



Tragfähige Frontend-Architekturen

Till Schulte-Coerne

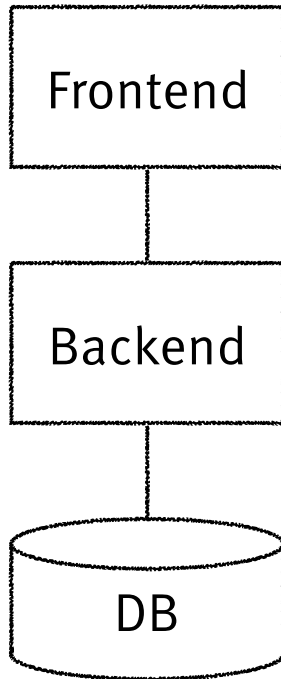
JavaLand 2014

Wir lösen das – persönlich!



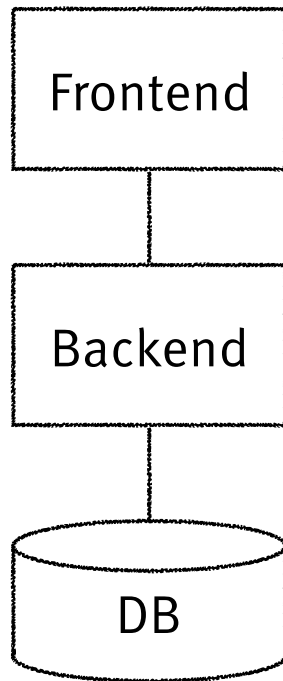
Frontend-Architektur?

Software-Architektur!



Fertig!

Was wo hin?



Rest

Geschäftslogik

Daten

HTML?

CSS?

JavaScript?

PHP?

JSF?

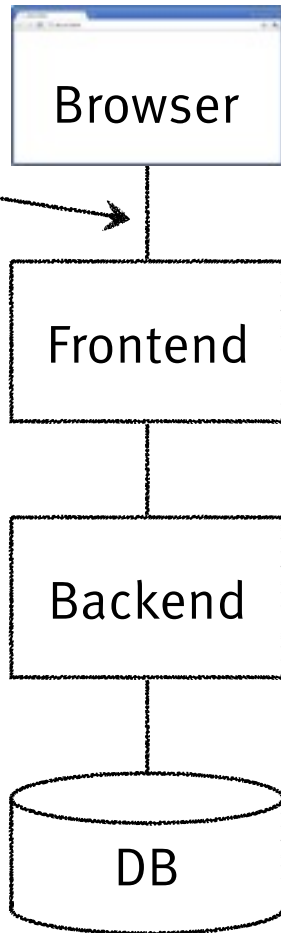
Play?

HTTP?

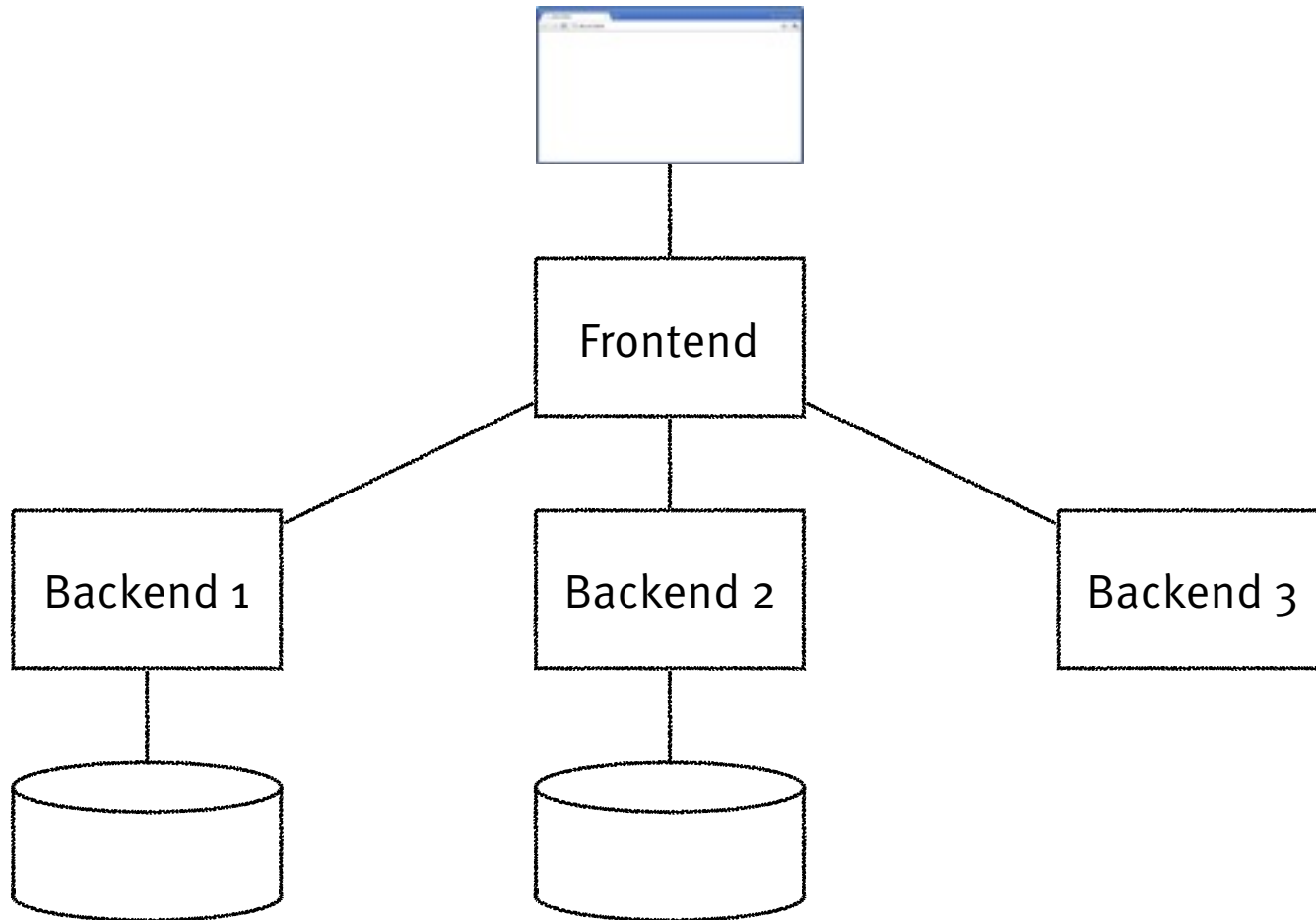
...?

„Realität“

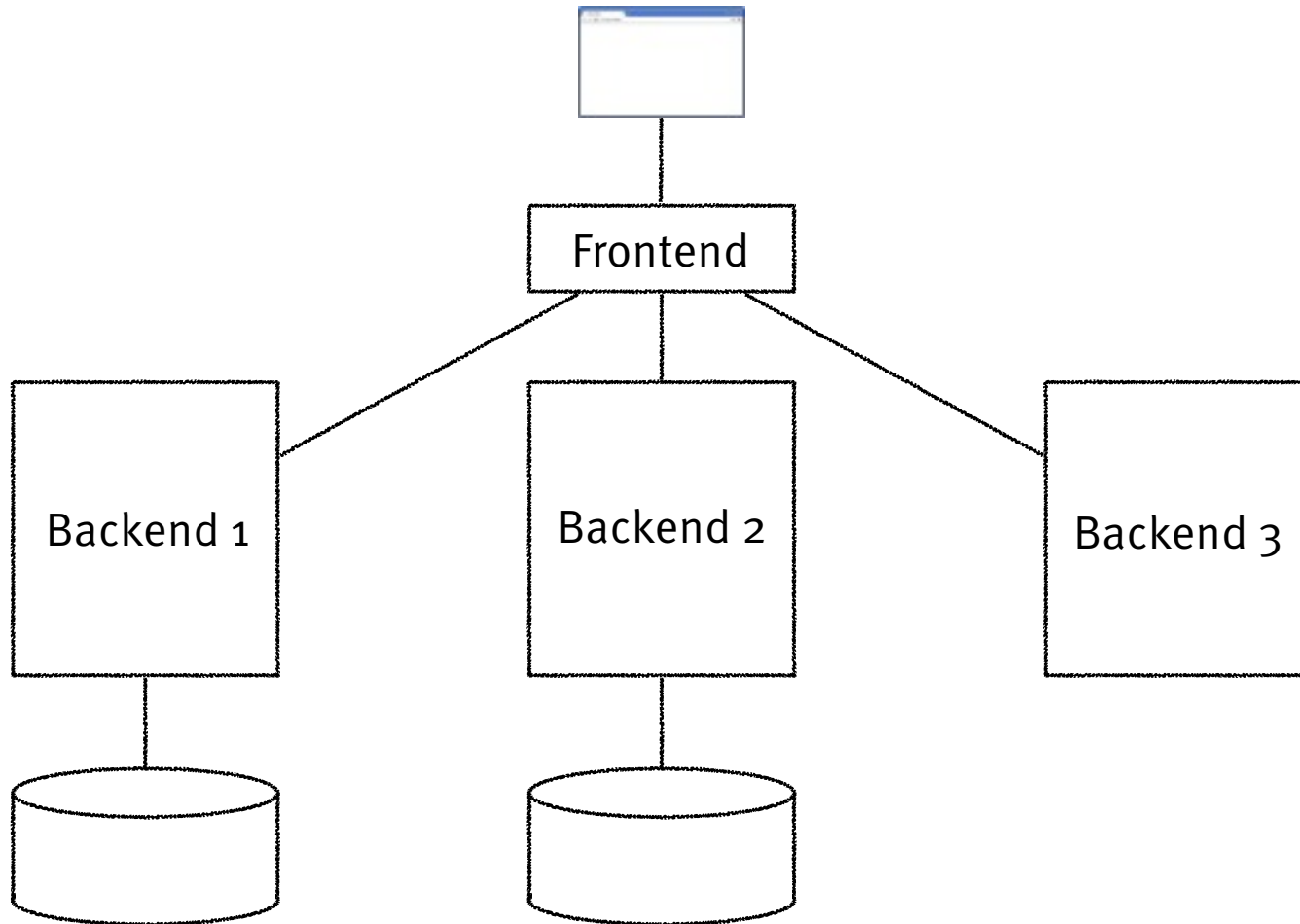
Das „Internet“



Im Großen



Soll



„Klassischer Schnitt“

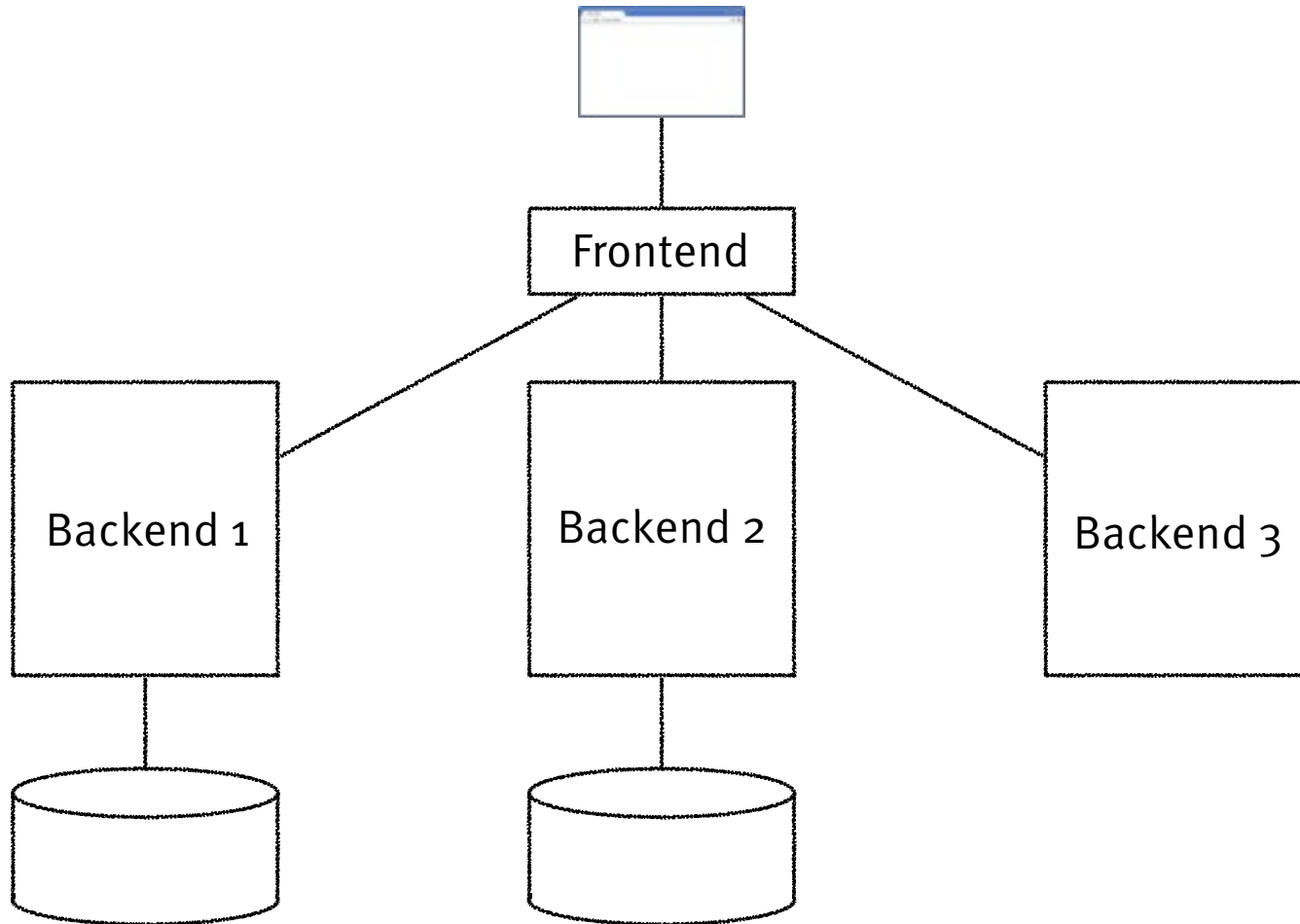
Klassisches Enterprise-Modell

Fette Backend-Services ?

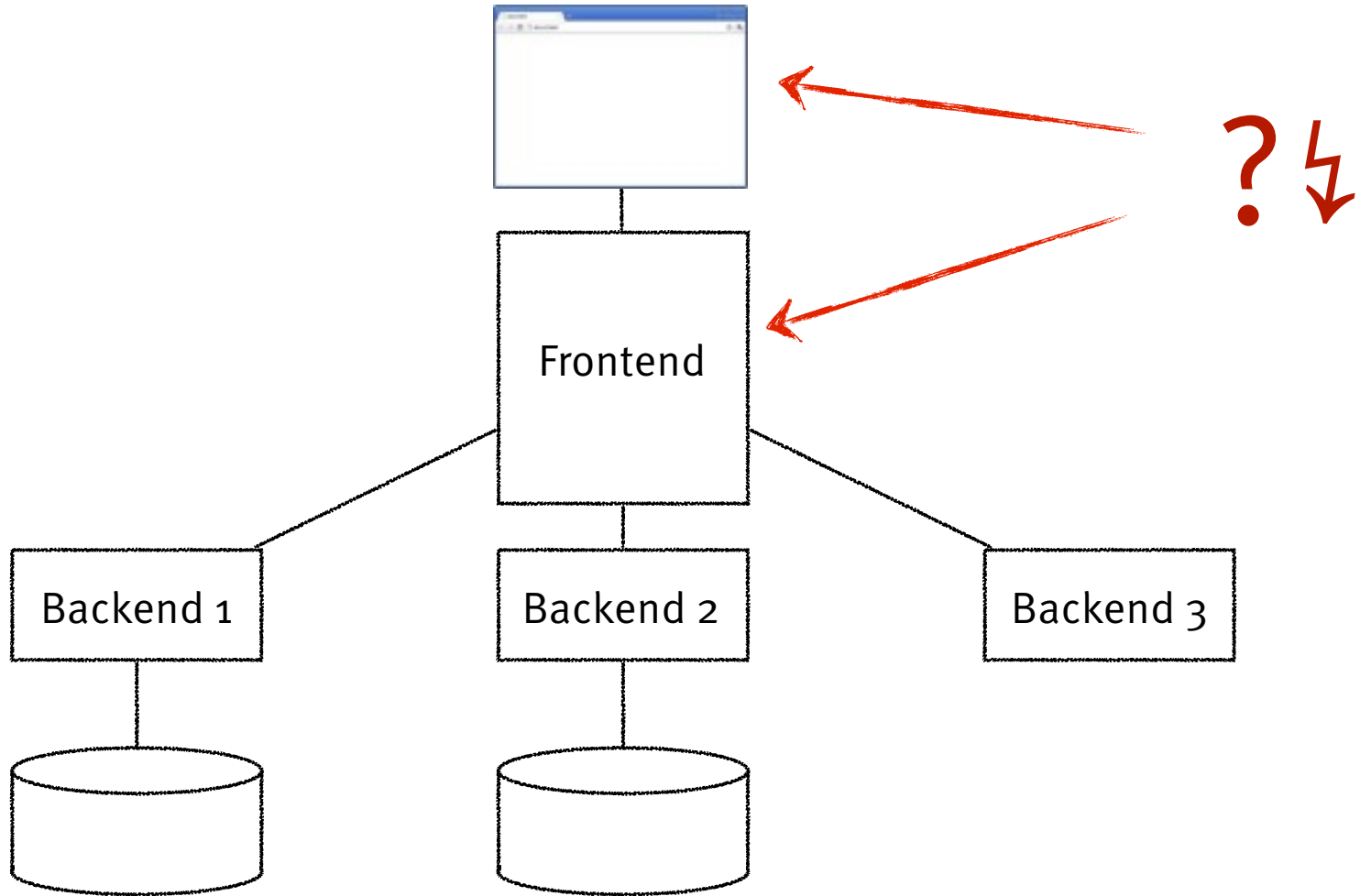
Wenig Frontend-Logik

Browser == Notweniges Übel

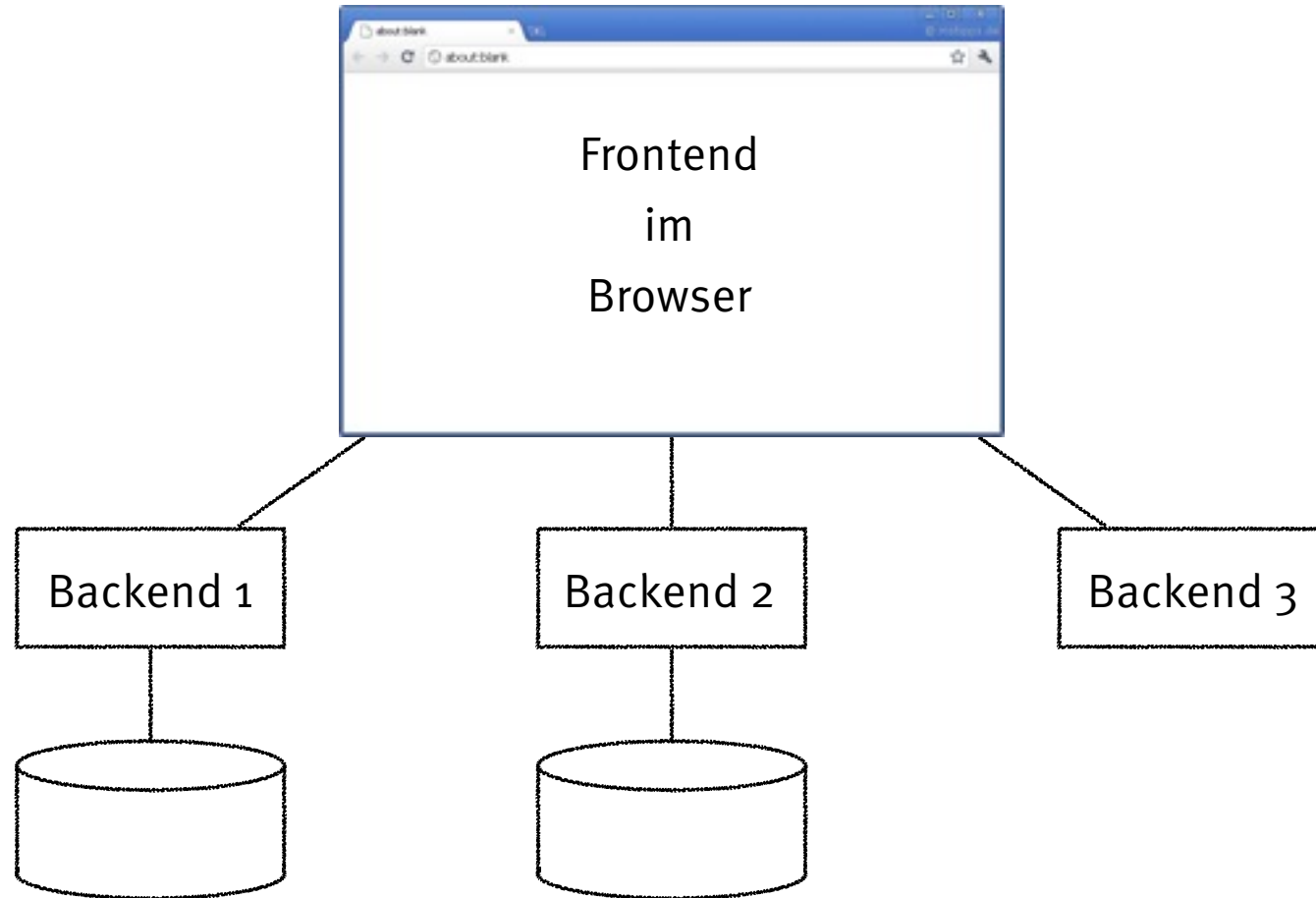
Soll



Ist



Logische Weiterentwicklung?



Single Page Apps

Rich Client

~~MVC, MVP oder MVVM~~

~~MVC oder MVP~~

~~MVC~~ im Browser

Mögliche Features

Data-Binding

Templating

**Applikationslogik
(Controller)**

Client-seitiges Routing

**Backend-Services per Ajax
(Modell)**

Backend-Services

Ajax + X

```
$.get("/foo.json", function(e) {  
  machWas(e.data);  
});
```

„plain jQuery“

Ember Data

```
App.Post = DS.Model.extend({  
  title: DS.attr('string'),  
  content: DS.attr('string'),  
  comments: DS.hasMany('App.Comment')  
});
```

```
App.Comment = DS.Model.extend({  
  author: DS.attr('string'),  
  content: DS.attr('string'),  
  post: DS.belongsTo('App.Post')  
});
```

Data-Binding

KnockoutJS

```
var myViewModel = {  
  personName: ko.observable('Bob'),  
  personAge: ko.observable(123)  
};
```

Hallo !

```
myViewModel.personName('Mary');
```

Templating

KnockoutJS

```
<div data-bind="template: { name: 'person-template', foreach: people }" />
```

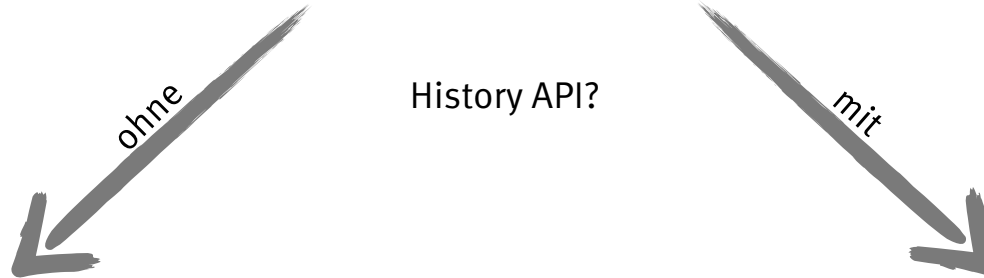
...

```
<script type="text/html" id="person-template">  
  <h3 data-bind="text: name"></h3>  
  <p>Age: <span data-bind="text: personAge"></span></p>  
</script>
```

Routing

```
$routeProvider.  
  when('/', {controller:ListCtrl, templateUrl:'list.html'}).  
  when('/edit/:projectId', {controller:EditCtrl, templateUrl:'detail.html'}).  
  when('/new', {controller:CreateCtrl, templateUrl:'detail.html'}).  
  otherwise({redirectTo:'/'})
```

AngularJS



http://example.com/index.html#/
http://example.com/index.html#/edit/4711
http://example.com/index.html#/new

http://example.com/
http://example.com/edit/4711
http://example.com/new

Frameworks und Bibliotheken

	Typ	Modelle	Routing	Templating	Data binding
AngularJS	Framework	Ressourcen	Optional	Eingebaut	Zentral
KnockoutJS	Bibliothek	—	Optional	Eingebaut	Zentral
EmberJS	Framework	Objekte	Zentral	Eingebaut	Re-Render
BackboneJS	Bibliothek	Ressourcen	Optional	Wahlweise	Re-Render
...					

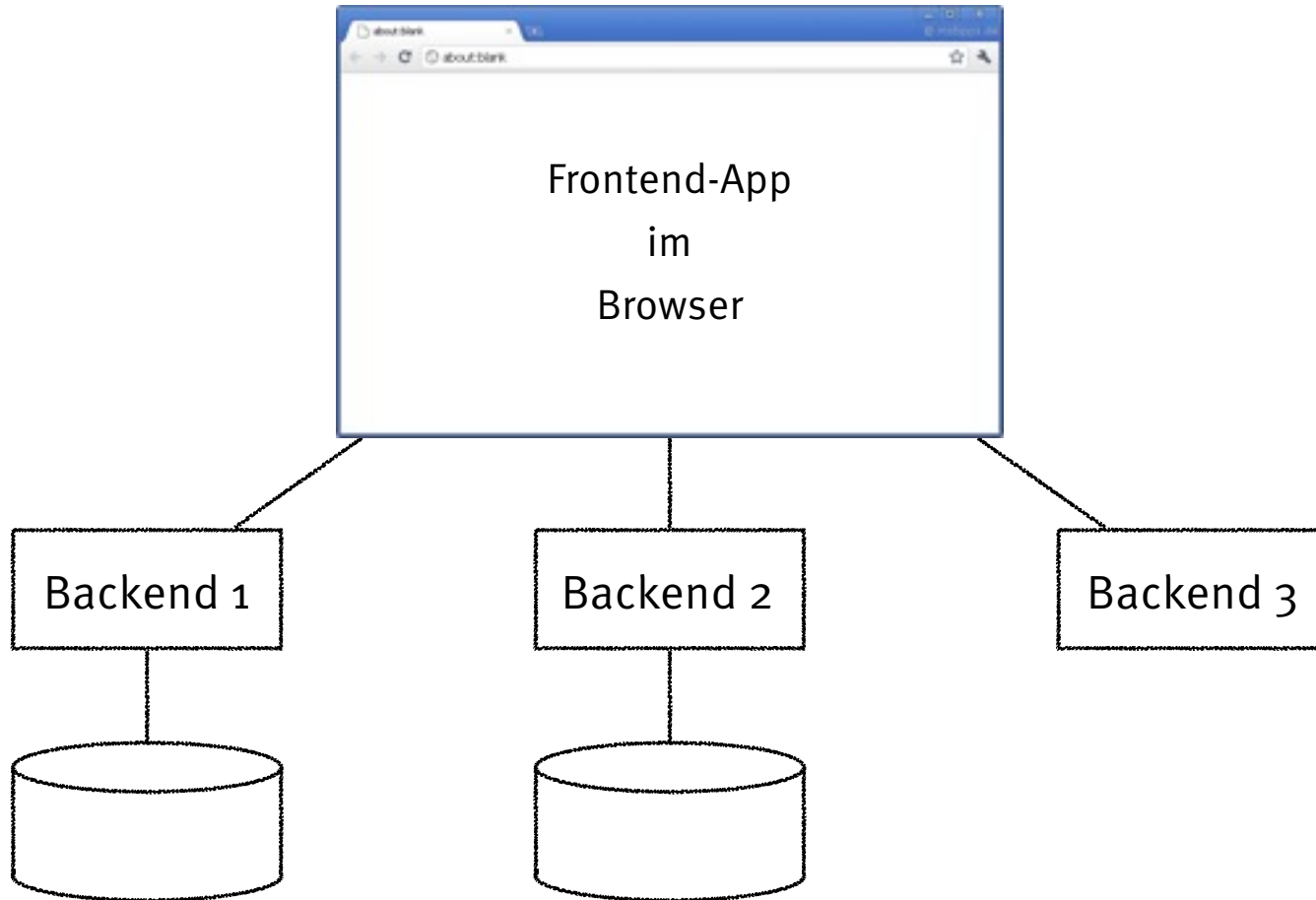
Single Page Apps

Neues Enterprise-Modell?

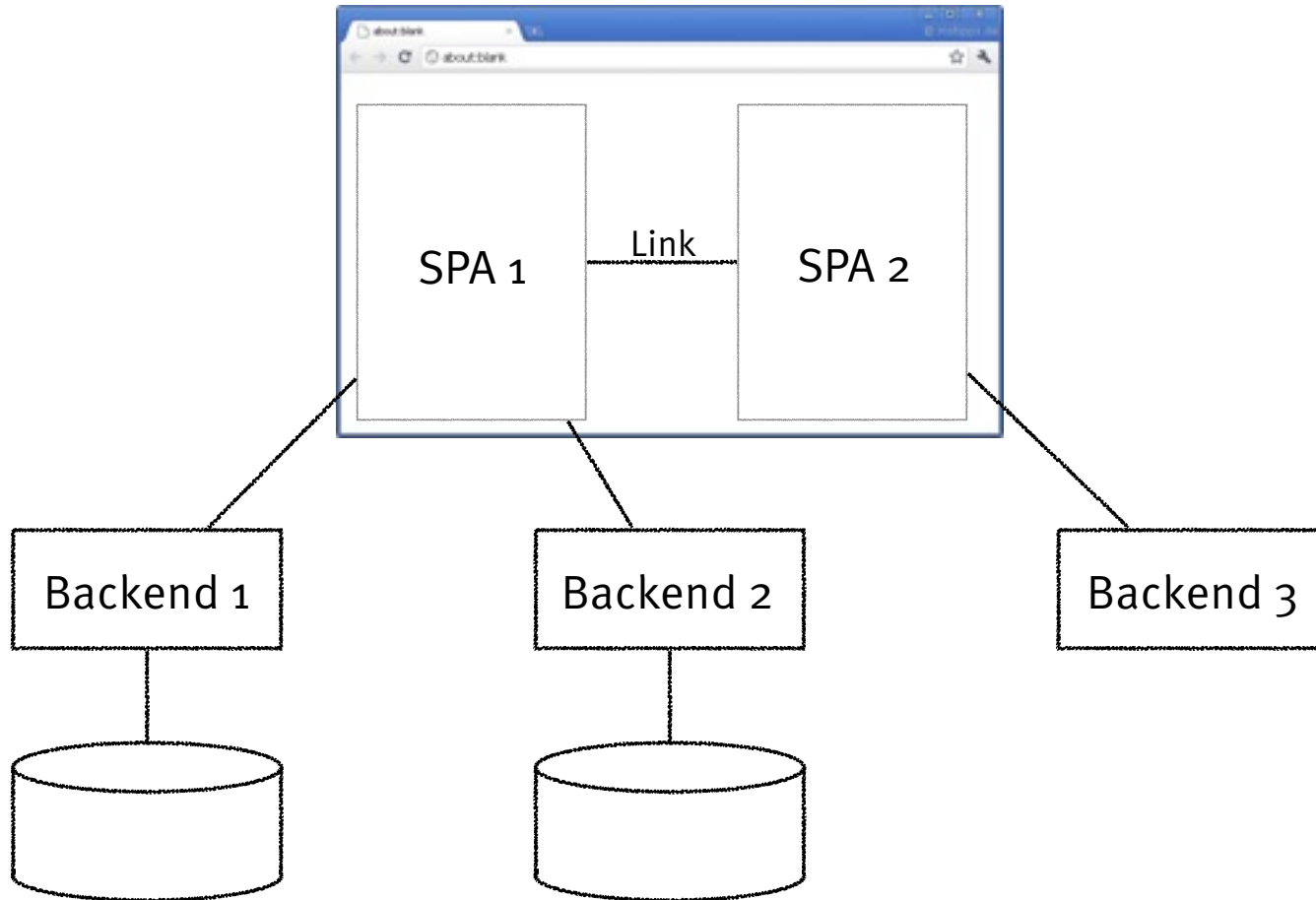
Tragfähige Architektur

Kein aufgeteiltes Frontend

Single Page Apps



Multi Single Page Apps



Zurück zum Anfang



Browser



Frontend



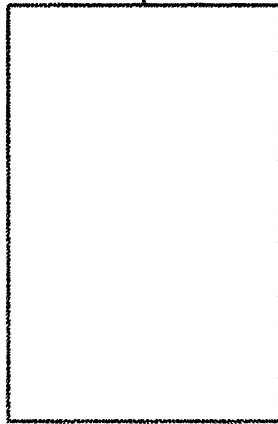
Backend



Daten



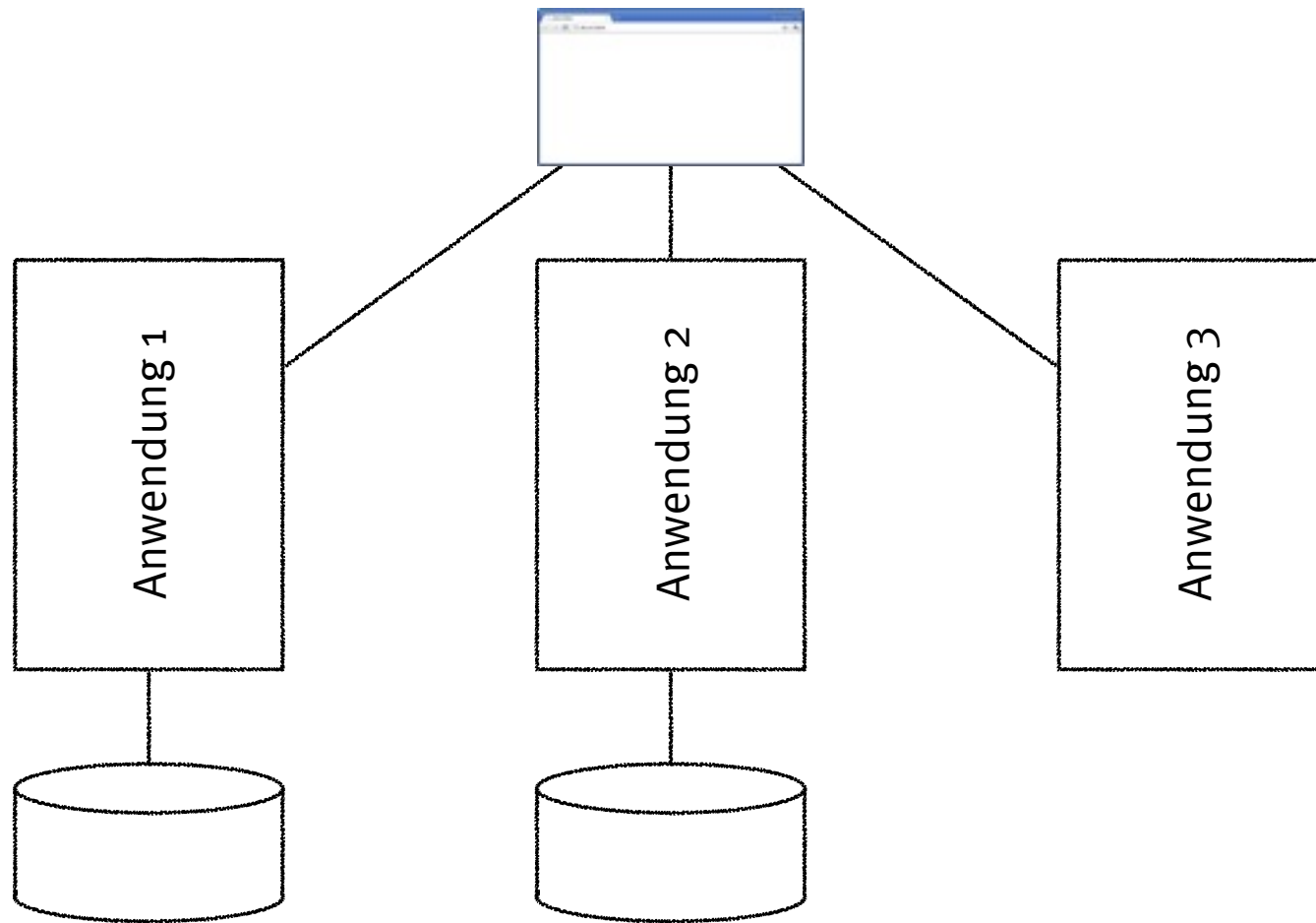
Browser



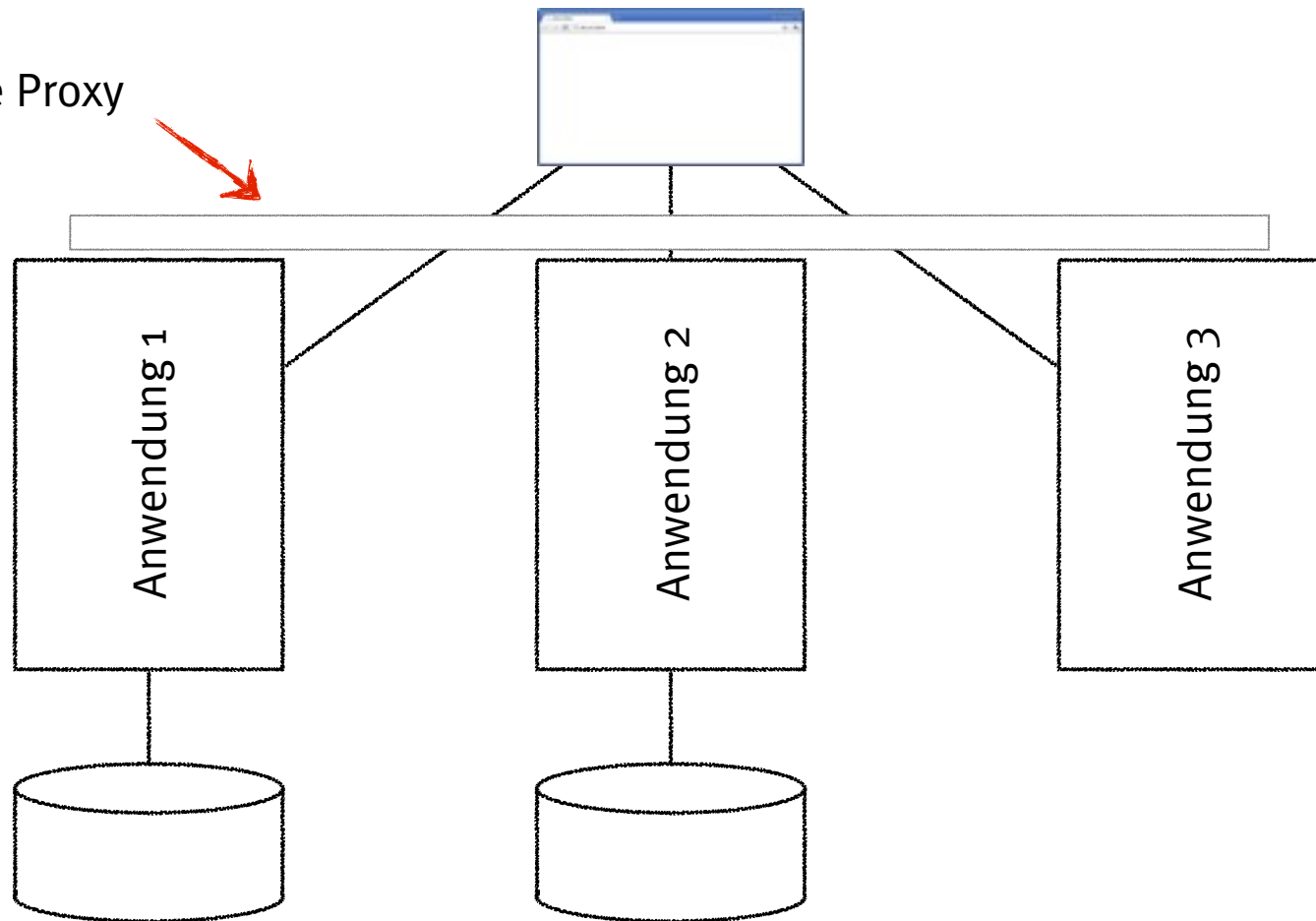
Backend



Daten



Reverse Proxy



ROCA

RESTful Web-Frontends

Deeplinks

„Nahtlose“ Übergänge

Zustandslosigkeit

Applikationslogik ausschließlich auf dem Server

Vertraue niemals dem Client

Nicht-Browser Clients (z.B. Google Crawler)

Keine Duplikation von Logik auf dem Client

Sämtliche Applikationslogik auf dem Server

+

Keine Duplikation von Logik auf den Client

=

Keine Applikationslogik auf dem Client!

Unobtrusive JavaScript verwenden

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

Ein Link

```
</a>
```



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

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

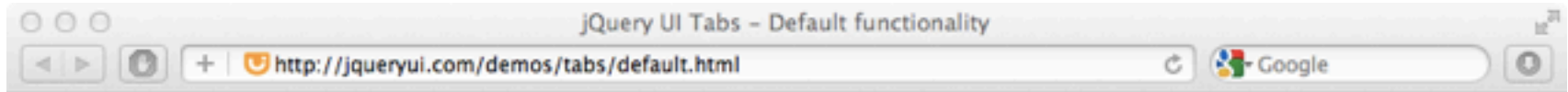
+

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

```
<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>
</div>
```

+

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



Nunc tincidunt

Proin dolor

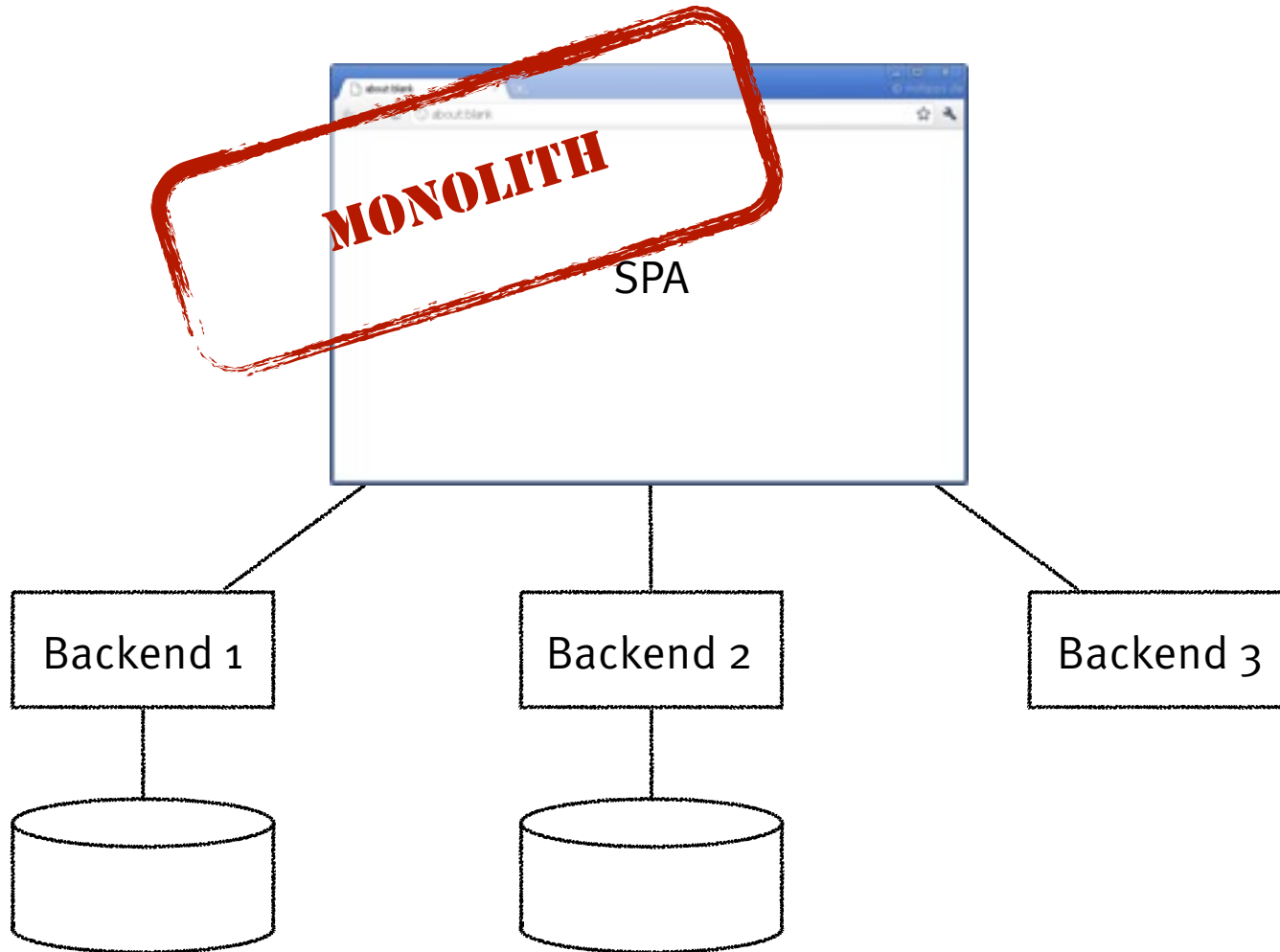
Aenean lacinia

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.

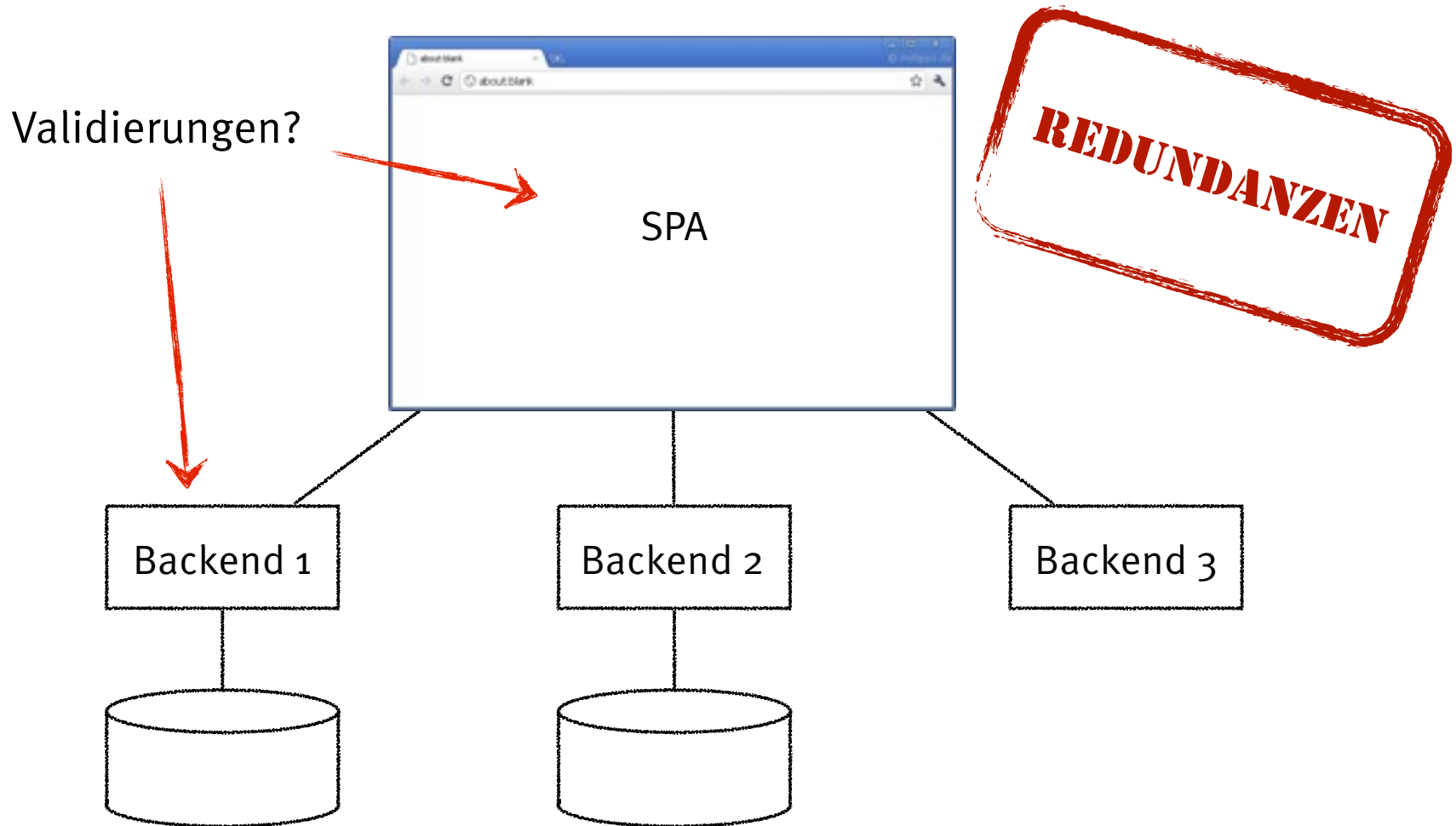
SPA vs. ROCA

Single Page Apps

Nachteile SPA



Nachteile SPA

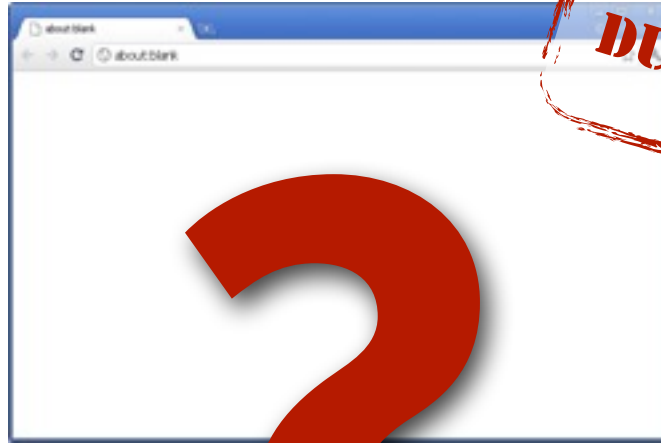


Nachteile SPA

Google

bing

YAHOO!



**KEINE „NATIVE“
DURCHSUCHBARKEIT**



elasticsearch.

Backend 1

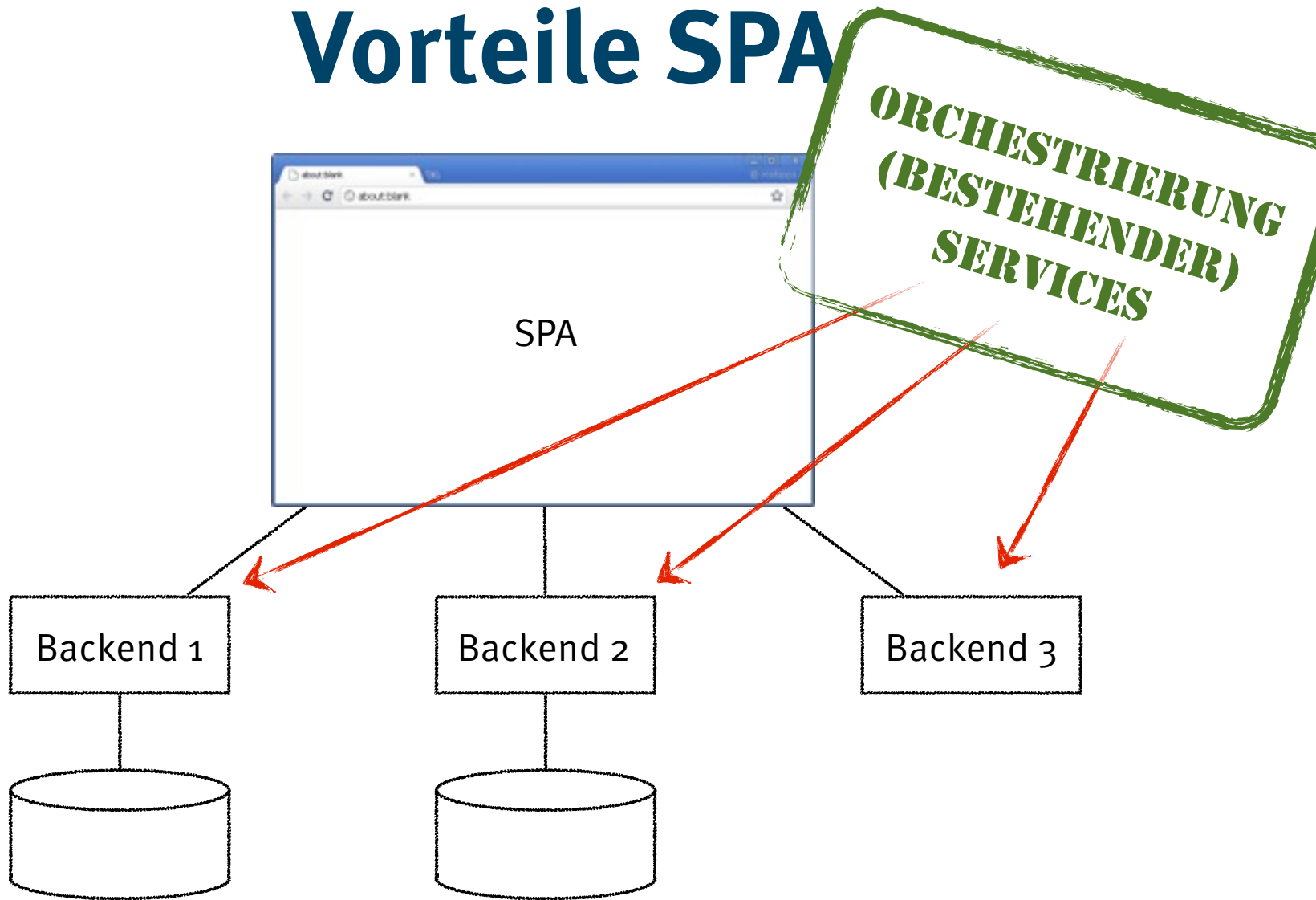


Backend 2



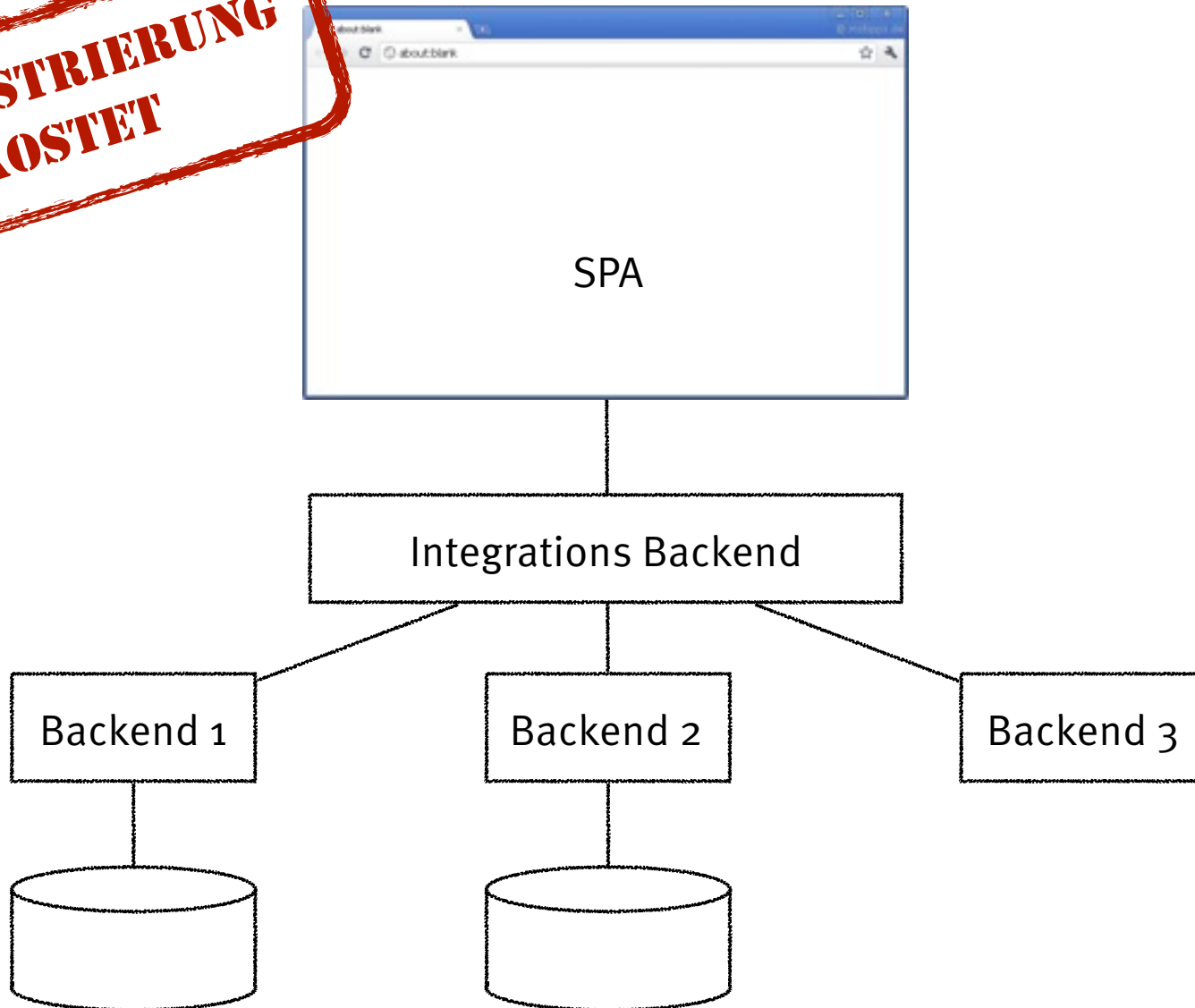
Backend 3

Vorteile SPA



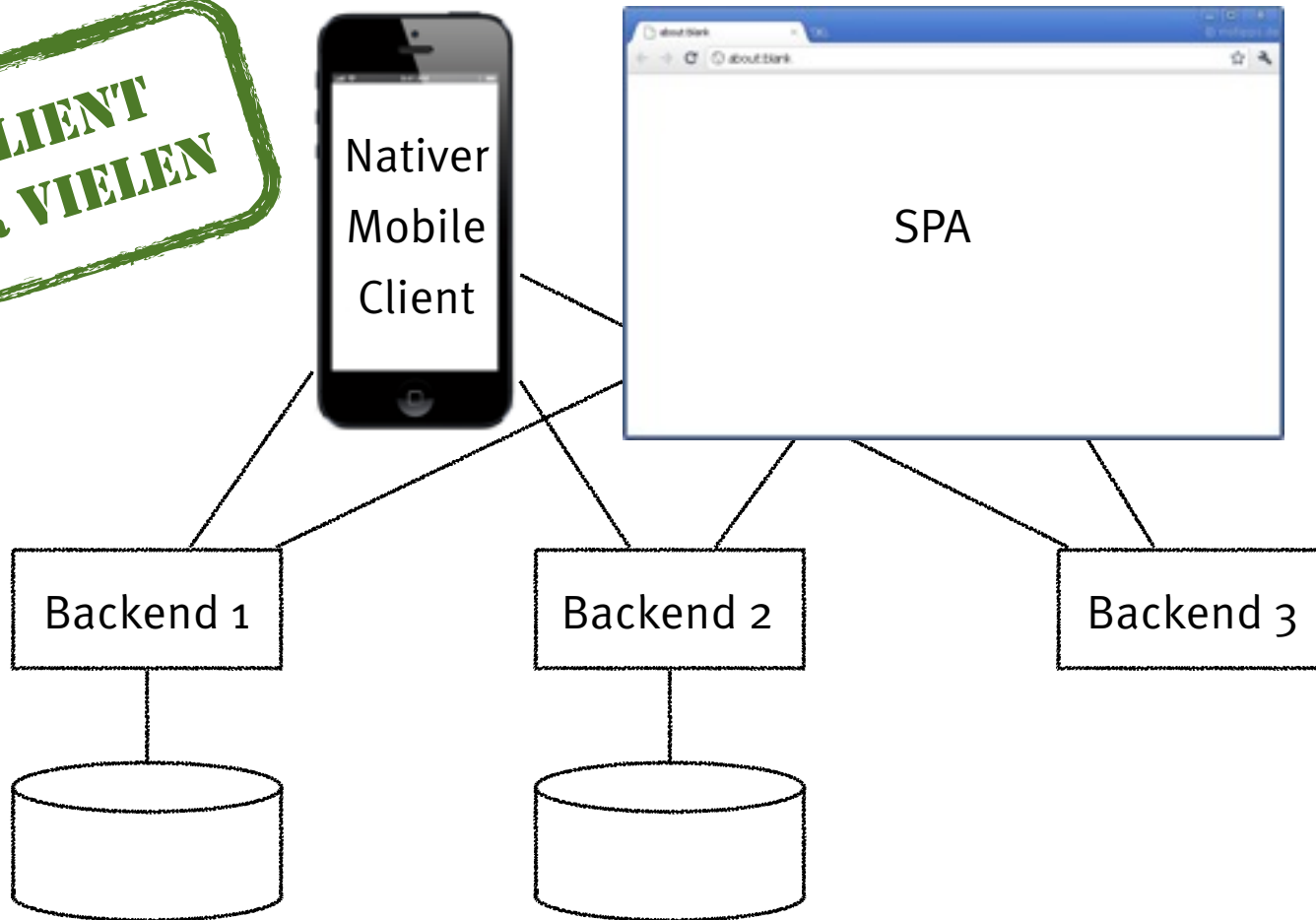
Vorteile SPA

**ORCHESTRIERUNG
KOSTET**

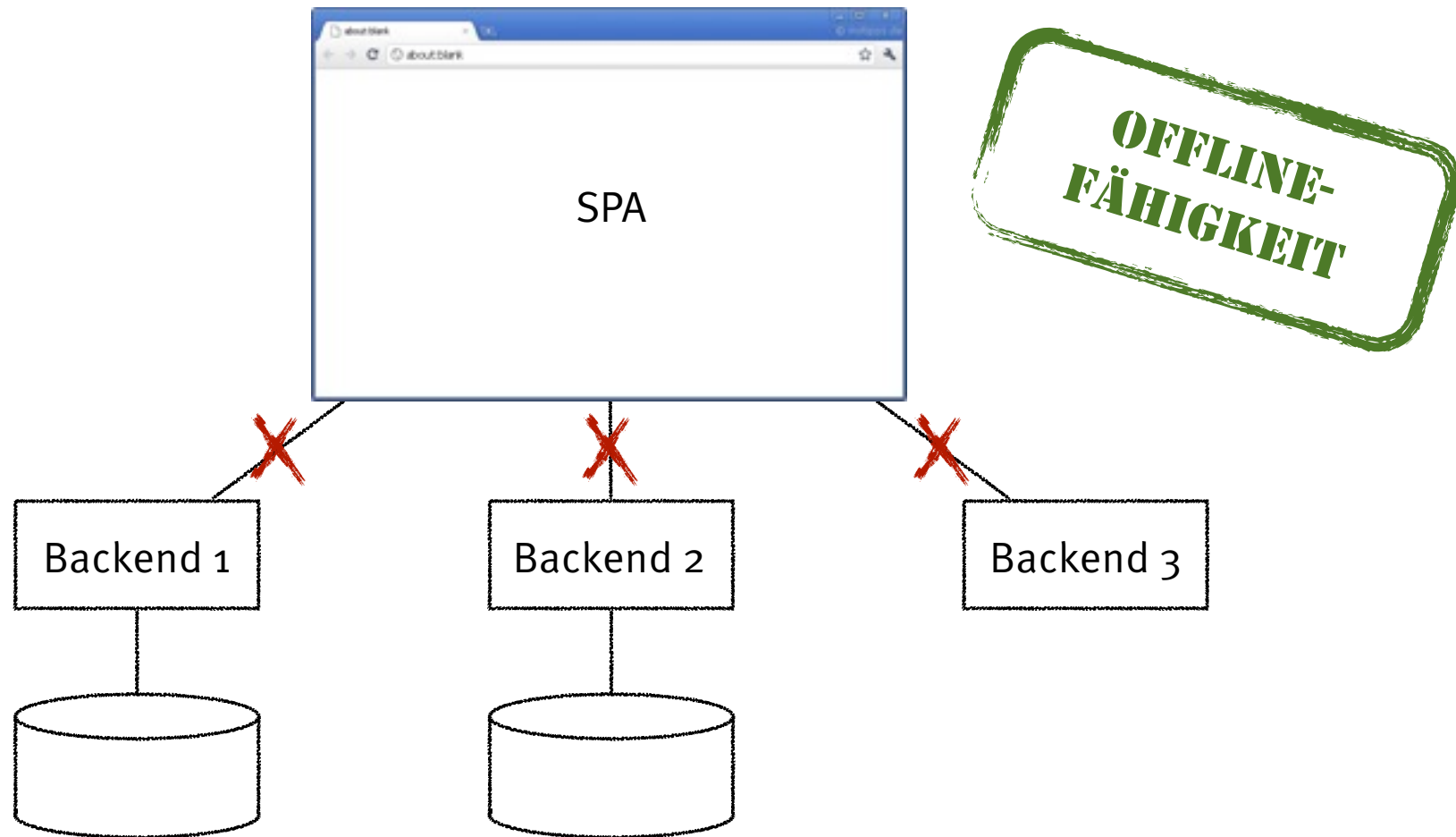


Vorteile SPA

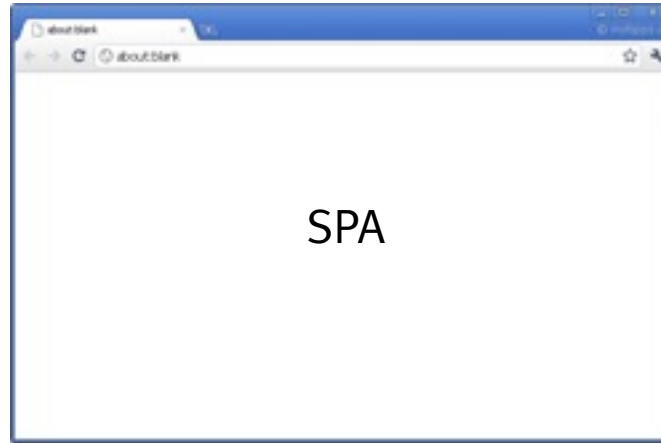
**EIN CLIENT
UNTER VIELEN**



Vorteile SPA

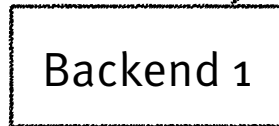


Vorteile SPA

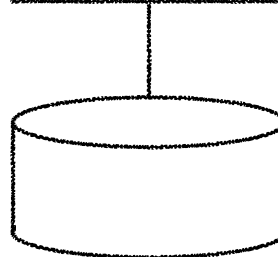


JavaScript

Scala



Backend 2



PHP

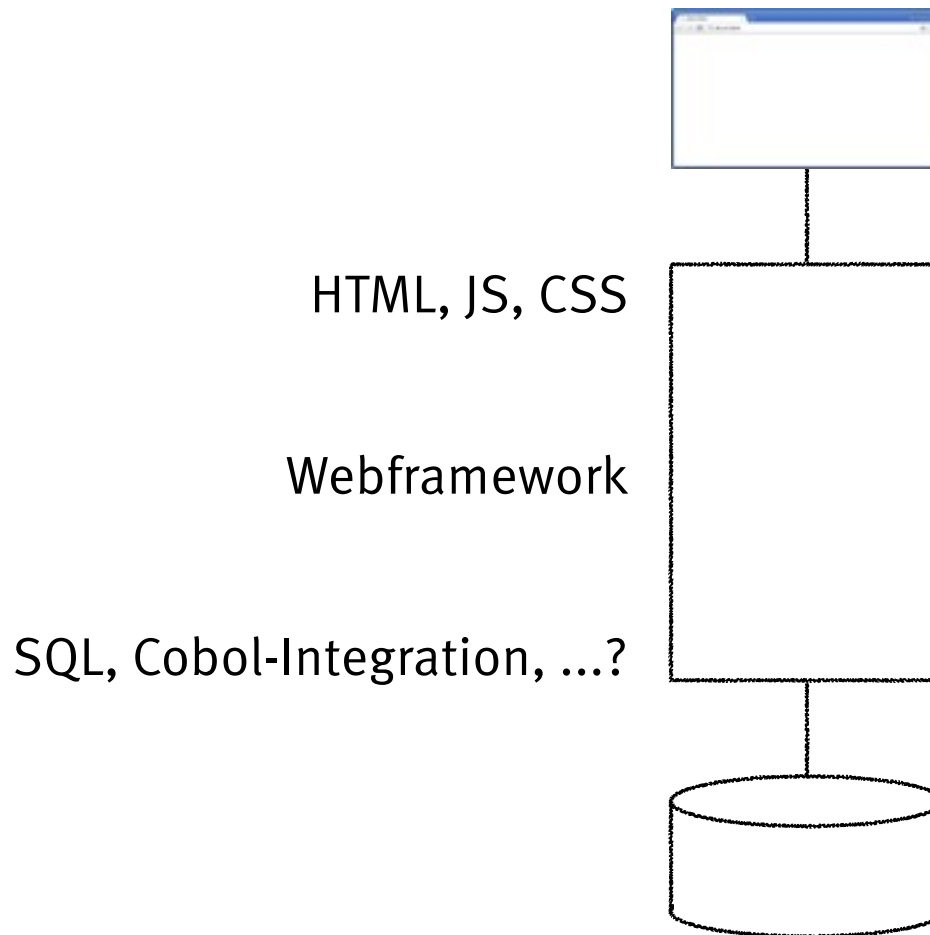
Backend 3

Java

ROCA

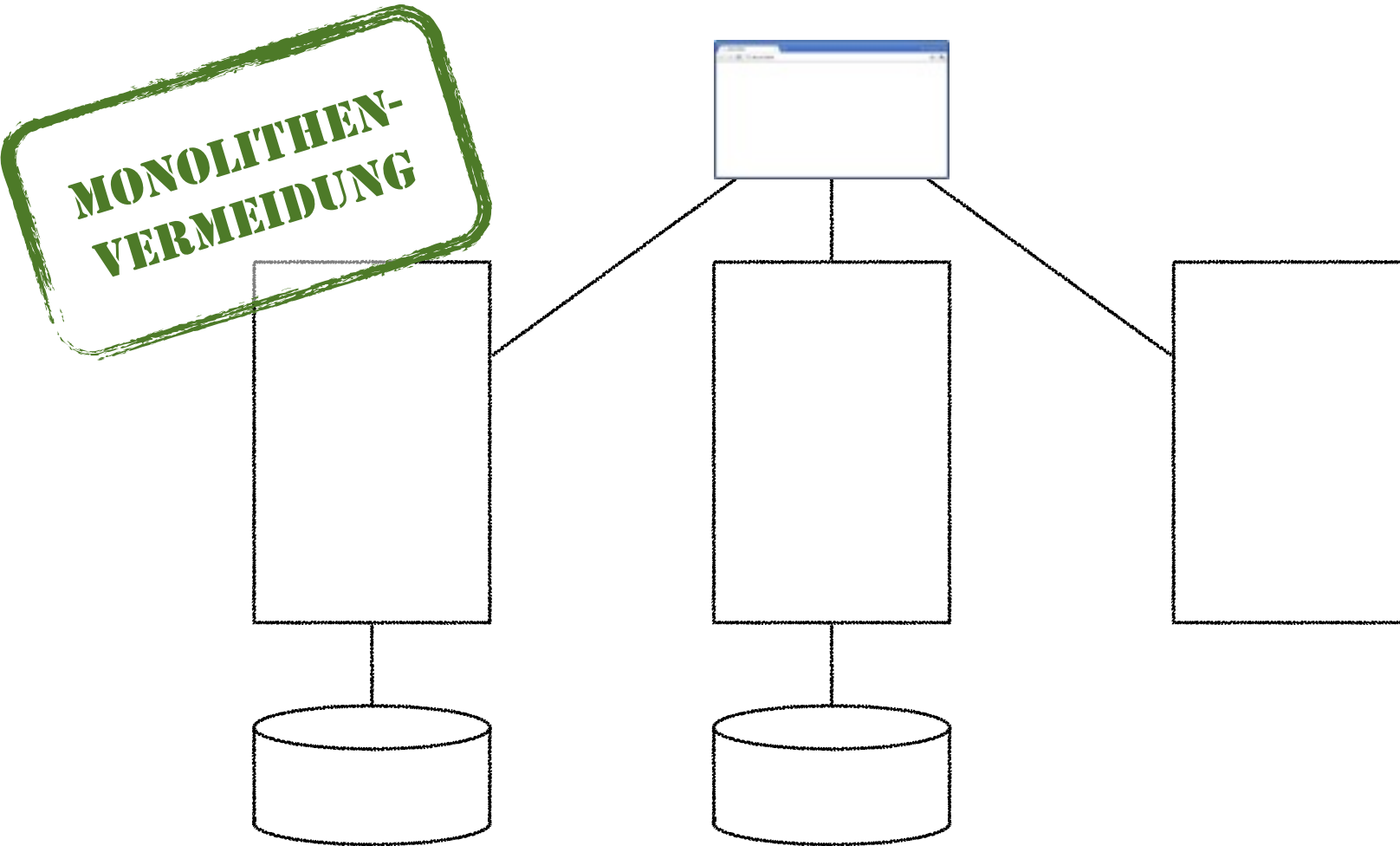
Vertikalen-Architektur

Nachteile ROCA

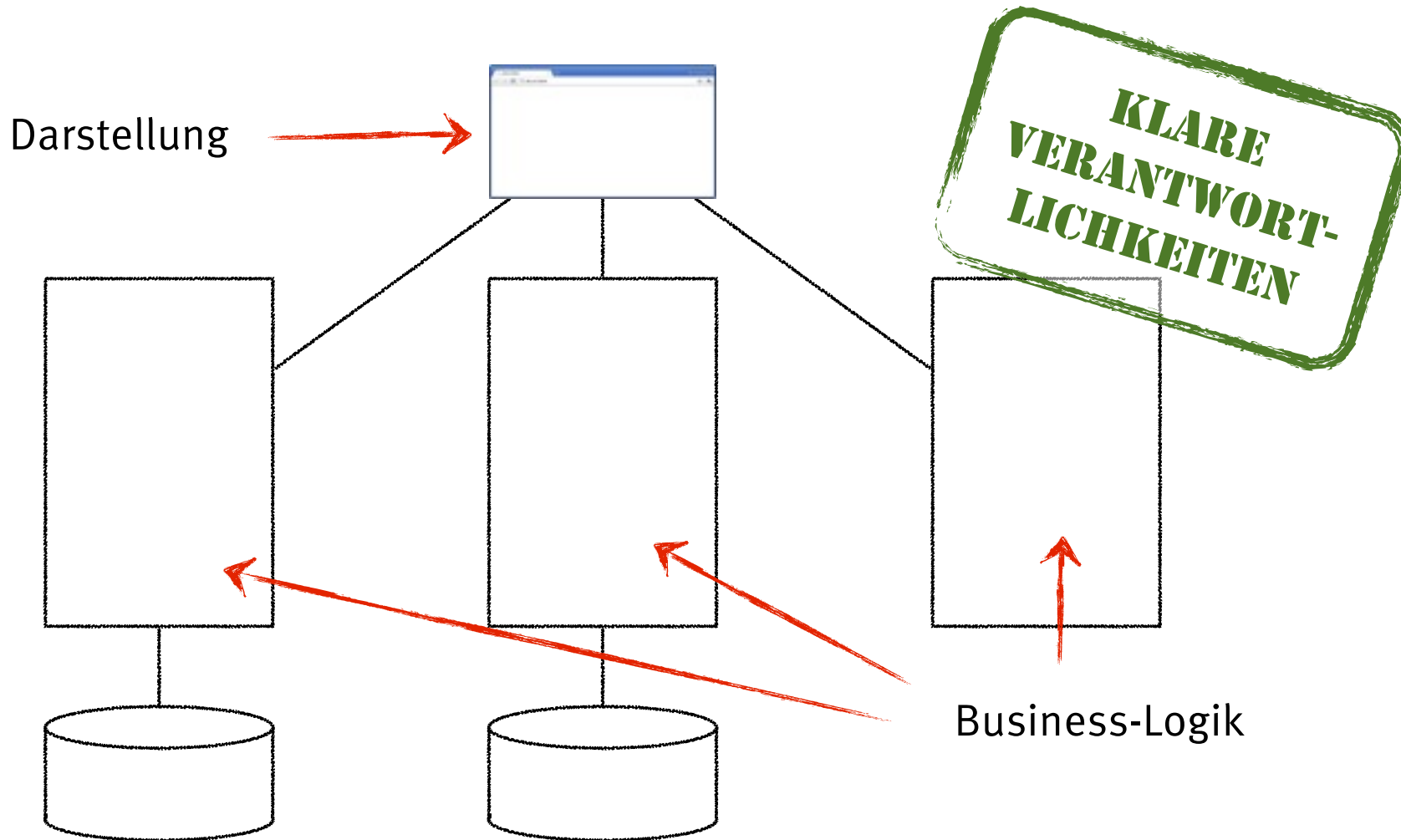


**VIEL
KNOW-HOW**

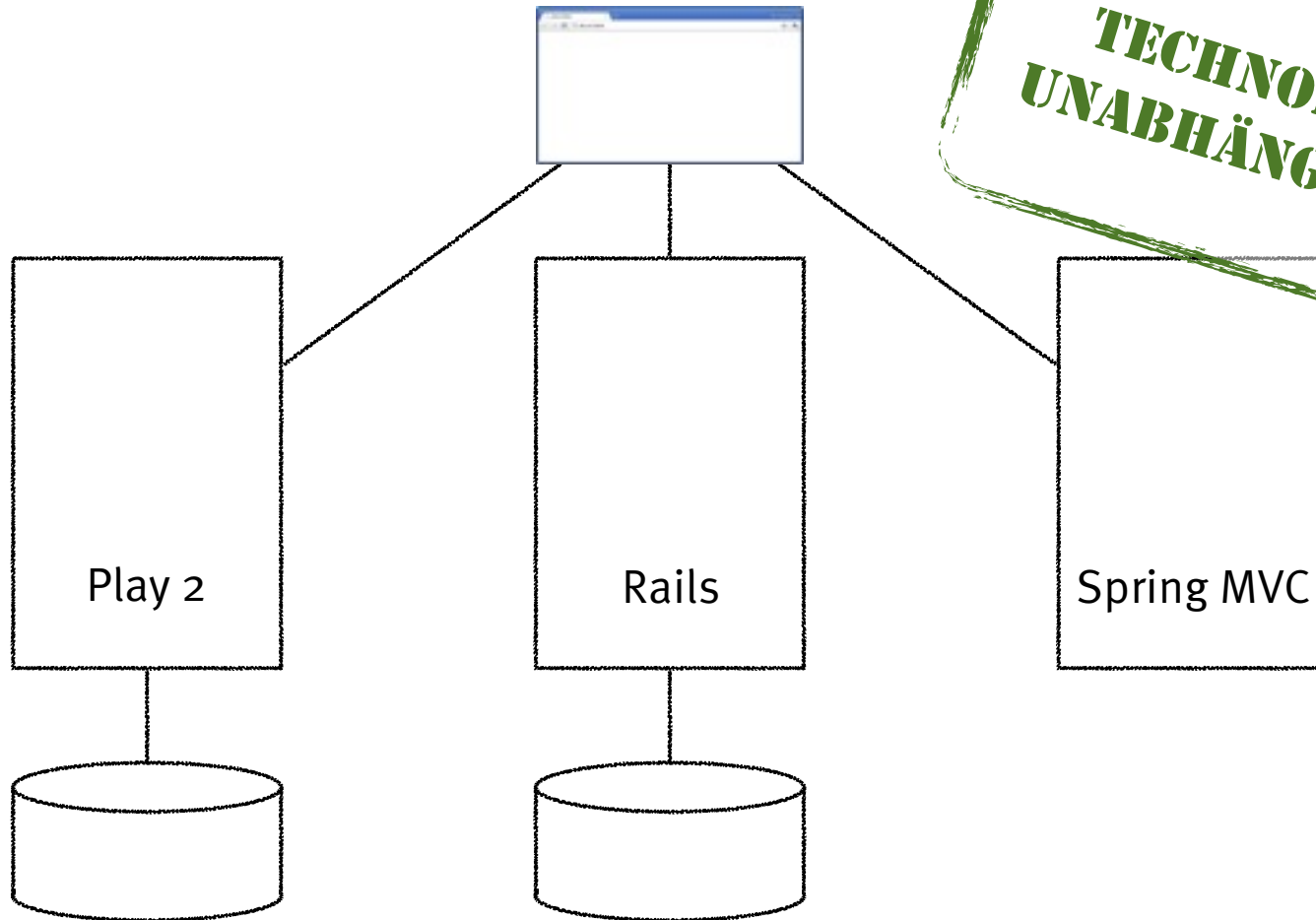
Vorteile ROCA



Vorteile ROCA



Vorteile ROCA



**TECHNOLOGIE-
UNABHÄNGIGKEIT**

Vielen Dank!
Fragen?
Anmerkungen?

Till Schulte-Coerne, @tillsc
till.schulte-coerne@innoq.com



www.innoq.com

innoQ Deutschland GmbH

Krischerstr. 100
40789 Monheim am Rhein
Germany
Phone: +49 2173 3366-0

Ohlauer Straße 43
10999 Berlin
Germany
Phone: +49 2173 3366-0

Robert-Bosch-Straße 7
64293 Darmstadt
Germany
Phone: +49 2173 3366-0

Radtkoferstraße 2
D-81373 München
Germany
Telefon +49 (0) 89 741185-270

innoQ Schweiz GmbH

Gewerbestr. 11
CH-6330 Cham
Switzerland
Phone: +41 41 743 0116

