



ROCA im Java-EE-Kontext

Endlich webkonforme Anwendungen bauen!

Holger Kraus

Frederik Dohr

Wir lösen das – persönlich!

innoQ

Java EE - Context



Java EE - Context





Jakub Korab
@jakekorab

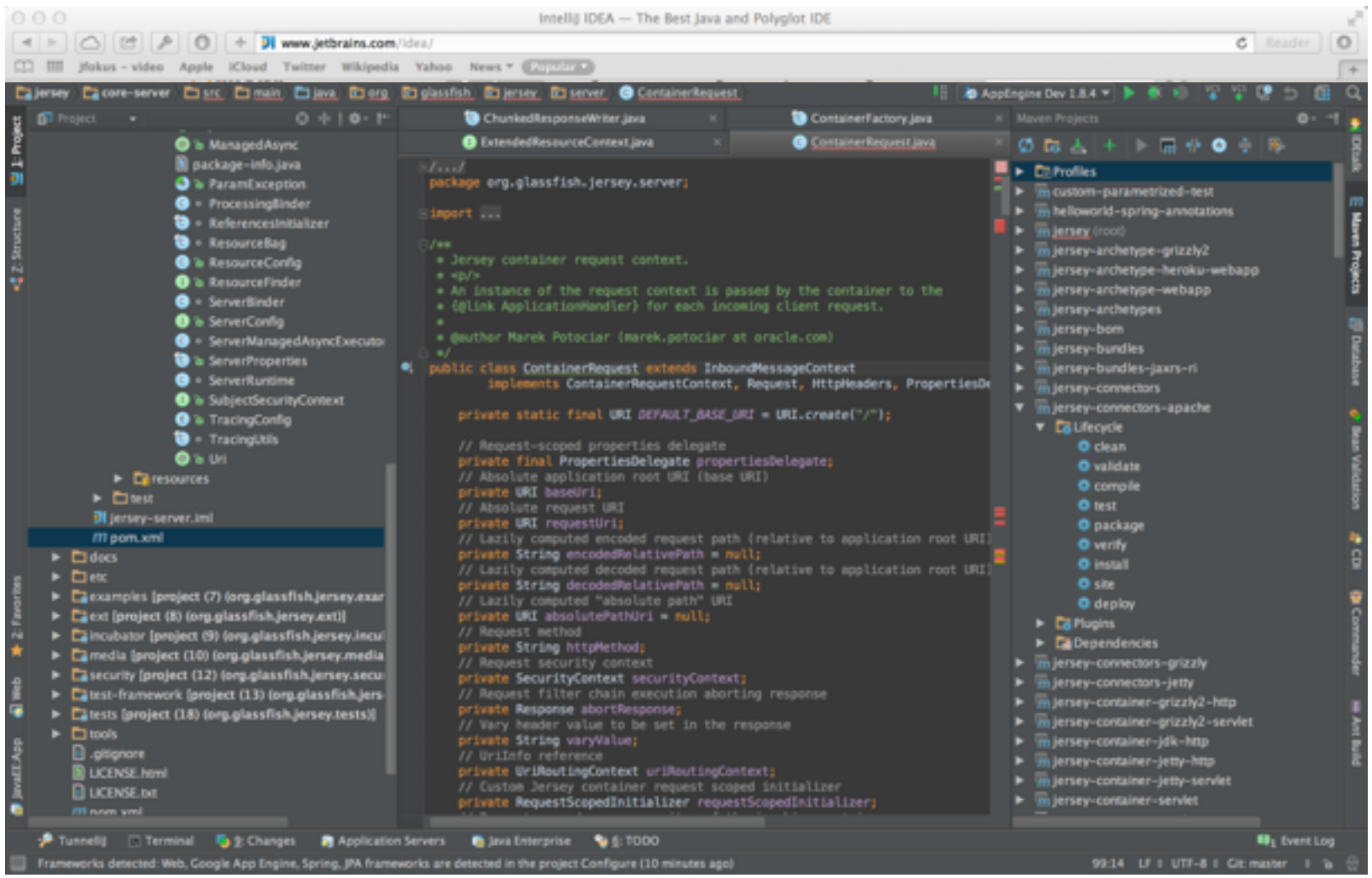
 Follow

Korab's Law pic.twitter.com/9tmE2nadVH

 Reply  Retweet  Favorite  More



<https://twitter.com/jakekorab/status/443701062143188992>



Servlet

JSP

JAX-WS

JAX-RS

JASPIC

JSR 88

JSF

CDI

JAX-RPC

JAXB

JACC

JSR 77

EJB

JTA

JAXR

JMS

JCA

RMI

JPA

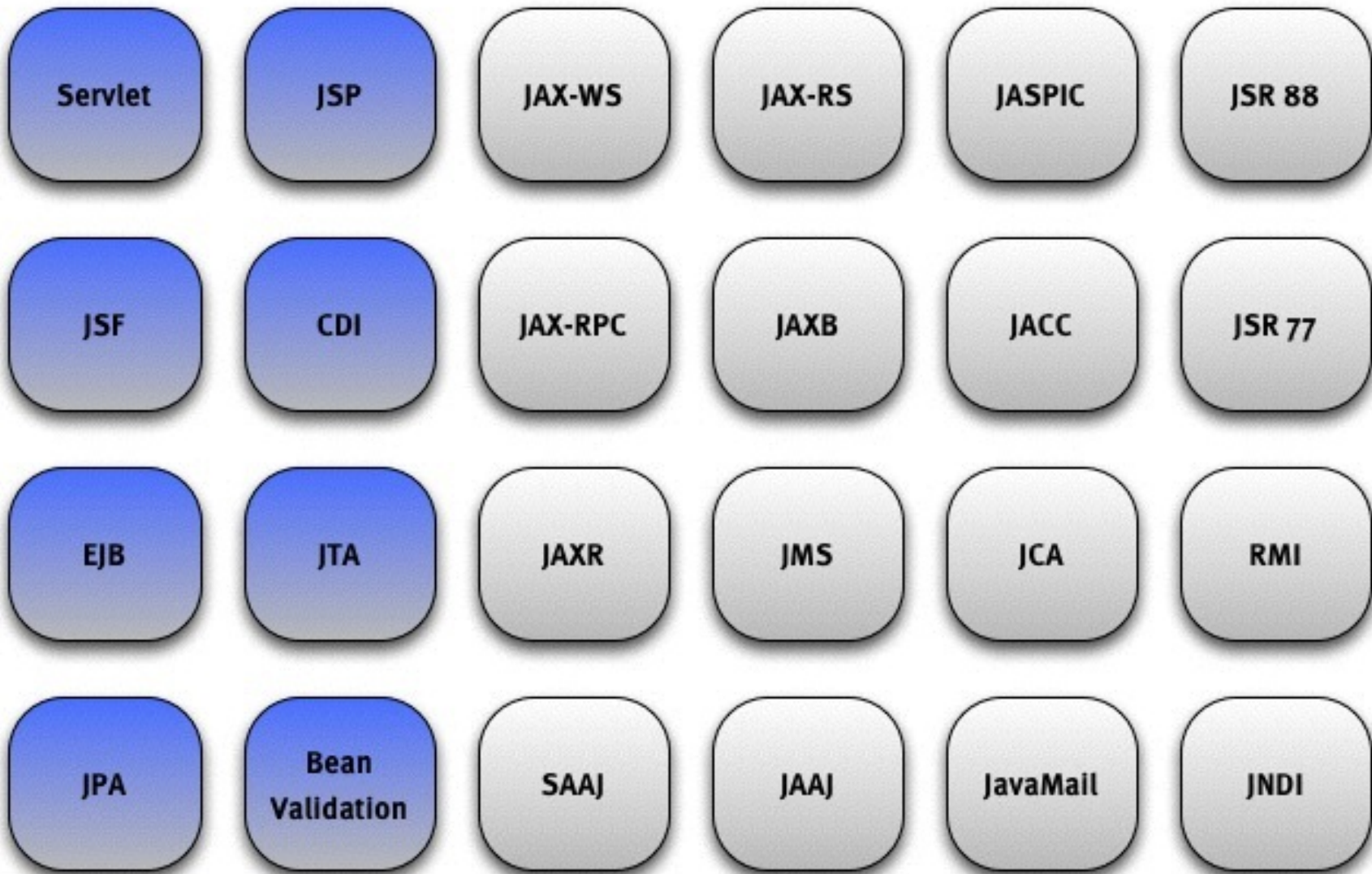
**Bean
Validation**

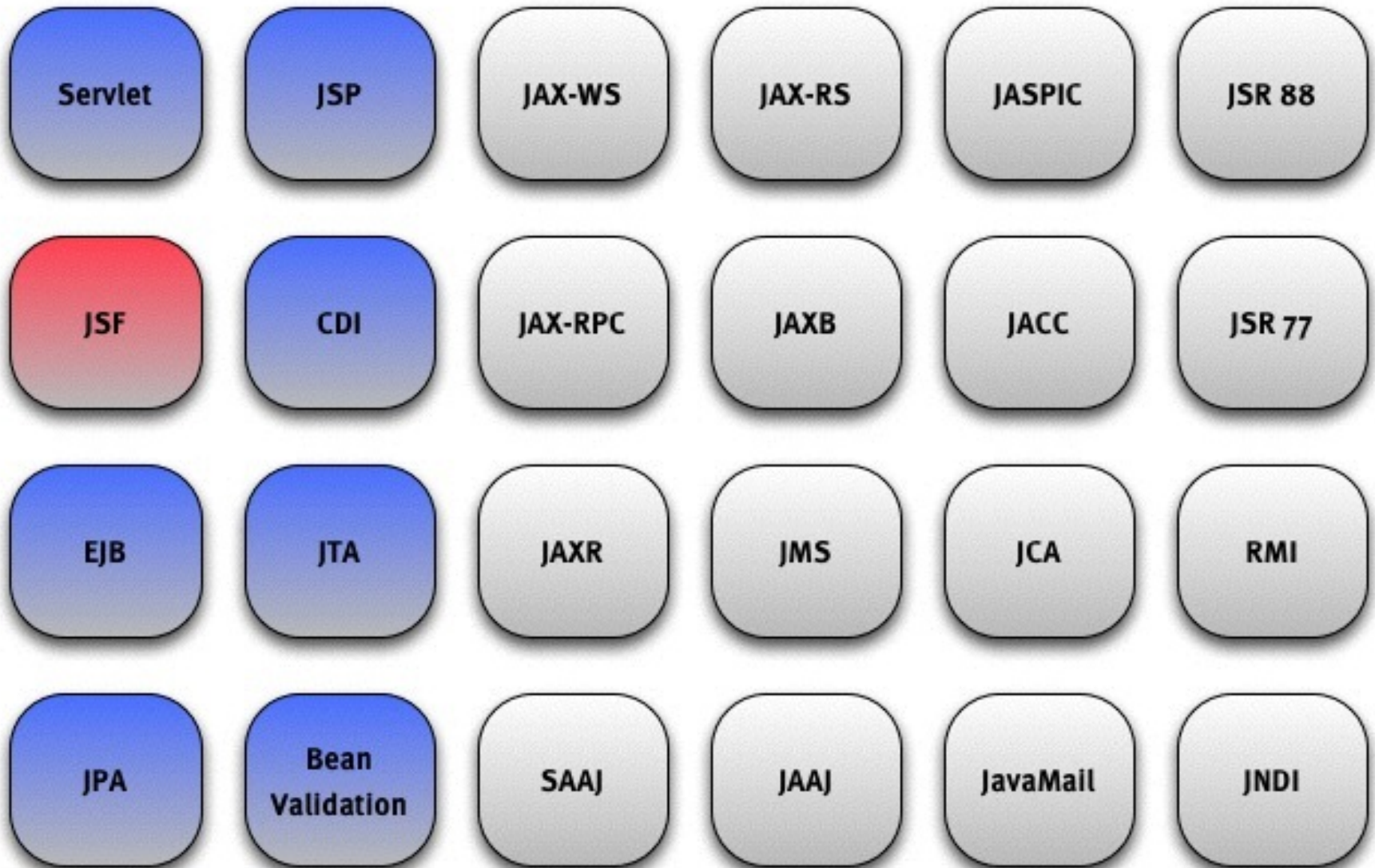
SAAJ

JAAJ

JavaMail

JNDI







UI Components

Validators

Converters

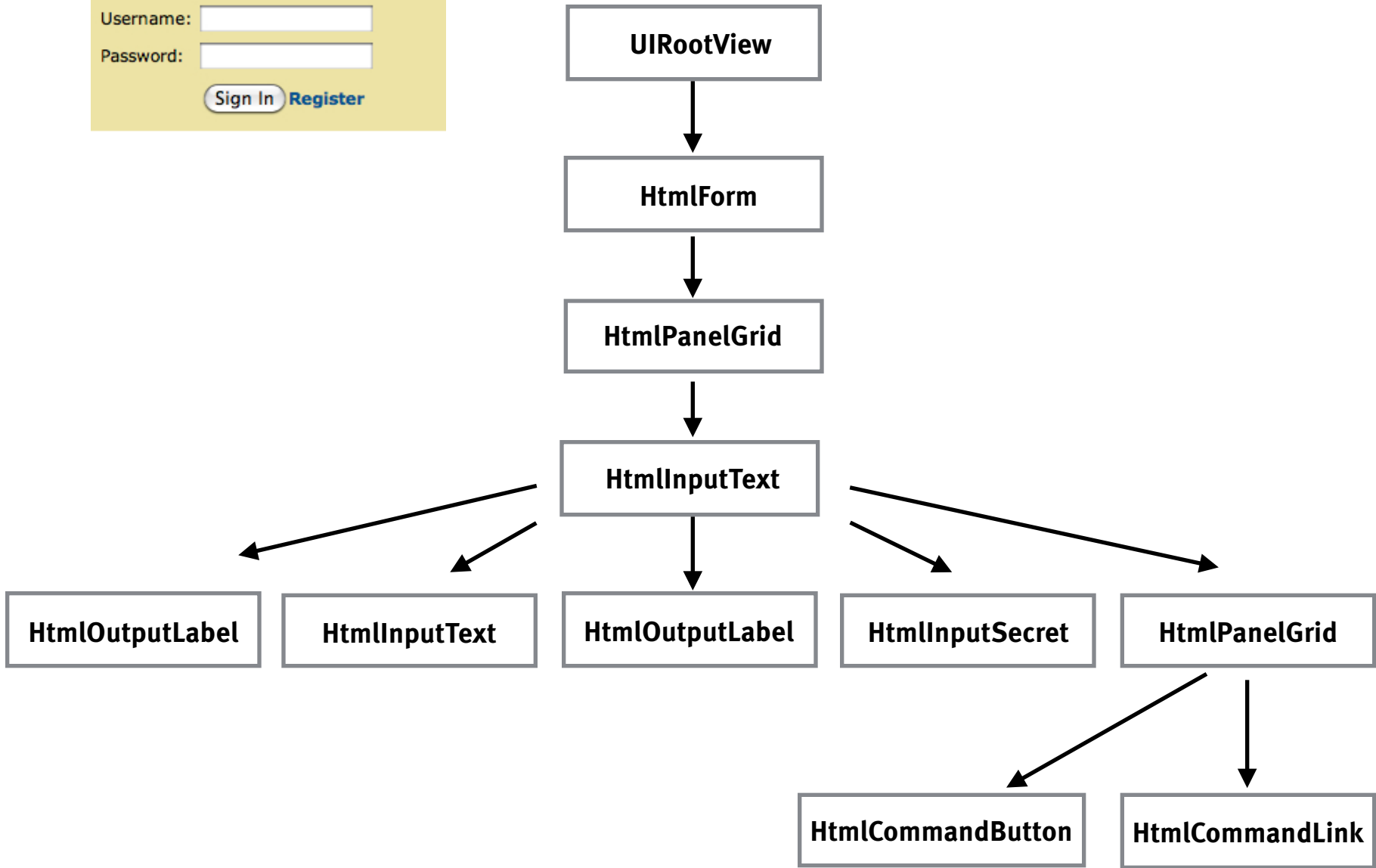
Events

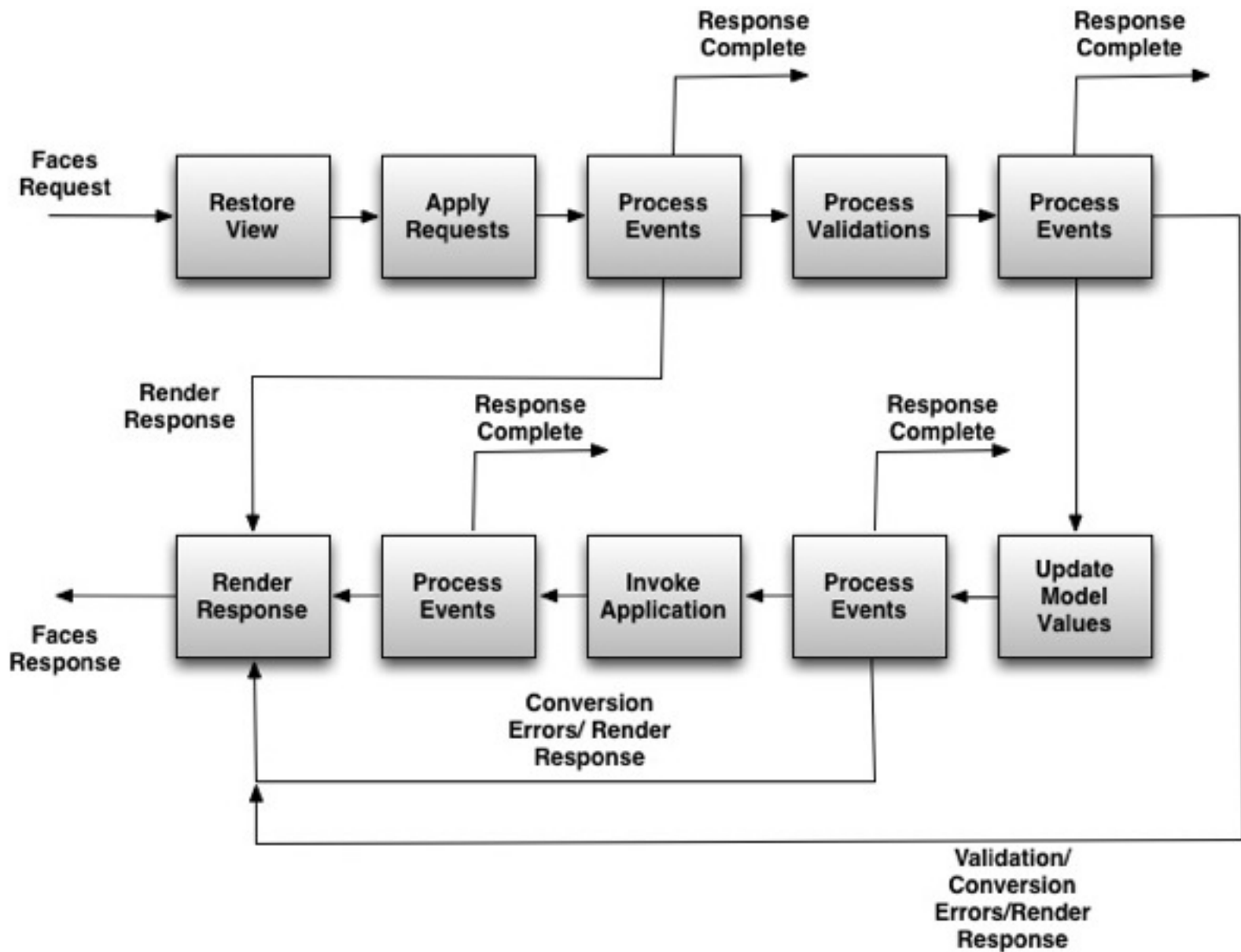
Managed Beans

Username:

Password:

[Register](#)





The
Complete
Reference

JavaServer Faces 2.0

JavaServer
Faces 2.0

- ✦ Get full coverage of JSF 2.0
- ✦ Learn how to use Ajax and build components the JSF 2.0 way
- ✦ Jump-start your understanding with numerous ready-to-run code samples

Ed Burns

Co-Spex Lead, JSE, Sun
Microsystems, Inc.

Chris Schalk

Google Developer Advocate

With Neil Griffin

JSF Team Lead, Liberty Portal

Edited by Herbert Schildt
Java Expert and Bestselling
Author

**Mc
Graw
Hill**

Burns
Schalk

**Mc
Graw
Hill**

Founding Principles of the Web

The web can be made to work with any type of information, on any device, with any software, in any language.

Founding Principles of the Web

“By design, the [...] Internet and the WWW are **non-hierarchical, decentralized** and **radically open**.

The web can be made to work with any type of information, on any device, with any software, in any language.

Founding Principles of the Web

“By design, the [...] Internet and the WWW are non-hierarchical, decentralized and radically open. The web can be made to work with any type of information, on any device, with any software, in any language.

You can link to any piece of information. You don't need to ask for permission.”

— Tim Berners Lee

Founding Principles of the Web

Simplicity

Modularity

Transparency

Tolerance

Rule of Least Power

“choosing the least powerful language suitable for a given purpose”

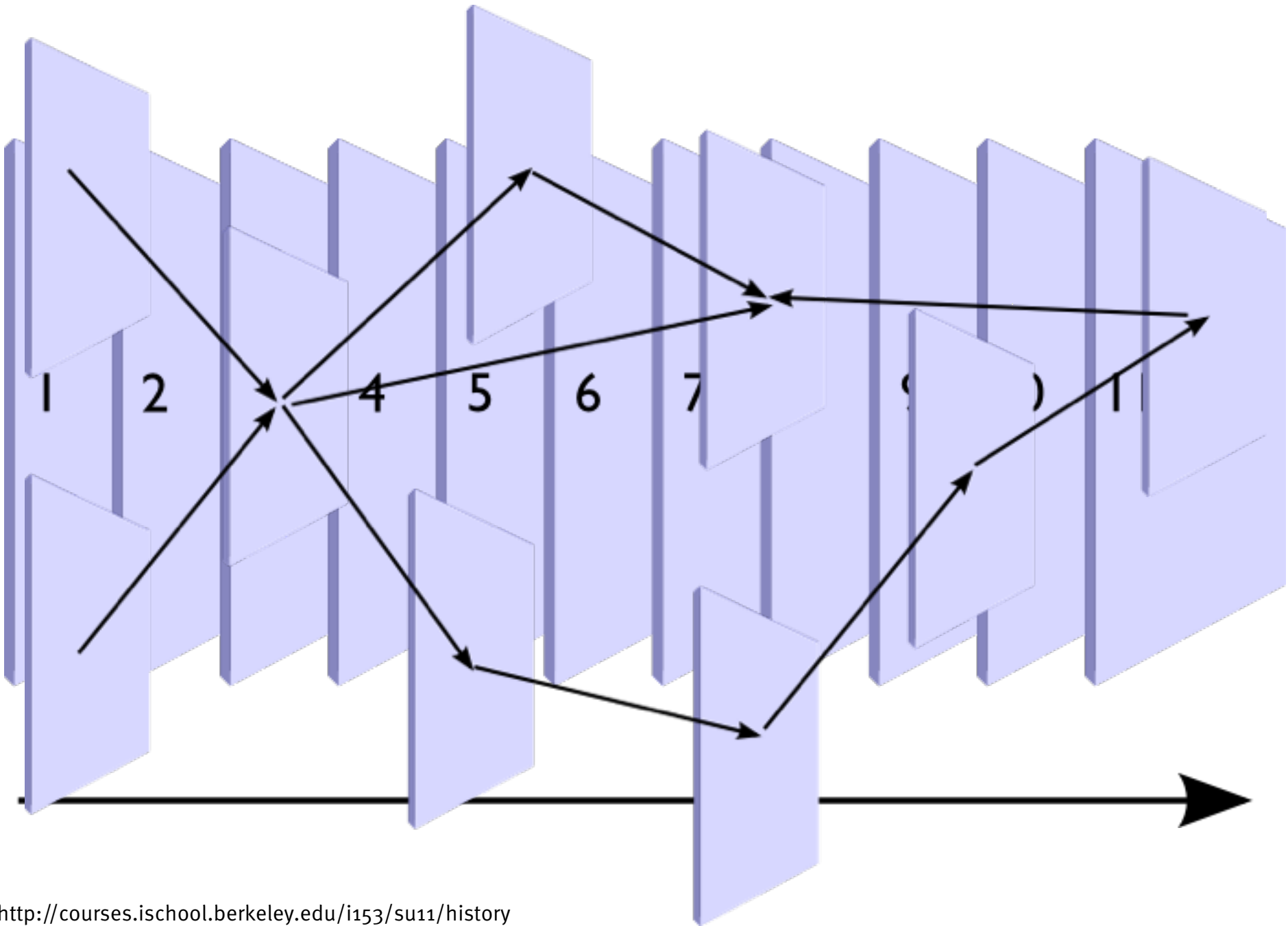
“the less powerful the language, the more you can do with the data stored in that language. If you write it in a **simple declarative form**, anyone can write a program to analyze it in many ways.”

— Axioms of Web Architecture

Rule of Least Power

“I chose HTML not to be a programming language because I wanted different programs to do different things with it: present it differently, extract tables of contents, index it, and so on.”

— Tim Berners Lee



<http://courses.ischool.berkeley.edu/i153/su11/history>

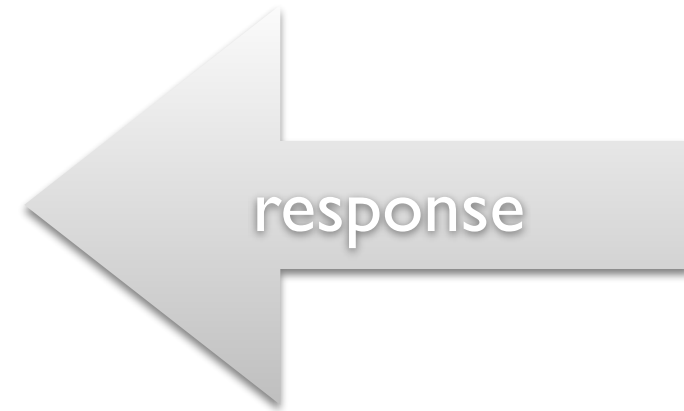
GET /index HTTP/1.1

Host: example.org



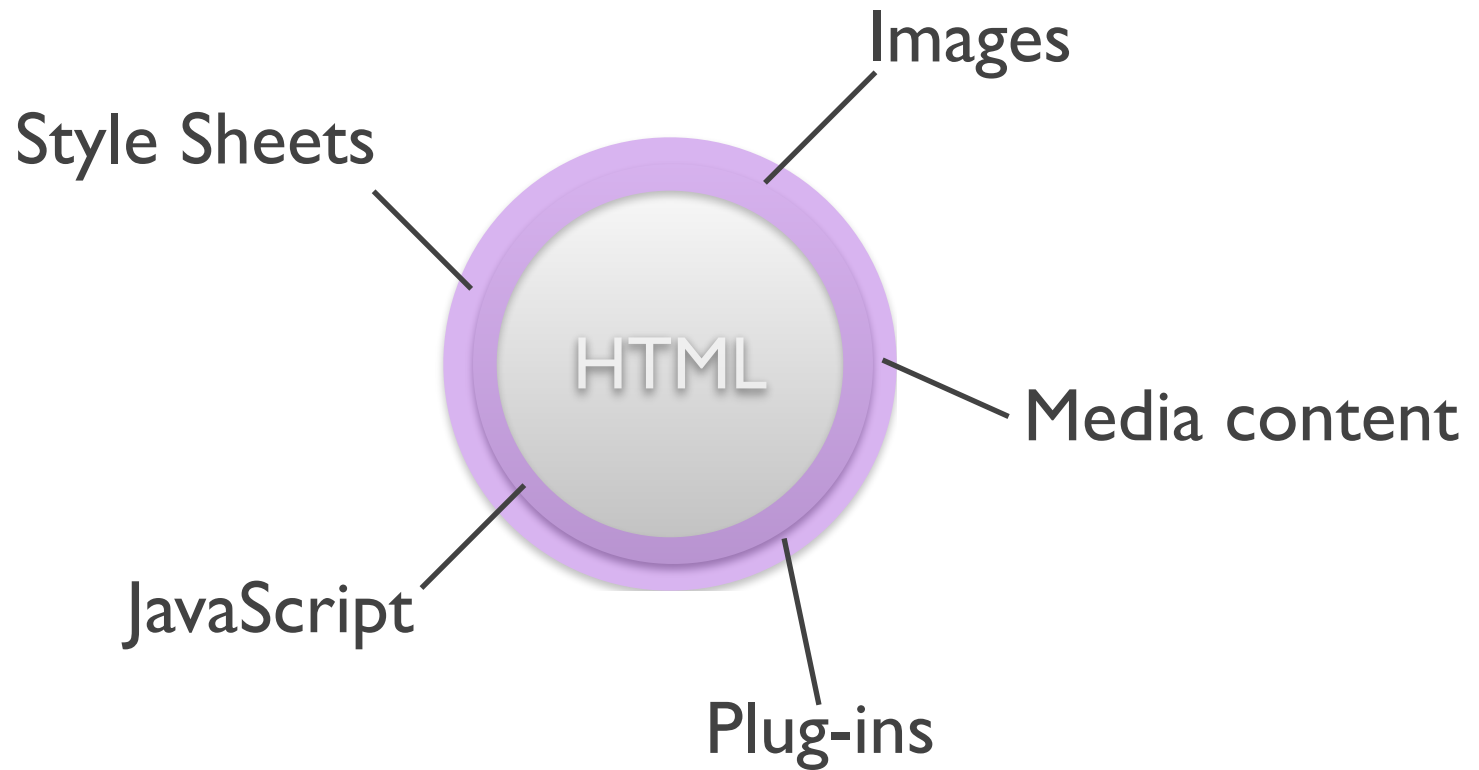
HTTP/1.1 200 OK

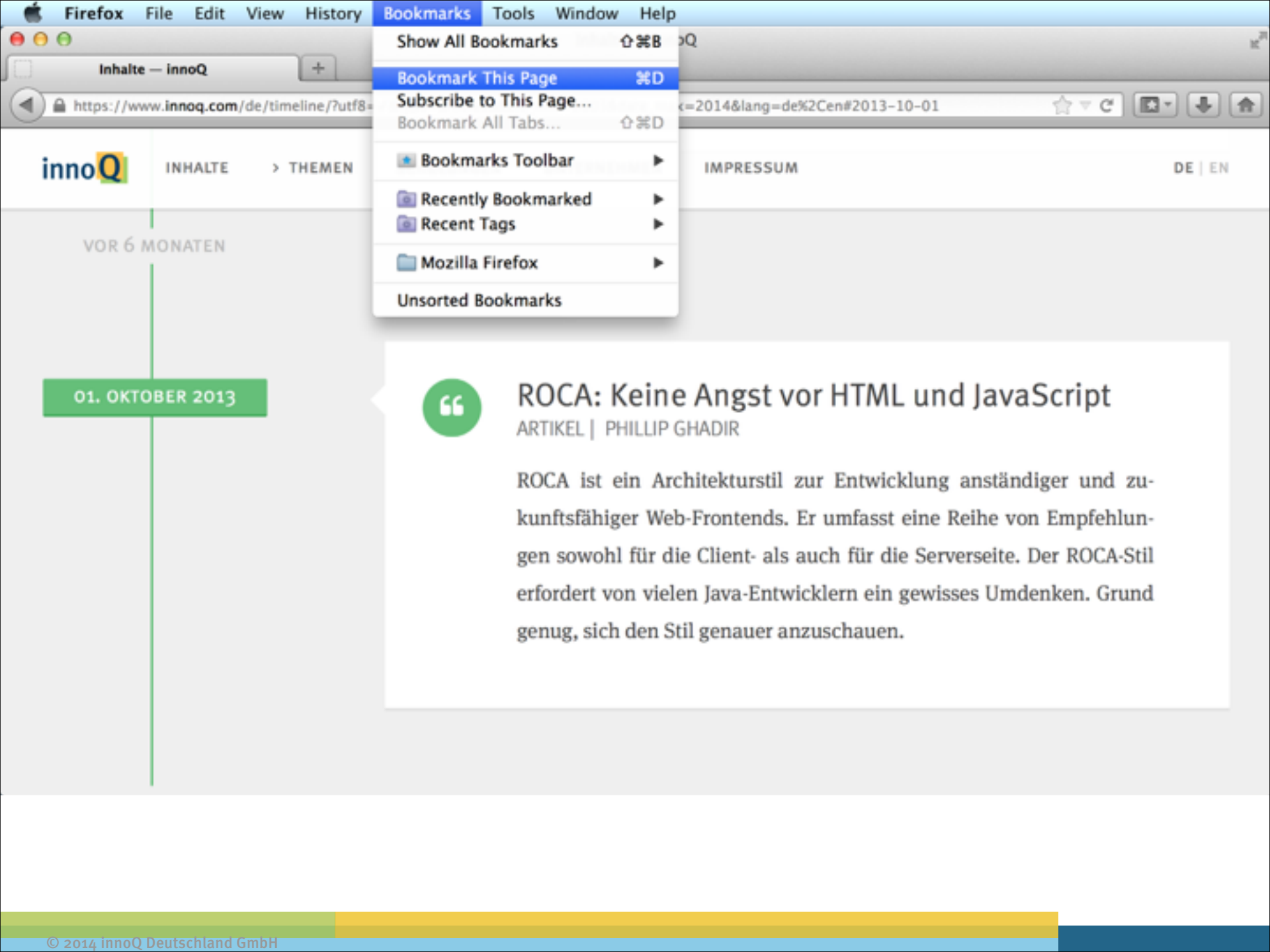
Content-Type: text/html



<html> ... </html>







VOR 6 MONATEN

01. OKTOBER 2013



ROCA: Keine Angst vor HTML und JavaScript

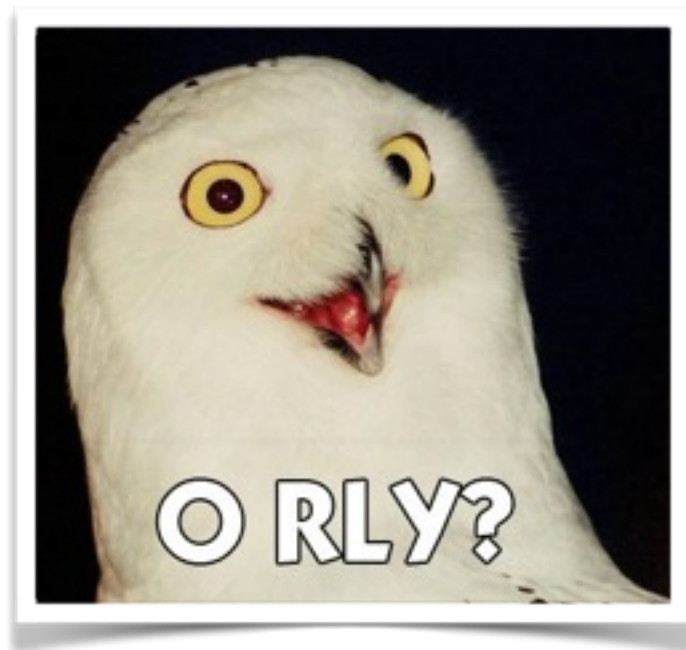
ARTIKEL | PHILLIP GHADIR

ROCA ist ein Architekturstil zur Entwicklung anständiger und zukunftsfähiger Web-Frontends. Er umfasst eine Reihe von Empfehlungen sowohl für die Client- als auch für die Serverseite. Der ROCA-Stil erfordert von vielen Java-Entwicklern ein gewisses Umdenken. Grund genug, sich den Stil genauer anzuschauen.

Semantic HTML

“The main point of semantics was accessibility and search engines.”

— @jcroft



<http://www.flickr.com/photos/saltyscott/212796193/>

```
<font size="big" color="#990099">  
  <b>Welcome to my homepage!</b>  
  <img size="under_construction.gif">  
  <p>  
    * Guns N' Roses  
    * Bon Jovi  
    * MC Hammer  
  </p>  
</font>
```

<h1>Hello from Facebook!</h1>

Justin Bieber

Beyoncé

Miley Cyrus

POSH

Plain

Old

Semantic

HTML



http://commons.wikimedia.org/wiki/File:Victoria_Beckham_2010.jpg

Unobtrusive JavaScript

```
<script type="text/javascript">  
document.write("hello world");  
</script>  
<script src="main.js"></script>
```

Unobtrusive JavaScript

```
<a href="javascript:doStuff();" >
```

...

```
</a >
```

Unobtrusive JavaScript

```
<a href="#"  
    onclick="doStuff();">
```

...

```
</a>
```


Unobtrusive JavaScript

```
<a href="/my-resource"  
    onclick="doStuff(this.href);">
```

...

```
</a>
```

Unobtrusive JavaScript

```
<a href="/my-resource"  
    class="interactive">
```

```
...    +
```

```
</a>    $("interactive").click(function() {  
        doStuff(this.href);  
    });
```

Components

`<input type="date">`

+

`jQuery("input[type=date]").datepicker();`

`<table class="sortable">...</table>`

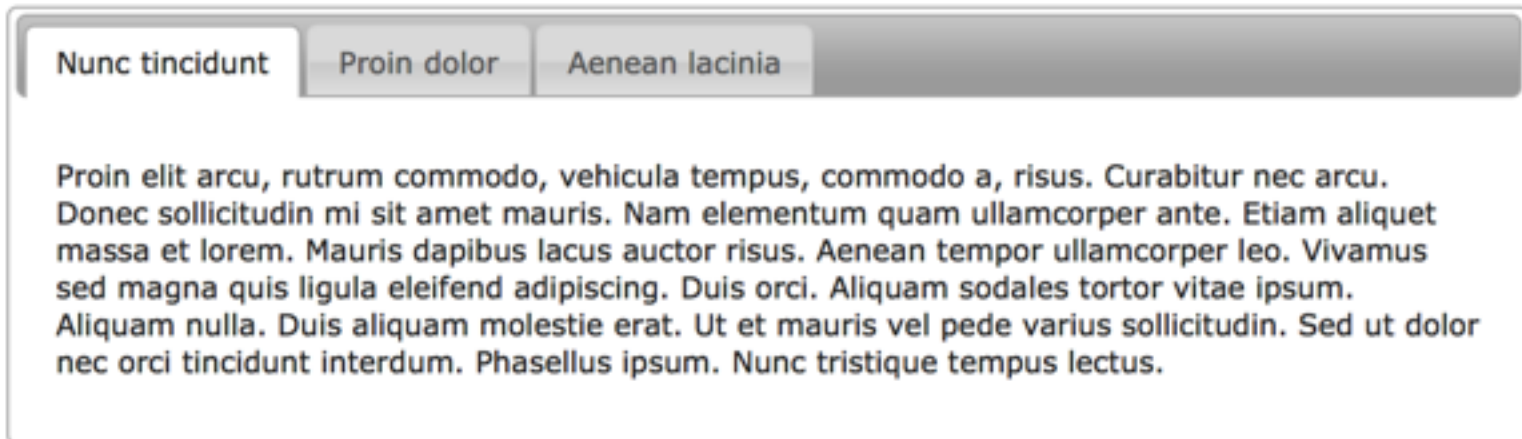
+

`Sortable.init(".sortable");`

```
<div class="tabs">
```

```
<ul>
```

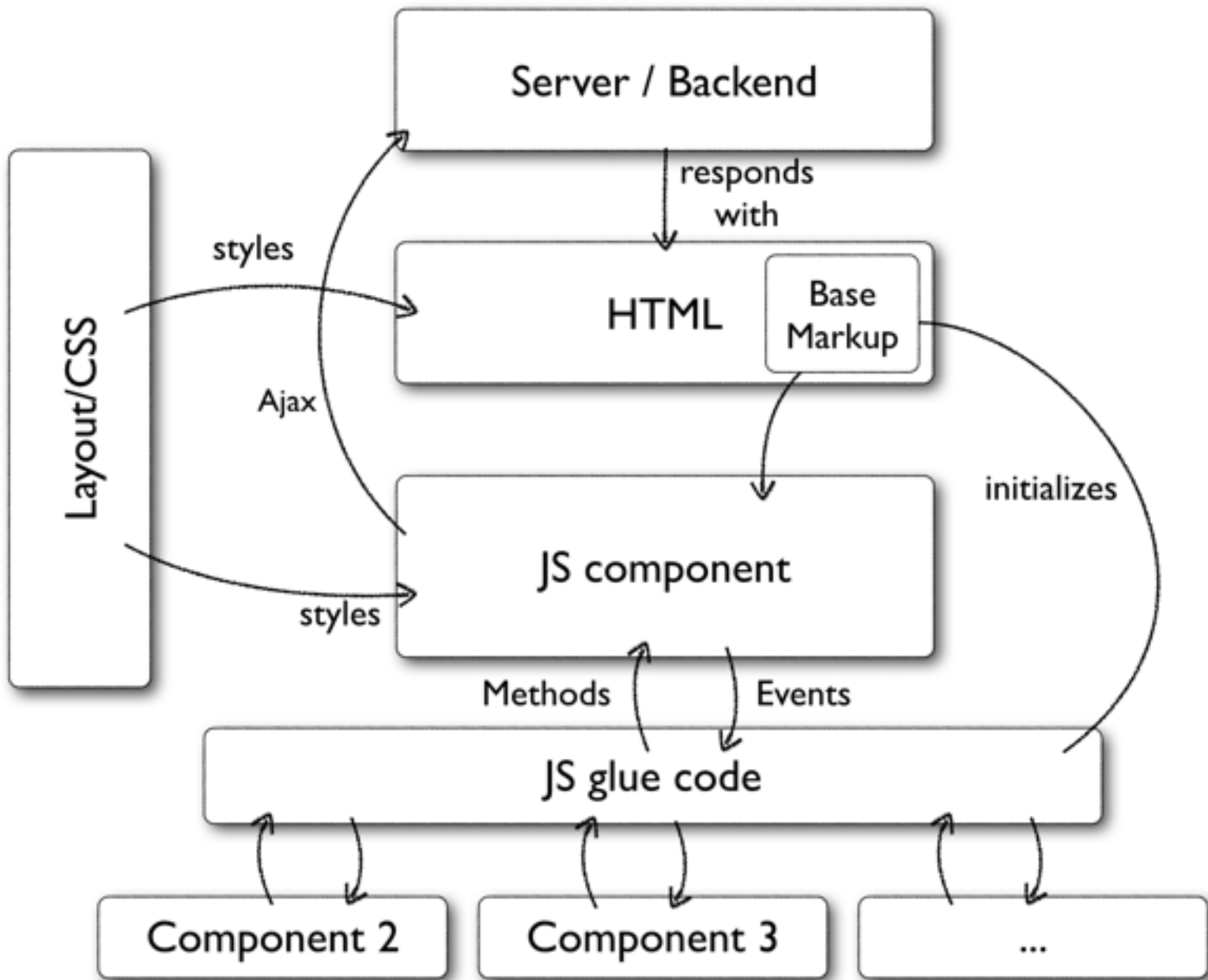
```
<li><a href="#about">About</a>
```



```
<div id="comments">...</div>
```

```
</div>
```

```
jQuery(".tabs").tabs();
```



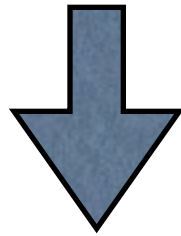
That was the client side...

of what we call

ROCA

**Now let's talk about the
server side**

**REST is more
then a RESTful API**



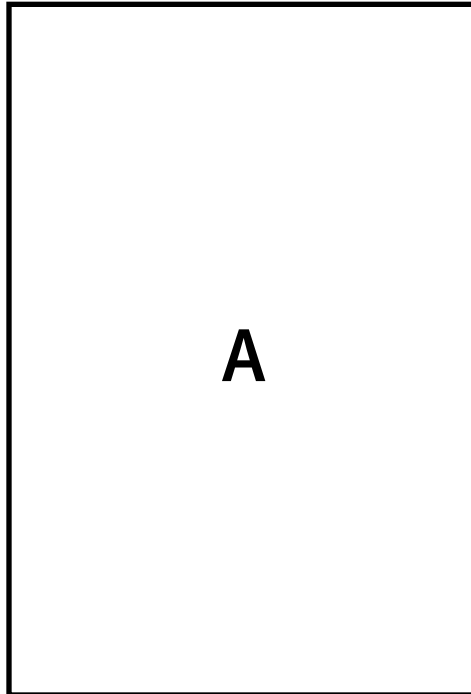
**REST is the
Architecture of the Web**

Avoid session state!

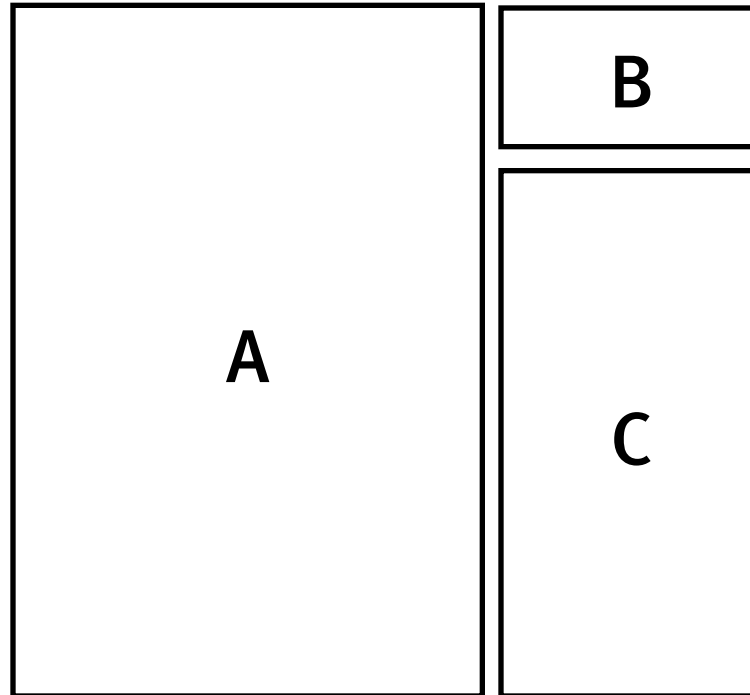
**Put all application logic on
the server!**

**Use URIs to identify a *single*
meaningful concept!**

http://.../A



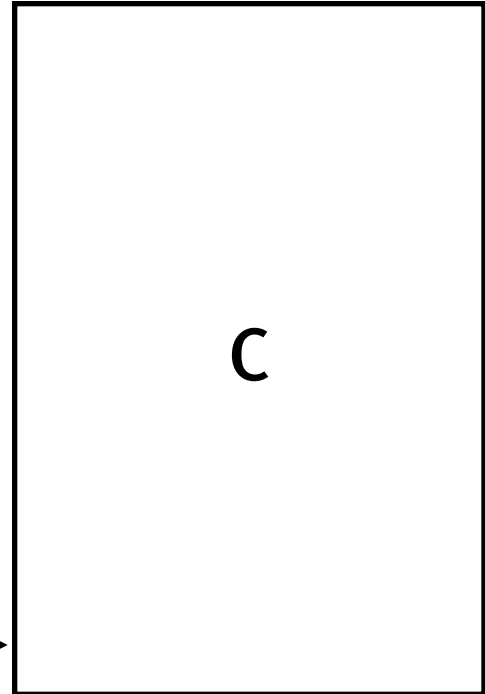
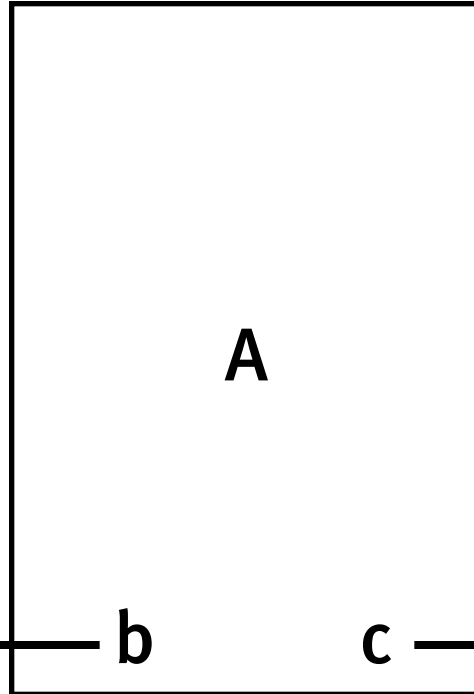
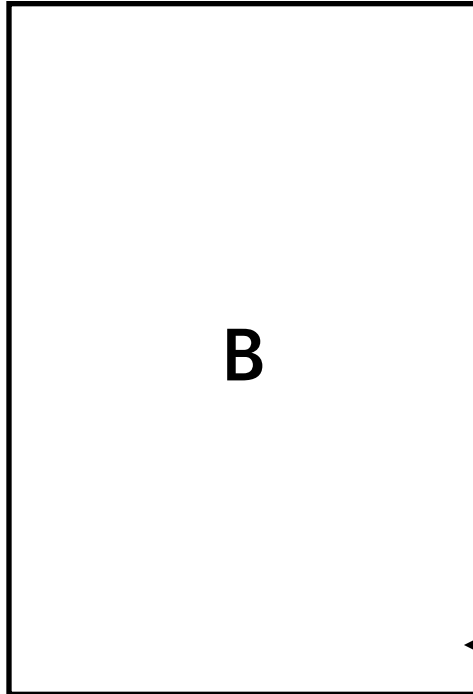
http://.../A



http://.../B

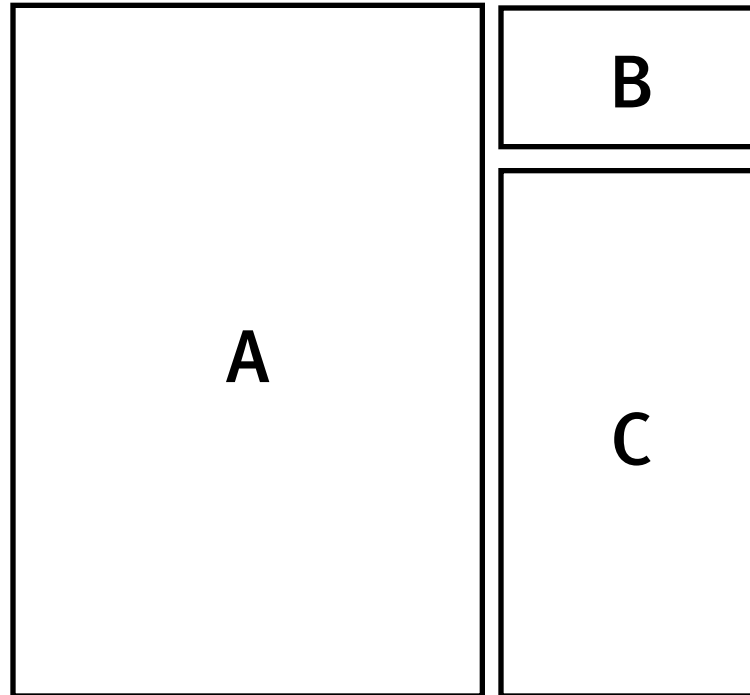
http://.../A

http://.../C



+ AJAX + CSS

http://.../A

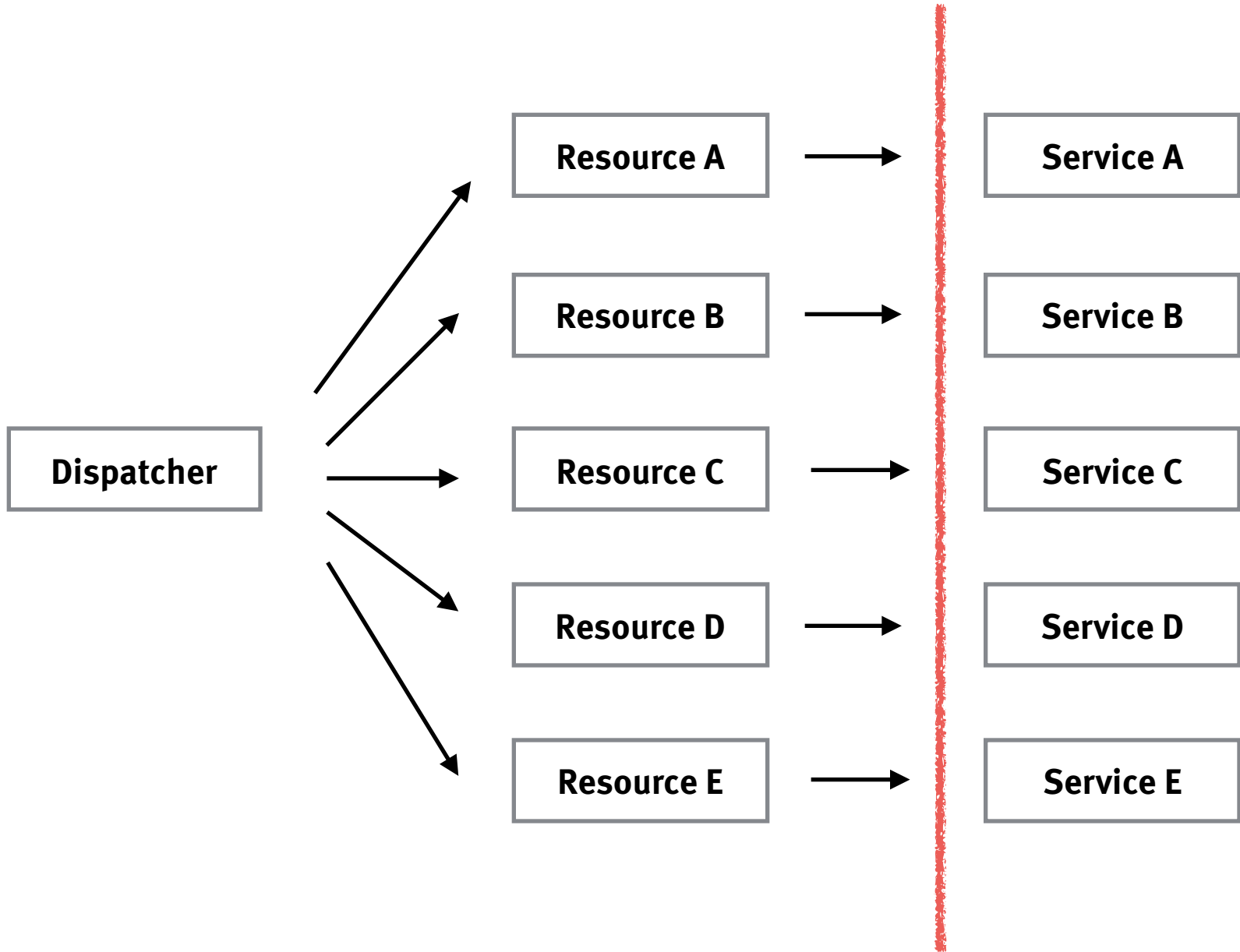


Alternatives to component based frameworks

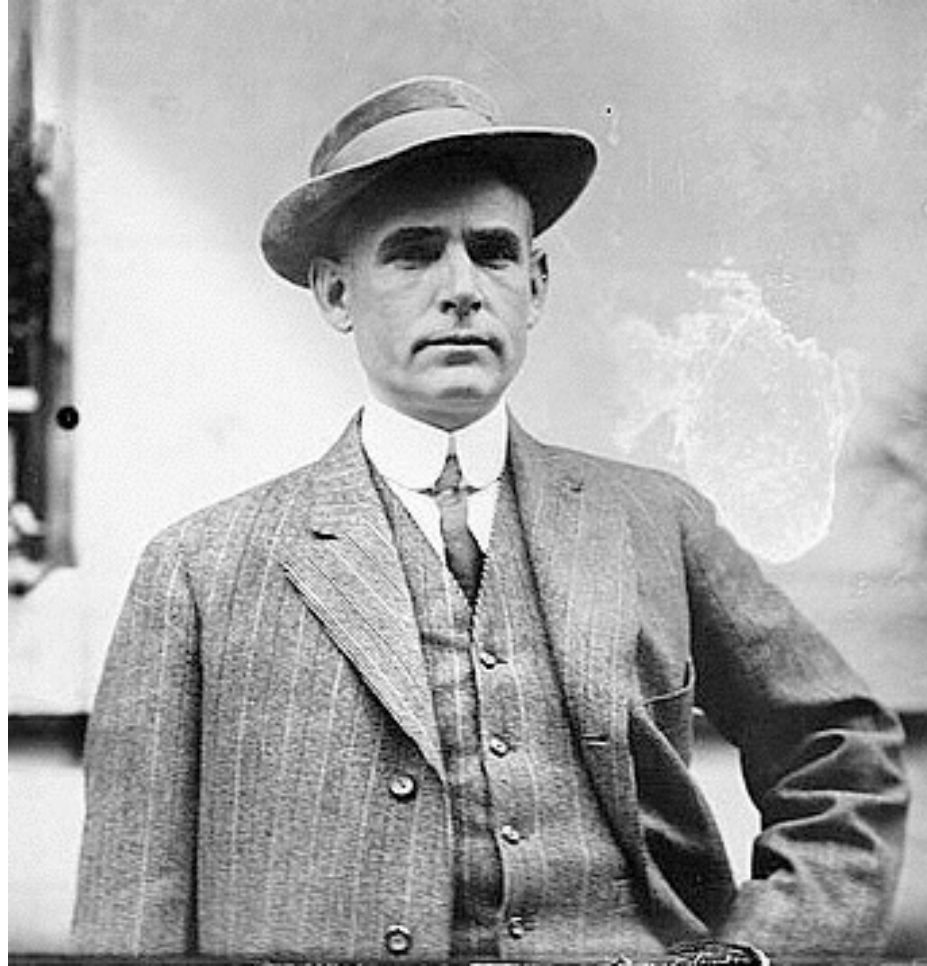








But how to explain it to my manager?

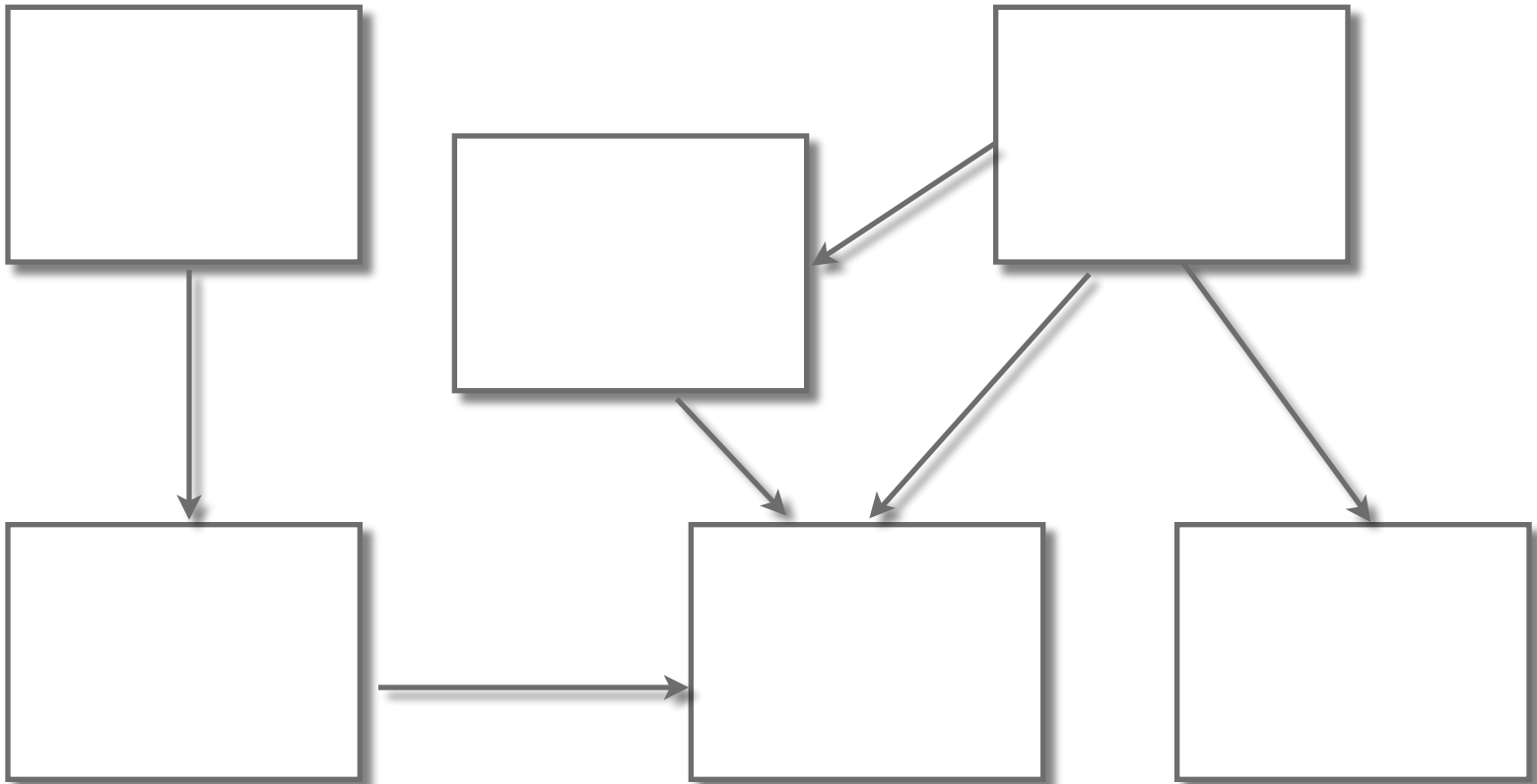


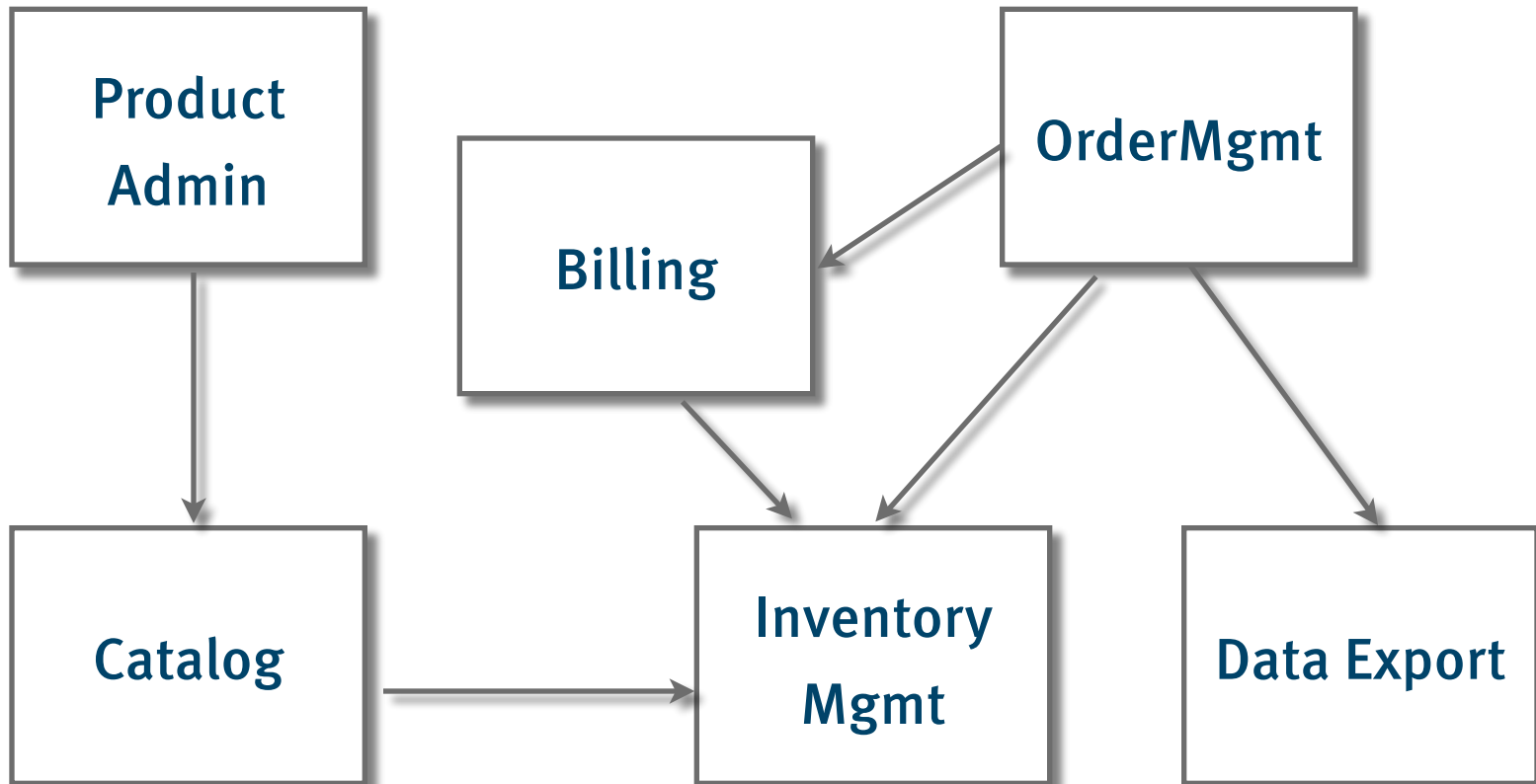
Let's talk about macro architecture!

UI

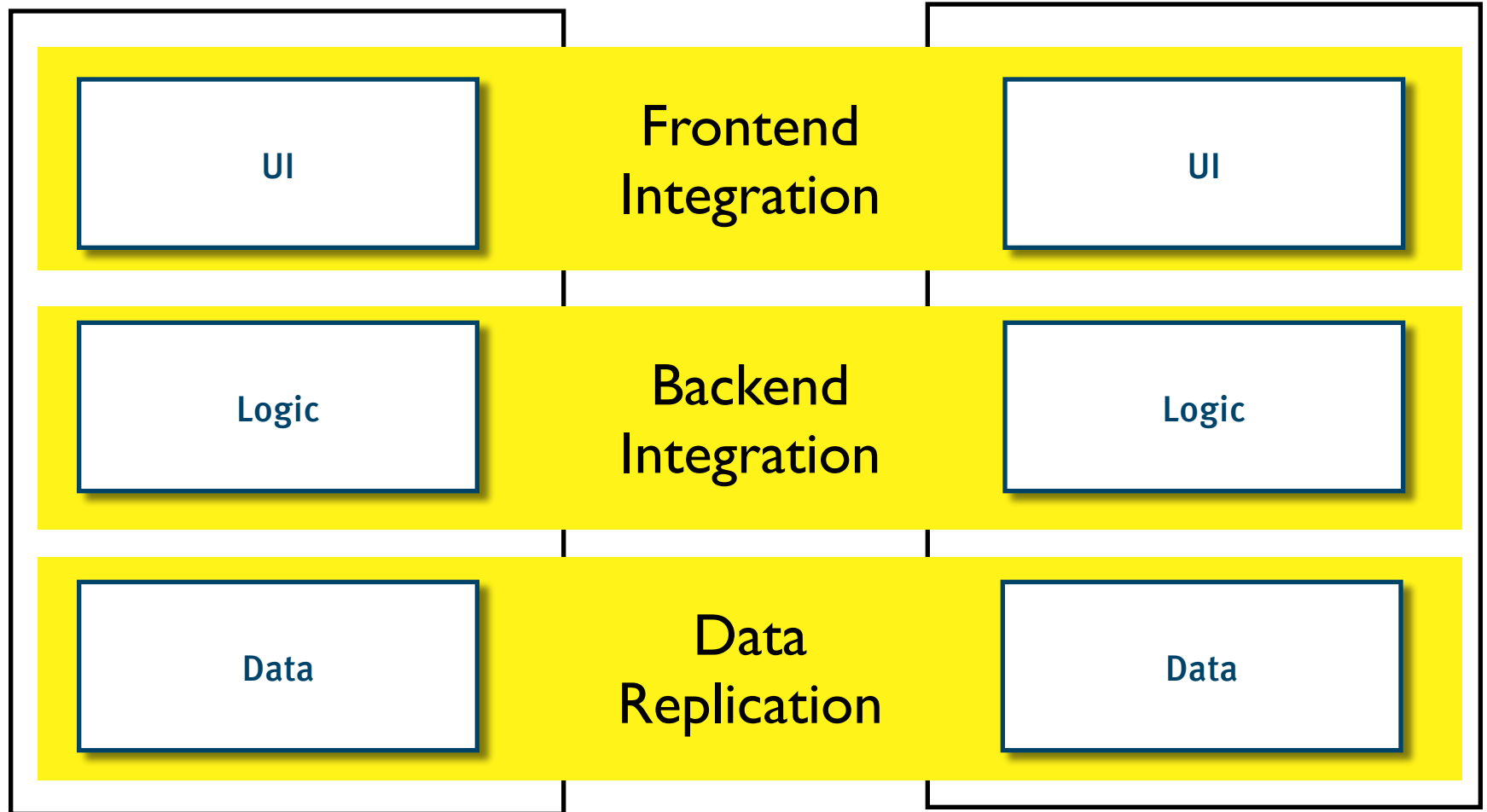
Logic

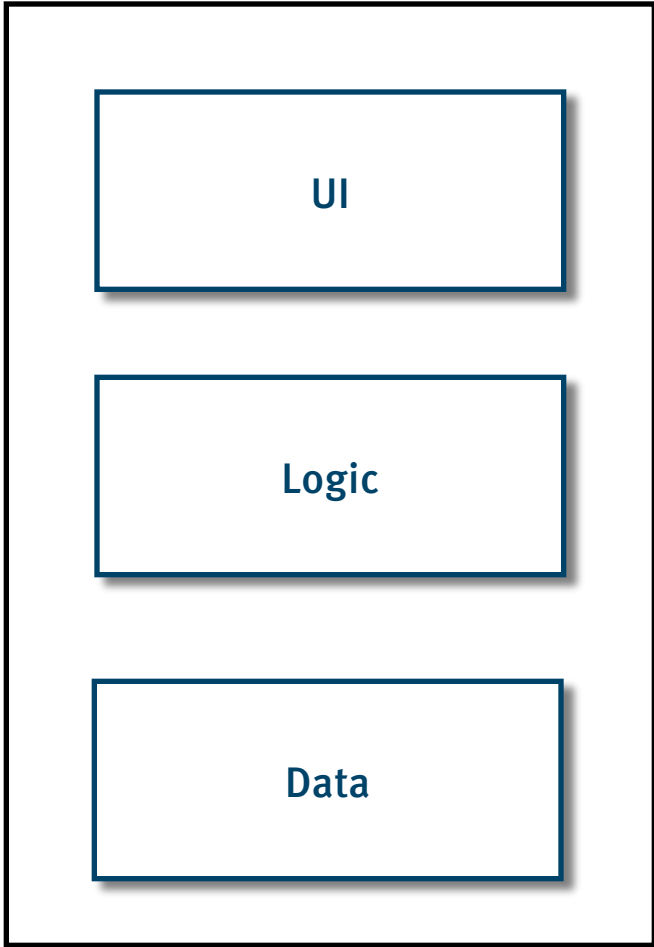
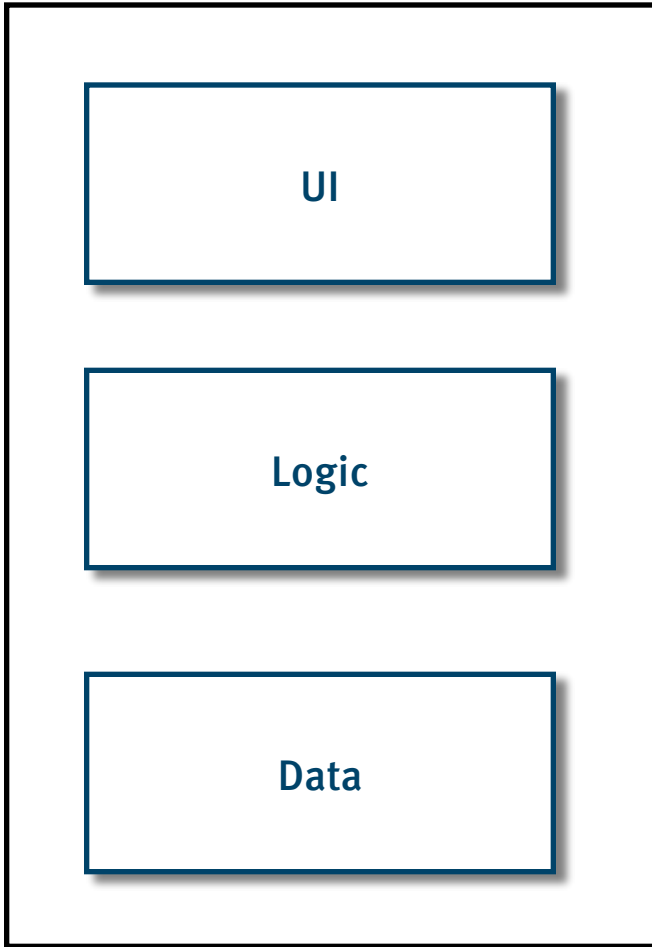
Persistence



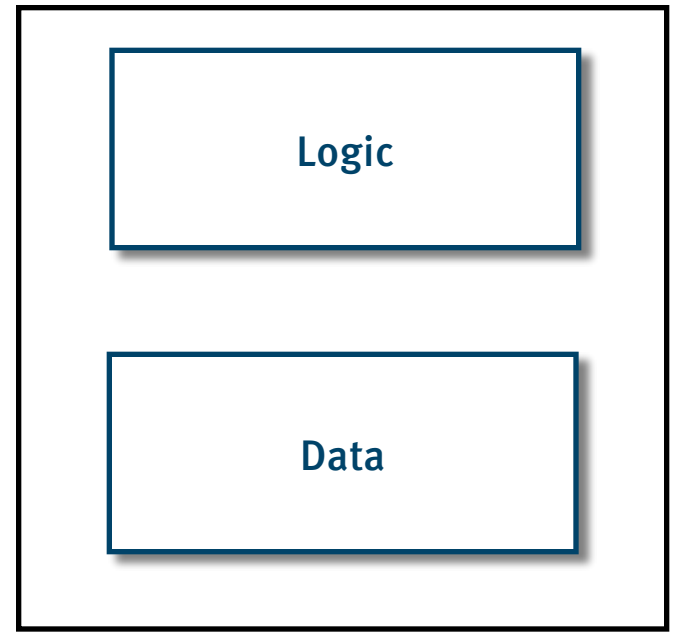
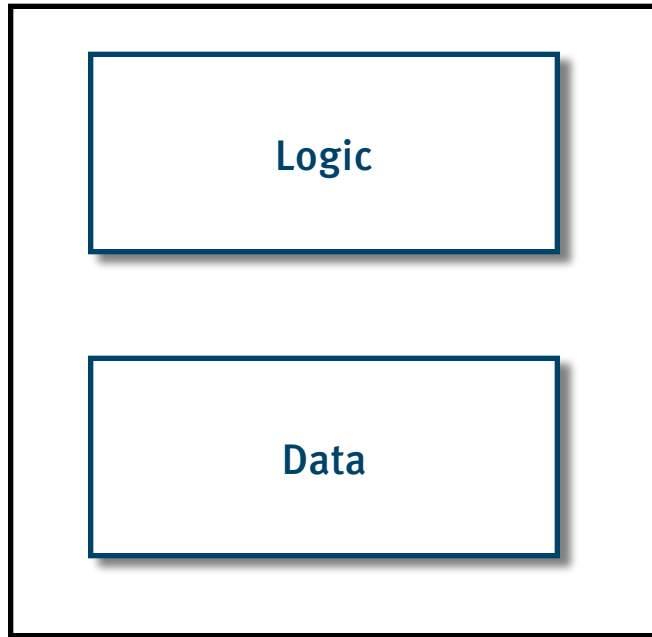
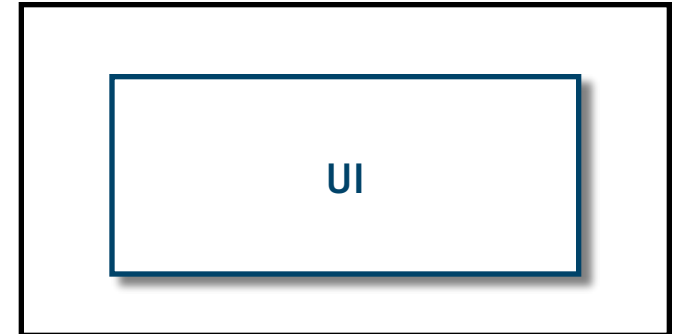
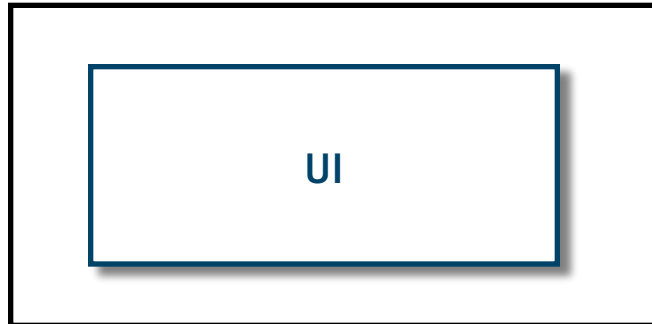


Integration Options

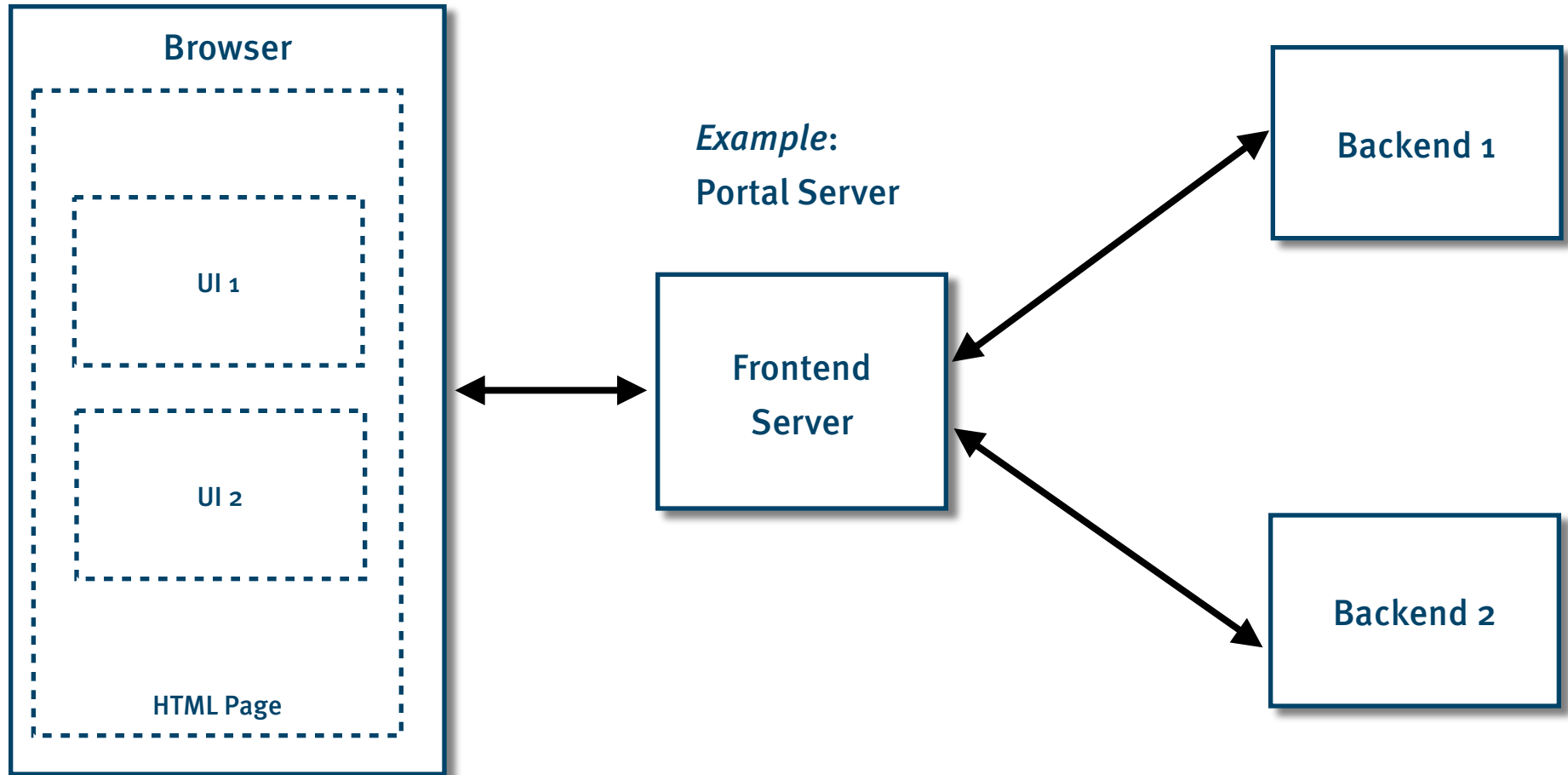




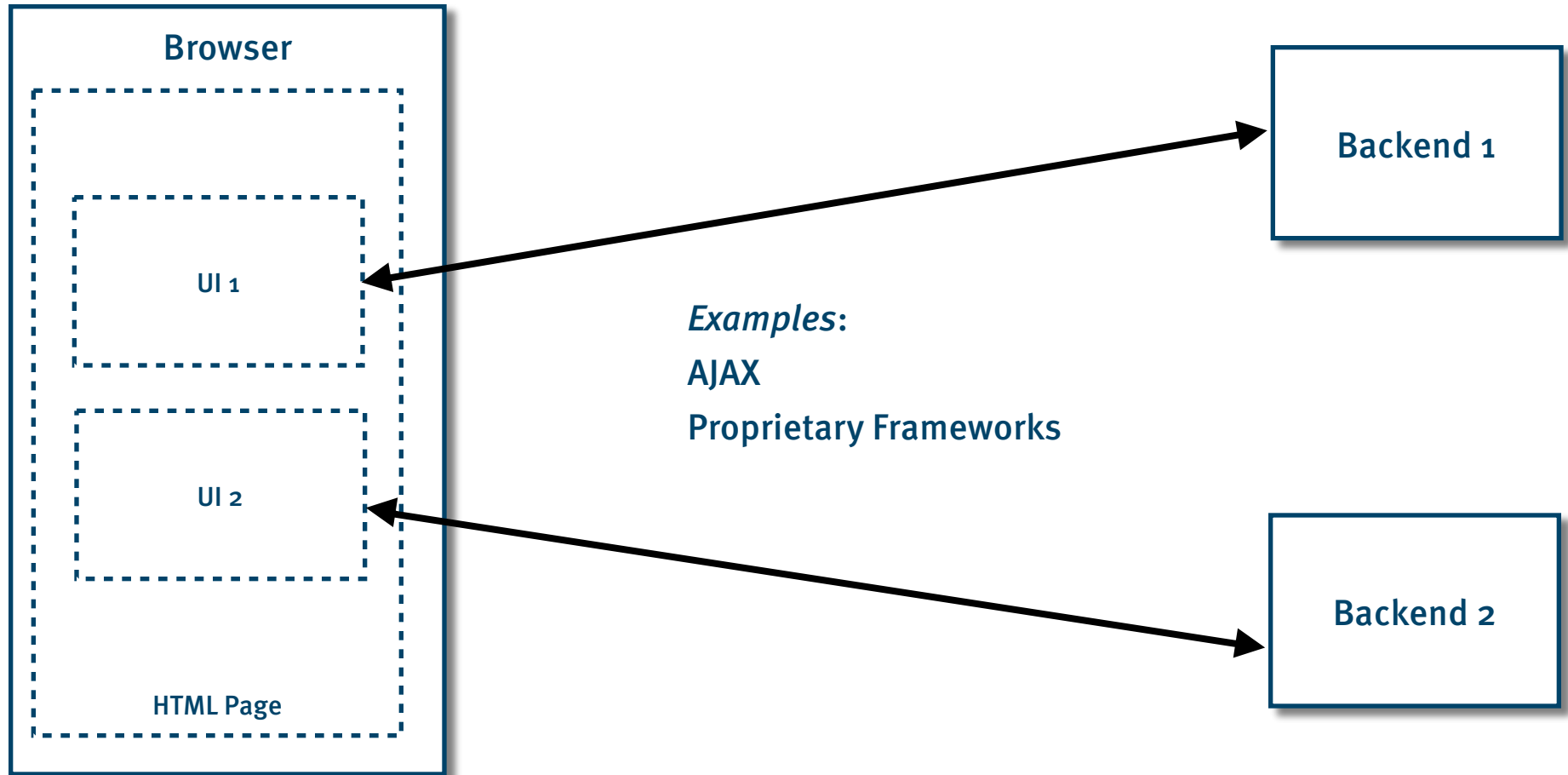
Frontend integration



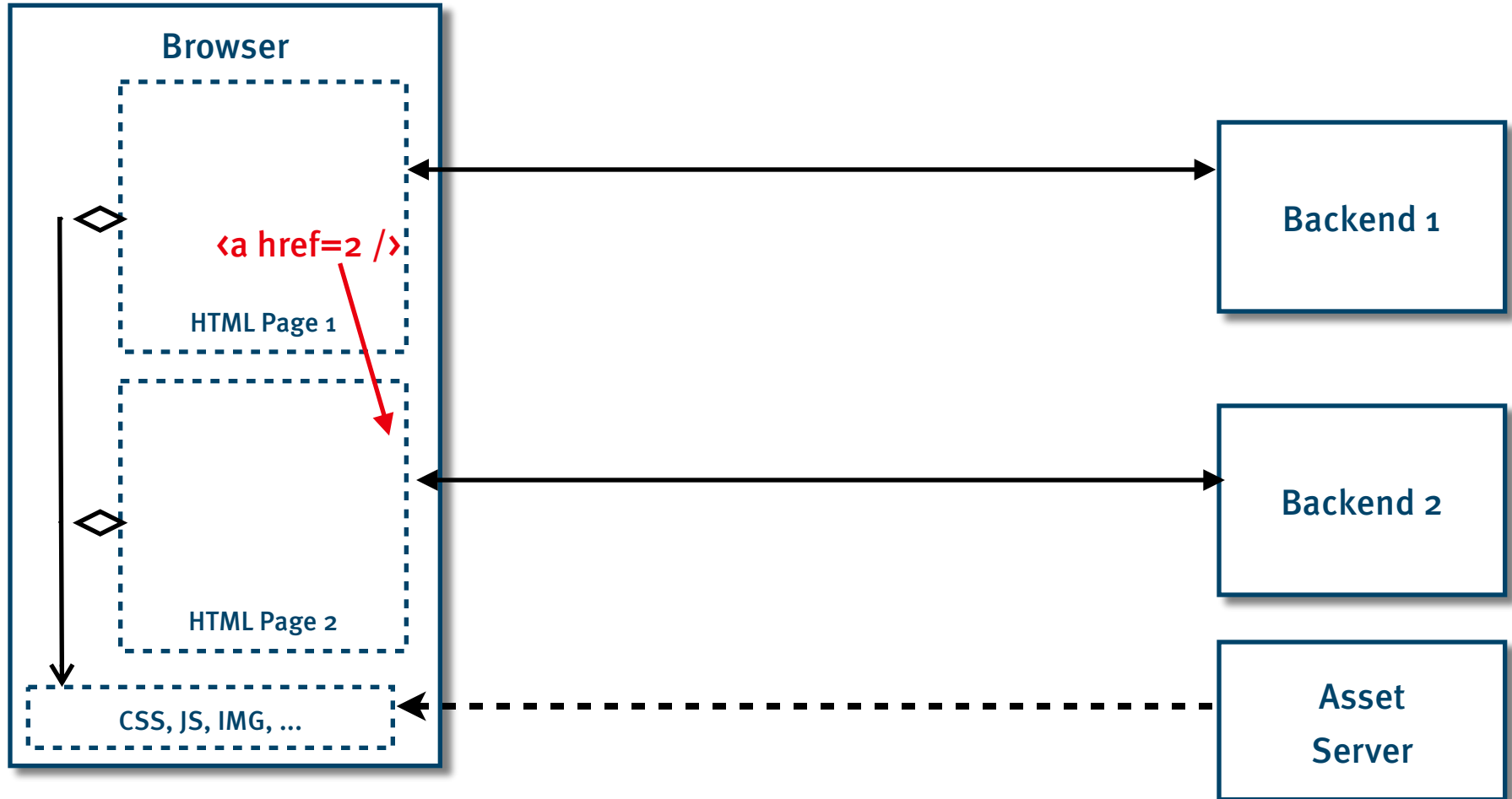
Server-side integration



Client-side integration



Links



Summary

Appreciate the Web

Design for the Web

Respect Boundaries

Embrace the core technologies of the Web



Vielen Dank!

innoQ

Wir lösen das – persönlich!