

# 프로젝트로 배우는 React 웹 애플리케이션 개발

Alex Kwon

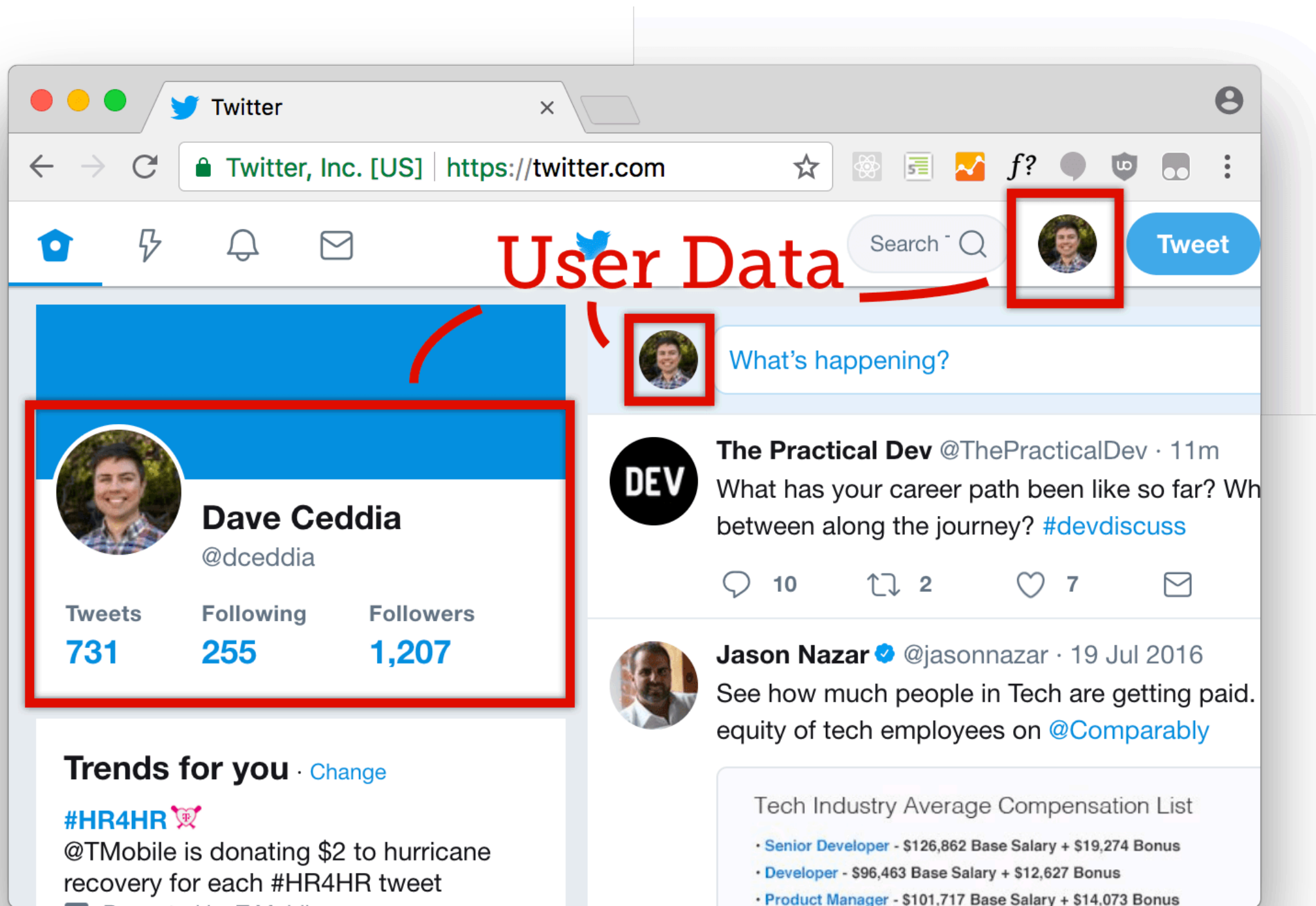
리덕스 활용하기 12/15



- Redux의 사용 목적
- Flux 아키텍처
- 실습: 냥포럼 redux로 리팩토링하기

- 애플리케이션의 상태(state) 관리 라이브러리
- 컴포넌트가 데이터에 직접 접근하기 위해 사용

# Redux의 사용 목적 - 복잡한 UI 구조



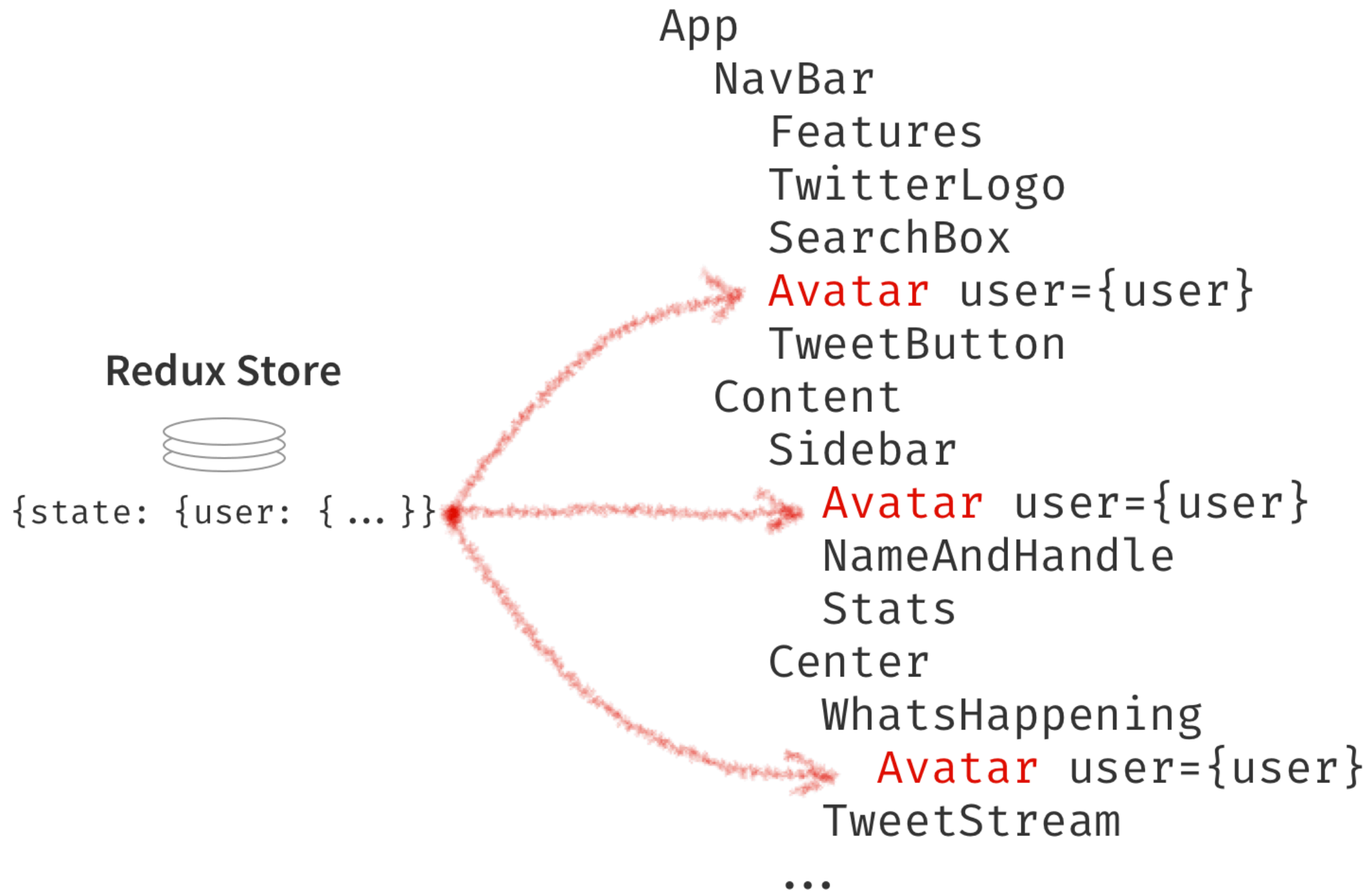
# Redux의 사용 목적 - 불필요한 Props의 전달



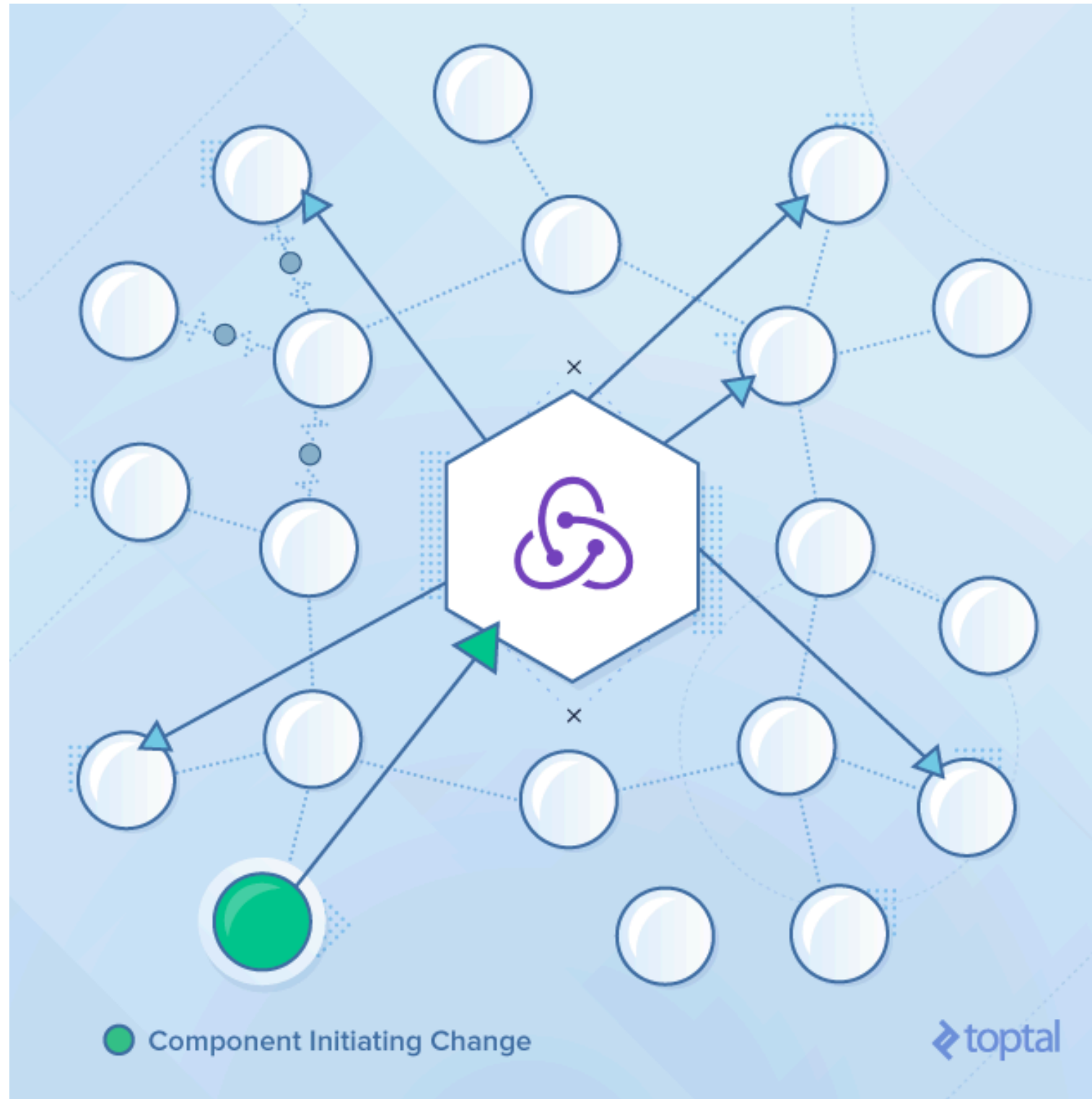
learnit

```
App state={user: { ... }}
  NavBar user={user}
  Features
  TwitterLogo
  SearchBox
  Avatar user={user}
  TweetButton
  Content user={user}
  Sidebar user={user}
    Avatar user={user}
    NameAndHandle user={user}
    Stats user={user}
  Center user={user}
    WhatsHappening user={user}
    Avatar user={user}
  TweetStream
  ...
```

# Redux의 사용 목적 - 데이터에 직접 접근



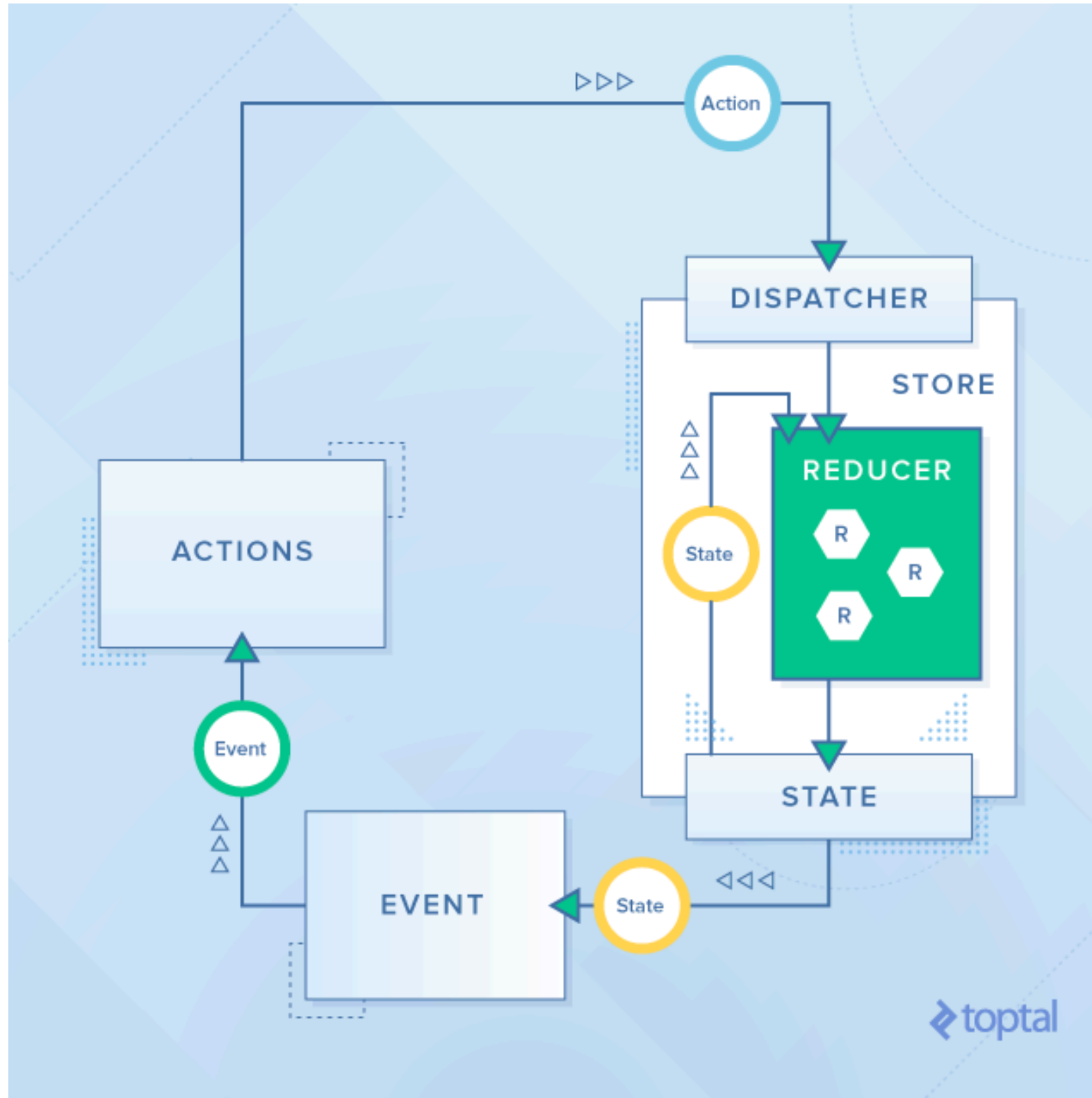
# Redux의 사용 목적 - 데이터에 직접 접근



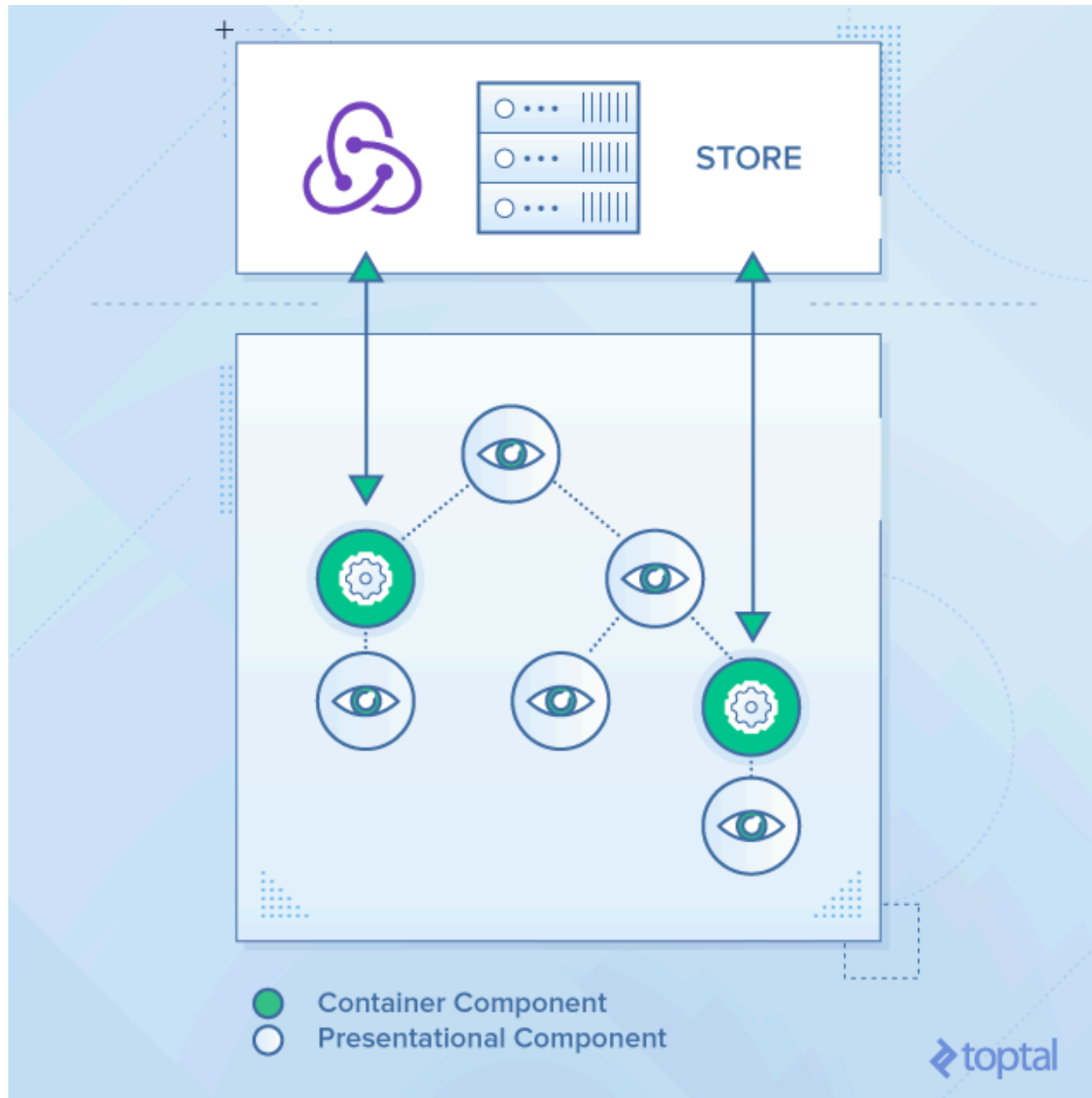
- 단일 state
- action으로만 state 데이터 변경 가능
- 각 action 별 state 데이터 처리 로직을 reducer에 작성
- action을 dispatch하여 reducer 호출
- \* 불변성 및 시간 흐름에 따른 제어에 용이



# Flux 아키텍처 - Redux 구조



# Flux 아키텍처 - 리액트에서 사용하기



1. Chrome => Redux DevTools 설치하기
2. `npm install redux react-redux redux-thunk`
3. store 구성하기: `createStore`, `combineReducer`, `applyMiddleware`
4. `<Provider/>`로 감싸기

1. 컴포넌트 connect & mapStateToProps 작성
2. actionTypes 작성
3. initialState & reducer 작성
4. action creator 작성
5. 컴포넌트에서 dispatch
6. reducer 로직 작성

- Flux 아키텍처 & Redux
- redux-thunk로 비동기 통신

## 복습

- “냥포럼” Redux로 리팩토링하기

## 과제

- Redux 공식문서 튜토리얼 따라하기 - <https://redux.js.org/basics> 에서 Basics 가이드를 따라하며 간단한 Todo 앱 만들기



다음에  
또!  
같이!  
만나요!