Introduction to Redux

What is Redux ?

- **Redux** is an open-source JavaScript library for managing application state.
- commonly used with libraries such as React or Angular for building user interfaces.
- Similar to (and inspired by) Facebook's Flux architecture
- was created by Dan Abramov and Andrew Clark.

Do I need Redux?

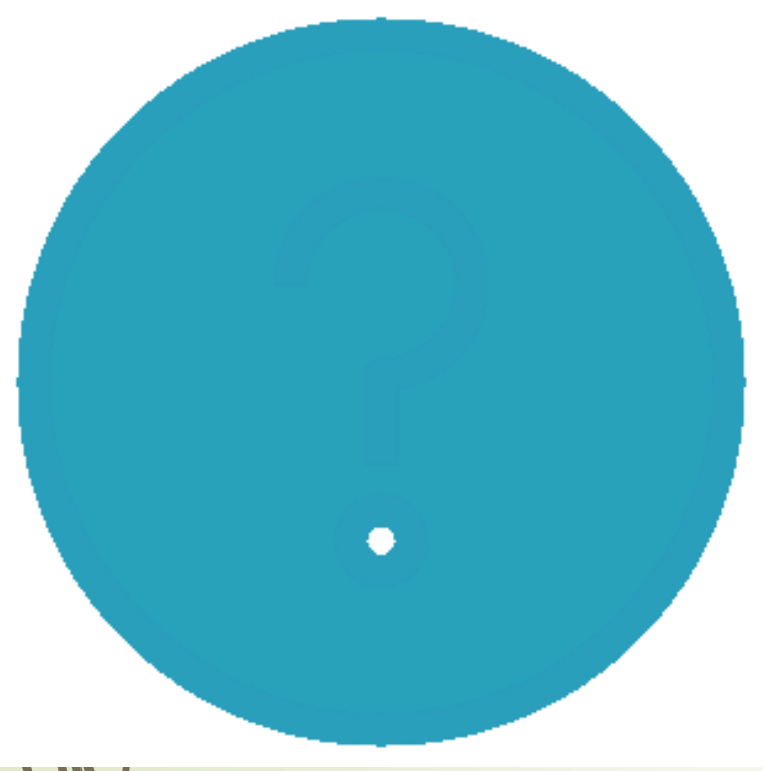
3 Principles

Flux vs Redux

Redux Flow

Do I need Redux?

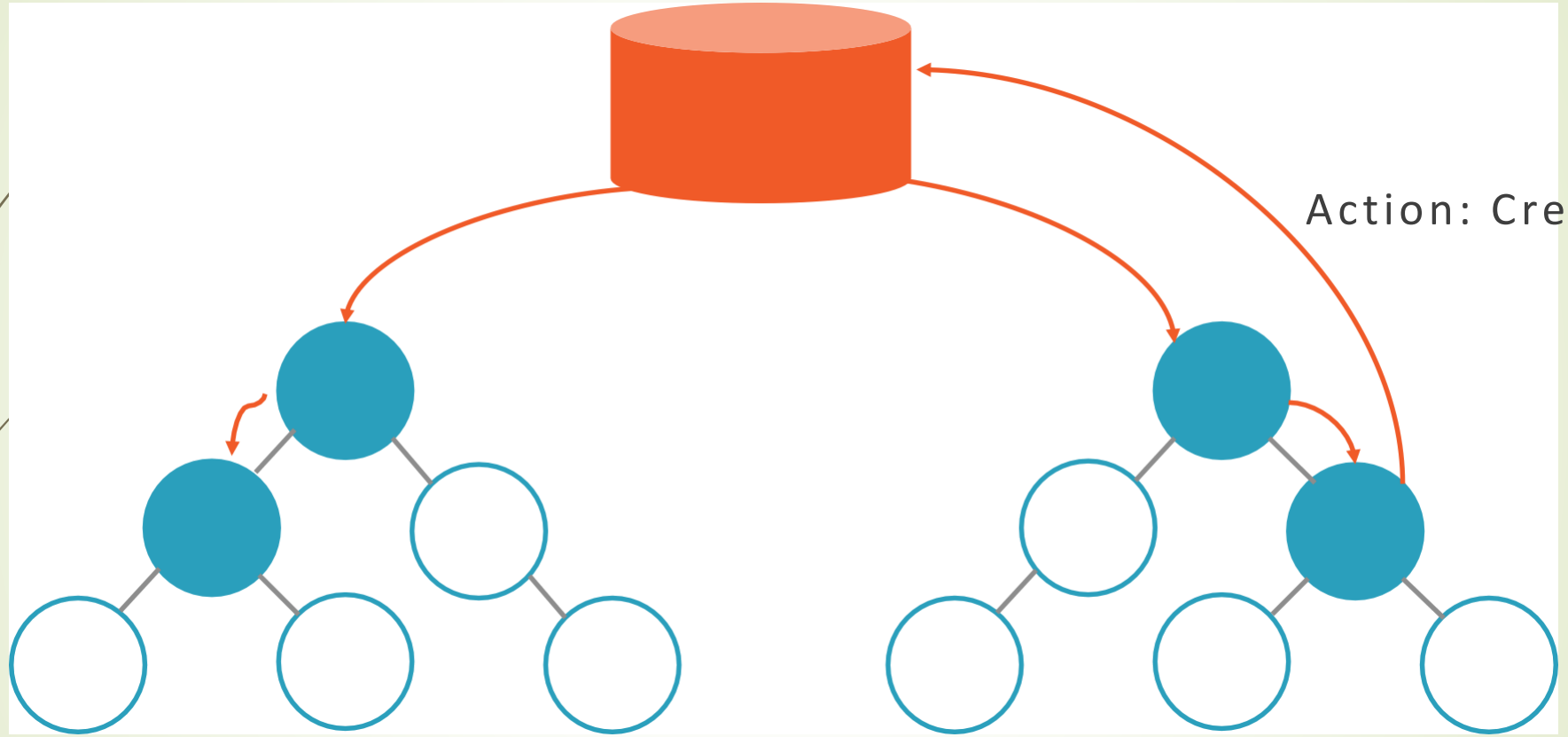
When Do INeed Redux?



- Complex data flows
- Inter-component communication

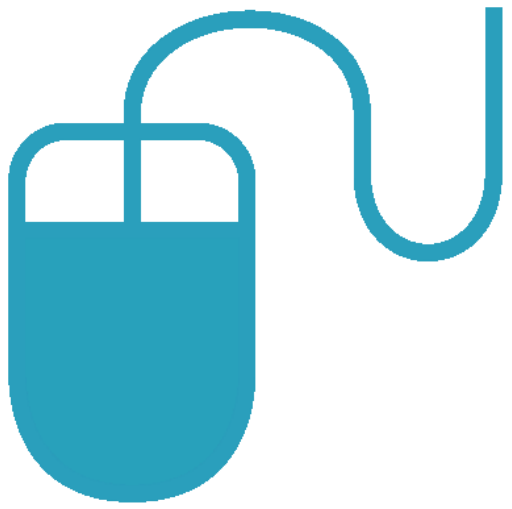
  Non-heirarchical data
- Many actions
- Same data used in multiple places

Store



Redux: 3 Principles

One immutable store



Actions trigger changes

Reducers update state

Flux vs Redux

Data down. Actions up.

Flux and Redux: Similarities

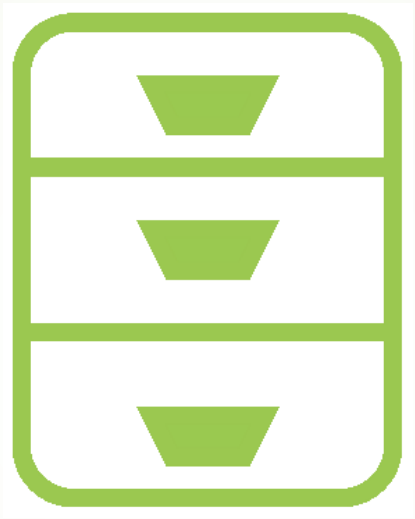
Unidirectional Flow

Actions

Stores

Redux: New Concepts

Reducers

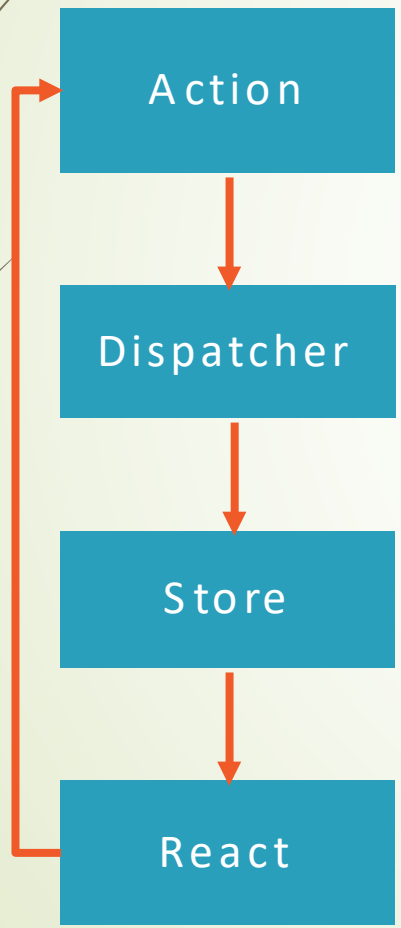


Containers

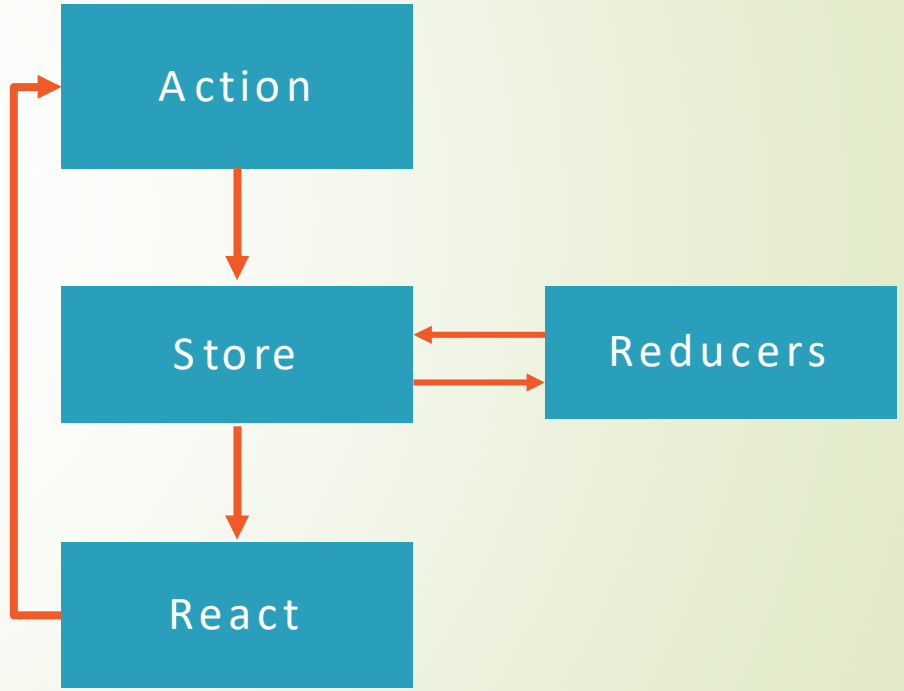
Immutability

Flux vs Redux

Flux



Redux



Flux

Stores contain state and changelogic

Redux

Store and change logicare separate

Flux

Stores contain state and changelogic

Multiple stores

Redux

Store and change logicare separate

One store

Flux

Stores contain state and change logic

Multiple stores

Flat and disconnected stores

Redux

Store and change logic are separate

One store

Single store with hierarchical reducers

Flux

Stores contain state and change logic

Multiple stores

Flat and disconnected stores

Singleton dispatcher

Redux

Store and change logic are separate

One store

Single store with hierarchical reducers

No dispatcher

Flux

Stores contain state and change logic

Multiple stores

Flat and disconnected stores

Singleton dispatcher

React components subscribe to stores

Redux

Store and change logic are separate

One store

Single store with hierarchical reducers

No dispatcher

Container components utilize connect

Flux

Stores contain state and change logic

Multiple stores

Flat and disconnected stores

Singleton dispatcher

React components subscribe to stores

State is mutated

Redux

Store and change logic are separate

One store

Single store with hierarchical reducers

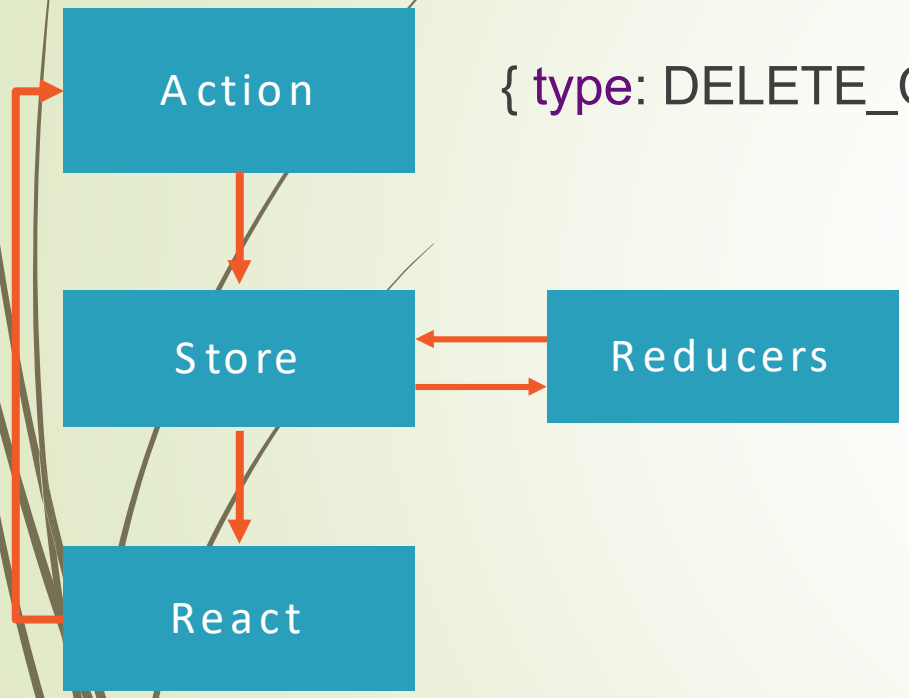
No dispatcher

Container components utilize connect

State is immutable

Redux Flow

Redux Flow



Notified via React-Redux

{ type: DELETE_COURSE, id: 5 }

```
function appReducer(state = defaultState, action) {
  switch(action.type) {
    case DELETE_COURSE:
    //return new state
  }
}
```