Java pour le Web

Jakarta EE

ex Java EE


Java EE 8
Jakarta EE 8
Jakarta EE 9
Jakarta EE 9.1
Jakarta EE 10
August 2017
September 2019
December 2020
May 2021
September 2022
Oracle Infrastructure
JCP Process
Javax.* Namespace
Eclipse Foundation Infrastructure
new Jakarta EE Specification Process (JESP)
new Licensing & new Logo
new jakarta.* Namespace
Java SE 8 Support
removal of deprecated Specs
Java SE 11 Support
multiple compatible Implementations on the release date available*
Jakarta EE Core Profil
Jakarta CDI Lite
Independent individual Spec developmenzt

Qu'est-ce que Jakarta EE

- Jakarta EE : **Jakarta Entreprise Edition** (ex 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 Jakarta EE

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

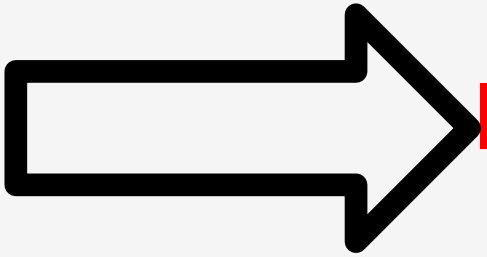
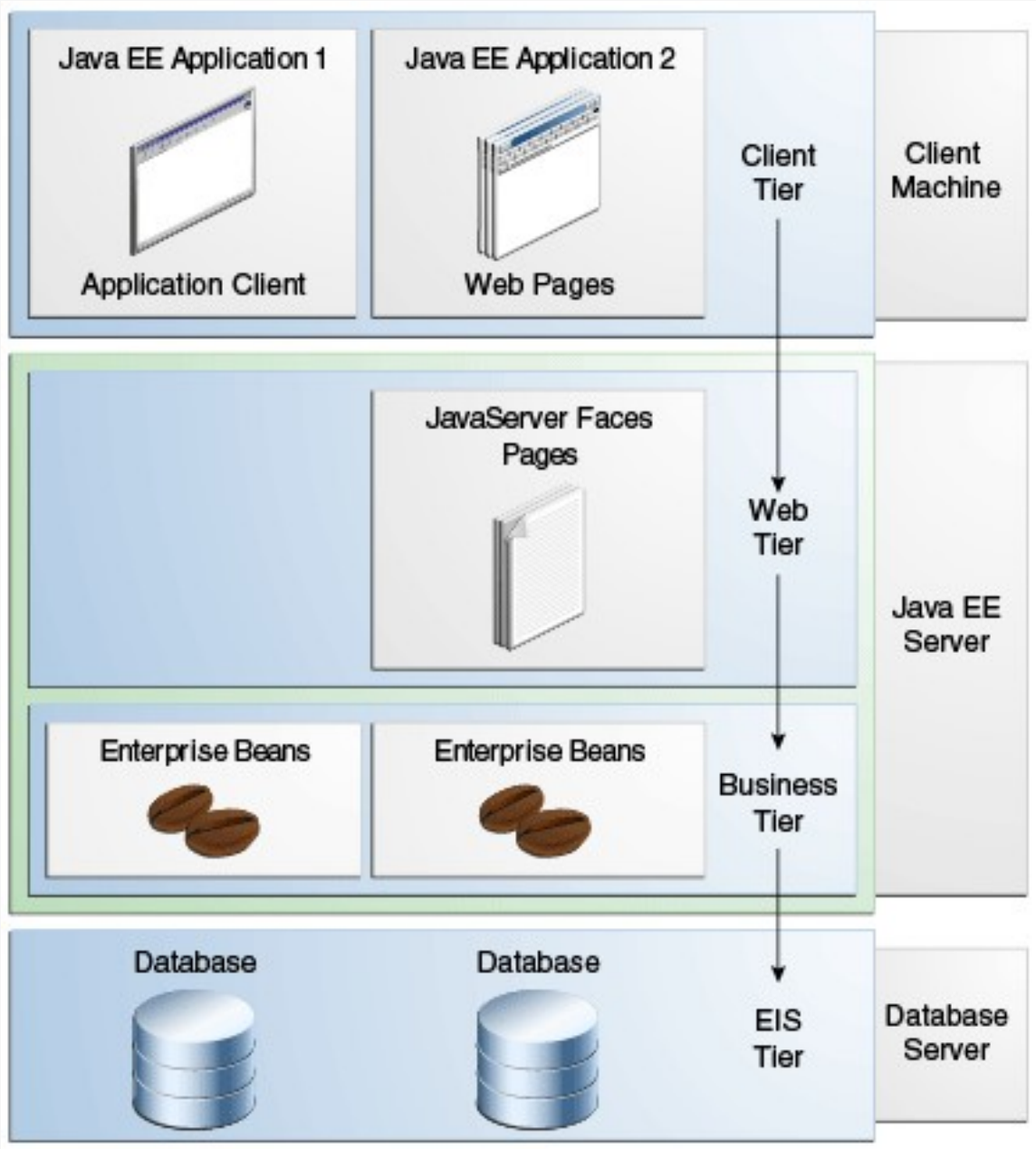


Plate-forme Jakarta EE

Qu'est-ce que Jakarta 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 Jakarta EE

Applications multi-tiers



Applications multi-tiers

- Les composants d'une application Jakarta EE sont considérées suivant 4 tiers :
  - **Client-tier** : partie tournant sur le client.
  - **Web-tier** : sur le serveur Jakarta EE.
  - **Business-tier** : sur le serveur Jakarta 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 Jakarta EE sont considérées comme des **applications 3-tiers** car elles sont distribuées sur **3 localisations** (virtuelles) différentes :
  - les machines clientes
  - le serveur Jakarta 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 Jakarta EE

Jakarta EE-components* et *Jakarta EE Containers

Jakarta EE-components

Jakarta EE-components

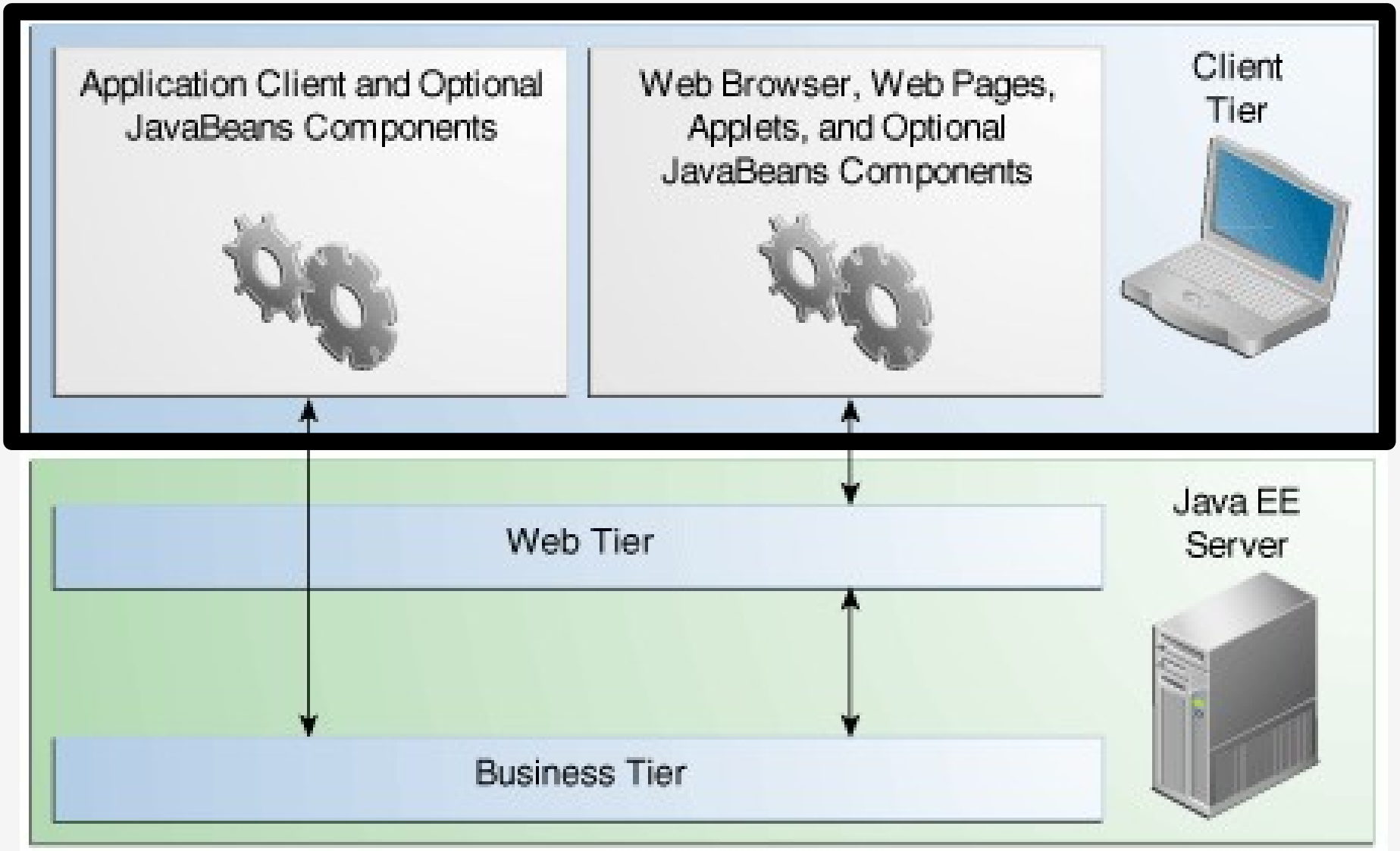
- Jakarta EE distincte 3 types de composants:
  - **I : Les applications clientes et les applets** : des composants qui tournent sur le **client**.
  - **II : Java Servlet, JavaServer Faces, et les JavaServer Pages (JSP)** : des composants qui tournent sur le **serveur**.
  - **III : Les Enterprise JavaBeans (EJB)** : également sur le serveur

- Différences avec des classes classiques :
  - **Vérifient la spécification Jakarta EE**
  - **Déployées sur un serveur Jakarta EE**

I. Les clients Jakarta EE

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

I. Les clients Jakarta 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

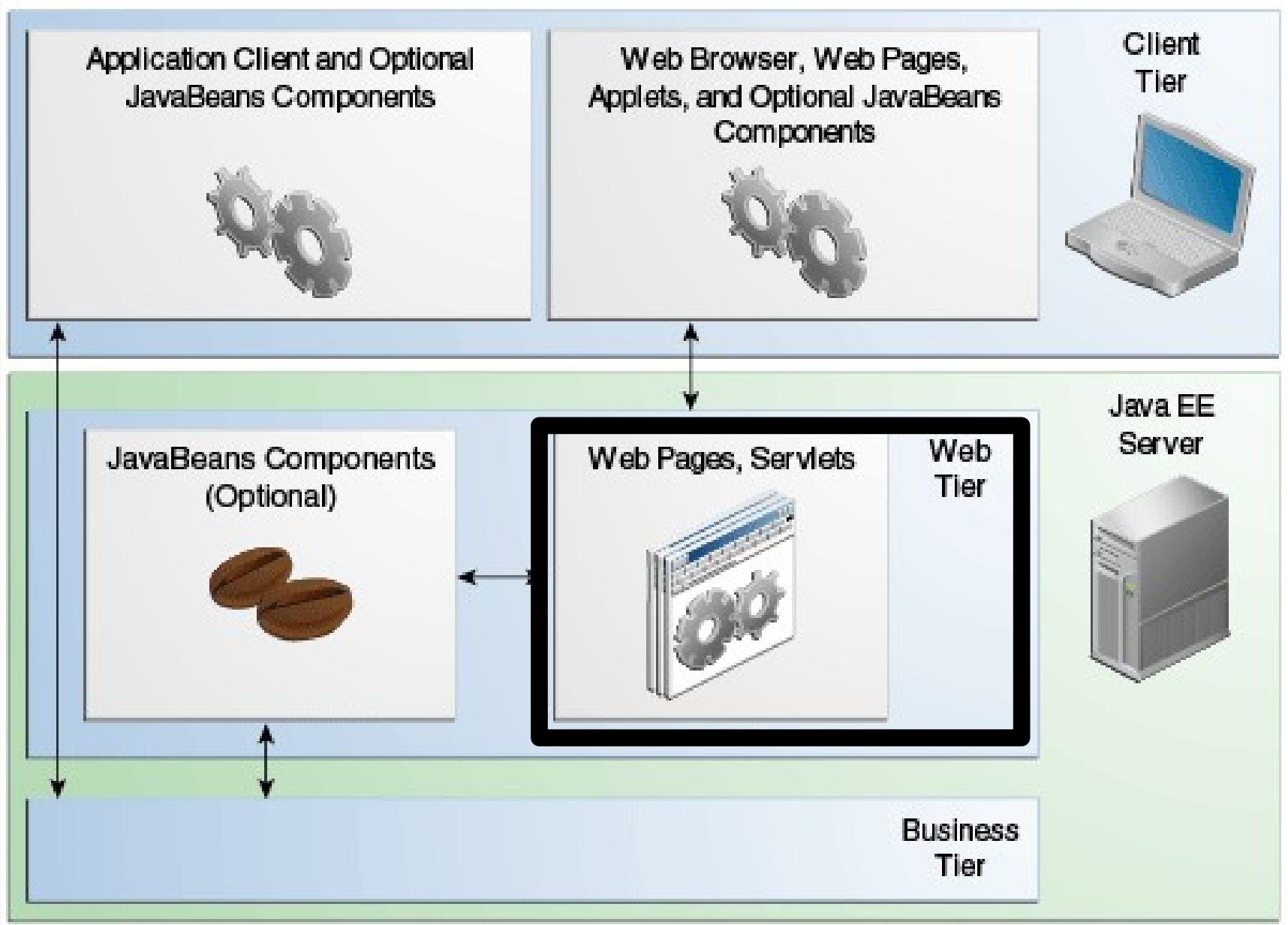
- Applications clientes (sur la machine cliente) :
  - plus riches en terme d'interface utilisateur (swing, etc.)
  - accèdent directement au business-tier
  - mais peut aussi accéder à des services fournis par le web-tier et interagir avec d'autres composants web
- On parle de clients lourds (ou thick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

Les clients Jakarta 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 Jakarta EE



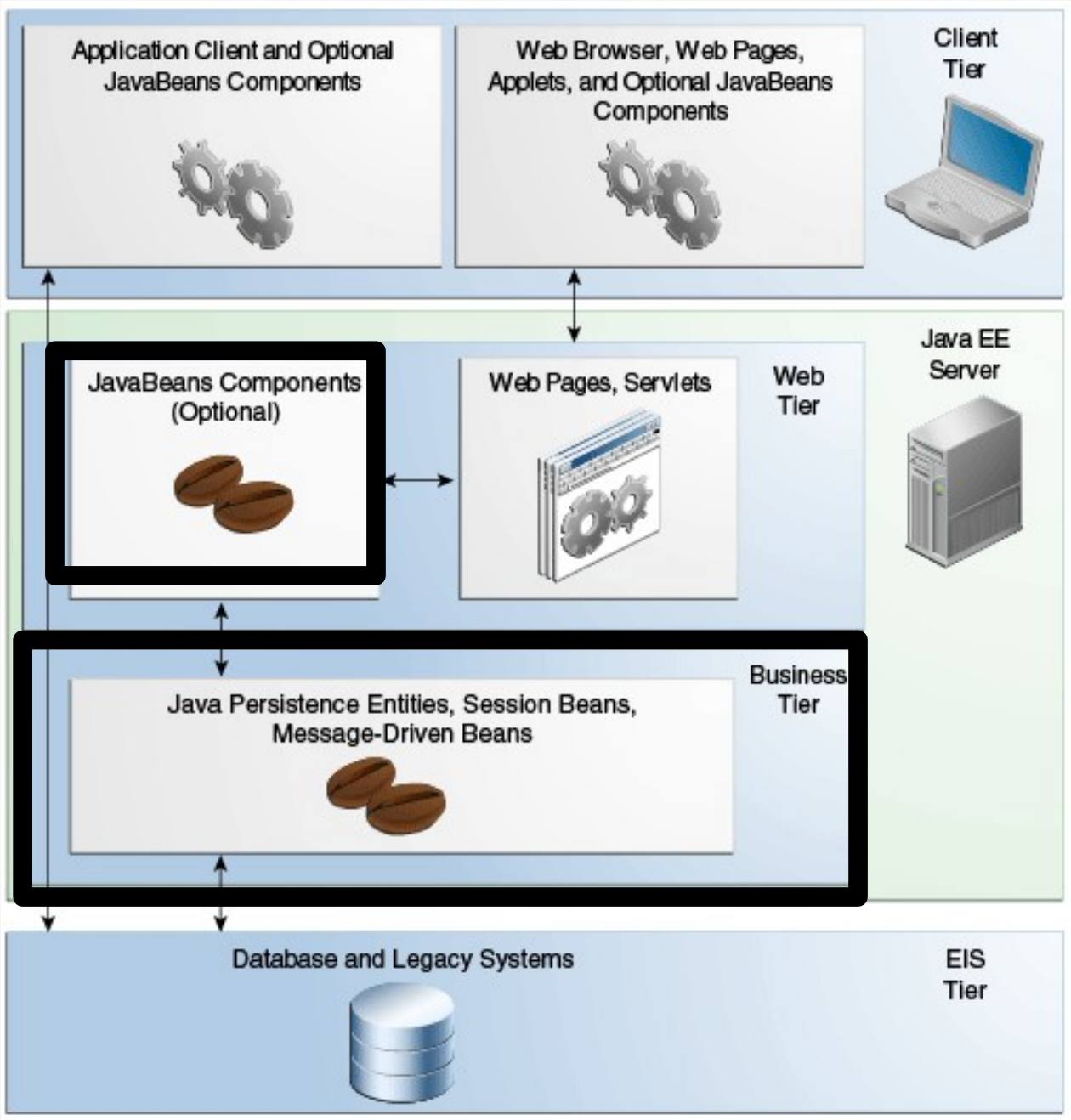
Les composants Web Jakarta EE

- Les composants Web Jakarta 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 techno utilisée par les JSP qui fournit des outils liés à la gestion des interfaces web (analogie composants swing/MVC).

- Note : Les pages html et les applets ne sont pas considérées comme des composants Web par la spécification Jakarta 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.)

Jakarta EE-Containers

Les conteneurs Jakarta EE

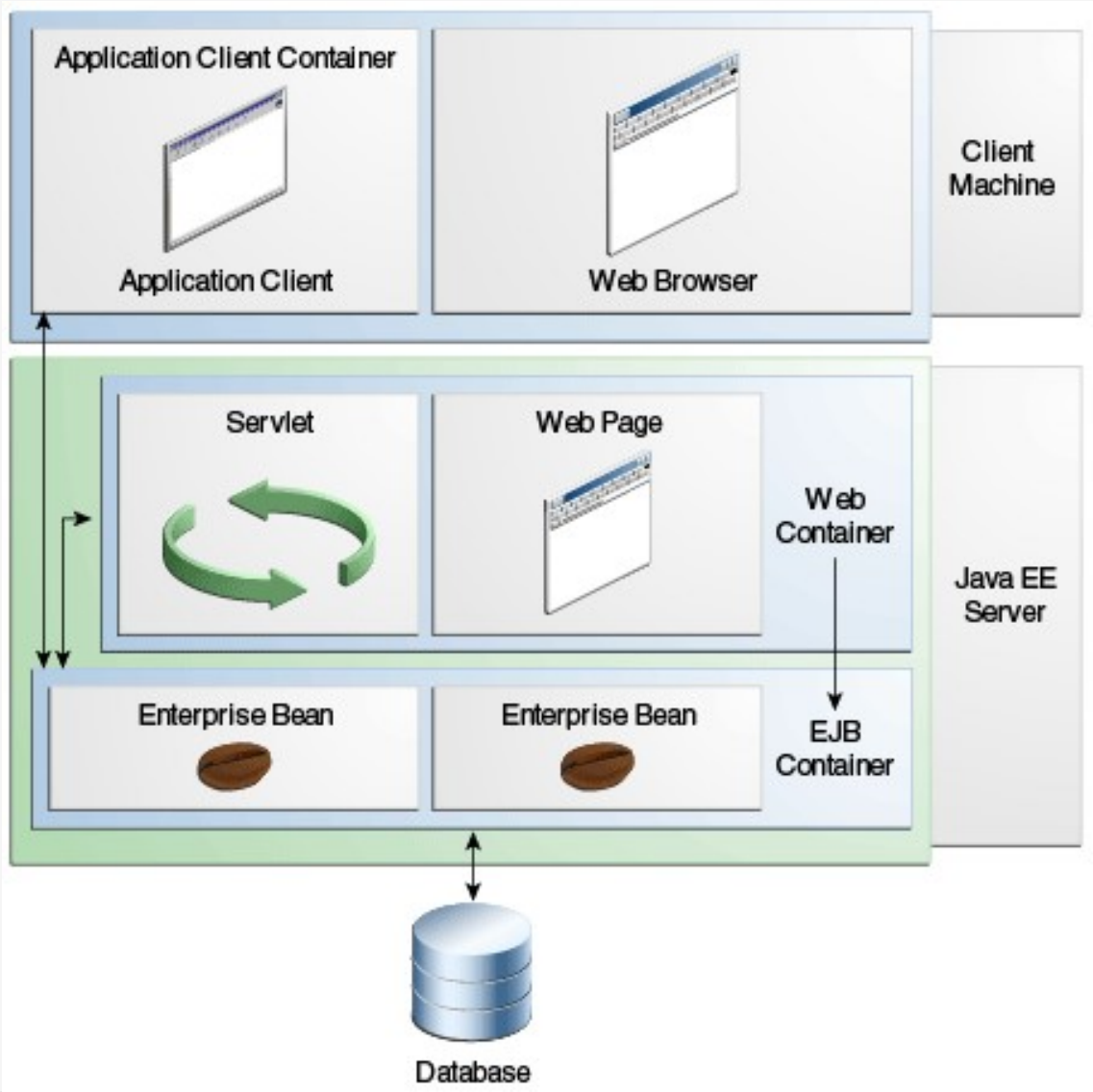
- Le **découpage en composants** est le premier point fort de la technologie Jakarta EE.

- Le deuxième est la **gestion par conteneur des composants** : pour chaque type de composants, le serveur Jakarta EE définit un conteneur qui fournit les services associés.

→ À chaque *component* son *container*

Les types de conteneurs Jakarta EE

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



Les conteneurs Jakarta 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 Jakarta EE**

  - (2) **déployé dans son conteneur.**

Les conteneurs Jakarta 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 Jakarta EE

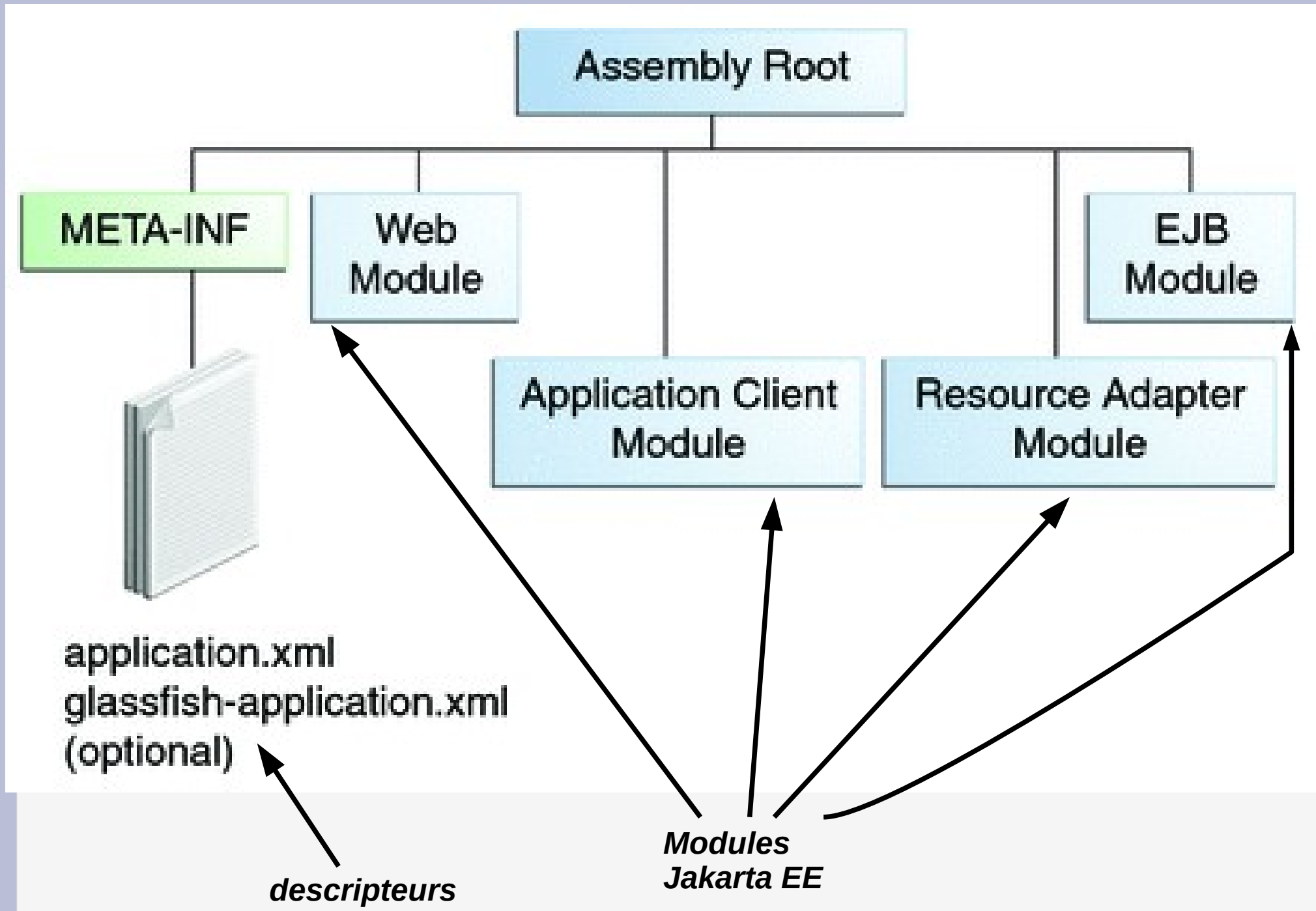
Pour le déploiement

- Une application Jakarta EE est donc composée d'un ensemble d'unités de programmation qui pourront ensuite être déployées sur n'importe quelle plate-forme compatible avec les spécifications Jakarta 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 Jakarta 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 Jakarta 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 Jakarta 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 Jakarta 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 Jakarta EE

- Un module Jakarta EE :
  - Un ou plusieurs composants Jakarta 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

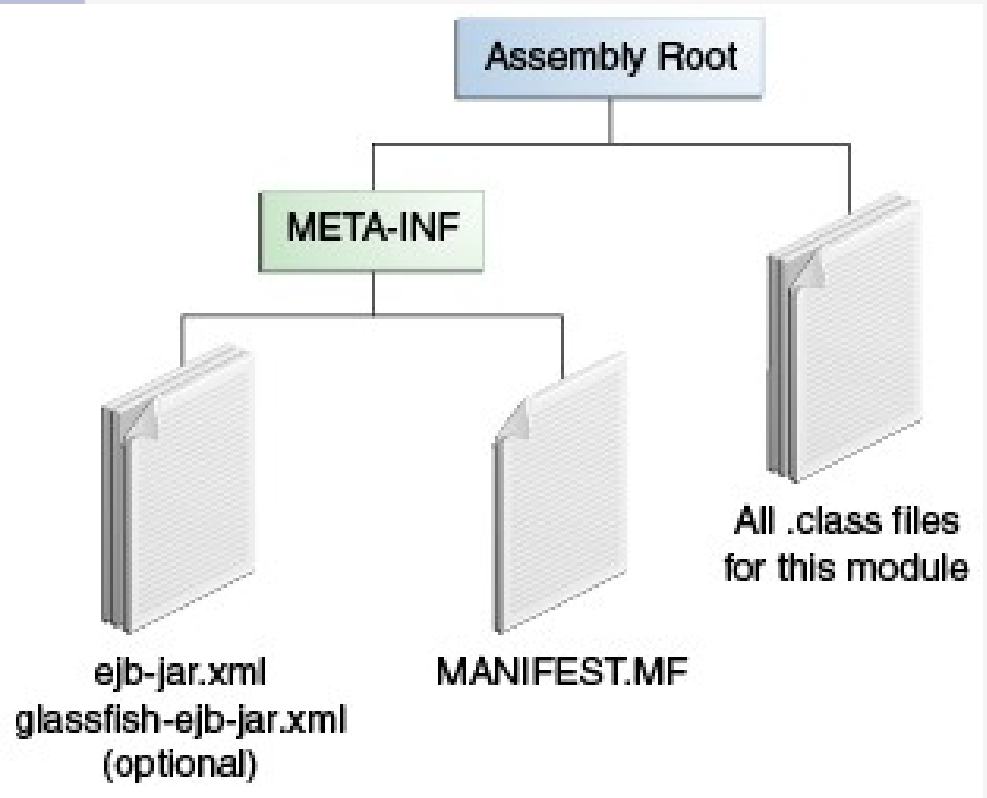
Les 4 types de module Jakarta EE

- **1. Les modules EJB**
  - Contenant des fichiers *class* et un *EJB deployment descriptor* (packagé en fichier jar avec extension .jar)

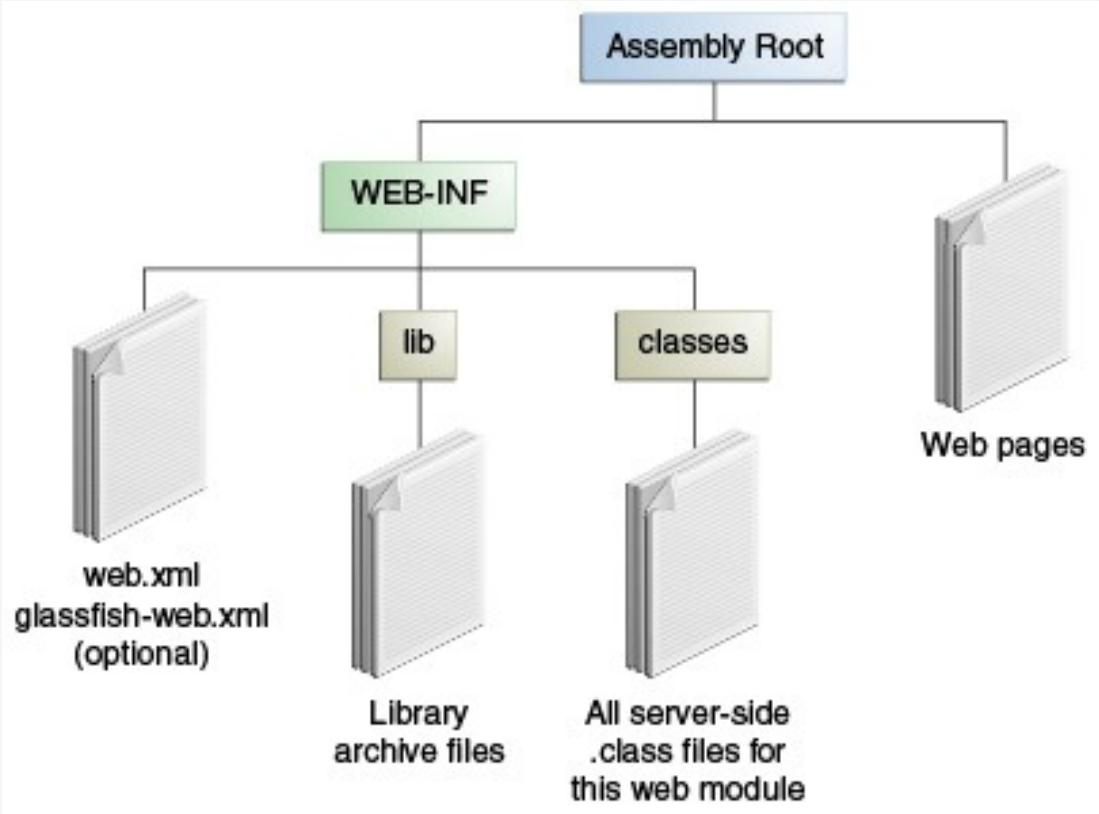
- **2. Les modules Web**
  - Contenant des fichiers servlets, JSP, GIF, HTML et un *web application deployment descriptor*. Packagé en fichier jar avec extension .war (Web ARchive)

Exemples de modules Jakarta EE

EJB JAR Modules .jar



Web Module Archive .war

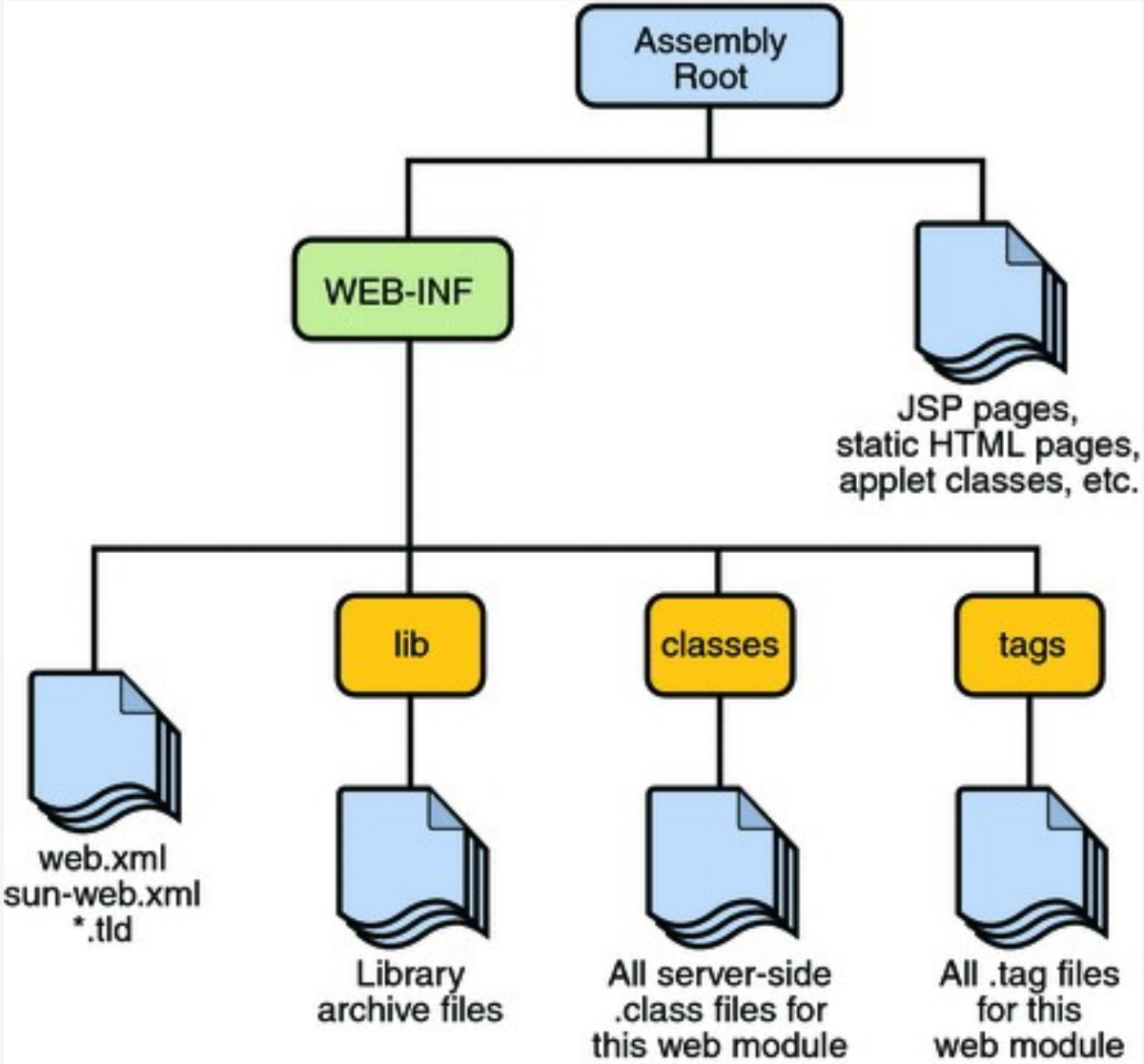


Les 4 types de module Jakarta EE

- **3. Les modules application cliente**
  - Contenant des fichiers *class* et un *application client deployment descriptor.* Packagés en fichiers .jar

- **4. Les ressource adapter modules**
  - Contenant toutes les interfaces, classes, librairies natives, documentation pouvant être nécessaires pour l'interaction avec le EIS + un *ressource adapter deployment descriptor.*
  - Ce qui forme ce qui est appelé *the (j2ee) connector architecture : connexion* avec un EIS particulier.
    - Packagés en fichiers jar avec extension **.rar** (*Resource Adapter aRchive*)

Module Web : fichier war



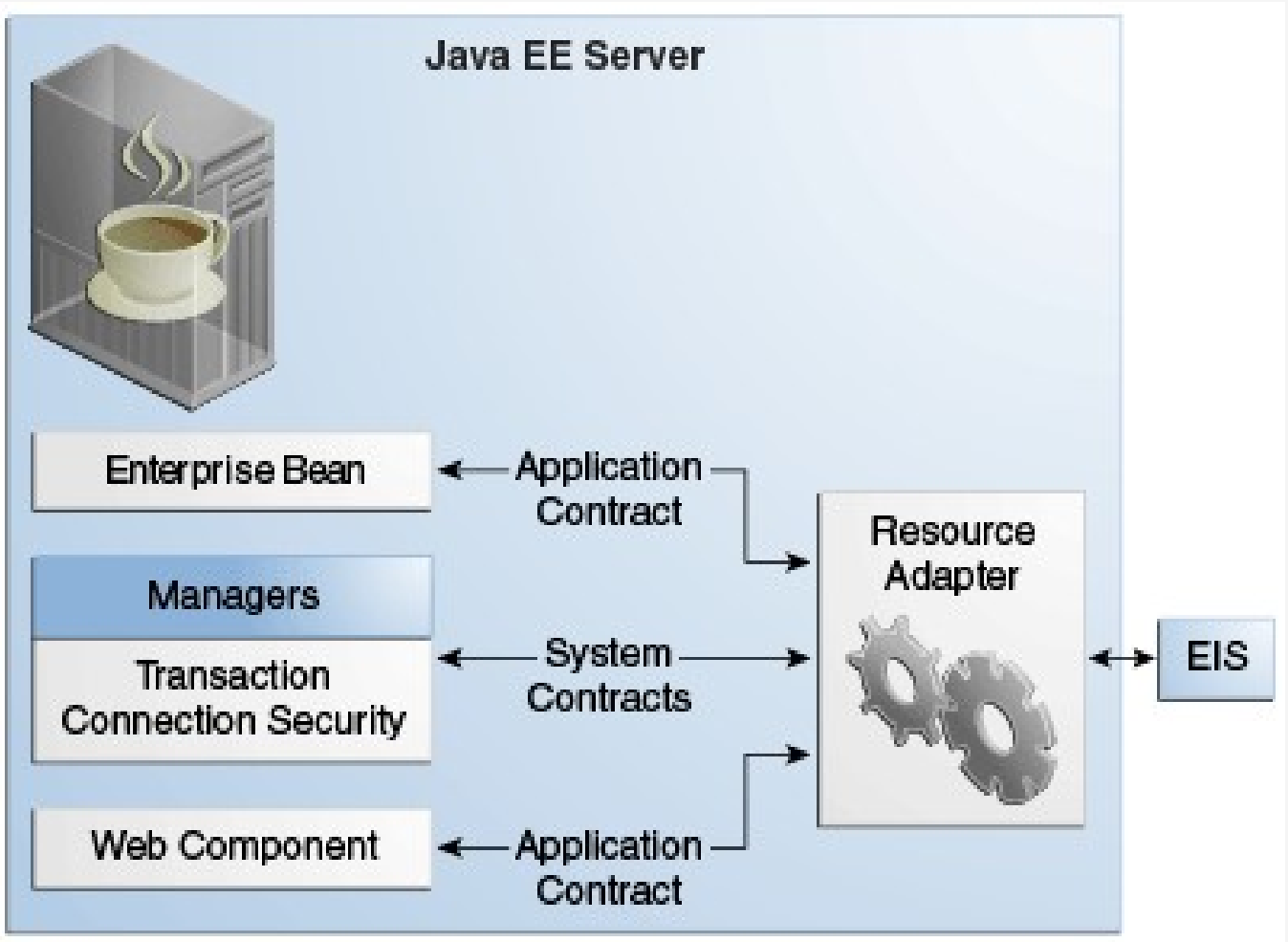
Résumé

- Le développement d'une application Jakarta EE :
  - (1) Récupérer et installer des produits et outils Jakarta EE
  - (2) Développer les composants Jakarta EE
  - (3) Assembler les composants Jakarta EE
  - (4) Déployer les composants Jakarta EE
- Chacune de ces tâches peut être effectuée par une personne différente (industrie)
  - Exemple : (2) développement d'un EJB (fichier jar) (3) combinés avec d'autres dans une application Jakarta EE packagée en fichier ear, (4) le fichier ear est ensuite utilisé par un administrateur pour installer l'application sur un serveur Jakarta EE (chez le client)

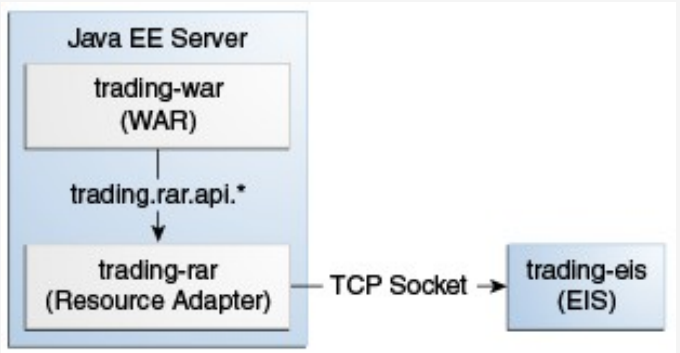
Les serveurs compatibles Jakarta EE

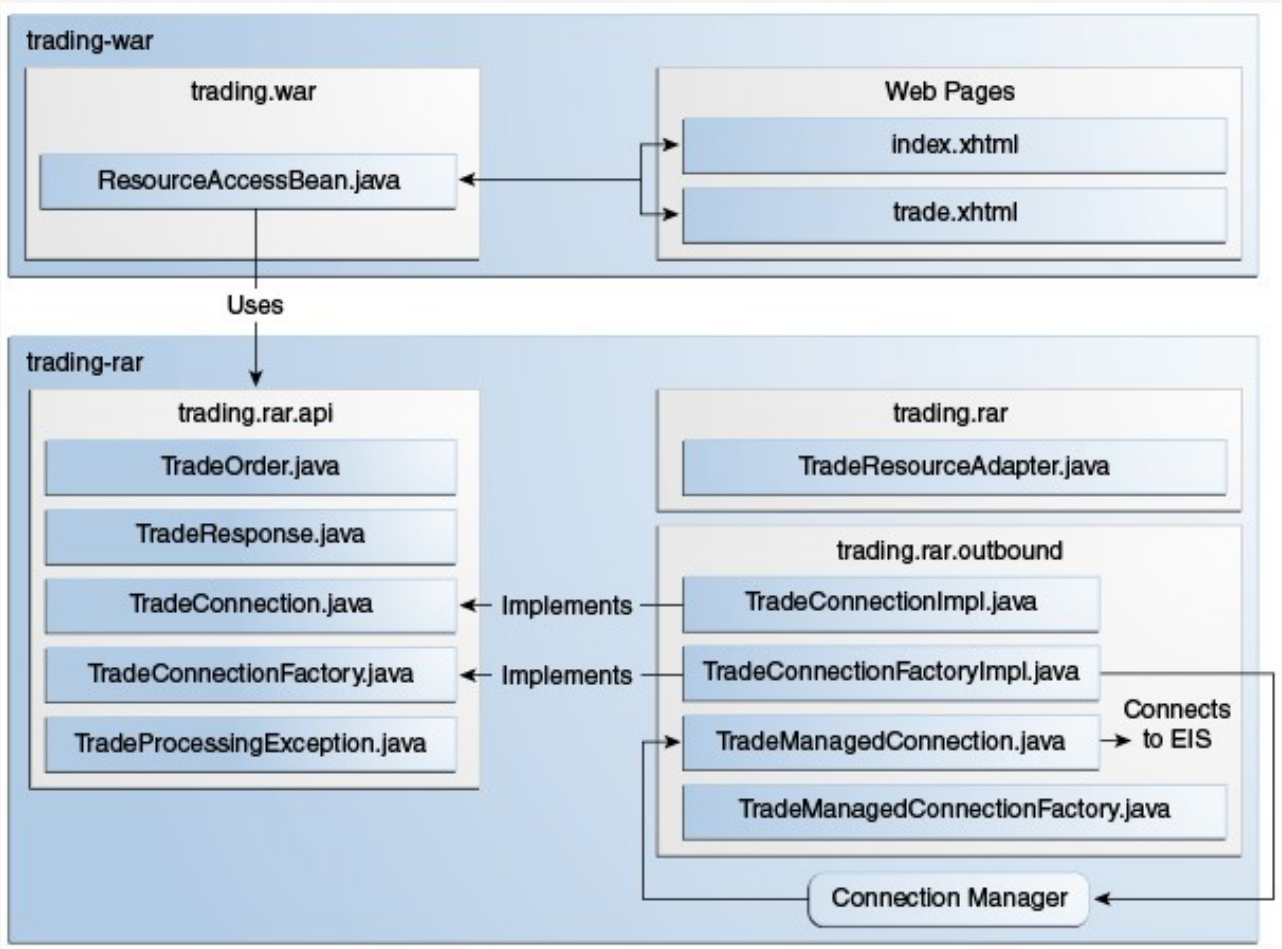
- **Sun Java System Application Server Platform Edition 9.0**, based on the open-source server GlassFish
- **JBoss Application Server Version 5**
- **WebLogic Application Server 10.0** from BEA Systems
- **SAP NetWeaver Application Server**, Jakarta EE 5 Edition from SAP
- **JEUS 6**, an Application Server from TmaxSoft
- **Apache Geronimo 2.0**
- **IBM WebSphere Application Server V7**
- **IBM WebSphere Application Server Community Edition 2.0**, based on Apache Geronimo
- **Oracle Containers for Jakarta EE 11**
- **GlassFish**
- **Apache OpenEJB via Apache Geronimo**

Resource Adapter

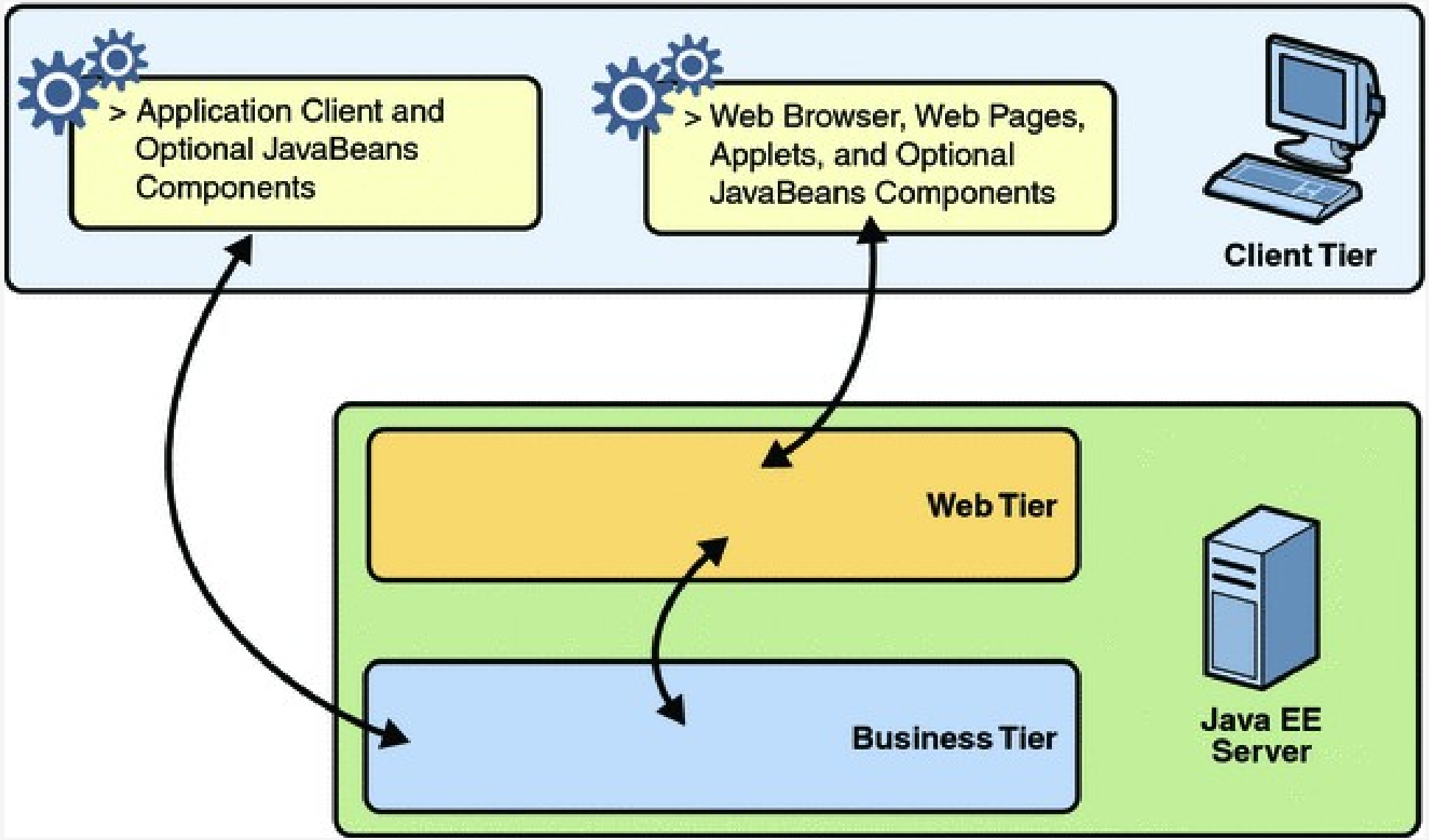


Resource Adapter exemple





I. Les clients Java EE



Applications multi-tiers