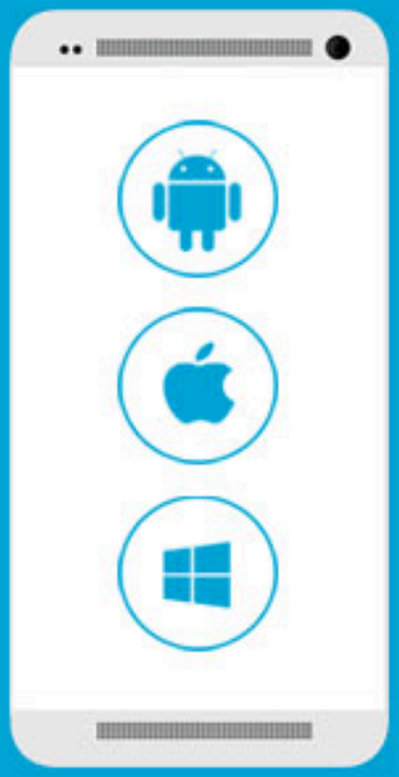
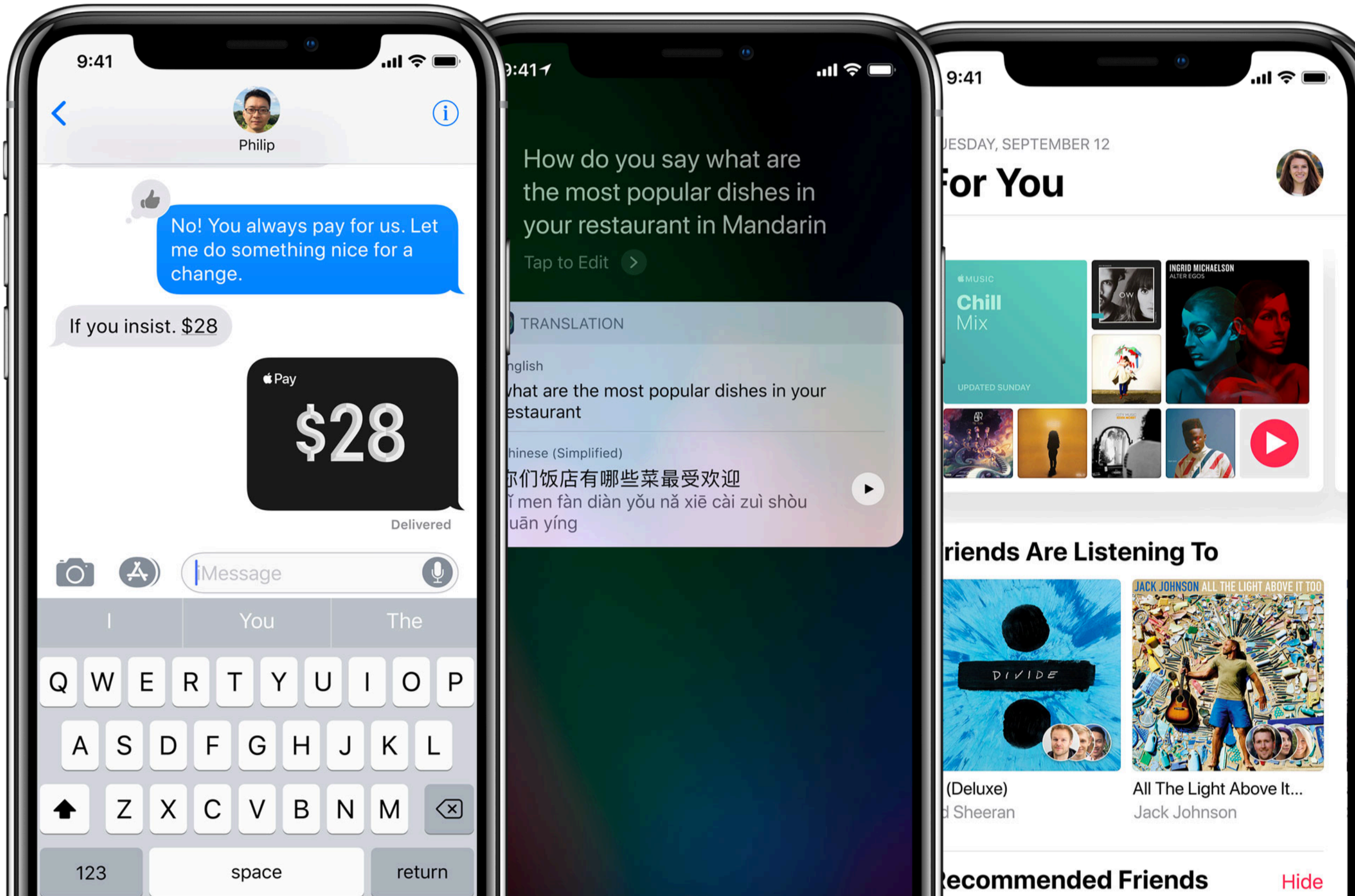


React Native

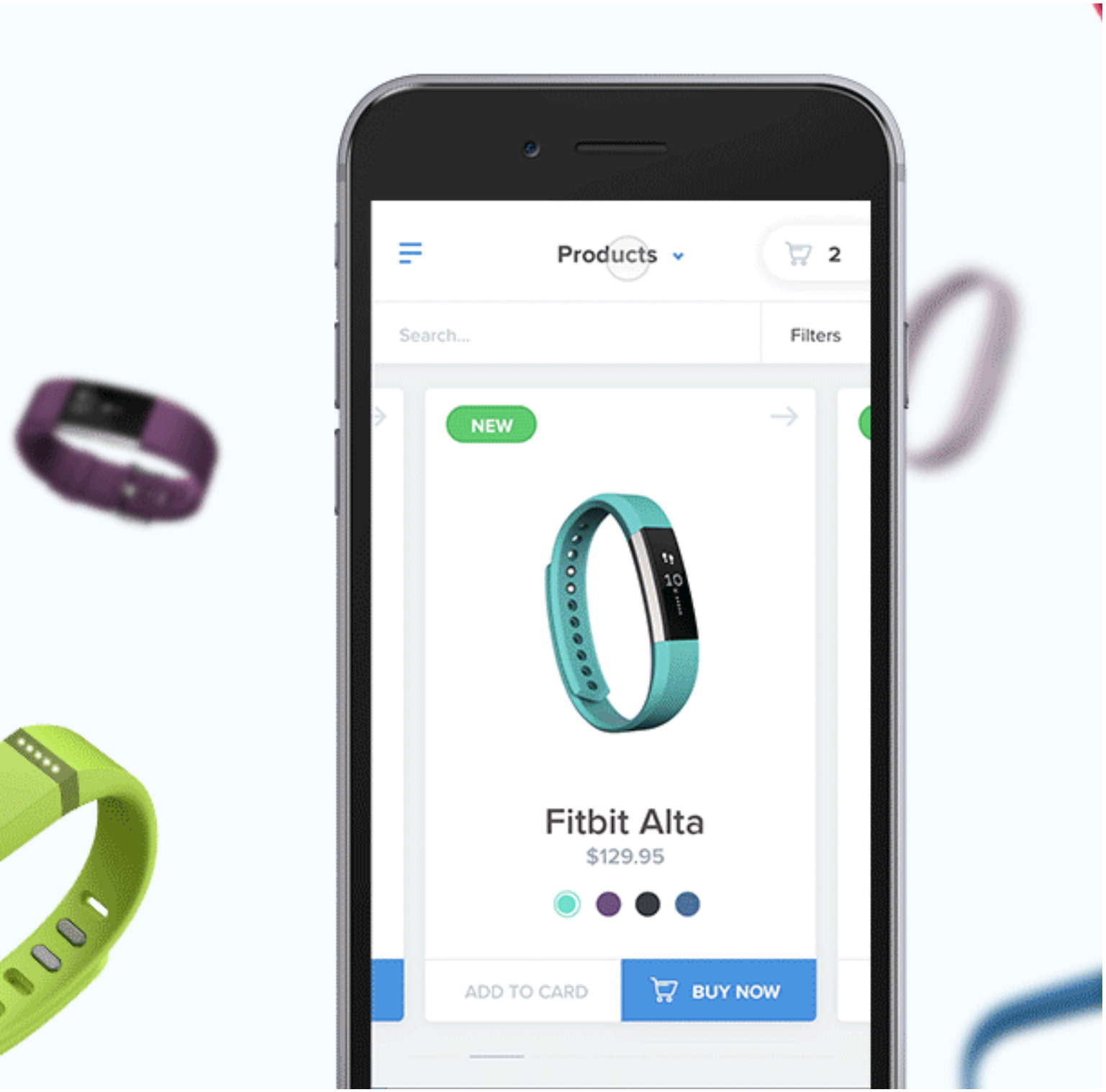
A framework for build native app using Javascript and React

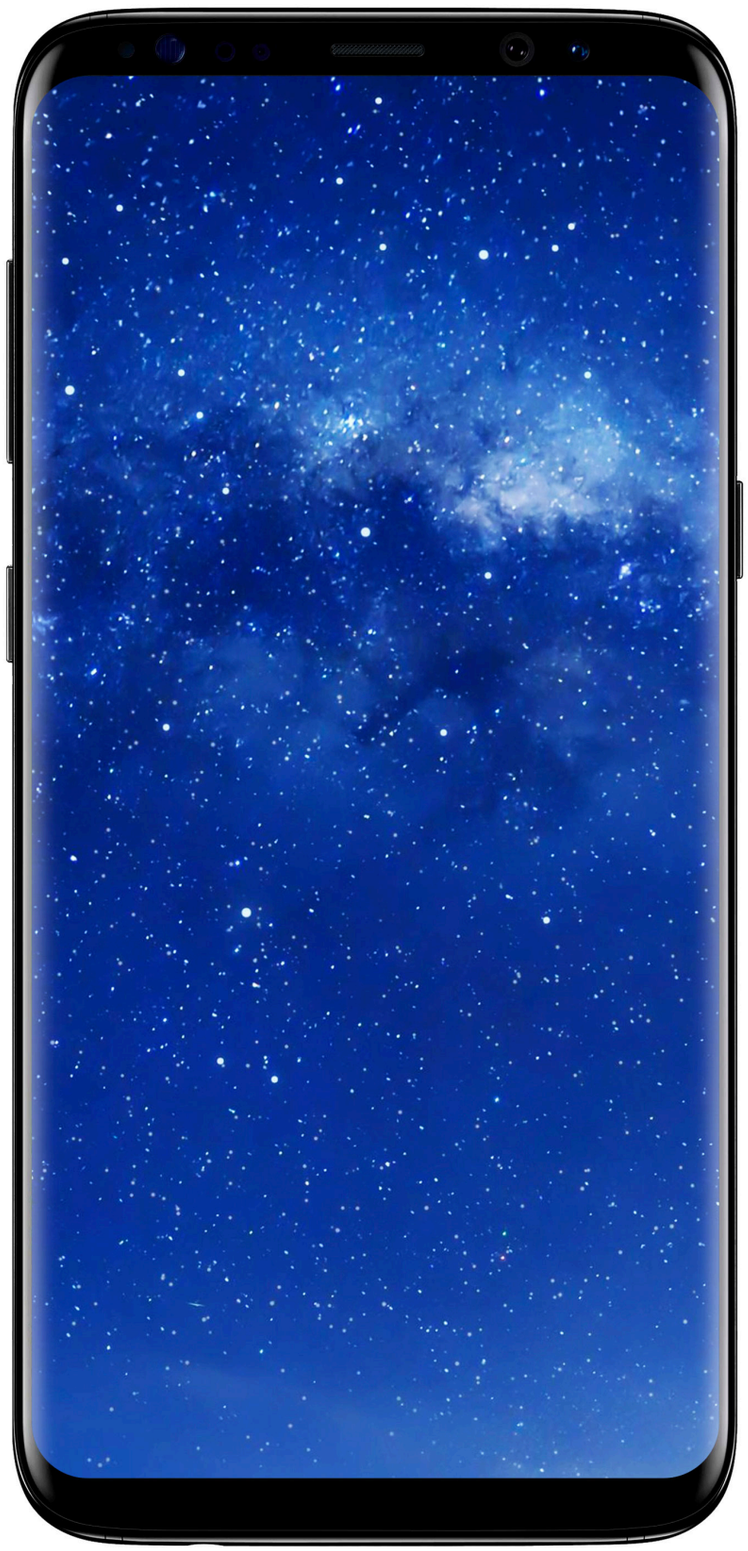
Speaker: Phan Thanh Tung - CoFounder of tungtung.vn
Email: tungptkh@gmail.com
Github: github.com/thanhtungdp

Iphone X


9:41
Philip
No! You always pay for us. Let me do something nice for a change.
If you insist. $28
Pay
$28
Delivered
iMessage
I
You
The
Q W E R T Y U I O P
A S D F G H J K L
Z X C V B N M
123
space
return
How do you say what are the most popular dishes in your restaurant in Mandarin
Tap to Edit
TRANSLATION
nglish
what are the most popular dishes in your estaurant
hinese (Simplified)
你们饭店有哪些菜最受欢迎
ǐ men fàn diàn yǒu nǎ xiē cài zuì shòu uān yíng
9:41
ESDAY, SEPTEMBER 12
For You
MUSIC
Chill Mix
UPDATED SUNDAY
INGRID MICHAELSON
ALTER EGOS
riends Are Listening To
DIVIDE
(Deluxe)
d Sheeran
JACK JOHNSON ALL THE LIGHT ABOVE IT TOO
All The Light Above It...
Jack Johnson
ecommended Friends
Hide

mobile-shopping-react-native





Agenda

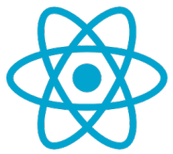
- Overview
- Show cases
- Best Example Lib
- Best Feature
- Know your react native work
- React concept
- From React to React Native
- How to learn
- Redux Concept
- Ecosystem

What's React Native

- A framework for building mobile app using Javascript and React library
- Build cross platform apps (Android / IOS)
- Uses almost all React.JS concepts including components, state, props, lifecycles, etc
- React Native develop by Facebook
- React Native Community

Real Mobile Apps

- React Native apps are not hybrid apps
- Do not run in a WebView
- Use the same fundamental building blocks as native app built with Swift/Objective C/Java
- Better and faster than Cordova / PhoneGap / Ionic



React Native

APACHE CORDOVA™

Show cases

Facebook

iOS · Android

Using React Native in the Facebook App

Facebook Ads Manager

iOS · Android

How We Built the First Cross-Platform React Native App

Instagram

iOS · Android

React Native at Instagram

F8

iOS · Android

Tutorial: Building the F8 conference app

Airbnb

iOS · Android

Hybrid React Native Apps at Airbnb

Skype

iOS · Android

Tesla

iOS · Android

Walmart

iOS · Android

React Native at Walmart Labs

Show cases

Artsy

iOS

React Native at Artsy

Baidu Mobile (手机百度)

iOS · Android

Baidu Mobile is a search engine used by over 600 million people in China

Bloomberg

iOS · Android

How Bloomberg Used React Native to Develop its new Consumer App

CBS Sports Franchise Football

Android

Award winning Fantasy Football league manager built with React Native

Chop

iOS

How we built Chop using React Native

Delivery.com

iOS · Android

React Native in an Existing iOS App: Getting Started

Discord

iOS

Using React Native: One Year Later

Flare by GoDaddy

iOS · Android

Social network that connects entrepreneurs to fellow entrepreneurs, consumers, investors

Show cases

Gyroscope

iOS

Building a visualization experience with React Native

Huiseoul (惠首尔)

iOS

Building a conversational E-commerce app in 6 weeks with React Native

JD（手机京东）

iOS · Android

JD.com is China’s largest ecommerce company by revenue and a member of the Fortune Global 500.

li.st

Android

Building li.st for Android with React Native

SoundCloud Pulse

iOS · Android

Why React Native worked well for us

Tencent QQ

Android

QQ is China's largest messaging platform, with

Townske

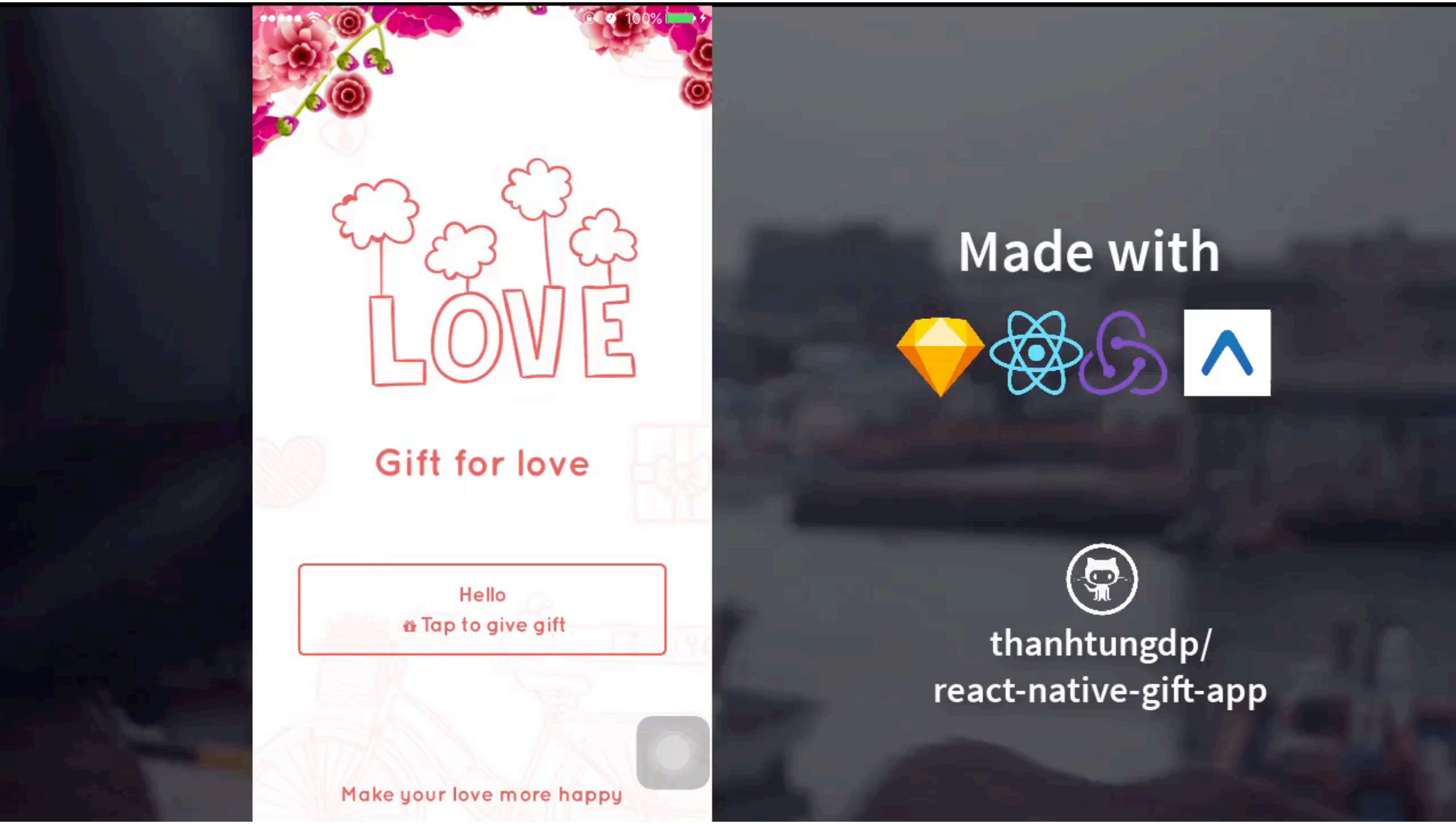
iOS

The experience of a web developer building an

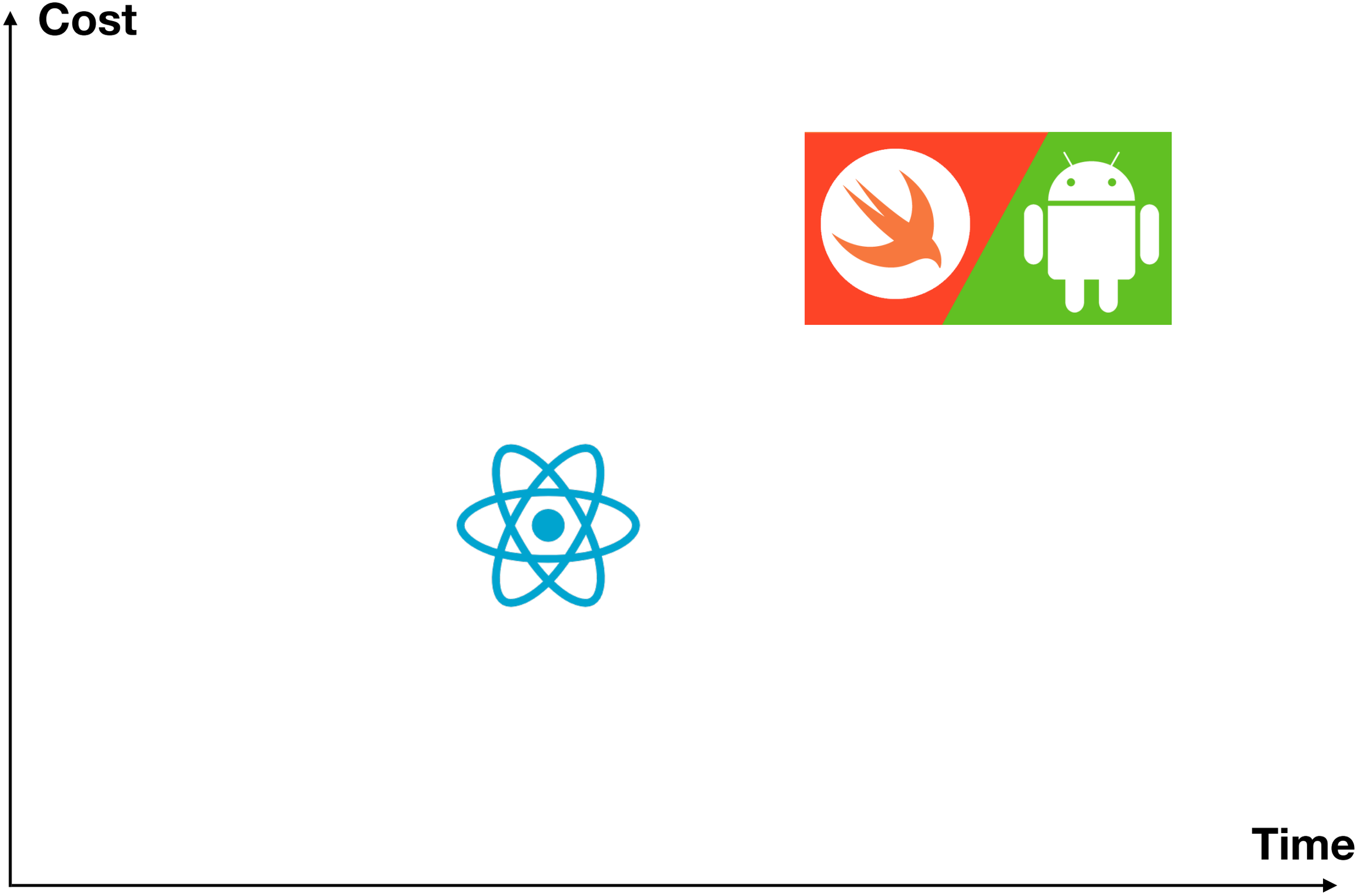
UberEATS

Powering UberEATS with React Native and Uber Engineering

Show cases


LOVE
Gift for love
Hello
Tap to give gift
Make your love more happy
Made with
thanhtungdp/
react-native-gift-app

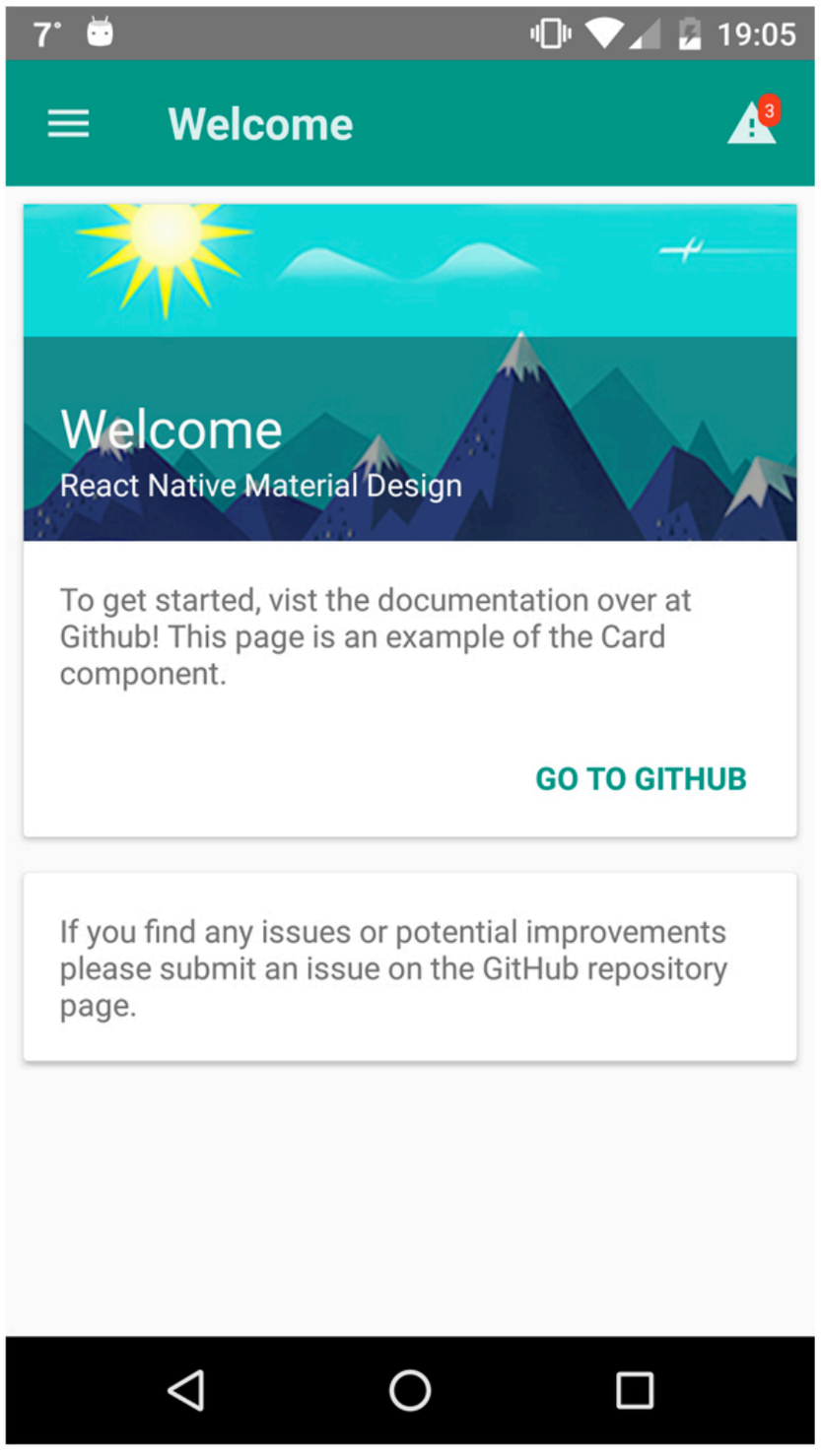
React Native For Startup

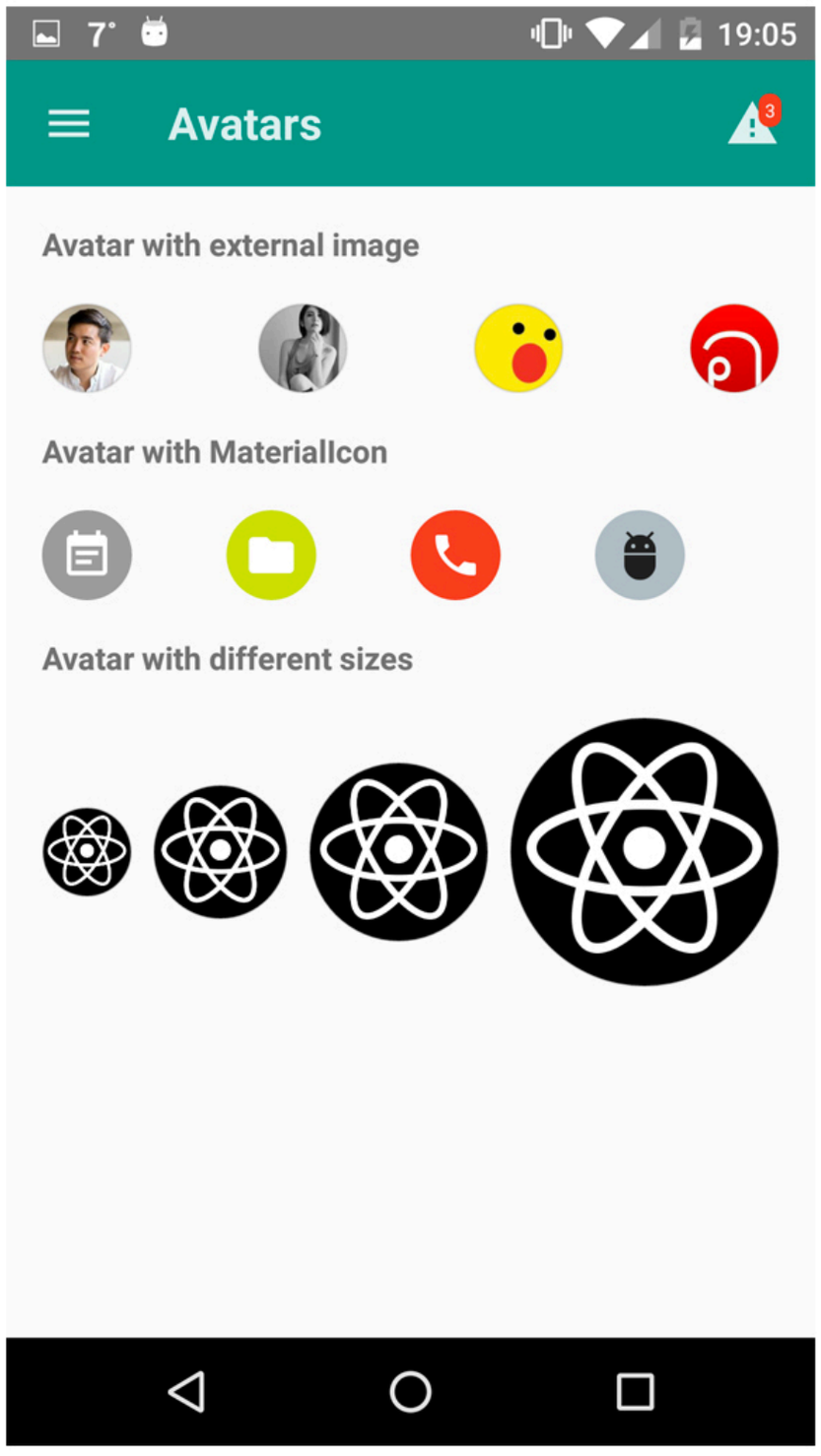


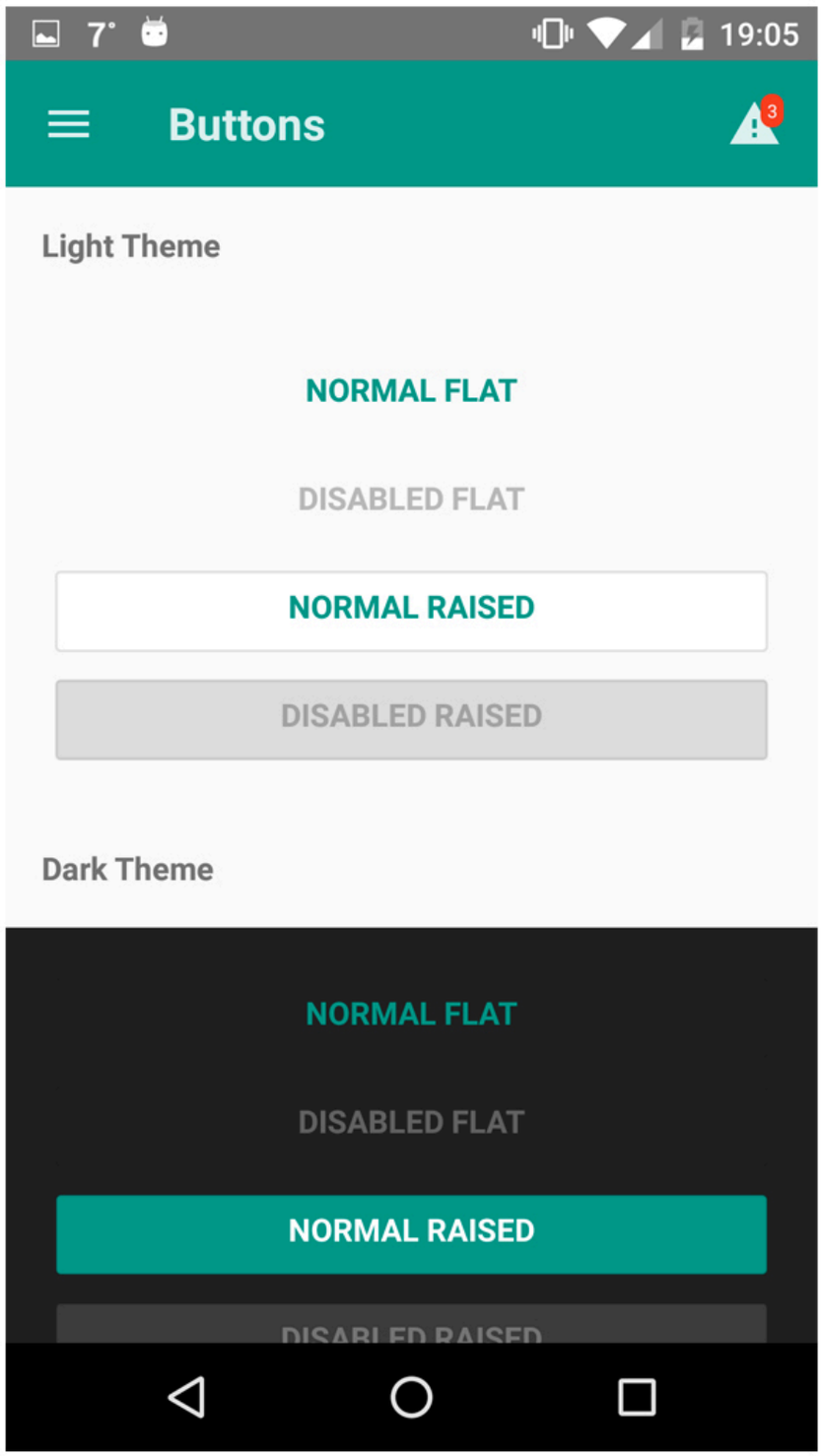
React Native

Example Lib

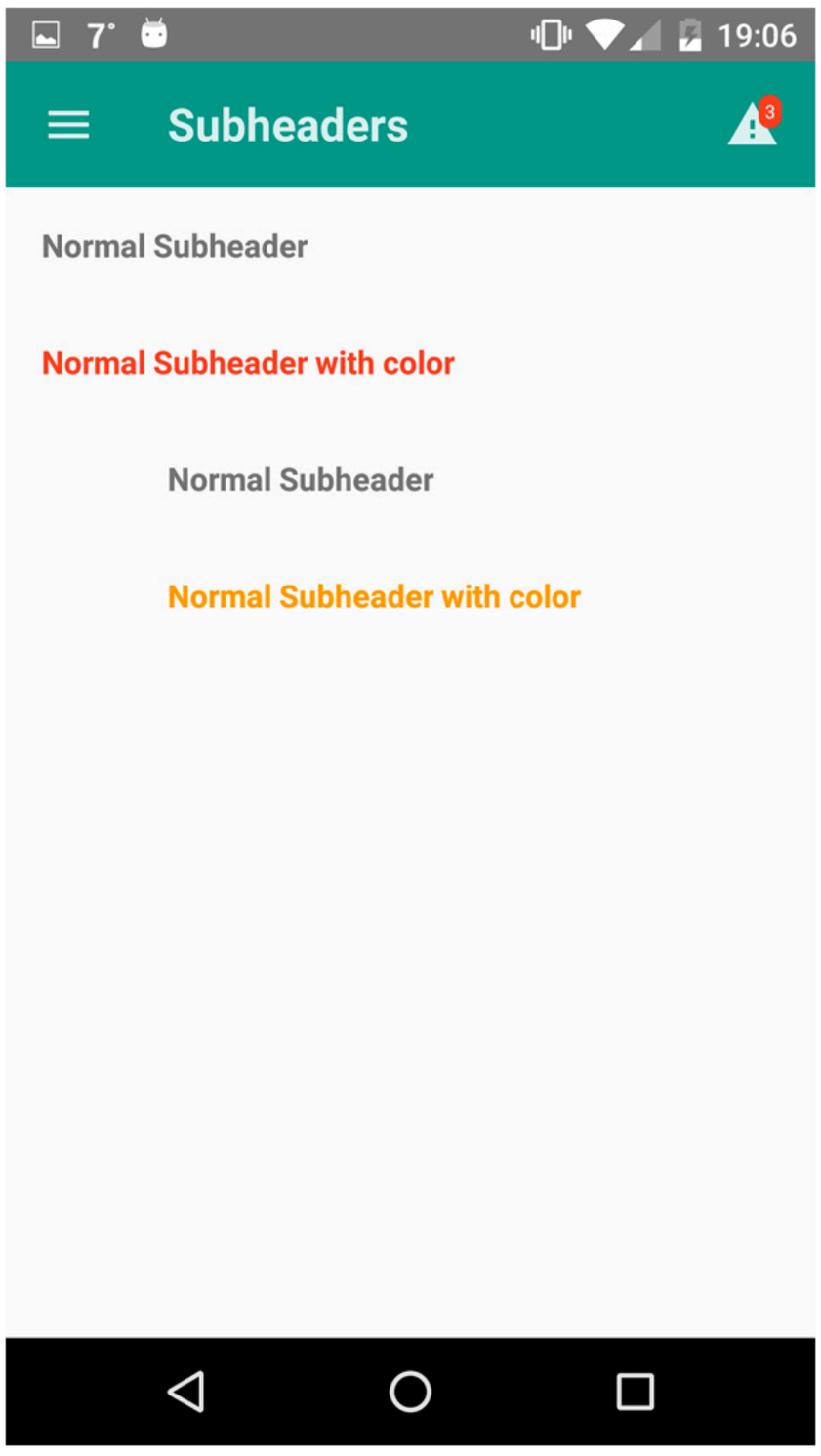
react-native-material-design

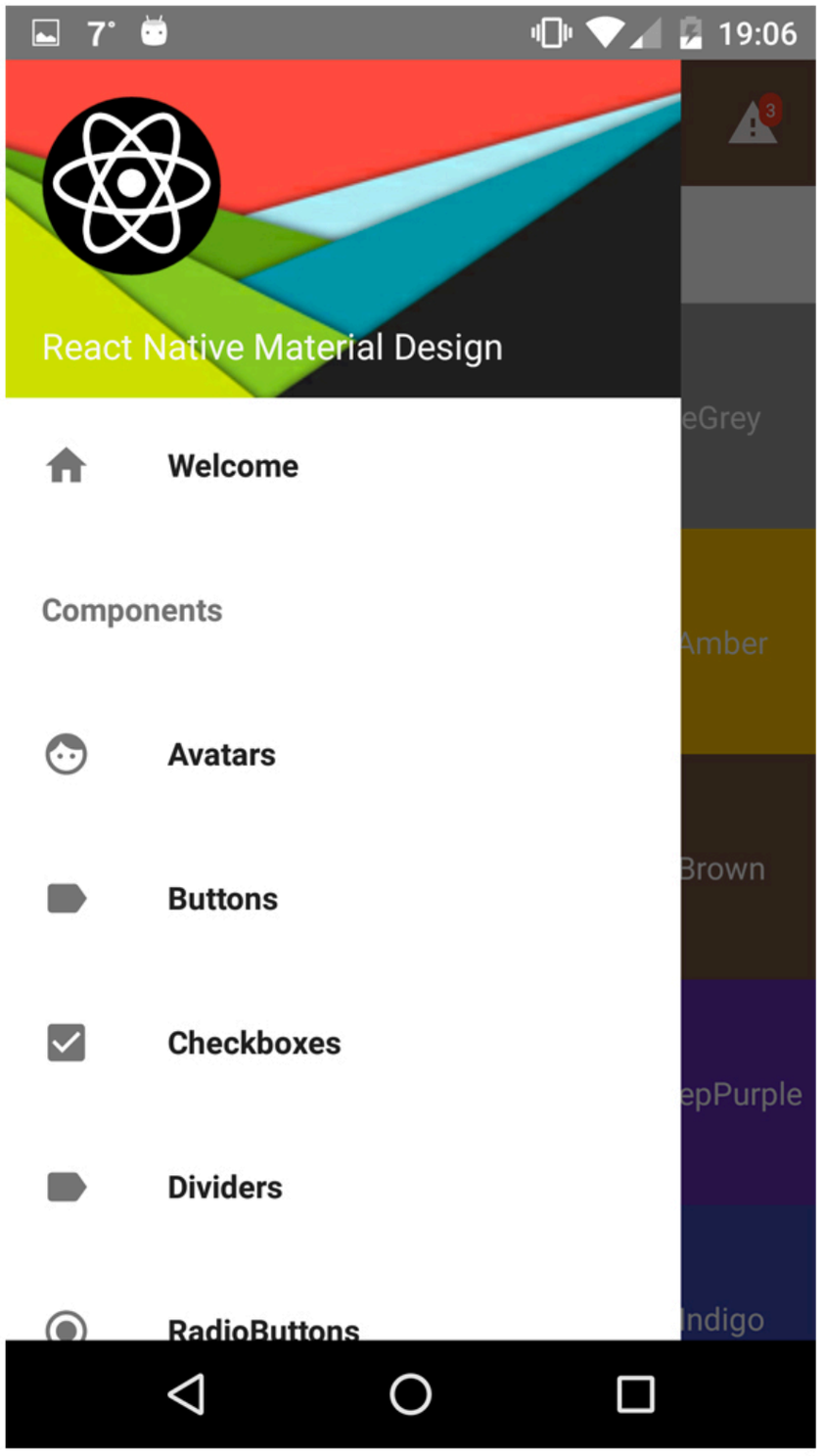




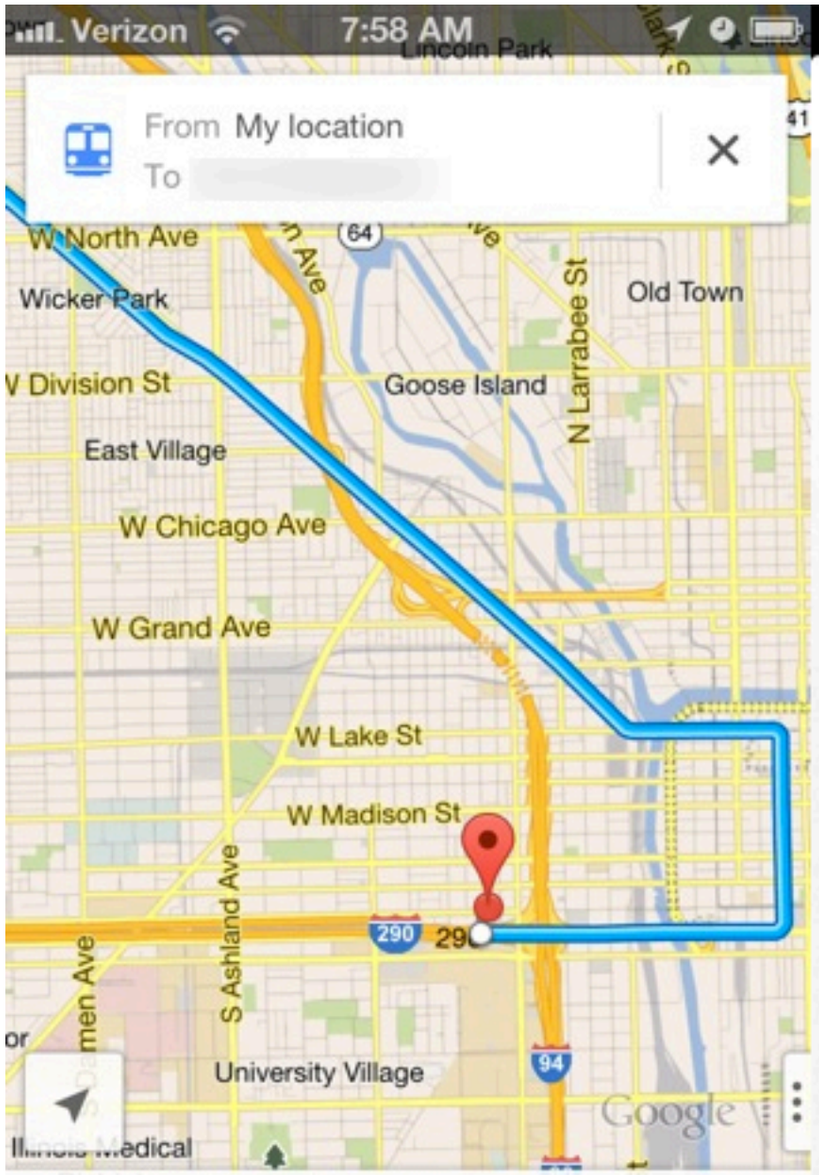


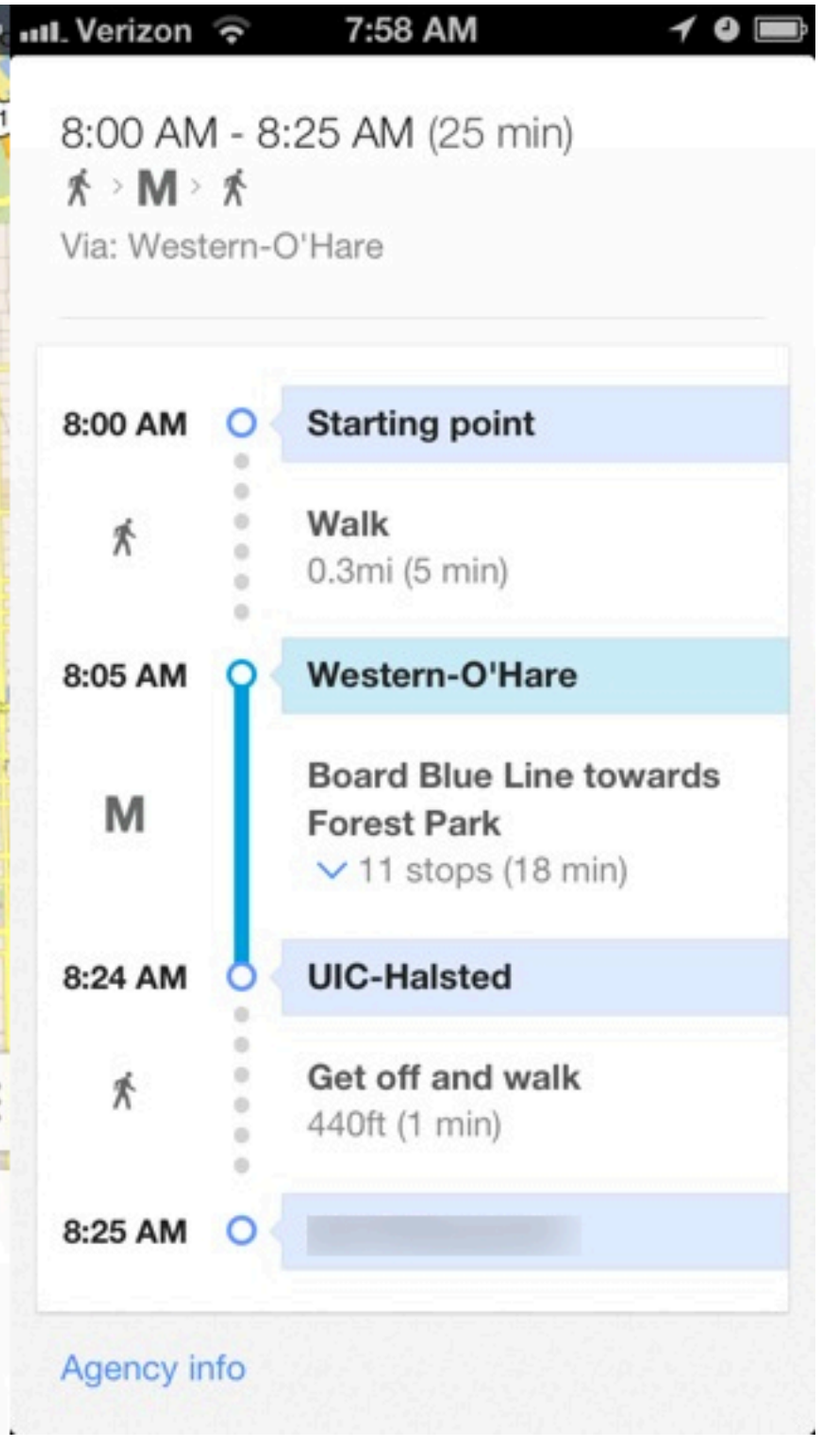
react-native-material-design

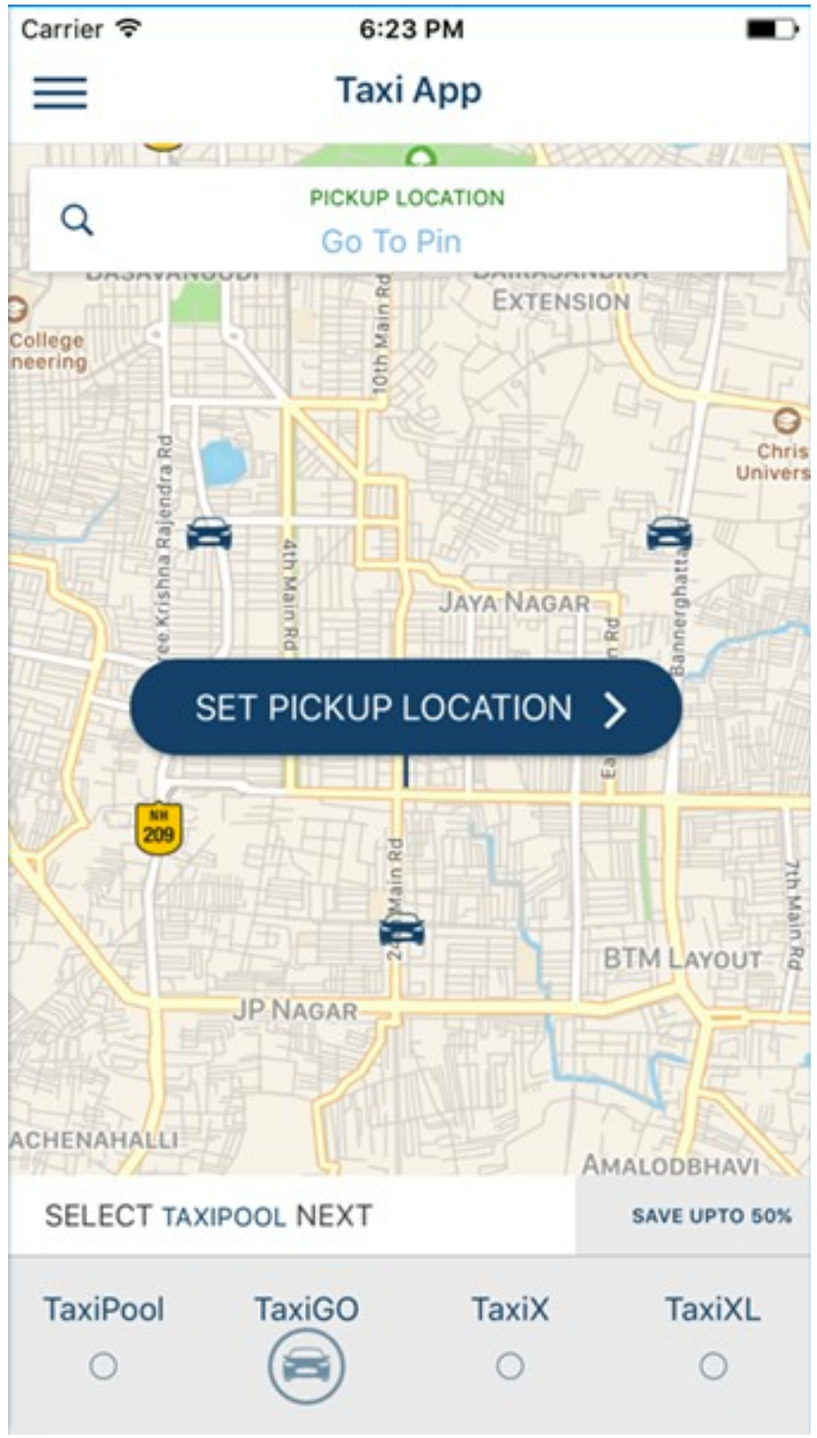




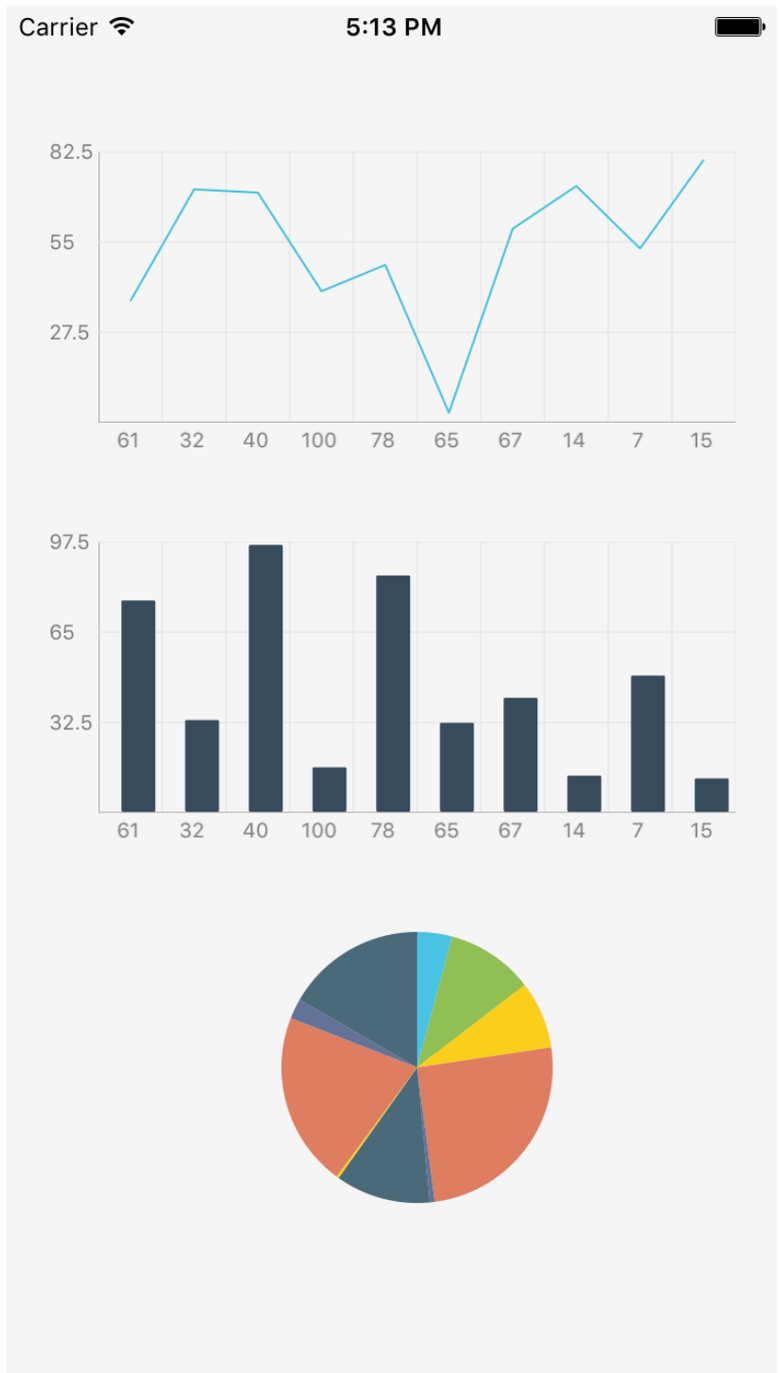
react-native-maps



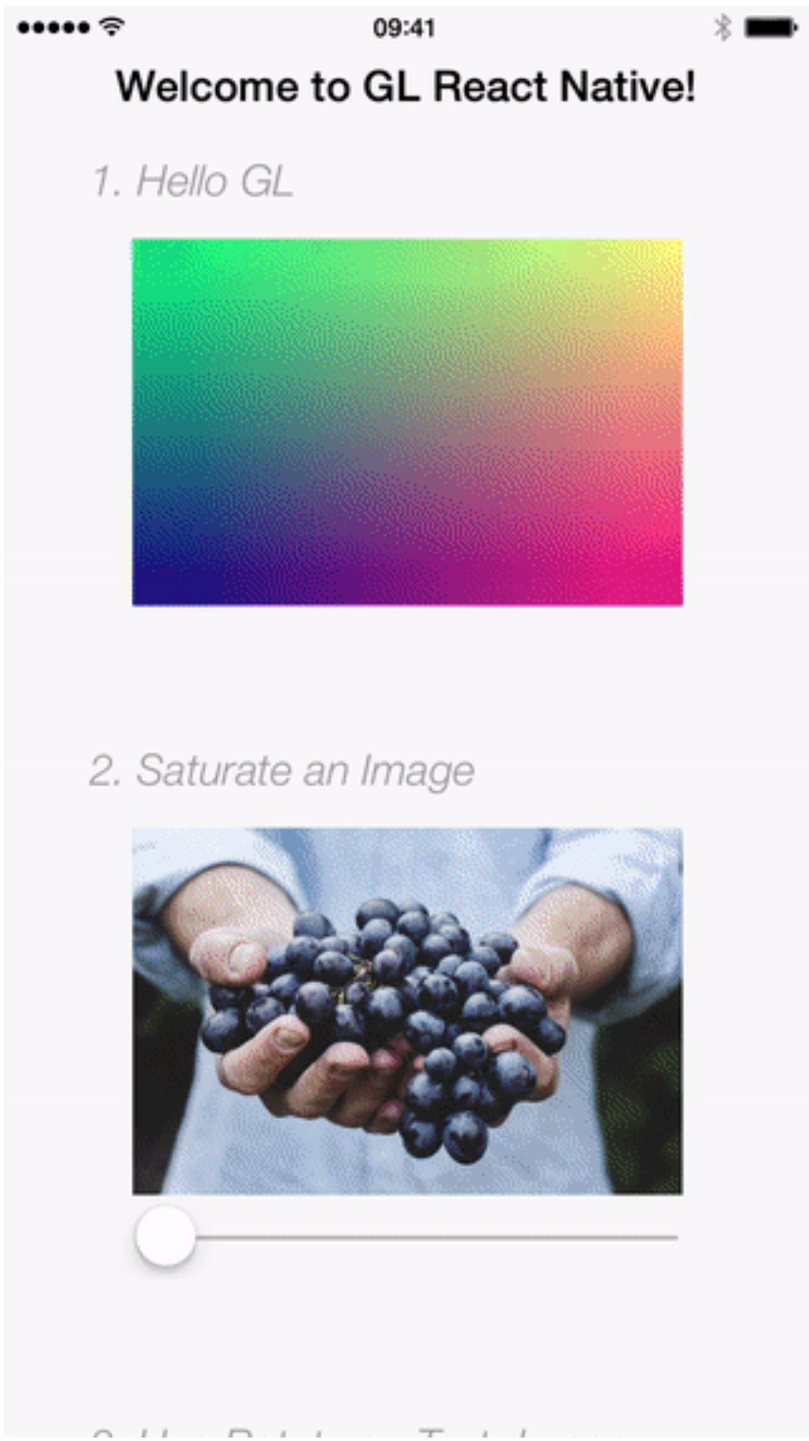




react-native-chart



gl-react-native



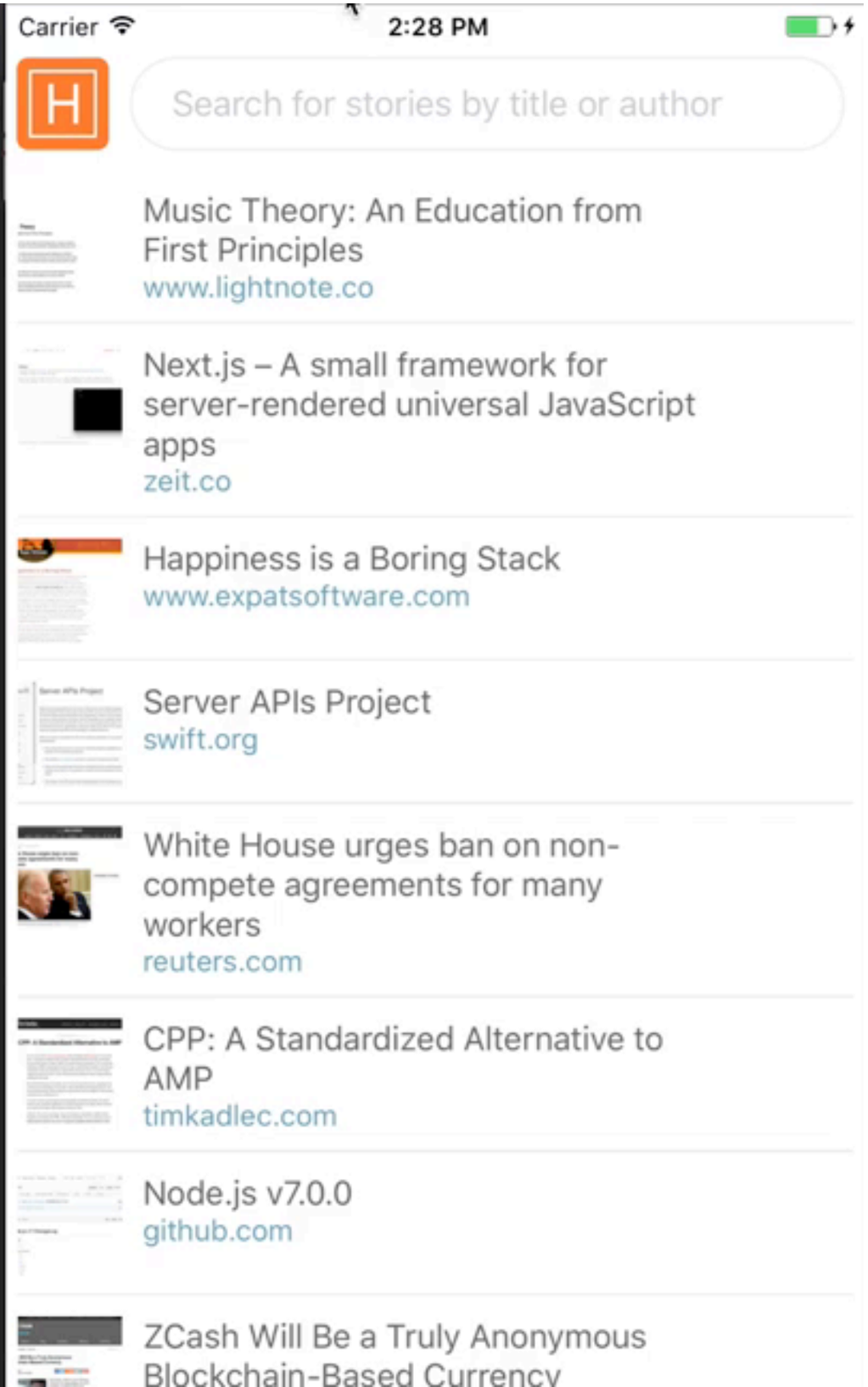
React Native Feature

React Native Feature

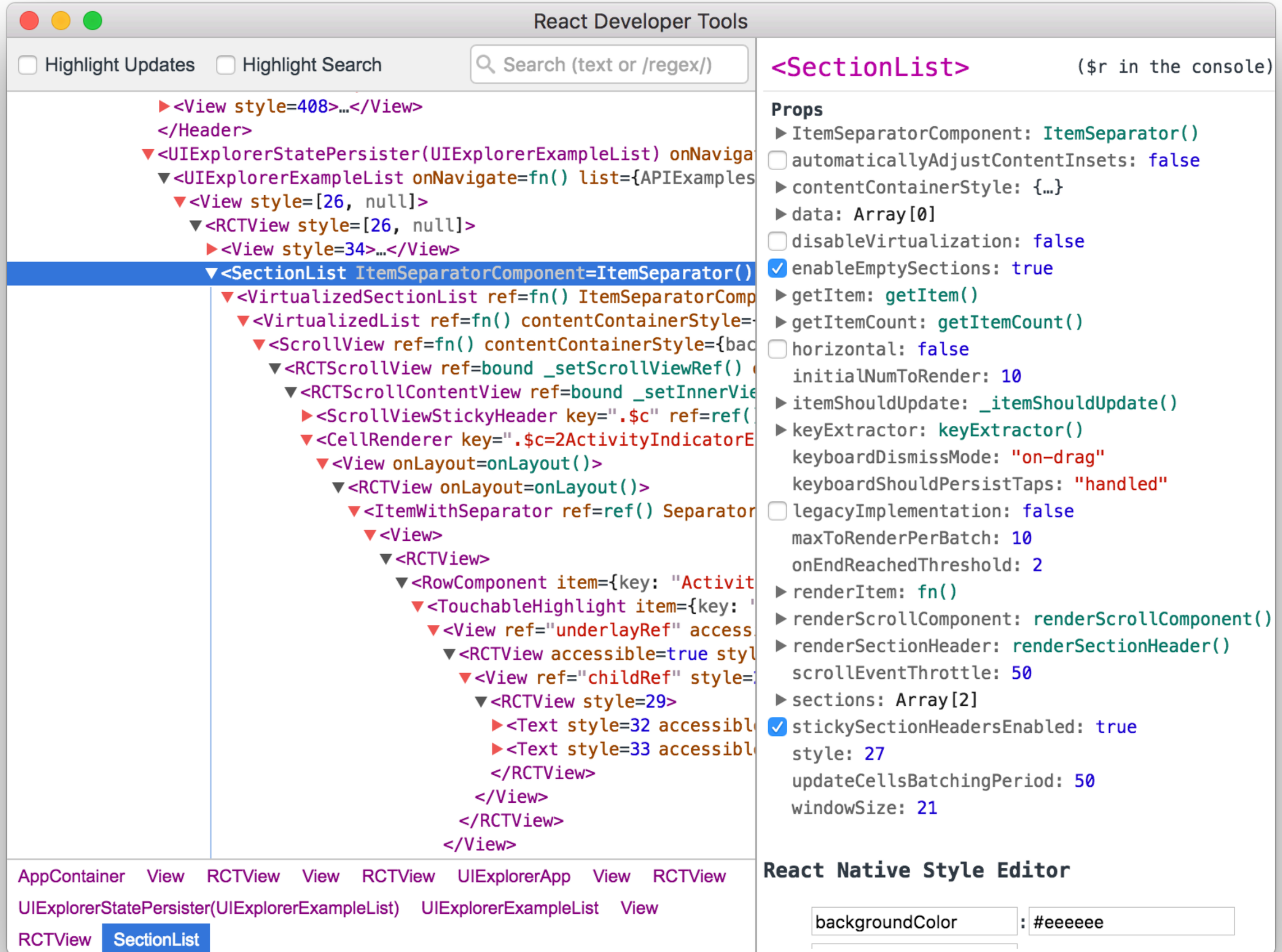
- Live hot reloading
- Debug in web
- Flex Style like Web App
- Shared code between web and App
- Bridge with native code: Swift, Java
- Codepush for deploy to Appstore, CHPlay
- Easy to learn, ~~NOT NEED MAC OS X~~

LIVE HOT RELOADING

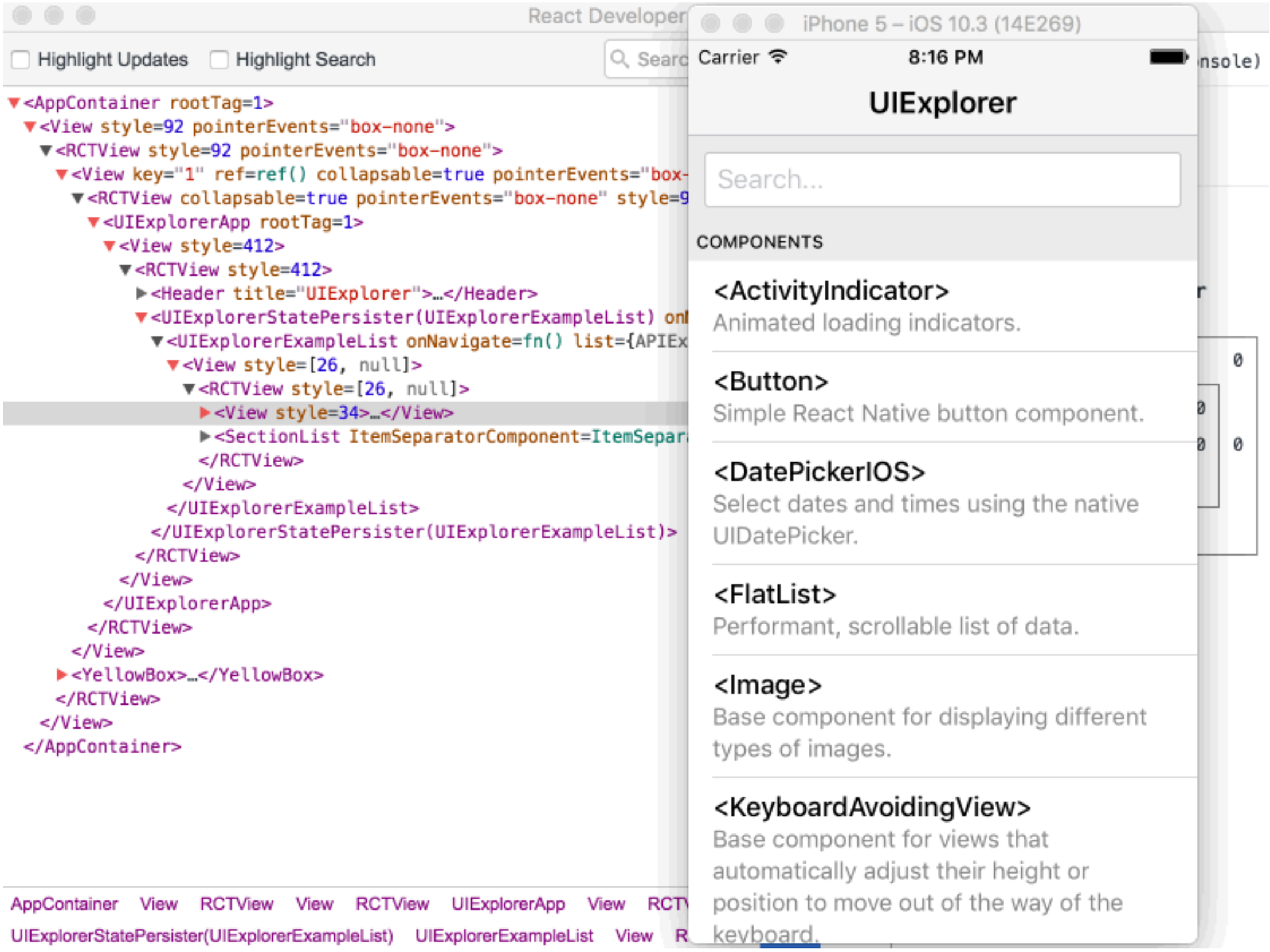
```
13      },
14      results: {
15        padding: 5
16      },
17      header: {
18        padding: 5,
19        marginTop: 20,
20        flexDirection: 'row'
21      },
22      logo: {
23        height: 45,
24        width: 45,
25        marginRight: 5
26      },
27      searchInput: {
28        height: 45,
29        borderColor: '#eee',
30        borderWidth: 1,
31        borderRadius: 22,
```



DEBUG TOOLS


React Developer Tools
Highlight Updates
Highlight Search
Search (text or /regex/)
<View style=408>…</View>
</Header>
<UIExplorerStatePersister(UIExplorerExampleList) onNaviga
<UIExplorerExampleList onNavigate=fn() list={APIExamples
<View style=[26, null]>
<RCTView style=[26, null]>
<View style=34>…</View>
<SectionList ItemSeparatorComponent=ItemSeparator()
<VirtualizedSectionList ref=fn() ItemSeparatorComp
<VirtualizedList ref=fn() contentContainerStyle=
<ScrollView ref=fn() contentContainerStyle={bac
<RCTScrollView ref=bound _setScrollViewRef()
<RCTScrollContentView ref=bound _setInnerVie
<ScrollViewStickyHeader key=".$c" ref=ref(
<CellRenderer key=".$c=2ActivityIndicatorE
<View onLayout=onLayout()>
<RCTView onLayout=onLayout()>
<ItemWithSeparator ref=ref() Separator
<View>
<RCTView>
<RowComponent item={key: "Activit
<TouchableHighlight item={key:
<View ref="underlayRef" access
<RCTView accessible=true styl
<View ref="childRef" style=
<RCTView style=29>
<Text style=32 accessibl
<Text style=33 accessibl
</RCTView>
</View>
</RCTView>
</View>
AppContainer View RCTView View RCTView UIExplorerApp View RCTView
UIExplorerStatePersister(UIExplorerExampleList) UIExplorerExampleList View
RCTView SectionList
<SectionList>
($r in the console)
Props
ItemSeparatorComponent: ItemSeparator()
automaticallyAdjustContentInsets: false
contentContainerStyle: {…}
data: Array[0]
disableVirtualization: false
enableEmptySections: true
getItem: getItem()
getItemCount: getItemCount()
horizontal: false
initialNumToRender: 10
itemShouldUpdate: _itemShouldUpdate()
keyExtractor: keyExtractor()
keyboardDismissMode: "on-drag"
keyboardShouldPersistTaps: "handled"
legacyImplementation: false
maxToRenderPerBatch: 10
onEndReachedThreshold: 2
renderItem: fn()
renderScrollComponent: renderScrollComponent()
renderSectionHeader: renderSectionHeader()
scrollEventThrottle: 50
sections: Array[2]
stickySectionHeadersEnabled: true
style: 27
updateCellsBatchingPeriod: 50
windowSize: 21
React Native Style Editor
backgroundColor
#eeeeee

DEBUG TOOLS


React Developer
Highlight Updates
Highlight Search
▼<AppContainer rootTag=1>
▼<View style=92 pointerEvents="box-none">
▼<RCTView style=92 pointerEvents="box-none">
▼<View key="1" ref=ref() collapsable=true pointerEvents="box-
▼<RCTView collapsable=true pointerEvents="box-none" style=9
▼<UIExplorerApp rootTag=1>
▼<View style=412>
▼<RCTView style=412>
▶<Header title="UIExplorer">…</Header>
▼<UIExplorerStatePersister(UIExplorerExampleList) onN
▼<UIExplorerExampleList onNavigate=fn() list={APIEx
▼<View style=[26, null]>
▼<RCTView style=[26, null]>
▶<View style=34>…</View>
▶<SectionList ItemSeparatorComponent=ItemSepar
</RCTView>
</View>
</UIExplorerExampleList>
</UIExplorerStatePersister(UIExplorerExampleList)>
</RCTView>
</View>
</UIExplorerApp>
</RCTView>
</View>
▶<YellowBox>…</YellowBox>
</RCTView>
</View>
</AppContainer>
AppContainer View RCTView View RCTView UIExplorerApp View RCT
UIExplorerStatePersister(UIExplorerExampleList) UIExplorerExampleList View R
iPhone 5 – iOS 10.3 (14E269)
Carrier
8:16 PM
UIExplorer
Search...
COMPONENTS
<ActivityIndicator>
Animated loading indicators.
<Button>
Simple React Native button component.
<DatePickerIOS>
Select dates and times using the native UIDatePicker.
<FlatList>
Performant, scrollable list of data.
<Image>
Base component for displaying different types of images.
<KeyboardAvoidingView>
Base component for views that automatically adjust their height or position to move out of the way of the keyboard.

FLEXBOX STYLE LIKE WEB

justify-content: **flex-start**

Home Services About Contact

justify-content: **center**

Home Services About Contact

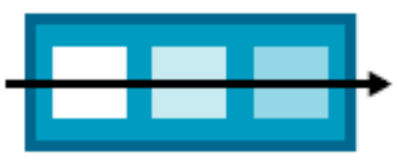
justify-content: **flex-end**

Home Services About Contact

justify-content: **space-between**

Home Services About Contact

justify-content: **space-around**

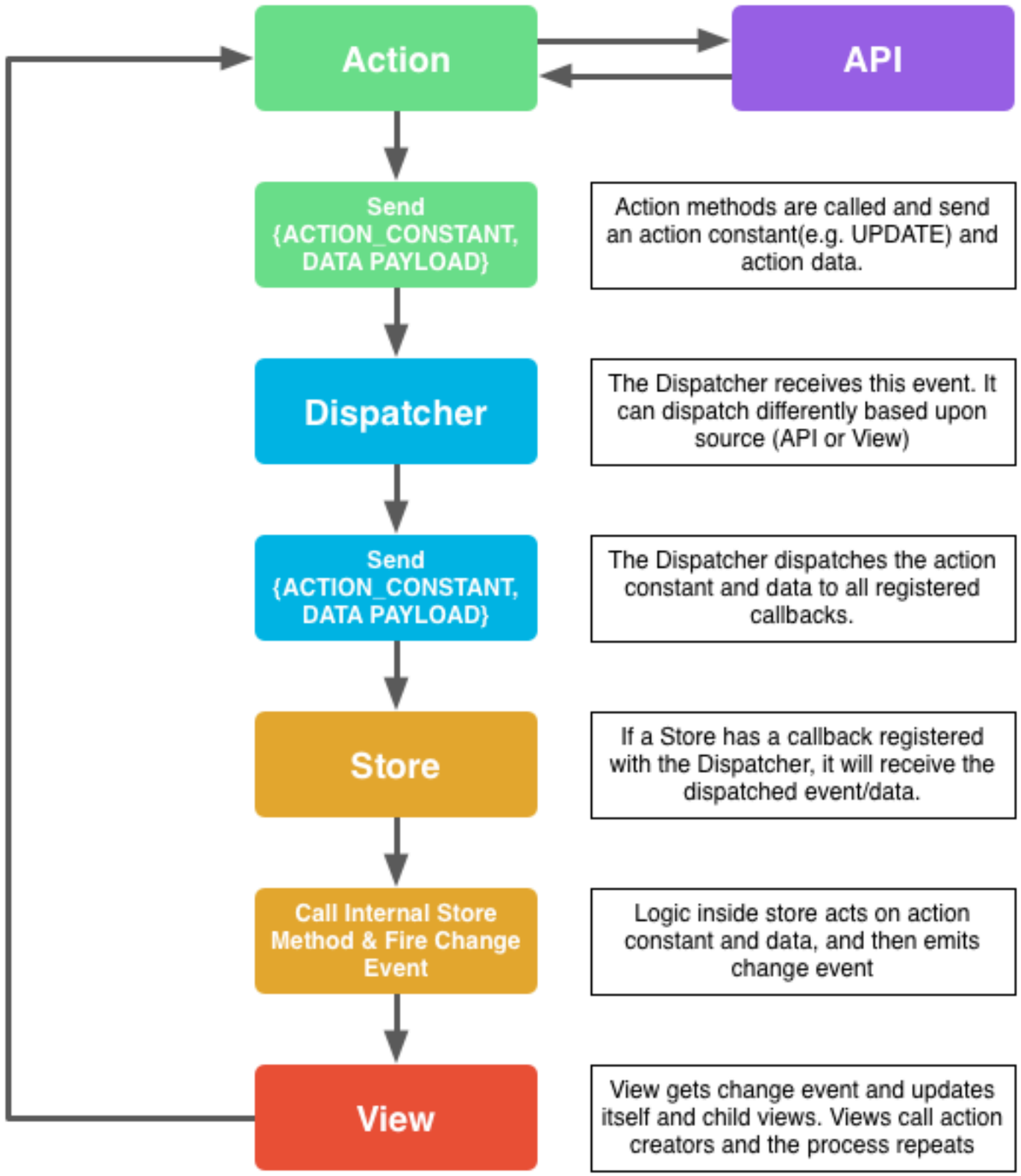


Home Services About Contact

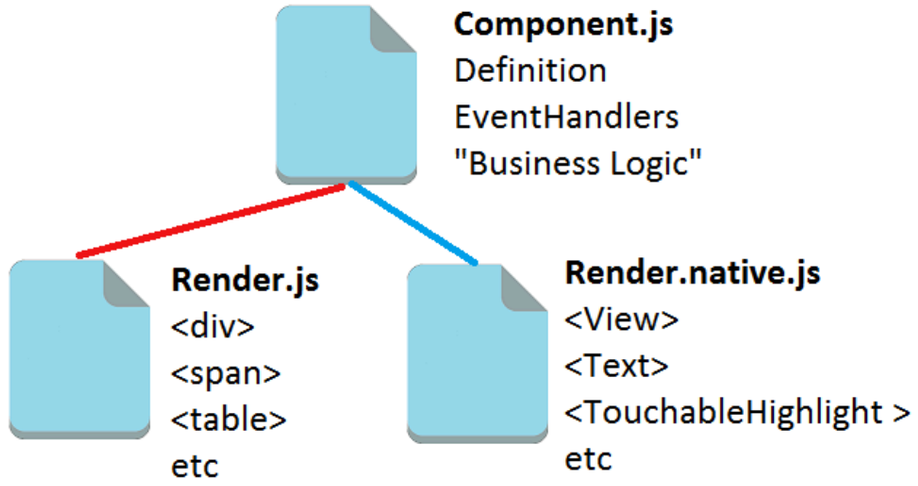
flexbox-direction: **row-reverse**
justify-content: **flex-start**

FLEXBOX EXAMPLE

SHARED CODE



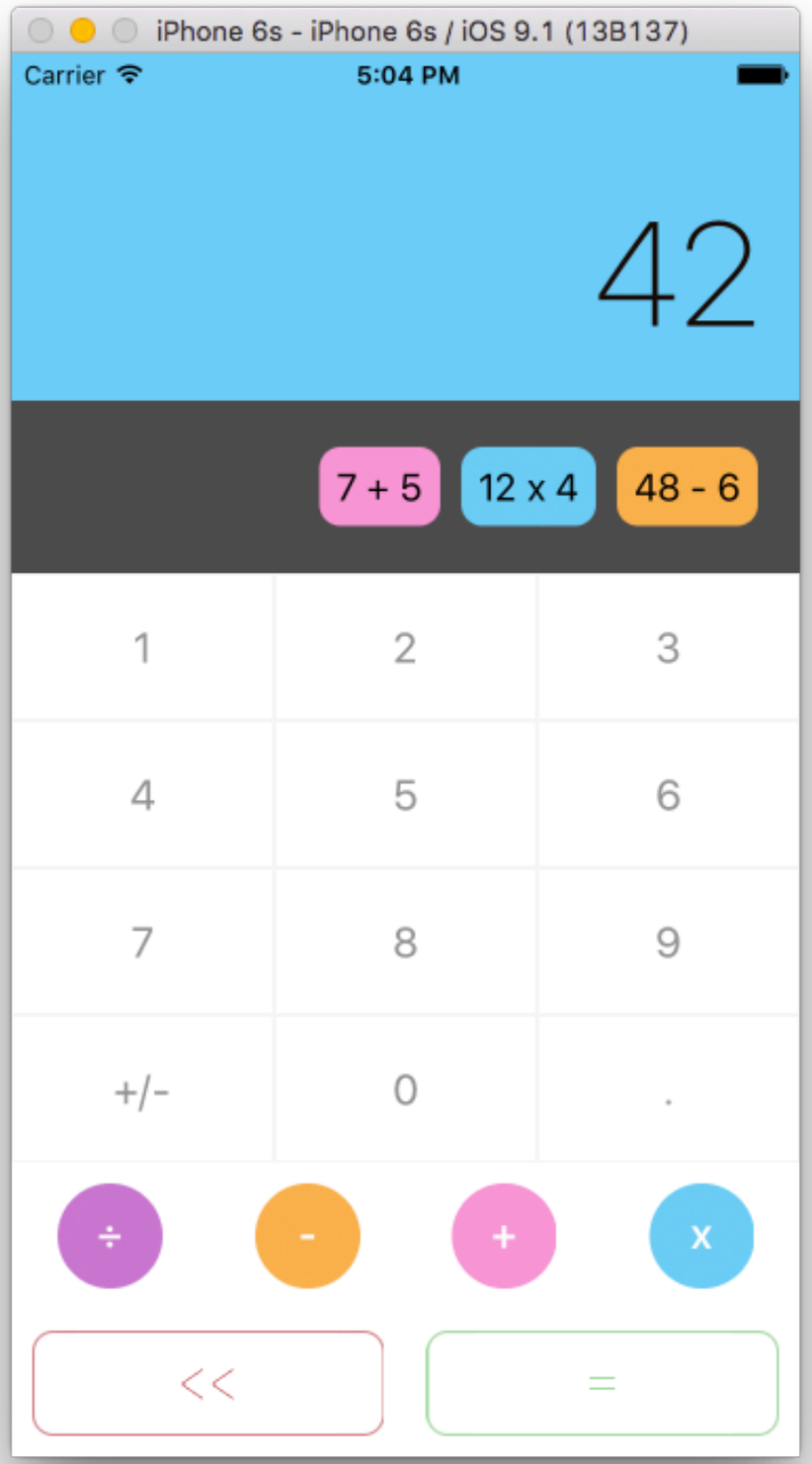
Business logic code

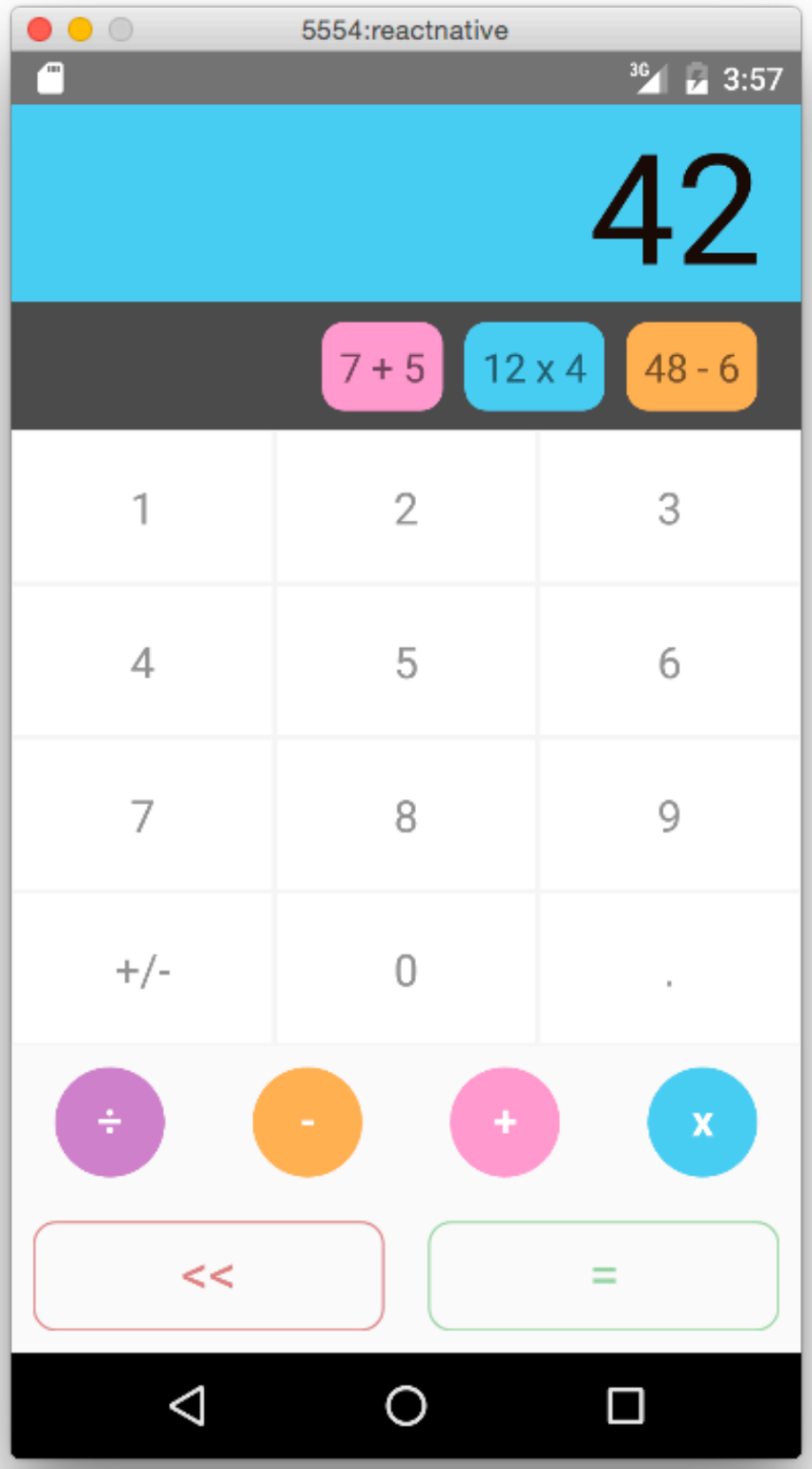


Seperator Component

EXAMPLE SHARED CODE: react-native-nw-react-calculator

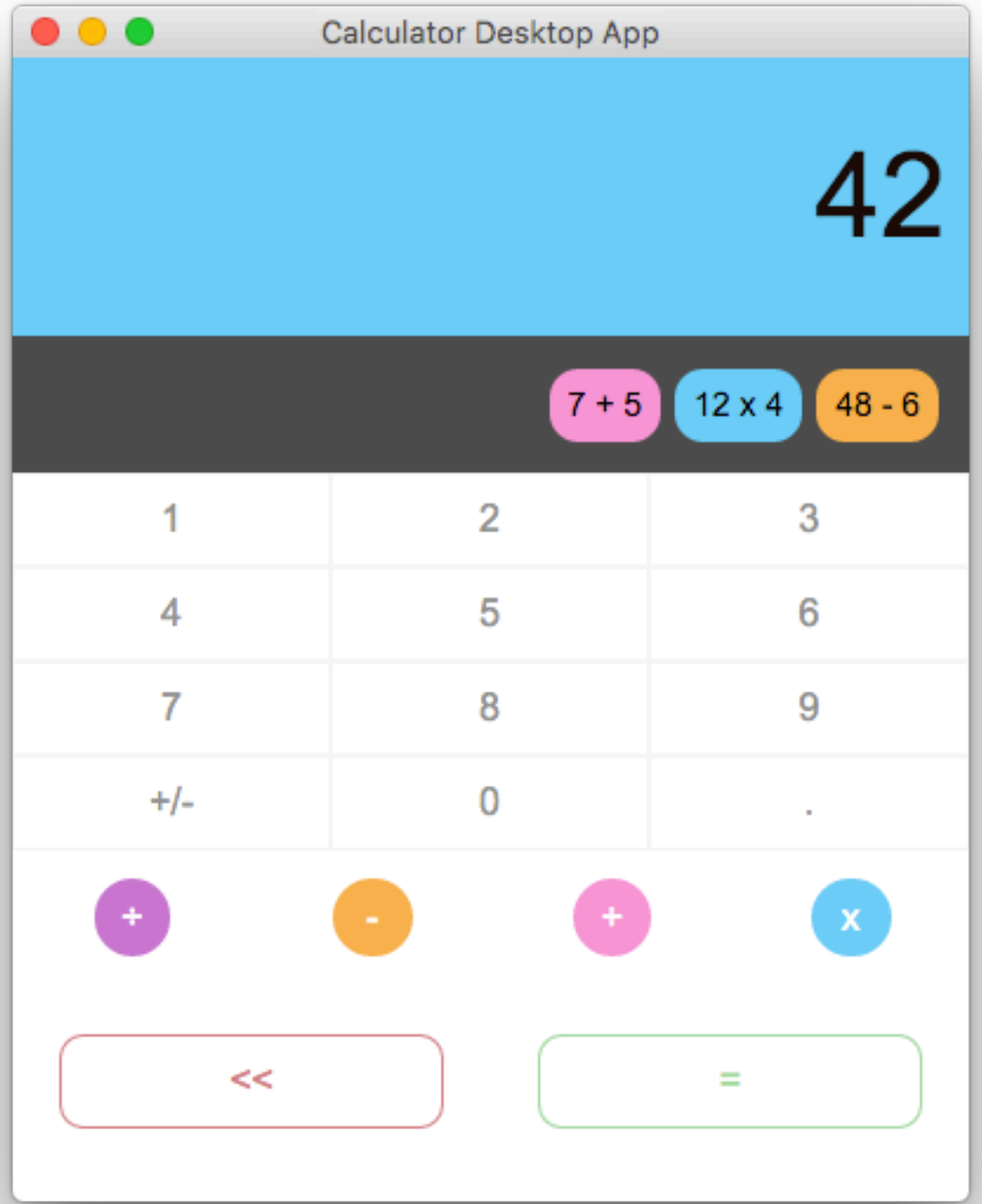
IOS & Android APP

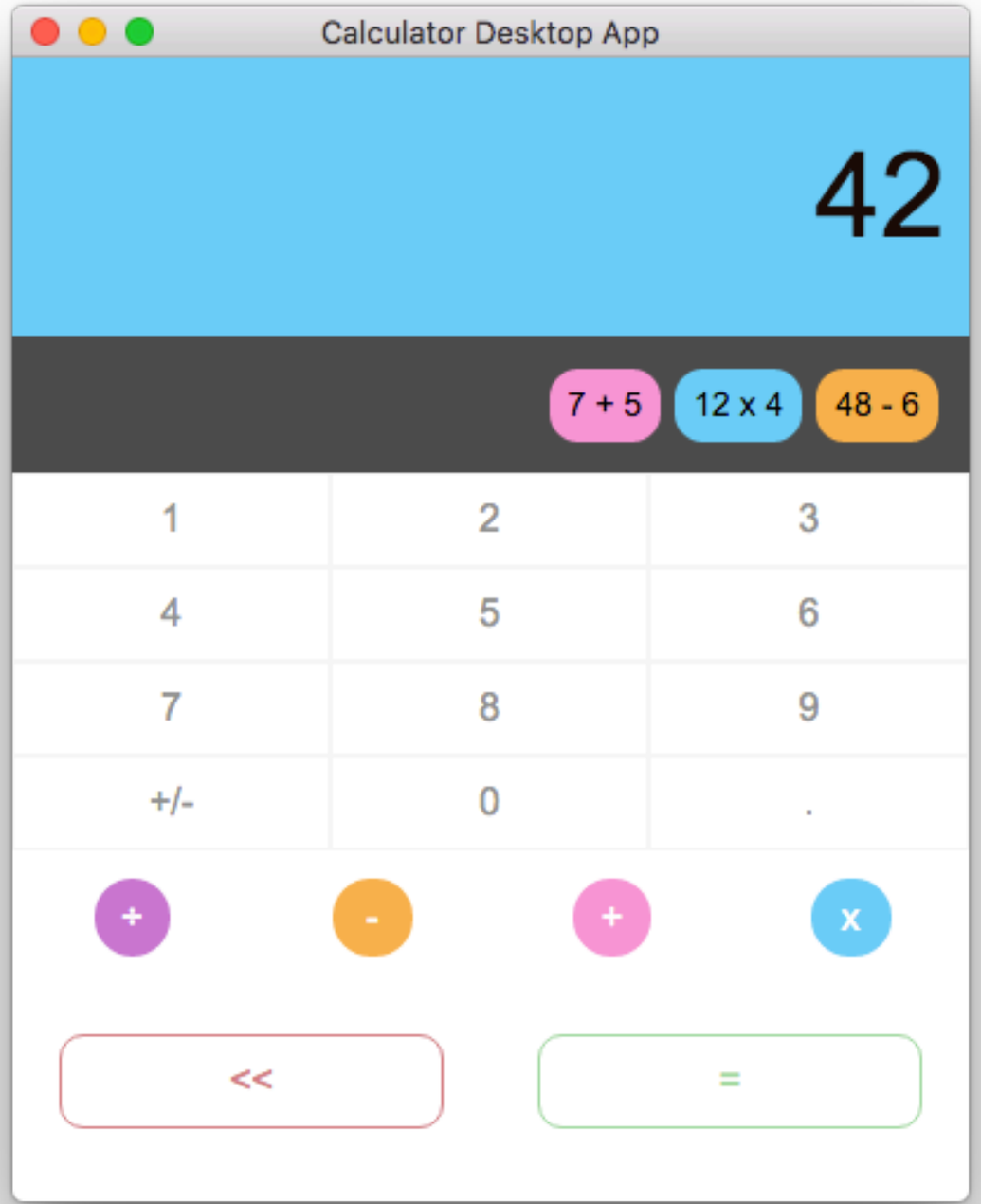




EXAMPLE SHARED CODE: react-native-nw-react-calculator

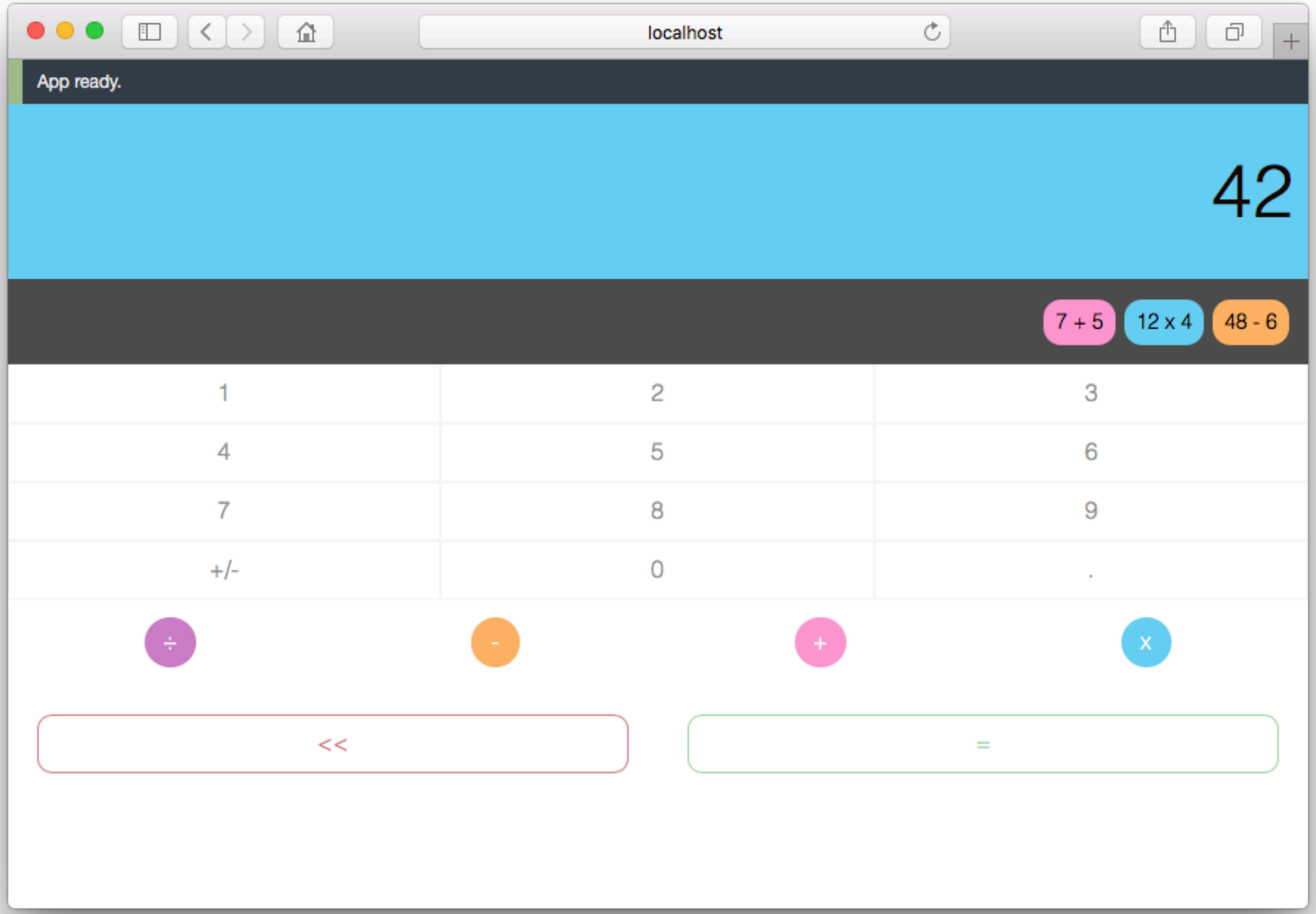
Desktop Apps (NW & Electron)





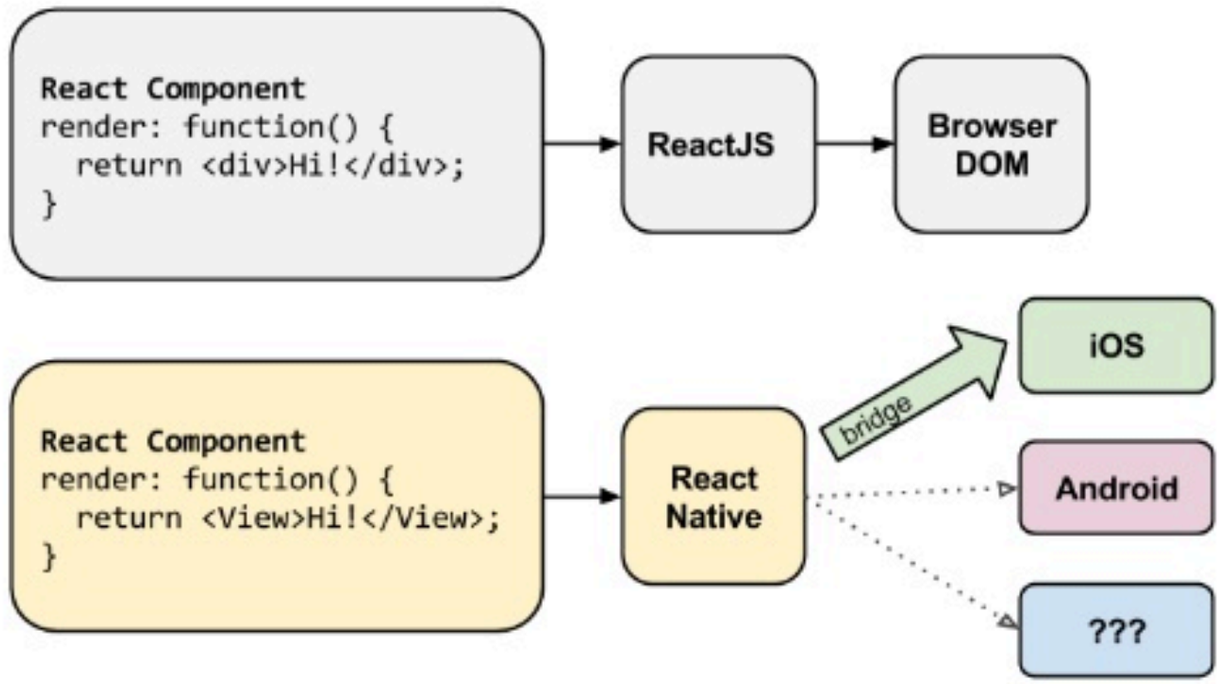
EXAMPLE SHARED CODE: react-native-nw-react-calculator

Website App



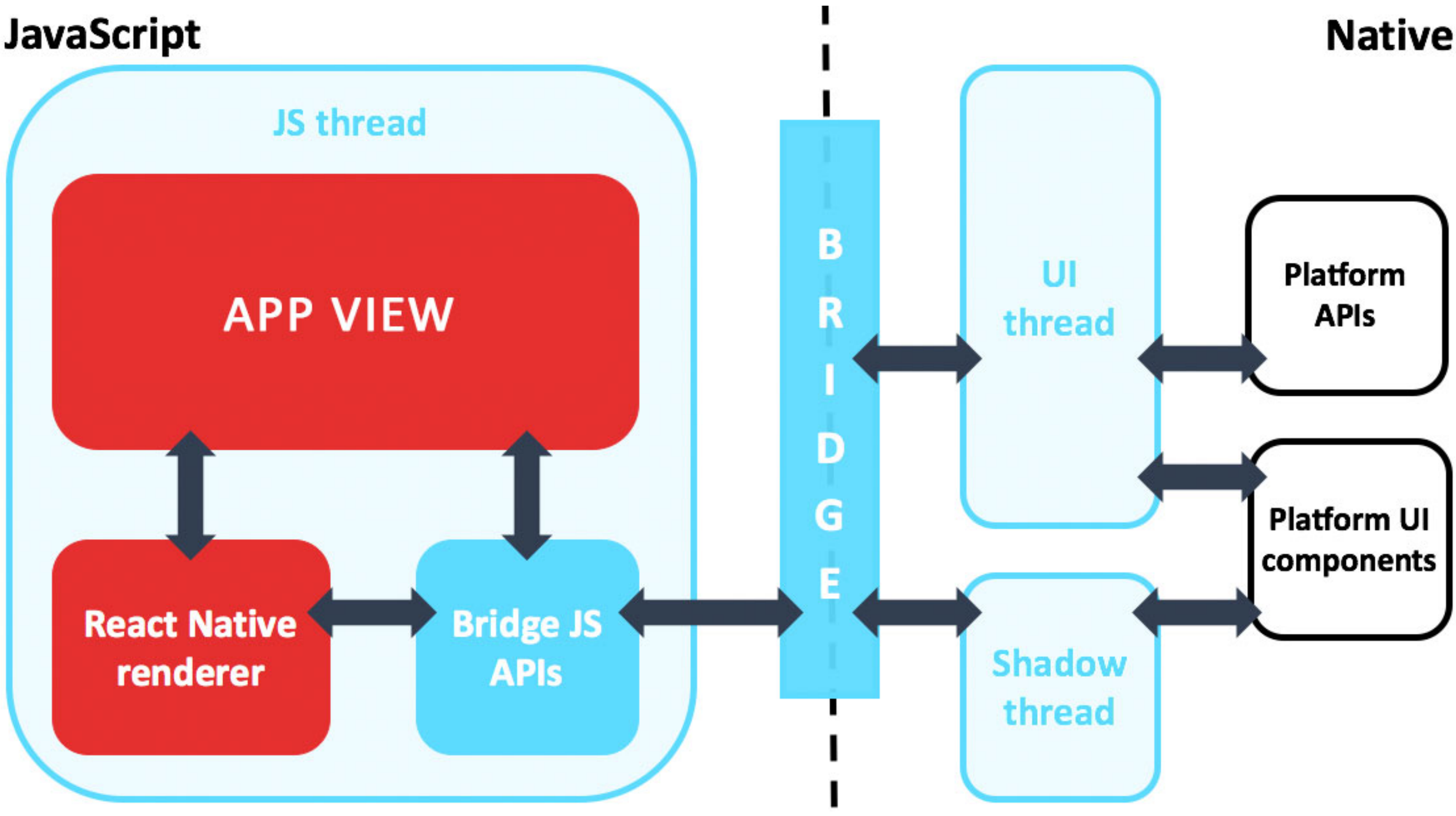
Easy to learn for web developer

- ES6 Standard
- WebDOM concept
- Learn once to write anywhere

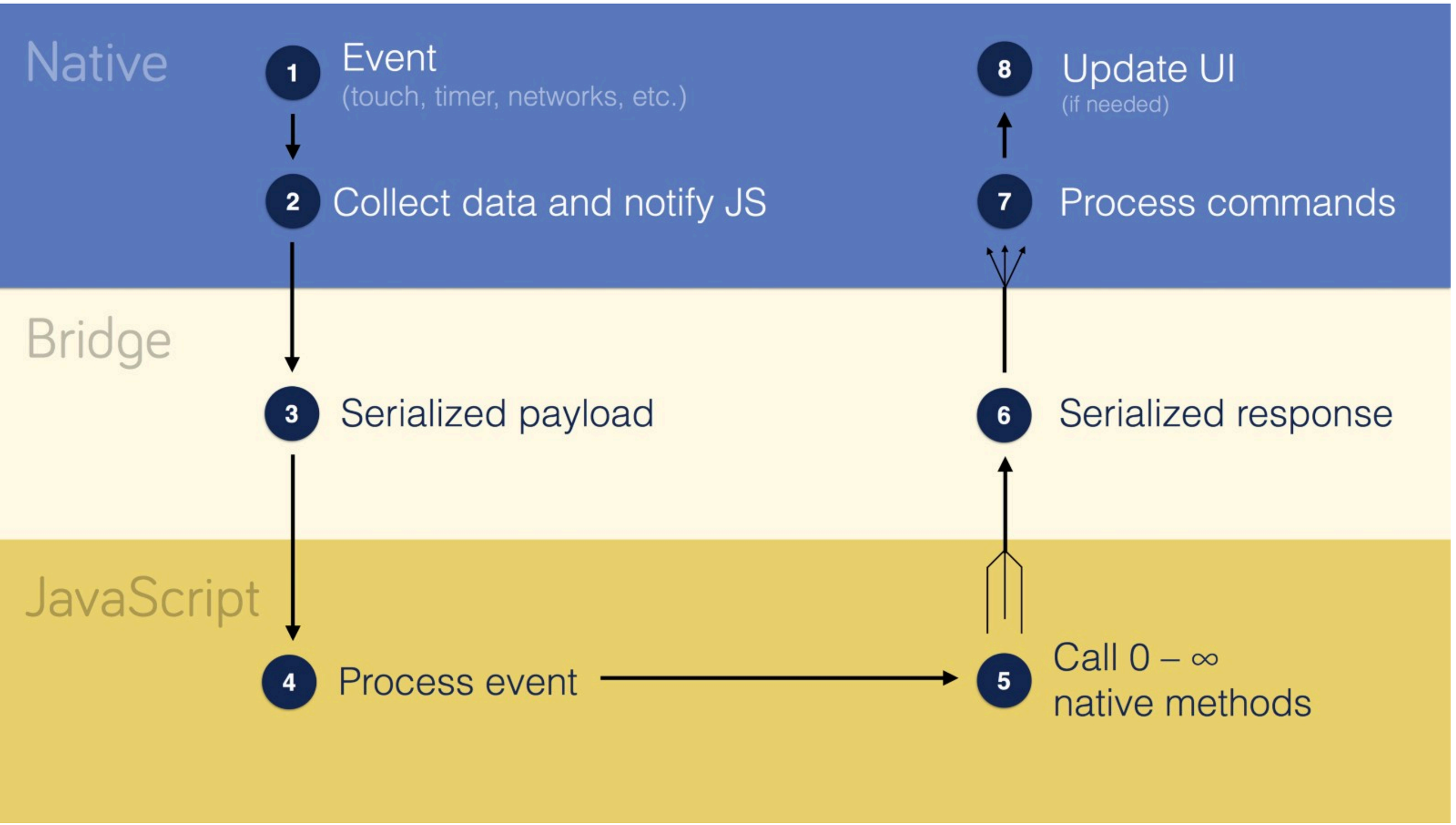


Know your React Native Work

BRIDGE MODULE



BRIDGE MODULE



CODEPUSH by Microsoft

Upload bundle.js to codepush

CodePush

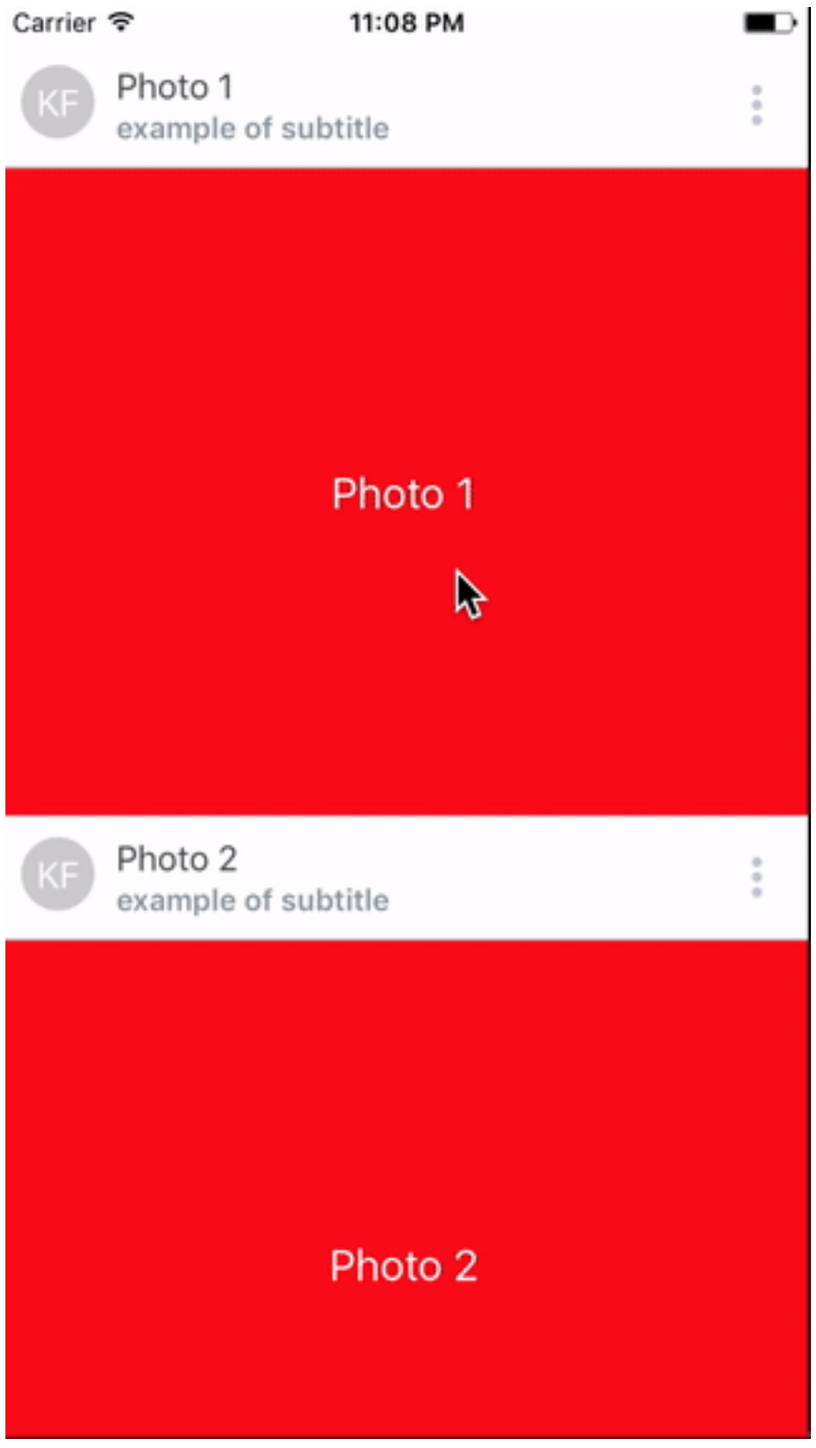
Auto update bundle.js

App build include: apk, ipa, bundle.js

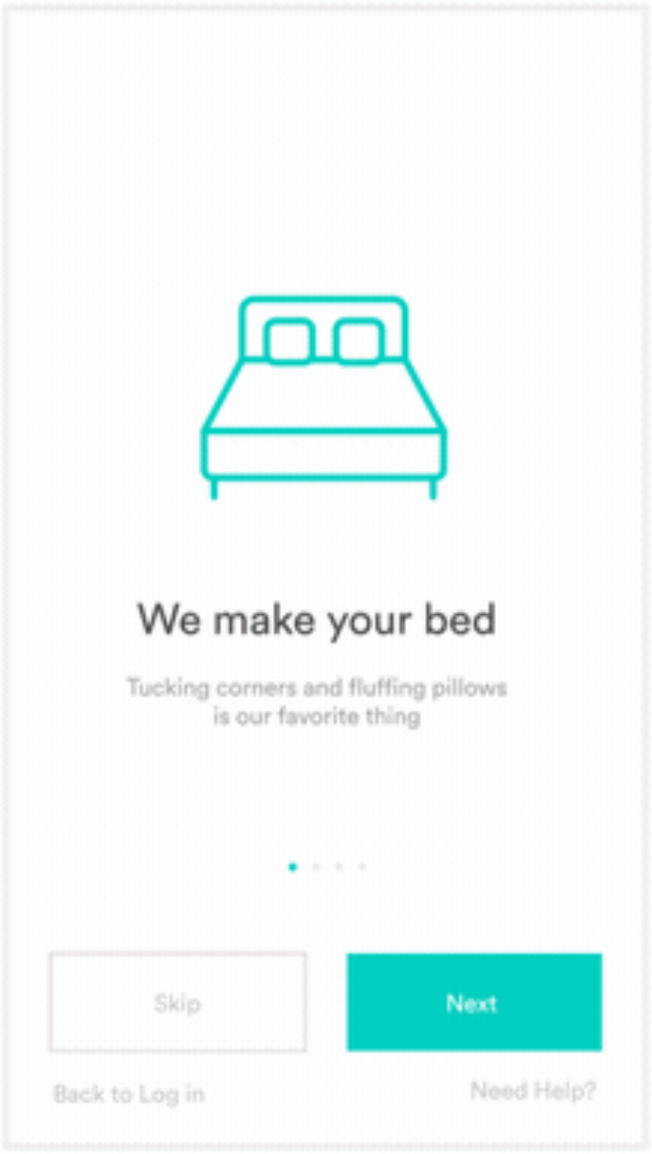
long time review

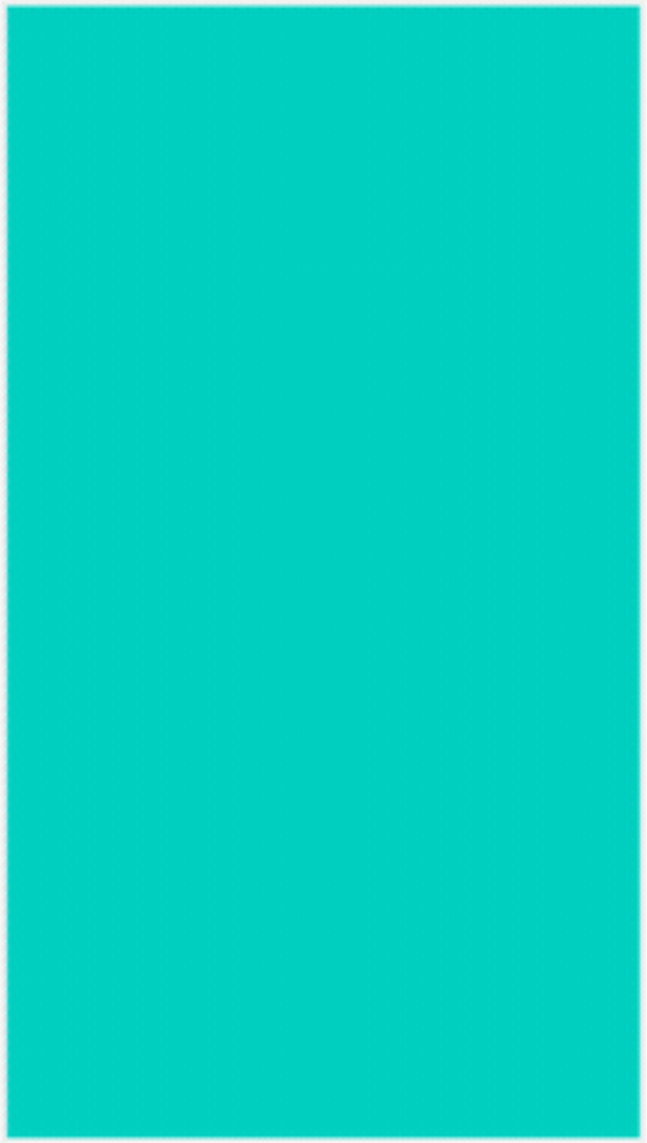
Not auto

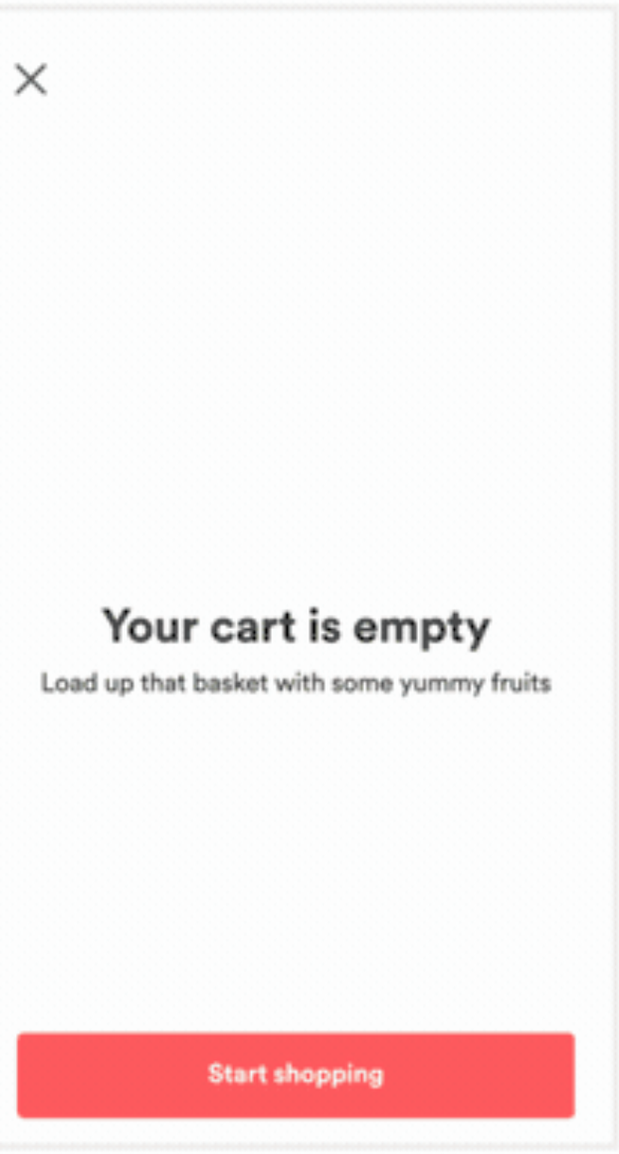
react-native-zoom-draggable-photo



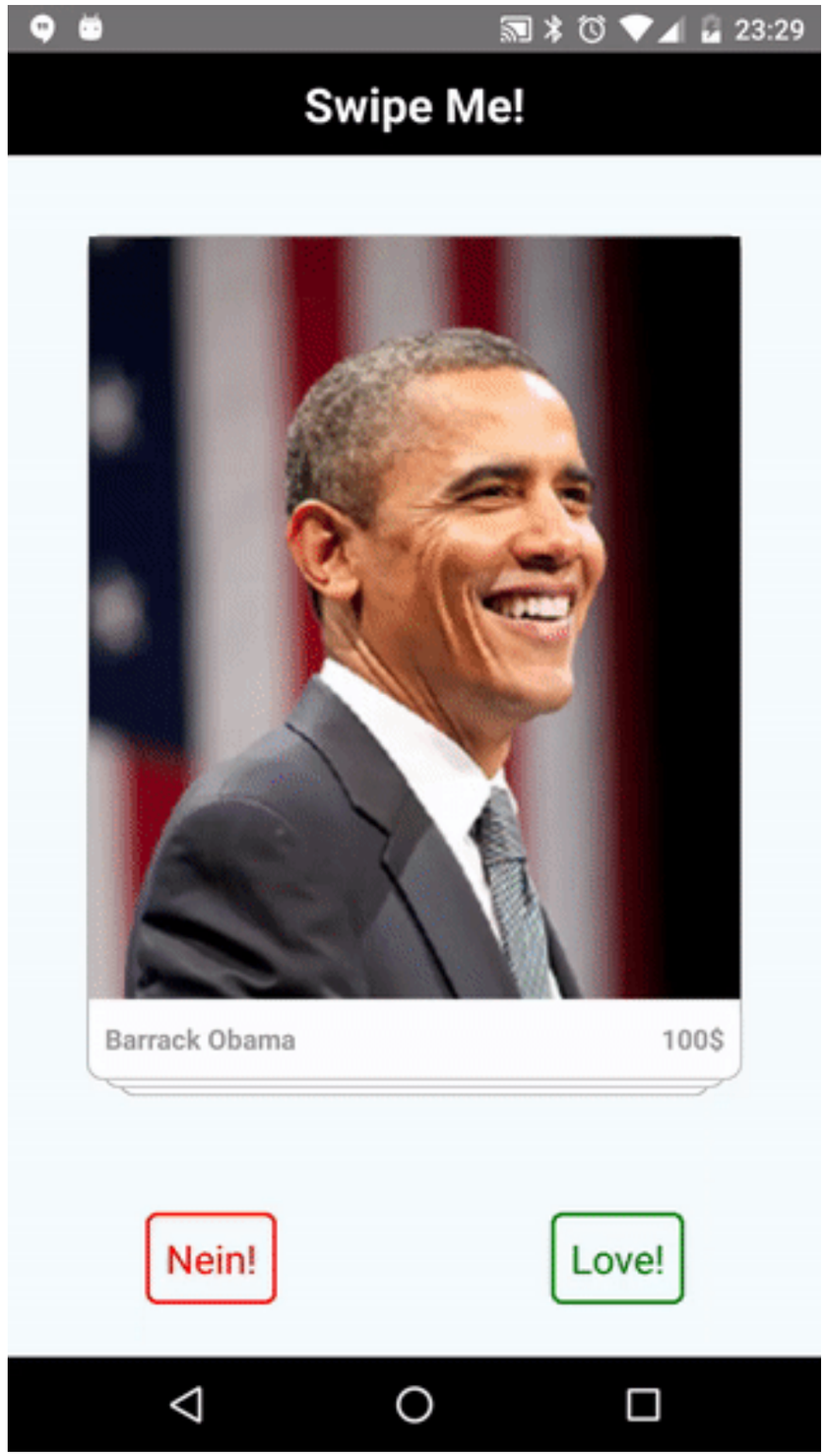
lottie-react-native



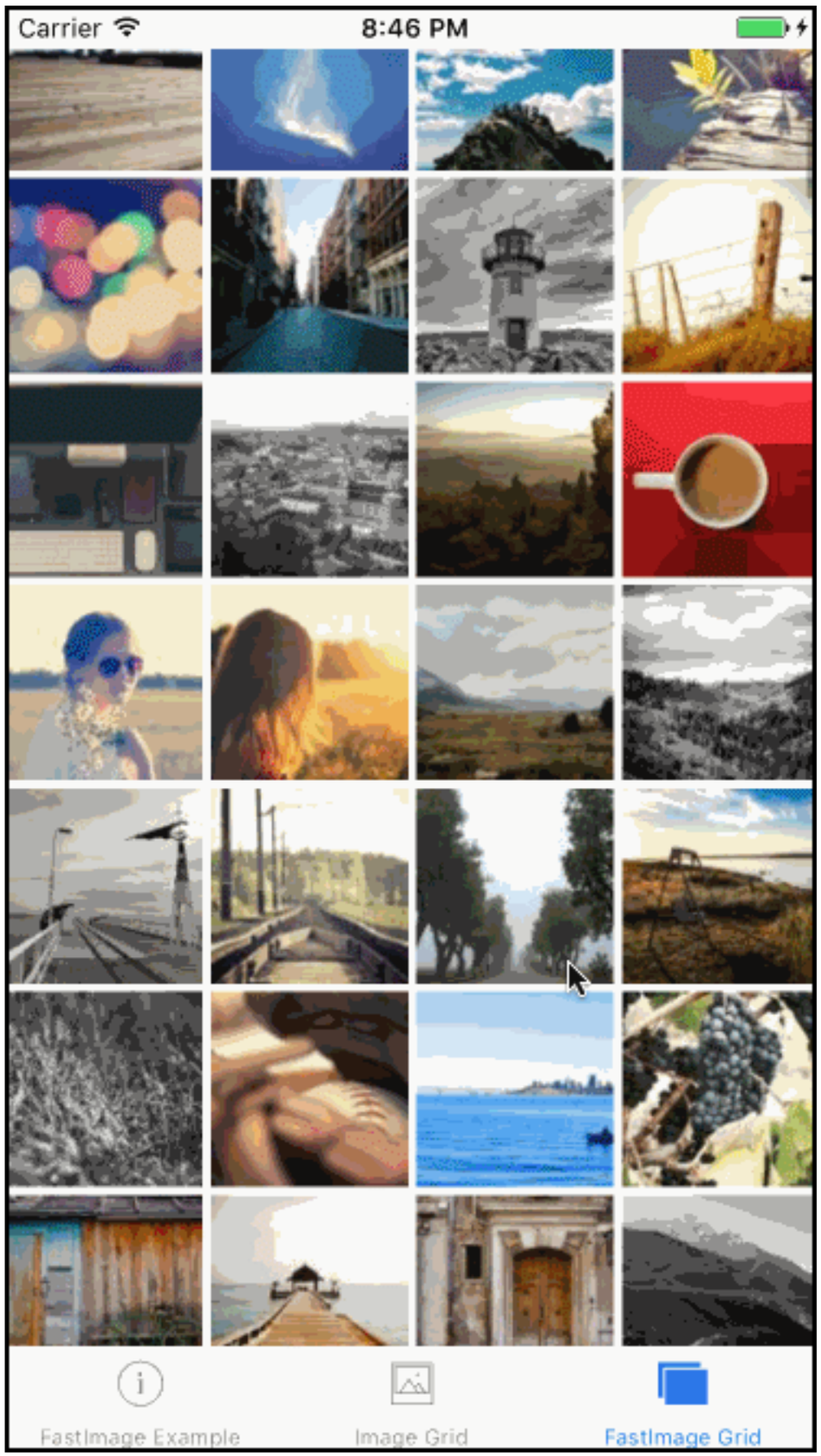


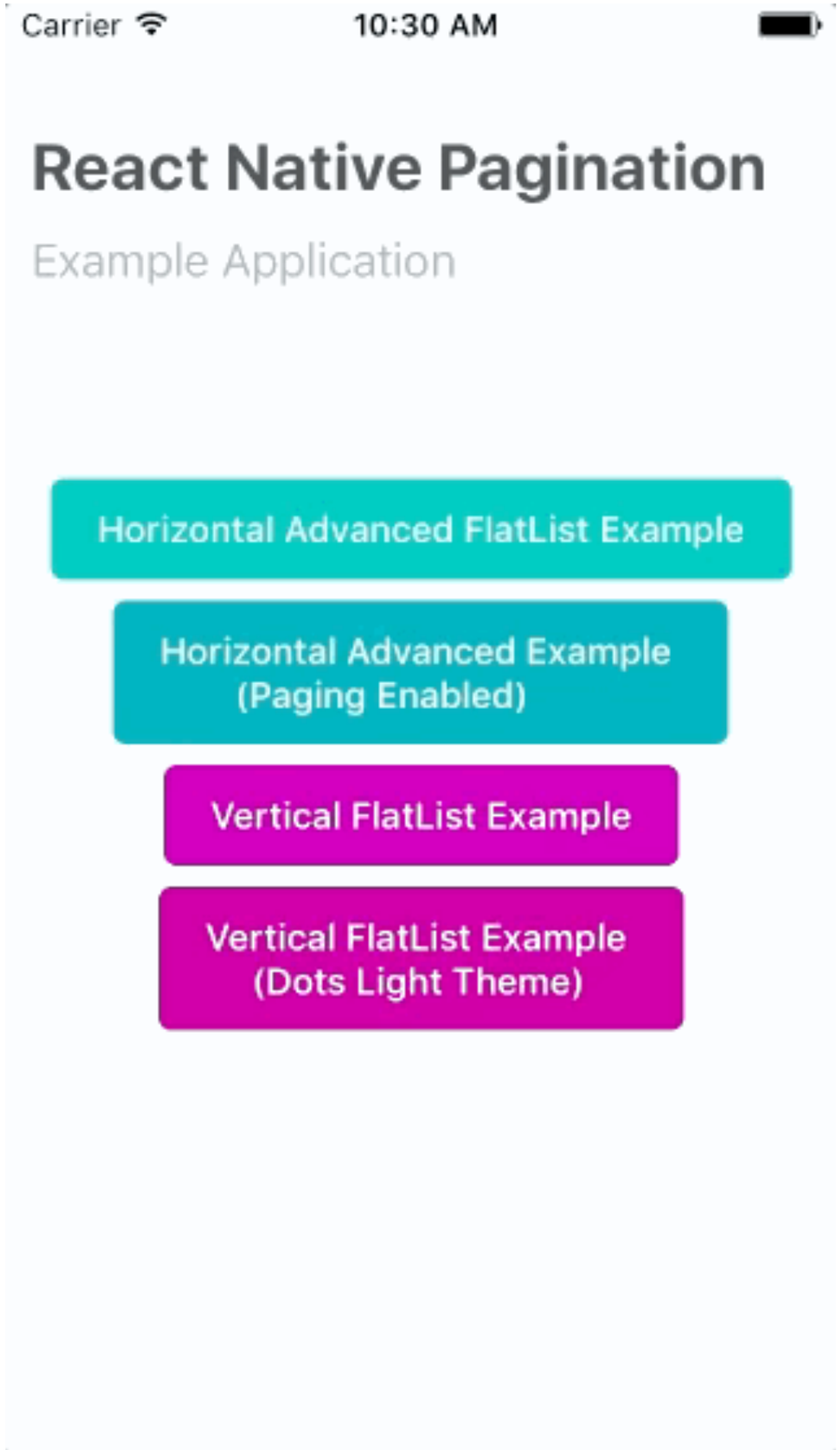


react-native-deck-swiper

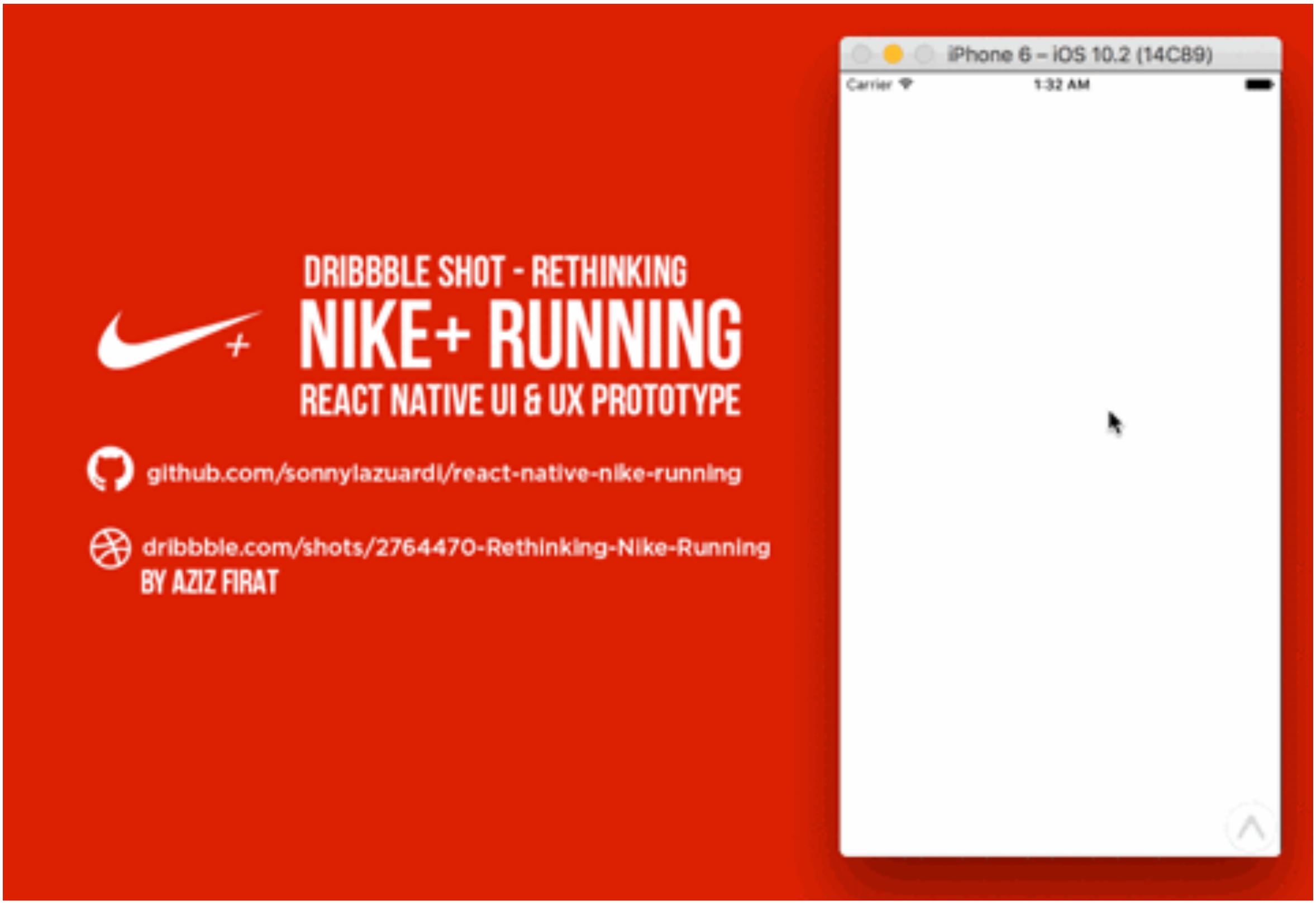


react-native-fast-image

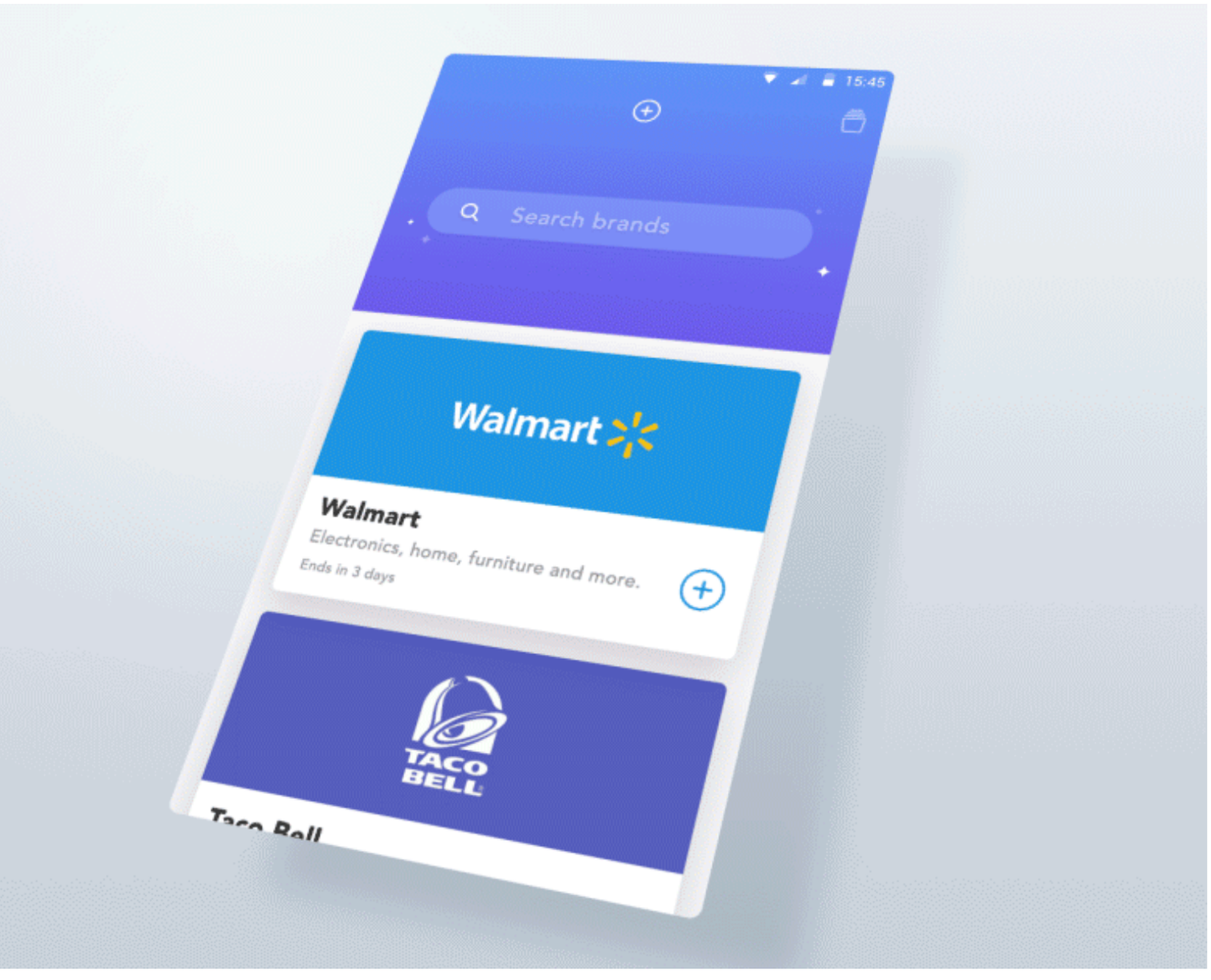




react-native-nike-running

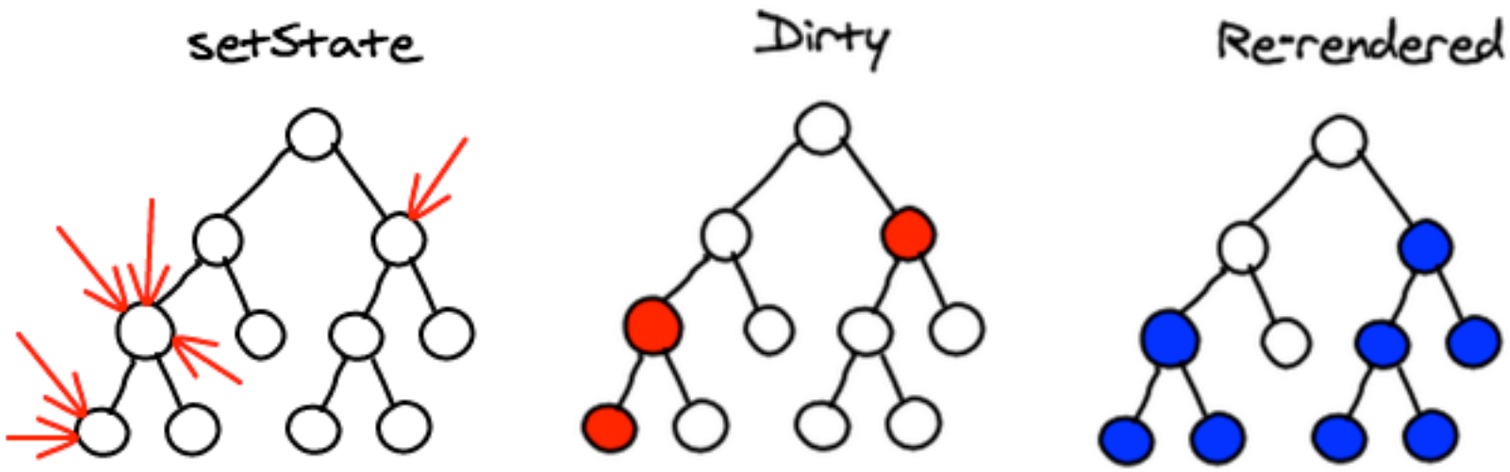


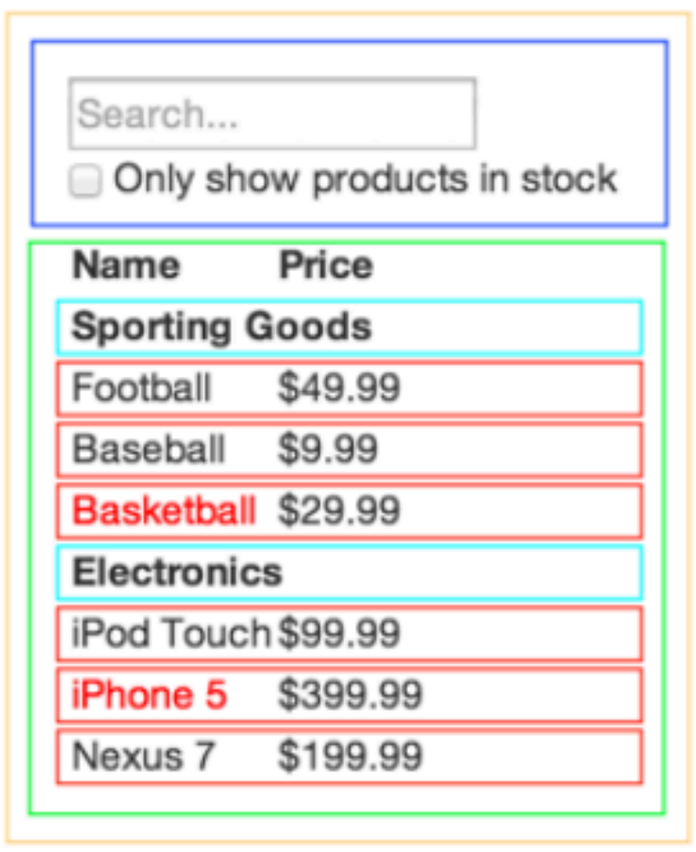
react-native-card-modal

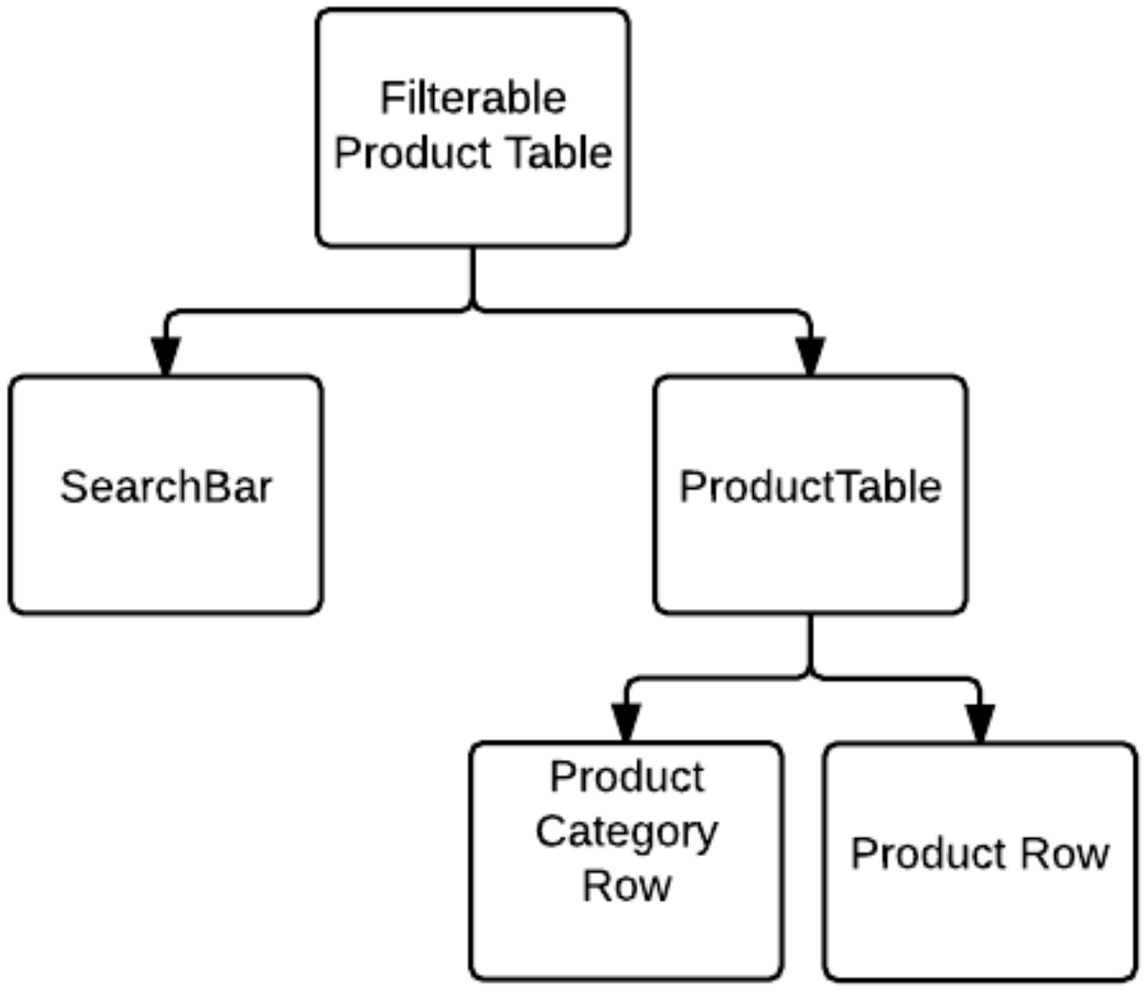


React Concept

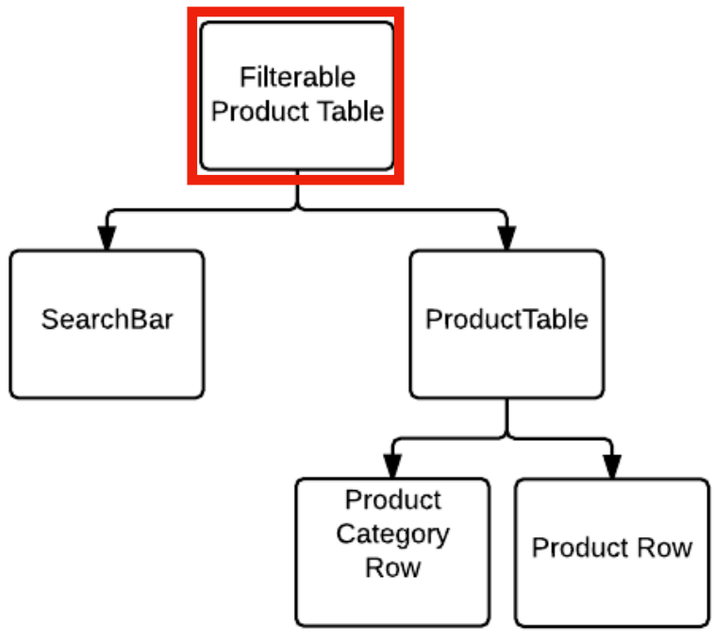
React Concept



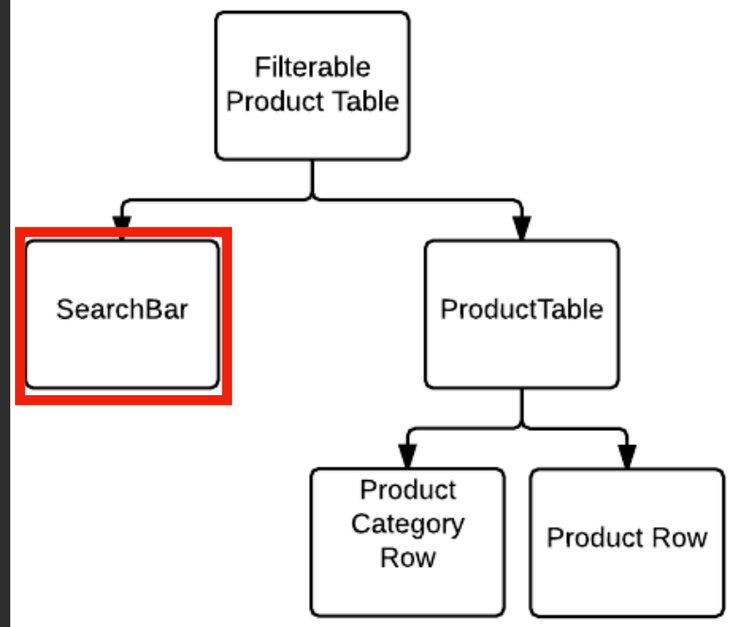




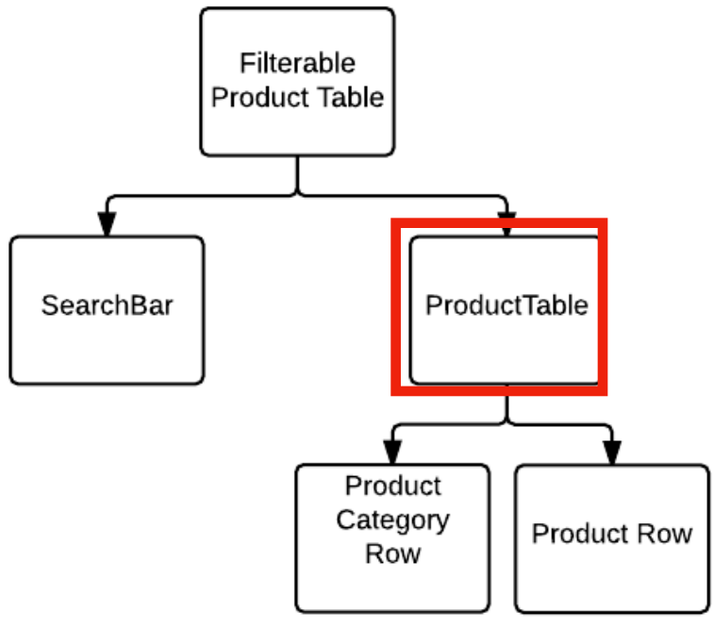
```
class FilterableProductTable extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      filterText: '',
      inStockOnly: false
    };
  }

  handleFilterTextInput(e) {
    this.setState({
      filterText: e.target.value
    });
  }

  handleInStockInput(e) {
    this.setState({
      inStockOnly: e.target.checked
    })
  }

  render() {
    return (
      <div>
        <SearchBar
          filterText={this.state.filterText}
          inStockOnly={this.state.inStockOnly}
          onFilterTextInput={this.handleFilterTextInput.bind(this)}
          onInStockInput={this.handleInStockInput.bind(this)}
        />
        <ProductTable
          products={this.props.products}
          filterText={this.state.filterText}
          inStockOnly={this.state.inStockOnly}
        />
      </div>
    );
  }
}
```



```jsx
class SearchBar extends React.Component {
  static propTypes = {
    filterText: PropTypes.string,
    inStockOnly: PropTypes.bool,
    onChangeFilterTextInput: PropTypes.func,
    onInStockInput: PropTypes.func
  }

  render() {
    return (
      <form>
        <input
          type="text"
          placeholder="Search..."
          value={this.props.filterText}
          onChange={this.props.onFilterTextInput}
        />
        <p>
          <input
            type="checkbox"
            checked={this.props.inStockOnly}
            onChange={this.props.onInStockInput}
            />
          {' '}
          Only show products in stock
        </p>
      </form>
    );
  }
}
```



```
class ProductTable extends React.Component {
  isEmptyProduct(product){
    return
      product.name.indexOf(this.props.filterText) === -1 ||
      (!product.stocked && this.props.inStockOnly)
  }
  renderRows(){
    var rows = [];
    var lastCategory = null;
    this.props.products.forEach((product) => {
      if (this.isEmptyProduct(product)) return;
      if (product.category !== lastCategory) {
        rows.push(
          <ProductCategoryRow
            category={product.category}
            key={product.category}
            />);
      }
      rows.push(
        <ProductRow
          product={product}
          key={product.name}
          />);
      lastCategory = product.category;
    });
    return rows;
  }

  render() {
    return (
      <table>
        <thead>
          <tr>
            <th>Name</th>
            <th>Price</th>
          </tr>
        </thead>
        <tbody>{this.renderRows()}</tbody>
      </table>
    );
  }
}
```



Intro React Native

DIFFERNCE REACT WEB & NATIVE

In React Web use

- In web use , <p>, , and other, … from native DOM

In React Native use

- <View> instead ,<form>
- <TextInput> instead <input />
- <Text> instead , <p>
- <TouchableOpacity> instead <button>

Convert

```
class FilterableProductTable extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      filterText: '',
      inStockOnly: false
    };
  }

  handleFilterTextInput(e) {
    this.setState({
      filterText: e.target.value
    });
  }

  handleInStockInput(e) {
    this.setState({
      inStockOnly: e.target.checked
    })
  }

  render() {
    return (
      <div>
        <SearchBar
          filterText={this.state.filterText}
          inStockOnly={this.state.inStockOnly}
          onFilterTextInput={this.handleFilterTextInput
          onInStockInput={this.handleInStockInput.bind(
        />
        <ProductTable
          products={this.props.products}
          filterText={this.state.filterText}
          inStockOnly={this.state.inStockOnly}
        />
      </div>
    );
  }
}
```

React Web

```
class FilterableProductTable extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      filterText: '',
      inStockOnly: false
    };
  }

  handleFilterTextInput(e) {
    this.setState({
      filterText: e.target.value
    });
  }

  handleInStockInput(e) {
    this.setState({
      inStockOnly: e.target.checked
    })
  }

  render() {
    return (
      <View>
        <SearchBar
          filterText={this.state.filterText}
          inStockOnly={this.state.inStockOnly}
          onFilterTextInput={this.handleFilterTextInput.bind(this)}
          onInStockInput={this.handleInStockInput.bind(this)}
        />
        <ProductTable
          products={this.props.products}
          filterText={this.state.filterText}
          inStockOnly={this.state.inStockOnly}
        />
      </View>
    );
  }
}
```

React Native

Convert

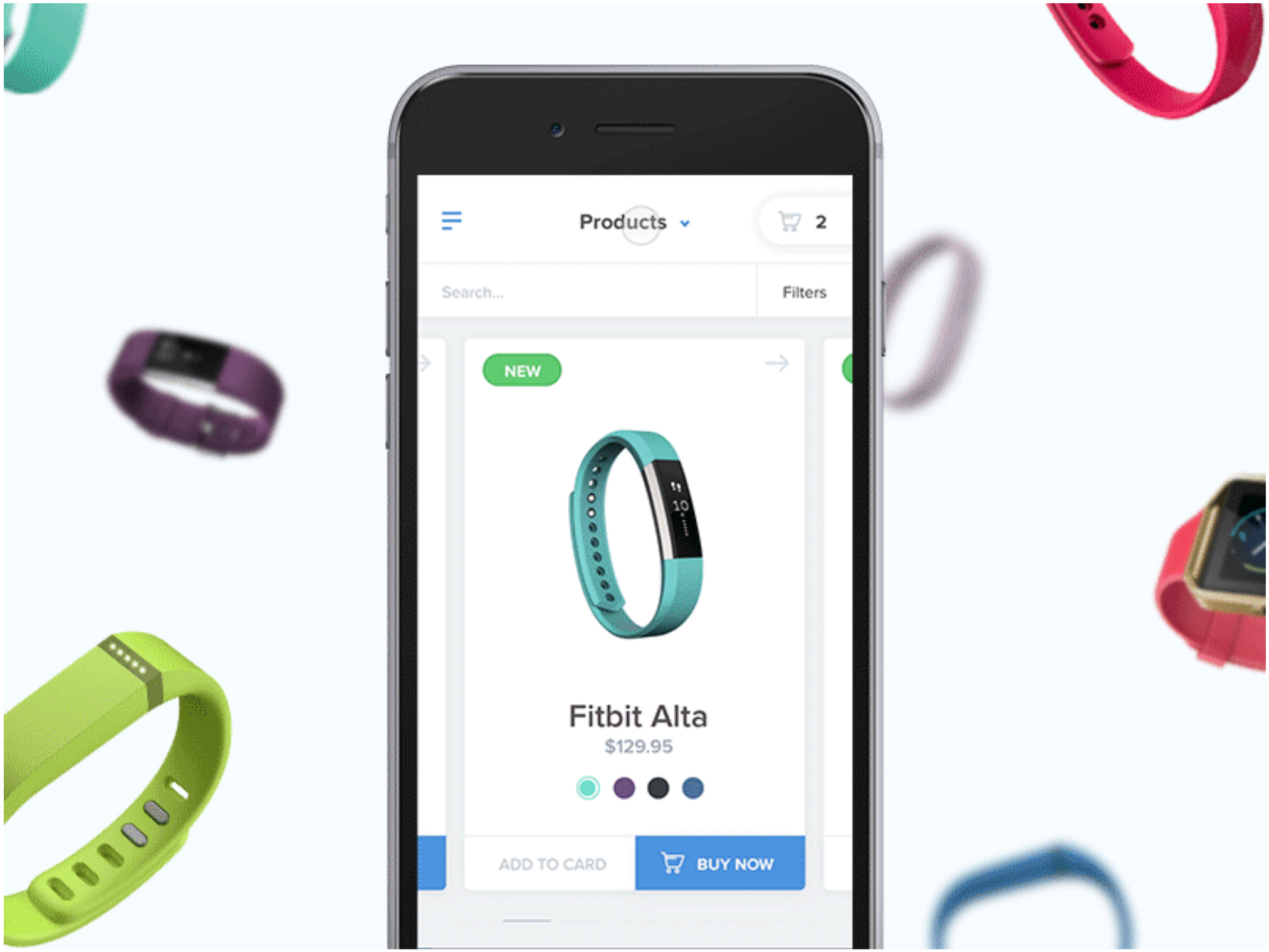
```jsx
class SearchBar extends React.Component {
  static propTypes = {
    filterText: PropTypes.string,
    inStockOnly: PropTypes.bool,
    onChangeFilterTextInput: PropTypes.func,
    onInStockInput: PropTypes.func
  }

  render() {
    return (
      <form>
        <input
          type="text"
          placeholder="Search..."
          value={this.props.filterText}
          onChange={this.props.onFilterTextInput}
        />
        <p>
          <input
            type="checkbox"
            checked={this.props.inStockOnly}
            onChange={this.props.onInStockInput}
            />
          {' '}
          Only show products in stock
        </p>
      </form>
    );
  }
}
```

```jsx
class SearchBar extends React.Component {
  static propTypes = {
    filterText: PropTypes.string,
    inStockOnly: PropTypes.bool,
    onChangeFilterTextInput: PropTypes.func,
    onInStockInput: PropTypes.func
  }

  render() {
    return (
      <View>
        <TextInput
          placeholder="Search..."
          value={this.props.filterText}
          onChangeText={this.props.onFilterTextInput}
        />
        <View>
          <Toggle
            checked={this.props.inStockOnly}
            onChange={this.props.onInStockInput}
            />
          <Text>
            Only show products in stock
          </Text>
        </View>
      </View>
    );
  }
}
```

React Web

React Native

mobile-shopping-react-native



Other react native modules

In React Web use

- PansResponder, Image, FlatList, ListView

Access Device feature

- Camera
- Map
- GeoLocation
- Image Process
- Notification, Permission, Networking, …

How to learn

- Learning ES6
- React Component, State & Props
- React Component Life Cycle
- React with Networking fetch data
- React Native Style Flex
- Redux Store Management

Guide Document

- reactnativeexpress.com
- react.express
- https://github.com/jondot/awesome-react-native

Original Document

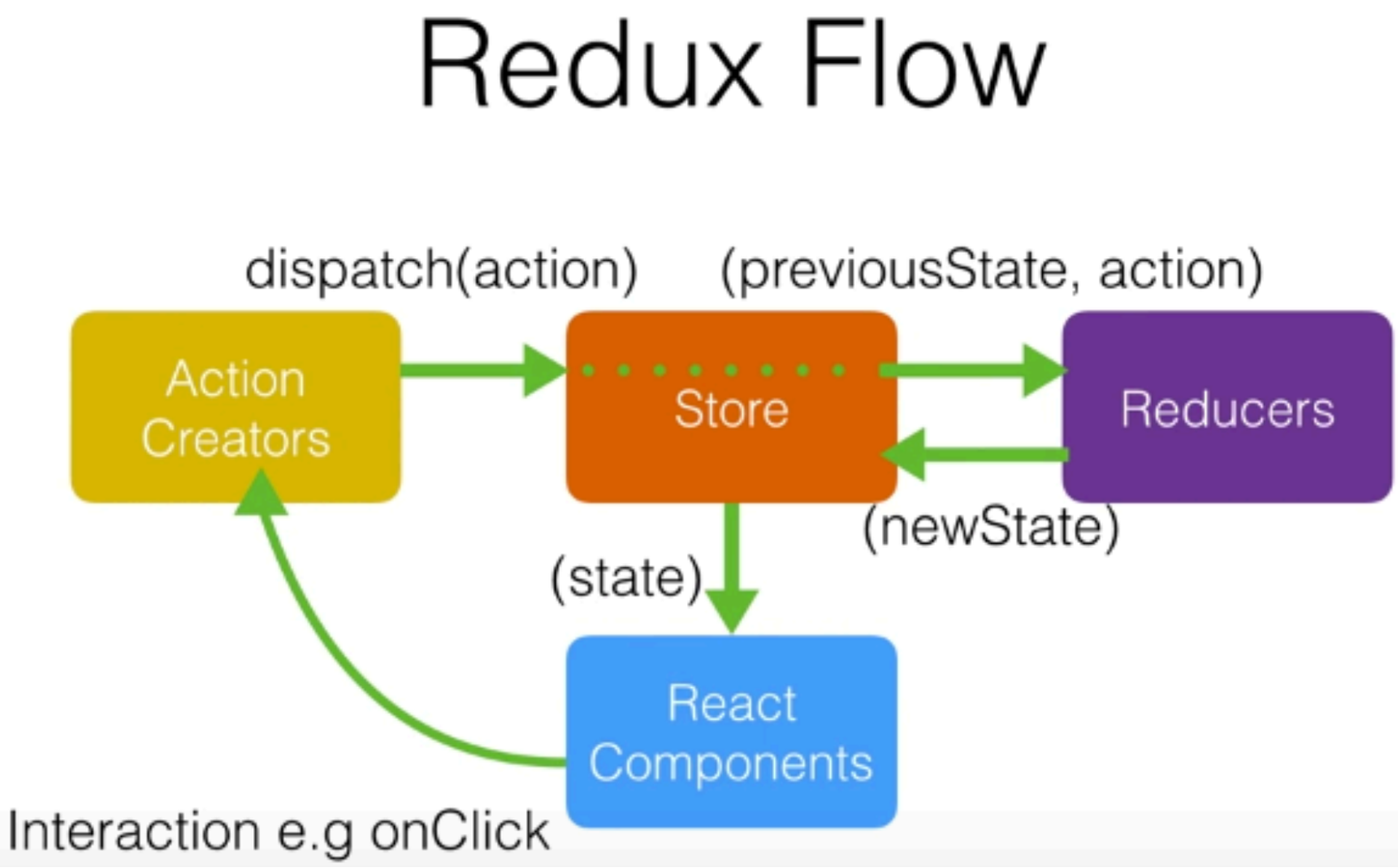
- https://facebook.github.io/react-native/
- https://facebook.github.io/react/

From zero to master

- https://github.com/thanhtungdp/learn-react-how-to

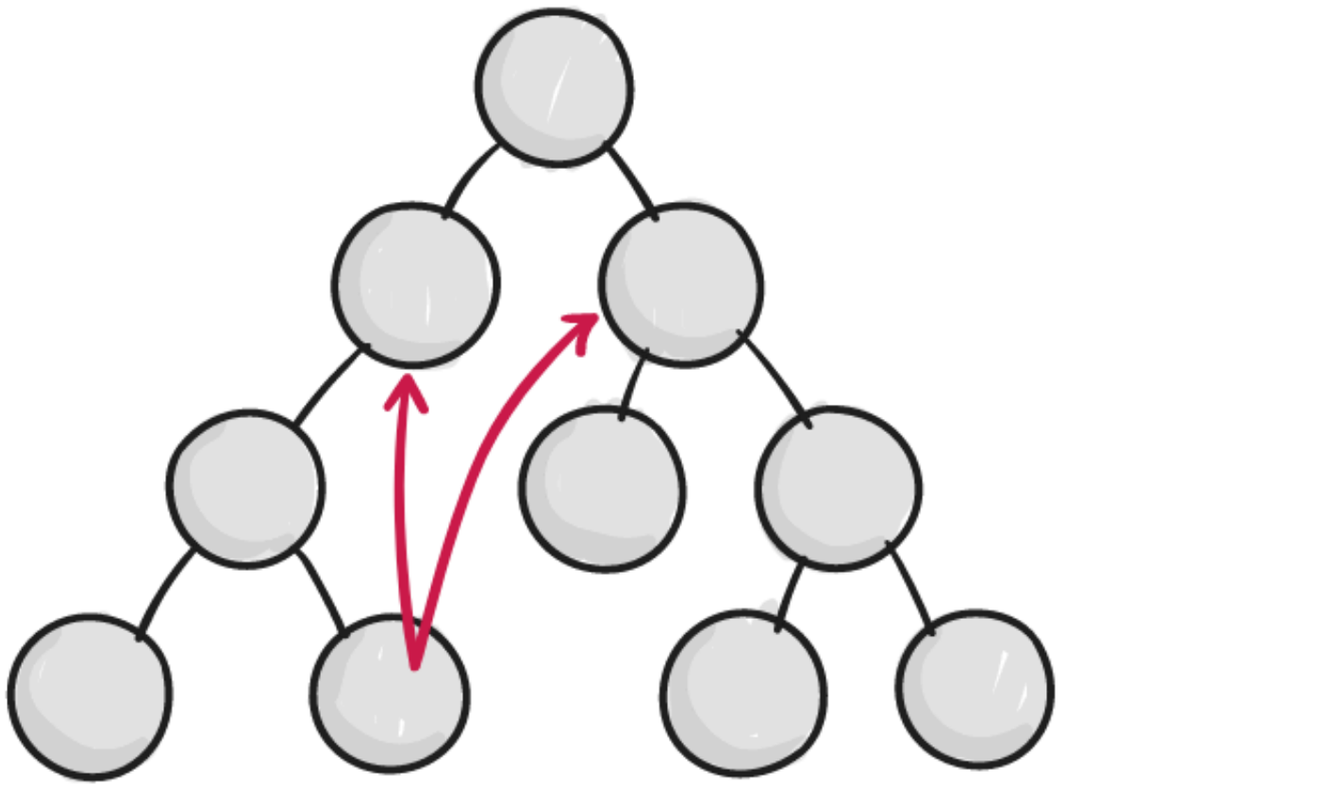
Intro Redux

- Redux is best practice for store management
- Can be shared code between native and web



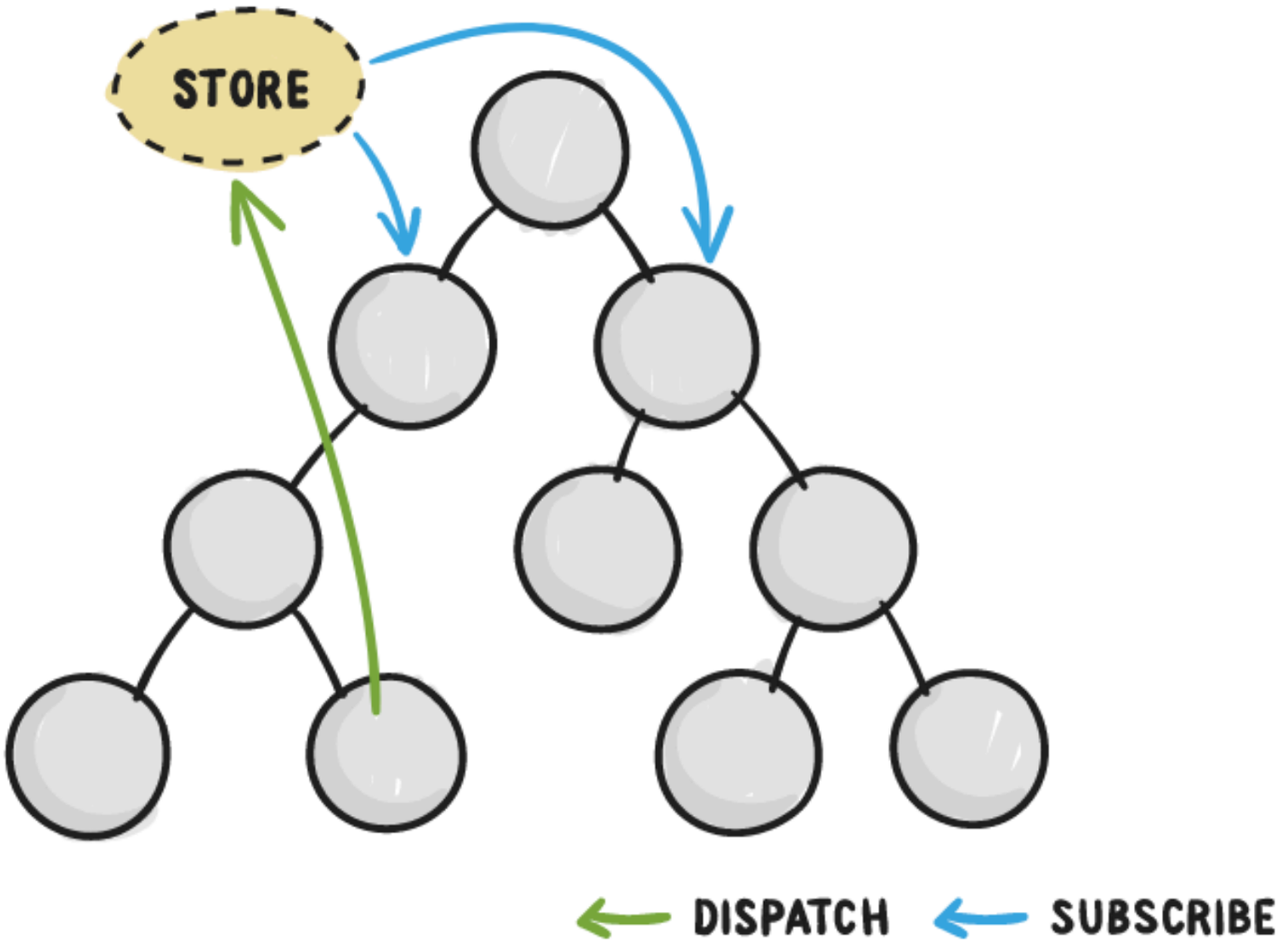
State Problem

- App like tree component
- Root State need pass to multiple component via props

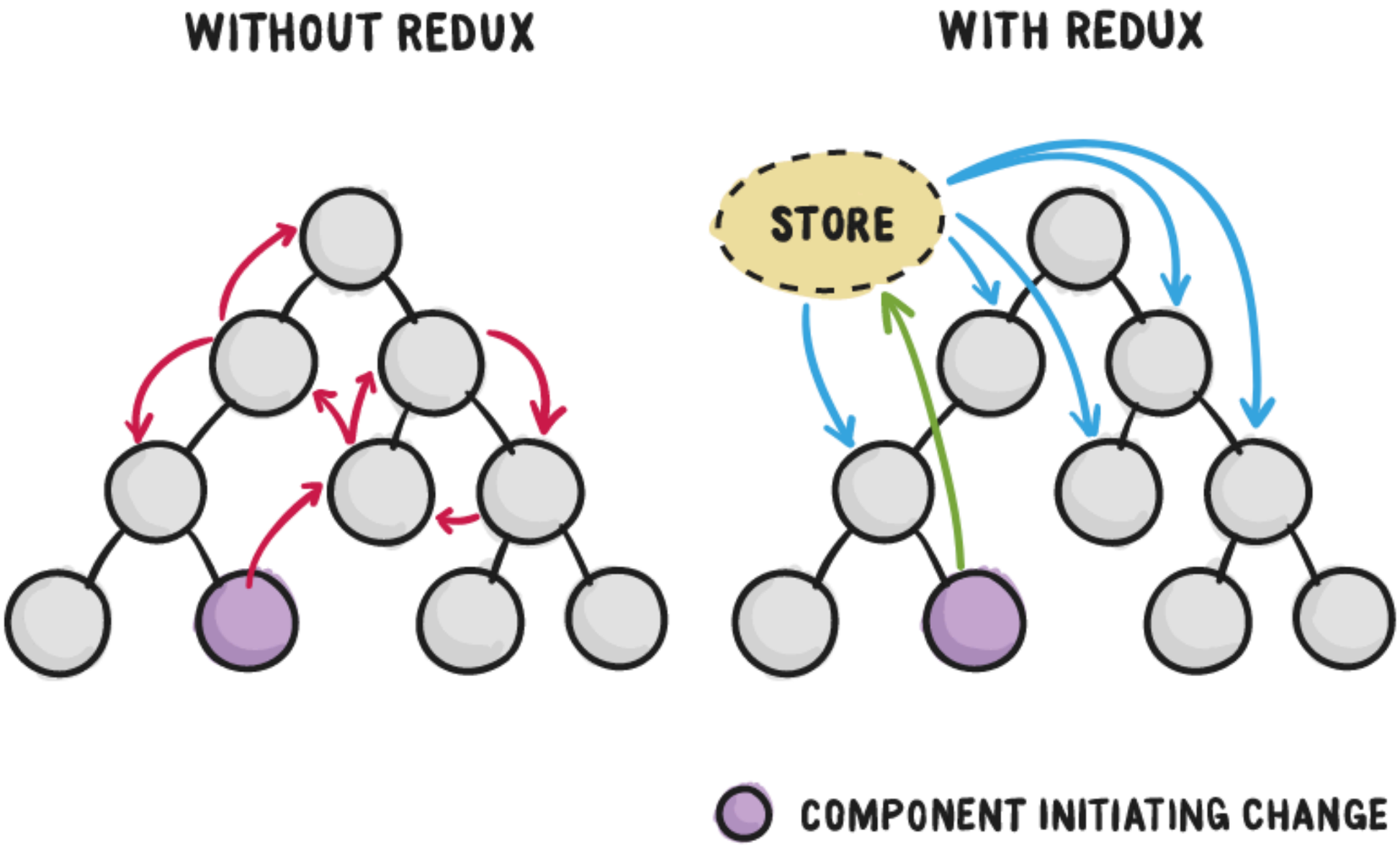


Redux Solution

- Store is single
- Can be pass store in component everywhere



Compare


WITHOUT REDUX
WITH REDUX
STORE
COMPONENT INITIATING CHANGE

Concept document

- https://code-cartoons.com/a-cartoon-intro-to-redux-3afb775501a6
- http://reduxjs.org

Real Project

- https://medium.com/react-native-training/redux-4-ways-95a130da0cdc

Best channel learning

- https://medium.com/react-native-training
- http://makeitopen.com/
- http://tiny.cc/reactnative-learn
- https://github.com/thanhtungdp/learn-react-how-to

Showcase


TUNGTUNG.VN
ÔN THI THPT QUỐC GIA 2017
Ôn thi THPT Quốc gia 2017
ngay trên điện thoại di động
Android của bạn
ĐÃ ÔN LÀ PHẢI ĐẬU ĐẠI HỌC
TÀI TRỢ BỞI
facebook.
http://tungtung.vn/app
tungtung ôn thi
Ôn thi THPT Quốc gia -Tùng..
tungtung co
5.0
INSTALLED

React Ecosystem by Javascript only

Frontend

Web

React Native

Ios/android app

flow

Type checker

State Management

Mobx

Redux

Query language

GraphQL

```
mutation($student: StudentInputType!) {
  createStudent(student: $student) {
    id,
    name,
    class_id
  }
}
```

Server Language

node.js

Express

StrongLoop

Ruby

Database

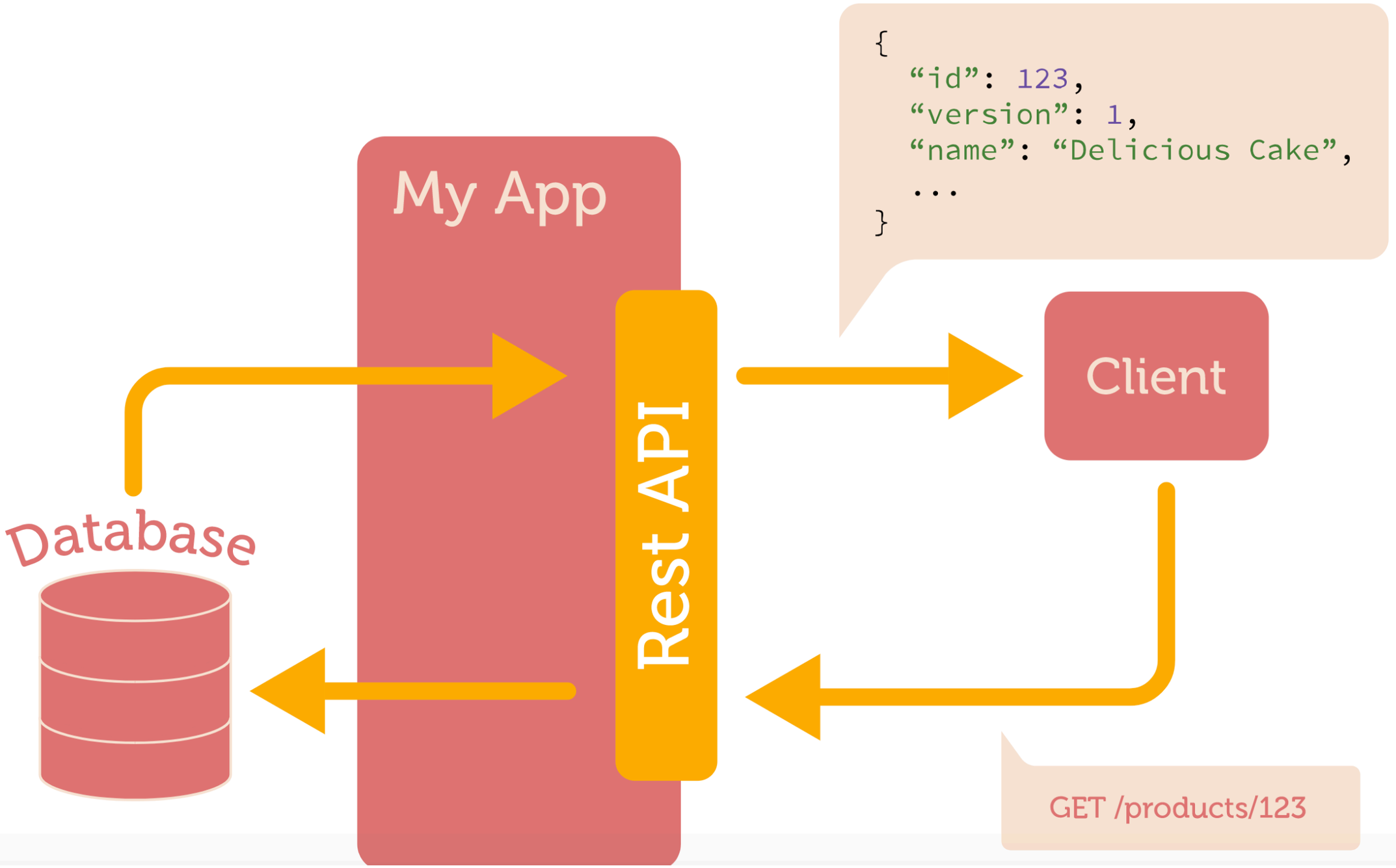
mongoDB

MySQL

PostgreSQL

GraphQL Short Intro

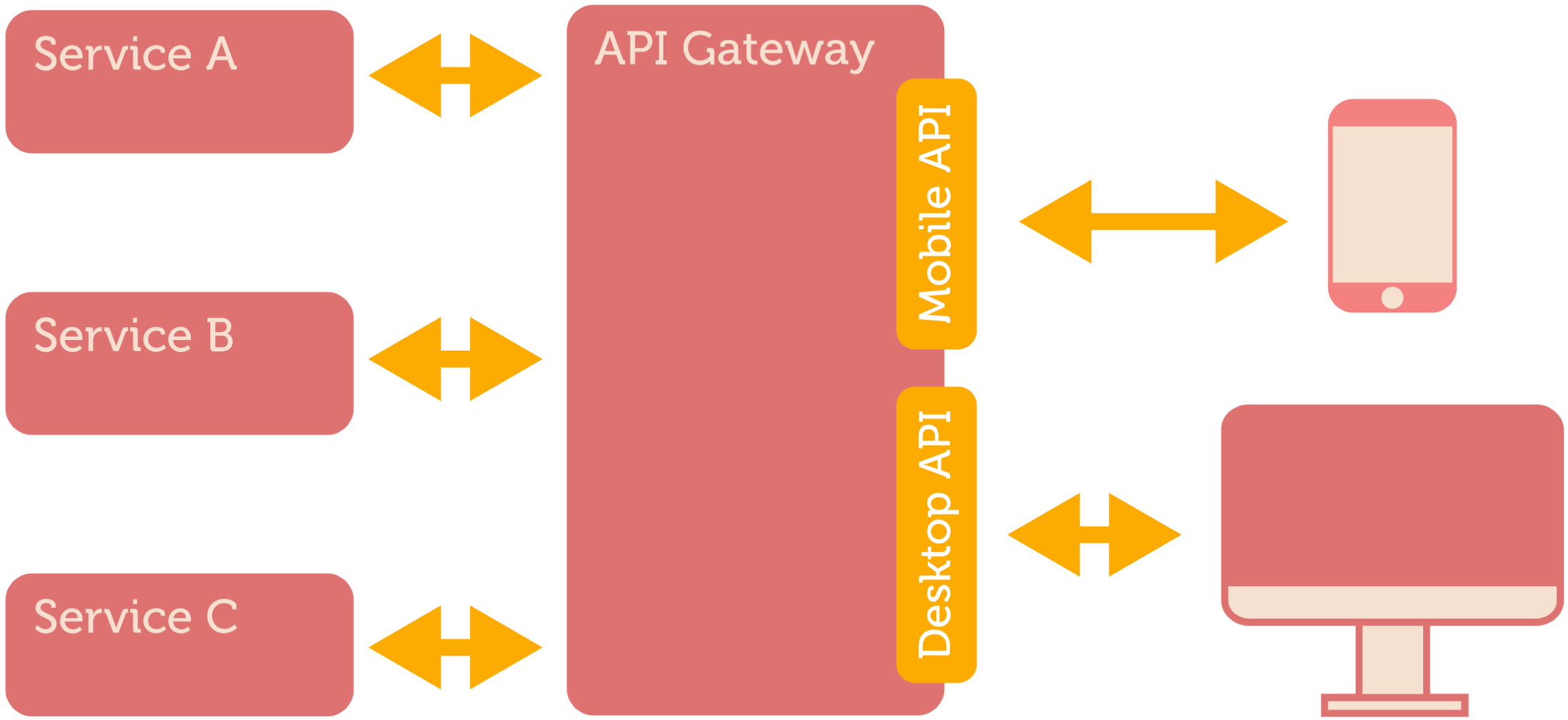
Typical Rest API



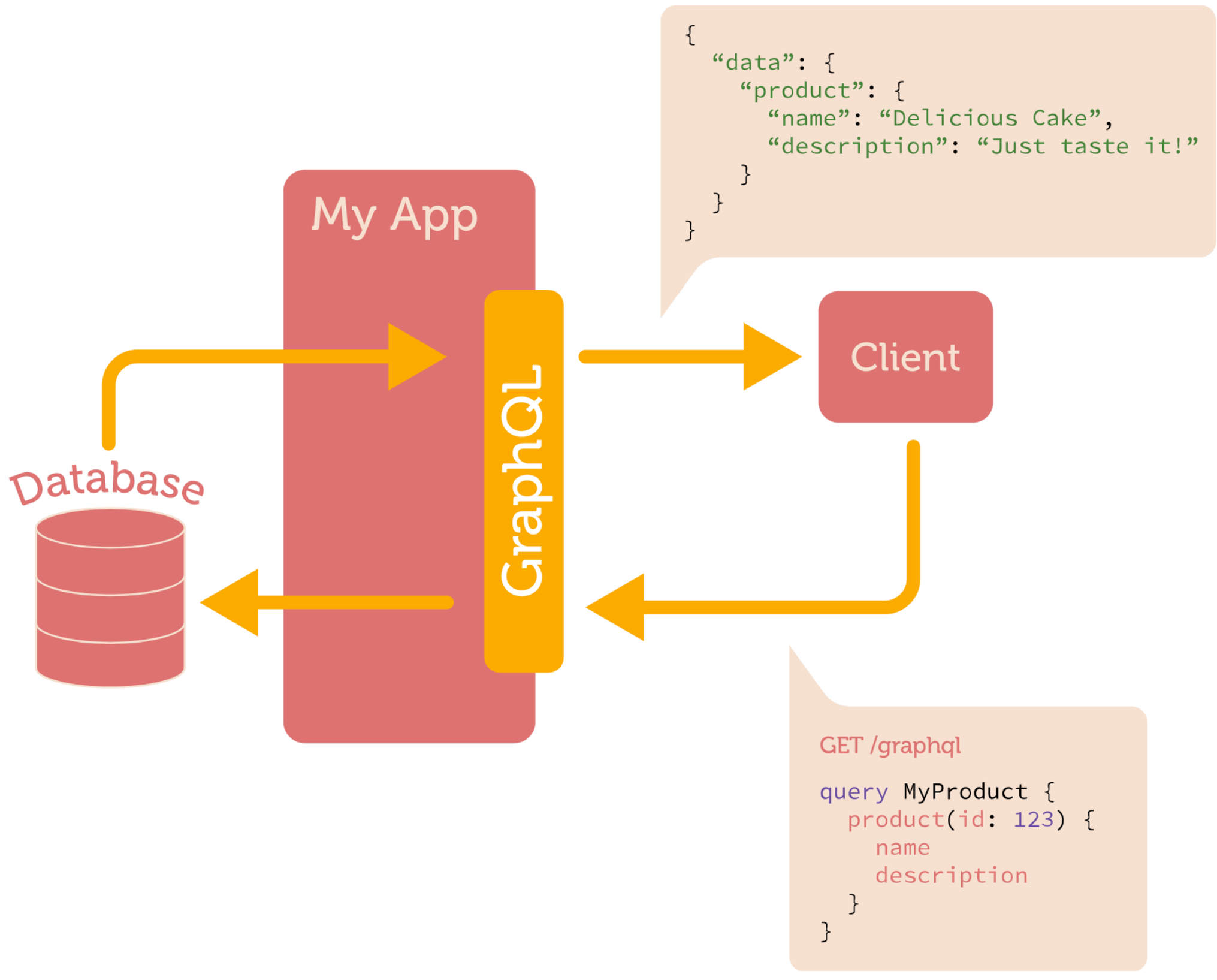
Common issues

- Over-fetching
  - /products?
    field=name&field=description&field=variants[*].price
- Under-fetching
  - /products?
    expand=productType&expand=variants[*].price.tax
- API changes and evolution
  - Versioning
  - Deprecation
  - Maintaince

API GATEWAYS



GraphQL Approach



GraphQL Response Structure

```
query MyProduct {
  product(id: 123) {
    name
    description

    picture {
      width
      height
      url
    }
  }
}
```

```
{
  "data": {
    "product": {
      "name": "Delicious Cake",
      "description": "Just taste it!"

      "picture": {
        "width": 150,
        "height": 150,
        "url": "http://..."
      }
    }
  }
}
```

GraphQL every field is function

```
query MyProduct {
  products {
    thumb: picture(size: 100) {
      width
    }
    fullSize: picture(size: 500) {
      width
    }
  }
}
```

```
{
  "data": {
    "products": [
      {
        "thumb": {
          "width": 100
        },
        "fullSize": {
          "width": 500
        }
      },
      ...
    ]
  }
}
```

Type System

```
type Picture {
  width: Int!
  height: Int!
  url: String
}
```

```
interface Identifiable {
  id: String!
}
```

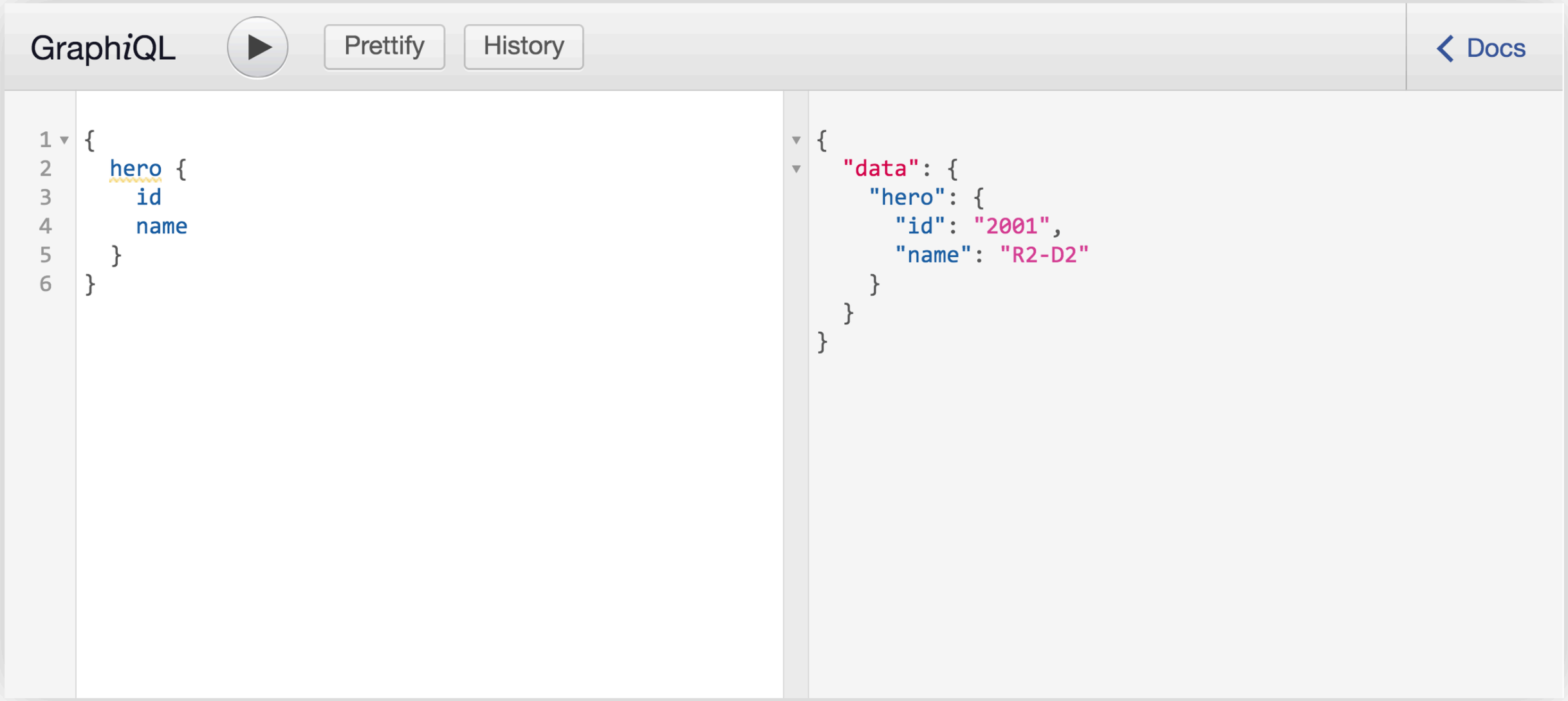
```
type Product implements Identifiable {
  id: String!
  name: String!
  description: String
  picture(size: Int): Picture
}
```

```
type Query {
  product(id: Int!): Product
  products: [Product]
}
```

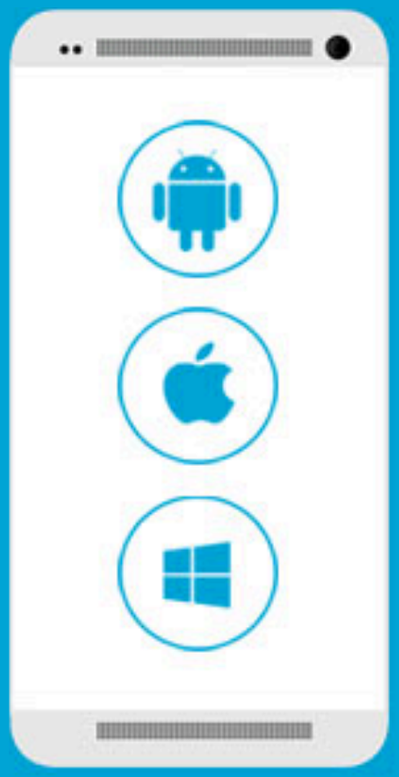
Mutation & Subscriptions

```
mutation ChangeStaff {
  changeName(productId: 123, newName: "Cheesecake") {
    id, version
  }

  setDescription(productId: 123, description: "Delicious!") {
    id, version
  }
}

subscription ProductEvents {
  nameChanged(productId: 123) { name }
  productDeleted { id }
}
```

GraphQL demo



FAQ’S



React Native

A framework for build native app using Javascript and React

Speaker: Phan Thanh Tung - CoFounder of tungtung.vn
Email: tungptkh@gmail.com
Github: github.com/thanhtungdp