



Building Rich Internet Applications with PHP and Zend Framework

Stanislav Malyshev

Software Architect, Zend Technologies



IDG: RIAs offer the potential for a fundamental shift in the experience of Internet applications, leading to applications that come closer to delivering on the promise of the Internet.

Overview – Building RIAs with PHP & Zend

- **RIAs: Advantages & Disadvantages**
- **Demo**
- **Why AJAX with PHP on *Zend Framework*?**
- **Code walkthrough**
- **What's next for RIAs with *PHP on Zend*?**
- **Future Demo**

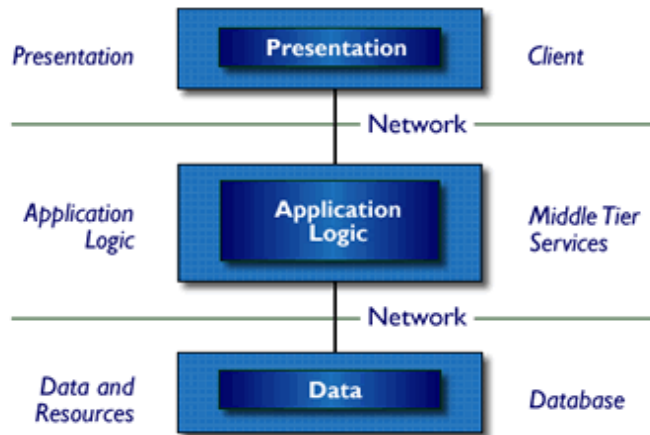
Rich Internet Applications

- **Goal: Deliver desktop experience in the browser**
- **Advantages of RIAs:**
 - Ajax is a response to the need for a richer and more easily deployable interface in the browser
 - Provide desktop-like feeling, including drag & drop, sliders, and UI changes without full page refreshes
 - More responsive
 - Less visible interaction with the server
 - Asynchronous interaction with the server
 - Leverage the deployment advantages of the browser

Rich Internet Applications

- **Disadvantages of RIAs:**

- Three tier architecture is significantly more complex



- **Requires in-depth knowledge of an additional language and platform—JavaScript & browser**

- Few established standards
- Need to deal with cross-browser compatibility

- **Not many tools in existence**



Demo – A Chat Sample Application

Credits:

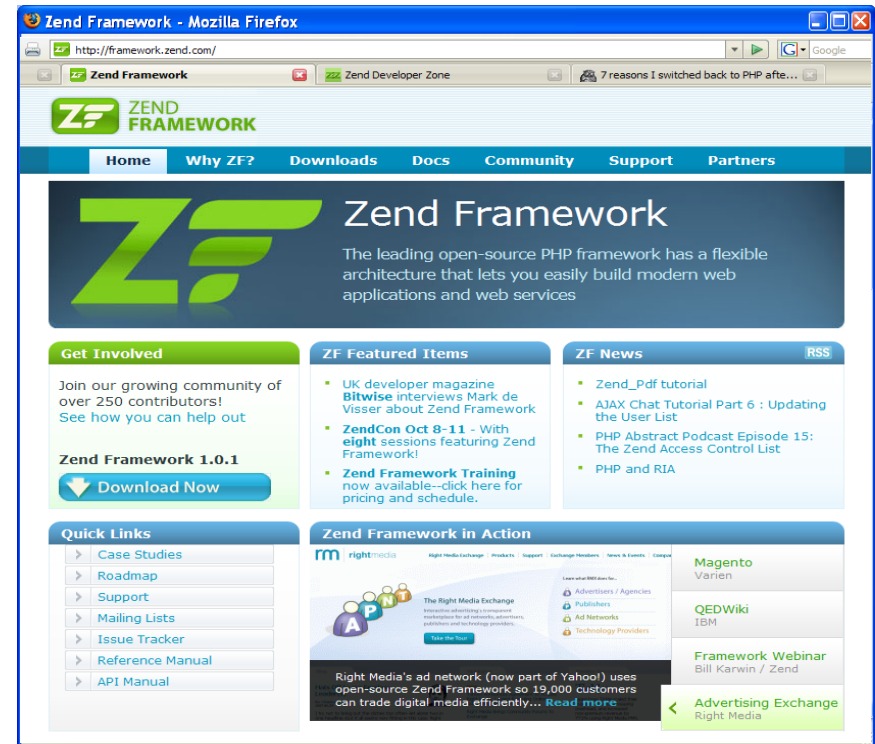
Stas Malyshev
Pádraic Brady

Sébastien Gruhier – Rich Window
prototype
script.aculo.us

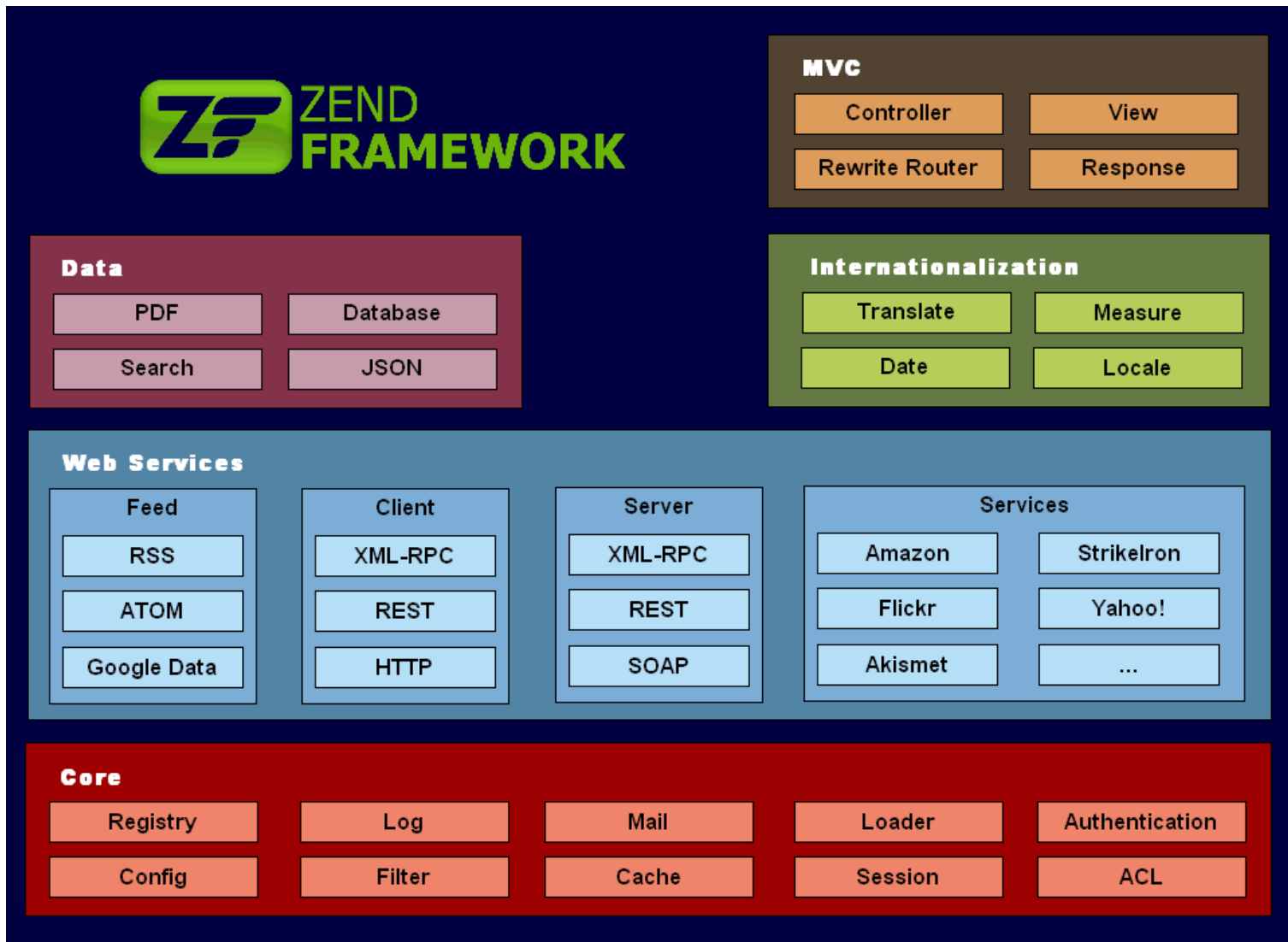
Code Walkthrough

- **Framework**
- **MVC**
- **Data handling**
- **Search**
- **Services communication**

- PHP 5 open-source web framework
- Object-oriented, unit-tests required, & focused on best practices for modern web apps
- Follows principle of “extreme simplicity,” making it easy to learn and easy to use for developers
- Community-based—led by team at Zend
- Open-source process, hosted at <http://framework.zend.com> under a business-friendly new BSD license
- Corporate contributions:



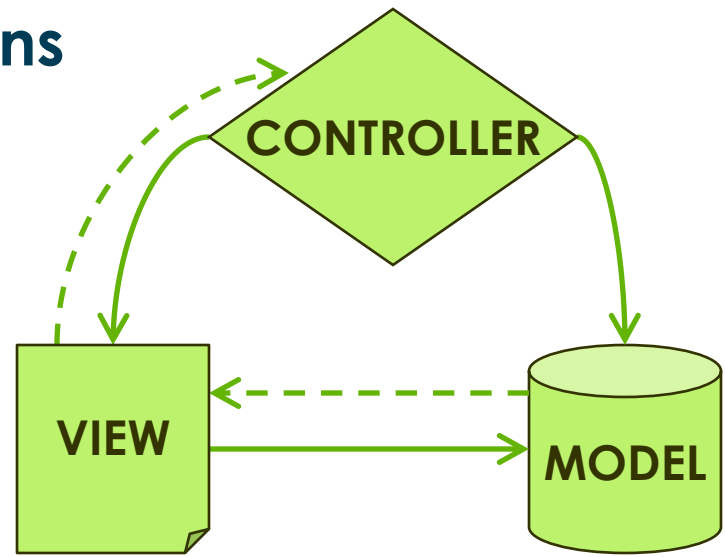
Zend Framework Architecture



What is the MVC component?

- **The heart of ZF web applications**

- **Model:** domain-specific data
- **View:** renders data to UI – PHP-based template engine
- **Controller:** processes events, invokes changes in model



- **Simple solution for most apps**

- Sensible defaults are built-in
- Flexible and extensible
- Supports advanced applications

How to use MVC: controllers

- **Controller classes handle groups of request URLs**

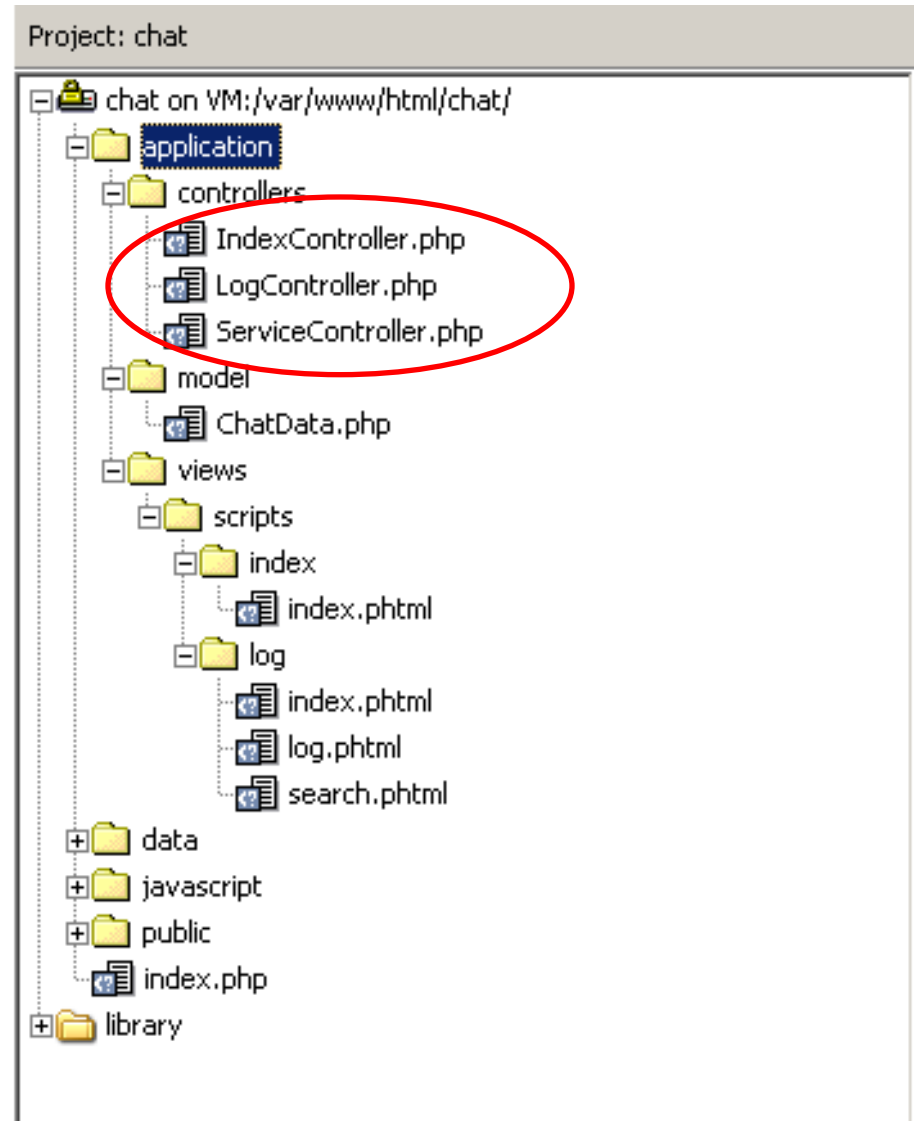
<http://zend.com/controller/action>

The default controller class is
“**IndexController**”

- **Action methods in each controller class handle individual requests**

<http://zend.com/controller/action>

The default action method is
“**indexAction()**”



Chat Application structure

/chat

IndexController

/ (index)

/ (index)

/name

/message

LogController

/log

/ (index)

/log

/search

ServiceController

/service

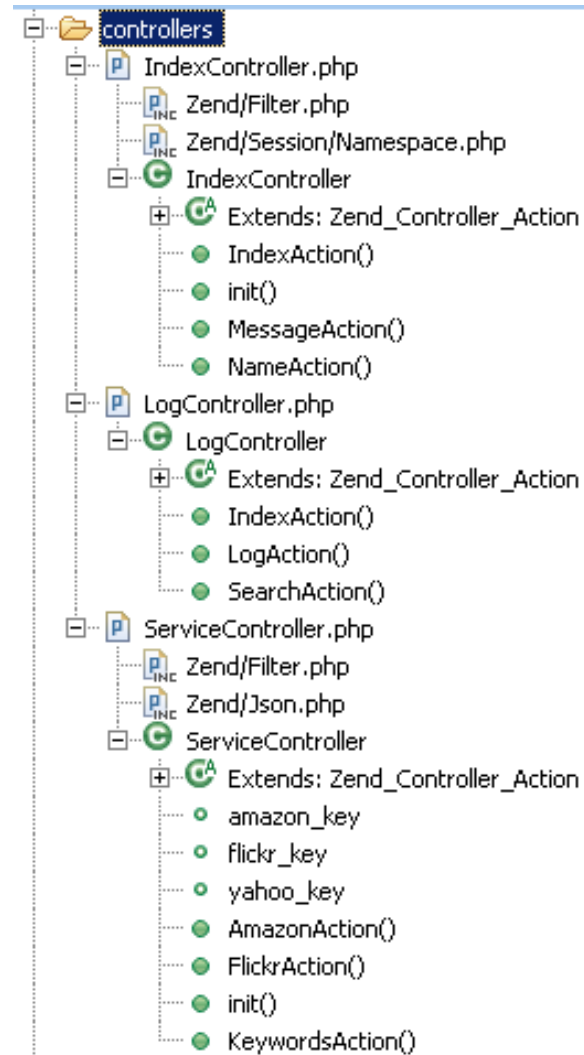
/keyword

/flickr

/amazon

Controller Actions

- Each controller action method is responsible for doing one specific task
e.g., **IndexController**:
 - `init()` for setup
 - Index is default for UI view
 - Message & Name are the AJAX action callbacks
- **Controller binds model and view together**
 - Message & Name don't get rendered by `Zend_View_Renderer`, but just return data



MVC entry point: index.php

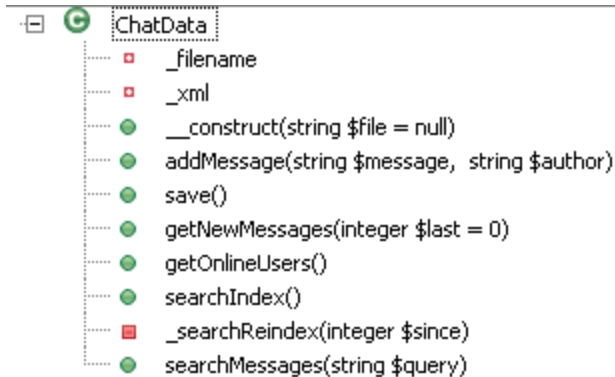
```
<?php
// Get previous session data from .xml file
$config = new Zend_Config(array(), true);
$config->datafile = './data/chat.xml';

// Store the config for other parts to use
Zend_Registry::set('config', $config);

// Setup and run the Front Controller
$controller = Zend_Controller_Front::getInstance();
$controller->setControllerDirectory('./application/controllers');
$controller->throwExceptions(true); // Exceptions ON for dev mode

// Go!
$controller->dispatch();
?>
```

Model



```
ChatData
├── _filename
├── _xml
├── __construct(string $file = null)
├── addMessage(string $message, string $author)
├── save()
├── getNewMessages(integer $last = 0)
├── getOnlineUsers()
├── searchIndex()
├── _searchReindex(integer $since)
└── searchMessages(string $query)
```

class ChatData

Encapsulates:

Session history

Search data

Implementation:

SimpleXML

Zend_Search_Lucene

Model: XML handling

Loading data

```
$this->_xml = simplexml_load_file($file);
```

Adding new message

```
$newMessage = $this->_xml->addChild('message');  
$newMessage->addChild('author', $author);  
$newMessage->addChild('timestamp', time());  
$newMessage->addChild('text', $message);
```

Saving data

```
$this->_xml->asXML($this->_filename);
```

Checking new messages

```
$newMessages = $this->_xml->xpath("/chat/message[timestamp>$last]");
```

Model: Search handling

Indexing

```
$index = Zend_Search_Lucene::open($indexfile);

$messages = $this->getNewMessages($since);

foreach($messages as $newmsg) {
    $doc = new Zend_Search_Lucene_Document();
    $doc->addField(Zend_Search_Lucene_Field::UnIndexed('timestamp',
        $newmsg['timestamp']));
    $doc->addField(Zend_Search_Lucene_Field::Text('author',
        $newmsg['author']));
    $doc->addField(Zend_Search_Lucene_Field::Text('text',
        $newmsg['text']));
    $index->addDocument($doc);
}
```

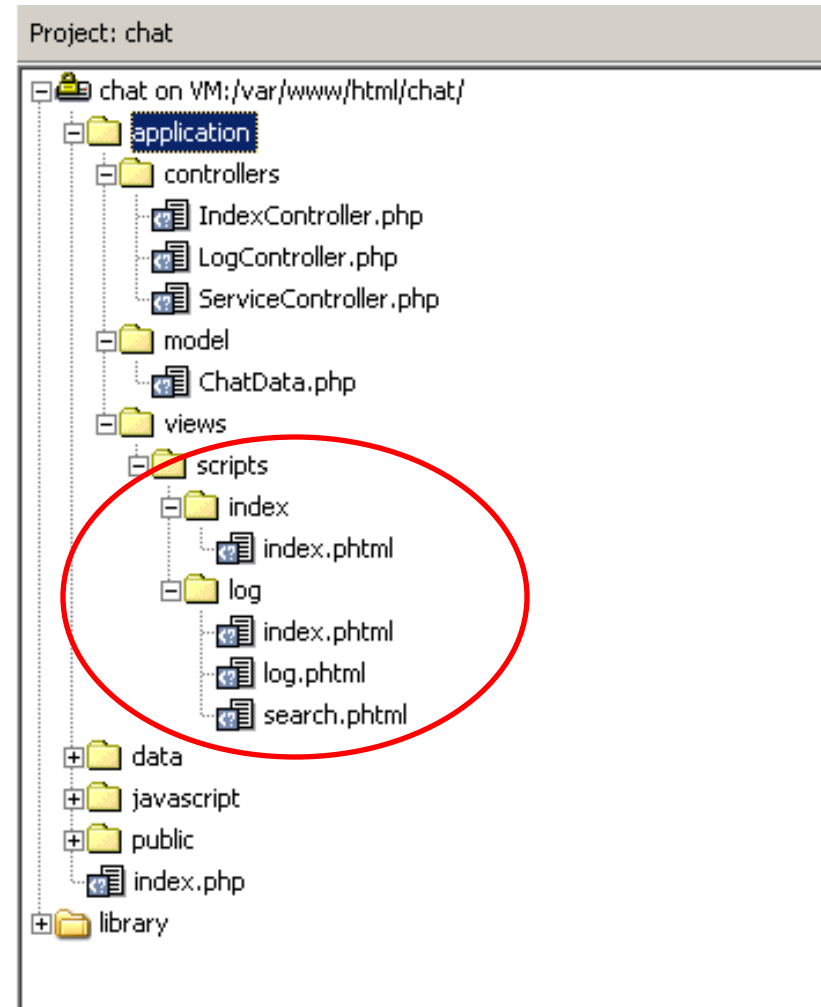
Searching

```
$index = Zend_Search_Lucene::open($indexfile);
$hits = $index->find($query);
return $hits;
```

View

- Each action has its own view template
- Templates rendered automatically
 - Unless requested not to, i.e.

```
$this->_helper->viewRenderer->  
    setNoRender(true);
```



View: displaying data

Display search results

```
<? if(count($this->hits)) { ?>
<span class="searchterm">Looking for '<?=
$this->query; ?>':</span><br/>

<? foreach($this->hits as $message) { ?>
[<?= date('c', $message->timestamp) ?>]
<span class="screenname"><?= $message->author;
?></span>:
<span class="msgtext"><?= $message->text; ?></span>
<br/>

<? } // if(count($this->hits))
} else { ?>
Nothing found for '<?= $this->query; ?>', sorry.
<? } ?>
```

AJAX communication - JSON

PHP data to JSON

```
// This function returns JSON (not a template),  
// so don't render a template  
$this->_helper->viewRenderer->setNoRender(true);  
  
$phpMessageArray = $data->getNewMessages($last);  
$onlineUsersArray = $data->getOnlineUsers();  
  
$jsonArray = array('newmessages'=>$phpMessageArray,  
                  'onlineusers'=>$onlineUsersArray);  
$responseJSON = Zend_Json::encode($jsonArray);  
  
$this->getResponse()->setHeader('Content-Type',  
                               'text/plain');  
$this->getResponse()->setBody($responseJSON);
```

AJAX communication - JSON

XML to JSON

```
$uri      = 'http://search.yahooapis.com';
$service  = '/ContentAnalysisService/V1/termExtraction';
$request  = array('appid' => $this->yahoo_key,
                 'context' => $text,
                 'output' => 'xml' );

$rest     = new Zend_Rest_Client();
$rest->setURI($uri);

$response = $rest->restPost($service, $request);

$this->getResponse()->setHeader('Content-Type',
                              'text/plain');
$this->getResponse()->setBody(
    Zend_Json::fromXML($response->getBody()));
```

Handling services - Flickr

```
$flickr = new Zend_Service_Flickr($this->flickr_key);
$flickrSearchoptions = array('page'=>1,
    'sort'=>'interestingness-desc');
$results = $flickr->tagSearch($keywords,$flickrSearchoptions);
// Collect results into PHP array
$phpRes = array();
foreach($results as $result) {
    $newres = array();
    $newres['id'] = $result->id;
    $newres['title'] = $result->title;
    $img = $result->Small; $newres['src'] = $img->uri;
    $newres['w'] = $img->width; $newres['h'] = $img->height;
    $newres['clickUri'] = @$result->Original->clickUri;
    $phpRes[] = $newres;
}
// Send the results out as JSON data
$this->getResponse()->setHeader('Content-Type', 'text/plain');
$this->getResponse()->setBody(Zend_Json::encode($phpRes));
```

What's next?

- **AJAX-enabled Form component in Zend Framework**
 - Working on requirements and proposal
- **AJAX support in development tools – Eclipse-based**
 - JavaScript editing – syntax highlighting, code completion
 - JavaScript debugging
 - Toolkit support (for instance, class browsers)
 - Opens up opportunity for using Flex
- **Significantly grow support for Web Services vendors**
- **Important enhancements to our Lucene implementation**
 - Range queries, wildcard queries
 - Support for Lucene 2.3 file format (faster, better, backwards compatible, ...)

Zend Component Model

What?

Toolkit for building AJAX applications

How?

Building blocks model – a-la VB/Delphi/ASP/JSF

Why?

RAD in PHP spirit – simplicity + power

Zend Component Model

- **Zend Framework**

- Component model
 - PHP component architecture
 - Server side messaging & persistence
 - Ajax controller
- Ajax Toolkit
 - Client side messaging
 - Client-server connectivity



- **Zend Studio (Eclipse)**

- Drag&drop of components
- Component configuration
- Ajax Toolkit support (class browsing, documentation, etc.)
- Javascript editing and debugging
- Advanced CSS support
- Cross language debugging (stepping from Javascript into PHP)



Zend Component Model

- **Component architecture**
 - Both visual and non-visual components
- **Rich-design time experience ala Delphi and Visual Basic**
 - Drag&drop of components onto the Web page
- **Component is self-contained incl. design and run-time rendering & property editors**
 - IDEs don't have to be hard coded per-component
 - Encourages eco-system of components
- **WYSI~~A~~WYG**
- **Programmatic model**
 - PHP developers like code and not declarative syntax

What are components?

- **PHP+JS+CSS client/server side bundle**
- **Self-describing components – can be used with any tool**
- **Visual RAD**
 - design time view – no data required
 - production view - reflects actual data
- **Vi-compatibility – no additional configurations or declarative syntax files, only PHP and CSS/JS on client side**
- **Extensibility – new components, extending existing components**

Components Demo

Mozilla Firefox

File Edit View History Bookmarks Tools Help del.jcio.us

http://127.0.0.1/cm/page2.php

id	name	street	city	state	zip	country
61	Jack S. Rocha	903-2288 Vehicu	Austin	Nevada	12299	USA
62	Jaime S. Moren	Ap #734-7314 A	Hoboken	North Da	49792	USA
63	Callie A. Gallow	P.O. Box 876, 32	Idabel	Wiscons	68115	USA
64	Sierra R. Todd	P.O. Box 111, 12	Richmond	MD	73755	USA
65	Kiara M. York	1312 Blandit St.	Wilson	ID	13465	USA
66	Melissa E. Nobl	Ap #724-7895 L	West Sacramento	Iowa	85636	USA
67	Jordan J. Muno	P.O. Box 866, 53	Longview	CA	71787	USA
68	Luke Y. Nichols	P.O. Box 800, 7E	Ponce	Michigan	04325	USA
69	Quynn T. Boyer	6184 Sed St.	Claremont	Michigan	05097	USA
70	Peter W. Le	P.O. Box 587, 65	Madison	Iowa	01992	USA

Page 7 of 10

Add to favorites Clean Remove from favorites

id	name	address
14	Madeson A. Adams	Athens, Texas
65	Kiara M. York	Wilson, ID
9	Sara M. Rivera	Boulder City, SC
91	Nadine G. Conner	Trenton, AL

1312 Blandit St., Wilson ID 13465

Zend Components Example App

Map Satellite Hybrid

POWERED BY Google

Map Data ©2007 TeleAtlas - Terms of Use

id	65
name	Kiara M. York
street	1312 Blandit St.
city	Wilson
state	ID
zip	13465
country	USA

Components Demo

Mozilla Firefox
File Edit View History Bookmarks Tools Help del.jicio.us
http://127.0.0.1/cm/page2.php

Zend Components Example App

Table component

id	name	street	city	state	zip	country
61	Jack S. Rocha	903-2288 Vehicu	Austin	Nevada	12299	USA
62	Jaime S. Moren	Ap #734-7314 A	Hoboken	Nevada	49710	USA
63	Callie A. Gallow	P.O. Box 876, 32	Idabel	Wisconsin	68115	USA
64	Sierra R. Todd	P.O. Box 111, 12	Richmond	MD	73755	USA
65	Kiara M. York	1312 Blandit St.	Wilson	ID	13465	USA
66	Melissa E. Nobl	Ap #724-7895 L	West Sacramento	Iowa	85636	USA
67	Jordan J. Muno	P.O. Box 866, 53	Longview	CA	71717	USA
68	Luke Y. Nichols	P.O. Box 800, 7E	Ponce	Michigan	04325	USA
69	Quynn T. Boyer	6184 Sed St.	Claremont	Michigan	5097	USA
70	Peter W. Le	P.O. Box 587, 65	Madison	Iowa	01992	USA

Page 7 of 10

GMap component

Frame component containing Table component

id	name	street	city	state	zip	country
65	Kiara M. York	1312 Blandit St.	Wilson	ID	13465	USA

Button component

1312 Blandit St., Wilson ID 13465

Demo – Server code – setup()

```
<?php
class My_Page extends Zend_Component_Page {
function setup() {
$this->DB1 = $this->create('DB_Table', 'DB1');
$this->DB1->driver = 'PDO_MYSQL';
$this->ContactsTable =
    $this->create('AJAX_Table_Paged', 'ContactsTable');
$this->ContactsTable->data = 'DB1';
$this->ContactsTable->rows = 10;

$this->Map = $this->create('Google_Map', 'Map');

$this->AddButton = $this->create('Button', 'AddButton');
$this->AddButton->text = "Add to favorites";
}
}
$page = new My_Page();
```

Demo – Server code - presentation

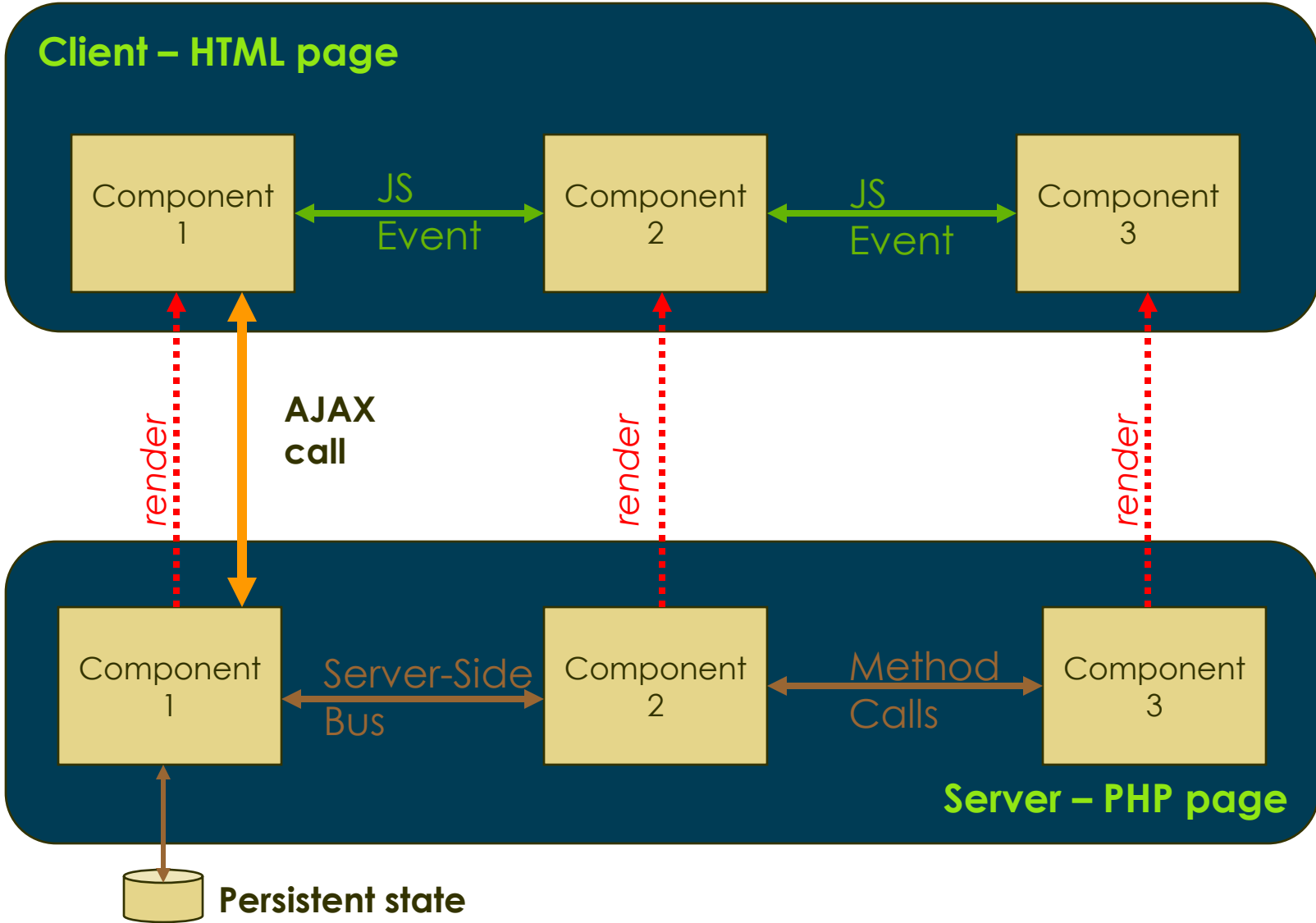
```
<html>
<head><title>Zend Components Demo App</title>
<link type="text/css" rel="stylesheet"
      href="<?= $page->staticURL('demo.css') ?>" />
<? $page->head(); ?></head>
<body>
<? $page->DB1->show(); ?>
<div id="page">
  <div id="header"><h1>Zend Components Example App</h1></div>
  <h2>Contacts</h2>
  <div id="ContactsTable" class="box">
<? $page->ContactsTable->show(); ?></div>
  <span id="AddButton"><? $page->AddButton->show(); ?></span>
  <div id="Map"><? $page->Map->show(); ?></div>
<script>...</script>
</body>
</html>
```


Demo – Client code

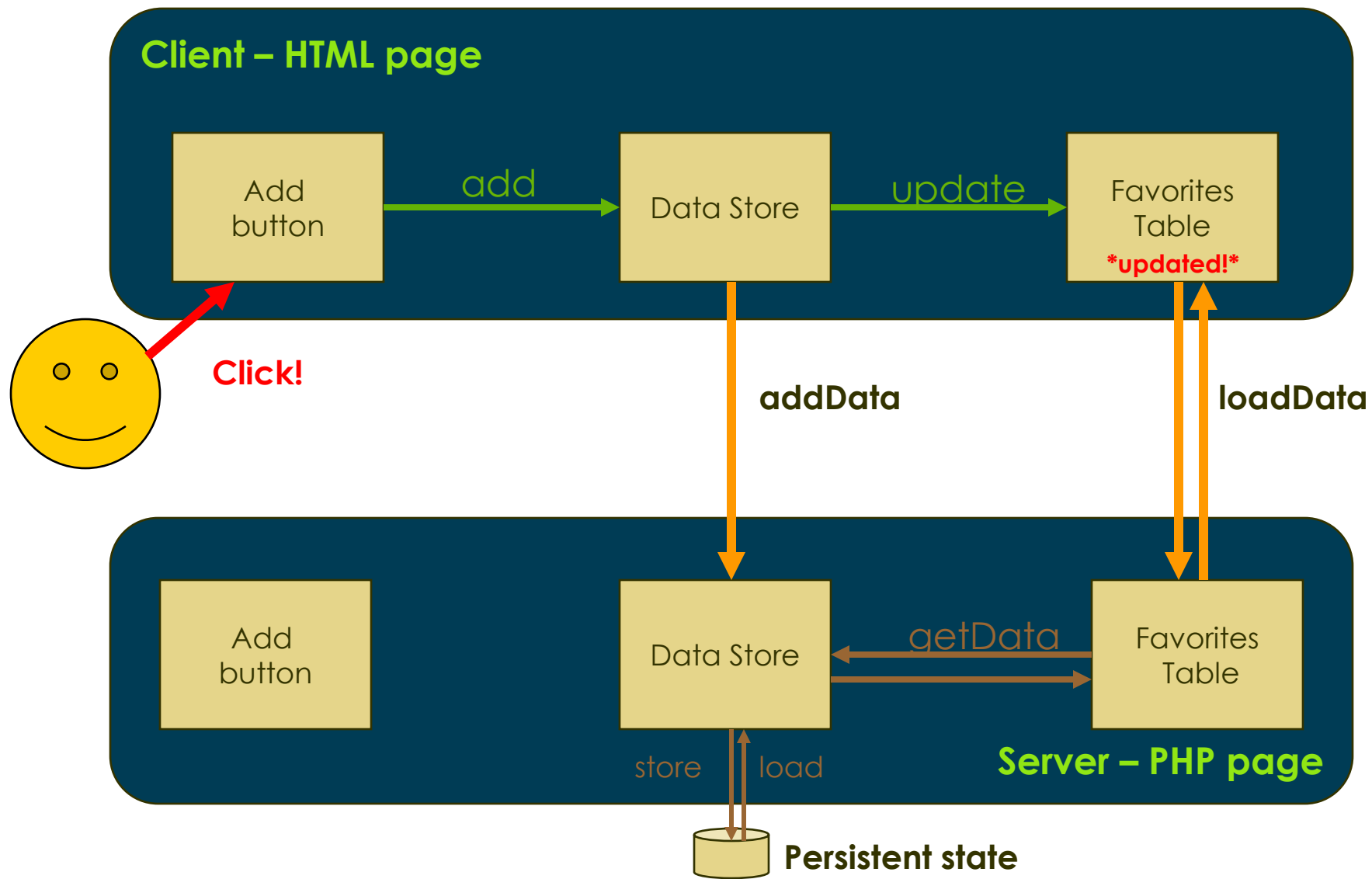
```
function mkUrl(data) {
    if(data[1]) {
        return ["/cm/getdata.php?id="+data[1][0]];
    }
}

zcConnect(ZC_ContactsTable, "selected",
    ZC_Map, "address", mkAddress);
zcConnect(ZC_ContactsTable, "selected",
    ZC_AddressFrame, "load", mkUrl);
zcConnect(ZC_DB3, "datachanged",
    ZC_FavoritesTable, "refresh");
zcConnect(ZC_FavoritesTable, "selected",
    ZC_Map, "address", mkAddress2);
zcConnect(ZC_CleanButton, "click", ZC_DB3, "clean");
zcConnect(ZC_RemoveButton, "click",
    ZC_DB3, "remove", delFav);
zcConnect(ZC_AddButton, "click", ZC_DB3, "add", addFav);
```

Component Interaction



Interaction Scenario



Zend Component Model - Future

- **Component nesting**
- **Drag & drop support**
- **Visual event wiring**
- **Event standardization**
- **Tools for extending components**



Thanks!

Stanislav Malyshev stas@zend.com

Please send feedback!

<http://devzone.zend.com/article/2442-PHP-and-RIA>