

EAI

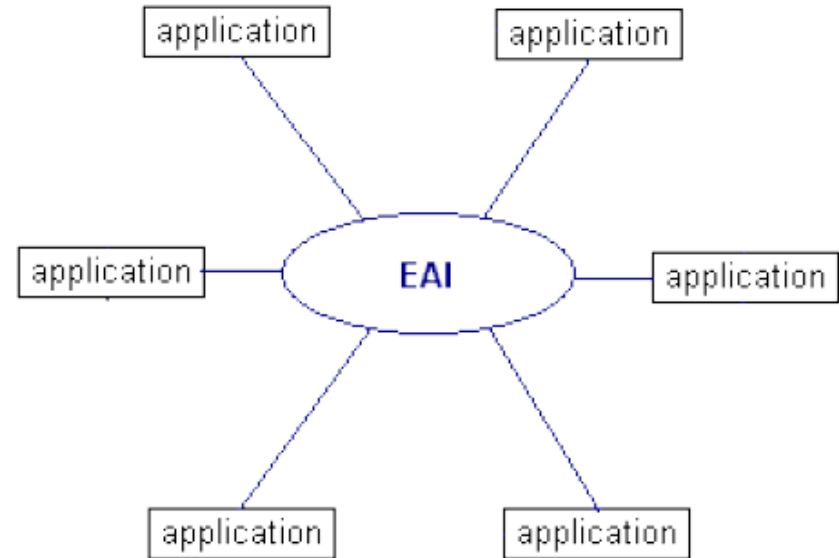
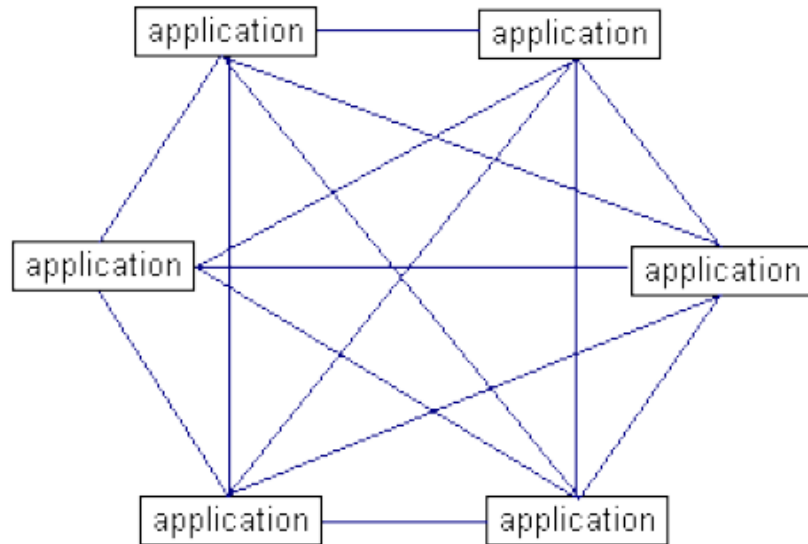
# Définition

- L'EAI est une notion ancienne mais toujours d'actualité. En effet, le besoin de faire communiquer des applications développées à des moments différents, dans des technologies différentes et de manière indépendante est réel si l'on veut au sein d'une entreprise aboutir à un système d'information global et performant.
- L'ERP est une première réponse à cette problématique. Elle ne couvre cependant pas tous les besoins de l'entreprise et nécessite une installation logicielle (et souvent matérielle) nouvelle avec tous les problèmes d'organisation et de formation qu'elle implique. En résumé, les ERP remplacent une partie du système d'information seulement, mais pas la totalité des applications métiers
- les ERP sont conçus pour communiquer parfaitement entre eux, mais pas avec d'autres applications.
- L'EAI se veut plus large et tient compte au mieux de l'existant. Le problème se pose souvent dans les termes suivants : étant donné un ensemble d'applications "communicantes" comment faire pour intégrer une nouvelle (qui peut d'ailleurs être ancienne) application ?

# Définition

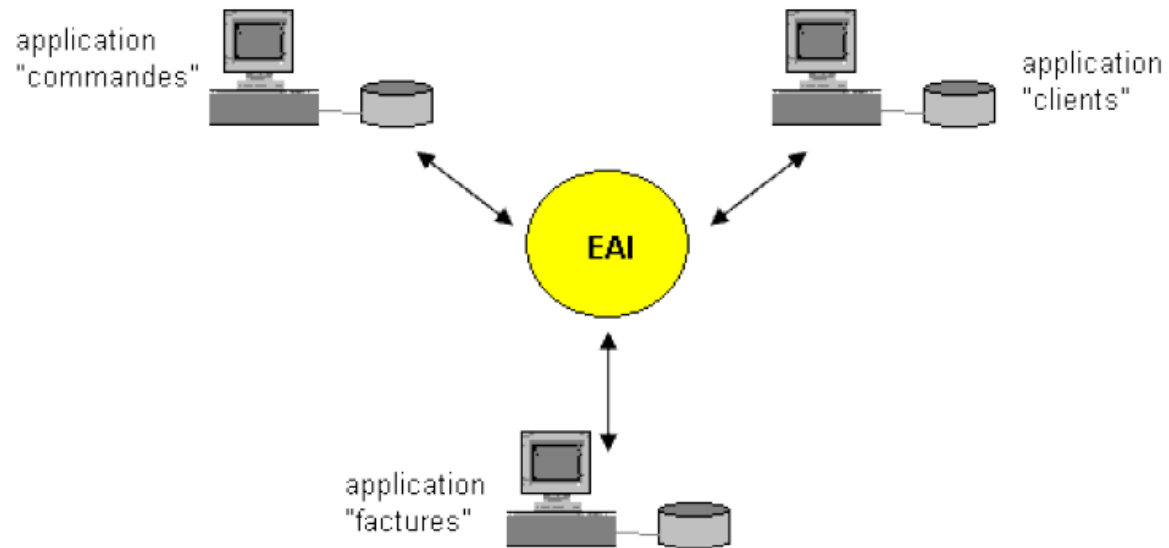
- "L'EAI est un ensemble de méthodes, d'outils et de services qui concourent à faire communiquer des applications hétérogènes dans le cadre de l'entreprise traditionnelle, répartie ou étendue."

# Modèle spagetti et solution EAI



On peut envisager de relier toutes les applications pour qu'elles communiquent par des interfaces spécifiques. Si il y a  $n$  applications, on sera donc conduit à réaliser  $n(n-1)/2$  interfaces ; donc pour 100 applications, 4950 interfaces !

# Exemple



- Exemples de plateformes EAI
  - WebMethods / Webmethods Fabric
  - Oracle Weblogic
  - IBM / Websphere Business Modeler
  - Microsoft / BizTalk

# Exemple

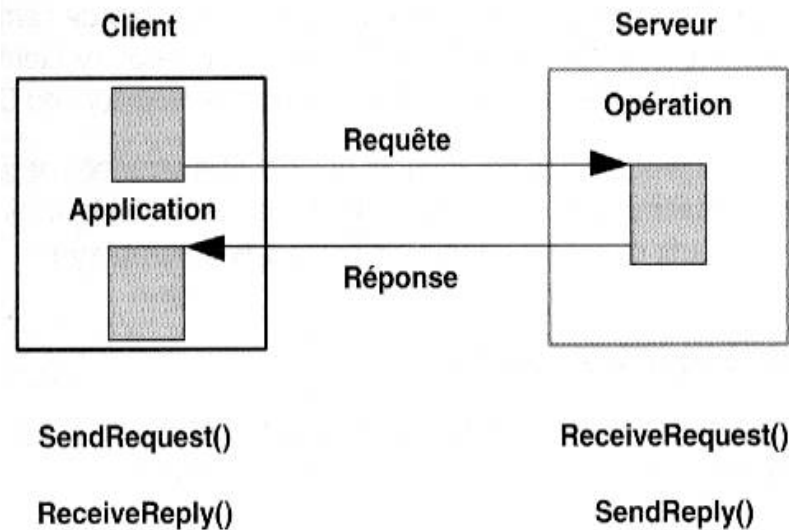
on peut distinguer 4 fonctions de l'EAI, en supposant que les systèmes informatiques de l'entreprise sont reliés par un Intranet :

- l'échange de données : il s'agit de prendre des données dans une application et de les intégrer dans une autre. Puisque en général, les systèmes informatique sont hétérogènes, il faut choisir des technologies comme les services Web.
- conversion de données : elle est souvent nécessaire pour que les applications communiquent. Cette conversion peut, par exemple, être effectuée par ETL, XSLT.
- routage de données : transmettre des informations vers la bonne application destinataire. SOAP peut être un bon candidat pour cette opération.
- gérer le processus global de bout en bout. Ceci implique la définition de processus métiers et on peut utiliser BPML.

- **Le modèle de base EAI** Dans sa version d'origine, une plateforme EAI est construite sur 4 couches :
  - le transport des données
  - l'extraction d'informations et l'insertion d'informations entre applications
  - les composants
  - le moteur d'intégration

# Modèles de communication entre applications

- Protocoles de type question-réponse



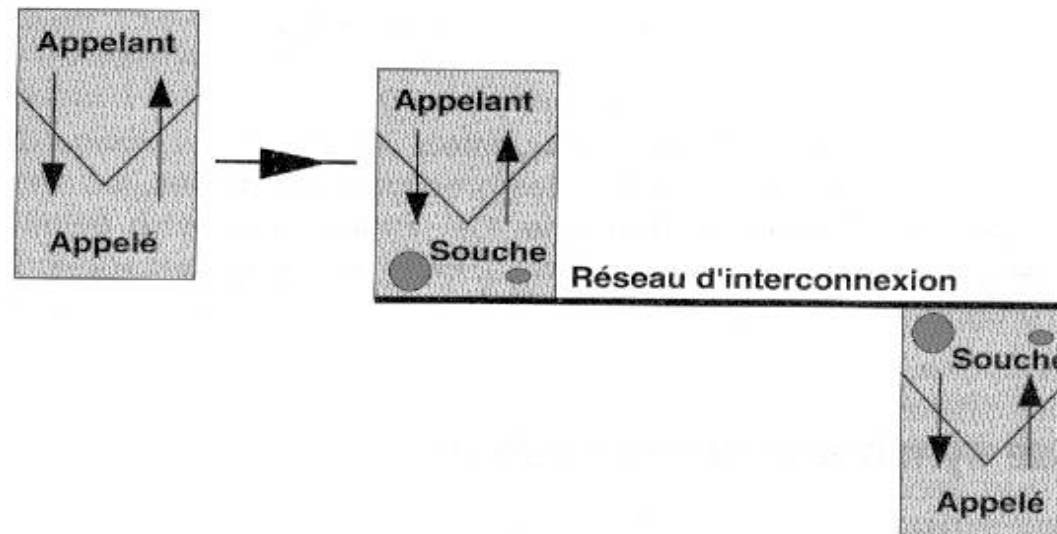
- Des opérations de transport permettant d'envoyer (Send) et de recevoir (*Receive*) des messages sont typiquement utilisées à cette fin. Les protocoles de question-réponse peuvent s'implémenter au-dessus d'une couche session. Celle-ci doit alors être établie entre le client et le serveur par des primitives de connexion (*Connect*), puis de déconnexion (*Disconnect*) en fin de session. Ils peuvent aussi être implémentés directement au-dessus d'un service de datagrammes, qui permet d'envoyer des messages à des ports associées aux clients et au serveur.

## 2 Assemblage-désassemblage des paramètres

- Lors de l'émission d'une requête, les paramètres doivent être arrangés et codés sous forme de message: c'est **l'assemblage**. A l'arrivée, ils doivent être remis en format interne de manière symétrique à partir du message reçu: c'est le **désassemblage**.
- **Assemblage (*Marshalling*)** : Procédé consistant à prendre une collection de paramètres et à les arranger et coder en format externe pour constituer un message à émettre.
- **Désassemblage (*Unmarshalling*)**: Procédé consistant à prendre un message en format externe et à reconstituer la collection de paramètres qu'il représente en format interne.

# Appel de procédure à distance

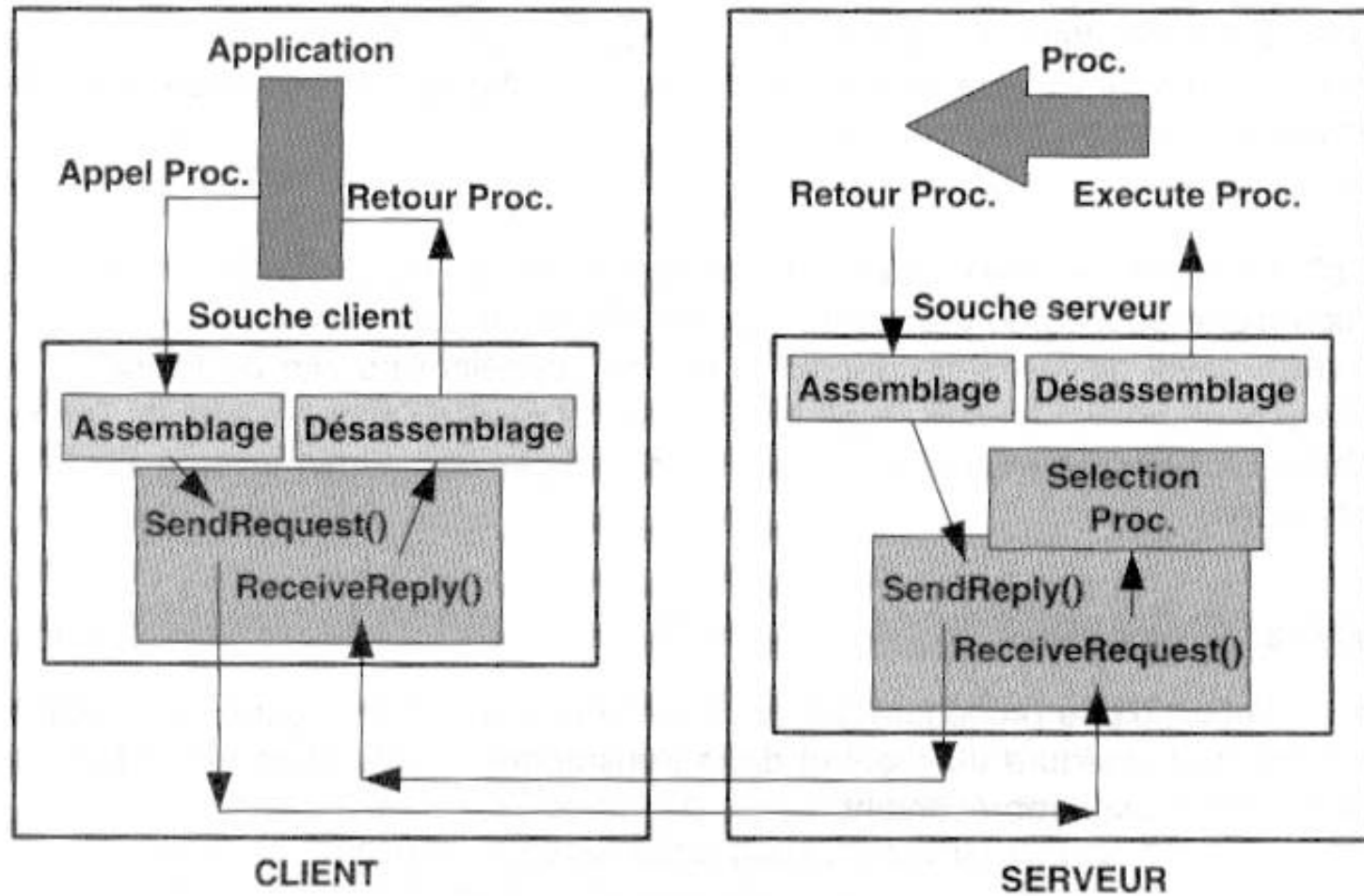
- Dans un système centralisé, l'appel d'opération (fonction ou procédure) s'effectue directement par un débranchement depuis l'appelant vers l'appelé. Ceci n'est plus possible dans un environnement distribué. Afin de rendre transparent le dialogue client-serveur, la technique d'appel de procédure à distance (RPC) a été introduite.



# Appel de procédure à distance

- **Appel de procédure à distance (Remote Procedure Call RPC)**
- Technique permettant d'appeler une procédure distante comme une procédure locale, en rendant transparents les messages échangés et les assemblages/ désassemblages de paramètres.
- Le RPC offre les fonctions de l'appelé à l'appelant sur le site de ce dernier. Il est réalisé par introduction d'une souche de procédure (en anglais, stub) pour transformer l'appel de procédure en un envoi de message depuis le site de l'appelant au site de l'appelé. Là, une souche de procédure symétrique reçoit le message et réalise l'appel effectif de l'appelé par débranchement. Les paramètres de retour transitent par un chemin inverse via le réseau d'interconnexion .
- **Souche (Stub):** Représentant d'une procédure sur un site client ou serveur capable de recevoir un appel de procédure du client et de le transmettre en format adapté à l'implémentation ou à son représentant.

# Appel de procédure à distance



# Appel de procédure à distance

- Les souches peuvent être générées automatiquement à partir d'un langage de description d'interface, permettant de spécifier les noms des procédures appelées, et les types des paramètres d'entrée et de sortie. Ce langage peut être défini comme une extension d'un langage existant (par exemple C), ou comme un langage indépendant de tout langage de programmation (par exemple *IDL Interface Definition Language*).

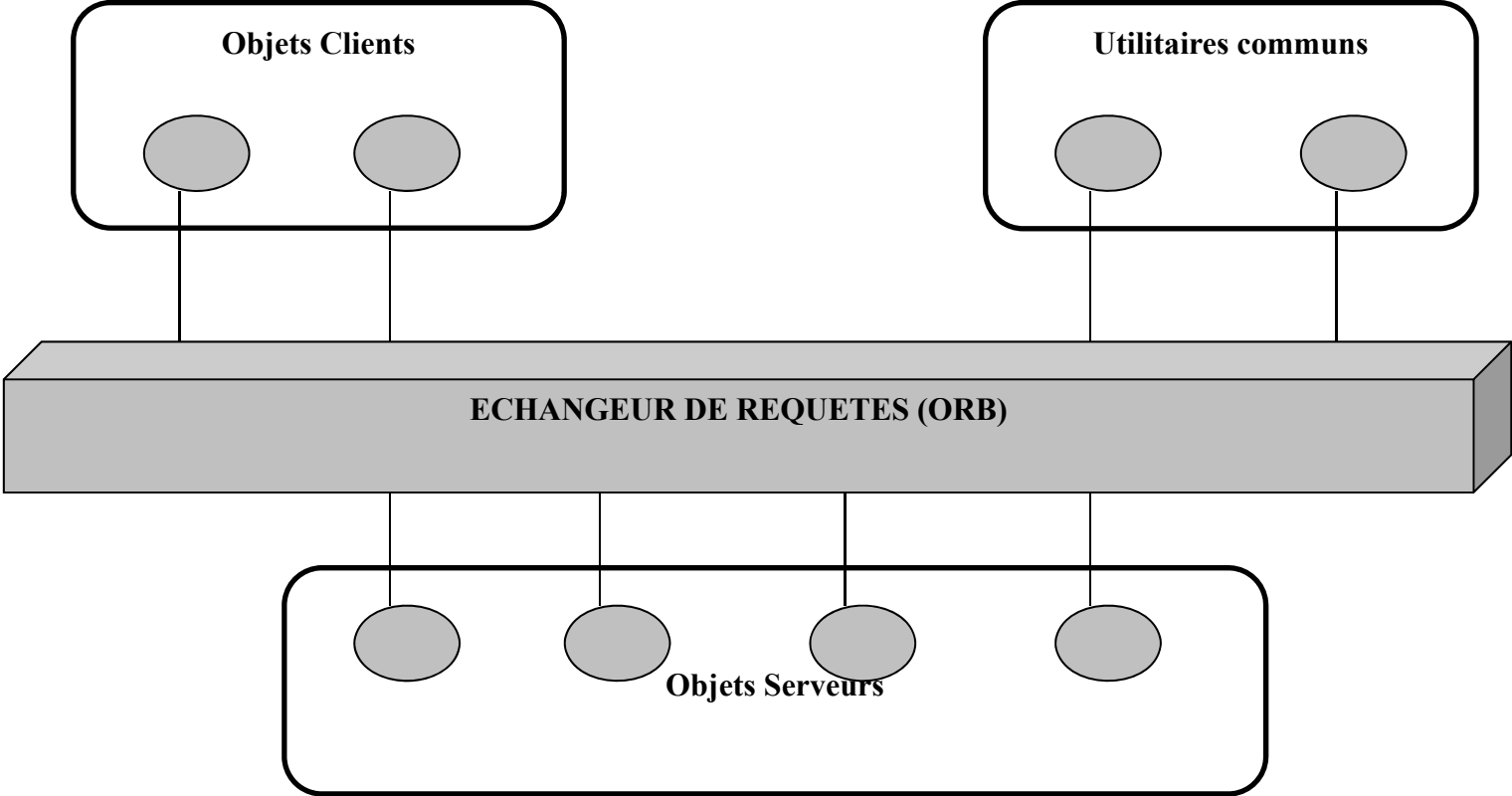
# Dialogue synchrone et asynchrone

- **Dialogue synchrone (Synchronous dialog)** Type de dialogue géré sans file d'attente, où les commandes d'émission et de réception sont bloquantes.
- Typiquement, dans le cas synchrone, le client attend le serveur pendant que celui-ci exécute une opération pour lui.
- **Dialogue asynchrone (Asynchronous dialog)**  
Type de dialogue géré avec file d'attente, où l'une au moins des commandes d'émission ou de réception est non bloquante.

# Les Middlewares

- Ensemble des services logiciels construits au-dessus d'un protocole de transport afin de permettre l'échange de requêtes et des réponses associées entre client et serveur de manière transparente.
- L'objectif d'un médiateur (Middleware) est donc d'assurer une liaison transparente, c'est-à-dire de cacher l'hétérogénéité des composants mis en jeu. Il s'agit en particulier d'assurer la transparence aux réseaux, aux SGBD, et dans une certaine mesure aux langages d'accès

# CORBA



# CORBA

- Architecture distribuée basée sur **un échangeur de requêtes entre objets** (*Object Request Broker ORB*) et des services standardisés. Parmi les services sont distingués les **services objets** et les **utilitaires communs**. Les premiers fournissent les fonctions de base pour utiliser et implémenter des objets utilisateurs au niveau des clients et des serveurs. Ce sont par exemple la gestion d'événements, de noms, de transactions. Les seconds sont des assemblages de services directement utilisés par l'utilisateur final, tel le courrier électronique ou la gestion de documents composites.