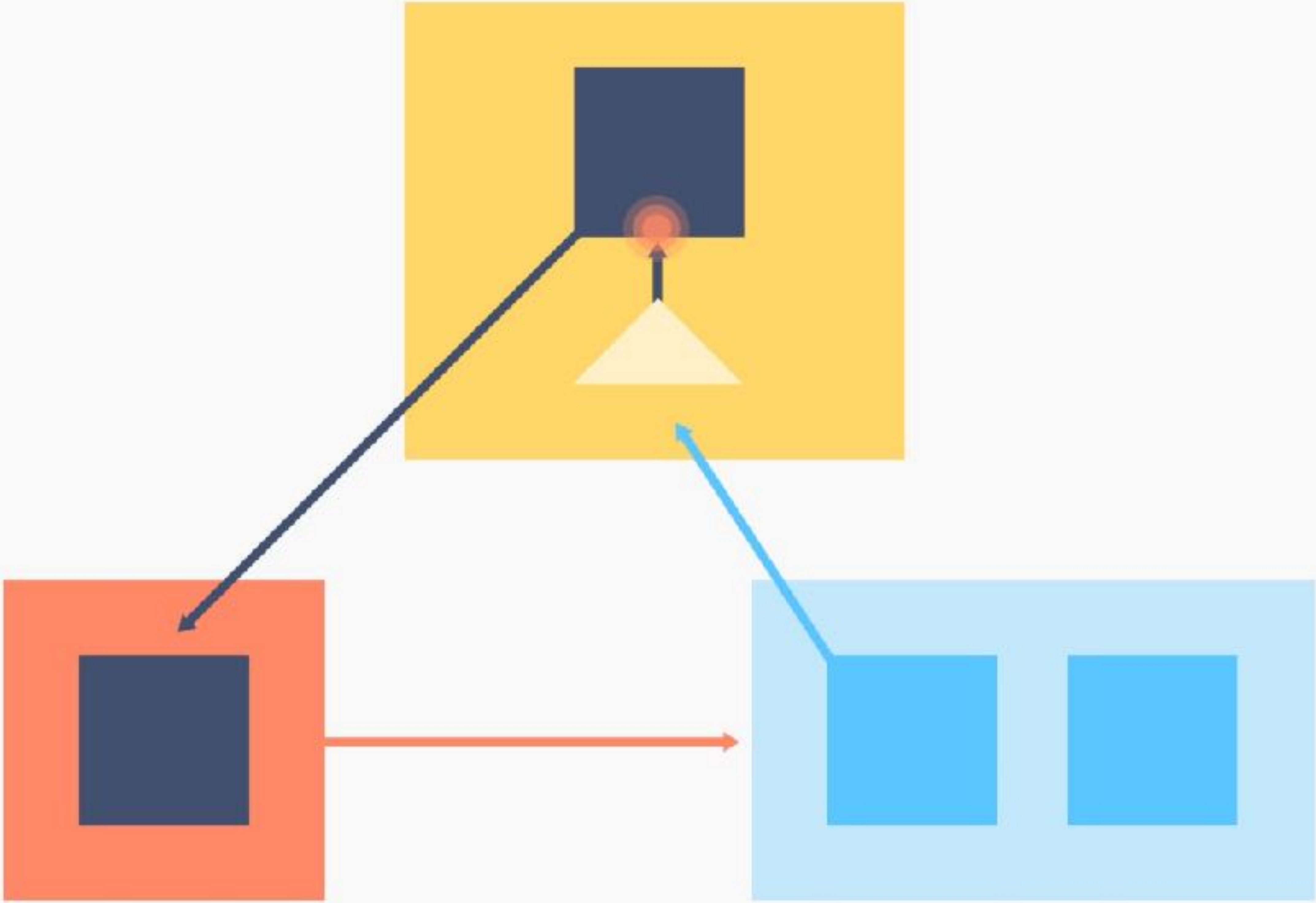
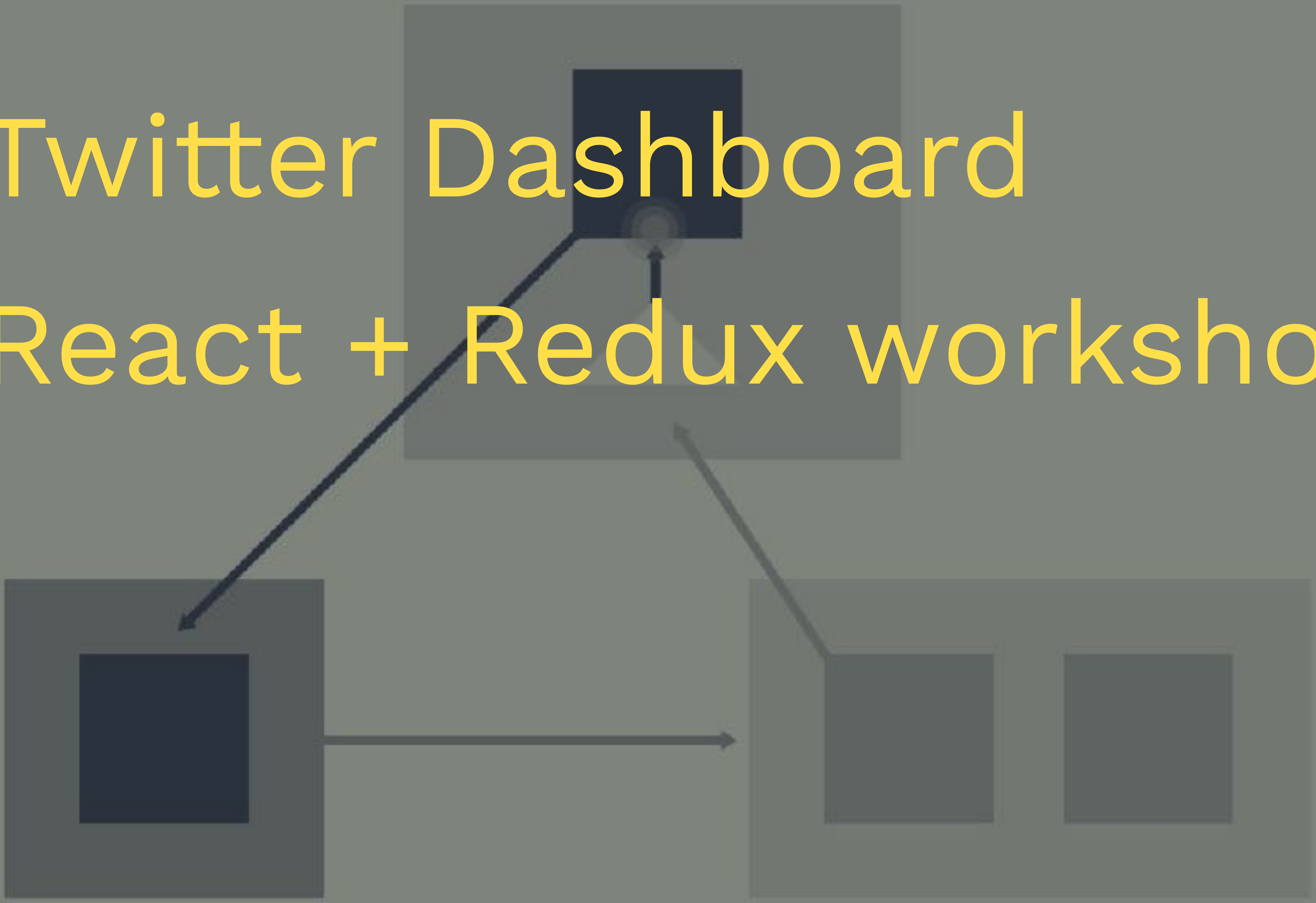


Twitter Dashboard

React + Redux workshop



// Ingar Almklov & Guro Seternes

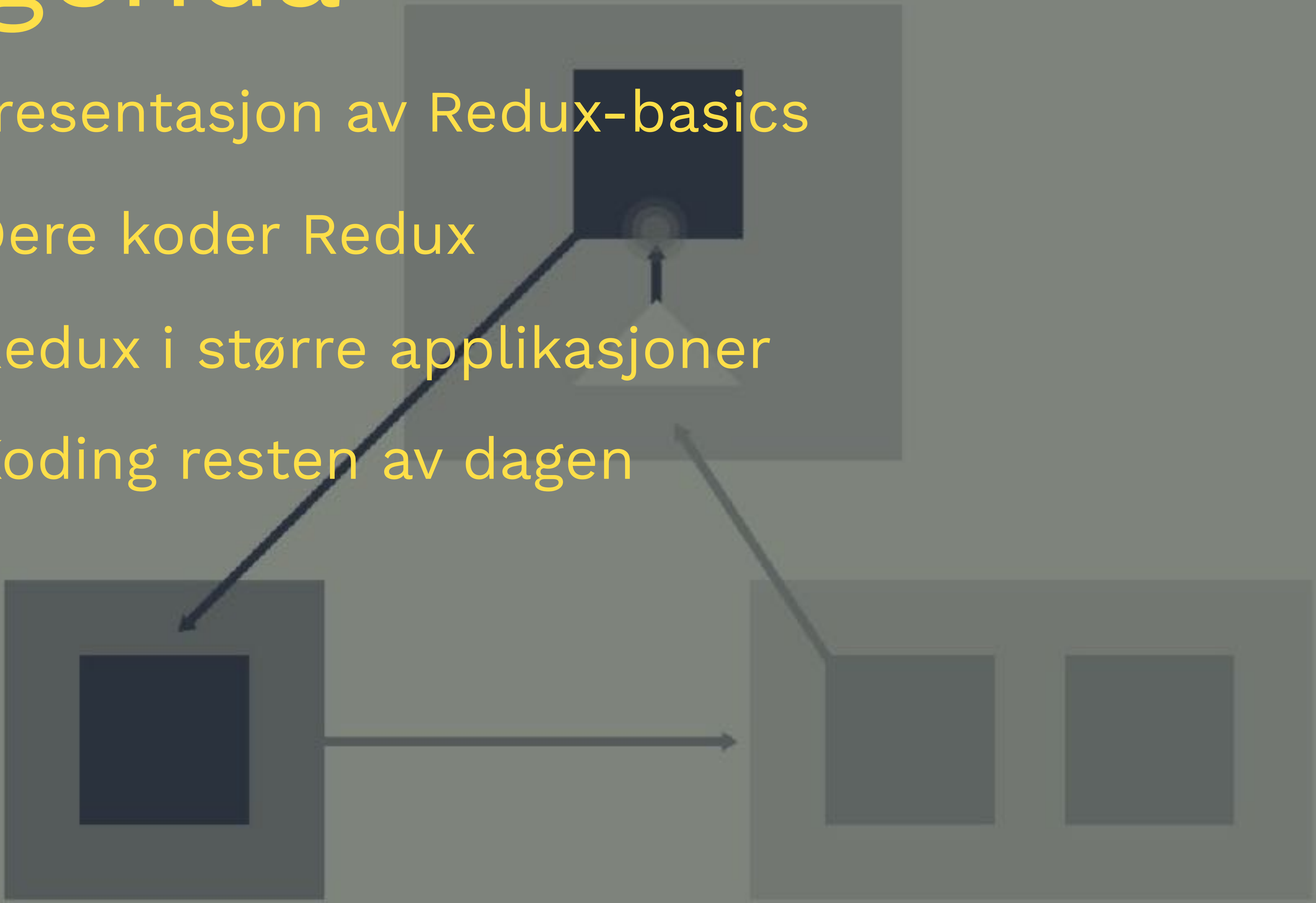
Agenda

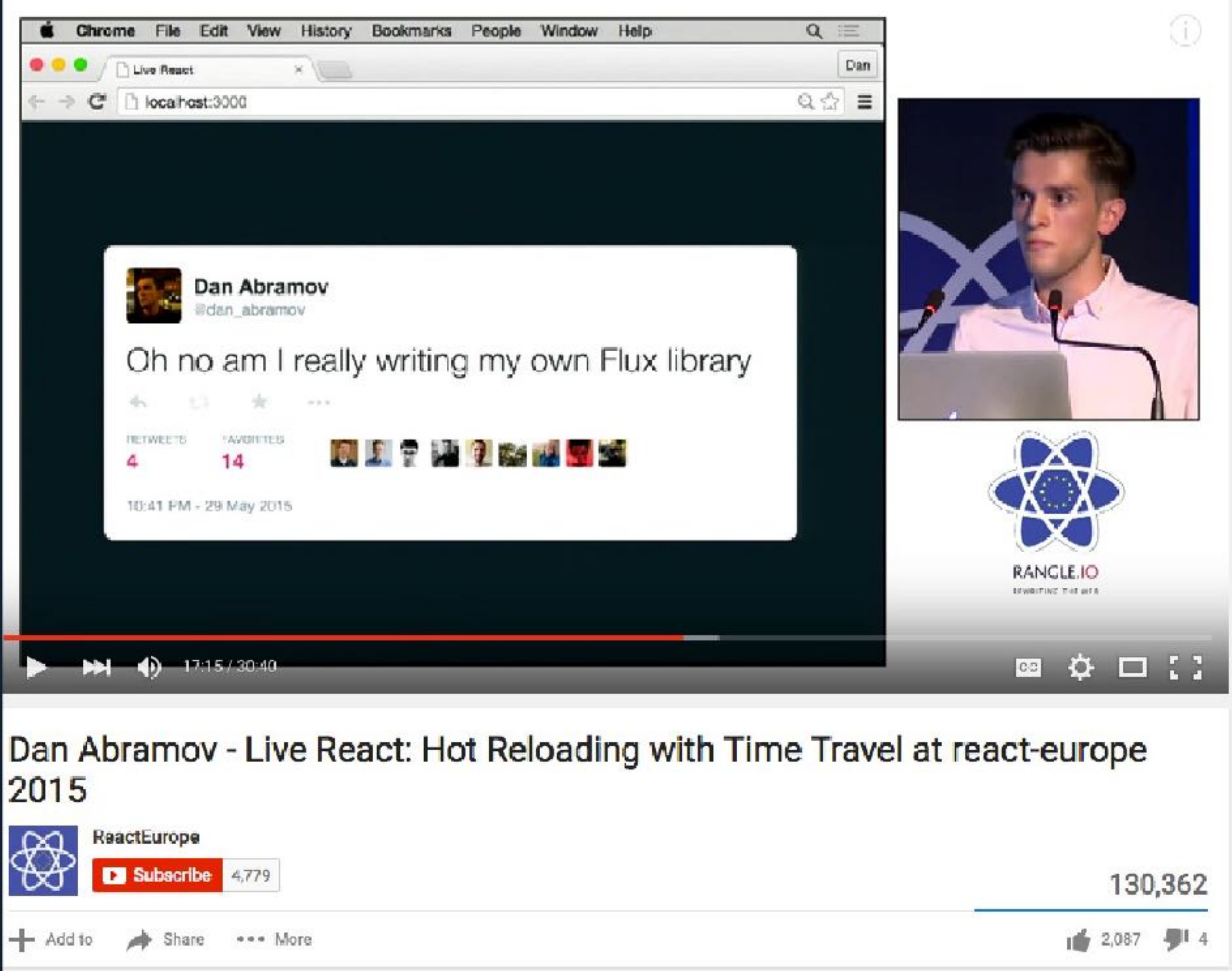
1. Presentasjon av Redux-basics

2. Dere koder Redux

3. Redux i større applikasjoner

2. Koding resten av dagen





Dan Abramov
@dan_abramov

Oh no am I really writing my own Flux library

RETWEETS 4 FAVORITES 14

10:41 PM - 29 May 2015

Dan Abramov - Live React: Hot Reloading with Time Travel at react-europe 2015

ReactEurope

Subscribe 4,779

130,362

2,087 4

Kvaliteter

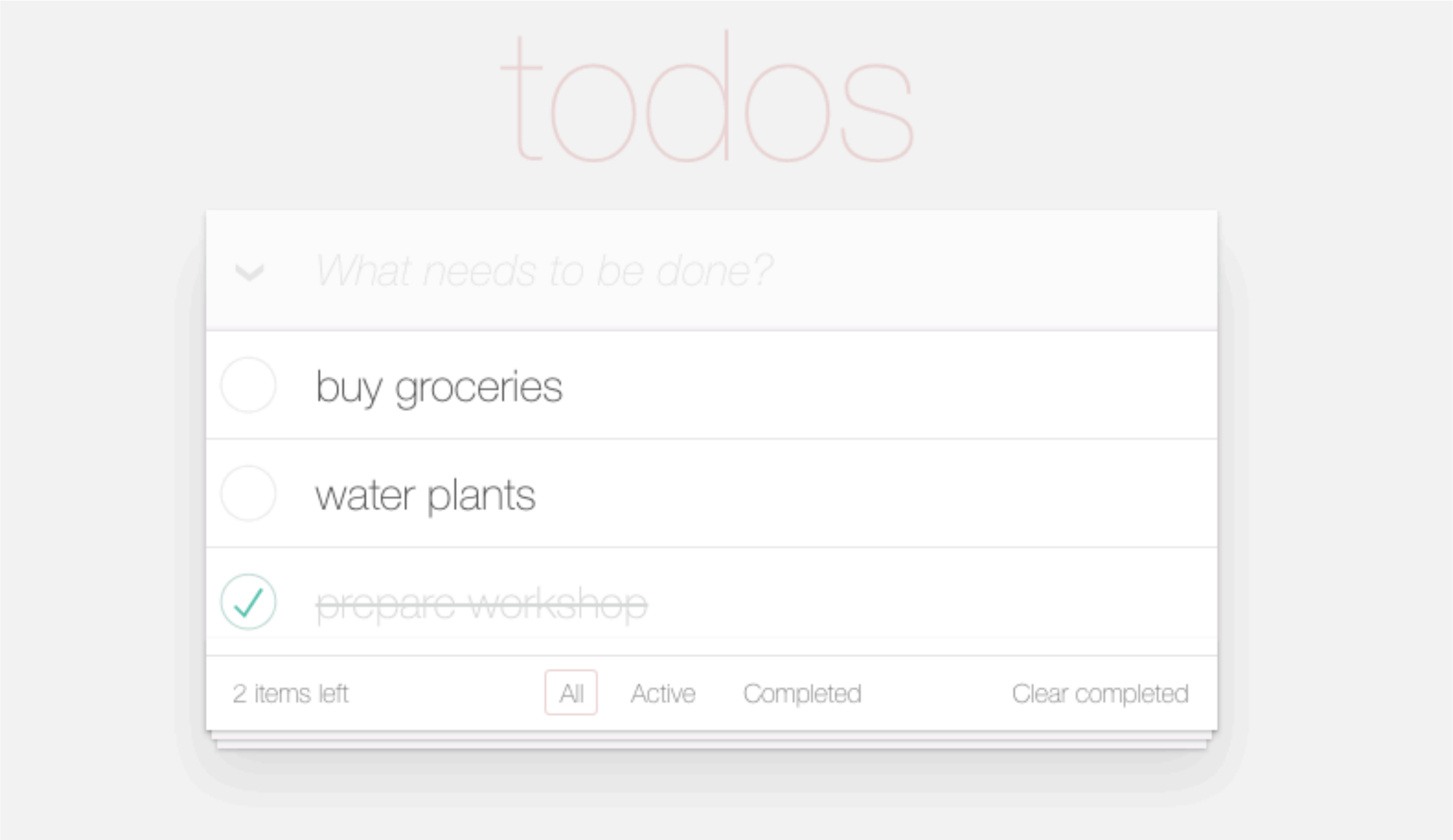
- funksjonelt inspirert
- svært testbart
- klar fordeling av ansvar
- egnet for store kodebaser med lang levetid

Eksempler på bad practice

```
var numberOfTodoItems= 0;

function numberOfTododsIsMoreThan(val) {
  return numberOfTodoItems > val;
}


function addTodoItem(newItem, items) {
  return items.push(newItem)
}
```

The Gist

» The whole state of your app is stored in a single object tree.



○ buy groceries

○ water plants

✓ ~~prepare workshop~~

2 items left

The Gist

- The whole state of your app is stored in a single object tree

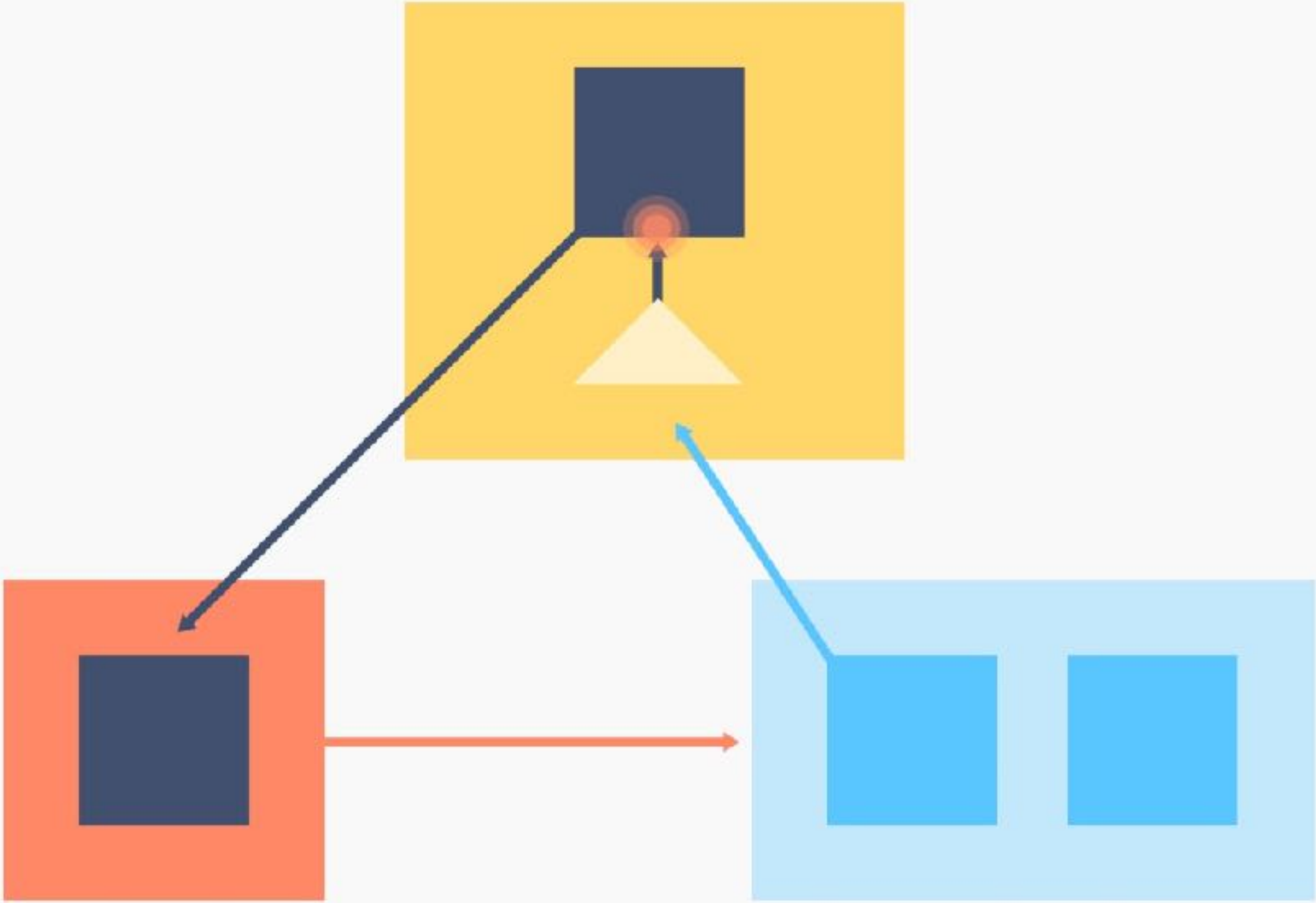
- The only way to change the state tree is to emit an action, an object describing what happened.

The Gist

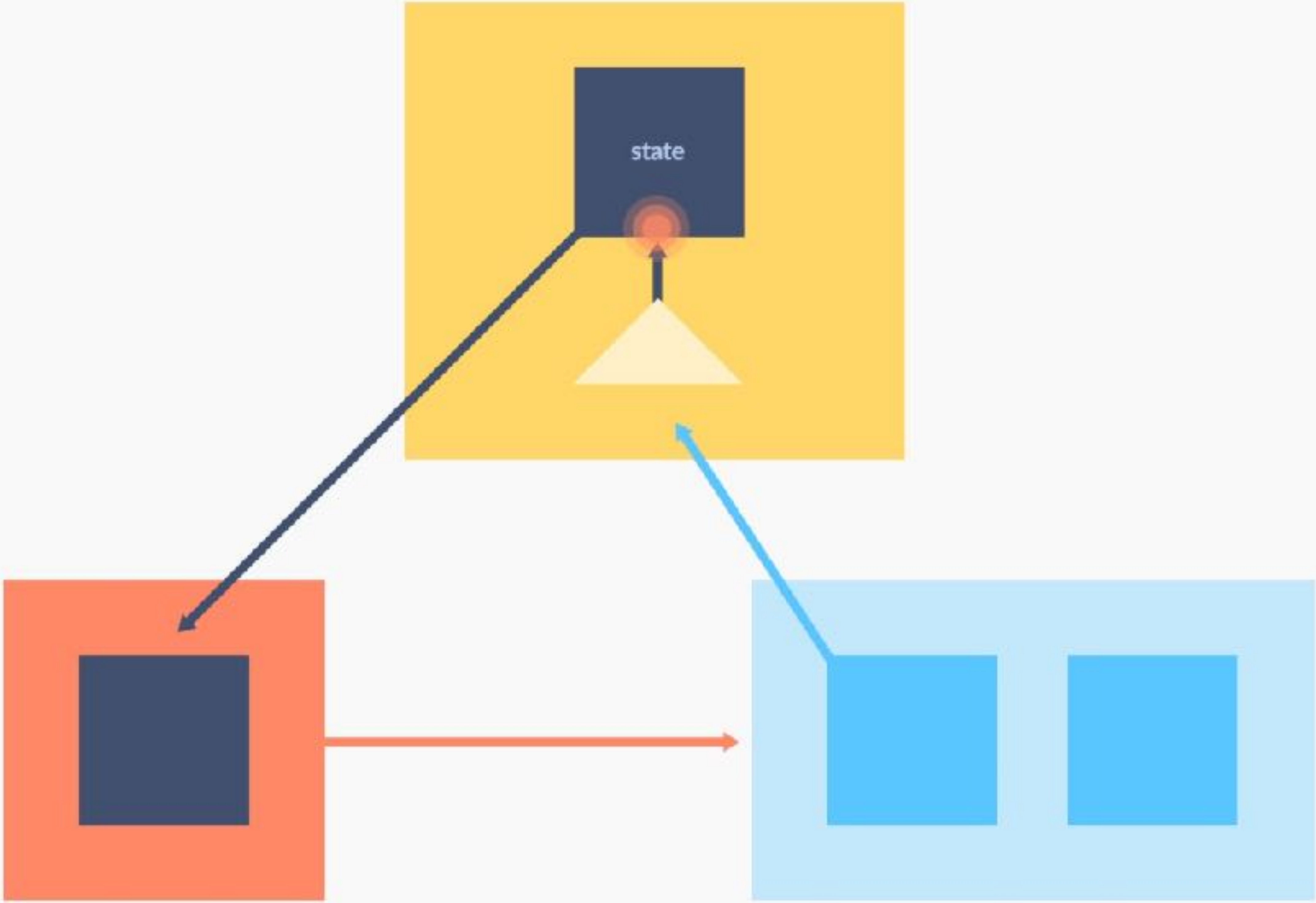
- The whole state of your app is stored in a single object tree

- The only way to change the state tree is to emit an action, an object describing what happened.

- To specify how the actions transform the state tree, you write stateless (pure) reducers.



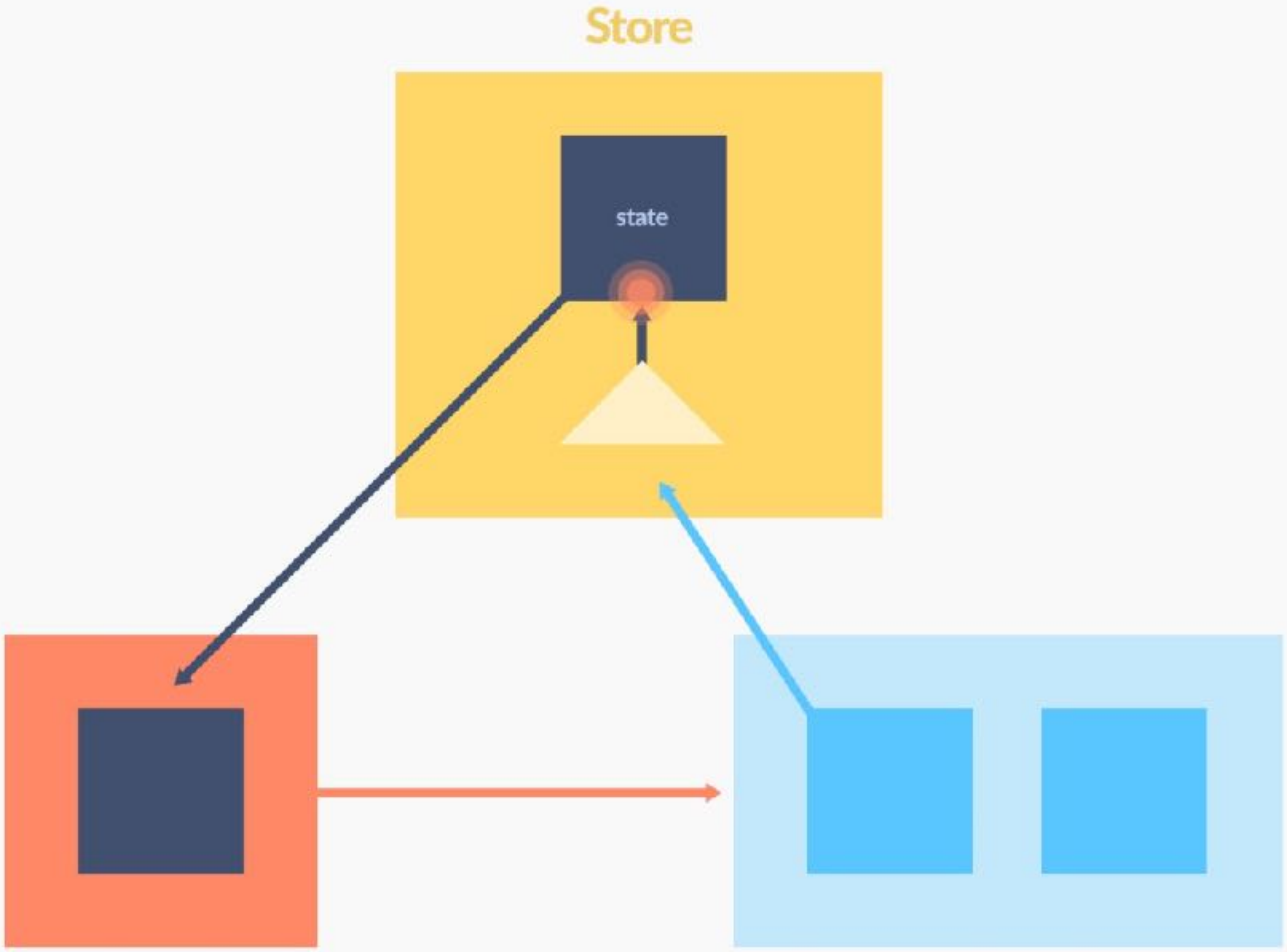
Store

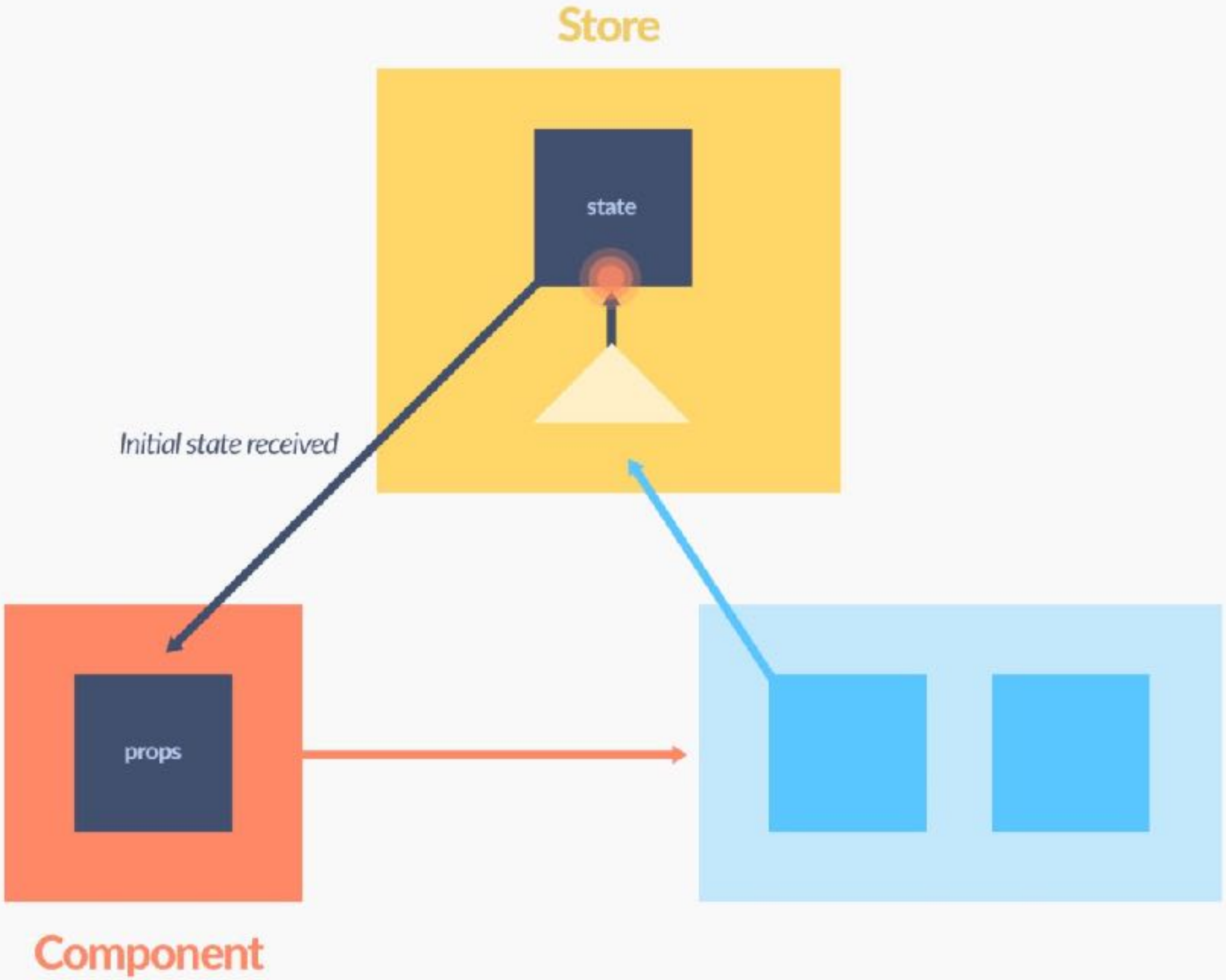


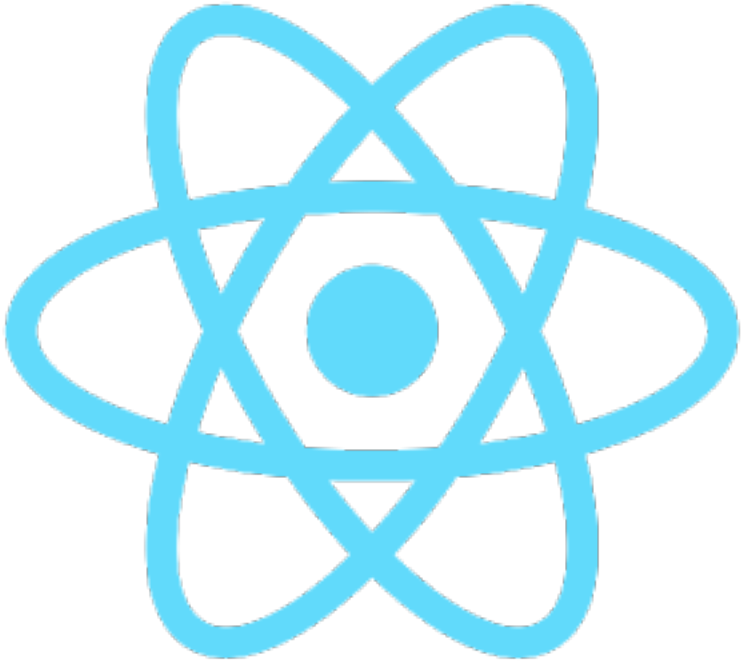
state

```
{
    todos: [],
```

```
{
  todos: [],
  activeFilter: 'completed',
}
```


Store
state


Store
state
Initial state received
props
Component



React

RE-RENDER
ALL THE THINGS
memegenerator.net

DATA => HTML

todos

}

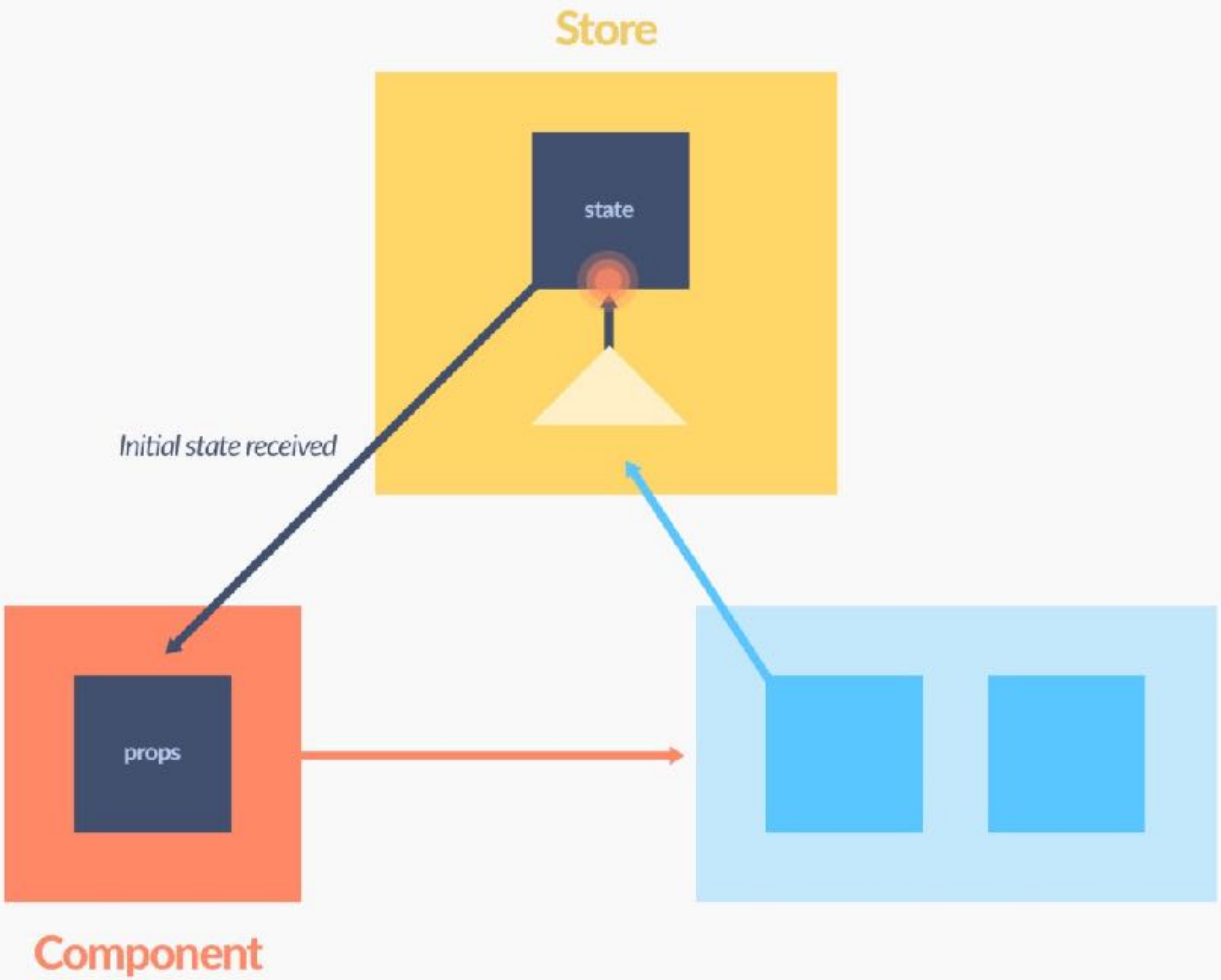
```
function HeadingComponent() {
  return
  <div>
    todos
  </div>
}
```

```
import { render } from 'react-dom';

render(<HeadingComponent/>, document.body);
```

todos for {props.name}

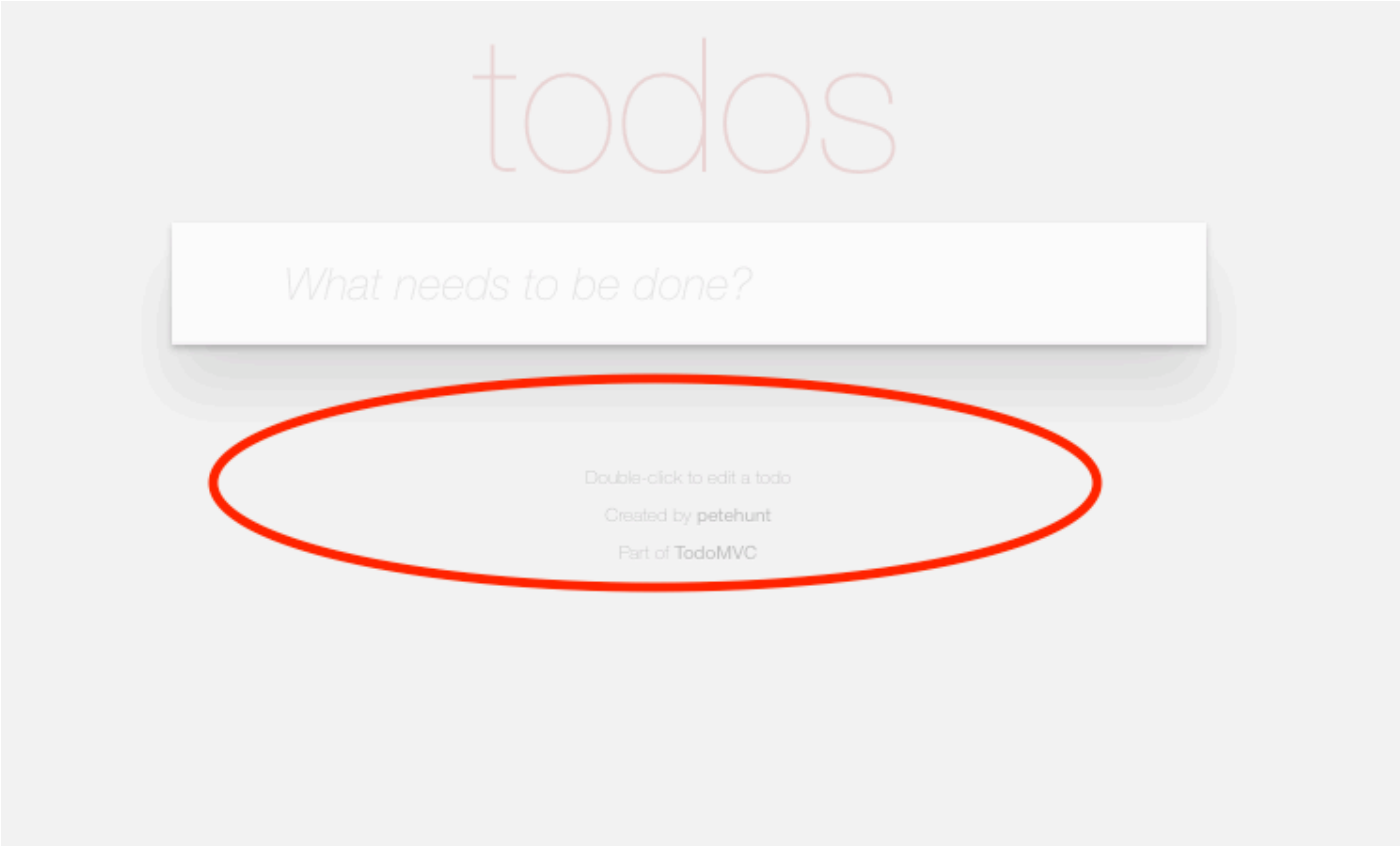
```
function HeadingComponent(props){
  return
  <div>
    todos for {props.name}
  </div>
}

render(<HeadingComponent name="Ingar"/>,
document.body)
```

```
function TodoAppComponent(){
  return
    <div>
      <HeadingComponent name="Ingar" />
      <TodoListComponent
        todos={['buy groceries', 'play fifa']}
      />
    </div>;
}
```


Store
state
Initial state received
props
Component

```
{
    todos: [],
```

```
{
  todos: [],
  activeFilter: 'completed',
}
```



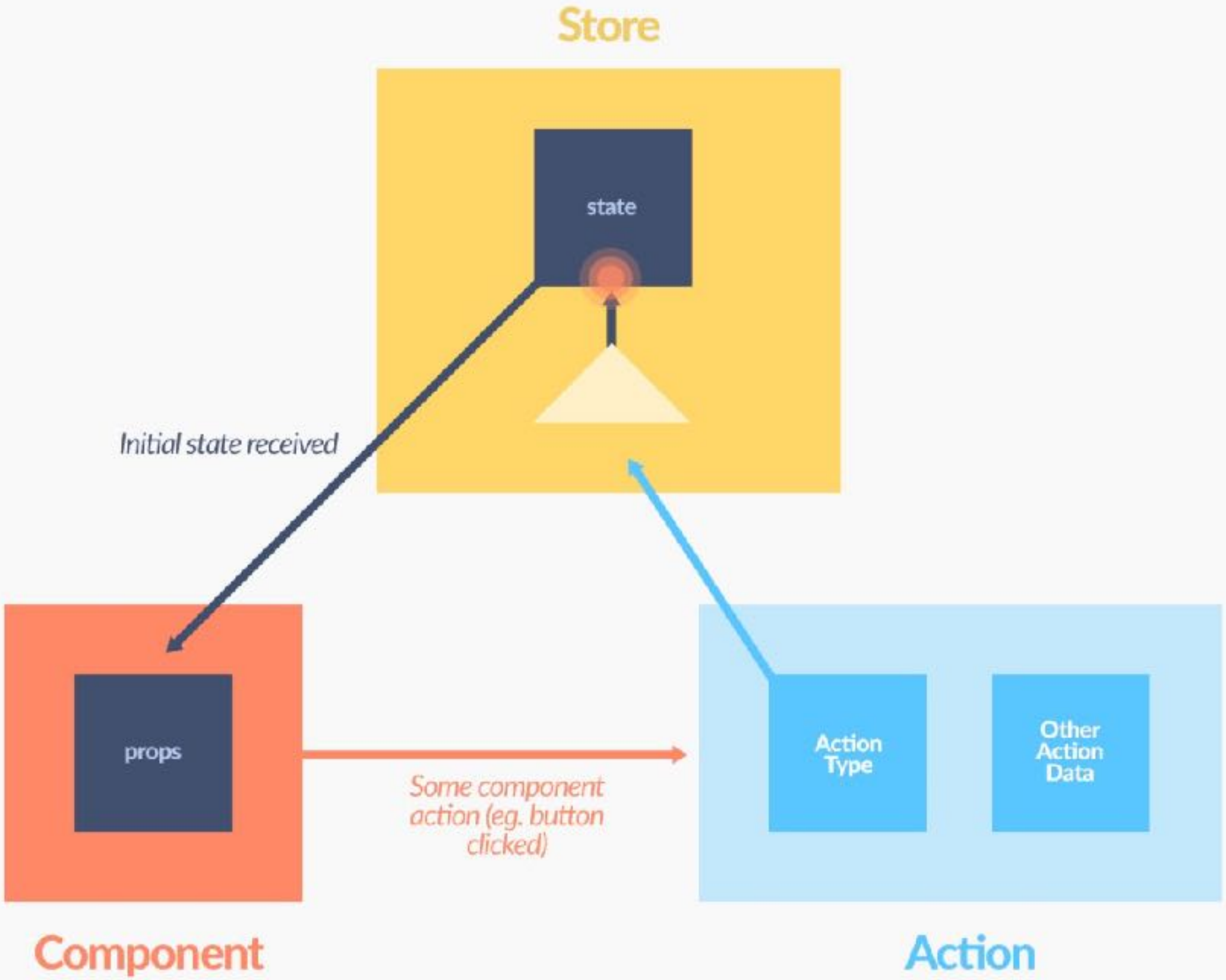
todos

What needs to be done?

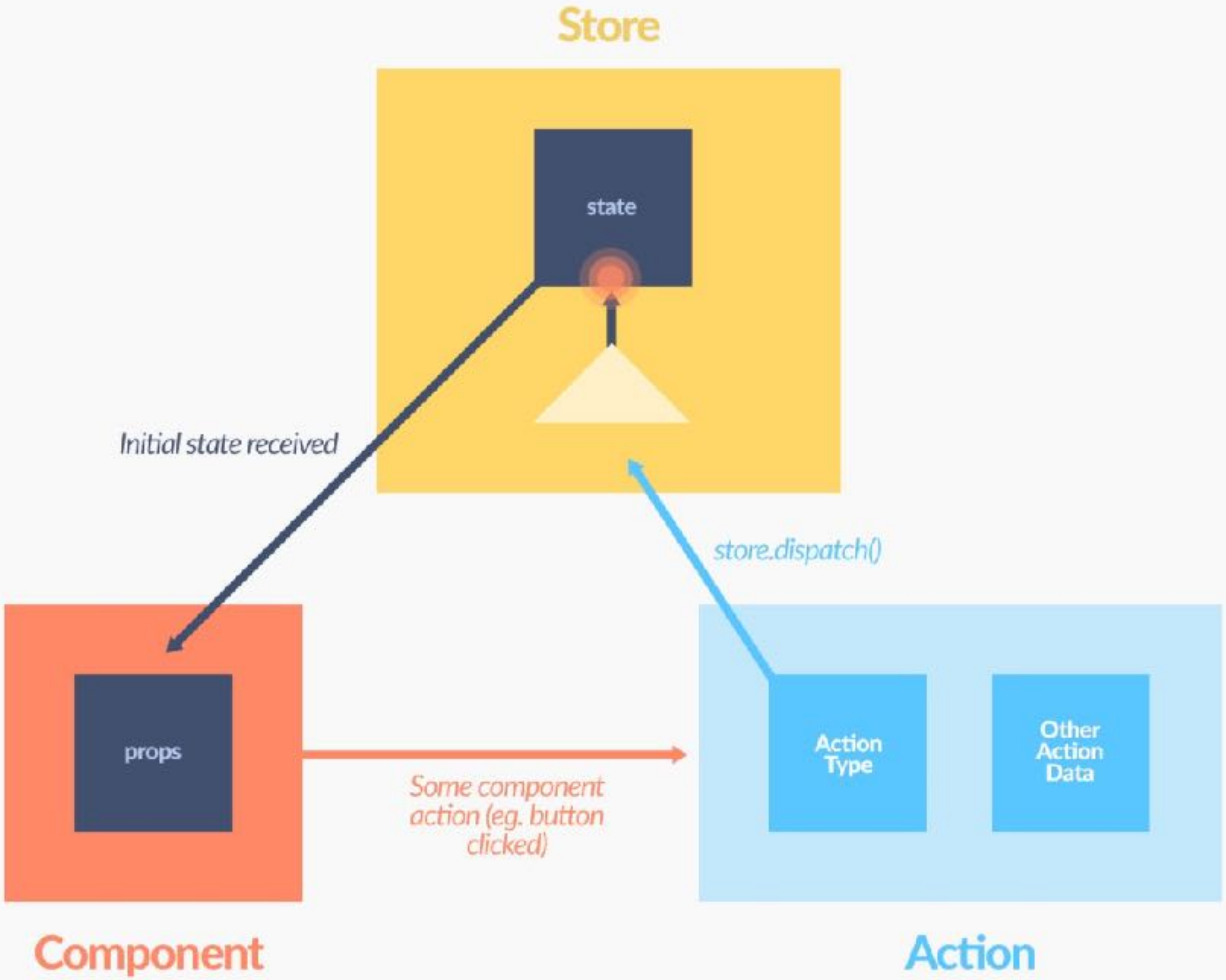
Double-click to edit a todo

Created by petehunt

Part of TodoMVC


Store
state
Initial state received
props
Some component action (eg. button clicked)
Action Type
Other Action Data
Component
Action

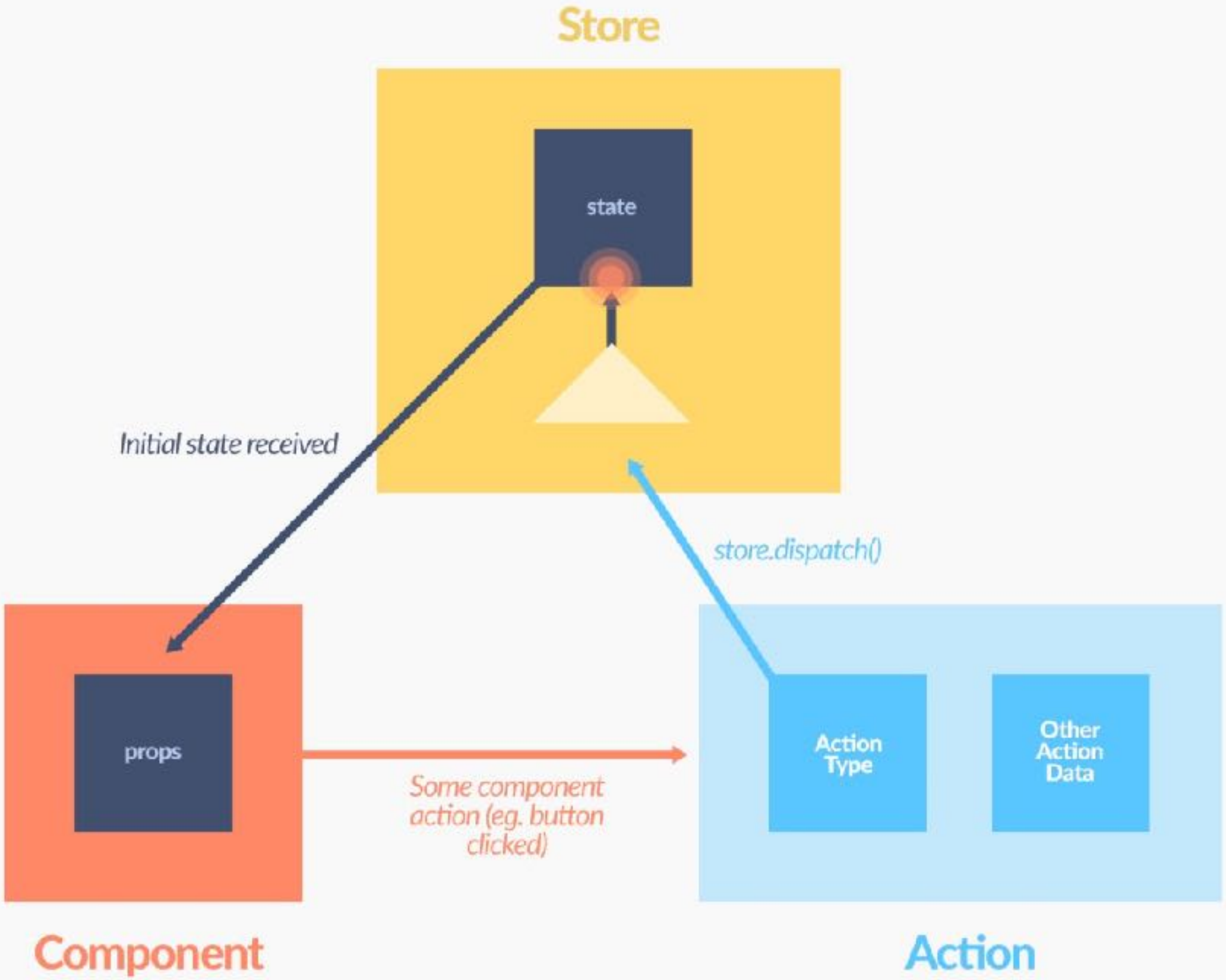
```
{
  // action type
  type: 'USER_ADDED_TODO',

  // action data
  item: 'buy groceries'
}
```

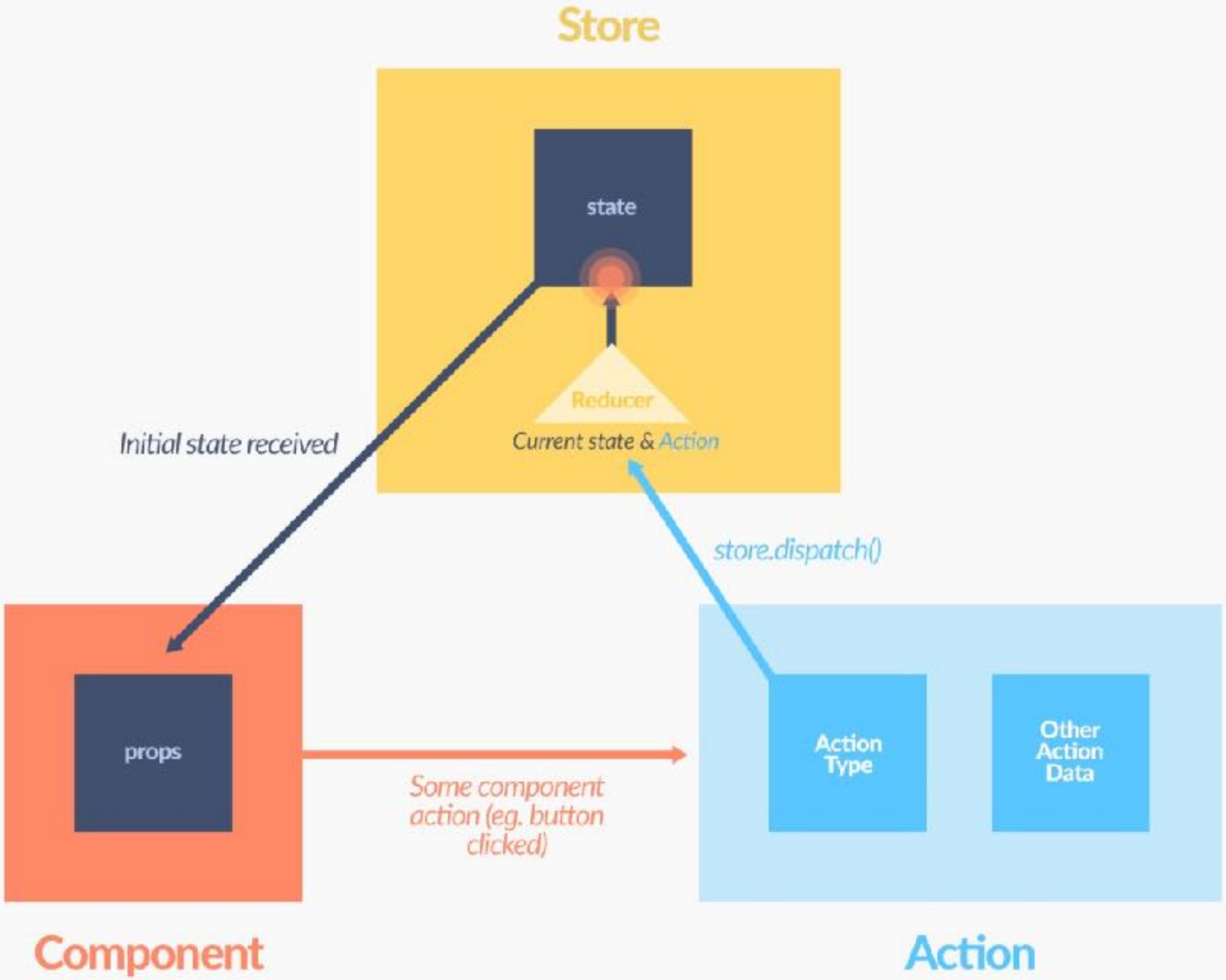

Store
state
Initial state received
store.dispatch()
props
Action Type
Other Action Data
Some component action (eg. button clicked)
Component
Action

})

```
store.dispatch({
  type: 'USER_ADDED_TODO',
  item: 'water plants'
})
```

```
function userAddedNewItem(item) {
  return {
    type: 'USER_ADDED_TODO',
    item
  }
}

store.dispatch(userAddedNewItem('water
plants'));
```


Store
state
Initial state received
store.dispatch()
props
Some component action (eg. button clicked)
Action Type
Other Action Data
Component
Action



Store

state

Reducer

Current state & Action

Initial state received

store.dispatch()

props

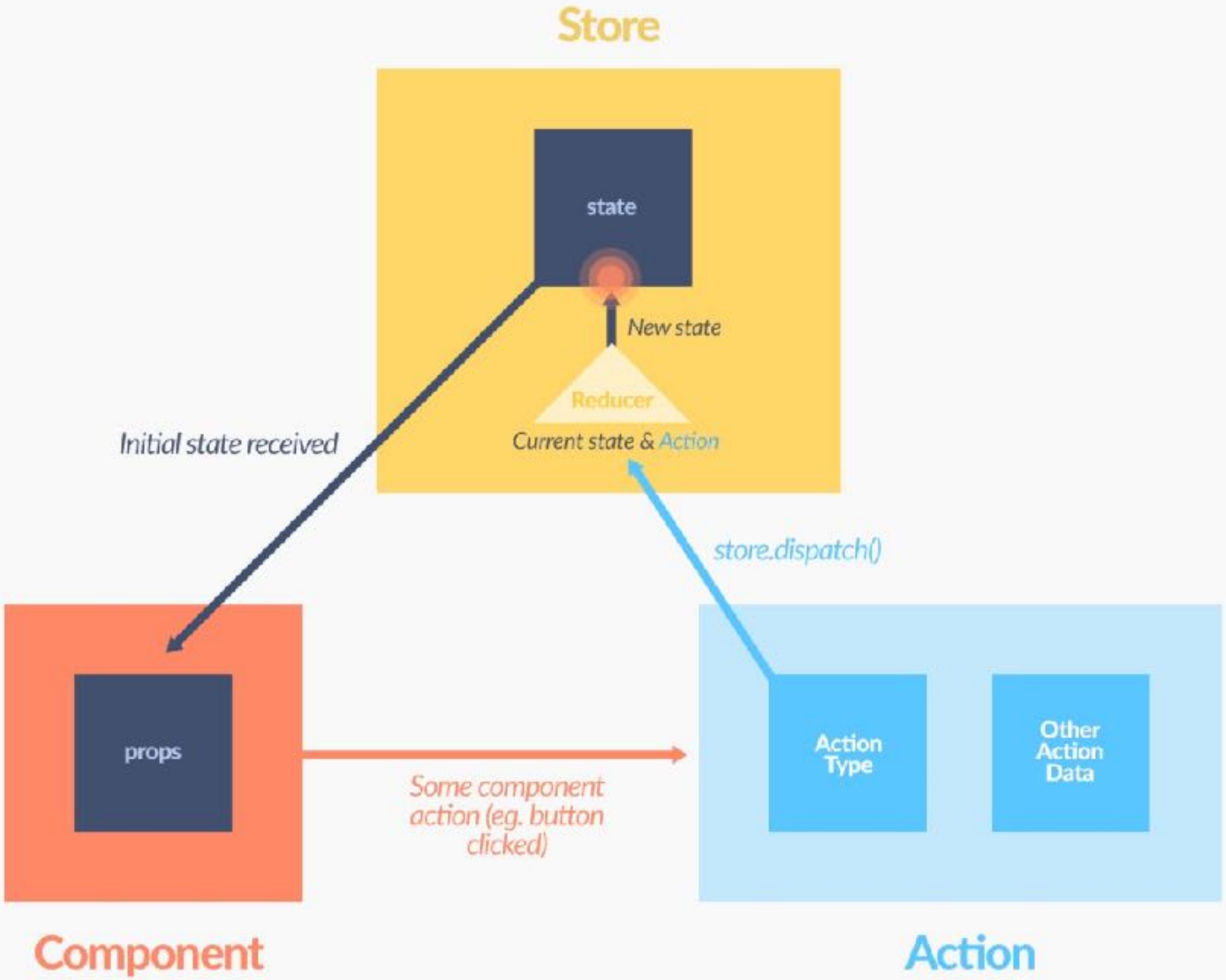
Some component action (eg. button clicked)

Action Type

Other Action Data

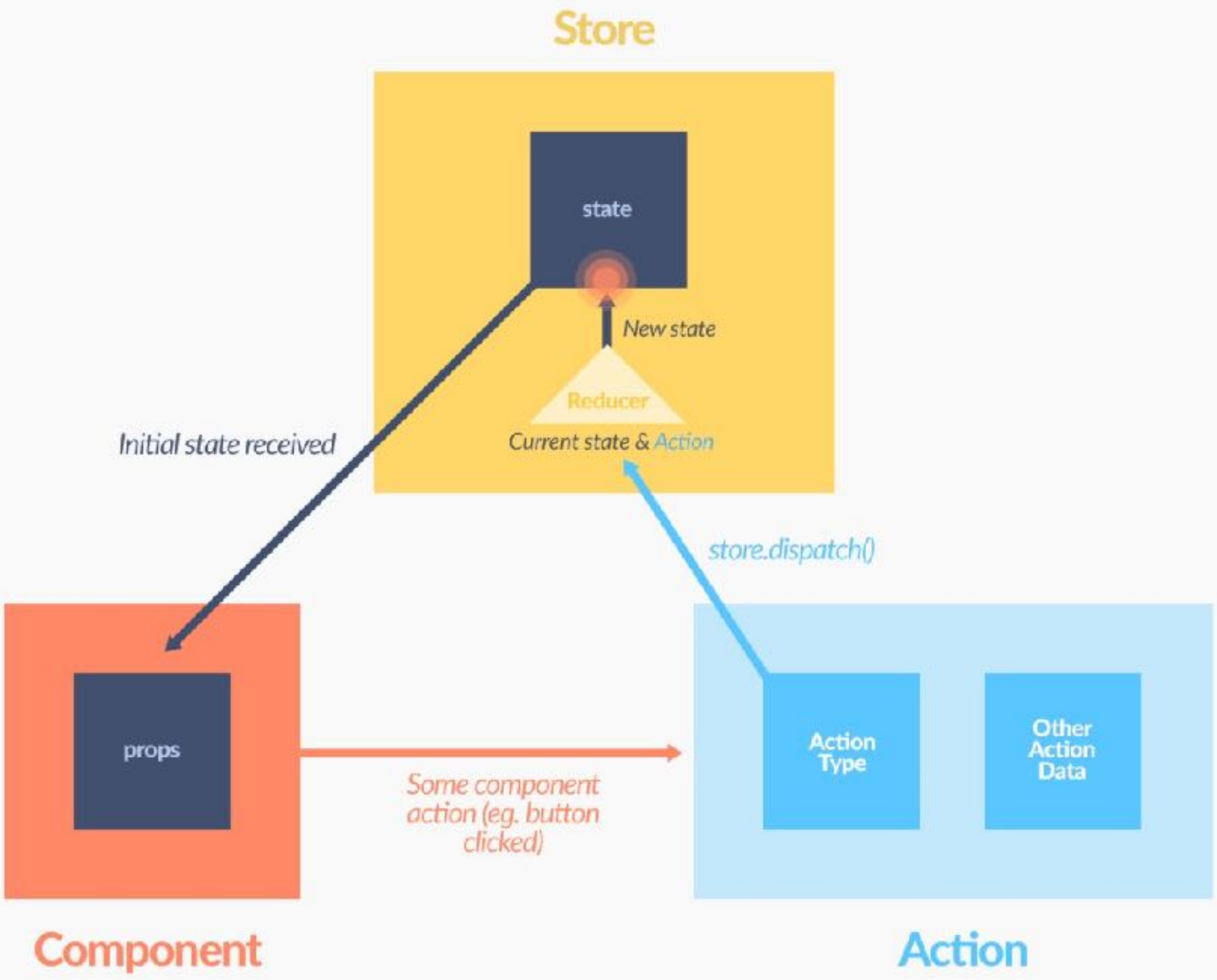
Component

Action

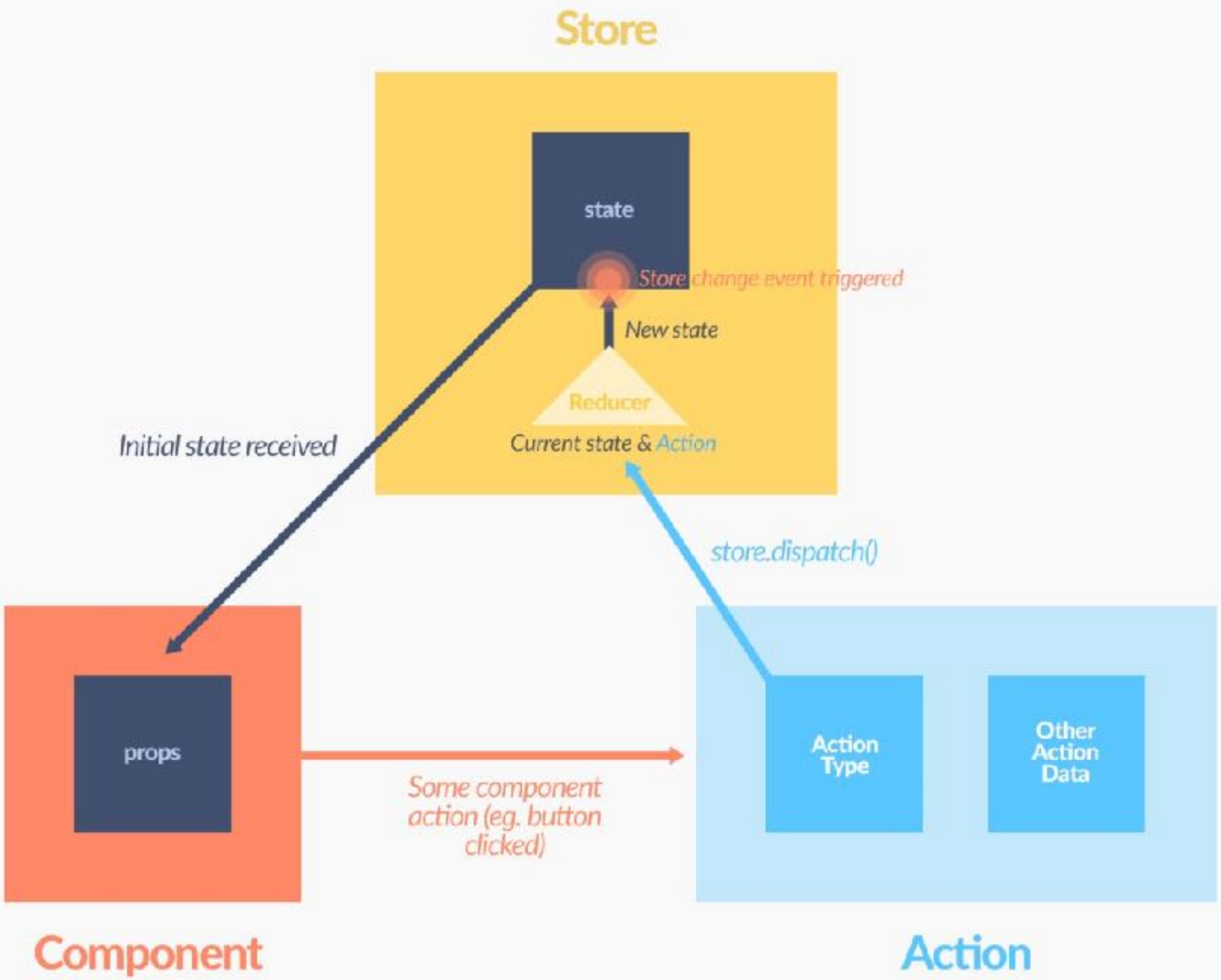
Store
state
New state
Reducer
Current state & Action
Initial state received
store.dispatch()
props
Some component action (eg. button clicked)
Action Type
Other Action Data
Component
Action

state, action => state

```
function reducer(state=initialState, action) {
  if (action.type === 'TODO_ITEM_ADDED') {
    return [...state, action.item];
  }
  return state;
}
```



Store
state
New state
Reducer
Current state & Action
Initial state received
store.dispatch()
props
Some component action (eg. button clicked)
Action Type
Other Action Data
Component
Action



Store

state

Store change event triggered

New state

Reducer

Current state & Action

Initial state received

store.dispatch()

props

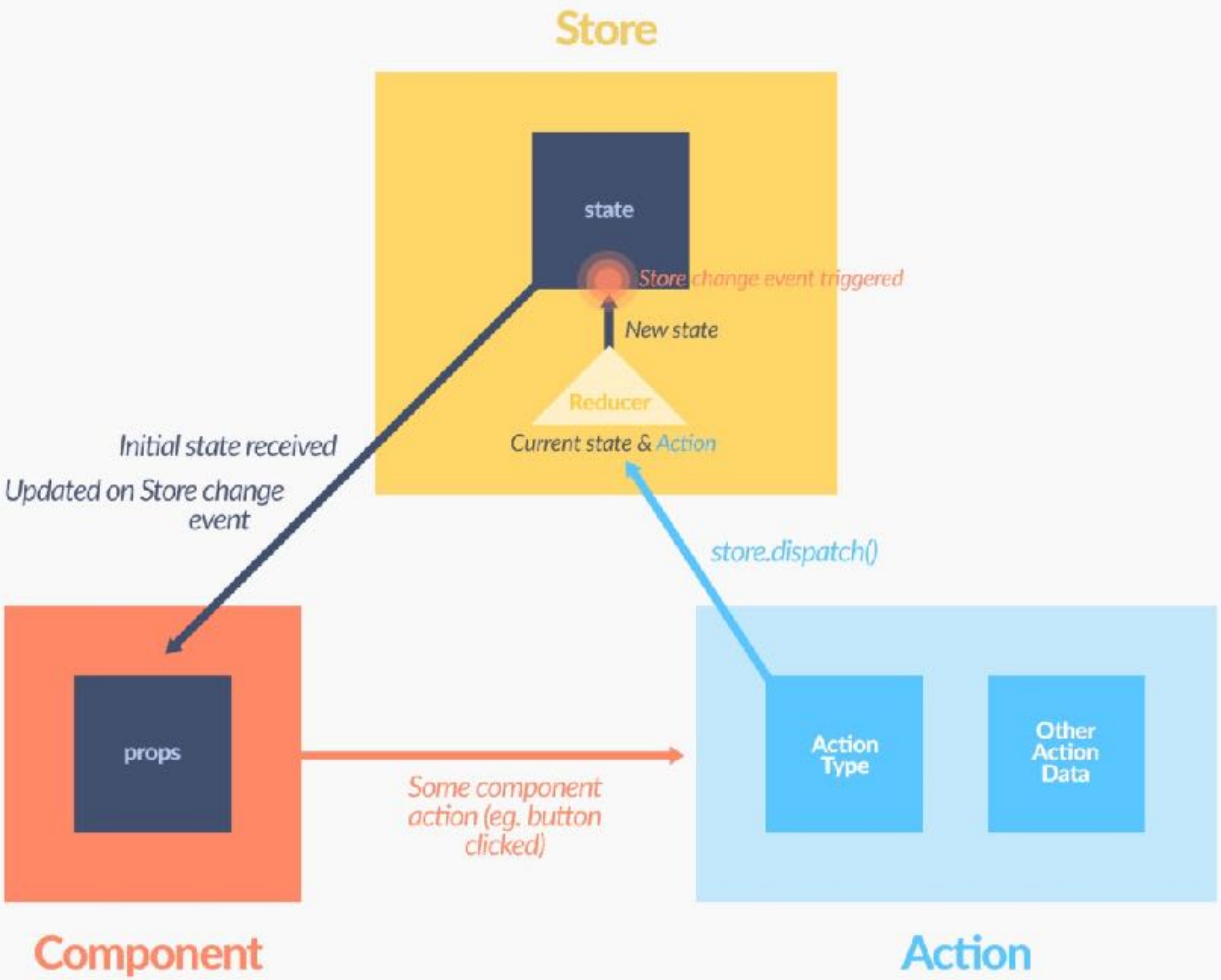
Some component action (eg. button clicked)

Action Type

Other Action Data

Component

Action



Store

state

Store change event triggered

New state

Reducer

Current state & Action

Initial state received

Updated on Store change event

store.dispatch()

props

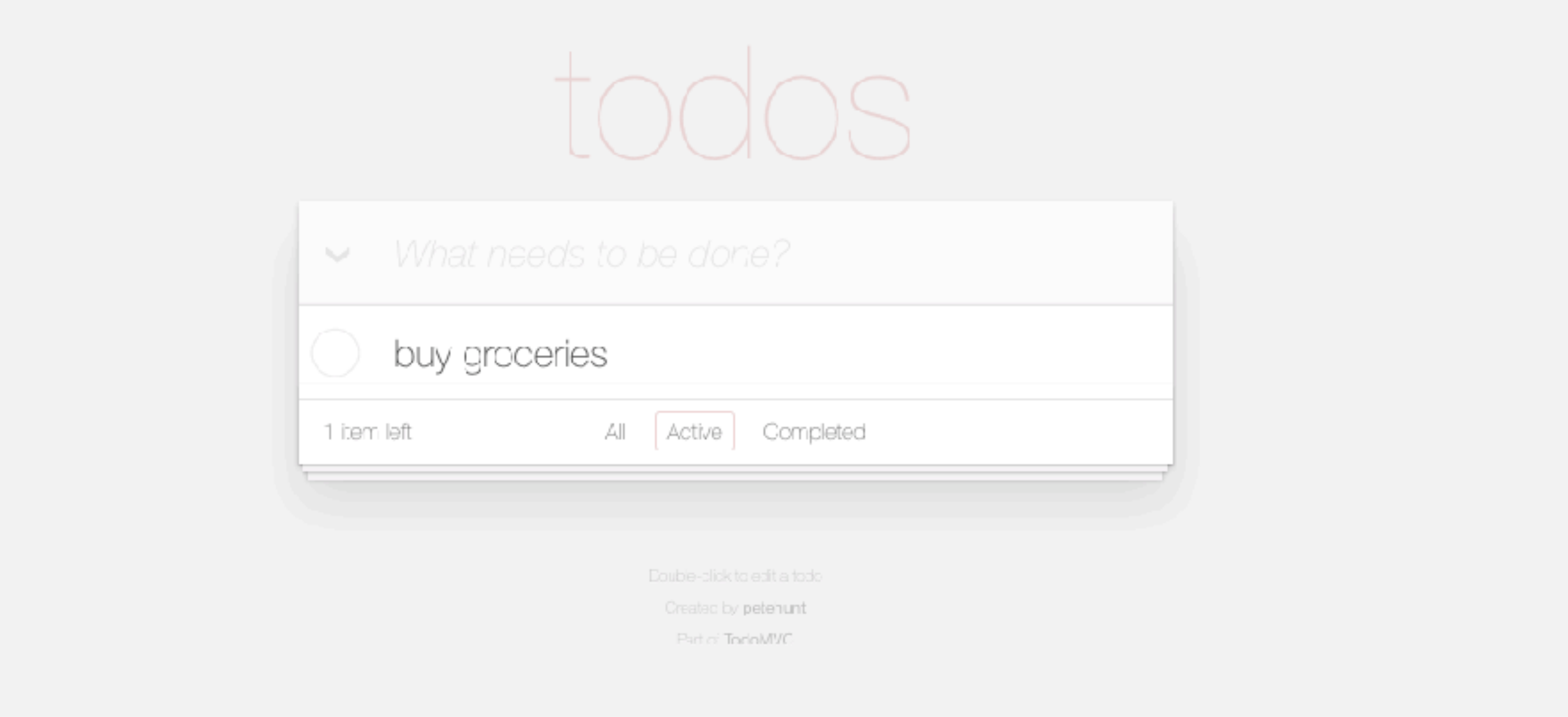
Some component action (eg. button clicked)

Action Type

Other Action Data

Component

Action



buy groceries

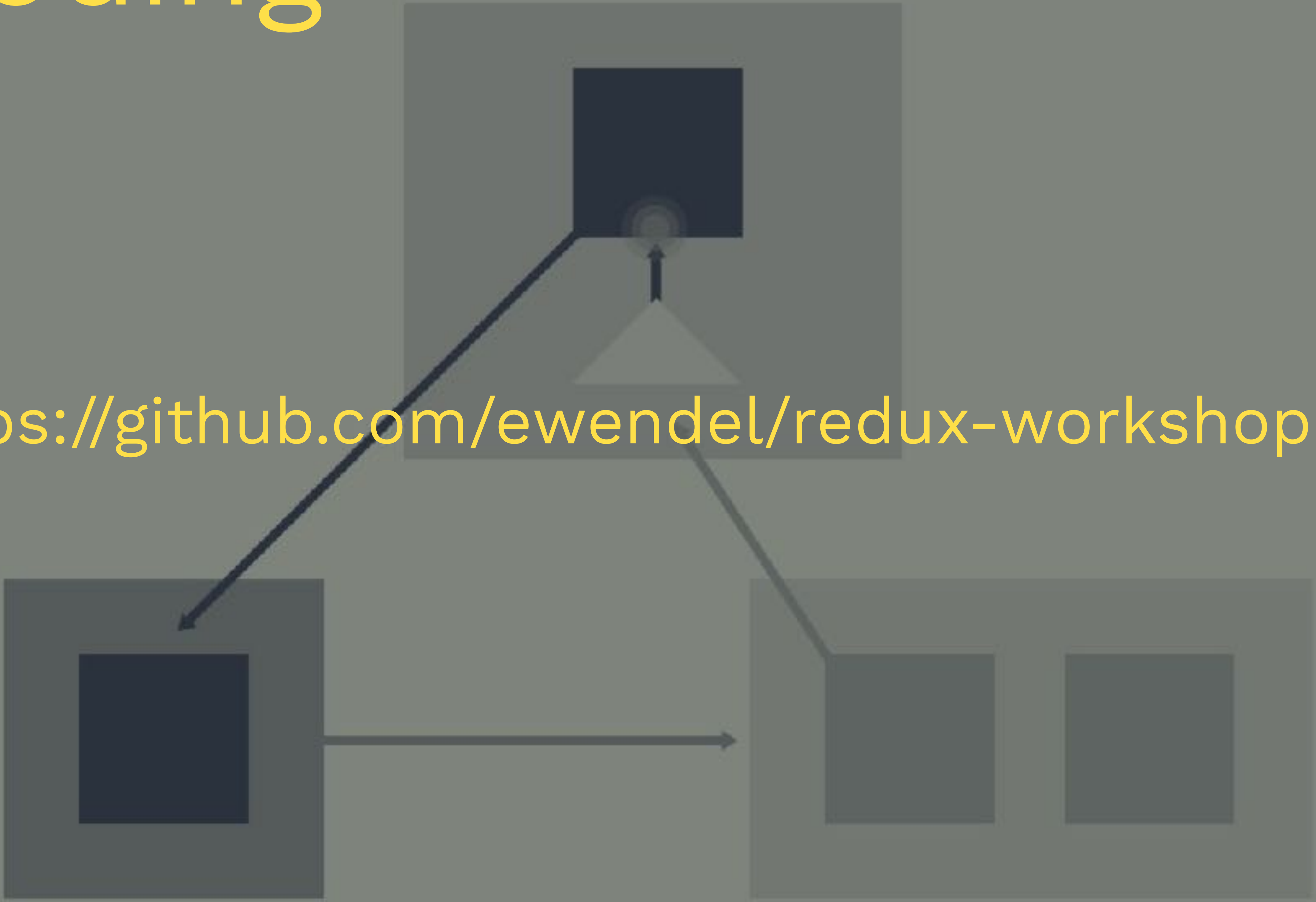
1 item left

Double-click to edit a todo

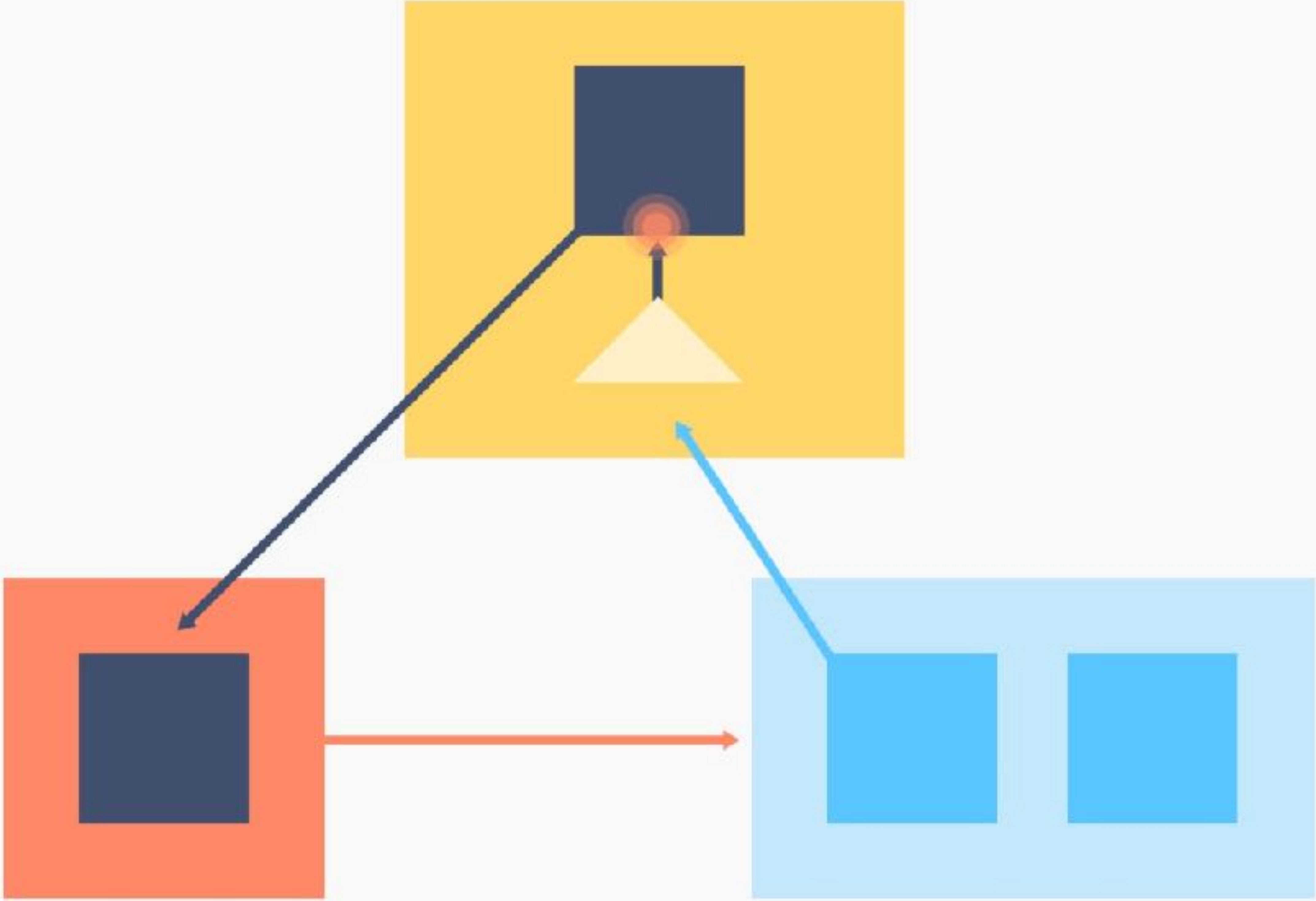
Created by petehunt

Part of TodoMVC

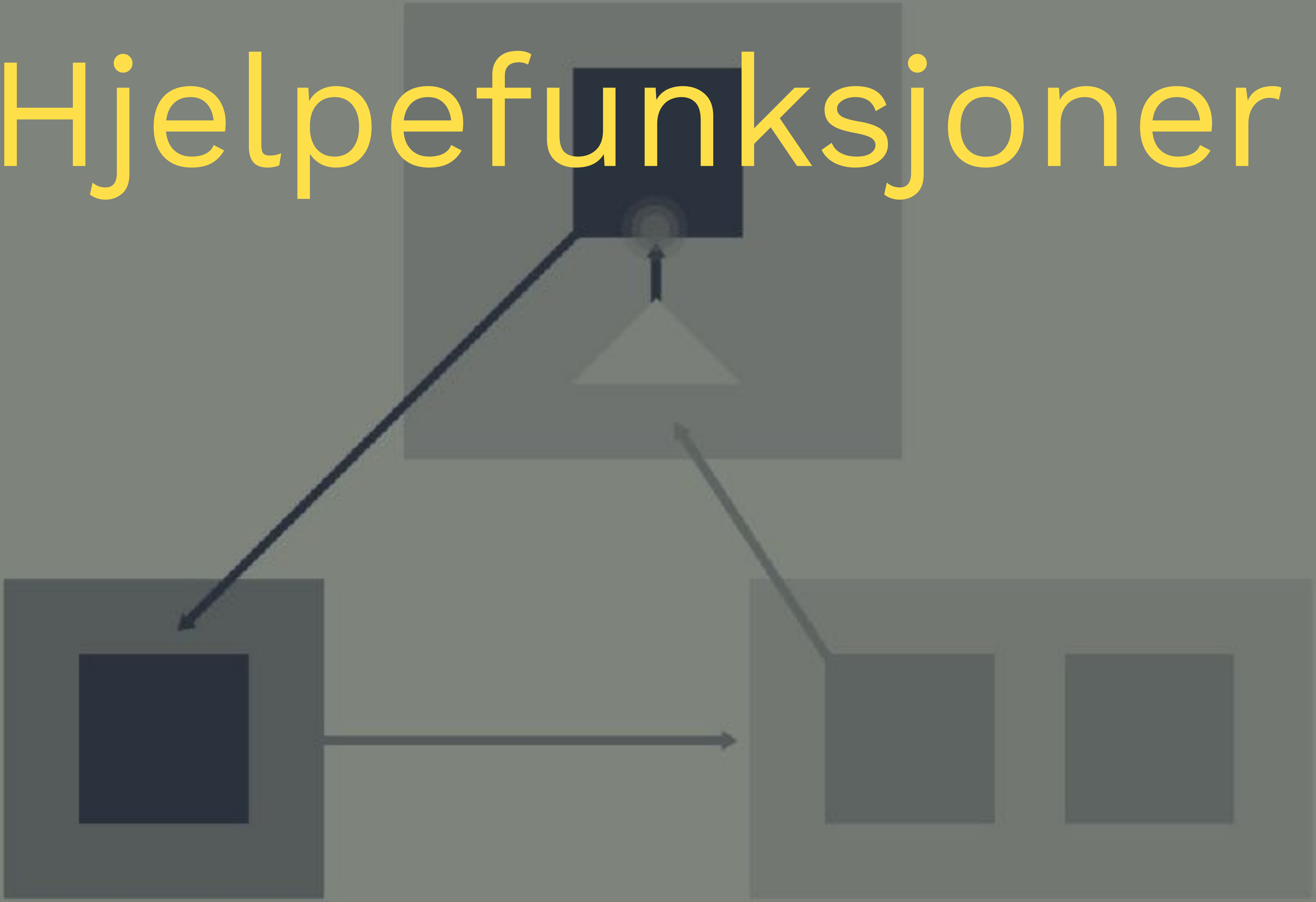
Koding



https://github.com/ewendel/redux-workshop



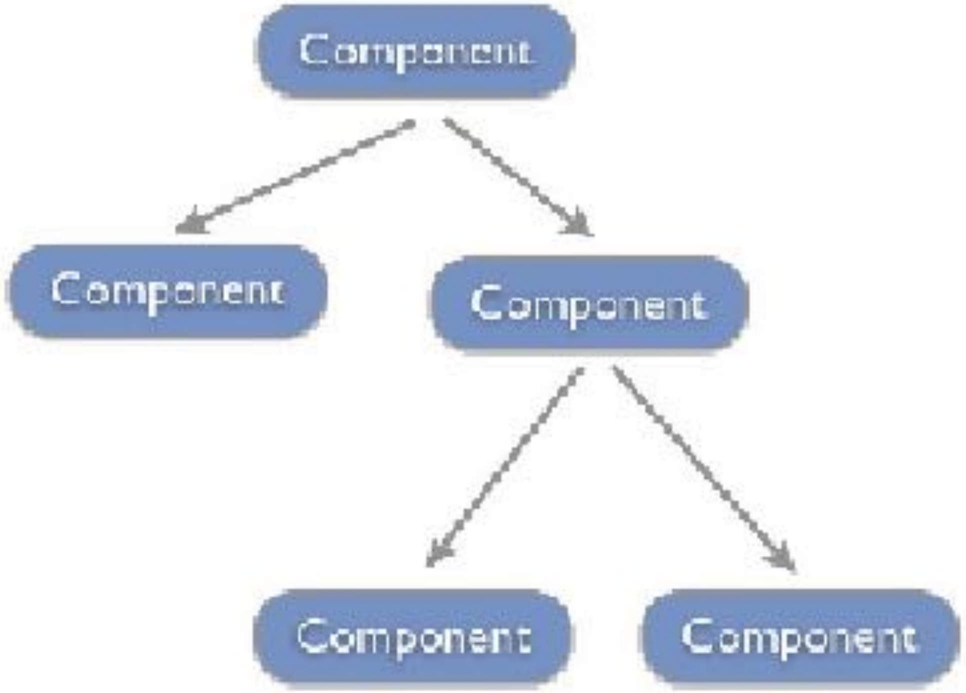
Hjelpefunksjoner



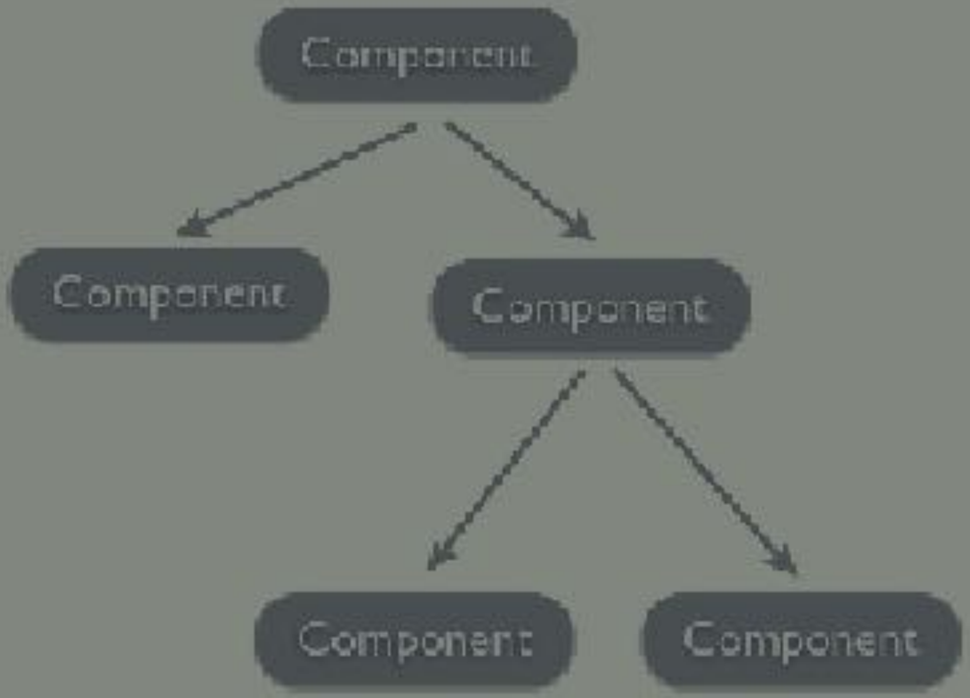
```
import { combineReducers } from 'redux';

import { connect, Provider } from 'react-redux';
```

```
combineReducers({
  tweets: tweetReducer,
  viewState: viewStateReducer,
  route: routeReducer
})
```

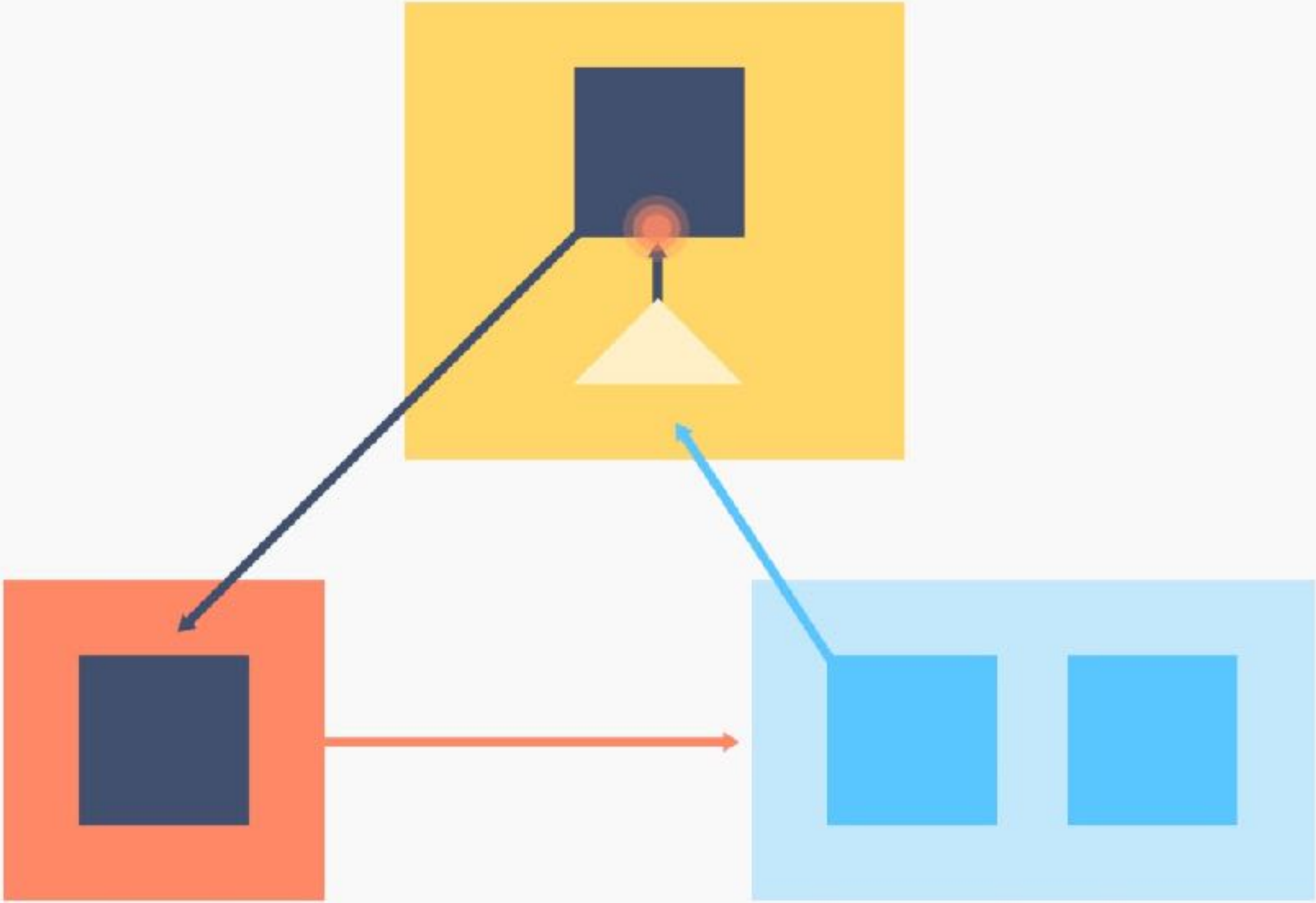
```
const mapStateToProps = state => {
  return {
    hasTweets: state.tweets.length > 0,
    mostRecentTweet: state.tweets[0]
  }
};

connect(mapStateToProps)(MyComponent):
```


Component
Component
Component
Component
Component

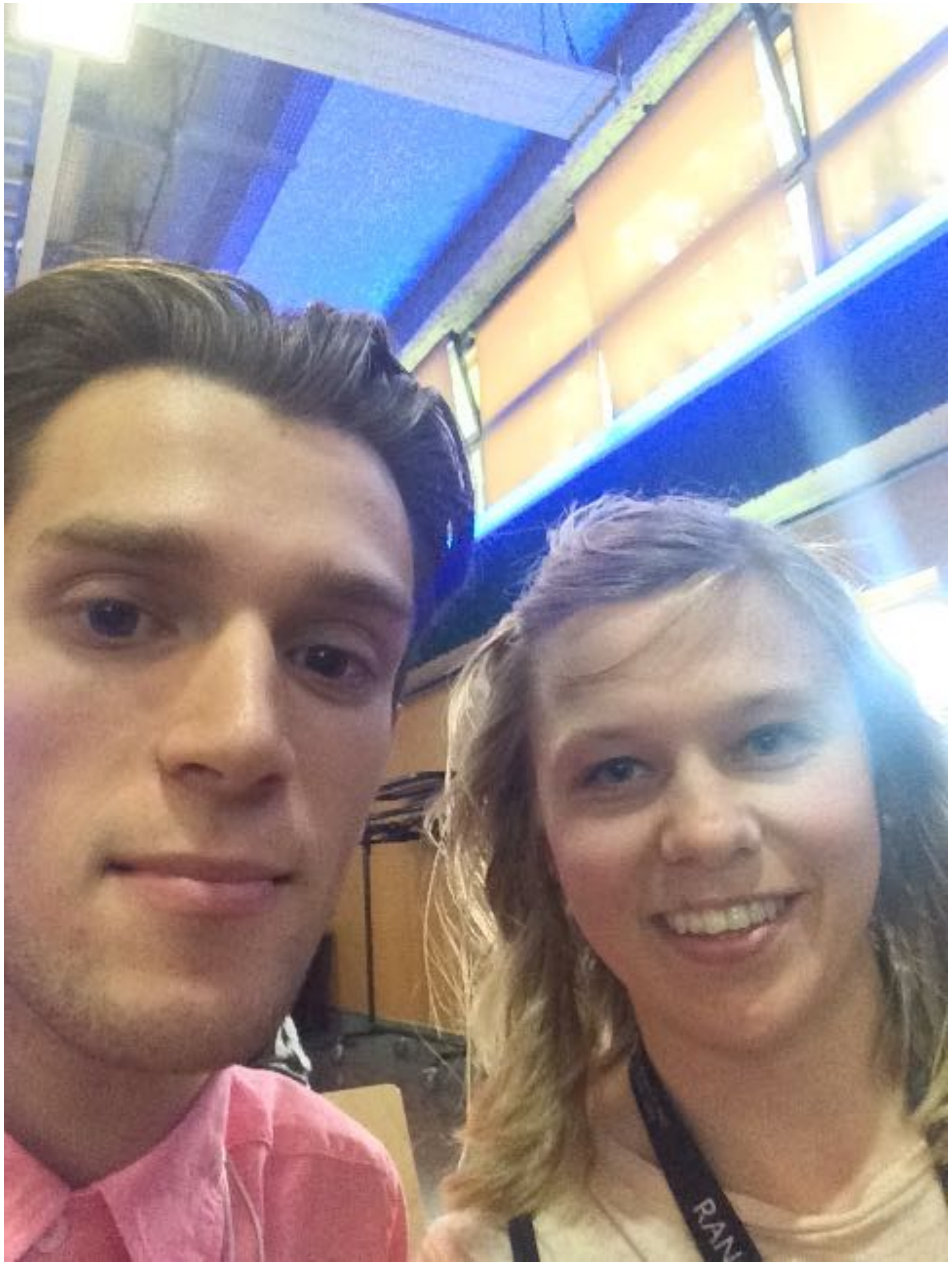
Provider



```
<Provider store={store}>
  <App/>
</Provider>
```



Mannen
bak Redux



```
function (state = [], action) {
    switch (action.type) {
    case TWEET_RECEIVED:
            return [
                    …state,
                    action.tweet
            ] ;
    default :
            return state;
    }
}
```