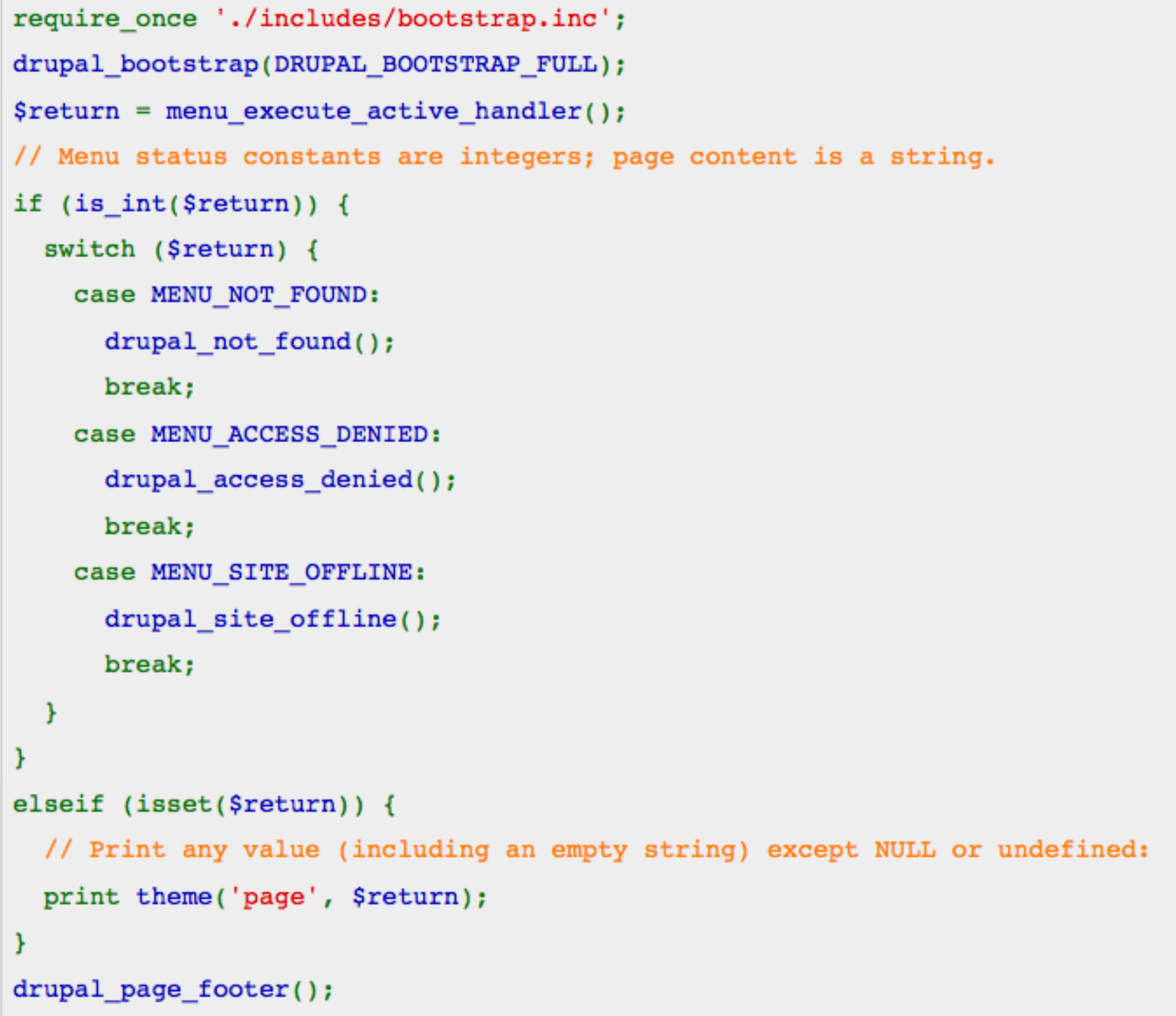
Page serving and rendering (XHTML/XML/JSON, etc.) - now and Drupal 7

Peter Wolanin

August 29, 2008

Drupalcon, Szeged

All pages go through index.php



```
require_once './includes/bootstrap.inc';
drupal_bootstrap(DRUPAL_BOOTSTRAP_FULL);
$return = menu_execute_active_handler();
// Menu status constants are integers; page content is a string.
if (is_int($return)) {
  switch ($return) {
    case MENU_NOT_FOUND:
      drupal_not_found();
      break;
    case MENU_ACCESS_DENIED:
      drupal_access_denied();
      break;
    case MENU_SITE_OFFLINE:
      drupal_site_offline();
      break;
  }
}
elseif (isset($return)) {
  // Print any value (including an empty string) except NULL or undefined:
  print theme('page', $return);
}
drupal_page_footer();
```

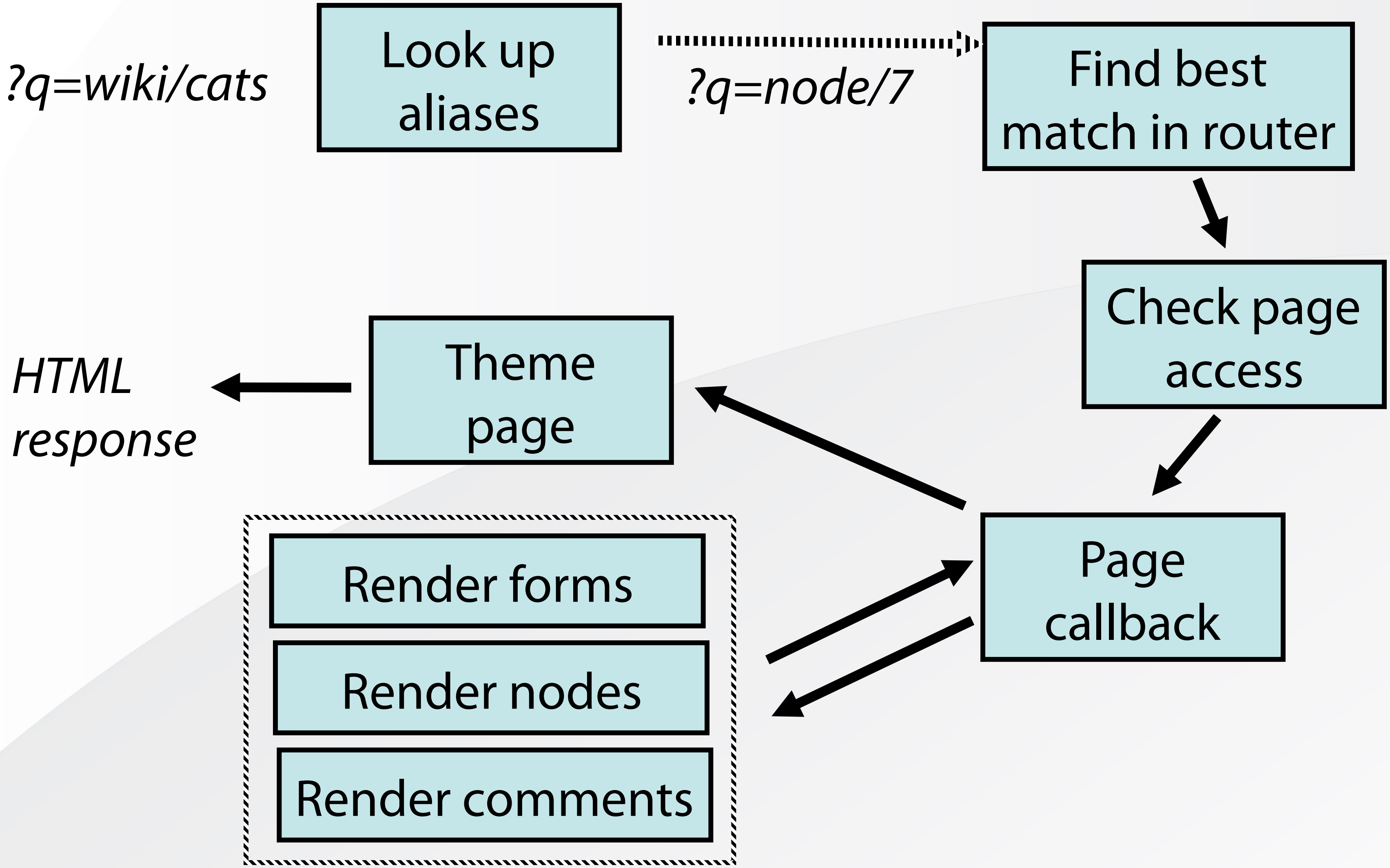
Menu system architecture

- Drupal finds the page to serve based on the entries in {menu_router}. The router items are independent of links shown on the page.
- The incoming path is broken down into its possible matches based on its parts.
- Drupal only examine patterns that actually exist as router items (the masks).
- The single best match is loaded and determines Drupal's response.

admin/content/taxonomy/edit/vocabulary/1

- [0] => admin/content/taxonomy/edit/vocabulary/%
- [1] => admin/content/taxonomy/edit/%/1
- [2] => admin/content/taxonomy/%/vocabulary/1
- [3] => admin/content/taxonomy/edit/vocabulary
- [4] => admin/content/taxonomy/edit/%
- [5] => admin/content/taxonomy/%/vocabulary
- [6] => admin/content/%/%/%
- [7] => admin/%/taxonomy/%/vocabulary
- [8] => admin/content/taxonomy/edit
- [9] => admin/content/taxonomy/%
- [10] => admin/%/taxonomy/edit
- [11] => admin/content/taxonomy
- [12] => admin/content/%
- [13] => admin/%/taxonomy
- [14] => admin/content
- [15] => admin/%
- [16] => admin

From Path to Output, 6.x



?q=wiki/cats

Look up aliases

?q=node/7

Find best match in router

Check page access

Page callback

Render forms

Render nodes

Render comments

Theme page

HTML response

The Problem (per eaton)

- We build the node as a structured array, but collapse it to html -- the structural information is lost to themers and template designers.
- The rendering workflow is still not flexible -- hard-coded calls to comment_render() and ugly functions like node_show()
- Themers who want to print out small portions of a node, or 'reshuffle' the layout go back to square one.

The Problem (per eaton) cont’d

- In 5.x nodeapi had no way of knowing the intended use of the node, save the coarse 'teaser/full' boolean - partially fixed in 6.x with “build modes”
- See: drupal.org/node/134478 “Refactor page (and node) rendering”

The Problem (per nedjo)

- How can we fetch a page in JSON rather than rendered HTML format, to enable selective updating of specific page elements as opposed to full page refreshes.
- See: drupal.org/node/145551 “Enable loading and rendering into JSON, XML, etc.”

The Problem (per starbow)

- When a user has made changes (blocks admin) and then clicks a link such as "configure" they lose all of their work.
- For pop-ups and AHAH we need the main content of the page with little ornament.
- drupal.org/node/18830 “Popups in Drupal 7- Plugable renderers for generating content”
- drupal.org/node/193311”Ajax Popups in Drupal 7: Adding Modal Dialogs to Help, Confirmations and Filter tips”

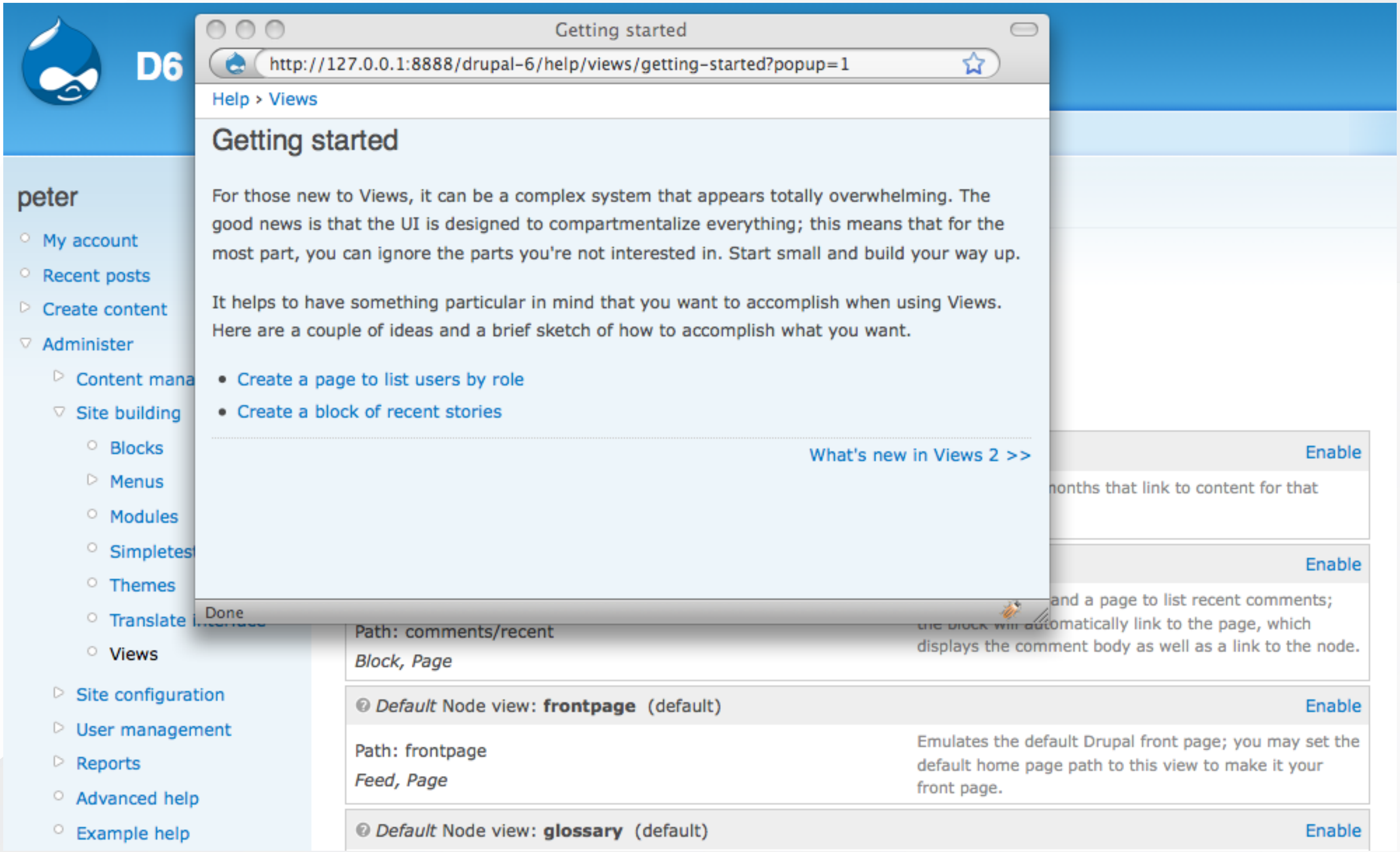
Rendering into HTML limits us

- Currently the page callback returns a concatentation of *already rendered* content.
- To re-arrange, print multiple copies, or alter we must use string parsing and/or re-render.
- We have no simple way to transform this into another format.
- We can use something like XSLT to map XHTML to a specific XML type, but…

D6 steps forwards core/contrib

- XSLT requires substantial server configuration, so not really a core solution.
- A number of contributors have modules for Drupal 6 (or Drupal 5) that demonstrate things we may want in core in D7.
- Note: this is a very incomplete list. The Q&A time at the end is where you can share more examples…

Popups - Advanced Help



Getting started

http://127.0.0.1:8888/drupal-6/help/views/getting-started?popup=1

Help › Views

Getting started

For those new to Views, it can be a complex system that appears totally overwhelming. The good news is that the UI is designed to compartmentalize everything; this means that for the most part, you can ignore the parts you're not interested in. Start small and build your way up.

It helps to have something particular in mind that you want to accomplish when using Views. Here are a couple of ideas and a brief sketch of how to accomplish what you want.

- Create a page to list users by role
- Create a block of recent stories

What's new in Views 2 >>

Done

Enable

nonths that link to content for that

Enable

and a page to list recent comments; the block will automatically link to the page, which displays the comment body as well as a link to the node.

Path: comments/recent

Block, Page

Default Node view: **frontpage** (default) Enable

Path: frontpage

Feed, Page

Emulates the default Drupal front page; you may set the default home page path to this view to make it your front page.

Default Node view: **glossary** (default) Enable

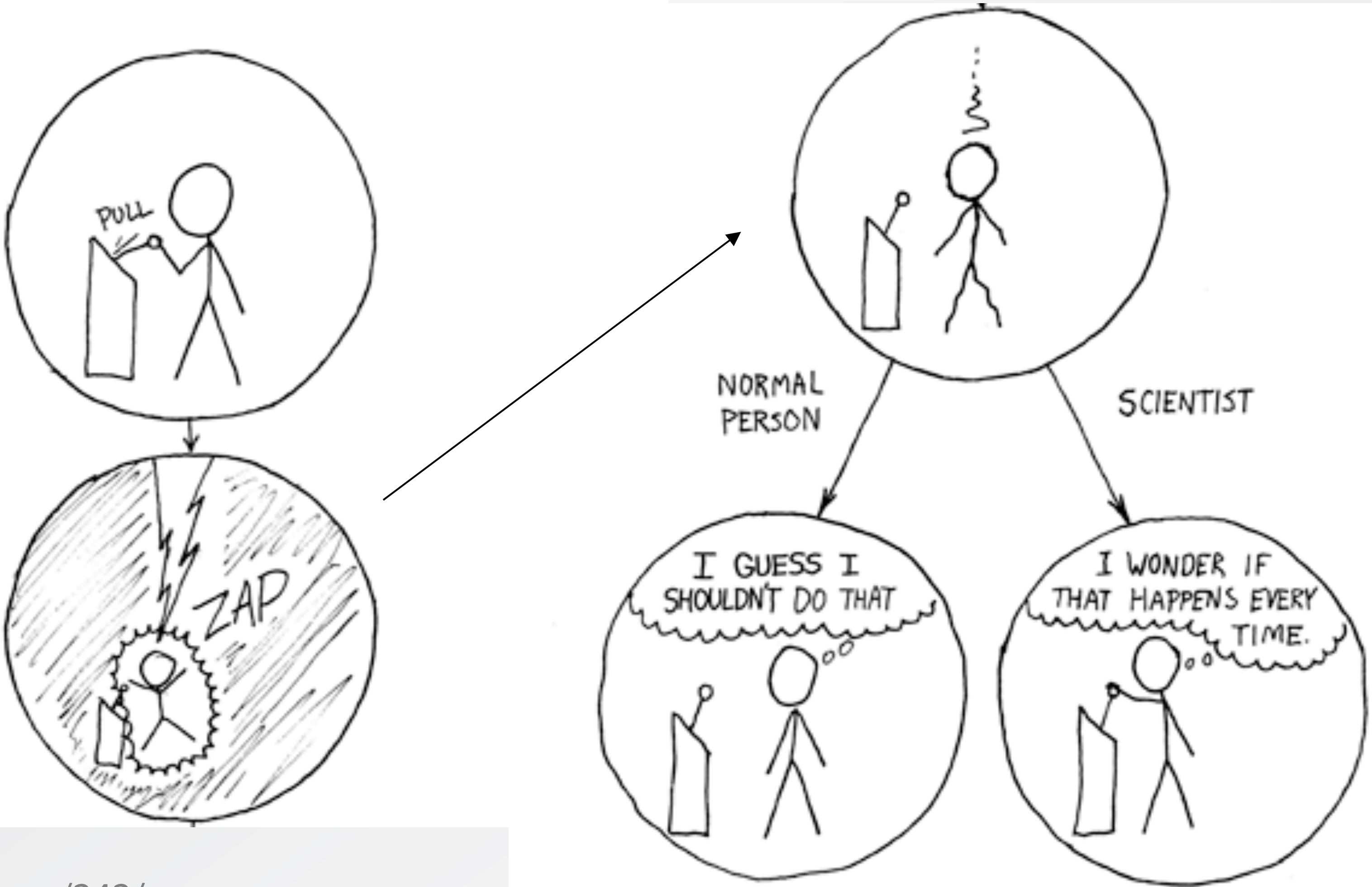
JSON

- Form API uses AHAH (JSON) to reload page elements.
- drupal.org/node/174008 shows how to substitute a JSON template file based on the headers from jQuery: 'XmlHttpRequest' == $_SERVER ['HTTP_X_REQUESTED_WITH']
- Looking ahead: we can switch rendering based on header or query strings.

Exporting Views

- Views output to XML/RDF/JSON/XHTML drupal.org/project/views_datasource groups.drupal.org/node/11387
- The project consists of 4 Views style plugins (and related row plugins) - i.e. we need different theme() functions (mostly template files here).

What could we do in D7 core?

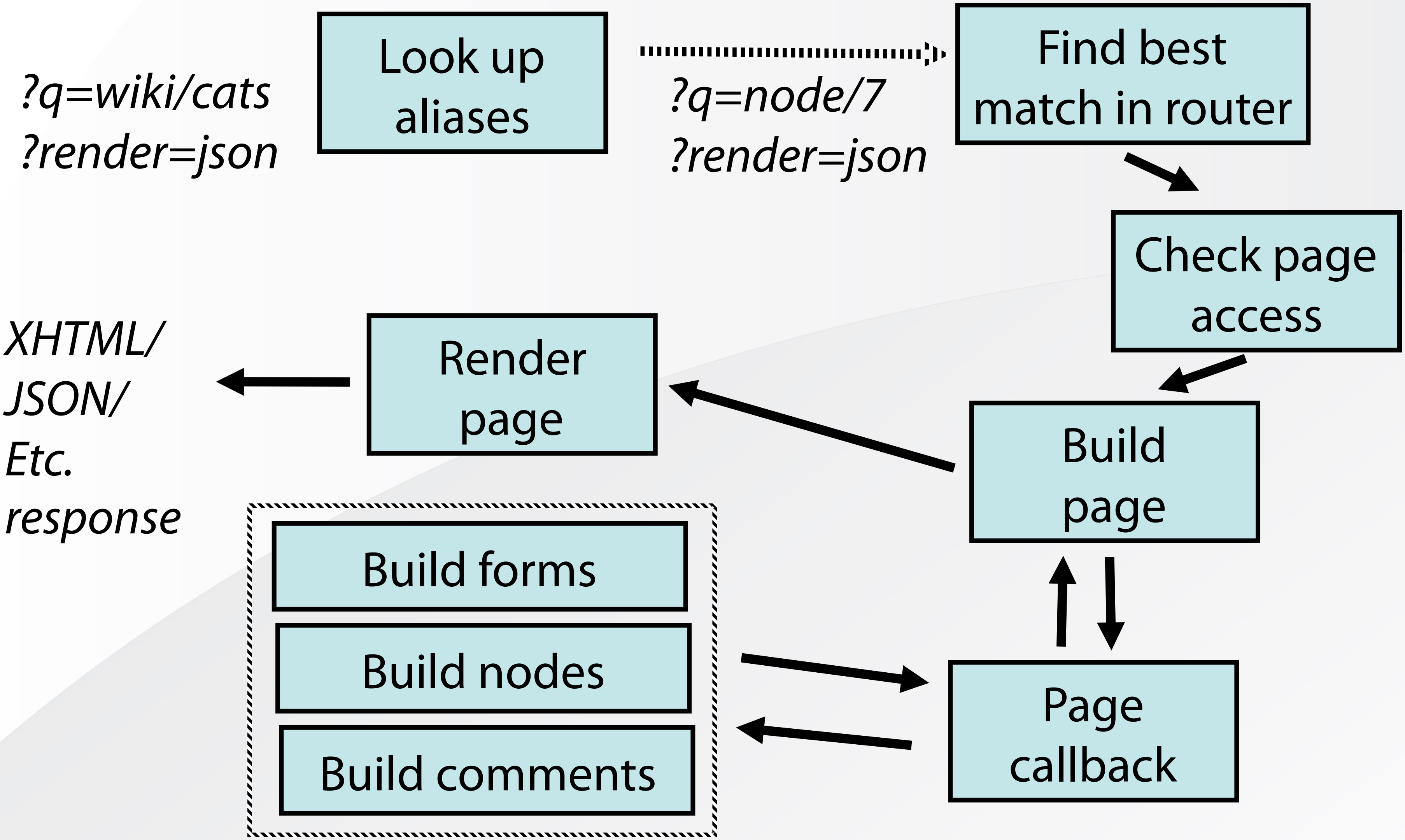

PULL
ZAP
NORMAL PERSON
SCIENTIST
I GUESS I SHOULDN'T DO THAT
I WONDER IF THAT HAPPENS EVERY TIME.

http://xkcd.com/242/

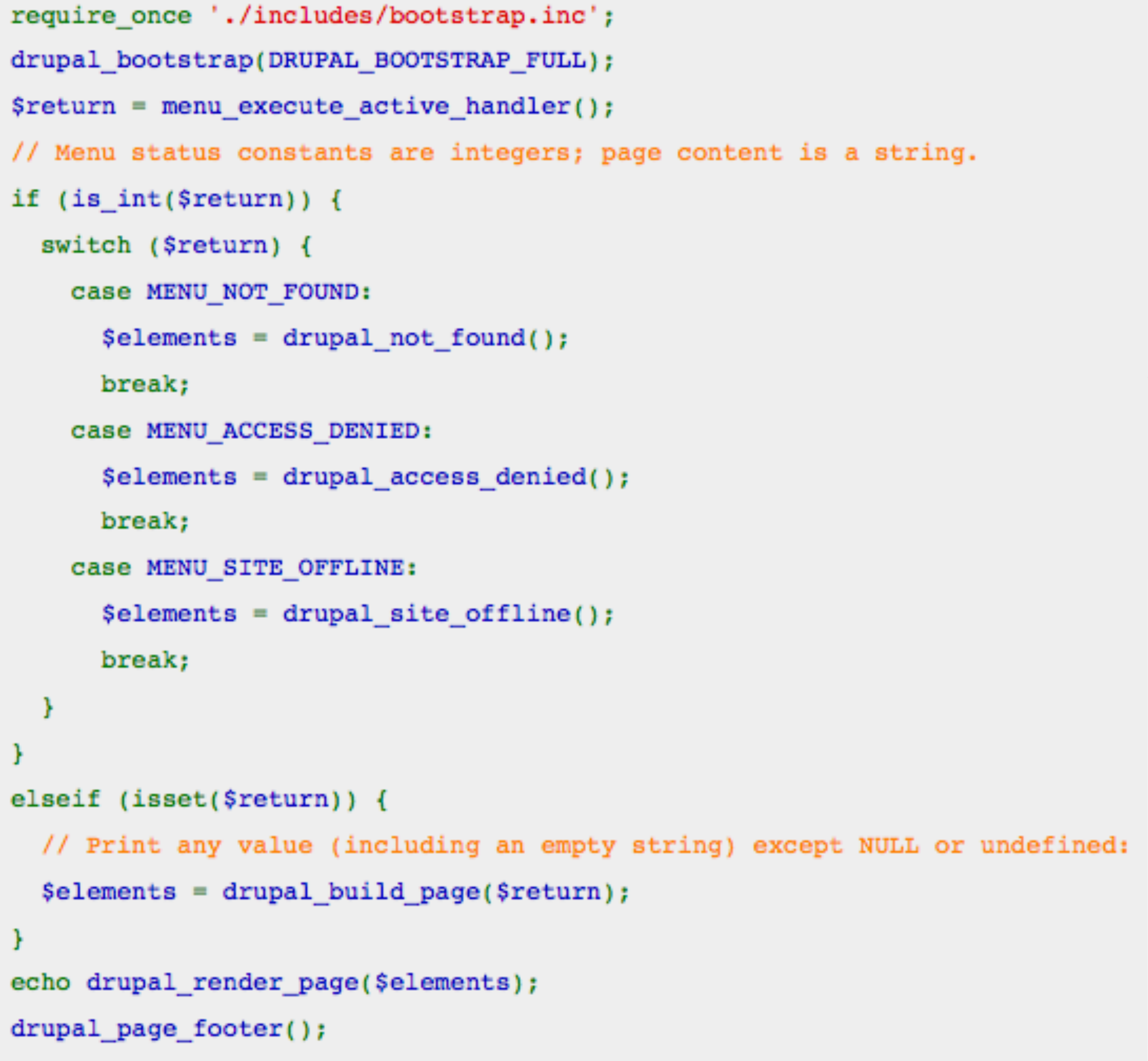
Structured data instead of HTML

- Drupal already generates much of it’s content as structured data, especially node content and forms.
- This data is rendered into a HTML string before the page callback returns it to index.php.
- We could leverage the structure we already create returning the unrendered data to index.php.

From Path to Output, 7.x ?



Index.php changes just a little



```
require_once './includes/bootstrap.inc';
drupal_bootstrap(DRUPAL_BOOTSTRAP_FULL);
$return = menu_execute_active_handler();
// Menu status constants are integers; page content is a string.
if (is_int($return)) {
  switch ($return) {
    case MENU_NOT_FOUND:
      $elements = drupal_not_found();
      break;
    case MENU_ACCESS_DENIED:
      $elements = drupal_access_denied();
      break;
    case MENU_SITE_OFFLINE:
      $elements = drupal_site_offline();
      break;
  }
}
elseif (isset($return)) {
  // Print any value (including an empty string) except NULL or undefined:
  $elements = drupal_build_page($return);
}
echo drupal_render_page($elements);
drupal_page_footer();
```

Parallel theme() systems?

- Having structured data does not magically let us turn it into meaningful XML, etc.
- We have a large number of theme_ functions and templates crafted to turn data into XHTML
- At least a partial subset would need to be supplied for each desired output format.
- Some JSON applications may be easier - we can partially render the data while retaining structure.

Road map phase 1

- Pass back structured data to index.php from all page handlers.
- Change drupal_get_form() and node content.
- Convert theme_table and theme_item_list to take a single structured element.
- Add a function to render child elements, but leave an array structure.
- Develop API/registry for renders and alternate theme functions.

Thank you (via drupal.org)

- Jeff Eaton
- Barry Jaspan
- Nedjo Rogers
- Tao Starbow
- Stefan Nagtegaal
- Allister Beharry
- Rob Roy
- Larry Garfield
- and many more…

Questions/to help

Peter Wolanin
Drupal/IRC: pwolanin
peter.wolanin@acquia.com

Update the latest patch, help convert more tables/lists, etc:

http://drupal.org/node/134478

Come to the code sprint Saturday!