CS390
WEB APPLICATION DEVELOPMENT

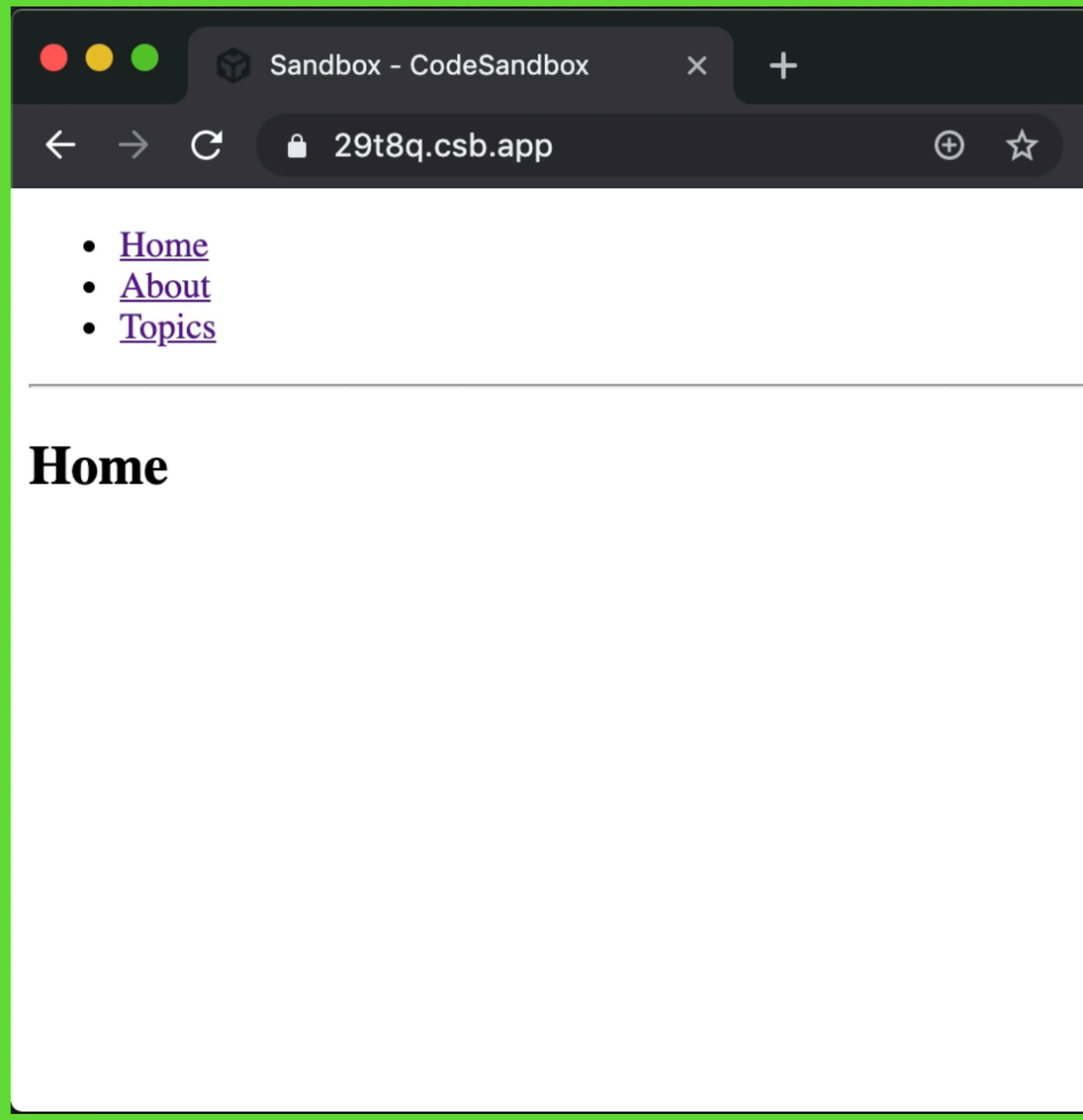
NISARG KOLHE

Routing

Routing

React apps work more like apps than websites.

Adding routing makes your UI in sync with the URL, making it behave more like a website.



Routing

Use **react-router**.
Easy to use.
Cross compatible with React Native.

https://reacttraining.com/react-router/

```
npm i --save react-router
```

Talking with the server

Talking with the server

There are some things which you **can't** or just **shouldn't** do on the front-end.

Eg. authentication, storing global data, heavy computations etc.

Talking with the server

Use **Axios**.
Makes XMLHttpRequests from the browser.
Returns a **promise**.

```
npm i --save axios
```

When to make a request?

- componentDidMount()
- componentDidUpdate()
- Action function
- **NOT** in render()!

Example request

Specify the URL and HTTP request type

Callback function returning the data

Callback function in case the request fails

Callback function which always executes after the query

```
axios.get('/user?ID=12345')
  .then(function (response) {
    // handle success
    console.log(response);
  })
  .catch(function (error) {
    // handle error
    console.log(error);
  })
  .finally(function () {
    // always executed
  });
```

Resources to learn more about Axios

Using Axios with React:

https://alligator.io/react/axios-react/

Axios docs:

https://github.com/axios/axios

Hooks!

Hooks

lets you use **state** and **lifecycle** features without using **class** and React component lifecycle methods.

Custom hooks let you share **state logic** between components!

State hook

```
class Example extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      count: 0
    };
  }

  render() {
    return (
      <div>
        <p>You clicked {this.state.count} times</p>
        <button
          onClick={() =>
            this.setState({count:this.state.count+1})
          }>
          Click me
        </button>
      </div>
    );
  }
}
```

State hook

```
class Example extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      count: 0
    };
  }

  render() {
    return (
      <div>
        <p>You clicked {this.state.count} times</p>
        <button
          onClick={() =>
            this.setState({count:this.state.count+1})
        }>
          Click me
        </button>
      </div>
    );
  }
}
```

with hooks:

```
function Example() {
  const [count, setCount] = useState(0);

  return (
    <div>
      <p>You clicked {count} times</p>
      <button onClick={() =>
        setCount(count + 1)
      }>
        Click me
      </button>
    </div>
  );
}
```

State hook

```
[count, setCount] = useState(0)
```

Effect hook

```
componentDidMount() {
    document.title = `You clicked ${this.state.count} times`;
}

componentDidUpdate() {
  document.title = `You clicked ${this.state.count} times`;
}
```

Effect hook

```
componentDidMount() {
    document.title = `You clicked ${this.state.count} times`;
}

componentDidUpdate() {
    document.title = `You clicked ${this.state.count} times`;
}
```

with hooks:

```
useEffect(() => {
   document.title = `You clicked ${count} times`;
});
```

Effect hook

```
componentDidMount() {
    document.title = `You clicked ${this.state.count} times`;
}

componentDidUpdate() {
    document.title = `You clicked ${this.state.count} times`;
}

componentWillUnmount() {
    document.title = `Component unmounted`;
}
```

with hooks:

```
useEffect(() => {
   document.title = `You clicked ${count} times`;
   return () => { document.title = `Component unmounted`; }
});
```

Resources to learn more about Hooks

React Hooks Documentation:

https://reactjs.org/docs/hooks-intro.html

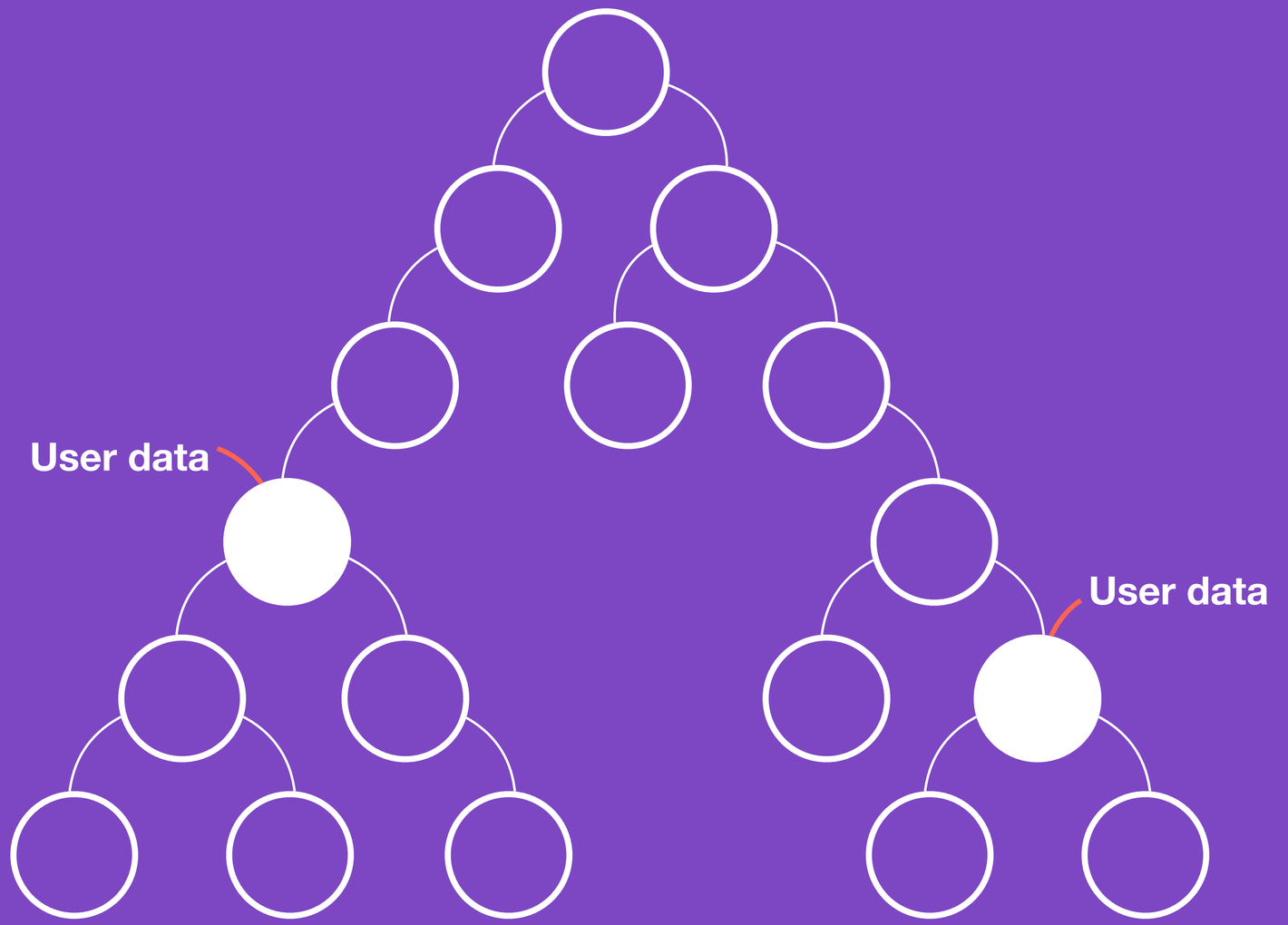
Why React Hooks?:

https://tylermcginnis.com/why-react-hooks/

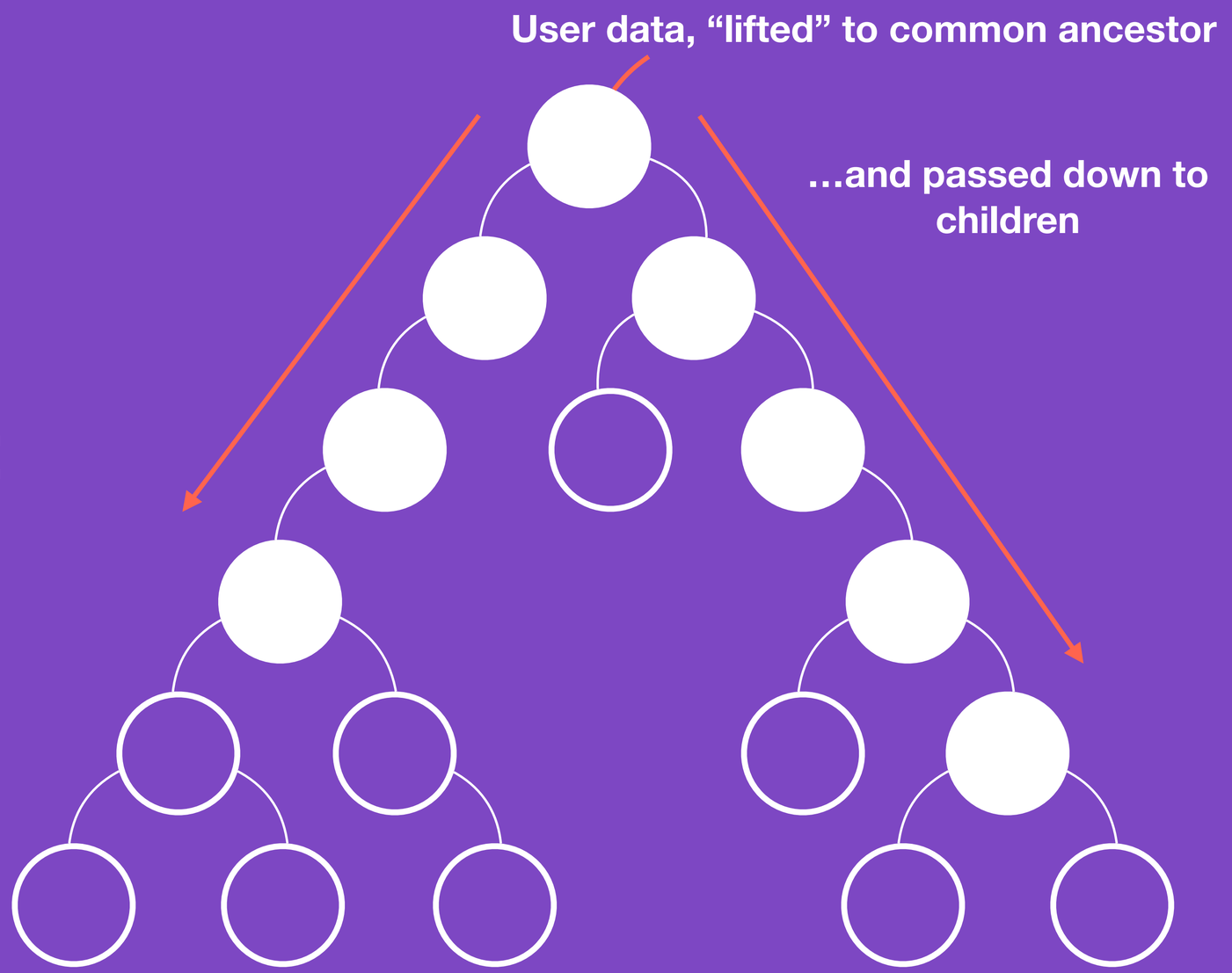
Redux

Do I need Redux?

What if components in different parts need the same data?

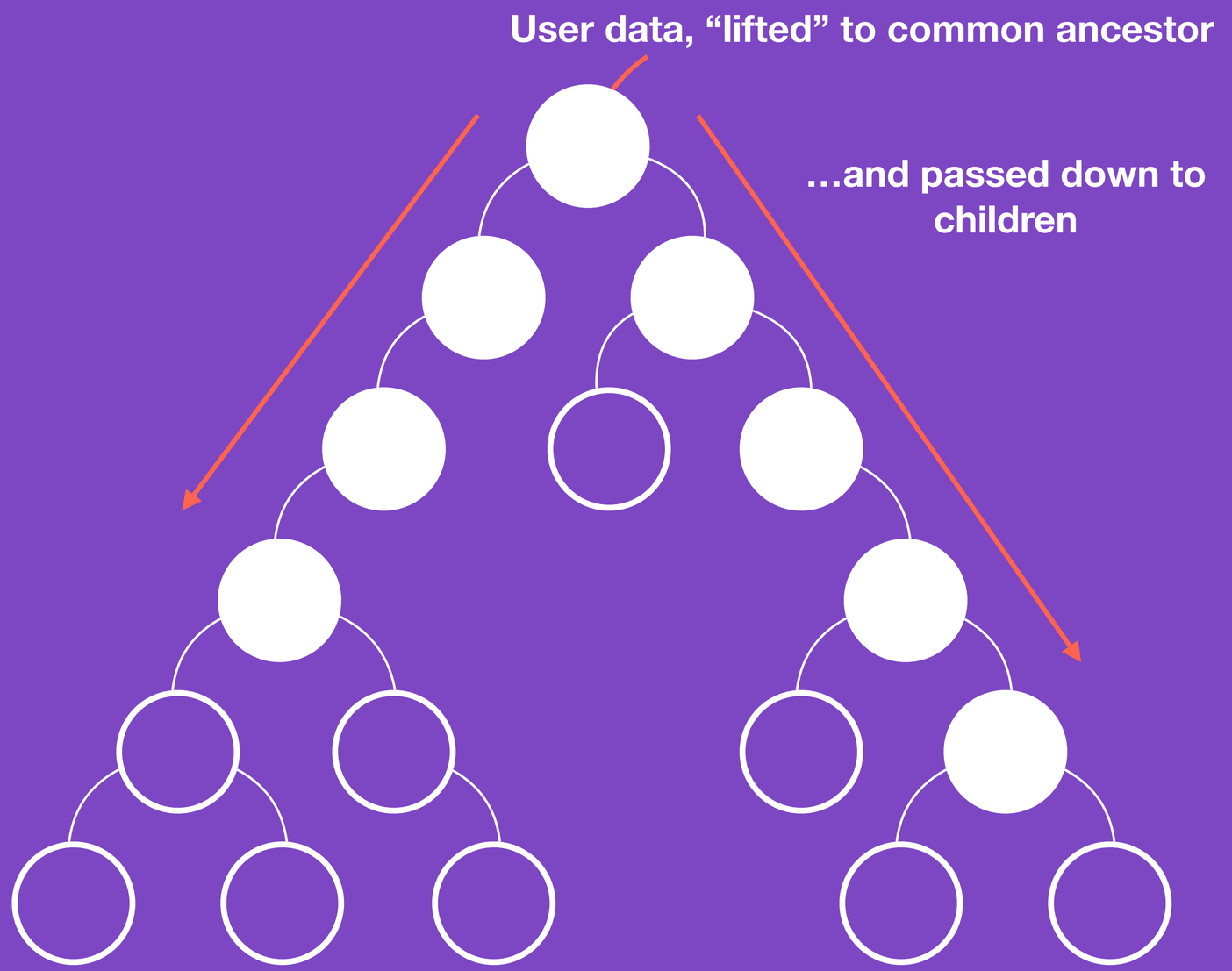


Solution 1:
Lift the state to the first common ancestor.

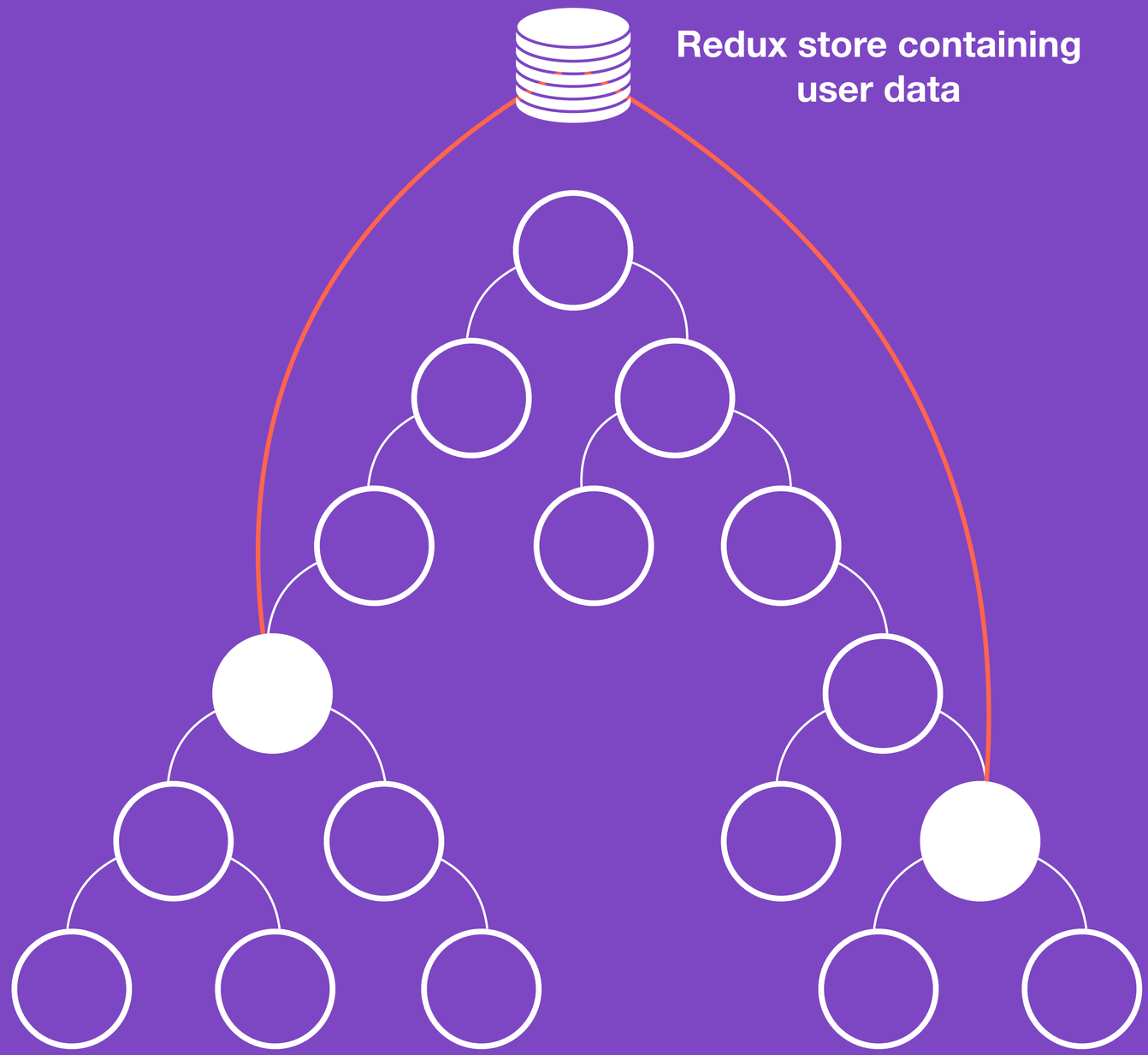


Solution 1:
Lift the state to the first common ancestor.

Problem:
Prop drilling, components having props just to pass it down



Solution 2:
Use a single source of truth (Redux)



Resources to learn Redux

LearnCode.academy Redux tutorial (highly recommended):

https://www.youtube.com/playlist?list=PLoYCgNOIyGADIlc3iUJzygCqC8Tt3bRXt

A Cartoon Intro to Redux:

https://code-cartoons.com/a-cartoon-intro-to-redux-3afb775501a6

Is that it in React?

Portals

Testing

Higher-Order Components

Render props

Is that it in React? Nope.

Context

Forwarding Refs

Relay

Typechecking With PropTypes