

Objets distribués

Architectures, état de l'art, pratiques
Partie A-introduction

Plan

- Cette séance:
 - A-vue d'ensemble sur les SàOD (45mn)
 - B-Java RMI (35mn)

- Prochaine séance:
 - C-Corba
 - Architecture et modèle objet de CORBA
 - Les composants fournis par l'ORB
 - Le langage IDL (Interface Definition Language)
 - Mise en œuvre en C++ (java sera vu en TP)
 - Le service de nom (Naming Service)

A-Vue d'ensemble

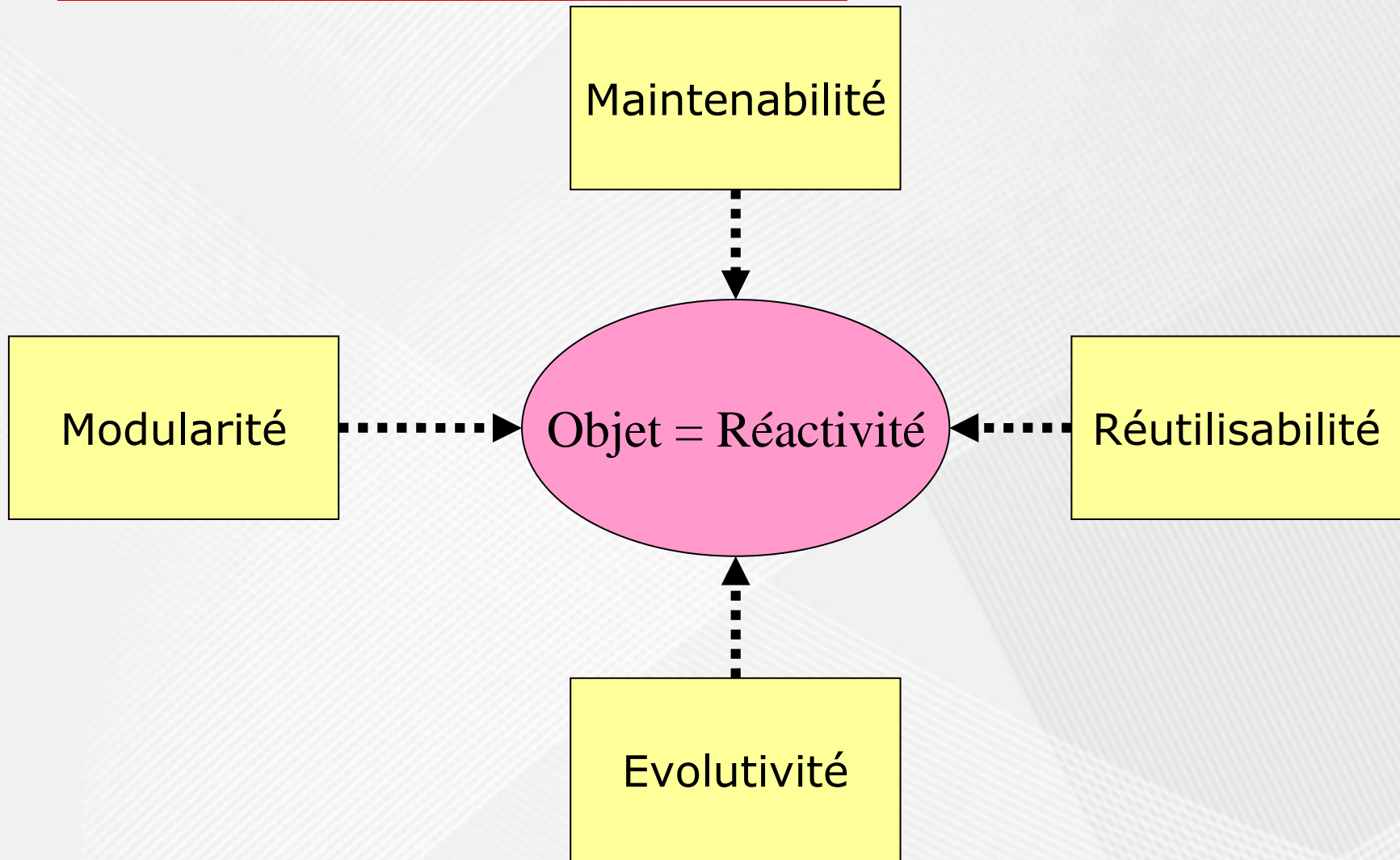
De quoi s'agit il ?

- Idées de base :
 - Faire évoluer le client/serveur
 - Mieux l'intégrer aux L3G
 - Utiliser les techniques « objet »

- Il est prouvé que les objets favorisent :
 - La Réutilisabilité
 - La Réduction du Time-to-market
 - Meilleur "retour sur investissement" (profit++ pour une entreprise) (Exemple: un prg de contrôle de Trafic aérien, temps-réel, composé de +1.000.000 LDCs en c/s, distribué sur une centaine de machine - Cout prévu : 500 \$ par LDC - Cout réel : 700-900 \$ par LDC)

- Principe des objets répartis:
 - Evolution de la programmation objet
 - Les objets étaient « prisonniers » d'une même machine
 - Permettre à des objets de dialoguer à travers un réseau hétérogène

Pourquoi l'objet métier?

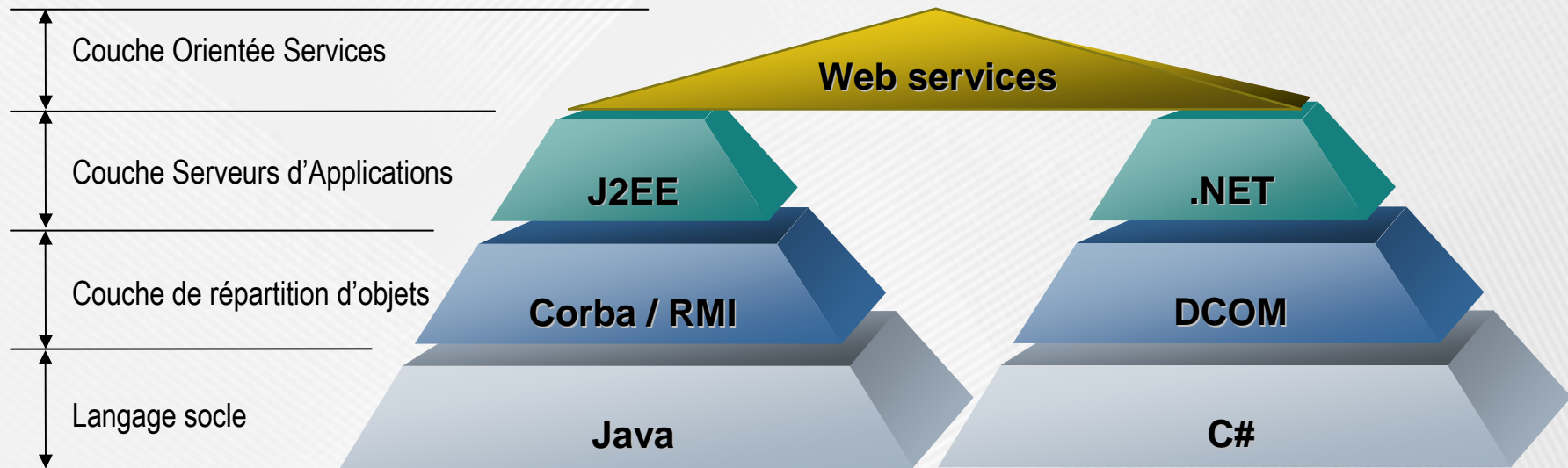


De quoi s'agit il ?

Réponse à des besoins d'entreprise

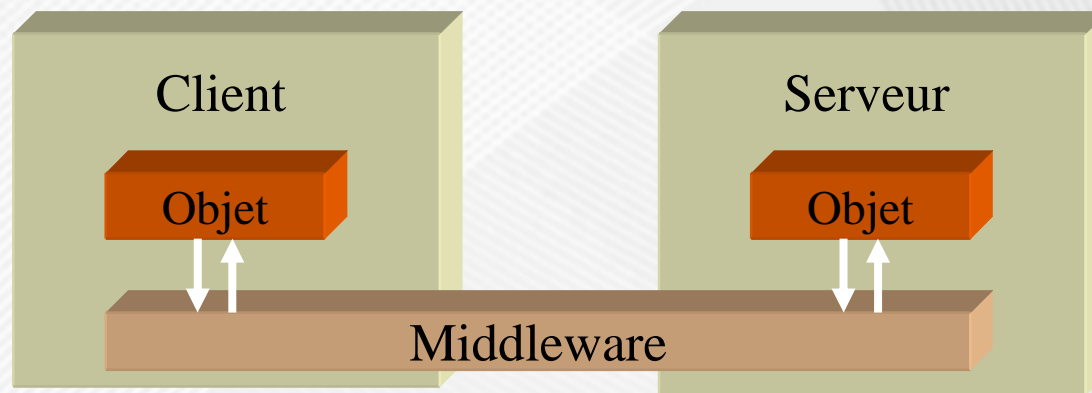
- Intégration des applications et des BDs existantes
- Répartition des données sur des sites géographiques distants
- Distribution des traitements effectués sur les données réparties
- Echange de données entre applications existantes et hétérogènes
 - applications manipulant des données le plus souvent au format propriétaire
- Gestion des accès concurrentiels
- Ouverture
- Sécurité

Vision d'ensemble



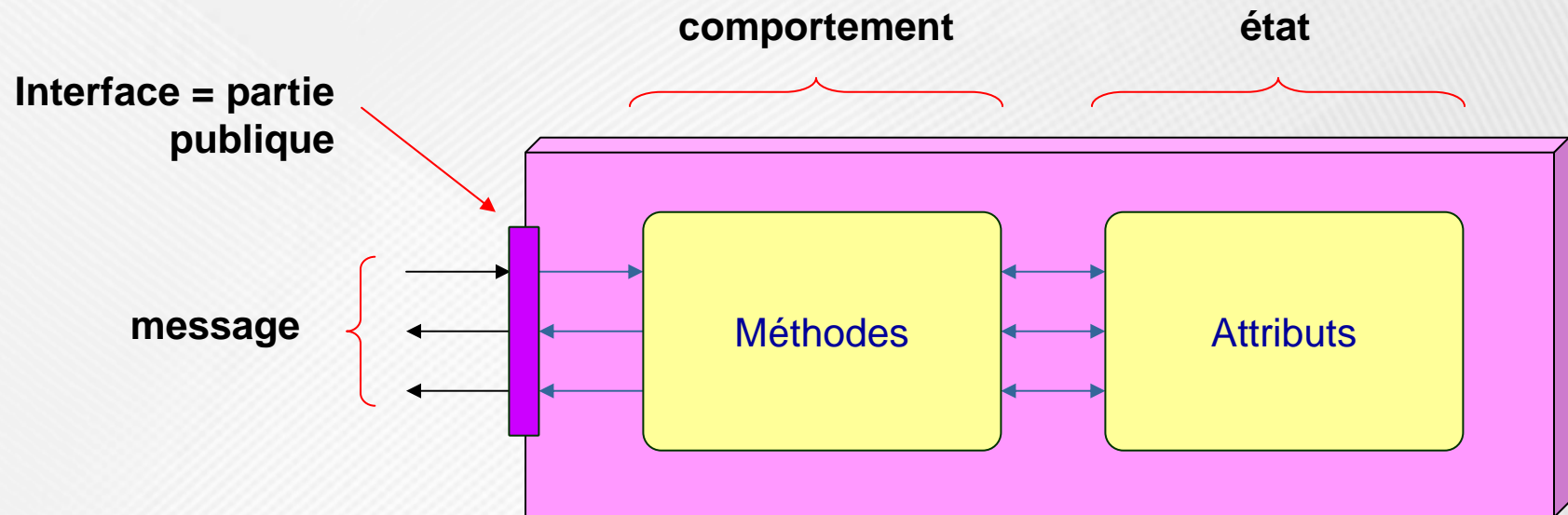
De quoi s'agit il ?

- Intérêts :
 - Faire coopérer de manière logicielle des machines hétérogènes
 - Gérer le déploiement et la maintenance des applications
 - Utiliser et répartir la puissance de calcul et de stockages sur plusieurs serveurs
- Principe simplifié:



Principe d'encapsulation

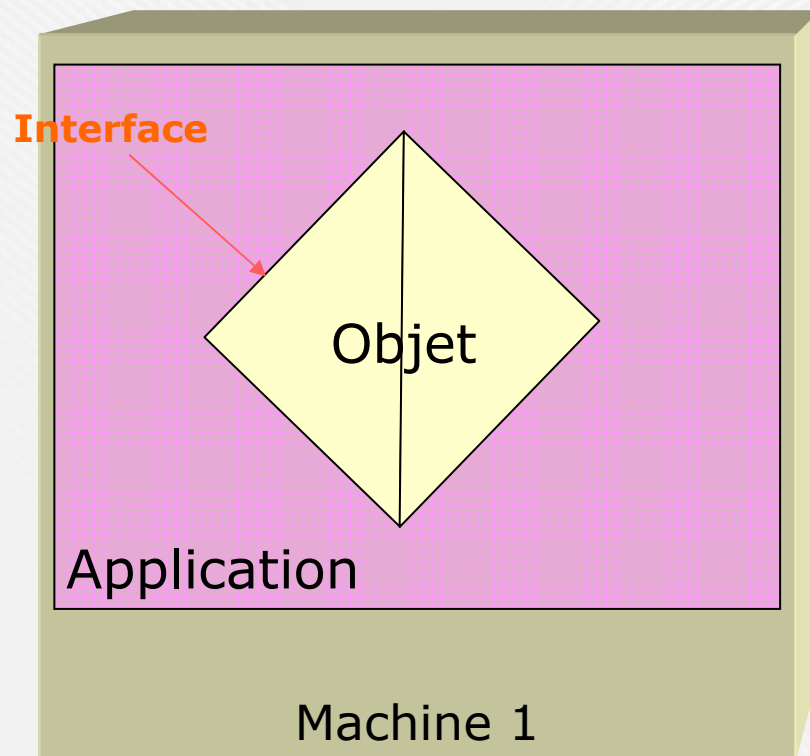
- Comment doit on voir un objet?



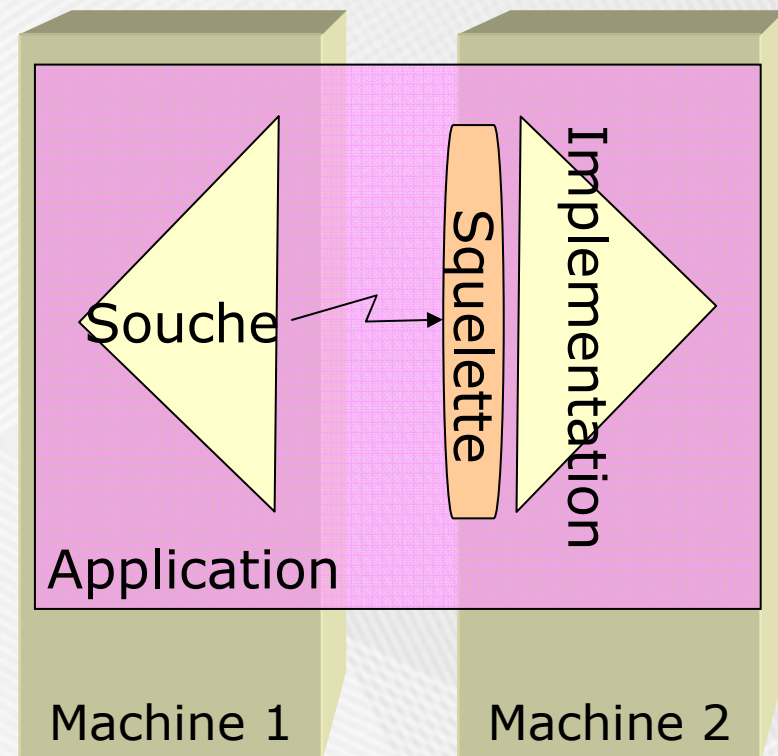
De quoi s'agit il ?

- Concept: comment sont vu les objets..

Mode classique



Mode Distribu 



Avantages et inconvénients

□ Avantages :

- Modélisation plus naturelle car plus proche du monde réel (en théorie grâce au modèle objet !)
- modèle qui assure la réutilisabilité des composants
- indépendance de localisation des objets clients et serveurs

□ Inconvénients :

- absence d'un état global observable (pas de mémoire commune)
- performance
- Déboguage complexe

Objectifs des saod

- Appeler une méthode d'un objet se trouvant sur une autre machine exactement de la même manière que s'il se trouvait au sein de la même machine
- Pouvoir retrouver un objet existant d'après son « nom »
- Passer des paramètres et récupérer des résultats
 - Pouvoir le faire également pour des objets ?
- Gestion du cycle de vie d'un objet
 - Déclencher la création d'un objet sur une autre machine
 - Détruire une instance également

Offre actuelle

3 offres concurrentes :

CORBA

- De l'Object Management Group (OMG)
- Libre et ouvert

RMI

- De Sun
- Lié à Java

COM+ (Dcom)

- De M\$
- Lié à Windows

Corba en bref

- ❑ Spécification d'une architecture proposée par l'OMG
 - CORBA=Common Object Request Broker Architecture
- ❑ Offre une solution au problème d'interopérabilité d'applications hétérogènes.
- ❑ Solution basée sur :
 - la distribution d'objets
 - le principe d'invocation de méthodes distantes
 - le langage IDL
 - le protocole IIOP
- ❑ Egalement :
 - l'intégration de services implantés par des fournisseurs de logiciels

Corba en bref

- ❑ Définit un nouveau langage de spécification d'interfaces d'objets distribués : Interface Definition Language (IDL), indépendant de tout langage de programmation.
- ❑ Projection du langage IDL vers des langages de programmation.
Exemples :
 - ❑ IDL -> C
 - ❑ IDL -> C++
 - ❑ IDL -> Java etc...
- ❑ Transmission des messages entre objets distribués assurée par le protocole IIOP (Internet Inter ORB Protocol).
 - IIOP est une implémentation de la spécification GIOP (General Inter ORB Protocol) proposée par l'OMG.
 - IIOP est au dessus de TCP/IP sur OSI

Où est utilisé Corba?

- ❑ visibroker: 450 clients actifs en 2005 (grands comptes)

- ❑ Telecom
 - Vodaphone IT (OmniORB)
 - Bouygues Telecom (Infrastructure ISIS)
 - NASA's Goddard Space Flight Center (Orbacus)
 - Telescope spatial Hubble (Orbacus est utilisé dans le système de navigation)

- ❑ orbit pour Gnome (IHM de Linux)
- ❑ Le système de sécurité sociale allemand (Visibroker)
- ❑ sun's neo Object request broker (Sun Solaris)

- ❑ Voir pour d'autres exemples
 - <http://www.orbacus.com/aboutus/cs/>
 - <http://www.omg.org/middleware/webcast/presentations/>

DCOM en bref

- Architecture de Microsoft offrant une solution au problème de la distribution et de la communication d'objets sur plates-formes clientes hétérogènes (= différentes versions de Windows !)
- Objets issus de langages différents (Visual C, C++ et autres) pour l'OS Windows.
- On en parle parce que cette technologie touche le plus grand nombre de machines dans le monde (PC)
- Architecture basée sur :
 - OLE (Object Linking and Embedding)
 - COM (Component Object Model)
 - DCOM (Distributed Component Object Model)
 - (les composants ActiveX)

DCOM en bref

- ❑ **OLE** : technologie de Microsoft permettant de faire dialoguer ses produits Bureautique. Ensemble de bibliothèques et d'applications pour la composition, le stockage, l'échange et l'intégration de documents Microsoft.
- ❑ **DCOM** : technologie qui définit les extensions du modèle COM pour faire un système d'objets distribués basé sur les RPC. Il permet à des composants Microsoft de communiquer à travers le réseau.
- ❑ Les composants ActiveX peuvent être chargés pour être exécutés localement sur le poste client de type Windows.
- ❑ Possibilité d'invoquer et d'exécuter des méthodes distantes.

DCOM en bref

- ❑ DCOM spécifie les interfaces entre les composants au niveau binaire (et non au niveau source)
- ❑ Impossible à mettre en œuvre sans un outil MS (Visual C++) avec la librairie ATL (« ActiveX Template Library »)
- ❑ Génération automatique
 - Des « talons » (proxys et stubs)
 - Des « type libraries » pour l'appel dynamique et l'introspection avec le service automation
- ❑ Les interfaces :
 - sont identifiées dans le système par des nombres uniques à 128 bits
 - GUID ("Globally Unique Identifier"), ou UUID ("Universally Unique Identifier") unique dans le monde.
 - Le(s) GUID des composants COM sont appelés les CLSID ("CLaSS Identifier") et sont stockés dans la base de registre sous HKEY_CLASSES_ROOT

Java/RMI

- Java RMI permet à du code Java d'appeler une méthode Java sur un objet serveur distant. Java RMI est implanté en utilisant des mécanismes de sockets et de sérialisation propriétaires.
- On peut dire que Java/RMI est une simplification opérationnelle de Corba.
- Avantages :
 - Simple d'utilisation : distribution des objets sécurisée, ramasse-miette distribué
 - synchronisation des threads, traitements distribués des exceptions
- Inconvénients :
 - Spécifique à Java
 - Obligation d'une machine virtuelle sur le client et sur le serveur.