

Discovering the Need for React



Chris Minnick

CEO, WATZTHIS

@chrisminnick www.watzthis.com



Pretest



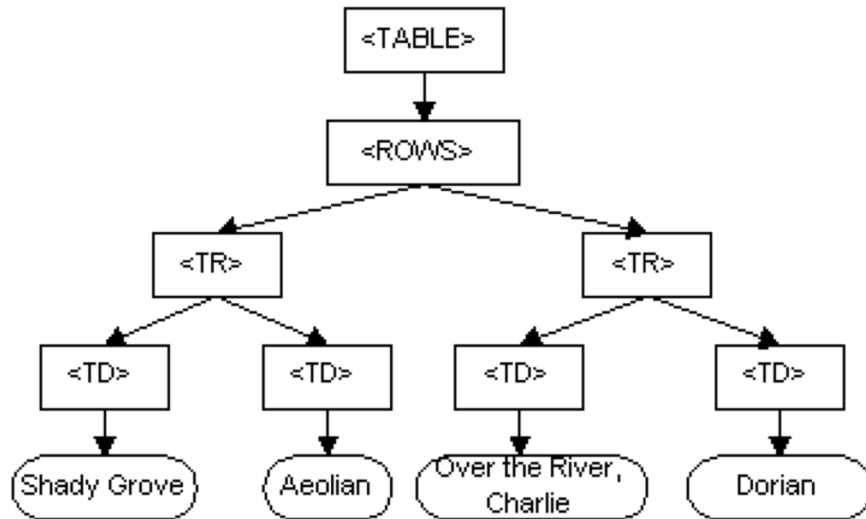
What is declarative programming?

What are React components?

What is the Virtual DOM?



What Is the DOM?



Programming API for Web Pages

Enables Dynamic Web Pages

Exposes Methods and Properties



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

```
<script>
```

```
    document.getElementById("userName").innerHTML = "Chris";
```

```
</script>
```

Pure JavaScript DOM Manipulation



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

```
<script>
```

```
    $( '#userName' ).html( "Chris" );
```

```
</script>
```

jQuery DOM Manipulation



```
<div ng-app="myApp" ng-controller="myCtrl">

  <p ng-bind-html="myText"></p>

</div>

<script>

  app.controller('MainCtrl', function($scope) {

    $scope.content = "Chris";

  });

</script>
```

Angular DOM Manipulation



Imperative vs. Declarative



Imperative

Start at the dot.

Walk to the lake.

Turn right and continue for 5 km.

Turn left and walk 2 km.



Declarative

Go to the 'x'.



```
<div id="app"></div>
<script>
  class UserInfo extends React.Component({
    render() {
      return (
        <p>{this.props.userName}</p>
      );
    }
  });

  ReactDOM.render(<UserInfo userName="Chris" />, document.getElementById('app'));
</script>
```

React DOM Manipulation



React Elements

```
const element = React.createElement("tr", null,  
    React.createElement("td", null, "data"),  
    React.createElement("td", null, "data")  
);  
  
ReactDOM.render(element, document.getElementById('app'));
```



React.Component

```
class MyComponent extends React.Component {  
  render() {  
    return ();  
  }  
}
```



React.Component

```
class TableRow extends React.Component {  
  render() {  
    return React.createElement("tr", null,  
      React.createElement("td", null, "data"),  
      React.createElement("td", null, "data")  
    );  
  }  
}
```



```
ReactDOM.render(<TableRow>, document.getElementById( ' app' ));
```

Output

```
<tr>
```

```
  <td>data</td>
```

```
  <td>data</td>
```

```
</tr>
```



```
class TableRow extends React.Component {  
  render() {  
    return (<tr><td></td><td></td></tr>);  
  }  
}  
ReactDOM.render(<TableRow>, document.getElementById( 'app' ));
```

Output: <tr><td></td><td></td></tr>



```
class MyTable extends React.Component {  
  render() {  
    return (<table><TableRow><TableRow><TableRow></table>);  
  }  
}  
  
ReactDOM.render(<MyTable>, document.getElementById('app'));
```

<table>

<tr><td></td><td></td></tr>

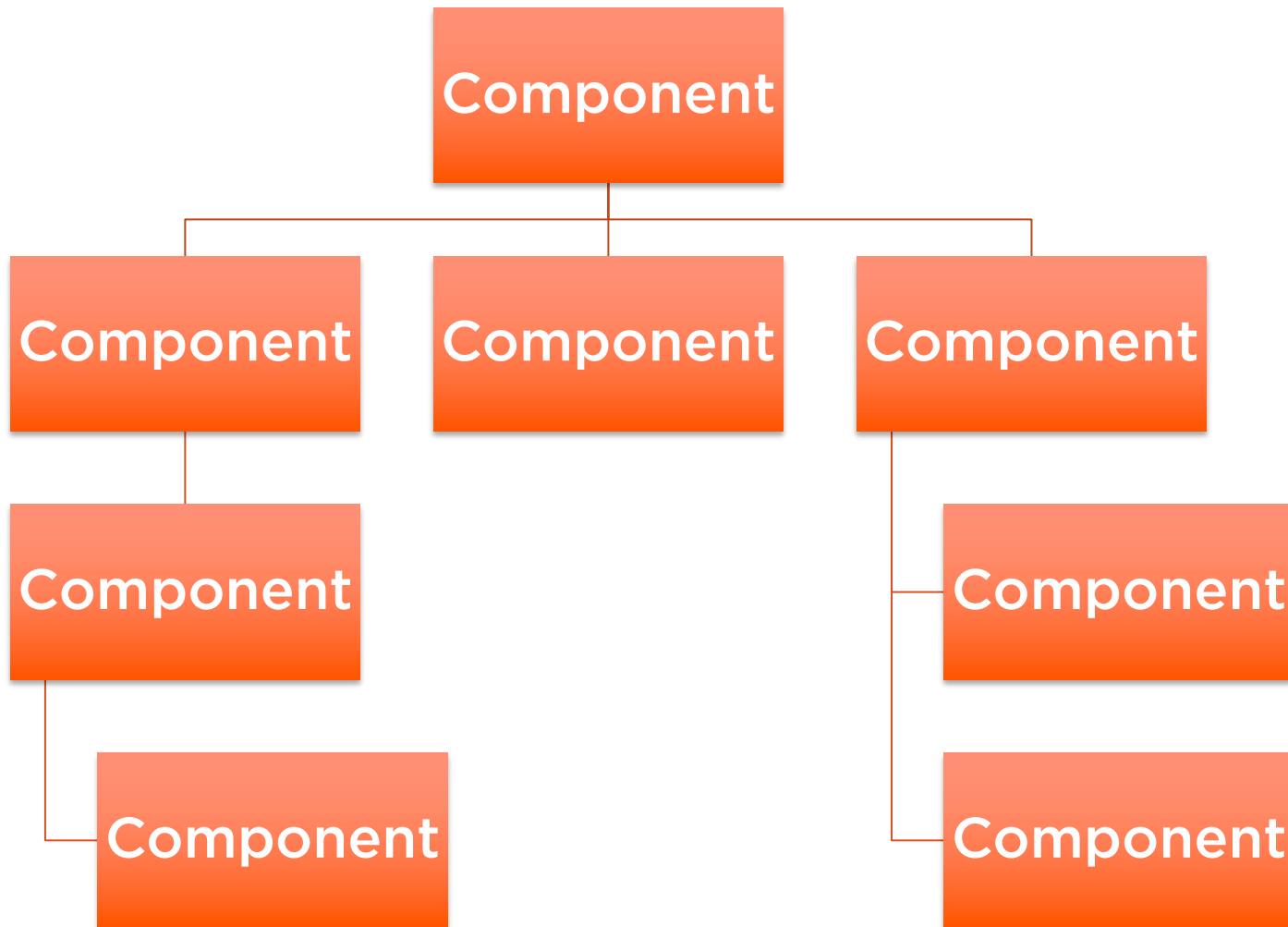
<tr><td></td><td></td></tr>

<tr><td></td><td></td></tr>

</table>



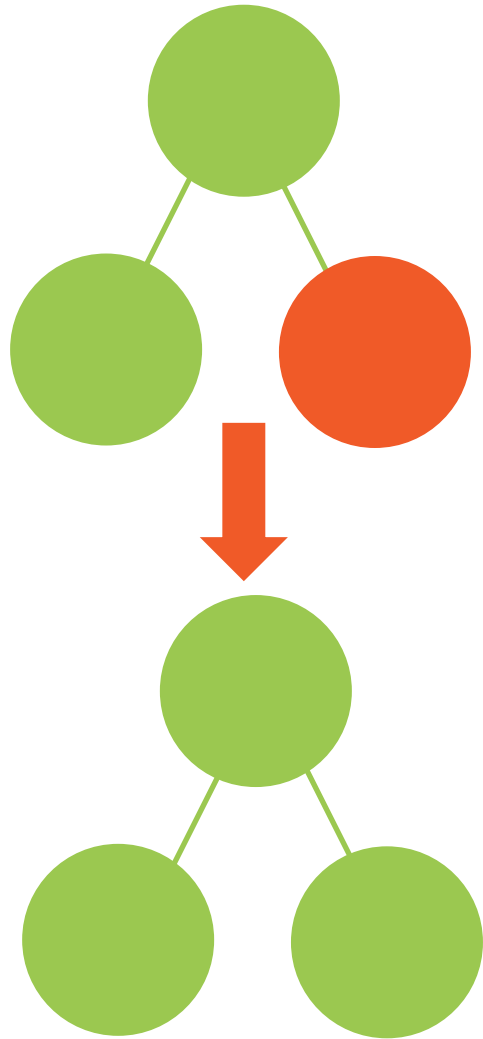
React Data Flow



Data



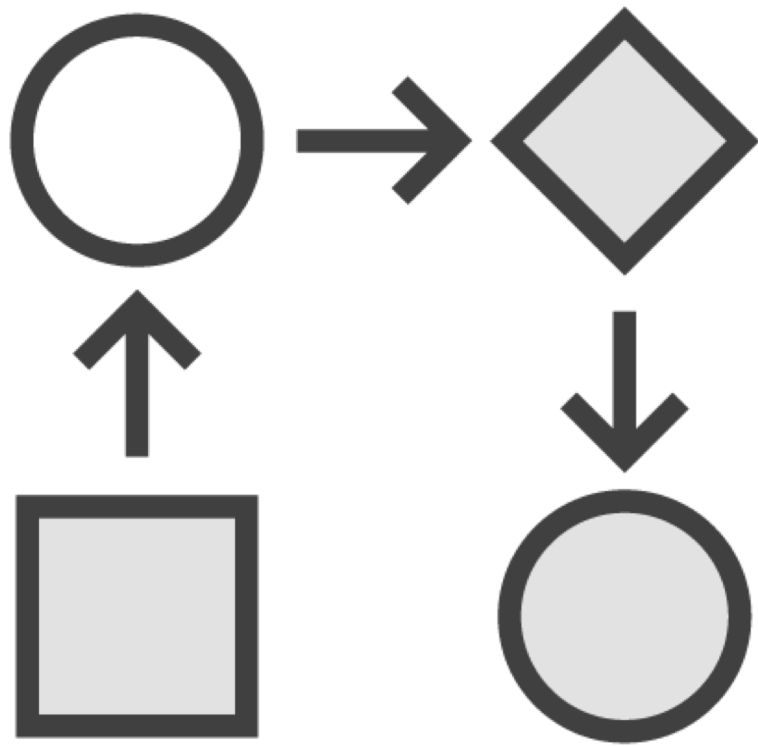
Virtual DOM



An Abstraction of the Browser DOM

React Holds Two Virtual DOMs

React Optimizes DOM Updates



React Development Process

Create a Component Hierarchy

Build a Static Version

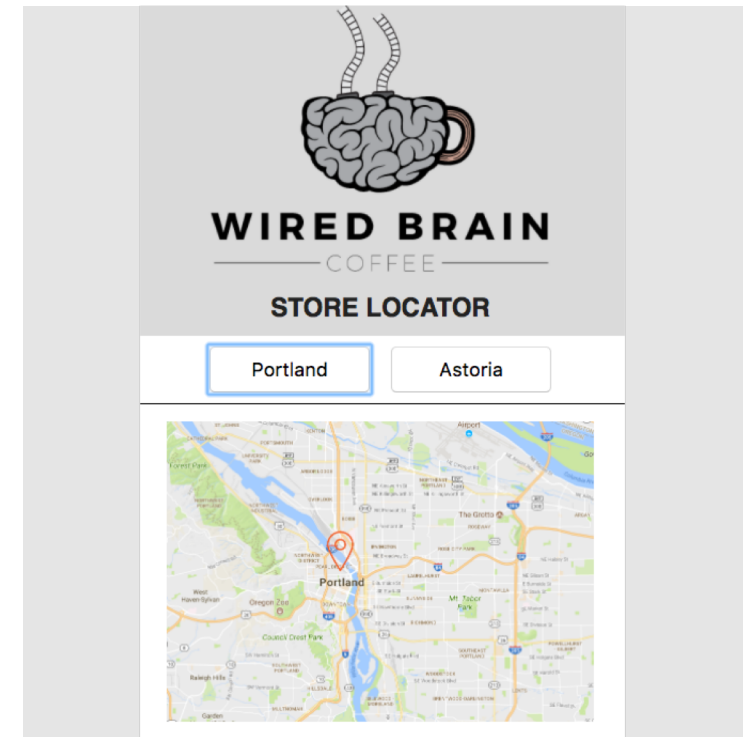
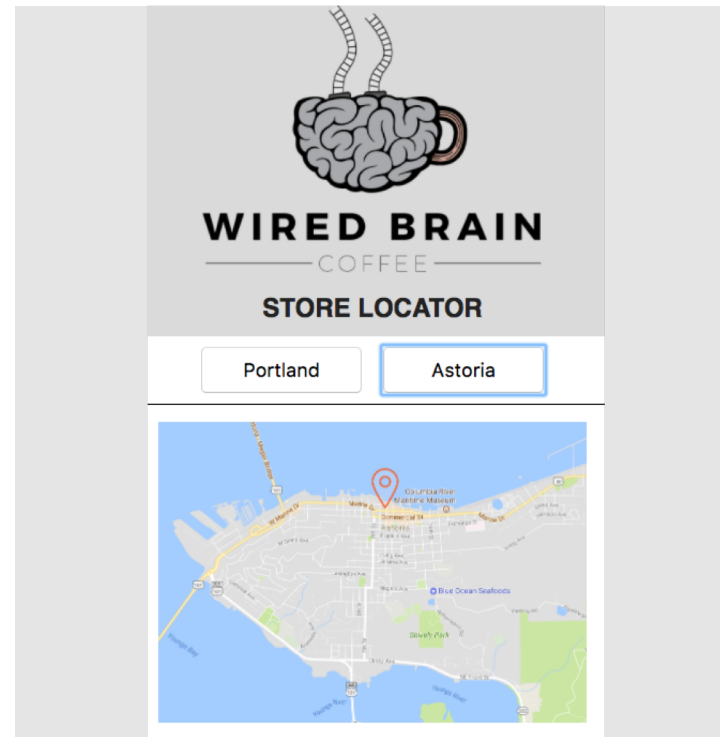
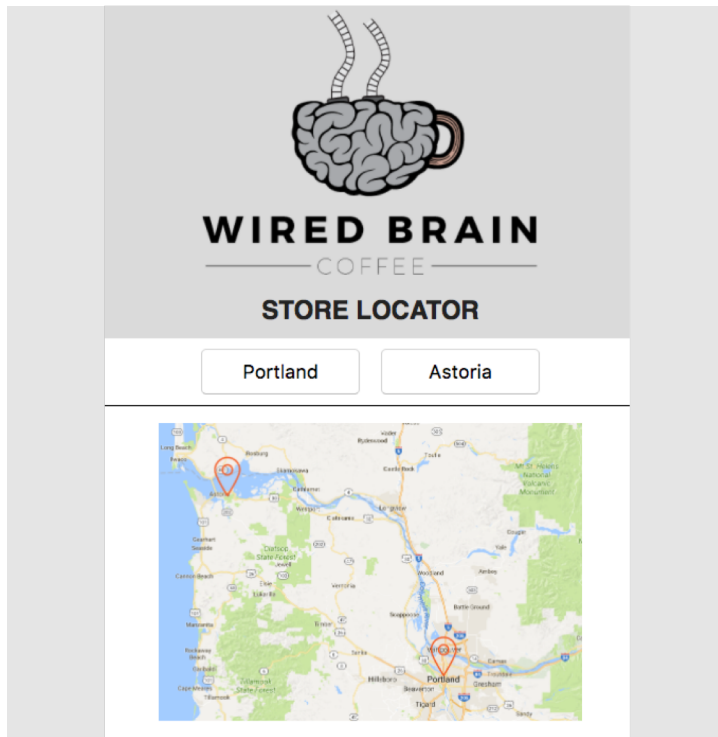
Identify the Minimal UI State

Identify Where State Should Live

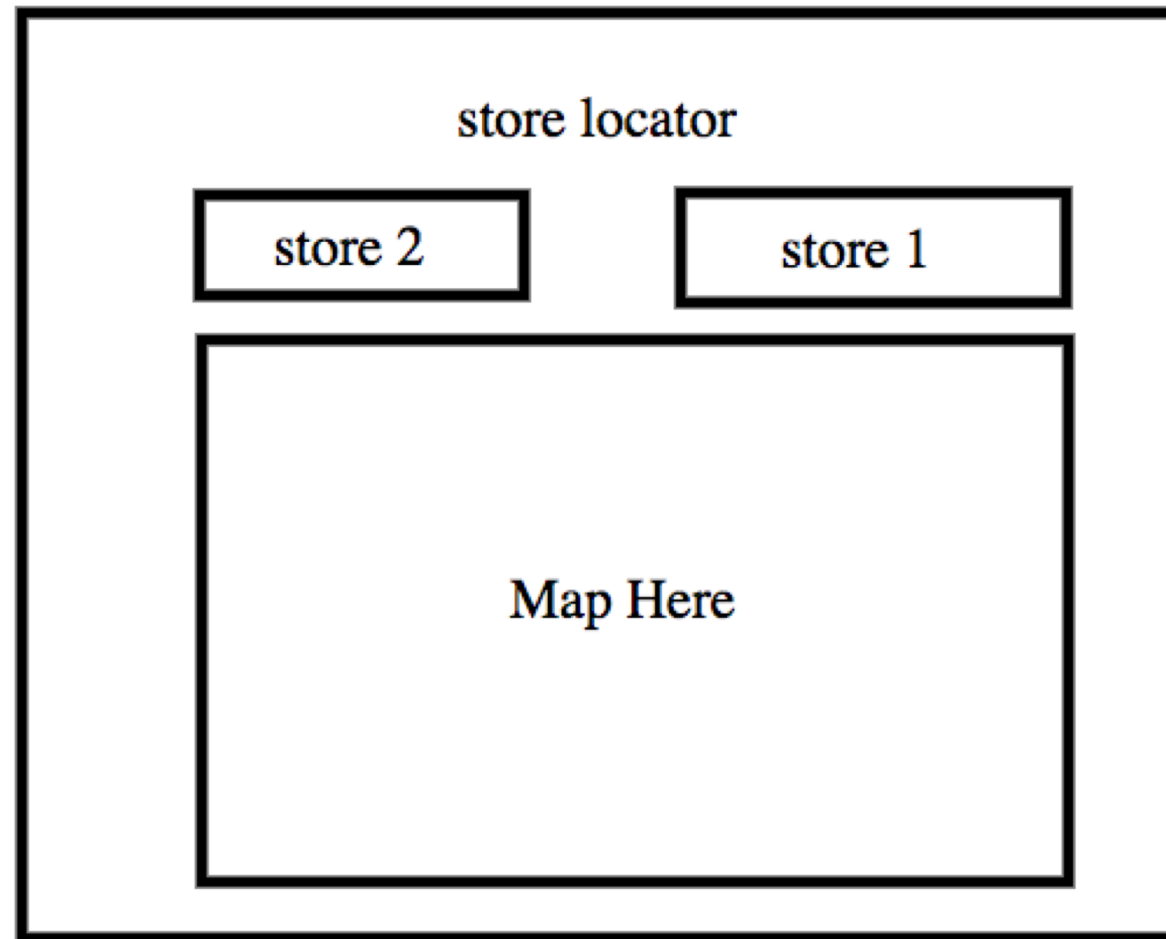
Add Inverse Data Flow



Prototype v2.0



Break Up the UI



Demo



Building It up with Components



Demo



Passing Data Between Components



Summary



Components return elements

Elements describe output

ReactDOM renders elements

Virtual DOM updates the browser DOM



Refactoring with React

COMING UP NEXT



Chris Minnick

CEO, WATZTHIS

@chrisminnick www.watzthis.com

