

What is “Redux”?

A state management system for
cross-component or app-wide
state

What Is Cross-Component / App-Wide State?

Local State

State that belongs to a single component

E.g. listening to user input in a input field; toggling a "show more" details field

Should be managed component-internal with `useState()` / `useReducer()`

Cross-Component State

State that affects multiple components

E.g. open/ closed state of a modal overlay

Requires "prop chains" / "prop drilling"

App-Wide State

State that affects the entire app (most/ all components)

E.g. user authentication status

Requires "prop chains" / "prop drilling"

OR: React Context or Redux

What is “Redux”?

A state management system for cross-component or app-wide state



Don't we have “React Context” already?

React Context – Potential Disadvantages

Complex Setup / Management

In more complex apps, managing React Context can lead to deeply nested JSX code and / or huge “Context Provider” components

Performance

React Context is not optimized for high-frequency state changes

React Context – Complex Setup

```
return (  
  <AuthProvider>  
    <ThemeProvider>  
      <UIInteractionContextProvider>  
        <MultiStepFormContextProvider>  
          <UserRegistration />  
        </MultiStepFormContextProvider>  
      </UIInteractionContextProvider>  
    </ThemeProvider>  
  </AuthProvider>  
);
```

React Context – Complex Setup

```
function AllContextProvider() {  
  const [isAuth, setIsAuth] = useState(false);  
  const [isEvaluatingAuth, setIsEvaluatingAuth] = useState(false);  
  const [activeTheme, setActiveTheme] = useState('default');  
  const [ ... ] = useState(...);  
  
  function loginHandler(email, password) { ... };  
  
  function signupHandler(email, password) { ... };  
  
  function changeThemeHandler(newTheme) { ... };  
  
  ...  
  
  return (  
    <AllContext.Provider>  
  
    </AllContext.Provider>  
  )  
}
```

React Context – Performance



sebookmarkage commented on 18 Dec 2018

Member



My personal summary is that new context is ready to be used for low frequency unlikely updates (like locale/theme). It's also good to use it in the same way as old context was used. I.e. for static values and then propagate updates through subscriptions. It's not ready to be used as a replacement for all Flux-like state propagation.

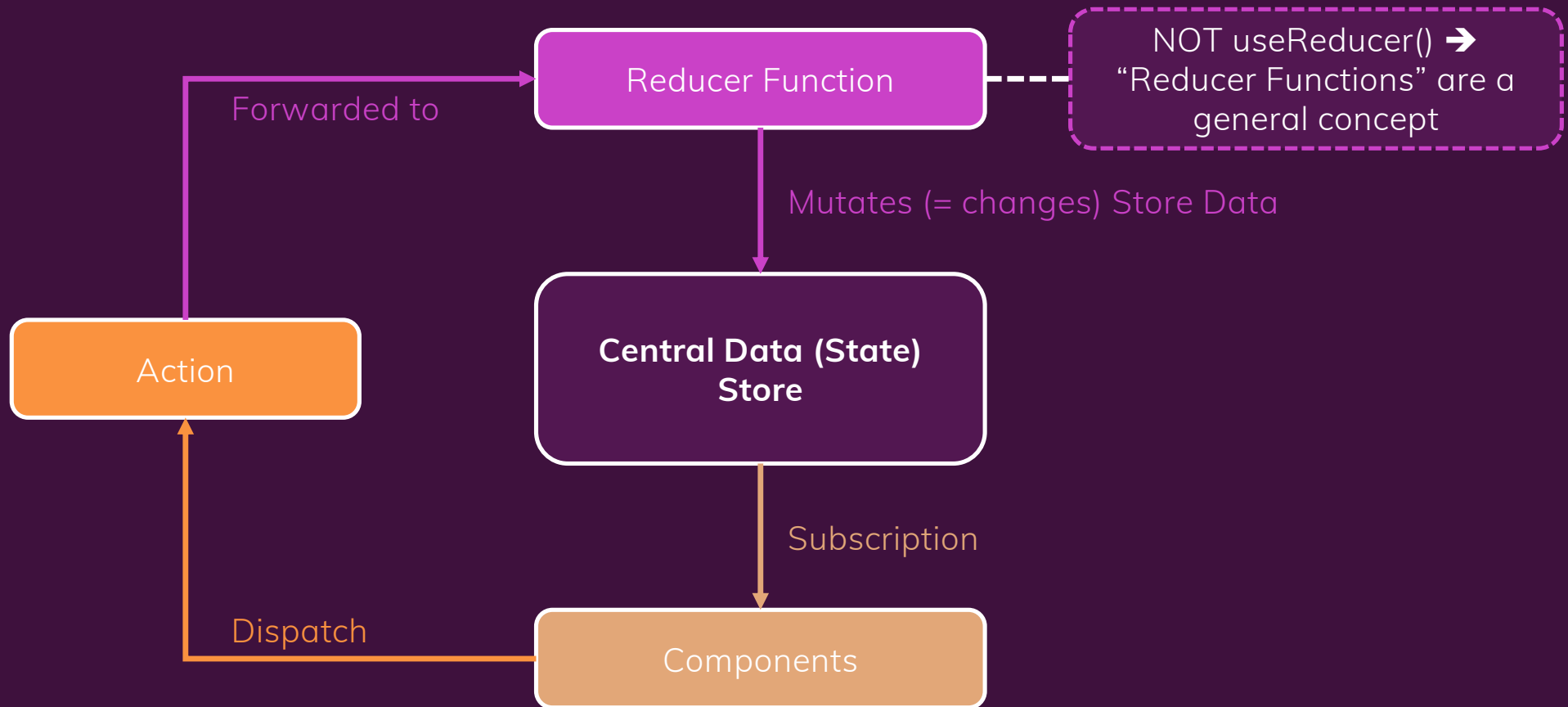


55

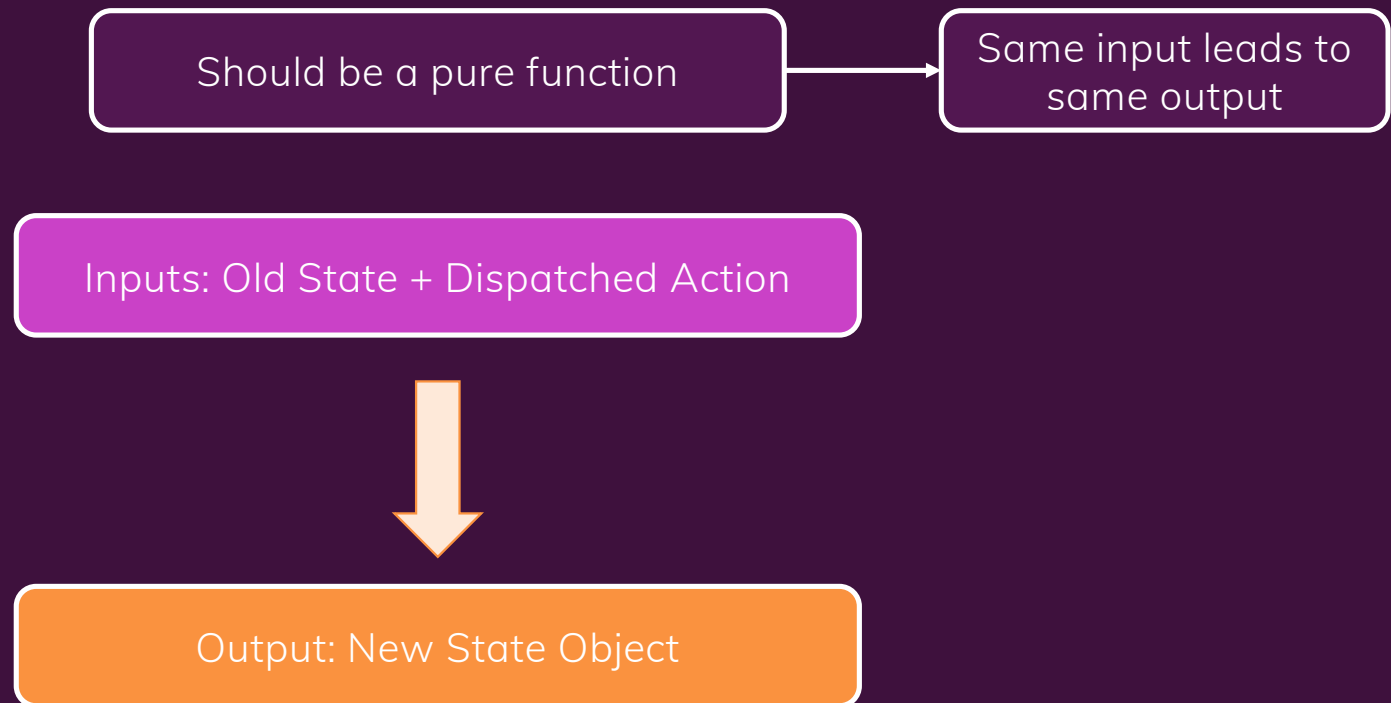


4

Core Redux Concepts



The Reducer Function



The Role Of Immutability

State updates must be done in an immutable way!

Objects and arrays are reference values in JavaScript



Changes made to an object property affect ALL places where the object gets used

New object / array copies (also of nested objects / arrays) must be created when producing a new state