



المدرسة العليا للتكنولوجيا - الصويرة
ⵜⴰⵎⴰⵔⵜ ⵜⴰⵎⴰⵏⴰⵢⵜ ⵜⴰⵖⴻⵔⴰⵏⵜ - ⵎⴰⵔⴰⵎⴰⵏⴰ
ÉCOLE SUPÉRIEURE DE TECHNOLOGIE - ESSAOUIRA

Java Enterprise Edition

Introduction à la programmation JEE

Licence Professionnelle Ingénierie des Systèmes Informatiques et Logiciels

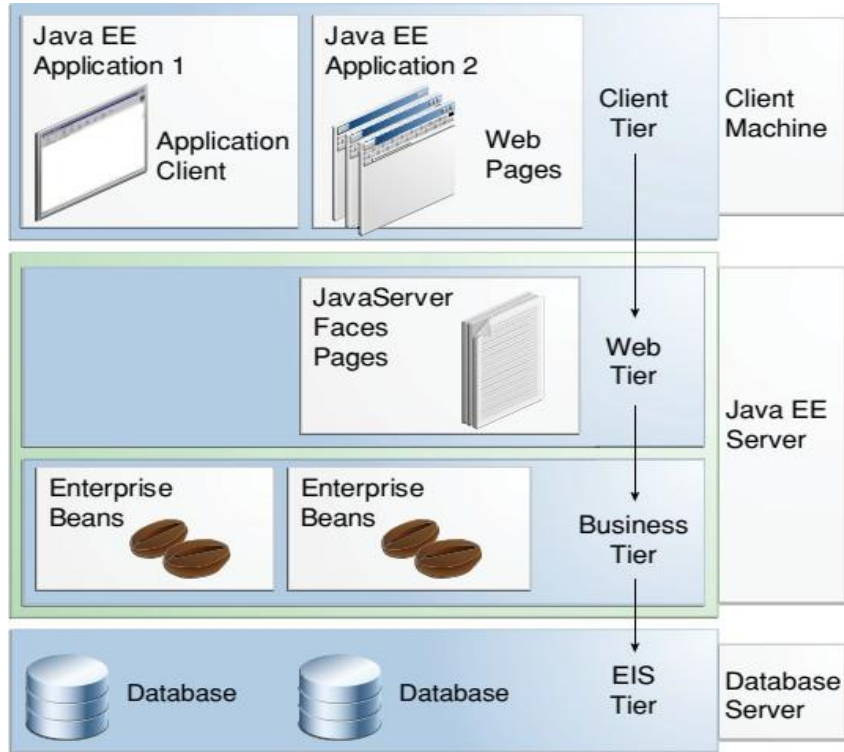
Introduction

- JEE définit un **modèle** pour le développement des applications d'entreprise ***multi tiers*** basées sur le web et faisant intervenir plusieurs composants ***distribués***.
- Avantages :
 - ***Extensible***: Nouveaux besoins -> nouveaux services
 - ***Disponible***: tolérance aux pannes et haute disponibilité
 - ***Portable***: : Indépendance des plateformes
 - ***Sécurisée***: règles de contrôle d'accès, détachement des opérations d'authentification
 - ***Maintenable***: maintenance et test d'intégration assez facile
 - ***Scalable***: 1 utilisateur pareil que 1000 utilisateurs.

Architecture n-tiers

- C'est une architecture **client-serveur** dans laquelle une application est exécutée par plusieurs composants logiciels distincts.
- Exemple : Architecture 3-tiers
 - Le **tier** de **présentation** : interfaces utilisateurs;
 - Le **tier** des **règles de gestion**: applications qui contiennent la logique métier de l'entreprise;
 - Le **tier** des **données** : les points d'accès aux données.
- Avantages :
 - Le lien entre les niveaux est défini et limité à des interfaces;
 - Les interfaces assurent la modalité et l'indépendance;
 - Technologie à chaque niveau.

Les applications multi tiers JEE



- Le tier client s'exécutant sur une machine cliente.
- Le tier web s'exécutant sur un serveur JEE.
- Le tier métier s'exécutant sur un serveur JEE.
- Le tier système d'information d'entreprise s'exécutant sur un serveur de données.

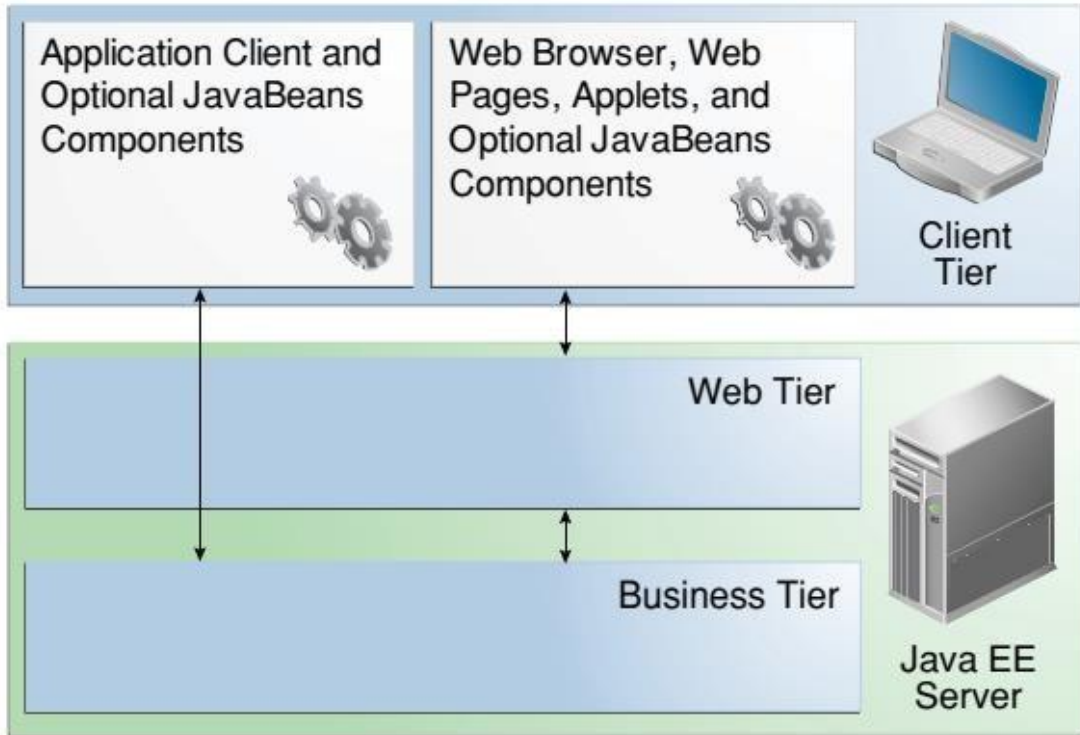
Les composants JEE

- Les composants client:
 - Application cliente
 - Application avec une GUI (AWT; SWING) ou application console.
 - Client web
 - Pages web dynamiques
 - Navigateur web
 - Applet
- Les composants Java Bean
 - Classes java avec certaines règles avec des Getters/Setters
- Les composants web:
 - Servlets; pages web créées par JSF/JSP
- Les composants métier:
 - Enterprise bean

Communication entre composants

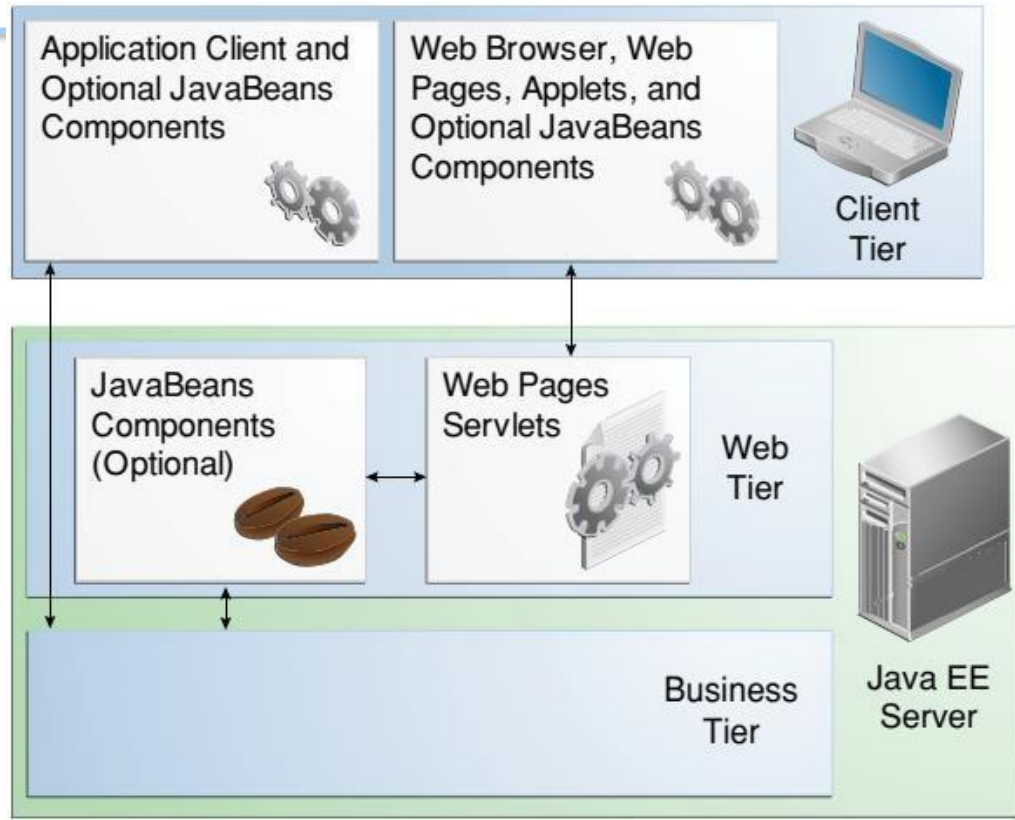
- Du point de vue client:

- Pour un client léger (application Swing par exemple): l'accès au composant métier est direct (via JNDI).
- Un client web accède au composant métier via le composant web.



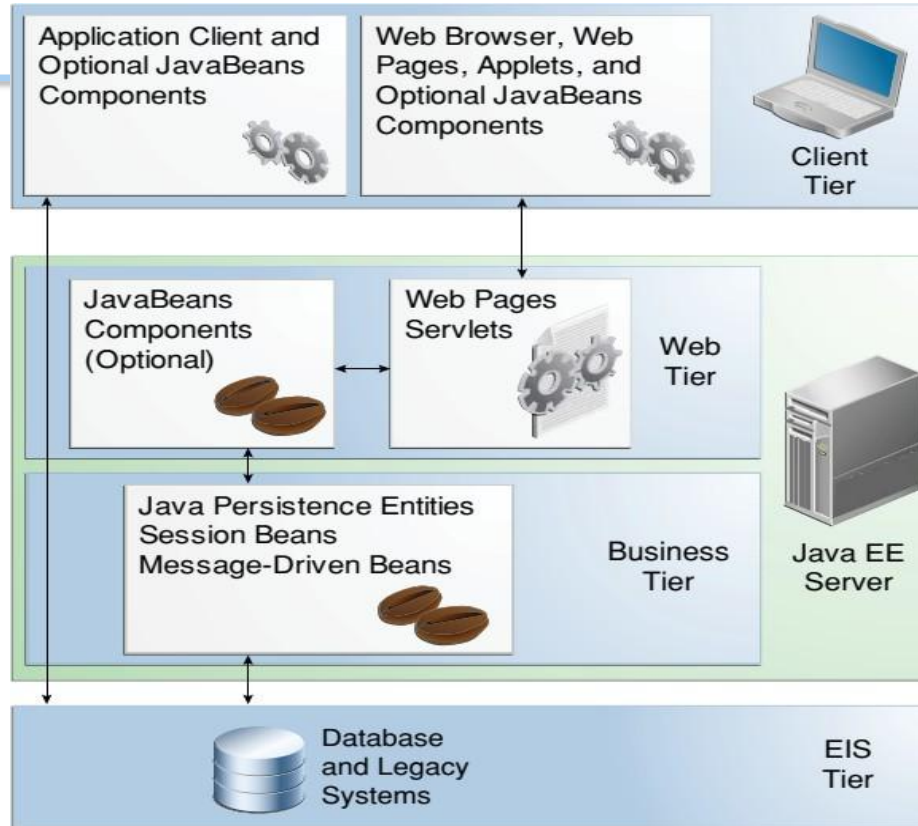
Communication entre composants

- Du point de vue composant web:

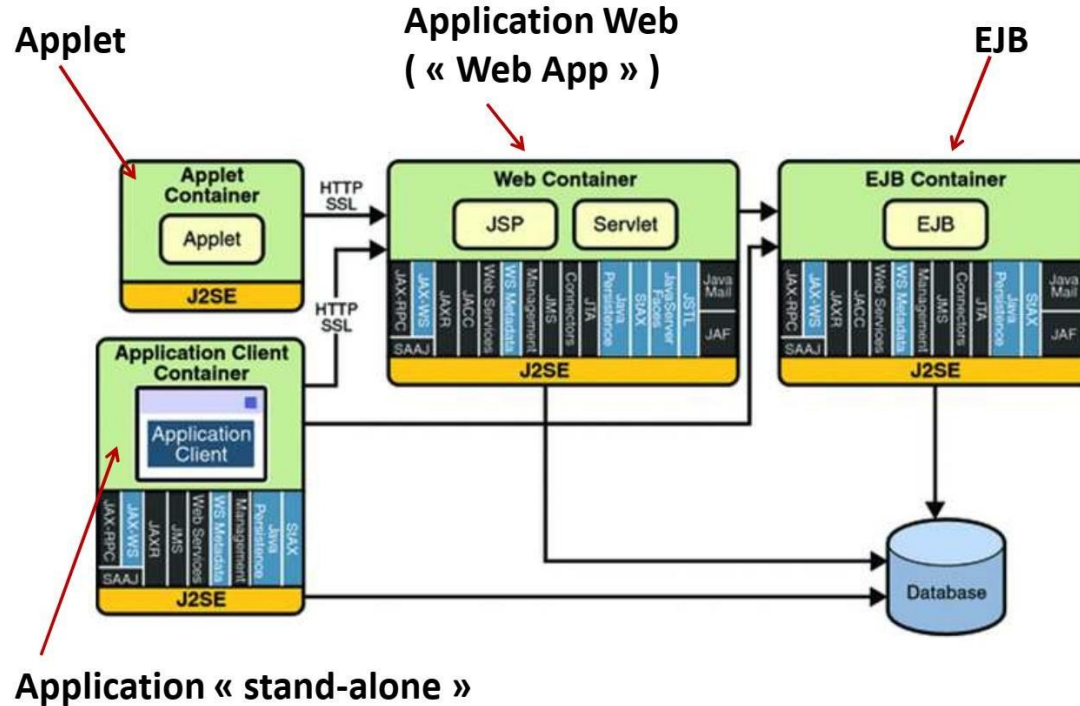


Communication entre composants

- Du point de vue composant métier:



Les conteneurs JEE



Les services des conteneurs JEE

- Le modèle de **sécurité** JEE permet de configurer un composant web ou un Enterprise Bean de telle sorte que les ressources du système ne seront accessibles que par les utilisateurs autorisés.
- Le modèle **transactionnel** JEE permet de spécifier les relations entre les méthodes qui font partie d'une seule transaction pour qu'elles soient traitées comme une seule unité.
- Le service de **localisation** JNDI offre une interface unifiée à de multiples services de nommage et de répertoire au sein de l'entreprise de telle sorte que les composants de l'application accèdent à ses services.

Assemblage et déploiement d'une application JEE

- Chaque application JEE est **empaqueté** dans une ou plusieurs unités standards pour son déploiement dans les plateformes JEE.
- Chaque unité contient :
 - Des composants fonctionnels tels que les Enterprise Bean, les pages web, les Servlets ou les Applets.
 - Un fichier de description de déploiement (optionnel).

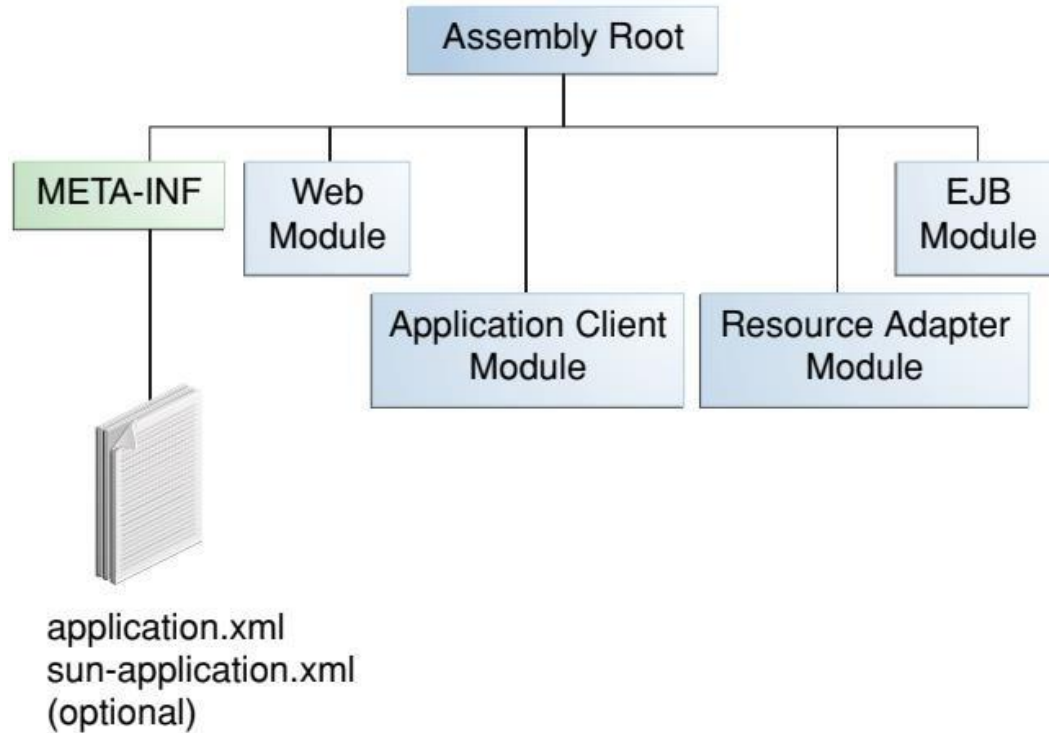
Packaging

- Une application JEE est délivré dans une archive java **.JAR**, ou dans une archive web **.WAR** ou dans une archive d'entreprise **.EAR**.
- Avec les modules et ces différentes archives, il est possible d'assembler plusieurs applications JEE pour utiliser leurs composants sans avoir besoin de faire un surplus de codage.
- Un fichier EAR contient des modules JEE et optionnellement un descripteur de déploiement.

Les modules JEE

- Modules EJB: qui contiennent des fichiers de classe pour les beans d'entreprise et un descripteur de déploiement EJB. Les modules EJB sont fournis sous forme de fichiers JAR avec une extension **.jar**.
- Les modules web: qui contiennent des fichiers de classe de servlet, des fichiers Web, des fichiers GIF et HTML et un descripteur de déploiement d'application Web. Les modules Web sont fournis sous forme de fichiers JAR avec une extension **.war** (archive Web).
- Les modules d'application client: qui contiennent des fichiers de classe et un descripteur de déploiement de l'application. Les modules client sont fournis sous forme de fichiers JAR avec une extension **.jar**.
- Les modules adaptateur de ressource: qui contiennent toutes les interfaces Java, les classes, les bibliothèques natives et toute autre documentation, ainsi que le descripteur de déploiement de l'adaptateur de ressources. Les modules d'adaptateur de ressources sont fournis sous forme de fichiers JAR avec une extension **.rar** (archive de l'adaptateur de ressources).

Le fichier EAR



Les APIs JEE

- **JNDI** (Java Naming and Directory Interface) permet de localiser et d'utiliser les ressources.
- **JDBC** (Java Database Connectivity) est une API qui permet aux programmes JAVA de se connecter et d'interagir avec les base de données SQL.
- **JTA** (Java Transaction API) définit des interfaces standards entre un gestionnaire de transactions et les éléments impliqués dans celle-ci : l'application, le gestionnaire de ressource et le serveur.
- **JPA** (Java Persistence API) est un standard Java utilisé pour la persistance des données qui utilise le principe de Mapping objet/relationnel (**ORM**).

Les APIs JEE

- **Servlet** qui est un composant coté serveur, écrit en Java, dont le rôle est de fournir une trame générale pour l'implémentation de paradigmes " requête-réponse ".
- **JSP** (Java ServerPage) est une extension de la notion de servlet permettant de simplifier la génération de pages web dynamiques. **EJB** : Composants distribués transactionnels, utilisé pour coder la logique métier des applications.
- **Authentication JEE** qui fournit des services d'authentification en se basant sur les concepts d'utilisateur, de domaines et de groupes.

Les APIs JEE

- **JCA (JEE Connector Architecture)** est une API de connexion au système d'information de l'entreprise, notamment aux systèmes dits «Legacy» tels que les ERP.
- **JavaMail** est une API permettant l'envoi de courrier électronique.
- **JMS (Java Message Service)** fournit des fonctionnalités de communication asynchrone (appelées **MOM** pour Middleware Object Message) entre applications.
- **RMI-IIOP** est une API permettant la communication synchrone entre objets.

La gouvernance du projet JEE

- Java Enterprise Edition (**JEE**) est une ***spécification publique*** qui englobe ***plusieurs technologies***.
- Les spécifications évoluent selon un processus de gestion de projet bien défini.
- Le "**Java Community Process**" (**JCP**) décrit le processus d'évolution des plates-formes Java:
 - <http://www.jcp.org/>
- Il s'agit d'un ensemble de procédures formelles utilisées pour développer les spécifications Java.
- Les nouveautés ou les évolutions font l'objet de "JSR" (Java Specification Request). La liste des JSR en cours d'étude est consultable sur le site du JCP.
- Exemple de JSR:
 - **JSR 315** Servlet 3.0
 - **JSR 317** JPA 2.0
 - <https://www.jcp.org/en/jsr/detail?id=317>

JEE et les frameworks

- Ne pas confondre les **frameworks Java** et les **composants normalisés** de **Java EE**.
- Exemple : frameworks de présentation



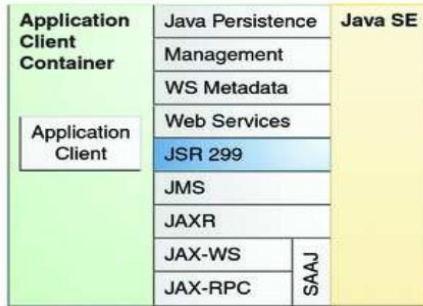
**Non normalisé
(pas de JSR)**




Normalisé

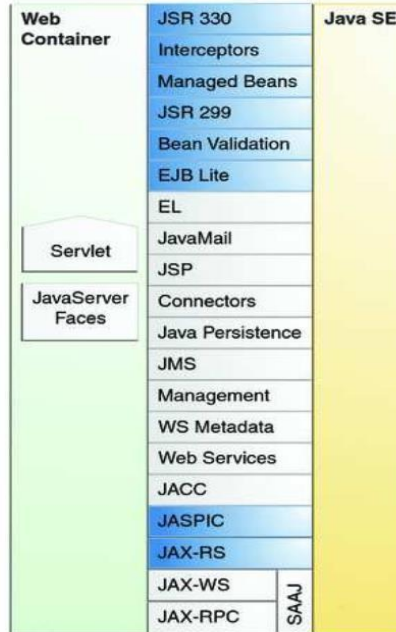
Les containers JEE


Application Client Container



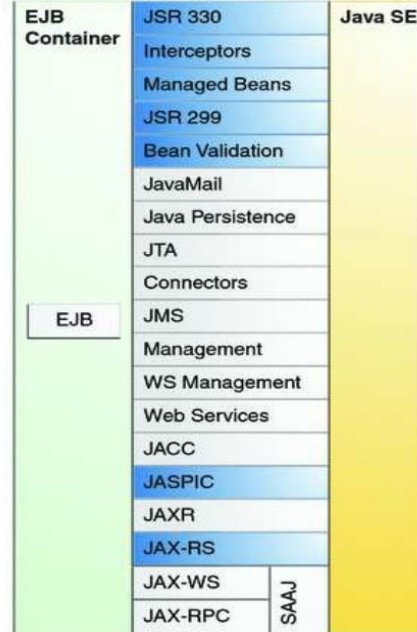
 New in Java EE 6


Web Container



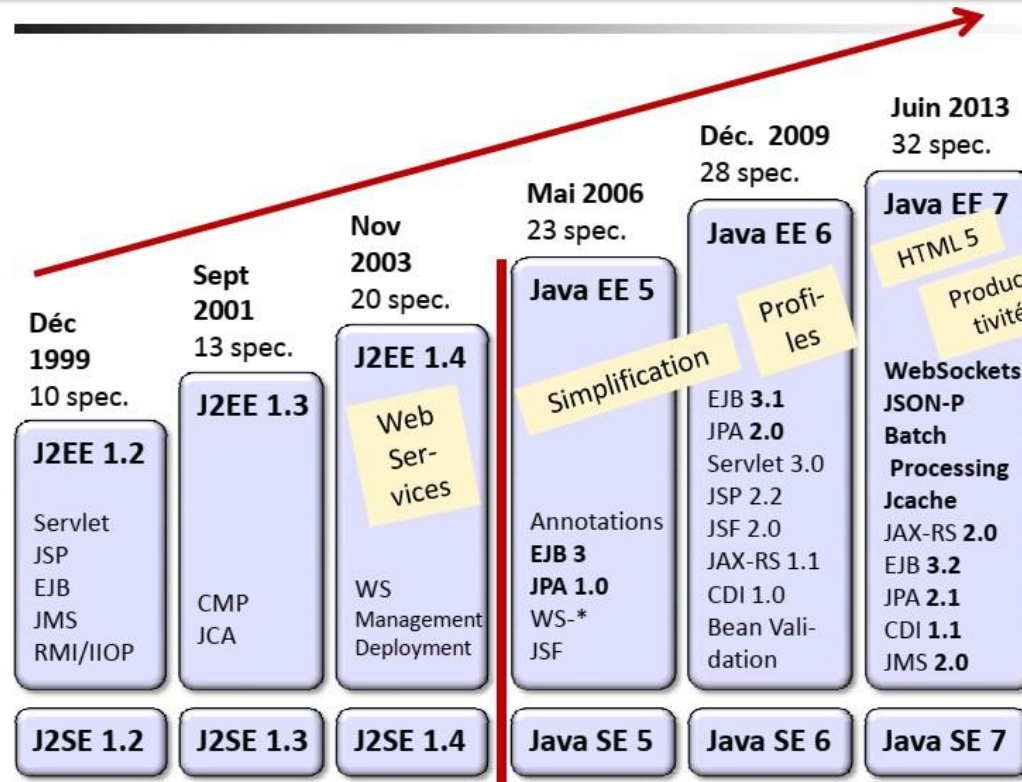
 New in Java EE 6

EJB Container



 New in Java EE 6

De J2EE à Java EE



Le modèle multi tiers JEE (vue par APIs)

