

Flutter VS React Native

A comparison study on Hybrid technologies for secure mobility solution



www.optisolbusiness.com

© 2020 OptiSol Business Solutions Private Limited. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose without the written permission of OptiSol Business Solutions. Any hard copies of this document are to be regarded as temporary reference copies only.

TABLE OF CONTENTS


About the Author	4
Glimpse of Cross Platform	5
Overview of Flutter and React Native	6
Flutter Vs. React Native	7
<i>Programming Languages</i>	7
<i>Stability</i>	8
<i>Community Support</i>	9
<i>Performance</i>	10
<i>Documentation</i>	11
<i>UI Components</i>	12
<i>Build & Release Automation</i>	13
Conclusion	14



▶ ABOUT THE AUTHOR








Sathish Kumar | Technology Manager

 [linkedin.com/in/sathish-chennai](https://www.linkedin.com/in/sathish-chennai)

Broad experience over 10+ years of software development and leadership with mobility solution as passion. He has been working with OptiSol for around 9+ years as Technology Manager – Mobility practice. He can quickly design and implement solutions to bring value to any Enterprise and startup organization.

Area of Excellence

-  **Mobile:** Objective-C, Swift (native iPhone/iPad/ iwatch), Hybrid applications (React Native / Cordava) Cocoa Touch, cocos2d, Social Media Connect APIs, Payment Integration SDKs, iAds
-  **Analytics and performance monitoring:** Google Analytics, Flury, Crashlytics, Instabugs, New Relic, Appsflyer and Mixpanel.
-  **Cloud Services:** Amazon (S3 storage / Lambda scripts / Ec2 deployments), Wowza Live streaming (GoCoder / Wowza Cloud / Stream Engine). GCP (Push Notification / Storage / Database)
-  **Continues Integration:** Jenkins applied with (iTunes / Fabric / Hockey) distributions.
-  **Web:** Core Java, J2EE, Spring, Struts, JSF, Hibernate, Play, Microservice architecture.

GLIMPSE OF CROSS PLATFORM

What is Cross Platform Mobile Development?

Cross-platform mobile development refers to the development of mobile apps that can be used on multiple mobile platforms. It can involve a company, developing original app in a singular environment for development that will then allow the app to be sent to many different native platforms.

Why Cross Platform Mobile App Development?

Cross platform mobile development has multiple advantages for a developer. First of all, it allows you to reuse a significant account of code, meaning there's less time and effort to put in. This makes the production both cheaper and faster. For a business, less time spent in development also equals to an earlier release and, in turn, quicker profits. In addition to that, releasing one cross platform app instead of multiple native apps lets them save on marketing.





OVERVIEW OF FLUTTER & REACT NATIVE

Flutter and React Native are both cross-platform mobile development frameworks. React Native was developed by Facebook and was initially released in May 2015. It was later discovered by a larger community and ever since, it has become mainly community-driven. Flutter, on the other hand, is considered as a newer framework; its first stable version was released in December 2018. Apps using React Native are written in JavaScript, while apps using Flutter are written in Dart. Flutter became a stable framework compared with React Native and most of the recent apps chose Flutter framework competing with performance.

Title	Flutter	React Native
First Release	2017	2015
Created By	Google	Facebook
Technology	Dart	Java Script
Time to Market	Fast	Slower than Flutter
Documentation	Smaller but clear and precise	Extensive but bit clumsy
Hot Reload	Supporting	Supporting
IDE Support	VS Code, Android Studio, IntelliJ IDEA	Almost every IDE possible
Native Performance	Great	Good
Used By	Google, Google Ads, Alibaba, Tencent, Hamilton Musical, JD Finance	Facebook, Instagram, Uber

Let's take a look at the two competitors in this detailed overview to decide which one is a potentially better tool to develop a cross-platform application.

FLUTTER VS. REACT NATIVE: PROGRAMMING LANGUAGES

Flutter

Uses Dart-Dart is fairly new. It is a language not a script, although learning curve for this language is moderate especially for experienced developers familiar with C# and Java can adopt easily. I understand that when compared with JavaScript, it requires developer to know compiler and structural language. But that's not without acknowledging fact that it's very easy to get started with Flutter's demo app.

React Native

It works on JavaScript. It is a widely adopted language by the developers, with a lot of support from the online community and there are many apps that are able to identify in market in terms of mobile and web applications.





FLUTTER VS. REACT NATIVE: STABILITY

Flutter

The Beta 2 version arrived with so much of improvements and rich features. With the developer community also growing strong, Flutter will only evolve to become more stable for deployment soon. People are starting to migrate to develop the app with this framework since it has authorized community center to render the tools required to develop the expected app modules.

React Native

Backed by an already popular and established framework, React Native is reliable. Despite the dependency on 3rd party tools, React Native still is the more stable of the two.



FLUTTER VS. REACT NATIVE: COMMUNITY SUPPORT

The developer community is one of the crucial factors to consider here, as developers can easily adopt and share knowledge to perform expected results. Learning framework improves as more community support established.

Flutter

The Flutter community has been growing dynamically. There are frequent meetups, conferences, and events discussions happening in their connected community.

React Native

It has an edge when it comes to community support. It is older than Flutter (established in 2015) and there are tons of events and meetups happening across the world. GitHub has a React Native developer community and offline events are also organized by the community.



“

React Native has good community support compared with flutter since it has been a long in battle, Flutter community rapidly growing with high end of organized support to the developer.

”

FLUTTER VS. REACT NATIVE: PERFORMANCE



Flutter

It uses the Dart framework which has most of the components in-built, hence it's bigger in size and often does not require the bridge to communicate with the native modules. In short, Flutter has everything needed for app development in the Flutter engine itself.

React Native

Its architecture heavily relies on JS runtime environment architecture, also known as JavaScript bridge. The JavaScript code is compiled into native code at runtime. In short, React Native uses the JavaScript bridge to communicate with the native modules.

“

Flutter engine has most of the native components in the framework itself and it doesn't always need a bridge to communicate with the native components. React Native, however, uses the JavaScript bridge to communicate with native modules, which results in poor performance.

”



FLUTTER VS. REACT NATIVE: DOCUMENTATION

The process of setting up the developer machine to use the new framework takes time. It requires lot of configuration of software installations. The technology should have proper documentation to get users up and running.

Flutter

The getting started guide for Flutter has detailed information on IDE setup and platform setup for both iOS and Android. You can read all the required setup details on Flutter installation for macOS [here](#). On top of this, Flutter has a CLI tool called flutter doctor which can guide developers through the setup. There is a separate page on how to configure the editors to get going with Flutter. Once all the setup is done, we can create and run new Flutter app from CLI.

```
$ flutter create MyProject  
$ cd MyProject  
$ flutter run
```

React Native

The documentation directly jumps to the step of creating a new project. A new React Native project can be created and run on iOS simulator using commands. There is no setup guide for Android projects in the React Native document.

```
$ react-native init MyProject  
$ cd MyProject  
$ react-native run-ios
```

“

Flutter offers better documentation and CLI support for setup and configuration.

”

FLUTTER VS. REACT NATIVE



UI COMPONENTS

Flutter - Rich in Components

Flutter framework is bundled with UI rendering components, device API access, navigation, testing, stateful management and loads of libraries. This rich set of components removes the need to use third-party libraries.

React Native - Less Components

Core React Native framework provide UI rendering and device access APIs alone. In order to access most of the native modules, it has to rely on 3rd party libraries. React Native is too much dependent on 3rd party libraries.

“

Flutter rich in development APIs and UI components while React Native is too much dependent on 3rd party libraries.

”

FLUTTER VS. REACT NATIVE: BUILD & RELEASE AUTOMATION

Flutter

Flutter documentation for building and releasing Android and iOS apps. On top of this, Flutter has officially documented the deployment process with *fastlane* here.

React Native

The React Native official documentation doesn't have any automated steps to deploy the iOS apps to App Store. However, it provides a manual process for deploying the app from Xcode. The process of using fastlane to ship React Native apps is described in this article. This means that React Native has to rely on third-party libraries for build and release automation.



“

Flutter has a great build automation tooling and can be used to deploy apps from the command line. React Native apps lack support for the CLI tools that are officially supported for build automation.

”

CONCLUSION

Having worked with both Flutter and React Native, we can say that even though Flutter is a younger framework, it does not lag behind on React Native. Both frameworks have their advantages: Weighing the result, Flutter would gain the upper hand but React Native is also an alternative.



If having a consistent UI for iOS and Android is important, Flutter could be the better choice.

Flutter came out as the winner in this match. Most of the industry experts have predicted that Flutter is the future of mobile app development. Considering the comparison above, it's clear that Flutter has entered the cross-platform mobile development race very strongly.



OptiSol - an IT services firm offering DIGITAL solutions for Startups and Enterprises through custom web and mobile application design & development, Artificial Intelligence and product implementations. With presence over a decade along with 200+ smart workforce in India and global offices at US, Ireland, Sweden, Dubai and UK. Our development methodology is driven by design thinking with tools for collaboration and our key differentiator is our People.



NASSCOM



OptiSol Business Solutions Private Limited

Baid Hi Tech Park, 5th Floor, Block # 38, East Coast Road,
Thiruvanmiyur, Chennai – 41

+91-44-24512206 (India) | +1415-233-4737, +1908-838-0191 (USA)

www.optisolbusiness.com | info@optisolbusiness.com