

## Chapitre 2 : Les solutions de développement mobiles

### Introduction

Afin de développer des applications mobiles, plusieurs technologies sont offertes aux développeurs. Ces technologies sont en constante évolution et une multitude de plateformes de développement technique est apparue. Nous distinguons 3 types d'applications mobiles : native, web et multiplateformes.

Dans ce chapitre, nous détaillons chaque type d'application en présentant sa définition, ses avantages et inconvénients et évidemment des exemples de solutions.

## I. Application native

### I.1. Définition

Une application mobile native est une application développée pour s'installer et s'exécuter sur un système d'exploitation mobile spécifique. Ce type d'applications est développée en utilisant un environnement proposé par le propriétaire de l'OS mobile qui impose un (des) langage(s) de développement(s). Les applications mobiles de chaque système d'exploitation sont hébergées dans des boutiques en ligne (Stores).

### I.2. Avantages

Les applications natives peuvent avoir accès à toutes les fonctionnalités du terminal mobile. Autre fait, son contenu est optimisé au maximum pour la mobilité. Un autre avantage est que les applications natives sont utilisables sans connexion internet.

### I.3. Inconvénients

Les technologies différentes pour chaque constructeur des terminaux mobiles nécessitent de développer des applications pour chaque plateforme (Android, iOS, Windows Phone,...) ce qui augmente leurs coûts de développements.

Le fait d'avoir plusieurs versions d'applications natives pour chaque plateforme nécessite des compétences pointues en développement.

Un autre point faible, des problèmes de compatibilité peuvent se poser après les mises à jour de l'application natives.

### I.4. Exemples de solutions natives

Le tableau suivant illustre les 3 principaux systèmes d'exploitation mobiles ainsi que leurs langages et environnements de développement.

SYSTÈME D'EXPLOITATION	LANGAGE(S)	IDE <sup>1</sup>
iOS	Objective C et Swift	Xcode
Android	Java, Kotlin	<b>Android Studio</b>
Windows	C#, VB.NET	Visual Studio

---

<sup>1</sup> IDE : Integrated Development Environment ou « environnement de développement » est un logiciel qui rassemble un certain nombre d'outils permettant de développer d'autres logiciels

## II. Application web

### II.1. Définition

Une application web (Web Application) est une application mobile développée avec les outils de développement web classiques : HTML5, CSS3 et JavaScript.



Figure 1: HTML, CSS, JS

Elle est accessible grâce aux navigateurs web des terminaux mobiles indépendamment de leurs caractéristiques, marques et systèmes. Il existe 2 types d'applications web :

#### 1.1. Application web classique

Développée en HTML, accessible et utilisable depuis un navigateur internet. Elle ne nécessite ni de télécharger, ni d'installer l'application.

#### 1.2. Application web progressive (PWA : Progressive Web App)

Une application web progressive est une version allégée d'un site internet qui peut être utilisée en offline, télécharge son contenu de manière progressive (politique de cache, grâce au service worker), propose une expérience utilisateur riche (responsive, plein écran, gestion de l'écran...), adapte sa structure et son contenu selon la plateforme. Elle peut être ainsi installée sur un téléphone mobile sans avoir besoin de diffuser une application sur les stores traditionnels.

**Exemples :**    [www.pinterest.fr](http://www.pinterest.fr)                      [www.uber.com](http://www.uber.com)                      [www.9gag.com](http://www.9gag.com)

### II.2. Avantages

Les applications web sont très simple à mettre en œuvre parce qu'elles utilisent des technologies web ce qui permet à de telles applications d'être adaptatives (Responsive design<sup>2</sup>).

En plus, elles offrent un gain en temps et en coût de développement parce que la même application, hébergée dans le serveur, est appelée par les différents navigateurs sans tenir compte des spécificités du système d'exploitation du terminal mobile client.

Les PWA combinent le meilleur des nouvelles technologies web et des applications natives. Les fonctionnalités disponibles sont par exemple :

- Créer un raccourci du site ou de l'application directement sur l'écran d'accueil du visiteur
- Recevoir des notifications push (par exemple, à chaque nouvel article, à chaque nouveauté sur le site, comme pour une application)
- Accéder aux fonctionnalités du téléphone telles que l'appareil photo, la géolocalisation, le micro, la boussole...
- Atteindre les contenus hors-connexion grâce à un mode hors-ligne

---

<sup>2</sup> Responsive Design : Web Adaptatif : Technique permettant d'adapter un site web selon le support d'affichage

### II.3. Inconvénients

Les applications web sont moins fluides et réactives que les applications natives : Elles dépendent du navigateur du terminal et non directement du système puisqu'elles n'utilisent pas la mémoire embarquée du terminal mobile ce qui demande un temps de chargement trop long.

En outre, ce type d'applications nécessite forcément une connexion internet et présente des limites dans la communication avec les différents capteurs du terminal mobile tels que GPS, Bluetooth,...

## III. Application multiplateforme (Cross-platform)

### III.1. Définition

Une application mobile multiplateforme permet de maintenir une seule base de code qui peut s'exécuter sur plusieurs systèmes d'exploitation. Elle est opérationnelle sur tous les systèmes et ne nécessitent qu'un seul développement. Il en existe 2 sortes (la différence est dans le langage utilisé) :

#### 1.1. Application native générée

C'est développé avec C#, du Ruby ou du Javascript

**Exemples** : Xamarin, Titanium

#### 1.2. Application hybride

L'application sera codée avec les outils classiques du web (HTML5/CSS3/JavaScript) et rendue native en la mettant dans une WebView.

Ce type d'application peut être développé en utilisant un **Framework** qui génère un fichier exécutable qui, une fois installé, lance une application mobile sans que l'utilisateur s'aperçoive de la présence du WebView.

**Exemples** : ionic, Flutter, React Native, ...

### III.2. Avantages

Les applications multiplateformes sont développées avec des technologies web connues ce qui ne nécessite pas une formation supplémentaire pour les développeurs. Ceci permet de réduire le coût et le temps de développements en regroupant le code.

En outre, grâce aux **WebView**, il est possible d'exécuter du code web au sein d'un environnement natif. Ainsi, il est possible de lancer la caméra du terminal (natif), via une simple fonction Javascript (web). Ce qui donnerait ceci par exemple :

```
function capturePhoto() {  
    // Prend une photo en utilisant la fonction camera du téléphone  
    navigator.camera.getPicture(onPhotoDataSuccess, onFail, {  
        quality: 50,  
        destinationType: destinationType.DATA_URL  
    });  
}
```

### III.3. Inconvénients

Le développement multiplateforme peut entraîner une adaptation inévitable à l'appareil. Par exemple, le bouton menu spécifique à Android n'existe pas chez Apple, ce qui oblige le développeur à tenir compte de cet aspect en séparant le traitement selon l'appareil concerné engendrant ainsi des tests supplémentaires.

## Conclusion

Le schéma en couche ci-dessous présente un résumé des différents types d'applications mobiles :

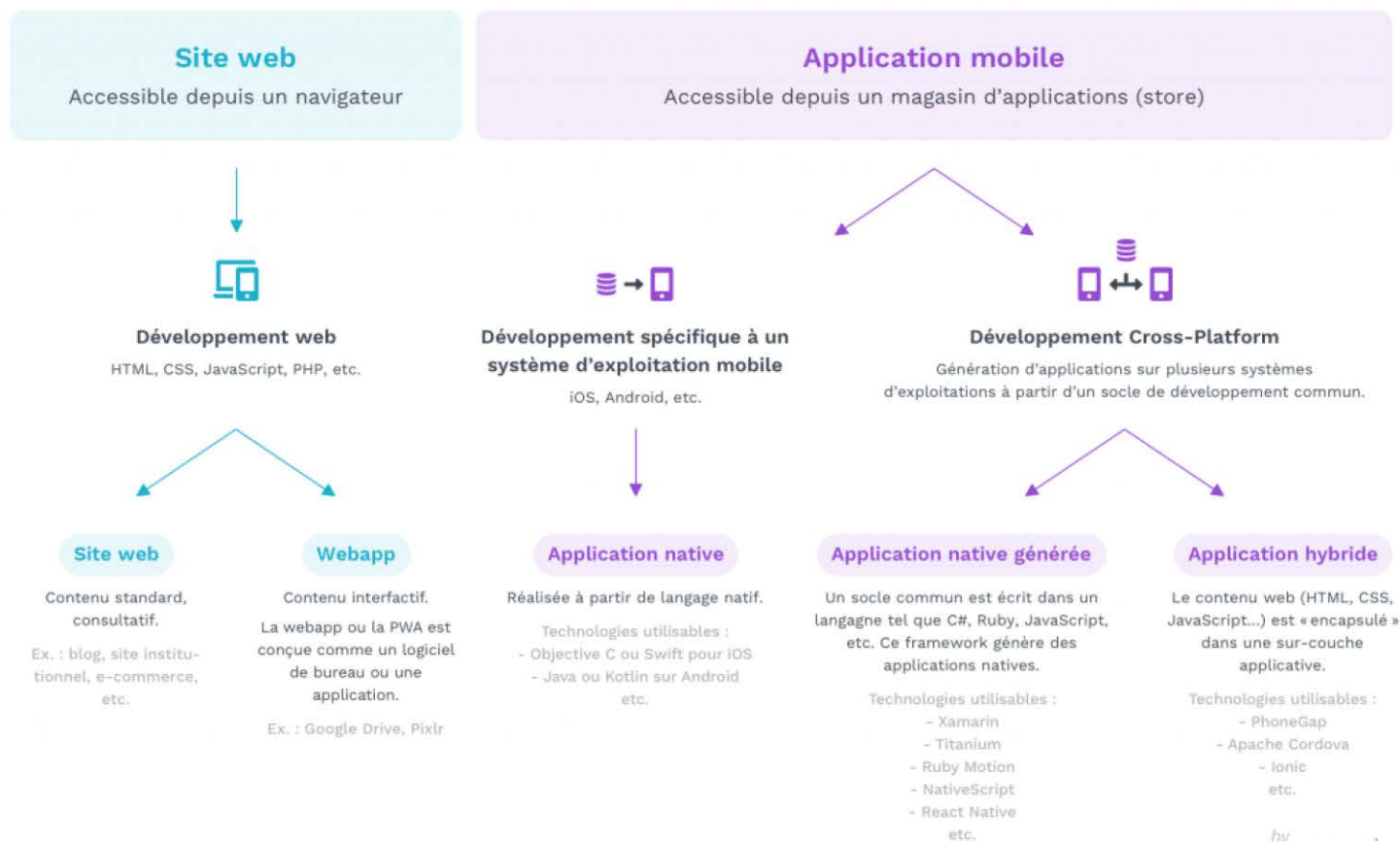


Figure 2 : Différents types d'applications mobiles

Les chapitres qui suivent traitent uniquement le développement natif pour le système d'exploitation Android.