



Piter
2018

Дмитрий Пацура
lowl.io

Архитектура React Native
для JS-разработчика

Дмитрий Пацура

Пишу на многое чём)

<https://github.com/ovr>

GHubber

PHPSA



lowl.io

Типичный доклад про RN

О бизнес требованиях

Введение в технологию

Внутренности

Пять глав

- » Разработка на React Native
- » Изучение внутренностей 😜
- » Ограничение производительности
- » Анимации 🌈
- » Реальные примеры 😎

Что такое RN?

Build native mobile apps using JavaScript and React

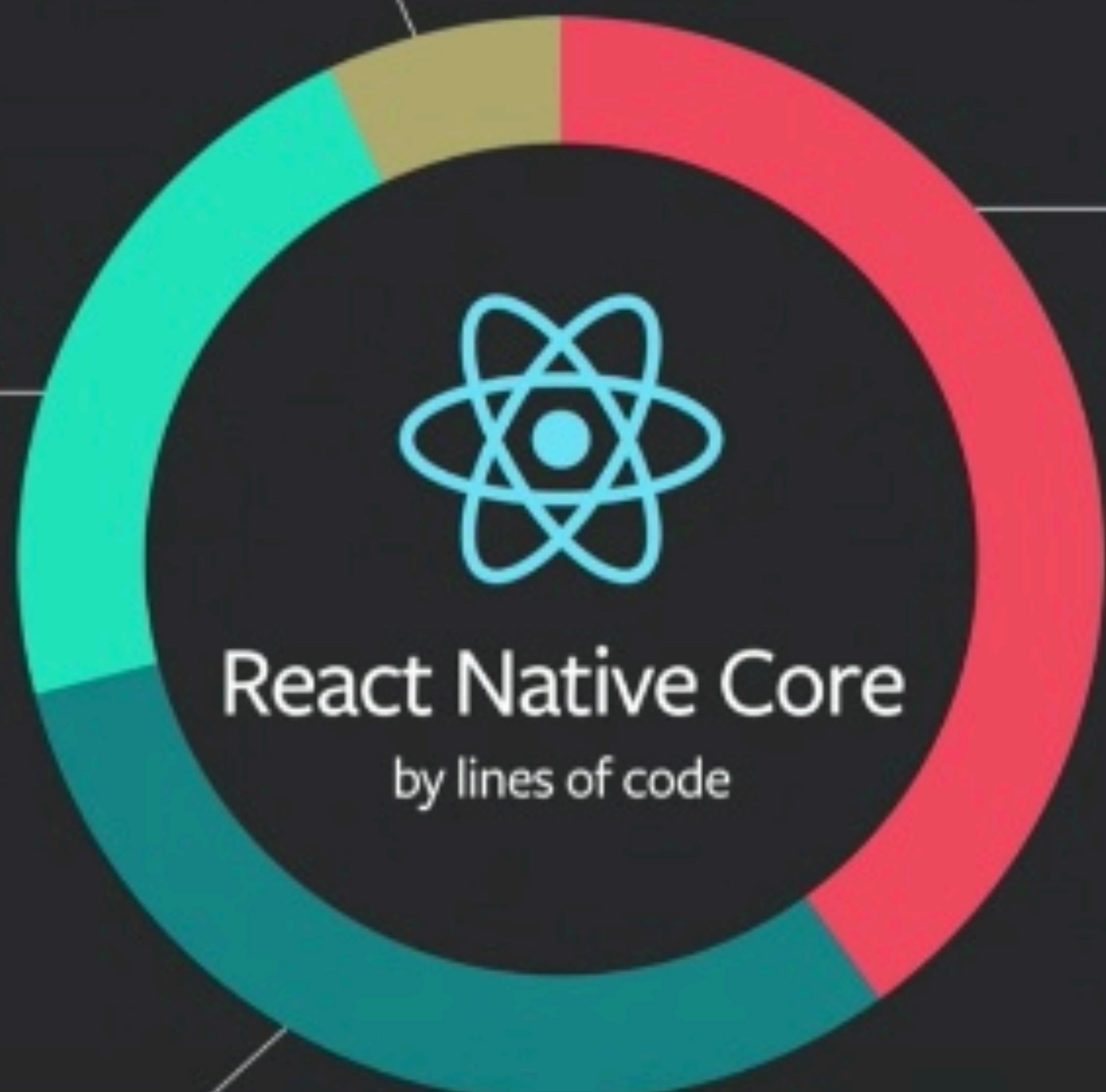
A React Native app is a real mobile app

C/C++ (7%)

Objective C (22%)

Java (31%)

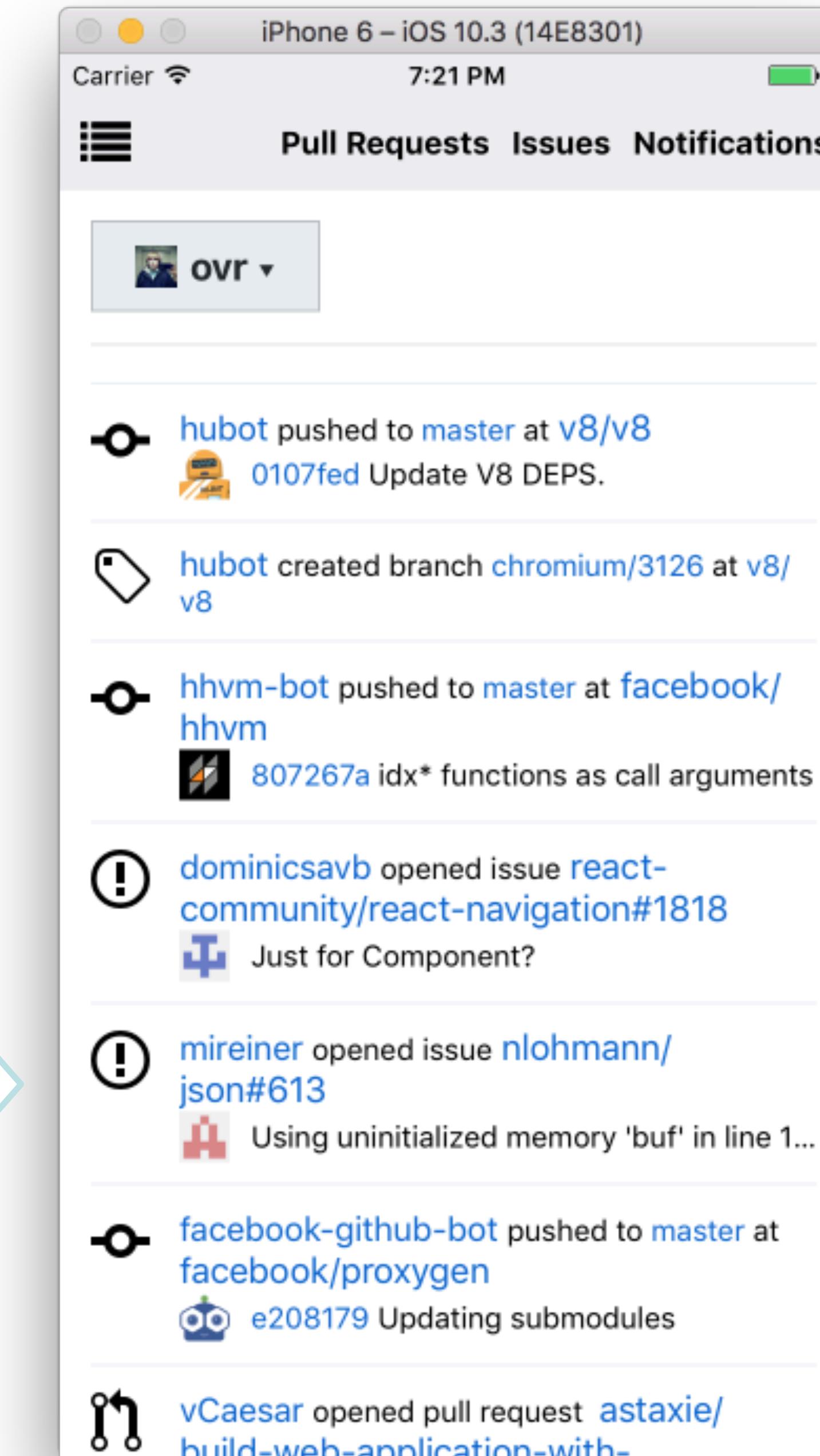
JavaScript (40%)



Metro Bundler
HTTP: 8081

Babel

Browser:
WebSocket



ES6+

JSX

Polyfill

Babel

Metro (Bundler)

```
'syntax-async-functions',
'syntax-class-properties',
'syntax-trailing-function-commas',
'transform-class-properties',
'transform-es2015-function-name',
'transform-es2015-arrow-functions',
'transform-es2015-block-scoping',
'transform-es2015-classes',
'transform-es2015-computed-properties',
'check-es2015-constants',
'transform-es2015-destructuring',
['transform-es2015-modules-commonjs', { strict: false, allowTopLevelThis: true }],
'transform-es2015-parameters',
'transform-es2015-shorthand-properties',
'transform-es2015-spread',
'transform-es2015-template-literals',
'transform-es2015-literals',
'transform-flow-strip-types',
'transform-object-assign',
'transform-object-rest-spread',
'transform-react-display-name',
'transform-react-jsx',
'transform-regenerator',
['transform-es2015-for-of', { loose: true }],
require('../transforms/transform-symbol-member'),
```

ES 2015

Common JS

Async / Await

JSX

```
export default class App extends Component<{}> {
  render = () => (
    <View style={styles.container}>
      <Image
        source={require('./assets/logo.png')}
      />
      <Text style={styles.spb}>Piter</Text>
      <Text style={styles.year}>2018</Text>
    </View>
  )
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: 'center',
    alignItems: 'center',
    backgroundColor: '#000',
  },
  spb: {
    fontSize: 56,
    color: '#fff'
  },
  year: {
    fontSize: 24,
    color: '#DB39C1'
  }
});
```



```
<View style={styles.container}>
  <Image
    source={require('./assets/logo.png')}
  />
  <Text style={styles.spb}>Piter</Text>
  <Text style={styles.year}>2018</Text>
</View>
```

<View> – блочный элемент
<Text> – текст

Android TV

iOS TV

Debug

Android

iOS

WebSocket

JavaScriptCore

Chrome / V8

JavaScriptCore

Evaluate JavaScript programs from within an app,
and support JavaScript scripting of your app.

iOS

Android

Built-in

External

<https://github.com/facebook/android-jsc>

Осторожно спойлеры!

iOS

Android

64 bit

32 bit

<https://github.com/SoftwareMansion/jsc-android-buildscripts>

```
Running application "ghubber" with appParameters
  ▼ action Navigation/RESET @ 17:28:23.612
    prev state ► Object {accountFeed: Object}
      action ► Object {type: "Navigation/RESET"}
      next state ► Object {accountFeed: Object}
  ▼ action ACCOUNT_FEED_REQUEST @ 17:28:23.673
    prev state ► Object {accountFeed: Object}
      action ► Object {type: "ACCOUNT_FEED_REQUEST"}
      next state ► Object {accountFeed: Object}
  ▼ action APP_ORGANIZATIONS_SUCCESS @ 17:28:23.673
    prev state ► Object {accountFeed: Object}
      action ► Object {type: "APP_ORGANIZATIONS_SUCCESS"}
      next state ► Object {accountFeed: Object}
  ▼ action ACCOUNT_FEED_SUCCESS @ 17:28:27.568
    prev state ► Object {accountFeed: Object}
      action ► Object {type: "ACCOUNT_FEED_SUCCESS"}
      next state ► Object {accountFeed: Object}
```

iPhone 6 – iOS 10.3 (14E8301) 5:27 PM :{}_. __DEV__

Carrier Pull Requests Issues Notifications

oVR

! tarang9211 opened issue react-community/react-navigation#1820
Modal swipe back goes to previous stack...

React Native: Development (RCTBatchedBridge)

Reload

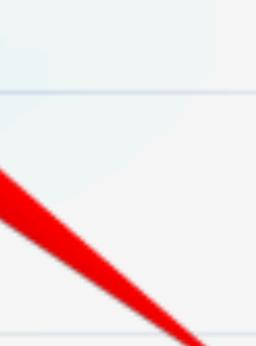
Debug JS Remotely

Enable Hot Reloading

Show Inspector

Show Perf Monitor

Cancel



Unhandled JS Exception: Error

```
throwOnImmutableMutation  
index.ios.bundle?platform...=true&minify=false:4052:14
```

```
_trimBreadcrumbs  
index.ios.bundle?platform...true&minify=false:49177:25
```

```
_trimPacket  
index.ios.bundle?platform...true&minify=false:49158:79
```

```
_sendProcessedPayload  
index.ios.bundle?platform...true&minify=false:49328:32
```

```
_send  
index.ios.bundle?platform...true&minify=false:49309:43
```

```
_processException  
index.ios.bundle?platform...true&minify=false:49135:19
```

```
_handleStackInfo  
index.ios.bundle?platform...true&minify=false:49061:31
```

```
captureException  
index.ios.bundle?platform...true&minify=false:48427:34
```

```
reportFatalError  
index.ios.bundle?platform...v=true&minify=false:685:37
```

```
__guard  
index.ios.bundle?platform...=true&minify=false:2482:36
```

```
callFunctionReturnFlushedQueue  
index.ios.bundle?platform...=true&minify=false:2365:19
```

```
git:(master*) $ react-native log-ios  
/projects/ovr/ghubber/node_modules (5ms)  
ing system. See log(1) for more information.  
rror>: CoreLocation: Error occurred while trying to retrieve motion  
ror>: □ nw_connection_get_connected_socket_block_invoke 1547 Connec  
ror>: □ nw_connection_get_connected_socket_block_invoke 1548 Connec  
ror>: □ nw_connection_get_connected_socket_block_invoke 1549 Connec  
ror>: □ nw_connection_get_connected_socket_block_invoke 1550 Connec  
ror>: □ nw_connection_get_connected_socket_block_invoke 1551 Connec  
ror>: □ nw_connection_get_connected_socket_block_invoke 1552 Connec  
ror>: □ nw_connection_get_connected_socket_block_invoke 1553 Connec  
ror>: □ nw_connection_get_connected_socket_block_invoke 1554 Connec  
ge[5216] <Warning>: Pasteboard change listener callback port <NSMach
```

Dismiss (ESC)

Reload JS (⌘R)

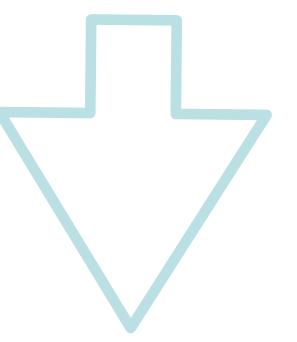
Copy (⌘C)

Изучение внутренностей 😜

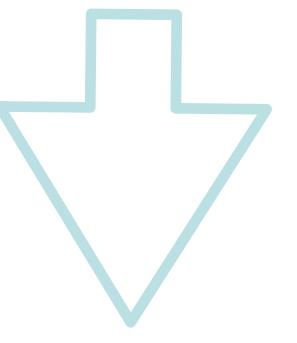
MainApplication.java

```
public class MainApplication extends Application {
    private final ReactNativeHost mReactNativeHost = new ReactNativeHost(this)
{
    @Override
    protected List<ReactPackage> getPackages() {
        return Arrays.asList(
            new MainReactPackage(),
        );
    }
};
```

Application



RNHost



JSC

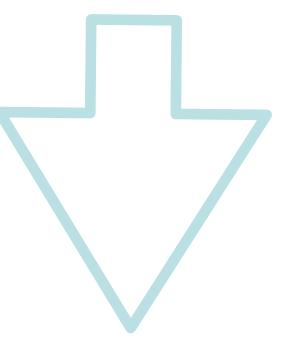
MainActivity

```
import com.facebook.react.ReactActivity;

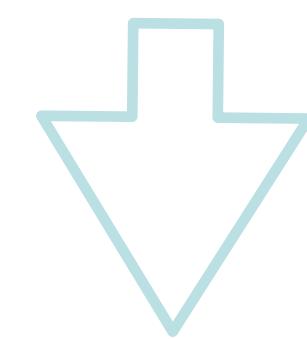
public class MainActivity extends ReactActivity {

    /**
     * Returns the name of the main component registered from JavaScript.
     * This is used to schedule rendering of the component.
     */
    @Override
    protected String getMainComponentName() {
        return "app_name";
    }
}
```

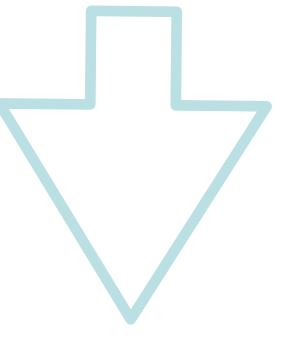
Application



RNHost



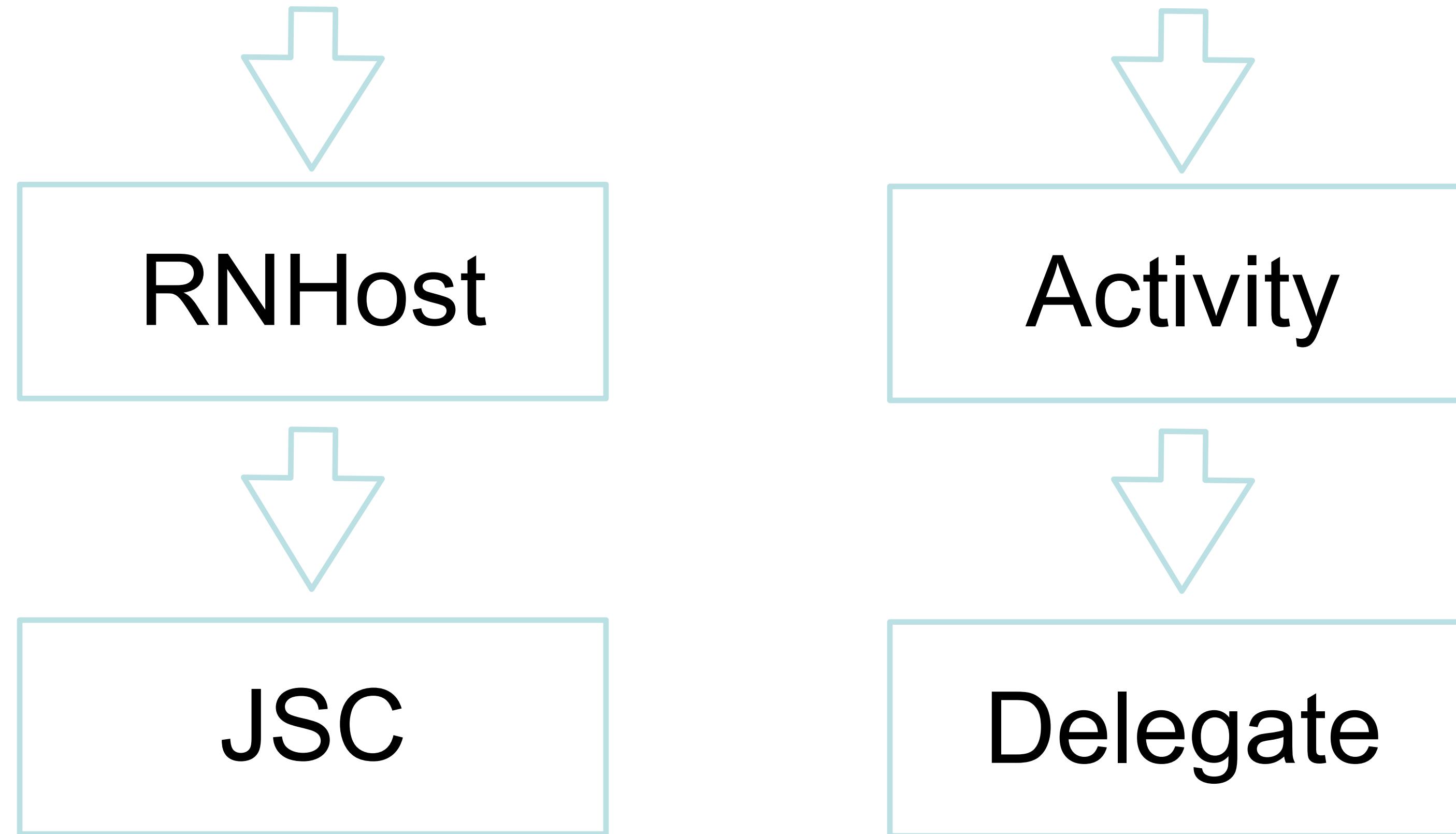
Activity



JSC

```
public abstract class ReactActivity extends Activity {  
  
    private final ReactActivityDelegate mDelegate;  
  
    protected ReactActivity() {  
        mDelegate = new ReactActivityDelegate(this, getMainComponentName());  
    }  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        mDelegate.onCreate(savedInstanceState);  
    }  
  
    @Override  
    protected void onPause() {  
        mDelegate.onPause();  
    }  
  
    @Override  
    protected void onResume() {  
        mDelegate.onResume();  
    }  
  
    @Override  
    protected void onDestroy() {  
        mDelegate.onDestroy();  
    }  
}
```

Application

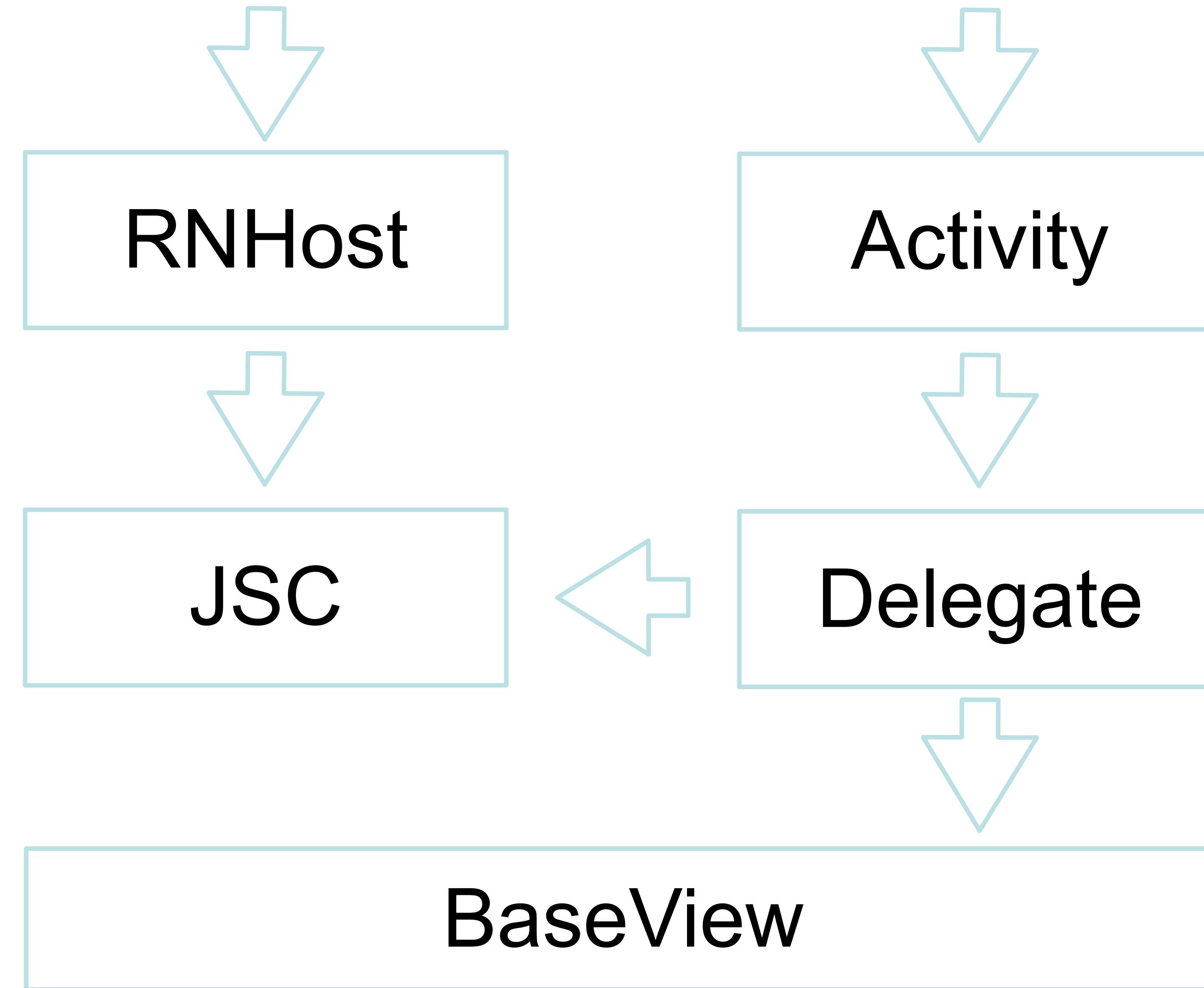


ReactActivityDelegate

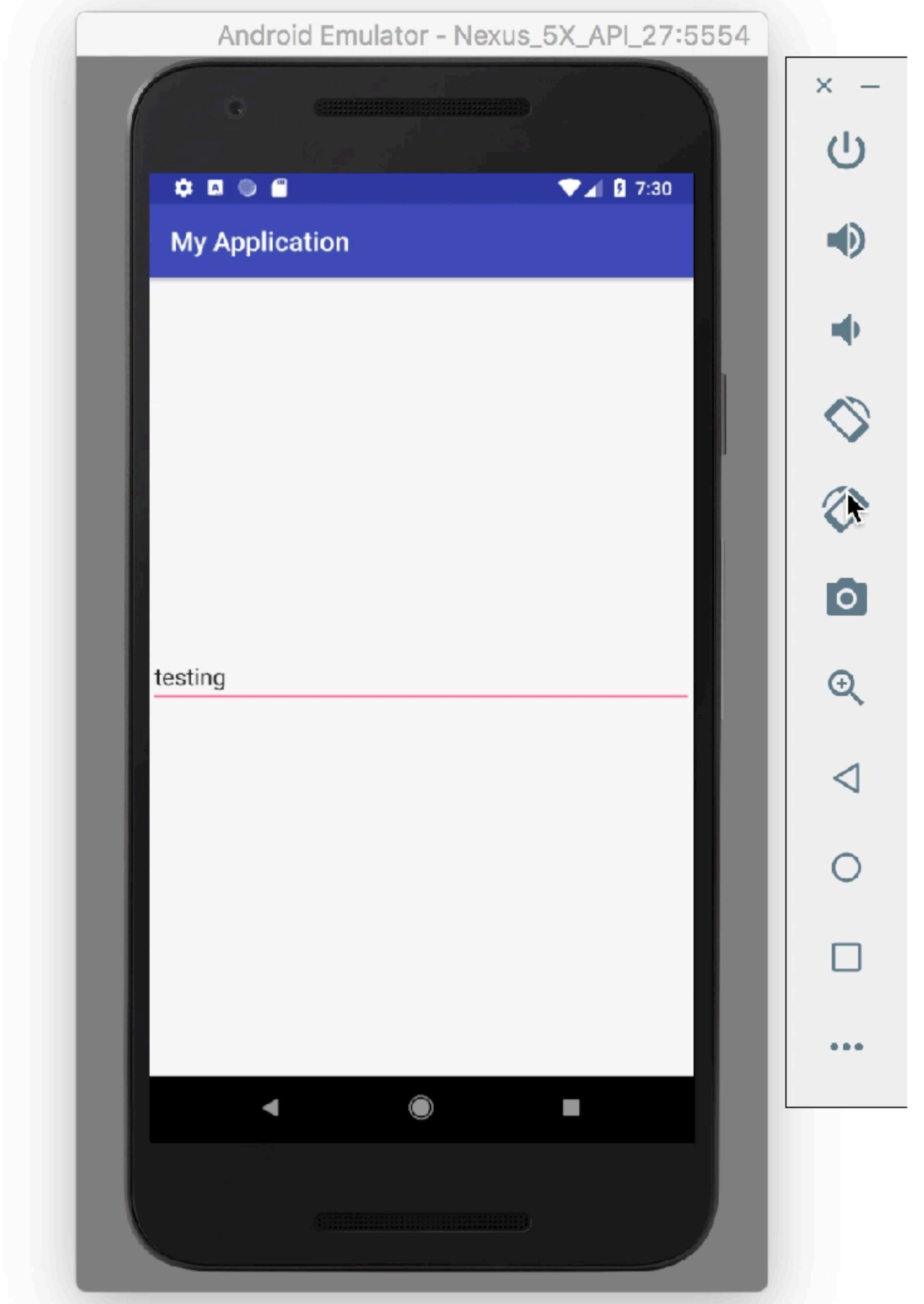
```
protected void onCreate(Bundle savedInstanceState) {  
    loadApp(mMainComponentName);  
}
```

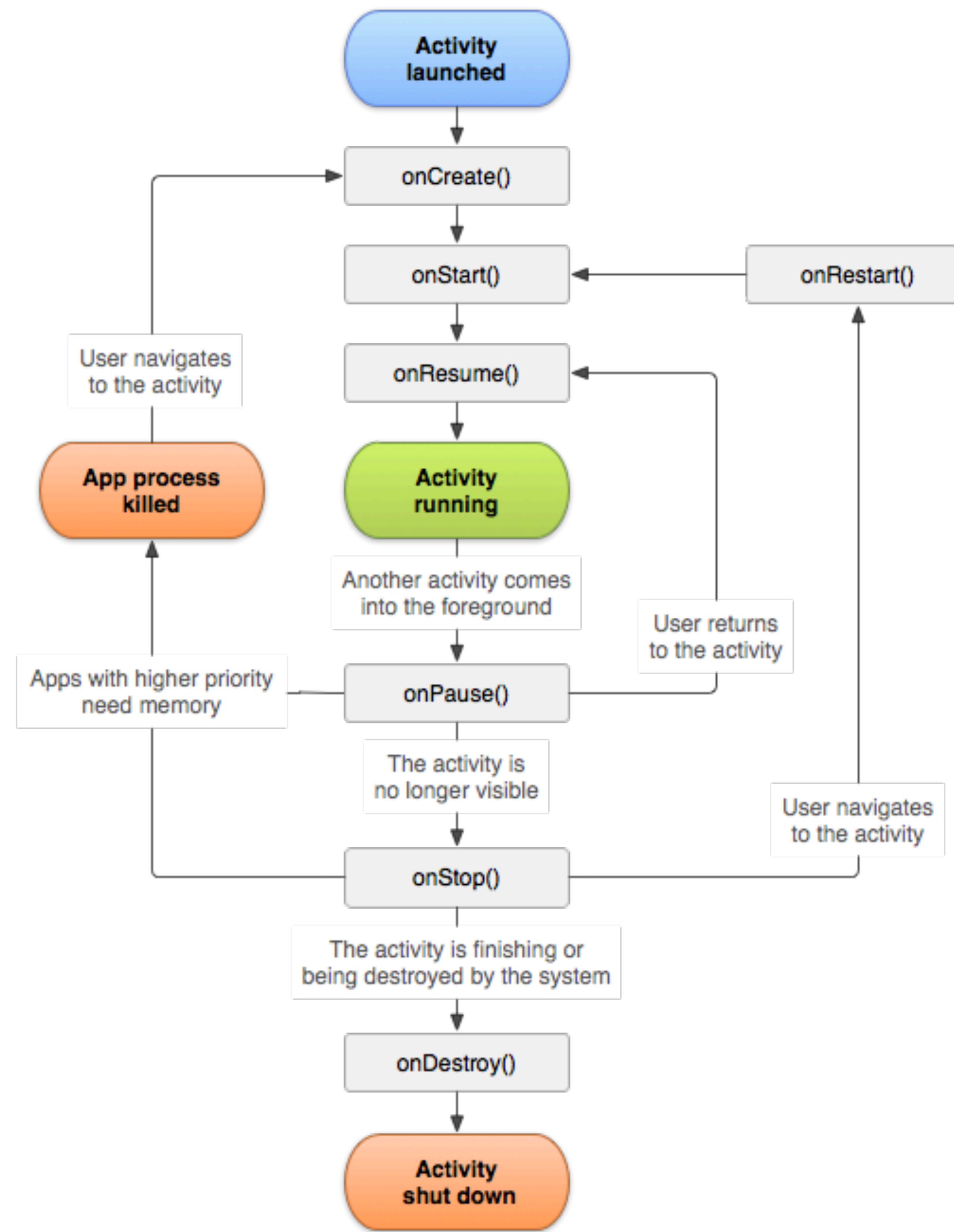
```
protected void loadApp(String appKey) {  
    mReactRootView = createRootView();  
    mReactRootView.startReactApplication(  
        getReactNativeHost().getReactInstanceManager(),  
        appKey,  
        getLaunchOptions());  
    getPlainActivity().setContentView(mReactRootView);  
}
```

Application



Есть ли Android разработчики в зале?





MAIN / UI



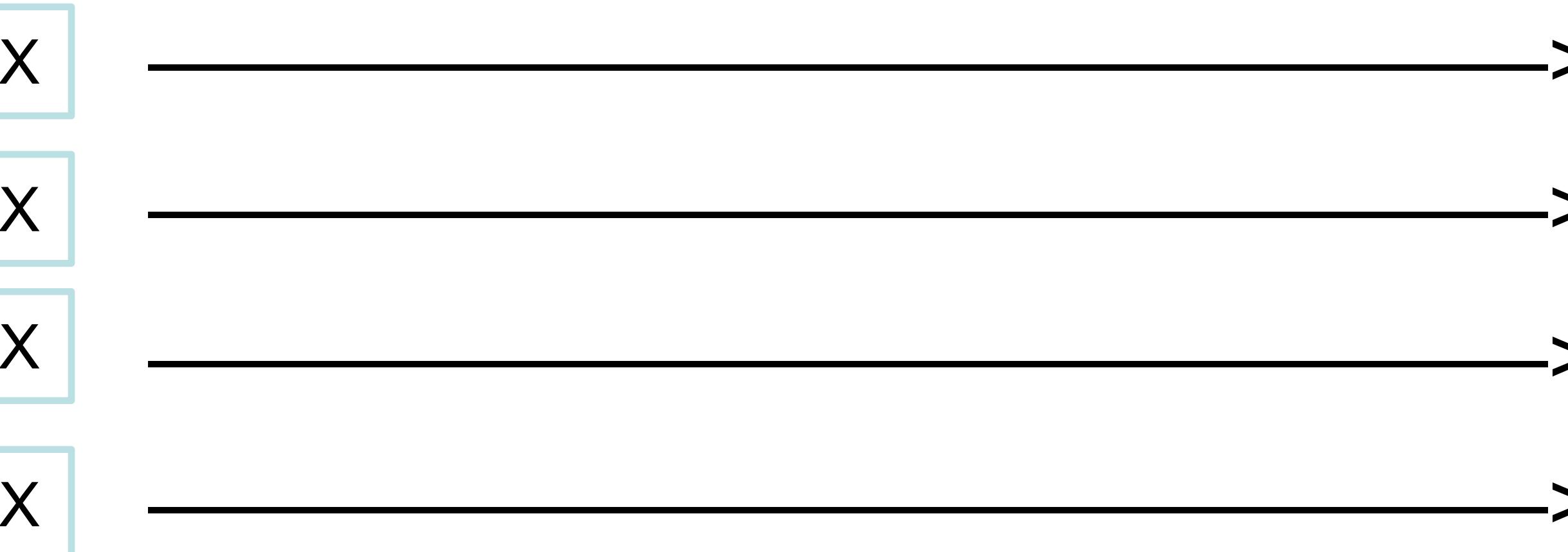
JSC (JS)



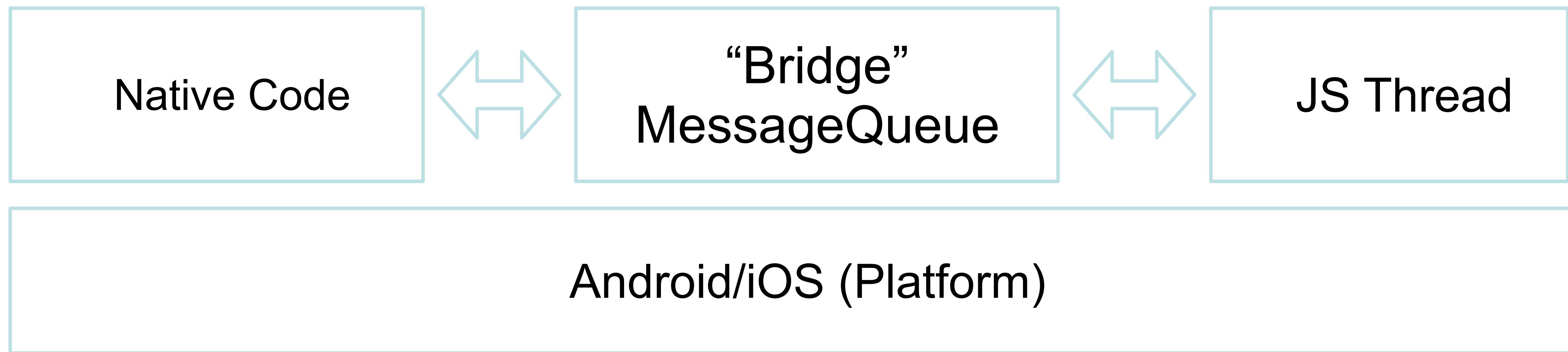
Bridge



**Native
Modules**



Коммуникация JS <=> Native



Коммуникация JS <=> Native

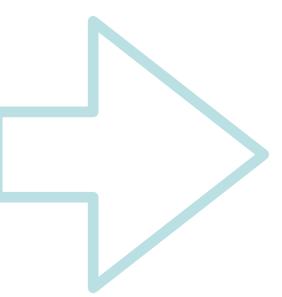
- » JavaScript -> Native
- » Native -> JavaScript

Вызов нативного кода из JS

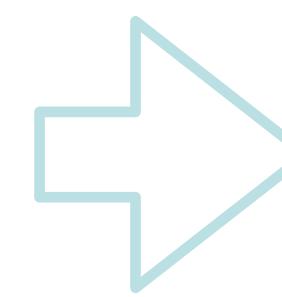
```
var RNDDeviceInfo = require('react-native')
.NativeModules
.RNDDeviceInfo;

export default {
  getUniqueId: function() {
    return RNDDeviceInfo.uniqueId;
  },
  getInstanceID: function() {
    return RNDDeviceInfo.instanceId;
  },
};
```

index.js

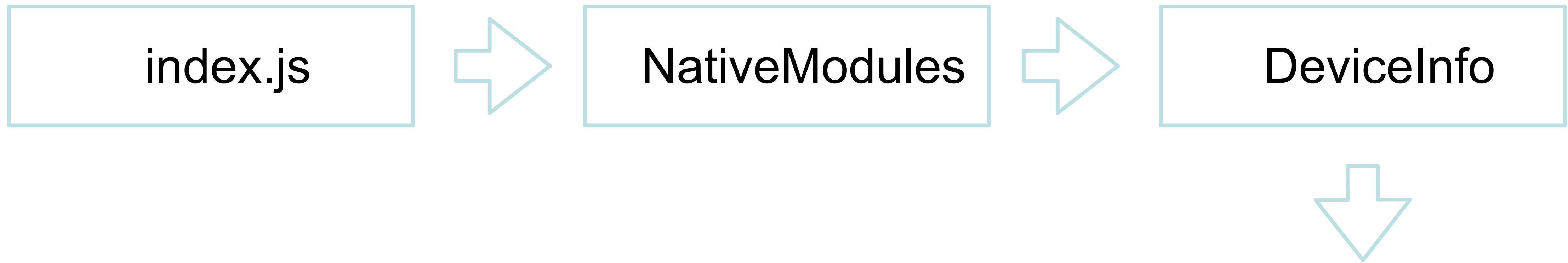


NativeModules



```
let NativeModules : {[moduleName: string]: Object} = {};  
  
if (global.nativeModuleProxy) {  
  NativeModules = global.nativeModuleProxy;  
} else {  
  // remote by debug JS  
}  
  
export NativeModules;
```

NativeModules = Proxy[]



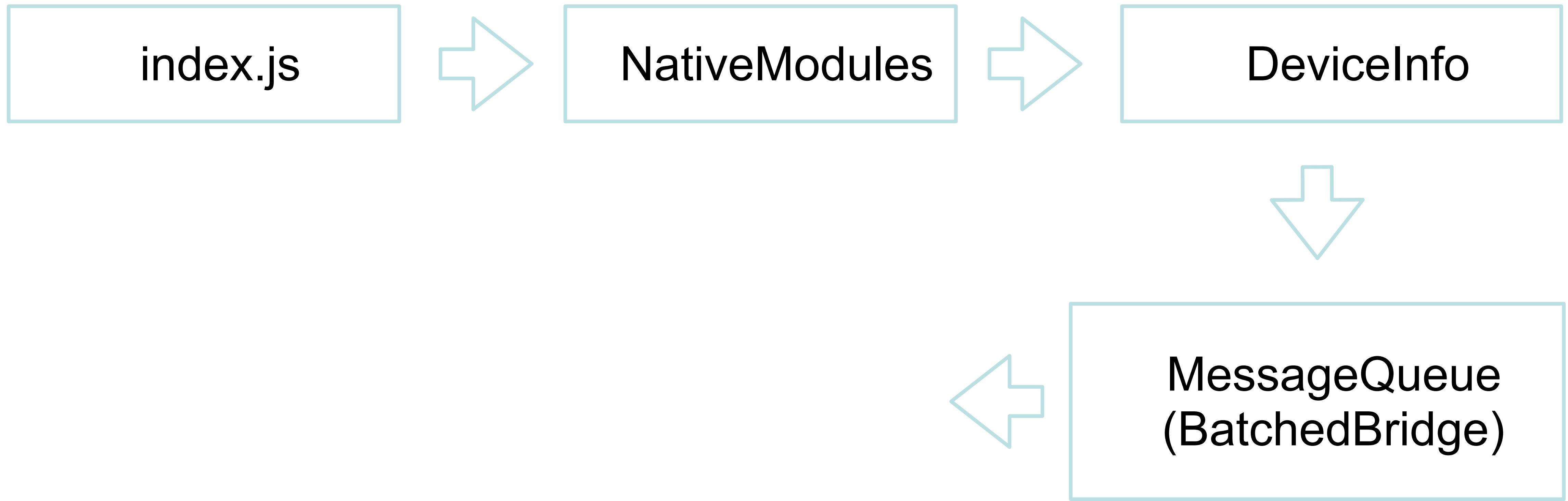
DeviceInfo = Proxy[] -> genModule -> genMethod

```
// promise
new Promise((resolve, reject) => {
  BatchedBridge.enqueueNativeCall(moduleID, methodID, args,
    (data) => resolve(data),
    (errorData) => reject(createErrorFromErrorData(errorData)));
})

// sync
global.nativeCallSyncHook(moduleID, methodID, args);

// async
BatchedBridge.enqueueNativeCall(moduleID, methodID, args, onFail, onSuccess);
```

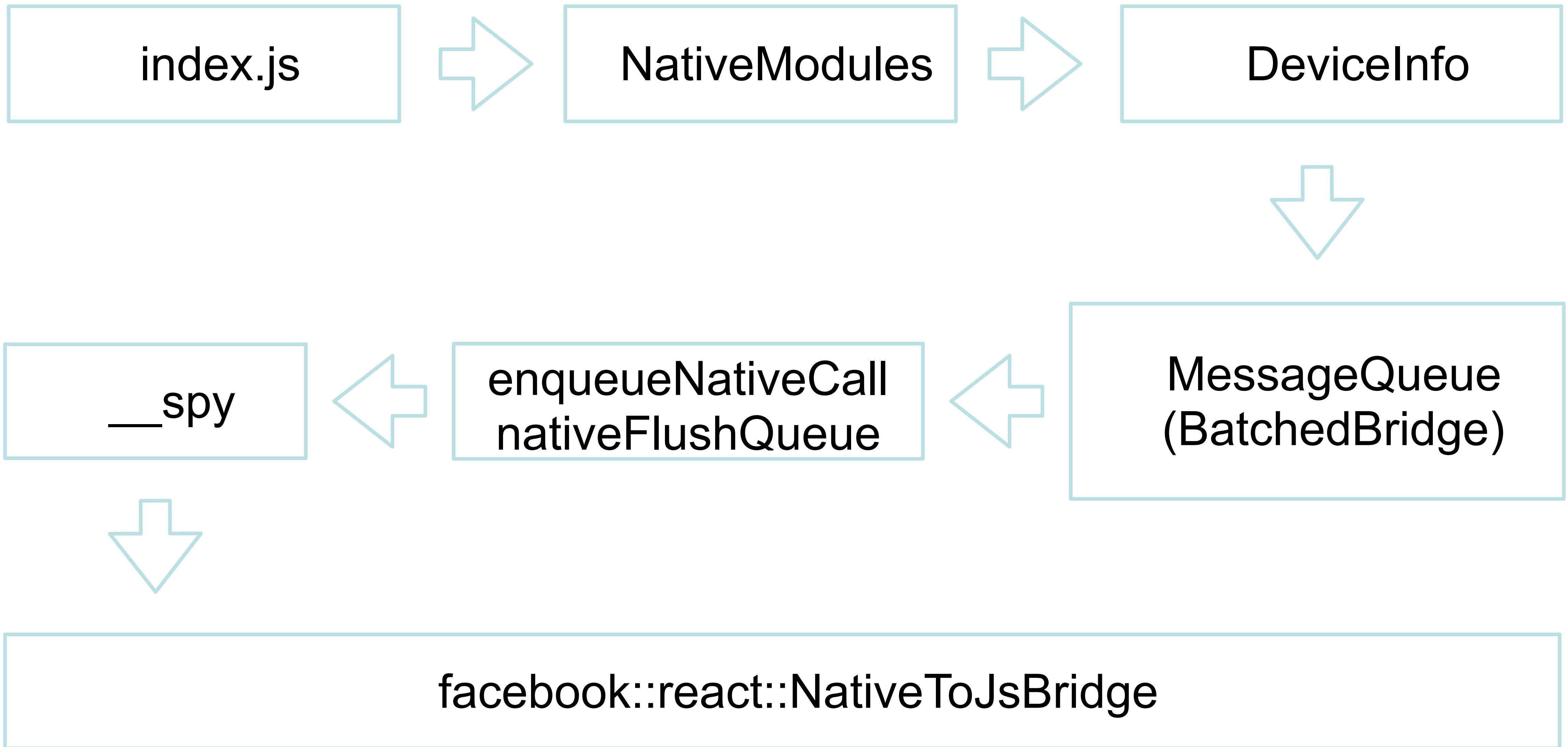
promise/sync/async



enqueueNativeCall

```
const MODULE_IDS = 0;
const METHOD_IDS = 1;
const PARAMS = 2;

this._queue = [[], [], [], 0];
this._queue[MODULE_IDS].push(moduleID);
this._queue[METHOD_IDS].push(methodID);
this._queue[PARAMS].push(params);
```



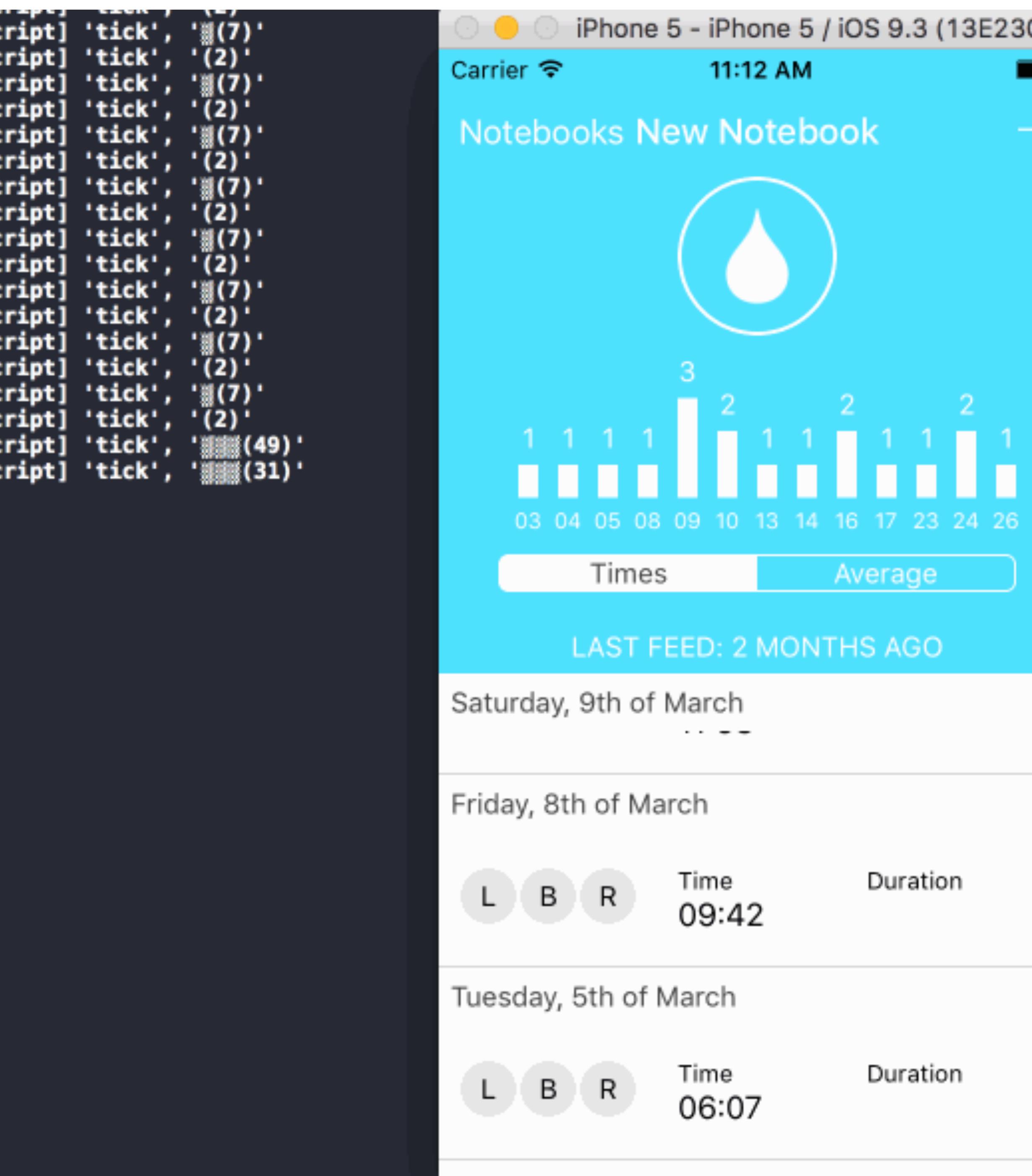
Как примерно это выглядит



MessageQueue - Spy Mode

```
import MessageQueue
from 'react-native/Libraries/BatchedBridge/MessageQueue'

MessageQueue.spy(true);
```



React Native Snoopy

Touch System Example - <TouchableWithoutFeedback />

```
1 N->JS RCTEventEmitter.receiveTouches({...})  
2 JS->N : Timing.createTimer([3,130,1520964688746,false])  
3 JS->N : Timing.createTimer([4,500,1520964688747,false])  
4 JS->N : UIManager.setJSResponder([3,false])  
5 N->JS : RCTEventEmitter.receiveTouches({...})  
6 JS->N : UIManager.clearJSResponder([])  
7 JS->N : Timing.deleteTimer([4])  
8 JS->N : Timing.deleteTimer([3])
```

Layout

```
N->JS : AppRegistry.runApplication(["demorn", {"rootTag":61, "initialProps":{}}])  
JS->N : UIManager.createView([2,"RCTRawText",61,{"text":"Welcome to CodeFest 2018!"}])  
JS->N : UIManager.createView([3,"RCTText",61,{"fontSize":42,"textAlign":"center","margin":10,"accessible":true,"allowFontScaling":true,"ellipsizeMode":"tail"}])  
JS->N : UIManager.setChildren([3,[2]])  
JS->N : UIManager.createView([4,"RCTView",61,{"flex":1,"justifyContent":"center","alignItems":"center","backgroundColor":4294311167}])  
JS->N : UIManager.setChildren([4,[3]])  
JS->N : UIManager.createView([5,"RCTView",61,{"pointerEvents":"box-none","flex":1}])  
JS->N : UIManager.setChildren([5,[4]])  
JS->N : UIManager.createView([6,"RCTView",61,{"flex":1,"pointerEvents":"box-none"}])  
JS->N : UIManager.setChildren([6,[5]])  
JS->N : UIManager.setChildren([61,[6]])
```

JSX

```
<View style={styles.container}>
  <Text>
    Welcome to HolyJS 2018!
  </Text>
</View>
```

JSX -> JS

```
<View style={styles.container}>
  <Text>
    Welcome to CodeFest 2018!
  </Text>
</View>
```

```
React.createElement(
  View,
  { style: styles.container },
  React.createElement(
    Text,
    null,
    "Welcome to CodeFest 2018!"
  )
);
```

JSX -> UI



react-reconciler

This is an experimental package for creating custom React renderers.

Its API is not as stable as that of React, React Native, or React DOM, and does not follow the common versioning scheme.

Use it at your own risk.

```
var Reconciler = require('react-reconciler');

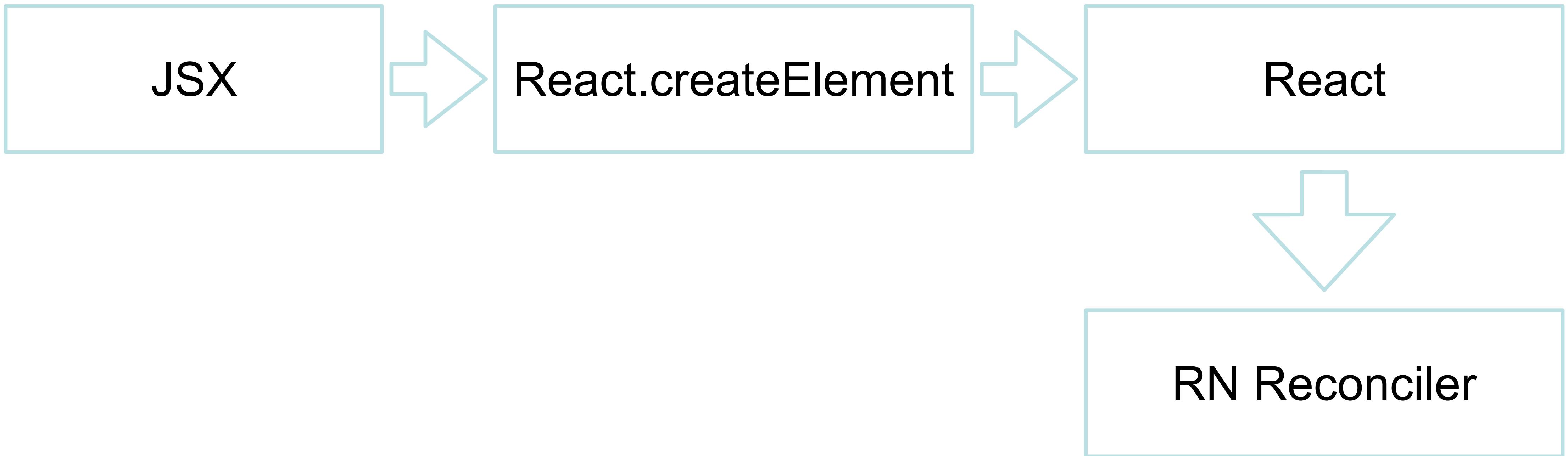
var ReconcilerConfig = {
  // You'll need to implement some methods here.
  // See below for more information and examples.
};

var MyRenderer = Reconciler(ReconcilerConfig);

var RendererPublicAPI = {
  render(element, container, callback) {
    // Call MyRenderer.updateContainer() to schedule changes on the roots.
    // See ReactDOM, React Native, or React ART for practical examples.
  }
};

module.exports = RendererPublicAPI;
```

JSX -> UI



Renderer/ReactNative

```
var NativeRenderer = reactReconciler({
  createInstance: function(...) {
    UIManager.createView(
      hostContext,
      type.uiViewClassName,
      rootContainerInstance,
      viewConfig
    );
  },
  createTextInstance: function(...) {
    UIManager.createView(..., {
      text: text
    });
  },
  finalizeInitialChildren: function(parentInstance) {
    UIManager.setChildren(parentInstance._nativeTag, nativeTags);
  },
  ...
})
```

<Text /> What is it?

```
var RCTText = createReactNativeComponentClass(  
  viewConfig.uiViewClassName,  
  () => viewConfig,  
);  
var RCTVirtualText = RCTText;  
  
const Text = createReactClass({  
  displayName: 'Text',  
  render(): React.Element<any> {  
    return <RCTText {...props} />;  
  },  
});
```

FabricUIManager and UIManager

FabricUIManager

- `appendChild`
- `appendChildToSet`
- `createNode`
- `cloneNode`
- `cloneNodeWithNewProps`
- `cloneNodeWithNewChildrenAndProps`
- `cloneNodeWithNewChildren`

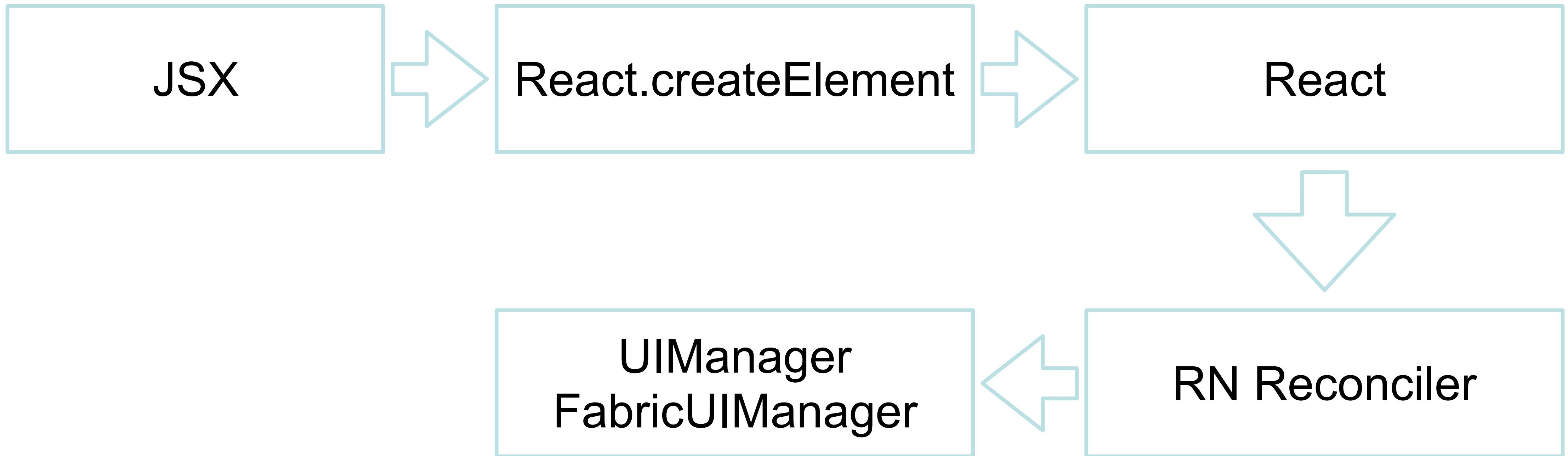
UIManager

- `UIManager.measure`
- `UIManager.measureInWindow`
- `UIManager.measureLayout`
- `UIManager.updateView`

Shared

Java/Objc

JSX -> UI



Yoga

PERFORMANCE

CROSS PLATFORM

EASY TO LEARN

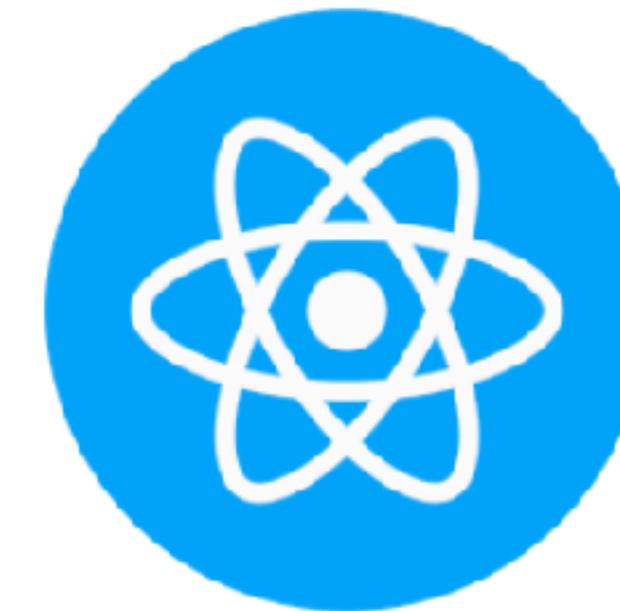
Used by



Litho

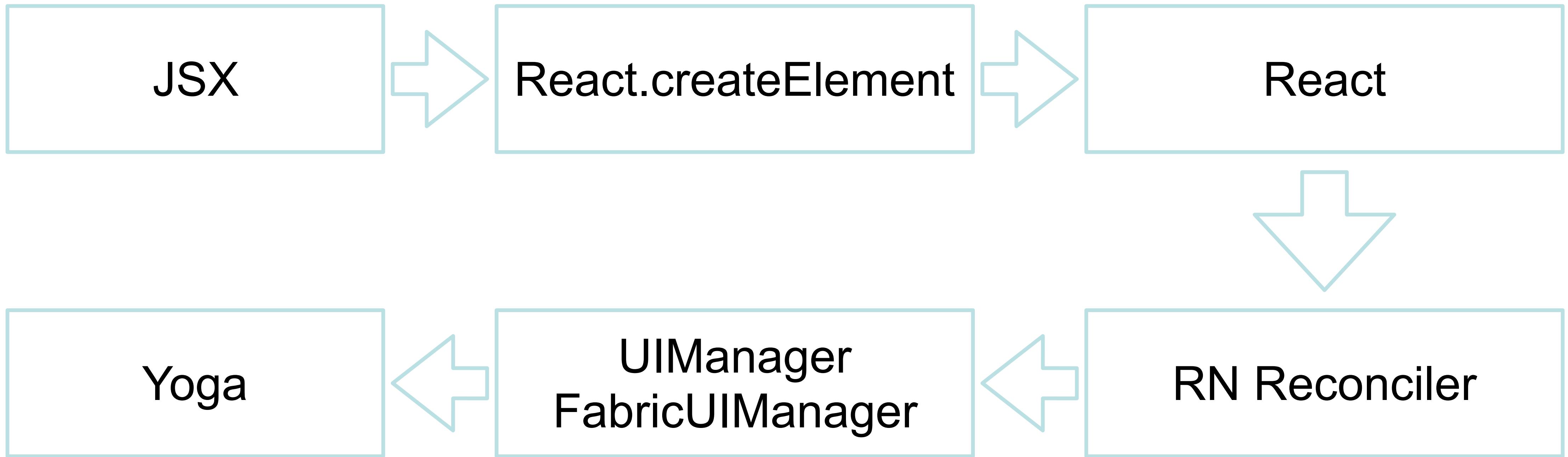


ComponentKit



React Native

JSX -> UI



Вызов JS из нативного кода

Вызов JS из нативного кода

AppState

AppState can tell you if the app is in the foreground or background, and notify you when the state changes.

AppState is frequently used to determine the intent and proper behavior when handling push notifications.

```
class AppState extends NativeEventEmitter {  
  constructor() {  
    super(RCTAppState);  
  
    this.addListener(  
      'appStateDidChange',  
      (appStateData) => {  
        this.currentState = appStateData.app_state;  
      }  
    );  
  }  
}
```

```
@ReactModule(name = "AppState")
public class AppStateModule extends ReactContextBaseJavaModule
    implements LifecycleEventListener {

    @Override
    public void onHostResume() {
        mAppState = "active";
        sendAppStateChangeEvent();
    }

    @Override
    public void onHostPause() {
        mAppState = "background";
        sendAppStateChangeEvent();
    }

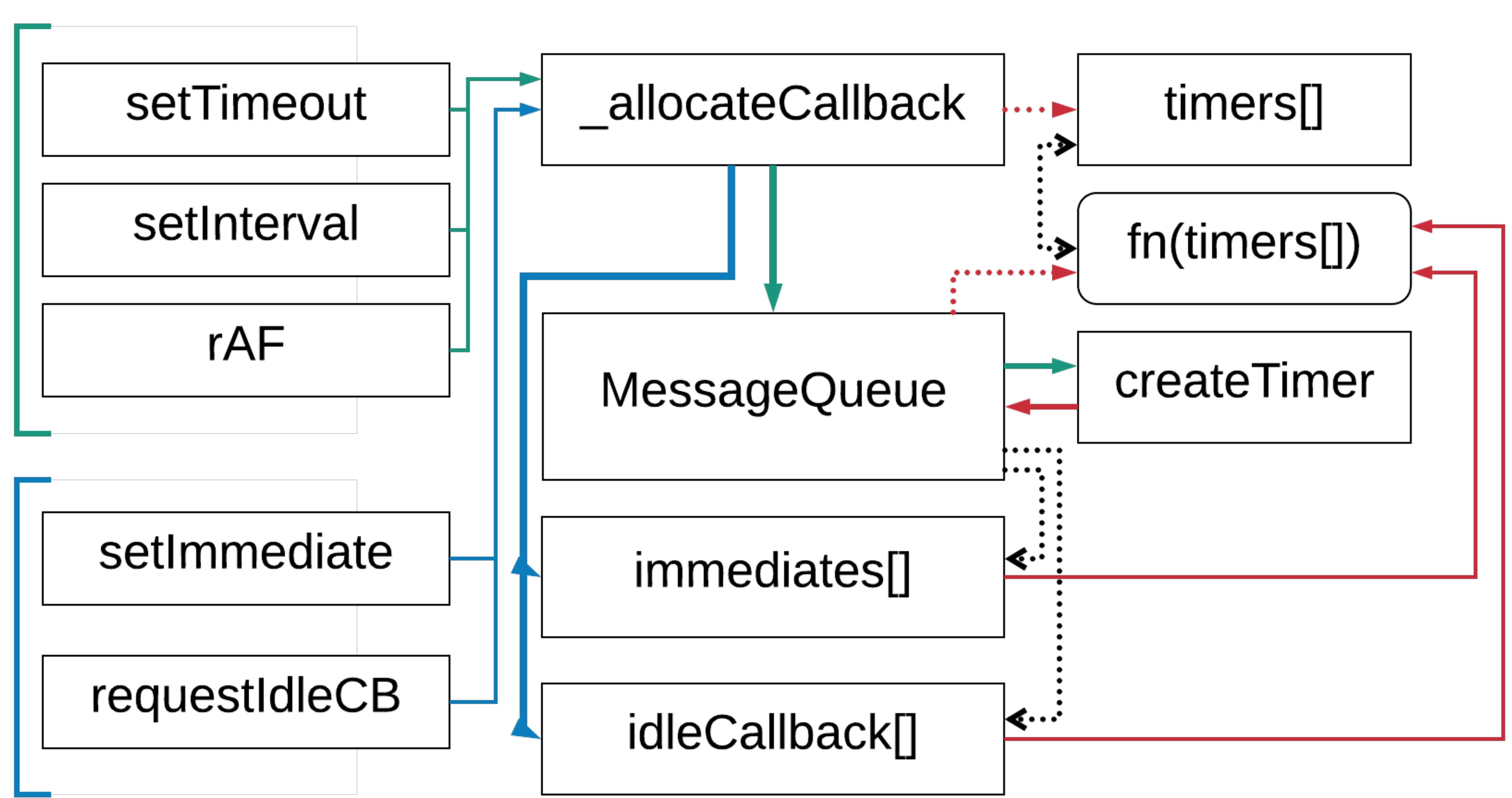
    private void sendAppStateChangeEvent() {
        WritableMap appState = Arguments.createMap();
        appState.putString("app_state", mAppState);

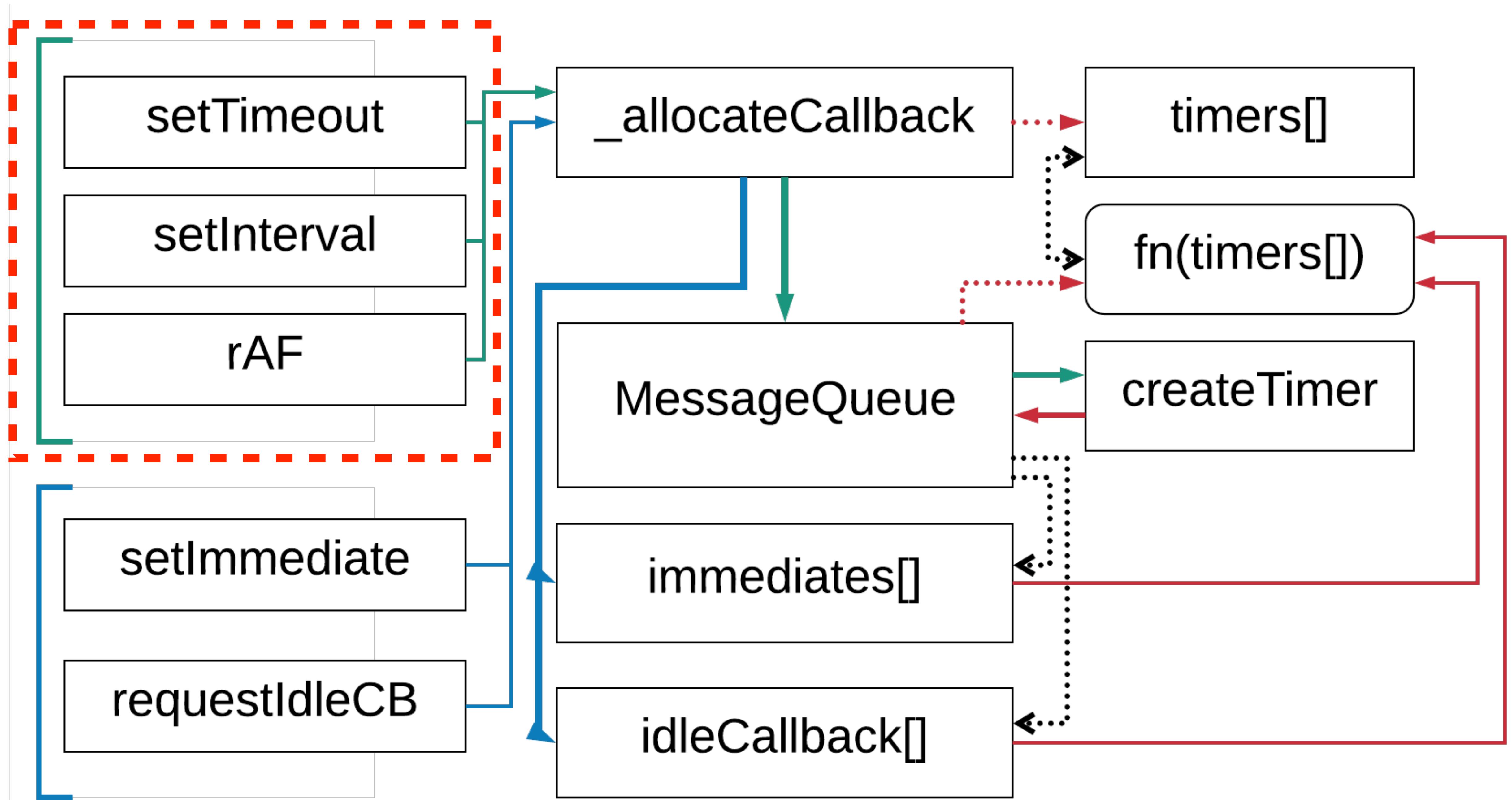
        getReactApplicationContext().getJSModule(RCTDeviceEventEmitter.class)
            .emit("AppStateDidChange", appState);
    }
}
```

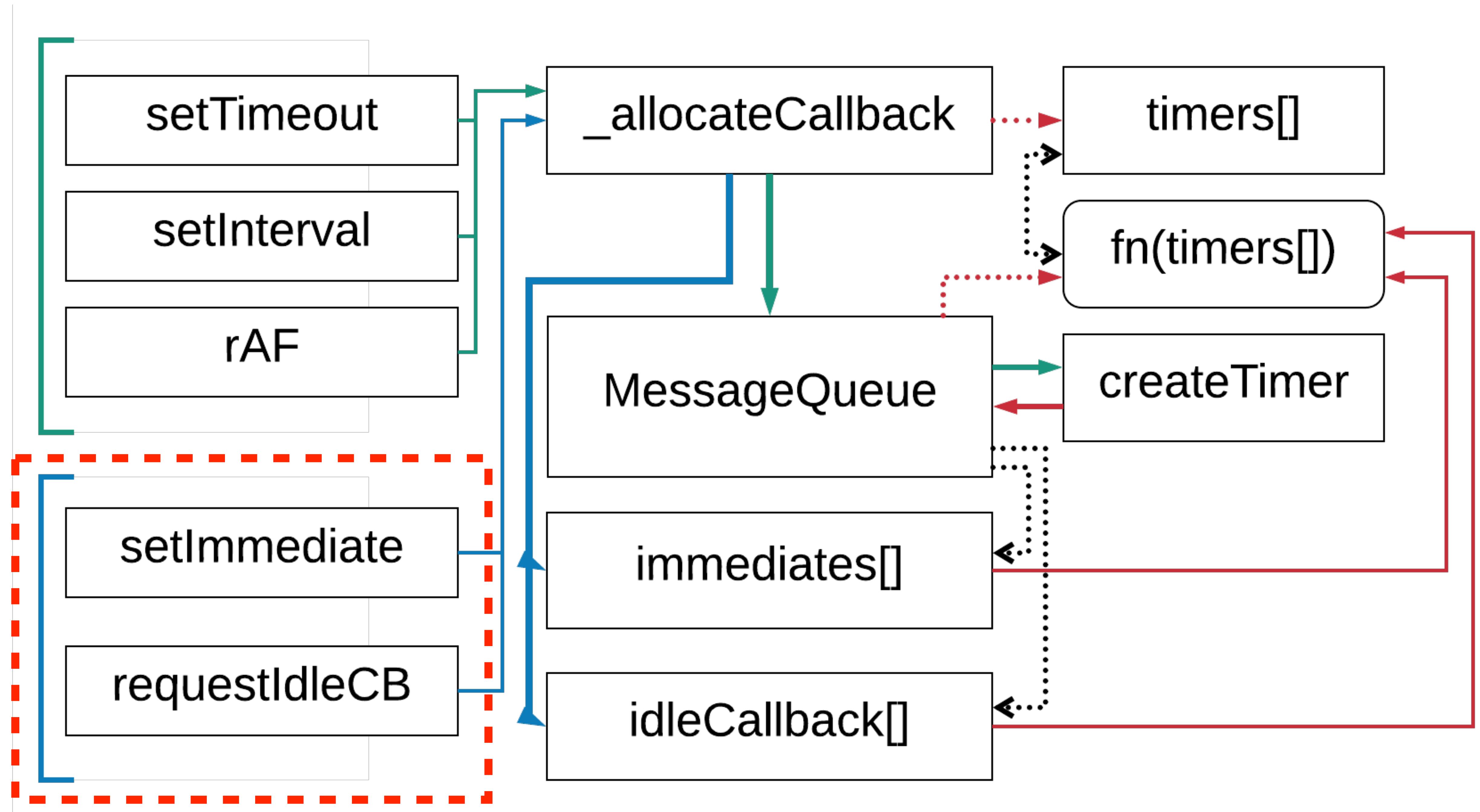
JavaScript Timers

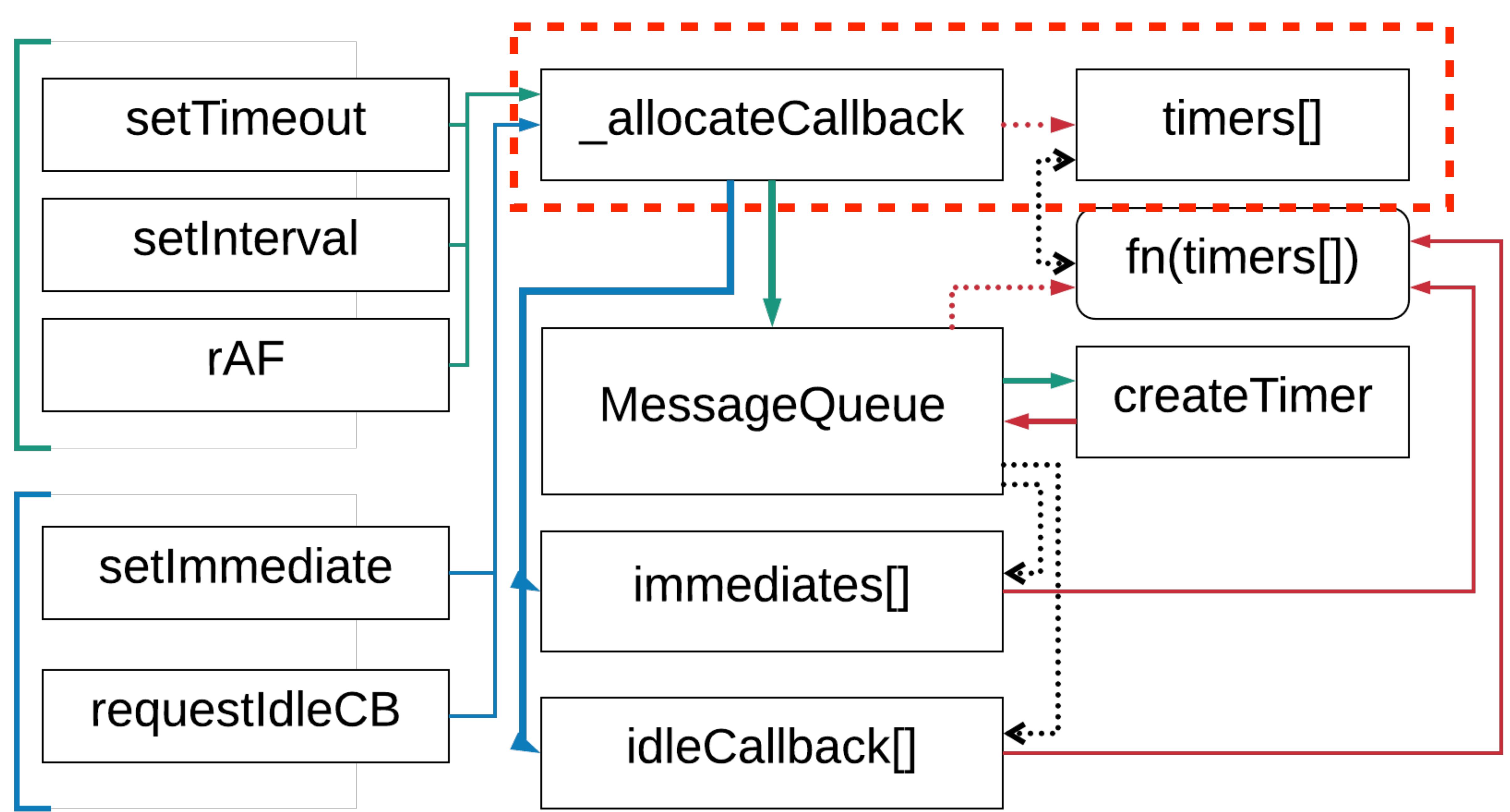
JSTimers.js

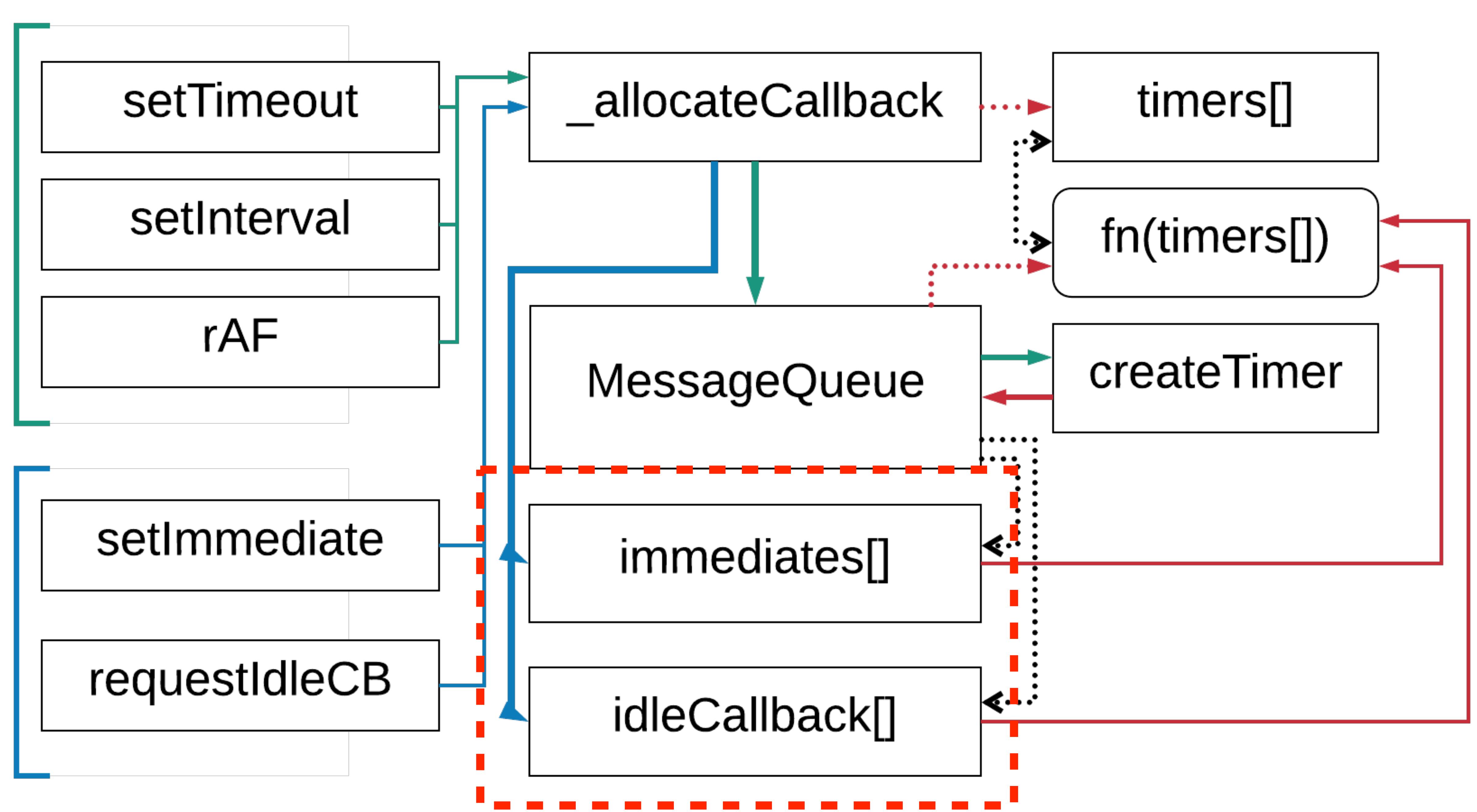
- setTimeout -> NativeModule.Timing.createTimer
- setInterval -> NativeModule.Timing.createTimer
- setImmediate -> immediates: Array<number>
- requestAnimationFrame -> NativeModule.Timing.createTimer
- requestIdleCallback -> requestIdleCallbacks: Array<number>

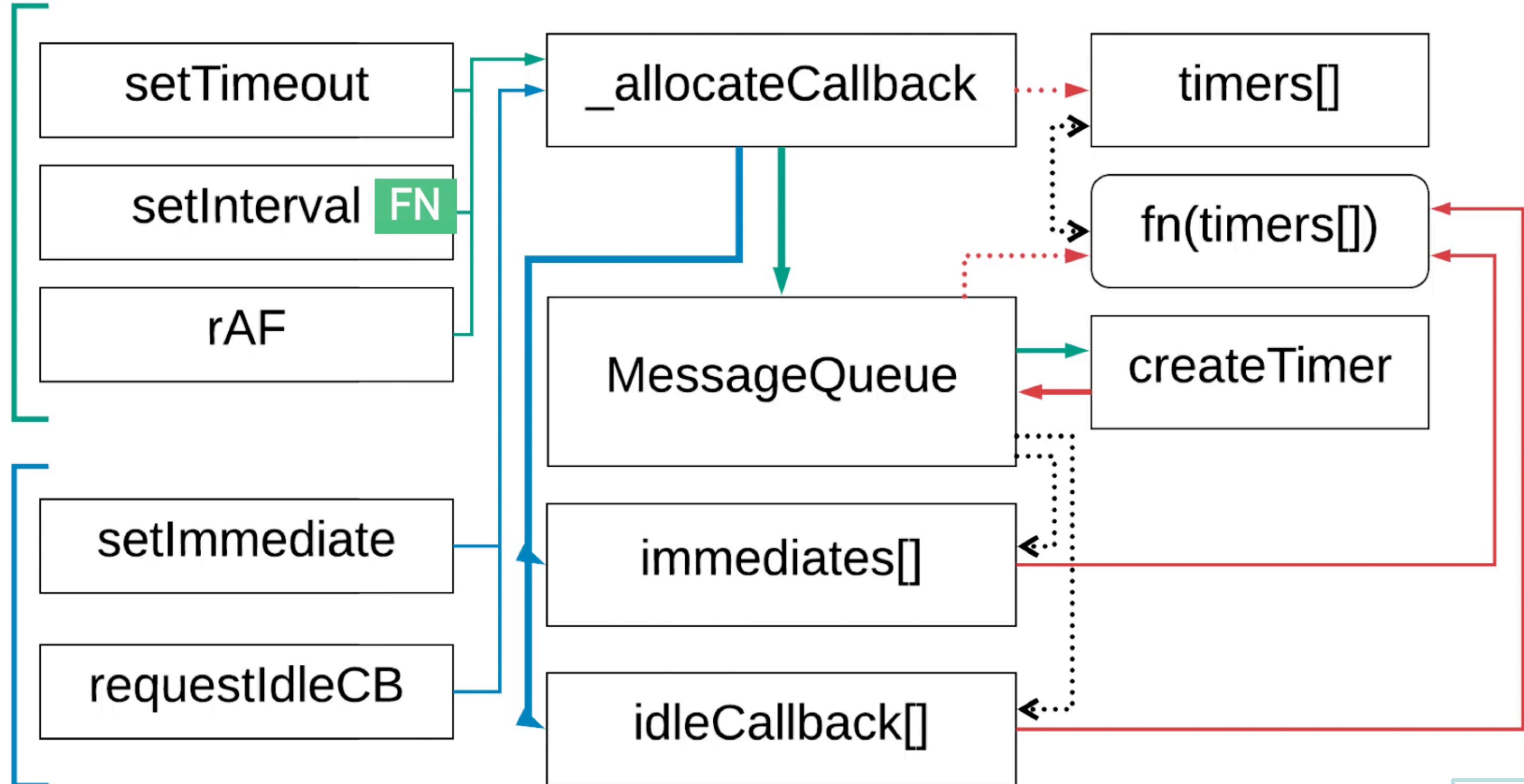


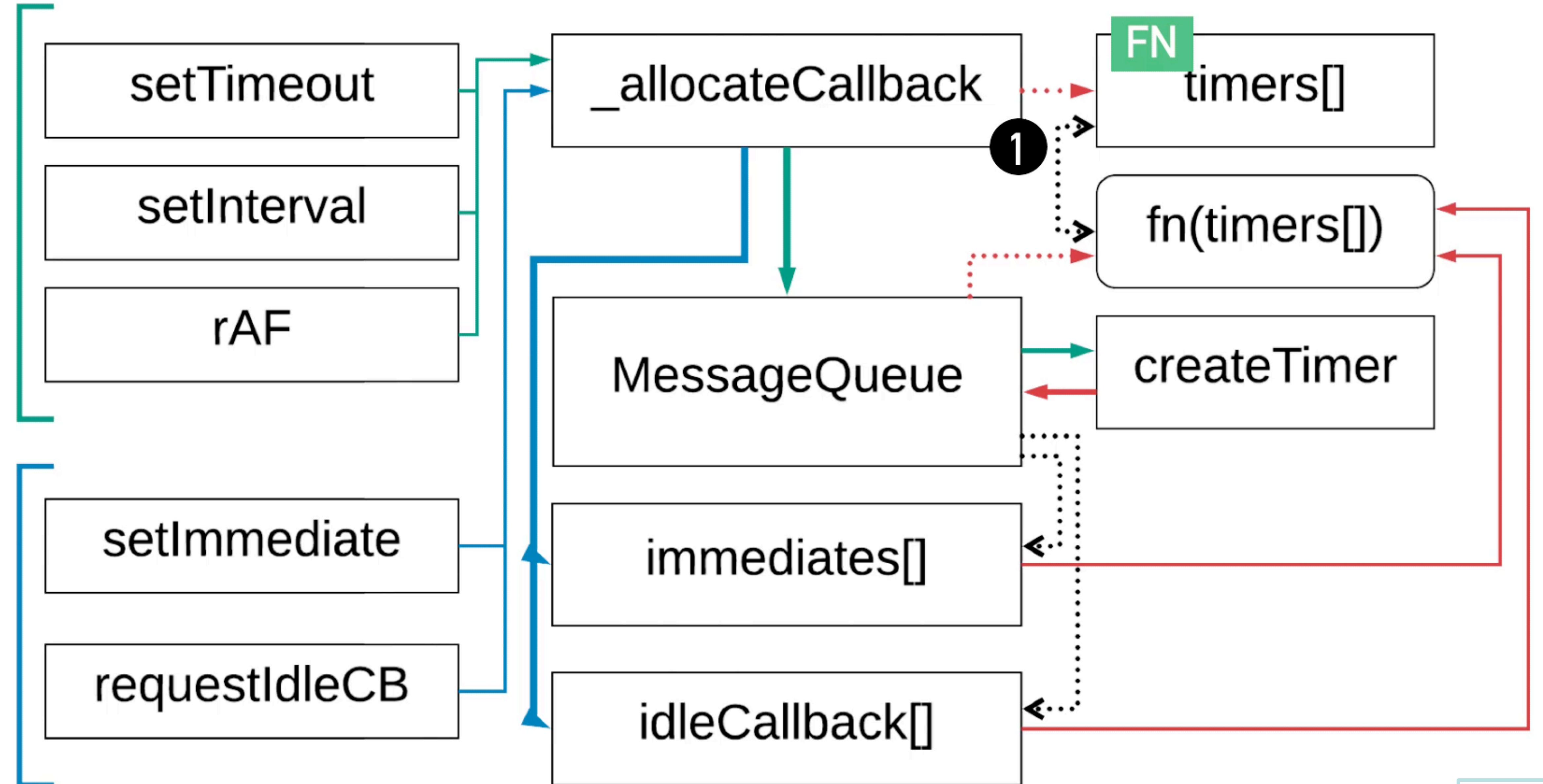


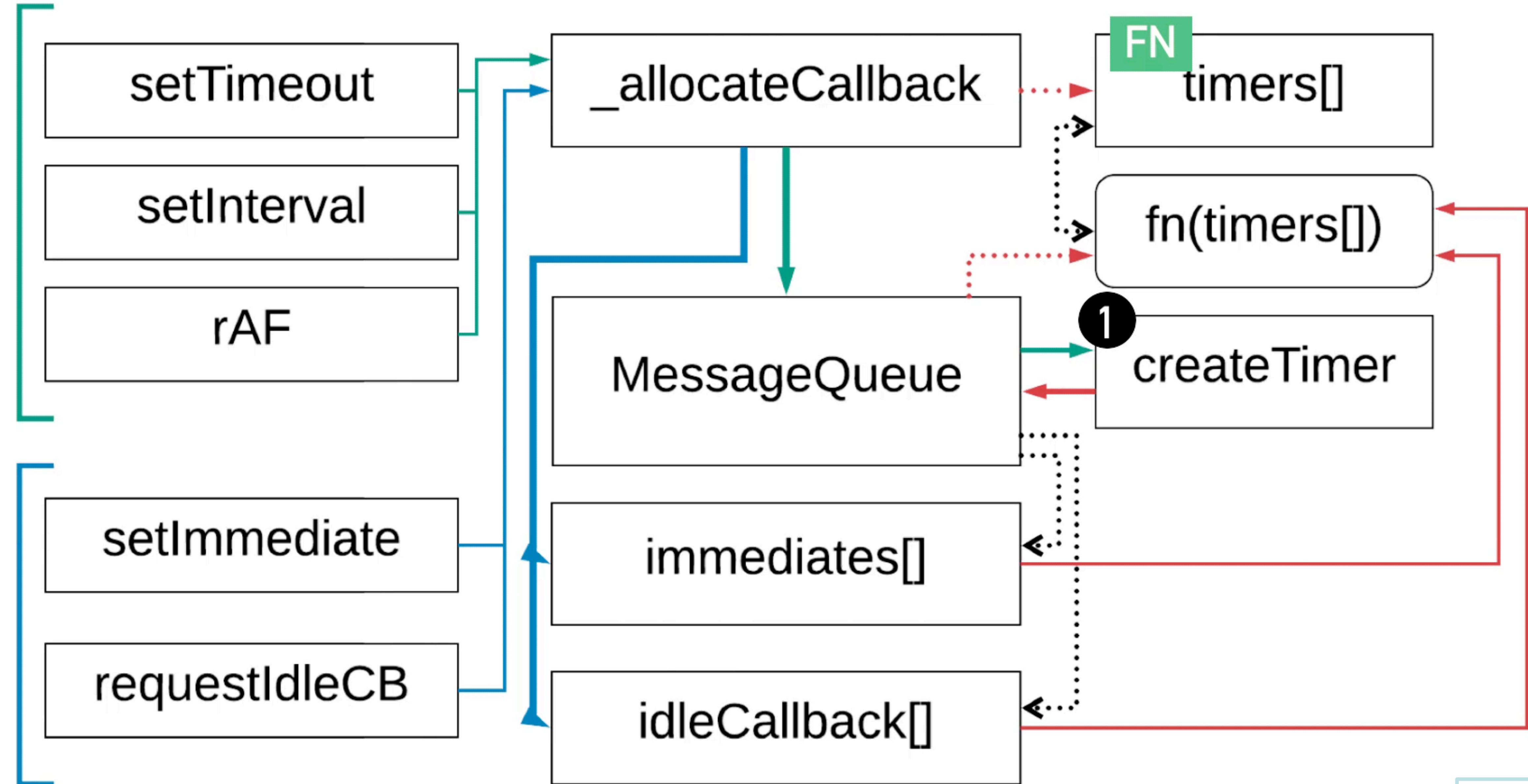


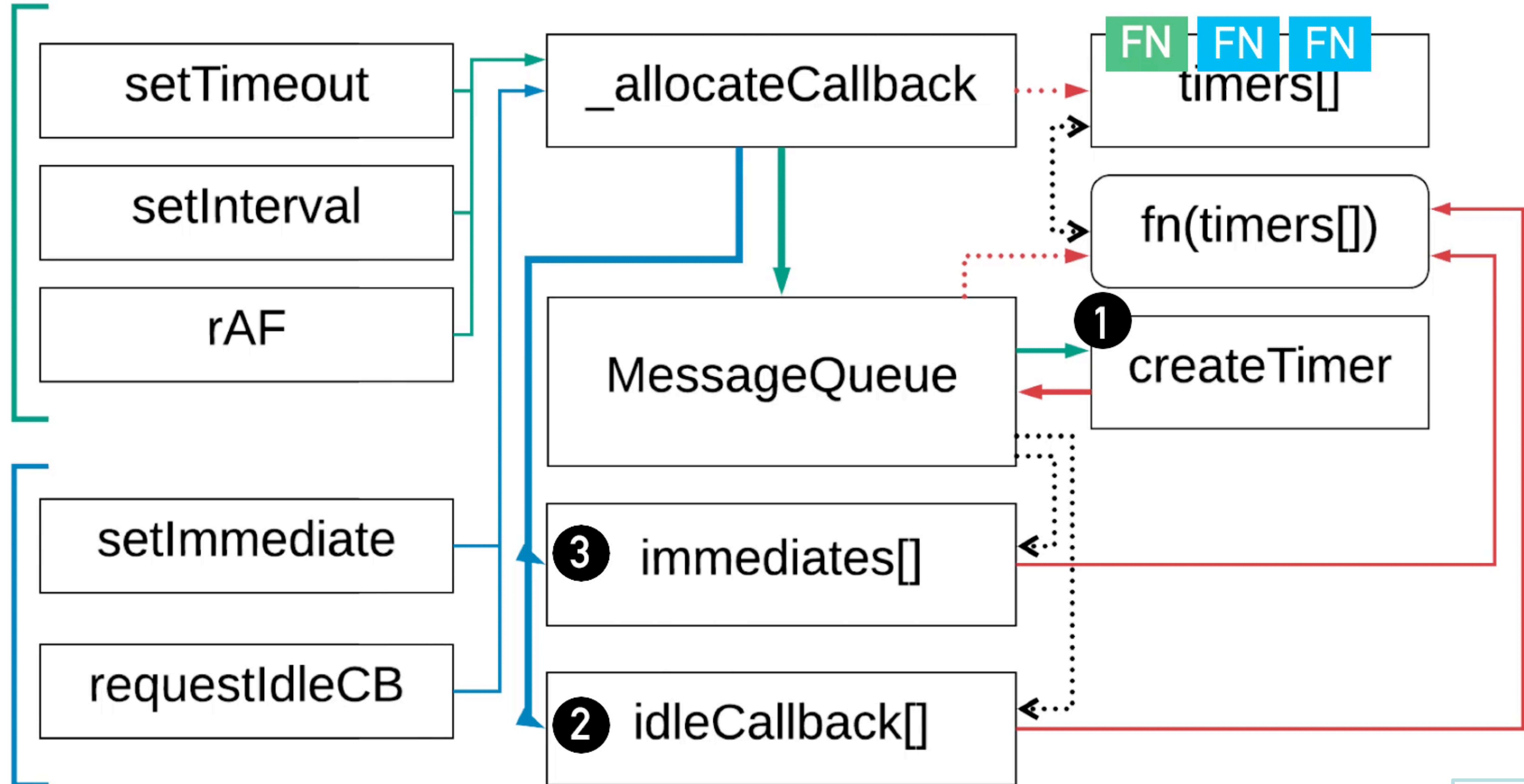




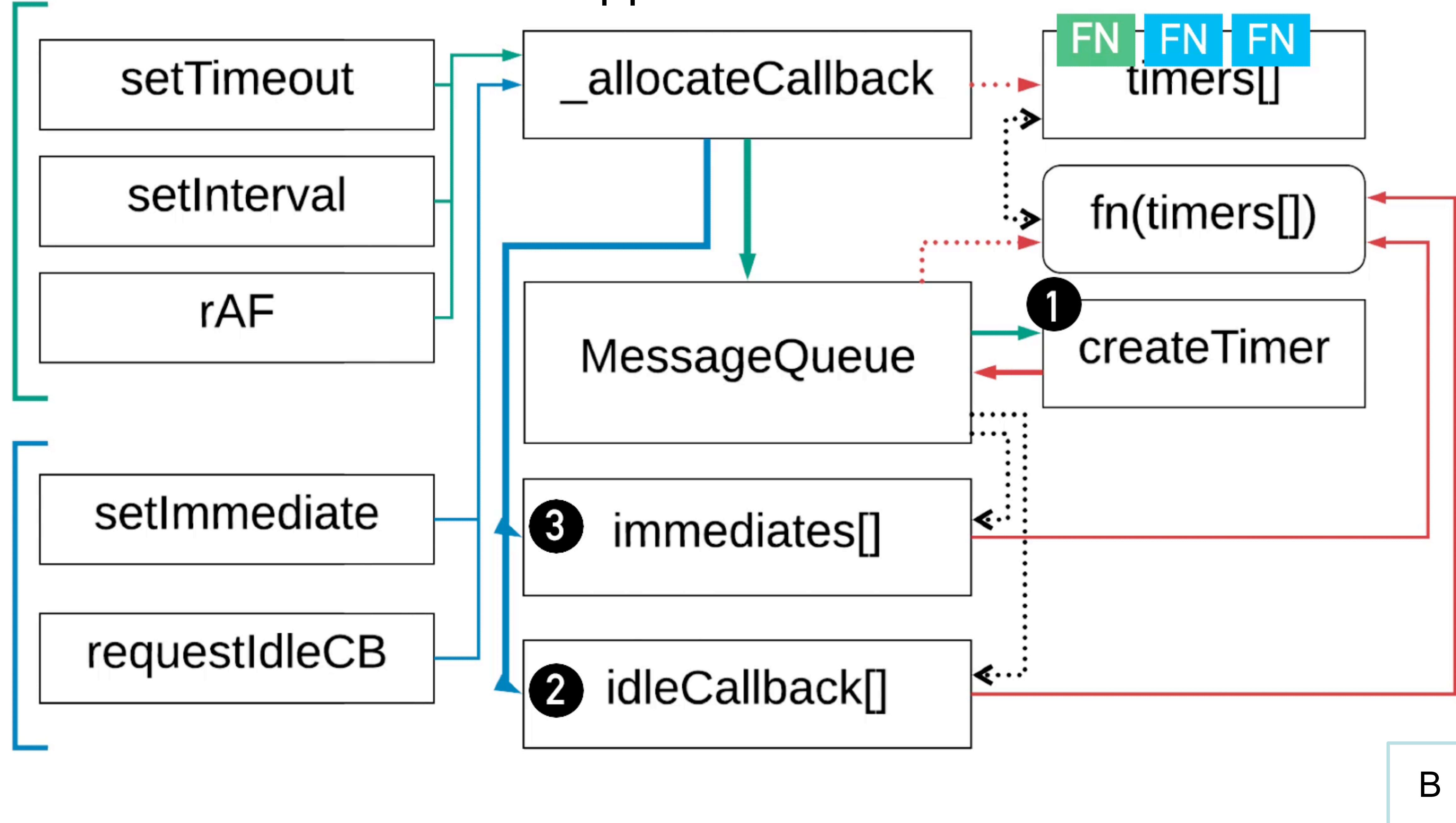




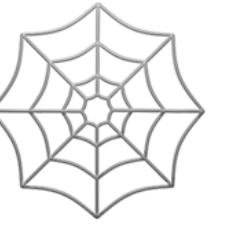




Где ошибка?



Работа с сетью



Fetch API



```
async function getUser(id: string): Promise<any> {
  try {
    const response = await fetch(
      `https://mysuper.api/v1/user/${id}`
    );

    if (response.status === 200) {
      return (await (await response.clone()).json());
    }
  } catch {}

  throw new Error('Unexpected!');
}
```

react-native/Libraries/Network/fetch.js

```
/* globals Headers, Request, Response */
```

```
'use strict';
```

```
import whatwg from 'whatwg-fetch';
```

```
if (whatwg && whatwg.fetch) {  
  module.exports = whatwg;  
} else {  
  module.exports = {fetch, Headers, Request, Response};  
}
```

Мало кто вспомнит, Мало кто поймет



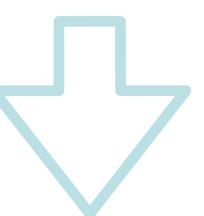
Ajax / Web 2.0

XMLHttpRequest

fetch-wg



XMLHttpRequest



Networking



NSURLRequest

OkHttp

iOS

Android

А как мы асинхронно отсылали файлы без перехода?)

```
<form action="do_stuff.aspx" method="post" target="my_iframe">
    <input type="submit" value="Do Stuff!" />
</form>
```

-- when the form is submitted, the server response will appear in this iframe -->

```
<iframe name="my_iframe" src="not_submitted_yet.aspx"></iframe>
```

Ограничение производительности в RN

#1 JSC Heavy Load

```
work(milliseconds) {
    console.log('work start');

    const start = new Date().getTime();

    for (let i = 0; i < 1e7; i++) {
        if ((new Date()).getTime() - start) > milliseconds) {
            break;
        }
    }

    console.log('work finish');
}
```

Notify

DO WORK IN JS

Simple Notifcation Title
Awesome text

Simple Notifcation Title
Awe

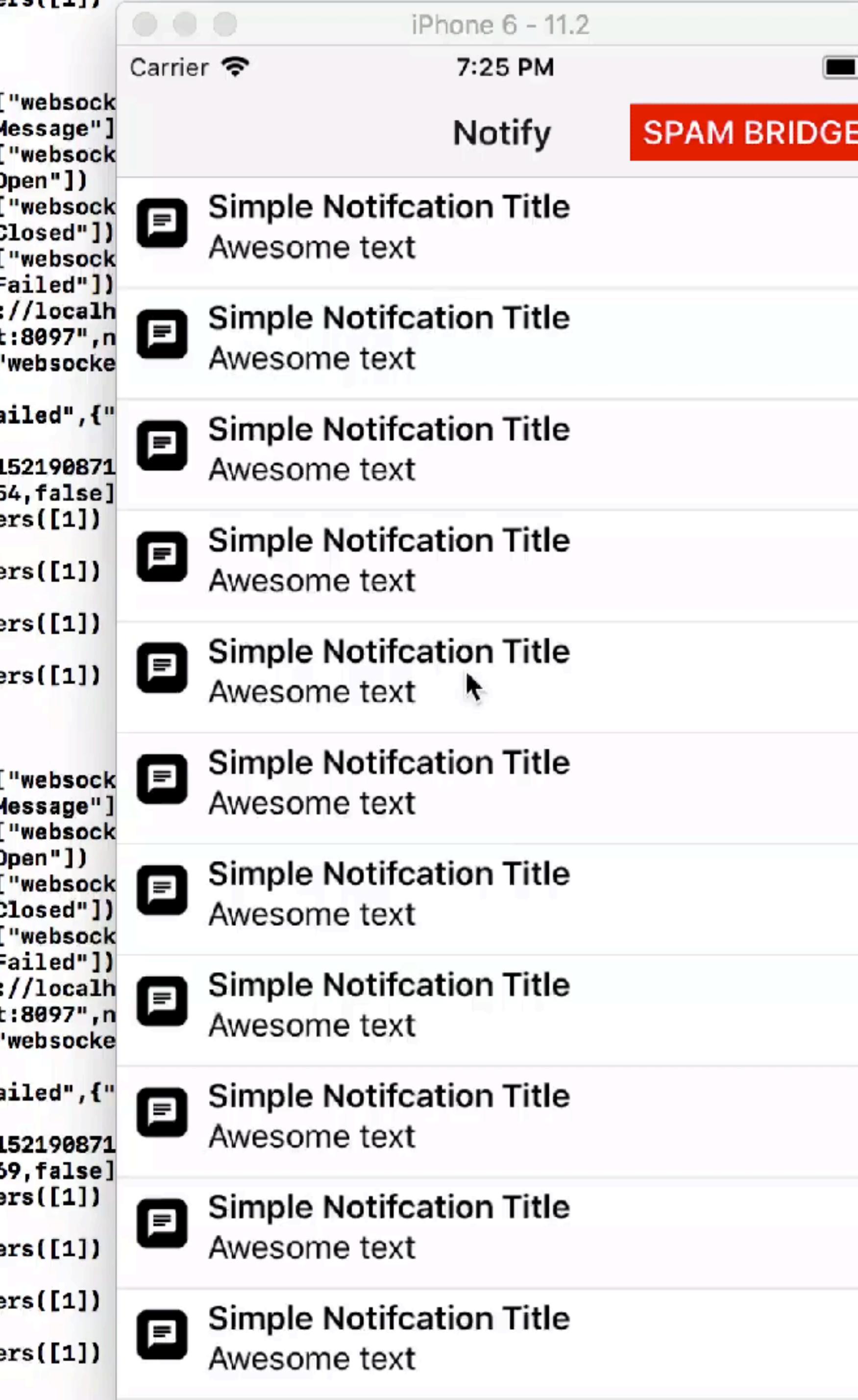
RAM	JSC	Views	UI	JS
874.54	0.00	419	60	60
MB	MB	521		

 Simple Notifcation Title
Awesome text

#2 Hight Message Queue traffic (The Bridge)

```
spamBridge() {  
    for (let i = 0; i < 500; i++) {  
        this.initTimer();  
    }  
}  
  
initTimer() {  
    const that = this;  
    setTimeout(function() {  
        that.initTimer();  
    }, 1);  
}
```

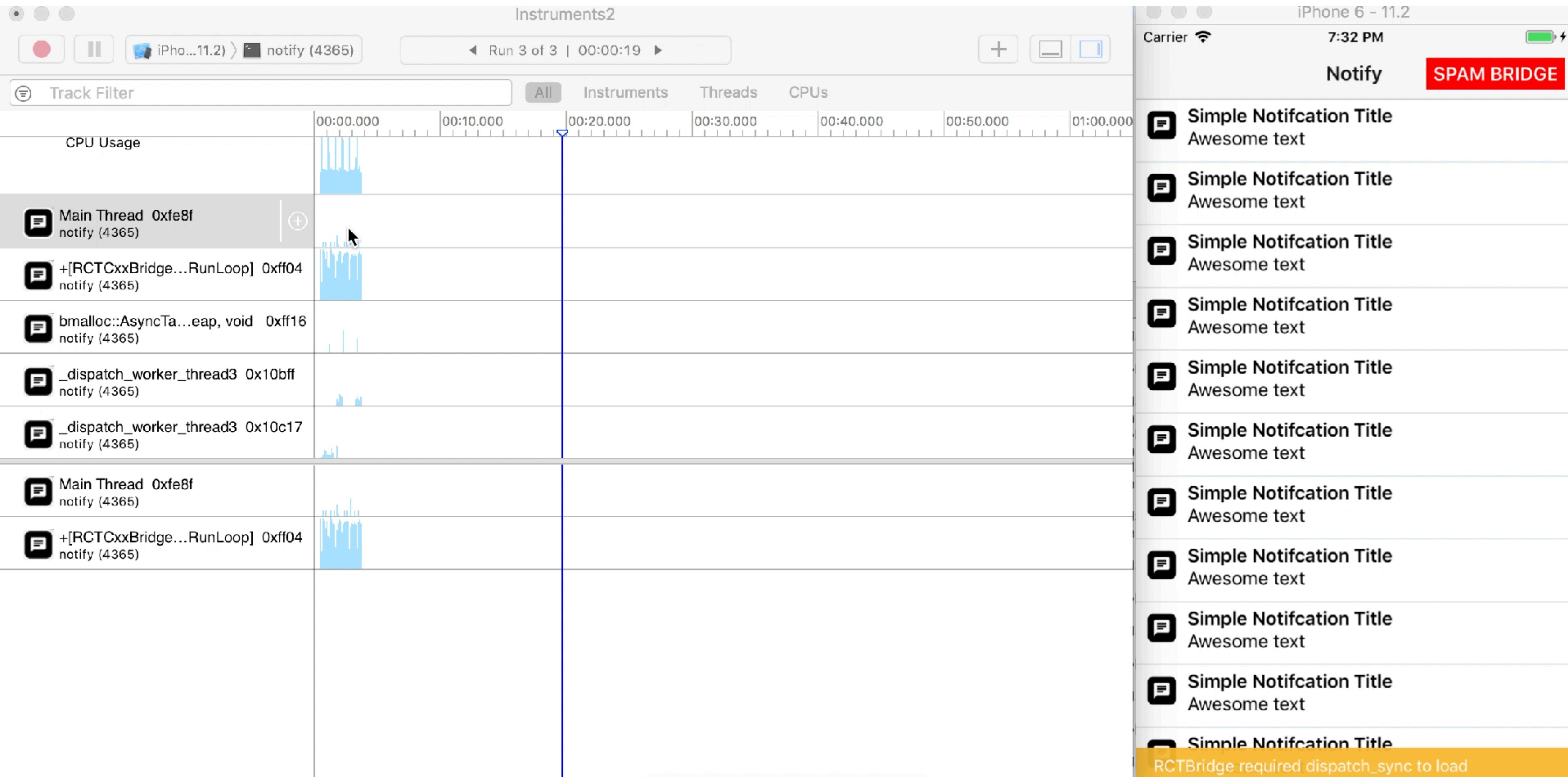
```
2018-03-24 19:25:10.100 [info][tid:com.facebook.react.JavaScript] JS->N : WebSocketModule.removeListeners([1])
2018-03-24 19:25:10.135850+0300 notify[18683:11038647] JS->N : WebSocketModule.removeListeners([1])
2018-03-24 19:25:12.134 [info][tid:com.facebook.react.JavaScript] N->JS : JSTimers.callTimers([[72]])
2018-03-24 19:25:12.134355+0300 notify[18683:11038647] N->JS : JSTimers.callTimers([[72]])
2018-03-24 19:25:12.135 [info][tid:com.facebook.react.JavaScript] JS->N : WebSocketModule.addListener(["websocketOpen"])
2018-03-24 19:25:12.134745+0300 notify[18683:11038647] JS->N : WebSocketModule.addListener(["websocketMessage"])
2018-03-24 19:25:12.135 [info][tid:com.facebook.react.JavaScript] JS->N : WebSocketModule.addListener(["websocketClose"])
2018-03-24 19:25:12.134964+0300 notify[18683:11038647] JS->N : WebSocketModule.addListener(["websocketFailed"])
2018-03-24 19:25:12.135 [info][tid:com.facebook.react.JavaScript] JS->N : WebSocketModule.connect(["ws://localhost:8097"])
2018-03-24 19:25:12.141518+0300 notify[18683:11038647] JS->N : WebSocketModule.connect(["ws://localhost:8097", "n"])
2018-03-24 19:25:12.154 [info][tid:com.facebook.react.JavaScript] N->JS : RCTDeviceEventEmitter.emit(["websocketConnection refused","id":25])
2018-03-24 19:25:12.153686+0300 notify[18683:11038647] N->JS : RCTDeviceEventEmitter.emit(["websocketFailed","refused","id":25])
2018-03-24 19:25:12.154 [info][tid:com.facebook.react.JavaScript] JS->N : Timing.createTimer([73,2000,152190871])
2018-03-24 19:25:12.154397+0300 notify[18683:11038647] JS->N : Timing.createTimer([73,2000,1521908712154],false)
2018-03-24 19:25:12.155 [info][tid:com.facebook.react.JavaScript] JS->N : WebSocketModule.removeListeners([1])
2018-03-24 19:25:12.154761+0300 notify[18683:11038647] JS->N : WebSocketModule.removeListener([1])
2018-03-24 19:25:12.155 [info][tid:com.facebook.react.JavaScript] JS->N : WebSocketModule.removeListener([1])
2018-03-24 19:25:12.154992+0300 notify[18683:11038647] JS->N : WebSocketModule.removeListener([1])
2018-03-24 19:25:12.155 [info][tid:com.facebook.react.JavaScript] JS->N : WebSocketModule.removeListener([1])
2018-03-24 19:25:12.155302+0300 notify[18683:11038647] JS->N : WebSocketModule.removeListener([1])
2018-03-24 19:25:12.155 [info][tid:com.facebook.react.JavaScript] JS->N : WebSocketModule.removeListener([1])
2018-03-24 19:25:12.155625+0300 notify[18683:11038647] JS->N : WebSocketModule.removeListener([1])
2018-03-24 19:25:14.155 [info][tid:com.facebook.react.JavaScript] N->JS : JSTimers.callTimers([[73]])
2018-03-24 19:25:14.154859+0300 notify[18683:11038647] N->JS : JSTimers.callTimers([[73]])
2018-03-24 19:25:14.156 [info][tid:com.facebook.react.JavaScript] JS->N : WebSocketModule.addListener(["websocketOpen"])
2018-03-24 19:25:14.155638+0300 notify[18683:11038647] JS->N : WebSocketModule.addListener(["websocketMessage"])
2018-03-24 19:25:14.156 [info][tid:com.facebook.react.JavaScript] JS->N : WebSocketModule.addListener(["websocketClose"])
2018-03-24 19:25:14.156018+0300 notify[18683:11038647] JS->N : WebSocketModule.addListener(["websocketFailed"])
2018-03-24 19:25:14.156 [info][tid:com.facebook.react.JavaScript] JS->N : WebSocketModule.addLister(["websocketOpen"])
2018-03-24 19:25:14.156346+0300 notify[18683:11038647] JS->N : WebSocketModule.addLister(["websocketClosed"])
2018-03-24 19:25:14.156622+0300 notify[18683:11038647] JS->N : WebSocketModule.addLister(["websocketFailed"])
2018-03-24 19:25:14.156998+0300 notify[18683:11038647] JS->N : WebSocketModule.connect(["ws://localhost:8097"])
2018-03-24 19:25:14.156998+0300 notify[18683:11038647] JS->N : WebSocketModule.connect(["ws://localhost:8097", "n"])
2018-03-24 19:25:14.169 [info][tid:com.facebook.react.JavaScript] N->JS : RCTDeviceEventEmitter.emit(["websocketConnection refused","id":26])
2018-03-24 19:25:14.169324+0300 notify[18683:11038647] N->JS : RCTDeviceEventEmitter.emit(["websocketFailed","refused","id":26])
2018-03-24 19:25:14.170 [info][tid:com.facebook.react.JavaScript] JS->N : Timing.createTimer([74,2000,152190871])
2018-03-24 19:25:14.170118+0300 notify[18683:11038647] JS->N : Timing.createTimer([74,2000,1521908714169],false)
2018-03-24 19:25:14.171 [info][tid:com.facebook.react.JavaScript] JS->N : WebSocketModule.removeListeners([1])
2018-03-24 19:25:14.170646+0300 notify[18683:11038647] JS->N : WebSocketModule.removeListener([1])
2018-03-24 19:25:14.171 [info][tid:com.facebook.react.JavaScript] JS->N : WebSocketModule.removeListener([1])
2018-03-24 19:25:14.170950+0300 notify[18683:11038647] JS->N : WebSocketModule.removeListener([1])
2018-03-24 19:25:14.171 [info][tid:com.facebook.react.JavaScript] JS->N : WebSocketModule.removeListener([1])
2018-03-24 19:25:14.171196+0300 notify[18683:11038647] JS->N : WebSocketModule.removeListener([1])
2018-03-24 19:25:14.171 [info][tid:com.facebook.react.JavaScript] JS->N : WebSocketModule.removeListener([1])
2018-03-24 19:25:14.171404+0300 notify[18683:11038647] JS->N : WebSocketModule.removeListener([1])
```



Лагает, но не лагает

- JS Thread - **heavy load**
- “The bridge” thread - **heavy load**
- UI Thread (main) - **ok**





+[RCTCxxBridge...oop] 0x1406e | 

notify (4792)

+[RCTCxxBridge runRunLoop] 0x1406e > Profile > Root

Weight	Self Weight	Symbol Name
2.39 min	100.0%	0 s ▾ notify (4792) 
2.39 min	99.9%	0 s  ▶ thread_start libsystem_pthr...
4.00 ms	0.0%	0 s  ▶ <Call stack limit reached> 



-  140437.0 std::__1::__function::__func<facebook::react::NativeToJsBridge::runOnExecutorQueue(s...
-  140436.0 void std::__1::__invoke_void_return_wrapper<void>::__call<facebook::react::NativeToJsB...
-  140436.0 decltype(std::__1::forward<facebook::react::NativeToJsBridge::runOnExecutorQueue(st...
-  140436.0 facebook::react::NativeToJsBridge::runOnExecutorQueue(std::__1::function<void (face...
-  140436.0 std::__1::function<void (facebook::react::JSExecutor*)>::operator()(facebook::react::JS...
-  140423.0 std::__1::__function::__func<facebook::react::NativeToJsBridge::callFunction(std::__1::b...
-  140423.0 void std::__1::__invoke_void_return_wrapper<void>::__call<facebook::react::NativeToJsB...
-  140423.0 decltype(std::__1::forward<facebook::react::NativeToJsBridge::callFunction(std::__1::ba...
-  140423.0 facebook::react::NativeToJsBridge::callFunction(std::__1::basic_string<char, std::__1::ch...
-  140423.0 facebook::react::JSCExecutor::callFunction(std::__1::basic_string<char, std::__1::char_tr...
-  140280.0 facebook::react::JSCExecutor::callFunction(std::__1::basic_string<char, std::__1::char_tr...
-  140134.0 facebook::react::Object::callAsFunction(std::initializer_list<OpaqueJSValue const*>) co...
-  140134.0 facebook::react::Object::callAsFunction(OpaqueJSValue*, int, OpaqueJSValue const* c...
-  140134.0 JSObjectCallAsFunction
-  140134.0 JSC::profiledCall(JSC::ExecState*, JSC::ProfilingReason, JSC::JSValue, JSC::CallType, J...
-  140134.0 JSC::Interpreter::executeCall(JSC::ExecState*, JSC::JSObject*, JSC::CallType, JSC::Call...
-  140125.0 vmEntryToNative
-  140125.0 JSC::boundThisNoArgsFunctionCall(JSC::ExecState*)
-  140124.0 JSC::call(JSC::ExecState*, JSC::JSValue, JSC::CallType, JSC::CallData const&, JSC::JS...

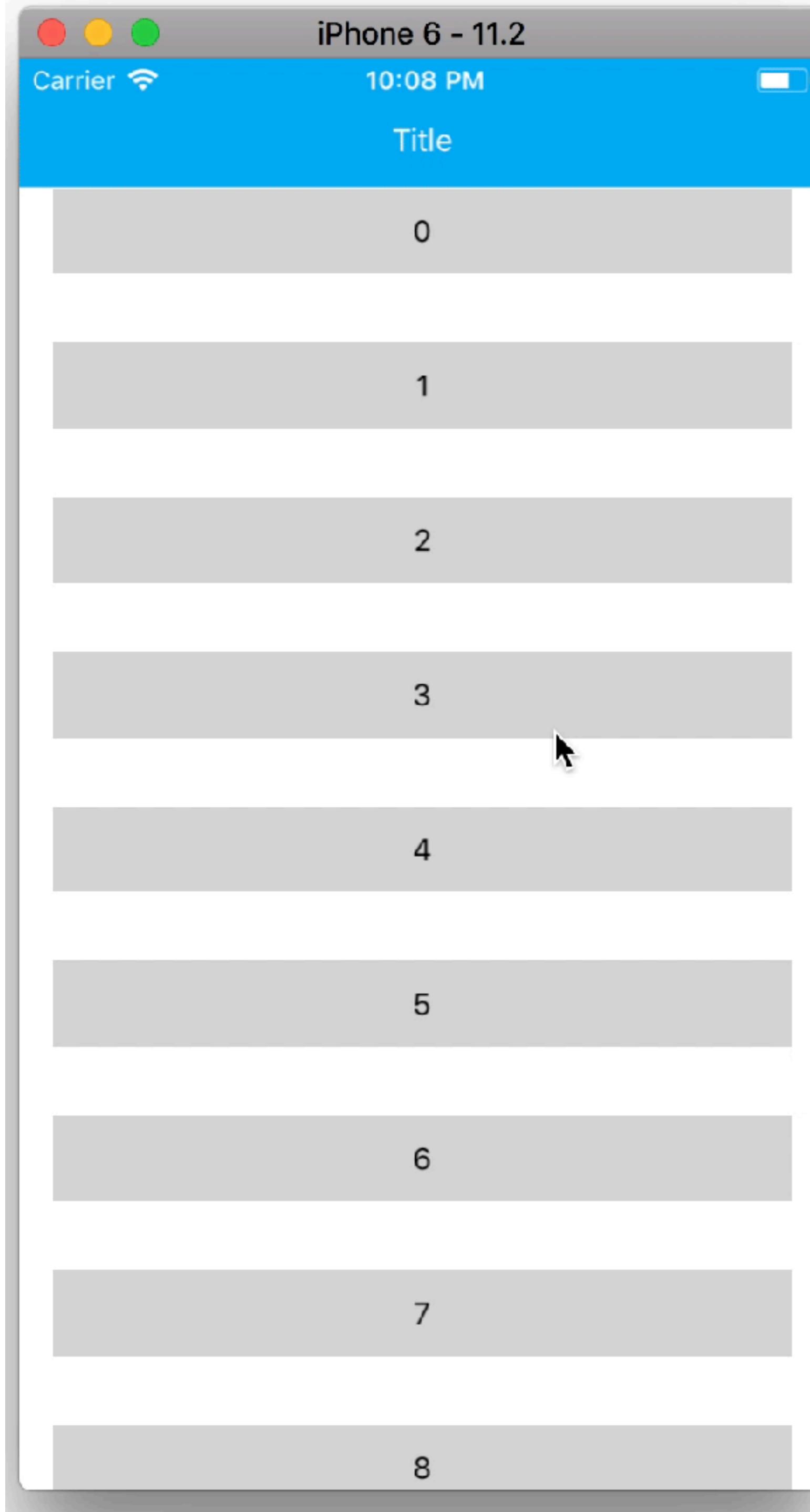
Input Filter

 Involves Symbol

Call Tree

Call Tree

Animations



```
<ParallaxView>  
  <View>  
    ...  
  </View>  
</ParallaxView>
```

Идея #1 - State - legacy RN < 0.43

```
export default class Profile extends Component {
  onScroll(event) {
    this.setState({
      headerHeight: countHeaderHeight(event.y)
    });
  }
  // Render + transform from this.state
  render() {}
}
```

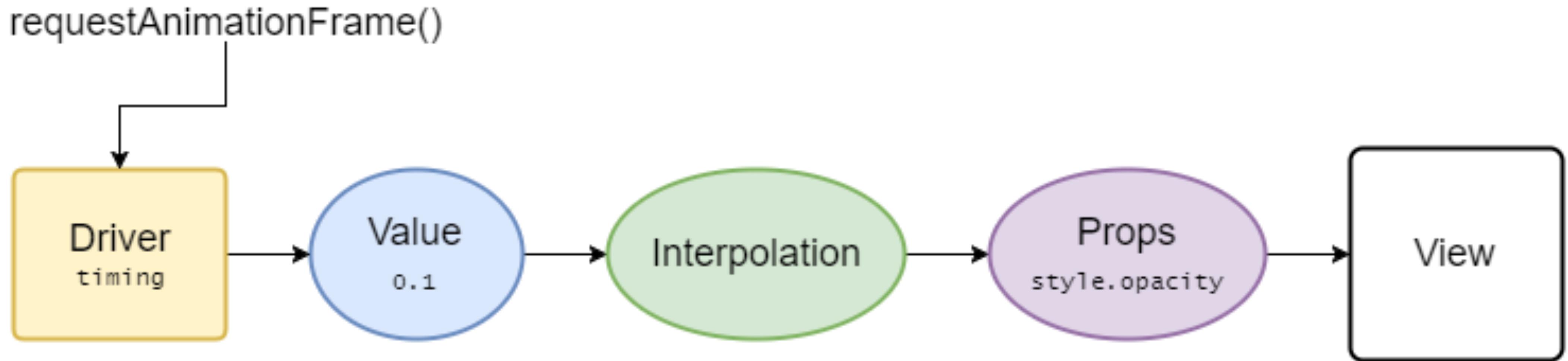
Идея #2- State + Direct Manipulation

```
export default class Profile extends Component {  
  onScroll(event) {  
    this.refs['container'].setNativeProps(...);  
  }  
  
  render() {}  
}
```

Идея #3 - Native Component

Идея #4 Animated.*

```
state = {  
  scrollY: new Animated.Value(0),  
};  
  
<View>  
  <AnimatedScrollView  
    style={styles.fill}  
    scrollEventThrottle={16}  
    onScroll={Animated.event(  
      [{ nativeEvent: { contentOffset: { y: this.state.scrollY } } } ],  
    )}  
  >  
  {this._renderScrollViewContent()}  
  </AnimatedScrollView>  
</View>
```

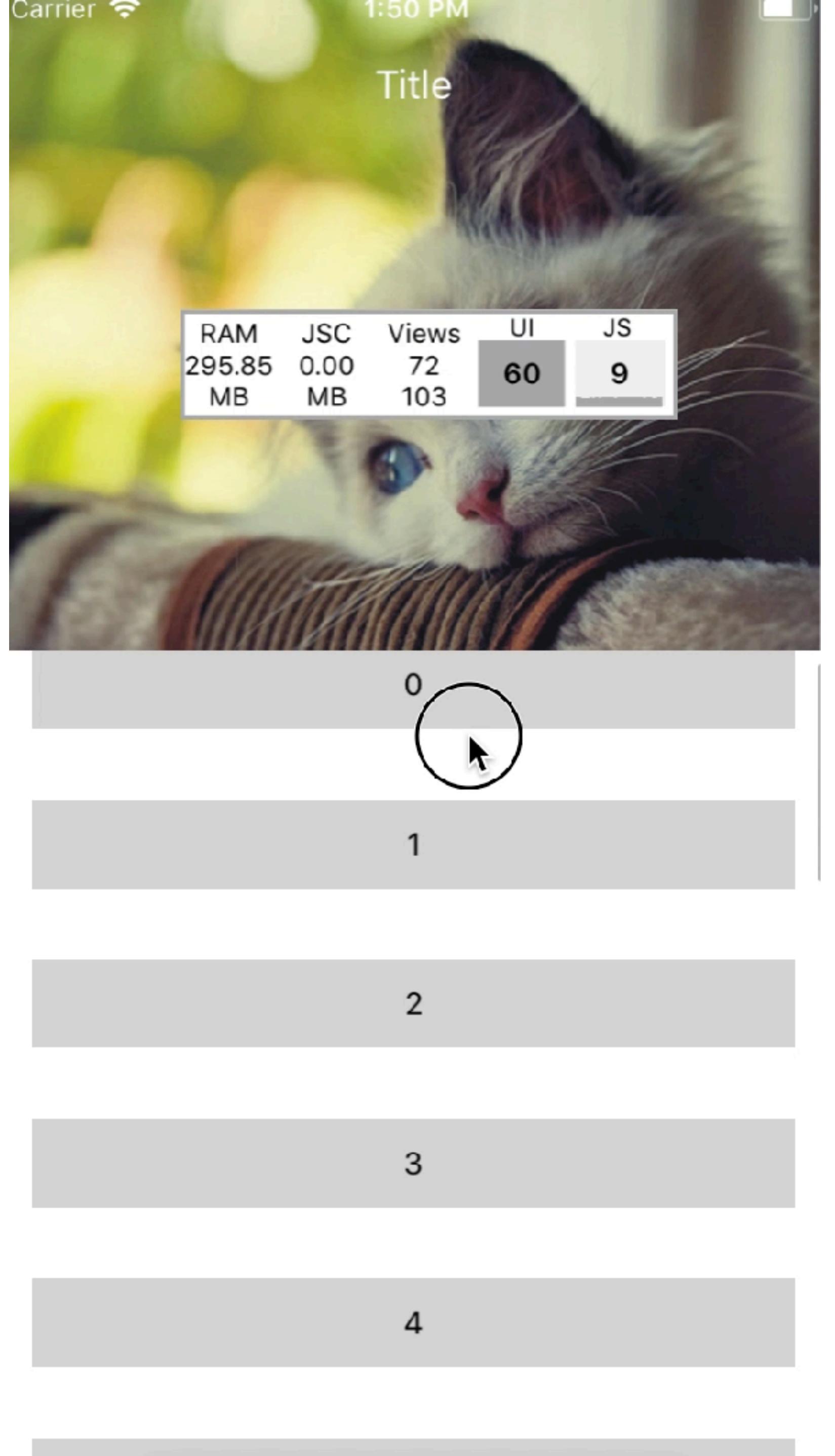


Driver
timing

Props

UIView
android.View

- » JS: requestAnimationFrame
- » JS: Calculate props -> <View />
- » JS: View. setNativeProps
- » JS to Native bridge
- » UIView / android.View updated



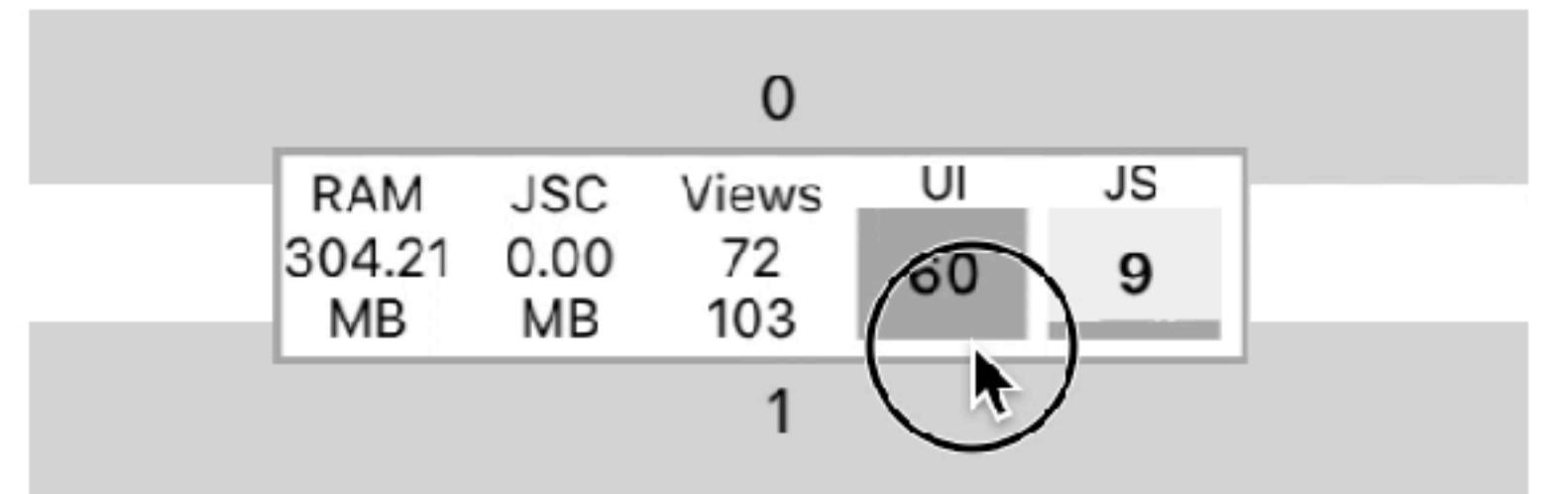
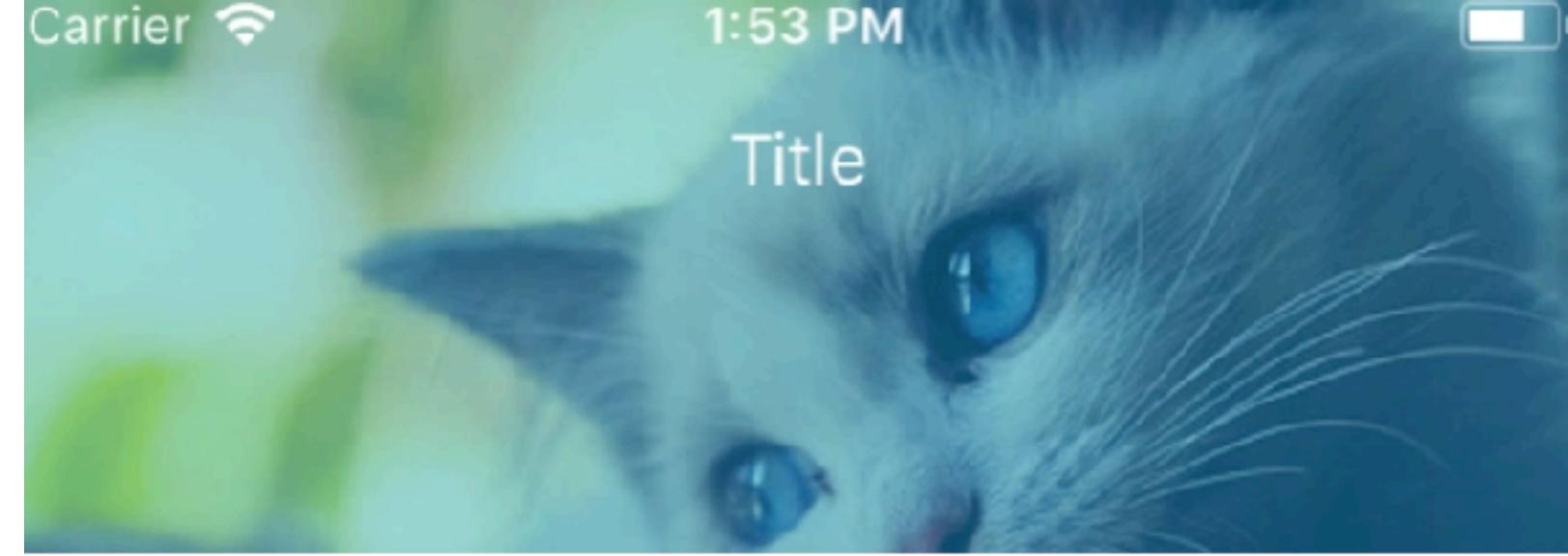
Animated.*
Hight Message Queue traffic

Идея #5 Animated.* + useNativeDriver

```
<View>
  <AnimatedScrollView
    style={styles.fill}
    scrollEventThrottle={1}
    onScroll={Animated.event(
      [{ nativeEvent: { contentOffset: { y: this.state.scrollY } } }],
      { useNativeDriver: true }
    )}
  >
    {this._renderScrollViewContent()}
  </AnimatedScrollView>
</View>
```

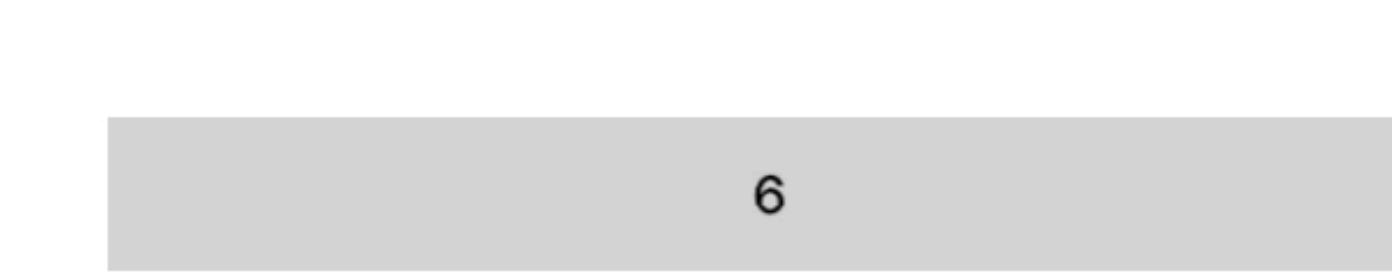
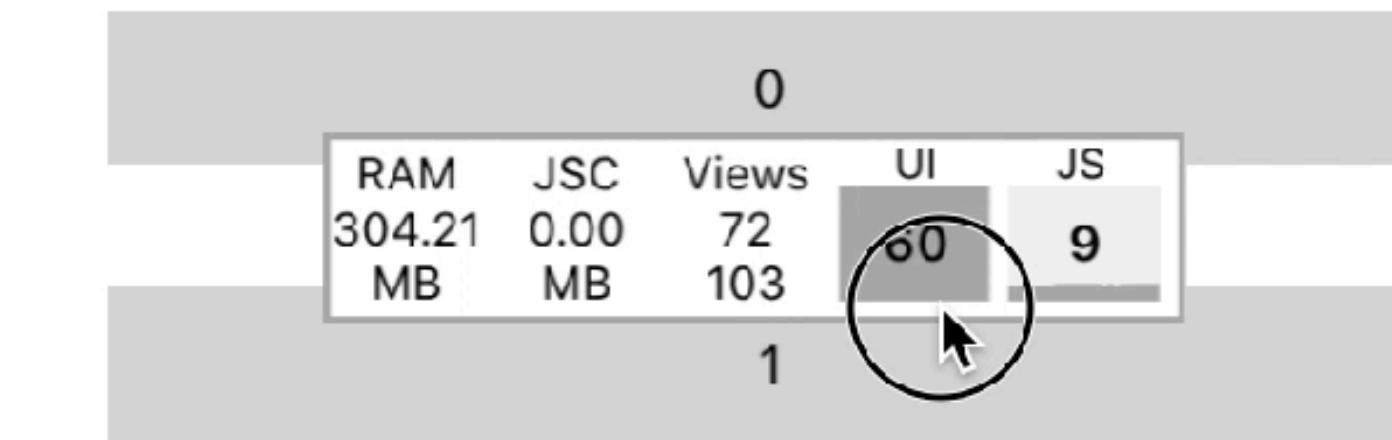
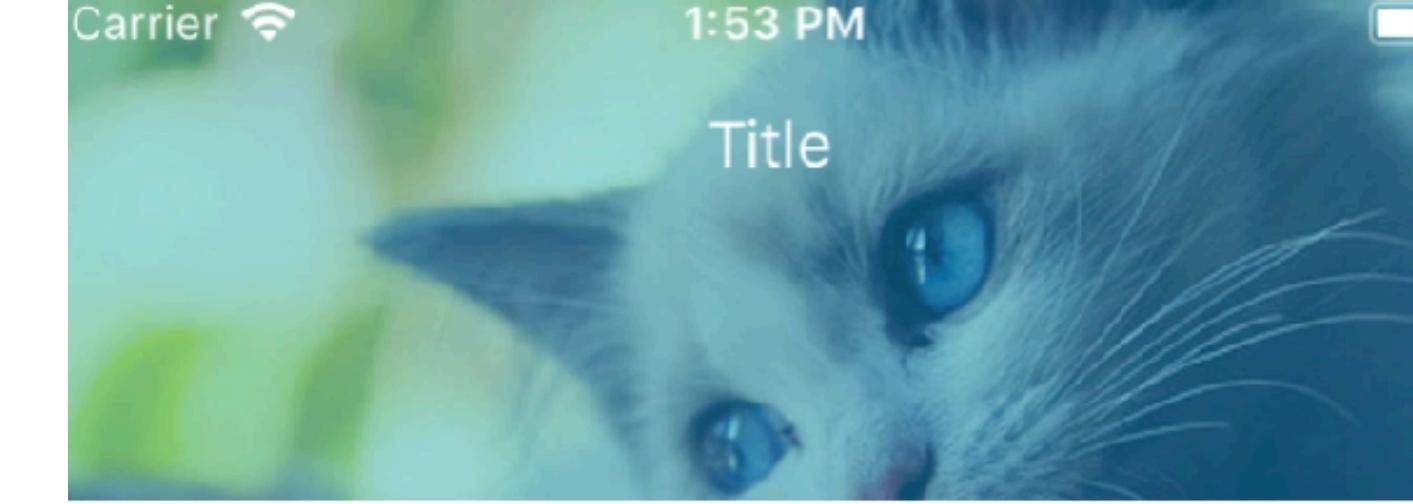
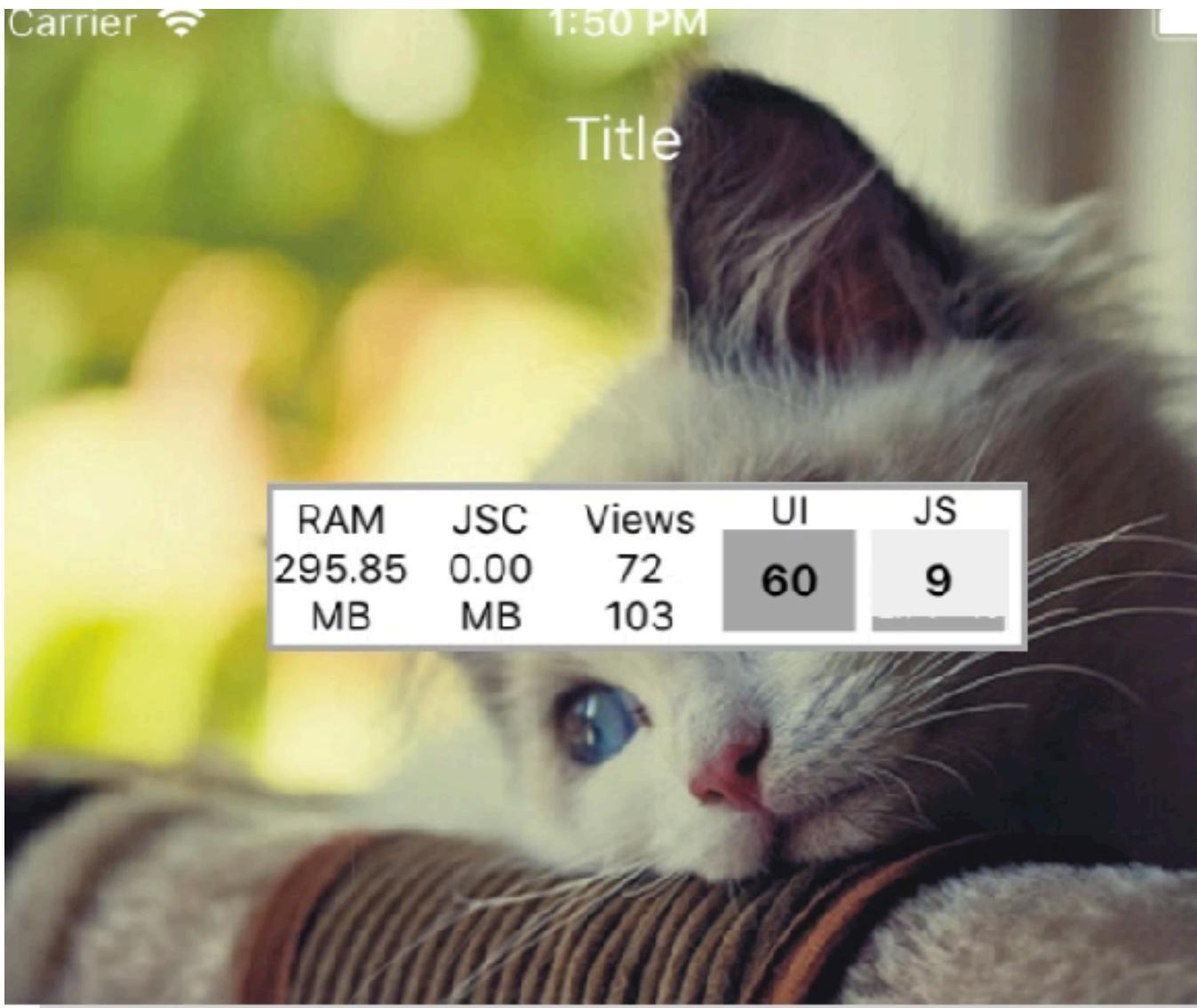
Идея #5 Animated.* + useNativeDriver

- » JS: requestAnimationFrame
- » JS: Calculate props -> <View />
- » JS: View. setNativeProps
- » JS to Native bridge
- » Native: CADisplayLink / android.View.Choreographer
- » Native: Calculate and update Props
- » UIView / android.View updated



**Animated.* + useNativeDriver
Hight Message Queue traffic**

Animated.* vs Animated.* + useNativeDriver



Реальные примеры 

```
class ProfileRepositories extends PureComponent<Props> {
  componentDidMount() {
    this.props.fetchRepositories(this.props.navigation.getParam('id'));
  }

  render() {
    const { repositories } = this.props.state;

    return (
      <FlatList
        style={styles.list}
        data={repositories}
        keyExtractor={(repository: RepositoryEntity) => repository.id}
        renderItem={({ item }) => (
          <RepositoryRow
            repo={item}
            onPress={() => showRepository(item)}
          />
        )}
        onEndReachedThreshold={0.5}
        ItemSeparatorComponent={RowSeparator}
      />
    );
  }
}
```

```
export function fetchRepositories(username: string): ThunkAction {
  return dispatch => {
    dispatch({
      type: PROFILE_REPOSITORIES_REQUEST
    });

    try {
      const payload = getRepositoriesByUsername(
        username,
        {
          sort: 'pushed',
          'per_page': 50
        }
      );
      dispatch({
        type: PROFILE_REPOSITORIES_REQUEST_SUCCESS,
        payload
      });
    } catch (e) {
      dispatch({
        type: PROFILE_REPOSITORIES_REQUEST_FAIL
      });
    }
  };
}
```

Выводы

- Ничего страшного внутри нету
- JavaScript разработчики заменят всех
- Инкапсуляция кода над платформами
- Нативные модули пишутся только когда вы четко знаете что это единственный вариант



3 Вопроса - 3 Ответа



Что будет если Facebook прекратит разработку React Native?

Жить дальше...

У нас есть команда iOS так и Android разработчиков, мы хотим писать новый функционал на RN, что делать?

Жить дальше...

Facebook диктует `@flow` как статический
анализатор во всех примерах, но как же
`typescript`?

`@types/react-native`

Вопросы?)



<https://github.com/ovr>

<https://github.com/ovr/ghubber>

talk@dmtry.me

<https://telegram.me/ovrweb>