Web Development: You're Doing it Wrong

Stefan Tilkov, innoQ

QCon London 2013

Session # 9770

#qconlondon

@stilkov

Annoying your users

Forbid the use of the back and forward buttons

Send them to the home page when they hit “refresh”

… or ensure the browser pops up a warning window

Make sure they can’t open a second browser window

Let them see UI chrome and ads first, content last

Make sure they can't bookmark or send a link

Don’t let Google index anything

Ensure disabling JavaScript gives them a blank page

Things that seem like a good idea, but aren’t

Fix HTTP’s basic flaw: its statelessness

Go beyond the page model because we’re r/w now

Avoid CSS because who understands how it works?

Avoid HTML because that's so 20th century

Avoid JavaScript because it’s not for real programmers

Extend the client/server model with server push because polling? don’t even …

Let’s take a step back

Let’s build a generic client runtime

Define a protocol so it can work with any server

Allow it to mix services from different backends

Define a generic, declarative data format

Separate content and layout

Allow for extensibility with client-side scripting

Make it work on any device, with any resolution

We’re not done yet

Standardize it, with (rough) consensus

Ensure there are multiple client, server and intermediary implementations to choose from

Have every client OS ship with an implementation of the generic client

Let every programming environment support it

How hard can it be?

Competing with the Web may not be the best idea

Web-centric web UIs

Server-side components	Avoid HTML, JS, CSS Trade Familiarity for Complexity Session-centric
ROCA	Server-side POSH Client-side components Web-centric
Single Page Apps	Advanced Client Frameworks Server-side REST APIs

ROCA

(Resource-oriented Client Architecture)

Make your HTML semantic

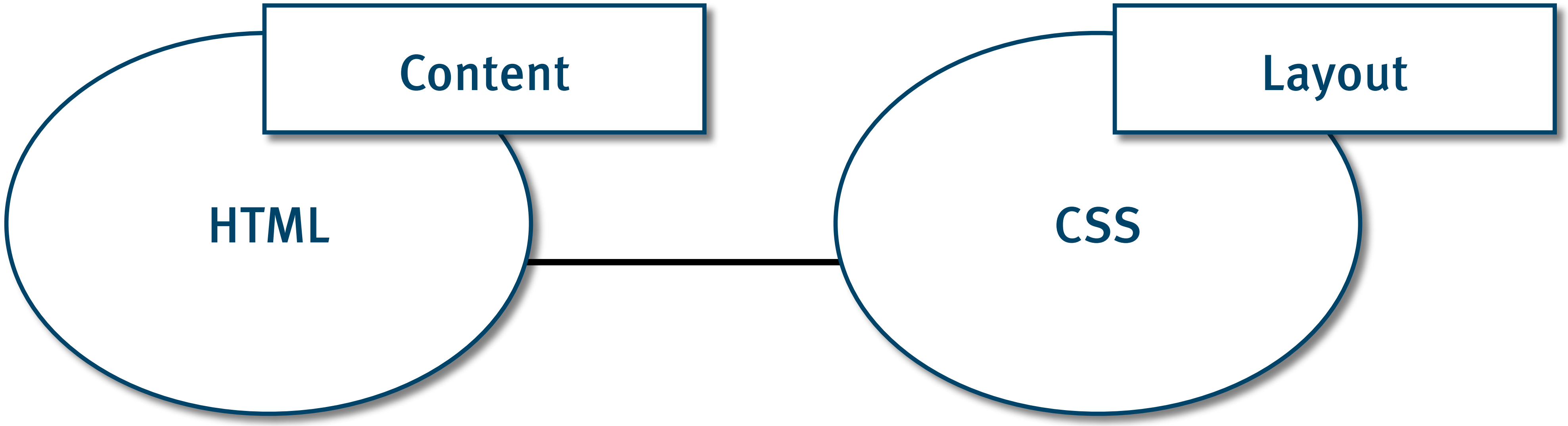
Minimal load times

API usage

Accessibility

Self-descriptiveness

Readability


Content
HTML
Layout
CSS

schema.org

```
<div>
  <h1>Avatar</h1>
  <span>Director: <span>James Cameron</span>
    (born August 16, 1954)</span>
  <span>Science fiction</span>
  <a href="../movies/avatar-theatrical-trailer.html"</a>

</div>
```

schema.org

```
<div itemscope itemtype="http://schema.org/Movie">
  <h1>Avatar</h1>
  <span>Director: <span>James Cameron</span>
    (born August 16, 1954)</span>
  <span>Science fiction</span>
  <a href="../movies/avatar-theatrical-trailer.html"</a>

</div>
```

schema.org

```
<div itemscope itemtype="http://schema.org/Movie">
  <h1 itemprop="name">Avatar</h1>
  <span>Director: <span itemprop="director">James Cameron</span>
    (born August 16, 1954)</span>
  <span itemprop="genre">Science fiction</span>
  <a href="../movies/avatar-theatrical-trailer.html"
     itemprop="trailer">Trailer</a>
</div>
```

Use Javascript *unobtrusively*

```
<a href="javascript:doSomething();">
  Some Link
</a>
```

```
<a href="#"
   onclick="doSomething();">
  Some Link
</a>
```

```
<a href="/some-resource"
   onclick="doSomething(this.href);">
  Some Link
</a>
```

```
<a href="/some-resource"
   class="whatever">
  Some Link
</a>
```

```
<a href="/some-resource"
   class="whatever">
  Some Link
</a>
```

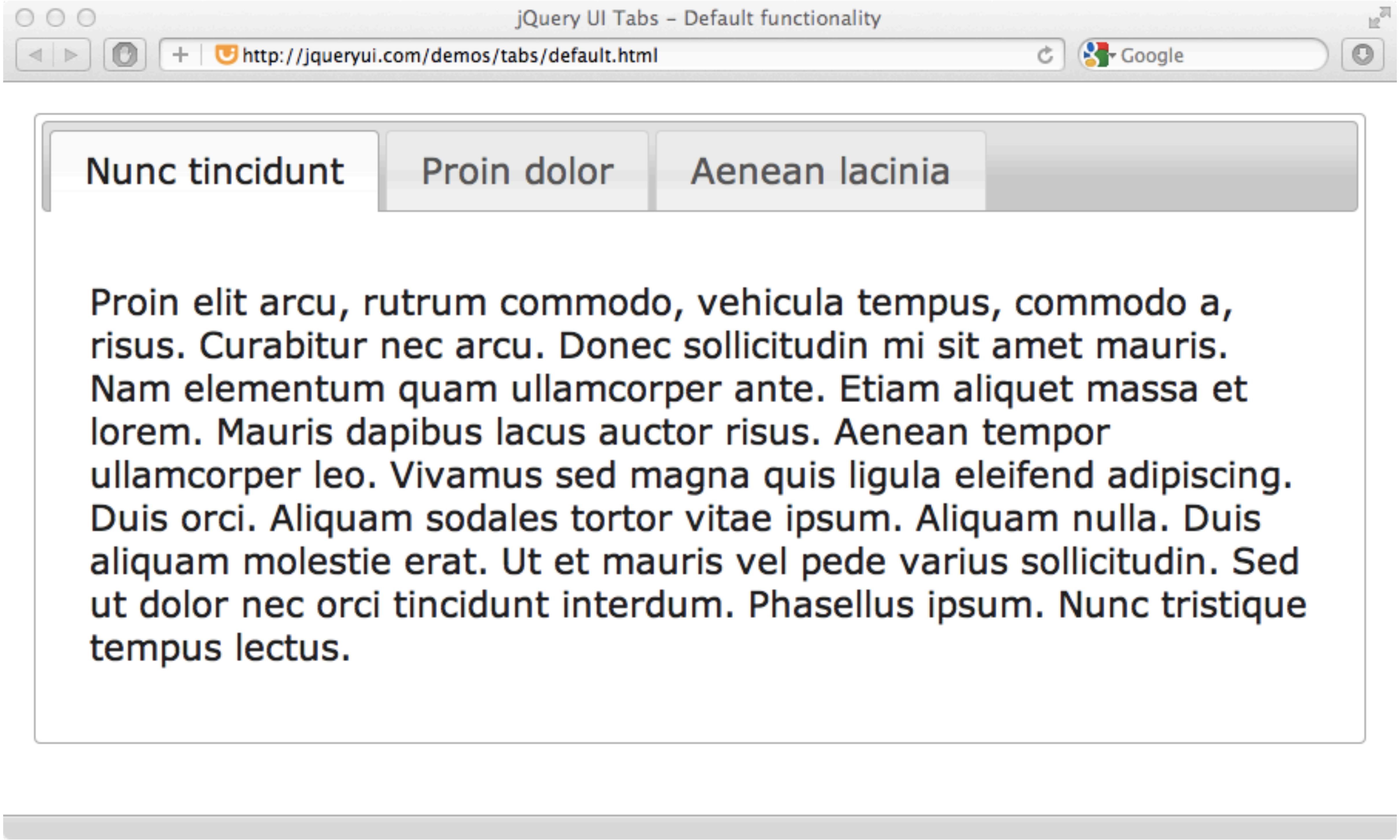
+

```
$("a.whatever").click(function() {
  doSomething(this.href);
});
```

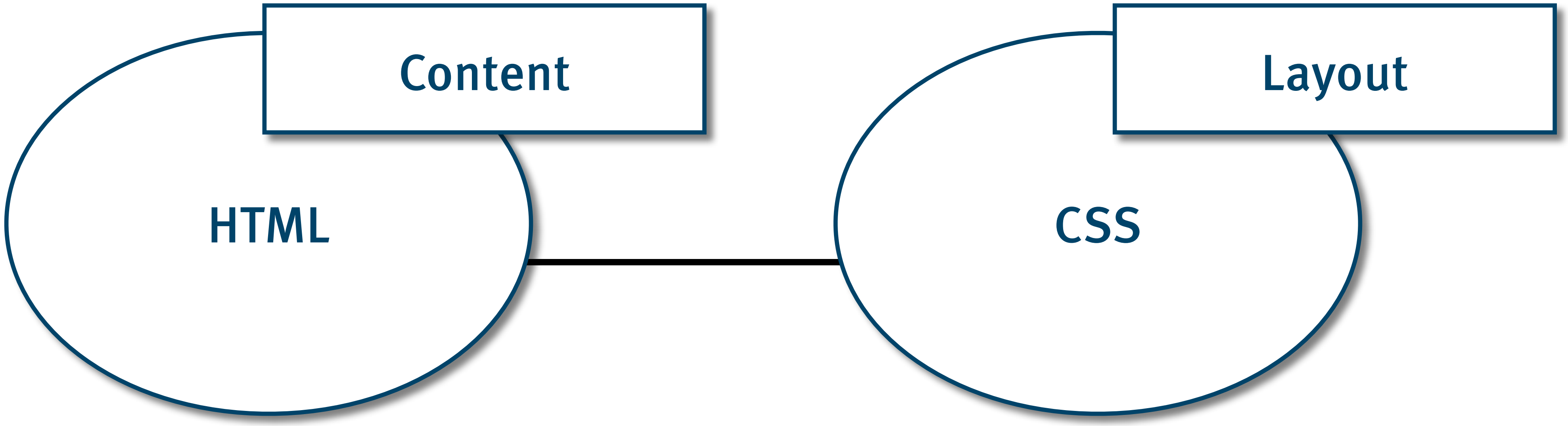
```html
<div id="tabs">
  <ul>
    <li><a href="#tabs-1">Nunc tincidunt</a></li>
    <li><a href="#tabs-2">Proin dolor</a></li>
    <li><a href="#tabs-3">Aenean lacinia</a></li>
  </ul>
  <div id="tabs-1">
    <p>Proin elit arcu, rutrum commodo, vehicula ...</p>
  </div>
  <div id="tabs-2">
    <p>Morbi tincidunt, dui sit amet facilisis feugiat, odio ...</p>
  </div>
  <div id="tabs-3">
    <p>Mauris eleifend est et turpis. Duis id erat ...</p>
  </div>
```

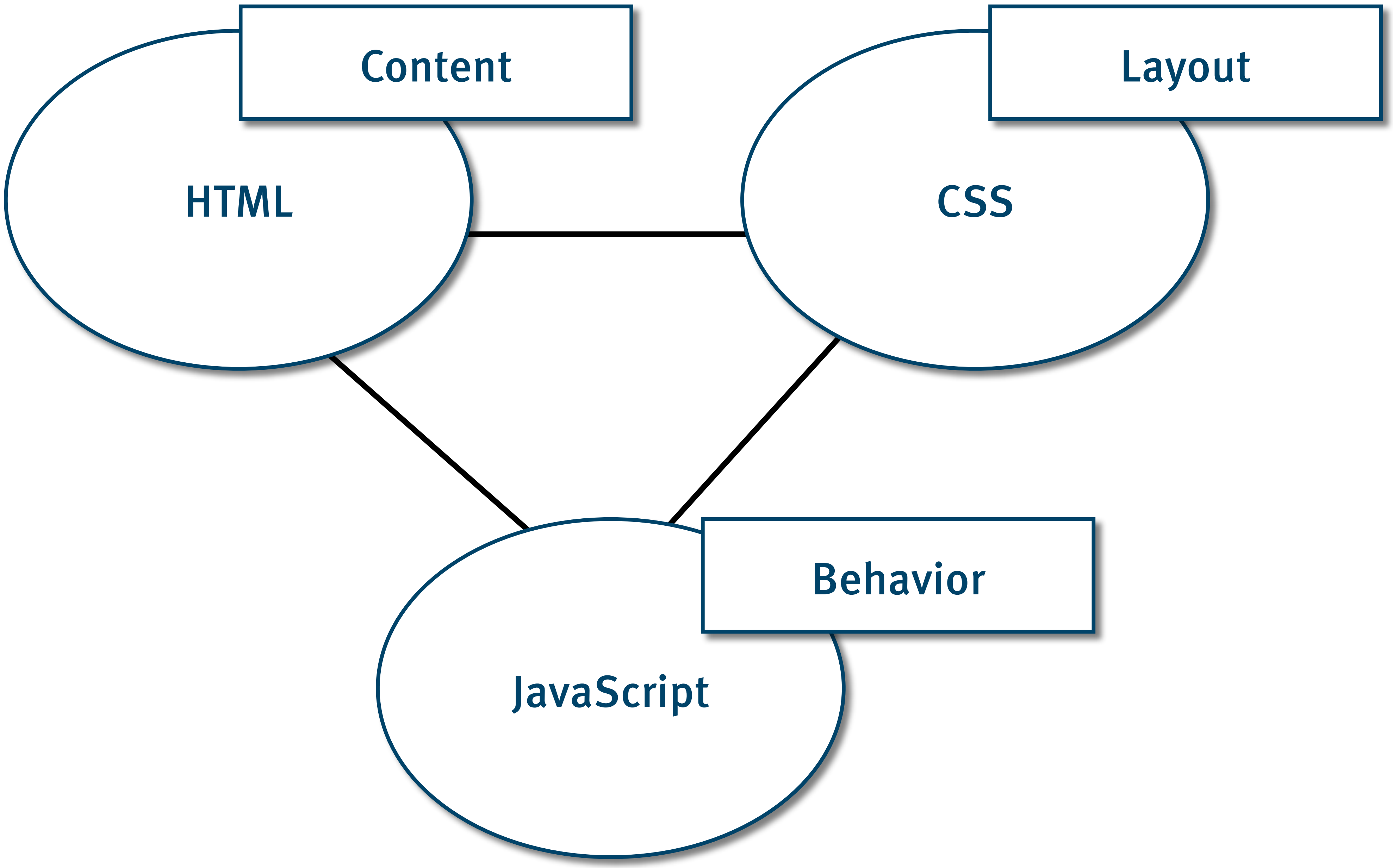
+

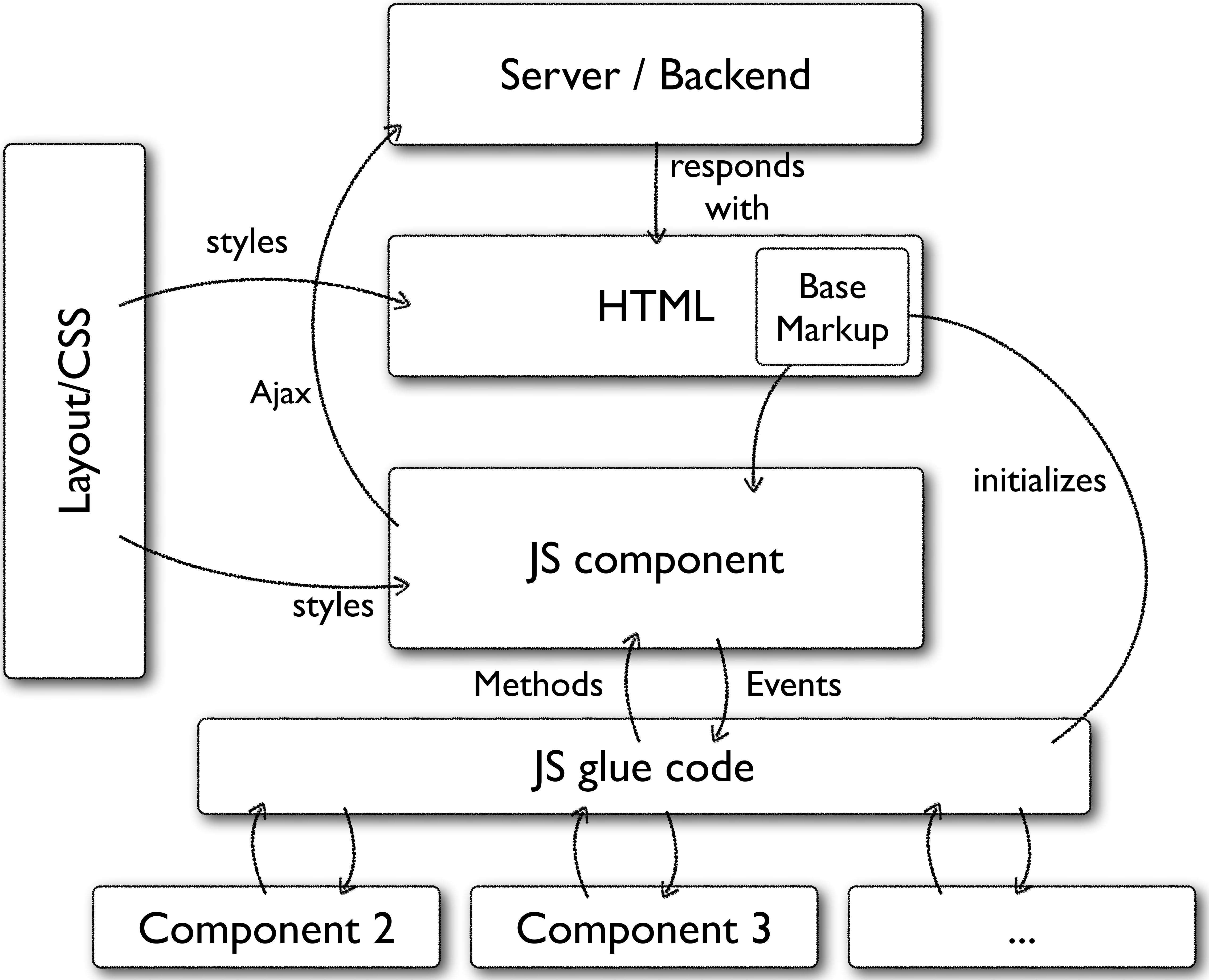
```javascript
$("#tabs").tabs();
```


jQuery UI Tabs – Default functionality
http://jqueryui.com/demos/tabs/default.html
Google
Nunc tincidunt
Proin dolor
Aenean lacinia

Proin elit arcu, rutrum commodo, vehicula tempus, commodo a, risus. Curabitur nec arcu. Donec sollicitudin mi sit amet mauris. Nam elementum quam ullamcorper ante. Etiam aliquet massa et lorem. Mauris dapibus lacus auctor risus. Aenean tempor ullamcorper leo. Vivamus sed magna quis ligula eleifend adipiscing. Duis orci. Aliquam sodales tortor vitae ipsum. Aliquam nulla. Duis aliquam molestie erat. Ut et mauris vel pede varius sollicitudin. Sed ut dolor nec orci tincidunt interdum. Phasellus ipsum. Nunc tristique tempus lectus.

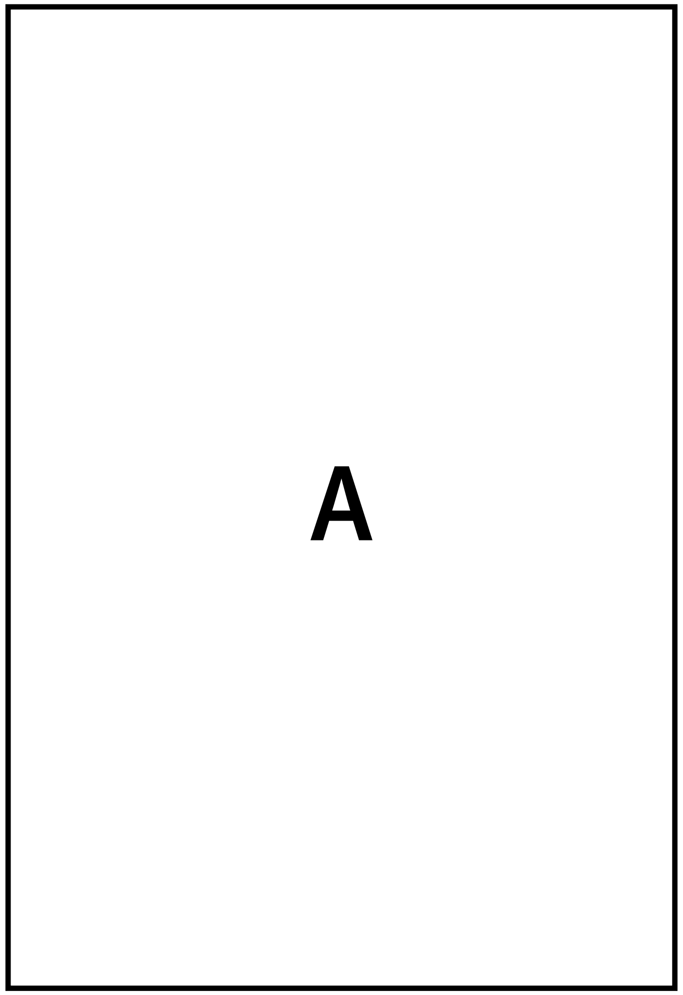

Content
HTML
Layout
CSS


Content
HTML
Layout
CSS
Behavior
JavaScript


Server / Backend
responds with
HTML
Base Markup
styles
Layout/CSS
Ajax
initializes
JS component
styles
Methods
Events
JS glue code
Component 2
Component 3
...

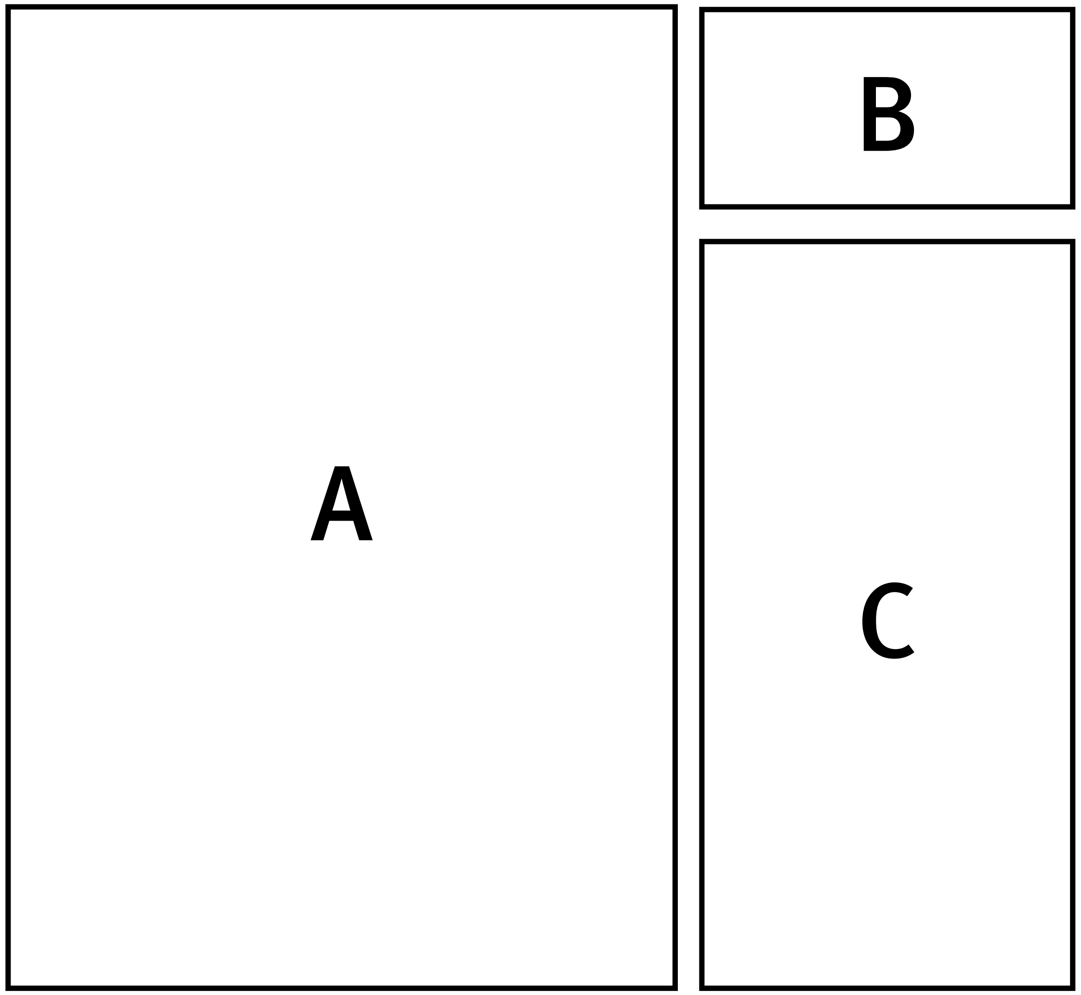
Use URIs to identify a *single* meaningful concept

http://.../A



A

http://.../A

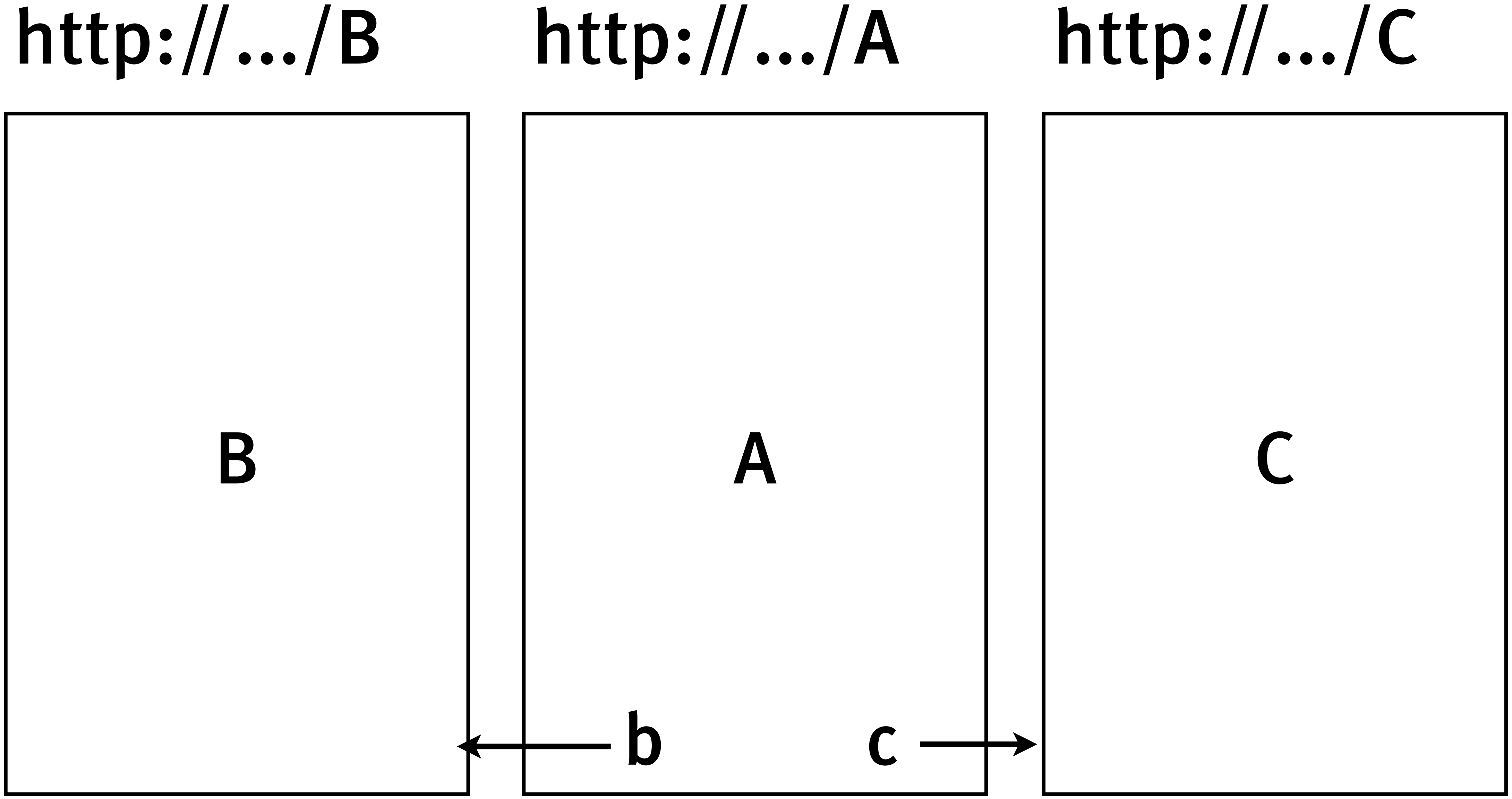


B

A

C

?



http://.../B

http://.../A

http://.../C

B

A

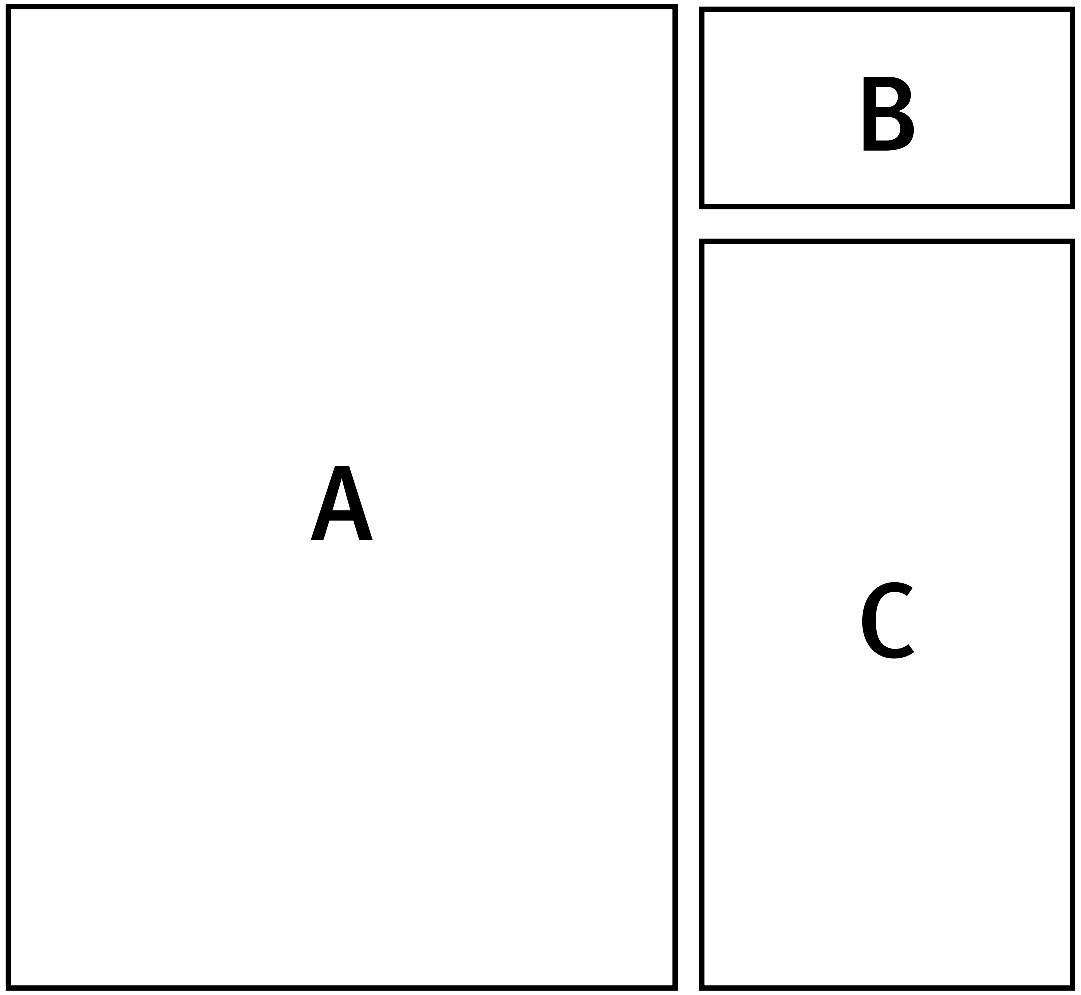
C

b

c

+ AJAX + CSS

http://.../A



A

B

C

Don’t *disable* Browser Features – *use them*

Multiple tabs & windows

Bookmarks

Back/Forward/Refresh

Personalization

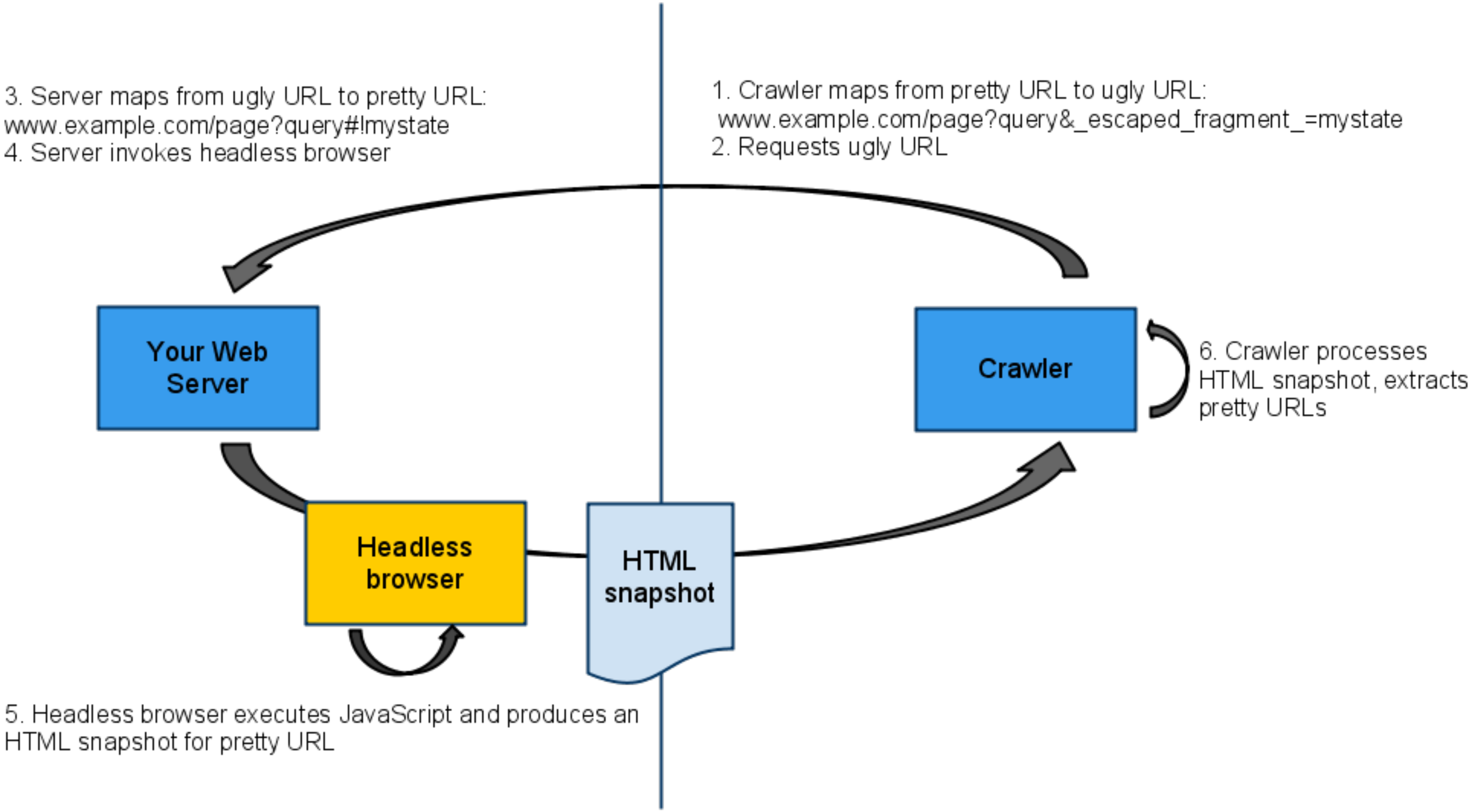
Menus

Put *application* logic on the server *only*

Don’t trust your clients

RESTful APIs for free

Non-browser clients
(e.g. Google Crawler)



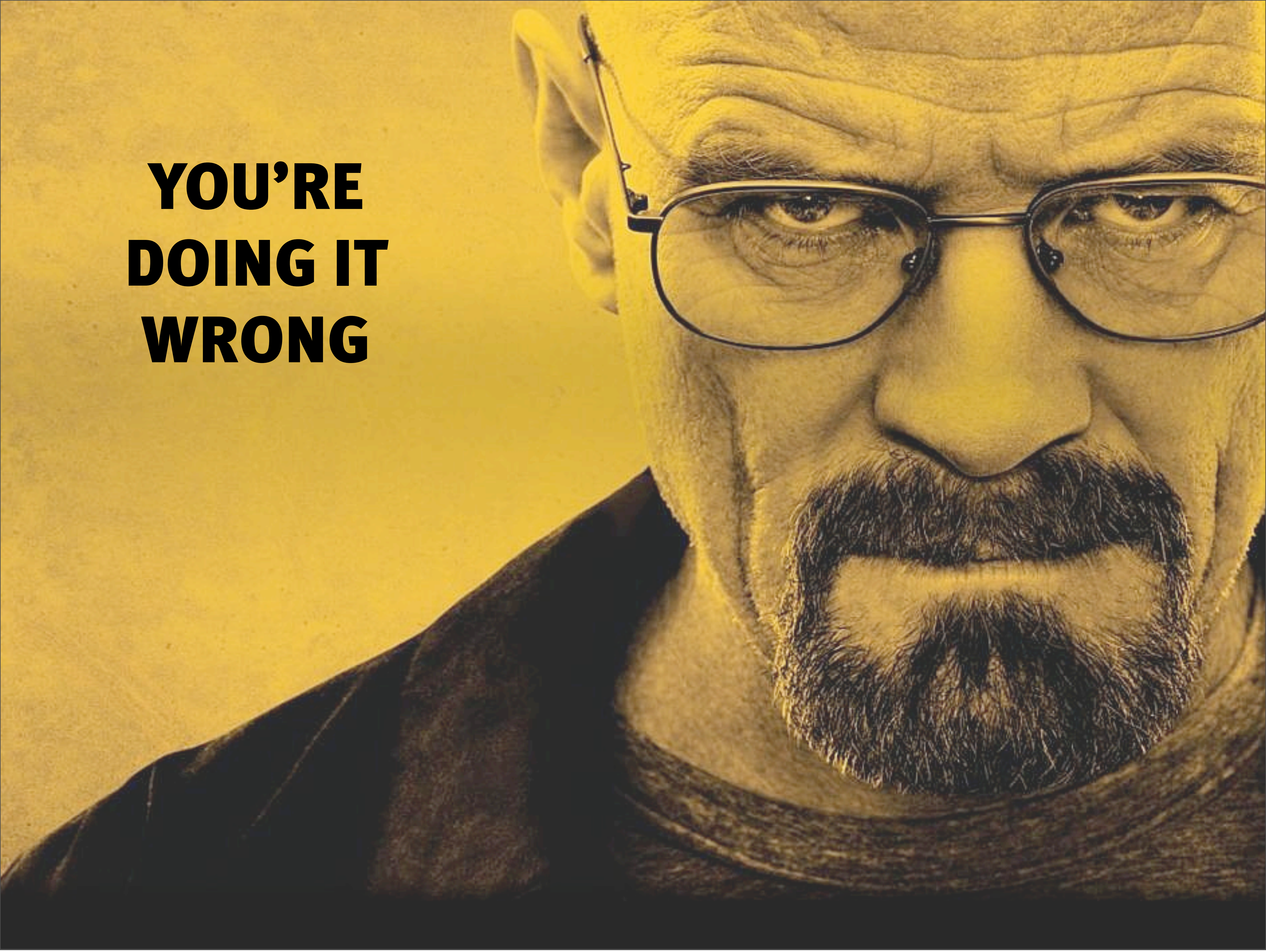
1. Crawler maps from pretty URL to ugly URL:
www.example.com/page?query&_escaped_fragment_=mystate
2. Requests ugly URL

3. Server maps from ugly URL to pretty URL:
www.example.com/page?query#!mystate
4. Server invokes headless browser

5. Headless browser executes JavaScript and produces an HTML snapshot for pretty URL

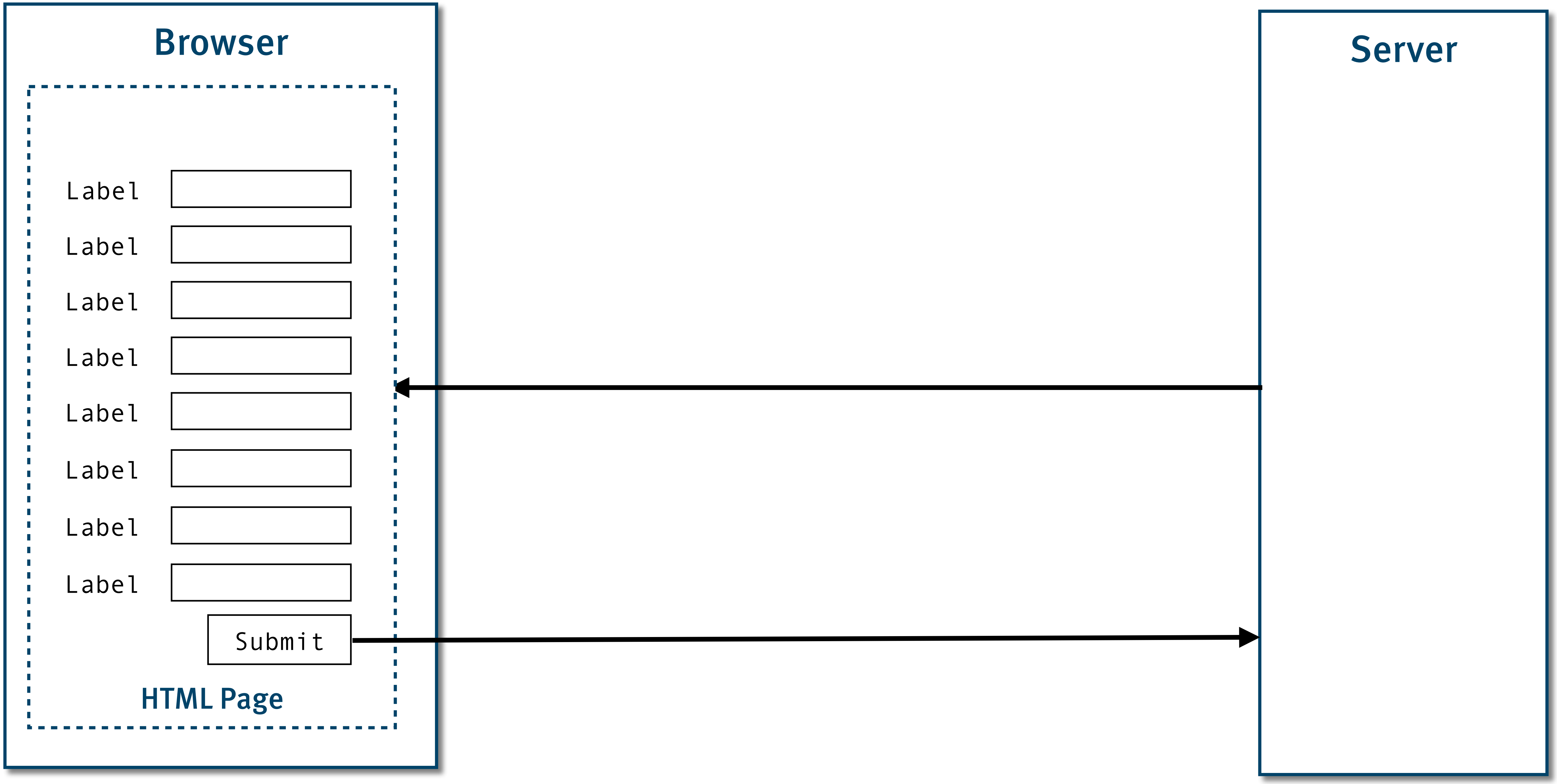
6. Crawler processes HTML snapshot, extracts pretty URLs

http://googlewebmastercentral.blogspot.com/2009/10/proposal-for-making-ajax-crawlable.html



YOU’RE
DOING IT
WRONG

Avoid session state



Browser

Label

Label

Label

Label

Label

Label

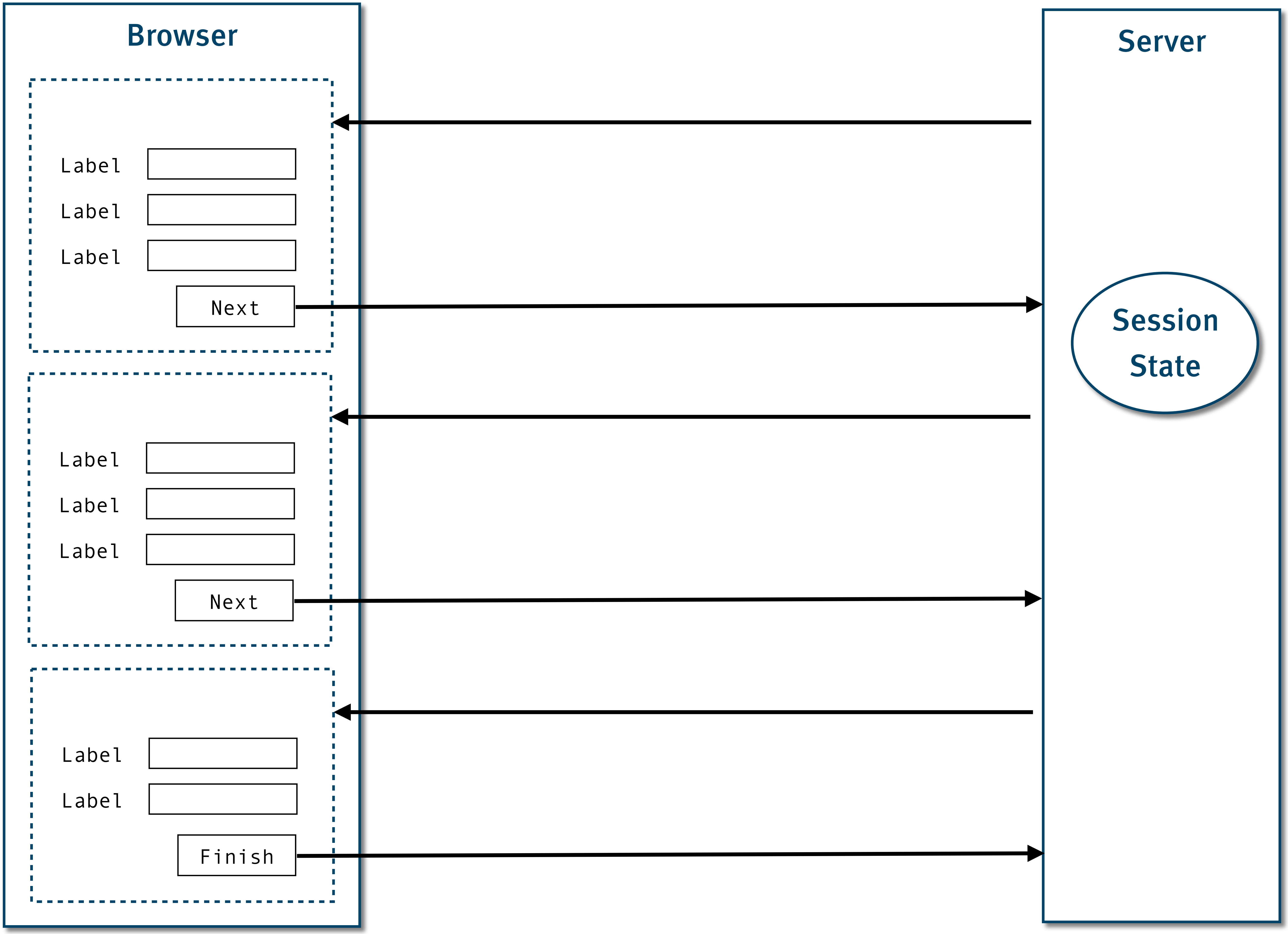
Label

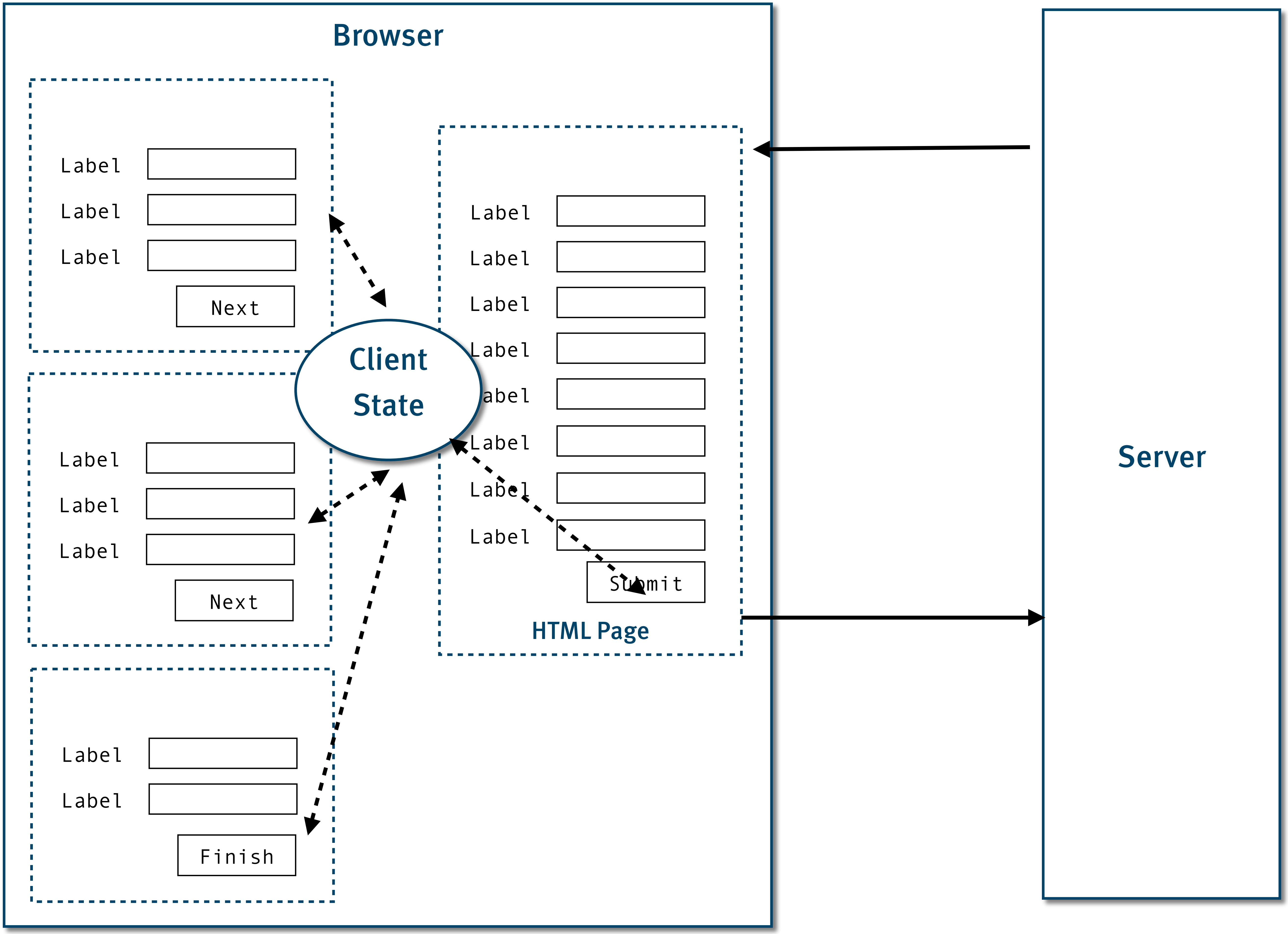
Label

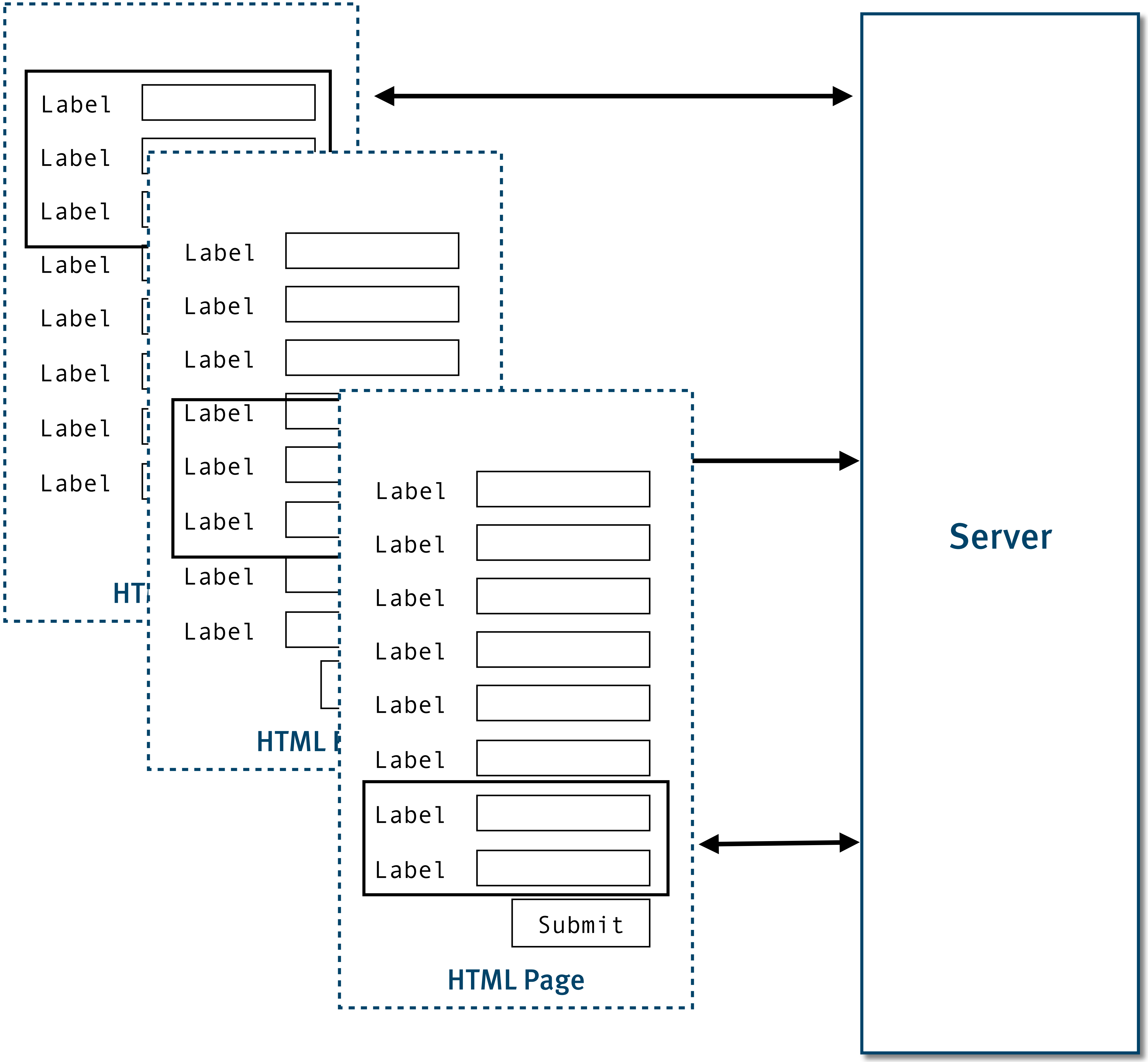
Submit

HTML Page

Server


Browser
Server
Label
Label
Label
Next
Session State
Label
Label
Label
Next
Label
Label
Finish


Browser
Label
Label
Label
Next
Label
Label
Label
Next
Label
Label
Finish
Client State
Label
Label
Label
Label
Label
Label
Label
Label
Submit
HTML Page
Server

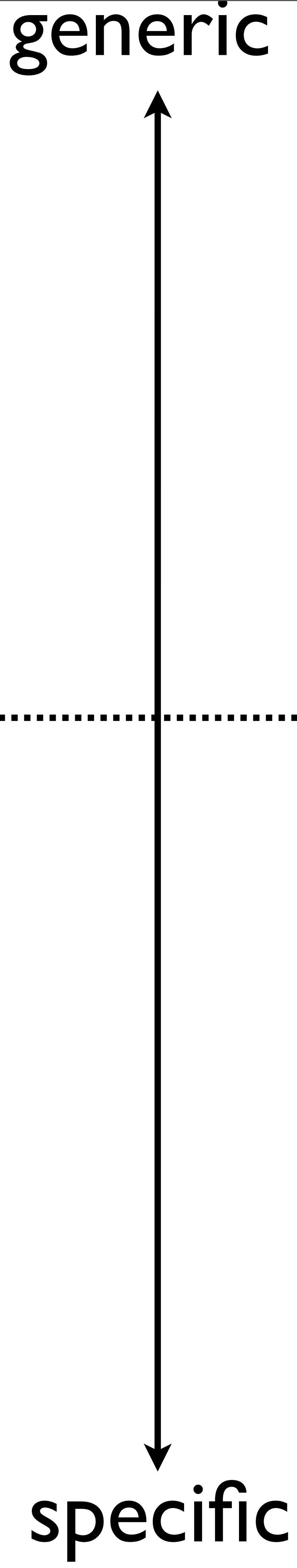

Label
Label
Label
Label
Label
Label
Label
Label
HT
Label
Label
Label
Label
Label
Label
Label
Label
HTML P
Label
Label
Label
Label
Label
Label
Label
Label
Submit
HTML Page
Server

ROCA

http://roca-style.org/

So why don’t we do this?

Obsessive Specialization Disorder



generic

specific

```
interface Resource {
     Resource(URI u)
     Response get()
     Response post(Request r)
     Response put(Request r)
     Response delete()
     Link[] links()
     Form[] forms()
}
```

```
class MyResource implements Resource {
     …
     Response post(Request r) {
          …
     }
     …
}
```

Severe Tunneling Fixation

simple	easy
complex	hard

In summary …

Not everything new is good,

not everything old is bad.

(Surprise.)

Learning details of how the web works might be time well spent

Working with the web is a lot easier than fighting it

Thank you!

Q&A

Stefan Tilkov
stefan.tilkov@innoq.com
http://www.innoq.com/blog/st/
@stilkov
Phone: +49 170 471 2625

innoQ

innoQ Deutschland GmbH
Krischerstr. 100
40789 Monheim am Rhein
Germany
Phone: +49 2173 3366-0
http://www.innoq.com

innoQ Schweiz GmbH
Gewerbestr. 11
CH-6330 Cham
Switzerland
Phone: +41 41 743 0116
info@innoq.com