REVIEWING **PROPS**

REVIEWING **PROPS**

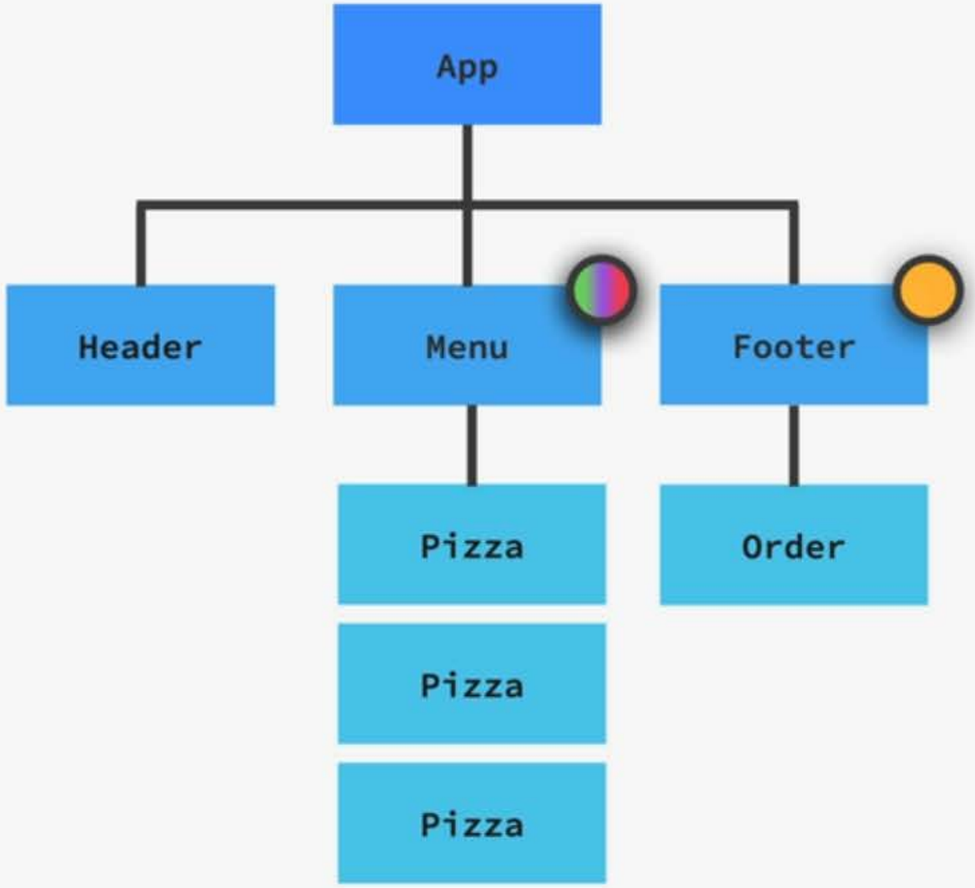
PROPS

- 👉 Props are used to pass data from **parent components** to **child components** (**down** the component tree)

REVIEWING **PROPS**

PROPS

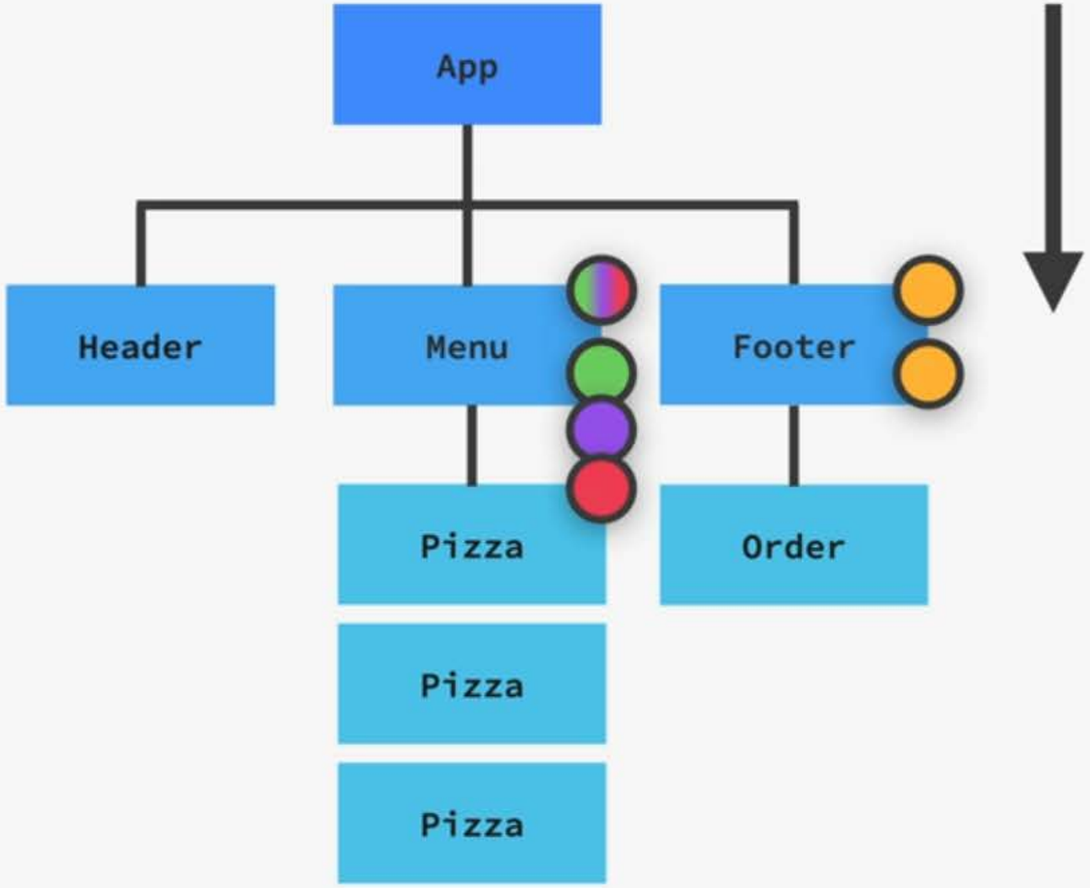
- 👉 Props are used to pass data from **parent components** to **child components** (**down** the component tree)



REVIEWING **PROPS**

PROPS

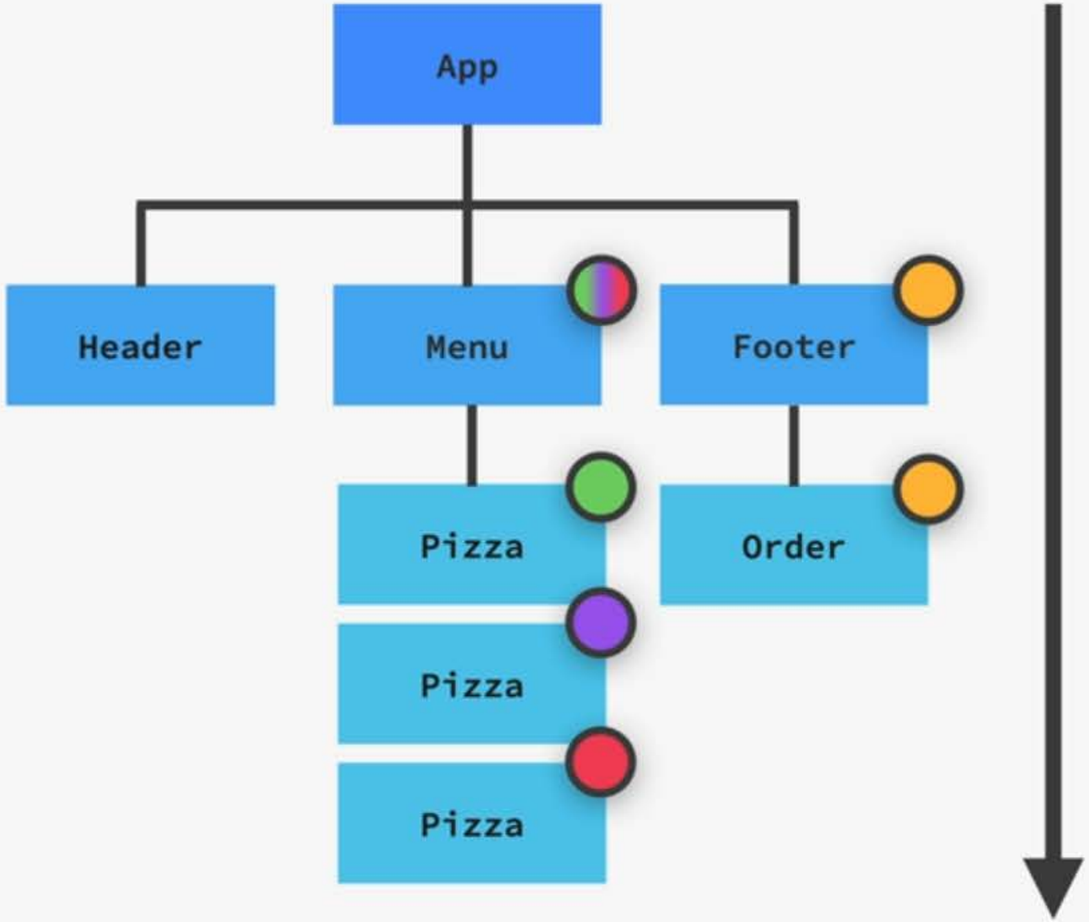
- 👉 Props are used to pass data from **parent components** to **child components** (**down** the component tree)



REVIEWING **PROPS**

PROPS

👉 Props are used to pass data from **parent components** to **child components** (**down** the component tree)



REVIEWING **PROPS**

PROPS

- 👉 Props are used to pass data from **parent components** to **child components** (**down** the component tree)
- 👉 Essential tool to **configure** and **customize** components (like function parameters)

REVIEWING **PROPS**

PROPS

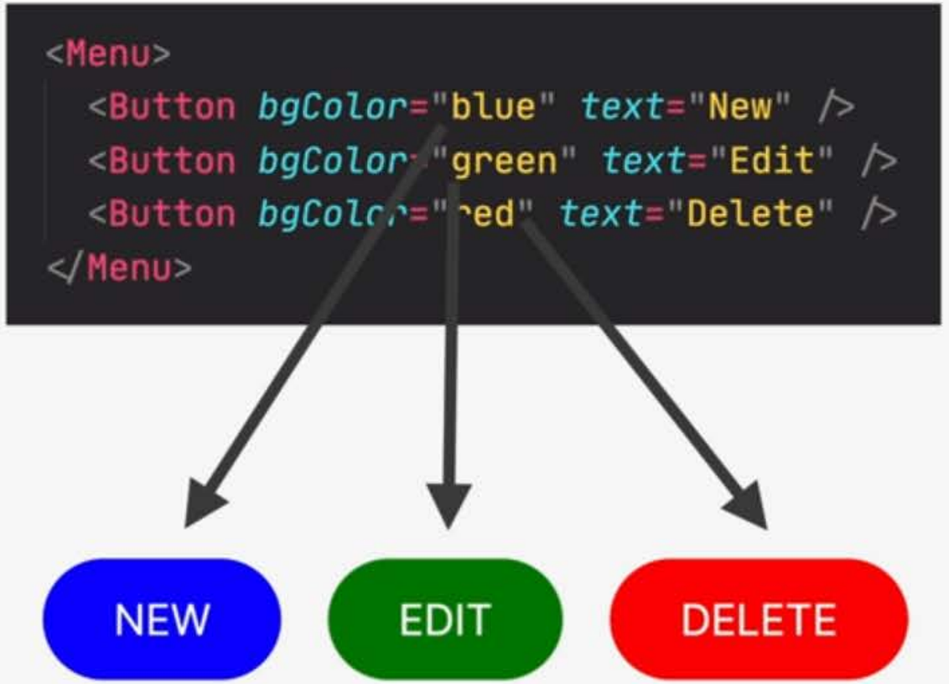
- 👉 Props are used to pass data from **parent components** to **child components** (**down** the component tree)
- 👉 Essential tool to **configure** and **customize** components (like function parameters)

```
<Menu>
  <Button bgColor="blue" text="New" />
  <Button bgColor="green" text="Edit" />
  <Button bgColor="red" text="Delete" />
</Menu>
```

REVIEWING **PROPS**

PROPS

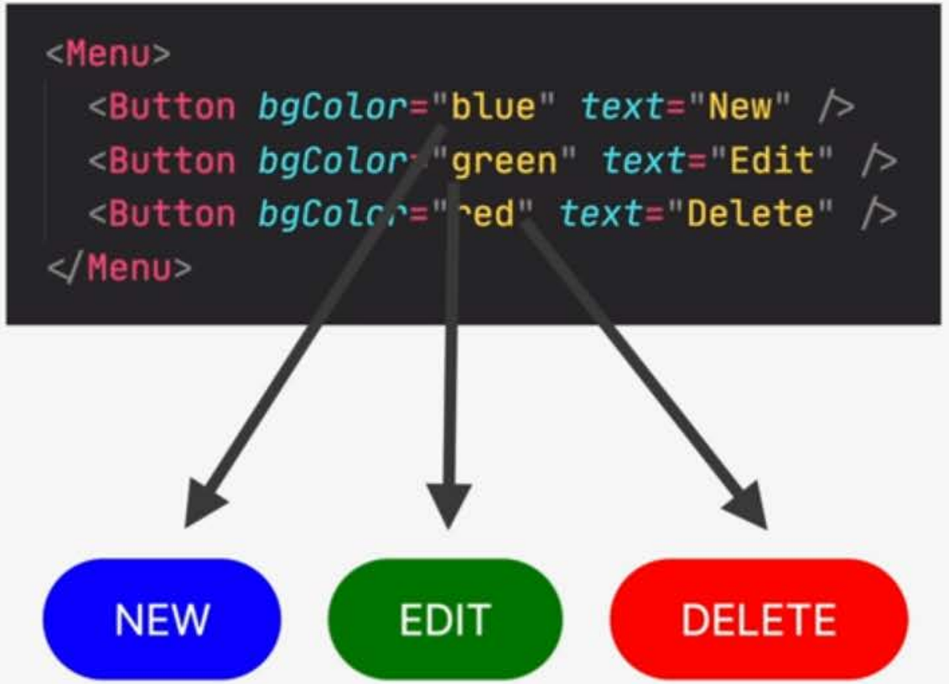
- 👉 Props are used to pass data from **parent components** to **child components** (**down** the component tree)
- 👉 Essential tool to **configure** and **customize** components (like function parameters)



REVIEWING **PROPS**

PROPS

- 👉 Props are used to pass data from **parent components** to **child components** (**down** the component tree)
- 👉 Essential tool to **configure** and **customize** components (like function parameters)
- 👉 With props, parent components **control** how child components look and work



REVIEWING **PROPS**

PROPS

- 👉 Props are used to pass data from **parent components** to **child components** (**down** the component tree)
- 👉 Essential tool to **configure** and **customize** components (like function parameters)
- 👉 With props, parent components **control** how child components look and work
- 👉 **Anything** can be passed as props: single values, arrays, objects, functions, even other components

REVIEWING **PROPS**

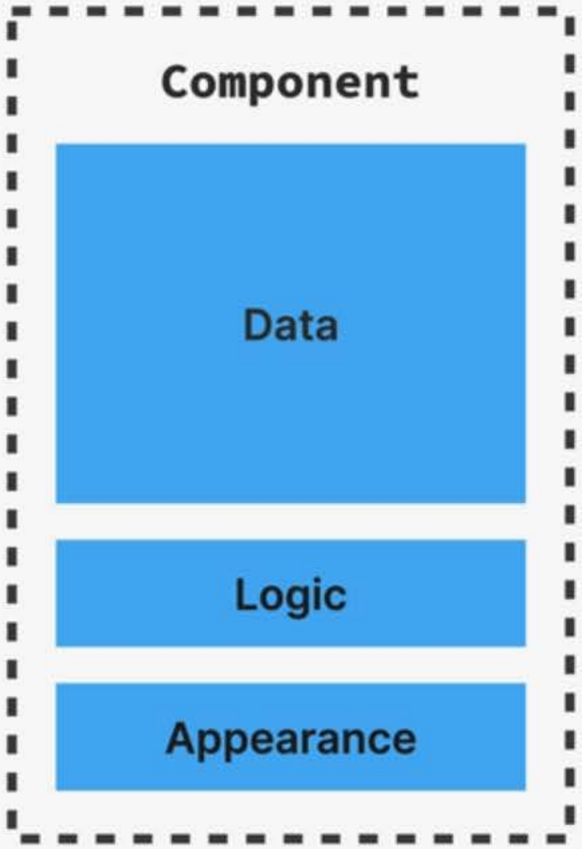
PROPS

- 👉 Props are used to pass data from **parent components** to **child components** (**down** the component tree)
- 👉 Essential tool to **configure** and **customize** components (like function parameters)
- 👉 With props, parent components **control** how child components look and work
- 👉 **Anything** can be passed as props: single values, arrays, objects, functions, even other components

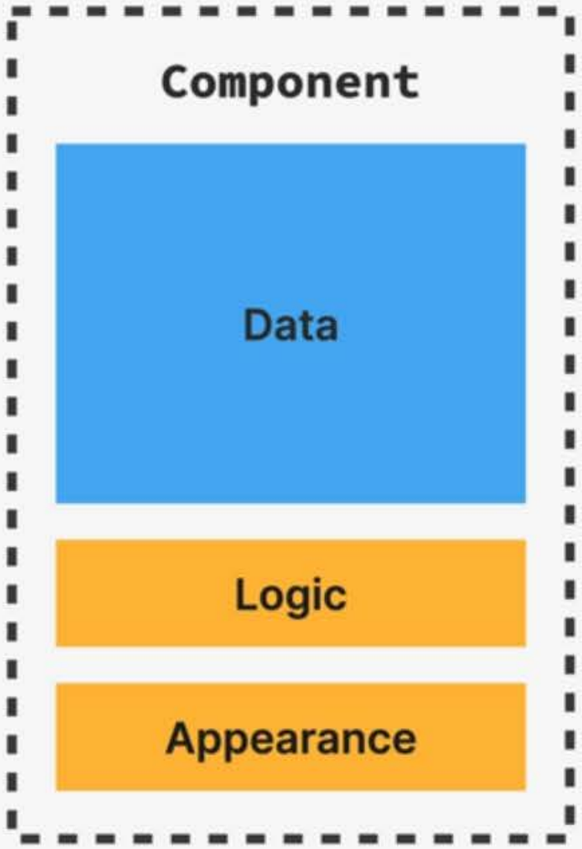
```jsx
function CourseRating() {
  const [rating, setRating] = useState(0);

  return (
    <Rating
      text="Course rating"
      currentRating={rating}
      numOptions={3}
      options={["Terrible", "Okay", "Amazing"]}
      allRatings={{ num: 2390, avg: 4.8 }}
      setRating={setRating}
      component={Star}
    />
  );
}

function Star() {
  // To do
}
```

PROPS ARE **READ-ONLY!**

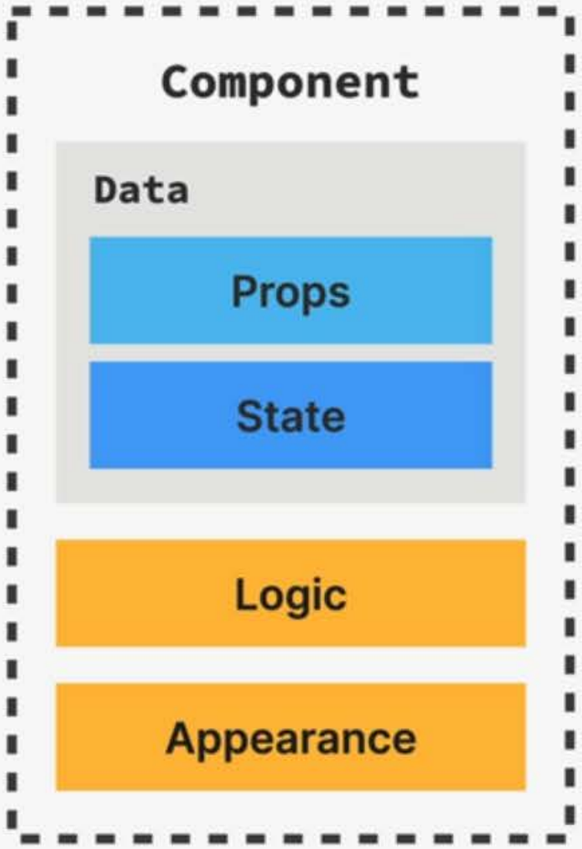
PROPS ARE **READ-ONLY!**



PROPS ARE **READ-ONLY!**

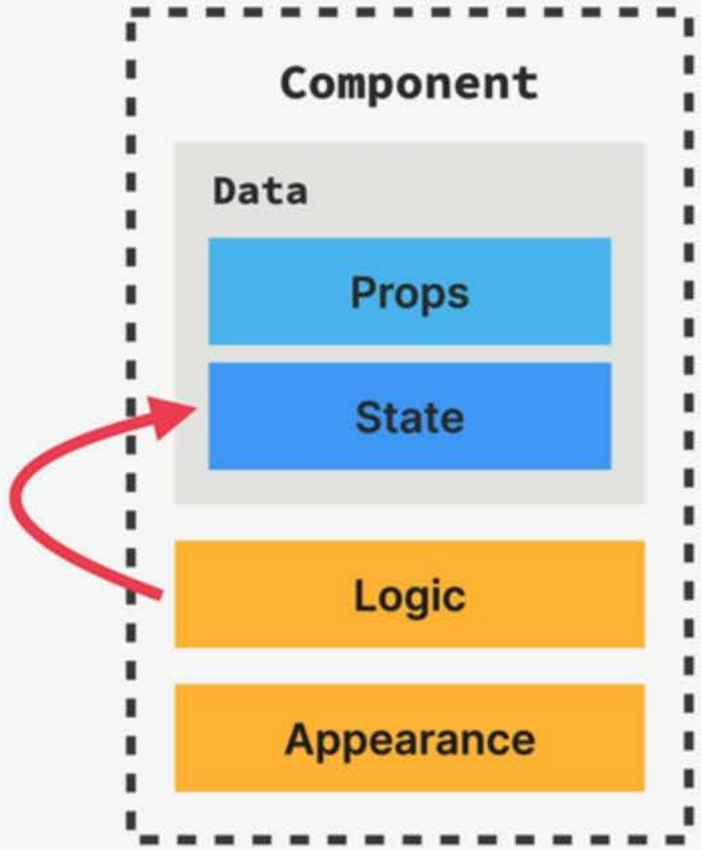


PROPS ARE **READ-ONLY!**



PROPS ARE **READ-ONLY!**

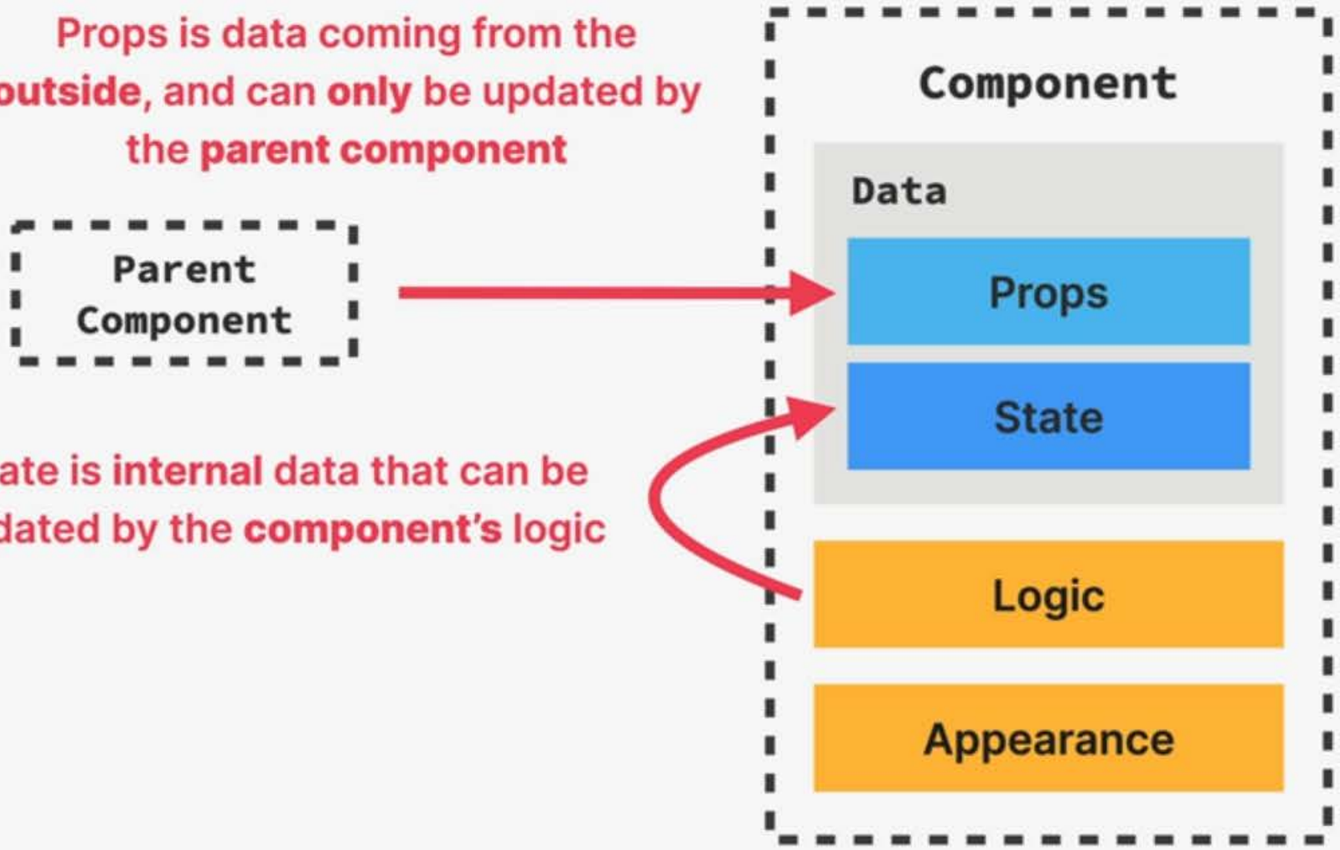
State is internal data that can be updated by the **component's** logic



PROPS ARE **READ-ONLY!**

Props is data coming from the **outside**, and can **only** be updated by the **parent component**

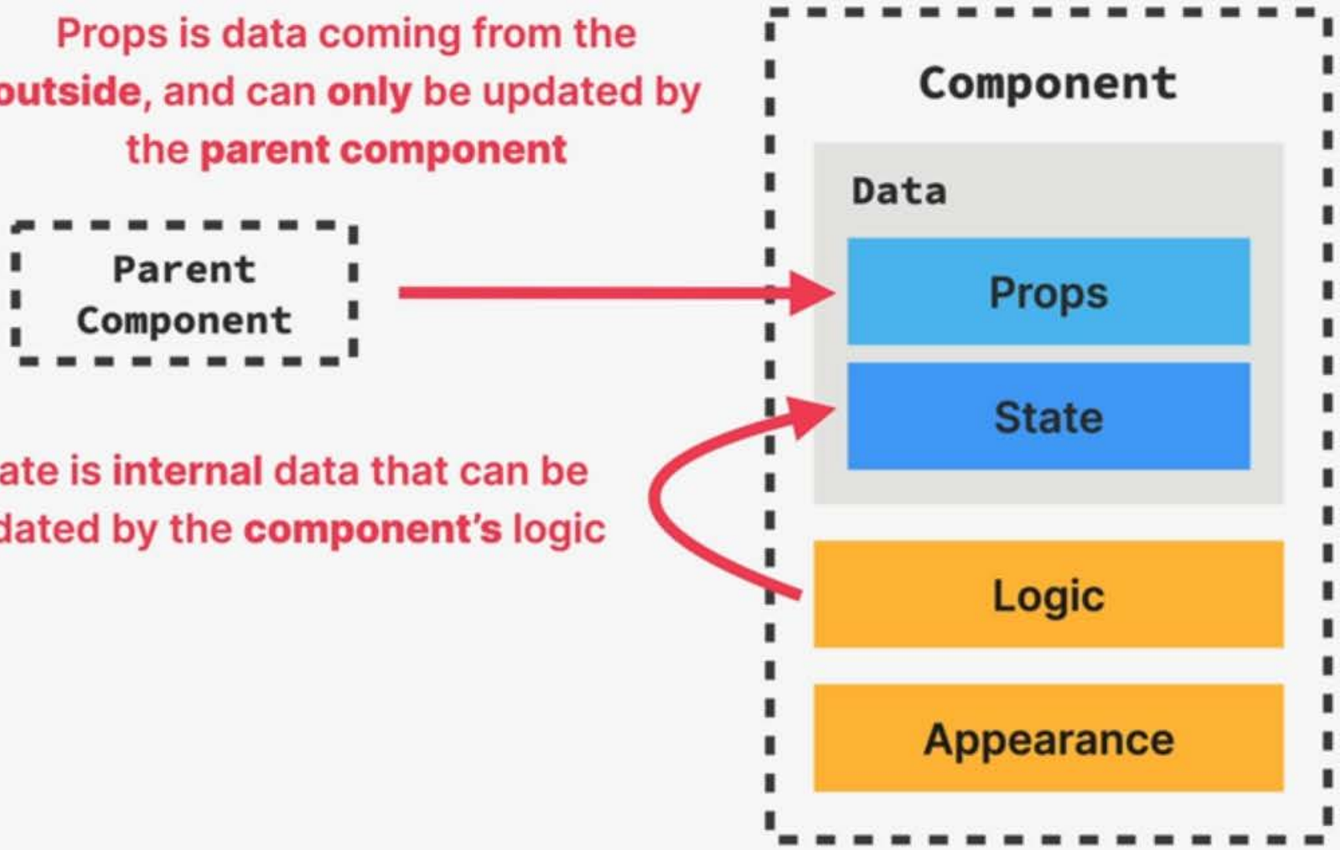
State is internal data that can be updated by the **component's** logic



PROPS ARE **READ-ONLY!**

Props is data coming from the **outside**, and can **only** be updated by the **parent component**

State is internal data that can be updated by the **component's** logic

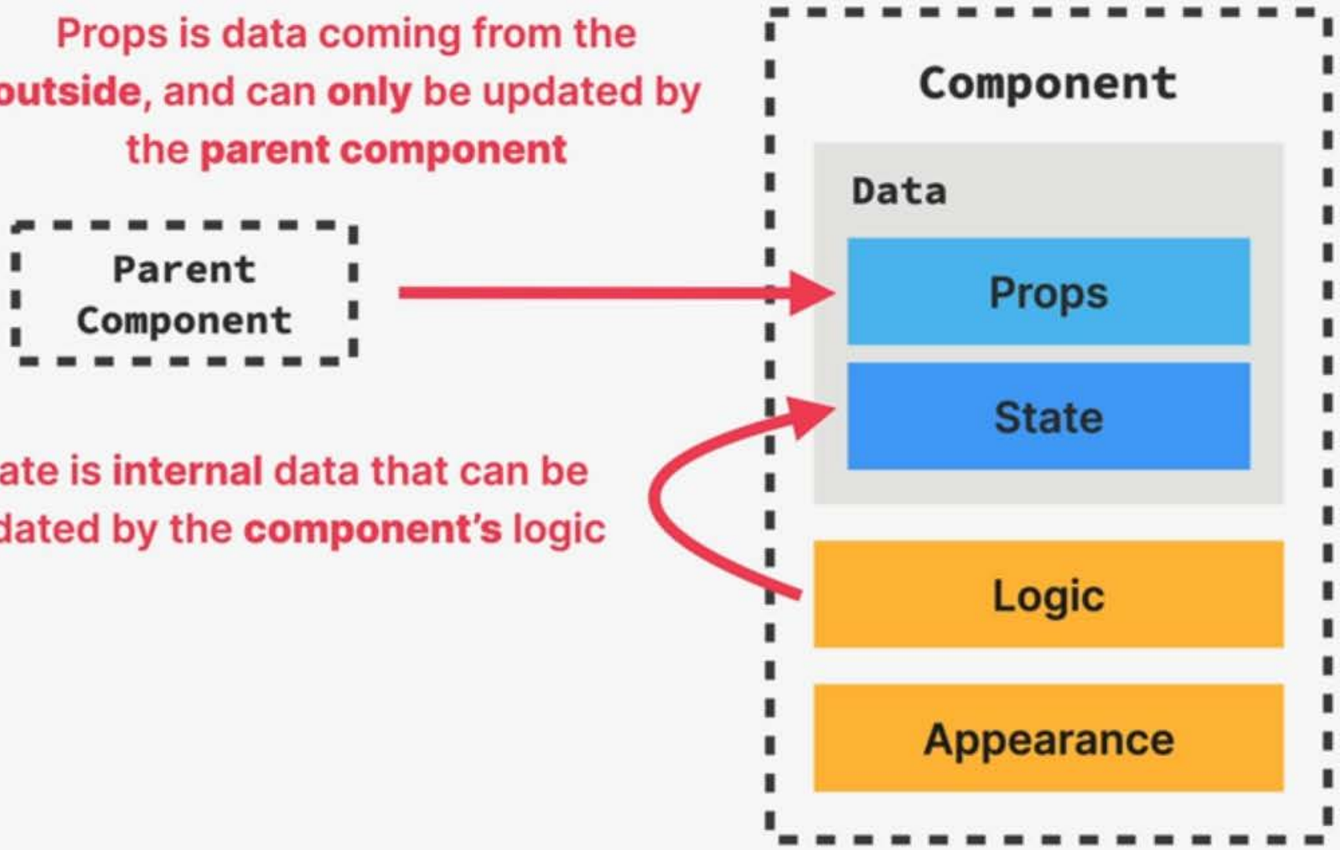


☝ Props are read-only, they are **immutable**! This is one of React's strict rules.

PROPS ARE **READ-ONLY!**

Props is data coming from the **outside**, and can **only** be updated by the **parent component**

State is internal data that can be updated by the **component's** logic



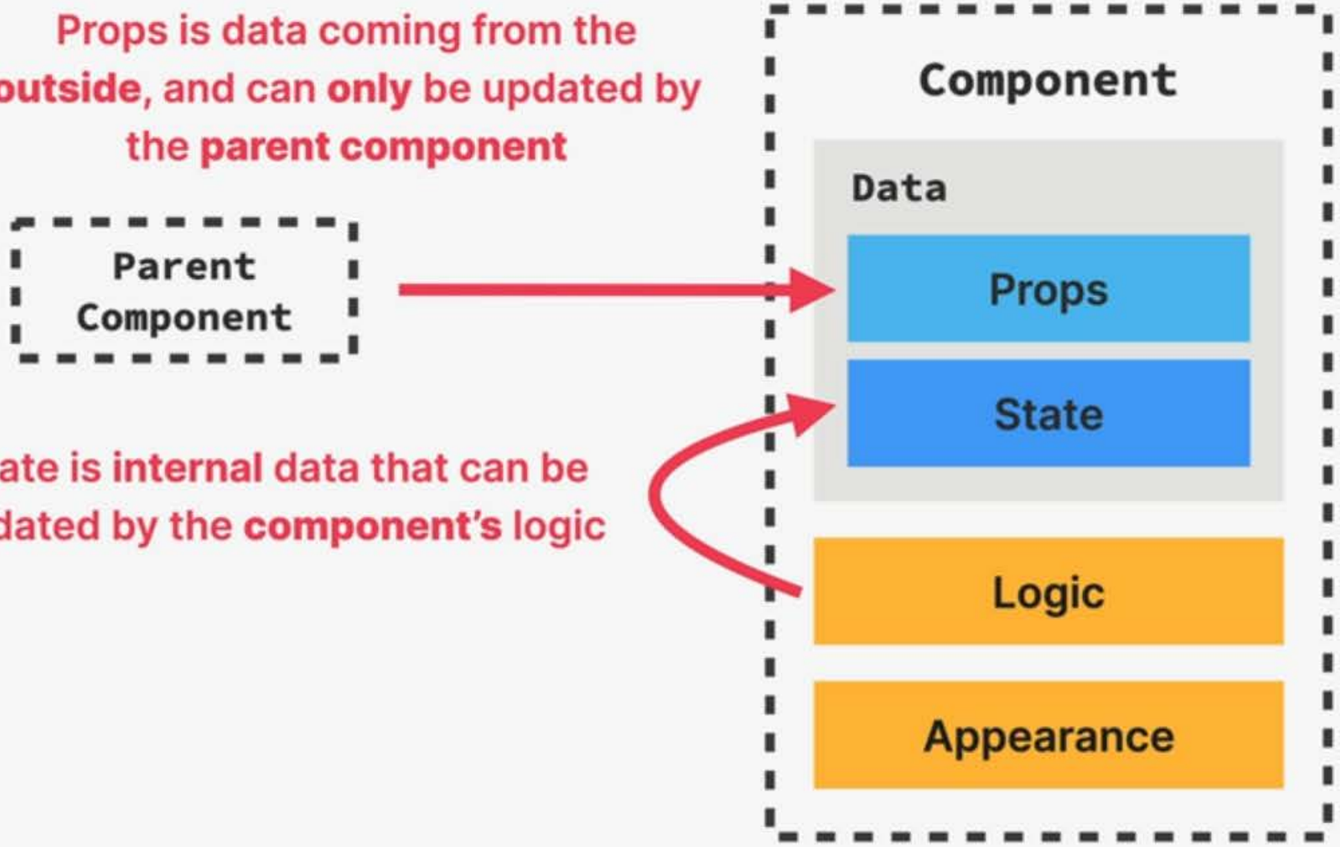
☝ Props are read-only, they are **immutable**! This is one of React's strict rules.

☝ If you need to mutate props, you actually **need state**

PROPS ARE **READ-ONLY!**

Props is data coming from the **outside**, and can **only** be updated by the **parent component**

State is internal data that can be updated by the **component's** logic



☝ Props are read-only, they are **immutable**! This is one of React's strict rules.

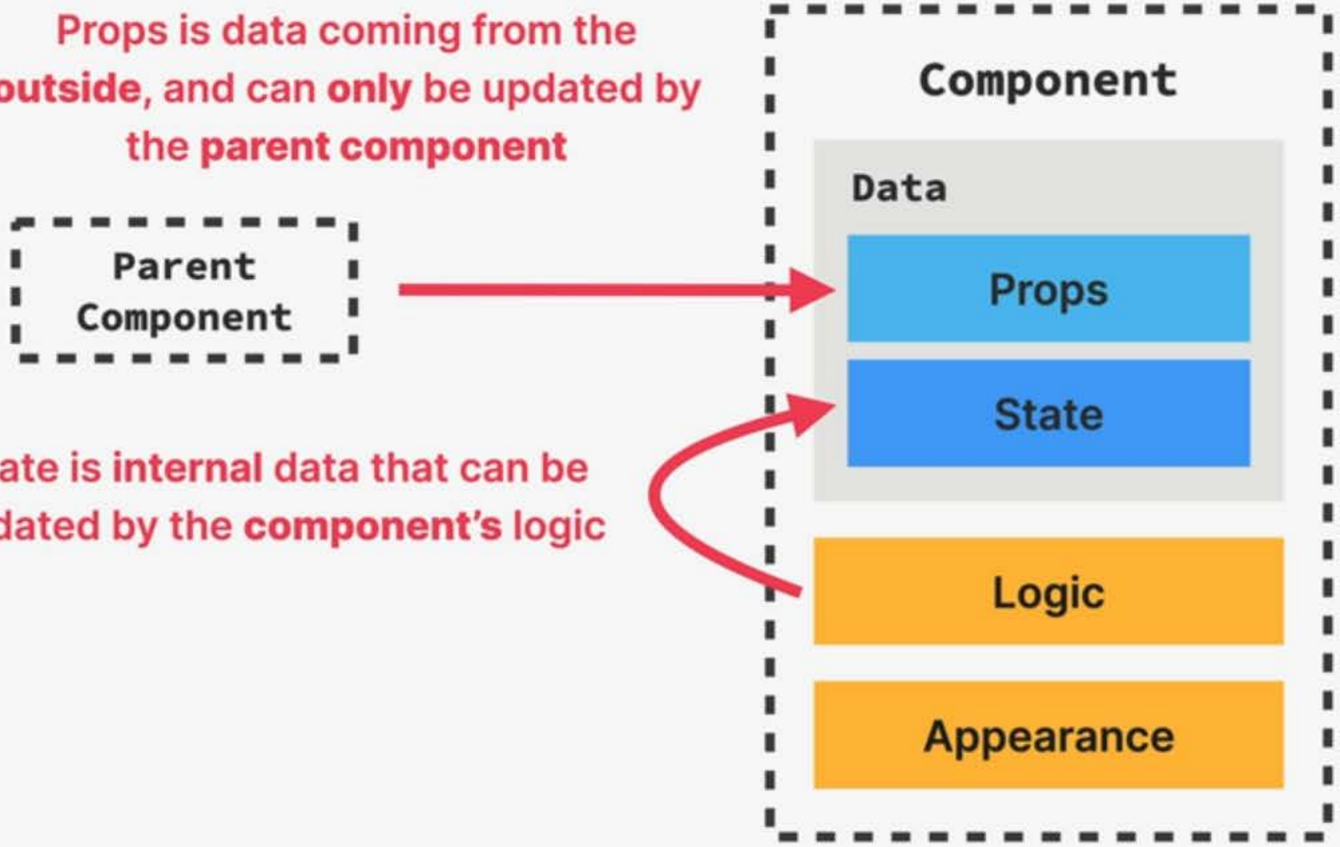
☝ If you need to mutate props, you actually **need state**

↓ WHY?

PROPS ARE **READ-ONLY!**

Props is data coming from the **outside**, and can **only** be updated by the **parent component**

State is internal data that can be updated by the **component's** logic



☝ Props are read-only, they are **immutable**! This is one of React's strict rules.

☝ If you need to mutate props, you actually **need state**

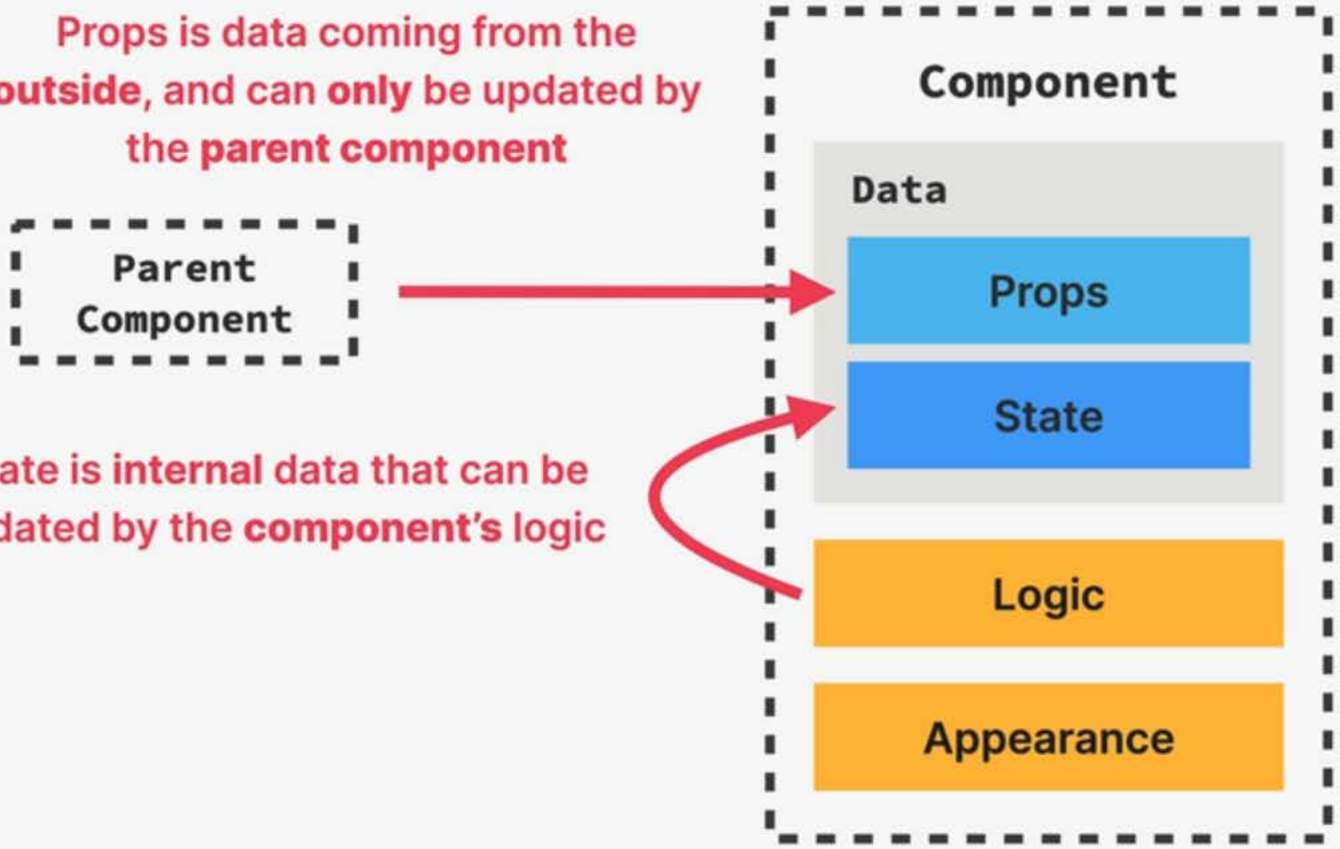
⬇ WHY?

👉 Mutating props would affect parent, creating **side effects** (not pure)

PROPS ARE **READ-ONLY!**

Props is data coming from the **outside**, and can **only** be updated by the **parent component**

State is internal data that can be updated by the **component's** logic



☝ Props are read-only, they are **immutable**! This is one of React's strict rules.

☝ If you need to mutate props, you actually **need state**

⬇ WHY?

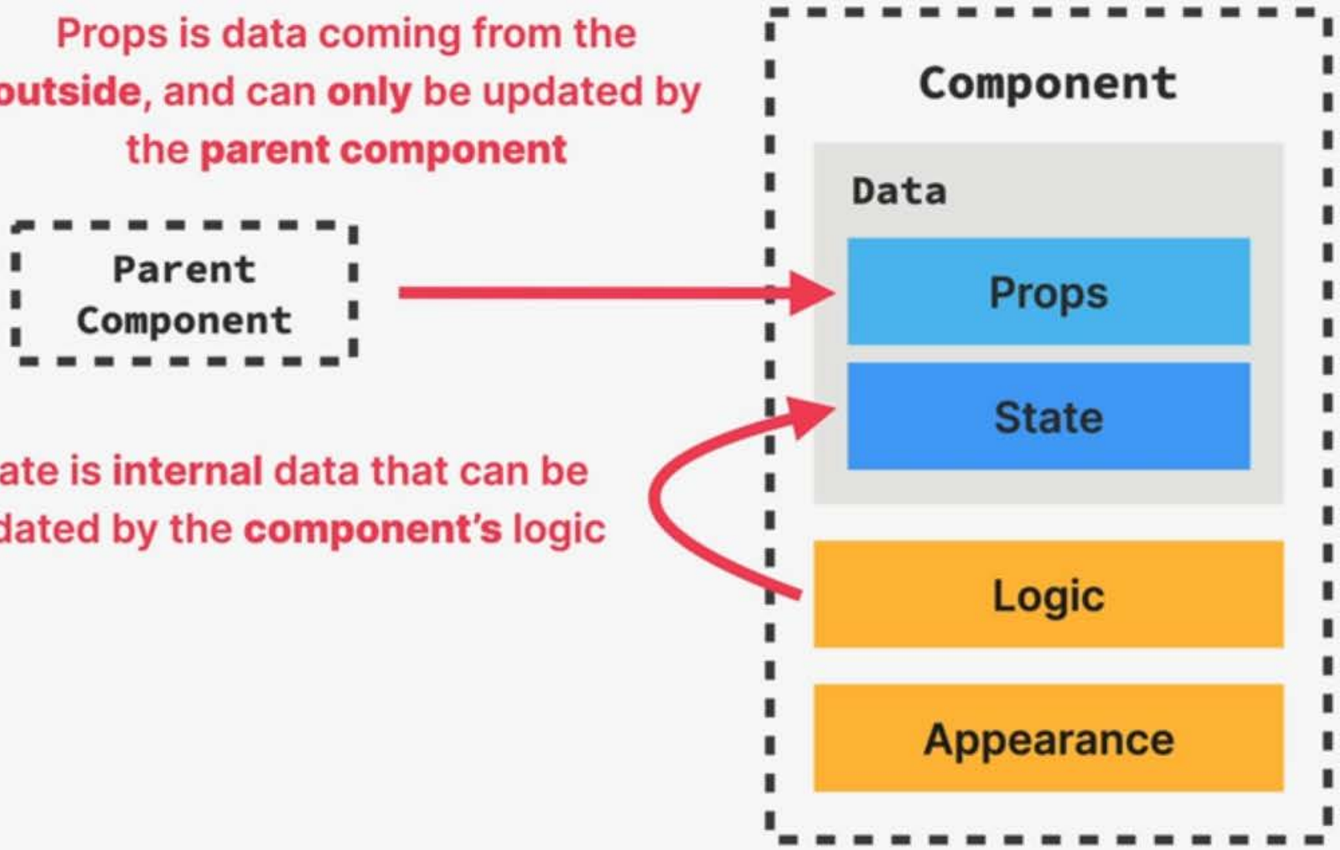
👉 Mutating props would affect parent, creating **side effects** (not pure)

👉 Components have to be **pure functions** in terms of props and state

PROPS ARE **READ-ONLY!**

Props is data coming from the **outside**, and can **only** be updated by the **parent component**

State is internal data that can be updated by the **component's** logic



☝ Props are read-only, they are **immutable**! This is one of React's strict rules.

☝ If you need to mutate props, you actually **need state**

⬇ WHY?

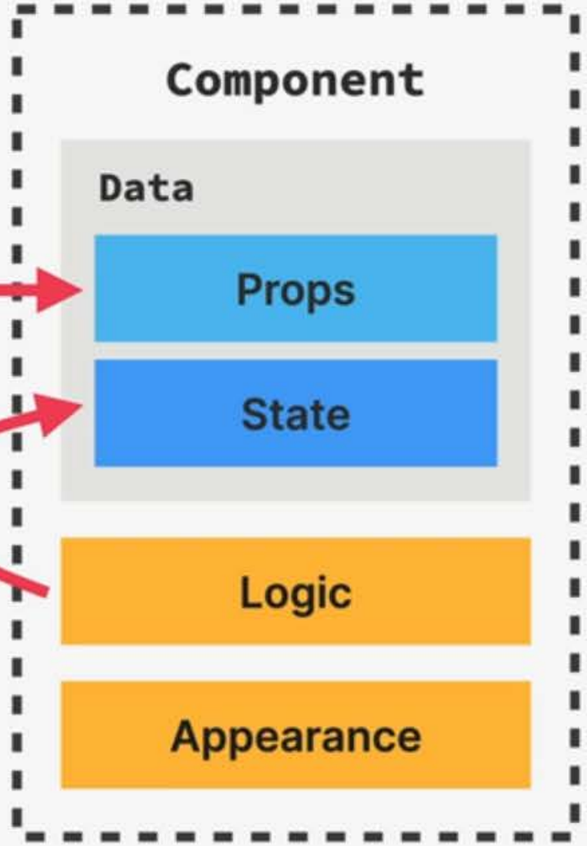
- 👉 Mutating props would affect parent, creating **side effects** (not pure)
- 👉 Components have to be **pure functions** in terms of props and state
- 👉 This allows React to optimize apps, avoid bugs, make apps predictable

PROPS ARE **READ-ONLY!**

Props is data coming from the **outside**, and can **only** be updated by the **parent component**

Parent Component

State is internal data that can be updated by the **component's** logic




```
let x = 7;

function Component(){
  x = 23;
  return <h1>Number {x}</h1>
}
```

Don't do this!

☝ Props are read-only, they are **immutable**! This is one of React's strict rules.

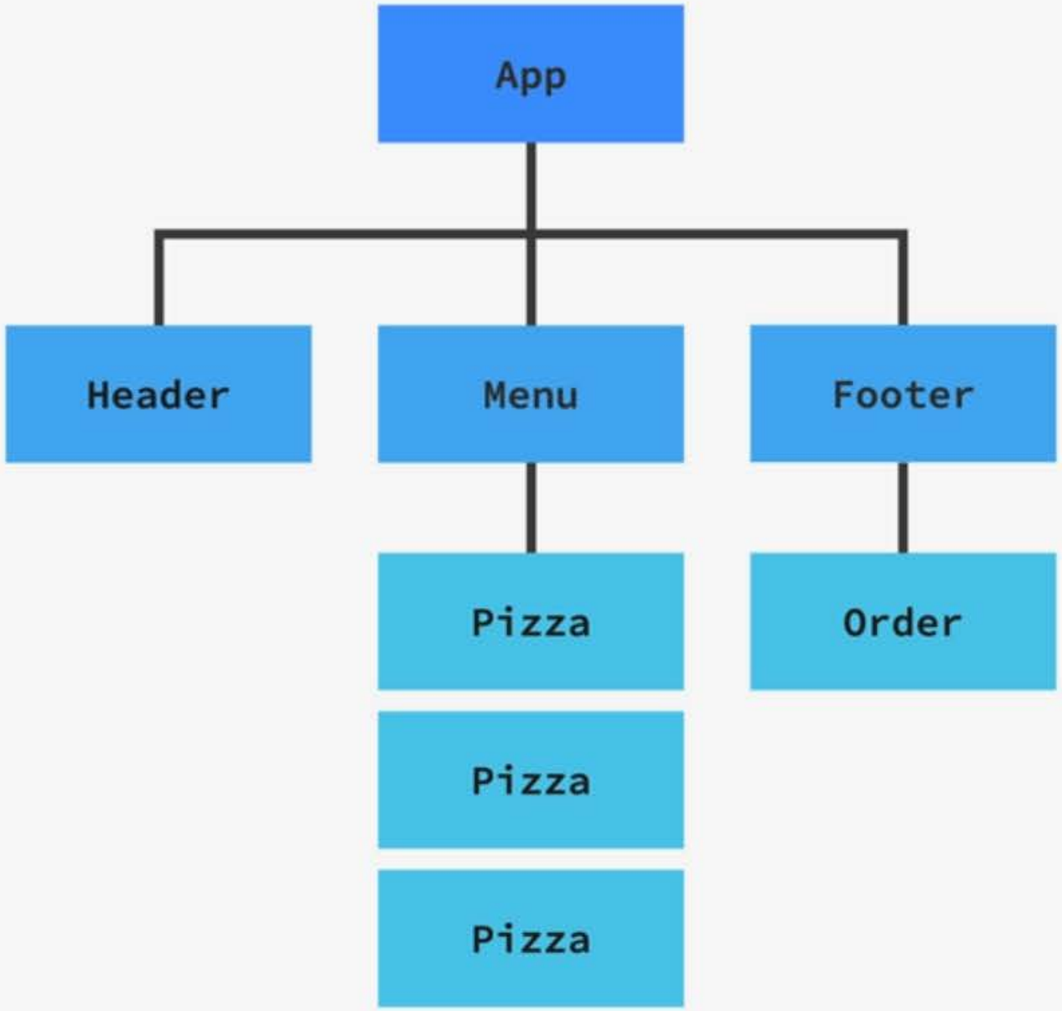
☝ If you need to mutate props, you actually **need state**

⬇ WHY?

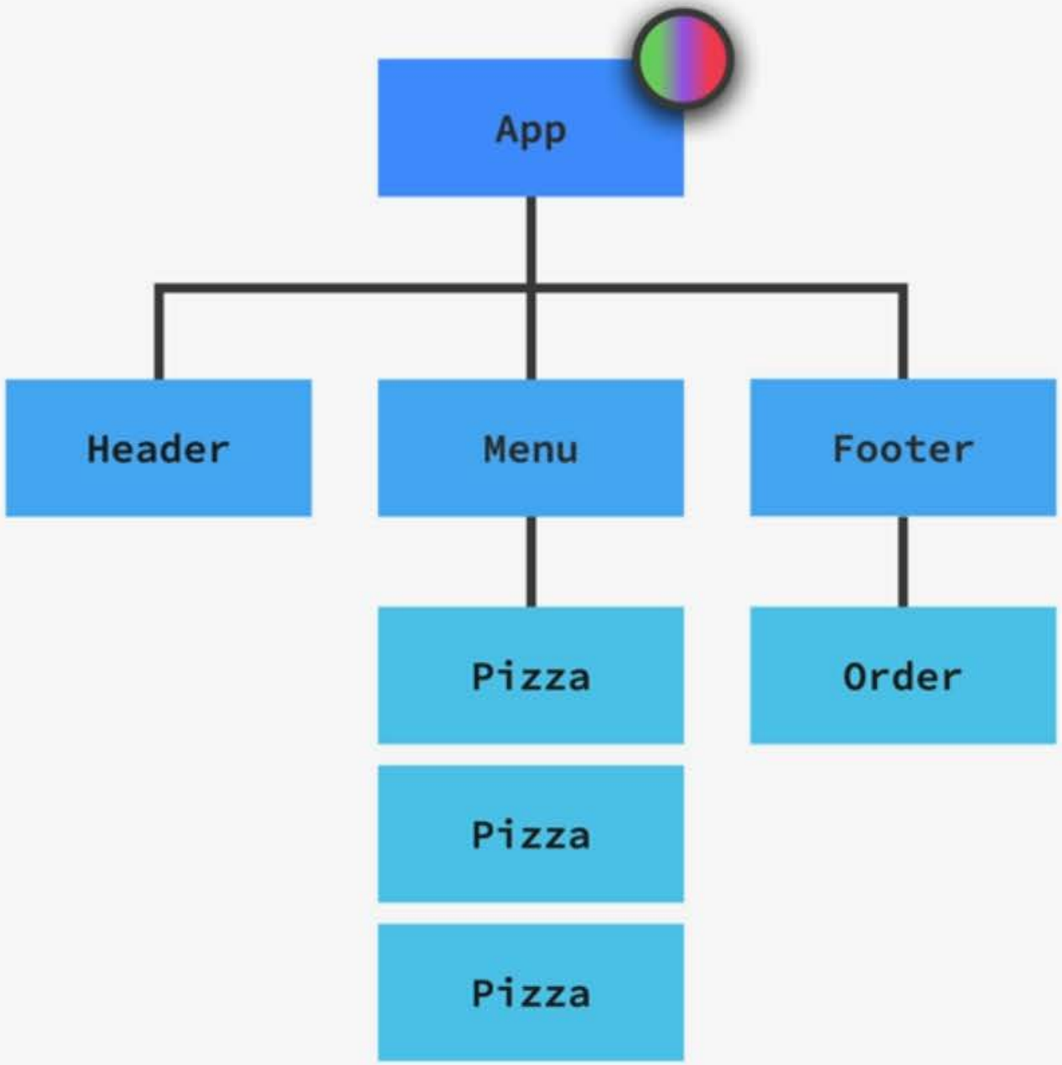
- 👉 Mutating props would affect parent, creating **side effects** (not pure)
- 👉 Components have to be **pure functions** in terms of props and state
- 👉 This allows React to optimize apps, avoid bugs, make apps predictable

ONE-WAY DATA FLOW

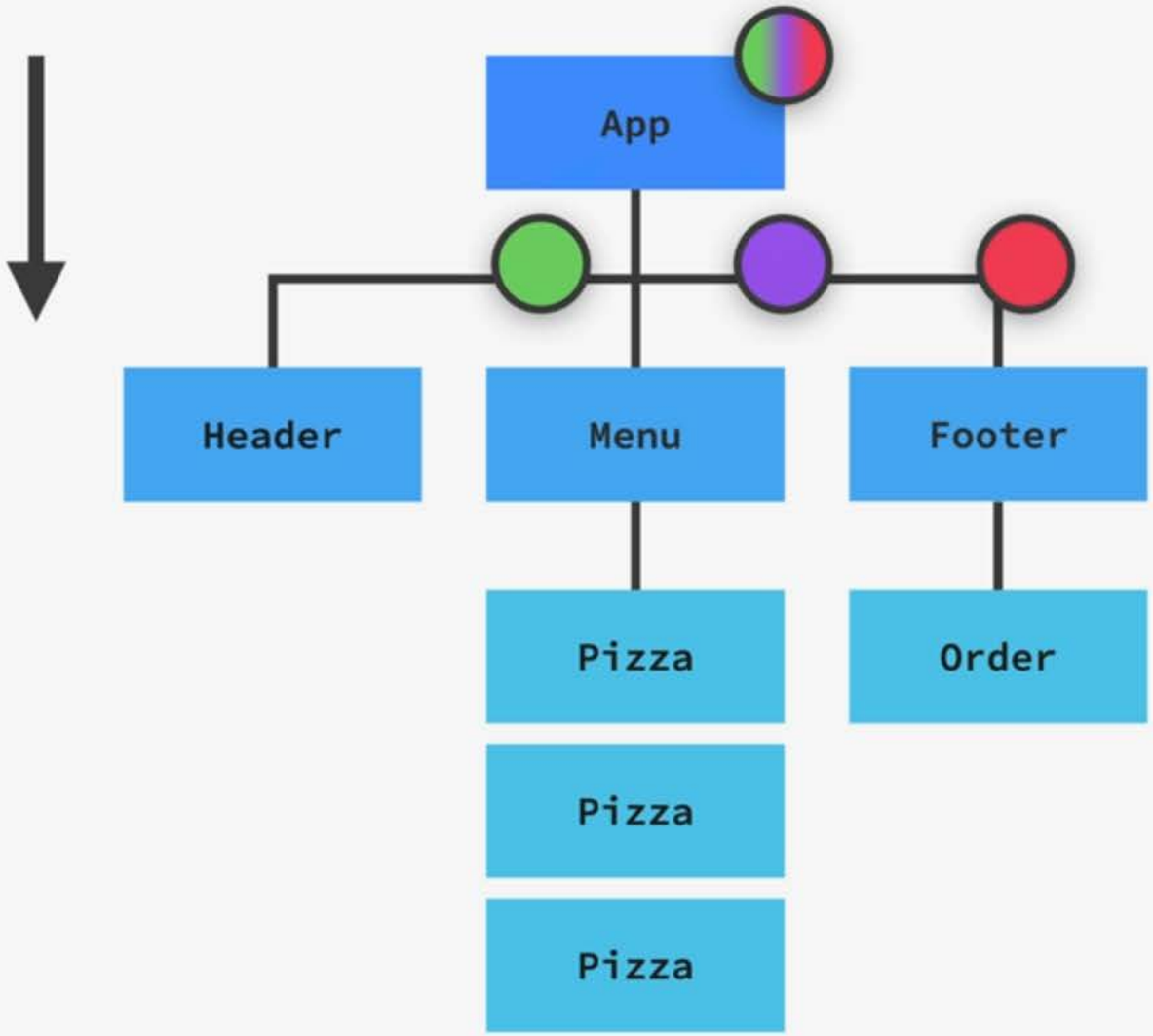
ONE-WAY DATA FLOW



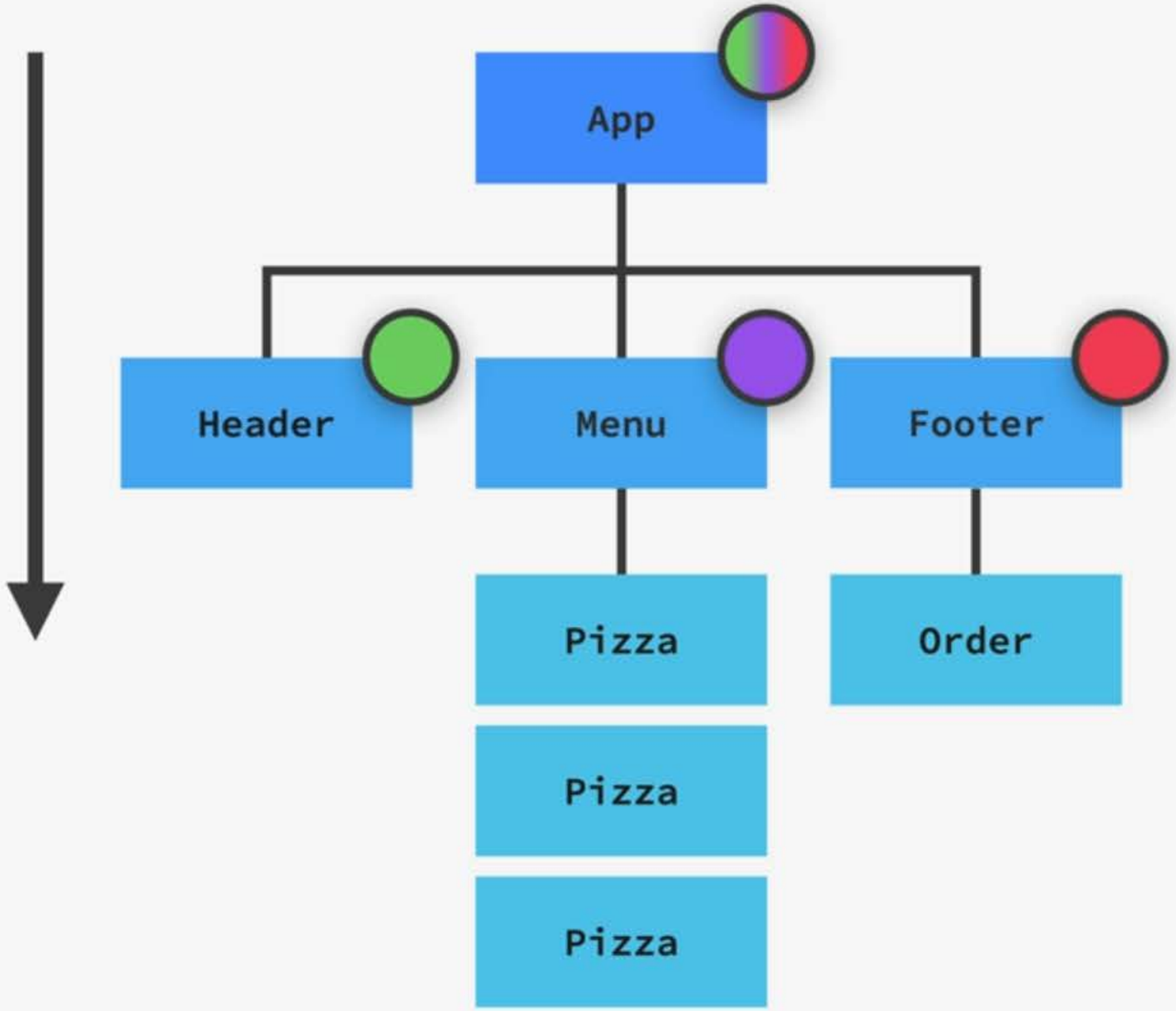
ONE-WAY DATA FLOW



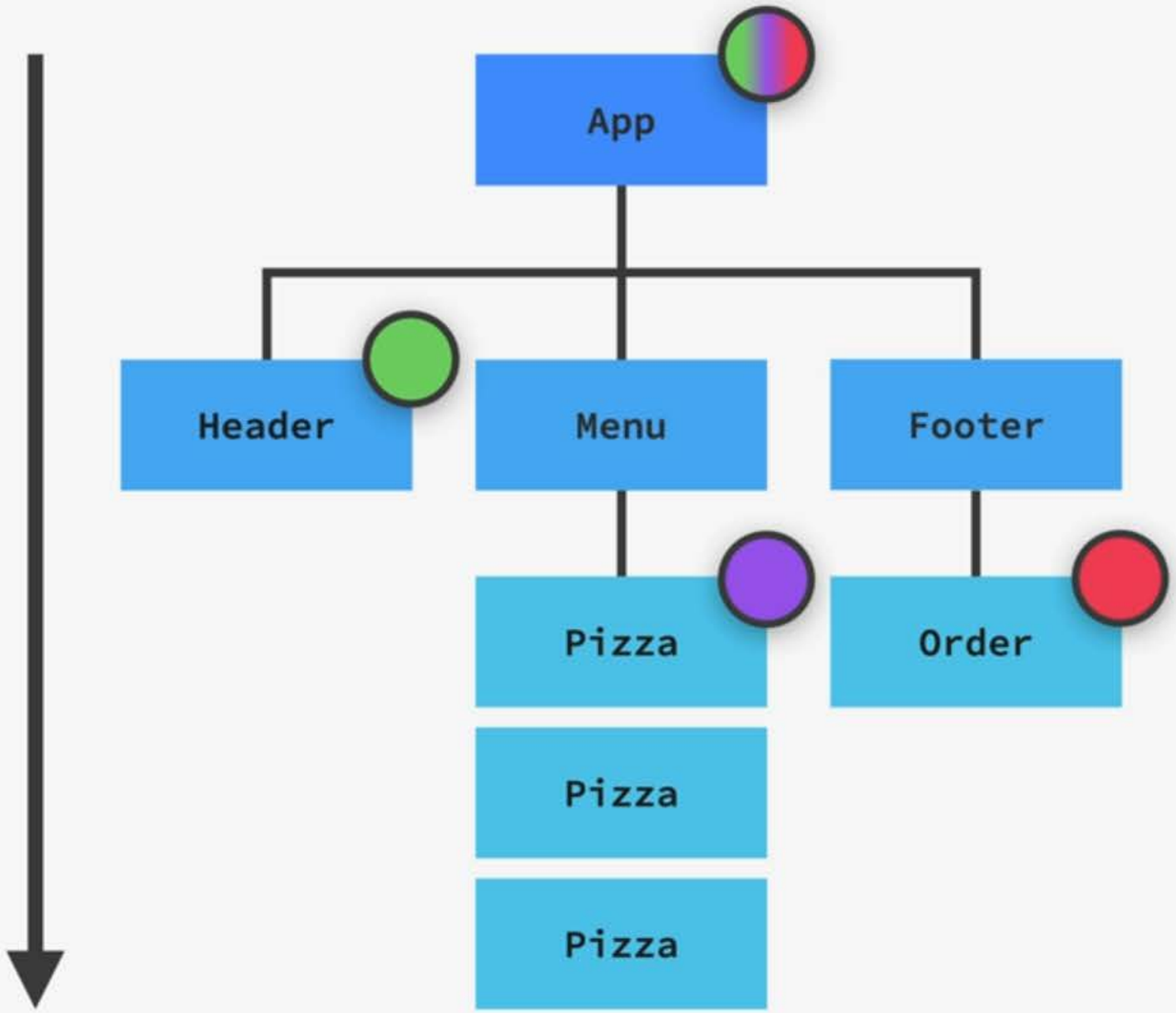
ONE-WAY DATA FLOW



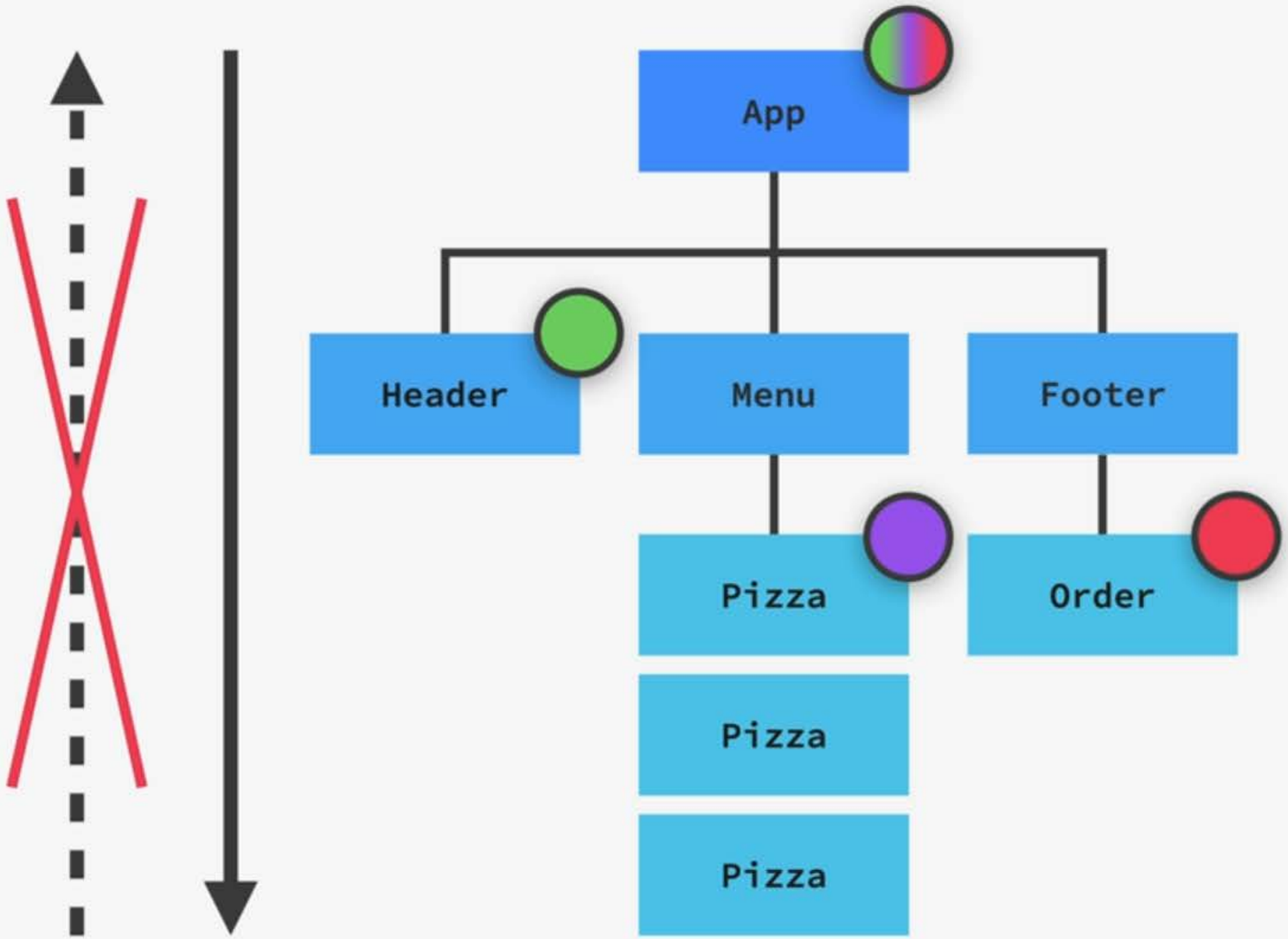
ONE-WAY DATA FLOW



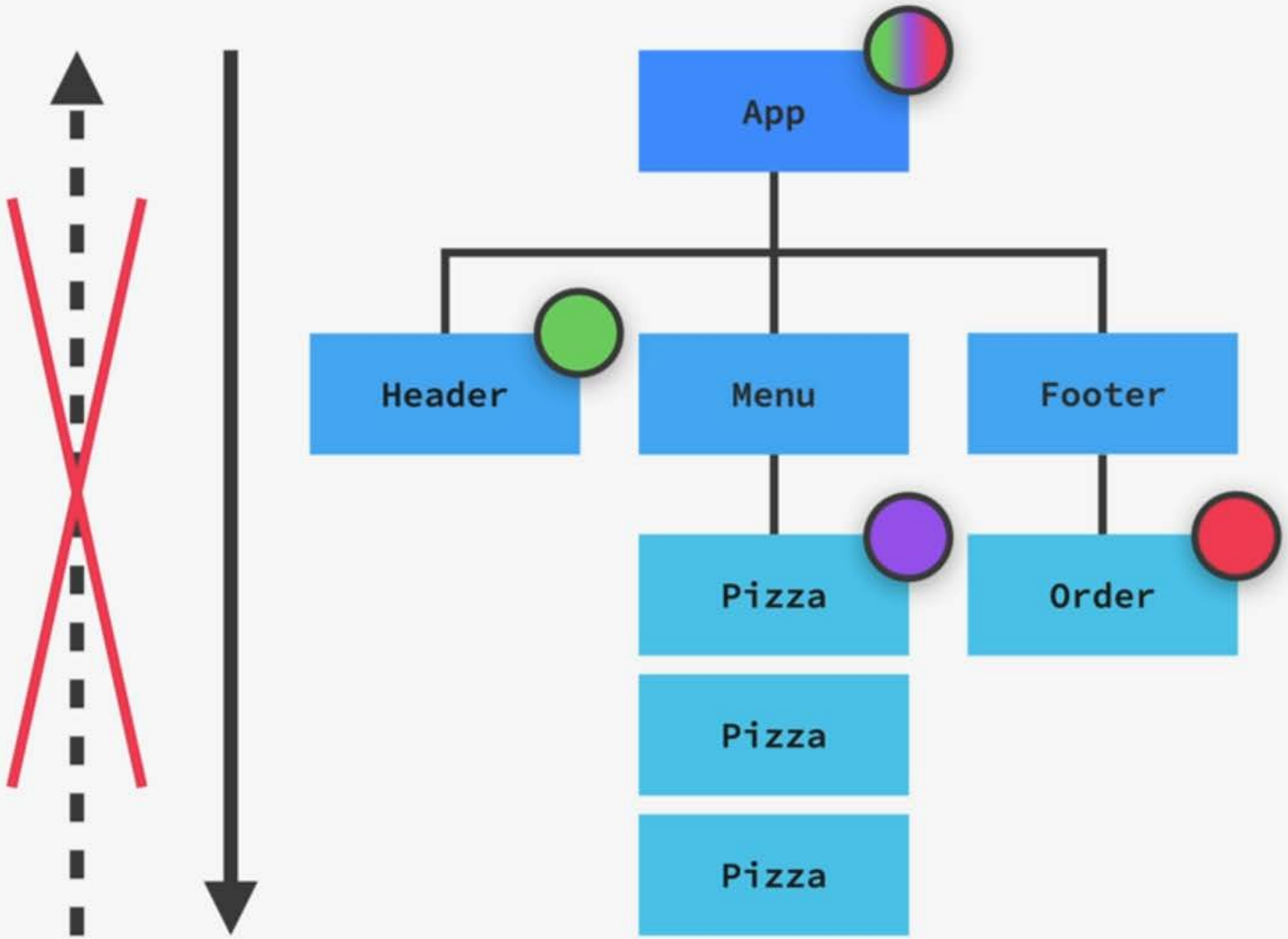
ONE-WAY DATA FLOW



ONE-WAY DATA FLOW

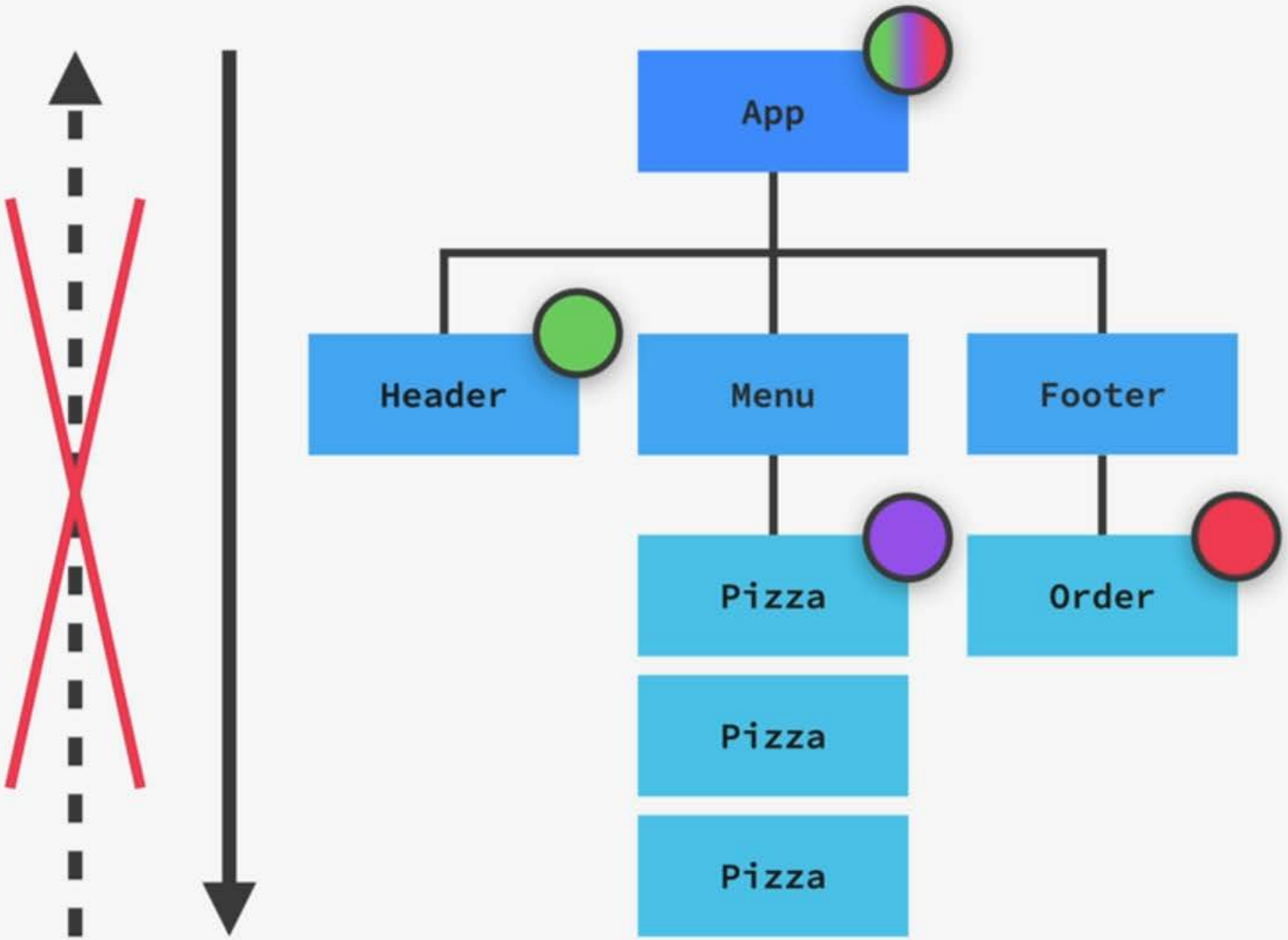


ONE-WAY DATA FLOW



*Angular has **two-way** data flow*

ONE-WAY DATA FLOW

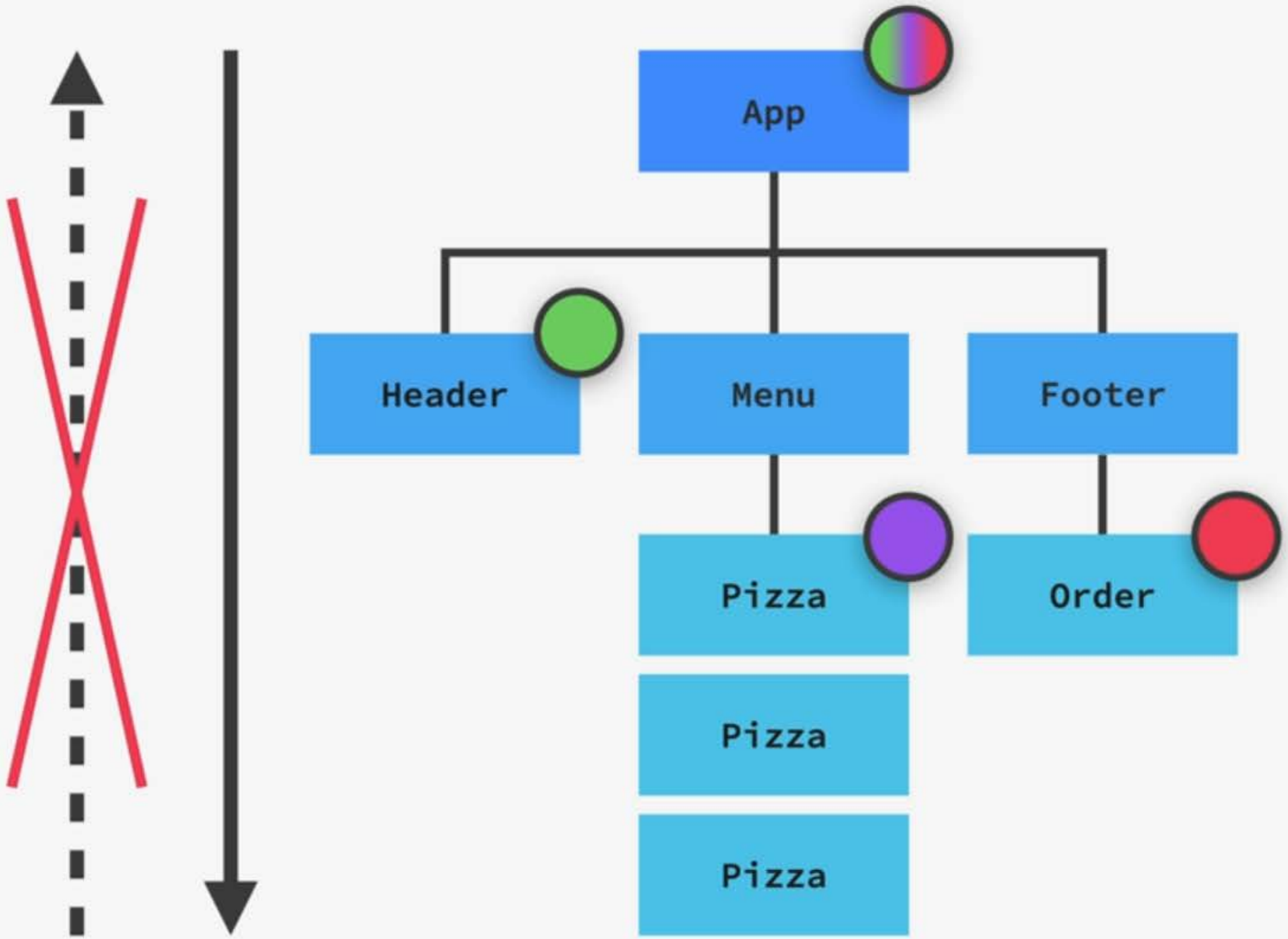


ONE-WAY DATA FLOW...

👍 ... makes applications more predictable and easier to understand

*Angular has **two-way** data flow*

ONE-WAY DATA FLOW



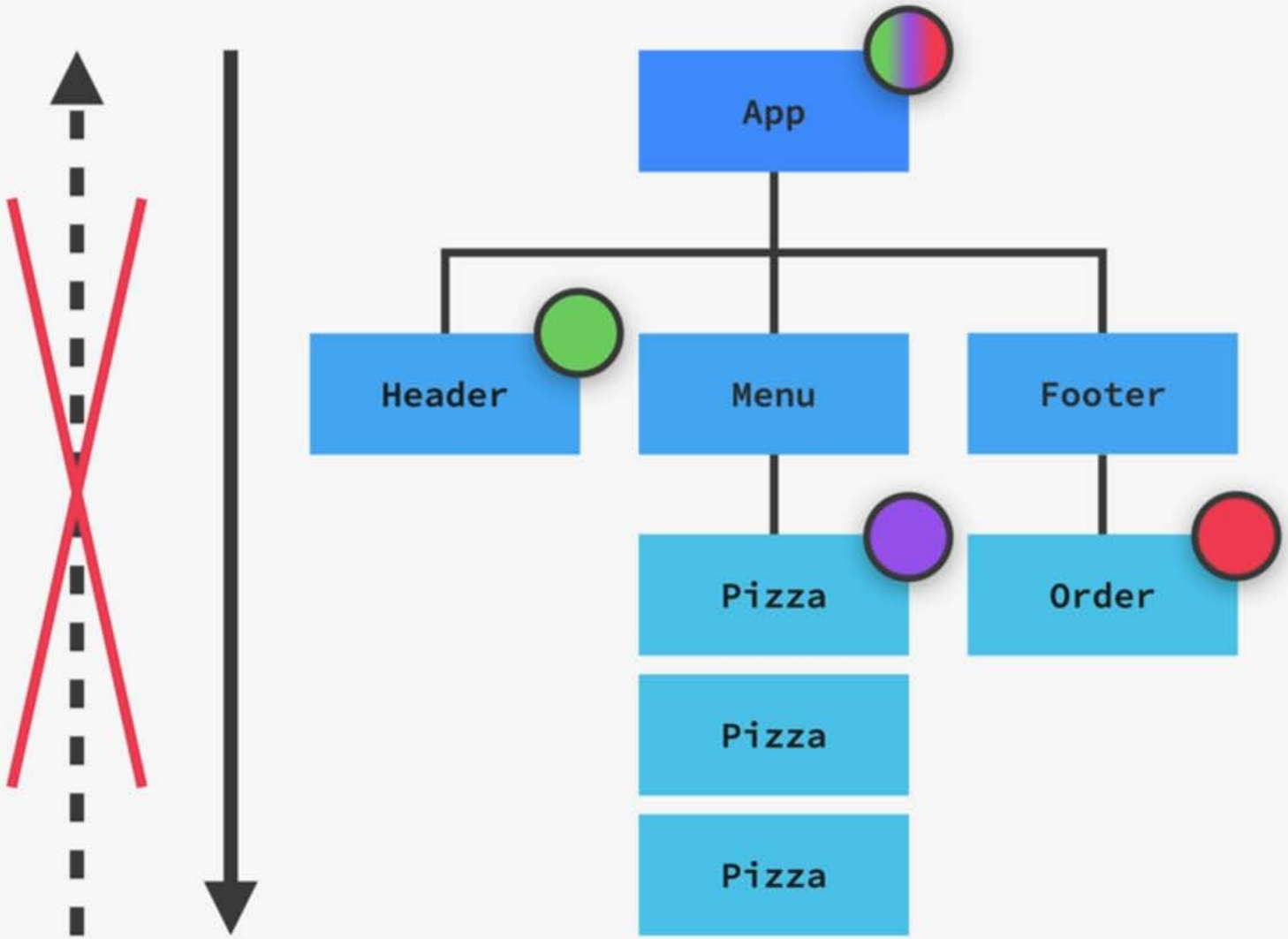
ONE-WAY DATA FLOW...

👍 ... makes applications more predictable and easier to understand

👍 ... makes applications easier to debug, as we have more control over the data

*Angular has **two-way** data flow*

ONE-WAY DATA FLOW



ONE-WAY DATA FLOW...

👍 ... makes applications more predictable and easier to understand

👍 ... makes applications easier to debug, as we have more control over the data

👍 ... is more performant

*Angular has **two-way** data flow*