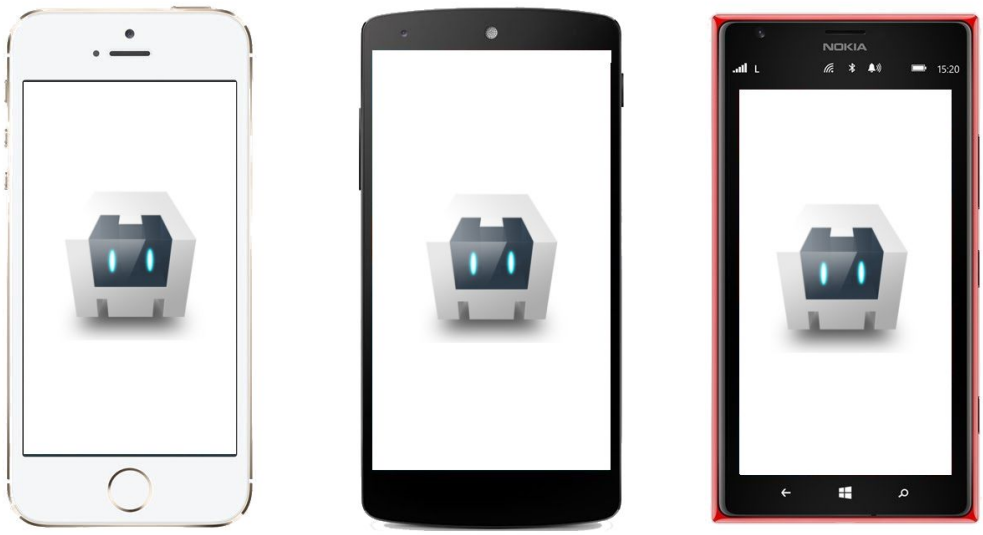
PhoneGap / Cordova



Nicolas GIGOU

M2 MIAGE, ISTIC, 2016/2017

Introduction

According to the statita.com site, almost 180’000 different mobile applications have been downloaded all over the world during 2015, all platforms concerned. In 2017, about 270’000 downloads are expected. This change of 33% suggests the potential of the mobile applications market. Some contenders stand out from others and are growing while others are losing revenue. While giants like Apple (iOS) and Google (Android) compete to win market share in a global pie, other organizations benefit to nibble their piece. It is when PhoneGap and Cordova are coming. But how these two tools can they make their way among the competitors taking advantage of market inertia and how can they impose their ideals and wishes to a community on other technological alternatives?

What is PhoneGap / Cordova ?

PhoneGap and Cordova are both projects which allow, all web developers, to easily develop and implement hybrid mobile applications. The notion of technological hybridity means that instead of using native technologies and languages (the Android language is used for Android devices, the Objective-C language for the Apple devices), we use web technologies like the ones we are developing websites. Normally, the application should not be able to run on these devices but the PhoneGap and Cordova projects offer the possibility to, like a schematic way, pick your application and put it in a box corresponding to the target operating system. Then, if you want to make it run on your iPhone, wrap it within an iOS box.

PhoneGap or Cordova?

Some people often confused PhoneGap and Cordova. It seems important to know the difference, even if it is not marvelous, between both of them. To distinguish both projects and justify a premium version of Cordova, the PhoneGap developers are working on a set of robust integrated tools like a powerful CLI (command line interface), a desktop application, a mobile application (to check the result of your application within several devices) and PhoneGap Build, a builder tool to easily package the assets and prepare the application to publish it on stores (Google Play Store, AppStore, ...). These services are paying and depend on how many projects or developers are concerned. If it is for a single developer for a single private project, there is a free plan (it is free no matter how many public projects there is). But if you add, by example, the number of private projects (from 1 to 3), you will have to pay $120/year. For 25 private projects and 10 private collaborators, you will have to pay $900/year. Cordova being an open-source project, it is totally free but does not provide these additional services.

Understand the past to glimpse the future

Previously, I said that Adobe is the owner of the PhoneGap project. In fact, PhoneGap was initially a Nitobi project and has been created in 2008. This Canadian company sold the PhoneGap project to Adobe after an official agreement in 2011. It was a wish from Nitobi to help the employees to focus solely on the development and continue its work on efficient development across mobile platforms. Before being sold, it was around 600.000 downloads for the open source framework and thousands of applications built using it. In 2011, the chief executive officer, M. Andre Charland said “We share the same philosophy about enabling extraordinary mobile and Web applications. Becoming part of the Adobe family with its industry-leading tools and technologies opens up amazing new opportunities for PhoneGap and our customers.”. Even after some years, they continue to follow their beliefs.

You may asking what their beliefs are. The first one is about the technical barriers encountered. The revolution imposed by PhoneGap is the possibility to use the web languages which has reached a critical mass. By now, everyone can publish anything from anywhere. It makes it kind of universal. The second belief could be discussed but the IT history shows us that all technology deprecates. Even if a specific language know an incredible rise in reputation and global use, undoubtedly, it will experience a decline in its use or even total abandonment.

But what about Cordova now ? I focused mainly on the original project but as I said before, it has been forked in 2011. Cordova corresponding so to the open source core of PhoneGap (PhoneGap is Cordova plus extra features). Even if at the beginning, the differences between those two were minimal, it quickly diverges. When Cordova was following a free plan process, Adobe purposes premium solutions for PhoneGap users by build out a proprietary set of services around PhoneGap ecosystem.

Along the years, Cordova known some overlays like Ionic. The development environment is always evolving to help developers to implement native applications using the Webview engine of your smartphone. Just like his brother PhoneGap, Cordova wanted to follow a creed which the development team is also always: make Cordova renders obsolete. Currently, most people do not realize that Cordova is capable of doing almost as many things as the languages called 'native'. The only real limit is the availability of 'bridge' plugins. This notion of bridge is quite simple to understand. The heart of Cordova, depending on the operating system of your phone, will provide libraries to make the transition, a 'bridge' between a call to your function in your code to access to the GPS (Global Positioning System) or to the camera by example and the native function of your phone.

It is in this way that dialogue your application and your phone, be it Android, iOS or BlackBerry. To return to the plugins, if the ones you want is not available, it is up to you to develop it according to your needs. The documentation provided by the parent team make it easily possible to implement. To take one example, let's talk about the Geolocation API. While the GPS on mobile phones became popular thanks to the iPhone, web browsers other

smartphones have built it much later. It is precisely in this kind of situation comes Cordova to compensate for that lack of technology and meet the needs of developers in access to native features and components. Today, the situation is the same for the API of the camera of your phone. Too few mobile browsers allow use of this feature. Cordova therefore is developing access using libraries for the entire community.

As you can see, PhoneGap and Cordova tend themselves to disappear in order to force the hand of mobile browser to add new features and future, standardize the application source code. However, it is important to speak about developers who find that it is not proper or natural to mobile applications with source code that is not initially planned for this. How can the community decide between the two camps? Are there any real form of competition between the two or is it rather a kind of parallel partner offering features not included in the functions of dictionaries of current smartphones? Is performance an important factor in the balance of the decision?

Contention with native developments?

Before speaking concretely face contention to native developments, it is important to distinguish three levels of applications. The first layer is the so-called native who, for a given operating system, developed in a given language. The second layer is the one favored by PhoneGap and Cordova, hybrid applications, feeding web languages and native base for mounting applications. The third and final layer is the web applications. Let us focus on this second layer for the moment as it is that the subject matter. When looking contenders, we realize that a technology in particular stands out: Titanium by Appcelerator.

Titanium may appears to be the same as PhoneGap but when you look closer there, you realize that Titanium is a framework for native applications when PhoneGap is a framework for hybrid applications. Where PhoneGap encapsulates the application in an operating system defined during the compilation of the project, Titanium working on a JavaScript interpreter included in most current smartphones (V8 for Android and JavaScriptCore for iOS). The fact is that PhoneGap is more flexible than its contender because Titanium is not working for distributions such as Windows Phone, Bada and WebOS. Let orient the research to web contenders. This type of application is still different from before because it requires a permanent internet connection to operate while an application using the WebView of your phone does not need. This is a potential response in terms of competition but limited its comparison in terms of mobile applications. Finally, the third type, the native applications. These applications are the real contenders of a framework like Cordova and PhoneGap. The four main actors in the mobile native development are Apple with the Objective-C language, Android with the language of the same name, BlackBerry with the Java and Windows with mainly C# and .NET.

Hardly comparable, it is a totally different world. The hybrid applications have seen the light to allow web developers to implement mobile applications while native ones are made for it, designed for it, built for it. The performance are obviously better for the native ones. Why ?

Native application are using a language made for your smartphone and hybrid applications need a first interpreter of the source code. Then it needs a native wrapper to suit the app perfectly. This additional layer damages the performance, especially on heavy projects.

The limits

Actually, the real limit encountered do not depend on PhoneGap or even Cordova but rather from your phone. The quality of the user interface will vary based on the Webview and rendering engine of your smartphone. If the interpretation of your code failed, you will not have the desired result. Concretely, the best one is the iOS rendering engine. It provides the best performance. Android is second with its functional Webview. Another source of limit, weakness could be the standard cross browser issues. The web environment knows so many different ecosystem to handle and if you do not test it correctly, you are not safe to get an expected issue. Finally, as a third possible issue, I would like to talk about the gap between the native-quality UI performance and the hybrid-quality one. Even with advanced framework for hybrid applications like Sencha, it is still an unreachable aim.

Conclusion

The ending of this study comes by itself when trying to search some responses to the problematic “how these two tools can they make their way among the competitors taking advantage of market inertia and how can they impose their ideals and wishes to a community on other technological alternatives?”. Neither PhoneGap nor Cordova really want to stand up versus the native applications. They are rather some kind of precursors trying to force a technical evolution of the existing projects using only web languages and allowing web developers to participate to this race for continuous improvement. They are a new well of possible answers to growing user needs. Their common belief to seize to exist could be a possible result in some years but our environment still needs to involve to become a new kind of universal language platform.

Bibliography

"A high-level summary of what PhoneGap is all about." *Adobe*. Adobe PhoneGap. <http://phonegap.com/about/>.

"Apache Cordova." *Joe Bowser*. Wikipedia, the free encyclopedia. 14 September 2016. <https://en.wikipedia.org/wiki/Apache_Cordova>.

"Adobe Announces Agreement to Acquire Nitobi, Creator of PhoneGap." *Stefan Offermann*. Adobe Systems. 3 October 2011. <http://www.adobe.com/aboutadobe/pressroom/pressreleases/201110/AdobeAcquiresNitobi.html>.

"PhoneGap Beliefs, Goals, and Philosophy." *Adobe*. Adobe PhoneGap. 9 May 2012. <http://phonegap.com/blog/2012/05/09/phonegap-beliefs-goals-and-philosophy/>.

"The Last Word on Cordova and PhoneGap." *Maxime Lynch*. Ionic. 6 March 2014. <http://blog.ionic.io/what-is-cordova-phonegap/>.

"Comparing Titanium and PhoneGap." *Appcelerator*. Appcelerator Inc. 12 May 2012. <http://blog.ionic.io/what-is-cordova-phonegap/>.

"Applications natives, web ou hybrides : que choisir ?." *Appsolute*. 3 August 2015. <http://blog.ionic.io/what-is-cordova-phonegap/>.

"PhoneGap Build: Pricing Plans Update" *Andrew Lunny.* Adobe PhoneGap. 13 October 2011. <http://blog.ionic.io/what-is-cordova-phonegap/>.