The PHP Company

The MVC architecture of ZF2

by Enrico Zimuel (enrico@zend.com)

Senior Software Engineer
Zend Framework Core Team
Zend Technologies Ltd

About me

@ezimuel

enrico@zend.com

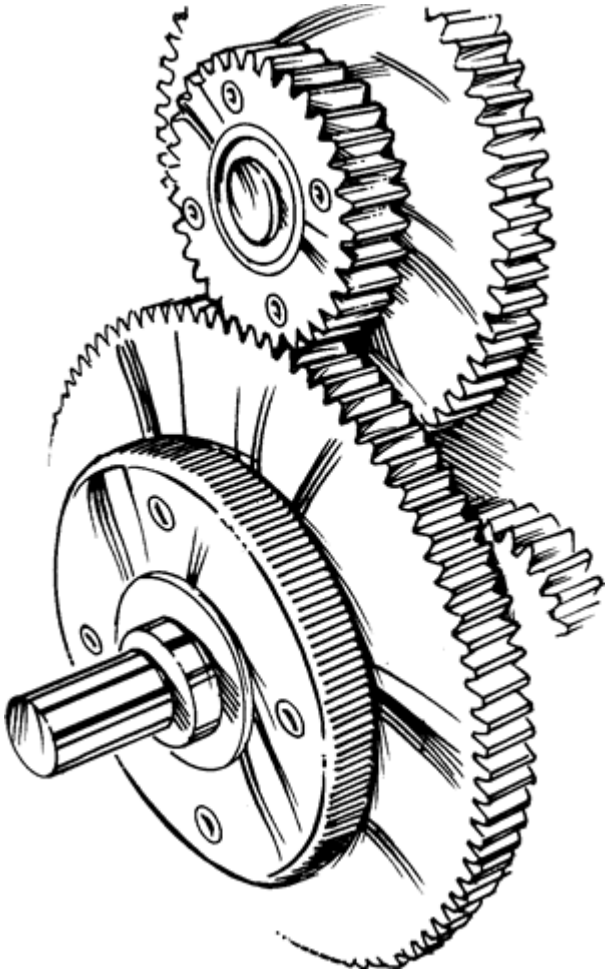
www.zimuel.it

- Enrico Zimuel
- Software Engineer since 1996
  - Assembly x86, C/C++, Java, Perl, PHP
- PHP Engineer at Zend Technologies in the Zend Framework Team
- International speaker on PHP and computer security topics
- Researcher programmer at Informatics Institute of University of Amsterdam
- Co-founder of PUG Torino (Italy)

ZF2 in a slide

- New architecture
  - MVC, Di, Events, Service, Module
- Performance improvement (lazy loading)
- Requirement: **PHP 5.3.3**
- PSR-2 compliant
- Packaging system (pyrus, composer)
- ZF 2.0.3 last stable
  - http://framework.zend.com

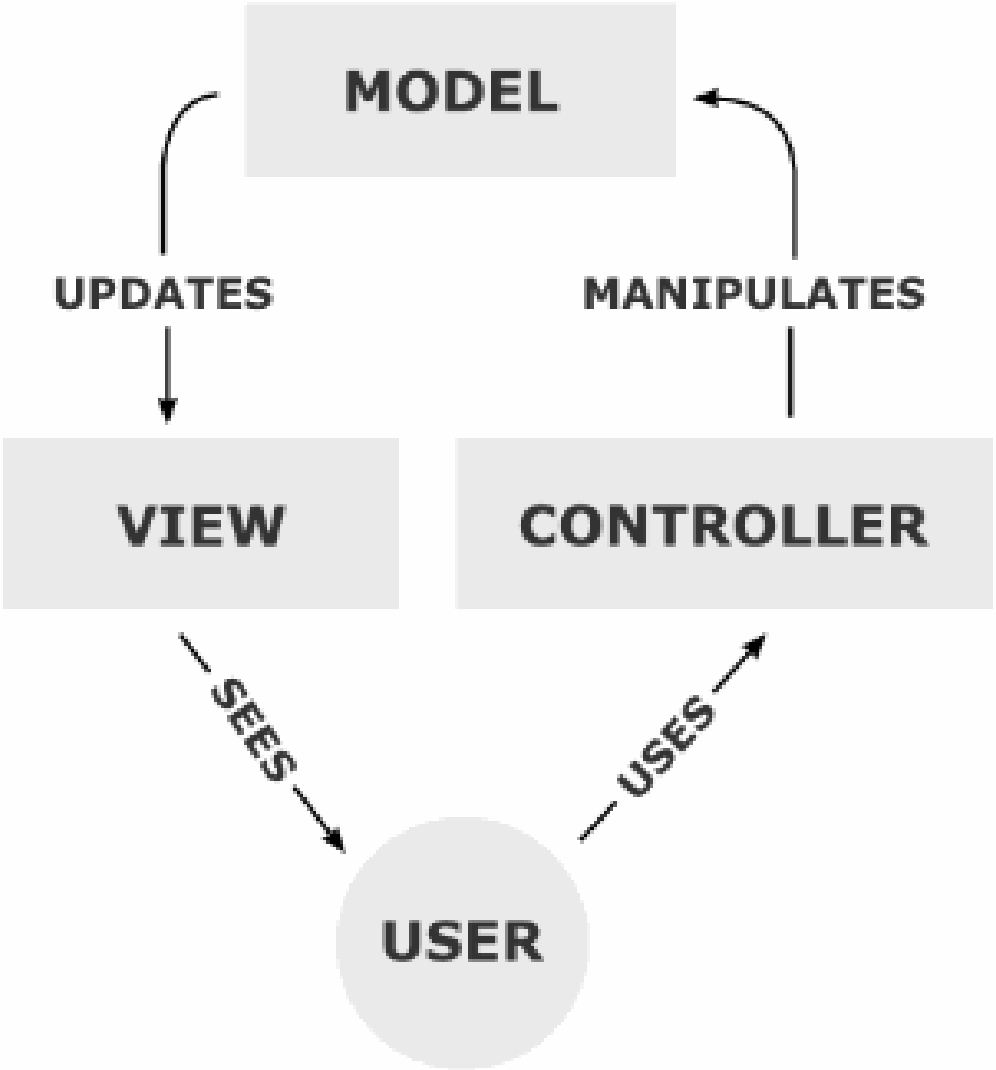
A new core



- The **ZF1** way:
  - Singleton, Registry, and Hard-Coded Dependencies
- The **ZF2** approach:
  - Aspect Oriented Design and Dependency Injection

Model View Controller

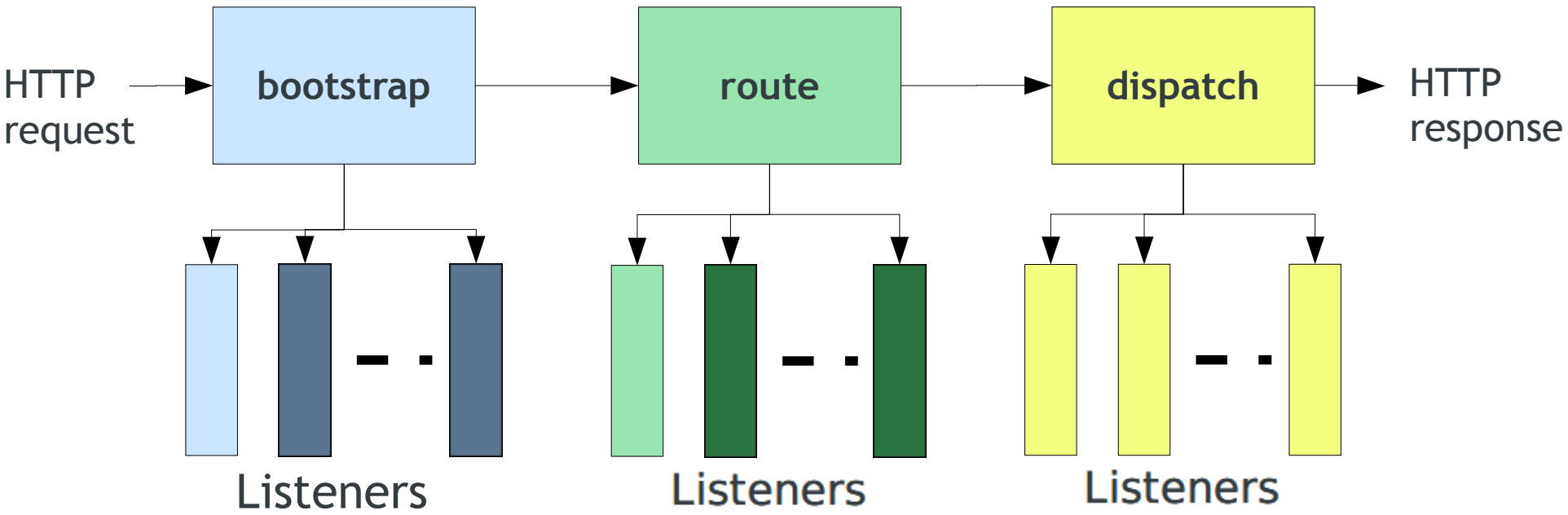
MVC – Model, View, Controller



MVC

The central idea behind MVC is code reusability and separation of concerns

MVC architecture of ZF2



- Everything is an **event**

A common workflow

```
use Zend\ServiceManager\ServiceManager;
use Zend\Mvc\Application;

/* … */

$services = new ServiceManager($servicesConfig);
$app = new Application($appConfig, $services);
$app->bootstrap();
$response = $app->run();
$response->send();
```

Default services

- Application expects a ServiceManager, configured with the following services:
  - EventManager
  - ModuleManager
  - Request
  - Response
  - RouteListener
  - Router
  - DispatchListener
  - ViewManager

ZF2
Skeleton Application

ZF2 skeleton application

- https://github.com/zendframework/ZendSkeletonApplication
- Install using composer:
  - curl -s https://getcomposer.org/installer | php --
  - php composer.phar create-project --repository-url="http://packages.zendframework.com" zendframework/skeleton-application path/to/install

composer.json

```
{
    "name": "zendframework/skeleton-application",
    "description": "Skeleton Application for ZF2",
    "license": "BSD-3-Clause",
    "keywords": [
        "framework",
        "zf2"
    ],
    "homepage": "http://framework.zend.com/",
    "require": {
        "php": ">=5.3.3",
        "zendframework/zendframework": "2.*"
    }
}
```

Directory tree

- config
- data
- module
- public
- vendor

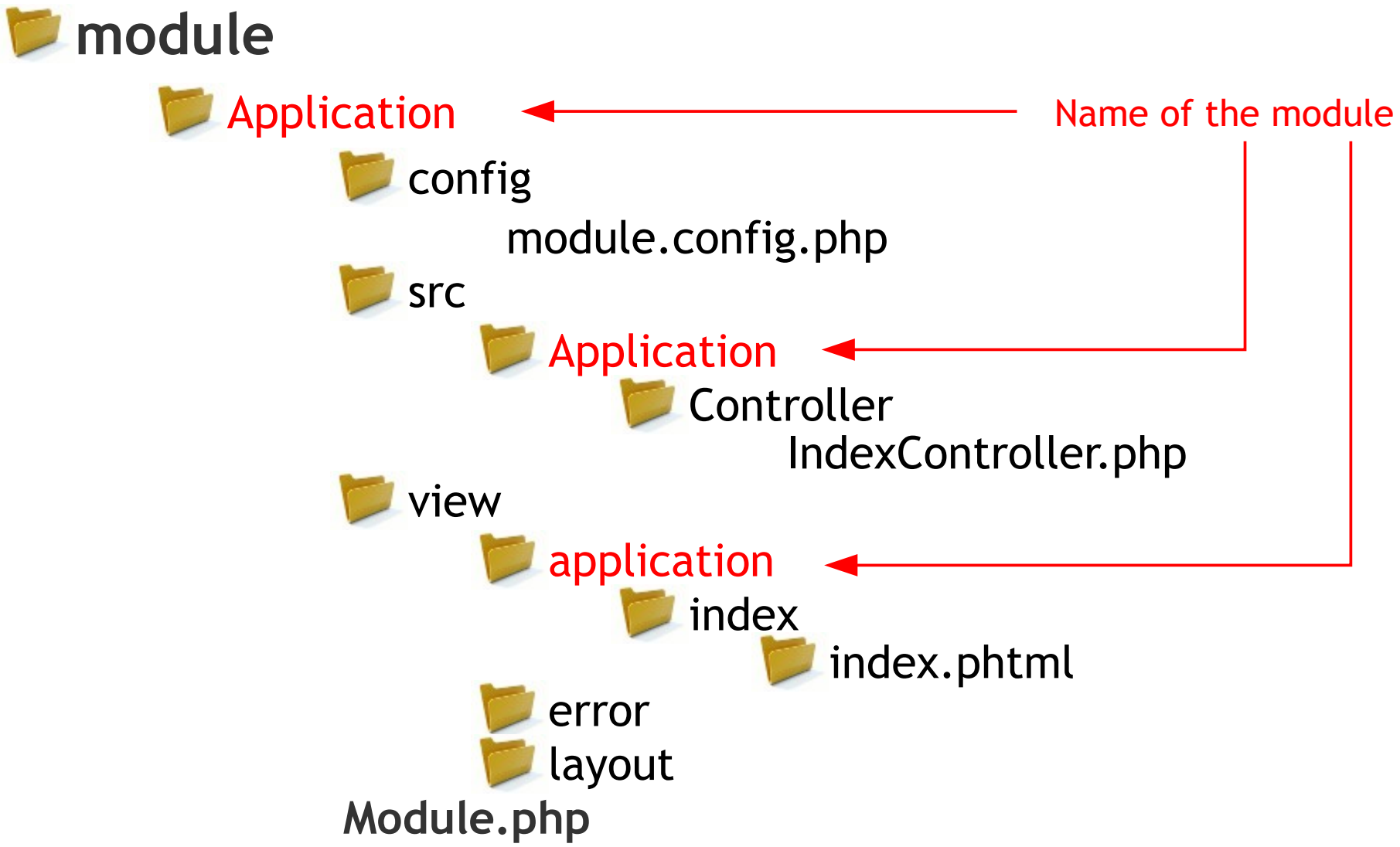
Config folder

- **config**
  - autoload
  - application.config.php
- data
- module
- public
- vendor

Data folder

- config
- **data**
  - cache
- module
- public
- vendor

Module folder



Public folder

- public
  - images
  - js
  - css
  - .htaccess
  - index.php

Vendor folder

- config
- data
- module
- public
- **vendor**
  - zendframework

configuration

/config/application.config.php

```
return array(
    'modules' => array(
        'Application',
    ),
    'module_listener_options' => array(
        'config_glob_paths'    => array(
            'config/autoload/{,*.}{global,local}.php',
        ),
        'module_paths' => array(
            './module',
            './vendor',
        ),
    ),
);
```

public folder

public/.htaccess

```
RewriteEngine On
RewriteCond %{REQUEST_FILENAME} -s [OR]
RewriteCond %{REQUEST_FILENAME} -l [OR]
RewriteCond %{REQUEST_FILENAME} -d
RewriteRule ^.*$ - [NC,L]
RewriteRule ^.*$ index.php [NC,L]
```

Front controller (public/index.php)

```php
<?php
/**
 * This makes our life easier when dealing with paths. Everything is relative
 * to the application root now.
 */
chdir(dirname(__DIR__));

// Setup autoloading
include 'init_autoloader.php';

// Run the application!
Zend\Mvc\Application::init(include 'config/application.config.php')->run()->send();
```

Zend\ServiceManager

- The ServiceManager is a **Service Locator** implementation
- A Service Locator is a well-known object in which you may **register objects** (more in general services) and later **retrieve** them
- Driven by configuration

Types of Services

- Explicit (name => object pairs)
- Invokables (name => class to instantiate)
- Factories (name => callable returning object)
- Aliases (name => some other name)
- Abstract Factories (unknown services)
- Scoped Containers (limit what can be created)
- Shared (or not; you decide)

module

Modules by default

A **module** is all related code and assets that *solve a specific problem*.

Modules **inform the MVC** about services and event listeners

Modules for ZF2

- The basic unit in a ZF2 application is a **Module**
- Modules are **“Plug and play”** technology
- Modules are simple:
  - A namespace
  - Containing a single classfile: **Module.php**

Develop Modules

- Modules contain all logic related to a discrete application problem.
  - Controllers
  - Entities
  - Plugins
  - Etc.
- 99% of the time, you will write *modules*

\module\Application\Module.php

```
namespace Application;

use Zend\Mvc\ModuleRouteListener;
use Zend\Mvc\MvcEvent;

class Module
{
    public function getConfig()
    {
        return include __DIR__ . '/config/module.config.php';
    }
}
```

/module/Application/config/module.config.php (routing part)

```
return array(
    'router' => array(
        'routes' => array(
            'home' => array(
                'type' => 'Zend\Mvc\Router\Http\Literal',
                'options' => array(
                    'route'    => '/',
                    'defaults' => array(
                        'controller' => 'Application\Controller\Index',
                        'action'     => 'index',
                    ),
                ),
            ),
...
```

/module/Application/config/module.config.php (routing part 2)

```
...
'application' => array(
    'type'    => 'Literal',
    'options' => array(
        'route'    => '/application',
        'defaults' => array(
            '__NAMESPACE__' => 'Application\Controller',
            'controller'    => 'Index',
            'action'        => 'index',
        ),
    ),
    'may_terminate' => true,
    'child_routes' => array(
        'default' => array(
            'type'    => 'Segment',
            'options' => array(
                'route'    => '/[:controller[/:action]]',
                'constraints' => array(
                    'controller' => '[a-zA-Z][a-zA-Z0-9_-]*',
                    'action'     => '[a-zA-Z][a-zA-Z0-9_-]*',
                ),
                'defaults' => array(
                ),
...
```

/module/Application/config/module.config.php (controller & translator part)

```
...
'service_manager' => array(
        'factories' => array(
            'translator' => 'Zend\I18n\Translator\TranslatorServiceFactory',
        ),
    ),
    'translator' => array(
        'locale' => 'en_US',
        'translation_file_patterns' => array(
            array(
                'type'     => 'gettext',
                'base_dir' => __DIR__ . '/../language',
                'pattern'  => '%s.mo',
            ),
        ),
    ),
    'controllers' => array(
        'invokables' => array(
            'Application\Controller\Index' => 'Application\Controller\IndexController'
        ),
    ),
...
```

/module/Application/config/module.config.php (view)

```
...
'view_manager' => array(
        'display_not_found_reason' => true,
        'display_exceptions'       => true,
        'doctype'                  => 'HTML5',
        'not_found_template'       => 'error/404',
        'exception_template'       => 'error/index',
        'template_map' => array(
            'layout/layout'           => __DIR__ . '/../view/layout/layout.phtml',
            'application/index/index' => __DIR__ . '/../view/application/index/index.phtml',
            'error/404'               => __DIR__ . '/../view/error/404.phtml',
            'error/index'             => __DIR__ . '/../view/error/index.phtml',
        ),
        'template_path_stack' => array(
            __DIR__ . '/../view',
        ),
    ),
...
```

/module/Application/src/Application/Controller/IndexController.php

```
namespace Application\Controller;

use Zend\Mvc\Controller\AbstractActionController;
use Zend\View\Model\ViewModel;

class IndexController extends AbstractActionController
{
    public function indexAction()
    {
        return new ViewModel();
    }
}
```

Thank you!

- More information
  - http://framework.zend.com
- IRC channels (freenode)
  - #zftalk, #zftalk.dev

zend framework 2

The **most popular framework** for modern, high-performing PHP applications