

Licence 1

Informatique, Développement d'Application

Cours : Introduction aux technologies mobile



Séquence 5: Approches de développement

Objectifs

- A l'issue de cette séquence, vous serez en mesure de :
 - Décrire les différentes approches de développement mobile
 - Citer les avantages et inconvénients de chaque approche
 - Donner quelques chiffres clés sur les application et systèmes d'exploitation mobiles
 - Expliquer les défis du développement d'applications mobiles



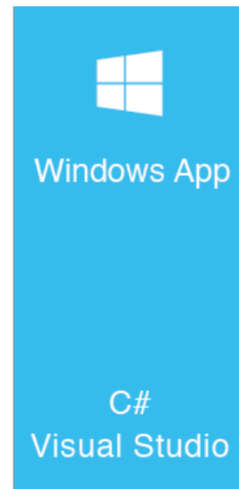
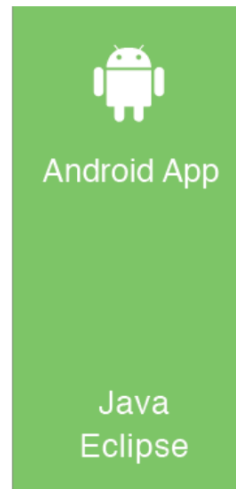
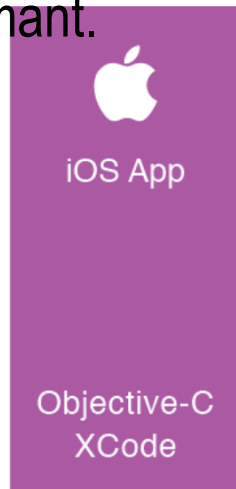
Différentes approches pour le Développement Mobile

- Plusieurs options pour développer une application mobile :
 - **Développement natif** (Objective C ou swift, java, C#...)
 - **Solution cross plateforme** : solutions comme Xamarin permettent de construire des applications iOS, Android et Windows mobile dans un environnement de développement Microsoft (C#)
 - **Solutions hybrides** : Des technologies de type Phonegap ou Cordova permettent d'embarquer une web view dans une application native.
 - **Développement web**: utilisant le HTML5, CSS, JavaScript.



L'approche native: L'Approche en Silo

- **Les applications natives** sont spécifiques à une plate-forme mobile donnée (iOS ou Android) utilisant les outils de développement et le langage pris en charge par la plate-forme (par exemple Xcode et Objective-C avec iOS, Eclipse et Java avec Android). Les applications natives ont un design plus apprécié et fonctionnent plus performant.
- **Créer l'application plusieurs fois**
 - Plus de développeurs
 - Plus de maintenance
 - Différents environnements



L'approche native: L'Approche en Silo

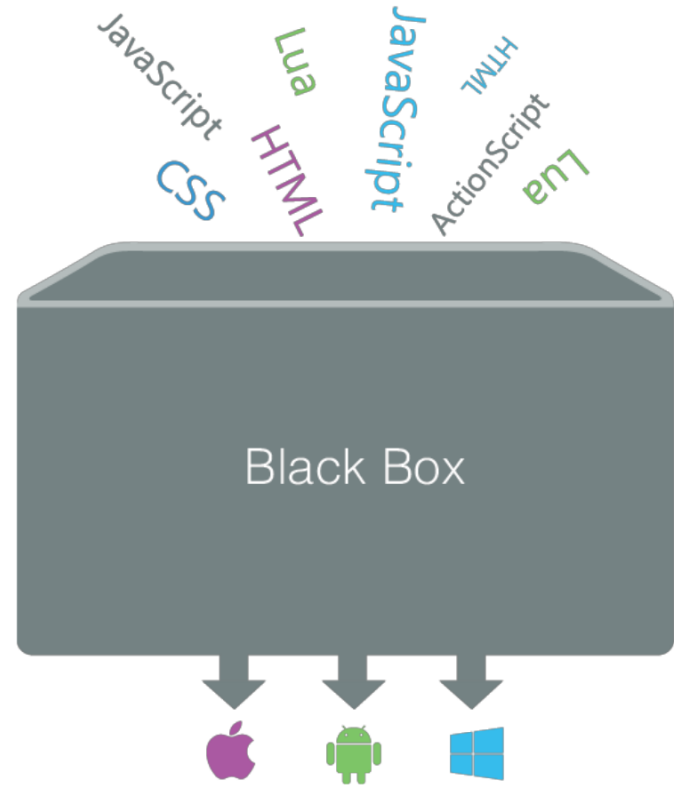
- **Avantages :**
 - Chaque système d'exploitation a sa version native – Possibilité de faire appel aux fonctionnalités de l'appareil utilisé (telle que la camera, la géolocalisation, le gyroscope ...) et ainsi les exploiter de manière infinie...
 - Profiter d'une visibilité sur les stores d'applications et optimisation du référencement – Accès aux APIs natives – Plus simple à développer et plus d'outils à disposition et de support en cas de problème – Intégration à la facturation des stores si l'objectif est de rentabiliser le produit (Apps payantes) – Facilité à se conformer aux ergonomies spécifiques de chaque système d'exploitation afin d'avoir la meilleure expérience utilisateur possible.

L'approche native: L'Approche en Silo

- **Inconvénients :**
 - Le cout de développement est important car il faut développer une version spécifique pour chaque système d'exploitation – Les performances sont très souvent meilleures en natif – Toutes les mises à jour nécessiteront un coût de développement supplémentaire multiplié par le nombre de plate-formes utilisées

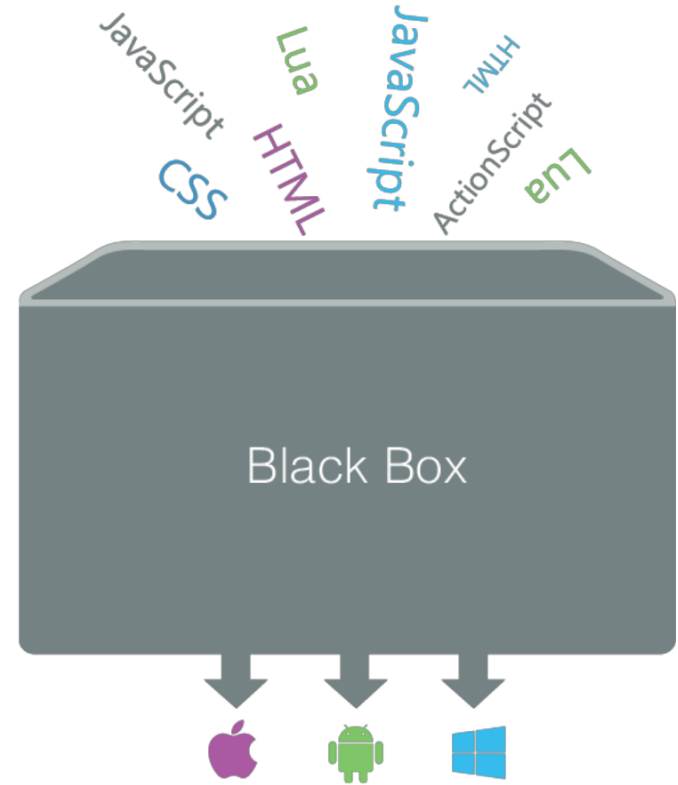
L'approche web: Write Once, Run Anywhere

- Utilise des technologies Web standard, généralement HTML5, JavaScript et CSS. Développée à grand renfort de codes javascript et autres boites à outils de type [AngularJS](#) ou [React](#)
- Approche de développement mobile permettant de créer des applications mobiles multiplateformes fonctionnant sur plusieurs appareils.



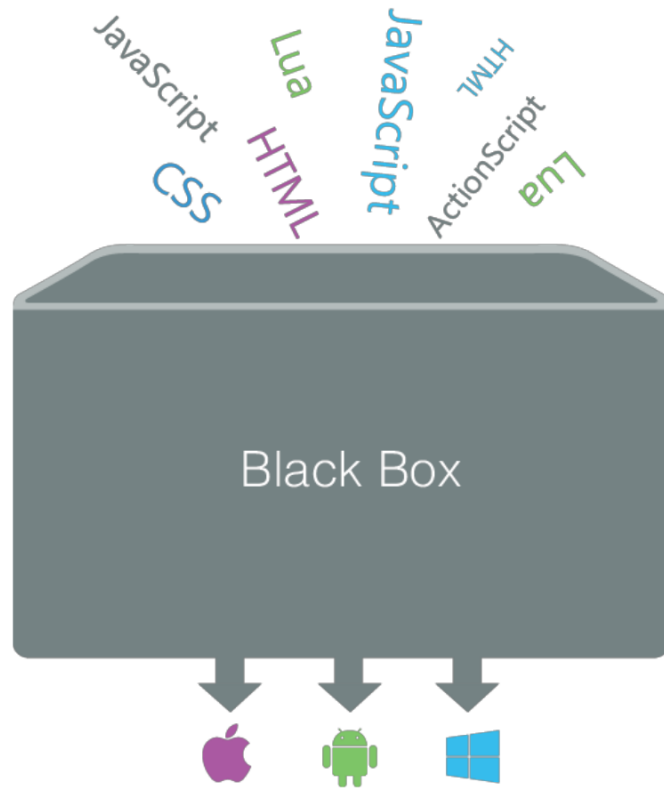
L'approche web: Write Once, Run Anywhere

- **Avantages :**
 - Développement web classique donc très bien maîtrisé par les développeurs. – Utilisation de HTML5, CSS3, JavaScript et de frameworks – Responsive design pour un affichage optimal et adapté aux différents supports (smartphones, tablettes, desktop...) – Facilité de mise à jour du contenu



L'approche web: Write Once, Run Anywhere

- **Inconvénients :**
 - Nécessité d'avoir une connexion internet pour accéder au contenu
 - Pas de visibilité sur les stores d'application
 - Accessible via le navigateur.
 - Pas d'icone d'application
 - Ne s'exécute pas en plein écran



Application native vs. Application Web

- Développement spécifique selon le navigateur web
- Manque d'intégration avec l'OS(capteurs, partage, etc)
- Expérience utilisateur
- Performance « dégradées »
- Beaucoup ont essayé le web pour finalement revenir au natif

Facebook Web



Facebook natif



Approche hybride

- Une application hybride est comme une application Web, principalement construite à l'aide de HTML5 et de JavaScript, puis encapsulée dans un conteneur natif mince qui donne accès aux fonctionnalités de la plateforme native.
- Il existe de nombreux frameworks mobiles hybrides tels que Ionic, NativeScript, React Native, Xamarin, PhoneGap etc. PhoneGap est un exemple de conteneur le plus populaire pour créer des applications mobiles hybrides.
- Le développement hybride combine le meilleur (ou le pire) des mondes natif et web.

Approche multiplateformes

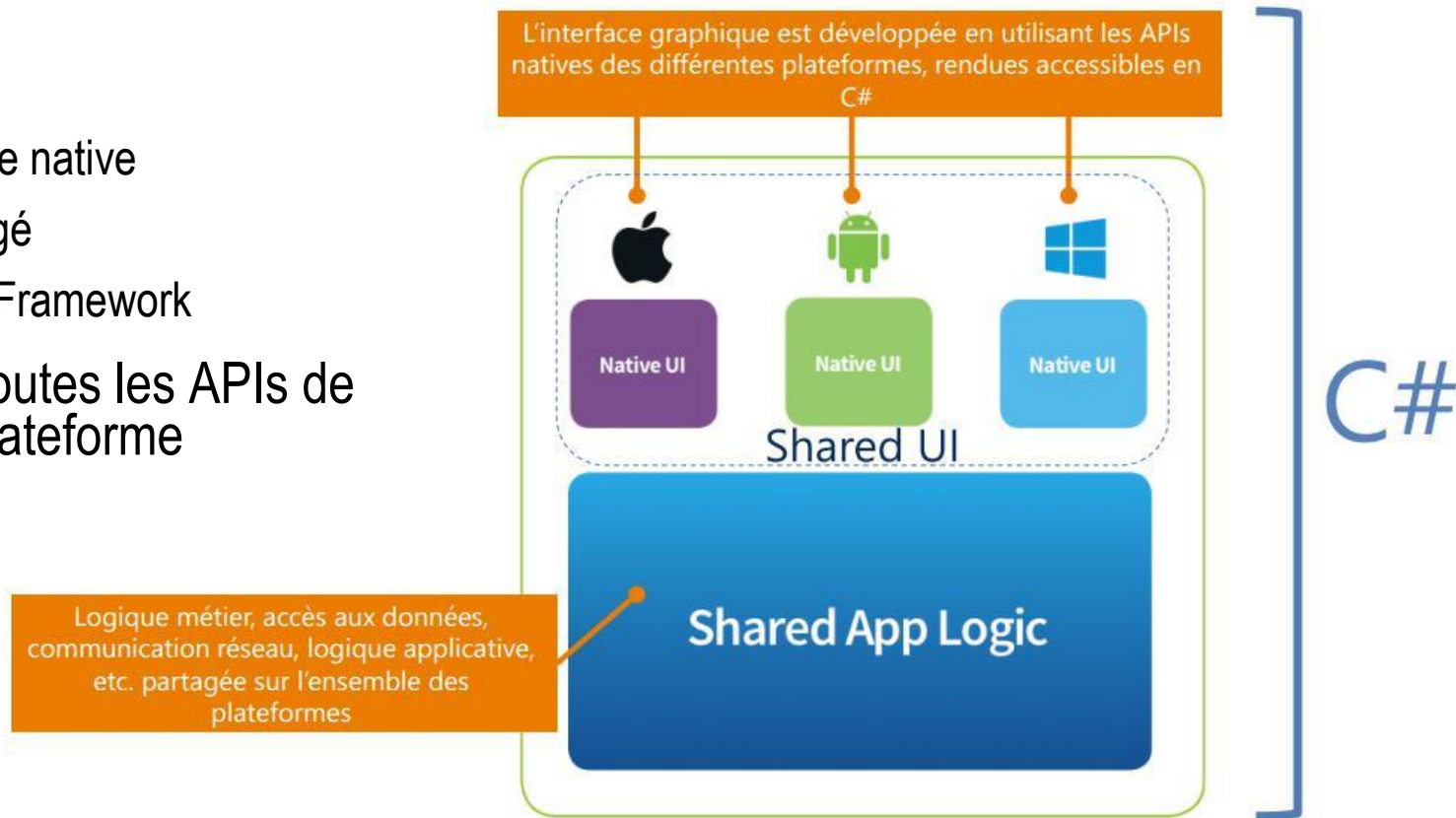
- Créer des applications Android, iOS et Windows Phone - de la logique métier à l'interface utilisateur - avec un code presque 100% commun.
- Le marché des applications multiplateformes devrait atteindre 7,5 millions de dollars d'ici 2018, et le nombre d'outils de développement multiplateforme est en hausse. Source <https://blog.octo.com/etat-de-lart-des-solutions-cross-platform-mobile/>

Approche multiplateformes

- **Avantages :**
 - Chaque système d'exploitation a sa version native – Profiter d'une visibilité sur les stores d'applications – Les modifications et mises à jour seront effectives sur chaque plate-forme – Economie de budget – Idéale pour des applications de jeu en 2D ou en 3D
- **Inconvénients :**
 - Problème d'accès aux APIs – Problème de validation des applications par les stores – Difficile à maintenir et à faire évoluer – Toutes les fonctionnalités des appareils ne peuvent pas encore être exploitées – Problème d'ergonomie. UX et UI non optimisées

Approche multiplateformes: L'approche de Xamarin

- UI native
- Performance native
- Code partagé
- C# & .NET Framework
- Accès à toutes les APIs de chaque plateforme



Défis du développement d'applications mobiles

- Un paysage de fournisseurs concurrentiel et fluide (Apple, le consortium Android, notamment Amazon, RIM, HP) signifie que les applications doivent être multiplateformes pour une large adoption
- Pas d'appareil "standard" (qu'en est-il des appareils iOS, Windows Phone?)
- Faible entrée de bande passante (dans la plupart des cas, qu'en est-il des tablettes?)
- Taille d'écran limitée (tablettes?)
- Un manque de fiabilité de la connectivité et du périphérique (accès réseau, alimentation, lumière ambiante, bruit, au moins pour l'instant)
- Intégration des compromis avec les services cloud et d'entreprise

Supports du développement d'applications mobiles

- Langages orientés objet de troisième génération (iOS - Objective C ou Swift, Android - Java, Windows Phone - C #)
- Langages de script (JavaScript, Ruby)
- Structures multiplateformes - Titane, RhoMobile, Xamarin, PhoneGap
- Intégré dans des "frameworks" spécifiques pour le développement d'applications mobiles

Points clés

- iOS et Android ont participé à réduire la **fragmentation de l'industrie mobile** et ont attiré ainsi rapidement de nombreux développeurs
- iOS et Android sont à l'origine de l'**essor des smartphones et des marchés de l'Internet mobile**, tandis que les tentatives précédentes de type WAP et de « Walled gardens » ont échoué.
- Ce contexte a été favorable au développement d'**une économie dynamique des applications mobiles en Europe**, avec près de 1,5 million d'emplois et 13 milliards EUR de revenus générés (applications payantes, publicités) en 2016.

<https://fr.idate.org/evolution-de-lecosysteme-mobile/>

Points clés

- De nombreuses **success stories** ont été rendues possibles indépendamment de iOS et Android, notamment dans le domaine des jeux sur mobile (ex. Angry Bird de Rovio) ou dans l'écosystème de Facebook.
- **La concurrence entre les acteurs des terminaux a été renforcée**, plus intense qu'avant le lancement de iOS et Android. Android en particulier ayant notamment favorisé l'arrivée de nouveaux entrants grâce à un abaissement des prix.
- **Les développeurs utilisent couramment de multiple plateformes**, notamment en tirant profit des outils cross-platform.

<https://fr.idate.org/evolution-de-lecosysteme-mobile/>

A retenir...

- **Il existe plusieurs approches de développement d'une application mobile.**
- Les applications natives ont un design plus apprécié et un fonctionnement plus performant. Dans le contexte d'un développement utilisant l'approche native, il va falloir créer l'application plusieurs fois, ce qui nécessite donc plus de développeurs, plus de maintenance sur différents environnements.
- **Inconvénients de l'approche native:** Toutes les mises à jour nécessiteront un coût de développement supplémentaire multiplié par le nombre de plateformes utilisées.

A retenir...

- Il existe de nombreux frameworks mobiles hybrides tels que Ionic, NativeScript, React Native, Xamarin, PhoneGap etc. PhoneGap est un exemple de conteneur le plus populaire pour créer des applications mobiles hybrides. Cette approche permet de créer des applications Android, iOS et Windows Phone - de la logique métier à l'interface utilisateur - avec un code presque 100% commun.
- Le marché des applications multiplateformes devrait atteindre 7,5 millions de dollars d'ici 2018, et le nombre d'outils de développement multiplateforme est en hausse.

A retenir...

- **iOS et Android ont participé à réduire la fragmentation de l'industrie mobile et ont attiré ainsi rapidement de nombreux développeurs**
- Ce contexte a été favorable au développement d'une économie dynamique des applications mobiles en Europe, avec près de 1,5 million d'emplois et 13 milliards EUR de revenus générés en 2016.
- De nombreux success stories ont été rendus possibles indépendamment de iOS et Android, notamment dans le domaine des jeux sur mobile ou dans l'écosystème de Facebook.
- La concurrence entre les acteurs des terminaux a été renforcée, plus intense qu'avant le lancement de iOS et Android. Android en particulier ayant favorisé l'arrivée de nouveaux entrants grâce à un abaissement des prix.

Travaux dirigés

- Qu'est-ce qu'une approche de développement ?
- Lister les approches de développement mobile
- Décrire les approches de développement mobile
- Décrire brièvement chaque approche
- Lister 03 avantages et 03 inconvénients de chaque approche
- Expliquer l'approche multiplateforme et donner quelques raisons liées à son succès
- Lister 5 défis du développement mobile
- Quels sont les supports du développement d'applications mobiles ?

Travaux dirigés

- Quelles sont les motivations liées au développement Cross-Platform ?
- Qu'est-ce-que Xamarin?
- Qu'est qui a été à l'origine de l'essor des marchés de l'Internet mobile ?
- Citer une des applications mobiles de jeux qui connaît un success-story actuellement au Sénégal
- Faire un résumé des points clés sur les plateformes mobiles

Bibliographie et Webographie

- <https://developer.xamarin.com/guides/>
- <https://docs.microsoft.com/en-us/aspnet/core/>
- <http://scr.sad.supinfo.com/articles/single/3532-construire-une-interface-utilisateur-avec-xamarin#idm140445263718048>
- <http://www.e-naxos.com/Blog/post/Strategie-de-developpement-Cross-Platform-Partie-2.aspx>
- ARTP, Rapport sur le marché des télécommunications au Sénégal en 2016, http://www.artpsenegal.net/anciensite/images/documents/Rapport_Observatoire_T2_2016.pdf

Bibliographie et Webographie

- Mobile Applications: Architecture, Design, and Development: Architecture, Design, and Development (Hewlett-Packard Professional Books) Apr 2004 by Valentino Lee (Author), Heather Schneider (Contributor), Robbie Schell (Contributor)
- Mobile Design and Development Brian Fling, Publisher: O'Reilly
- Programming the Mobile Web Maximiliano Firtman, Publisher: O'Reilly Media