

Le développement cross plateforme avec Xamarin

Module 2 – Présentation de Xamarin



Objectifs

- Comprendre la vision multiplateforme proposée par Xamarin
- Comprendre l'architecture de Xamarin, ses forces et ses contraintes
- Pouvoir installer l'environnement de développement pour débiter un projet Xamarin

Vision du développement multiplateforme

- Le développement natif :
 - tout type d'applications
 - performances élevées
 - intégration totale à la plateforme

Le développement multiplateforme de type WebView :

- coûts, temps passés et compétences requises divisés par deux
- moins de performances
- intégration moins importante à la plateforme

Présentation de Xamarin

Vision du développement multiplateforme

Xamarin a pour ambition de concilier les deux approches

Proposer les performances et l'intégration du natif

Diminuer les coûts, le temps et les compétences requises pour le développement



Vision du développement multiplateforme

- La vision du développement multiplateforme avec Xamarin repose sur une règle simple :
 - Si un composant fonctionne de la même manière sur toutes les plateformes, il est écrit en multiplateforme
 - Calculs métiers
 - Contrôle de surface
 - Appels serveurs
 - Réaction à une interaction utilisateur
 - Si un composant doit s'adapter à la plateforme (visuel ou comportement), il est écrit en code natif
 - Vue
 - Composants bas niveau : accéléromètre, GPS, appareil photo

Vision du développement multiplateforme

Avec Xamarin on mutualise le code qui n'interagit pas directement avec la plateforme

On écrit en code natif le code qui nécessite d'être adhérent à la plateforme

Architecture de Xamarin

Mais comment écrire du code multiplateforme alors qu'Android et iOS n'utilisent pas les mêmes langages natifs ?

En utilisant un langage qui peut interpréter sur les deux plateformes !

Java ?

C++ ?

Python ?

Architecture de Xamarin

Xamarin utilise le langage C# qui peut être interprété sur Android, iOS et Windows



Architecture de Xamarin

Avec C#, Xamarin permet d'écrire le code multiplateforme dans un seul et même langage

Xamarin propose d'écrire également le code natif en C#

L'ensemble des SDK natifs sont accessibles

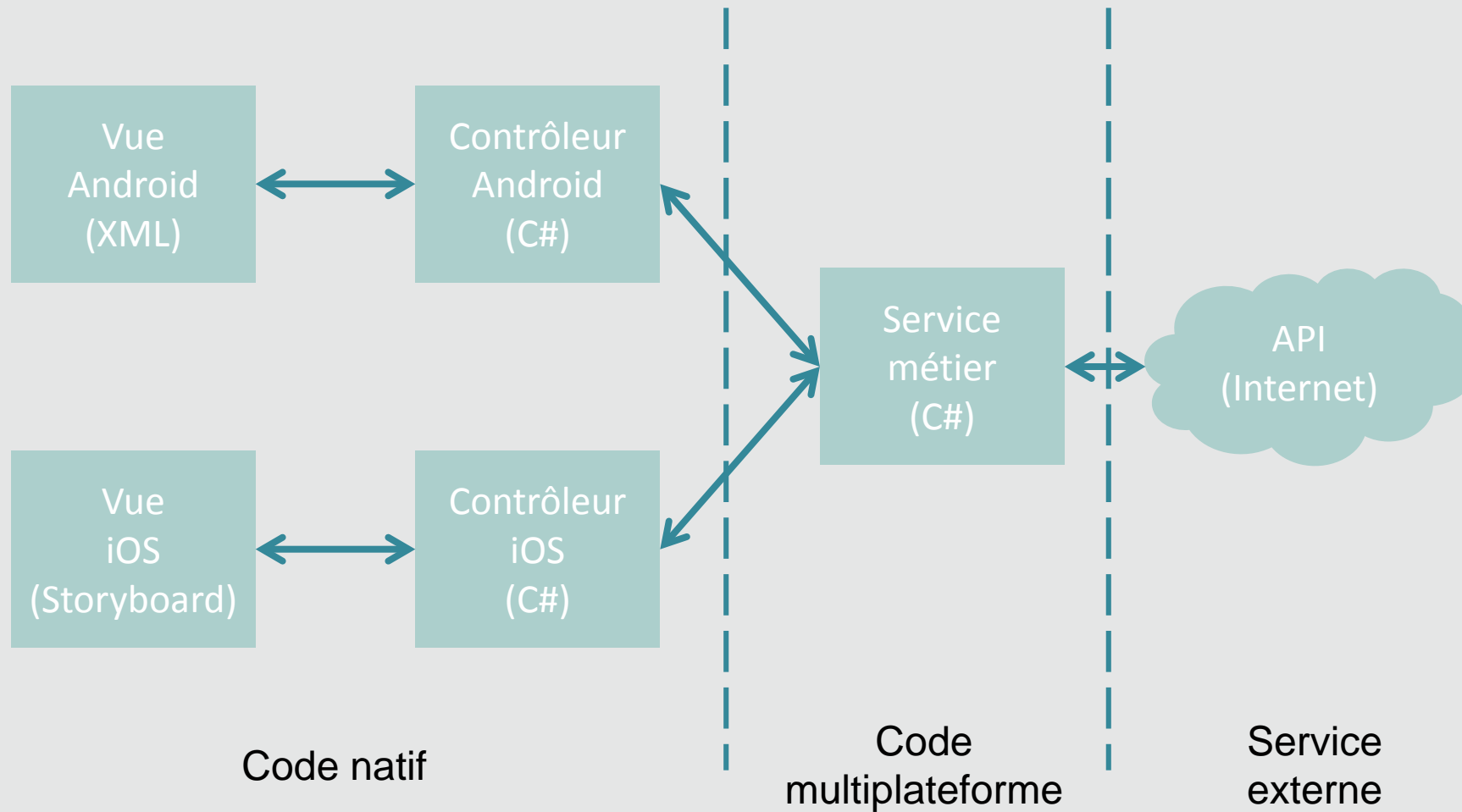
Architecture de Xamarin

Xamarin repose sur C# pour proposer le développement des composants natifs comme des composants multiplateformes

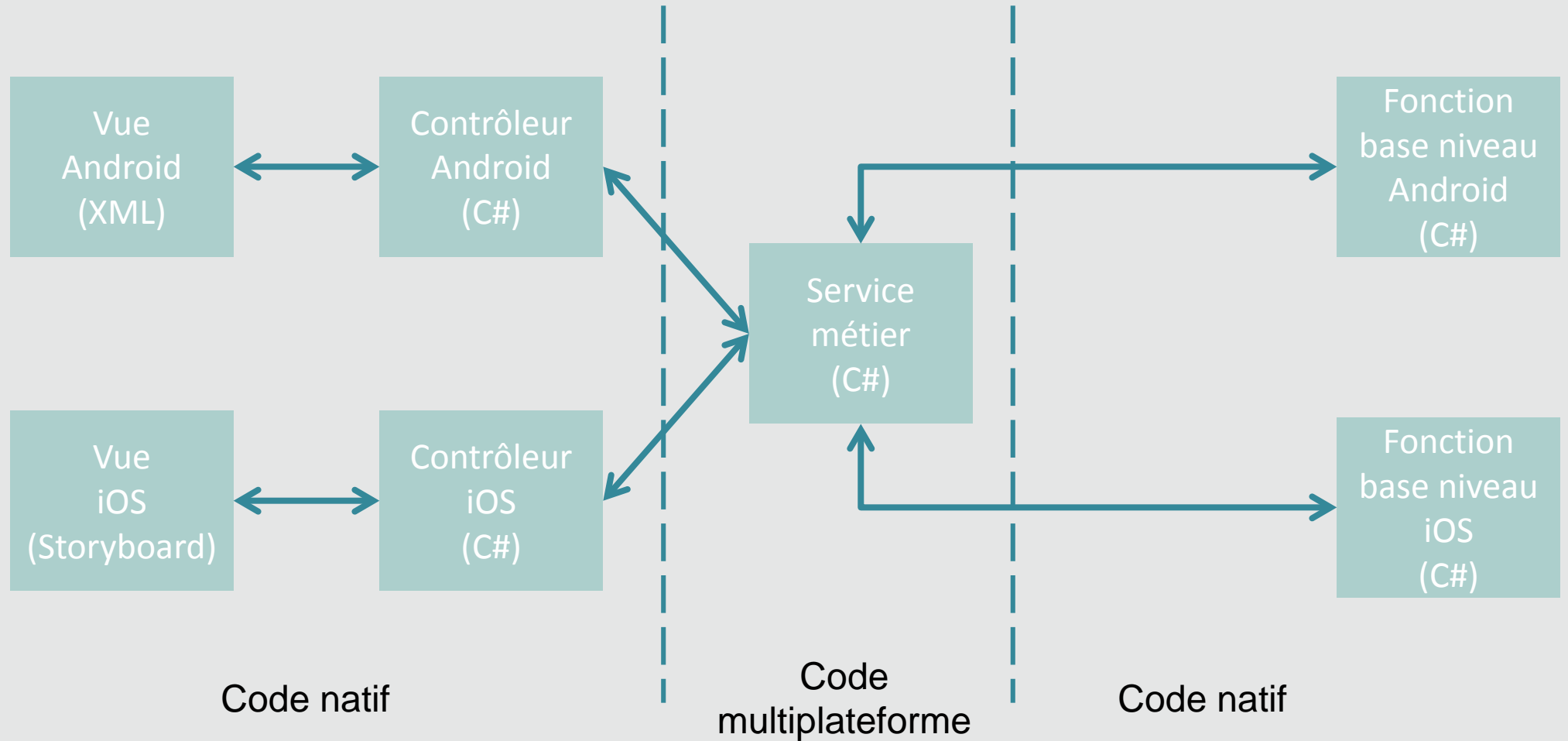
La clé de voûte de ce fonctionnement est le compilateur Mono qui permet d'interpréter C# sur toutes les plateformes cibles

Ce choix n'est pas dû au hasard car les créateurs de Xamarin ont participé à la conception et au développement de Mono !

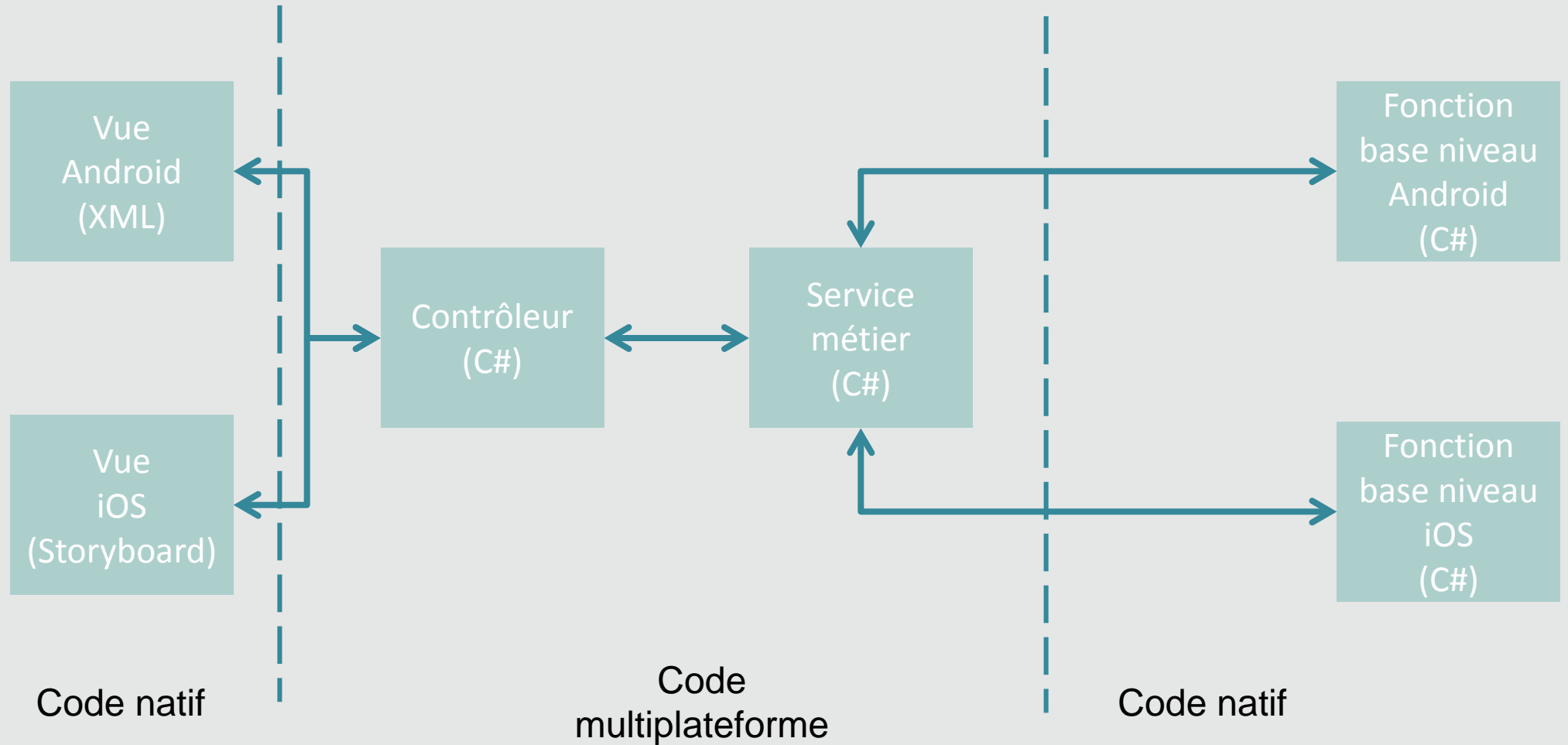
Architecture de Xamarin



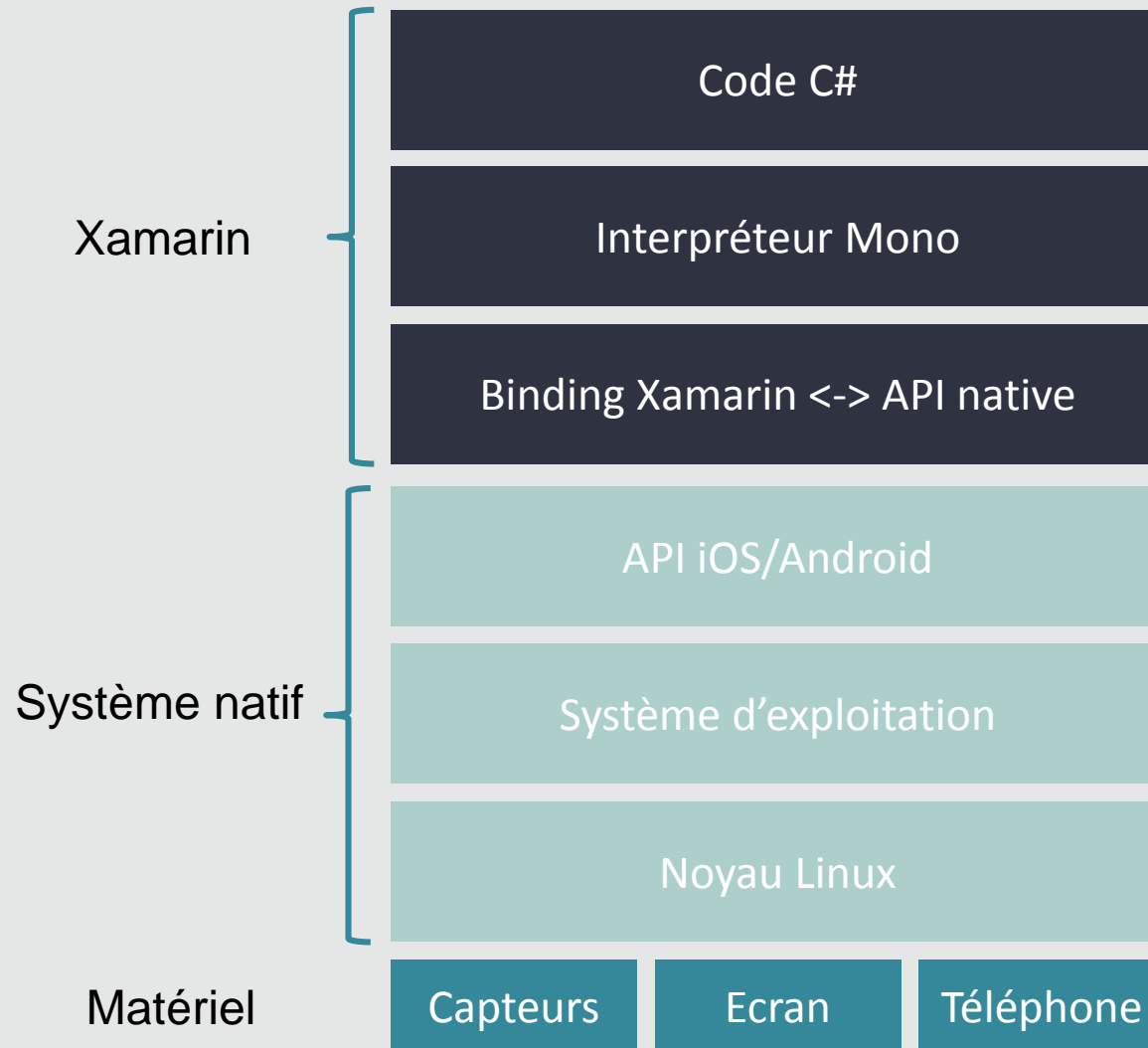
Architecture de Xamarin



Architecture de Xamarin



Architecture de Xamarin



Architecture de Xamarin

En fonction de l'approche utilisée, Xamarin promet de pouvoir mutualiser entre 50 et 80% du code source

Les 50 à 20% restants constituent la vue, qui doit être implémentée en natif pour respecter les contraintes de performance et d'intégration à la plateforme

Forces et faiblesses de Xamarin

Les +

50 à 80% du code mutualisé

Les performances du natif

Une interface complètement intégrée
à la plateforme

Un langage unique : C#

Forces et faiblesses de Xamarin

Les -

Nécessité de se former à C#

Nécessité d'utiliser l'IDE Visual Studio

Développement en double des vues

Conception et architecture orientées mutualisation

Forces et faiblesses de Xamarin

Xamarin se situe entre développement natif et développement multiplateforme (type WebView)

Fait bénéficier des avantages du natif tout en économisant sur les compétences requises, le budget et le planning

Il faut néanmoins maîtriser C#, l'IDE Visual Studio et avoir une bonne connaissance des API natives

Historique de Xamarin

Date	Evènement
1999	Création de la société Ximian ayant pour objectif de porter C# et .NET Framework sous Linux
2001	Le projet open source Mono est lancé
2011	La société Xamarin est lancée et annonce qu'elle supportera le développement et le support de Mono
2012	Sortie de Xamarin.Mac utilisant Mono (et donc C#/.NET Framework) pour développer des applications OSX
2013	Sortie de Xamarin 2.0 utilisant Mono pour développer des applications OSX, iOS et Android
2016	Microsoft annonce le rachat de Xamarin et la mise sous licence open source de celui-ci
2018	Xamarin continue de développer son écosystème et son offre de service

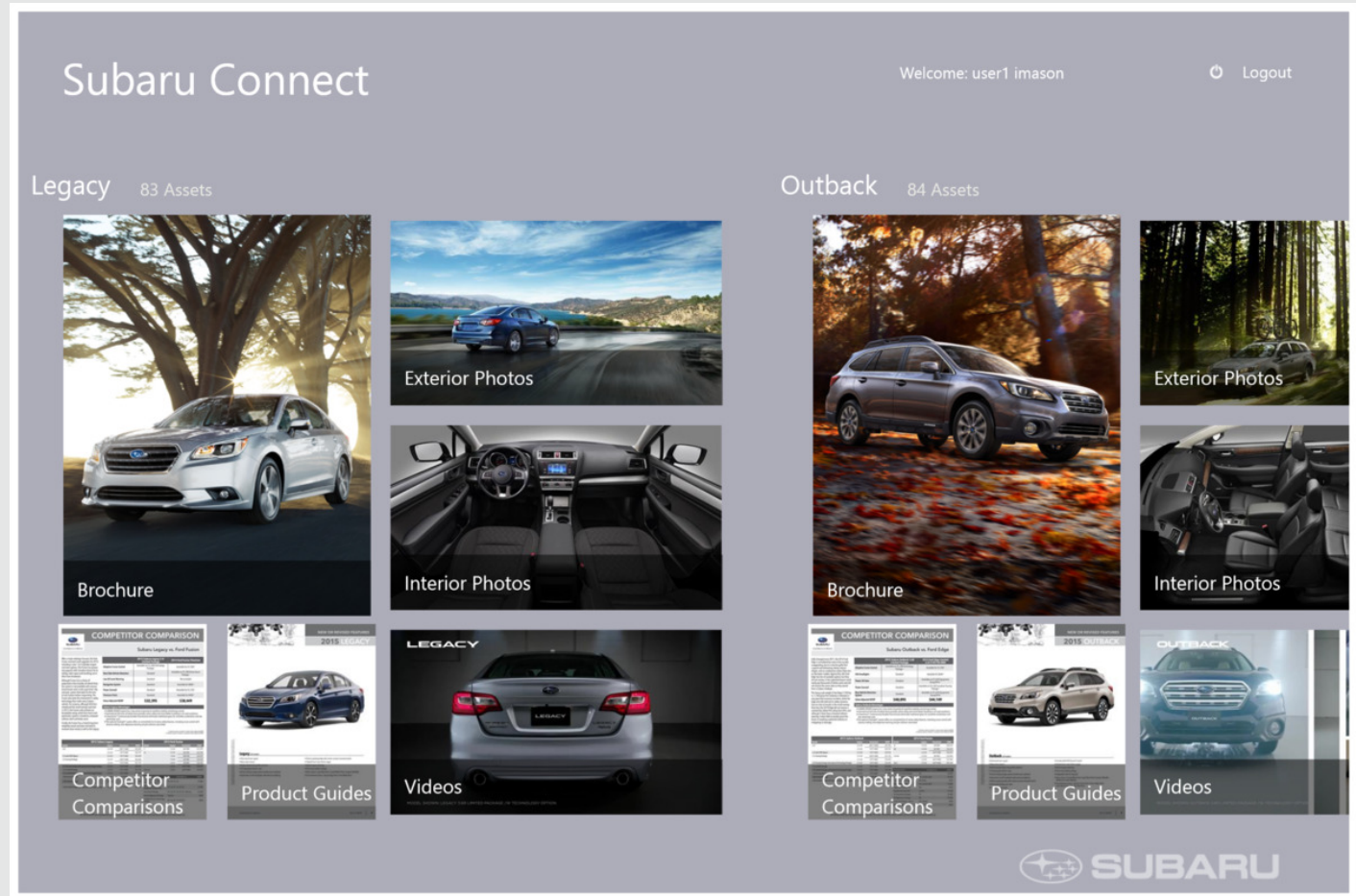
Historique de Xamarin

Xamarin est aujourd'hui un framework et une suite d'outils open source et gratuits

Xamarin est adhérent à l'IDE Visual Studio et requiert une licence professionnelle ou entreprise pour disposer de l'intégralité des outils

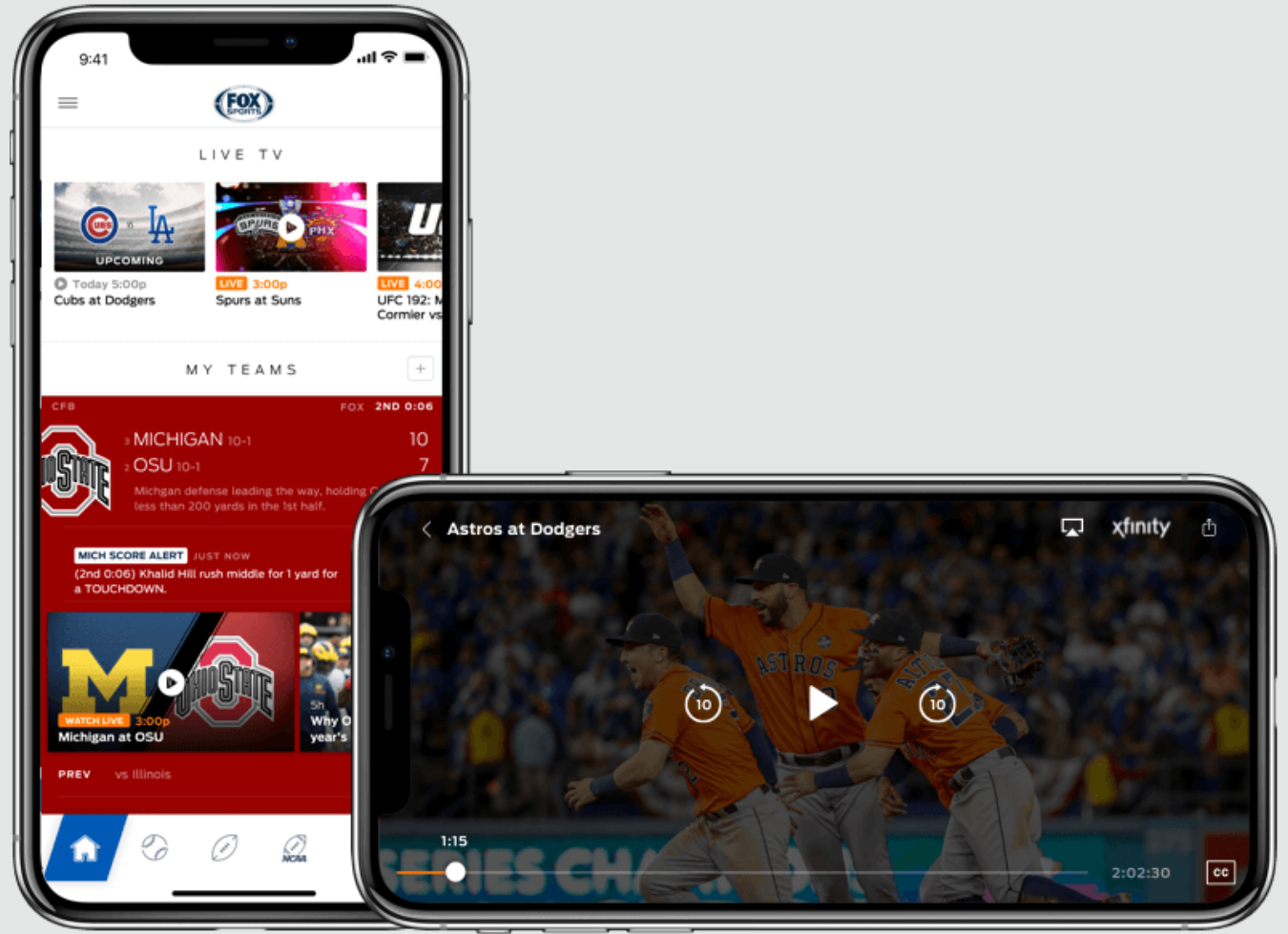


Exemples d'applications réalisées avec Xamarin



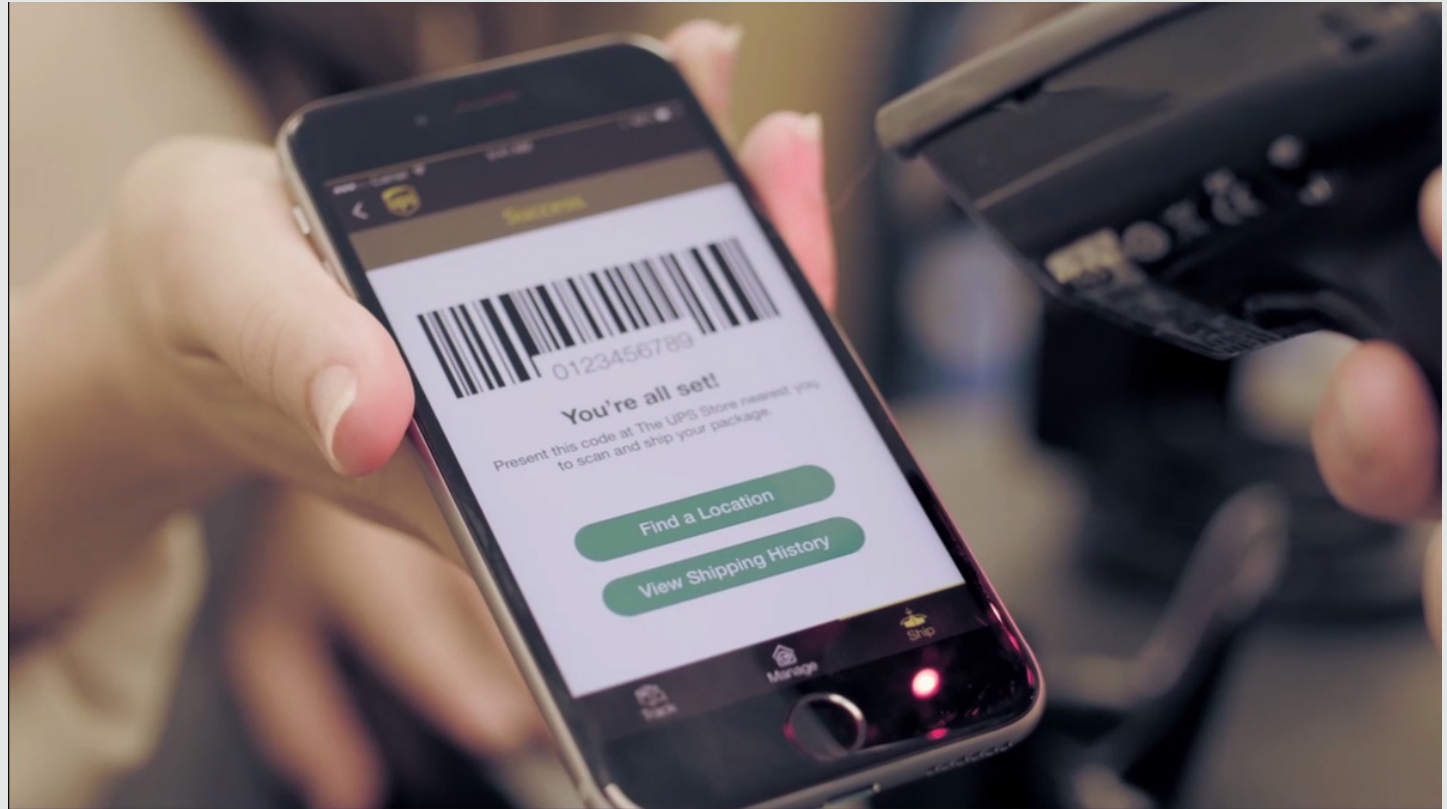
Présentation de Xamarin

Exemples d'applications réalisées avec Xamarin



Présentation de Xamarin

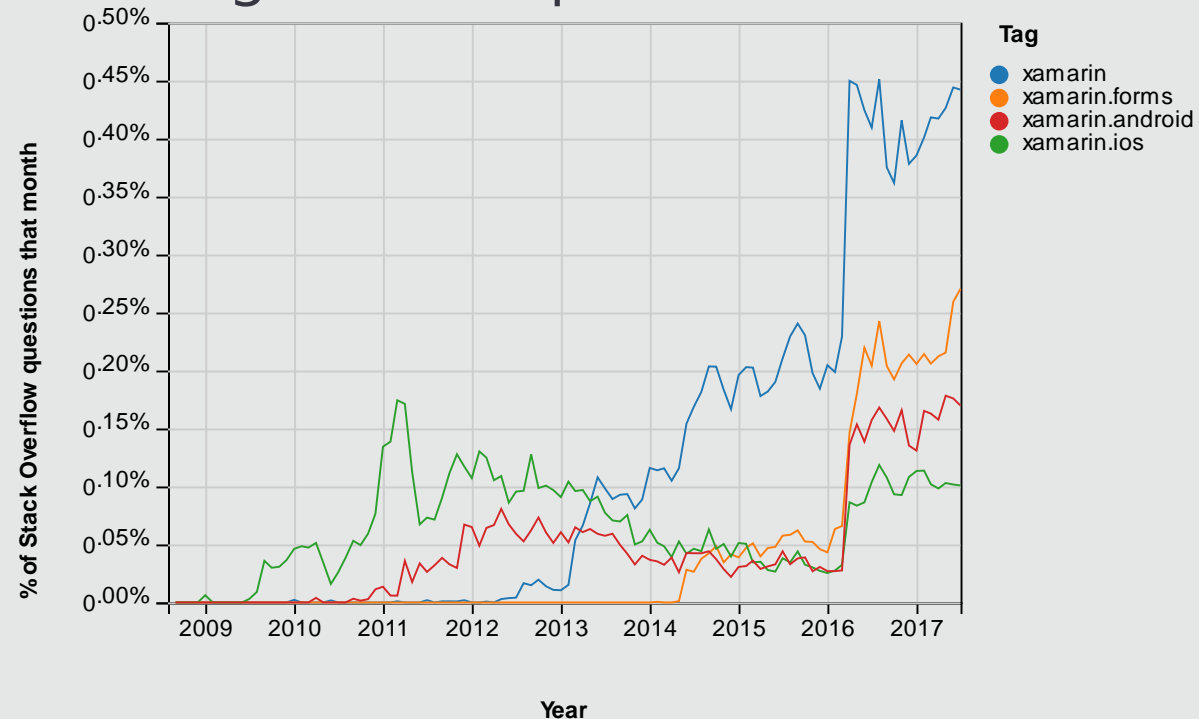
Exemples d'applications réalisées avec Xamarin



Exemples d'applications réalisées avec Xamarin

Xamarin rassemble aujourd'hui plus d'un million de développeurs et des milliers d'applications

Sa popularité ne cesse de grandir : la preuve avec Stack Overflow !



Environnement de développement

- Pour développer avec Xamarin, il est nécessaire d'installer plusieurs outils :
 - SDK
 - Compilateur
 - Plateformes mobiles

Présentation de Xamarin

Environnement de développement



Un PC avec Windows 10

Un smartphone Android (facultatif)

Visual Studio

L'écosystème Xamarin

Présentation de Xamarin

Environnement de développement



Un Mac avec OSX

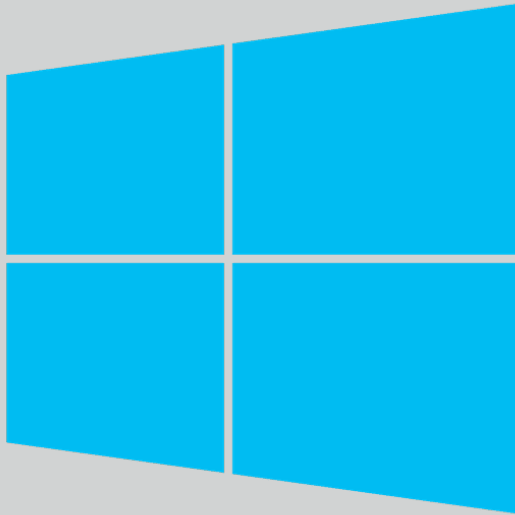
Un iPhone pour la validation des livrables

Visual Studio for Mac

L'écosystème Xamarin

Présentation de Xamarin

Environnement de développement



Un PC avec Windows 10 Professionnel

Visual Studio

L'écosystème Xamarin

Démonstration



Présentation de Xamarin

Installation de l'environnement de développement

- Installer l'environnement de développement Xamarin sous Windows
 - Télécharger Visual Studio
 - Installer les modules Xamarin pour Visual Studio
 - Vérifier l'installation de Visual Studio



TP

