

Application Web Application Native Application Hybride / Cordova



Application Native

Le développement natif consiste à développer avec les outils et langages propres à chaque système d'exploitation.

Tableau des langages et IDE* les plus courants pour chaque système d'exploitation

SYSTÈME D'EXPLOITATION	LANGAGE(S)	IDE*
iOS (Apple)	Objective C et Swift	Xcode
Android	Java	Android Studio ou Eclipse
Windows Phone	C# (se dit "C sharp")	Visual Studio

* IDE : environnement de développement intégré. C'est un ensemble d'outils pour augmenter la productivité des développeurs.

Ses avantages

- Plus rapide et plus fiable : meilleure réactivité et résolution supérieure = meilleure expérience utilisateur
- Permet un accès plus facile aux fonctionnalités de l'appareil
- Notification push : permet d'alerter l'utilisateur sur du nouveau contenu, ou une offre promotionnelle.
- Ne nécessite pas obligatoirement internet : possibilité d'accéder à l'application sans connexion web.
- Respecte les codes design de chaque plateforme (= expérience utilisateur optimale)

Ses inconvénients

- Construite pour une seule plateforme (iOS, Android...)
- Langage compliqué (Java, Swift, Objective-C) => coût plus élevé
- La maintenance d'une application native est compliquée

Web App

C'est la version mobile d'un site Web, adaptée à la taille et caractéristique de l'écran des appareils mobiles.

Elle fonctionne comme un site web, aucune différence de plateforme, de système d'exploitation, de code.

ex : Pathé cinéma, Le Monde, Mc Donald's

WebApp

Avantages

- ★ accessible via plusieurs navigateurs et plusieurs systèmes d'exploitation -> aucun problème de compatibilité + recherche facile pour l'utilisateur
- ★ stockée sur un serveur -> pas d'installation -> gain mémoire
- ★ mise à jour automatique
- ★ plus facile et rapide à développer (et moins chère)

WebApp

Inconvénients

- ★ nécessite une connexion internet -> si mauvaise qualité connexion -> impact sur la rapidité de l'application
- ★ créée en une seule version et pour toutes plateformes -> plus lente et moins adaptées à chaque plateforme -> moins performante
- ★ pas d'accès aux fonctionnalités spécifiques aux mobiles (=>PWA)

Application hybride

- ★ Un seul et unique code pour plusieurs plateformes
- ★ L'application est développée à partir de langage web (HTML5, JavaScript, CSS...)
- ★ L'application s'appuie sur des technologies natives mobiles pour utiliser certaines fonctionnalités du smartphone
- ★ L'application est téléchargeable depuis les stores (magasins d'applications) et sera installée sur le mobile, contrairement à la WebApp qui n'est consultable que depuis le navigateur
- ★ Les applications hybrides seront encapsulées dans une surcouche adaptée aux différents systèmes d'exploitation souhaités, c'est là qu'intervient Cordova (le développement d'application hybride repose sur des solutions comme phonegap/Cordova, qui permettent de créer une application indépendante à partir de pages webs)

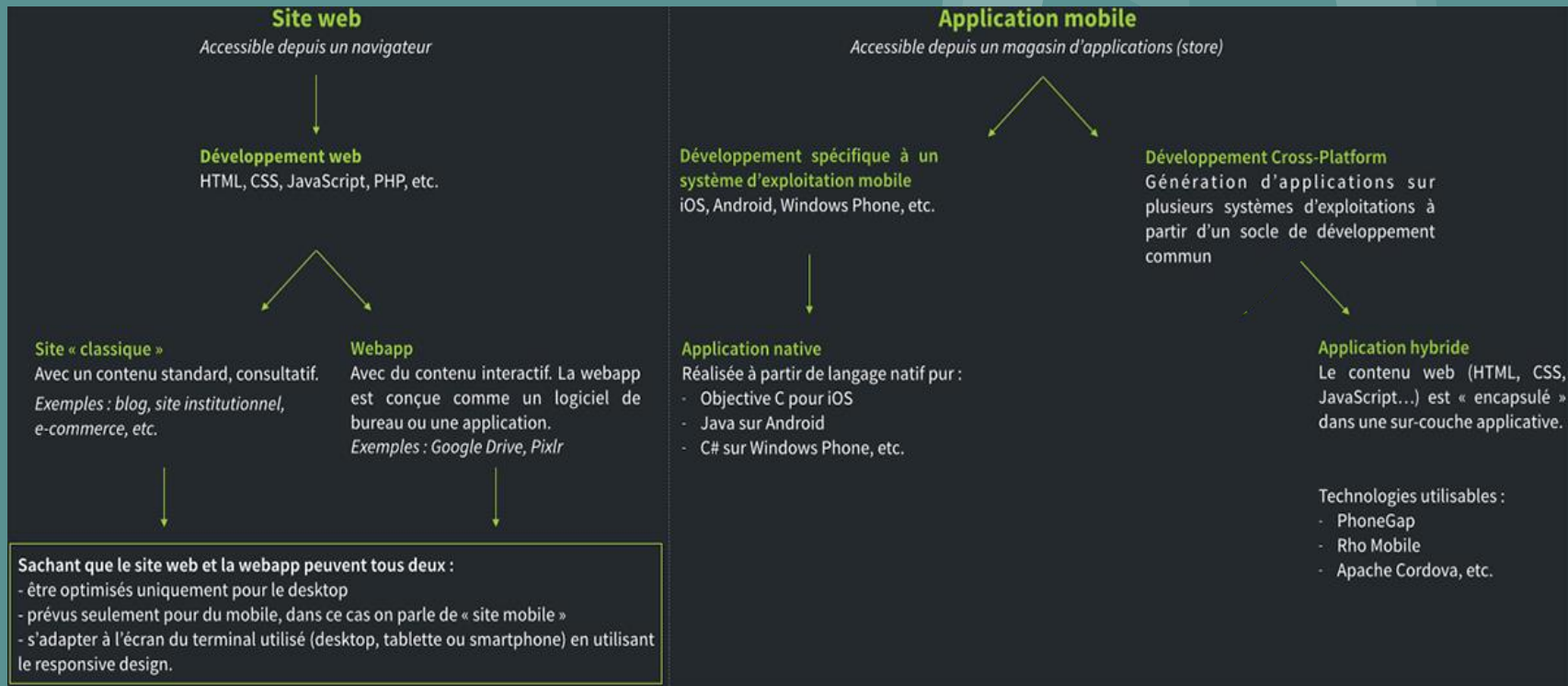
Les avantages

- ★ C'est une technologie multi-plateforme, ce qui est un gain de temps (si le projet n'est pas trop complexe) et donc un gain d'argent
- ★ La maintenance de l'application est plus facile puisqu'il n'y a qu'une seule version à revoir pour plusieurs plateformes
- ★ Les compétences web sont plus répandues que les compétences mobiles, les coûts de développement sont donc généralement moins importants
- ★ L'application mobile hybride est conçue pour être téléchargeable, elle peut donc être monétisée et bénéficie des classements dans les stores
- ★ Le stockage des informations, du type compte utilisateur, accès aux fichiers... se fait en local

Les inconvénients

- ★ Les performances de l'application sont moins bonnes et moins stables puisque le système est moins adapté à chaque plateforme
- ★ Il faut s'assurer de la compatibilité du projet avec les technologies hybrides (PhoneGap, Cordova ou Rho Mobile)
- ★ Les contenus trop lourds peuvent avoir de plus forts impacts sur les performances qu'avec une application native.
- ★ Le mode hors-ligne est plus limité et plus délicat à concevoir (mais reste possible)
- ★ En utilisant une application hybride, on devient dépendant d'un logiciel tiers (en l'occurrence Cordova) et donc de sa compatibilité avec les nouvelles versions d'OS
- ★ Développer en hybride permet de mutualiser le développement sur les deux principaux systèmes d'exploitation (AppleStore et GooglePlay) mais elles peuvent être refusées sur les autres plateformes.

Comparaison web app / app native / app hybride



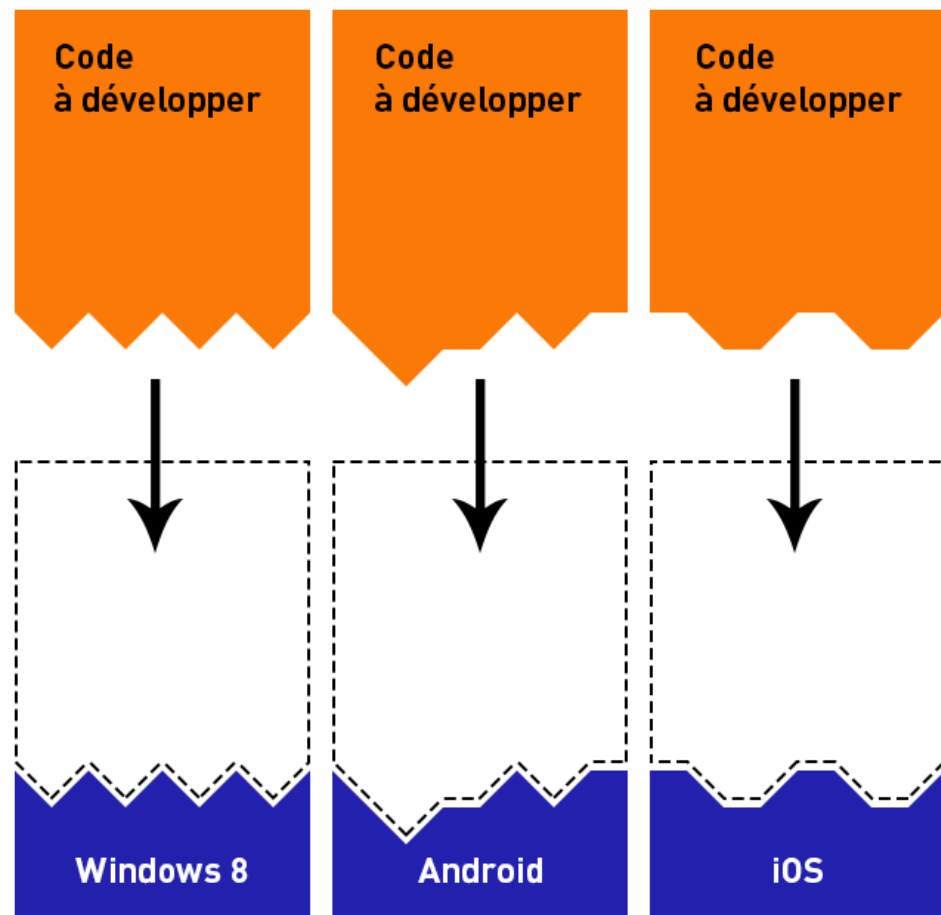
Cordoquoi ?

Apache Cordova ou plus anciennement Apache Callback ou PhoneGap est développé par la Fondation Apache.

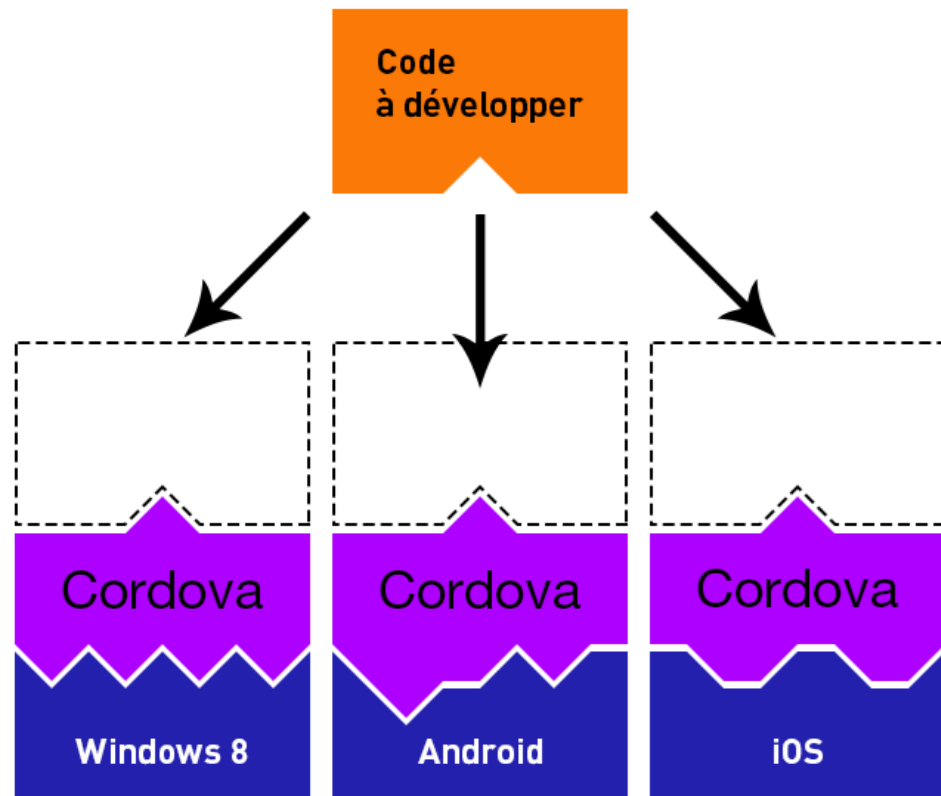
Il permet de créer des applications pour différentes plateformes (Android, iOS, Ubuntu, Windows 8...) en HTML, CSS et JavaScript.

Les applications qui en résultent sont hybrides, ce qui signifie qu'elles ne sont ni vraiment natives, ni purement basées sur les langages HTML, CSS et JavaScript.

Sans Cordova



Avec Cordova



Comment ça marche ?

Au début vous allez créer vos fichiers HTML, CSS et Javascript et créer votre application web normalement.

Ensuite vous allez demander à Cordova de construire votre application.

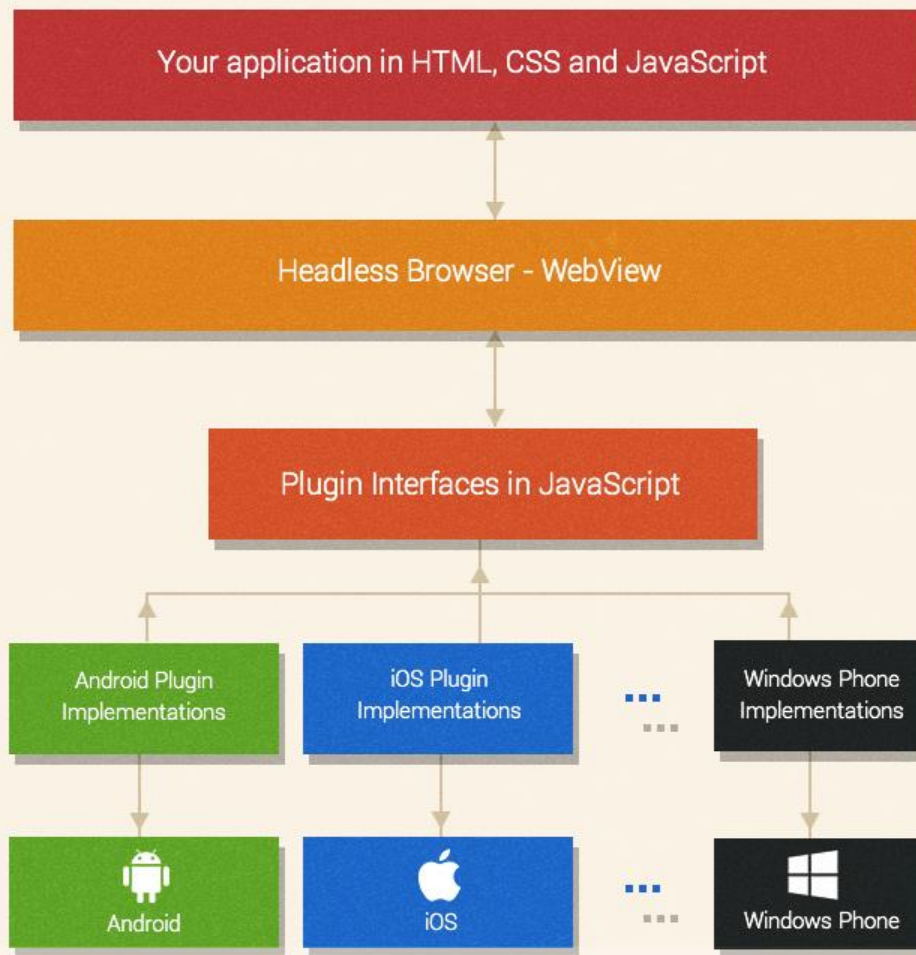
Automatiquement, il va créer un nouveau projet android ou iOS qui sera écrit en langage natif et va ensuite ouvrir une WebView qui va contenir vos fichiers HTML.

Qu'est ce qu'une webview ?

La webview donne la possibilité d'ouvrir n'importe quelle URL à l'intérieur de votre application, en gros il faut voir une webview comme un chrome miniature qui est utilisé pour montrer du contenu dans votre application hybride.

La webview utilise la librairie Cordova, librairie qui permet d'interagir avec certains éléments du téléphone même si l'application est écrite en HTML, CSS, JS, par exemple : Accéléromètre, boussole, GPS, appareil photo.

Au final, l'utilisateur va se retrouver avec une application mobile qui sera compilée en langage natif mais qui ouvrira tout simplement votre page HTML.



Les plugins

Apache cordova vient avec un lot de plugins pré-développé qui donne accès à la caméra du mobile, GPS, Explorateur de fichiers et d'autres.

Vous pouvez télécharger d'autres plugins sur le site de cordova.

Cordova Plugins



What is a Cordova plugin?

A plugin is a bit of add-on code that provides JavaScript interface to native components. They allow your app to use native device capabilities beyond what is available to pure web apps.

[+ Contribute Plugins](#)[? Plugin Help](#)

Must Support Platform(s):

Android	iOS	Windows	Blackberry	Ubuntu	Firefox OS
macOS	WP8	Browser			

3253 result(s) found

Sort: Quality▼

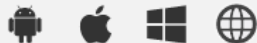
parse-phonewap-plugin-push

v1.2.3 by **gortok**



Register and receive push notifications.

Last updated **458 days ago**



cordova-plugin-camera-roll-location

v0.3.3 by **mixersoft**



get location metadata with photos from CameraRoll

Last updated **435 days ago**



Installation des dépendances

Pour installer Cordova vous aurez besoin des packages suivant

- ★ Java
- ★ Apache Ant (Task Runner)
- ★ Android SDK (Kit de développement d'android)
- ★ NodeJS (Installation de Cordova par node package manager ou npm)
- ★ Git

Fonctionnalités	App Native	App Web	App Hybride
Performances/rapidité	+++	+	++
Accès aux fonctionnalités du mobile	+++	-	++
Mode hors-ligne	++	-	++
Disponibilité sur plateformes (IOS, Windows et Android)/portabilité	++ (si développé sur chaque plateforme)	+++	++
Design optimisé par plateforme	+++	-	++
Mise à jour automatique des pages	-	+++	+
Utilisation mémoire locale	+++	-	++
Coût	-	+++	++
Langage	Objective-C/Java/Swift	HTML5, CSS, JS	HTML5, CSS, JS