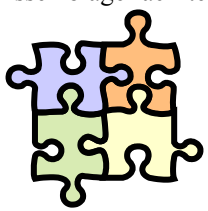
CORBA

(Common Request Broker Architecture)

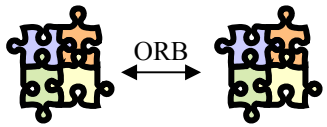
CORBA, introduction (1/4)

❑ Les systèmes répartis permettent de créer des applications basées sur des composants auto-gérables, interagissant et se promenant librement sur un réseau

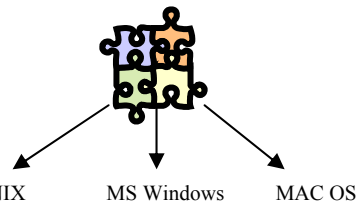
Assemblage facilité



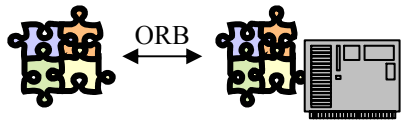
Interopérabilité



Portabilité



Coexistence



❑ La notion de composant est essentielle : un composant est un objet autonome prêt à être branché sur des réseaux, des applications, des outils. Ainsi, un composant minimal se caractérise par les notions suivantes

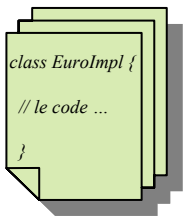
Brique logicielle réutilisable, prête à l'emploi et qui ne dépend d'aucune application

- élément logiciel binaire et prêt à l'emploi
- destiné à exécuter un ensemble limité de tâches dans un domaine particulier
- interopérabilité (composants qui interagissent sans à priori se connaître)
- interface pour permettre à l'extérieur d'accéder aux services du composant

❑ Les objets métier sont des super composants rendant des services particulier pour un métier donné

CORBA, introduction (2/4)

❑ Le développement d'applications à base de composants apporte de multiples avantages. D'un point de vue conceptuel, les principaux sont les suivants



• le développeur dispose de composants d'origines diverses qui peuvent interagir facilement

• l'utilisateur peut créer ses propres applications avec des scripts et des composants

• le commercial peut vendre des suites taillées sur mesures

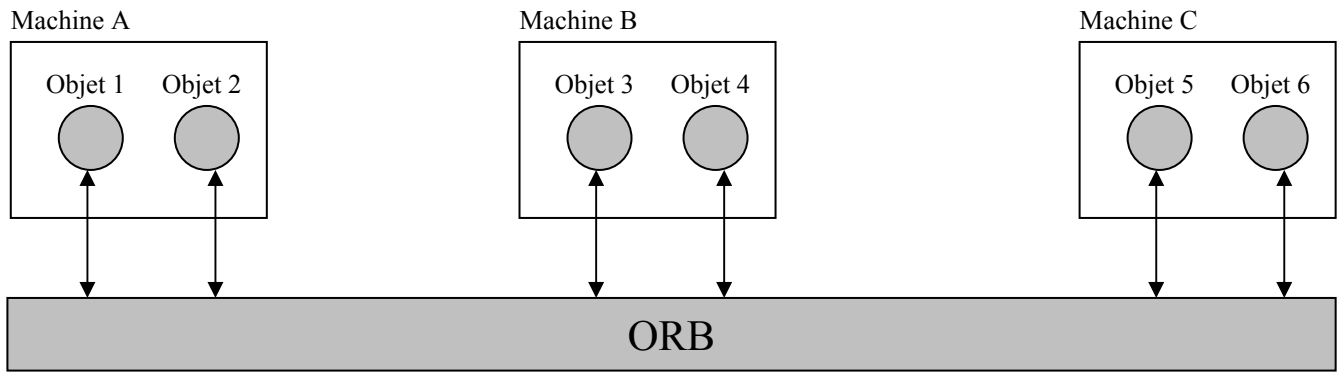
CORBA, introduction (3/4)

❑ CORBA est un middleware orienté objet spécifié par l'OMG (Object Management Group)

• CORBA est une spécification d'une architecture distribuée d'objets hétérogènes qui coopèrent via un bus logiciel

• CORBA permet à un ensemble d'objets distribués d'interopérer

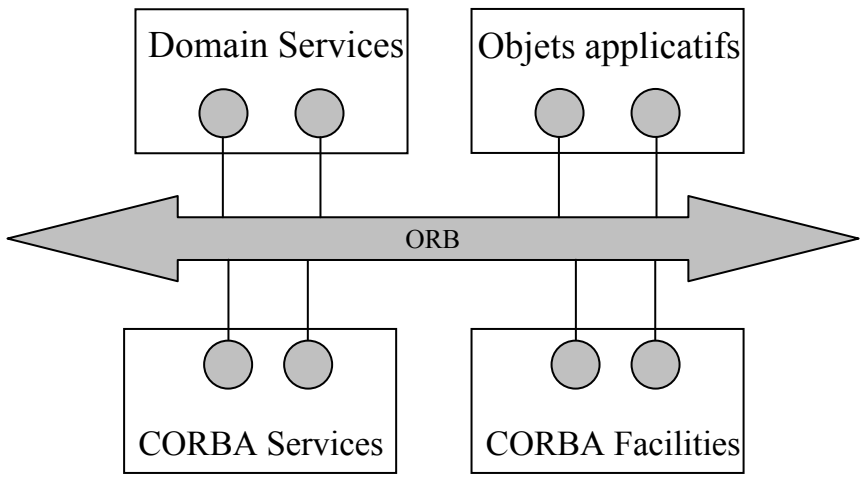
❑ CORBA fournit une vision unique d'une architecture distribuée dans des environnements hétérogènes



CORBA, introduction (4/4)

- Les points clés
  - masquer l'hétérogénéité des langages de programmation, des systèmes d'exploitation, des machines, …
  - assurer l'interopérabilité entre composants, entre fournisseurs multiples
  - langage commun, l'IDL
  - bus logiciel, l'ORB
  - modèle Client – Serveur orienté objet

- L'architecture ouverte proposée par l'OMG, l'OMA (Object Management Architecture), peut se résumer ainsi

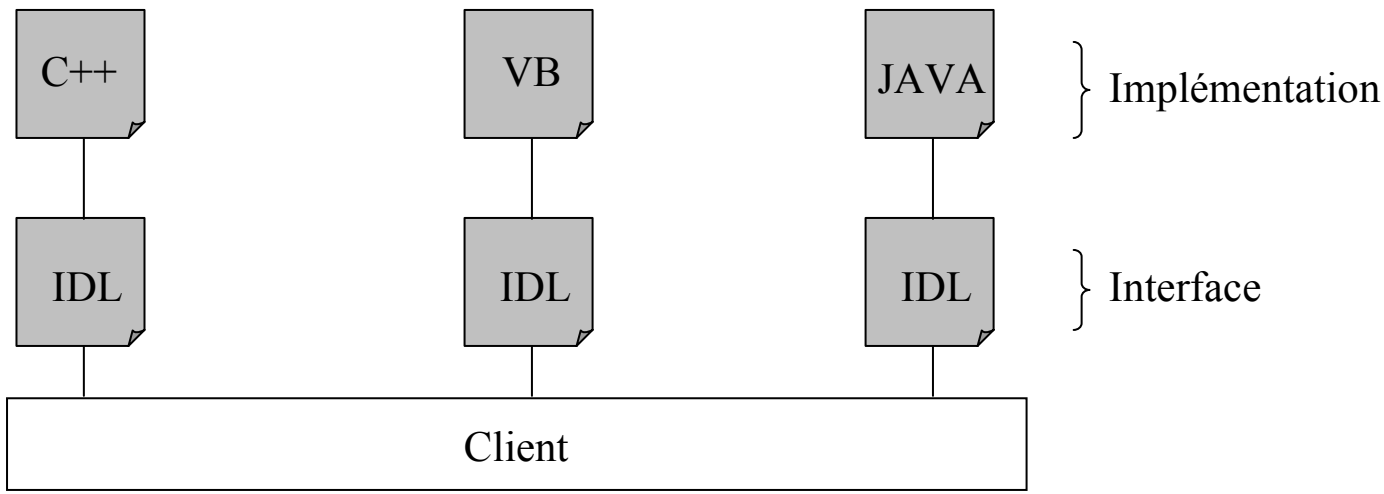


L'IDL

❑ La base de l'interopérabilité de CORBA est que les spécifications sont basées sur des interfaces écrites dans un langage neutre, l'IDL (Interface Definition Language)

```
module Convertisseur {
   interface Euro {
      attribute double  taux;
      double toEuro(in double devise);
   };
};
```

❑ L'IDL est un langage purement déclaratif qui permet de décrire le contrat que le composant offre au monde extérieur

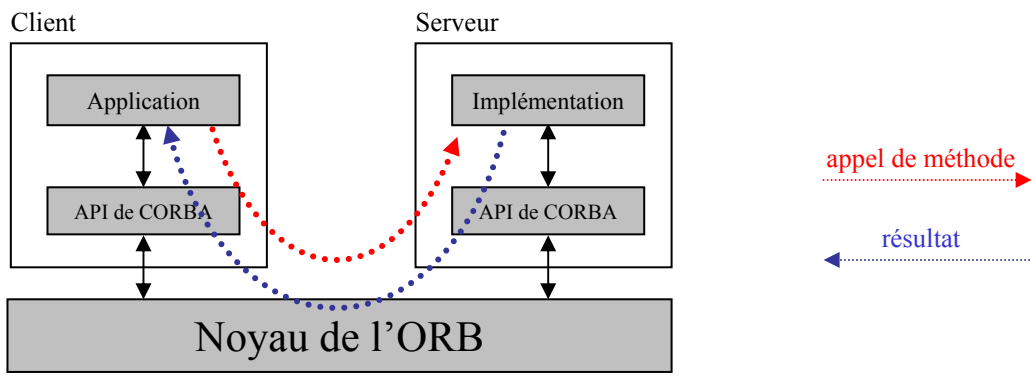


L'ORB (1/6)

❑ L'ORB (Object Request Broker) est le négociateur de requêtes à objets autrement dit le bus à objets, qui fournit un ensemble très riche de services de middleware réparti

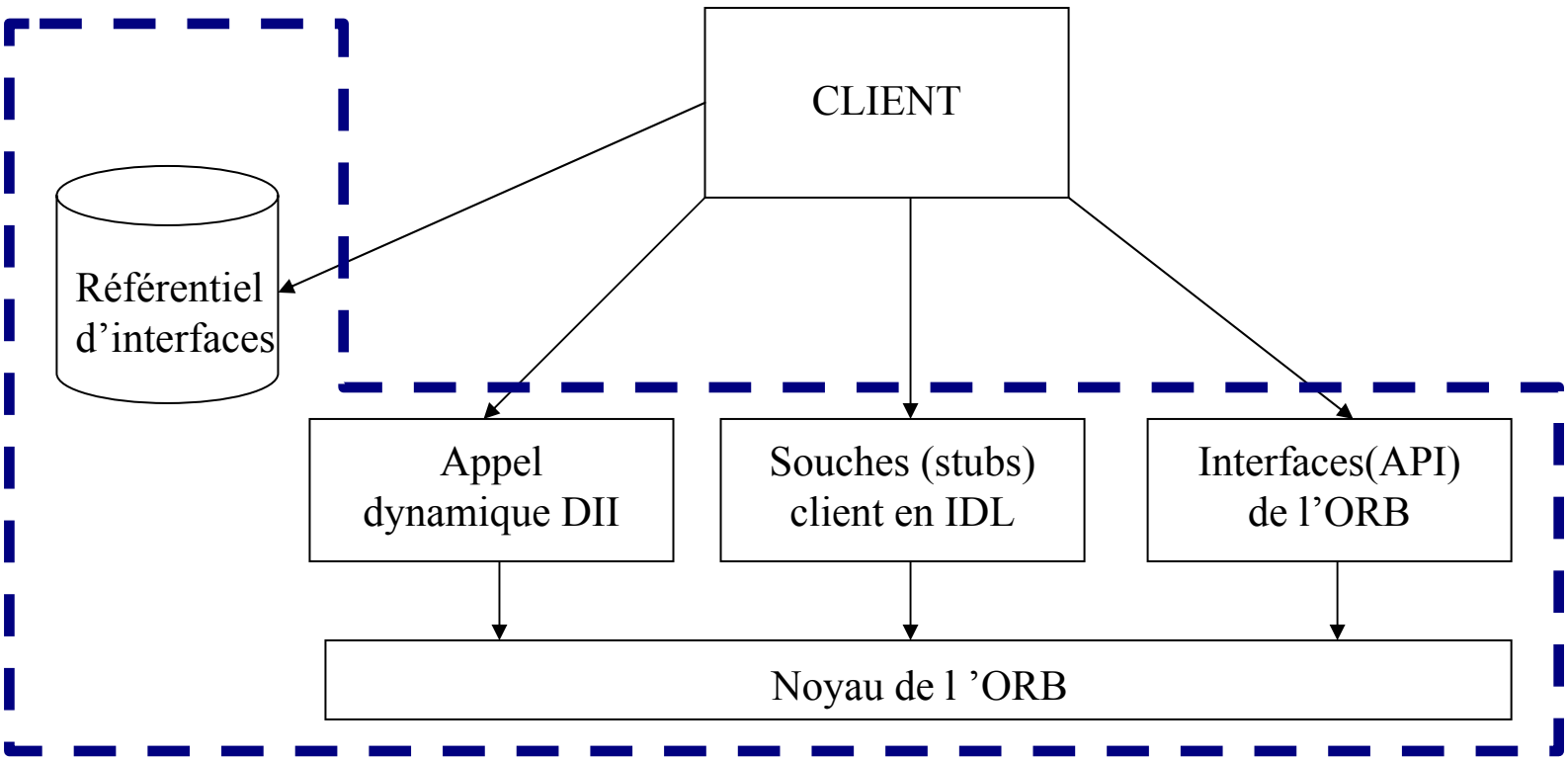
- il établit les relations entre objet, indépendamment de la localisation de ces derniers
- il permet donc aux composants de se découvrir, les uns les autres et d'interagir

❑ La notion de Client – Serveur est relative à une situation donnée. Pour l 'ORB, un objet peut être ou client ou serveur selon les besoins.



L'ORB (coté client) (2/6)

- La structure d'un ORB « coté client » en CORBA 2.0 peut être schématisée ainsi



L'ORB (coté client) (3/6)

❑ Les souches (stubs) fournissent les interfaces statiques des services objet du serveur. Pour chaque objet distant, le client doit posséder la souche correspondante à l'objet. Grâce à cette souche, l'encodage (et le décodage) de l'appel d'une méthode (avec ses paramètres) pourra être réalisé (marshaming) et envoyé au serveur

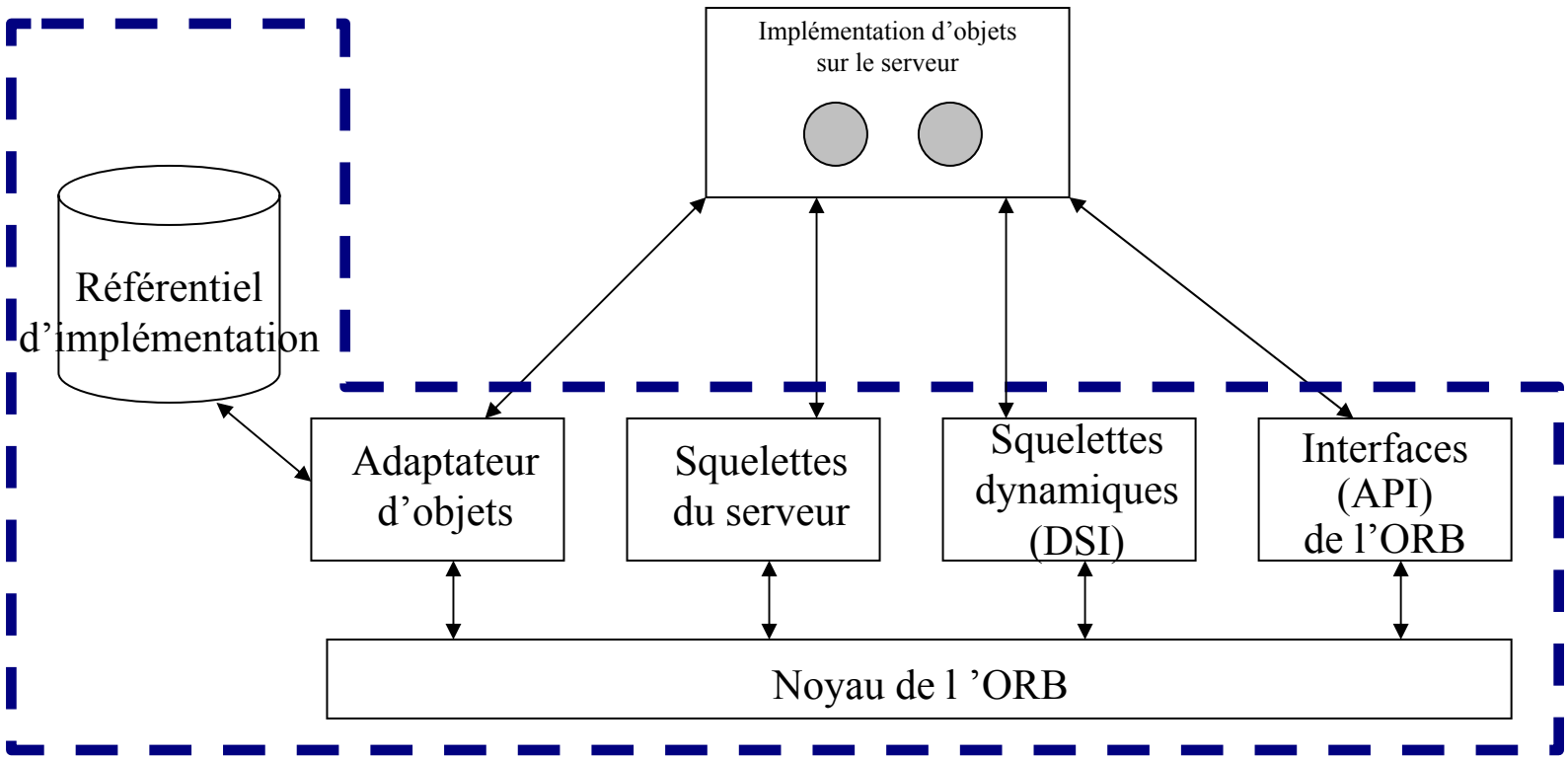
❑ Les appels dynamiques se font grâce aux interfaces à appel dynamique (DDI), qui permettent donc de découvrir la méthode à exécuter au moment de l'exécution

❑ Les Référentiel d'interfaces est une « base de données » qui contient toutes les versions exécutables (versions binaires) des interfaces définies en IDL. Des API de CORBA permettent d'accéder à ces métadonnées (données décrivant un composant)

❑ Les interfaces de l'ORB sont des API permettant d'accéder à certains services de l'ORB (transformation d'une référence d'objet en son nom en clair par exemple)

L'ORB (coté serveur) (4/6)

- La structure d'un ORB « coté serveur » en CORBA 2.0 peut être schématisée ainsi



L'ORB (coté serveur) (5/6)

- Les squelettes (skeletons) du serveur ont le même rôle que les stubs : elles décrivent en IDL les services fournies par le serveur

- Les appels dynamiques se font via les interfaces de squelettes dynamiques (DSI) : elles permettent l'appel de méthodes pour des composants serveur qui ne possèdent pas de skeletons

- Le référentiel d'implémentation contient l'ensemble des classes supportées par le serveur, les objets en mémoire et leurs OIDs (Object IDentifier)

- Les interfaces de l'ORB ont le même rôle que chez le client

- L'adaptateur d'objet (BOA, POA, …) se situe au dessus du noyau de l'ORB : c'est lui qui va traiter les demandes d'appels de méthodes. Pour ce faire, il fournit un environnement d'exécution pour créer les instances des objets sur le serveur. L'adaptateur va également enregistrer les objets dans le Référentiel d'implémentations

L'ORB (coté serveur) (6/6)

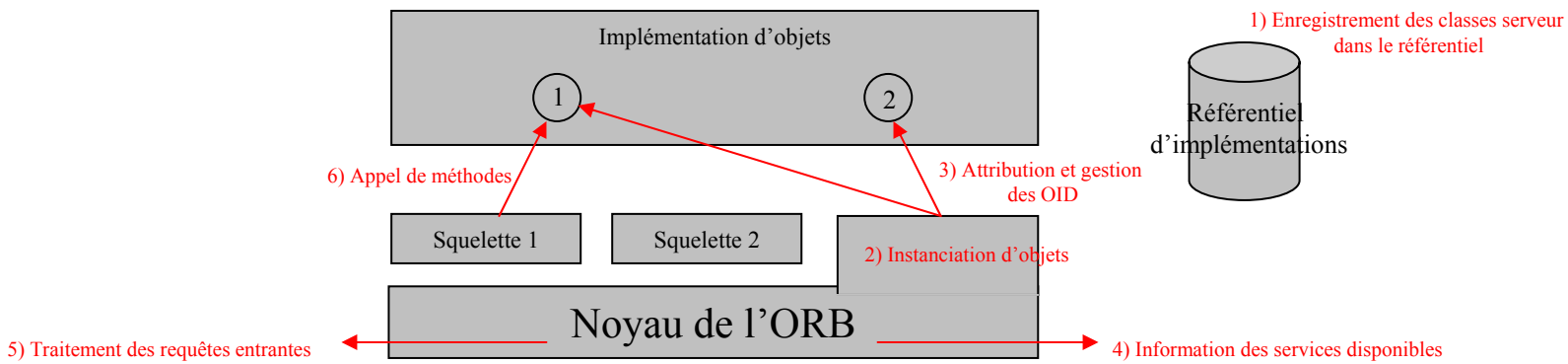
❑ L'adaptateur d'objet (OA, Object Adaptater) fournit un environnement complet pour que les instances d'objets puissent vivre sur le serveur. Son principal rôle est d'associer une requête portant sur un objet référencé au code spécifique qui doit servir la requête

❑ Toute machine voulant être un serveur d'objets doit posséder au moins un adaptateur d'objets

❑ Plusieurs types d'OA ont été spécifiés

- BOA (Basic OA) avant CORBA 2.2
- POA (Portable OA) après CORBA 2.2
- etc, …

❑ Un OA réalise principalement les actions suivantes

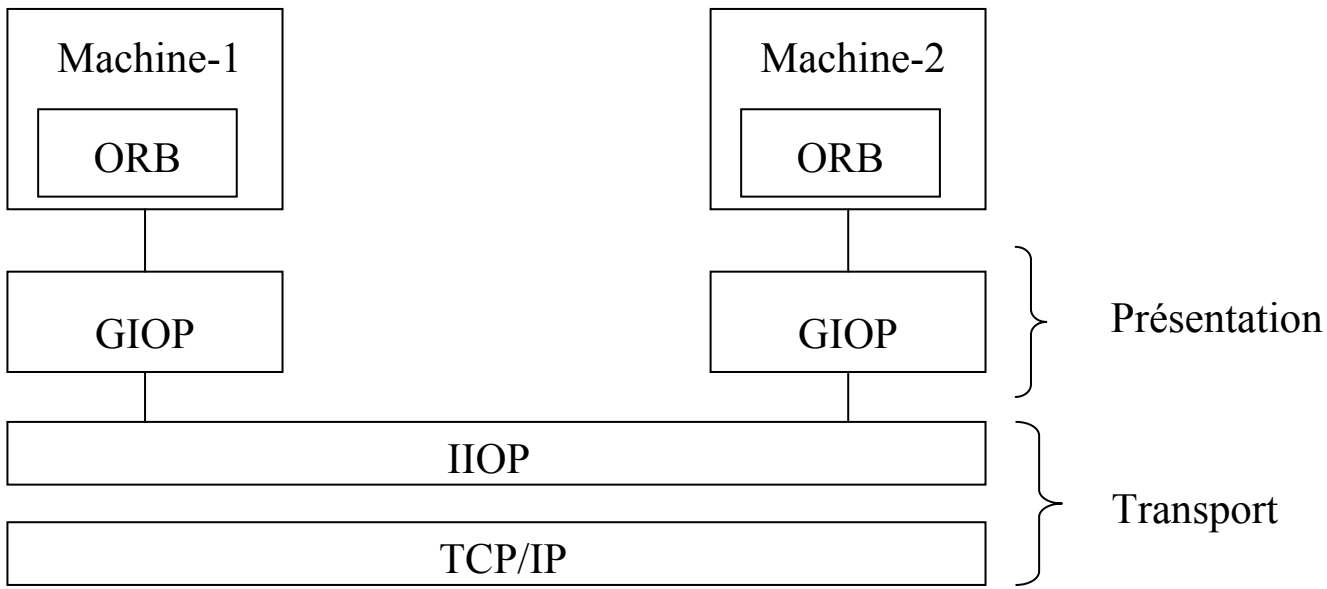


Architecture inter-ORB

❑ L'OMG a définit des standards pour faire communiquer 2 ORB. Les 2 protocoles principaux sous CORBA sont les suivants

• GIOP (General Inter-ORB Protocol) est situé au dessus de tout protocole de transport : ce protocole spécifie comment deux ORB peuvent communiquer

• IIOP (Internet Inter-ORB Protocol) est une spécialisation de GIOP au dessus de TCP/IP et permet d'acheminer des messages au « format GIOP » entre 2 machines



CORBA Services

- CORBA fournit un ensemble de services au niveau système. Ces services sont proposés sous forme de composants

- Ces services viennent enrichir les services de base de l'ORB

- Entre autres, les services suivants sont disponibles

  - cycle de vie : gestion de la création, copie, suppression, … des composants sur le bus
  - persistance : stockage des composants, …
  - nommage : localisation des composants par leurs noms en clair
  - etc, …

Common Facilities

❑ Les utilitaires communs (Common Facilities) sont des composants qui fournissent des services directement utilisables par les objets métier.

❑ Ces services sont encore appelés canevas applicatifs, par opposition aux services de base, appelés canevas système

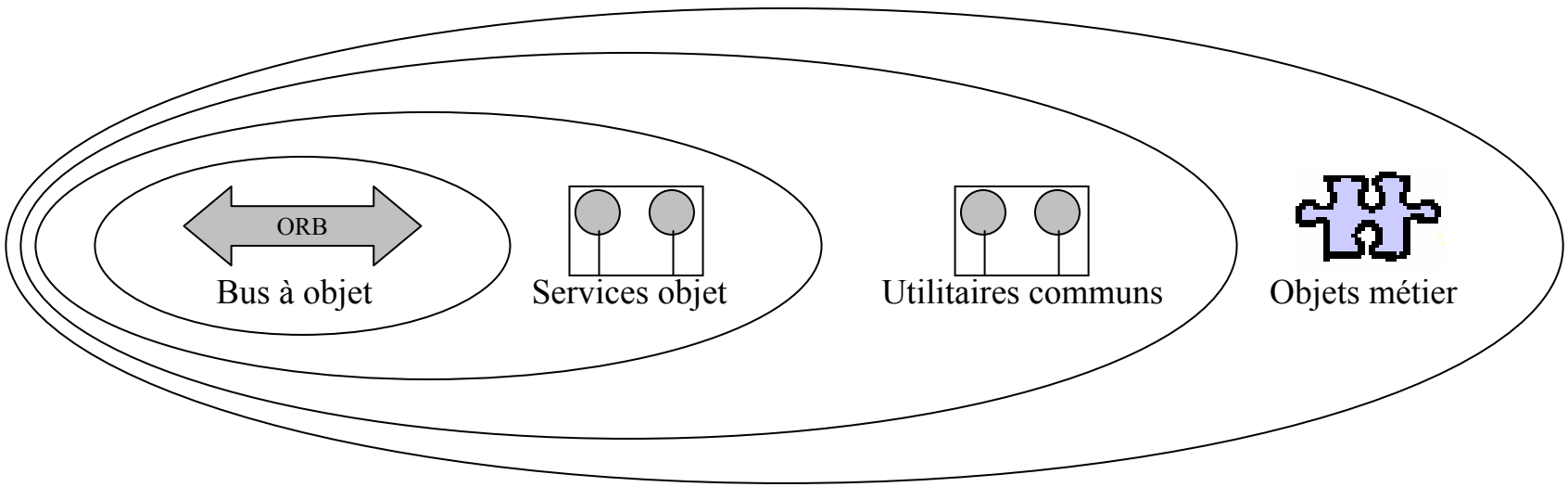
❑ On peut citer les services suivants

- gestion de l'information avec par exemple la gestion des documents composites
- services d'administration avec notamment les gestion des composants répartis
- etc, …

Business Object

❑ Les objets métier (Business Objet) sont les briques applicatives que l'on va utiliser pour assembler des applications

❑ Les objets métier sont la partie « utile » d'un point de vue de l'utilisateur final : ces objets métier se servent des services présentés ci-avant



Références

- Les cours du DESS MIAGe de l'Université Paul Sabatier de Toulouse (Mr DESPRATS T. et Mr STEFF Y.)

- DANIEL J., *Au Coueur de CORBA,* Vuibert

- BONJOUR M., FALQUET G., GUYOT J., LE GRAND A., *Java : de l'esprit à la méthode,* Vuibert

- NICOLAS C., AVARE C., NALMAN F., *Java 1.1,* Eyrolles