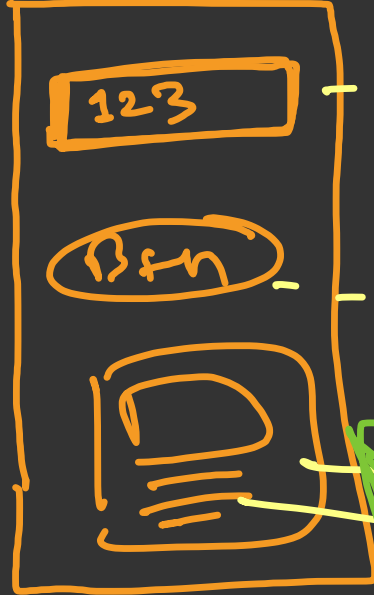


State Management

UI layer



Data



onClick
Btn

~~input~~
~~info~~

Dynamic
Data

Presenter state
visible

Why do you need state??

↳ Dynamic Behaviour

Name

Aksh

value \Rightarrow Aksh

Interactivity

let name \leftarrow "Aksh"

① Local State
(Component)

✓

~~React~~
{ Context
API }

② Props Drilling
(Passing data
b/w Components)

✓

③ State Management Lib.

✓

<Card Component />



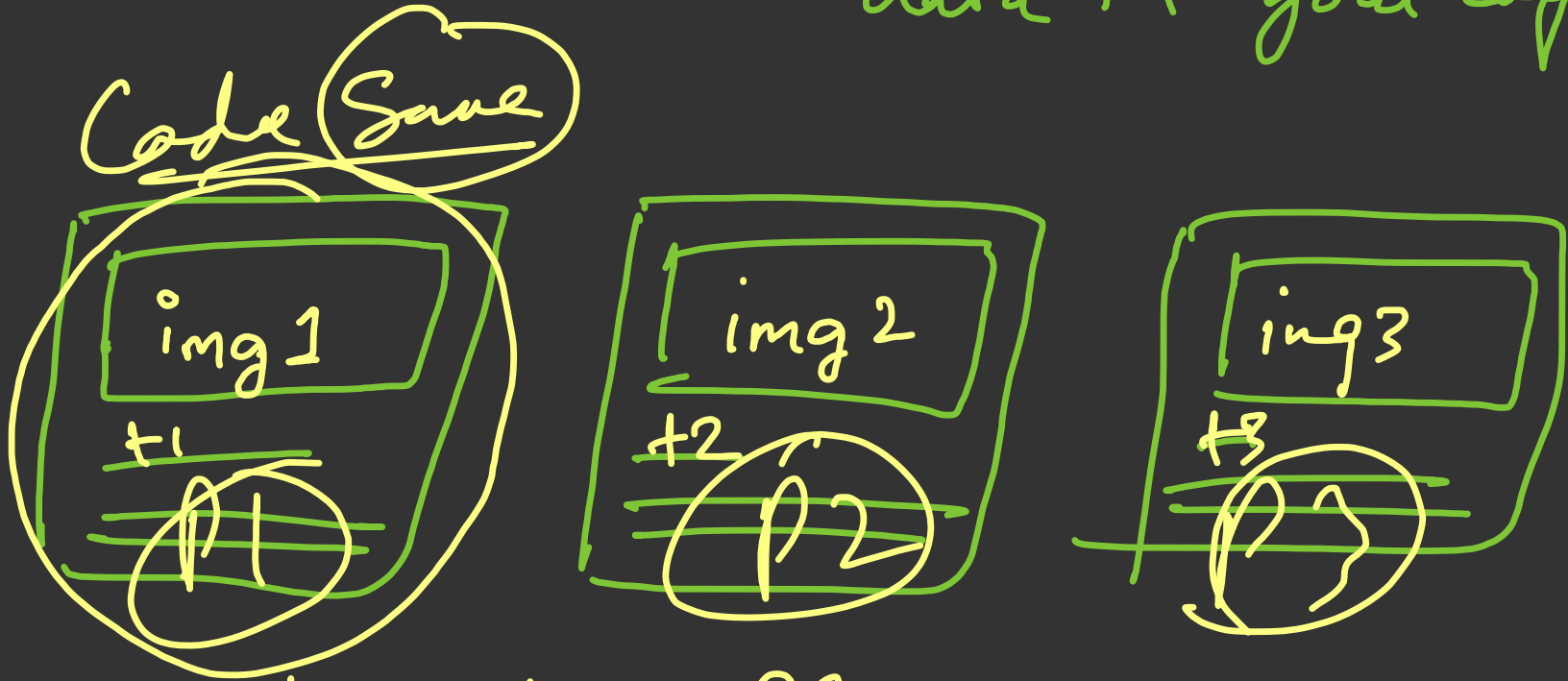
⇒ {img-url,
title,
description,}

{local
State}

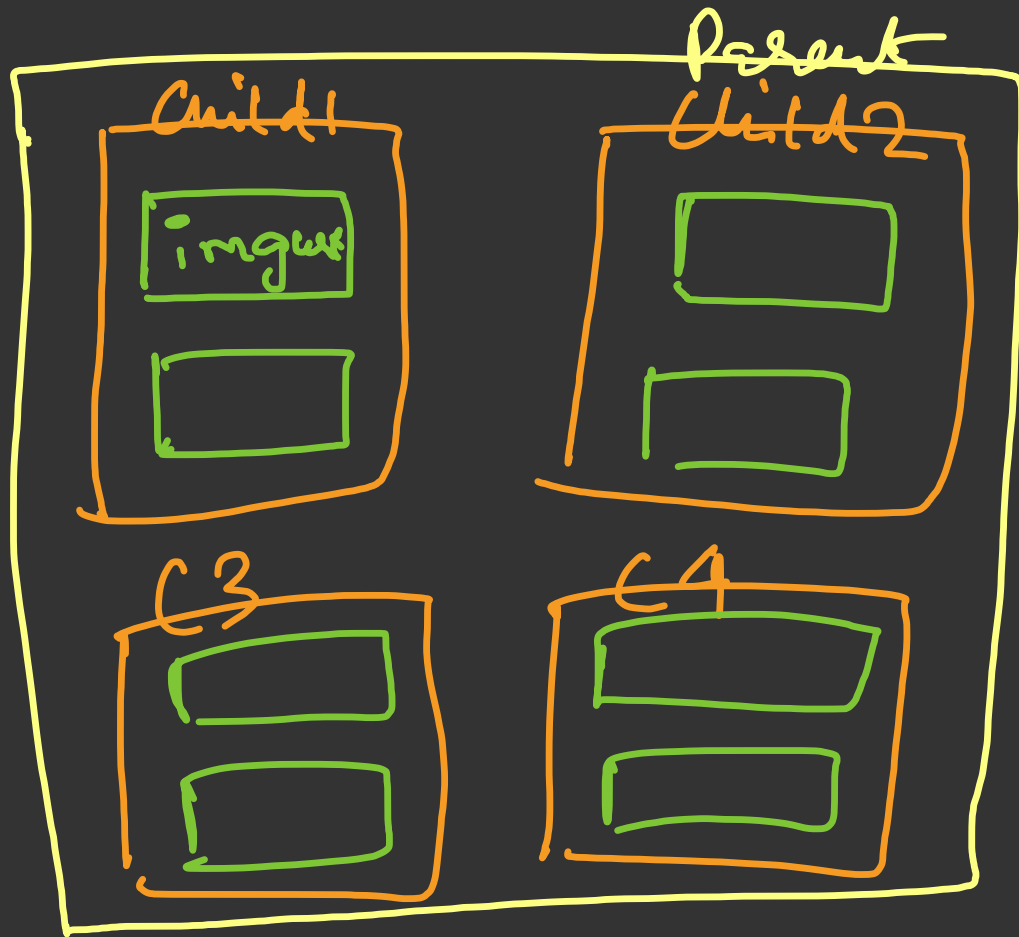
data



Local State → any dynamic data in your app.



data changes



Local State

$x \rightarrow$ Component level

Parent S.V.



Child



Grand Child



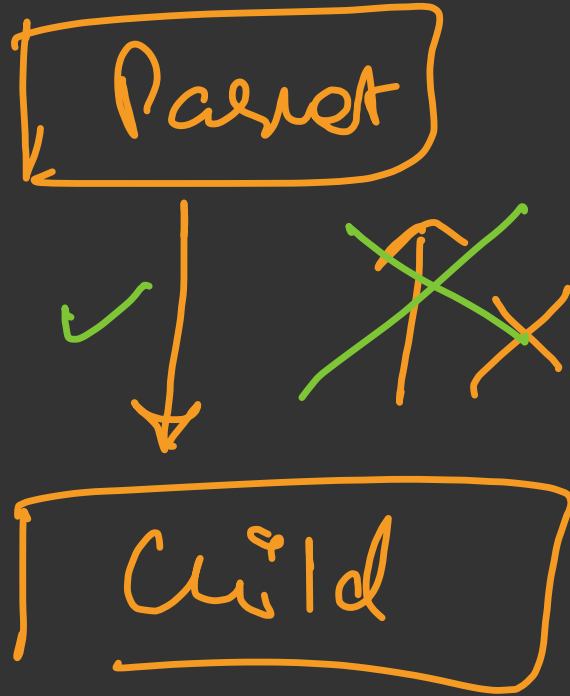
S-V



S.V.

State variable

Data flow - Unidirectional



Books

books = [B1, B2, B3]

T1: d1

T2: d2

T3: d3

Passing props

Parent
props

Child

P1



C1

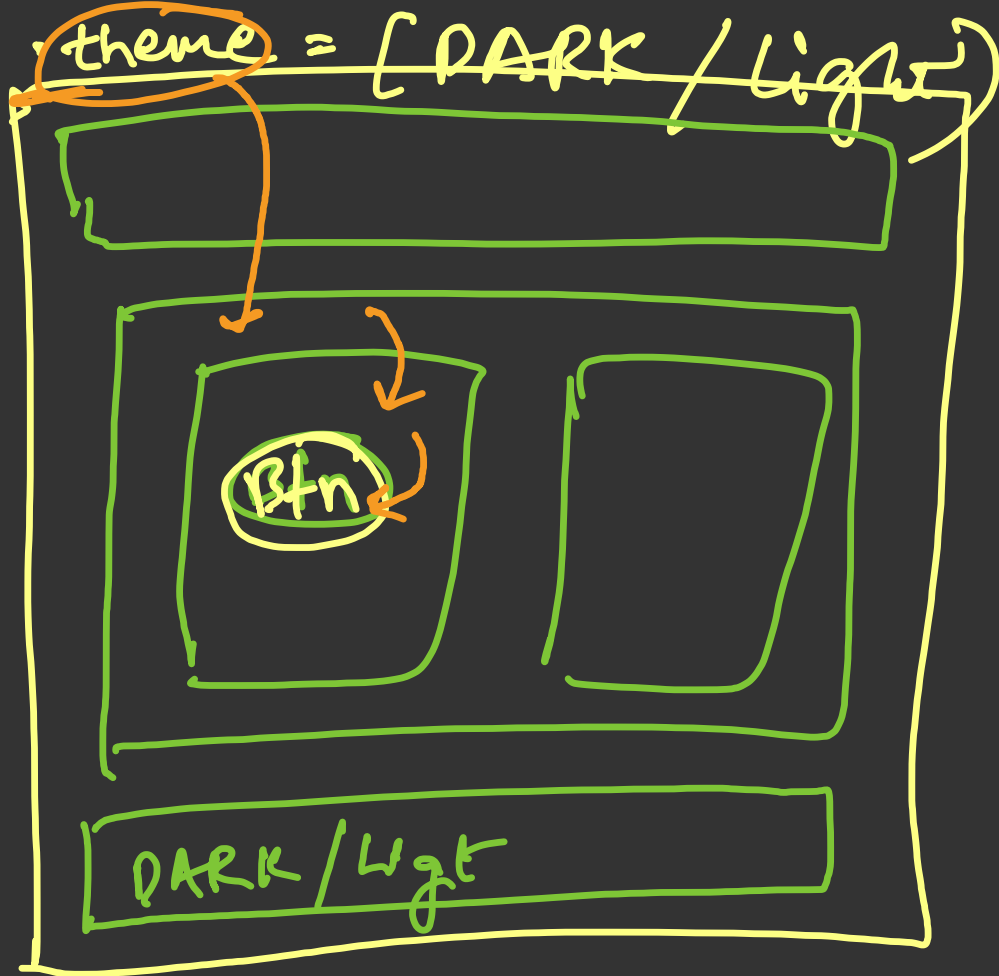


QC1

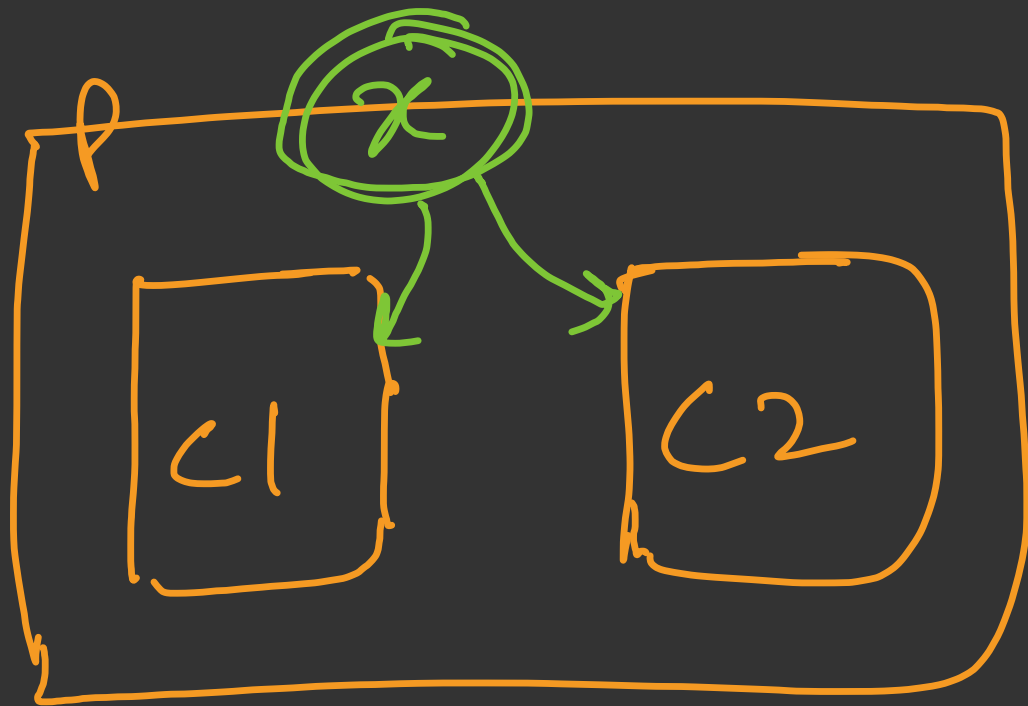


QCL

Pups drilling



Props de l'up

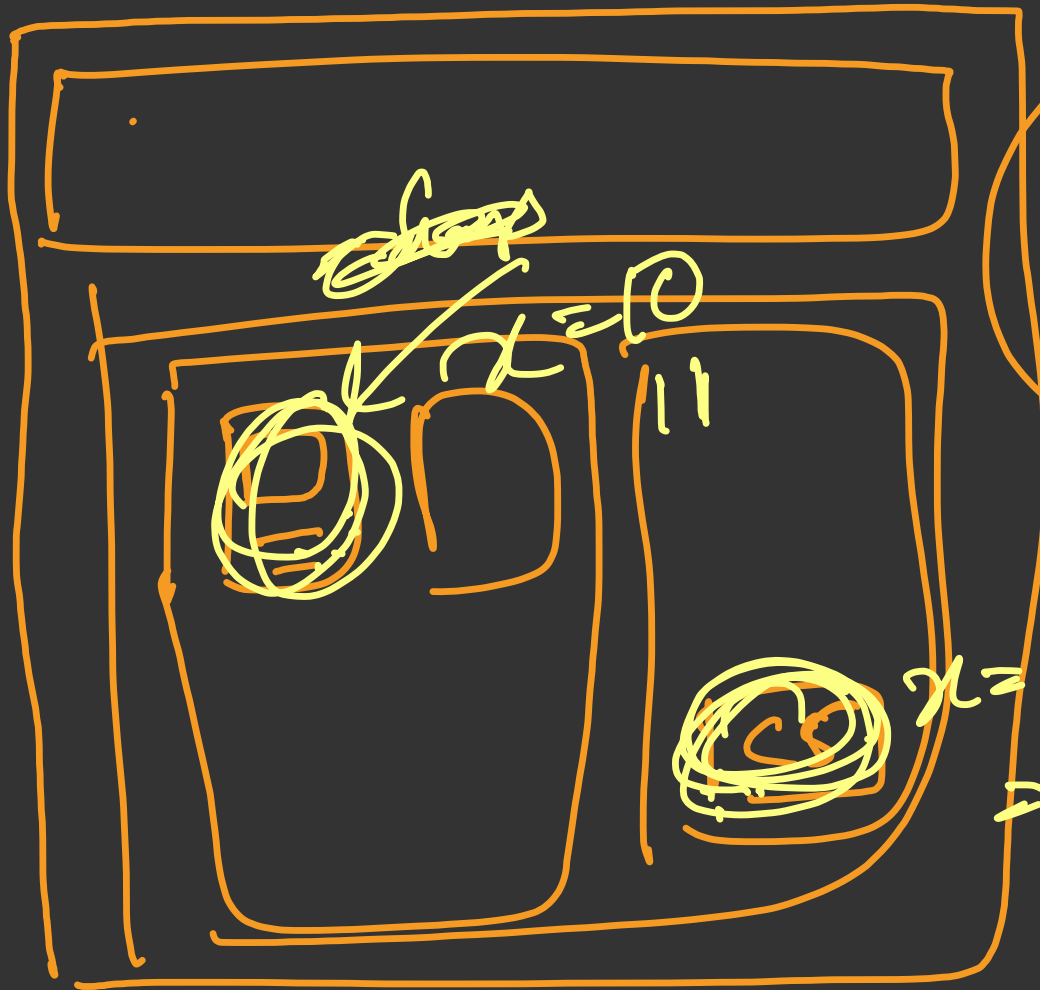


Lifting State
up

State Manger Libraries

Redux / ngRx

Zustand ↑↑↑

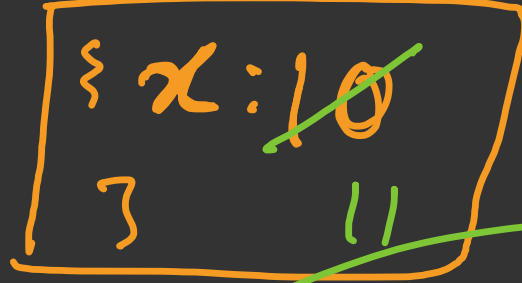


Huge
Carpet
Hierarchy

App

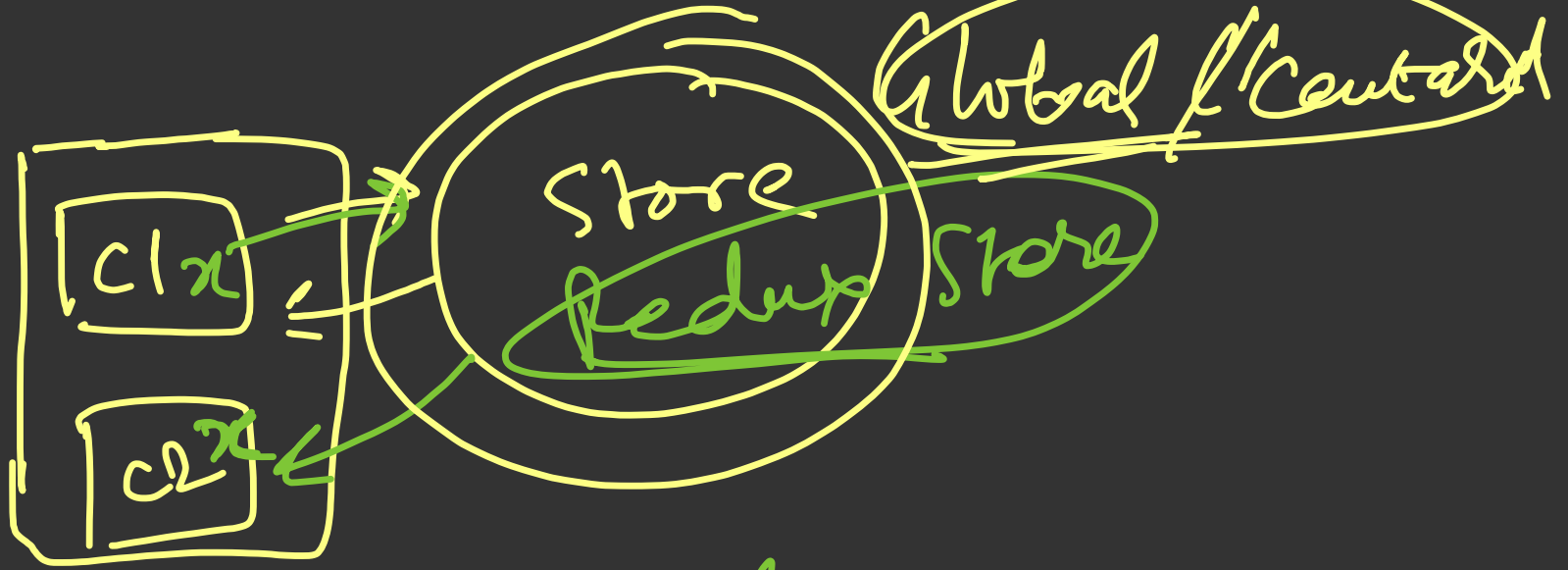


State (Central)



ReduxStore

c1-subscribed
c2 - to
state



#large scale

{ data management
(state)}

4
DON'T use State Management
library for very small
projects

Context
API

Debugging is easy

Actions & Reducer

Central Store