



Développement Avancé d'Applications Web

– Chapitre 1 –

Modèle MVC et Application Web JEE

Dr Bouanaka Chafia

NTIC

chafia.bouanaka@univ-constantine2.dz

Etudiants concernés

Faculté/Institut	Département	Niveau	Spécialité
Nouvelles technologies	TLSI	Licence 3	Génie Logiciel (GL)

Prérequis :

- La programmation Orientée Objets
- Le langage Java

Objectifs Spécifiques du cours

- Comprendre le patron de conception **MVC(Modèle/Vue/Contrôle)**
- Structurer les composantes d'une application Web selon de modèle MVC
- Connaitre l'implémentation JEE du modèle MVC

Section 1

Modèle MVC

Modèle MVC

Patron de Conception

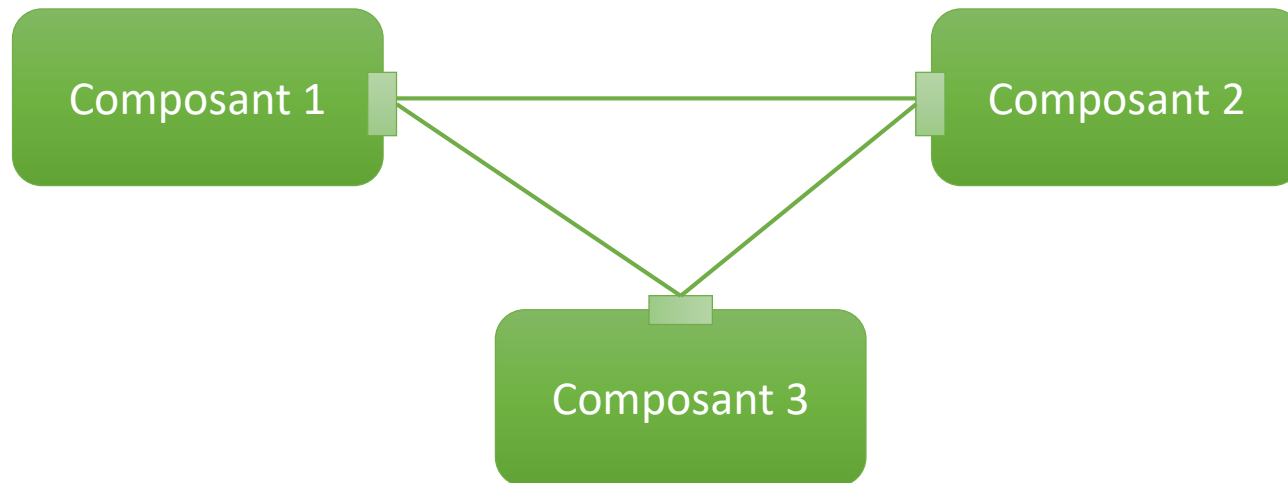
- Les **patrons de conception** (design patterns) décrivent des solutions standards pour répondre à des problèmes d'**architecture** et de **conception des logiciels**.
- Un **patron de conception** est un concept destiné à résoudre **des problèmes récurrents** suivant le paradigme objet;
- Un Patron de conception est une solution standard pour répondre à type ou une catégorie de problèmes.
- C'est une architecture logicielle pour structurer une application.

Modèle MVC

Architecture d'un Système Logiciel

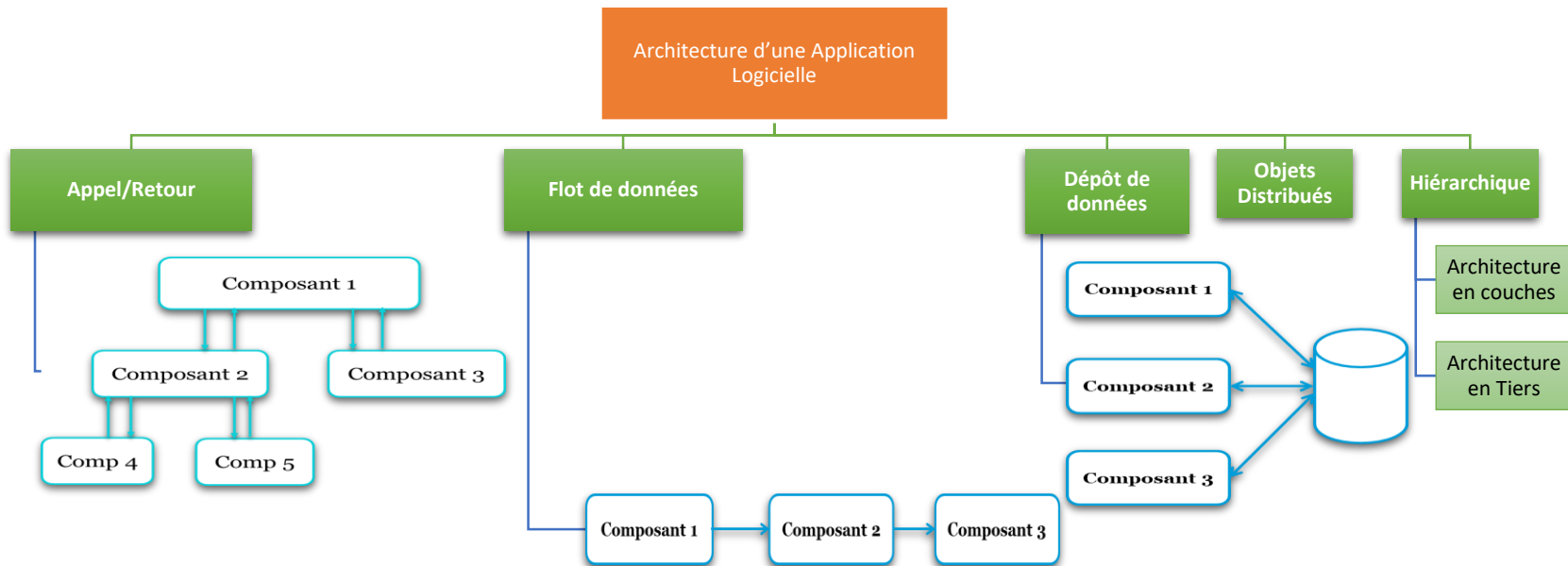
L'architecture d'un système logiciel décrit son organisation générale :

- **ses composants**
- les **relations** entre ses composants et avec l'environnement
- les principes de **conception** et d'**évolution** du système



Styles d'Architecture

- Un style d'architecture est caractérisé par une forme particulière d'**interaction** entre ses composants.
- Cinq catégories principales d'architectures sont alors identifiées:



Modèle MVC

Motivation du modèle MVC

Application Interactive : constituée de deux grandes parties :

- **Front-end** (la partie visible) : l'IHM (Interface Homme-Machine), concerne deux parties :

- **les entrées** : actions de l'utilisateur;
- **les sorties** : la visualisation des résultats de l'action de l'utilisateur

- **Back-end** (la partie cachée) : englobe :

- le stockage,
- l'accès et le traitement des données.
- C'est le **Noyau de l'application**

- Décomposition de l'application en deux parties :

- La présentation;
- Les données et les traitements associés

Modèle MVC

Motivation du Modèle MVC

Le patron architectural MVC concerne les applications interactives et vise à séparer (**découpler**) les deux éléments :

- Données et Traitements,
- Présentation.
- Ce découplage vise à :
 - Isoler la donnée elle-même de sa présentation ;
 - Distinguer la consultation de la modification.
- Ce découplage facilite :
 - l'évolution des interfaces utilisateur,
 - Proposer plusieurs versions d'interfaces adaptées à différents contextes.
- **L'interaction entre les deux éléments est réalisé via une couche contrôle.**

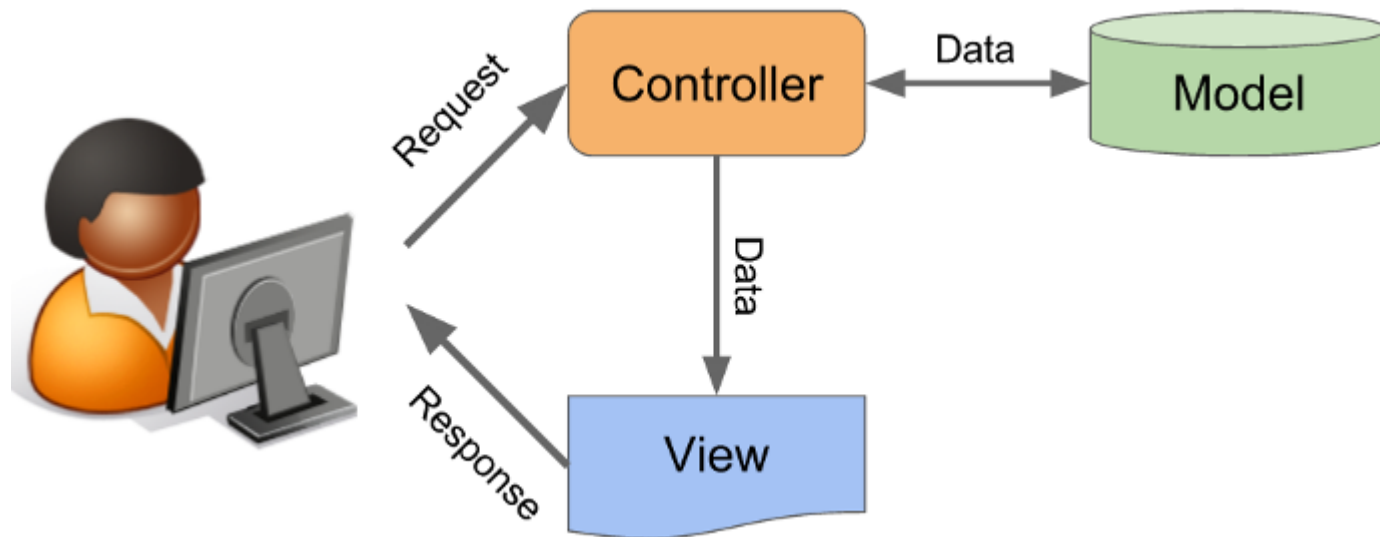
Modèle MVC

Motivation du Modèle MVC

Principe du modèle MVC

Les 3 composantes suivantes d'une donnée sont distinguées et séparées :

- Le modèle (la structure de la donnée et les traitements associés)
- La vue (la représentation de la donnée pour affichage)
- Le contrôleur (les moyens de modifier la valeur)



Modèle MVC

La couche Modèle

Le Modèle définit les fonctionnalités de l'application :

- Représentation des données ;
- Accès aux données ;
- Traitement des données.
- **Rôle**
 - Encapsuler les propriétés d'une donnée
 - Être indépendant des vues et des contrôleurs
- **Conséquences**
 - Définir les accesseurs (les getters et les setters dans classe) aux propriétés de cette donnée.

Modèle MVC

La couche Vue

La vue affiche les données, provenant exclusivement du modèle, à l'utilisateur et/ou reçoit ses actions.

- Aucun traitement autre que la gestion de la présentation n'est réalisé dans cette couche.
- **Rôle**
 - Récupérer les données utilisateur (les entrées) et les transmettre au modèle à travers la couche contrôle ;
 - Visualiser les données issues du Modèle;
 - Assurer la consistance de la visualisation avec la représentation des données dans le modèle.
- **Conséquences**
 - La vue interagit avec le contrôleur pour mettre à jour l'affichage si le modèle change.

Modèle MVC

La couche Contrôle

La couche **contrôle sert à traiter les événements en provenance de l'interface utilisateur pour :**

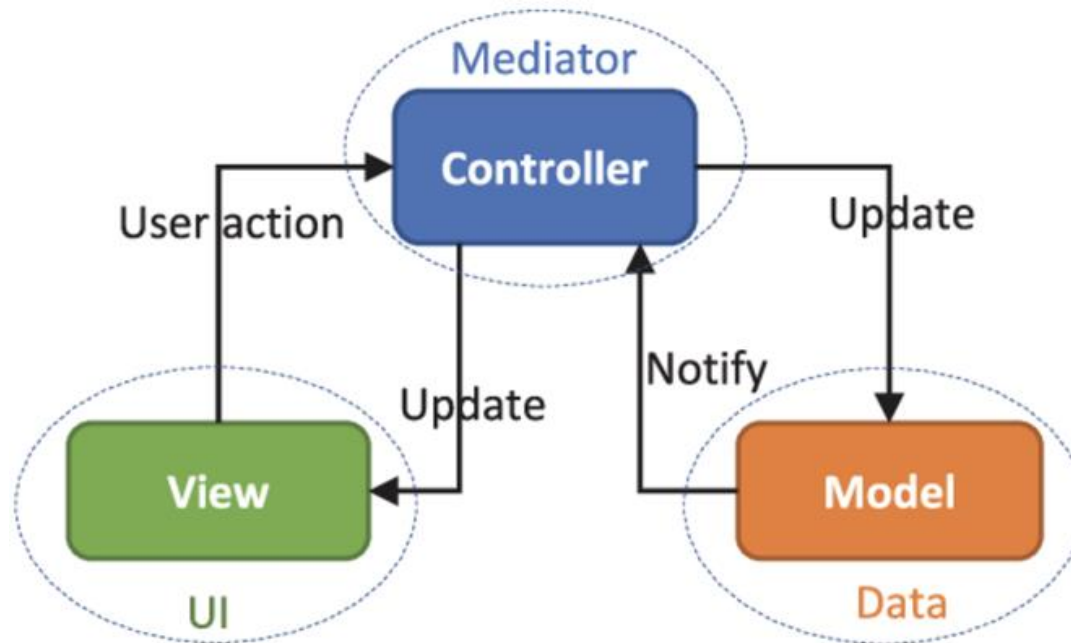
- les transmettre au modèle pour le faire évoluer
- ou à la vue pour modifier son aspect visuel (pas de modification des données affichées mais des modifications de présentation (couleur de fond, affichage ou non de la légende d'un graphique, ...).

Rôle :

- Gestion des entrées de l'application :
 - **Comportement de l'application** face aux actions de l'utilisateur (ex: clic sur un bouton submit);
 - Traduction des actions de l'utilisateur en actions sur le modèle;
- Traduction des **résultats des traitements** du modèle (réactions) et des **actions de l'utilisateur** en action sur la vue (modification appropriée de la vue)

Modèle MVC

La couche Contrôle



Model-View-Control(MVC) design pattern.

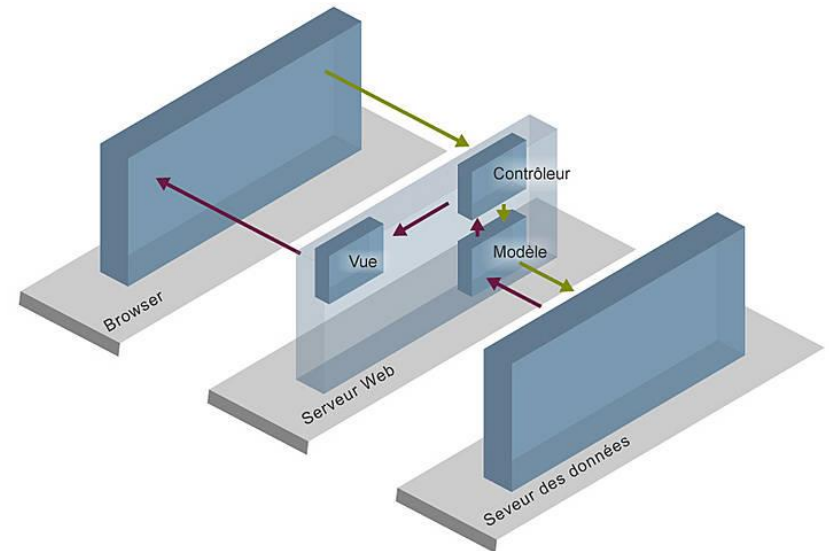
Section 2 :

Modèle MVC pour les Applications Web

Modèle MVC pour les Applications Web

Architecture d'une Application Web

- Une application web est basée sur une architecture Client serveur à **3-tiers** (Client / Serveur Web / Serveur de base de données)
- L'architecture MVC d'une application Web est constituée de trois parties ou tiers qui sont toutes déployées dans le serveur Web:

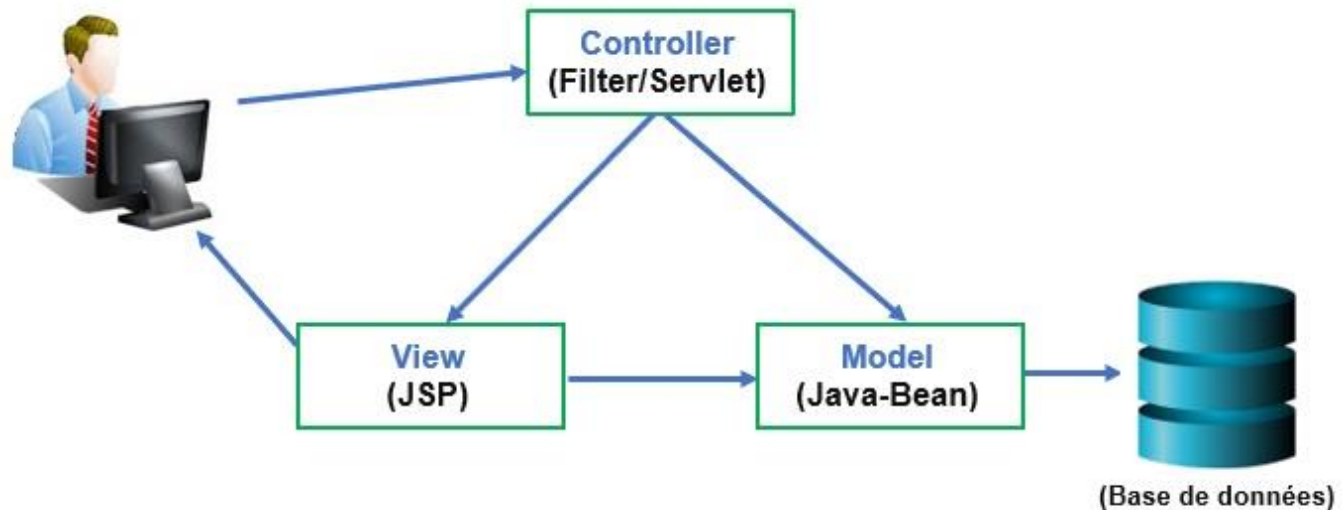


- **La vue (Présentation)** : les pages Web ;
- **Le Modèle (Métier)** : regroupe :
 - **Les traitements** de l'application Web ;
 - **Les Données** : regroupe toutes les données manipulées par l'application Web ainsi que leur représentation (la Base de Données).
- **Le Contrôle** : l'intermédiaire entre la vue et le modèle, il réalise les interactions entre les deux couches.

Modèle MVC pour les Applications Web

Principe de fonctionnement :

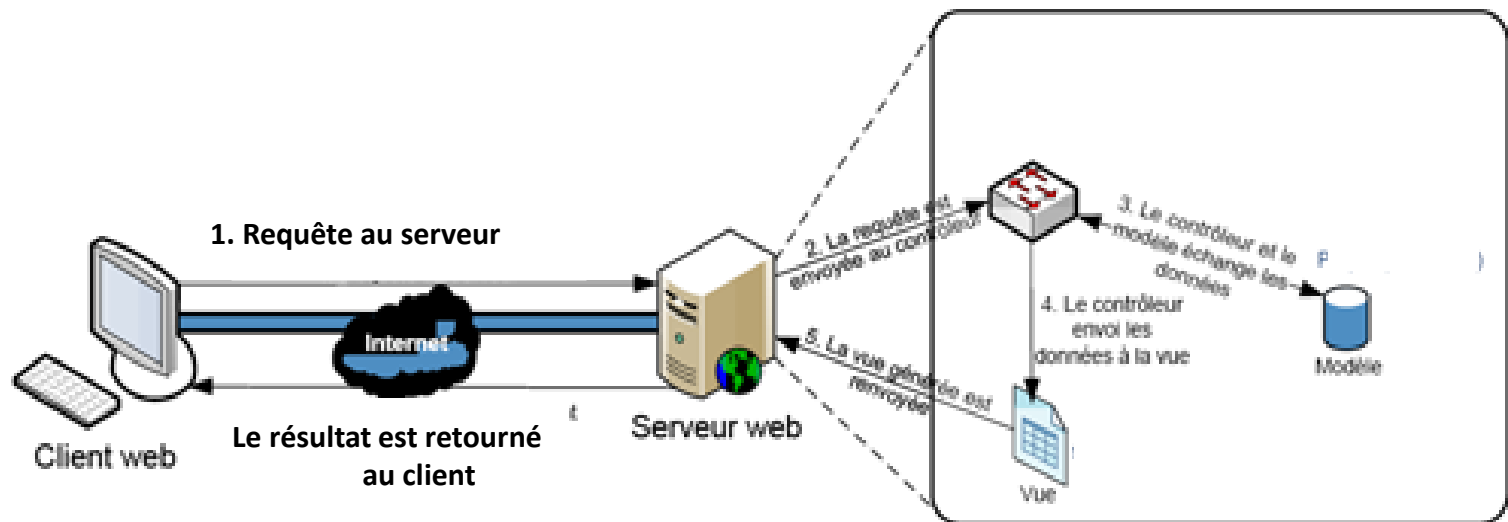
- Par le biais d'une **page web**, l'utilisateur émet une requête HTTP au **serveur web** en cliquant sur un lien ou sur un bouton.



Modèle MVC pour les Applications Web

Principe de fonctionnement :

- La requête du client est prise en charge par le contrôleur.
- Le **contrôleur** déclenche les traitements définis dans le **modèle** et qui sont associés à la requête de l'utilisateur.
- Le **contrôleur** sélectionne ensuite la page web affiche de la réponse et lui transmet les entités (objets) contenant les données à afficher.
- La page web construit la réponse en faisant appel aux accesseurs des entités.
- La réponse HTTP est transmise au **navigateur client** qui l'affiche sous forme de page web.



Modèle MVC pour les Applications Web

La couche Contrôle

Définition (Contrôle)

Le contrôleur gère les interactions avec l'utilisateur :

- s'occupe de la réécriture des URLs
- détermine quels traitements doivent être réalisés
- Identifie les pages Web à envoyer au navigateur Web de l'utilisateur pour les afficher

Modèle MVC pour les Applications Web

La couche Vue

Définition (Vue)

- concerne la forme (représentation)
- la vue concerne principalement la représentation des données du modèle à l'utilisateur
- il peut donc exister plusieurs vues

Modèle MVC pour les Applications Web

La couche Modèle

Définition (Modèle)

- Englobe la logique métier de l'application Web
- Définit la représentation des données et les traitements qui lui sont associés

Section 3 :

Implémentation JEE du Modèle MVC

Implémentation JEE de MVC

Définition

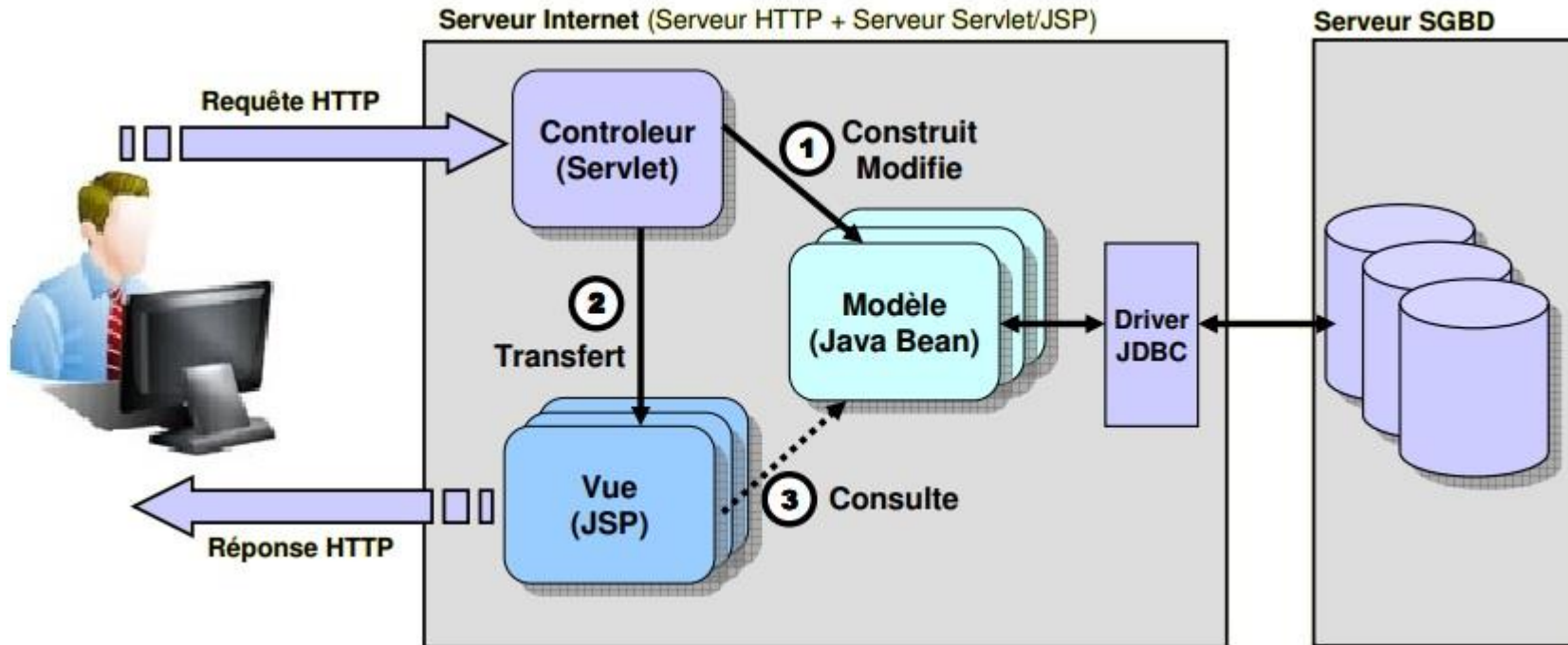
JEE : Java Enterprise Edition

- Le modèle MVC est implémenté dans JEE par les entités :
 - La vue : les pages Web (HTML & JSP)
 - Le contrôle : des servlets
 - Le modèle : des classes java (des EJB et des JavaBean)
- L'environnement JEE fournit un ensemble d'APIs permettant de développer des sites Web dynamiques avec une technologie Java.
- définie par Sun <http://java.sun.com/>
- basée sur Java : Standard Edition, Enterprise Edition, Micro Edition
- applications types : systèmes d'information entreprise, commerce électronique...
- ensemble de technologies pour construire des applications réparties
- La dernière version est : JEE 8 (depuis 2017)

Implémentation JEE du Modèle MVC

Architecture d'une Application JEE

- Le modèle MVC est implémenté dans JEE par les entités suivantes :
 - La vue : **les pages Web** (HTML & JSP)
 - Le contrôle : **des servlets**
 - Le modèle : **des classes java** (des EJB et des JavaBean)



Implémentation JEE du Modèle MVC

MVC 1

- **Modèle**

- Des classes

- **Vue**

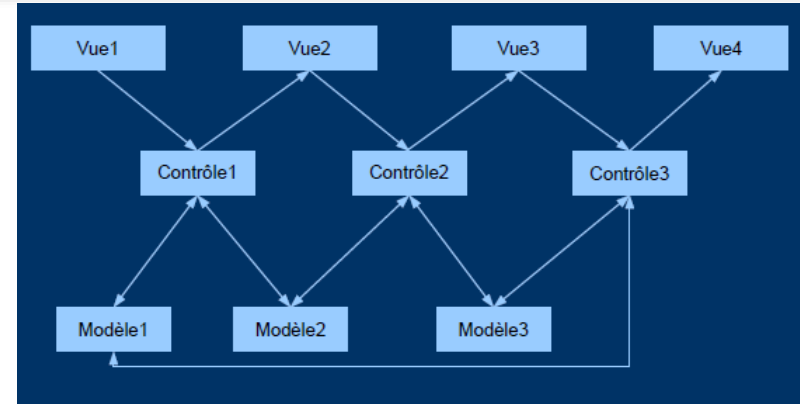
- Les pages web

- **Contrôleur**

- Une classe de contrôle qui traite l'information de la page source en modifiant les données si nécessaire
- Affiche la nouvelle page

→ **La vue sert à configurer le contrôle**

→ **Plusieurs Contrôleurs**



Implémentation JEE du Modèle MVC

MVC 1

- **Modèle**

- Des classes

- **Vue**

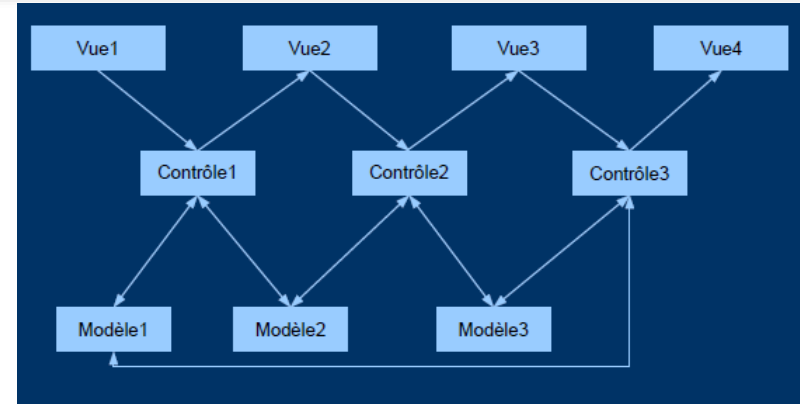
- Les pages web

- **Contrôleur**

- Une classe de contrôle qui traite l'information de la page source en modifiant les données si nécessaire
- Affiche la nouvelle page

→ **La vue sert à configurer le contrôle**

→ **Plusieurs Contrôleurs**



Implémentation JEE du Modèle MVC

MVC 2

Inconvénient MVC1 :

- Multiplication du nombre de contrôleurs
- Lourdeur de déclaration au niveau du Serveur d'Application
- Mises à jour peu faciles

•Modèle MVC2

- Un seul contrôleur servant d'aiguilleur vers le modèle et la vue
- Elle a l'architecture simple :

```
switch(pageSource) {
```

```
    case page1 : pageSuite=traitement1();afficher(pageSuite);break;
```

```
    case page2 : pageSuite=traitement2();afficher(pageSuite);break;
```

```
    ...
```

```
}
```

