

React

For Telenor Comoyo
 March 31th 2014



Tor Arne Kvaløy





Major game changers

- 1999 Java
- 2002 XML
- 2004 Log4j
- 2005 JUnit
- 2006 Maven
- 2006 Spring
- 2007 Tomcat
- 2008 JQuery
- 2014 React



Current challenges

- Long distance between template and model
- Basically separating technologies
 - HTML/JSP and jquery
 - Mustache/Handlebars and JS-model
 - Bindings
- Hard to test







React

- Components
 - "Template" and "model"
 - Inputs
 - State
 - Render themselves
 - Composable
- Testable
- Pre-rendered on the server



Hello World (component)

```
<div id="hello"> </div>
```

```
var Hello = React.createClass({  
  render: function() {  
    return span(null, "Hello World");  
  }  
});  
React.renderComponent(Hello(), document.getElementById("hello"));
```



Test

```
var hello = RTU.renderIntoDocument(Hello());
```

```
assert.equals(  
    hello.getDOMNode().firstChild.innerHTML,  
    "Hello World");
```




Props and javascript

```
var List = React.createClass({
  render: function() {
    return ul(null,
      this.props.names.map(function(name) {
        return li(null, name);
      })
    );
  }
});
var names = ["James", "Paul", "Phillip"];
React.renderComponent(List({names:names}));
```



Composition and props

```
var Hello = React.createClass({
  render: function() {
    return Heading({text = "My heading"});
  }
});
```

```
var Heading = React.createClass({
  render: function() {
    return h1(null, this.props.text);
  }
});
```



Composition and signature

```
var Hello = React.createClass({
  render: function() {
    return Heading({text = "My heading"},
      span(null, "Hello World"));
  }
});
```

```
var Heading = React.createClass({
  render: function() {
    return div(null,
      h1(null, this.props.text),
      this.props.children
    );
  }
});
```



Events, state, render and fun

```
var ClickMe = React.createClass({
  getInitialState: function() {
    return {clicks: 0};
  },
  handleClick: function() {
    this.setState({clicks: this.state.clicks + 1});
  },
  render: function() {
    return div(null,
      button(onClick: this.handleClick, "Click me"),
      div(null, "Clicked " + this.state.clicks + " times"));
  }
});
```



References

```
var ClickMe = React.createClass({
  getInitialState: function() {
    return {clicks: 0};
  },
  handleClick: function() {
    this.setState({clicks: this.state.clicks + 1});
  },
  render: function() {
    return div(null,
      button({onClick: this.handleClick, ref:"button"}, "Click me"),
      div({ref:"clicks"}, "Clicked " + this.state.clicks + " times"));
  }
});
```



Test button click

```
var clickMe = RTU.renderIntoDocument(ClickMe());
```

```
RTU.Simulate.click(clickMe.refs.button.getDOMNode());
```

```
assert.equals(  
    clickMe.refs.clicks.getDOMNode().innerHTML,  
    "Clicked 1 times");
```

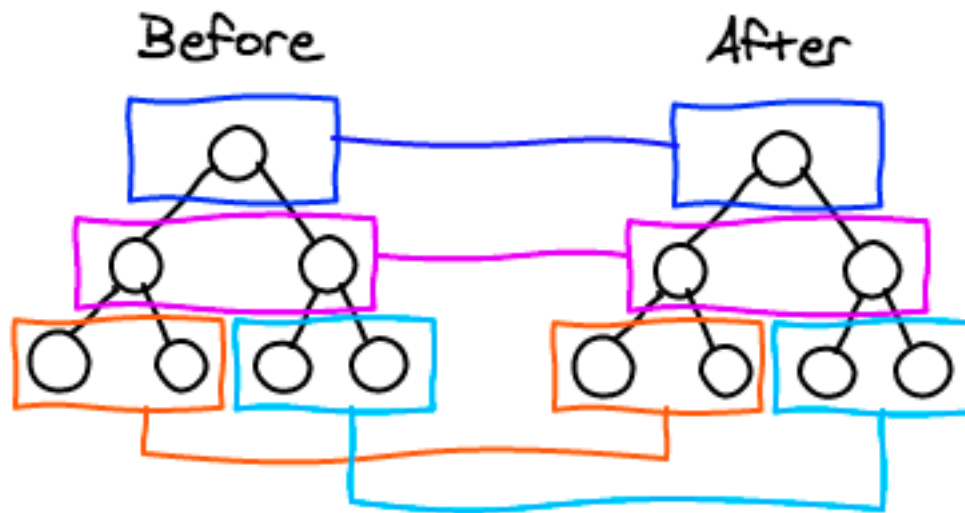


render: function()

- Returns a virtual DOM
- Diffs it with the previous virtual DOM
- Computes the minimal number of updates
- Executes the updates

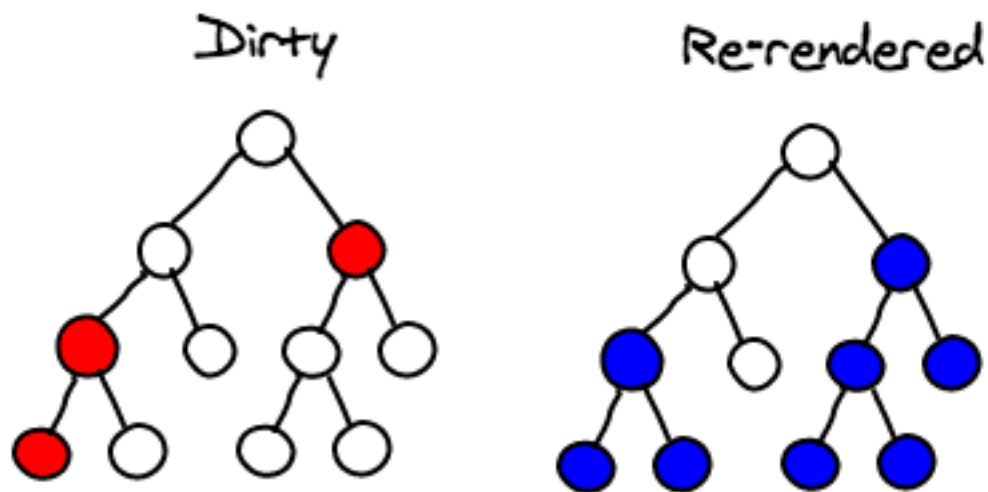


Diff algorithm - Level by level: $O(n)$





Diffs only dirty sub-trees





Pre-render on the server

- Why?
 - Avoid waiting for javascripts to run
 - Search Engine friendly
 - Mobile/3G friendly
- How?
 - `React.renderComponentToString`
 - `data-react-checksum`
 - Browser receives complete HTML
 - Browser initialize React-components and compares the checksums
 - Only event handlers will be attached



State and flow

ClickMe:

```
getInitialState: function() {
  return {click: 0};
},
handleClick: function() {
  this.setState({clicks: this.state.clicks + 1});
},
render: function() {
  return div(null,
    ButtonComp{handleClick: this.handleClick},
    ClicksComp{click: this.state.clicks});
}
```

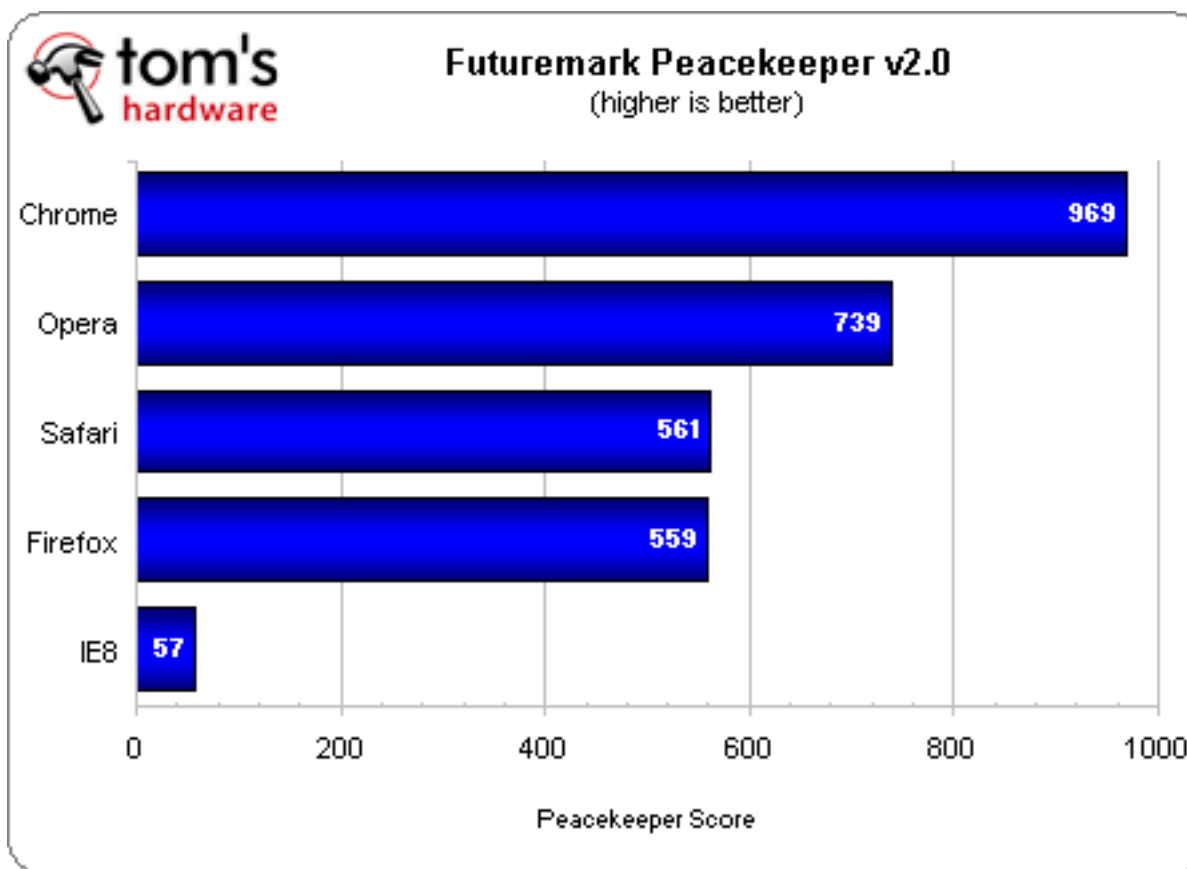
ButtonComp:

```
render: function() {
  return
  button({onClick:this.props.handleClick});
}
```

ClicksComp:

```
render: function() {
  return h1(null, this.props.clicks);
}
```

2,38 % on IE8





Statics

```
var MyComp = React.createClass({  
  statics: {  
    customMethod: function() {  
      return "hello";  
    }  
  },  
  render: function() {  
    ....  
  }  
});
```



Test

```
var myComp = MyComp();
```

```
assert.equals(myComp.customMethod(),"Hello")
```



Size

	Minified	gzip
React	112	31
Angular	104	38
Ember	263	73
JQuery	83	29



React

- Components
 - Inputs
 - State
 - Render themselves
 - Composable
- Testable
- Pre-rendered on the server

