



A look at the Zend Framework

Silvan Zurbrügg, Liip AG

Webtuesday, 13. November 2007

 **Buzz**

PHP5 Library for Web-development

Simplicity & Productivity

Corporate friendly licensing

Agile, well tested codebase

Loosely coupled components



History

Initiated in October 2005 by Zend in California

First preview in March 2006

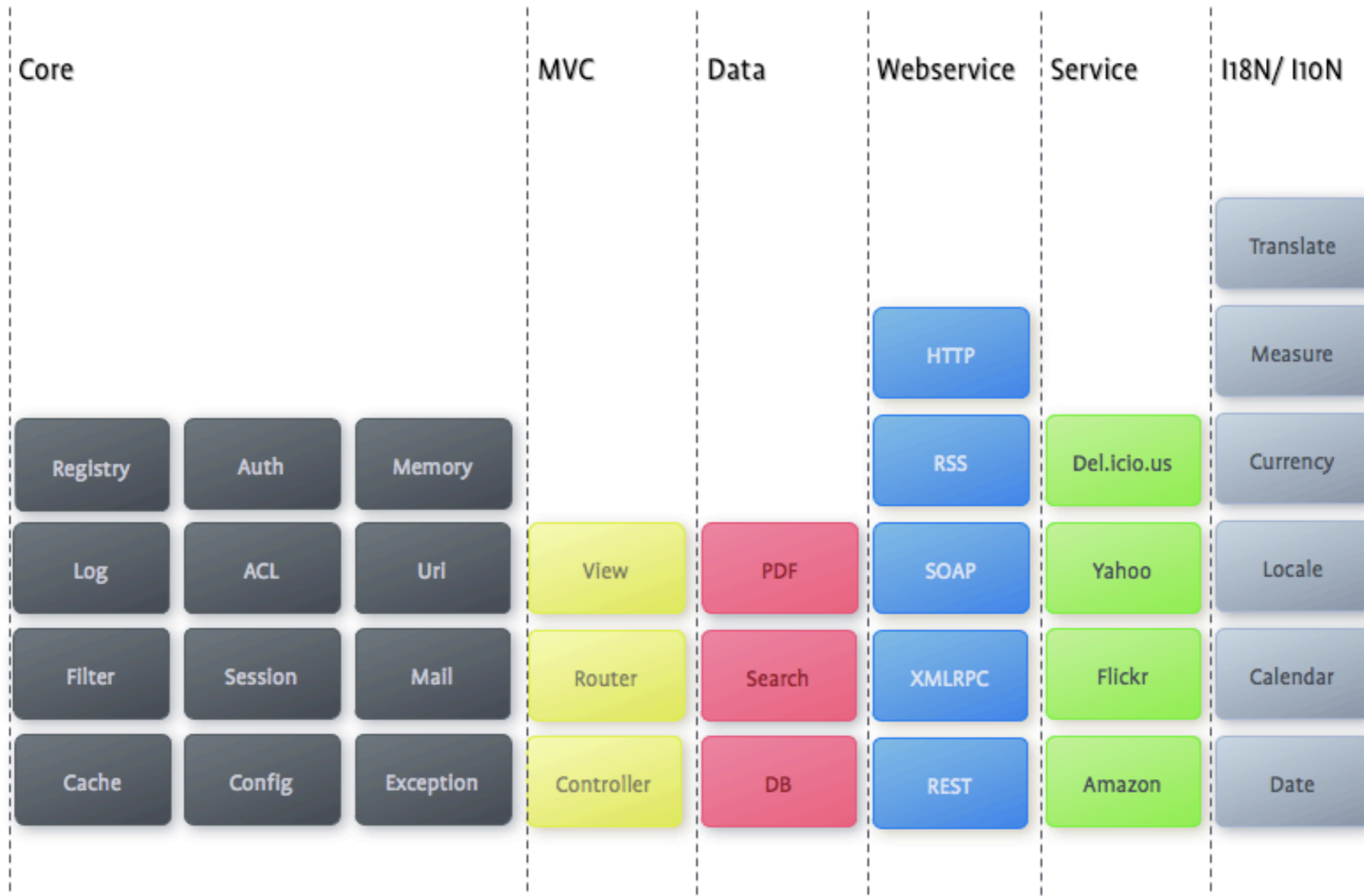
12 official releases since 2006

Current release: 1.0.2

Check the weekly summaries on:

http://devzone.zend.com/tag/Weekly_Summaries

Components

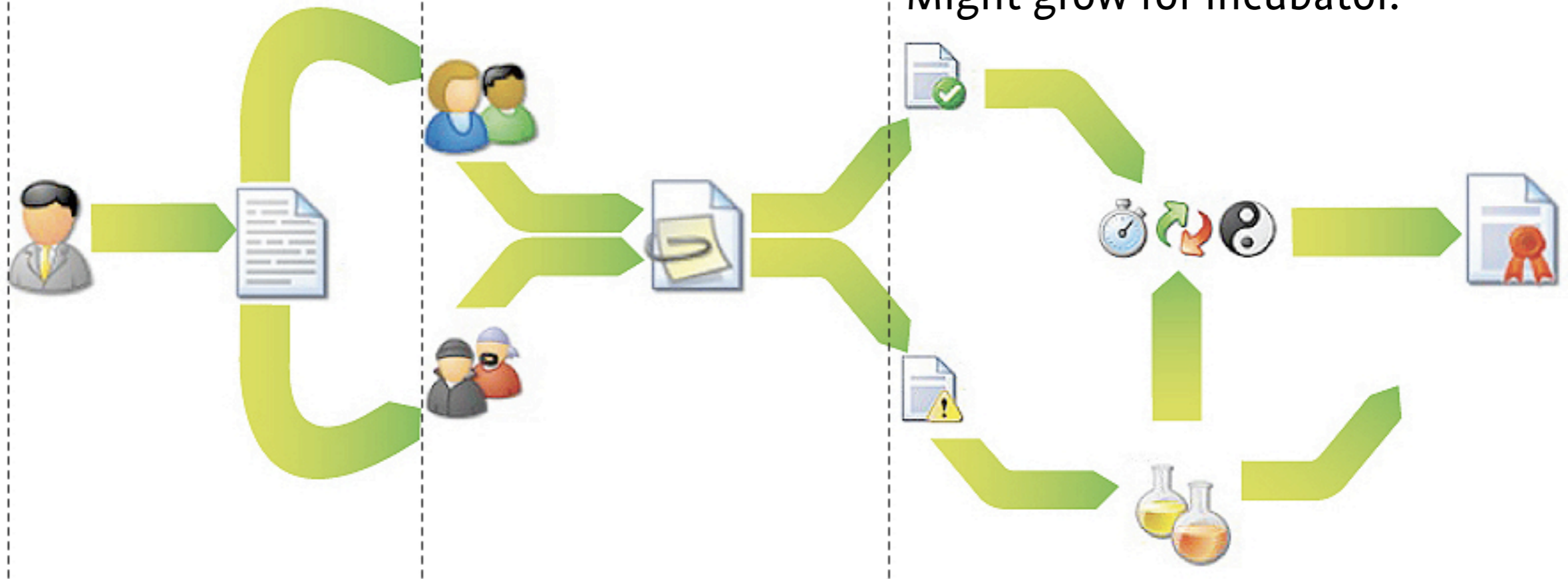


Lifecycle

Developer submits
a proposal

Review, feedback
from community
and core team

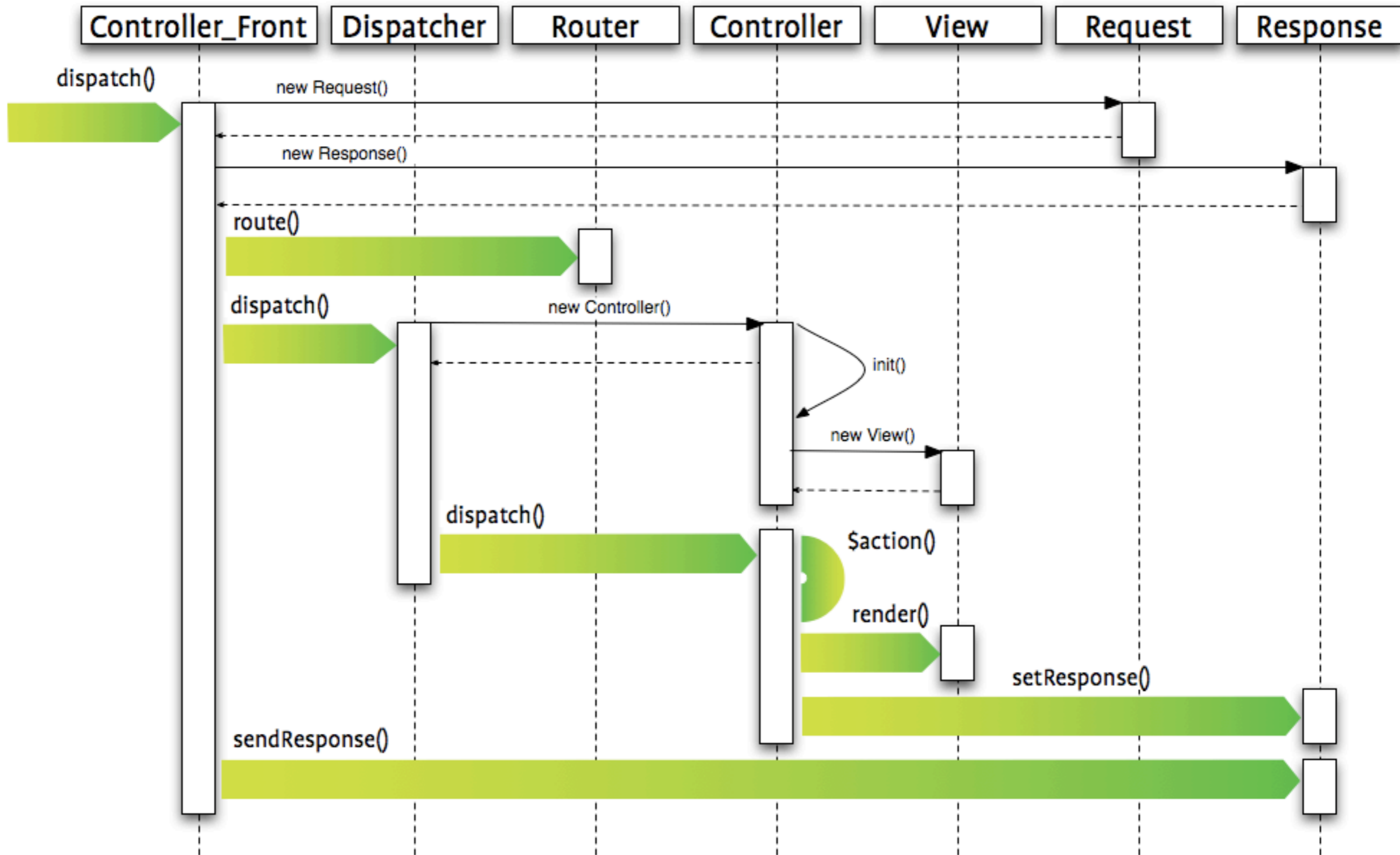
Ok'ed proposal goes to incubator,
finally to core after revisions.
Not ok'ed proposal goes to laboratory
Might grow for incubator.





A more detailed look at Zend Framework's MVC

Zend MVC request sequence



Zend MVC url mapping

Default:

`http://example.com/tag/add/`

-> `TagController->addAction()`

Using Modules:

`http://example.com/tagmodule/tag/add`

-> `Tagmodule_TagController->addAction()`

Zend_Controller_Router

- Decompose URI-endpoint to determine the Module, Controller and Action.
- Called once before dispatching the first controller.
- Available Routers:
 - `Router_Route()`
 - `Router_Route_Static()`
 - `Router_Route_Regex()`

Zend_Controller_Router

Mapping all actions for a module to a single Controller:

```
$route1 = new Zend_Controller_Router_Route('tag/:action/:id',  
    array('controller' => 'Tag_Index'));  
  
$router = new Zend_Controller_Router_Rewrite();  
$router->addRoute('myroute', $route1);  
  
[...]  
  
$controller->setRouter($router);
```



Zend_Controller

```
Class Tag_IndexController extends Zend_Controller_Action {

    // Initialise the controller and prepare the view
    public function init() {
        $this->viewSuffix = 'xsl';
        $this->view = $this->initView();
    }

    // The default index-Action
    public function indexAction() {
        $datadom = new DOMDocument();
        $datadom->loadXML('<response><saywhat>Hello World</saywhat></response>');
        $this->view->data = $datadom;
        $this->render();
    }

    // Instantiates and returns the View
    public function initView() {
        $view = new Zend_View_Xslt();
        if ($view instanceof Zend_View_Abstract) {
            $view->addScriptPath('/var/www/sf/dev/views/scripts/');
            return $view;
        }
    }
}
```



Zend_Controller

```
[...]
```

```
// Tell the view to render the xml.  
public function render($action = null, $name = null, $noController = false) {  
    if ($this->view instanceof Zend_View_Abstract) {  
        $script = $this->getViewScript($action, $noController);  
        $this->getResponse()->appendBody(  
            $this->view->render($script),  
            $name  
        );  
    }  
}
```

Zend_Controller

- Forwarding:
`$this->_forward($actionName, $ctrlName);`
- Redirecting:
`$this->_redirect($url);`

Zend_View

- Default view rendering is done via:
`Zend_Controller_Action_Helper_ViewRenderer`
- Simple execution of view-scripts through inclusion:
`include func_get_arg(0);`
- Disable default view renderer:
`$ctrl->setParam('noViewRenderer', true);`



Zend_View

Let's create Zend_View_Xslt ...

```
Class Zend_View_Xslt extends Zend_View_Abstract {

    public function render($name) {
        $xsl = '';
        foreach($this->getScriptPaths() as $sp) {
            if (file_exists($sp . $name)) {
                $xsl = $sp . $name;
                break;
            }
        }

        if (!empty($xsl) && $this->data instanceof DOMDocument) {
            $xslDom = new DOMDocument();
            if ($xslDom->load($xsl)) {
                $xslDom->xinclude();
                $xslProc = new XsltProcessor();
                $xslProc->registerPHPFunctions();
                $xslProc->importStylesheet($xslDom);

                try {
                    $xml = $xslProc->transformToDoc($this->data);
                    if ($xml instanceof DOMDocument) {
                        return $xml->saveXML();
                    }
                }
            }
        }
    }
}
```

[...]

Plugins and Helpers

- **Action Helpers**

Hook into `preDispatch()` or `postDispatch()` of the Controller.

Access Helpers directly from Controllers
`$this->_helper->myHelper->myfunc()`

- **Plugins**

Hook into `routeStartup()`, `routeShutdown()`,
`dispatchLoopStartup()`, `preDispatch()`,
`postDispatch()`, `dispatchLoopShutdown()`



Demo:

Searching and reading

Blogposts offline.



Ingredients

Indexer:

- Zend_Config
- Zend_Db
- Zend_Db_Select
- Zend_Search_Lucene
- Zend_Cache
- Zend_Http_Client

Ingredients

Frontend

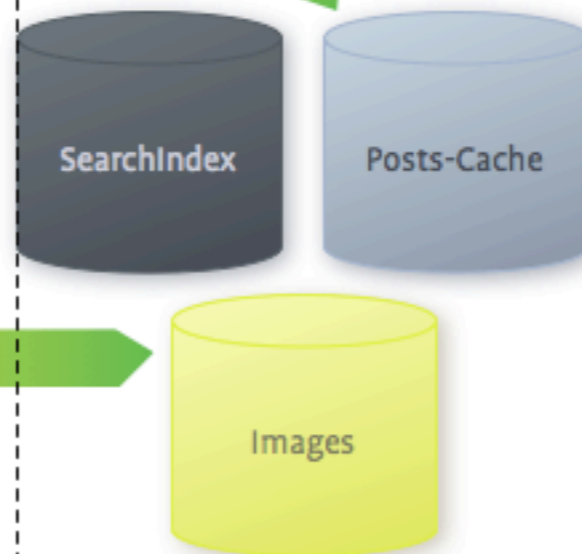
- Zend's MVC (Controller, View)
- Zend_Search_Lucene
- Zend_Cache

How it works

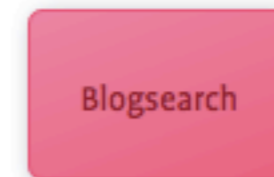
Blog (Online)



Indexer (On/Offline)



Blogsearch (Offline)





Questions?

- Slides will be available on either <http://blog.var.cc/> or <http://slides.liip.ch/>
- Feel free to contact me:
silvan@liip.ch

 **Thank you :)**