CDAW - Back-end

— L'architecture MVC pour le Web —

Luc Fabresse

luc.fabresse@imt-lille-douai.fr

Plan

Modèle-Vue-Contrôleur

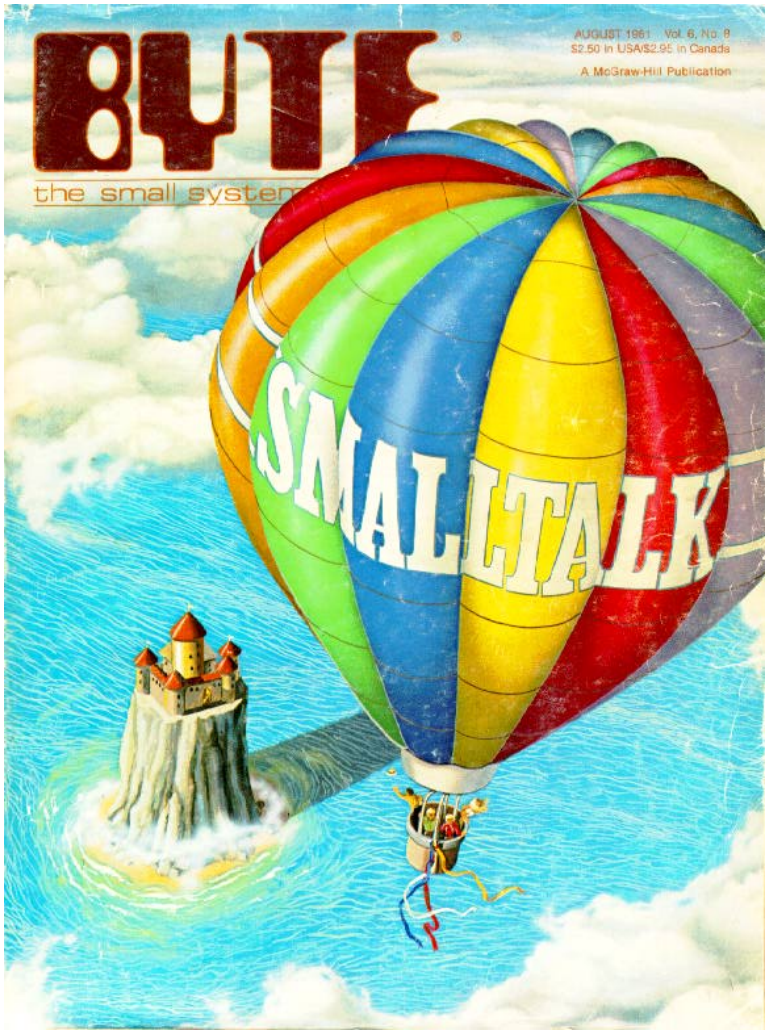
Modèle-Vue-Contrôleur est un patron architectural qui permet de structurer un programme afin que ce dernier soit réutilisable, maintenable et évolutif.

Principe du MVC

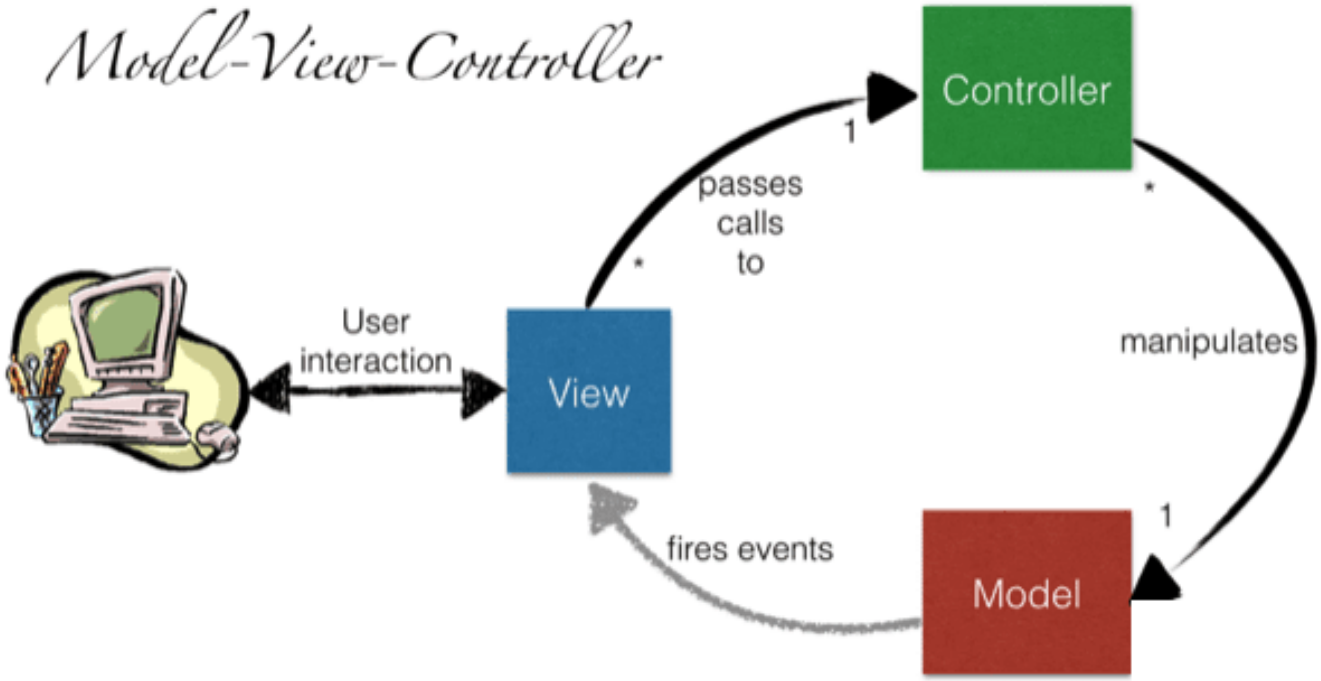
Séparer une application en trois “parties” logiques :

- Modèle : point d’accès permettant de manipuler les “objets” (généralement via une base de données)
- Contrôleur : gère la logique de l’application
- Vue : s’occupe de l’affichage des informations (met en forme les informations calculées par le contrôleur)

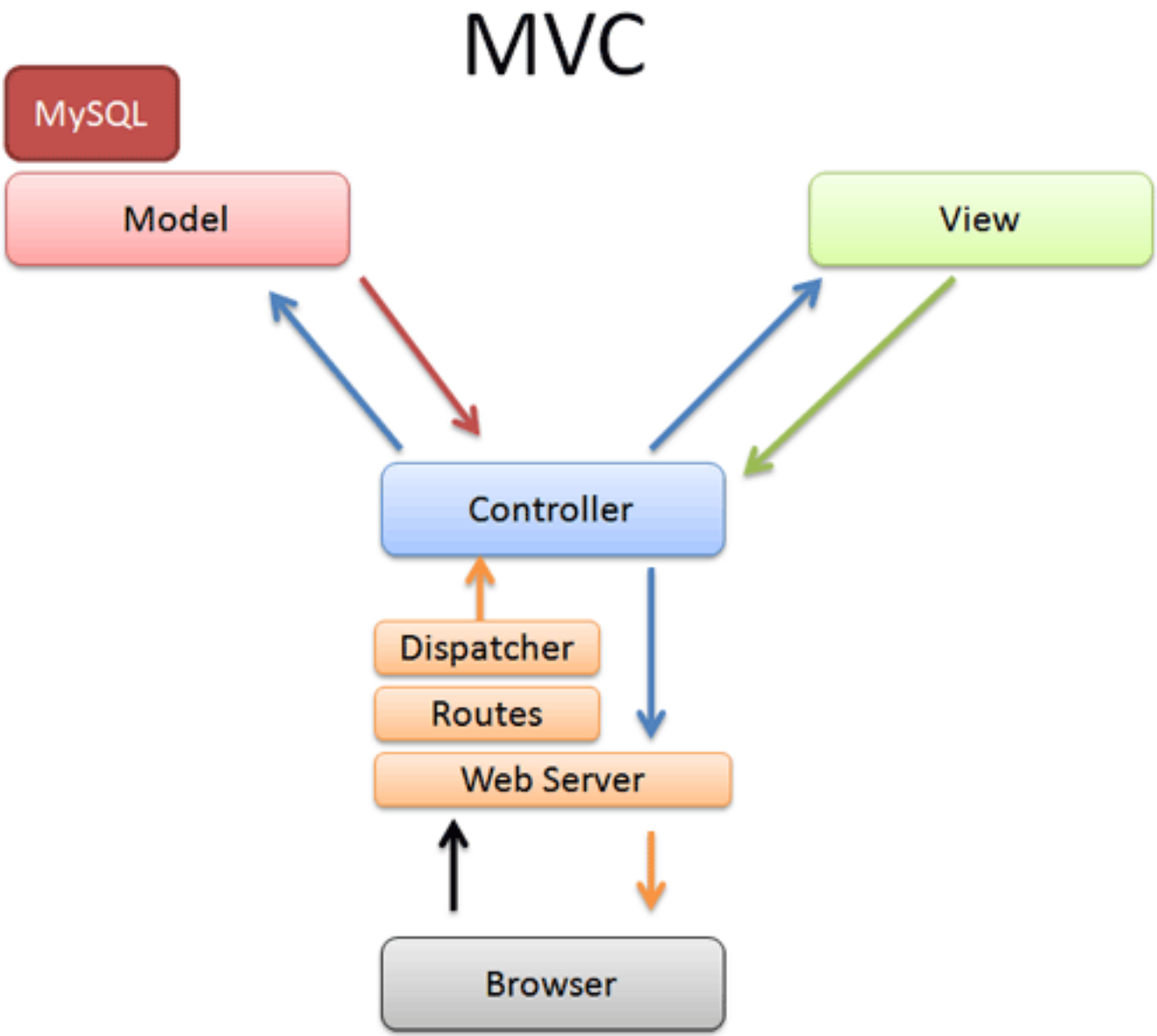
Historique de MVC



Principe général de MVC



MVC pour le Web



Utilisation de “Frameworks Web”

Un framework est un ensemble de classes, d’outils et de conventions qui peuvent être étendus ou adaptés par un développeur afin de développer une application spécifique.

Propriétés des “Frameworks Web”

Par rapport à une bibliothèque classique, un framework se distingue par les propriétés suivantes :

Inversion de contrôle Le framework contrôle le workflow de l’application et appelle le code écrit par celui qui implémente l’application cible

Extensibilité Un framework peut être étendu (souvent par héritage et redefinition dans les frameworks objets) afin qu’un développeur puisse intégrer le code spécifique de l’application qu’il développe. Il faut bien comprendre que le code du framework lui même n’est pas modifé (voir modifiable).

Exemples de Frameworks Web PHP basés sur MVC

Les plus connus / utilisés

- Zend Framework
- Laravel
- Symfony 2
- CakePHP
- Lithium
- PHPOnTrax
- CodeIgniter
- ...

De plus “simples”

- Slim `https://www.slimframework.com/`
- Silex `https://silex.symfony.com/`

Exemples de Frameworks Web PHP basés sur MVC

Les plus connus / utilisés

- Zend Framework
- Laravel
- Symfony 2
- CakePHP
- Lithium
- PHPOnTrax
- CodeIgniter
- ...

De plus “simples”

- Slim `https://www.slimframework.com/`
- Silex `https://silex.symfony.com/`

Dans la suite de ce cours :

Aucun Framework Web ne sera (devra être) utilisé

Objectifs de ce cours

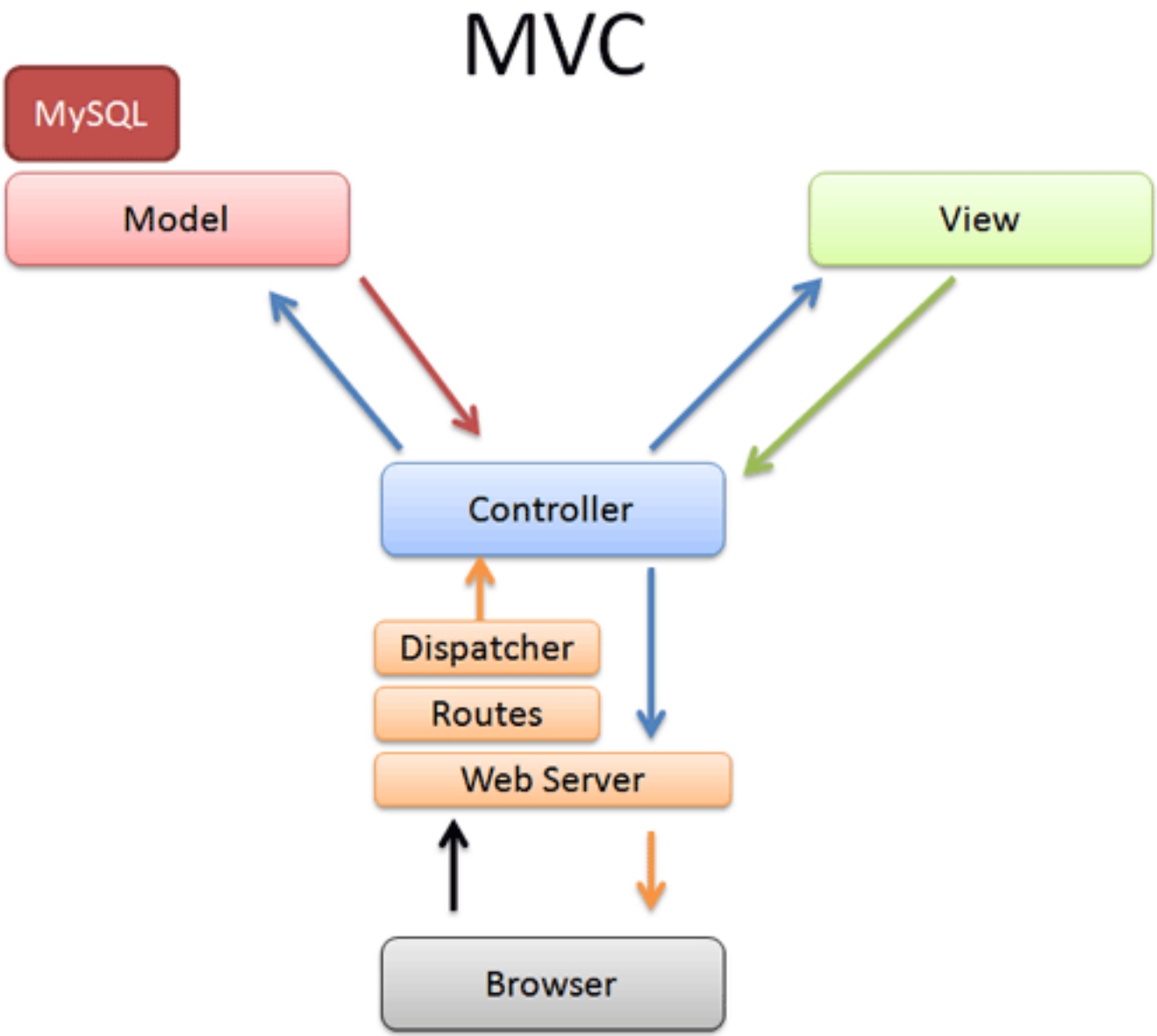
Aspects abordés :

- Un exemple d'implémentation possible de MVC
- Simple (à but pédagogique)
- Orienté Objet

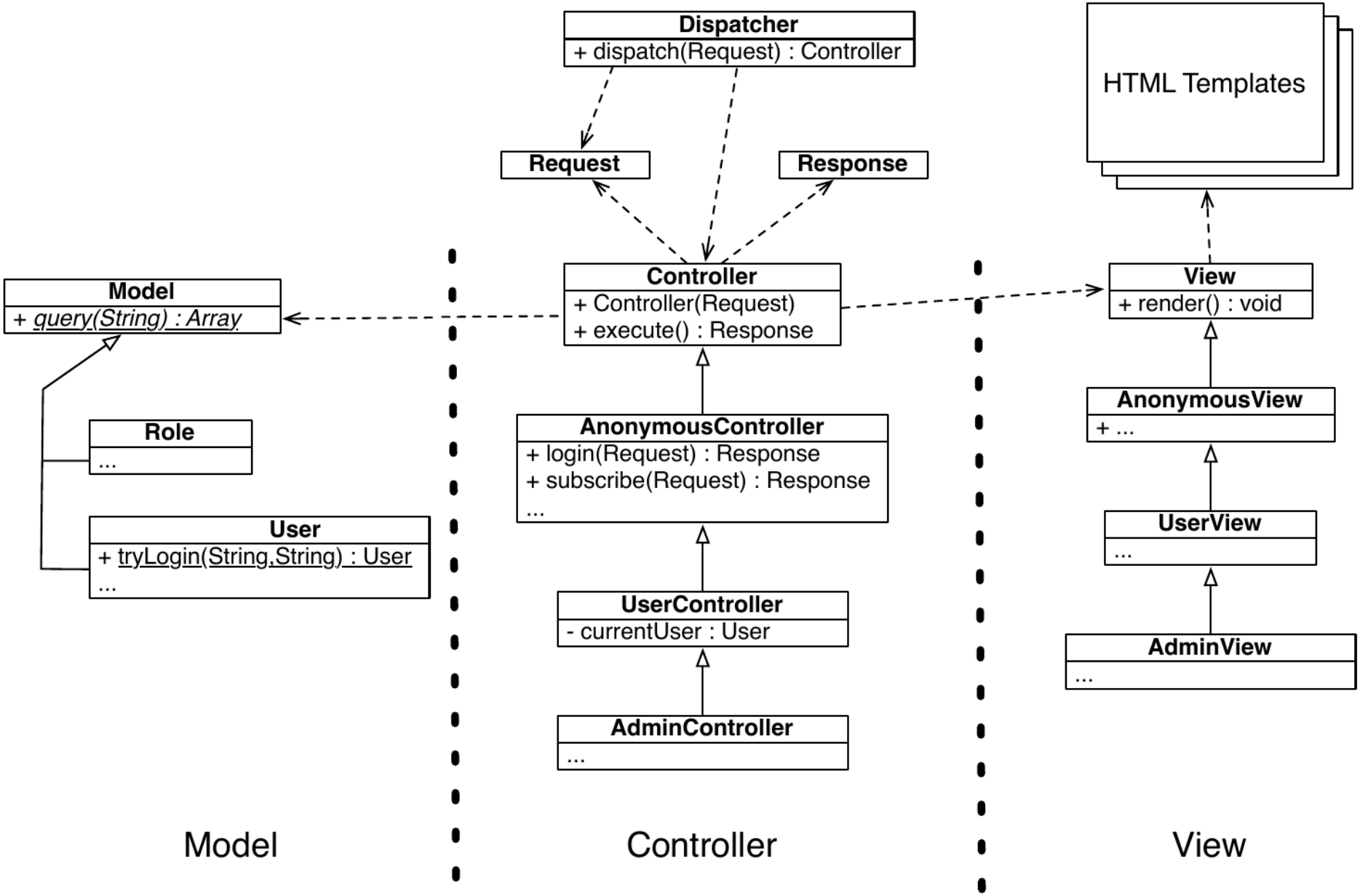
Aspects non abordés :

- Performance
- Sécurité

Plan


MVC
MySQL
Model
View
Controller
Dispatcher
Routes
Web Server
Browser

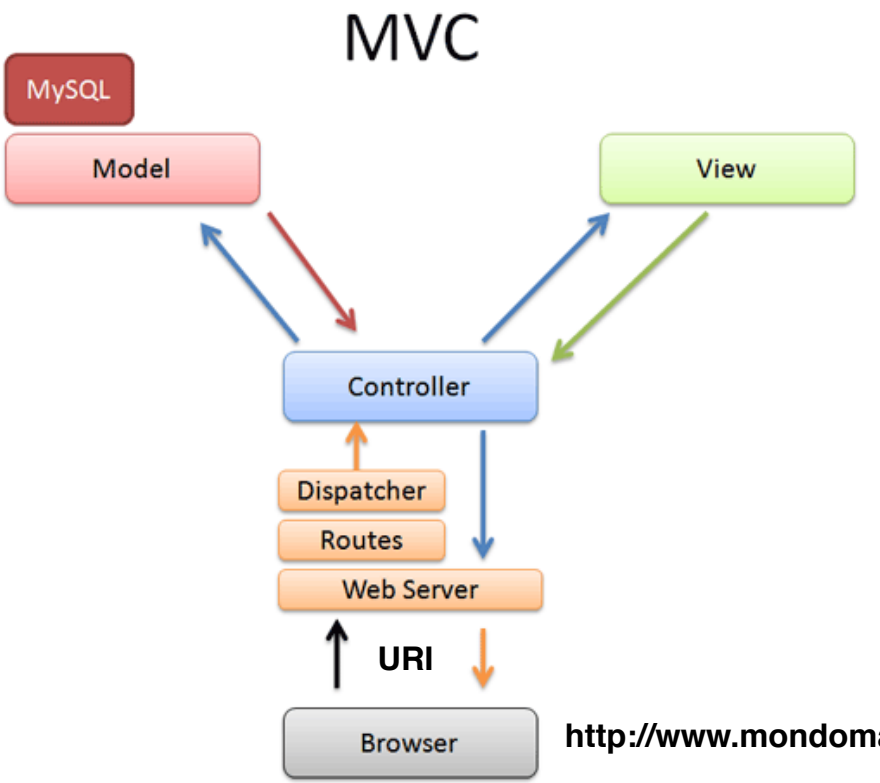
MVC = 3 couches distinctes



Plan

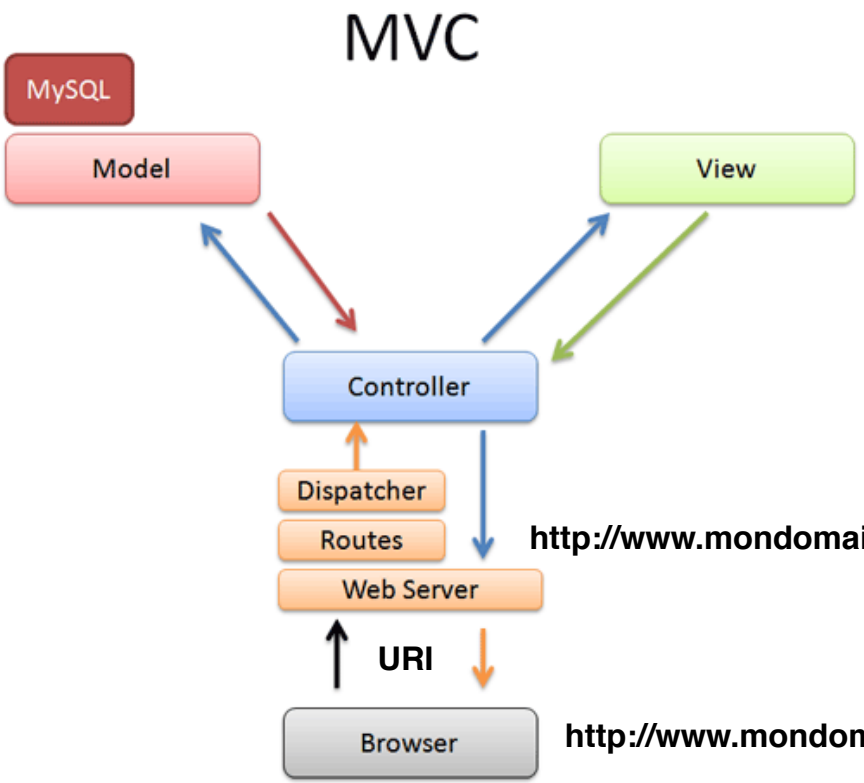
Plan

Le serveur Web : ré-écriture d'URI (1)



http://www.mondomaine.fr/user/view/luc/

Le serveur Web : ré-écriture d'URI (1)



http://www.mondomaine.fr/index.php?controller=user&action=view&param1=luc

mod_rewrite

http://www.mondomaine.fr/user/view/luc/

Fichier .htaccess à la racine du site

```
RewriteEngine on
RewriteCond $1 !^(index\.pl|index\.cgi|index\.html|index\.php|robots\.txt)
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule ^(.*)$ http://eden.imt-lille-douai.fr/~luke/index.php?$1 [P]
```

Découpage d'URI en PHP

Exemple de fichier `index.php` à la racine du site

```php
<?php
  $params = $_SERVER['QUERY_STRING'];
  $params = explode('/',$params);
  print_r($params);
?>
```

Exemple de requête

`http://eden.imt-lille-douai.fr/~luke/anonymous/login/yoda/jedi`

```
Array ( [0] => anonymous [1] => login [2] => yoda [3] => jedi )
```

Plan

Le fichier `index.php`

Responsabilité

- Point d'entré unique de l'application
- Chef d'orchestre du MVC

Exemple : index.php

```php
error_reporting(E_ALL); // always show all PHP errors

... // Load all application settings, ...

// Reify the current request
$request = Request::getCurrentRequest();

try {
   // Instantiate the adequat controller according to the current request
   $controller = Dispatcher::dispatch($request);

   // Execute the requested action
   $response = $controller->execute();
} catch (Exception $e) {
   $response = new ErrorResponse();
   $response->setErrorMessage($e->getMessage());
}

$response->echo();
```

Plan

L'objet `request`

Pourquoi ?

Une ‘façade‘ (cf. Design Pattern) pour accéder aux données en `GET`, `POST`, ...

Plan

Exemple d'action d'un contrôleur

```php
public function inscription($request) {
   $view = new View($this,'inscription');
   $response = new Response();
   $response->interceptEchos();
   $view->render()
   return $response;
}
```

Capturer les echo

http://php.net/outcontrol

```php
ob_start();
echo "hello";
$output = ob_get_clean();
echo "<p class='cool'>" . $output . "</p>";
```

```
<p class='cool'>hello</p>
```

Plan

Le “Dispatcher”

Responsabilité

Instancier le “bon” contrôleur en fonction des paramètres présents dans la requête.

Exemples

```
GET /users => UserController
POST /users => UserController
GET /user/{id} => UserController

GET /games => GameController
POST /games => GameController
GET /game/{id} => GameController

...
```

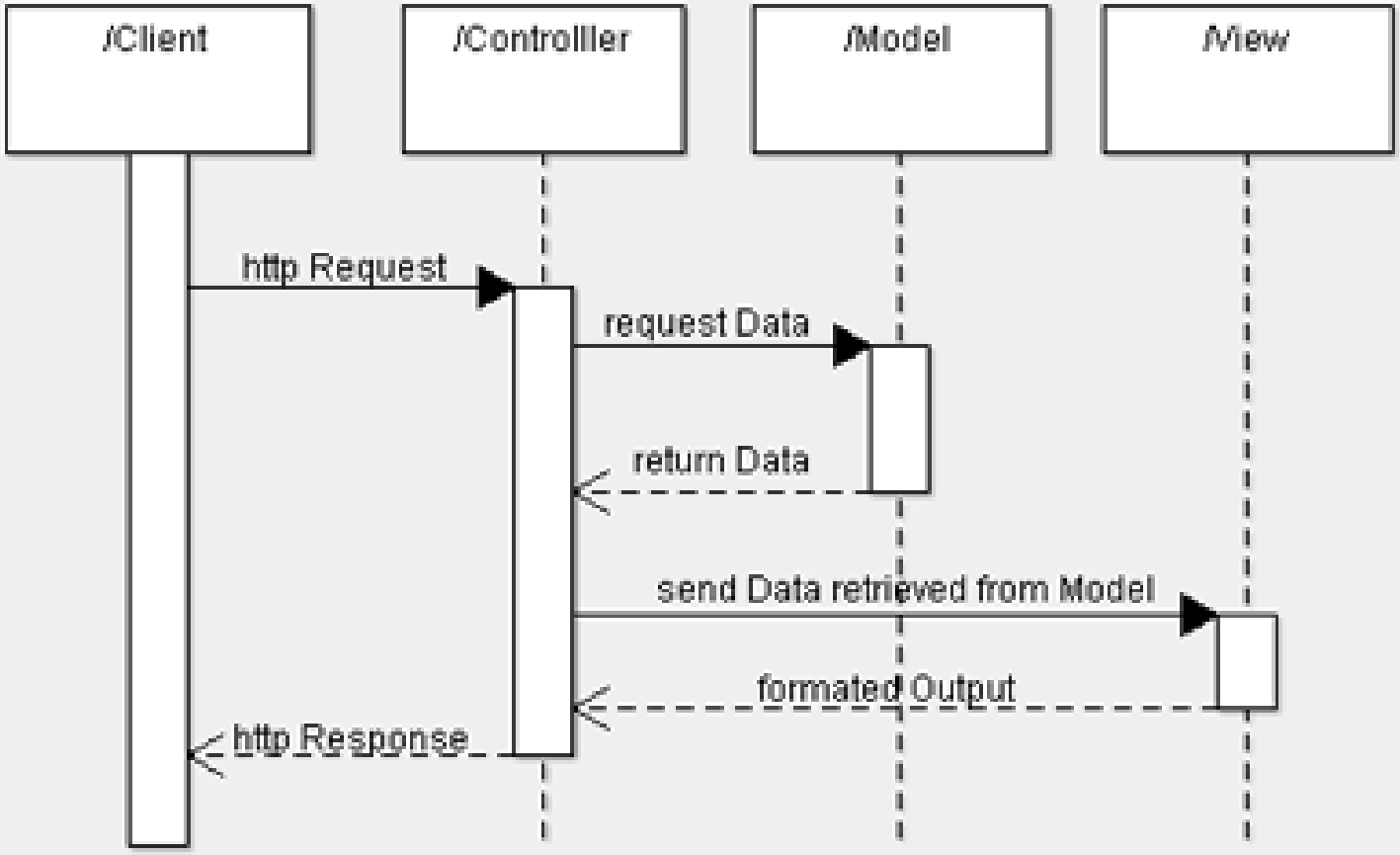
Plan

Contrôleur

Responsabilité d'un contrôleur

- Chef d'orchestre de l'exécution d'une action
- Déclenche les traitements sur les objets métiers
- Instancie l'objet vue adéquat

Contrôleur : Diagrammme de séquence



Exemple : Controller.class.php

```
class Controller extends MyObject {

   public function __construct($request) {
      $this->request = $request;
   }

   // action
   public function execute() {
      $methodName = $this->request->getActionName();
      $this->$methodName($this->request);
   }
}
```

Exemple de Contrôleur

Exemple (incomplet) : UserController.class.php

```
class UserController extends Controller {
   protected $user;

   public function __construct($request) {
      parent::__construct($request);
      session_start();

      $userId = NULL;
      if(($request->has('user')))
         $userId = $request->read('user');

      if(!is_null($userId))
         $this->user = User::getWithId($userId);
      ...
   }

   // action
   public function profile($args) {
      $v = new UserView($this->user);
      $v->renderProfile();
   }

   // action
   public function game($args) {
      $v = new UserView($this->user);
      $v->renderGame();
   }
}
```

Plan

Responsabilité d'un objet modèle

- Représente une entité métier de l'application
- Assure la persistence des données (lien SGBD, ...)
- Permet la recherche d'entité(s) métier

Exemple d'utilisation d'un modèle `User`

Exemples

```php
// Rechercher des modèles
$bestPlayers = User::getBestPlayers();

$user = User::tryLogin('luke','skywalker');

if(isset($user)) {
   // Accéder aux champs d'un modèle
   echo $user->nom();
   echo $user->login();

   // Chargement en cascade de modèles
   echo $user->getRole()->getLabel();
   ...
}
```

Exemples

```
class Model extends MyObject {

   protected static function db(){
      return DatabasePDO::singleton();
   }

   // Exécute la requête $sql et retourne des objets modèles
   protected static function query($sql){
      $st = static::db()->query($sql) or die("sql query error ! request : " . $sql);
      $st->setFetchMode(PDO::FETCH_CLASS | PDO::FETCH_PROPS_LATE, get_called_class());
      return $st;
   }
   ...
}
```

PDO::FETCH_CLASS et get_called_class()

Chaque tuple résultat de la requête sera automatiquement une instance de la classe courante. Exemple :

```
$users = User::query("select * from User");
```

La classe Model (2)

PDO::FETCH_PROPS_LATE

Permet que le constructeur de la classe soit exécuté avant que PDO affecte les valeurs du tuple résultat dans les variables d'instances de l'objet.

Exemples

```
class Model extends MyObject {
   ...
   protected $props;

   public function __construct($props = array()) {
      $this->props = $props;
   }

   public function __get($prop) {
      return $this->props[$prop];
   }

   public function __set($prop, $val) {
      $this->props[$prop] = $val;
   }
   ...
}
```

Requêtes pour un modèle : User.sql.php

Factoriser vos requêtes

Un fichier de requêtes pour chaque modèle

Exemple : User.sql.php

```php
<?php

User::addSqlQuery('USER_LIST',
  'SELECT * FROM USER ORDER BY USER_LOGIN');

User::addSqlQuery('USER_GET_WITH_LOGIN_AND_PASSWORD',
  'SELECT * FROM USER WHERE USER_LOGIN=:login');

User::addSqlQuery('USER_CREATE',
  'INSERT INTO USER (USER_ID, USER_LOGIN, USER_EMAIL, USER_ROLE,
USER_PWD, USER_NAME, USER_SURNAME) VALUES
(NULL, :login, :email, :role, :pwd, :name, :surname)');

User::addSqlQuery('USER_CONNECT',
  'SELECT * FROM USER WHERE USER_LOGIN=:login and USER_PWD=:password');

?>
```

User.class.php

```php
class User extends Model {

   protected static $table_name = 'USER';

   public static function getList() {
      return parent::exec('USER_LIST'); }

   public static function create($login, $pwd, $mail,$nom,$prenom) {
      $sth = parent::exec('USER_CREATE',
         array( ':login' => $login,
            ':email' => $mail,
            ':role' => 1,
            ':pwd' => $pwd,
            ':name' => $prenom,
            ':surname' => $nom));
      return static::tryLogin($login, $pwd);
   }

  // ...
}
```

Exemple : User.class.php

```php
<?php
class User extends Model {

   // ...

   public function id() { return $this->props[self::$table_name.'_ID']; }
   public function roleId() { return $this->props[self::$table_name.'_ROLE']; }
   public function role() { return Role::getWithId($this->roleId()); }
   public function isAdmin() { return ($this->role()->isAdmin()) || ($this->role()->isSuperAdm
   public function isSuperAdmin() { return $this->role()->isSuperAdmin();

}
?>
```

Plan

Responsabilité d'une vue

Mets les données issues de la couche modèle en forme dans la réponse HTTP. Exemples :

- Affiche les données en HTML / CSS / JS
- Affiche les données en format JSON
- ...

Plan

Exemple : View.class.php

```php
class View extends MyObject {
   protected $args;
   protected $templateNames;

   public function __construct($controller, $templateName,$args = array()) {
      parent::__construct();
      $this->templateNames = array();
      $this->templateNames['head'] = 'head';
      $this->templateNames['top'] = 'top';
      $this->templateNames['menu'] = 'menu';
      $this->templateNames['foot'] = 'foot';
      $this->templateNames['content'] = $templateName;
      $this->args = $args;
      $this->args['controller'] = $controller;
   }

   public function setArg($key, $val) {
      $this->args[$key] = $val;
   }

   public function render() {
      $this->loadTemplate($this->templateNames['head'], $this->args);
      $this->loadTemplate($this->templateNames['top'], $this->args);
      $this->loadTemplate($this->templateNames['menu'], $this->args);
      $this->loadTemplate($this->templateNames['content'], $this->args);
      $this->loadTemplate($this->templateNames['foot'], $this->args);
   }
// .... see next slide
```

Exemple : View.class.php

```
   public function loadTemplate($name,$args=NULL) {
      $templateFileName = __SITE_PATH . '/templates/'. $name . 'Template.php';

      if(is_readable($templateFileName)) {
         if(isset($args))
            foreach($args as $key => $value)
               $$key = $value;
         require_once($templateFileName);
      }
      else
         throw new Exception('undefined template "' . $name .'"');
   }
}
```

Exemple : View.class.php

```php
<?php

class UserView extends View {

   public function __construct($controller,$templateName, $args) {
      parent::__construct($controller,$templateName,$args);

      $this->templateNames['menu'] = 'userMenu';
      $this->templateNames['top'] = 'userTop';

      if(!$this->args['user'])
         throw new Exception('a user must be defined');
      }
}

?>
```

Templates : headTemplate.php

Exemple

```
<!doctype html>
<html>
   <head>
      <meta charset="utf-8">
      <title>ChessDie</title>
      <link rel="stylesheet" href="styles/marron.css" type="text/css" media="screen" title="ma
      <!-- <link rel="alternate stylesheet" href="styles/noir.css" type="text/css" media="scre
  </head>
<body>
   <div id="page">
```

Exemple

```
<div id="content">

<h2>Vos Informations</h2>

<table>
   <tr>
      <th>Login : </th>
      <td><?php echo $user->login() ?></td>
   </tr>
   <tr>
      <th>Nom :</th>
      <td><?php echo $user->nom() ?></td>
   </tr>
   <tr>
      <th>Prenom :</th>
      <td><?php echo $user->prenom() ?></td>
   </tr>
   <tr>
      <th>Mail :</th>
      <td><?php echo $user->mail() ?></td>
   </tr>
</table>

</div>
```

Plan

Exemple : View.class.php

```
class Model extends MyObject {
   ...
   // redefine correctly in subclasses
   // may use json_encode
   public asbtract function toJSON();
}
```

Exemple : View.class.php

```
class View extends MyObject {
   protected $models;

   public function __construct($models) {
      parent::__construct();
      $this->models = $models;
   }

   public function render() {
      $jsonResponse = "";
      foreach($this->models as $m) {
         $jsonResponse .= $m->toJSON();
      }
      return $jsonResponse;
   }
}
```

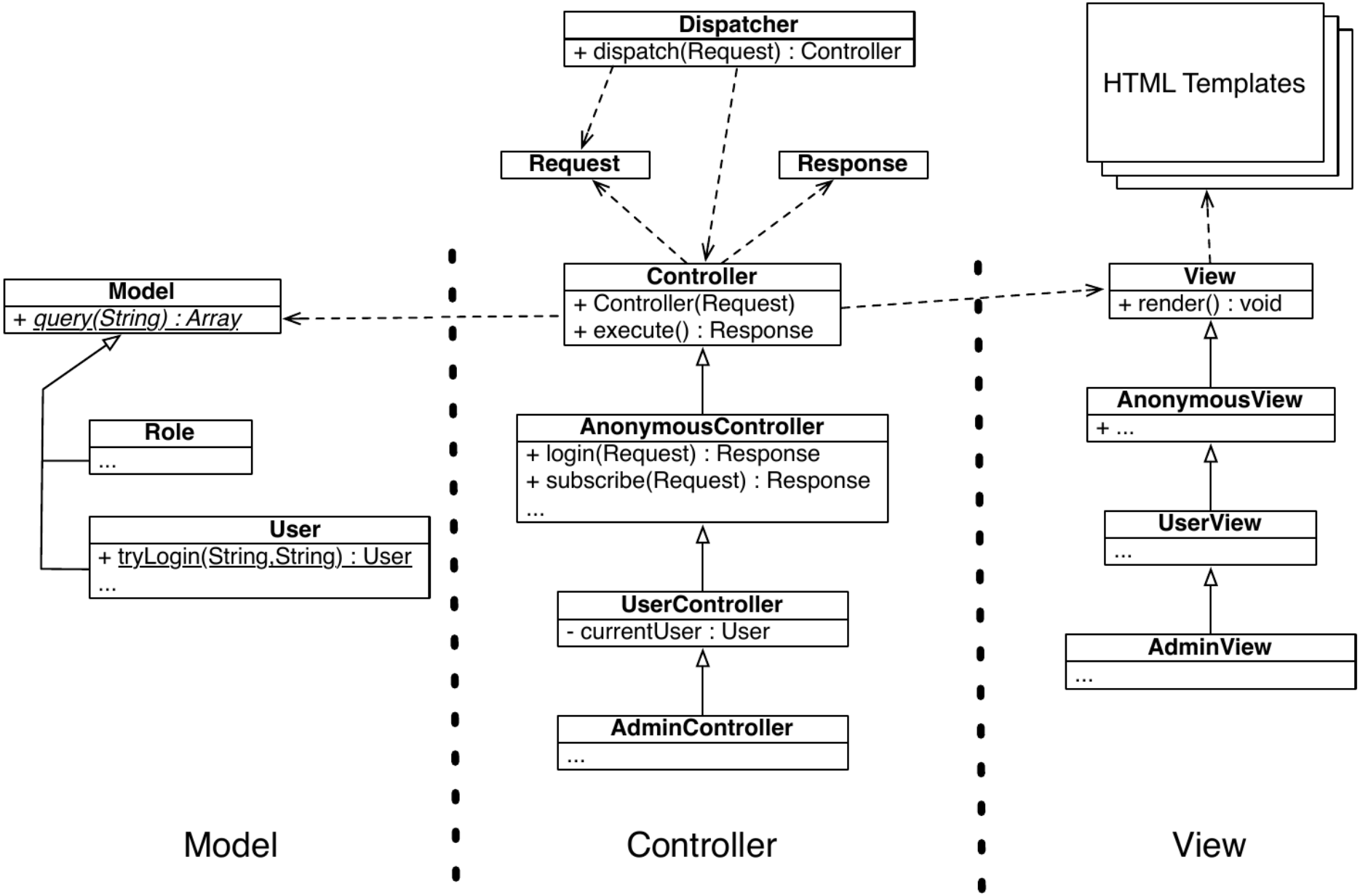
Plan

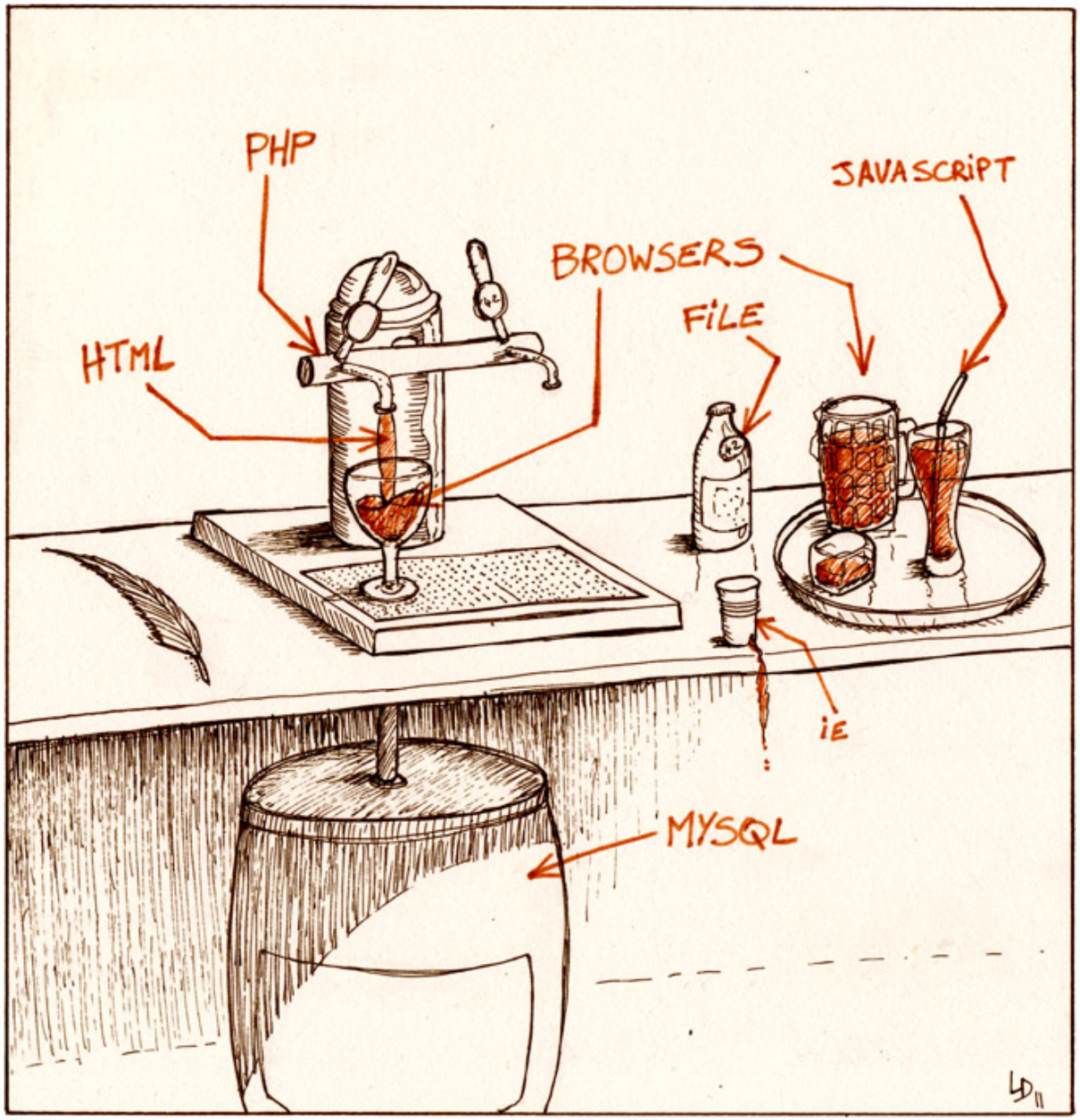
Autres besoins

- Réfier la session
- Réfier les formulaires `<form>`
- Réfier les tableaux `<table>`
- Gestion des erreurs
- ...

Plan

MVC = 3 couches distinctes





Luc Damas