Introduction to React

Zürich ReactJS Meetup · February 25th, 2015

Andreas Moor


Saul Maddox
@poeticninja
Follow
Internet trends after the @reactjs conference.
#JavaScript :)
gifbin.com

Agenda

Presentation

- What is it?
- Hello, React!
- How does it work?
- Why should I use it?

Hands-on

- Problem statement
- Solution
- Discussion

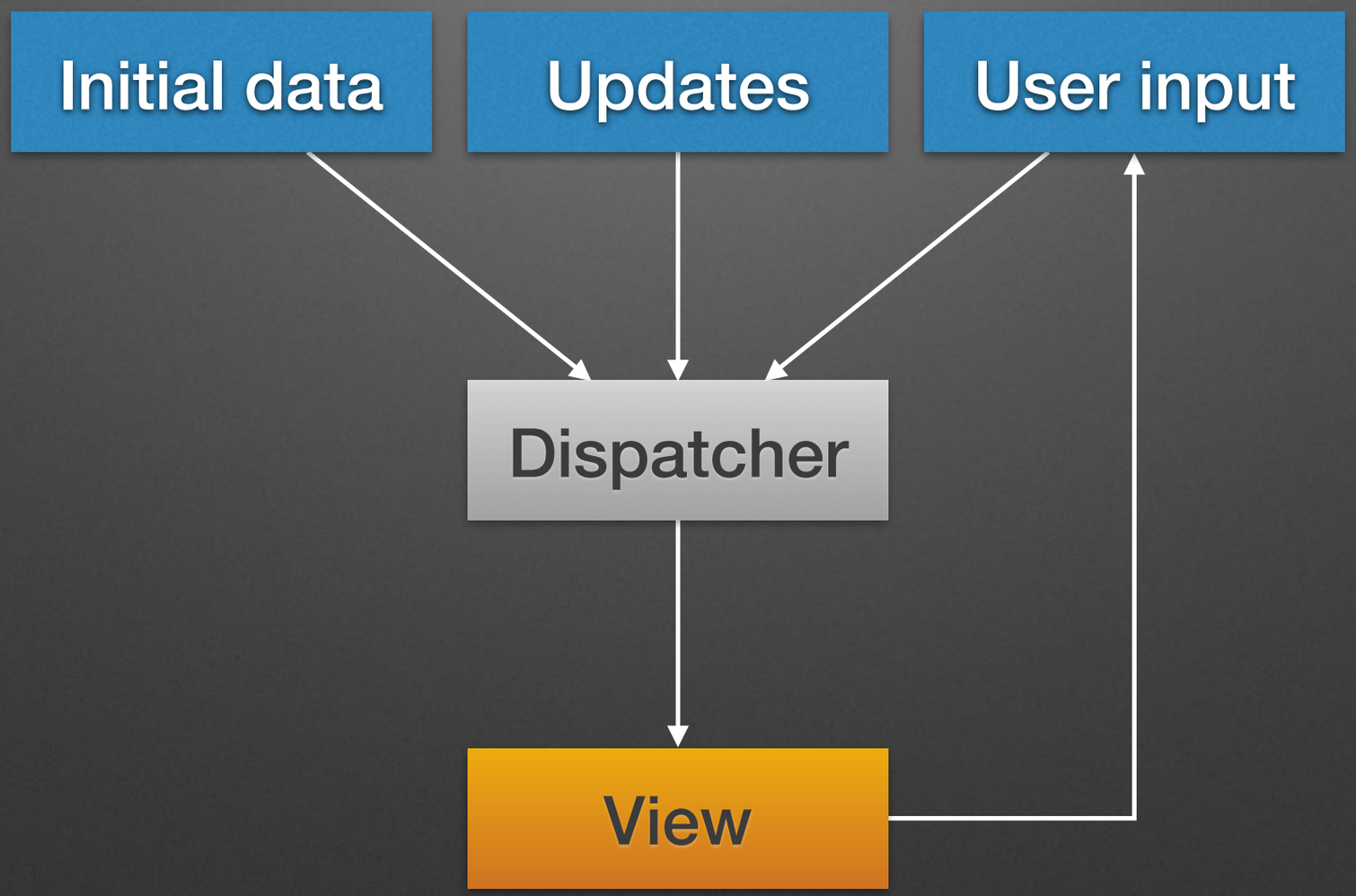
Drinks

What is it?

- JavaScript library for building user interfaces
- Developed by Facebook, used on facebook.com, Instagram, iOS app
- Around since 2013
- Open source (BSD license)
- Embraces “reactive” approach to state management:

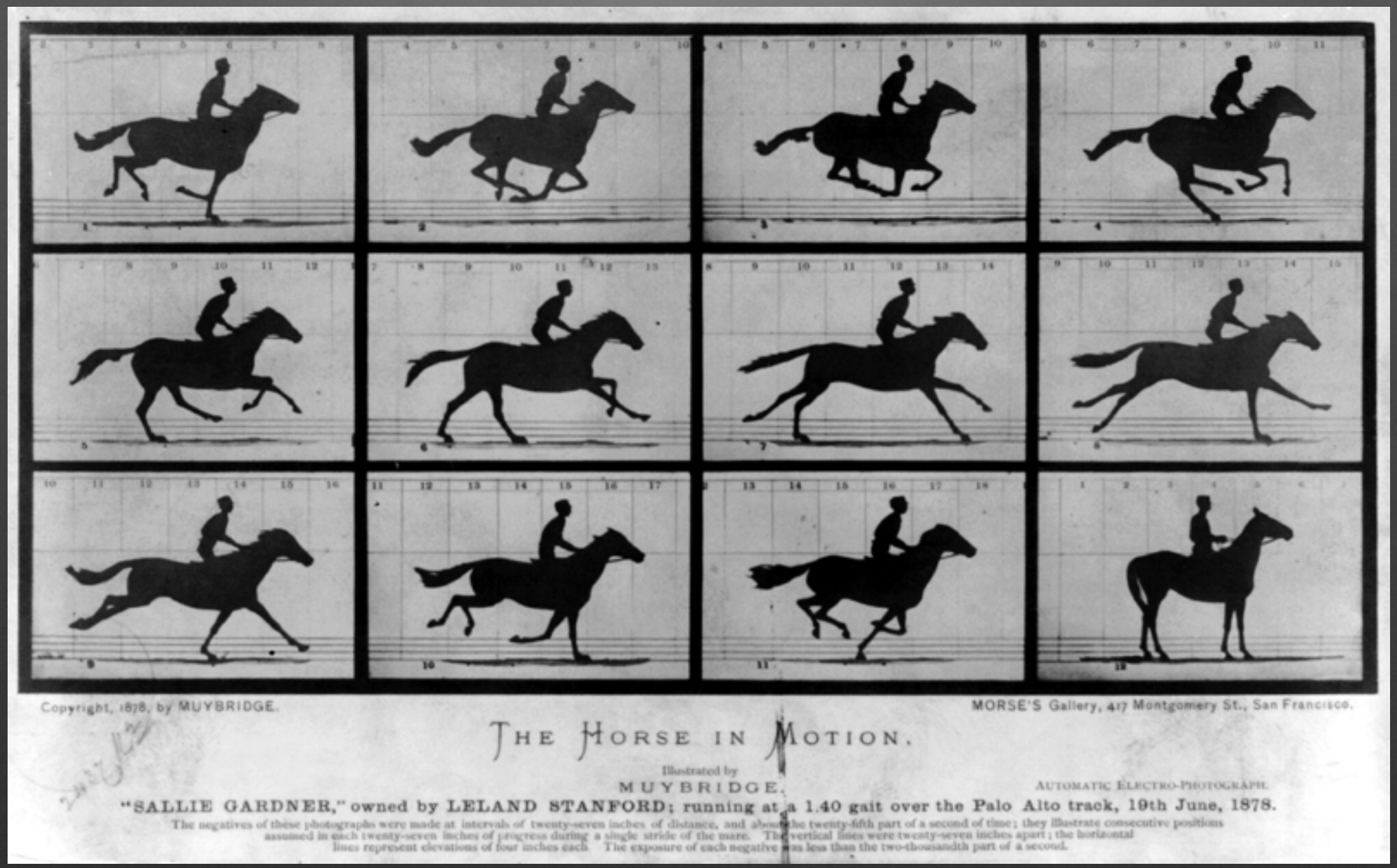
> “… reactive programming is a programming paradigm oriented around data flows and the propagation of change.” —Wikipedia

Application structure



Mutable DOM

The React way



Hello, React

```
// specify component
var HelloWorld = React.createClass({
  render: function() {
    return React.DOM.h1(null,
                        this.props.msg);
  }
});

// mount instance into DOM
React.render(React.createElement(HelloWorld,
                                 {msg: 'Hello, World!'}),
             document.getElementById('app'));
```

Hello, JSX

```
var HelloWorld = React.createClass({
  render: function() {
    return <h1>
             {this.props.msg}
           </h1>;
  }
});


React.render(<HelloWorld msg='Hello, World!' />,
             document.getElementById('app'));
```

JSX

- JSX must be transformed to JS before execution
- Can be done offline or in browser (JSXTransformer.js)
- Completely optional
  - `React.createElement('div', *props*, *children*)`
  - `React.DOM.div(*props*, *children*)`
  - `<div *props*>*children*</div>`

Hello, state

```
var CountingButton = React.createClass({
  getInitialState: function() {
    return {count: 0};
  },
  _handleClick: function() {
    this.setState({count: this.state.count+1 });
  },
  render: function() {
    return (
      <div>
        <h1>{this.state.count}</h1>
        <button onClick={this._handleClick}>Increment</button>
      </div>
    );
  }
});
```

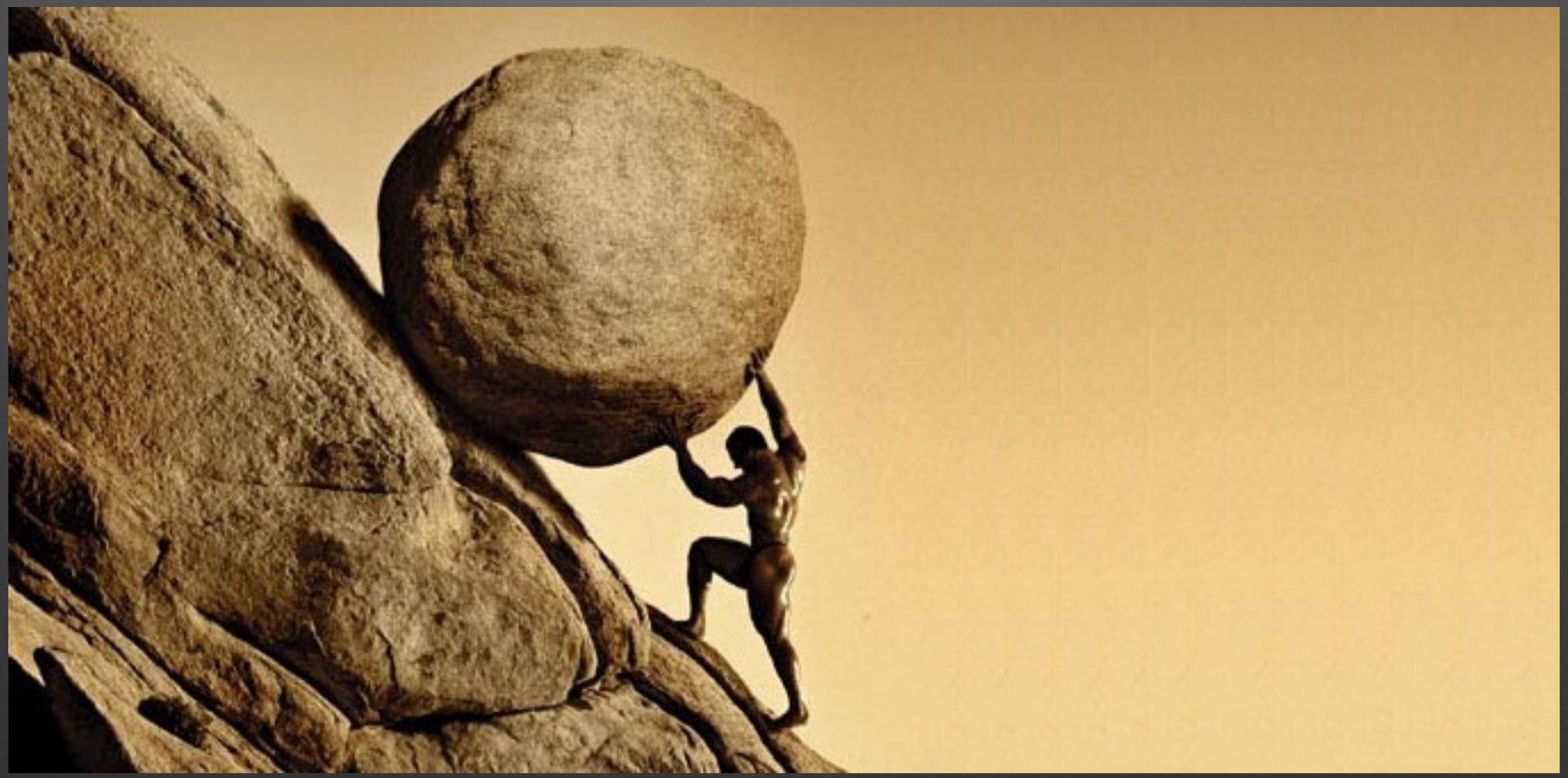
Components

- `render` method describes what component looks like at any point in time
- State machine defined by props and state
- Other methods may be implemented to hook into lifecycle stages (e.g. `getInitialState`, `componentDidUpdate`)

Data flow

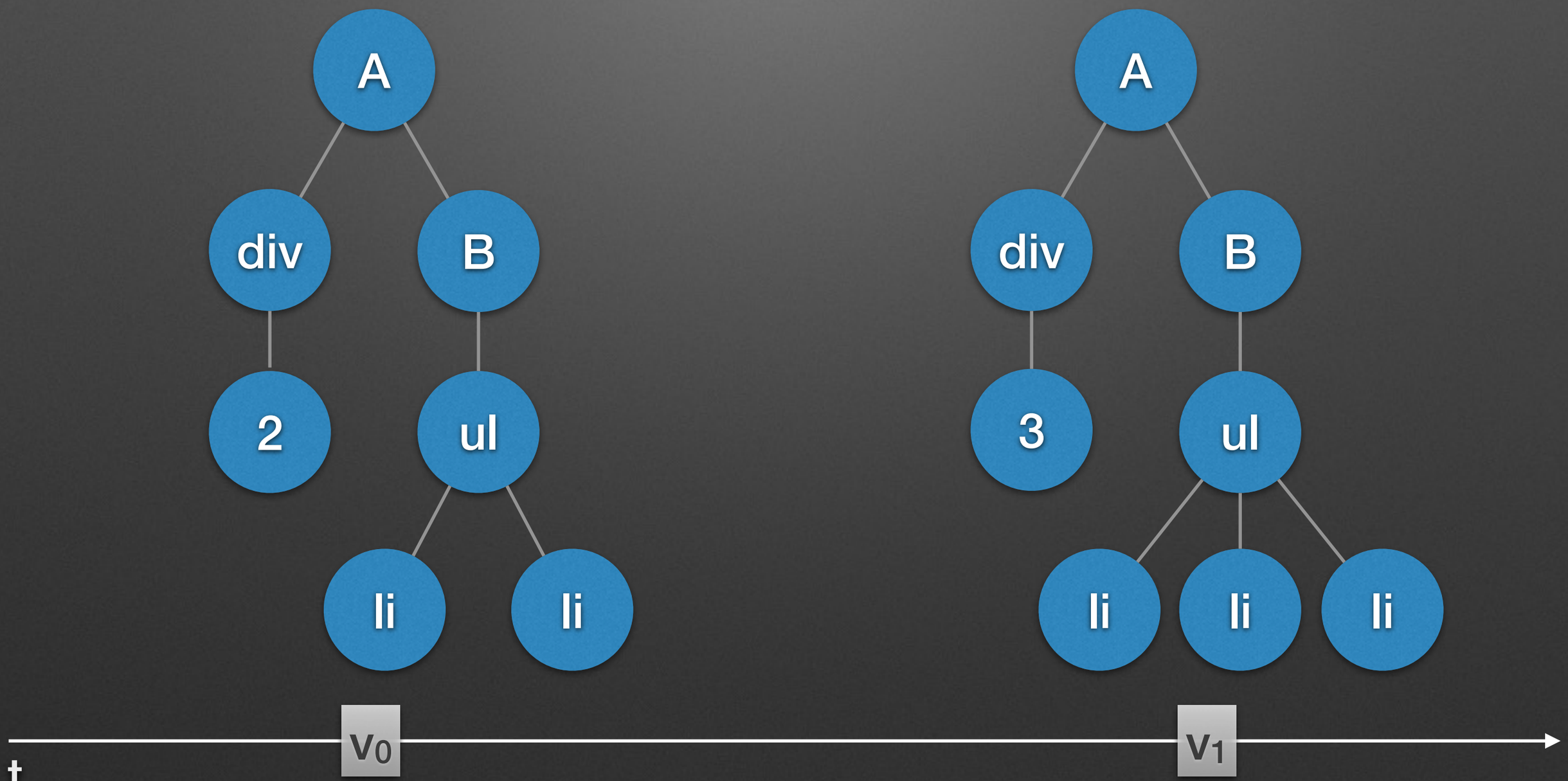
- Props are immutable and flow down
  the component tree, from owner to ownee
- State is mutable and internal to the component
- Call to `this.setState` triggers re-rendering
  of entire (sub)tree

How does it work?



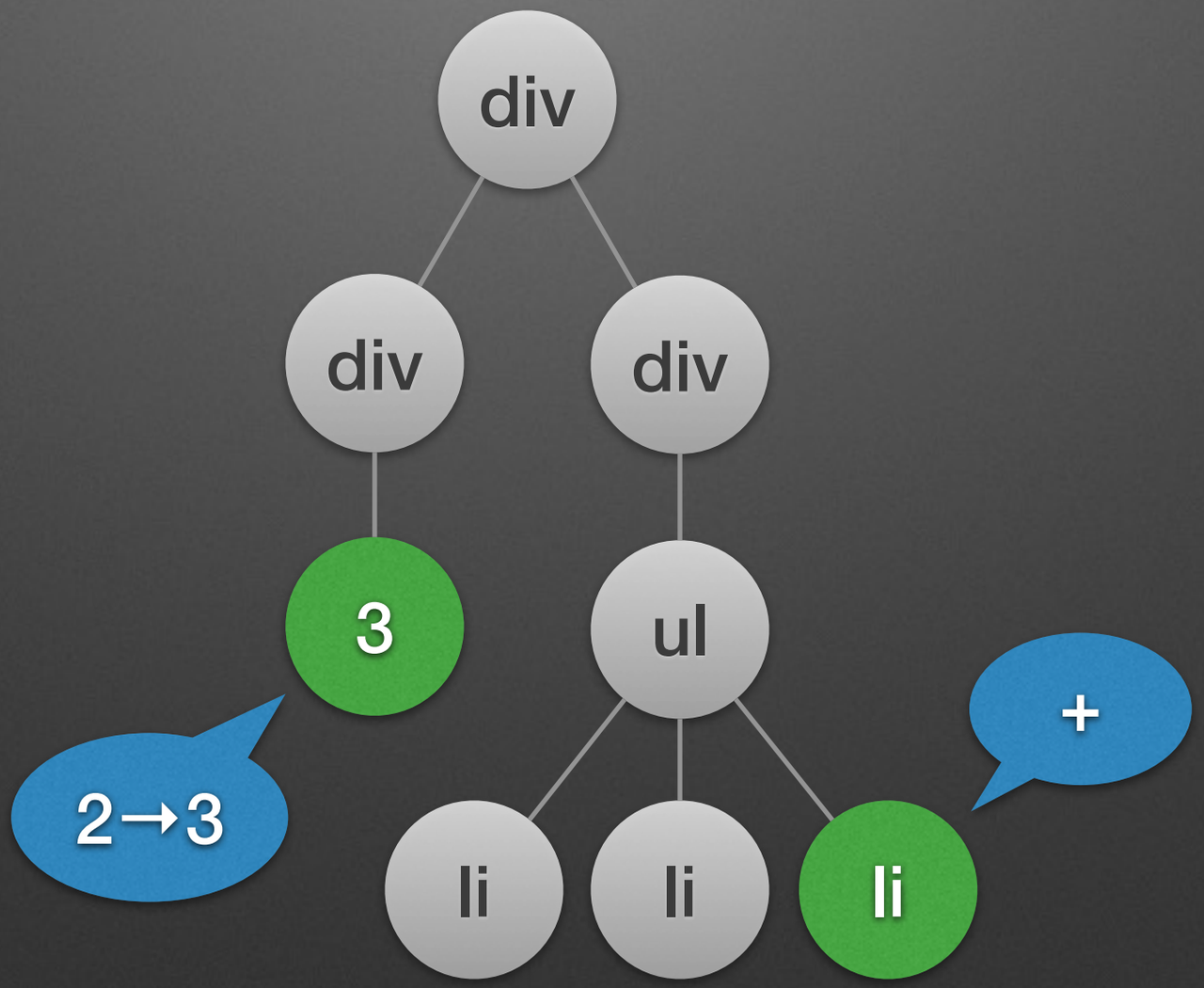
Virtual DOM: render

f(props, state) = Virtual DOM v_t



Virtual DOM: reconciliation

$diff(v_{t-1}, v_t) \rightarrow$ DOM operations $\rightarrow DOM_t$



Virtual DOM: summary

- Abstraction over DOM
- Translates declarative code into DOM API calls to manipulate real DOM
- Diffing of virtual DOMs minimizes set of operations necessary to produce next UI view

Events

- Event handlers are passed to components as camelCased props

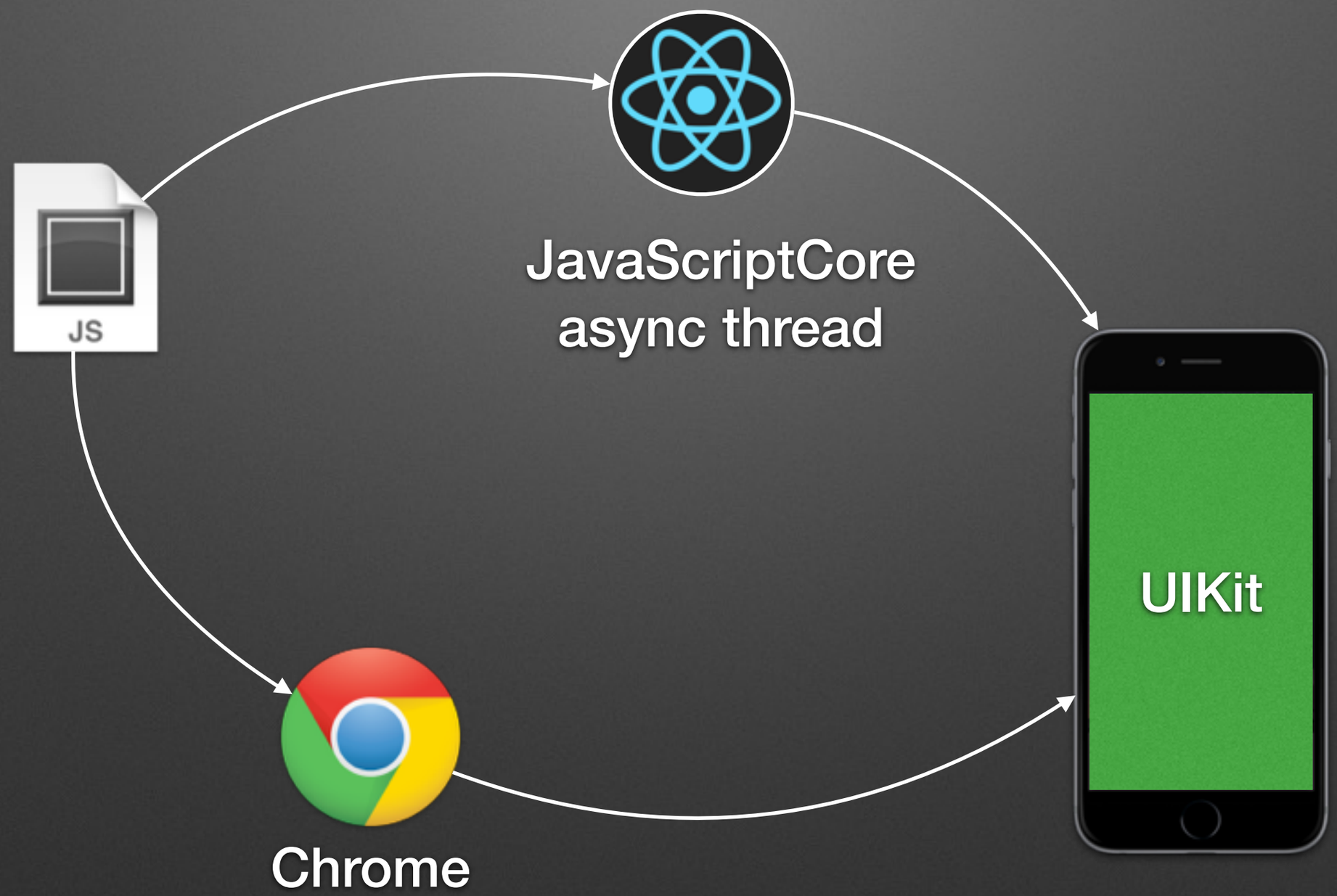
```
<a onClick={this._handleClick}>Click!</a>
```

- Autobinding of handlers to component instance
- Handlers receive cross-browser synthetic events consistent with W3C spec
- Event delegation: Single event listener, has map of all nodes

Why should I use React?

- Simple
- Declarative
- Composable
- Fast

Coming soon: React Native



Hands-on

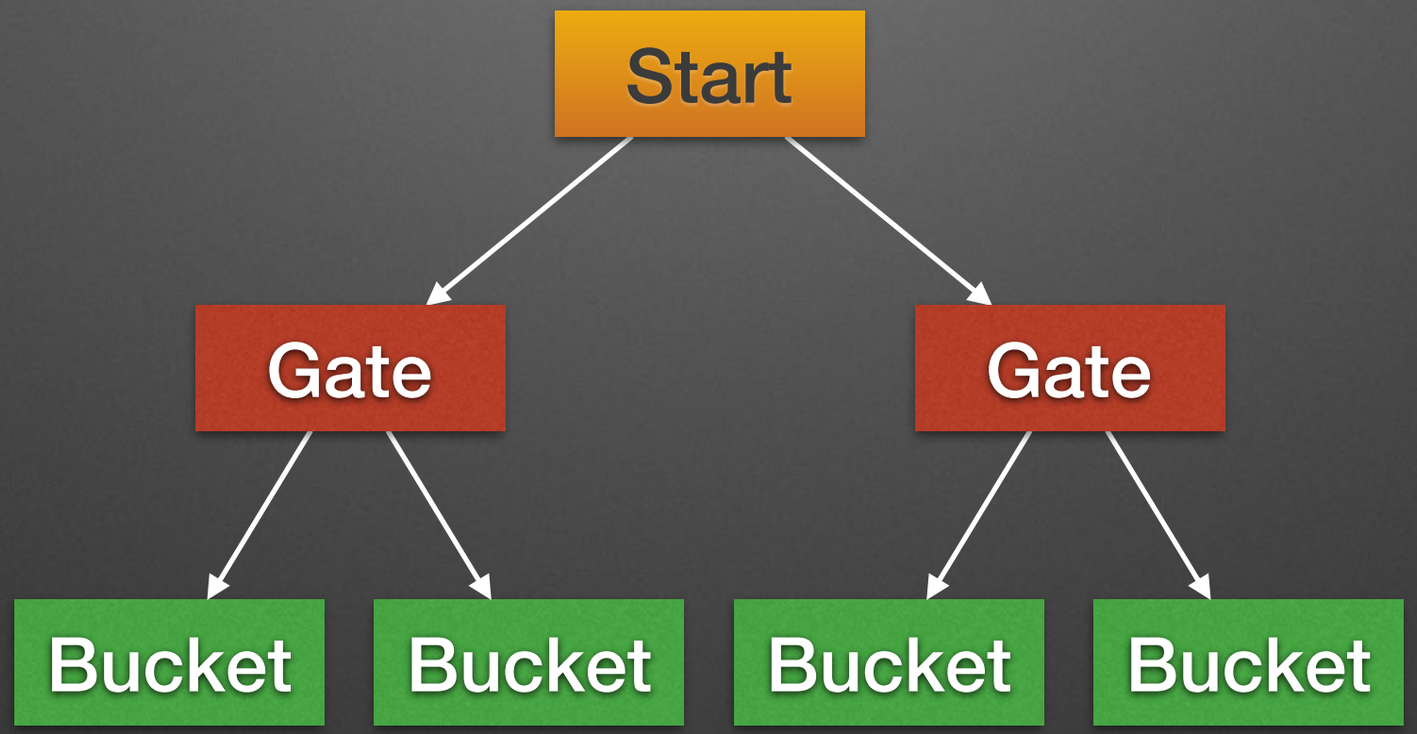
1. Clone
   ```
   git clone https://github.com/garamond/react-intro.git
   ```
2. Start
   ```
   python -m SimpleHTTPServer
   ```
3. Read
   ```
   http://localhost:8000
   ```
4. Write
   ```
   $EDITOR app.jsx
   ```

Thinking in React

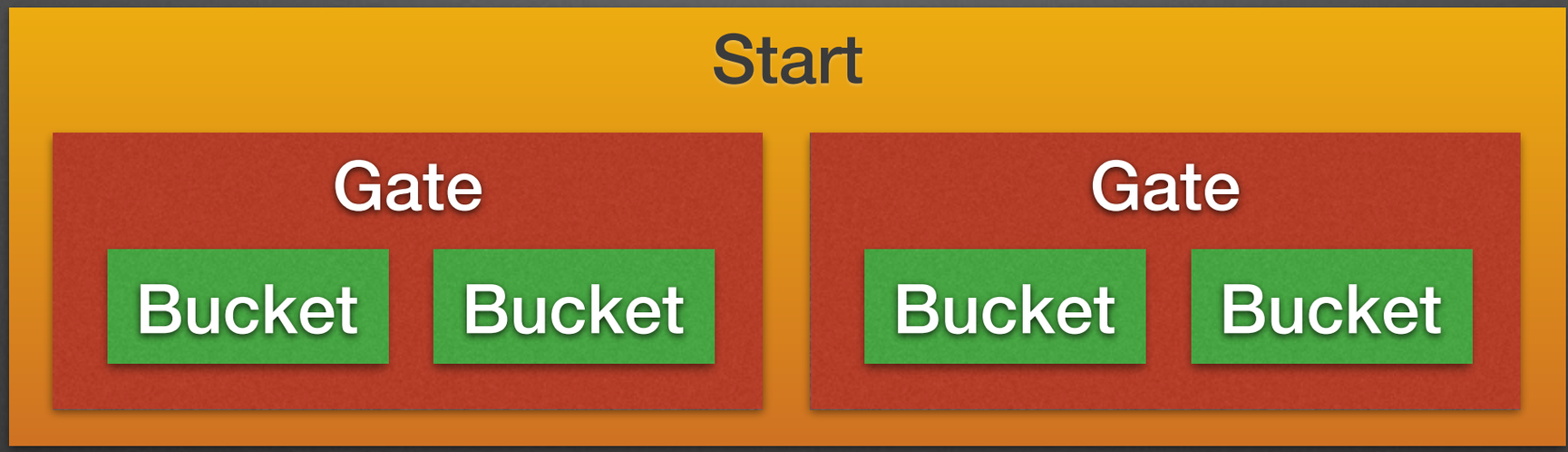
1. Start with a mock
2. Break the UI into a component hierarchy
3. Build a static version in React
4. Define state and wire up components

Adapted from: http://facebook.github.io/react/docs/thinking-in-react.html

Components: hierarchy



Components: ownership



More React

- Removing User Interface Complexity, or Why React is Awesome
  bit.ly/react-awesome
- Decomplexifying Code with React (video)
  bit.ly/react-decomplexify
- Introducing React Native (video)
  bit.ly/react-native

Andreas Moor
reactzh@grmnd.ch

github.com/garamond/react-intro

meetup.com/Zurich-ReactJS-Meetup/events/220455804/