

# Redux

**A single source of truth**

# Redux

- **What is redux**
  - **Global state management**
  - **It makes complex things easy**
- **When should I use it**
  - **You'll know when you need redux, if you aren't sure if you need it, you don't**

**Problem: sharing data  
to distant components**



# Why tho? (react does state management!)

You should use Redux when you have reasonable amounts of data changing over time, you need a single source of truth, and you find that approaches like keeping everything in a top-level React component's state are no longer sufficient.

# Steps to hook up Redux

- You need to do all the steps
- You don't need to go in this order. I think the following order is helpful. The mini project is slightly different, but this order is used by the afternoon project.

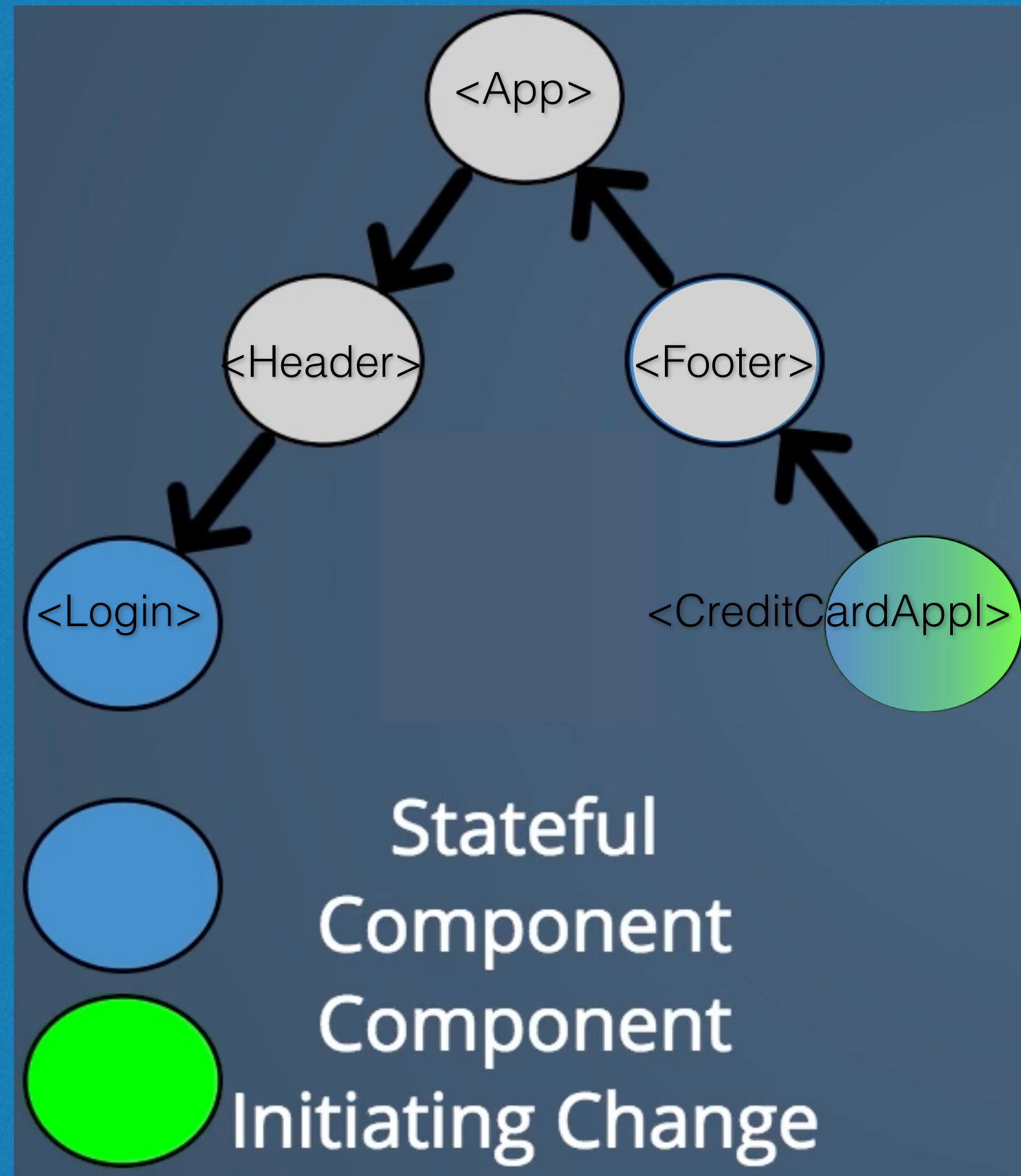
# Steps - Part 1 - Read

- Create store
- Hook app to store with `<Provider>` in `index.js`
- Hook components to store with `connect()` and `mapStateToProps`



# Steps - Part 2 - Write

- Create reducer with actions and action creators
- Dispatch actions with dispatch (recommend to use `mapDispatchToProps`)

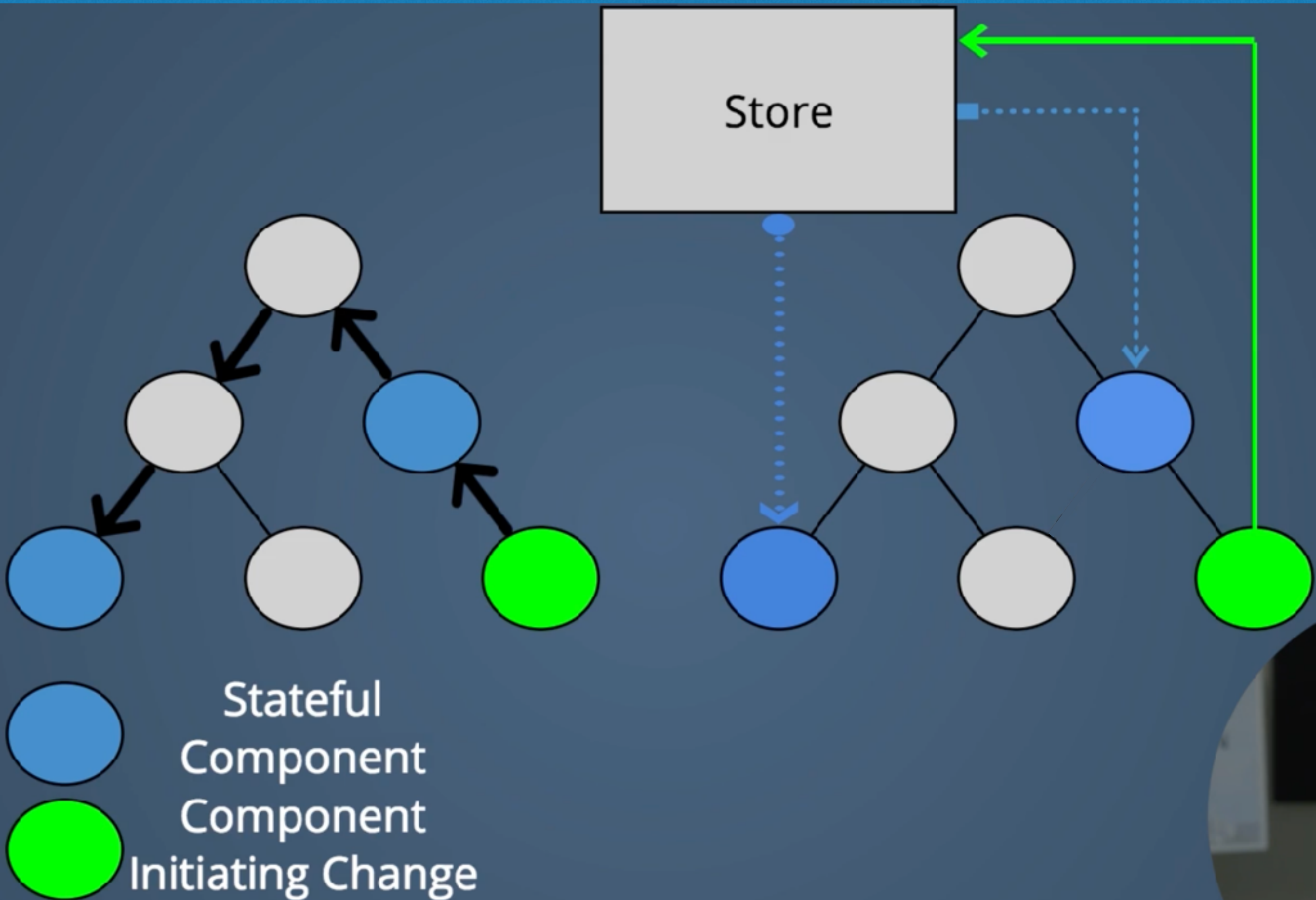


```
<App>
  <Header>
    <Login />
  </Header>
  <Footer>
    <CreditCardAppl />
  </Footer>
</App>
```



# Unidirectional Data Flow

- This means that all data follow the same lifecycle pattern
  - Call an action
  - Redux store calls reducer function you gave it
  - Redux store saves the complete state tree returned by the reducer



- App
- Header
- Login info
- Footer
- Credit card application

Log In or:

Welcome [name] | (00 hidden)

## Welcome

Sign in for the best experience

Not on Amazon? [Start here](#)

## The dress shop



[Shop women's dresses](#)

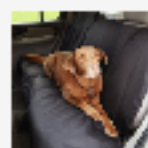
## AmazonBasics



Home



Bluetooth



Pets



Computer

[Explore all AmazonBasics](#)

**AMAZON ORIGINAL**  
**sneaky pete**

Included with Prime Video

Ad feedback

## Deal of the day

**\$41.25**

1 for \$49.99 (21% off)



[Shop all deals](#)

[Back to top](#)

### Get to Know Us

[Company](#)  
[About Amazon](#)  
[Investor Relations](#)  
[Amazon Devices](#)

### Make Money with Us

[Sell on Amazon](#)  
[Sell Your Services on Amazon](#)  
[Sell on Amazon Business](#)  
[Sell Your Apps on Amazon](#)  
[Become an Affiliate](#)  
[Advertise Your Products](#)  
[Self-Publish with Us](#)  
[Become an Amazon Vendor](#)  
[Sell Your Subscription on Amazon](#)  
[> See all](#)

### Amazon Payment Products

[Amazon Rewards Visa Signature Cards](#)

Hi [name], apply now!

### Let Us Help You

[Your Account](#)  
[Your Orders](#)  
[Shipping Rates & Policies](#)  
[Amazon Prime](#)  
[Returns & Replacements](#)  
[Manage Your Content and Devices](#)  
[Amazon Assistant](#)  
[Help](#)



# Common Vocab

- **Action** - An action is a plain object that represents an intention to change the state. Actions are the only way to get data into the store.
- **Reducer** - A reducer (also called a reducing function) is a function that accepts an accumulation and a value and returns a new accumulation. They are used to reduce a collection of values down to a single value.
- **Store** - A store is an object that holds the application's state tree.
- **Action Creator** - An action creator is, quite simply, a function that creates an action.
- **Dispatch Function** - a function that accepts an action it then dispatches the action to the store.