



PHP Web Application Framework

Lucas Greve, Eric Freese,
Max Stein, Brandon Shelton

Overview

- Aims to speed up web application development
- Simplifies & separates: display code, data management, and business logic
- Marketed towards enterprise development
- Runs on most platforms
- Includes framework & developer tools

Design Philosophy

- Clean design and code readability
- Agile development principles such as **DRY**, **KISS**, and **XP**
- Web best practices and design patterns
- Focus on application logic not configuration
 - **Convention over configuration**—the developer needs to configure only the unconventional
- Simple to setup and use, but extensible.

Services Provided

- Templates
- AJAX
- Actions provide business logic
- Data abstraction & ORM
 - Database agnostic
- Filters system
- Extensible plug-in architecture

Services Provided (cont.)

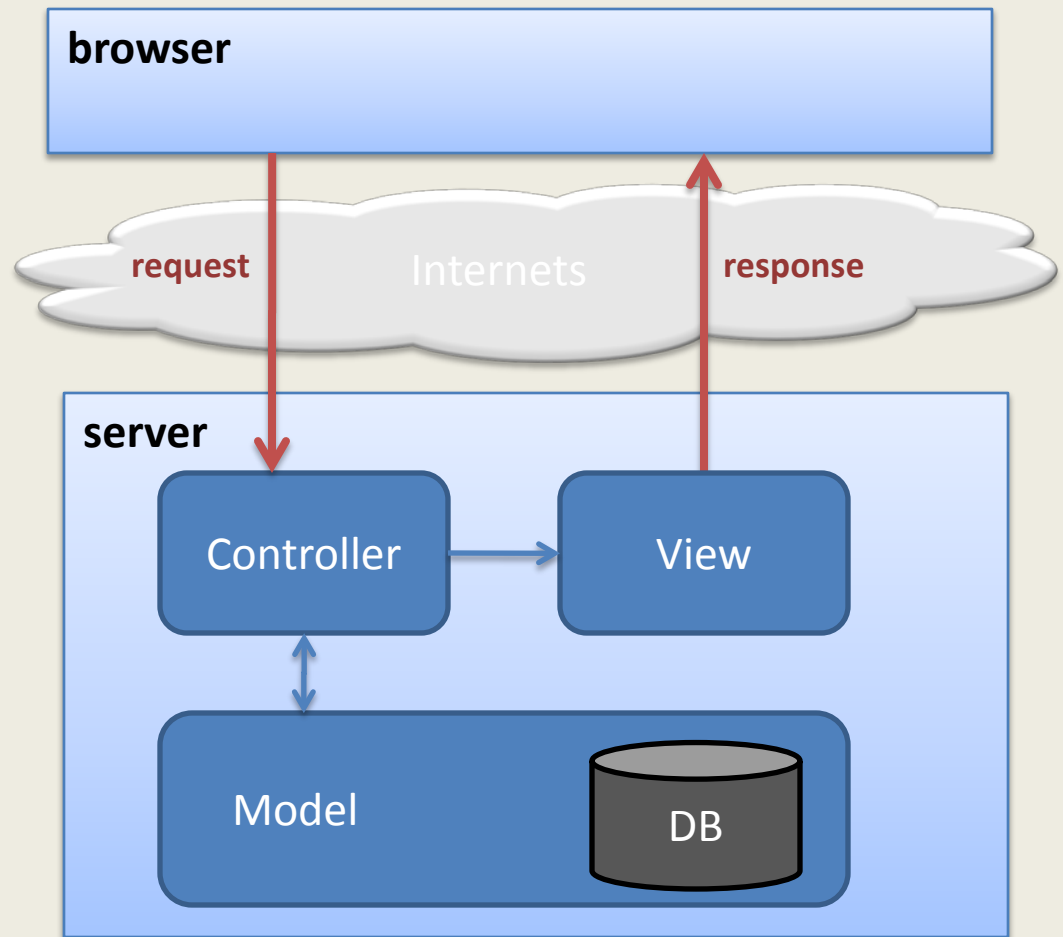
- Authentication system
- Code generation tools
 - Admin generation
- Built in unit & functional testing

MVC Architecture

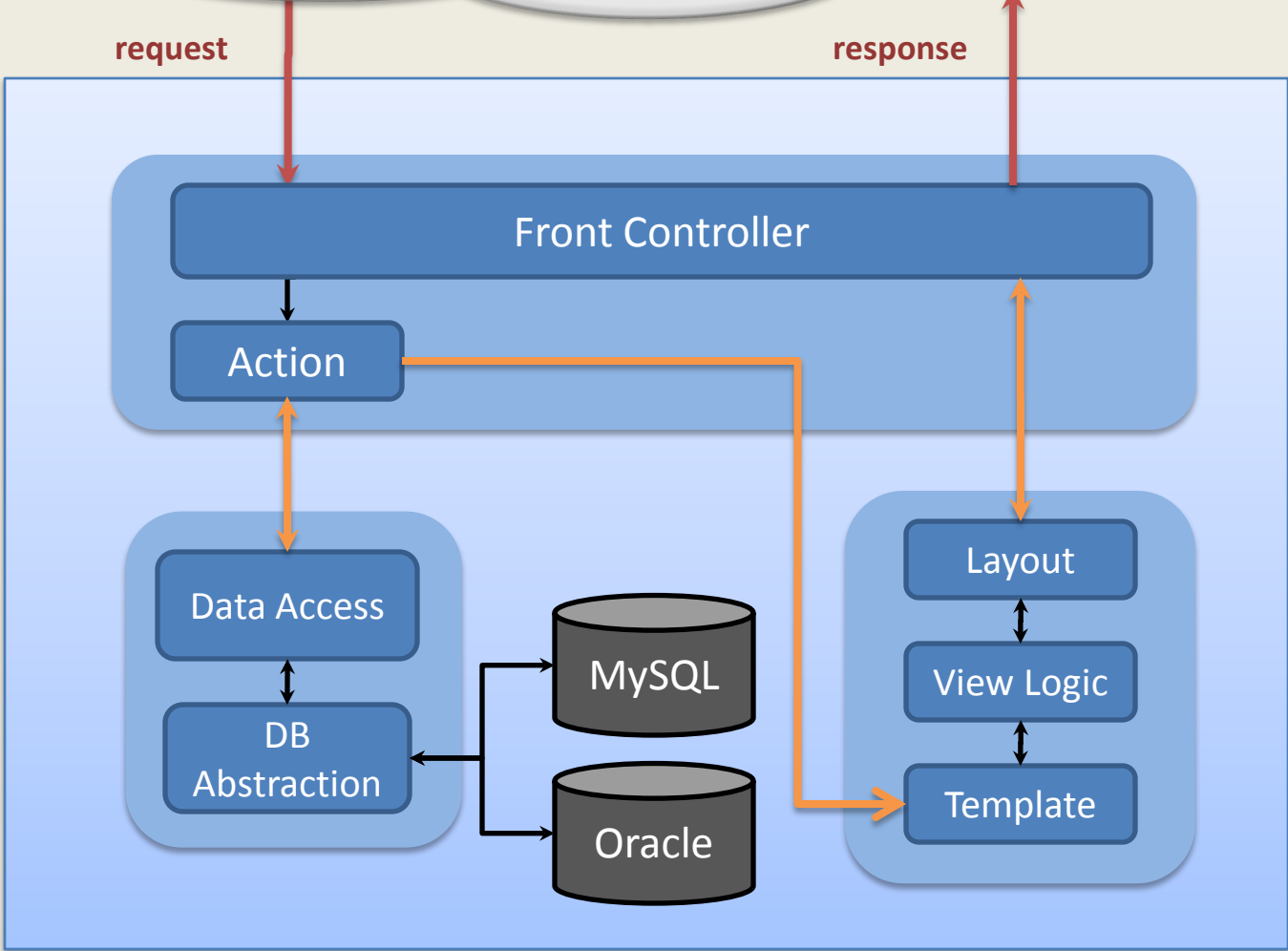
Model domain specific representation of data/business logic

View is the presentation layer, face of application

Controller responds to user actions, changes model or view as necessary, is glue



Internet



Straight PHP

```
<?php
// Connecting, selecting database
$link = mysql_connect('localhost', 'myuser', 'mypassword');
mysql_select_db('blog_db', $link);
// Performing SQL query
$result = mysql_query('SELECT date, title FROM post', $link);
?>
<html>
  <head>
    <title>List of Posts</title>
  </head>
  <body>
    <h1>List of Posts</h1>
    <table>
      <tr><th>Date</th><th>Title</th></tr>
<?php
// Printing results in HTML
while ($row = mysql_fetch_array($result, MYSQL_ASSOC))
{
    echo "\t<tr>\n";
    printf("\t\t<td> %s </td>\n", $row['date']);
    printf("\t\t<td> %s </td>\n", $row['title']);
    echo "\t</tr>\n";
}
?>
      </table>
    </body>
</html>
<?php
// Closing connection
mysql_close($link);
?>
```

Symfony

Controller

```
<?php

// Requiring the model
require_once('model.php');

// Retrieving the list of posts
$post = getAllPosts();

// Requiring the view
require('view.php');
```

Model

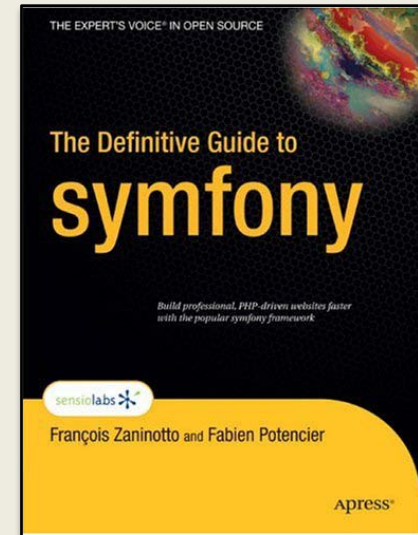
```
function getAllPosts()
{
    ...
    // Connecting to database
    // Performing SQL query
    // Filling up the array
    ...
    $posts = array();
    while ($row = fetch_results($result))
    {
        $posts[] = $row;
    }
    return $posts;
}
```

View

```
<html>
  <head>
    <title>List of Posts</title>
  </head>
  <body>
    <h1>List of Posts</h1>
    <table>
      <tr><th>Date</th><th>Title</th></tr>
      <?php foreach ($posts as $post): ?>
        <tr>
          <td><?php echo $post['date'] ?></td>
          <td><?php echo $post['title'] ?></td>
        </tr>
      <?php endforeach; ?>
    </table>
  </body>
</html>
```

References/Sources

- ***The Definitive Guide to symfony.***
François Zaninotto, Fabien Potencier.
Open Source.
- **Symfony Project Website.**
<http://www.symfony-project.org/>
- **Learn symfony: a Beginners Tutorial.**
<http://www.sitepoint.com/article/symfony-beginners-tutorial/>.
- **Symfonians:** Another symfony
Community. <http://symfonians.net/>



symfony In Action

Demo

