CS5220 Advanced Topics in Web Programming

React for Building Single-Page Applications (SPA)

Chengyu Sun
California State University, Los Angeles

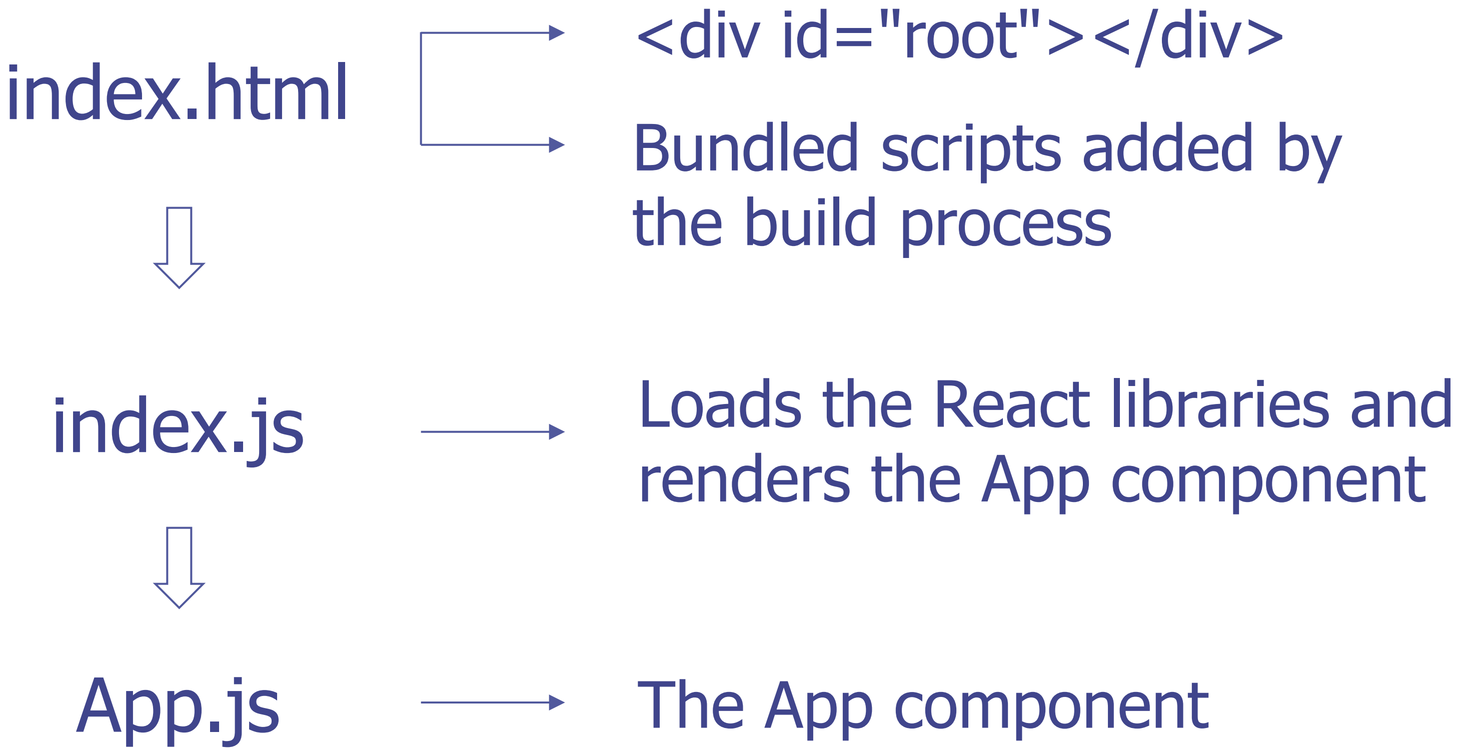
Create React App

- The recommended way to create a new single-page app

```
npx create-react-app <name>
```

- See `package.json` for build commands
- Use `npm run eject` to see the tooling underneath

Basic App Structure



index.html

<div id="root"></div>

Bundled scripts added by the build process

index.js

Loads the React libraries and renders the App component

App.js

The App component

Understand App.js

- Import and export
  - ES6 module
  - Webpack imports
- `className`

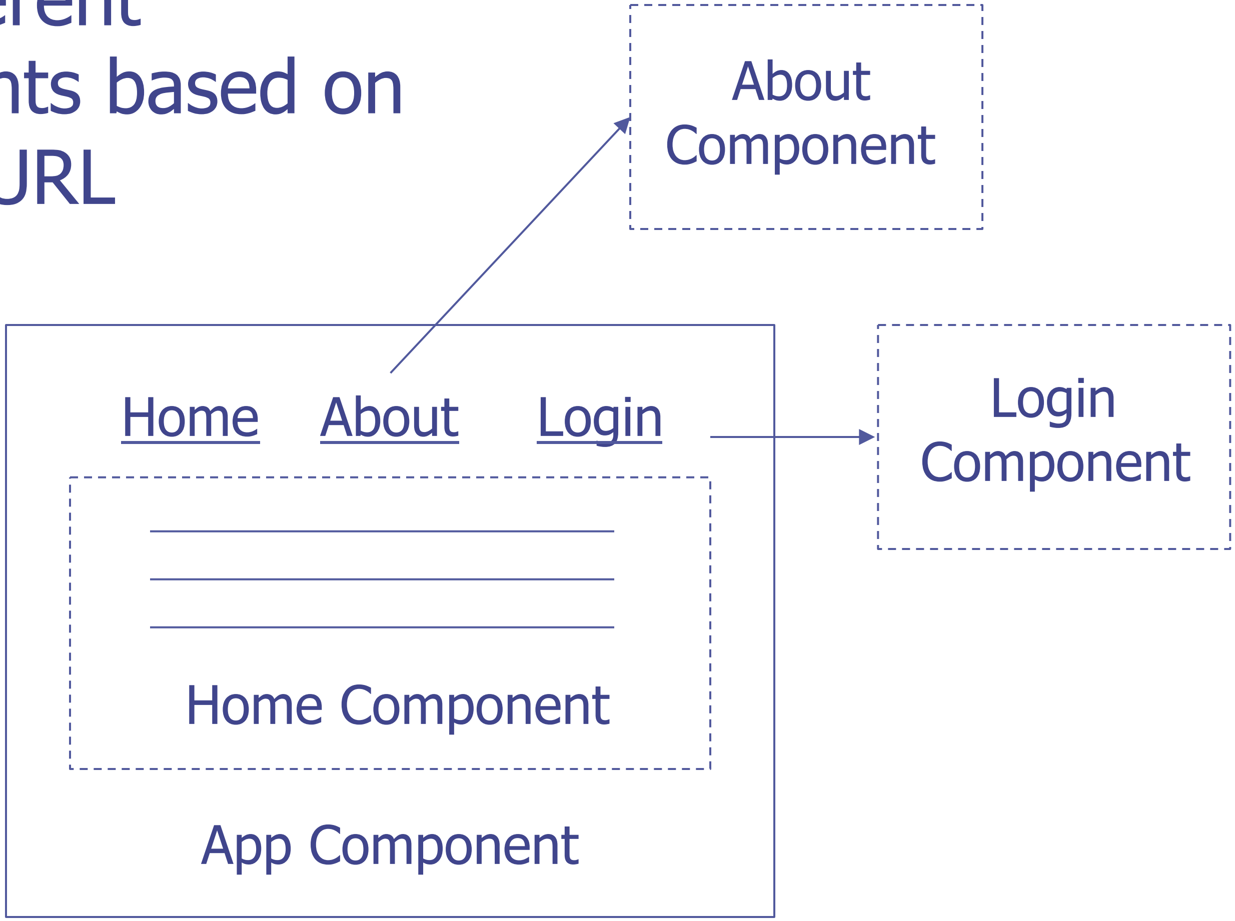
The Need for Routing

- Usually we have different pages in a web application
- How do we have different "pages" in a SPA?

Home About Login

Routing

- Load different components based on different URL



React Router

- React Router is the most popular routing solution for React

```
npm install react-router-dom
```

Basic Usage of React Router

…

```
<Router>
    <ul>
        <li><Link to="/">Home</Link></li>
        <li><Link to="/about">About</Link></li>
   </ul>
    <div id="content">
        <Route path="/" exact><Home /></Route>
        <Route path="/about"><About /><Route>
    </div>
</Router>
```

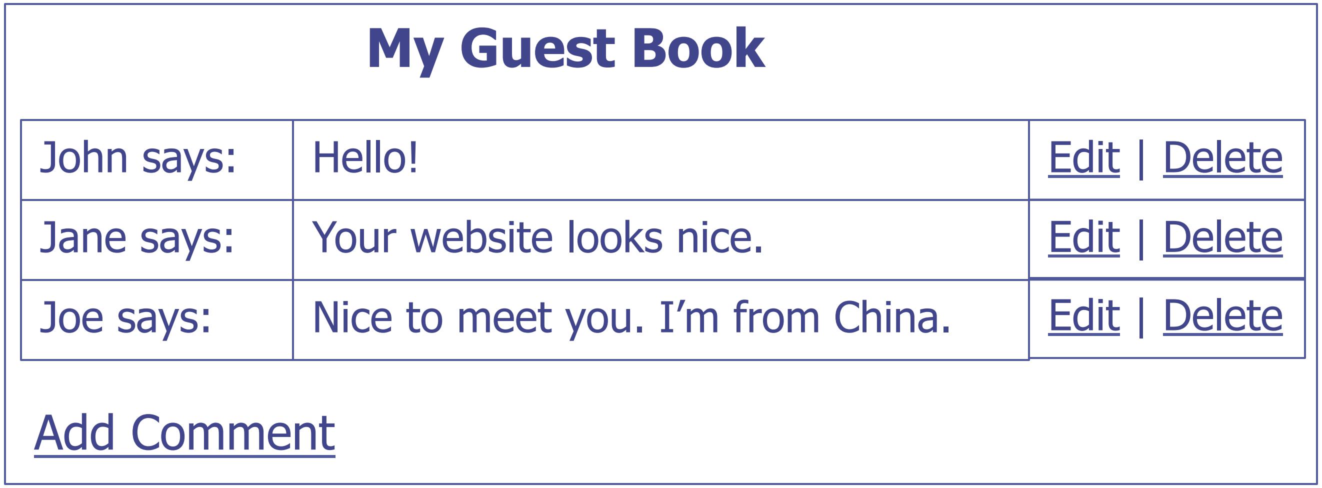
… Basic Usage of React Router

- `<Router>` component listens to URL changes and loads/unloads components
- `<Route>` maps a path to a component
  - `exact` matches the full path instead of just the prefix
- `<Link>` replaces `<a>` so the requests do not go back to the server

Other React Router Components

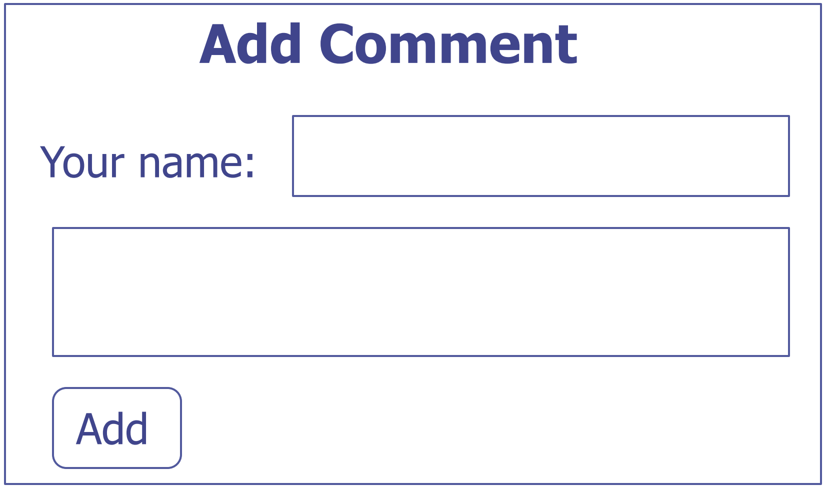
- `<BrowserRouter>` vs `<HashRouter>`
  - BrowserRouter relies on HTML 5 History API which may not be available on older browsers
  - BrowserRouter also relies on the web server configured to serve index.html regardless of the path
  - HashRouter has # in URLs which is aesthetically unpleasing
- `<Link>` vs `<NavLink>`
  - NavLink allow more styling (e.g. activated link)
- `<Switch>` renders the first matched `<Route>` in its children

Guest Book Example



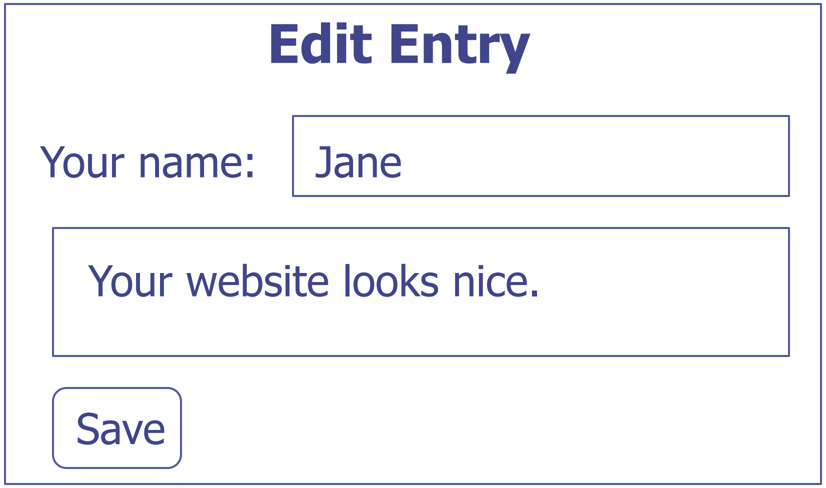
My Guest Book

John says:	Hello!	Edit | Delete
Jane says:	Your website looks nice.	Edit | Delete
Joe says:	Nice to meet you. I’m from China.	Edit | Delete



Add Comment

Your name:



Edit Entry

Your name: Jane

Your website looks nice.

GuestBook Source Code

- REST API in Node.js - https://github.com/cysun/guestbook-node-api
- React SPA - https://github.com/cysun/guestbook-react-spa
  - Uses React Boostrap and hooks

Load Data from Server

- When?
- How?

Component Lifecycle Methods

- Mounting (invoked only once)
  - `constructor()`
  - `render()`
  - `componentDidMount()`
- Updating
  - `render()`
  - `componentDidUpdate()`
- Unmounting (invoked only once)
  - `componentWillUnMount()`

Fetch vs Axios

- Both are Promise-based
- Fetch is a standard Web API
  - May not be available on older browsers
  - Available for React Native
- Axios is a popular JavaScript HTTP client library
  - Treat error status codes as error so it can be handled in `catch()` instead of `then()`

Map GuestBook Entries to Rows

Array map() method converts one array into another

When a list of elements are rendered, each element should a string `key` attribute

JavaScript expression

```
<tbody>
{
   entries.map(entry => (
      <tr key={entry.id.toString()}>
         <td>{entry.name}</td>
         <td>{entry.message}</td>
      </tr>
   ))
}
</tbody>
```

Forms Are Special ...

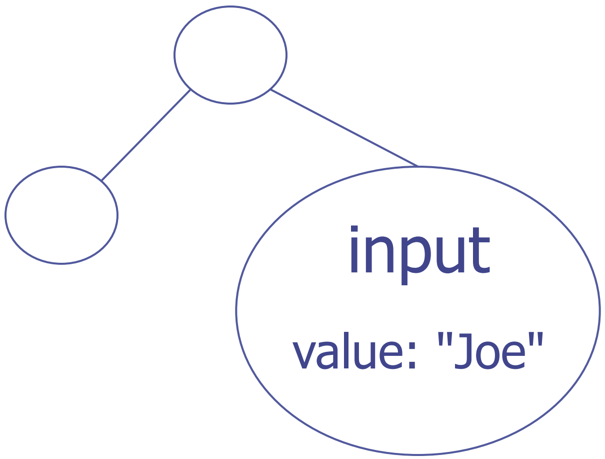
```
this.state = { name: "Joe" };

<input type="text" value={this.state.name} />
```

React

state: { name: "Joe" }

DOM



… Forms Are Special

- Form elements keep their internal states in DOM (e.g. `value` for `<input>`)
- The data kind of goes from view → state instead of the state → view
- React only allows *one-way binding*

Two Ways to Deal with Forms

- Use ref to reference DOM elements and get their values
- Controlled Components (recommended)
  - Handle `onChange` event
  - Set state in event handler

```
<input type="text"
   onChange={ event => this.setState(
      {name: event.target.value}) } />
```

Submit a Form

- Handle the `onSubmit` event of the form, e.g.

```
<form onSubmit={
    event => {
        event.preventDefault();
        axios.post("/api", {
            this.state.name,
            this.state.message
        })
        .then( ()=>history.push("/") );
    }
}>
```

Event handler

Prevent browser from submitting the form

"Redirect" to /

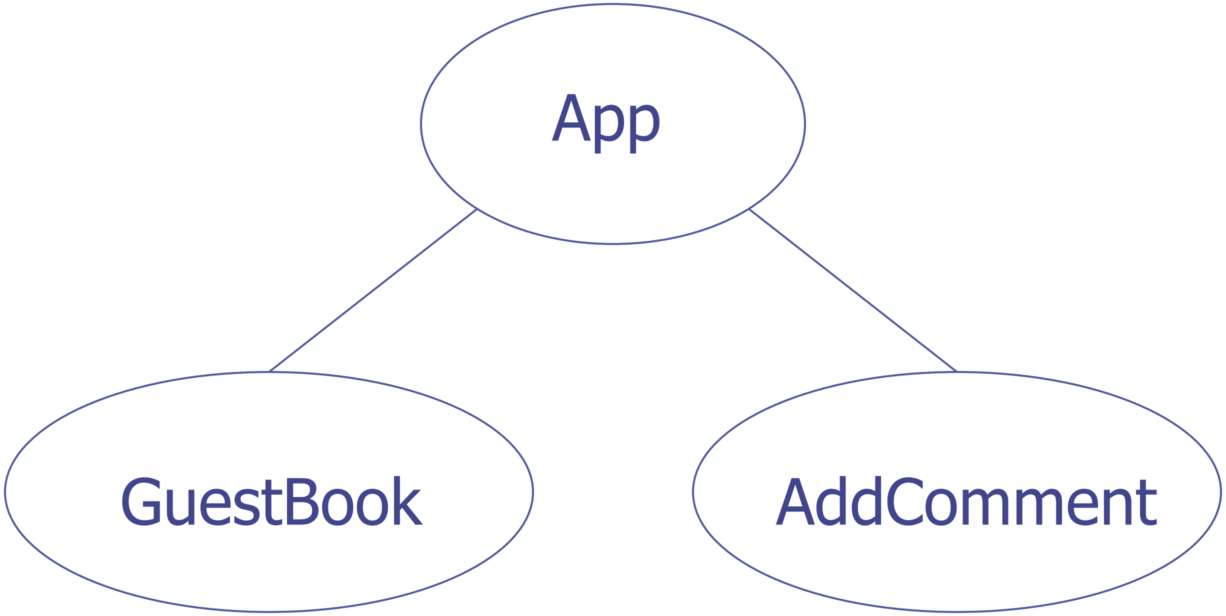
A hook provide by React Router

Keeping Data in SPA …

- It's inefficient to query the server every time the GuestBook component is mounted
- Maybe we want to keep a local copy of GuestBook in the SPA (the data may get stale but it's more efficient)

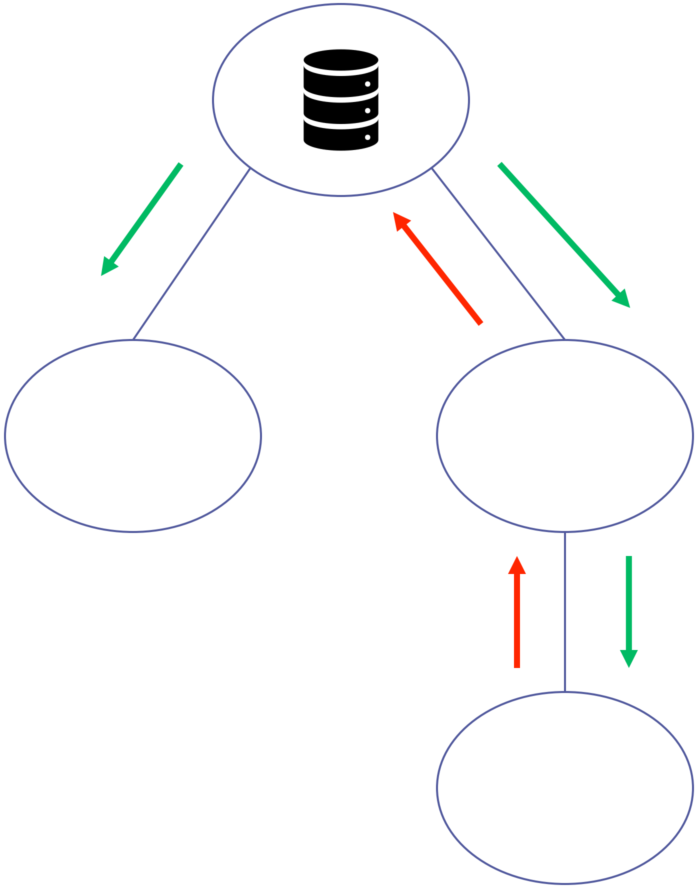
… Keeping Data in SPA

- Where the data should be stored?

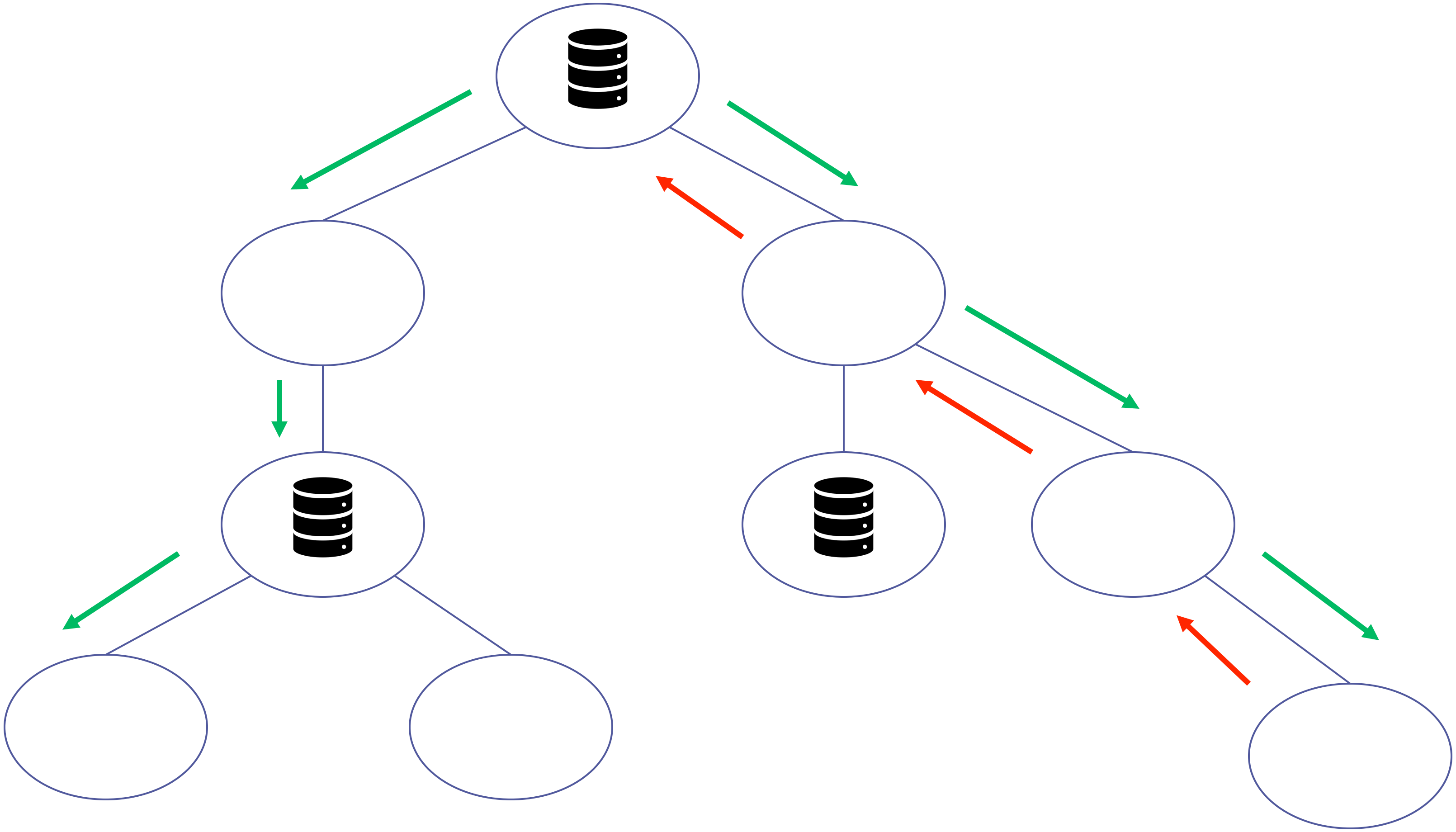


Basic State Management in React

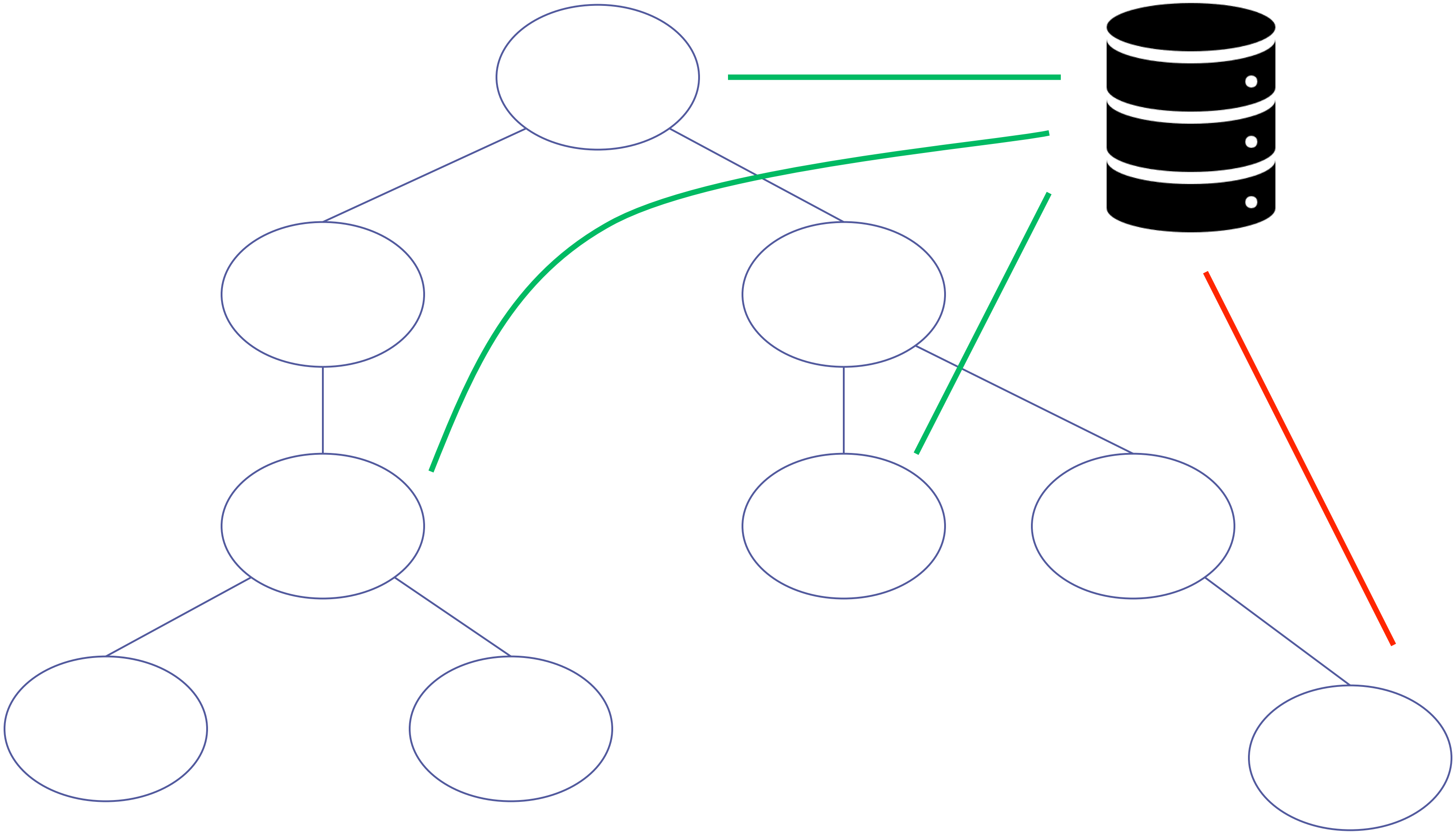
- States stored in *root component*
- Parent passes data down to children as *properties*
- Parent also passes callback functions to children as properties, and children can use those callbacks to send changes back up



State Management Can Get Difficult and Messy



Store States in One Place



State Management Solutions

- React Context API
- Redux
  - More mature and performant
  - More features and tool support

Some React Resources

- React Bootstrap
- React Icons
- Awesome React Components
- bit.dev for publishing, sharing, and searching components

Readings

- React
- React Router
- Redux and React Redux