

Components --- need to **communicate** with each other

Passing props -----

```
C1
|
| <C2 num=3 />
C2 ( does not want it )
|
| <C3 num=props.num />
C3
|
|
| <C4 num=props.num />
|
|
| <C5 num =props.num />
C5 is using the num
```

This COMPLEXITY between Component communication = **PROPS DRILLING** ----- PROBLEM

Instead PROPS DRILLING ----- Another Solution to **SHARE DATA** BETWEEN COMPONENTS ----**REDUX STORE**

Redux = independent node library
Which will be used with React !!!!

We are planning to integrate React + Redux

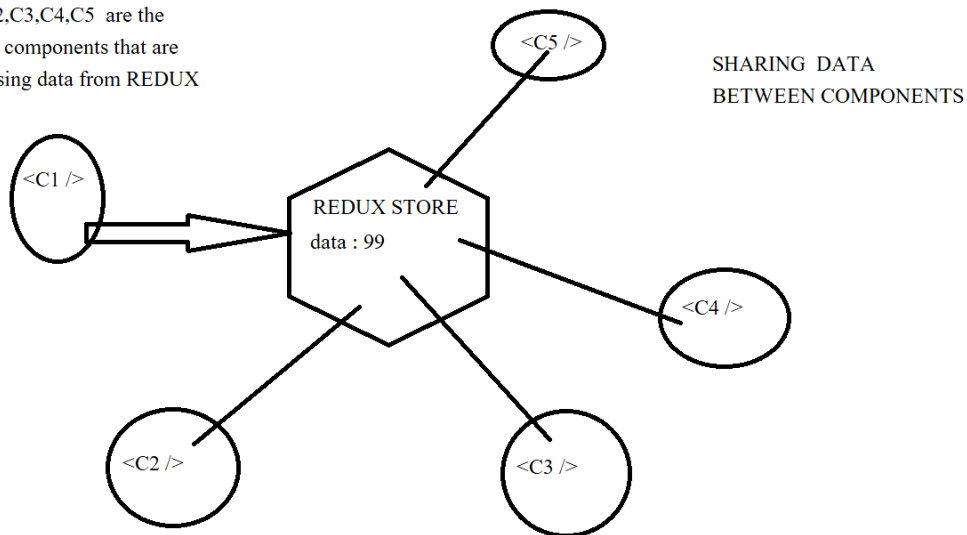
Why to use redux ?

REDUX is a library that can create a STORE / STORAGE

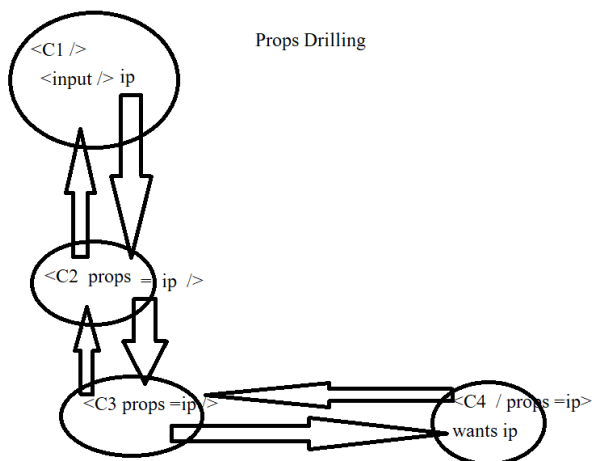
STORE of what ? DATA ---ANY type of data

We will keep this store on the client side (BROWSER)

C1,C2,C3,C4,C5 are the React components that are accessing data from REDUX store



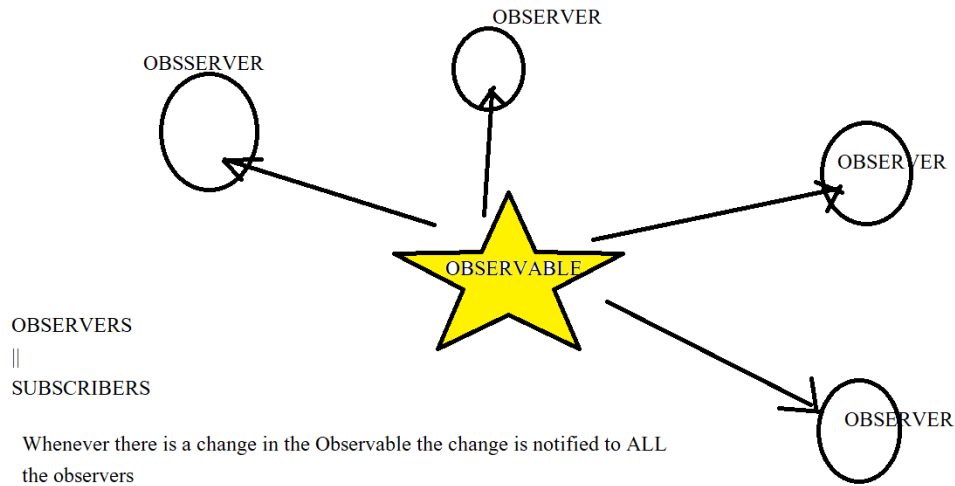
This is solving the problem of PROPS drilling



Cd to my-app (your project)
 npm install react-redux --save
 npm install redux --save

[Connect | React Redux \(react-redux.js.org\)](https://react-redux.js.org)

Redux --- it is kind of Observable design pattern



-
1. We may want to **see** the CURRENT STATE of the shared DATA in REDUX STORE
 2. We may want to **change** the CURRENT STATE of the shared DATA in REDUX STORE

The REDUX Observable is defined as a REDUCER function !!

REDUCER function tells ?

What state is stored ? **State parameter**

What operations can be done on the state by the Observers ?

Action parameter

Reducer function returns the CURRENT STATE (after applying action , if any)

3. Once the reducer is written , we can pass it to the createStore(reducer) that will actually create a storage as per the reducer !!
 4. The job of **dispatch** is to change the state !! What do we want to send ? WE want to send the action on the reducer .
-

Web Servers ----- AWS , Tomcat , React Dev Server

Express Web Server = Node Web Server that uses a node library called as "express"

To download the node library ----- `npm install (name of the library)`
Node libraries are downloaded in `node_modules` --- Where we installed node , in the
react template project(my-app)

If I want to create my own node project

1. Create a folder of your choice
2. Cd to that folder
3. `npm init`
4. It creates a `package.json` file that is a DESCRIPTOR of my project
5. Install all the node libraries that are relevant to this project
6. For example I install express node library
 - i. `npm install express --save`

MERN stack