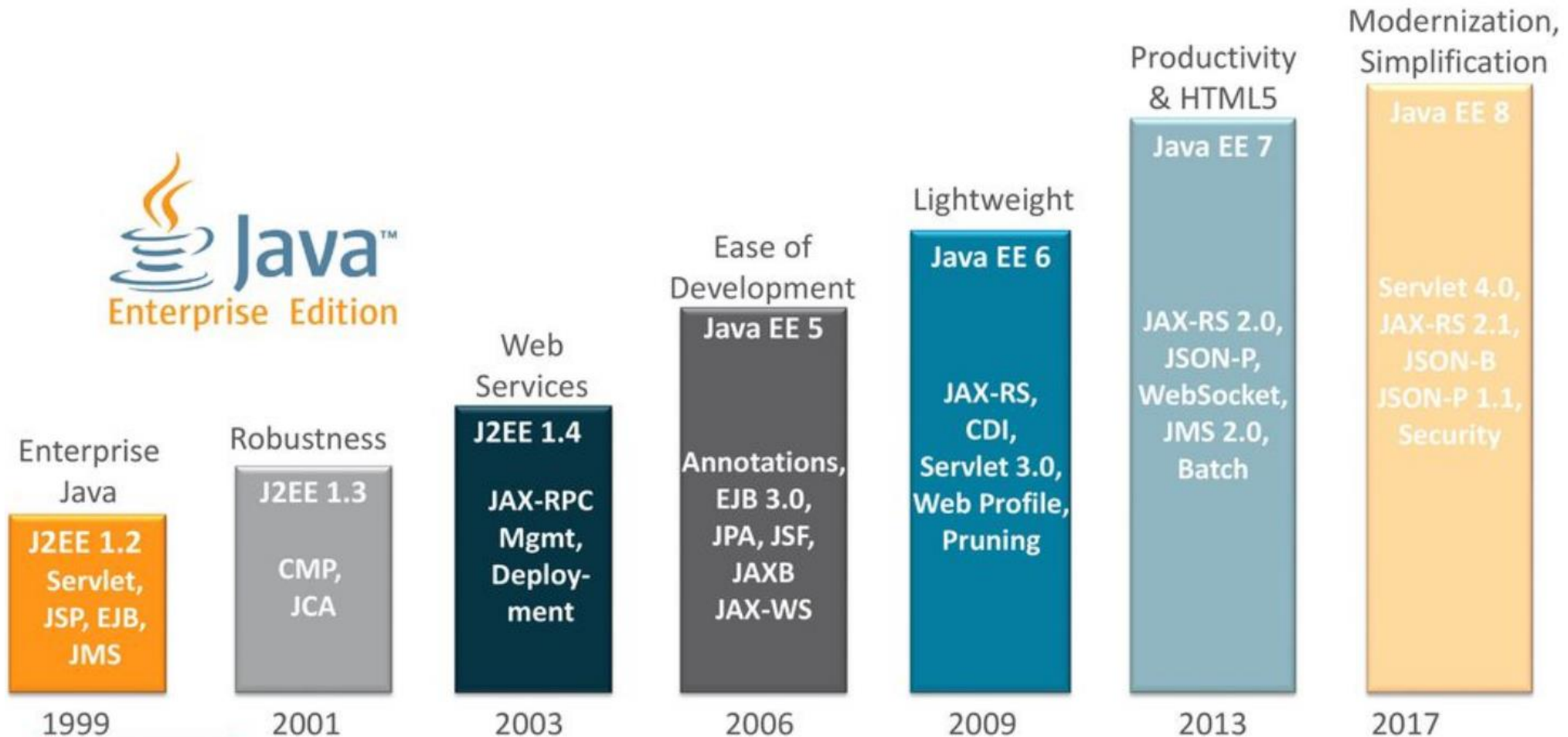


# Introduction à Java EE 8

Enseignant: Zouheir Ouni

Classe : CCDAD.1

# Historique



# Qu'est-ce que Java EE

**Java EE : Java Enterprise Edition (J2EE)**

---

## 1. Une technologie

outils liés au langage Java  
+ des spécifications

ET

## 2. Un modèle de développement

applications découpées en tiers

# Qu'est-ce que Java EE

- **Une technologie:**

---

- Le langage Java
- La machine virtuelle (JVM)
- Des APIs (le JDK + APIs applicatives)
- Des serveurs respectant le standard Java EE (JSR)



**Plate-forme Java EE**

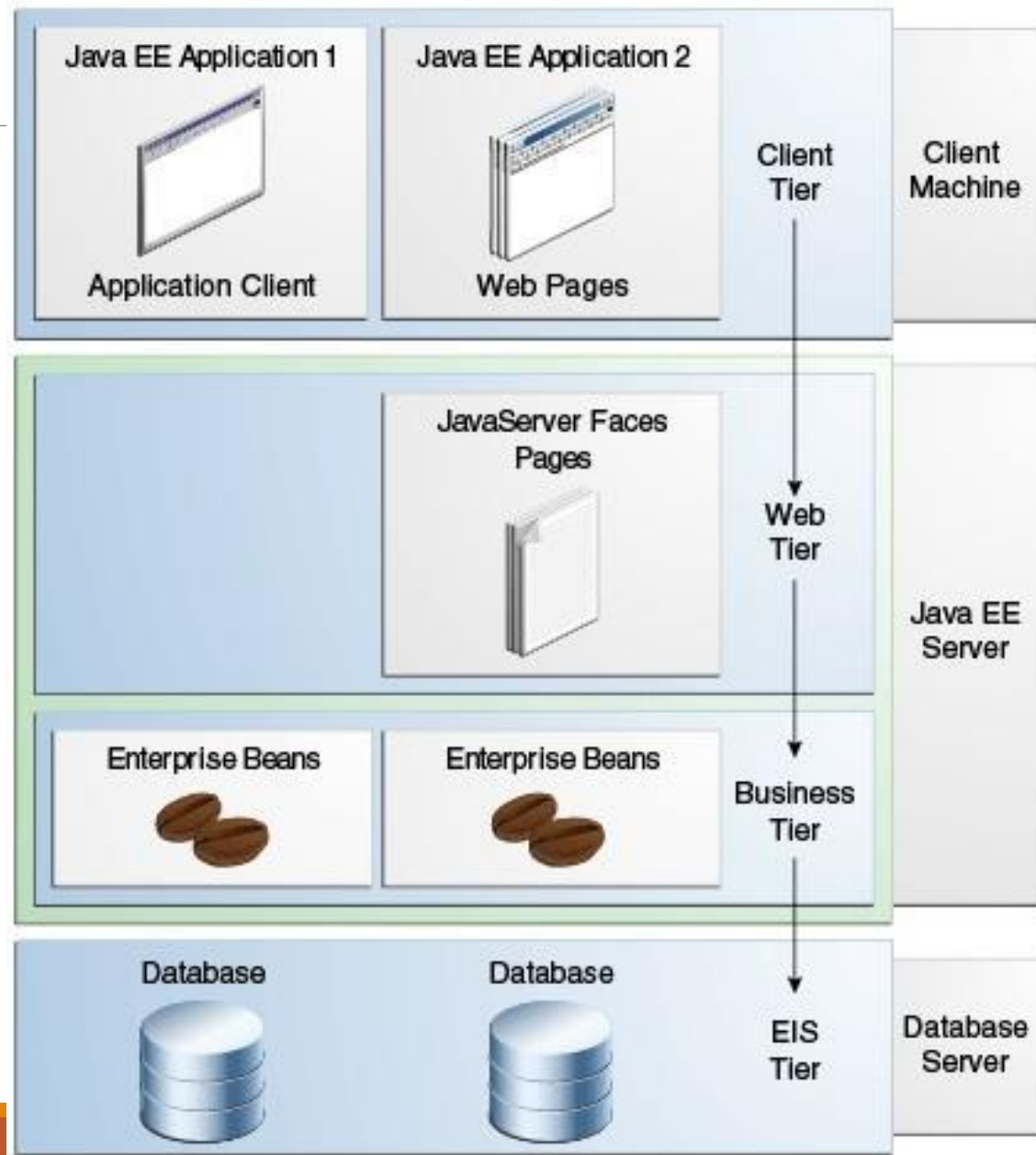
# Qu'est-ce que Java EE

- **Un modèle de développement:**

---

  - Développement en **tiers (multitiers)** : applications découpées logiquement (correspondance avec le déploiement : clients, serveurs, SGBDs,...)
  - Ce modèle partitionne le travail en 2 parties :
    - Les **aspects métiers/présentation**, à la charge du développeur
    - Les services standards fournies par la plate-forme Java EE

# Applications multi-tiers



# Applications multi-tiers

- Les composants d'une application Java EE sont considérées suivant 4 tiers :
  - **Client-tier** : partie tournant sur le client.
  - **Web-tier** : sur le serveur Java EE.
  - **Business-tier** : sur le serveur Java EE.
  - **Enterprise information system (EIS)-tier** : le logiciel appartenant au système d'information et s'exécutant sur le serveur correspondant (EIS server).

# Applications multi-tiers

- Les applications Java EE sont considérées comme des **applications 3-tiers** car elles sont distribuées sur **3 localisations** (virtuelles) différentes :
  - les machines clientes
  - le serveur Java EE
  - les systèmes d'informations (Bds, etc.)
- C'est donc une extension du modèle 2-tiers classique client/serveur : *ajout d'une couche applicative entre client et SIs*



---

# Terminologie Java EE

***Java EE-components et Java EE Containers***

---

# *JavaEE-components*

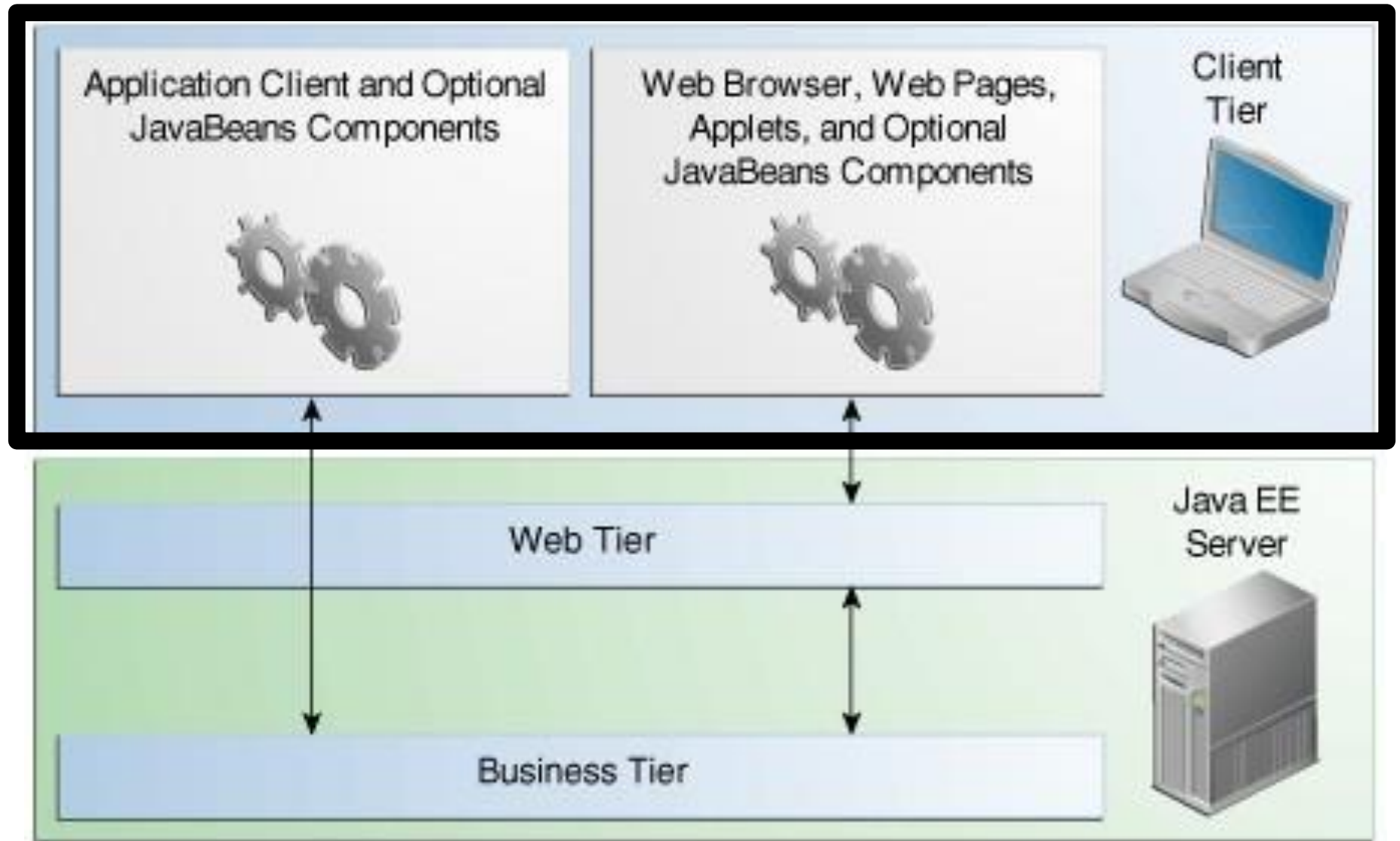
# Java EE-components

- Java EE distingue 3 types de composants:
  - **I : Les applications clientes et les applets** : des composants qui tournent sur le **client**.
  - **II : Java Servlet, JavaServer Faces, et les JavaServer Pages (JSP)** : des composants qui tournent sur le **serveur**.
  - **III : Les Enterprise JavaBeans (EJB)** : également sur le serveur
- Différences avec des classes classiques :
  - **Vérifient la spécification Java EE**
  - **Déployées sur un serveur Java EE**

# I. Les clients Java EE

- On distingue 2 types de clients Java EE :
  - (1) les **clients Web**
  - (2) les **applications clientes.**

# I. Les clients Java EE



# Clients Web

- **Un Client Web est considéré suivant 2 parties:**
  - 1) des pages web dynamiques générées par le Web-tier
  - 2) un navigateur qui affiche les pages générées
- On parle de **client léger** (thin client) : toutes les opérations complexes sont exécutées par le serveur

# Applets

- Une page web reçue depuis le web-tier peut contenir une applet : une petite application cliente écrite en Java exécutée par le navigateur
  - Nécessite un plugin contenant une JVM
  - Et parfois un fichier contenant des règles de sécurité
- On leur préfère aujourd'hui les composants web :
  - pas besoin de plugin ou de fichier particulier sur le client
  - Séparation nette entre le design de la page et les couches applicatives.

# Applications clientes

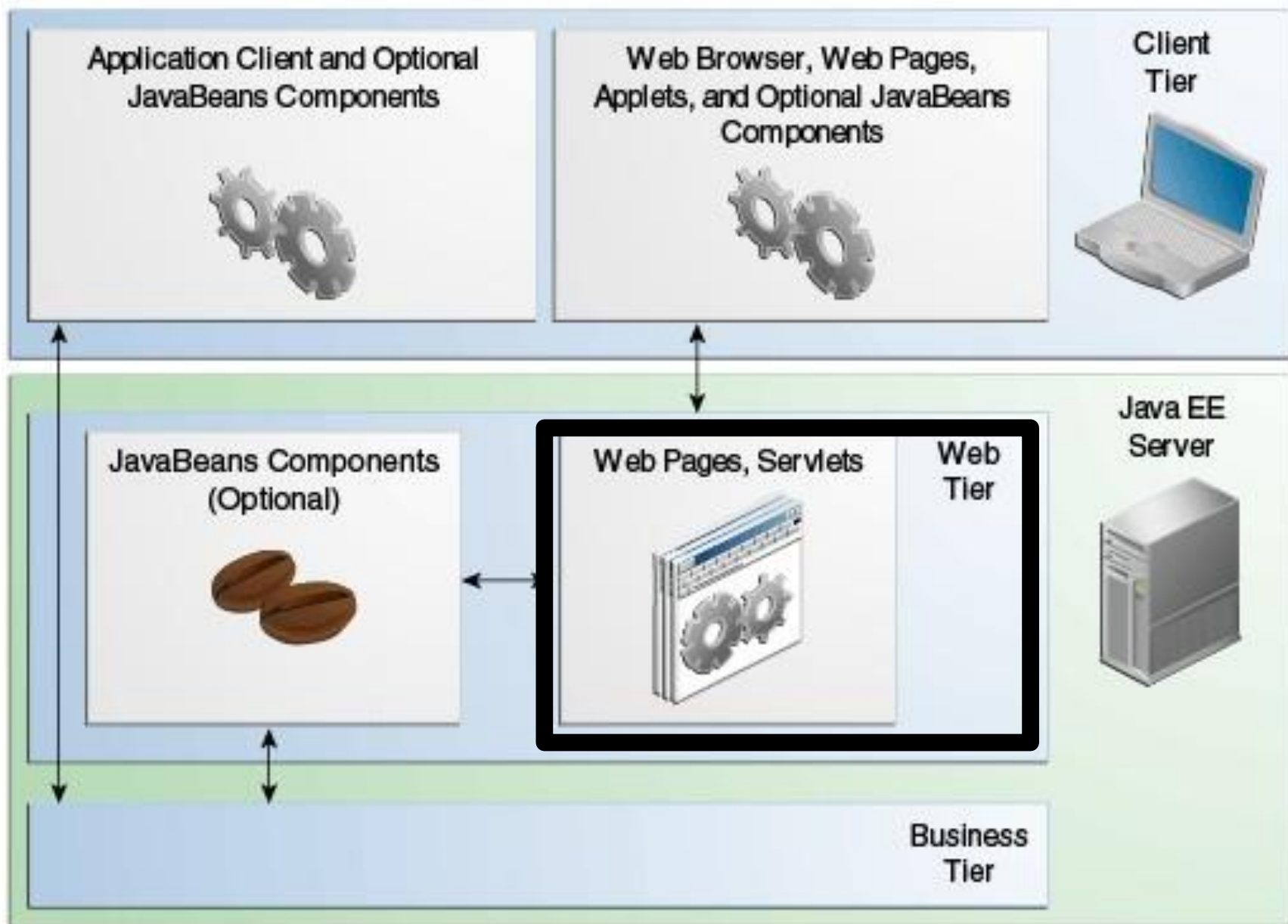
- Applications clientes (sur la machine cliente) :
  - plus riches en terme d'interface utilisateur (swing, etc.)
  - accèdent directement au business-tier
  - mais peut aussi accéder à des services fournis par le web-tier et interagir avec d'autres composants web
- On parle de **clients lourds** (ou thick clients)



# Les clients Java EE

- Choisir entre client léger ou lourd dépend du type d'applications et du contexte de déploiement :
- client léger :
  - facilite la distribution, le déploiement et la maintenance de l'application
  - mais limite l'expérience utilisateur (latence réseau, GUI limité, etc.)

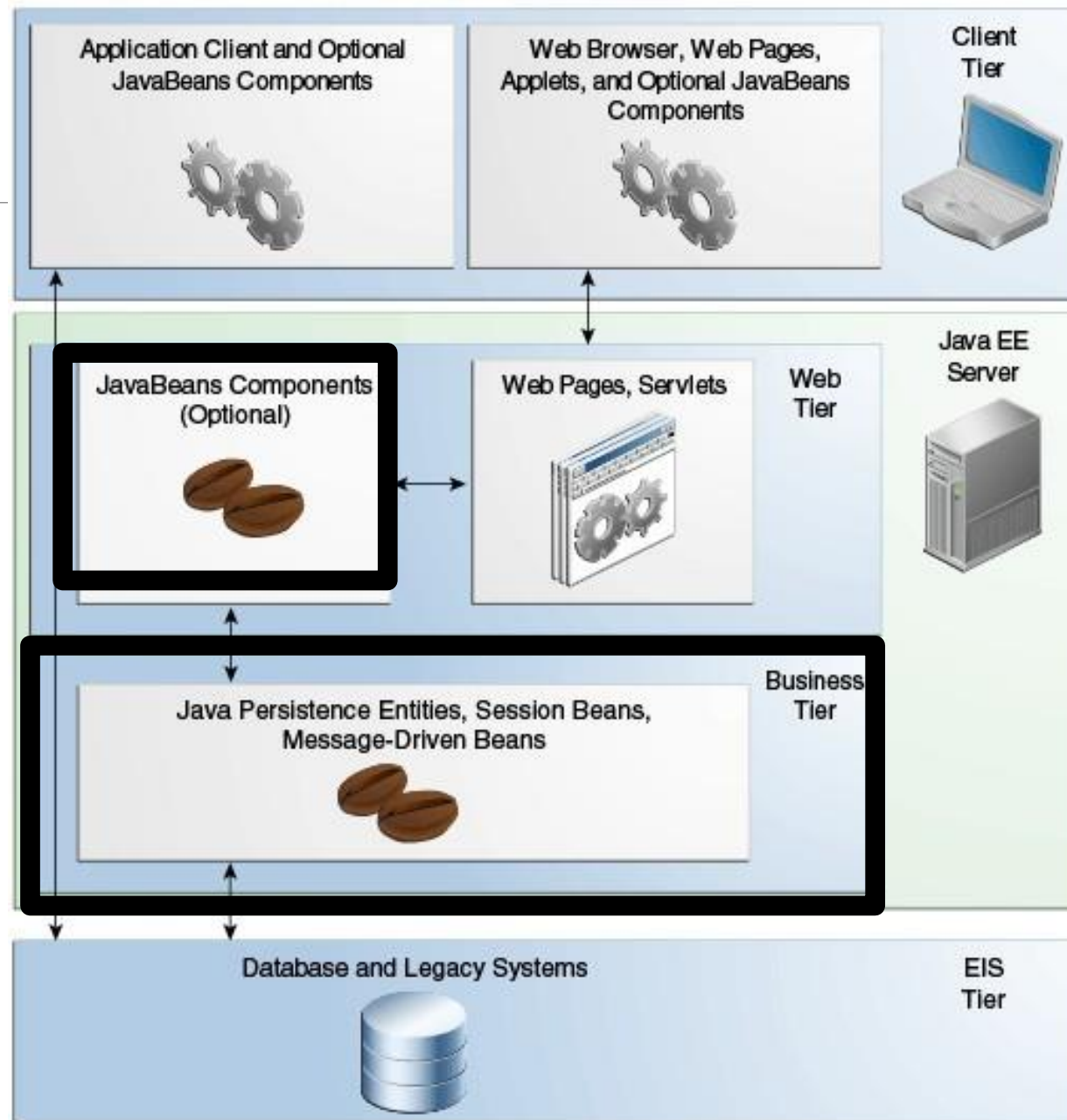
# II. Les composants Web Java EE



# Les composants Web Java EE

- Les composants Web Java EE sont soit :
  - des **Servlets** : des classes générant des pages web
  - des **JSP** : html statique + appels Java (« à la PHP »)
- *JavaServer Faces (JSF)* est une technologie utilisée par les JSP qui fournit des outils liés à la gestion des interfaces web
- Note : Les pages html et les applets ne sont pas considérées comme des composants Web par la spécification Java EE

# III. Business components



# Business-tier et EIS-tier

## • **Business tier**

---

Les parties liées au domaine d'application (les **composants métiers**) sont appelés des **Business components** (ils sont localisés dans le **Business-tier** ou le **web-tier**)

## • **EIS-tier**

Le tier lié au **système d'information** (de l'entreprise) concerne tout le logiciel lié à l'infrastructure de l'entreprise (ERP, système transactionnel, base de données, etc.)

# Example page JSP

---

```
<!DOCTYPE html>
<html>

<head>

<title>Insert title here</title>

</head>

<body>

  <h1>Hello JSP</h1>

  <%    java.util.Date date = new java.util.Date();    %>

  <h2>    Now is    <%=date.toString()%>    </h2>

</body>

</html>
```

# Ma première Servlet

---

```
ServletOutputStream out = response.getOutputStream();
```

```
out.println("<html>");
```

```
out.println("<head><title>Hello Servlet</title></head>");
```

```
out.println("<body>");
```

```
out.println("<h3>Hello World</h3>");
```

```
out.println("This is my first Servlet");
```

```
out.println("</body>");
```

```
out.println("<html>");
```

---

# *JavaEE-Containers*



# Les conteneurs Java EE

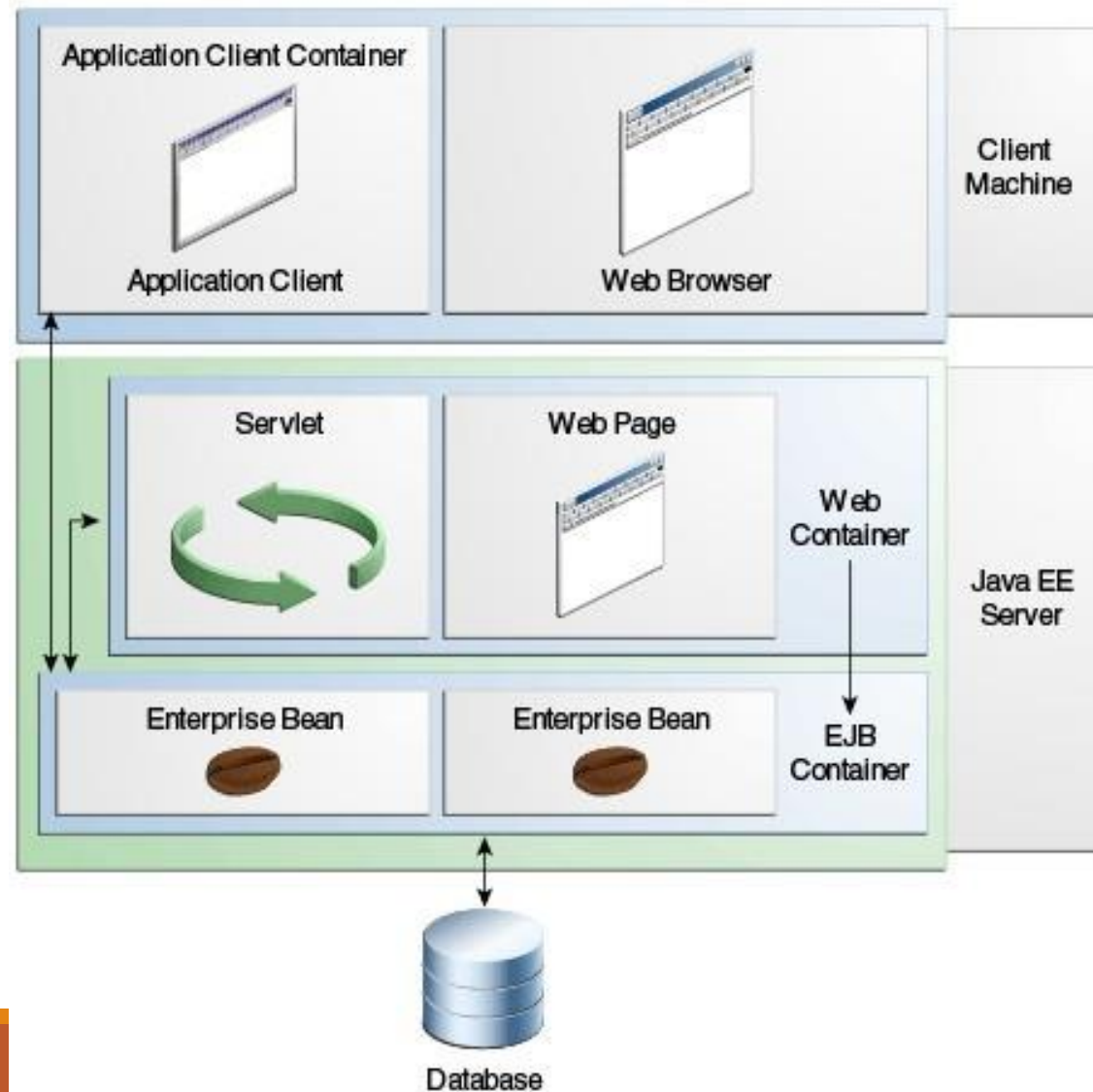
---

- Le **découpage en composants** est le premier point fort de la technologie Java EE.
- Le deuxième est la **gestion par conteneur des composants** : pour chaque type de composants, le serveur Java EE définit un conteneur qui fournit les services associés.

→ À chaque *component* son *container*

# Les types de conteneurs Java EE

- Le déploiement consiste à placer les composants dans les conteneurs adaptés :



# Les conteneurs Java EE

- Un conteneur (**container**) est l'interface entre le composant et les services de bas niveaux nécessaires
- Pour pouvoir être exécuté, un composant / application web doit être :
  - (1) **assemblé dans un module Java EE**
  - (2) **déployé dans son conteneur.**

# Les conteneurs Java EE

- Le processus d'assemblage consiste principalement à paramétrer les services fournis par le conteneur :
  - Pour chaque composant
  - Et pour l'application elle-même.
- Par ex., il s'agit de définir les services de sécurité (login par ex.), d'espace de nommage, etc.

---

# Packaging d'une application Java EE

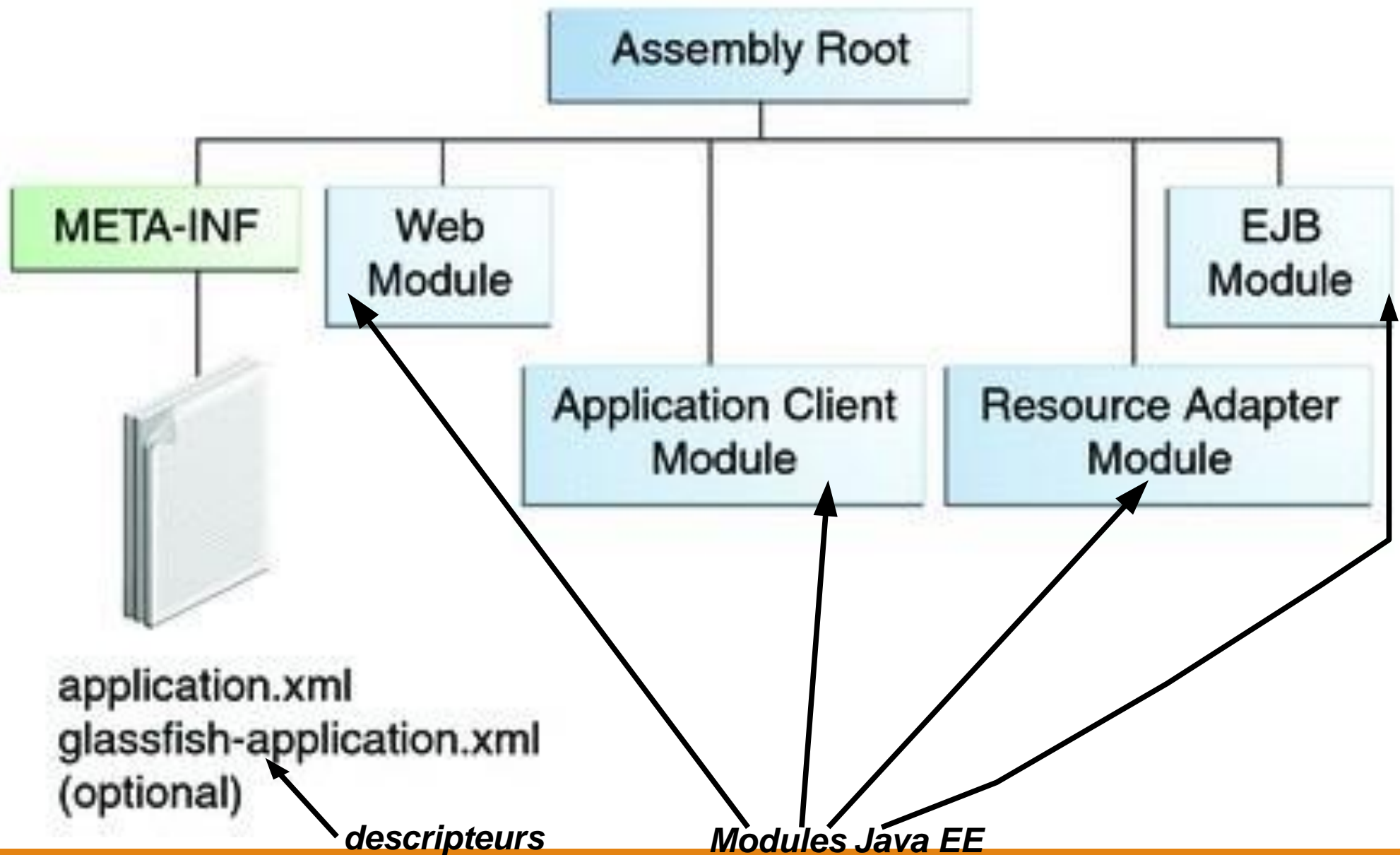
# Pour le déploiement

- Une application Java EE est donc composée d'un ensemble d'unités de programmation qui pourront ensuite être déployées sur n'importe quelle plateforme compatible avec les spécifications Java EE.
- **Chaque unité contient :**
  - Un ou plusieurs composants fonctionnels (ejb, pages JSP, servlet, etc.)
  - Des descripteurs de déploiement (optionnels) qui spécifient le contenu du composant.
- On parle ainsi du packaging d'une application

# Les fichiers EAR

- Une application Java EE est distribuée sous forme d'un fichier *Enterprise Archive* (EAR)
- C'est en fait un simple fichier jar avec l'extension .ear
- Ce fichier contient :
  - des modules Java EE (.jar, .war, .rar)
  - Les descripteurs de déploiement (des fichiers xml).  
Ils peuvent donc être modifiés sans toucher le code de l'application.
- À l'exécution, le serveur lit les descripteurs pour utiliser les composants de manière adéquate.

# Les fichiers EAR





# Les descripteurs

- Il existe 2 types de descripteurs :
  - Les **Java EE deployment descriptor** : peuvent être utilisés pour configurer les paramètres de déploiement du module
  - Les **runtime deployment descriptor** : utilisé pour paramétrer l'environnement hôte, c'est-à-dire n'importe quelle plate-forme compatible avec la norme Java EE.
  - Par exemple, pour la plate-forme *Sun Java System Application Server Platform Edition 9*, le runtime descriptor définit des informations comme la racine du répertoire web.
  - Les noms de ces fichiers sont standards :  
**sun-moduleType.xml** (dans le répertoire META-INF)

# Module Java EE

- Un module Java EE :
  - Un ou plusieurs composants Java EE prévus pour un type particulier de conteneur et un descripteur de déploiement du type correspondant.
  - Par exemple, le descripteur de déploiement d'un module de type enterprise bean définit, entre autres, les droits d'accès aux beans correspondants