Intro to Redux

Cory House

Consultant

@housecor | www.reactjsconsulting.com

Agenda

Do I need Redux?

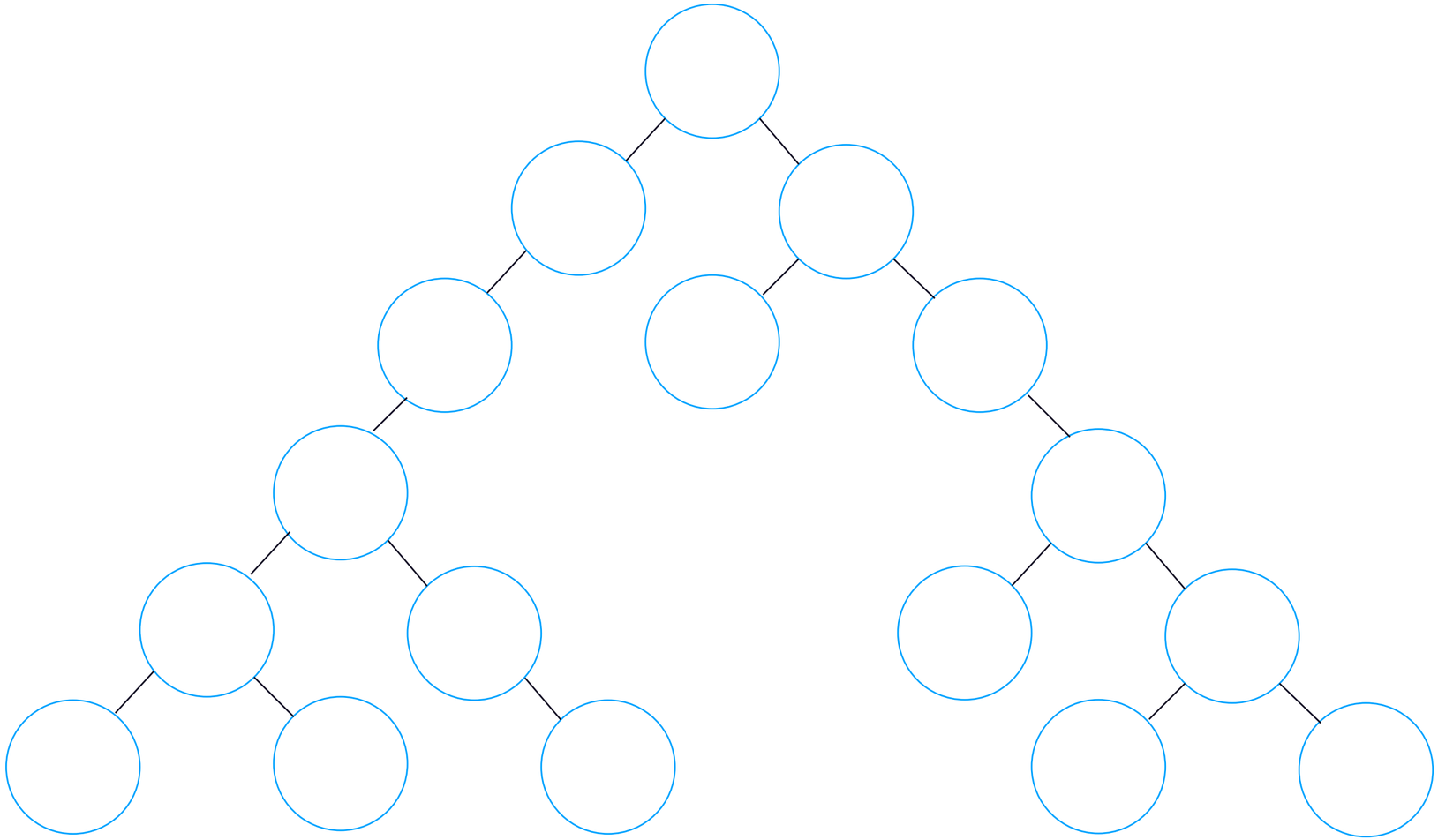
3 principles

Flux vs. Redux

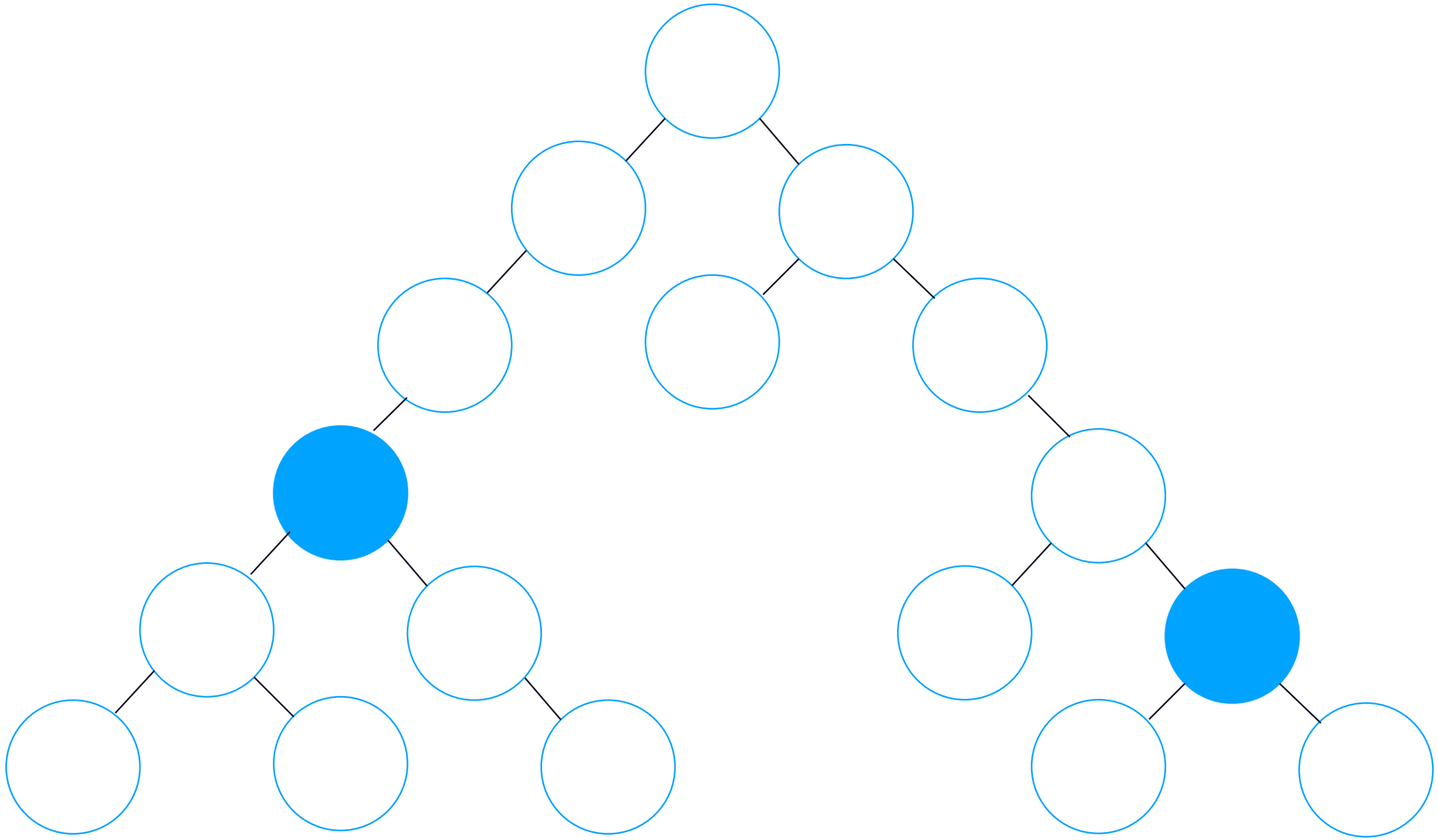
Redux flow

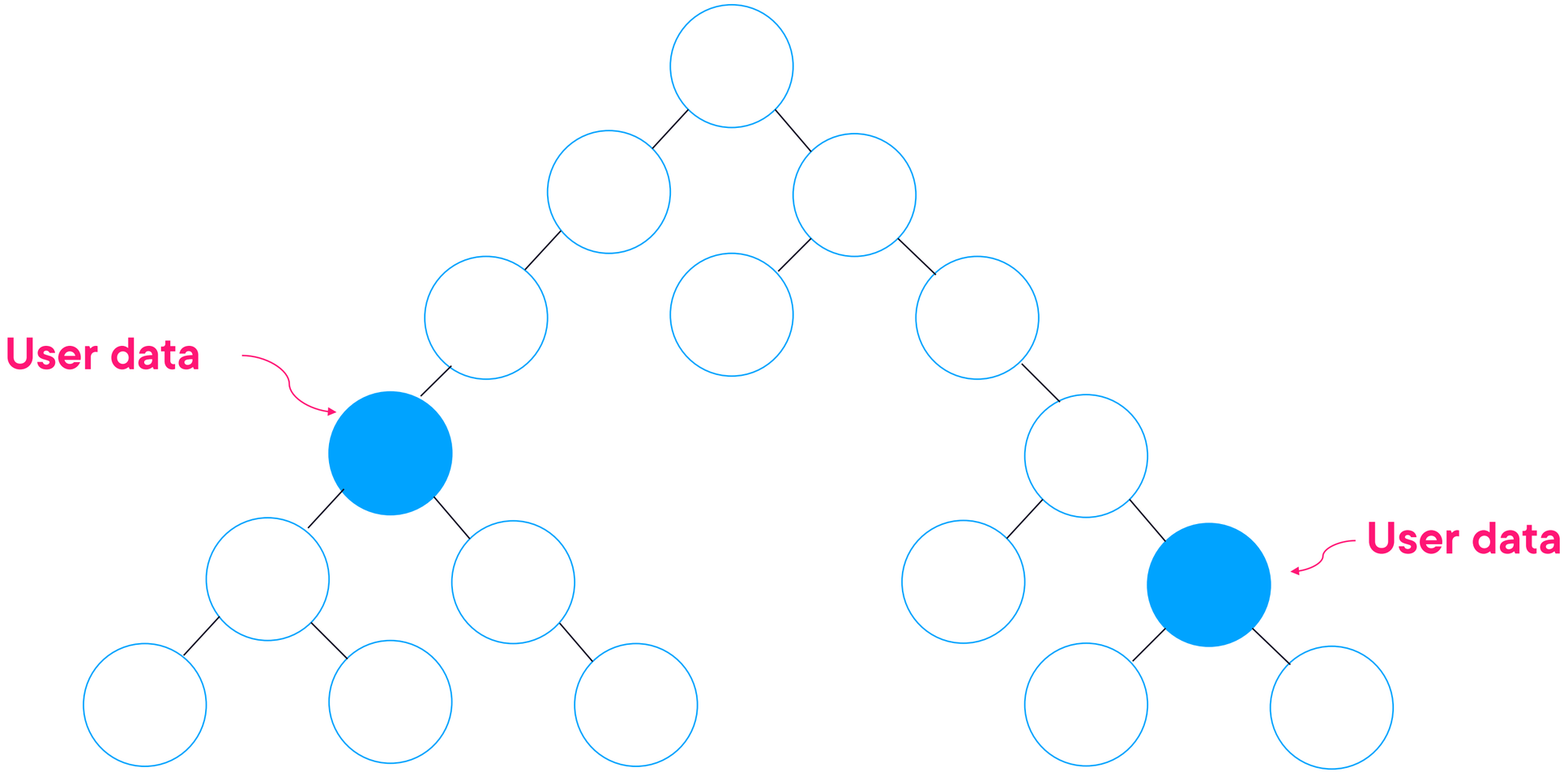
Do I Need Redux?

Plain JS	React	React context	React with Redux
Simple No setup			Complex Significant setup



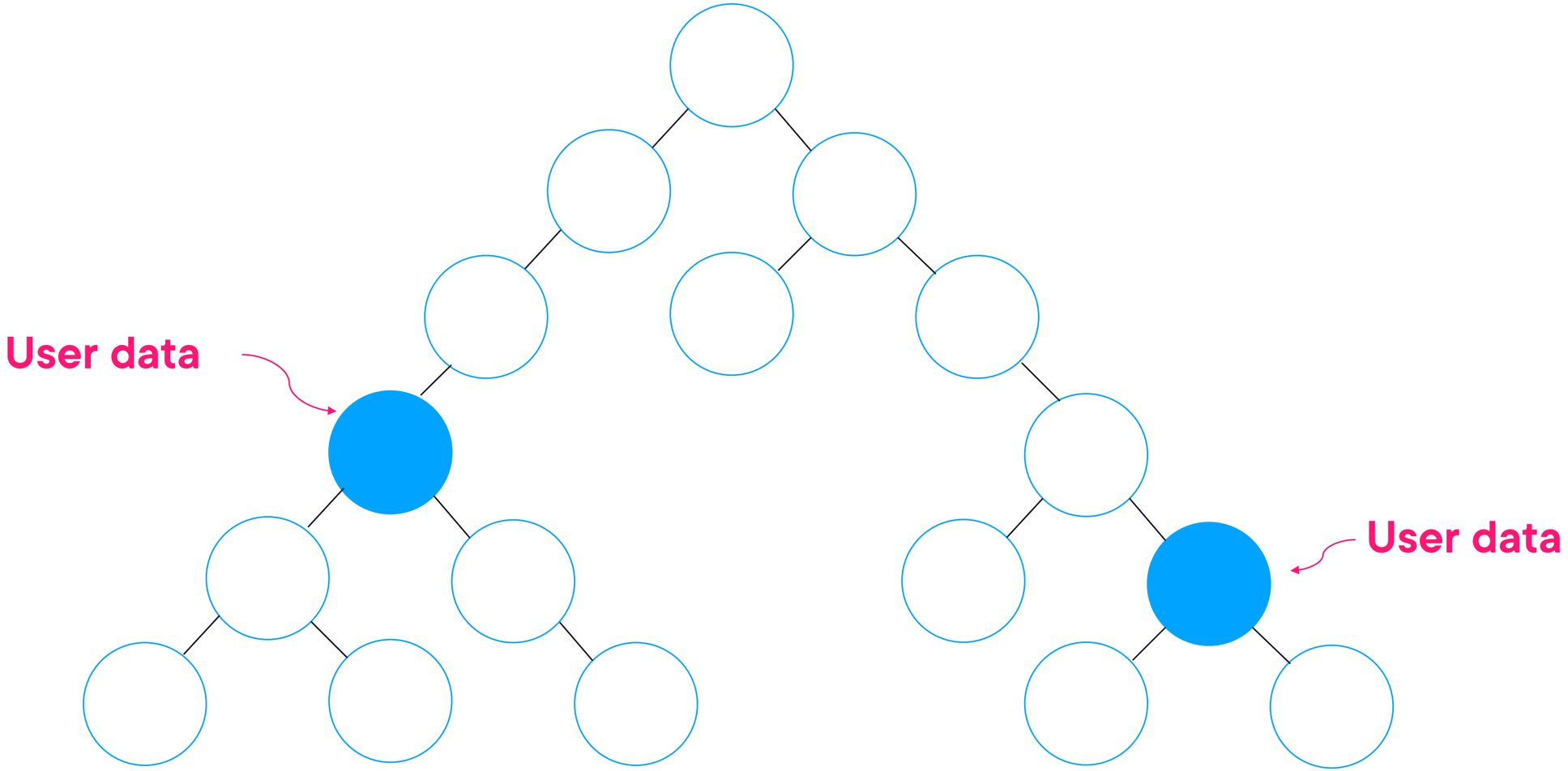
What if components in different parts of your app need the same data?




User data
User data

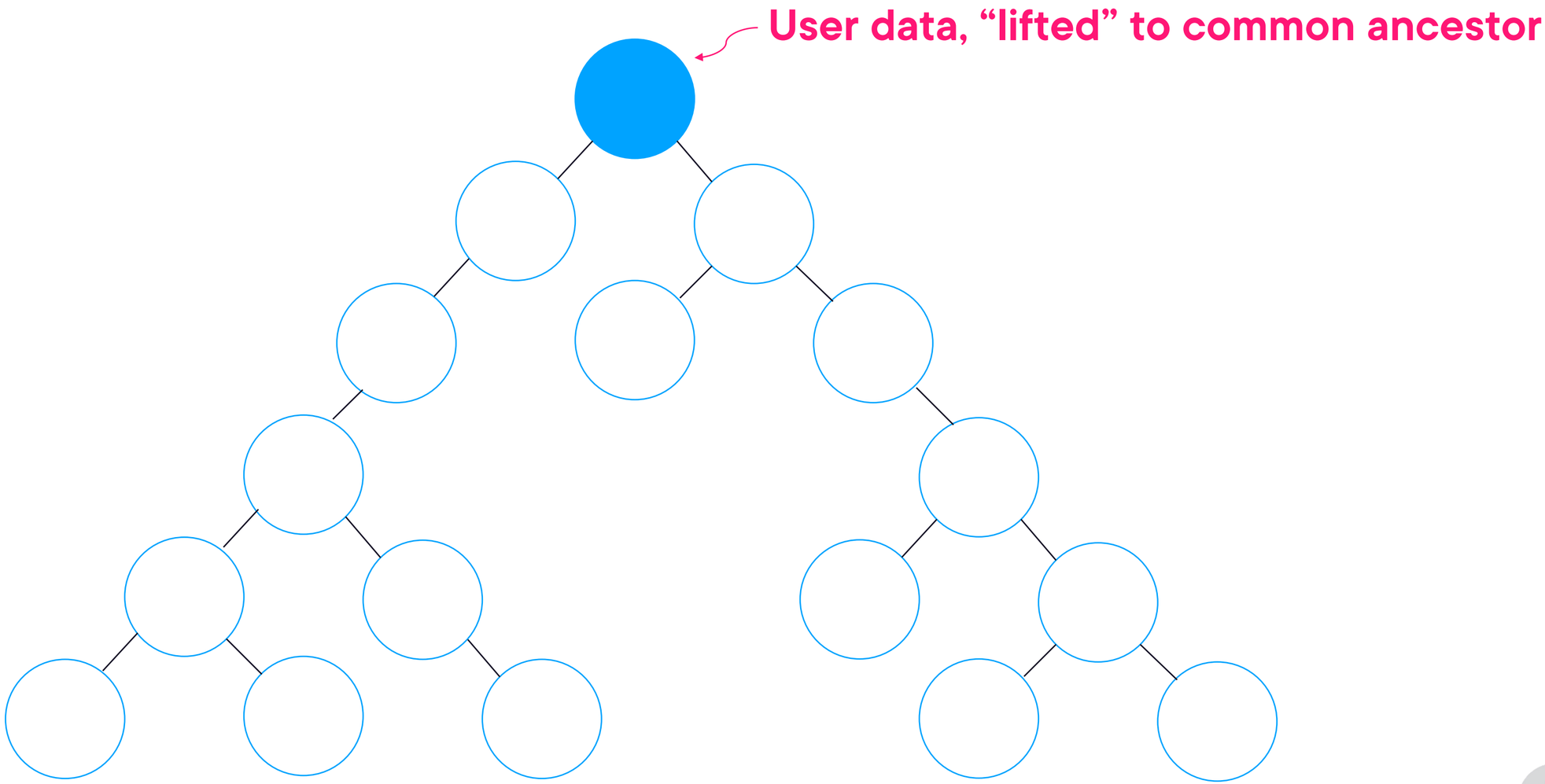
3 Solutions

1. **Lift State**



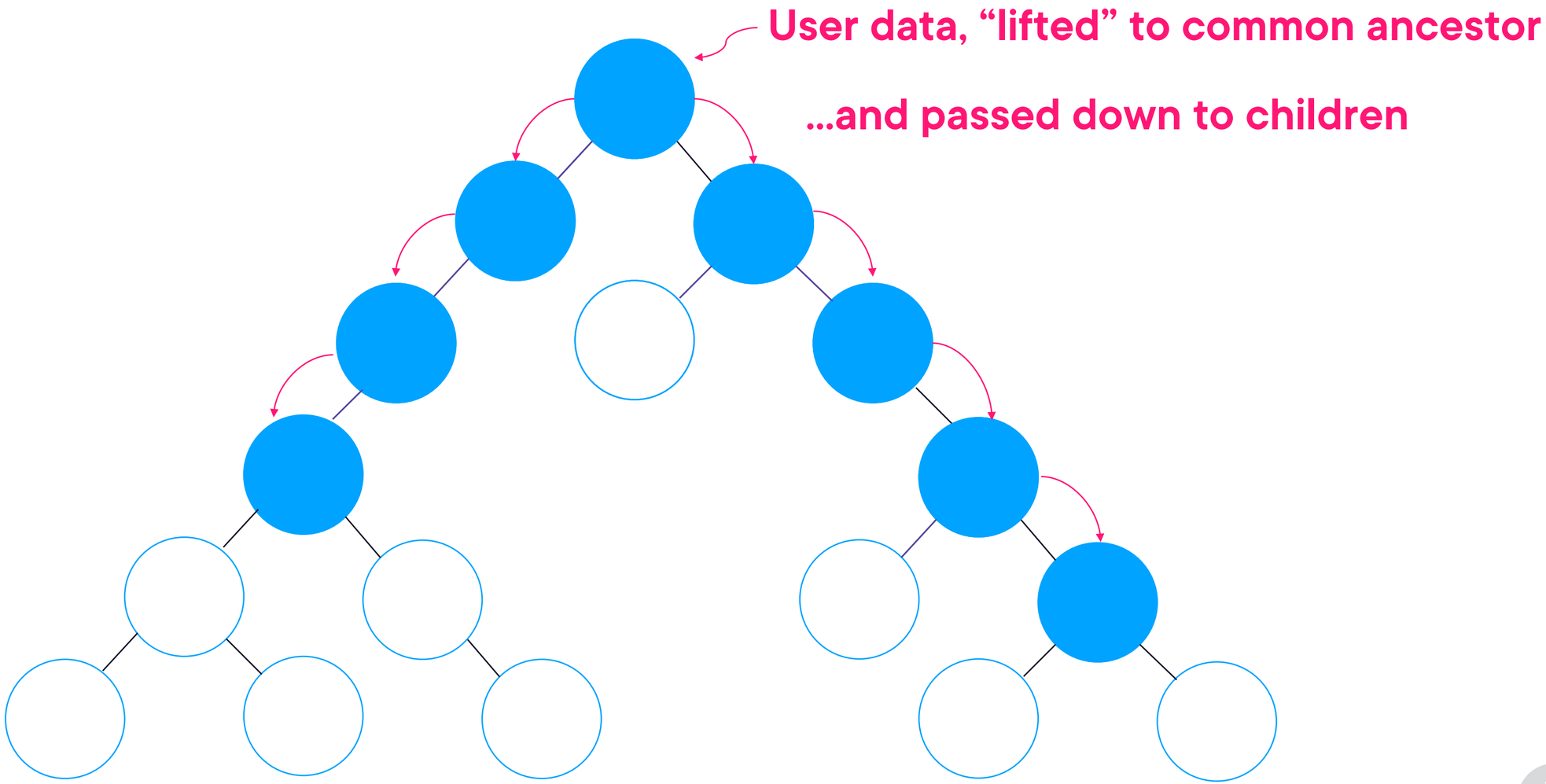
3 Solutions

1. **Lift State**



3 Solutions

1. **Lift State**

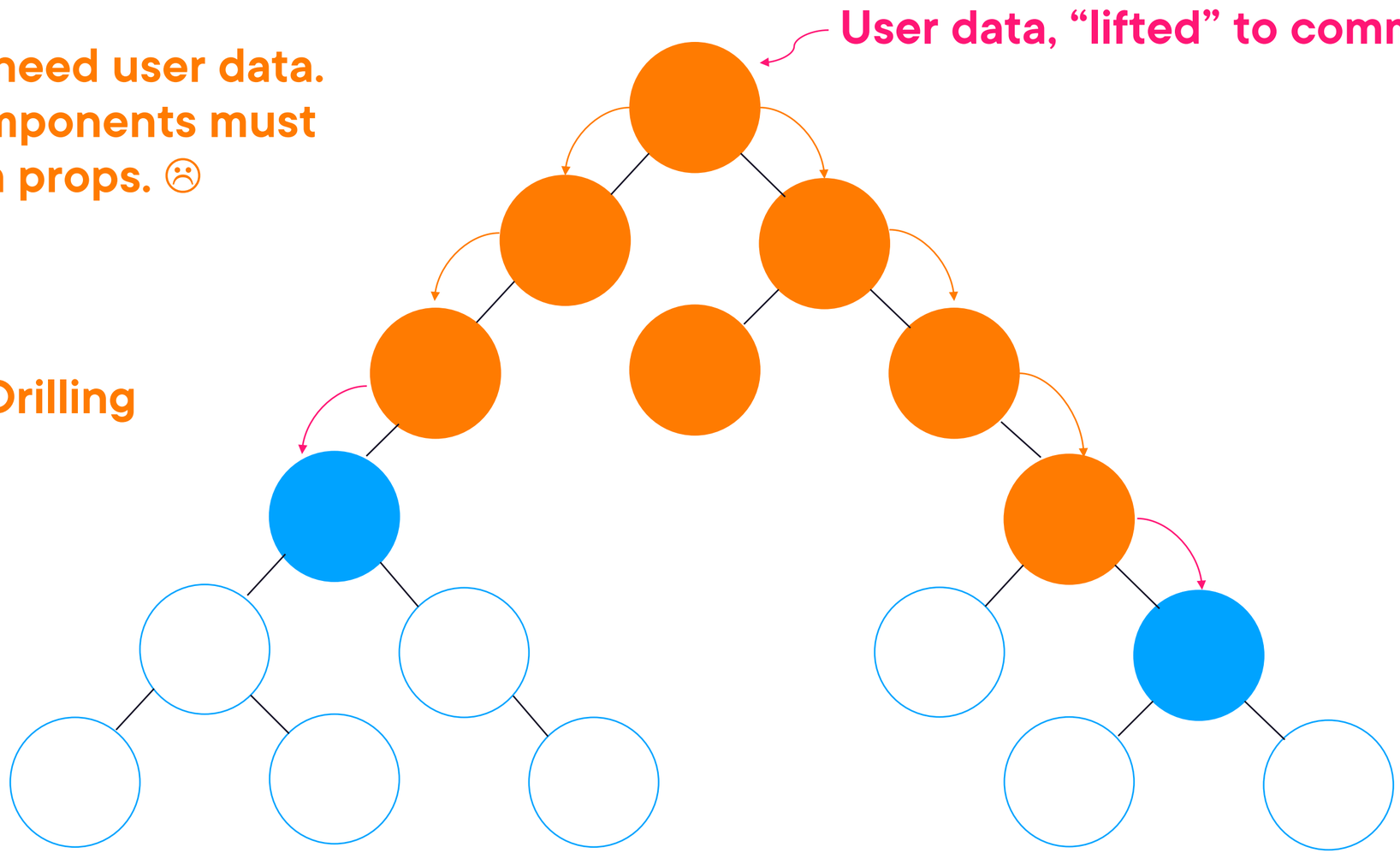


3 Solutions

1. **Lift State**

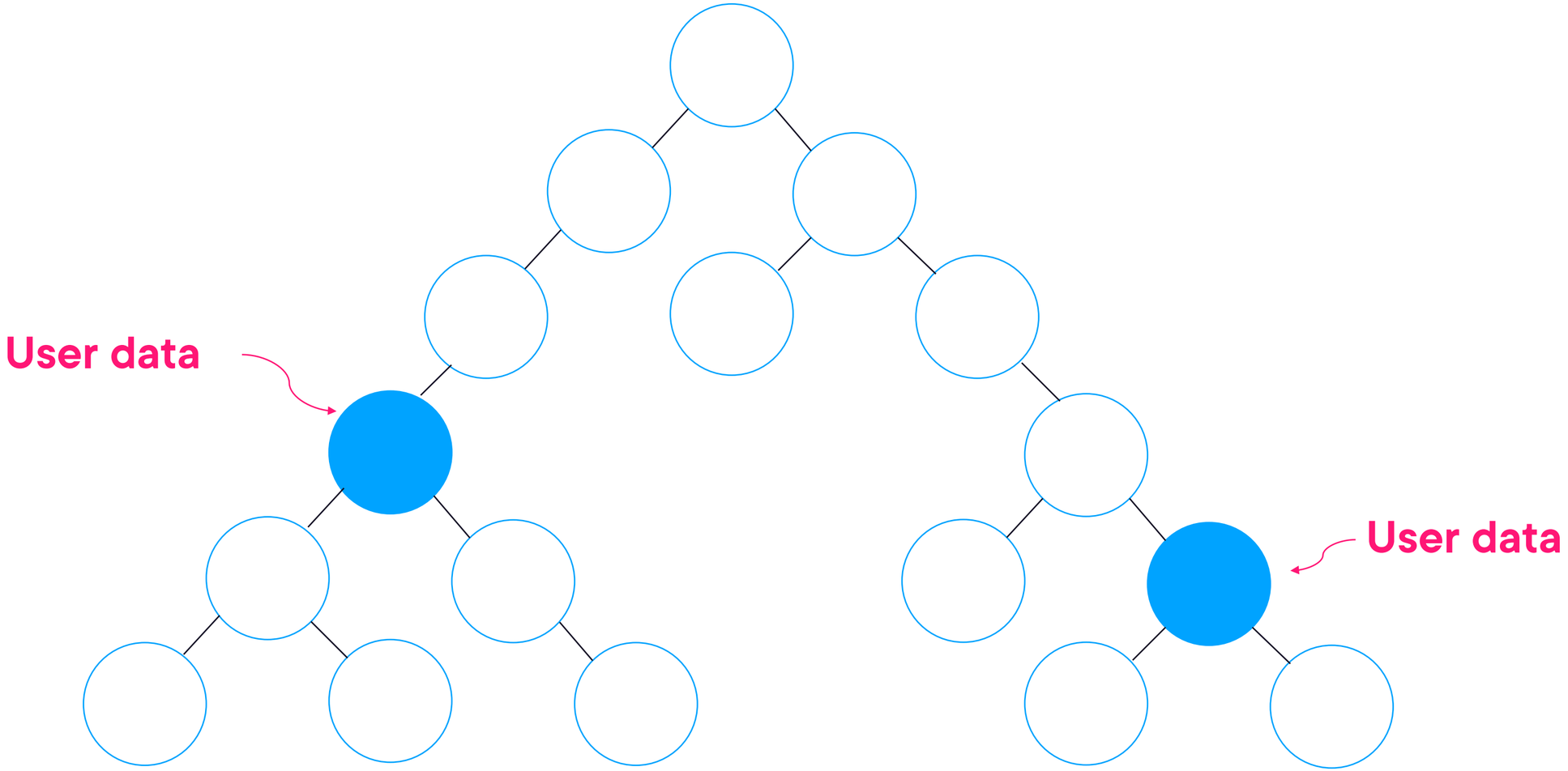
2 components need user data.
But 6 other components must pass it down on props. ☹

Problem: Prop Drilling



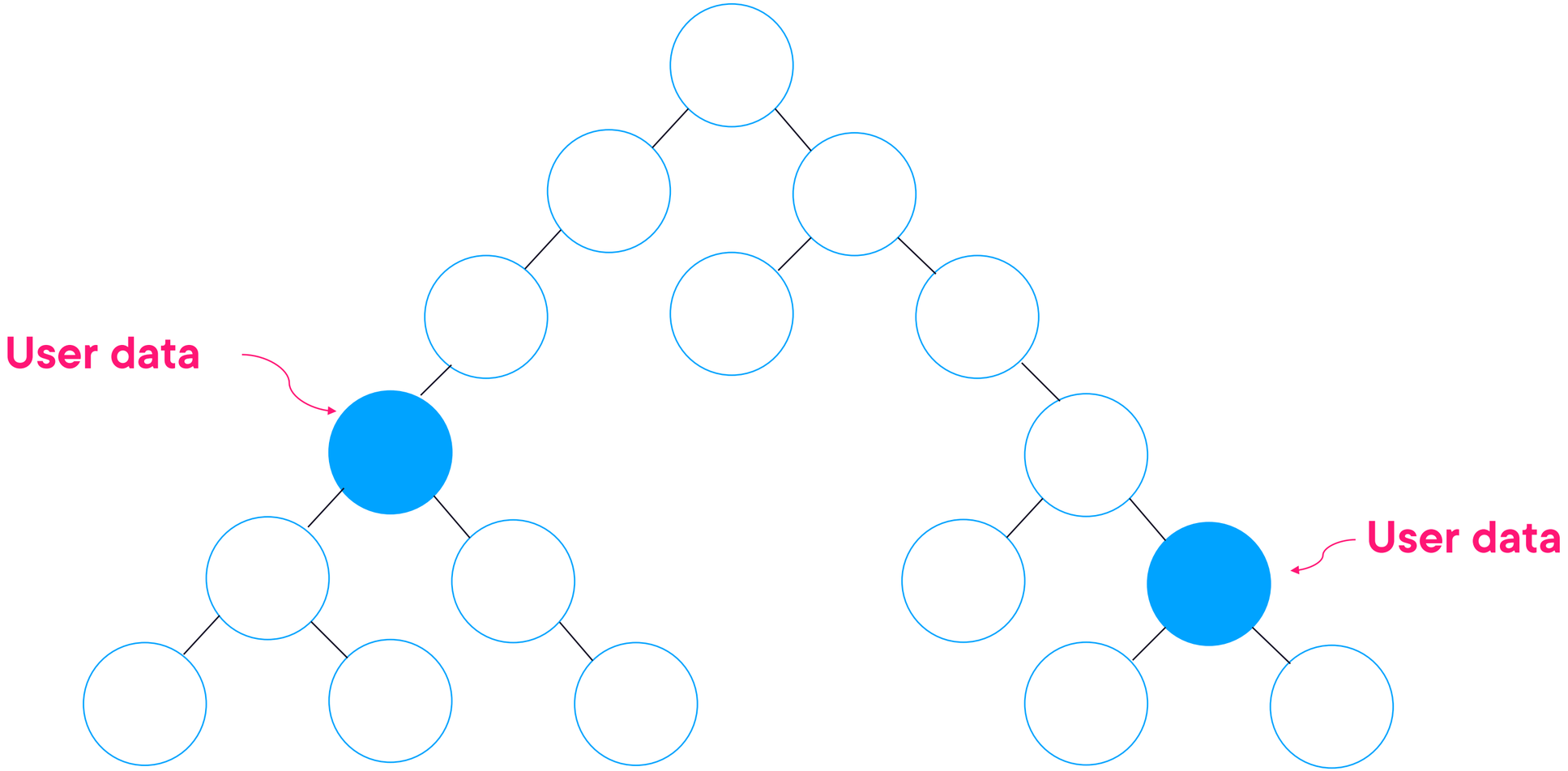
3 Solutions

1. Lift State
2. React Context



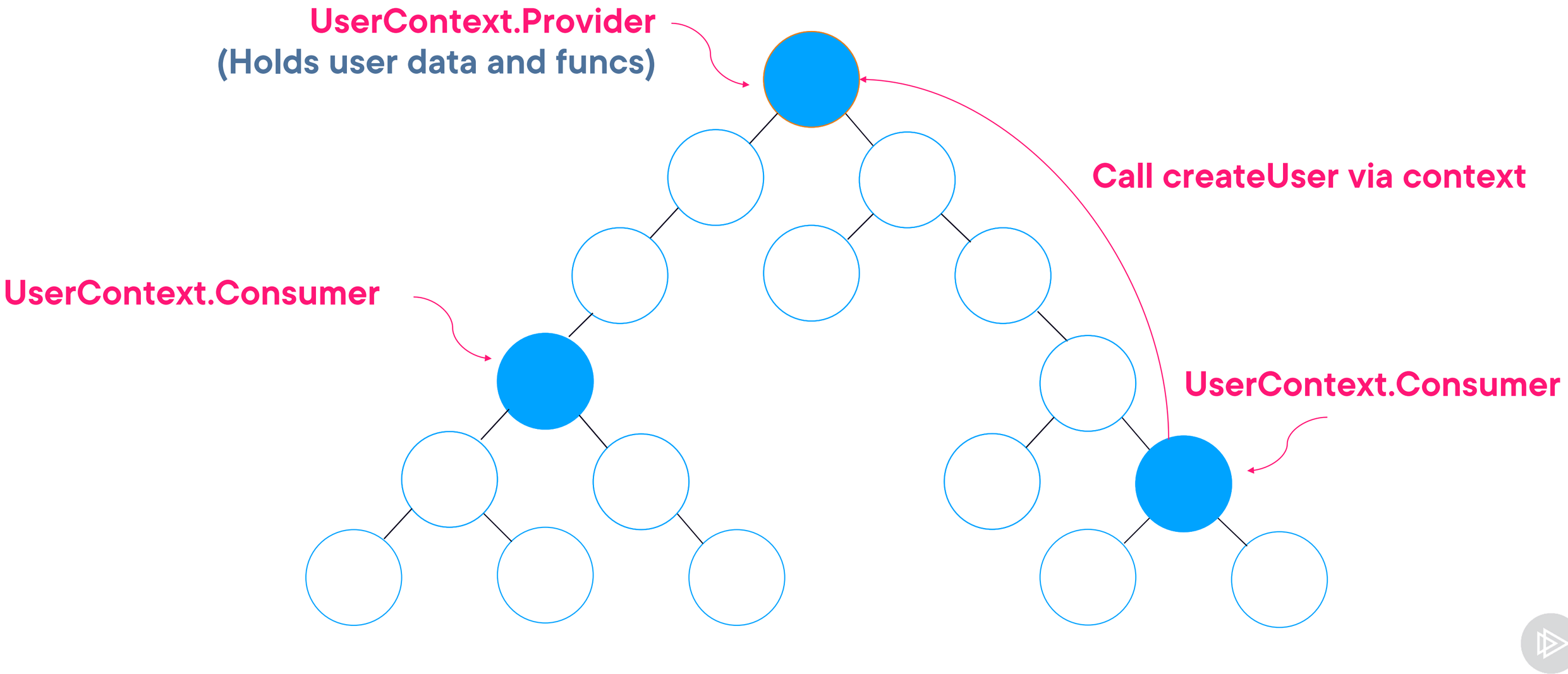
3 Solutions

1. Lift State
2. React Context



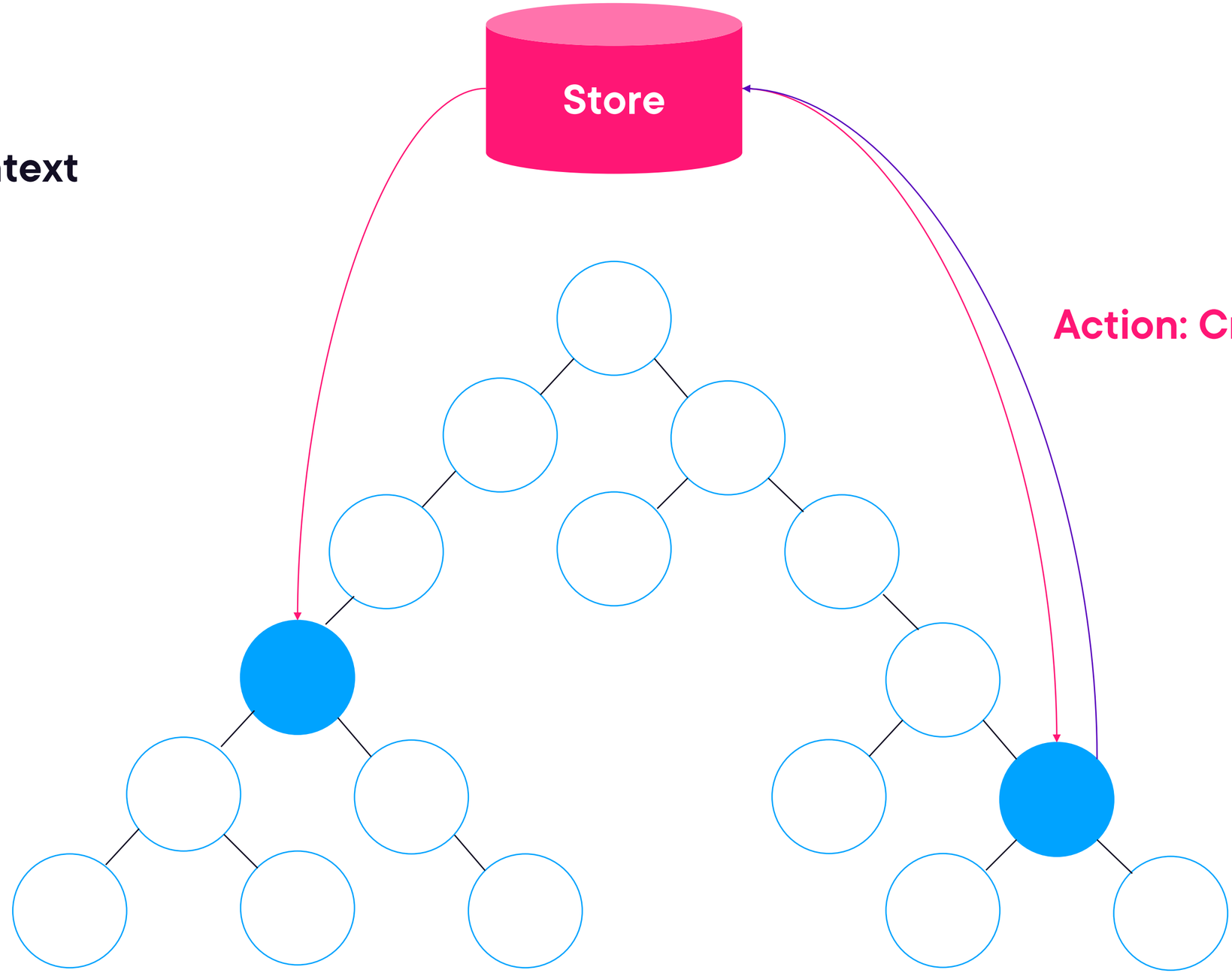
3 Solutions

1. Lift State
2. React Context



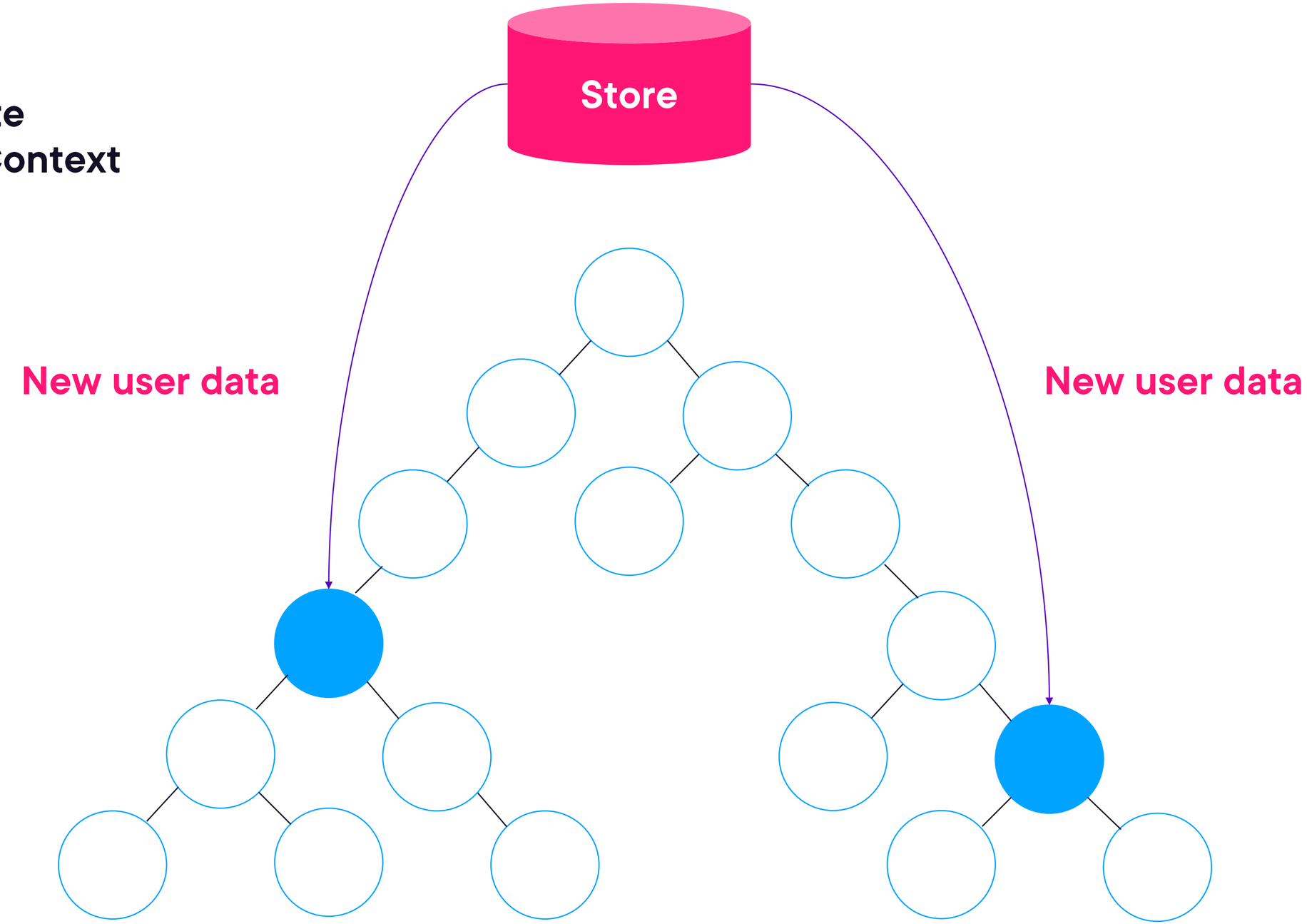
3 Solutions

1. Lift State
2. React Context
3. Redux



3 Solutions

1. Lift State
2. React Context
3. Redux



When Is Redux Helpful?

Complex data flows

Inter-component communication

Non-hierarchical data

Many actions

Same data used in many places

Pete Hunt

“...If you aren’t sure if you need it, you don’t need it.”

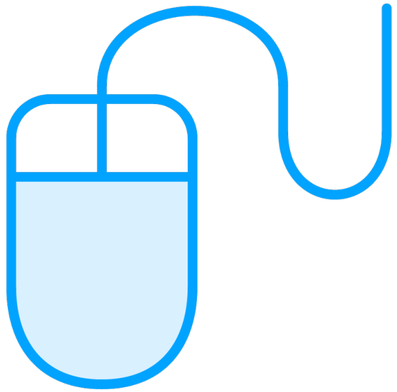
Recommendation

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

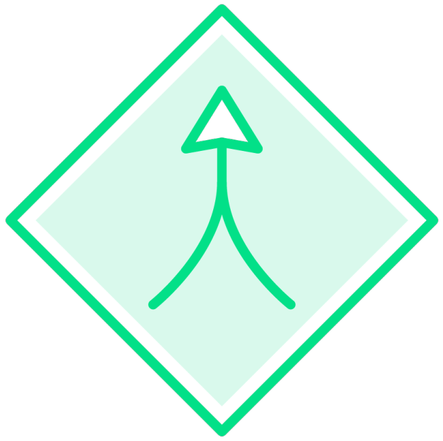
Each item on this list is more complex, but also more scalable.
Consider the tradeoffs.

Redux: 3 Principles

One immutable store



Actions trigger changes



Reducers update state

Example action:

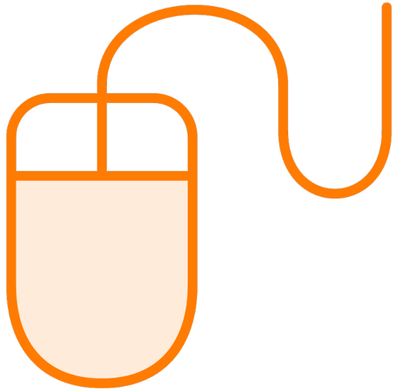
```
{
  type: SUBMIT_CONTACT_FORM,
  message: 'Hi.'
}
```

Flux vs. Redux

Data down.
Actions up.

Flux and Redux: Similarities

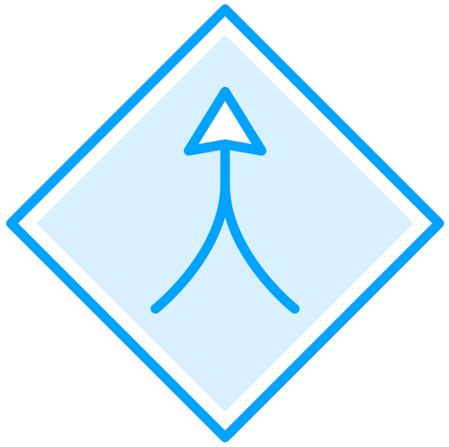
Unidirectional Flow



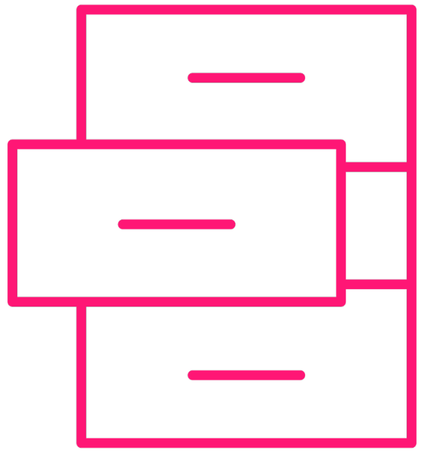
Actions

Stores

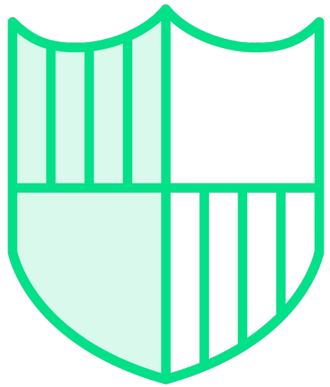
Redux: New Concepts



Reducers



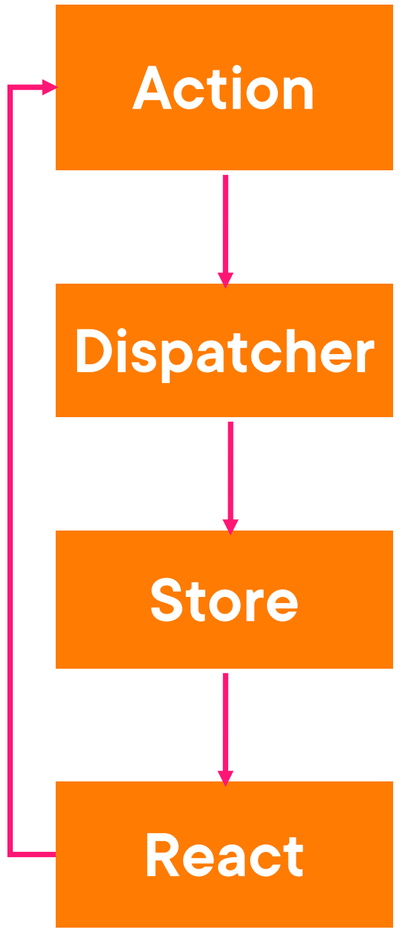
Containers



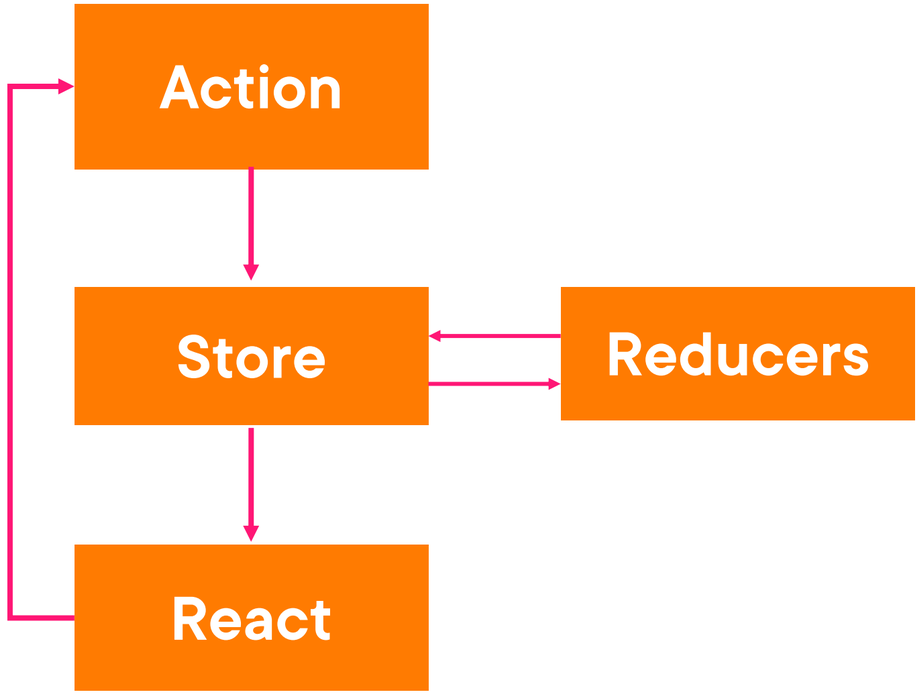
Immutability

Flux vs. Redux

Flux

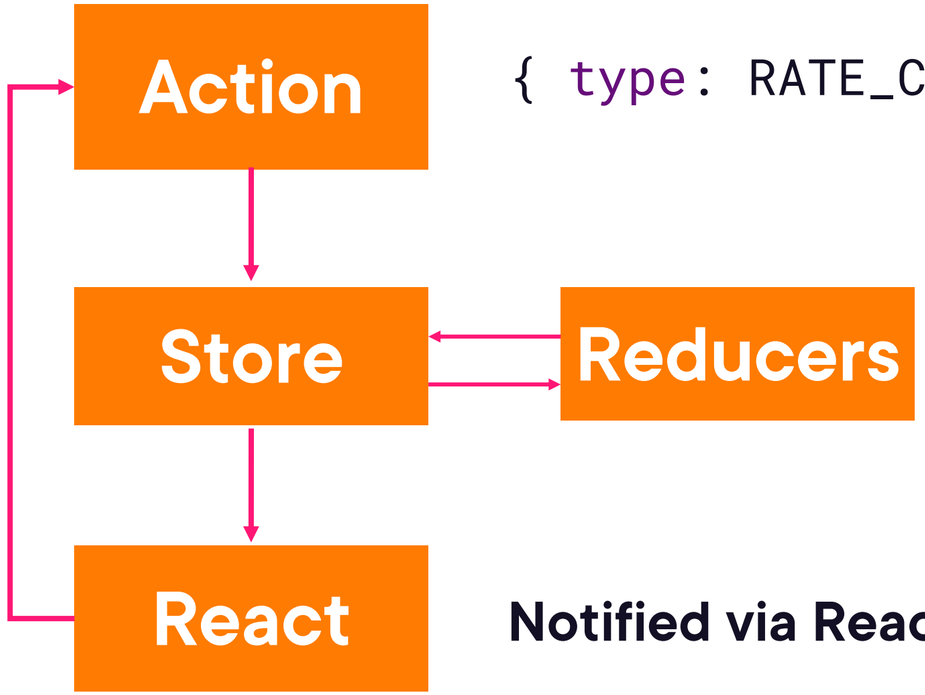


Redux



Redux Flow

Redux Flow




```
{ type: RATE_COURSE, rating: 5 }
```

```
function appReducer(state = defaultState, action) {
    switch(action.type) {
        case RATE_COURSE:
        //return new state
    }
}
```

Notified via React-Redux

Summary

If you need Redux, you'll know it.

3 Principles

- Immutable Store
- Actions trigger changes
- Reducers return updated state

Flux vs Redux

Unidirectional Redux Flow

Up Next:

Actions, Stores, and Reducers