ANECO ACADEMY

REACTJS - FLUX-REDUX

What is flux

Flux is a Javascript architecture or pattern for UI which runs on a unidirectional data flow and has a centralized dispatcher.

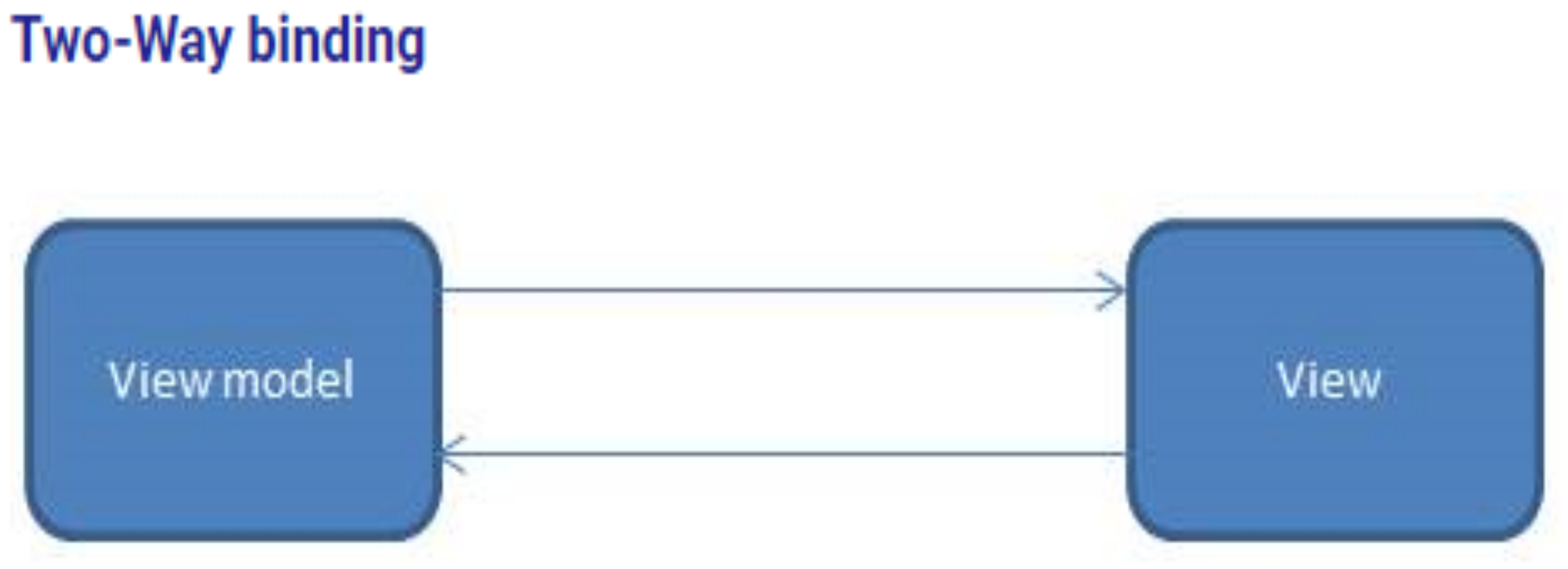
It is useful when your project has dynamic data and you need to keep the data updated in an effective manner.

It was created by Facebook, and complements React as view. This model is used to ease maintenance.

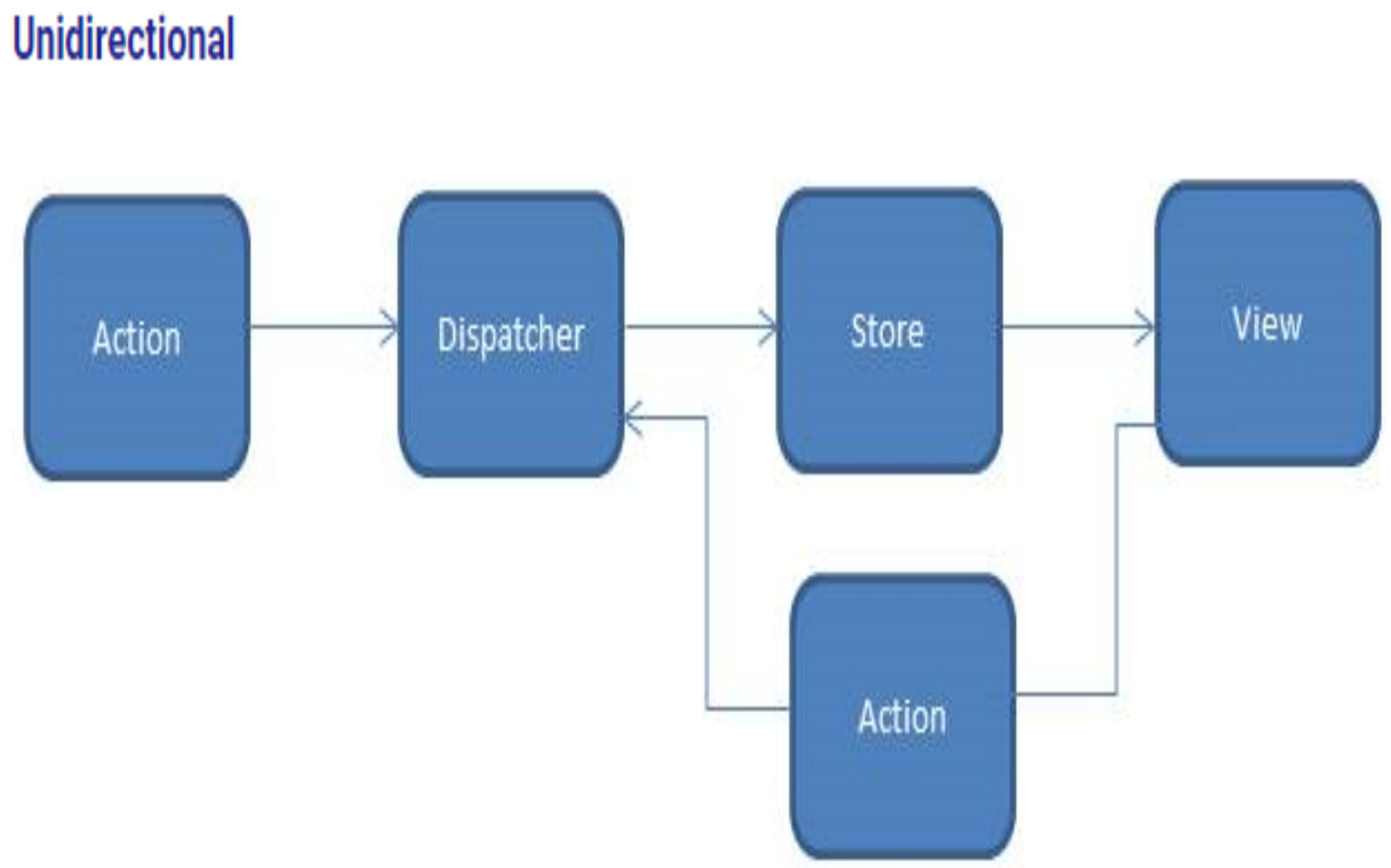
It has three primary components: Views, Stores, and Dispatcher.

As the MVC application grows, we find many numbers of views as models, which are all talking to each other, making it difficult to manage and debug.

TWO WAY DATA BINDING



UNIDIRECTIONAL FLOW



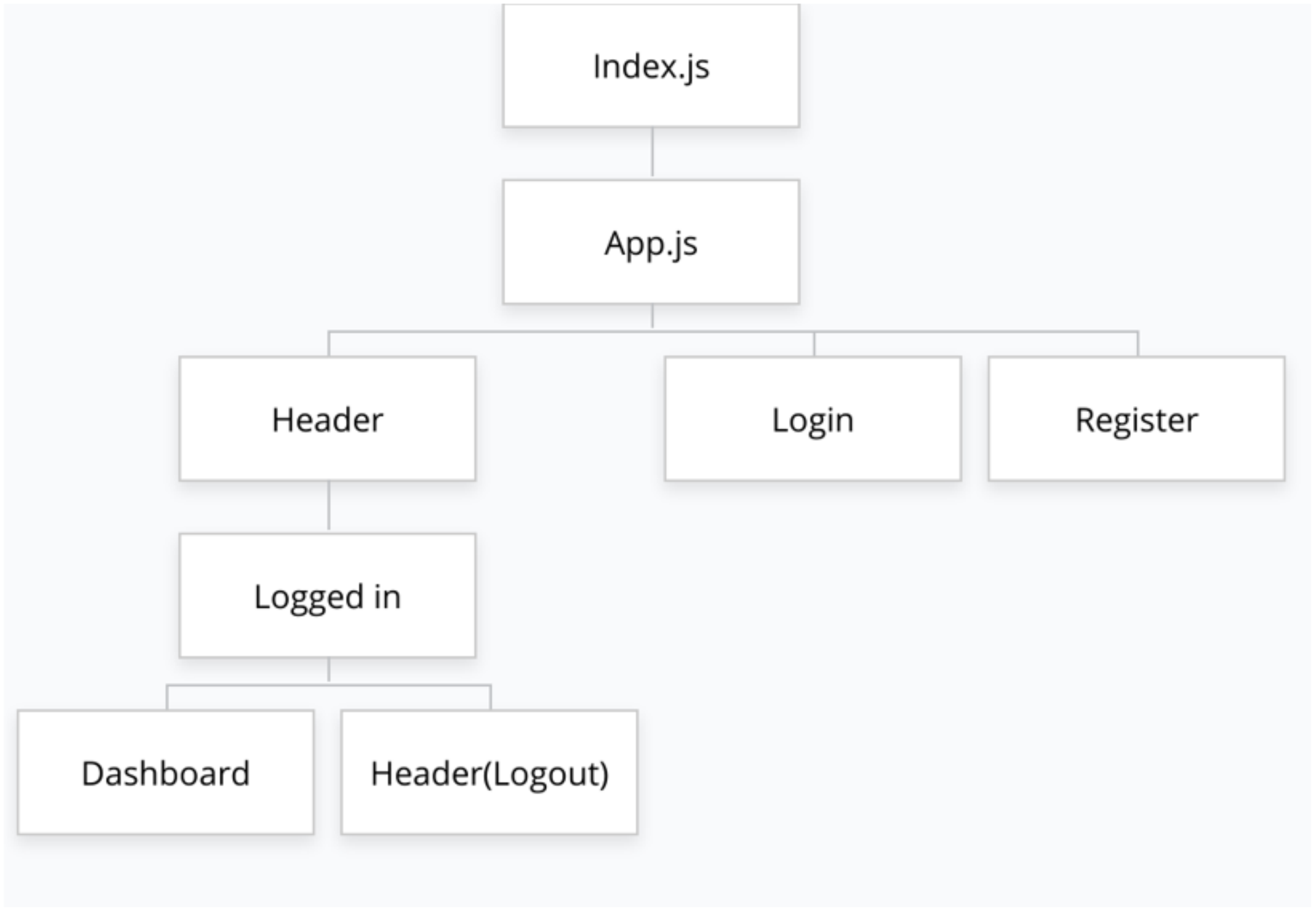
FLUX PROPERTIES

Action: Action describes user interaction that occurred in a component, such as a user clicking a button.
Dispatcher: Dispatcher methods are invoked by an action component. This emits an event with data that needs to go to a store, which is a singleton registry.
Store: The store listens to certain events from the dispatcher. When it listens to an event from the dispatcher, it will then modify its internal data and emit a different event for views.
View: Views are typically a react component. Views get data from the store and set up a listener to refresh itself when the store emits any changed events.

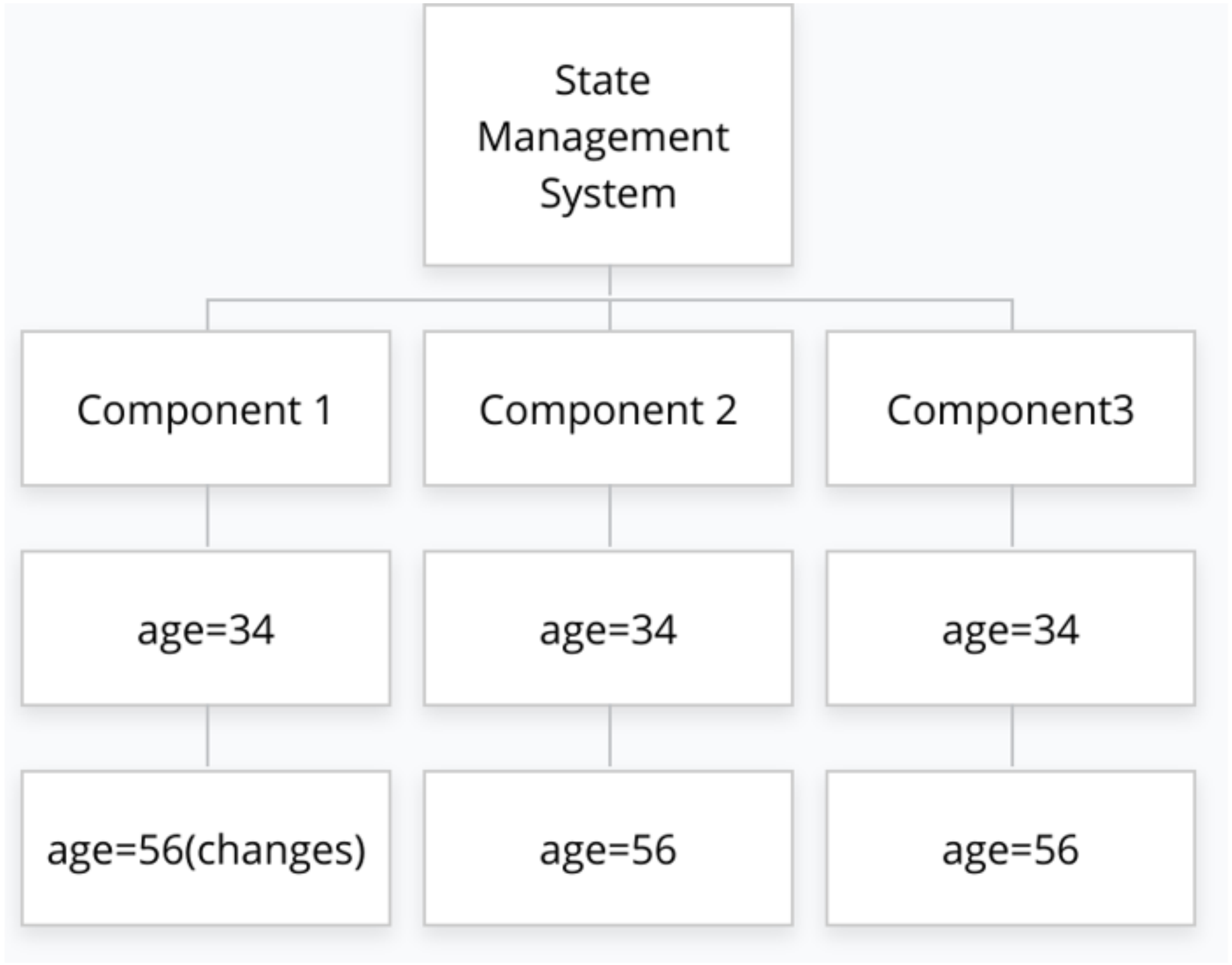
WHAT IS REDUX

Redux is an open-source JavaScript library for managing and centralizing application state. It is most commonly used with libraries such as React or Angular for building user interfaces. Similar to (and inspired by) Facebook's Flux architecture, it was created by Dan Abramov and Andrew Clark.

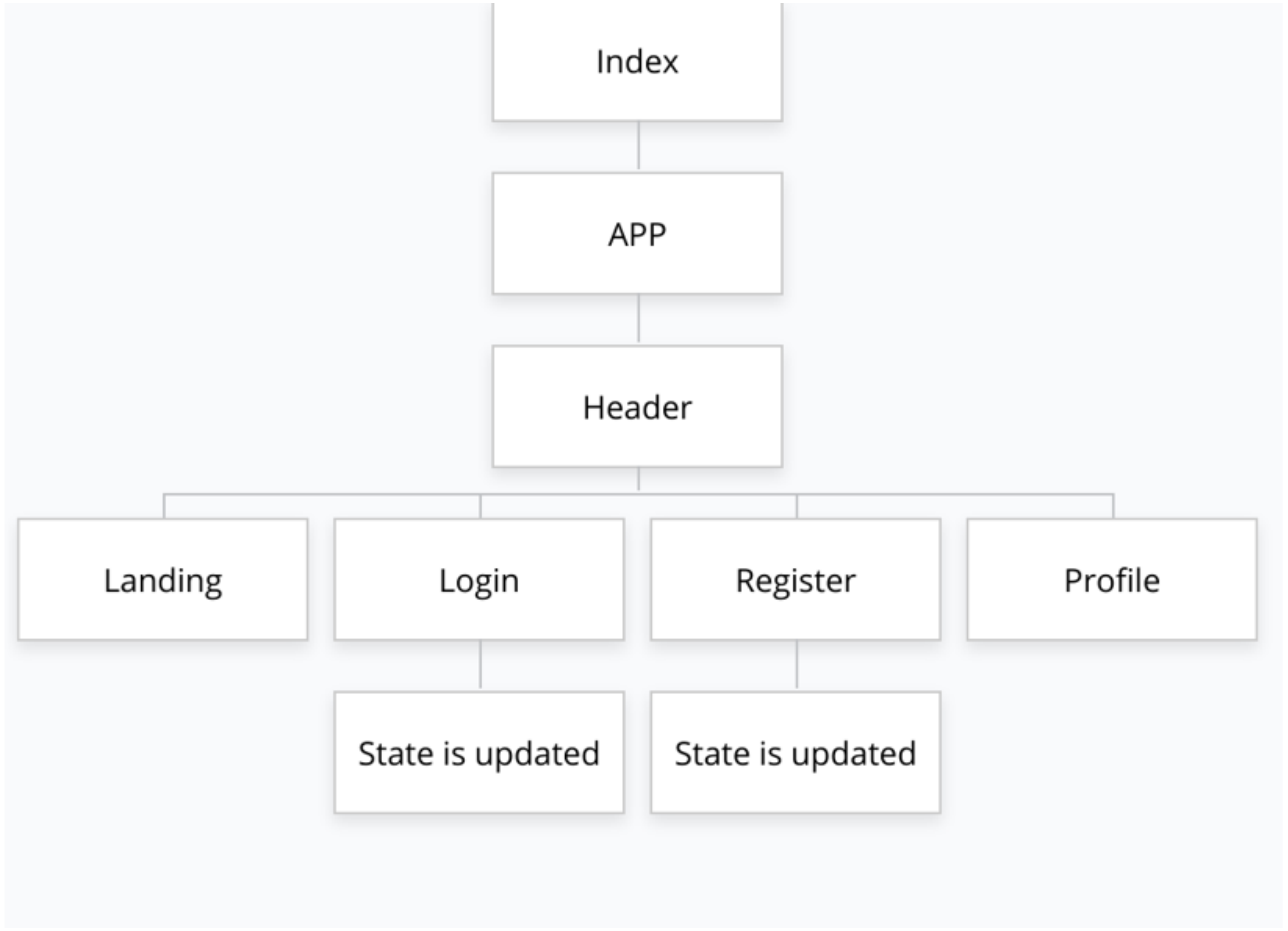
REDUX


Index.js
App.js
Header
Login
Register
Logged in
Dashboard
Header(Logout)

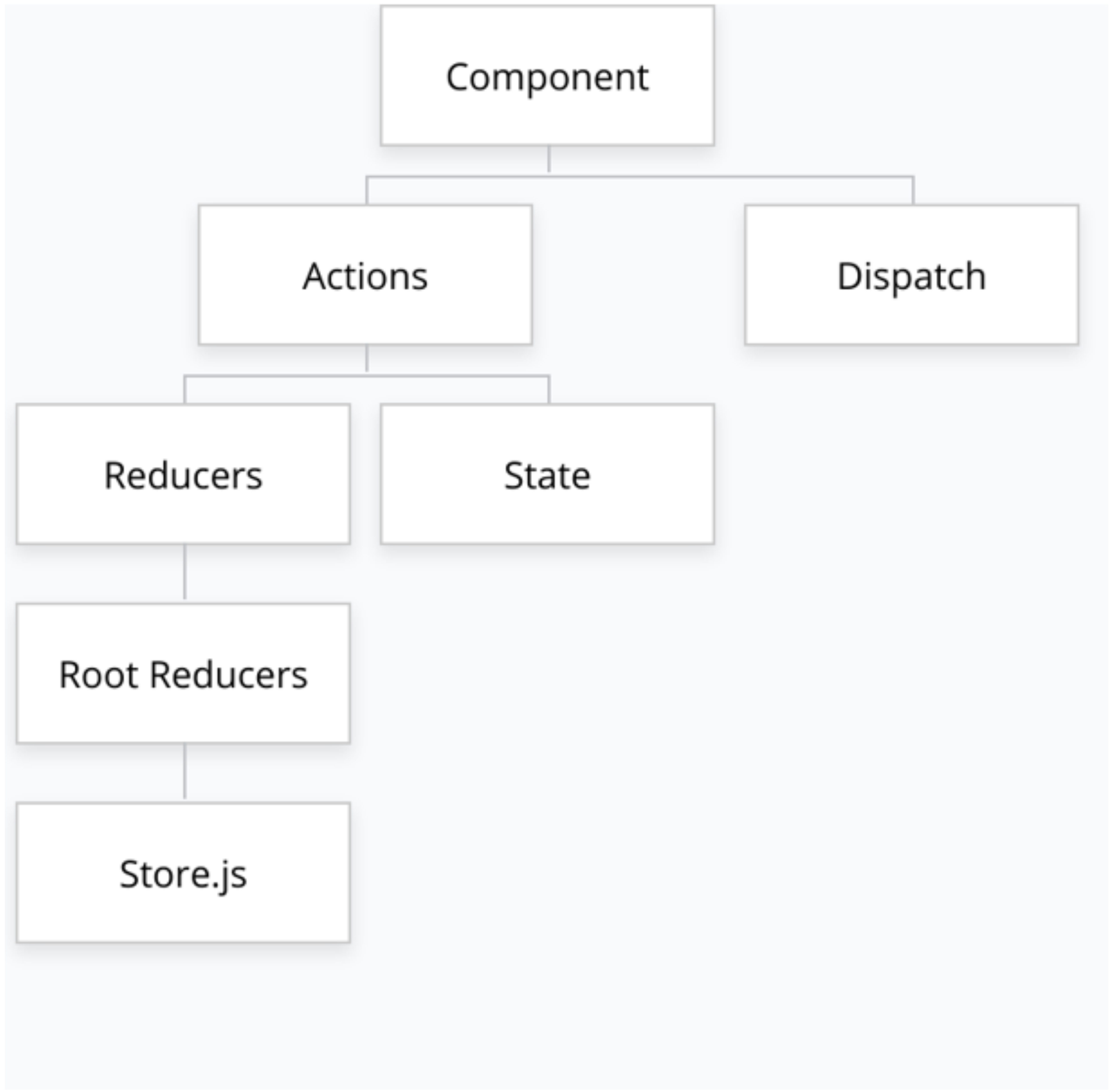
REDUX


State Management System
Component 1
Component 2
Component3
age=34
age=34
age=34
age=56(changes)
age=56
age=56

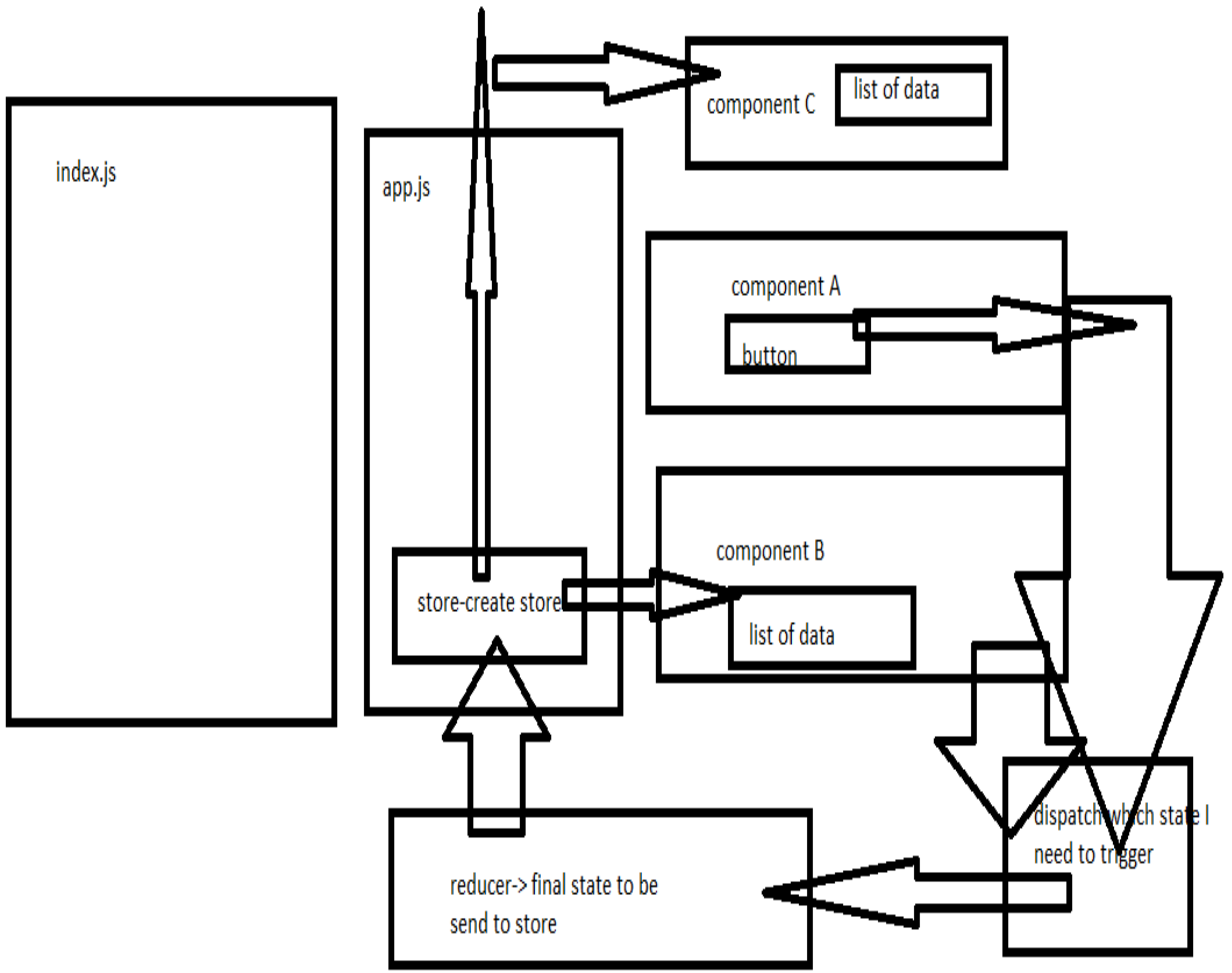
REDUX


Index
APP
Header
Landing
Login
Register
Profile
State is updated
State is updated

REDUX


Component
Actions
Dispatch
Reducers
State
Root Reducers
Store.js

REDUX


index.js
app.js
component C
list of data
component A
button
component B
list of data
store-create store
dispatch which state I need to trigger
reducer-> final state to be send to store

REDUX PACKAGE

Install the following package:

1. npm install redux
2. npm install react-redux

REDUX STEPS

1. Create the store
2. Create the root reducers
3. Create the state reducers
4. Dispatch
5. Access the state