

The Built-in React Hooks

`useRef`, `useContext`, `useMemo` and `useCallback`



Peter Kellner

Developer, Consultant and Author

ReactAtScale.com @pkellner linkedin.com/in/peterkellner99

The `useRef` Hooks Has Two main Uses in React

Access DOM nodes and react elements

Store local variables associated with function components that persist between renders



The `useRef` Hook Has Two Main Uses in React

Access DOM nodes and react elements

Store local variables associated with function components that persist between renders



The `useRef` Hooks Has Two main Uses in React

Access DOM nodes and react elements

Store local variables associated with function components that persist between renders



The useContext Built in React Hook





What is React Context?

Used to share data between components

Has a JavaScript object assigned to it

All components at our below in hierarchy have
access to that JavaScript object assigned





What is React Context?

Used to share data between components

Has a JavaScript object assigned to it

All components at our below in hierarchy have access to that JavaScript object assigned

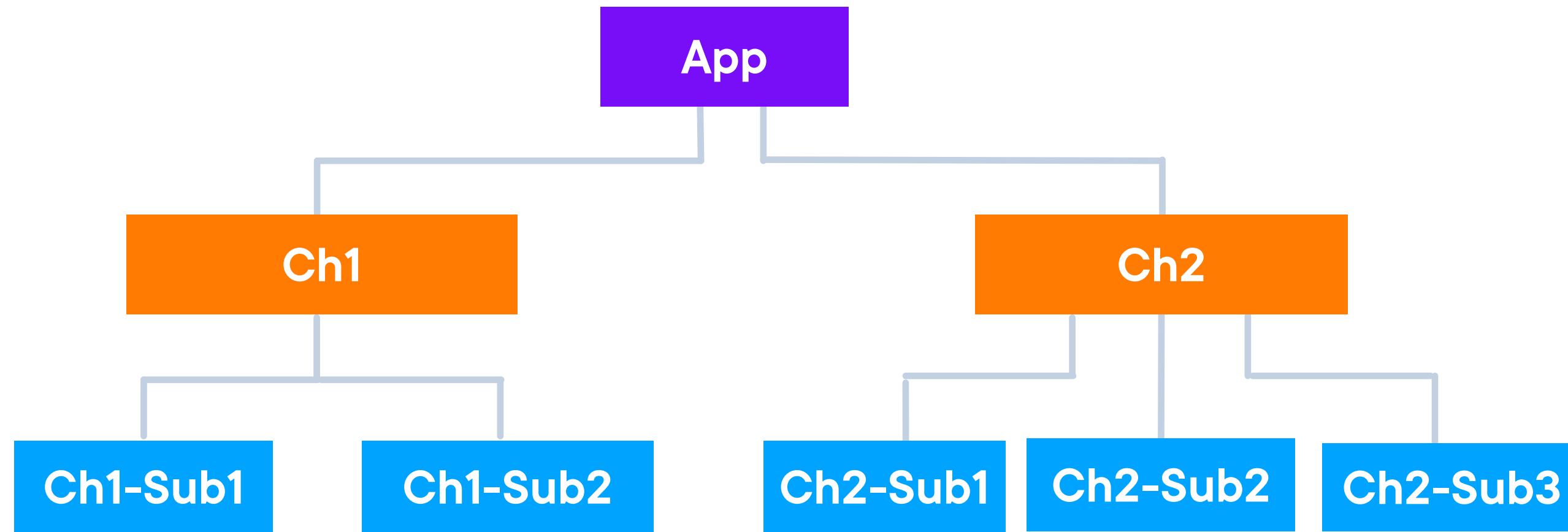


Inversion of Control

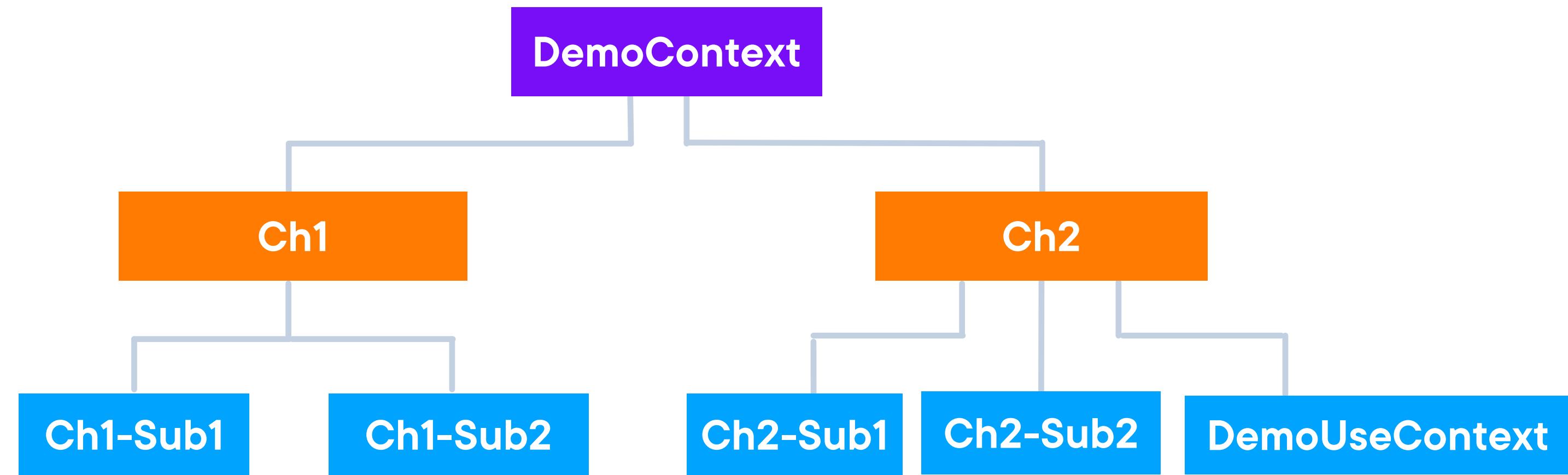
Inversion of control is used to increase modularity of the program and make it extensible.



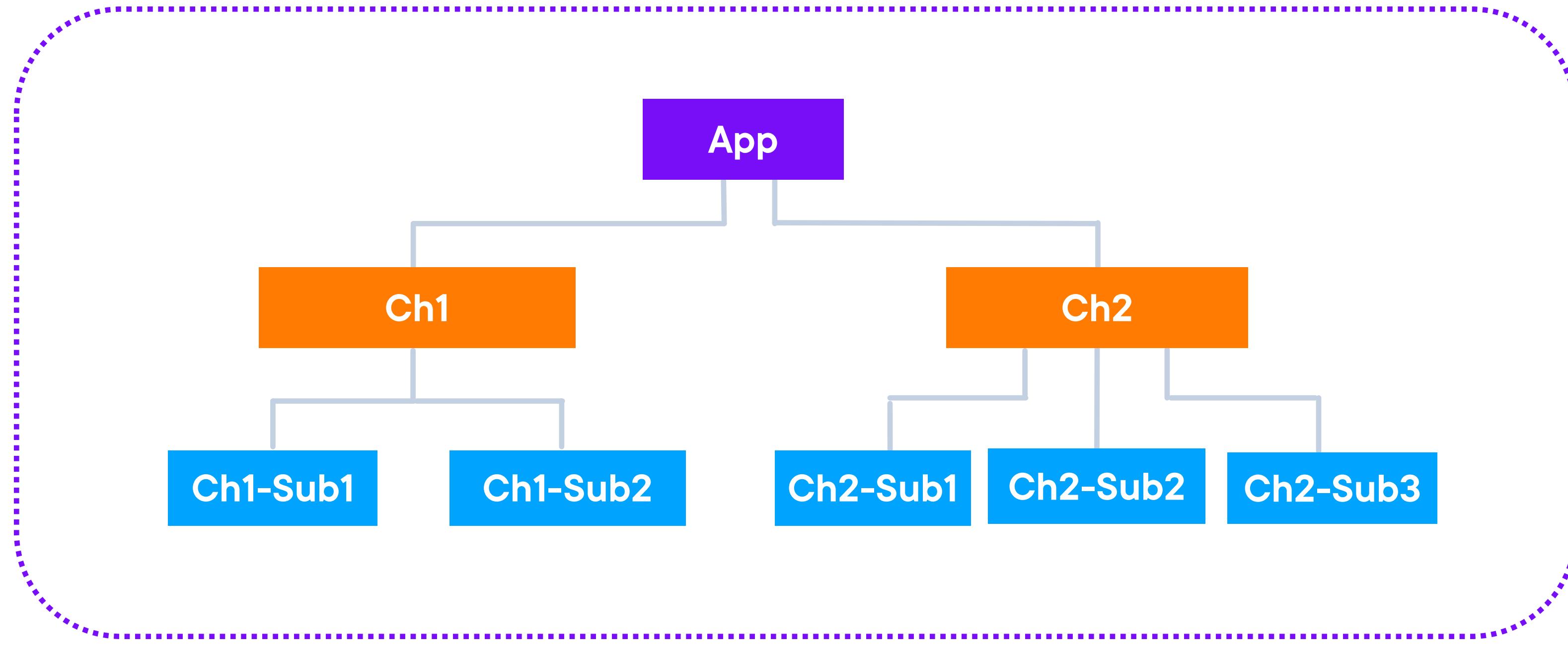
Component Hierarchy In a React App



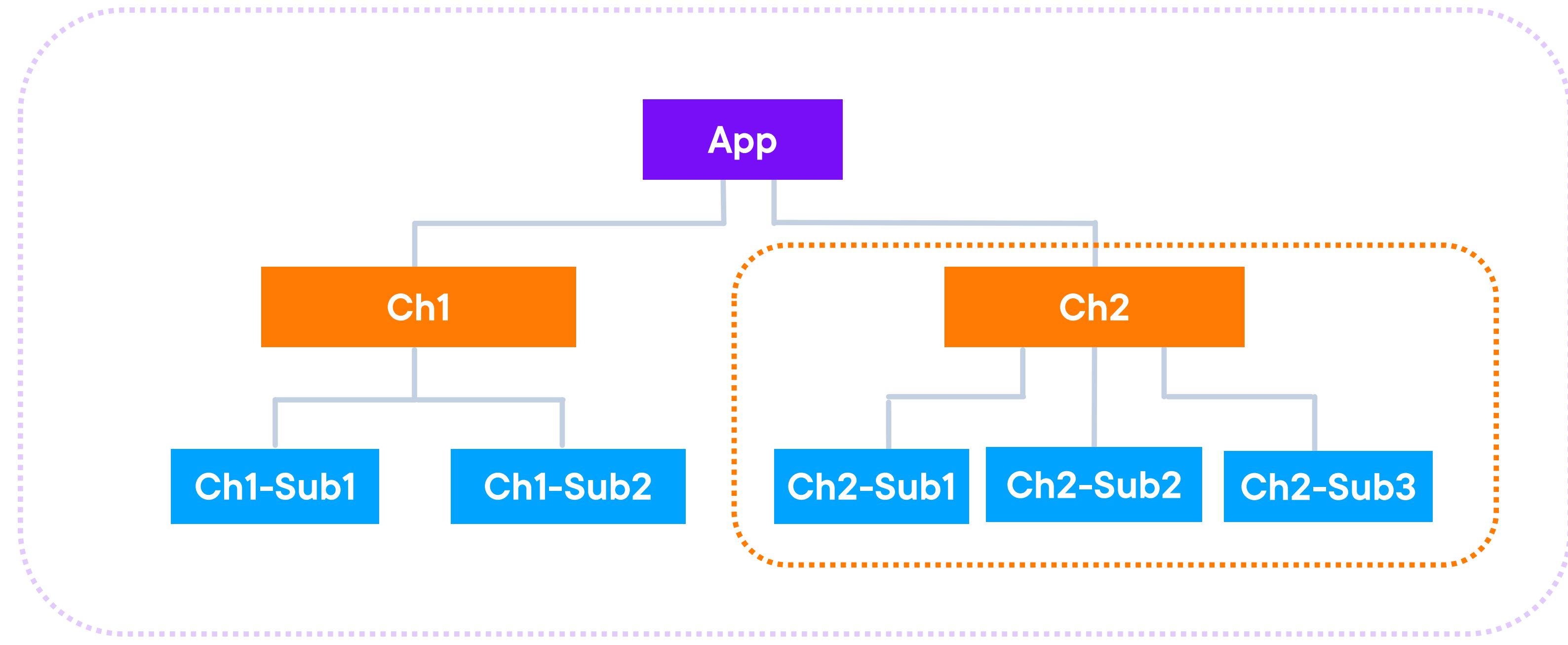
Component Hierarchy In a React App



Component Hierarchy In a React App



Component Hierarchy In a React App



/pages/demo-context

```
import { createContext, useState }  
  from "react";  
import DemoUseContext from  
  "../../src/components/misc/demo-usecontext";  
  
export const ThemeContext =  
  createContext();  
  
function Ch2() {  
  return <DemoUseContext />  
}  
  
export default function DemoContext() {  
  const [theme, setTheme] = useState("dark");  
  return (  
    <ThemeContext.Provider  
      value={{ theme, setTheme }}>  
      <Ch2><DemoUseContext /></Ch2>  
    </ThemeContext.Provider>  
  );  
}
```

..../src/..../demo-usecontext.js

```
import { useContext } from "react";  
import { ThemeContext }  
  from "../../src/components/misc/demo-usecontext";  
  
export default function DemoUseContext() {  
  const { theme, setTheme } =  
    useContext(ThemeContext);  
  return (  
    <div className="container">  
      <h1>The Theme is: {theme}</h1><hr />  
      <button  
        onClick={() => {  
          setTheme(theme === "dark" ?  
            "light" : "dark");  
        }}>Toggle Theme  
      </button>  
    </div>  
  );  
}
```

Takeaways

React team has included 15 useful hooks

Most often used hooks

- useState and useReducer
- useEffect
- useRef
- useContext

Other hooks

- Great docs at <https://reactjs.org/docs>



Up Next:

Using Context and Custom React Hooks for a Redux Like Experience

