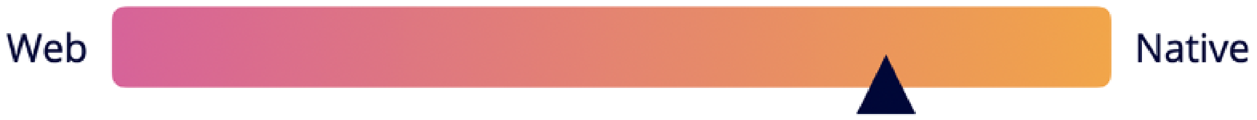
marino_

Native & Hybrid Mobile Development

Hybrid development is a spectrum, depending on the context it can make sense to lean more on web technologies or native.

Hybrid Development Spectrum



Web

Native

At **Marino,** it is common for clients to ask about developing their application entirely using a cross-platform hybrid framework.
The potential benefits of a single codebase to maintain with a shared user interface between iOS and Android are obviously appealing, and cross-platform apps do have
their place.

When your mobile app is core to your business however, especially when mobile app user experience has been identified as a key differentiator much of the cross-platform promise turns to myth.

> ***When your mobile app is core to your business however, especially when mobile app user experience has been identified as a key differentiator, much of the cross-platform promise turns to myth.***

The Single Codebase Myth

Cross platform frameworks can only ever support a subset of native features so for all but the most basic apps, native code will still be required.

Different cross platform frameworks rely on different plugin mechanisms to access native features. Oftentimes cross platform apps consist of an assortment of open source third party plugins or naïve native code written by web developers.

The potential for a single unified codebase is instead spread across separate cross platform, iOS, and Android codebases and reliant upon 3rd party plugin support.

The Single Codebase Myth

Mobile users are trained by the interactions and UI idioms built in the most popular apps and come to expect those, as well as the latest platform features, in all the apps on their platform.

Cross platform frameworks typically either attempt to imitate the native controls of each platform or present their own so when leading digital experiences are required, custom platform specific UI work will follow, even where it can be done using cross-platform technology.

> ***“Mobile users are trained by the interactions and UI idioms built in the most popular apps and come to expect those”.***

Performance

Native apps are orders of magnitude faster than cross platform apps. Shorter load times and a smoother look and feel overall leads to a better user experience.

App and cloud monitoring company Dynatrace found that:

> ***"While 79% of consumers would retry a mobile app only once or twice if it failed to work the first time, only 16% would give it more than two attempts. Poor mobile app experience is likely to discourage users from using an app again."***

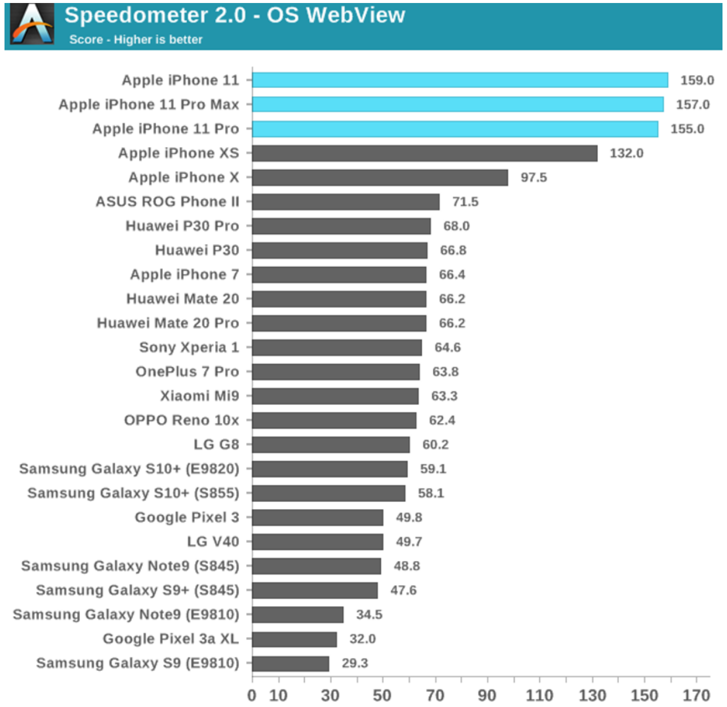
The most common frameworks rely on HTML, CSS and JavaScript. The performance of these technologies across devices and operating systems varies widely so relying on it for the entire mobile app will result in an inconsistent experience for different customers, even the best of which will not compare to native performance.

Performance

As is illustrated here, iPhones are well ahead in JavaScript performance and. Notably, the Pixel 3 from Google, last year's flagship Android device performs at much the same level as the iPhone 6s from 2015.

To achieve even a competitive experience will require work to heavily optimise for Android users, which typically make up the majority of our customers' users today.

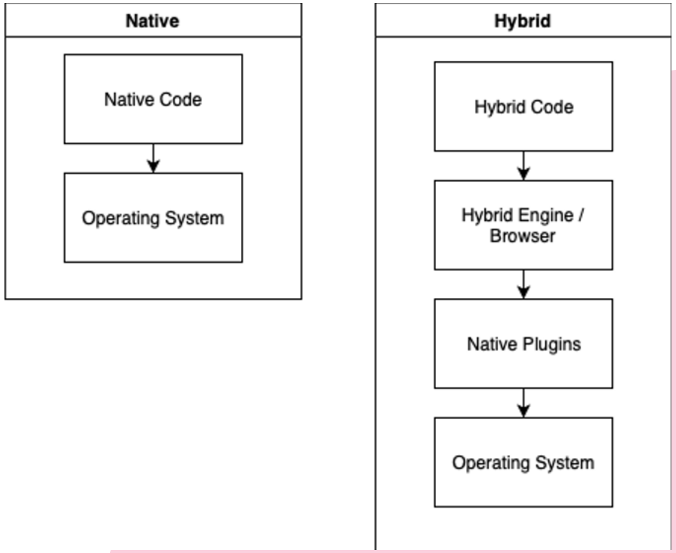
Scores for a selection of recent phones running the Speedbench 2.0 JavaScript benchmark.



Security

The architecture of cross platform hybrid apps contains more components than a native app.

Native apps are built from native code which talks to the Operating System SDK directly. Hybrid apps are built from non-native code which must be interpreted by a hybrid engine or run via a browser component.



Security

Where the hybrid app needs to call out to native SDK functions, say to perform device authentication, hybrid frameworks typically rely on a plugin system.

It is often the case that vendors of hybrid systems will list this as a strength. However, each of these additional layers add complexity that you are not in control of. This adds a security risk to each layer and to the connections between each layer.

Many plugins used by the most popular frameworks are small open source efforts, often not developed by platform experts. For applications where security is paramount adding such dependencies should be done with extreme caution.

Development Tooling

Both Android and iOS are well supported with development tools from Apple, Google and the wider ecosystem.

Several development steps are required to meet compliance standards for enterprise apps, including developing to code standards, code static analysis and security checking.

The tools which carry out these steps always support the latest versions of the native development SDKs but often lag the current versions of Hybrid frameworks, and do not necessarily support them at all.

> ***A common problem with hybrid app frameworks is that organisations can get locked into specific versions and cannot progress beyond it.***

Previously we worked with a client who had chosen a product which used Ionic 1.3.3 , while Ionic was on version 4.

The changes between these two versions meant that there was on simple path to move to the latest version. Additionally, the vendor had made a significant investment in Ionic 1.3.3 code, which meant they had no immediate plans to move.

This scenario doesn't occur with native development as Google and Apple always want to ensure that developers can easily move to the latest versions of their SDKs to access the latest features.

Popular Apps switching to Native

FACEBOOK

Facebook moved from a HTML5 based app to native:

Today, Mark Zuckerberg revealed that Facebook's mobile strategy relied too much on HTML5, rather than native applications. Not only was this a big mistake with mobile, but Zuckerberg says that its biggest mistake period was the focus on HTML5. This is the first time that the Facebook CEO has openly admitted this, but things are looking good for the new iOS native app. According to Zuckerberg, people are consuming twice as many feed stories since the update to the new iOS app

TEAMWORK

Teamwork moved from HTML5 wrapped apps to native .

"One upside to going down the native route is that users can see you care as you've tailored an app to their OS. Sure they'll be distinct, the Android one will have a strong material design feel, but we'll strive to keep in sync with each other and Teamwork Chat in general when it comes to features as well as the overall experience."

AIRBNB

AirBNB have a very detailed write up of why the moved from a react native app to a native app . On maintaining the quality bar AirBNB said:

"We built shared element transitions, parallax, and were able to dramatically improve the performance of some screens that used to frequently drop frames. However, some technical challenges such as initialization and the async first render made meeting certain goals challenging."

On writing once instead of twice:

"Only a small percentage of our app was React Native. In addition, large amounts of bridging infrastructure were required to enable product engineers to work effectively. As a result, we wound up supporting code on three platforms instead of two. We saw the potential for code sharing between mobile and web and were able to share a few npm packages but beyond that, it never materialized in a meaningful way."

https://techcrunch.com/2012/09/11/mark-zuckerberg-our-biggest-mistake-with-mobile-was-betting-too-much-on-html5/
https://www.facebook.com/notes/facebook-engineering/under-the-hood-rebuilding-facebook-for-ios/10151036091753920/
https://medium.com/airbnb-engineering/react-native-at-airbnb-f95aa460be1c
https://adamlynch.com/under-the-hood-of-a-hybrid-app/

It is clear from these examples, and from Marino's own extensive experience delivering mobile apps to tens of millions of end users over the past decade, that to deliver the best experience to end users, and long-term viability to our clients that you should generally lean more to the native end of the hybrid development spectrum and deploy cross platform approaches tactically.

marino_

Marino are your perfect partner to plan, design and craft experiences that are secure, easy to use, and exceptional.

Let’s Chat