

Here's The Plan



Do I need Redux?

3 Principles

Flux vs Redux

Redux Flow



Do I need Redux?



Vanilla JS

jQuery

React

React + Redux

Simple

No setup

Complex

Significant setup



When Do I Need Redux?



Complex data flows

Inter-component communication

Non-heirarchical data

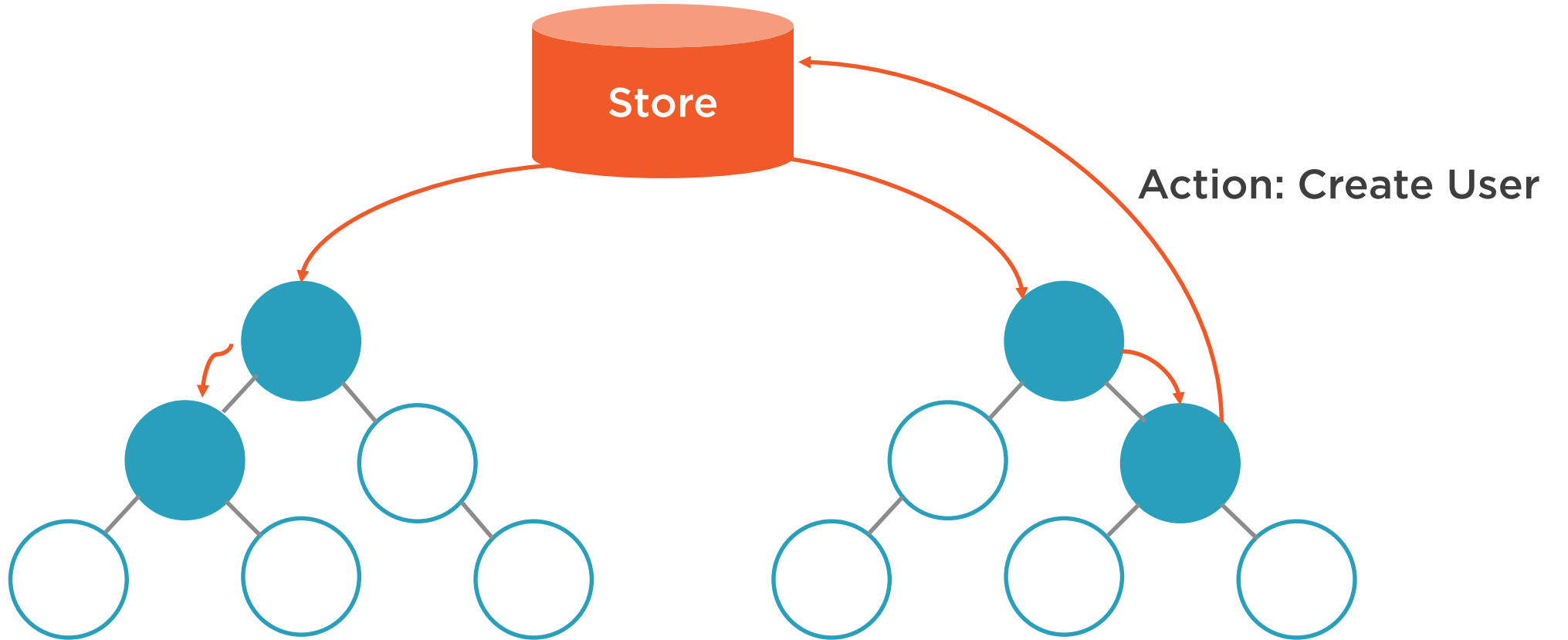
Many actions

Same data used in multiple places

“...If you aren't sure if you need it,
you don't need it.”

Pete Hunt on Flux and Redux

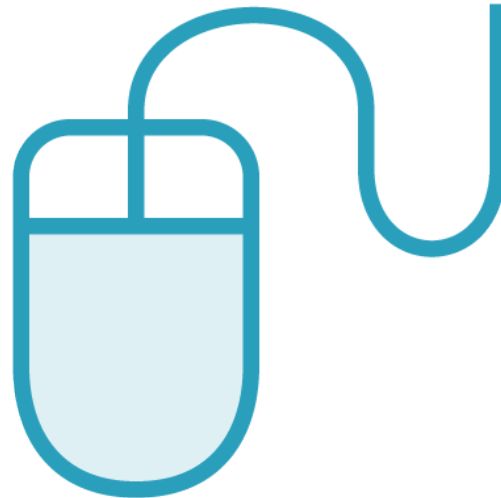




Redux: 3 Principles



One immutable store



Actions trigger changes



Reducers update state

Flux vs Redux



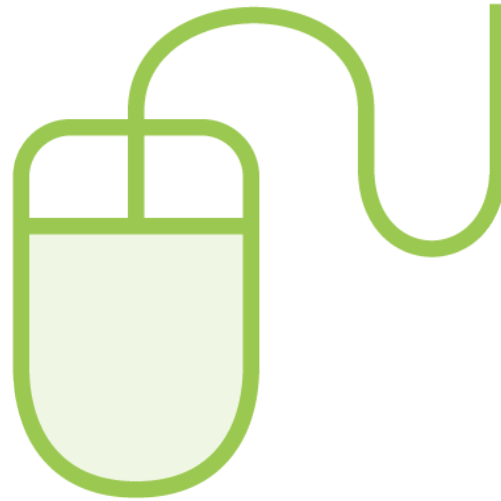
Data down.
Actions up.



Flux and Redux: Similarities



Unidirectional Flow



Actions



Stores

Redux: New Concepts



Reducers



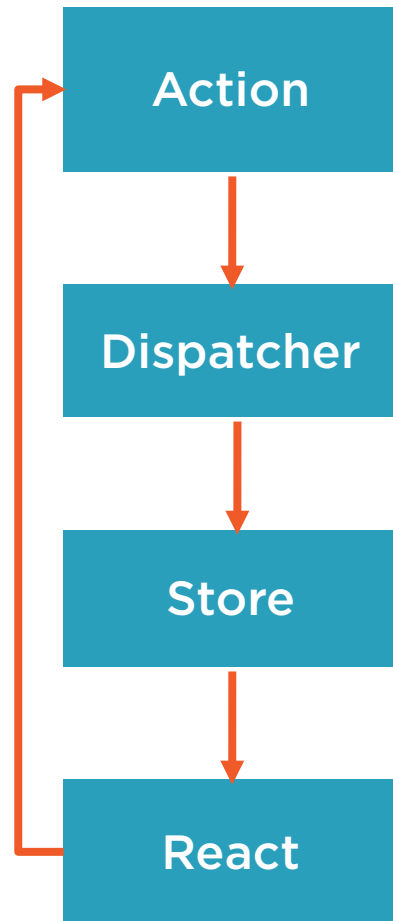
Containers



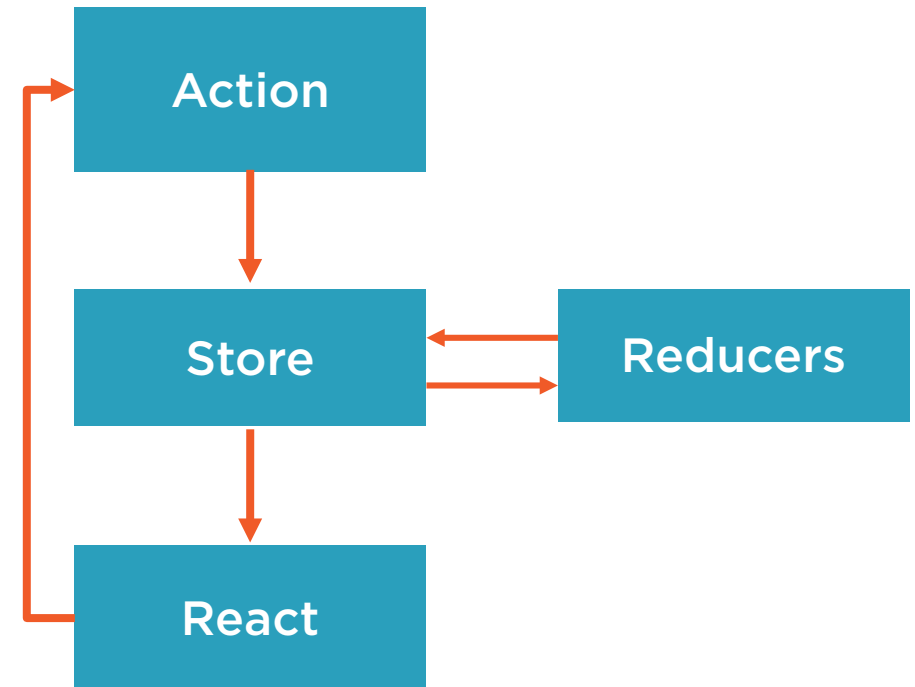
Immutability

Flux vs Redux

Flux



Redux



Flux

Stores contain state and change logic

Redux

Store and change logic are separate



Flux

Stores contain state and change logic

Multiple stores

Redux

Store and change logic are 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

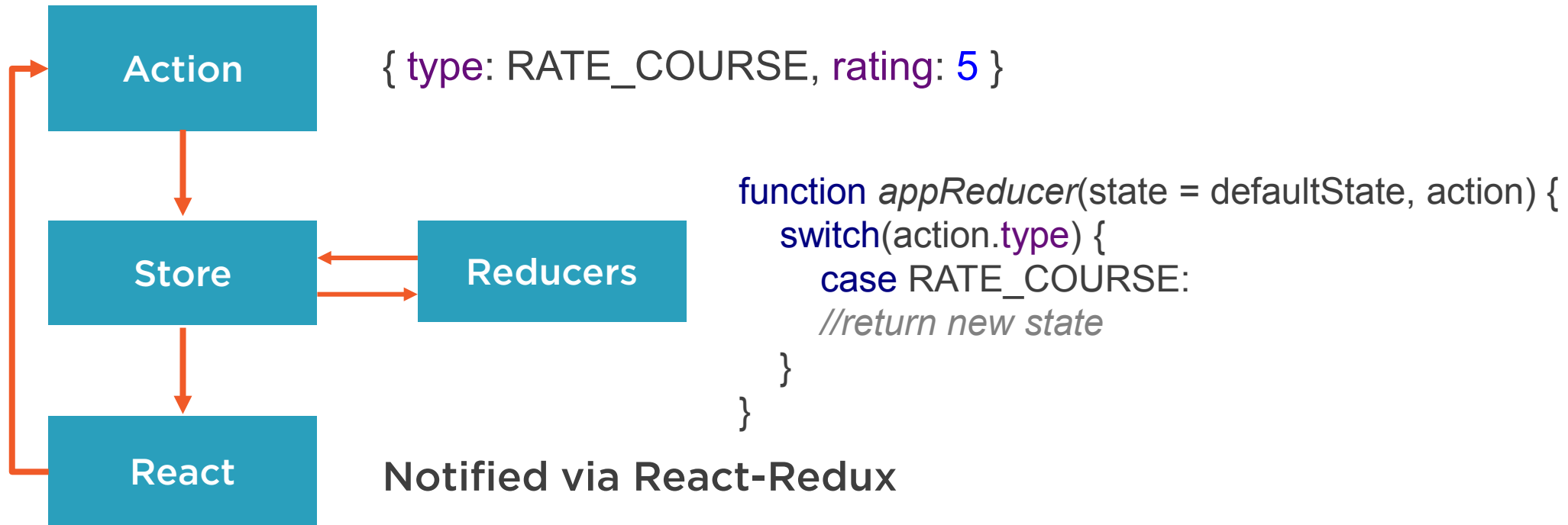
State is immutable



Redux Flow



Redux Flow



Summary



If you need Redux, you'll know it.

3 Principles

- Immutable Store
- Actions trigger changes
- Reducers return updated state

Flux vs Redux

Unidirectional Redux Flow

Next up: Actions, stores, and reducers

