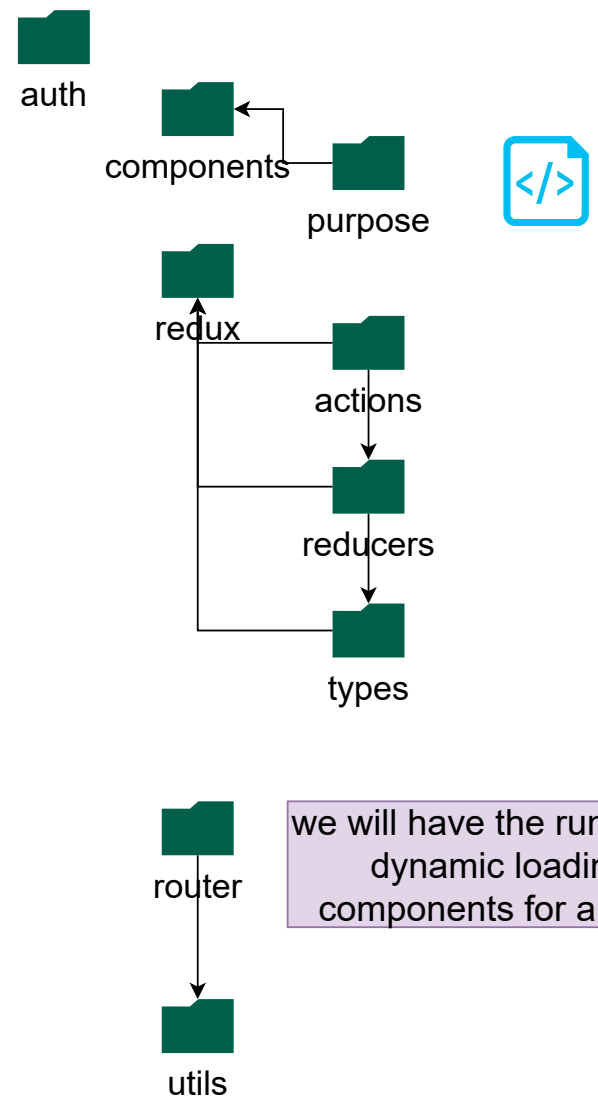


Component: small reusable piece of UI . in react ==>
we will use the js based functions.

There are 2 types of components :

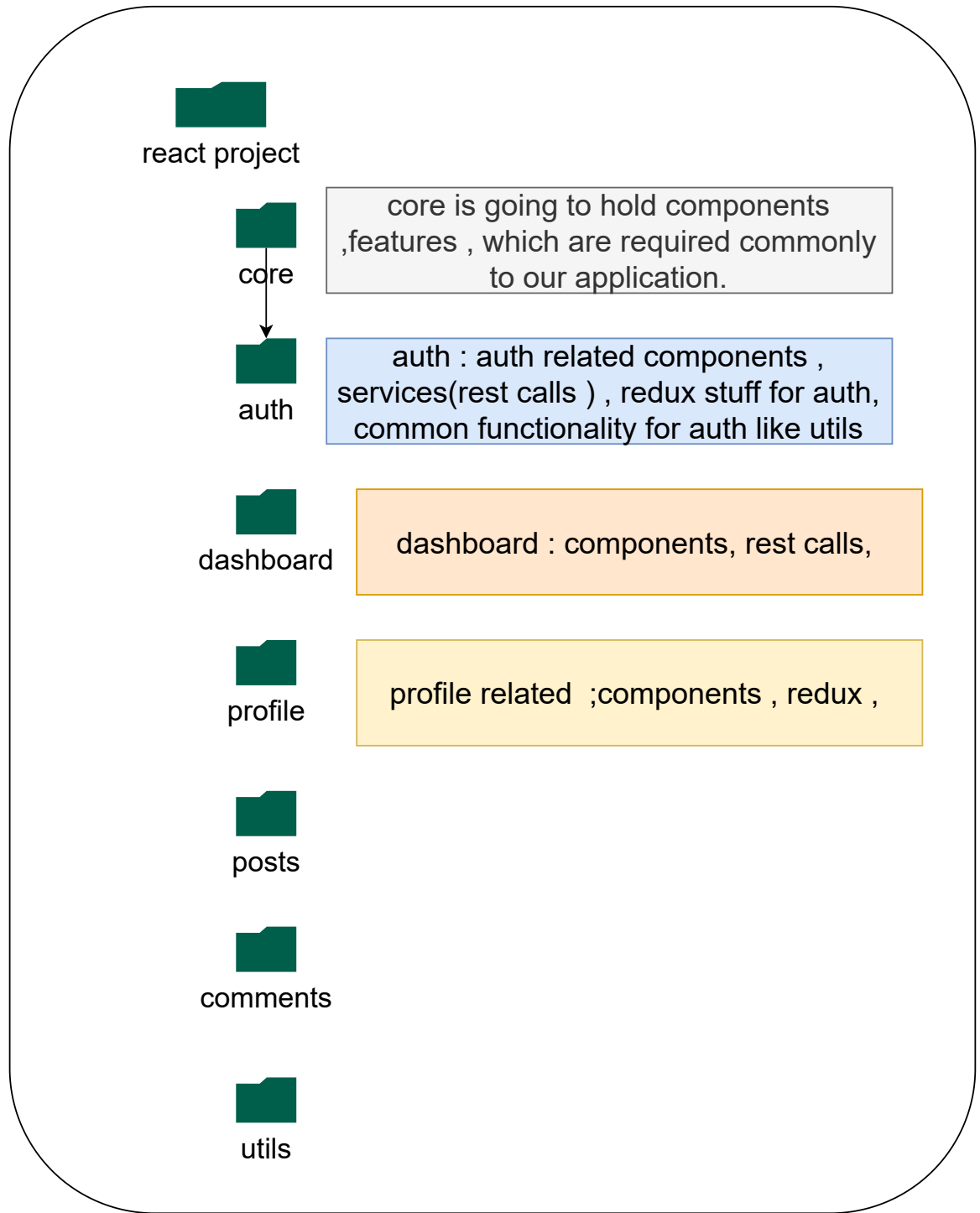
1. smart component : is a component which is capable of taking the decision / performing the task.
2. dumb component.: which is responsible for accepting / returning / sharing the data to UI.

**module : don't
module concept
consider module
nothing but a
separation of o
for our convin**

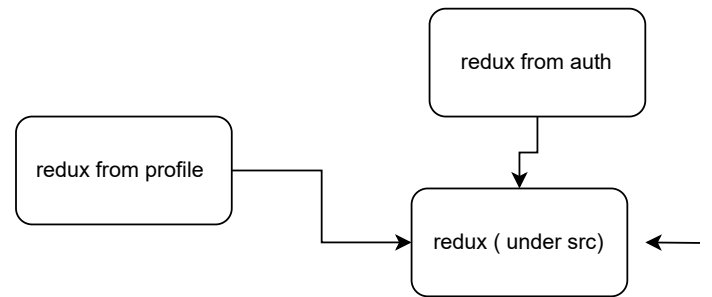


we will have the runtime
dynamic loading
components for auth

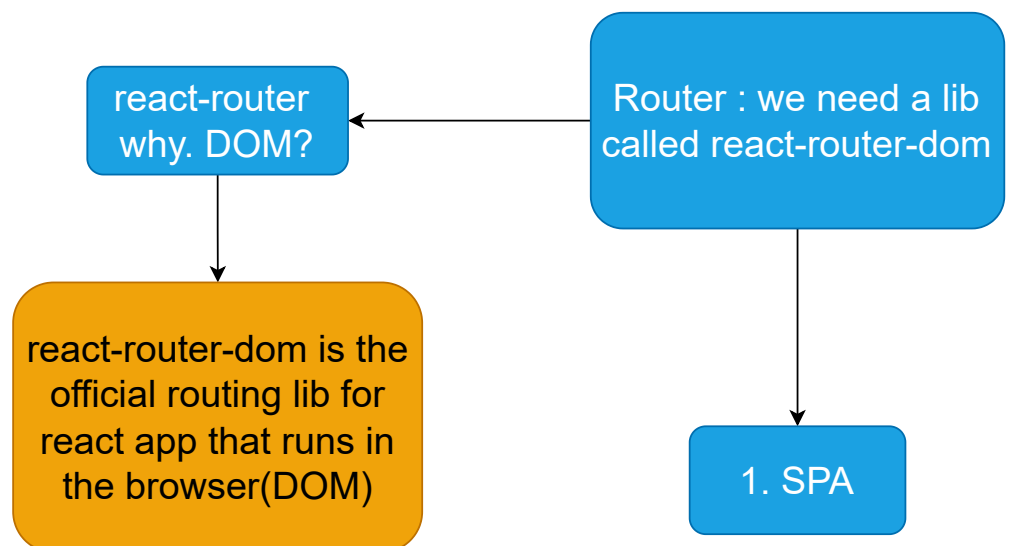
Don't have
it . we will
module is
a logical
our project
experience .

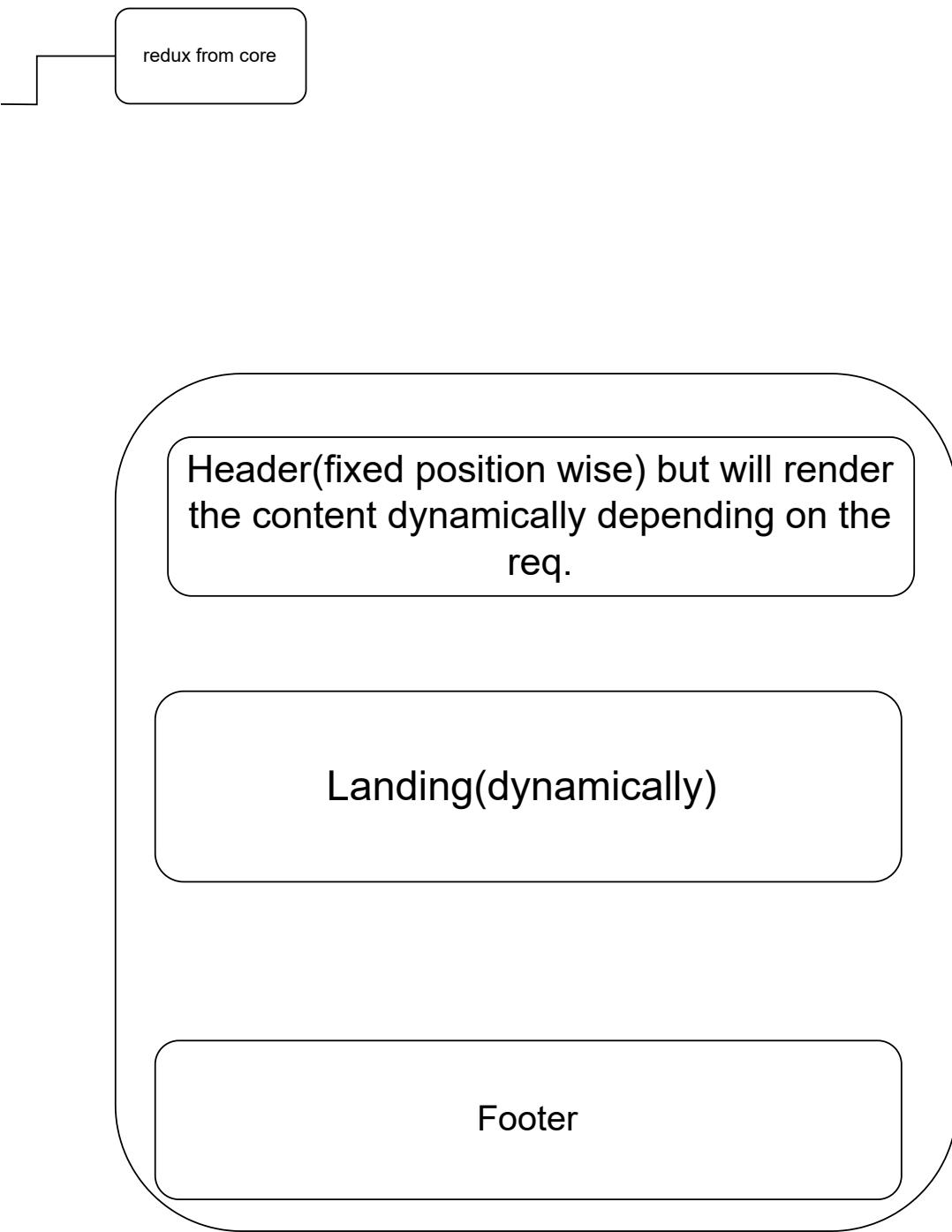


some loading /
for the
module .



HomePage





redux from core

Header(fixed position wise) but will render the content dynamically depending on the req.

Landing(dynamically)

Footer

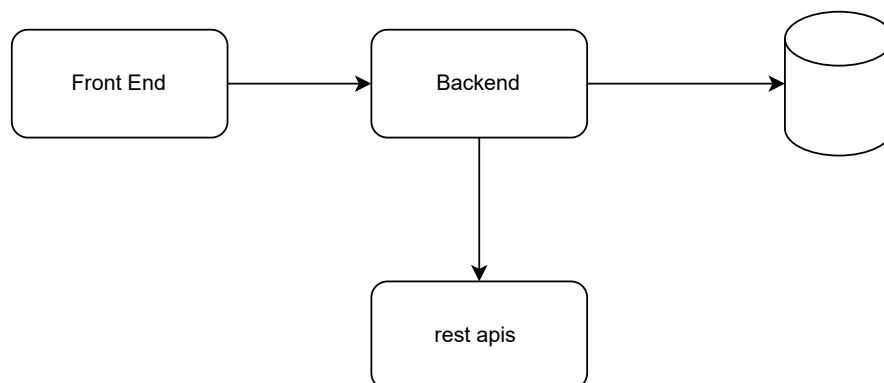
Form handling :

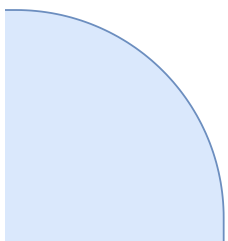
1. we have to define the onChange and onSubmit events .

onChange : when there is change in the input
onSubmit : to submit the form.

3. declare the state
4. to capture the updated values we have to use state in onChange.
5. onSubUmit : we will capture the data from state and we will utilize it for our rest call.

1. we must collect the form data
2. we mst print the form data.
3. we should perform hte rest call
4. we should handle post rest call part i.e. success or failure.





Email address

name@example.com

Example select

1

Example multiple select

1

2

3

4

Example textarea

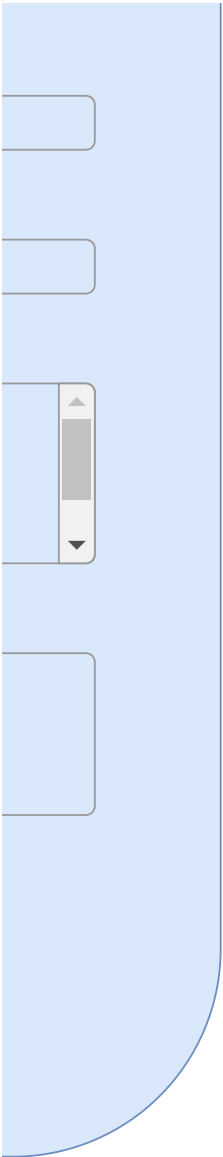
when u state changes , react re
render the data automatically.
this state its not a permant one
its in-memory one.

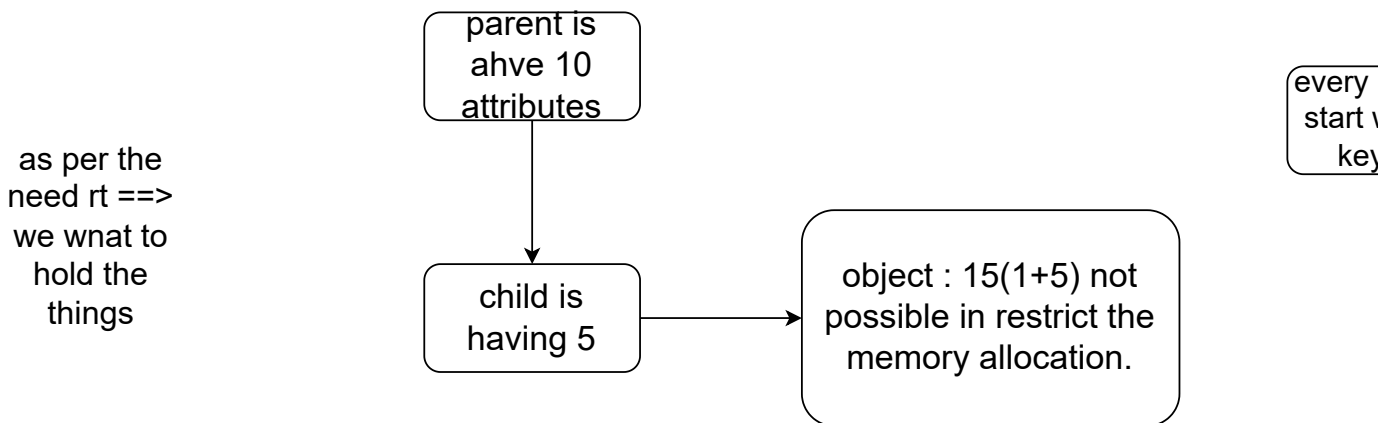
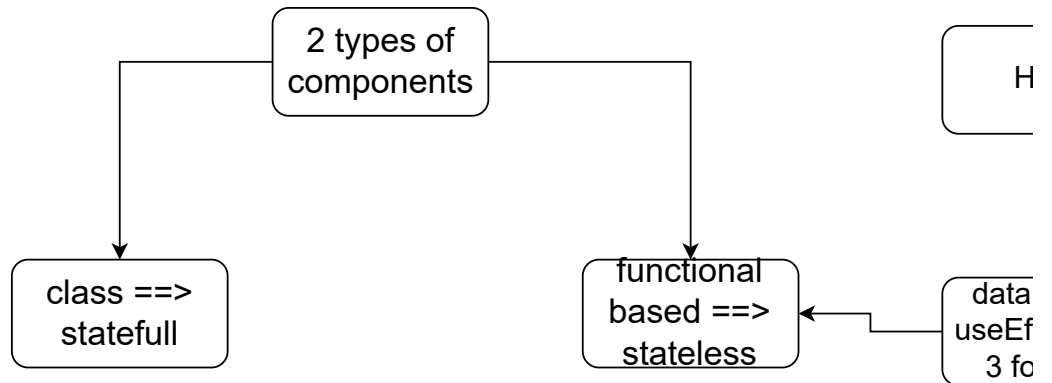
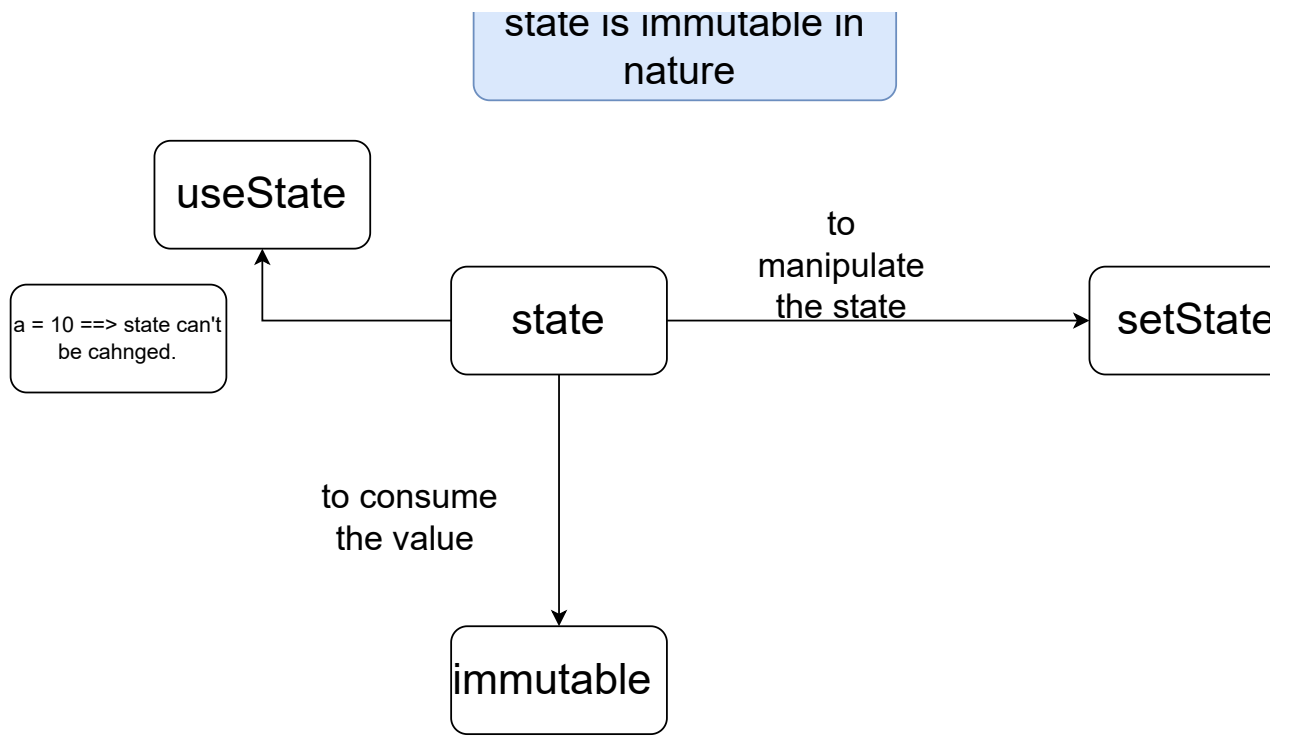
do we need some one to
hold the data (which is
accepted from the form.

state : its an object
that stores data
about a
component.

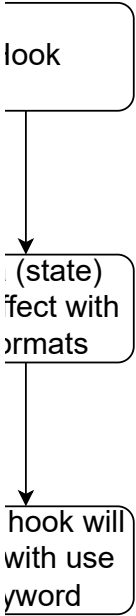
form

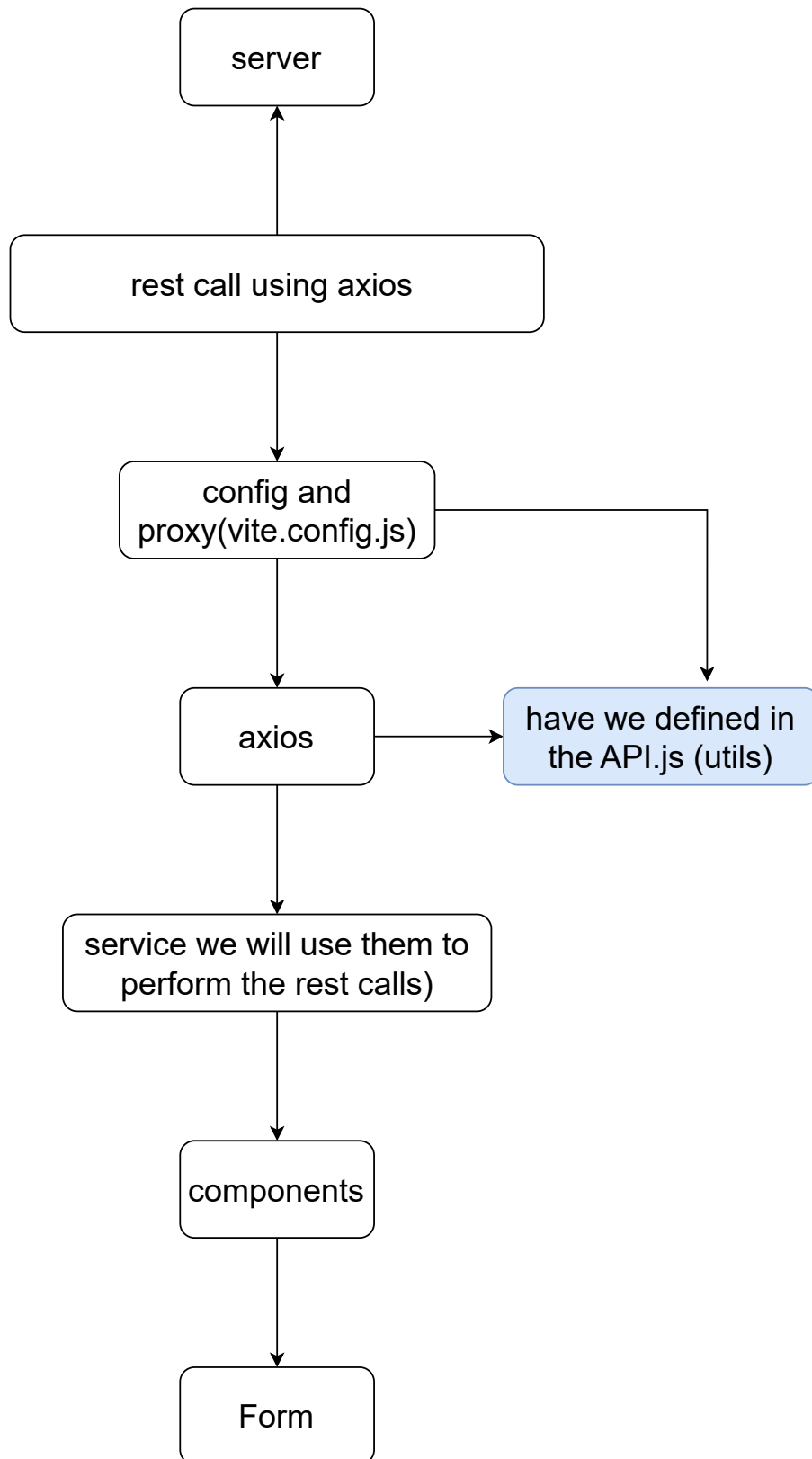
.....

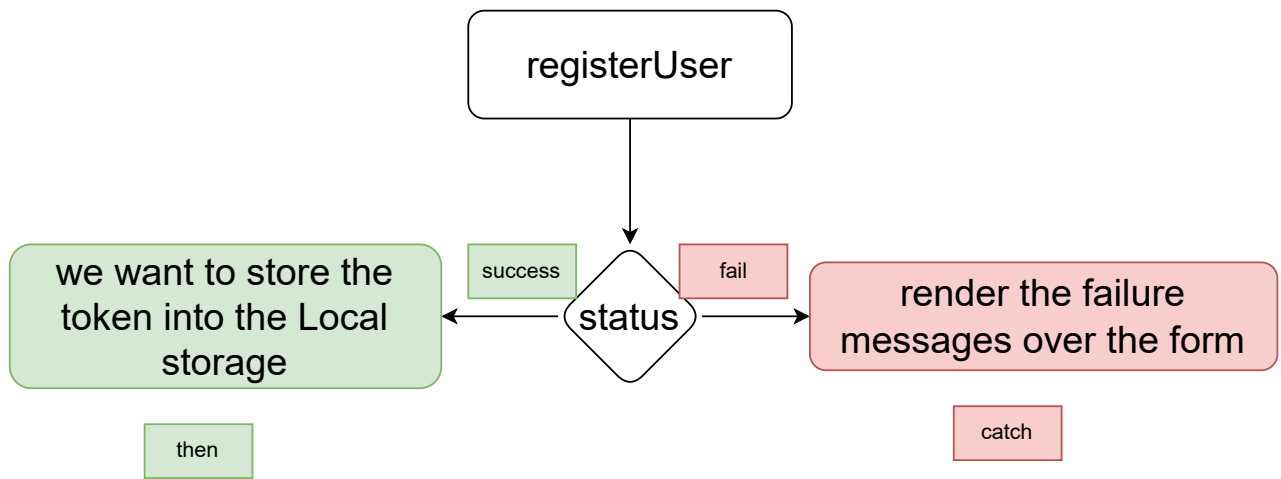




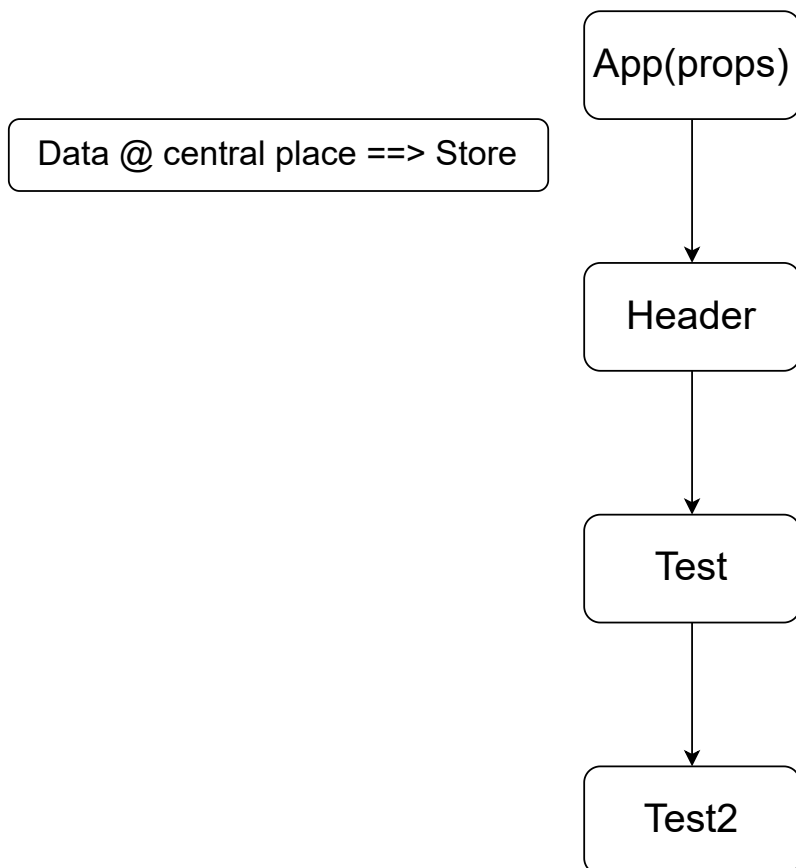
;







Props : they are used to pass data from a parent component down to a child component. They are read-only in nature==> a child component cannot modify the props it receives from the parent. but parent will change it then child will get the update.



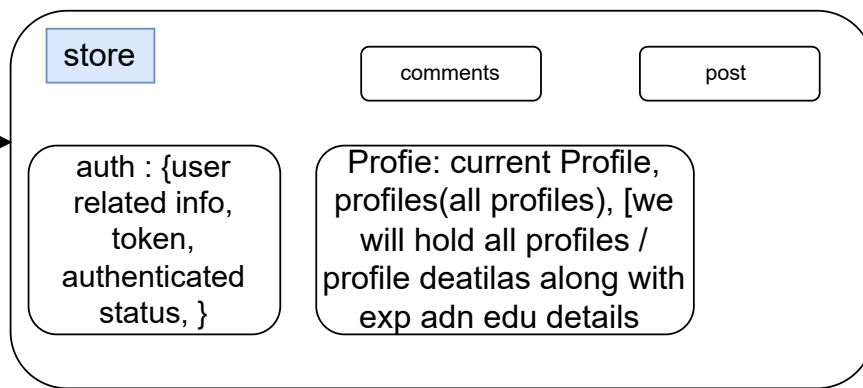
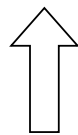
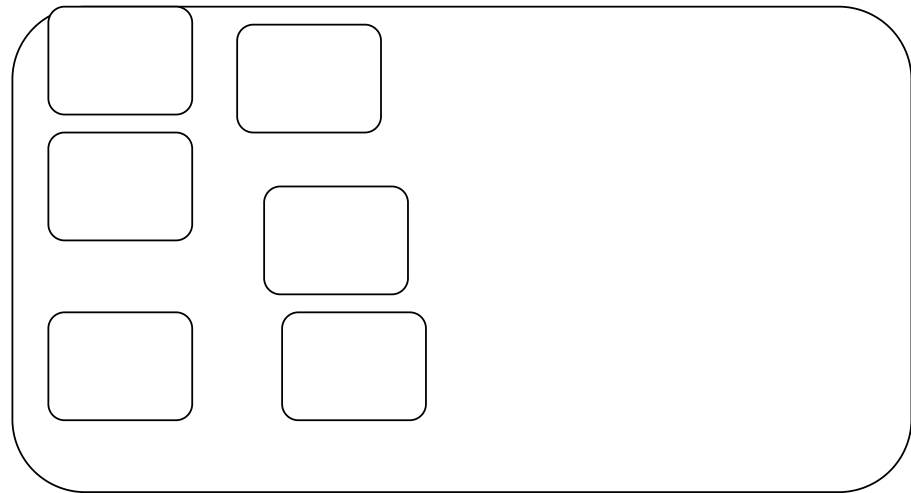
16.8 version onwards they give this facility to Functional component .

class based component

RCC would be the parent

, life cycle , states ==>
class based compl

Reducer :
manipulat



useSelector

immutable

: are responsible for
ion / update in store

actions must return the data and
will be returned directly to reducer
implicitly

Actions

useDispatch

useDispatch : is a hook ,
actions

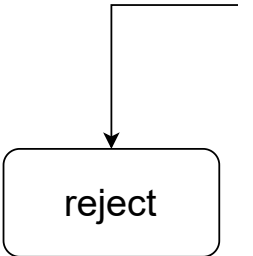
or

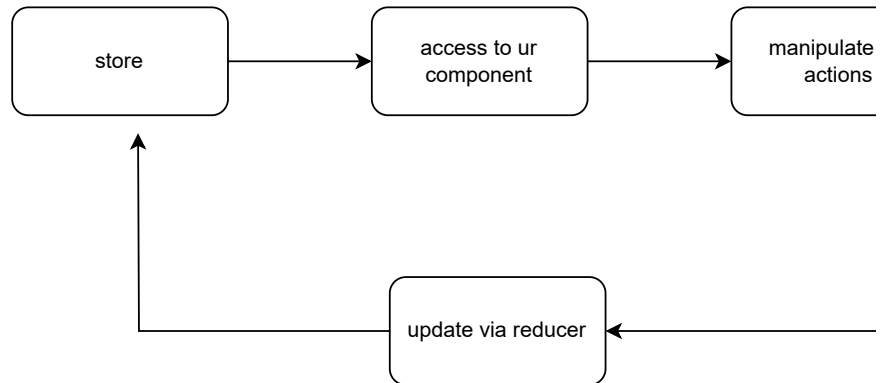
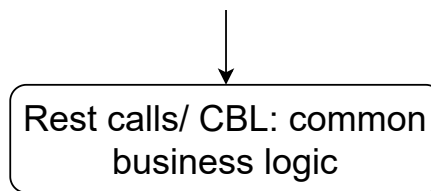


atch

to call

[





Slice : a slice is a redux toolkit (RTK) is a collection of state, reducers, actions.
state : its a part from ur store where ur specific module data will be managed.
reducers : how the state changes : it will update ur store accordingly.
actions : functions to make some changes as per the demand.
it is basically a modular piece of redux sture that groups logic for one feature / module like auth, cart, proifle .etc.

when we will perform the register user functionality

