



Introduction

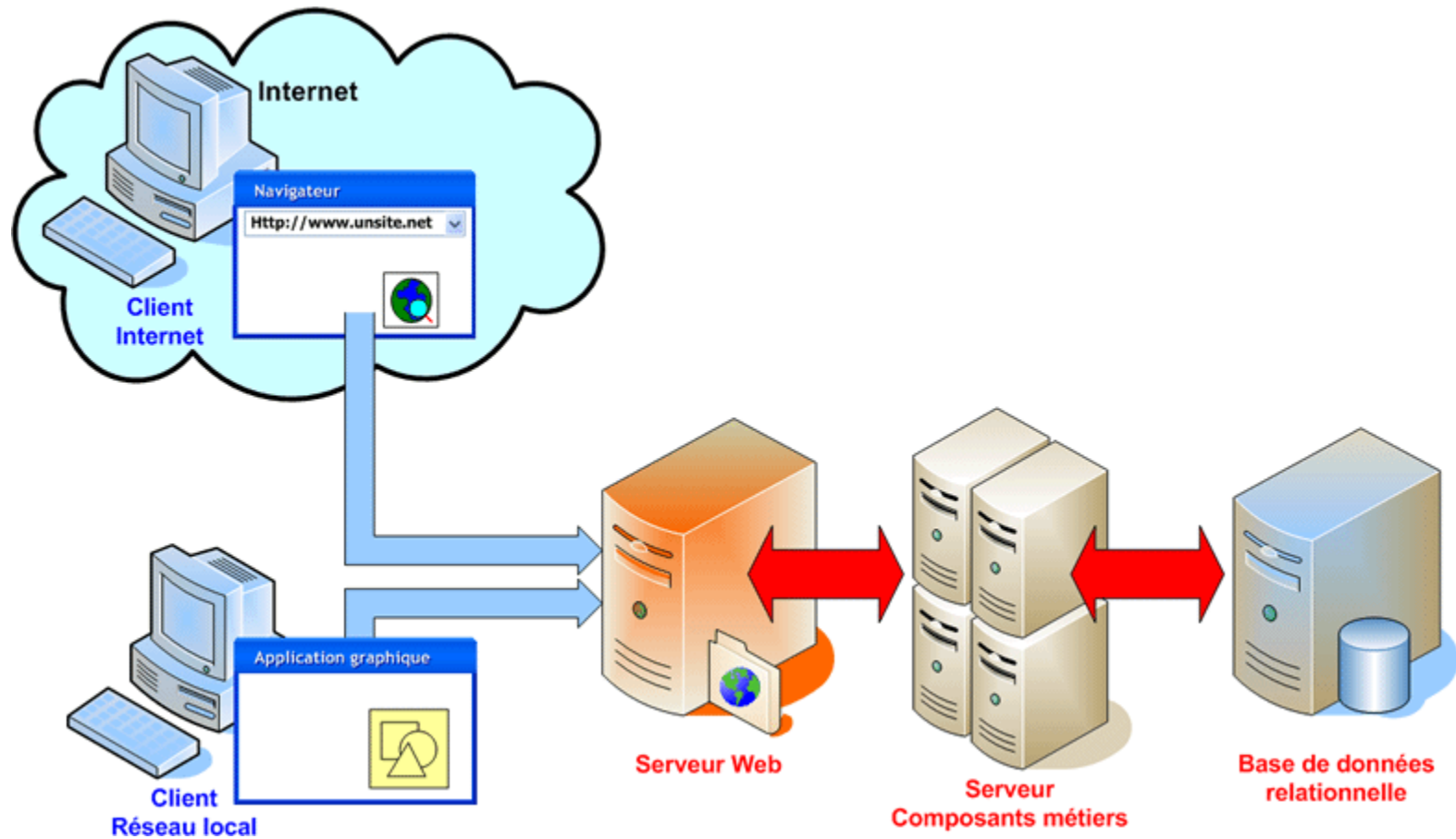
Présentation

A decorative orange line runs horizontally across the bottom of the slide, starting from the left edge, extending to the right, and then curving downwards at an angle before continuing horizontally to the right edge.

■ JavaEE : Java Enterprise Edition

- ✗ Une collection de composants, de conteneurs et de services permettant de créer et de déployer des applications distribuées au sein d'une architecture standardisée.
- ✗ Destiné aux gros systèmes d'entreprise.
- ✗ Les logiciels employés à ce niveau ne fonctionnent pas sur un simple PC mais requièrent une puissance beaucoup plus importante.
- ✗ Les applications doivent être constituées de plusieurs composants pouvant être déployés sur des plate-formes multiples afin de disposer de la puissance de calcul nécessaire.
- ✗ Java EE fournit un ensemble de composants standardisés facilitant le déploiement des applications, des interfaces définissant la façon dont les modules logiciels peuvent être interconnectés, et les services standards, avec leur protocole associé, grâce auxquels ces modules peuvent communiquer.

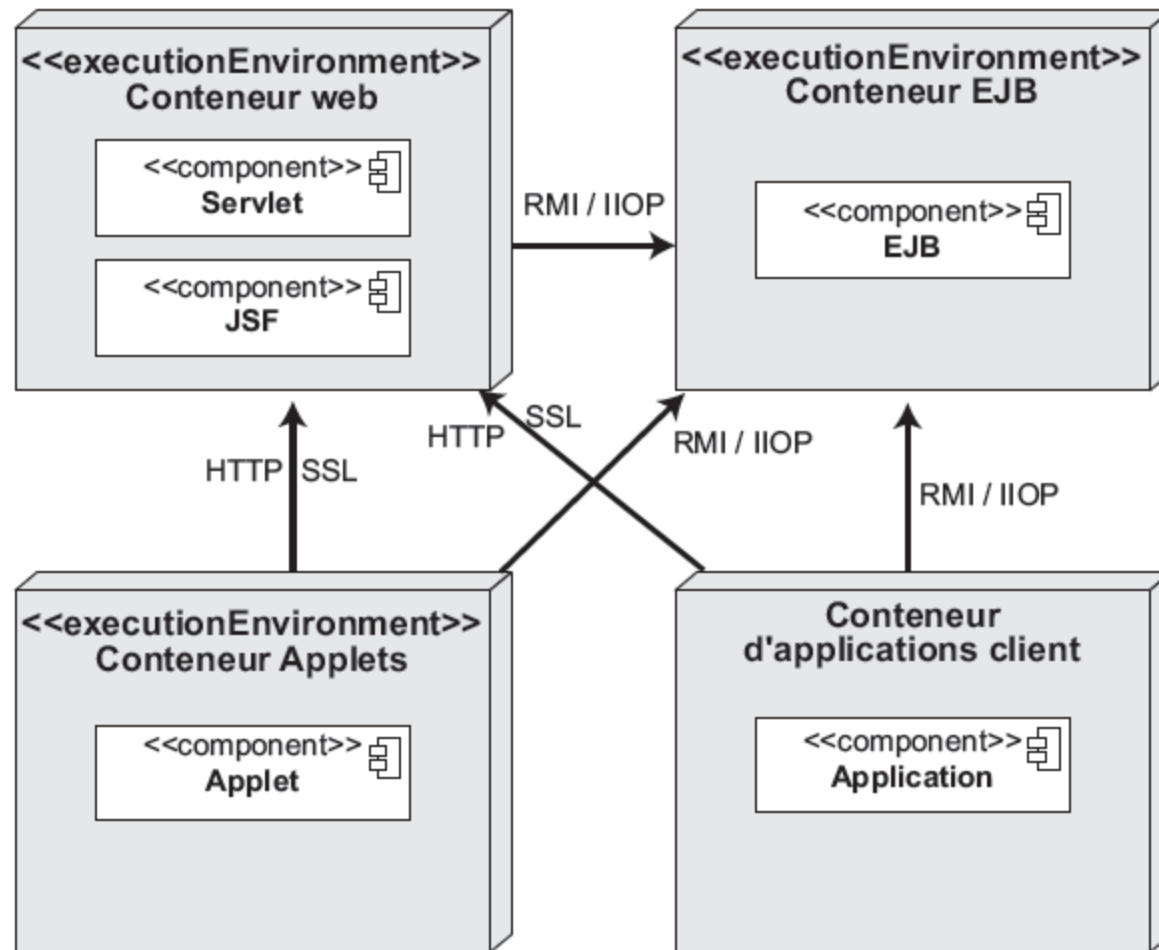
Présentation de Java EE



- **Java EE repose sur des standards**

- ✗ C'est une spécification centrale qui chapeaute un certain nombre d'autres JSR.
- ✗ Les standards facilitent la communication et les échanges.

- **Java EE est un ensemble de spécifications implémentées par différents conteneurs.**
- **Les conteneurs :**
 - ✗ Des environnements d'exécution Java EE qui fournissent certains services aux composants qu'ils hébergent : gestion du cycle de vie, injection de dépendances, etc.
- **Les composants :**
 - ✗ Doivent respecter des contrats bien définis pour communiquer avec l'infrastructure de Java EE et avec les autres composants,
 - ✗ Doivent être assemblés en respectant un certain standard (fichiers archives) avant d'être déployés.
- **Java EE et Java SE**
 - ✗ Java EE étant un sur-ensemble de la plate-forme Java SE,
 - ✗ Les API de cette dernière peuvent donc être utilisées par n'importe quel composant de Java EE.



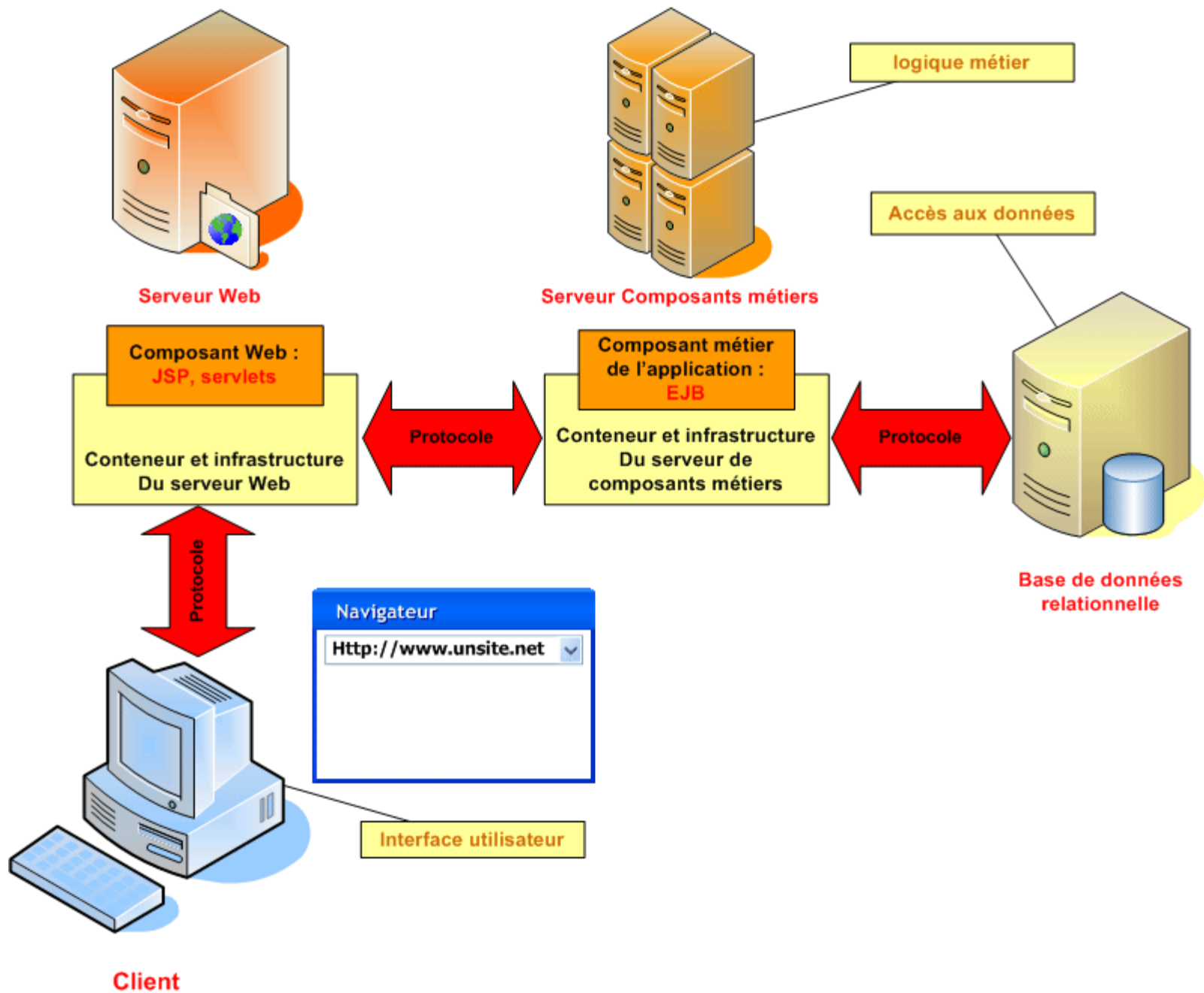
- L'environnement d'exécution de Java EE définit quatre types de composants que doivent supporter toutes les implémentations :
 - ✗ Les *applets* sont des applications graphiques exécutées dans un navigateur web. Elles utilisent l'API Swing pour fournir des interfaces utilisateurs puissantes.
 - ✗ Les *applications* sont des programmes exécutés sur un client. Il s'agit le plus souvent d'interfaces graphiques ou de programmes non interactifs qui ont accès à toutes les fonctionnalités de la couche métier de Java EE.
 - ✗ Les *applications web* (composées de servlets, de filtres de servlet, d'écouteurs d'événements web, de pages JSP et de JSF) s'exécutent dans un conteneur web et répondent aux requêtes HTTP envoyées par les clients web. Les servlets permettent également de mettre en place des services web SOAP et REST.
 - ✗ Les *EJB* (Enterprise Java Beans) sont des composants permettant de traiter la logique métier en modèle transactionnel. On peut y accéder localement et à distance via RMI (ou HTTP pour les services web SOAP et REST).

- **Java EE est découpée en domaines logiques appelés conteneurs**

- ✗ Chacun d'eux joue un rôle spécifique, supporte un ensemble d'API et offre des services à ses composants (sécurité, accès aux bases de données, gestion des transactions, injection de ressources, etc.).
- ✗ Les conteneurs cachent les aspects techniques et améliorent la portabilité.
- ✗ Il faut comprendre les possibilités et les contraintes de chaque conteneur selon le type d'application.

➔ *Pour développer une couche de présentation web, il faut écrire une application JSF et la déployer dans un conteneur web, non dans un conteneur EJB.*

➔ *Pour une application web appelle une couche métier, il faut utiliser à la fois un conteneur web et un conteneur EJB.*



- **Les navigateurs web :**

- ✗ Fournissent des conteneurs d'applets pour exécuter les composants applets.

- **Le conteneur d'applications client (ACC) :**

- ✗ Contient un ensemble de classes et de bibliothèques Java ainsi que d'autres fichiers afin d'ajouter l'injection, la gestion de la sécurité et le service de nommage aux applications Java SE (applications Swing, traitements non interactifs ou, simplement, une classe avec une méthode main()).

- **Le conteneur web (ou conteneur de servlets) :**

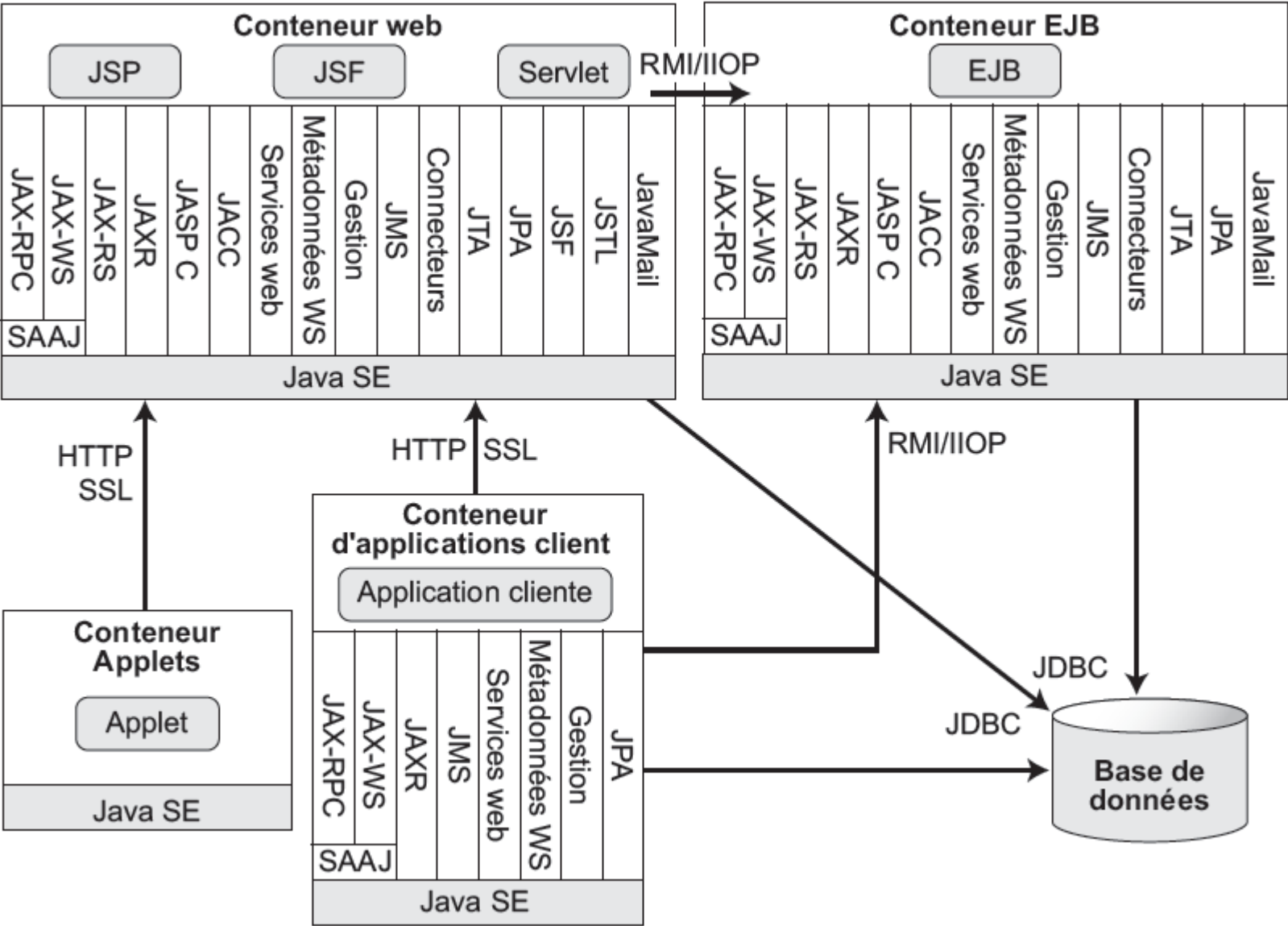
- ✗ Fournit les services sous-jacents permettant de gérer et d'exécuter les composants web (servlets, JSP, filtres, écouteurs, pages JSF et services web).
- ✗ Il est responsable de l'instanciation, de l'initialisation et de l'appel des servlets et du support des protocoles HTTP et HTTPS.

- **Le conteneur EJB :**

- ✗ Responsable de la gestion de l'exécution des beans entreprise contenant la couche métier de votre application Java EE.
- ✗ Il crée de nouvelles instances des EJB, gère leur cycle de vie et fournit des services comme les transactions, la sécurité, la concurrence, la distribution, le nommage ou les appels asynchrones.

- **Les conteneurs fournissent les services sous-jacents à leurs composants.**
 - ✗ LE développeur peut donc se concentrer sur l'implémentation de la logique métier au lieu de résoudre les problèmes techniques auxquels sont exposées les applications d'entreprise.

Java EE : Services



- **JTA :**

- ✗ Ce service offre une API de démarcation des transactions utilisée par le conteneur et l'application.

- **JPA :**

- ✗ Fournit l'API pour la correspondance modèle objet - modèle relationnel (ORM, Object-Relational Mapping).
- ✗ JPQL (Java Persistence Query Language) permet d'interroger les objets stockés dans la base de données sous-jacente.

- **JMS :**

- ✗ Permet aux composants de communiquer de façon asynchrone par passage de messages.
- ✗ Ce service fournit un système d'envoi de message fiable, point a point (P2P).

- **JNDI (Java Naming and Directory Interface) :**

- ✗ Cette API, incluse dans Java SE, permet d'accéder aux systèmes d'annuaires et de nommage.

- **JavaMail :**

- ✗ Le but de cette API consiste à simplifier l'envoi de courrier électronique par les applications.

- **JAF (JavaBeans Activation Framework) :**

- ✗ Cette API, incluse dans Java SE, est un framework pour la gestion des données des différents types MIME.
- ✗ Elle est utilisée par JavaMail.

- **XML :**

- ✗ La plupart des composants Java EE peuvent éventuellement être déployés à l'aide de descripteurs de déploiements XML, et les applications doivent souvent manipuler des documents XML.
- ✗ JAXP (Java API for XML Processing) permet d'analyser des documents XML à l'aide des API SAX et DOM, ainsi que pour XSLT.
- ✗ StAX (Streaming API for XML) est une API d'analyse XML par flux.

- **JCA :**

- ✗ Les connecteurs permettent d'accéder à EIS (Enterprise Information Systems) à partir d'un composant Java EE, que ce soient des bases de données, des mainframes ou des programmes ERP (Enterprise Resource Planning).

- **Sécurité JAAS (Java Authentication and Authorization Service) :**

- ✗ Fournit les services permettant d'authentifier les utilisateurs et d'assurer le contrôle de leurs droits d'accès.

■ Services web :

- ✗ Java EE reconnaît les services web SOAP et REST. JAX-WS (Java API for XML Web Services) remplace JAX-RPC (Java API for XML-based RPC) et fournit le support des protocoles SOAP et HTTP. JAX-RS (Java API for RESTful Web Services) fournit le support des services web reposant sur REST.

■ Gestion :

- ✗ Java EE définit des API pour gérer les conteneurs et les serveurs à l'aide d'un bean d'entreprise spécialisé.
- ✗ L'API JMX (Java Management Extensions) fournit également une aide à la gestion.

■ Déploiement :

- ✗ La spécification de déploiement Java EE définit un contrat entre les outils de déploiement et les produits Java EE afin de standardiser le déploiement des applications.

- Les composants déployés dans les conteneurs peuvent être invoqués via différents protocoles.
- HTTP :
 - ✗ HTTP est le protocole du Web, omniprésent dans les applications modernes. L'API côté client est définie par le packaging `java.net` de Java SE.
 - ✗ L'API côté serveur de HTTP est définie par les servlets, les JSP et les interfaces JSF, ainsi que par les services web SOAP et REST.
 - ✗ HTTPS est une combinaison de HTTP et du protocole SSL (Secure Sockets Layer).
- RMI-IIOP :
 - ✗ RMI (Remote Method Invocation) permet d'appeler des objets distants indépendamment du protocole sous-jacent.
 - ✗ RMI-IIOP est une extension de RMI permettant de l'intégrer à CORBA.

- **Pour être déployés dans un conteneur :**

- ✗ Les composants doivent d'abord être empaquetés dans une archive au format standard. Java SE définit les fichiers jar (Java Archive), qui permettent de regrouper plusieurs fichiers (classes Java, descripteurs de déploiement ou bibliothèques externes) dans un seul fichier compressé (reposant sur le format ZIP).
- ✗ Java EE définit différents types de modules ayant leur propre format de paquetage reposant sur ce format jar commun.

- **Un module d'application client contient :**

- ✗ Des classes Java et d'autres fichiers de ressources empaquetés dans un fichier jar.
- ✗ Ce fichier peut être exécuté dans un environnement Java SE ou dans un conteneur d'application client.

- **Un module EJB contient :**

- ✗ Un ou plusieurs beans de session et/ou des beans pilotés par des messages (MDB, Message Driven Bean) assemblés dans un fichier jar (souvent appelé fichier jar EJB).

- **Un module d'application web contient :**

- ✗ Des servlets, des JSP, des pages JSF, des services web ainsi que tout autre fichier web associé (pages HTML et XHTML, feuilles de style CSS, scripts JavaScript, images, vidéos, etc.).
- ✗ Depuis Java EE 6, un module d'application web peut également contenir des beans EJB Lite
- ✗ Tous ces composants sont assemblés dans un fichier jar portant l'extension .war (souvent désigné sous le terme de fichier war, ou Web Archive).

- **Un module entreprise peut contenir :**

- ✗ Zéro ou plusieurs modules d'applications web, zéro ou plusieurs modules EJB et d'autres bibliothèques classiques ou externes.
- ✗ Toutes ces composantes sont assemblées dans une archive entreprise (un fichier jar portant l'extension .ear) afin que le déploiement de ces différents modules se fasse simultanément et de façon cohérente.

Serveurs d'applications

THE GREAT JAVA APPLICATION SERVER DEBATE

TOMCAT, JBOSS, JETTY, GLASSFISH, LIBERTY
PROFILE, WEBLOGIC & WEBSHERE



jetty://



