

O'REILLY®

Velocity

CONFERENCE

BUILD RESILIENT SYSTEMS AT SCALE



velocityconf.com

#velocityconf

React & Redux in Hulu

程墨 (Morgan Cheng)

About Me

@morgancheng



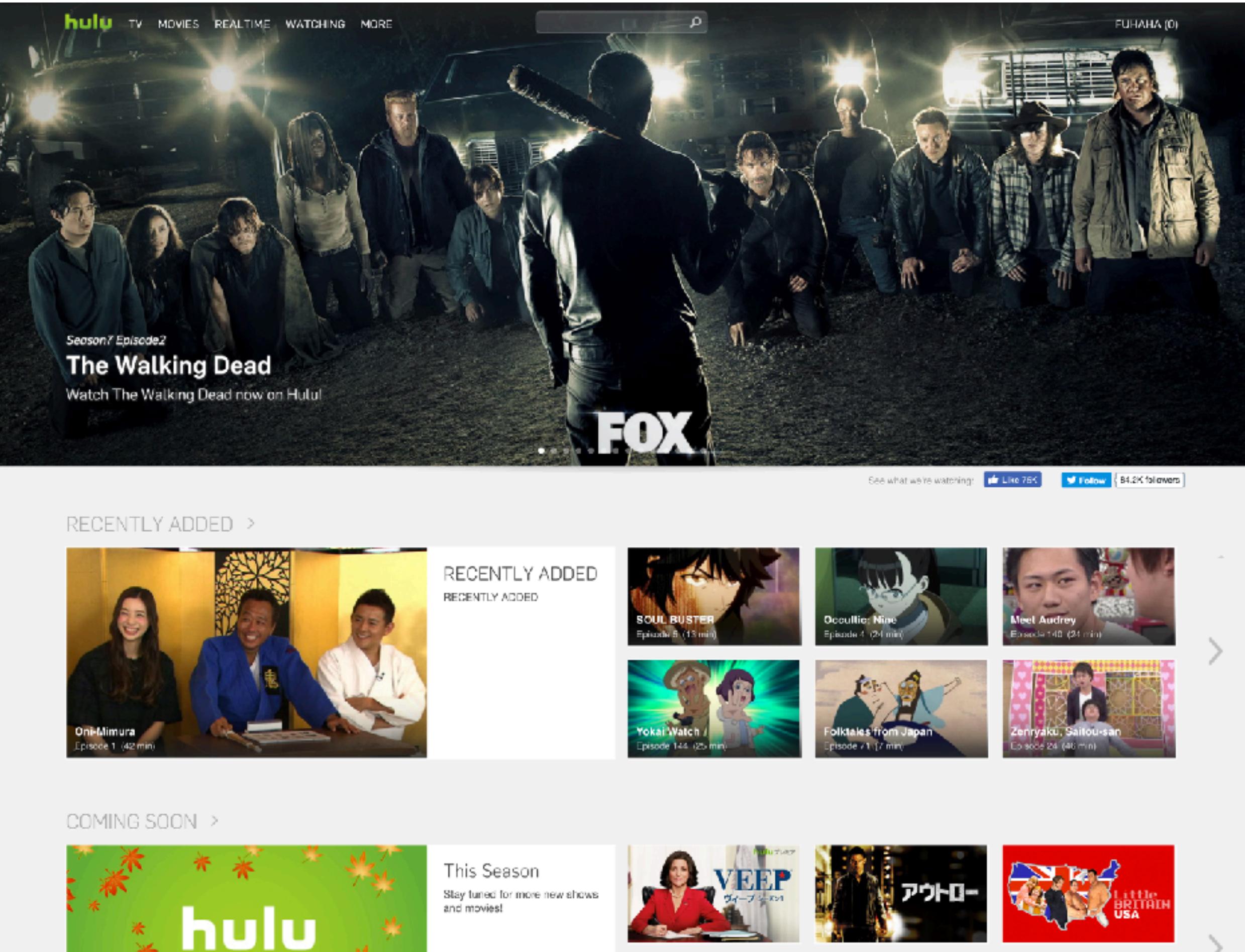
YAHOO!



#velocityconf

O'REILLY®
Velocity

About Hulu



#velocityconf

O'REILLY®
Velocity

Challenges

- System Complexity
 - Legacy code in jQuery & Backbone
 - UI non-predictable
 - Lack of component isolation
- Performance Bottleneck
 - Initial AJAX roundtrip for browser-side-rendering



#velocityconf

O'REILLY®
Velocity

Tech Stack

jQuery

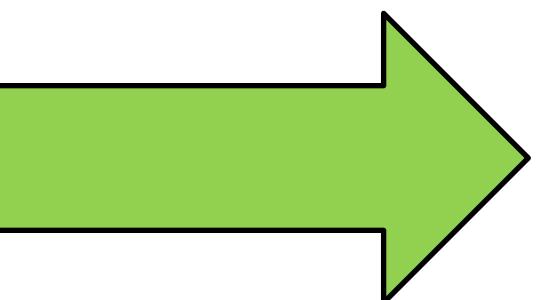
Backbone

Ruby on Rails

React

Redux

Node + Express



React vs. jQuery

#velocityconf

O'REILLY®
Velocity

Code in jQuery

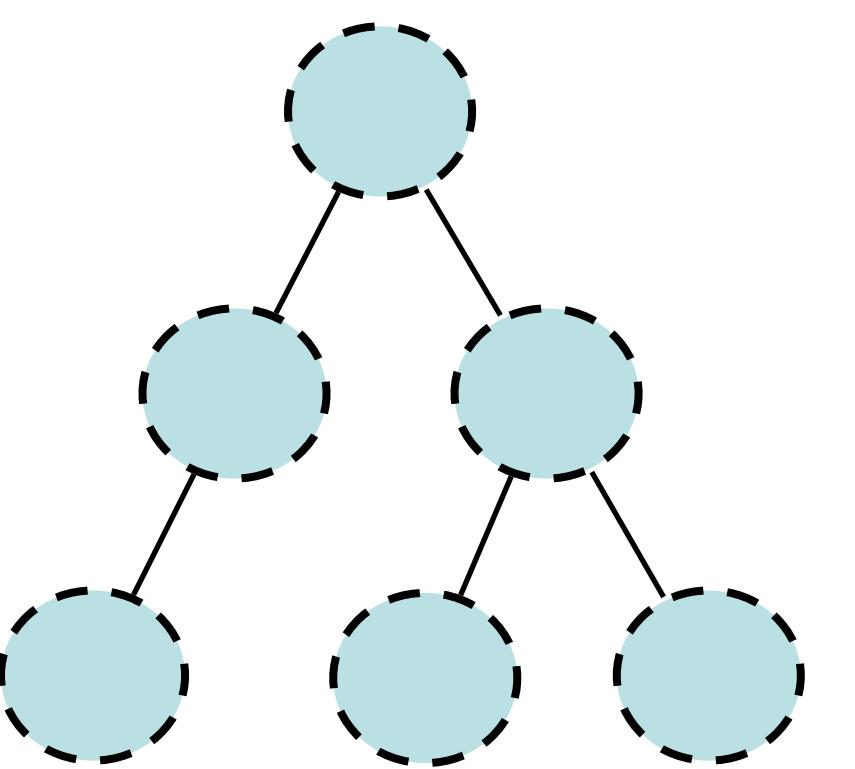
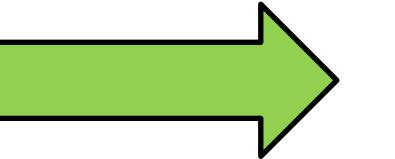


#velocityconf

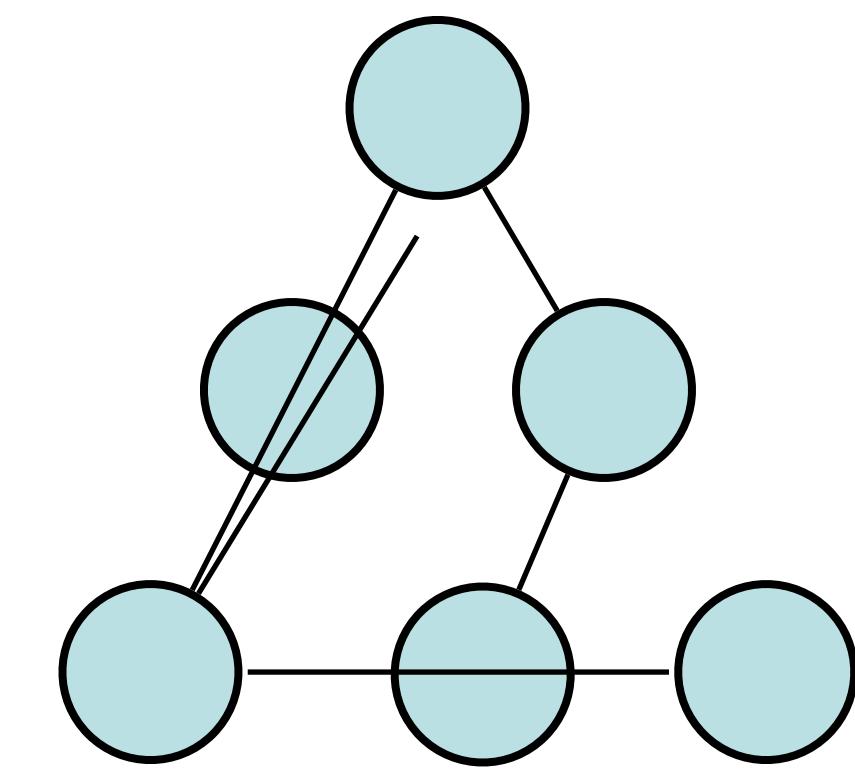
O'REILLY®
Velocity

```
<div>  
  <Foo/>  
  <Bar>children</Bar>  
</div>
```

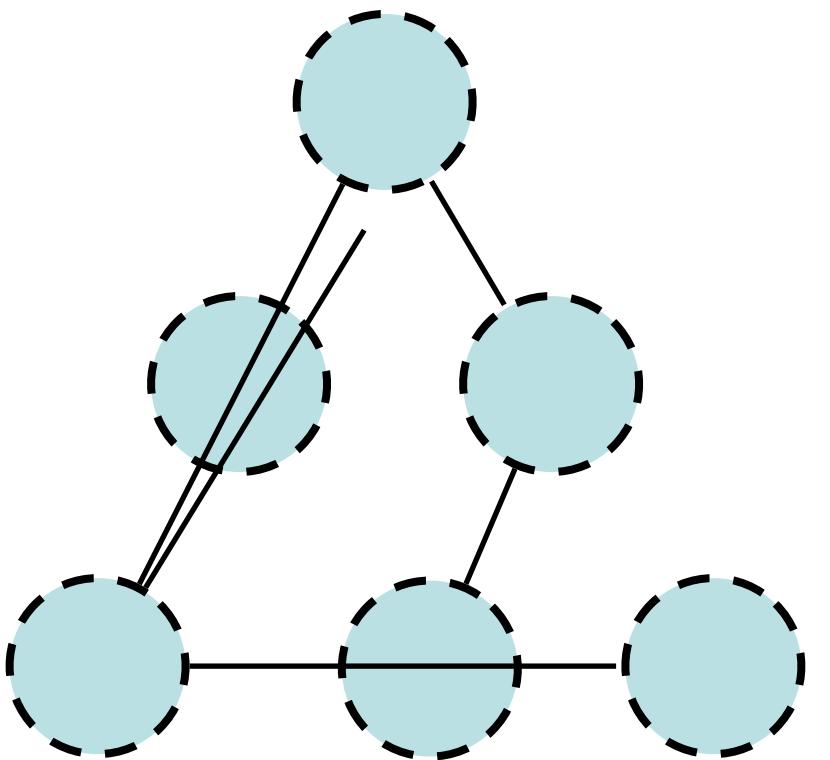
JSX



Virtual DOM

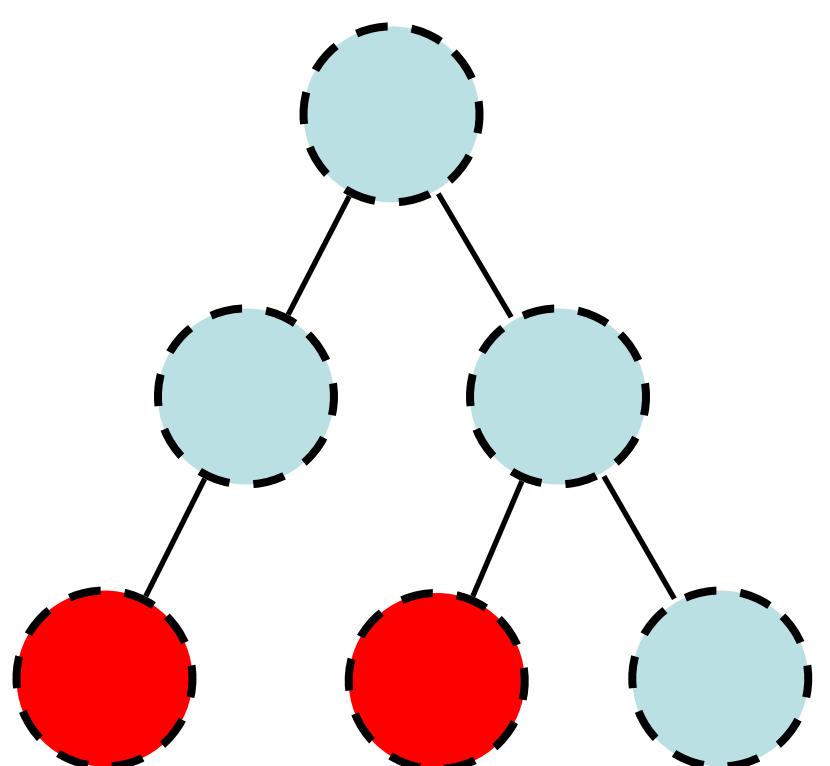


DOM

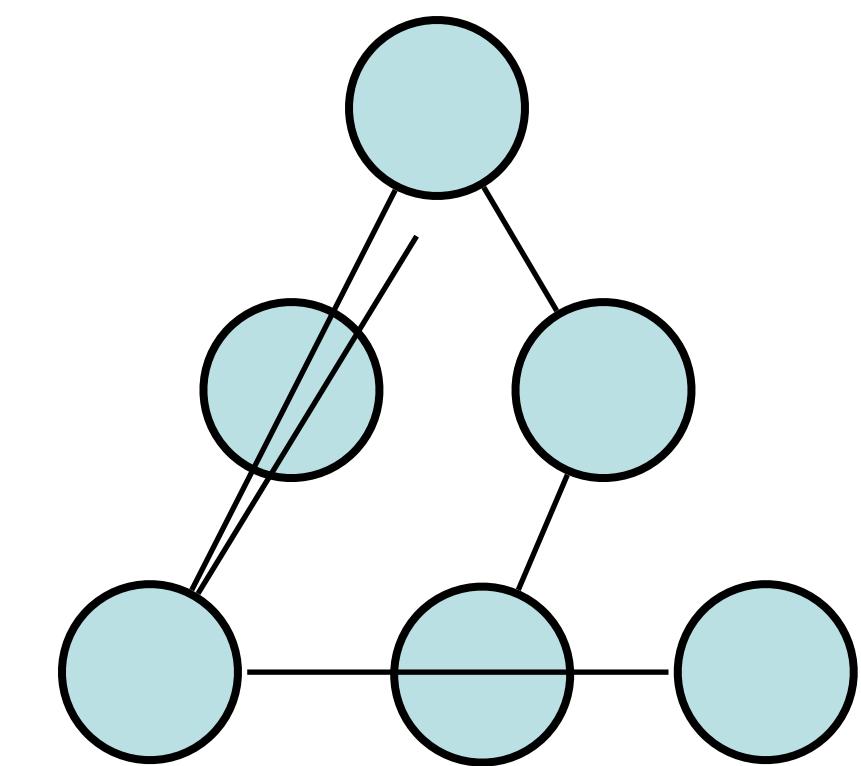


Previous Virtual DOM

```
<div>
  <Foo/>
  <Bar>children</Bar>
</div>
```



Virtual DOM

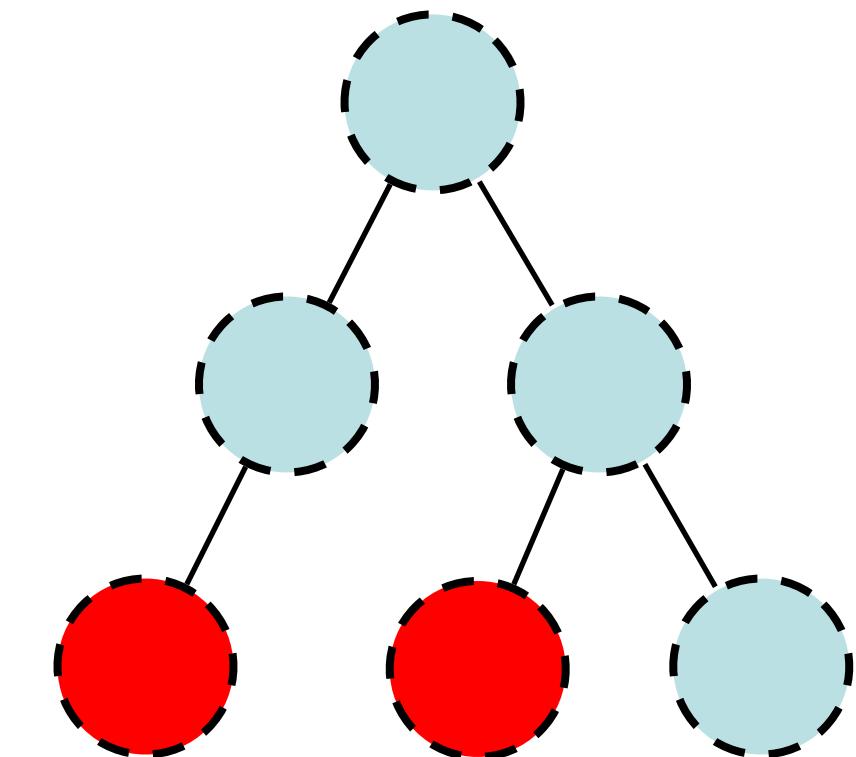


DOM

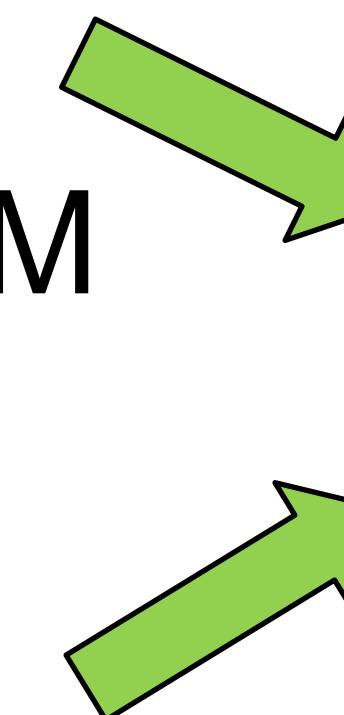
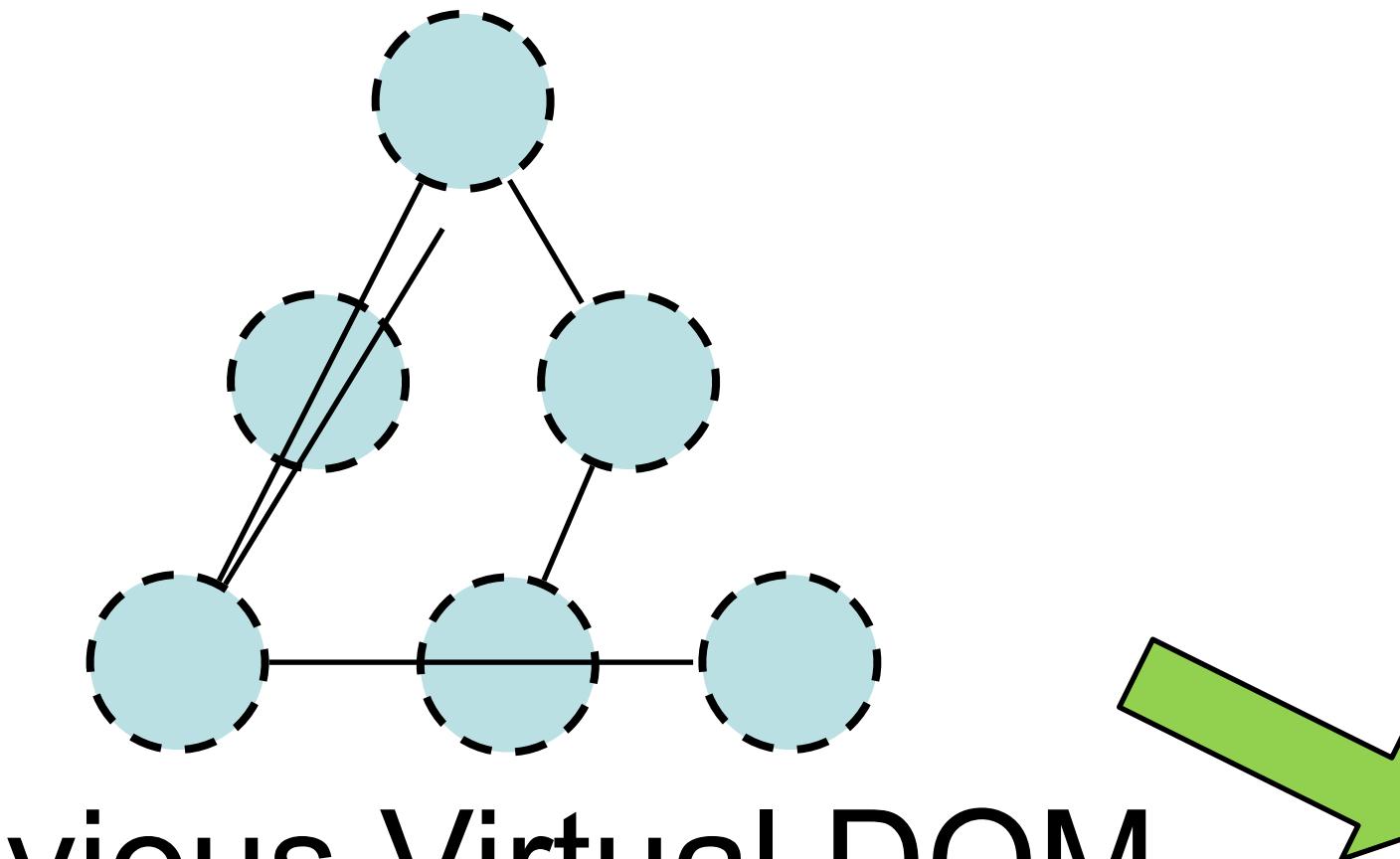
```
<div>
  <Foo/>
  <Bar>children</Bar>
</div>
```

JSX

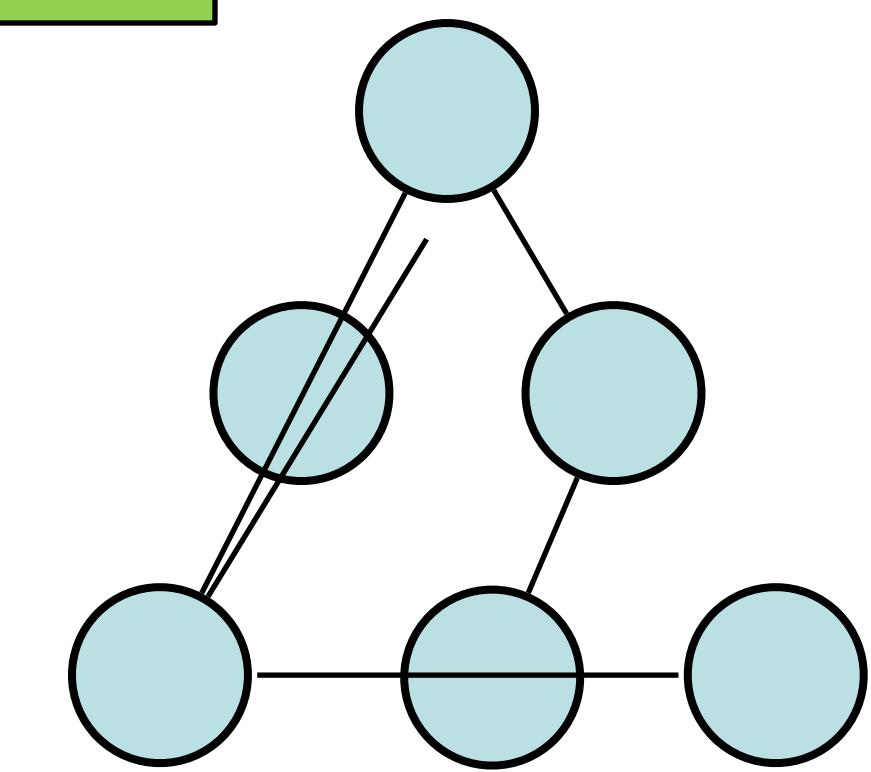
Previous Virtual DOM



Virtual DOM



Diff

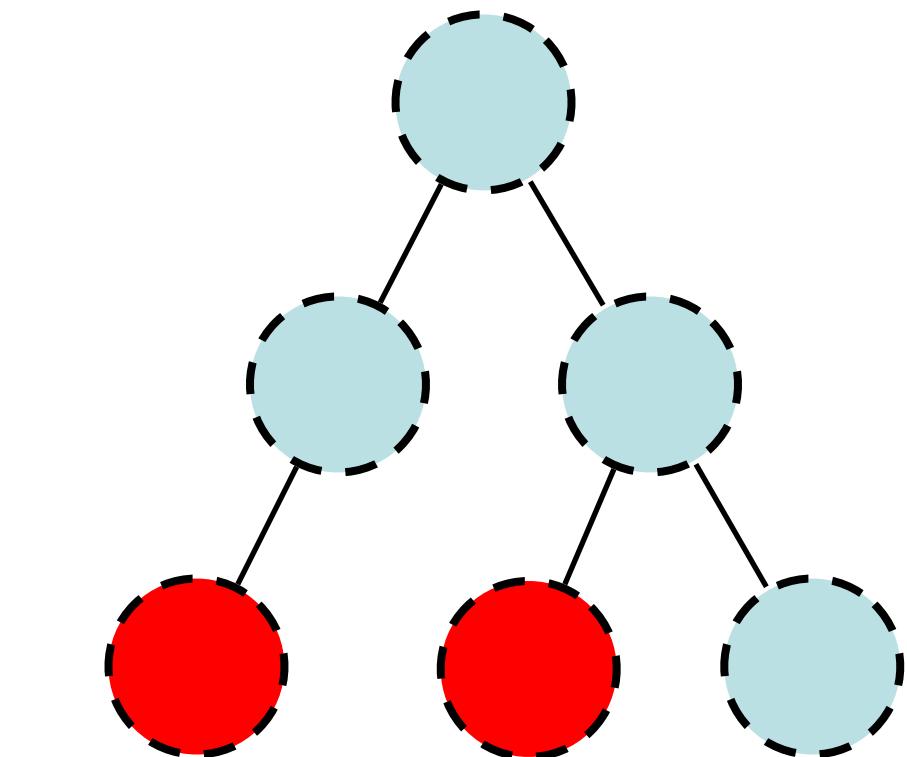


DOM

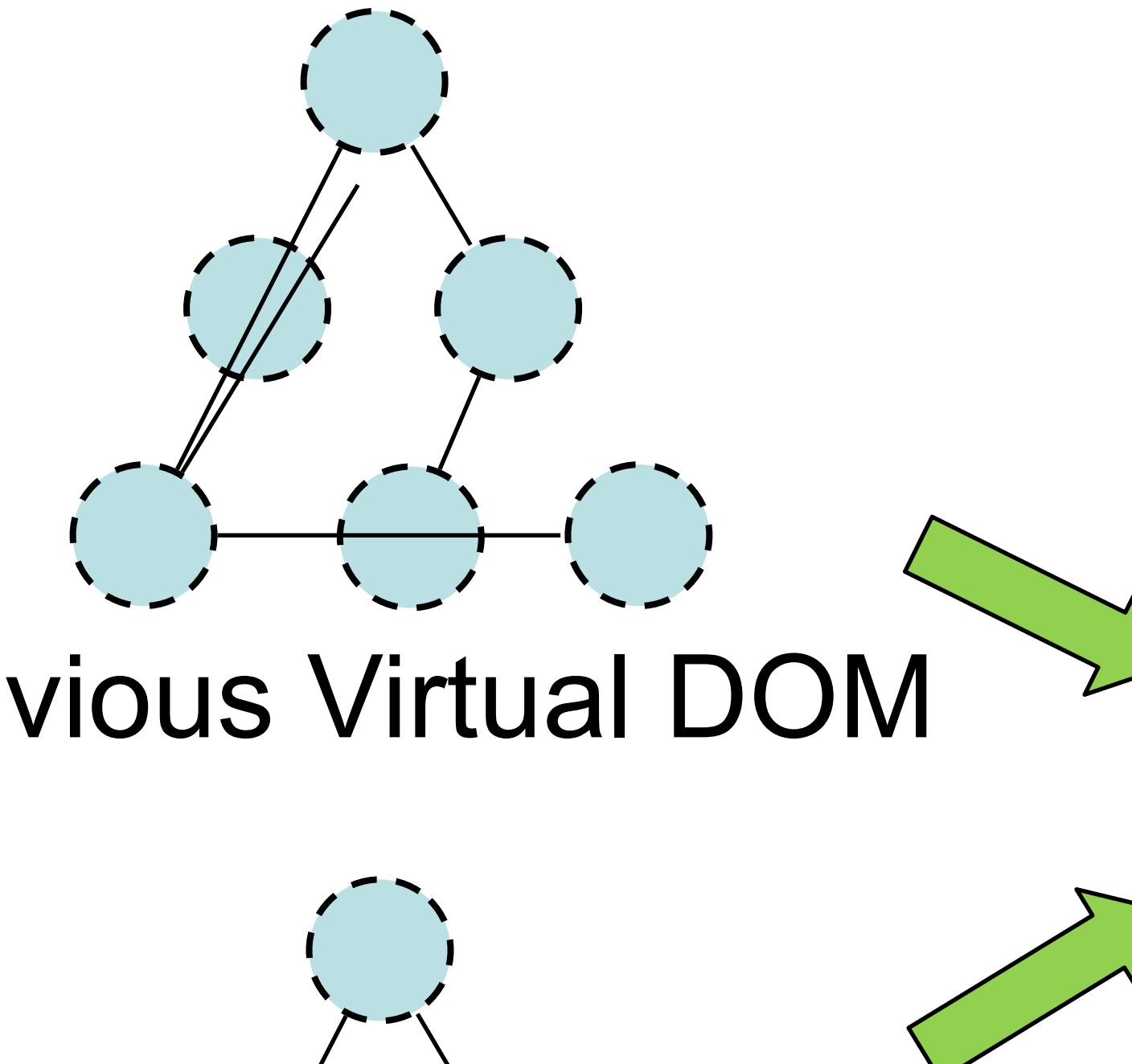
```
<div>
  <Foo/>
  <Bar>children</Bar>
</div>
```

JSX

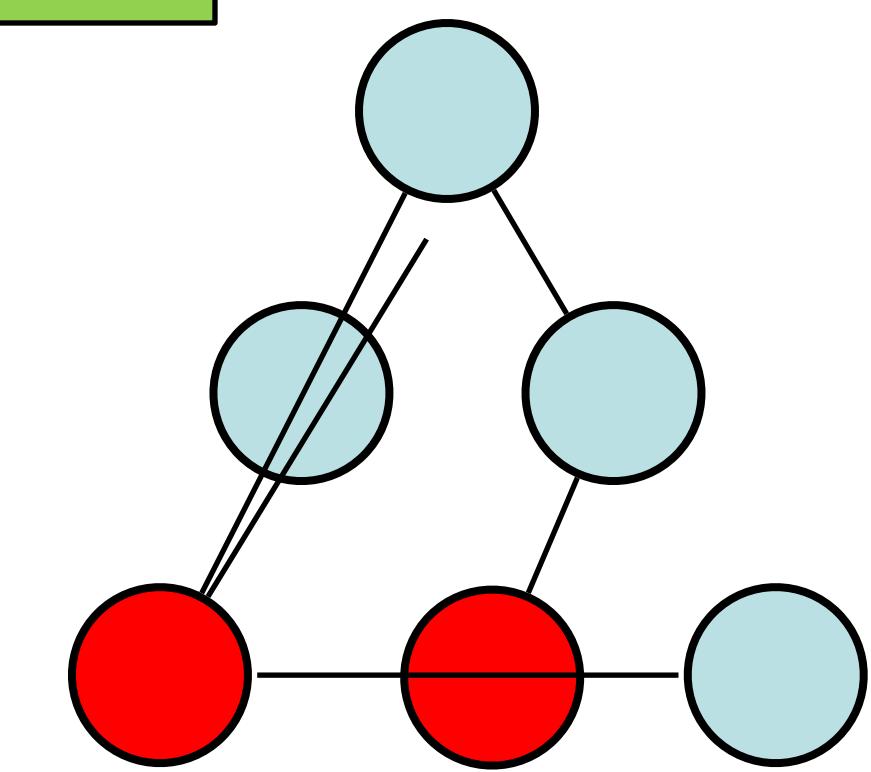
Previous Virtual DOM



Virtual DOM



Diff



DOM

React App = Keep Rendering

#velocityconf

O'REILLY®
Velocity

UI = f(state)

#velocityconf

O'REILLY®
Velocity

React mixes JS/Style/Markup

```
if (process.env.BROWSER) {  
  require('./style.scss');  
}  
  
const divStyle = {  
  color: 'blue'  
};  
  
function render() {  
  
  return (  
    <div onClick={this.onClick} style={divStyle}>Click Me</div>  
  );  
}
```

Isomorphic/Universal React

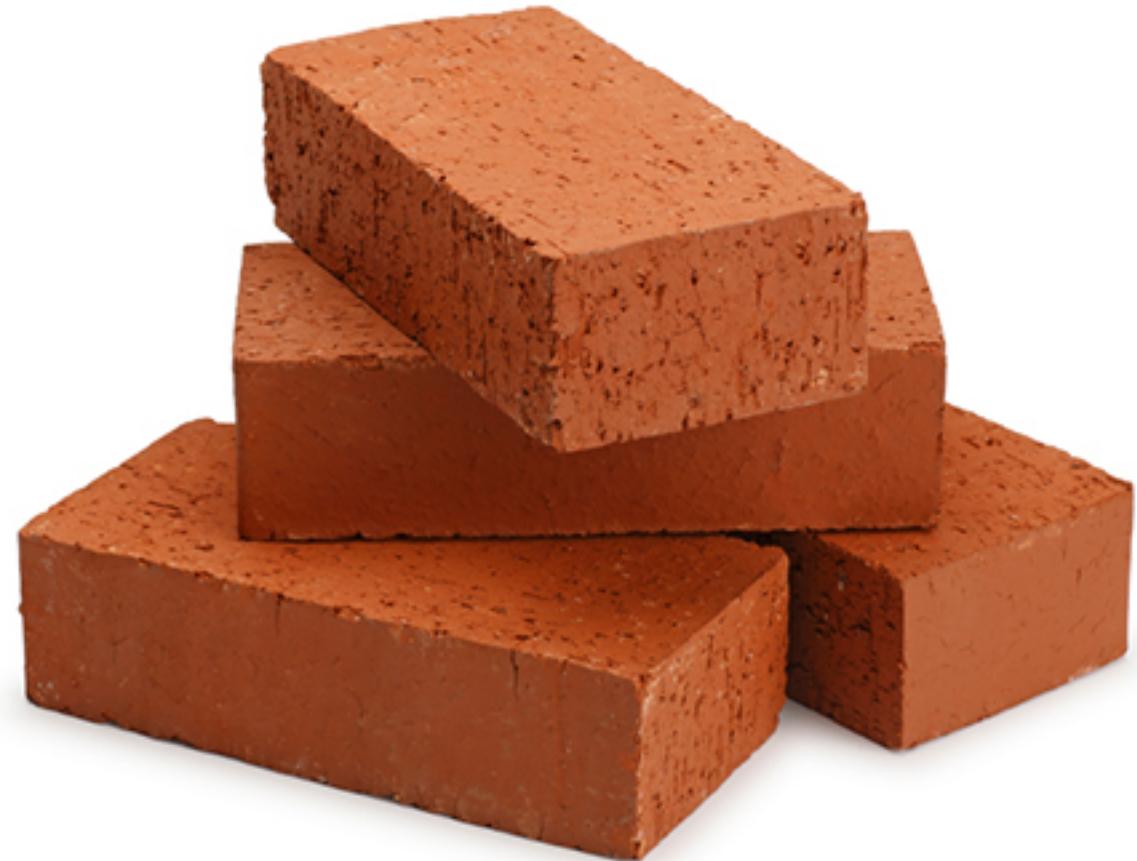
```
const appHtml = ReactDOMServer.renderToString(component);
```

Server Side

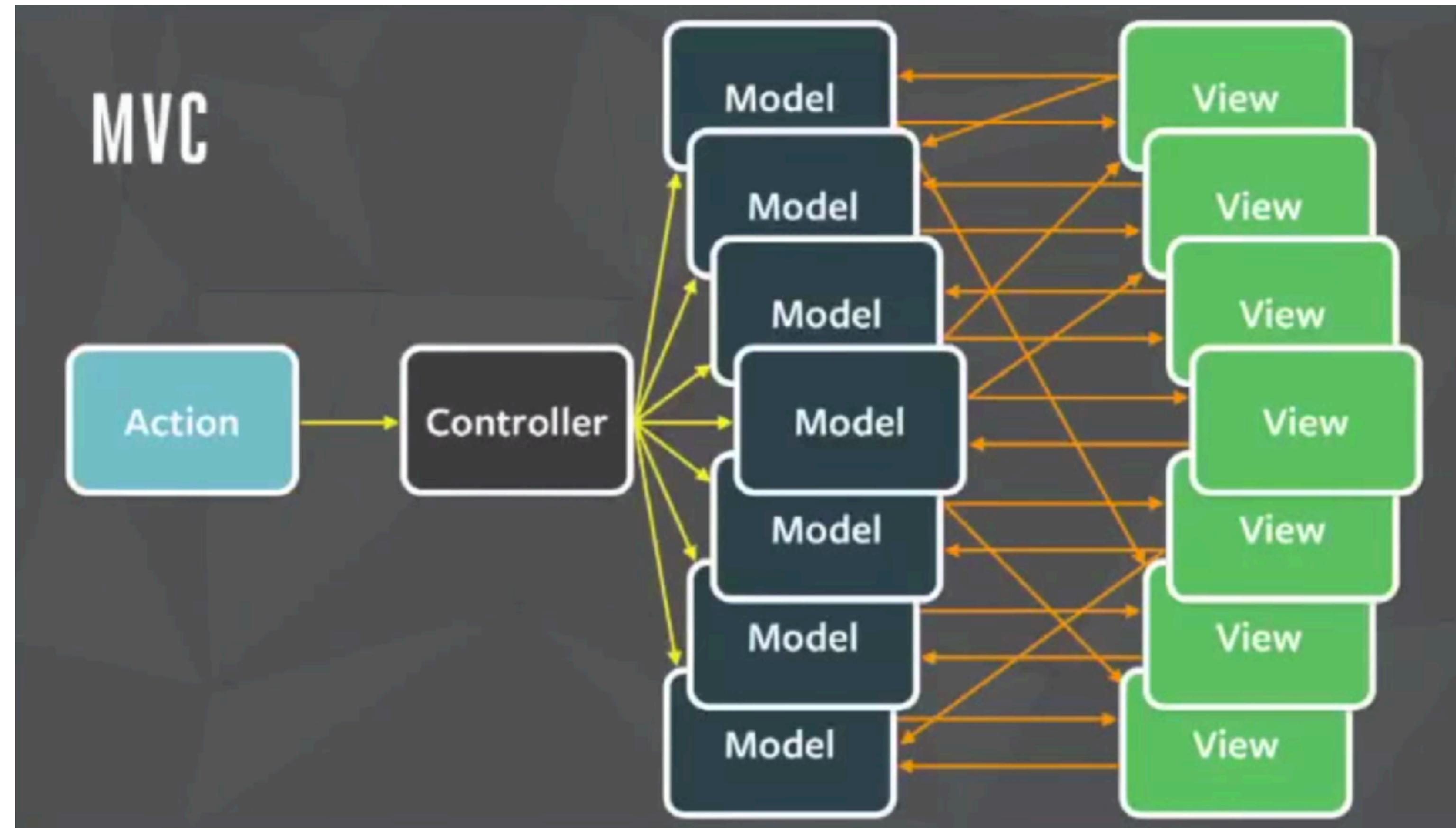
Browser Side

```
ReactDOM.render(component,  
  window.document.getElementById("appRoot"));
```

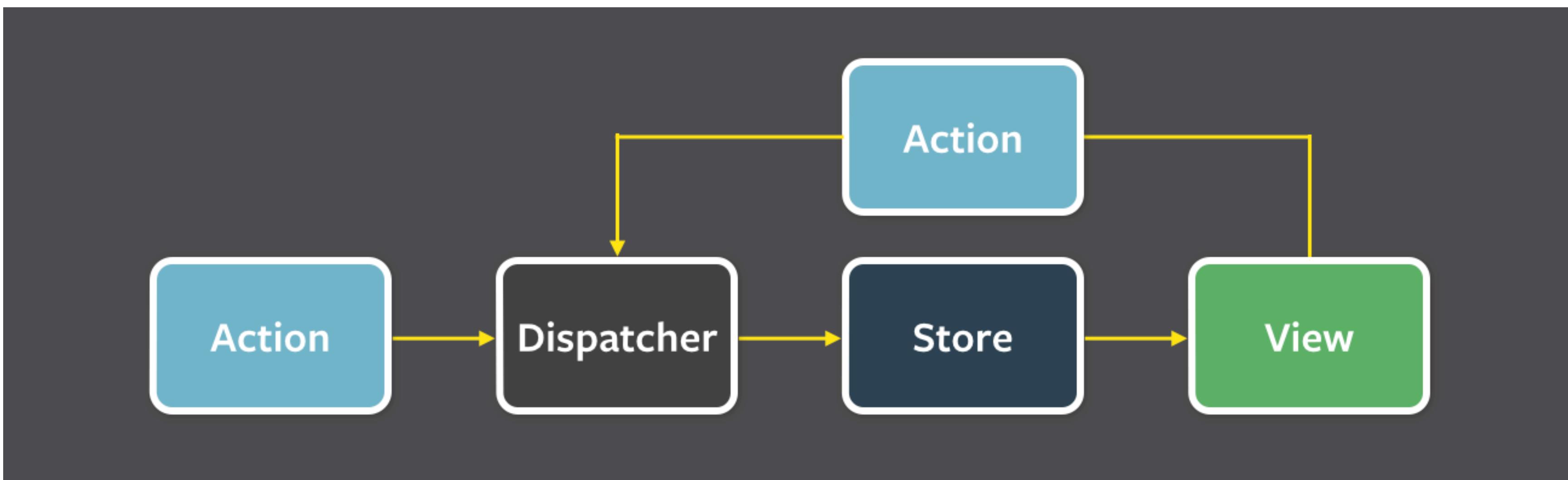
React Components are bricks, not building



Problem of MVC



Flux



Drawbacks of Flux

- Inter-dependency among multiple stores
- Hard to support server-side-rendering
- Hard to do hot-reloading

Redux = Reducer + Flux

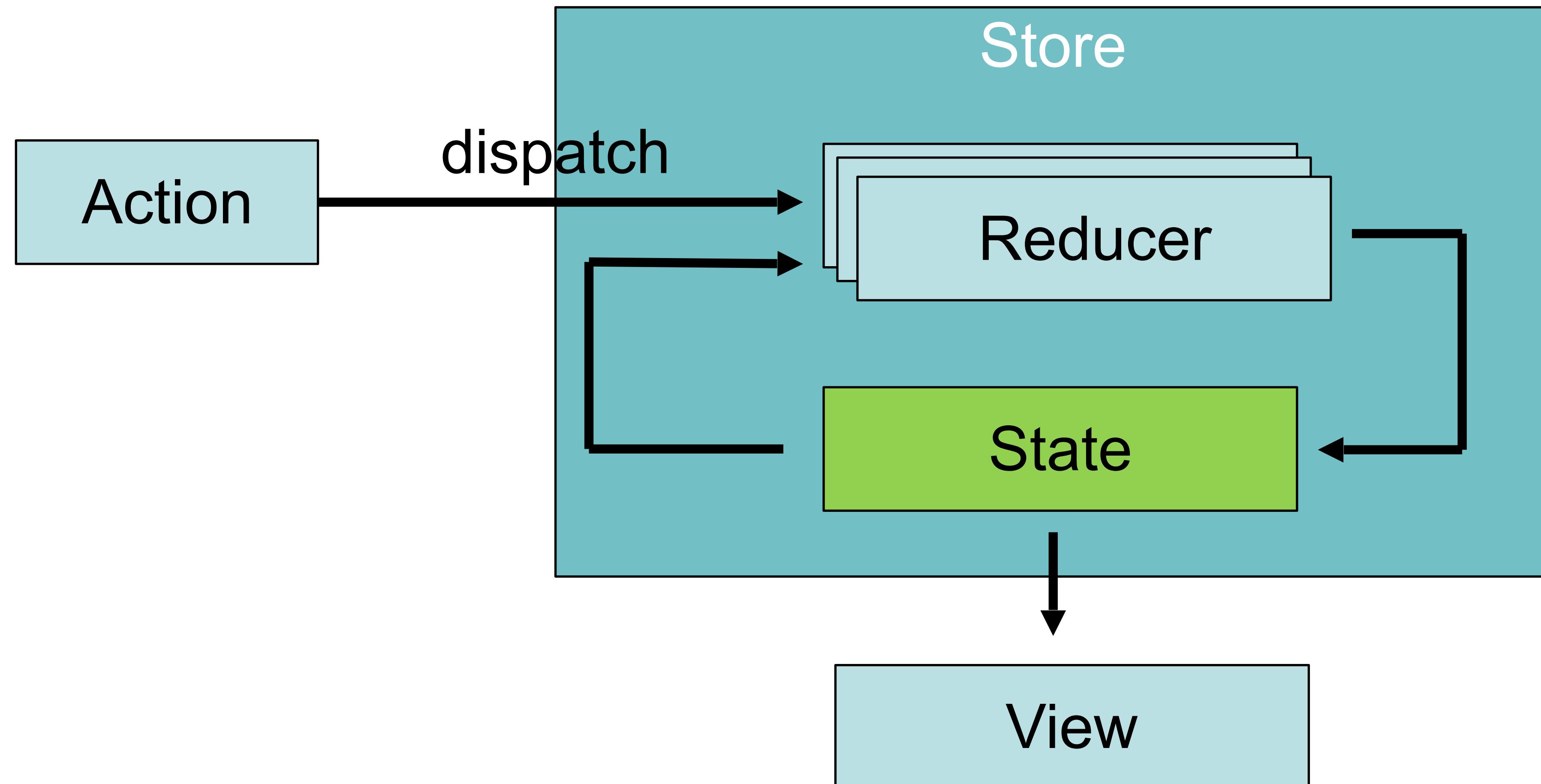
#velocityconf

O'REILLY®
Velocity

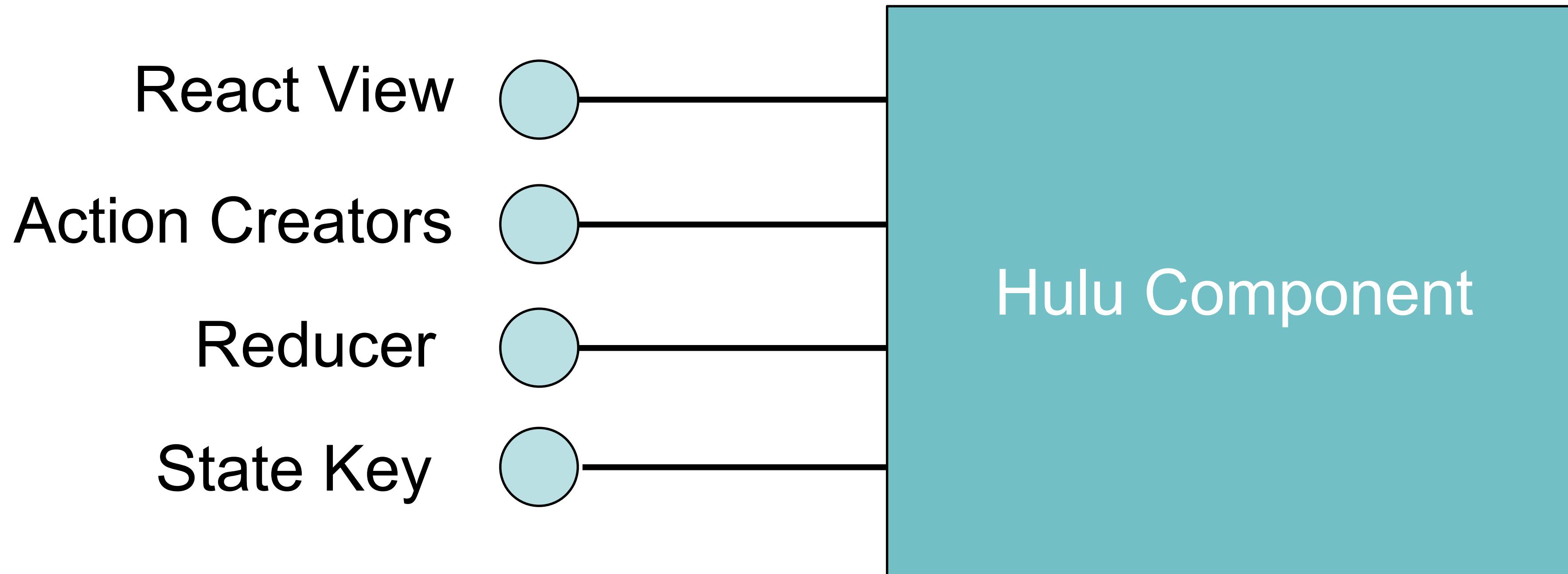
Reducer

(state, action) => newState

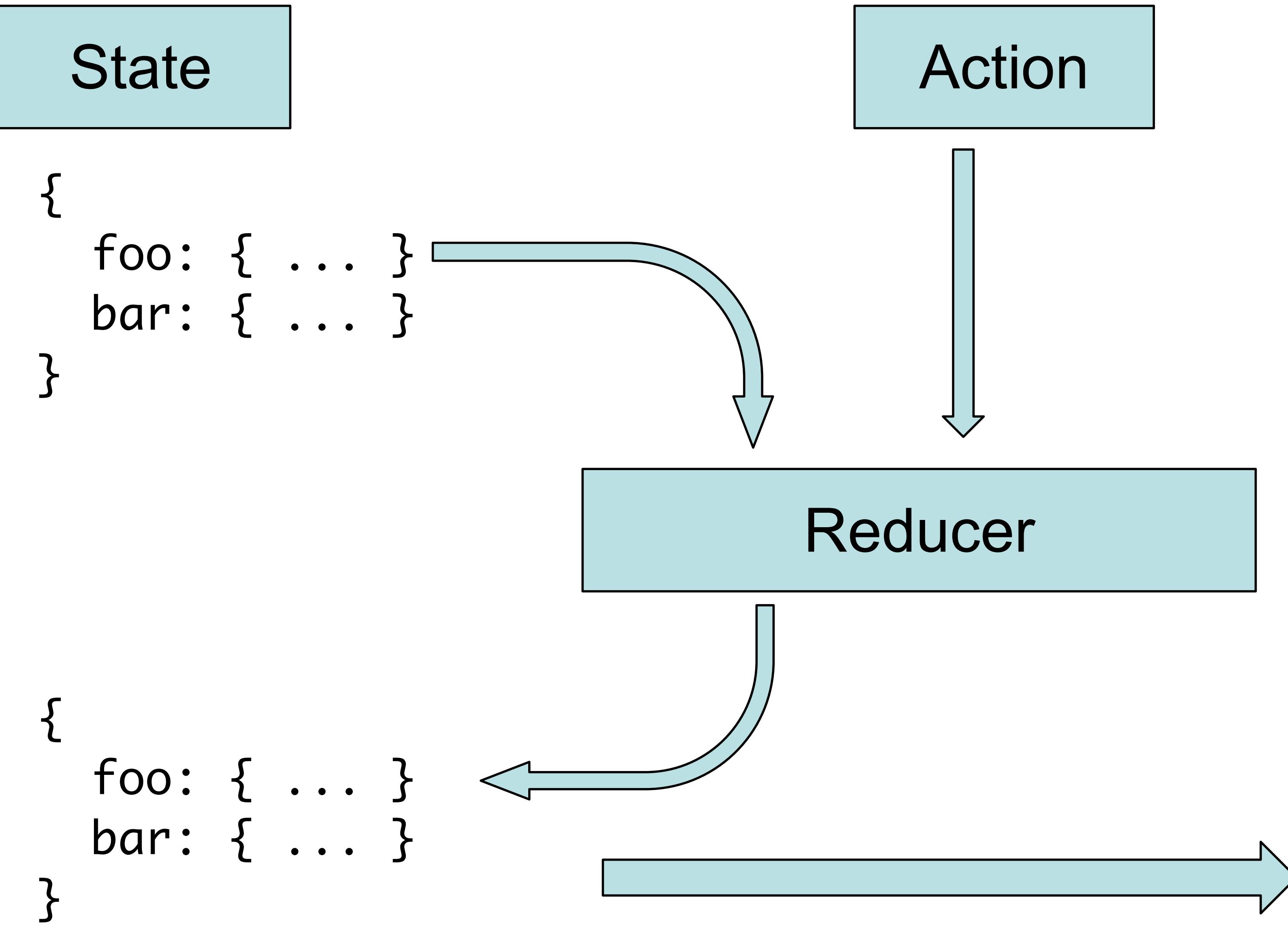
Redux



From Perspective of Component



```
export {view, actions, reducer, stateKey}
```



React + Redux = Win!

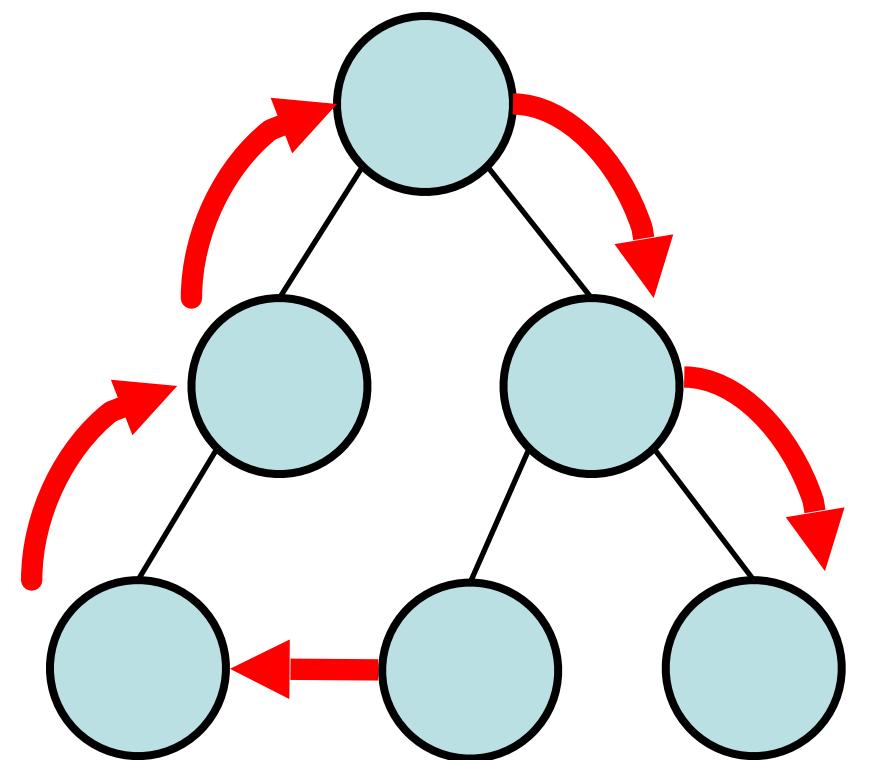
#velocityconf

O'REILLY®
Velocity

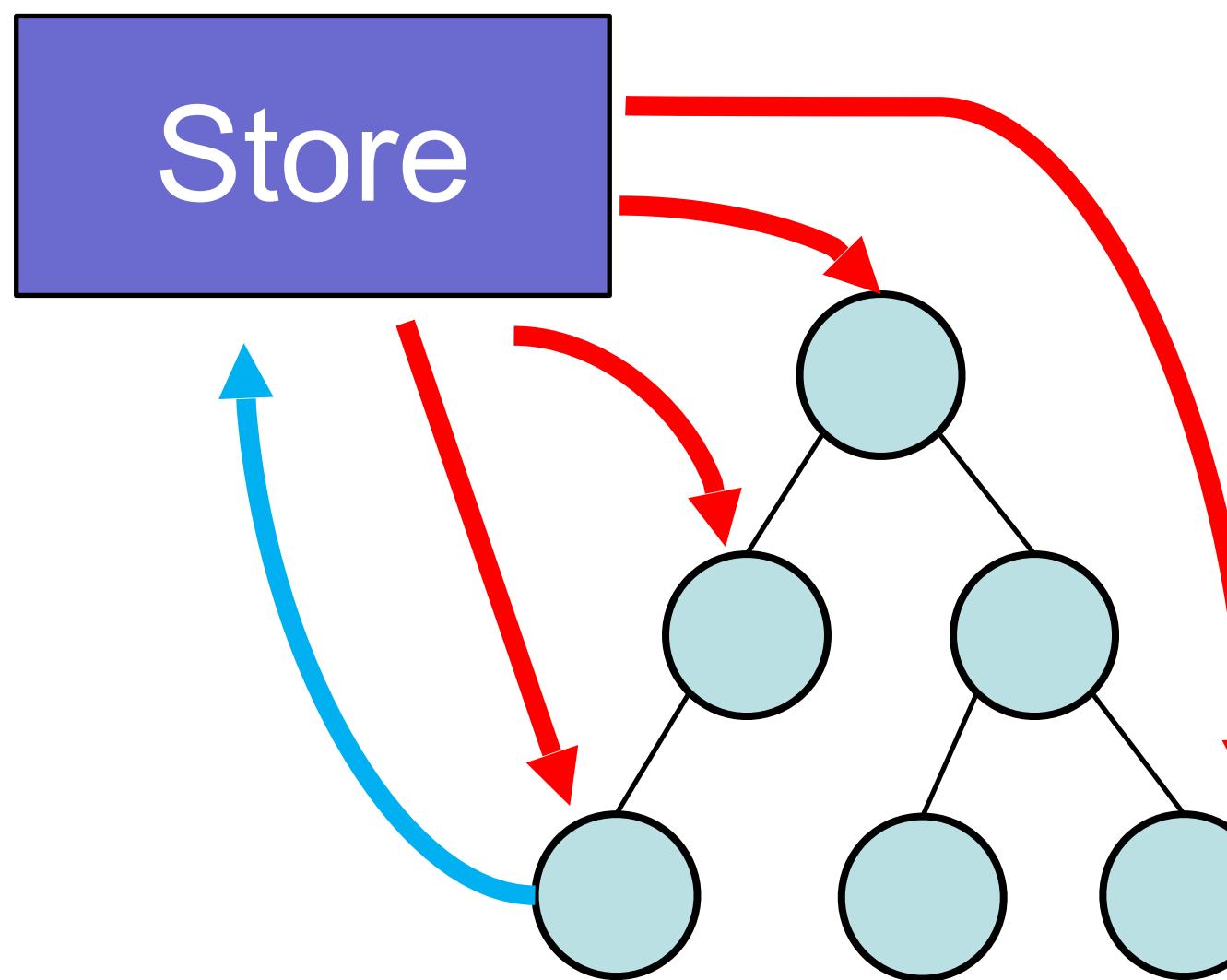
We get *Performance*
and *Maintainability* at
the same time with
React & Redux



Without Redux



With Redux



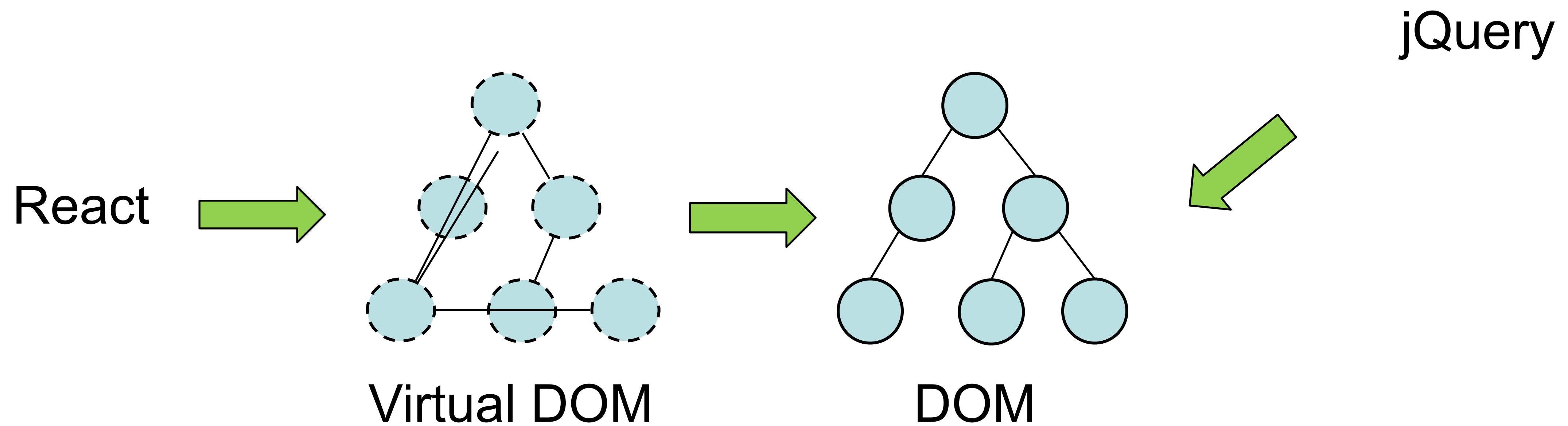
Constraints over Conventions



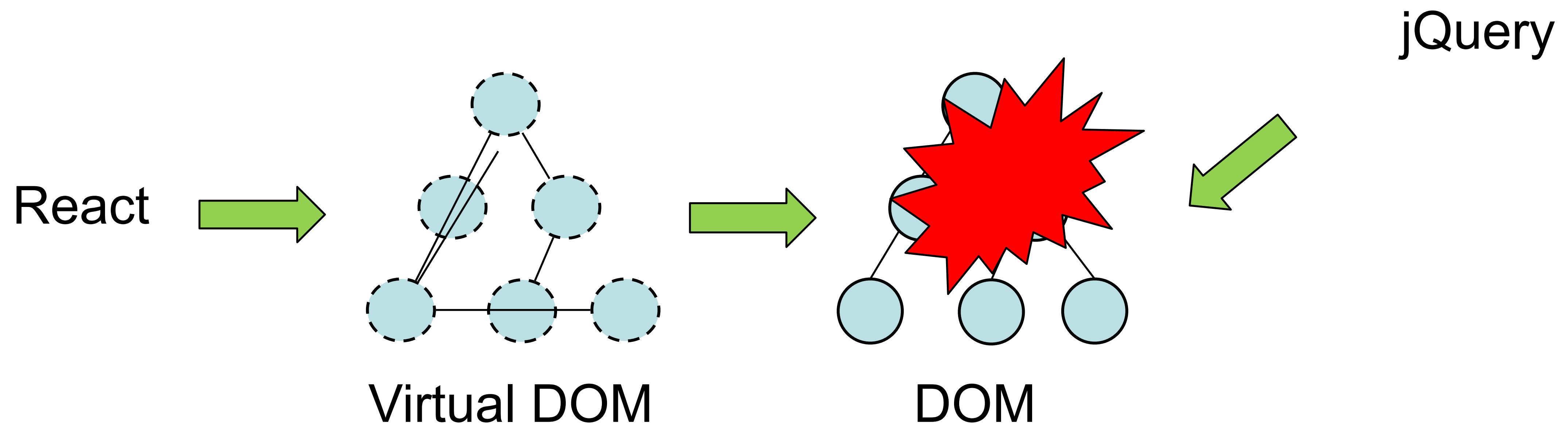
- jQuery Dies Hard
- Legacy Code
- Third Party JavaScript Code
 - e.g. Optimizely



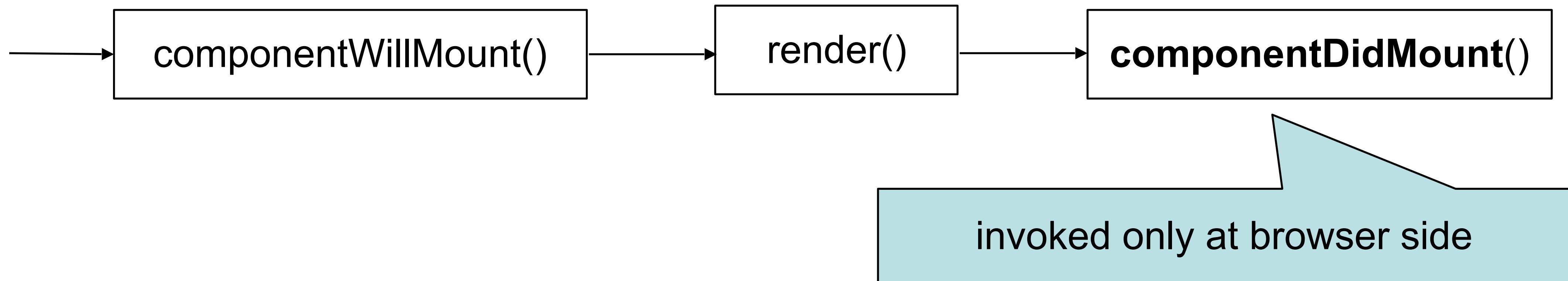
React conflicts with jQuery



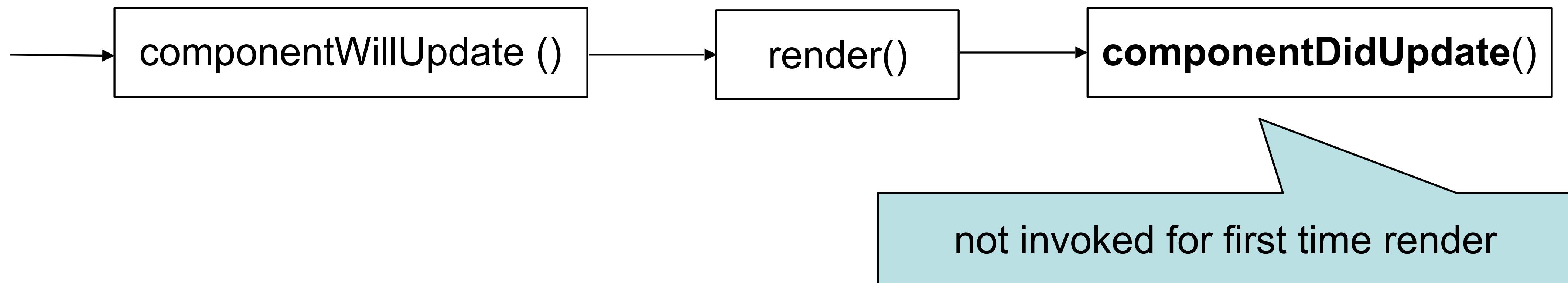
React conflicts with jQuery



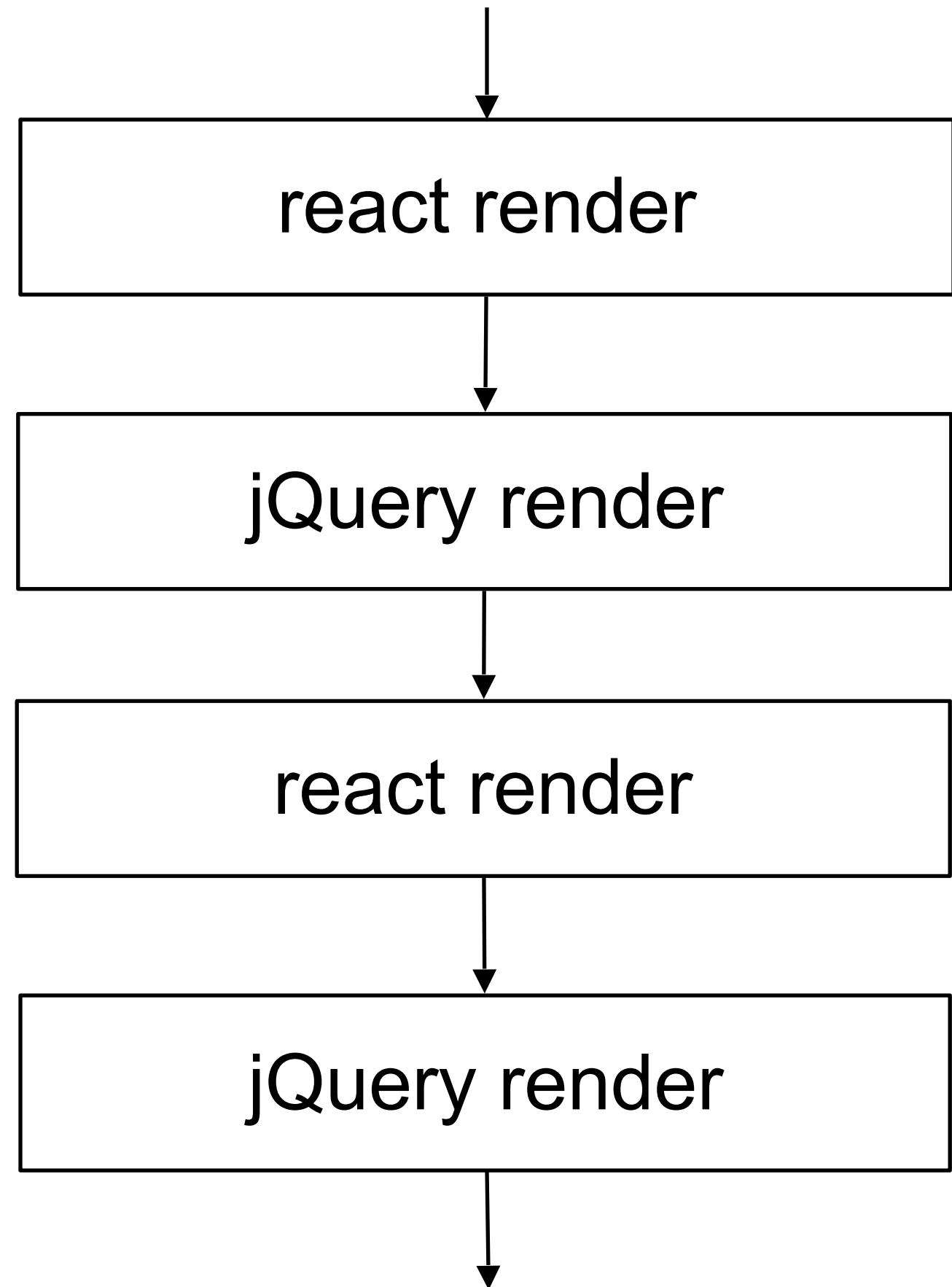
React Mounting Phase



React Updating Phase



```
class Foo extends React.Component{  
  componentDidMount() {  
    // do DOM manipulation with jQuery  
  }  
  
  componentDidUpdate() {  
    // do DOM manipulation with jQuery  
  }  
  
  render() {  
    // render JSX for react  
  }  
}
```



Dehydrate & Rehydrate



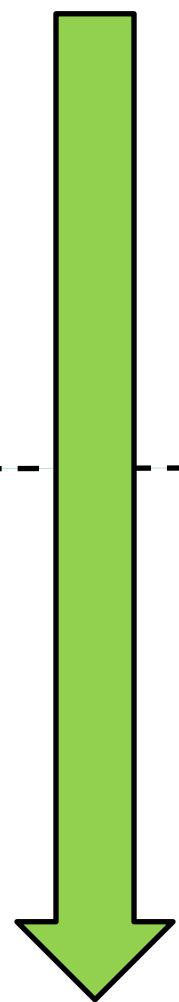
#velocityconf

O'REILLY®
Velocity

$$\text{State} + \text{React} = \text{HTML}$$

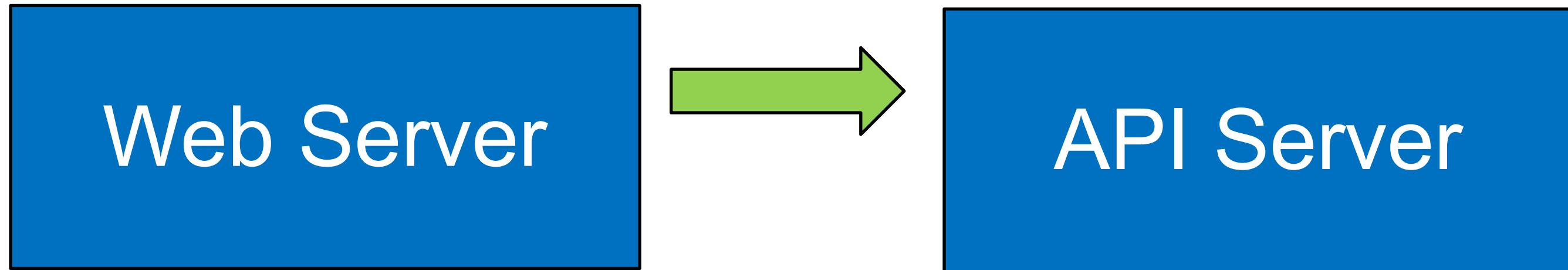
Server Side

Browser Side



$$\text{Dehydrated State} + \text{React} = \text{Rehydrated HTML}$$

API Request

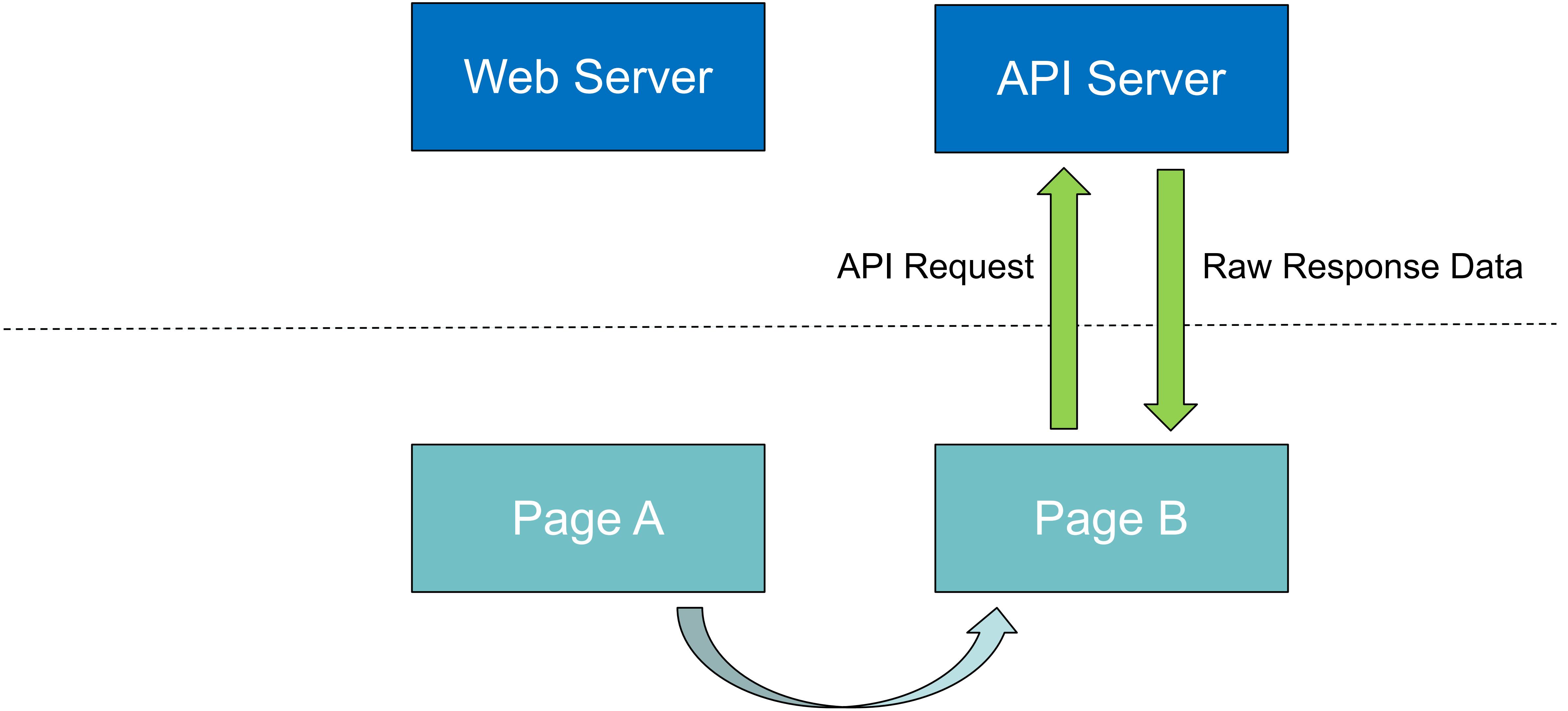


Page Request

HTML
&

Dehydrated Data

Page A



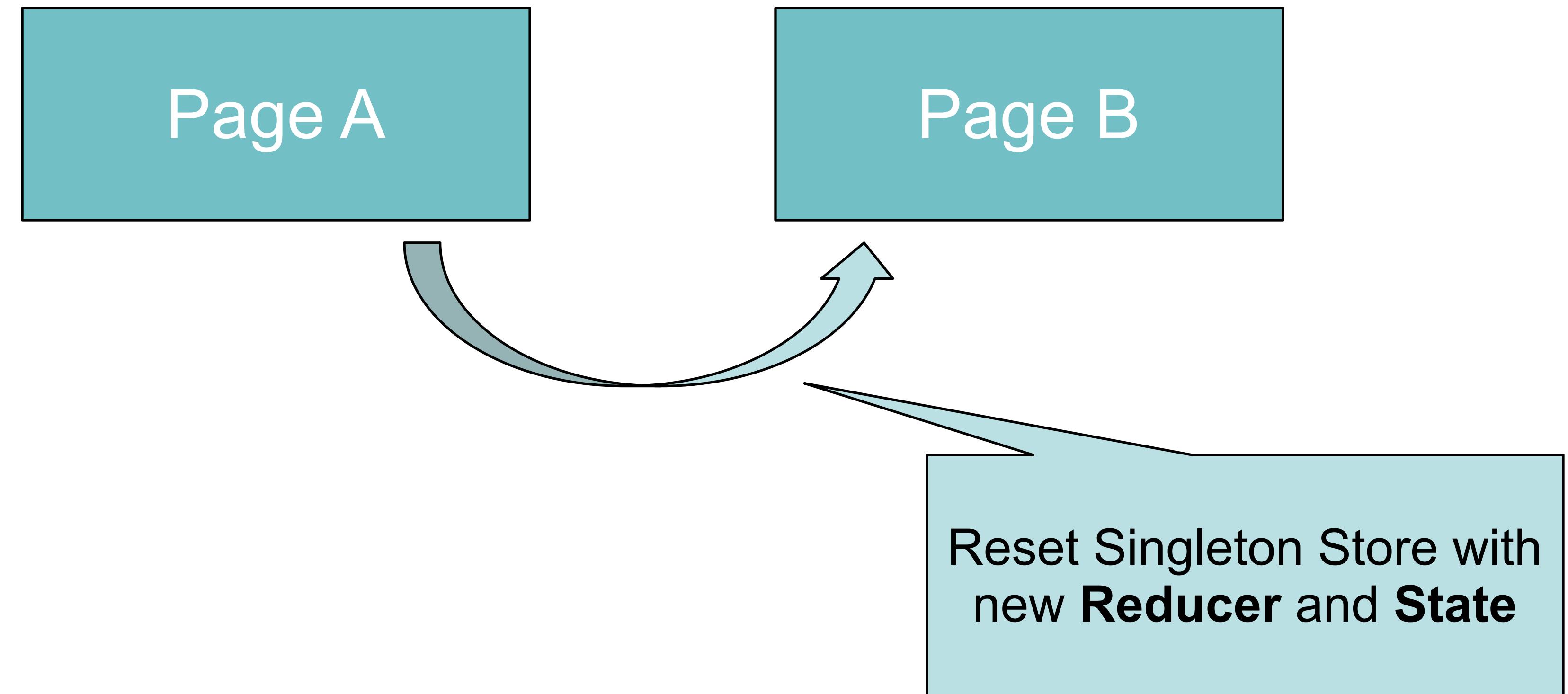
Chunked Code



#velocityconf

O'REILLY®
Velocity

On SPA Navigation



Faster React Server Rendering



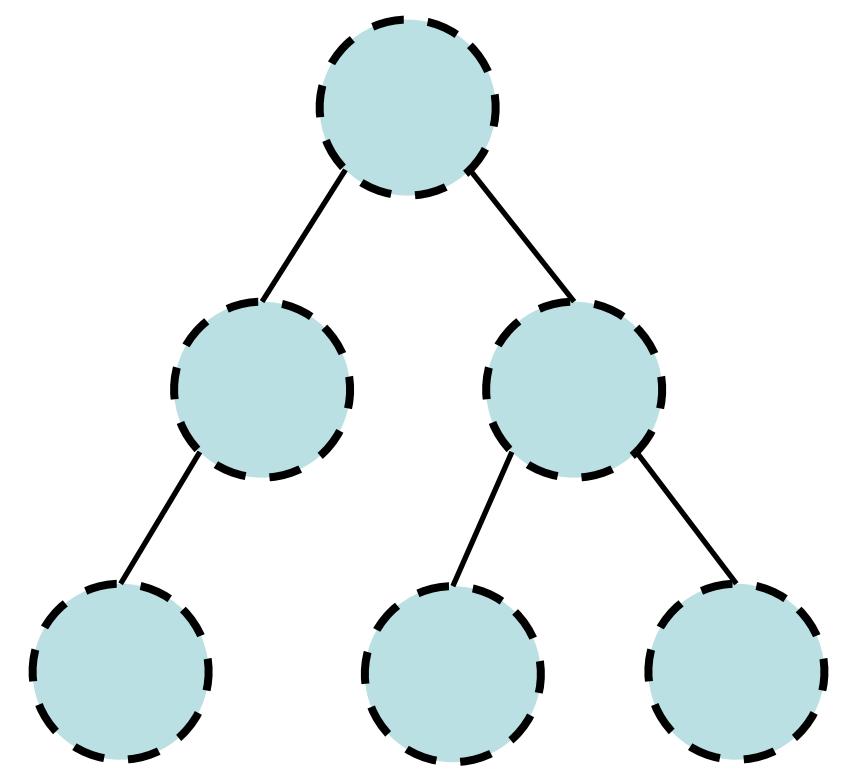
#velocityconf

O'REILLY®
Velocity

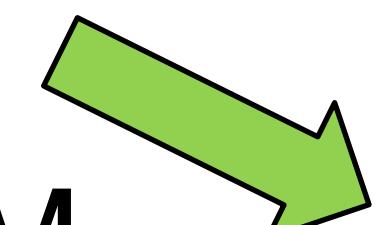
- Set Production Mode
 - **NODE_ENV=production node app.js**
- ~~Use Minified React.js~~
 - `require('react/dist/react.min.js');`
 - After v0.14, **DO_NOT_USE_OR_YOU_WILL_BE FIRED**
- Babel-react-optimize Preset
 - <https://github.com/thejameskyle/babel-react-optimize>

Avoid Wasted-Rendering-Time

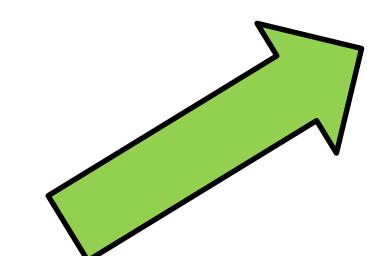
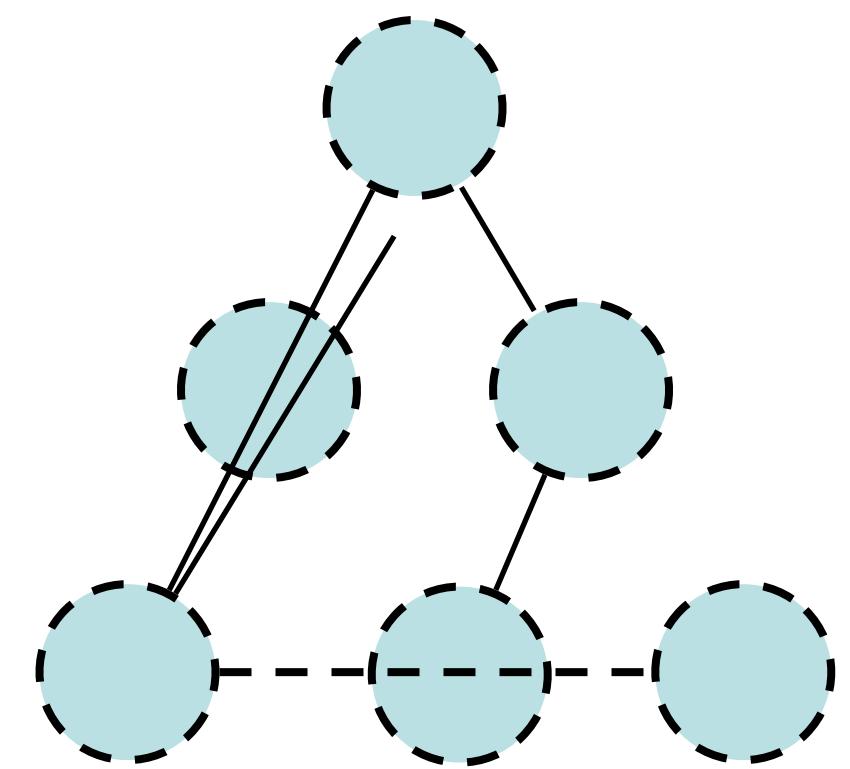




Previous Virtual DOM



Waste If Same



Virtual DOM

Avoid Dynamic *props*

```
<Foo onClick={ function() { doSomething(); } } />
```



```
<Bar style={ {margin: "0px" } }/>
```



Incremental Page Rendering



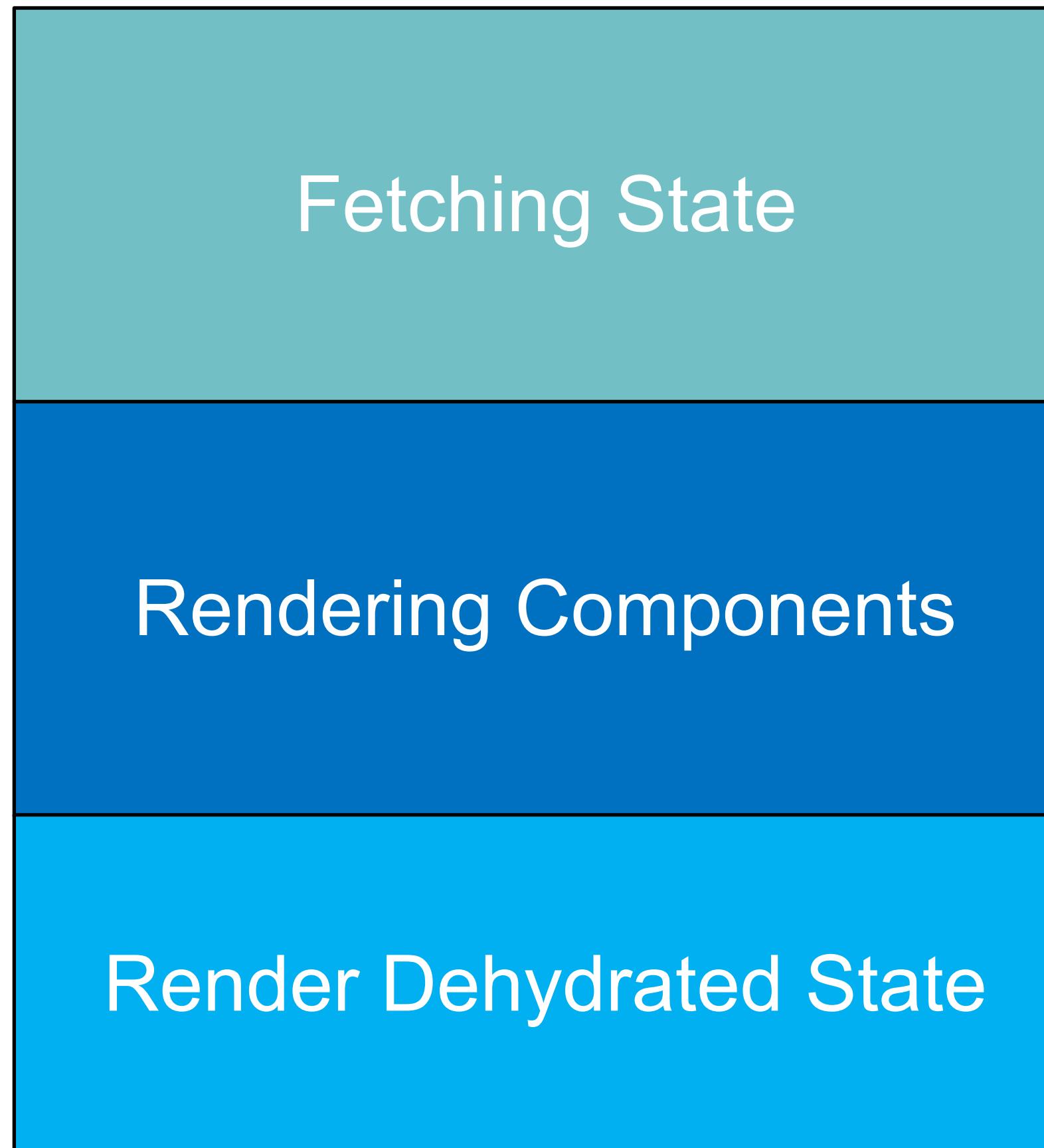
React-Router-Redux

```
<Provider store={store} >
  <Router history={history}>
    <Route path="/" component={App}>
      <Route path="foo" component={Foo} />
      <Route path="bar" component={Bar} />
    <Route>
  </Router>
<Provider>
```



Maps Path to Page Component

Server Rendering Process



`fetchState().then(render)`

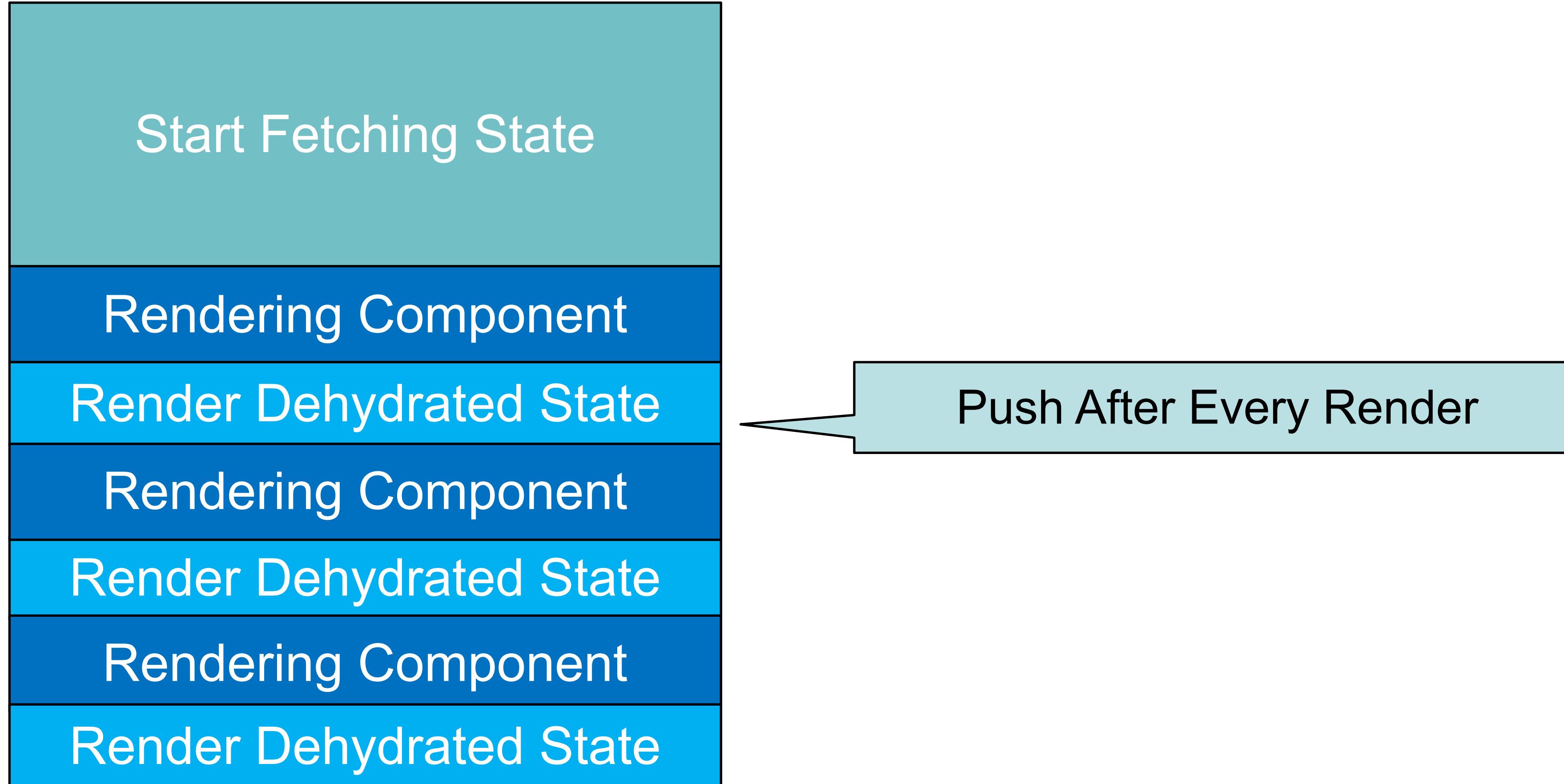
`ReactDOMServer.renderToString(...)`

```
<script>
  window.initState = ...
</script>
```

Page Should Not Be Component

```
class Page {  
  
    getComponents() {  
        return [  
            Header,  
            Foo,  
            Bar,  
            Footer  
        ];  
    }  
}
```

Server Rendering Process Improved



Constraint over Conventions

#velocityconf

O'REILLY®
Velocity

Search **Hulu** in WeChat



Thanks
Any Question?