

# The Butler Project

Larry Garfield



In the beginning...



"Drupal doesn't have multiple layouts. It has one layout it bends a lot."

--Earl Miles, c.2009



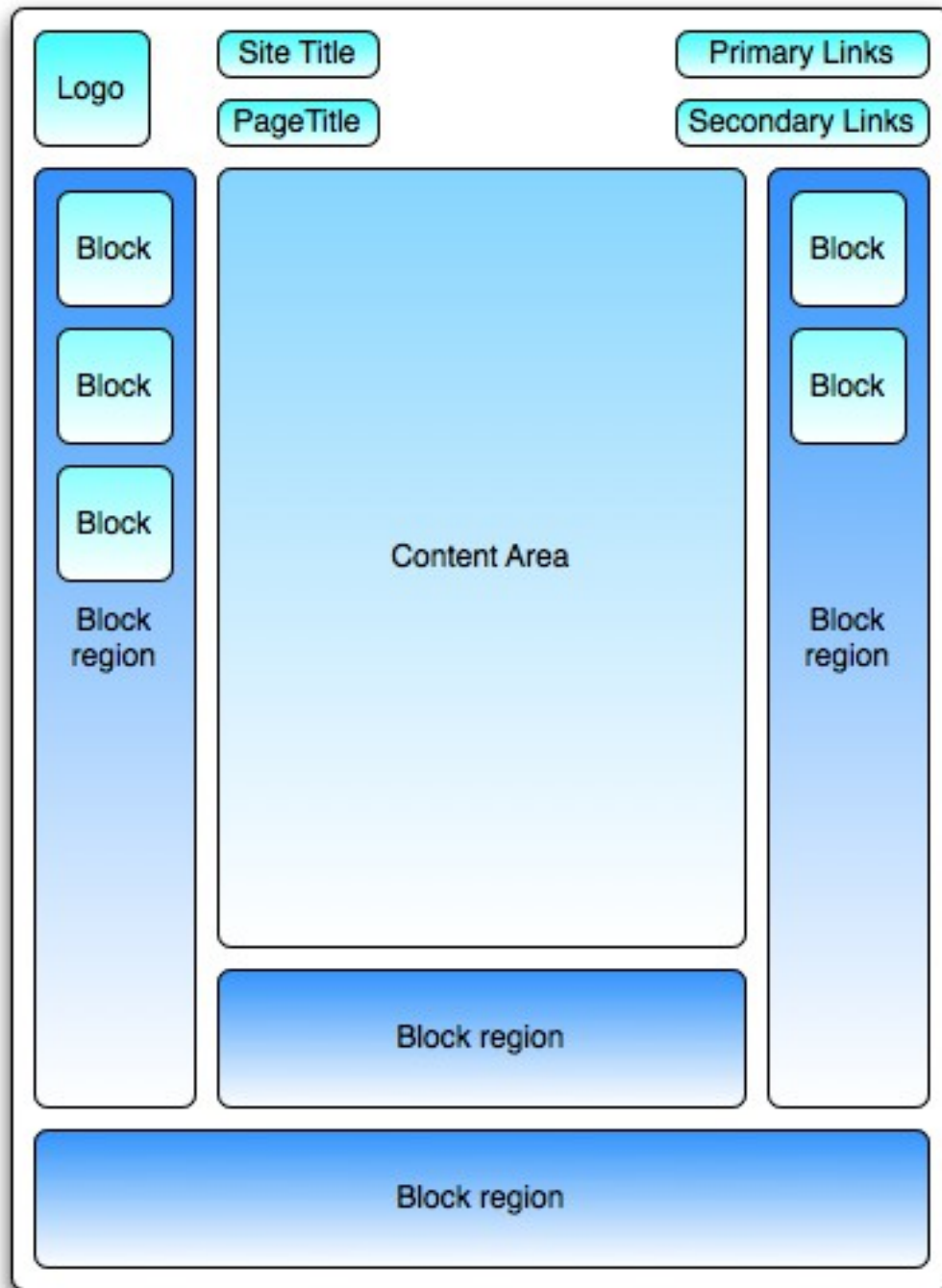
# DrupalCon San Francisco: Blocks TNG

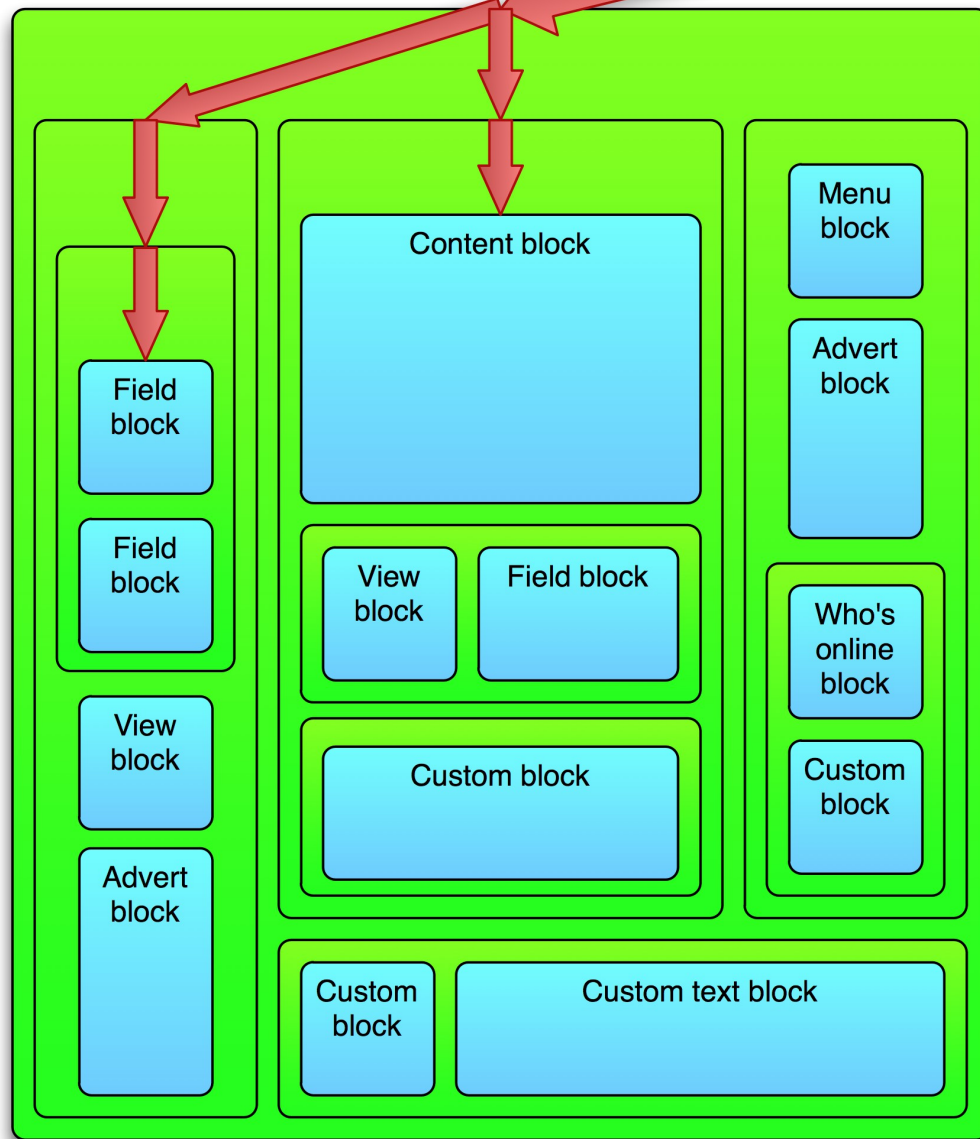
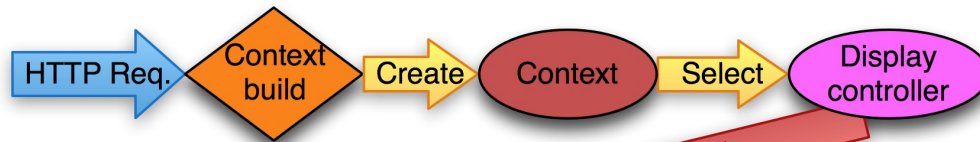


# Layout fail: There Can Be Only One

- Blocks: totally inadequate
- Contrib
  - Panels: PHP 4, complex, hacks around core
  - Context/spaces
  - Fancy hacks (views\_attach?)
- hook\_page\_alter
  - Limits of push-based
  - Huge arrays eat memory, require intimate knowledge







<Insert lots of discussion here>

<http://groups.drupal.org/butler>





# Five Fails

- 1) No unified context
- 2) One Layout to Rule them All
- 3) No plugin mechanism
- 4) HTML-only
- 5) Performance



# DrupalCon Copenhagen: The Butler Roadmap



# The Butler Project

*"Give Drupal a unified, powerful context system that will support smarter, context-sensitive, easily cacheable block-centric layouts and non-page responses using a robust unified plugin mechanism."*



1) Unified, powerful **context system**

2) Unified **plugin** mechanism

3) **Smart block**-centric layouts

4) Any type of **response**



One step at a time...



# Step 1: Unified Context Broker



drupal\_match\_path()                      global \$language

user\_access()                      book\_build\_active\_trail()

global \$user                      menu\_get\_active\_object()

drupal\_eval()                      arg()                      menu\_get\_active\_trail()

global \$theme\_info

Spaces                      global \$pager\_array

Panels                      pig entrails                      og\_get\_group()

variable\_get('language\_negotiation', LANGUAGE\_NEGOTIATION\_NONE)



Drupal has too much context





Drupal has no context



# Drupal needs context



Drupal needs context *management*





*"Be a good chap and let me know what language I should be displaying..."*

# Mediator object

The mediator pattern provides a unified interface to a set of interfaces in a subsystem.

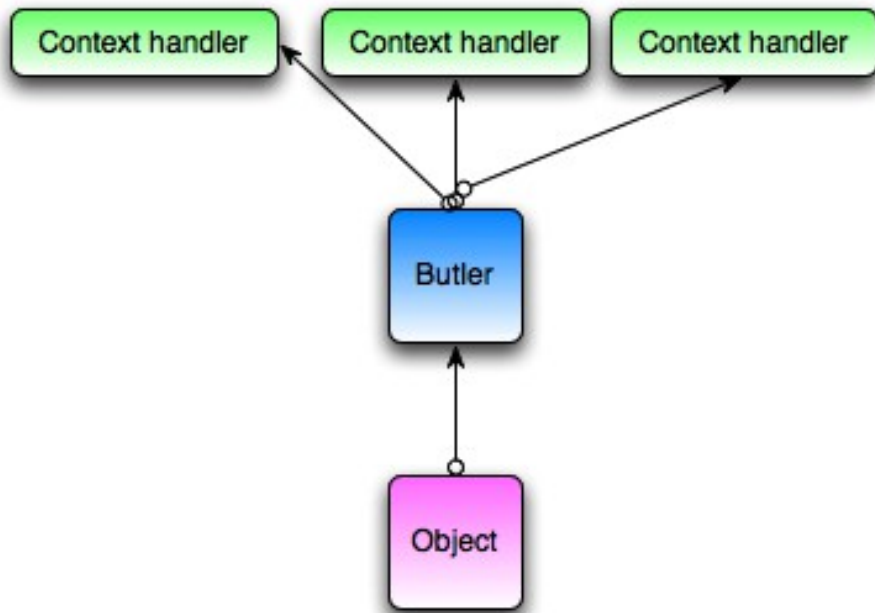
*--Wikipedia*



# The Butler/Mediator

- Intermediary for context questions
- Register context handler
- Request result from Butler context
- Plus the HTTP request itself
  - PECL HTTP?
  - David Naughton (nihiliad)





# Work in progress...

<http://drupal.org/project/butler>

(Courtesy Copenhagen)





```
class Foo {
    protected $context;

    public function __construct($context) {
        $this->context = $context;
    }

    public function behave() {
        $node = $this->context['node'];
        $var = $this->context['http:get:var'];
        // ...
    }
}
```



```
function behave() {  
    $context = drupal_get_context();  
    $language = $context['language'];  
    $foo = $context['http:header:foo'];  
}
```



```
$butler->registerHandler('http:get', 'CtxHandlerHttp',  
array('query' => 'GET'));
```

```
$butler->registerHandler('http:post', 'CtxHandlerHttp',  
array('query' => 'POST'));
```

```
$butler->registerHandler('http:header', 'CtxHandlerHeader');
```



```
$context->registerHandler('og', 'ContextOgHandler');  
// ...  
$current_og = $context['og:nid'];
```



# Context mocking

**Tell the Butler to lie to your children**



```
$new_ctx = $butler->clone();  
$new_ctx->registerHandler('og', 'AlternateHandler');  
$new_ctx['og:nid'] = 5;  
  
$new_ctx->lock();  
$plugin = get_plugin('stuff', 'blah', $new_ctx);
```



# Step 2: Plugins

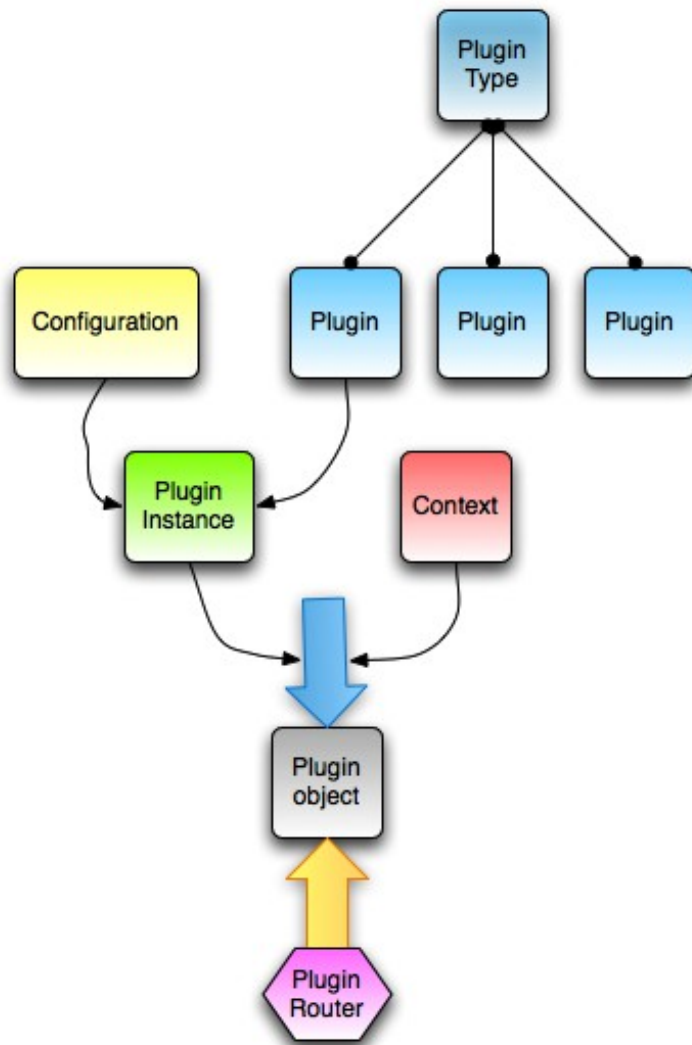


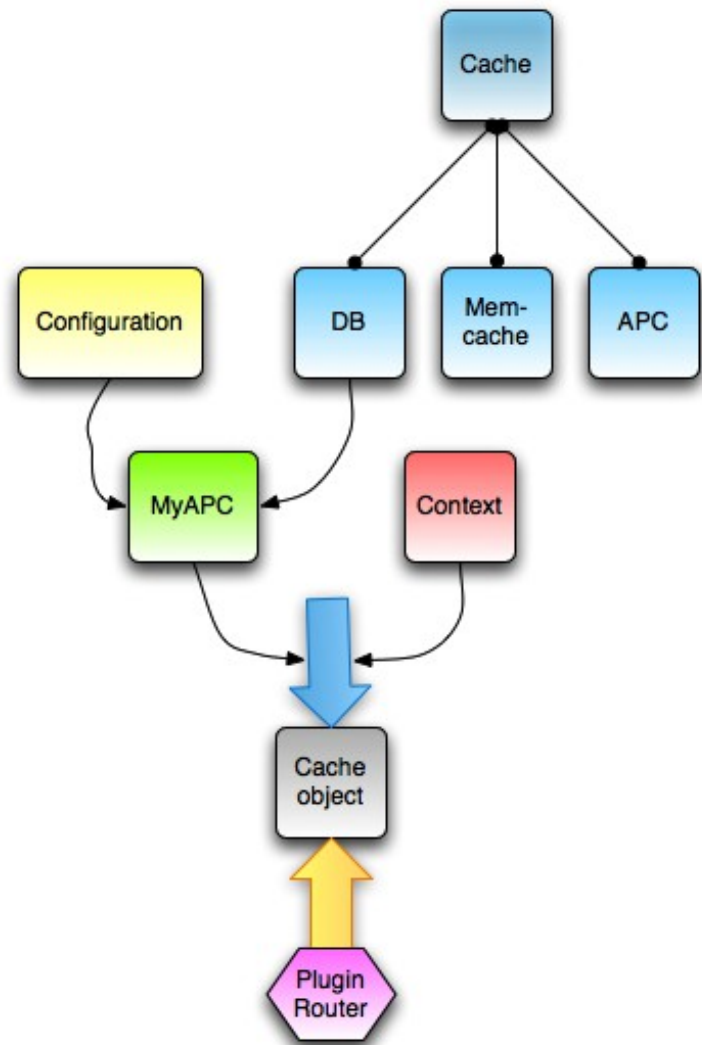
# We have too many

- One-off hacks
- Alt. include files
- Callback functions
- Pseudo-hooks
- caching system with OO
- Views
- Plugins/Handlers
- Ctools Plugins
- Webform components
- Etc.







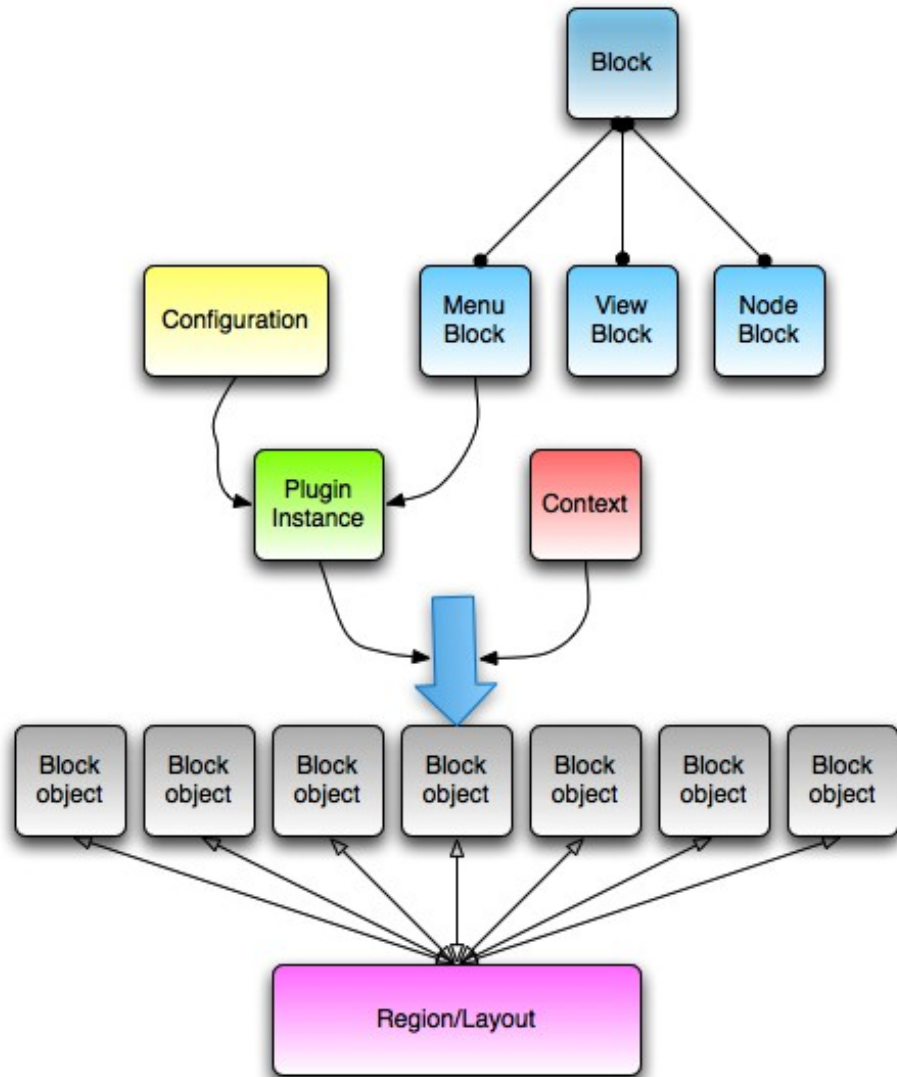


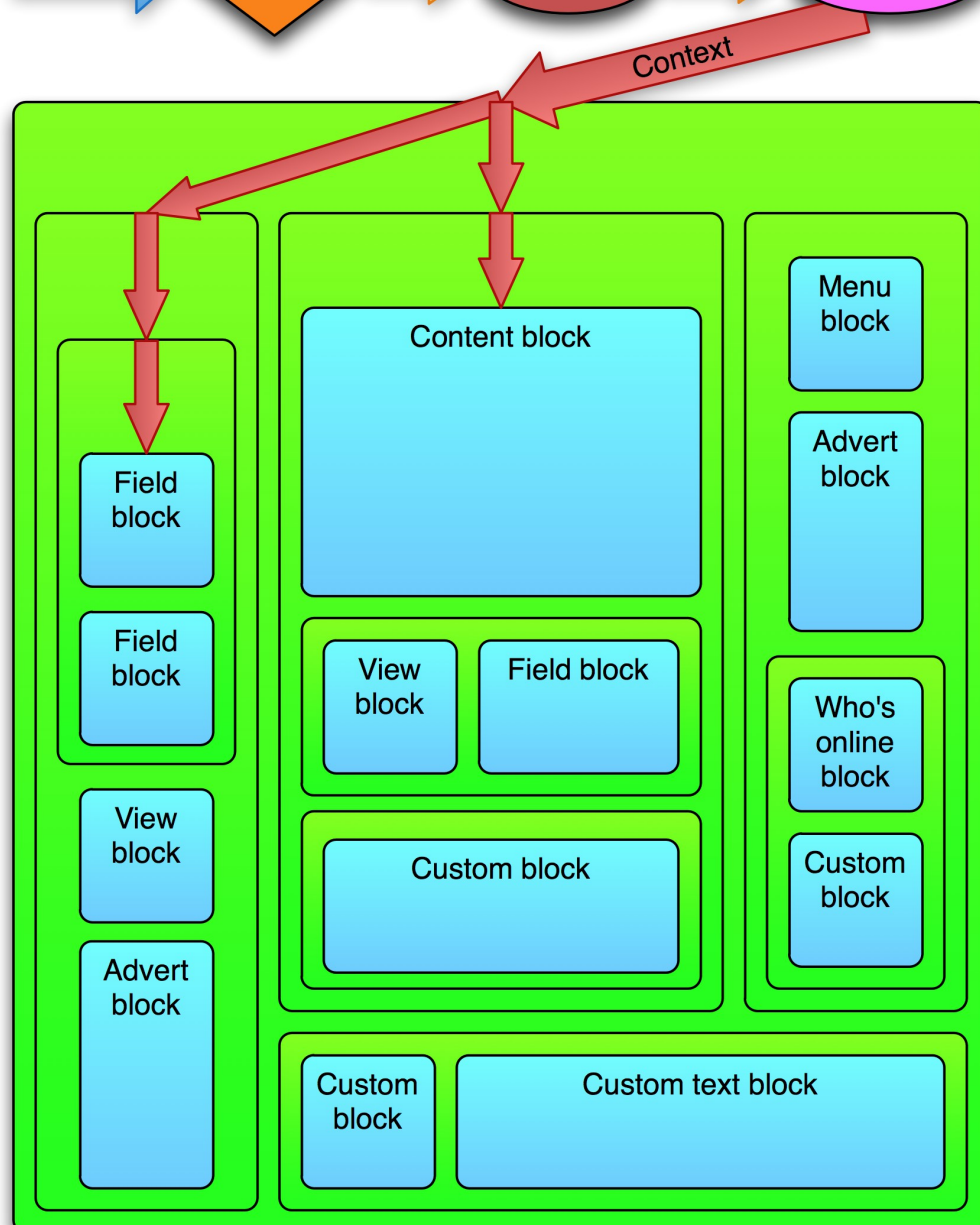
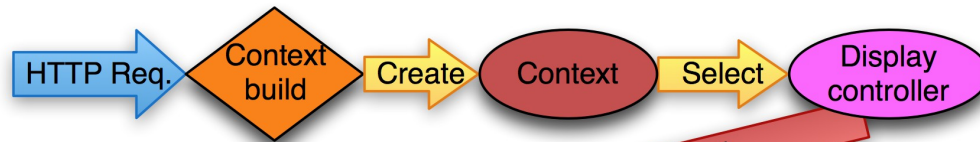
# Advantages

- One API to learn
- Vertical extensibility
- Well-encapsulated
- Picklable to code
- Well-understood problem space

And...







# Open questions

- Context caching
- Plugin model (ctools? Zend Framework?)
- Bootstrap
- Transition plan



# Discuss

