

# Design and Build of an Application for Android and IOS with the Same Code

<sup>1</sup>Rohit, <sup>2</sup>Dr. Sandeep Sharma

<sup>1</sup>M. Tech Scholar, <sup>2</sup>Associate Professor

Department of Computer Science & Engineering

BRCM College of Engineering & Technology, Bahal, Bhiwani, India

*Abstract - Nowadays application development is at its peak, we can see in the App Store and the play store there are multiple applications there, and everyone wants new updates, and secure applications that should be very helpful for users while they are using applications. This is when users want applications to work according to their requirements. Because every user needs there Are some changes in the application like in-game or in-banking applications. For Android applications, users can get an update in Play Store, and for IOS applications users can get an update in App Store. But there are some differences between Android and IOS applications. While Android Applications are working to improve their segments like changes in filters and many other changes to make user-friendly, IOS applications are doing changes like boosting the speed and making it more comfortable so that users can't face any problems while using the application. Android applications are less in size while IOS applications are big in size. Android applications are less secure than IOS applications while IOS applications are less effective there are very less changes as compared to Android applications.*

**Keywords:** - React Native, Swift, C#, Java, Kotlin, Python

## INTRODUCTION

as we have already discussed that IOS is using different languages and Android is using different languages to make applications because these developers face many issues developing the same application for different OS. Now we introduce the language React Native & Flutter. React Native is introduced by Facebook and Flutter is introduced by Google These are two main languages that help developers to build applications for both Android as well as for IOS with the same code. React Native provides better usability to users and also helps developers to build an application for both, there are many useful features in React Native that other languages haven't.

Main Features of React Native: -

1. Write once and use it everywhere
2. Modular and Intuitive

3. Speed Up Development

4. Live Reload

Main Feature of Flutter: -

1. Hot reload
2. Cross-platform development
3. Native Performance
4. Open-source

As you can see both are open-source languages that help developers to develop better applications for both Android as well as for IOS.

What's the main use and advantage of these languages?

1. Less cost and high work.
2. More effective and useful applications.
3. Less engineer cost because the same developers can make both applications.

React Native is introduced by Facebook and Flutter is introduced by Google These are two main languages that help developers to build applications for both Android as well as for IOS with the same code. With the help of these languages, we find the solution that we don't need two different codes for different OS we can make both applications with the same code. Here are some features of React Native and Flutter There was a difference in codes between Android and IOS

### Libraries of React Native.

- o React Native Paper
- o React Native Elements
- o NativeBase
- o React Native UI Kitten
- o RNUI: React Native UI Library
- o Teaset
- o Shoutem UI
- o Lottie for React Native
- o React Native Maps
- o React Native Gifted Chat

### **The attribute of React Native**

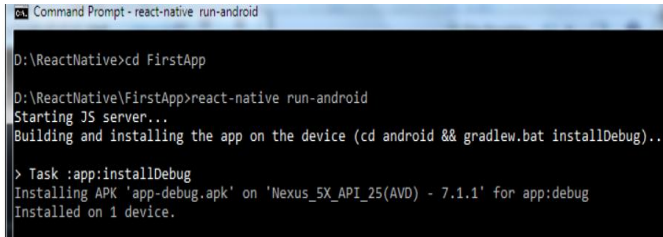
- Write once and use it everywhere.
- UI Focused
- Community Support
- Tried and Trusted

- Support and Third-Party Libraries.

**The attribute of Flutter**

- Data-Object.
- Data-Array
- Identifier
- Entity

The command that is used to create Android applications in react native: -  
 react-native run-android is the command that is used to build Android applications



The command that is used to create IOS applications in react native: -

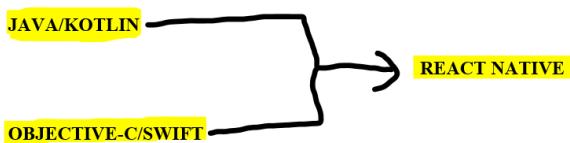
npx react-native init myproject is the command that is used to build IOS applications.

install react-native-mmkv-storage using npm:

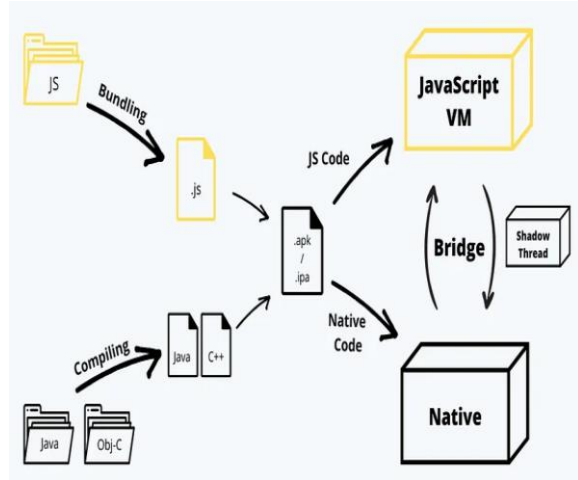
npm install react-native-mmkv-storage install #for iOS only

```
npx react-native run-ios
```

we have already discussed how React Native and Flutter work, and now we will discuss the implementation and we will learn how it works we will see the process. The main reason to introduce React Native is that we have to combine two different languages into one. As we know java and Kotlin are used for Android and Objective-C and Swift are used for IOS. If we can combine these two languages codes, JavaScript code, and Native Code together so we are able to make one code and one application that is usable for both Android as well as for IOS. Native Code can execute directly on devices, while JavaScript needs a virtual machine to run applications. IOS apps have in-built JavaScript and we called JavaScript Core, which is written in C++



They help us to execute the JavaScript code. Android doesn't have any built-in JavaScript Engine on their devices. JavaScriptCore has to run its code along with React Native code to make it specific and useful Java, Object-C, and JavaScript are very different languages, and we know that they are unable to connect to each other, each language has different syntax and its own structure and has different Pros and cons. We have to connect all the languages indirectly to make combined languages so that they can work together and understand the data and the data would be JSON. We are calling this process a Bridge.



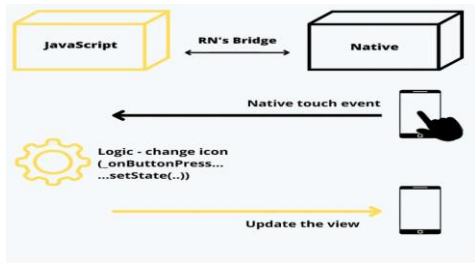
Java, Object-C, and JavaScript are very different languages, and we know that they are unable to connect to each other, each language has different syntax and its own structure and has different Pros and cons. We have to connect all the languages indirectly to make combined languages so that they can work together and understand the data and the data would be JSON. We are calling this process a Bridge.

React Native's Bridge: -

This is the most important part of React Native Architecture Because this allows us to exchange data from one language to another language using JSON Messages. The logic of application development is the same as web development where we have to create front-end and back-end and then we have to combine through API. We are creating a front end with the help of React Native and for Back-end we are using Node.js While we are creating Front-end, we have to use JavaScript also to create the front end with

With the help of JavaScript code, we are able to set the value of the front end, we have to set the color width and screen according to the devices these all are we can do with the help of JavaScript. There are many libraries we must install while creating any application.

We must merge our JavaScript and native code to make any application. But we have to write code



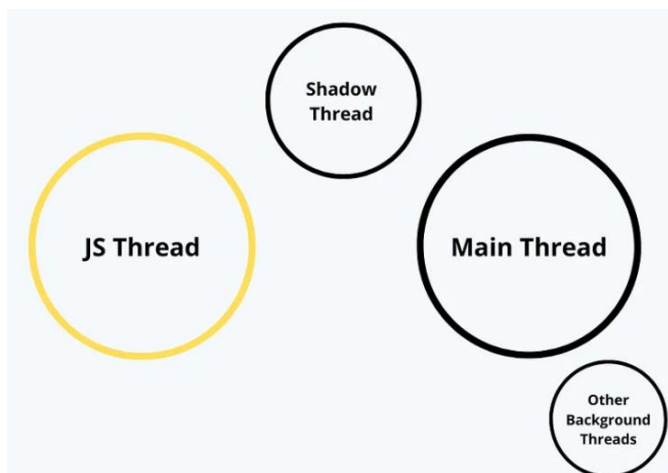
in such a way that both languages must understand their code and they can transfer data easily. When we are navigating from one screen to another screen, we have to use our native libraries to navigate our screen, and to design our screen we are using JavaScript code. As we can see in the below diagram JavaScript message is transferred to the native code via the bridge, if the native wants to share any message to JavaScript, they also have to use Bridge to transfer the message.

As we know JavaScript is working on the screen when JavaScript transfers the message to native, native is showing output according to JavaScript and we know that native is transferring data to JavaScript code, and the same JavaScript is doing.

There are also some issues with the bridge, as it is asynchronous, so it creates problems that can sometimes appear with edge cases. Asynchronous is helping to transfer data very fast as its process is very fast. Sometimes users need some more speed, time asynchronous is not enough and that time synchronous is better.

There are some Threads in React Native: -

When we are running any applications, the device will start three main threads, and there are so many additional background threads. There are many threads that are working internally which means that they depend on languages.



**Main Thread:** -Main Native thread in which applications will be running, on the mobile screen we can see there are multiple screens, they are responsible for the user interface and rendering

the user interface on the mobile screens. This is the thread that is running on fully natively built applications.

**JavaScript Thread:** - This is the logic of the build that will be executed (JavaScript and React code).

**Shadow Thread:** - This is launched with a JavaScript thread called Shadow Thread. The main work of this is to compute the position of the view and set a tree of layout that is coded in JavaScript Thread.

**Native Modules Thread:** - This is used when the application is searching for access to a platform API. It means that when a mobile application needs access to mobile storage to save the data in mobile storage.

**React Native's Bridge:** -

This is the most important part of React Native Architecture Because this allows us to exchange data from one language to another language using JSON Messages.

The logic of application development is the same as web development where we have to create front-end and back-end and then we have to combine through API. We are creating a front end with the help of React Native and for Back-end we are using Node.js While we are creating Front-end, we have to use JavaScript also to create the front end with the help of JavaScript code we are able to set the value of front-end, we have to set the color width and screen according to the devices these all are we can do with the help of JavaScript. There are many libraries we must install while creating any application.

We must merge our JavaScript and native code to make any application. But we have to write code in such a way that both languages must understand their code and they can transfer data easily. When we are navigating from one screen to another screen, we have to use our native libraries to navigate our screen, and to design our screen we are using JavaScript code.

As we can see in the below diagram JavaScript message is transferred to the native code via the bridge, if the native wants to share any message to JavaScript, they also have to use Bridge to transfer the message.

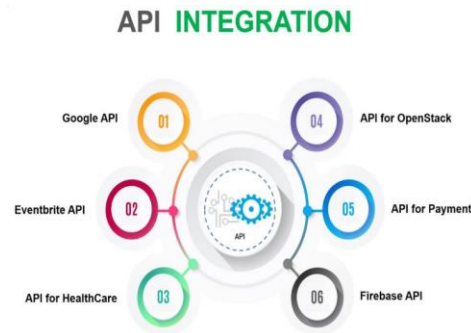
As we know JavaScript is working on the screen when JavaScript transfers the message to native, native is showing output according to JavaScript and we know that native is transferring data to JavaScript code, and the same JavaScript is doing.

There are also some issues with the bridge, as it is asynchronous, so it creates problems that can sometimes appear with edge cases. Asynchronous is helping to transfer data very fast as its process is very fast. Sometimes users need some more speed, time asynchronous is not enough and that time synchronous is better.

As we have written the front end in react native, for the back end we have to use node.js and then we have to combine the front end and the back end to combine the front and back end we are using API. With the help of API, we are able to connect the front end to

the back end, without API we are unable to make any application.

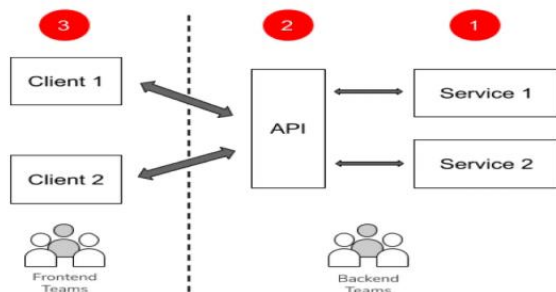
Let's talk about some API integration technology that is helpful for us to build any applications for Android as well as for IOS. We can write code in any language to build any application but without the help of API we are unable to build any application this is a very important point of any application that is connected users directly to the backend.



Let's check the working of the API.



Fig 4.7 API integration use



**CONCLUSION**

In this research, we have tried to find the option to make applications for both Android and IOS with the same code. Earlier we have to write two different codes for Android and IOS to make any application.

Now we introduced React Native and Flutter languages that help us to make an application for both Android and IOS with the same code we haven't any need now to make applications with different many libraries in both coding languages help us to provide flexibility to make application also they are providing the very good interface as compared to another language. They are helping us to provide security also because one language is introduced by Facebook itself (React Native) and another is introduced by Google Itself (Flutter) so there is no issue of security.

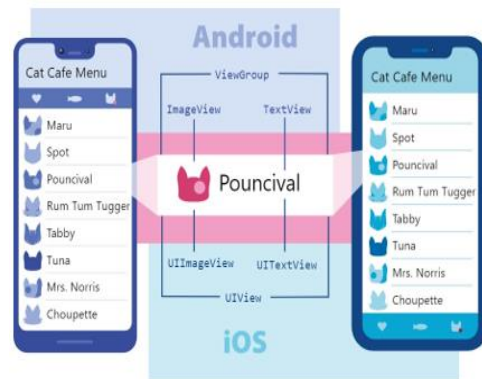
**The attribute of React Native**

- Write once and use it everywhere.

- UI Focused
- Community Support
- Tried and Trusted
- Support and Third-Party Libraries.

**The attribute of Flutter**

- Data-Object.
- Data-Array
- Identifier
- Entity



**REFERENCE**

[1] Richard Kho, "React Native By Example: Native mobile development with React".  
 [2] Vladimir Novick, "React Native – Building Mobile Apps with JavaScript".  
 [3] Emilio Rodriguez Martinez, "React Native Blueprints: Create eight exciting native cross-platform mobile applications with JavaScript".  
 [4] Bonnie Eisenman, "Learning React Native: Building Native Mobile Apps with JavaScript (2nd edition)".  
 [5] Devin Abbott, "Devin Abbott 12/2017 – Revision 1 07/2018 – Revision 5" Fullstack React Native"  
 [6] Nader Dabit 2018 (in progress) "React Native in Action"  
 [7] Eric Masiello 01/2017 "Mastering React Native."  
 [8] Jonathan Lebensold "React Native Cookbook Bringing the Web to Native Platforms."