جامعة التقنية والعلوم التطبيقية
University of Technology and Applied Sciences

LESSON 3-Redux Fundamentals

CSSE4103 – FULL STACK DEVELOPMENT

What is a State in ReactJS?

- The **state** is a built-in **React object** that is used to **contain data or information about the component**. A <u>component’s state</u> can change over time; whenever it changes, the <u>**component re-renders.**</u>

What is State Management?

- As your application grows, it helps to be more intentional about how your state is organized and how the data flows between your components. State management in React refers to the process of managing and controlling the state of a React application.

The Four Kinds of React State to Manage

- There are four main types of state you need to properly manage in your React apps:

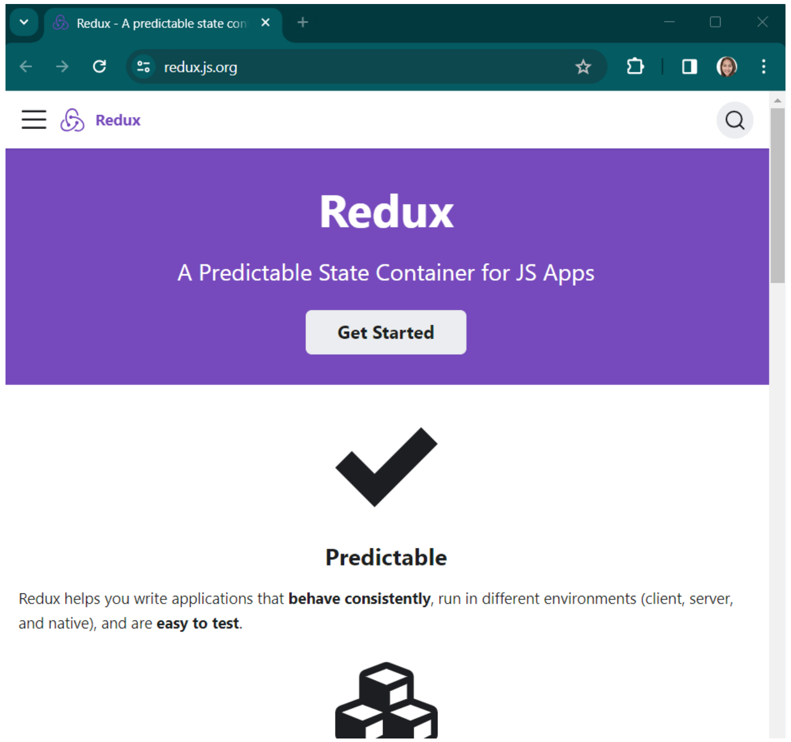
1. **Local (UI) state** – Local state is data we manage in one or another component. Local state is most often managed in React using the useState hook.
2. **Global (UI) state** – Global state is data we manage across multiple components. Popular state management libraries include Redux, MobX, and the Context API in React.
3. **Server state** – Data that comes from an external server that must be integrated with our UI state.
4. **URL state** – Data that exists on our URLs, including the pathname and query parameters.

Examples of State Management Libraries

- Redux
- MobX
- Vuex
- Zustand
- GraphQL
- Jotai
- Recoil
- Rematch
- RxJS
- Valtio
- React Context
- Flux

What is Redux?

- Redux is a library for managing global application state.
- Redux is typically used with the React-Redux library for integrating Redux and React together.
- Redux Toolkit is the recommended way to write Redux logic.


Redux - A predictable state con
redux.js.org
Redux
Redux
A Predictable State Container for JS Apps
Get Started
Predictable

Redux helps you write applications that **behave consistently**, run in different environments (client, server, and native), and are **easy to test**.

https://redux.js.org/

Redux Libraries and Tools

1. React-Redux
2. Redux Toolkit
3. Redux DevTools Extension

Redux Terms and Concepts

Immutability

- "Mutable" means "changeable". If something is "immutable", it can never be changed.
- In order to update values immutably, your code must make copies of existing objects/arrays, and then modify the copies.

Redux Terminologies

- **Actions** - An action is a plain JavaScript object that has a type field. You can think of an action as an event that describes something that happened in the application.
- **Action Creators** - An action creator is a function that creates and returns an action object.
- **Reducers** - A reducer is a function that receives the current state and an action object, decides how to update the state if necessary, and returns the new state: (state, action) => newState.

Redux Terminologies

- **Store** - The current Redux application state lives in an object called the store.
- **Dispatch** - The Redux store has a method called dispatch. You can think of dispatching actions as "triggering an event" in the application.
- **Selectors** - Selectors are functions that know how to extract specific pieces of information from a store state value.

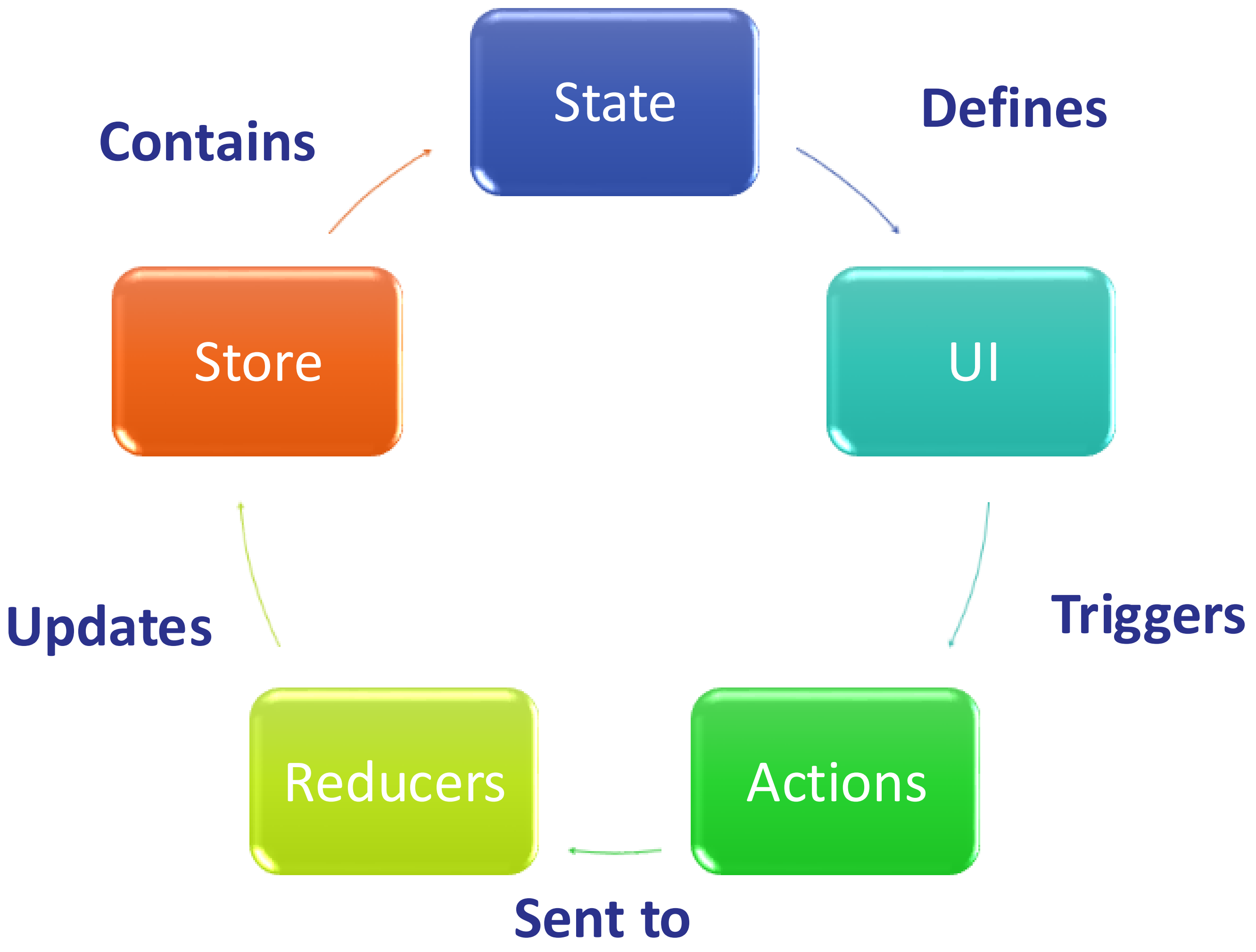
Redux Application Data Flow

The **"one-way data flow",** which describes this sequence of steps to update the app:

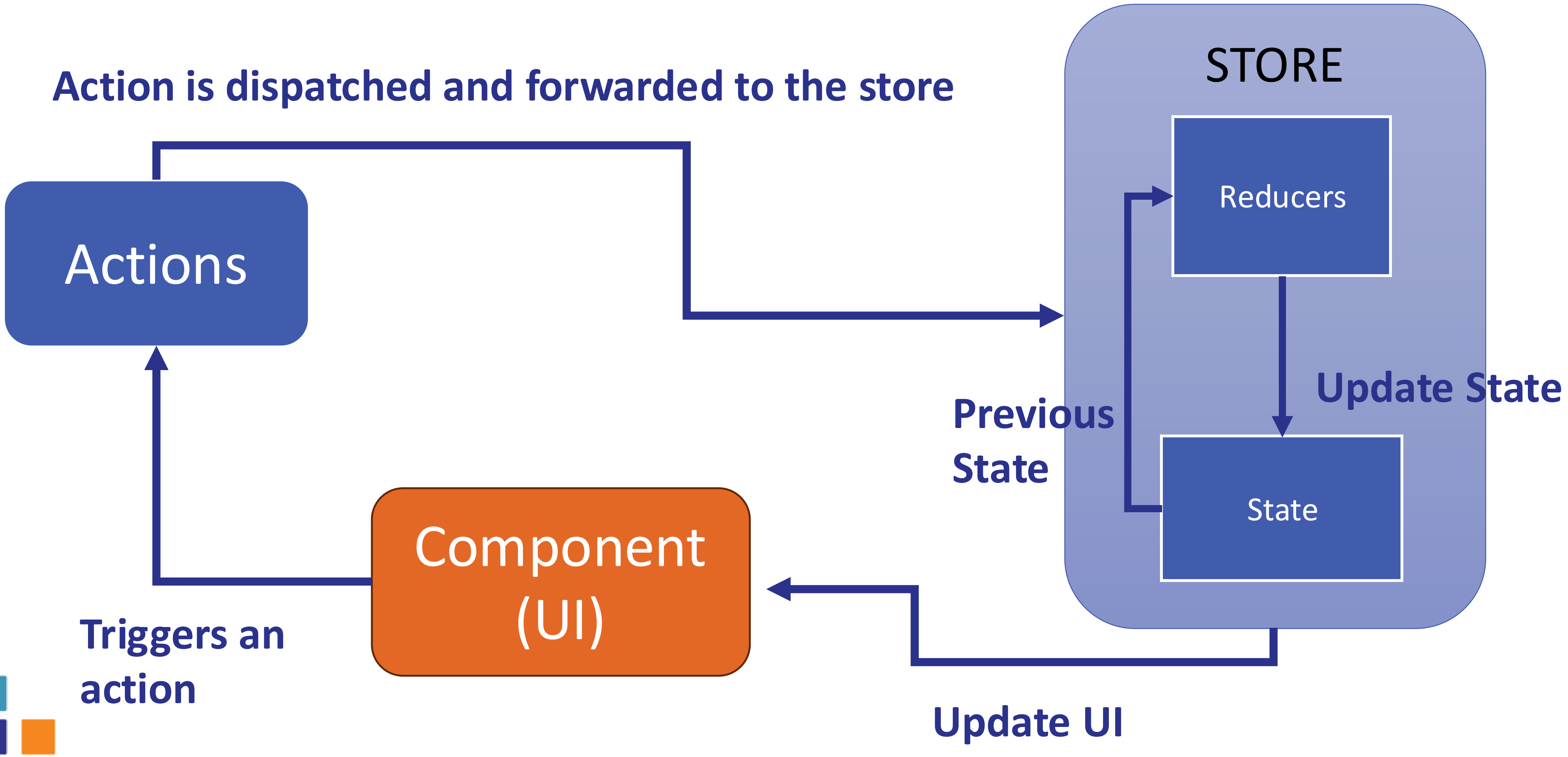
1. State describes the condition of the app at a specific point in time.
2. The UI is rendered based on that state.
3. When something happens (such as a user clicking a button), the state is updated based on what occurred.
4. The UI re-renders based on the new state.

How Redux Works

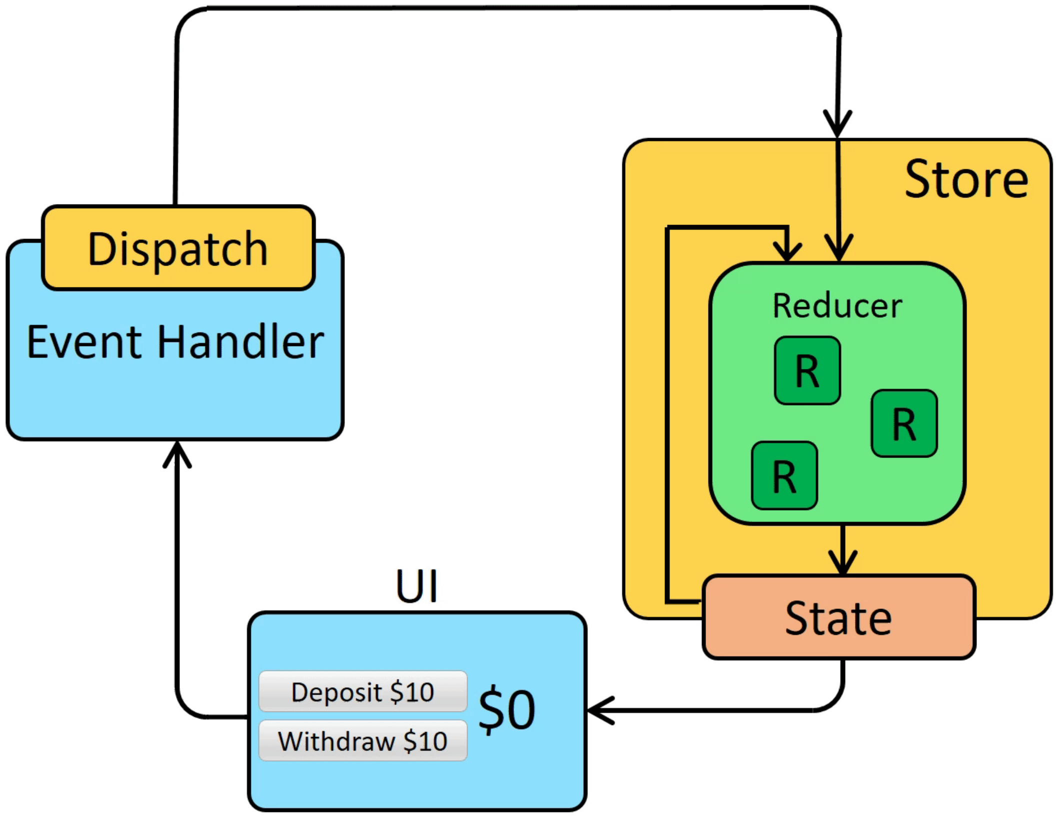
Three building parts: **actions**, **store**, and **reducers**



Redux Application Data Flow



Redux Application Data Flow



Source: https://redux.js.org/tutorials/essentials/part-1-overview-concepts

References

1. **Managing State**
https://react.dev/learn/managing-state
2. **ReactJS State: SetState, Props and State Explained**
https://www.simplilearn.com/tutorials/reactjs-tutorial/reactjs-state
3. **How to Manage State in Your React Apps**
https://www.freecodecamp.org/news/how-to-manage-state-in-your-react-apps/
4. **Redux Essentials, Part 1: Redux Overview and Concepts**
https://redux.js.org/tutorials/essentials/part-1-overview-concepts
5. **React state management: What is it and why to use it?**
https://www.loginradius.com/blog/engineering/react-state-management/