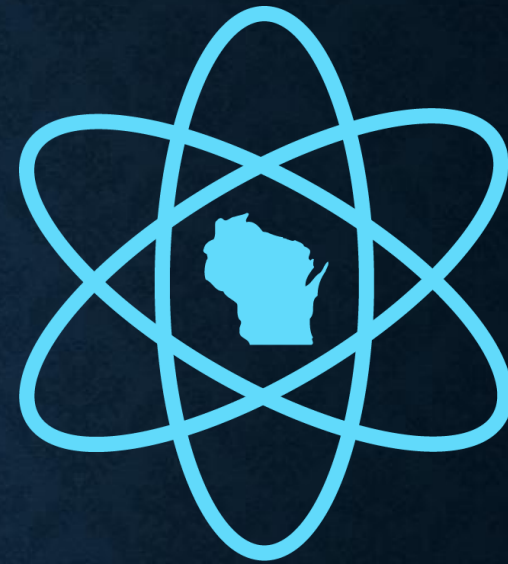
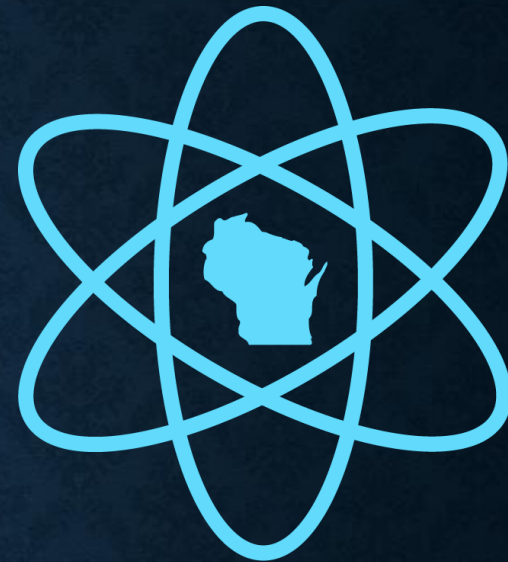


RYAN WACHHOLZ



- Worked at NM for 8 years as a web developer
- Have predominantly worked with Backbone
- Have only worked with ReactJS for 3 months (so I welcome all questions but may not have all the answer)

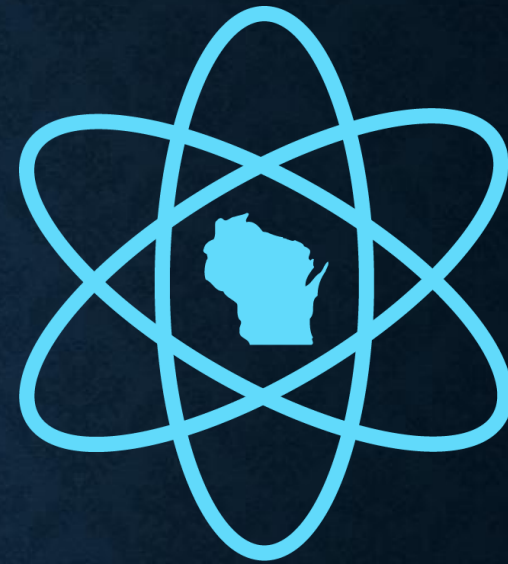




WHAT IS REACTJS?

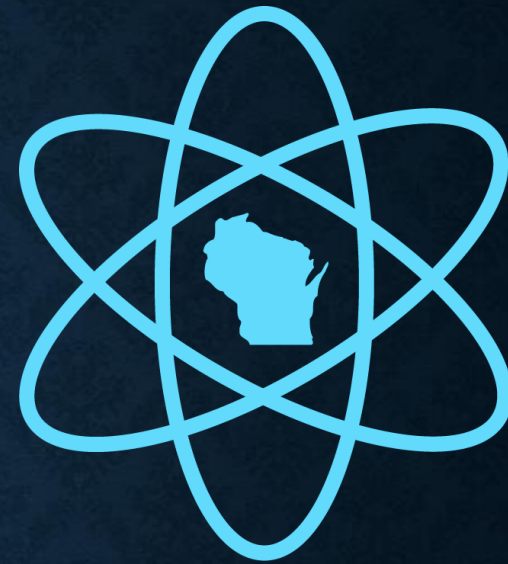
Why the heck is it so cool?

AGENDA



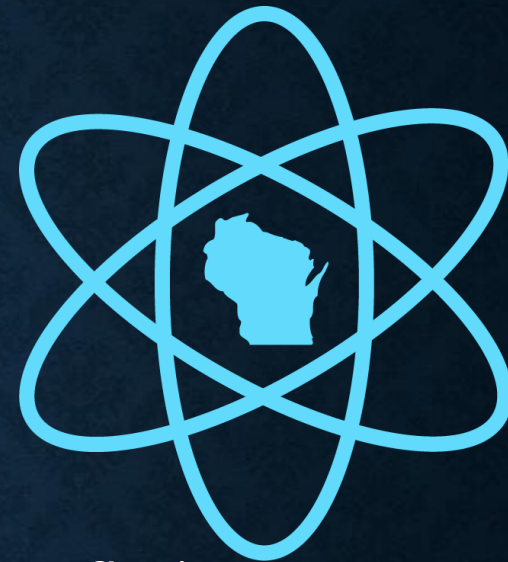
- [Documentation](#) - About ReactJS
- JSX
- React components, Props, Specs, Lifecycle, and State
- The Virtual DOM
- Live coding...dun dun dunnn

REACTJS



- Facebook
- It is declarative view layer
- Abstracting out the DOM
- Re-rendering on every state change
- Well defined component lifecycle
- Encourages unidirectional data flow
- Server side and client side rendering
- Virtual DOM

JSX

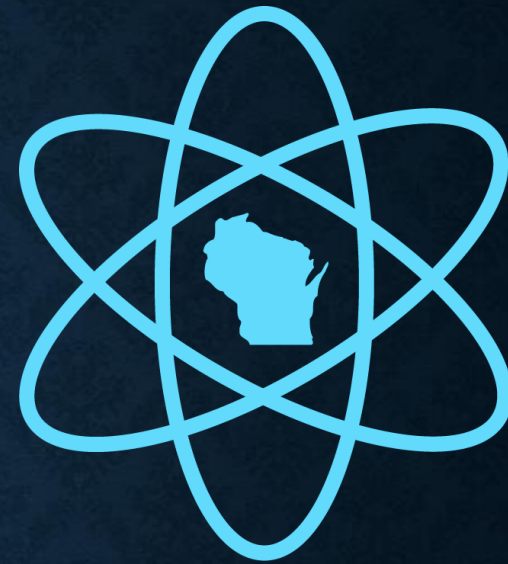


- JavaScript XML syntax transform
- Other frameworks put JavaScript in HTML...here put the HTML in JavaScript
- View and template are the same which are represented in JSX
- It all compiles into JavaScript!

```
// Vanilla JS
render: function(){
  return (
    React.DOM.div({className: 'blue-boarder'},
      React.DOM.a({href: 'http://reactjswisconsin.com/'},
        'ReactJS Milwaukee'),
    );
};
```

```
// JSX
render: function(){
  return (
    <div className='blue-boarder'>
      <a href='http://reactjswisconsin.com/'>
        ReactJS Milwaukee
      </a>
    </div>
  );
};
```

COMPONENTS

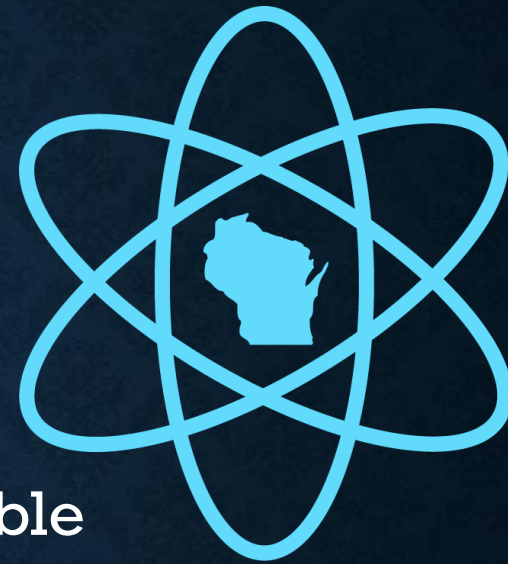


- Components are stateful...which is cool!
- Re-renders on its own...this must be SLOW! ...it's not!

```
var HelloComponent = React.createClass({
  render: function(){
    return (
      <h1>Hello, {this.props.groupName}!</h1>
    );
  }
});

ReactDOM.render(
  <HelloComponent groupName={this.state.groupName}/>,
  document.getElementById('myDiv')
);
```


PROPS

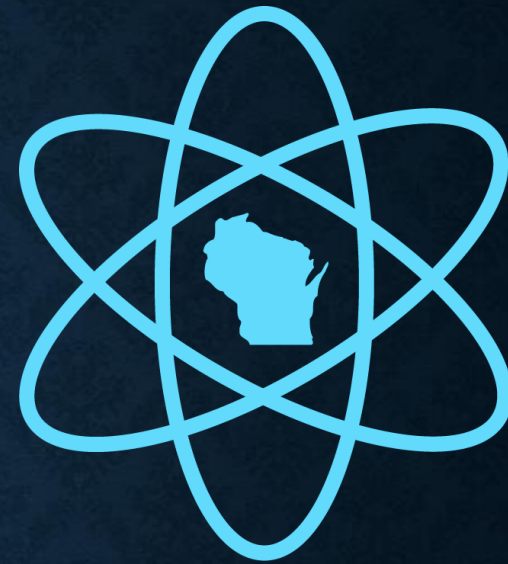


- Define components
- A component should never change its props, they're immutable

```
var MessageComponent = React.createClass({
  render: function(){
    return (
      <h1>{this.props.greeting}, {this.props.group}!</h1>
    );
  }
});

ReactDOM.render(
  <MessageComponent greeting={'Hello'} group={"ReactJS Wisconsin"} />,
  document.getElementById('myDiv')
);
```

SPECS AND LIFECYCLE



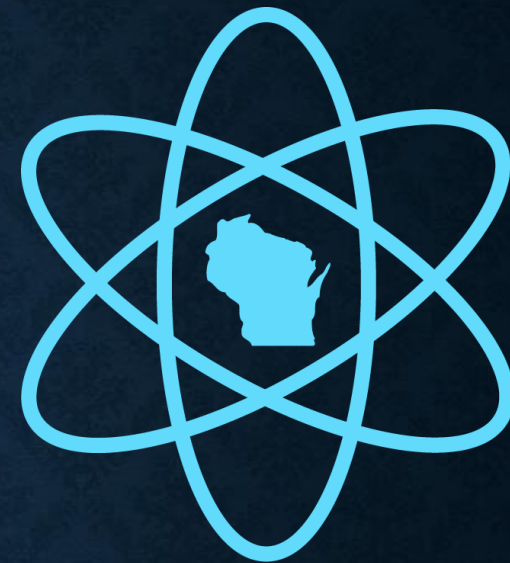
- Specs
 - `getInitialState` – Return value is the initial value for state.
 - `getDefaultProps` – Sets fallback props values if props aren't supplied.
 - ...
- Lifecycle Methods
 - `componentWillMount` – Invoked once, on both client & server before rendering occurs.
 - `componentDidMount` – Invoked once, only on the client, after rendering occurs.
 - `shouldComponentUpdate` – Return value determines whether component should update.
 - ...
- There are more [here](#)

STATE



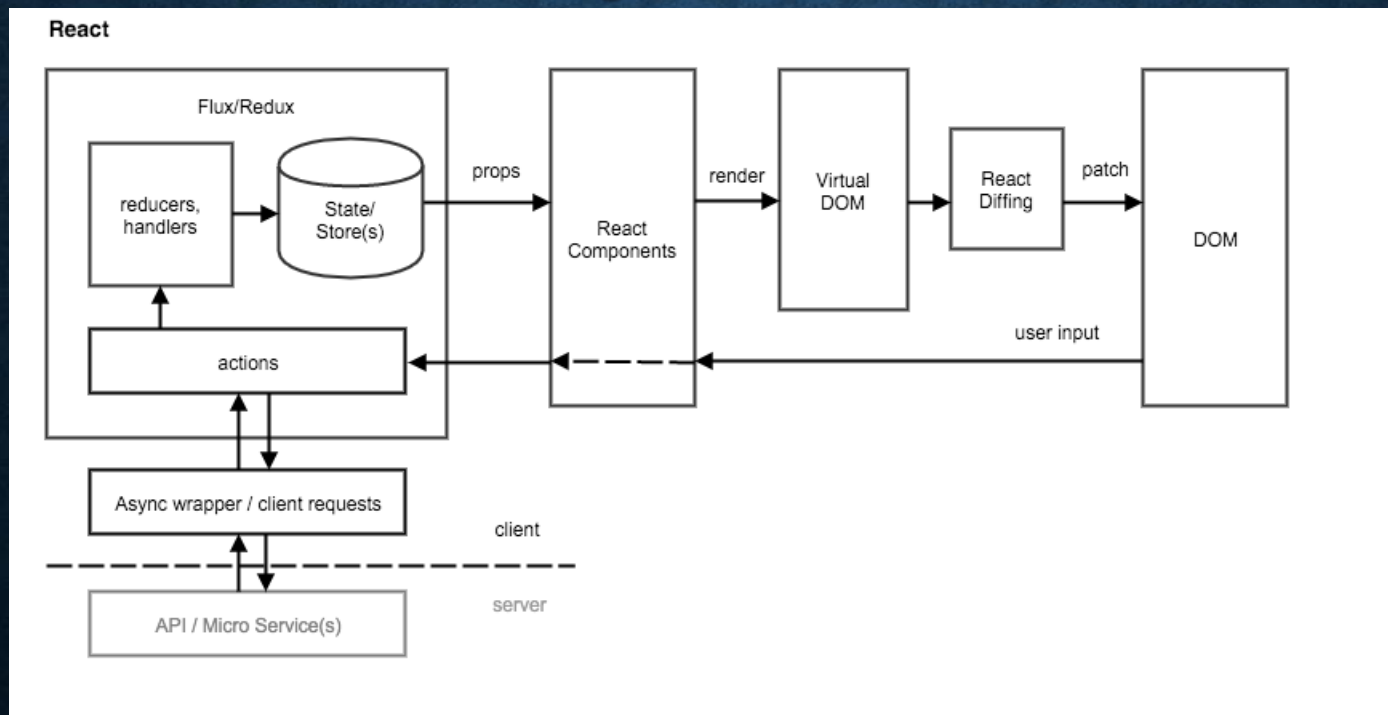
- Single source of truth
- `setState()` - auto triggers UI updates leveraged by the virtual DOM!

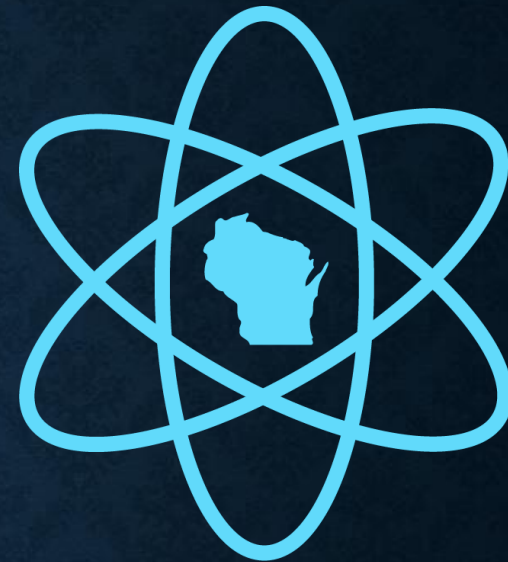
```
var MessageComponent = React.createClass({
  getInitialState: function() {
    return {
      group: 'ReactJS Wisconsin'
    }
  },
  render: function() {
    return (
      <h1>Hi, {this.state.group}!</h1>
    );
  }
});
```



VIRTUAL DOM

- A JavaScript tree of React elements and components
- No manual DOM manipulation needed
- No total refresh when data changes





LETS SEE SOME LIVE EXAMPLES