

Création de site web

## **Sites web MVC**

**Damien Nouvel**

# Sites web MVC

## Plan



- Architecture MVC
- Modèle / base de données
- Vues / interface
- Contrôleur / application

# Sites web MVC

## Plan



- Architecture MVC
- Modèle / base de données
- Vues / interface
- Contrôleur / application

# Sites web MVC

## Architecture MVC

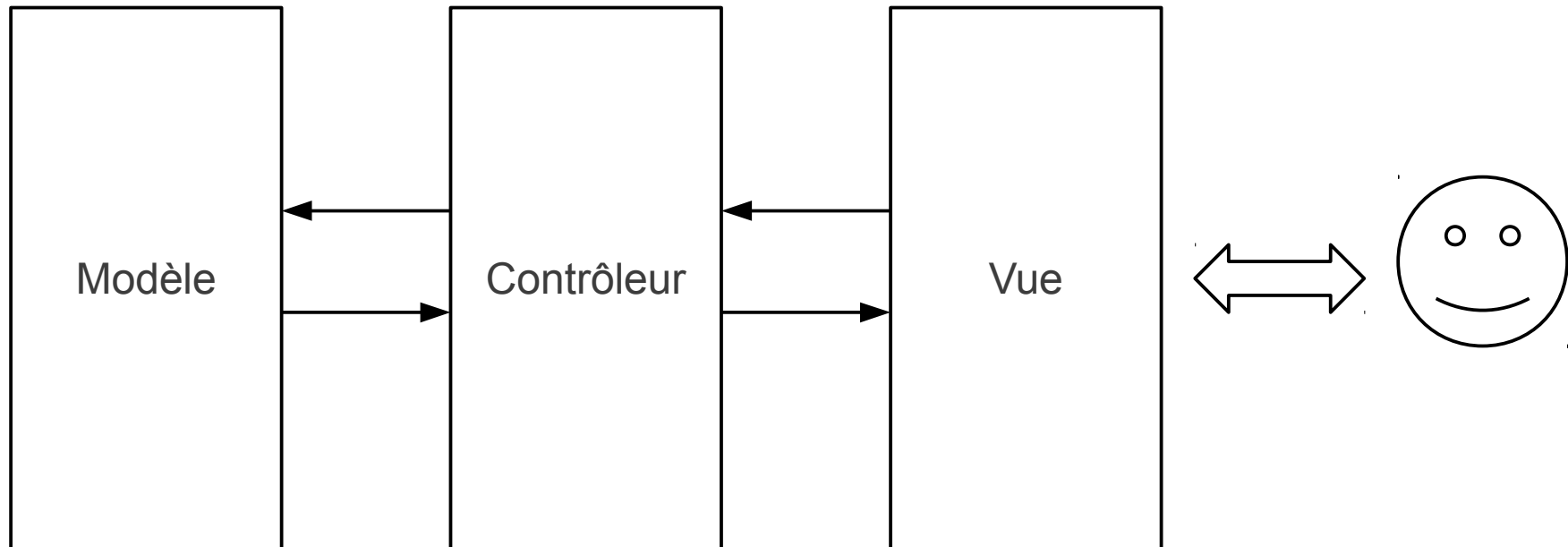


- MVC
  - Méthode de conception d'Interface Homme Machine (IHM) (Reenskaug, 1979)
  - Modèle - Vue - Contrôleur
    - Modèle : gestion des données et des fonction pour y accéder
      - Couche de persistance des données
    - Vue : affichage des données
      - Interfaces avec l'utilisateur
    - Contrôleur : synchronisation entre la vue et les données
      - Application, traitements, flux de données
  - Vue indépendante des données ( $\neq$  trois-tiers, couches)
  - Java (Swing, SWT, Spring, Struts), PHP (Symfony, Cake, Zend), Python (Django, Turbogears), Flash (Flex), Ruby (Rails)

# Sites web MVC

## Architecture MVC

- Schéma de fonctionnement



# Sites web MVC

## Plan



- Architecture MVC
- Modèle / base de données
- Vues / interface
- Contrôleur / application

# Sites web MVC

## Modèle / base de données

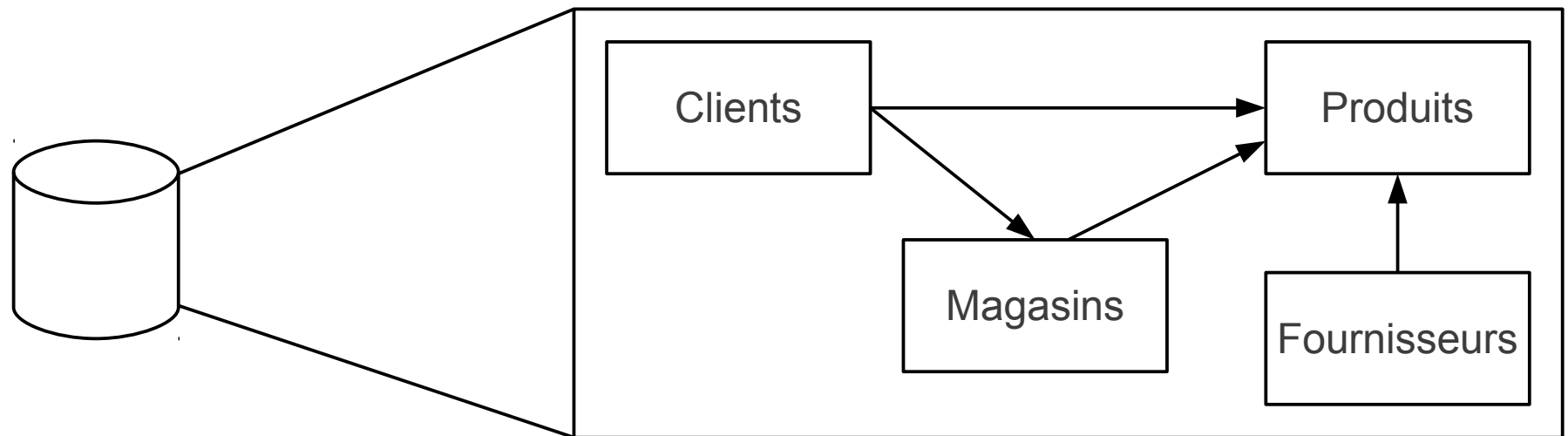


- Modèle (de données)
  - Description des « objets » manipulés par le système
    - Personnes, évènements, produits, instruments, etc.
  - Couche de persistance
    - Base de données
    - Fichiers (XML, CSV...)
  - Moteur de persistance
    - Fonctions de manipulation des objets
      - Liste, lecture, écriture
    - Fonctions « standard » qui peuvent potentiellement être appliquées à tous les objets
    - Interface entre le langage (Java, PHP, .Net...) et les données
    - Optimisation en lecture, droits d'accès en écriture, etc.

# Sites web MVC

## Modèle / base de données

- Bases de données
  - Gestion des données standardisée
  - Structuration des données sous formes d' « objets »
    - Chaque objet est défini par un ensemble d' « attributs »
    - Les objets peuvent correspondre aux classes du langage
    - Des associations peuvent être faites entre les objets





# Sites web MVC

## Modèle / base de données

- Bases de données (suite)
  - Base de données « relationnelle »
    - Objets décrits par des « tables » (tableau Excel)
    - Chaque attribut est une colonne
      - Typée : entier, chaîne, booléen, etc.
    - Objets identifiés par des « clés »
      - Indexation pour accès rapide aux données, associations entre données

ID	Nom	Prénom	Email	Magasin	Newsletter	...
1	Nouvel	Damien	nouvel@...	3	o	
2	Dupond	Roger	r.dup@...	2	n	
3	Smith	Bob	...	5	n	
4	Colas	Régis	...	3	o	
...						

# Sites web MVC

## Modèle / base de données



- Bases de données (suite)
  - Langage de «requête » sur la base de données
    - SQL : Structured Query Language (Chamberlain & Boyce, 1970)
  - Exemple classique en PHP / MySQL

```
// Connexion à la base de données
mysql_connect('localhost', 'login', 'password');
mysql_select_db('nom_de_la_base',$db);
// Requête SQL
$res = mysql_query('SELECT prenom, nom FROM client');
// Traitement des résultats
while($client = mysql_fetch_assoc($res)){
    echo 'Client'.$client['prenom'].' '.$client['nom'] ;
    ...
}
```

# Sites web MVC

## Plan



- Architecture MVC
- Modèle / base de données
- Vues / interface
- Contrôleur / application

- Utilisation de « templates »
  - Présentation des données
  - Utilisation des inclusions pour assembler la page
  - Définit les possibilités d'interaction avec l'utilisateur
    - Liens, formulaires, pagination, etc.
  - Peut-être dépendante des objets à afficher
    - Liste des clients, fournisseurs, magasins
    - Détail d'un client, fournisseur, magasin
    - Formulaire pour modifier un client, fournisseur, magasin
  - Peut faire appel à des éléments génériques
    - Liste d'objets (avec pagination)
    - Élément de formulaire

- Utilisation de « templates » (suite)
  - Programmation de l'interface
  - Exemple classique en PHP

```
<ul>
<?php foreach($objets as $objet){ ?>
  <li>'
  
  <p class="titre"><?php echo $objet['nom']; ?></p>
  <p class="resume"><?php echo $objet['description'];?></p>
  <a class="detail" href="<?php echo $objet['lien']; ?>">Détails</a>
  </li>
<?php } ?>
</ul>
```

# Sites web MVC

## Plan



- Architecture MVC
- Modèle / base de données
- Vues / interface
- Contrôleur / application

# Sites web MVC

## Contrôleur / application



- Le contrôleur tient compte des évènements
  - Selon les interactions possibles avec l'interface
  - Lien entre la vue et les données
  - Par ex. :
    - L'utilisateur demande une page :
      - Sélectionner les données dans la base pour alimenter le modèle
      - Passer les éléments du modèle aux vues correspondant à la page
    - L'utilisateur demande une liste de produits :
      - Sélectionner les produits dans la base de données
      - Trier les produits (ordre par défaut ou demandé par l'utilisateur)
      - Passer les objets à la vue « liste »
    - L'utilisateur valide un formulaire pour un produit
      - Demander au modèle de mettre à jour / insérer le produit
      - Aller vers la vue succès / erreur

# Sites web MVC

## Contrôleur / application



- Le contrôleur définit l'application
  - Flux des données
  - Changements de vues / de pages
  - De manière classique, principales pages :
    - Dédiées (page d'accueil, contact, etc.)
    - Le « contenu » (section du site, article)
    - Les listes d'objets :
      - Mécanisme de liste paginée
      - Affichage d'élément de liste selon l'objet
      - Liens selon les droits (lecture, écriture, suppression)
    - Le détail d'un objet (spécifique)
    - La modification d'un objet
      - Même vue pour l'insertion ou la modification



# Sites web MVC

## Contrôleur / application



- Interactions complexes avec modèle / vues
  - Par ex. :

```
if($vue == 'liste-produits'){
    $produits = modele->getProduits($criteres);
    vue_centrale->afficherListe($produits);
    boites_droite->listeCourte($produits->produitsLies());
    if($utilisateur->isClient()){
        boites_droite->ajouterPromos();
    }
}

if($vue == 'detail-produit'){
    $produit = modele->getProduit($id);
    vue_centrale->afficherElement($produit);
    boites_droite->listeCourte($produit->accessoires());
}
```