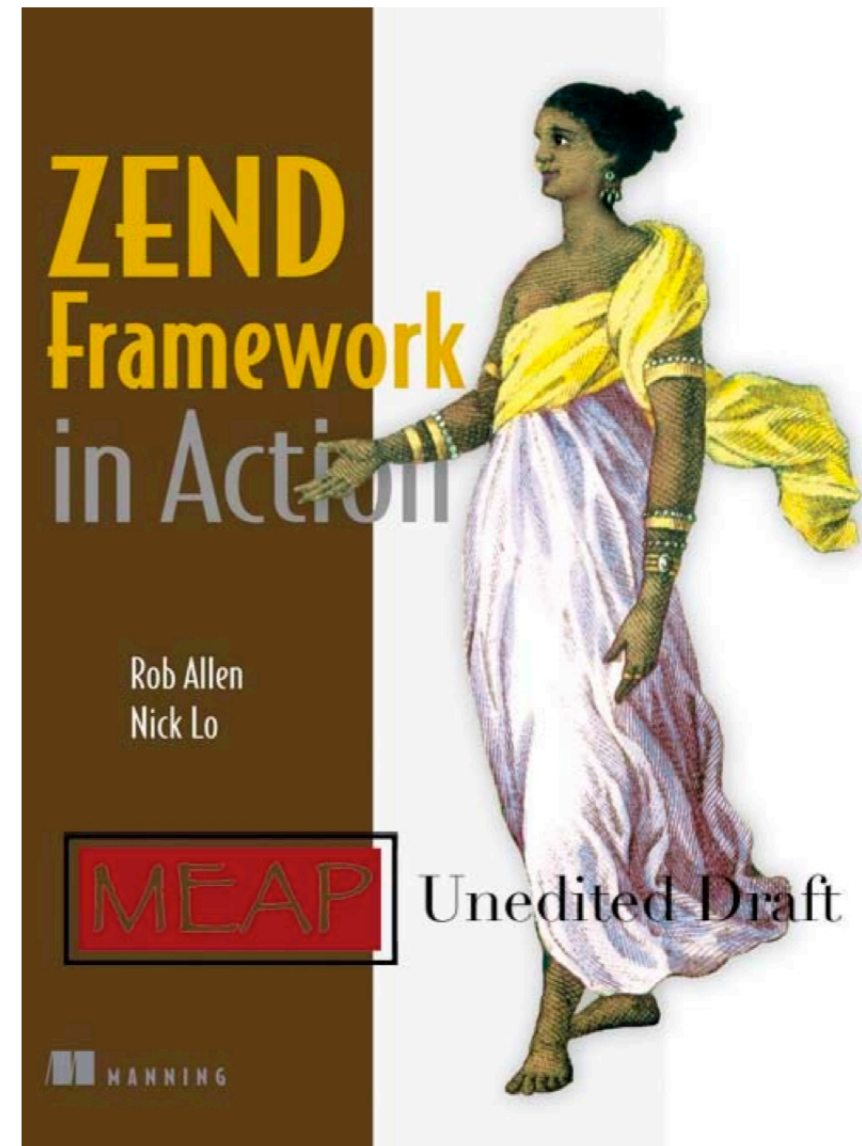


First steps with

# Zend Framework

Rob Allen

# Rob Allen?



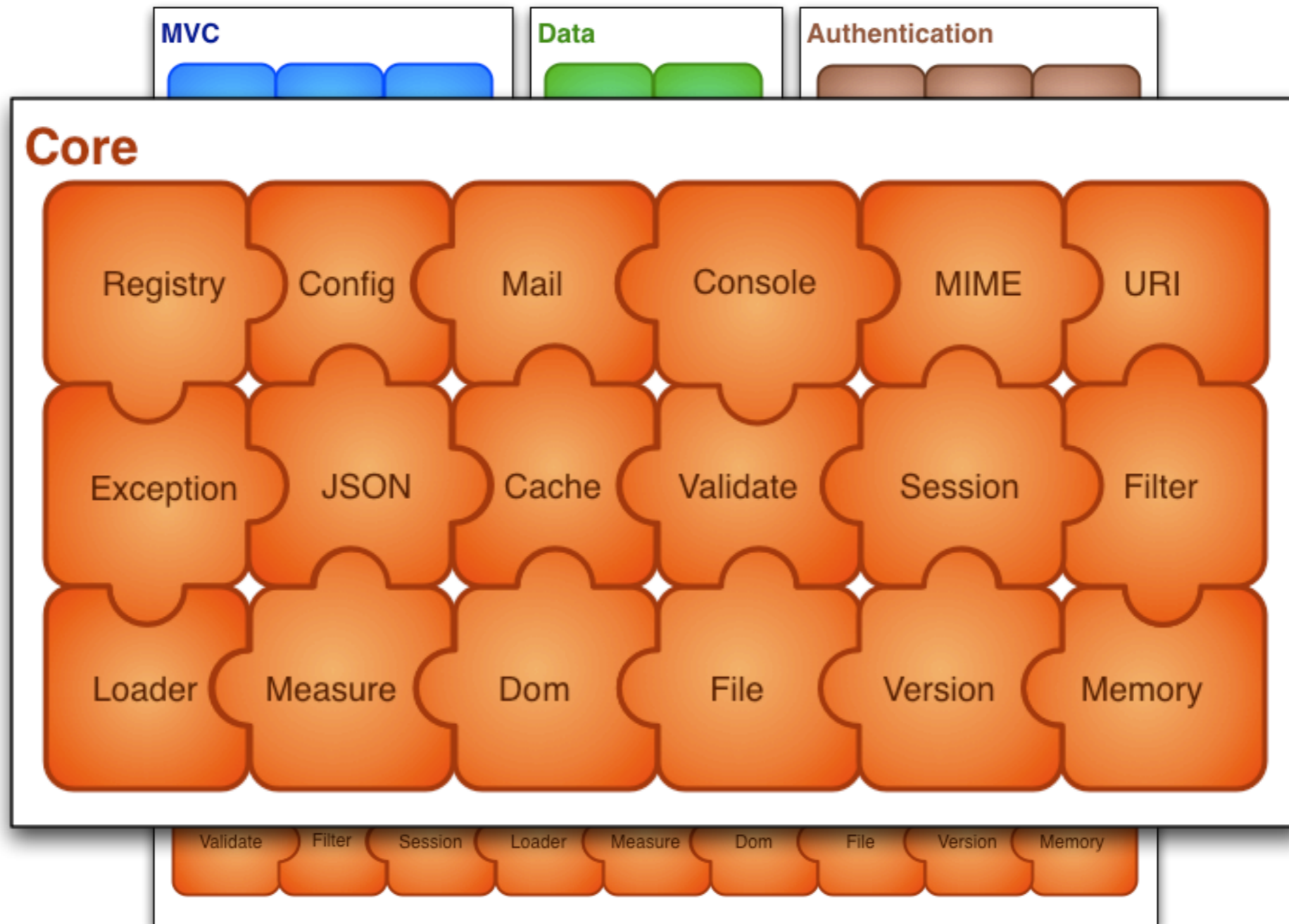
# What is ZF?

- Use-at-will PHP5 Framework
- Open source - BSD license
- Documented
- Quality assured
- Certification
- Actively maintained by Zend

# ZF Philosophy

- Use-at-will
- Simple usage 80% of the time
- Agile practices
- Showcase current trends

# ZF Components



# Getting started

- Use individual components in your current applications
- Start a new application using the ZF MVC components

# Using Zend\_Cache in an existing application

# The current code

```
function getNews(PDO $dbh)
{
    $stmt = $dbh->query('SELECT *
        FROM news ORDER BY date_created DESC');
    $stmt->setFetchMode(PDO::FETCH_ASSOC);
    $list = $stmt->fetchAll();
    return $list;
}

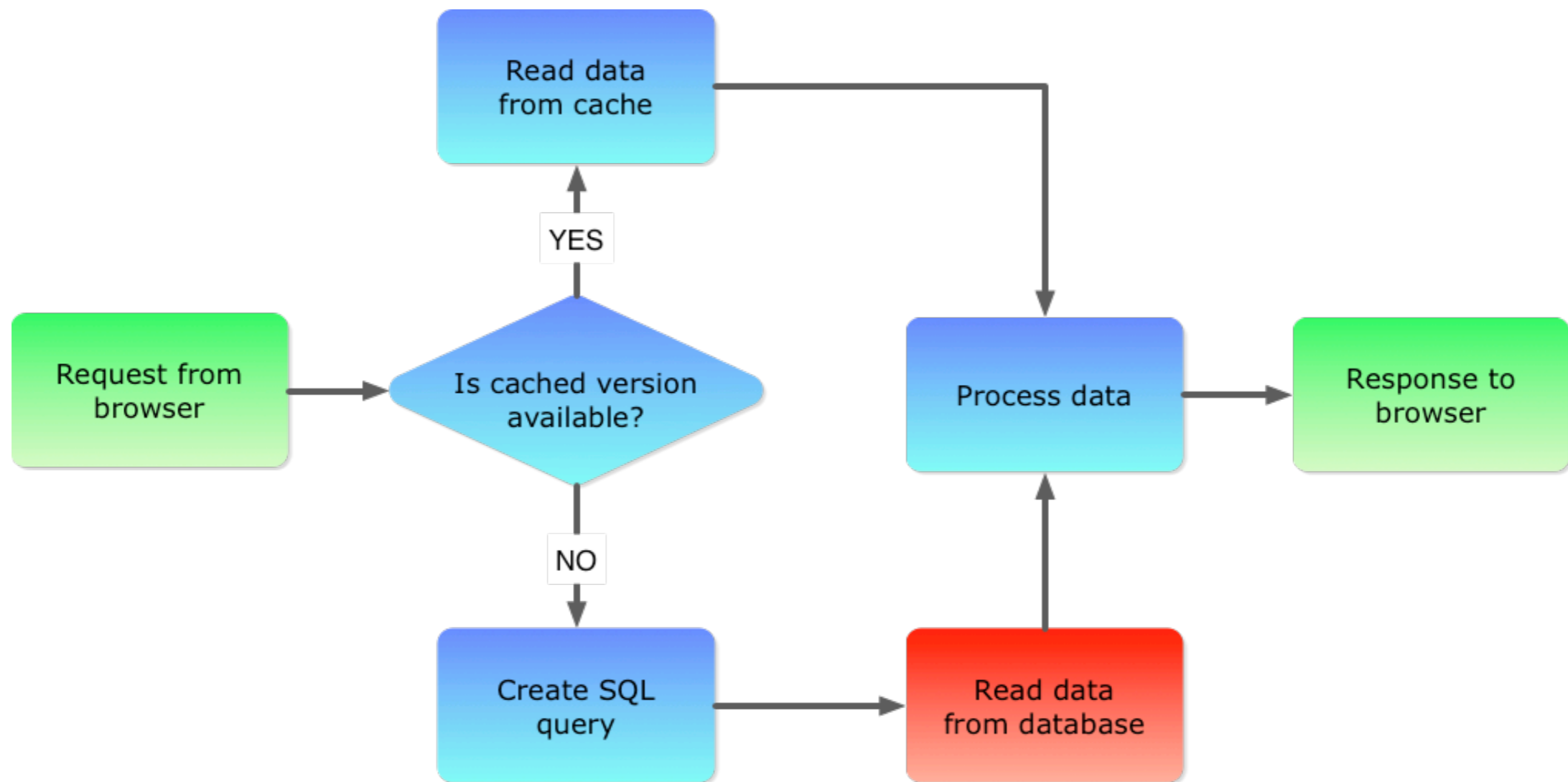
$news = getNews($dbh);
```



# The HTML

```
<h1>News</h1>
<ul>
  <?php foreach($news as $row) : ?>
    <li><?= $row['title'] ?> -
      <?= $row['date_created'] ?></li>
  <?php endforeach; ?>
</ul>
```

# Zend\_Cache diagram



# Add Zend\_Cache

1. Add Zend Framework to lib/Zend folder
2. Create a cache data folder
3. Set up the cache
4. Wrap cache code around database query
5. That's it!

# Zend\_Cache set-up

```
function initCache($cacheDir)
{
    require_once('Zend/Cache.php');
    $feOpts = array(
        'lifetime' => '7200',
        'automatic_serialization' => true);
    $bkOpts = array(
        'cache_dir' => $cacheDir);
    $cache = Zend_Cache::factory('Core', 'File',
        $feOpts, $bkOpts);
    return $cache;
}
```

# Zend\_Cache in use

```
function getNewsFromCache(PDO $dbh,  
    Zend_Cache_Core $cache)  
{  
    $cacheId = 'latestNews';  
    $list = $cache->load($cacheId);  
    if($list === false) {  
        $list = getNews($dbh);  
        $cache->save($list, $cacheId);  
    }  
  
    return $list;  
}
```

Unique



Re-use  
existing  
code



# Zend\_Cache in use

```
$cacheDir = dirname(__FILE__) . '/cache';  
$cache = initCache($cacheDir);
```

```
$news = getNewsFromCache($dbh, $cache);
```

(the HTML does not change)

# Useful components

for integrating into an existing code base:

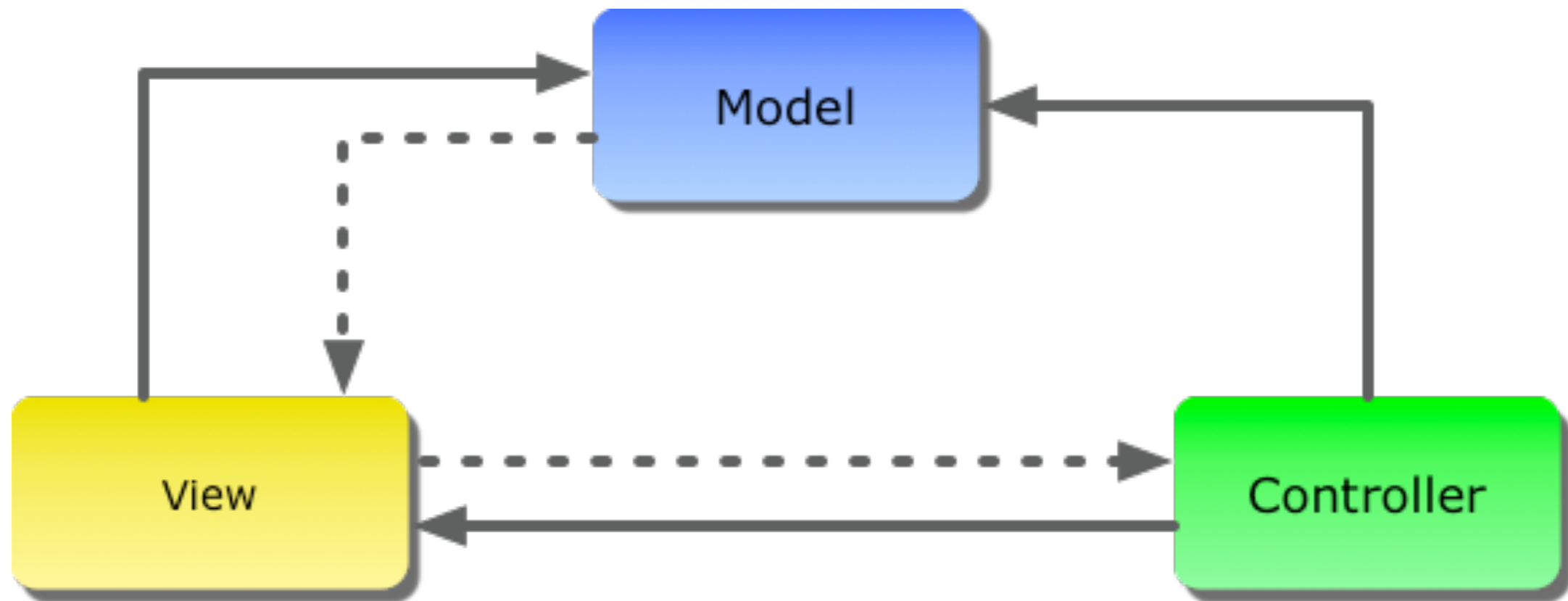
- Cache
- PDF
- Search
- Auth: Captcha, OpenId, LDAP, InfoCard
- Web services, Logging and WildFire
- Mail

A new  
**Zend Framework**  
application

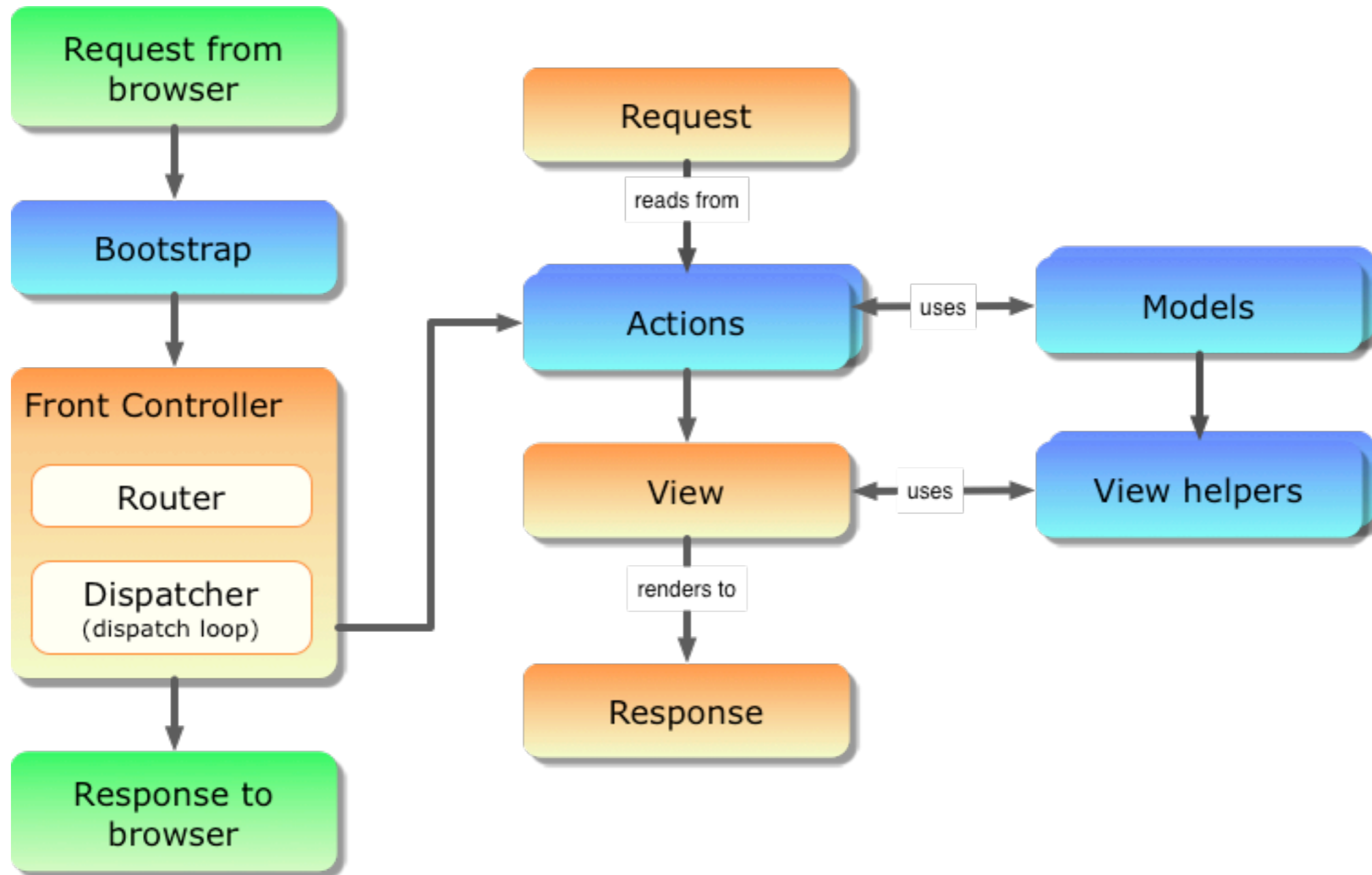


# Model View Controller

(Most Vexing Conundrum)



# ZF's MVC



# Routing: URLs

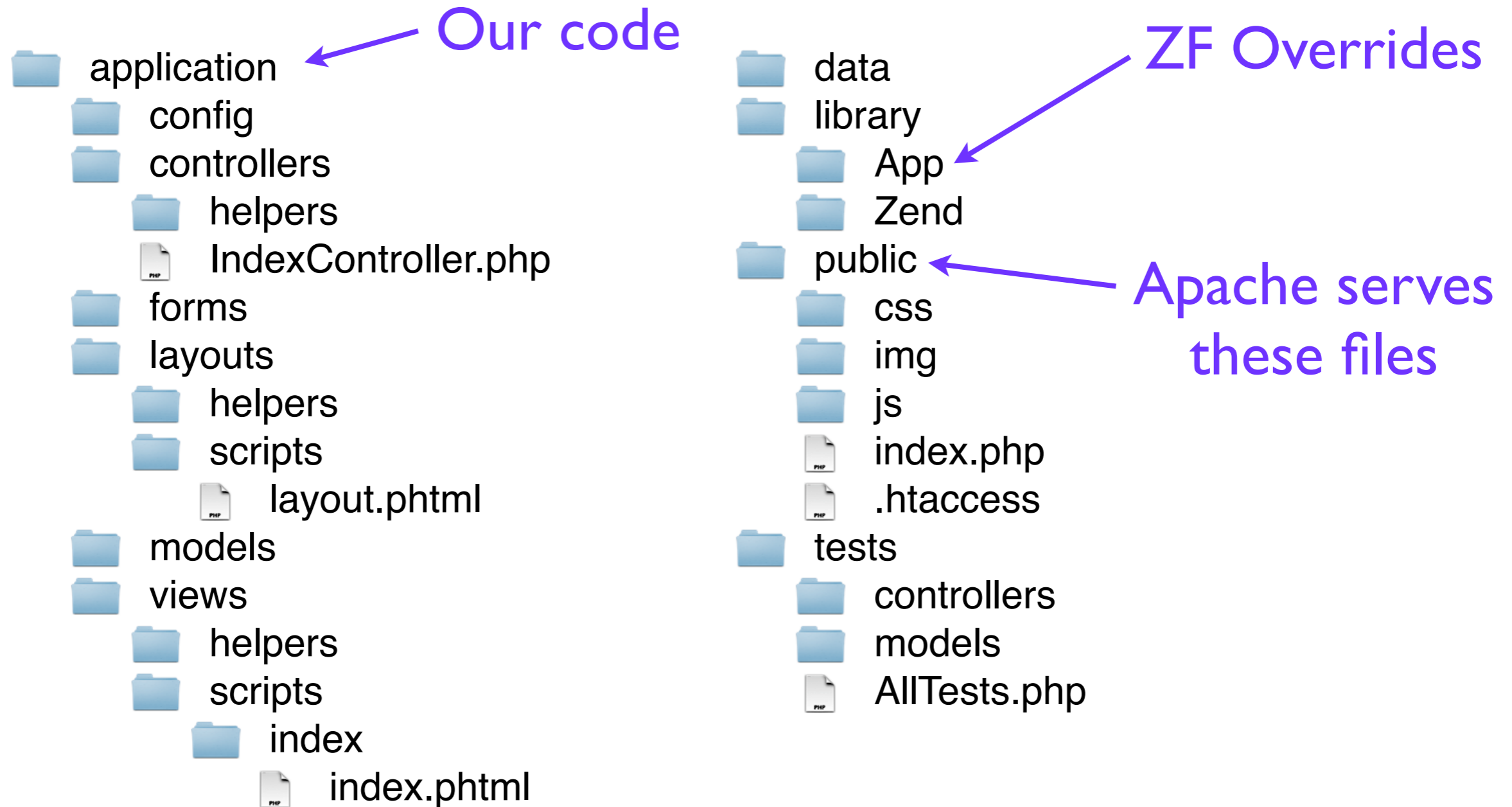
`/controller/action/param1/value1/`

or create your own:

`/news/2008/11`

```
$route = new Zend_Controller_Router_Route(
    'news/:year/:month',
    array(
        'controller' => 'news',
        'action'      => 'archive'
    ),
);
$router->addRoute('newsarchive', $route);
```

# Directory structure



# Bootstrapping

public/.htaccess

public/index.php

application/bootstrap.php

# .htaccess

```
php_value date.timezone "UTC"  
php_value short_open_tag "1"
```

← For view files

```
php_value error_reporting "8191"  
php_value display_errors "1"
```

← Don't do this in production!

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

Redirect to index.php  
if request doesn't exist

# index.php

```
define('BASE_PATH',  
    realpath(dirname(__FILE__) . '/'));  
define('APPLICATION_ENVIRONMENT', 'development');
```

```
set_include_path(BASE_PATH . '/library' .  
    PATH_SEPARATOR . get_include_path());
```

```
require_once "Zend/Loader.php";  
Zend_Loader::registerAutoload();
```

```
require BASE_PATH . '/application/bootstrap.php';  
bootstrap();
```

```
Zend_Controller_Front::getInstance()->dispatch();
```

Process  
request




# bootstrap.php

```
function bootstrap()
{
    $config = new Zend_Config_Ini(
        BASE_PATH . '/application/config/app.ini',
        APPLICATION_ENVIRONMENT);
    Zend_Registry::set('config', $config);
    Zend_Registry::set('env', APPLICATION_ENVIRONMENT);

    // Front Controller
    $frontController = Zend_Controller_Front::getInstance();
    $frontController->setControllerDirectory(
        BASE_PATH . 'application/controllers');
}
```

Singleton

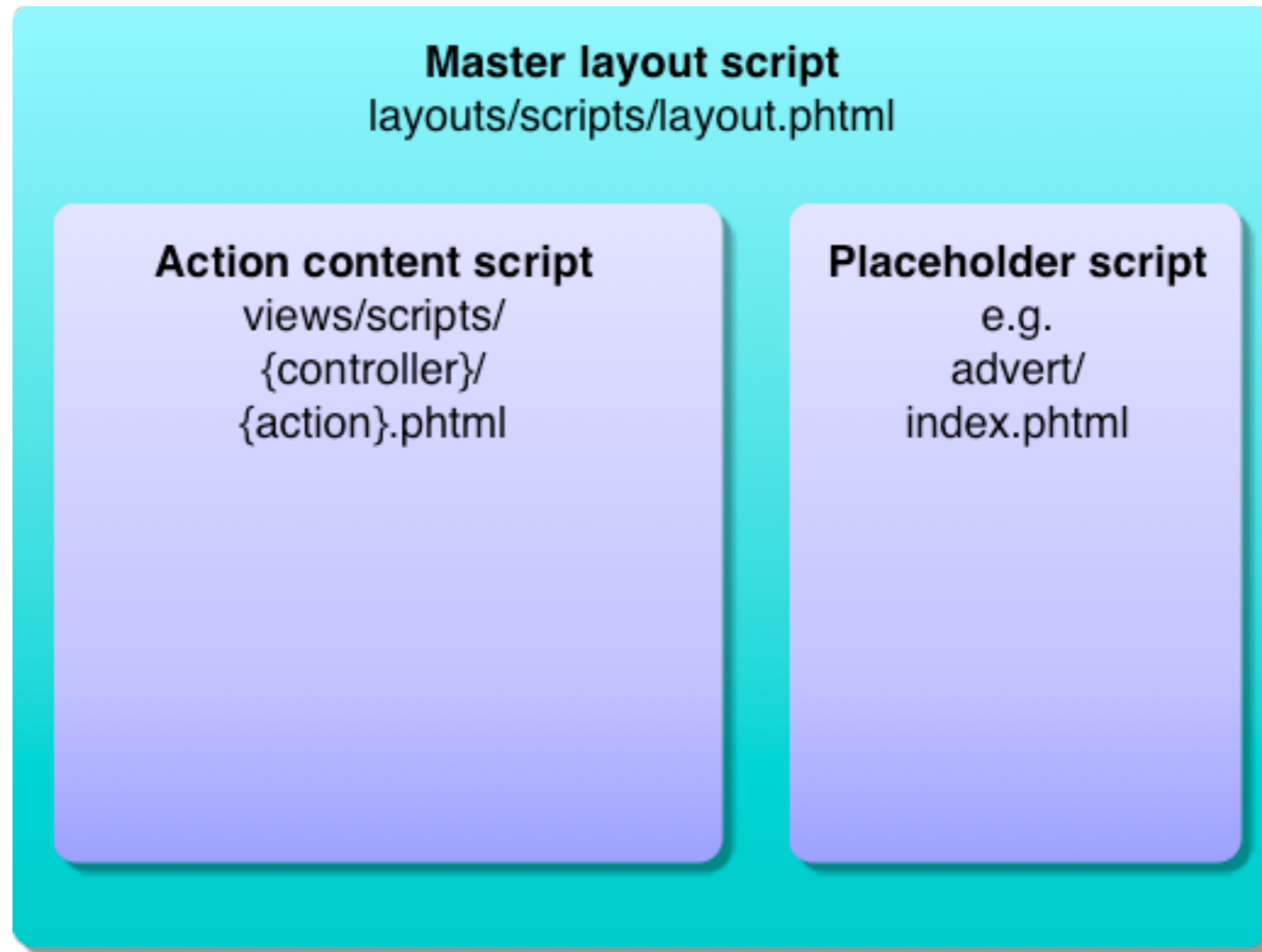




# IndexController.php

```
class IndexController extends Zend_Controller_Action
{
    public function indexAction()
    {
        $this->view->headTitle('Home');
        $this->view->title = 'Welcome';
    }
}
```

# Composite View



# Set up View

Initialise in bootstrap():

```
Zend_Layout::startMvc(BASE_PATH  
    . 'application/layouts/scripts');  
  
$view = Zend_Layout::getMvcInstance()->getView();  
$view->doctype('XHTML1_STRICT');  
$view->headTitle()->setSeparator(' - ');
```

# layout.phtml

```
<?= $this->doctype() ?>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta http-equiv="Content-Type"
        content="text/html; charset=utf-8" />
  <?= $this->headTitle('My Website');?>
  <?= $this->headLink()->prependStylesheet(
        $this->baseUrl().'/css/global.css') ?>
</head>
<body>
<div id="header"></div>
<div id="content"><?= $this->layout()->content ?></div>
<div id="footer"></div>
</body></html>
```

# index.phtml

```
<h1><?= $this->title; ?></h1>
```

```
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam lobortis nibh ac quam. Aenean urna nisi, semper ut, porttitor quis, tempor sed, eros. Etiam porta lorem. etc.</p>
```

```
<p><a href="<?= $this->url(array('controller'=>'about', 'action'=>'contact'));">Contact us</a></p>
```

# View helper

```

```

```
class Zend_View_Helper_BaseUrl
    extends Zend_View_Helper_Abstract
{
    function baseUrl()
    {
        $fc = Zend_Controller_Front::getInstance();
        return $fc->getBaseUrl();
    }
}
```

Return, don't echo!

# Model

The business logic:

- Database
- Web services
- Local files

# Database

## Zend\_Db\_Adapter

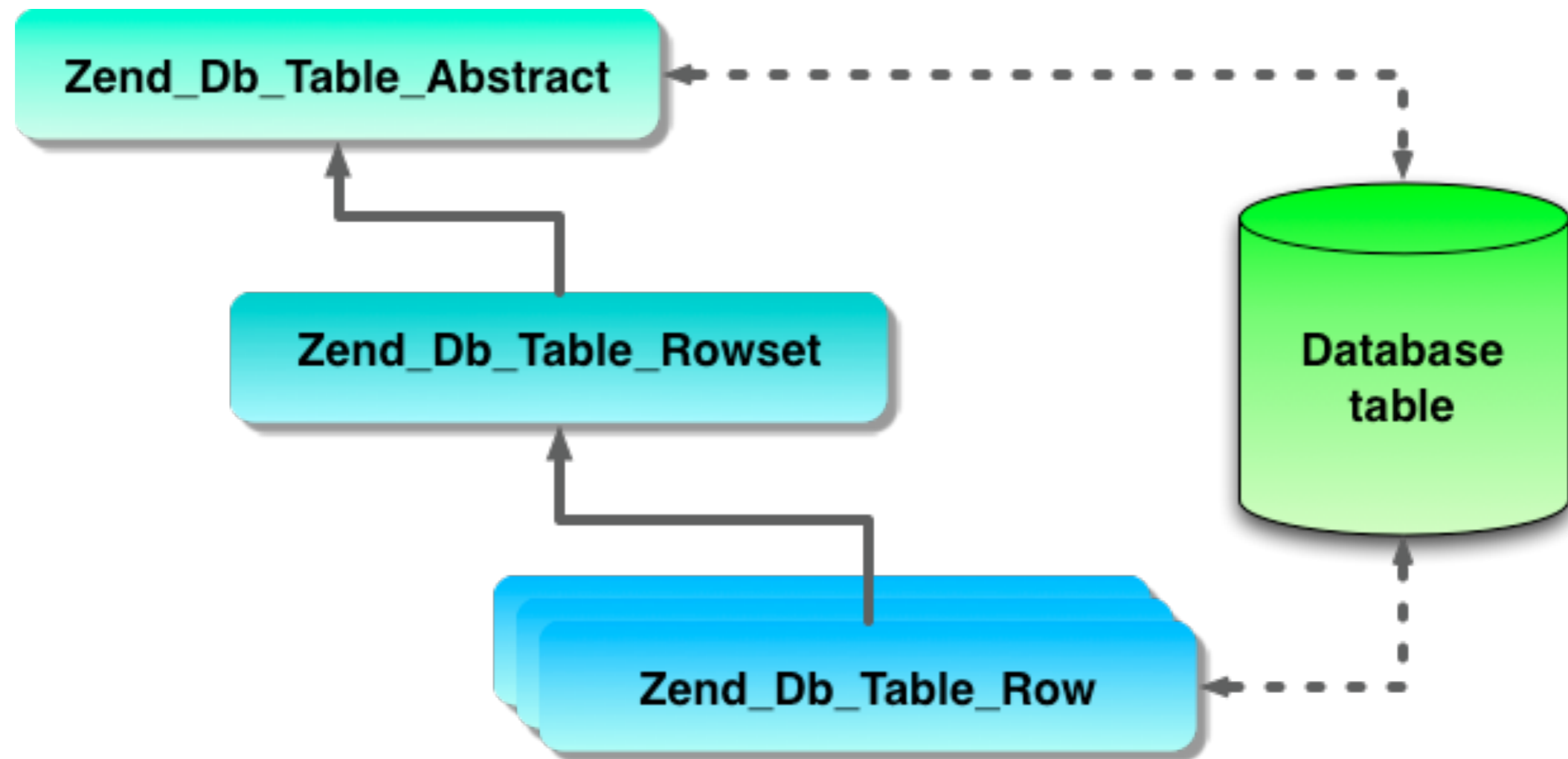
- Database abstraction

## Zend\_Db\_Table

- Table Data Gateway



# Image of Zend\_Db



# Set up database

## config/app.ini:

```
database.adapter = PDO_MYSQL  
database.params.username = rob  
database.params.password = itsasecret!  
database.params.dbname = phpnw
```

## in bootstrap():

```
$db = Zend_Db::factory($config->database);  
Zend_Db_Table_Abstract::setDefaultAdapter($db);  
Zend_Registry::set('dbAdapter', $db);
```

# A Model

## Authors

id: 1  
forename: Rob  
surname: Allen  
date\_created: 2008-11-01

id: 2  
forename: Douglas  
surname: Adams  
date\_created: 2008-11-02

# Table class

```
class Authors extends Zend_Db_Table_Abstract
{
    protected $_name = 'authors';
    protected $_rowClass = 'Author';

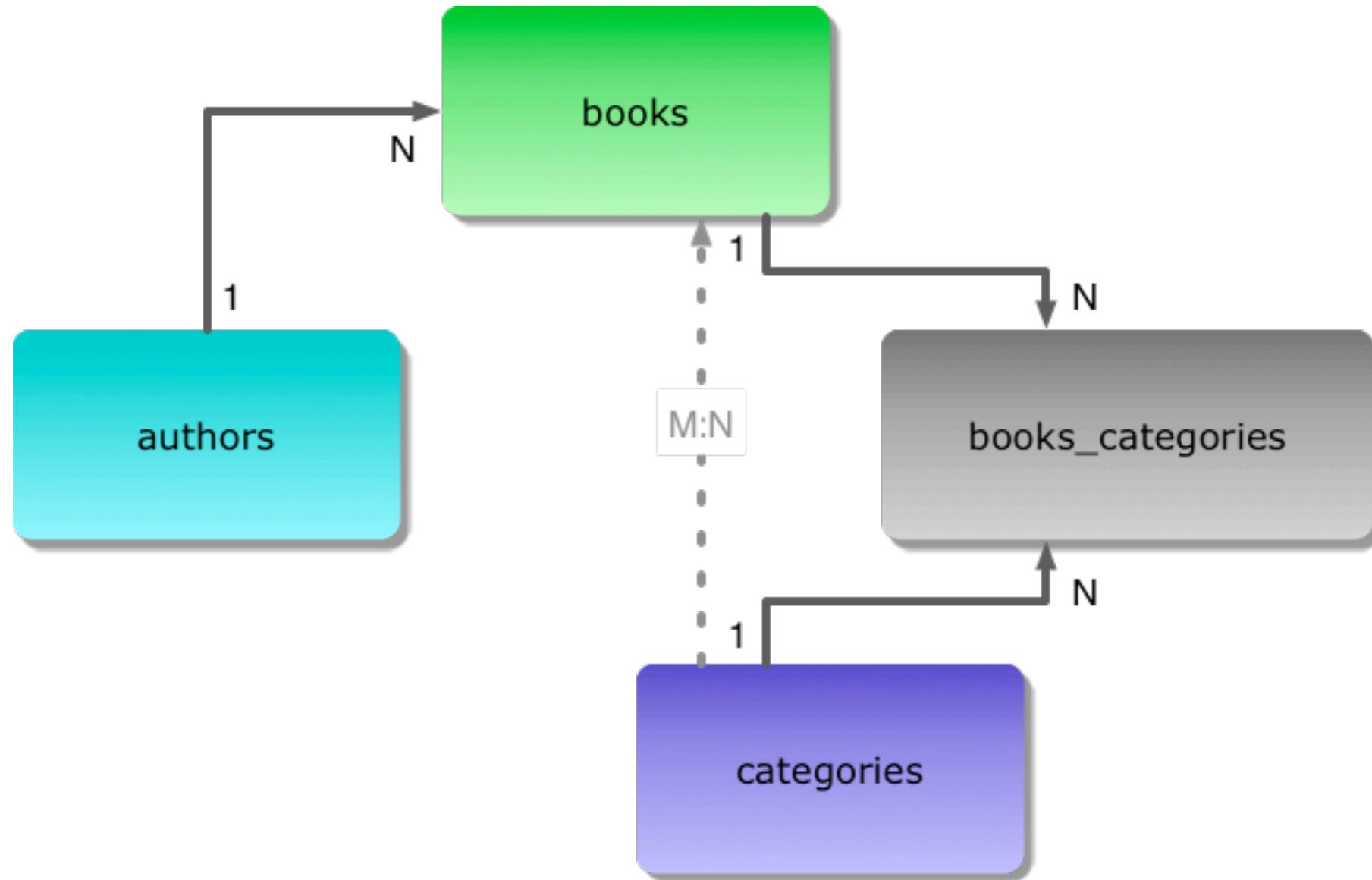
    public function fetchBySurname($surname)
    {
        $select = $this->select();
        $select->where('surname LIKE ?', $surname.'%');
        $select->order('surname ASC');
        $rowset = $this->fetchAll($select);
        return $rowset;
    }
}
```

# Row class

```
class Author extends Zend_Db_Table_Row_Abstract
{
    public function name()
    {
        return $this->forename . ' ' . $this->surname;
    }

    protected function _insert()
    {
        $this->date_created = date('Y-m-d H:i:s');
    }
}
```

# Relationships



# One to many

Authors

id: 1  
forename: Rob  
surname: Allen

id: 2  
forename: Douglas  
surname: Adams

Books

id: 2  
author\_id: 1  
title: Zend Framework in Action

id: 1  
author\_id: 2  
title: The Hitchhiker's Guide to the Galaxy

id: 3  
author\_id: 2  
title: The Restaurant at the End of the Universe

# One to many

```
class Books extends Zend_Db_Table_Abstract
{
    protected $_name = 'books';
    protected $_rowClass = 'Book';

    protected $_referenceMap    = array(
        'Author' => array(
            'columns'           => 'author_id',
            'refTableClass'    => 'Authors',
            'refColumns'       => 'id'
        )
    );
}
```



# Fetching 1:N

```
class Author extends Zend_Db_Table_Row_Abstract
{
    public function fetchBooks()
    {
        return $this->findDependentRowset('Books');
    }
}
```

```
// Controller
$authorsTable = new Authors();
$douglasAdams = $authorsTable->fetchById(2);
$books = $douglasAdams->fetchBooks();
```

# The other way

```
class Book extends Zend_Db_Table_Row_Abstract
{
    public function fetchAuthor()
    {
        return $this->findParentRow('Authors');
    }
}
```

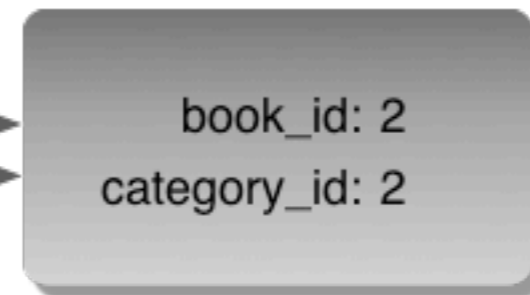
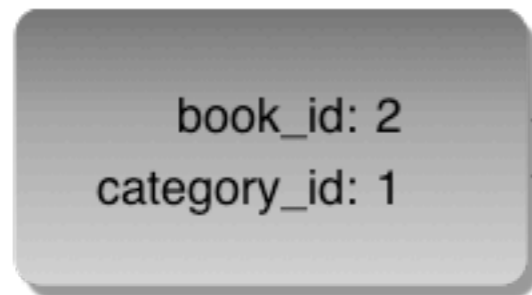
```
// Controller
$booksTable = new Books();
$hhgttg = $booksTable->fetchById(1);
$douglasAdams = $hhgttg->fetchAuthor();
```

# Many to many

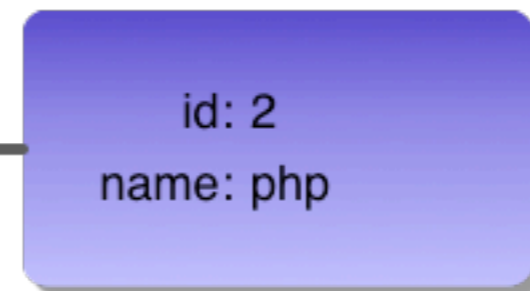
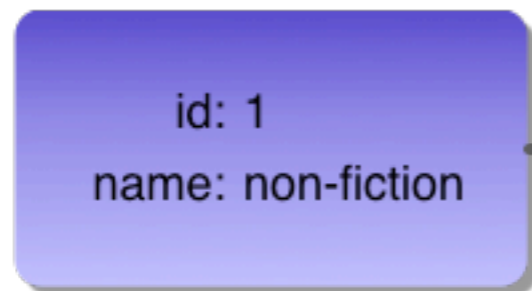
books



books\_categories



categories



# Many to many

```
class BooksCategories extends Zend_Db_Table_Abstract
{
    protected $_name = 'books_categories';
    protected $_referenceMap = array(
        'Book' => array(
            'columns' => array('book_id'),
            'refTableClass' => 'Books',
            'refColumns' => array('id')
        ),
        'Category' => array(
            'columns' => array('category_id'),
            'refTableClass' => 'Categories',
            'refColumns' => array('id')
        )
    );
}
```

# Fetching M:N

```
class Book extends Zend_Db_Table_Row_Abstract
{
    public function fetchCategories()
    {
        return $this->findManyToManyRowset(
            'Categories', 'BooksCategories');
    }
}

// Controller
$booksTable = new Books();
$zfia = $booksTable->fetchById(2);
$categories = $zfia->fetchCategories();
```

# That's it!

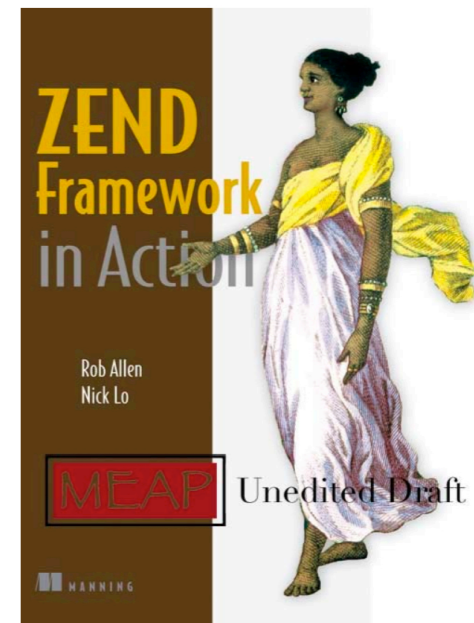
(The basics, anyway)

# Getting involved

- Mailing list
- Maintain the wiki
- Submit bug reports
- Fix bugs! (sign the CLA first...)
- Propose new components

# Available resources

- <http://framework.zend.com>
- Community
- Books - Buy my book!  
<http://www.zendframeworkinaction.com>
- Consultancy





# Thank you

Provide feedback on this talk: <http://joind.in/81>