

# Les Plate-formes .NET et Compact .NET

Jean-Yves Tigli, [tigli@polytech.unice.fr](mailto:tigli@polytech.unice.fr)

<http://www.tigli.fr>

Polytech'Nice Sophia Antipolis – SI5 – Master IFI  
Université de Nice – Sophia Antipolis



# Un survol de .NET

- ✓ Introduction
- ✓ Services Web
- ✓ Le framework
- ✓ Le C.L.R (Common Language Runtime)
- ✓ Les formulaires Windows
- ✓ Les formulaires Web
- ✓ L'accès aux données (ADO.NET)
- ✓ Les langages

# .NET

## C'est quoi

- ✓ **Une vision (celle de Bill & Co) de l'évolution des technologies du Web**
  - Des sites Web offrent des services Web qui coopèrent
  - De nouveaux terminaux complètent les PCs
  - Les IHM deviennent adaptable et personnalisable
  - Respect a priori des standards du Web

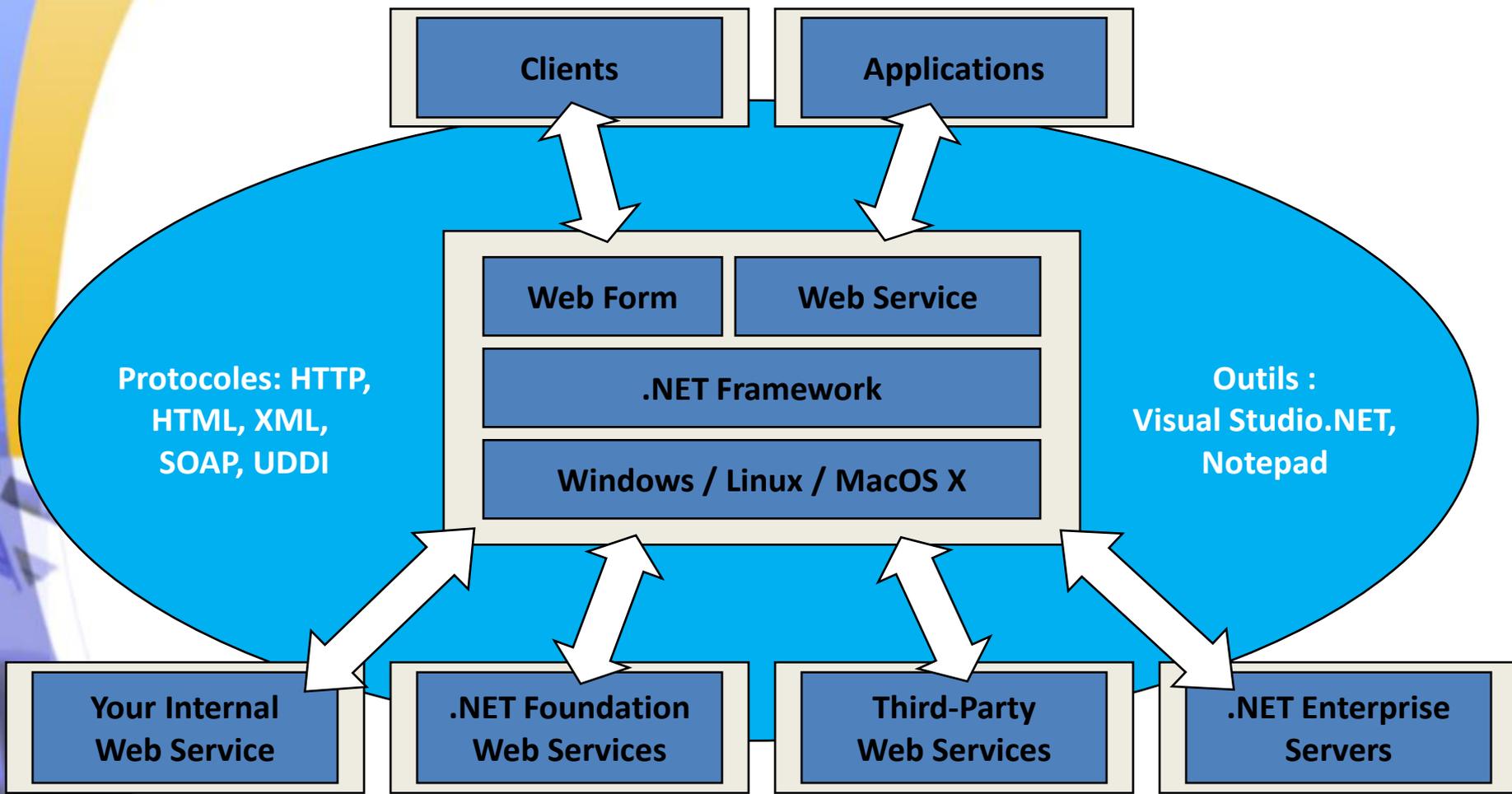
# .NET C'est quoi

- ✓ Une vision (celle de Bill & Co) de l'évolution des technologies du Web
- ✓ Une plate-forme qui supporte cette vision
  - .NET Framework et Visual Studio.NET
  - .NET Enterprise Servers
    - Base de données, messages, Intégration, proxy, sécurité, mobilité, gestion du contenu, orchestration des services, ...
  - .NET Building Block Services
    - Passport et .NET My Services ("Hailstorm")
  - **Objectif** : rendre la construction d'applications Web aisées (par assemblage de Web services)

# .NET C'est quoi

- ✓ Une vision (celle de Bill & Co) de l'évolution des technologies du Web
- ✓ Une plate-forme qui supporte cette vision
- ✓ Un modèle pour offrir des services (et faire du business)
  - Les applications sont des services
  - Abonnement à des services
  - Référencement d'applications

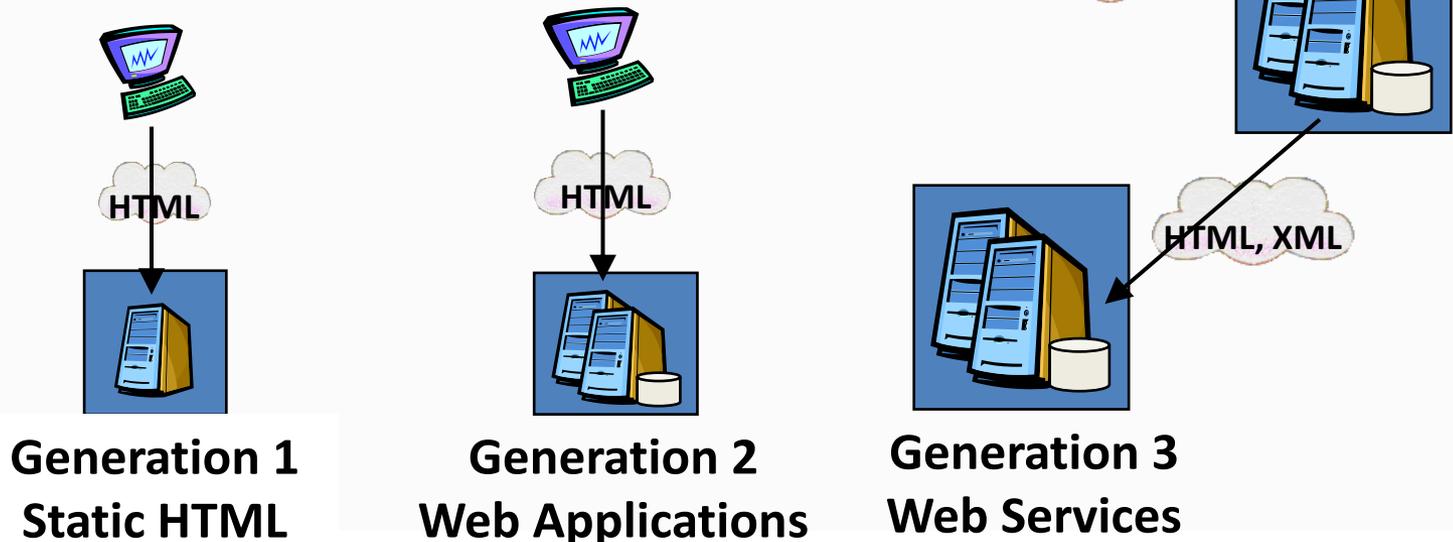
# La plate-forme .NET



# Services Web

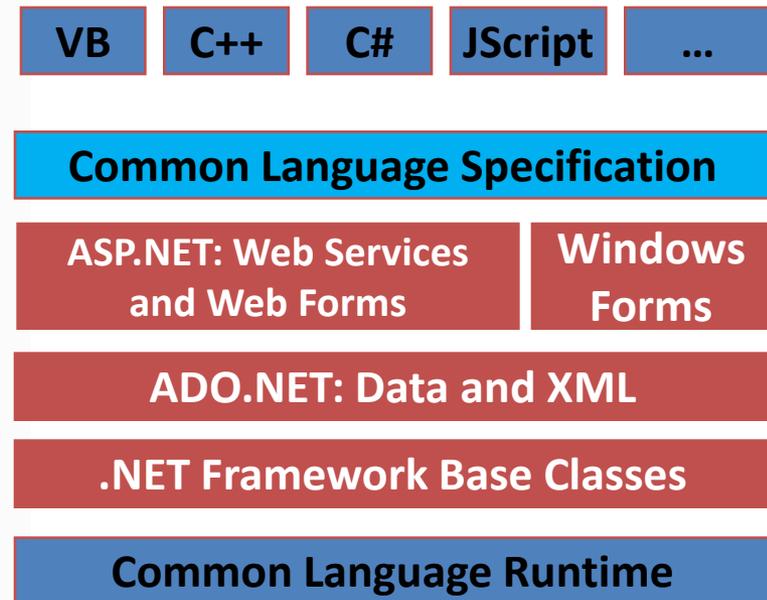
## Une évolution du Web

- ✓ Des composants applicatifs accessibles à partir des standards du Web
- ✓ Le cœur de l'architecture .NET
- ✓ Offrir des services à l'aide du Web
- ✓ Construit avec les standards actuels
  - HTTP, XML, SOAP, UDDI, WSDL, ...



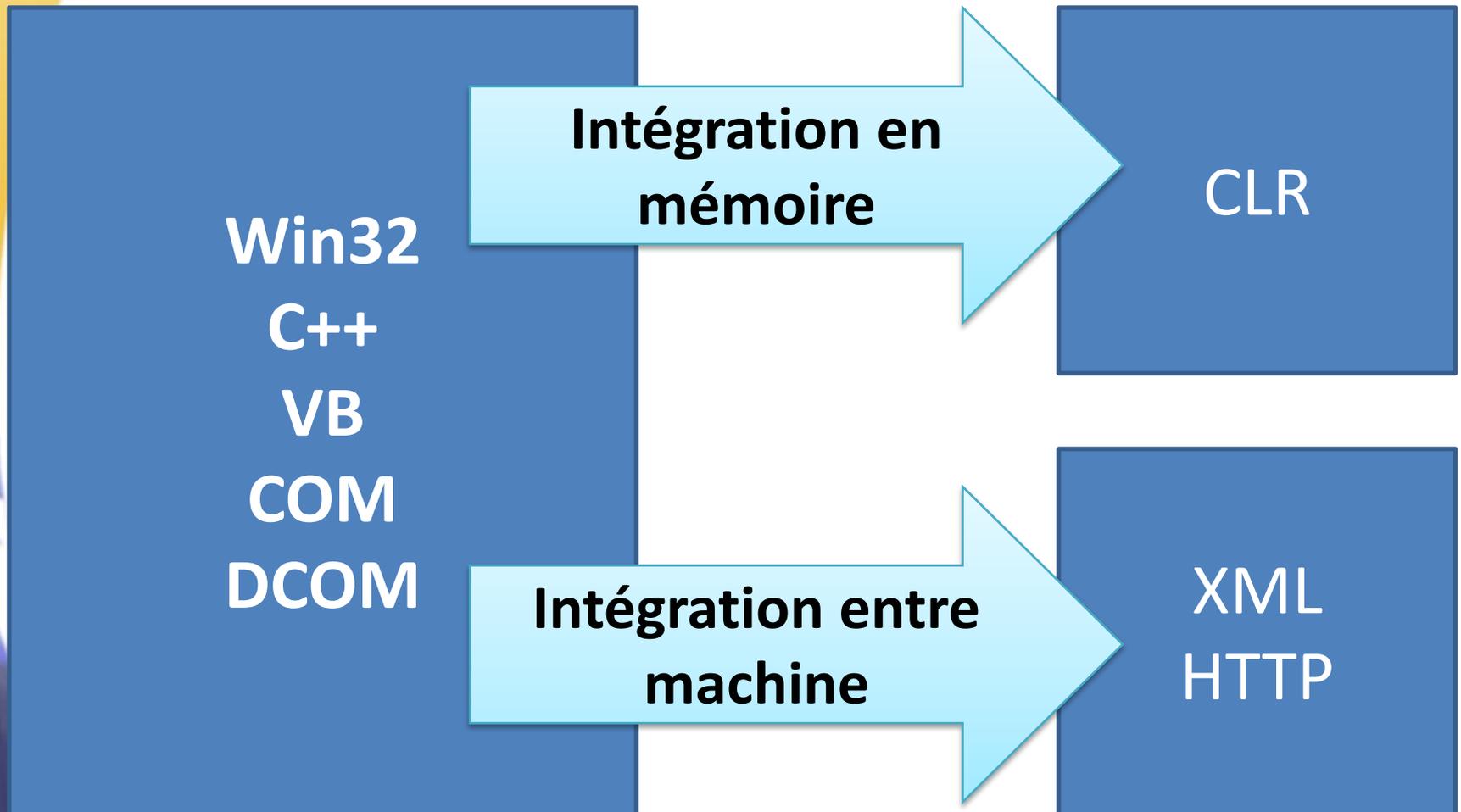
# Le framework .NET

- ✓ Un ensemble de technologies pour développer et utiliser des composants :
  - Formulaires Web
  - Services Web
  - Applications Windows
- ✓ Des outils pour développer des applications
  - Développement
  - Mise au point
  - Déploiement
  - Maintenance

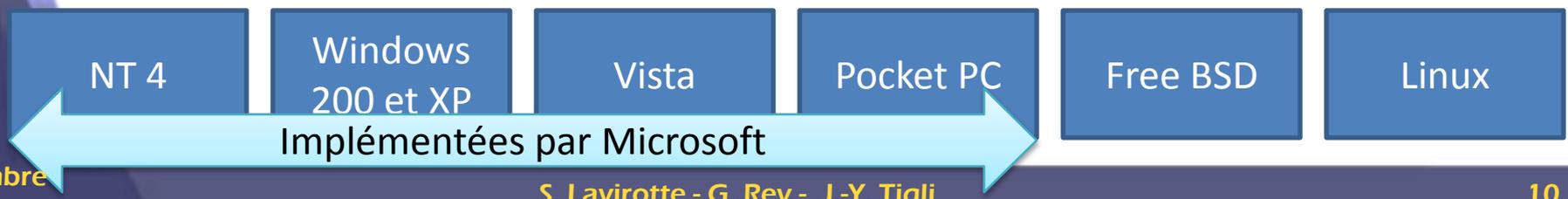
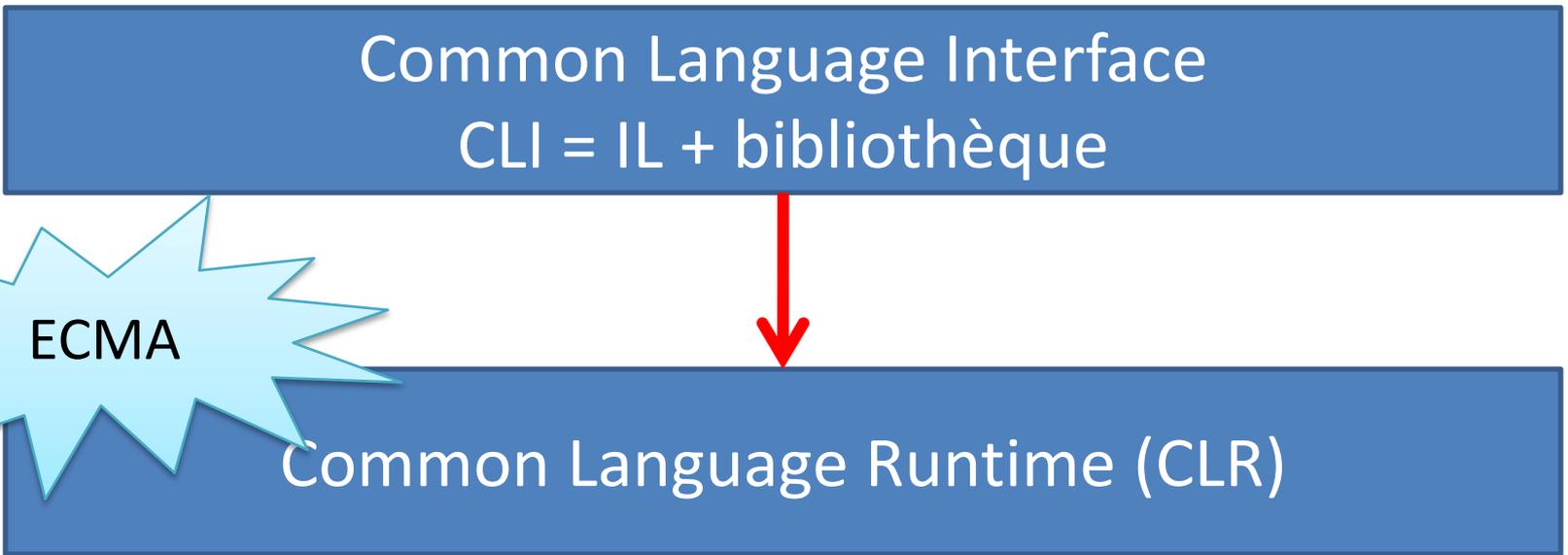


Visual Studio.NET

# La plateforme .NET

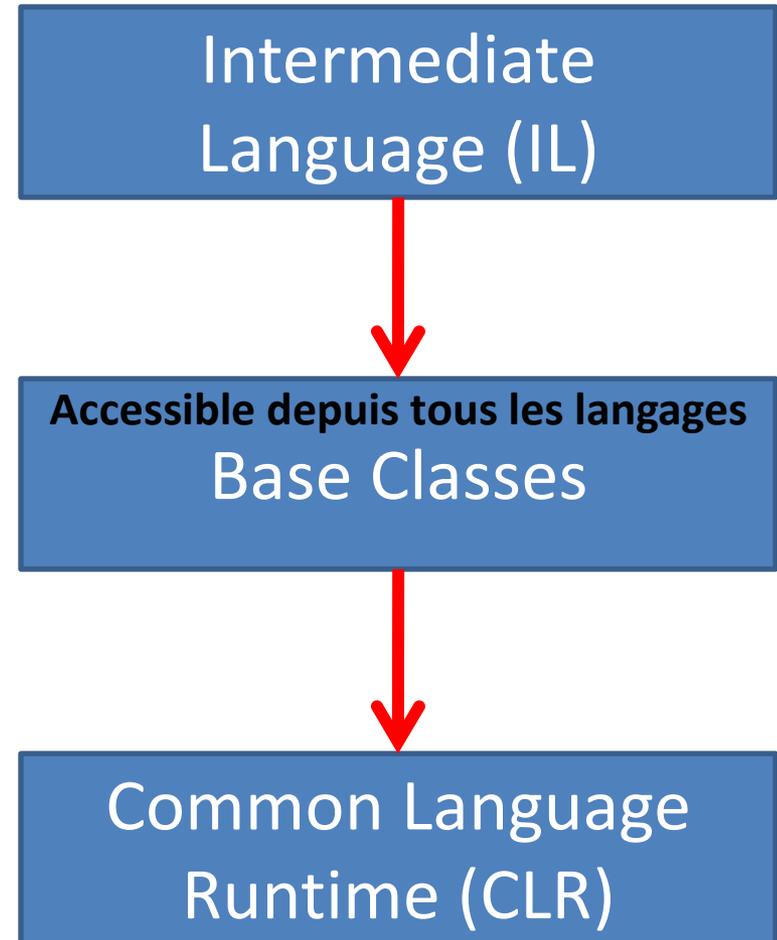


# La plateforme .NET



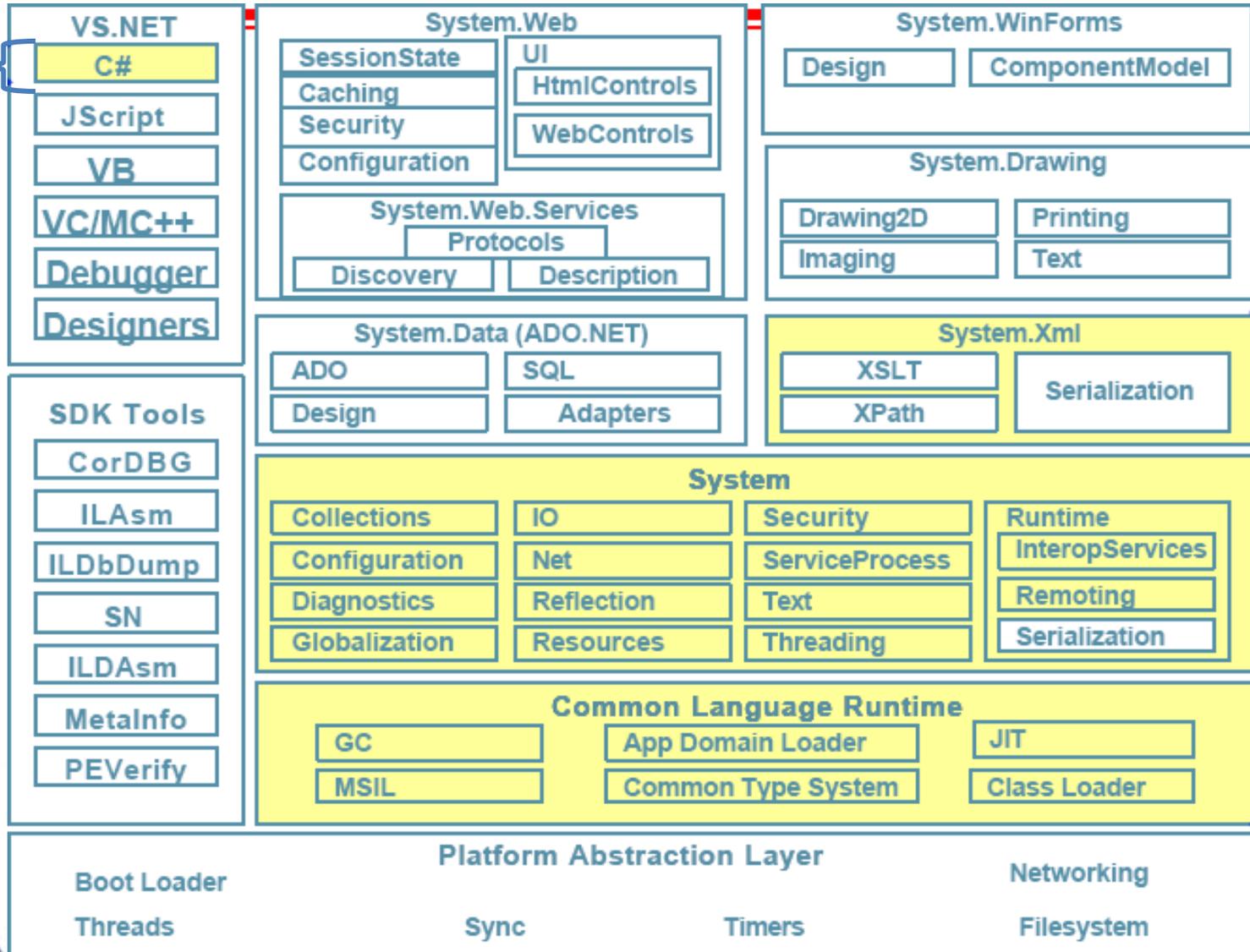
# Le .Net Framework

- ✓ Entre le langage intermédiaire (IL) et le machine virtuelle (CLR) se trouvent les classes du .Net Framework
- ✓ Ces classes sont accessibles par tous les langages puisque accessibles depuis l'IL
- ✓ Une partie de ces classes ont été intégrées dans les spécifications du CLR
  - Modèle d'objet commun
  - Types unifiés ⚠
  - Meta-Data
  - Modèle d'exception commun



# CLI Standards

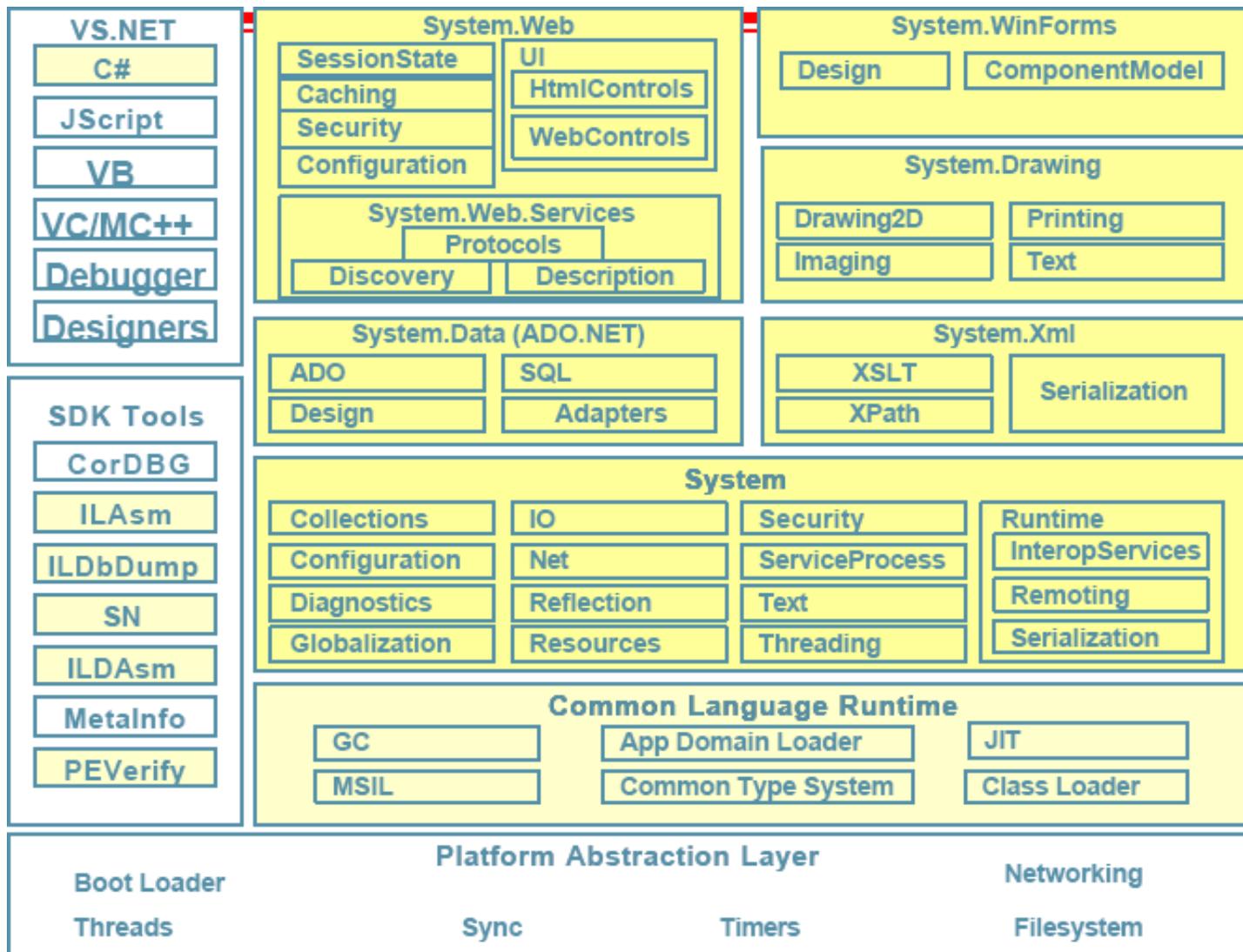
ECMA 334



ECMA 335

# Le framework .NET

## Les classes du framework



# Le framework .NET

## Les classes du framework

### System.Web

#### Services

Description

Discovery

Protocols

Caching

Configuration

#### UI

HtmlControls

WebControls

Security

SessionState

### System.Windows.Forms

Form

MessageBox

Button

ListControl

### System.Drawing

Drawing2D

Imaging

Printing

Text

### System.Data

OleDb

Design

SQL

SQLTypes

### System.Xml

XSLT

XPath

Serialization

### System

Collections

Configuration

Diagnostics

Globalization

IO

Net

Reflection

Resources

Security

ServiceProcess

Text

Threading

Runtime

InteropServices

Remoting

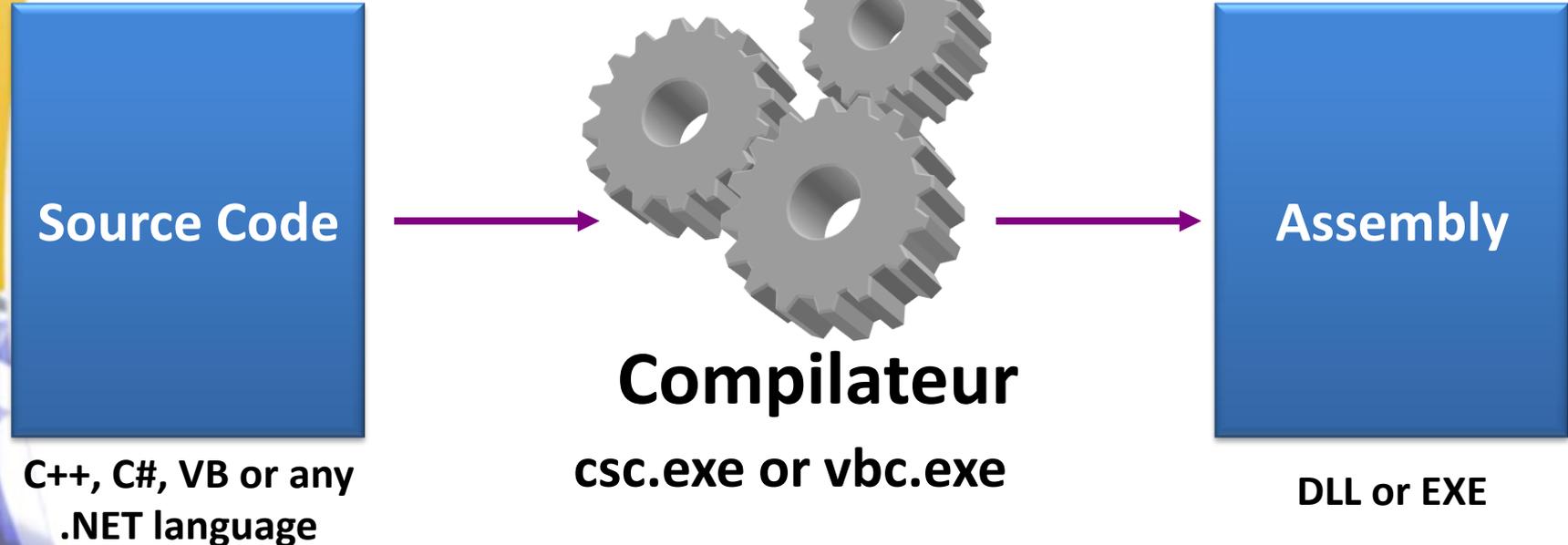
Serialization

# Common Language Runtime

## Buts

- ✓ **Développement de services**
  - Permettre une interopérabilité entre les langages
  - Augmenter la productivité (chacun écrit dans le langage qu'il maîtrise, le plus adapté, ...)
- ✓ **Services déploiement**
  - Simple, sûr
  - Gestion des versions – NO MORE 'DLL HELL'
- ✓ **Services à l'exécution**
  - Performance
  - Extensibilité
  - Sûreté et disponibilité
  - Sécurité

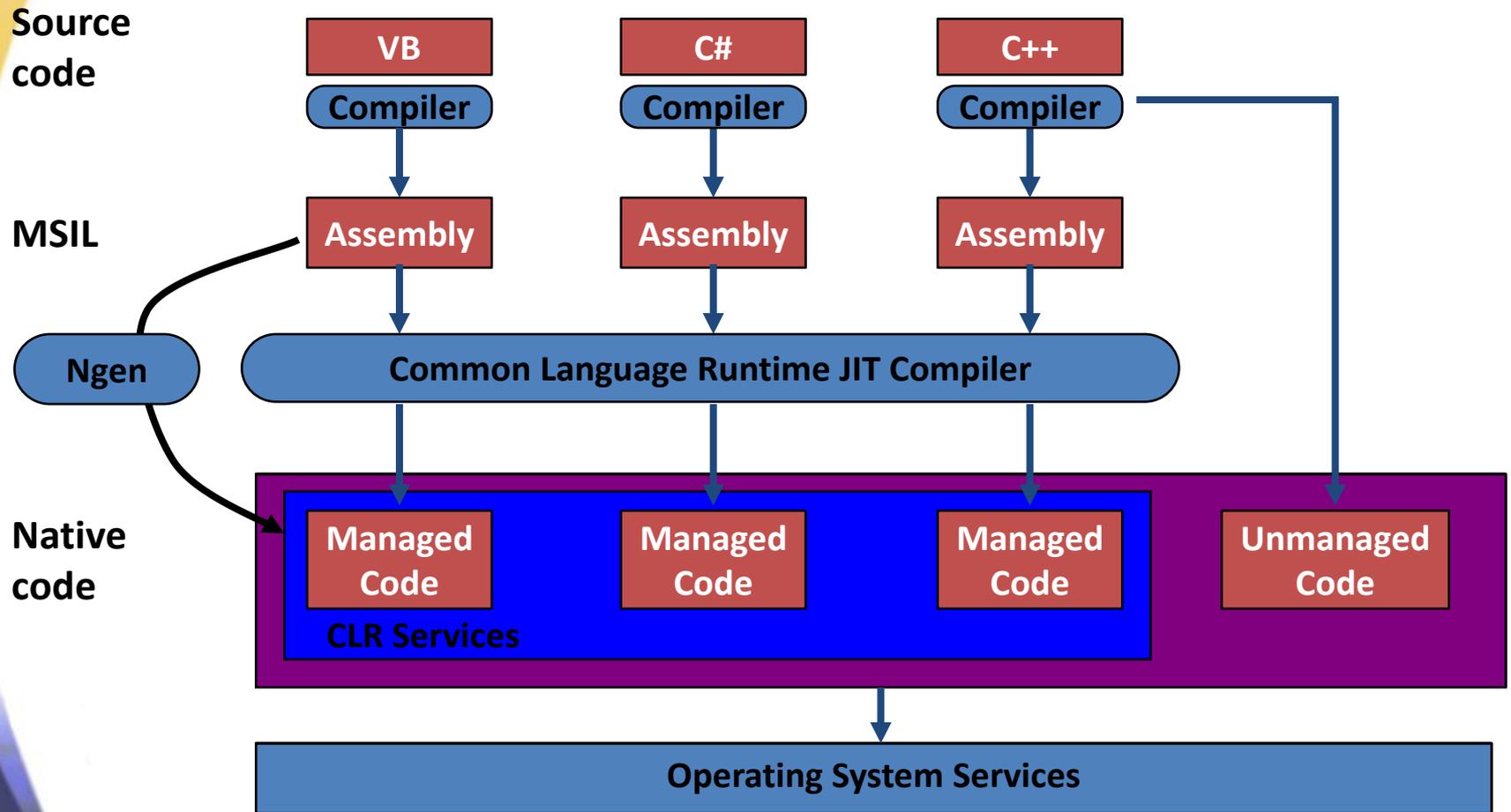
# Common Language Runtime Compilation



# Common Language Runtime Assemblies

- ✓ **Assembly : archive**
  - Unité logique de déploiement
  - Contient Manifest, Metadata, MSIL and ressources
- ✓ **Manifest**
  - Metadata à propos des composants présent dans un assembly (version, types, dépendences, etc.)
- ✓ **Type Metadata**
  - Définition complète de tous les types présent dans l'assembly : attributs, méthodes, paramètres, ressources...
- ✓ **Microsoft Intermediate Language (MSIL, IL)**
  - Tous les langages sont compilés en IL (managed code)
  - IL est toujours compilé en code natif avant exécution (JIT compiler)
- ✓ **Ressources**
  - .bmp, .jpg

# Common Language Runtime Modèle d'exécution



# Common Language Runtime Services

- ✓ Gestion du Code
- ✓ Conversion du code MSIL en natif
- ✓ Chargement et exécution du 'managed code'
- ✓ Création et gestion des metadata
- ✓ Contrôle des types
- ✓ Insertion et exécution des politiques de sécurité
- ✓ Gestion mémoire
- ✓ Gestion multi-langage des exceptions
- ✓ Interopérabilité entre les objets du framework .NET et les objets COM et les DLLs Win32
- ✓ Liaison tardive
- ✓ Services pour les développeurs (profiling, debugging, etc.)

# Common Language Runtime et ses implémentations

- ✓ Framework .Net est suffisant pour programmer, c'est la version officielle
  - pas accès aux sources
- ✓ ROTOR est une implémentation Microsoft de la technologie .Net
  - <http://msdn.microsoft.com/sscli>
  - implémentation complète de standard ISO/ECMA (c'est un sur ensemble de la norme)
  - Fonctionne avec FreeBSD, Mac OS X et Windows XP (License de type BSD)
  - 100% du code source est disponible
  - Il n'y a pas d'implémentation correcte de ROTOR pour Linux
- ✓ Ximian : MONO est un projet indépendant
  - <http://www.go-mono.org>
  - compatibilité avec les produits .Net
  - fonctionne sur Linux (licence GPL)
  - actuellement : C#, JIT très efficace, pas d'IDE
- ✓ DotGNU : Portable .Net
  - [http://www.southern-storm.com.au/portable\\_net.html](http://www.southern-storm.com.au/portable_net.html)
  - implémentation du standard ISO/ECMA
  - fonctionne sous GNU/Linux, c'est un projet GNU
    - mais aussi sous Windows, Solaris, NetBSD, FreeBSD, and MacOS X.
    - le runtime a été testé sur les processeurs x86, PowerPC, ARM, Sparc, PARISC, s309, Alpha, and IA-64
  - Actuellement : C#, pas de JIT, de nombreux outils

# Languages C#

- ✓ **Nouveau langage créé pour .NET**
- ✓ **Evolution de C++ (comme Java)**
- ✓ **Concepts clés :**
  - Composants orientés
  - Tout est objet
- ✓ **Soumis à l'ECMA pour standardisation**
- ✓ **Utilise les classes du framework .NET**
  
- ✓ **Les types non signés de C# ne sont pas conformes au CLS**

# Des informations complémentaires ?

## ✓ HTTP

- <http://msdn2.microsoft.com/en-us/library/aa767734.aspx>
- HTTP Essentials, Stephen Thomas, 2001, Wiley, ISBN 0471-39823-3

## ✓ Cookies

- <http://msdn2.microsoft.com/en-us/library/ms970174.aspx>

## ✓ MIME

- [http://www.ufaq.org/navcom/mime\\_tutorial.html](http://www.ufaq.org/navcom/mime_tutorial.html)
- <http://www.irvine.com/~mime/>

## ✓ Networks

- <http://msdn.microsoft.com/library/periodic/period99/ntp99b3.htm>

# Des informations complémentaires ?

## ✓ XML

- <http://msdn.microsoft.com/xml/default.asp>
- <http://www.w3.org/XML/>
- Essential XML, Don Box, Aaron Skonnard, John Lam, Addison Wesley, 2000, ISBN 0-201-70914-7

## ✓ .NET

- <http://www.microsoft.com/net/>
- <http://msdn.microsoft.com/net/>
- <http://www.gotdotnet.com>
- msnews.microsoft.com news server
  - microsoft.public.dotnet.general newsgroup

# Du framework .net au compact framework .net

Jean-Yves Tigli, [tigli@polytech.unice.fr](mailto:tigli@polytech.unice.fr)

<http://www.tigli.fr>

Polytech'Nice Sophia Antipolis – SI5 – Master  
IFI

Université de Nice – Sophia Antipolis



- ✓ **1. Présentation**
- ✓ **2. Comparaison avec le Framework .NET**
  - a. Ressemblances avec le Framework .NET
  - b. Différences avec le Framework .NET



# Présentation

- ✓ Le Compact Framework .NET est un sous-ensemble de la bibliothèque de classes du Framework .NET, mais contient également des classes spécialement conçues à son intention.
- ✓ Il s'agit d'un environnement indépendant du matériel permettant d'exécuter des programmes sur divers périphériques informatiques à ressources limitées :
  - ✓ assistants numériques personnels (PDA, Personal Digital Assistant) tels que le Pocket PC, téléphones mobiles, décodeurs, périphériques informatiques automobiles et périphériques personnalisés intégrés au système d'exploitation Windows CE .NET.



# Présentation

- ✓ Pour simplifier, on peut dire que le Compact Framework .NET apporte aux périphériques la puissance de développement du Framework .NET.
- ✓ Le Compact Framework .NET offre également les avantages suivants :
  - Exécution de programmes indépendants du matériel et des systèmes d'exploitation
  - Prise en charge de protocoles réseaux courants et connexion transparente avec les services Web XML
  - Performances optimales dans la génération de code natif à l'aide de la compilation juste-à-temps

# Comparaison avec le Framework .NET

Ressemblances avec le Framework .NET

Différences avec le Framework .NET

# Ressemblances avec le Framework .NET

- ✓ **Le Compact Framework .NET possède des ressemblances avec le .NET Framework.**
- ✓ **Common Language Runtime**
  - Dans les deux infrastructures, les CLR (Common Language Runtime) tirent parti de l'exécution de code managé, de la compilation de code juste-à-temps (JIT, Just In Time) et du garbage collector.
  - Ils prennent en charge la Spécification de langage commun (CLS, Common Language Specification).
- ✓ **Assemblies et formats de fichier**
  - Dans les deux infrastructures, les applications utilisent les assemblies.

# Ressemblances avec le Framework .NET

- ✓ **Prise en charge de plusieurs langages**
  - L'interopérabilité interlangage fait partie intégrante du .NET Compact Framework.
  - Visual C# et Visual Basic .NET sont les deux premiers langages pris en charge.
  
- ✓ **Respect des normes**
  - La technologie de base implémentée dans les deux infrastructures est conforme à la norme CLI (Common Language Infrastructure) de l'association ECMA (European Computer Manufacturers Association).

# Différences avec le Framework .NET

- ✓ Il existe beaucoup de différences entre le Compact Framework .NET et le Framework .NET
- ✓ COM interop et fonctions de rappel
  - Le Compact Framework .NET n'assure pas l'interopérabilité (versions 1.0 et 2.0) avec les objets COM, mais vous pouvez utiliser PInvoke (platform invoke) pour accéder à des fonctions DLL natives qui, à leur tour, peuvent appeler des objets COM.
- ✓ Répertoire en cours
  - La fonctionnalité de répertoire en cours est absente sur les périphériques qui exécutent Windows CE .NET ; par conséquent, le .NET Compact Framework ne prend pas en charge les méthodes `Directory.GetCurrentDirectory` et `Directory.SetCurrentDirectory`.

# Différences avec le Framework .NET

## ✓ Données

- Le Compact Framework .NET fournit une implémentation de sous-ensemble de ADO.NET et inclut le fournisseur de données SQL Server CE .NET. L'espace de noms [System.Data.OleDb](#) n'est pas pris en charge.

## ✓ Tableaux

- Au contraire de certains langages, le Common Language Runtime ne prend pas en charge les limites inférieures autres que zéro et lève une exception [MissingMethodException](#) si le premier élément d'un tableau n'est pas zéro.

# Différences avec le Framework .NET

## ✓ ASP.NET

- Le Compact Framework .NET est essentiellement une plateforme de client élaboré et ne prend pas en charge ASP.NET.

## ✓ Assemblys et GAC (Global Assembly Cache)

- Dans les version 1.0 et 2.0, le .NET Compact Framework ne prend pas en charge les assemblies composés de plusieurs modules, mais prend en charge les assemblies satellites.

## ✓ Classes

- Le .NET Compact Framework prend en charge un sous-ensemble de la Bibliothèque de classes .NET Framework.

# Différences avec le Framework .NET

- ✓ **Types de données et précision en virgule flottante**
  - Seule `Math.Round(double a)` est prise en charge ;  
`Math.Round(double a, double b)` n'est pas prise en charge.
  
- ✓ **Délégués**
  - Les délégués asynchrones, en particulier les méthodes `BeginInvoke` et `EndInvoke`, ne sont pas pris en charge.
  
- ✓ **Événements**
  - Le .NET Compact Framework prend en charge les événements `GotFocus` et `LostFocus`, mais non les événements `Activated` et `Deactivated`.

# Différences avec le Framework .NET

- ✓ **Installation et fichiers CAB**
  - Vous pouvez utiliser des fichiers CAB et créer des applications MSI pour distribuer vos applications.
  
- ✓ **Test des performances**
  - Le .NET Compact Framework ne prend en charge ni le profilage de code, ni le fichier Perfmon.exe du Moniteur système.
  
- ✓ **Réflexion**
  - Le .NET Framework ne prend pas en charge l'espace de noms [System.Reflection.Emit](#).

# Différences avec le Framework .NET

## ✓ Sérialisation

- Pour des raisons de taille et de performances, le .NET Compact Framework ne prend en charge ni la sérialisation binaire à l'aide de BinaryFormatter, ni la sérialisation SOAP à l'aide de SoapFormatter.

## ✓ Minuterias

- Les méthodes `Timer.Start` et `Timer.Stop` ne sont pas prises en charge, mais vous pouvez démarrer et arrêter une minuterie en affectant à la propriété `Timer.Enabled` la valeur `true` or `false`.