REACT(ION)

RAJU GANDHI

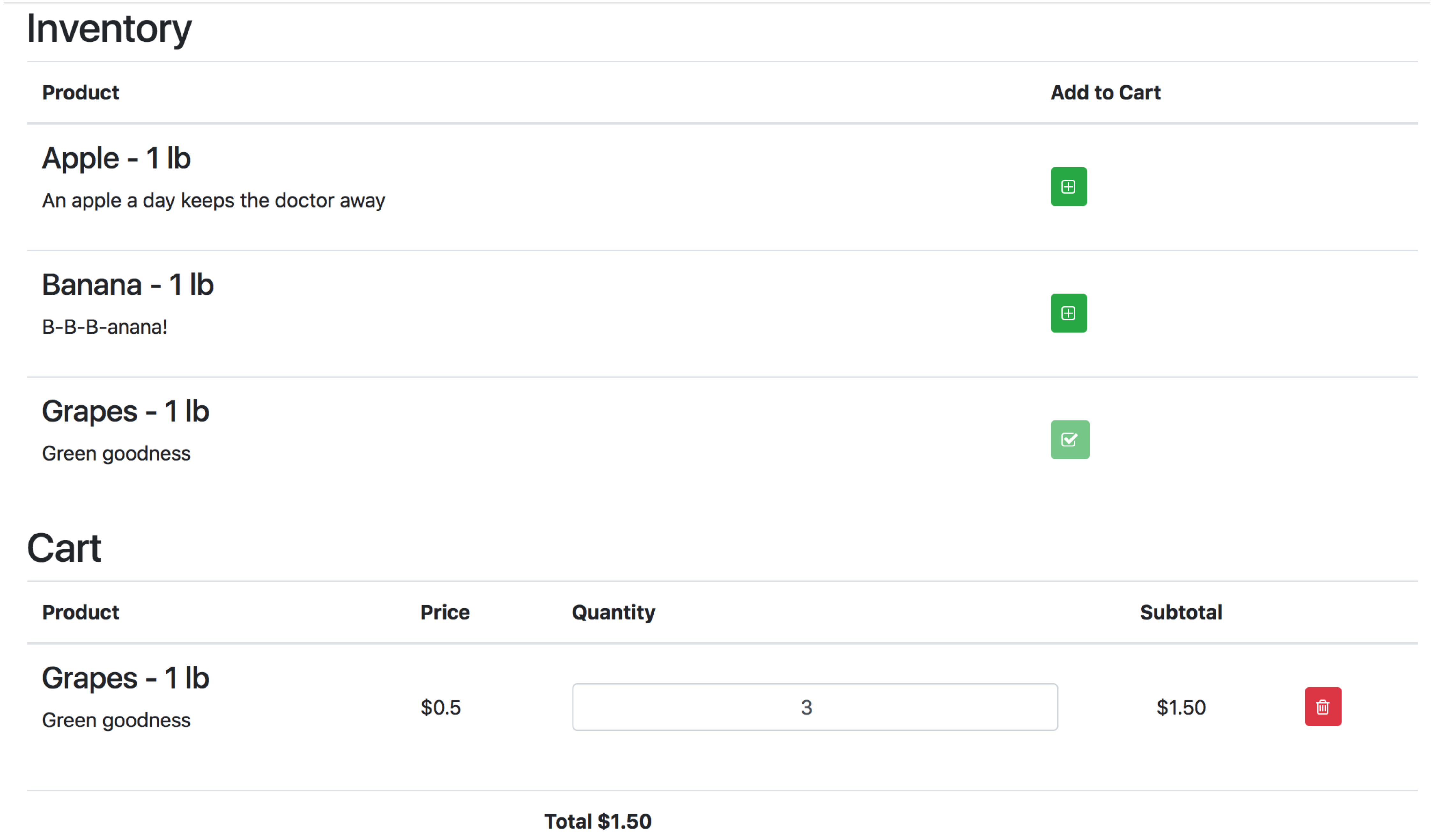
@LOOSELYTYPED

REACT.JS

REACT

- VIEW LIBRARY FROM FACEBOOK
- COMPONENT ORIENTED
- ESCHEWS FUNCTIONAL PROGRAMMING

COMPONENTS!



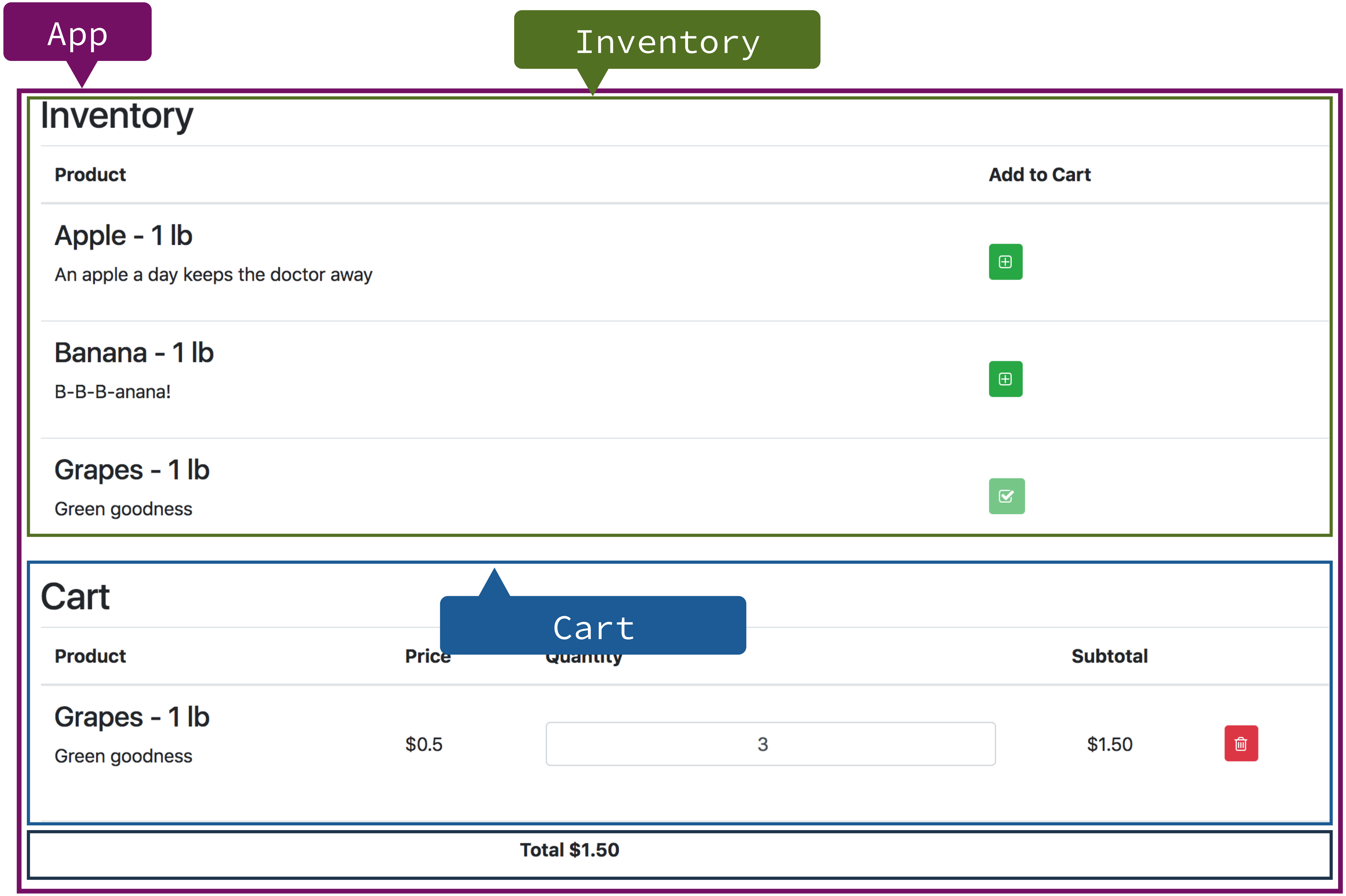
Inventory

Product	Add to Cart
Apple - 1 lb An apple a day keeps the doctor away	⊞
Banana - 1 lb B-B-B-anana!	⊞
Grapes - 1 lb Green goodness	☑

Cart

Product	Price	Quantity	Subtotal	
Grapes - 1 lb Green goodness	$0.5	3	$1.50	🗑

Total $1.50



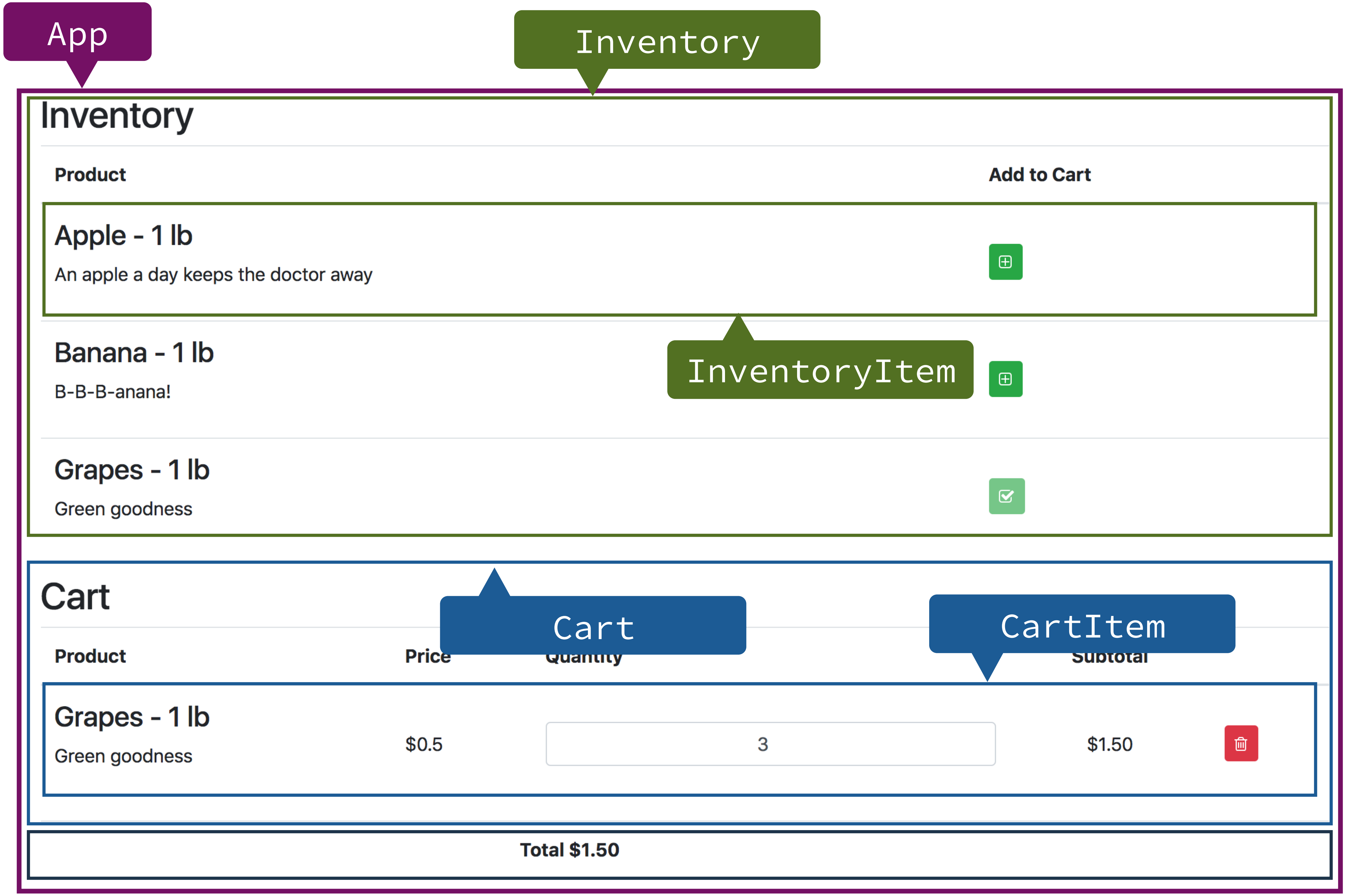
Inventory

Product	Add to Cart
Apple - 1 lb An apple a day keeps the doctor away	⊞
Banana - 1 lb B-B-B-anana!	⊞
Grapes - 1 lb Green goodness	☑

Cart

Product	Price	Quantity	Subtotal	
Grapes - 1 lb Green goodness	$0.5	3	$1.50	🗑

Total $1.50



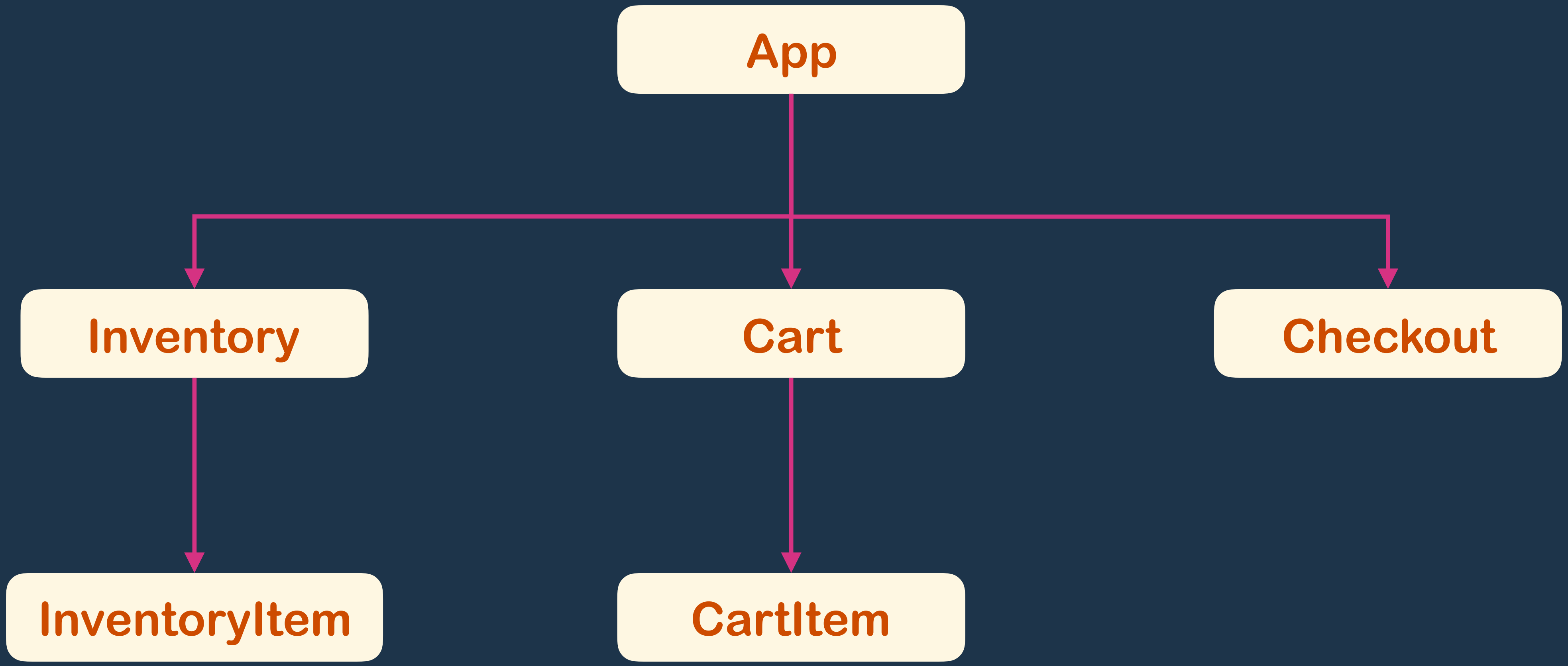
Inventory

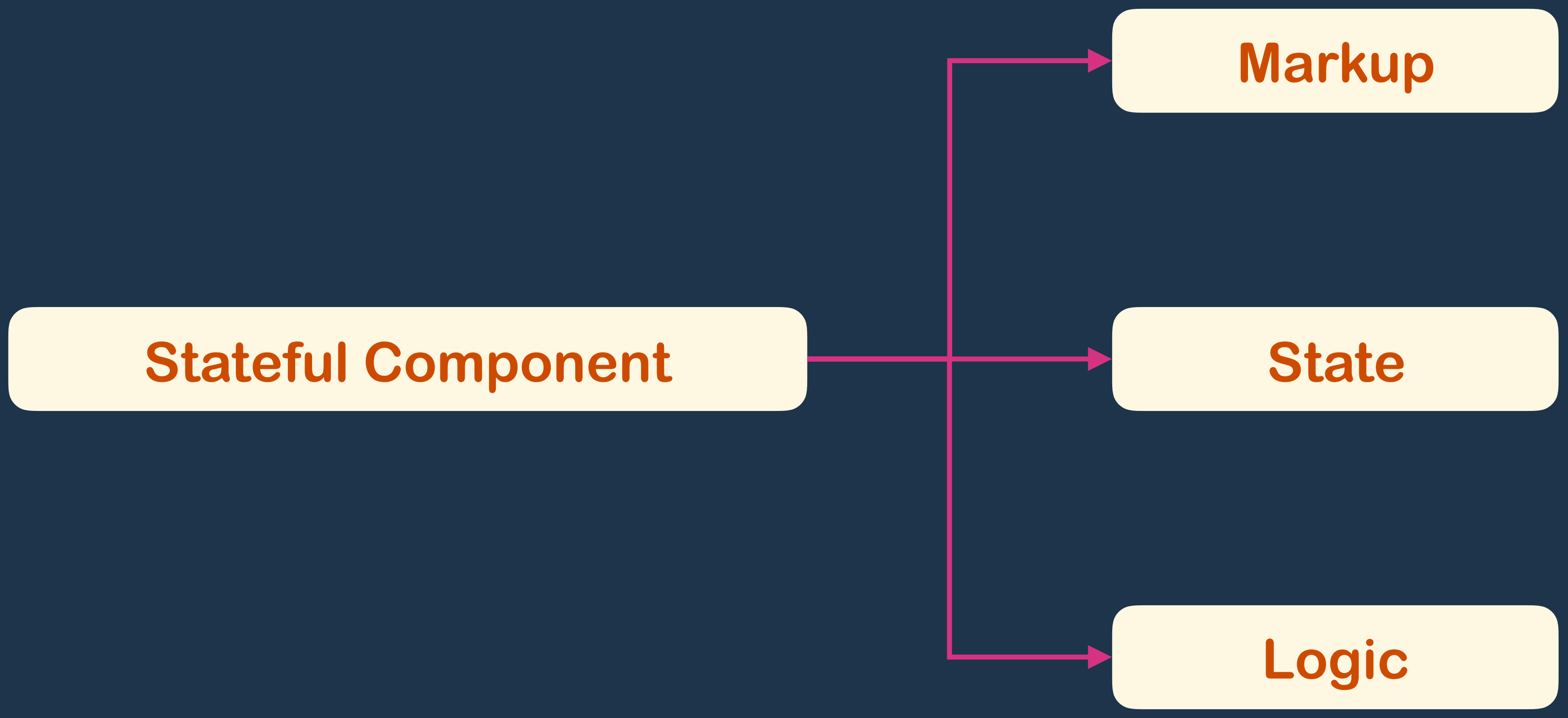
Product	Add to Cart
Apple - 1 lb An apple a day keeps the doctor away	⊞
Banana - 1 lb B-B-B-anana!	⊞
Grapes - 1 lb Green goodness	☑

Cart

Product	Price	Quantity	Subtotal	
Grapes - 1 lb Green goodness	$0.5	3	$1.50	🗑

Total $1.50


App
Inventory
Cart
Checkout
InventoryItem
CartItem

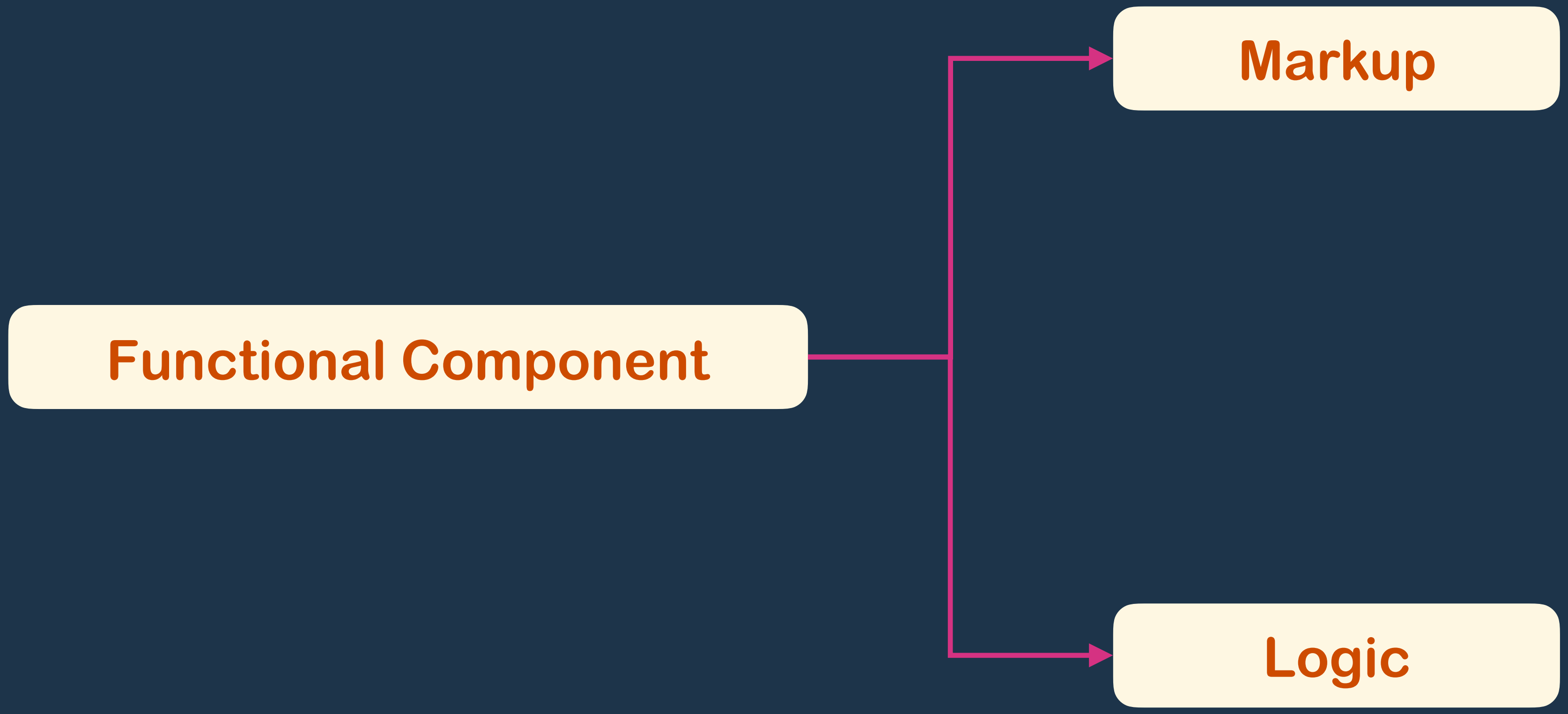


Stateful Component

Markup

State

Logic



Functional Component

Markup

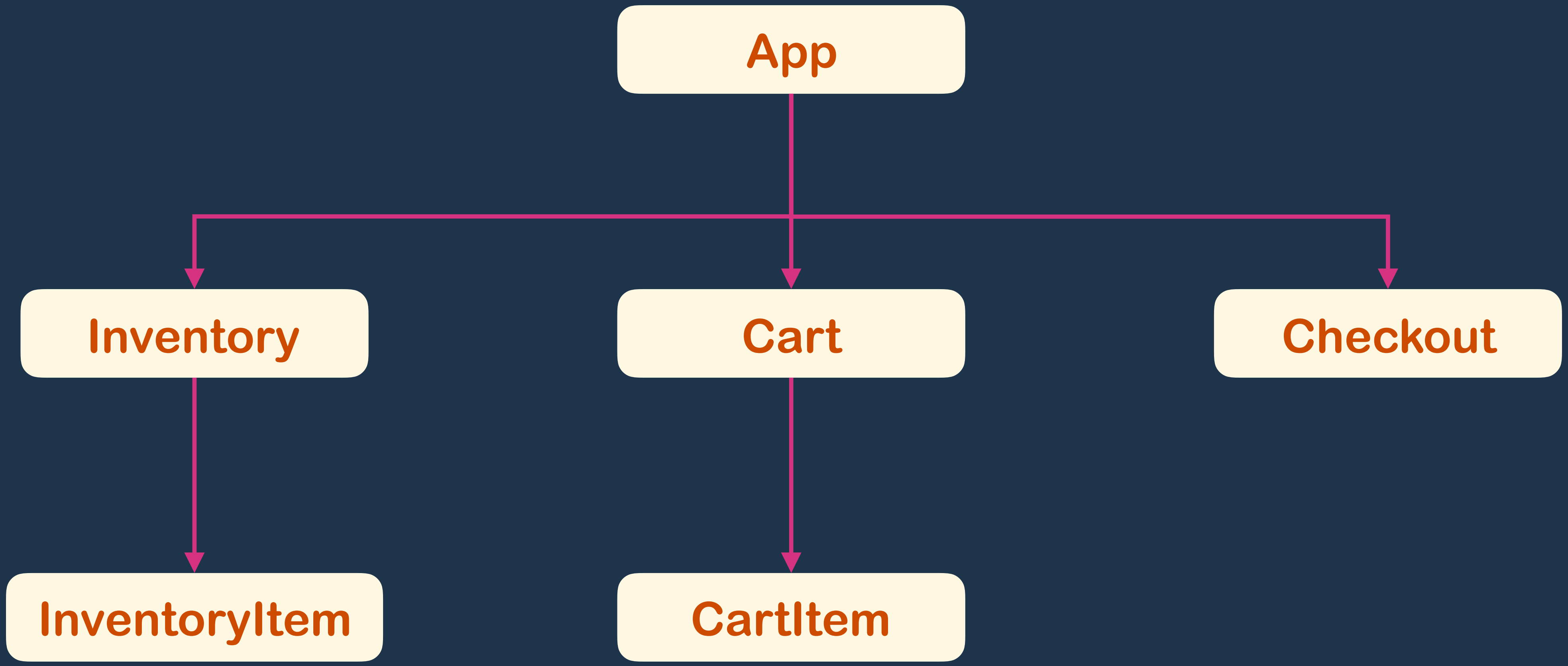
Logic

COMPONENTS

- RENDERS ITS OWN UI
- GETS INPUTS AS PROPS
- SENDS OUTPUTS AS EVENTS
- CAN MAINTAIN INTERNAL STATE
- CAN HAVE A LIFECYCLE MANAGED BY REACT

VIEW = f(APP STATE)

UNIDIRECTIONAL



App

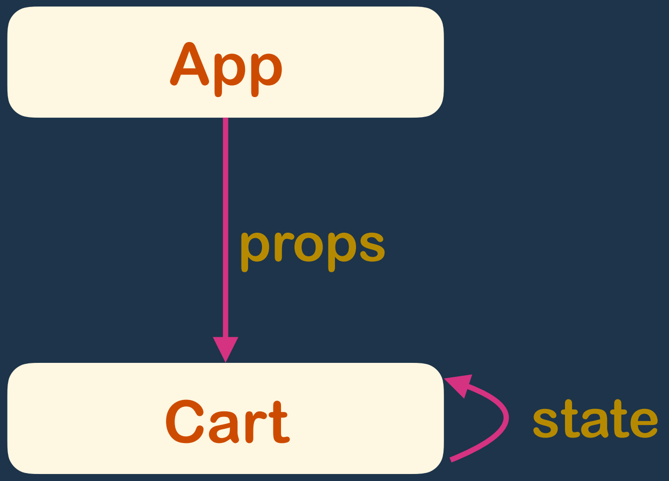
Inventory

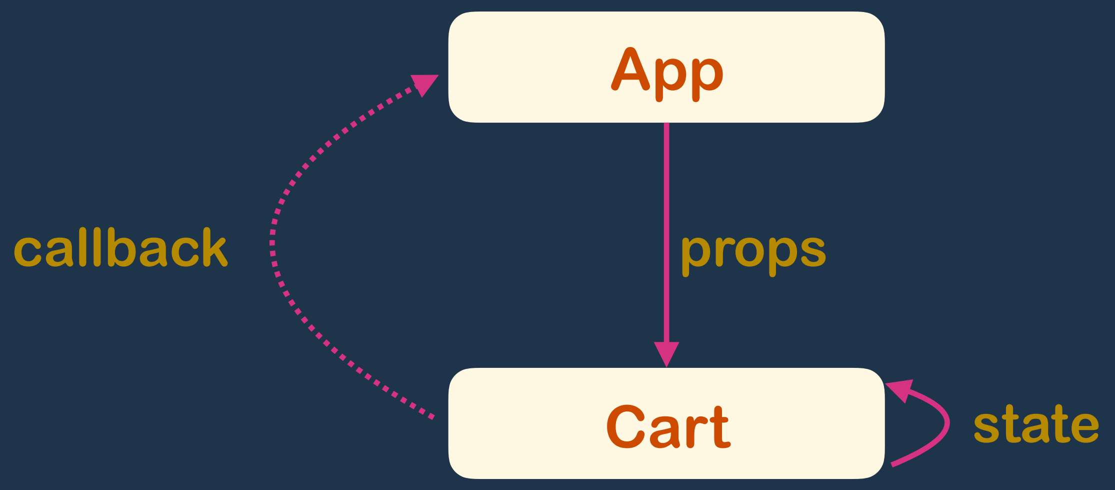
Cart

Checkout

InventoryItem

CartItem


App
props
Cart
state



App

callback

props

Cart

state

```
export default class App extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      inventory: [],
      cartItems: [],
    }
  }
  render() {
    return (
      <div className="container">
        {/* Other components */}
        <h2>Cart</h2>
        <Cart
          cartItems={this.state.cartItems}
          removeFromCart={this.removeFromCart}
          updateQuantity={this.updateQuantity}
        />
      </div>
    );
  }

  removeFromCart = item => { /* code here */ }
  updateQuantity = (item, quantity) => { /* code here */ }
}
```

```
export default class App extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      inventory: [],
      cartItems: [],
    }
  }
  render() {
    return (
      <div className="container">
        {/* Other components */}
        <h2>Cart</h2>
        <Cart
          cartItems={this.state.cartItems}
          removeFromCart={this.removeFromCart}
          updateQuantity={this.updateQuantity}
        />
      </div>
    );
  }

  removeFromCart = item => { /* code here */ }
  updateQuantity = (item, quantity) => { /* code here */ }
}
```

state

pass in props

```
export default class App extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      inventory: [],
      cartItems: [],
    }
  }
  render() {
    return (
      <div className="container">
        {/* Other components */}
        <h2>Cart</h2>
        <Cart
          cartItems={this.state.cartItems}
          removeFromCart={this.removeFromCart}
          updateQuantity={this.updateQuantity}
        />
      </div>
    );
  }

  removeFromCart = item => { /* code here */ }
  updateQuantity = (item, quantity) => { /* code here */ }
}
```

value prop

```
export default class App extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      inventory: [],
      cartItems: [],
    }
  }
  render() {
    return (
      <div className="container">
        {/* Other components */}
        <h2>Cart</h2>
        <Cart
          cartItems={this.state.cartItems}
          removeFromCart={this.removeFromCart}
          updateQuantity={this.updateQuantity}
        />
      </div>
    );
  }

  removeFromCart = item => { /* code here */ }
  updateQuantity = (item, quantity) => { /* code here */ }
}
```

callback prop

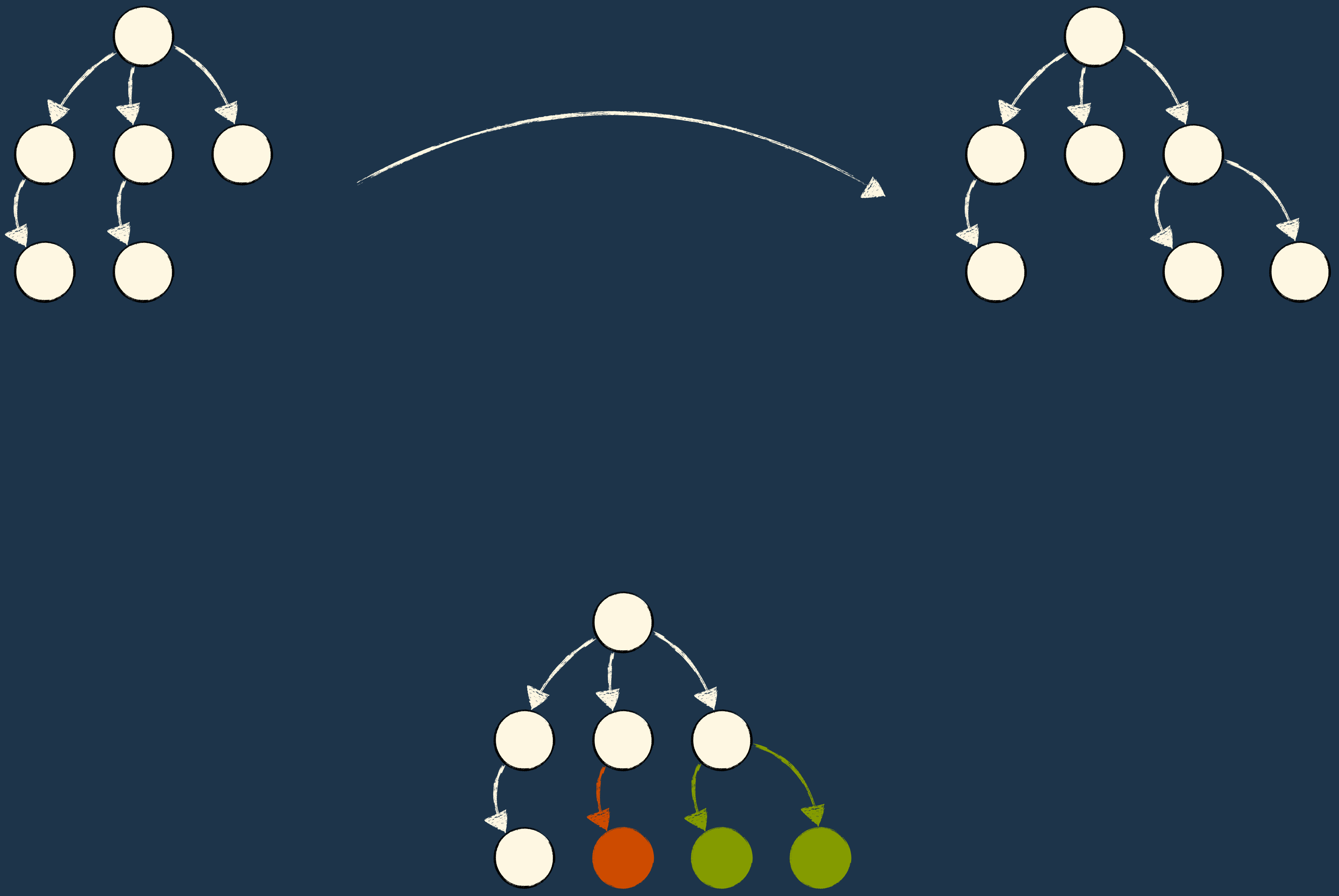
PROPS ARE IMMUTABLE!

STATE IS MUTABLE!*

VIRTUAL DOM

vDom

vDom



Real DOM

JSX

JSX

- AN EASIER WAY TO WRITE REACT.CREATEELEMENT
- REPRESENTS THE “MARKUP” IN COMPONENTS

```
return (
  <Checkout
    total={this.calculateTotal()} />
);
```

becomes

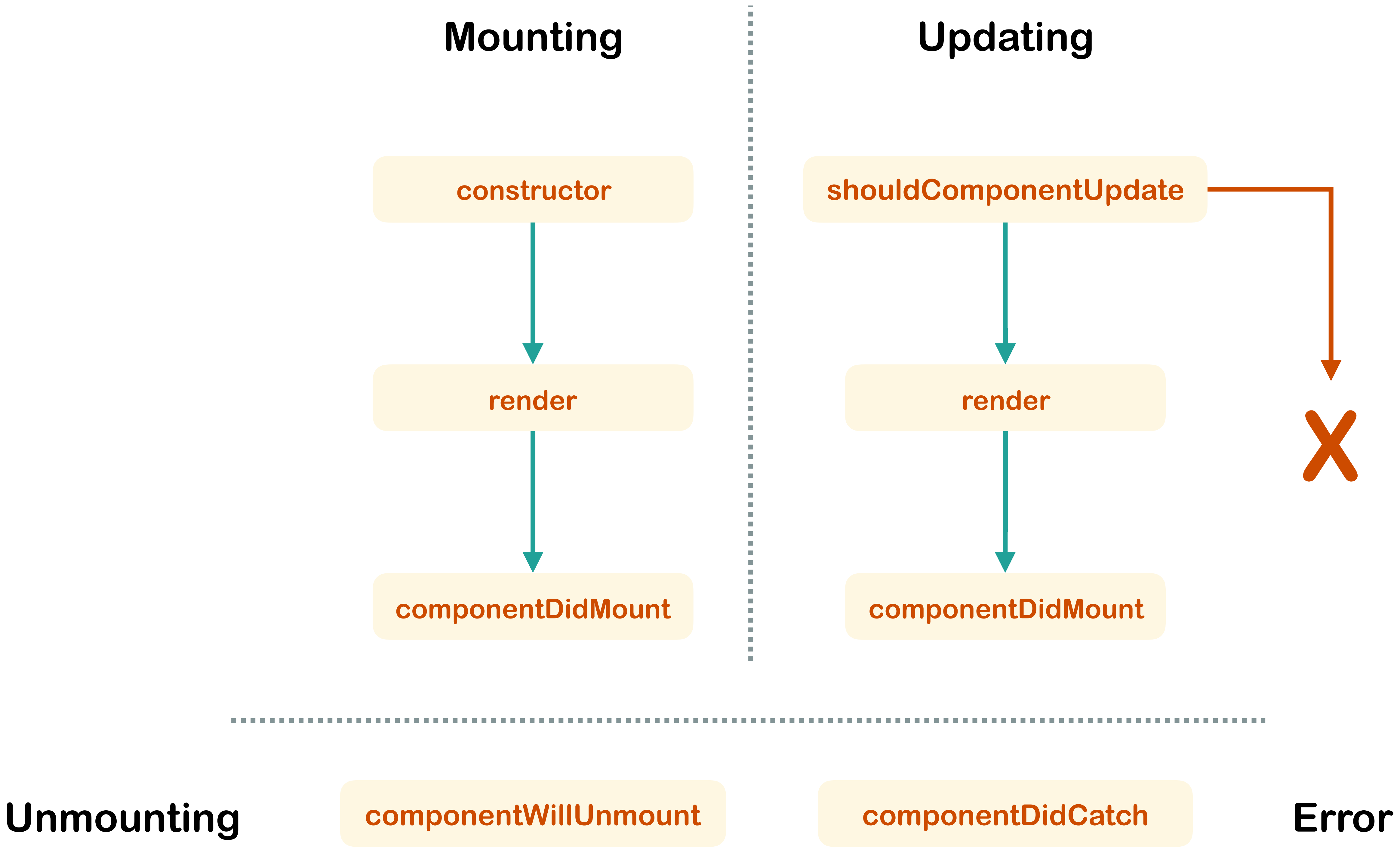
```
return React.createElement(Checkout, {
  total: this.calculateTotal()
});
```

```
<tbody>
  {
    cartItems.map(({ item, quantity }) =>
      <CartItem
        key={item.id}
        item={item}
        quantity={quantity}
        removeFromCart={removeFromCart}
        updateQuantity={updateQuantity}
      />
    )
  }
</tbody>
```

```
cartItems.length > 0
  ? <table id="cart" className="table table-hover table-condensed">
    {/* (thead>tr>td*n)+(tbody>tr>td*n) */}
  </table>
  : <div className="alert alert-success" role="alert">
    <h4 className="alert-heading">Empty Cart</h4>
  </div>
```

JSX IS JAVASCRIPT!

LIFECYCLE



Mounting

constructor

render

componentDidMount

Updating

shouldComponentUpdate

render

componentDidMount

X

Unmounting

componentWillUnmount

componentDidCatch

Error

FUNCTIONAL

IMMUTABILITY

UNIDIRECTIONAL

FUNCTIONAL COMPONENTS

SIMPLE

TOOLING

TOOLING

- REACT DEV TOOLS
- FACEBOOK “CREATE REACT APP”
- TESTING REACT WITH JEST

WHERE NEXT?

NEXT ACTION ITEMS

- REACT DOCS
- ROAD TO REACT (FREE EBOOK)
- REACT STARTER KIT
- REACT TUTORIAL
- 5 PROJECTS TO HELP YOU LEARN REACT

THANKS!!!