Chapter 4

MVC Architechture

CT313H – WEB TECHNOLOGIES

Objective

Introduce the MVC architecture and

apply this architecture in designing web applications

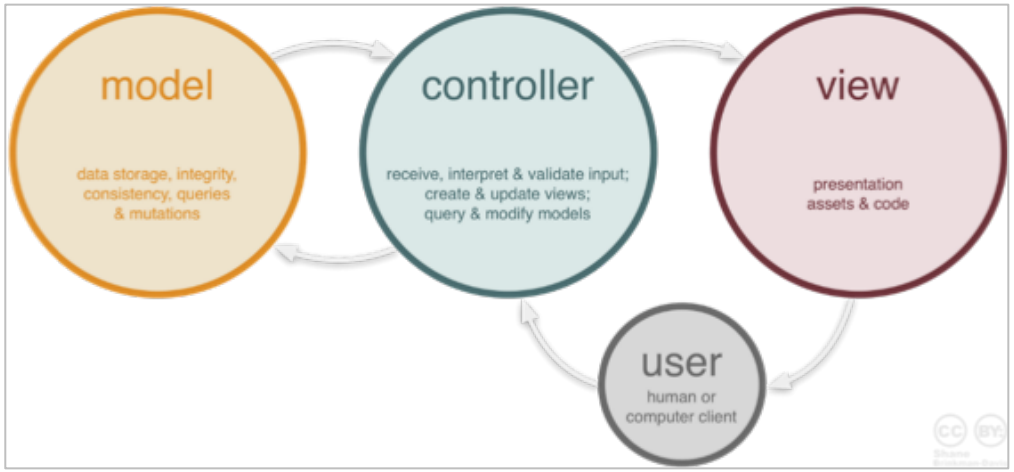
Content

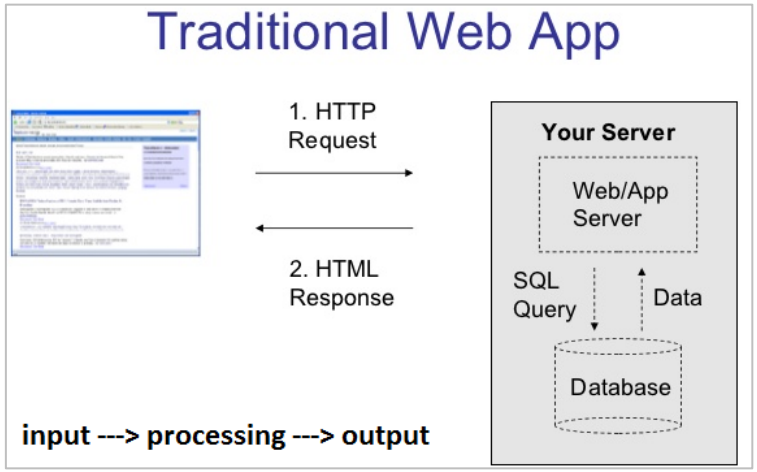
- MVC architecture
  - What is MVC?
  - MVC components
  - Why MVC?
  - MVC processing flow
  - Front Controller
- Appilcation of MVC in web app development
  - Rewrite machemism
  - Project folder structure
  - Common routers

Introduction to MVC Architecture

What is MVC?

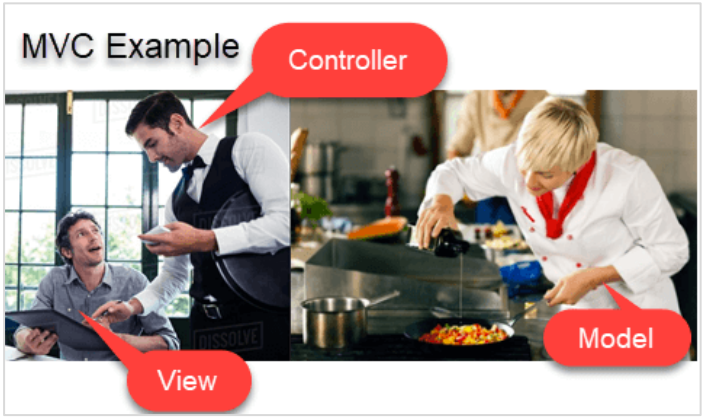
- MVC = Model – View – Controller
- An architecture pattern for application development
- The most popular architecture used in web applications
- An app is separated into 3 main logical components





MVC components

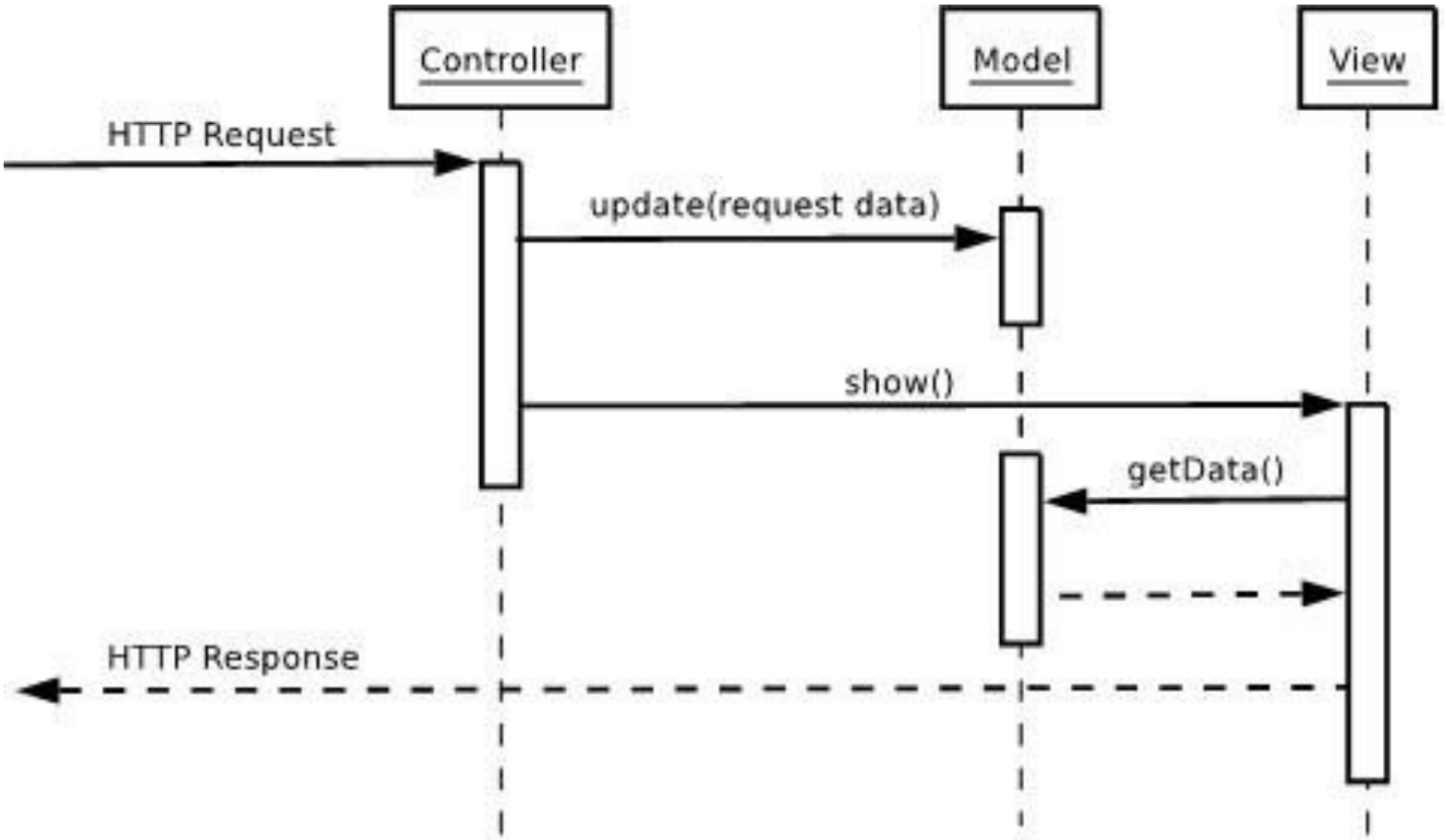
- Model:
  - Stores, manipulates data and process the business logics
  - Fat model: most of data processing tasks is handled by the model
- View:
  - Represents the data and interacts with users
- Controller:
  - An interface between Model and View components
  - Handles the user interaction



Why MVC?

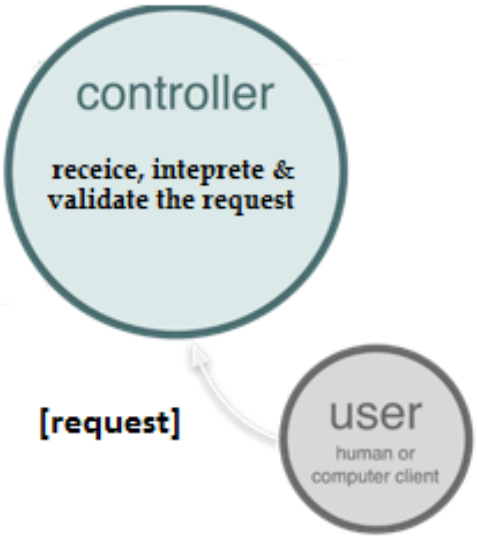
- Based on the idea of separation of concerns
- Each component has a separate responsibility
  - **SRP**: Single Responsibilty Principle
  - **DRY**: Don't Repeat Yourself
- It helps you to avoid complexity (by dividing an app into 3 units)
- Easy code maintenance, easy to extend and grow (reusability)
- Components can be develop concurrently
- Etc.

MVC workflow

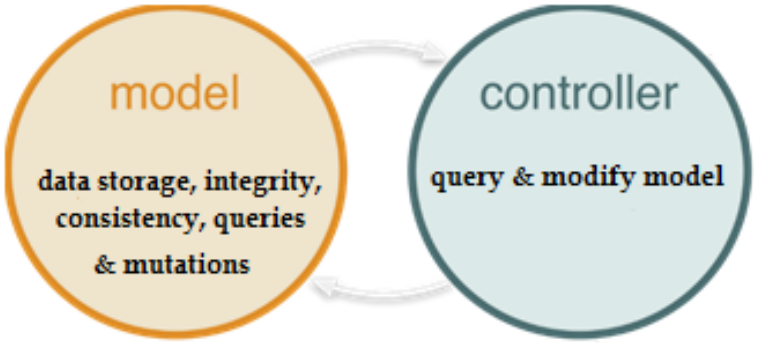


MVC workflow

1) Request is sent to Controller

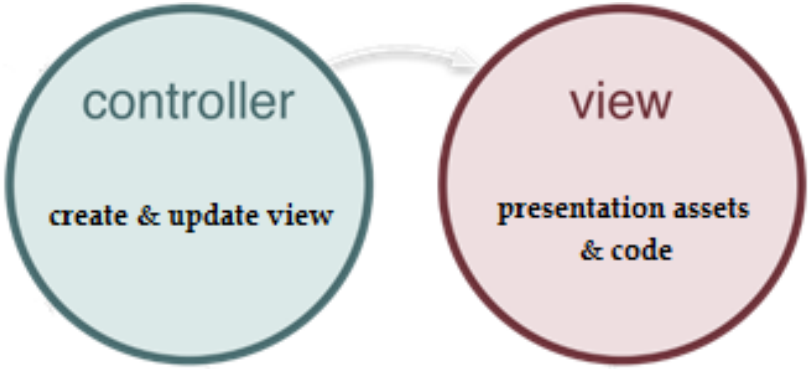


2) Controller calls an appropriated Model to process the request (query or update data,…)

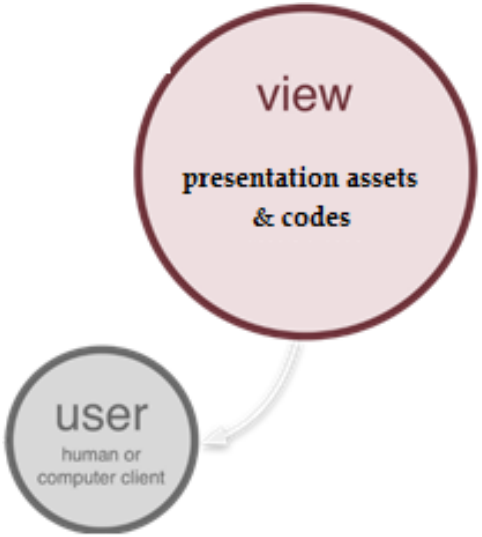


MVC workflow

3) Data produced by Model will be sent to View by Controller



4) View shows data to users



Front Controller

- In a web application, there may be a special controller called Front Controller:
  - an entry point to centralize the processings of the request to the application (usually the index.php)

- Front Controller responsibility:
  - Loads the dependencies
  - Processes the request (throught an appropriated model)
  - Sends back the response to browser

Building MVC Web Applications

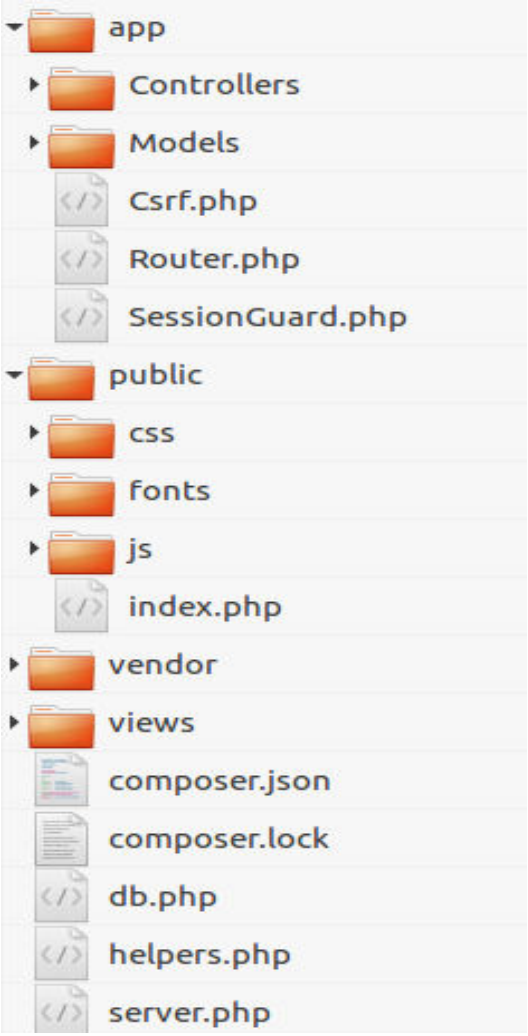
Building MVC Web Application

- Use web server "rewrite" mechanism to transfer all HTTP requests to the Front Controller (index.php)
- Front Controller uses a Router to route the request to a particular processing code
  - Router analyses the request (HTTP method + URL) to identify the processing code
- Building MVC web apps: define a set of rules for Router
  - Rule = HTTP + URL pattern + Handle
  - URLs in the Router rules are logical URLs (i.e. it is not corresponding to a physical resource URL)

Project folder structure

URL Rewrite in Apache

```
<VirtualHost *:8080>
  DocumentRoot "C:/xampp/apps/phonebook/public"
  ServerName localhost

  #Set access permission
  <Directory "C:/xampp/apps/phonebook/public">
    AllowOverride None
    Require all granted

    RewriteEngine On
    RewriteCond %{REQUEST_FILENAME} !-f
    RewriteCond %{REQUEST_FILENAME} !-d
    RewriteRule ^.*$ index.php [L]
  </Directory>
</VirtualHost>
```



Popular routers

- slim/slim
- nikic/fast-route
- klein/klein
- altorouter/altorouter
- aura/router
- noahbuscher/macaw

noahbuscher/macaw

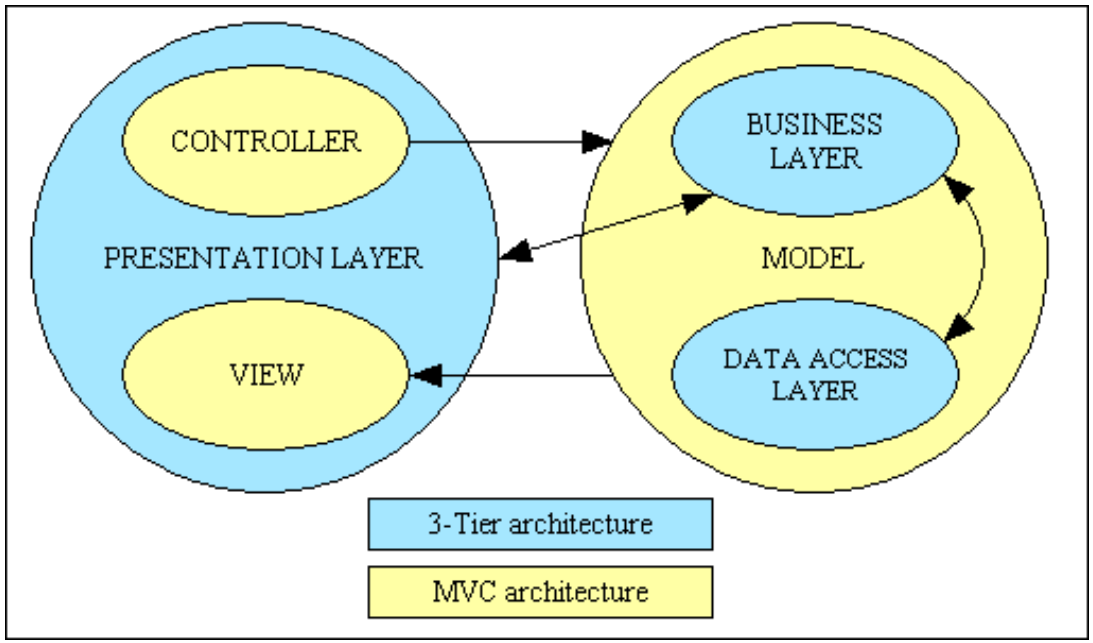
//public/index.php

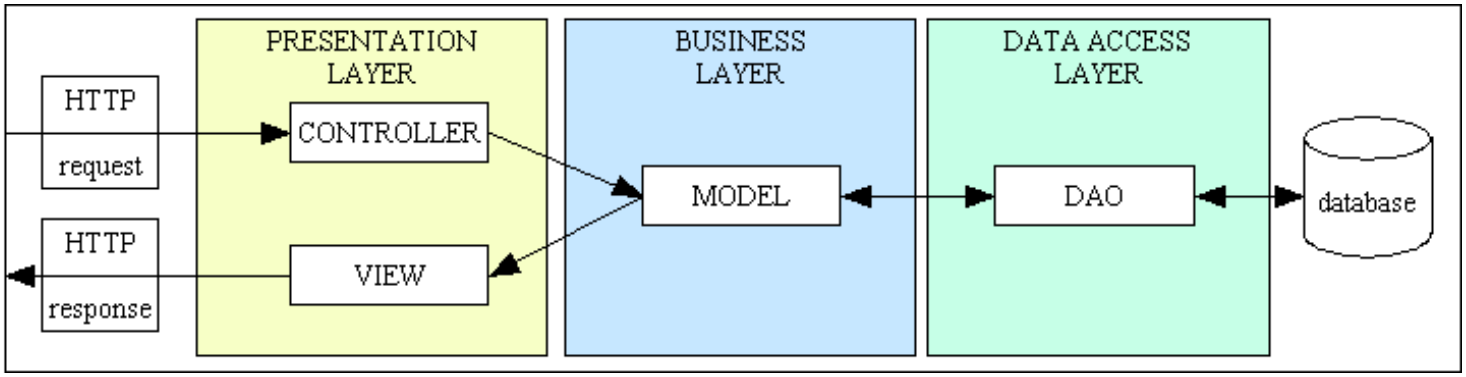
```php
<?php
  require 'vendor/autoload.php';
  use \NoahBuscher\Macaw\Macaw;

  Macaw::get('/',
       'Controllers\Demo@index');
  Macaw::get('page',
       'Controllers\Demo@page');
  Macaw::get('view/(:num)',
       'Controllers\Demo@view');

  Macaw::dispatch();
?>
```

//app/Controllers/Demo.php

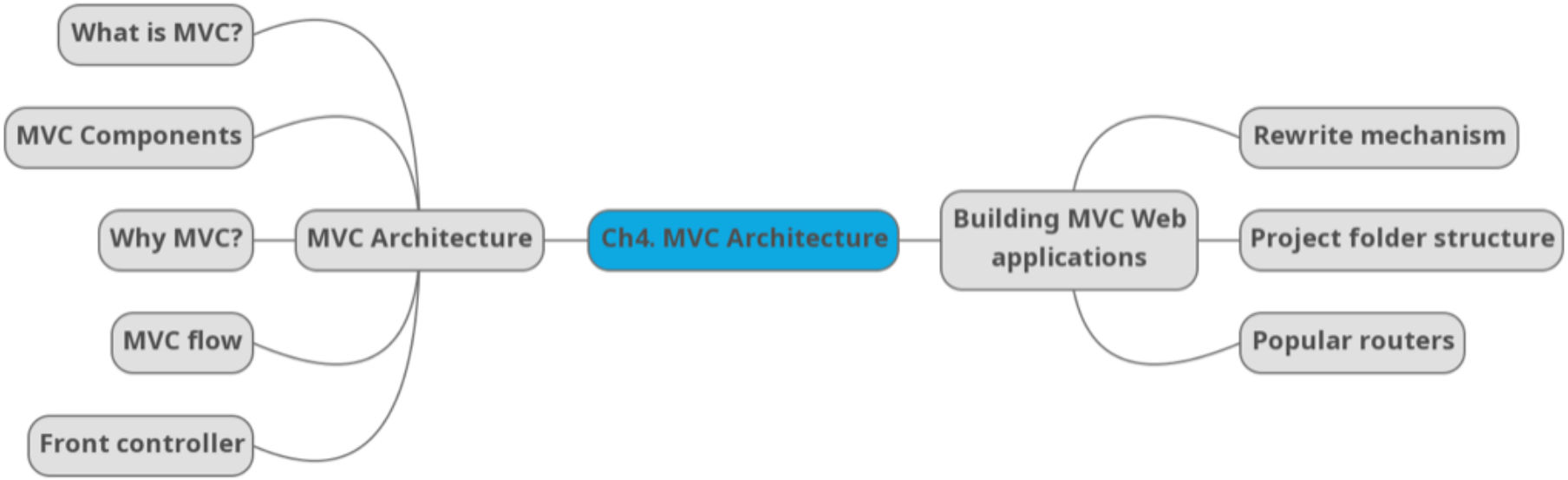
```php
<?php
namespace Controllers;

class Demo {
  public function index() {
    echo 'home';
  }
  public function page() {
    echo 'page';
  }
  public function view($id) {
    echo $id;
  }
}
?>
```

MVC and 3-tier architecture





Recap


What is MVC?
MVC Components
Why MVC?
MVC Architecture
MVC flow
Front controller
Ch4. MVC Architecture
Building MVC Web applications
Rewrite mechanism
Project folder structure
Popular routers

Question?

CT313H – WEB TECHNOLOGIES