39. Redux & Context api in React

- Fixing problem rewriting Code logic
- Two places, same logic

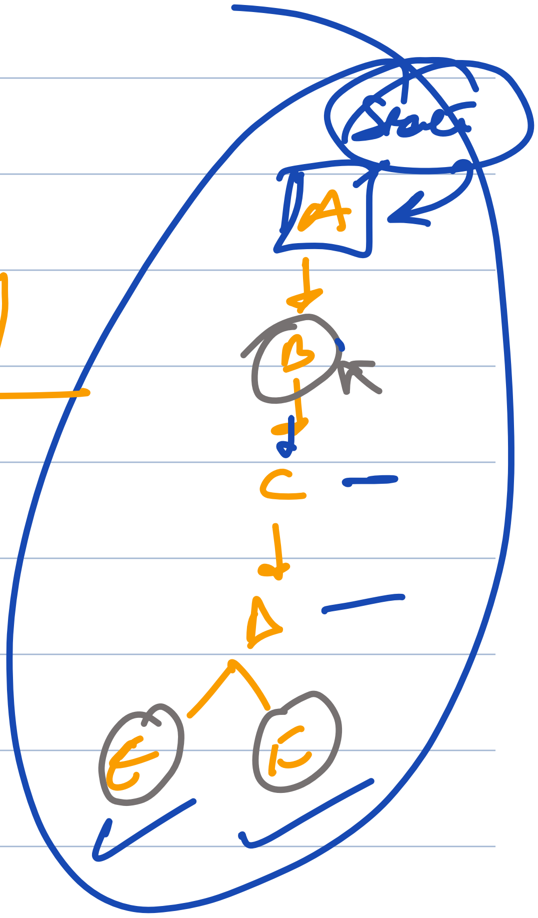
Solution 1 :

① Lifting the State up

+

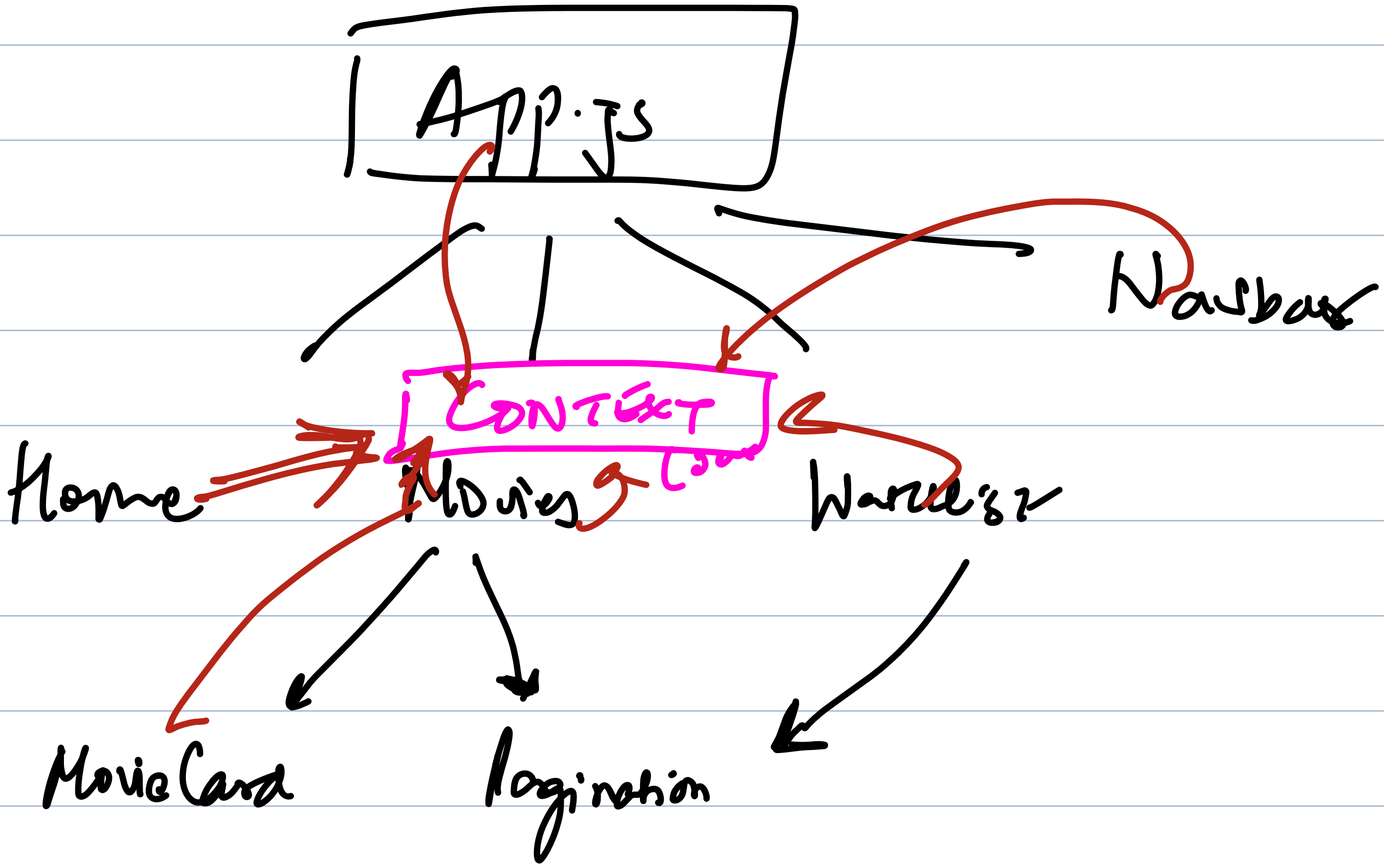
Problem

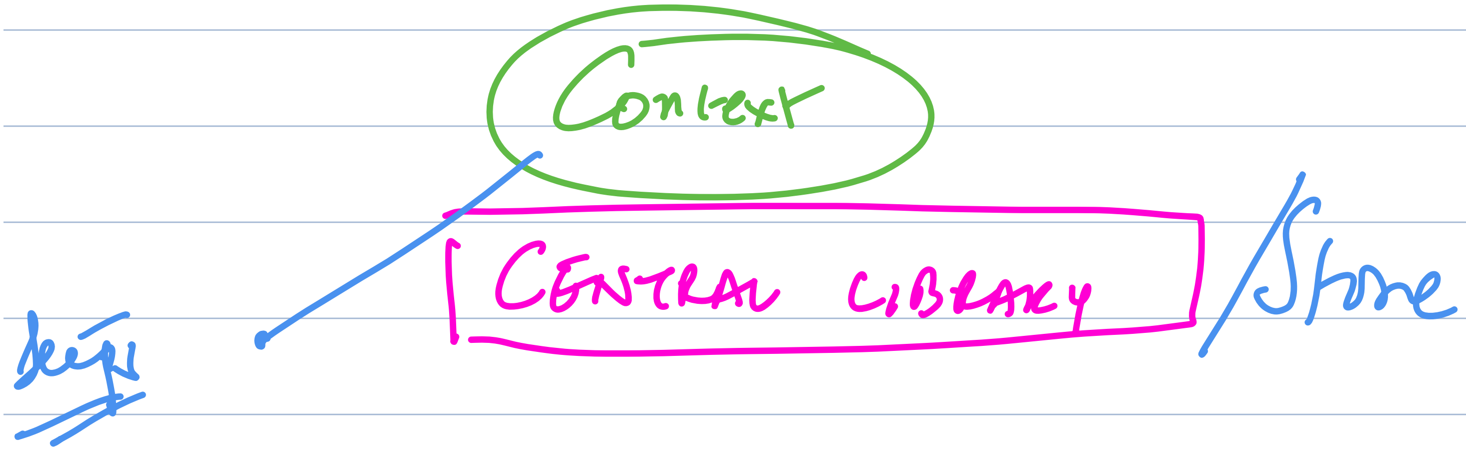
When descendants need a prop, and height of Component tree is huge, we have to pass props in whole chain.



② Context API .

↳ Built in with React





1) Create a Context (React.createContext())

2) Provide the Context

3) Consume the Context

Children → props inbuilt to react that refer to any Component that we will wrap

```
<Home>
    <About />
</Home>
```

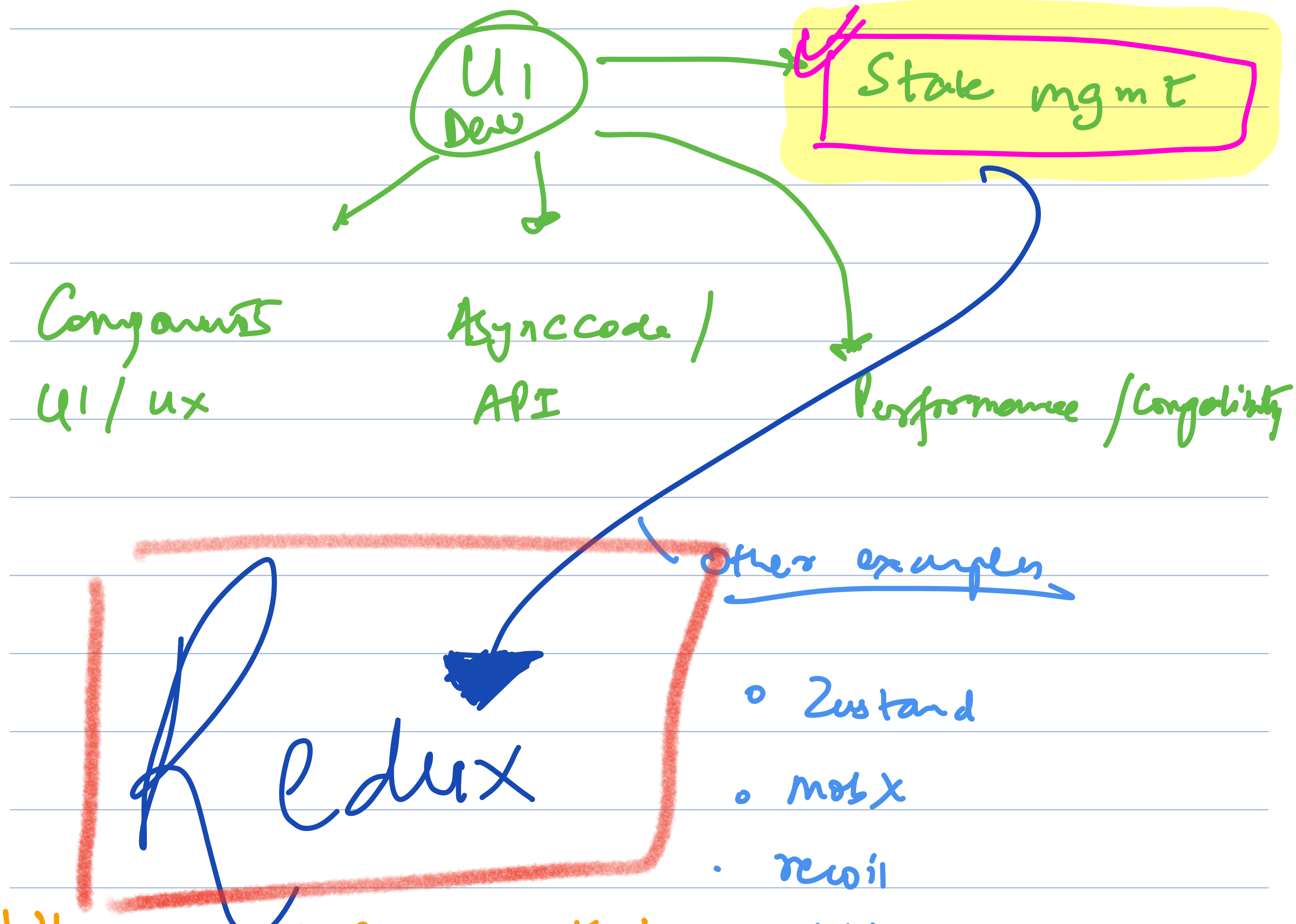
```
function Home (props)        {
    const {Children} = props.

        <props.children />
}
```

</About>

Agenda. (second half)

- Redux
- Why redux, features
- redux-toolkit, reduxjs
- Integrate redux with react (Sample)



Other examples

- Zustand
- mobx
- recoil
- ...
- ⋮

Why we need State mgmt library

↳ Global state mgmt

↳ Prop drilling (hand to read)

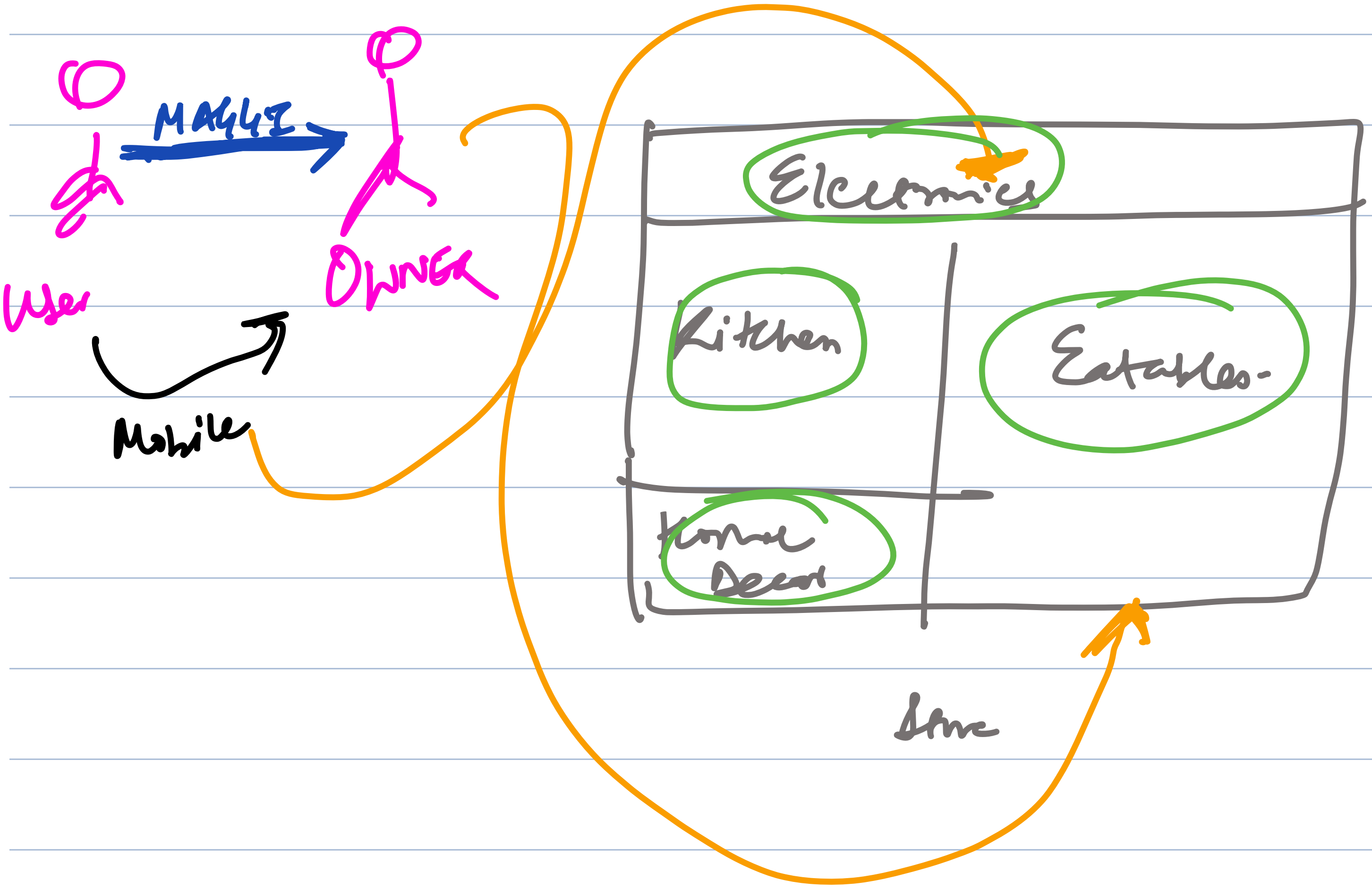
↳ Performance Issues

↳ When using Context API,
any change in Context value
causes re-render in all the
components that consume that Context.
these, lead to perf. issues in large
application.

Redux

- ○ 3rd party library used for state mgmt
- ○ DAN ABRAMOV
  - L react/FB team
- ○ Can be used with other UI frameworks.
- ○ Central state mgmt feature.
- ○ Predictability
- ○ Single source of truth.

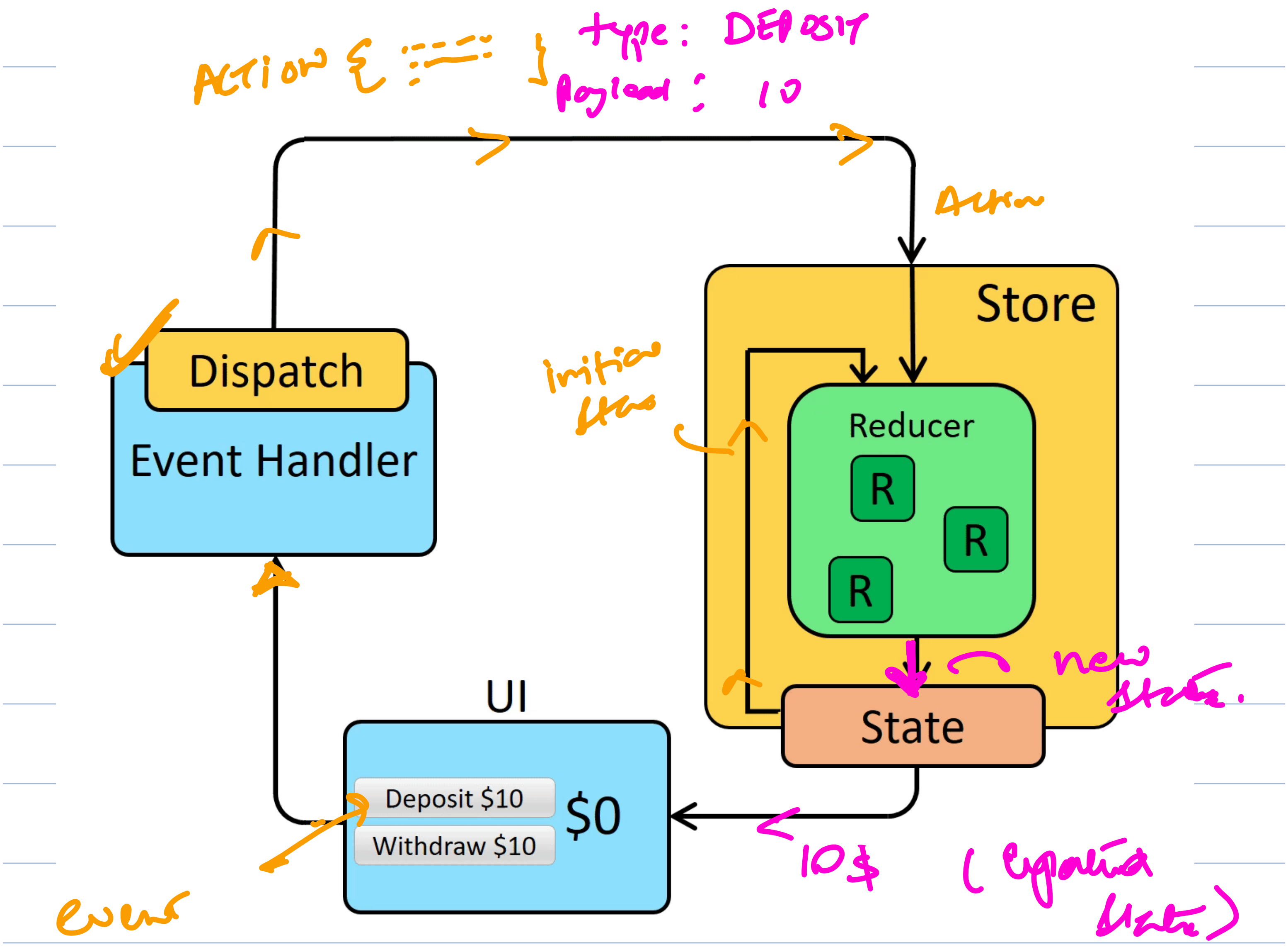
Redux ⟺ Super Market



Sections ⟹ Slices

(in Supermarket) (Stores State of a particular feature)

- All Slices Combine together to make up a Redux Store.
- User will not directly get item, it has to go via Owner.
  Owner lays the ground rules~



Dispatch → method used to send Actions to the store

Actions → plain JS Objects that describe what exactly happened

- Only source of information for the store.

Store → · State of your application lives

· Holds the state & allows access to it.

Reducer → · Reducers specify how the UI state change will affect. How to respond to that action

· Pure functions.
&

I/P → Previous state, action

O/P → Return new state

Redux → Redux toolkit

(initial)
Not
maintained

MAIN PARTS OF REDUX TOOLKIT

1) SLICE

- Part of application state
- Contain
  - └ State
  - └ actions
  - └ dispatchers

For specific part of the application.

2) ACTIONS

JS object.

Describe What happend

Eg: Click { type: 'counter/incr' }

Tell redux what to do

3) REDUCERS

- Pure functions. (initial state, Action) ⇒ new State
- Know how to do work based on whatever Action tell.

4) STORE

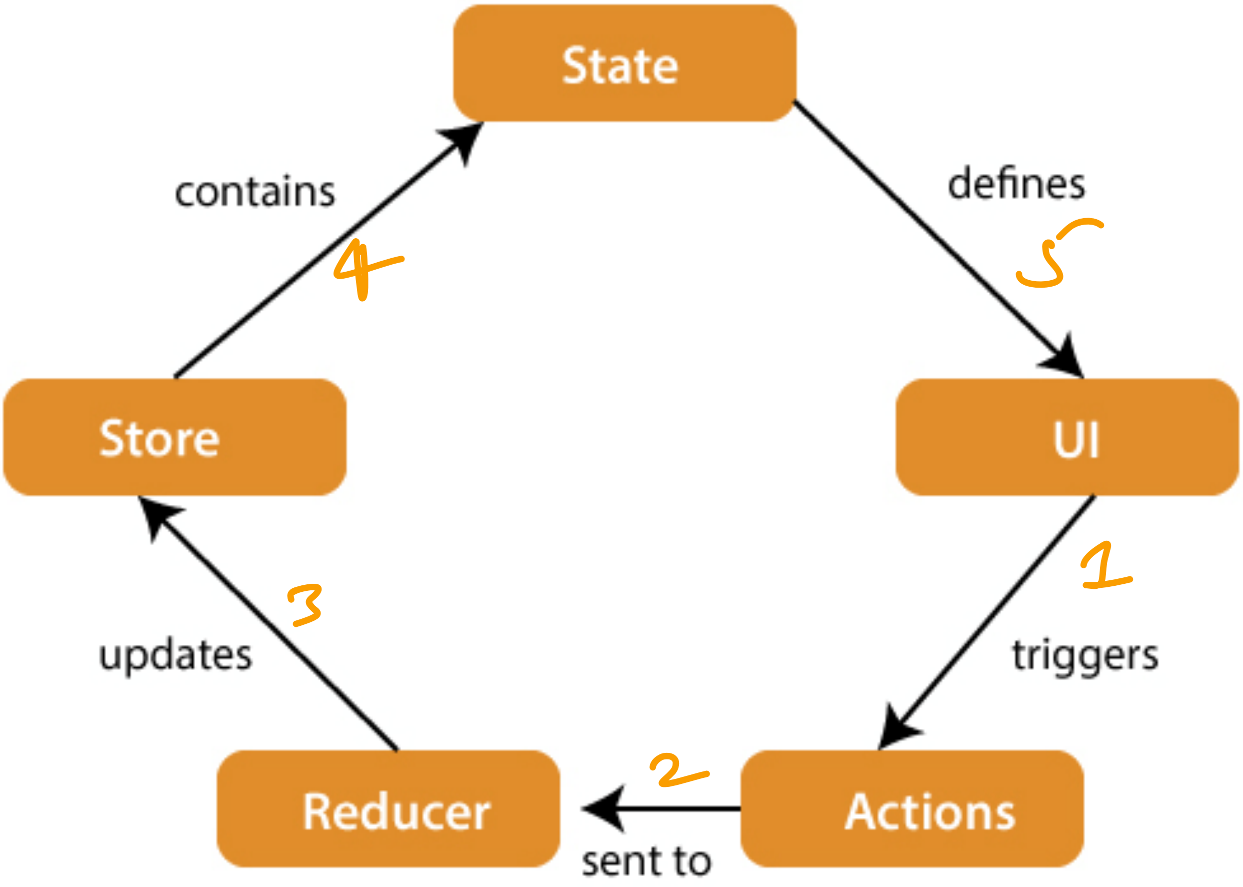
: App State lives.

- Action, Reducer Come together to manage state

5) DISPATCH

(methods used to send Actions to store

When you dispatch an action, you are telling Redux to update the state



HW → Todolist with redux toolkit