

## Introduction aux systèmes distribués L'essentiel de CORBA et SOA

Nathanaël Cottin



contact@ncottin.net  
<http://www.ncottin.net>

version 0.0.4 – 2010

## Évolution des systèmes d'information

- Modèle monolithique
- Modèle deux tiers
- Modèle trois tiers
- Modèle multi tiers
- Composants distribués

## Quelques problèmes

- Hétérogénéité
- Communication
- Disponibilité
- Gestion de charge

## Définitions

### Système distribué

Ensemble d'entités chargées de fournir intégralement ou partiellement des services aux utilisateurs

### Service

Composant local ou distant :

- Constitué d'un ensemble de traitements
- Accessible via le réseau
- Consultable par le biais de son interface

## Définitions

### Hôte

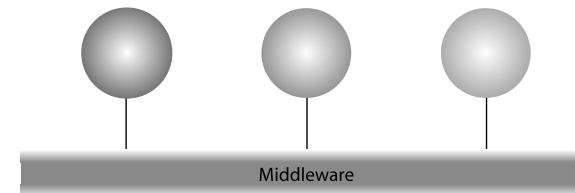
Héberge les services et met à leur disposition :

- Mémoire
- Calcul
- Espace de stockage

## Définitions

### Middleware (intergiciel)

Couche d'abstraction logicielle prenant en charge les appels entre composants distribués



## Objectifs

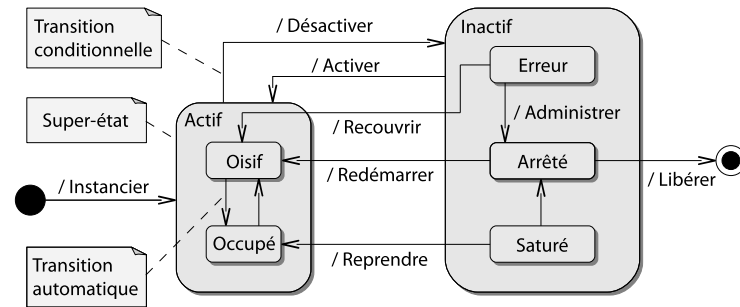
- Efficacité
- Flexibilité
- Consistance
- Robustesse

⇒ Permettent de définir des critères de Qualité de Service

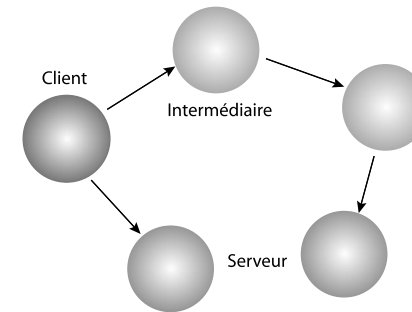
## Principaux avantages

- Mise à l'échelle
- Fonctionnement en mode dégradé
- Support de montée de charge
- Modularité
- Évolutivité
- Transparence des appels
- Indépendance entre interface et traitements

## Cycle de vie d'un composant distribué



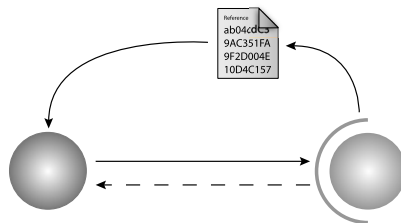
## Classification des composants distribués



Grappe

Système distribué = graphe orienté

## Processus d'invocation général



## Transparences

- À la localisation
- À l'accès
- À l'hétérogénéité
- À l'extension de ressources
- Aux pannes

## Difficultés de mise en œuvre

- Gestion de l'hétérogénéité
- Gestion des composants
- Horloge globale
- Cohérence des données
- Fonctionnement en mode dégradé
- Disponibilité, détection d'arrêt

## Object Management Group

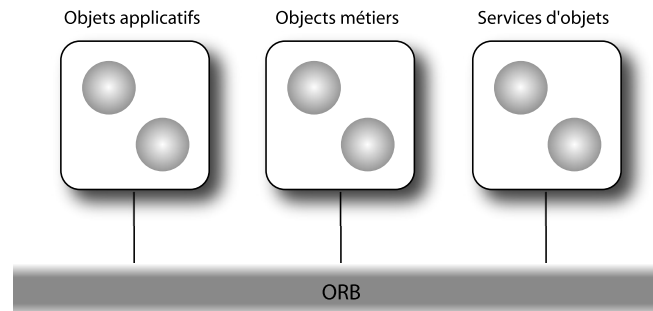
### Rôle d'approbation et promotion de CORBA

- Plus de 900 membres
- 1991 : CORBA 1.0
- 2003 : CORBA 3.0

### Triple vision de l'OMG :

- Indépendance des plate-formes et langages de programmation cibles
- Indépendance interfaces – traitements
- Vue utilisateur locale

## Object Management Architecture



## General Inter-ORB Protocol

### Homogénéisation des échanges

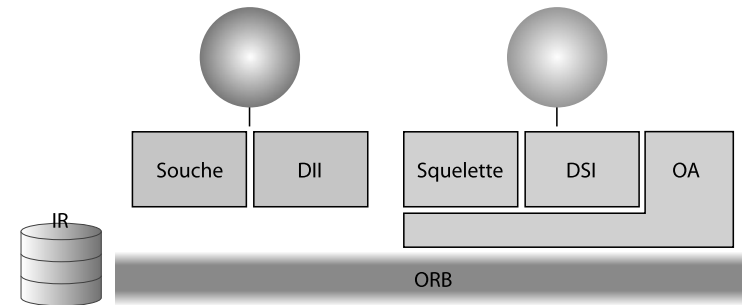
- GIOP définit un format d'échanges commun
- Protocole IIOP

## Principaux éléments

- ORB
- IOR
- IDL
- Services communs

## Object Request Broker

- Bus à objets
- Achemine les requêtes
- Gère la transparence des invocations



## Interoperable Object Reference

Représente la référence réseau d'un objet. Elle contient :

- Adresse de l'hôte
- Port réseau
- Identifiant local

## Interface Description Language

- Description des comportements
- Déclaration des types de données
- Définition des exceptions
- Déclaration des constantes
- Signatures des opérations publiques
- Projection dans divers langages
- Héritage multiple sans surcharge

## Exemple de contrat IDL

```
interface timer {

    enum TimeReq { UTCTime, GeneralizedTime };

    exception AlreadyStarted {};
    exception NotStarted {};

    long getTime(in TimeType tType);
    void start() raises(AlreadyStarted);
    long stop() raises(NotStarted);
}
```

## Projections IDL

Exemples de langages :

- C++
- Java
- Python
- Smalltalk
- C
- Perl
- Ada
- COBOL

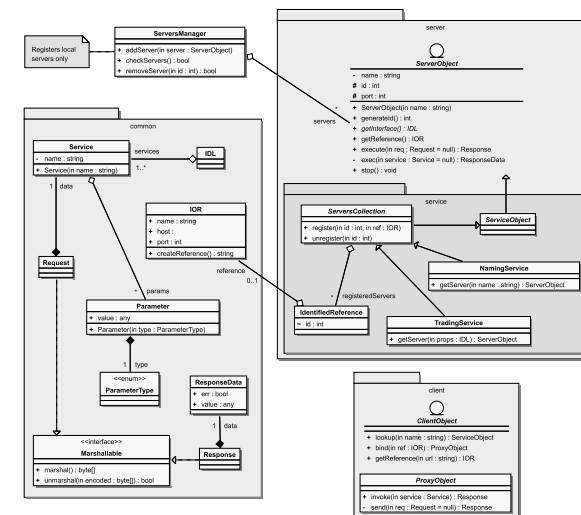
## Services communs

- Nommage
- Vendeur
- Évènement
- Persistance
- Cycle de vie
- Transaction
- Sécurité
- Temps

Remarque

Services = objets fournisseurs

## Modèle



## Outils

### Commerciaux :

- Inprise Visibroker (Java, C++)
- IONA Orbix (Java, C++)
- IONA Orbacus

### Libres ou gratuits :

- IONA Orbacus (pour universités)
- MICO
- JavaORB

## Limites des solutions tierces

- Interactions sur l'ensemble de l'Internet
- Protocoles spécifiques
- Hétérogénéité des protocoles supportés
- Limitations à la découverte de services
- Le demandeur doit disposer des souches pour invoquer le composant
- Construction dynamique des souches difficile à mettre en œuvre
- Pérennité remise en question
- Complexité générale

## Présentation

- Architecture distribuée à objet «à la mode»
- Tire parti du vécu de CORBA
- Repose sur de nombreux standards
- Communications via "XML over HTTP"
- En constante évolution
- Notion de service web

## Présentation

### (R)évolution

- Ne révolutionne pas les concepts des systèmes distribués
- Apporte des facilités d'utilisation

### Cependant...

- Accroît les possibilités des requêtes
- Permet de définir une multitude de langages

## Principaux éléments

- WSDL (Web Services Description Language)
- SOAP (Simple Object Access Protocol)
- UDDI (Universal Description Discovery and Integration)

## Composition de services

### Service composite

Service constitué d'un ensemble de sous-services appelés lors de l'exécution d'une requête

Utilisation du langage BPEL

## Références

- P. Boulet*, Distributed Systems, présentation, 28 septembre 2006
- F. Singhoff*, Introduction aux systèmes à objets répartis, cours, 2006
- J.-M. Vanel*, Web Services, cours, 2004
- N. Cottin*, Introduction à CORBA, présentation, 2001
- N. Cottin*, DOMS : une architecture de suivi de la Qualité de Service dans les Systèmes Répartis Objet, mémoire de master, 2000
- H. Balen*, Distributed Object Architectures with CORBA, Cambridge University Press, 2000