Pratibodh
A Journal for Engineering
A free and Open Access Journal
Homepage: https://pratibodh.org

Comparison study of React Native Versus Flutter

Chintan Grover, Opal Jain, Aman Kaushik, Dr. Vinita Mathur
Department of Artificial Intelligence & Data Science, Department of Artificial Intelligence & Data Science Jaipur Engineering College & Research Centre, Jaipur Engineering College & Research Centre
[1]chinxgrover@gmail.com, [2]opal8029@gmail.com [3]kaushikaman027@gmail.com, [4]vinitamathurai@jecrc.ac.in

Abstract

In the mobile development market, iOS and Android emerged as pivotal players. iOS, powered by Swift and Objective-C, is celebrated for its exceptional user experience, while Android, utilizing Kotlin and Java has a very large user base. Due to the presence of these two OS, the mobile apps are to be developed separately for both platforms. Cross-platform frameworks allow developers to build applications for both iOS and Android using the same code, increasing efficiency and decreasing the company's effort and cost. This research conducts a comparative analysis of React Native and Flutter, two widely used cross-platform frameworks. It delves into the architecture, programming languages, and their impact on developer productivity. The research paper compares the two platforms on UI components, compatibility, and community growth. The research provides concise yet insightful guidance for developers, highlighting both the advantages and disadvantages of cross-platform frameworks. It aims to aid developers in exploring the complexities of modern cross-platform app development.

Keywords: Flutter, React Native, Cross Platforms, Android, iOS, Programming

Article Status

Available online :

1. Introduction

In the contemporary era, mobile applications have become indispensable, facilitating connectivity and convenience in daily life. As the reliance on smartphones continues to grow, the demand for efficient and cost-effective app development solutions has become paramount. Enter cross-platform frameworks, such as React Native and Flutter, offering a transformative solution by allowing developers to use a single codebase for both iOS and Android. This not only streamlines the development process but also significantly reduces the associated costs and efforts. In this context, a careful exploration of React Native and Flutter becomes crucial.

React Native, championed by Facebook and launched in 2015, empowers developers with the flexibility of using familiar languages like JavaScript and TypeScript. Renowned for its efficiency in promoting code reuse, React Native facilitates a swift and cost-effective development workflow. Additionally, its thriving community contributes to an ever-expanding ecosystem of resources and support.

Flutter, developed by Google and introduced in 2017, takes a unique approach with the utilization of Dart. Renowned for creating visually appealing, natively compiled applications, Flutter stands out with its expressive UI and a comprehensive widget library. The hot reload feature further accelerates development cycles, enhancing overall productivity.

This research undertakes a comparative analysis of React Native and Flutter, exploring their architectural intricacies, programming languages, and respective impacts on developer productivity. The study aims to provide a clear understanding of the advantages and disadvantages of these cross-platform frameworks, allowing developers to make informed decisions before starting the whole development process.

2.1 React Native:

React Native adopts a JavaScript-centric architecture. The core of the framework operates in a separate thread from the native components, allowing asynchronous communication. The JavaScript thread handles the logic and UI rendering, while the native modules communicate with the device's native APIs. The "Bridge" acts as a communication channel between the JavaScript thread and native modules. This bridge involving architecture produced lag in the applications as all the computations were handled asynchronously.

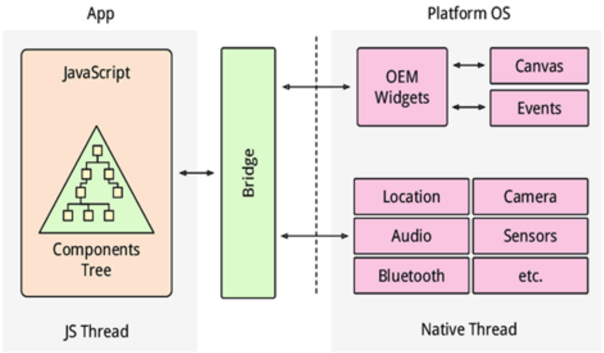

Figure:1 – Old Architecture of React Native

The new architecture (available from React Native 0.68 version) has dropped the idea of bridge and introduced JavaScript Interface (JSI), this will provide a direct and native interface to JS objects. As a result of this, now synchronous execution of functions can take place, moreover, JS can invoke functions concurrently.

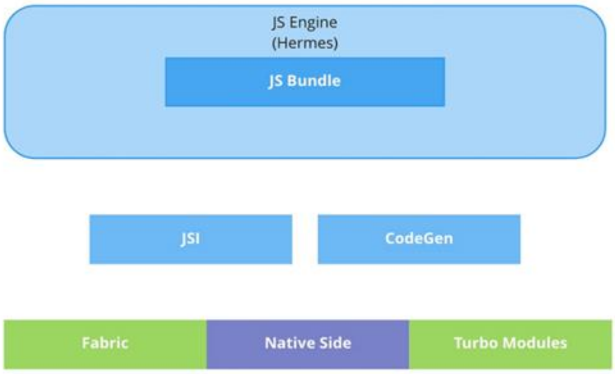

Figure:2 – New Architecture of React Native

2.2 Flutter:

Flutter, on the other hand, embraces a layered architecture where everything, including the UI components and rendering engine, is written in Dart. The framework doesn't rely on a bridge; instead, it uses a compiled programming language (Dart) to communicate directly with the device's hardware. This results in faster performance. Flutter's unique architecture, known as the "Layered Architecture," allows for the creation of expressive and customizable UIs.

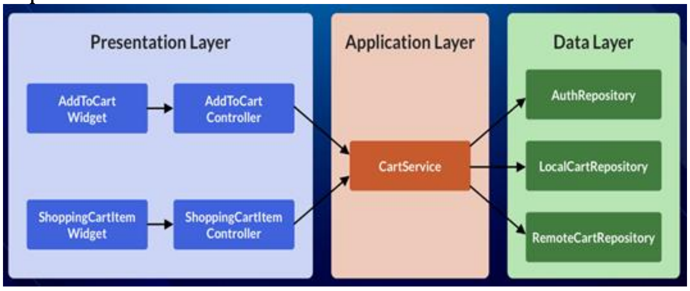

Figure:3 – Architecture of Flutter

3.Evaluating on the basis of developer productivity

Both React Native and Flutter prioritize developer productivity, offering features like hot reload that enables developers to instantly view changes without rebuilding the entire application. This helps to accelerate development cycles. React Native's advantage lies in its use of familiar web technologies (JavaScript), making it accessible to a broader developer base. Flutter, with its focus on consistent UI and use of Dart, provides a streamlined development experience. With a head start in the market, React Native has a substantial user base and an active community. The community contributes to the framework's growth, providing support, sharing resources, and creating a diverse range of plugins and packages. The framework has a robust ecosystem of third-party libraries, making it easier for developers to find solutions to common challenges. On the other hand, Flutter is not fully mature yet as it has a smaller collection of libraries and packages. Although its user base is growing rapidly; sometimes very few options are available to the developer.

The choice between them often depends on the developer's background, project requirements, and the specific features each framework brings to the table. Ultimately, both frameworks aim to enhance developer efficiency in the realm of cross-platform app development.

Both React Native and Flutter prioritize developer productivity, offering features like hot reload that enable developers to instantly view changes without building the app again.

4. Performance analysis of Flutter and React Native

The architecture of Flutter is that it directly translates into native code while in the case of react native, we know that it uses JavaScript Interface (JSI) hence there are some performance issues with React Native. In the case of flutter, Its architecture helps it to start up apps faster and it has fewer performance issues.

Another parameter to discuss would be the App's size. It is observed that Flutter apps are larger than their corresponding React Native apps. This is because Flutter apps are compiled ahead of time but React Native apps are compiled at runtime only and only the part of the package/library which is necessary is included in the bundle.

In conclusion, we can say that Flutter apps than React native apps are much faster as they compile the code directly to C, which is closer to native. However, the react native's new architecture has increased its efficiency but still, this point goes to flutter.

5. Community growth

React Native has been in the market for a longer time than Flutter so it has a larger user base in comparison with Flutter. We have mentioned this earlier that React Native has a larger set of libraries for developers. As of now, the demand for a React Native Developer is more in the market as compared to a Flutter developer

.

The probable reason of React Native's popularity is JavaScript which has been one of the widely used languages from over a decade.

But on social media platforms like Reddit and Twitter, the story is different as Flutter has more followers. Also, it has more starts on GitHub. This shows that Flutter has a very fast-growing community and in the future, it might take over react native's popularity. There is a Google Trends chart showing the results of searches made on the web regarding both platforms.

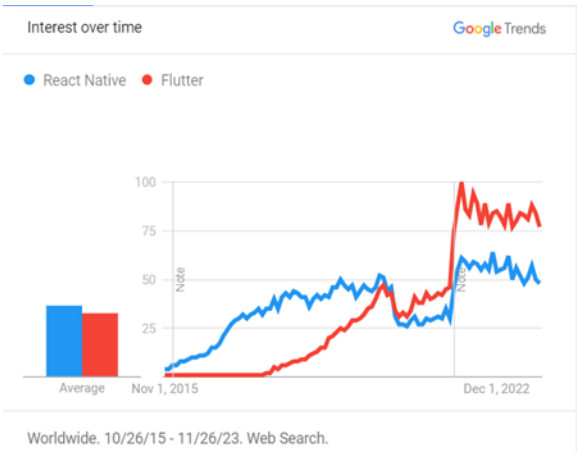

Figure:4 –Google Trends graph

Some popular applications which were created using Flutter are :- eBay, Google Ads, Baidu, Alibaba, Dream11, Philips Hue, ByteDance Apps and many more.
Some popular applications which were created using React Native are :- Facebook, Instagram, Tesla, Walmart, Airbnb, Discord, Wix, Myntra, Bloomberg, UberEats, Ads Manager, Skype, Pinterest and many more.

6. Advantages and Disadvantages

In this section, we will try to cover pros and cons of flutter and react native separately and then conclude this paper. Starting with Flutter first.

6.1 Advantages of Flutter:

•Offers hot reloading i.e. the application refreshes automatically on code changes.
•It has faster app start time than React Native.
•It is one of the most popular frameworks with a fast growing community.
•Flutter comes with a rich set of customizable widgets that allows developers to create complex UIs easily.
•There is no bridge connecting to native modules as seen in the case of React Native, therefore it’s applications are faster and better performing.

6.2 Disadvantages of Flutter:

•Though it’s community is growing actively, but it does not have a mature developer base. So there are limited no. of third party libraries available for developers.
•Flutter uses Dart as programming language, which is not as popular and widely used as its counterpart (in this study) JavaScript. So developers may face some issues while learning it.
•Dart Flutter apps usually have large sizes as compared to React Native because of it dependencies.
•It performs poorly when a lot of animations are used in the application and the frame rate drops drastically.

6.3 Advantages of React Native:

•Just like Flutter, it also has hot reloading feature that allows developers to view the results of code changes immediately.
•React Native has a larger community of developers than Flutter. This means than more plugins and third party libraries are available for developers to use,
•It uses JavaScript language, which has a very large user base, so it’s comparatively easier for developers to learn this framework.
•The bundle size created with React native is generally smaller than that of Flutter.
•It is able to handle animation computations in a better way than Flutter and offers better frame rates.

6.4 Disadvantages of React Native:

•Though mature, but it still has some performance issues and glitches that needs to be resolved even after the introduction of JavaScript Interface (JSI).
•Its popularity is somewhat constant due to the emergence of other cross platform frameworks.
•Performance of React Native application is slow as they use an interface between JavaScript and native components.
•Implementing some functionalities may require native app developers as full control over native components is not given.

Conclusion:

We can surely say that cross platform frameworks such as React Native and Flutter are fast and very cost effective way of developing applications for company. Both Flutter and React Native are quite popular and has rich documentation resources to learn from. This study does not declare a winner between the two, but explores both the frameworks and highlight out their pros and cons in different ways.
We can conclude that one should use Flutter when the application involves heavy computations or if the developer is not already familiar with JavaScript. It is good for making MVPs (Most viable product). We should use React Native when we wish to develop web app as well and our application does not have heavy computations. We should also consider choosing it in case of applications dealing with Bluetooth usage and custom communication.
We can also conclude that React Native has large user base currently in the market but Flutter’s popularity is growing at a faster rate and might surpass React Native someday.

References and notes:

1.Sharjeel Morjab Khan, Comparitive Analysis of Flutter and React Native,29/08/2022.
http://ijaims.smiu.edu.pk/ijaims/index.php/AIMS/article/view/19/8
2.Pete Peranzo,: Flutter vs. React Native: Pros and Cons, Imaginovation, 05/11/2023. https://imaginovation.net/blog/flutter-vs-react-native-features-pros-cons-outlook/
3.MindK, Pros and cons of React Native for cross-platform app development, https://www.mindk.com/blog/react-native-pros-and-cons/
4.Ekrem Gulcuoglu, Comparision of Flutter and React Native Platforms, 31/12/2021.
https://dergipark.org.tr/en/pub/iuyd/issue/67614/888243
5.Andrew Baisden, React Native vs. Flutter, Log Rocket 14/06/2022.
https://blog.logrocket.com/react-native-vs-flutter/

6.Trung Tran, Flutter vs. React Native: Performance Rundown, Orient Software, 23/11/2021. https://www.orientsoftware.com/blog/flutter-vs-react-native-performance/