

REACT NATIVE

FROM A MOBILE (IOS)
DEVELOPER PROSPECTIVE

ME

```
let sex = ♂  
var job = "iOS Dev"  
var company = 🚀⼯厂 // Rocketfarm
```

```
let ❤️ = 🐦 // "Swift"  
❤️🐦 < ❤️👑 == true
```

```
var like = 💻 + 📖 + ✎
```

COMING SOON. :)

var author = 🧑 + 📖 + 🐦

Swift High Performance

Kostiantyn Koval

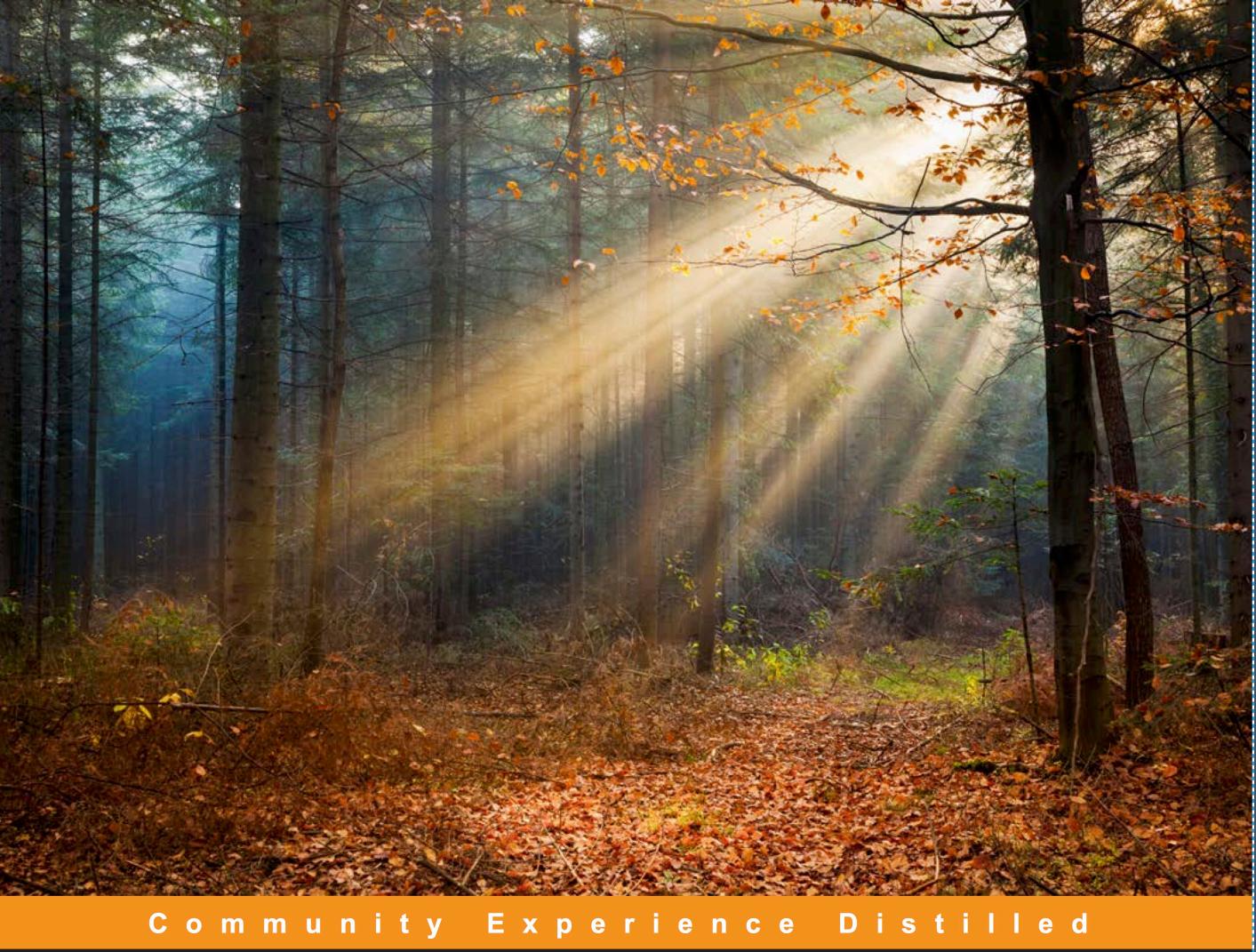
[PACKT
PUBLISHING]

Swift High Performance

Leverage Swift and enhance your code to take your applications to the next level

Kostiantyn Koval

[PACKT
PUBLISHING]



FIRST REACTION



A dramatic scene featuring a lightning storm with numerous bright blue and white bolts striking across a dark sky. In the foreground, two vehicles are driving away from the viewer on a road. On the left is a red DeLorean DMC-12 with its headlights on, and on the right is a silver GMC Vandura van. Both vehicles have California license plates. The ground in the foreground is partially obscured by a large, bright orange and yellow flame or fire effect.

BACK TO THE FUTURE

BACK TO THE FUTURE

- > 10 JULY 2008
- > MID-2010
- > 2 APRIL 2009

BACK TO THE FUTURE

- › JULY 10 2008: IPHONE 3G, APP STORE AND FACEBOOK APP
- › MID-2010: FACEBOOK'S "WRITE ONCE, RUN EVERYWHERE"
- › 2 APRIL 2009 PHONEGAP - WINS PEOPLE'S CHOICE AWARD AT WEB 2.0 CONFERENCE

STAGE 2



STAGE 2



- > BE CURIOUS
 - > OPEN MINDED
 - > LIFE LEARNING
-

THE GOAL IS TO LEARN

STAGE 3



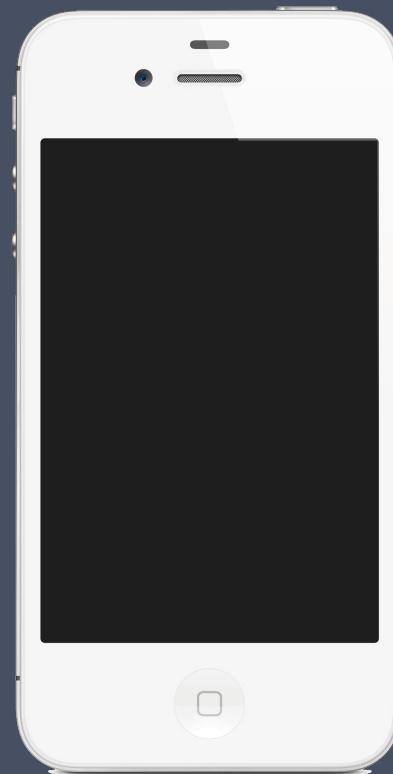
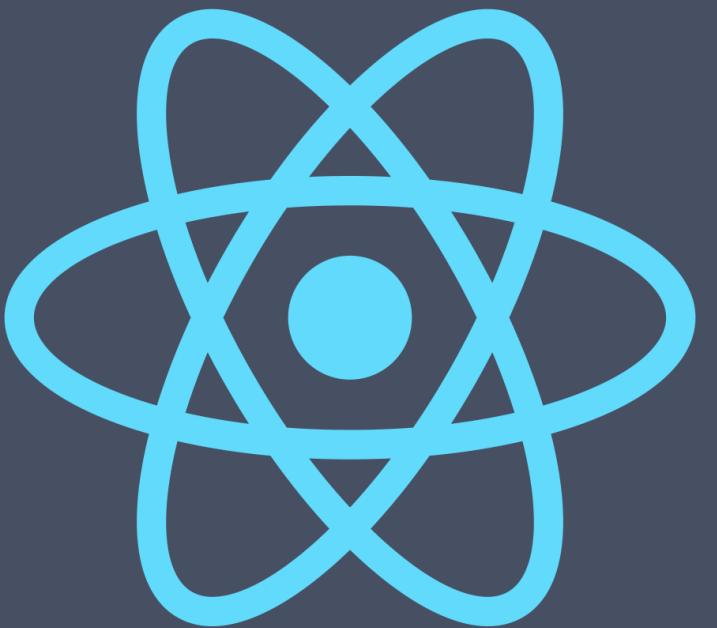
STAGE 3



- > NATIVE UI
- > DECLARATIVE UI
- > LEARN ONCE. WRITE EVERYWHERE

NATIVE APP

WRITE IN WEB. GET NATIVE AP



```
<Text>  
  Welcome to React Native!  
</Text>
```

```
<ListView  
  dataSource= {this.state.dataSource}  
  renderRow= {this.renderMyCell}  
  style= {styles.listView}  
/>
```



```
UITextView *textView = [UITextView new];  
textView.text = @"Welcome to React Native!";
```

```
UITableView *tableView = [UITableView new];  
tableView.dataSource = self;  
[tableView registerClass:[MyCell class] forCellReuseIdentifier:@"MyCellId"];
```

EACH REACT NATIVE UI COMPONENT
MAPS TO IOS NATIVE UI ELEMENT

* PLATFORM SPECIFIC NATIVE COMPONENT

BUT IT'S JS 😞

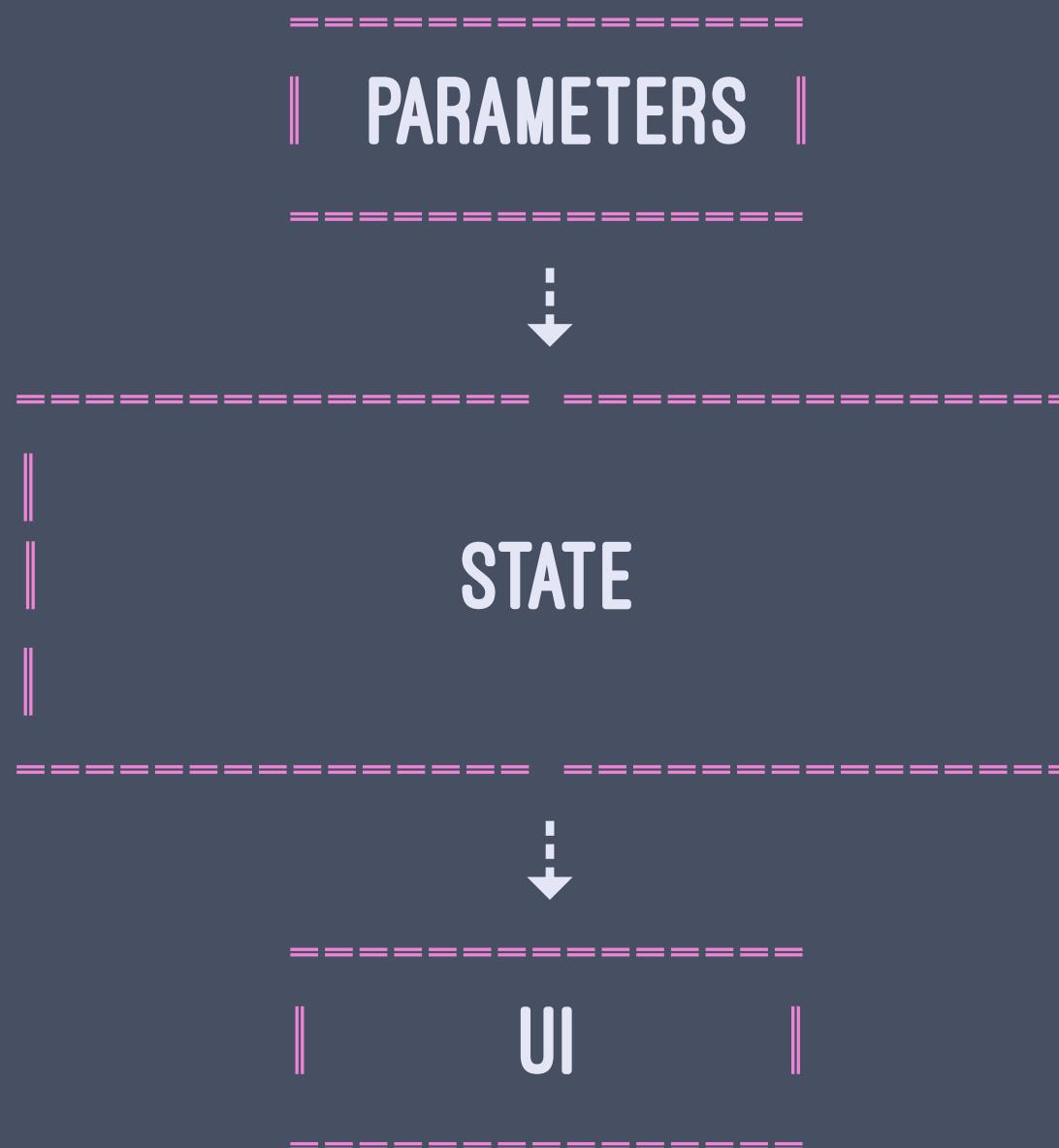
THE BENEFITS ARE BIGGER,
JUST TRUST ME 😊

JS

DECLARATIVE UI

- > STATE → UI
- = = =
- > INDEPENDENT COMPOSABLE COMPONENTS
- > LOCAL COMPONENTS STATE. ~~NO GLOBAL~~
- > ONE WAY DATA FLOW

DECLARATIVE UI - COMPONENT



LEARN ONCE. WRITE EVERYWHERE

REACT

: REACT JS :

: REACT NATIVE IOS :

: REACT NATIVE ANDROID :

TIME TO FACE REAL PROJECTS PROBLEMS



TIME TO FACE REAL PROJECTS PROBLEMS

REACT - JS



NATIVE - (SWIFT OR OBJC)

TIME TO FACE REAL PROBLEMS

- > HARDWARE ACCESS
- > NATIVE CODE (**SWIFT AND OBJECTIVE-C**)
 - > PERFORMANCE

=====

- > RELIABILITY
- > SUPPORT AND COMMUNITY

NATIVE CODE (OBJC)

```
//Objective-C
@implementation MyComponent
RCT_EXPORT_MODULE();

//RCT_EXPORT_METHOD = - (void)
RCT_EXPORT_METHOD(sendString:(NSString *)string) {
    NSLog(@"Received: %@", string);
}

@end
```

= = =

```
//JS
var ObjcComponent = require('react-native').NativeModules.MyComponent;
ObjcComponent.sendString('Hello Objc');
```

NATIVE CODE (SWIFT)

```
//Swift
@objc(MySwiftComponent) class MySwiftComponent : NSObject {

    @objc func sendString(x: String) {
        print("Received: \(x)")
    }
}
```

```
//Objective-C
@interface RCT_EXTERN_MODULE(MySwiftComponent, NSObject)

RCT_EXTERN_METHOD(sendString:(NSString *)x)

@end
```

```
//JS
var SwiftComponent = require('react-native').NativeModules.MySwiftComponent;
SwiftComponent.sendString('Swift all the things 😎');
```

NATIVE → JS

```
//Objective-C
RCT_EXPORT_METHOD(getString:(RCTResponseSenderBlock)callback {
    callback(@[[NSNull null], @"Result"]);
}
```

```
@objc func getString(response: RCTResponseSenderBlock ) {
    response([NSNull(), "Result"])
}
//+ RCT_EXTERN_METHOD(getString:(RCTResponseSenderBlock)response) in Objc
```

```
//JS
ObjcComponent.getString((error, param) => {
    console.error(error + " " + param);
});
```

NATIVE → JS (PROMISES)

```
RCT_EXPORT_METHOD(getStringAsync:(RCTPromiseResolveBlock)resolve rejecter:(RCTPromiseRejectBlock)reject) {  
  
    BOOL yesNo = YES;  
    if (yesNo) {  
        resolve(@"Yes");  
    } else {  
        reject([NSError errorWithDomain:@"NO!" code:0 userInfo:nil]);  
    }  
}  
  
@objc func getStringAsync(resolve: RCTPromiseResolveBlock, reject: RCTPromiseRejectBlock) {  
    let yesNo = true  
    if (true) {  
        resolve("Yes")  
    } else {  
        reject(NSError(domain:"NO!", code:0, userInfo:nil))  
    }  
}  
//+ RCT_EXTERN_METHOD(getStringAsync:(RCTPromiseResolveBlock)resolve reject:(RCTPromiseRejectBlock)reject)
```

= = JS = =

NATIVE → JS (PROMISES)

```
//Objective-C
- (void)getStringAsync:(RCTPromiseResolveBlock)resolve rejecter:(RCTPromiseRejectBlock)reject
```

```
//JS
ObjcComponent.getStringAsync().then((value) => {
  console.log(value);
}, (error) => {
  console.log(error);
});
```

```
// ES7 + Babel JavaScript compiler
async function fetchString() {
  let string = await ObjcComponent.getStringAsync();
  console.log("Async +await: " + string);
}
```

NATIVE → JS (EVENTS)

```
#import "RCTEventDispatcher.h"

@synthesize bridge = _bridge; // Implementing RCTBridgeModule

- (void)sendEventToJS {
    [self.bridge.eventDispatcher sendAppEventWithName:@"MyEvent"
                                                body:@{@"param": @"Hello JS"}];
}

var { NativeAppEventEmitter } = React;
var subscription = NativeAppEventEmitter.addListener('MyEvent', (event) =>
    console.log("§ Event: " + event)
);
```



BUT `synthesize` IS NOT AVAILABLE IN SWIFT

UI COMPONENT

```
@interface RCTMyViewManager : RCTViewManager
@end

@implementation RCTMyViewManager

RCT_EXPORT_MODULE()

- (MyView *)view {
    return [MyView new];
}

@end

//JS
var { requireNativeComponent } = require('react-native');
var MyView = requireNativeComponent('RCTMyView', null);
module.exports = MyView

render: function() {
    <MyView> </MyView>
}
```

UI COMPONENT



- > CUSTOM PROPERTIES
- > EVENTS
- > JS COMPONENT WRAPPER
- > STYLES

...

LIMITATIONS

- > METHODS: INSTANCE ONLY
- > INSTANCE CREATION CONTROL: NONE.
1 INSTANCE ONLY. CREATED AND CACHED BY NativeModules
- > INITIALIZER: +(instancetype)new
- > ARGUMENTS: JSON TYPES + RCTCONVERT
- > COMMUNICATION: ASYNCHRONOUS

PERFORMANCE



PERFORMANCE



60 FPS

PERFORMANCE

- > REACT HANDLES DOM MANIPULATION AND RE-RENDERING
 - > ANIMATIONS + NAVIGATION TRANSITION
 - > REACT JS CODE RUNS ON BACKGROUND THREAD
(MAIN THREAD = UI ONLY)

JAVASCRIPTCORE

IOS 7 - WEBKIT JAVASCRIPT ENGINE
* NO JIT SUPPORT

STABILITY

- > SOLID ARCHITECTURE 💪
- > BETA V0.12.0
- > RAPID DEVELOPMENT/CHANGES
- > APPS WORKS FINE. NO CRASHES
 - > IN PRODUCTION
 - FACEBOOK GROUPS. FACEBOOK APP. LEANPUB ...

SUPPORT AND COMMUNITY

> OPEN SOURCED 

ALL IS IN YOUR HANDS +

PR ARE WELCOMED 

> ACTIVE ON TWITTER 

> DOCUMENTATION + EXAMPLES

=====

> STACKOVERFLOW 

QUESTIONS

WILL IT REPLACE NATIVE (SWIFT) - NO

WAS IT WORTH - YES

DOES REACT HAS ITS USER GROUP - YES

DO I WANT TO CODE MORE IN REACT - HELL YEAH !!!

↑ CONTACT ME 

THANKS

Q&A



- > [GITHUB.COM/KOSTIAKOVAL/PRESENTATIONS](https://github.com/kostiaoval/presentations) "REACT_NATIVE"
- > [@KOSTIAKOVAL](https://twitter.com/kostiaoval)
- > [KOSTIAKOVAL.GITHUB.IO](https://kostiaoval.github.io)