

# JAVASCRIPT SINGLE-PAGE APPLIKATIONEN

Slides: [github.com/runjak/nook2017](https://github.com/runjak/nook2017)

Demo code: [github.com/runjak/simpleChess](https://github.com/runjak/simpleChess)

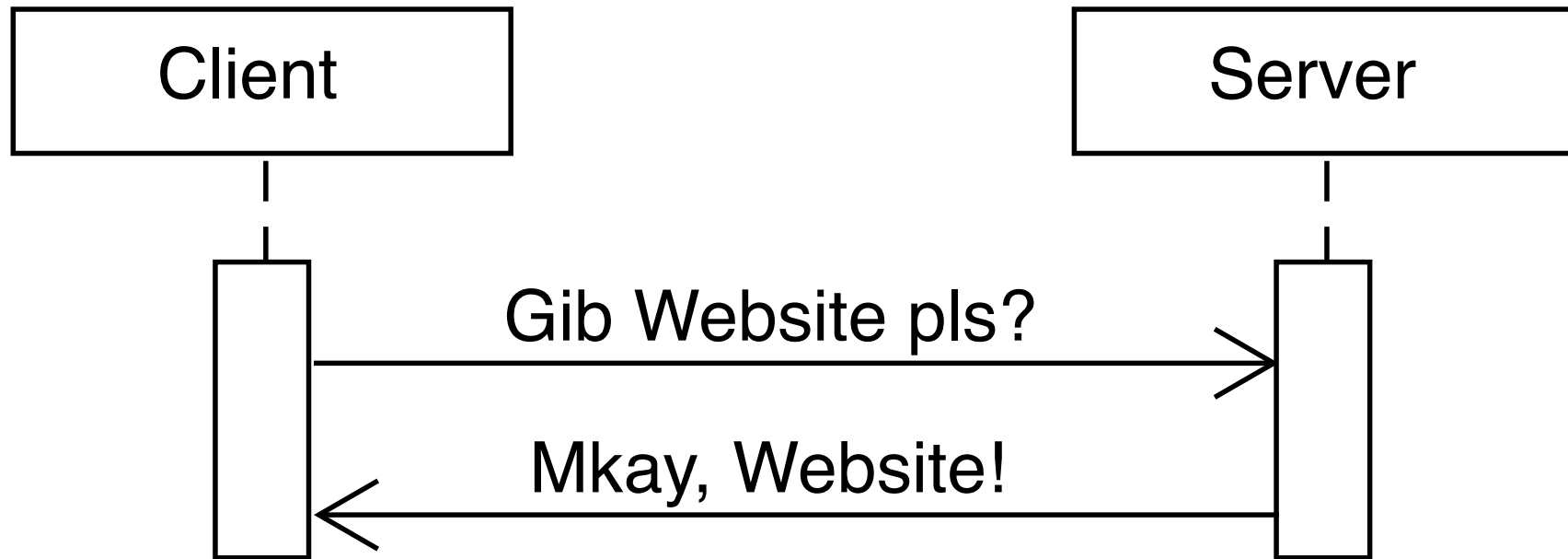
# ÜBERSICHT

- Intro (prelude to tragedy)
- Bisherige Ansätze
- React
- Kram zu React
- Das Problem mit dem State
- Die Flux-Architektur
- Code und Tooling begucken

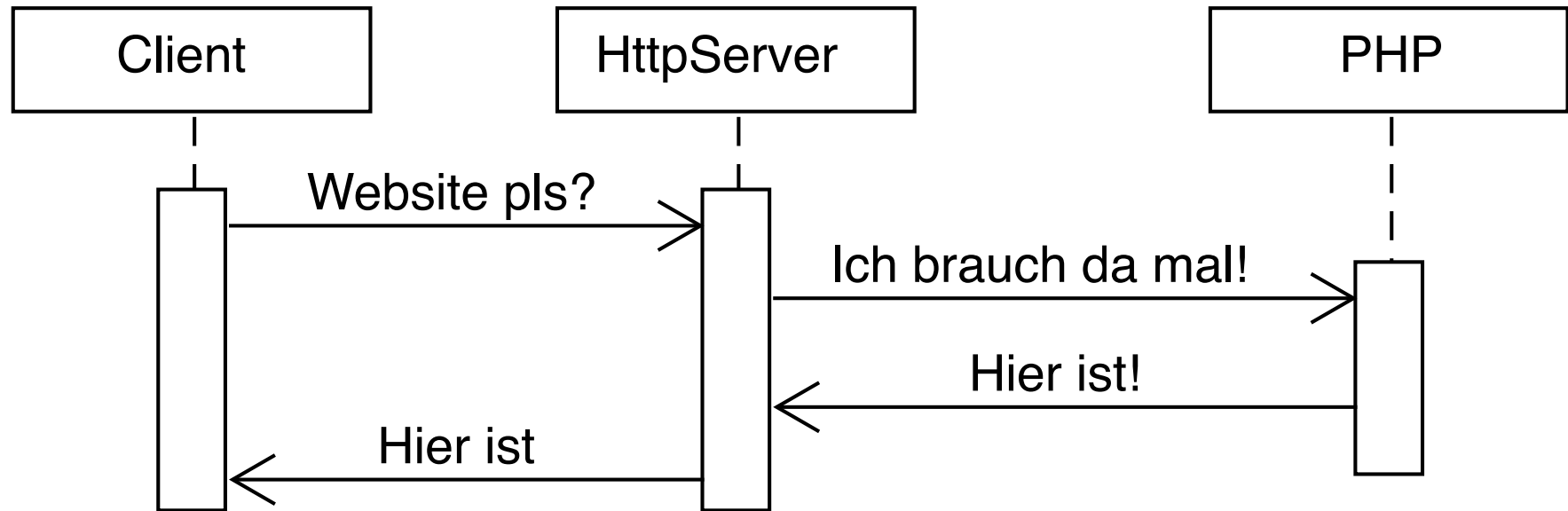
**HTML → BROWSER**

...aber wie?

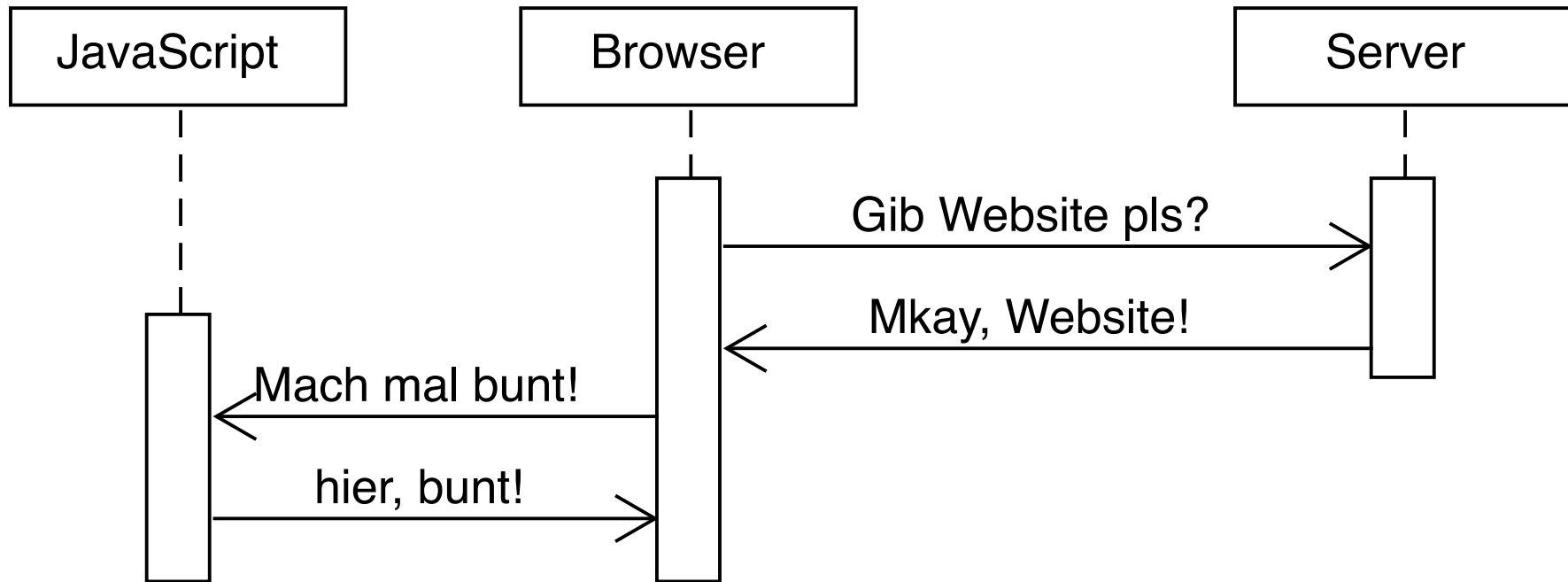
## Statisches HTML vom Server



# Dynamisches HTML vom Server



# Dynamisches HTML im Browser



# **BISHERIGE ANSÄTZE**

...also zum Rendern im Browser

# Vanilla JS

```
var body = document.querySelector('body');  
body.innerHTML = '<h1>Hello World!</h1>';
```



# jQuery

```
$( 'body' ).html( '<h1>Hello World!</h1>' );
```

## handlebars / mustache

```
<div class="entry">
  <h1>{{ title }}</h1>
  <div class="entryContent">
    {{ content }}
  </div>
</div>
```

# React

```
import React from 'react';
import { render } from 'react-dom';

render(
  <h2>'Hello World! '</h2>,
  document.getElementById('root'),
);
```

# REACT

Damit kann man user interfaces bauen

[reactjs.org](https://reactjs.org)

# REACT

- Was macht es, und wie?
- Es rendert ein UserInterface
- Es soll möglichst nur Änderungen rendern  
- das aber schnell
- Es soll aus Komponenten aufgebaut sein
- ...die stecken wir dann zusammen.

# REACT

- Was macht es nicht?
- Mit APIs sprechen
- CSS / styling
- Den Techstack kleiner

# **DIE PROBLEME UM REACT HERUM**

# MODULE

...in grauer Vorzeit:

```
<script />
```

CommonJS:

```
const modul = require('modul');
```

AMD:

```
require(['modul'], (modul) => {...});
```

ES6:

```
import Mainthing, { otherThing } from 'module';
```



# BIBLIOTHEKEN

Bibliothek	Zweck
Redux	State management
Reselect	An den State kommen
redux-thunk	Gewürz für Redux
styled-components	Styling/CSS
lodash	Taschenmesser

# TOOLING

Tool	Zweck
<a href="#">webpack</a>	zusammenbauen
<a href="#">eslint</a>	linter
<a href="#">jest</a>	tests
<a href="#">flow</a>	typechecking

Hilfe beim Setup: [create-react-app](#)

# MUTABLE STATE

...gar nicht mal so geil.

# SCHWIERIGER MIT MUTABLE STATE:

- Nebenläufigkeit
- Veränderungen bemerken
- Code verstehen
- Code testen
- Refactoring

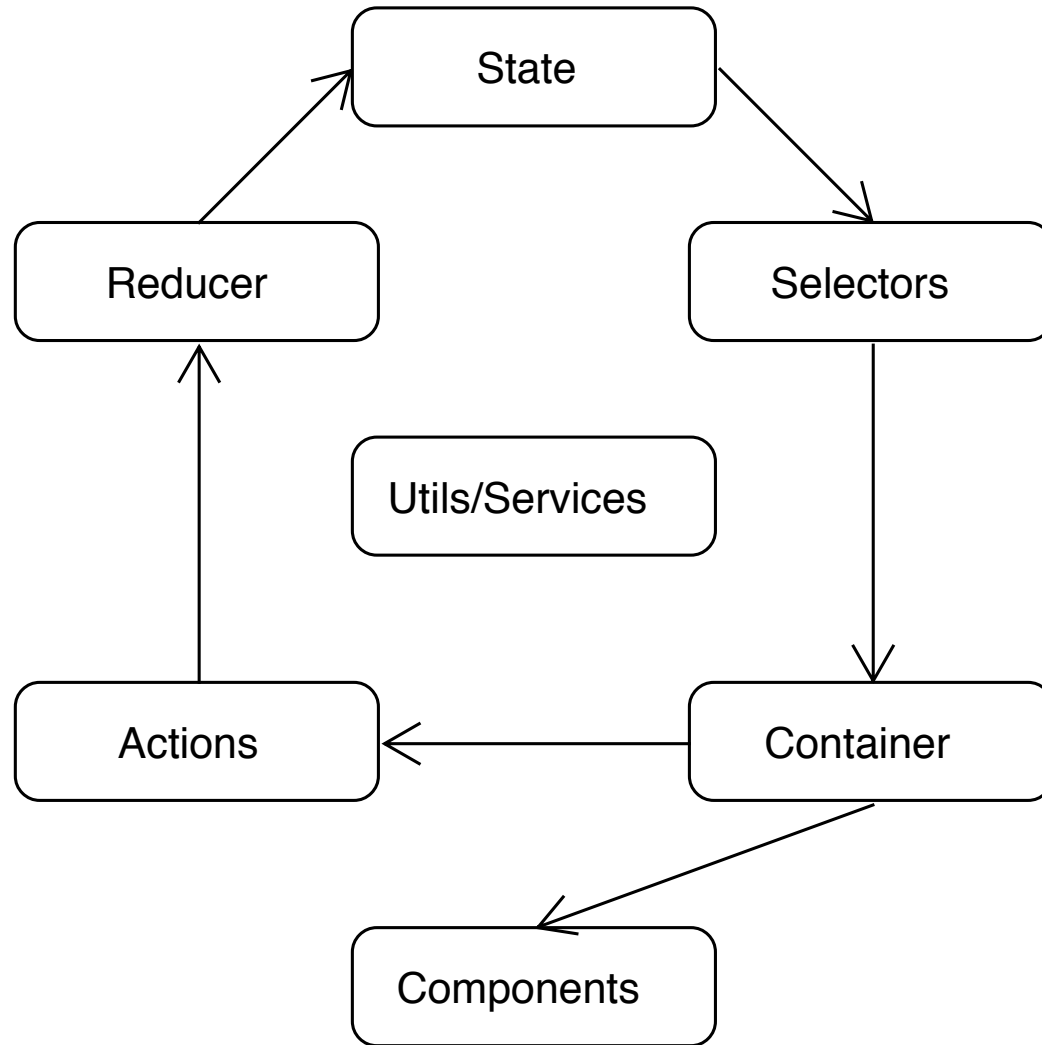
Aber: wir wollen ja auch was machen!

# **DIE FLUX-ARCHITEKTUR**

...Facebook hat da mal was 'neu' erfunden.

# DIE IDEE:

- Allen State zusammen isolieren
- Transaktionen auf immutable state
- Clever mitbekommen, wo sich was ändert







Wie sieht sowas aus?

```
import React from 'react';

function SimpleButton(props) {
  const { label, clickHandler } = props;

  return (
    <button onclick="{clickHandler}">
      {label}
    </button>
  );
}

export default SimpleButton;
```

# FRAGEN?

Slides: [github.com/runjak/nook2017](https://github.com/runjak/nook2017)

Demo code: [github.com/runjak/simpleChess](https://github.com/runjak/simpleChess)

Es war mir ein inneres Blumenessen

