

Offline Apps et Compilation Native

Programmation Web avancée et mobile – Mif38

Aurélien Tabard

La direction :

des applications Web accédant aux API natives

Les alternatives techniques pour le développement mobile

▶ HTML 5

- ▶ Progressive apps + responsive + device APIs

▶ Natif mobile

- ▶ iOS, Android, Windows 10 Mobile, Tizen, Sailfish OS...

▶ Cross-platform mobile

- ▶ Xamarin Studio (C#), Adobe Air (Flex/Actionscript), Titanium (XML/Javascript)

▶ Web embarqué

- ▶ Cordova
- ▶ ReactNative

Web applications

- ▶ Initiative du W3C, menée par le WebApp WG
- ▶ Différentes spécifications visant à standardiser la notion d'application Web
 - ▶ Web IDL
 - ▶ Web Components
 - ▶ Widgets (interfaces & packages)
 - ▶ DOM (Level 4, Events, shadow...)
 - ▶ Service Workers
 - ▶ Device API
 - ▶ Offline
 - ▶ ...
- ▶ Spécifications / implémentations en cours

Progressive Web Apps

Trois principes

- ▶ Fiable
- ▶ Rapide
- ▶ Engageant

Les principes

Progressif - Pour n'importe quel utilisateur indépendamment du navigateur : on améliore progressivement.

Responsif - S'adapte au dispositif.

Indépendant de la connectivité - Fonctionne hors-ligne ou en mauvaises conditions grâce aux services-workers

App-like - Impression d'utiliser une application, avec un modèle d'*app shell* qui sépare les fonctionnalisés de l'application du contenu.

Frais - Toujours à jour grâce au processus d'update des service worker.

Les principes

Sûr - Servi en HTTPS.

Découvrable - comme une “app” grace au manifest (standard du W3C) et le *registration scope* des service worker qui permettent aux moteurs de recherche de les trouver.

(ré-)Engageant- Facilite sur le retour sur l’application via les notifications push.

Installable - S’installe facilement sur le “home screen” des utilisateurs mobiles

Liabale - Permet de facilement partager une URL.

Les briques de base

- ▶ Design responsif
- ▶ App-shell
- ▶ Manifest
- ▶ Service-workers

Les alternatives techniques pour le développement mobile

- ▶ Reponsive Web

- ▶ HTML 5

- ▶ Natif mobile

- ▶ iOS, Android, Windows, Tizen, Firefox OS...

- ▶ Cross-platform mobile

- ▶ Xamarin Studio (C#), Adobe Air (Flex/Actionscript), Titanium (XML/Javascript)

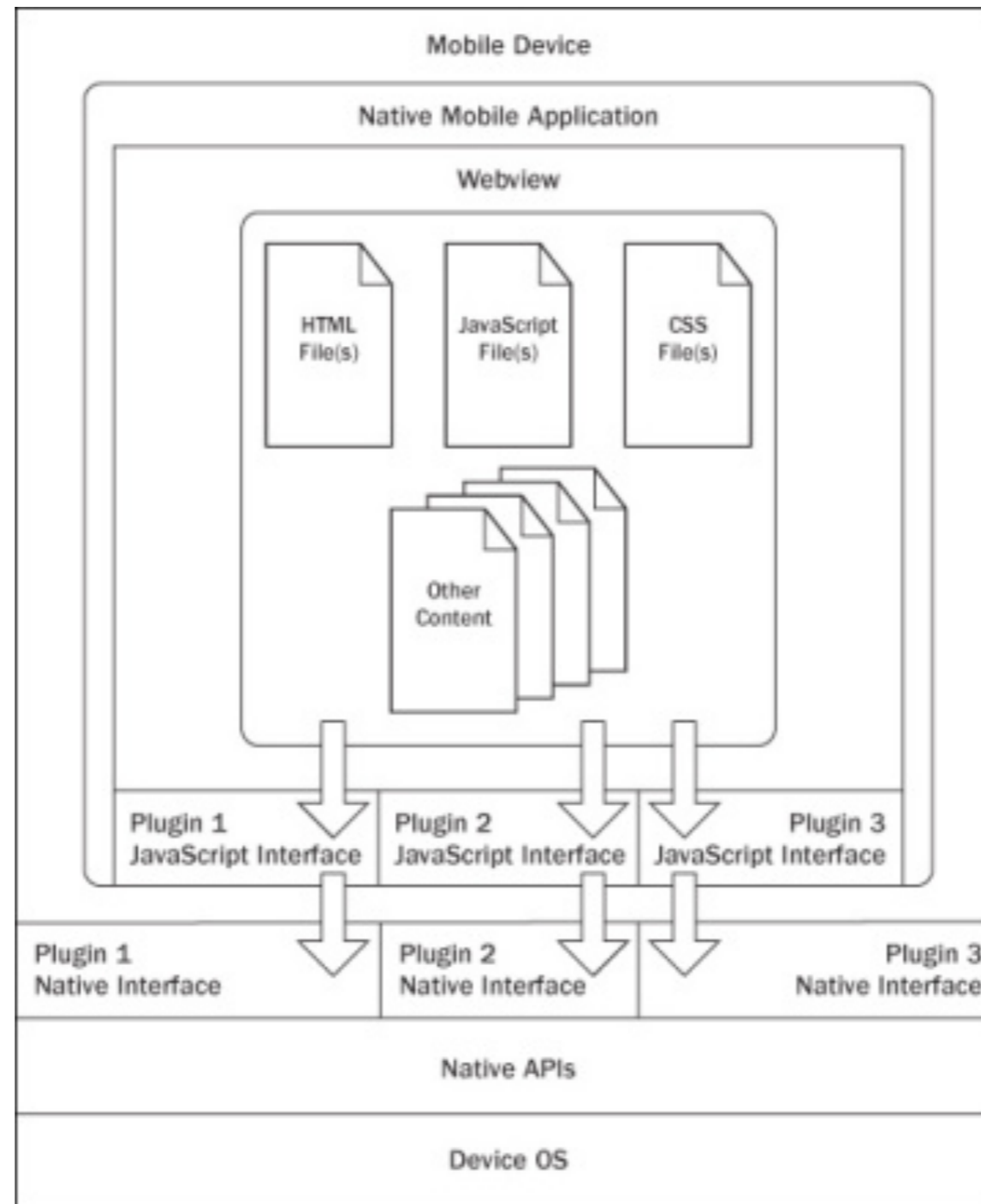
- ▶ **Web embarqué**

- ▶ **Cordova**
- ▶ Electron
- ▶ ReactNative

Cordova : détails pratiques

- ▶ Principe :
 - ▶ écrire une application Web et la faire compiler pour plusieurs OS mobiles
- ▶ OS pris en charge :
 - ▶ Android, iOS, BlackBerry, Windows Phone, Tizen, Firefox OS
- ▶ Permet d'accéder à certaines API natives
 - ▶ API JS spécifiques (pas toujours conformes aux specs du W3C)
 - ▶ Les SDKs sont nécessaires pour accéder aux API (donc les licences associées)
- ▶ Exécution en plein écran dans le navigateur natif
 - ▶ le coeur est toujours une WebView.
- ▶ PhoneGap (<http://phonegap.com>) est la surcouche de Cordova la plus populaire. Racheté en 2011 par Adobe.

Architecture



Cordova : détails pratiques

- ▶ Mode CLI (pas d'IDE)
 - ▶ Compilation, émulation
 - ▶ Utilisation de npm pour gérer les plugins (console améliorée, accès aux capteurs...)
- ▶ Possibilité de faire tourner l'application
 - ▶ Sur un appareil connecté en USB (nécessite les drivers)
 - ▶ Sur un émulateur (Android Virtual Device) à créer avec le AVD manager