

MICROSOFT ET L'OPEN SOURCE, COMME CHIEN ET CHAT

artragis, Blue Shark

28 janvier 2016

Table des matières

1	Introduction	5
2	“Les promoteurs du libre ? Des communistes”	7
3	.NET sur Linux, l’histoire mouvementée de Mono	9
4	Le prix des licences en baisse, le web vers le libre	11
5	De la .NET Foundation à la libération du .NET Core	13
6	Sources principales	15

1 Introduction

Le 12 novembre 2014, Microsoft a annoncé¹ qu'une majeure partie du framework .NET serait rendue public sous **licence MIT** afin de promouvoir le développement multiplateforme de site web sous ASP.NET.

Cette nouvelle a été rapidement suivie d'une annonce de la transformation de Visual Studio Professionnal en **Visual Studio Community Edition** sous certaines conditions.

Avant d'en arriver là, Microsoft est passé par plusieurs étapes dans sa réflexion, penchons-nous sur ces dernières.

1. [source officielle](#)

2 “Les promoteurs du libre ? Des communistes”

Alors qu’il était encore PDG de Microsoft, Bill Gates a dû faire face à la montée en puissance des logiciels libres et open source tels que OpenOffice en même temps que le piratage des licences de ses logiciels. Alors qu’il donnait une interview, il estimait alors :

Il existe une nouvelle sorte de communistes modernes qui veut être exemptée de taxes rémunérant les musiciens, les cinéastes et les éditeurs de logiciels.¹ Source : Bill Gates

Depuis sa fondation, Microsoft base son *business model* sur la vente de logiciels. Rapidement la branche Windows a été la plus rentable de toutes suivie de la branche Office.

L’arrivée de Azure (cloud de Microsoft) et des consoles Xbox n’a pas vraiment remis en cause ce modèle qui n’a connu qu’un seul et unique trimestre déficitaire dans **toute son histoire** ! D’après S.Balmer, cela était dû à une forte dépréciation des actifs financiers liés à une filiale spécialisée dans la publicité qui était sensée concurrencer Google.

Depuis sa création, en 1975, Microsoft n’aura connu que trois PDG. L’histoire commence avec Bill Gates amenant la firme qu’il a co-fondée avec P.Allen au sommet de sa gloire jusqu’aux premiers procès antitrust des années 90.

Suite à ces premiers procès antitrust, et désireux de se concentrer sur son engagement humanitaire avec sa femme, B.Gates cède la direction de Microsoft à Steeve Balmer qui y restera jusqu’en 2014 lorsqu’il cèdera sa place à Satya Nadella.

Balmer sera le premier à infléchir la stratégie de Microsoft vers des cycles de développement plus rapides² et à tenter de faire venir un maximum de développeurs vers les technologies Microsoft.

L’arrivée de Satya Nadella à la tête de la firme de Redmond permettra de concrétiser cette inflexion pour faire entrer Microsoft dans une ère plus libre et ouverte à l’open source, bien loin des considérations de B.Gates.

1. En VO et dans son contexte ça donne : “No, I’d say that of the world’s economies, there’s more that believe in intellectual property today than ever. There are fewer communists in the world today than there were. There are some new modern-day sort of communists who want to get rid of the incentive for musicians and moviemakers and software makers under various guises. They don’t think that those incentives should exist.” [source](#).

2. Internet Explorer, par exemple sort désormais une version majeure par an plutôt que tous les 2 ou 3 ans (IE6, comme XP ayant été un ovni) !

3 .NET sur Linux, l'histoire mouvementée de Mono

En 2002, Microsoft décide de réunir les différentes technologies qu'il édite dans un cadre commun qu'il appelle le Framework .NET. Rapidement, les différents langages de Microsoft tels que VB ou Visual C++ seront intégrés à cette plateforme.

La plateforme .NET se base sur une machine virtuelle comparable à la Java Virtual Machine. Microsoft l'appelle CLI (Common Language Infrastructure) et nommera le langage intermédiaire généré par cette machine virtuelle CLR (Common Language Runtime). L'idée est double :

- Proposer, comme pour Java, un environnement de développement multiplateforme (les programmes écrits grâce à la plateforme .NET sont dès le départ disponibles pour Windows 98, 2000, NT, XP et leurs équivalents côté serveur).
- Permettre d'unifier les bibliothèques développées avec les différentes technologies. Ainsi vous pouvez développer un outil en Visual Basic .NET qui utilisera des bibliothèques écrites en Visual C++ ou plus tard en C#. Comme toutes ces technologies utilisent .NET, vous n'avez pas besoin d'adapter quoi que ce soit !

Rapidement, les développeurs remarqueront que le premier objectif n'est pas vraiment atteint : seules les plateformes Windows sont supportées. Une équipe se rassemblera alors autour du projet **Mono**, qui aura pour but d'écrire un interpréteur de CLR pour Linux et Mac.

Heureusement pour les développeurs de Mono, la CLI sera normalisée en majeure partie sous le doux nom de **ECMA335**. Et Mono tentera de coller à cette norme.

Afin de s'assurer que l'outil sera utilisé par un maximum de personne, le créateur de Mono, Miguel de Icaza, fonde la société Ximian afin d'assurer un support client fiable aux utilisateurs de Mono.

Pourtant, dès le commencement, les retards de développement se font sentir¹. Microsoft a un rythme de développement assez élevé et si .NET 1.0 a été publié en 2002, l'année a vu le .NET4.5.2 naître, très loin devant cette première norme.

Microsoft sera durant ces années très peu propice à partager son savoir faire avec l'équipe de Mono, ne participant à son développement que lorsque l'image même de .NET menace d'être écornée si Mono prend trop de retard.

En 2003, peu de temps après sa création, la société Ximian qui était le principal mainteneur de Mono est rachetée par Novell. Avec ce rachat, le développement de Mono va sensiblement s'accélérer.

En 2006, Microsoft attaque Novell, la société qui sponsorisait le développement de Mono à cette époque, pour violation de brevet. Néanmoins, au vu de la popularité grandissante des outils Mono,

1. Aujourd'hui encore le support dit **"Everything in .NET 4.5 except WPF, WWF, and with limited WCF and limited ASP.NET 4.5 async stack."**

notamment utilisés par SecondLife², Microsoft accordera à Novel le droit d'utiliser les technologies développées pour Mono qui imitaient les brevets .NET³.

En 2007, un événement va bouleverser toute la stratégie de Microsoft : Apple sort son iPhone, première version. Le "virage du mobile" doit alors être négocié et ni Windows Mobile, ni Windows Phone 7.5 n'y parviendront.

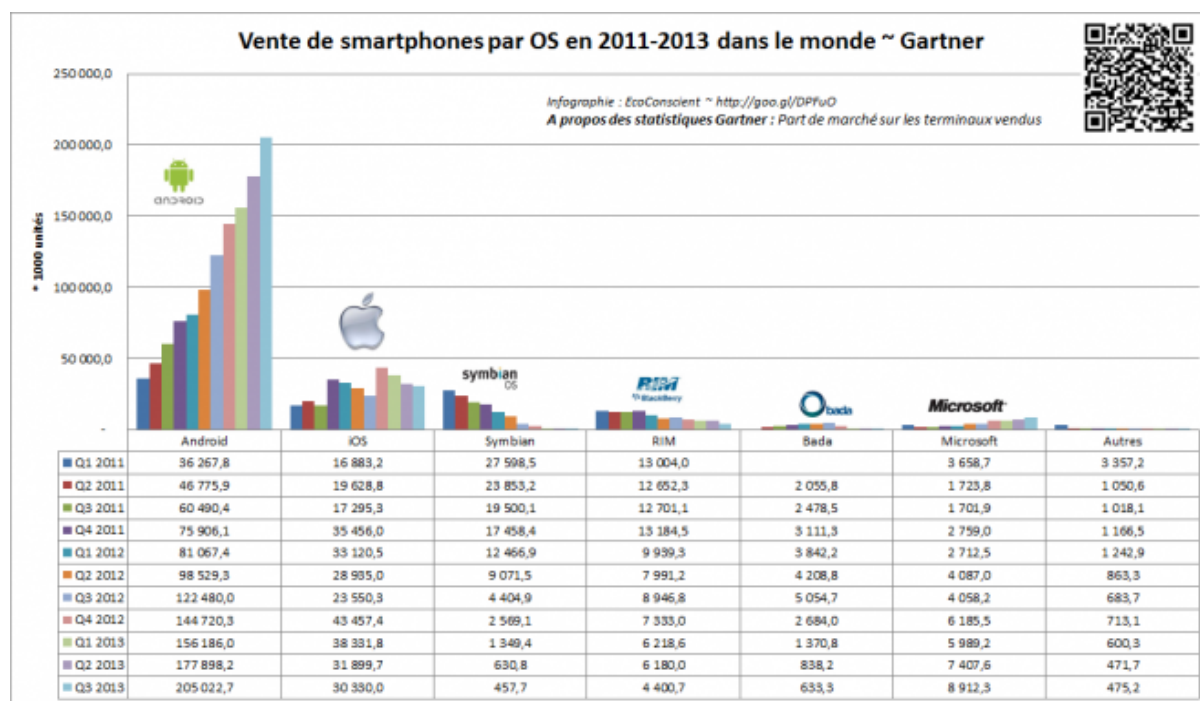


Figure 3.1 – Ventes de mobiles en 2011->2013 (gartner/eco-conscient)

L'avenir de Mono va se jouer en 2011 lorsqu'une firme, nommée Xamarin est créée suite au rachat de Novell par un fond d'investissement, **AtacheMate**.

Cette société a pour but de créer des applications natives sur toutes les plateformes présentes sur le marché (IOS, Android, Windows Phone 7.5 puis 8) à partir d'un code C# compilé depuis Mono justement.

L'année 2011 sera d'autant plus importante qu'Unity3D, le **moteur de création de jeu vidéo** intégrera Javascript et C# comme langage de *scripting*. L'intégration de C# se fera via Mono pour qu'Unity3D soit portable sur les différents OS. De même Sony utilisera C# et Mono pour la PS Suite.

Voyant dans cette société un potentiel énorme de création d'applications pour leur store alors peu fourni en comparaison avec la concurrence, Microsoft fait de Xamarin un de ses principaux partenaires dans le développement mobile. Pour la première fois, le projet Mono est soutenu officiellement par Microsoft, ce qui lui permet de voir s'ouvrir de nouvelles portes.

Pour autant, Mono n'assure pas un support complet de ce qu'on peut trouver dans la plateforme .NET. En effet, le support du WPF n'est pas assuré. Microsoft n'a toujours pas donné les sources de cette bibliothèque graphique et la garde pour son propre environnement. De ce fait, Mono se cantonne au vieillissant WinForm qu'elle implémente grâce à **GTK#**.

2. Plus précisément pour le scripting des mods de SecondLife

3. Ce qui a donné lieu à plusieurs **montées de bouclier** dans le monde du libre car seuls Novel et ses clients étaient alors concernés par cet accord.[

4 Le prix des licences en baisse, le web vers le libre

Le premier pas vers une intégration bien plus conséquente du libre dans le développement de logiciels utilisant .NET fut la création du gestionnaire de package NuGet. Installable en tant que plugin depuis Visual Studio 2010, il est intégré par défaut depuis la version 2012.

Ce gestionnaire de package, à l'instar de [pip](#) pour python, [npm](#) pour JS ou [Maven](#) pour Java, permet de résoudre les dépendances dans vos projets mais aussi de vérifier la compatibilité avec les licences.

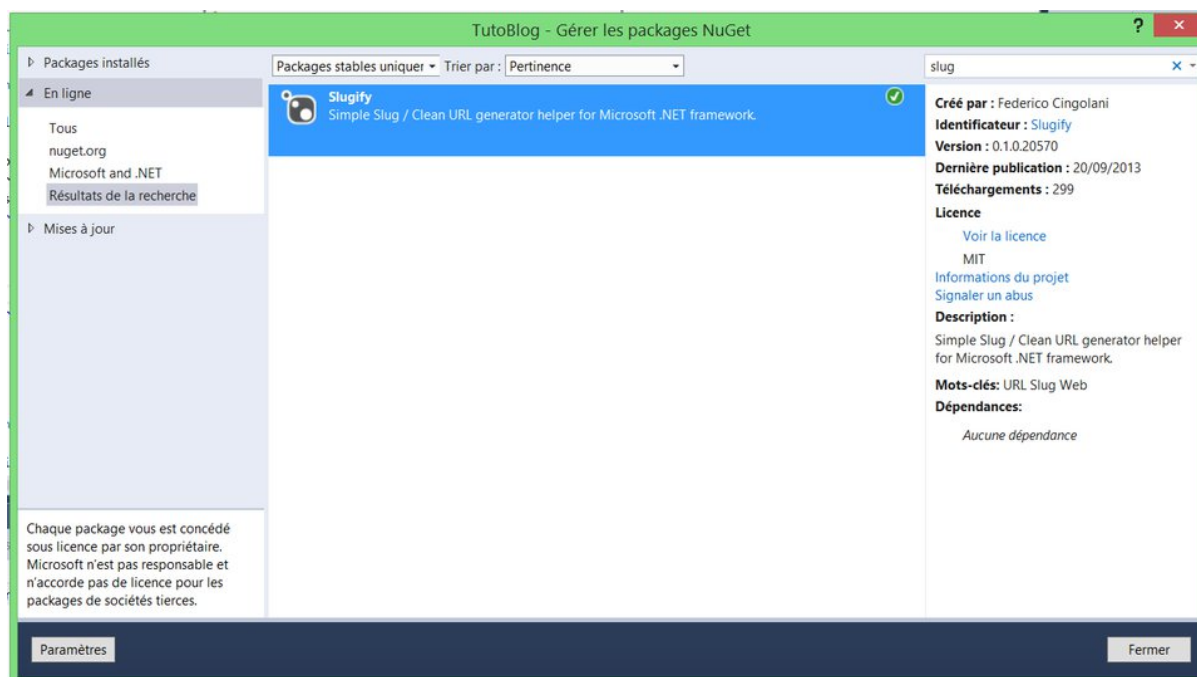


Figure 4.1 – Exemple d'installation d'un package

Lorsque vous créez une bibliothèque de code, le partage de cette bibliothèque, la gestion de ses versions peut facilement se faire sur un dépôt NuGet privé, sur [nuget.org](#) ou sur un miroir personnel puisque le code de nuget.org est lui même [libre](#). L'exploration de NuGet ou d'un de ses miroirs étant fortement facilitée par la mise en place d'une [API REST](#).

Depuis l'arrivée de Satya Nadella à la tête de Microsoft, une stratégie assumée d'ouverture se met en place. Bien que le PDG de la firme de Redmond soit encore parfois accusé d'utiliser la célèbre stratégie des 3E "Embrace Extend Extinguish"¹, plusieurs actions sont venues rassurer le monde du libre.

1. Adopte, étends et extermine, la stratégie d'introduction des produits par [Microsoft](#)

Cette stratégie commerciale s'appuie sur un changement radical de politique de vente : toute licence OEM² vendue sur un *device* dont l'écran fait moins de 10 pouces est gratuite. De plus, la direction de Microsoft adhère au constat qu'aujourd'hui, l'open source est passé de "toléré" à "prévu".

Dans cette optique, le développement de ASP.NET MVC ainsi que d'autres modules tels que SignalR a dès le départ été rendus *open source*.



Figure 4.2 – SignalR, le framework de temps réel pour ASP.NET

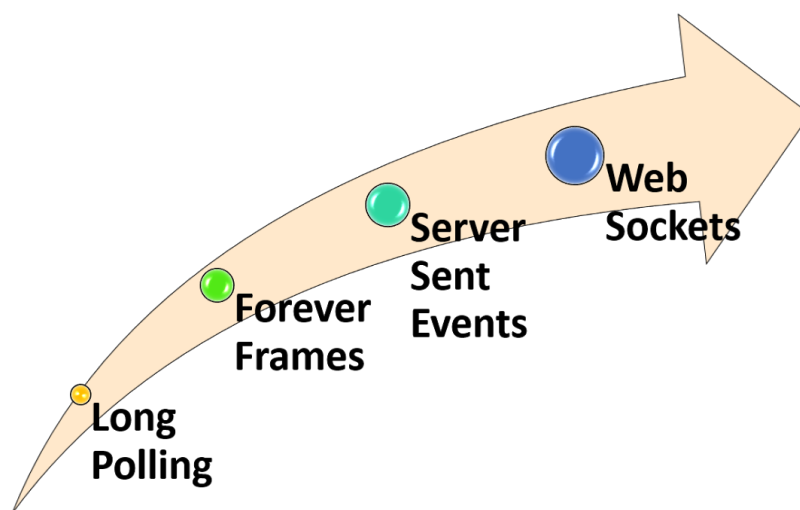


Figure 4.3 – SignalR, le framework de temps réel pour ASP.NET

2. L'acronyme originel signifie *Original Equipment Manufacturer* et désigne dans l'industrie les pièces détachées vendues dans des packages minimalistes. Les licences "OEM" ont été mises en place pour limiter la réutilisation des logiciels préinstallés dans les ordinateurs grand public. Si vous changez l'ordinateur, vous n'avez pas le droit d'utiliser les logiciels OEM de l'ancien. Cette pratique est fortement décriée dans le monde du libre.

5 De la .NET Foundation à la libération du .NET Core

Nous arrivons donc en 2014. Microsoft publie Windows 8.1 suite à l'échec relatif de la version 8.0, de même les WindowsPhone équipés de la version 8 de toutes les gammes sont désignés comme éligibles à WindowsPhone 8.1. Microsoft annonce la création des **applications universelles**. L'expérience de la mise en open source de ASP.NET MVC s'étant révélée positive, Microsoft décide de créer en collaboration avec Xamarin la .NET Foundation.

Cette fondation réunit les **projets** considérés comme majeurs par ses fondateurs, Xamarin en pilotant plus de dix !

L'un des plus gros projets de cette fondation est la création d'un compilateur open source utilisant au maximum les technologies de **compilation à la volée** pour :

- Augmenter fortement les performances
- Améliorer l'efficacité, la justesse et l'intégration des systèmes d'auto complétion dans les IDE
- Permettre par la suite d'envoyer un programme utilisant .NET qui n'a pas besoin de l'installation complète du framework pour être partagé car le code sera natif.

C'est à partir de la .NET Foundation que la stratégie d'ouverture de Microsoft passe désormais et, il y a peu, une partie du framework a été complètement ouverte et publiée sous licence MIT, soit l'une des plus permissives !

Pour comprendre quelles parties ont été publiées, le mieux reste de vous fournir le **schéma officiel** :

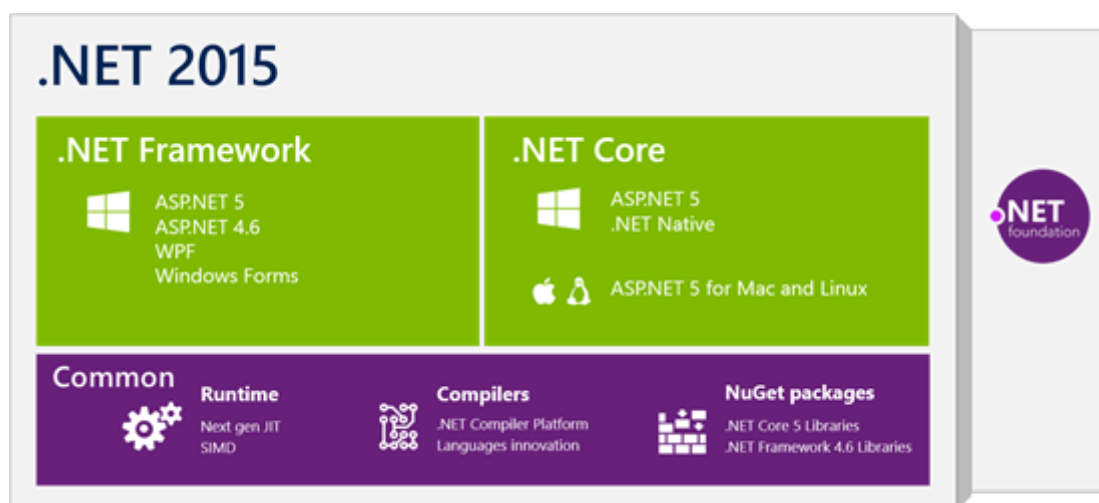


Figure 5.1 – Architecture du .NET Core

Ce qu'il faut comprendre c'est que Microsoft lie toujours très fortement ses bibliothèques graphiques (WinForm et WPF) à son système d'exploitation et refuse donc d'en libérer les sources pour aider les équipes de Mono. Le but est clairement d'éviter que les distributions Linux puissent copier l'identité visuelle que l'on trouve depuis les version 2010 d'office et 2012 de Visual Studio.

Notons toutefois qu'avec la politique visant à unifier les applications sur tout l'écosystème Microsoft, l'entreprise continue sa stratégie de libération de ses produits de développement. Le dernier en date est **Chakra**, le moteur javascript intégré à Edge.

6 Sources principales

- Annonce officielle de Microsoft ([en](#))
- Détail des apports ([fr](#), [NextINpact](#))
- Fiche Wikipédia de [Microsoft](#)
- Le blog de Xamarin ([en](#))
- Article de developpez.com sur Roslyn ([fr](#))
- Le github de .NET ([en](#))

*[PDG] : Président Directeur Général, il dirige le conseil d'administration (président) et l'entreprise au quotidien (directeur général). [CLI] : Common Language Infrastructure [CLR] : Common Language Runtime [.NET] : A prononcer "dote nette", bien qu'aucun mariage ne soit en jeu. [WPF] : Windows Presentation Foundation : un framework qui permet de créer des interfaces graphiques modernes qui s'appuie beaucoup sur la programmation asynchrone**