

REDUX

A JavaScript state management library

STATE

4

0



MENU



TJ Holowaychuk 🐥
@tjholowaychuk

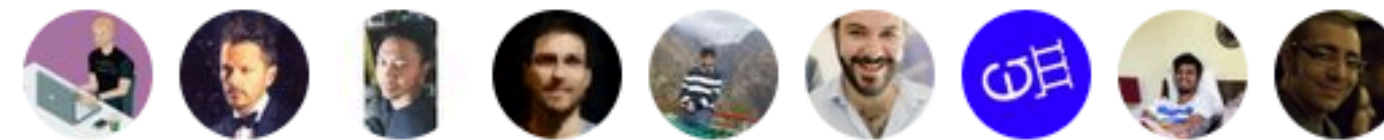
Following



my js state management library: {}

12:51 AM - 29 Jan 2018

289 Retweets 1,431 Likes



44



289



1.4K



4

0

```
{  
  player1Score: 4,  
  player2Score: 0,  
  paddle1Pos: 3,  
  paddle2Pos: 20,  
  ball: [30, 12],  
  ballSpeed: [-5, 3]  
}
```

MENU

Gmail

More

1-13 of 13

COMPOSE

Inbox

Starred

Important

Sent Mail

Drafts (1)

Bananatag Outbox

More

Primary		Social		Promotions		+	
<input type="checkbox"/>			Bananatag	Perfect Timing - Schedule emails right from Gmail with Email Scheduling from Bananatag			
<input type="checkbox"/>			Email Link Clicked	'Bananatag is awesome!' [Andrew Benes] - View details be			
<input type="checkbox"/>			Bananatag Systems	Getting Started with Email Scheduling - Email Scheduling T			
<input type="checkbox"/>			Bananatag Systems	Welcome to Bananatag - Welcome to Bananatag			
<input type="checkbox"/>			Email Opened	'New feature' [andrew@bananatag.com] - View details belc			
<input type="checkbox"/>			Andrew Benes	Fwd: 🎵 Playlists for a year of new possibilities 🎵 - Forward			
<input type="checkbox"/>			Chris Wagner	Sales Contact Form Received - From: Bananatag <info@b			
<input type="checkbox"/>			Google	Google Account password changed - Bananatag Systems			
<input type="checkbox"/>			YouTube	Congratulations on your first YouTube upload! - YouTube h			
<input type="checkbox"/>			YouTube	Welcome to YouTube - YouTube Help Center Change Em			
<input type="checkbox"/>			Gmail Team	Customize Gmail with colors and themes - To spice up you			
<input type="checkbox"/>			Gmail Team	Import your contacts and old email - You can import your c			
<input type="checkbox"/>			Gmail Team	Get Gmail on your mobile phone - Access Gmail on your m			

Bananatag is awesome!

Andrew Benes

Bananatag is awesome!

Try it today at bananatag.c

0 GB (0%) of 15 GB used
Manage

©2015 Google - Te

Bananatag is awesome!

Andrew Benes

Bananatag is awesome!

Try it today at bananatag.com

Send

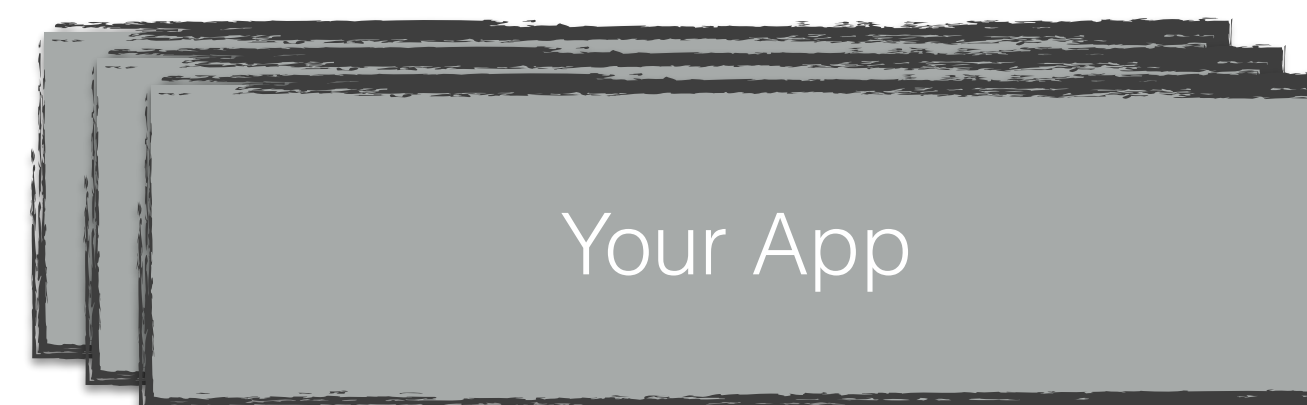
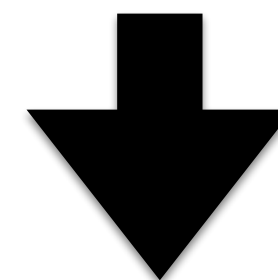


WHAT REDUX IS

- **State Management Library:**
 - A small tool for containing, accessing and affecting a set of information — often called “state”
- If you **do not** have problems with state management, you might find the benefits of Redux harder to understand.

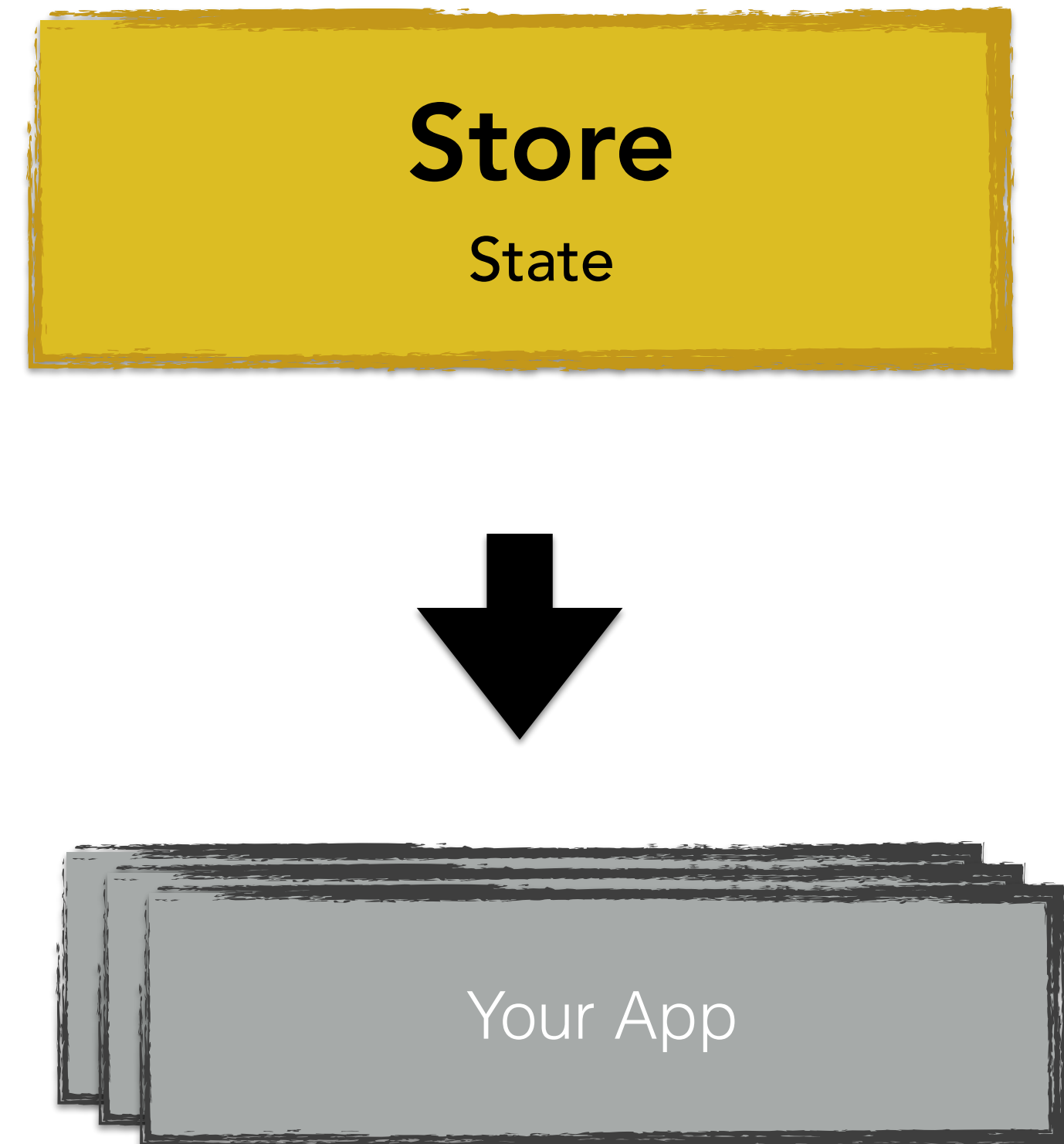
PRINCIPLES

- **Single source of truth**
- **Data is read-only**
- **Changes can be requested through actions and are made with pure functions**



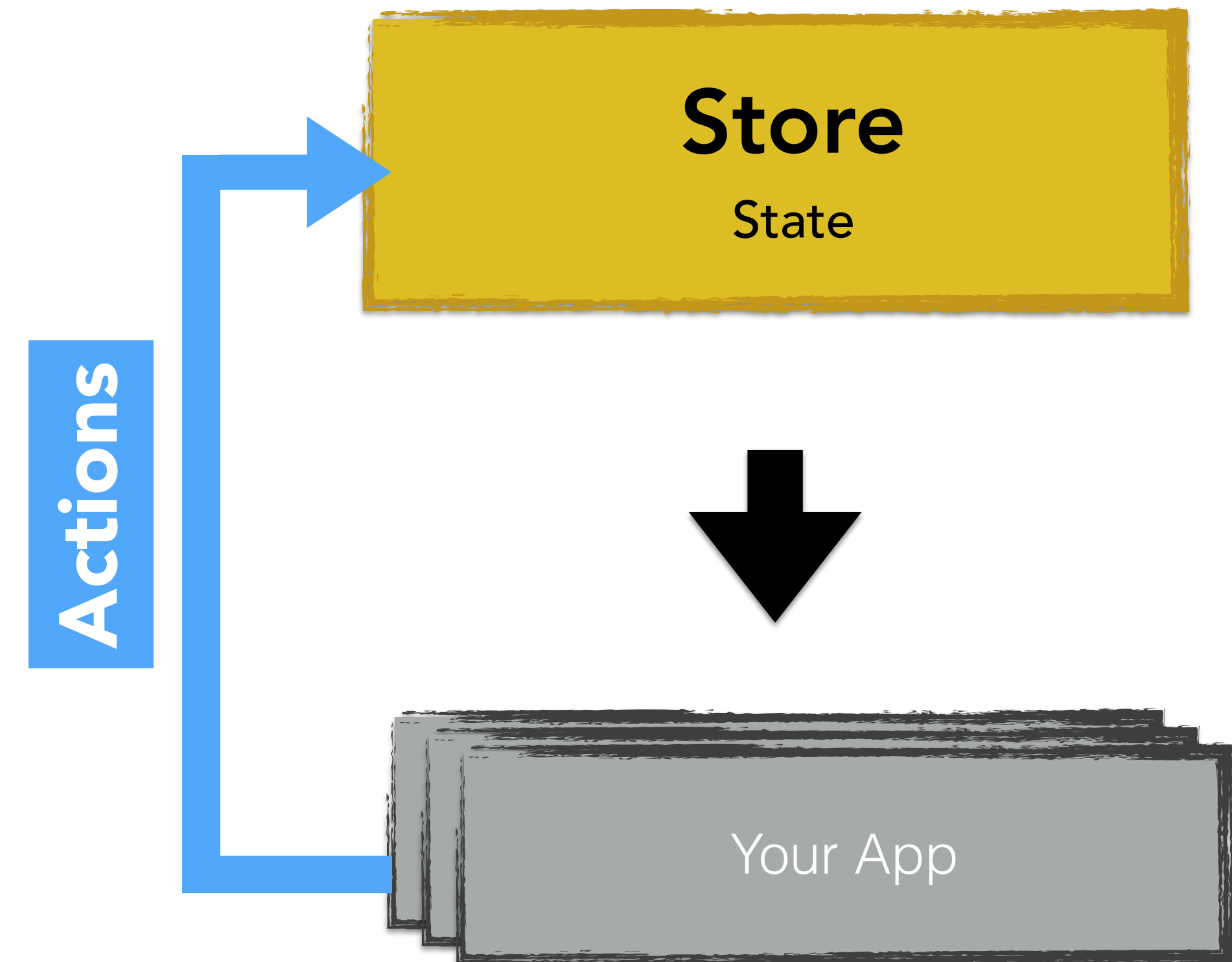
THE STORE

- ◉ The single holder of information
- ◉ **Read-Only:** Provides methods to **access state** & **listen for state changes**
- ◉ Store can receive dispatched signals (actions) meant to affect state.



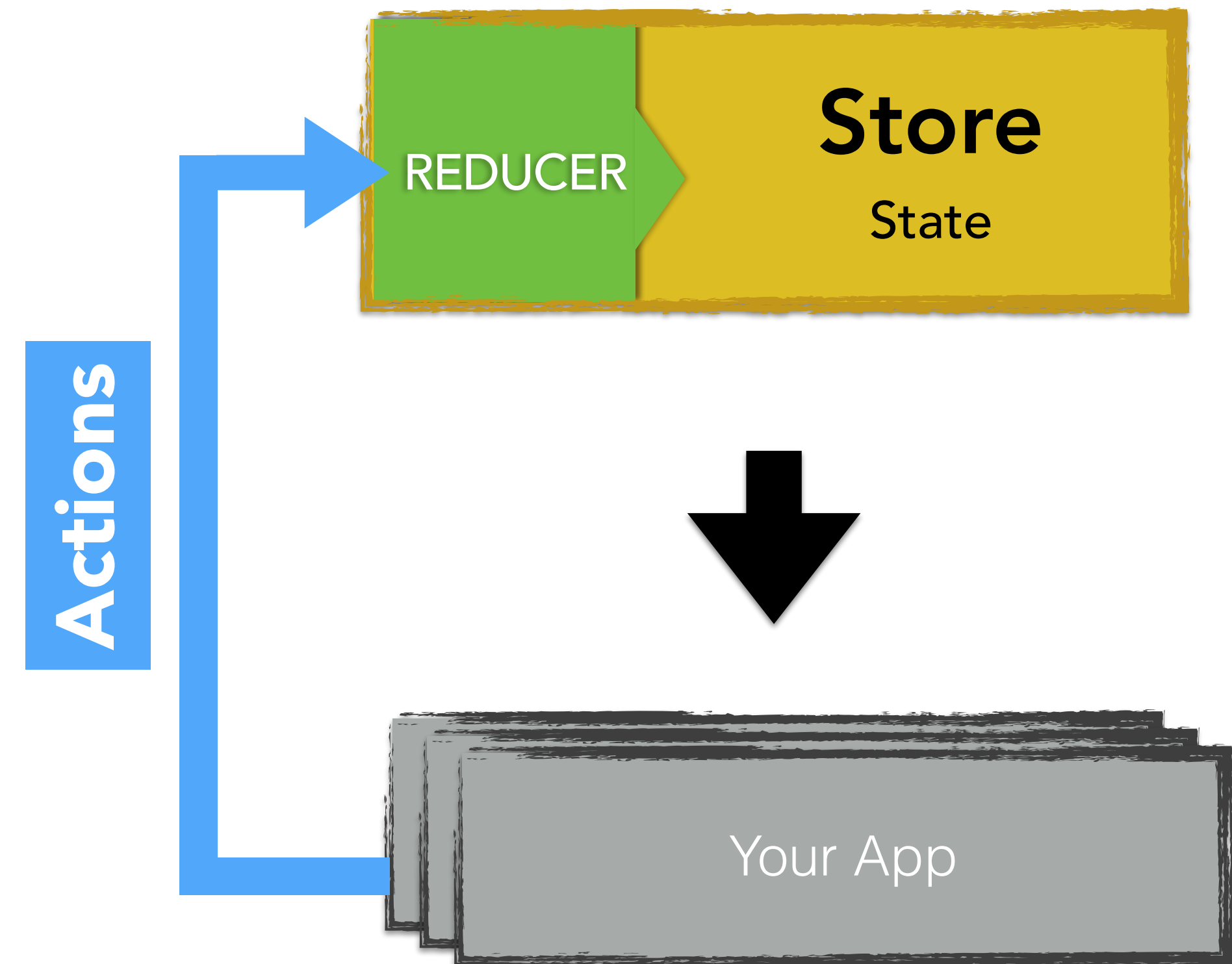
ACTIONS

- Store can receive dispatched signals (actions) meant to affect state.
- Loosely defined as “**things that happen in your app that affect state**”
- Dispatching an action triggers the **reducer** to produce a new state



INSIDE THE STORE: REDUCERS

- Dispatching an action triggers the **reducer** to produce a new state
- Decides: based on this signal (action), what the new state should be.



talk is cheap
show me the
CODE



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

const reducer = (state, action) {
  /* check the action, maybe produce a new state */
  return state;
};

const store = createStore(reducer);
```



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

const reducer = (state = {balance: 0}, action) {
  /* check the action, maybe produce a new state */
  return state;
};

const store = createStore(reducer);
store.getState() // {balance: 0}
```




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

const reducer = (state = {balance: 0}, action) {
  /* check the action, maybe produce a new state */
  return state;
};

const store = createStore(reducer);
store.getState() // {balance: 0}

store.dispatch({type: 'DEPOSIT', amount: 100})
```


Action



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

const reducer = (state = {balance: 0}, action) {
  if(action.type === 'DEPOSIT')
    return {balance: state.balance + action.amount};
  else
    return state
};

const store = createStore(reducer);
store.getState() // {balance: 0}

store.dispatch({type: 'DEPOSIT', amount: 100})

store.getState() // {balance: 100}
```




THE STORE

- Single holder of state
- You can check the state, be notified of changes, but you cannot directly modify the state inside the store.





ACTIONS \cong BANK TRANSACTIONS

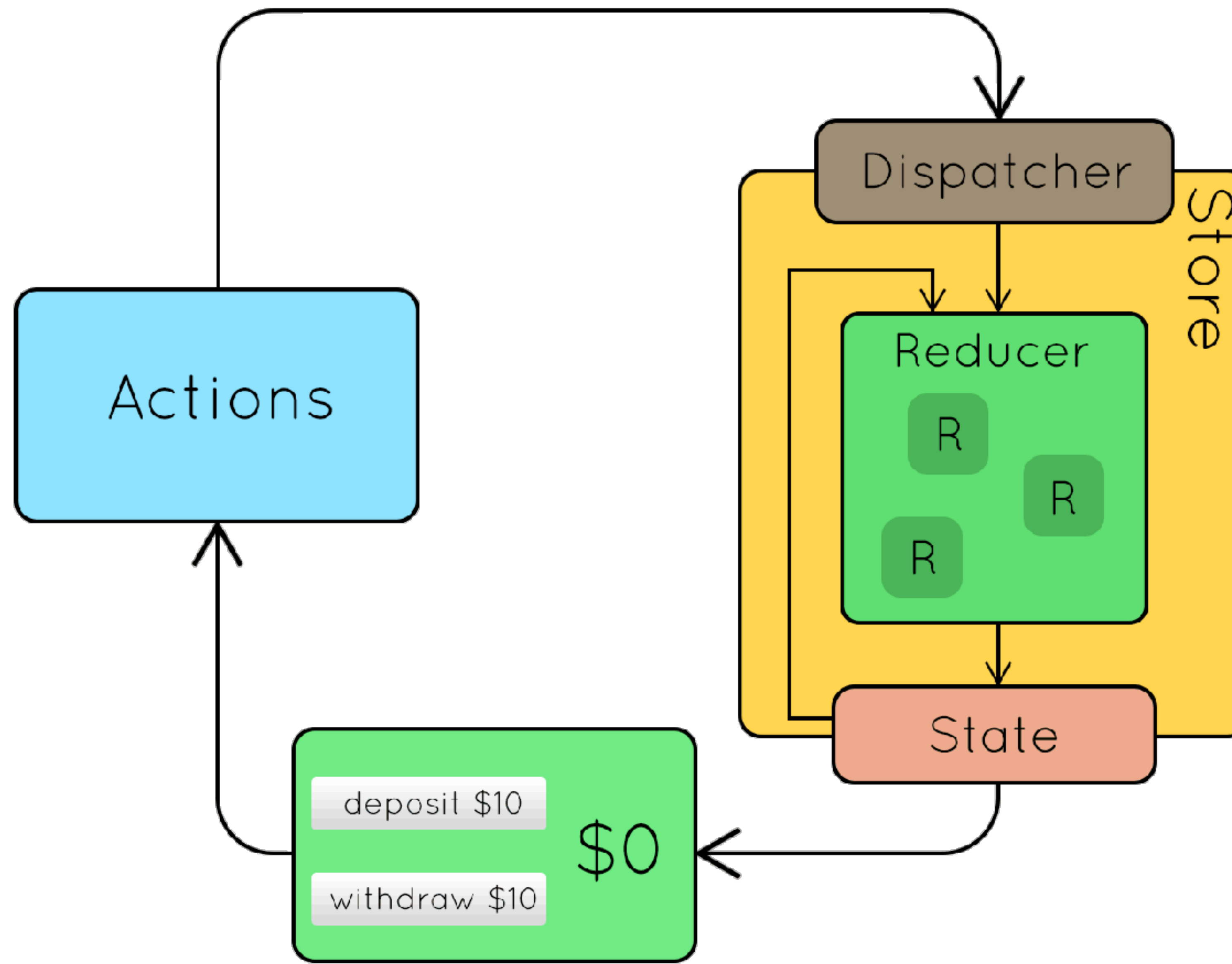
Transaction	Amount	Balance
Create Account	—	\$0.00
Deposit	\$200.00	\$200.00
Withdraw	(\$50.00)	\$150.00
Deposit	\$100.00	\$250.00



REDUCER \cong BANKER



- ◉ **Calculates** your balance
- ◉ Produces a **new** balance (**state**) after every transaction.



Hello
my name is

Reducer

ARRAY.PROTOTYPE.REDUCE



ARRAY SUM

```
const sum = [1, 2, 3].reduce((total, next) => {  
  return total + next;  
});  
  
console.log(sum) // 6
```

**“REDUCE” A COLLECTION OF
VALUES INTO A SINGLE VALUE**

REDUCER IN REDUX: TURN A SERIES OF ACTIONS INTO A SINGLE OUTCOME

Lab