

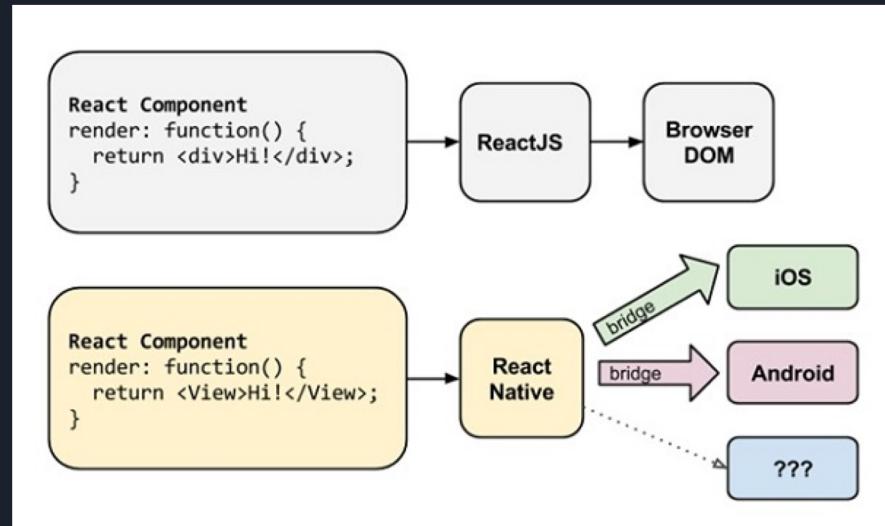


React Native in Production

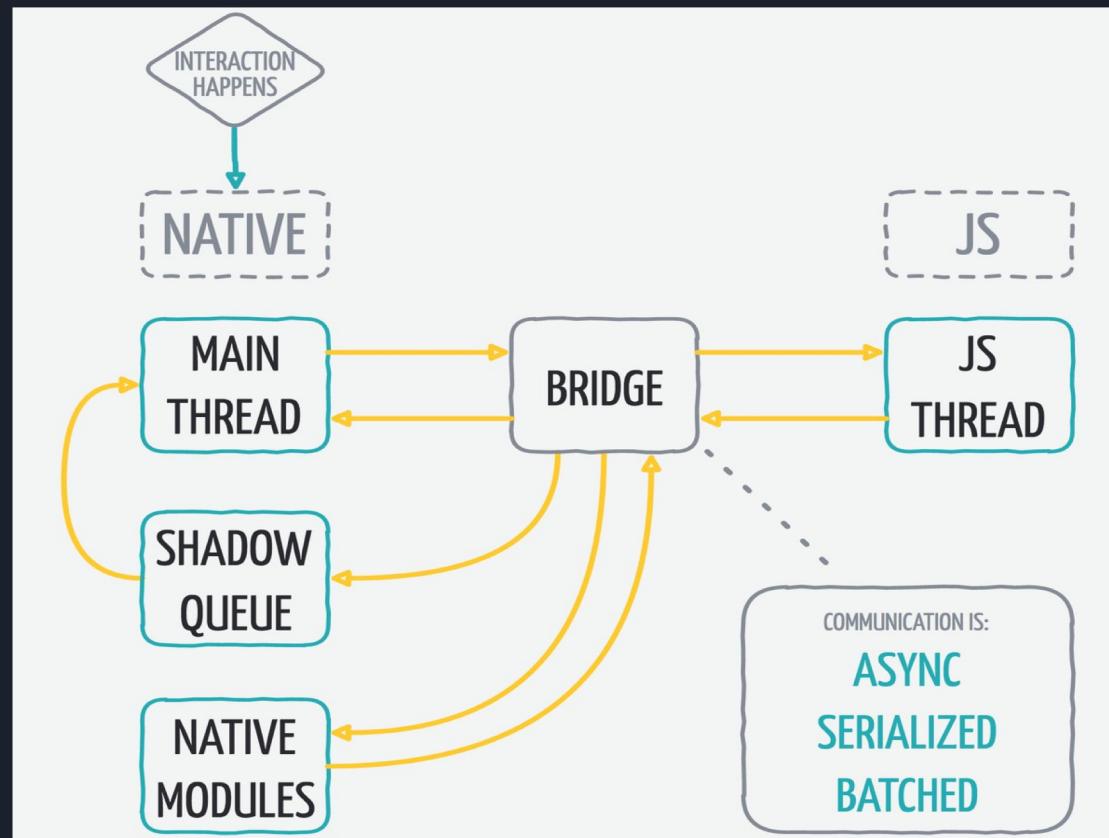
Mauro Jurinčić, Leo Brdar, Filip
Stojanac, Tomislav Raguž

What is React Native?

- Framework for building Cross-Platform Native Apps using React
- Binds JSX to
 - iOS Touch
 - Android UI
- Comprises a suite of technologies
- Near full native speed
- MIT Licensed (since [Feb 17, 2018](#))



High level overview



React

```
import * as React from 'react';

export default class AllOrders extends React.Component {
  renderOrder = ({ id, createdAt, status }) => (
    <div className="single-order" key={id}>
      <h2>Order #{id}</h2>
      <p>Order date: {createdAt}</p>
      <p>Status: {status}</p>
    </div>
  );
  render() {
    const { orders } = this.props;
    return (
      <div>
        { !orders.length && orders.map(this.renderOrder)}
      </div>
    );
  }
}
```

React Native

```
import * as React from 'react';
import { View, Text, StyleSheet } from 'react-native';

export default class AllOrders extends React.Component {
  renderOrder = ({ id, createdAt, status }) => (
    <View style={styles.singleOrder} key={id}>
      <Text>Order #{id}</Text>
      <Text>Order date: {createdAt}</Text>
      <Text>Status: {status}</Text>
    </View>
  );
  render() {
    const { orders } = this.props;
    return (
      <View>
        { !orders.length && orders.map(this.renderOrder)}
      </View>
    );
  }
}
```



Technology stack

Need to get familiar with

- JavaScript (ES6 flavour and up)
- JSX (merger of ES and HTML)
- CSS/FlexBox
- npm/yarn
- Chrome Debugger
- Xcode / Android Studio
- React JS
- React Native

Good to know

- NodeJS
- Flow
- Redux
- Jest



Coming from the browser

- Mobile navigation conventions
- Setting up the environment
- Basic knowledge of target platforms and tools required
- Responding to soft keyboard
- Various exposed APIs
- Publishing and updating apps
- Styling not as powerful or reliable, no global styling



Why did we choose it?

- Small team of web (JavaScript) developers
- Constraints on resources/time
- Limited native app expertise
- Needed iOS app, but trying to stay ahead of future client needs
- API-driven mobile client (heavy lifting was done on the API)
- Reuse what we know
- Write code once - business logic, cross-platform UI
- Maintain UX and quality of native apps
- Improve DX- hot reloading, less context switching, familiar ecosystem of tools



How we started?

- Started in August 2016, with RN 0.30
- Majority of community modules and projects were in early stages (Expo, React-Navigation)
- No real support for hardware functions
- Android support was just introduced
- Mainly focusing on writing our own components and navigation
- No easy way of debugging

Dickey's Barbecue

AUTHENTICITY
WITH A CONSCIENCE.

We are committed to sourcing,
only responsibly raised, quality meats.

DICKEY'S
BARBECUE PIT

Everyday menu

Catering order

Your Profile | Track Your order



Menu



Cart



Settings

Back

Add to Cart

Item 1



Total item price: \$ 6.55 *

* Price changes after choosing some options

Description Ulparcie nihictur arum repelitate optam quatesEt, iumquides ea volorehenda nonsenihitia sus, autect es valoris aditatus dolorer estiaeperit quam sitaspitum, quatates aute voluptatinum alienatiaecusae ne eum fuga. Et entio inihil ipsume la del ilignitius voluptatem.

Chose your toppings

Quantity



1



Add special instructions

Enter note

Add to Cart



Cart

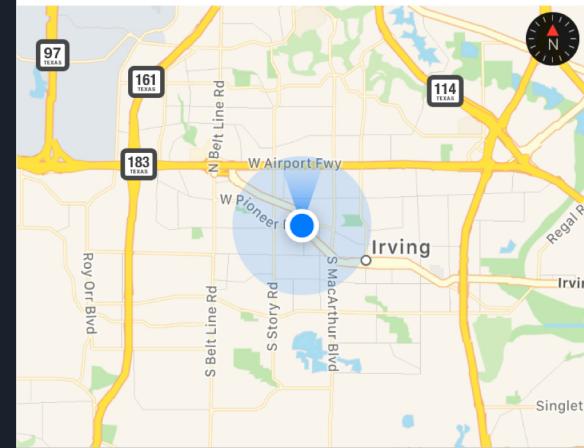


Settings

Back

Store

Search Store



Plano-14th St

1211 14th Street · 5.8 miles

Closed Now

Grand Prairie-Dalworth St.

2469 Dalworth St · 14 miles

Opened Now



Cart



Settings



Curbside Pick Up

Central

4610 N Central Expressway, Dallas, TX



05/26/2018 11:30 AM

\$42.59 >

Check In

New Order

My Orders



Order



Cart



Loyalty



More

< Family Packs & More

ADD TO CART

Family Pack
feeds up to 6 people

Starting at \$ 37.50

445 to 8324 cal*

*Calorie count depends on menu item options

Includes a choice of 2 meats (1 lb. each), 3 medium sides, 6 rolls and barbecue sauce.

Options

First Meat *
Chicken Breast 445 cal \$0.00 >Second Meat *
St. Louis Style Pork Ribs 1370 cal \$0.00 >First Medium Side *
Caesar Salad 255 cal \$0.00 >Second Medium Side *
Cabbage Slaw 692 cal \$0.00 >Third Medium Side *
>

Order



Cart



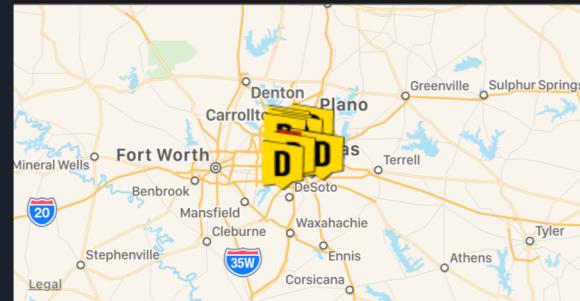
Loyalty



More

< Find Location

Location



Central

4610 N Central Expressway, Dallas • 0.23 mi.

Services:



Pickup

ORDER



ORDER



ORDER

Wycliff

Participating stores may vary on the app. Please go to www.dickeys.com to find your location if it is not available.



Order



Cart



More



The Good Parts

- Resources/time savings
- Cross-platform
- Yoga layout system (flexible)
- Unified design across platforms
- Animated API (60 FPS goodness)
- Native Module bridge
- Hot reloading (iteration speed)
- It's just JavaScript
- Community
- State management (same as web)
- Platform-specific components & styles
- Networking

The Good Parts – layout smartphone/tablet

No SIM ⌂ 3:44 PM 🔋

Back Guest Favorites

1 Meat Plate
Starting at \$ 11.50
Slow-smoked and rubbed with our Dickey's rib rub, served with 2 sides... 

2 Meat Plate
Starting at \$ 12.50
Try your favorite combination of 2 slow-smoked meats, rubbed with... 

Pork Rib Plate
Starting at \$ 14.50
Slow-smoked and rubbed with our Dickey's rib rub, served with 2 sides... 

Beef Brisket Classic Sandwich
Starting at \$ 6.50
Includes a choice of chopped or sliced delicious slow-smoked bris... 

Mac & Cheese
Starting at \$ 2.50

 Order  Cart  Loyalty  More

iPad ⌂ 3:45 PM 🔋 100%

Everyday Menu Guest Favorites

1 Meat Plate
Starting at \$ 9.95
Slow-smoked and rubbed with our Dickey's rib rub, served with 2 sides and a roll. 

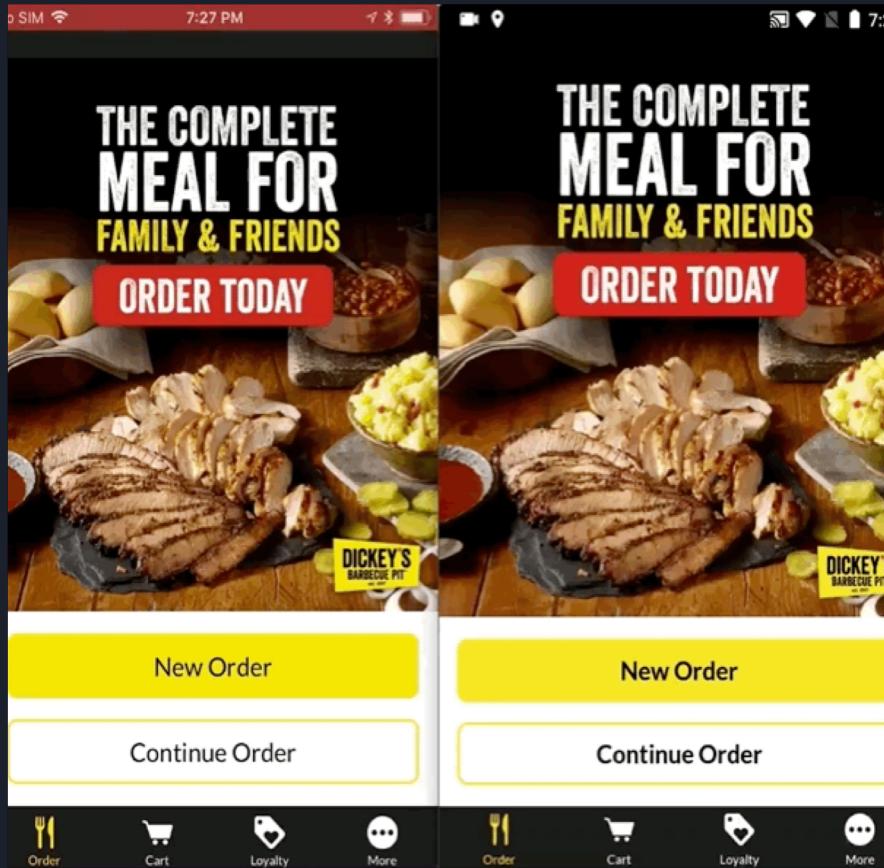
2 Meat Plate
Starting at \$ 10.95
Try your favorite combination of 2 slow-smoked meats, rubbed with our Dickey's rib rub and served with 2 sides and a roll. 

Pork Rib Plate
Starting at \$ 13.50
Slow-smoked and rubbed with our Dickey's rib rub, served with 2 sides and a roll. 

Beef Brisket Classic Sandwich
Starting at \$ 5.50
Includes a choice of chopped or sliced delicious slow-smoked brisket on a Brioche bun 

 Order  Cart  Loyalty  More

The Good Parts – unified design



The Good Parts – JS ecosystem



LODASH

JavaScript utility library



MOMENT

Simple date manipulation

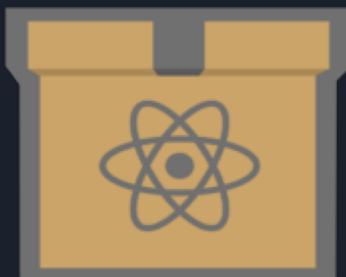


RELAY

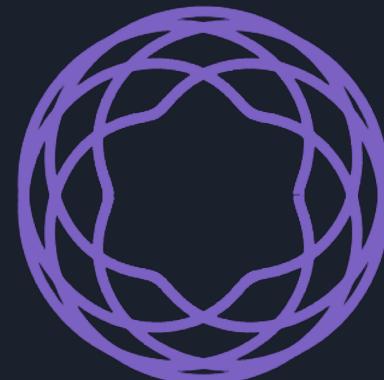
The Good Parts – community



EXPO



RNPM



REACT NAVIGATION

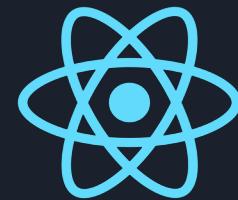
The Good Parts – state management



REDUX



RELAY



CONTEXT API

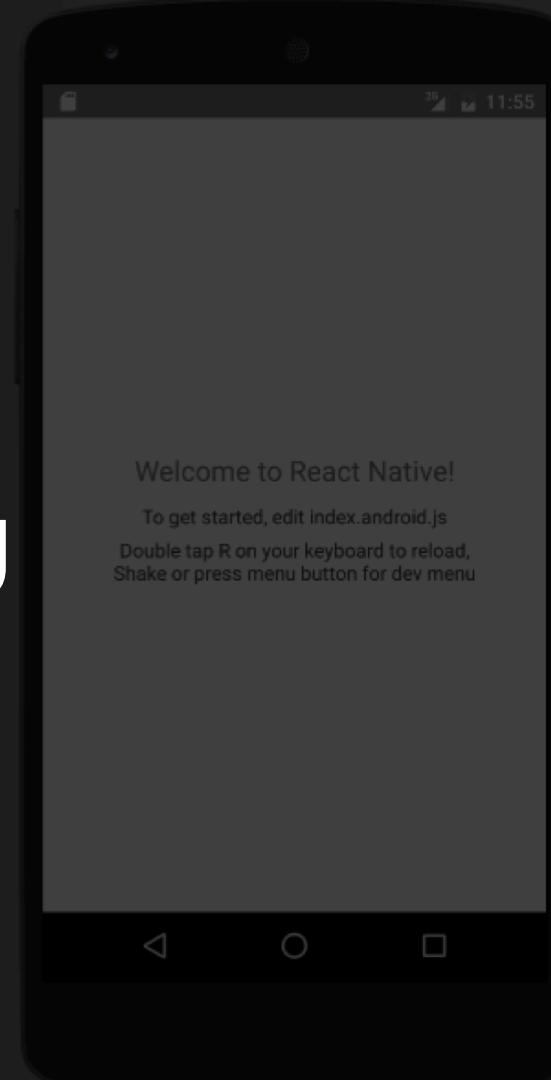


MOBX

```
import React, { Component } from 'react';
import {
  AppRegistry,
  StyleSheet,
  Text,
  View
  - from 'react-native';

export default class AwesomeProject extends Component {
  render() {
    return (
      <View style={styles.container}>
        <Text style={styles.welcome}>
          Welcome to React Native!
        </Text>
        <Text style={styles.instructions}>
          To get started, edit index.android.js
        </Text>
        <Text style={styles.instructions}>
          Double tap R on your keyboard to reload,{'\n'}
          Shake or press menu button for dev menu
        </Text>
      </View>
    );
  }
}
```

Hot reloading





The Good Parts – platform-specific code

- > CustomComponent.ios.js
- > CustomComponent.android.js

```
// No extension specified
import CustomComponent from './CustomComponent';
```



The Good Parts – platform-specific code

```
import { Platform, StyleSheet } from 'react-native';

const styles = StyleSheet.create({
  paddingTop: (Platform.OS === 'ios') ? 30 : 0,
  platformDependent: Platform.select({
    ios: { color: 'red' },
    android: { color: 'blue' }
  })
});
```



The Good Parts – networking

- New Fetch API Supported
- Raw XMLHttpRequest implemented if required
- WebSockets Supported



The Bad Parts

- Documentation
- Not pixel perfect
- 3rd party library maturity, making a library requires a very wide skill-set
- Some basic things are harder than they should be eg. image caching, custom fonts
- Breaking version changes, library version compatibility tables
- Write-once, debug everywhere
- Native SDK-s updates
- Android maturity
- Optimal developer experience requires a MacBook
- Asynchronous bridge architecture limitations (UI callbacks, native error handling)
- JavaScriptCore inconsistencies



Debugging can be really hard

- Unexpected bugs in RN:
 - left: 100% moves the component 110%
 - NetInfo false negatives
 - AsyncStorage not responding
 - Unable to update inputs programmatically
- Some native components attributes are not available through RN (e.g. ScrollView Y offset)
- Some actions (e.g. logging out) can require you to manage the navigation stack by hand which is hard and mistake prone
- Native errors – “x app has crashed”, no context or anything to help debugging
 - Bugsnag to the rescue
- You wake up one day, the Play Store was updated and your build no longer completes



Navigators

- Native navigation
 - built on top of the OS's existing navigation stack
 - React Native Navigation (wix) 
 - Native Navigation (Airbnb)
- JavaScript navigation
 - emulates what the native stack does
 - React Navigation 
 - React Native Router Flux
 - React Router Native (bindings for React Router)

```
const MainContent = TabNavigator(  
  {  
    OrderTab: {  
      screen: Exercise1  
    }  
  },  
  {  
    initialRouteName: 'OrderTab',  
    tabBarComponent: TabBarBottom,  
    tabBarPosition: 'bottom'  
  }  
);  
  
const TopNavigator = StackNavigator({  
  MainRoute: {  
    screen: MainContent  
  },  
  Second: {  
    screen: Exercise2,  
    navigationOptions: ({ navigation: { state: { params = {} } } }) => ({  
      title: params.title  
    })  
  }  
}, {  
  initialRouteName: 'MainRoute',  
  mode: 'modal'  
});
```



Tips & Tricks

- Always check <http://www.awesome-react-native.com/>
- Lock dependencies (^ and ~ are sometimes your enemies)
- Use shouldComponentUpdate or PureComponent
- Attach keys when mapping arrays in the render
- Get rid of the ListView (use FlatList/SectionList/VirtualizedList)
- Do NOT think React Native can solve every problem or be used in every scenario
- Look at <https://canny.io/product-pains> for trending problems



What we expect to improve

- Stabilize and improve the navigation libraries
- Maintain support for libraries in the React Native ecosystem
- Improve the experience for adding native libraries

- call synchronously into JavaScript threads fix asynchronous bridge (State of React Native 2018)



Conclusion

- Good for API-driven apps and short-term apps (for example an app for a one-shot event like a concert or a conference)
- Fast release cycle
- Many cons but still the best platform for cross platform applications
- Amazing JavaScript tools that fit together really well help you only write the domain specific code you need to write



State of React Native 2018

AirBnb Sunsetting React Native