**What is React Native?**

React Native (also known as RN) is a popular JavaScript-based mobile app framework that lets you build rendered mobile apps for iOS and Android. The framework allows you to create applications for different platforms using the same codebase.

**React Native history**

When Facebook first decided to make its service available on mobile devices, instead of building their own app like many of the leading tech players of the time, they decided to use an HTML5-based mobile web page. However, the solution hasn't stood the test of time, leaving plenty of room for user interface and performance improvements. In fact, in 2012, Mark Zuckerberg admitted that "the biggest mistake we made as a company was that we relied too much on HTML, not native."

Shortly thereafter, in 2013, Facebook developer Jordan Walk made a groundbreaking discovery - he found a way to create UI elements for iOS apps using JavaScript. This sparked a fire, and a special hackathon was created to learn even more how much can be done for mobile development using (so far traditionally web-based) JavaScript solutions.

This is how React Native was born. Originally developed for iOS only, Facebook quickly followed and supported Android before releasing the framework in 2015. Just three years later, React Native was already the second largest project on GitHub in terms of contributors. In 2019, it ranked sixth with over 9,100 entries.

### **React vs. React Native**

React.js is a JavaScript library that helps companies create beautiful UIs. One of the main features of React is that it can run on the client side and render on the server side, and these parts can interact with each other. Therefore, it is widely used to build high-performance web applications and user interfaces.

Whereas React Native is a mobile app development framework that builds mobile apps using React.js. It allows you to create a rich mobile user interface from declarative components. React Native offers features like Hot Reloading to develop apps faster and with less effort. React Native libraries were released by Facebook in 2015, giving the React architecture to Android, iOS and Windows apps.

React.js uses virtual DOM to create better UX. Building the DOM takes time because the pages are big today. But, React.js gets the job done faster using the virtual DOM. This way, React.js uses an abstract copy of the document object model and exposes changes for one component without affecting the rest of the UI. This is what makes React.js one of the best solutions for quickly updating and creating dynamic UIs.

React Native takes this a step earlier. It uses native APIs to render parts of the user interface that can be reused on both iOS and Android platforms. As well as the Java API for rendering Android components and the Objective-C API for writing iOS components. It also uses JavaScript for the rest of the code and customization of the application for each platform. This gives React Native the ability to reuse components and codes.

With React Native Framework, you can render UI for iOS and Android platforms.

It is an open source platform and may be compatible with other platforms such as Windows or tvOS in the near future.

Since React Native components have the appropriate permissions, you can reuse those components to build Android and iOS apps.

You can either include REACT Native components in your existing application code, or reuse Cordova based code using a plugin. However, your existing application must be built using Cordova and Ionic code.

React Native Development is relatively simple, fast, and efficient.

REACT Native is a great choice for developers with JavaScript background, as there is no need to learn Java for Android or Swift for iOS.

React Native is focused on the user interface, which provides faster application loading and smoother experience.

Another reason the REACT Native framework has gained popularity in recent years is that it helps serve both platforms at the same time and end the holy war of iOS or Android development choices for mobile developers. Consequently, giant companies like Facebook, Instagram, Skype, Airbnb, Tesla, Walmart, Baidu Mobile, Bloomberg, UberEATS Vogue and many more have turned their sights on it.

Moreover, using the React Native Framework fills the gap between targeting the broad market and making a profit. With fewer iOS users compared to Android users, but still generating more revenue, many industries are faced with a dilemma: should they focus on making more profit by building iOS apps or building user power with Android apps alone.

**How it works?**

There are two important threads running in every React Native app.

One is the main thread, which also runs in every standard native application. It handles the display of UI elements and handles custom gestures.

Another is specific to React Native. Its job is to execute JavaScript code in a separate JavaScript engine. JavaScript deals with the business logic of an application. It also defines the structure and functionality of the user interface.

These two threads never interact directly and never block each other.

**How do streams interact?**

Between these two threads is the so-called bridge, which is the core of React Native. The bridge has three important characteristics.

Asynchronous. This provides asynchronous communication between threads. This ensures they never block each other.

Rented. It transfers messages from one thread to another in an optimized way.

Serializable. The two threads never use or work with the same data. Instead, they exchange serialized messages.

**What does the development process look like?**

We now understand the basics of React Native architecture. It would be interesting to see what the development process looks like.

First, we open our project with our favorite editor. Let's say we have a component named Greeting that only displays the text that says "Hello!"

A very simple React Native component

To run an iOS app, we need to run react-native run-ios from the command line. This will launch the application on a simulator or on a real device if we have it connected to a computer. The result will be like this:

Our simple iOS app

If we change our minds and want our app to say "Hi!" instead of "Hello!" we can open our editor and change the text. Then in the simulator we can press Command + R as we do when reloading the web page. The change will be immediately visible! Instead of waiting for the build process, which can take a minute or so, we get instant feedback. This makes development pretty fast.

Instant reload like in a web app

To run our Android app, we need to run react-native run-android. To do this, we can completely reuse our Greeting component. This is because the component does not contain platform-specific code. React Native takes care of providing Android TextViews instead of iOS UIViews. Reusability of code is one of the strengths of this technology.

Our Android app that uses the same Welcome component.

**Benefits of React Native**

### We've discussed products built using React Native, so let's move on to the benefits of React Native development and why you should choose it as your solution for building your mobile app.

### - *Code reuse - cross-platform development*

### The reusability of code is the biggest benefit of React Native, and it indicates that apps can run efficiently across multiple platforms, which is really appreciated by CEOs and product owners. They can integrate 90% of the built-in code reuse framework for both operating systems.

### Discord engineers say: “We tried React Native the day it was released for Android. We were surprised at how easy and fast we were able to get our all-inclusive iOS app up and running on Android - it only took two days and it was created! "

### Another great news is that it is possible to use web app code to develop mobile apps as long as they both use React Native. It also speeds up development time as it includes pre-developed components that are included in the open source library.

### - *Large developer community.*

### React Native is an open source JavaScript framework that allows developers to share their knowledge with a development framework that is available to everyone for free.

### If any developer encounters a problem while developing an application, they can turn to the community for support (as of mid-2020, there are about 50,000 active contributors to the React Native tag on Stack Overflow).

### There will always be someone who can help them solve their problems - this will also have a positive effect on improving their programming skills.

### - *Cost effectiveness.*

### Another benefit of React Native development is higher ROI. As mentioned earlier, this is because developers can use the same code to build iOS and Android apps.

### This means you don't need to hire two separate iOS and Android development teams to complete the project; a small group is enough to build it. The cost of developing apps with React Native is much lower than apps built using languages ​​that don't allow cross-platform development.

## *- Quick update.*

## A quick update allows developers to launch an application by updating it to new versions and changing the user interface. Changes are immediately visible and the developer is relieved of the need to rebuild the entire application.

## This has two significant benefits: it saves time — because programmers save compilation time and improve performance — because they don’t lose any state when they make changes to the application.

## *- Simple interface*

## React Native development uses React JavaScript to build the interface of the app, which makes it more responsive and faster with reduced load times, which improves the overall user experience. With its reactive user interface and component-based approach, the framework is ideal for building applications with both simple and complex designs.

## *- Fast apps*

## Some argue that React Native code can negatively impact application performance. Although JavaScript does not run as fast as native code, the difference is imperceptible to the human eye. To further prove this, we decided to test comparing two versions of a simple application written in React Native and Swift - both achieved similar performance results.

## *- For the future*

## Given the speed with which the framework has taken over the market and its simple approach to solving development problems, the future of React Native for cross-platform applications looks bright. Although it has several disadvantages, which we will discuss in the next section, the speed and ease of development makes up for them.

## With all this in mind, let's now take a look at why React Native might potentially not be the right fit for you.

**React Native: Risks and Disadvantages**

Here are four major potential drawbacks you need to be aware of before you decide to develop a React Native app.

*- Lack of some custom modules*

Although React Native has been around for a few years now, some custom modules either leave room for improvement or are completely absent. This means that you may need to run three separate codebases (for React Native, iOS, and Android) instead of one.

However, this is not a common occurrence. Unless you're developing an app from scratch or trying to hack an existing one, you most likely won't run into these issues.

*- Compatibility and debugging issues*

While this may sound unexpected - React Native is used by leading tech players after all - it is still in beta testing. Your developers may encounter various compatibility issues with packages or debugging tools. If your developers are not proficient in React Native, this can negatively impact your development as they spend time on lengthy troubleshooting.

*- Scalability*

In most cases, React Native will work very well for you, even if your application turns into a very complex and complex solution over time. After all, companies like Facebook and Skype have had great success with the framework and have been using it consistently for years. That said, some companies have decided to stop using React Native.

Airbnb, for example, decided to use a framework for its mobile app back when the company was just a budding startup. Over time, however, React Native proved unsuitable for the company's growth plans, and Airbnb resorted to developing two of its own apps. With current advances in RN and the right choice of software architecture, scalability issues can be easily prevented.

- *Need help from native developers*

Remember the "bridge" feature we mentioned earlier in this post? As shown, React Native connects JavaScript with native mobile code. This means that if you take on a developer who does not know native mobile development, it will be difficult for him to include native code in the RN codebase. As a result, you will need the help of Android or iOS developers to guide them through the process. If you are a small company, you might not want to hire a mobile app developer, as it incurs additional costs.

One way to solve this problem is to get a software consultant to lend you a helping hand with native iOS and Android elements.

**React Native is your all-in-one cross-platform development framework**

React Native is a great solution for building apps that run smoothly regardless of the platform or system they're running on. It is a framework that both companies and developers love. This saves a lot of labor time, which speeds up the development process as well as being economical. This gives you access to a wider audience right away - since a single app can be used by both Android and iOS users, the launch can be synchronized, and companies with less development budgets don't have to choose just one platform.

Many fantastic products have been built using React Native, including Skype, Facebook, Pinterest and UberEats, further proving this is a framework worth considering for app development. While the native solution for iOS and Android is still the best option for projects that require a seamless UI and top-notch performance, React Native is a great option if your budget is tight and where a satisfying user experience is sufficient.

React Native is a great option if your project doesn't need a complex interface, access to native functionality (such as media players), and when you only want to develop for one platform. Last but not least, React Native is a good choice if your budget is tight.