

.NET : introduction

Achref El Mouelhi

Docteur de l'université d'Aix-Marseille
Chercheur en programmation par contrainte (IA)
Ingénieur en génie logiciel

`elmouelhi.achref@gmail.com`



Plan

- 1 Introduction
- 2 .NET Framework
- 3 .NET Core
- 4 Microsoft .NET
- 5 Xamarin & .NET MAUI
- 6 .NET Standard
- 7 .NET (.NET 5 → .NET 10)
- 8 Terminologie officielle
- 9 Versions
- 10 Installation

.NET

.NET

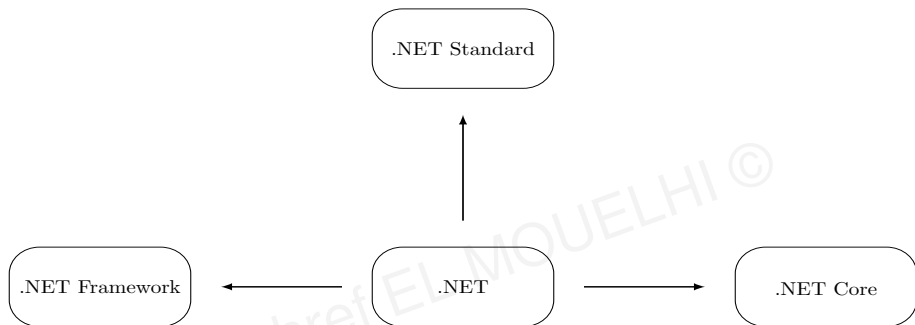
.NET



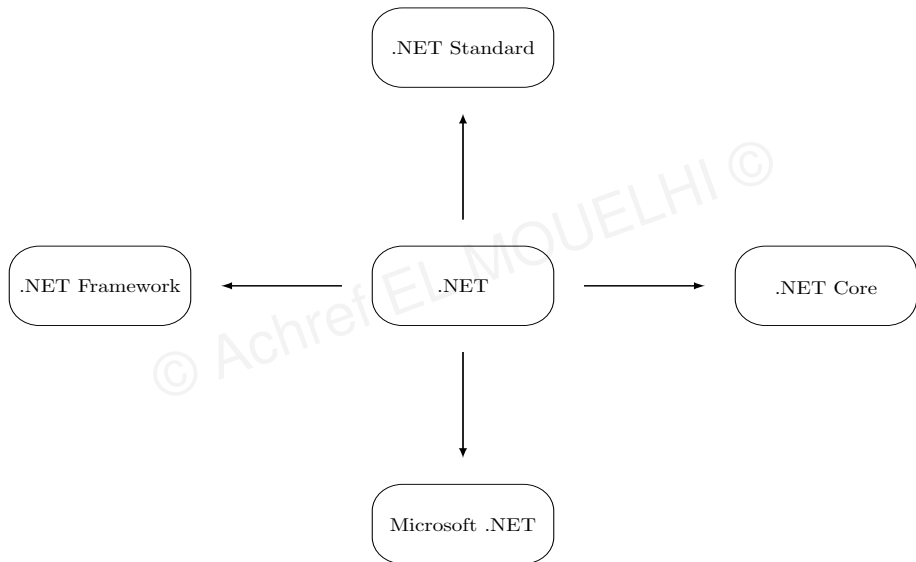
.NET



.NET



.NET



Quelle différence ?

- **.NET** (anciennement **.NET Core**)
- **.NET Framework**
- **.NET Standard**
- **Microsoft .NET**

.NET Framework ?

- Cadre logiciel (cadriciel) : plateforme de développement présenté officiellement par **Microsoft** en 2002
- Créé par **Anders Hejlsberg** le créateur de :
 - **Turbo Pascal**
 - **C#** intégré au **.NET Framework**
 - **TypeScript** (utilisé notamment par **Angular** de **Google**)
- Permettant de développer des programmes (applications) fonctionnant dans un environnement **Microsoft** (sous **Windows**).
- Plusieurs langages supportés : **C#, C++, F#, J#, VB.NET...**

.NET

Composants .NET Framework

= **CLR** + **ASP.NET** + **ADO.NET** + **Entity Framework** + **LINQ** + **WinForms** + **WCF** + **WPF** + ...

- **CLR** (pour **C**ommon **L**anguage **R**untime) : Machine virtuelle de **.NET Framework**.
- **ASP.NET** : framework de création d'applications Web.
- **ADO.NET** : accès aux données.
- **Entity Framework** : ORM de **Microsoft**
- **LINQ** : langage intégré pour requêter des données.
- **WinForms** : composant de **.NET Framework** pour les applications **Desktop**.
- **WPF** (pour **W**indows **P**resentation **F**oundation) : composant remplaçant de **WinForms**.
- **WCF** (pour **W**indows **C**ommunication **F**oundation) : composant de la plateforme **.NET Framework** permettant la création d'applications orientées service.

Exemple

Exemple d'application développée avec **.NET Framework : Paint.NET**

.NET Core ? (renommé en .NET depuis 2020)

- Cadriciel open-source présenté officiellement par **Microsoft** en 2016
- Compatible **Windows**, **macOS** et **Linux**
- Écrit en **C#** et **C++**
- Permettant de développer seulement des
 - Applications consoles
 - Applications Web et APIs REST
 - Microservices
 - Applications multiplateformes avec **.NET MAUI**
- Langages supportés : **C#**, **F#**, **VB.NET** (partiellement (aucun support pour **C++**))

.NET

= CoreCLR + ASP.NET Core + Entity Framework Core + LINQ + MAUI + ...

- **CoreCLR** \cong **CLR** dans **.NET Framework**.
- **ASP.NET Core** \cong **ASP.NET** dans **.NET Framework**.
- **Entity Framework Core** \cong **Entity Framework** dans **.NET Framework**.
- **MAUI** (**M**ulti-platform **A**pp **U**ser **I**nterface) : composant open-source de **Microsoft** pour créer des applications natives mobiles et desktop avec **C#** et **XAML**.

Remarques

- **.NET Core** était une nouvelle plateforme réécrite depuis zéro.
- À partir de **.NET 5**, elle a évolué vers le nom unifié **.NET**.

.NET

.NET Framework

- Certains composants sont open-source.
- Compatible uniquement avec **Windows**.
- Possibilité d'implémenter des **API REST** mais pas optimisé pour microservices.
- **WPF** et **WinForms** supportés
- Incompatible avec **MAUI**.
- Moins performant et moins évolutif.

.NET Core

- Open source.
- Compatible avec **Windows, Linux, et Mac OS**.
- Supporte ASP.NET Core, APIs REST, microservices
- **WinForms + WPF** sur **Windows** uniquement
- Support **MAUI** (mobile + desktop).
- Performances hautes.

Quand choisir **.NET Framework** ?

Uniquement si vous devez :

- Maintenir une application Windows existante
- Utiliser des technologies non migrables (**WCF...**)
- Supporter des bibliothèques disponibles uniquement sous **.NET Framework**

© Acti...

Quand choisir **.NET Framework** ?

Uniquement si vous devez :

- Maintenir une application Windows existante
- Utiliser des technologies non migrables (**WCF...**)
- Supporter des bibliothèques disponibles uniquement sous **.NET Framework**

Remarque

Pour tout nouveau projet, utiliser **.NET**.

Microsoft .NET : Écosystème **Microsoft** autour de .NET

- **Microsoft .NET** = { des outils et technologies } **Microsoft** autour de .NET
 - **Windows**
 - **Visual Studio**
 - **.Net Framework**
 - **.Net**

.NET

Microsoft .NET : Écosystème **Microsoft** autour de .NET

- **Microsoft .NET** = { des outils et technologies } **Microsoft** autour de .NET
 - **Windows**
 - **Visual Studio**
 - **.Net Framework**
 - **.Net**

Visual Studio ?

Un IDE (Integrated Development Environment) principal pour développer en **C#, F#, VB.NET**.

Question

Dans un environnement **.NET**, comment créer une application mobile ?

- **Android**
- **iOS**
- **Windows**

© Achref EL M...

Question

Dans un environnement **.NET**, comment créer une application mobile ?

- **Android**
- **iOS**
- **Windows**

Réponse

On peut utiliser

- **Xamarin** (ancien, remplacé depuis 2022, fin de support 2024)
- **.NET MAUI** (recommandé) : framework officiel
- Utilise **C#** et **XAML**

.NET Standard

- Était une spécification permettant de créer des bibliothèques compatibles avec toutes plateformes **.NET : Framework, Core et Xamarin**.
- Avec la sortie de **.NET 5** en 2020, le Standard est devenu obsolète.
- **.NET Standard** est désormais remplacé par le **.NET** lui-même comme plateforme principale de développement et de partage de bibliothèques.

© Achret

.NET

.NET Standard

- Était une spécification permettant de créer des bibliothèques compatibles avec toutes plateformes **.NET : Framework, Core et Xamarin**.
- Avec la sortie de **.NET 5** en 2020, le Standard est devenu obsolète.
- **.NET Standard** est désormais remplacé par le **.NET** lui-même comme plateforme principale de développement et de partage de bibliothèques.

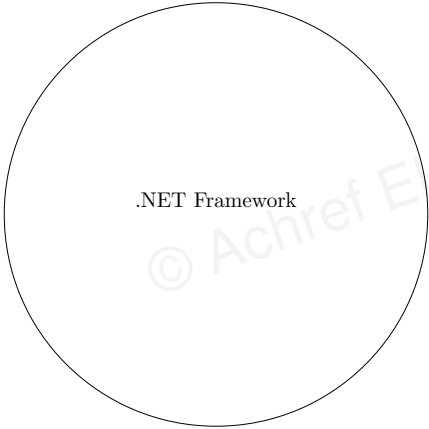
Remarque

Pour qu'une application **.NET Framework** fonctionne sur le runtime **.NET** moderne, **Microsoft** fournit un outil appelé **.NET Upgrade Assistant** pour aider à automatiser une grande partie de ce processus de migration.

Quel problème ?

- Si on développe une bibliothèque de classes dans **.NET Framework**
- On ne peut l'utiliser dans **.NET Core**

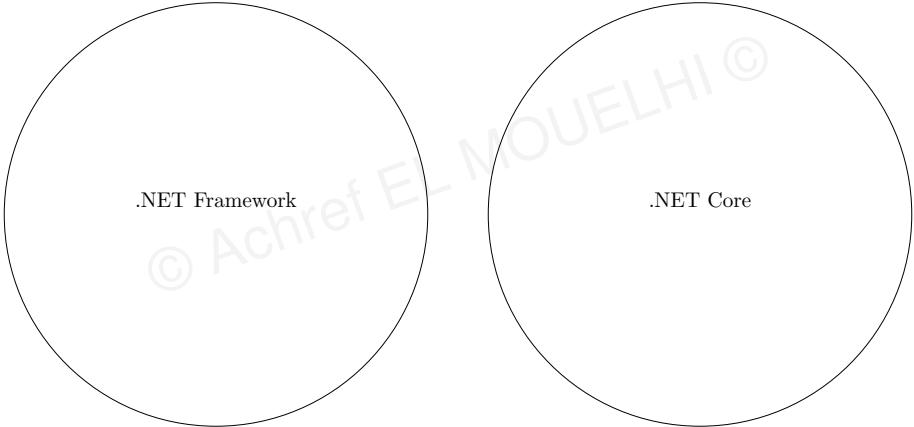
.NET Core vs .NET Framework



.NET Framework

© Achref EL MOUELHI ©

.NET Core vs .NET Framework

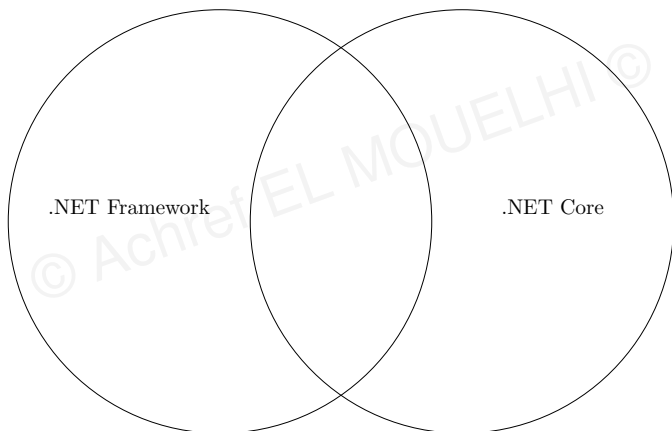


.NET Framework

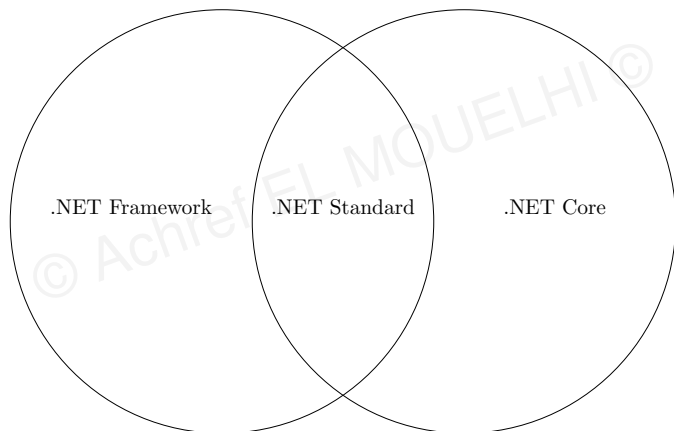
The diagram consists of two large, empty circles positioned side-by-side. The left circle is labeled '.NET Framework' and the right circle is labeled '.NET Core'. A faint, diagonal watermark reading '© Achref EL MOUELHI ©' is visible across both circles.

.NET Core

.NET Core vs .NET Framework



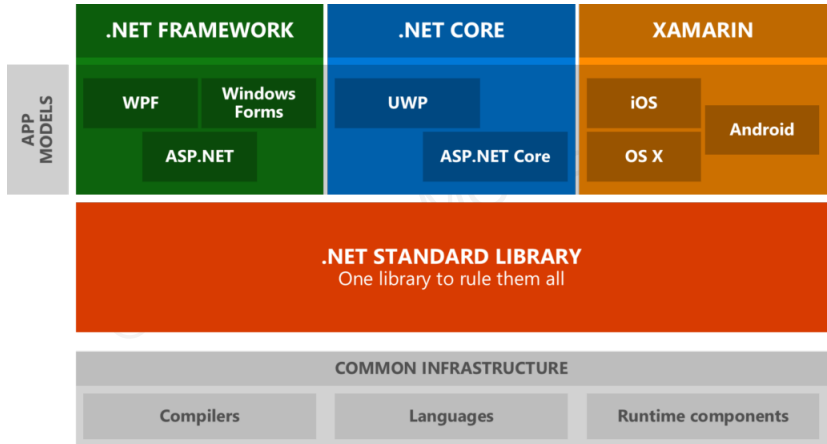
.NET Core vs .NET Framework



Solution Historique/de Transition : **.NET Standard**

- une spécification qui peut être utilisée dans toutes les plateformes **.NET** : **.NET Framework**, **.NET Core** ou **Xamarin**.
- utilisé uniquement pour le développement de bibliothèques de classes.

.NET



.NET

.NET

- Avant la création de **.NET Core** : **.NET** signifiait essentiellement **.NET Framework**
- Depuis **.NET 5** : **.NET** = plateforme unifiée (remplaçant **.NET Framework**, **.NET Core**, **Xamarin**)

Terminologie officielle

- Le passé : **.NET Framework** \Rightarrow ancienne plateforme, limitée à **Windows** (version 4.8 est la dernière majeure).
- La transition : Depuis **.NET Core** \Rightarrow nouvelle plateforme multiplateforme et open-source (versions 1.0 à 3.1).
- L'unification : **.NET** \Rightarrow à partir de la version 5.0, **Microsoft** a abandonné le suffixe **Core** pour signaler la fusion de toutes ses plateformes (**Core**, **Framework**, **Xamarin**).

Remarques

- Lorsque les développeurs disent **.NET moderne**, ils font référence à l'implémentation actuelle, par opposition au **.NET Framework** historique.
- **.NET moderne** = **.NET 5, 6, 7, 8, 9, 10...**
- L'appellation technique officielle de **Microsoft** est simplement **.NET**. Mais pour éviter la confusion avec **.NET Framework**, on dit souvent **.NET moderne**.

.NET

Historique des versions C#, .NET et .NET Framework

Année	Version C#	.NET Core / .NET	.NET Framework
2002	C# 1.0	—	1.0
2003	C# 1.x	—	1.1
2005	C# 2.0	—	2.0
2007	C# 3.0	—	3.0 / 3.5
2010	C# 4.0	—	4.0
2012	C# 5.0	—	4.5 / 4.5.1
2015	C# 6.0	.NET Core 1.0 (preview)	4.6
2016	C# 6.0	.NET Core 1.0 / 1.1	4.6.1 / 4.6.2
2017	C# 7.0 / 7.1	.NET Core 2.0	4.7 / 4.7.1
2018	C# 7.2 / 7.3	.NET Core 2.1 / 2.2	4.7.2
2019	C# 8.0	.NET Core 3.0 / 3.1	4.8 / 4.8.1
2020	C# 9.0	.NET 5.0	—
2021	C# 10.0	.NET 6.0 (LTS)	—
2022	C# 11.0	.NET 7.0	—
2023	C# 12.0	.NET 8.0 (LTS)	—
2024	Pas de version majeure C#	.NET 9	—
2025	C# 13.0	.NET 10 (LTS)	—

Remarques

- Depuis **.NET 5** (2020), **Microsoft** publie une nouvelle version de **.NET** chaque année, en novembre..
- **C#** est aligné sur **.NET** depuis **C# 9** (donc 1 version par an en théorie).
- Certaines années n'ont pas eu de nouvelle version majeure : **C# 13** était prévu en 2024 mais repoussé en 2025.
- Les versions **LTS** (supportées 3 ans) sortent tous les 2 ans, avec les versions paires.

Visual Studio Community 2026 : téléchargement et installation

- Aller sur le lien <https://visualstudio.microsoft.com/fr/downloads/>
- Choisir la version communauté **Visual Studio Community 2026**
- Télécharger puis lancer l'installation

Visual Studio Community 2026 : quelques raccourcis

- `Ctrl` + `K`, `Ctrl` + `D` : formater
- `Ctrl` + `S` : sauvegarder
- `Ctrl` + `K`, `Ctrl` + `C` : commenter (//)
- `Ctrl` + `K`, `Ctrl` + `U` : décommenter
- `Ctrl` + `Shift` + `:` : commenter / décommenter (/ * */)
- `Ctrl` + `F` : chercher
- `Ctrl` + `D` : dupliquer la ligne sélectionnée
- **Autres** : <https://docs.microsoft.com/fr-fr/visualstudio/ide/productivity-shortcuts?view=vs-2026>

Versions de Visual Studio

Année	Version Visual Studio	Remarques
1997	Visual Studio 97	Première version
1998	Visual Studio 6.0	Avant .NET
2002	Visual Studio .NET (2002)	Introduction de .NET Framework 1.0
2003	Visual Studio .NET 2003	.NET Framework 1.1
2005	Visual Studio 2005	.NET Framework 2.0
2008	Visual Studio 2008	.NET Framework 3.5
2010	Visual Studio 2010	.NET Framework 4.0
2012	Visual Studio 2012	.NET Framework 4.5
2015	Visual Studio 2015	Support .NET Framework 4.6
2017	Visual Studio 2017	Support .NET Core 1.x / 2.x
2019	Visual Studio 2019	.NET Core 3.x / début .NET moderne
2022	Visual Studio 2022	Première version 64-bits, support .NET 6/7/8/9
2025	Visual Studio 2026	Nouvelle version majeure, support .NET 10 (et futures)