

Sharing State and Functions via Context



Cory House

REACT CONSULTANT AND TRAINER

@housecor reactjsconsulting.com



Agenda



Why context?

Use context to share our cart

- Wrap provider in a custom component
- Share context via a custom hook



Do I need context?



Plain JS

React

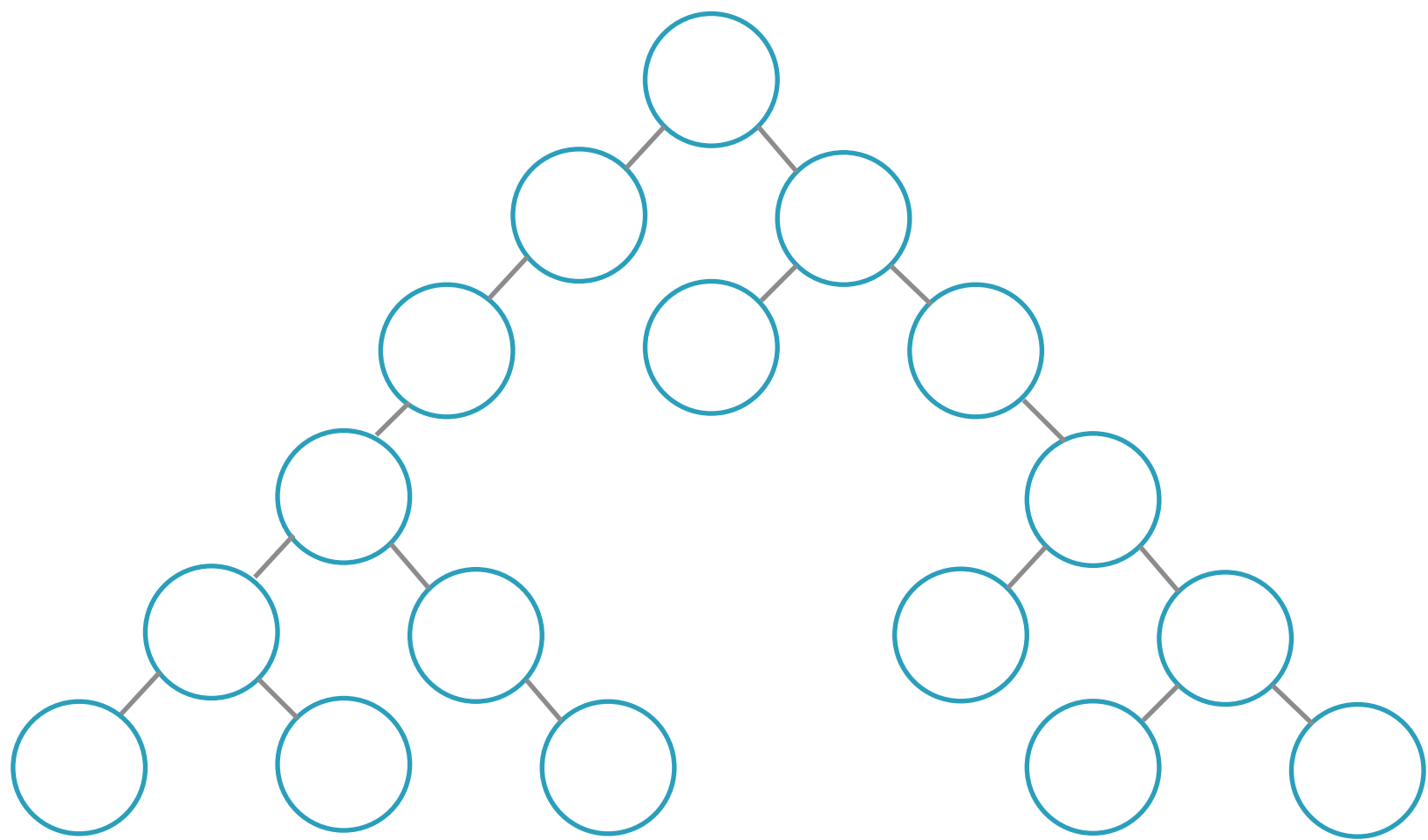
React
context

React with
Redux

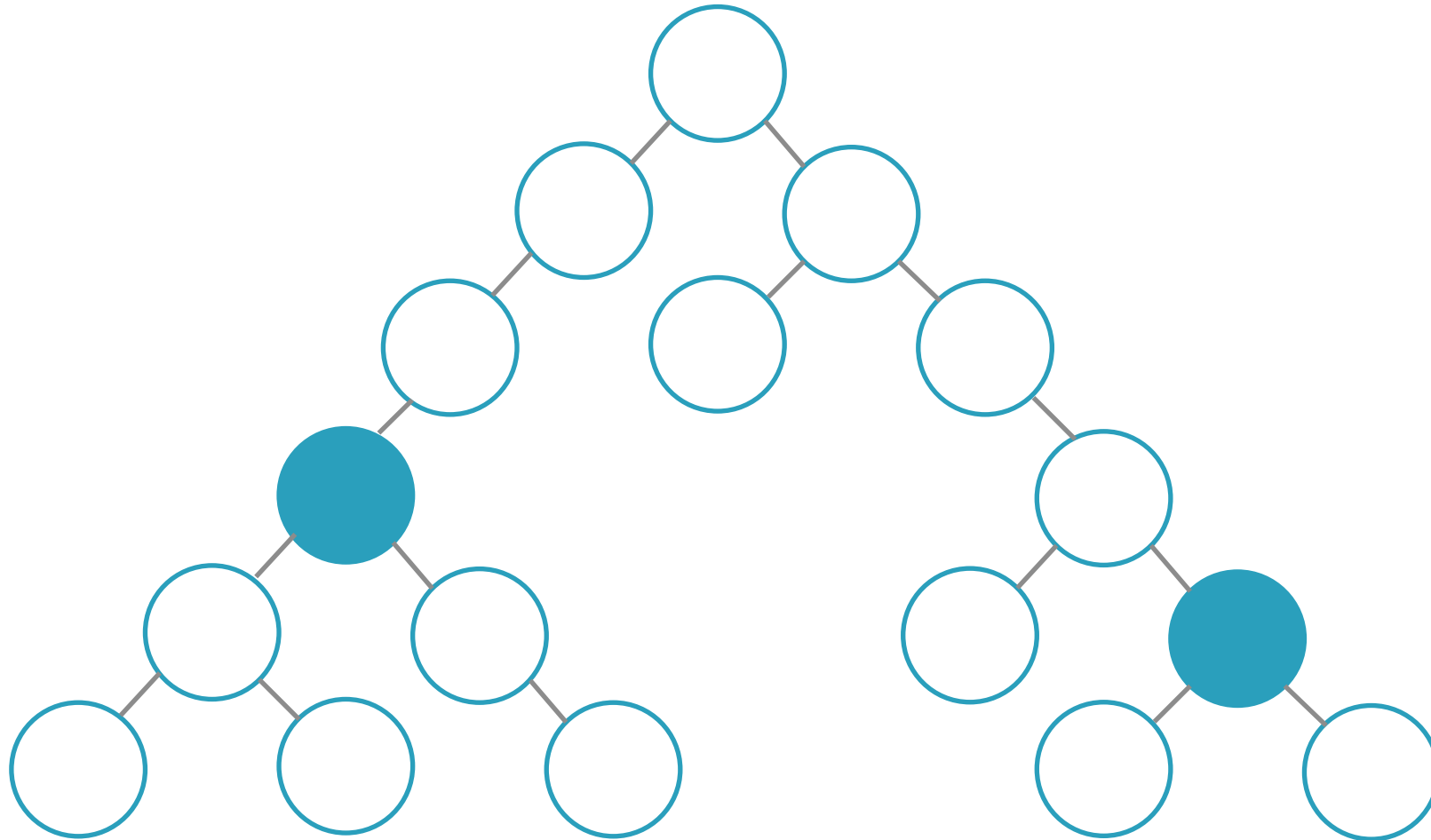
Simple
No setup

Complex
Significant setup



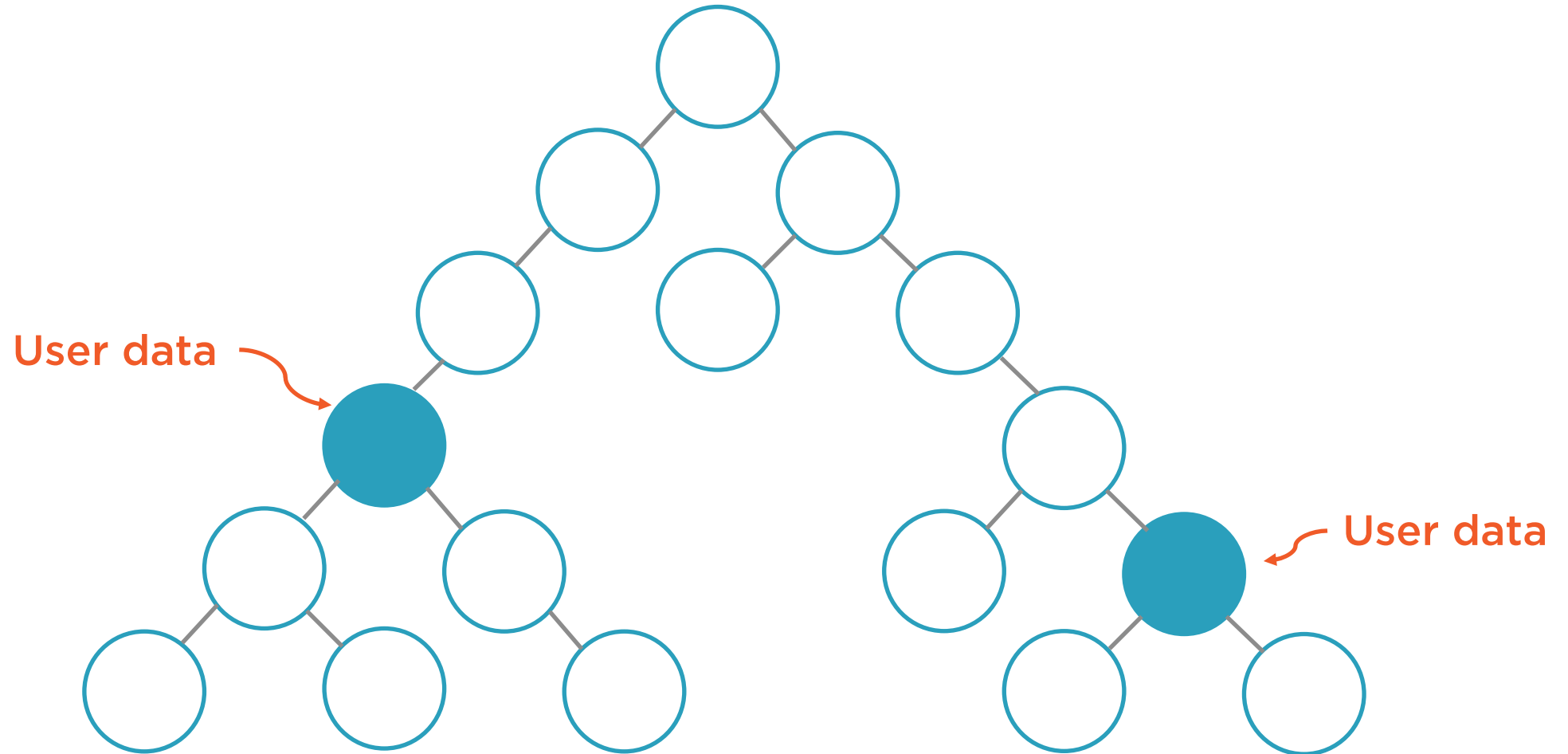


What if components in different parts of your app need the same data?



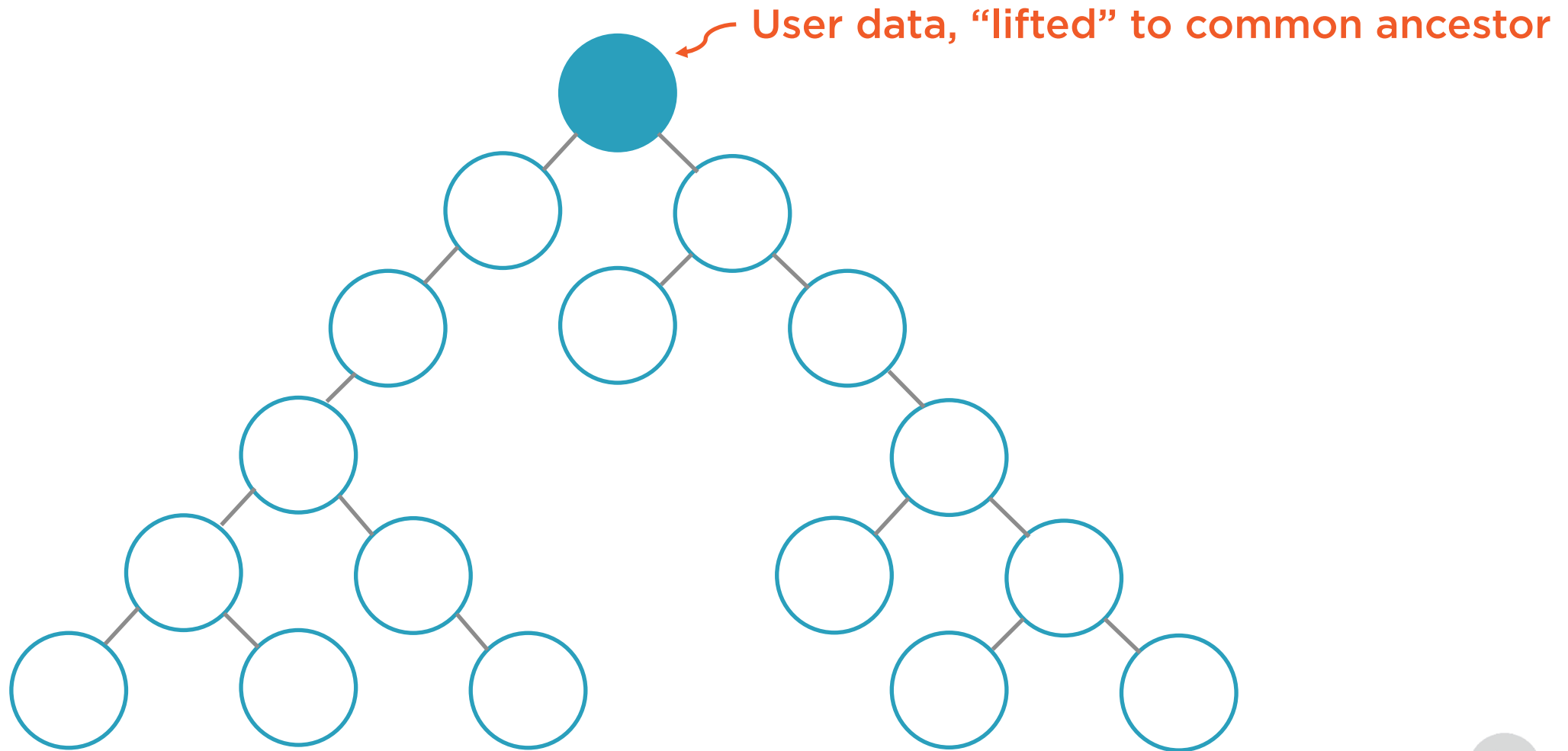
3 Solutions

1. Lift State



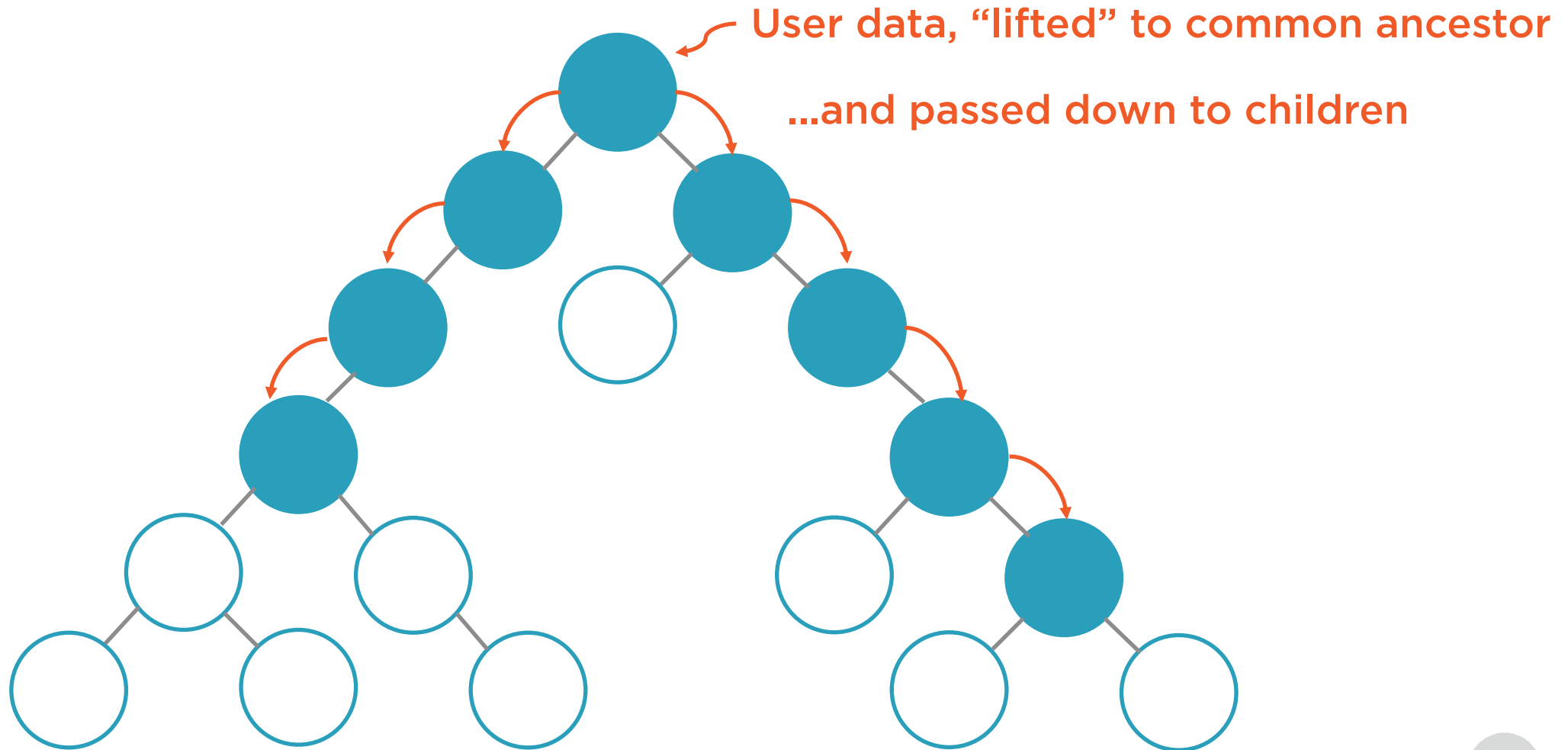
3 Solutions

1. Lift State



3 Solutions

1. Lift State

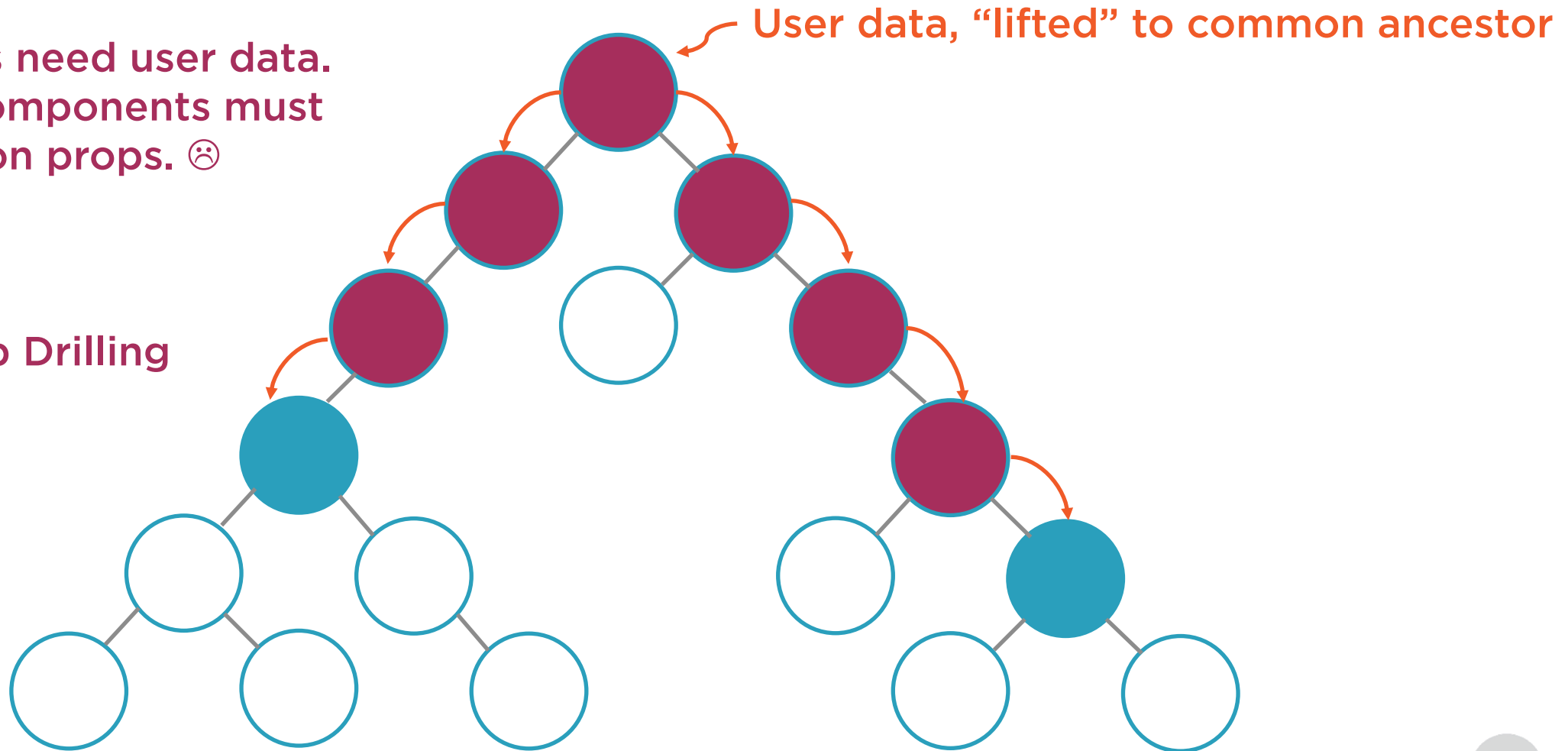


3 Solutions

1. Lift State

2 components need user data.
But 6 other components must
pass it down on props. 😞

Problem: Prop Drilling

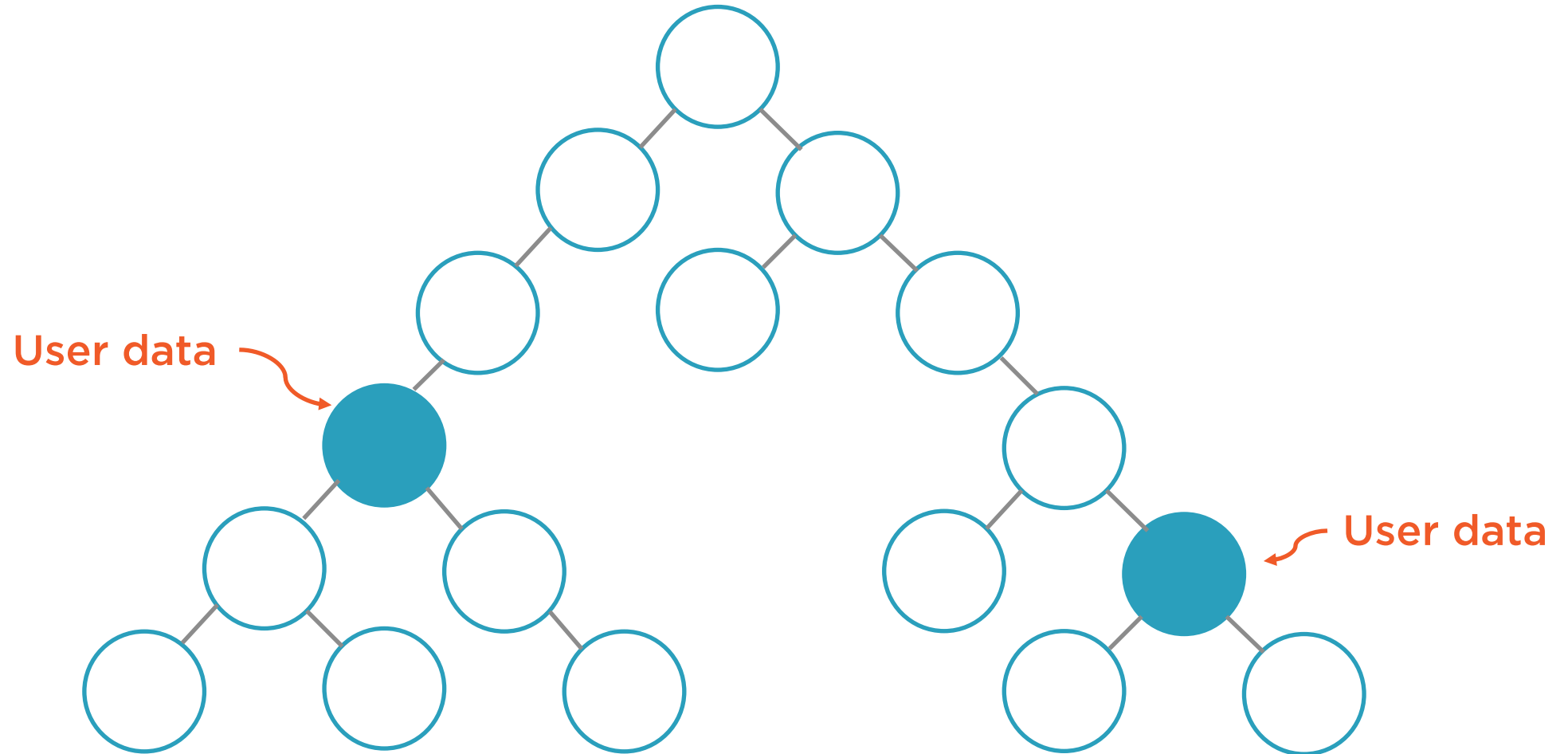


So is “prop drilling” bad?



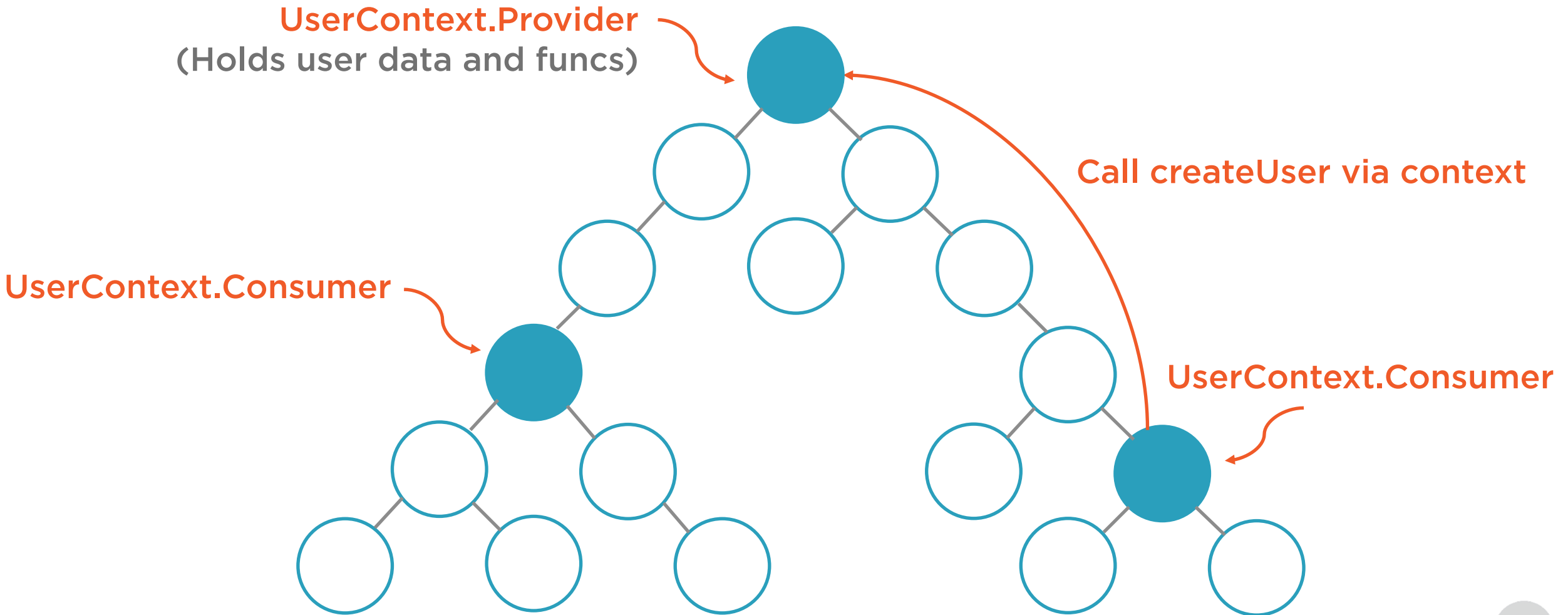
3 Solutions

1. Lift State
2. React context



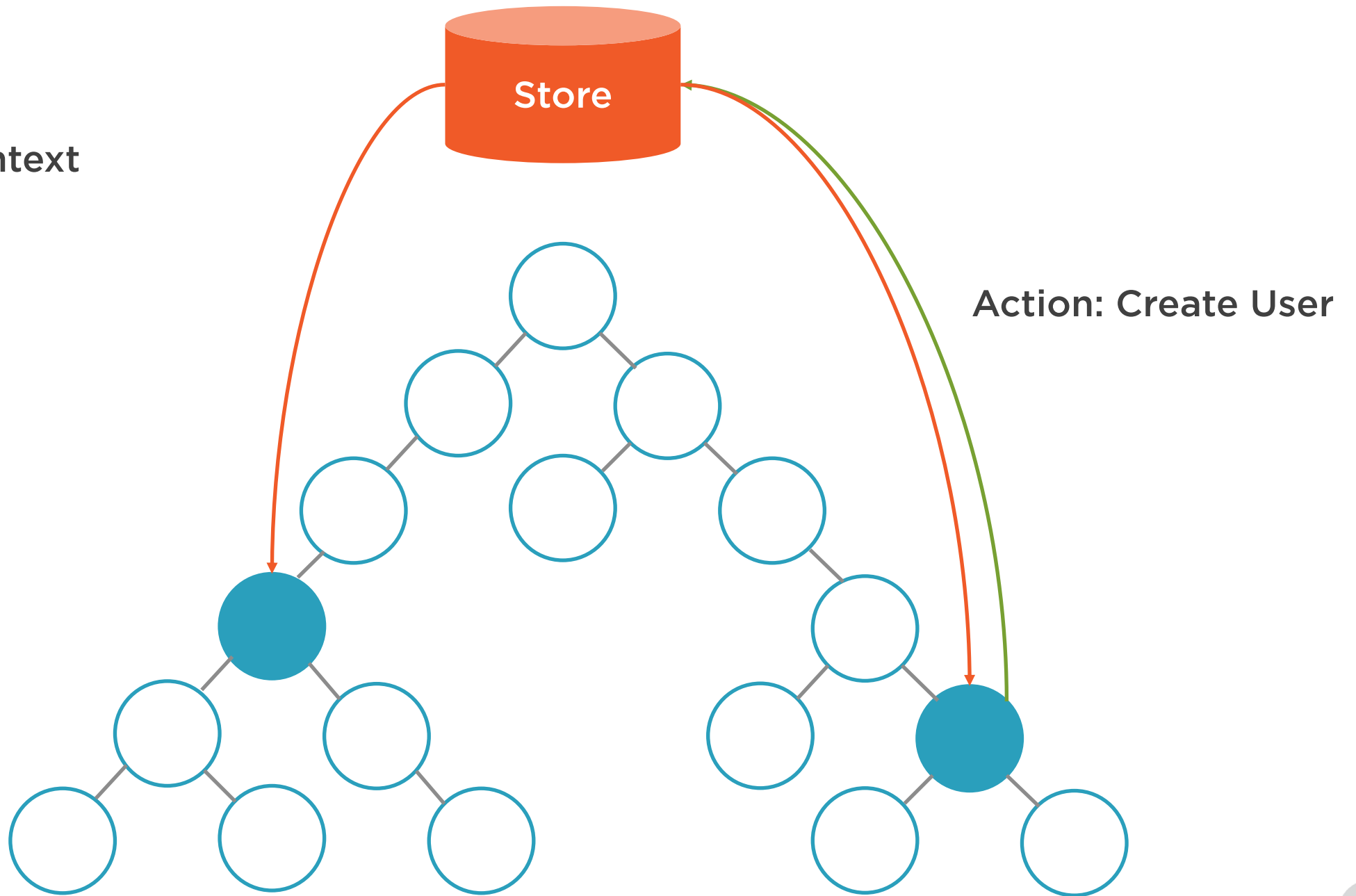
3 Solutions

1. Lift State
2. React context



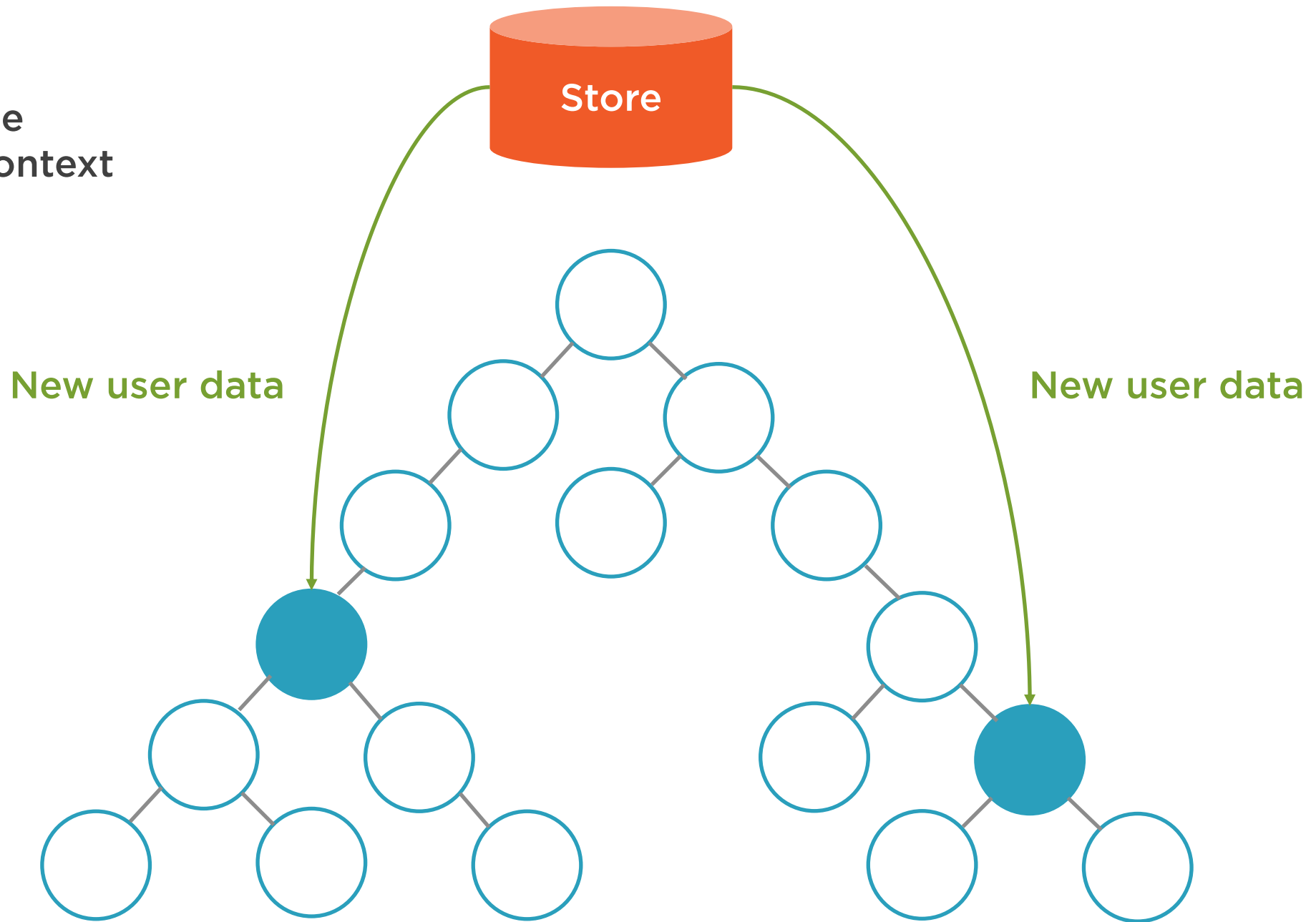
3 Solutions

1. Lift State
2. React context
3. Redux



3 Solutions

1. Lift State
2. React context
3. Redux



When are Context or Redux Helpful?



Global data and functions

Complex data flows

Inter-component communication

Many actions





Home

Browse

Search...

Paths

Channels

Bookmarks

Q&A



Building Applications with React and Redux

by Cory House

Learn how to use React, Redux, React Router, and modern JavaScript to build an app with React. Use Webpack, Babel, Jest, React Testing Library, Enzyme, and more to build a custom React development environment and build process from the ground up.

Resume Course

Bookmark

Add to Channel

Download Course

Dev environment, complete!

- Babel
- Webpack
- ESLint
- Mocha
- Express
- Jest

Course author



Cory House

Cory is the principal consultant at reactjsconsulting.com, where he has helped dozens of companies transition to React. Cory has trained over 10,000 software developers at events and businesses...

Course info

Level Intermediate

Rating ★★★★★ (1572)

My rating ★★★★★

Duration 6h 39m

Updated 12 Mar 2019

Share course

Table of contents

Description


Transcript

Exercise files

Discussion

Learning Check

Related Courses

This course is part of:  React Path

Expand All



Course Overview



1m 36s



My take

1. Start with state in a single component
2. Lift state as needed
3. Try context or Redux when lifting state gets annoying

Each item on this list is more complex, but also more scalable.
Consider the tradeoffs.



Why Centralize Context?



Easier to use



Enforce consistency



Better performance



Summary



Global data or functions?

- Consider context or Redux

Context: Two pieces:

- Provider: Provides data/funcs
- Consumer: Consumes data/funcs

Consider “wrapping” context

- Provider component
- useContext hook
- Require context provider

Next up: State in class components

