

# Master mention Informatique M1

## Construction d'Applications Réparties

Lionel Seinturier

USTL-LIFL - Équipe GOAL/Projet INRIA Jacquard

Lionel.Seinturier@lifl.fr

2006/2007

1

## Objectifs du cours

Appréhender la conception d'applications réparties

- motivations et concepts
- architectures et exemples
- problèmes et solutions

Comprendre les solutions industrielles

- Internet et sockets Java
- Objets répartis en Java (RMI & CORBA)
- WWW, XML, SOAP, Web services

Maîtriser par la pratique (beaucoup de TP)

- Java, WWW et CORBA

2

## Sommaire

- I. Introduction aux applications réparties
- II. Applications réparties en mode message
- III. Internet et servlet
- IV. Client/serveur et objets
- V. Java RMI
- VI. Le bus logiciel CORBA
- VII. Web Services
- VIII. Applications réparties et composants

3

## Organisation

Chaque semaine : 1 cours, 1 TD, 1 TP

- début cours : semaine 38 (pas de cours semaine 40 - 3/10)
- début TD : semaine 39 (25/9)
- début TP : semaine 40 (2/10)
- interruption pédagogique : semaine 44 (30/10)
- fin cours/TD/TP : semaine 51 (18/12)

4 groupes de TD/TP, 3 enseignants

- J. Dubus, A.-F. Le Meur (x2), L. Seinturier

<http://www.fil.univ-lille1.fr/portail/>

- Procédure de remise de TPs (à venir)

4

# Organisation du contrôle

Un examen

4 sujets de TPs ⇒ Note de contrôle continu

- chaque sujet de TP dure 2 ou 3 séances
- utilise le langage Java
- démonstration de chaque TP en séance
- les TPs sont relevés et notés
  - relève par l'application PROF uniquement (mail refusé)
  - dates de remise fermes
  - binôme ∈ même groupe (pas de changement de binôme)

5

# I. Introduction aux applications réparties

Lionel Seinturier

LIFL - équipe GOAL/Projet INRIA Jacquard

6

# Pourquoi des applications réparties?

La plupart des applications informatiques sont réparties

Besoins propres des applications

- Communication, coopération
- Partage d'information, accès à des ressources distantes
- Répartition intrinsèque des applications

Possibilités techniques

- Coût et performances des machines et des communications
- Interconnexion généralisée
  - Interpénétration Informatique-télécom-télévision
    - Les applications informatiques utilisent les technologies télécom
    - Les réseaux de télécom sont des systèmes informatiques
    - La télévision devient numérique et interactive

7

# Au sommaire

Deux exemples d'applications réparties

- La télévision interactive
- Le commerce électronique

Les problématiques de la construction d'applications réparties

8

# Télévision Interactive

Fonctions : fourniture d'un ensemble de services aux clients

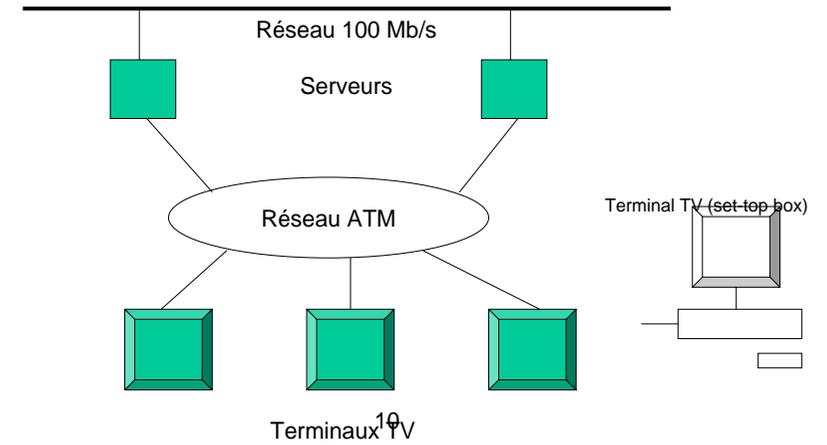
- Diffusion de programme à la demande
- Télé-achat
- Jeux interactifs

Contraintes

- Interface client familière (télécommande)
- Disponibilité
- Performances
- Coût (terminal simple)
- Extensibilité

# Télévision Interactive

Architecture du réseau



# Télévision Interactive

Services et objets

<u>Services de l'applications</u>	
Connexion	
Livraison média	
Livraison fiable	
Gestion média	
Gestion fichiers	
Gestion terminal TV	
Diffusion amorce	
<u>Système de communication</u>	
Noms	Contrôle serveurs
Persistance	Audit ressource
Authentification	

Spécification des services

- Interfaces (IDL)

Réalisation des services

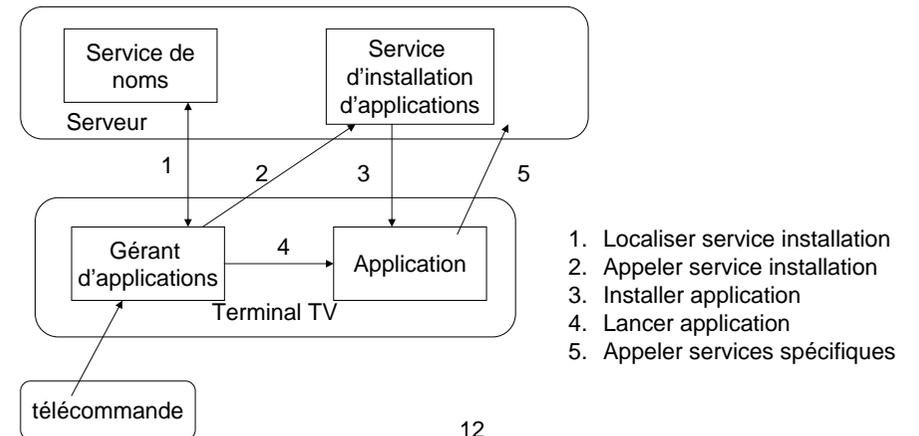
- 1 service = 1 objet
- désignation par référence
- service de noms

Disponibilité

- duplication active
- serveur primaire/secondaire

# Télévision Interactive

Lancement d'une application



1. Localiser service installation
2. Appeler service installation
3. Installer application
4. Lancer application
5. Appeler services spécifiques

# Télévision Interactive

## Commentaires

- Contraintes d'une application grand public
  - Disponibilité -> duplication active/passive, reprise
  - Performances -> bon dimensionnement : répartition de charge
  - Simplicité de l'interface -> terminal TV
  - Passage à l'échelle -> évolution incrémentale
- Génie logiciel d'une grande application répartie
  - Organisation client/serveur
  - Modèle objets
  - Utilisation systématique de l'IDL
  - Service de noms évolué
  - Capacité d'évolution

13

# Commerce électronique

Fonction : exécution de transactions commerciales entre clients et fournisseurs

- Recherche de produits et de services (catalogues)
- Commande simple ou groupée
- Paiement
- Livraison

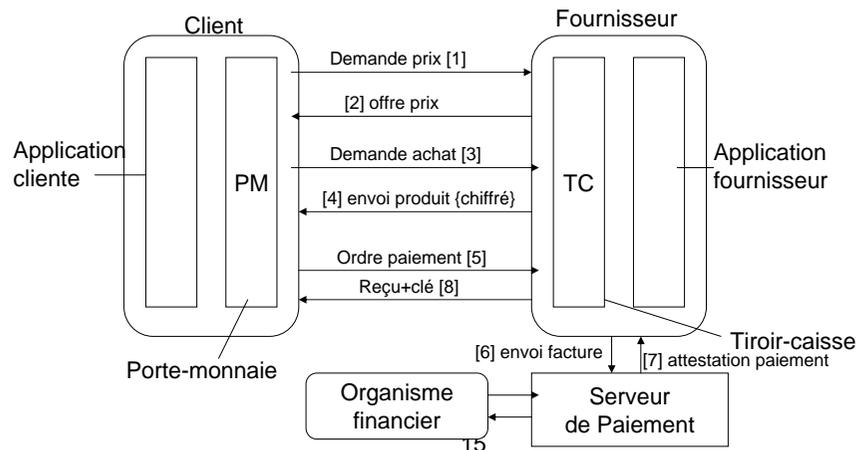
## Contraintes

- Protection des informations confidentielles (client et fournisseur)
- Respect des règles de concurrence
- Respect des garanties fournisseur -> client (offre sincère, exécution du contrat)
- Respect des garanties client -> fournisseur (identité, paiement)
- Respect des droits de propriétés (licences, droit d'auteur)
- Disponibilité permanente du service

14

# Service de commerce électronique

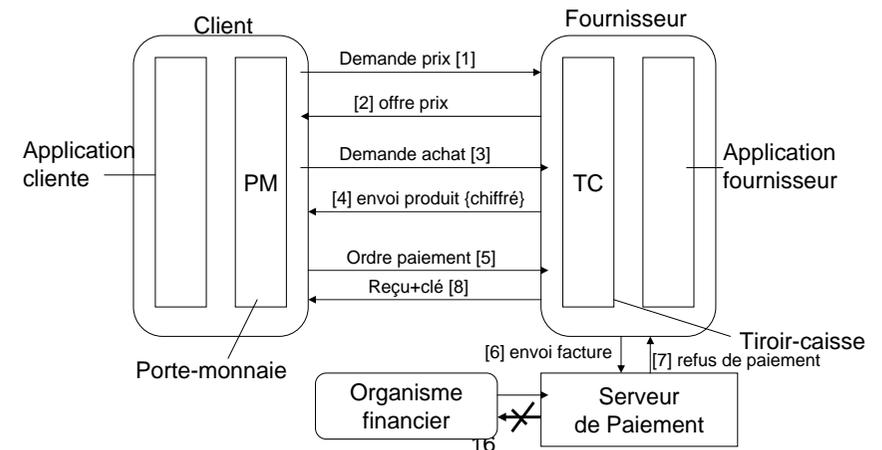
Cas d'un produit disponible sous forme électronique



15

# Service de commerce électronique

Cas d'un produit disponible sous forme électronique



16

## Tolérance aux fautes Point de vue du client

### Pannes dans les étapes de 1 à 4

- Aucune transaction financière n'a eu lieu
- Pas de paiement, pas de livraison

### Paiement après émission de [5] (ordre de paiement)

- Le client a accepté le produit
- Pas de réponse du serveur
- Le client prend l'initiative de la reprise (contacte fournisseur ou serveur pour connaître l'état de la requête)

### Erreurs possibles

- Ordre de paiement non transmis au serveur : annuler (délai de garde)
- Ordre de paiement transmis et accepté (transaction) : le client finira par recevoir la clé (si besoin, du serveur)

17

## Tolérance aux fautes Point de vue du fournisseur

### Pannes dans les étapes de 1 à 5

- Aucune transaction financière n'a eu lieu
- Pas de paiement, pas de livraison

### Pannes après émission de [6] (facture)

- Le fournisseur finira par obtenir une réponse du serveur (au besoin en renvoyant la facture [6])
- Propriété transactionnelle sur [6]+[7] (envoi facture et résultat du règlement) : annulation possible si panne durable

18

## Service de commerce électronique

### Commentaires

- Sécurité
  - Confidentialité (secret des informations)
  - Intégrité (pas de modifications non désirées)
  - Authentification
    - Des partenaires
    - Du contenu des messages
- Tolérance aux fautes
  - Atomicité des transactions commerciales (paiement + livraison)
  - Garanties assurées par le serveur (état défini, opérations transactionnelles)
  - Pas d'hypothèses sur sites extérieurs au serveur

19

## Caractéristiques et besoins des applications réparties

Organisation

Coordination

Communication

Sécurité

Disponibilité

Capacité de croissance

Génie logiciel

- Construction
- Administration

20

## Organisation des applications réparties

### Client-serveur

- Exécution « synchrone » requête- réponse
- Extensions : serveurs coopérants; service discontinu

### Objets partagés (organisations diverses)

#### Flots de communication

- Discrets (messages) ou continus (multimédia)

### Code mobile

- Machine virtuelle - cache l'hétérogénéité
- Problèmes de sécurité

### Agents

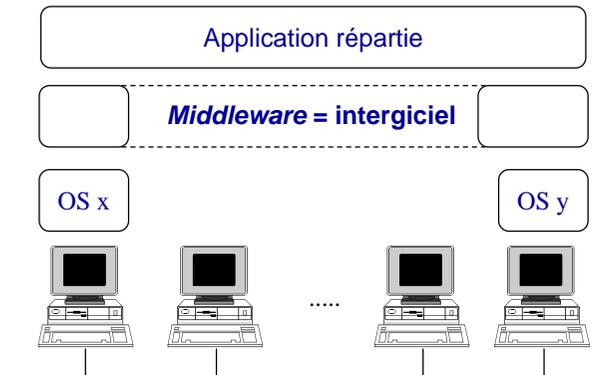
- Programme agissant pour le compte d'une entité cliente
- Agents fixes ou mobiles, statiques ou adaptatifs
- Coopération entre agents

21

## Organisation des applications réparties Un schéma commun : le « middleware »

« *Middleware is everywhere* »

© IBM

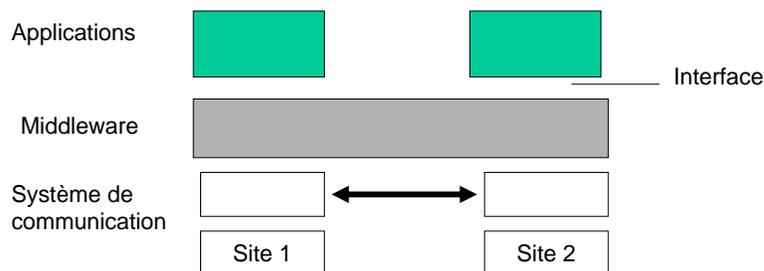


22

## Organisation des applications réparties Un schéma commun : le « middleware »

« *Middleware* »: couche logicielle destinée à

- Masquer l'hétérogénéité des machines et des systèmes
- Masquer la répartition des traitements et données
- Fournir une interface commode aux applications (modèle de programmation + API)



23

## Bibliographie

W. Rosso. *Java RMI*. O'Reilly, 2002.

D. Flanagan. *Java in a Nutshell*. O'Reilly, 2001.

Kurose, Ross. *Computer Networking*. Addison Wesley, 2001.

J. Siegel. *CORBA 3 Fundamentals and Programming*. Wiley, 2000.

J.-M. Geib, C. Gransart, P. Merle. *CORBA: des concepts à la pratique*. Dunod, 2ème édition, 1999.

B. Meyer. *Object-Oriented Software Construction*. Prentice Hall 1997.

R. Orfali, D. Harkey, J. Edwards. *The Essential Client/Server Survival Guide*. Wiley 1996.

A. Tanenbaum. *Distributed Operating Systems*. Prentice-Hall 1995.

Le site Penser en Java : <http://penserenjava.free.fr/>

24