

Introduction JAVA EE

Plan

1. **Présentation JAVA EE.**
2. **Serveurs d'applications**
3. **Types d'architectures**
4. **Architecture JAVA EE**
5. **Les servlets Java**
6. **Les pages jsp**
7. **Les EJB**

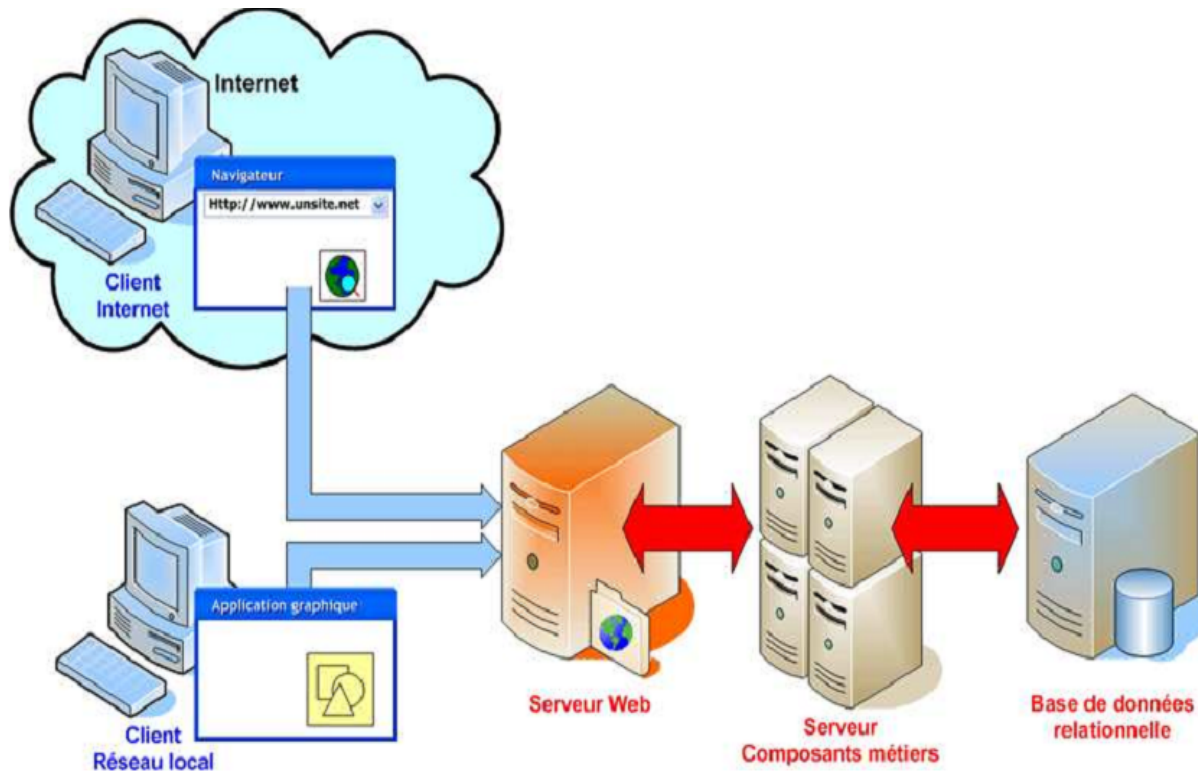
1-QUE VEUT DIRE JAVA EE ?

- Signifie Java Entreprise Edition et représente essentiellement des applications d'entreprise.

Inclut le stockage sécurisé des informations, ainsi que leur manipulation et leur traitement : factures clients, calculs d'amortissement, réservation de vols, etc.

- Ces applications peuvent avoir des interfaces utilisateurs multiples, par exemple une interface Web pour les clients, accessible sur Internet et une interface graphique fonctionnant sur les ordinateurs de l'entreprise sur le réseau privé de celle-ci.

EXEMPLE 1



- Gérer les communications entre systèmes distants
- S'occuper des différents protocoles de communication
- s'occuper également automatiquement de la base de données
- ...

Caractéristiques

- Java EE reprend tout ces concepts de la programmation réseaux en rajoutant la sécurité.
- Java EE est une architecture performante, elle a été mise en œuvre pour le monde des entreprises qui ont besoin d'un système stable qui accepte facilement **la montée en charge sans poser de problème de sécurité.**
- Java EE exploite au maximum **la technologie des objets distribués**, ce qui permet d'avoir un système simple à utiliser côté clients.

2- SERVEURS D'APPLICATIONS

- Tout comme les bibliothèques d'interfaces graphiques comme Swing fournissent les services nécessaires au développement d'application graphiques.
- Les serveurs d'applications mettent à disposition les fonctionnalités permettant de réaliser des applications d'entreprise
- Exemples : communication entre ordinateurs, mis en place de protocole adaptés, gestion des connexions avec une DB, présentation de pages Web, gestion des transactions, etc

QU'EST-CE QUE JAVA EE ?

- Java EE est souvent synonyme d' Entreprise JavaBeans pour de nombreux développeurs. Mais, Java EE est beaucoup plus que cela.
- Java EE est une collection de composants, de conteneurs et de services permettant de créer et de déployer des applications distribuées au sein d'une architecture standardisée.
- Java EE est logiquement destiné aux gros systèmes d'entreprise.
- 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.

3-ARCHITECTURE MULTITIERS

Un des thèmes du développement d'applications Java EE est la décomposition de celles-ci en plusieurs niveaux ou tiers.

Généralement, une application d'entreprise est composée de trois couches fondamentales (d'où le terme décomposition en **trois tiers**) :

LA PREMIÈRE : PRÉSENTATION

- a pour rôle d'afficher les données pour l'utilisateur et de collecter les informations qu'il saisit.
- Cette interface est souvent appelée couche de présentation car sa fonction consiste à présenter les données à l'utilisateur et à lui permettre de fournir des informations au système.
- Est la partie de l'application responsable de la création et du contrôle de l'interface présentée à l'utilisateur et de la validation de ses actions.

LA DEUXIEME : LA LOGIQUE MÉTIER

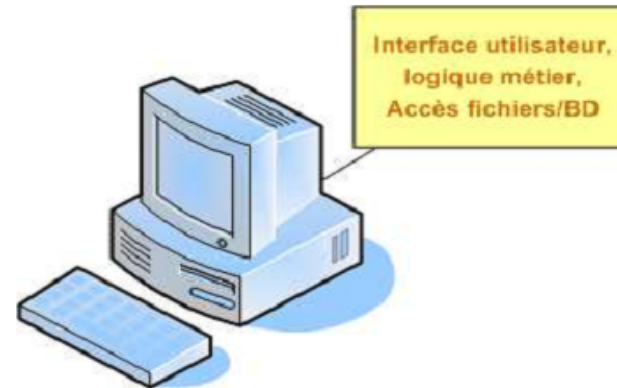
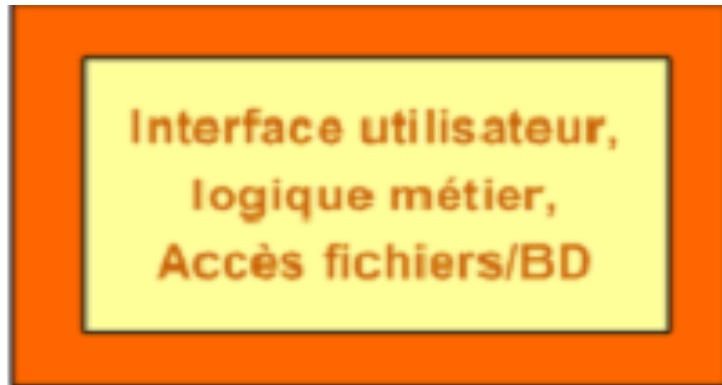
- permet à l'application de fonctionner et de traiter les données.
- La logique métier est mise en œuvre à partir des règles métier.

Exemple : Dans une application de paye, par exemple, la logique métier multiplie les heures travaillées par le salaire horaire pour déterminer combien chaque employé doit toucher.

LA TROISIEME : PERSISTANCE

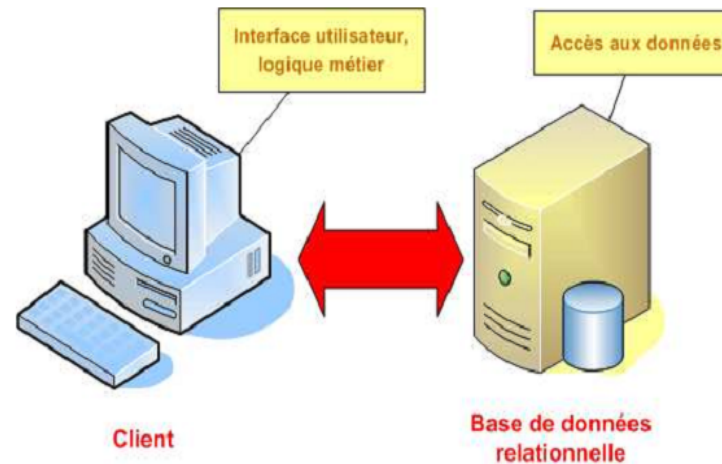
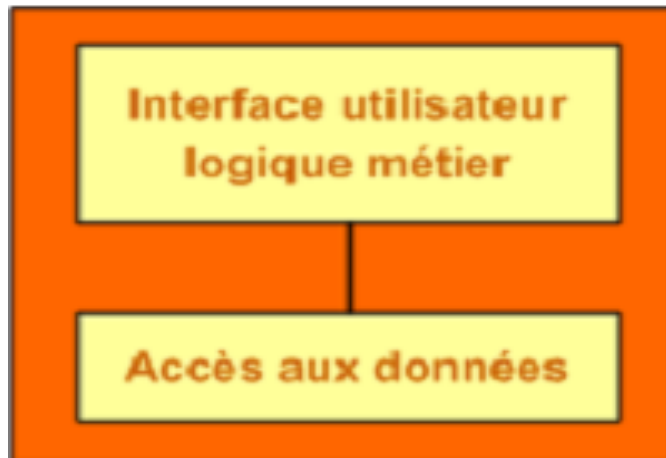
- Toutes les applications d'entreprise ont besoin d'écrire et de lire des données.
- Cette fonctionnalité est assurée par la couche d'accès de données, également appelée couche de persistance, qui assure la lecture, l'écriture à partir des différentes sources.

ARCHITECTURE 1 TIER



- Les applications bureautiques sont conçues pour fonctionner sur un ordinateur unique.
- Cette architecture monolithique est appelée simple tiers car toutes les fonctionnalités sont comprises dans une seule couche logicielle.

ARCHITECTURE 2 TIERS



- La raison de cette approche est de centraliser les données afin de permettre à plusieurs utilisateurs d'y accéder simultanément.
- Les données peuvent ainsi être partagées entre plusieurs utilisateurs de l'application.
- Cette architecture est appelée client-serveur, qui dans notre approche peut être représentée en deux tiers.

INCONVÉNIENT ARCHITECTURE 2 TIERS

- la logique chargée de la manipulation des données et de l'application des règles métiers afférentes est incluse dans l'application elle-même.

- Un Contre Exemple :

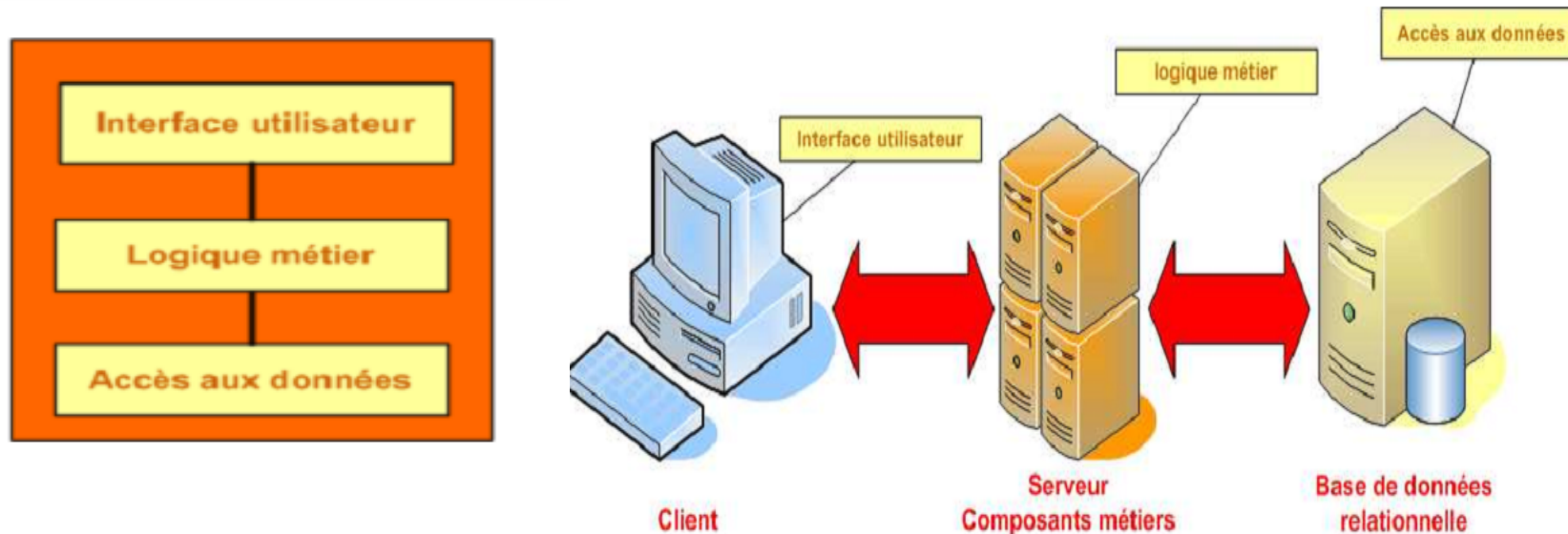
lorsque plusieurs applications doivent partager l'accès à une base de données. Il peut y avoir, par exemple, une règle stipulant qu'un client affichant un retard de paiement de plus de 90 jours verra son compte suspendu. Il n'est pas compliqué d'implémenter cette règle dans chaque application accédant aux données client.

Toutefois, si la règle change et qu'un délai de 60 jours est appliqué, il faudra mettre à jour toutes les applications, ce qui peut être contraignant.

INCONVENIENT ARCHITECTURE 2 TIERS

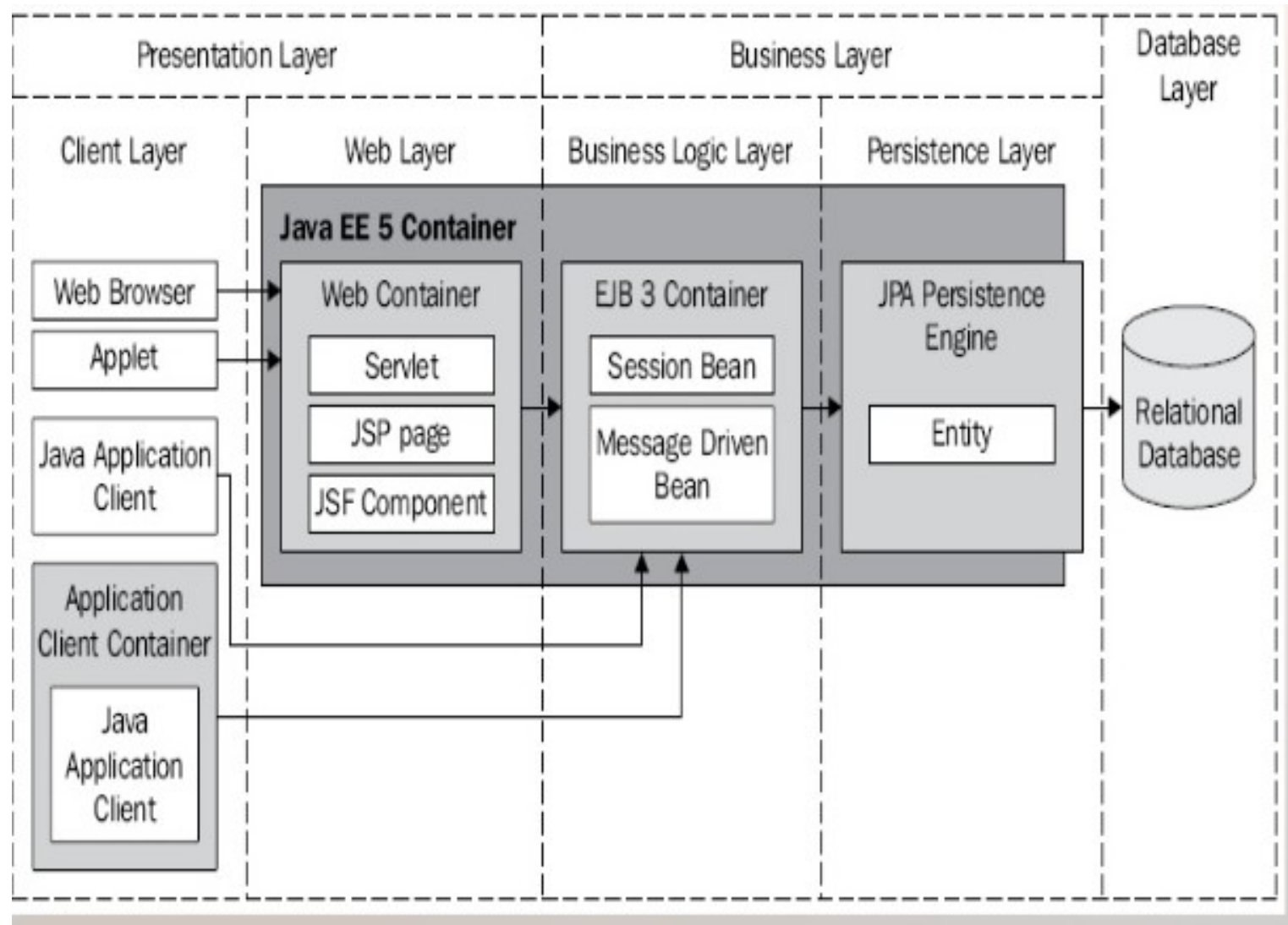
- Pour éviter ce genre de problème
- la solution consiste à séparer physiquement les règles métier en les plaçant sur un serveur où elles n'auront à être remise à jour qu'une seule fois
- Cette solution ajoute un troisième tiers à l'architecture client-serveur

ARCHITECTURE 3 TIERS



- la logique métier est extraite de l'application cliente.
- Elle n'est plus responsable que de la présentation de l'interface à l'utilisateur et de la communication avec le tiers médian.
- Son rôle est réduit à la couche présentation.

4- ARCHETECTURE Java EE



CÔTÉ CLIENT

- peut être une application console (texte seulement) écrite en Java.
- une application dotée d'une interface graphique développée en Swing.

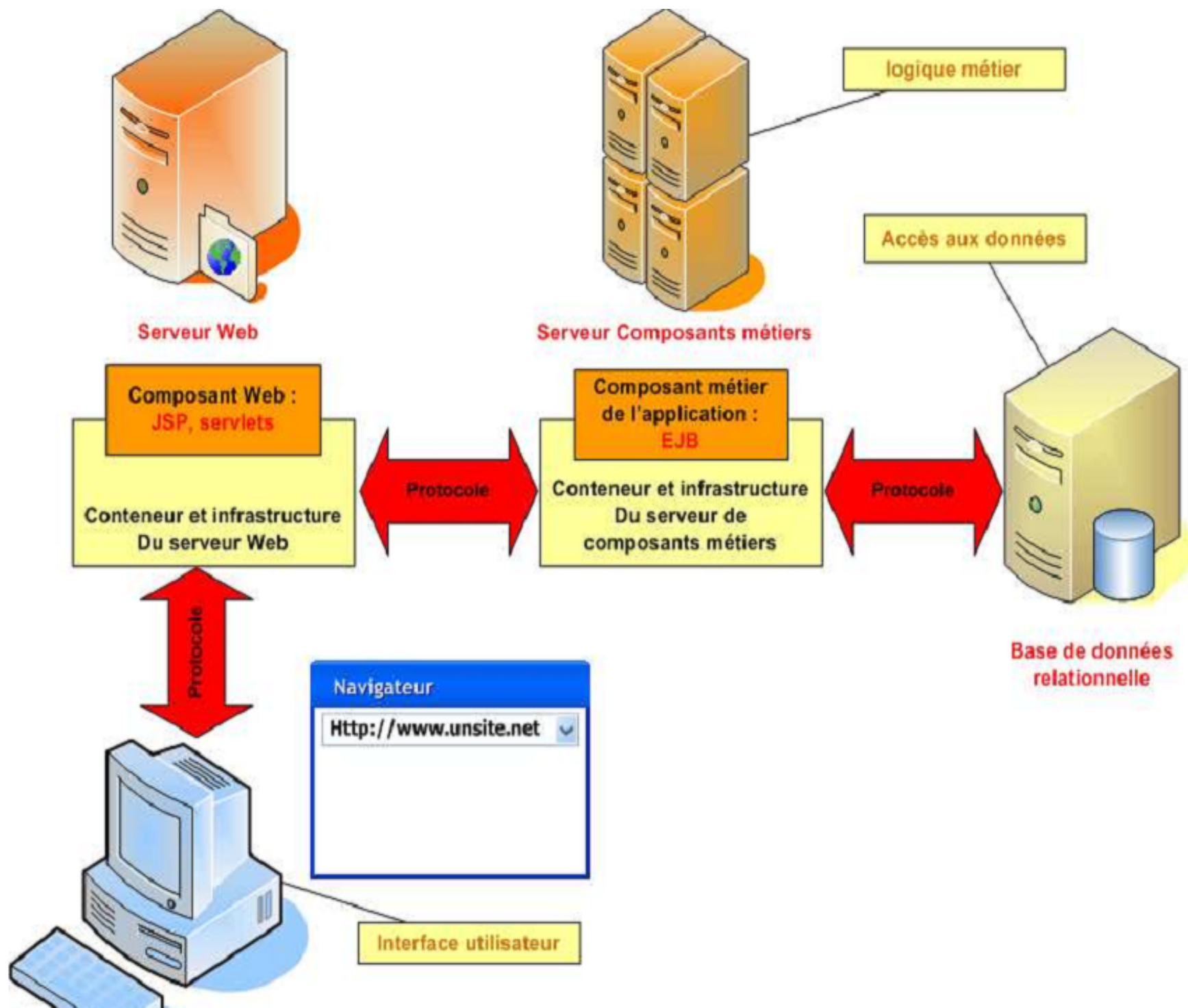
en raison de la quantité importante de code, ce client est appelé client lourd.

- peut également être conçu pour être utilisé à partir du Web.
- Ce type de client fonctionne à l'intérieur d'un navigateur Web.

Vu que la plus grande partie du travail est reportée sur le serveur le client est léger.

CÔTÉ SERVEUR

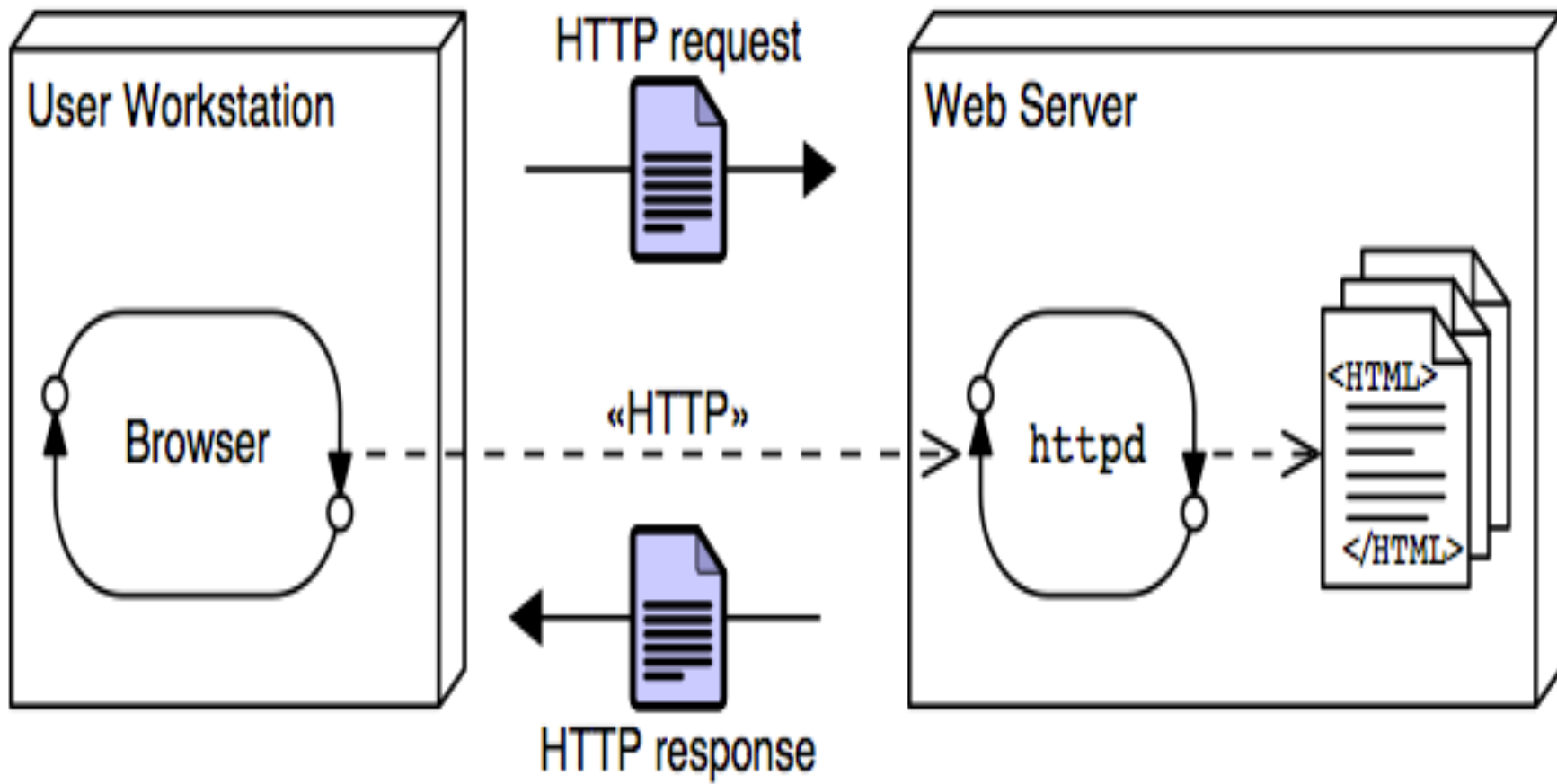
- Les composants déployés sur le serveur peuvent être classés en deux groupes.
- **Les composants Web sont réalisés à l'aide de servlets ou de JavaServer Pages (JSP)**
- **Les composants métiers, dans le contexte Java EE, sont des Entreprise JavaBeans (EJB).**



5-Les Servlets Java

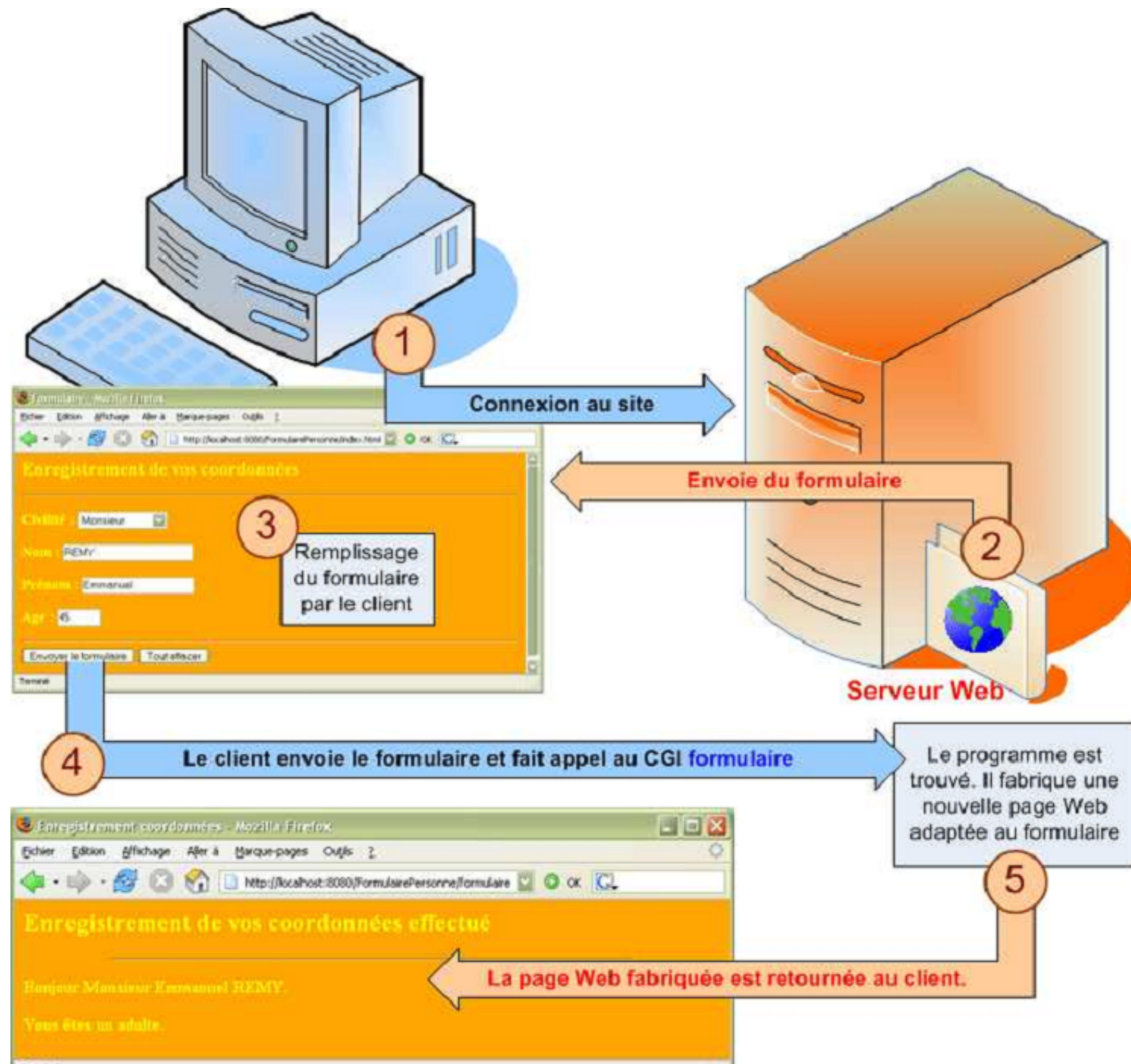
Statiques :

- Vous avez sans doute l'habitude d'accéder à des pages HTML statiques à l'aide d'un navigateur envoyant une requête à un serveur Web
- Ce serveur renvoi cette page qui est stockée sur son disque.
- **le serveur ici : joue le rôle d'un bibliothécaire virtuel qui renvoie le document demandé.**

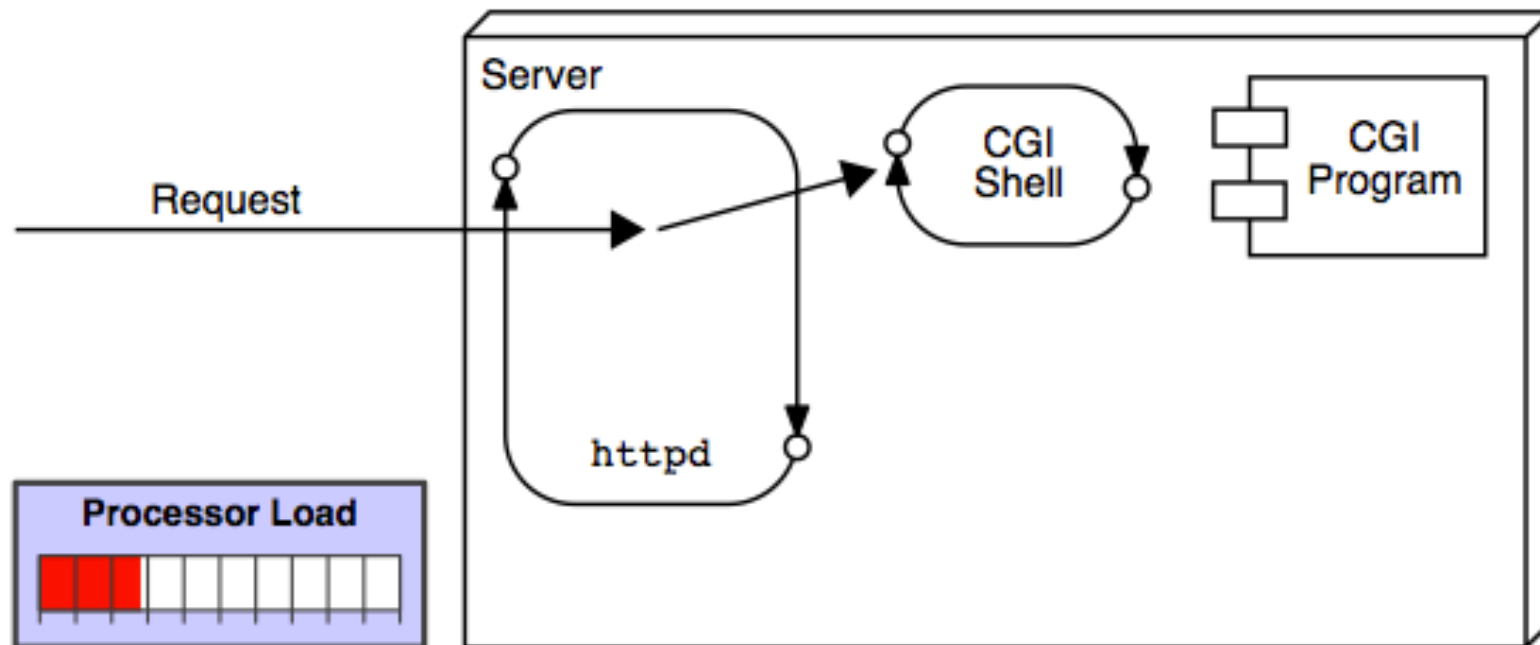


Dynamiques :

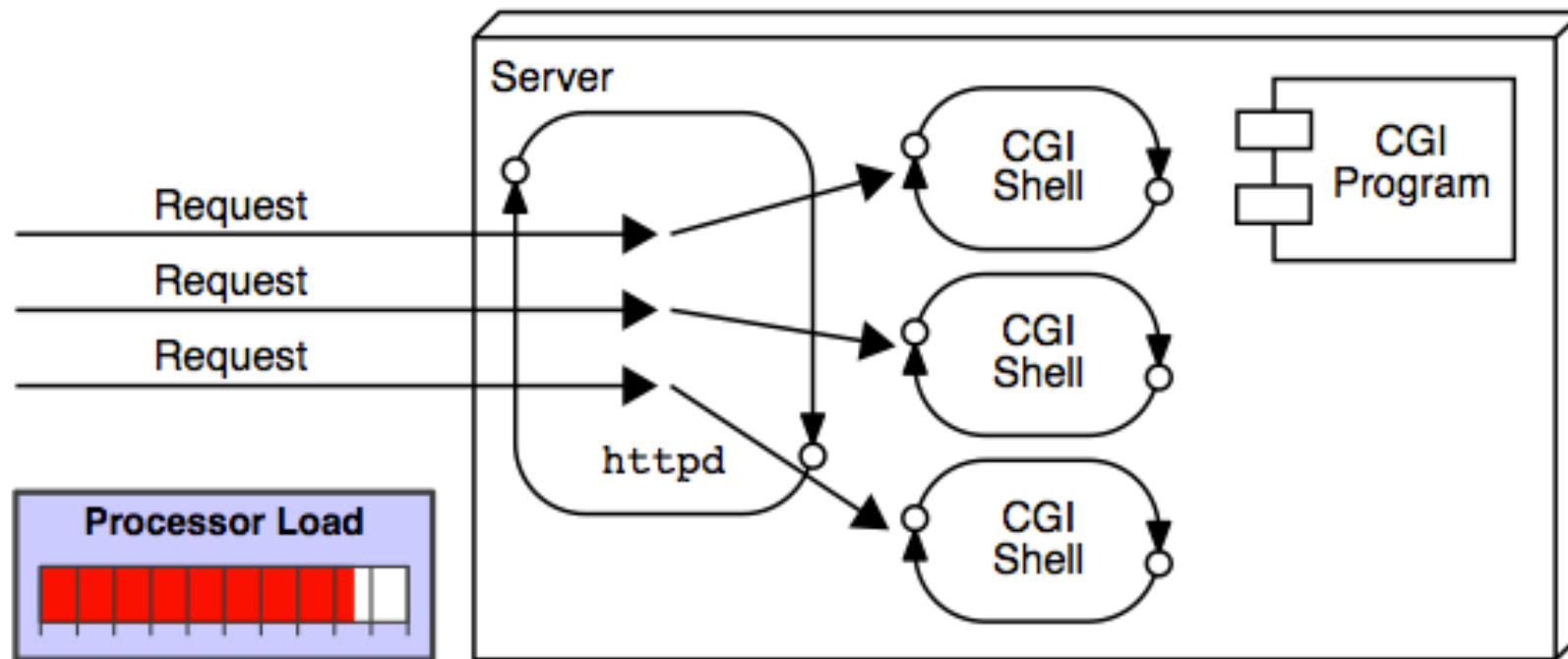
- le modèle précédant ne permet pas d'accéder à des pages dynamiques, dont le contenu serait créé à la demande.
- Supposons par exemple que le client souhaite obtenir une liste des documents HTML correspondant à certains critères
- **Dans ce cas, il est nécessaire de créer une page HTML différente en fonction des critères spécifiés par le client.**



LES SERVLETS JAVA vs CGI



LES SERVLETS JAVA vs CGI



LES SERVLETS JAVA vs CGI

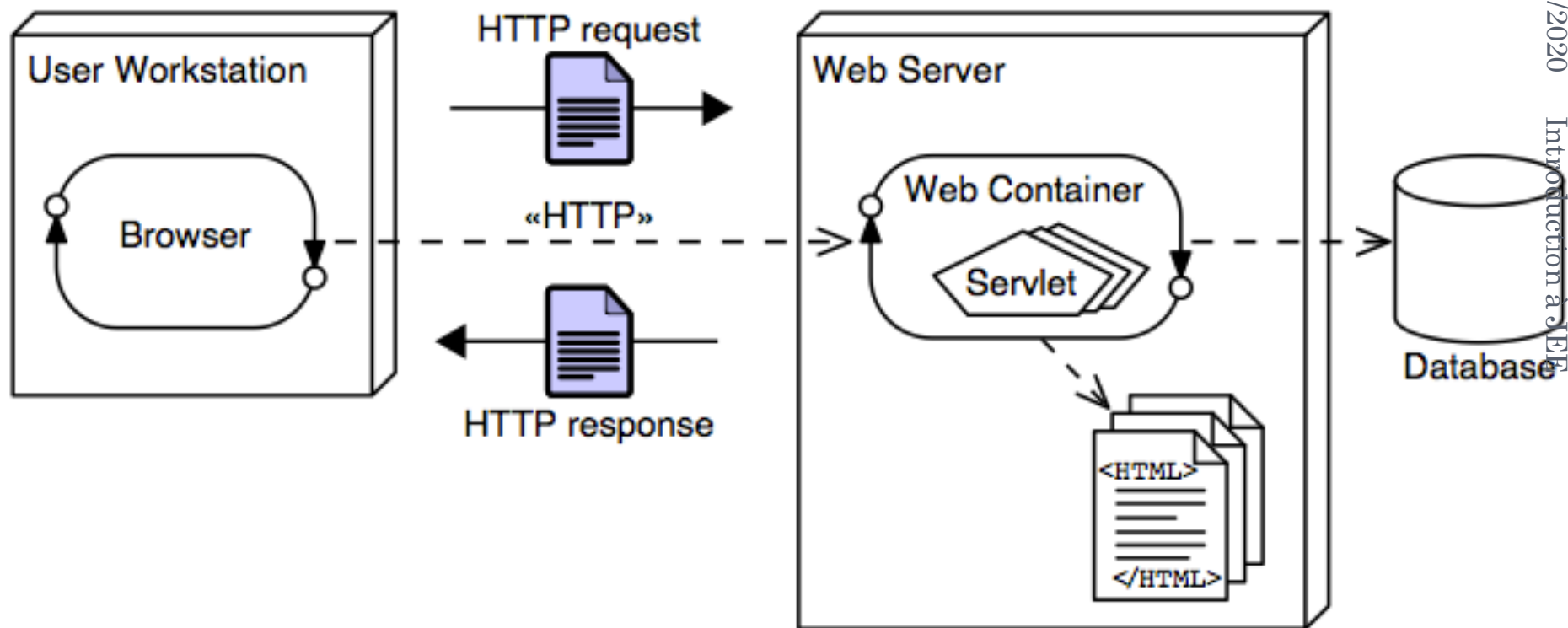
Avantages CGI

- Écrit par plusieurs langages de programmation.
- Relativement facile pour un WEB designer.

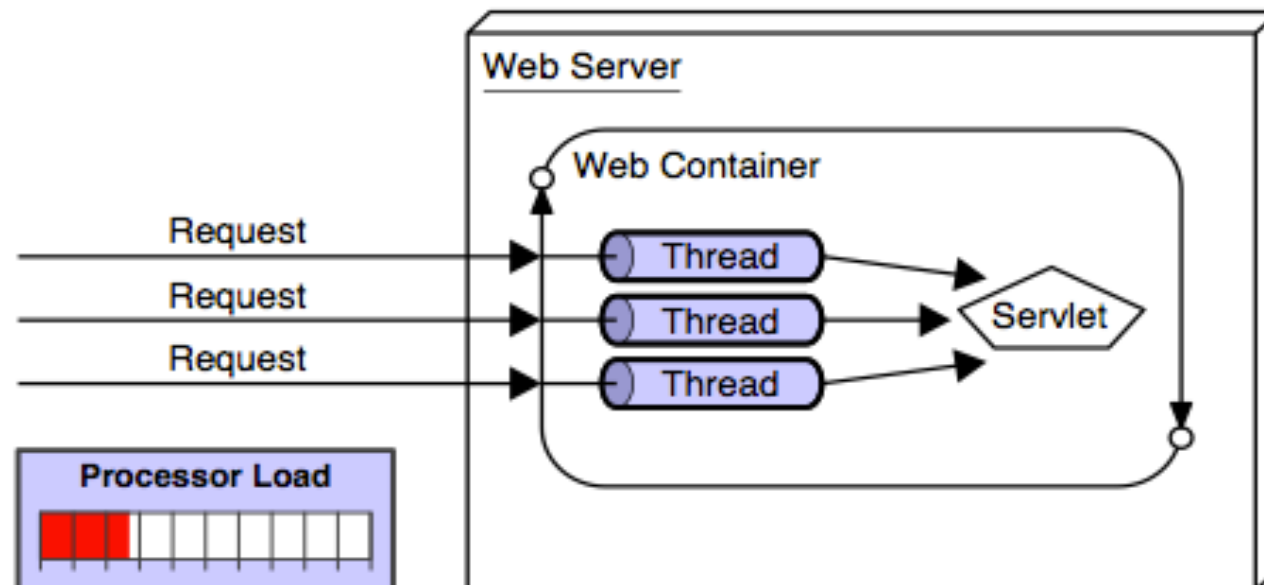
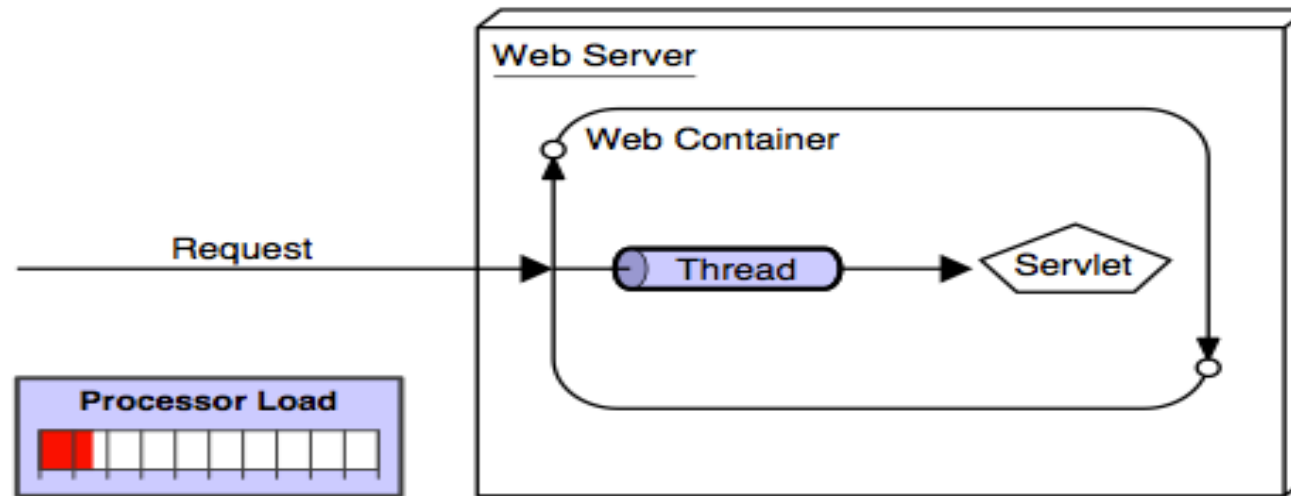
Inconvénients

- Chaque Shell est trop lourde.
- Logique métier est mélangée avec HTML (présentation).
- Le langage n' est pas tjrs sécurisé ou OO.
- Le langage n' est pas tjrs de plateforme indépendante.

LES SERVLETS JAVA



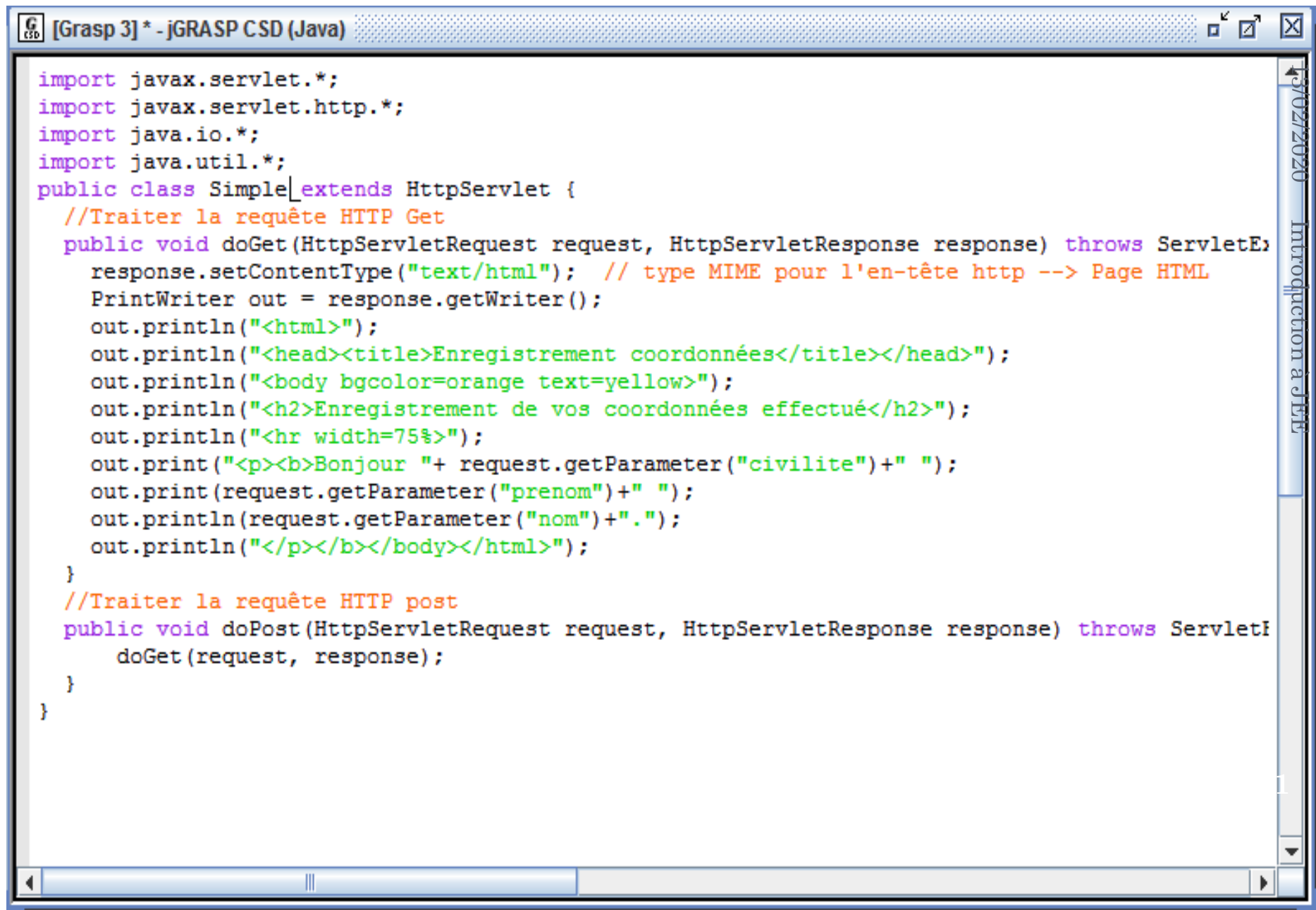
LES SERVLETS JAVA



LES SERVLETS JAVA

- Une servlet est un composant implémentant l'interface `javax.servlet.Servlet`.
- Son invocation est la conséquence de la requête du client, dirigé vers cette servlet
- Le serveur Web reçoit une demande adressée à une servlet sous la forme d'une requête HTTP
- Il transmet la requête à la servlet concernée, puis renvoie la réponse fournie par celle du client .
- La servlet construit la réponse et renvoie sous forme de code HTML.

LES SERVLETS JAVA



```
[Grasp 3] * - jGRASP CSD (Java)
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
import java.util.*;
public class Simple extends HttpServlet {
    //Traiter la requête HTTP Get
    public void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException {
        response.setContentType("text/html"); // type MIME pour l'en-tête http --> Page HTML
        PrintWriter out = response.getWriter();
        out.println("<html>");
        out.println("<head><title>Enregistrement coordonnées</title></head>");
        out.println("<body bgcolor=orange text=yellow>");
        out.println("<h2>Enregistrement de vos coordonnées effectué</h2>");
        out.println("<hr width=75%>");
        out.print("<p><b>Bonjour " + request.getParameter("civilite") + " ");
        out.print(request.getParameter("prenom") + " ");
        out.println(request.getParameter("nom") + ".");
        out.println("</p></b></body></html>");
    }
    //Traiter la requête HTTP post
    public void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException {
        doGet(request, response);
    }
}
```

6-LES JAVA SERVER PAGES (JSP)

- PHP (PHP Hypertext Preprocessor)

```
<? for ( $i=0; $i<10; $i++ ) { ?>  
    <TR><TD><? echo $i ?></TD><TD><? echo ($i * $i) ?></TD></TR>  
<? } ?>
```

- ASP (Active Server Pages)

```
<% FOR I = 0 TO 10 %>  
    <TR><TD><%= I %></TD><TD><%= (I * I) %></TD></TR>  
<% NEXT %>
```

- JSP (JavaServer Pages)

```
<% for ( int i=0; i<10; i++ ) { %>  
    <TR><TD><%= i %></TD><TD><%= (i * i) %></TD></TR>  
<% } %>
```


LES JAVA SERVER PAGES (JSP)

- Les JavaServer Pages, ou JSP, servent, comme les servlets, à créer du contenu Web de manière dynamique.
- Ces deux types de composants représentent à eux seuls un très fort pourcentage du contenu des applications Web.
- Les JSP sont des documents de type texte, contenant du code HTML ainsi que des scriptlets (et/ou des expressions), c'est-à-dire des morceaux de code Java.
- Les développeur des pages JSP peuvent mélanger du contenu statique et du contenu dynamique

LES JAVA SERVER PAGES (JSP)

```
<HTML>
<HEAD>
<TITLE>La date par Jsp</TITLE>
</HEAD>
</BODY>
<font size="+3">
<p> Démonstration de la génération de page dynamique </p>
<%@ page language="java" import = "mesdates.*" %>
<font size=6 color="red">
    <%= ladate.getMessage() %>
    <br>
    <%= ladate.getDate() %>
</font> <br>
    <%!
        String [] semaine = {"dimanche", "lundi", "mardi", "mercredi", "jeudi", "vendredi", "samedi"};
    %>
<%
    int jour = ladate.getJour() - 1 ;
    out.println (semaine[jour]);
    if (jour != 1) out.println (" , on travaille aujourd'hui");
    else out.println (" , c'est férie aujourd'hui");
%>
<font>
<form method="get">
changer le message : </br>
    <input type="text" name="message" size="50">
    <input type="submit" value="Valider">
</form>
<IMG SRC="<%= ladate.getDrapeau() %> "/>
<form method="get">
    choisissez votre pays :
</br>
    <select NAME="pays">
        <option VALUE=0> france&nbsp;
        <option VALUE=1 selected>angleterre&nbsp;
        <option VALUE=2>bretagne&nbsp;
```

LES JAVA SERVER PAGES (JSP)

- Les pages JSP s'exécutent, en fait, sous la forme de servlets.
- Donc elles disposent du même cycle de vie.
- Les pages JSP simplifient la création de pages générées dynamiquement sur un serveur HTTP, en utilisant une démarche opposée à celle des servlets.
- Dans la démarche de la plateforme J2EE :
les servlets serviront à traiter les requêtes des clients
les pages JSP serviront plus à la présentation.

7-LES ENTREPRISE JAVABEANS

- Généralement, lorsque parlons de Java EE, nous pensons immédiatement au EJB
- Les EJB sont des composants Java qui implémentent la logique métier de l'application.
- L'architecture Java EE comporte un serveur qui sert de conteneur pour les EJB.
- Ce conteneur charge tous les composants à la demande et invoque les opérations qu'ils exposent, en appliquant les règles de sécurité et en contrôlant les transactions.

LES ENTREPRISE JAVABEANS

- Le conteneur d'EJB fournit automatiquement toutes la plomberie et le câblage nécessaire pour la réalisation d'applications d'entreprise.
- La création des EJB ressemble beaucoup à celle des objets RMI.
- Ils existent plusieurs types d' EJB

RMI

