

Global App State With useContext and useReducer Like Redux



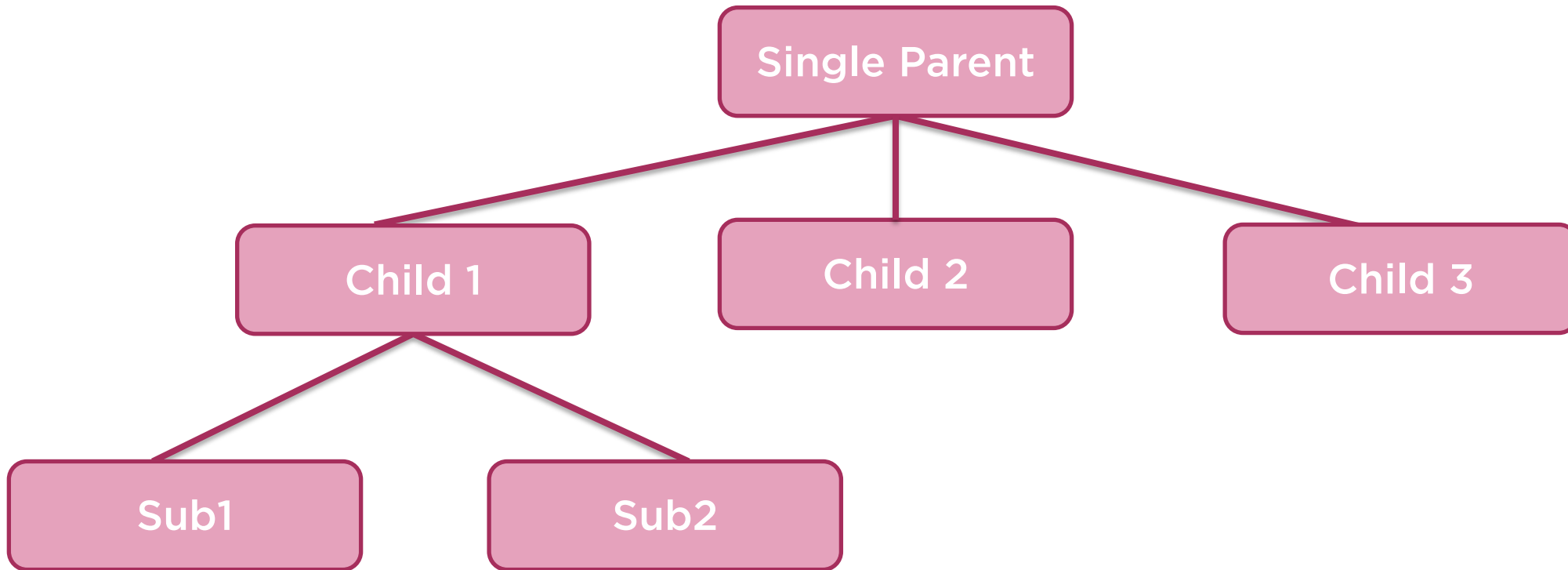
Peter Kellner

DEVELOPER, CONSULTANT AND AUTHOR

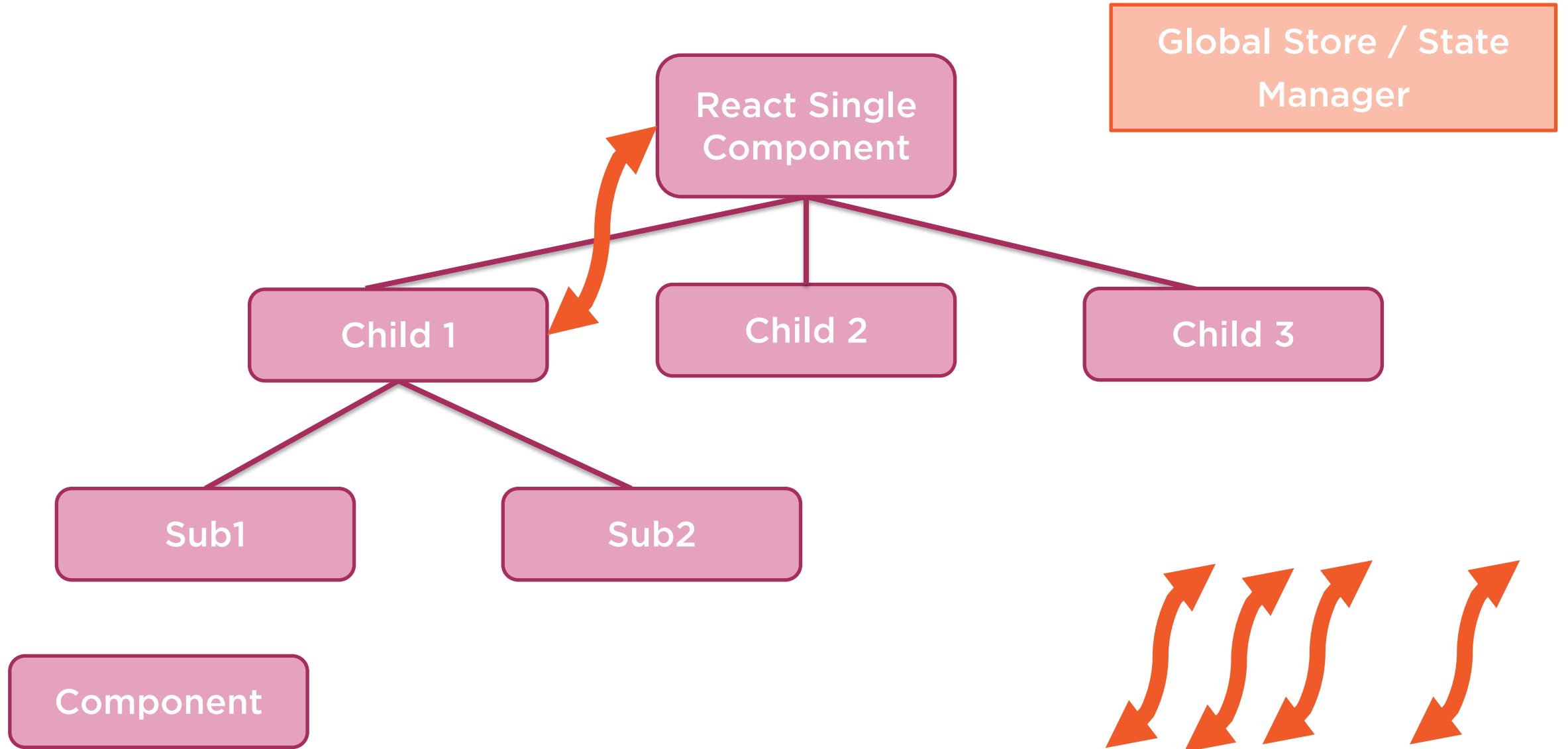
@pkellner [linkedin.com/in/peterkellner99](https://www.linkedin.com/in/peterkellner99) ReactAtScale.com

React apps are combined
of many components
working together

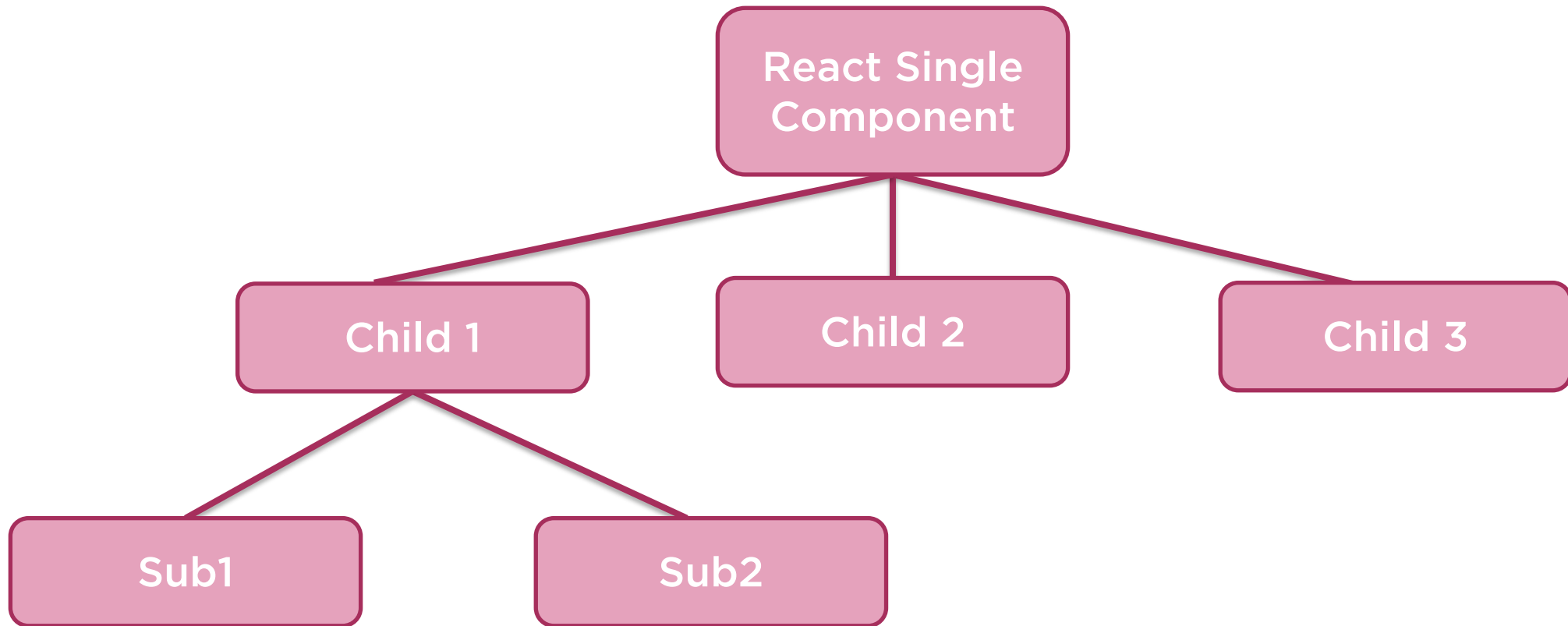
React Component Tree



React Component Tree



React Component Tree



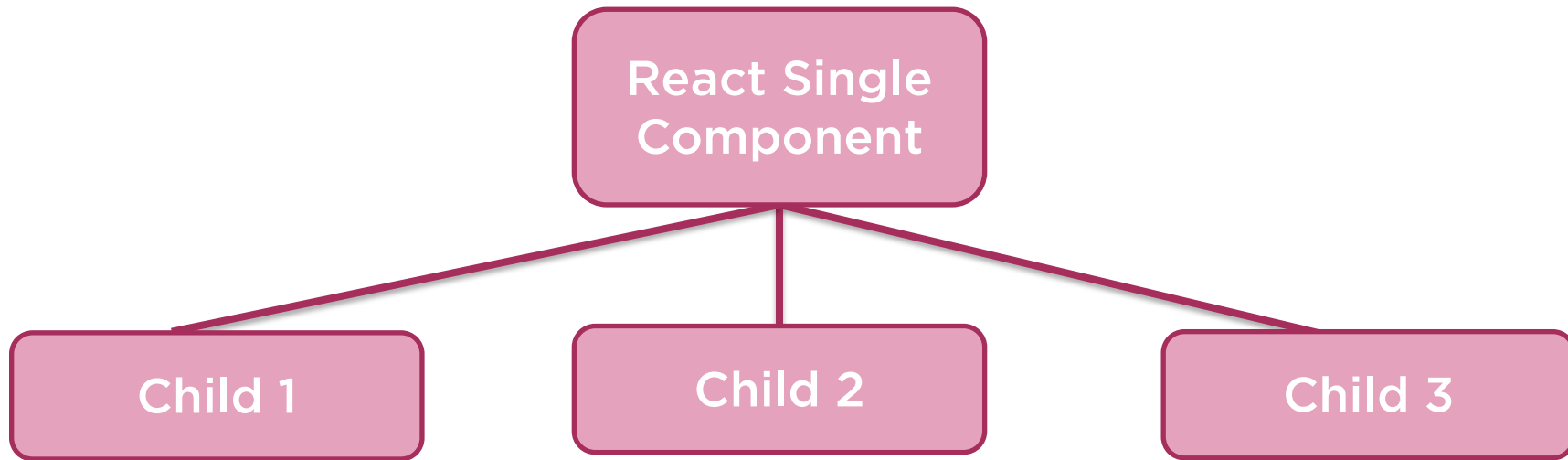
Component

React Component Tree

React Single
Component

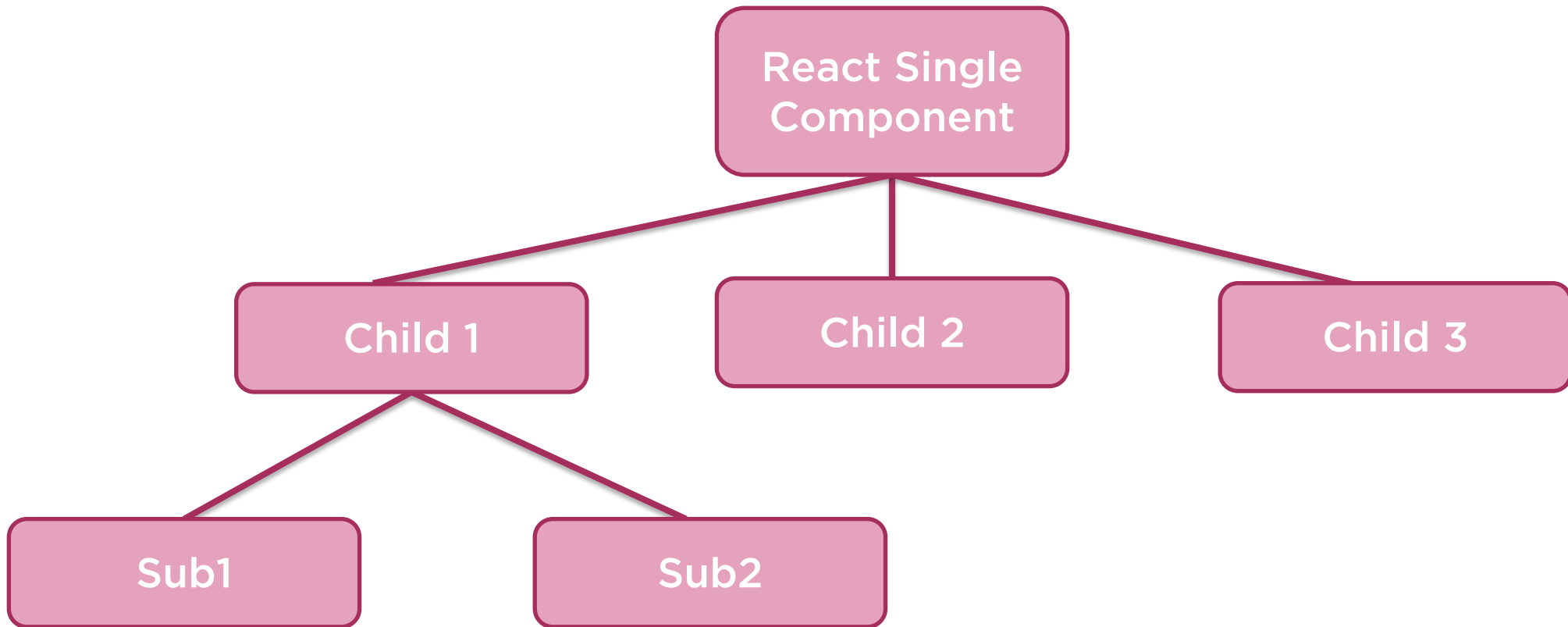
Component

React Component Tree



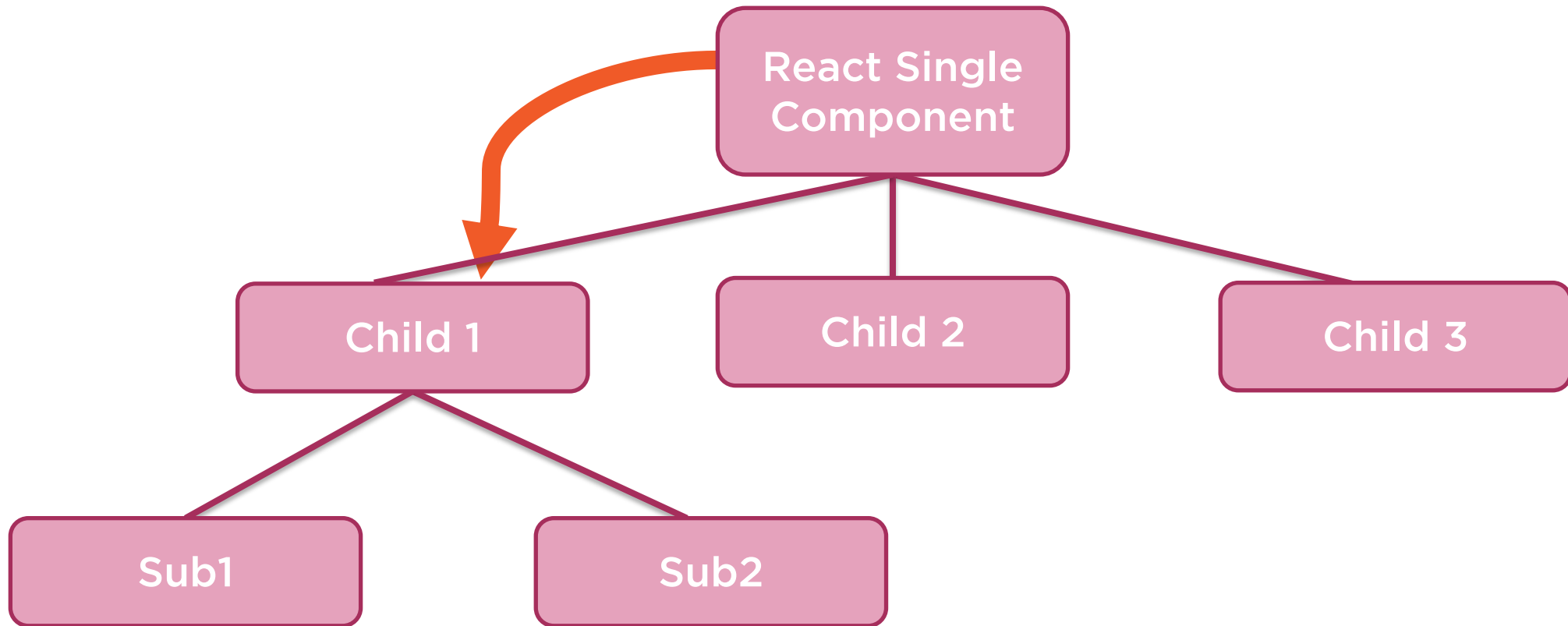
Component

React Component Tree



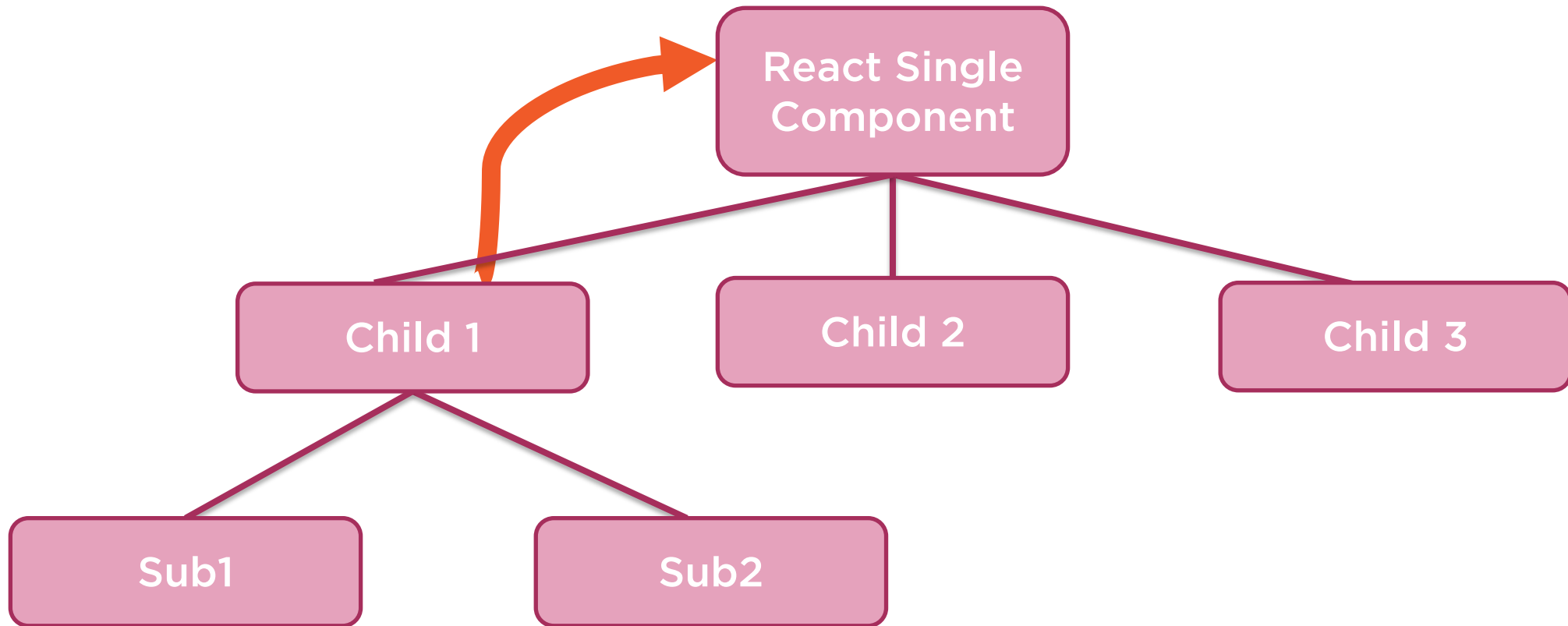
Component

React Component Tree



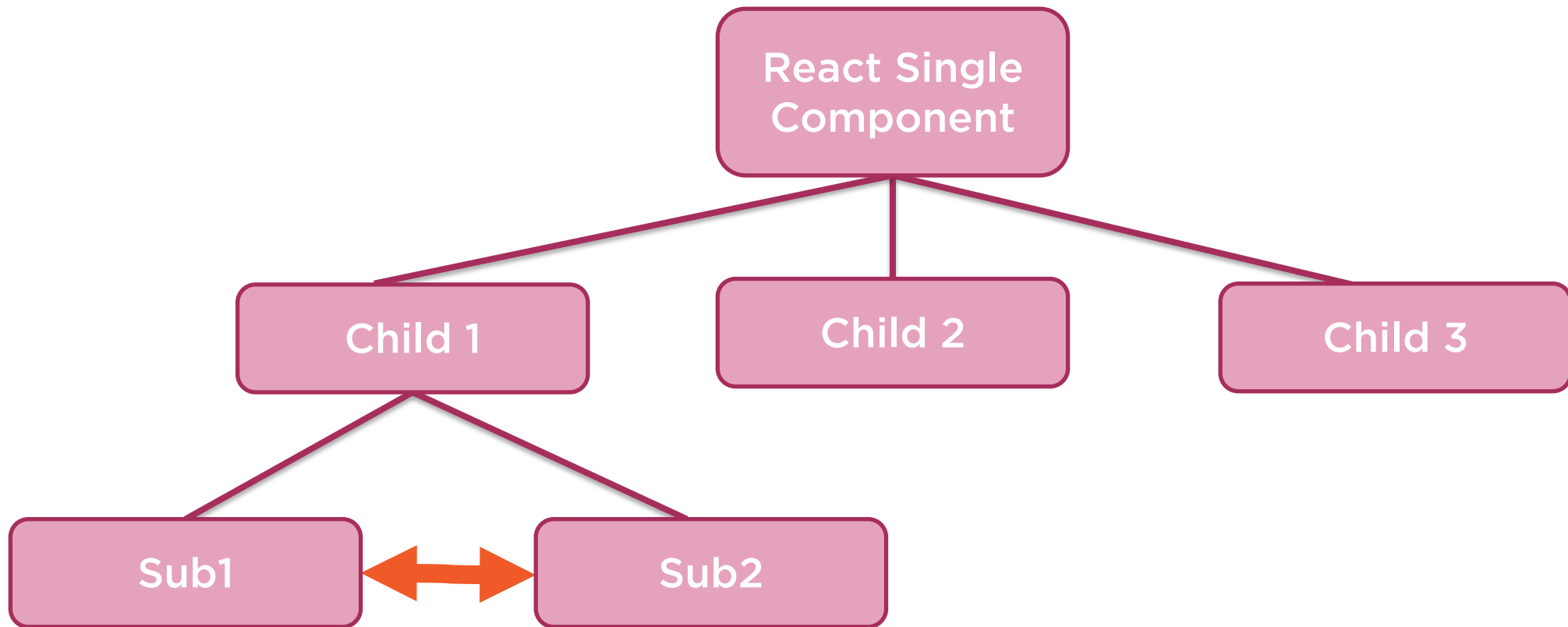
Component

React Component Tree



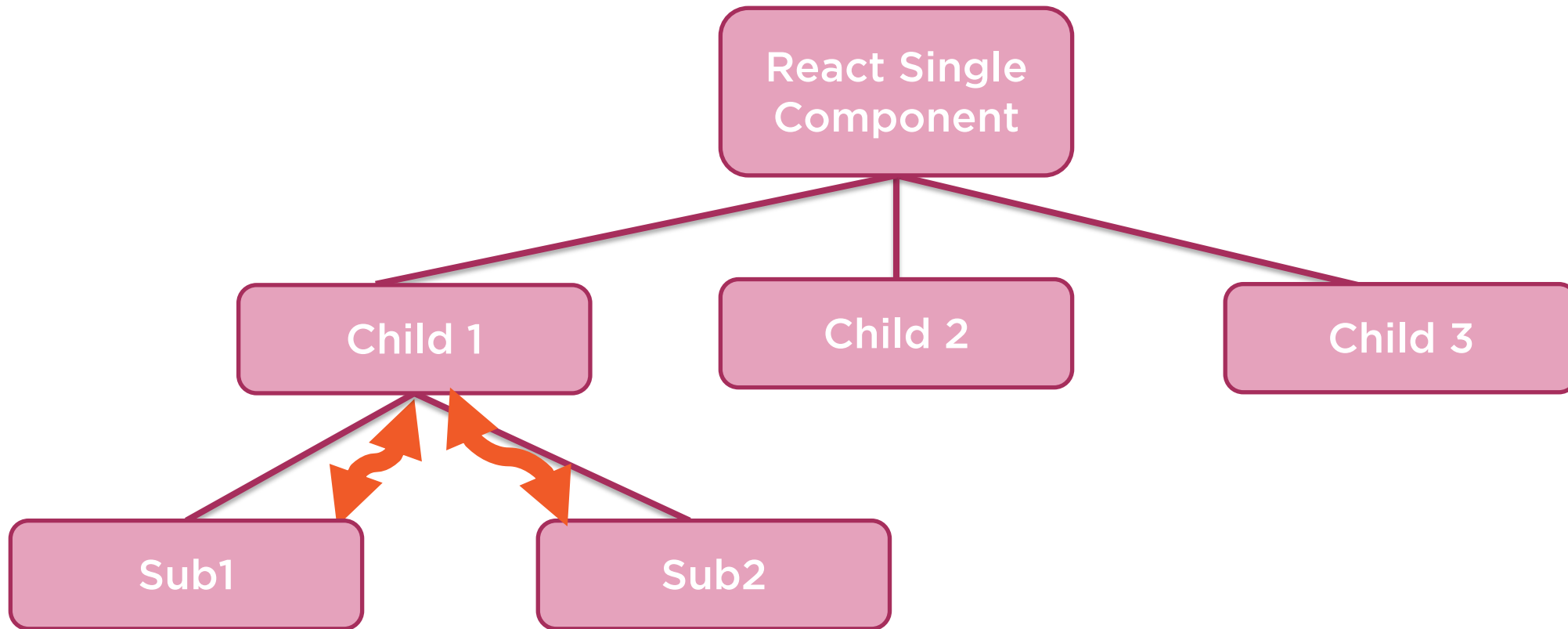
Component

React Component Tree



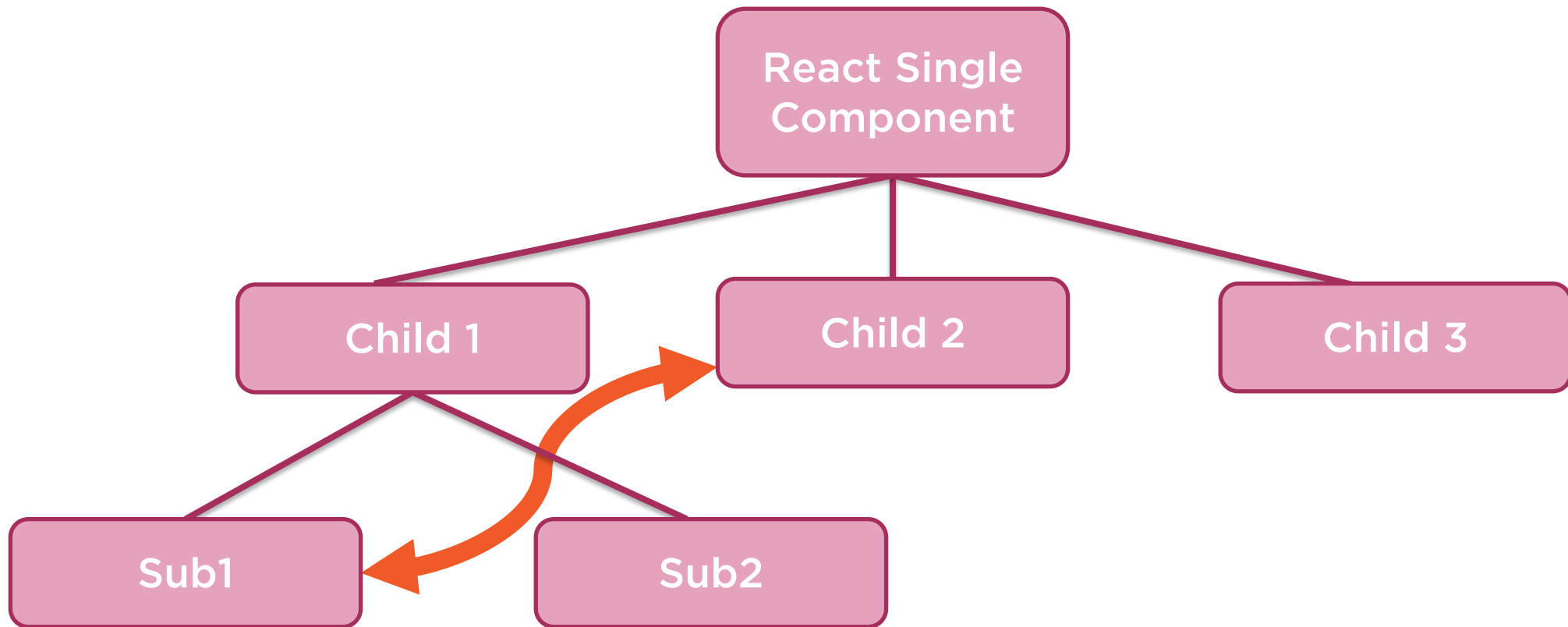
Component

React Component Tree



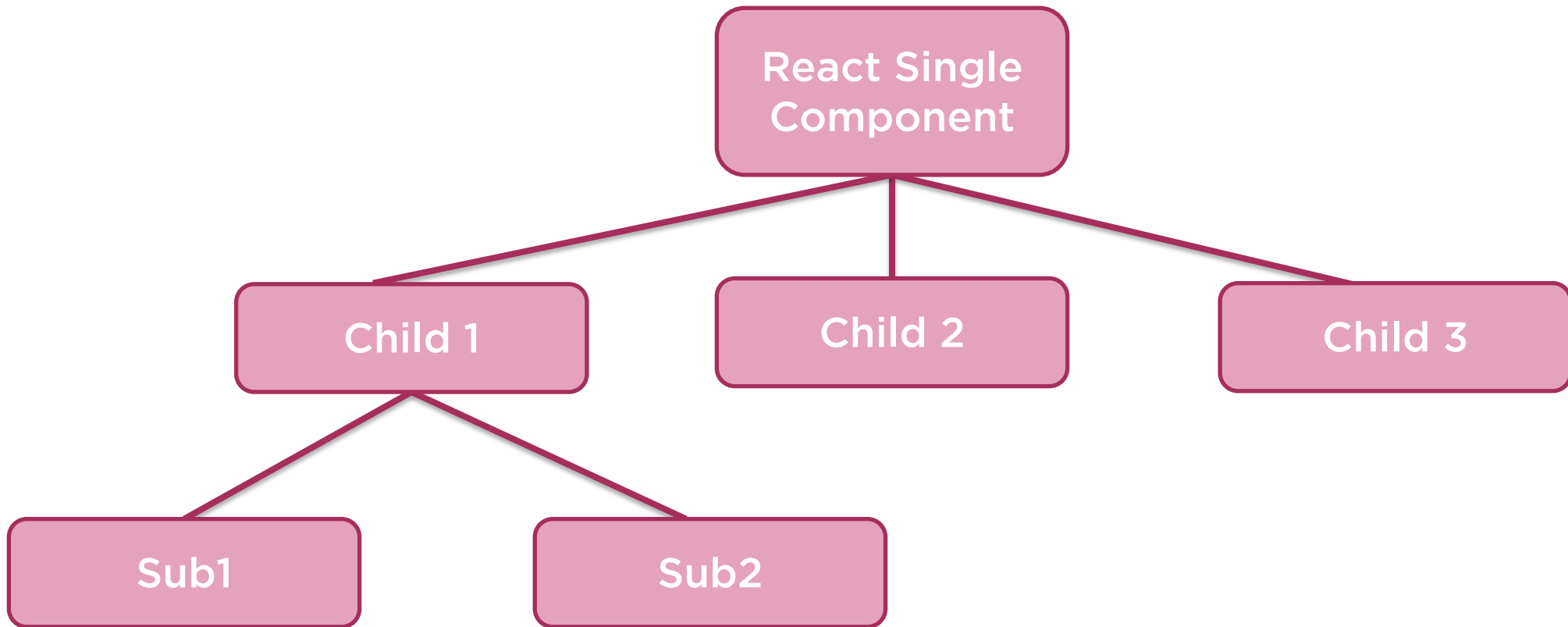
Component

React Component Tree



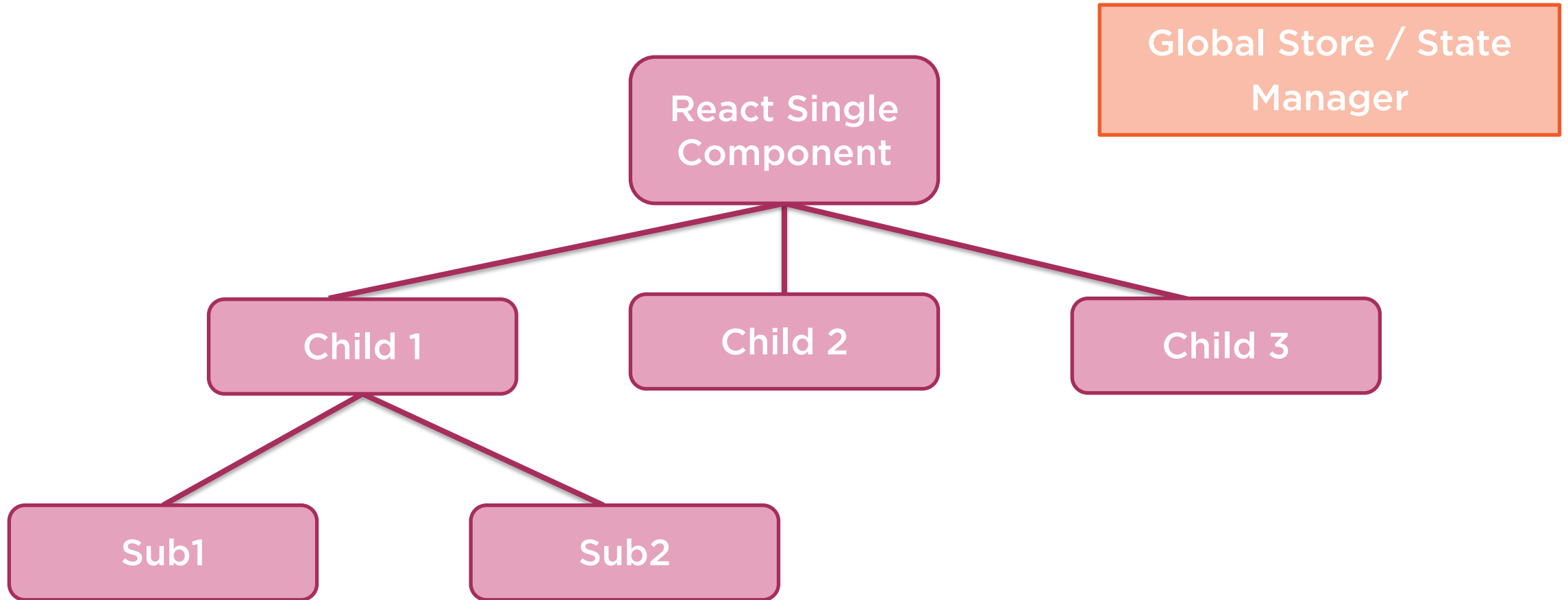
Component

React Component Tree



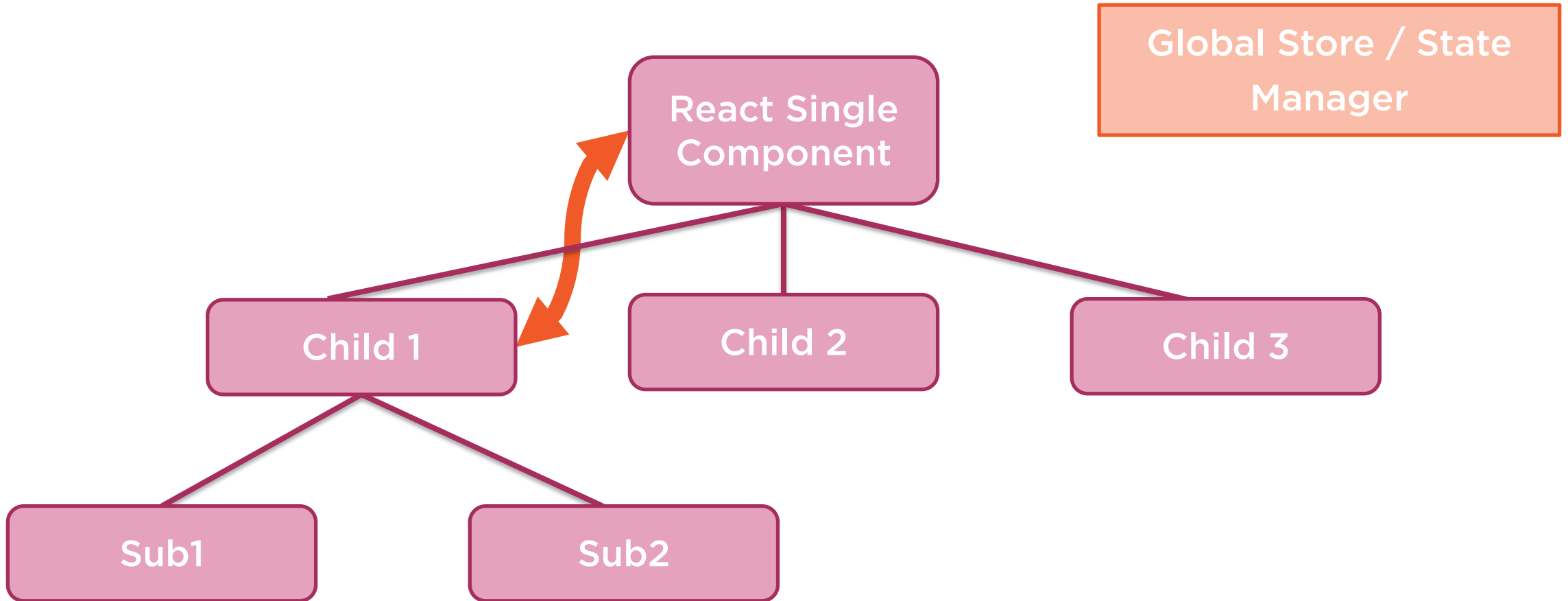
Component

React Component Tree



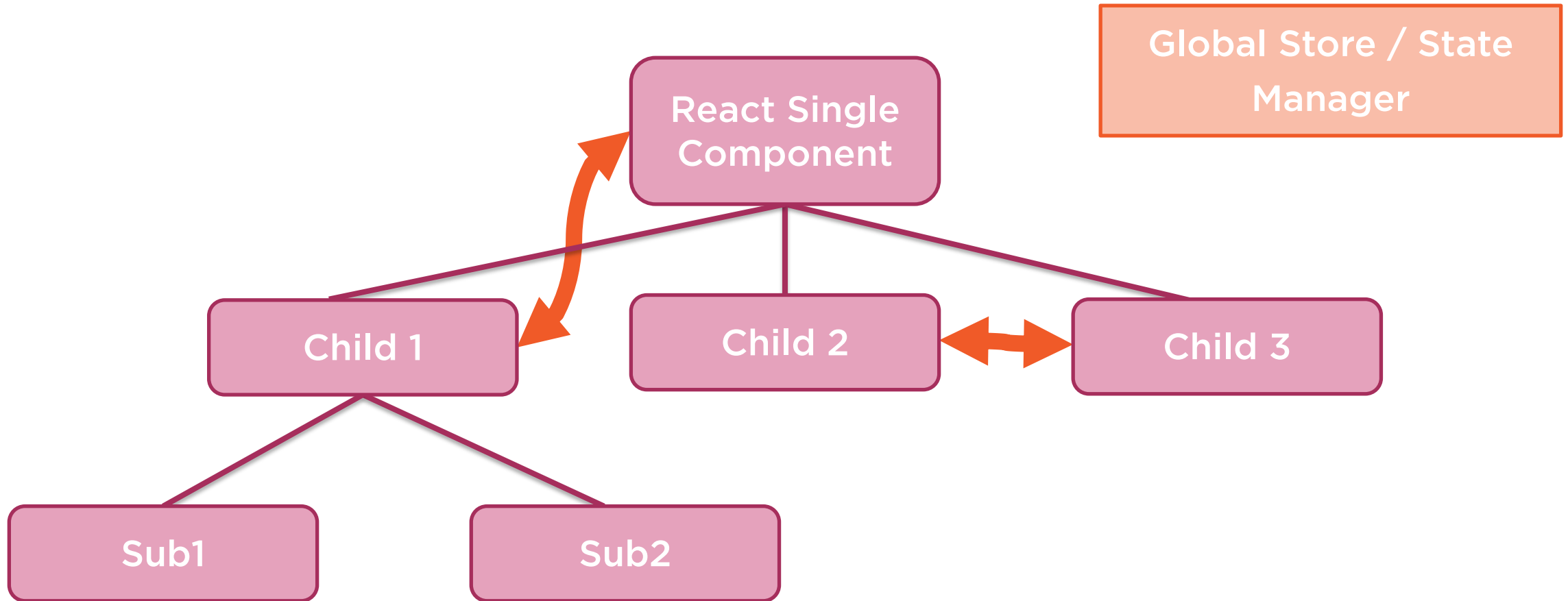
Component

React Component Tree



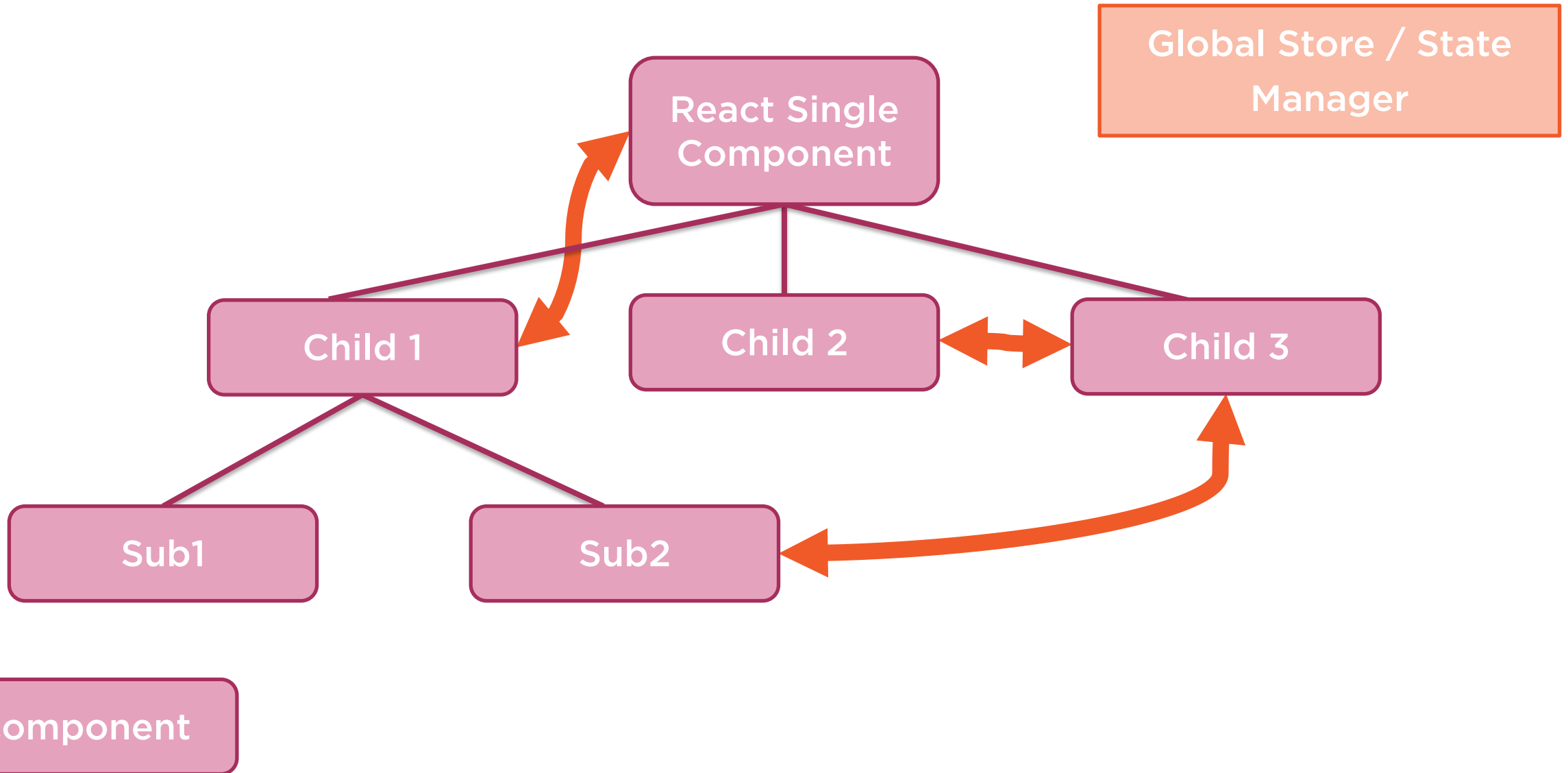
Component

React Component Tree

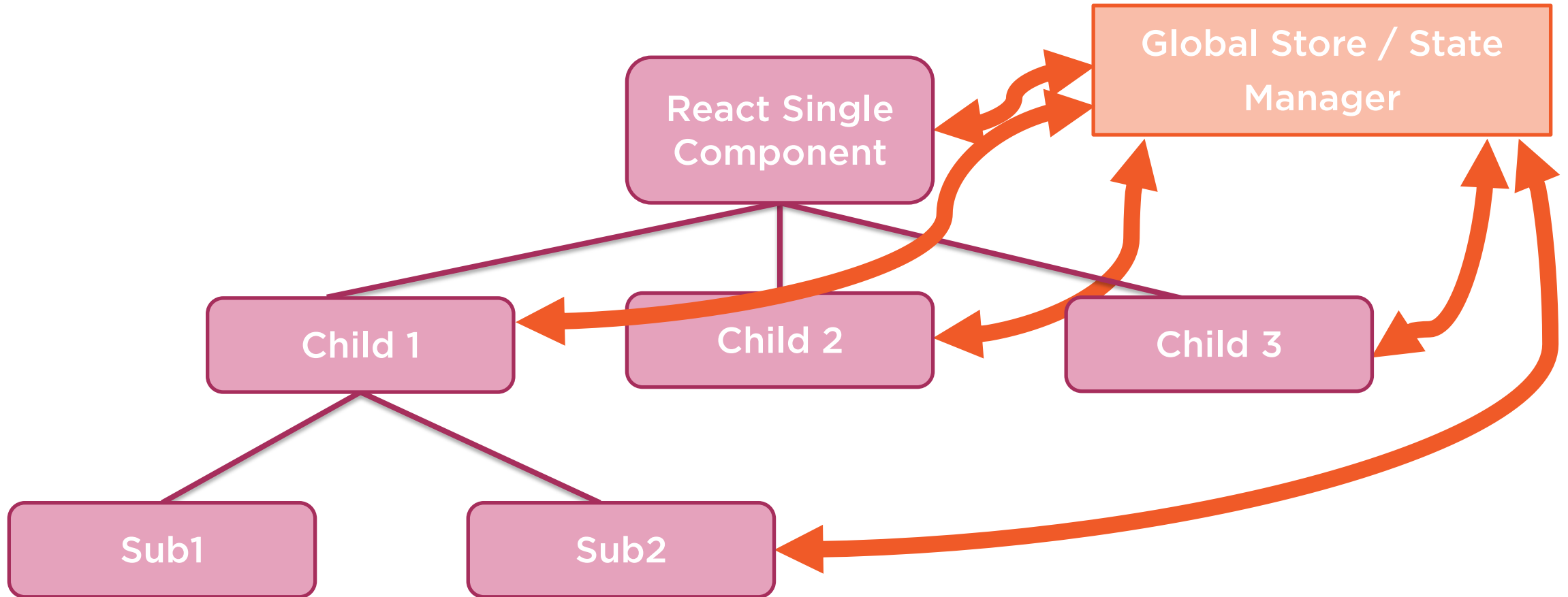


Component

React Component Tree

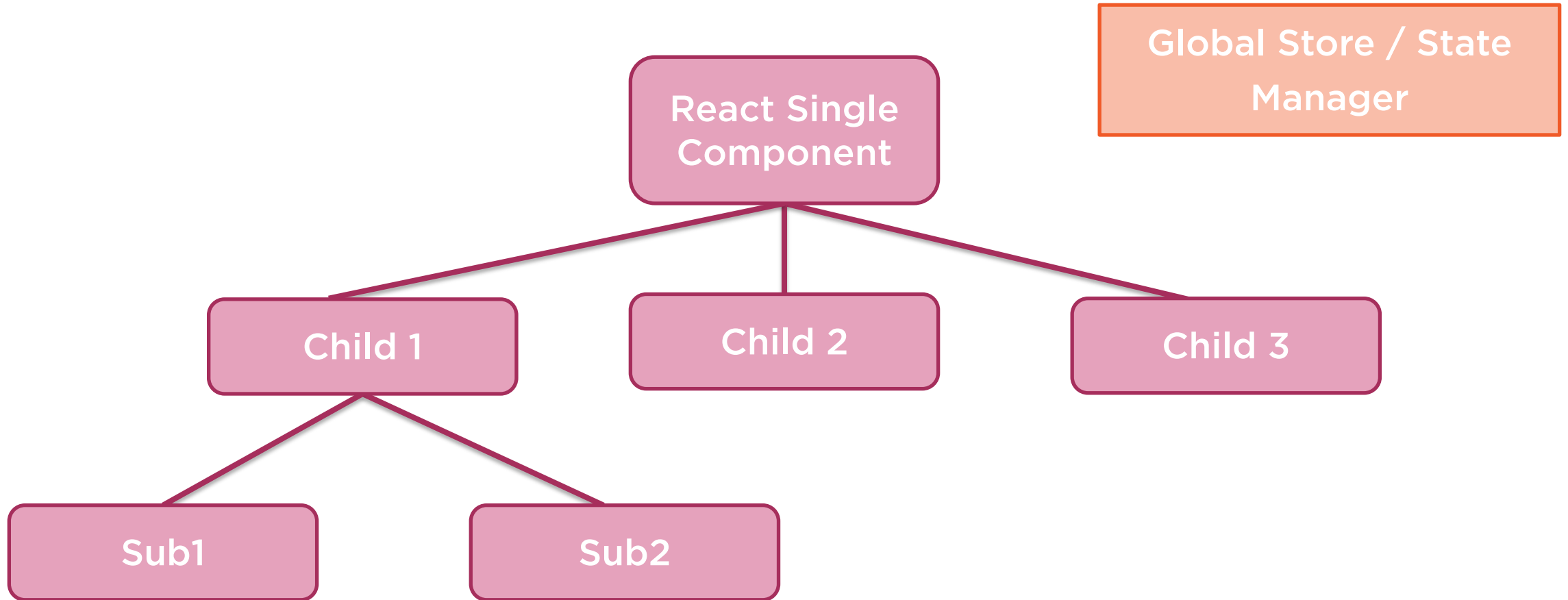


React Component Tree



Component

React Component Tree

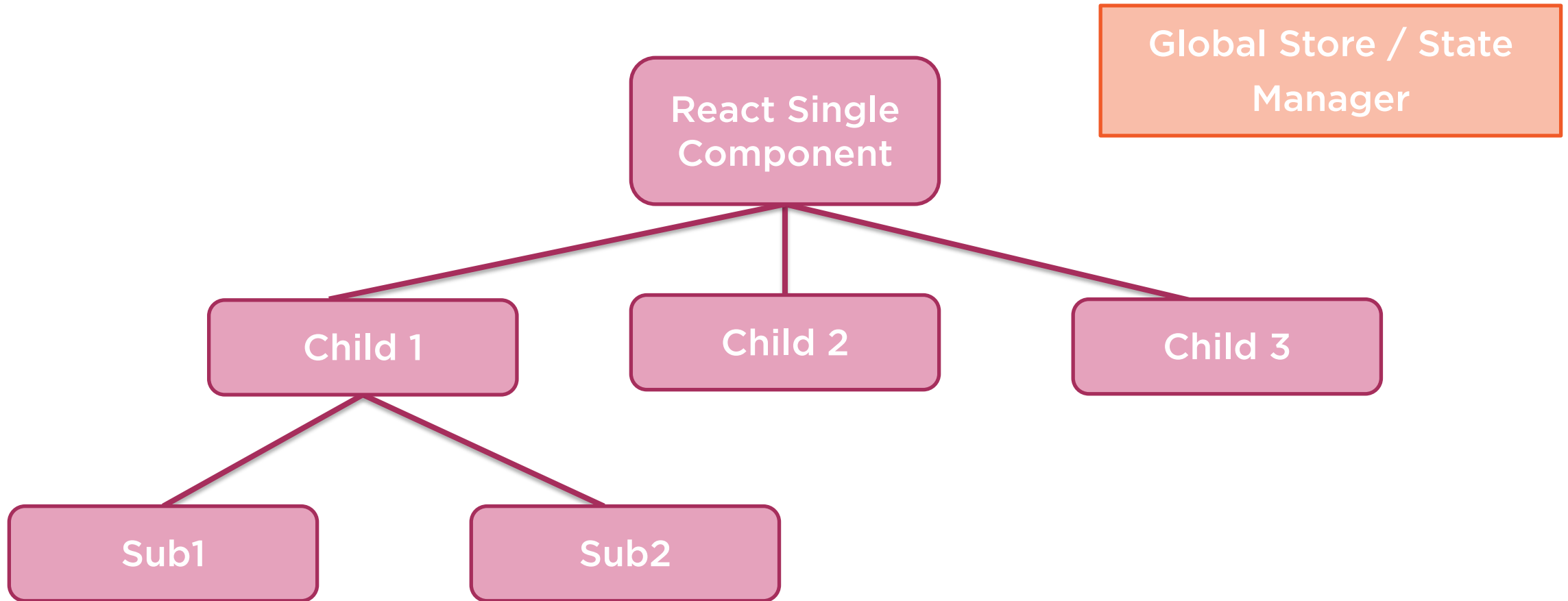


Component

Redux

Redux can be thought of as a central store that holds an entire applications state. Each of the app components can access the stored state without requiring passing data or functions from one component to another.

React Component Tree



Component

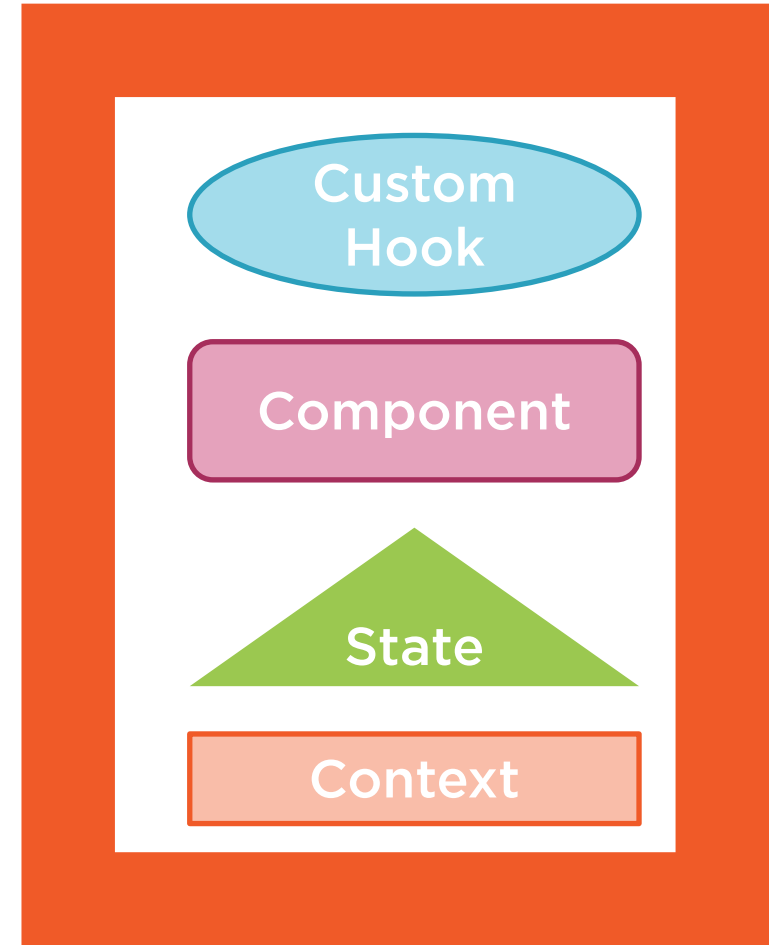
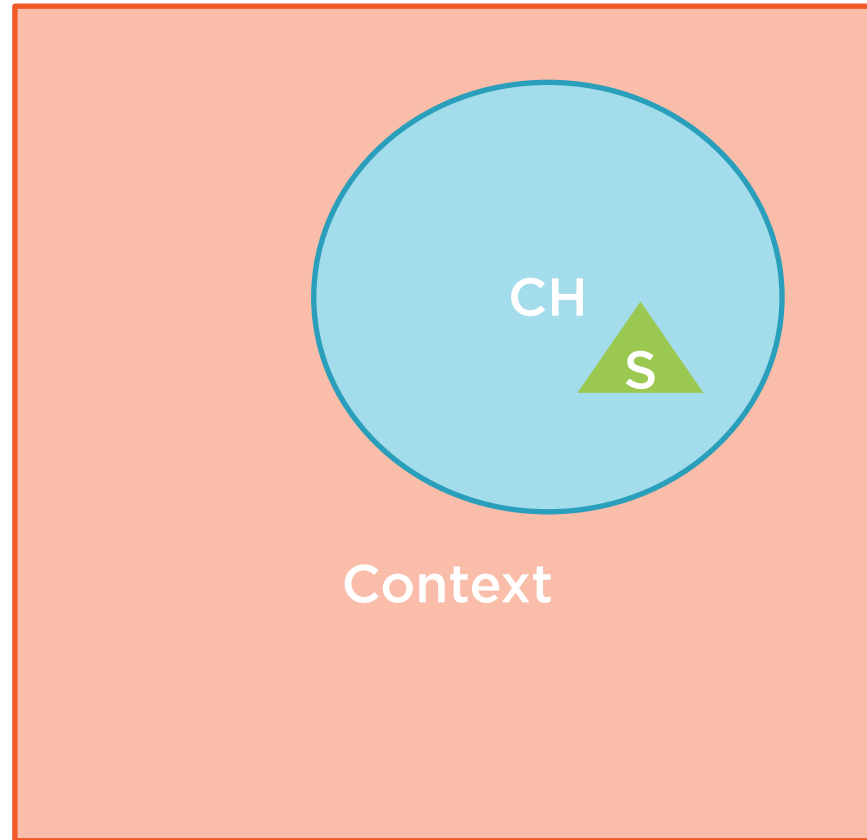
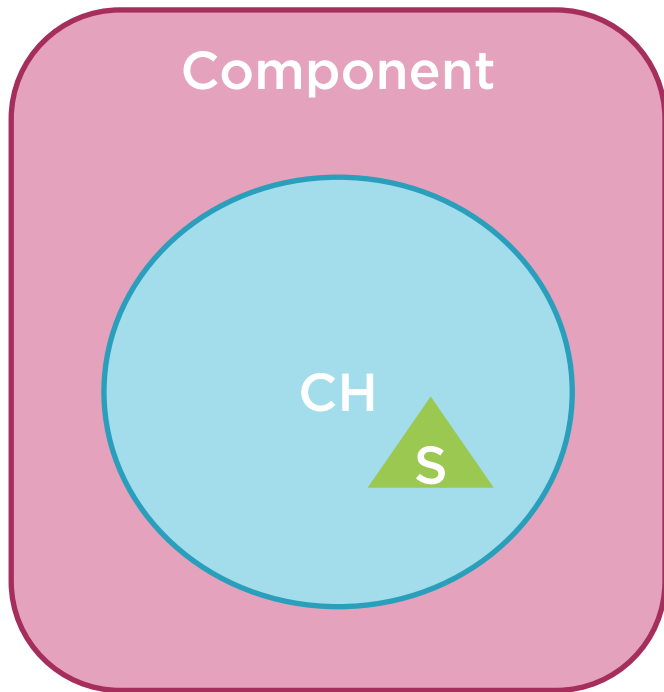
React Component Tree



The diagram consists of a large, light pink rounded rectangle with a dark pink border. Centered within this rectangle is the text "React Single Component" in a white, sans-serif font.

React Single Component

Components With Only Custom Hooks



LEGEND

Colorizing Bug Coding Plan

```
<GlobalContext>
```

```
  <App>
```

```
    <SpeakersList>
```

```
      <SpeakersImageComponent />
```

```
      <SpeakersImageComponent />
```

```
      <SpeakersImageComponent />
```

```
      <SpeakersImageComponent />
```

```
      <SpeakersImageComponent />
```

```
    </SpeakersList>
```

```
  </App>
```

```
</GlobalContext>
```

Colorizing Bug Coding Plan

Speakers List Component

**Calls `forceImageRerender()`
when Saturday or Sunday
checkbox changes**

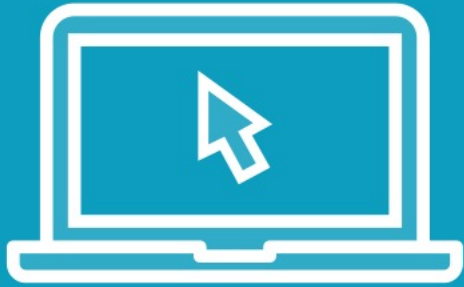
Speaker Image Component

**Renders image component
when `imageRerenderIdentifier`
changes value**

Speakers Data Manager Context

**Returns `imageRerenderIdentifier`
and `forceImageRerender()`**

Demo



Make code changes to fix our colorizing bug

Takeaways and Course Wrap



Whirlwind run through of auth

useContext enabler for auth

Top level component shares auth

React Hooks brings JavaScript functions
to life