

.NET

Sommaire

- Introduction .NET
 - Environnement de développement
 - .NET Framework
 - Mono et Xamarin
 - .NET Core
 - .NET 5
 - .NET Runtime et .NET SDK
 - .NET Standard
 - Intermediate language
 - Dotnet CLI
 - C#
 - Scénarios de développement spécialisés

Environnement de développement

- Microsoft propose 4 editeurs / IDE
 - Visual Studio Code
 - Github Codespaces
 - Visual Studio 2019
 - Visual Studio 2019 pour MAC
- Visual Studio Code:
 - Cross platform (Windows, macOS, Linux)
 - Gratuit et léger
 - Supporte beaucoup de langage grâce aux extensions

Environnement de développement

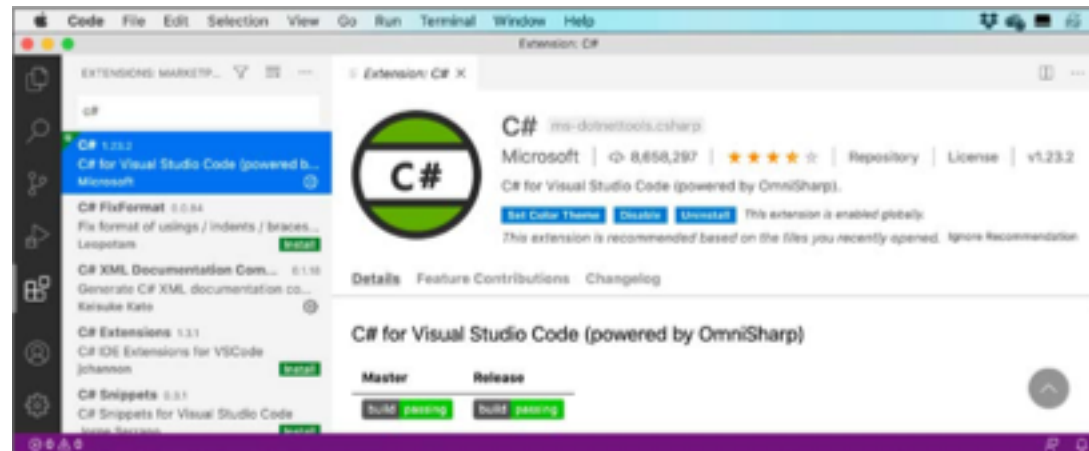
- Github codespaces
 - Basé sur Visual Studio Code
 - Hébergé dans le cloud
 - Disponible depuis un navigateur
 - Organisations utilisant GitHub Team ou GitHub Enterprise Cloud
- Visual Studio 2019
 - Windows uniquement
 - Seul IDE pour faire des application Windows (WPF /UWP)

Environnement de développement

- Visual Studio 2019 pour MAC
 - macOS uniquement
 - Surtout utile pour le développement mobile (iOS / iPad)
 - Nécessite Xcode pour le développement mobile

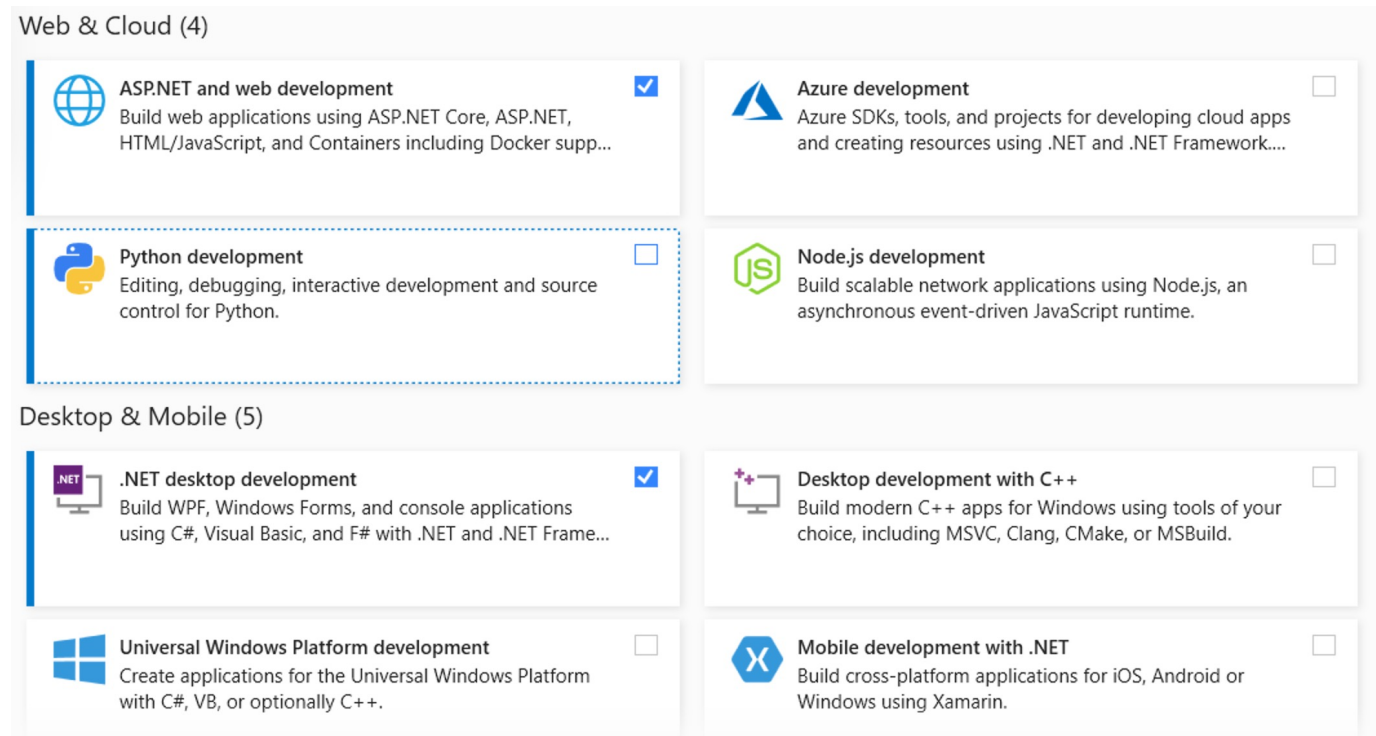
Environnement de développement

- Visual Studio Code Configuration:
 - Pré-requis:
 - vscode: <https://code.visualstudio.com>
 - .NET 5: <https://dotnet.microsoft.com/download>
 - Installer C# extension et C# for Visual Studio Code



Environnement de développement

- Visual Studio 2019 Configuration:
 - Pré-requis:
 - Visual Studio Community: <https://visualstudio.microsoft.com>



Environnement de développement

Other Toolsets (6)



Data storage and processing



Connect, develop, and test data solutions with SQL Server, Azure Data Lake, or Hadoop.



Data science and analytical applications



Languages and tooling for creating data science applications, including Python and F#.



Visual Studio extension development



Create add-ons and extensions for Visual Studio, including new commands, code analyzers and tool windows.



Office/SharePoint development



Create Office and SharePoint add-ins, SharePoint solutions, and VSTO add-ins using C#, VB, and JavaScript.



Linux development with C++



Create and debug applications running in a Linux environment.



.NET cross-platform development



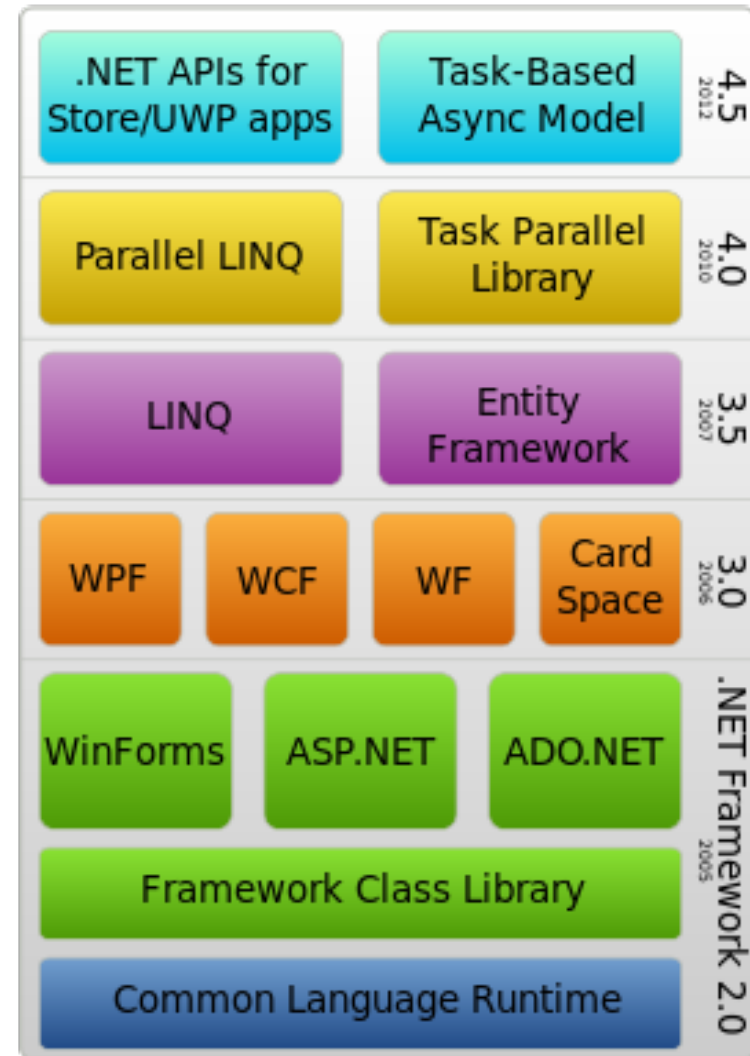
Build cross-platform applications using .NET, ASP.NET Core, HTML/JavaScript, and Containers including Docker...

.NET Framework

- Sortie en 2002, compatible depuis Windows 98
- Plateforme de développement qui comprend:
 - Common Language Runtime (CLR) qui gère l'exécution du code
 - Base Class Library (BCL) qui fournit les bibliothèques de base pour créer des applications
 - Framework Class Library (FCL) qui contient la BCL + ASP.NET, WPF, ADO.NET...
- C'est un composant officiel du système d'exploitation Windows
- Toutes les applications sur un ordinateur partagent la même version du CLR et des bibliothèques du Global Assembly Cache (GAC)
- Supporte plusieurs langage (C#, VB, PowerShell...)

.NET Framework

- La CLR est une machine virtuelle d'application qui fournit les services de sécurité, gestion de la mémoire et gestion des exceptions.
- On dit que le code écrit avec .NET Framework est du code managé



.NET Framework

- Avantages:
 - Framework intégré à Windows
 - Mis à jour et maintenu par Microsoft à travers Windows Update
- Inconvénients:
 - Windows only
 - Legacy Plateforme
 - Contraignant si votre application a besoin d'une version spécifique d'une bibliothèque dans le GAC par exemple
 - Rarement mis à jour, pour ne pas causer de problèmes de rétrocompatibilité

.NET Framework

- Demo .NET Framework et Visual Studio 2019

Mono / Xamarin

- Sortie en 2004
- Implémentation du .NET Framework développé par des tiers
- Open source et fait partie de la fondation .NET
- C'est la 1ère version cross-plateforme de .NET
- A toujours un petit train de retard par rapport a .NET Framework
- <https://www.mono-project.com>

Mono / Xamarin

- C'est aujourd'hui la base d'autre framework comme Xamarin ou Unity
- Unity est une plateforme de développement de jeux vidéo
- Xamarin est une plateforme pour créer des application Mobile (Android, iOS, tvOS, watchOS, macOS, and Windows) avec .NET et C#
 - Racheté par Microsoft en 2016
 - Gratuit et open source depuis 2016
 - Fait partie des composants de Visual Studio 2019 (Windows et macOS)
 - Le framework permet d'accéder aux fonctionnalités natives
 - Support du XAML et du pattern MVVM

Mono / Xamarin

- Demo Xamarin

.NET Core

- Microsoft a réécrit le framework .NET pour le découplé de Windows et le simplifier
- Sortie en 2016, il est le successeur de .NET Framework
- Framework gratuit, open source et cross-plateforme
- Il comprend aussi une CLR appelé CoreCLR et des bibliothèques CoreFX
- CoreCLR et CoreFX sont compatible avec CLR et FCL
- .NET Core 1.0 cible essentiellement le developement web avec ASP.NET Core
- A partir de .NET Core 3.0, il est possible de faire des application de bureau pour Windows

.NET Core

- Avantages:
 - Cross-plateforme (Windows, macOS, Linux)
 - Idéal pour le développement web
 - Activement mis à jour par Microsoft
 - Peut être déployé côte à côte avec une application
 - Les mises à jour de .NET Core n'affectent donc plus les autres applications sur la même machine
- Inconvénients:
 - Quelques problèmes de compatibilité avec de vieilles bibliothèques pour Windows

.NET 5

- Sortie en Novembre 2020, c'est l'unification de toute la plateforme .NET (sauf mobile prévu pour novembre 2021)
- C'est l'équivalent de la dernière version de .NET Core

.NET 5

- Demo .NET 5

.NET Runtime et .NET SDK

- Le SDK est mis a jour plus souvent que le runtime ce qui peut entrainé des nom de version différentes
- Le .NET Runtime est versionné major.minor.patch
- Le SDK suit le versionning major.minor du runtime

Change	.NET Runtime	.NET SDK (*)
Initial release	5.0.0	5.0.100
SDK patch	5.0.0	5.0.101
Runtime and SDK patch	5.0.1	5.0.102
SDK feature change	5.0.1	5.0.200

.NET Runtime et .NET SDK

- Les mises à jour de NET Runtime sont compatibles avec une version majeure telle que 5.x et les versions mises à jour du SDK .NET permettent de créer des applications qui ciblent les versions précédentes du runtime
- Vous pouvez donc supprimer en toute sécurité les anciennes versions.
- Pour voir les SDK et Runtime installé sur votre machine avec:
 - dotnet --list-sdks
 - dotnet --list-runtimes
- Pour les désinstaller il existe un outil: .NET Uninstall Tool
 - <https://docs.microsoft.com/en-us/dotnet/core/additional-tools/uninstall-tool>

.NET Standard

- En 2019 il y avait .NET Core, .NET Framework et Xamarin pour 3 scénarios différents
- Pour gérer la compatibilité entre ces 3 plateformes, Microsoft a défini une spécification pour un ensemble d'API que toutes les plateformes .NET pourraient implémenter
- Cela permet de partager du code facilement entre les plateformes
- Ce n'est qu'un standard, il n'y a rien à installer.
- Pour être compatible .NET Standard X il suffit d'utiliser la plateforme .NET qui respecte le standard X
- .NET Standard 2.0 est implémenté par les dernières versions de .NET Framework, .NET Core et Xamarin
- Normalement avec la sortie de .NET 6 (novembre 2021), ce standard ne sera utile pour partager du code avec .NET Framework

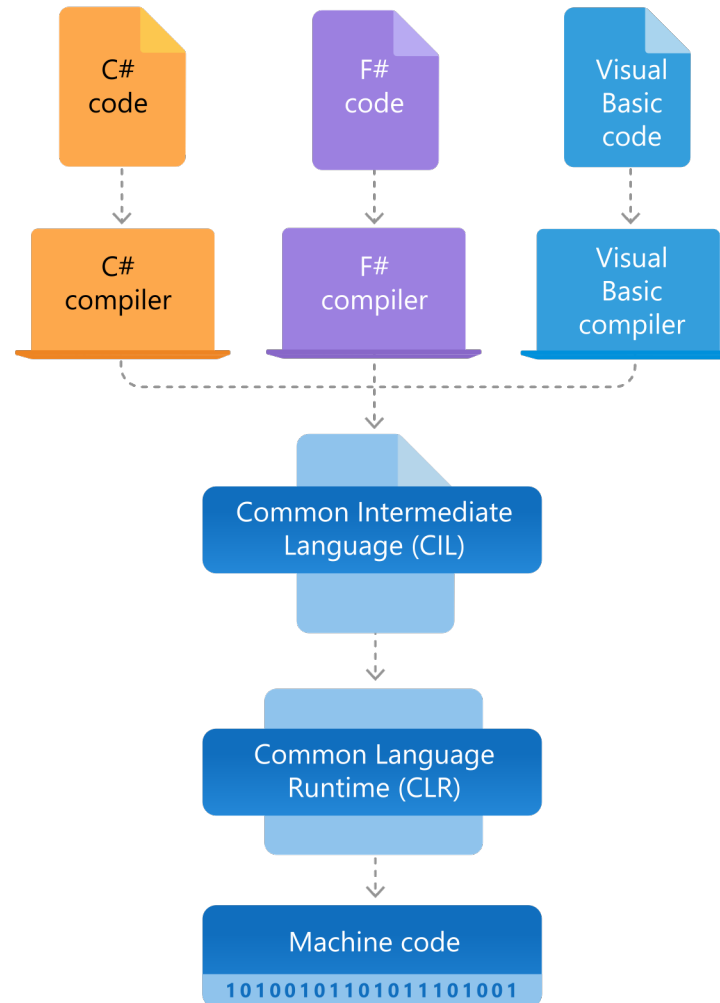
.NET Standard

.NET Standard	1.0	1.1	1.2	1.3	1.4	1.5	1.6	2.0	2.1
.NET Core	1.0	1.0	1.0	1.0	1.0	1.0	1.0	2.0	3.0
.NET Framework	4.5	4.5	4.5.1	4.6	4.6.1	4.6.1 ¹	4.6.1 ¹	4.6.1 ¹	N/A ²
Mono	4.6	4.6	4.6	4.6	4.6	4.6	4.6	5.4	6.4
Xamarin.iOS	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.14	12.16
Xamarin.Mac	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.8	5.16
Xamarin.Android	7.0	7.0	7.0	7.0	7.0	7.0	7.0	8.0	10.0
Unity	2018.1	2018.1	2018.1	2018.1	2018.1	2018.1	2018.1	2018.1	TBD
Universal Windows Platform	8.0	8.0	8.1	10.0	10.0	10.0.16299	10.0.16299	10.0.16299	TBD

Intermediate language

- Le compilateur C# (Roslyn) utilisé par la CLI dotnet converti votre code C# en langage intermédiaire (IL) code et le stock dans une assembly (DLL ou EXE)
- Les instructions IL sont comme des instruction en assembleur qui seront exécuté par la CLR
- Au moment de l'exécution, la CLR charge l'assembly et le compilateur Just in Time (JIT) le compile en instruction CPU natives
- Pour .NET 5 il existe donc une CLR pour macOS, une pour linux et une pour Windows qui pourra exécuter le même code IL
- Plusieurs langages peuvent être utilisés pour créer du code IL (C#, VB, F#...)

Intermediate language



Intermediate language

- Demo ILSpy

dotnet CLI

- Chaine d'outils cross-plateforme pour le développement, la création, l'exécution et la publication d'application .NET
- .NET CLI est inclus avec le .NET SDK
- Les commandes de bases:
 - dotnet new console: pour créer un nouveau projet console
 - dotnet build: pour compiler le projet
 - dotnet restore: pour installer les packages NuGet
 - dotnet run: pour lancer le projet
 - dotnet pack: pour créer un package NuGet
 - dotnet clean: pour supprimer ce qui a été compilé
 - dotnet new sln: pour créer un fichier solution nécessaire avec Visual Studio 2019

dotnet CLI

- La gestion des packages:
 - `dotnet add package nom_du_package`: ajout d'un package NuGet
 - `dotnet add reference fichier_csproj`: ajout d'un autre projet en référencé
 - `dotnet remove package nom_du_package` : supprime un package NuGet
 - `dotnet remove reference fichier_csproj`: supprime un projet référencé

dotnet CLI

- Demo et presentation de NuGet

C#

- Sortie en 2002 avec la première version de .NET
- Langage orienté objet à type statique depuis la version 1.0
- C# 2.0 (2005): support des types generics et type nullable
- C# 3.0 (2007): LINQ (Language Integrated Queries), l'inférence de type et les expressions lambda
- C# 4.0 (2010): mot clé dynamic et paramètre optionnel / nommé
- C# 5.0 (2012): Simplification du code asynchrone avec les mots clés async / await

C#

- C# 6.0 (2015): import static, interpolated strings, propriété en lecture seul
- C# 7.0 / 7.1 / 7.2 / 7.3 (2017-2018): Focus sur la programmation fonctionnelle
 - Nouveau tuple
 - Pattern matching
 - Fonction local
- C# 8.0 (2019): type référence nullable / non nullable, simplification du switch et méthode d'interface par défaut
- C# 9.0 (2020): type record, amélioration du pattern matching

Scénarios de développement spécialisés

- Application Console
- Windows Forms
 - Création d'applications de bureau pour les Windows
 - Concepteur visual
 - Glisser-déplacer de contrôles visuels
 - Faciles à créer et déployer
- Windows Presentation Foundation (WPF)
 - Création d'applications de bureau pour les Windows
 - Extensible Application Markup Language (XAML)
 - Indépendante de la résolution
 - Moteur de rendu vectoriel

Scénarios de développement spécialisés

- UWP
 - Création d'applications clientes pour Windows 10
 - API commune sur tous les appareils qui exécutent Windows 10
 - Peut utiliser des API Win32
 - Microsoft Store
 - Vignettes dynamiques
 - Notifications push
 - C#, C++, Visual Basic et JavaScript

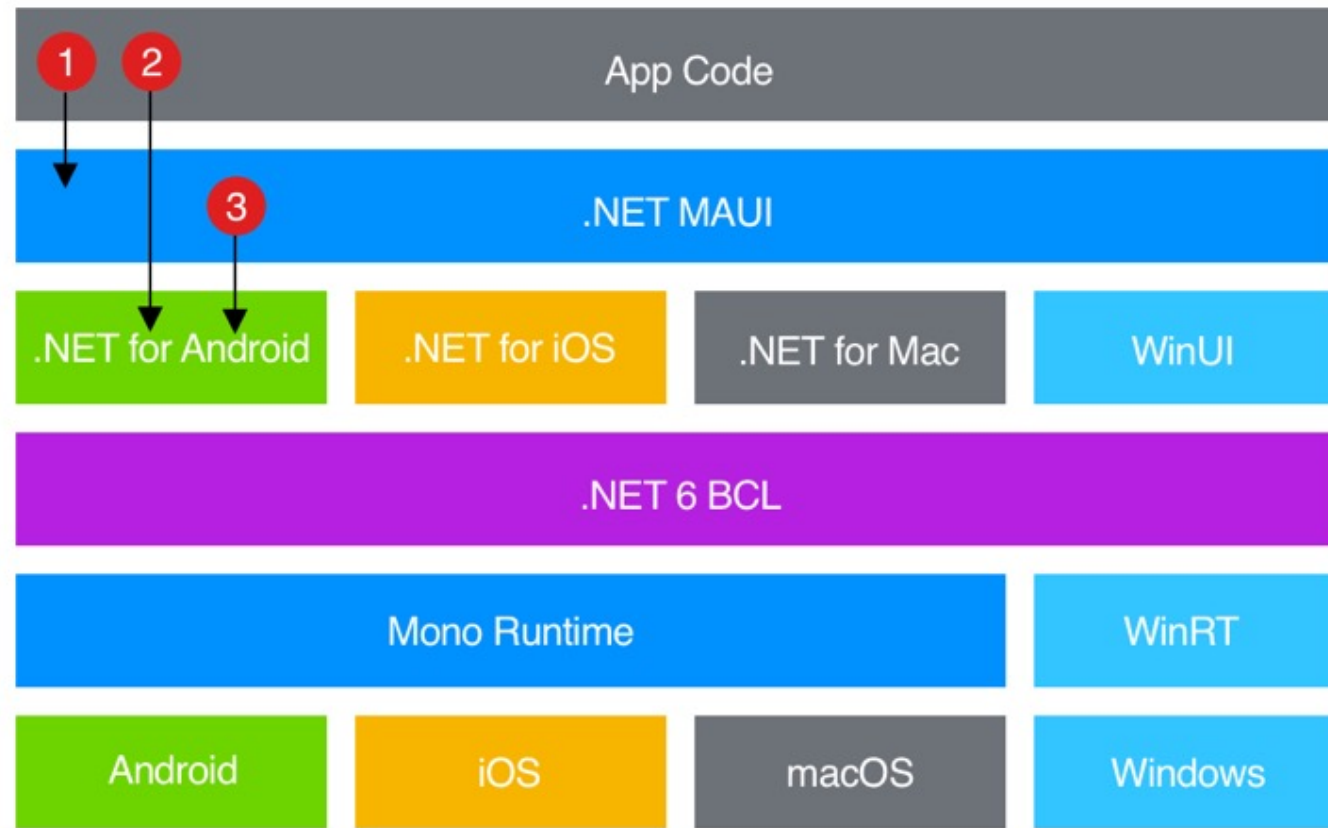
Scénarios de développement spécialisés

- .NET MUAI (.NET 6)
 - Cross platform framework pour application native mobile et bureau
 - C# + XAML
 - Evolution de Xamarin Forms (+performant, +extensible)
 - Idéal pour partager l'UI
 - Facile de passer de XF a MUAI

Scénarios de développement spécialisés

- Android: Compilent à partir de C# dans un langage intermédiaire (IL) puis compilé juste à temps (JIT) dans un assembly natif au lancement de l'application.
- iOS: Entièrement compilées à l'avance (AOT) à partir de C# dans le code d'assemblage ARM natif.
- macOS: utilise Mac Catalyst, solution d'Apple qui apporte votre application iOS sur bureau
- Windows: utilise WinUI 3 pour créer des applications natives pouvant cibler le bureau Windows et la plate-forme Windows universelle (UWP).

Scénarios de développement spécialisés



Scénarios de développement spécialisés

- ASP.NET
 - Création d'application Web / API
 - ASP.NET Core est une refonte d'ASP.NET 4.x
 - Multiplateformes
 - Léger et modulaire
 - Modèle MVC: framework puissant pour applications testables
 - Razor page: scénarios orientés page facile et productif
 - Blazor: interface web côté client en .NET (C# au lieu de JavaScript)

Scénarios de développement spécialisés

- Possibilité d'hébergement ASP.NET:
 - Kestrel (serveur web cross plateforme qui est inclus et activé par défaut)
 - IIS (Internet Information Services est le serveur web flexible et sécurisé historique de Microsoft)
 - HTTP.sys (alternative à Kestrel pour windows uniquement)
 - Nginx
 - Apache
 - Docker

Scénarios de développement spécialisés

- Windows Service
 - Création d'applications exécutables longue
 - Peuvent être lancés automatiquement au démarrage de l'ordinateur
 - Pas d'interface utilisateur
- Windows Communication Foundation (WCF)
 - Application orientée service
 - Permet d'envoyer des données sous forme de messages asynchrones d'un point de terminaison de service à un autre
 - Hébergé par une application ou IIS
 - Message au format XML ou binaire