



Xamarin

Chapitre II

Introduction à Xamarin et MAUI .NET





Xamarin

Plan

- ❶ Modèles natifs de développement mobile
- ❷ Modèle de développement Xamarin
- ❸ Développement C# de bout-en-bout
- ❹ Modèles Xamarin traditionnel vs. Xamarin.Forms vs. MAUI .NET
- ❺ Compilation des applications (Xamarin.Forms et MAUI .NET)
- ❻ Compilation des contrôles (Xamarin.Forms seulement)
- ❼ Services de dépendance (Xamarin.Forms seulement)
- ❽ Compilation des contrôles (MAUI .NET seulement)



Xamarin

1. Modèles natifs de développement mobile (1/2)

- Xamarin est une technologie facilitant le **partage du code source entre plateformes !**
- Le résultat de compilation => **Application native**

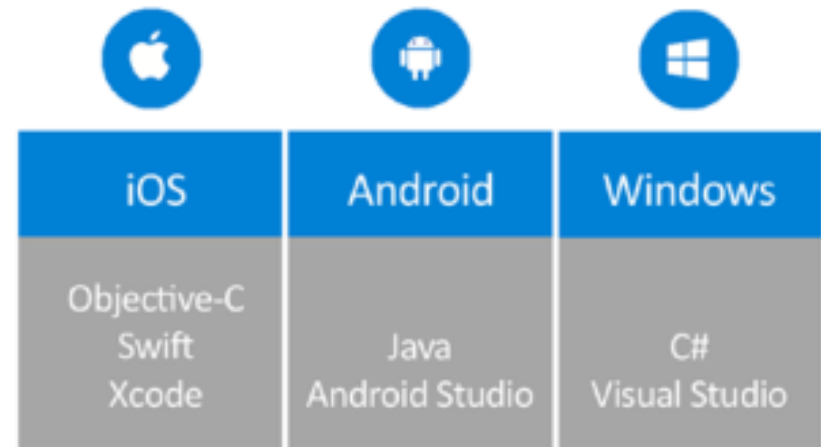




1. Modèles natifs de développement mobile (2/2)

- **Approche traditionnelle de développement d'applications natives (avant Xamarin)**
 - Approche en « Silo »
 - L'application est développée 3 fois (pour les 3 plateformes différentes)

Silo Approach



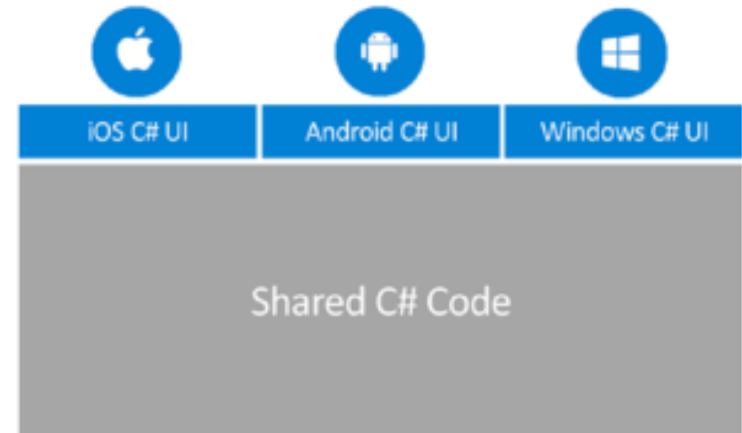


Xamarin

2. Modèles de développement Xamarin (1/1)

- **Approche Xamarin!**
 - Base de code C# partagée
 - Accès à une API native à 100%
 - Performances élevées!

Xamarin's Approach





3. Développement C# de bout-en-bout (1/2)

- Avec les plateformes .NET Core et .NET standard
 - Développement **multi-plateforme** et **cross-plateforme**
 - Architecture **distribuée** et **intégrée**
 - Basée sur les langages de développement .NET (**par exemple C#**)
- Les applications mobiles sont facilement intégrées avec le Cloud en utilisant une seule API

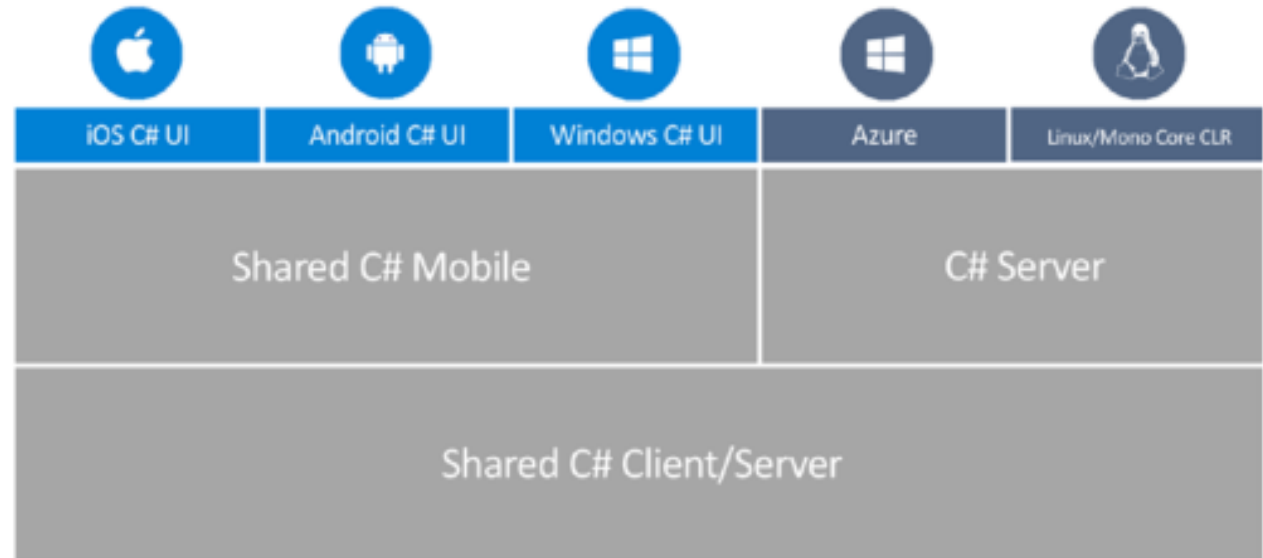


Xamarin

3. Développement C# de bout-en-bout (2/2)

Dév. d'applications mobiles

End-to-End C# Development



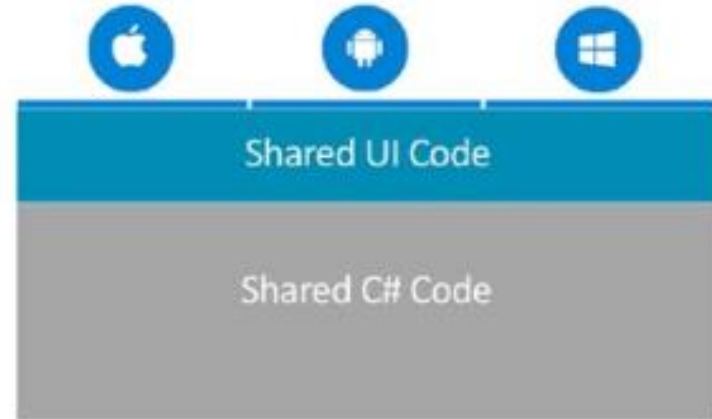


Xamarin

4. Modèles Xamarin traditionnel vs. Xamarin.Forms vs. MAUI .NET (1/1)



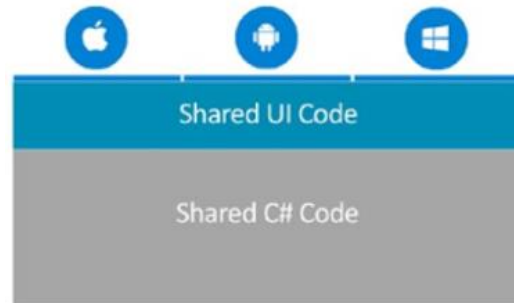
Traditional Xamarin



Xamarin.Forms

MAUI .NET

Multi-Platform App UI =



+ ressources communes

Projet unique !



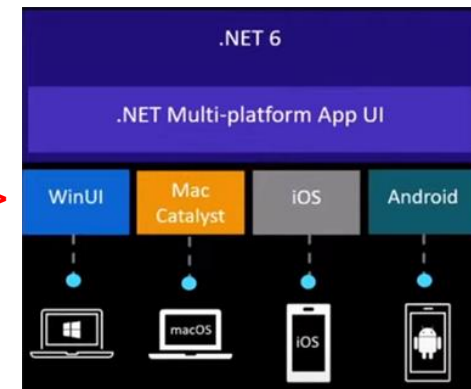
5. Compilation des applications (Xamarin.Forms et MAUI .NET) (1/4)

- Les applications Xamarin et MAUI .NET sont compilées et exécutées avec la **Mono runtime**
 - Cette dernière est **portée vers toutes les plateformes cibles** de Xamarin/MAUI .NET
- Remarque
 - Pour windows
 - Xamarin.Forms : création d'application **UWP** (compatibles Win 8.1, Win 10 et Win 11)
 - MAUI .NET : création d'applications **WinUI 3** (compatibles Win 10 et Win 11)

Xamarin.Forms =>



MAUI .NET =>





5. Compilation des applications (Xamarin.Forms et MAUI .NET) (2/4)

- Structure et exécution d'une application Xamarin/MAUI sur **Android**





Xamarin

5. Compilation des applications (Xamarin.Forms et MAUI .NET) (3/4)

- Structure et exécution d'une application Xamarin/MAUI sur iOS et iPadOS

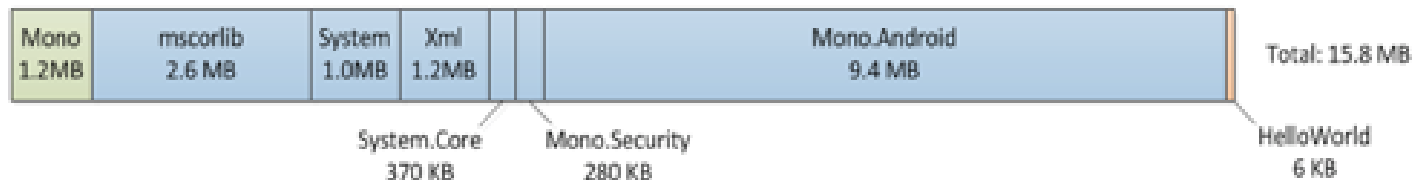




5. Compilation des applications (Xamarin.Forms et MAUI .NET) (4/4)

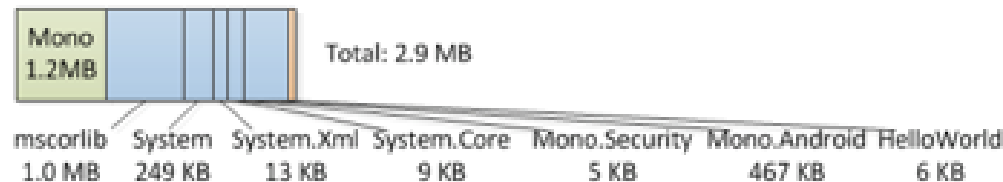
- Exemple de taille d'application Android entre le développement (**Debug**) et la publication (**Release**)

When developing



Linker

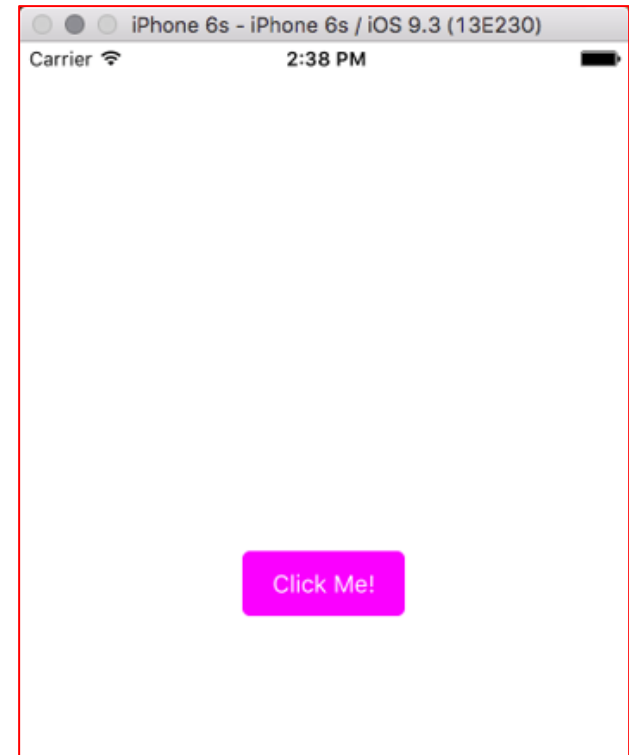
When publishing





6. Compilation des contrôles (Xamarin.Forms seulement) (1/3)

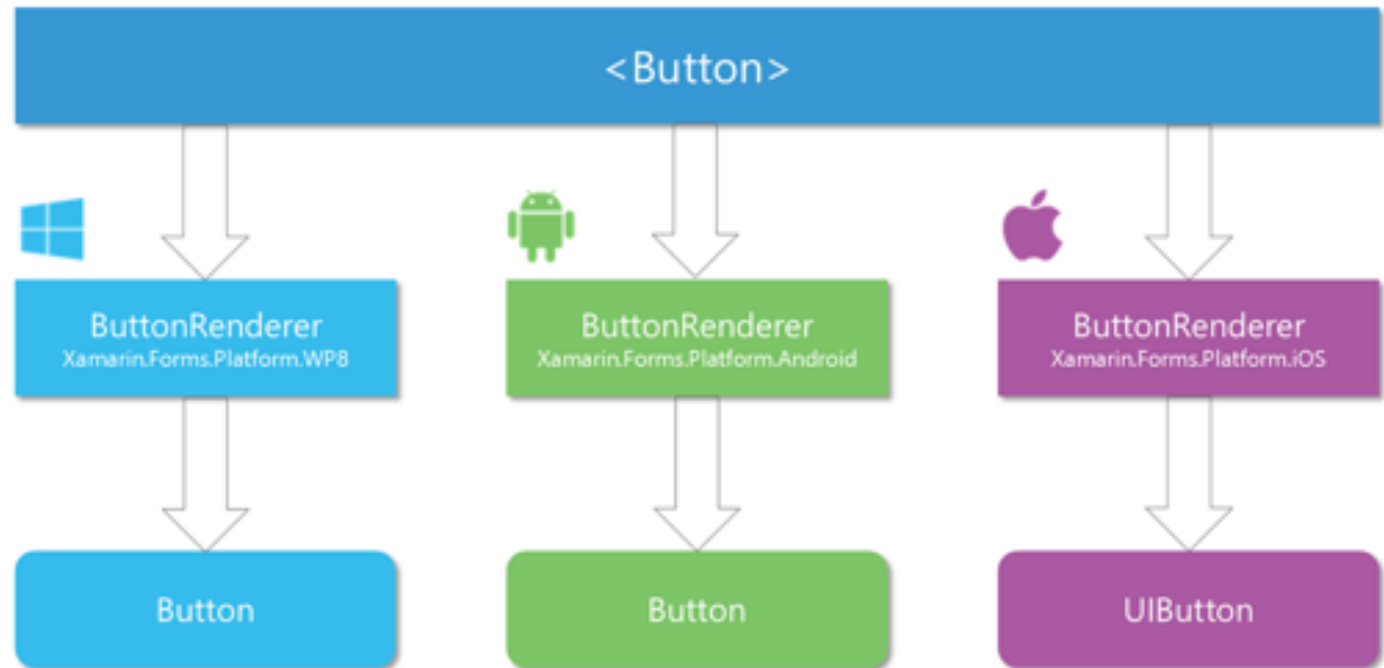
- Soit l'application Xamarin.Forms suivante :
 - Une interface simple avec un bouton « Click Me! » au milieu





6. Compilation des contrôles (Xamarin.Forms seulement) (2/3)

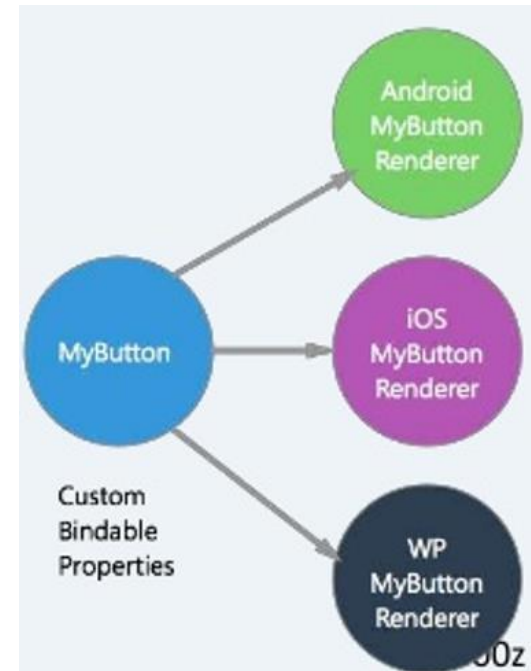
- Comment marche le bouton?
 - Un **render de plateforme** se charge de transformer le bouton Xamarin.Forms en bouton natif (Android, iOS ou Windows)
 - => **Résultat** : utilisation des classes natives pour le bouton





6. Compilation des contrôles (Xamarin.Forms seulement) (3/3)

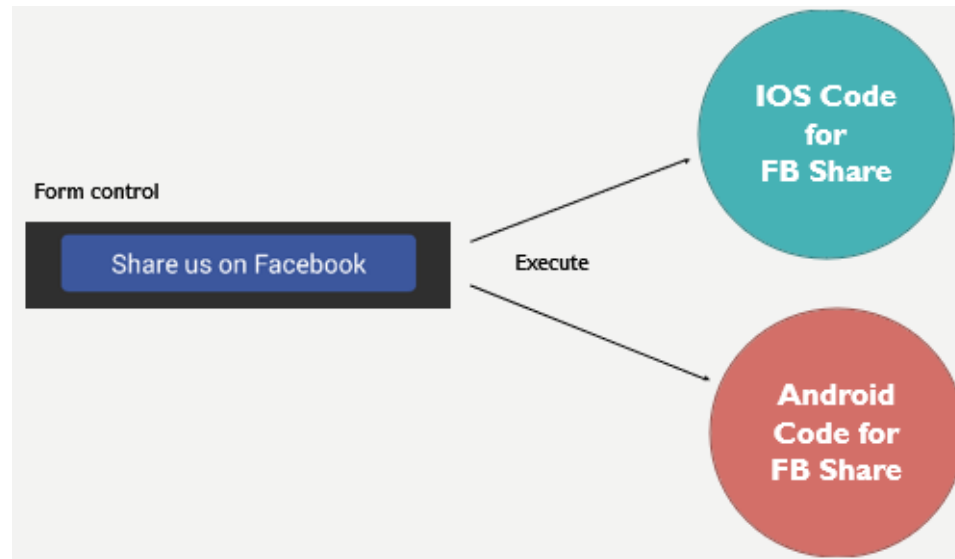
- Et si on compte **personnaliser** un contrôle Xamarin.Forms?
 - Le **render** personnalisé?
 - La personnalisation du render s'effectue d'une manière spécifique à chaque plateforme
 - Mais toujours en C#!





7. Services de dépendance (Xamarin.Forms seulement) (1/2)

- Ajout d'une fonctionnalité non existante à Xamarin.Forms?
 - => Utilisation du « **Dependency Service** »
 - => Implémentation de la fonctionnalité pour chaque plateforme
 - Exemple : **bouton de partage Facebook**

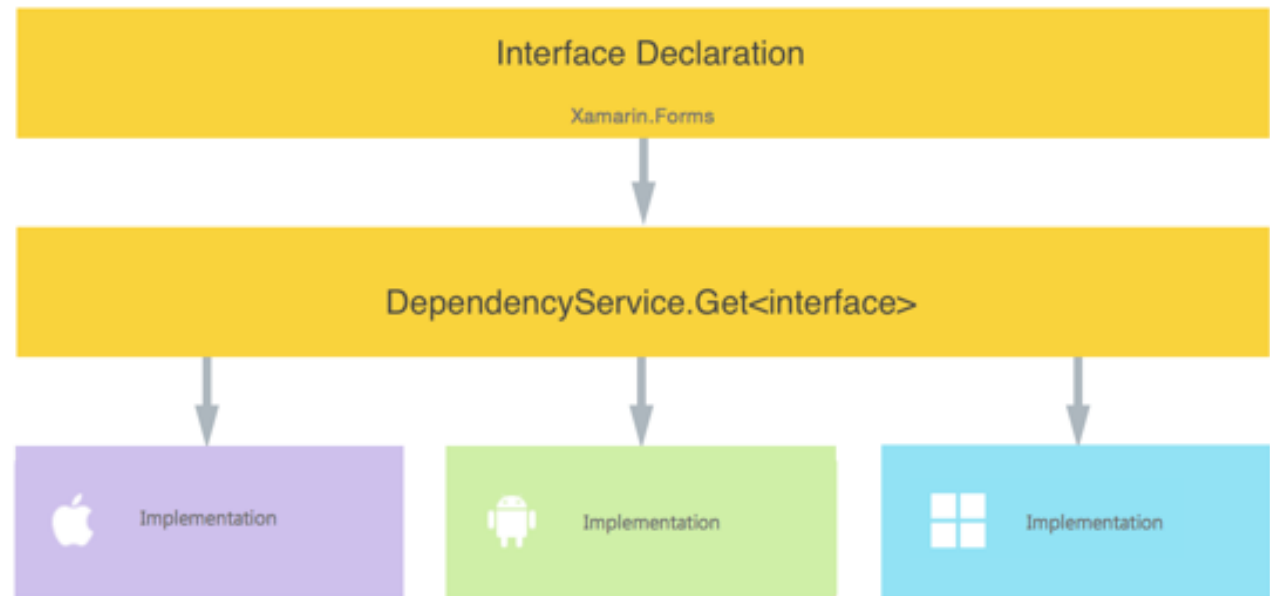




Xamarin

7. Services de dépendance (Xamarin.Forms seulement) (2/2)

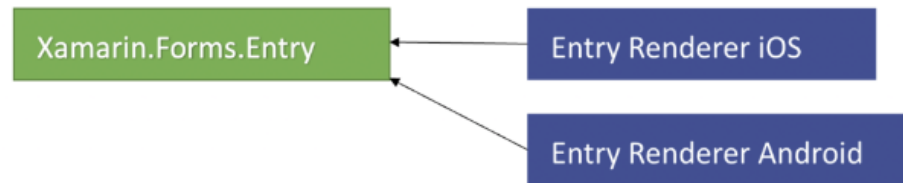
- Déclaration du service!





8. Compilation des contrôles (MAUI .NET)

- Les **renderers** en **Xamarin.Forms** sont remplacés par les **handlers** en **MAUI .NET**
 - Xamarin.Forms (**renderers**)



- MAUI .NET (**handlers** ou **Slim renderers**) => Découplage entre le **rendering spécifique à la plateforme** et la **plateforme UI**

