Abdullah S.

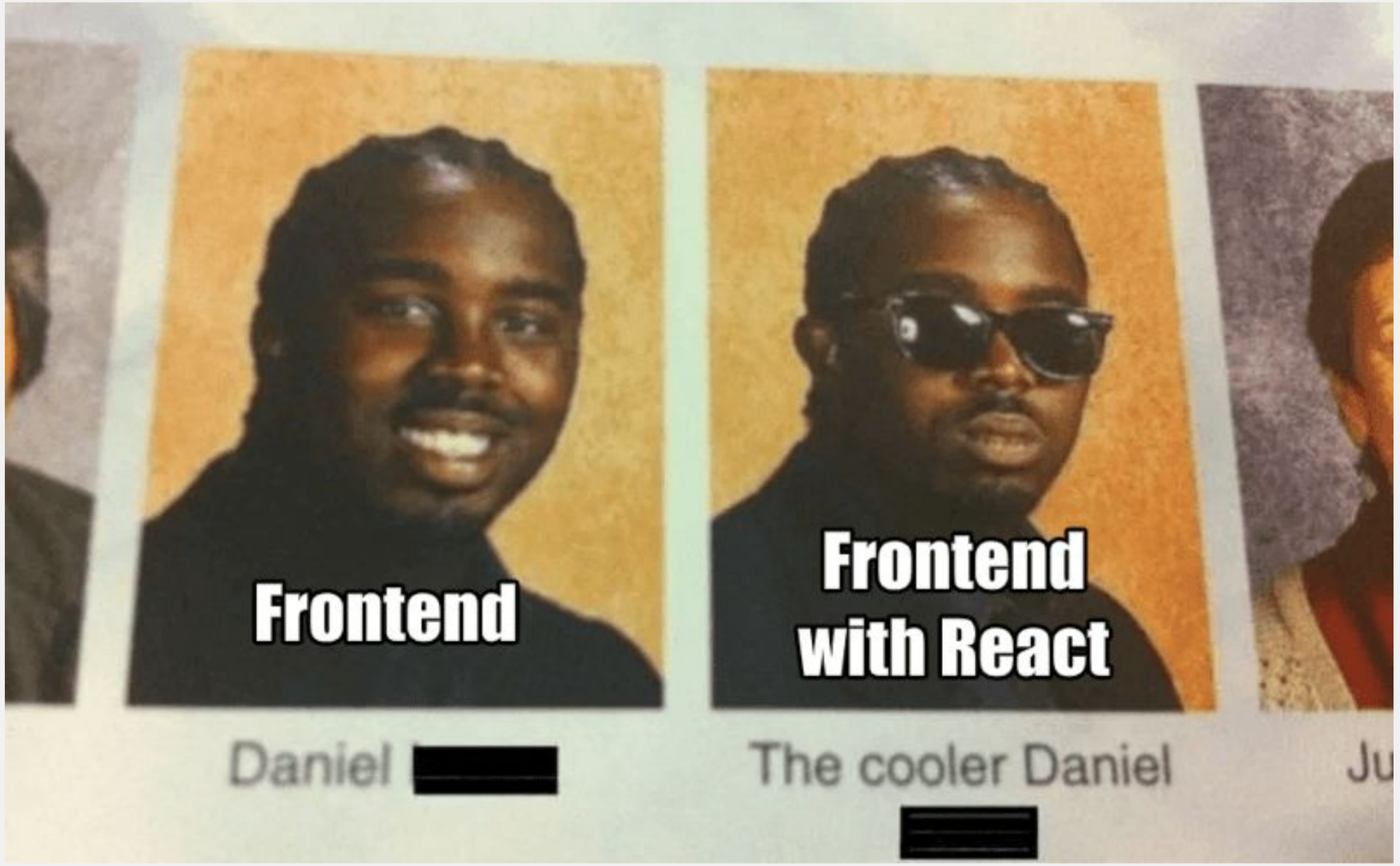
CSC309 Week 7

React, Frontend Testing (Unit, E2E)

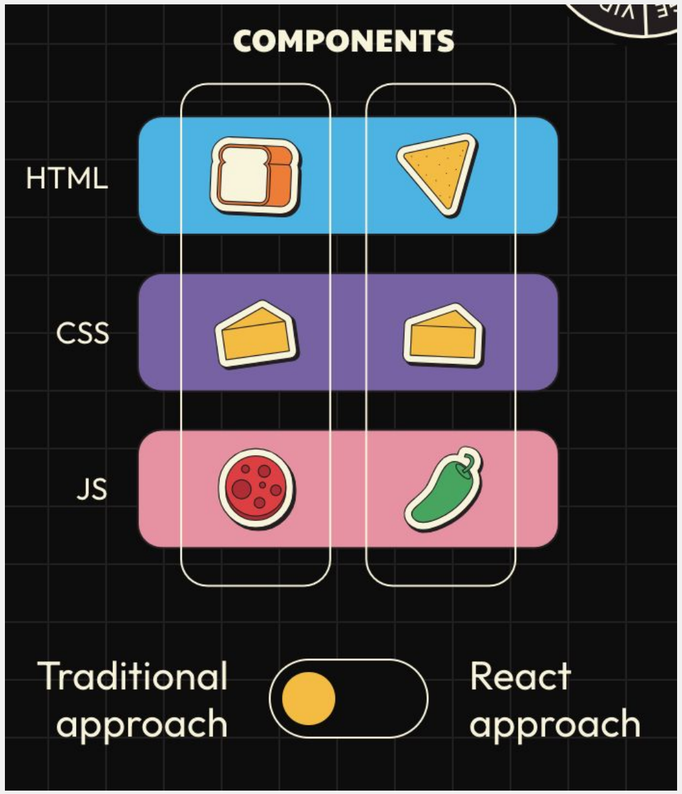
HTTPS://WWW.CS.TORONTO.EDU/~KIANOOSH/COURSES/CSC309H5/

26 FEBRUARY, 2025

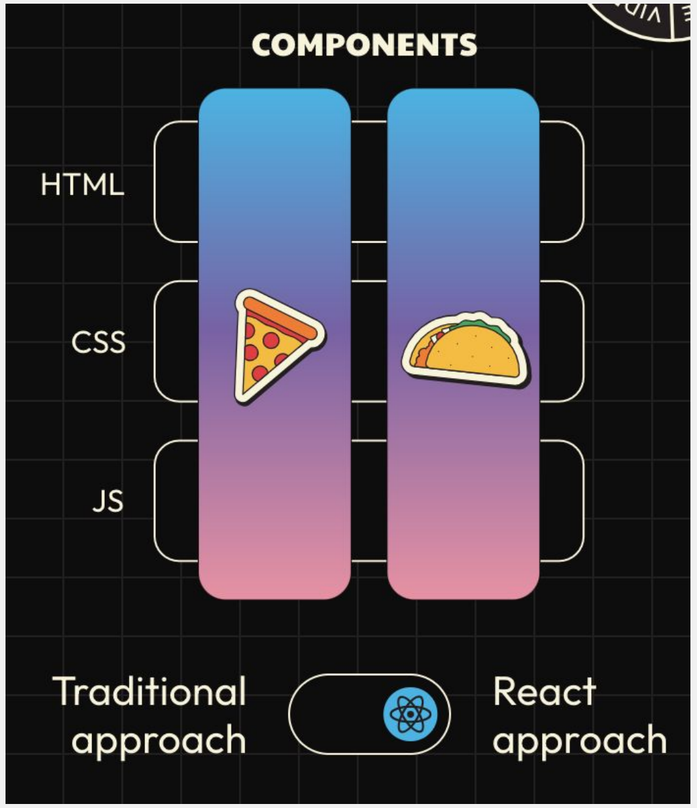
React


Frontend
Frontend with React
Daniel [redacted]
The cooler Daniel
[redacted]
Ju

React: High Level Review



Traditionally

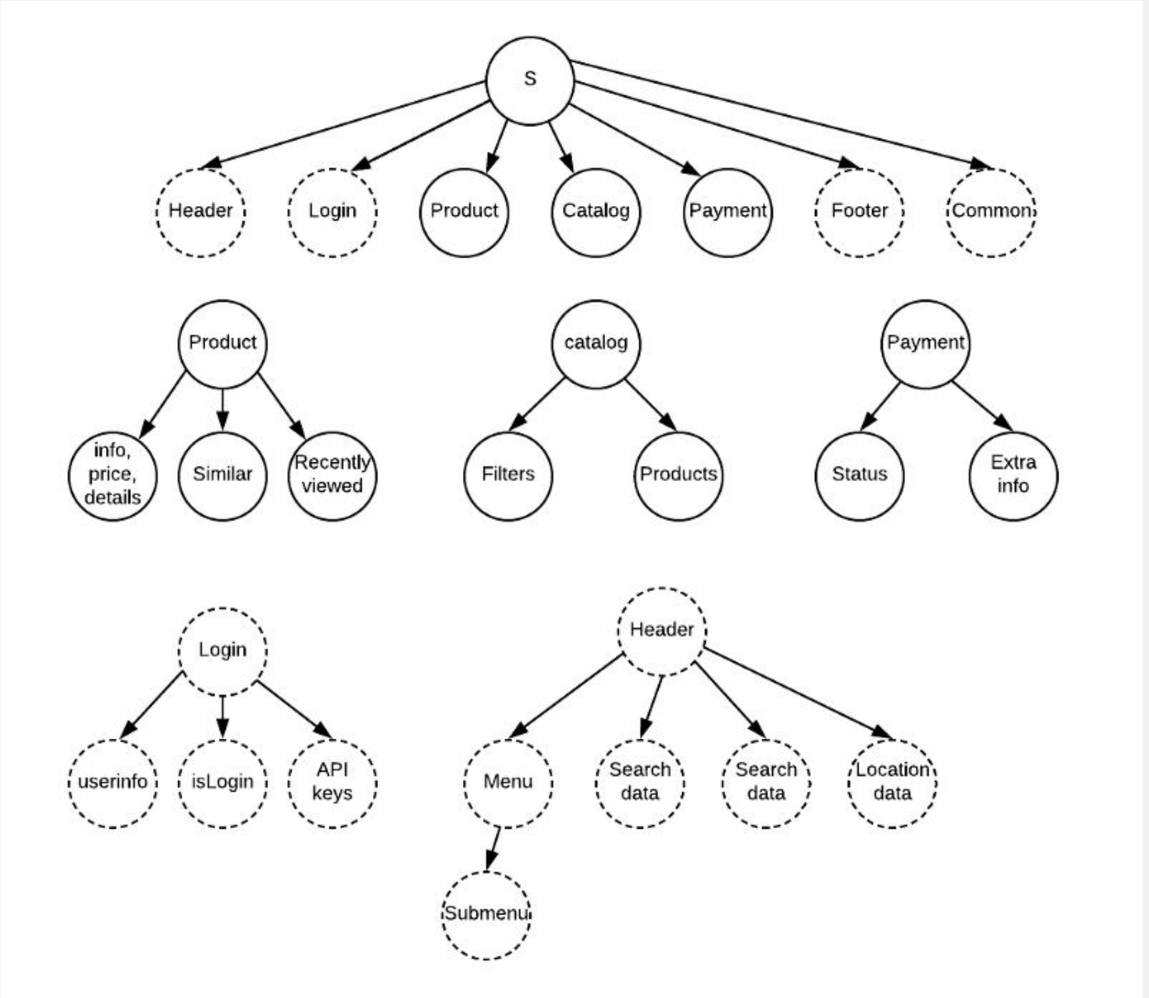


React

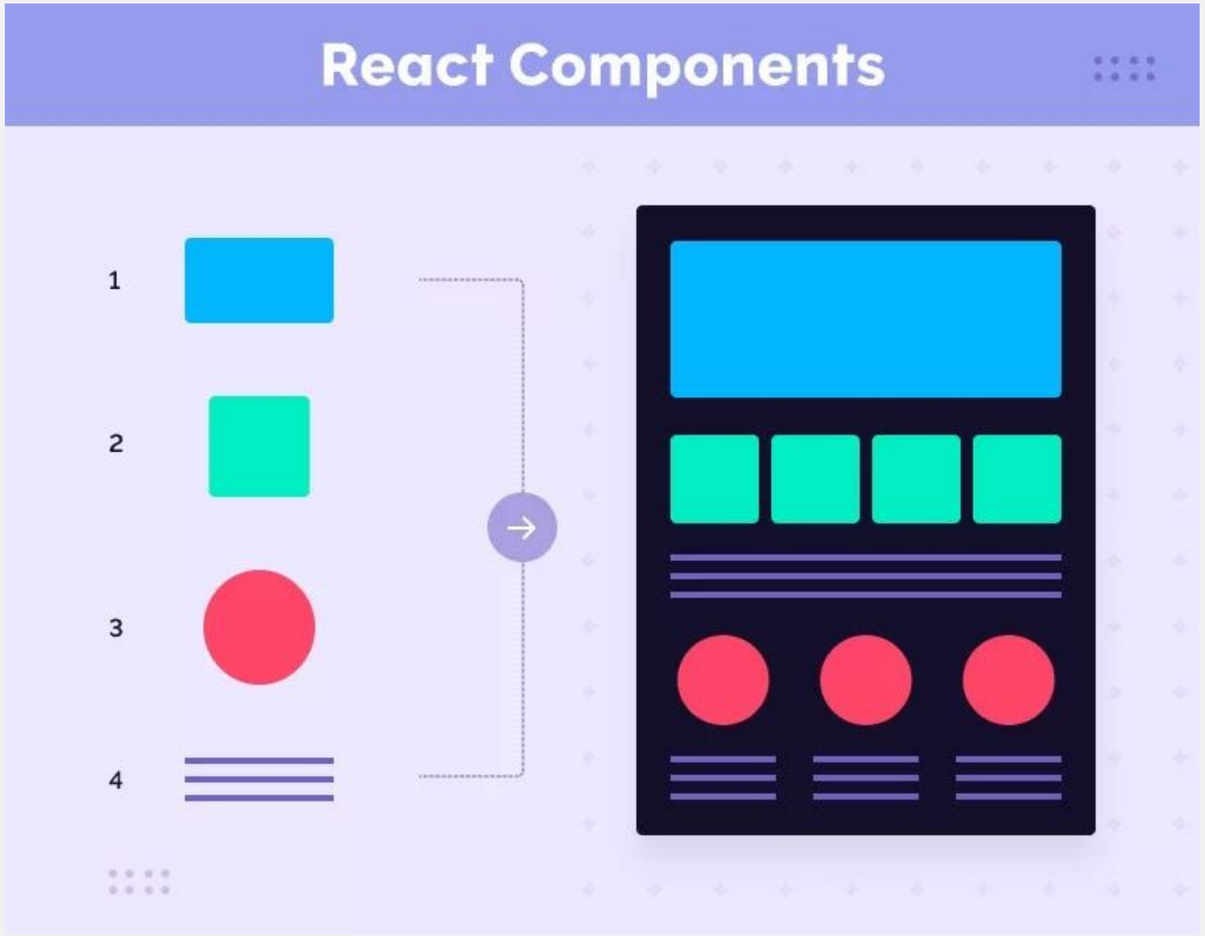
Visualizing React Apps

Think of a **tree** of “components”

Components can be used to create other components (powerful)



Visualizing React Apps



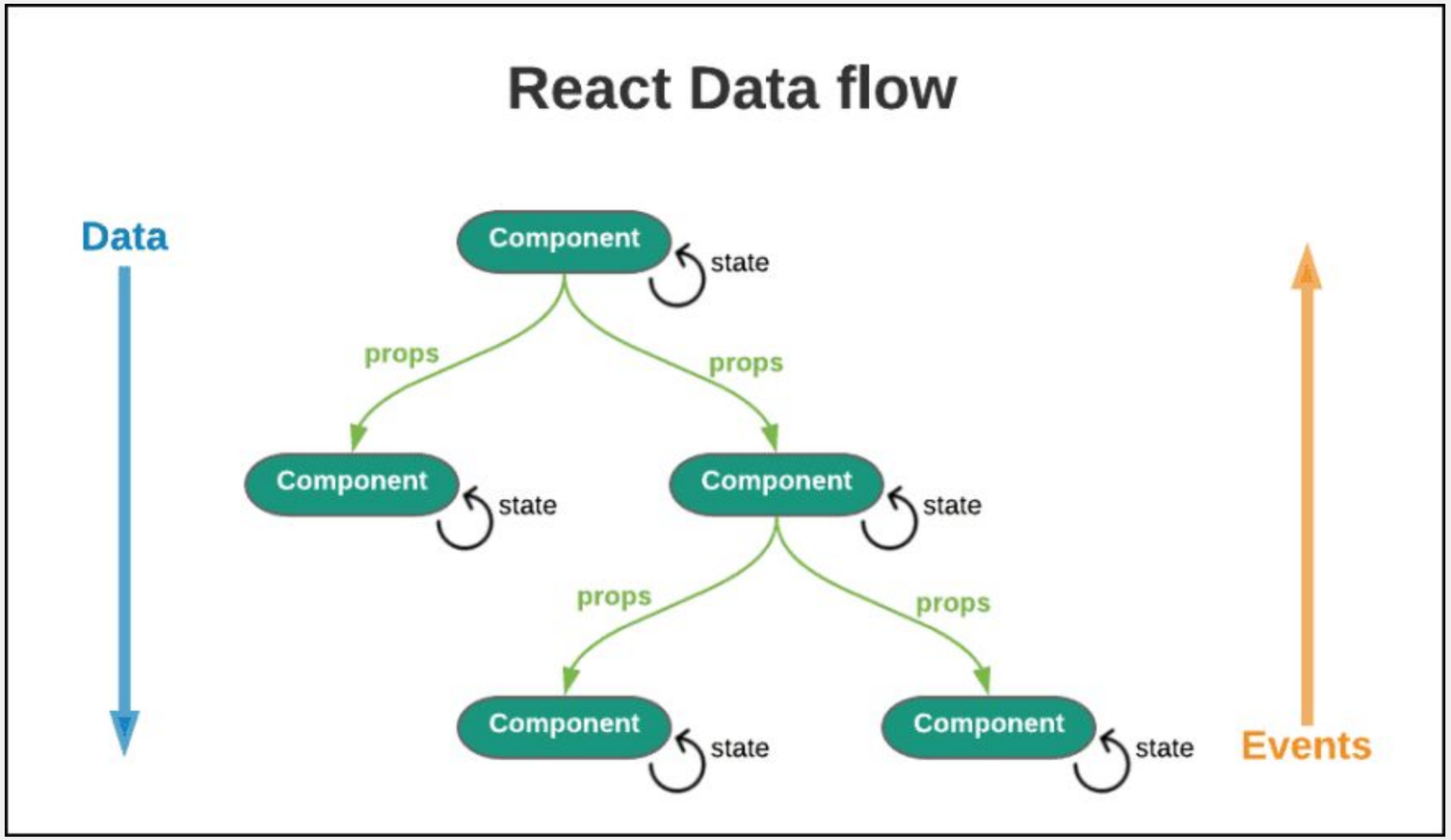
Recall: State variables

What are they?

React state variables are variables used within a React component to hold data that can change over time.

When these variables are updated, React re-renders the component to reflect the new data. In function components, state is typically managed using the useState hook, which provides a current state value and a function to update it.

React: Data Flow



React: Local vs Global State

TYPES OF STATE: LOCAL VS. GLOBAL STATE

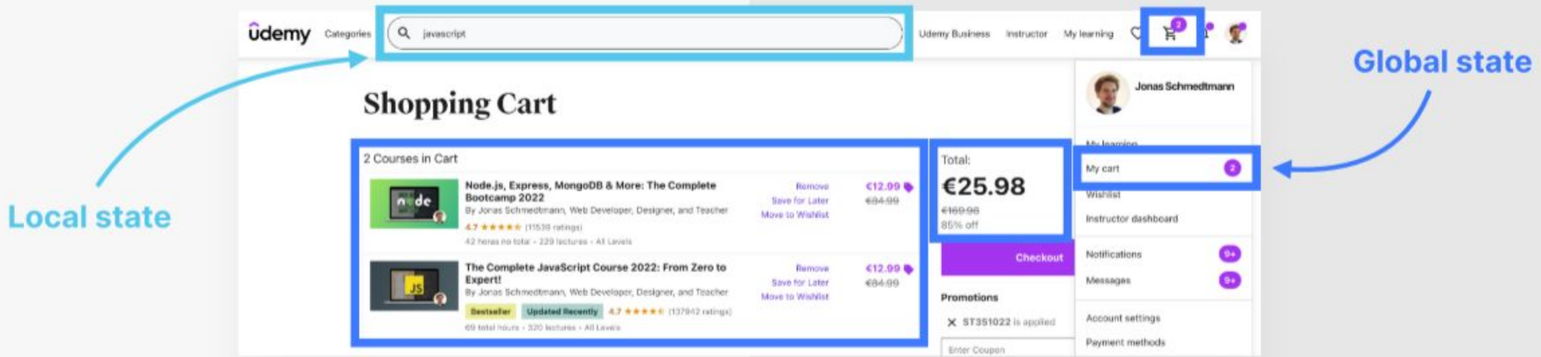
LOCAL STATE

- 👉 State needed **only by one or few components**
- 👉 State that is defined in a component and **only that component and child components** have access to it (by passing via props)
- ☝️ ***We should always start with local state***

GLOBAL STATE

- 👉 State that **many components** might need
- 👉 **Shared** state that is accessible to **every component** in the entire application

Context API Redux



State management can get complex and is ESSENTIAL to good frontend engineering

State management: level 1

When it comes to managing state, you have a bunch of options…

For starters, you can use prop drilling..

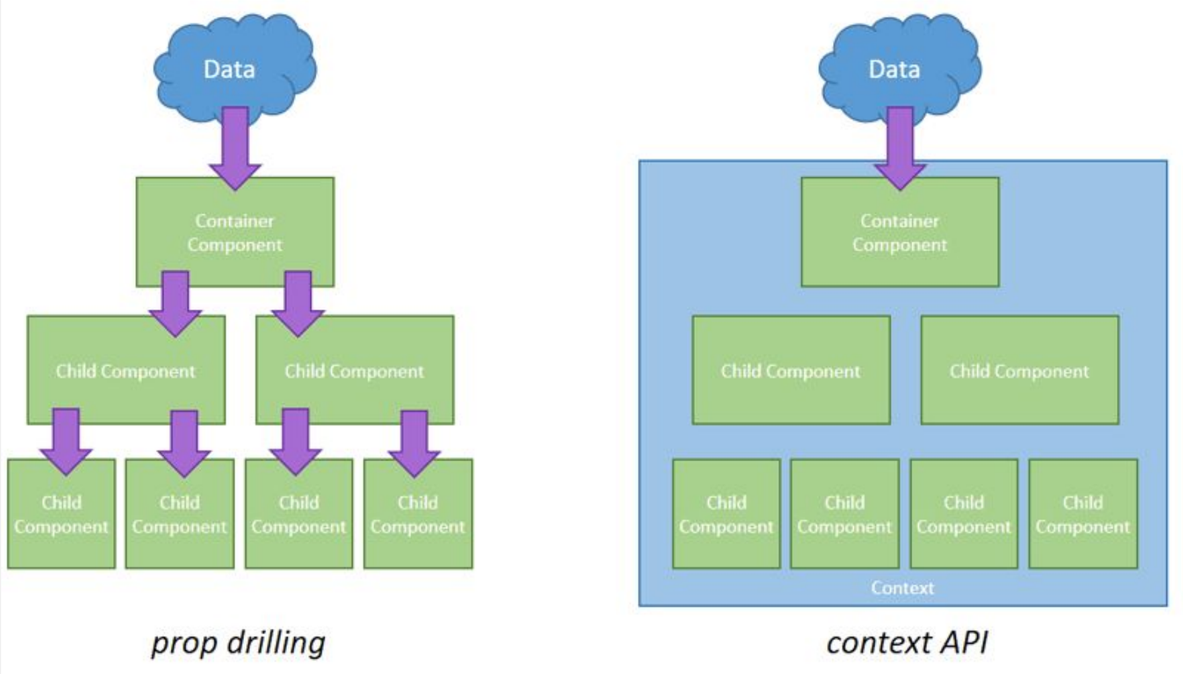
i.e define the state in a higher component and pass it down to the components that need it.

Obviously, not the cleanest approach; also hard to debug and reason with.

State management: level 2

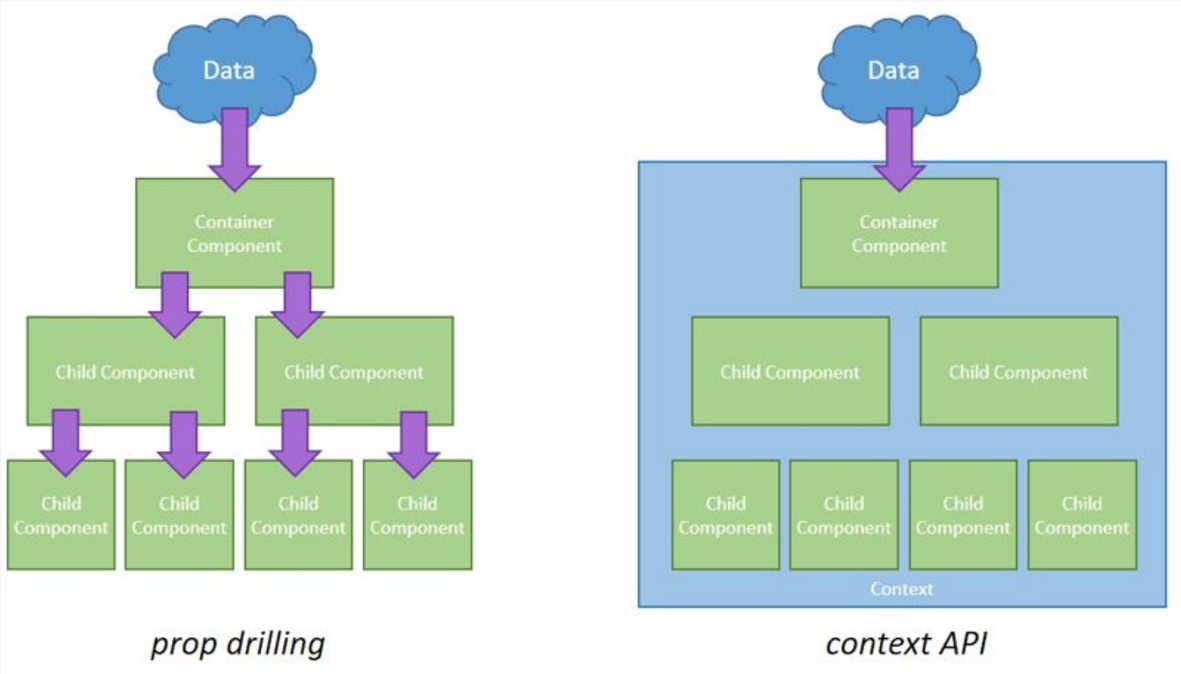
To level it up, you can use the `useContext` hook / API that comes with React.

This will let you define a global state where you can store your values



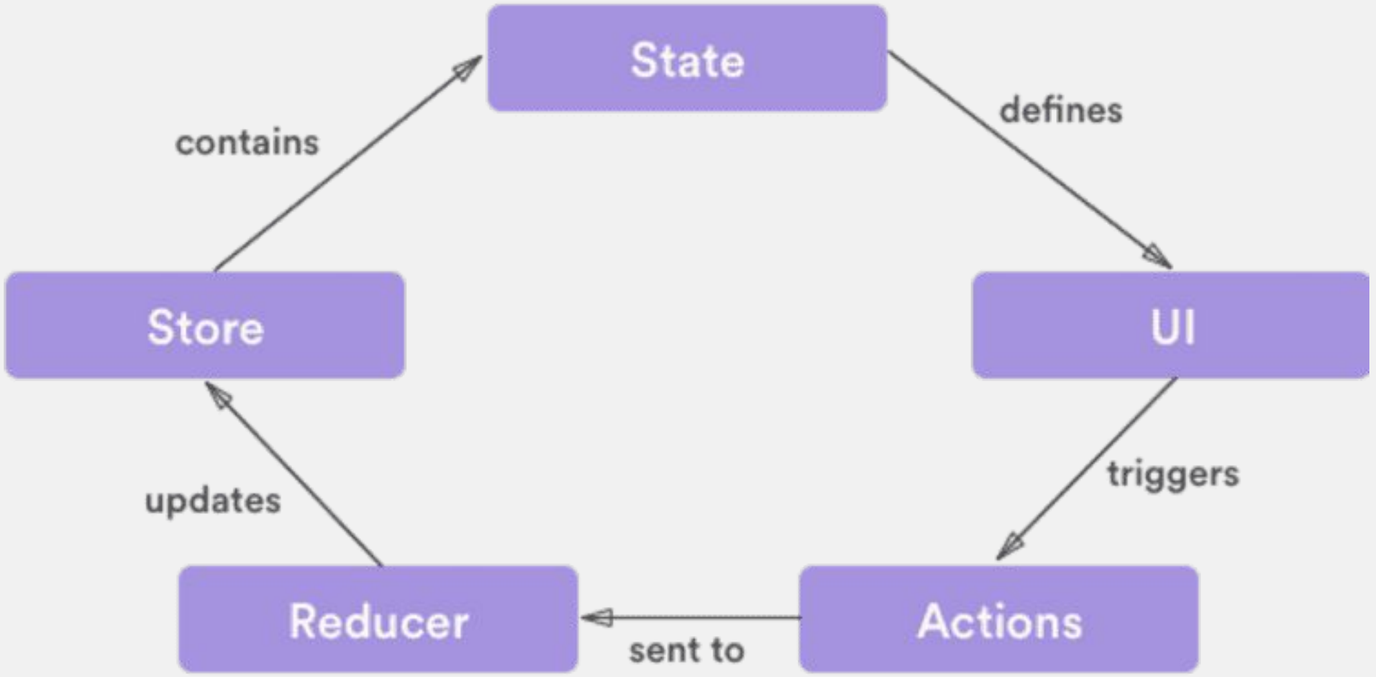
State management: level 2

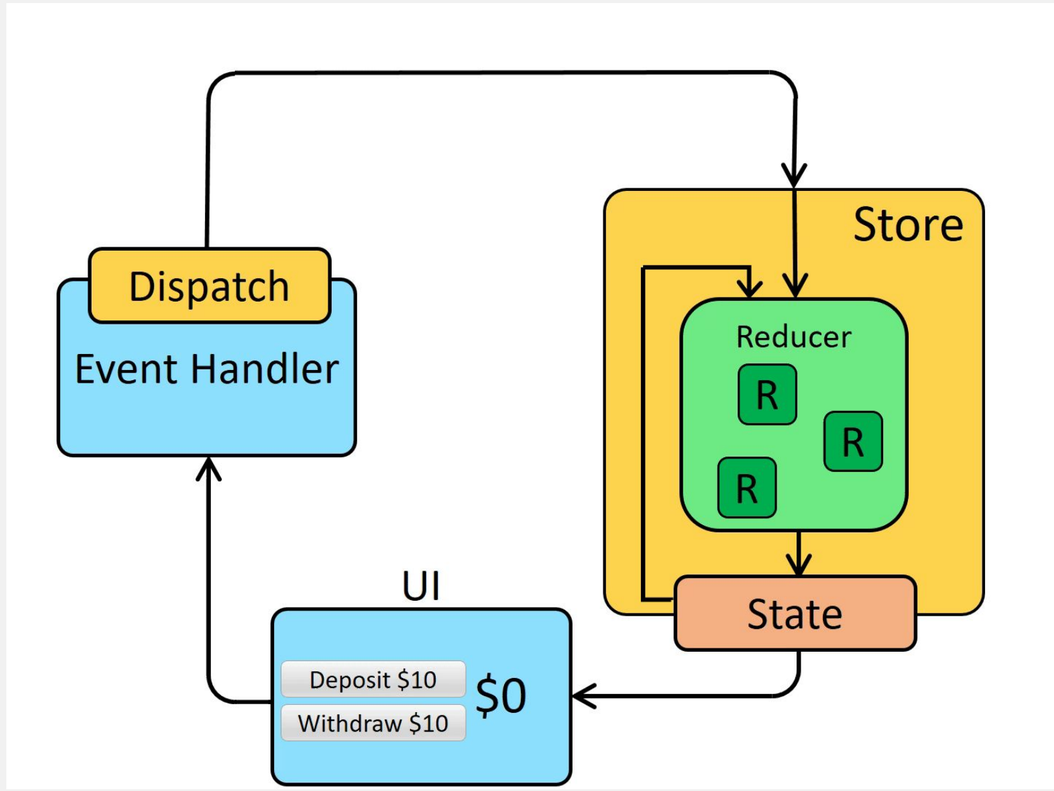
Can you see the problem with context? Think of the data flow…



State manage: Level 3

Commonly used at the enterprise level (for good reasons)





Redux design pattern for state management
(React Redux Toolkit)

State manage: alternatives

Though Redux is great, and though it is fairly easier to implement nowadays with the help of libraries such as Redux Toolkit,

it may not be worthwhile for projects at a smaller scale…

You have some alternative tools that can provide you efficient state management with less complexity:

- Zustand, Recoil, Nanostores, and many more.

I recommend Zustand

Client vs Server State

We differentiated between global and local state… but there is another important distinction.

Client state: managed entirely on the client (browser) and typically stored in memory within React components. This includes data that affects the UI and is transient or local to a component.

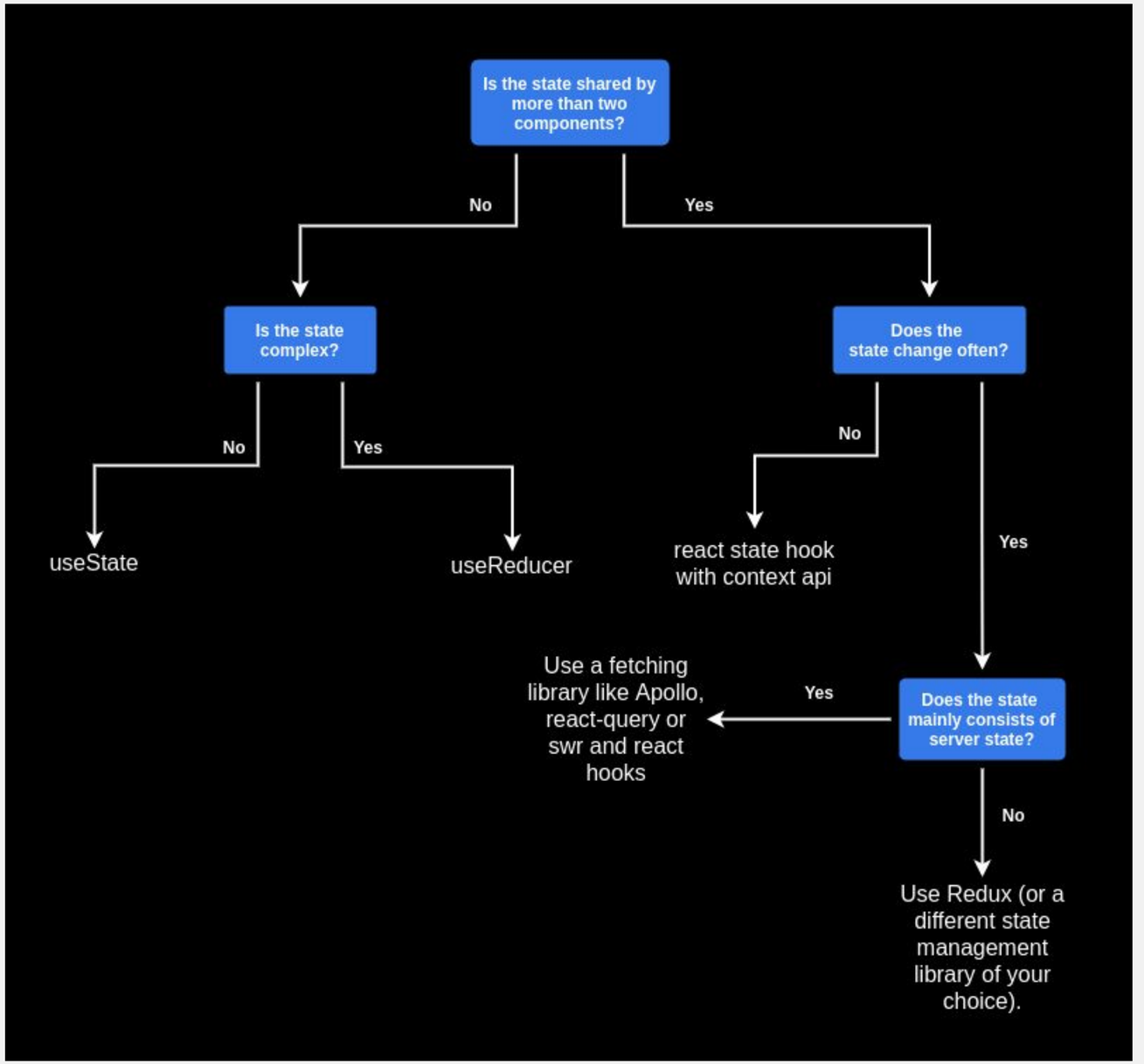
Ex: Form inputs, modal visibility, toggles, local caching of user interactions.

Server state: Resides on a remote server (or backend) and is accessed by your React app via API calls. This state represents data that is shared across users and maintained centrally (like a database).

Ex: User profiles, posts, comments, or any data fetched from an API.

Challenges: Synchronization, Caching and invalidation, error handling

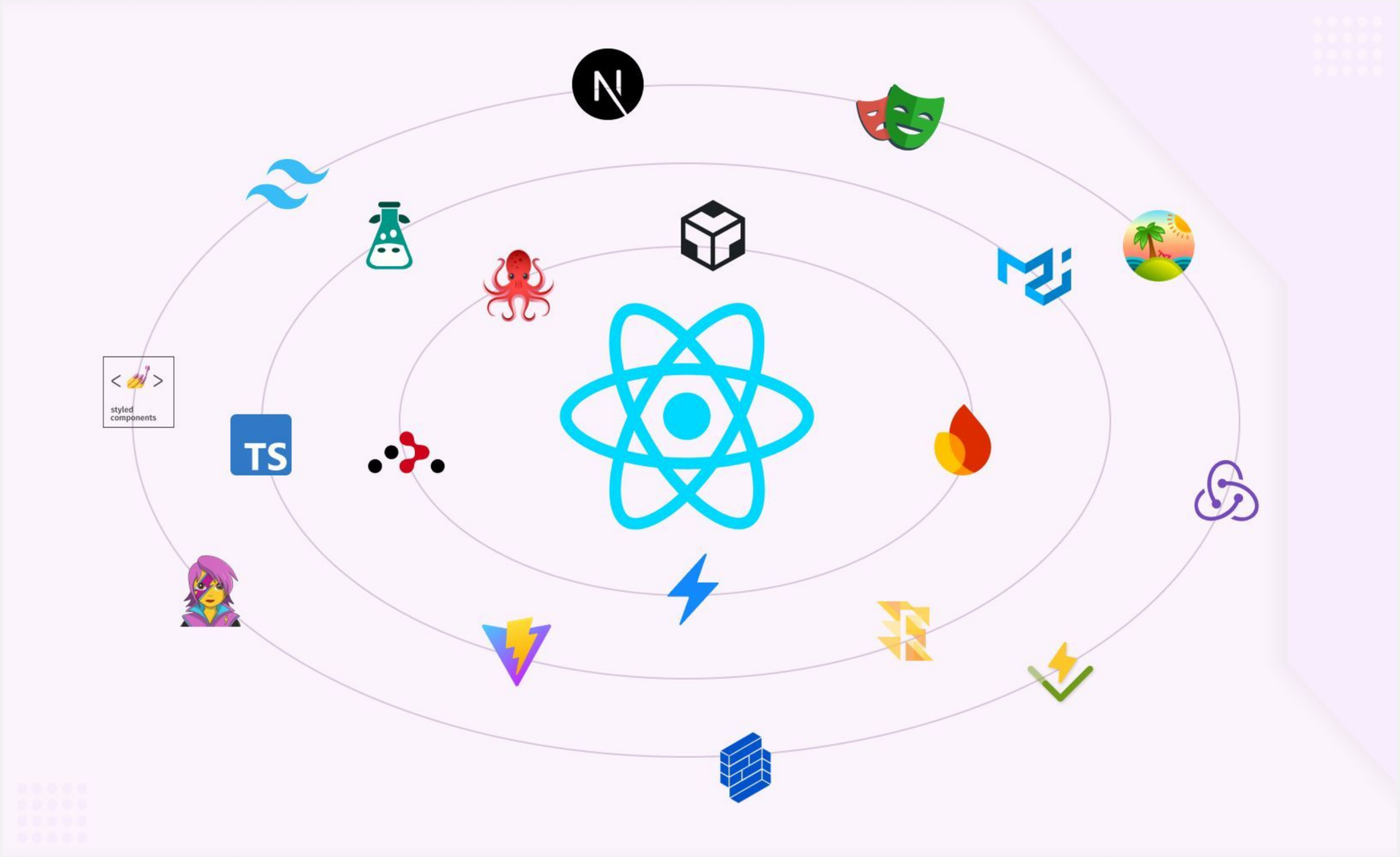
This can get even more complex!


Is the state shared by more than two components?
No
Yes
Is the state complex?
Does the state change often?
No
Yes
No
Yes
useState
useReducer
react state hook with context api
Use a fetching library like Apollo, react-query or swr and react hooks
Yes
Does the state mainly consists of server state?
No
Use Redux (or a different state management library of your choice).

As you can see, state management can get complex

Thinking about your state design earlier for your application can help with development!

React Ecosystem



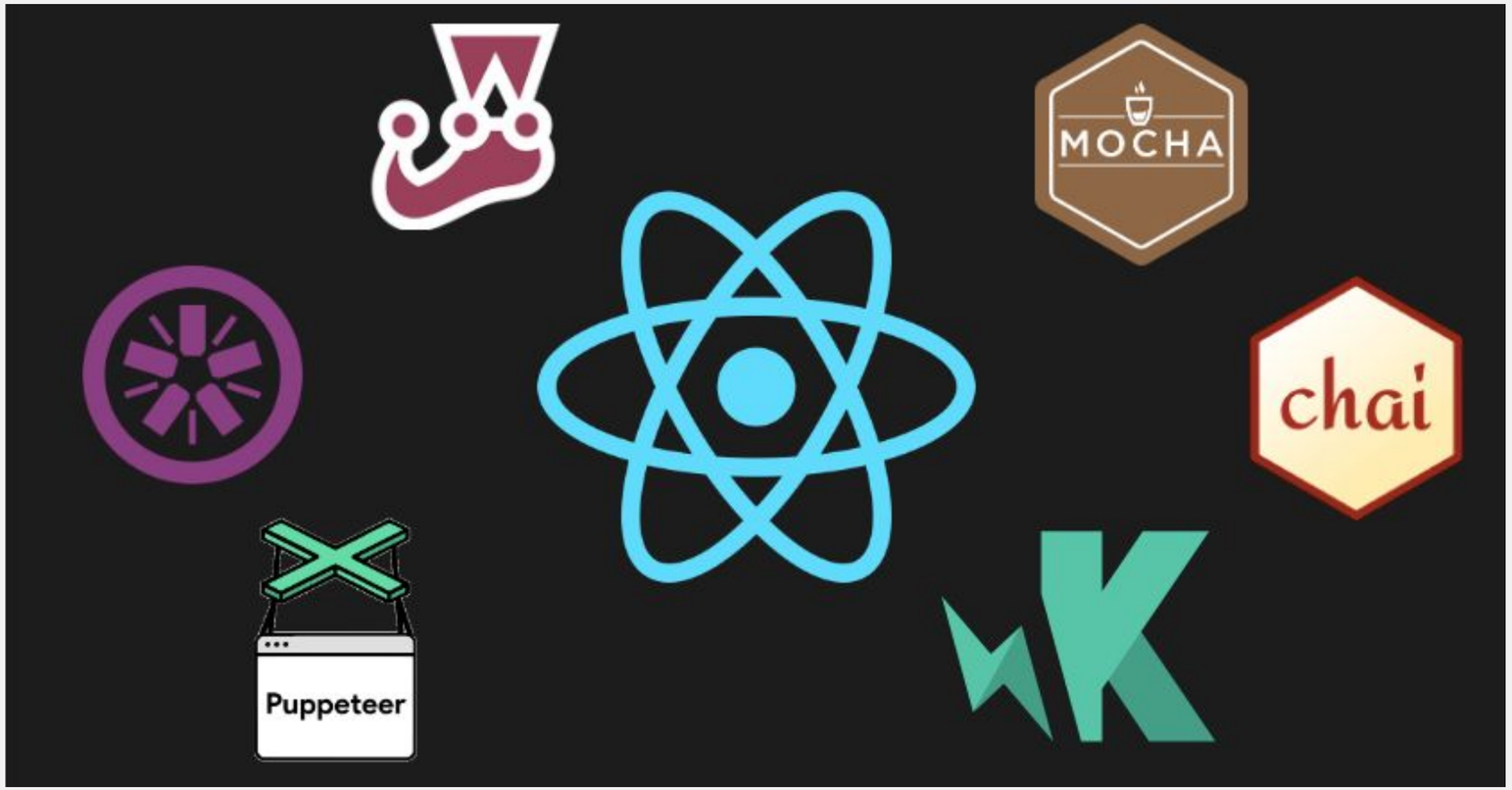
Testing...

Writing unit tests

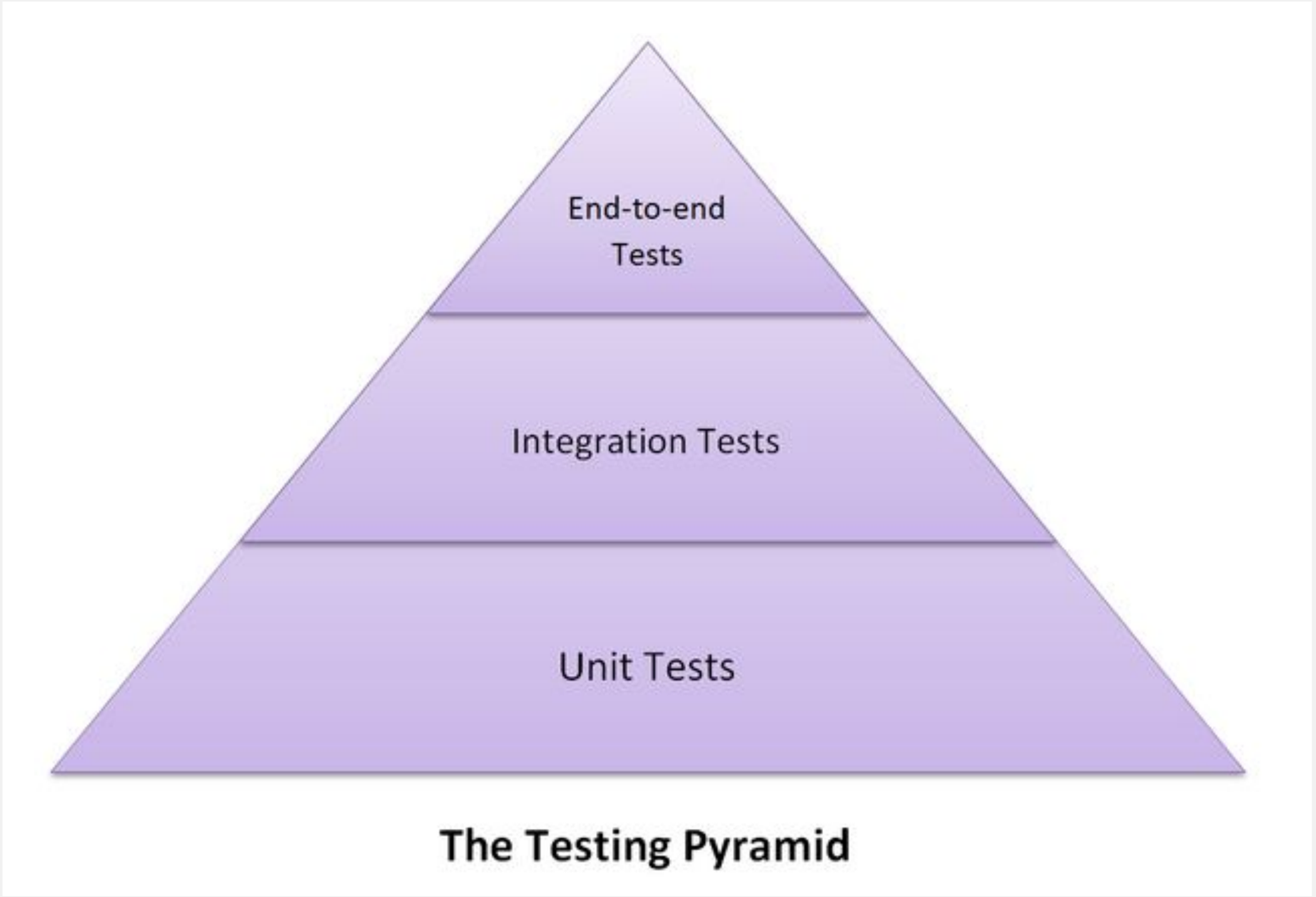
Pros:	Cons:
-they'll improve the quality of my code -it'll take like 10 mins max -literally everyone says that I should	-i don't wanna

CONCLUSION: i will not write unit tests

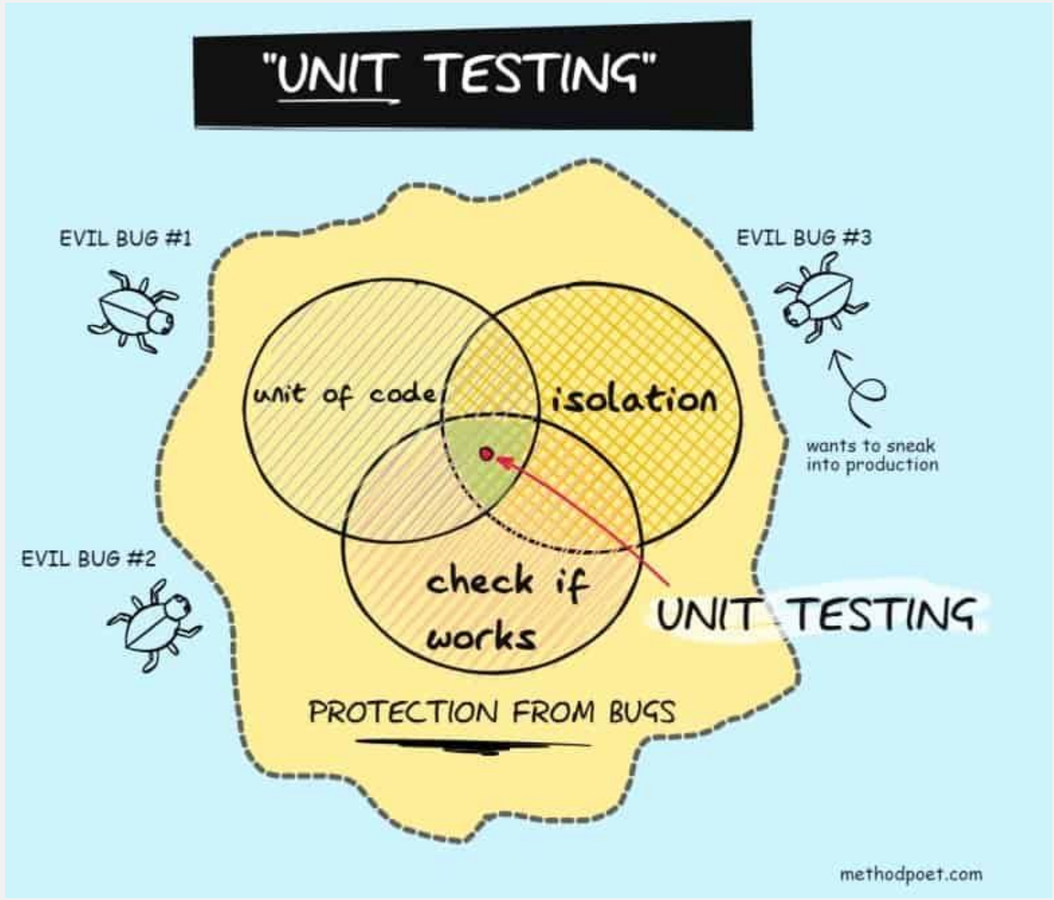
A ton of libraries available..



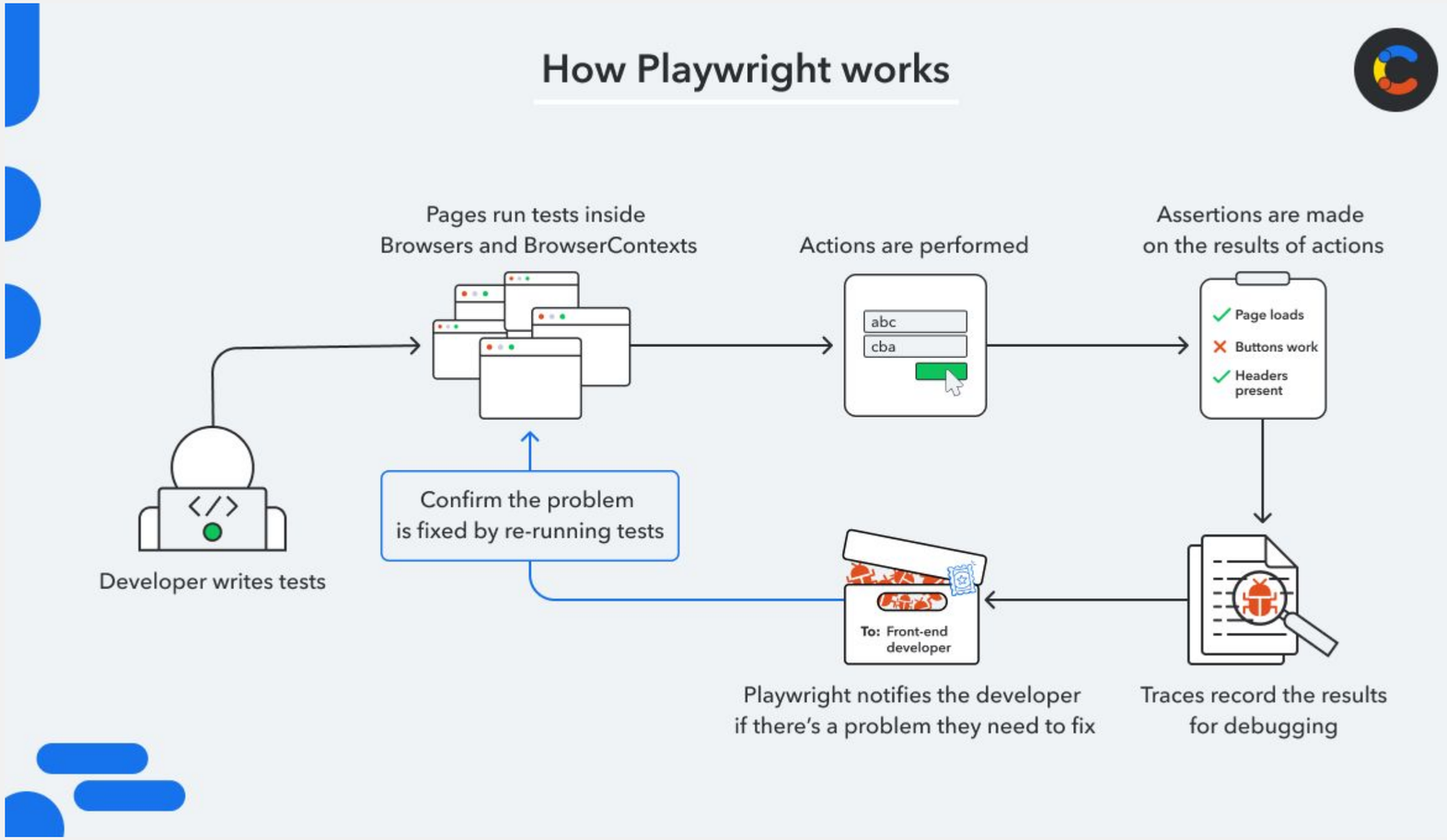
The testing pyramid



Unit testing



End to End (E2E) tests



Exercise

Remember this?

Question 2: Web in the Nineties! [24 marks]

Analyze the following web page and answer the questions based on its behavior. Assume the code works as intended – do not look for syntax or other potential errors.

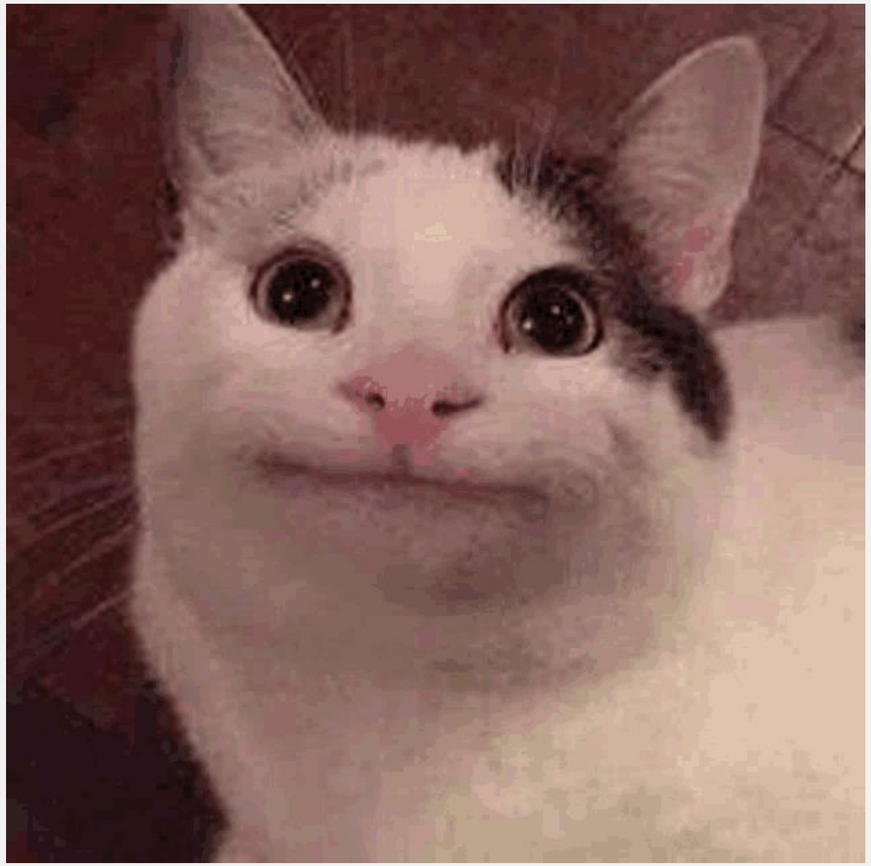
```
<!DOCTYPE html>
<html>
  <head>
    <title>Midterm Example</title>
    <script>
      var greetingText = "Hello from JavaScript!";

      function showGreeting() {
        document.getElementById("greeting").innerText = greetingText;
      }

      var count = 0;
      function incrementCounter() {
        count++;
        document.getElementById("counter").innerText = count;
      }
    </script>
  </head>
  <body>
    <main>
      <section>
        <p id="greeting">Click the button to see a greeting.</p>
        <button id="greetingButton" onclick="showGreeting()">
          Show Greeting
        </button>
      </section>

      <section>
        <p>Counter Value: <span id="counter">0</span></p>
        <button id="counterButton" onclick="incrementCounter()">
          Increment Counter
        </button>
      </section>
    </main>
  </body>
</html>
```

Exercise

Follow along with me

We will build the same application in React and write some unit tests…

Link to the midterm



Challenge

Now, challenge for you.

Implement **part d** of the question

And write a unit test for it.

Once done, email me your submission for tracking purposes:
abdullah.shahid@mail.utoronto.ca

ABDULLAH

Thank You

CSC309 Week 7

26 FEBRUARY, 2025