

MY LUCKY DAY

03/2018



DEV MOBILE : NATIF VS HYBRIDE

1

NATIF VS HYBRIDE

Application Native, Hybride ou Web, comment faire son choix ?

Pourquoi My Lucky Day s'y intéresse ?

- Demandes régulières des clients
- Développement des compétences et du savoir-faire
- Réduction potentielle des coûts de développement et de maintenance

Les différentes solutions du marché

- Javascript/HTML > Transparent web browser (PWA)
- Javascript/HTML > Webview (Cordova, PhoneGap, Ionic)
- Javascript/HTML > Compilation native (Appcelerator)
- Langage spécifique > Compilation native (Xamarin, React Native, C++)
- Outils de conversion de langage > natif (J2ObjC, Go Mobile)



PWA (Progressive Web App)

Définition

- Application web-responsive, conçue en « Single Page App », avec un look&feel proche du natif

Les avantages

- Utilisation immédiate sans installation
- Accès à certaines fonctions natives (photos, vidéos, push)
- Gestion des déploiements : pas de dépôt sur les stores
- Fonctionnement « offline »

Les inconvénients

- Pas d'accès à toutes les fonctions natives
- Fragmentation des versions
- Problème de performance si webview
- Compatibilité non complète sous Safari iOS



Cordova, PhoneGap, Ionic

Définition

- Application avec un socle natif basique, codée en JS/HTML, avec un rendu dans une webView

Les avantages

- Langage Javascript très répandu
- Accès à certaines fonctions natives
- Communauté des développeurs « web » compatible

Les inconvénients

- Performances
- Dépendance aux mises à jour et aux évolutions des outils
- UI nécessitant quand même un code spécifique suivant l'OS



Appcelerator

Définition

- Application codée en JS/HTML, convertie en code natif et compilée pour chaque OS cible

Les avantages

- Langage Javascript très répandu
- Accès aux fonctions natives
- Communauté des développeurs « web » compatible
- Performances du natif

Les inconvénients

- Framework de développement spécifique
- UI nécessitant un code spécifique suivant l'OS

Framework en abandon (bugs, instabilité du code généré)



Xamarin

Définition

- Application codée en C#, compilée en natif pour chaque OS cible

Les avantages

- Langage mature
- Accès aux fonctions natives
- Performances du natif

Les inconvénients

- Framework de développement spécifique
- Difficulté de trouver des développeurs expérimentés (plutôt C# back-end)
- UI nécessitant un code spécifique suivant l'OS

Testé et abandonné par EuroTunnel (% code mutualisé faible, instabilité FWK)



React Native

Définition

- Application codée en React (Javascript) , compilée en natif pour chaque OS cible

Les avantages

- Créé par Facebook
- Langage Javascript très répandu
- Logique UI et « métier » mutualisée
- Communauté des développeurs Web React
- Rechargement « à chaud »
- Performances du natif

Les inconvénients

- Framework de compilation spécifique
- UI nécessitant un code spécifique suivant l'OS (« Learn once, write anywhere »)

Prise en main complexe, pas d'interface pour construire les écrans



C++

Définition

- Partie de code en C++, compatible XCode et Android Studio

Les avantages

- Langage mature
- Code mutualisé

Les inconvénients

- Interfaçage avec le langage natif (JNI, Djinni)
- Disponibilité des développeurs C++
- Ne concerne pas l'UI, nécessité d'un code natif par OS



J2ObjC

Définition

- Convertisseur Java > Objective C

Les avantages

- Performances du natif

Les inconvénients

- Qualité de la conversion
- Objective C en cours de remplacement par le Swift
- Pas de conversion de l'UI
- Conversion à chaque modification



Go Mobile

Définition

- Langage Go, converti en Java et ObjC en mode natif

Les avantages

- Développé par Google
- Nombreuses API
- Langage unique

Les inconvénients

- Langage spécifique
- Qualité de la conversion
- Pas de conversion de l'UI
- Conversion à chaque modification



Conclusion ?

- Pas d'outil « miracle »
- La plupart des dévs-mobiles/clients rencontrés a fait le choix du natif

Quel outil choisir (si choix de l'hybride) ?

- React Native en priorité !
- Xamarin en 2^{ème} choix

Quand faire de l'hybride ?

- Quand le code « métier » est conséquent (algorithmes complexes, traitement de données, etc...)
- Quand la part d'UI/UX est inférieure à 40% du projet

