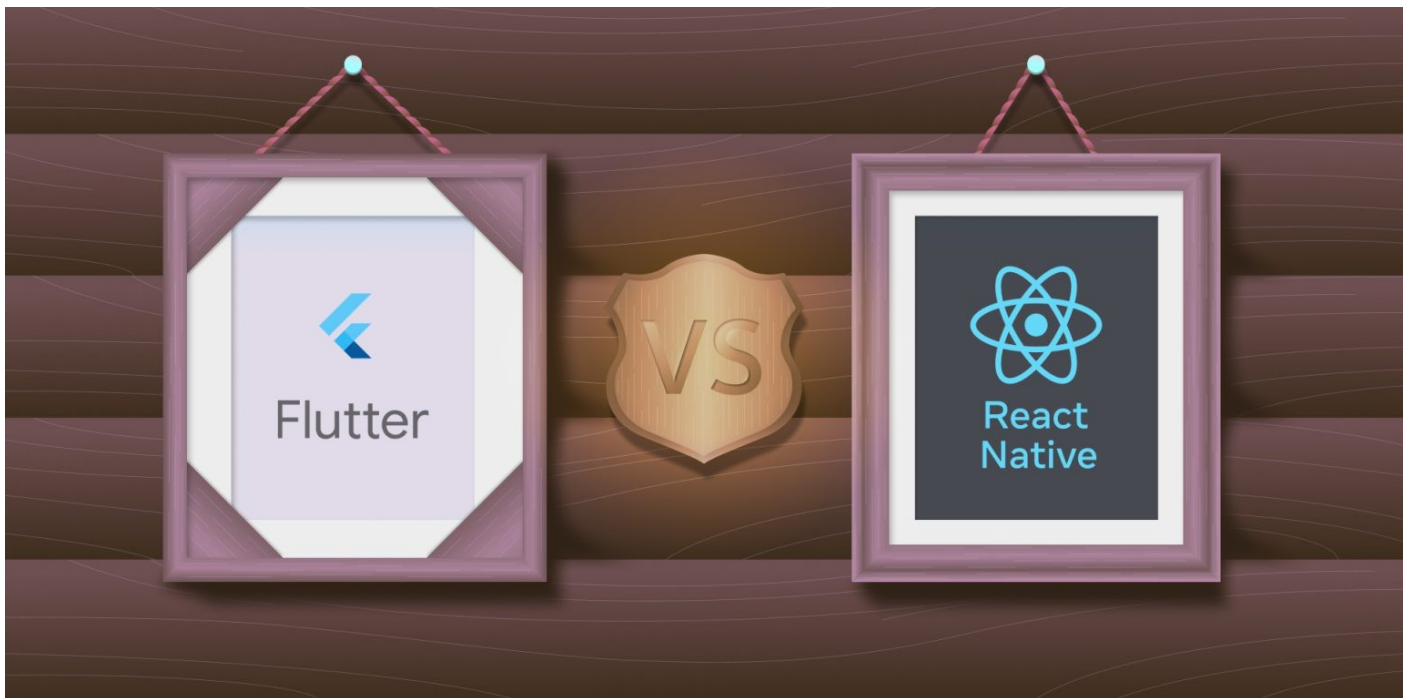


# Flutter vs. React Native: A Comparison of Cross-Platform Frameworks

MARCH 14, 2024 · CLARITY STAFF



If you are a mobile app developer, you might have faced the dilemma of choosing between Flutter and React Native for your **cross-platform app development**. Both frameworks are backed by tech giants (Google and Facebook, respectively) and have gained popularity and traction in recent years.

But which one is **better** for your project? What are the **pros and cons** of each framework? In this blog post we will help you compare Flutter vs. React Native in terms of:

- Architecture
- Performance
- UI Components
- Development Experience Community and Support

Microsoft

Product ▾

Pricing

Resources ▾

Sign  
inGet  
started

Clarity  
Blog

we will answer these questions and help you make an informed decision about which tool to use for your upcoming app.

## Architecture



Flutter is a **cross-platform UI framework** that allows you to create beautiful and fast applications for multiple platforms using a single codebase. Flutter was first released in May 2017 and has since grown steadily in popularity. Flutter uses a programming language called **Dart**, which is a modern, object-oriented, and optionally typed language that compiles to native code or JavaScript.

Flutter has its **own rendering engine called Skia**, which draws every pixel on the screen. This means that Flutter does not rely on the native platform widgets or components, but rather creates its **UI elements called widgets**. Widgets are the building blocks of Flutter apps, and they can be composed, customized, and animated to create rich and responsive user interfaces. Flutter also provides a **hot reload and hot restart feature**, which enables you to see the changes in your code instantly without losing the app state or restarting the app. This makes development faster and more enjoyable. Flutter also supports **stateful hot reload**, which preserves the app state even when you change the app logic.

Flutter 3 introduced some major changes and improvements to the framework. For example, you can now use Flutter to create applications for web, desktop (Windows, Mac, Linux), and embedded devices, in addition to iOS and Android. Flutter also improved its support for **Fir** ▼ ifificantly.

Microsoft

Product ▾

Pricing

Resources ▾

Sign  
inGet  
startedClarity  
Blog

has since become one of the most popular frameworks for mobile app development. React Native uses the same principles and concepts as React, a popular library for building user interfaces for web applications.

React Native uses a JavaScript bridge to communicate with the native platform modules and components. This means that React Native does not create its own UI elements, but rather uses the native platform widgets or components that are exposed to JavaScript. React Native also supports hot reloading and live reloading, which enable you to see the changes in your code without restarting the app or losing the app state.

## Performance



Flutter has a performance advantage over React Native, because it does not need a bridge to communicate with the native platform. This reduces the overhead and latency of the communication, and allows Flutter to achieve smoother animations and transitions. Flutter also has a hot reload and hot restart feature, which enables faster development and testing cycles.

React Native has a performance disadvantage compared to Flutter, because it needs a bridge to communicate with the native platform. This increases the overhead and latency of the communication, and may cause some performance issues or bottlenecks. React Native also has a hot reload and hot refresh feature, but it is not as fast or reliable as Flutter's.



Flutter has a rich set of UI components, called widgets, that are highly customizable and expressive. Flutter widgets are based on the Material Design guidelines for Android, and the Cupertino design guidelines for iOS. Flutter also supports creating custom widgets, as well as using third-party widgets from pub.dev or other sources.

React Native has a basic set of UI components, called components, that are mostly based on the native platform components. React Native components are not as customizable or expressive as Flutter widgets, and may not always match the look and feel of the native platform. React Native also supports creating custom components, as well as using third-party components from npm or other sources.

## Development Experience

Flutter has a better development experience than React Native, because it offers more tools and features to help developers create high-quality apps. Some of these tools and features include:

- Dart DevTools: A suite of debugging and performance tools that integrate with IDEs like VS Code and Android Studio.
- Flutter Inspector: A tool that allows developers to inspect and modify the widget tree and layout of their app.
- Flutter Outline: A tool that shows the structure and hierarchy of the widgets in the code editor.
- Flutter Test: A framework that supports unit testing, widget testing, and integration testing of Flutter apps.
- Flutter Driver: A tool that supports end-to-end testing of Flutter apps on real devices or simulators.

React Native has a worse development experience than Flutter, because it lacks some of the tools and features that Flutter offers. Some of these tools and features include:

- React DevTools: A suite of debugging and performance tools that integrate with browsers like Chrome and Firefox.
- React Native Debugger: A standalone app that combines React DevTools with other tools like Redux DevTools and Reactotron.



## Community and Support



Flutter has a smaller but growing community and support than React Native, because it is a newer framework that was released in 2017. However, Flutter has gained a lot of popularity and adoption in recent years, especially among startups and indie developers. Some of the notable apps that use Flutter include:

- Google Pay
- Alibaba
- eBay
- BMW
- Nubank

React Native has a larger but more fragmented community and support than Flutter, because it is an older framework that was released in 2015. However, React Native has also faced some challenges and controversies in recent years, such as the re-architecture project, ▾ ing

Microsoft

Clarity  
Blog

Product ▾

Pricing

Resources ▾

Sign  
inGet  
started

– Facebook

– Instagram

– Airbnb

– Uber

– Skype

## Device Compatibility

Flutter and React Native both support Android devices with ARM processors and Apple devices from iPhone 4s and onward. Both also support Android and iOS simulators for developing apps.

Apple devices require iOS 9 and above for Flutter apps, and iOS 11 and above for React Native apps. Android devices require API level 19 (Android 4.4) and above for Flutter apps, and API level 23 (Android 6.0) for React Native apps.

The minimum supported version for Expo-based React Native apps is Android 5 or iOS 10 and above.

## Which framework should I use?

Well, at first you would need to examine your or your team's existing skill set, the goals and requirements of your project.

If you already know JavaScript, writing mobile apps in React Native will be easier to develop and maintain. However, if you're looking for better performance, stability, and a more cohesive environment between ecosystems, you should consider giving Flutter a try.

# Stop guessing. Get Clarity.

Free forever. Built to grow with your business. No limits on traffic.

[Sign up with Clarity](#)