

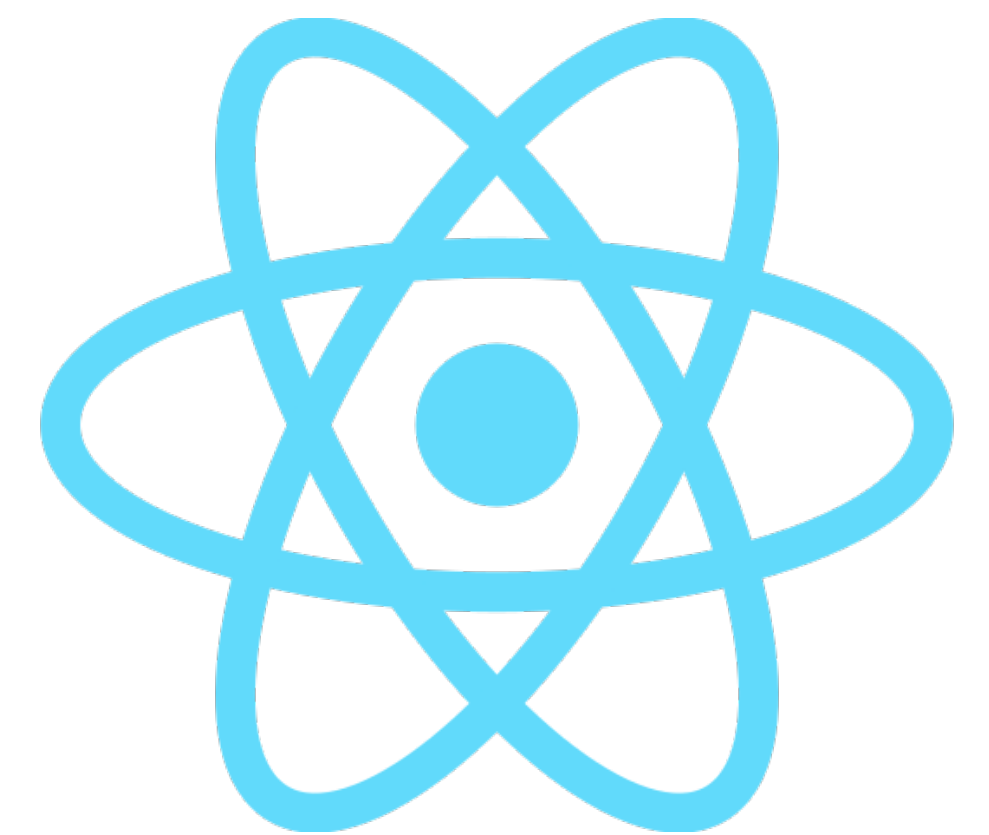


Ислам Рустамов

Team Lead

islam.rustamov@fojin.tech

Внутреннее устройство React Native



Как работает React Native

Проблемы и примеры их решения

Сравнения производительности

Как работает React Native

Компоненты


```
11 import {Text, View, Image, Button} from 'react-native';
```

```
49 return (  
50   <View style={styles.container}>  
51     <View style={styles.block}>  
52       <Text>Compressed image:</Text>  
53       {!!compressedImage && (  
54         <Image source={{ uri: compressedImage }} style={styles.image} />  
55       )}  
56     </View>  
57   </View>  
58 );
```

```
11 import {Text, View, Image, Button} from 'react-native';
```

```
49 return (  
50   <View style={styles.container}>  
51     <View style={styles.block}>  
52       <Text>Compressed image:</Text>  
53       {!!compressedImage && (  
54         <Image source={{ uri: compressedImage }} style={styles.image} />  
55       )}  
56     </View>  
57   </View>  
58 );
```



```
11 import {Text, View, Image, Button} from 'react-native';
```

```
49 return (  
50   <View style={styles.container}>  
51     <View style={styles.block}>  
52       <Text>Compressed image:</Text>  
53       {!!compressedImage && (  
54         <Image source={{ uri: compressedImage }} style={styles.image} />  
55       )}  
56     </View>  
57   </View>  
58 );
```

```
60 // ...
61 ReactNativePrivateInterface.UIManager.createView(
62     current,
63     type.uiViewClassName,
64     renderLanes,
65     updatePayload
66 );
67 // ...
```



```
60 // ...
61 ReactNativePrivateInterface.UIManager.createView(
62     current,
63     type.uiViewClassName,
64     renderLanes,
65     updatePayload
66 );
67 // ...
```



```
972 RCT_EXPORT_METHOD(createView
973     : (nonnull NSNumber *)reactTag viewName
974     : (NSString *)viewName rootTag
975     : (nonnull NSNumber *)rootTag props
976     : (NSDictionary *)props)
977 {
978     // ...
```

```
241 /** Invoked by React to create a new node with a given tag, class name and properties. */
242 public void createView(int tag, String className, int rootViewTag, ReadableMap props) {
243     // ...
```



```
972 RCT_EXPORT_METHOD(createView
973     : (nonnull NSNumber *)reactTag viewName
974     : (NSString *)viewName rootTag
975     : (nonnull NSNumber *)rootTag props
976     : (NSDictionary *)props)
977 {
978     // ...
```

```
241 /** Invoked by React to create a new node with a given tag, class name and properties. */
242 public void createView(int tag, String className, int rootViewTag, ReadableMap props) {
243     // ...
```

```
972 RCT_EXPORT_METHOD(createView
973     : (nonnull NSNumber *)reactTag viewName
974     : (NSString *)viewName rootViewTag
975     : (nonnull NSNumber *)rootTag props
976     : (NSDictionary *)props)
977 {
978     // ...
```

```
241 /** Invoked by React to create a new node with a given tag, class name and properties. */
242 public void createView(int tag, String className, int rootViewTag, ReadableMap props) {
243     // ...
```

Как работает React Native

ДВИЖКИ

Как работает React Native

Hermes

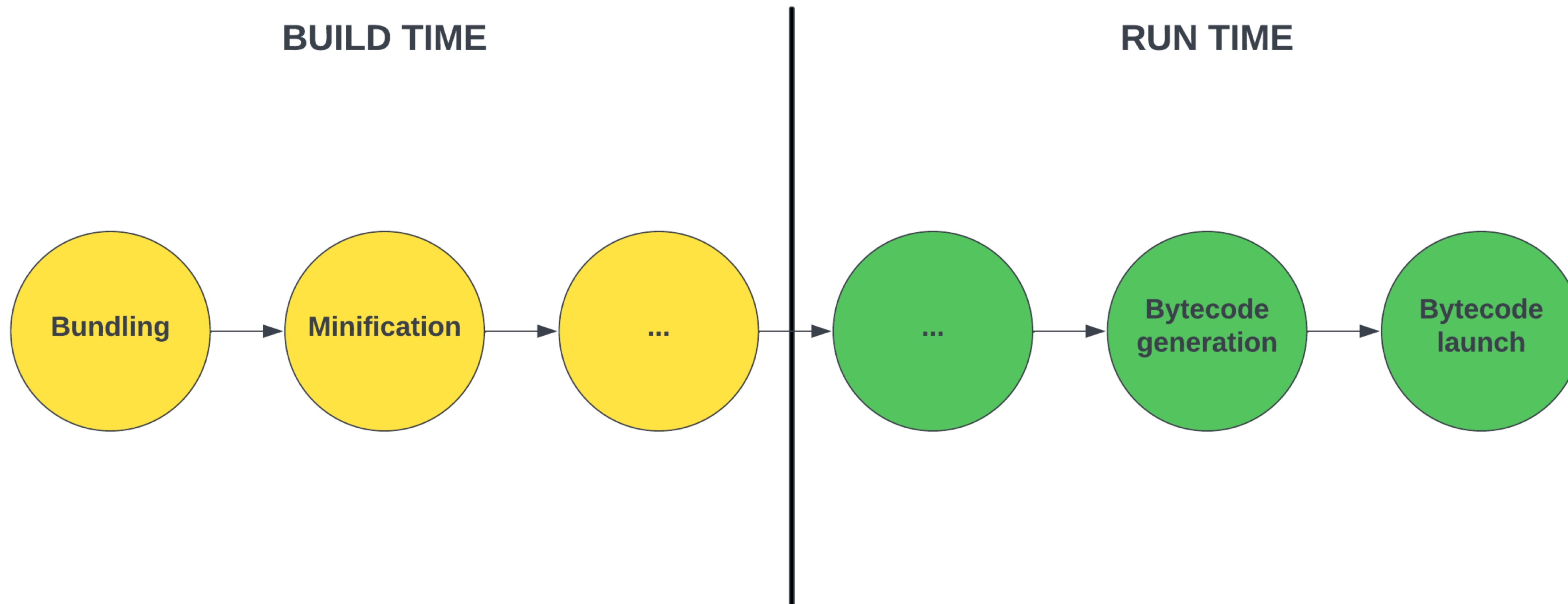


Как работает React Native

JSC



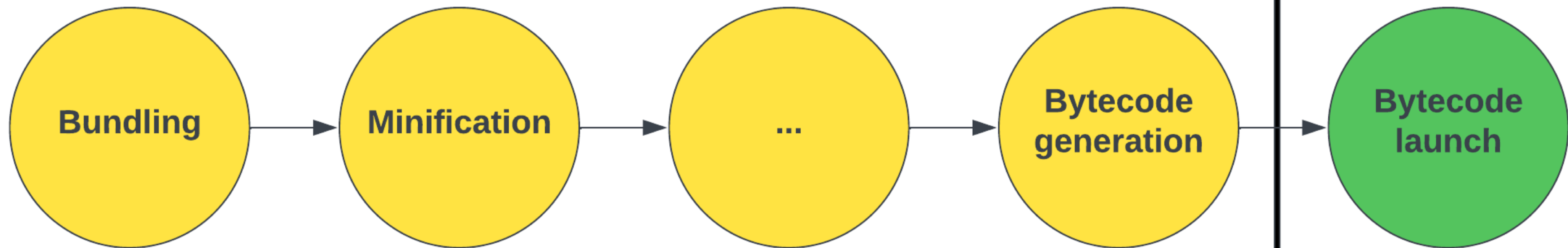
JIT



AOT

BUILD TIME

RUN TIME



Как работает React Native

В чем еще крутость Hermes

Как работает React Native

mmap

Как работает React Native



app-release-JSC.apk 71,3 MB





Modified: Today, 15:33



app-release-hermes.... 55,8 MB

Modified: Today, 15:31

Как работает React Native

	app-release-JSC.apk 71,3 MB Modified: Today, 15:33	
	app-release-hermes... 55,8 MB Modified: Today, 15:31	

Как работает React Native

Бандлеры

Как работает React Native

Metro





tmikov commented on Feb 10, 2021

Contributor

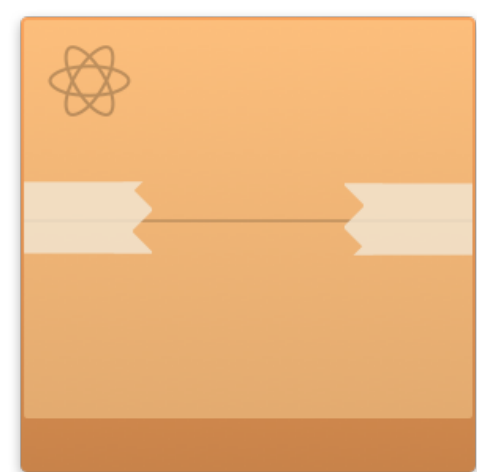


No, Hermes doesn't need minified input. Minifiers primarily rename local variables and parameters and perform some simple AST transformations, which have very little, if any, impact on the size and performance of a compiled Hermes bundle. We recommend disabling minification with Hermes, but haven't really pushed for it aggressively since the Metro pipeline also supports JSC which does benefit from minification.



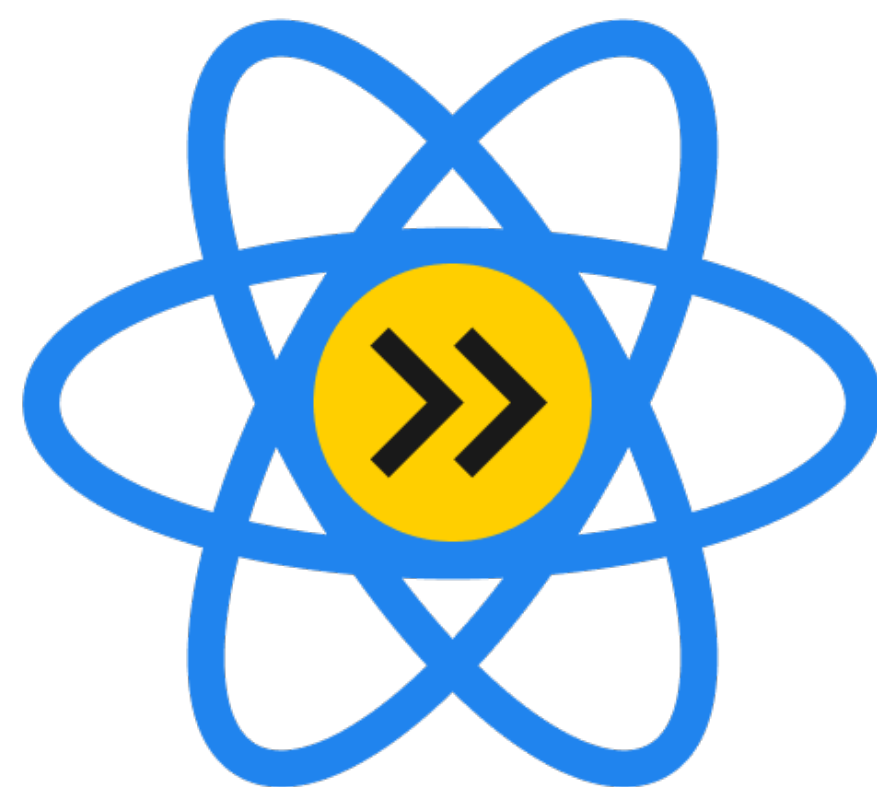
<https://github.com/facebook/hermes/issues/452>

Аналоги Metro



HAUL

haul (deprecated)



esbuild

re.pack

re.pack

Как работает React Native

**Где работает быстрее
new Date()?**

Как работает React Native

В 800 раз быстрее на JSC

<https://github.com/facebook/hermes/issues/930>

Как работает React Native

Нативные модули

```
22 @ReactModule(name = PhotoCompressorModule.NAME)
23 public class PhotoCompressorModule extends NativePhotoCompressorSpec {
24     public static final String NAME = "PhotoCompressor";
25
26     @Override
27     public void compressPhoto(String uri, double quality, Promise promise) {
28         // ...
29     }
30
31     @Override
32     public void getSizeInBytes(String uri, Promise promise) {
33         // ...
34     }
```

<https://www.npmjs.com/package/react-native-photo-compressor>

```
22 @ReactModule(name = PhotoCompressorModule.NAME)
23 public class PhotoCompressorModule extends NativePhotoCompressorSpec {
24     public static final String NAME = "PhotoCompressor";
25
26     @Override
27     public void compressPhoto(String uri, double quality, Promise promise) {
28         // ...
29     }
30
31     @Override
32     public void getSizeInBytes(String uri, Promise promise) {
33         // ...
34     }
```

<https://www.npmjs.com/package/react-native-photo-compressor>

```
14 export default function App() {
15     const [photo, setPhoto] = useState<string>();
16
17     async function openCamera() {
18         // ...
19         const compressedPhoto = await compressPhoto({ uri: 'some/photo.png', quality: 1 });
20         // ...
21         setPhoto(compressedPhoto);
22     }
```

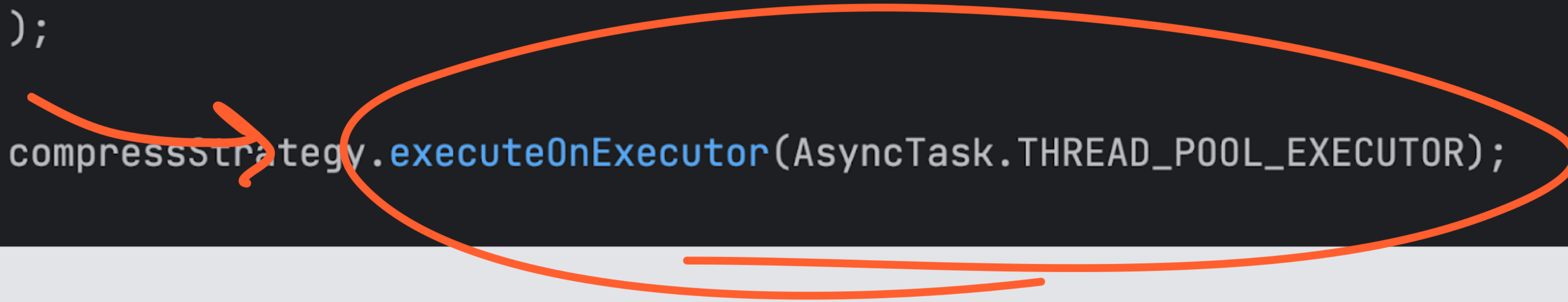


```
14 export default function App() {
15     const [photo, setPhoto] = useState<string>();
16
17     async function openCamera() {
18         // ...
19         const compressedPhoto = await compressPhoto(uri: 'some/photo.png', quality: 1);
20         // ...
21         setPhoto(compressedPhoto);
22     }
```



```
38  @Override
39  public void compressPhoto(String uri, double quality, Promise promise) {
40      CompressStrategy compressStrategy = new CompressStrategy(
41          mContext,
42          uri,
43          quality,
44          promise
45      );
46
47      compressStrategy.executeOnExecutor(AsyncTask.THREAD_POOL_EXECUTOR);
48  }
```

```
38  @Override
39  public void compressPhoto(String uri, double quality, Promise promise) {
40      CompressStrategy compressStrategy = new CompressStrategy(
41          mContext,
42          uri,
43          quality,
44          promise
45      );
46
47      compressStrategy.executeOnExecutor(AsyncTask.THREAD_POOL_EXECUTOR);
48  }
```



```
91     private static class CompressStrategy extends GuardedAsyncTask<Void, Void> {
92         ...
93         @Override
94         protected void doInBackgroundGuarded(Void... params) {
95             ...
96             bitmap.compress(Bitmap.CompressFormat.JPEG, (int) mQuality, out);
97             ...
98         }
99     }
```

```
91 private static class CompressStrategy extends GuardedAsyncTask<Void, Void> {
92     ...
93     @Override
94     protected void doInBackgroundGuarded(Void... params) {
95         ...
96         bitmap.compress(Bitmap.CompressFormat.JPEG, (int) mQuality, out);
97         ...
98     }
99 }
```

Есть 2 типа модулей

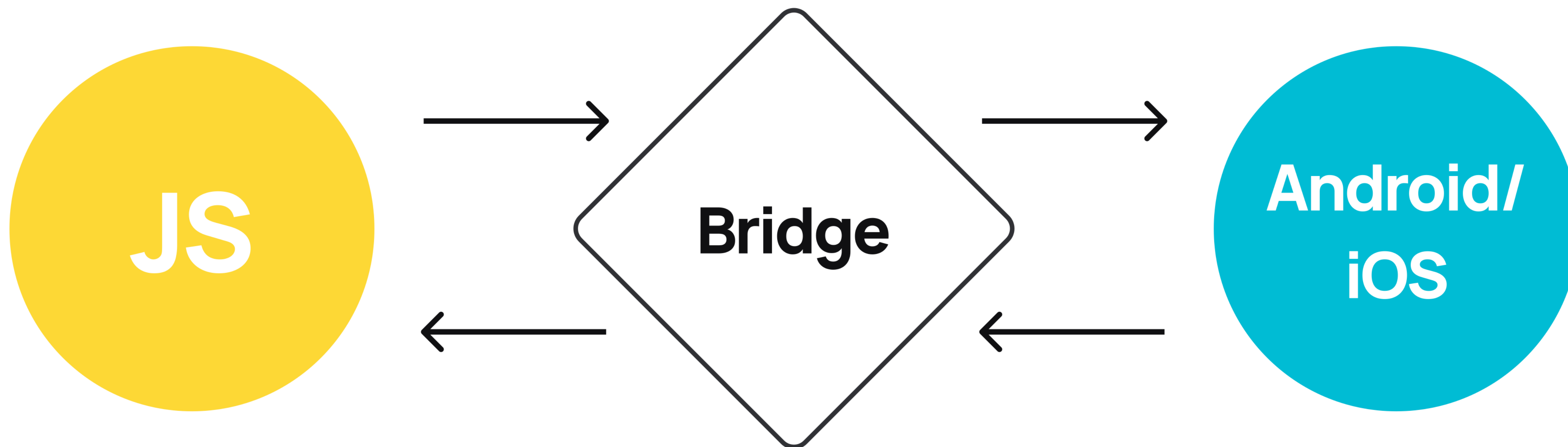
Турбо модули

Нативные модули

Как работает React Native

Архитектура

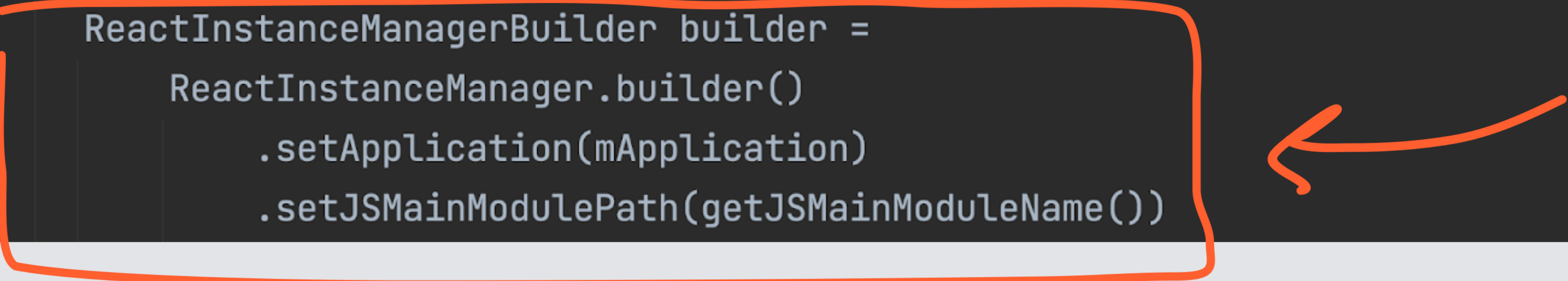
Старая архитектура



```
118  /**
119     * ...
120     * An instance of this manager is required to start JS
121     * application.
122     * ...
123  */
124  @ThreadSafe
125  public class ReactInstanceManager {
```

```
67     protected ReactInstanceManager createReactInstanceManager() {  
68         ReactMarker.logMarker(ReactMarkerConstants.BUILD_REACT_INSTANCE_MANAGER_START);  
69         ReactInstanceManagerBuilder builder =  
70             ReactInstanceManager.builder()  
71                 .setApplication(mApplication)  
72                 .setJSMainModulePath(getJSMainModuleName())
```

```
67     protected ReactInstanceManager createReactInstanceManager() {  
68         ReactMarker.logMarker(ReactMarkerConstants.BUILD_REACT_INSTANCE_MANAGER_START);  
69         ReactInstanceManagerBuilder builder =  
70             ReactInstanceManager.builder()  
71                 .setApplication(mApplication)  
72                 .setJSMainModulePath(getJSMainModuleName())
```



```
806
807     this.mCreateReactContextThread = new Thread((ThreadGroup)null, new Runnable() {
808     public void run() {
809         ReactMarker.LogMarker(ReactMarkerConstants.REACT_CONTEXT_THREAD_END);
810         synchronized(ReactInstanceManager.this.mHasStartedDestroying) {
```

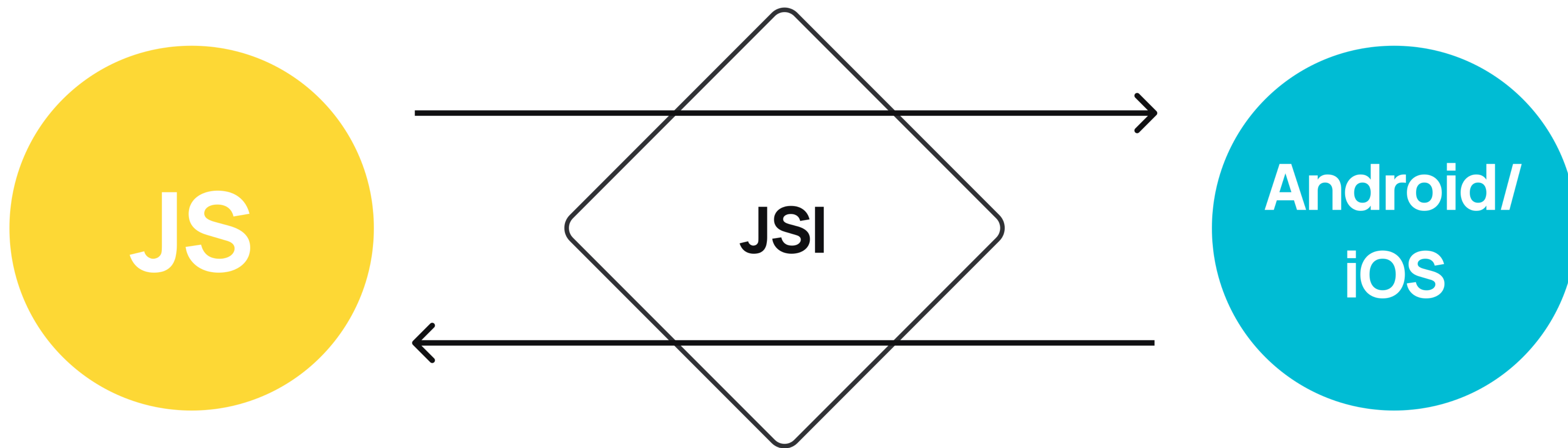
```
862     }
863     }, s: "create_react_context");
864     ReactMarker.LogMarker(ReactMarkerConstants.REACT_CONTEXT_THREAD_START);
865     this.mCreateReactContextThread.start();
```



```
806  
807     this.mCreateReactContextThread = new Thread((ThreadGroup)null, new Runnable() {  
808     public void run() {  
809         ReactMarker.LogMarker(ReactMarkerConstants.REACT_CONTEXT_THREAD_END);  
810         synchronized(ReactInstanceManager.this.mHasStartedDestroying) {
```

```
862     }  
863 }, s: "create_react_context");  
864 ReactMarker.LogMarker(ReactMarkerConstants.REACT_CONTEXT_THREAD_START);  
865 this.mCreateReactContextThread.start();
```

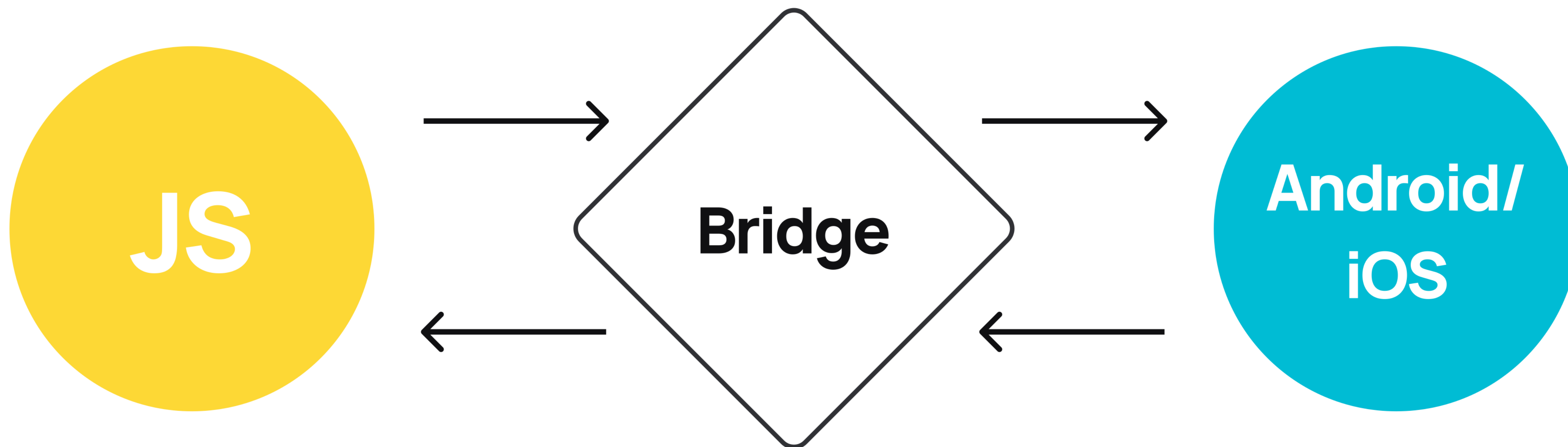
Новая архитектура



Как работает React Native

Так в чем разница между турбо и обычными модулями?

Старая архитектура



Как работает React Native

Вы думаете что JSI не было раньше?

```
21   import {NativeModules} from 'react-native';
```

```
426   get NativeModules(): NativeModules {  
427     return require('./Libraries/BatchedBridge/NativeModules');  
428   },
```

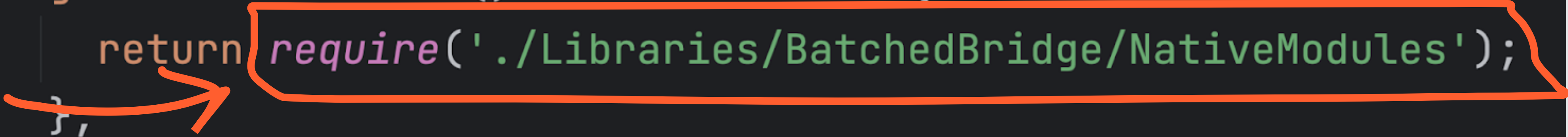
```
21 import {NativeModules} from 'react-native';
```



```
426 get NativeModules(): NativeModules {  
427     return require('./Libraries/BatchedBridge/NativeModules');  
428 },
```

```
21 import {NativeModules} from 'react-native';
```

```
426 get NativeModules(): NativeModules {  
427     return require('./Libraries/BatchedBridge/NativeModules');  
428 },
```

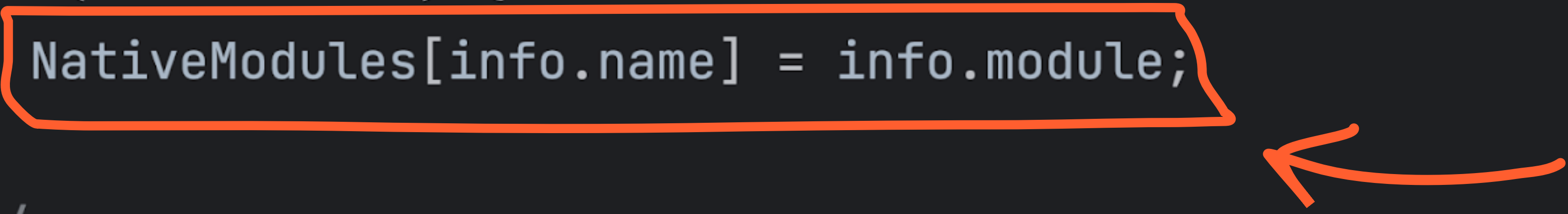


```
176 let NativeModules: {[moduleName: string]: $FlowFixMe, ...} = {};  
177 // ...  
178 const info = genModule(config, moduleID);  
179 v if (!info) {  
180     return;  
181 }  
182  
183 v if (info.module) {  
184     NativeModules[info.name] = info.module;  
185 }  
186 // ...  
187  
188 module.exports = NativeModules;
```

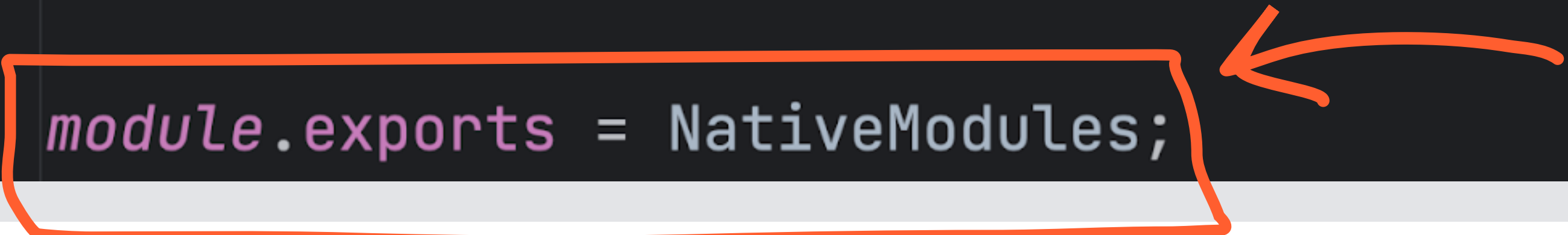


```
176 let NativeModules: {[moduleName: string]: $FlowFixMe, ...} = {};  
177 // ...  
178 const info = genModule(config, moduleID);  
179 v if (!info) {  
180     return;  
181 }  
182  
183 v if (info.module) {  
184     NativeModules[info.name] = info.module;  
185 }  
186 // ...  
187  
188 module.exports = NativeModules;
```

```
176 let NativeModules: {[moduleName: string]: $FlowFixMe, ...} = {};  
177 // ...  
178 const info = genModule(config, moduleID);  
179 v if (!info) {  
180     return;  
181 }  
182  
183 v if (info.module) {  
184     NativeModules[info.name] = info.module;  
185 }  
186 // ...  
187  
188 module.exports = NativeModules;
```



```
176 let NativeModules: {[moduleName: string]: $FlowFixMe, ...} = {};  
177 // ...  
178 const info = genModule(config, moduleID);  
179 v if (!info) {  
180     return;  
181 }  
182  
183 v if (info.module) {  
184     NativeModules[info.name] = info.module;  
185 }  
186 // ...  
187  
188 module.exports = NativeModules;
```



```
176 let NativeModules: {[moduleName: string]: $FlowFixMe, ...} = {};  
177 // ...  
178 const info = genModule(config, moduleID);  
179 √ if (!info) {  
180     return;  
181 }  
182  
183 √ if (info.module) {  
184     NativeModules[info.name] = info.module;  
185 }  
186 // ...  
187  
188 module.exports = NativeModules;
```

```
29 function genModule(  
30   // ...  
31 ) {  
32   // ...  
33   const module = {};  
34   methods &&  
35     methods.forEach((methodName, methodID) => {  
36       // ...  
37       module[methodName] = genMethod(moduleID, methodID, methodType);  
38     });
```



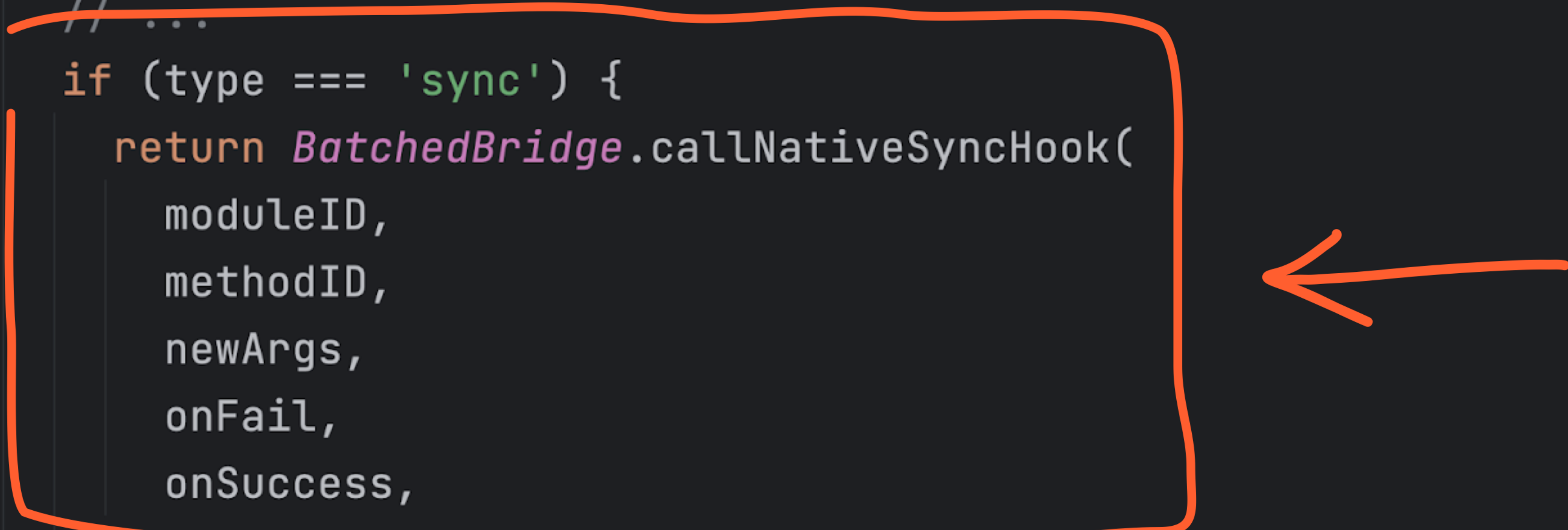
```
29 function genModule(  
30   // ...  
31 ) {  
32   // ...  
33   const module = {};  
34   methods &&  
35   methods.forEach((methodName, methodID) => {  
36     // ...  
37     module[methodName] = genMethod(moduleID, methodID, methodType);  
38   });
```

```
29 function genModule(  
30   // ...  
31 ) {  
32   // ...  
33   const module = {};  
34   methods &&  
35     methods.forEach((methodName, methodID) => {  
36       // ...  
37       module[methodName] = genMethod(moduleID, methodID, methodType);  
38     });
```



```
99 function genMethod(moduleID: number, methodID: number, type: MethodType) {
100     let fn = null;
101     if (type === 'promise') {
102         // ...
103     } else {
104         fn = function nonPromiseMethodWrapper(...args: Array<mixed>) {
105             // ...
106             if (type === 'sync') {
107                 return BatchedBridge.callNativeSyncHook(
108                     moduleID,
109                     methodID,
110                     newArgs,
111                     onFail,
112                     onSuccess,
113                 );
```

```
99 function genMethod(moduleID: number, methodID: number, type: MethodType) {
100   let fn = null;
101   if (type === 'promise') {
102     // ...
103   } else {
104     fn = function nonPromiseMethodWrapper(...args: Array<mixed>) {
105       // ...
106       if (type === 'sync') {
107         return BatchedBridge.callNativeSyncHook(
108           moduleID,
109           methodID,
110           newArgs,
111           onFail,
112           onSuccess,
113         );
```



```
470 Value JSIExecutor::nativeCallSyncHook(const Value *args, size_t count) {
471     // ...
472     MethodCallResult result = delegate_->callSerializableNativeHook(
473         *this, moduleId, methodId, dynamicFromValue(*runtime_, args[2]));
```

```
136 MethodCallResult JavaNativeModule::callSerializableNativeHook(
137     // ...
138 ) {
139     // ...
140     return method->invoke(instance_, wrapper_->getModule(), params);
141 }
```

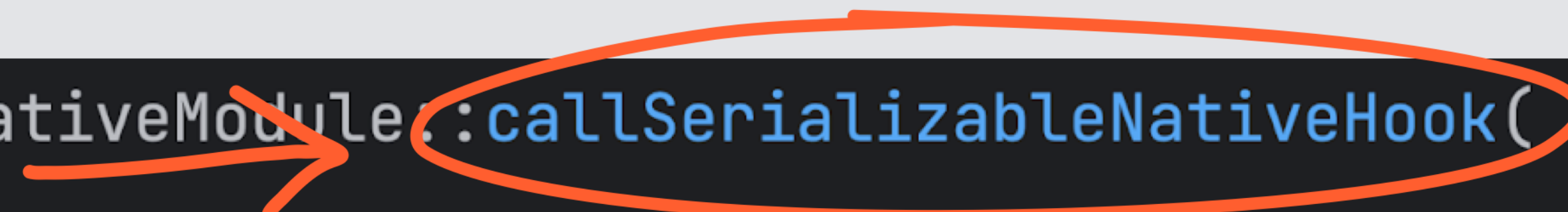


```
470 Value JSIExecutor::nativeCallSyncHook(const Value *args, size_t count) {  
471     // ...  
472     MethodCallResult result = delegate_->callSerializableNativeHook(  
473         *this, moduleId, methodId, dynamicFromValue(*runtime_, args[2]));
```

```
136 MethodCallResult JavaNativeModule::callSerializableNativeHook(  
137     // ...  
138 ) {  
139     // ...  
140     return method->invoke(instance_, wrapper_->getModule(), params);  
141 }
```


```
470 Value JSIExecutor::nativeCallSyncHook(const Value *args, size_t count) {
471     // ...
472     MethodCallResult result = delegate_>callSerializableNativeHook(
473         *this, moduleId, methodId, dynamicFromValue(*runtime_, args[2]));
```

```
136 MethodCallResult JavaNativeModule::callSerializableNativeHook(
137     // ...
138 ) {
139     // ...
140     return method->invoke(instance_, wrapper_>getModule(), params);
141 }
```



```
470 Value JSIExecutor::nativeCallSyncHook(const Value *args, size_t count) {
471     // ...
472     MethodCallResult result = delegate_>callSerializableNativeHook(
473         *this, moduleId, methodId, dynamicFromValue(*runtime_, args[2]));
```

```
136 MethodCallResult JavaNativeModule::callSerializableNativeHook(
137     // ...
138 ) {
139     // ...
140     return method->invoke(instance_, wrapper_>getModule(), params);
141 }
```



Проблемы и примеры их решения

Проблемы и примеры их решения

Пример 1

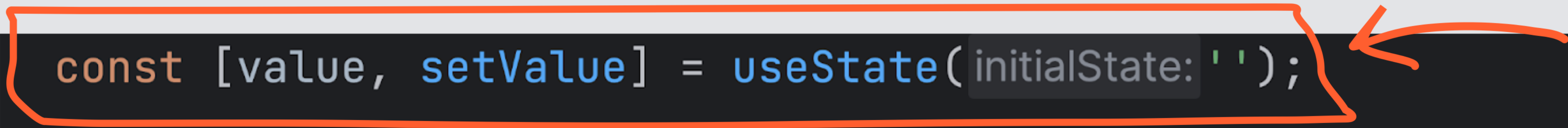
Проблемы и примеры их решения

Номер телефона

(123) 4|

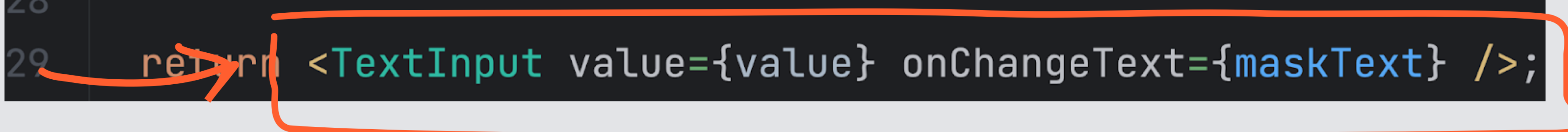
```
23   const [value, setValue] = useState(initialState: '');
24
25 >   function maskText(text: string) {...}
28
29   return <TextInput value={value} onChangeText={maskText} />;
```

```
23  const [value, setValue] = useState(initialState: '');  
24  
25 >  function maskText(text: string) {...}  
28  
29  return <TextInput value={value} onChangeText={maskText} />;
```



```
23   const [value, setValue] = useState(initialState: '');  
24  
25 > function maskText(text: string) {...} ←  
28  
29   return <TextInput value={value} onChangeText={maskText} />;
```

```
23   const [value, setValue] = useState(initialState: '');  
24  
25 > function maskText(text: string) {...}  
28  
29   return <TextInput value={value} onChangeText={maskText} />;
```



Hey hi
Thanks a lot for reaching out.

Нативные SDK предоставляют доступ к событиям до, во время и после нажатия на клавишу

The SDK provides callbacks for events, before, during and after the text editing occurs. For instance, your code may prevent accepting a key press altogether.





AFAP, when I compare to web/React etc., the only option you





В то время как в JS (React Native, React, ...) у вас есть только onChange

some time ago I considered making an **input** part for web, and this was one of the reasons I decided not to.

Let me know if I answered your question (-)

14/08


```
19  ↓ public interface TextWatcher extends NoCopySpan {  
    13 implementations  
20  ↓ public void beforeTextChanged(CharSequence s, int start,  
21                                     int count, int after);  
22  
    13 implementations  
23  ↓ public void onTextChanged(CharSequence s, int start, int before, int count);  
24  
    12 implementations  
25  ↓ public void afterTextChanged(Editable s);  
26 }
```



```
19  ↓ public interface TextWatcher extends NoCopySpan {  
    13 implementations  
20  ↓ public void beforeTextChanged(CharSequence s, int start,  
21     int count, int after);  
22  
    13 implementations  
23  ↓ public void onTextChanged(CharSequence s, int start, int before, int count);  
24  
    12 implementations  
25  ↓ public void afterTextChanged(Editable s);  
26 }
```

```
1022 @Override
1023 public void beforeTextChanged(CharSequence s, int start, int count, int after) {
1024     mPreviousText = s.toString();
1025 }
1026
1027 @Override
1028 public void onTextChanged(CharSequence s, int start, int before, int count) {
1029     ...
1030     mEventDispatcher.dispatchEvent(new ReactTextChangedEvent(...));
1031     ...
1032 }
1033
1034 @Override
1035 public void afterTextChanged(Editable s) {}
1036 }
```

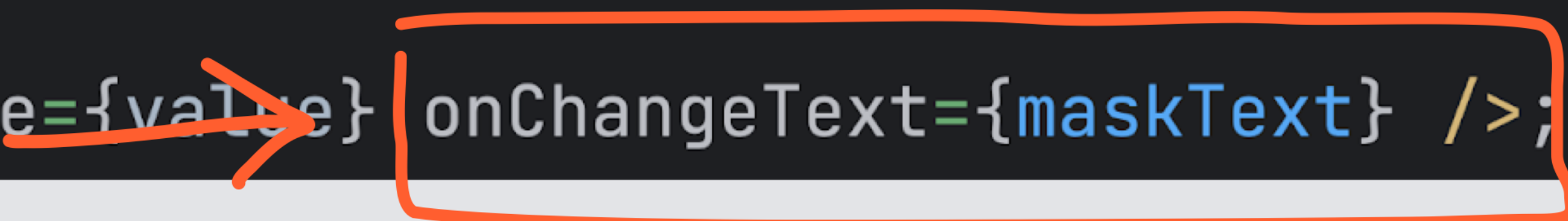
```
1022 @Override
1023 public void beforeTextChanged(CharSequence s, int start, int count, int after) {
1024     mPreviousText = s.toString(),
1025 }
1026
1027 @Override
1028 public void onTextChanged(CharSequence s, int start, int before, int count) {
1029     ...
1030     mEventDispatcher.dispatchEvent(new ReactTextChangedEvent(...));
1031     ...
1032 }
1033
1034 @Override
1035 public void afterTextChanged(Editable s) {}
1036 }
```



```
1022 @Override
1023 public void beforeTextChanged(CharSequence s, int start, int count, int after) {
1024     mPreviousText = s.toString();
1025 }
1026
1027 @Override
1028 public void onTextChanged(CharSequence s, int start, int before, int count) {
1029     ...
1030     mEventDispatcher.dispatchEvent(new ReactTextChangedEvent(...));
1031     ...
1032 }
1033
1034 @Override
1035 public void afterTextChanged(Editable s) {}
1036 }
```

```
1022 @Override
1023 public void beforeTextChanged(CharSequence s, int start, int count, int after) {
1024     mPreviousText = s.toString();
1025 }
1026
1027 @Override
1028 public void onTextChanged(CharSequence s, int start, int before, int count) {
1029     
1030     mEventDispatcher.dispatchEvent(new ReactTextChangedEvent(...)); 
1031 }
1032
1033
1034 @Override
1035 public void afterTextChanged(Editable s) {}
1036 }
```

```
23  const [value, setValue] = useState(initialState: '');  
24  
25 > function maskText(text: string) {...}  
28  
29  return <TextInput value={value} onChangeText={maskText} />;
```



Решение:

Потратить 10 секунд на гуглежку

react-native-text-input-mask

Text input mask for React Native on iOS and Android.

npm package

3.2.0

license

MIT

<https://github.com/react-native-text-input-mask>

Проблемы и примеры их решения

Номер телефона

+7 (

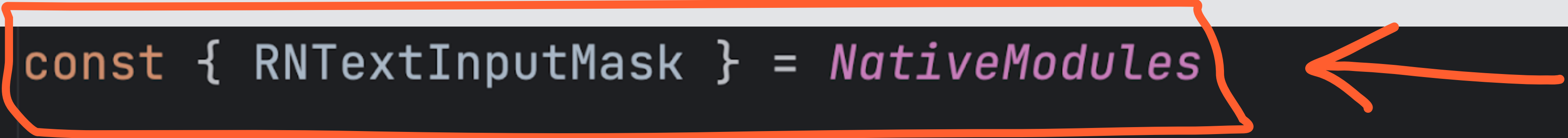
```
81     return (  
82         <TextInput  
83             {...rest}  
84             ref={input}  
85             value={maskedValue}  
86             onChangeText={async (masked) => {  
87                 // ...  
88             }}  
89         />  
90     )
```

```
81     return (  
82         <TextInput  
83             {...rest}  
84             ref={input}  
85             value={maskedValue}  
86             onChangeText={async (masked) => {  
87                 // ...  
88             }}  
89         />  
90     )
```

```
13  const { RNTextInputMask } = NativeModules
14
15  export const { mask, unmask, setMask } = RNTextInputMask
16  // ...
17  const input = useRef<TextInput>( initialValue: null )
18  // ...
19  const nodeId = findNodeHandle(input.current)
20  if (nodeId) {
21    setMask(nodeId)
22  }
```

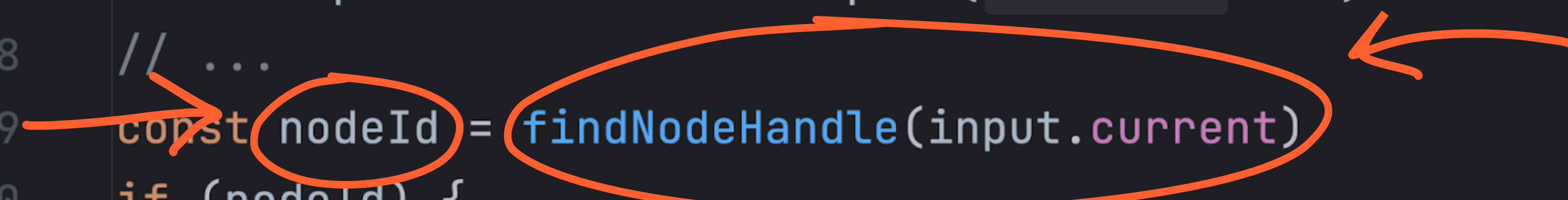


```
13  const { RNTextInputMask } = NativeModules
14
15  export const { mask, unmask, setMask } = RNTextInputMask
16  // ...
17  const input = useRef<TextInput>( initialValue: null )
18  // ...
19  const nodeId = findNodeHandle(input.current)
20  if (nodeId) {
21    setMask(nodeId)
22  }
```



```
13  const { RNTextInputMask } = NativeModules
14
15  export const { mask, unmask, setMask } = RNTextInputMask
16  // ...
17  const input = useRef<TextInput>( initialValue: null)
18  // ...
19  const nodeId = findNodeHandle(input.current)
20  if (nodeId) {
21    setMask(nodeId)
22  }
```

```
13  const { RNTextInputMask } = NativeModules
14
15  export const { mask, unmask, setMask } = RNTextInputMask
16  // ...
17  const input = useRef<TextInput>( initialValue: null)
18  // ...
19  const nodeId = findNodeHandle(input.current)
20  if (nodeId) {
21    setMask(nodeId)
22  }
```




```
13  const { RNTextInputMask } = NativeModules
14
15  export const { mask, unmask, setMask } = RNTextInputMask
16  // ...
17  const input = useRef<TextInput>( initialValue: null)
18  // ...
19  const nodeId = findNodeHandle(input.current)
20  if (nodeId) {
21    setMask(nodeId) ←
22  }
```




```
55     @ReactMethod
56     fun setMask(tag: Int) {
57         // ...
58         val editText = nativeViewHierarchyManager.resolveView(tag) as EditText
59         // ...
60         MaskedTextChangedListener(
61             // ...
62             field = editText,
63         )
64     }
```

```
55 @ReactMethod
56 fun setMask(tag: Int) {
57     // ...
58     val editText = nativeViewHierarchyManager.resolveView(tag) as EditText
59     // ...
60     MaskedTextChangedListener(
61         // ...
62         field = editText,
63     )
64 }
```




```
55 @ReactMethod
56 fun setMask(tag: Int) {
57     // ...
58     val editText = nativeViewHierarchyManager.resolveView(tag) as EditText
59     // ...
60     MaskedTextChangedListener(
61         // ...
62         field = editText,
63     )
64 }
```



```
9 import com.redmadrobot.inputmask.MaskedTextChangedListener
```

```
1  override fun onTextChanged(text: CharSequence, cursorPosition: Int, before: Int, count: Int) {  
2      // ...  
3      val mask: Mask = this.pickMask(textAndCaret)  
4      val result: Mask.Result = mask.apply(textAndCaret)  
5      // ...  
6  }
```

<https://github.com/RedMadRobot/input-mask-android>

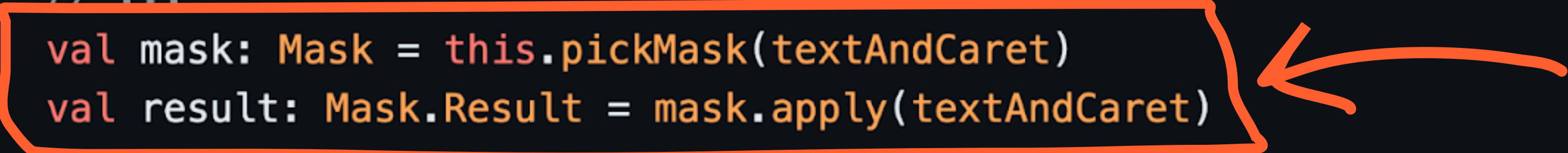
```
9 import com.redmadrobot.inputmask.MaskedTextBoxChangeListener
```

```
1 override fun onTextChanged(text: CharSequence, cursorPosition: Int, before: Int, count: Int) {  
2     // ...  
3     val mask: Mask = this.pickMask(textAndCaret)  
4     val result: Mask.Result = mask.apply(textAndCaret)  
5     // ...  
6 }
```

<https://github.com/RedMadRobot/input-mask-android>

```
9 import com.redmadrobot.inputmask.MaskedTextBoxChangeListener
```

```
1 override fun onTextChanged(text: CharSequence, cursorPosition: Int, before: Int, count: Int) {  
2     // ...  
3     val mask: Mask = this.pickMask(textAndCaret)  
4     val result: Mask.Result = mask.apply(textAndCaret)  
5     // ...  
6 }
```

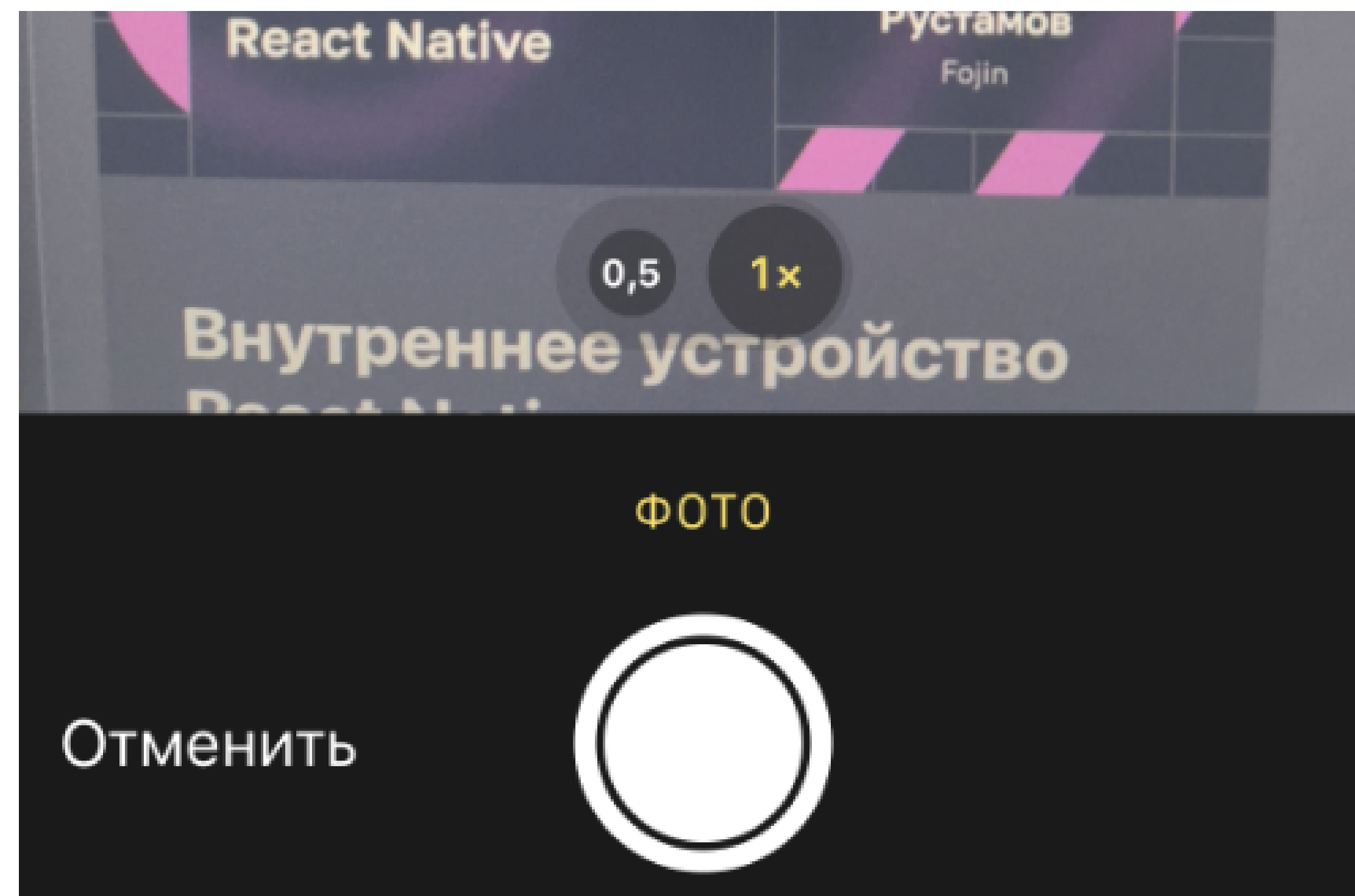
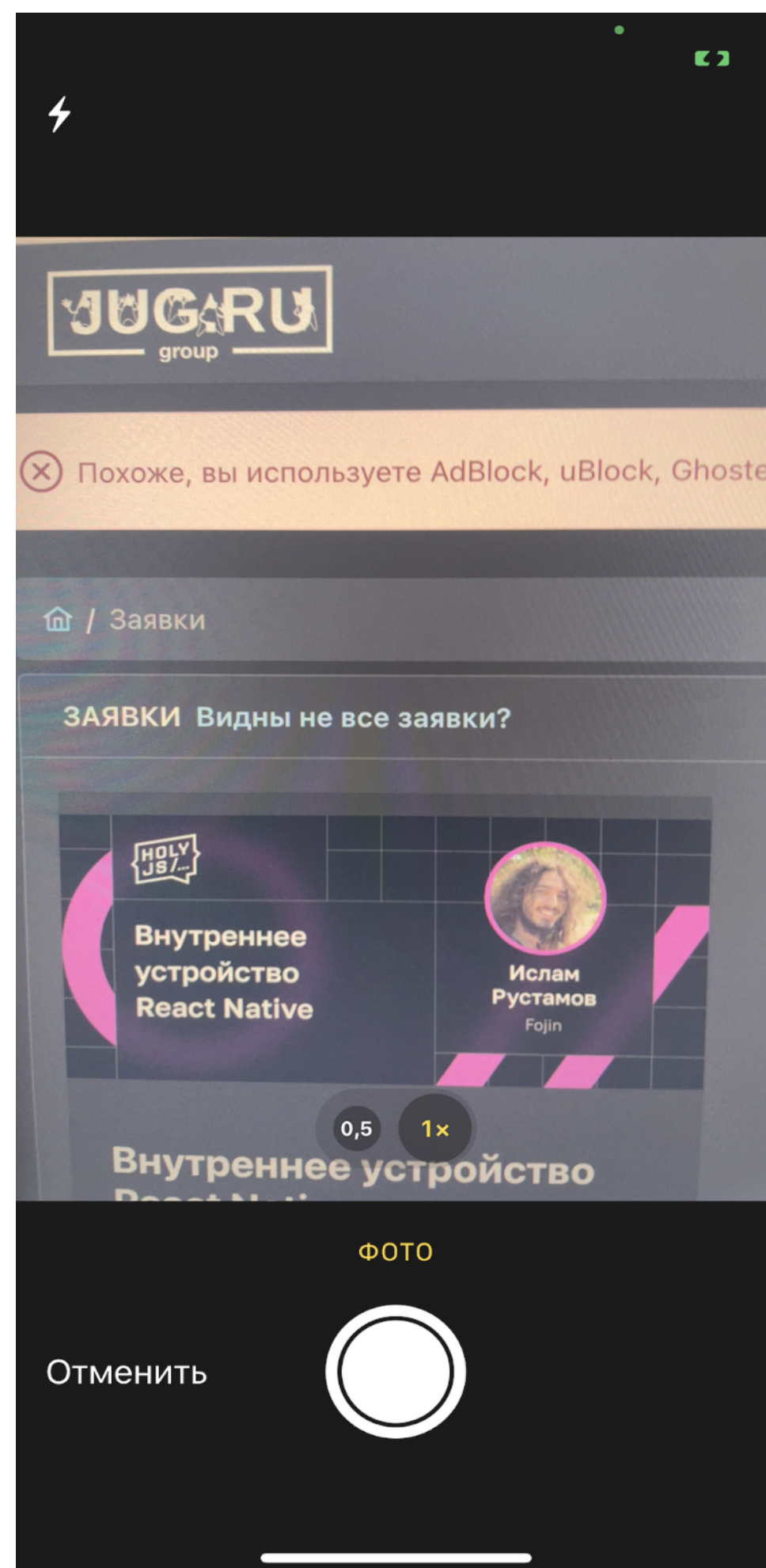


<https://github.com/RedMadRobot/input-mask-android>


```
1022 @Override
1023 public void beforeTextChanged(CharSequence s, int start, int count, int after) {
1024     mPreviousText = s.toString();
1025 }
1026
1027 @Override
1028 public void onTextChanged(CharSequence s, int start, int before, int count) {
1029     ...
1030     mEventDispatcher.dispatchEvent(new ReactTextChangedEvent(...));
1031     ...
1032 }
1033
1034 @Override
1035 public void afterTextChanged(Editable s) {}
1036 }
```

Проблемы и примеры их решения

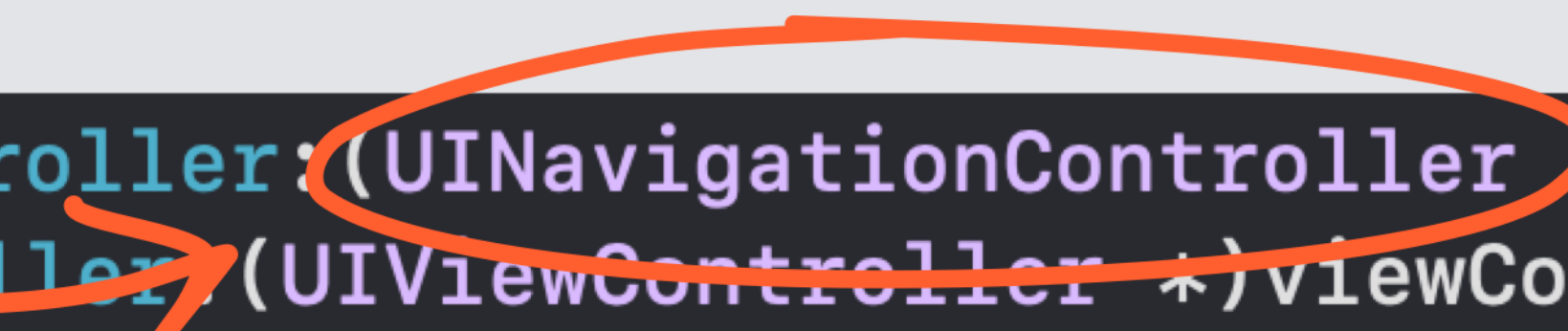
Пример 2



<https://github.com/react-native-image-picker>


```
132 - (void)navigationController:(UINavigationController *)navigationController
133     willShowViewController:(UIViewController *)viewController
134     animated:(BOOL)animated {
135     [self hideFlipButtonInSubviews:viewController.view];
136 }
```

```
132 - (void)navigationController:(UINavigationController *)navigationController
133     willShowViewController:(UIViewController *)viewController
134     animated:(BOOL)animated {
135     [self hideFlipButtonInSubviews:viewController.view];
136 }
```



```
132 - (void)navigationController:(UINavigationController *)navigationController
133     willShowViewController:(UIViewController *)viewController
134     animated:(BOOL)animated {
135     [self hideFlipButtonInSubviews:viewController.view];
136 }
```

```
132 - (void)navigationController:(UINavigationController *)navigationController
133     willShowViewController:(UIViewController *)viewController
134     animated:(BOOL)animated {
135     [self hideFlipButtonInSubviews:viewController.view];
136 }
```





```
68 - (void)hideFlipButtonInSubviews:(UIView *)view {
69     if ([[view class] description] isEqualToString:@"CAMFlipButton"] ||
70         [[view class] description] isEqualToString:@"CMKFlipButton"]) {
71         [view removeFromSuperview];
72     } else {
73         for (UIView *subview in [view subviews]) {
74             [self hideFlipButtonInSubviews:subview];
75         }
76     }
}
```

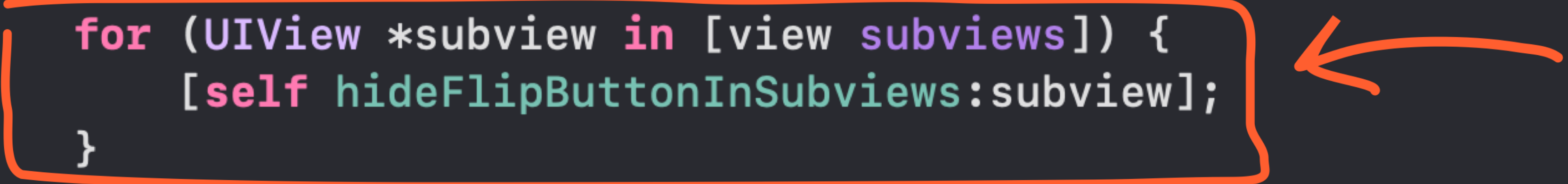


```
68 - (void)hideFlipButtonInSubviews:(UIView *)view {
69     if ([[view class] description] isEqualToString:@"CAMFlipButton"] ||
70         [[view class] description] isEqualToString:@"CMKFlipButton"]) {
71         [view removeFromSuperview];
72     } else {
73         for (UIView *subview in [view subviews]) {
74             [self hideFlipButtonInSubviews:subview];
75         }
76     }
}
```

```
68 - (void)hideFlipButtonInSubviews:(UIView *)view {
69     if ([[view class] description] isEqualToString:@"CAMFlipButton"] ||
70         [[[view class] description] isEqualToString:@"CMKFlipButton"]) {
71         [view removeFromSuperview];
72     } else {
73         for (UIView *subview in [view subviews]) {
74             [self hideFlipButtonInSubviews:subview];
75         }
76     }
}
```



```
68 - (void)hideFlipButtonInSubviews:(UIView *)view {
69     if ([[view class] description] isEqualToString:@"CAMFlipButton"] ||
70         [[view class] description] isEqualToString:@"CMKFlipButton"]) {
71         [view removeFromSuperview];
72     } else {
73         for (UIView *subview in [view subviews]) {
74             [self hideFlipButtonInSubviews:subview];
75         }
76     }
}
```



Проблемы и примеры их решения

Как сделать то же самое на андроиде?

```
88 cameraIntent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
```

```
102 currentActivity.startActivityForResult(cameraIntent, requestCode);
```

```
88 cameraIntent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
```

```
102 currentActivity.startActivityForResult(cameraIntent, requestCode);
```



```
88 cameraIntent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
```

```
102 currentActivity.startActivityForResult(cameraIntent, requestCode);
```



```
182  @Override
183  public void onActivityResult(
184      Activity activity,
185      int requestCode,
186      int resultCode,
187      Intent data
188  ) {
189      // ...
```

Проблемы и примеры их решения

Через интенты - никак

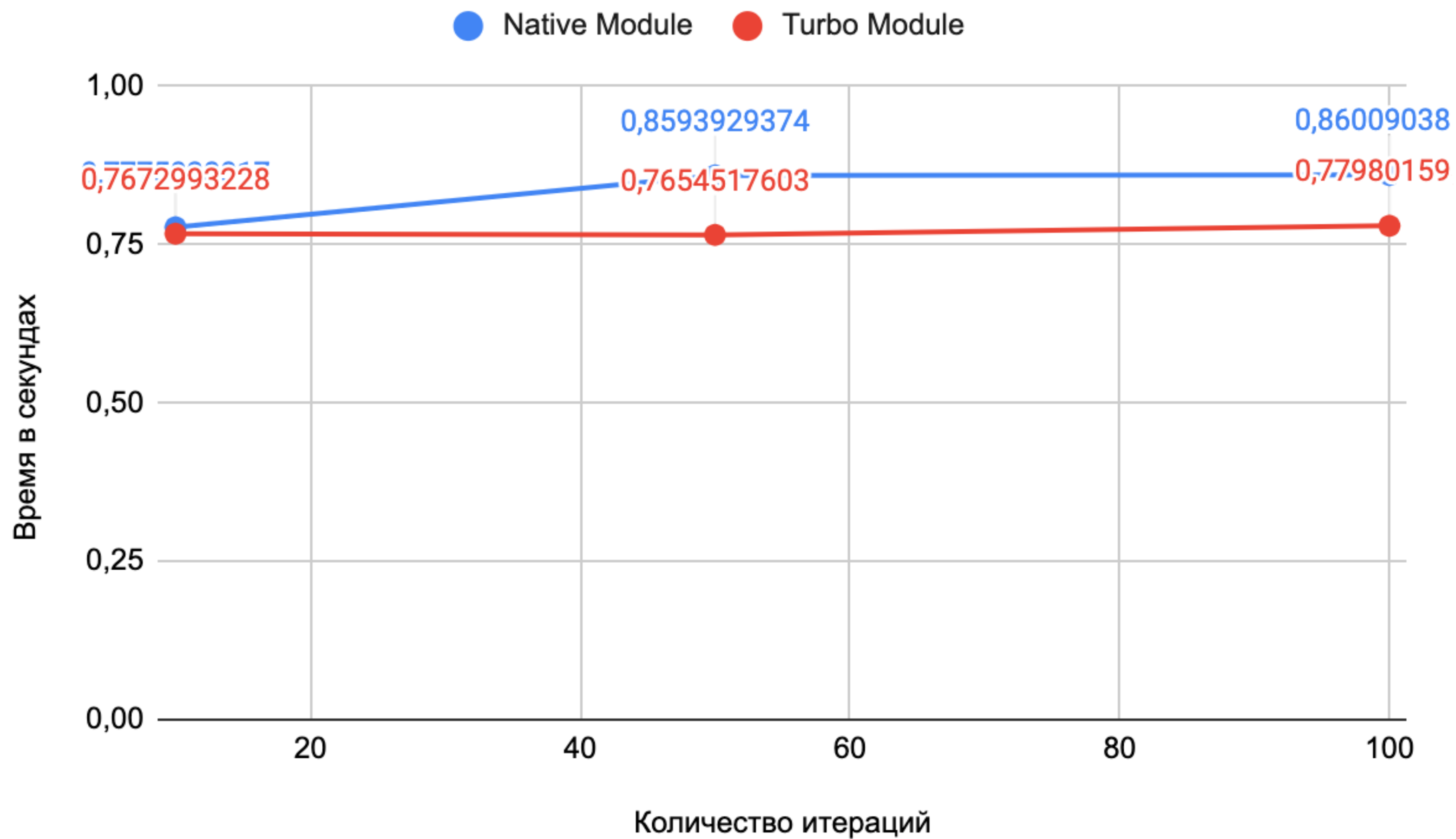
Сравнение производительности

Сравнения производительности

Турбо VS не турбо

- 1. Запускаем сжатие N раз**
- 2. Каждое сжатие засекаем и берем среднее**
- 3. Увеличиваем N и повторяем**

Меньше = лучше



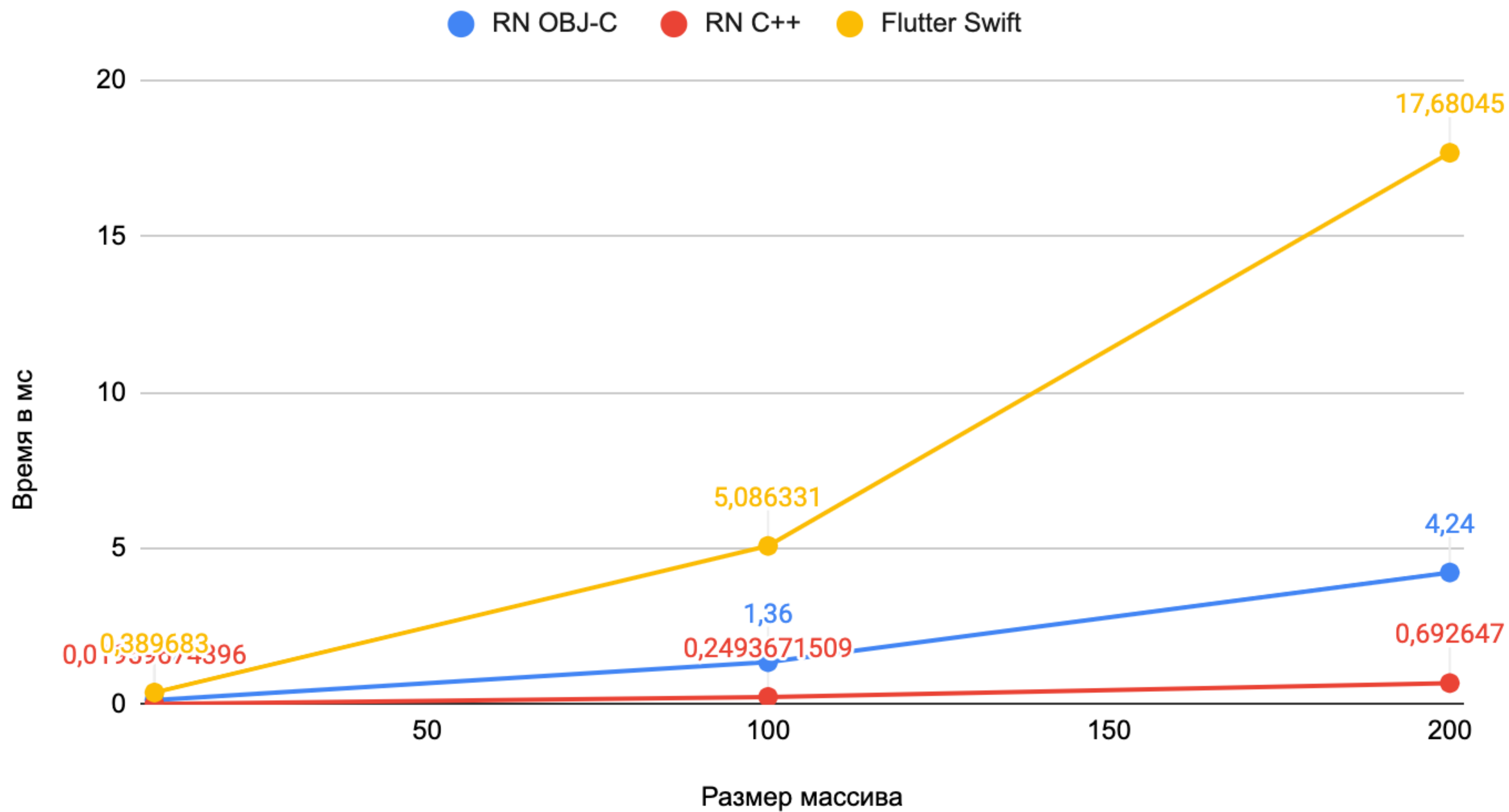
Сравнения производительности

Взаимодействие с нативным КОДОМ

- 1. Турбо модуль на C++**
- 2. Турбо модуль на Objective-C**
- 3. Platform Channel на Swift**

- 1. Запускаем бабл сортировку
1000 раз**
- 2. Берем среднее**
- 3. Увеличиваем размер массива**

Меньше = лучше



Сравнения производительности

Тестирование тяжелой анимации

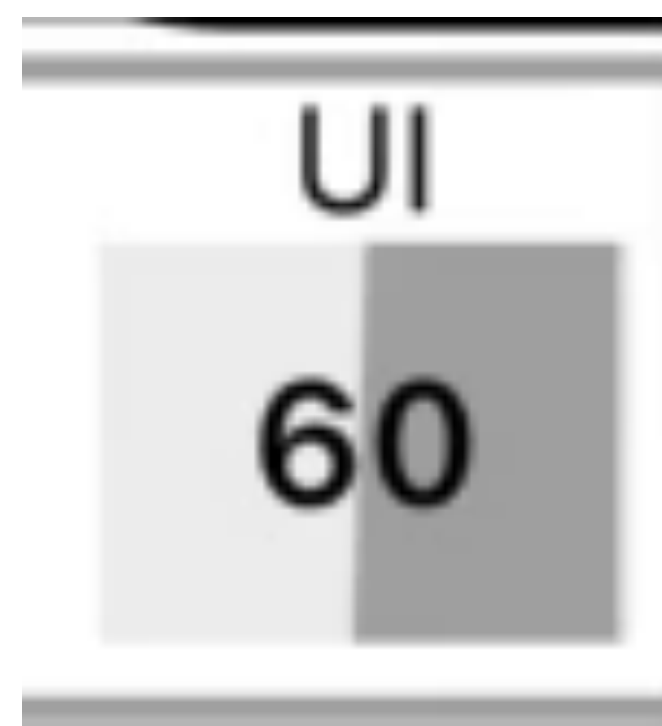
Reanimated



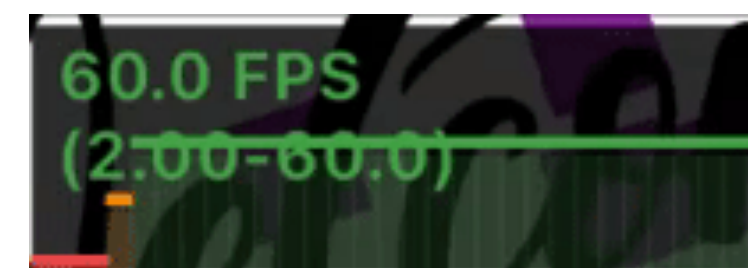
Flutter



100 элементов

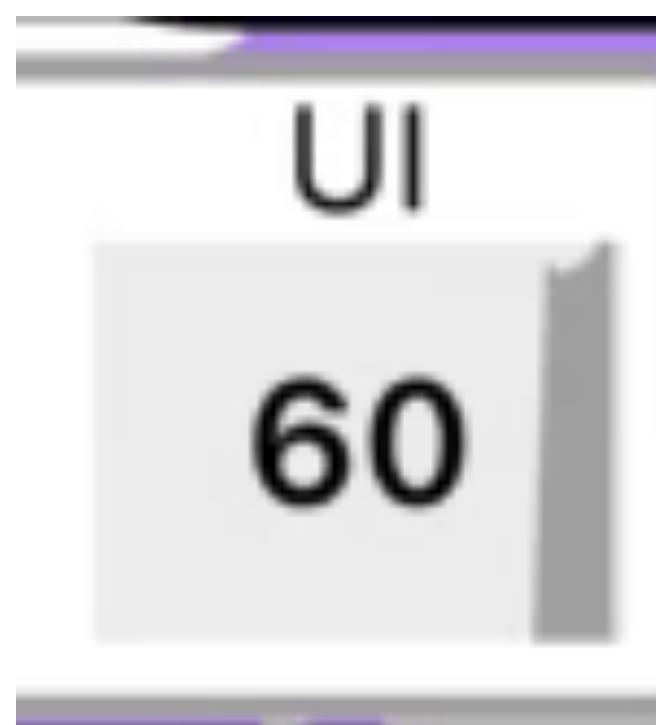


~60 FPS

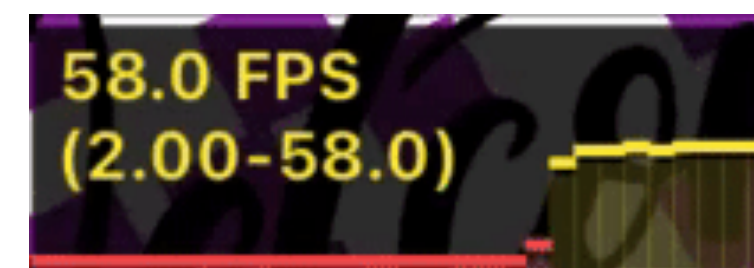


~60 FPS

500 элементов



~60 FPS

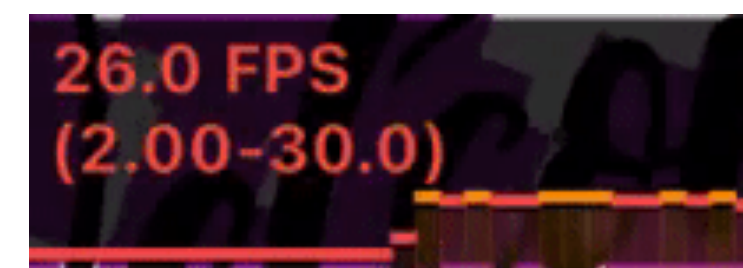


~60 FPS

1500 элементов



~20 FPS



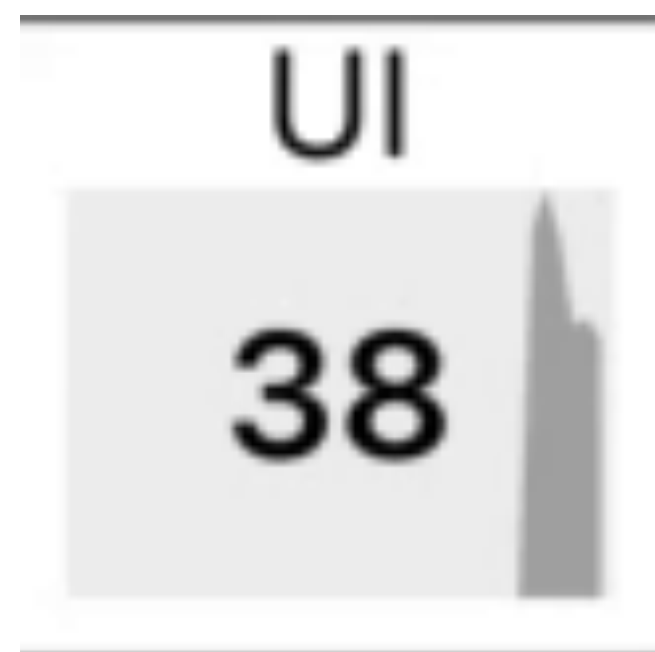
~30 FPS

React Native Skia

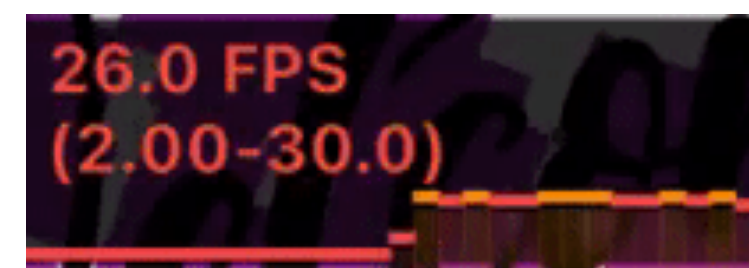
High Performance 2D Graphics

<https://github.com/Shopify/react-native-skia>

1500 элементов

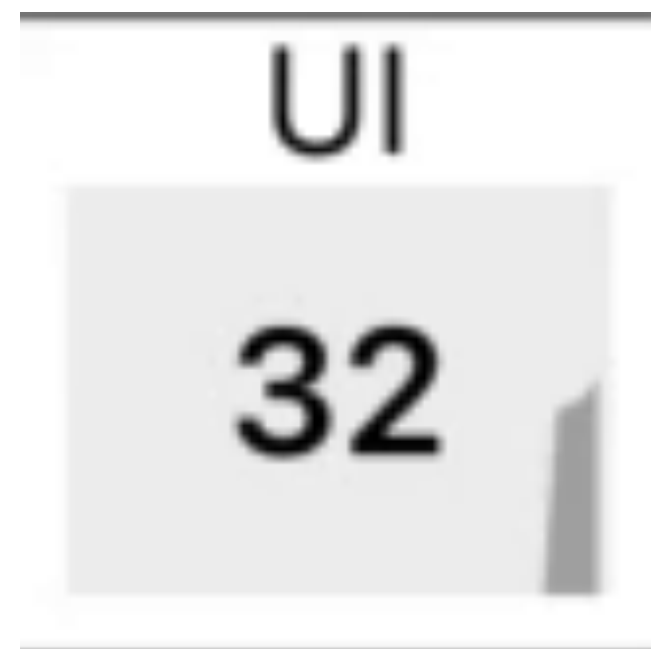


~37 FPS

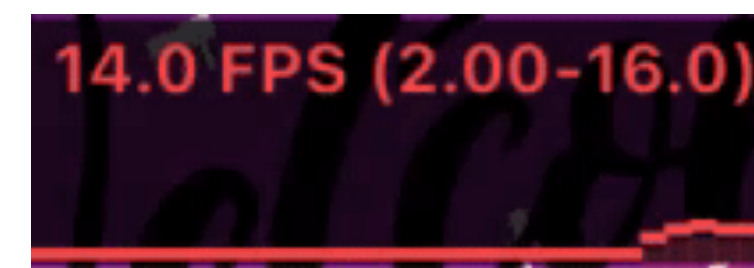


~30 FPS

3000 элементов



~30 FPS



~15 FPS



wcandillon 3 weeks ago

Maintainer

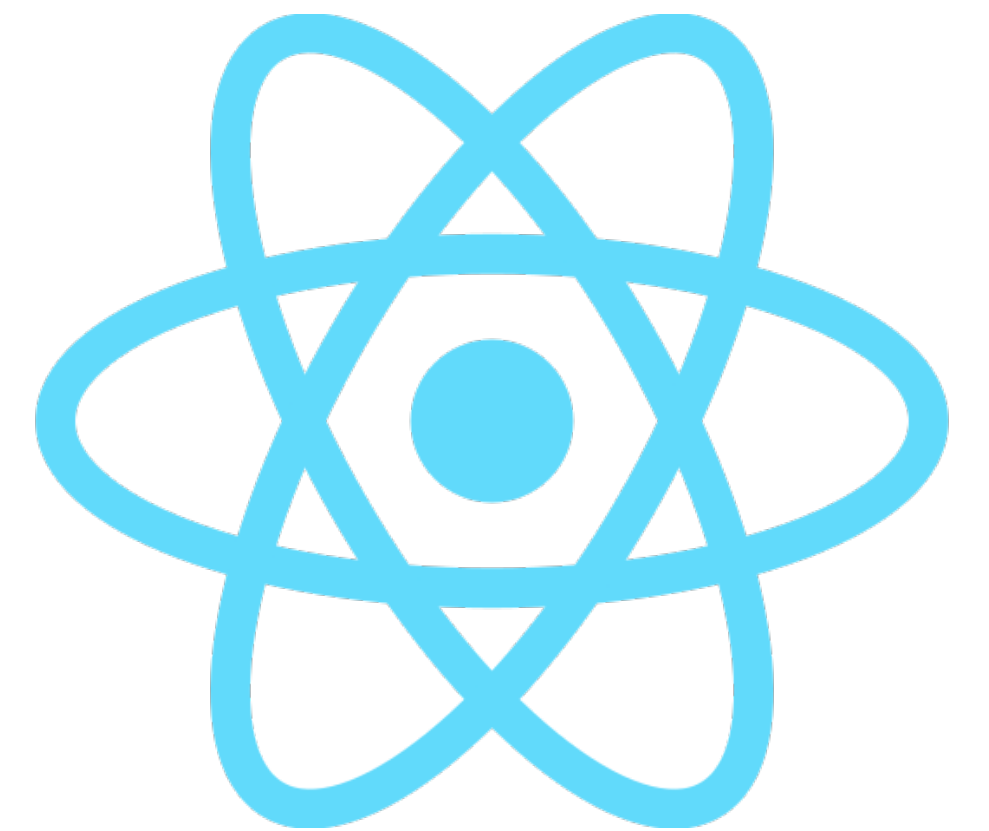
Indeed, threads have nothing to do with performance; it's more about leaving the thread free to do other things.

In the case of Reanimated 3, I mentioned the UI thread because it appears to be much faster and leaves the JS thread free to perform other tasks.

<https://github.com/Shopify/react-native-skia/discussions/1824>

Какие итоги?

Узнали новое о RN



Bridge - это не так страшно

Hermes - это не всегда круто (но почти всегда круто)

На что способен RN

RN может всё (что может iOS и Android)



Но надо учить нативную разработку...



**RN становится лучше
и лучше**



Откуда брался хейт RN'а?

**Спасибо
за внимание!**

t.me/islamelninio



Использованные
ИСТОЧНИКИ