



Uni-Directional Data Flow for Better State Management

Daniel Labios | Research Project | Redux | Cohort-19

Redux is a Java Script
Library

Designed for Predictable
State management

Created By Andrew Clark
& Dan Abramov in 2015

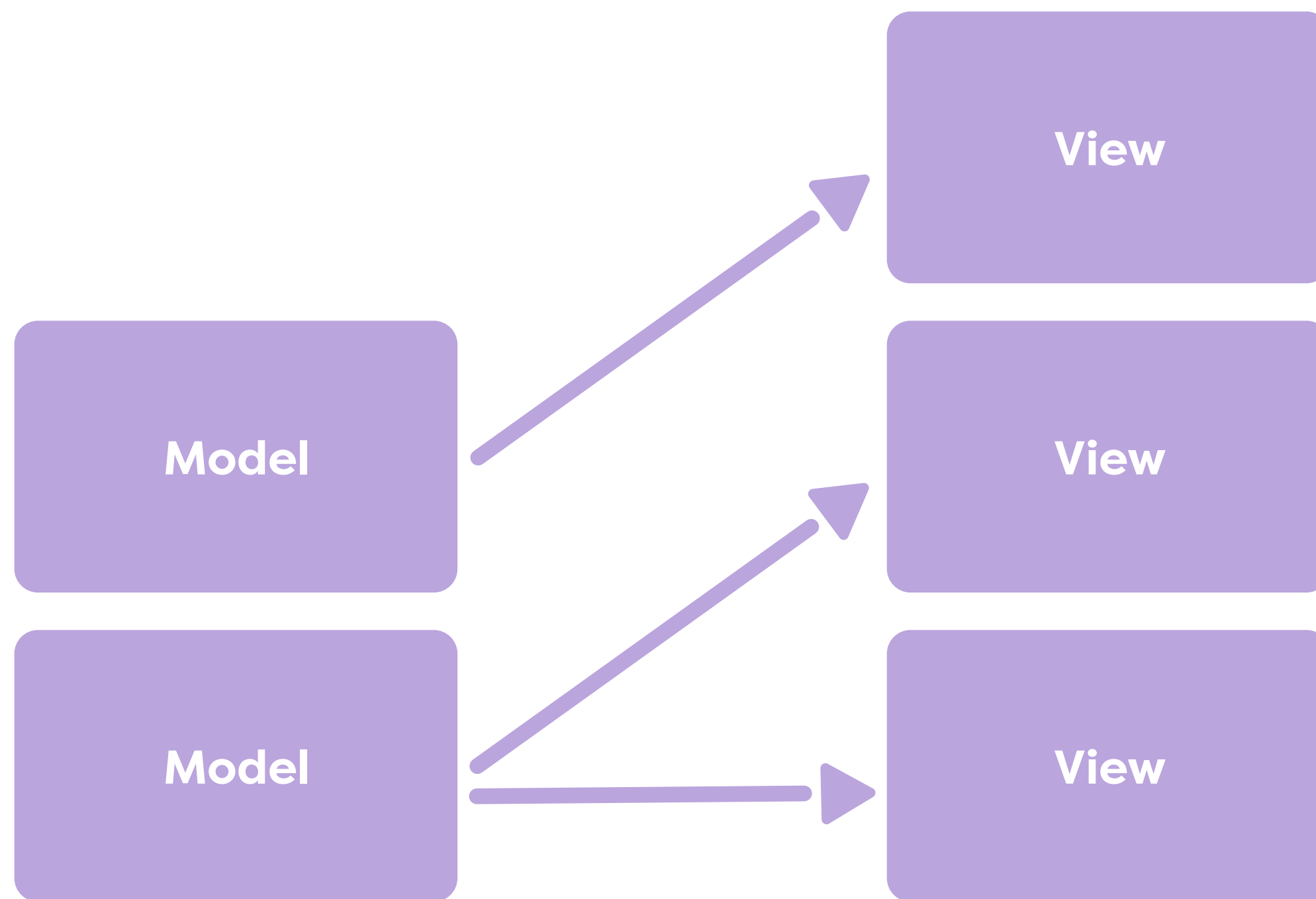
Based on Facebooks Flux
“pattern for Client-Side
Architecture

Most Popular Flux Based
JS Library



Underlying Problem

~~MVC~~ Simplified

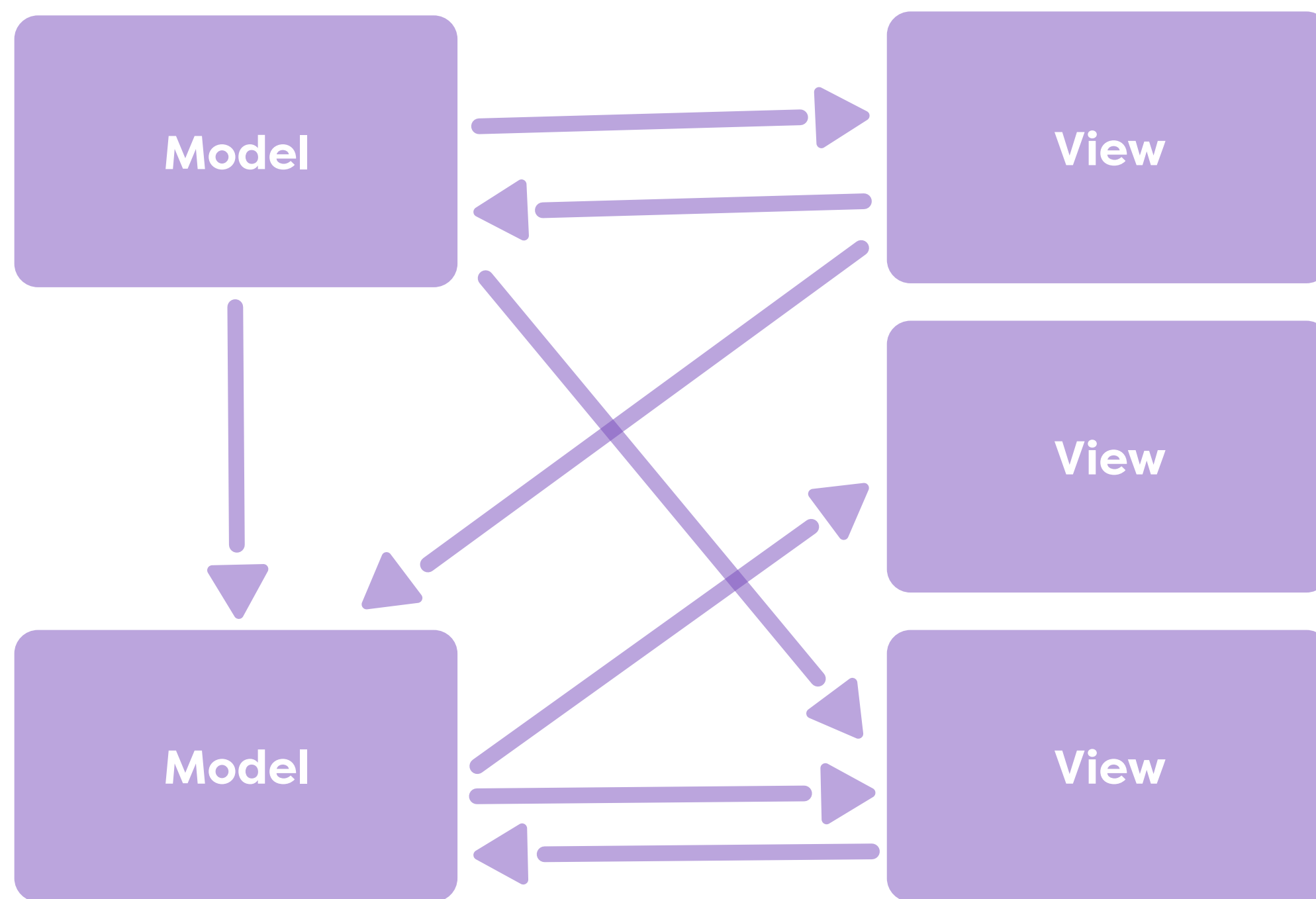


Model - View - Controller

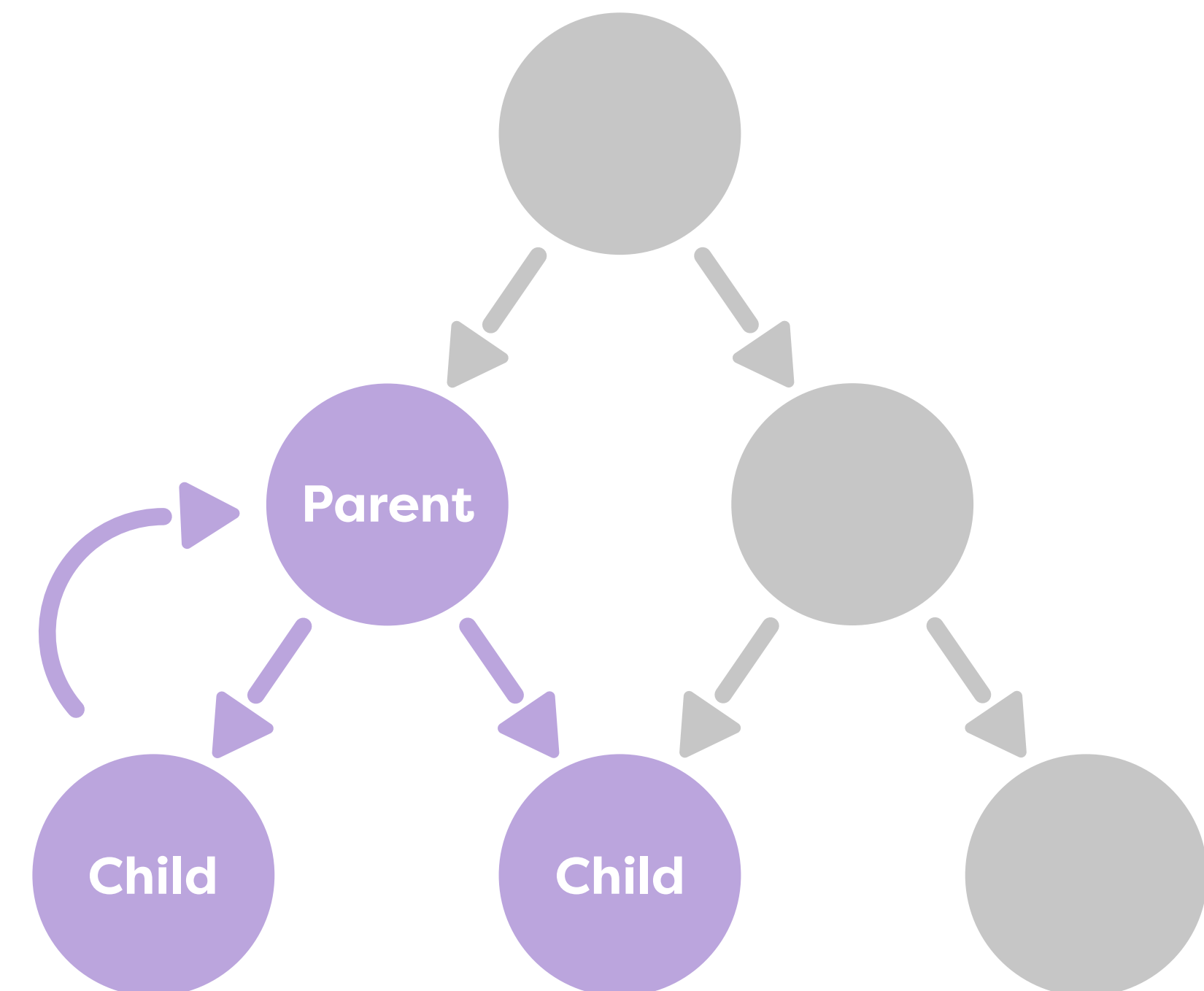
Bi-Directional Data Flow

Underlying Problem

MVC At Larger Scale

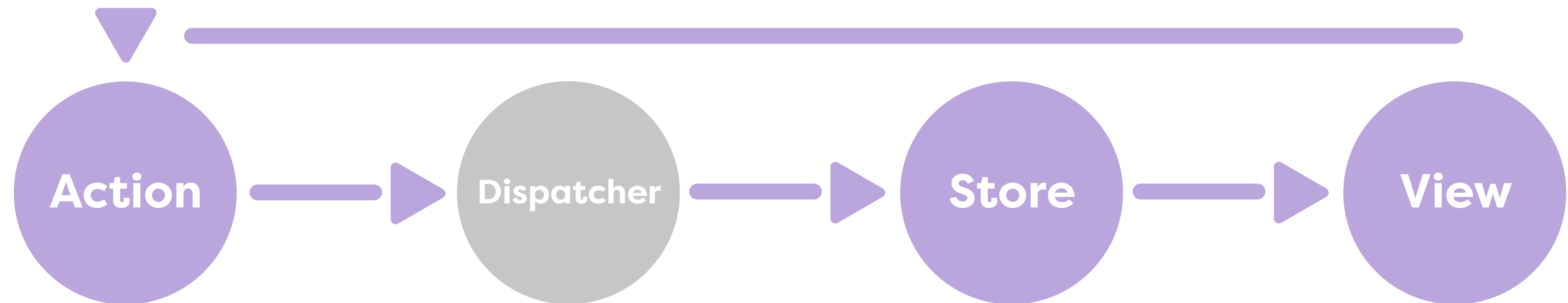


React State Management Issue



Facebook's Solution

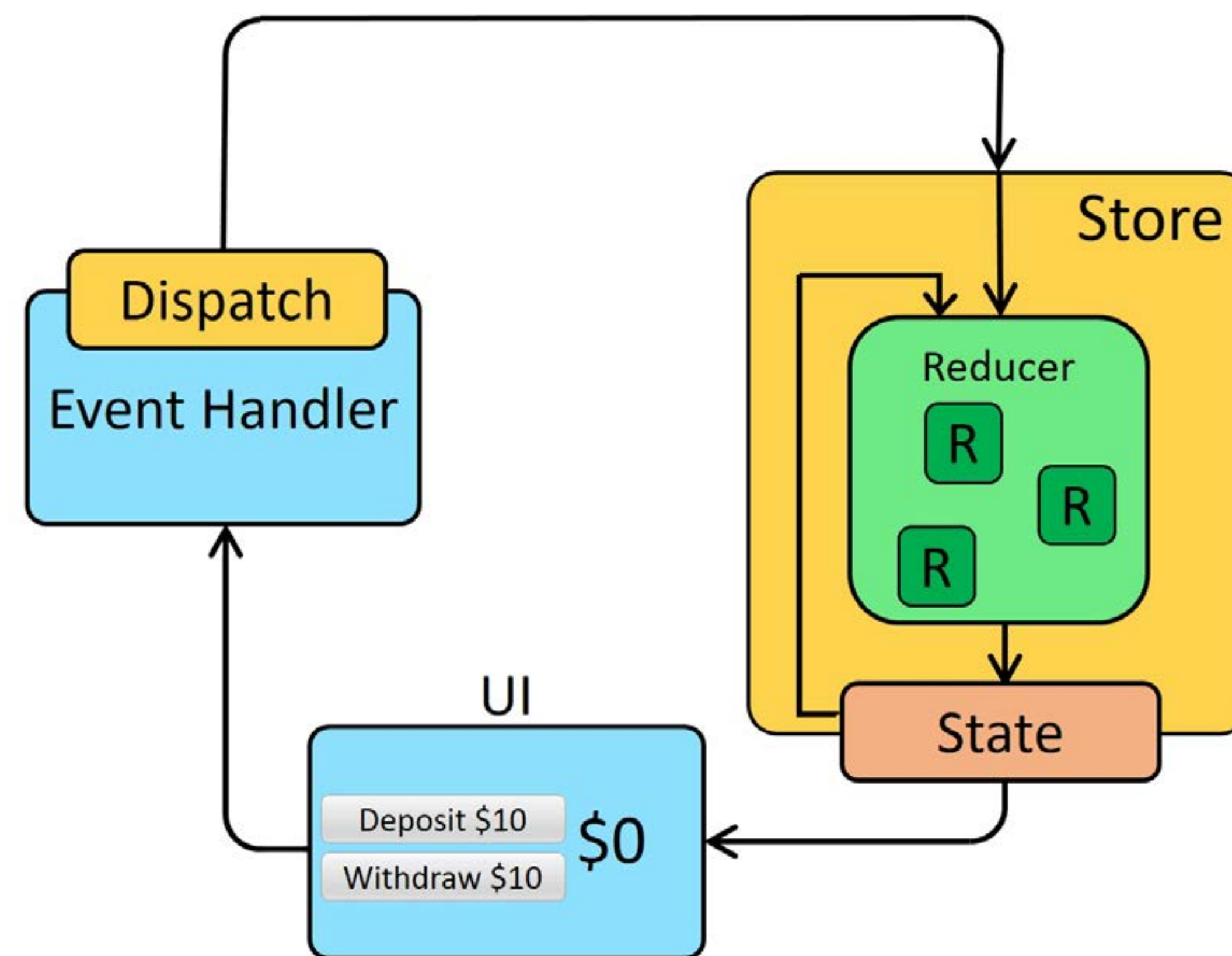
Uni-Directional Data Flow



Found in Facebook's Flux Doc

Redux's Model

Application Data Flow



Found in Redux's Tutorial Documentation

Redux's Model

The Store
Is The Single-Source of Truth

The UI Selects Data in the Store
& Renders the Component State based on it

When an Event is triggered, the UI Dispatches an Action

An Action is a JavaScript Object (POJO) with a *type field*
and a *payload*

The Action is picked up in the Store using Reducers

A Reducer is a function that take an Action (decided by *type field*) & State, figures out how to update the state using the *payload* - if any, then updates the State by mutating it

A Store is divided by Slices. Slices groups together reducers and other logic together for easier management.

Server-Side Connection's are done using MiddleWare called Thunks

Example of an Action Object

```
const addTodoAction = {  
  type: 'todos/todoAdded',  
  payload: 'Buy milk'  
}
```

Store

