Application Development using Flutter and React Native: Cross-Platform Development

Chakradhar Avinash Devarapalli

Software Developer
Email: *avinashd7[at]gmail.com*

Abstract: ***Mobile application development has been in trend since the launch of smartphones. Most of the businesses are now handled on mobile phones. Even Google has made it mandatory to build mobile-friendly websites. Mobile application developers need to develop apps using the most suitable platform for better results and customer use. Here we have compared the most trending mobile application development platforms that can be used to write a single codebase and deploy it on both Android and IOS. This paper also delves into the reusability and performance considerations of both platforms. The future developments of both these cross-platforms have also been discussed by considering the most emerging technologies. The research aims to provide enough information to the developers to understand both these platforms, make better decisions, and produce efficient mobile applications. This paper provides a great resource to all the developers and stakeholders to utilize both these cross platforms and produce the best mobile application projects.***

Keywords: Mobile platforms, Cross-platform Framework, Mobile App Development, Flutter, React Native

1. Introduction

A vast trend worldwide has been seen that consumers spend a lot of money to buy different mobile. This trend clarifies the usage of mobile devices in daily life. Every year spending money on mobile apps is increasing continuously. Mobile apps are a new way of interacting with users on mobile devices rather than desktop sites. The statistics show that site visiting requests on mobile devices are increasing and their reach is about 50% as compared to desktop sites [1].

Different Operating systems are used by different manufacturers of the mobile device. IOS and Android are two widely used operating systems in mobile devices. Statistics show that a quarter of mobile users use Android and roughly 23% hold IOS mobile devices [2]. Any mobile application that has a diverse audience should be published on Android and IOS. The data demonstrate that 97% of mobile use these operating systems which means a vast audience is targeted. Android and IOS have their own distributed platforms, those are Playstore and AppStore Respectively. Many companies prefer native applications over cross-platform applications due to their background services which can be utilized in native applications more efficiently. Native application cost is also another bottleneck that diverts the companies to Cross-Platform Framework. Product-based applications are very easy to maintain and develop using one codebase using cross-platform like React Native and Flutter.

With the emergence of the cross-platform application, native applications are underrated to adopt by different companies, but not obsolete. There are still many areas where only native applications are applicable. Cross-platform applications combine both primary operating systems using one codebase which is easy to develop and maintain.

In 2015, React native was launched in the mobile industry which revolutionized the application using a single codebase idea and usage of the prior most adopted JavaScript Language. It comes with a new idea which is supported by the React Framework of Web Development backed by Facebook. It uses the architecture which JSI (Java Script Interface) which connects React Framework JavaScript Code and the native Layer of the Operating system to build mobile application code.

In 2017, the launch of the new cross-platform Flutter with its new programming language Dart. Flutter is a software development kit that is backed by Google interestingly it also owns the Android Operating system. It supports Android, IOS, Desktop, Web, and embedded systems which makes it more popular than React Native due to its diversity. Dart Language is a new language but its syntax matches with Java and Koltin which are extensively used in Android development. Flutter emerges with many tech giants as Trustees like eBay, Google ads, and Alibaba Group which enable it to produce a large user community.

2. Literature Review

Mobile phones become significant since the launch of the Smartphone with its extensive and diverse features. But different operating systems need different codebases for applications which is solved by the emergence of 2 famous frameworks Flutter and React Native. The single codebase is revolutionizing the mobile applications [3].

Flutter is cross-platform which encompasses the Android and IOS with extensive flexibility. Flutter utilizes the proprietary widget rather than converting them to native components. Its component is smooth and flexible due to optimized architecture which uses C/C++ language. Flutter provides the hot reload option that only updates the change and keeps the unchanged existing app to build the application rapidly.

Dart is an object-oriented and easy-to-model language that can tackle the frontend and backend of applications as well.

Dart provides the facility of integration of Android and other third-party APIs by different plug-ins. Flutter provides a gutsy User interface and fast response time making it popular among others.

React Native platform is equipped with the React Library which open-source JavaScript framework that is the backbone of any website. It is very popular due to its large community and optimized Document object model which works like tree Data Structures. This framework provides ease to developers who are already familiar with JavaScript, and only Understand the mobile application architecture and it is good to go for mobile development.

React native utilizes JavaScript to combine the various components of the application, which further converts the markup element into the native component which appears like a native application. The native component is appealing and efficient but it takes time to build and sync the code of the application. It uses an approach that keeps its User Interface thread away from the Background JS thread which is responsible for the functionality of the application which yields performance-intensive applications.

3. Benefits of Cross-platform

These cross-platforms provide many benefits that strengthen the application development industry which is why many companies prefer this framework over the native application. It offers a diverse range of benefits. Some of them are listed below:

- **Reduce Cost:** It provides a more cost-efficient solution than a native application for each operating system. The technique is very simple to develop and maintain single code which will be applied to Android and IOS.
- **Widespread:** companies can broadcast the product and service to all large market shares by targeting different platforms using one codebase for the whole application.
- **Rich User Experience:** Client retention is very crucial for any company, its consistent user experience is provided to the audience which will help them to promote and sell their product and service. It is a very useful feature of the cross-platform.
- **Native Features:** Flutter and React Native continuously develop plug-ins that enable native features like Bluetooth, Sensors, Audio, etc. The native feature enhances the user experience of the application.
- **Model Building:** Cross-platform enables the companies to develop the rapid model of an idea and test it on Android and IOS, which helps to take preemptive steps to increase the efficiency and optimization of the application rather than develop it on multiple platforms. This helps the company to use their capital in the right way.

4. Problem Statement

Cross Platform does not bridge with core components that work with the platform for audio and video application program interface. Although it provides an alternative solution which will be discussed in the solution section. Reactive native and flutter do not come up with the file access core component. They use alternative ways to access the file and manipulate the user for required tasks.

Cross platforms have limitations in Update lag and Missing features. When operating systems update with new features these cross-platform come with solutions and take time, due to which companies are unable to use the latest feature as soon as possible. Despite of Update lag, there are many missing features where cross-platform does not work appropriately due to security and its core usage of operating system. Many apps use the core component of the operating system to do a specific job which cross-platform lacks. These problems can be addressed by understanding the Architecture.

5. Architecture

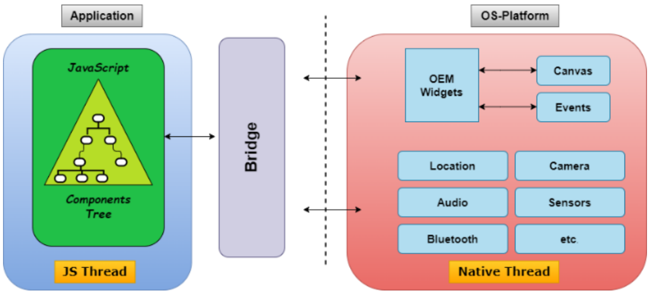

Figure 1: React Native Architecture

React native uses the DOM document object model which works like a tree data structure. The developer writes the code in the JavaScript language and then it splits into two threads one is the main (background) thread and another is the UI thread. The companioned we designed in JavaScript converted into native UI Components using the bridge layer as shown in the above figure. Although the conversion of the JavaScript reacts component is time-consuming it looks like a native component of the platform which is appealing to customers.

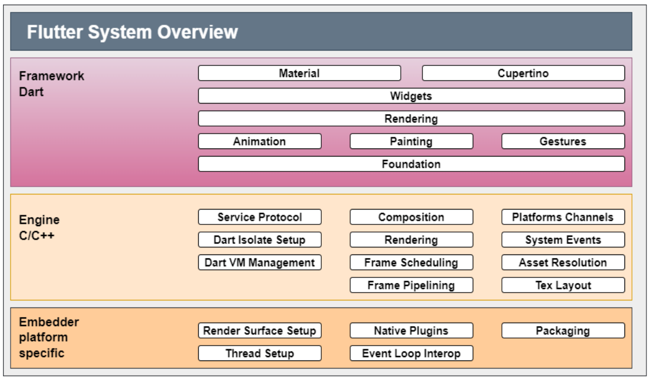

Figure 2: Flutter System Overview

Flutter architecture consists of three tiers; one is the Dart language tier where the developer writes the code then this code bridges with the C++ engine after that it is finally converted into machine code which is executable code for mobile devices. There are many elements involved in these tiers. The developer needs to understand the architecture and compare it with the requirements of the application. This architecture enables them to optimize the app as much as possible. The right use of the component in the right place is very crucial to optimize cross-platform applications.

6. Solutions

Bridging the front end with mobile applications is possible by cross-platform applications. Some applications are discussed earlier. In this section, we come up with solutions. Which is demonstrated in the architectural diagram.

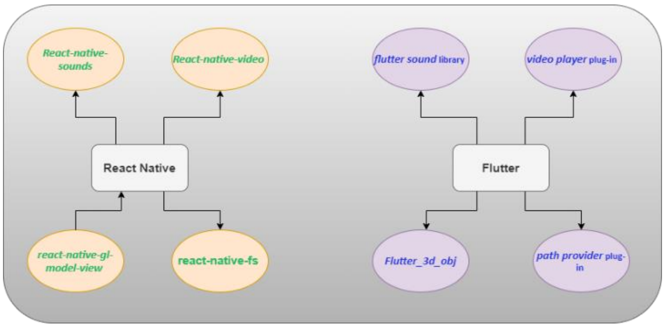

Figure 3: React Native Vs Flutter

React native provides the solution of the *React-native-sounds* library which provides the components for playing sound clips on IOS and Android. *React-native-video* is a library that is utilized for playing and recording the video [4]and[5]. Flutter offers the *flutter sound* library which supports all audio file formats for Android and IOS. *A video player* plug-in is used for displaying video on Flutter widgets[6] and [7].

The react-native-gl-model-view is a library used in React Native. It uses WebGL (Web Graphics Library) for rendering interactive 3d Graphics[8]. *Flutter-3d-obj* is a Flutter library that paves the way to develop around for displaying 3D images by hooking into OpenGL on Platforms because Flutter does not support OpenGL Directly [9].

The react-native-fs is a library of the react native which supports native file system access for Android and IOS. Each operating system needs configuration to grant permission for accessing the file system [10]. Flutter provides a solution for the file system with the help of the *path-provider* plug-in which is commonly used for the location on the primary operating system [11].

7. Research Impact

Given the detailed analysis of cross-platform and how it affects the performance of native applications, the writing allows to equip the developers with this useful technology to reflect efficiency in their systems as well as the process of development. Certain difficulties for the mobile application can only be solved with the use of Cross-Platform. Solutions are suggested for the developers to overcome the hindrances existing to use React Native and Flutter.

8. Future Developments

Cross-platforms like React Native and Flutter hold a very strong potential for the future. These platforms are now moving toward multi-platform expansion where the developers will be able to codebase for web and desktop platforms as well. Flutter has already moved toward this step. Another future development is performance optimization to fill the gap between native and cross-platform application performance. Moreover, these platforms need to be aligned with the latest technologies. With the trend of Virtual Reality (VR) and AR, these cross-platforms need to integrate functionalities with these technologies.

9. Conclusion

React Native and Flutter offer immense solutions with superb performance and uses. Both platforms have reduced the maintenance cost and made it possible to codebase for both Android and IOS. The developer needs to choose the platform by considering the various factors associated with each platform. These cross-platforms will play an important role in the future of mobile application development.

References

[1] gs. StatCounter, "Desktop vs Mobile vs Tablet Market Share Worldwide," [Online]. Available: https://gs.statcounter.com/platform-market-share/desktop-mobile-tablet/worldwide. [Accessed 21 March 2024].

[2] gs.statecounter, "Mobile Operating System Market Share Worldwide," February 2024. [Online]. Available: https://gs.statcounter.com/os-market-share/mobile/worldwide. [Accessed 19 March 2024].

[3] E. Erturk, "2021 Proceedings Book of the 12th Annual CITRENZ Conference," in *CITRENZ Conference*, September 2021.

[4] W. Wu, "React Native vs Flutter, cross-platform mobile," Metropolia University of Applied Sciences, 1 March 2018.

[5] A. Fomushkin, "Practical application of advanced React Native concepts," Metropolia University of Applied Sciences, 30 September 2019.

[6] pu.dev, "flutter_sound package documentation," [Online]. Available: https://pub.dev/documentation/flutter_sound/latest/. [Accessed 20 March 2024].

[7] pub.dev, "video_player package documentation," [Online]. Available: https://pub.dev/documentation/video_player/latest/. [Accessed 20 March 2024].

[8] Fentaw and A. Eshetu, "Cross platform mobile application development : a comparison study of React Native Vs Flutter," University of Jyväskylä, 2020.

[9] M. K. Fahnbulleh and X. SHUOBO, "Examining the Usage of Flutter to Design a Yatch in 3D," *International Journal of Research Studies in Computer Science and Engineering (IJRSCSE)* , vol. 1, no. 1, pp. 1-11, 2021.

[10] H. Brito, Á. Santos, J. Bernardino and A. Gomes, "Mobile development in Swift, Java and React Native: an experimental evaluation in audioguides," IEE, 2019.

[11] P. Tyagi, Building Cross-Platform Mobile Apps for Android, iOS, Web & Desktop, Boca Raton, 13 August 2021.