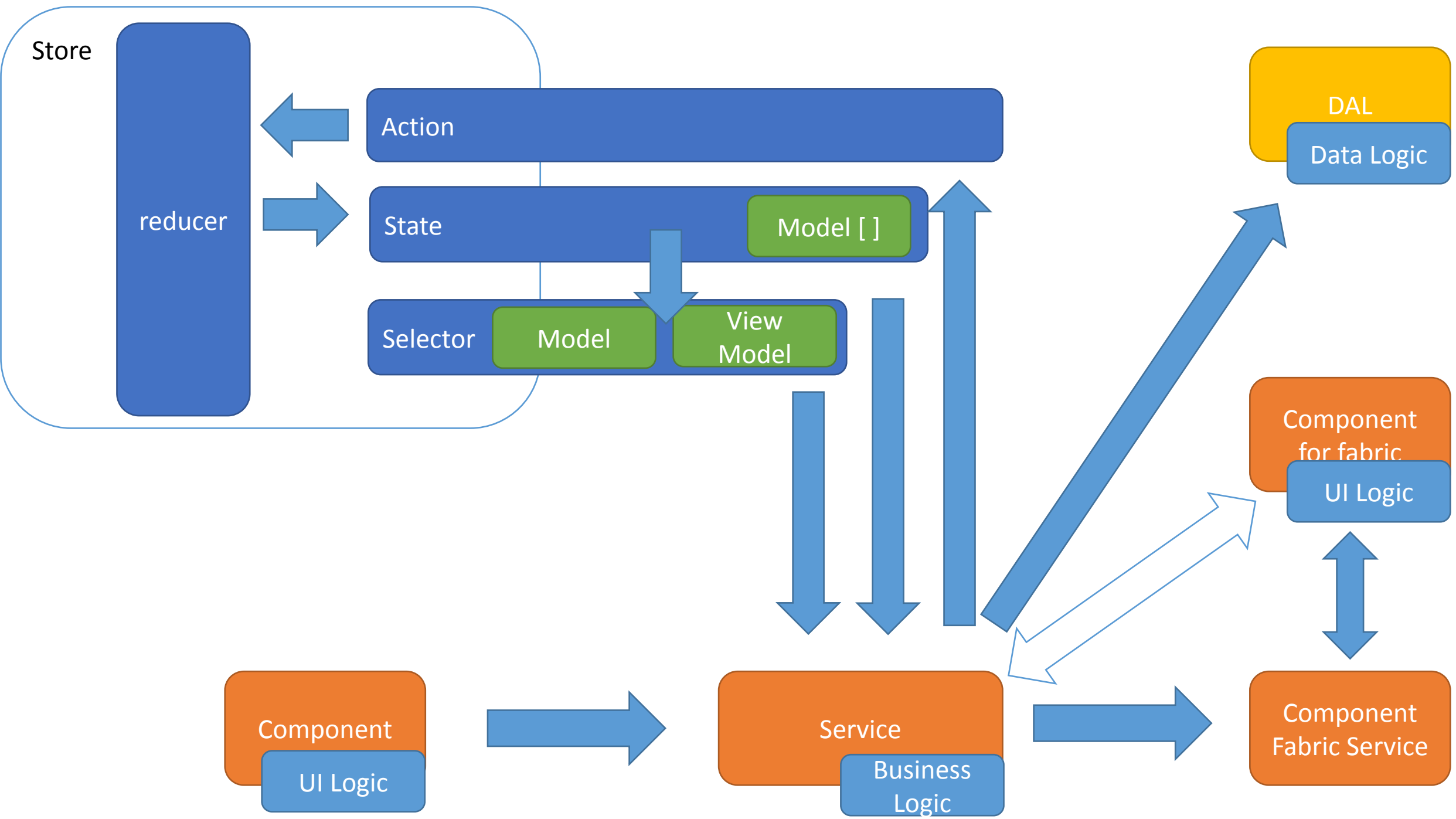


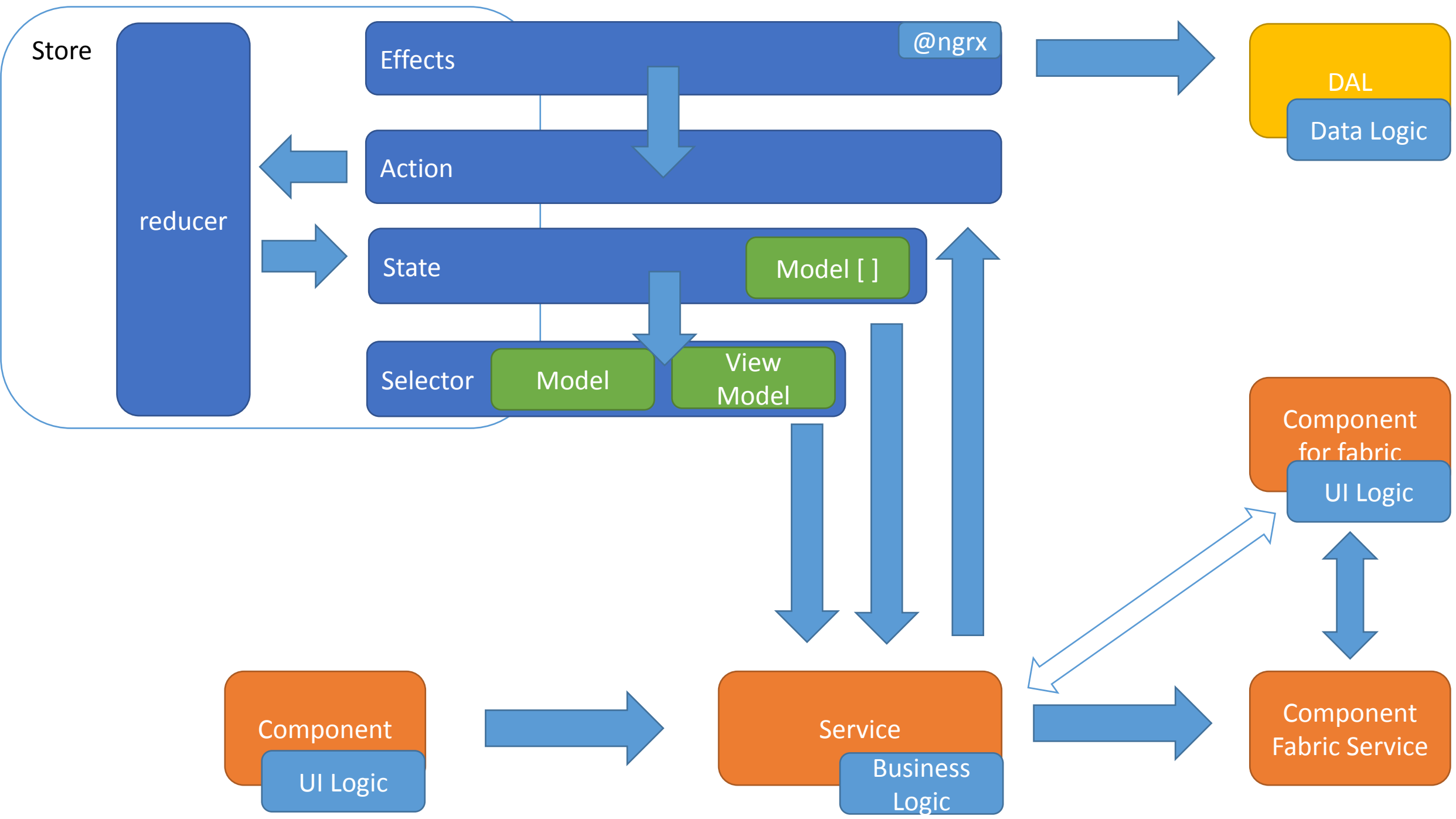
```
1 import * as layout from '../actions/layout';
2
3 export interface State {
4   showSidenav: boolean;
5 }
6
7 const initialState: State = {
8   showSidenav: false,
9 };
10
11 export function reducer(state = initialState, action: layout.Actions): State {
12   switch (action.type) {
13     case layout.CLOSE_SIDENAV:
14       return {
15         showSidenav: false,
16       };
17
18     case layout.OPEN_SIDENAV:
19       return {
20         showSidenav: true,
21       };
22
23     default:
24       return state;
25   }
26 }
```




```
import { createSelector } from 'reselect'

const getVisibilityFilter = (state) => state.visibilityFilter
const getTodos = (state) => state.todos

export const getVisibleTodos = createSelector(
  [ getVisibilityFilter, getTodos ],
  (visibilityFilter, todos) => {
    switch (visibilityFilter) {
      case 'SHOW_ALL':
        return todos
      case 'SHOW_COMPLETED':
        return todos.filter(t => t.completed)
      case 'SHOW_ACTIVE':
        return todos.filter(t => !t.completed)
    }
  }
)
```



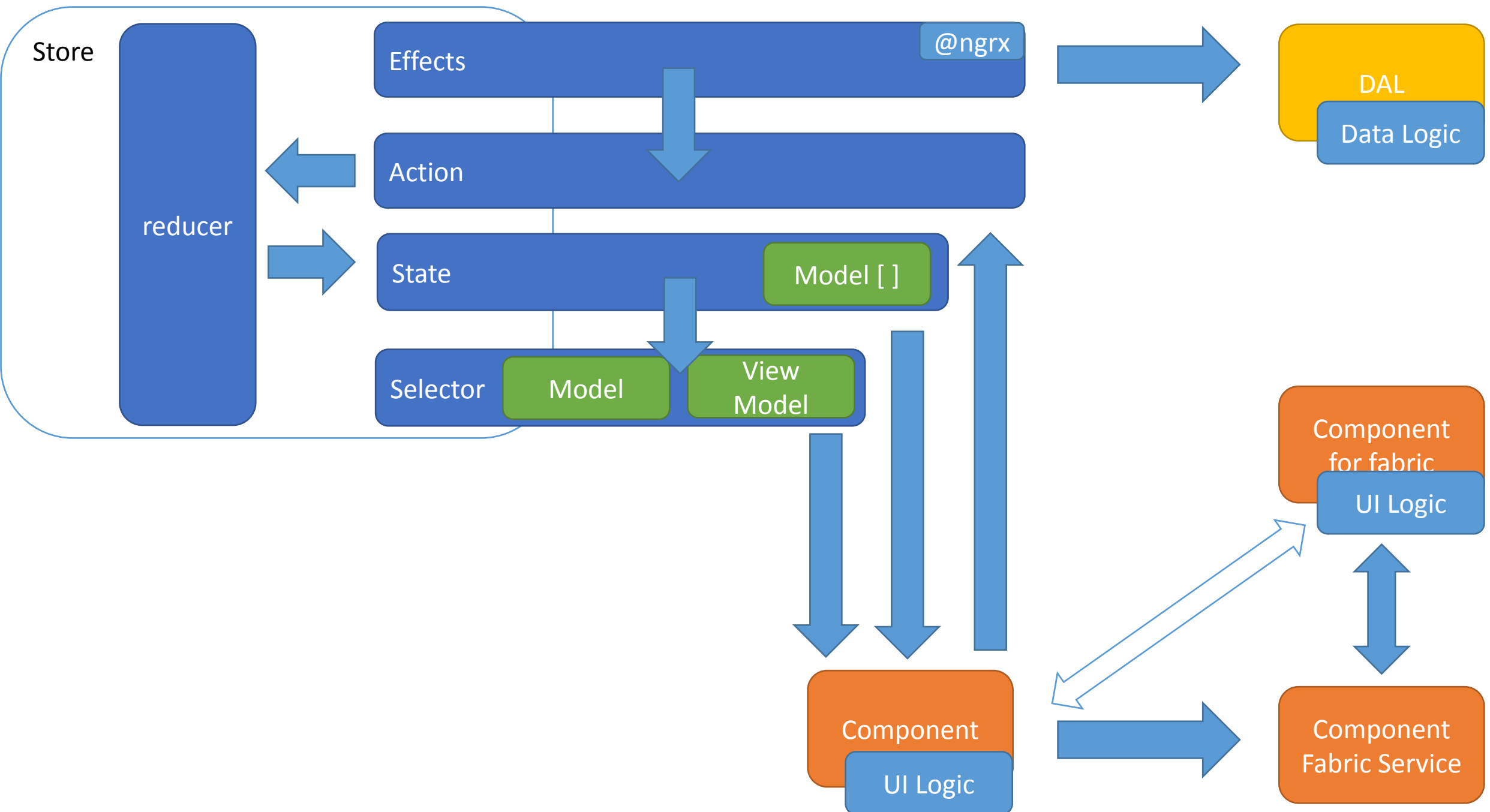
```
//category.effects.ts

import {Injectable} from '@angular/core';
import {Effect, Actions} from '@ngrx/effects';

import ...

...
@Injectable()
export class CategoryEffects {
  constructor (
    private actions$: Actions,
    private categoryActions: CategoryActions,
    private svc: CategoryService
  ) {}

  @Effect ()
  loadCategories$ = this.actions$
    .ofType (CategoryActions.LOAD_CATEGORIES)
    .switchMap (() => this.svc.getCategories ())
    .map ((categories: Category []) =>
this.categoryActions.loadCategoriesSuccess (categories))
}
```



Angular Detection Strategy

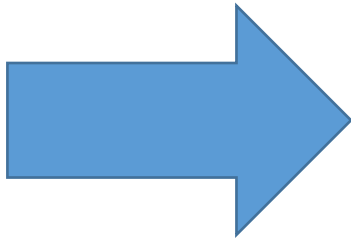
OnPush

Default

Redux

State

Selector

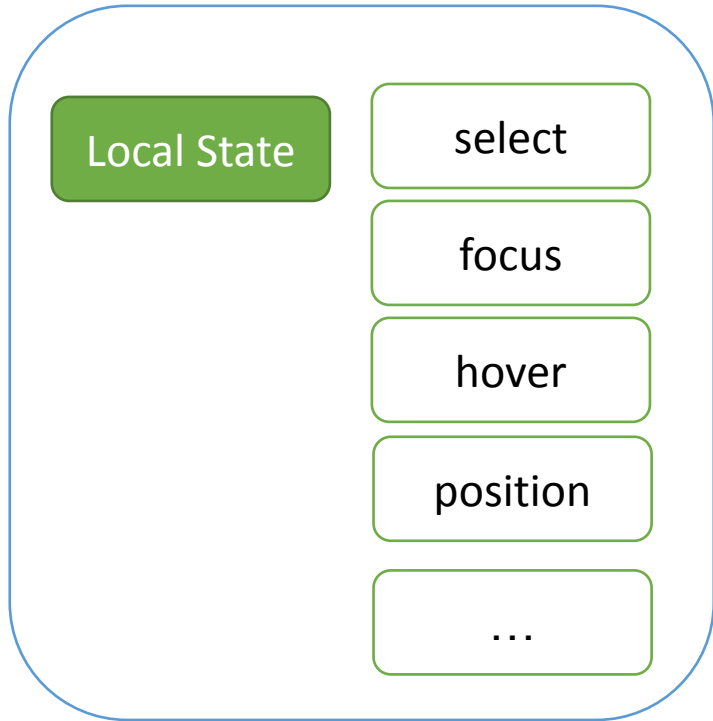


changes

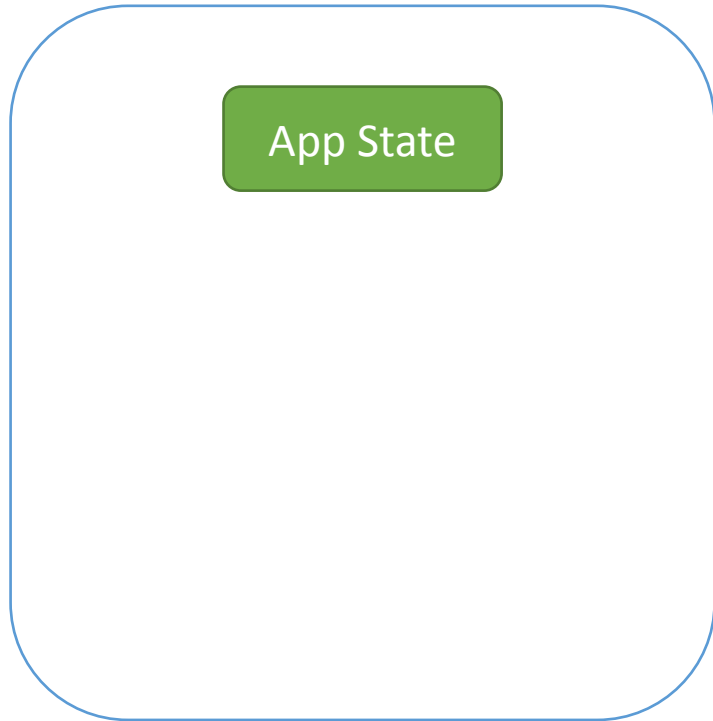
===

reference

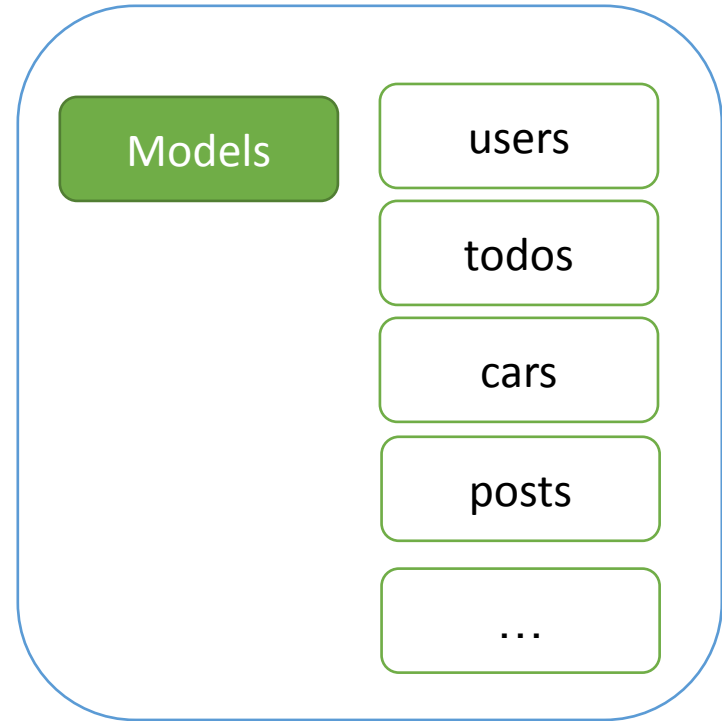
value



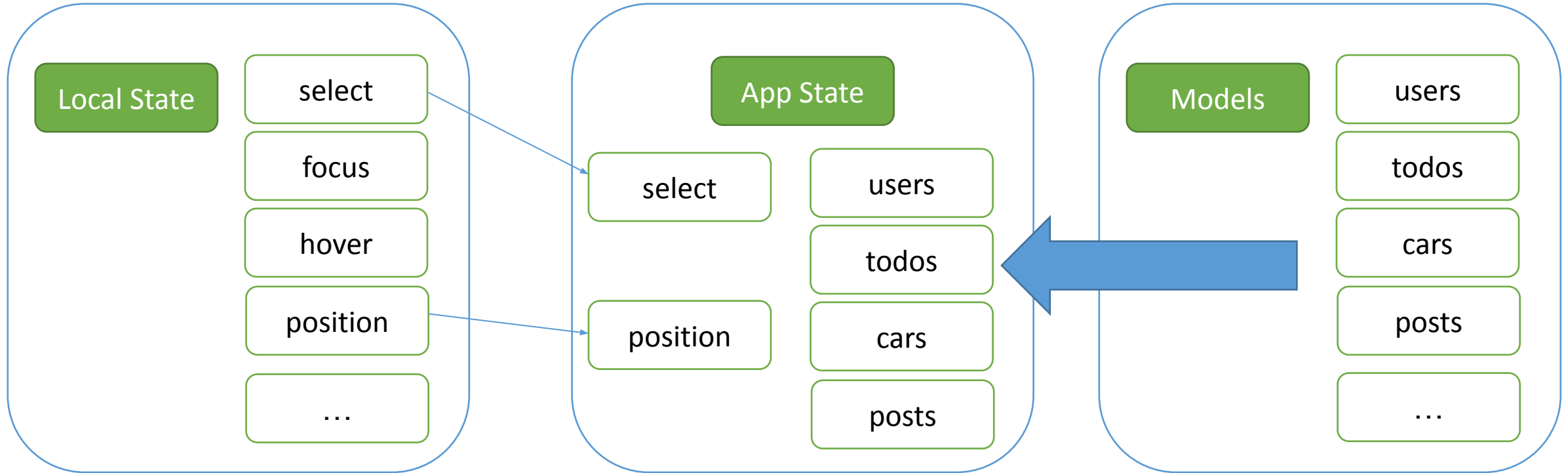
UI / Service



REDUX



Server



UI / Service

REDUX

Server

App State

user

select

App State

ui state /
local state

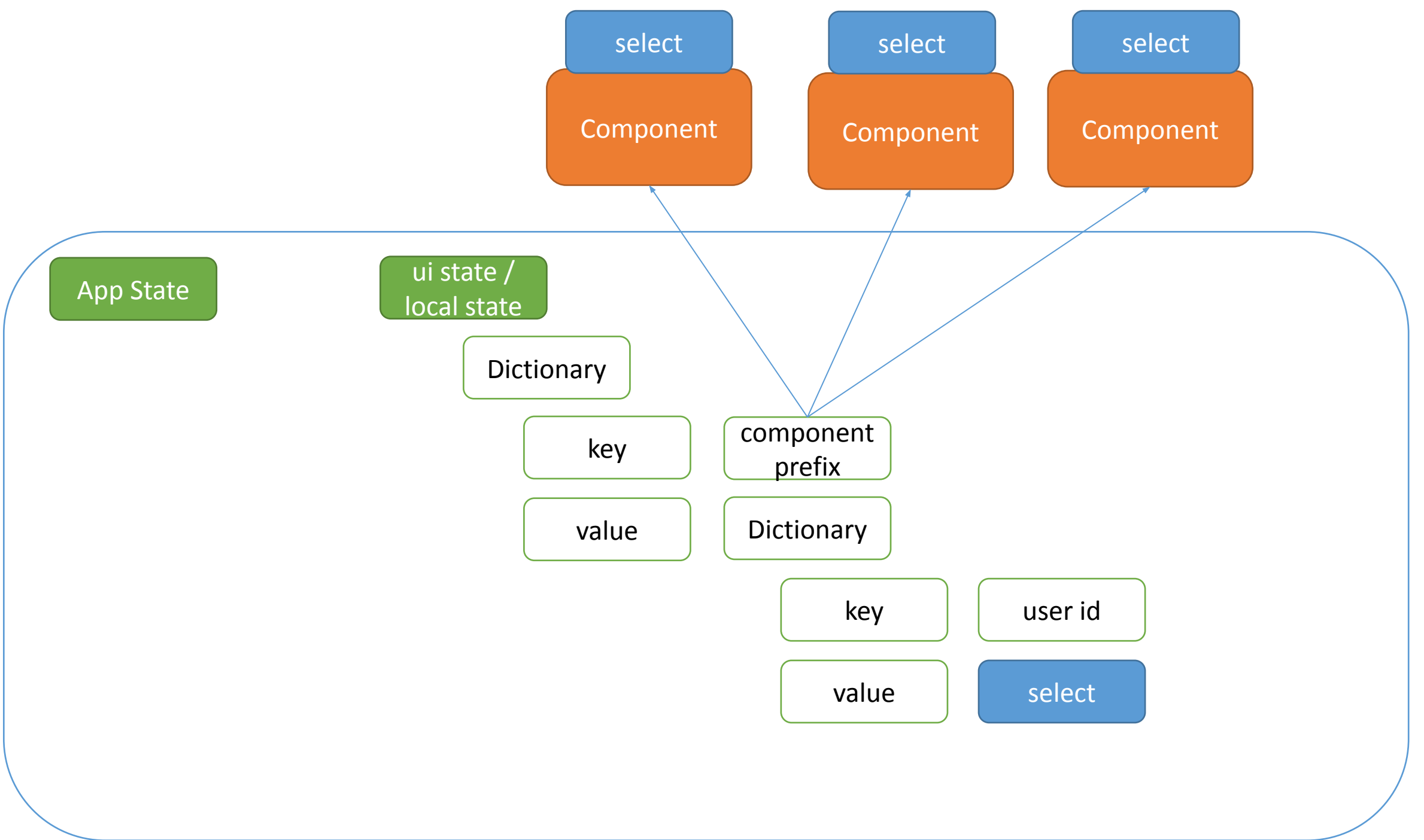
Dictionary

key

user id

value

select



App State

user

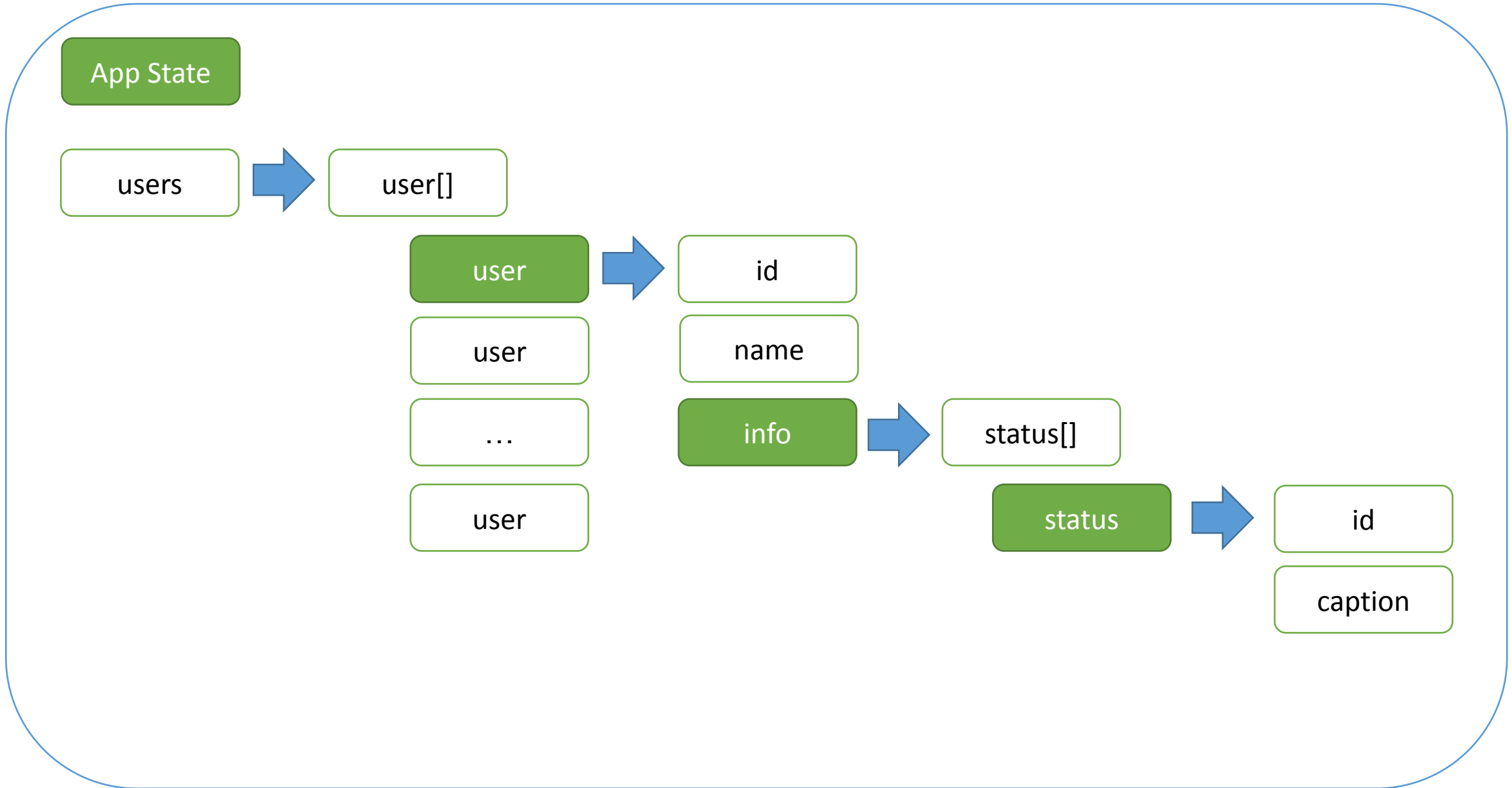
Dictionary

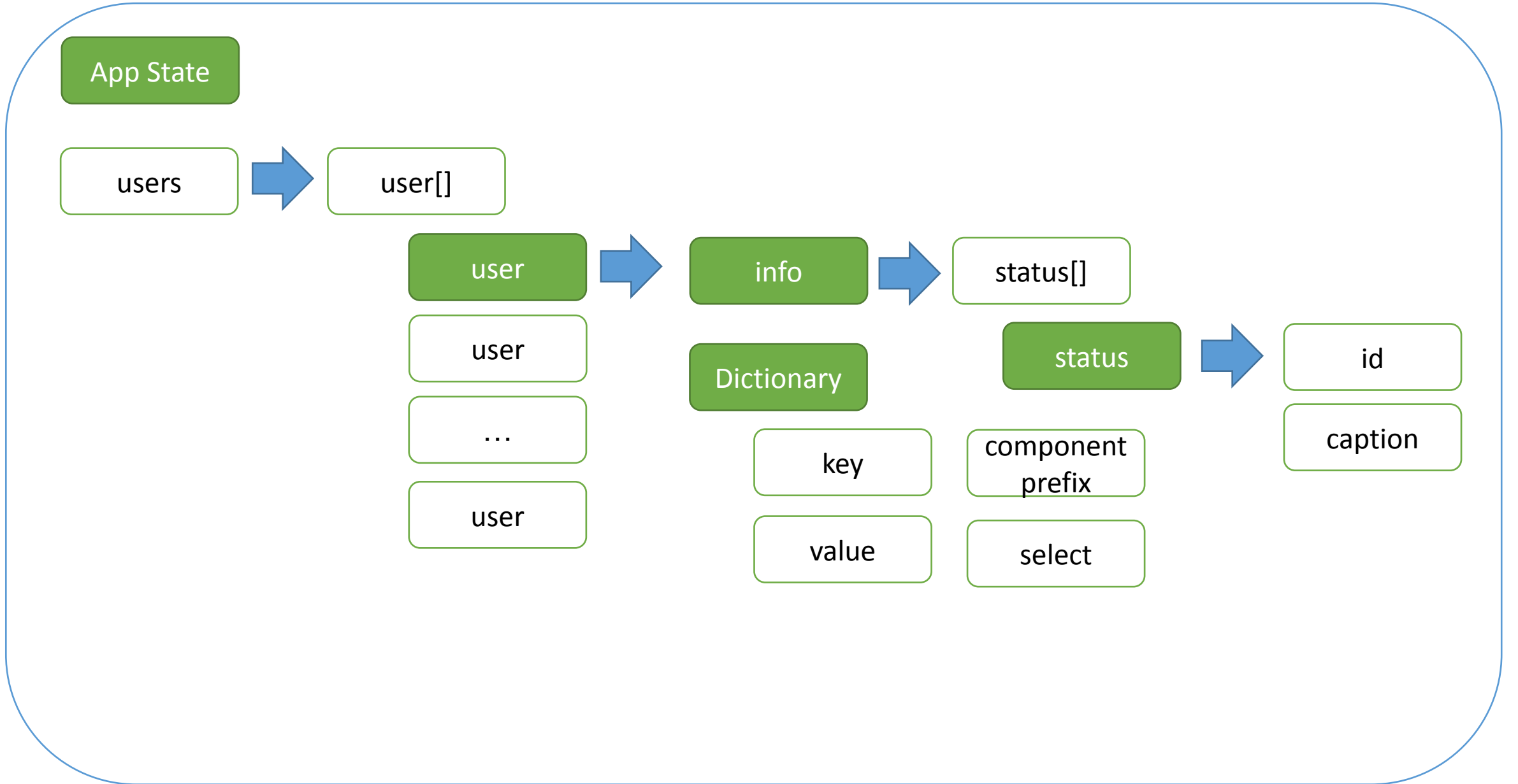
key

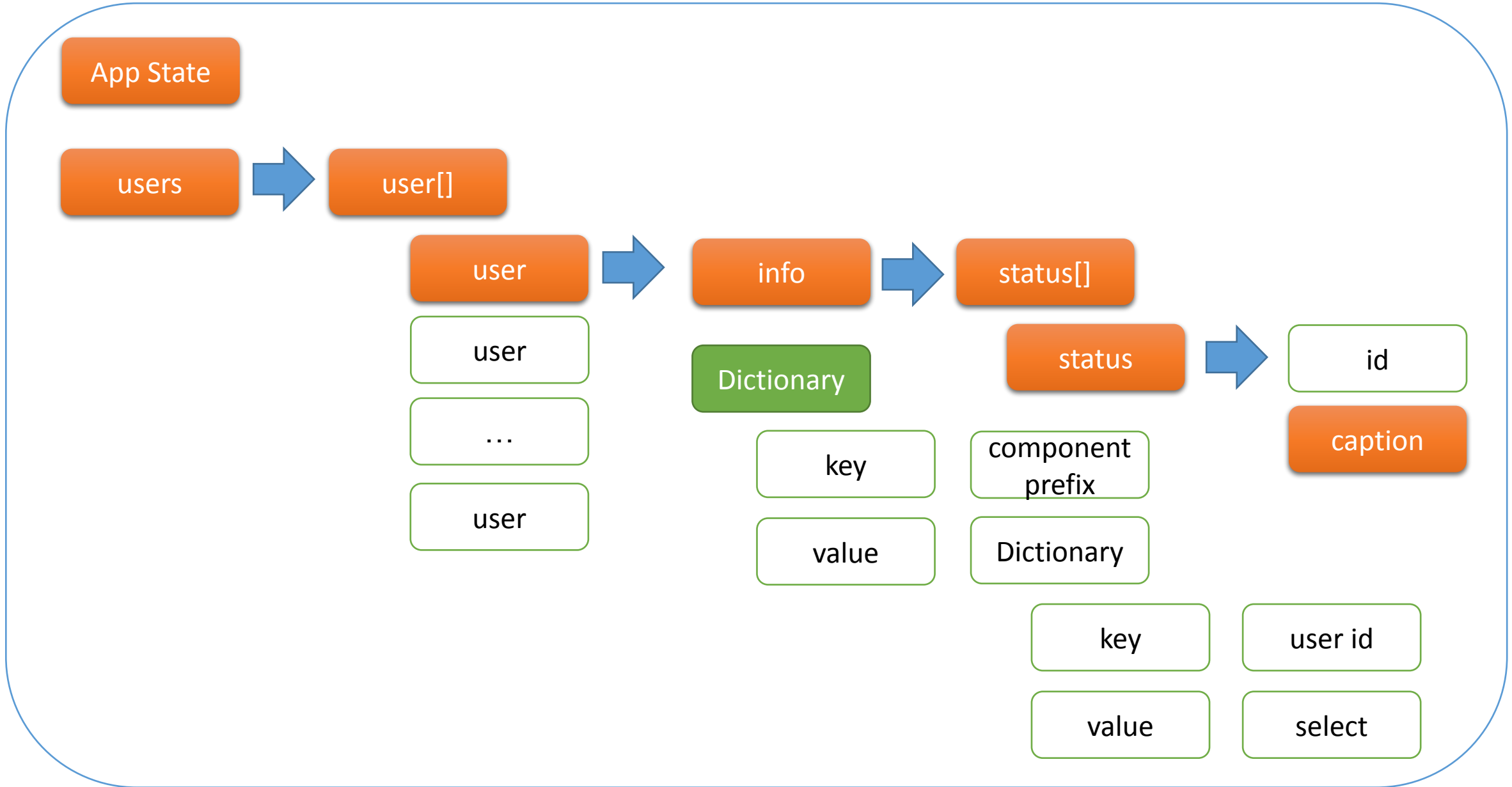
component
prefix

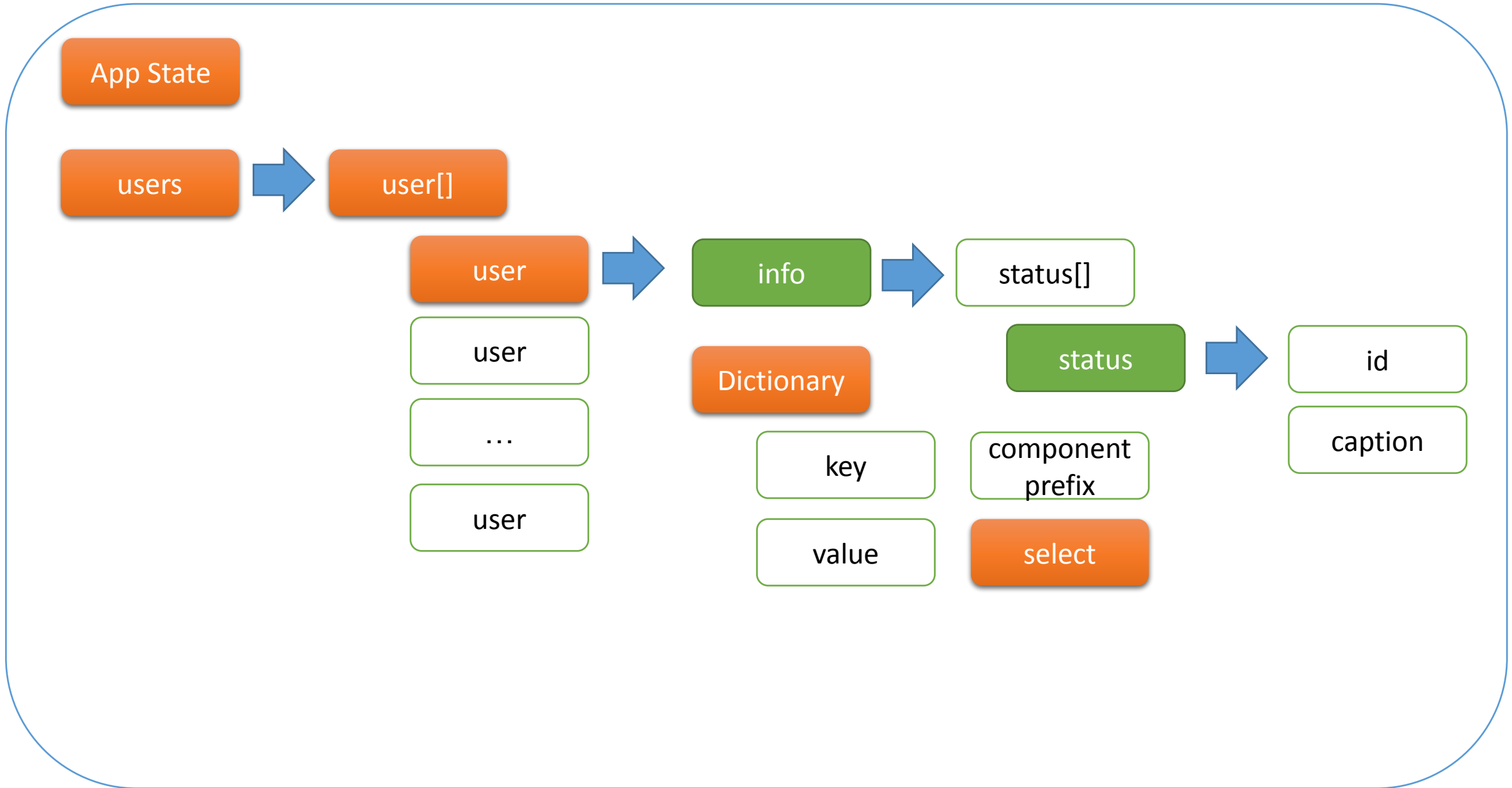
value

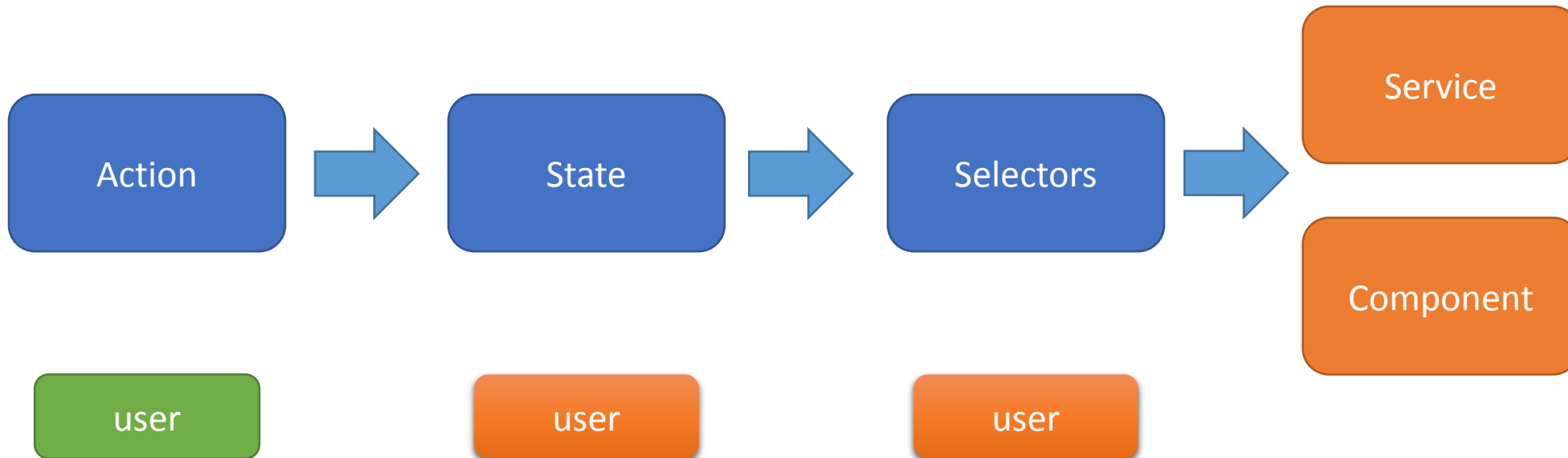
select

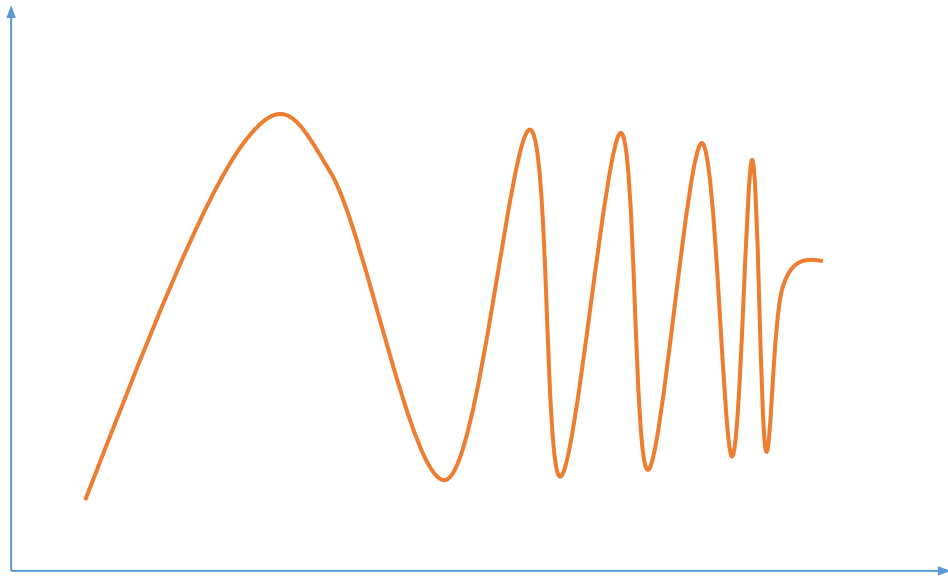




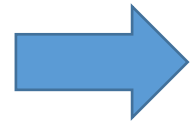






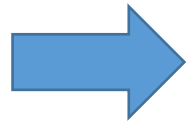


user



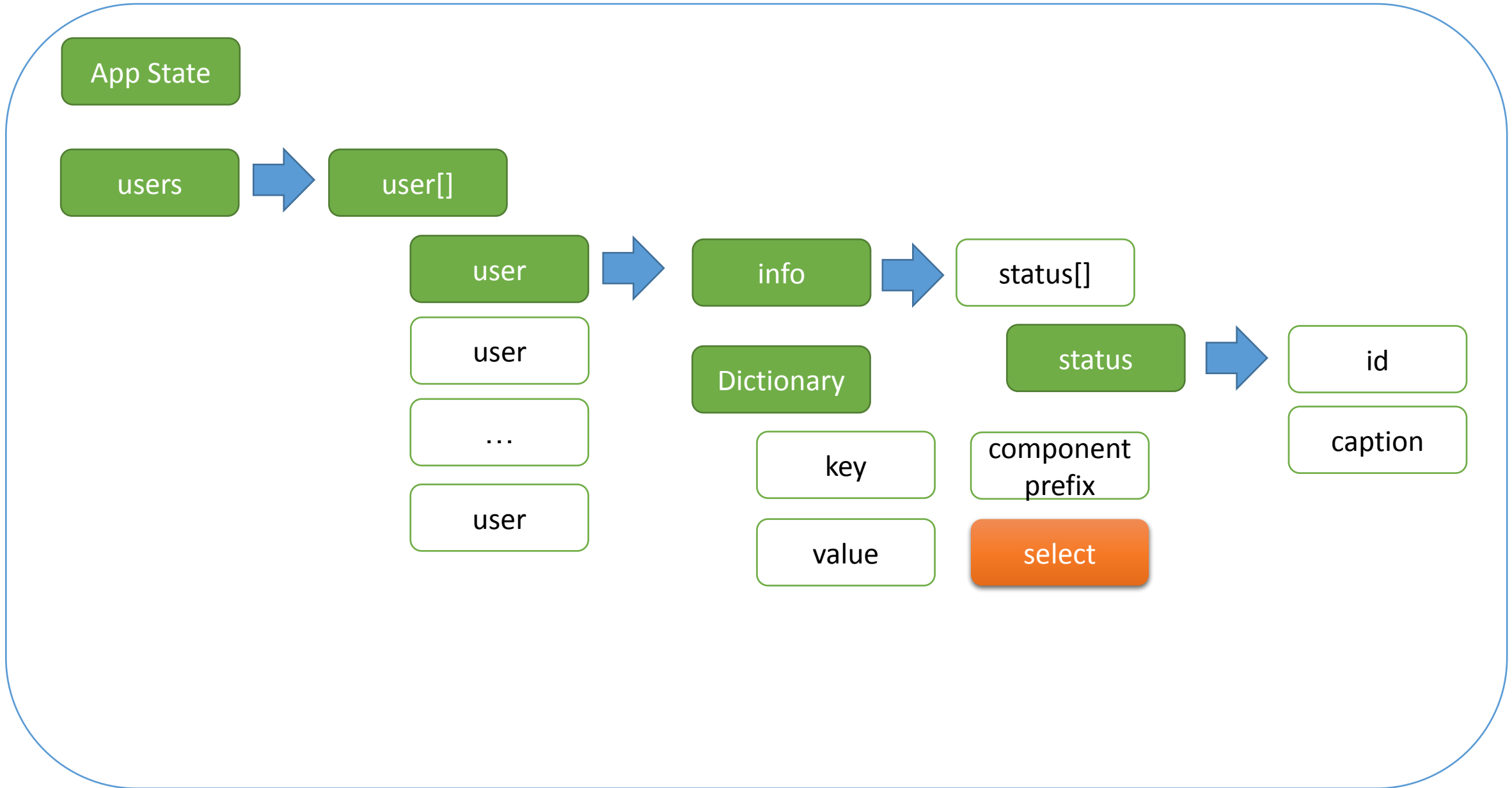
rendering

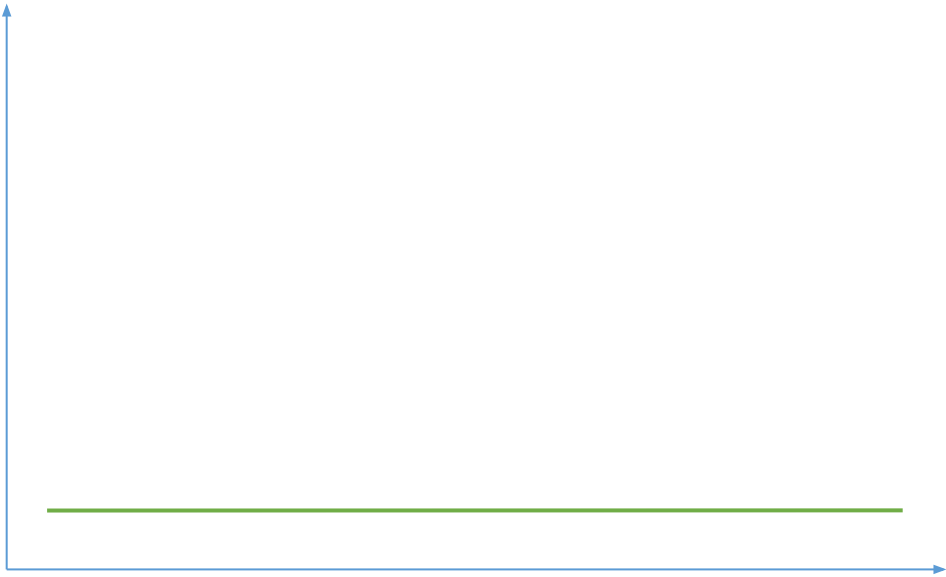
calculation



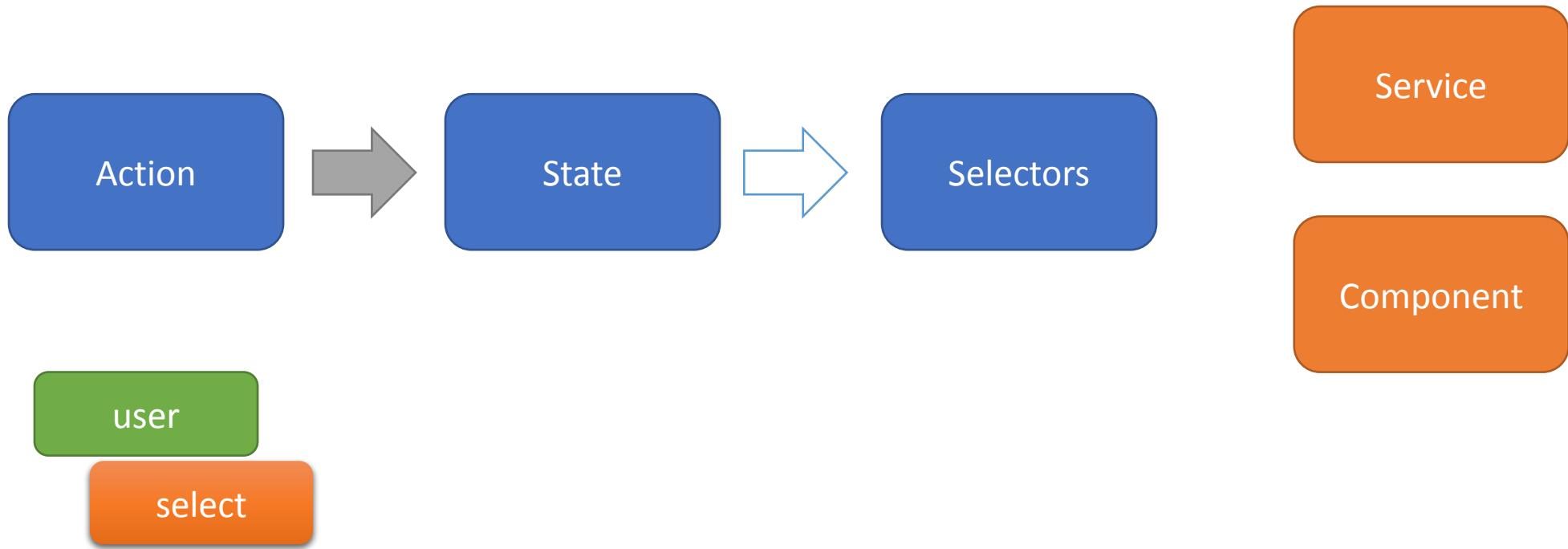
performance

side effects





user



Action

State

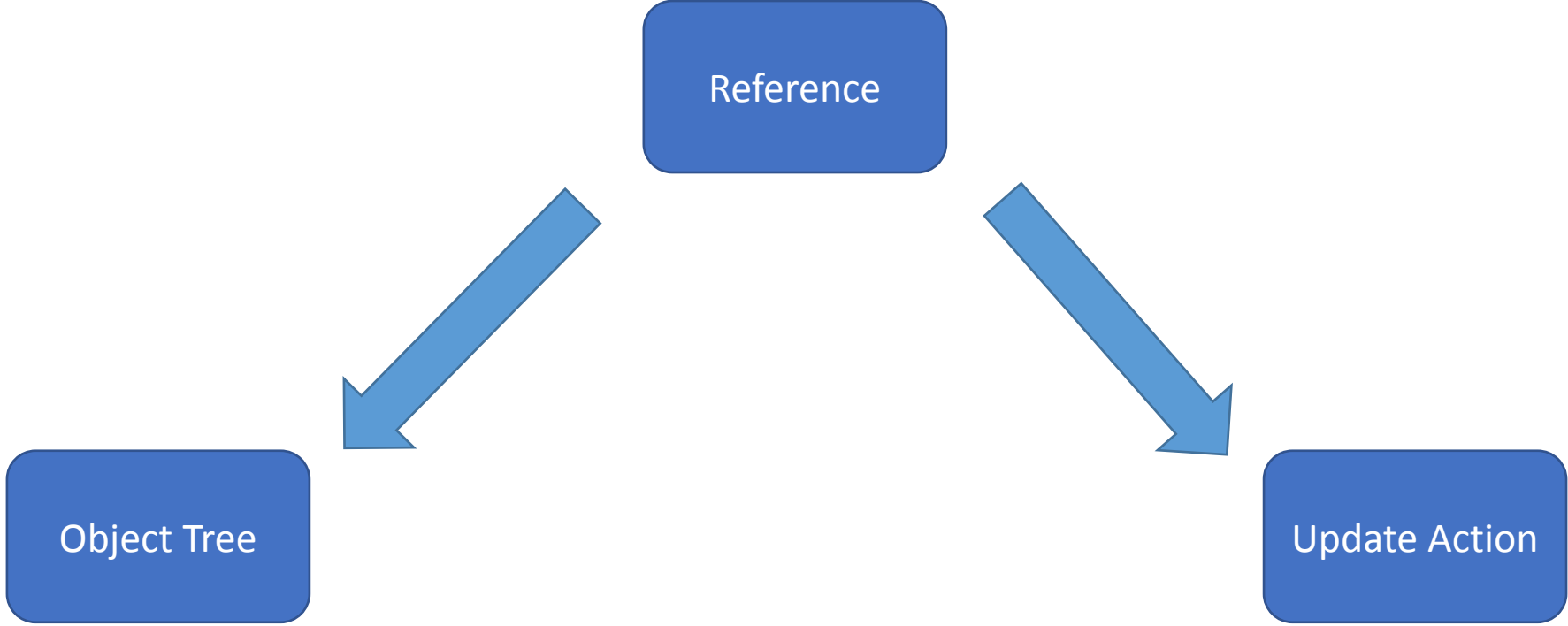
Selectors

Service

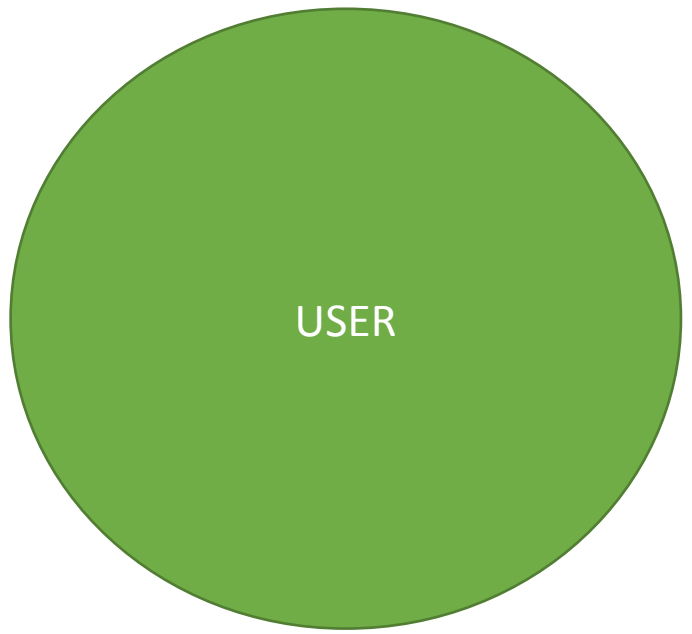
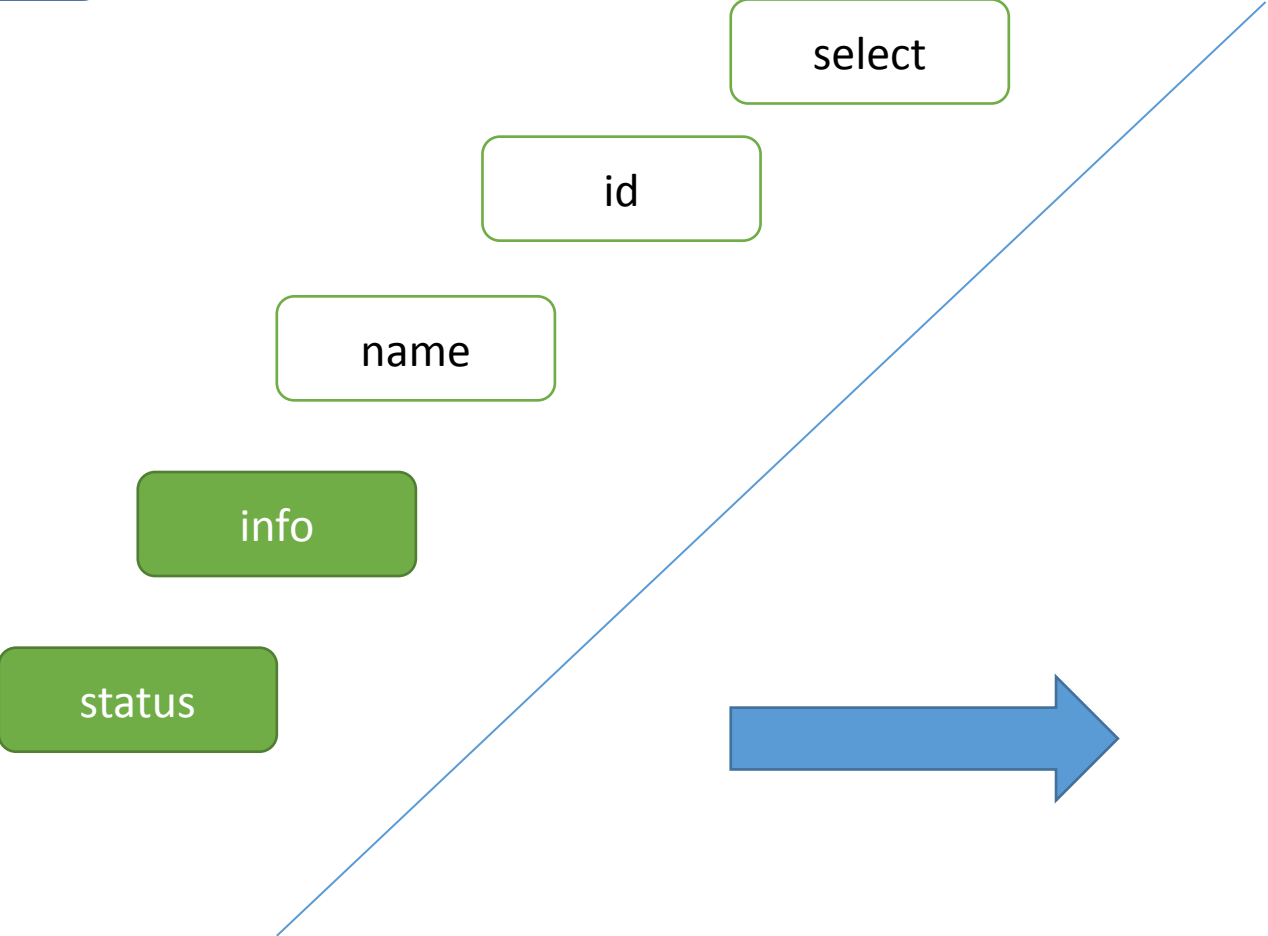
Component

user

select



Object Tree



Update Action

users

user[]

user

name



Auto
Immutable JS



Manual
new {..., old}



users

user[]

user

name