Phase2

PHASE2TECHNOLOGY.COM

Drupal Is Not Your Web Site

Develop for high-scale, fragmented sites

DrupalCon Los Angeles 2015

Tobby Hagler
@thagler d.o: tobby

Phase2

Presumptions and Expectations

- Intermediate level session, but experienced with site building with Drupal and basic Devops
- Passing familiarity with the topics presented here, but not significant experience with them
- Will be an overview of concepts; an introduction designed to lead you to thinking about your site architecture
- Suggested modules and resources for implementation

Overview

- What a Web site really is
- The Origin and the Edge
- How does Drupal fit into your complete web sites's stack?
- Other elements of a Web site to consider
- Removing things from Drupal to help overall performance
- Real-time and custom content updates on your Web site

A Brief Analogy

The Mighty Mississippi

Where does the Mississippi River begin?

Thousands of smaller estuaries converge to give the Mississippi a start

The mouth of the Mississippi looks nothing like the origin

Content Inclusion

From Origin to final output, many other sources get included at different levels of the stack

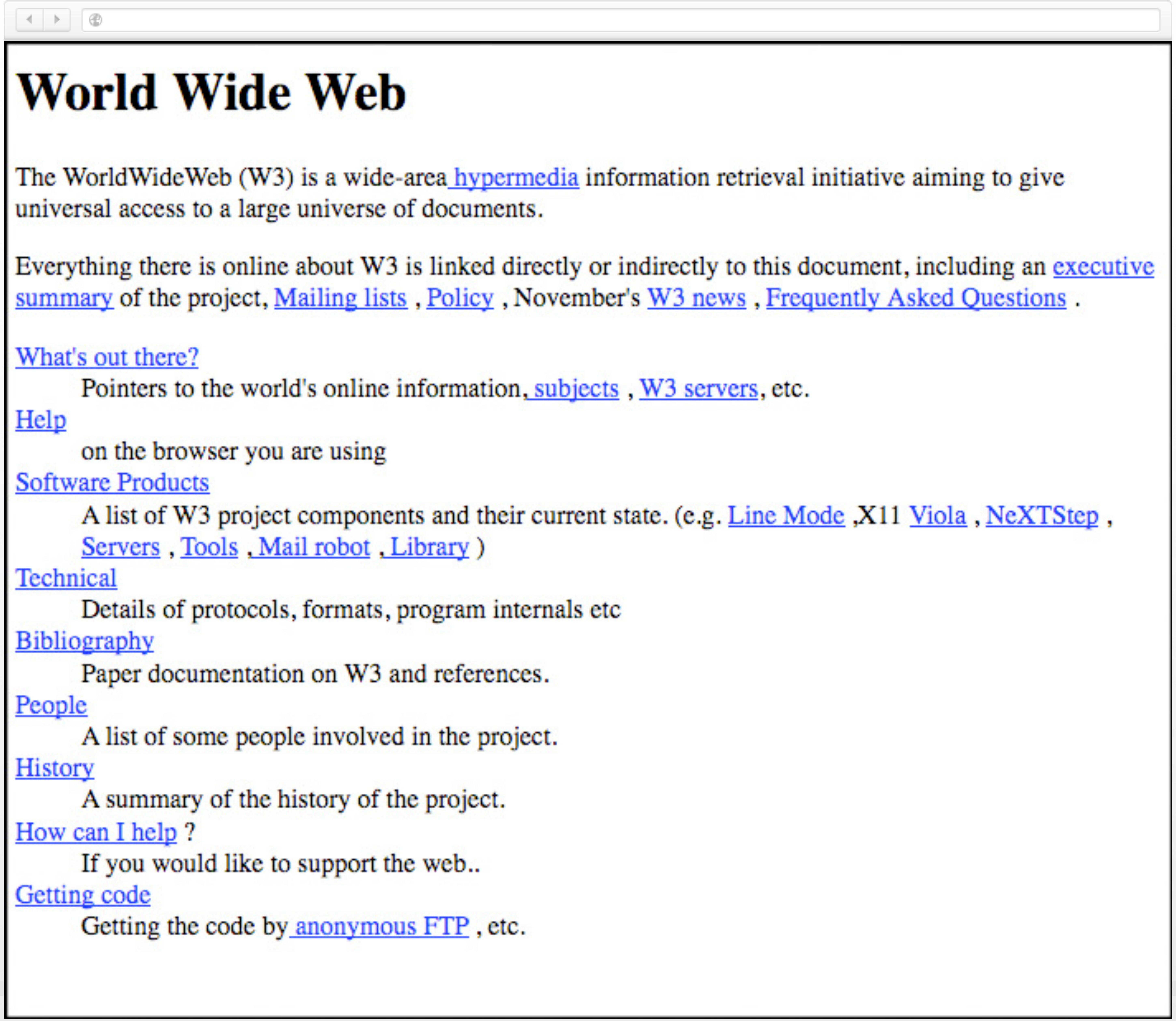
Origin isn’t aware of other sources or how the final output will appear

The Web Site

> *A location connected to the Internet that maintains one or more pages on the World Wide Web.*

Merriam-Webster

http://info.cern.ch/hypertext/WWW/TheProject.html



World Wide Web

The WorldWideWeb (W3) is a wide-area hypermedia information retrieval initiative aiming to give universal access to a large universe of documents.

Everything there is online about W3 is linked directly or indirectly to this document, including an executive summary of the project, Mailing lists , Policy , November's W3 news , Frequently Asked Questions .

What's out there?

Pointers to the world's online information, subjects , W3 servers, etc.

Help

on the browser you are using

Software Products

A list of W3 project components and their current state. (e.g. Line Mode ,X11 Viola , NeXTStep , Servers , Tools , Mail robot , Library)

Technical

Details of protocols, formats, program internals etc

Bibliography

Paper documentation on W3 and references.

People

A list of some people involved in the project.

History

A summary of the history of the project.

How can I help ?

If you would like to support the web..

Getting code

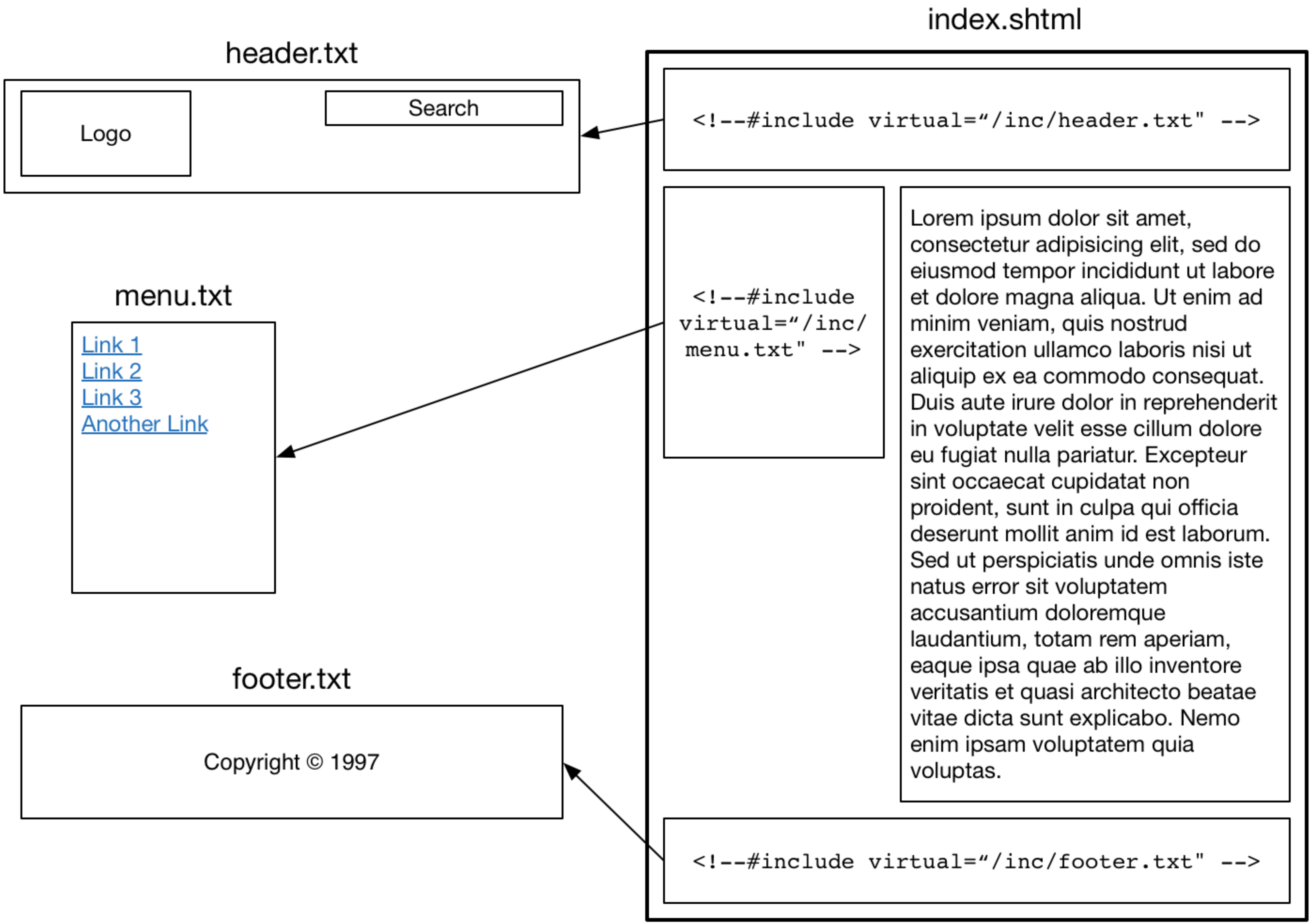
Getting the code by anonymous FTP , etc.

Common Page Assembly Methods

- Static (flat) HTML files
- Server-side Includes (SSI)
- CGI, Scripting and template languages
- Content Management Systems (CMS)
- Edge-side Includes (ESI)
- Client-side assembly (Big Pipe)

Server-Side Includes (SSI)

- HTTP/Web server parses (.shtml) files for commands
- <!--#include file="/path/to/file.txt" -->
- <!--#include virtual="menu.cgi" —>
- <!--#if expr="${SomeBoolean}" --> … <!--#endif -->



header.txt

Logo

Search

menu.txt

Link 1
Link 2
Link 3
Another Link

footer.txt

Copyright © 1997

index.shtml

```
<!--#include virtual=“/inc/header.txt" -->
```

```
<!--#include
virtual=“/inc/
menu.txt" -->
```

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas.

```
<!--#include virtual=“/inc/footer.txt" -->
```

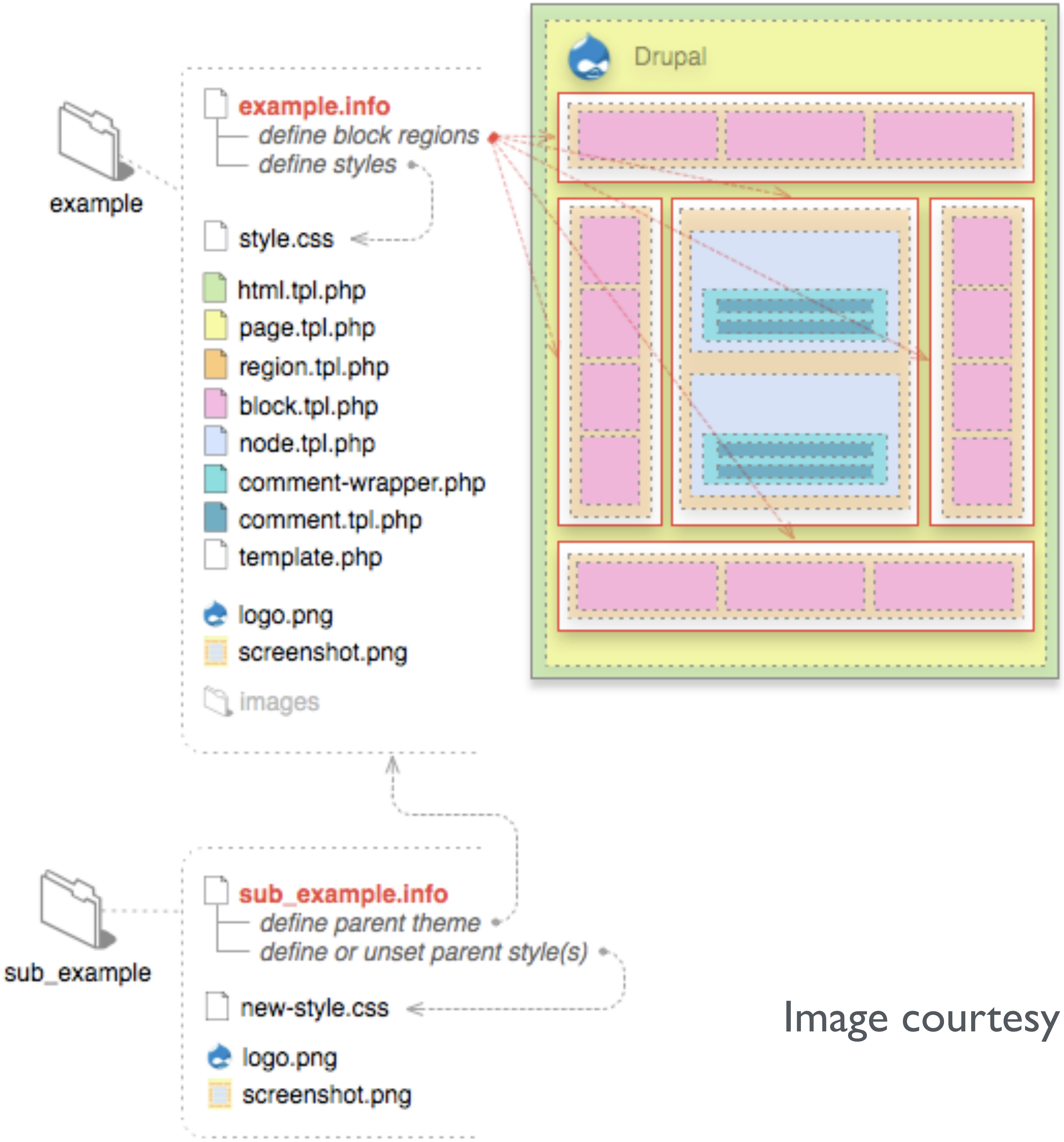


Image courtesy drupal.org

Drupal Theming & Page Assembly

- User makes request for a Web page “/node/123”
- Drupal bootstrap process begins, callback function executed, hooks invoked, etc.
- Content is dynamically rendered to HTML via theme functions *
- HTML document is delivered to the user

* Ignore cache and other steps for now

The Web browser doesn’t care how the page is ultimately generated. In the end, it will receive HTML that it will parse, and make requests for additional elements (images, CSS files, JavaScript files, etc.) and deliver the final presentation to the end user.

The Web browser doesn’t care how the page is ultimately generated. In the end, it will receive HTML that it will parse, and make requests for additional elements (images, CSS files, JavaScript files, etc.) and deliver the final presentation to the end user.

What the browser displays may be different than the HTML it receives. JavaScript may hide content or make additional requests for more things to display (e.g., Single-page apps, Big Pipe)

Why is Drupal not my Web site?

A Typical Drupal Implementation

- Nodes contain articles, stories, blog posts, etc.
- Blocks provide ancillary content that can be repeated
- Views lists pages of related or similar content
- Panels allows the arrangement of content pieces
- Theme provides design and user experience

So Drupal *is* my Web site?

With this simple implementation, what Drupal generates and the HTML documents the user receives are essentially the same.

2 + 2 = 4 and 2 + x* = 4 are also essentially the same.

* Where x = 2

Performance Considerations

Document delivery costs

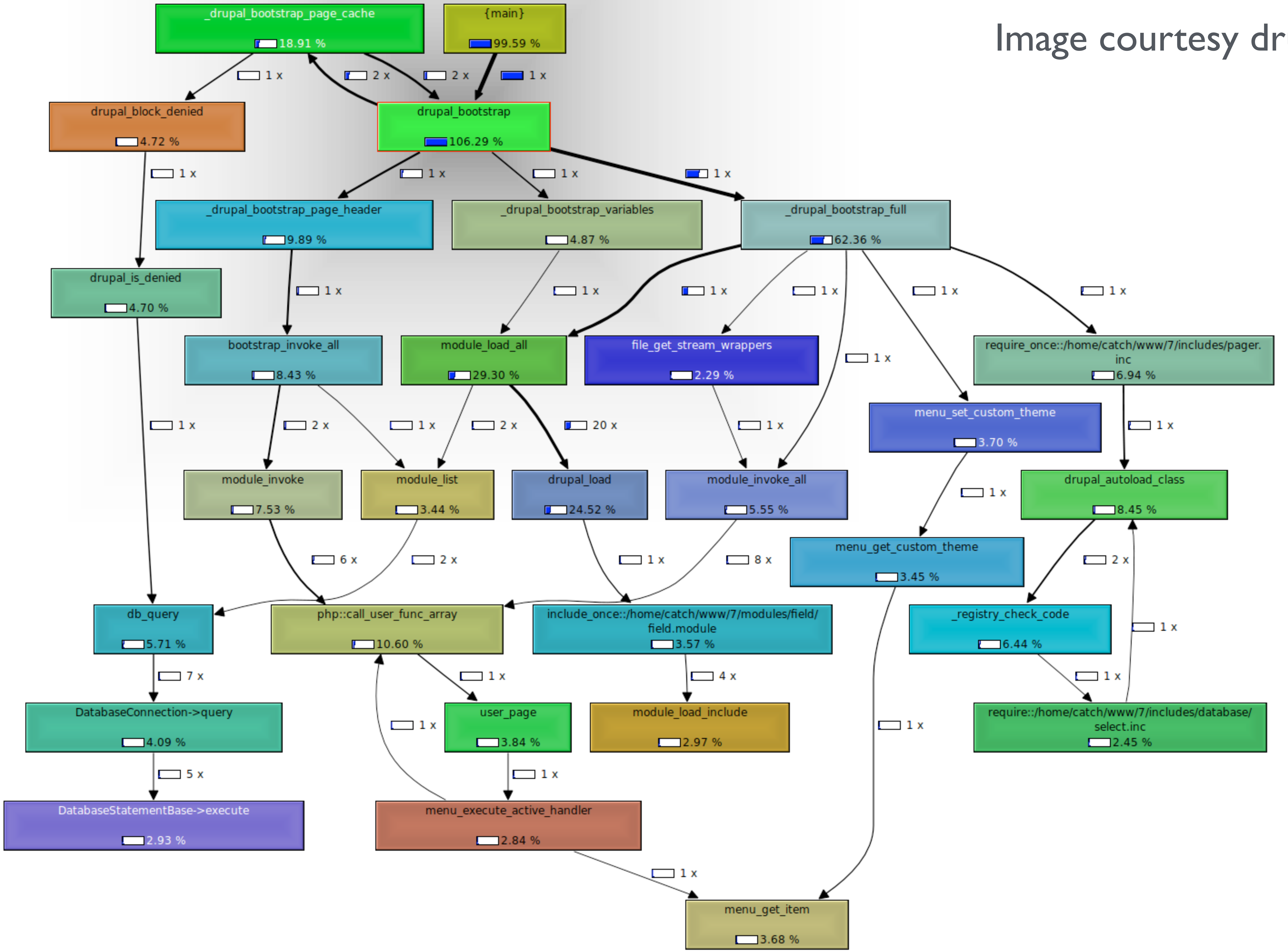
- Static HTML costs (practically) nothing, file I/O; difficult to maintain compared with a CMS
- SSI is largely file I/O and somewhat CPU intensive
- PHP code execution is memory intensive and CPU bound
- MySQL uses CPU, memory, and I/O heavily
- Resources limit the number of simultaneous users (traffic)

PHP execution is costly

Having the Web server invoke PHP, and having PHP retrieve content and render a full HTML document on every page request is resource intensive.

Drupal employs a number of caching systems and optimization to reduce server stress.

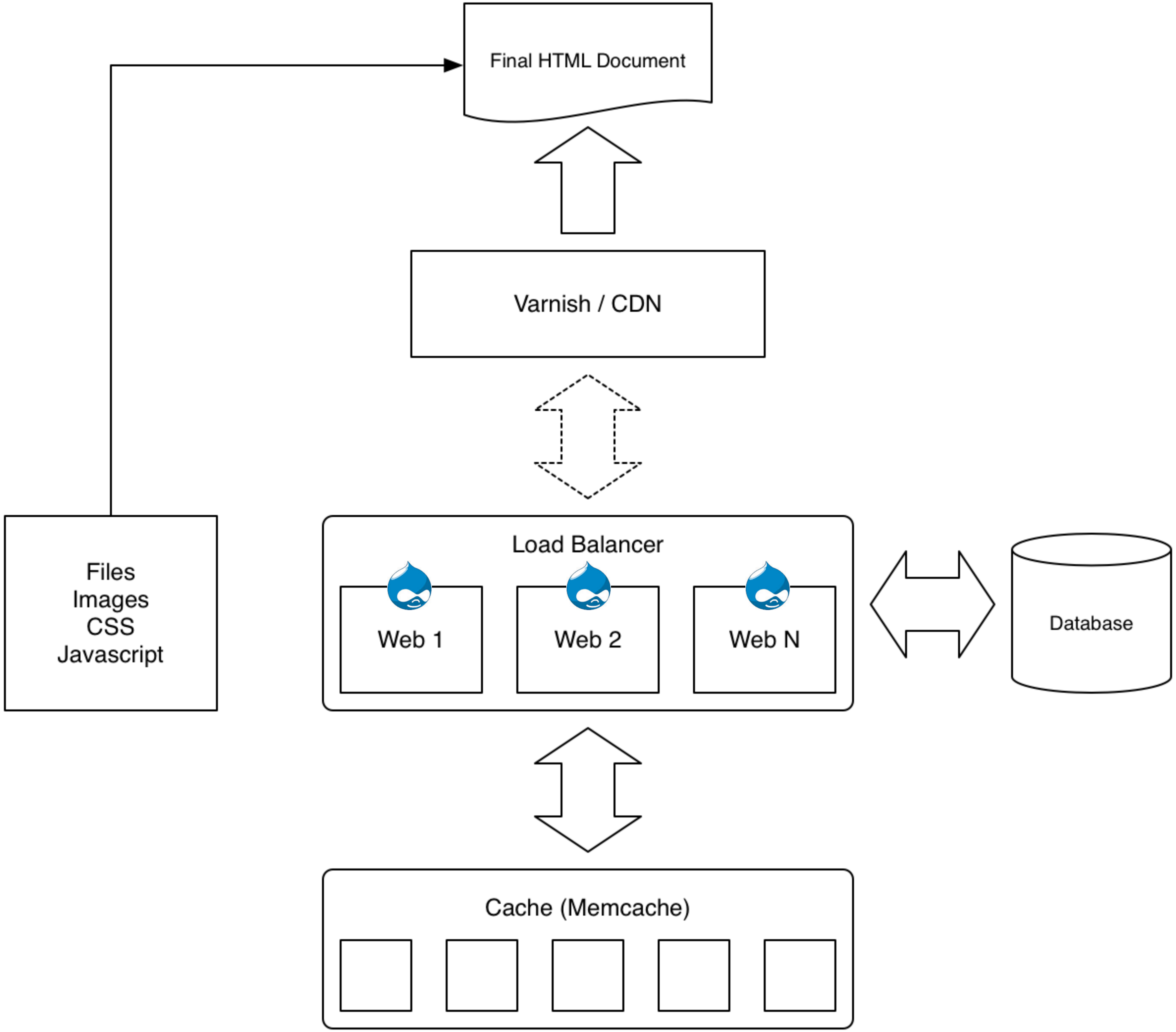
Image courtesy drupal.org



Scaling Drupal

To be able to handle traffic, introduce multiple Web servers that all serve Drupal’s content.

Employ various caching schemes in front and behind Drupal.


Final HTML Document
Varnish / CDN
Files
Images
CSS
Javascript
Load Balancer
Web 1
Web 2
Web N
Database
Cache (Memcache)

Drupal is the back-end

When behind Varnish or a CDN, the (unauthenticated) Web user does not interact with Drupal directly*. The request is handled by delivering content from cache, returning the same HTML document sent to any other user requesting the same path. Drupal does not and *cannot* return something different since Drupal never receives the HTTP request.

* Assuming cache is already primed

Fragmentation

Drupal and Fragmentation

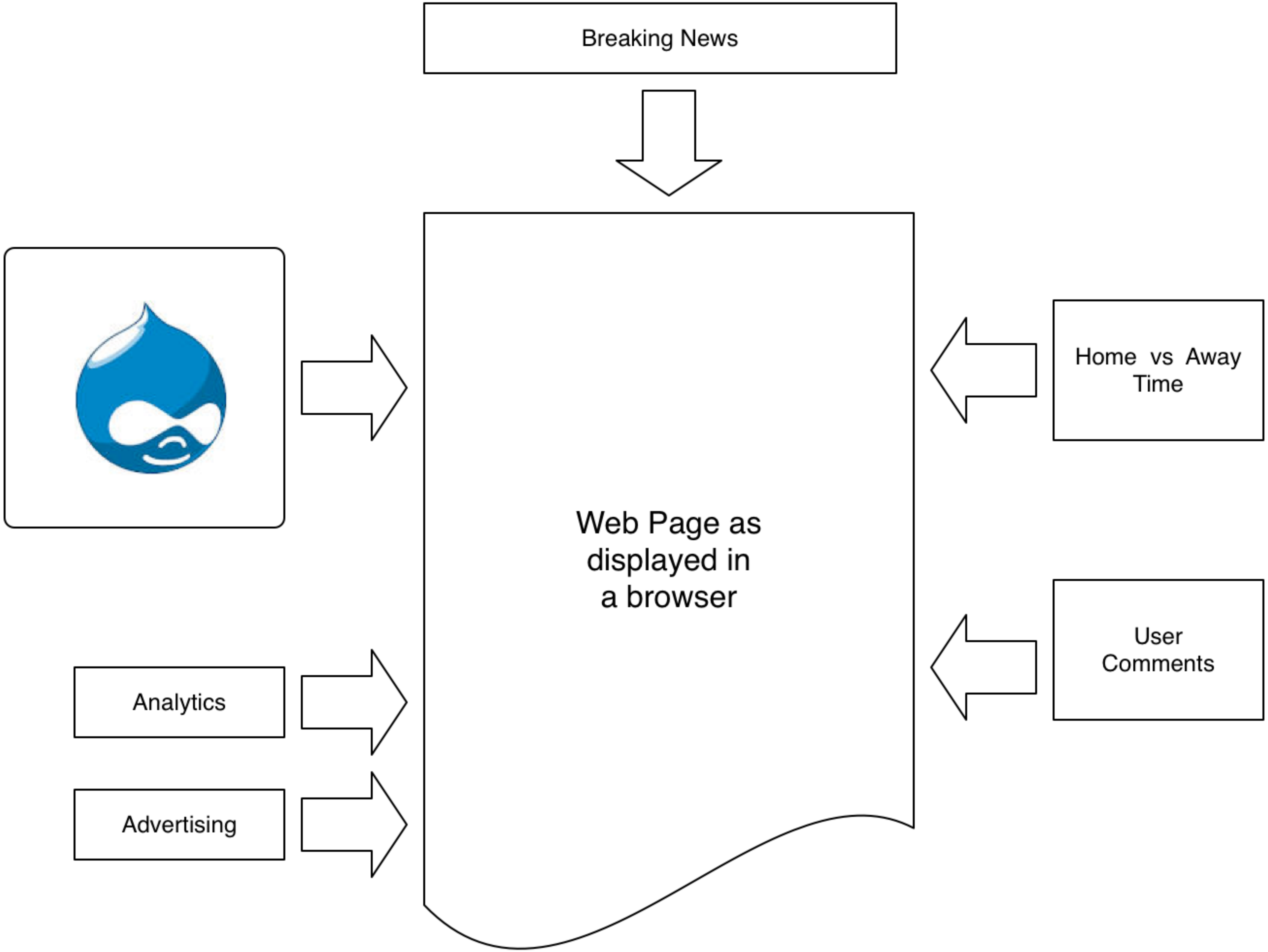
Drupal is already doing enough. It doesn’t need to do additional processing to fetch content and ingest it. Offload some of the work to other services. It’s OK to decentralize.

Use 3rd-party services because that’s where the content originates.

Sometimes Drupal isn’t the best thing to handle all of your content

Assembling Fragments

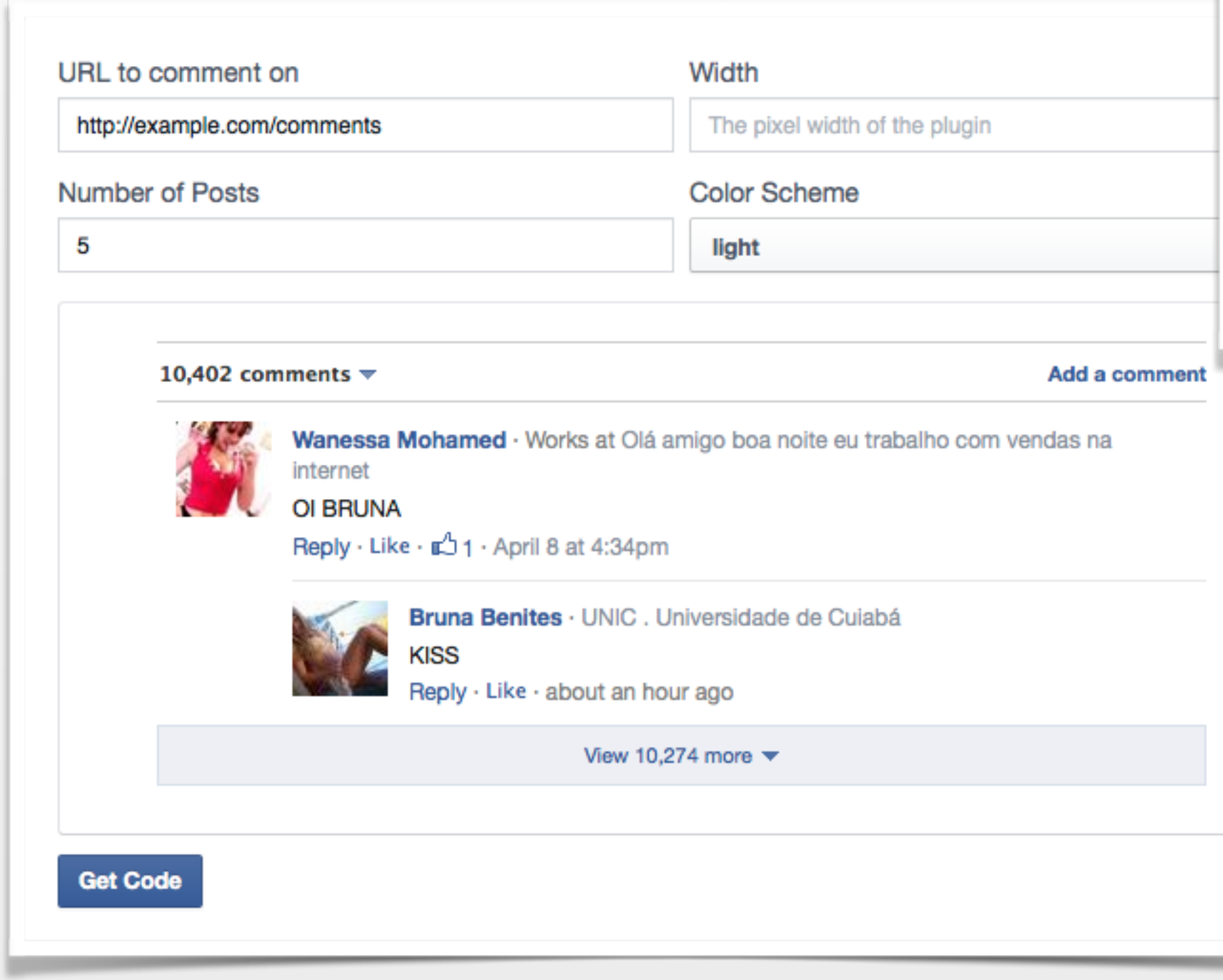
- Client-side Javascript
  - AJAX
  - WebSockets
- Varnish/CDN Edge-Side Includes
  - Include content into a Varnish-cached page


Breaking News
Home vs Away Time
User Comments
Web Page as displayed in a browser
Analytics
Advertising

Typical 3rd-Party Content

- User comments (Facebook, Disqus, LiveFyre)
- Twitter
- Real-time scores or standings
- Web pages hosted on legacy systems
- Breaking news alerts
- Analytics

Externally Hosted Comments



URL to comment on

http://example.com/comments

Width

The pixel width of the plugin

Number of Posts

5

Color Scheme

light

10,402 comments ▾ Add a comment

Wanessa Mohamed · Works at Olá amigo boa noite eu trabalho com vendas na internet

OI BRUNA

Reply · Like · 👍1 · April 8 at 4:34pm

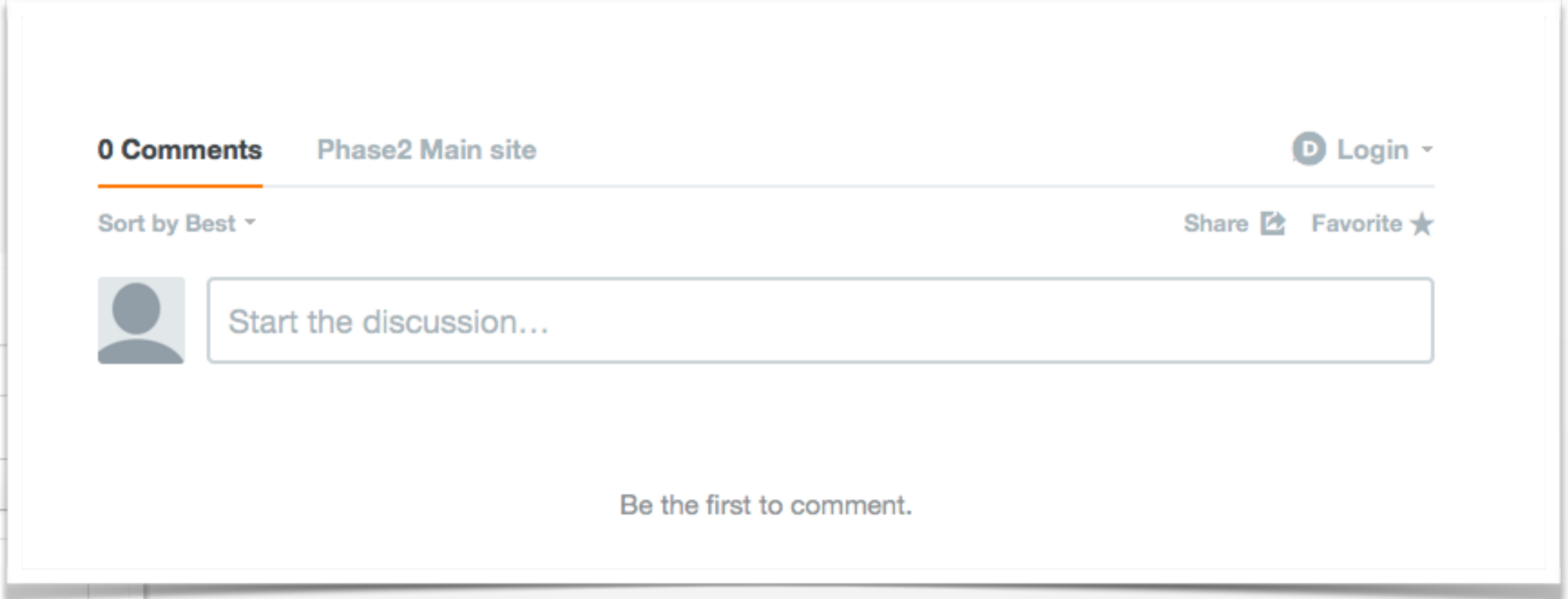
Bruna Benites · UNIC . Universidade de Cuiabá

KISS

Reply · Like · about an hour ago

View 10,274 more ▾

Get Code



0 Comments Phase2 Main site Login ▾

Sort by Best ▾ Share Favorite ★

Start the discussion…

Be the first to comment.

Externally Hosted Comments

Most commenting systems provide code (HTML, JavaScript, etc.) that can be placed on the page.

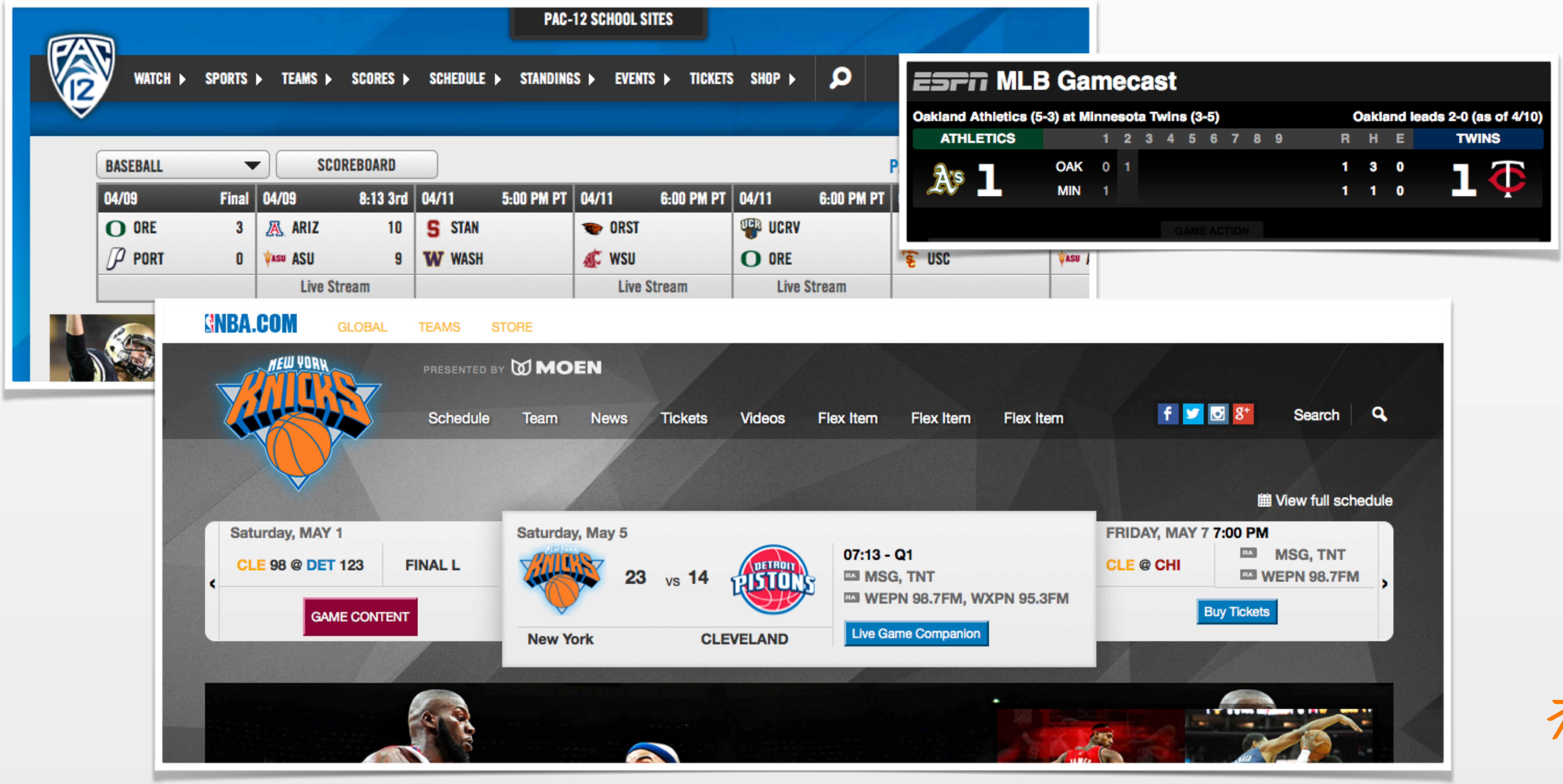
Simply adding code and simple references to the parent page are all that is needed. Drupal is otherwise unaware of that content.

Client-side Assembly

Implementing Real-Time Updates

- WebSocket specification
  - WebSocket JavaScript Interface
- Socket.IO — Node.js project
  - JavaScript framework and WebSocket API with long polling
- Jetty — Java servlets and HTTP server

Live Scoring & Real-Time Updates



PAC-12 SCHOOL SITES

04/09 Final	04/09 8:13 3rd	04/11 5:00 PM PT	04/11 6:00 PM PT	04/11 6:00 PM PT
ORE 3	ARIZ 10	STAN	ORST	UCRV
PORT 0	ASU 9	WASH	WSU	ORE
	Live Stream		Live Stream	Live Stream

USC

ESPN MLB Gamecast

Oakland Athletics (5-3) at Minnesota Twins (3-5)

Oakland leads 2-0 (as of 4/10)

ATHLETICS		1	2	3	4	5	6	7	8	9	R	H	E	TWINS
1	OAK	0	1								1	3	0	1
	MIN	1									1	1	0	

Saturday, MAY 1
CLE 98 @ DET 123
FINAL L
GAME CONTENT

Saturday, May 5
23 vs 14
New York
CLEVELAND
07:13 - Q1
MSG, TNT
WEPN 98.7FM, WXPN 95.3FM
Live Game Companion

FRIDAY, MAY 7 7:00 PM
CLE @ CHI
MSG, TNT
WEPN 98.7FM
Buy Tickets

Socket.io

```
<script src="/socket.io/socket.io.js"></script>
<script>
  var socket = io.connect('http://localhost');
  socket.on('scores', function (data) {
    console.log(data);
    socket.emit('score event', { my: 'data' });
  });
</script>
```

Live Scoring & Real-Time Updates

- The node template places a placeholder with a token for the score block
- Browser makes asynchronous HTTP request to the scoring server, establishing a web socket between server and client
- Score changes are pushed to browser without polling or updating the page
- Drupal is unaware of the content of that placeholder

Edge-Side Includes

Varnish and many CDNs established the ESI specification.

Allows for fragments of content to reside elsewhere and be assembled on the Edge.

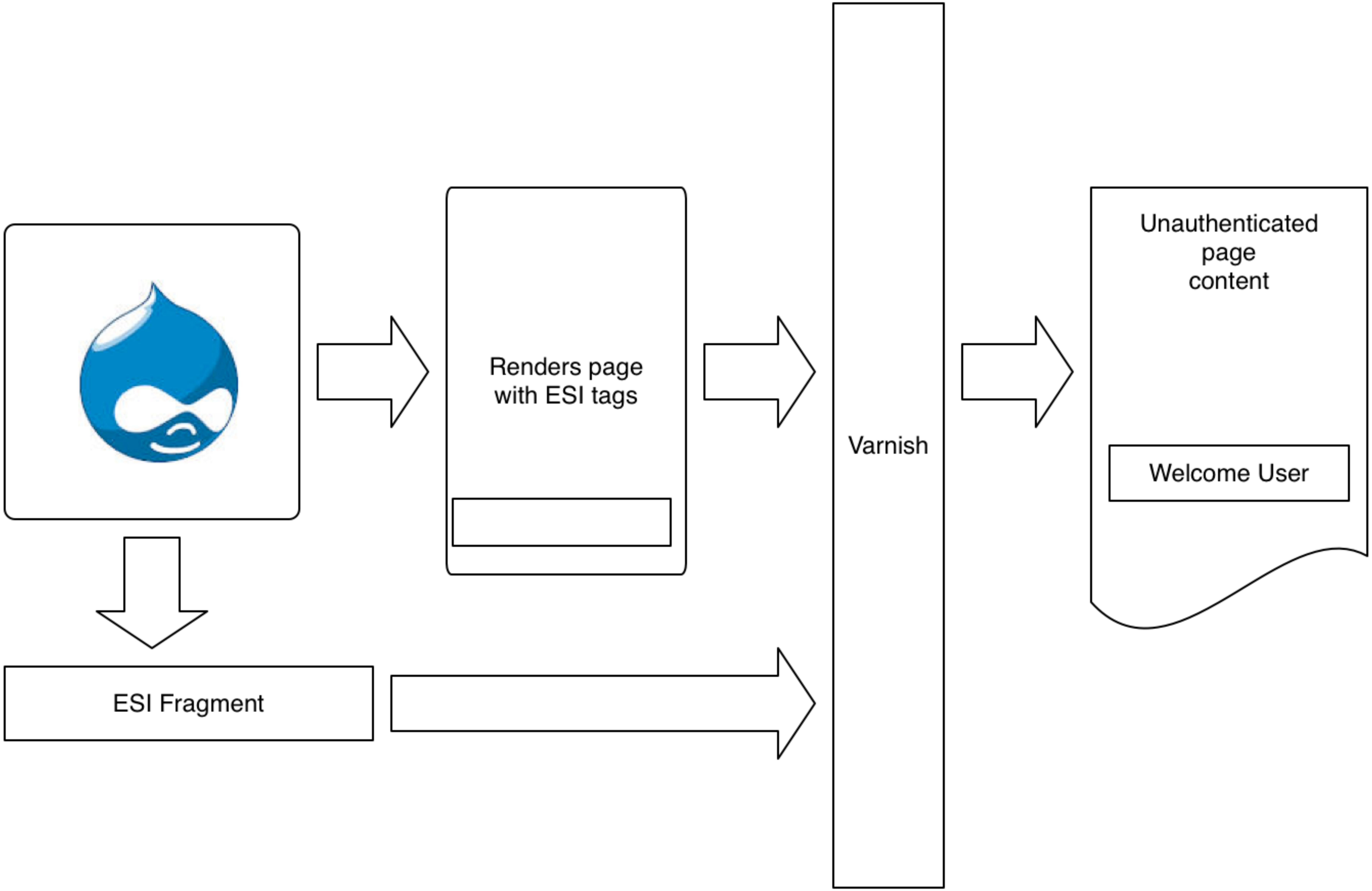
Allows for mixing of unauthenticated (cached) content with authenticated elements (i.e., “Welcome user”).

ESI Fragments

- Register a menu callback in Drupal to generate content with an abbreviated callback
- /esi/user/welcome/f1667cc8f3f9f55a
- <esi:include src=“/path/to/callback”/>
- <esi:include src="http://example.com/some/path"/>

Authenticated/Unauthenticated

- Authenticated traffic bypasses much of the caching layer
- Use ESI to provide customized service on an otherwise unauthenticated, cached page
- An unauthenticated page is heavily cached; ESI provides the customization users are used to seeing


Renders page with ESI tags
Varnish
Unauthenticated page content
Welcome User
ESI Fragment

Drupal Modules and Resources

- drupal.org/project/cdn
- drupal.org/project/esi
- socket.io
- www.varnish-cache.org/docs/3.0/tutorial/esi.html
- www.akamai.com/html/support/esi.html

Questions?

Tobby Hagler

Software Architect

Email: thagler@phase2technology.com

Twitter: @thagler

Drupal.org: tobby

Phase2

PHASE2TECHNOLOGY.COM