




Intergiciels

Dr. Philippe Merle
 ADAM / INRIA Futurs – GOAL / LIFL - USTL
<http://www.lifl.fr/~merle>




Contenu du cours

1. Intergiciels
 - ◆ Principes, classes, fonctions et illustrations
2. Patrons de conception des intergiciels
 - ◆ Briques élémentaires communes

■ Cours non exhaustif car multitude de travaux de recherche et de solutions industrielles

© Novembre 2007, P. Merle Intergiciels 2




Applications réparties

■ De nombreuses applications informatiques sont réparties de fait

- ◆ Services Internet, e.g. DNS, courrier électronique, news, etc.
- ◆ Sites Web de commerce électronique
- ◆ Carte bleue
- ◆ Téléphonie cellulaire
- ◆ Télévision interactive
- ◆ Gestion des réseaux de télécommunication
- ◆ Grid computing
- ◆ Peer to Peer



■ Définition minimaliste : coopération d'un ensemble de logiciels s'exécutant sur plusieurs machines reliées par des réseaux de communication

© Novembre 2007, P. Merle Intergiciels 3

 **Pourquoi des applications réparties ?** 

- **Besoins propres des applications**
 - ◆ Partage et/ou transfert d'information
 - ◆ Accès à des ressources (matérielles ou logiciels) distantes
 - ◆ Coopération / synchronisation de traitements répartis
 - ◆ Répartition « intrinsèque » des ressources
- **Possibilités techniques**
 - ◆ Coût et performances des machines et des communications
- **Besoin d'interconnexion généralisée**
 - ◆ Système d'information des entreprises
 - ◆ Interpénétration informatique - télécom - télévision

© Novembre 2007, P. Merle Intergiciels 4

 **L'interconnexion des systèmes d'information** 


Ventes


Ingénierie


Fabrication


Comptabilité

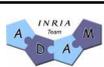

Livraison/
réception

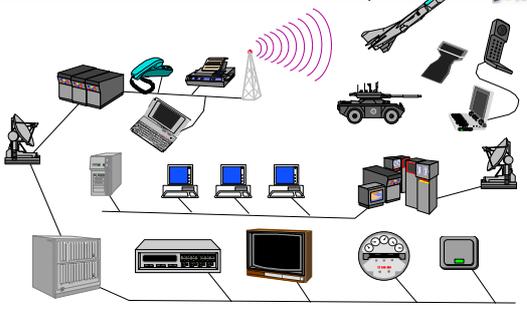

Inventaire


 Paiement/
facturation

**Les entreprises ont besoin de logiciels coopérants !
Les entreprises réparties ont besoin de logiciels distribués !**

© Novembre 2007, P. Merle Intergiciels 5

 **L'interconnexion des réseaux de communication** 



Les équipements répartis ont besoin de logiciels distribués !

© Novembre 2007, P. Merle Intergiciels 6

INRIA **Problèmes à prendre en compte** INRIA

- **Hétérogénéité**
 - ◆ Réseaux
 - ◆ Machines
 - ◆ Systèmes d'exploitation
 - ◆ Langages de programmation
 - ◆ Formats d'échange de données
 - ◆ ...
- **Répartition**
 - ◆ Communication / coopération / coordination des logiciels répartis
 - ◆ Sécurité de fonctionnement contre les
 - ❖ Pannes du matériel et du logiciel
 - ❖ Fraudes => sécurité
 - ◆ ...
- **Interopérabilité entre logiciels hétérogènes et répartis**

© Novembre 2007, P. Merle Intergiciels 7

INRIA **Introduction à l'intergiciel** INRIA

■ **L'intergiciel est la couche "du milieu" (Middleware)**

© 2003-2006, S. Krakowiak Intergiciels 8

INRIA **Fonctions de l'intergiciel** INRIA

■ **L'intergiciel a quatre fonctions principales**

- ◆ Fournir une interface ou API (*Applications Programming Interface*) de haut niveau aux applications
- ◆ Masquer l'hétérogénéité des systèmes sous-jacents (OS, machines, réseaux)
- ◆ Rendre la répartition invisible ("transparente") des traitements et données
- ◆ Fournir des services répartis d'usage courant

■ **L'intergiciel vise à faciliter la programmation répartie**

- ◆ Développement, évolution, réutilisation des applications
- ◆ Portabilité des applications entre plates-formes
- ◆ Interopérabilité d'applications hétérogènes

© 2003-2006, S. Krakowiak Intergiciels 9

 **Classes d'intergiciel** 

- **Objets et composants répartis**
 - ◆ Java RMI, CORBA, .NET, Enterprise Java Beans, ...
- **Message-Oriented Middleware (MOM)**
 - ◆ Message Queues, Publish-Subscribe, ...
- **Intégration d'applications**
 - ◆ Web Services, JBI, SCA, ...
- **Coordination**
 - ◆ Jini, outils de *workflow*, ...
- **Accès aux données, persistance**
 - ◆ JDBC, JDO, SDO, ...
- **Support d'applications mobiles**
 - ◆ Informatique ubiquitaire, réseaux de capteurs, ...

© 2003-2006, S. Krakowiak Intergiciels 10

 **Services principaux d'un intergiciel** 

- **Communication**
 - ◆ ≠ modes : appels synchrones client/serveur, messages asynchrones
- **Annuaire**
 - ◆ Désignation par noms symboliques
 - ◆ Recherche / courtage par propriétés
- **Transaction**
 - ◆ Tolérance aux pannes d'une séquence d'opérations
 - ◆ Propriétés ACID (Atomicité – Cohérence – Isolation – Durabilité)
- **Sécurité**
 - ◆ Authentification utilisateurs, protection des échanges, etc.
- **Persistance**
 - ◆ Accès aux sources de données (BD, fichiers)
 - ◆ Stockage et interrogation
- **Conteneur**
 - ◆ Isolation entre applications
 - ◆ Fourniture / contrôle des services / ressources utilisés
- **Administration**
 - ◆ Installation, activation, supervision, reconfiguration, etc.

© Novembre 2007, P. Merle Intergiciels 11

 **Standards pour l'interopérabilité** 

- **Nécessité de normaliser la couche intergiciel !**
- **Standards sont des consensus pour gérer l'interopérabilité**
 - ◆ Définis en commun
 - ◆ Appliqués par tous
- **Exemples de standards**
 - ◆ J2EE de SUN
 - ◆ .NET de Microsoft
 - ◆ CORBA de l'OMG
 - ◆ Web Services du W3C
- **De nombreux organismes de standardisation**
 - ◆ International Standardization Organization (ISO)
 - ◆ International Telecommunication Union (ITU)
 - ◆ TeleManagement Forum
 - ◆ Object Management Group (OMG)
 - ◆ World Wide Web Consortium (W3C)
 - ◆ ...



© Novembre 2007, P. Merle Intergiciels 12

Quelques modèles de composants pour construire les applications

- Modèles orientés IHM / client**
 - Sur machines utilisateurs
 - OLE, COM, ActiveX de Microsoft
 - Java Beans de SUN
- Modèles orientés métier / serveur**
 - Sur machines serveurs
 - Conteneurs en liaison avec BD, transactions, sécurité
 - COM+, MTS,NET de Microsoft
 - Entreprise Java Beans de SUN
- Modèles généralistes**
 - Fractal du consortium ObjectWeb
 - Composants CORBA du consortium OMG

© Novembre 2007, P. Merle Intergiciels 13

Environnement d'exécution d'un composant

© Novembre 2007, P. Merle Intergiciels 14

Environnement d'exécution d'un composant

- Conteneur**
 - Encapsulation d'un composant (et ses composantes)
 - Prise en charge (masque) les services systèmes
 - nommage, sécurité, transaction, persistance ...
 - Prise en charge partielle des communication
 - invocations et événements
 - Techniquement par interposition (ou délégation)
- Structures d'accueil**
 - Souvent nommées serveurs d'applications
 - Espace d'exécution des conteneurs et des composants
 - Médiateur entre les conteneurs et les services systèmes
 - Chargement des paquetages de composants

© Novembre 2007, P. Merle Intergiciels 15

Environnement d'exécution d'un composant

- Idée sous-jacente**
 - Composants = code métier + configuration des services techniques
 - Conteneur + serveur = services techniques (non fonctionnels)
- Objectifs**
 - Faciliter conception, développement et maintenance
 - Meilleure évolutivité / évolutions des solutions techniques

© Novembre 2007, P. Merle Intergiciels 16

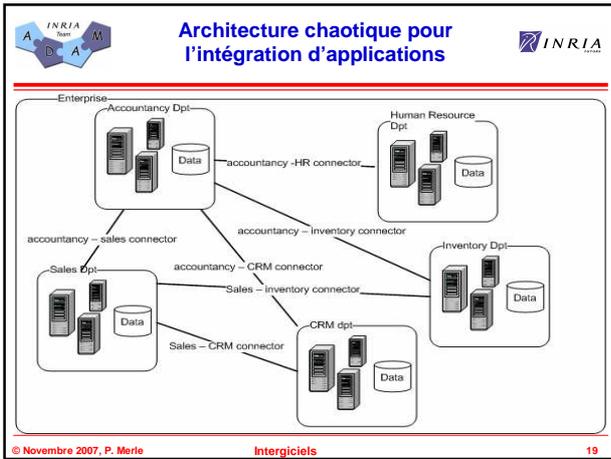
Notion de serveur d'applications Standard JEE – serveur OW2 JOnAS

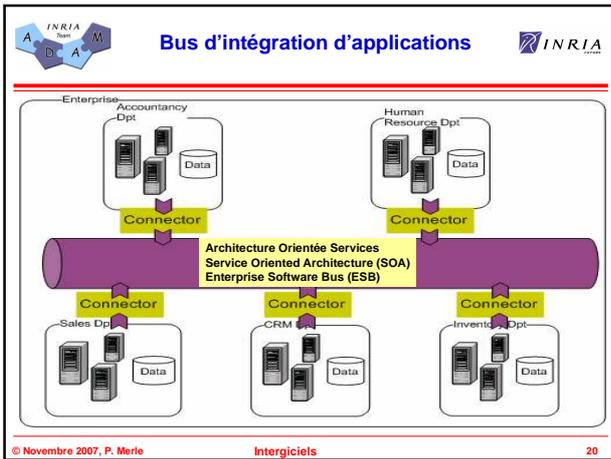
© Novembre 2007, P. Merle Intergiciels 17

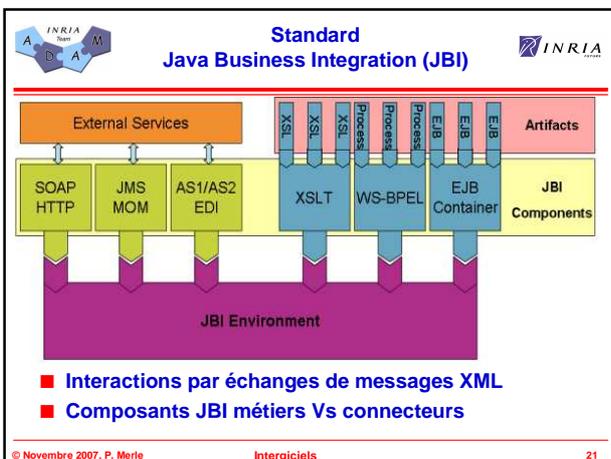
L'interconnexion des systèmes d'information

Les entreprises ont besoin de logiciels coopérants !
Les entreprises réparties ont besoin de logiciels distribués !

© Novembre 2007, P. Merle Intergiciels 18







INRIA **PETALS**
un environnement JBI distribué

■ Construit avec des composants Fractal
■ Projet R&D ANR JONES (début 2005 – fin 2007)

© Novembre 2007, P. Merle Intergiciels 22

INRIA **Standard Service Component Architecture (SCA)**

■ Mariage SOA et composants

- ◆ Divers langages d'implantation (Java, C++, etc.)
- ◆ Divers langages d'interface (Java, WSDL, etc.)
- ◆ Diverses couches de transport (SOAP, Java RMI, CORBA, etc.)

© Novembre 2007, P. Merle Intergiciels 23

INRIA **Interopérabilité entre composants SCA et Fractal**

Projet R&D ANR SCOrWare (début 2007 – fin 2008)

Composants Fractal pour SOA Composants SCA reconfigurables

© Novembre 2007, P. Merle Intergiciels 24

