Dennis Kiewning Follow
Jan 4 · 5 min read

Researches: React -
All Sides

React Native - The Good, the Bad and the Ugly

A brief summary of the pros and cons of the Facebook solution for native app development and how it compares to already established approaches.

How to understand React Native

React Native is a tool created by Facebook and is used for native app development. To understand the pros and cons of working with React Native, however, it is important to understand the difference between native, web and hybrid apps first. So lets start with a short summary of these known technologies before we dive deeper into the details of the solution recommended by Facebook.

The Good — Native Apps

Most apps on mobile devices are native apps. So when you think about apps, you usually think about native apps. iOS and Android each require a specific programming language in which their native app has to be written. Developers will also use the development tools and interface elements from Apple and Google, to create apps that look and behave as iOS and Android users would expect.

> *Think of native apps as the best case scenario from a user perspective. They offer the fastest, most reliable and most responsive experience.*

this is what we think off with an app

The Bad — Web Apps

So called web apps are the complete opposite of native apps. They aren't even real apps. You don't need to download a web app, they load in browsers like Chrome or Safari and even don't take up any storage on a users device. Developers are not depending on development tools from Google or Apple. Instead, web Apps rely on web technologies that are used to create websites. This allows them to be created very quickly, but they often lack some necessary functionalities of native apps.

> *Web Apps are imitating native apps with web technologies, which almost always results in a bad user experience.*

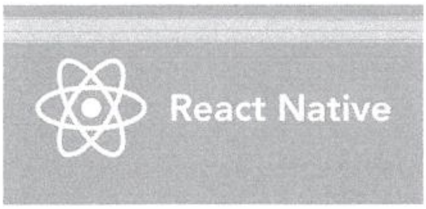
lacking native connections and functionalities

And the Ugly — Hybrid Apps

A hybrid app is the combination of both native and web apps. You install it like a native app, but it actually remains a web app on the inside. And like web apps, hybrid apps are mostly built with standard web technologies, only without lacking the functionality of native apps. They are also a good solution when it comes to cross platform development. So instead of building two different apps for each platform (as previously mentioned native iOS and Android apps require a specific programming language each), developers only have to write one code, which than can be reused for the other.

> *Hybrid apps, like web apps, use web technologies to run in a simplified browser within your app. But even if they don`t lack native functionality, their user experience remains really bad compared to a native solution.*

Cordova for instance: been there done that.
=> Not all to good



So what does React Native do?

React Native combines some of the advantages of native app development with the benefits of cross platform approaches without sacrificing flexibility in platform specific requirements. Meaning:

- React Native renders real native UI elements, making apps look and behave like native iOS and Android apps.
- Developers can reuse their code from iOS for Android or the other way around. With a few exceptions (e.g. supporting platform specific features like 3D Touch for iOS), there is no longer any need to have multiple code bases, one for each platform will do.

Pros using React Native

Learn Once Write Everywhere

The developers of react native follow a "Learn Once Write Everywhere" approach. The goal is to equip developers with a tool that only needs to be learned once and can then be reused to start developing for different platforms immediately. So it's no longer necessary for a developer to learn different programming languages and development concepts for multiple platforms. And if a Developer is already familiar with React (Facebooks approach to improve web development), he only needs to acquire little additional knowledge in order to be able to build mobile apps with React Native.

React Native only requires
a bit more knowledge
over React

Fast, Faster, React Native

A big advantage are the developer tools that React Native brings along. So called real-time reloading in particular: During development, a server is running on the developers machine that communicates with the running app. As soon as the developer saves his code in the editor he is using, it is loaded in the background and sent to the app. In this way, the result of the work is immediately visible and developers receive quick feedback on their changes.

Save Time — Save Money

With these pros and the state of the tools, by using React Native you can actually save some time compared to traditional native development. This is of course marks a critical factor if the budget of a project is limited, but you don't want to compromise on the scope.

Cons using React Native

Who is Your God Now

By using React Native, you are completely dependent on Facebook and have to rely on the fact that with every new software update from Apple or Google, Facebook has to follow up fast by adapting the React Native Framework accordingly within a very short time. And even if its a very unlikely scenario that Facebook will not update their React Native Framework in time (Mainly because the Facebook app itself is built using React Native), its always good to keep that it in mind.

facebook owns it
i rely on them for
updates

React Native can not replace a native Look and Feel

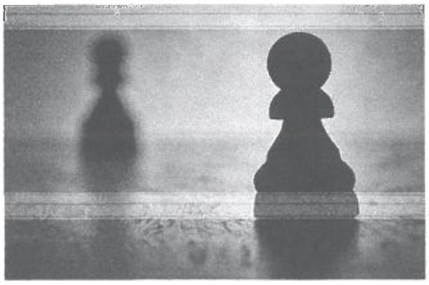
A bigger problem with React Native is the fact, that it does not fully support all native features available within iOS and Android. Therefore, if a developer doesn't want to wait for Facebook to provide special support for the feature he needs, he might still have to stick to native code in some cases. Especially when it comes to navigation components for screen transitions, React Native really lacks supporting the native look and feel of iOS and Android apps.

Unfortunatly not to
bad anymore

Google and Apple have very different Design Guidelines

Sharing one codebase for multiple platforms is one of the greatest benefits of React Native, but also results in a big problem: Google and Apple require different Designs. Human Interface Guideline for iOS, and Material Design for Android, which have a totally different look and feel. So when using the same code, cross-platform apps written with React Native can look very unprofessional compared to tailored native apps.

I will focus on 1
Platform for starters



The Bottom Line

A native iOS and Android solution is still the best choice for projects in which user experience and performance have absolute priority. But if a budget is limited and a reasonable UX and performance still needs to be achieved, then React Native is definitely worth considering. So React Native seems good for apps with simple UI and limited animations, but not for long life or performance-critical apps.

> *Each approach has its pros and cons, but in the end a native solution will always have the greatest benefits and even beats Facebooks React Native solution.*

Thanks for reading! If you've enjoyed this article, please leave a

Say Hello On: Twitter | LinkedIn