sourcemind

STATE MANAGEMENT

REDUX

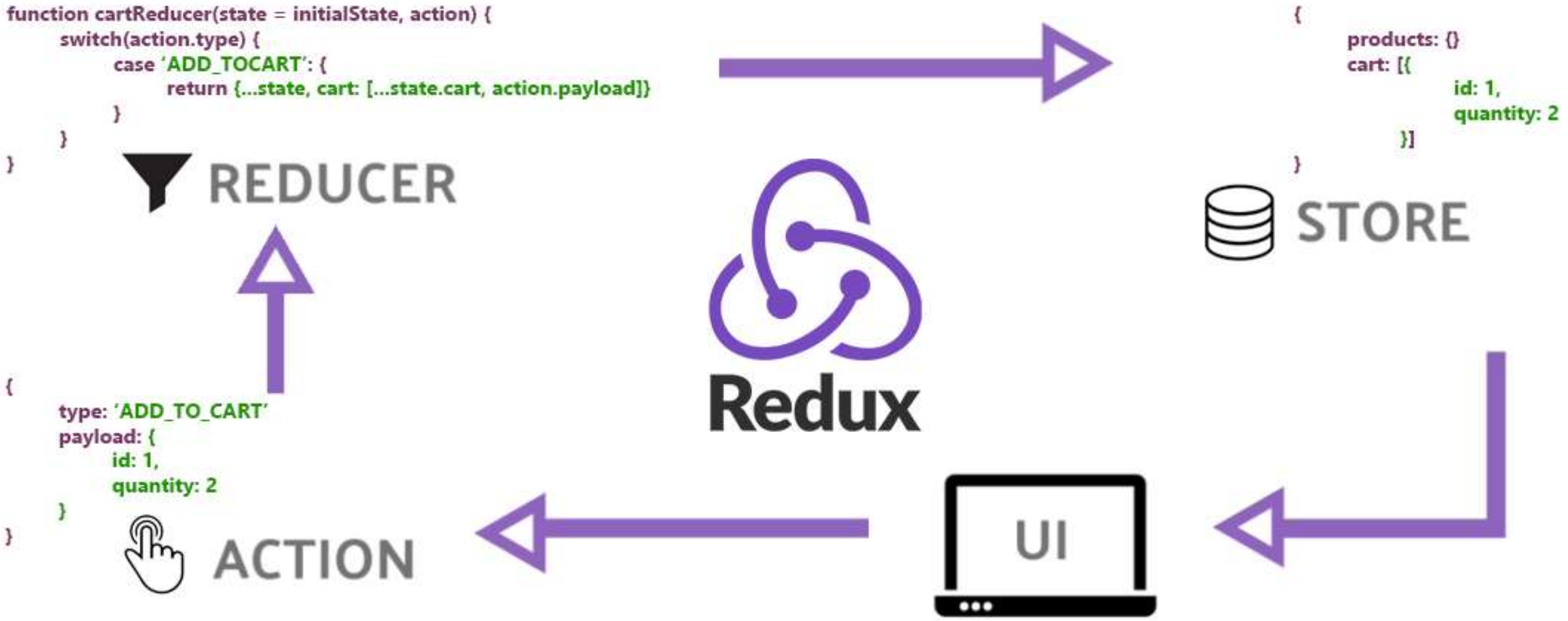
STATE MANAGEMENT

React state management is **a process for managing the data that React components need in order to render themselves**.
This data is typically stored in the component's state object.

1. Recoil
2. Jotai
3. Redux
4. Rematch
5. Mobx
6. Hookstate

A Predictable State Container for JS Apps

sourcemind


function cartReducer(state = initialState, action) {
switch(action.type) {
case 'ADD_TOCART': {
return {...state, cart: [...state.cart, action.payload]}
}
}
}
REDUCER
{
products: {}
cart: [{
id: 1,
quantity: 2
}]
}
STORE
Redux
{
type: 'ADD_TO_CART'
payload: {
id: 1,
quantity: 2
}
}
ACTION
UI

sourcemind

Creating Store

```
import { createStore } from 'redux'

function todos(state = [], action) {
  switch (action.type) {
    case 'ADD_TODO':
      return state.concat([action.text])
    default:
      return state
  }
}

const store = createStore(todos, ['Use Redux'])

store.dispatch({
  type: 'ADD_TODO',
  text: 'Read the docs'
})

console.log(store.getState())
// [ 'Use Redux', 'Read the docs' ]
```

Store Methods

- `getState()`
- `dispatch(action)`
- `subscribe(listener)`
- `replaceReducer(nextReducer)`

Provider

```
import React from 'react'
import ReactDOM from 'react-dom/client'

import { Provider } from 'react-redux'
import store from './store'

import App from './App'

// As of React 18
const root = ReactDOM.createRoot(document.getElementById('root'))
root.render(
  <Provider store={store}>
    <App />
  </Provider>
)
```

Action

```
import React from 'react'
import { useDispatch } from 'react-redux'

export const CounterComponent = ({ value }) => {
  const dispatch = useDispatch()

  return (
    <div>
      <span>{value}</span>
      <button onClick={() => dispatch({ type: 'increment-counter' })}>
        Increment counter
      </button>
    </div>
  )
}
```

useSelector

```
import React from 'react'
import { useSelector } from 'react-redux'

export const CounterComponent = () => {
  const counter = useSelector((state) => state.counter)
  return <div>{counter}</div>
}
```