

# Introduction à Java EE

Olivier Cailloux

LAMSADE, Université Paris-Dauphine

Version du 24 octobre 2017

# Java EE

- Java EE ?

# Java EE

- Java EE ? Java Platform, Enterprise Edition

# Java EE

- Java EE ? Java Platform, Enterprise Edition
- JCP ?

# Java EE

- Java EE ? Java Platform, Enterprise Edition
- JCP ? Java Community Process

# Java EE

- Java EE ? Java Platform, Enterprise Edition
- JCP ? Java Community Process
- API ?

# Java EE

- Java EE ? Java Platform, Enterprise Edition
- JCP ? Java Community Process
- API ? Application Programming Interface

# Java EE

- Java EE ? Java Platform, Enterprise Edition
- JCP ? Java Community Process
- API ? Application Programming Interface

## Java EE

- [technologies](#)
- Spécifications, dont API
- Implémentation de référence



# Java EE : Processus

- Java EE fortement appuyée sur standards ouverts
- Standards du W3C / IETF ?

# Java EE : Processus

- Java EE fortement appuyée sur standards ouverts
- Standards du W3C / IETF ? [HTTP](#), [HTML](#), [XML](#), [WSDL](#), ...

# Java EE : Processus

- Java EE fortement appuyée sur standards ouverts
- Standards du W3C / IETF ? [HTTP](#), [HTML](#), [XML](#), [WSDL](#), ...
- JCP : implication de « la communauté » pour standards Java
- JCP définit les JSR : standards utilisés en Java SE ou Java EE
- JSR ?

# Java EE : Processus

- Java EE fortement appuyée sur standards ouverts
- Standards du W3C / IETF ? [HTTP](#), [HTML](#), [XML](#), [WSDL](#), ...
- JCP : implication de « la communauté » pour standards Java
- JCP définit les JSR : standards utilisés en Java SE ou Java EE
- JSR ? Java Specification Request

# Java EE : Processus

- Java EE fortement appuyée sur standards ouverts
- Standards du W3C / IETF ? [HTTP](#), [HTML](#), [XML](#), [WSDL](#), ...
- JCP : implication de « la communauté » pour standards Java
- JCP définit les JSR : standards utilisés en Java SE ou Java EE
- JSR ? Java Specification Request
- Spécifications tiennent compte de nombreux avis d'horizons divers
- Exemples ?

# Java EE : Processus

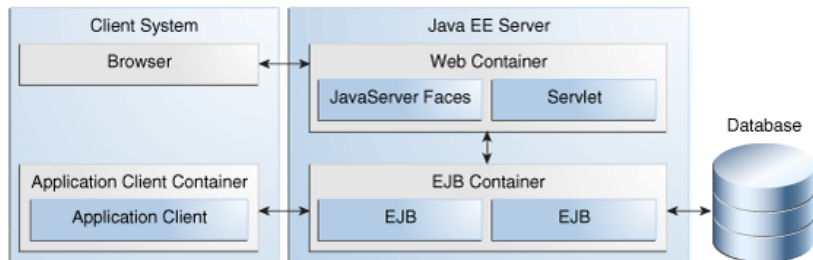
- Java EE fortement appuyée sur standards ouverts
- Standards du W3C / IETF ? [HTTP](#), [HTML](#), [XML](#), [WSDL](#), ...
- JCP : implication de « la communauté » pour standards Java
- JCP définit les JSR : standards utilisés en Java SE ou Java EE
- JSR ? Java Specification Request
- Spécifications tiennent compte de nombreux avis d'horizons divers
- Exemples ? JSR 221 : JDBC 4.0 ; JSR 338 : JPA 2.1 ; JSR 345 : EJB 3.2 ; JSR 342 : Java EE 7 ; JSR 346 : CDI...

# Java EE : Processus

- Java EE fortement appuyée sur standards ouverts
- Standards du W3C / IETF ? [HTTP](#), [HTML](#), [XML](#), [WSDL](#), ...
- JCP : implication de « la communauté » pour standards Java
- JCP définit les JSR : standards utilisés en Java SE ou Java EE
- JSR ? Java Specification Request
- Spécifications tiennent compte de nombreux avis d'horizons divers
- Exemples ? JSR 221 : JDBC 4.0 ; JSR 338 : JPA 2.1 ; JSR 345 : EJB 3.2 ; JSR 342 : Java EE 7 ; JSR 346 : CDI...
- Tensions entre standard ouvert et contrôle ! (2010, Apache [quitte](#) le comité JCP ; Doug Lea [également](#), en faveur de OpenJDK...)

# Conteneurs

- Un produit conforme Java EE fournit trois *conteneurs*
  - Conteneur EJB
  - Conteneur web
  - Conteneur application client
- Contenant des *composants* (du type adéquat)
- Chacun fournit des services pour le développeur
- Fournit l'accès aux API (différents conteneurs, différentes API)





# Composants

- *Composant* : une unité logicielle assemblée dans une application Java EE avec ses classes et fichiers liés et communiquant avec d'autres composants.
- Code Java compilé normalement
- Assemblé dans une application Java EE : peut utiliser les services ; doit se conformer aux spécifications
- Exécution gérée par le conteneur (pas de `main`, par exemple)

# Composant EJB

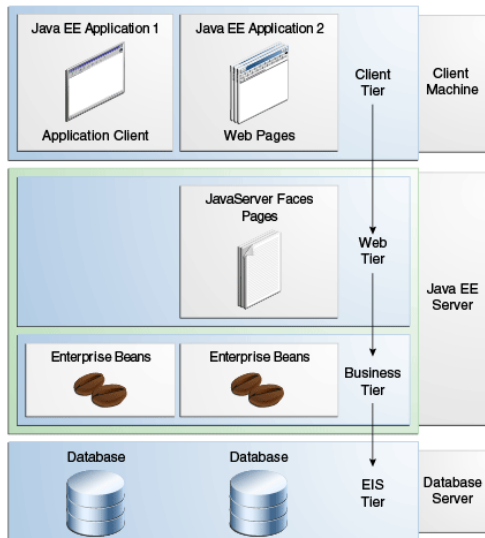
## EJB

- *Enterprise* Java Bean
  - Composant « business », sur le serveur
  - Service pouvant être appelé localement ou à distance
  - Deux types : session bean, message-driven bean
- 
- Le conteneur rend l'EJB accessible de l'extérieur
  - Permet le Remote Method Invocation, sorte de RPC
  - Le conteneur instancie, facilite la sérialisation, ...

# Composant Web

- Java Servlet n. m.
- JavaServer Faces

# Couches (« tier »)



- Ajout d'une couche multithread entre le client et le serveur classique
- Souvent :  
presentation, logic, data tier
- Couche web : peut être également appelé un client web (pourquoi?).

# Deux applications Java EE

## BDD $\Leftrightarrow$ EJB $\Leftrightarrow$ client

- entreprise A : niveau de stock calculable d'après BDD
- EJB : requête pour obtenir le niveau de stock
- composant client (fournisseur de A) : contacte l'EJB

## BDD $\Leftrightarrow$ EJB $\Leftrightarrow$ Servlet

- entreprise A : niveau de stock calculable d'après BDD
- EJB : requête pour obtenir le niveau de stock
- Web : servlet répondant à HTTP GET (client léger)
- Et puis ? Quel client final ?

Options pour client non-java ?

# Modules

- *Module* Java EE : fichier archive compressé
- Ensemble de composants pour un même conteneur (typiquement)
- Éventuellement : un descripteur de déploiement (`.xml`) pour ce type de conteneur (standard Java EE ou par produit)
- Éventuellement : des pages HTML statiques ; des classes utilité, ...
- Les descripteurs surchargent les annotations

# Module Web

## Module Web

- Fichier `.war`
- Fichiers `.class` servlets et autres dans `WEB-INF/lib` ou `WEB-INF/classes`
- Fichiers web statiques (`.html`, images, ...) dans `root`
- `WEB-INF/web.xml` : descripteur pour conteneur Web
- `META-INF/glassfish-web.xml` : descripteur pour glassfish
- `META-INF/MANIFEST.MF`

# Assemblage et déploiement

- Application Java EE composée d'un ou plusieurs modules
- On peut déployer un module seul (.war, .jar)
- Ou assembler les modules dans un fichier Enterprise Archive (.ear)
- EAR : plusieurs modules et év. descripteur d'application (META-INF/application.xml, META-INF/glassfish-application.xml)

## Déploiement

- Procédure dépend du serveur d'application Java EE
- Typiquement : déplacer l'archive (.war, .ear, .jar) dans un répertoire du serveur
- Accès depuis l'environnement de développement via plug-ins



# Services

Exemples :

- Managed beans
- CDI
- RestFul
- JSF
- Bean validation
- JAXB, JAX-WS, JNDI (aussi dans Java SE)

# GlassFish Server Tools

- Démarrer, arrêter le serveur
- Déployer des paquets
- Application : console d'administration
- Base de données

## À vous

- « Installez » GlassFish (copie depuis `/usr/local/glassfish-4.1/glassfish`)
- Démarrez votre serveur (cf. bin/, <http://localhost:8080>, <http://localhost:4848>)
- Désactivez l'écoute extérieure
- Lisez les logs

# Licence

Cette présentation, et le code LaTeX associé, sont sous [licence MIT](#). Vous êtes libres de réutiliser des éléments de cette présentation, sous réserve de citer l'auteur.

Le travail réutilisé est à attribuer à [Olivier Cailloux](#), Université Paris-Dauphine.

# Le terme Java

Terme *Java* adopté en 1995 (“as an example of yet another name that would never work”) (source: [Java World](#))



# Le terme Java

Terme *Java* adopté en 1995 (“as an example of yet another name that would never work”) (source: [Java World](#))



# A jar full of Java beans, please

- JAR File : introduits à la version 1.1. Une collection de fichiers `.class`.
- Java Bean (aussi version 1.1). ([specs](#)) : un composant logiciel pour assemblage (par exemple, un bouton AWT, une feuille de calcul à placer dans un document).



## Fun fact

Voyons le nombre magique des fichiers `.class`...

# Servlet et session HTTP

- Un servlet s'exécute dans un contexte
- Entre autres, une session HTTP
- Objet `HttpSession` rendu disponible par conteneur (injection ou accès via paramètres servlet)
- Le serveur tente de tracker la session via un cookie par exemple
- Le serveur fait expirer le cookie après un temps configurable
- Un bon site fonctionne même sans cookies !
- Classe implémente `HttpSessionListener` pour recevoir notifications `sessionCreated`, `sessionDestroyed`.  
L'annoter `@WebListener`. (Pattern ?)

# Servlet et session HTTP

- Un servlet s'exécute dans un contexte
- Entre autres, une session HTTP
- Objet `HttpSession` rendu disponible par conteneur (injection ou accès via paramètres servlet)
- Le serveur tente de tracker la session via un cookie par exemple
- Le serveur fait expirer le cookie après un temps configurable
- Un bon site fonctionne même sans cookies !
- Classe implémente `HttpSessionListener` pour recevoir notifications `sessionCreated`, `sessionDestroyed`.  
L'annoter `@WebListener`. (Pattern ? Observateur !)