React Native Again…

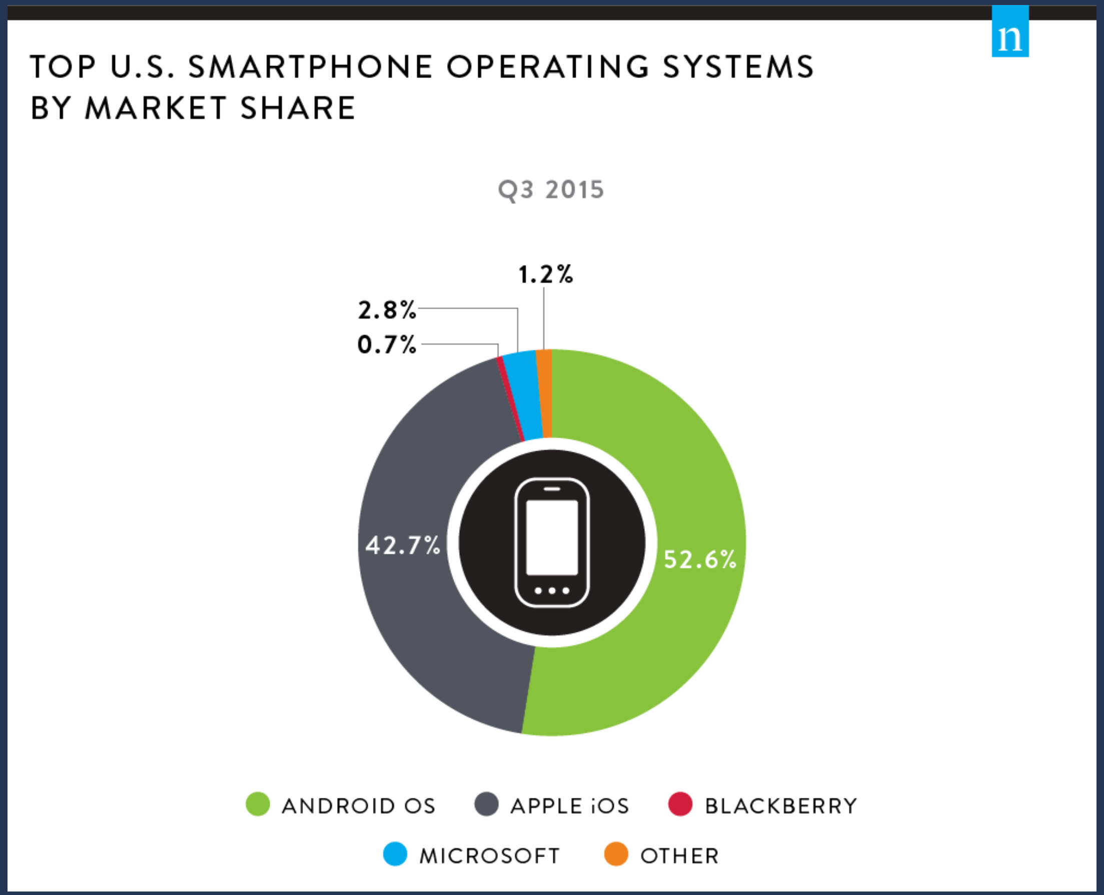
Download **Expo** from the Apple/Google Play Store to play along with the demo

- Problems with mobile development
- React Native - how does it work?
- Expo
- Demo

Problems with mobile development

Problem #1 - Fragmentation

Reaching 95% of mobile users involves developing for 2 platforms



Problem #2 - Frameworks

First generation of cross-platform mobile frameworks suck

ionic

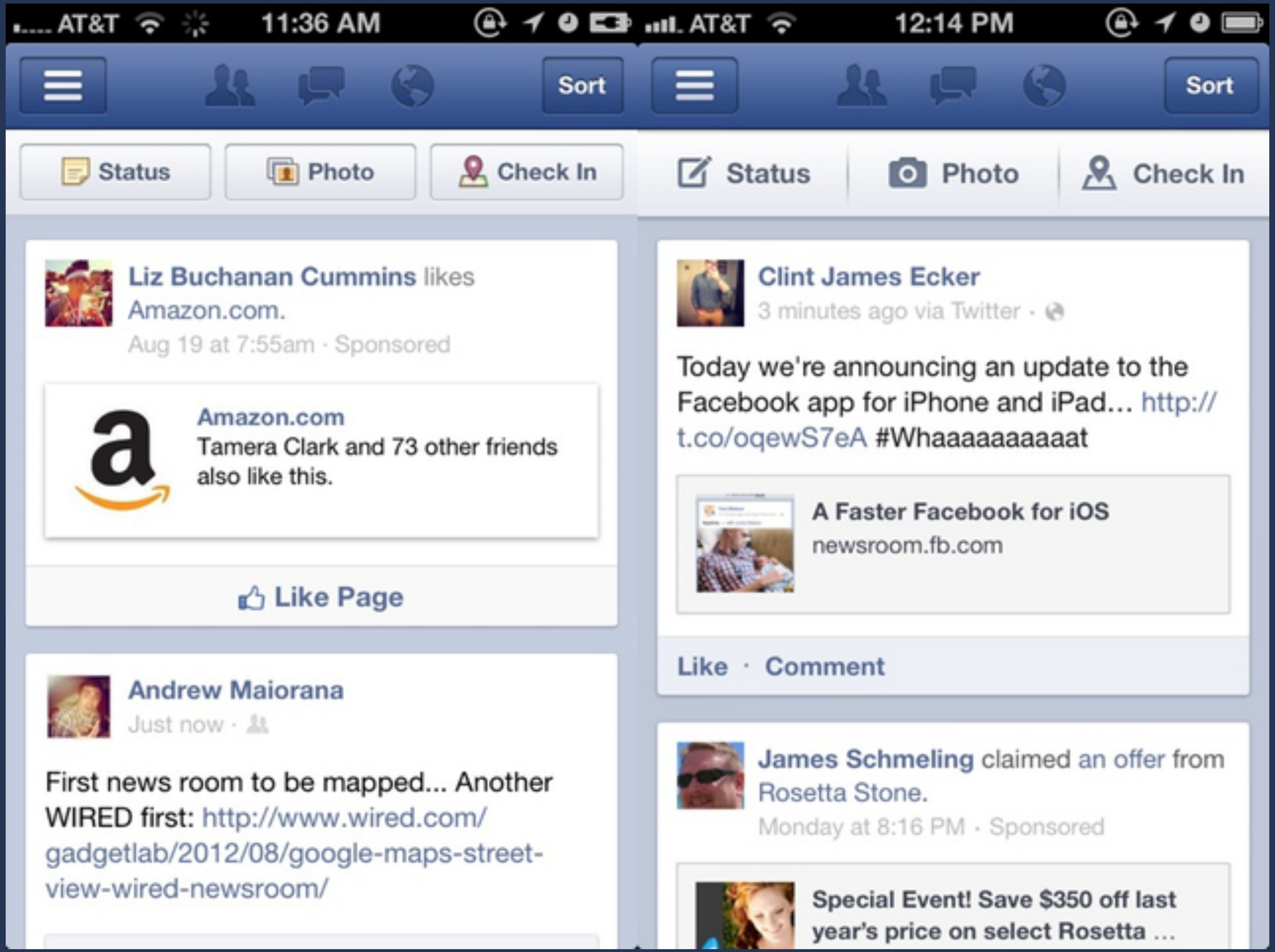
PhoneGap

Xamarin

Problem #2 cont...

They suck because:

- Web views have poor performance
- The promise of write-once-run-anywhere isn't a good fit for mobile



Liz Buchanan Cummins likes Amazon.com.
Aug 19 at 7:55am · Sponsored

Amazon.com
Tamera Clark and 73 other friends also like this.

Like Page

Andrew Maiorana
Just now ·

First news room to be mapped... Another WIRED first: http://www.wired.com/gadgetlab/2012/08/google-maps-street-view-wired-newsroom/

Clint James Ecker
3 minutes ago via Twitter ·

Today we're announcing an update to the Facebook app for iPhone and iPad... http://t.co/oqewS7eA #Whaaaaaaaaat

A Faster Facebook for iOS
newsroom.fb.com

Like · Comment

James Schmeling claimed an offer from Rosetta Stone.
Monday at 8:16 PM · Sponsored

Special Event! Save $350 off last year's price on select Rosetta ...

Under the hood: Rebuilding Facebook for iOS

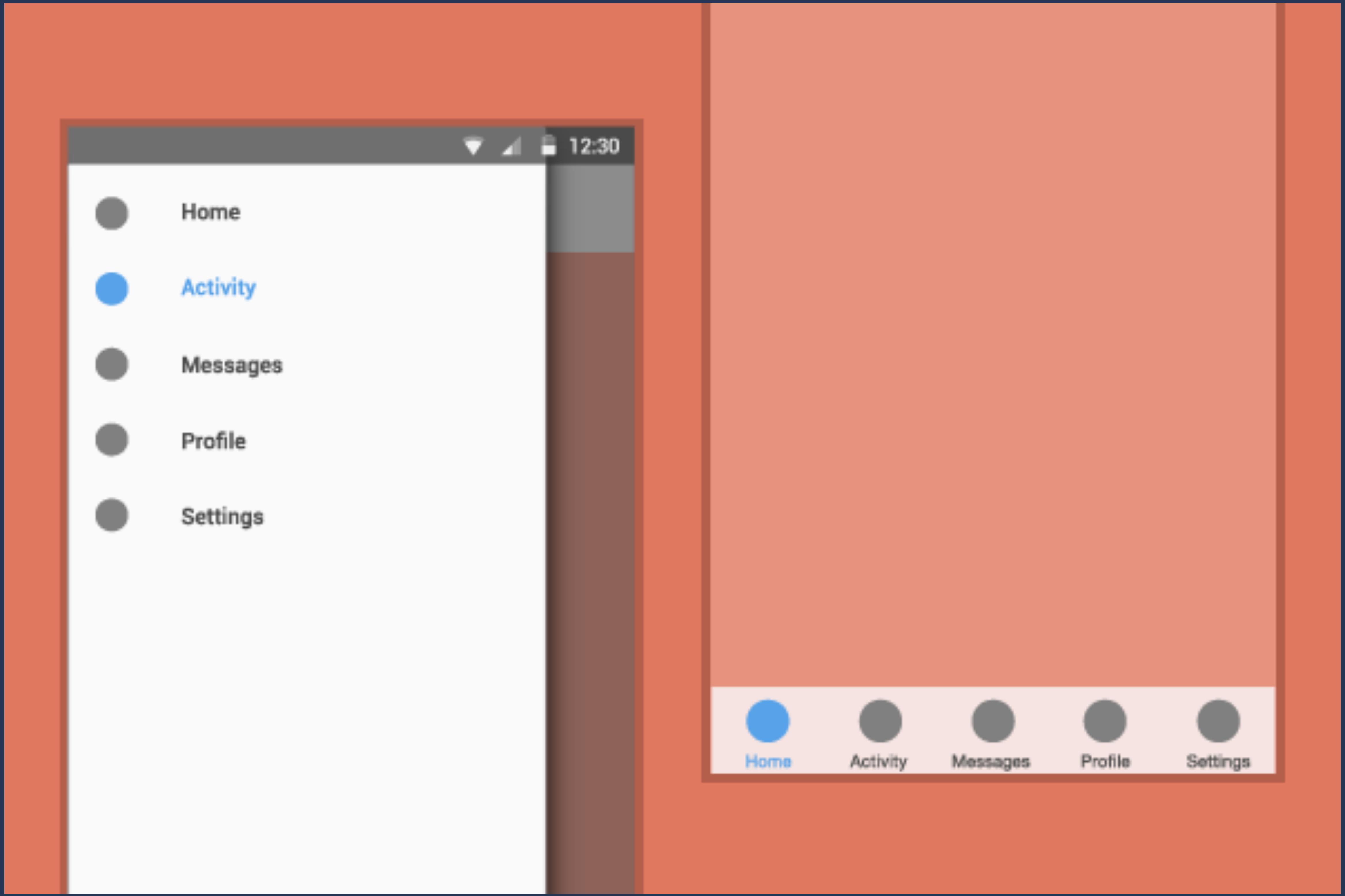
"...while utilizing web technology has allowed us to support more than 500 million people using Facebook...people expect a fast, reliable experience and our iOS app was falling short..."

-- **Jonathan Dann**, Senior Facebook Engineer

Problem #3 - Native == $$$

Native app development is expensive:

- Developers need to know 2 languages
- Platform APIs are not aligned
- Design guidelines for Android/iOS are different


12:30
Home
Activity
Messages
Profile
Settings
Home
Activity
Messages
Profile
Settings

React Native

- *Learn* once write anywhere
- Native performace in Javascript
- Supports iOS, Android
- Apple TV, Windows Phone, OSX support coming soon
- Live updates

How does it work?

Javascript core communicates with native APIs via the 'bridge'

The bridge helps us communicate with the native code by marshalling types, setting properties and passing events back to our Javascript

Initialising native code

- Implement a subclass of `RCTViewManager` which does most of the work
- Add a macro `RCT_EXPORT_MODULE()`
- Implement a method `-(UIView *)view` - returning an instance of the native class

Calling native code from JS

In your javascript, simply import this new (MyClass) class...

```
// MyComponent.js

import { requireNativeComponent } from 'react-native';

// requireNativeComponent automatically resolves this to "RNTMyClassManager"
module.exports = requireNativeComponent('RNTMyClass', null);
```

Setting data on a native component via props

Native - `RNTMyClass.m`

```
RCT_EXPORT_VIEW_PROPERTY(awesomeEnabled, BOOL)
```

Javascript - `MyComponent.js`

```
<MyClass awesomeEnabled={true} />
```

Document your props

```
MapView.propTypes = {
  /**
   * When this property is set to `true` awesomeness is enabled
   */
  awesomeEnabled: React.PropTypes.bool,
};
```

More complex setters require conversion functions to modify raw JSON into native objects before assigning to class properties

Receiving events back from a native component

```objc
// RNTMyClass.h
@property (nonatomic, copy) RCTBubblingEventBlock onChange;

// RNTMyClassManager.h
RCT_EXPORT_VIEW_PROPERTY(onChange, RCTBubblingEventBlock)

// implement callback

- (void)awesomeView:(RNTAwesome *)awesomeView awesomenessDidChange:(BOOL)awesome
{
  // check callback set, if not return
  if (!awesomeView.onChange) {
    return;
  }

  awesomeView.onChange(@{
    @"awesome": @(awesome)
  });
```

Receiving events cont...

```
class AwesomeView extends React.Component {
  constructor() {
    this._onChange = this._onChange.bind(this);
  }
  _onChange(event: Event) {
    console.log(event.nativeEvent.awesome);
  }
  render() {
    return <RNTAwesome onChange={this._onChange} />;
  }
}
```

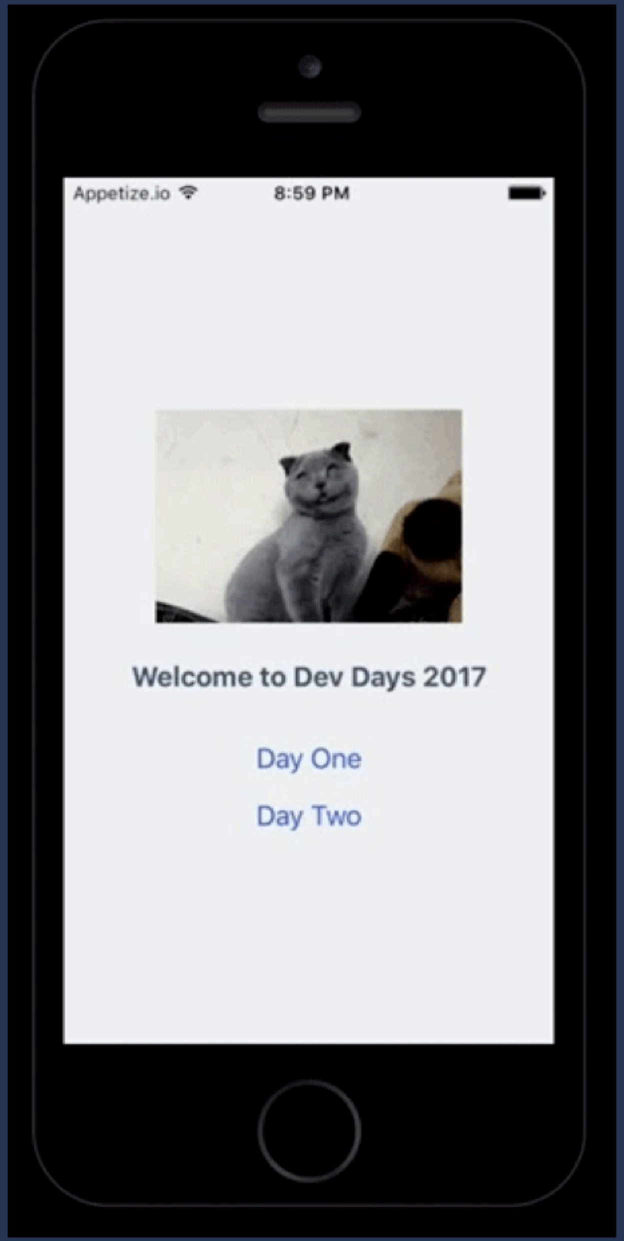
Expo

- React Native framework that helps you get up and running quickly
- Prototype ideas and share with your team/friends
- Pre-defined components (Facebook/Google login, Image Picker, Permissions)
- Web IDE (Snack - https://snack.expo.io)

Demo

- https://snack.expo.io/BJkDa_XGW

So something went wrong...


Appetize.io
8:59 PM
Welcome to Dev Days 2017
Day One
Day Two