

# Monorepo: React Web & React Native

Eugene Zharkov



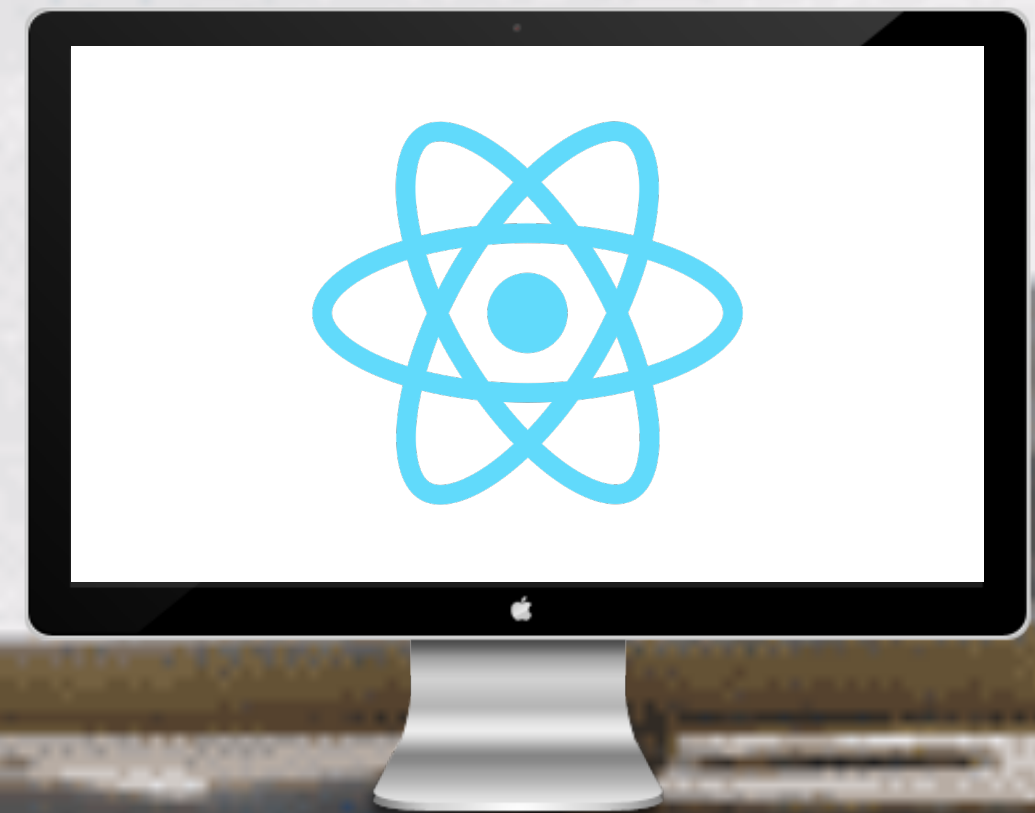
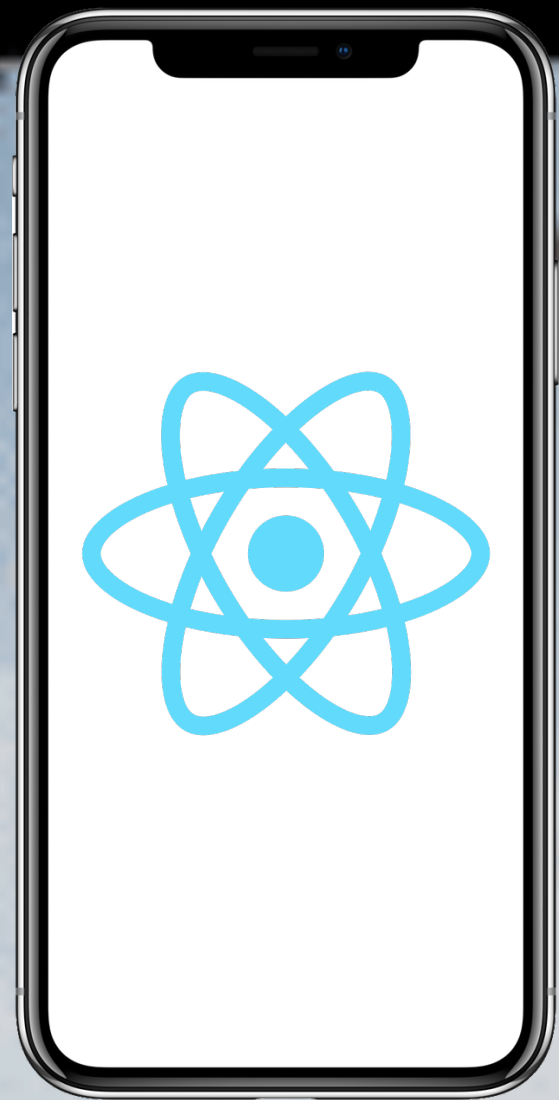
# Eugene Zharkov

React Native

Frontend, Full Stack (sometimes)

 @2j2e





# React Native for Web





**We're done, thank you**



Browser

React





Browser  
implementation

TC39

Browser

React

any other bug

babel





**Browser**

**React**

**React Native for Web**

**React Native**

**Native Platform**





**WHEN PEOPLE ASK ME**



**HOW LIFE IS GOING**



# What to expect

- React Native platform bugs, controls and behavior issues
- React Native for Web bugs, limits, interpretation
- 3rd party components and markup control
- Styling
- Hard to integrate with existed web solutions.



**#2** Extract **shared** to a separate  
package



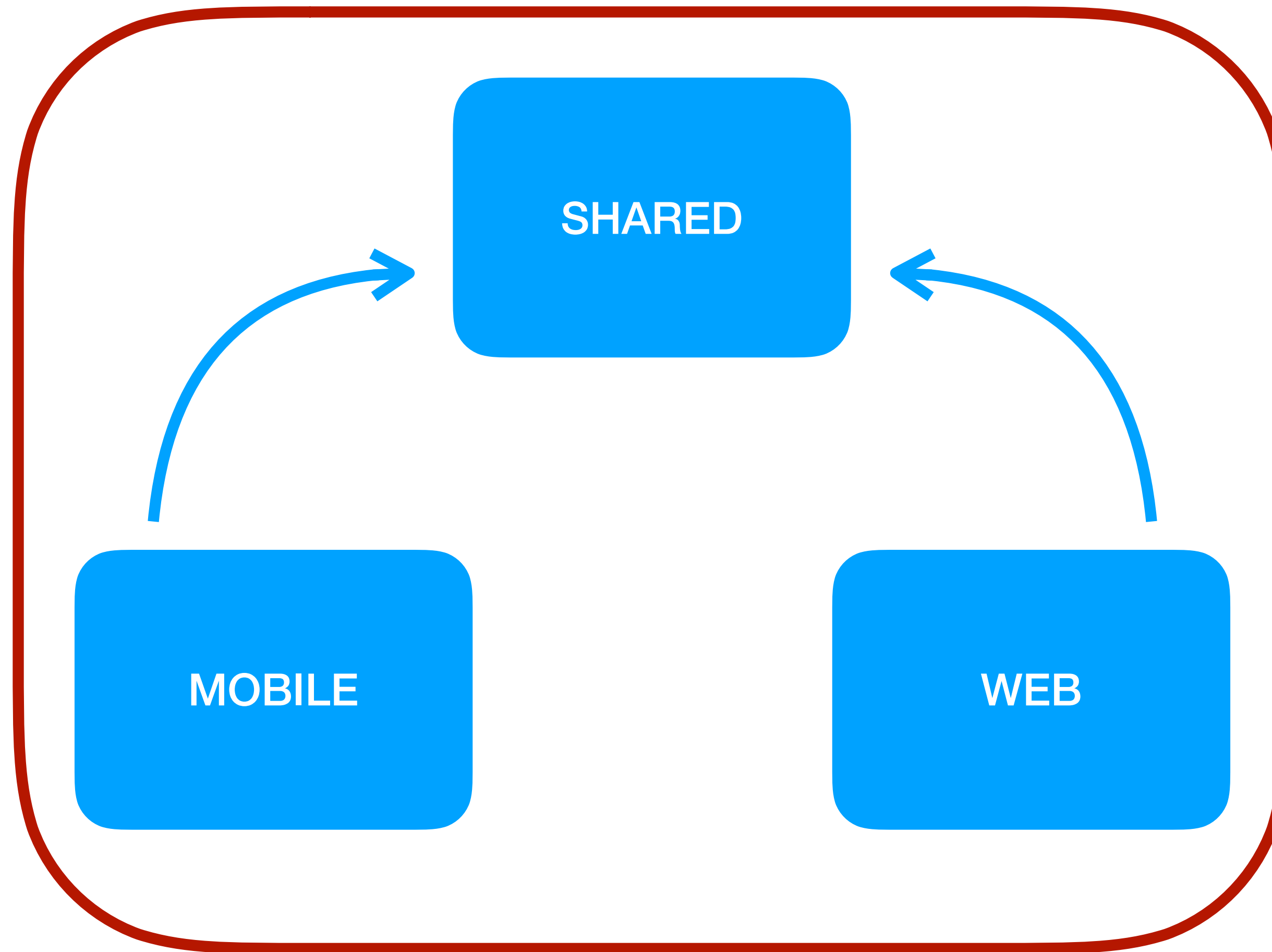
# Issue #0

You need to share a code between web and mobile.





# Monorepo



**npm link**




# npm link

```
{  
  "name": "mobile",  
  "dependencies": {  
    "bar": "file:../core"  
  }  
}
```

# Issue #1


Metro bundler doesn't support symlinks

 facebook / metro

[Code](#) [Issues 112](#) [Pull requests 12](#) [Projects 0](#) [Insights](#)

## Follow symlinks? #1

Open jeanlauliac opened this issue on Feb 27, 2017 · 155 comments

 jeanlauliac commented on Feb 27, 2017 • edited ▾

Do you want to request a *feature* or report a *bug*?

Feature

What is the current behavior?

I'm removing a test from `DependencyGraph-test` that looked like so, but first I want to open with the `6` module.



## Issue #2

Build - OK

Development - NOPE

```
{  
  "name": "mobile",  
  "dependencies": {  
    "bar": "file:../core"  
  }  
}
```

# WML by WIX

Copies changed files from  
source to destination folders

```
npm install -g wml
```

```
wml add ~/core ~/mobile/node_modules/core
```

```
wml start
```



# Issue #3

Last update 2 years ago

Unstable

```
npm install -g wml
```

```
wml add ~/core ~/mobile/node_modules/core
```

```
wml start
```





# #3 Metro Bundler



# Example

```
const path = require('path');

const currentDir = __dirname;

const config = {
  watchFolders: [
    path.resolve(currentDir, '..'),
  ],

  resolver: {
    extraNodeModules: {
      react: path.resolve(currentDir, 'node_modules/react'),
      'react-native': path.resolve(currentDir, 'node_modules/react-native'),
      .....
    },
  },
};
```

```
watchFolders: [  
  path.resolve(currentDir, ' .. '),  
],
```

Specify any additional (to projectRoot) watch folders, this is used to know which files to watch.



```
resolver: {
  extraNodeModules: {
    react: path.resolve(currentDir, 'node_modules/react'),
    'react-native': path.resolve(currentDir, 'node_modules/react-native'),
    .....
  },
}
```

Which other node\_modules to include besides the ones relative to the project directory. This is keyed by dependency name.

# Issue #4

Dependency hell

## LEVELS OF HELL:

Limbo

Lust

Gluttony

Greed

Anger

Heresy

Violence

Fraud

Treachery

Dependencies



## **#4 Lerna**

Tool for managing JavaScript projects with multiple packages.

# Lerna

→ mono **lerna init**

lerna **notice cli** v3.14.1

lerna **info** Initializing Git repository

lerna **info** Creating package.json

lerna **info** Creating lerna.json

lerna **info** Creating packages directory

lerna **success** Initialized Lerna files

→ mono git:(master) **ls**

lerna.json package.json packages

# lerna.json

```
{  
  "packages": [  
    "packages/*"  
  ],  
  "version": "0.0.0"  
}
```



# lerna.json

```
{  
  "packages": [  
    "packages/*"  
  ],  
  "version": "0.0.0",  
  "npmClient": "yarn"  
}
```

# Lerna

**lerna** **create** **core** --yes

**lerna** **create** **web** --yes

**lerna** **create** **mobile** --yes

# Manage dependencies

```
lerna add date-fns —scope=@prefix/shared
```

...OR...

```
lerna add @prefix/shared
```



# **#5 Web vs Mobile**

# Issue #4

Core code should be platform  
agnostic

```
// component
import addToCart from '@core'

class Product extends PureComponent {
  render() {
    const { product } = this.state
    return (
      <div>
        <button onClick={() => {
          this.props.dispatch(addToCart(product))
        }}>Add to Cart</button>
      </div>
    )
  }
}

export default connect()(Product)

// feature
const addToCart = (newItem) => {
  // Check logic ...
  // Business logic ...
  cart.add(newItem)
  // Web related code
  localStorage.setItem('cart', cart)
}
```

```
const addToCart = (newItem) => {  
  // Check logic ...  
  // Business logic ...  
  cart.add(newItem)  
  // Web related code  
  localStorage.setItem('cart', cart)  
}
```





```
const addToCart = (newItem) => {  
  // Check logic ...  
  // Business logic ...  
  cart.add(newItem)  
  // Web related code  
  localStorage.setItem('cart', cart)  
}
```

Web

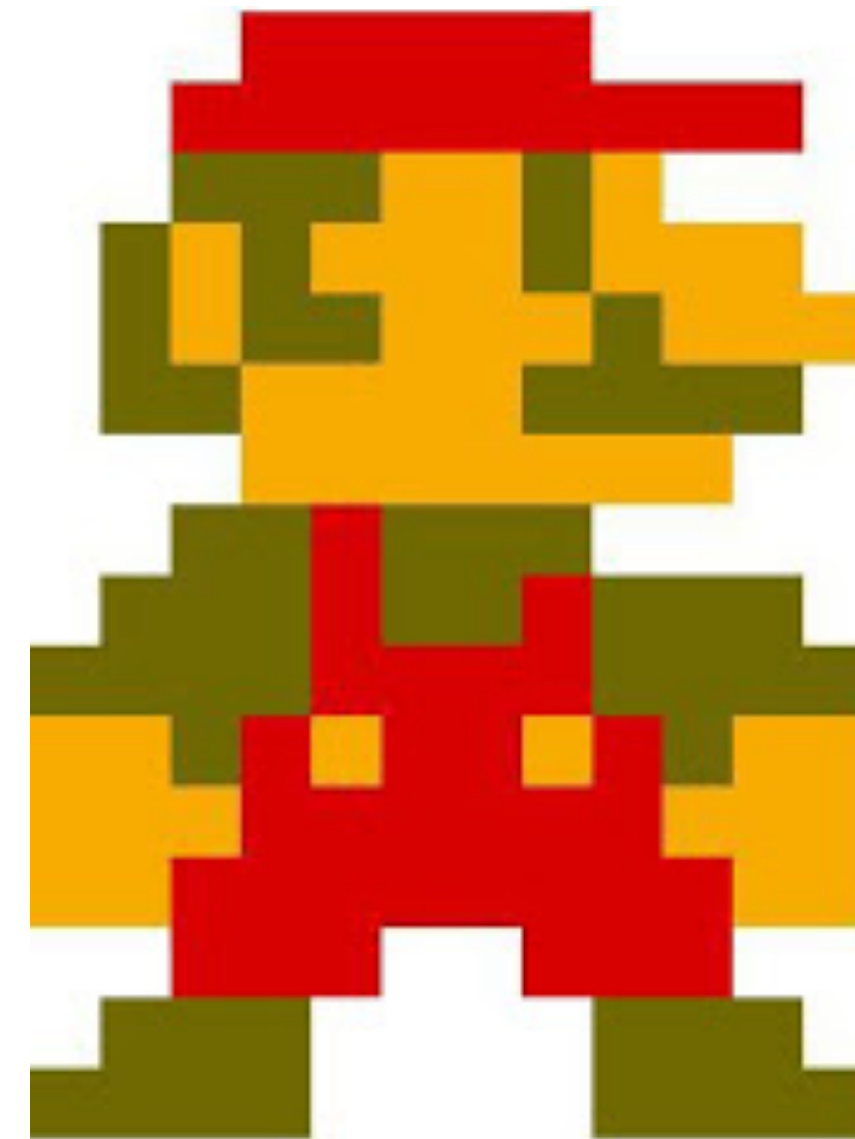
```
import { AsyncStorage } from 'react-native';  
  
await AsyncStorage.setItem('cart', cart);
```

React  
Native

# Issue #5

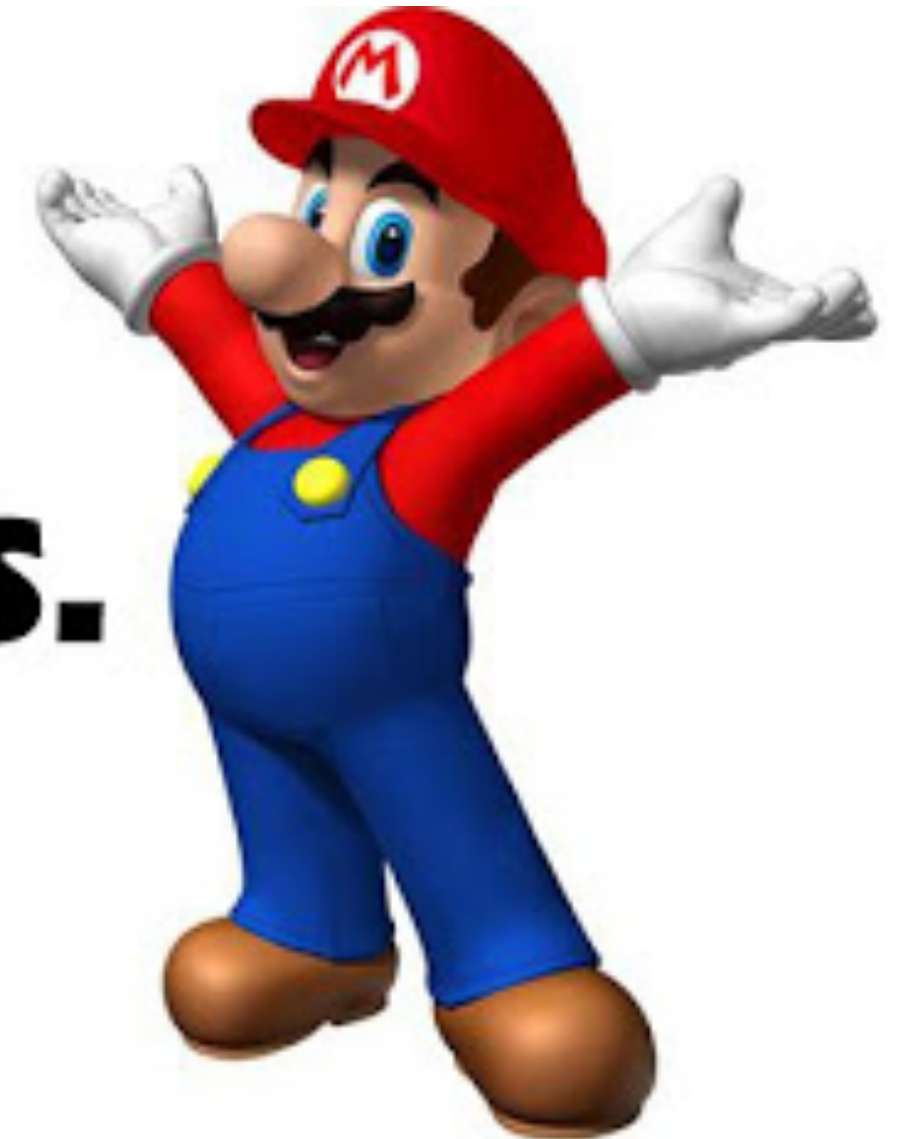
Web and Mobile application  
should behave similar

React Web



React Native

**VS.**

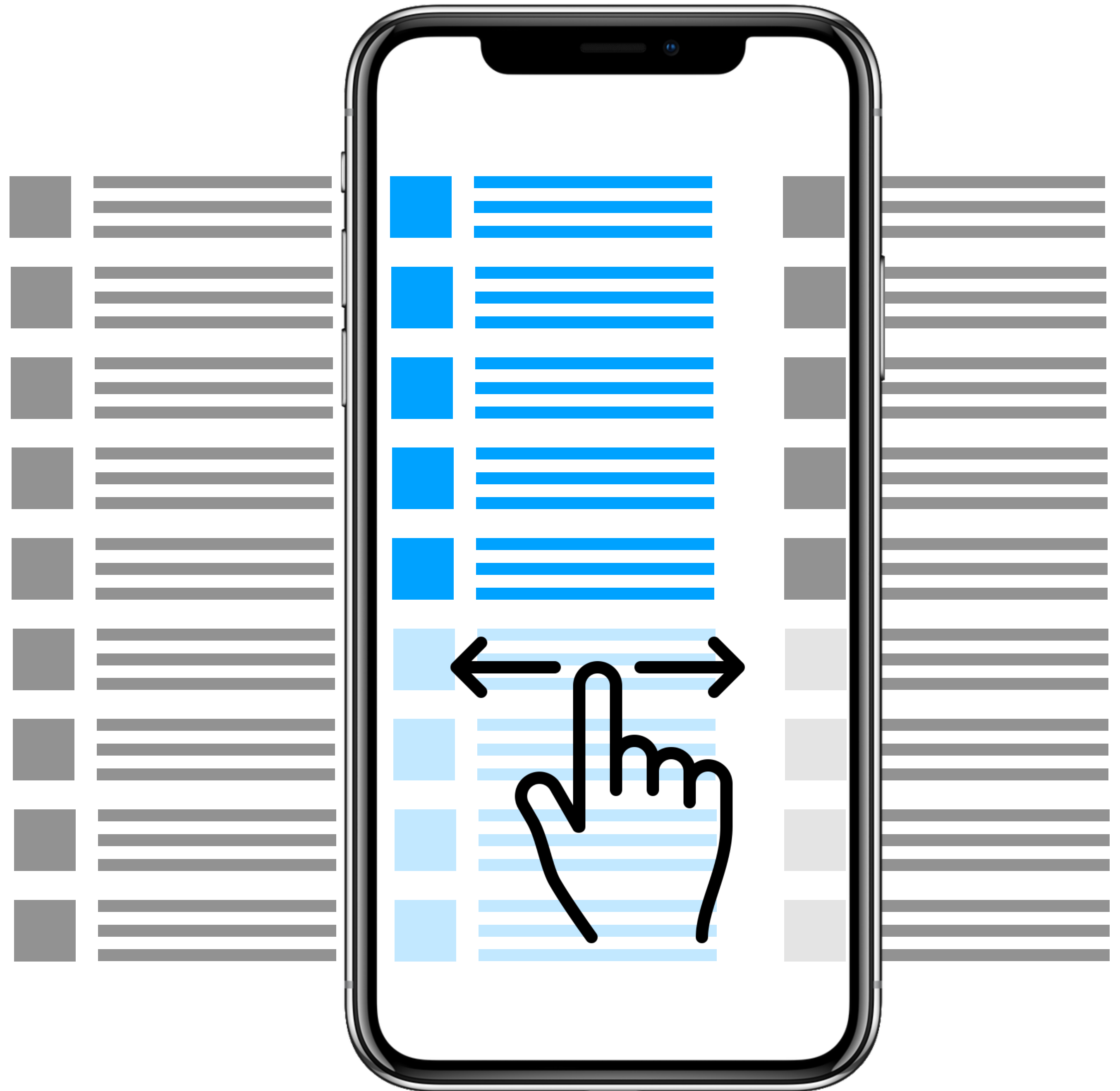


# Web page layout





# Mobile screen layout



# What to expect

- Data loading strategy can be different
- Data source structure can be different, SelectList for example

```
<SectionList
  sections=[
    {title: 'Title1', data: ['item1', 'item2']},
    {title: 'Title2', data: ['item3', 'item4']},
    {title: 'Title3', data: ['item5', 'item6']},
  ]}
```



# The End

- Do not be very positive with your **shared** code estimations
- Check each idea **before** “**yes**, we do it”
- Talk to your Product Owner, share the potential issues and limits
- Relax and write the code



**Q**uestion**?**

**@2j2e**