



Administration de JBoss Application Server 7

- Présentation de Java et Java EE
- Présentation de JBoss EAP 6 et JBoss AS 7
- Le mode standalone : installation et administration
- Administration centralisée
- Administration des applications
- Accès aux données
- Administration en ligne de commande
- Lien avec le serveur HTTP et clustering
- Sécurité

Présentation de Java et Java EE

Présentation de Java

- **Java a été créée chez Sun au début des années 90 :**

- ▶ Annoncé et mis à disposition en 1995
- ▶ Notion d'applet intégrée dans les navigateurs de Netscape et de Microsoft en 1996
- ▶ Principales caractéristiques : portable et gratuit



- **Java est une plate-forme d'exécution d'applications comprenant :**

- ▶ Un langage objet (avec gestion automatique de la mémoire) Une
- ▶ machine virtuelle (JVM : Java Virtual Machine)
- ▶ Un ensemble de librairies. Les librairies officielles sont 'packagées' selon 3 architectures cibles : Java SE, Java EE et Java ME

Les librairies Java

- **Java SE (Standard Edition) :**
 - ▶ Applications clientes avec ou sans interfaces
 - ▶ graphiques Disponible en téléchargement :
 - ▶ JRE (Java Runtime Environment) pour le déploiement, contient la JVM et les librairies
 - ▶ JDK (Java Development Kit) : JRE + outils de base pour le développement + code source
- **Java EE (Enterprise Edition) :**
 - ▶ Applications installées sur des serveurs S'appuie sur Java SE
 - ▶ Nécessite un serveur d'applications
- **Java ME (Micro Edition) :**
 - ▶ Applications embarquées (Téléphone portable, PDA, ...)

La machine virtuelle Java (1)

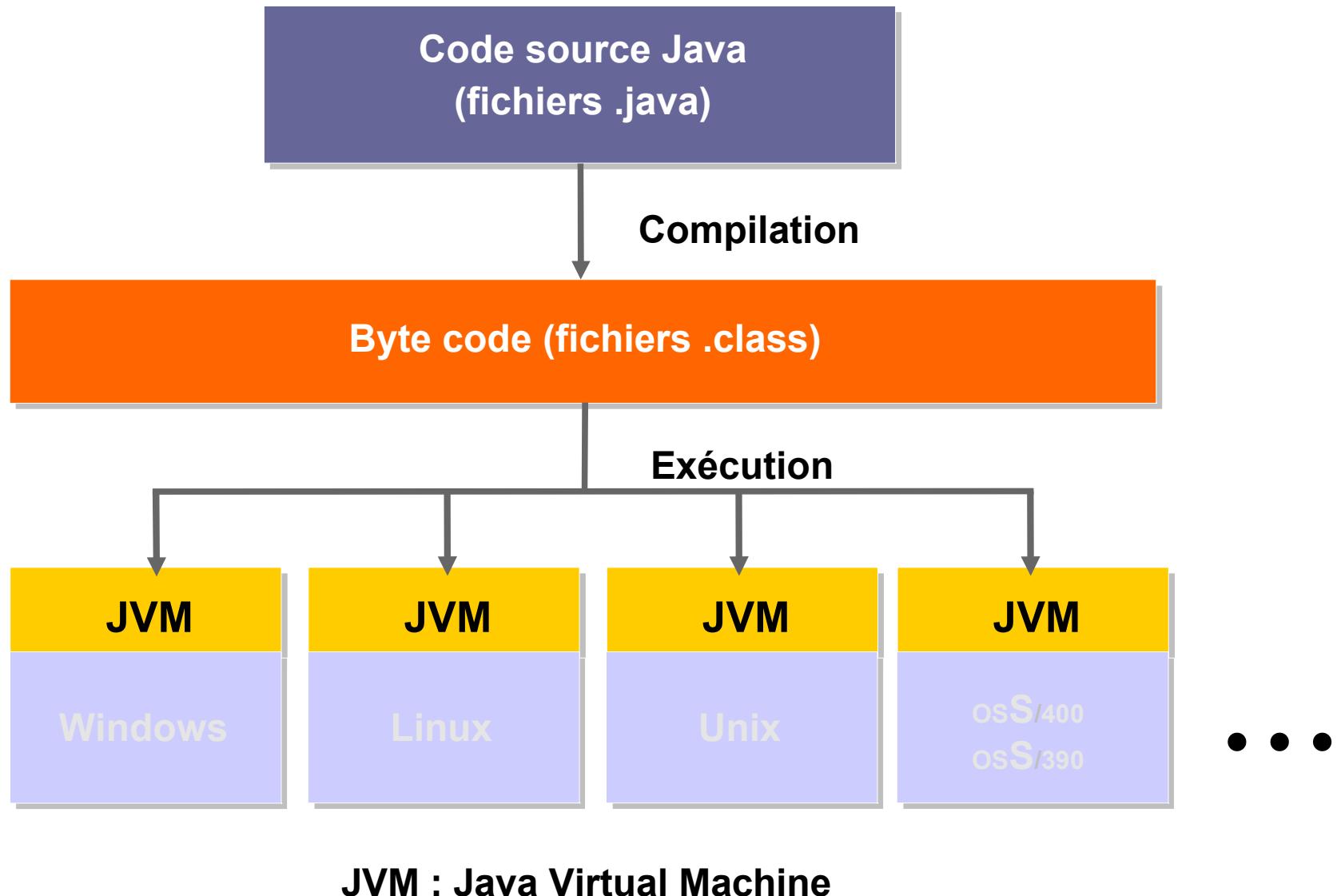
- **Les applications Java sont décrites dans des fichiers texte contenant le code source :**

- ▶ Fichiers .java

- **Les fichiers .java ne sont pas déployés :**

- ▶ Un compilateur est utilisé pour produire des fichiers binaires
 - ▶ Le compilateur génère du code binaire spécifique à la plate-forme Java et non directement du code machine
 - ▶ Ce code binaire est appelé bytecode
 - ▶ Le bytecode déployé est le même quelque soit la plate-forme matérielle
 - ▶ Le bytecode n'est pas directement exécutable, il est reconnu et exécuté par une machine virtuelle Java
 - ▶ La machine virtuelle est un exécutable spécifique à chaque plate-forme

La machine virtuelle Java (2)



Les fichiers JAR

- **Structure des applications :**

- ▶ Une application Java est constituée de nombreux fichiers (1 classe = 1 fichier .java)
- ▶ Les fichiers sources sont organisés en répertoires. Le terme 'package' est utilisé (1 répertoire = 1 package). Le compilateur produit un ou plusieurs fichiers .class par fichier .java

- **L'organisation en packages est conservée :**

- ▶ Une application Java est donc une arborescence de répertoires contenant des fichiers .class

- **Pour simplifier le déploiement, l'arborescence est livrée dans des 'fichiers archives' :**

- ▶ Fichiers JAR (Java Archive), extension .jar
- ▶ Techniquement ce sont des fichiers zip

Java Entreprise Edition (Java EE)

- **Java s'est progressivement imposé côté serveur grâce aux spécifications J2EE :**
 - ▶ Spécifications techniques couvrant les besoins des applications serveur, notamment pour les applications Web
 - ▶ Participation de grands acteurs (IBM, Oracle, Sun, SAP, ...)
 - ▶ Existence d'implémentations commerciales (IBM WebSphere, Oracle WebLogic, ...) et open-source : JBoss, Glassfish, Jonas, Tomcat, Jetty...
 - ▶ Portabilité assurée entre les serveurs d'applications
- **Depuis 2006 le nom officiel est Java EE**

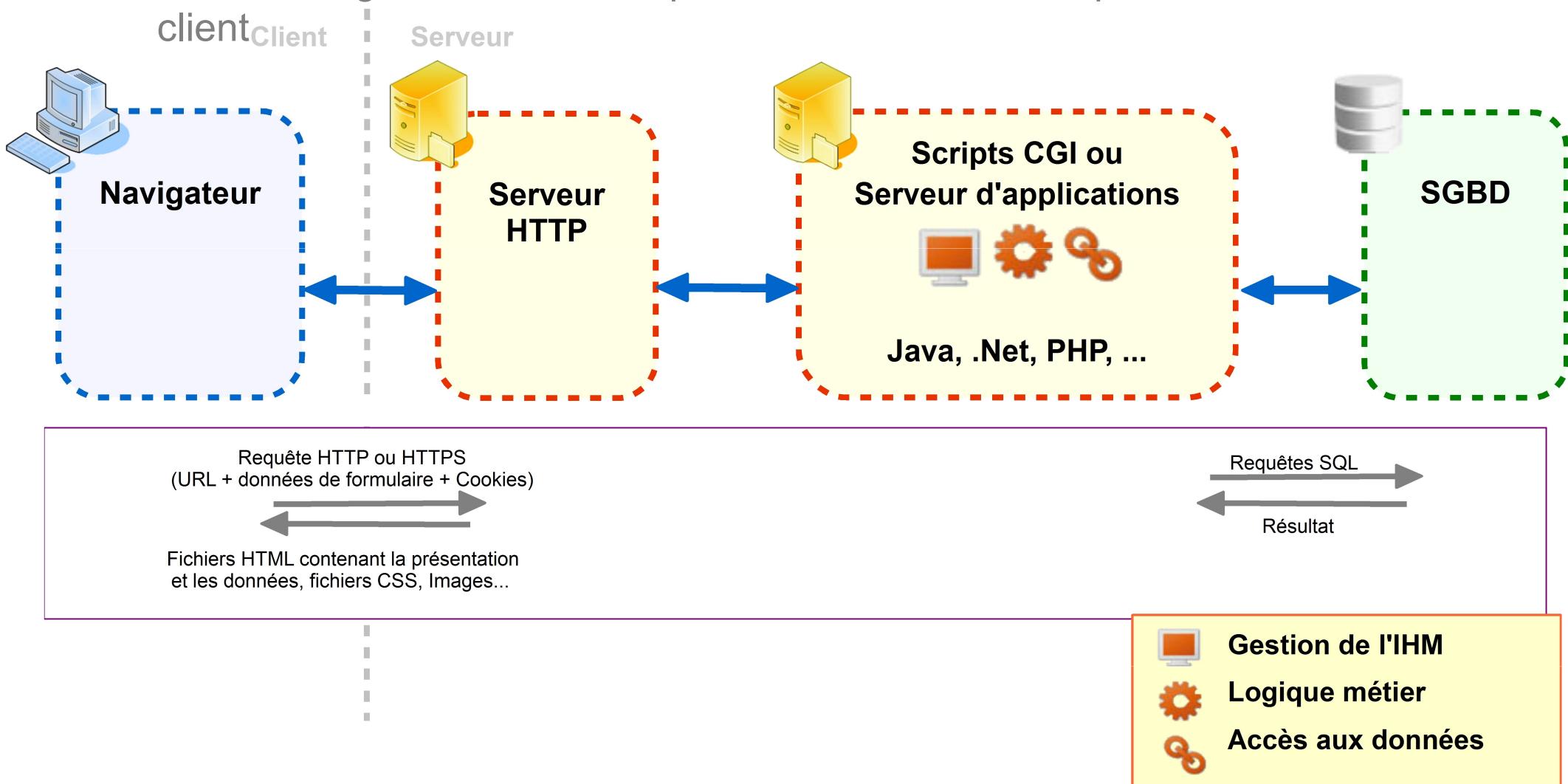
Contenu de la spécification Java EE

- **Un modèle applicatif :**
 - ▶ Servlet, JSP et EJB
 - ▶ Complété par les JSF depuis Java EE 5
- **Des services techniques :**
 - ▶ Gestion des transactions, accès aux données, sécurité, ...
 - ▶ Certains services de Java SE sont largement utilisés par les serveurs Java EE : CORBA, RMI, JNDI, JDBC
- **Des formats de livraison et de déploiement :**
 - ▶ Fichiers EAR et WAR
 - ▶ Notion de modules : module Web (WAR), module EJB. Un EAR est composé de plusieurs modules

Les applications Web

- La logique de l'application réside côté serveur

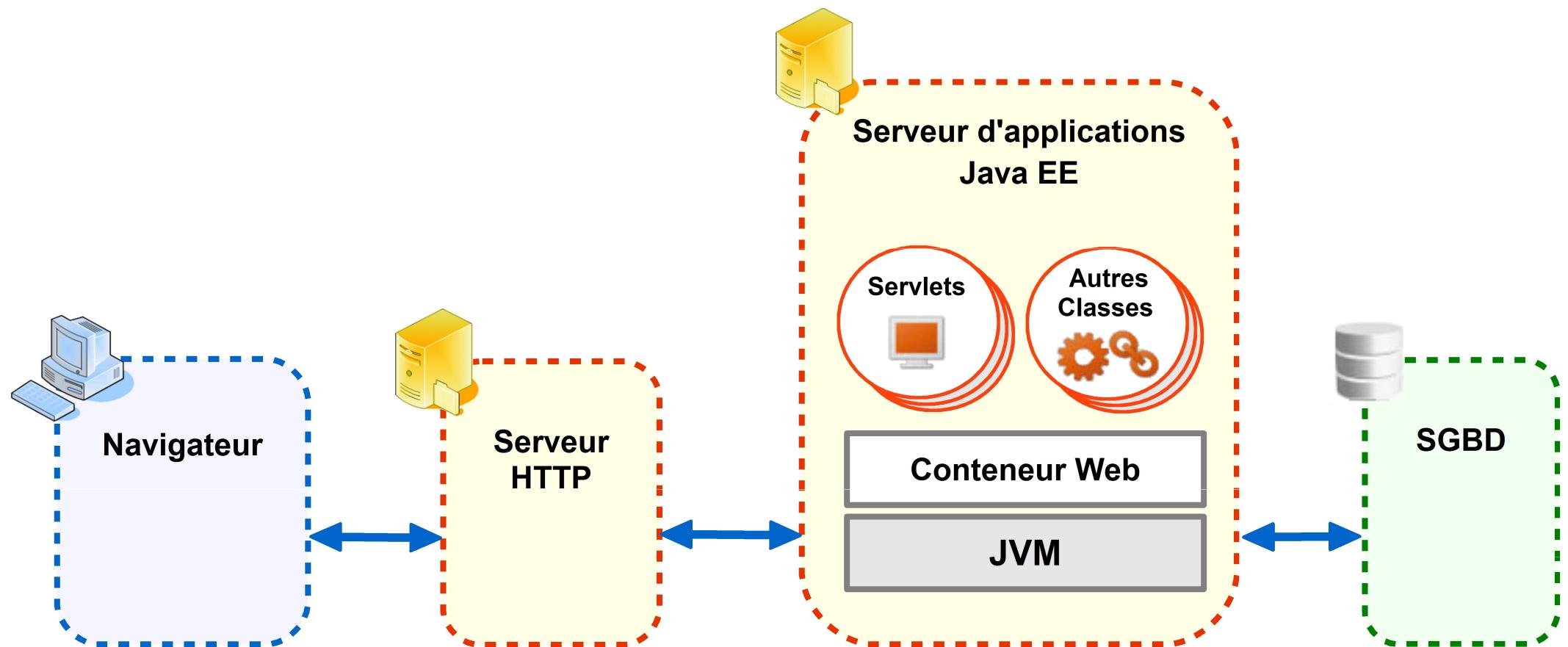
- ▶ Le serveur génère des descriptions d'écran affichées par le client



Les Servlets

- **Classes Java répondant à une requête HTTP**

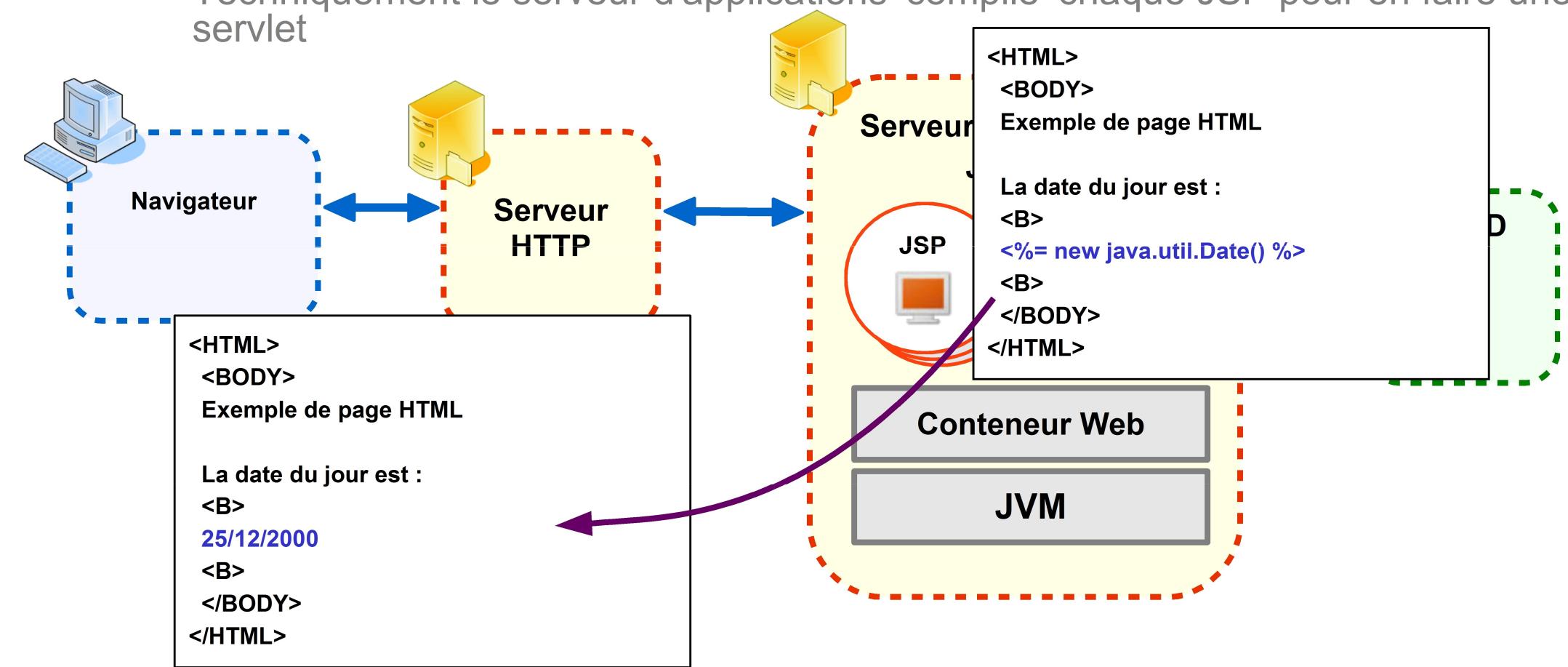
- ▶ Le serveur JEE gère le cycle de vie des servlets et leur invocation en fonction des requêtes HTTP entrantes



Les JSP (Java Server Pages)

- Se présentent comme des pages HTML :

- ▶ Mais elles contiennent du code Java, exécuté côté serveur avant que la page ne soit retournée au navigateur
- ▶ Techniquement le serveur d'applications 'compile' chaque JSP pour en faire une servlet



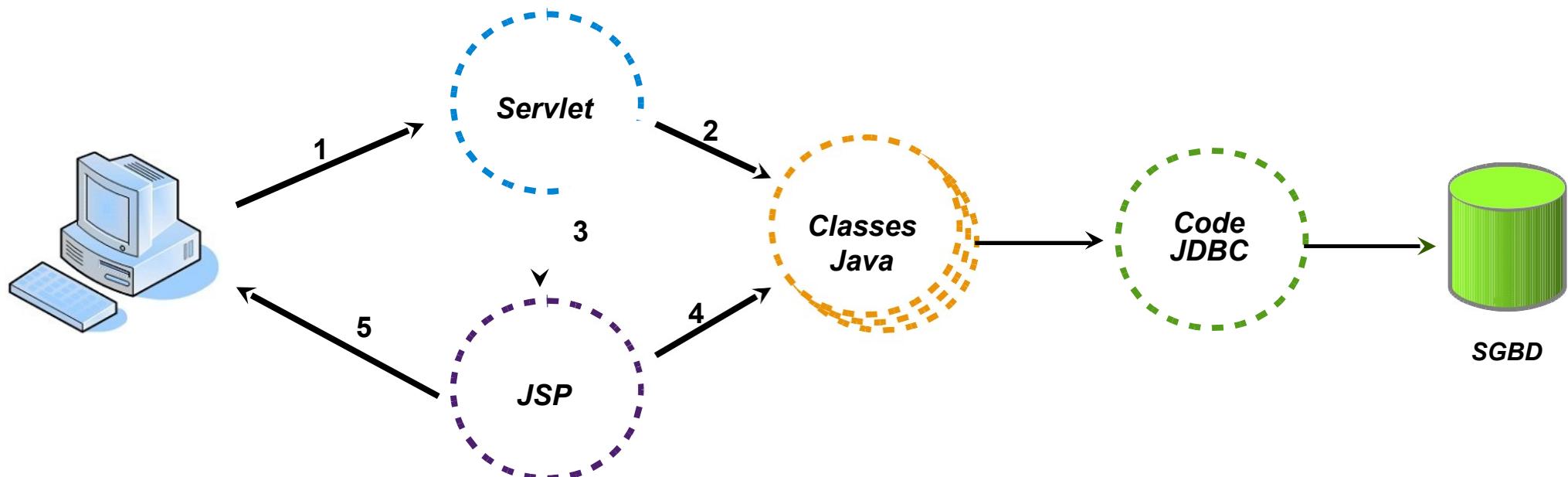
Java EE et le modèle MVC (1)

- **Le modèle applicatif de Java EE est constitué de plusieurs spécifications techniques autonomes :**
 - ▶ Servlets, JSP, JSF et EJB
- **L'utilisation conjointe de ces spécifications permet la mise en œuvre du modèle MVC :**
 - ▶ Séparation en 3 couches : Modèle – Vue – Contrôleur
- **Plusieurs combinaisons sont possibles :**
 - ▶ Contrôleur : Servlet ou JSF
 - ▶ Vue : Servlet, JSP ou JSF
 - ▶ Modèle : POJO ou EJB.
 - ▶ Les classes Java standard sont parfois dénommées JavaBeans
 - ▶ EJB signifie 'Entreprise JavaBean'
 - ▶ JavaBean est un terme générique signifiant 'composant Java'

Java EE et le modèle MVC (2)

- **Le modèle classique :**

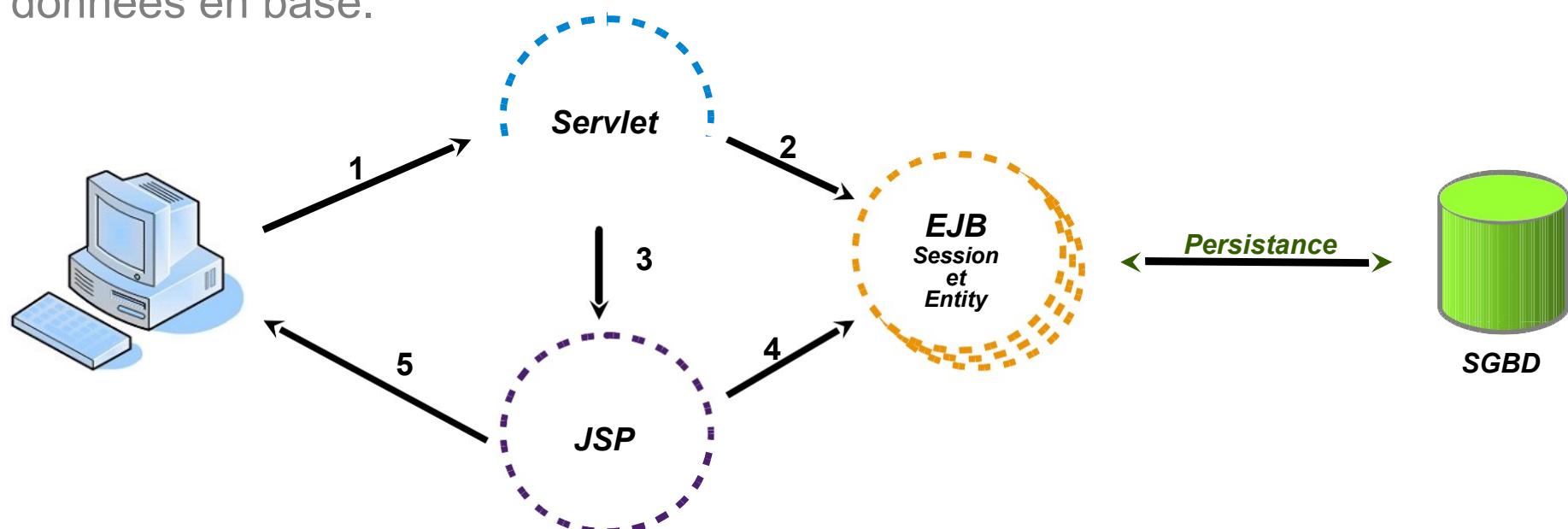
- ▶ Servlet + JSP + JavaBean + code JDBC



Rôles des EJB

- Historiquement, la spécification EJB s'attaquait à deux problématiques :

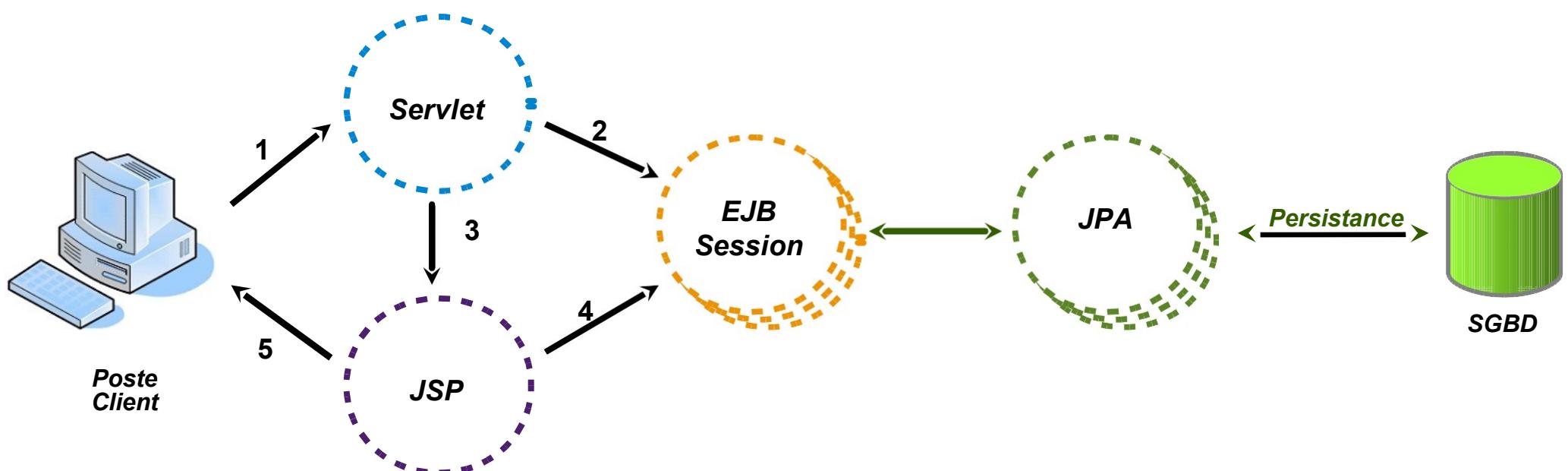
- ▶ Distribution des composants : les composants peuvent être répartis sur plusieurs machines. Les EJB sont accessibles à distance.
- ▶ Persistance des composants : certains composants représentent des données stockées dans une base de données, le conteneur d'EJB gère la synchronisation des variables d'instance d'un composant persistant avec les données en base.



JPA

- Depuis Java EE 5, la gestion de la persistance est définie dans sa propre spécification :

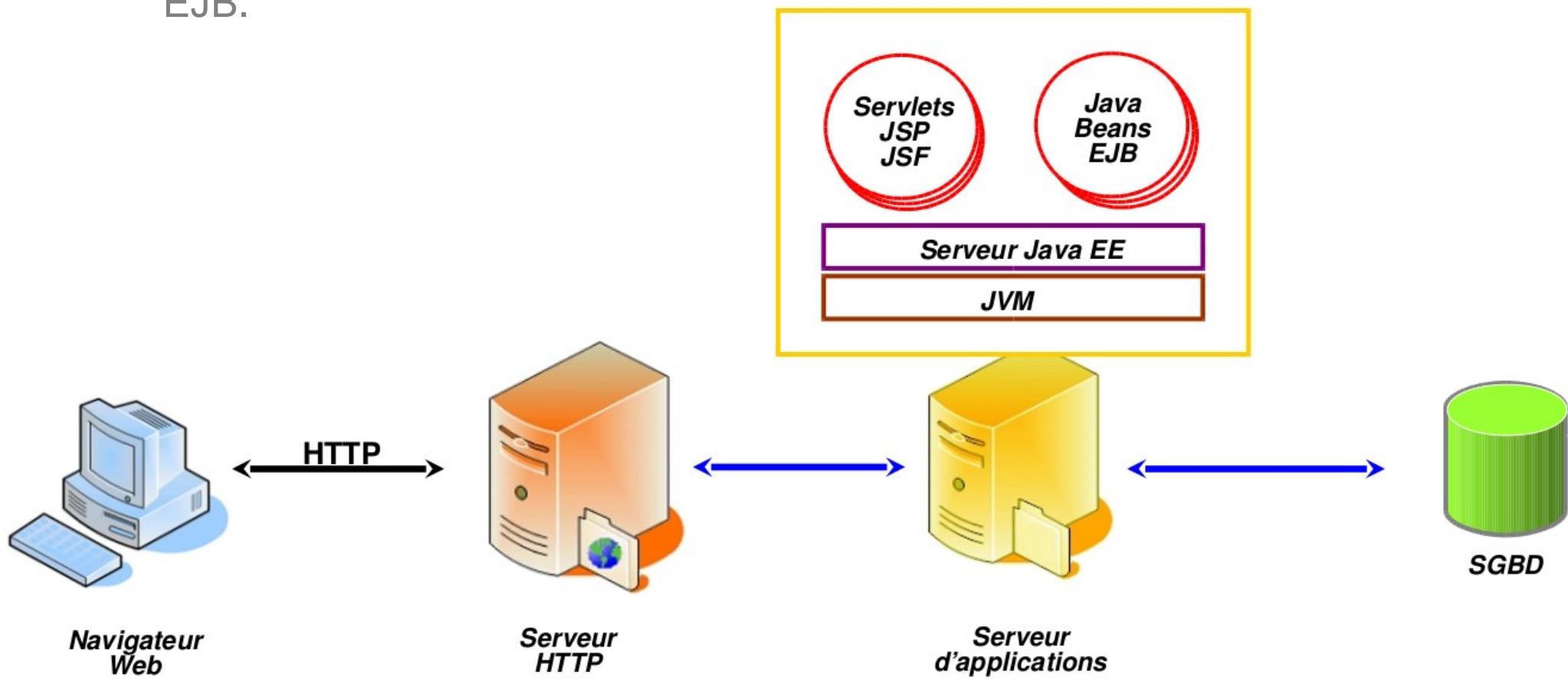
- ▶ JPA (Java Persistence API), remplace la notion d'EJB Entité
- ▶ Hibernate, EclipseLink et OpenJPA sont les principales implémentations de
- ▶ JPA Les EJB sont notamment utilisés pour gérer l'aspect transactionnel :
 - ▶ l'utilisation des EJB peut être remplacé par d'autres solutions, notamment Spring



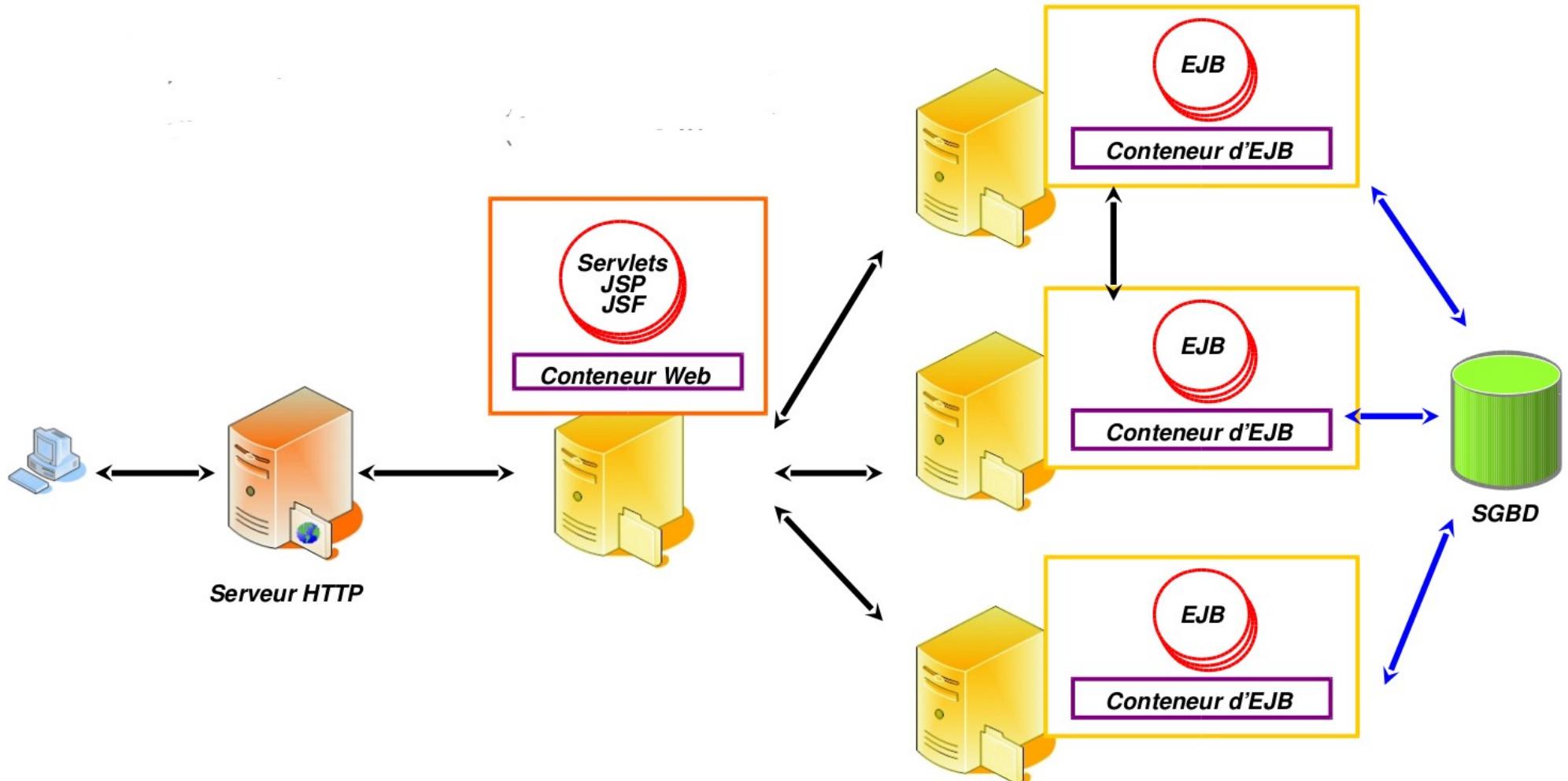
Architectures de déploiement (1)

- Application Web classique :

Partie présentation : Servlet, JSP et JSF. Partie modèle : JavaBeans ou EJB.

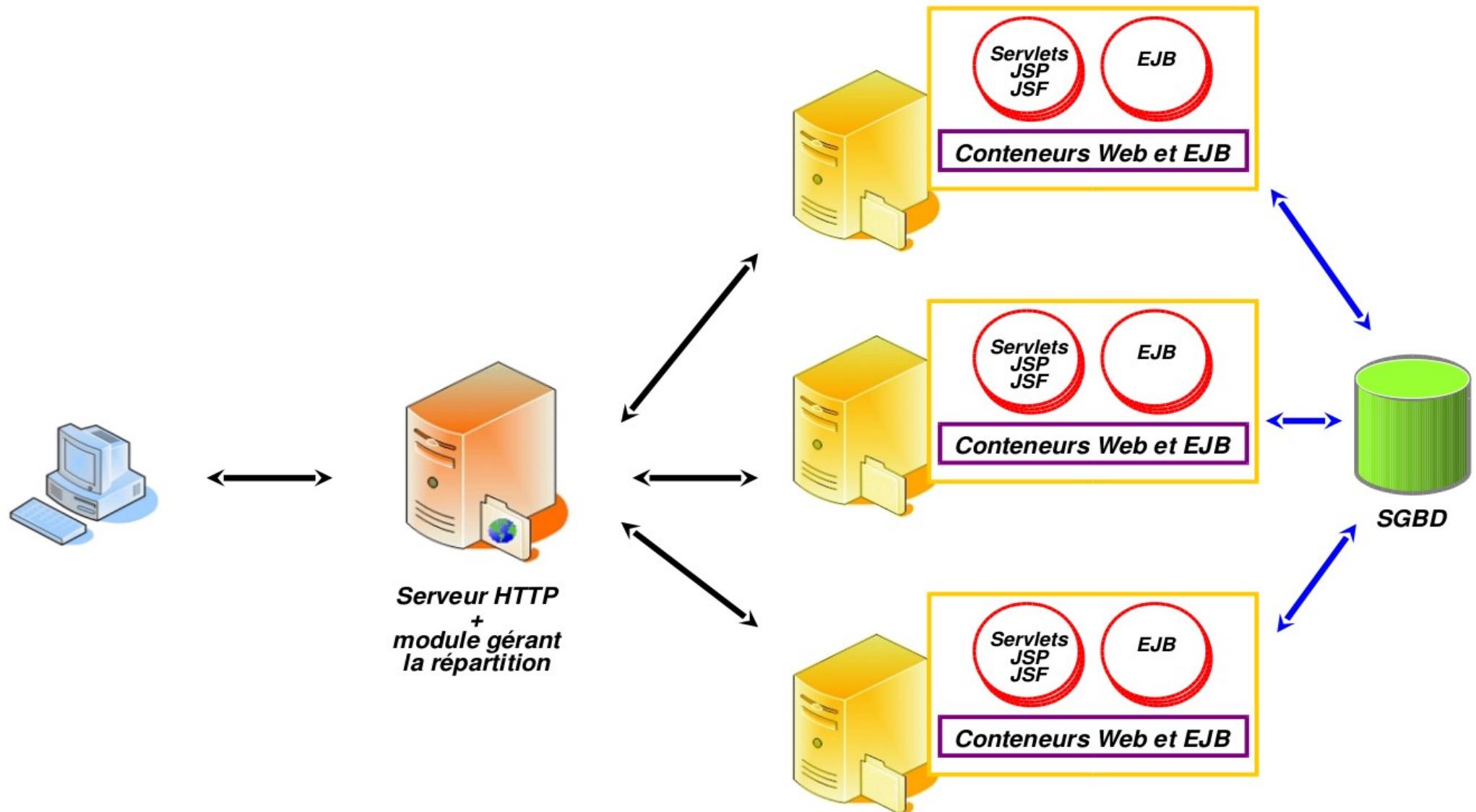


Architectures de déploiement (2)



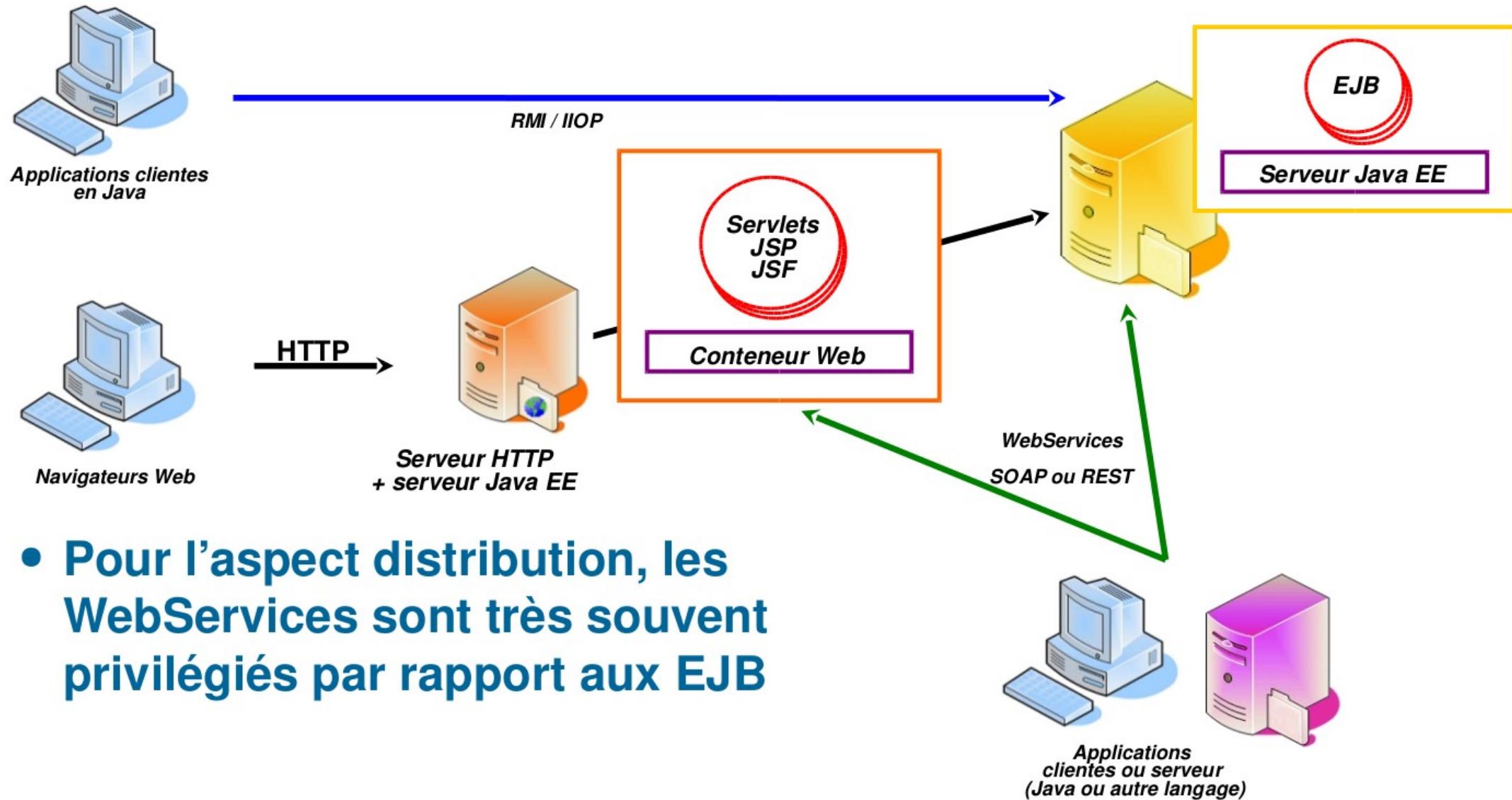
Architectures de déploiement (3)

- Répartition des requêtes HTTP par le serveur HTTP :



Architectures de déploiement (3)

- Les EJB sont accessibles de différentes façons :



Java EE : historique

- **J2EE 1.2 (1999) : Servlet 2.2, JSP 1.0, EJB 1.1**
- **J2EE 1.3 (2001) : Ajout de J2EE Connector Architecture**
- **J2EE 1.4 (2003) : Support des WebServices**
- **Java EE 5 (2006) :**
 - ▶ Servlet 2.5, JSP 2.1, EJB 3.0, JPA 1.0
 - ▶ Inclusion des JSF. Support des WebServices renforcé
- **Java EE 6 (décembre 2009) :**
 - ▶ EJB 3.1, JSF 2.0, JPA 2.0
 - ▶ Support de l'approche REST pour les WebServices (spécification JAX-RS)
 - ▶ CDI (Context Dependency Injection) : reprend des concepts de Spring
 - ▶ utilisation d'annotations pour demander la création d'objets et leur injection dans des variables

Java EE 6 : la notion de profil

- **Java EE 6 introduit la notion de profil :**

- ▶ 'Full profile' : implémentation de toutes les spécifications Java EE 6
- ▶ 'Web profile' : implémentation d'un sous-ensemble des spécifications
- ▶ répond à la demande de disposer de conteneurs plus légers

API	Web Profile	Full Profile
Servlet 3.0	✓	✓
JSP 2.2	✓	✓
JSTL 1.2	✓	✓
EL 1.2	✓	✓
JSF 2.0	✓	✓
CDI 1.0	✓	✓
EJB 3.1 (Lite)	✓	✓
EJB 3.1 (Full)		✓
JPA 2.0	✓	✓
JTA 1.1	✓	✓
JMS 1.1		✓
JavaMail 1.4		✓
JAX-WS 2.2		✓
JAX-RS 1.1		✓
JAXB 2.2		✓
JACC 1.0		✓
JCA 1.6		✓

- **Jboss AS 7.0**

- ▶ Web Profile

- **Jboss AS 7.1**

- ▶ Full Profile

Présentation de JBoss Application Server

Présentation

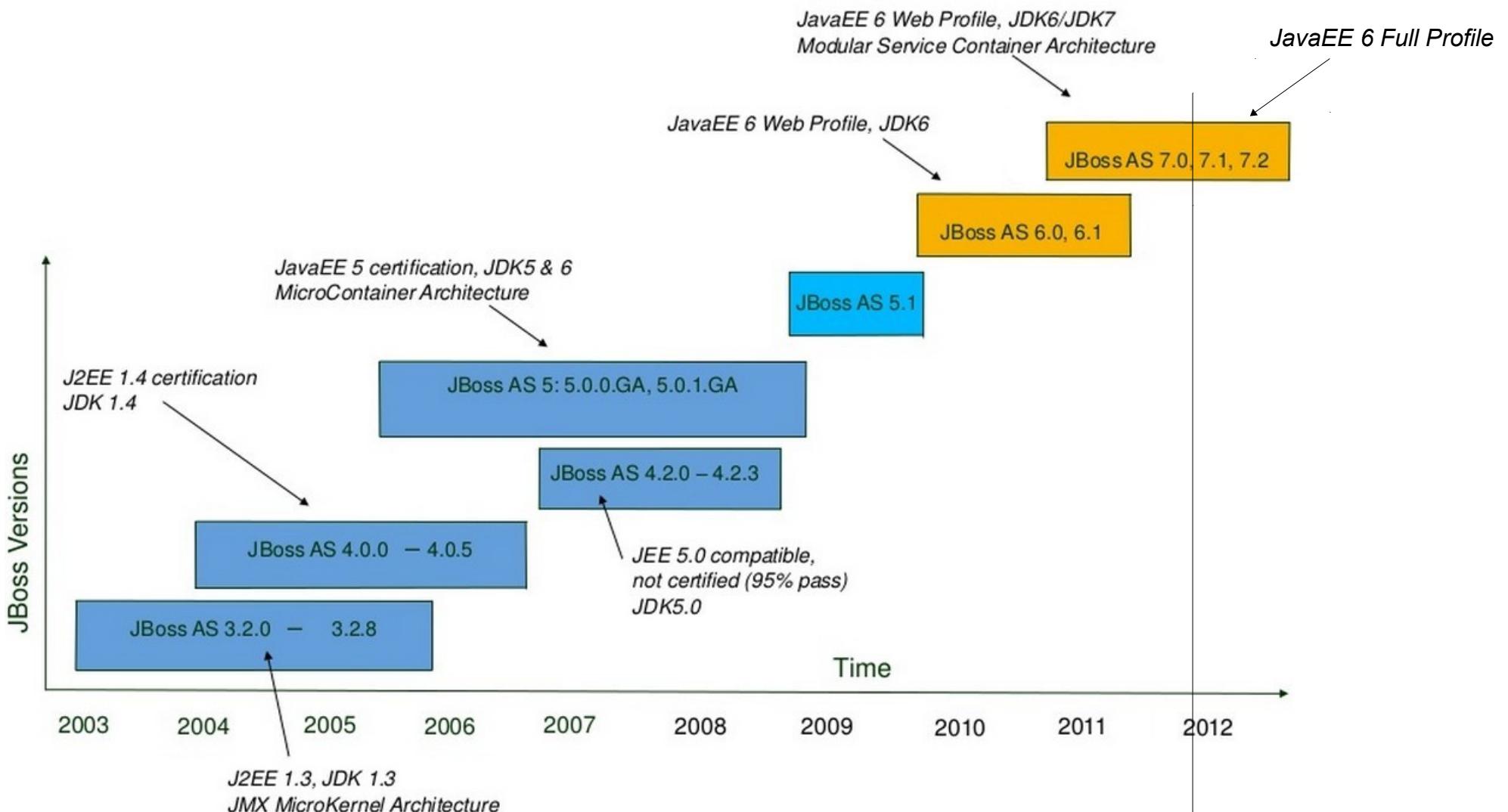
- **JBoss Application Server (JBoss AS)**

- ▶ Une implémentation open source de Java EE
- ▶ Écrit en Java : le seul prérequis pour le fonctionnement est l'installation d'un JRE

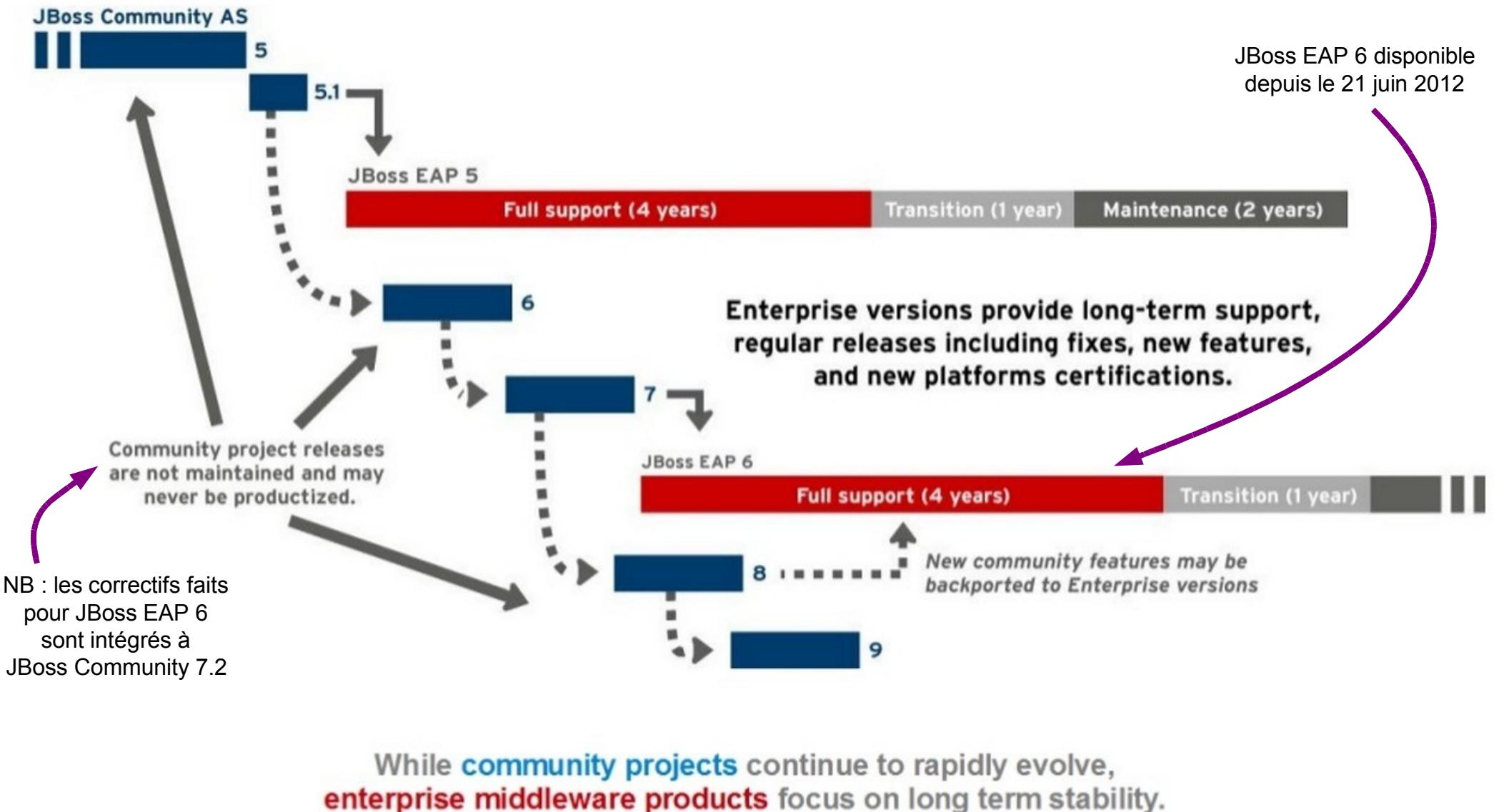
- **Son développement a commencé en 1999**

- ▶ Par un français : Marc Fleury, qui l'a cédé en Avril 2006 à Red Hat
- ▶ Depuis le développement se poursuit en open-source sous le nom JBoss Community AS
- ▶ RedHat propose une version avec support nommée JBoss EAP (Enterprise Application Platform)

Historique des versions de JBoss



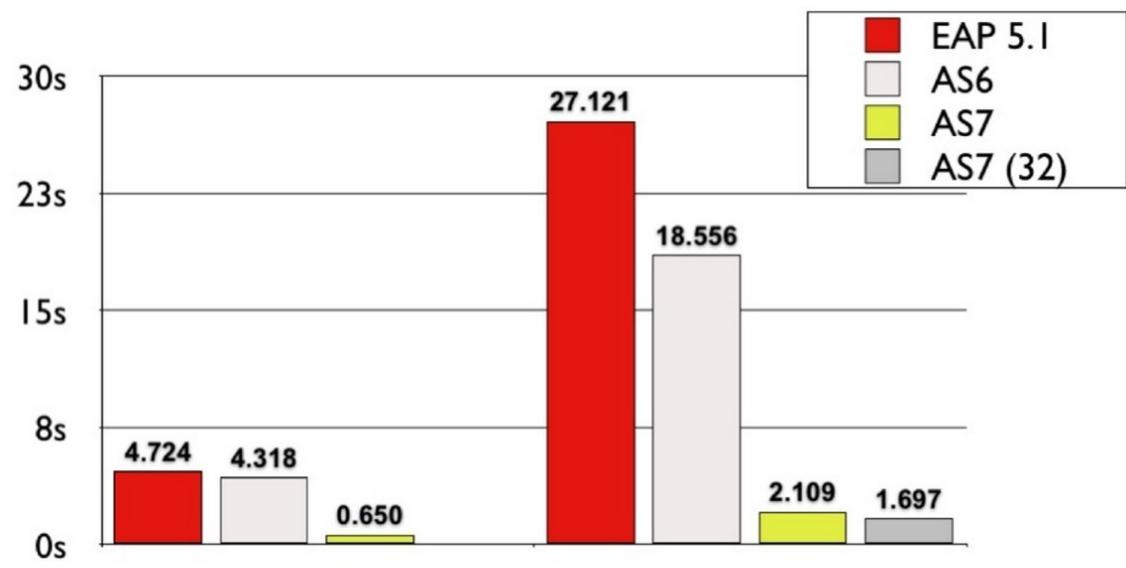
JBoss Community vs JBoss EAP



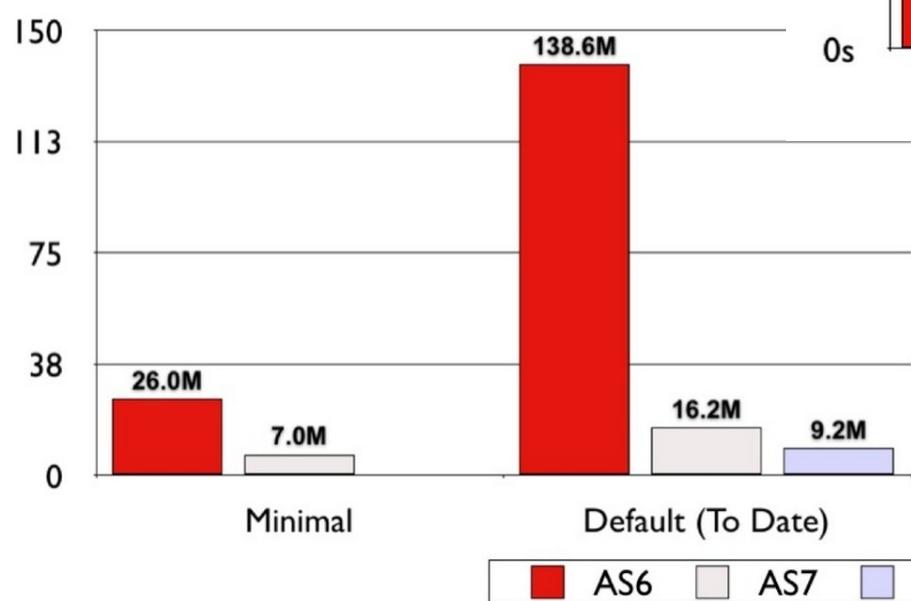
- **La version 7 est une refonte profonde de JBoss AS :** améliorations
 - ▶ des performances notables, notamment le temps de démarrage refonte du socle :
 - ▶ notions de modules et de services
 - ▶ fichiers de configuration simplifiés
 - ▶ possibilité d'administration centralisée
 - ▶ administration possible de plusieurs
 - ▶ façons :
 - ▶ console d'administration
 - ▶ ligne de commande
 - ▶ édition manuelle des fichiers
 - ▶ API Java et REST
 - ▶ support de Java EE 6 :
 - ▶ Web profile pour JBoss AS7.0
 - ▶ Full profile pour JBoss AS 7.1 et JBoss EAP 6

JBoss 7 : performances

Boot Time Results



Memory Comparison



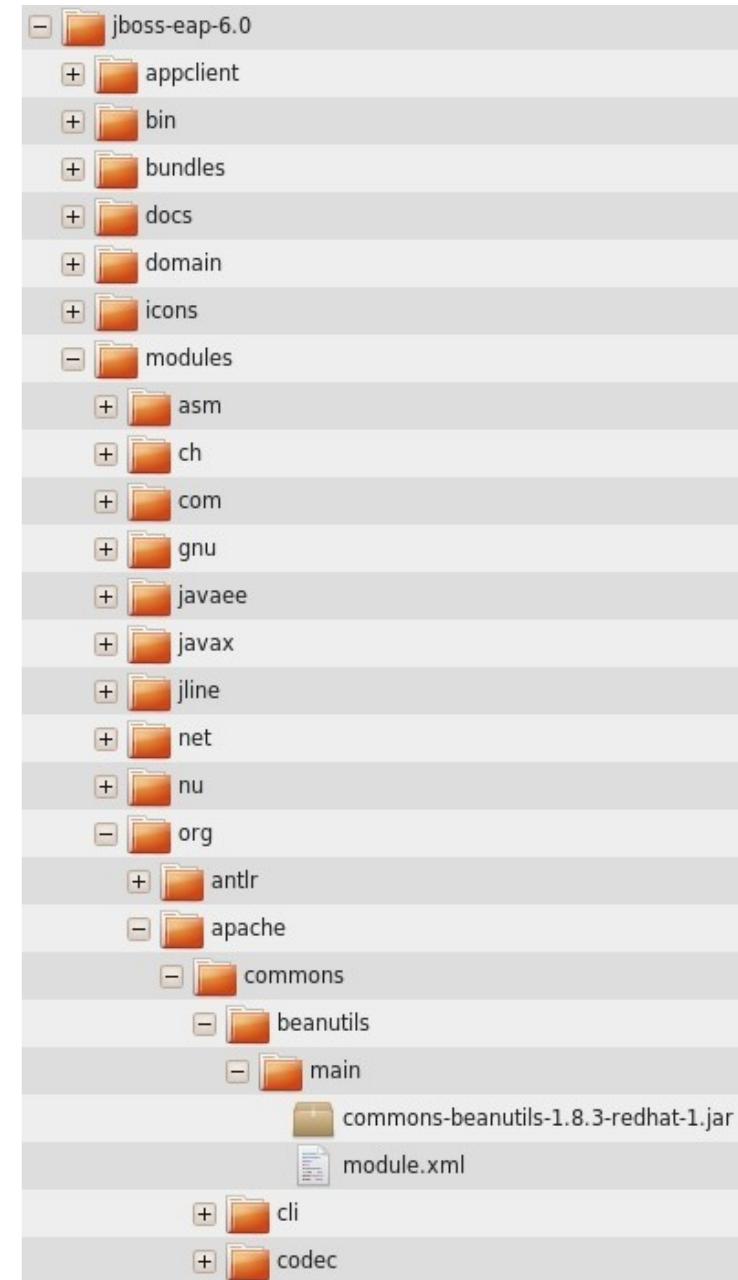
JBoss 7 : le socle

- **Une caractéristique importante de JBoss 7 est la réécriture du socle :**

- ▶ adoption d'une approche complètement modulaire
- ▶ basée sur le projet JBoss Modules
 - ▶ n'utilise pas OSGi mais est compatible avec OSGi

- **Intérêts de la notion de modules :**

- ▶ une arborescence contient l'ensemble des modules
- ▶ chaque module précise ses dépendances
- ▶ à l'exécution le noyau charge les modules dynamiquement selon les besoins
- ▶ par défaut les modules sont isolés les uns des autres
 - ▶ chacun dispose de son propre classloader



JBoss 7 : fichier de configuration simplifié

- La configuration d'un serveur est regroupée dans un seul fichier :

- ▶ fichier structuré pour être lisible
- ▶ modifiable manuellement mais ce mode n'est plus conseillé
- ▶ le fichier est édité par les outils d'administration (console et CLI)
 - ▶ il peut être réécrit à tout moment si le serveur est en exécution

The screenshot shows the JBoss Enterprise Application Platform 6.0 administration console. The left sidebar has sections like Profile, Connector, Container, Core, Infinispan, OSGI, Security, and Web. Under General Configuration, the 'Interfaces' section is selected. The main panel shows 'Network Interfaces' and 'Available Interfaces'. In the 'Available Interfaces' list, 'management' is selected. Below it are 'public' and 'unsecure'. At the bottom, there's a 'Sauvegarder' (Save) button. To the right, a large block of XML code represents the configuration file. A red arrow points from the 'Name' field in the interface configuration to the XML code for the 'management' interface.

```
<?xml version='1.0' encoding='UTF-8'?>

<server xmlns="urn:jboss:domain:1.3">

  <extensions>
    <extension module="org.jboss.as.clustering.infinispan"/>
    <extension module="org.jboss.as.configadmin"/>
  ...
  <subsystem xmlns="urn:jboss:domain:datasources:1.1">
    <datasources>
      <datasource jndi-name="java:jboss/datasources/ExampleDS">
        ...
      </datasource>
    </datasources>
  </subsystem>
</server>
```

```
<interfaces>
  <interface name="management">
    <any-address/>
  </interface>
</interfaces>
```

JBoss 7 : deux modes de fonctionnement

- **Depuis JBoss AS 7, nous avons le choix entre deux modes**

- ▶ Le mode autonome « standalone »
 - ▶ Démarre un processus indépendant, administré individuellement
 - ▶ Similaire aux versions précédentes de JBoss
- ▶ Le mode en domaine « domain »
 - ▶ Permet de démarrer plusieurs instances avec une seule commande
 - ▶ Permet d'administrer les instances sur plusieurs machines de façon centralisée

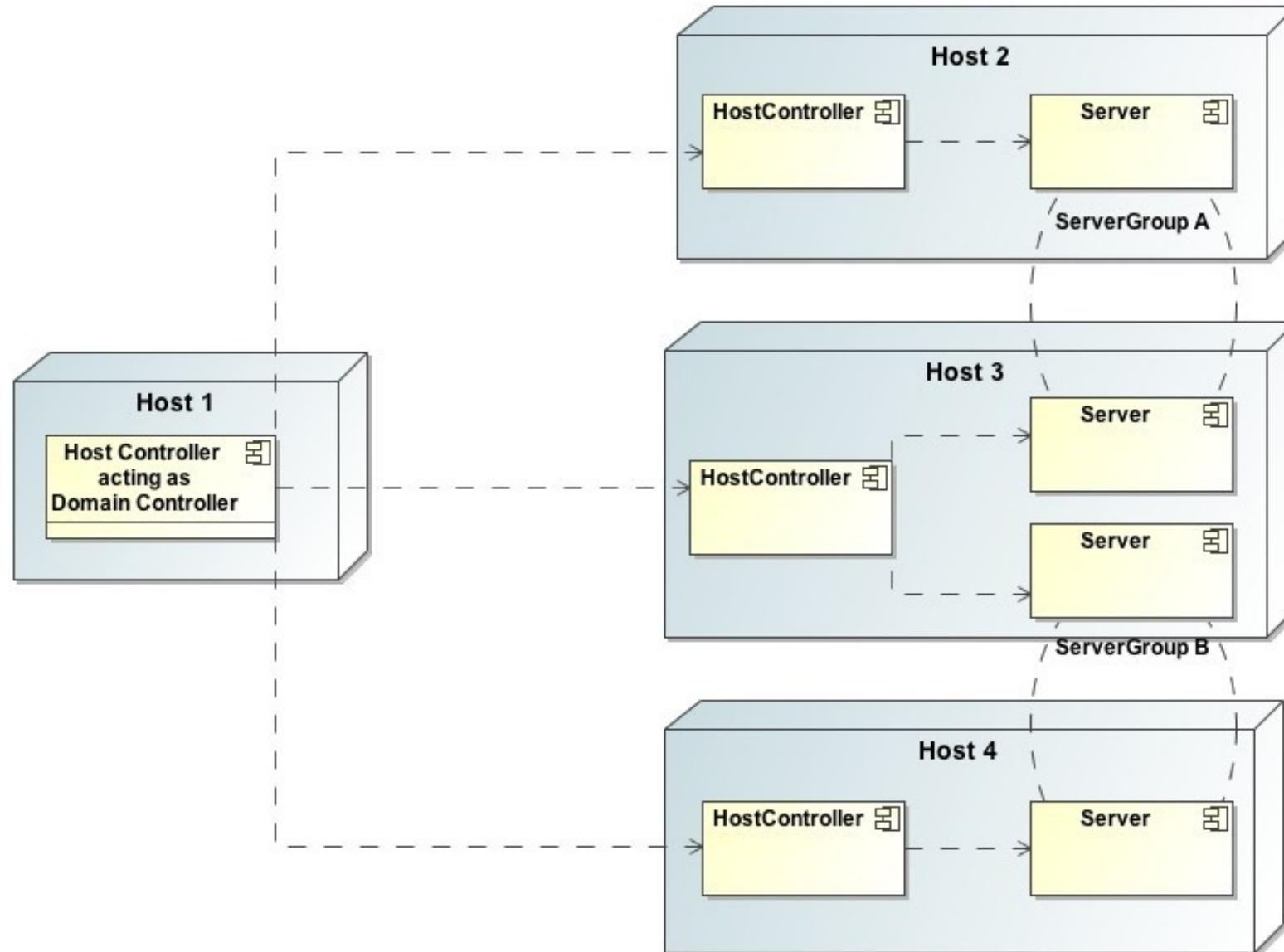
- **Le mode « domain » comporte 3 types d'acteurs**

- ▶ Des serveurs JBoss (host) sur lesquels sont déployées les applications
- ▶ Un host controller par machine, contrôlant les serveurs JBoss de cette machine
- ▶ Un domain controller qui pilote l'ensemble de l'environnement

- **Les deux modes sont indépendants du fonctionnement en cluster**

- ▶ On peut faire des cluster avec des serveurs standalone

JBoss 7 : administration centralisée

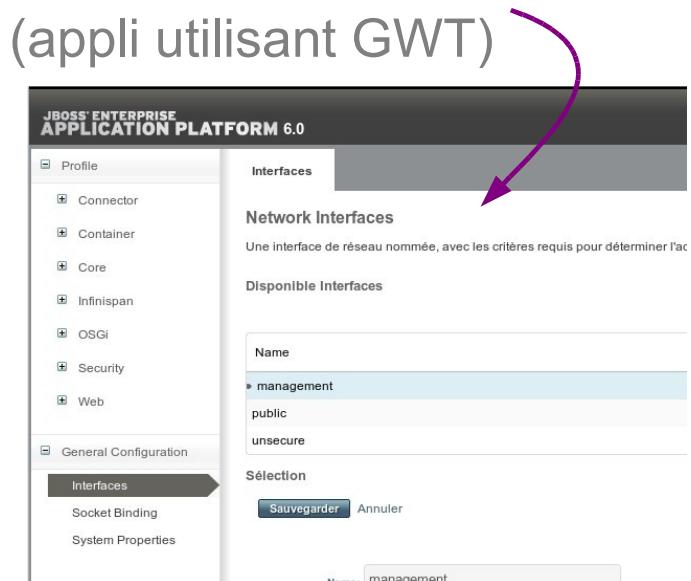


JBoss 7 : outils d'administration

- Les outils d'administration de JBoss ont été entièrement revus
- Plusieurs solutions :

- ▶ une console d'administration sous forme d'applicatif Web (appli utilisant GWT)
- ▶ un outil en ligne de commande le CLI

```
[standalone@localhost:9999 /] /subsystem=datasources:read-resource()
{
  "outcome" => "success",
  "result" => {
    "xa-data-source" =>
    undefined,
    "data-source" =>
    {"ExampleDS" =>
    undefined},
    ...
}
```



- ▶ une API

```
localhost:9990/management/subsystem/datasources/data-source?json.pretty
{
  "xa-data-source" : null,
  "data-source" : {"ExampleDS" : null},
  "jdbc-driver" : {"h2" : null}
}
```

- ▶ du code Java

- **RedHat propose plusieurs produits/solutions en complément du serveur d'applications notamment :**

- ▶ JBoss Developer Studio
 - ▶ environnement de développement basé sur Eclipse
 - ▶ disponible en open-source sous le nom JBoss Tools
- ▶ JBoss ON (Operations Network) :
 - ▶ outils de supervision / administration
 - ▶ complète la console de JBoss EAP (administration de plusieurs domaines, gestion des versions des livrables)
- ▶ OpenShift :
 - ▶ Plateforme Cloud
 - ▶ Basée sur RedHat Enterprise Linux
 - ▶ Mise en place rapide d'instances JBoss AS

Le mode standalone

Le mode standalone

- Installation d'un serveur JBoss
- Structures des répertoires
- Démarrage et arrêt d'un serveur JBoss
- Le fichier standalone.xml
- La console d'administration
- Gestion et configuration des logs

Installation JBoss Community

- La distribution de JBoss AS est un simple fichier compressé :
 - ▶ l'installation d'un JRE est pré-requise : Java SE 6 ou plus récent
- Pour télécharger la version community :
 - ▶ <http://www.jboss.org/jbossas/downloads/>



7.1

Name	Version	Description	Size	Release date	License	Release notes	Download
7.1.1.Final	Brontes	Certified Java EE 6 Full Profile	127MB	2012-03-09	LGPL	Release Notes ↗	ZIP ↗
		Certified Java EE 6 Full Profile	127MB	2012-03-09	LGPL	Release Notes ↗	TAR.GZ ↗
		Quickstarts		2012-03-09	AL 2		site ↗

Installation JBoss EAP 6 (1)

- **Les versions de JRE officiellement supportées pour EAP 6 sont :**

- ▶ Oracle JDK 1.6.x et 1.7.x IBM JDK
- ▶ 1.6.x et 1.7.x
- ▶ OpenJDK 1.6.x et 1.7.x

- **Installation du JRE sur Red Hat Enterprise Linux :**



Installation avec yum :

```
yum      install  java-1.6.0-openjdk-devel  
ou  
yum      install  java-1.7.0-openjdk-devel
```

- ▶ Si plusieurs JRE / JDK, utiliser l'outil *alternatives* pour choisir celui par défaut :

```
yum      install  alternatives  
/usr/sbin/alternatives --config  java
```

Installation JBoss EAP 6 (2)

- **JBoss EAP 6 est disponible sous plusieurs formes :**

- ▶ fichiers compressés (.zip ou .tar.gz), le format privilégié
- ▶ fichier JAR avec installateur intégré (mode graphique ou silencieux) fichier
- ▶ RPM via Red Hat Network

- **Utilisation de l'installateur :**

- ▶ mode graphique :

```
java -jar jboss-eap-6.0.0-installer.jar
```

- ▶ mode interactif :

```
java -jar jboss-eap-6.0.0-installer.jar -console
```

- ▶ mode silencieux :

```
java -jar jboss-eap-6.0.0-installer.jar fichierXMLInstall.xml
```

(un premier fichier XML d'installation peut être généré à partir des modes graphique ou console)

Installeur graphique (1)

• Déroulement de l'installeur :

The screenshot illustrates the graphical installer for JBoss EAP 6.0.0 GA, showing five sequential steps:

- Step 1: User Creation** (Yellow box)
A dialog titled "Créer un utilisateur" (Create a user) is shown. It contains a note: "Dans cette section, un nom d'utilisateur et un mot de passe sont définis. Cet utilisateur sera ajouté à ManagementRealm et pourra ainsi être utilisé pour accéder à la console de gestion, ainsi que toute autre application sécurisée à l'aide de ManagementRealm." Below this are three input fields:
 - Tapez le nom d'utilisateur admin :
 - Saisissez le mot de passe admin :
 - Saisissez à nouveau le mot de passe admin :A yellow box labeled "1" is positioned above the note.
- Step 2: Quickstart Installation** (Yellow box)
A dialog titled "Installation de Quickstarts" (Install Quickstarts) is shown. It contains a note: "Vous souhaitez installer Quickstarts ? Quickstarts sont des applications prêtes à être créées et déployées sur EAP6 rapidement et facilement." Below this are two radio buttons:
 - Oui
 - NonA yellow box labeled "2" is positioned above the note.
- Step 3: Java JDK Selection** (Yellow box)
A dialog titled "Sélectionnez le chemin d'installation" (Select the installation path) is shown. It contains a note: "Un JDK de la version 1.6 au minimum est requis afin de pouvoir utiliser toutes les fonctionnalités d'EAP 6." Below this is a text input field with the path "/usr/lib/jvm/java-6-sun/" and a "Naviguer" (Browse) button. A yellow box labeled "3" is positioned above the note.
- Step 4: Maven Repository Configuration** (Yellow box)
A dialog titled "Sélectionnez le chemin d'installation" (Select the installation path) is shown. It contains a note: "Spécifiez un référentiel maven ci-dessous. Ceci permettra à quickstarts d'être créé avec les mêmes artefacts qui ont été utilisés pour créer EAP 6. Cette option de configuration est très utile pour les développeurs, mais peut être ignorée dans un environnement de production. Le fichier settings.xml indiqué aura un nouveau profil ajouté nommé jboss-eap6 et qui possédera le référentiel donné configuré." Below this are two radio buttons:
 - Spécifiez un chemin vers un référentiel maven local existant contenant des artefacts Red Hat et un fichier settings.xml valide
 - Ignorer la personnalisation du référentiel mavenA yellow box labeled "4" is positioned above the note.
- Step 5: Maven Repository Configuration (Continuation)** (Yellow box)
This is a continuation of Step 4, showing the same configuration dialog with the second radio button selected. A yellow box labeled "5" is positioned above the note.

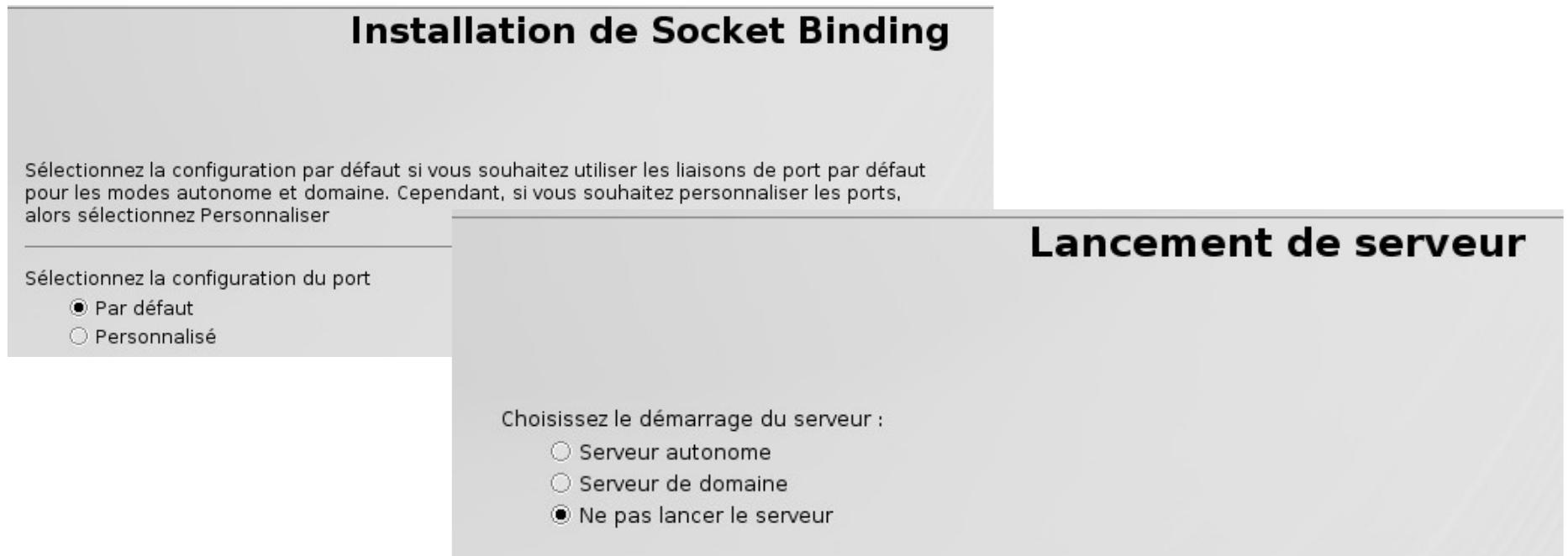
Installeur graphique (2)

- Installation totale ou partielle :

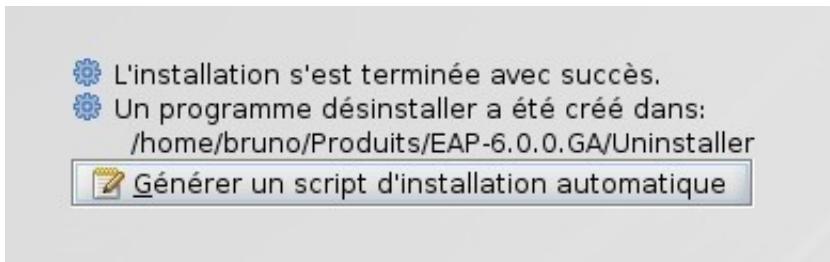


Installeur graphique (3)

- Dernières étapes :



- Après l'installation :



Arborescence

- **Contenu de <JBoss_HOME> :**

- ▶ appclient : fichier de configuration pour le conteneur d'application cliente
- ▶ bin : répertoire contenant les commandes principales (démarrage, ajout d'un utilisateur, lancement du CLI, ...)
- ▶ bundles : contient les modules OSGi
- ▶ docs : schémas XML utilisés par les fichiers de configuration
- ▶ domain : fichiers de configuration pour la mise en œuvre d'un domaine JBoss
- ▶ modules : contient les modules JBoss
- ▶ standalone : fichiers de configuration pour la mise en œuvre d'un serveur autonome
- ▶ jboss-modules.jar : JAR de 'bootstrap'



Démarrer le serveur

- Pour démarrer un serveur standalone :

```
<JBOSS_HOME>/bin/standalone.sh
```

- ▶ Si le démarrage se passe bien la console affiche le message suivant :

```
21:06:10,827 INFO [org.jboss.as] (Controller Boot Thread) JBAS015874:  
JBoss EAP 6.0.0.GA (AS 7.1.2.Final-redhat-1) a démarré en 1515ms - 134 sur 214 services ont démarré  
(79 services sont passifs ou à la demande)
```

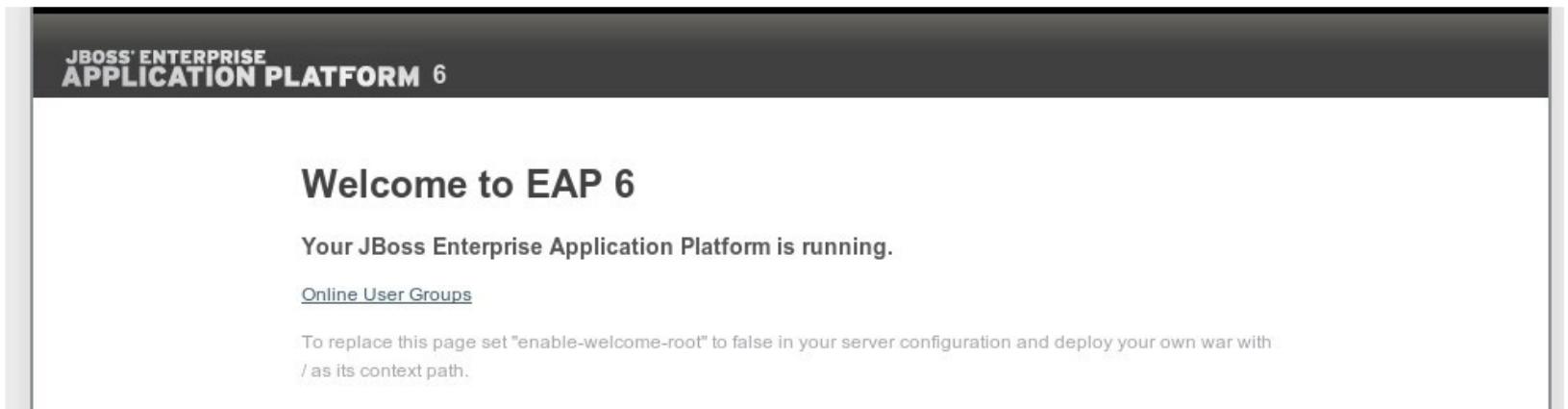
- ▶ Vérifier la présence du processus :

```
bruno@BLC4-Ubuntu:~$ ps -ef | grep java  
bruno 16776 16754 0 21:06 pts/0 00:00:05 java -D[Standalone] -server  
-Xms1303m -Xmx1303m -XX:MaxPermSize=256m -Djava.net.preferIPv4Stack=true  
-Dorg.jboss.resolver.warning=true -Dsun.rmi.dgc.client.gcInterval=3600000  
-Dsun.rmi.dgc.server.gcInterval=3600000  
-Djboss.modules.system.pkgs=org.jboss.byteman -Djava.awt.headless=true  
-Djboss.server.default.config=standalone.xml
```

Tester le serveur

- Vérifier que le serveur accepte les requêtes HTTP :

- ▶ <http://localhost:8080>



Welcome to AS 7

Your JBoss Application Server 7 is running.

However you have not yet added any users to be able to access the admin console.

To add a new user execute the add-user.sh script within the bin folder of your AS 7 installation and enter the requested information.

By default the realm name used by AS 7 is "ManagementRealm" this is already selected by default.



TP
1.1

/

Démarrage en service (Linux)

- Le démarrage peut se faire via un service :

- ▶ copier le fichier contenu du répertoire JBOSS_HOME/bin/init.d dans /etc/init.d

```
sudo cp jboss-as-standalone.sh jboss-as.conf /etc/init.d
```

- ▶ déclarer le service :

```
sudo chkconfig --add jboss-as-standalone.sh
```

RedHat

```
sudo update-rc.d jboss-as-standalone.sh defaults
```

Debian

- ▶ éditer les fichiers :

- ▶ jboss-as.conf pour renseigner la variable JBOSS_HOME

- ▶ jboss-as-standalone.sh pour renseigner les variables :

- ▶ JBOSS_HOME

- ▶ JBOSS_CONFIG : le nom du fichier de configuration, par exemple domain.xml dans le cas d'utilisation du mode domaine

- ▶ JBOSS_SCRIPT : le nom du script de lancement, par défaut standalone.sh, utiliser domain.sh pour le mode domaine

- ▶ démarrer le service :

```
sudo service jboss-as-standalone.sh start
```

Démarrage en service (Windows)

- **Sous Windows, c'est un peu plus compliqué**

- ▶ Il faut télécharger des connecteurs natifs
- ▶ <http://www.jboss.org/jbossweb/downloads/jboss-native-2-0-10>
- ▶ Décompresser ce qui a été téléchargé dans le dossier d'installation de JBoss
- ▶ AS Dans service.bat, modifier les lignes 84 et 94 et les remplacer par

```
call jboss-cli.bat -connect command=:shutdown >> shutdown.log 2>&1
```

- ▶ Si plusieurs serveurs sont utilisés sur la même machine Il faut configurer ls script CLI pour se connecter au bon serveur En ajoutant –controller=host:mport à l'appel du

- ▶ Installer le service

```
service install
```

- ▶ Les informations complètes sont disponibles sur le Wiki
 - ▶ <https://community.jboss.org/wiki/RunningJBossAS7AsAWindowsService>

Arrêter le serveur

- **Plusieurs solutions pour un arrêt propre :**

- ▶ arrêt du service
- ▶ Ctrl+C dans la console
- ▶ kill sur le process

- **Exemple d'affichage après arrêt :**

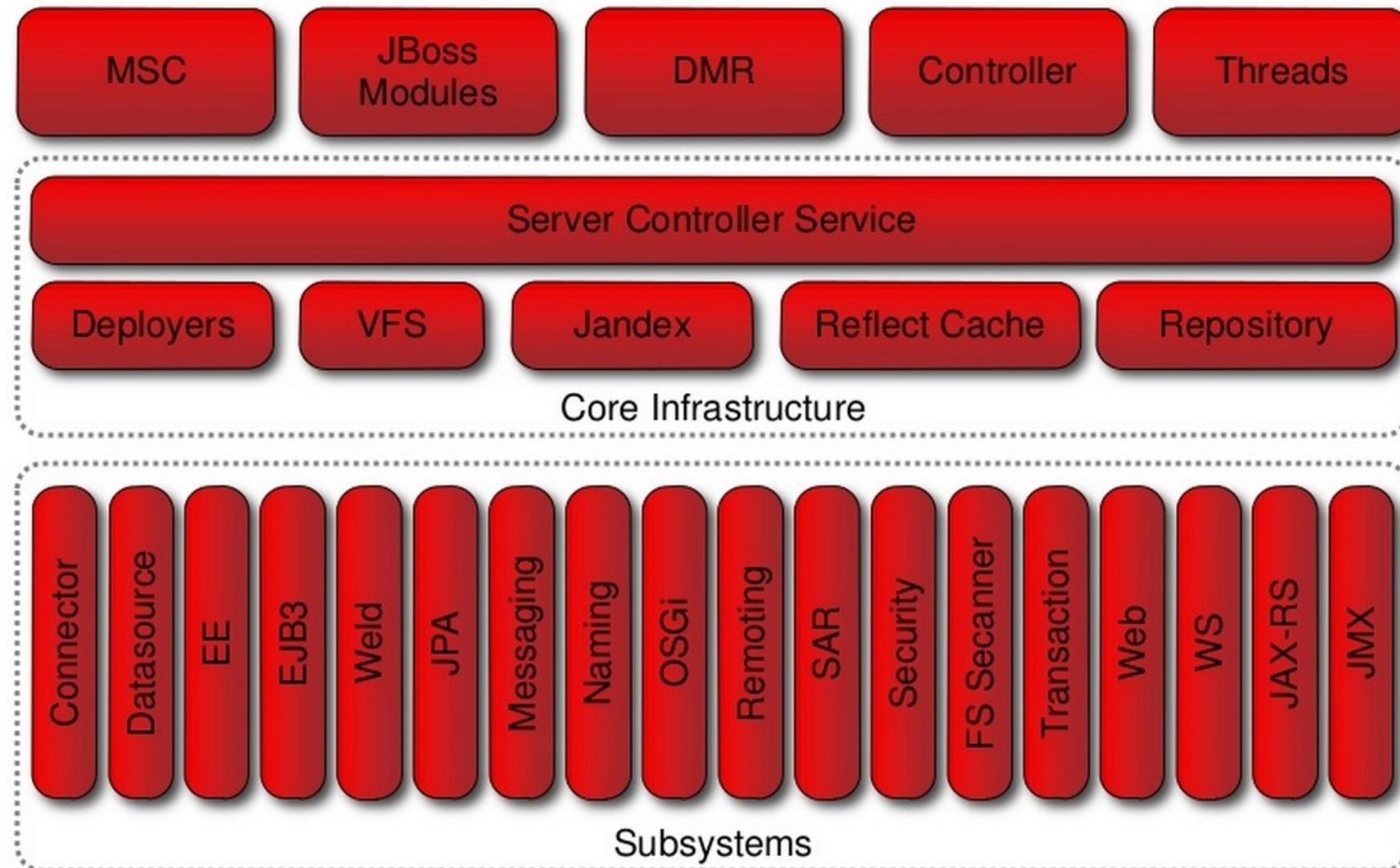
```
22:13:57,195 INFO [com.arjuna.ats.jbossatx] ARJUNA032014:  
Stopping transaction recovery manager  
22:13:57,204 INFO [org.jboss.as] JBAS015950: JBoss EAP  
6.0.0.GA (AS 7.1.2.Final-redhat-1) arrêté en 10ms
```

Le fichier standalone.xml

- La configuration d'un serveur autonome se trouve dans le fichier **<JBOSS_HOME>/standalone/configuration/standalone.xml** :

- ▶ édition manuelle possible mais non conseillée le fichier contient plusieurs grandes sections :
 - ▶ <extension> : liste de modules enrichissant le socle du serveur
 - ▶ <path> : définition de variables, pointant vers des répertoires, utilisables dans la suite du fichier
 - ▶ <management> : divers paramètres contrôlant l'accès aux outils d'administration du serveur
 - ▶ <profiles> : section la plus importante
 - ▶ *un profil définit un ensemble de sous-systèmes.*
 - ▶ *chaque sous-système active une fonctionnalité du serveur (ex : JSF, JPA, JAX-RS, ...) des attributs de configuration peuvent être déclarés (ex : une datasource, ...)*
 - ▶ <interfaces> : liste des interfaces réseau pour accéder au serveur
 - ▶ <socket-binding-group> : définition des numéros de ports
 - ▶ <system-properties> : variables d'environnements passées à la JVM
 - ▶ <deployments> : définition des applications déployées

Liste des sous-systèmes



Source : <http://fr.slideshare.net/dandreadis/jboss-as7-reloaded> (présentation par le responsable des développements de JBoss AS)

Interfaces

- La notion d'interface permet de configurer précisément les adresses IP du serveur qui seront exposées
- Par défaut, 3 interfaces sont définies :
 - ▶ 'management' pour les fonctionnalités d'administration (console et CLI) 'public'
 - ▶ pour les accès HTTP aux applications
 - ▶ 'secure' pour les appels IIOP
- Modifier les définitions pour autoriser les accès à distance :

```
<interfaces>
    <interface name="management">
        <inet-address value="${jboss.bind.address.management:192.168.0.28}" />
    </interface>
    <interface name="public">
        <any-address />
    </interface>
    <interface name="unsecure">
        <inet-address value="${jboss.bind.address.unsecure:127.0.0.1}" />
    </interface>
</interfaces>
```

Exposer la console sur l'adresse 192.168.0.28

Exposer le port HTTP sur toutes les adresses réseaux du serveur

Options de lancement

- **La commande standalone.sh accepte plusieurs options :**

- ▶ -h : liste des autres options
- ▶ -b, -b= : adresse réseau à utiliser pour la définition l'interface 'public'
 - ▶ identique aux versions précédentes de JBoss
- ▶ -b<nom_interface>, -b<nom_Interface>= : adresse réseau pour une interface donnée
 - ▶ ex : -bmanagement=192.168.0.50 pour l'interface 'management'
- ▶ -c, -c=, --server-config= : nom du fichier de configuration à utiliser (standalone.xml par défaut)
 - ▶ Attention, chemin relatif au répertoire de configuration
- ▶ -D : passage d'une propriété système à la JVM
- ▶ -P, -P=, --properties : lien vers un fichier properties dont les propriétés seront passés à la JVM
- ▶ -v, -V, --version : version du serveur



Accès à la console d'administration

- Par défaut l'accès aux fonctions d'administration est sécurisé :

- ▶ Nécessité de créer au moins un utilisateur administratif
- ▶ Utiliser la commande <JBOSS_HOME>/bin/add-user.sh ou add-user.bat

```
bruno@BLC4-Ubuntu:~/Produits/EAP-6.0.0.GA/jboss-eap-6.0/bin$ ./add-user.sh
Quel type d'utilisateur souhaitez-vous ajouter ?
  a) Management User (mgmt-users.properties)
  b) Application User (application-users.properties)
(a) :
Saisir les informations sur le nouvel utilisateur
Domaine (ManagementRealm) :
Nom d'utilisateur : admin1
Mot de passe :
Saisir mot de passe à
nouveau :
L'utilisateur 'admin1' va être ajouté pour le domaine 'ManagementRealm'
Est-ce correct oui/non ? y
Utilisateur 'admin1' ajouté au fichier '/home/bruno/Produits/JBoss-EAP-
6.0.0.GA/jboss-eap-6.0/standalone/configuration/mgmt-users.properties'
Utilisateur 'admin1' ajouté au fichier '/home/bruno/Produits/JBoss-EAP-
6.0.0.GA/jboss-eap-6.0/domain/configuration/mgmt-users.properties'
```

La console d'administration

`http://localhost:9990/console`

2 onglets :

- Runtime : vue sur le contexte d'exécution
- Profile : vue sur le fichier de configuration

The screenshot shows the JBoss Enterprise Application Platform 6.0 administration console. The URL `http://localhost:9990/console` is displayed at the top left. At the top right, there are two tabs: **(0) Messages**, **Profile**, and **Runtime**. A purple arrow points from the text "2 onglets :" to the **Runtime** tab. The main content area is titled **Standalone Server** and **Server: blc4-ubuntu**. It displays the server configuration status: "The server configuration is up to date!". Below this, there are fields for **Code Name:** *Steropes*, **Release version:** *7.1.2.Final-redhat-1*, and **Server State:** *running*. At the bottom, there are tabs for **Extensions** and **Environment Properties**. The footer bar includes the text *1.3.1.Final-redhat-1*, **Settings**, and **Logout**.

Paramétrage de la console
En fait un seul paramètre : la langue

Settings

Locale: *fr*

Autoriser l'accès à distance

1 → Onglet 'Profile'

2 → Section 'Interfaces'

3 → Choix d'une interface :
-management pour les services d'administration (console, CLI)
- public pour les applications

4 et 6→ Cliquer sur 'Edit' pour passer en édition puis sur 'Save' pour valider les modifications

5 → Indiquer une adresse ou une 'wildcard'

7 → Comme indiqué le serveur doit être redémarré

The screenshot shows the JBoss Enterprise Application Platform 6.0 interface. The title bar reads "JBoss' ENTERPRISE APPLICATION PLATFORM 6.0". The top right corner shows a message "A server reload is required!" and "(3) Messages". The navigation bar includes "Profile" (highlighted with a yellow box and the number 1) and "Runtime".

The main content area is titled "Network Interfaces" with the sub-section "Available Interfaces". A table lists network interfaces:

Name
management
public
unsecure

Below the table are sections for "Selection" and "Advanced". The "Selection" section contains "Save" and "Cancel" buttons, and fields for "Name" (set to "management"), "Inet Address", "Nic", "Loopback", and "Address Wildcard" (set to "Any Address"). The "Advanced" section contains "Nic Match" and "Loopback Address".

Yellow numbered callouts point to specific elements:

- 1: Profile tab in the navigation bar.
- 2: Interfaces tab in the left sidebar.
- 3: "management" interface entry in the list.
- 4: "Save" button in the modal dialog.
- 5: "Address Wildcard" dropdown in the "Selection" section.
- 6: "management" entry in the "Name" field of the modal dialog.
- 7: Message "A server reload is required!" in the top right corner.

In the bottom right corner, there is a small chalkboard with the text "TP 1.3 /".

Configuration des ports

The screenshot shows two pages of the JBoss Enterprise Application Platform 6.0 configuration interface. A purple arrow points from step 3 on the top page to step 4 on the bottom page.

Top Page (Step 1): Shows the 'Socket Bindings' tab selected in the 'Profile' section. A yellow box labeled '1' is at the top right. The left sidebar has 'Socket Binding' highlighted with a yellow box labeled '2'.

Bottom Page (Step 4): Shows the 'Socket Bindings: Group standard-sockets' page. The 'Inbound' tab is selected. A yellow box labeled '3' is on the 'View →' button next to the 'standard-sockets' group name. The left sidebar has 'Socket Binding' highlighted with a yellow box labeled '2'. A purple arrow points from the 'View →' button to the 'Edit' button at the bottom of the table on the bottom page.

Table Data (Bottom Page):

Name	Port	MCast Port
ajp	8009	
http	8080	
https	8443	
management-http	<code>\$(jboss.management.http.port:9990)</code>	
management-https	<code>\$(jboss.management.https.port:9443)</code>	
management-native	<code>\$(jboss.management.native.port:9999)</code>	

Variables et propriétés système

- Possibilité d'utiliser des variables dans le fichier de configuration :

- ▶ variables passées sous forme de propriétés à la JVM pour
- ▶ utiliser la variable dans le fichier de configuration :

```
${nomVariable:valeurParDéfaut}
```

- ▶ si la variable est définie sa valeur sera utilisée, sinon c'est la valeur par défaut qui est prise en compte
- ▶ Exemples :

- ▶ adresse IP de l'interface 'public' (particularité la variable jboss.bind.address peut être initialisée avec l'option -b)

```
<interface name="public">  
  <inet-address value="${jboss.bind.address:127.0.0.1}"/>
```

- ▶ décalage des numéros de ports (pratique pour lancer plusieurs serveurs à partir du même fichier de configuration, ./standalone.sh -Djboss.socket.binding.port-offset=100)

```
<socket-binding-group name="standard-sockets" default-interface="public"  
  port-offset="${jboss.socket.binding.port-offset:0}">
```

Paramétrages de la JVM

- **Adapter les paramètres mémoire de la JVM est un besoin classique :**

- ▶ -Xms : taille mémoire initiale
- ▶ -Xmx : taille mémoire maximale

- **Pour un serveur JBoss AS 7 en mode standalone :**

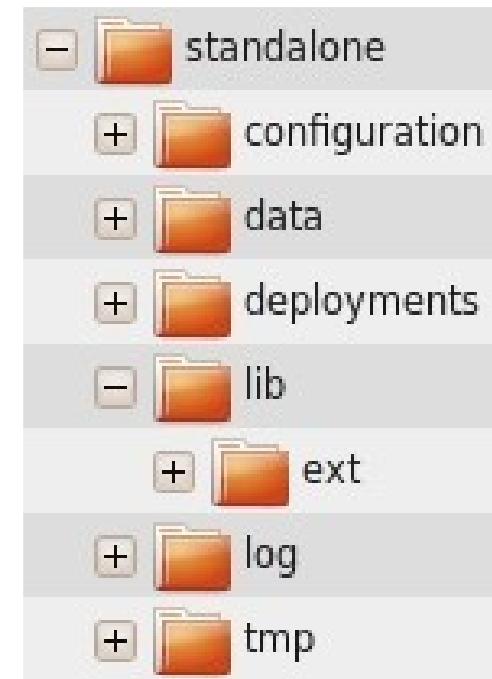
- ▶ ces paramètres ne sont pas dans le fichier standalone.xml car la JVM sera déjà lancée lors de la lecture du fichier
- ▶ modifier le fichier <JBoss_HOME>/bin/standalone.conf ou <JBoss_HOME>\bin\standalone.conf.bat

```
# Specify options to pass to the Java VM.  
#  
if [ "x$JAVA_OPTS" = "x" ]; then  
    JAVA_OPTS="-Xms1303m -Xmx1303m -XX:MaxPermSize=256m -Djava.net.preferIPv4Stack=true -Dorg.jboss.resolver.warning=true  
-Dsun.rmi.dgc.client.gcInterval=3600000 -Dsun.rmi.dgc.server.gcInterval=3600000"  
    JAVA_OPTS="$JAVA_OPTS -Djboss.modules.system.pkgs=$JBoss_MODULES_SYSTEM_PKGS -Djava.awt.headless=true"  
    JAVA_OPTS="$JAVA_OPTS -Djboss.server.default.config=standalone.xml"  
else  
fi echo "JAVA_OPTS already set in environment; overriding default settings with values: $JAVA_OPTS"
```

Arborescence du répertoire standalone

- **Le répertoire 'standalone' :**

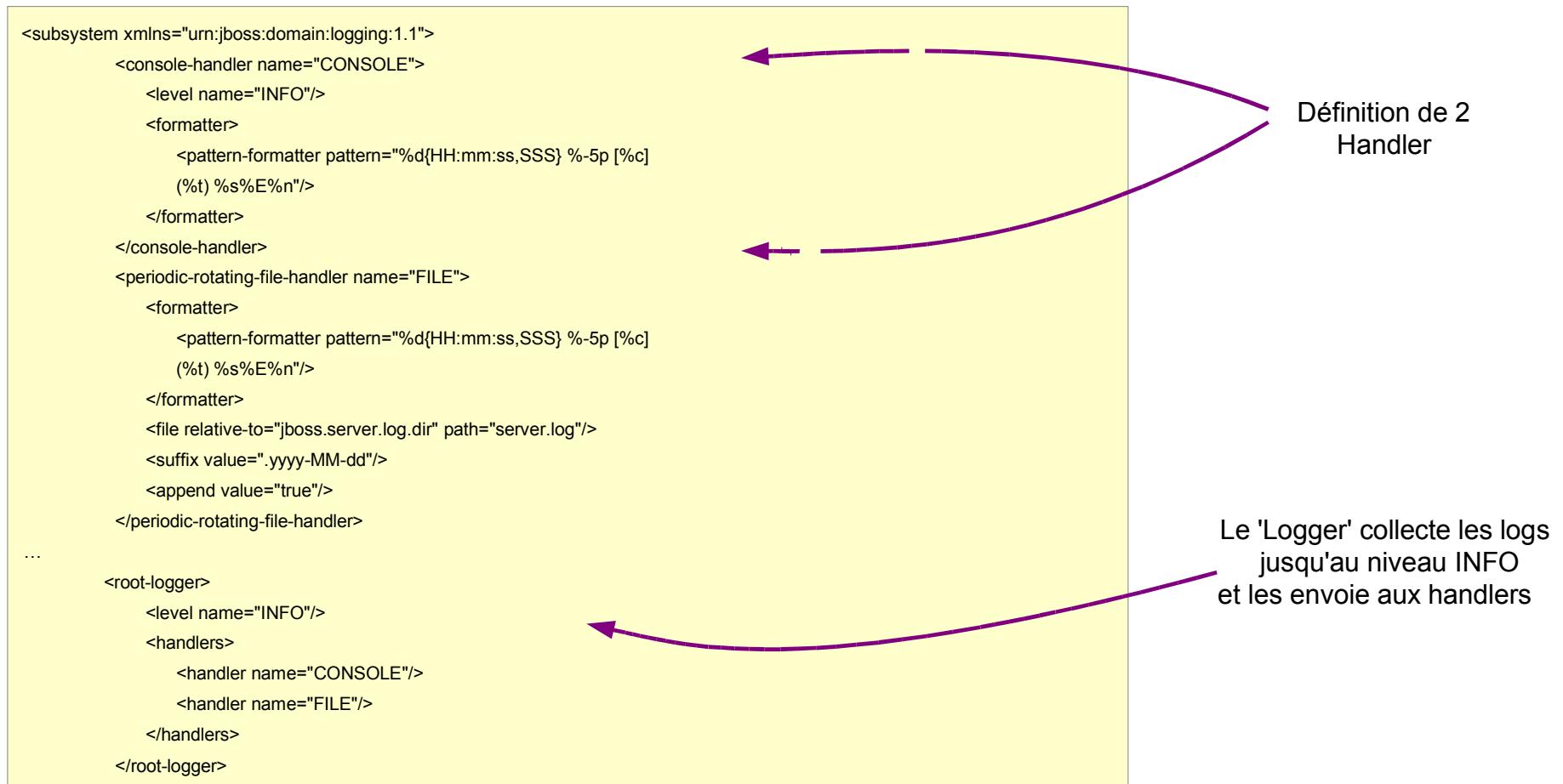
- ▶ configuration : emplacement par défaut des fichiers de configuration
- ▶ data : données internes
- ▶ deployments : répertoire contenant les applications. Par défaut, supporte le déploiement automatique des fichiers copiés dans le répertoire
- ▶ lib/ext : possibilité de placer des JARS qui seront chargés par le serveur. L'utilisation du répertoire <JBoss_HOME>/modules est préférable.
- ▶ log : fichiers de log



Gestion des logs (1)

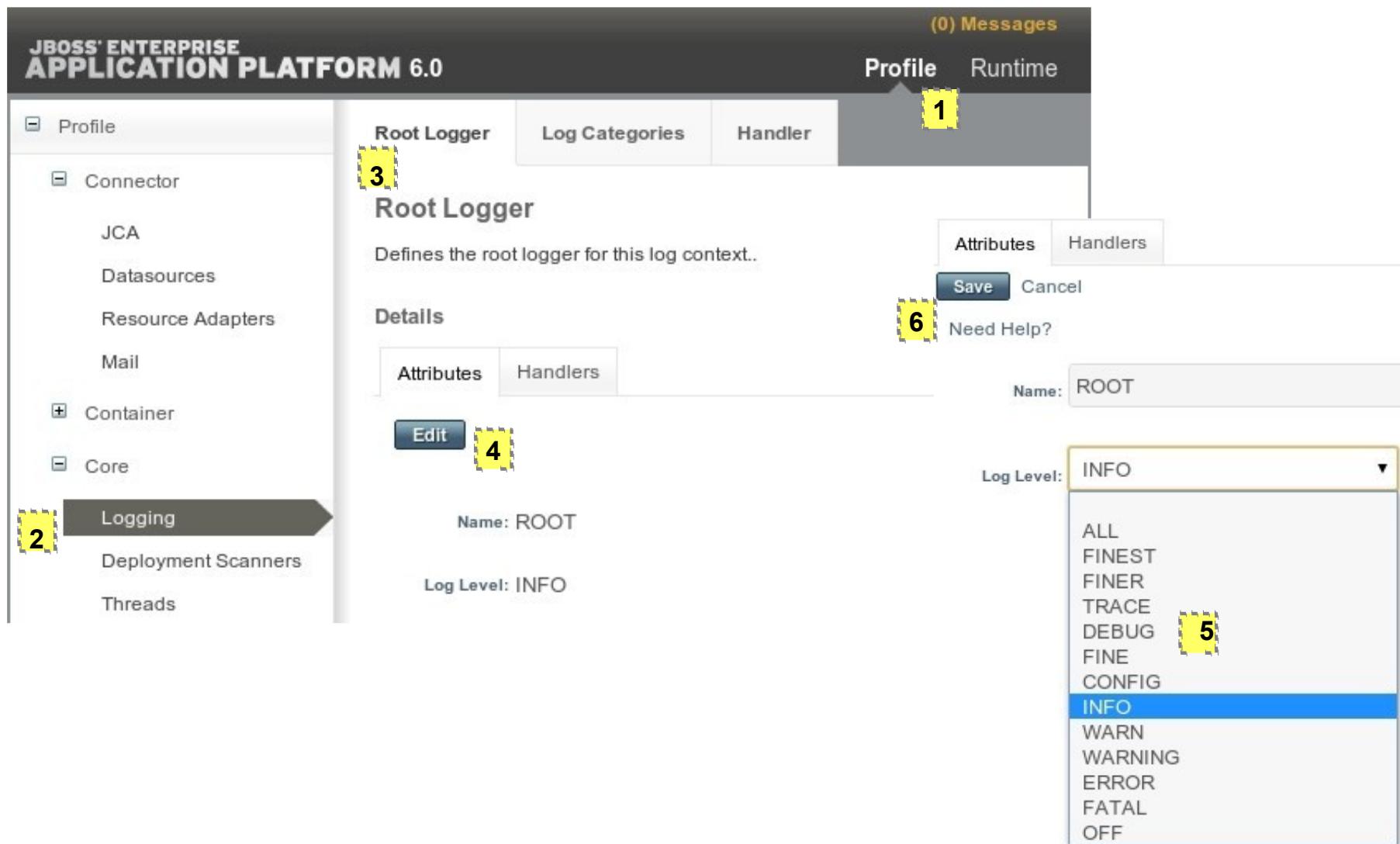
- **Un sous-système définit les règles de gestion des logs :**

- ▶ par défaut les messages de log vont dans la console et dans le fichier server.log
- ▶ server.log le fichier server.log est archive périodiquement



Gestion des logs (2)

- Configuration du niveau de log par la console :



Gestion des logs (3)

- Configuration des handlers :

The screenshot shows the JBoss Enterprise Application Platform 6.0 administration console under the 'Logging' section. The left sidebar is collapsed, and the main area shows the 'Handler' tab selected under 'Root Logger'. The 'Periodic' tab is active. A table lists a single handler named 'FILE'. The 'Edit' button for this handler is highlighted with a yellow box and labeled '5'. The 'Details' section shows the configuration for the 'FILE' handler:

Name	Log Level
FILE	

Details:
Name: FILE
Encoding:
Suffix: .yyyy-MM-dd
Formatter: %d{HH:mm:ss,SSS} %-5p [%c] (%t)
File Relative To: jboss.server.log.dir
File Path: server.log
Append: true



Administration centralisée

- Les possibilités du mode 'Domain' de JBoss 7
- Les principales notions : domain controller, host controller, process controller, server
- Mise en oeuvre d'un domaine
- Administration d'un domaine
- Les fichiers domain.xml et host.xml

- **JBoss 7 introduit la notion de domaine :**

- ▶ Ensemble d'instances de JBoss AS administré de façon centralisée
- ▶ Un domaine est constitué d'hôtes
- ▶ Sur chaque hôte s'exécutent un ou plusieurs **serveurs d'applications**
- ▶ Le '**Domain Controller**' est une instance spécialisée qui gère le référentiel de configuration :
 - ▶ le référentiel est défini dans le fichier domain.xml un seul 'Domain Controller' par domaine
- ▶ Sur chaque hôte, une instance spécialisée, nommée '**Host Controller**' communique avec le contrôleur de domaine
- ▶ Sur chaque hôte, le '**Process Controller**' est chargé du pilotage des serveurs d'applications

Mise en place d'un domaine

- **Pour créer un domaine :**

1 - Installer JBoss sur chaque machine

2 - Sur la machine qui jouera le rôle de Domain Controller :

- ▶ éditer le fichier host.xml pour autoriser les connexions distantes
- ▶ utiliser la commande add-user pour créer un utilisateur administratif pour chaque hôte distant

5 – Sur les autres machines éditer le fichier host.xml pour :

- ▶ donner un nom logique à l'hôte
- ▶ autoriser les connexions distantes
- ▶ renseigner les informations de connexion au contrôleur
- ▶ indiquer le mot de passe pour se connecter au contrôleur

7 – Définir les serveurs d'applications de chaque hôte :

- ▶ possible en éditant les fichiers domain.xml et host.xml
- ▶ à faire de préférence à travers à la console du Domain Controller ou la CLI une fois les contrôleurs démarrés

Domain Controller (1)

- **Editer le fichier host.xml du Domain Controller :**

- ▶ optionnel : modifier le nom logique d'hôte vérifier que le *Domain Controller* est indiqué comme étant local
- ▶ modifier la définition de l'interface

```
<host name="master" xmlns="urn:jboss:domain:1.3">
...
<domain-controller>
  <local/>
</domain-controller>

<interfaces>
  <interface name="management">
    <inet-address value="192.168.0.50"/>
  </interface>
</interfaces>
...
...
```

- **Lancer le Domain Controller :**

- ▶ <JBOSS_HOME>/bin/domain.sh

- **Deux processus sont lancés :**

- ▶ Le *Process Controller* chargé de démarrer/arrêter les autres
- ▶ JVM Un *Host Controller* qui jouera le rôle de *Domain Controller*

```
ps -ef | grep java
bruno      1882  1821  0 00:10 pts/1          00:00:00 java -D[Process Controller] ...
bruno      1899  1882  6 00:10 pts/1          00:00:04 java -D[Host Controller] ...
```

Domain Controller (2)

- **Créer des utilisateurs administratifs pour chaque hôte :**

- ▶ le nom d'utilisateur sera le nom logique donné à chaque hôte lors de la création, répondre 'yes' à la dernière question

```
bruno@BLC4-Ubuntu:~/Produits/EAP-6.0.0.GA/jboss-eap-6.0/bin$ ./add-user.sh
Quel type d'utilisateur souhaitez-vous ajouter ?
  a) Management User (mgmt-users.properties)
  b) Application User (application-users.properties)
(a): a

Saisir les informations sur le nouvel utilisateur
Domaine (ManagementRealm) :
Nom d'utilisateur : hote2 Mot de passe :
Saisir mot de passe à nouveau :
L'utilisateur 'hote2' va être ajouté pour le domaine 'ManagementRealm'
Est-ce correct oui/non ? y
Utilisateur 'hote2' ajouté au fichier '/home/bruno/Produits/JBoss-EAP-6.0.0.GA/jboss-eap-6.0/standalone/configuration/mgmt-users.properties'
Utilisateur 'hote2' ajouté au fichier '/home/bruno/Produits/JBoss-EAP-6.0.0.GA/jboss-eap-6.0/domain/configuration/mgmt-users.properties'
Is this new user going to be used for one AS process to connect to another AS
process e.g. slave domain controller?
yes/no? y
To represent the user add the following to the server-identities definition
<secret value="YWRtaW4=" />
```

Valeurs à recopier dans le fichier host.xml de l'hôte correspondant

Domain Controller (3)

- **Créer des utilisateurs administratifs pour chaque hôte :**
 - ▶ Attention, dans la version 7.1.1, la création des utilisateurs n'affiche pas le mot de passe encodé en Base64
 - ▶ On peut encoder manuellement le mot de passe saisi en base64 pour le mettre dans le fichier host.xml
 - ▶ `echo 'password' | base64`

Host Controller (1)

- Sur chaque hôte à rattacher au domaine :

- ▶ éditer le fichier host.xml modifier le nom d'hôte ajouter le mot de passe
- ▶ encodé en base64
- ▶ renseigner les informations de connexions au Domain Controller
- ▶ autoriser les connexions externes

```
<host name="host2" xmlns="urn:jboss:domain:1.3">
    <management>
        <security-realms>
            <security-realm name="ManagementRealm">
                <server-identities>
                    <!-- Replace this with either a base64 password of your own
                         or use a vault with a vault expression -->
                    <secret value="YWRtaW4=" />
                </server-identities>
                ...
                <authentication>
                    ...
                </authentication>
                <domain-controller>
                    <remote host="192.168.0.50" port="${jboss.domain.master.port:9999}"
                           security-realm="ManagementRealm"/>
                </domain-controller>
            </security-realm>
        </security-realms>
    </management>
    <interfaces>
        <interface name="management">
            <inet-address value="192.168.0.49"/>
        </interface>
    </interfaces>
</host>
```

The diagram shows a yellow rectangular box containing an XML configuration snippet for a JBoss host. Several purple arrows point from the list items on the left to specific parts of the XML code. The first arrow points to the opening tag of the host element. The second arrow points to the 'secret' element within the 'server-identities' section. The third arrow points to the 'remote' element within the 'domain-controller' section. The fourth arrow points to the 'inet-address' element within the 'interface' section.

Host Controller (2)

- Lancer le Host Controller :

- ▶ <JBoss_HOME>/bin/domain.sh

- Deux processus sont lancés :

- ▶ Le *Process Controller* chargé de démarrer/arrêter les autres JVM
- ▶ Un *Host Controller* qui va se connecter au Domain Controller notamment pour récupérer le fichier domain.xml

```
[Host Controller] 22:55:41,031 INFO [org.jboss.as.host.controller] (Controller Boot Thread) JBAS010927: Enregistré dans le contrôleur du domaine
[Host Controller] 22:55:41,095 INFO [org.jboss.as] (Controller Boot Thread)
JBAS015874: JBoss EAP 6.0.0.GA (AS 7.1.2.Final-redhat-1) (Host Controller) a démarré en 2636ms - 11 sur 11 services ont démarré (0 services sont passifs ou à la demande)
```

- Le Domain Controller reçoit la demande de connexion :

```
[Host Controller] 00:55:38,024 INFO [org.jboss.as.domain] (slave-request-threads - 1) JBAS010918: Hôte esclave distant enregistré "hote2", JBoss EAP 6.0.0.GA (AS 7.1.2.Final-redhat-1)
```

Options de lancement

- **Principales options de lancement de la commande domain.sh :**

- ▶ --domain-config=filename : utiliser un autre fichier que le fichier domain.xml
- ▶ --host-config=filename : utiliser un autre fichier que le fichier host.xml
- ▶ --backup : créer une copie locale de la configuration du domaine
- ▶ --cached-dc : si le Domain Controller n'est pas accessible, utiliser la configuration sauvegardée
 - ▶ ne fonctionne que si un lancement avec un --backup a été effectué
- ▶ --pc-address=<adresse> et --pc-port=<port> : pour configurer l'adresse et le port du Process Controller sans modifier le fichier host.xml
- ▶ --master-address=<adresse> et --master-port=<port> : idem pour le Domain Controller
- ▶ --interprocess-hc-address=<adresse> et --interprocess-hc-port=<port> : adresse et port pour la communication entre le Host Controller et le Process Controller
 - ▶ vérification des ports utilisés : netstat -natup | grep java



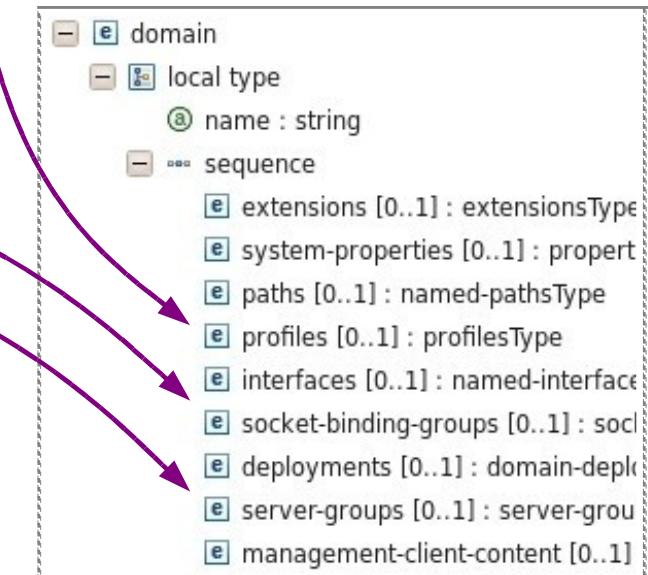
Configuration d'un domaine

- La configuration des serveurs d'un domaine est donc répartie entre les fichiers `domain.xml` et `host.xml` :

- ▶ le fichier `domain.xml` utilisé est celui du Domain
- ▶ Contrôler le fichier `host.xml` est spécifique à chaque hôte

- Principaux éléments du fichier `domain.xml` :

- ▶ des profils (un profil est un ensemble de sous-systèmes) des listes de ports
- ▶ des groupes de serveurs
 - ▶ un groupe de serveur est en ensemble de paramètres qui seront 'hérités' par les serveurs rattachés au groupe

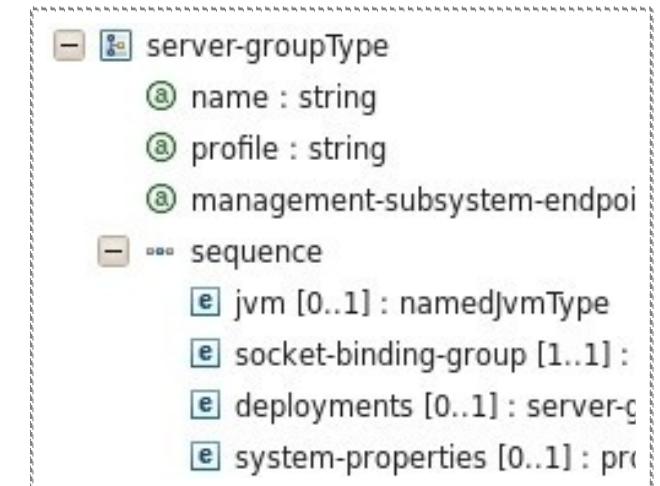


Configuration d'un groupe de serveurs

- Contenu d'un groupe de serveurs :

- ▶ des paramètres de JVM
- ▶ une liste de numéros de ports
- ▶ des applications déployées
- ▶ des propriétés système

```
<server-groups>
    <server-group name="main-server-group" profile="full">
        <jvm name="default">
            <heap size="1303m" max-size="1303m"/>
            <permgen max-size="256m"/>
        </jvm>
        <socket-binding-group ref="full-sockets"/>
    </server-group>
    <server-group name="other-server-group" profile="full-ha">
        <jvm name="default">
            <heap size="1303m" max-size="1303m"/>
            <permgen max-size="256m"/>
        </jvm>
        <socket-binding-group ref="full-ha-sockets"/>
    </server-group>
</server-groups>
```



Grammaire complète
d'un *server-group*

Création d'un groupe de serveurs (1)

- **Création d'un groupe de serveurs avec la console :**

- ▶ NB : création des serveurs dans un second temps

The screenshot shows the JBoss Enterprise Application Platform 6.0 interface for creating a server group. The host selected is 'hote2'. The 'Server Groups' tab is active. A table lists existing groups: 'main-server-group' (Profile: full) and 'other-server-group' (Profile: full-ha). A modal dialog titled 'Create Server Group' is open, showing fields for Name ('serverGroup1'), Profile ('default'), and Socket Binding ('standard-sockets').

Group Name	Profile
main-server-group	full
other-server-group	full-ha

Modal Dialog: Create Server Group

- 1. Server tab
- 2. Server Groups in the left sidebar
- 3. Add button in the Available Group Configurations table
- 4. Name field in the Create Server Group dialog
- 5. Profile dropdown in the Create Server Group dialog
- 6. Socket Binding dropdown in the Create Server Group dialog
- 7. Save button in the Create Server Group dialog

Création d'un groupe de serveurs (2)

- La définition du groupe peut être complétée :

- ▶ variables d'environnement
- ▶ paramètres de JVM

The screenshot shows the JBoss Enterprise Application Platform 6.0 management console. The left sidebar has a 'Server Groups' section selected. The main area displays 'Available Group Configurations' with three entries: 'main-server-group' (Profile: full), 'other-server-group' (Profile: full-ha), and 'serverGroup1' (Profile: default). A purple oval highlights the 'serverGroup1' row. Two arrows point from the text 'variables d'environnement' and 'paramètres de JVM' to the 'Attributes' and 'JVM Configuration' tabs at the bottom of the screen, respectively. The 'Attributes' tab is active, showing fields for 'Name' (jvm1), 'Heap Size' (256m), 'Max Heap Size' (1024m), 'Permgen Size' (empty), and 'Max Permgen Size' (empty). The 'JVM Configuration' tab is also visible.

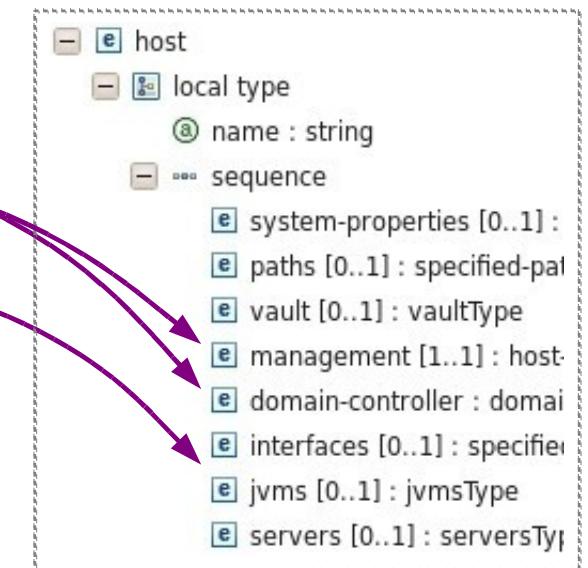
Configuration d'un hôte (1)

- Principaux éléments du fichier

- ▶ **host.xml** : les informations de connexion au Domain Controller éventuellement des définitions de JVM
- ▶ qui l'emportent sur celles définies dans le fichier domain.xml

```
<domain-controller>
    <remote host="192.168.0.50" port="${jboss.domain.master.port:9999}"
           security-realm="ManagementRealm"/>
...
<jvms>
    <jvm name="default">
        <heap size="64m" max-size="256m"/>
        <permgen size="256m" max-size="256m"/>
        <jvm-options>
            <option value="-server"/>
        </jvm-options>
    </jvm>
...
<servers>
    <server name="server-two" group="main-server-group" auto-start="true">
        <!-- server-two avoids port conflicts by incrementing the ports in
            the default socket-group declared in the server-group -->
        <socket-bindings port-offset="150"/>
    </server>
...

```



Grammaire complète
d'un *host*

Configuration d'un hôte (2)

- Configuration d'un hôte par la console :

- ▶ sélectionner l'hôte **2**
- ▶ éditer les définitions de JVM, d'interfaces et de propriétés système

The screenshot shows the JBoss Enterprise Application Platform 6.0 administration console. The top navigation bar includes 'JBoss ENTERPRISE APPLICATION PLATFORM 6.0', a message count '(10) Messages', and tabs for 'Profiles', 'Server' (which is selected), and 'Runtime'. The left sidebar has a 'Host' dropdown set to 'hote2' (marked with a yellow box labeled '2'), a 'Server' section with 'Server Configurations' and 'Server Groups', a 'Host Settings' section with 'JVM Configurations' (selected and highlighted with a yellow arrow and box labeled '3'), and 'Interfaces' and 'Host Properties' sections.

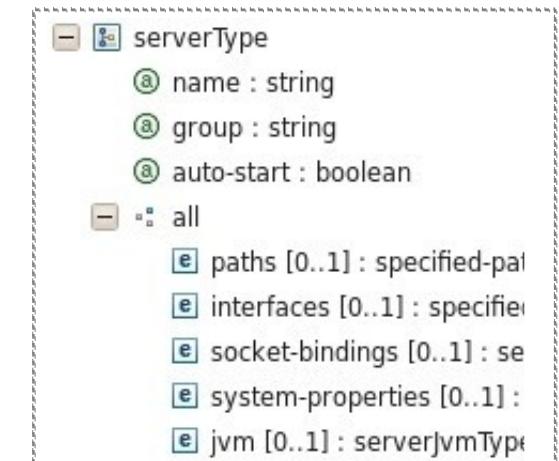
The main content area is titled 'Host JVM Configurations' and contains the following text: 'These JVM configuration are applicable to any server on a host. JVM configurations can be assigned to server configuration by name.' Below this is a table titled 'Available JVM Configurations' with two columns: 'Name' and 'Actions'. The table shows one entry: 'default'. At the bottom of the table are 'Add' and 'Remove' buttons. Navigation icons and a page number '1-1 of 1' are at the bottom right of the table.

Définition de serveurs (1)

- **Informations sur chaque serveur :**

- ▶ nom du serveur-group auquel est rattaché le
- ▶ serveur attribut de démarrage automatique
- ▶ paramètres de JVM

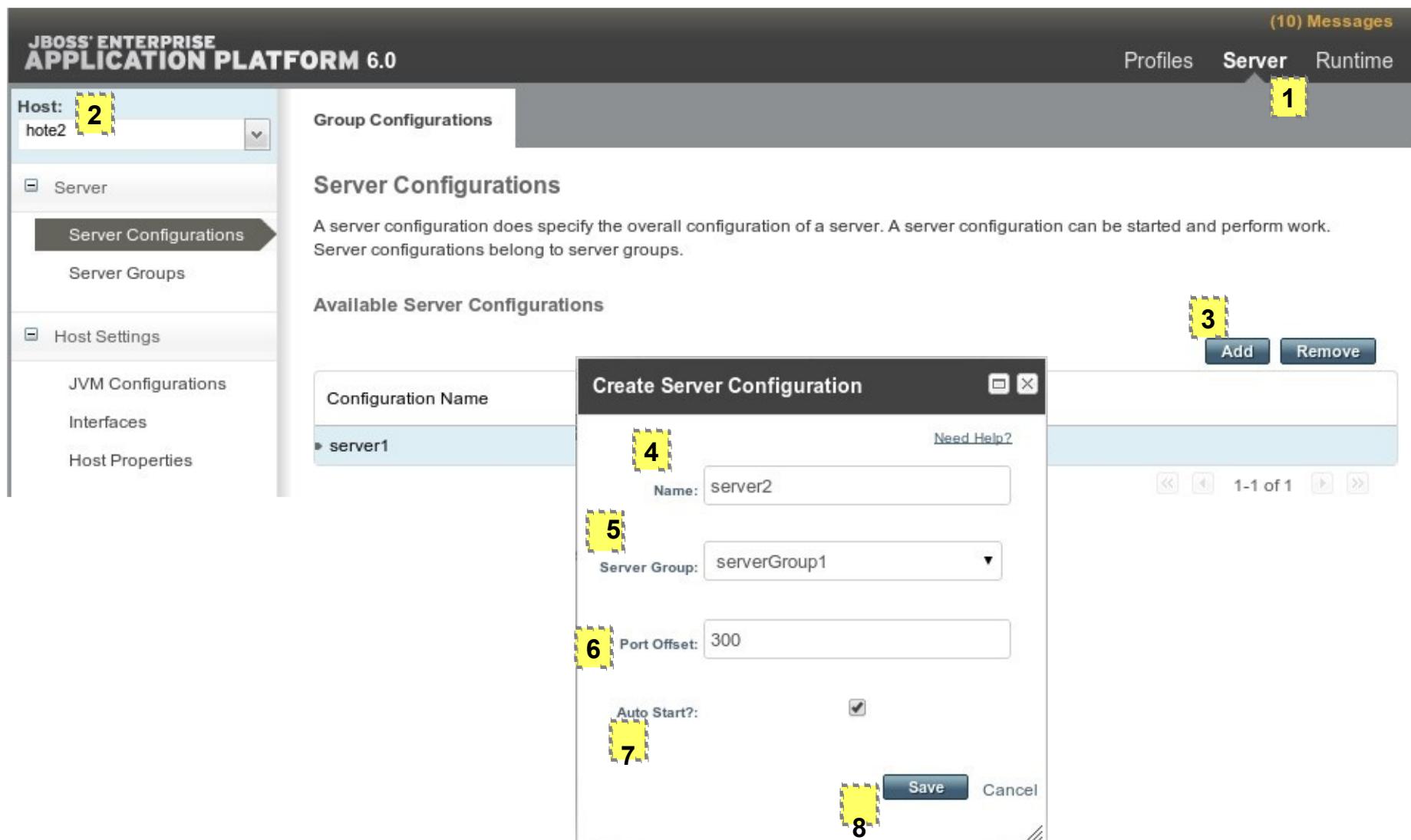
```
<servers>
    <server name="server-one" group="main-server-group">
        <!-- Remote JPDA debugging for a specific server
        <jvm name="default">
            <jvm-options>
                <option value="-Xrunjdwp:transport=dt_socket,address=8787,server=y,suspend=n"/>
            </jvm-options>
        </jvm>
        -->
    </server>
    <server name="server-two" group="main-server-group" auto-start="true">
        <!-- server-two avoids port conflicts by incrementing the ports in the default socket-group declared in the server-group -->
        <socket-bindings port-offset="150"/>
    </server>
    <server name="server-three" group="other-server-group" auto-start="false">
        <!-- server-three avoids port conflicts by incrementing the ports in the default socket-group declared in the server-group -->
        <socket-bindings port-offset="250"/>
    </server>
</servers>
```



Grammaire complète
d'un server

Définition de serveurs (2)

- Ajouter des serveurs à un hôte par la console :



Définition de serveurs (3)

- Compléter la définition d'un serveur :

- ▶ variables
- ▶ d'environnement

The screenshot shows the JBoss Enterprise Application Platform 6.0 interface. The left sidebar shows navigation options: Host (set to host2), Server (selected), Server Configurations (highlighted), Server Groups, Host Settings, JVM Configurations, Interfaces, and Host Properties. The main content area is titled 'Server Configurations' and contains a description: 'A server configuration does specify the overall configuration of a server. A server configuration can be started and perform work. Server configurations belong to server groups.' Below this is a table titled 'Available Server Configurations' with two rows:

Configuration Name	Server Group
server1	main-server-group
server2	serverGroup1

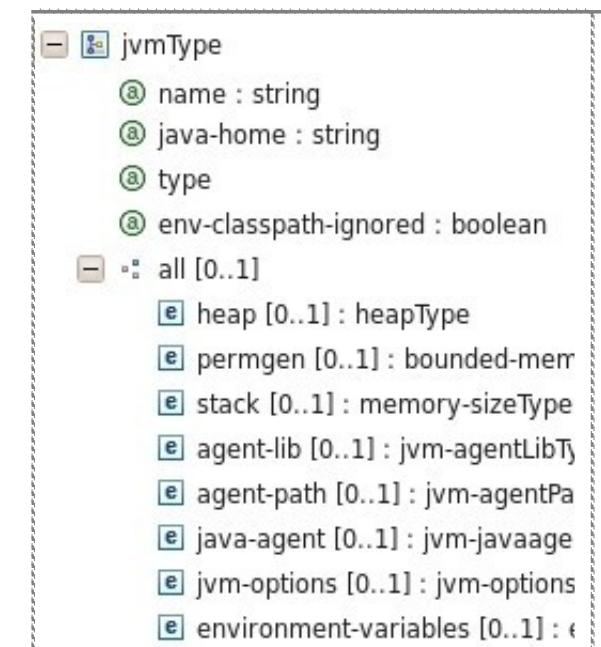
Below the table are buttons for 'Add' and 'Remove'. At the bottom, there are tabs for 'Attributes', 'JVM Configuration', and 'System Properties', with 'System Properties' currently selected. A table for system properties is shown with columns: Key, Value, Boot-Time?, and Option. Two purple arrows point from the text 'variables' and 'd'environnement' in the list above to the 'System Properties' table.

Définition de serveurs (4)

- **Paramètres des JVM :**

- ▶ contrairement au mode standalone, le paramétrage des JVM se fait via les fichiers de configuration domain.xml et host.xml
 - ▶ informations utilisées par le Process Controller
- ▶ Plusieurs définitions de JVM peuvent apparaître
 - ▶ utilisation d'un attribut name pour les distinguer
- ▶ Les définitions peuvent se faire à plusieurs niveaux :
 - ▶ au niveau de l'hôte
 - ▶ au niveau d'un groupe de serveur
 - ▶ au niveau d'un serveur
- ▶ le niveau le plus bas l'emporte

```
<jvms>
  <jvm name="default">
    <heap size="64m" max-size="256m"/>
    <permgen size="256m" max-size="256m"/>
    <jvm-options>
      <option value="-server"/>
    </jvm-options>
  </jvm>
</jvms>
```



Grammaire du type
jvmType

Pilotage des serveurs

- La console permet l'arrêt et le démarrage des serveurs :

- ▶ NB : sélectionner un serveur d'un hôte pour voir les serveurs de cet hôte

The screenshot shows the JBoss Enterprise Application Platform 6.0 interface. The left sidebar has a 'Server' dropdown set to 'server1' (2). The 'Server Instances' tab is selected (3). The top right shows 'Messages (10)' (1), 'Profiles', 'Server', and 'Runtime'. The main content area displays the 'Server Status (Host: host2)' (4). It explains that server instances represent runtime state, including VM status and subsystem specific state. A table lists two servers: 'server1' in the 'main-server-group' (status: Active) and 'server2' in 'serverGroup1' (status: Inactive). At the bottom, it shows 'Availability' and 'Environment Properties' tabs, and details for 'Server Instance: server1' (Running?: true) and 'Server Configuration: server1'. A green chalkboard in the bottom right corner displays 'TP 2.2 /'.

Server	Server Group	Status	Active
server1 (3)	main-server-group	✓	
server2	serverGroup1	✗	

Monitoring

- La console permet la consultation de plusieurs indicateurs de fonctionnement des serveurs :

The screenshot shows the JBoss Enterprise Application Platform 6.0 monitoring interface. The top navigation bar includes 'Profiles', 'Server' (selected), and 'Runtime'. A message count of '(10) Messages' is shown in the top right. The left sidebar has a 'Server' dropdown set to 'server1' (marked with a yellow box 2) and a tree view with nodes: Domain, Server Instances, Manage Deployments, Server Status (selected), JVM (highlighted with a dark grey arrow), Datasources (marked with a yellow box 3), JPA, JMS Destinations, Transactions, Web, Webservices, Runtime Operations, and OSGi.

Virtual Machine Status

Java HotSpot(TM) Server VM

Operating System: Linux 2.6.32-42-generic
Processors: 8

Heap Usage (mb)

Metric	Actual
Max:	1248
Used:	319
Committed:	1248
Init:	1303

Non Heap Usage (mb)

Metric	Actual
Max:	304
Used:	41
Committed:	258
Init:	258

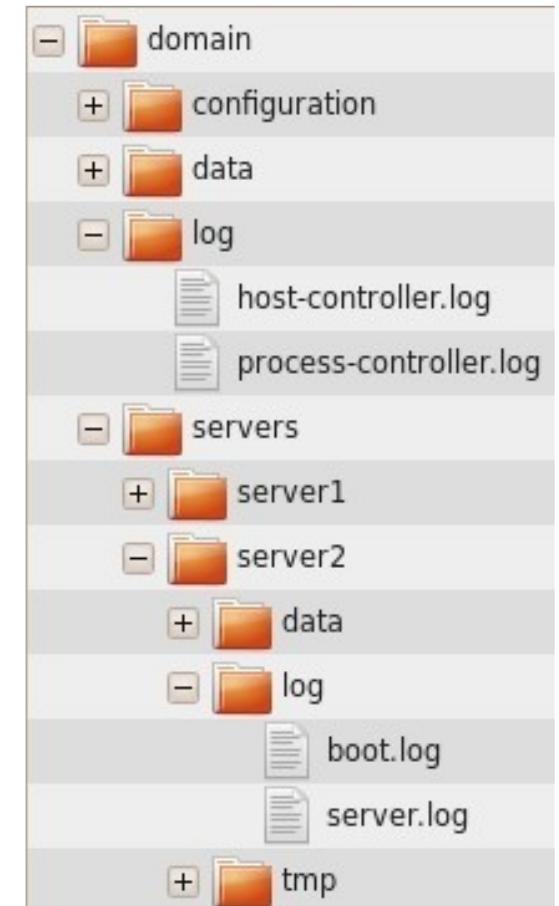
Thread Usage

Metric	Actual
Live:	63
Daemon:	23

Arborescence du répertoire domain

- **Le répertoire 'domain' :**

- ▶ configuration : emplacement par défaut des fichiers de configuration
- ▶ data : données internes
- ▶ log : fichiers de log du Process Controller et du Host Controller
- ▶ servers : un sous répertoire pour chaque serveur de l'hôte
- ▶ servers/serverX/data : données internes
- ▶ servers/serverX/log : fichiers de log du serveur



Administration des applications

Administration des applications

- Les formats de déploiement
- Procédures de déploiement d'un WAR et d'un EAR
- Déploiement par copie de fichiers
- Déploiement par la console et par ligne de commande
- Pilotage des applications
- Mise à jour et désinstallation
- Déploiement dans un domaine
- ClassLoaders et JBoss 7

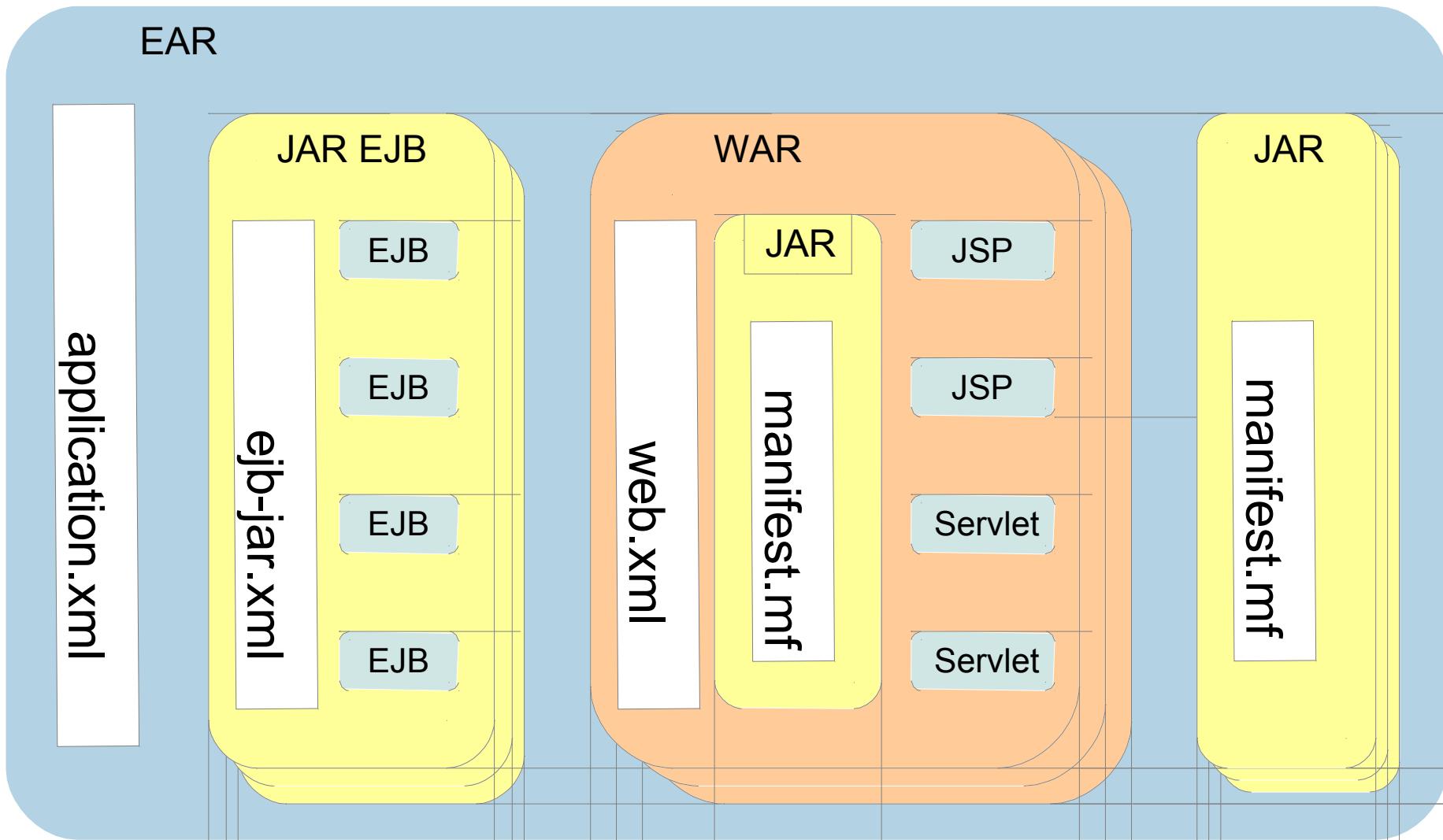
Composants d'une application d'entreprise (1)

- **La spécification Java EE définit deux principaux formats de déploiement :**
 - ▶ WAR : Web Application aRchive
 - ▶ EAR : Enterprise Application aRchive
- **Un WAR contient une application Web**
 - ▶ Constituée de Servlets, de JSP, de classes Java, de ressources diverses (.png,.gif, .css, .js, ...)
- **Un EAR peut contenir différents types de modules**
 - ▶ Modules Web : WAR
 - ▶ Modules EJB : JAR contenant des EJB
 - ▶ Des librairies de classes Java (.jar) utilisées par les modules

Composants d'une application d'entreprise (2)

- La spécification Java EE définit précisément l'organisation des différents formats de déploiement
 - ▶ WAR et EAR sont déployés sous forme de fichiers .war et .ear
 - ▶ Comme les .jar, ces fichiers sont en fait des fichiers zip
 - ▶ La spécification définit l'arborescence contenue dans ces fichiers
- Pour garantir la portabilité, les paramétrages des applications sont placés dans des fichiers XML dont la grammaire et l'emplacement sont strictement définis
 - ▶ EAR : fichier META-INF/application.xml
 - ▶ WAR : fichier WEB-INF/web.xml
 - ▶ EJB : fichier META-INF/ejb-jar.xml
- Depuis Java EE 5, ces 'descripteurs de déploiement' peuvent être remplacés par des annotations dans le code

Composants d'une application d'entreprise (3)



Procédures de déploiement

- **Plusieurs procédures de déploiement sont possibles :**

- ▶ par la console
- ▶ par la CLI
- ▶ par copie de fichier dans un répertoire surveillé en permanence

- **En mode domaine :**

- ▶ le déploiement par copie de fichier n'est pas supporté
- ▶ les applications déployés par la console ou la CLI seront envoyées sur les hôtes concernés

- **En mode standalone, le déploiement par copie de fichier peut être déclenché :**

- ▶ automatiquement dès la copie du fichier
- ▶ manuellement via la création de fichier 'marqueurs' (ex : mon_appli.dodeploy)

Deployment Scanner (1)

- Le déploiement par copie de fichiers du mode standalone est configurable dans le fichier `standalone.xml` :

- ▶ sous-système 'deployment-scanner'

```
<subsystem xmlns="urn:jboss:domain:deployment-scanner:1.1">
    <deployment-scanner path="deployments" relative-to="jboss.server.base.dir" scan-interval="5000"/>
</subsystem>
```

- ▶ par défaut, déploiement automatique pour les fichiers compressés et manuel pour les arborescences
 - ▶ évite que le déploiement commence avant que les fichiers soient terminés de copier

- Déploiement manuel :

- ▶ Créer le fichier `.dodeploy` :

- ▶ exemple, pour un war nommé `MonAppli.war`, copier le fichier dans le répertoire de déploiement et créer le fichier `MonAppli.war.dodeploy`

 <code>standalone-deployment-scanner-type</code>
③ <code>name</code> : string
③ <code>path</code> : anySimpleType
③ <code>relative-to</code> : string
③ <code>scan-enabled</code> : boolean
③ <code>scan-interval</code> : int true
③ <code>auto-deploy-zipped</code> : boolean
③ <code>auto-deploy-explored</code> : boolean false
③ <code>auto-deploy-xml</code> : boolean
③ <code>deployment-timeout</code> : int

Grammaire du type
`deployment-scanner`
(en bleu
valeurs par défaut)

Deployment Scanner (2)

- **Liste des fichiers marqueurs :**

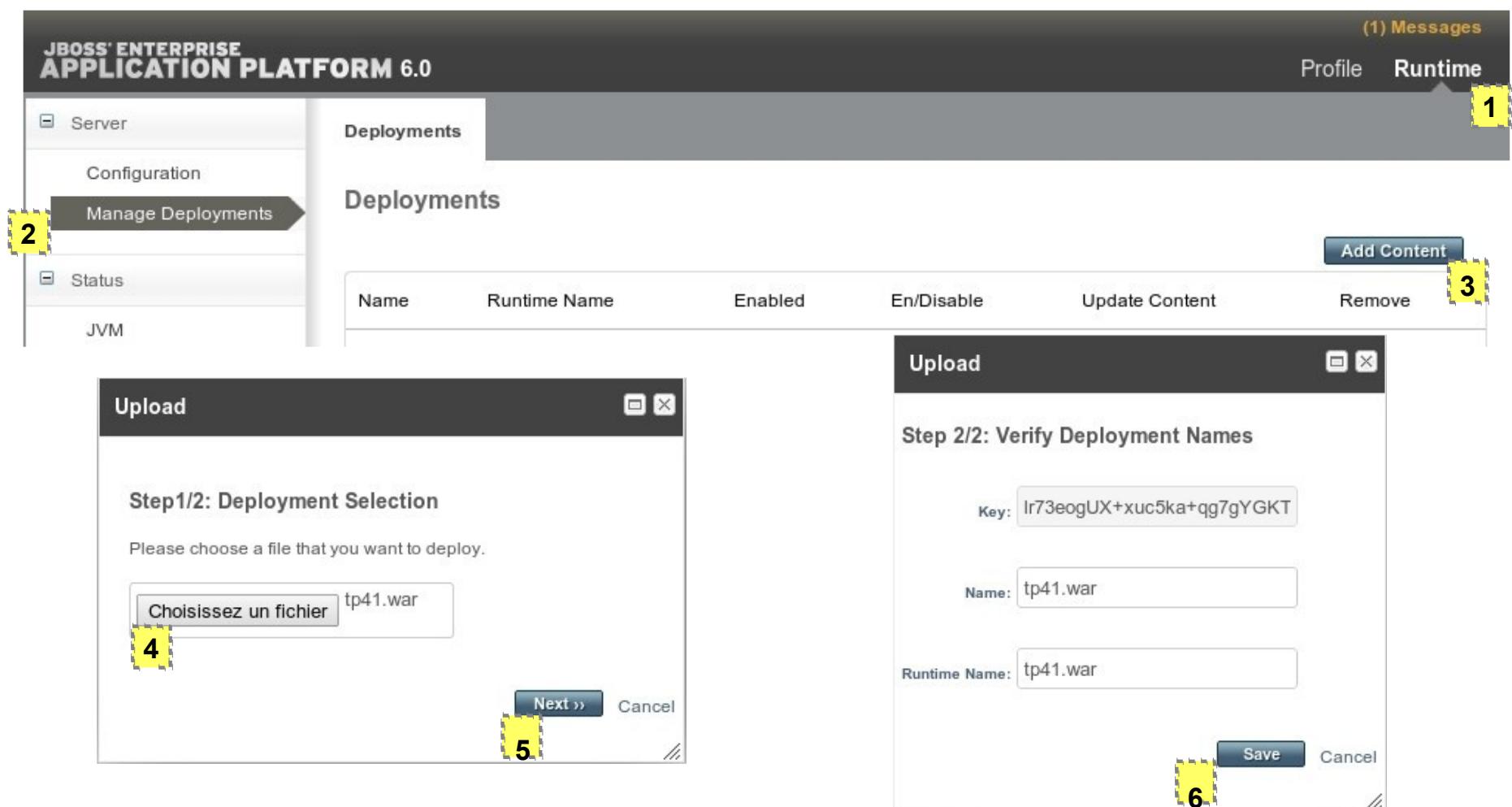
- ▶ créés par l'administrateur :
 - ▶ .dodeploy : pour demander le déploiement
 - ▶ .skipdeploy : pour désactiver le déploiement automatique sur une application donnée, supprimer le fichier pour réactiver le déploiement de l'application
- ▶ créés par le serveur :
 - ▶ .deployed : l'application a été correctement déployée, supprimer le fichier marqueur pour désinstaller l'application
 - ▶ .undeployed : l'application a été désinstallée, supprimer le fichier marqueur pour réinstaller l'application
 - ▶ .isdeploying et .isundeploying : procédure correspondant en cours d'exécution
 - ▶ .failed : échec du déploiement, consulter le fichier
 - ▶ .pending : installation suspendue, en attente de résolution d'un problème



Déploiement par la console (1)

- **Déploiement par la console en mode autonome :**

- ▶ upload du fichier à travers la
- ▶ console Procédure :



Déploiement par la console (2)

- **Une fois l'application déployée :**

- ▶ le fichier est stocké dans <JBOSS_HOME>standalone\data\content
 - ▶ arborescence non manipulable
- ▶ possibilité de l'activer et de la désactiver sans suppression du fichier

The screenshot shows the JBoss Enterprise Application Platform 6.0 interface. On the left, there's a sidebar with 'Server' and 'Status' sections. Under 'Status', 'Manage Deployments' is highlighted. The main area is titled 'Deployments' and contains a table:

Name	Runtime Name	Enabled	En/Disable	Update Content	Remove
tp41.war	tp41.war	∅	Enable	Update Content	Remove

A yellow box labeled '1' highlights the 'Enable' button. Below the table, a message says '1-1 of 1'. On the right, there are tabs for 'Profile' and 'Runtime'. At the top right, it says '(2) Messages'.

Below the main interface, a modal dialog box is open with the title 'Are you sure?'. It contains the question 'Enable tp41.war?' and two buttons: 'Confirm' (highlighted with a yellow box labeled '2') and 'Cancel'. A purple arrow points from the 'Confirm' button to the 'Enable' button in the table above. Another purple arrow points from the 'Enable' button in the table to the 'Enable' button in the modal dialog.

Log sur les applications

- **Messages concernant le cycle de vie des applications :**

- ▶ Installation

```
21:44:32,872 INFO [org.jboss.as.repository] (HttpManagementService-threads - 3) JBAS014900: Contenu ajouté dans location  
/home/bruno/Produits/JBoss-EAP-6.0.0.GA/jboss-eap-  
6.0/standalone/data/content/96/bef77a88145fec6e73991afaa83b81818a4fe6/content
```

- ▶ Démarrage

```
15:54:58,565 INFO [org.jboss.as.server.deployment] (MSC service thread 1-4) JBAS015876: Lancement du déploiement de "tp41.war"  
15:54:59,319 INFO [org.jboss.as.osgi] (MSC service thread 1-15) JBAS011907: Register module: Module "deployment.tp41.war:main" from Service  
Module Loader  
15:54:59,579 INFO [org.jboss.web] (MSC service thread 1-14) JBAS018210: Enregistrement du contexte web /tp41 15:54:59,607  
INFO [org.jboss.as.server] (HttpManagementService-threads - 8) JBAS018559: Déploiement de "tp41.war"
```

- ▶ Arrêt

```
21:35:21,259 INFO [org.jboss.as.osgi] (MSC service thread 1-8) JBAS011908: Unregister module: Module "deployment.tp41.war:main" from  
Service Module Loader  
21:35:21,350 INFO [org.jboss.as.server.deployment] (MSC service thread 1-2) JBAS015877: Arrêt du déploiement de tp41.war en 96ms  
21:35:21,389 INFO [org.jboss.as.server] (HttpManagementService-threads - 1) JBAS018558: Annulation du déploiement de "tp41.war"
```

- ▶ Désinstallation

```
21:39:39,724 INFO [org.jboss.as.repository] (HttpManagementService-threads - 2) JBAS014901: Contenu supprimé de la location  
/home/bruno/Produits/JBoss-EAP-6.0.0.GA/jboss-eap-  
6.0/standalone/data/content/96/bef77a88145fec6e73991afaa83b81818a4fe6/content
```

Console et copie de fichier

- Les applications déployées par copie de fichier apparaissent dans la console :

- ▶ mais uniquement en consultation

The screenshot shows the JBoss Enterprise Application Platform 6.0 deployment manager interface. The left sidebar has 'Manage Deployments' selected. The main area is titled 'Deployments' and lists two entries:

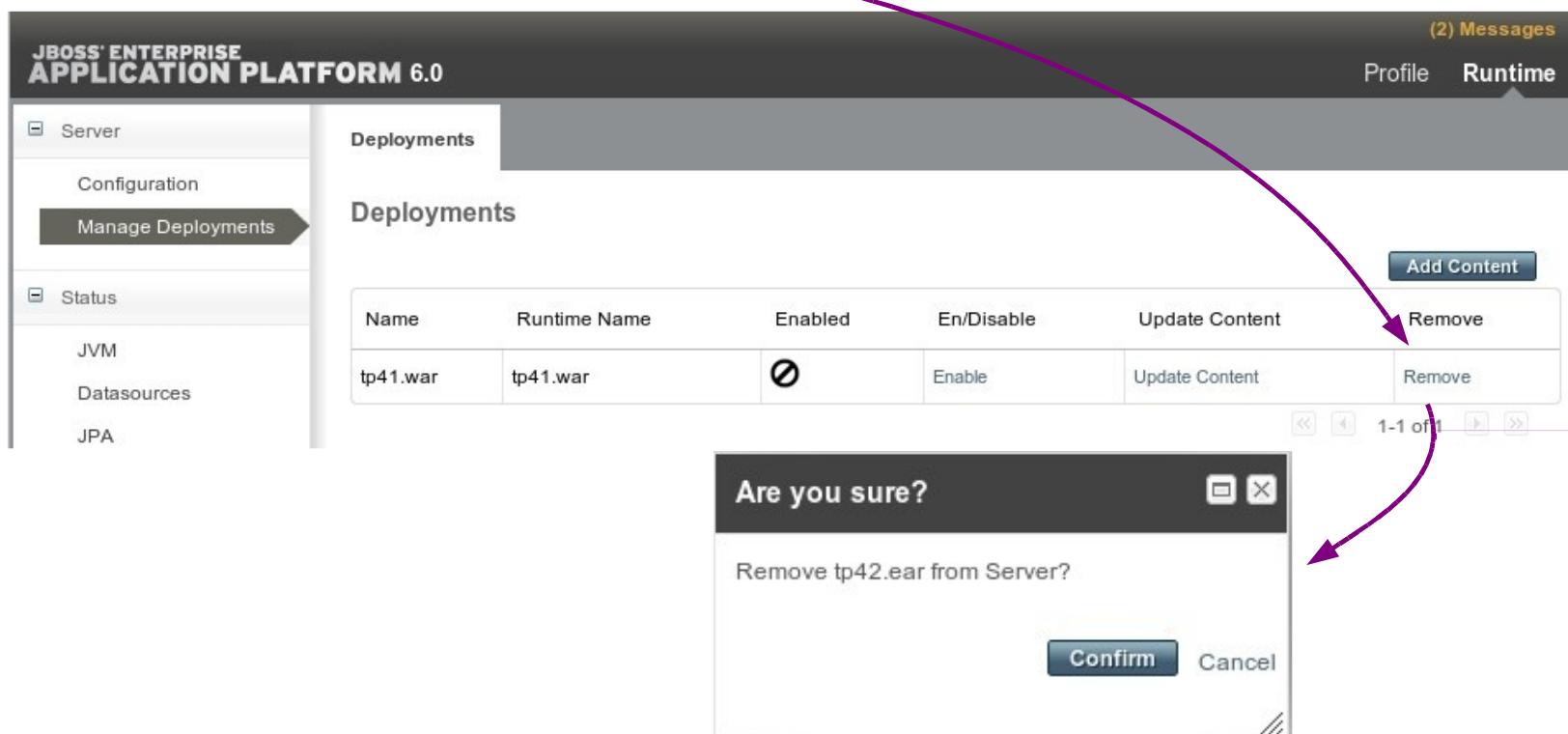
Name	Runtime Name	Enabled	En/Disable	Update Content	Remove
tp42.ear	tp42.ear	<input checked="" type="checkbox"/>	Disable	Update Content	Remove
tpDataSource1.ear File System Deployment	tpDataSource1.ear	<input checked="" type="checkbox"/>			

A purple arrow points from the text 'mais uniquement en consultation' to the 'tpDataSource1.ear' entry in the table.

Désinstallation d'une application

- Procédure de désinstallation :

- ▶ si utilisation de la copie de fichier, supprimer le fichier avec la console, utiliser le lien 'Remove'



```
21:39:39,724 INFO [org.jboss.as.repository] (HttpManagementService-threads - 2) JBAS014901: Contenu supprimé de  
location /home/bruno/Produits/JBoss-EAP-6.0.0.GA/jboss-eap-  
6.0/standalone/data/content/96/bef77a88145fec6e73991afaa83b81818a4fe6/content
```

Déploiement dans un domaine (1)

- Etapes de déploiement d'applications dans un domaine :

- ▶ ajouter l'application dans le référentiel géré par le Domain Controller 3
- ▶ associer l'application à un ou plusieurs groupe de 5

The screenshot shows the JBoss Enterprise Application Platform 6.0 deployment interface. The left sidebar shows navigation links: Server (selected), Domain, Server Instances (with Manage Deployments highlighted), Server Status, JVM, Datasources, and JNDI. The main content area has tabs: Deployment Content (selected) and Server Group Deployments. Under Deployment Content, there's a Content Repository table:

Name	Runtime Name	Add to Groups	Update Content	Remove
tp41.war	tp41.war	Add to Groups	Update Content	Remove
tp42.ear	tp42.ear	Add to Groups		

A modal window titled "Select server groups" is open over the table, with the number 4 pointing to its title bar. This modal lists server groups:

Server Group	Profile	Add to Group
main-server-group	full	<input type="checkbox"/>
other-server-group	full-ha	<input type="checkbox"/>
serverGroup1	default	<input checked="" type="checkbox"/>

The number 5 points to the checked checkbox in the last row. At the bottom of the modal, there are buttons for "Save" and "Cancel". The top right corner of the interface shows 1 (Messages), 2 (Profiles), 3 (Server), and 6 (Runtime).

Déploiement dans un domaine (2)

- Une fois déployée l'application peut être activée/désactivée et supprimée

The screenshot shows the JBoss Enterprise Application Platform 6.0 management console. The left sidebar has a tree view with 'server1' selected. The main area has tabs for 'Deployment Content' and 'Server Group Deployments'. The 'Server Group Deployments' tab is active, showing 'Server Groups' with 'main-server-group' and 'other-server-group'. Below this, 'serverGroup1' is expanded, showing 'Deployments for serverGroup1' with one entry: 'tp42.ear' (Runtime Name 'tp42.ear', Enabled checked). A yellow box labeled '1' is at the top right of the window. A yellow box labeled '2' is on the sidebar under 'Manage Deployments'. A yellow box labeled '3' is over the 'Server Group Deployments' tab. A yellow box labeled '4' is over 'serverGroup1'. A yellow box labeled '5' is over the 'Disable' link for 'tp42.ear'. A yellow box labeled '6' is over the 'Remove' link for 'tp42.ear'.

Name	Runtime Name	Enabled	En/Disable	Remove
tp42.ear	tp42.ear	<input checked="" type="checkbox"/>	Disable	Remove

TP 3.3

Descripteurs de déploiement JBoss

- Pour compléter les informations contenues dans les descripteurs de déploiement standards, JBoss propose ses propres descripteurs :

- ▶ jboss-web.xml
 - ▶ informations complémentaires pour un WAR
 - ▶ emplacement : <mon_WAR>/WEB-INF/
- ▶ jboss-ejb3.xml
 - ▶ informations complémentaires pour un JAR contenant des EJB
 - ▶ emplacement : <mon_JAR_avec_des_EJB>/META-INF/
- ▶ jboss-deployment-structure.xml
 - ▶ informations sur la stratégie de classloader à utiliser
 - ▶ emplacement : <mon_EAR>/META-INF ou dans
<mon_WAR>/WEB-INF/

- **Détail de la grammaire :**

- ▶ JBoss EAP 6 Development Guide, 15-1 p 358

- **Modification du nom de contexte d'une application déployée sous forme d'un fichier WAR :**

- ▶ élément 'context-root' :

```
<?xml version="1.0" encoding="UTF-8"?>
<jboss-web>
    <context-root>/test</context-root>
</jboss-web>
```

- ▶ NB : pour utiliser le contexte par défaut ('/'), il faut modifier la définition du sous-système web :

```
<subsystem xmlns="urn:jboss:domain:web:1.1"
default-virtual-server="default-host" native="false">
    <connector name="http" protocol="HTTP/1.1" scheme="http"
socket-binding="http"/>
    <virtual-server name="default-host" enable-welcome-root="false">
        <alias name="localhost"/>
        <alias name="example.com"/>
    </virtual-server>
</subsystem>
```

JBoss et gestion des ClassLoaders

- Par défaut, JBoss 7 respecte les préconisations des spécifications Java EE sur la gestion des classloaders :
 - ▶ chaque modules d'un EAR est chargé dans son propre classloader les modules sont isolés les uns des autres
- Possibilité de modifier le comportement par défaut :
 - ▶ fichier <EAR_HOME>/META-INF/jboss-deployment-structure.xml

```
<jboss-deployment-structure>
    <ear-subdeployments-isolated>false</ear-subdeployments-isolated>
</jboss-deployment-structure>
```



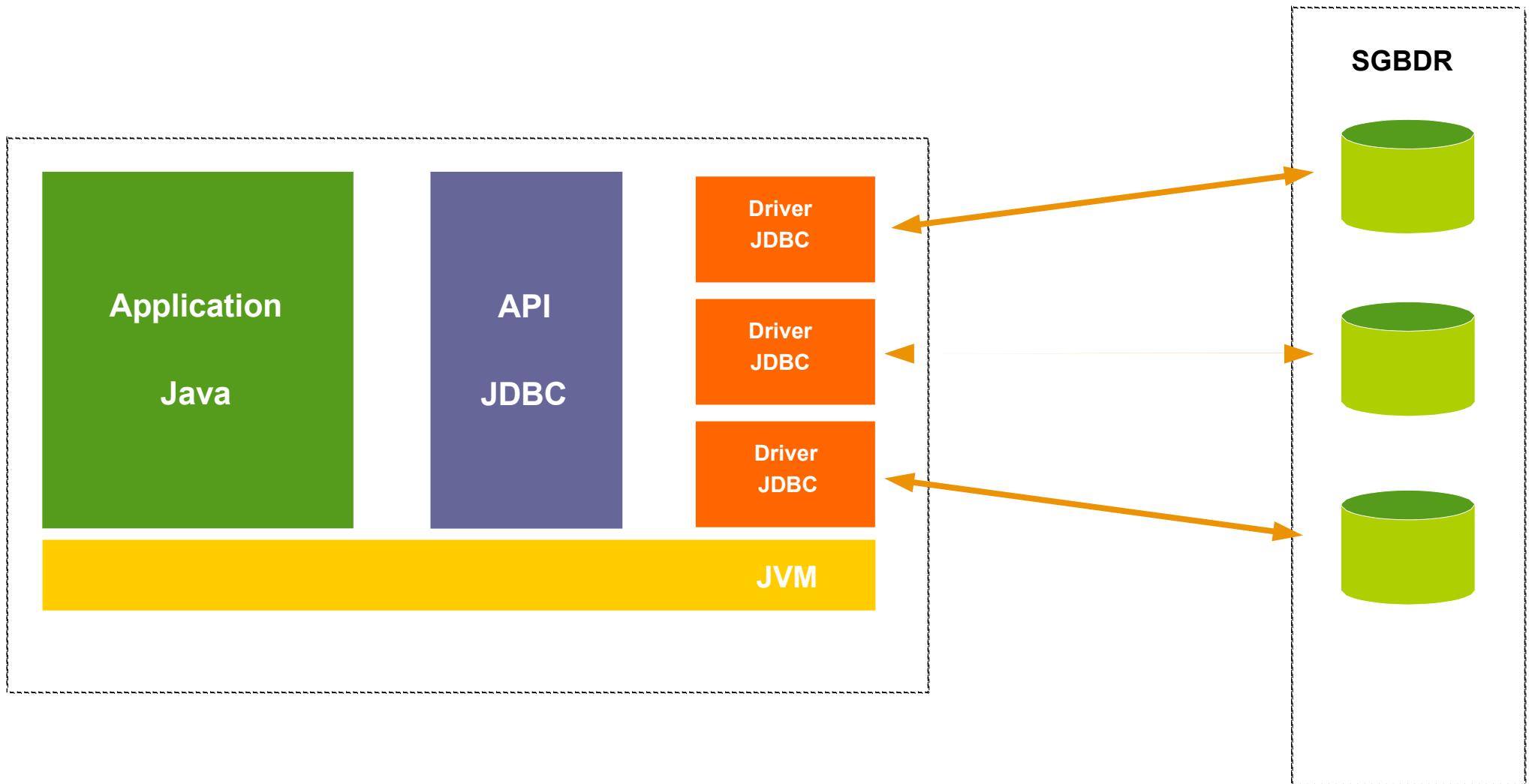
Accès aux données

- **Principes de JDBC et notion de DataSource**
- **Installation d'un driver JDBC**
- **Définition d'une DataSource**
- **Paramétrage d'un pool de connexions**

Présentation de JDBC (1)

- **JDBC : Java DataBase Connectivity**
- **API standard d'accès aux bases de données**
- **Indépendance vis-à-vis de la base ciblée**
- **Ensemble de classes et interfaces**
 - ▶ packages java.sql, javax.sql
- **Envoi de requêtes SQL**
- **Drivers JDBC**
 - ▶ Nécessaires pour faire le lien entre l'API JDBC et le moteur de la base de données. Disponibles sous forme de fichiers JAR.
 - ▶ Fournis par les éditeurs de base de données.

Présentation de JDBC (2)



Présentation de JDBC : types de drivers

- **Type 1**

- ▶ Pont JDBC-ODBC et driver ODBC. Inclus dans le JDK. Pas vraiment intéressant.

- **Type 2**

- ▶ Driver s'appuyant sur le client natif (DLL, .so) de la base de données.

- **Type 3**

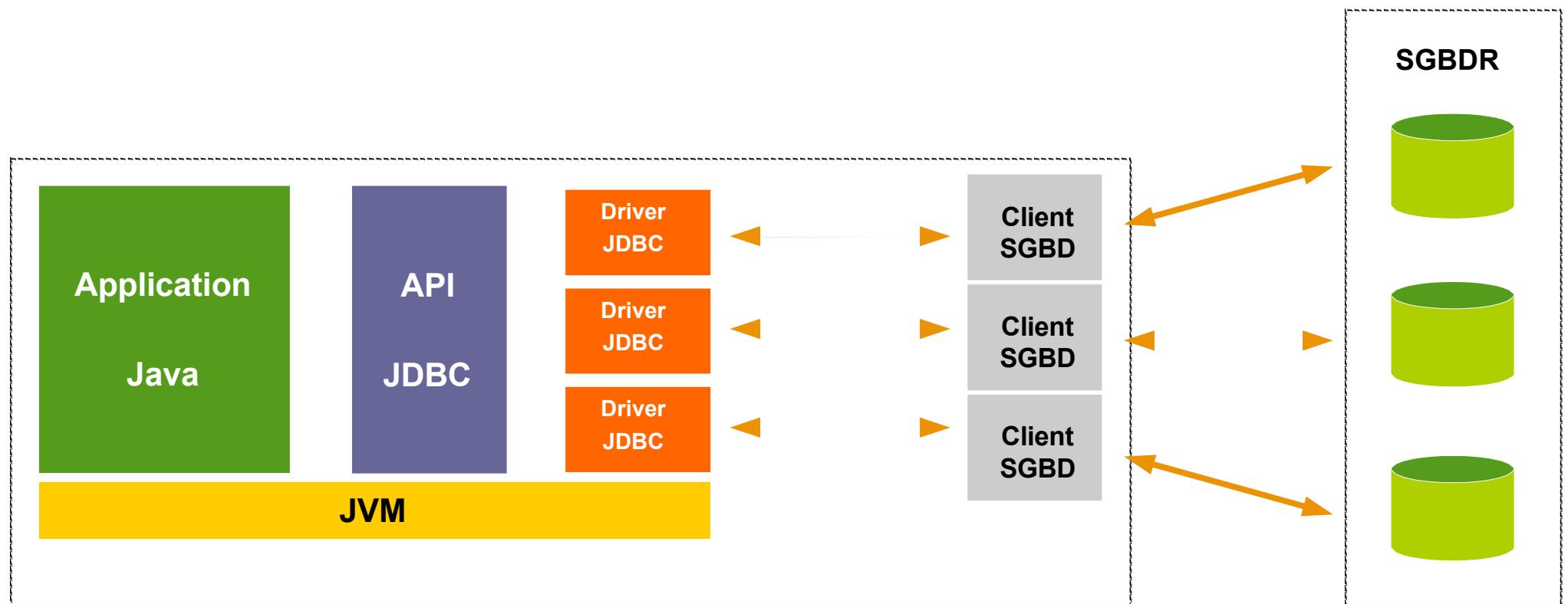
- ▶ Driver Java utilisant un protocole propriétaire. Installation côté client et côté serveur. Conçus pour être utilisés dans le cadre d'applets Java.

- **Type 4**

- ▶ Driver 100% Java implémentant le protocole natif de la base de données. Installation très simple : constitué d'un seul fichier JAR.

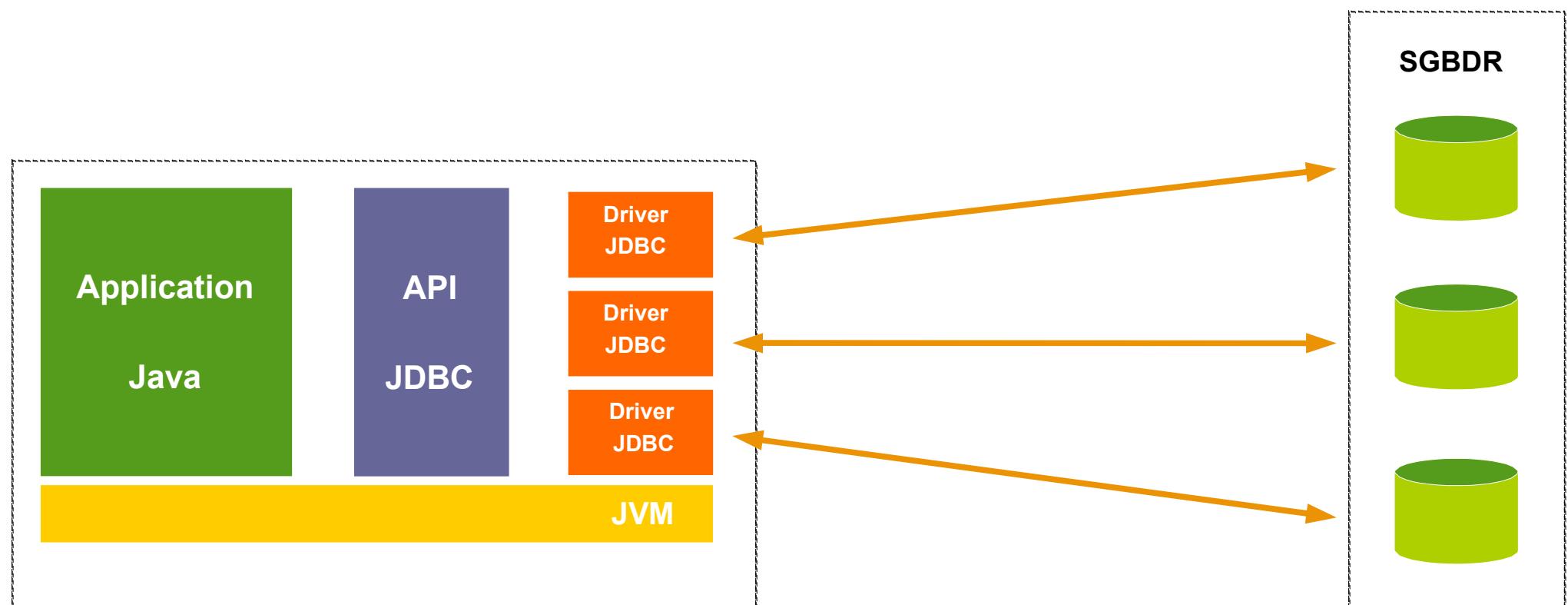
Présentation de JDBC : driver de type 2

- Utilise le client natif du SGBDR
- Traduit les appels JDBC en appel au client natif
- Exemple: Oracle JDBC OCI driver



Présentation de JDBC : driver de type 4

- Client du SGBDR écrit complètement en Java
- Fourni par l'éditeur du SGBDR
- Très simple à installer



- **Les informations de connexions aux bases sont généralement externalisées**
- **Une DataSource JDBC permet à l'application de demander une connexion en utilisant un nom logique :**
 - ▶ Les informations de connexions sont renseignées côté serveur d'applications
 - ▶ Remplace l'utilisation de la classe DriverManager de JDBC
 - ▶ Support des pools de connexions de façon portable
 - ▶ Utilisation de JNDI pour retrouver la DataSource :

```
javax.sql.DataSource ds =  
(javax.sql.DataSource) ctx.lookup("jdbc/MaBase");
```

Intérêt des pools de connexions

- **Les connexions et déconnexions aux bases de données sont coûteuses en terme de ressources :**
 - ▶ Temps de connexion
 - ▶ Mémoire utilisée côté client et côté serveur
- **Dans le cas d'une application Web, il serait trop coûteux de réserver une connexion à chaque utilisateur :**
 - ▶ HTTP fonctionne en mode déconnecté (ne permet de savoir quand l'utilisateur quitte l'application)
 - ▶ Le nombre d'utilisateurs peut être important
- **Les serveurs Java EE proposent un mécanisme de pool de connexions qui permet :**
 - ▶ De mettre en commun les connexions
 - ▶ De limiter le nombre de connexions ouvertes

Installation d'un driver JDBC (1)

- Depuis JBoss 7, l'installation d'un driver JDBC est possible de deux façons :

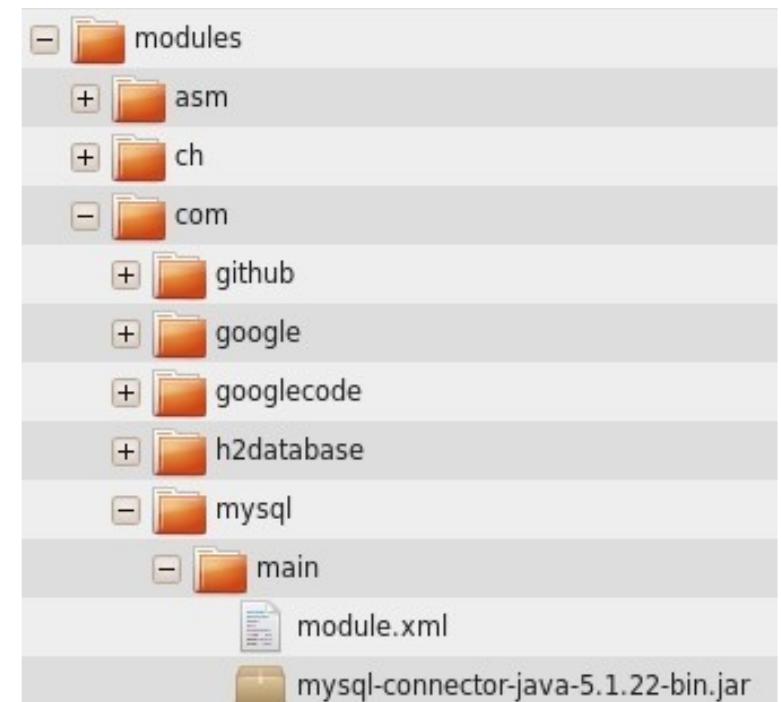
- ▶ en utilisant la notion de modules de JBoss 7
 - ▶ c'est le mode explicitement privilégié
 - ▶ nécessite la création d'un module JBoss manuellement et sa copie sur tous les hôtes du domaine
- ▶ en déployant le JAR du driver comme une application classique
 - ▶ procédure simple,
 - ▶ permet notamment la diffusion du JAR vers chacun des hôtes mais :
 - ▶ *problèmes avec certains drivers*
 - ▶ *informations manquantes pour utiliser la notion de DataSource XA*

Installation d'un driver JDBC (2)

- **Pour installer un driver sous forme de module :**

- ▶ placer le fichier jar dans une arborescence sous <JBOSS_HOME>/modules/
 - ▶ ex : com.mysql
 - ▶ créer un sous répertoire 'main' pour y mettre le JAR et le fichier module.xml
- ▶ créer le fichier module.xml
 - ▶ l'attribut 'name' doit correspondre à l'arborescence

```
<?xml version="1.0" encoding="UTF-8"?>
<module xmlns="urn:jboss:module:1.0" name="com.mysql">
    <resources>
        <resource-root path="mysql-connector-java-5.1.22-bin.jar"/>
    </resources>
    <dependencies>
        <module name="javax.api"/>
        <module name="javax.transaction.api"/>
    </dependencies>
</module>
```



Installation d'un driver JDBC (3)

- Une fois le driver installé sous forme de module, il faut le déclarer dans le profil :

- ▶ sous-système 'jboss:admin:datasources'



```
<subsystem xmlns="urn:jboss:domain:datasources:1.1">
<datasources>
    <datasource jndi-name="java:jboss/datasources/ExampleDS" pool-name="ExampleDS" enabled="true" use-java-context="true">
        <connection-url>jdbc:h2:mem:test;DB_CLOSE_DELAY=-1</connection-url>
        <driver>h2</driver>
        <security>
            <user-name>sa</user-name>
            <password>sa</password>
        </security>
    </datasource>
    <drivers>
        <driver name="h2" module="com.h2database.h2">
            <xa-datasource-class>org.h2.jdbcx.JdbcDataSource</xa-datasource-class>
        </driver>
        <driver name="mysql" module="com.mysql">
            <xa-datasource-class> com.mysql.jdbc.jdbc2.optional.MysqlXADataSource
                </xa-datasource-class>
        </driver>
    </drivers>
</datasources>
</subsystem>
```

Configuration d'une DataSource (1)

- Le sous-système 'jboss:admin:datasources' contient aussi les définitions de DataSource :

```
<subsystem xmlns="urn:jboss:domain:datasources:1.1">
  <datasources>
    <datasource jta="false" jndi-name="java:/ds/TestMySQL" pool-name="TestMySQL" enabled="false" use-ccm="false">
      <connection-url>jdbc:mysql://192.168.0.49:3306/test</connection-url>
      <driver-class>com.mysql.jdbc.Driver</driver-class>
      <driver>mysql</driver>
      <pool>
        <min-pool-size>0</min-pool-size>
        <max-pool-size>10</max-pool-size>
        <prefill>true</prefill>
      </pool>
      <security>
        <user-name>root</user-name>
        <password>admin</password>
      </security>
      <validation>
        <validate-on-match>false</validate-on-match>
        <background-validation>false</background-validation>
      </validation>
      <statement>
        <share-prepared-statements>false</share-prepared-statements>
      </statement>
    </datasource>
```

Configuration d'une DataSource (2)

- Ajout d'une DataSource par la console :

JBoss' ENTERPRISE APPLICATION PLATFORM 6.0

(2) Messages

Profile Runtime

1

2

3

Activation/Désactivation

4

Obligation d'utiliser le préfixe 'java:/'

5

6

Profile

Connector

JCA

Datasources

Resource Adapters

Mail

Container

Core

Infinispan

Datasources

X A Datasources

JDBC Datasources

JDBC datasource configurations.

Available Datasources

Name	JNDI	Enabled?
ExampleDS	java:jboss/datasources/ExampleDS	<input checked="" type="checkbox"/>

Add Remove Disable

Create Datasource

Step 1/3: Datasource Attributes

Name: TestMySQL

JNDI Name: java:/ds/TestMySQL

Next > Cancel

Step 2/3: JDBC Driver

Select one of the deployed JDBC driver.

Choose Driver

Name

postgresql-9.1-902.jdbc4.jar

mysql

h2

1-3 of 3

Next > Cancel

Step 3/3: Connection Settings

Connection URL: jdbc:mysql://localhost:3306/Test

Username: admin

Password: ****

Security Domain:

Need Help?

Done Cancel

Configuration d'une DataSource (3)

- La liste des DataSources permet :

- ▶ d'activer/désactiver une DataSource
- ▶ de modifier sa configuration
- ▶ de tester la DataSource

The screenshot shows the JBoss Enterprise Application Platform 6.0 interface for managing JDBC DataSources. The left sidebar menu is open, showing various configuration sections like Profile, Connector, JCA, and Datasources (which is highlighted with a yellow box and labeled '2'). The main content area displays the 'JDBC Datasources' configuration, with tabs for 'Datasources' and 'XA Datasources'. The 'Datasources' tab is selected, showing a table of available DataSources:

Name	JNDI	Enabled?
ExampleDS	java:jboss/datasources/ExampleDS	<input checked="" type="checkbox"/>
TestMySQL	java:/ds/TestMySQL	<input checked="" type="checkbox"/>

Below the table, there's a 'Selection' section with tabs for Attributes, Connection (which is highlighted with a yellow box and labeled '5'), Security, Properties, Pool, and Validation. An 'Edit' button is located next to the Connection tab. At the bottom, the Connection URL is set to 'jdbc:mysql://192.168.0.49:3306/test' and the New Connection Sql field is empty.

A purple arrow originates from a chalkboard on the right labeled 'TP 4.1' and points to the 'Test Connection' button, which is highlighted with a yellow box and labeled '4'.

Datasource Connection

Successfully created JDBC connection.
Successfully connected to database TestMySQL.

OK

Profile Runtime

1

2

3

4

5

TP 4.1

- **Fichiers -ds.xml :**

- ▶ dans les versions précédentes de JBoss AS la déclaration de chaque DataSource se faisait dans un fichier spécifique
- ▶ ce mode de déploiement reste supporté
- ▶ mais déconseillé explicitement pour l'exploitation

6.1.5. Deployment of -ds.xml files

In JBoss Enterprise Application Platform 6, datasources are defined as a resource of the server subsystem. In previous versions, a *** -ds.xml** datasource configuration file was required in the deployment directory of the server configuration. *** -ds.xml** files can still be deployed in JBoss Enterprise Application Platform 6, following the schema available here:

http://docs.jboss.org/ironjacamar/schema/datasources_1_1.xsd.



Warning

This feature should only be used for development. It is not recommended for production environments, because it is not supported by the JBoss administrative and management tools.

EAP 6 – Administration and Configuration Guide – section 6.1.5

Référence de DataSource (1)

- **La notion de DataSource permet d'isoler l'application des paramètres spécifiques à la base de données :**

- ▶ mais le nom JNDI utilisé dans le code de l'application et celui défini par l'administrateur doivent être identiques

- **La notion de référence permet de créer un niveau d'indirection :**

- ▶ Le développeur déclare l'existence d'une référence vers une DataSource dans le fichier web.xml. Il donne un nom à la référence et l'utilise dans le code

```
<resource-ref id="MaRef">
    <res-ref-name>MaReferenceDeDataSource</res-ref-name>
    <res-type>javax.sql.DataSource</res-type>
```

...

- ▶ Lors du déploiement, l'administrateur doit associer cette référence à une DataSource existante
- ▶ Avec JBoss, l'association se fait dans le fichier jboss-web.xml :
 - ▶ ce qui diminue largement l'intérêt de cette notion

Référence de DataSource (2)

- Ouverture d'une connexion :

```
Context ctx = new InitialContext();
DataSource ds = (DataSource) ctx.lookup(dsName);
Connection connexion = ds.getConnection();
```

- Sans utilisation de la référence :

```
dsName      = "java:/ds/MaDataSource"
```

- Si utilisation de la référence :

```
dsName      = "java:comp/env/MaReferenceDeDataSource"
```

- Fichier jboss-web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<resource-ref>
    <res-ref-name>MaReferenceDeDataSource</res-ref-name>
    <jndi-name>java:/ds/MaDataSource</jndi-name>
</resource-ref>
```

Tracer les appels JDBC

- Au niveau de chaque DataSource, il est possible d'activer la trace de tous les appels JDBC :

- ▶ Modifier la définition de la DataSource :

```
<subsystem xmlns="urn:jboss:domain:datasources:1.1">
  <datasources>
    <datasource jta="false" jndi-name="java:/ds/TestMySQL"
                pool-name="TestMySQL" enabled="false" use-ccm="false" spy="true">
      ...
    </datasource>
  </datasources>
</subsystem>
```

- ▶ Ajouter un logger :

- ▶ les traces apparaîtront dans le fichier de log du serveur

```
<logger category="jacorb.config">
  <level name="ERROR"/>
</logger>
<logger category="jboss.jdbc.spy">
  <level name="DEBUG"/>
</logger>
<root-logger>
  <level name="INFO"/>
  <handlers>
    <handler name="CONSOLE"/>
    <handler name="FILE"/>
  </handlers>
</root-logger>
```

Administration en ligne de commande

Administration en ligne de commande

- Lancement de la CLI (Command Line Interface)
- Le mode interactif
- Les principes de bases
- Exemples d'utilisation
- Le mode batch
- Sauvegarde de la configuration

- **Lancement du CLI :**

- ▶ <JBOSS_HOME>/bin/jboss-cli.sh ou <JBOSS_HOME>\bin\jboss-cli.bat

- **Par défaut utilisation en mode interactif**

- **Options de lancement :**

- ▶ --connect : pour se connecter immédiatement à un serveur
 - ▶ --controller=host:port : à utiliser avec l'option '--connect', permet d'indiquer l'adresse et le numéro de port (par défaut localhost et 9999)
 - ▶ ex : ./jboss-cli.sh --connect --controller=192.168.0.28:9999
 - ▶ --file : exécution des commandes se trouvant dans le fichier indiqué
 - ▶ --command : exécution de la commande indiquée
 - ▶ --commands : exécution des commandes passées en paramètre sous la forme d'une chaîne avec des virgules comme séparateurs
 - ▶ --user et --password : pour éviter le prompt
 - ▶ --gui : pour utiliser le mode graphique

Le mode interactif

- En mode interactif :

- ▶ utiliser la touche 'Tab' pour obtenir la liste des commandes

```
/bin$ ./jboss-cli.sh --connect  
[standalone@localhost:9999 /]  
  
batch          cd           clear  
command        connect       data-source  
deploy         deployment-info echo-dmr  
help          history       jdbc-driver-info  
ls             module        pwd  
quit          read-attribute read-operation  
undeploy       version       xa-data-source  
[standalone@localhost:9999 /]
```

- ▶ aide détaillée d'une commande :

```
deploy (file_path [--script=script_name] [--name=deployment_name] [--runtime-  
name=deployment_runtime_name]  
        [--force |           [--unmanaged]]  
        --disabled]  
      | --name=deployment_name  
      [--server-groups=group_name (,group_name)* | --all-server-groups]  
      [--headers={operation_header (;operation_header)*}]
```

DESCRIPTION

Deploys the application designated by the file_path or enables an already existing

...

Principes de bases du CLI (1)

- **La logique d'utilisation du CLI est la suivante :**

- ▶ chaque entrée dans les fichiers de configuration est identifiée par son chemin sur
- ▶ chaque entrée des 'opérations' peuvent être exécutées
- ▶ en mode interactif, il est possible de naviguer dans la configuration du serveur :
 - ▶ reprend les commandes classiques de navigation dans les répertoires :
 - ▶ *ls , cd et pwd*
 - ▶ le point de départ '/', est la racine de la configuration

```
[standalone@localhost:9999 /] pwd
/
[standalone@localhost:9999 /] ls
core-service
interface
subsystem
management-major-version=1
namespaces=[]
product-version=6.0.0.GA
release-version=7.1.2.Final-redhat-1
server-state=running
[standalone@localhost:9999 /] cd deployment
[standalone@localhost:9999 deployment] ls
postgresql-9.1-902.jdbc4.jar    tp42.ear
[standalone@localhost:9999 deployment]
```

extension
socket-binding-group
launch-type=STANDALONE
name=blc4-ubuntu
product-name=EAP
release-codename=Steropes
schema-locations=[]

tpDatasource.war

Principes de bases du CLI (2)

- Chaque élément de la configuration peut avoir des attributs et des sous-éléments :

- ▶ c'est la structure classique du format XML
- ▶ la commande 'ls' affiche les deux

```
[standalone@localhost:9999 deployment] ls
postgresql-9.1-902.jdbc4.jar      tp42.ear
[standalone@localhost:9999 deployment] cd tp42.ear
[standalone@localhost:9999 deployment=tp42.ear] ls

subdeployment
subsystem
content=[{"hash" => bytes { 0x53, 0x62, 0x92, 0x23, 0xdb, 0x21, 0xbe, 0x79, 0x8a, 0x37, 0x91,
0x0b, 0xe1, 0x53, 0x86, 0x14, 0x72, 0x60, 0x5e, 0x6a }}]
enabled=false
name=tp42.ear
persistent=true
runtime-name=tp42.ear
status=STOPPED
[standalone@localhost
:9999
deployment=tp42.ear]
```

Principes de bases du CLI (3)

- La commande 'ls -l' affiche plus de détail :

```
[standalone@localhost:9999 deployment=tp42.ear] ls -l
ATTRIBUTE      VALUE          TYPE
content        [{"hash" => bytes { 0x53, 0x62, 0x92, 0x23, 0xdb, 0x21, 0xbe,
0x79, 0x8a, 0x37, 0x91, 0x0b, 0xe1, 0x53, 0x86, 0x14, 0x72, 0x60, 0x5e,
0x6a }}] LIST
enabled       false          BOOLEAN
name          tp42.ear       STRING
persistent    true           BOOLEAN
runtime-name   tp42.ear       STRING
status        STOPPED        STRING

CHILD          MIN-OCCURS MAX-OCCURS
subdeployment  0             n/a
subsystem      0             n/a
```

- Sur chaque élément des 'opérations' peuvent être exécutées :

- ▶ la commande 'read-operation' affiche la liste des opérations de l'élément

```
[standalone@localhost:9999 deployment=tp42.ear] read-operation
add                  deploy          read-attribute
read-children-names  read-children-resources  read-children-types
read-operation-description  read-operation-names  read-resource
read-resource-description  redeploy         remove
undefine-attribute     undeploy         whoami
write-attribute
```

Principes de bases du CLI (4)

- La syntaxe d'exécution des opérations est :

- ▶ :<nom_operation>(nom_paramètre=valeur_paramètre,p2=v2)

- ▶ Exemples :

- ▶ opération sans paramètre, sans valeur de retour

```
[deployment=tp42.ear] :undeploy  
{"outcome" => "success"}
```

- ▶ opération sans paramètre, avec valeur de retour

```
:whoami  
{  
    "outcome" => "success",  
    "result" => {"identity" => {  
        "username" => "$local",  
        "realm" => "ManagementRealm"  
    }}  
}
```

- ▶ opération avec un paramètre

```
[deployment=tp42.ear] :read-attribute(name="status")  
  
{  
    "outcome" => "success",  
    "result" => "STOPPED"  
}
```

- ▶ plusieurs paramètres

```
[data-source=TestMySQL] :write-attribute(name="max-pool-size", value="20")  
{"outcome" => "success"}
```

- ▶ documentation d'une opération :

```
:read-operation-description(name="write-attribute")
```

- **Construction de commande :**

- ▶ en mode interactif, sans autres précisions, les opérations s'exécutent sur l'élément courant
- ▶ possibilité d'exécuter une commande sur un élément précis :
 - ▶ nécessaire en mode non interactif, les commandes de navigation n'étant pas disponibles
 - ▶ syntaxe : <id_élément>:<nom_opération>
 - ▶ exemple :

```
[standalone@localhost:9999 /] /subsystem=datasources/data-source=TestMySQL:write-attribute(name="max-pool-size", value="20")
{"outcome" => "success"}
```

Exemples de commandes utiles (1)

- **Arrêt d'un serveur standalone :**

- ▶ en mode interactif : `[standalone@localhost:9999 /] :shutdown`
- ▶ en mode commande : `./jboss-cli.sh --connect command=:shutdown`

- **Arrêt en mode domaine :**

- ▶ arrêt du Domain Controller : `/host=master:shutdown`
- ▶ arrêt d'un Host Controller : `/host=hote2:shutdown`
- ▶ liste des hôtes et des serveurs en mode interactif :

```
[domain@192.168.0.28:9999 /] cd host
[domain@192.168.0.28:9999 host] ls
hote2    master
[domain@192.168.0.28:9999 host] cd hote2
[domain@192.168.0.28:9999 host=hote2] cd server
[domain@192.168.0.28:9999 server] ls
server1    server2
```

- ▶ rechargement de la configuration :

/host=master:reload

`[standalone@localhost:9999 /] :reload`



Exemples de commandes utiles (2)

- **Gestion des applications en mode standalone :**

- ▶ Liste des applications déployées :

```
[domain@192.168.0.28:9999 /] deploy  
tp41.war      tp42.ear
```

- ▶ Déploiement :

```
[standalone@localhost:9999 /] deploy /home/bruno/temp/tpScript1.ear
```

- ▶ Déploiement en différant l'activation :

```
deploy /home/bruno/temp/tpScript1.ear --disabled
```

- ▶ Activation : `deploy --name=tpScript1.ear`

- ▶ Mise à jour :

```
deploy --force --name=tpScript1.ear /home/bruno/temp/tpScript2.ear
```

- ▶ Suppression :

```
undeploy tpScript1.ear
```



Exemples de commandes utiles (3)

- **Gestion des applications en mode domaine :**

- ▶ déploiement sur tous les serveurs

```
deploy ../tpScript1.ear --all-server-groups
```

- ▶ déploiement sur un groupe de serveurs :

```
deploy ../tpScript1.ear --server-groups=serverGroup1
```

- ▶ désinstallation sur tous les groupes où l'application était installée :

```
undeploy tpScript1.ear --all-relevant-server-groups
```

- ▶ désinstallation sur un groupe :

```
undeploy tpScript1.ear --server-groups=serverGroup1
```



Le mode 'batch'

- Par défaut les commandes sont exécutées immédiatement
- Possibilité d'exécuter une série de commandes en mode batch :
 - ▶ si une commande échoue toutes les précédentes sont annulées la
 - ▶ commande 'batch' permet d'entrer en mode batch
 - ▶ les commandes saisies ne sont plus exécutées immédiatement
 - ▶ la commande 'run-batch' lance l'exécution des commandes historisées
 - ▶ Exemple : déclaration d'un driver et d'une datasource associée

```
[domain@192.168.0.28:9999 /] batch
[domain@192.168.0.28:9999 / #] /profile=default/subsystem=datasources/jdbc-driver=mysql:add(driver-
name=mysql,driver-module-name=com.mysql,driver-xa-datasource-class-
name=com.mysql.jdbc.jdbc2.optional.MysqlXADataSource)
#1 /profile=default/subsystem=datasources/jdbc-driver=mysql:add(driver-name=mysql,driver-module-
name=com.mysql,driver-xa-datasource-class-name=com.mysql.jdbc.jdbc2.optional.MysqlXADataSource)
[domain@192.168.0.28:9999 / #] data-source add --profile=default --name=DSMySQL --jndi-
name=java:/ds/Test2MySQL --driver-name=mysql --connection-url=jdbc:mysql://192.168.0.49/test?
user=root&password=admin
#2 data-source add --profile=default --name=DSMySQL --jndi-name=java:/ds/Test2MySQL --driver-name=mysql
--connection-url=jdbc:mysql://192.168.0.49/test?user=root&password=admin
[domain@192.168.0.28:9999 / #] data-source enable --profile=default --name=DSMySQL
#3 data-source enable --profile=default --name=DSMySQL
[domain@192.168.0.28:9999 / #] run-batch
The batch executed successfully.
```

Sauvegarde de la configuration

- Que ce soit en mode standalone ou en mode domaine, les fichiers de configuration sont automatiquement sauvegardés :

- ▶ notamment dans le sous-répertoire *configuration/current* qui contient la version utilisée lors du dernier lancement
- ▶ en utilisant la CLI, il est possible de forcer la génération d'une sauvegarde:
 - ▶ pour l'utiliser, il faudra passer par les options, --config-file, --domain-config et --host-config, des commandes de lancements domain.sh et standalone.sh

```
[domain@192.168.0.28:9999 /] :take-snapshot
{
    "outcome" => "success",
    "result" => "/home/bruno/Produits/jboss-eap-
6.0/domain/configuration/domain_xml_history/snapshot/20120920-144652513domain.xml"
}
[domain@192.168.0.28:9999 /] :list-snapshots
{
    "outcome" => "success",
    "result" => {
        "directory" => "/home/bruno/Produits/jboss-eap-
6.0/domain/configuration/domain_xml_history/snapshot",
        "names" => ["20120920-144652513domain.xml"]
    }
}
```

Lien avec le serveur HTTP et clustering

Améliorer les performances

- Répartition de charge et haute disponibilité avec JBoss 7
- Mise en place du clustering
- L'affinité de session
- La réPLICATION des sessions
- Configuration de la JVM
- Optimisations des conteneurs Web et EJB

- **Par défaut chaque serveur d'applications est autonome :**

- ▶ la notion de domaine, rend possible l'administration centralisée à partir du Domain Controller :
 - ▶ ne change rien au fait que les serveurs ne communiquent pas entre eux le
 - ▶ Domain Controller n'intervient pas dans le traitement des requêtes
- ▶ l'utilisation des fonctionnalités dites 'HA' (High Availability) permet à plusieurs serveurs de se synchroniser pour exécuter certains services
 - ▶ utiliser pour offrir une solution de secours et pour répartir la charge

- **Activation des fonctionnalités HA :**

- ▶ l'activation du mode HA impacte la configuration des serveurs :
 - ▶ activation de sous-systèmes particuliers
 - ▶ paramétrages supplémentaires pour certains sous-systèmes
- ▶ le mode HA peut-être activé sur des serveurs 'standalone'
 - ▶ ne nécessite pas la mise en place d'un domaine

Activation des fonctionnalités HA

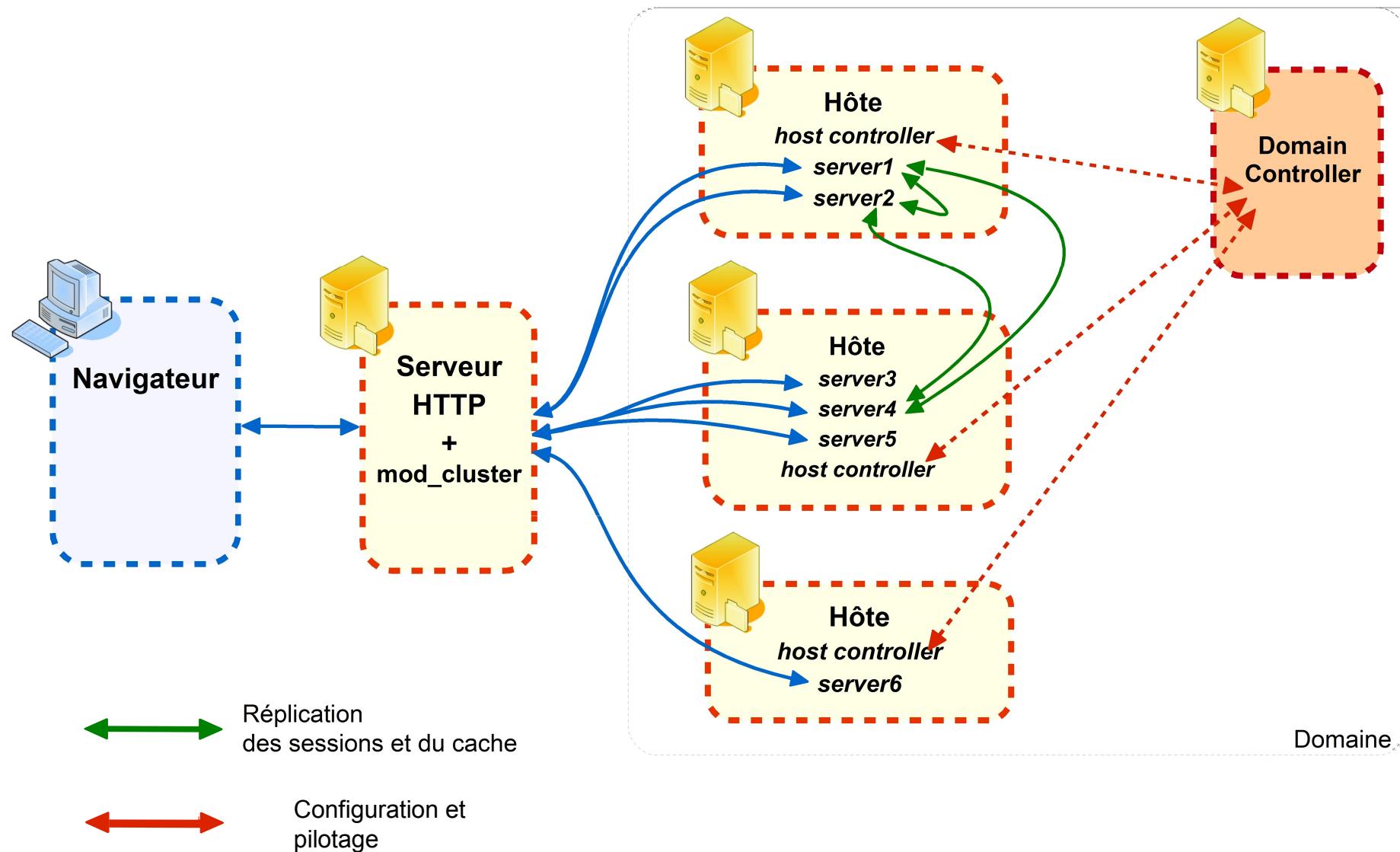
- L'activation du mode HA impacte fortement les fichiers de configuration
- Des fichiers préconfigurés sont fournis :
 - ▶ en mode standalone :
 - ▶ <JBOSS_HOME>/standalone/configuration/standalone-ha.xml
 - ▶ <JBOSS_HOME>/standalone/configuration/standalone-full-ha.xml
 - ▶ en mode domaine :
 - ▶ le fichier domain.xml contient des 'profiles' avec le HA activé ('ha' et 'full-ha')



Répartition de charge (1)

- **La répartition des requêtes HTTP entre plusieurs serveurs est un besoin classique :**
 - ▶ la répartition est assurée par un module installé dans le serveur HTTP
 - ▶ la solution privilégiée est l'utilisation du module nommé 'mod_cluster', nécessite l'utilisation de Apache HTTPD (packagé sous le nom de JBoss Enterprise Web Server dans le cadre de l'offre EAP)
 - ▶ d'autres possibilités, moins évoluées, permettent d'utiliser un serveur IIS ou iPlanet
 - ▶ l'activation du mode HA est notamment nécessaire pour que :
 - ▶ la réPLICATION des sessions soit opérationnelle
 - ▶ les serveurs puissent COMMUNIQUER avec le module mod_cluster

Répartition de charge (2)



Installation du serveur HTTP

- **Installation mod_cluster et Jboss sur la même machine**

- **Dans le fichier standalone-ha.xml modifier la configuration du sous-système mod_cluster :**

- ▶

```
<subsystem xmlns="urn:jboss:domain:modcluster:1.1">
    <mod-cluster-config proxy-list="127.0.0.1:6666"
        connector="ajp" advertise="false">
        <dynamic-load-provider>
            <load-metric type="busyness"/>
        ...
    </mod-cluster-config>
</subsystem>
```
- ▶ proxy-list indique à l'instance JBoss les infos pour se connecter au(x) serveur(s) Apache.

Installation du serveur HTTP

- Installation mod_cluster et Jboss sur des machines différentes
- Dans le fichier httpd.conf

▶ <IfModule manager_module> Listen
192.168.0.28:6666
ManagerBalancerName mycluster

```
<VirtualHost 192.168.0.28:6666>
<Location>
    Order deny,allow
    Deny from all
    Allow from
    192.168.0
</Location>
```

Installation du serveur HTTP

- **Deux solutions :**

- ▶ installer un serveur Apache et ajouter le module mod_cluster
- ▶ télécharger un serveur Apache préconfiguré avec mod_cluster :
 - ▶ via la page du projet mod_cluster : http://www.jboss.org/mod_cluster/downloads

The screenshot shows the Red Hat Customer Portal interface. At the top, there's a navigation bar with links for Home, Knowledge, Groups, Support, Downloads, Security, and Subscriptions. A search bar is also present. On the left, a sidebar lists various JBoss products: Enterprise Platforms (Web Server, Application Platform, BRMS Platform, Portal Platform, SOA Platform, Data Services Platform, Communications Platform, Enterprise Web Platform, Site Publisher, Data Grid), Enterprise Frameworks (Web Framework Kit, JBoss Hibernate, JBoss jBPM, JBoss Rules), and Developer Platform (JBoss Developer Studio). The main content area is titled "Software Downloads". It shows a message in a yellow box: "La page sélectionnée n'est pas encore disponible en français. Nous travaillons pour que tout le contenu soit disponible en français mais pour afficher cette page, nous sommes passés à l'anglais." Below this, there are dropdown menus for "Product" (set to "Application Platform") and "Version" (set to "6.0.0"). Underneath are tabs for "Releases (42)", "Patches (6)", and "Security Advisories (0)". A table lists software releases with columns for "Download File" (listing various Apache HTTP Server packages for different platforms like Windows Server 2008, Solaris, RHEL 6, etc.) and "Release Date" (all dated 06/12/2012). Each row has a "Download" link at the end.



Exemple

- **Lancement de deux serveurs en mode standalone sur la même machine avec le HA activé :**

- ▶ utilisation du décalage des numéros de port

```
./standalone.sh -Djboss.node.name=server1 --server-config=standalone-ha.xml  
-Djboss.socket.binding.port-offset=0
```

```
./standalone.sh -Djboss.node.name=server2 --server-config=standalone-ha.xml  
-Djboss.socket.binding.port-offset=100
```

- ▶ par défaut, les serveurs émettent des broadcasts, ils vont donc se découvrir automatiquement :

```
23:46:33,292 INFO [org.jboss.as.clustering] (ViewHandler,web,noeud2/web) JBAS010225: New  
cluster view for partition web (id: 1, delta: 1, merge: false) : [noeud2/web, noeud1/web]  
23:46:33,294 INFO [org.infinispan.remoting.transport.jgroups.JGroupsTransport]  
(ViewHandler,web,noeud2/web) ISPN000094: Received new cluster view: [noeud2/web|1]  
[noeud2/web, noeud1/web]
```

- ▶ Côté serveur HTTP les mêmes mécanismes de découverte automatique sont utilisés, la répartition de charge est effective sans actions supplémentaires

RéPLICATION de session

- Activation de la réPLICATION de session au niveau de chaque application :
 - ▶ <distributable/> dans le fichier web.xml, attribut standard
- La stratégie de réPLICATION peut être affinée :
 - ▶ via le fichier jboss-web.xml
 - ▶ détail EAP 6 Development guide, section 7.1

```
<jboss-web>
  <replication-config>
    <cache-name>custom-session-cache</cache-name>
    <replication-trigger>SET</replication-trigger>
    <replication-granularity>ATTRIBUTE</replication-granularity>
    <replication-field-batch-mode>true</replication-field-batch-mode>
    <use-jk>false</use-jk>
    <max-unreplicated-interval>30</max-unreplicated-interval>
    <snapshot-mode>INSTANT</snapshot-mode>
    <snapshot-interval>1000</snapshot-interval>
    <session-notification-policy>
      com.example.CustomSessionNotificationPolicy
    </session-notification-policy>
  </replication-config>
</jboss-web>
```

Activation et passivation de session

- Pour limiter la taille mémoire occupée par les sessions, le serveur peut en stocker une partie sur le disque :

- ▶ notions de passivation/activation
- ▶ configurable via le fichier jboss-web.xml
- ▶ web.xml

```
<jboss-web>
    <max-active-sessions>20</max-active-sessions>
    <passivation-config>
        <use-session-passivation>true</use-session-passivation>
        <passivation-min-idle-time>60</passivation-min-idle-time>
        <passivation-max-idle-time>600</passivation-max-idle-time>
    </passivation-config>
</jboss-web>
```

Sécurité

- Présentation de JAAS
- Domaines de sécurité
- Login Modules
- Utilisation de SSL

Présentation de JAAS

- **La sécurité Java EE est basée sur la notion de rôles :**

- ▶ le développeur indique que certaines ressources (EJB, servlets, ...) ne sont accessibles qu'aux utilisateurs ayant tel(s) rôle(s)
- ▶ l'administrateur configure le serveur d'applications pour lui donner accès au(x) registre(s) contenant les utilisateurs et leurs rôles
- ▶ le serveur d'applications vérifie que l'utilisateur courant à bien les rôles permettant d'accéder à la ressource demandée

- **Le service utilisé est une implémentation de JAAS :**

- ▶ Java Authentication and Authorization Service
- ▶ Gère les autorisations sur l'accès aux différentes ressources d'une application Redirige
- ▶ l'utilisateur vers un service de login au premier accès à une ressource protégée
- ▶ Les informations sur le registre à utiliser sont définies via la notion de LoginModule

Exemples côté développeur

- Sécurisation d'une ressource Web :

▶ fichier web.xml

```
...  
<security-constraint>  
    <display-name>SecurityConstraint</display-name>  
    <web-resource-collection>  
        <web-resource-name>WRCollection</web-resource-name>  
        <url-pattern>/accountList</url-pattern>  
        <http-method>GET</http-method>  
        <http-method>POST</http-method>  
    </web-resource-collection>  
    <auth-constraint>  
        <role-name>bankCustomer</role-name>  
    </auth-constraint>  
...  
...
```

- Sécurisation EJB :

```
@Stateful  
// Les rôles 'admin' et 'user' sont nécessaires pour accéder aux  
// méthodes de ce bean.  
@RolesAllowed({"admin", "user"})  
public class PanierDeCommandeBean implements PanierDeCommande {  
    private String nomClient;  
    private Collection produits;  
  
    // L'annotation @RolesAllowed est utilisable  
    // au niveau des méthodes  
    // Le niveau méthode écrase le niveau classe  
    @RolesAllowed("autreRôle") public  
        void validerCommande() {  
            // ...  
    }
```

```
@Stateless  
@DeclareRoles({"admin"})  
public class HelloBean implements Hello {  
    @Resource private SessionContext ctx;  
  
    public void test() {  
        //Vérifie que l'utilisateur possède le rôle indiqué  
        if(ctx.isCallerInRole("admin")) {  
            ...  
        }  
    }  
}
```

Domaines de sécurité (1)

- **Les informations de sécurité sont stockées dans les profils :**
 - ▶ Pour JBoss 7, mode standalone : standalone/configuration/standalone.xml
 - ▶ JBoss 7, mode domain : domain/configuration/domain.xml
- **Plusieurs stratégies de sécurité peuvent être configurées :**
 - ▶ Notion de 'domaine de sécurité'
 - ▶ Chaque domaine de sécurité spécifie les modules JAAS qui seront utilisés
- **Afin d'utiliser un domaine autre que celui par défaut, les développeurs peuvent spécifier le nom du domaine de sécurité :**

- ▶ Information dans le descripteur de déploiement JBoss
 - ▶ exemple pour un WAR, fichier WEB-INF/jboss-web.xml

```
<jboss-web>
    <security-domain>java:/jaas/my-domain</security-domain>
</jboss-web>
```

Domaines de sécurité (3)

- **Gestion par défaut :**

- ▶ la section 'management' des fichiers 'standalone.xml' et 'host.xml' définit un domaine de sécurité 'ApplicationRealm' qui est utilisé en dernier recours dans la configuration par défaut :
 - ▶ utilise les fichiers application-users.properties et application-roles.properties
 - ▶ commande add-user.sh permet d'ajouter des utilisateurs à ces fichiers

```
<management>
    <security-realms>
        <security-realm name="ManagementRealm">
            <authentication>
                <local default-user="$local"/>
                <properties path="mgmt-users.properties" relative-to="jboss.server.config.dir"/>
            </authentication>
        </security-realm>
        <security-realm name="ApplicationRealm">
            <authentication>
                <local default-user="$local" allowed-users="*"/>
                <properties path="application-users.properties" relative-to="jboss.server.config.dir"/>
            </authentication>
            <authorization>
                <properties path="application-roles.properties" relative-to="jboss.server.config.dir"/>
            </authorization>
        </security-realm>
    </security-realms>
</management>
```

Domaines de sécurité (4)

• Ajout d'un utilisateur applicatif avec la commande add-user.sh :

```
bruno@BLC4-Ubuntu:~/Produits/JBoss-EAP-6.0.0.GA/jboss-eap-6.0/bin$ ./add-user.sh
```

```
Quel type d'utilisateur souhaitez-vous ajouter ?  
a)Management User (mgmt-users.properties)  
b)Application User (application-users.properties)  
(a): b
```

```
Saisir les informations sur le nouvel utilisateur Domaine  
(ApplicationRealm) :
```

```
Nom d'utilisateur : user1 Mot de passe :
```

```
Saisir mot de passe à nouveau :
```

```
Quels rôles souhaitez-vous impartir à cet  
utilisateur ? (Veuillez saisir une liste séparée par des  
virgules, ou laisser vide) [ ]: Role1,Role2,Role3
```

```
L'utilisateur 'user1' va être ajouté pour le domaine 'ApplicationRealm'
```

```
Est-ce correct oui/non ? y
```

```
Utilisateur 'user1' ajouté au fichier '/home/bruno/Produits/JBoss-EAP-6.0.0.GA/jboss-eap-  
6.0/standalone/configuration/application-users.properties'
```

```
Utilisateur 'user1' ajouté au fichier '/home/bruno/Produits/JBoss-EAP-6.0.0.GA/jboss-eap-  
6.0/domain/configuration/application-users.properties'
```

```
Utilisateur 'user1' ajouté aux rôles Role1,Role2,Role3 dans le fichier '/home/bruno/Produits/JBoss-EAP-  
6.0.0.GA/jboss-eap-6.0/standalone/configuration/application-roles.properties'
```

```
Utilisateur 'user1' ajouté aux rôles Role1,Role2,Role3 dans le fichier '/home/bruno/Produits/JBoss-EAP-  
6.0.0.GA/jboss-eap-6.0/domain/configuration/application-roles.properties'
```

```
Is this new user going to be used for one AS process to connect to another AS process e.g. slave domain  
controller?
```

```
yes/no? n
```

Login modules (1)

- **Les Login Modules implémentent la stratégie d'authentification**
 - ▶ JBoss en propose plusieurs par défaut, qui peuvent être modifiés
- **Les Login Modules les plus utilisés**
 - ▶ UserRolesLoginModule
 - ▶ Donne accès à un ensemble de comptes d'utilisateurs définies dans un fichier xml
 - ▶ On peut définir le rôle de chaque utilisateur dans un autre fichier xml
 - ▶ IdentityLoginModule
 - ▶ Associe le même nom utilisateur (fixé sur le fichier xml) à tous les utilisateurs
 - ▶ DatabaseServerLoginModule
 - ▶ Permet de réaliser l'authentification à partir de la base de données
 - ▶ LdapLoginModule
 - ▶ authentification par LDAP
- **Les Login Modules peuvent être enchaînés**
- **On peut créer des Login Modules personnalisés**

Login modules (2)

- Implémentation de UserRolesLoginModule sur JBoss 7 :

```
<security-domain name="my-domain" cache-type="default">

    <authentication>

        <login-module code="UsersRoles" flag="required">

            <module-option name="usersProperties" value="${jboss.server.config.dir}/application-users.properties"/>

            <module-option name="rolesProperties" value="${jboss.server.config.dir}/application-roles.properties"/>

        </login-module>

    </authentication>

</security-domain>
```

```
# exemple de app-users.properties
# user=password
guest=guest
john=needle
```

```
# exemple de app-roles.properties
# user=role1,role2,... guest=guest,j2ee
john=guest,subscriber,publisher,durpublisher
```

Méthode d'authentification (1)

- **Le développeur peut choisir la façon dont l'authentification sera gérée côté client :**

- ▶ information dans le fichier web.xml
- ▶ plusieurs méthodes d'authentification prévues
 - ▶ DIGEST : mot de passe crypté
 - ▶ BASIC : authentification par boîte de dialogue
 - ▶ FORM : authentification par formulaire

- **Authentification DIGEST**

- ▶ Authentification par boîte de dialogue similaire à BASIC
- ▶ mot de passe est crypté (MD5)

Méthode d'authentification (2)

- **Authentification BASIC**

- ▶ Authentification par boîte de dialogue
- ▶ Le realm-name indique le nom de la zone protégée

```
...  
    <login-config>  
        <auth-method>BASIC</auth-method>  
        <realm-name>securityapp</realm-name>  
    </login-config>  
    <security-role>  
        <role-name>administrator</role-name>  
    </security-role>  
...
```



Méthode d'authentification (3)

• Authentification FORM

- ▶ Authentification par formulaire
- ▶ Il faut signaler les pages d'authentification et d'erreur dans le cas d'authentification par formulaire

```
<login-config>
    <auth-method>FORM</auth-method>
    <realm-name>Duke's Bank</realm-name>
    <form-login-config>
        <form-login-page>/logon.jsp</form-login-page>
        <form-error-page>/logonError.jsp</form-error-page>
    </form-login-config>
</login-config>
<security-role>
    <role-name>administrator</role-name>
</security-role>
```

```
<!-- Formulaire de login dans la page logon.jsp -->
<form method="POST" action="j_security_check">
    <table>
        <tr>
            <td>Login :</td>
            <td><input type="text" name="j_username"></td>
        </tr>
        <tr>
            <td>Mot de passe :</td>
            <td><input type="password" name="j_password"></td>
        </tr>
        <tr>
            <td><input type="submit" value="Entrer"></td>
            <td><input type="reset" value="Annuler"></td>
        </tr>
    </table>
</form>
```

TP 7.1



Conclusion