

# Hello!

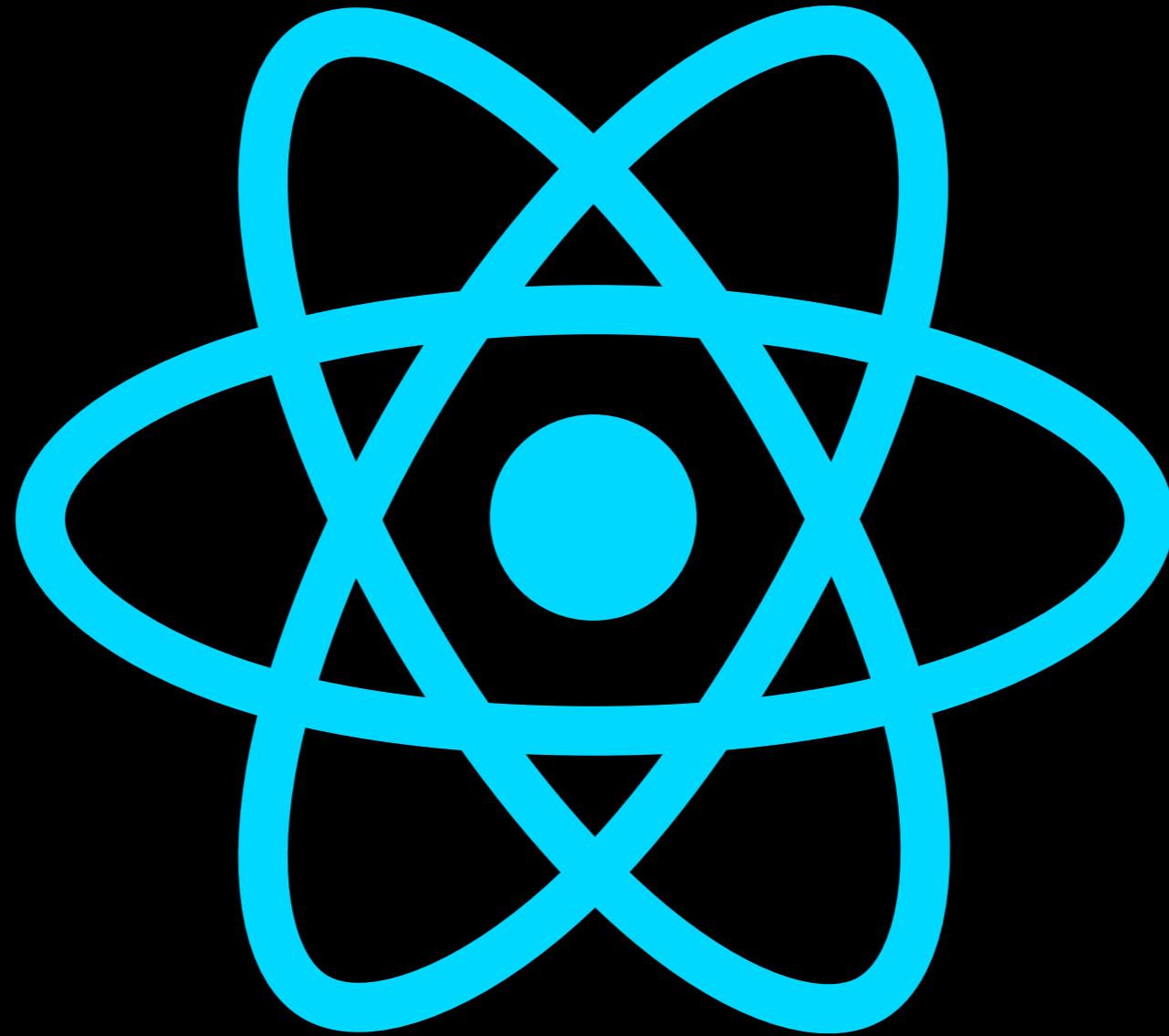
*Mark Hintz*

*@MarkHintz*

*Interactive Things*

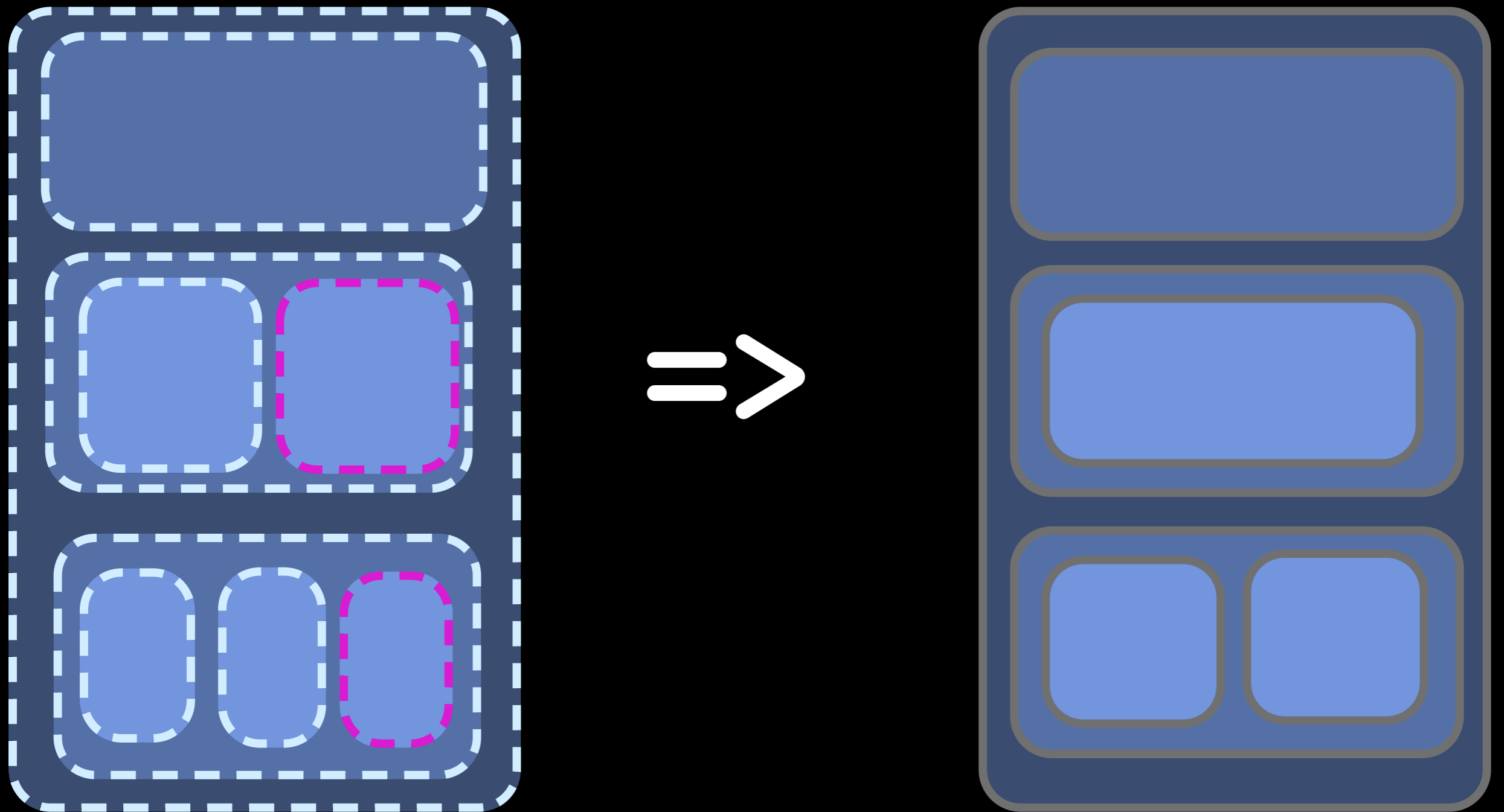
# React

## The Ideas Behind the Hype

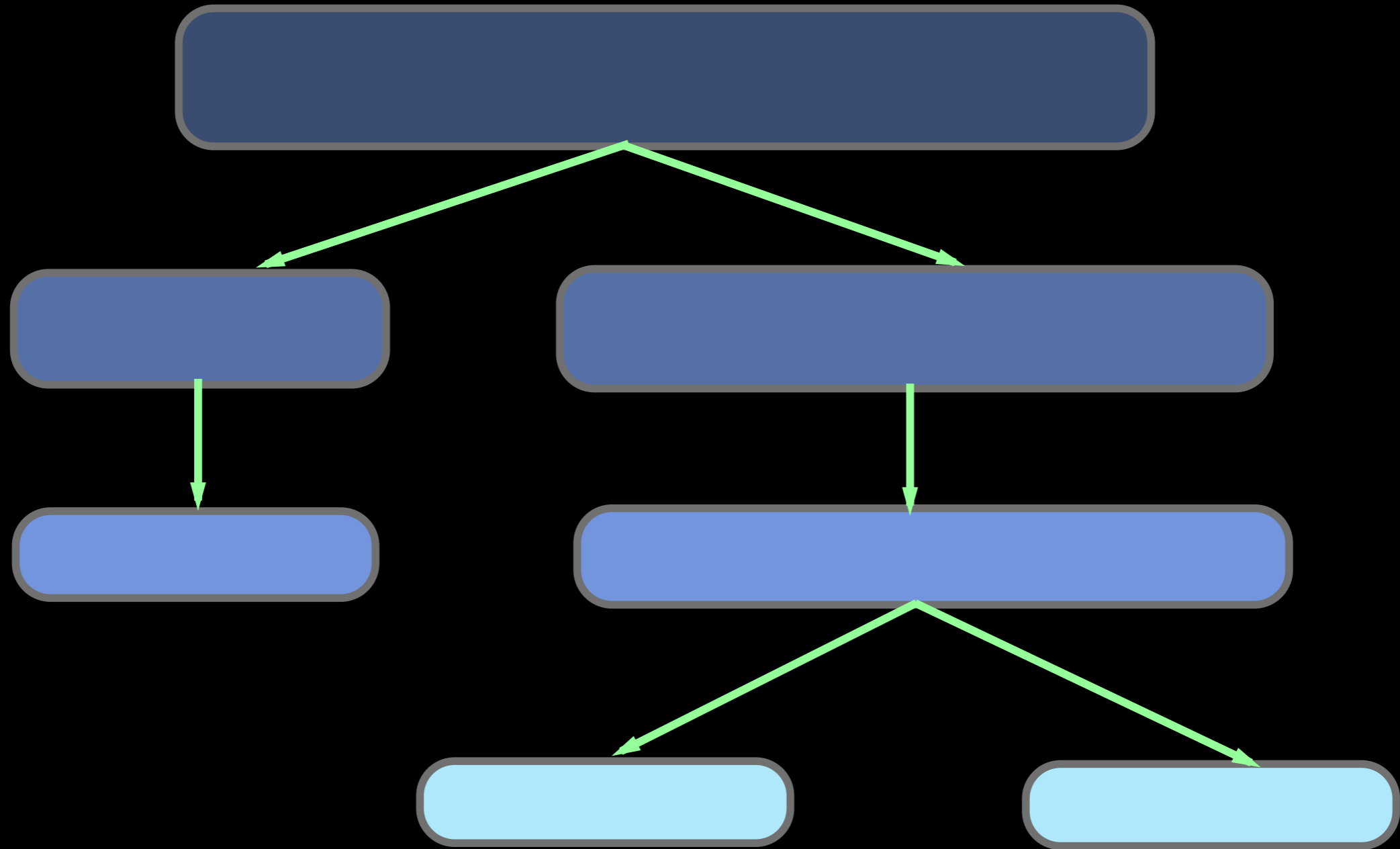


```
React.render(<App appState={Store.getState()} />, document.getElementById('app'));
```

# Rendering with Virtual DOM



# The Virtual Tree



# We have to talk about JSX

```
render: function() {
  return (
    <ContentBox className="ContentBox">
      <h1 className="ContentHeader">Header Content</h1>
      <h2 className={"ContentSubhead" + (this.props.hasSubhead ? "visible" : "")}>
        {this.props.subheadContent}
      </h2>
      <Tooltip
        text={this.props.tooltipText}
        side={'top'}
        top={this.props.contentTop}
        left={this.props.contentLeft}
      />
      <div className="ContentMain">{this.props.mainContent}</div>
    </ContentBox>
  )
}
```

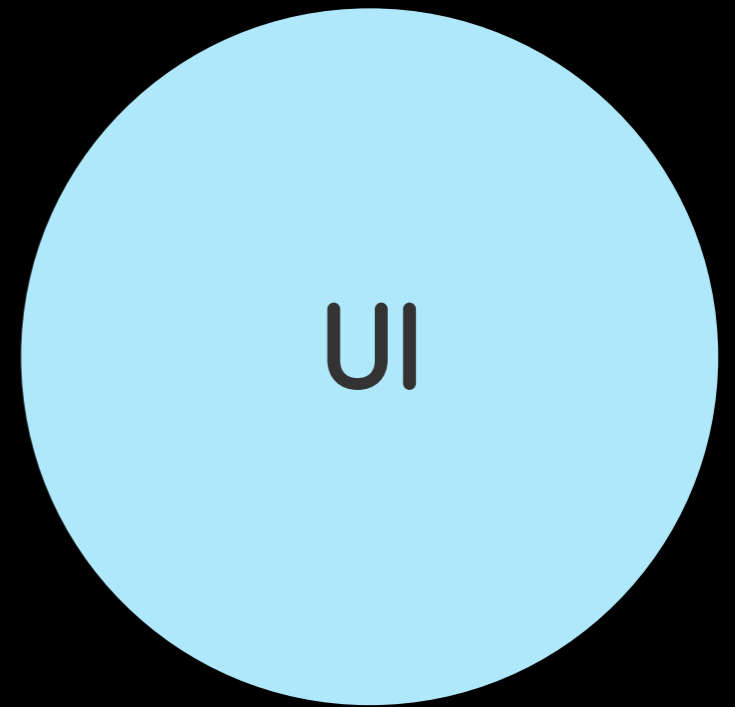
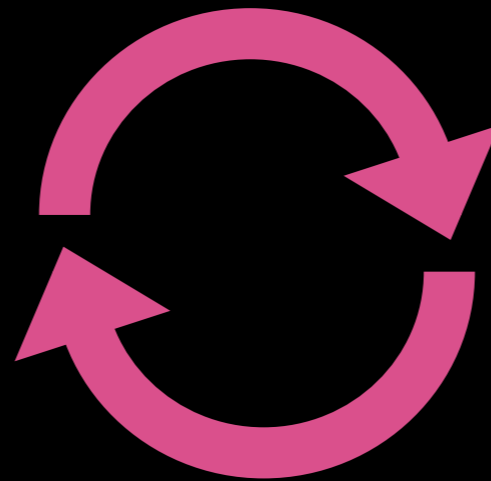
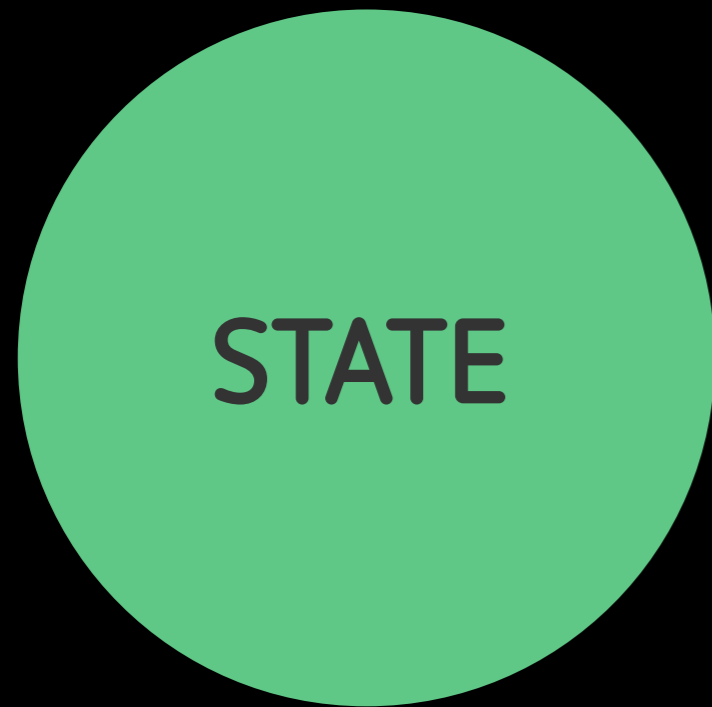
# But, JSX can be OK

- *Compilation step handled well by tooling*
- *The underlying API can change*
- *React separates concerns along different lines*
- *Rendering DOM is really the central concern of components*
- *The hard stuff happens elsewhere*

# Components and Properties

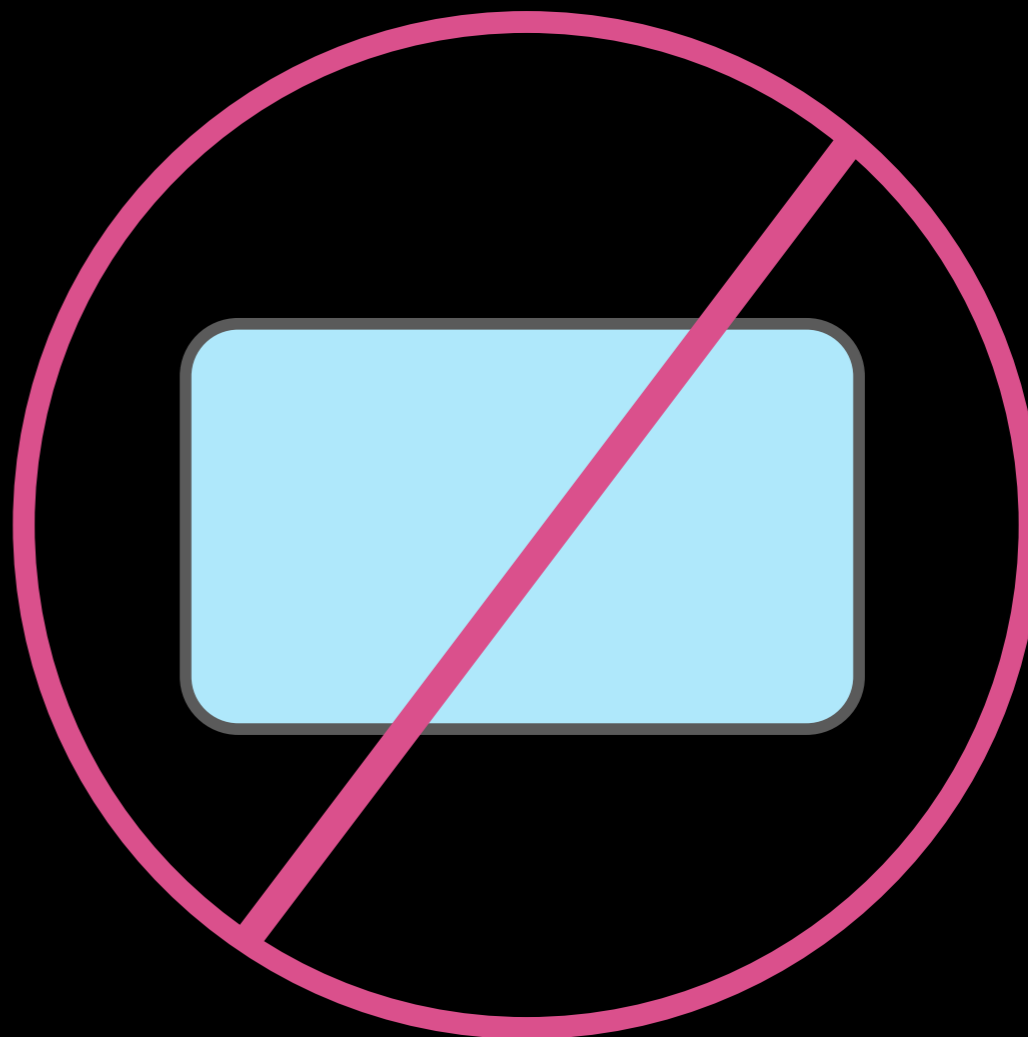
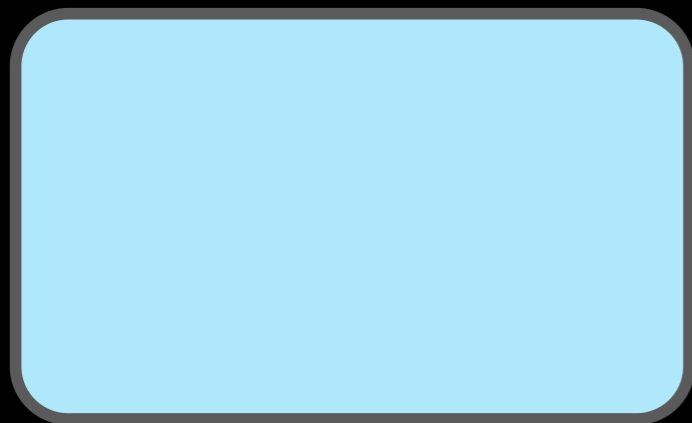
```
contents = render({ ...properties... })  
children = [child1({}), child2({}), child3({}), ...]
```



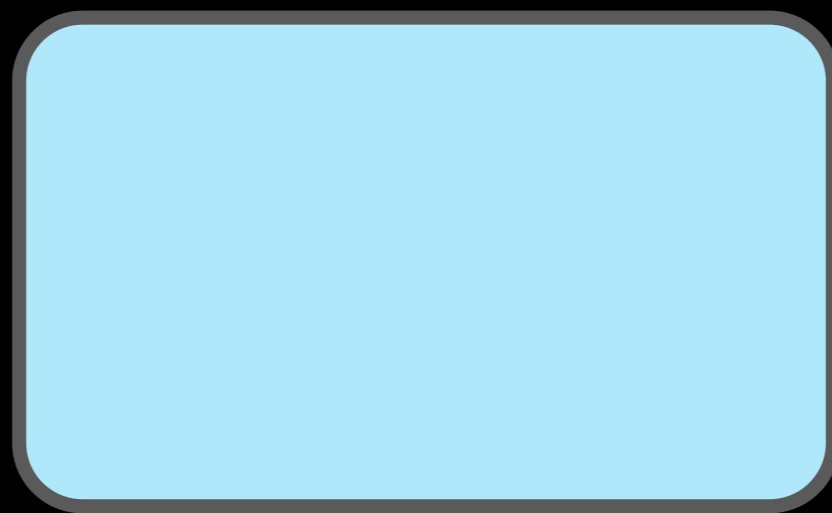


# The One Takeaway

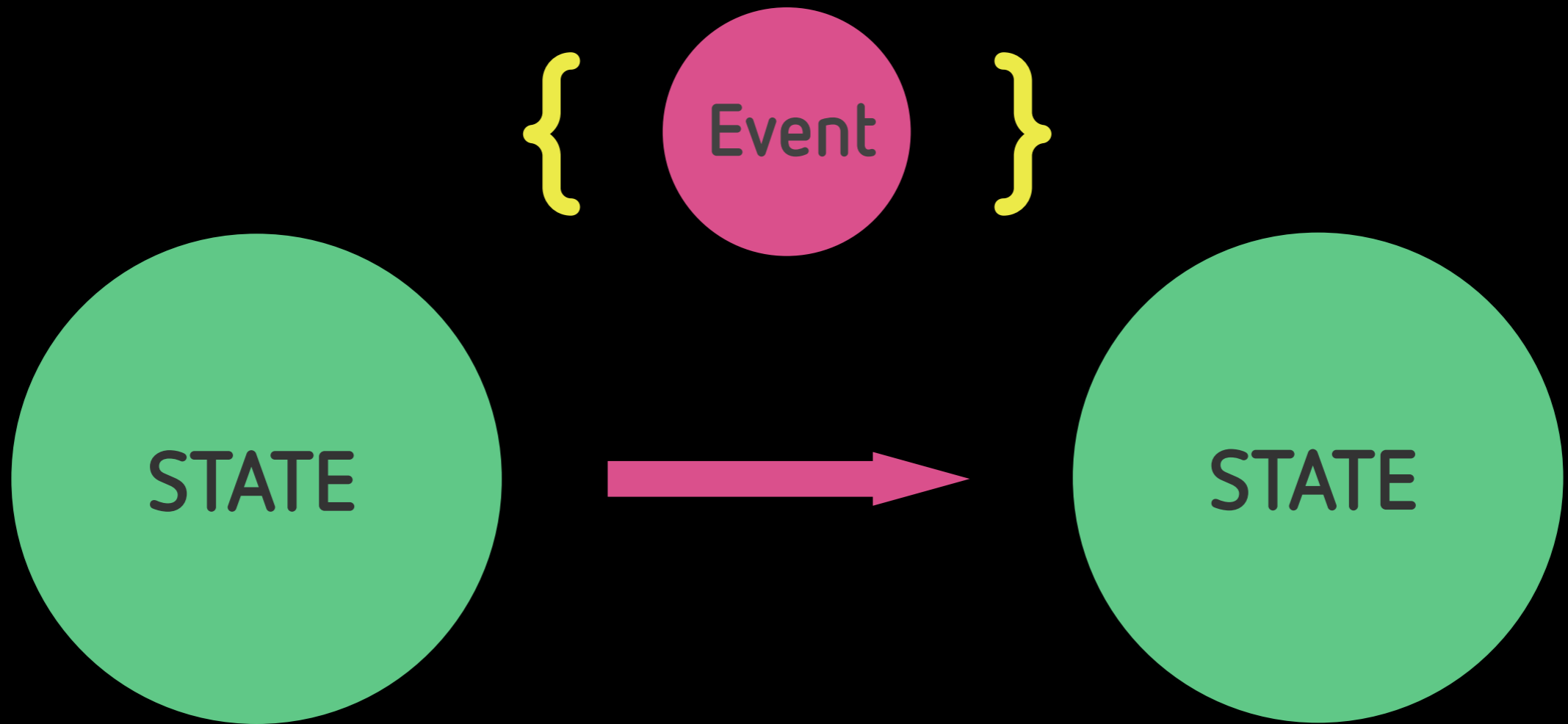
*when you remove as much as possible from your components any awareness of the application's broader state, and what caused them to look the way they look, things get a lot simpler*

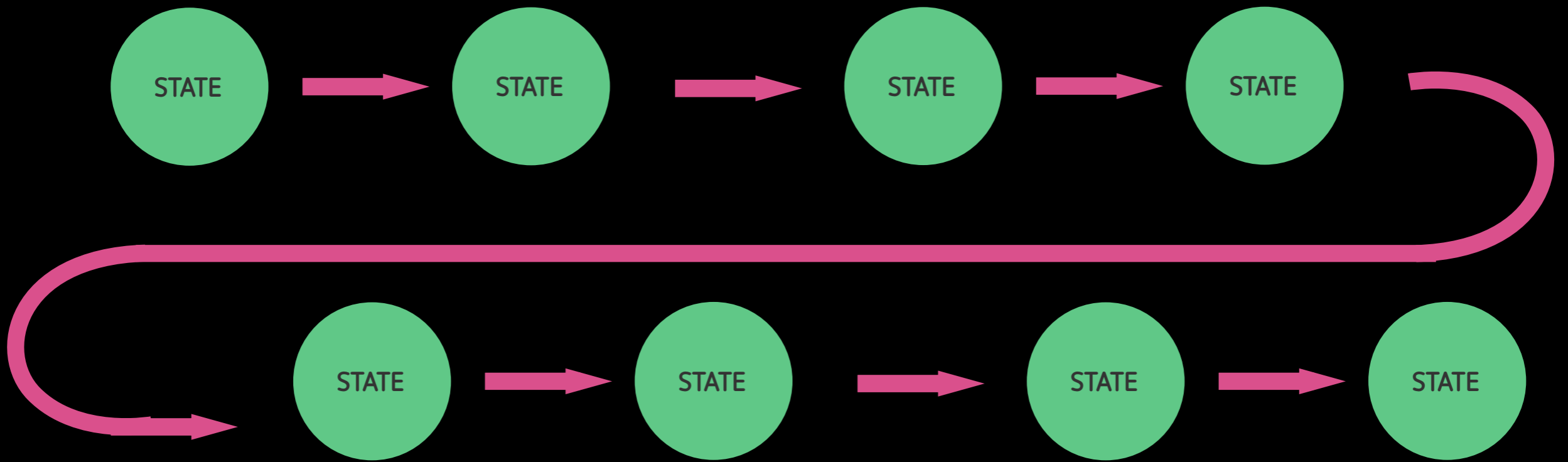


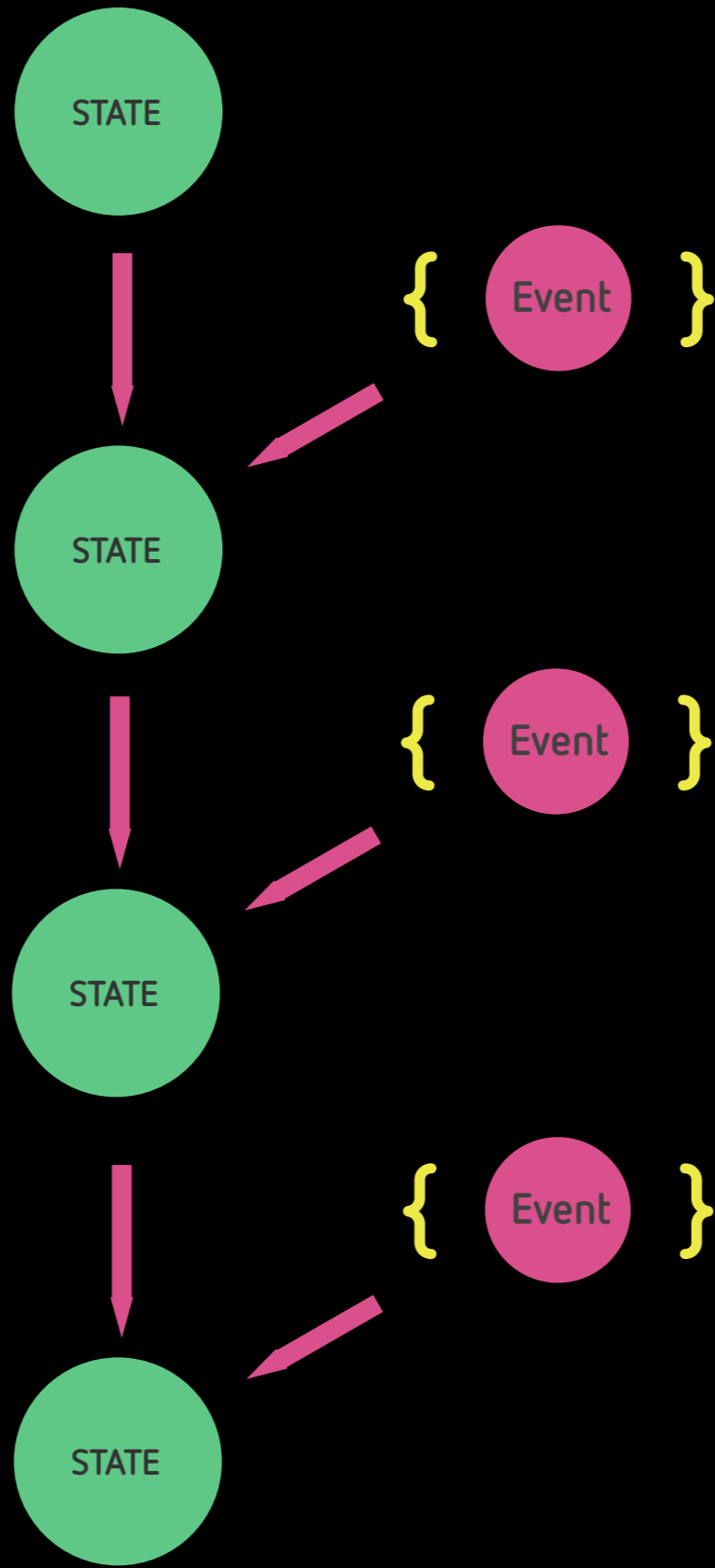
# "Dumb" Components













# That's Flux!

# That's Functional!

# Go Forth, and Explore!

# Thanks!

*Mark Hintz*

*tweet me with comments!*

*@MarkHintz*

*[interactivethings.com](http://interactivethings.com)*