

Architecture J2EE

Thierry Lecroq
(merci à Alexandre Pauchet (INSA Rouen))

Université de Rouen
FRANCE

Plan

- 1 Historique
- 2 Architecture J2EE
- 3 J2EE et applications Web

Historique (1/6)

Architecture 1 tiers

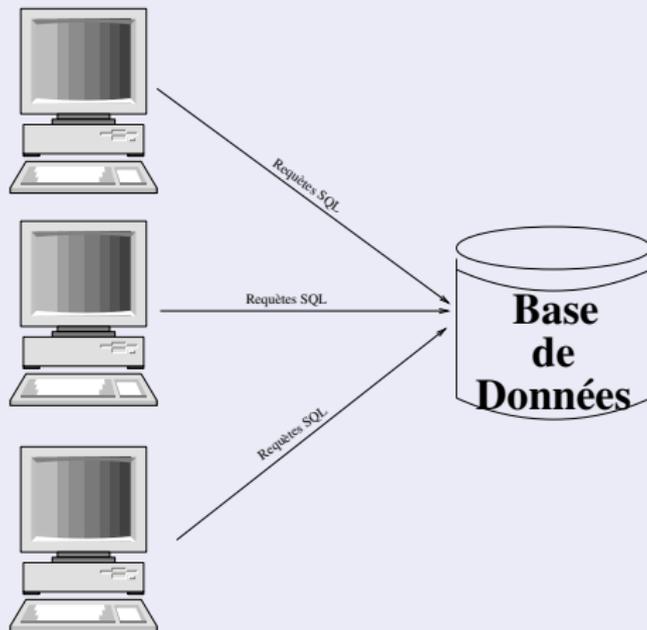
Les années 70, architecture Mainframe (1 tiers)



Historique (2/6)

Architecture 2 tiers

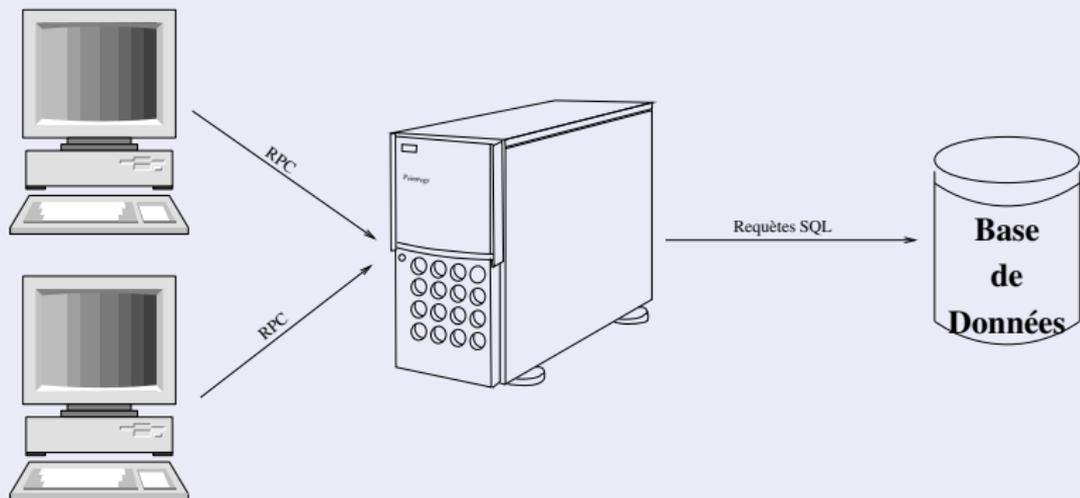
Les années 80, architecture 2 tiers



Historique (3/6)

Architecture 3 tiers

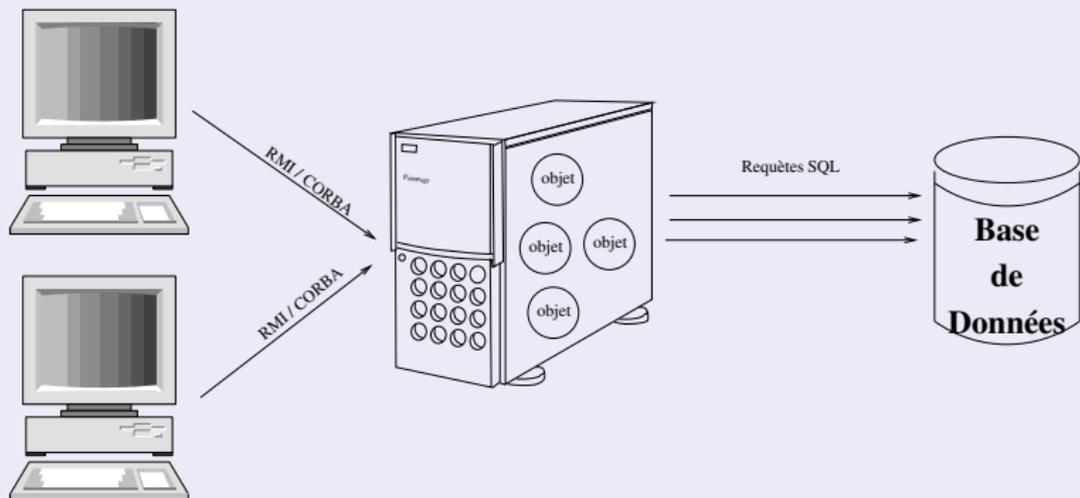
Fin des années 80, architecture 3 tiers (RPC)



Historique (4/6)

Architecture 3 tiers objet

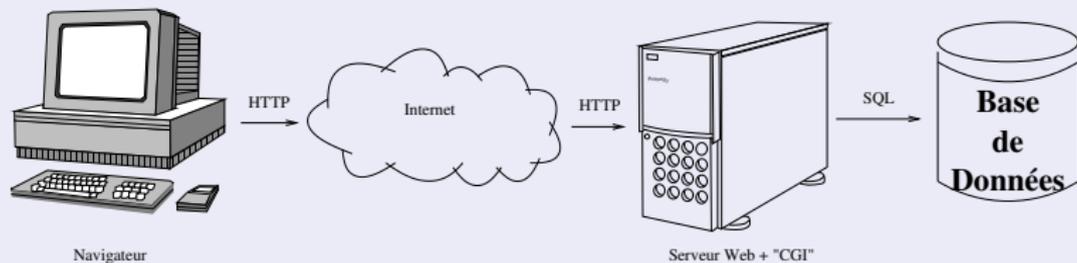
Les années 90, architecture 3 tiers (Objets)



Historique (5/6)

Architecture « Internet »

Les années 90, architecture « Internet »



Historique (6/6)

Paradigmes de programmation

Aspect programmation

- Années 70-80 : Programmation impérative – procédurale
 - ▶ alors que les premiers langages objets datent de la fin des années 60 !!
- Années 90 : Programmation objet
- Fin des années 90 : Programmation par composants
 - ▶ Les composants peuvent être physiquement distants
 - ▶ Si changement, pas besoin de tout recompiler et de tout relinker
 - ▶ Bonne programmation objet : seules les interfaces sont connues

Architecture J2EE (1/5)

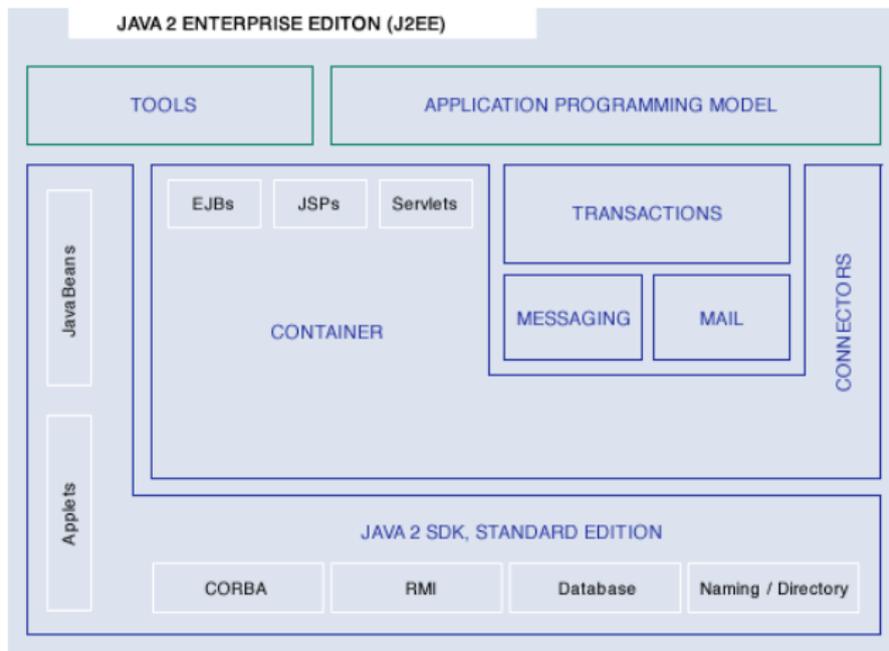
Motivations

Volonté de SUN

- Cadre de développement par composants avec services
- J2EE (Java 2 Edition Enterprise), qui propose des API :
 - ▶ L'invocation de méthodes distantes : RMI, CORBA, Web Services
 - ▶ L'accès aux bases de données relationnelles : JDBC
 - ▶ L'accès aux annuaires et services de nommage : JNDI
 - ▶ L'utilisation du XML : DOM et SAX
 - ▶ HTML dynamique et traitement de requêtes HTTP : JSP et Servlet
 - ▶ La gestion du Mail : Java Mail
 - ▶ La gestion des messages : Java Message Service
 - ▶ La gestion des composants : EJB
 - ▶ La connexion à des ERP : Java Connector
 - ▶ La gestion des droits d'accès : Java Authentication and Authorization Service

Architecture J2EE (2/5)

Description de l'architecture J2EE

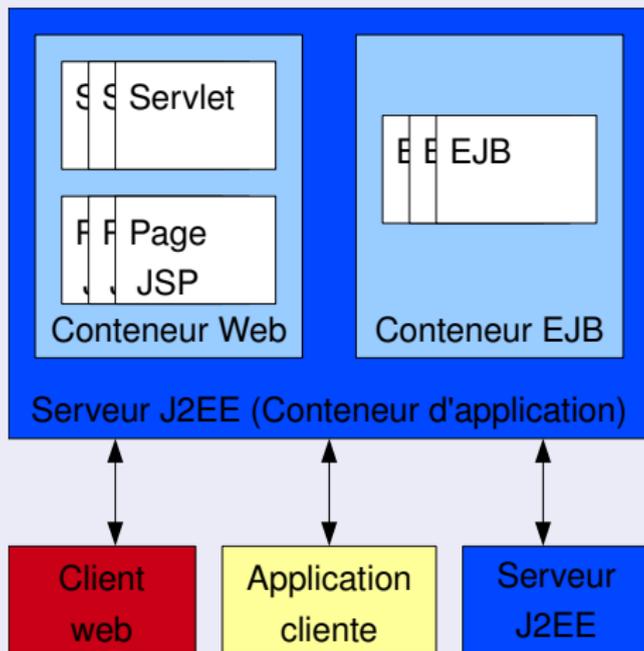


http://www.tria.de/it_consulting/produkte_dienstleistungen/java/java.html

Architecture J2EE (3/5)

Serveur d'application J2EE

Serveur d'application J2EE



Architecture J2EE (4/5)

Exemples de conteneurs

Serveurs J2EE

- JBoss : <http://www.jboss.org>
- Glassfish : <https://glassfish.dev.java.net/>
- JOnAS : <http://wiki.jonas.objectweb.org/xwiki/bin/view/Main/WebHome>
- Oracle AS : <http://www.oracle.com/appserver/index.html>

Conteneur Web

- Tomcat : <http://www.apache.org>
- Jetty : <http://jetty.mortbay.org/jetty/index.html>

Conteneur EJB3

- EasyBeans : <http://www.easybeans.net/xwiki/bin/view/Main/WebHome>

Architecture J2EE (5/5)

Application J2EE

Arborescence d'une archive EAR (Entreprise ARchive)

```
ApplicationJ2EE.ear
|_ fichierEJB.ejb3
|_ applicationWeb.war
|_ META-INF
    |_ MANIFEST.MF
    |_ application.xml
```

J2EE fournit les éléments pour la conception d'applications Web

- **servlets Java et JSP**

- ▶ Les servlets et JSP constituent les blocs de construction du développement d'applications web avec J2EE
- ▶ En terme J2EE, les servlets et pages JSP sont des **composants web**

- **application web**

- ▶ Collection de servlets et de pages JSP, d'autres classes annexes ou de bibliothèques de classes, ainsi que des ressources statiques telles que des documents HTML, XHTML ou XML, images, *etc.*

- **conteneur web**

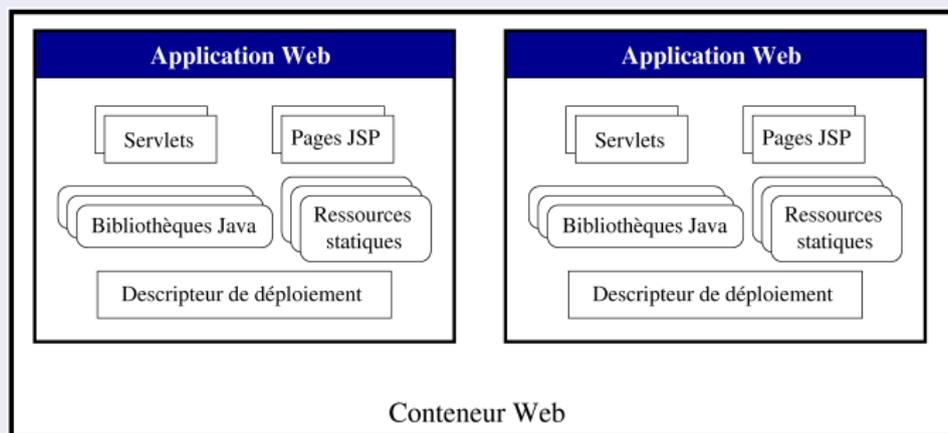
- ▶ Essentiellement un environnement d'exécution Java pour les applications web
- ▶ Responsable de l'initialisation, de l'invocation et de la gestion de la durée de vie des servlets Java et des pages JSP

J2EE et applications Web (2/3)

Application Web

Une application web est composée

- de Servlets/JSP
- de bibliothèques de classes annexes,
- de ressources statiques (images, HTML, XHTML, ...),
- un descripteur de déploiement (fichier XML)



J2EE et applications Web (3/3)

Fichier WAR

Arborescence d'une archive WAR (Web ARchive)

```
ApplicationWeb.war
|_ fichiers.html
|_ fichiers.jsp
|_ repertoires\fichiers
|_ src
|   |_ sourcesServlets.java
|   |_ sourcesbibliotheques.java
|_ META-INF
|   |_ MANIFEST.MF
|_ WEB-INF
    |_ web.xml
    |_ classes
        |_ Servlets.class
    |_ lib
        |_ bibliotheques.jar
```