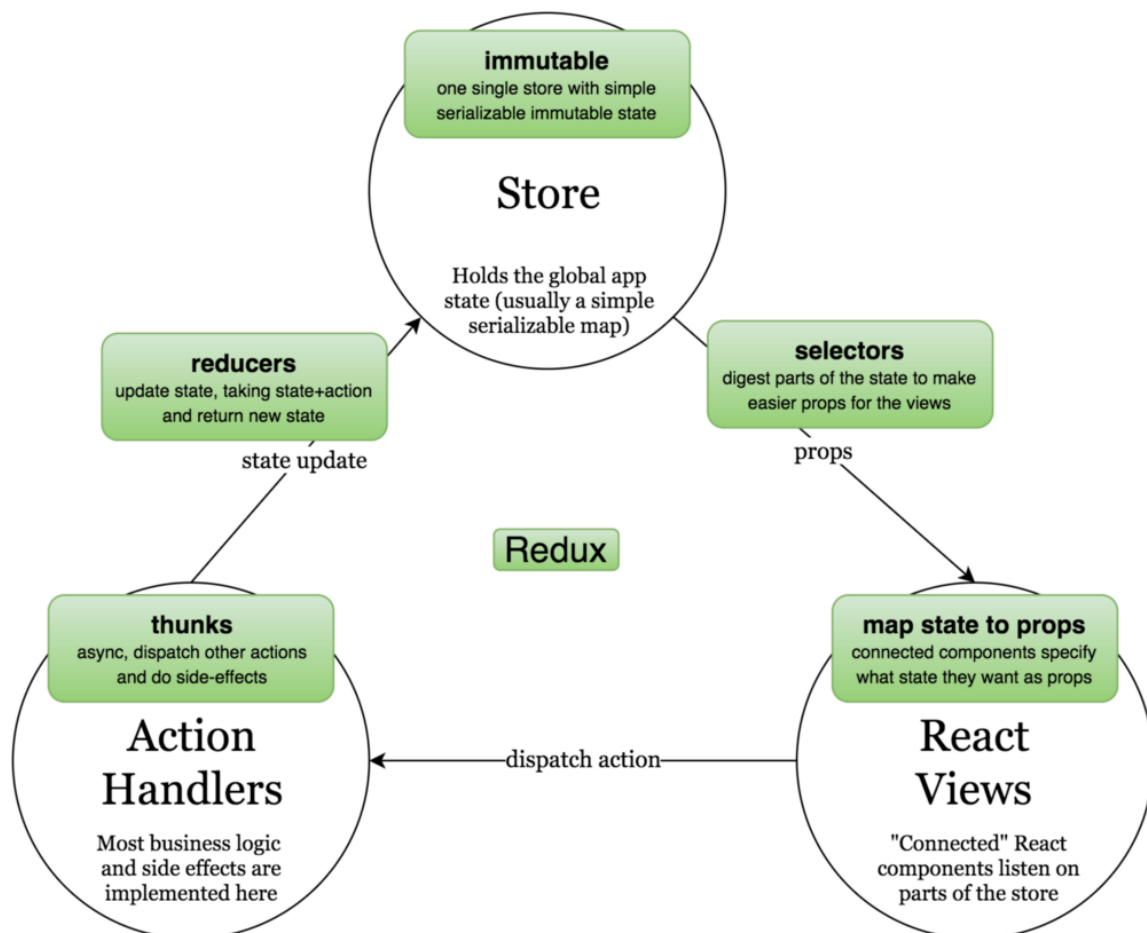
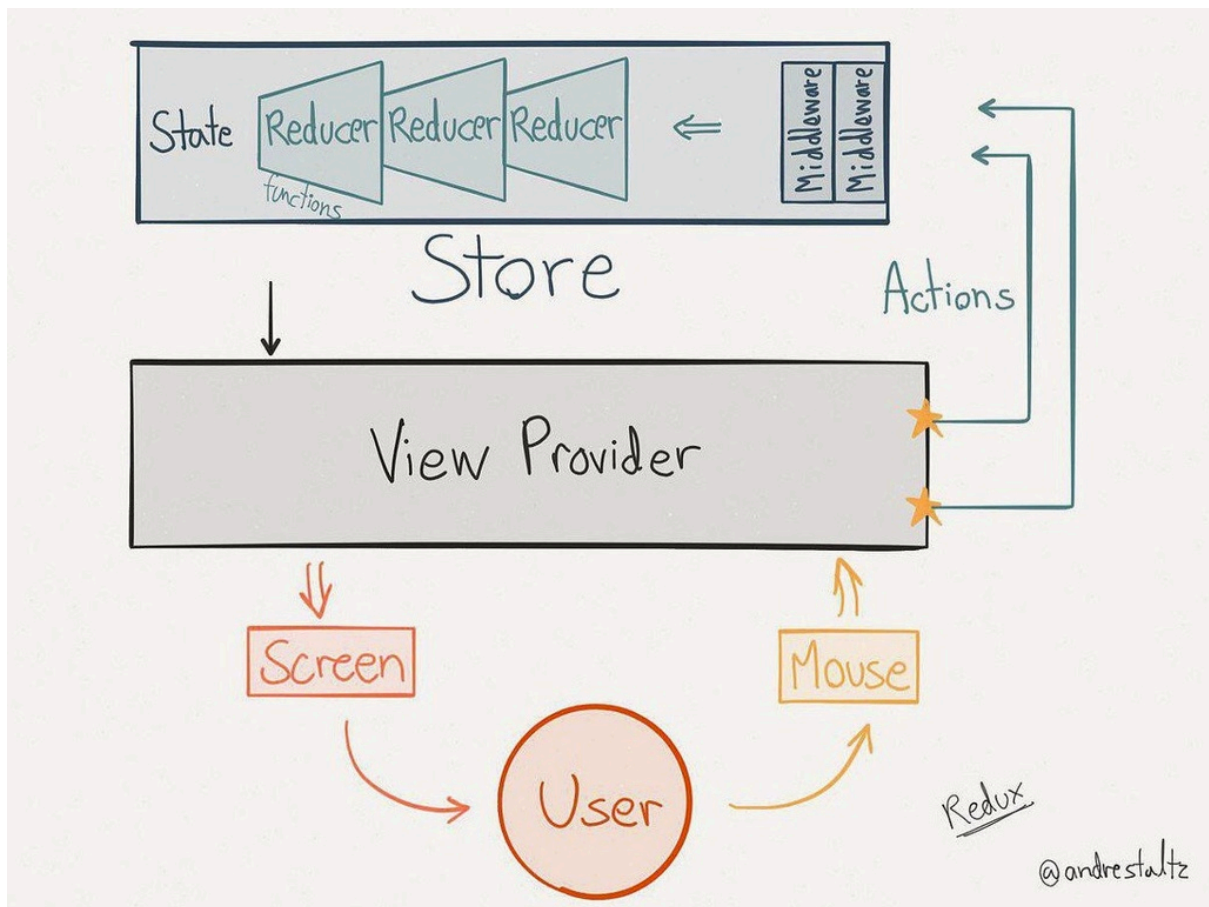


# Redux

The basic idea of Redux is to always make your data flow in one direction in the application (unidirectional flow), from your *Stores* to your (React) *Components*. *Components* call *Action Creators/Handlers*, they dispatch *Actions* that your *Stores* react to.





### STORE:

Stores the STATE of a container (stateful) component (view). The STORE controls the STATE. The single source of truth. The STORE “notifies” all the “listeners” that the STATE has changed hence allowing various parts of the UI to update (render) themselves according to the new STATE.

### REDUCER:

Services that are passed the STATE and an ACTION, and if the REDUCER (service) recognises the ACTION it can return a new STATE to the STORE. REDUCERS know how to mutate the STATE based on the requested action but never changes the STATE. It creates a copy of the STATE, then makes the requested changes to the copy and return the new modified STATE to the STORE.

### MIDDELWARE:

Middelware acts like interceptors for ACTIONS. Middelware can modify, stop or add more ACTIONS before the ACTION reaches the STORE. All the housekeeping tasks.

- Logging
- Authorisation to check whether the user has permission to run the ACTION
- API middelware that sends or receives something to or from the server

- Asynchronous events - AJAX requests
- Websockets operations

**ACTIONS:**

Actions describe the changes we want to make

- ADD\_USER
- ADD\_INGREDIENT
- DELETE\_USER

USER INTERFACE => ACTION => MIDDLEWARE => ACTION => REDUCER = NEW STATE => STORE => NEW STATE => USER INTERFACE

**STATE:**

The STATE should be considered the application's database and be designed separately from the presentation and business logic layers.

There are a few questions to consider when deciding whether to add something to the state:

1. Should this data be persisted across page refresh?
2. Should this data be persisted across route changes?
3. Is this data used in multiple places in the UI?

**STUFF:**

<https://medium.com/@rajaraodv/a-guide-for-building-a-react-redux-crud-app-7fe0b8943d0f>

**Tal Kol from Wix:**

<https://hackernoon.com/avoiding-accidental-complexity-when-structuring-your-app-state-6e6d22ad5e2a>

<https://hackernoon.com/redux-step-by-step-a-simple-and-robust-workflow-for-real-life-apps-1fdf7df46092>

**Dev Tools:**

<https://github.com/zalmoxisus/redux-devtools-extension>

**Reads:**

<http://redux.js.org/docs/advanced/UsageWithReactRouter.html>

<https://github.com/reactjs/react-router-redux>