

TypeScript



INF 3511 Programmation des Mobiles:

Environnements Mobiles

Licence Informatique Option Génie Logiciel

Année Universitaire 2018-2019

Ousmane SALL

Maître de Conférences CAMES

Université de THIES - UFR Sciences et Technologies -Département Informatique

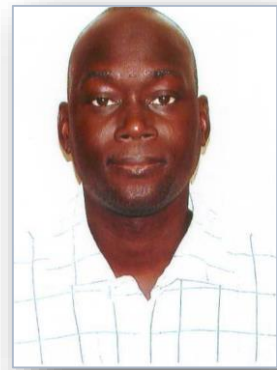
Partie 4:

Xamarin pour le développement d'Applications Mobiles



APACHE
CORDOVA™

A propos de moi

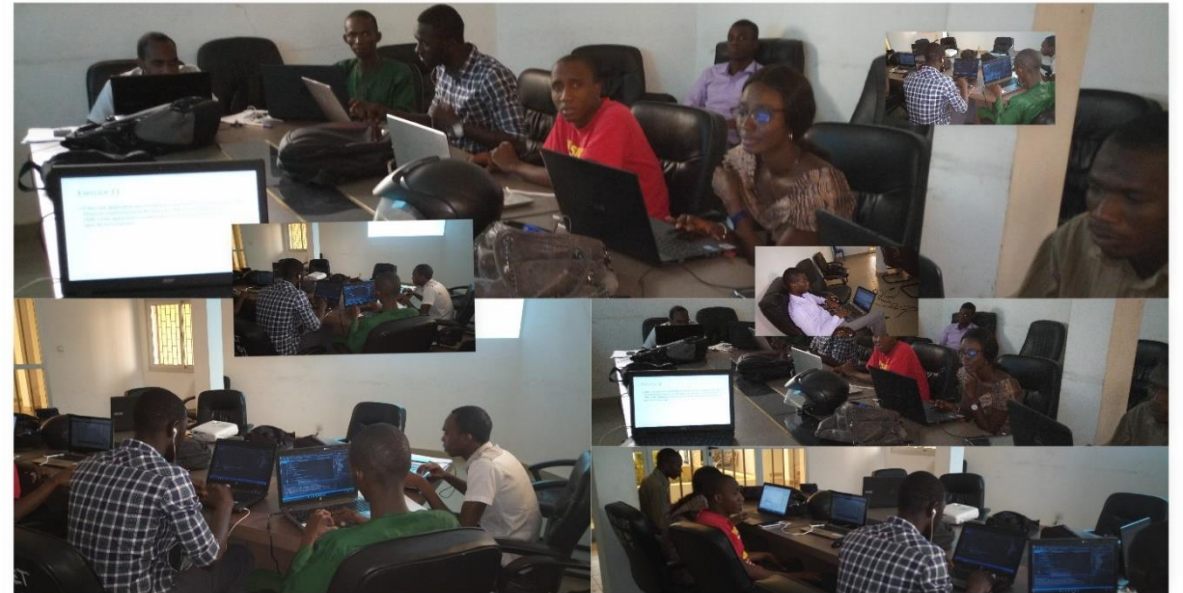


- Enseignant-Chercheur à l'UFR SET- Université de THIES <http://sites.univ-thies.sn/osall751/>
- Enseignements:
 - Algorithmique et Programmation(C, Java, PHP)
 - Programmation WEB dynamique(HTML 5 CSS, PHP, MySQL, CMS,...)
 - Programmation Java
 - Programmation JavaEE, JSF, Spring
 - Technologies Mobiles Android, Xamarin, Ionic
 - Programmation .Net, C#
 - Gestion de Projet Informatique
 - Génie Logiciel
- Contact:
 - osall@univ-thies.sn
 - UFR SET, Université de THIES -Dpt Informatique, BP 967 THIES.



Une sagesse chinoise...

« J'écoute et j'oublie; je lis et je comprends; je fais et j'apprends »
[Proverbe chinois]



Contenu

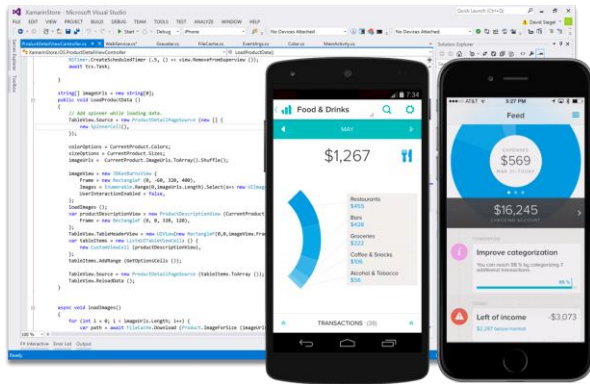
1. Généralités sur les Technologies Mobiles
2. Périphériques et Systèmes d'exploitations mobiles
3. Approches de développement mobile
- 4. Xamarin pour le développement d'Applications Mobiles**
5. Développement d'Applications Mobiles Hybrides avec Cordova et le framework Ionic 4
6. Développement d'Applications Natives avec Android



Xamarin

- Comme évoqué précédemment, Xamarin est un framework .Net de développement d'applications mobiles natives ou cross-plateforme, fondé en 2011 et racheté par Microsoft en 2016. L'avantage de Xamarin est de fournir une facilité de communication avec les produits microsoft (Azure, Sql server, etc).
- Cette technologie permet de développer des applications mobiles pour les plateformes suivantes :
 - Android
 - iOS
 - Windows Universal (qui comprend Windows Phone)
- Le développement Xamarin comprend deux approches : le développement natif, et le développement cross-platform.

Xamarin est un moyen de



Créer



Tester

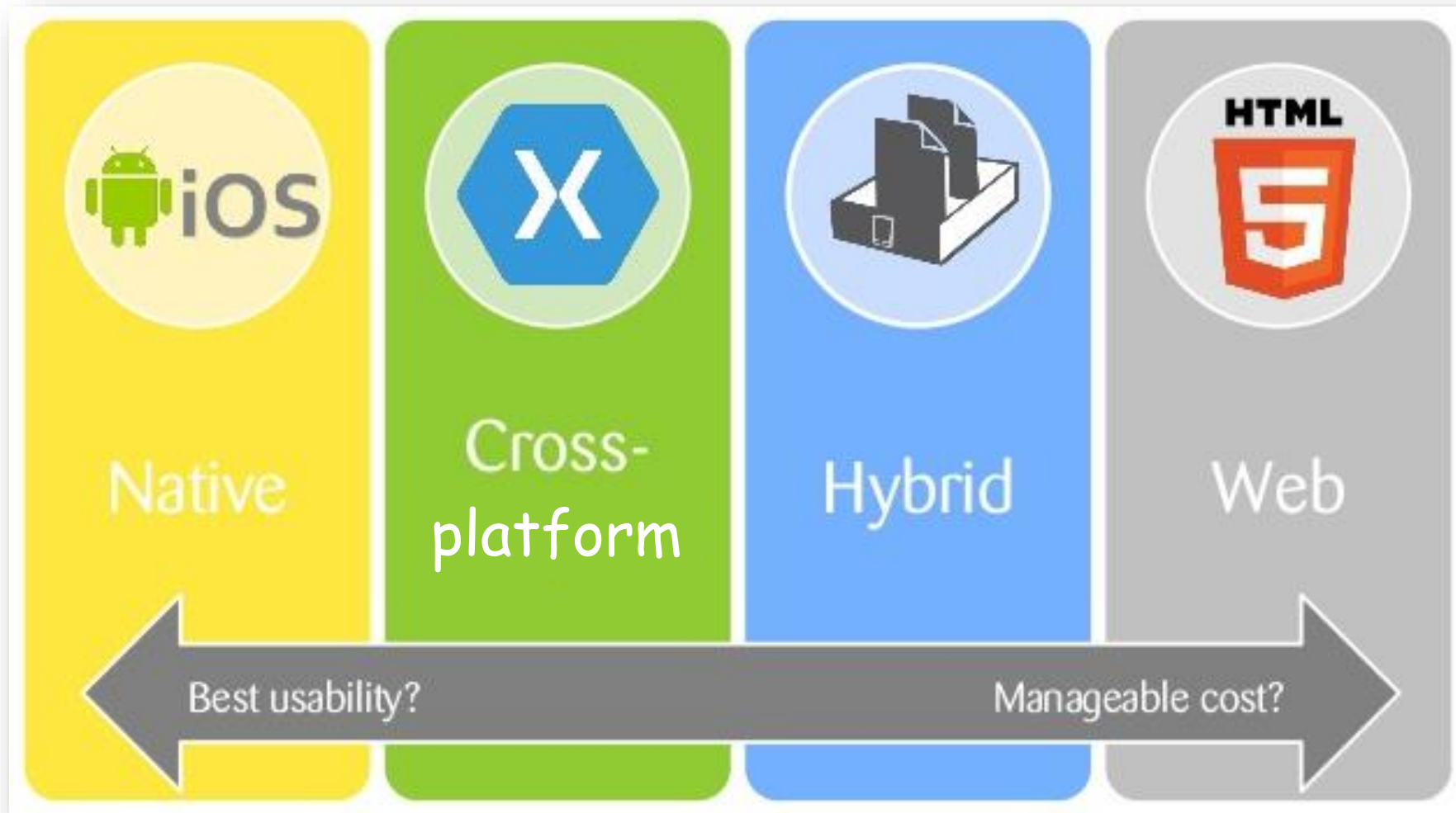


Monitorer

Xamarin	
	
Création	mai 2011
Fondateurs	Des anciens membres de Novell travaillant sur Mono ¹
Personnages clés	Miguel de Icaza, Nat Friedman
Forme juridique	Inc.
Siège social	 Cambridge, Massachusetts (États-Unis)
Direction	Nat Friedman (CEO) Miguel de Icaza (CTO) Joseph Hill (COO) ²
Actionnaires	Microsoft 
Activité	Logiciel
Produits	Mono (stewardship et support), Mono for Android, MonoTouch
Société mère	Microsoft
Effectif	50 ³
Site web	www.xamarin.com  [archive] 
modifier · modifier le code · voir wikidata	

Différentes approches pour le Développement Mobile: rappel

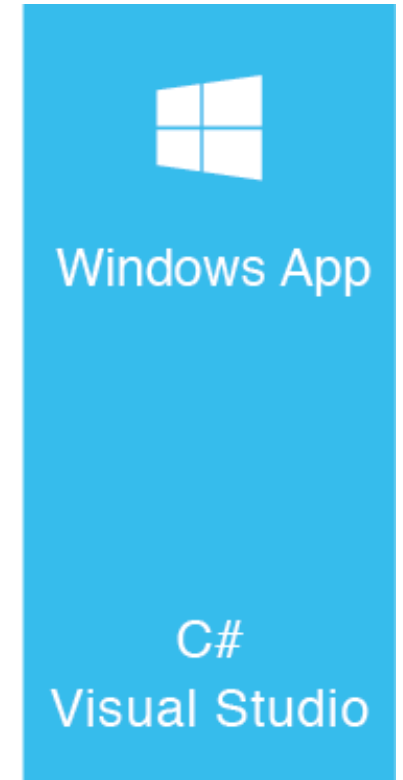
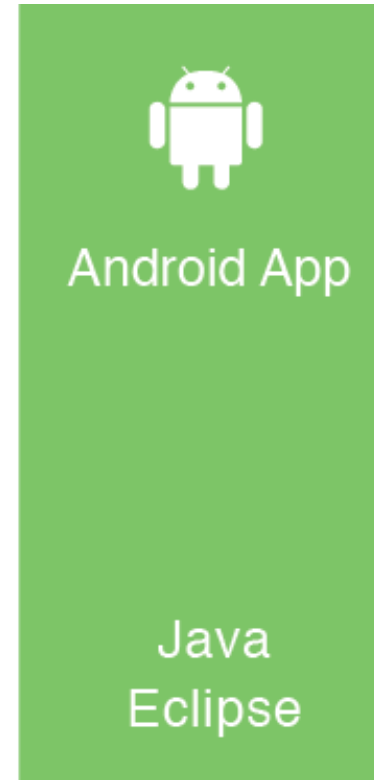
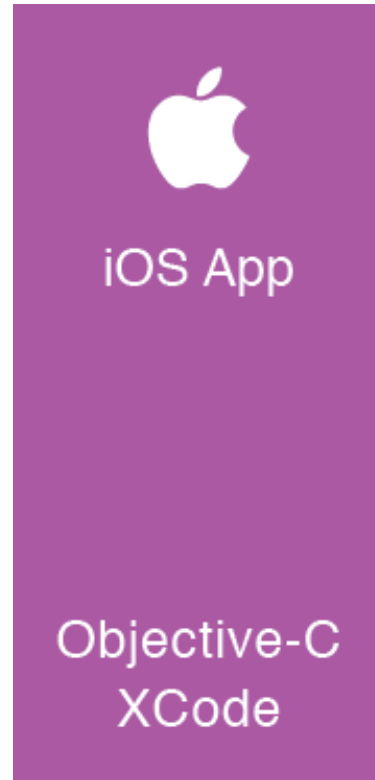
Approches pour le Développement Mobile



L'approche native: Silo Approach

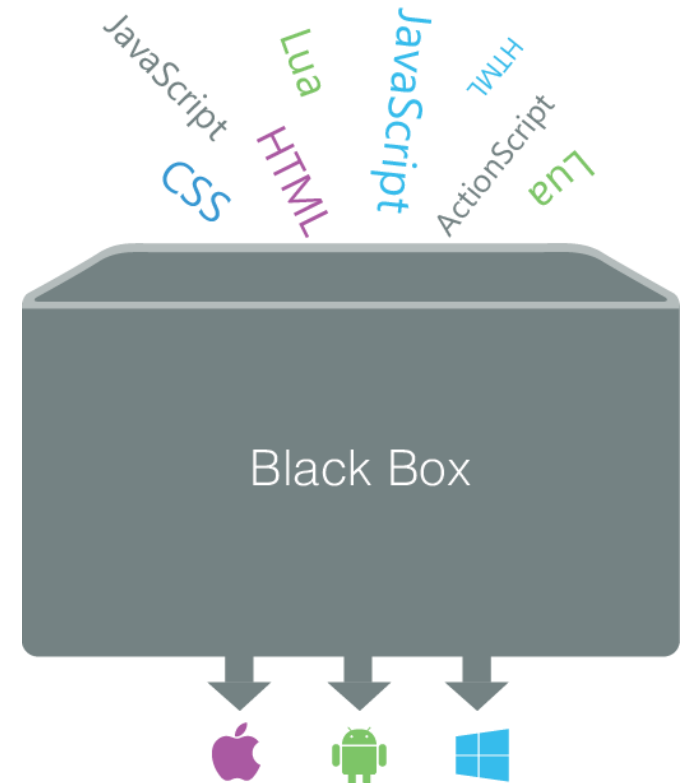
- Créer l'application plusieurs fois

- Plus de développeurs
- Plus de maintenance
- Différents environnements



L'approche Write Once, Run Anywhere

- Nécessaire d'avoir une connexion Internet en tout temps. Si l'utilisateur n'est pas connecté à Internet avec une utilisation illimitée des données, l'application peut consommer une quantité considérable de données en fonction de l'application.
- La vitesse de connexion Internet peut également limiter l'expérience de l'application.
- L'application peut faire un certain usage du matériel du téléphone à travers diverses interfaces disponibles dans le navigateur web mais est souvent limitée par ce que le fournisseur du navigateur a implémenté



L'approche hybride

- Application utilisant le [navigateur web](#) intégré du support ([Smartphone](#) ou [tablette](#)) et les technologies Web ([HTML](#), CSS et Javascript) pour fonctionner sur différents OS ([iOS](#), [Android](#), [Windows Phone](#), etc.).
- Une telle application utilise les fonctionnalités natives des Smartphones et peut être distribuée sur les plateformes d'applications telles que [l'AppStore](#), le [Google Play](#), etc.



L'approche hybride: inconvénients

- le code source est mélangé et la compilation se fait en de multiples étapes, ce qui peut fragiliser le fonctionnement de l'application mobile et rendre la maintenance longue et complexe ;
- Tout n'est pas réalisable en terme de fonctionnalités avec cette technologie, il faut s'assurer en amont de la compatibilité de votre projet (et de ses évolutions futures !) avec les technologies hybrides. Par exemple l'application peut ne pas supporter toutes les fonctionnalités.
- Des 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 ;
- Enfin, en utilisant une application hybride, nous devenons dépendant d'un logiciel tiers – en l'occurrence Cordova – et donc de sa compatibilité avec les nouvelles versions d'OS.

<http://www.mobizel.com/2015/08/developpement-dune-application-mobile-hybride-33/>

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 cross-platform

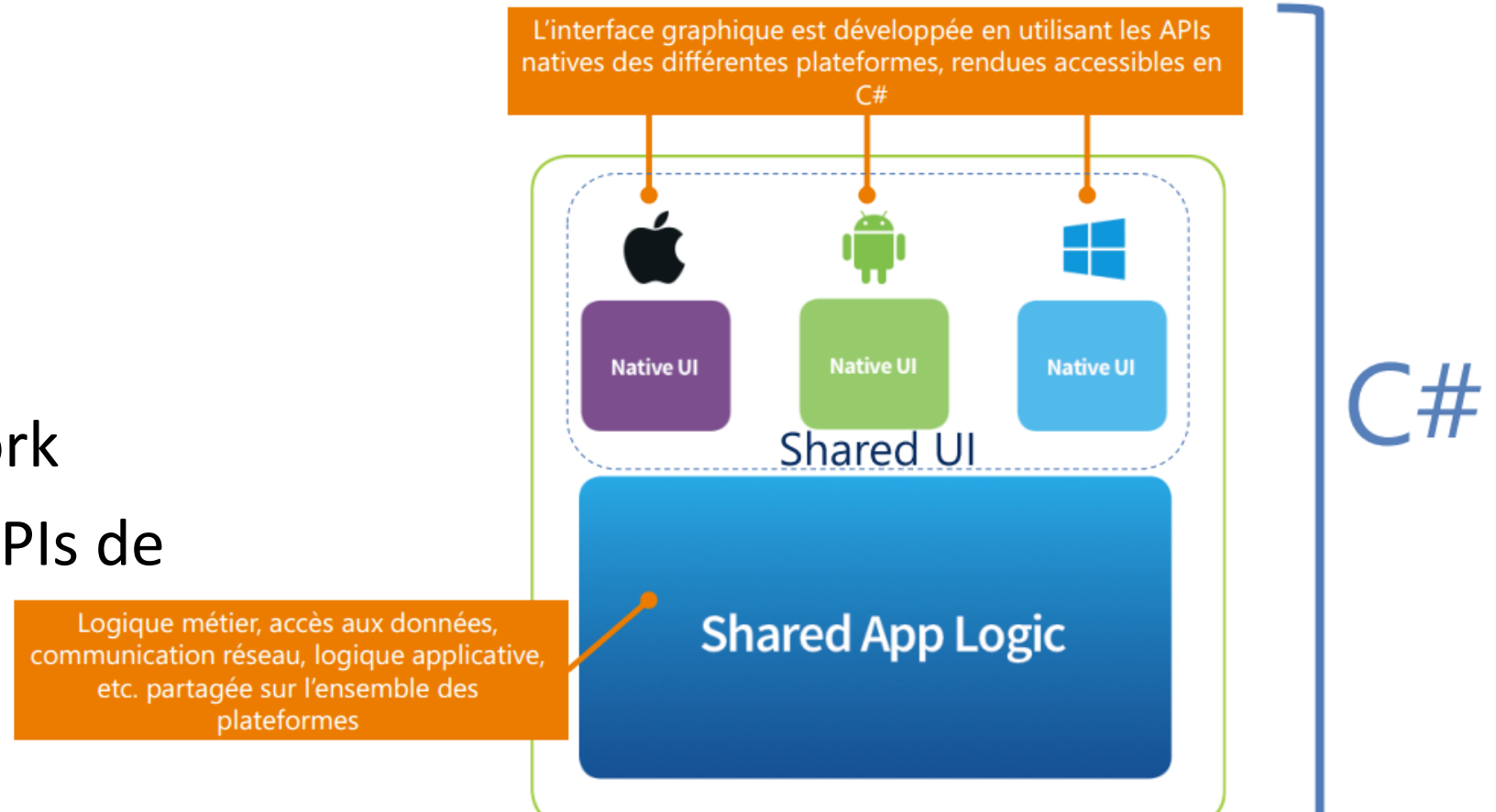
Quatre principales approches multiplateformes: Web, Hybrid, Interpreter and Cross-Compiler



L'approche de Xamarin

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

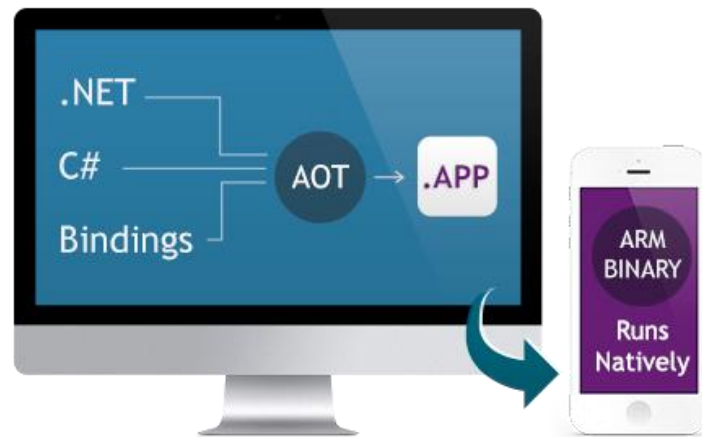
Bâtir une app native iOS, Android et Windows à partir d'une base de code C# unique.



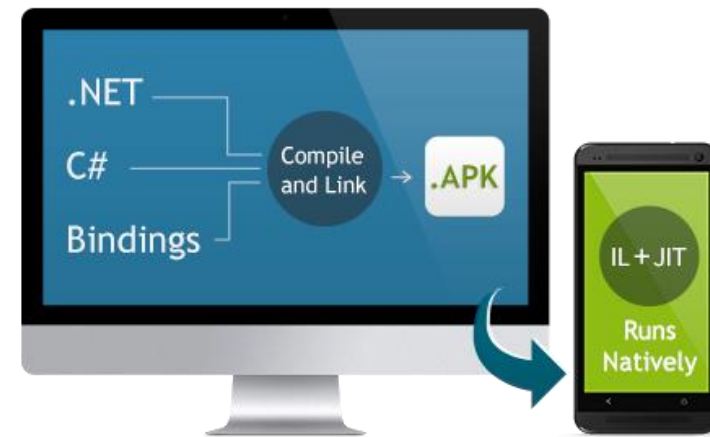
Xamarin recently introduced Xamarin.Forms a new library for cross platform user interface. We will touch up on this later, but this enables you to be highly productive, share code, but build out UI on each platform and access platform APIs. With Xamarin.Forms you now have a nice Shared UI Code layer, but still access to platform APIs. You can start from native, pick a few screens, or start with forms, and replace with native later.

Performance native

Tout ce que vous pouvez faire en Objective-C, Swift ou Java peut être fait en C# avec Xamarin et Visual Studio.



Xamarin.iOS compile votre application en binaire ARM pour l'App Store d'Apple.



Xamarin.Android utilise un système Just In Time lors de l'exécution.

Il n'y a pas de compromis sur les performances.
Les applications Xamarin ont une apparence native parce qu'elles sont natives.

Webography

- <https://ionicframework.com/>
- <https://openclassrooms.com/fr/courses/5098931-developpez-une-application-mobile-multiplateforme-avec-ionic-3>
- <https://developer.xamarin.com/guides/>
- <https://angular.io/>
- <http://typescriptlang.org>
- <http://www.e-naxos.com/Blog/post/Strategie-de-developpement-Cross-Platform-Partie-2.aspx>
- Tutos Xamarin sur Google et Youtube entre autres.