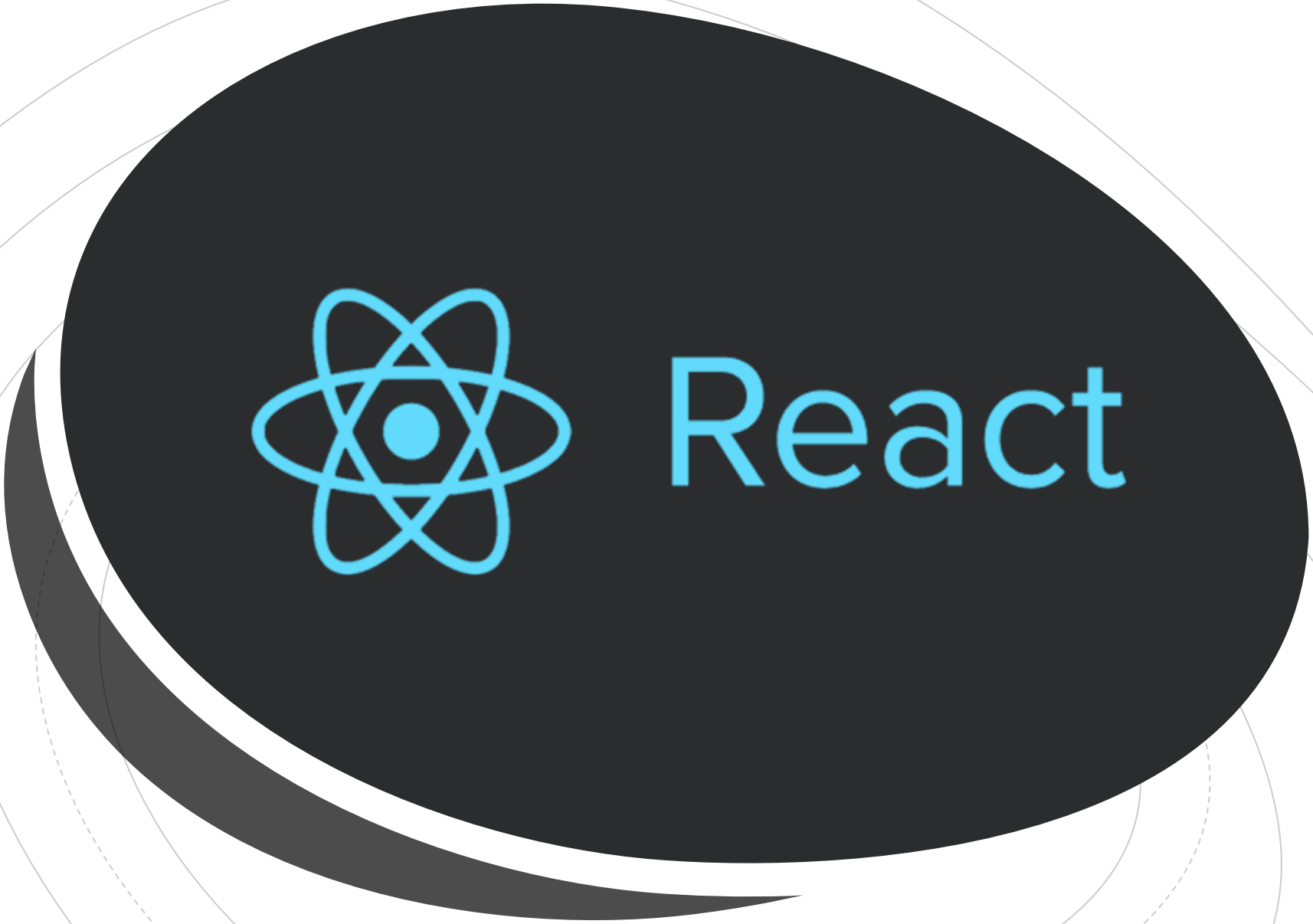


Full Stack Developmen t - I

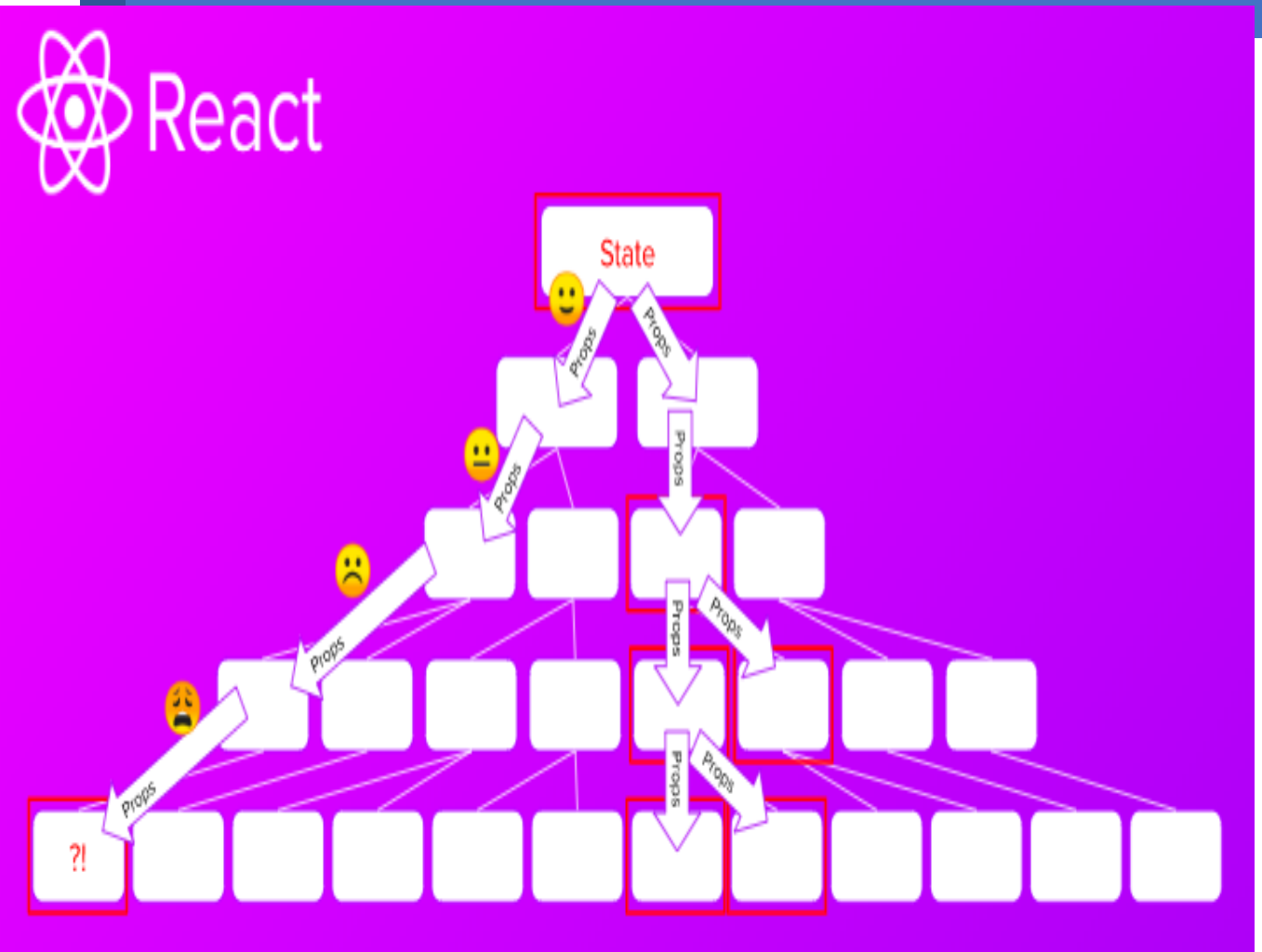
COMP 3123

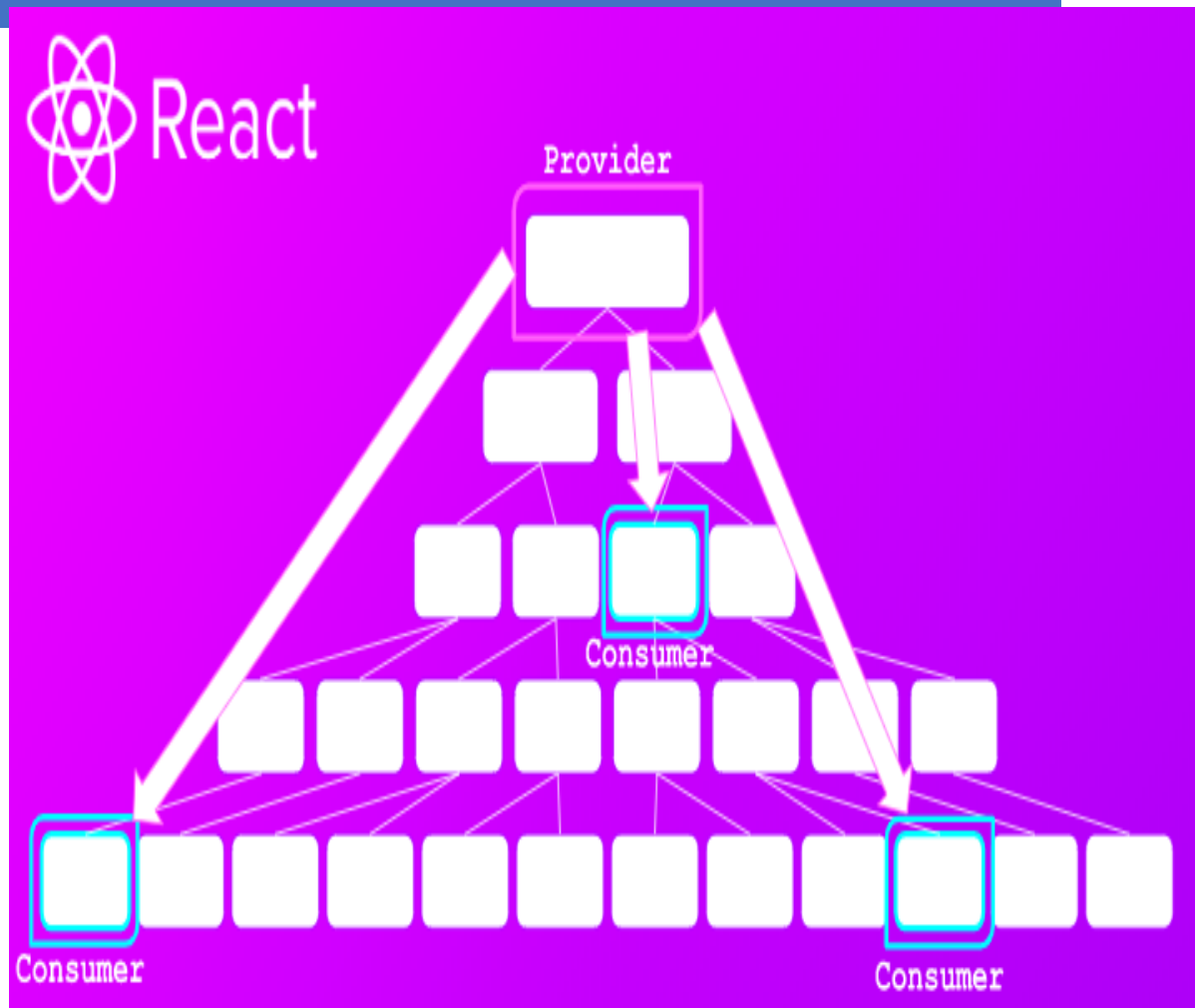
React- Context API

- Context API is a (kind of) new feature added in version 16.3 of React that allows **one to share state across the entire app (or part of it) lightly and with ease**.
- In a typical React application, **data is passed top-down (parent to child) via props**, but this can be cumbersome for certain types of props (e.g. locale preference, UI theme) that are required by many components within an application.
- ***Context provides a way to share values like these between components without having to explicitly pass a prop through every level of the tree.***
- React Context API is a way to essentially **create global variables** that can be passed around in a React app.
- This is the alternative to "**prop drilling**", or passing props from grandparent to parent to child, and so on.

Web Reference Link

React- Context API





Create Context

React.createContext() is all you need. It returns a consumer and a provider.

- **Provider**
  - The component that provides the value
  - Component that as it's names suggests provides the state to its children.
  - It will hold the "store" and be the parent of all the components that might need that store.
- **Consumer**
  - A component that is consuming the value
  - Consumer as it so happens is a component that consumes and uses the state.

Create Context (UserContext.js)

```
import React from 'react'
const UserContext = React.createContext()
export const UserProvider = UserContext.Provider
export const UserConsumer = UserContext.Consumer
export default UserContext
```

Providing Context (App.js)

The provider always needs to exist as a wrapper around the parent element, no matter how you choose to consume the values. Here we will wrap the entire App component in the Provider. Now creating some value (user) and passing it down as the Provider value prop.

```
import React from 'react'
import HomePage from './HomePage'
import { UserProvider } from './UserContext'
function App() {
  const user = { name: 'Tania', loggedIn: true }
  return (
    <UserProvider value={user}>
      <HomePage />
    </UserProvider>
  )
}
```

Consuming Context (HomePage.js)

The most common way to access Context from a class component is via the static contextType. If you need the value from Context outside of render, or in a lifecycle method, you'll use it this way.

```
import React, { Component } from 'react'
import UserContext from './UserContext'
class HomePage extends Component {
  static contextType = UserContext
  componentDidMount() {
    const user = this.context
    console.log(user) // { name: 'Tania', loggedIn: true }
  }
  render() {
    return <div>{user.name}</div>
  }
}
```

Thank You