

# Architecture JEE.

## *Objectifs attendus*

- ***Serveurs d'applications JEE***
  - Systèmes distribués
- ***Architectures JEE***
  - ***Normes JEE***
    - couches logicielles, n-Tiers
    - framework JEE et design patterns

# Objectifs à atteindre .

- **Répondre à la question**
  - qu'est ce que l'architecture JEE ?
- **Mettre en Pratique :**
  - réaliser une application JEE dans projet de 40h par équipe de 4



# Cible JEE.

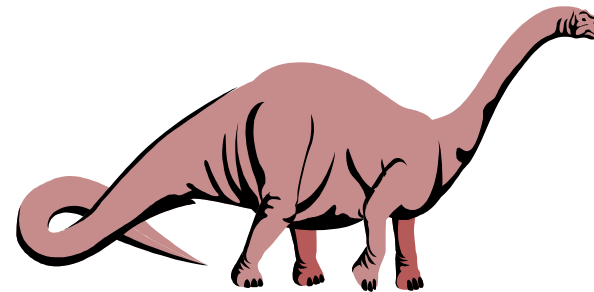
- *JEE , pour quelle informatique?*
- *Qu'est-ce qu'un serveur d'applications JEE ?*

# Systemes distribués.

- **Division des traitements en modules indépendants**
  - plus disponible
  - plus évolutif
  - plus maintenable



Systemes distribués : petits,  
rapides et facilement adaptables



Systemes monolithique : gros,  
lents et inadaptables

# Systemes distribués.

- **Disponibilité**
  - Définition
  - Haute disponibilité

# Systemes distribués.

- **Évolutivité**
  - Définition
  - Capacités de traitement

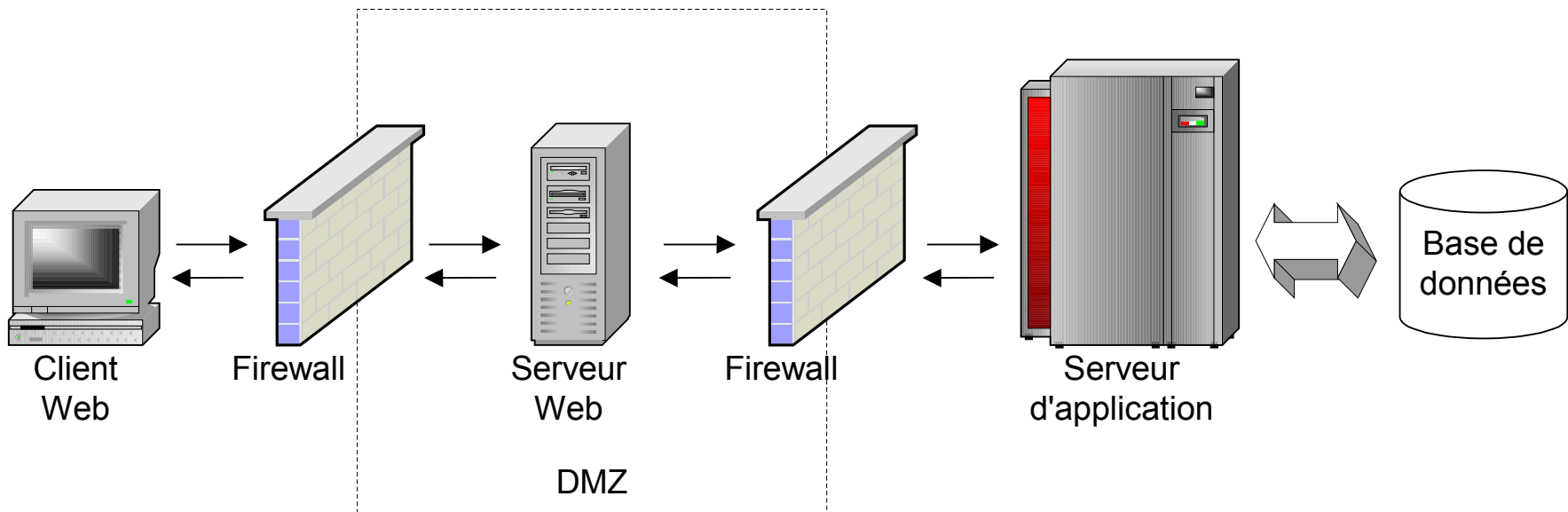
# Systemes distribués.

- **Maintenabilité**

- Définition
- Solution

# Serveurs d'applications JEE.

## ■ Architecture Web



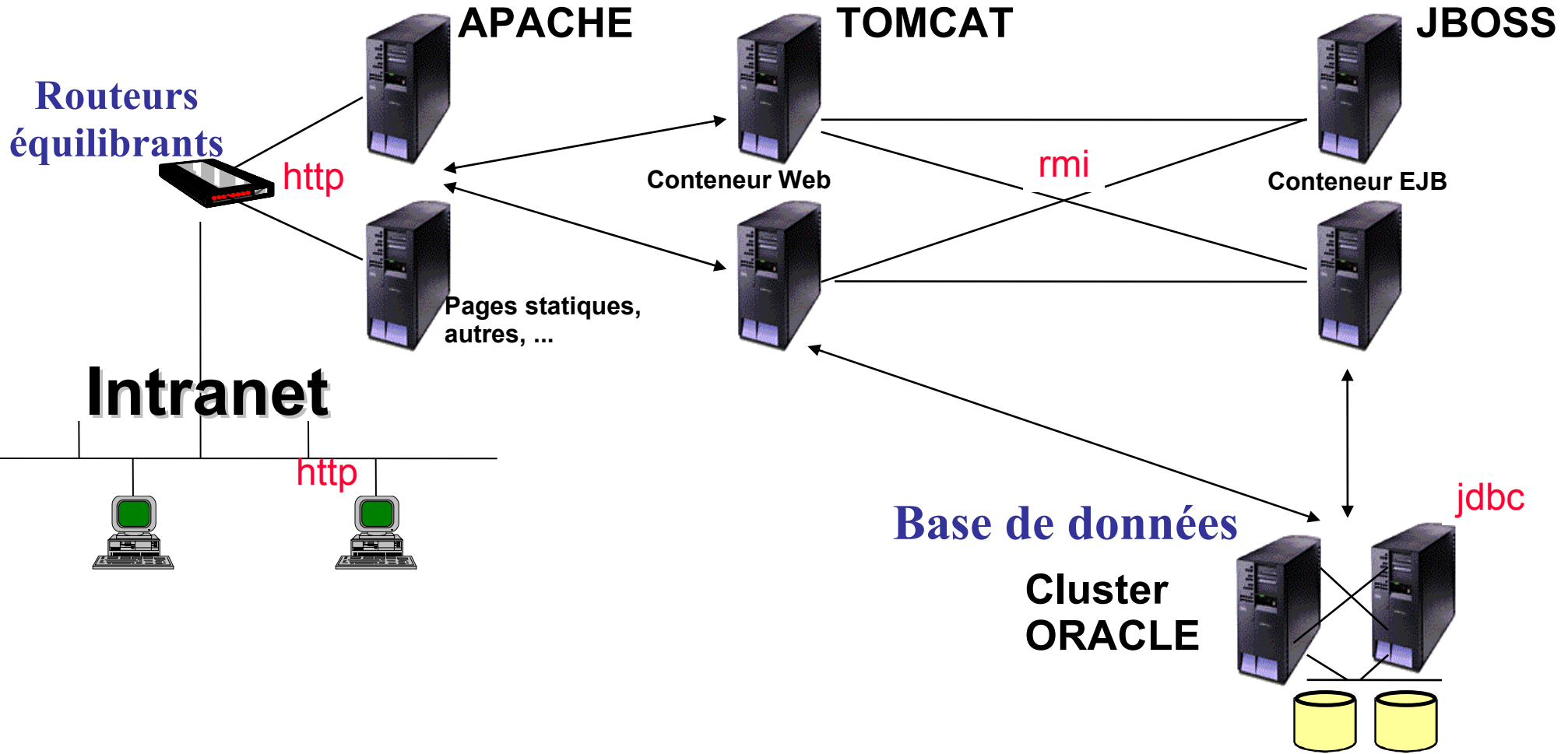


# Architecture: schéma de principe

Serveurs web

Serveurs présentation

Serveurs métiers



# Serveurs d'applications JEE.

## ■ Client Web

- un navigateur
- interprète les pages HTML ou XML
- exécute les applets ou du code JavaScript
- possède différents niveaux de sécurité configurable
- peut interagir avec un serveur d 'application via HTTP

# Serveurs d'applications JEE.

- **Application cliente**
  - applications autre qu'un navigateur
  - communique via JRMP, IIOP, TCP/IP, ...

# Serveurs d'applications JEE.

## ■ **Serveur Web**

- fourni du contenu Web (HTML, ...)
- communique via HTTP, ...
- traite des requêtes CGI
- peut être un proxy frontal d'un serveur d'applications

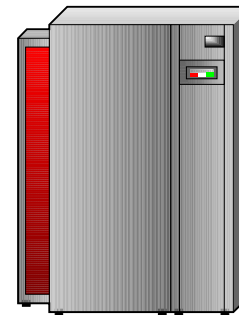
# Serveurs d'applications JEE.

## ■ Machine

- machine physique sur laquelle est installé une ou plusieurs instances de WebLogic Server
- différence entre les machines UNIX et machines non-UNIX (NT)



Machine 1  
NT



Machine 2  
UNIX

# Apport des serveurs d'application**S.**

## (1/2)

- **Permettent d'exécuter des composants**
  - Conformes aux technologies JEE
  - Indépendants du visuel et de l'accès aux données
  - Déployables dans un environnement
    - Permettant une large possibilité d'extension de puissance
    - S'affranchissant du lieu
- **Le composant le plus évolué est un « Enterprise Java Bean »**

# Apport des serveurs d'application<sup>s</sup>.

## (2/2)

- **Services d'administration**
  - Déploiement de servlets et de composants
  - Structuration en serveur, application
  - Gestion d'annuaires JNDI
  - Gestion de Pools et de Data sources
- **Modèle de sécurité applicable**
  - Au niveau de chaque composant
  - Au niveau de chaque méthode

# Implementation de Serveurs d'applications JEE.

- **Gamme WebLogic**
  - WebLogic Server 9.1
    - Compatible JEE 1.4
- **Gamme IBM WebSphere**
  - WebSphere V6
    - Compatible JEE 1.3
- **Gamme OpenSource**
  - Jboss AS v5
    - Compatible JEE 1.4, EJB3



# Ressources.

- **Site Sun sur JEE**

- <http://java.sun.com/JEE/>

- **Site BEA sur JEE**

- <http://dev2dev.bea.com/products/wlserver81/index.jsp>

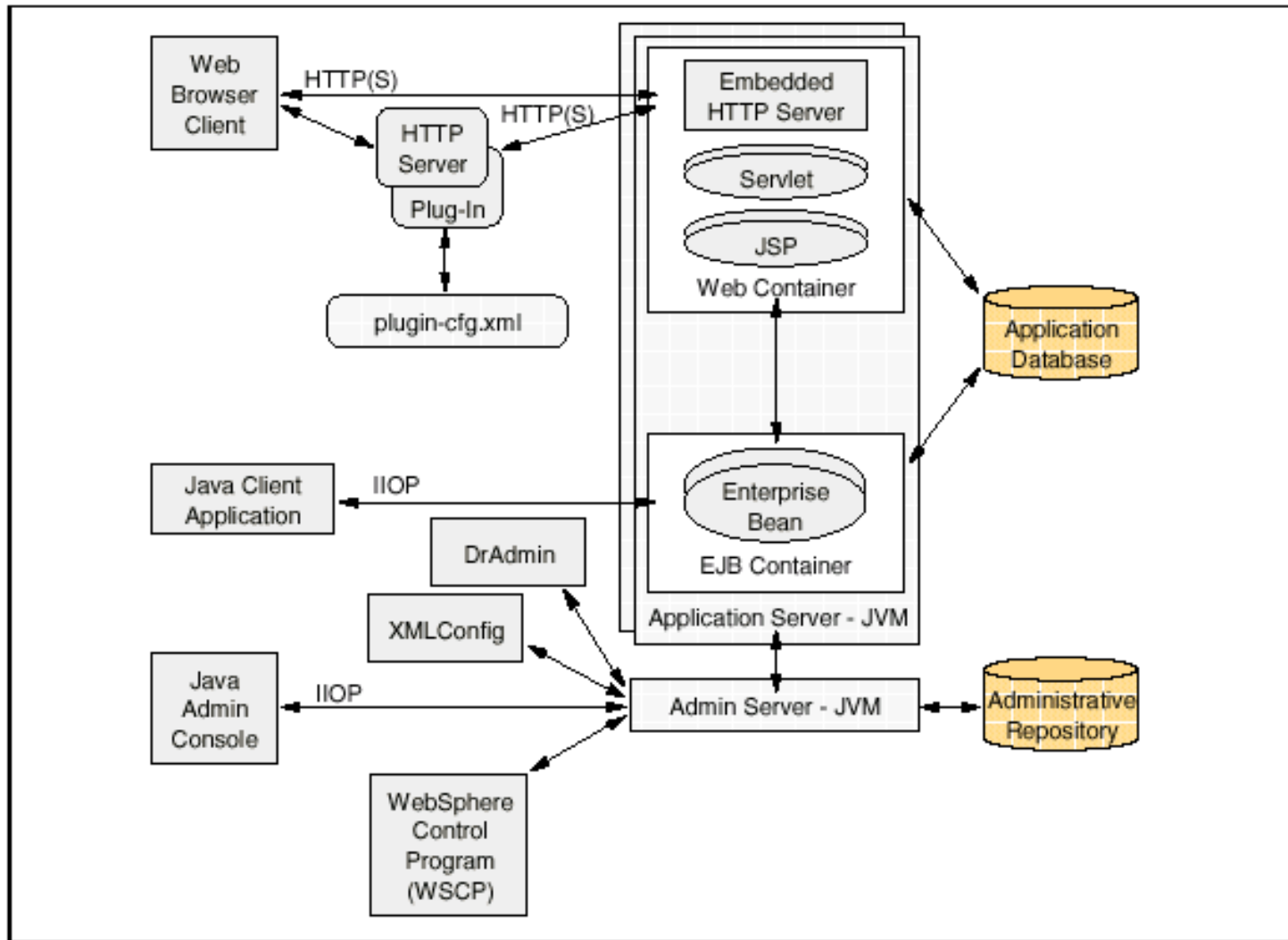
- **Your JEE Community**

- <http://www.theserverside.com/>

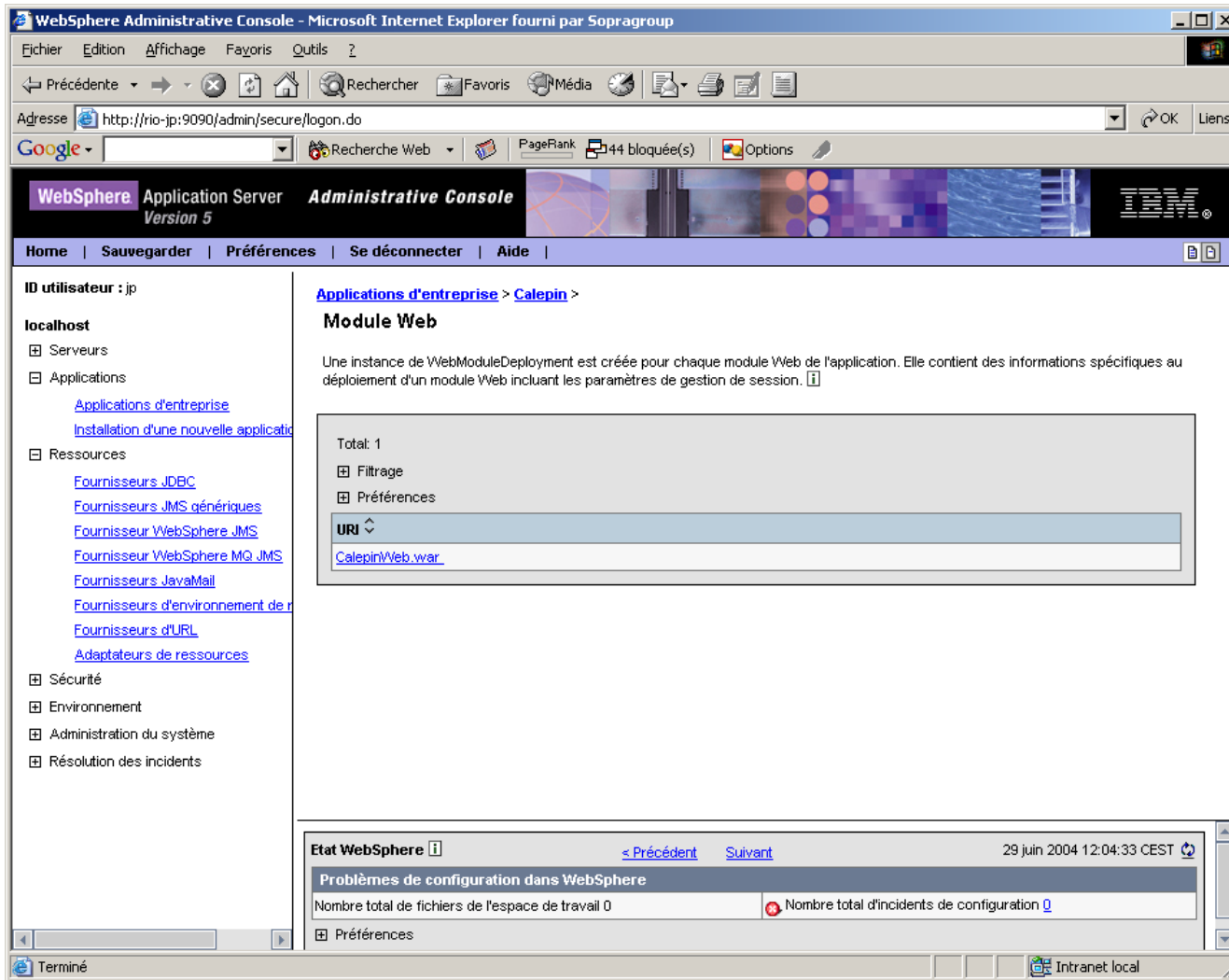
**Focus.**

# Implémentation IBM WebSphere

# Architecture de WebSphere 4/JEE 1.3

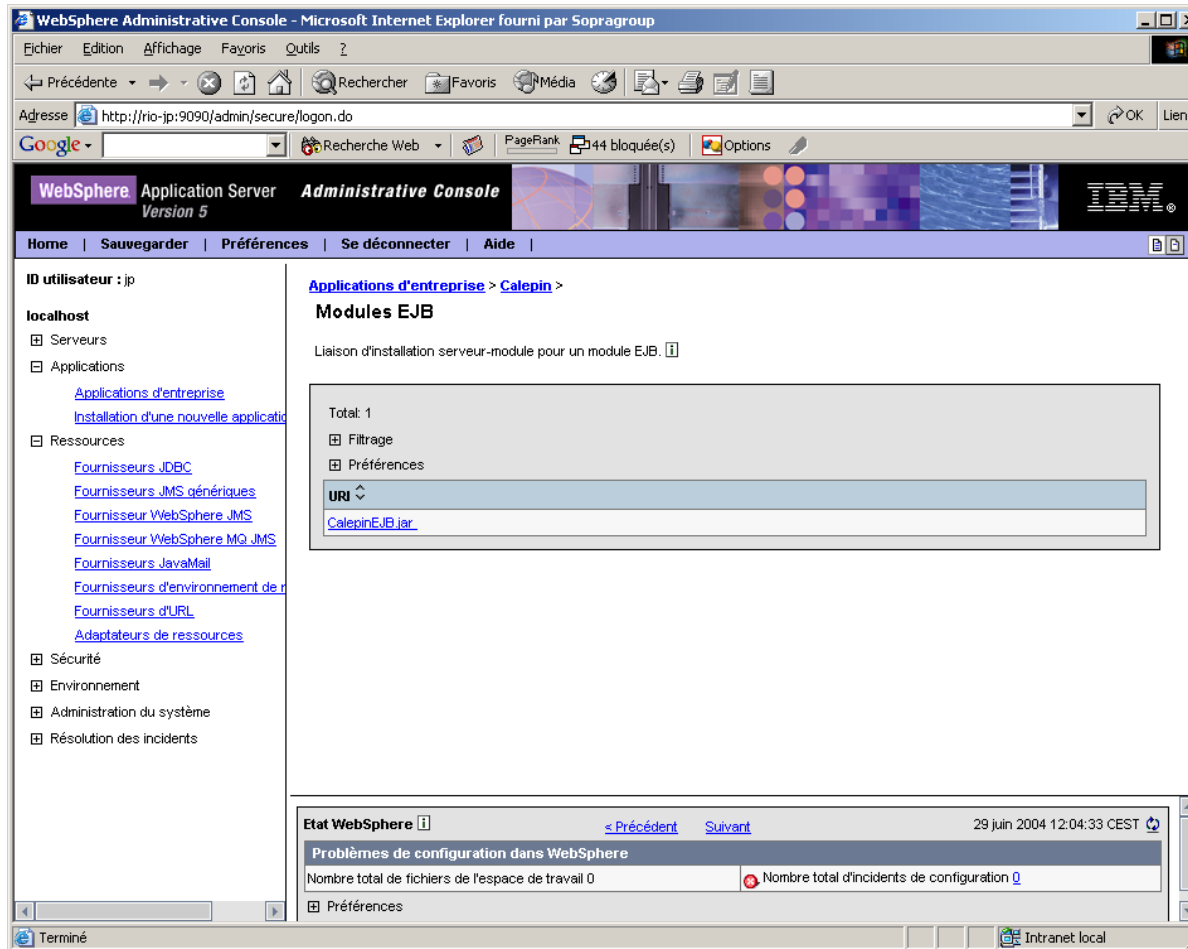


# Web container



## ■ Moteur de servlet 2.3 et JSP 1.2

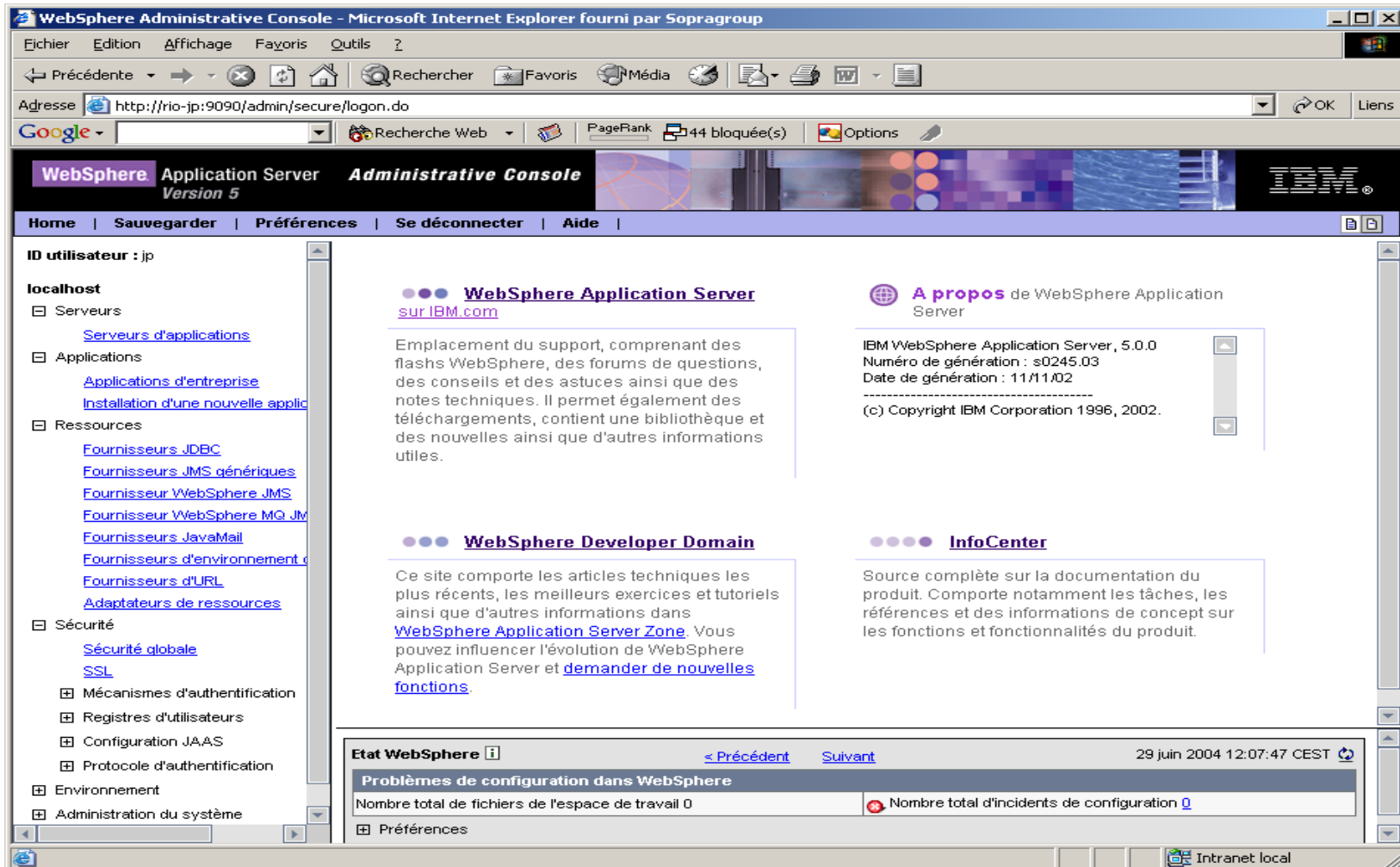
# EJB container



## ■ EJB container

- EJB modules selon la spécification EJB 2

# Web administrative console



# Fin Présentation JEE.

- ***Ce qu'il faut retenir :***
  - Systèmes distribués : disponible, évolutif et maintenable
  - Normes JEE : des nombreuses technologies JAVA qui évoluent
  - Serveur d'application JEE : un web container, un EJB container et une collection de services mis à disposition des applications

# Architecture JEE.

- ***Normes JEE***

- Les principales technologies JEE

- ***Développement distribué JEE***

- n tiers, n couches logicielles
  - Design Patterns

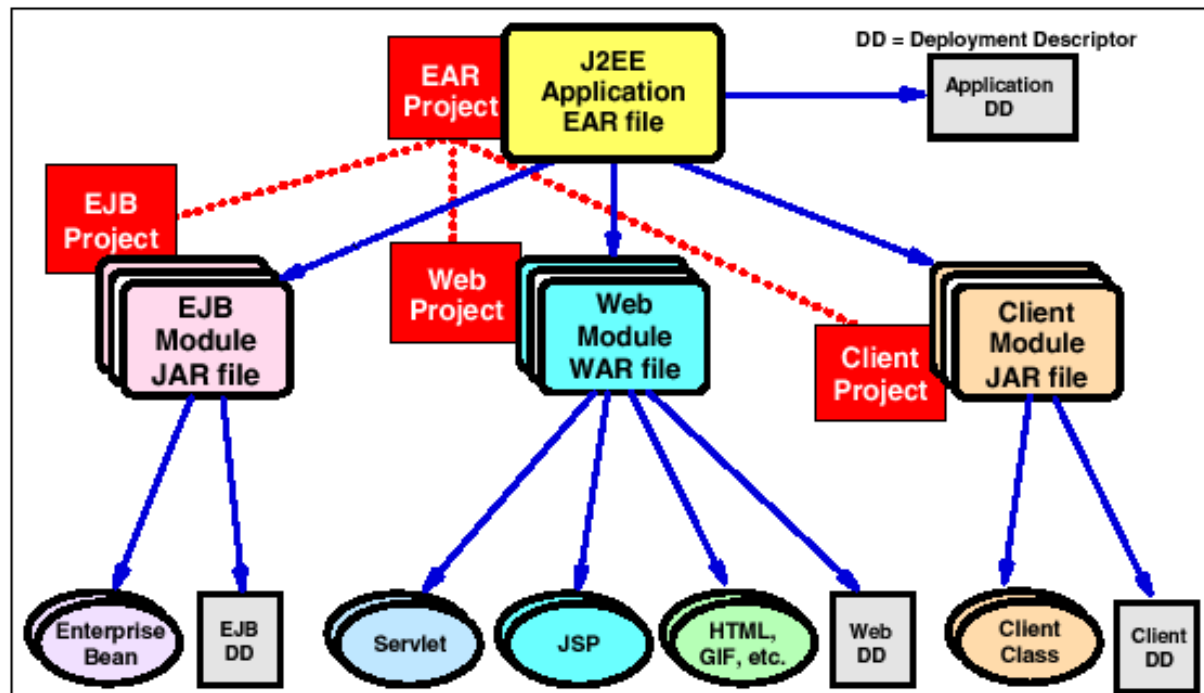


# Normes JEE

- **Définition de Java 2 Platform Enterprise Edition (JEE)**
- **Les applications déployées avec JEE**
  - adhèrent aux standards JEE
  - suivent les spécifications JEE
  - sont écrites en Java
  - sont déployables sur tous les serveurs implémentant JEE
- **Niveau Technologies JEE : JEE 5 et JDK 6**

# Architecture JEE.

- Une application JEE se compose de

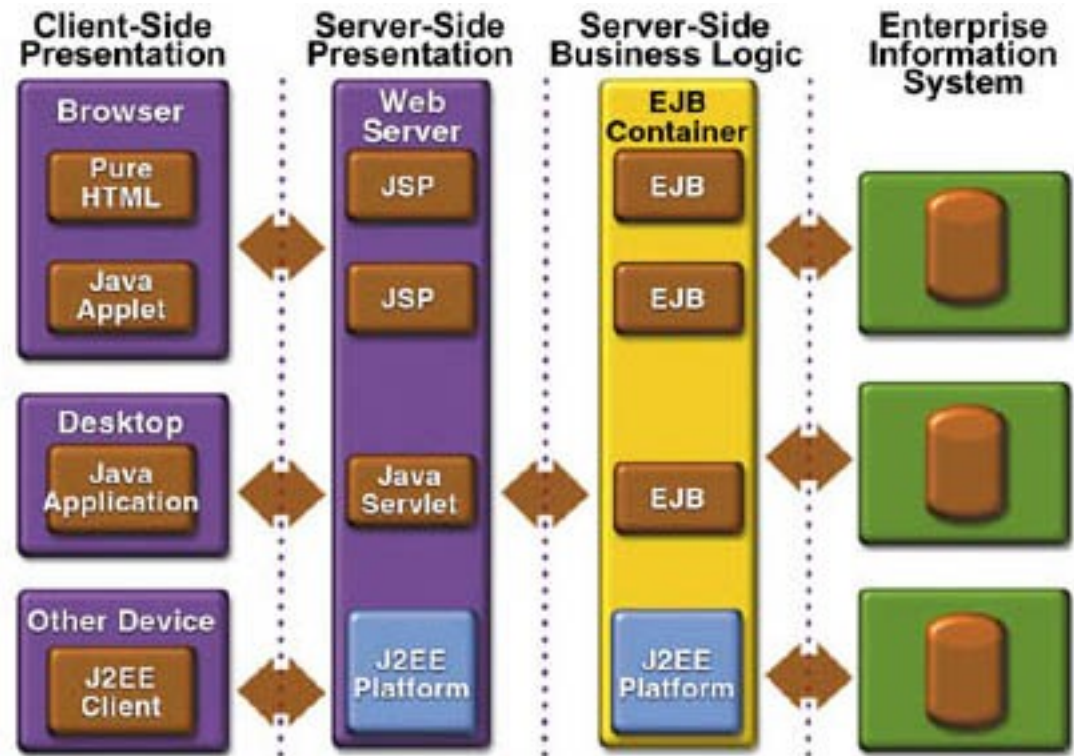


J2EE architecture shows the J2EE hierarchy and the matching support in the Application Developer.

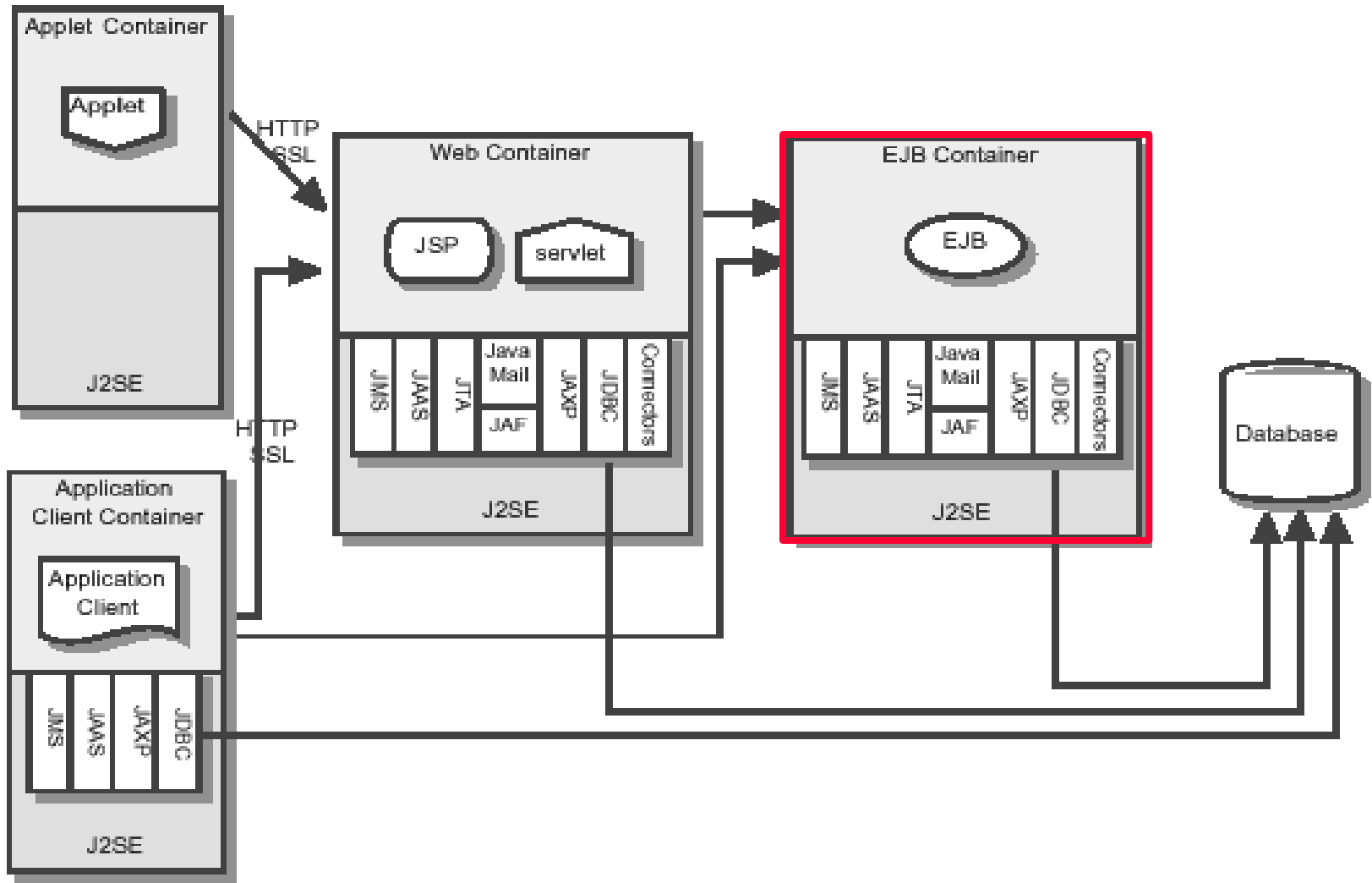
# Architecture JEE.

## ■ Une architecture JEE se décompose en n-tiers :

- partie cliente
- un Web Container
- un EJB Container
- partie métier



# JEE : 4 types de conteneur



# Normes JEE **1.3.**

- **Un serveur d 'application JEE 1.3 supporte les technologies JAVA**
  - 1 Java Data Base Connectivity 2.0
  - 2 Java Naming Directory Interface 1.2
  - 3 Remote Method Invocation (RMI-IIOP) 1.0
  - 4 Java Interface Definition Language
  - 5 Servlet 2.3
  - 6 Java Server Pages 1.2
  - 7 Enterprise Java Beans 2.0

# Normes JEE 1.3.

- 8 Java Transaction Architecture / Java Transaction Service (JTA/JTS) 1.1
- 9 JavaMail 1.2 includes Java Activation Framework 1.0
- 10 Java Message Service (JMS) 1.0.2
- 11 eXtended Markup Language
- 12 Java API for XML Parsing (JAXP) 1.1
- 13 Java API for XML-Based RPC (JAX-RPC) 1.0 (Web Services)
- 14 JEE Connector Architecture (JCA) 1.0
- 15 Java Management Extensions (JMX) 1.0
- Et les services Web

# Normes JEE .

- **Java Data Base Connectivity (JDBC)**
  - interface d'accès aux bases de données
  - spécifications sur les liens entre API et pilote de la base données

# Normes JEE.

- **Java Naming & Directory Interface (JNDI)**
  - API Java d 'accès au service de nommage et de répertoires
  - construit comme un pont vers des fournisseurs d 'annuaire (DNS, LDAP, ...)



# Normes JEE .

## ■ Servlets

- mécanisme de traitement de requêtes/réponses
- servlets HTTP
  - invoquées par des requêtes HTTP
  - déployables sur un serveur Web
  - leur résultat est un flux HTML envoyé au navigateur
  - peuvent gérer des données de niveau session

# Normes JEE .

- **Java Server Page (JSP)**
  - génère des pages Web dynamiquement
  - composée de :
    - code HTML pour la représentation des données
    - directives JSP
    - scriptlets (code Java inséré dans la page JSP)

# Normes JEE .

## ■ **Entreprise Java Beans (EJB)**

- composants distribués écrits en Java
- fournissent des services distribuables et déployables
- réutilisables sur les différents serveurs d'applications
- exécutés dans un conteneur (EJB container)
- intérêts des EJB :
  - distribution
  - sécurité
  - transactionnel
- WebLogic Server ( $\geq 7.0$ ) supporte les EJB 1.1 et les EJB 2.0

# Normes JEE .

- **Java Transaction API (JTA)**
  - API Java de gestion des transactions
  - WebLogic Server supporte les transactions locales et distribuées

# Normes JEE .

## ■ **Java Message Service (JMS)**

- API Java d'accès à un middleware de messagerie
- supporte :
  - le domaine Point à Point
  - le domaine Éditeur/Abonné
  - l'acheminement garanti des messages
  - les sessions transactionnelles

# Normes JEE .

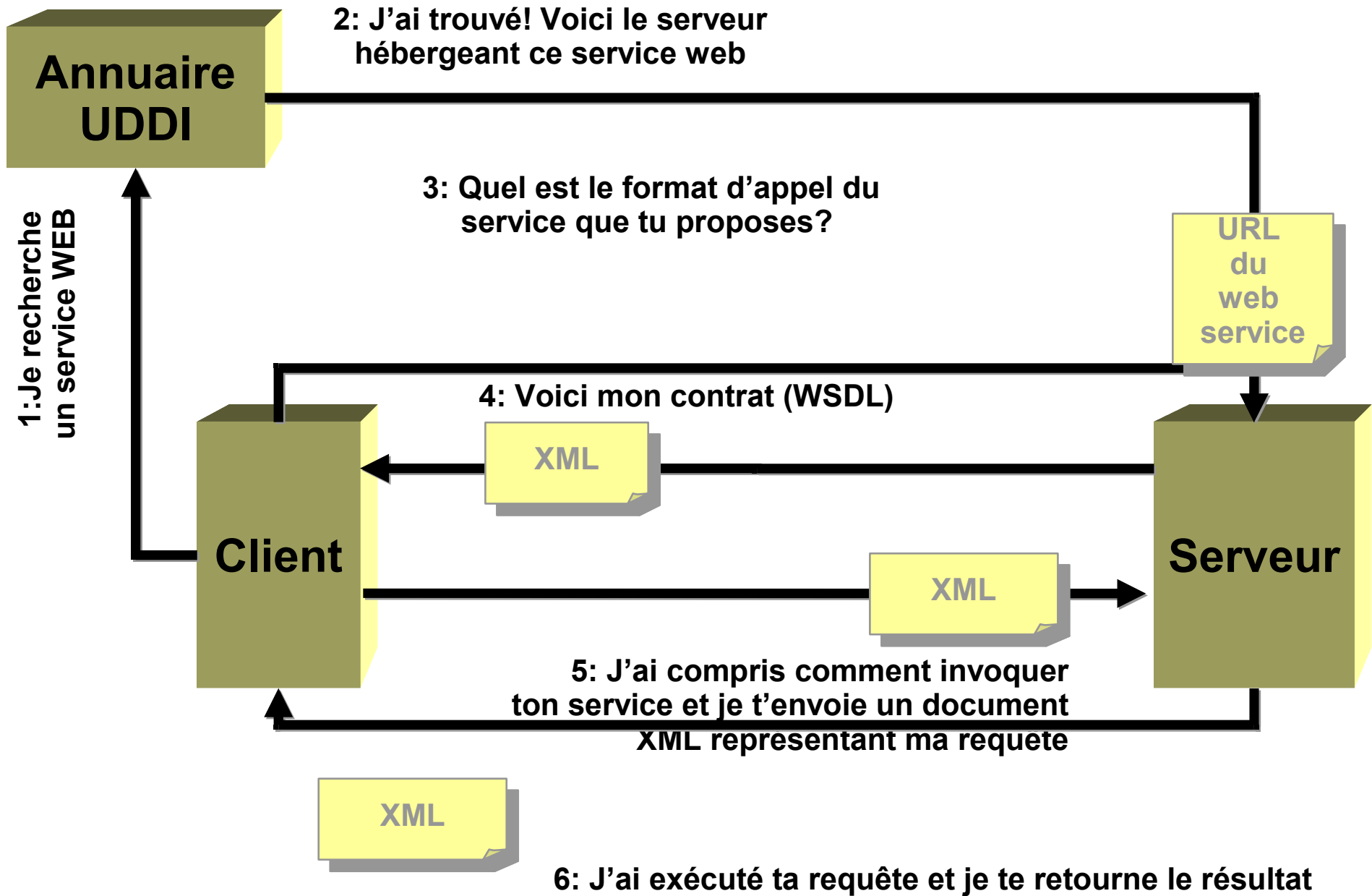
- **Java Management Extension (JMX)**
  - définit un standard de gestion d'infrastructure en Java
  - dissocie les éléments administrés de l'outil d'administration
  - les spécifications décrivent les Mbeans
  - l'administration interne de WebLogic Server est basée sur JMX



# Qu'est-ce qu'un Web Service?

- **Une « unité logique applicative » accessible en utilisant les protocoles standard d'Internet**
  - Réutilisable et basé sur un protocole (SOAP)
  - Indépendamment de
    - la plate-forme (UNIX, Windows, ...)
    - l'implémentation (VB, C#, Java, ...)
    - l'architecture sous-jacente (.NET, JEE, ...)
- **Décrit par un WSDL**
  - Définition d'une interface Web = contrat basé sur XML
  - Définit un schéma pour n'importe quel type d'interface
- **Enregistré dans un annuaire UDDI**
  - permet de retrouver le service dynamiquement

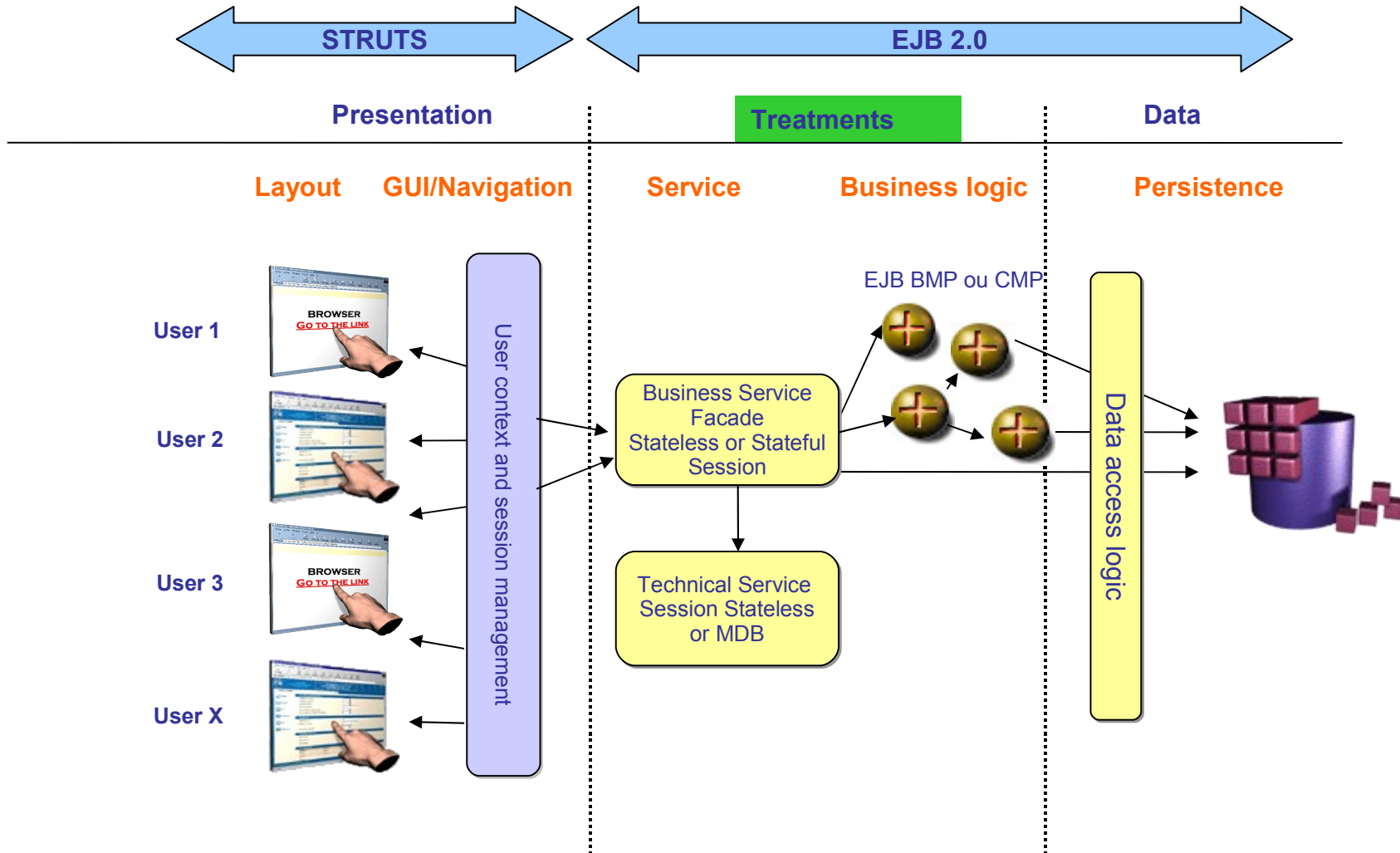




# Définition des couches

- **Couche présentation**
  - Framework MVC Struts 1.1
- **Couche Application (Business Delegate Layer)**
  - Classes simple Java et design patterns
- **Couche métier (Business Layer)**
  - Service
    - Métier : contrôleur métier
    - Technique
  - Objet Métier persistant (EJBs)
- **Couche d'accès aux données (Persistence layer)**
  - Liaison avec entrepôt de données
  - Gestion des opérations CRUD : Create, Read, Update, Delete.
- **Données (Physical Layer)**
  - Entrepôt des données de l'application
  - Schema modele physique de base de données, schema XML,...

# Framework Software Architecture



# Vue générale

## Patterns principaux utilisés

### ■ Business Delegate

<http://java.sun.com/blueprints/patterns/BusinessDelegate.html>

- Reduce coupling between Web and Enterprise JavaBeans™ tiers

### ■ Data Access Object (DAO)

<http://java.sun.com/blueprints/patterns/DAO.html>

- Abstract and encapsulate data access mechanisms

### ■ Service Locator

<http://java.sun.com/blueprints/patterns/ServiceLocator.html>

- Simplify client access to enterprise business services

### ■ Session Facade

<http://java.sun.com/blueprints/patterns/SessionFacade.html>

- Coordinate operations between multiple business objects in a workflow

### ■ Data Transfer Object

<http://java.sun.com/blueprints/patterns/TransferObject.html>

- Transfer business data between tiers

### ■ Fast Lane Reader

<http://java.sun.com/blueprints/patterns/FastLaneReader.html>

- Improve read performance of tabular data

# Architecture JEE. ■

## ■ ***Ce qu'il faut retenir :***

- Norme JEE : ensemble de technologies JAVA (~15) qui évoluent
- Développement application JEE distribuée
  - 4 containers : applet, client, web, EJB
  - Application d'entreprise JEE (EAR)
    - structure JEE de répertoires à respecter
    - Déclaration des modules qui la composent par descripteurs de déploiement XML :
      - *application.xml, client-application.xml, web.xml, ejb-jar.xml.*
- *Architecture n tiers, couches logicielles distribués*
- *Framework JEE et design patterns.*

# Objectifs atteints.



- **Vous savez répondre maintenant à la question**
  - qu'est ce que l'architecture JEE ?
- **Mise en Pratique : projet de 40h par équipe de 4**
  - Animation : 30h encadrés à partir du 7 mars 2006

# Objectifs Module enseignement TD

- Objectifs Module enseignement TD
  - Mettre en œuvre un serveur d'applications JEE
  - Décrire une architecture générale X-Net JEE
  - Utiliser les technologies JEE 1.4
  - Prototyper une application JEE 1.4
  
- Pré requis initiaux
  - Programmation objet avec Java
  - Développement Web avec Java
  - Compréhension de la notation UML
  - Connaissance des techniques de gestion de projet