



Java avancé

Les JSPs

Objectifs

- Comprendre le rôle d'une JSP
- JSP et architecture MVC
- Savoir réaliser une JSP simple
- Savoir faire communiquer une Servlet et une JSP



Qu'est-ce que c'est ?

- Une JSP, ou JavaServer Page, est une surcouche de HTML
 - ➔ Autrement dit, la syntaxe de ces fichiers sera la même que le HTML, avec des fonctionnalités supplémentaires.



Qu'est-ce que c'est ?

- Une JSP, ou JavaServer Page, est une surcouche de HTML
 - ➔ Autrement dit, la syntaxe de ces fichiers sera la même que le HTML, avec des fonctionnalités supplémentaires.
- Ils permettent de créer la Vue de notre application.



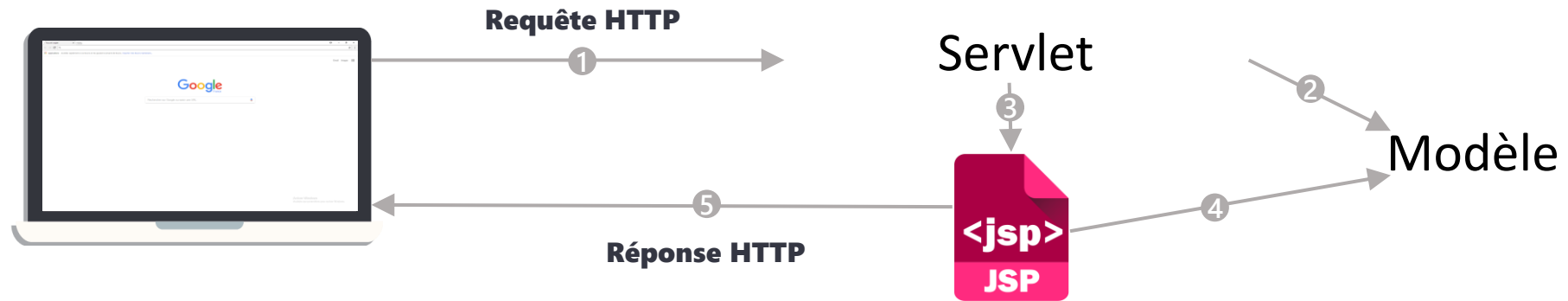
Qu'est-ce que c'est ?

- Une JSP, ou JavaServer Page, est une surcouche de HTML
 - ➔ Autrement dit, la syntaxe de ces fichiers sera la même que le HTML, avec des fonctionnalités supplémentaires.
- Ils permettent de créer la Vue de notre application.
- On n'y retrouve donc aucun traitement en dehors des instructions d'affichage.

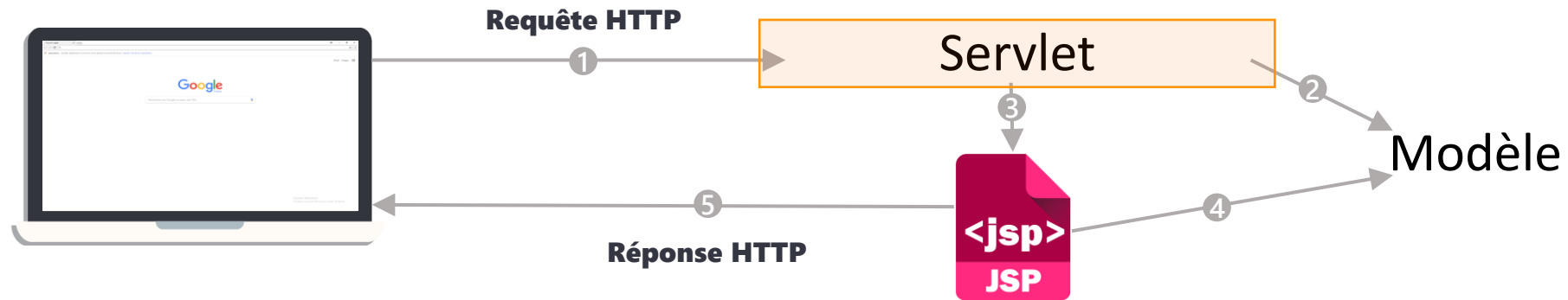
A partir d'ici on ne fera plus que des pages JSP qui remplacera HTML



Intégration dans une architecture MVC

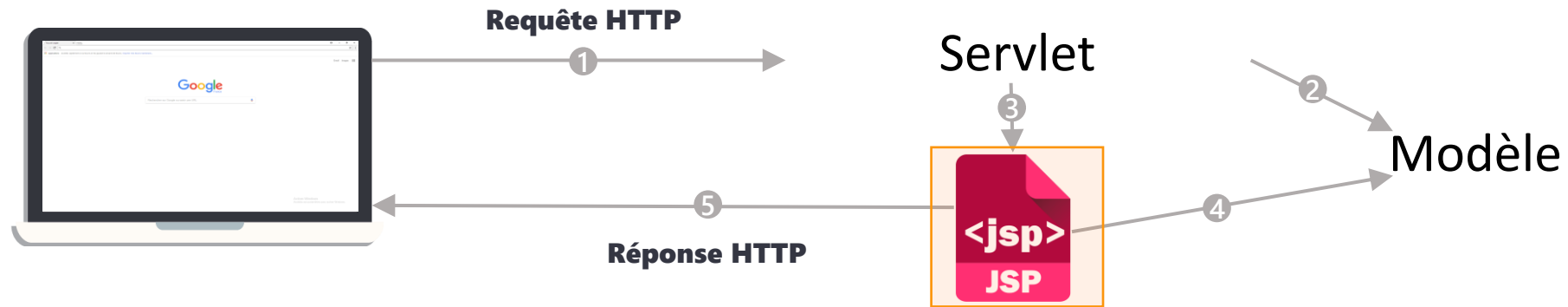


Intégration dans une architecture MVC



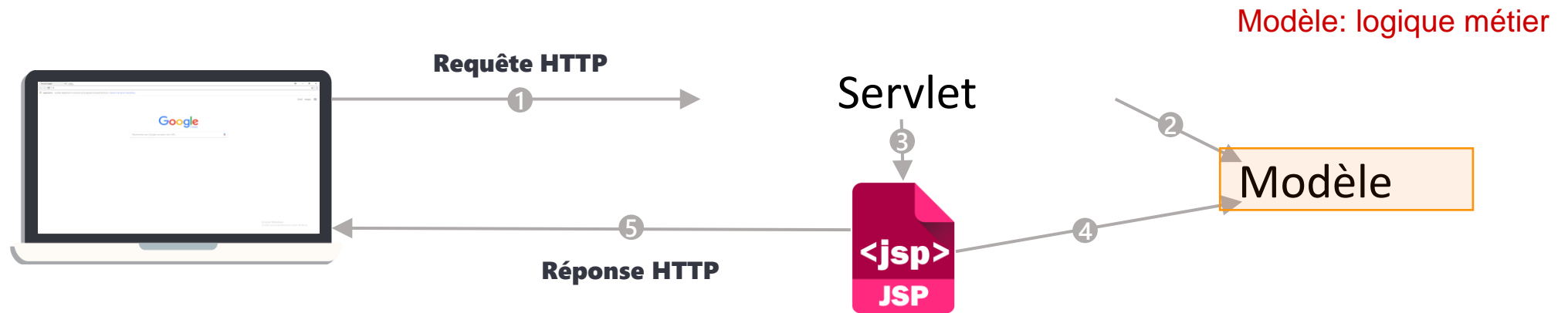
La Servlet représente le **Contrôleur** : c'est le composant de notre application qui réceptionne les saisies utilisateurs et envoie les informations à afficher.

Intégration dans une architecture MVC



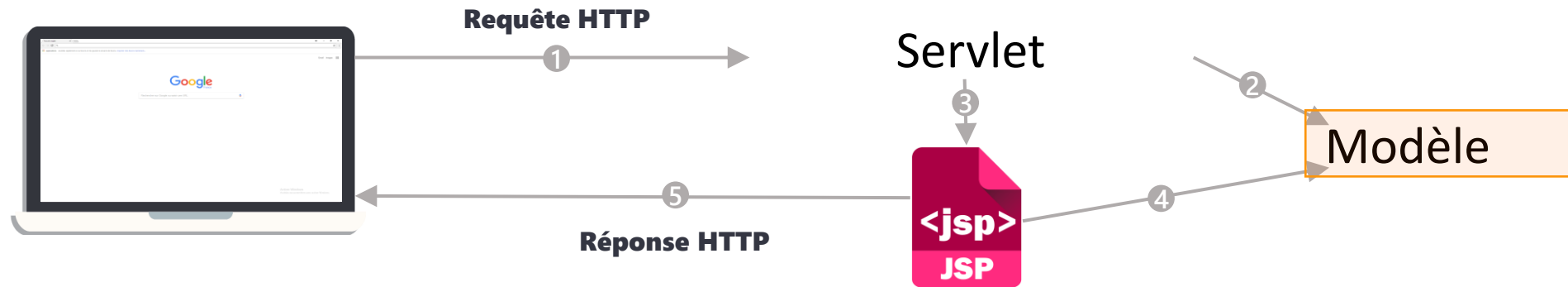
La JSP représente la **Vue** : c'est le composant de notre application qui détermine comme nos écrans sont censés s'afficher.

Intégration dans une architecture MVC



Enfin, le **Modèle** de notre application représente les données de l'application, ainsi que les traitements métiers associés à ces données.

Intégration dans une architecture MVC



Cette architecture est appelée « Modèle – Vue – Contrôleur » et est extrêmement courante sur les projets informatiques.

La création d'une page JSP

- Une directive "page"
- Un contenu HTML
- `<%@ page language="java"` <%@ est une directive

```
    contentType="text/html; charset=UTF-8"  
    pageEncoding="UTF-8"%>
```

```
<html>
```

```
<head>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
```

```
<title>Titre de la JSP</title>
```

```
</head>
```

```
<body>
```

```
</body>
```

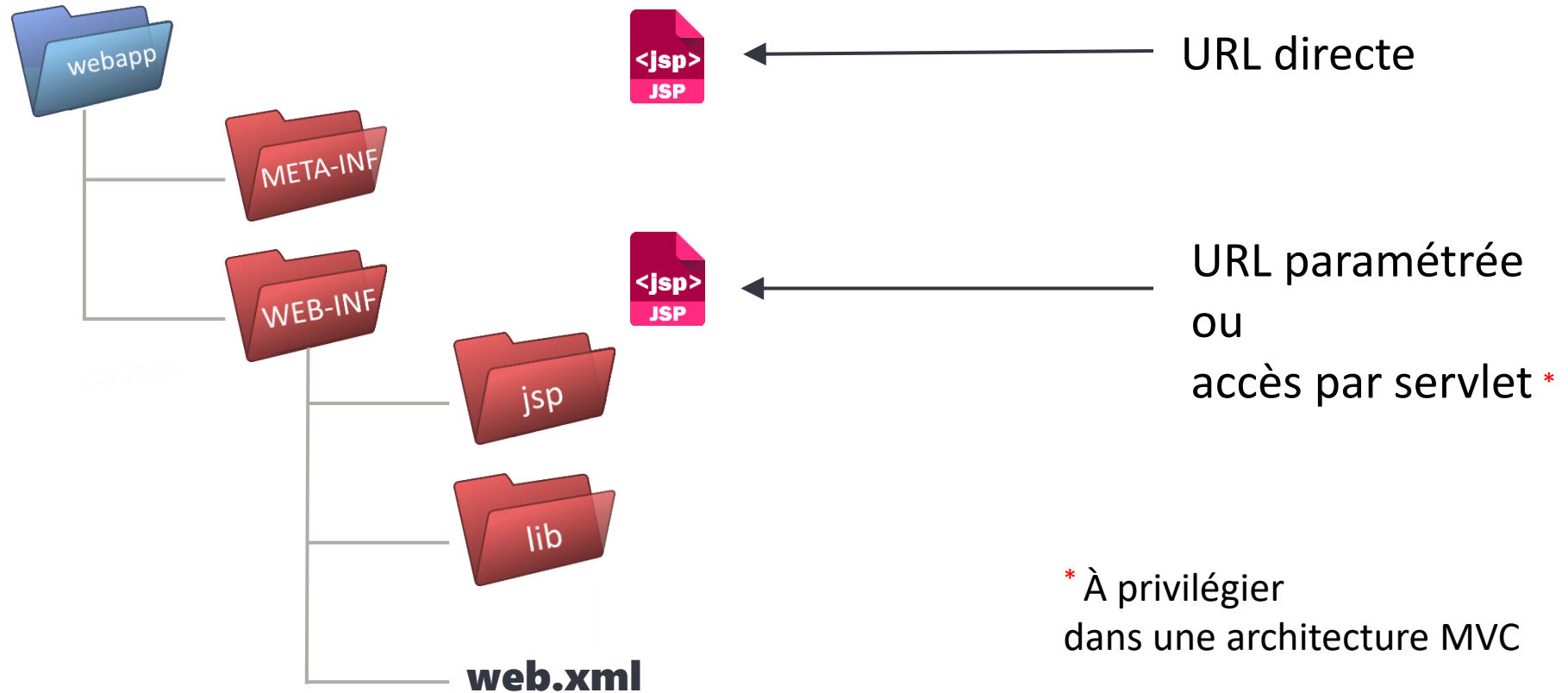
```
</html>
```



Les pages JSP vont aller dans le WEB-INF qui est privé (non visible dans l'URL)
Les Servlets ont accès aux variables du WEB-INF.



Les URL d'accès ?




Le paramétrage de l'URL

Peu élégant

web.xml

```
<!-- Déclaration d'une JSP -->
<servlet>
  <servlet-name>uneJSP</servlet-name>
  <jsp-file>/WEB-INF/uneJSP.jsp</jsp-file>
</servlet>

<!-- Association de la JSP à une ou des URL -->
<servlet-mapping>
  <servlet-name>uneJSP</servlet-name>
  <url-pattern>/url/de/la/jsp</url-pattern>
</servlet-mapping>
```



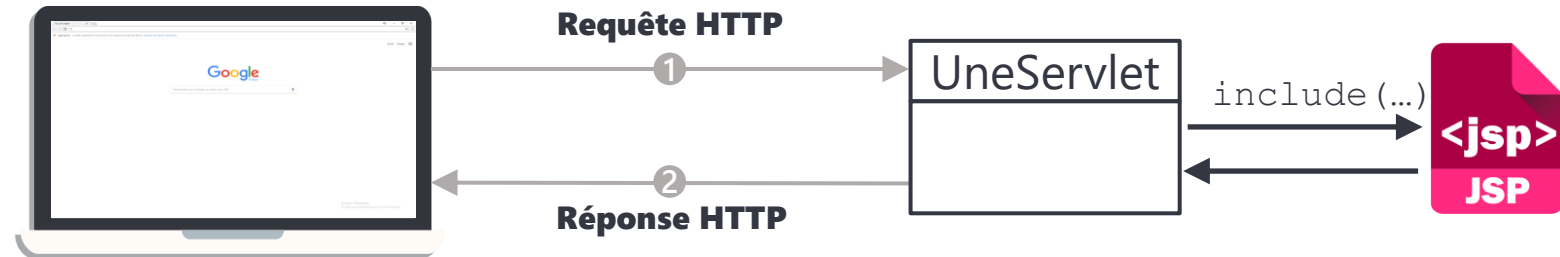
Demo

La première JSP

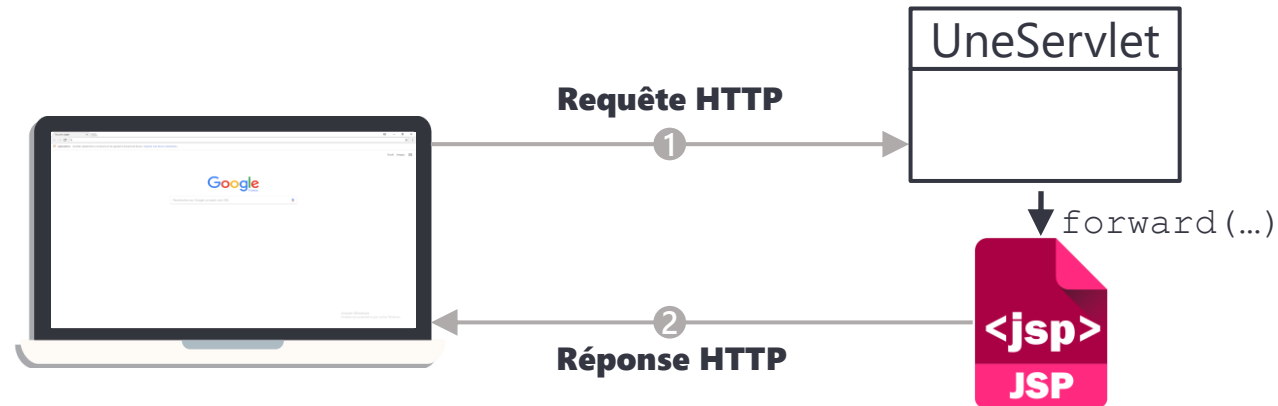


L'appel depuis une servlet

- Par inclusion



- Par délégation (MVC)



Les méthodes en jeu

Le request dispatcher est un objet fourni par le conteneur Web.
Il permet d'inclure ou de déléguer des traitements lors de la prise en charge d'une requête HTTP.
https://gayerie.dev/epsi-poei-201705/08_requestdispatcher_mvc.html



RequestDispatcher
<<interface>>

```
include(HttpServletRequest request,  
        HttpServletResponse response)
```

```
forward(HttpServletRequest request,  
        HttpServletResponse response)
```



Exemple d'inclusion et de délégation

Choisir l'une des 3

//Récupération du disptacheur

RequestDispatcher rd;

Le + simple

rd = request.getRequestDispatcher("URL");

rd = this.getServletContext().getRequestDispatcher("URL");

rd = this.getServletContext().getNamedDispatcher("Nom de la ressource");

JSP ou autre

//traitement...

//inclusion

rd.include(request, response);

//suite traitement

//traitement

//forward

rd.forward(request, response);

⚠ //terminé

Pas de traitement après un forward (<=> return)



Demo

La délégation



La communication entre la servlet et la JSP

- Dans la servlet

```
request.setAttribute("cle", "valeur de n'importe quel type");
```

attribut: ce que l'on va ajouter dans la Servlet.
Avantage: pas que des String mais tous les types

- Dans la JSP

```
${cle }
```

Expression langage

setAttribute("String",n'importe)

- La durée de vie est limitée à la requête



Demo

La communication entre la Servlet et la JSP



La directive page

```
<%@ page language="java"  
contentType="text/html; charset=UTF-8"  
pageEncoding="UTF-8"  
errorPage="nom de la page d'erreur associée"  
%>
```

```
<%@ page import="nom de la classe" %>
```



Atelier Annuaire pro





Java avancé

Les JSPs