

EINFÜHRUNG IN REACT.JS UND REDUX



NICOLE RAUCH
softwareentwicklung &
entwicklungscoaching

REACT.JS

EINE JAVASCRIPT LIBRARY ZUM ERSTELLEN VON USER INTERFACES

- Nur die UI
- Virtual DOM
- Data Flow

DATA FLOW

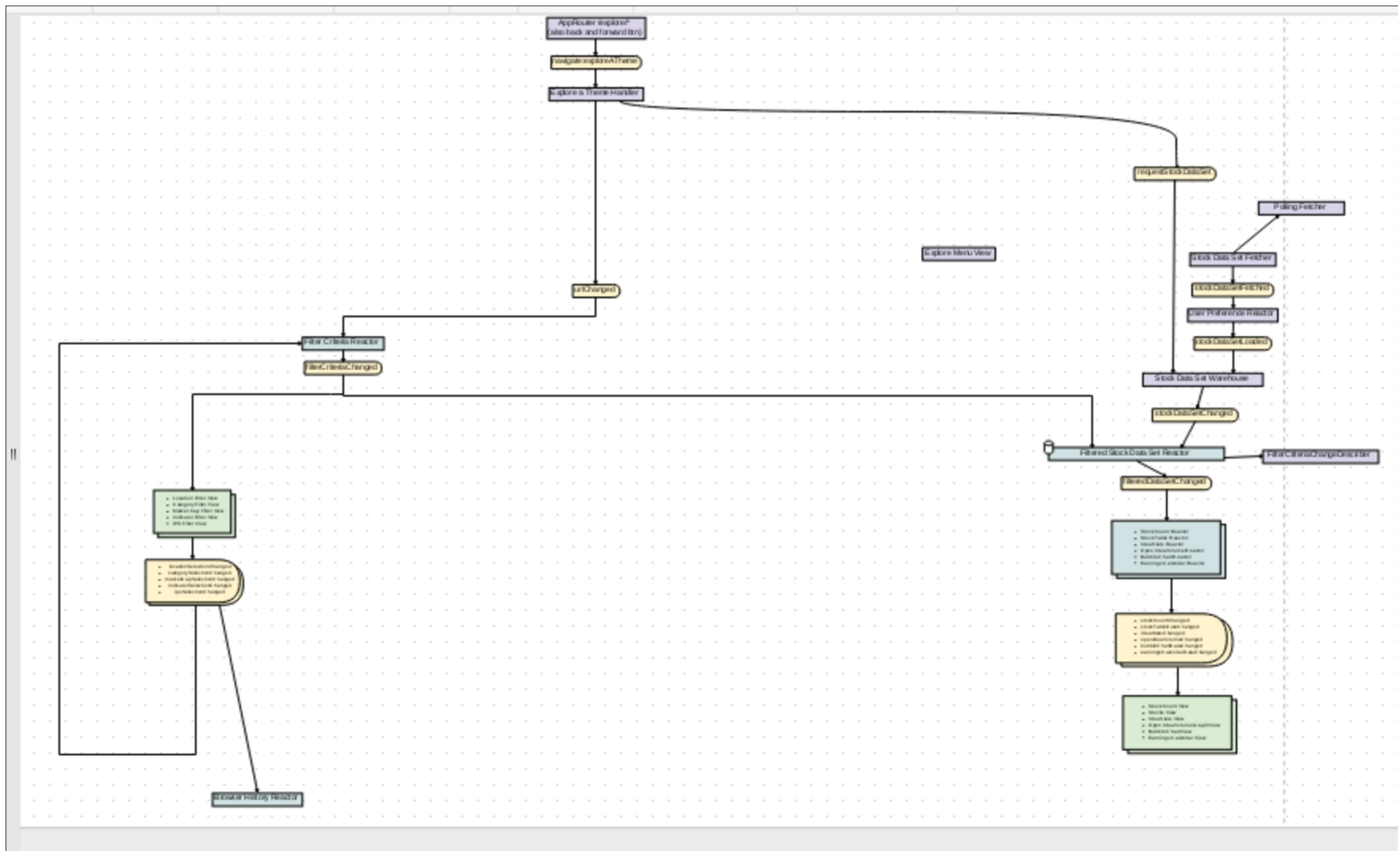
Unidirektional ohne Event-Chaos

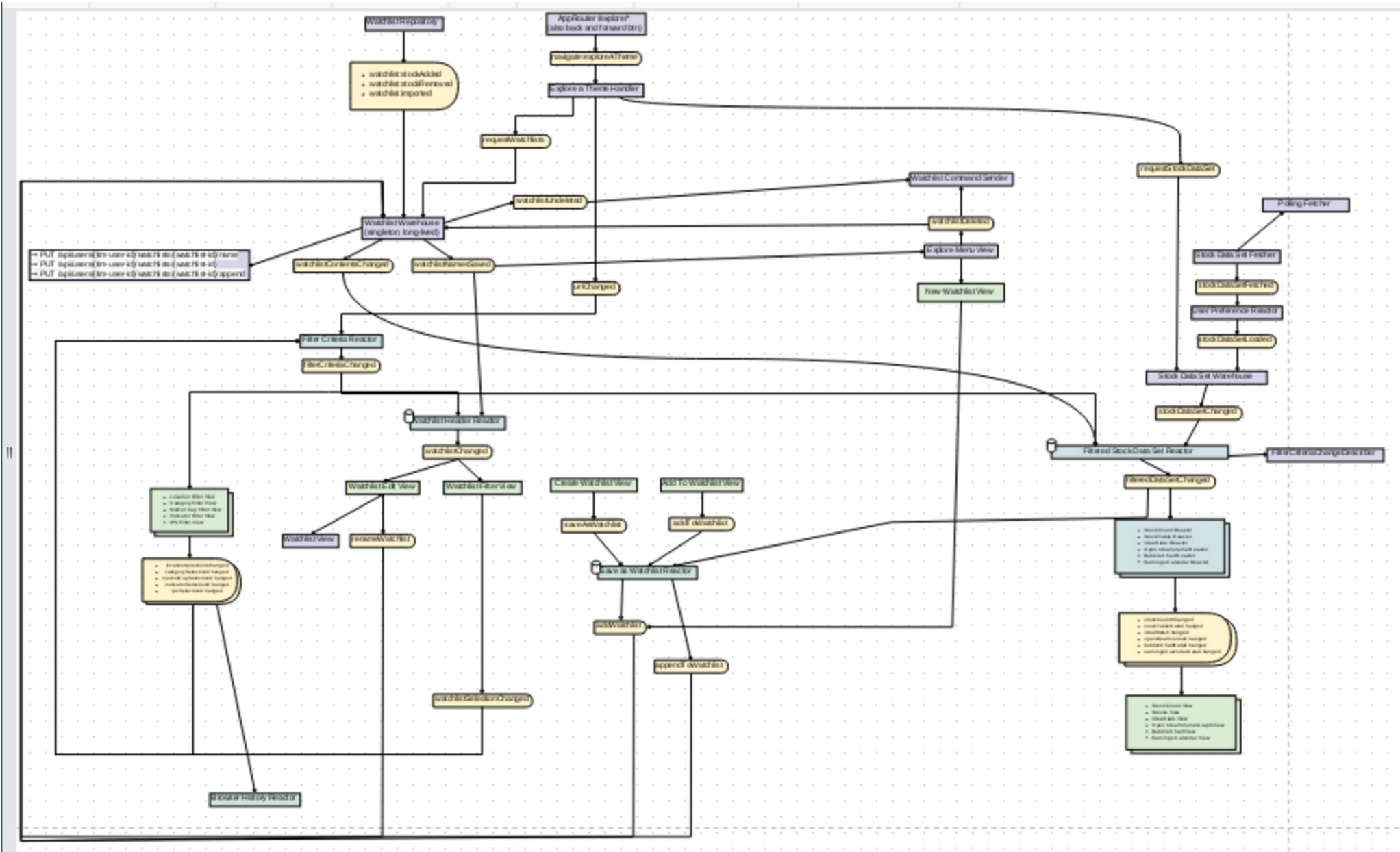


<http://www.wikihow.com/Image:Make-Spaghetti-Step-1.jpg>

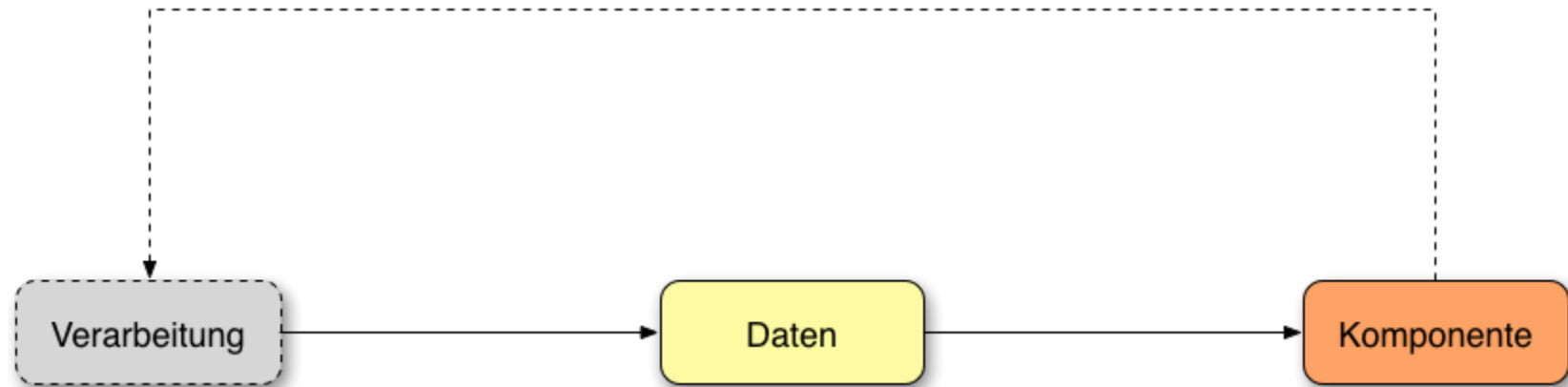


<https://www.flickr.com/photos/spilt-milk/5829575950>





REACT DATA FLOW



VIRTUAL DOM

Effizientes UI-Rendering inklusive

ÄNDERUNGEN AUF DER UI - WAS TUN?

Alles neuzeichnen? Flackert!

DOM manipulieren? Mühsam!

VIRTUAL DOM

Entwickler beschreibt das vollständige Resultat

React aktualisiert den DOM mit minimalen Änderungen

NUR DIE UI

Model ist kein Bestandteil von React

Kann beliebig implementiert werden

EINE EINFACHE KOMPONENTE

Hello World!

EINE EINFACHE KOMPONENTE

```
import React, { Component } from "react";

class HelloWorld extends Component {

  render() {
    return <p>Hello World!</p>;
  }
}
```

EINE EINFACHE KOMPONENTE

```
import React, { Component } from "react";  
  
class HelloWorld extends Component {  
  
  render() {  
    return <p>Hello World!</p>;  
  }  
}
```

EINE EINFACHE KOMPONENTE

```
import React, { Component } from "react";  
  
class HelloWorld extends Component {  
  
  render() {  
    return <p>Hello World!</p>;  
  }  
}
```


EINE EINFACHE KOMPONENTE

```
import React, { Component } from "react";

class HelloWorld extends Component {
  render() {
    return <p>Hello World!</p>;
  }
}
```

EINE EINFACHE KOMPONENTE

```
import React, { Component } from "react";

class HelloWorld extends Component {

  render() {
    return <p>Hello World!</p>;
  }
}
```

EINE EINFACHE KOMPONENTE

```
import React, { Component } from "react";

class HelloWorld extends Component {

  render() {
    return React.createElement('p', null, 'Hello World!');
  }
}
```

EINE EINFACHE KOMPONENTE

```
import React, { Component } from "react";

class HelloWorld extends Component {

  render() {
    return <p>Hello World!</p>;
  }
}
```

EINE EINFACHE KOMPONENTE

```
<HelloWorld />
```

EINE KOMPONENTE MIT DATEN

All Users

- Nicole Rauch
- Wilhelm Tell
- Che Guevara

Anzeige eines Users / Anzeige der Liste

EINE KOMPONENTE MIT DATEN

```
class User extends Component {  
  
  render() {  
    return <li>  
      <span>{this.props.user.firstName} {this.props.user.lastName}</span>  
    </li>;  
  }  
}
```

EINE KOMPONENTE MIT DATEN

```
class User extends Component {  
  
  render() {  
    return <li>  
      <span>{this.props.user.firstName} {this.props.user.lastName}</span>  
    </li>;  
  }  
}
```


VERWENDUNG

```
<ul>  
  { users.map((user, index) => <User user={user} key={"user_" + index} />) }  
</ul>
```

VERWENDUNG

```
<ul>  
  { users.map((user, index) => <User user={user} key={"user_" + index} />) }  
</ul>
```

VERWENDUNG

```
<ul>  
  { users.map((user, index) => <User user={user} key={"user_" + index} />) }  
</ul>
```

VERWENDUNG

```
<ul>  
  { users.map((user, index) => <User user={user} key={"user_" + index} />) }  
</ul>
```

VERWENDUNG

```
<ul>  
  { users.map((user, index) => <User user={user} key={"user_" + index} />) }  
</ul>
```

VERWENDUNG

```
<ul>  
  { users.map((user, index) => <User user={user} key={"user_" + index} />) }  
</ul>
```

ANLEGEN EINES BENUTZERS

New User

First Name:

Last Name:

Submit

ANLEGEN EINES BENUTZERS

```
class UserCreation extends Component {  
  
  render() {  
    return (  
      <div>  
        <div>New User</div>  
        <div> <label>First Name:</label> <input type="string" /> </div>  
        <div> <label>Last Name:</label> <input type="string" /> </div>  
        <div> <button onClick={ /* ... */ }>Submit</button> </div>  
      </div>  
    );  
  }  
}
```


ANLEGEN EINES BENUTZERS

```
class UserCreation extends Component {  
  
  render() {  
    return (  
      <div>  
        <div>New User</div>  
        <div> <label>First Name:</label> <input type="string" /> </div>  
        <div> <label>Last Name:</label> <input type="string" /> </div>  
        <div> <button onClick={ /* ... */ }>Submit</button> </div>  
      </div>  
    );  
  }  
}
```

ANLEGEN EINES BENUTZERS

```
class UserCreation extends Component {  
  
  render() {  
    return (  
      <div>  
        <div>New User</div>  
        <div> <label>First Name:</label> <input type="string" /> </div>  
        <div> <label>Last Name:</label> <input type="string" /> </div>  
        <div> <button onClick={ /* ... */ }>Submit</button> </div>  
      </div>  
    );  
  }  
}
```

ANLEGEN EINES BENUTZERS

```
class UserCreation extends Component {  
  
  render() {  
    return (  
      <div>  
        <div>New User</div>  
        <div> <label>First Name:</label> <input type="string" /> </div>  
        <div> <label>Last Name:</label> <input type="string" /> </div>  
        <div> <button onClick={ /* ... */ }>Submit</button> </div>  
      </div>  
    );  
  }  
}
```

DIE GESAMTE APPLIKATION

All Users

- Nicole Rauch
- Wilhelm Tell
- Che Guevara

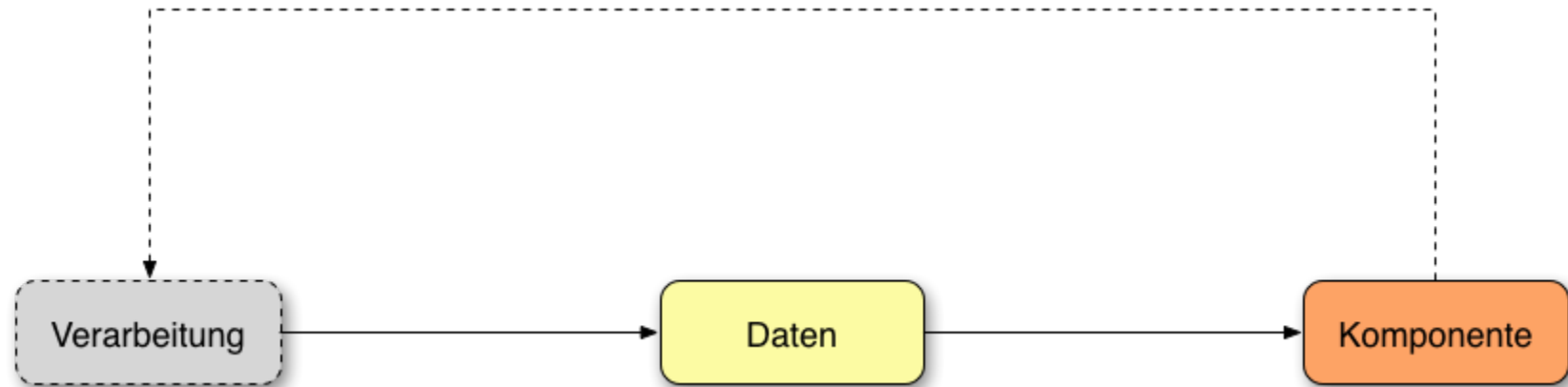
New User

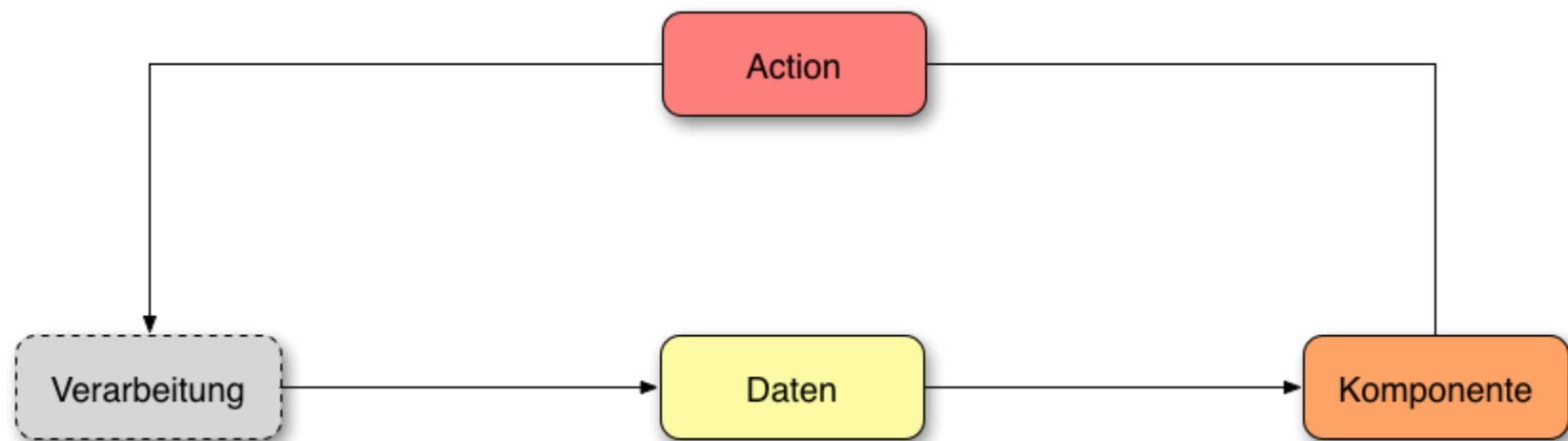
First Name:

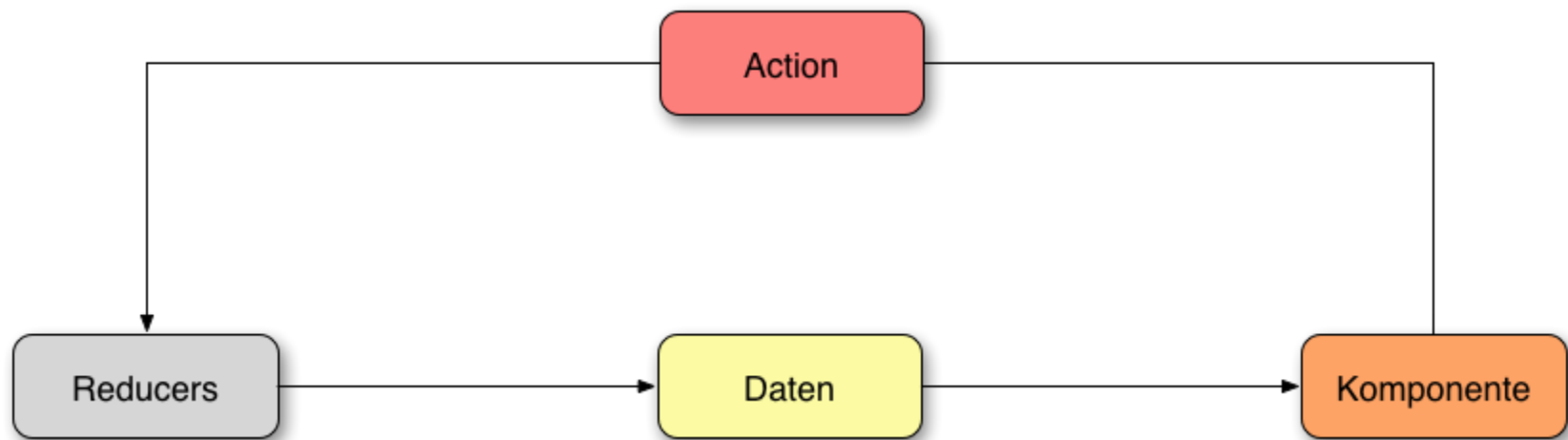
Last Name:

Submit

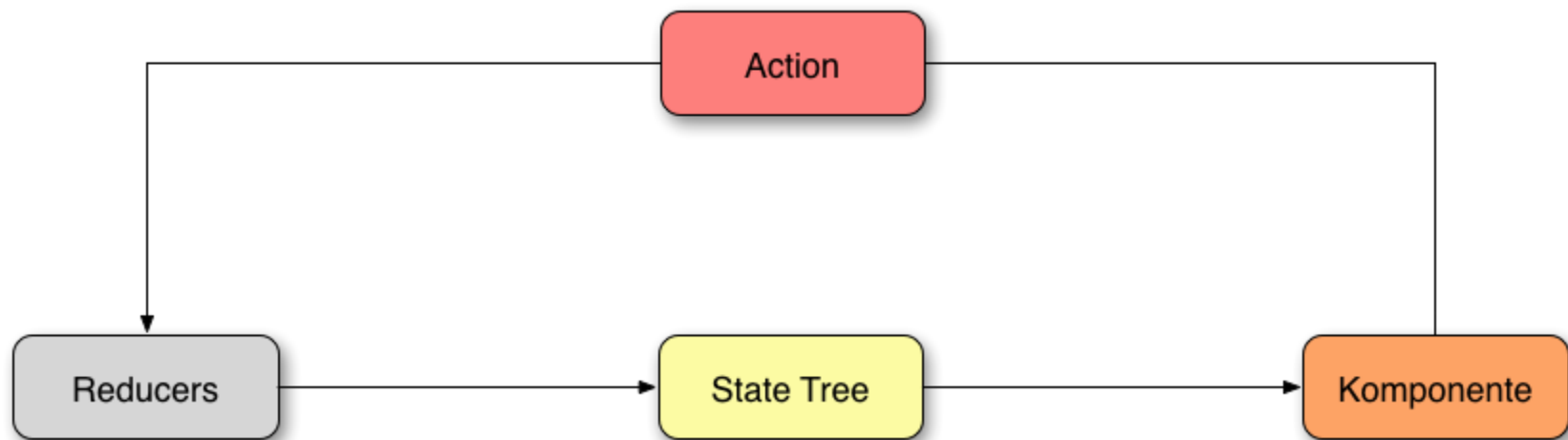
REACT DATA FLOW







REDUX DATA FLOW



ACTION

```
const USER_ADDED = "USER_ADDED";

function addUser(user) {
  return {
    type: USER_ADDED,
    payload: user
  }
}
```

Sehr ähnlich zu CQRS / Event Sourcing

STATE TREE

```
INITIAL_STATE = {  
  users: []  
};
```

REDUCERS

```
function reducers(state = INITIAL_STATE, action = {}){  
  return {  
    users: users(state.users, action)  
  };  
}
```

```
function users(state = INITIAL_STATE.users, action = {}) {  
  switch (action.type) {  
    case Events.USER_ADDED:  
      return state.concat(action.payload);  
  }  
  return state;  
}
```

INTEGRATION IN DIE KOMPONENTE

```
class UserCreation extends Component {  
  
  render() {  
    return (  
      <div>  
        <div>User Creation</div>  
        <div> <label>First Name:</label> <input type="string" /> </div>  
        <div> <label>Last Name:</label> <input type="string" /> </div>  
        <div> <button onClick={  
          /* ... */  
        }>Submit</button> </div>  
      </div>  
    );  
  }  
}
```

INTEGRATION IN DIE KOMPONENTE

```
class UserCreation extends Component {  
  
  render() {  
    return (  
      <div>  
        <div>User Creation</div>  
        <div> <label>First Name:</label> <input type="string" /> </div>  
        <div> <label>Last Name:</label> <input type="string" /> </div>  
        <div> <button onClick={  
          /* ... */  
        }>Submit</button> </div>  
      </div>  
    );  
  }  
}
```

INTEGRATION IN DIE KOMPONENTE

```
class UserCreation extends Component {  
  
  render() {  
    return (  
      <div>  
        <div>User Creation</div>  
        <div> <label>First Name:</label> <input type="string" /> </div>  
        <div> <label>Last Name:</label> <input type="string" /> </div>  
        <div> <button onClick={  
          () => { this.props.submitUser({firstName: ..., lastName: ...}) }  
        }>Submit</button> </div>  
      </div>  
    );  
  }  
}
```

```
<UserCreation submitUser={ user => dispatch(Actions.addUser(user)) } />
```

REACT HOT RELOADING

REDUX LOGGER

VIELEN DANK!



NICOLE RAUCH
softwareentwicklung &
entwicklungskoaching

@NicoleRauch
info@nicole-rauch.de
<http://www.nicole-rauch.de>

<https://github.com/NicoleRauch/VortragReactUndRedux>

Ganztägiger Workshop zu React und Redux