React Native

2018-04-25 // Best practices

@josomers

Image source: http://artcore.online/

CONNECT THE FUTURE

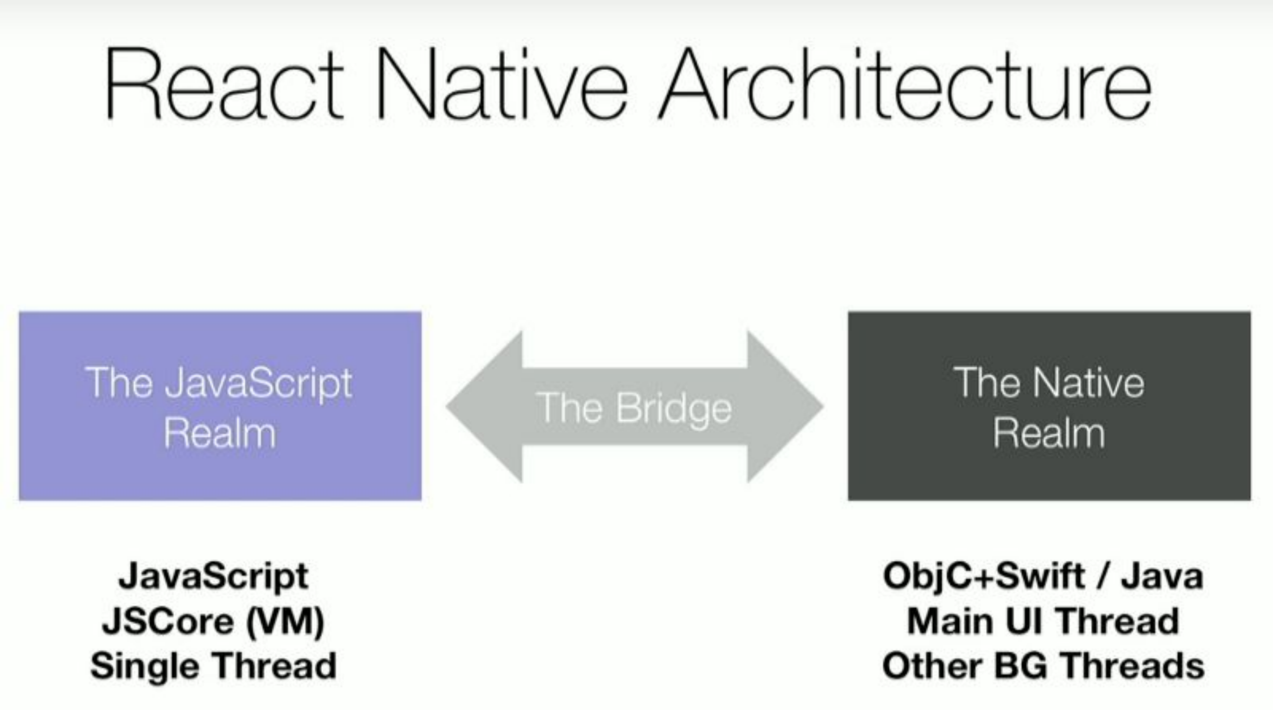
Innovating ports & cities through **evolution, not disruption**

Let’s talk about React Native…

- React Native
- Example: Artcore (by ArtSound)
- Best practices
- Native modules

React Native

- Shared “React” code layer between iOS and Android
- Native app, build upon web technology
- Native rendering of UI components
- Bridge between Javascript Realm & Native Realm



React Native Architecture

The JavaScript Realm

The Bridge

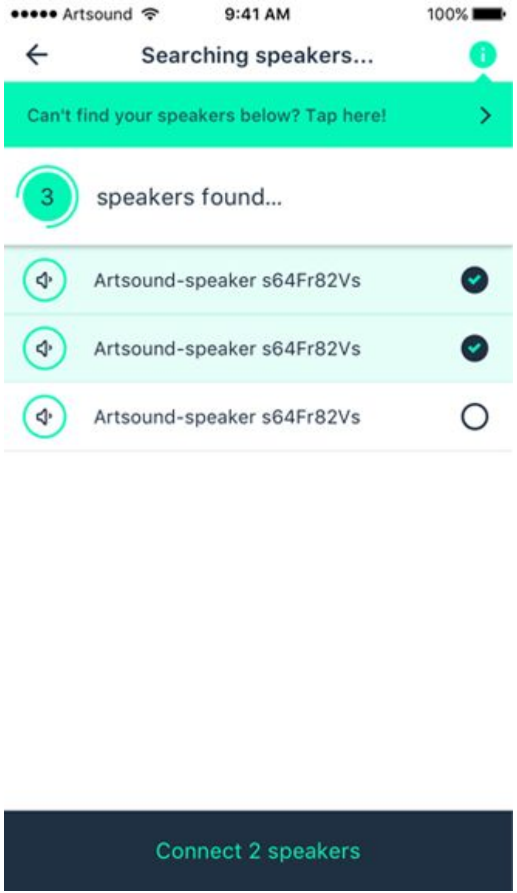
The Native Realm

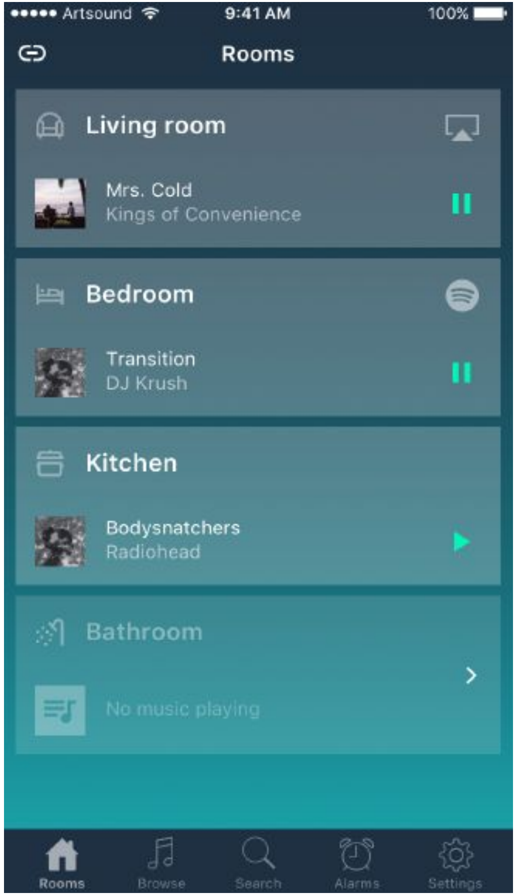
JavaScript
JSCore (VM)
Single Thread

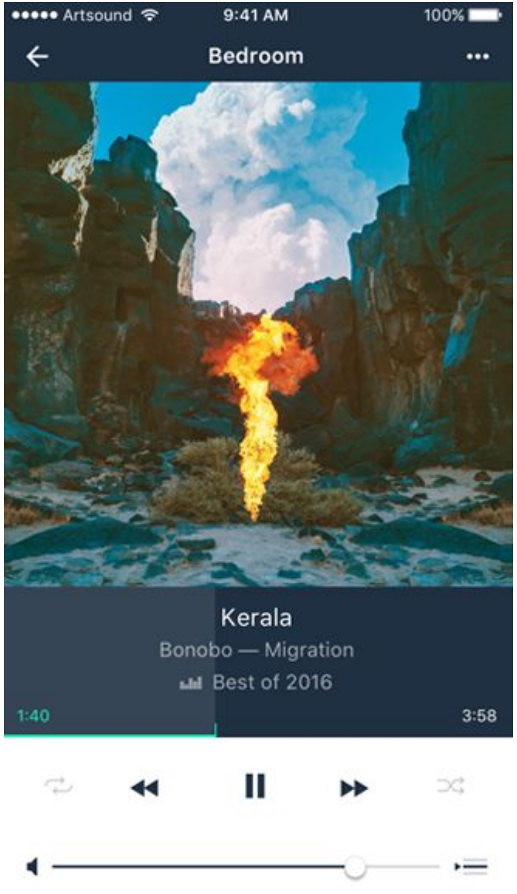
ObjC+Swift / Java
Main UI Thread
Other BG Threads

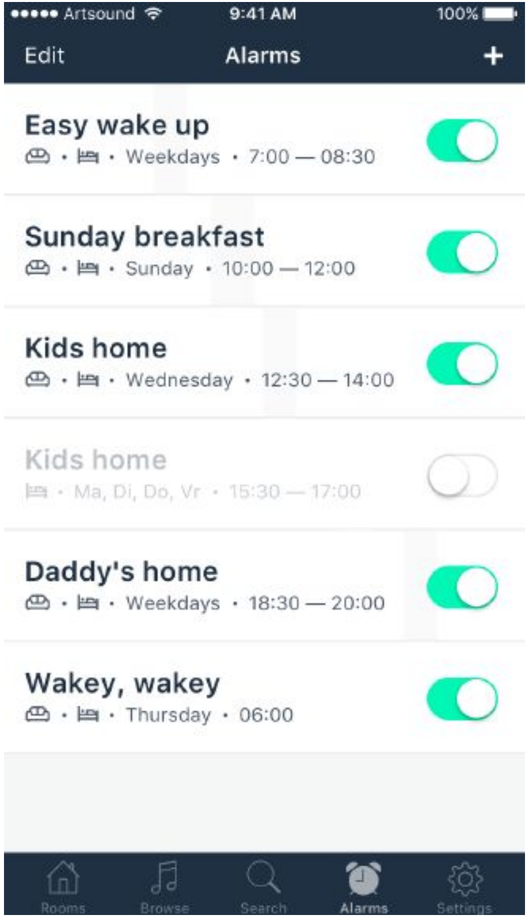
ARTCORE

BY ARTSOUND


●●●●● Artsound
9:41 AM
100%
Searching speakers...
Can't find your speakers below? Tap here!
3
speakers found...
Artsound-speaker s64Fr82Vs
Artsound-speaker s64Fr82Vs
Artsound-speaker s64Fr82Vs
Connect 2 speakers


●●●●● Artsound
9:41 AM
100%
Rooms
Living room
Mrs. Cold
Kings of Convenience
Bedroom
Transition
DJ Krush
Kitchen
Bodysnatchers
Radiohead
Bathroom
No music playing
Rooms
Browse
Search
Alarms
Settings


●●●●● Artsound
9:41 AM
100%
Bedroom
Kerala
Bonobo — Migration
Best of 2016
1:40
3:58



Alarms

Easy wake up
• • Weekdays • 7:00 — 08:30

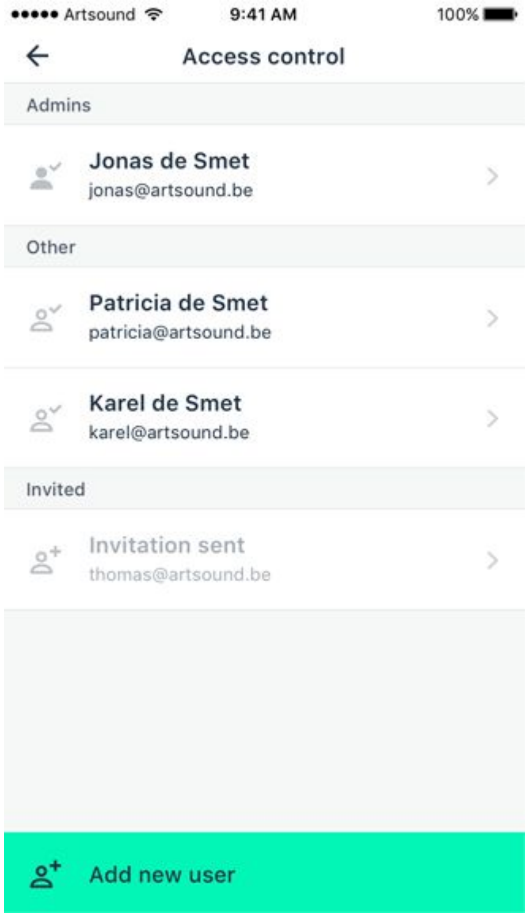
Sunday breakfast
• • Sunday • 10:00 — 12:00

Kids home
• • Wednesday • 12:30 — 14:00

Kids home
• Ma, Di, Do, Vr • 15:30 — 17:00

Daddy's home
• • Weekdays • 18:30 — 20:00

Wakey, wakey
• • Thursday • 06:00



Access control

Admins

Jonas de Smet
jonas@artsound.be

Other

Patricia de Smet
patricia@artsound.be

Karel de Smet
karel@artsound.be

Invited

Invitation sent
thomas@artsound.be

Product info

- **mDNS technology (native Module!)**
- Event based architecture
  - Cloud + App sockets
  - App + Speaker event polling
  - Redux
- Completely typed code base
- 85% unit test coverage with Jest
- Cross platform with React Native: Android (from 4.1) and iOS (from iOS 9.0)
- Completely automated CI cycle with Fastlane
- Integration with different streaming sources like Spotify / Tidal / vTuner and integration with NAS sources.
- …

Product info

- **mDNS technology (native Module!)**
- Event based architecture
  - Cloud + App sockets
  - App + Speaker event polling
  - Redux
- Completely typed code base
- 85% unit test coverage with Jest
- Cross platform with React Native: Android (from 4.1) and iOS (from iOS 9.0)
- Completely automated CI cycle with Fastlane
- Integration with different streaming sources like Spotify / Tidal / vTuner and integration with Airplay and NAS sources.
- ...

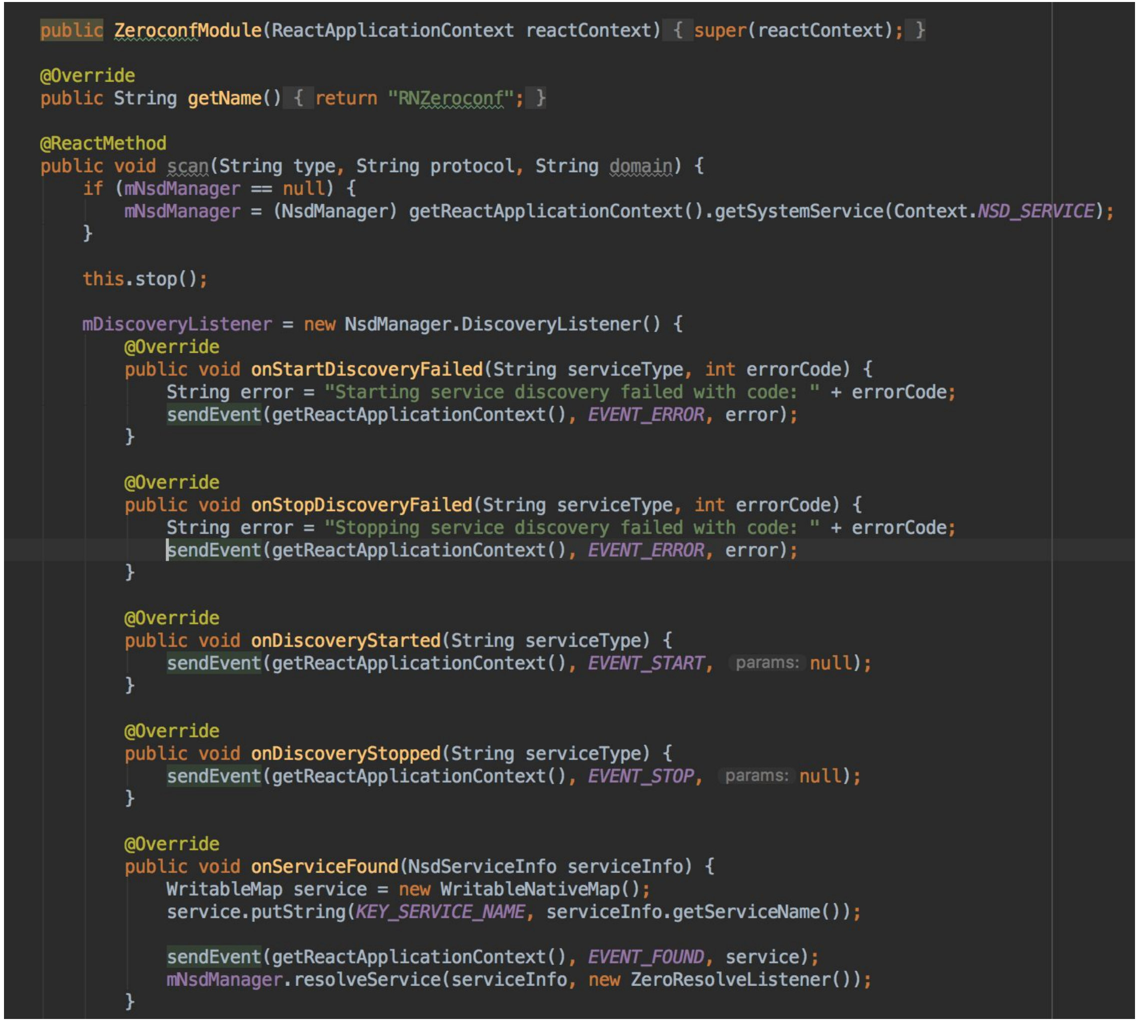
React Native Module

mDNS

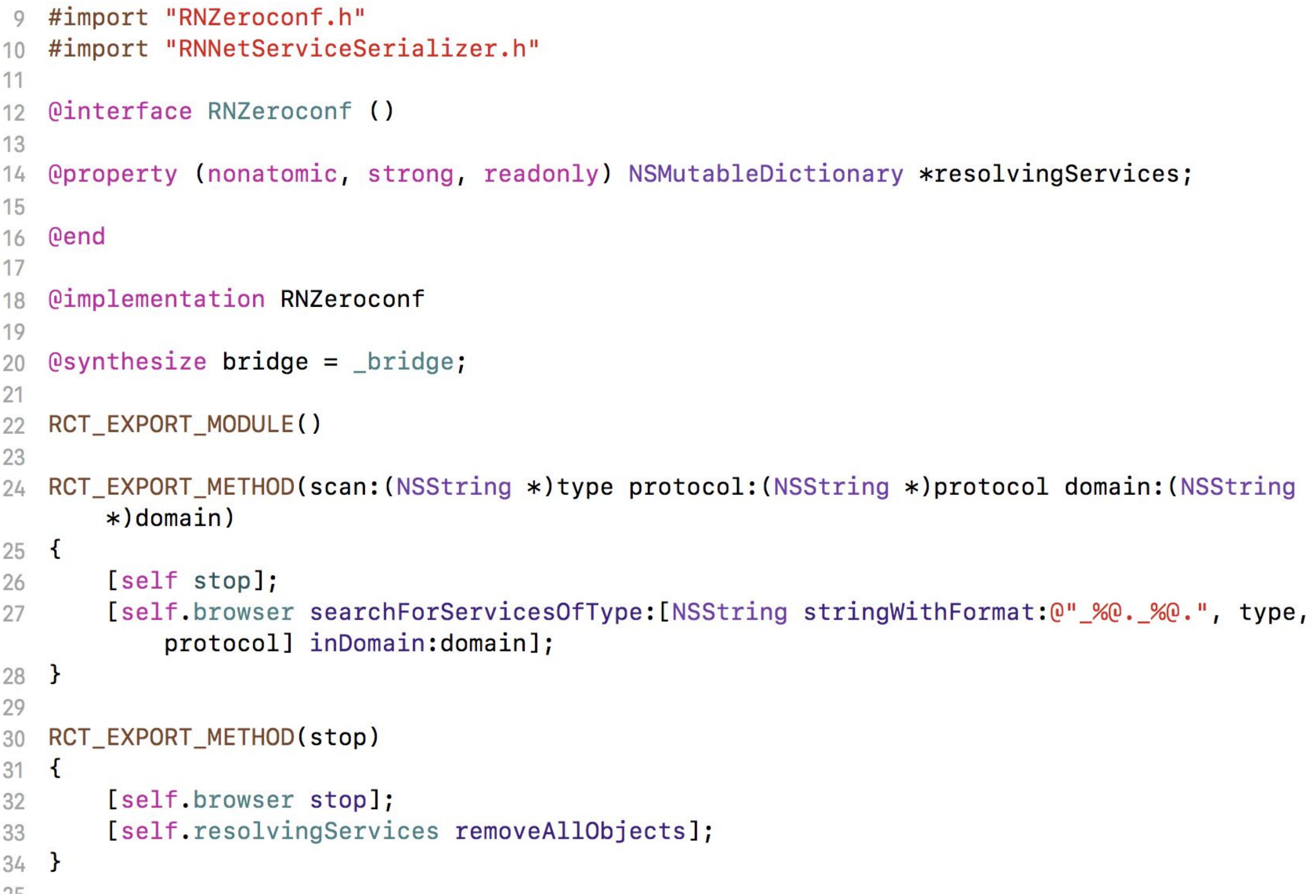
ZeroConf module

Native iOS / Android piece of code

React layer in between



```
public ZeroconfModule(ReactApplicationContext reactContext) { super(reactContext); }

@Override
public String getName() { return "RNZeroconf"; }

@ReactMethod
public void scan(String type, String protocol, String domain) {
    if (mNsdManager == null) {
        mNsdManager = (NsdManager) getReactApplicationContext().getSystemService(Context.NSD_SERVICE);
    }

    this.stop();

    mDiscoveryListener = new NsdManager.DiscoveryListener() {
        @Override
        public void onStartDiscoveryFailed(String serviceType, int errorCode) {
            String error = "Starting service discovery failed with code: " + errorCode;
            sendEvent(getReactApplicationContext(), EVENT_ERROR, error);
        }

        @Override
        public void onStopDiscoveryFailed(String serviceType, int errorCode) {
            String error = "Stopping service discovery failed with code: " + errorCode;
            sendEvent(getReactApplicationContext(), EVENT_ERROR, error);
        }

        @Override
        public void onDiscoveryStarted(String serviceType) {
            sendEvent(getReactApplicationContext(), EVENT_START,  params: null);
        }

        @Override
        public void onDiscoveryStopped(String serviceType) {
            sendEvent(getReactApplicationContext(), EVENT_STOP,  params: null);
        }

        @Override
        public void onServiceFound(NsdServiceInfo serviceInfo) {
            WritableMap service = new WritableNativeMap();
            service.putString(KEY_SERVICE_NAME, serviceInfo.getServiceName());

            sendEvent(getReactApplicationContext(), EVENT_FOUND, service);
            mNsdManager.resolveService(serviceInfo, new ZeroResolveListener());
        }
```



```
#import "RNZeroconf.h"
#import "RNNetServiceSerializer.h"

@interface RNZeroconf ()

@property (nonatomic, strong, readonly) NSMutableDictionary *resolvingServices;

@end

@implementation RNZeroconf

@synthesize bridge = _bridge;

RCT_EXPORT_MODULE()

RCT_EXPORT_METHOD(scan:(NSString *)type protocol:(NSString *)protocol domain:(NSString
    *)domain)
{
    [self stop];
    [self.browser searchForServicesOfType:[NSString stringWithFormat:@"_%@._%@.", type,
        protocol] inDomain:domain];
}

RCT_EXPORT_METHOD(stop)
{
    [self.browser stop];
    [self.resolvingServices removeAllObjects];
}
```

Product info

- mDNS technology (native Module!)
- **Event based architecture**
  - **Cloud + App sockets**
  - **App + Speaker event polling**
  - **Redux**
- Completely typed code base
- 85% unit test coverage with Jest
- Cross platform with React Native: Android (from 4.1) and iOS (from iOS 9.0)
- Completely automated CI cycle with Fastlane
- Integration with different streaming sources like Spotify / Tidal / vTuner and integration with Airplay and NAS sources.
- ...

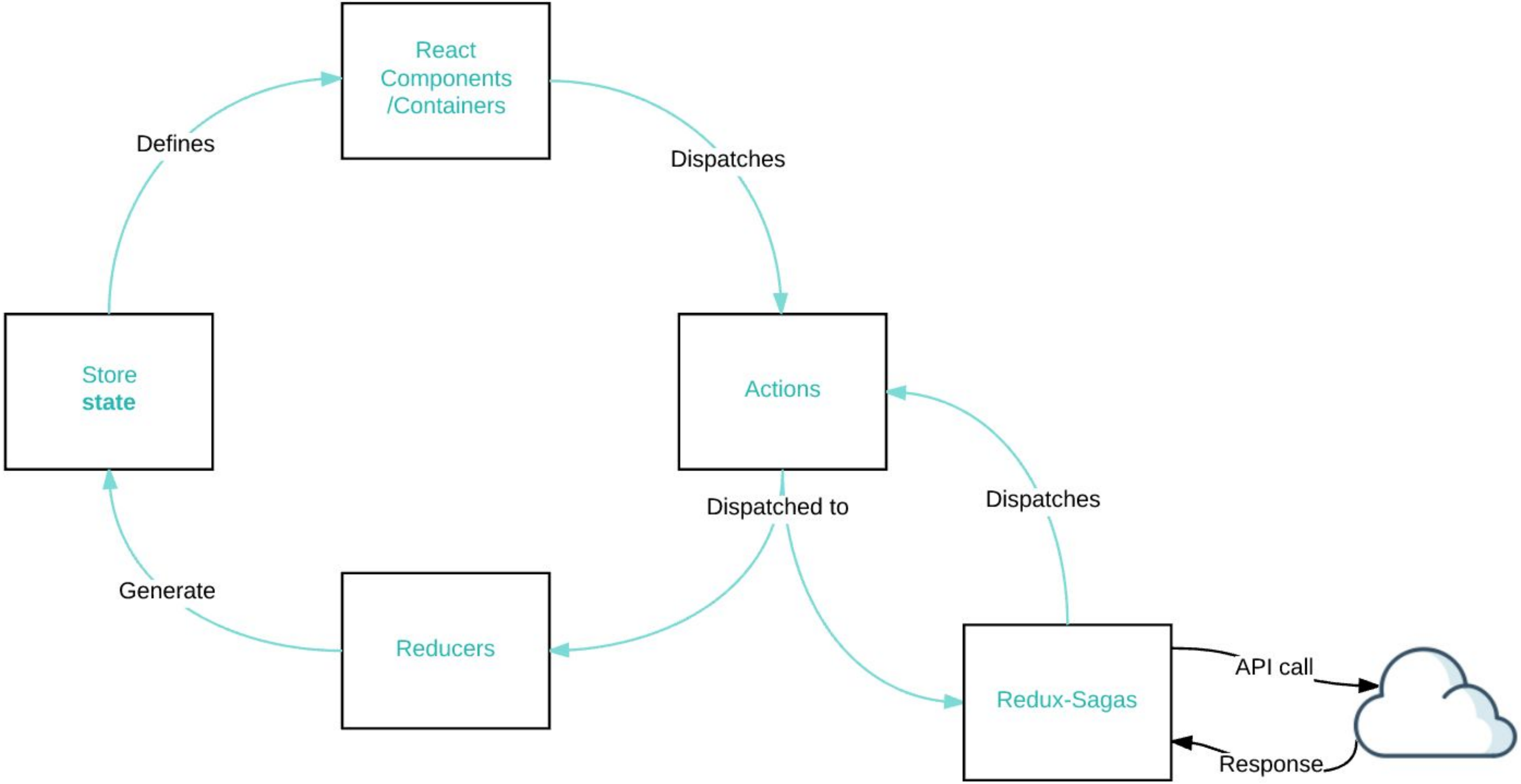
React

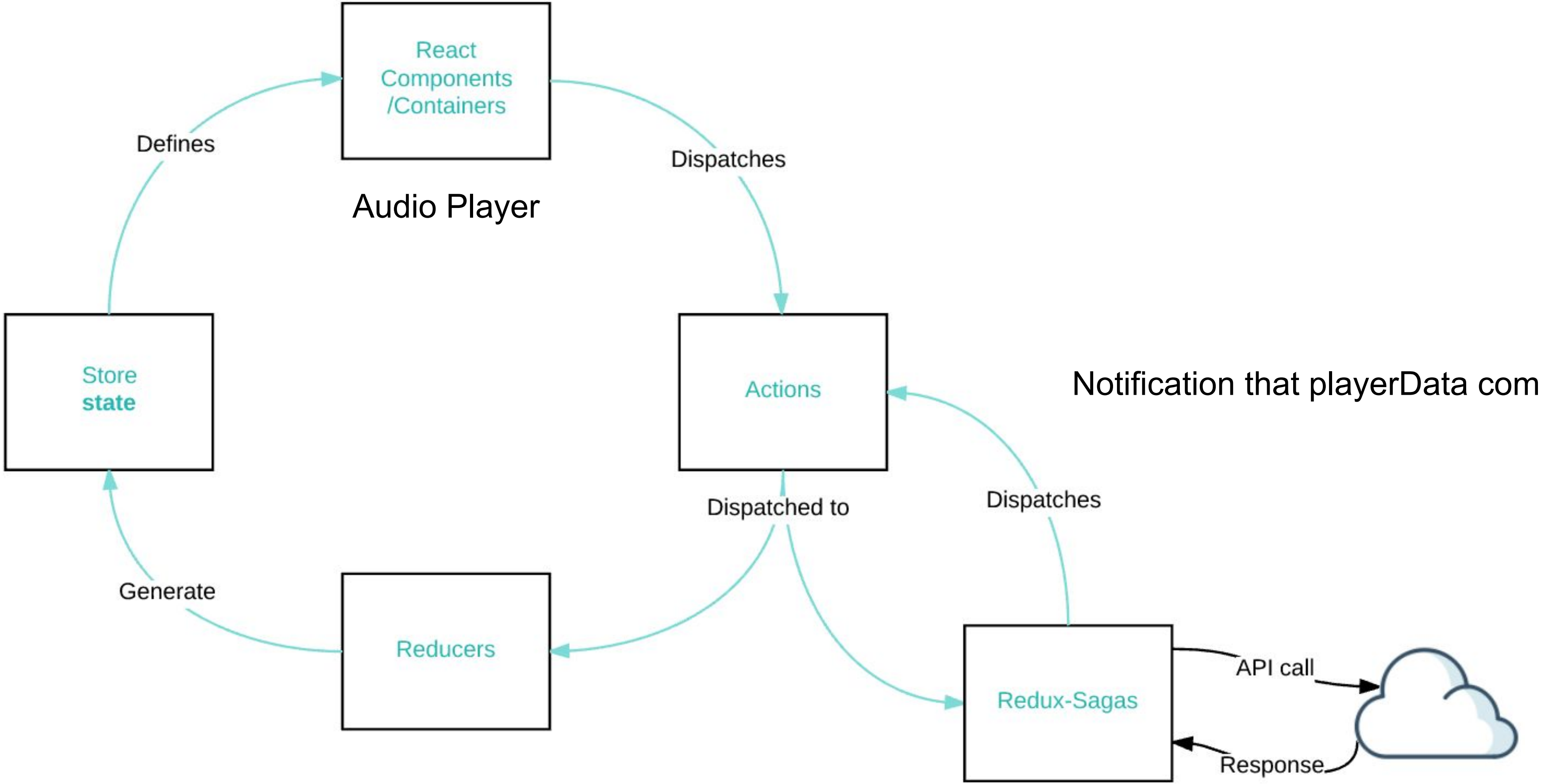
Redux

A predictable state container for JavaScript apps.

Perfect for event & state driven apps.

Offline capabilities


React
Components
/Containers
Dispatches
Actions
Dispatched to
Dispatches
Redux-Sagas
API call
Response
Reducers
Generate
Store
state
Defines



Audio Player

Notification that playerData comes in

PlayerData:
Timestamp / Album / Artist / Title / Queue info

Polling Task

Speaker API

Product info

- mDNS technology (native Module!)
- Event based architecture
  - Cloud + App sockets
  - App + Speaker event polling
  - Redux
- **Completely typed code base**
- 85% unit test coverage with Jest
- Cross platform with React Native: Android (from 4.1) and iOS (from iOS 9.0)
- Completely automated CI cycle with Fastlane
- Integration with different streaming sources like Spotify / Tidal / vTuner and integration with Airplay and NAS sources.
- ...

React

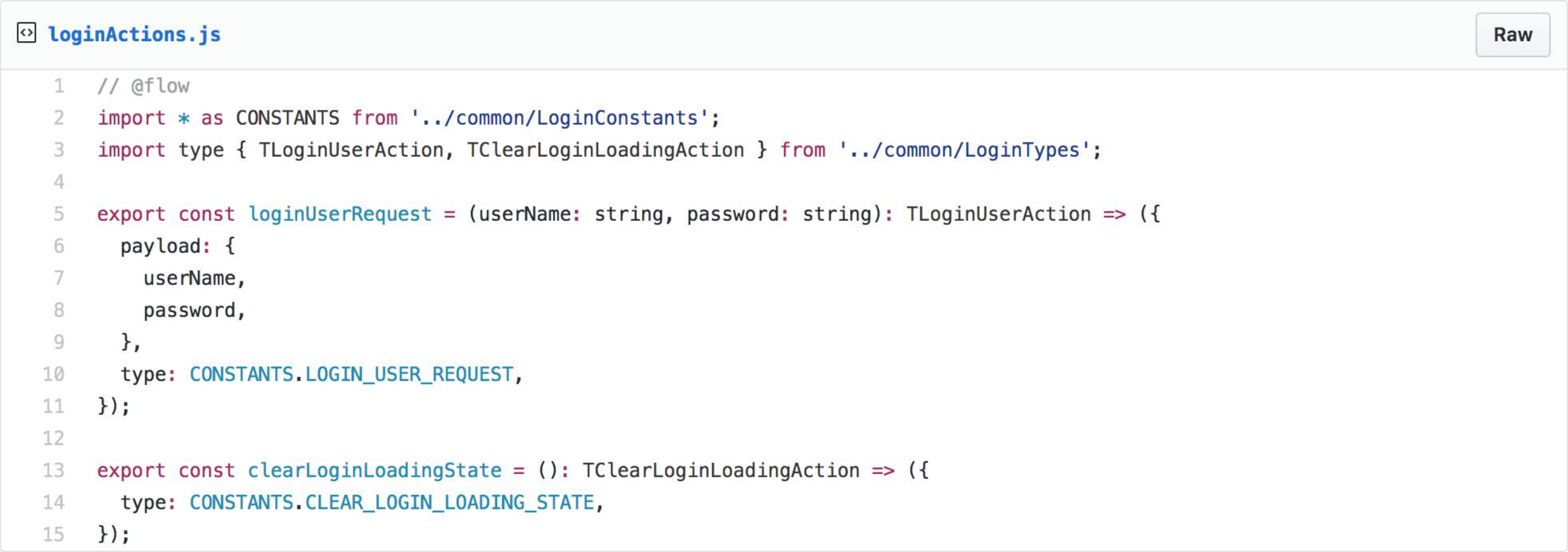
Flow

A static type checker for JavaScript by Facebook

Primitive type validation

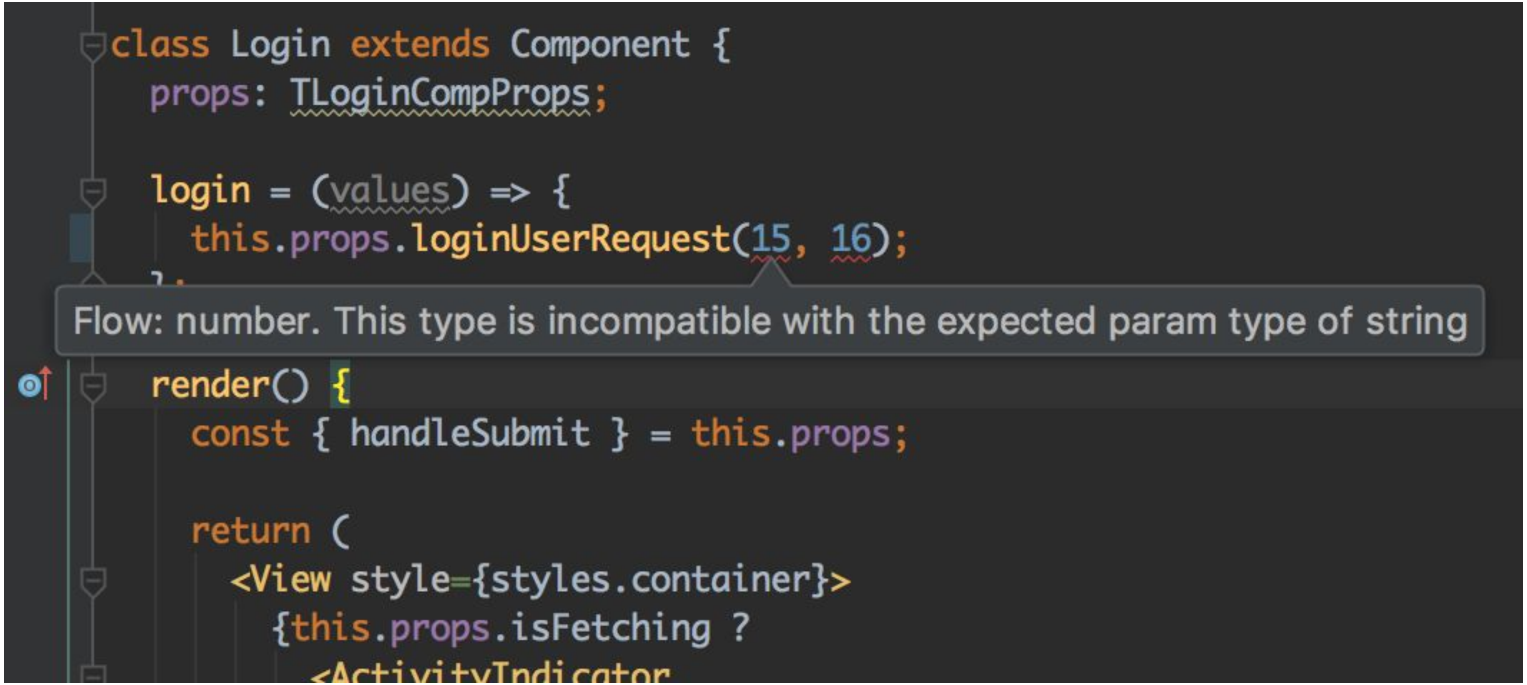
IDE integration

Make state immutable with + or $ReadOnly



loginActions.js

```javascript
// @flow
import * as CONSTANTS from '../common/LoginConstants';
import type { TLoginUserAction, TClearLoginLoadingAction } from '../common/LoginTypes';

export const loginUserRequest = (userName: string, password: string): TLoginUserAction => ({
  payload: {
    userName,
    password,
  },
  type: CONSTANTS.LOGIN_USER_REQUEST,
});

export const clearLoginLoadingState = (): TClearLoginLoadingAction => ({
  type: CONSTANTS.CLEAR_LOGIN_LOADING_STATE,
});
```



```
class Login extends Component {
  props: TLoginCompProps;

  login = (values) => {
    this.props.loginUserRequest(15, 16);
  };
```

Flow: number. This type is incompatible with the expected param type of string

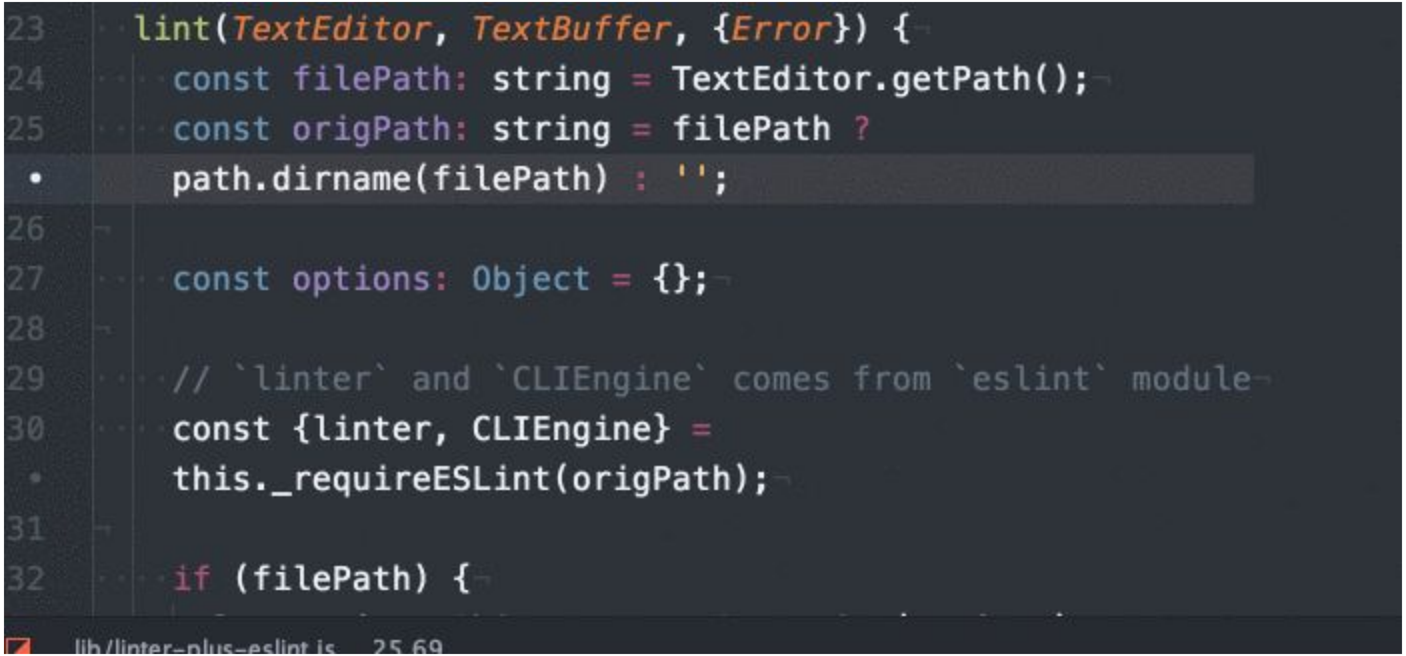
```
  render() {
    const { handleSubmit } = this.props;

    return (
      <View style={styles.container}>
        {this.props.isFetching ?
          <ActivityIndicator
```

React

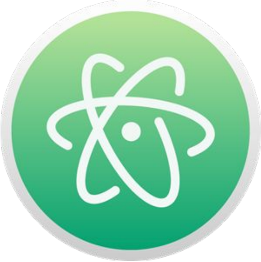
ESLint

ESLint is a tool for identifying and reporting on patterns found in JavaScript code.

IDE integration



```
  lint(TextEditor, TextBuffer, {Error}) {
    const filePath: string = TextEditor.getPath();
    const origPath: string = filePath ?
    path.dirname(filePath) : '';

    const options: Object = {};

    // `linter` and `CLIEngine` comes from `eslint` module
    const {linter, CLIEngine} =
    this._requireESLint(origPath);

    if (filePath) {
```



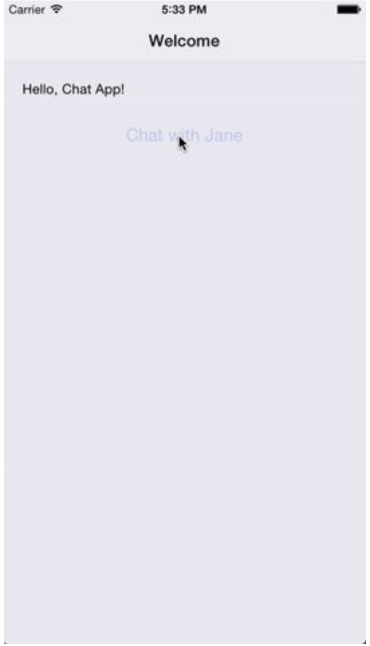
WS

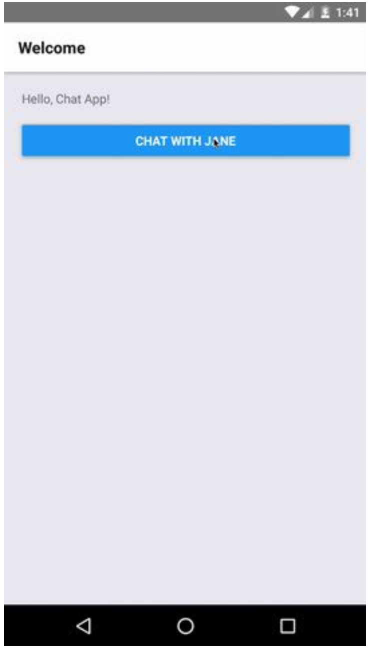
React

Experiences

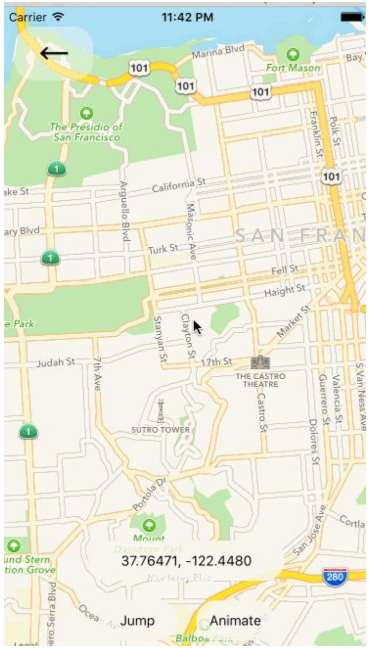
Recommendations

react-native-navigation

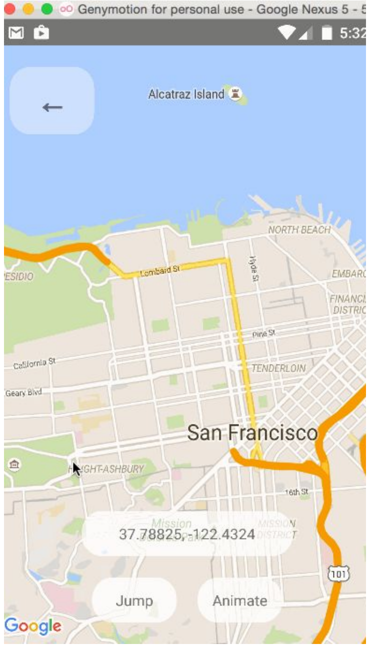




react-native-maps



37.76471, -122.4480



37.78825, -122.4324

Other libraries

What you should use:

- react-native-i18n ~> native locale in JS
- react-native-dotenv ~> environment variables in JS
- react-native-fbsdk ~> facebook SDK integration (cocoapods!)
- react-native-app-info ~> native app information in JS
- react-native-device-settings ~> device information and interaction in JS
- …

Tip #1

Start by implementing everything in JS for maximum productivity. Don’t over-optimize too early.

Source: https://medium.com/@talkol/performance-limitations-of-react-native-and-how-to-overcome-them-947630d7f440

Tip #2

Wait for the full product on a real device to see where your app falters. If traditional React optimizations fail, surgically move the troublesome parts to native.

Source: https://medium.com/@talkol/performance-limitations-of-react-native-and-how-to-overcome-them-947630d7f440

Tip #3

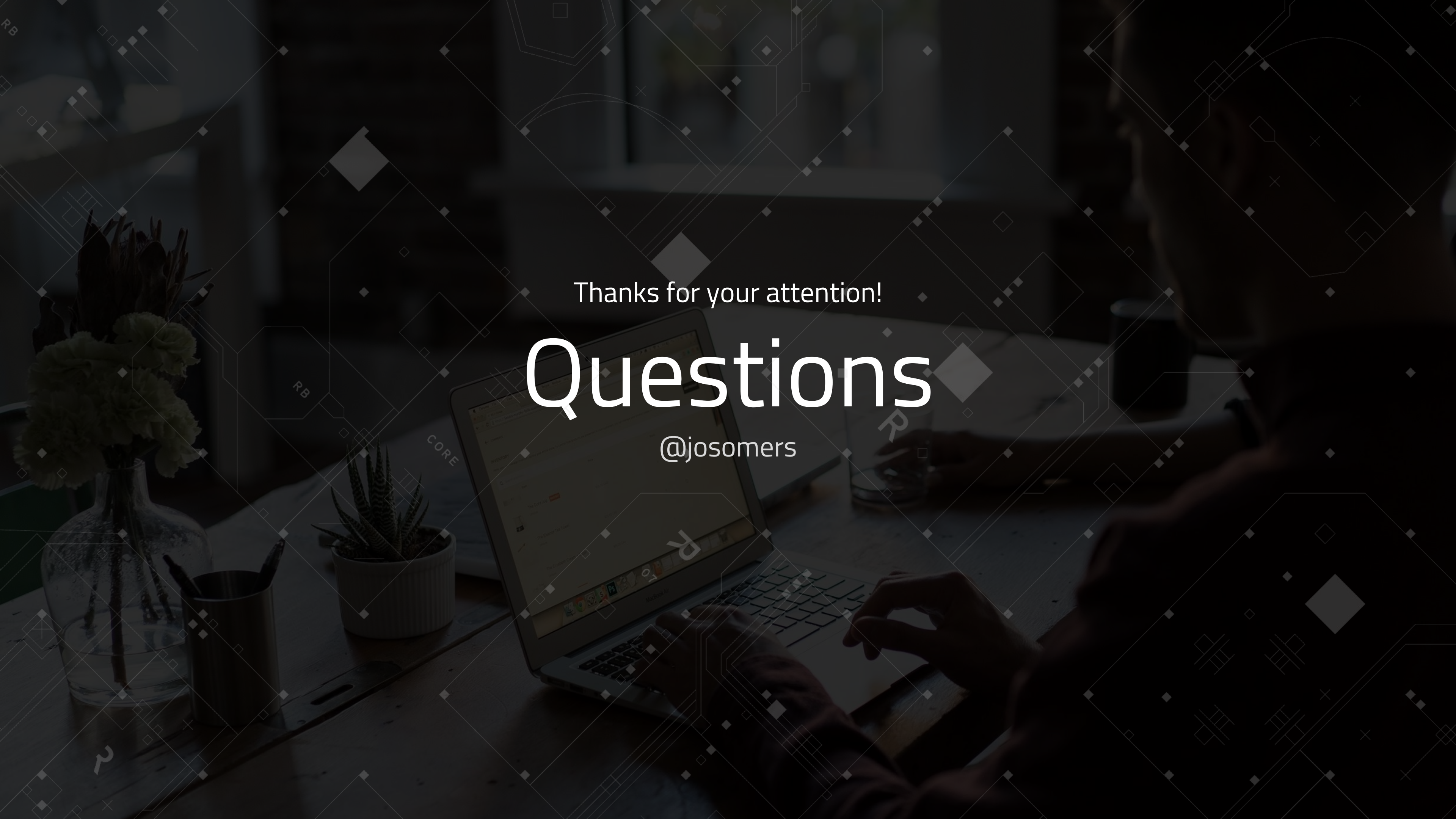
Encourage JS developers to dabble in native.

Source: https://medium.com/@talkol/performance-limitations-of-react-native-and-how-to-overcome-them-947630d7f440

Artcore by ArtSound

Demo

Showing the audio player, discussed in the previous slides.
Soon… in an App Store / Play Store near your…



Thanks for your attention!

Questions

@josomers