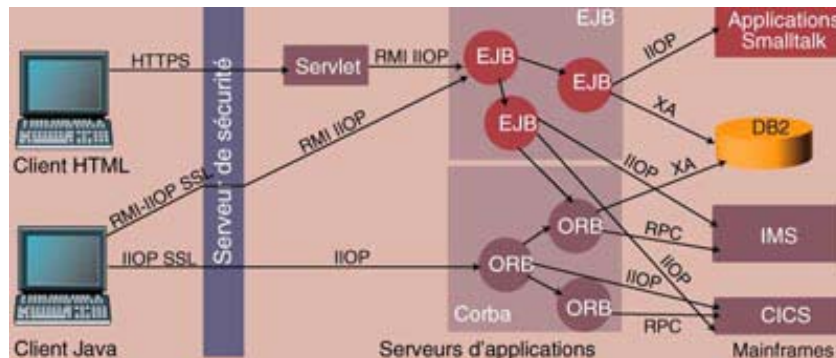
les middlewares transactionnels:

<http://www.01net.com/article/144349.html>

Thierry Jacquot, [01 Réseaux](#), le 01/03/2001 à 00h00

Moniteurs transactionnels et ORB ont leur place dans les infrastructures de commerce électronique. Leur rôle : faire en sorte que les sites supportent de fortes charges transactionnelles

Près de la moitié des middlewares vendus dans le monde en 2005 seront des brokers d'intégration, c'est-à-dire des logiciels de transformation et d'aiguillage des données. C'est, en tout cas, la prévision annoncée par le Gartner lors de sa conférence parisienne consacrée à l'intégration des applications. La progression rapide de ce type de produits se fait au détriment des MOM (Messaging-Oriented Middleware), les interfaces permettant des envois et des réceptions de messages asynchrones entre applications. [Renaud Edouard](#), [01 Informatique](#), le 15/06/2001 à 00h00



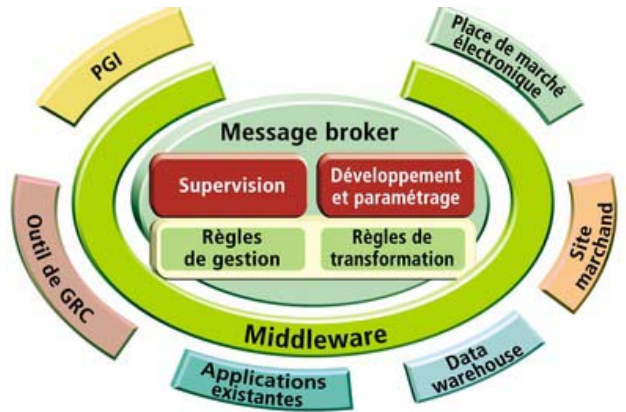
LA PLACE DES ORB DANS LES RELATIONS E-BUSINESS : LE CAS DE LA SOCIÉTÉ WINTERTUR

Ces ORB pourront être compatibles avec le modèle Corba, de l'OMG, ou avec les technologies COM+, de Microsoft.

les middlewares asynchrones:

Un coeur sur lequel se greffent les applications

Une architecture d'EAI typique est construite autour d'un moteur central, le message broker, qui exécute des règles de transformation de formats et de gestion pour recevoir, traduire et envoyer des messages en provenance de ou vers les applications.



ALI Loubna
loubnaali@yahoo.fr
Middleware ?

Les besoins de middleware :

Les technologies de l'information ne sont pas inertes, et la plupart des organismes disposent de plateformes hétérogènes dans leur infrastructure logicielle. Les applications doivent souvent être conçues pour fonctionner dans toutes sortes d'environnements. Un middleware est un programme qui connecte les applications ensemble, afin de leur permettre d'échanger des données.

Les systèmes informatiques sont de plus en plus complexes et surtout, grâce à l'essor de l'informatique nomade, se retrouvent partout. Alors qu'il y a quelques années la plupart des applications étaient conçues pour fonctionner dans un environnement d'exécution relativement bien connu et maîtrisé, cela n'est plus possible dans le contexte actuel, où les conditions d'exécution possibles pour une application donnée sont très variées.

D'une part, les développeurs ont à faire face à une grande diversité de plates-formes d'exécution et d'environnements logiciels associés. Le nombre et la variété de ces plate-formes n'a jamais été aussi grand, allant du simple téléphone portable doté de capacités minimales au cluster de plusieurs dizaines d'ordinateurs multi-processeurs, en passant par l'ordinateur de bureau. Pourtant, grâce au développement considérable d'internet, toutes ces machines hétéroclites participent du même réseau global et les applications actuelles doivent être capables de gérer cette diversité, voire d'en tirer partie.

D'autre part, la nature même de ces nouvelles plate-formes (assistants personnels, téléphones portables...), rend la tâche des développeurs encore plus complexe.

La solution:

L'utilisation d'un middleware est souvent préférable à la connexion en dur des applications entre elles, et pour y arriver, il doit être ajouté du code dans tous les systèmes qui ont besoin de discuter entre eux pour décrire les différentes interfaces. Un tel code coûte cher à écrire, et devient de plus en plus complexe lorsque le nombre d'applications connectées croît. En agissant comme une couche d'abstraction, le middleware permet aux différentes applications de discuter entre elles via un composant logiciel transparent.

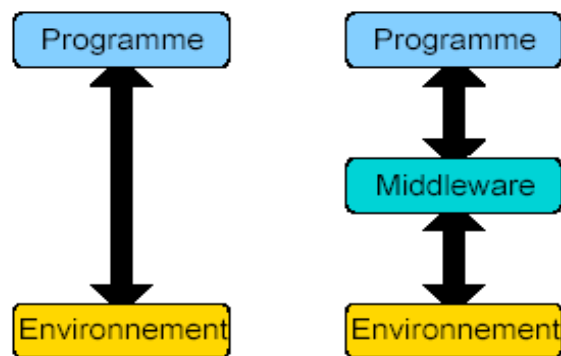
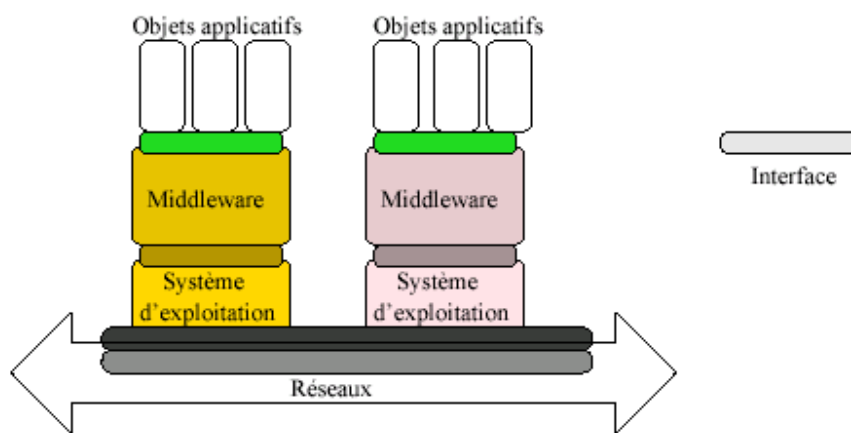


FIG. 1.1 – Introduction de la couche *middleware*



Middleware avantages:

Un des avantages de l'utilisation d'un middleware est la simplicité. Chaque application a juste besoin d'une interface avec le middleware lui-même, indépendamment du nombre de plateformes différentes à laquelle elle est intégrée. De plus, et grâce au concept d'abstraction, l'application n'a pas besoin d'être réécrite afin d'être intégrée dans les nouvelles plateformes.

Référence:

- http://www.jbase.com/news/flyers/french/MIDDLEWARE_F.pdf
- <http://rangiroa.essi.fr/riveill/rapports/2001/01-dea-david.pdf>
- <http://etna.int-evry.fr/cours/middleware/enseignement/polys/slidesS.pdf>

BENAHMED Mustapha
<mustapha.benahmed@laposte.net>
Synthèse d'informations tirées d'Internet sur la notion de
Middleware

Sites utilisés : www.webopedia.com et www.journaldunet.com

Définition Middleware ou Intergiciel : Logiciel autorisant des applications à communiquer les unes avec les autres, alors qu'elles n'étaient pas conçues jusque-là pour dialoguer ensemble. Il facilite ainsi l'accès à des données stockées dans des systèmes qui ne sont pas toujours compatibles.

(Source : http://encyclopedia.journaldunet.com/php/commun/definition.php?id=214&idctnr=21&id_cat=2&mode=1)

On peut ainsi dire que le middleware est une affaire de « plomberie » inter-applicative puisqu'il s'occupe de véhiculer les données entre les applications.

(Source : « L'EAI en sept questions » par [Cyril Dhénin](#), JDNet. http://solutions.journaldunet.com/0104/010424_eai.shtml)

Typologie Il existe différents types des middlewares:

- Par Messages : permettent à deux applications de communiquer par le biais de messages (échange par simple codage/décodage des messages).

- RPC (Remote Procedure Call)

C'est un protocole qui permet à un programme (à partir de l'ordinateur qui l'abrite) d'exécuter un autre programme sur un ordinateur serveur. Un développeur système qui utilise RPC n'a pas besoin de programmer de procédures spécifiques au serveur. En effet, le client envoie au serveur un message avec les bons arguments et le serveur lui retourne un message contenant les résultats du programme exécuté.

- Object Request Brokers (ORBs)

Composant du modèle de programmation CORBA qui joue le rôle de logiciel intermédiaire entre un client et un serveur. Dans le modèle CORBA, un client peut invoquer un service sans savoir quels serveurs sont reliés au réseau. Ce sont les différents ORBs qui reçoivent les requêtes, les acheminent vers les serveurs appropriés, et se chargent par la suite de retourner les résultats au client.

- Les moniteurs de transactions (Transaction Processing monitor) : Logiciel qui assume le contrôle des programmes utilisateurs exécutant des transactions et assure les services suivants: accès au réseau, gestion des messages, mise en oeuvre des programmes d'exécution des transactions, accès au fichier (gestion des données).

- Les systèmes d'accès aux bases de données

- Les environnements d'informatique distribuée (Distributed Computing Environment)

(Source : <http://www.webopedia.com/TERM/M/middleware.html>)

CHERIF Fethi
cfethi@yahoo.fr
Middleware

Le Middleware couvre toutes les couches logicielles qui permettent à des applications de communiquer à distance dans toute architecture, permettre à un programme de s'exécuter sur plusieurs machines reliées par un réseau (internet-intranet), Middleware c est l'intersection de plusieurs domaines d'informatique (Système d'Exploitation, réseau, langage de programmation...). Elle introduit l'aspect programmation orienté objet.

Permet de mettre en oeuvre des serveurs:

- serveur web, fichiers, BD, ...

- effectuant des traitement quelconque:CORBA, EJB, NET,...

les information échanger partager sont de nature différentes et sous des formats qu'il le sont tout autant(fichiers textes,images,message,données,etc...).

La diversité et la spécialisation des supports utilisés sont inversement proportionnelles à la facilité que les utilisateurs ont à se les échanger, les applications écrites dans différentes langages et sur système d'exploitation différents peuvent s'échanger sans aucun perte d'informations.

Le Middleware est un ensemble de programmes qui isole une application de l'ensemble des processus qui résultent de son lancement sur le système.

Le Middleware permet la communication entre des clients et des serveurs ayant des structures et une implémentation différentes.

Et pour que les clients permettent de trouver ces serveurs, le Middleware doivent fournir un moyen pour ce problème.

Différents types de matériels permettent de communiquer à distance (pc, mainframes, laptop, PDA,...).

Environnement Middleware pour client serveur:

-Sun	Java=support d'exécution universel
-OMG	CORBA= Middleware universel
-Microsoft	Web Service = l'interopérabilité universelle

Référence bibliographique:

<http://www.debrauwer.com/mda.html>

http://www.adae.pm.gouv.fr/upload/documents/desfray_xml_journeadae.pdf

<http://deasi.lip6.fr/DEA-SI/THESES/Prop-Theses09-02%2F03.html>

http://gizmoeti.free.fr/mda/toutmda_html/node10.html

<http://www-valoria.univ->

<ubs.fr/Jacques.Malenfant/ALP.OCM/Journee2003/Nano.pdf>

<http://www-src.lip6.fr/homepages/Lionel.Seinturier/middleware/cs.pdf>

<http://www.e-nef.com/ai/orchid/rapports/rapport/node18.html>

ELKOSANTINI Sabeur
skossentini@yahoo.fr
Middleware

Middleware

Les middlewares sont des mécanismes fondamentaux du système d'information. Ils permettent le transport de l'information entre des clients et des serveurs ayant des structures et une implémentation différentes.

Les utilisateurs de systèmes d'information ont de plus en plus besoin d'échanger et de partager des informations. Ces informations peuvent être de nature très différentes et sous des formats qui le sont tout autant (fichiers textes, images, messages, données, etc...). Et, plus récemment ces informations peuvent être situées à des endroits différents c'est à dire sur des postes distants, sur des réseaux différents ou encore sur l'Internet.

Concrètement, un Middleware est un ensemble de programmes qui isole une application de l'ensemble des processus qui résultent de son lancement sur le système.

Il existe plusieurs types de Middleware :

- Les middleware orientés objet (MOO).
- Les middleware orientés message (MOM).
- Les middleware orientés transaction (MOT).

Les middlewares orientés objet (MOO)

Les serveurs d'applications couvrent une grosse partie des besoins rencontrés en entreprise, il arrive néanmoins qu'ils se révèlent insuffisants. Le développement de composants applicatifs peut alors se révéler nécessaire.

Les middlewares orientés messages (MOM)

Les middlewares orientés messages sont les outils idéals d'*échange de données* entre applications ou composants logiciels. Ces messages ont une nature complètement générique, ils peuvent représenter tous types de contenu aussi bien du binaire (image, objets sérialisés) que du texte (document XML). Leur principal avantage est la simplicité de mise en œuvre.

Les middlewares orientés transaction (MOT)

Ce sont des middlewares fortement couplés aux moniteurs transactionnels. Ils sont appelés aussi des méthodes d'appel à distance.

Références

Le middleware: <http://www.e-nef.com/ai/orchid/rapports/rapport/node18.html>

Neoxia TechConsulting:

<http://www.neoxia.com/fr/template.php3?menu=techno&body=techno-middle>

GRANDGIRARD Emilie
egrandgir@9online.fr
MIDDLEWARE

Le terme de middleware est apparu à la fin des années 80, mais les middleware se sont vraiment développés dans le milieu des années 90. En effet, ils ont fourni l'interopérabilité entre des machines distantes, et ont donc permis principalement aux applications client-serveur de communiquer. On ne peut donc pas parler de middleware sans aborder la notion d'architecture client-serveur.

Architecture client / serveur

Aujourd'hui, la plupart des applications d'entreprise fonctionnent en mode client-serveur : une partie de l'application est stockée sur un serveur (au moins les données). Le poste client interroge alors le serveur qui lui renvoie des résultats. Il existe plusieurs types d'architecture client-serveur.

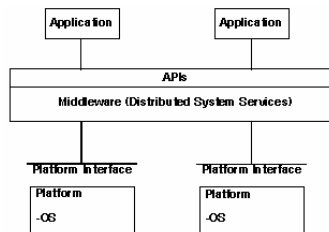
Au début, beaucoup choisissaient de placer la gestion de bases de données sur le serveur, et l'application (en terme de logique applicative) accompagnée de l'interface graphique chez le poste client. Mais, avec ce type d'architecture, le serveur a du mal à gérer les performances d'où le transfert de l'application sur le serveur.

Cependant, il faut éviter la surcharge du serveur : il est donc préférable d'adopter une architecture trois-tiers et d'ajouter un tiers chaque fois qu'on a une surcharge (n-tiers). C'est dans ce cadre qu'intervient le middleware : il est placé au milieu du client et du serveur pour gérer la logique applicative entre eux.

Définition middleware

Un middleware permet donc de lier efficacement les différentes composantes d'une application client-serveur (interfaces utilisateurs, applications et données). Comme son nom l'indique (« logiciel du milieu » en français), c'est un intermédiaire.

Plus généralement, il désigne toutes les couches logicielles qui permettent à des applications de communiquer à distance. Un middleware unifie ainsi l'accès à des machines hétérogènes en terme de CPU, OS, langage de programmation et représentation des données en mémoire.



Types de middleware

Il existe des middleware de plusieurs types, les principaux étant :

- les middleware d'accès aux données (drivers de base de données relationnelles par exemple) : ils mettent en communication les applications avec les différentes sources de données de l'entreprise ;
- les moniteurs transactionnels (par exemple Tuxedo) : leur fonction est de coordonner des transactions distribuées sur plusieurs machines ;
- les Message Oriented Middleware (ou MOM, par exemple MQ-Series) : il s'agit de serveurs qui servent d'intermédiaires dans la communication entre applications ; ils peuvent stocker des messages dans une file d'attente et attendre que l'application destinataire soit prête à recevoir le message qui lui est adressé.
- les Remote Procedure Call (appel de procédures à distance) : il offre la possibilité d'exécuter des portions d'application afin que le client soit déchargé d'une bonne part des traitements ;

- les Object Request Broker (Corba, DCOM, ou RMI) : leur but premier est de rendre plus transparente la communication avec des applications sur des machines distantes, grâce à ce qu'on appelle des « objets distribués ».

Avantages / Inconvénients

Le principal avantage des middleware est la transparence ; en effet, ils ne sont pas « vus » par les utilisateurs. Le principal inconvénient des middleware est la dépendance vis-à-vis du vendeur ; en effet, on est lié au vendeur pour tout ce qui concerne la maintenance et l'élargissement.

Conclusion : avenir

Beaucoup de middlewares existent aujourd'hui. De plus, vu la configuration actuelle des réseaux (de plus en plus hétérogènes), les middleware sont amenés à se développer et à évoluer.

Références :

- Client/Serveur Middleware
<http://www-src.lip6.fr/homepages/Lionel.Seinturier/middleware/cs.pdf>
- Middleware
<http://www.eecs.wsu.edu/~bakken/middleware-article-bakken.pdf>
- Middleware
<http://www.sei.cmu.edu/str/descriptions/middleware.html>
- What is middleware ?
<http://middleware.objectweb.org/>

GRUAT-LA-FORME France-Anne
France-anne.Gruat-la-forme@insa-lyon.fr
Les middlewares

Avant de commencer une synthèse sur le thème des middlewares, je tiens à préciser que les recherches sur le Net ont été moins fructueuses que pour les autres notions à développer. Néanmoins, les informations rassemblées permettent un exposé organisé de la manière suivante. Dans un premier temps, nous définirons ce qu'est un middleware et son rôle dans un environnement réparti. Puis nous verrons les modes de codage qu'il existe pour permettre à deux machines d'échanger des données. Nous concluons enfin par montrer les limites des middlewares.

Qu'est ce qu'un middleware ?

Un middleware joue le rôle d'une interface entre des clients et des serveurs ayant des structures de nature différente. Il permet l'échange d'informations dans tous les cas et pour toutes les architectures.

Il existe de nombreux middlewares différents selon l'environnement opérationnel dans lequel il va être exécuté et selon les fonctionnalités qu'il rempli. Il est important de choisir un middleware adapté à des besoins spécifiques sans quoi le protocole ou le codage peuvent être disproportionné, comme on le verra par la suite.

On distingue plusieurs grandes familles de middlewares dont les principales sont les logiciels d'accès aux bases de données, les services d'échanges entre objets applicatifs (MOM), les RPC et enfin les middlewares propriétaires, dédiés un environnement particulier.

Enfin, nous préciserons que l'interopérabilité ne passe pas forcément par l'utilisation de middlewares identiques, et vis versa, deux middlewares similaires ne peuvent pas forcément échanger. Le principal est que les API aient été identifiées, à l'origine de façon commune.

Protocole de communication et codage des données

Lorsque deux machines communiquent, que ce soit en mode point par point (TCP) ou lorsqu'elles doivent trier l'information qui leur est destinée, les données échangées doivent être compréhensibles des deux cotés. Pour cela le principe de l'encodage puis du décodage semble être la meilleure solution.

Un premier type d'encodage est d'associer à chaque donnée sa représentation ASCII. Néanmoins cette méthode comporte des risques d'erreurs si le récepteur ne connaît pas le type de données qu'il reçoit. la précision d'échange laisse donc à désirer....

Le codage XML, lui est un langage balise ce qui ne donne de l'importance qu'à la structure et à l'architecture du document à transmettre. La transmission est plus fiable mais néanmoins pas à la hauteur de ASN.

ASN est un langage de description objet, très précis mais lourd à mettre en place. c'est un langage explicite.

Lorsqu'une machine reçoit des données, elle ne sait pas nécessairement qu'en faire. C'est pourquoi, il est également utile de pouvoir coder les données définissant les opérations à effectuer. Nous retrouvons alors le protocole RPC qui permet d'encoder les données à employer.

Perspectives et limites des middlewares

Si nous récapitulons la synthèse, nous concluons que les middlewares sont conçus pour unifier l'accès à des ressources distantes, être indépendant des systèmes d'exploitation et du langage de programmation des applications. Nous avons également vu qu'il existe une multitude de middlewares. Les limites des middlewares se situent justement à ce niveau. Trop de middlewares impliquent que l'homogénéité est en danger. Certes des ponts relient les différents middlewares mais ce n'est qu'une solution provisoire. Après avoir renoncé à créer « LE » middleware unique (CORBA a eu cette prétention), l'OMG a décidé de passer au niveau supérieur d'abstraction pour parvenir réellement à enrayer l'hétérogénéité de la communication machine (MDA).

Bibliographie

WWW.recherche.enac.com

WWW.src.lip6.fr

WWW.a3dis.fr

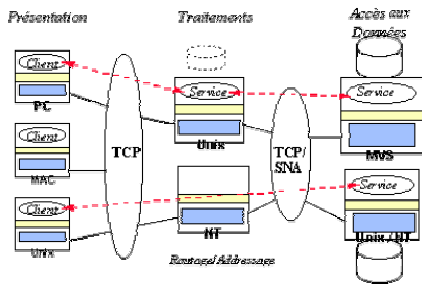
HOFFMANN Patrick
hoffmannp@free.fr
Middleware

Depuis longtemps les entreprises ont des applications et bases de données centralisées, qui étaient localisées sur des serveurs, et pouvaient être accédées depuis des postes clients. C'est ce qu'on appelle l'architecture 2-tiers.

Le concept est séduisant, et simple à utiliser dans un réseau local, mais devient plus gênant lorsqu'il s'agit de maintenir des logiciels clients sur des postes dispersés, et lorsque le débit transactionnel devient élevé ! Et les coûts de développement (nécessitant des spécialistes) sont en général élevés, malgré des difficultés d'interfaçage avec le Système d'information évidentes.

Le développement rapide et encouragé par tous les acteurs majeurs des technologies Internet/Intranet encouragent l'adoption du modèle 3-tiers (ou plus généralement n-tiers) composé du 1er tier = **Présentation** (ex : client léger : le navigateur), du 2^{ème} Tier = **Traitements** (ex : le serveur Internet), du 3^{ème} Tier = **Accès aux données** (ex : le serveur de Bases de données).

Modèle Technique Général des architectures " multi-tier "



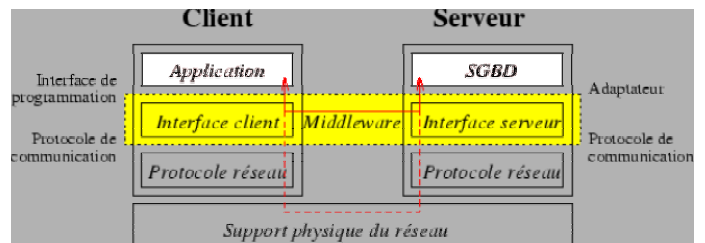
Et le middleware là-dedans ?

Littéralement "élément du milieu", c'est l'ensemble des couches réseau et services logiciel qui permettent le dialogue entre les différents composants d'une application répartie.

Le "Middleware" se définit donc comme l'ensemble des logiciels mis en oeuvre " en dessous des applications, au dessus des O.S. et entre les plates-formes " pour assurer les échanges entre les composants **Présentation**, **Traitement**

et **Accès aux données** dans les architectures Client/Serveur.

L'objectif principal du **middleware** est d'unifier l'accès et la manipulation de l'ensemble des services disponibles sur le réseau. Il masque ainsi la complexité des échanges inter-applications et permet ainsi d'élever le niveau des API utilisées par les programmes. Sans ce mécanisme, la programmation d'une application client-serveur serait extrêmement complexe et rigide.



Positionnement du middleware entre client et serveur

Le middleware se situe au niveau des couches Session, Présentation et Application du modèle OSI.

Principales fonctions attendues du Middleware

- la localisation des Services** (généralement sur la base d'un annuaire centralisé ou distribué)
- les échanges de messages** entre Clients et Services, en mode synchrone ou asynchrone.
- les mécanismes de formatage et conversion de données** (requis du fait de l'hétérogénéité des plates-formes).
- les fonctions de sécurité** (authentification, contrôle d'accès, chiffrement,...)
- l'administration** (configuration et exploitation) de l'ensemble des composants (système et application) des différentes plates-formes, généralement à partir d'un point central unique.

Choix du Middleware

Le choix du " Middleware " (et surtout sa mise en place) est déterminant pour une entreprise : elle peut opter pour un middleware proposé par un fournisseur de SGBD, qui sera très performant mais ne permettra pas, le plus souvent, l'accès à d'autres sources de données.

Les grandes entreprises ayant en général un environnement informatique hétérogène, cela devient un problème complexe dont la solution passe le plus souvent par l'intégration de plusieurs middlewares d'origines différentes : on en arrive à des clients lourds !

Exemples de middleware

SQL*Net (Interface propriétaire), ODBC (Interface standardisée), DCE

Bibliographie

Les types de middlewares :

<http://www.eurescom.de/~public-seminars/1997/DOT/slides/Session0/04Lamber/sld014.htm>

Documentation sur les architectures n-tiers

<http://remi.leblond.free.fr/probatoire/>

Autres

<http://perso.wanadoo.fr/philippe.baucour/pratiquer/tcp/tcparchitec.html>

<http://www.commentcamarche.net/initiation/client.php3>

http://perso.wanadoo.fr/thl/TechnoToile/Formation/Cours/4Tiers.htm#_Toc432498308

Pages anglaises

http://www01.giscafe.com/technical_papers/Papers/paper058/

<http://www-db.stanford.edu/~ullman/fcdb/jw-notes/middleware.html>

deux docs intéressantes :

Towards Model-Driven Middleware Maintenance

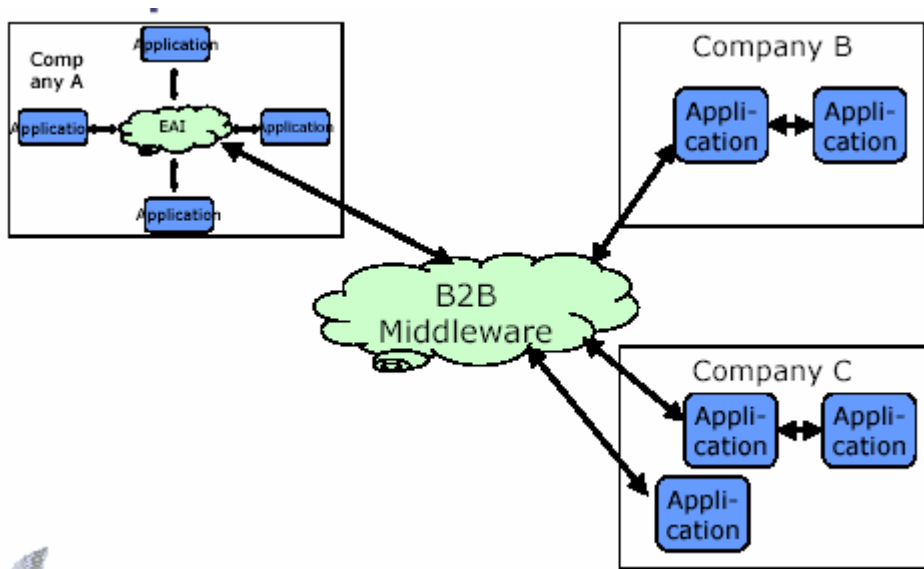
<http://www.softmetaware.com/oopsla2002/jahnkej.pdf>

A Model for Distributed System Services

<http://www.cis.upenn.edu/~lee/00cis640/papers/philip.pdf>

HUERTA Abraham Eneida
<n_huerta70@hotmail.com>
Middleware

Le logiciel personnalisé est un logiciel de connectivité qui se compose d'un ensemble de permettre les services qui permettent des processus multiples fonctionnant sur une ou plusieurs machines pour agir l'un sur l'autre à travers un réseau. Le logiciel personnalisé est essentiel aux applications d'unité centrale de migration aux applications client/serveur et à prévoir la communication à travers les plateformes hétérogènes. Cette technologie a évolué pendant les années 90 pour prévoir l'interopérabilité l'appui du mouvement aux architectures client/serveur. Les initiatives de logiciel personnalisé large-données de la publicité sont le Distributed Computing Environment de la base ouverte de logiciel COM/dcom communs de l'architecture de COBRA et du Microsoft de courtier de demande de l'objet de groupe de gestion d'objet



Adresses de sites Web visités

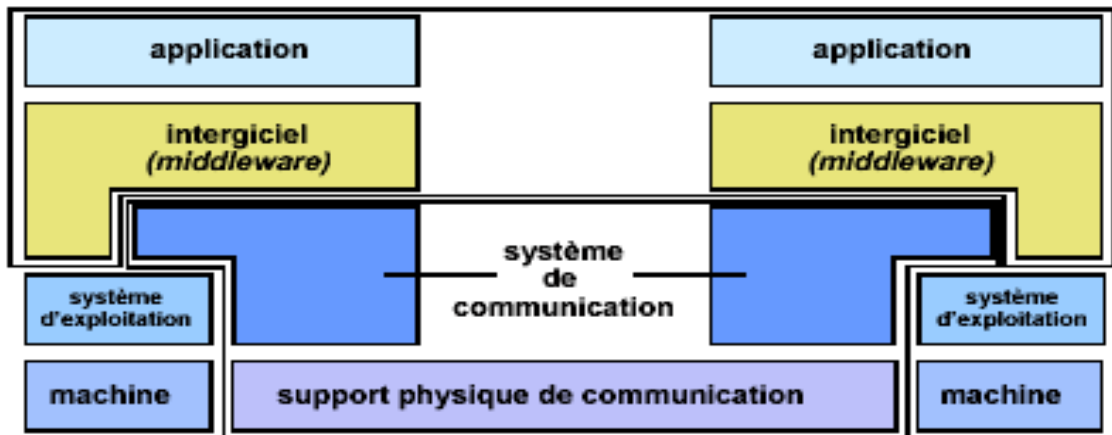
<http://www.iwaysoftware.com/middleware.html>

<http://www.sei.cmu.edu/str/descriptions/middleware.html>

http://www.iturls.com/English/TechHotspot/TH_41.asp

<http://techupdate.zdnet.com/techupdate/stories/main/0,14179,2850732,00.html>

L'intergiciel (middleware)



Le middleware a quatre fonctions principales

Fournir une interface ou API (Applications Programming Interface) de haut niveau aux applications

Masquer l'hétérogénéité des systèmes matériels et logiciels sous-jacents

Rendre la répartition aussi invisible ("transparente") **que possible**

Fournir des **services répartis** d'usage courant

Le middleware vise à faciliter la programmation répartie

Développement, évolution, réutilisation des applications

Portabilité des applications entre plates-formes

Interopérabilité d'applications hétérogènes

Les besoins des applications évoluent en permanence

Capacité de croissance

Qualité de service

Les applications réparties travaillent dans un environnement changeant

Mobilité **Logique** (code et données mobiles)

Physique (mobilité des utilisateurs, du matériel)

Connexion et déconnexion dynamique

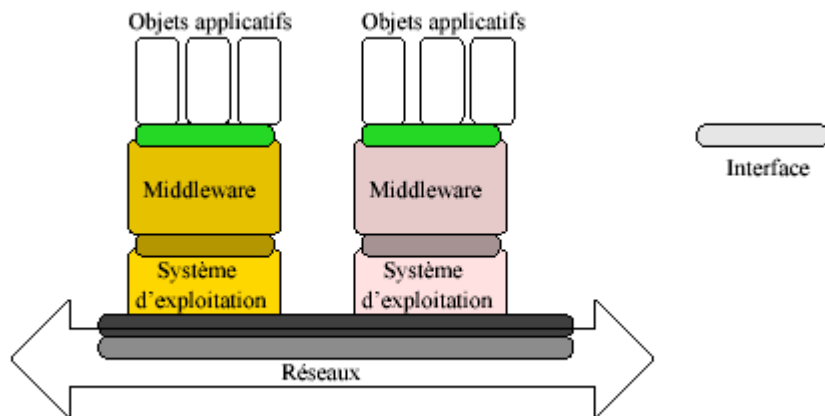
Qualité variable des communications

Réponse : adaptation du middleware et des applications

Découverte dynamique de services

Reconfiguration dynamique

Comportement adaptable



Un middleware fournit :

- Des interfaces de programmation standards (API),
- Des formats d'échange de données (couche présentation),
- Un protocole d'interopérabilité.

Middleware pour objets répartis

Héritage de deux technologies :

1. La programmation orientée objets : héritage, encapsulation et polymorphisme
2. Les RPC ou appel de procédure distante : répartition, hétérogénéité, emballage et déballage des données

<http://middleware2003.inf.puc-rio.br/>

<http://sardes.inrialpes.fr/people/krakowia/Enseignement/Option-M1/Flips/MWA-Intro.pdf>

<http://etna.int-evry.fr/cours/middleware/enseignement/polys/polyS.pdf>

<http://middleware.internet2.edu/>

<http://www.sei.cmu.edu/str/descriptions/middleware.html>

REYMONDON Francis
<freymondon@yahoo.fr>
Middleware : intergiciel ou les « logiciels du milieu »

La définition la plus générale de Middleware est la suivante :

« Un Middleware est un logiciel de communication qui permet à plusieurs processus s'exécutant sur une ou plusieurs machines d'interagir à travers un réseau. »

On distingue aujourd'hui quatre grandes classes de Middleware : l'exécution de transaction (Transactions Processing) qui est une classe de logiciel plutôt orienté "base de données", les RPC (Remote Procedure Calls) qui distribue l'exécution de routines sur un réseau, les MOM (Message oriented Middleware) qui permettent l'échange de données entre applications (ce qui nous intéresse ici), et les ORB (Object Request Broker) qui permettent la distribution d'objets sur un réseau de machines (CORBA).

Le plus fréquemment lorsqu'on parle de Middleware on entend :

Middleware est un terme générique pour tout programme qui sert de lien entre deux programmes déjà existants. (un exemple idiot : apache et mysql, le middleware peut donc être dans ce cas php, perl, java, ...). Autrement dit un middleware est un logiciel central autorisant des applications à communiquer les unes avec les autres par le biais de messages, alors qu'elles n'étaient pas conçues jusque-là pour dialoguer ensemble. Il facilite ainsi l'accès à des données stockées dans des systèmes qui ne sont pas toujours compatibles.

Quelle différence entre l'EAI et le "middleware" ?

Les middlewares, ne représentent en fait que le premier étage d'une offre d'EAI. En effet, les logiciels de middleware s'occupent de véhiculer les données entre les applications. Il s'agit par exemple des bus logiciels comme MQ Series d'IBM ou encore Rendezvous de Tibco.

A partir d'une carte des flux, le logiciel d'EAI prend en charge la transformation des données et joue ainsi le rôle de traducteur entre les applications. Enfin, une fois les données traduites, le logiciel d'EAI s'occupe de les router et s'appuie à cette fin sur les fameux middlewares. En résumé, alors que le middleware reste une affaire de plomberie inter-applicative, l'EAI est avant tout une affaire de processus.

Sources bibliographiques principales :

<http://www.recherche.enac.fr/~alliot/middle.pdf>

<http://rangiroa.essi.fr/riveill/rapports/2001/01-dea-crevola.pdf>

http://phortail.org/webntic/Le_middleware-54.html

http://www.linuxfrench.net/article.php?id_article=825

SAEZ Hector
<jaaipayofr@yahoo.fr>
Middleware

Le middleware est la couche que permet la coordination entre les composantes, cette couche est situé entre l'application et le réseau.

Le middleware permet transparence sur la distribution et il résout la hétérogénéité du:

- Hardware.
- Systèmes Opératifs.
- Réseaux.
- Langages de programmation.

Le middleware fourni aussi l'environnement de développement et d'exécution des applications distribuées.

On peut diviser le middleware en différentes couches:

- Middleware d'infrastructure: il augmente et encapsule les mécanismes de communication et de coïncidence des systèmes opératives pour créer composants de réseau orientés sur objet.
- Middleware de distribution: il prolonge le middleware d'infrastructure pour créer un modèle de programmation d'haut niveau. Ici on parle d' ORBs (COM, RMI, CORBA).
- Middleware de services communs: il définit les services d'haut niveau indépendants du domaine (notifications d'événements, persistance, ...)
- Services spécifiques du domaine: ils sont construits pour satisfaire les demandes de quelques domaines en particulier.

Types de middleware:

- Orientés au transitions: BEA Tuxedo, Microsoft Transaction Server.
- Orientés au messages: IBM MQSeries, Microsoft Message Queue.
- Procédures: Unix RPC.
- Orientés aux Objets: Corba, Java RMI, Microsoft DOTCOM.

Links:

<http://middleware.internet2.edu/>

Documents:

www.w3.org/2002/ws/chor/3/03/WSCWG-F2F-03-2003.ppt

Cours:

Cours d'été de l'université de Burgos, Juillet 2003 "Techniques avancées du langage Java".

Cours de Middleware:

<http://www-src.lip6.fr/homepages/Lionel.Seinturier/middleware/>