

# Context and Navigation

---



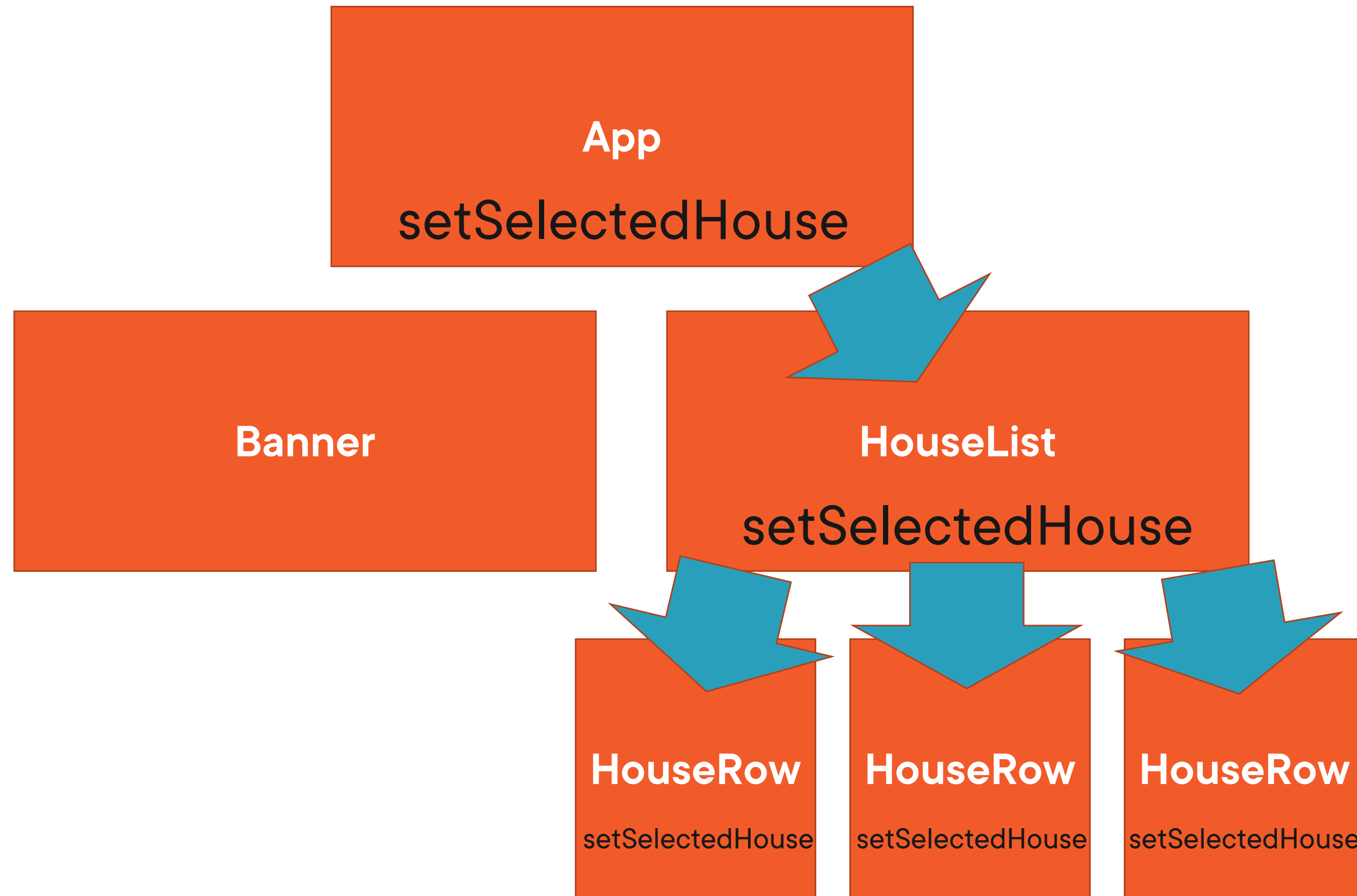
**Roland Guijt**

Freelance consultant and trainer

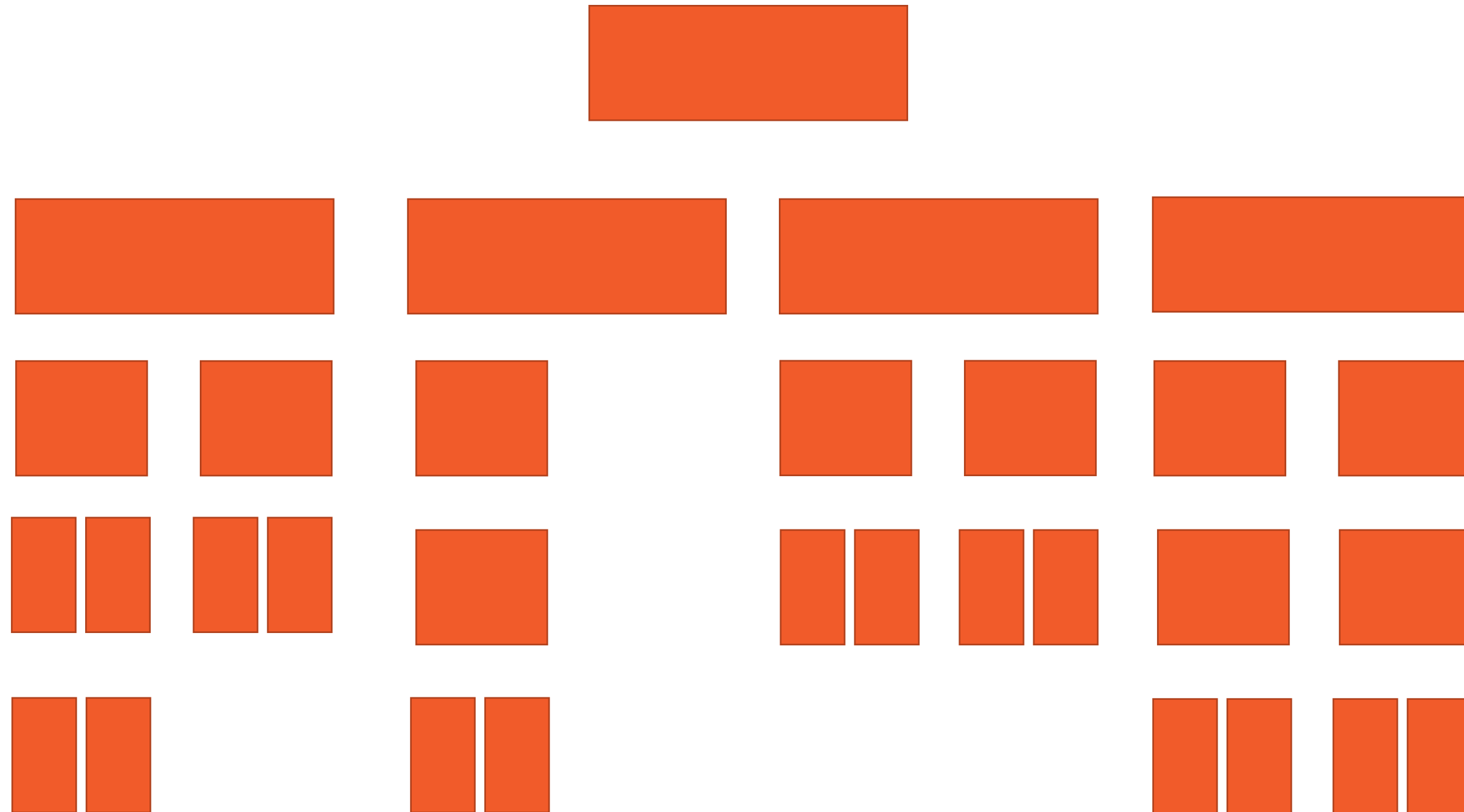
@rolandguijt roland.guijt@gmail.com



# Navigation So Far



# A Possible Future Component Hierarchy



# Context Recipe

In component providing context or separate module

```
const context = React.createContext("default value");
```

In the component providing context

```
<context.Provider value="some value">  
  //children  
</context.Provider>
```

In child components

```
const value = useContext(context);
```

```
<context.Consumer>  
  { value => /* render something based on value */ }  
</context.Consumer>
```



When the value of the context changes,  
all consumers will be re-rendered



What If More  
Features Are  
Needed?

**Build it yourself**

**next.js router**

**React Router**



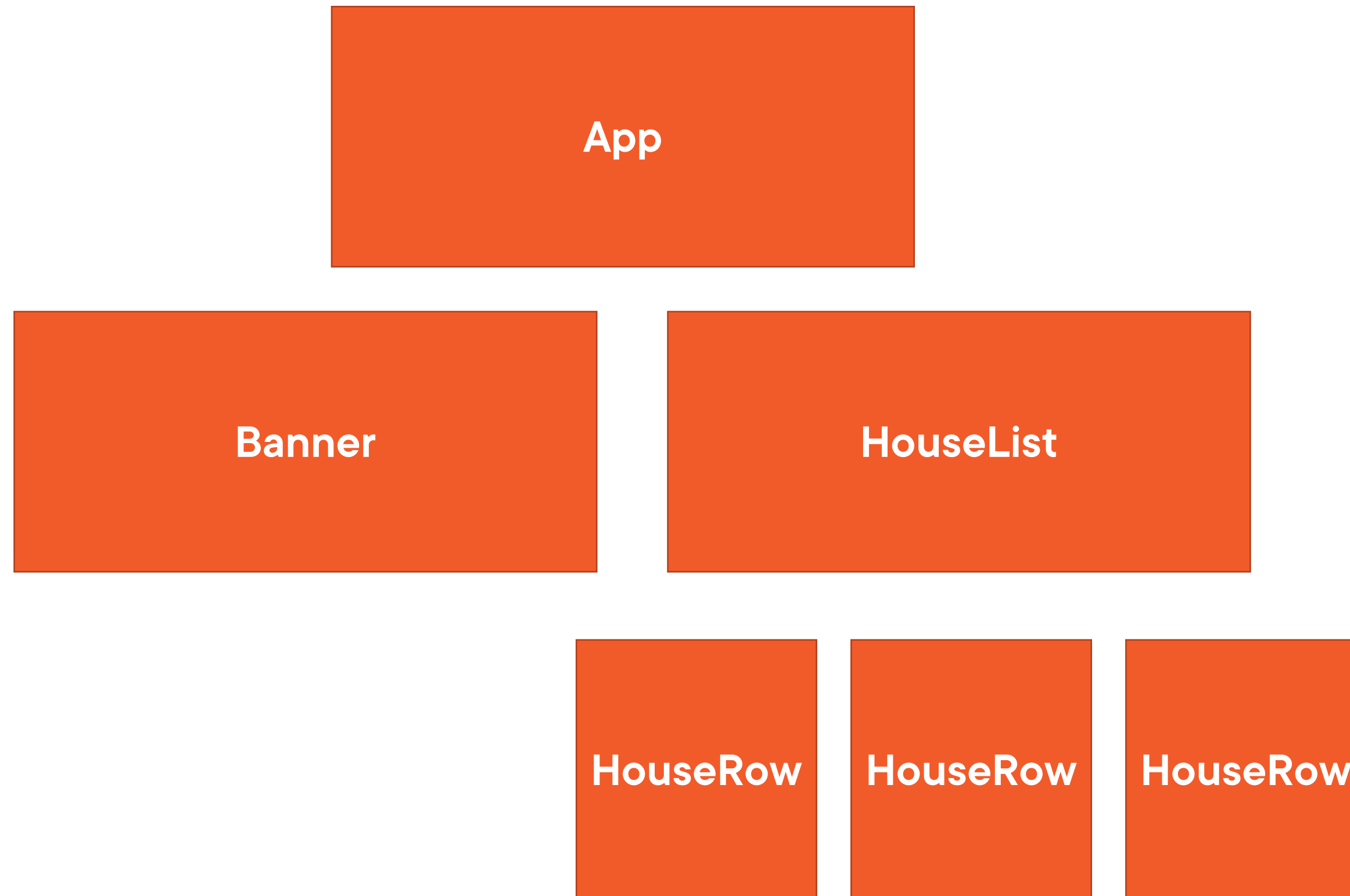
Consider

**Extra library dependency**

**It is all based on global state**

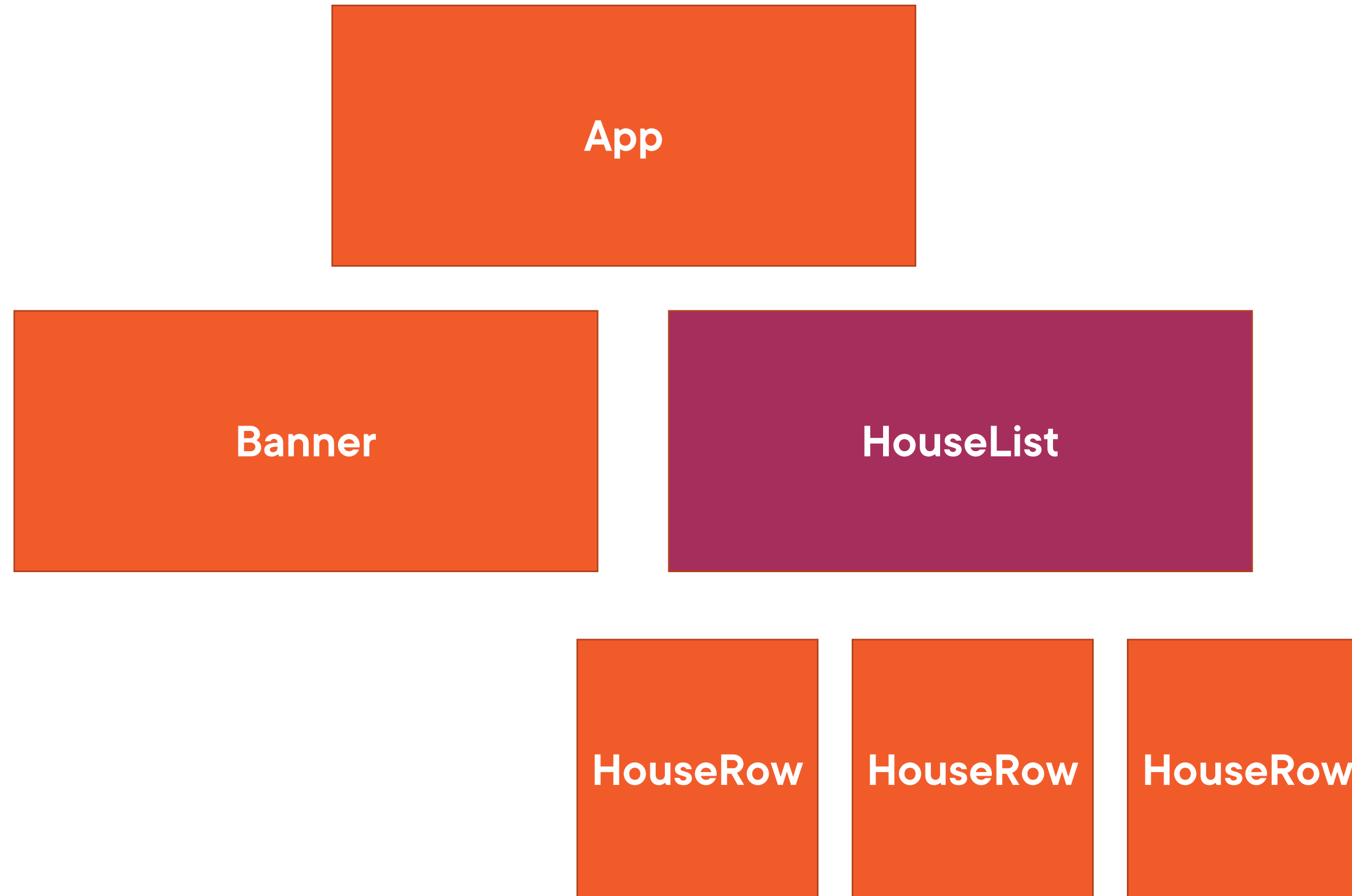


# Context On a Lower Component Level

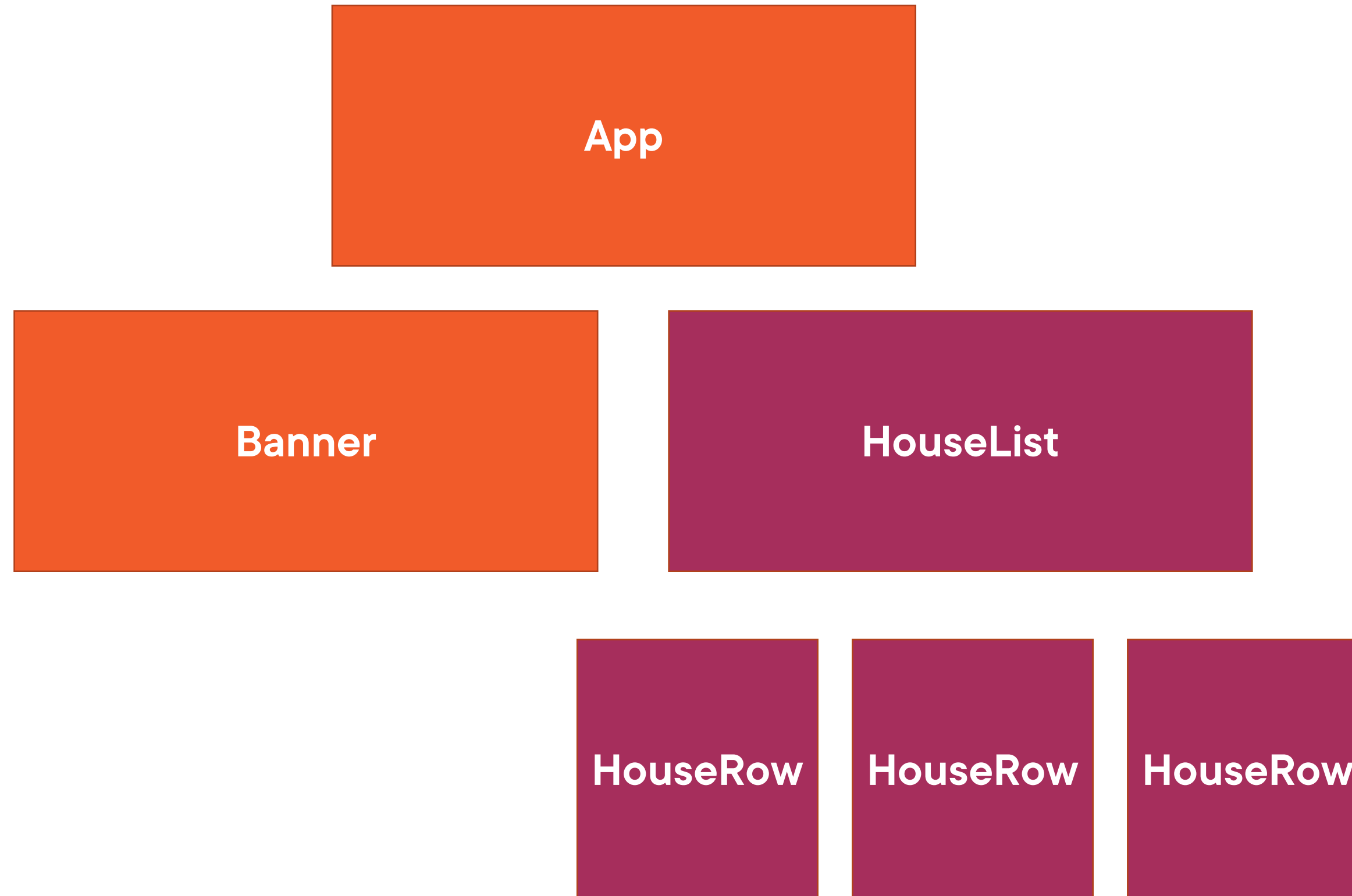




# Context On a Lower Component Level



# Context On a Lower Component Level



# When to Use Context and Implications

**When the same state has to be passed to many components**

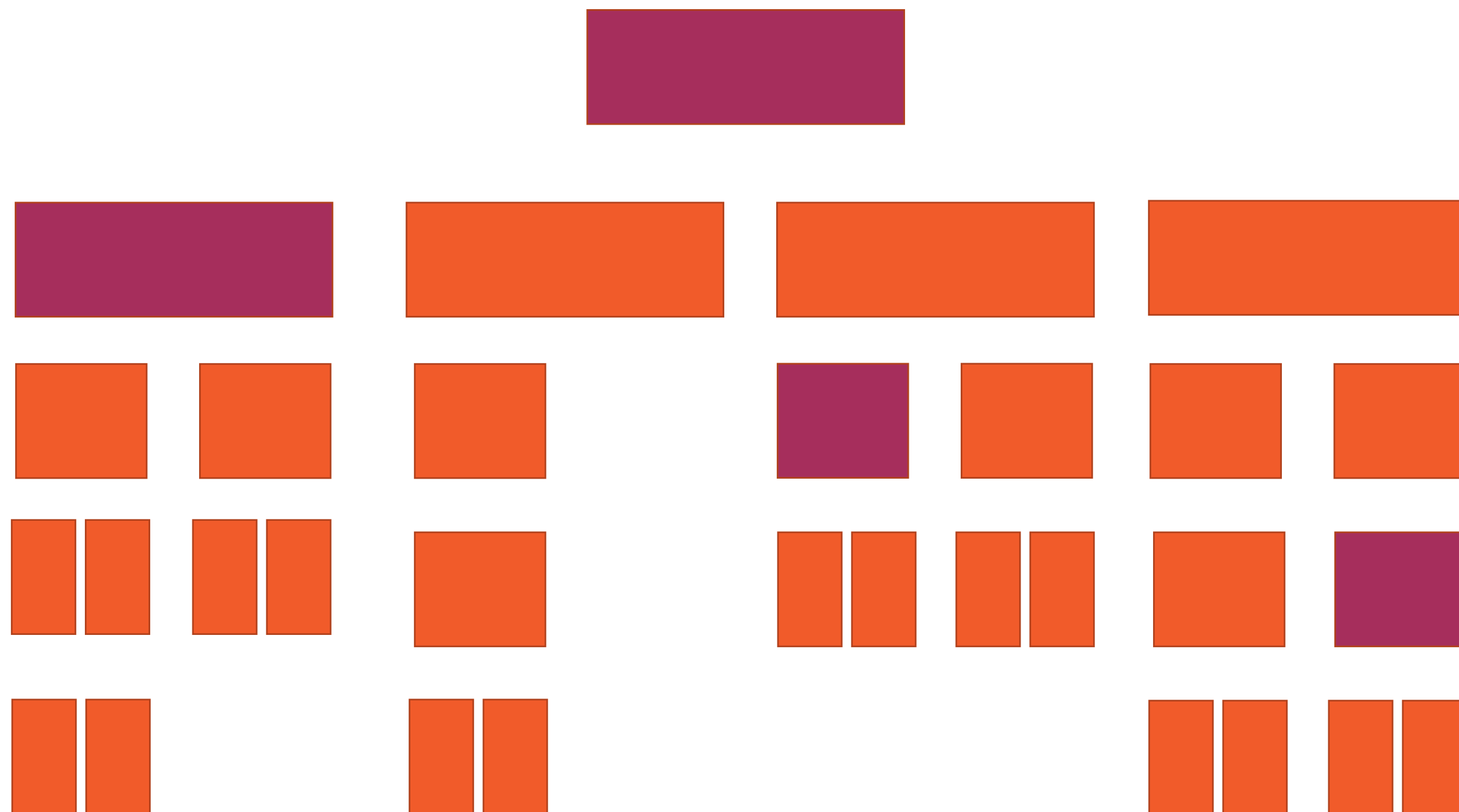
**Think about re-renders**

**Component reuse more difficult**

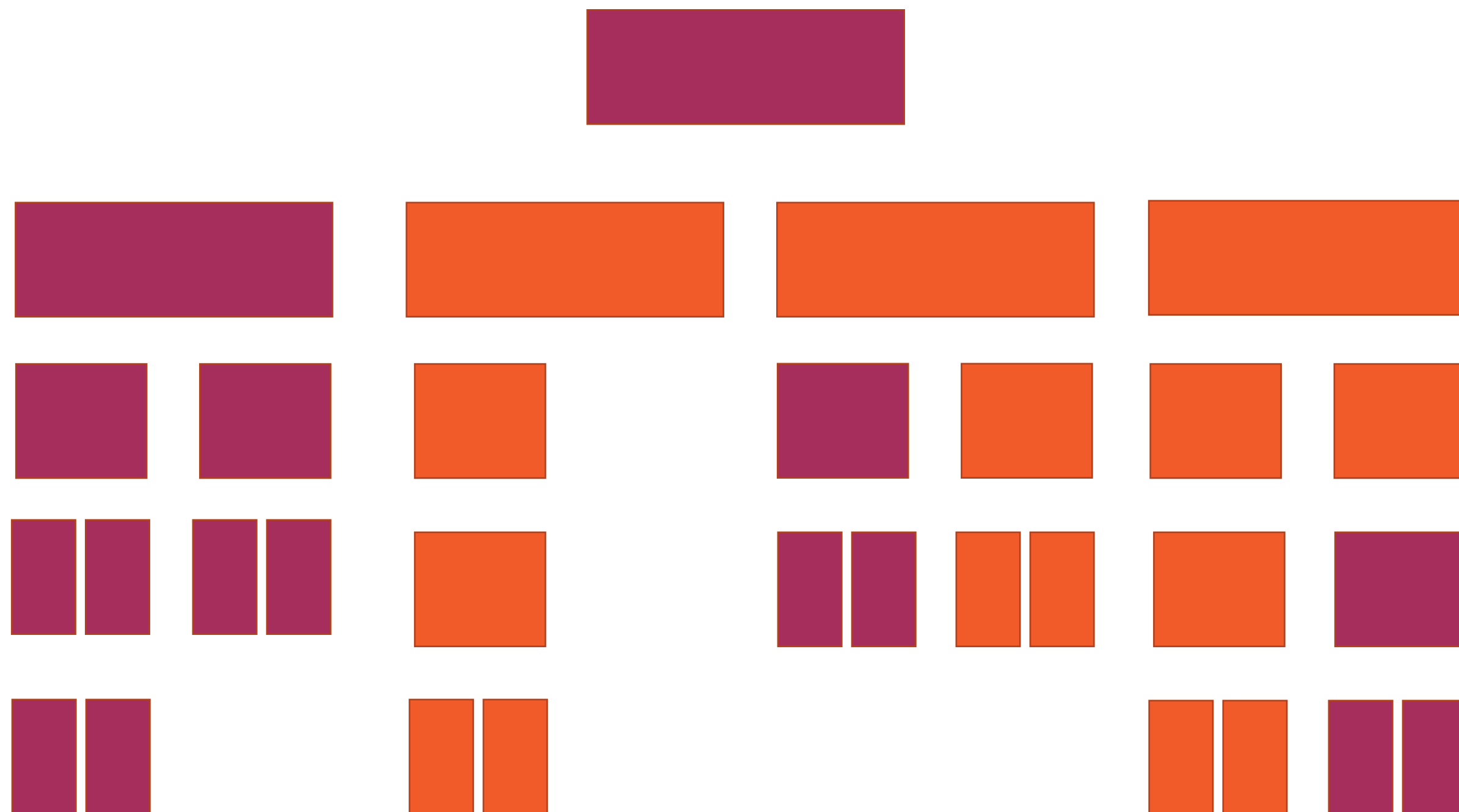
**Hidden state**



# Context and Re-Renders



# Context and Re-Renders



Next up:  
User Input and Forms

