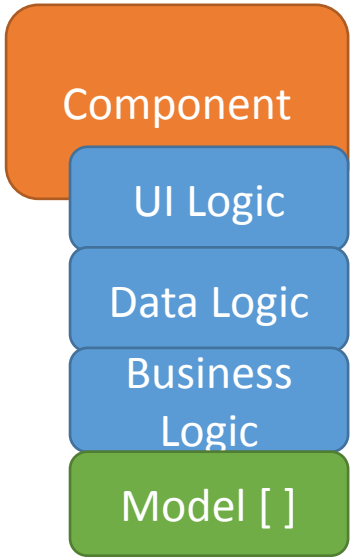
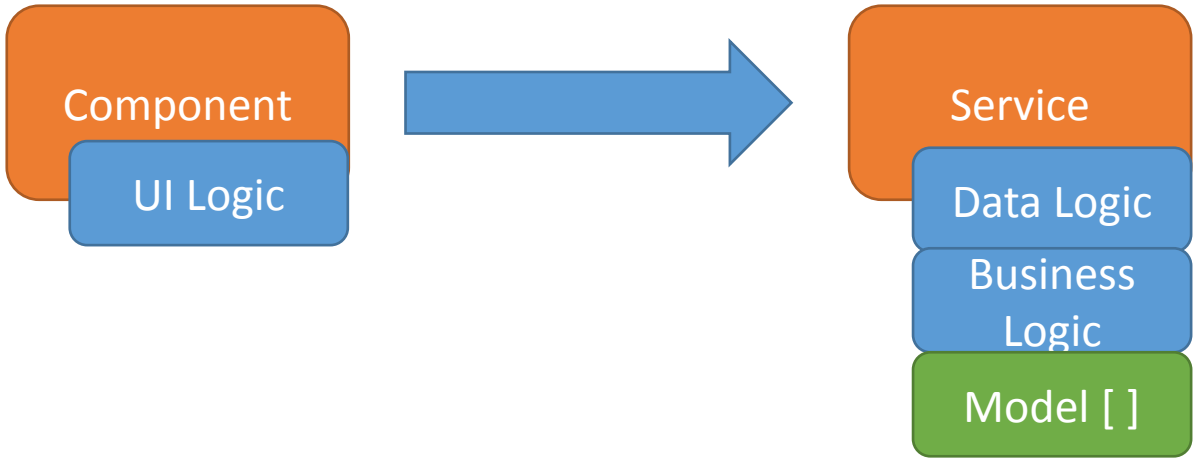
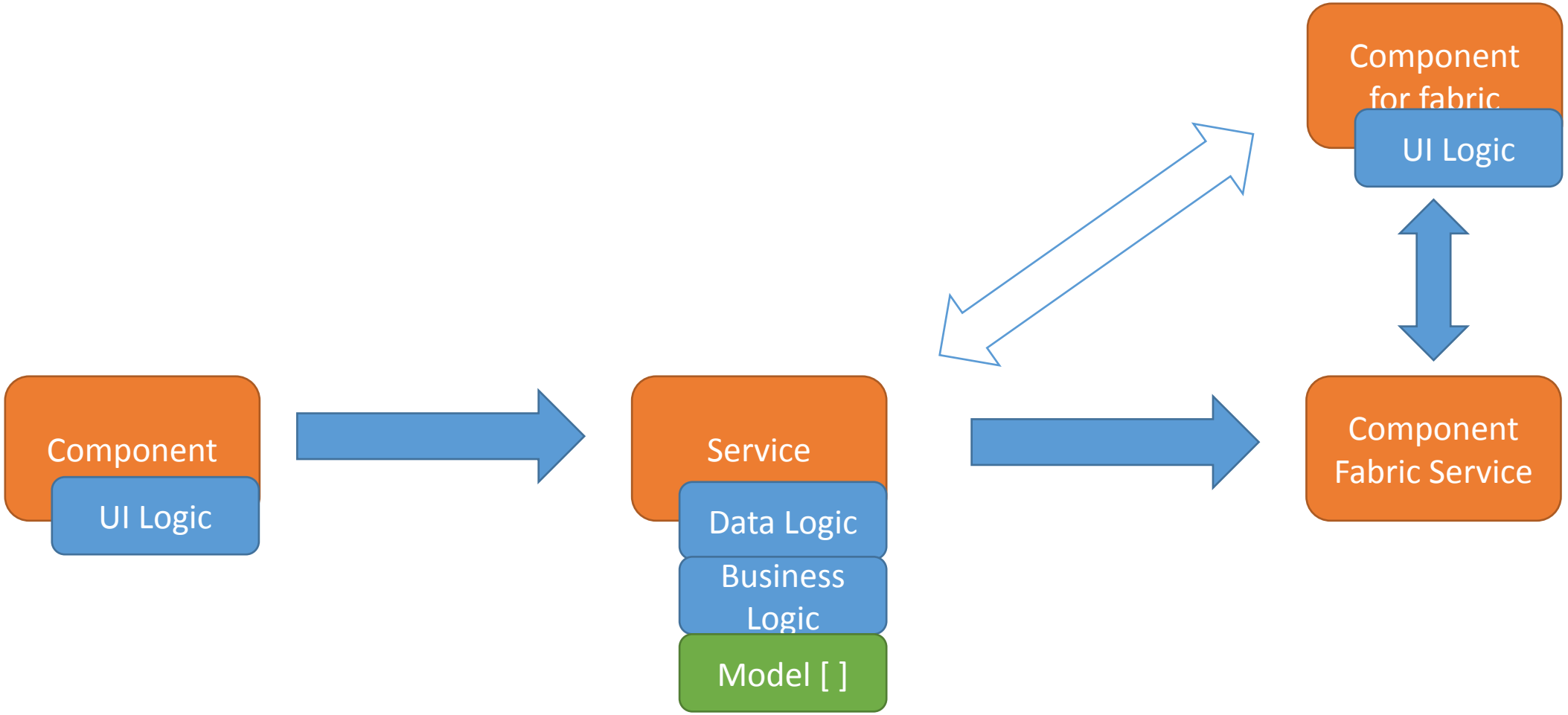
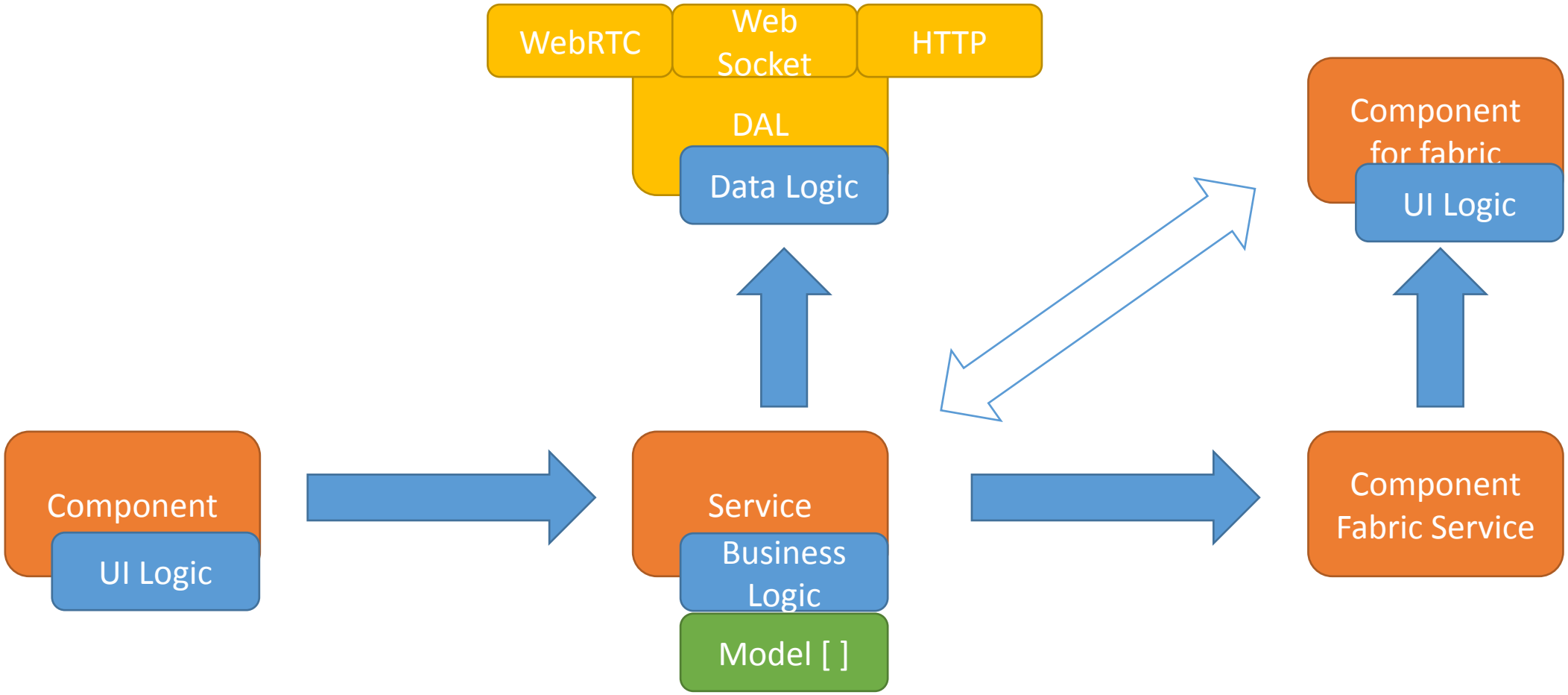
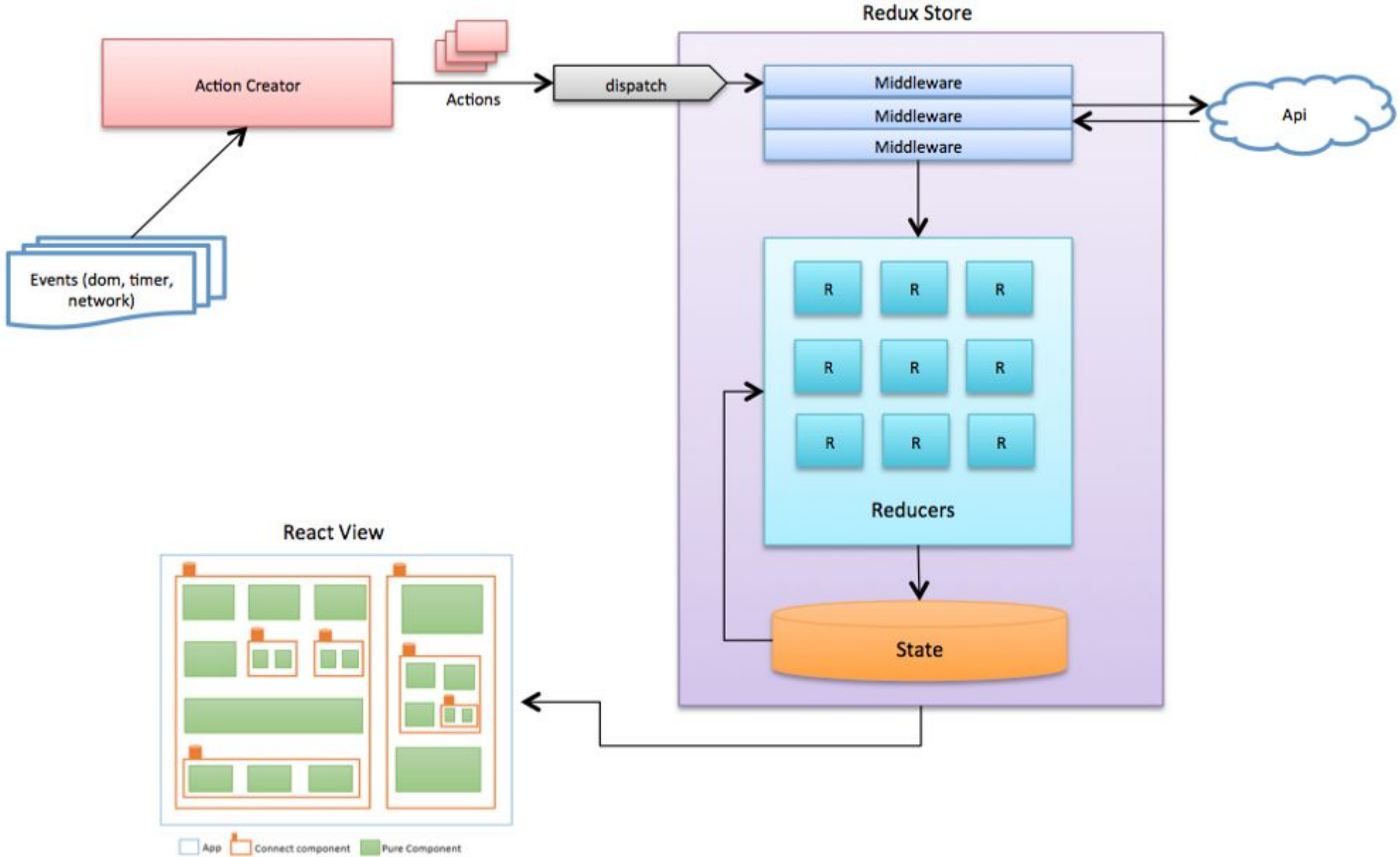
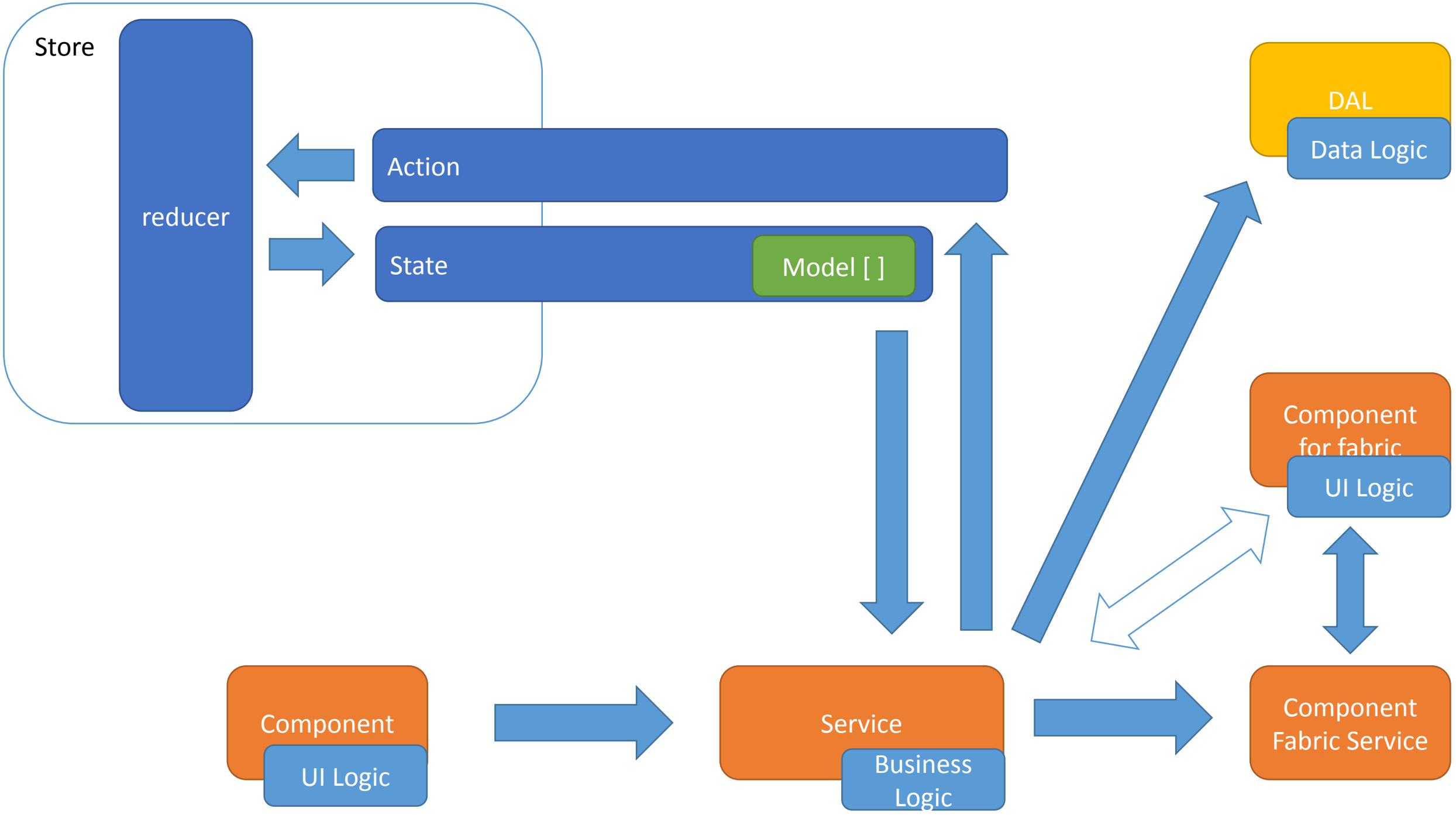

Component
UI Logic
Data Logic
Business Logic
Model [ ]


Component
UI Logic
Service
Data Logic
Business Logic
Model [ ]

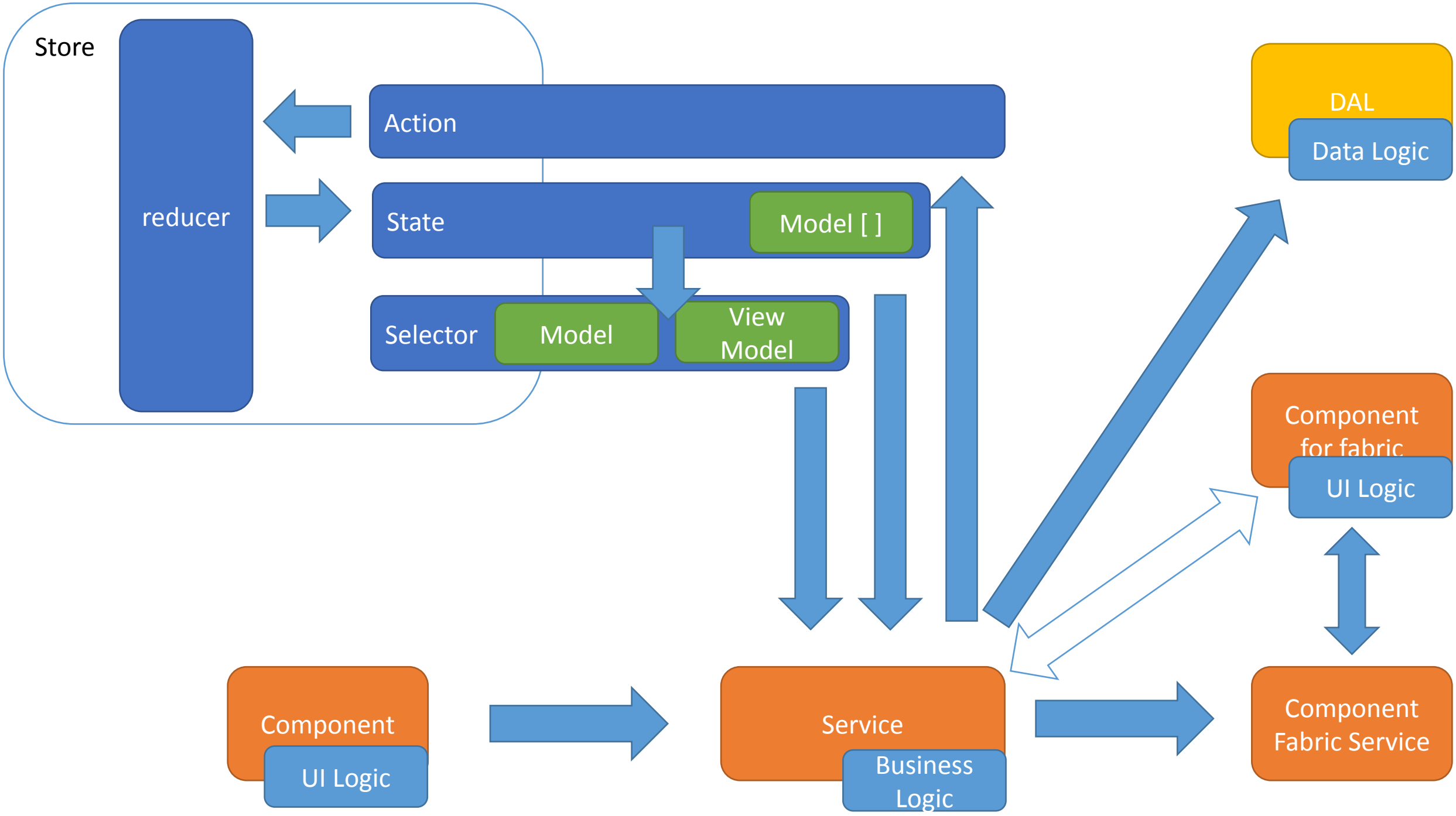

Component
UI Logic
Service
Data Logic
Business Logic
Model [ ]
Component for fabric
UI Logic
Component Fabric Service


WebRTC
Web Socket
HTTP
DAL
Data Logic
Component for fabric
UI Logic
Component
UI Logic
Service
Business Logic
Model [ ]
Component Fabric Service

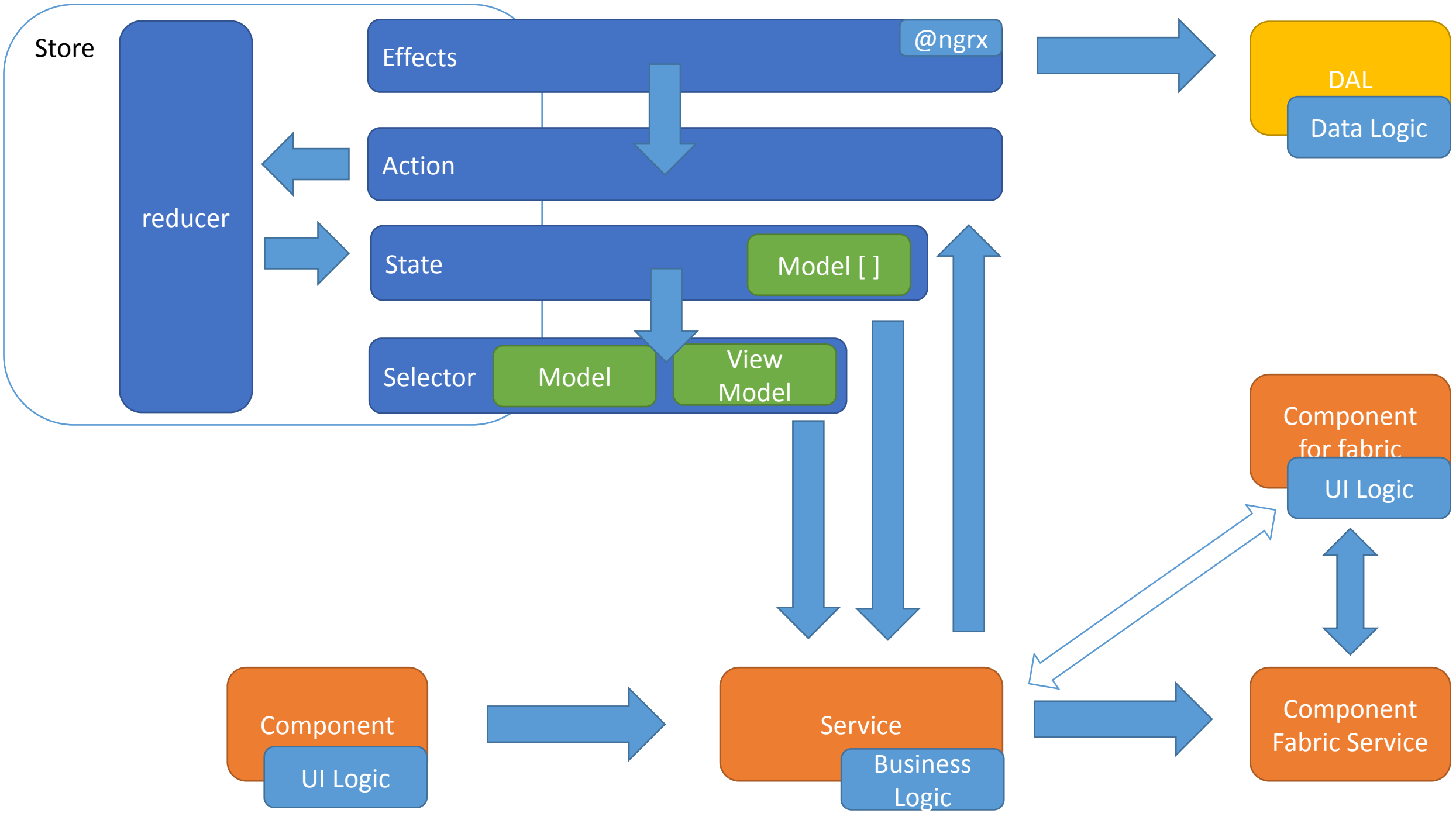

Redux Store
Action Creator
Actions
dispatch
Middleware
Middleware
Middleware
Api
Events (dom, timer, network)
R
R
R
R
R
R
R
R
R
Reducers
State
React View
App
Connect component
Pure Component


Store
reducer
Action
State
Model [ ]
DAL
Data Logic
Component
for fabric
UI Logic
Component
UI Logic
Service
Business
Logic
Component
Fabric Service

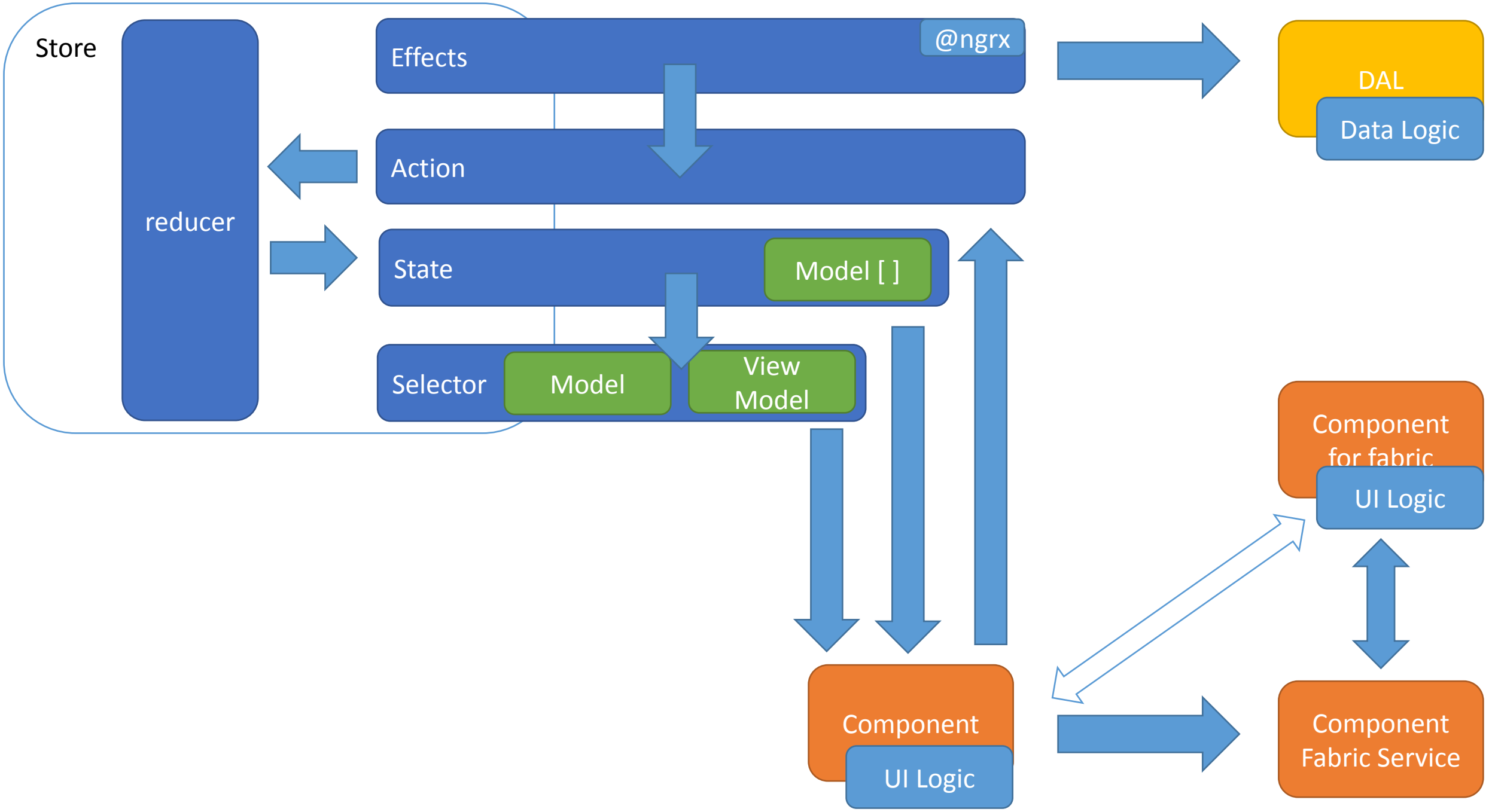
```ts
import * as layout from '../actions/layout';

export interface State {
  showSidenav: boolean;
}

const initialState: State = {
  showSidenav: false,
};

export function reducer(state = initialState, action: layout.Actions): State {
  switch (action.type) {
    case layout.CLOSE_SIDENAV:
      return {
        showSidenav: false,
      };

    case layout.OPEN_SIDENAV:
      return {
        showSidenav: true,
      };

    default:
      return state;
  }
}
```

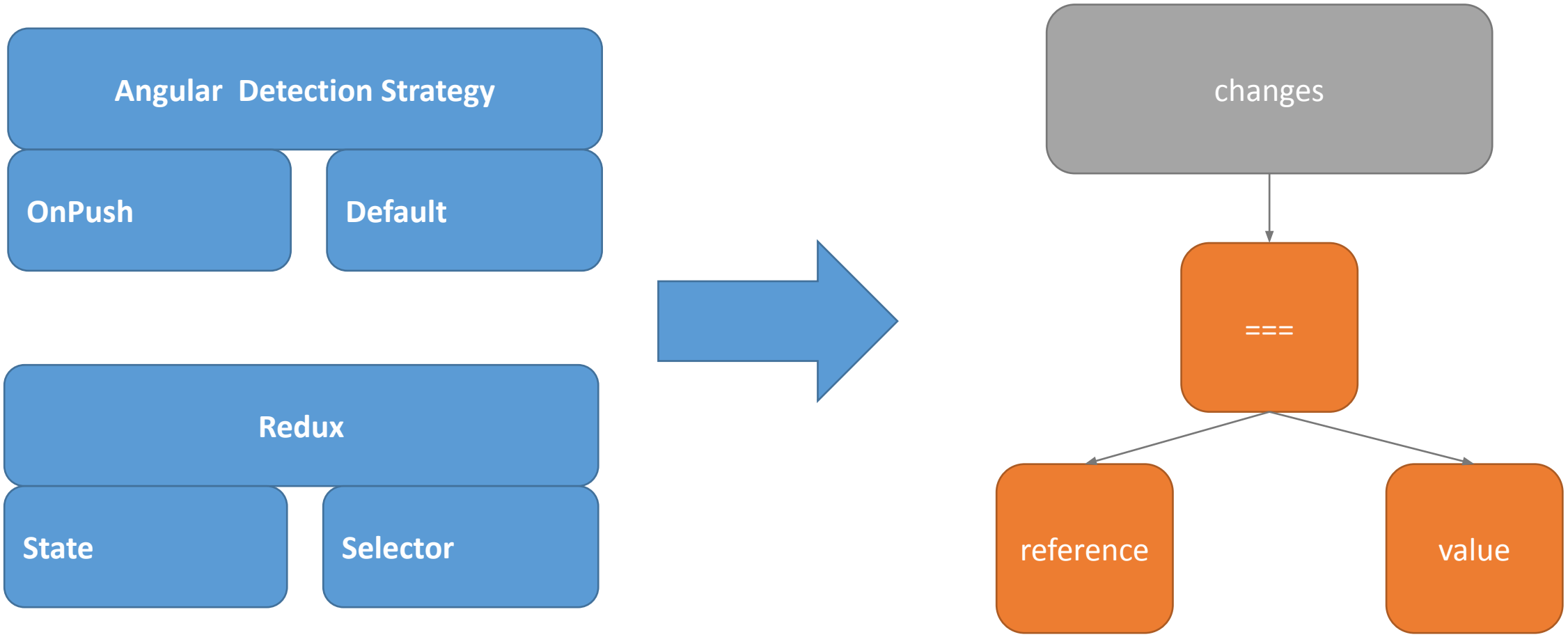

Store
reducer
Action
State
Model [ ]
Selector
Model
View Model
DAL
Data Logic
Component for fabric
UI Logic
Component
UI Logic
Service
Business Logic
Component Fabric Service

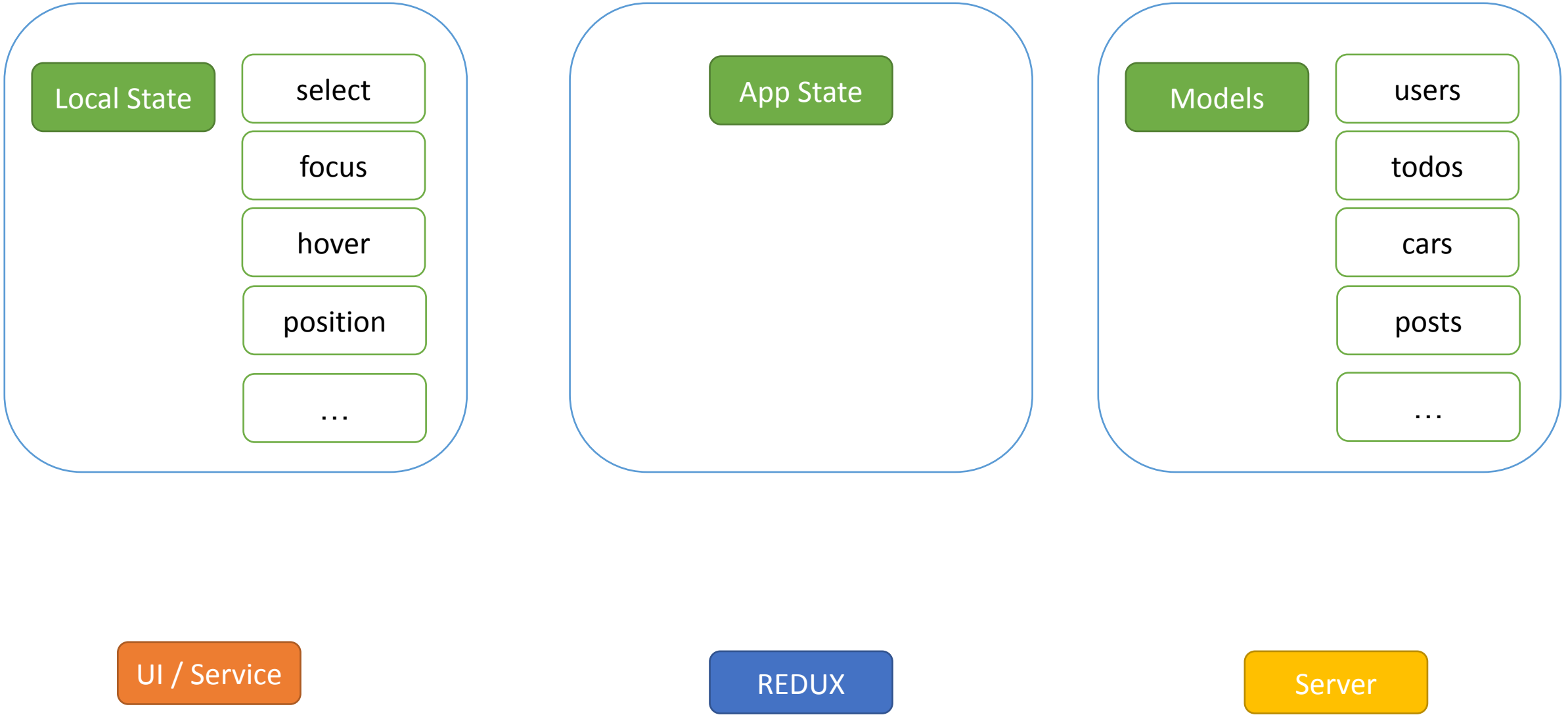
```
import { createSelector } from 'reselect'

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)
    }
  }
)
```

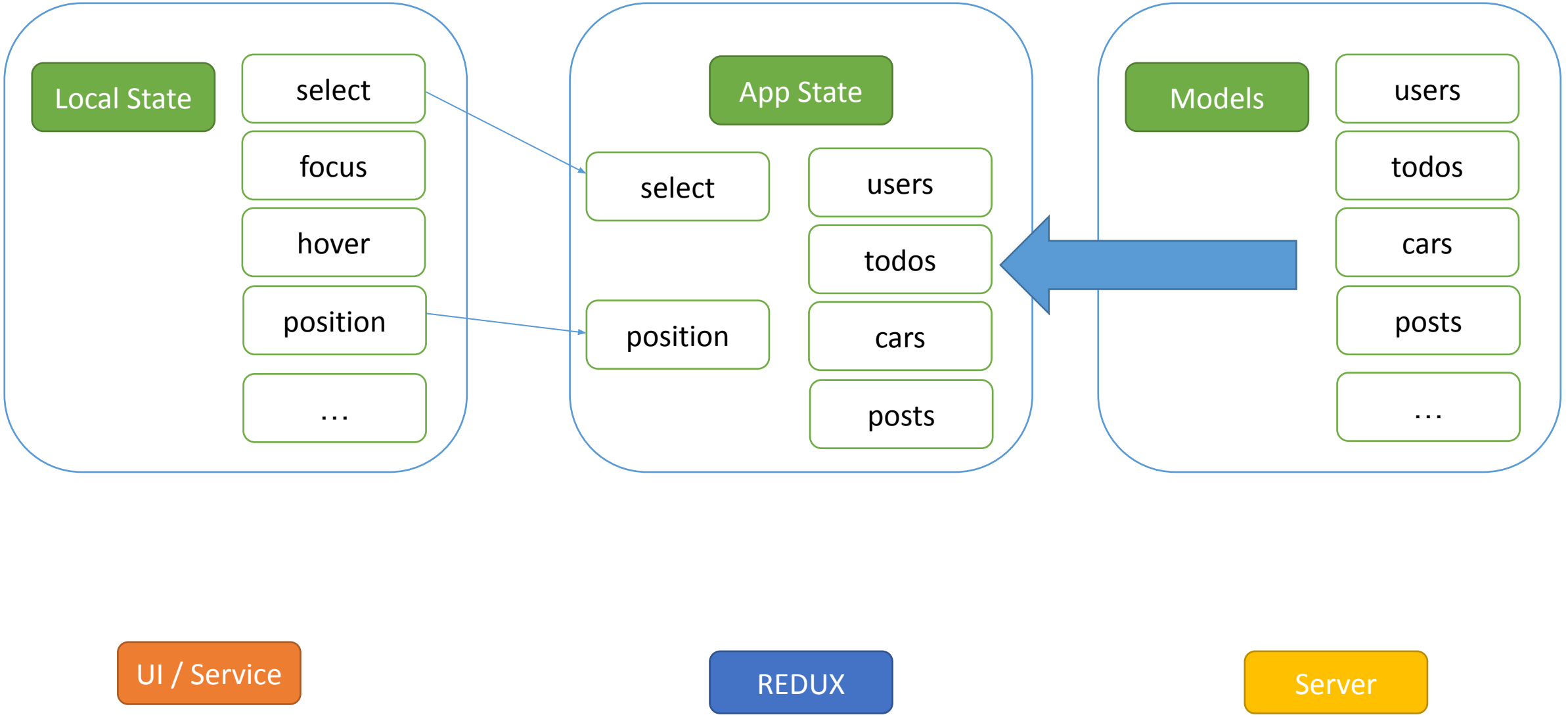

Store
reducer
Effects
@ngrx
Action
State
Model [ ]
Selector
Model
View Model
DAL
Data Logic
Component for fabric
UI Logic
Component
UI Logic
Service
Business Logic
Component Fabric Service

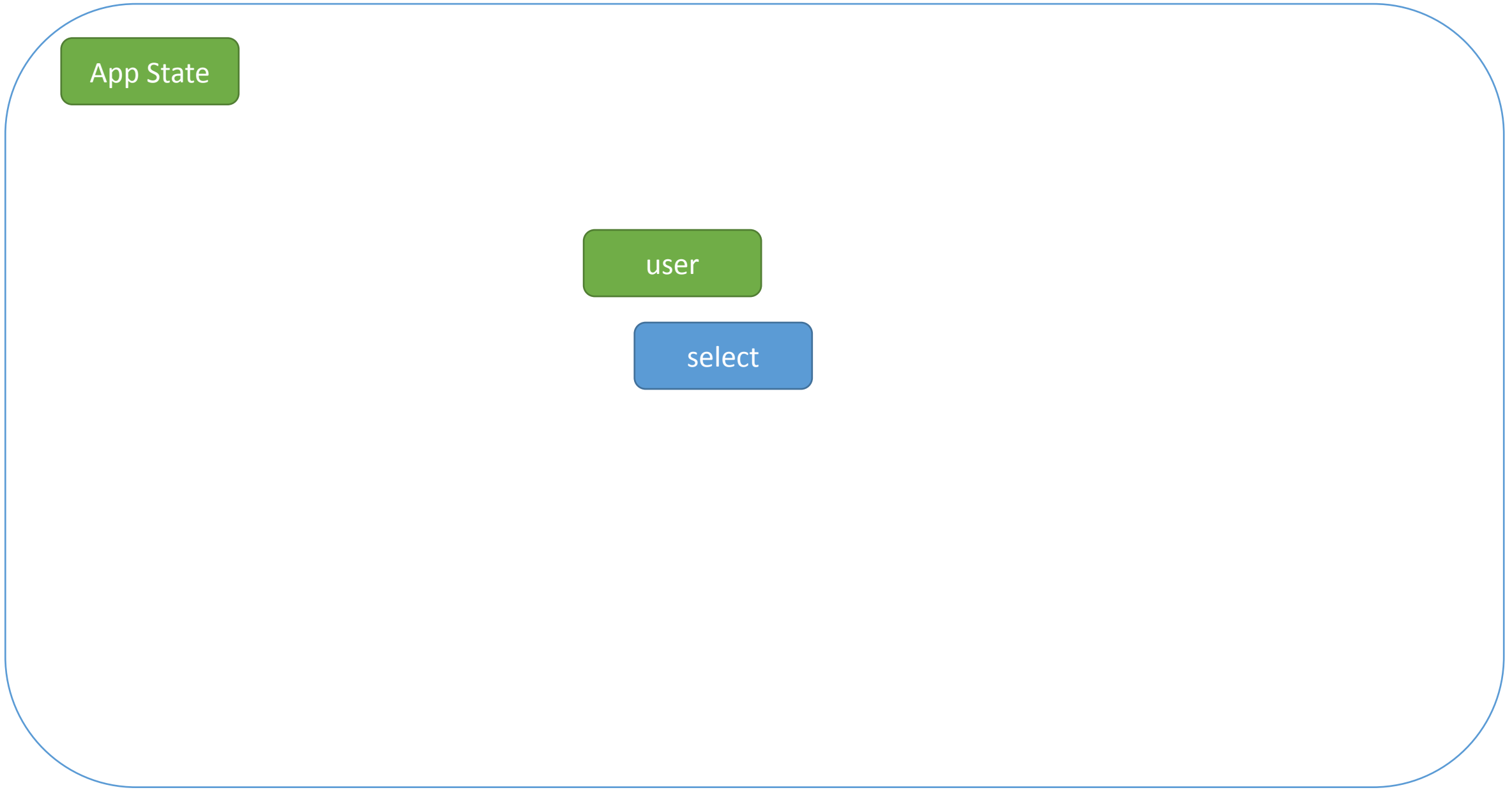
```
//category.effects.ts

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))
}
```

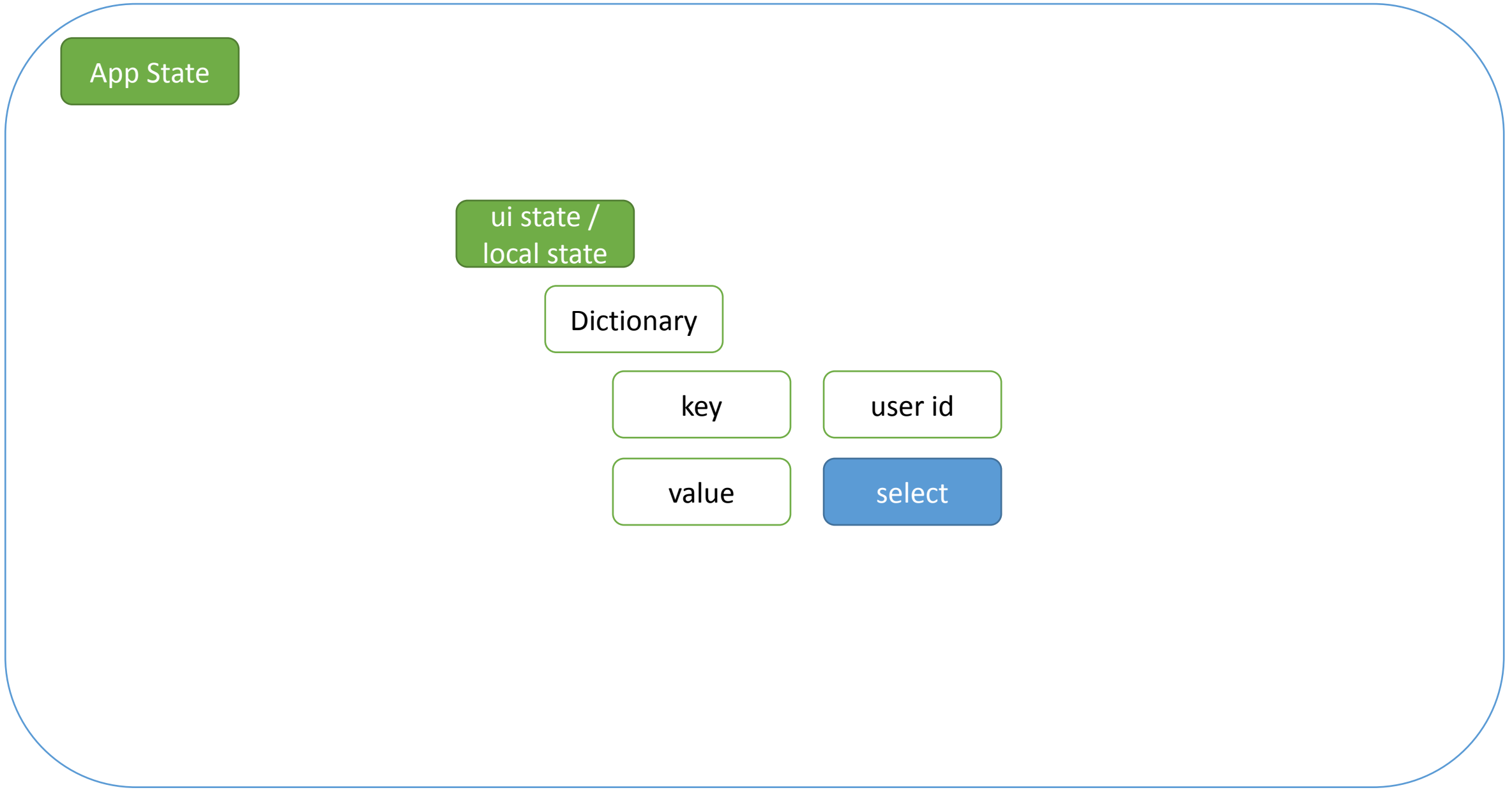

Store
reducer
Effects
@ngrx
Action
State
Model [ ]
Selector
Model
View Model
DAL
Data Logic
Component for fabric
UI Logic
Component
UI Logic
Component Fabric Service

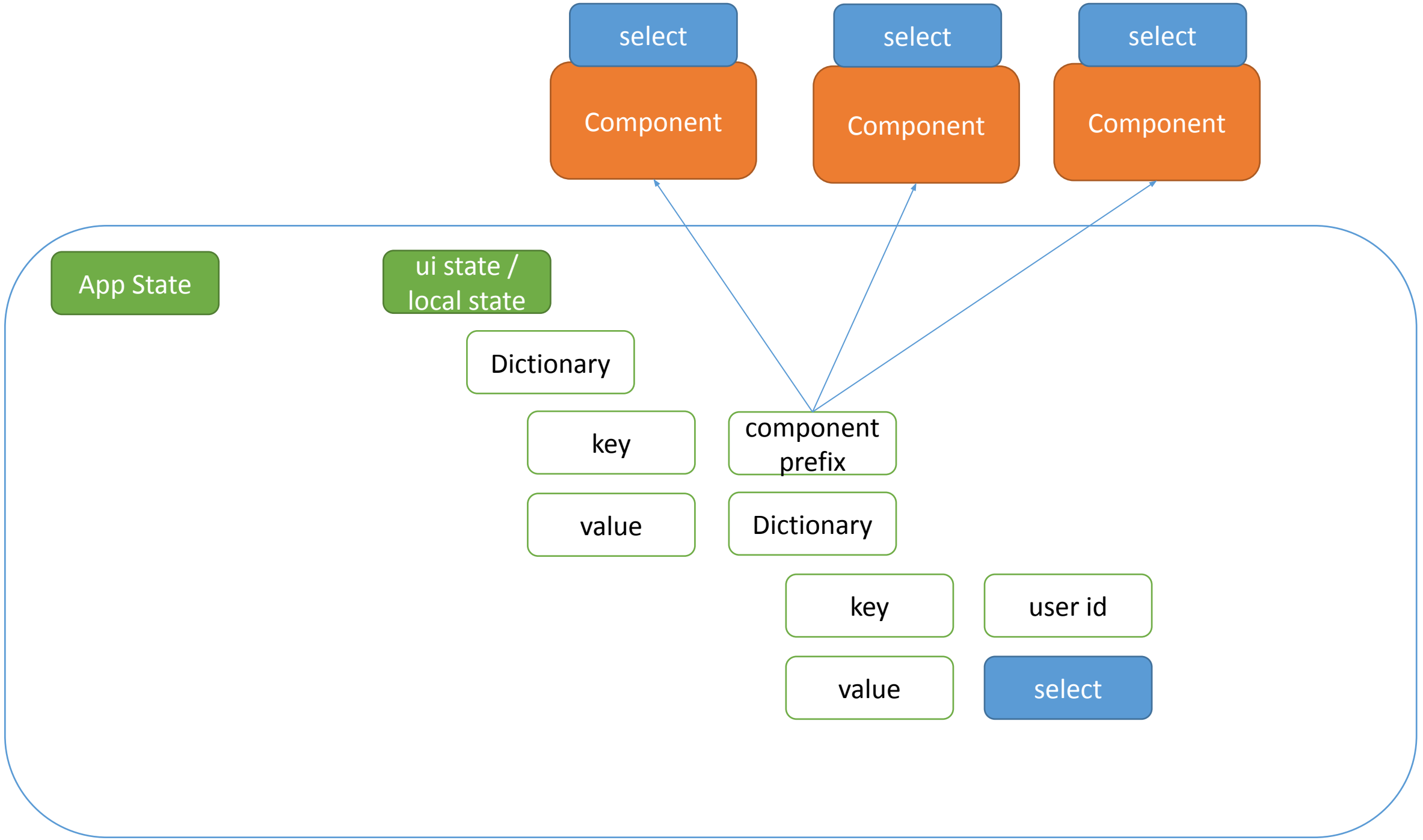

Angular Detection Strategy
OnPush
Default
Redux
State
Selector
changes
===
reference
value

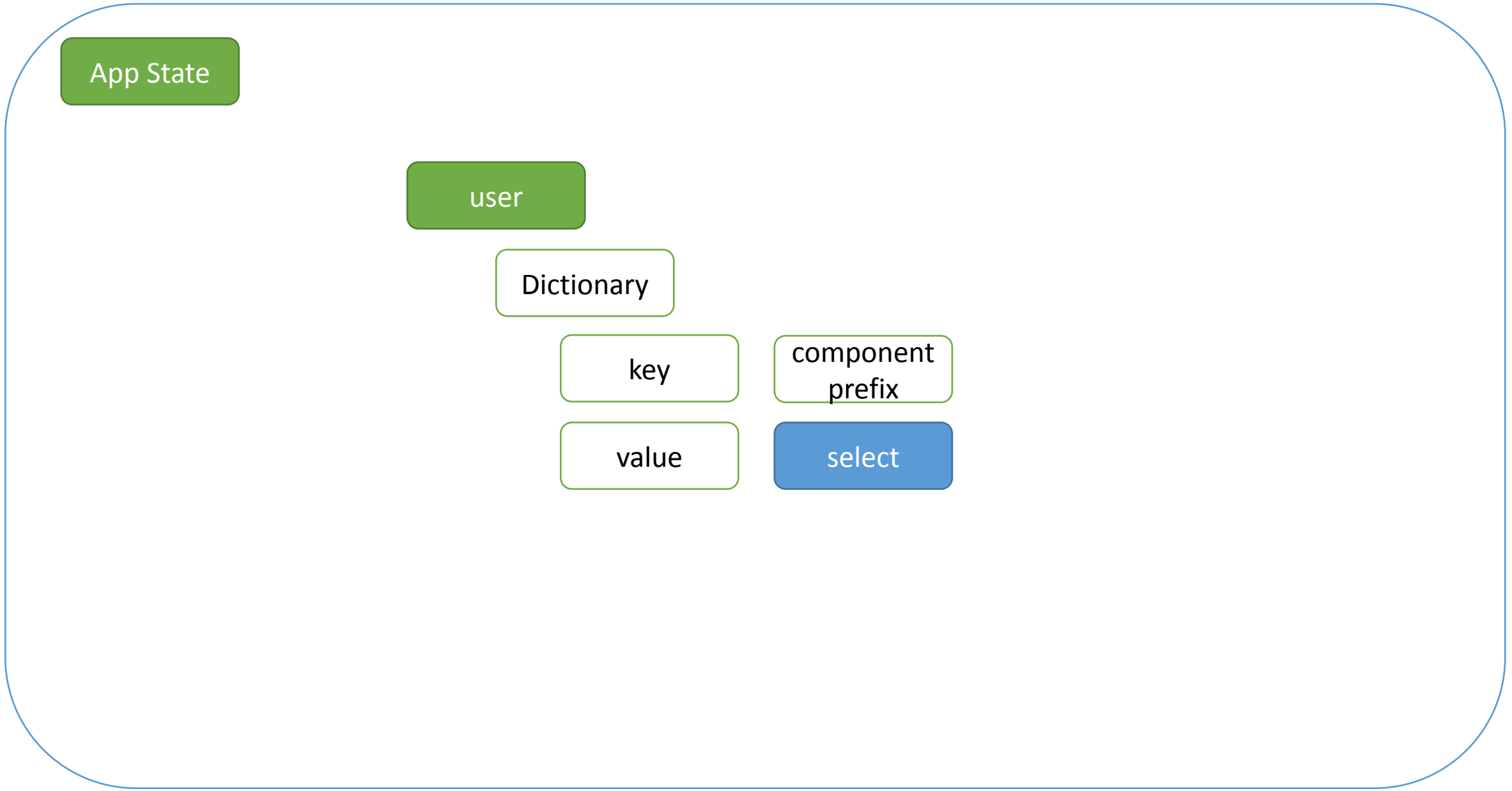

Local State
select
focus
hover
position
…
App State
Models
users
todos
cars
posts
…
UI / Service
REDUX
Server

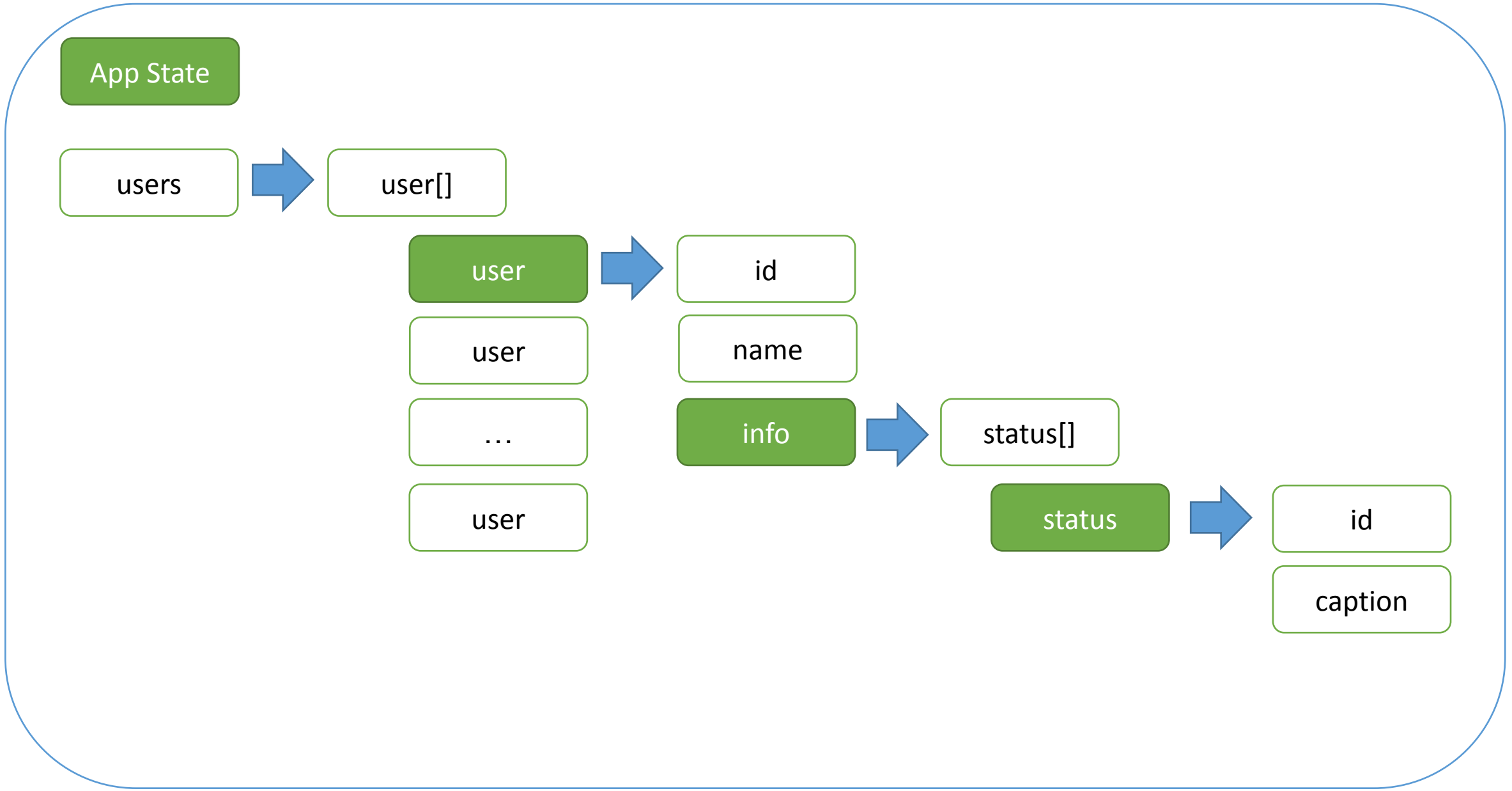

Local State
select
focus
hover
position
…
App State
select
position
users
todos
cars
posts
Models
users
todos
cars
posts
…
UI / Service
REDUX
Server

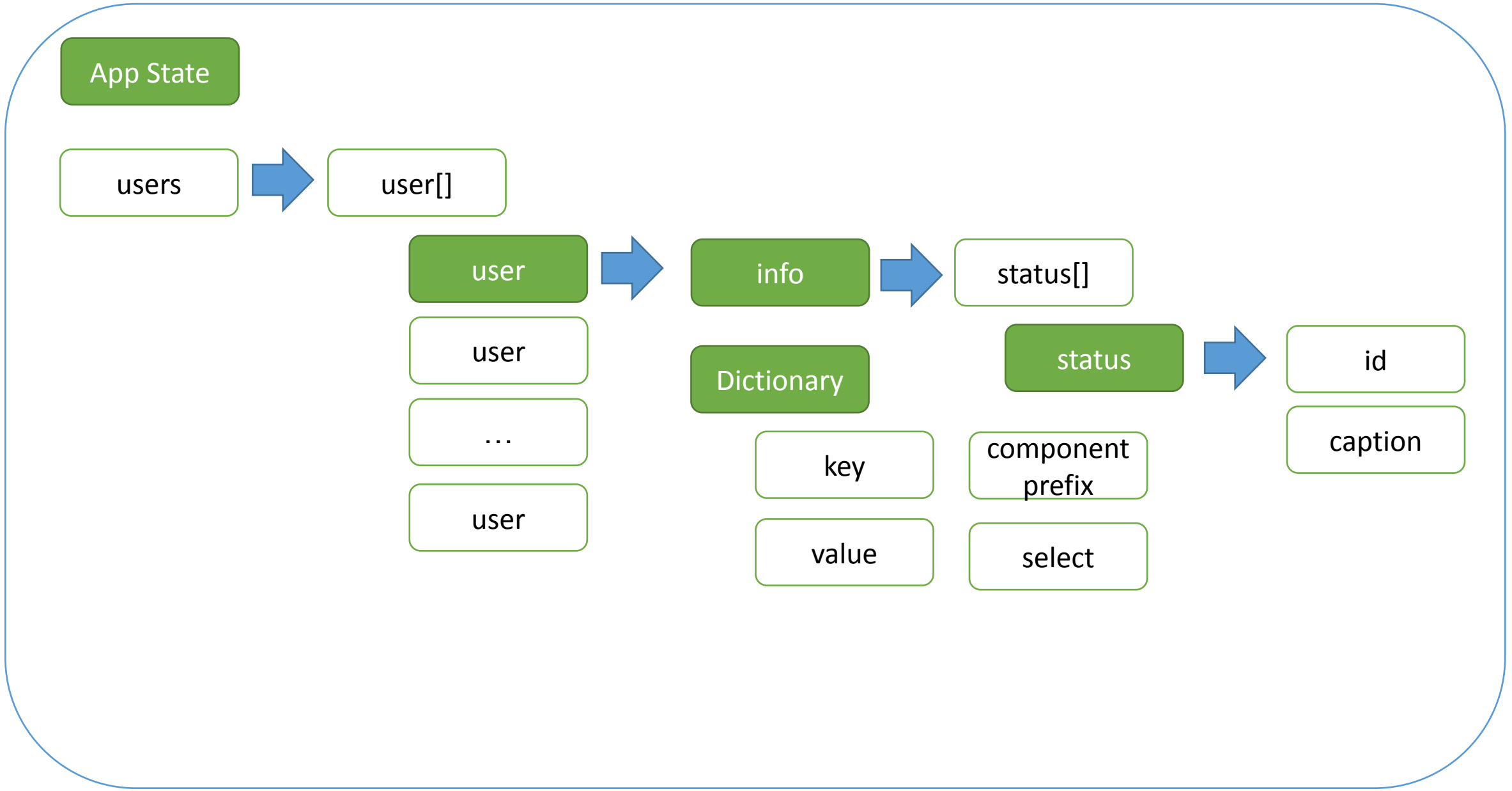

App State
user
select

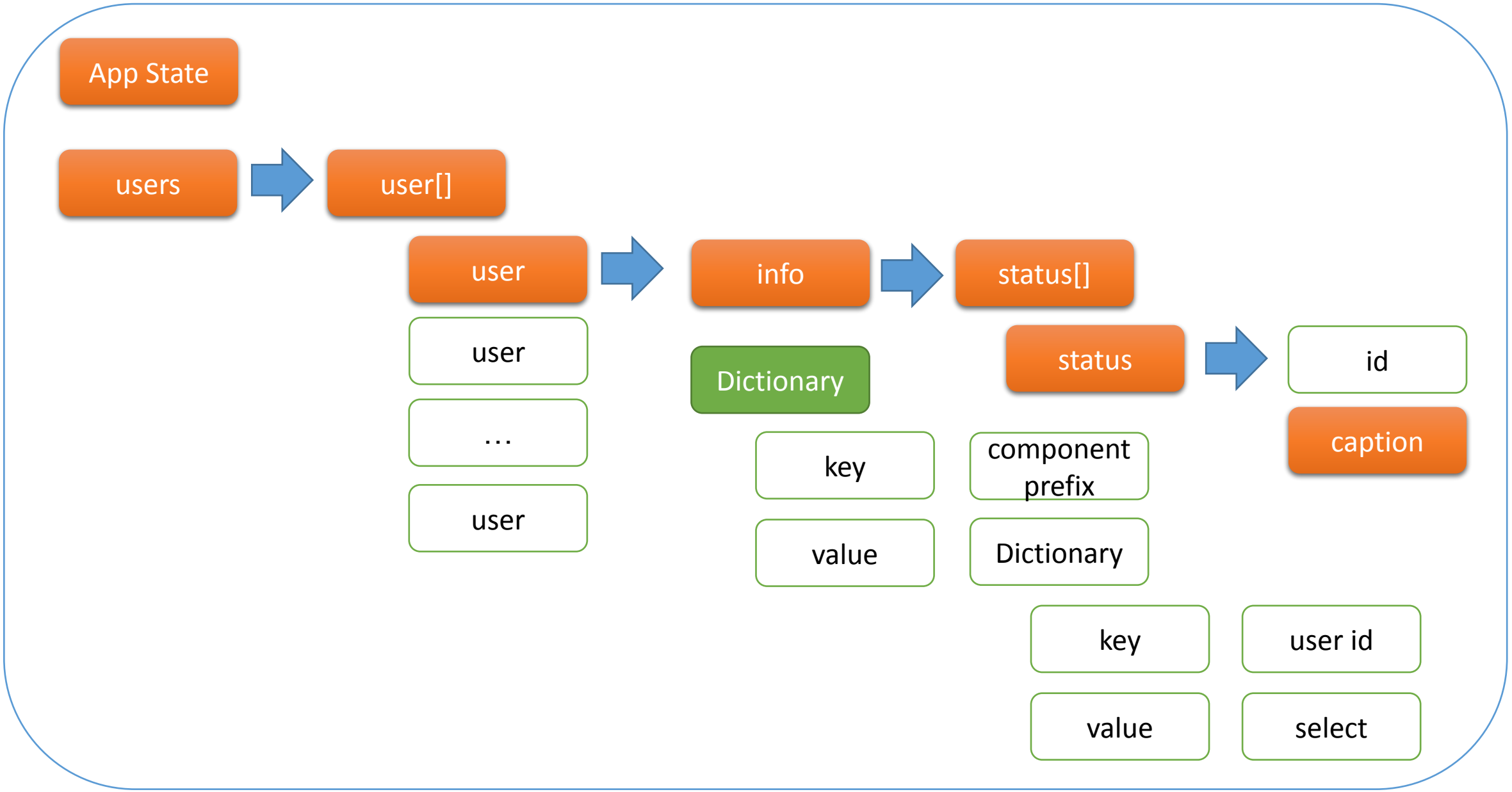

App State
ui state / local state
Dictionary
key
user id
value
select

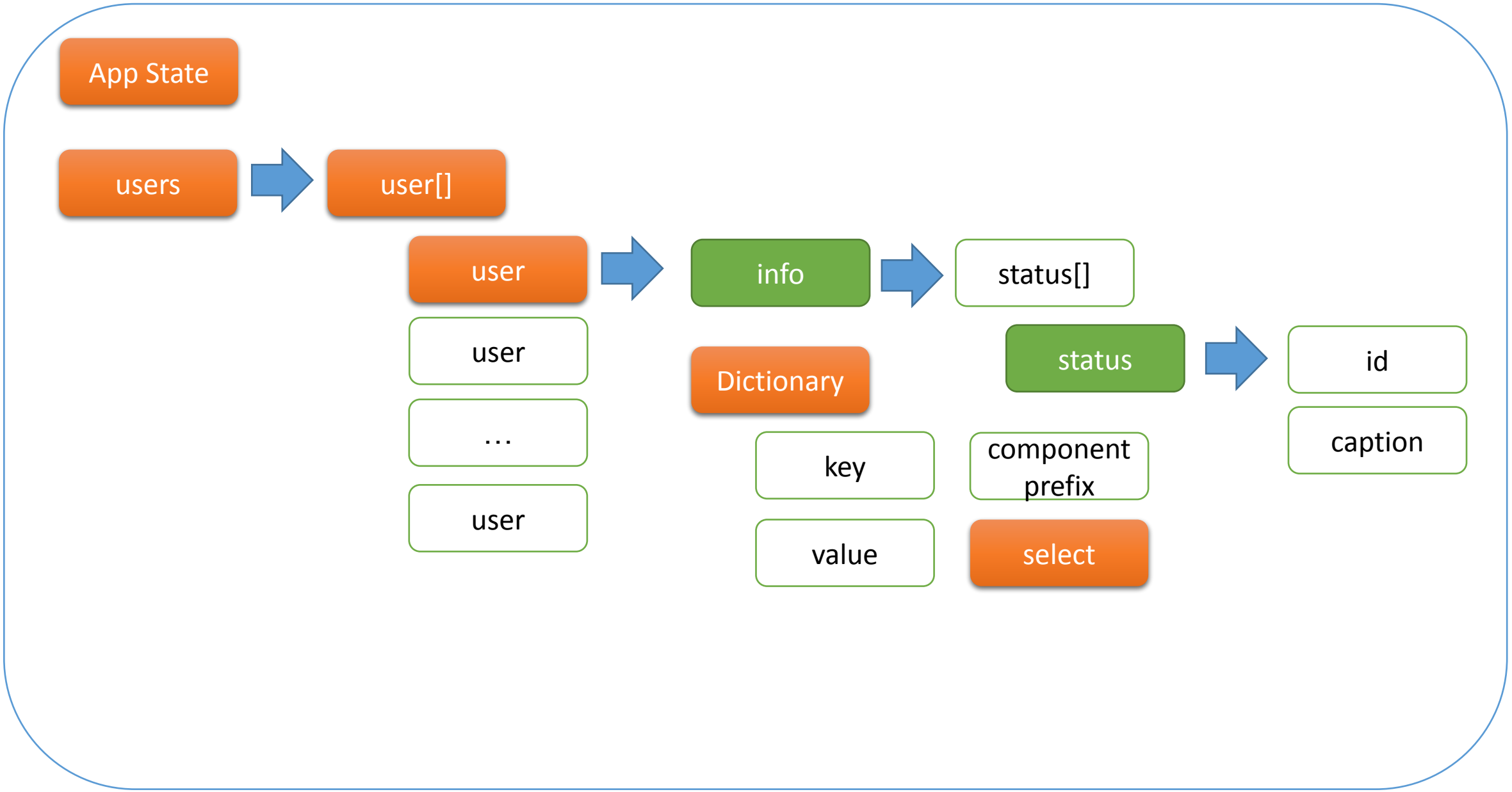

select
Component
select
Component
select
Component
App State
ui state / local state
Dictionary
key
component prefix
value
Dictionary
key
user id
value
select

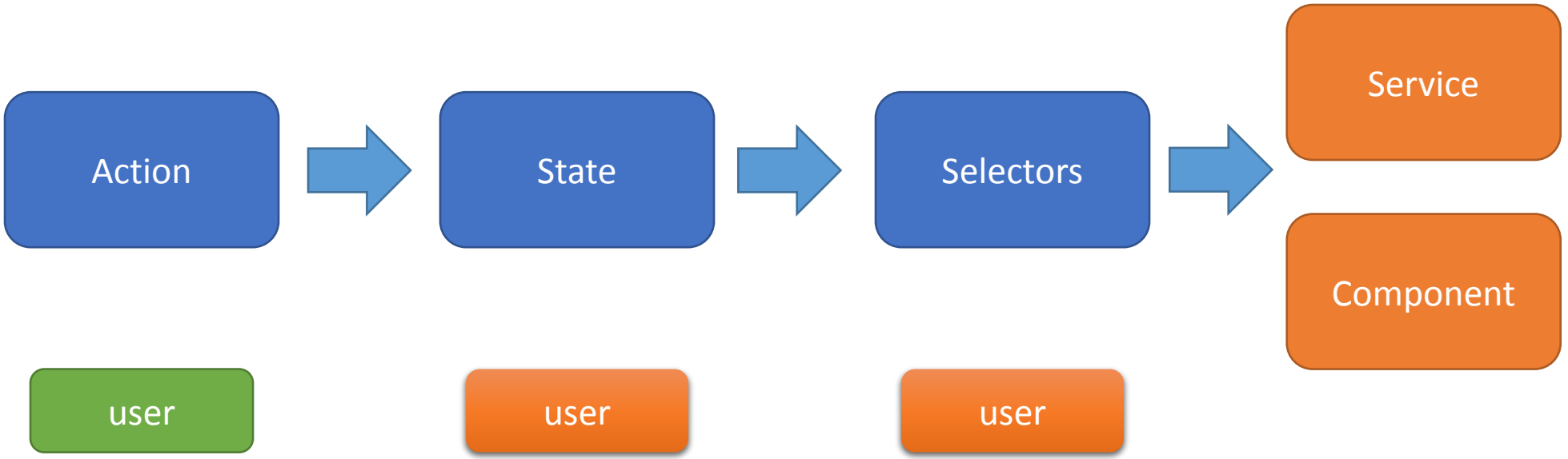

App State
user
Dictionary
key
component prefix
value
select

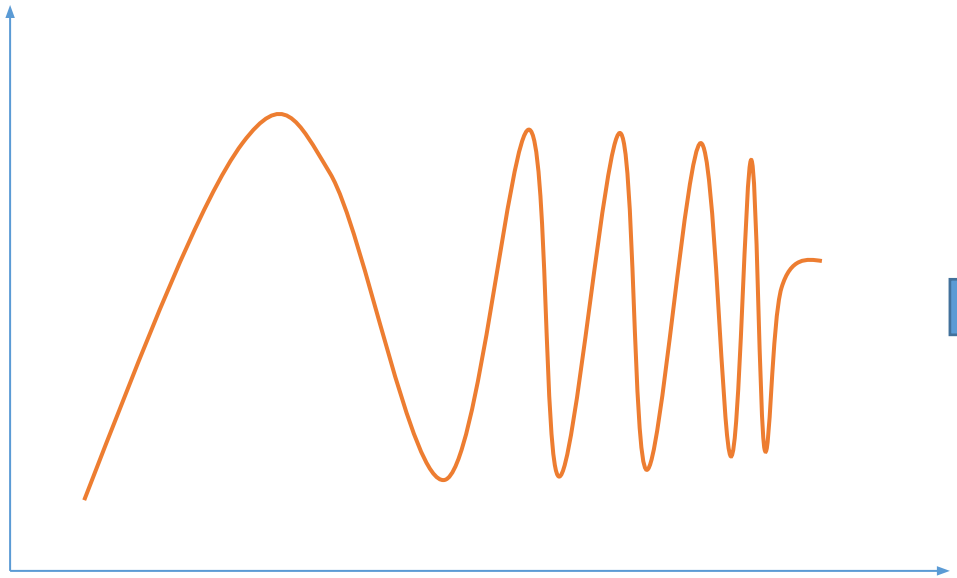

App State
users
user[]
user
user
…
user
id
name
info
status[]
status
id
caption


App State
users
user[]
user
user
…
user
info
Dictionary
key
value
status[]
status
component prefix
select
id
caption


App State
users
user[]
user
user
…
user
info
Dictionary
key
value
status[]
status
component prefix
Dictionary
key
value
id
caption
user id
select


App State
users
user[]
user
user
…
user
info
status[]
Dictionary
key
value
component prefix
select
status
id
caption


Action
State
Selectors
Service
Component
user
user
user



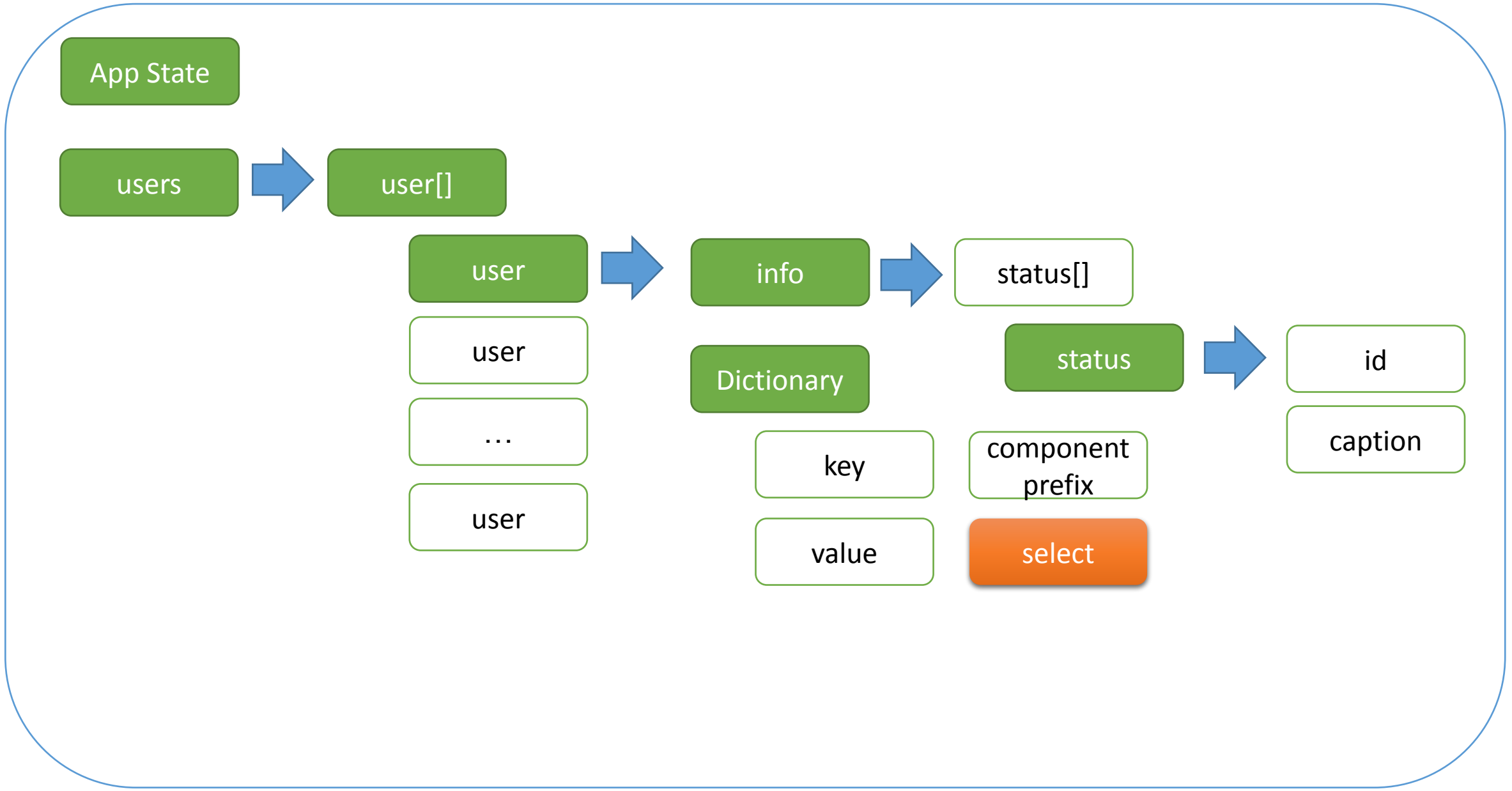
user

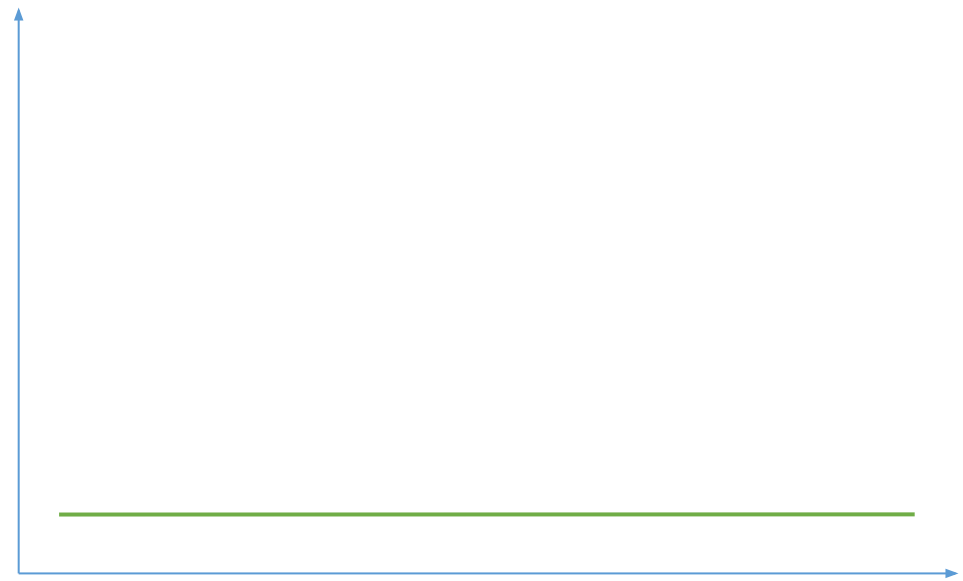
rendering

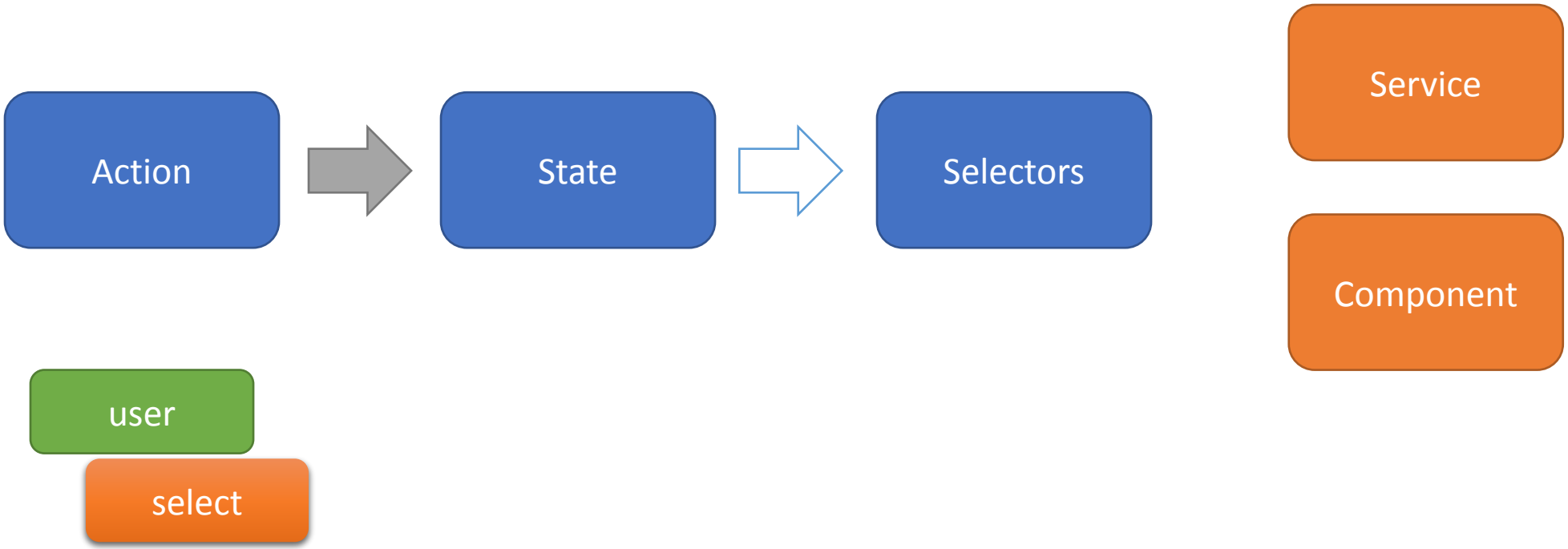
calculation

performance

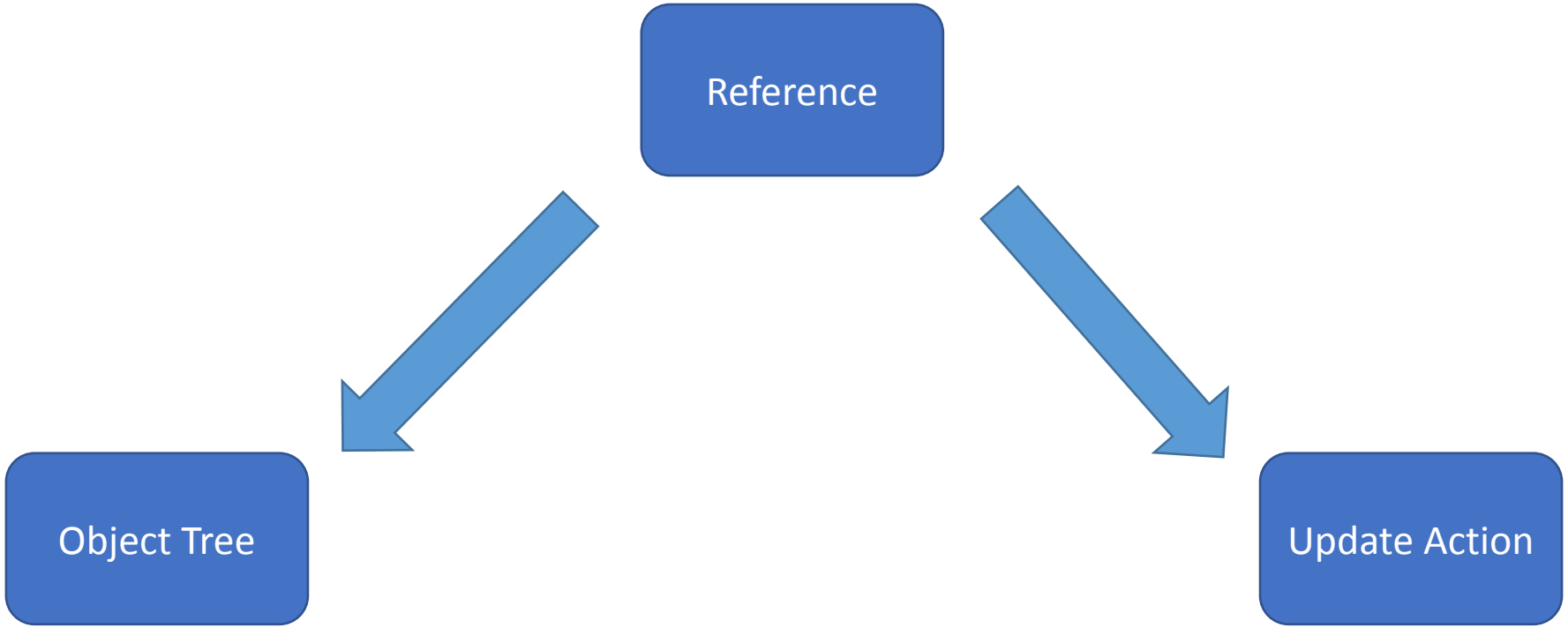
side effects

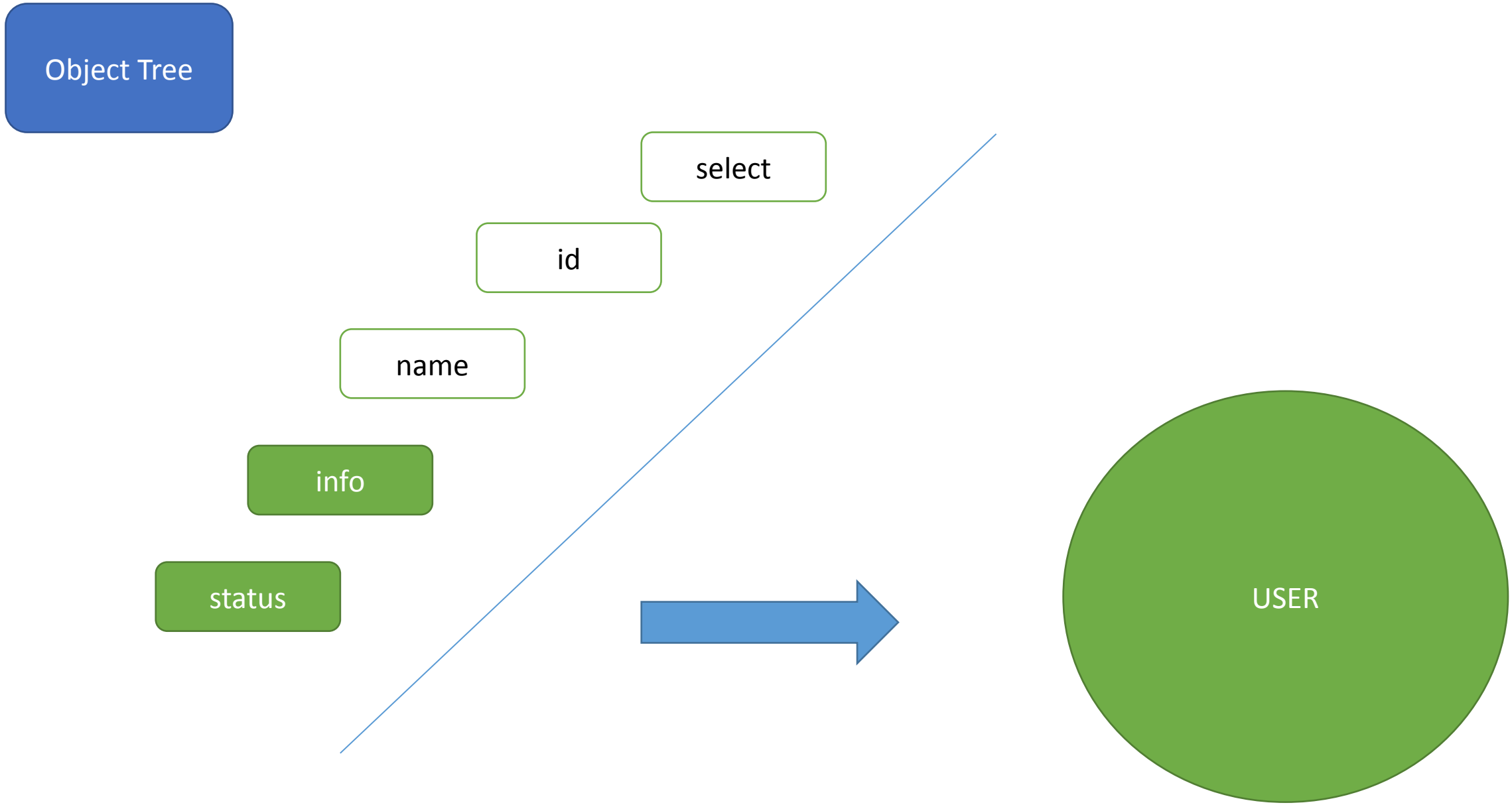

App State
users
user[]
user
user
…
user
info
Dictionary
key
value
status[]
status
component prefix
select
id
caption

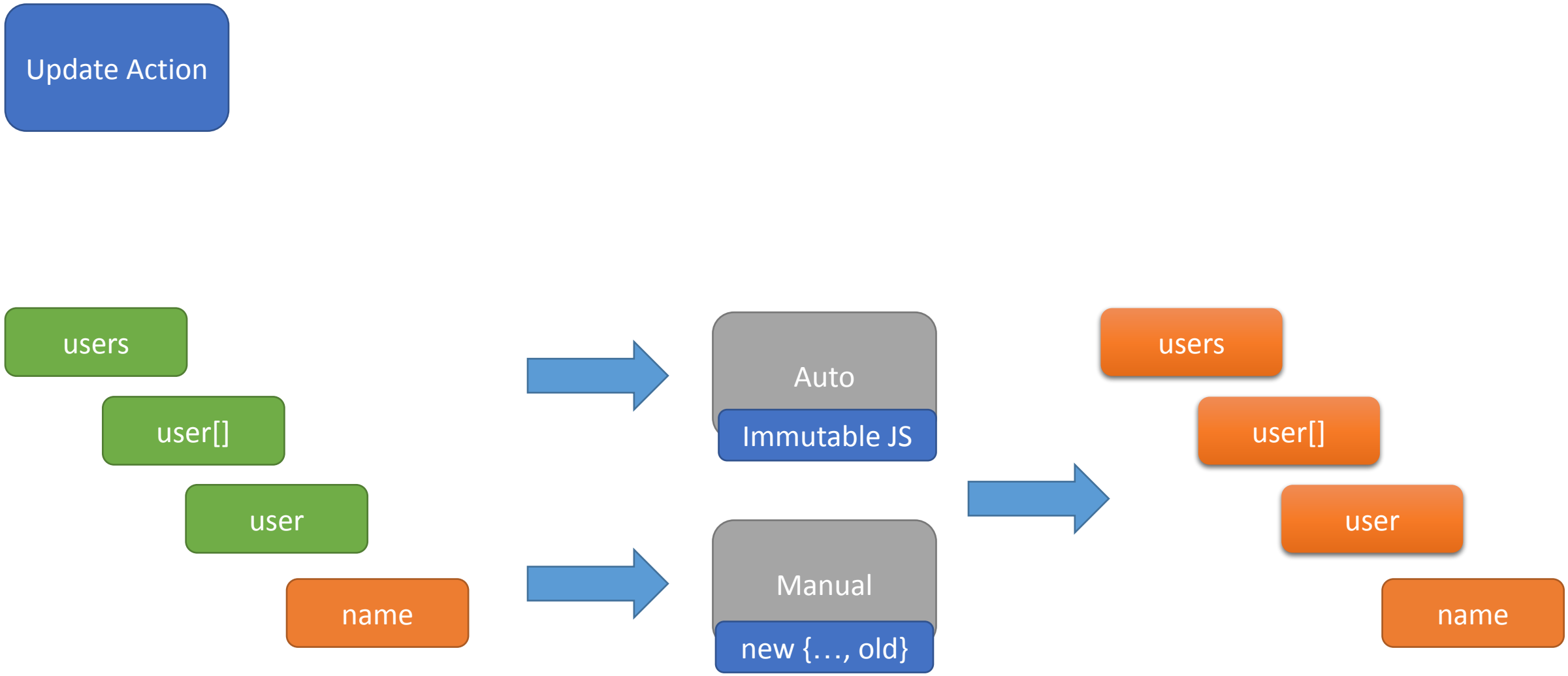

user


Action
State
Selectors
Service
Component
user
select


Action
State
Selectors
Service
Component
user
select


Reference
Object Tree
Update Action


Object Tree
select
id
name
info
status
USER


Update Action
users
user[]
user
name
Auto
Immutable JS
Manual
new {…, old}
users
user[]
user
name