

# INTRODUCTION FRAMEWORK.NET

UTOPIOS



ANTHONY DI PERSIO



## DECOUVERTE DU .NET

Première approche du Framework .NET

01

## QU'EST-CE QUE LA CLS ?

Définition de la « Common Language Specification »

02

## QU'EST-CE QUE LE CTS ?

Comprendre le « Common Type System »

03

## LA BASE CLASS LIBRARY

Comprendre la BCL et son fonctionnement

04

## TABLE DES MATIÈRES

05

## DÉFINITION DU CIL

Comprendre le « Common Intermediate Language »

06

## ÉTAPE D'ASSEMBLAGE

Comprendre l'assembly, sa composition et métadonnées

07

## L'ENVIRONNEMENT CLR

Utilisation de la « Common Language Runtime » en .NET

08

## LES SPÉCIFICATIONS .NET

Comprendre les spécifications .NET et .NET Core



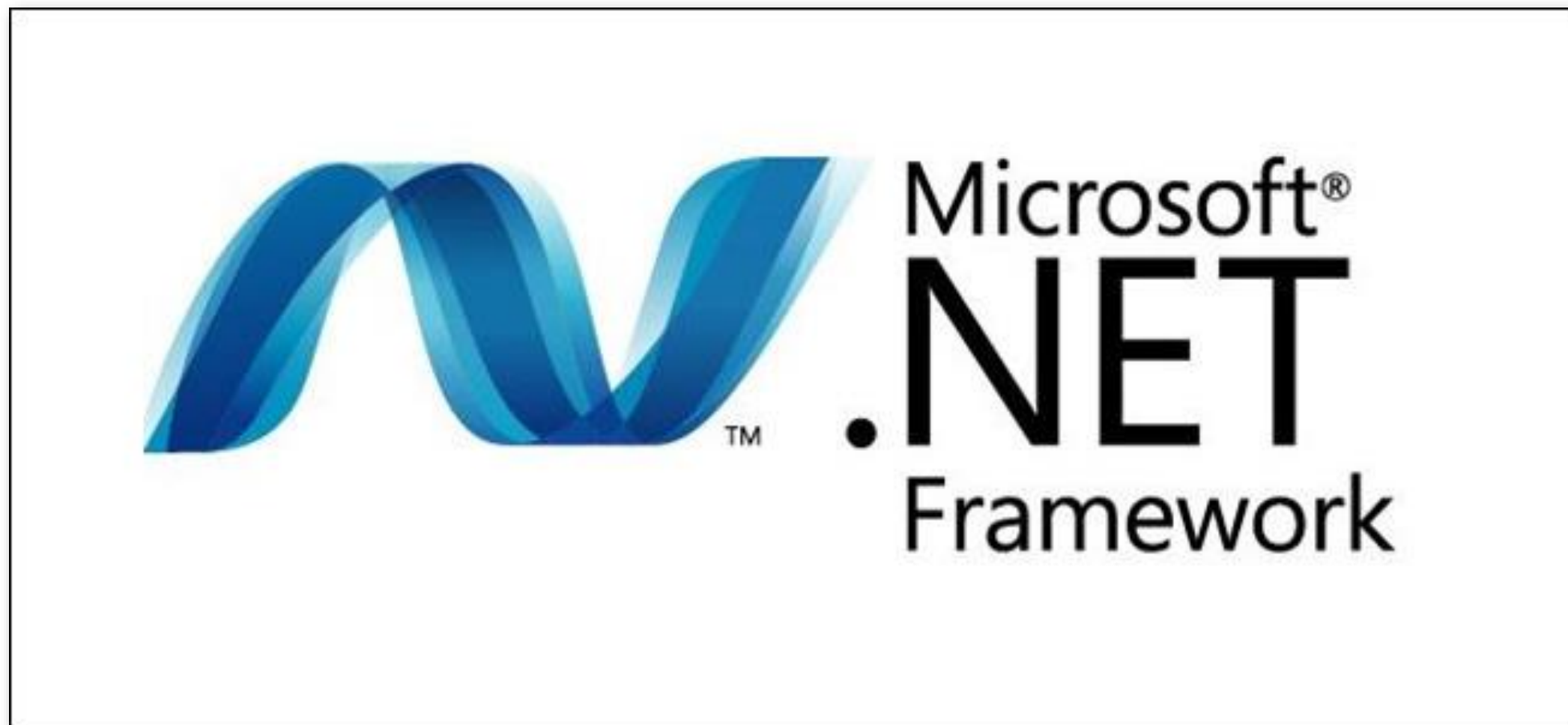
01

# DECOUVERTE DU .NET

Première approche du Framework .NET

# DECOUVERTE DU .NET

- .NET Framework est une technologie développée par Microsoft





# DECOUVERTE DU .NET

- Avant le .NET Framework
  - Les « MS Languages » n'étaient pas standardisés
  - Leurs syntaxes étaient différentes
  - De nombreuses difficultés de compatibilité
- Objectif du .NET Framework
  - Standardisation des langages et protocoles

# DECOUVERTE DU .NET

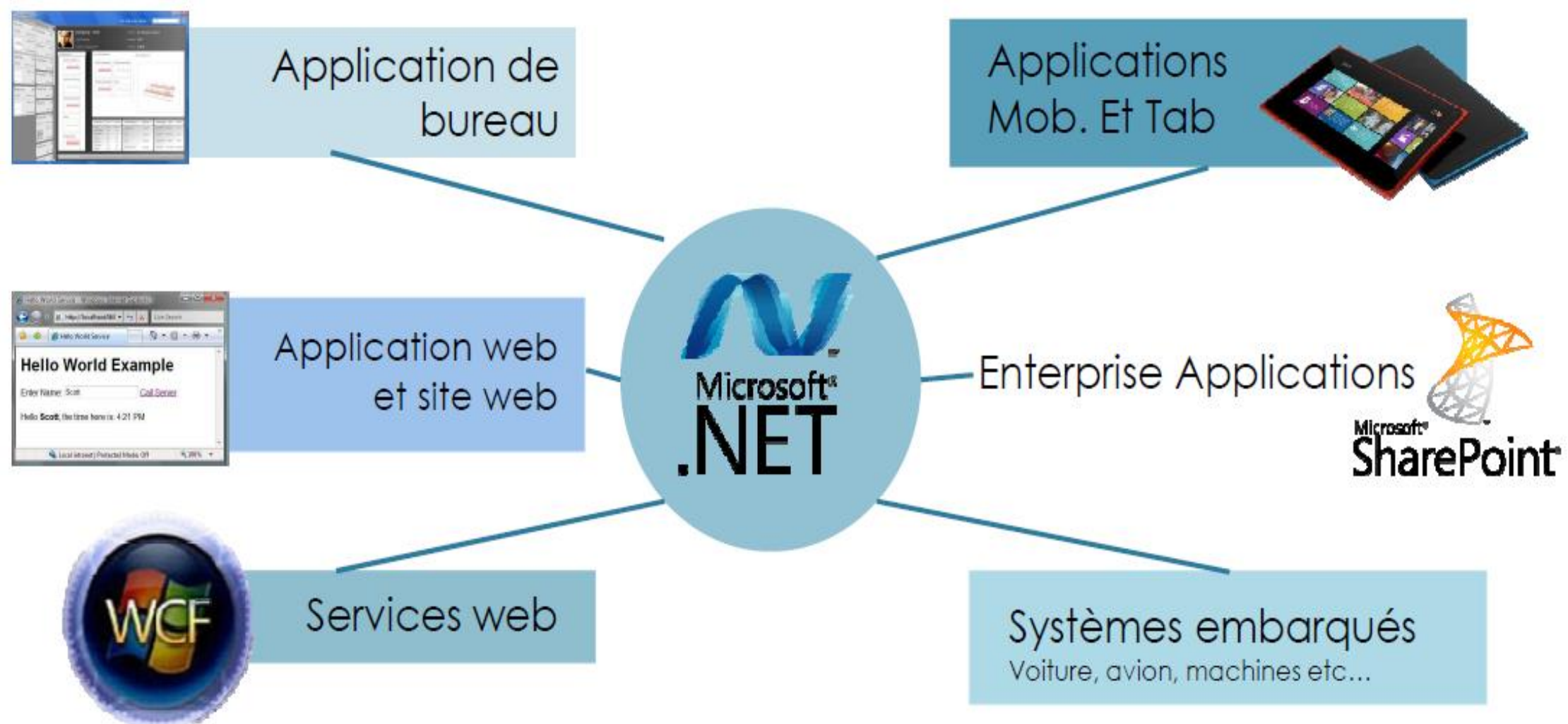
- **.NET Framework** est une technologie qui prend en charge la création et l'exécution d'applications et de services Web
- Il est conçu pour respecter les objectifs suivants :
  - Fournir un environnement de programmation cohérent et orienté objet, que le code objet soit stocké et exécuté localement, mais distribué sur le Web ou exécuté à distance
  - Rendre l'expérience du développeur cohérente entre les différents types d'applications, telles que les applications basées sur Windows et les applications Web.
  - Créer toutes les communications sur les normes du secteur afin d'assurer que le code basé sur **.NET Framework** s'intègre à tout autre code.

# DECOUVERTE DU .NET

- .NET fournit un environnement d'exécution de code qui :
  - Minimise le déploiement de logiciels et les conflits de contrôle de version.
  - Promeut l'exécution sécurisée du code, y compris le code créé par un tiers inconnu ou semi-approuvé.
  - Élimine les problèmes de performances des environnements scriptés ou interprétés.

# DECOUVERTE DU .NET

- .NET Plateforme pour simplifier le développement des applications web, Mais pas seulement...

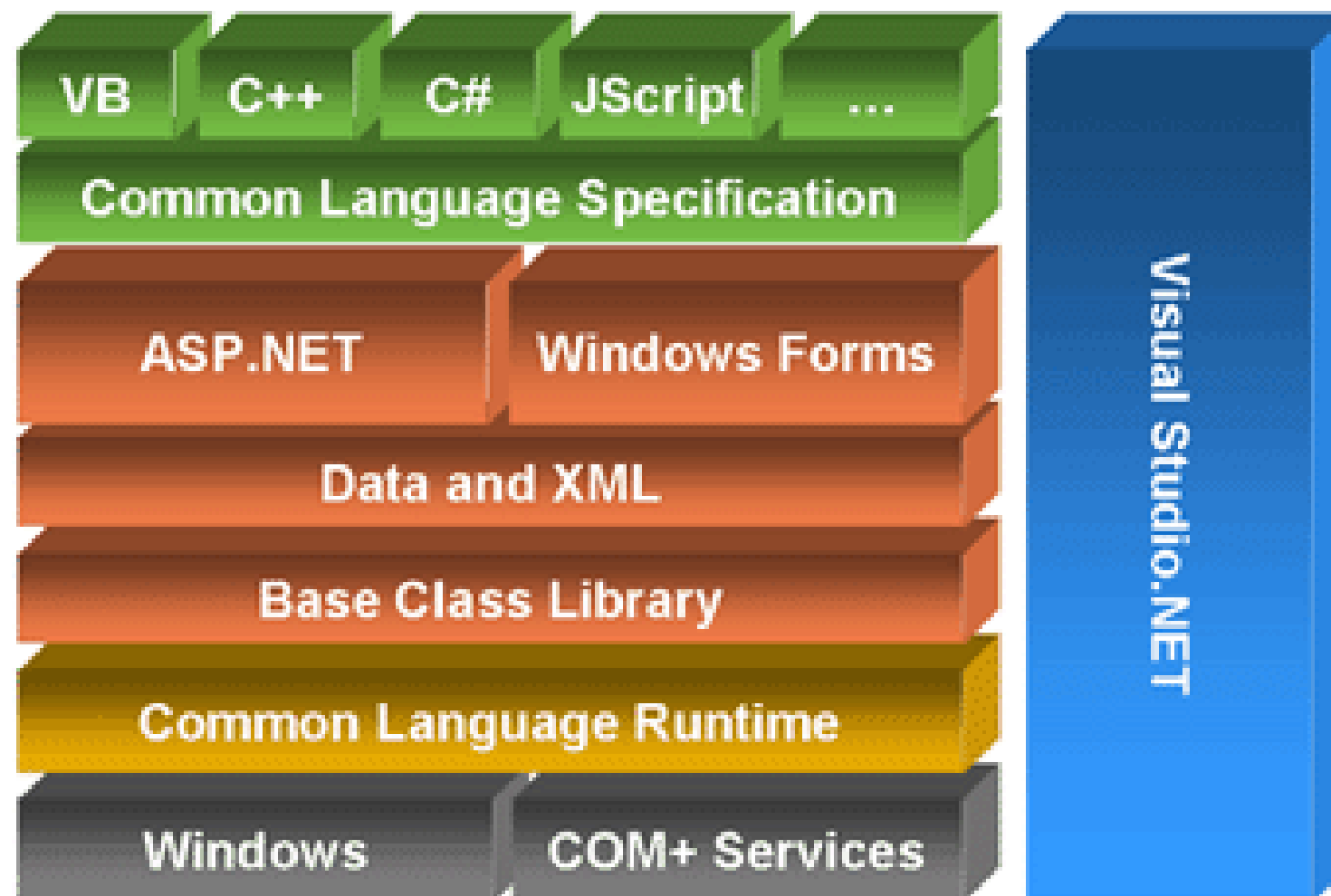




# DECOUVERTE DU .NET

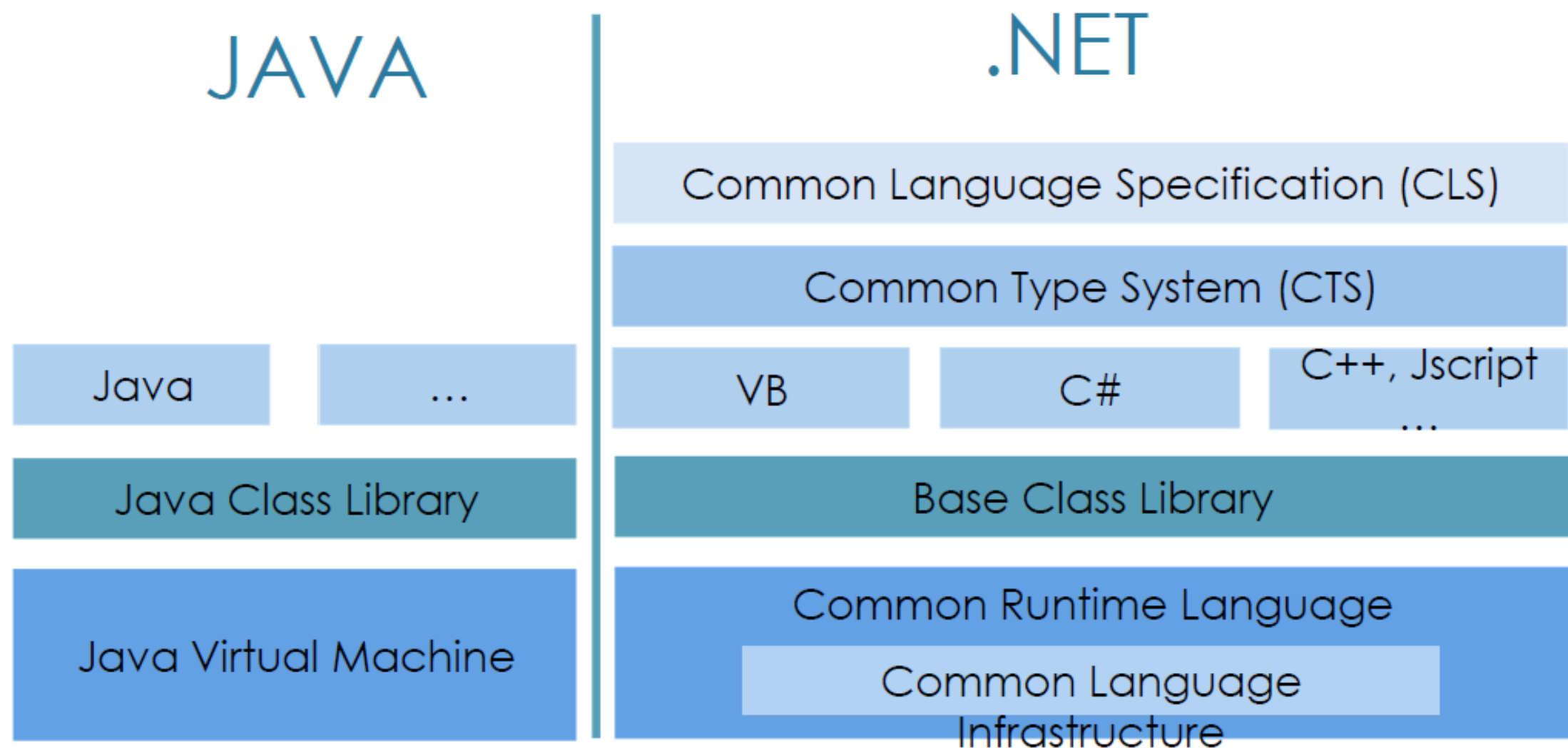
- .NET Framework pour standardiser le développement d'applications sur Windows... mais pas uniquement.

## .NET Framework Architecture



# DECOUVERTE DU .NET

- L'architecture du .NET, une réponse à Java





02

# QU'EST-CE QUE LA CLS ?

Définition de la « Common Language Specification »

# QU'EST-CE QUE LA CLS ?

- La Common Language Specification (CLS)
  - C'est une normalisation qui doit être respecté et implémenté par tout langage de programmation qui se veut capable de créer des applications .NET
  - Cette spécification s'intéresse à la normalisation d'un certain nombre de fonctionnalités comme par exemple la manière d'implémenter les types de données, les classes, les délégués, la gestion des événements, etc ...
  - Elle à pour objectif de garantir l'interopérabilité entre les langages



03

# QU'EST-CE QUE LE CTS ?

Première approche du Framework .NET

# QU'EST-CE QUE LE CTS ?

- Le Common Type System (CTS)
  - Le Système de Type Commun fournit un modèle orienté objet pour prendre en charge l'implémentation de différents langages sur une implémentation de .NET
  - Il définit un ensemble de règles que tous les langages doivent respecter pour utiliser les types
  - Il fournit une bibliothèque contenant les types primitifs de base utilisés dans le développement d'applications (ex: Boolean, Byte, Char...etc)



04

# LA BASE CLASS LIBRARY

Comprendre la BCL et son fonctionnement

# LA BASE CLASS LIBRARY

- La Base Class Library (BCL)
  - C'est une **API** de programmation multi langages qui permet d'unifier les développements
  - C' est une collection de types réutilisables qui s'intègrent parfaitement au **Common Language Runtime (CLR)**.
  - Elle est orientée objet et fournit des types à partir desquels votre propre code managé dérive des fonctionnalités.



# LA BASE CLASS LIBRARY

- La **BCL** une API de programmation multi langages ?
  - Langage classique de programmation
    - ✓ Une syntaxe propre au langage
    - ✓ Accompagné d'une bibliothèque de fonctions
  - Les langages pour programmer avec la **BCL .NET**
    - ✓ D'après Microsoft, elle supporte 27 langages
    - ✓ Les 3 langages les plus utilisés : VB, C#, C++
  - Chaque langage conforme au **CLS** peut instancier les classes de la BCL et utiliser les méthodes

# LA BASE CLASS LIBRARY

- La **BCL**, 3 couches de classes





# LA BASE CLASS LIBRARY

- Les classes de la **BCL** sont organisées sous forme d'espaces de noms hiérarchisés
  - Chaque espace de nom peut comporter :
    - ✓ Un ensemble de classes
    - ✓ Des sous-espaces de noms
  - L'accès à une classe se fait à l'aide son nom complet. Il se compose de :
    - ✓ La liste hiérarchique des espaces de noms
    - ✓ Auquel s'ajoute le nom de la classe en question
    - ✓ Ces noms sont reliés entre eux par des points

# LA BASE CLASS LIBRARY

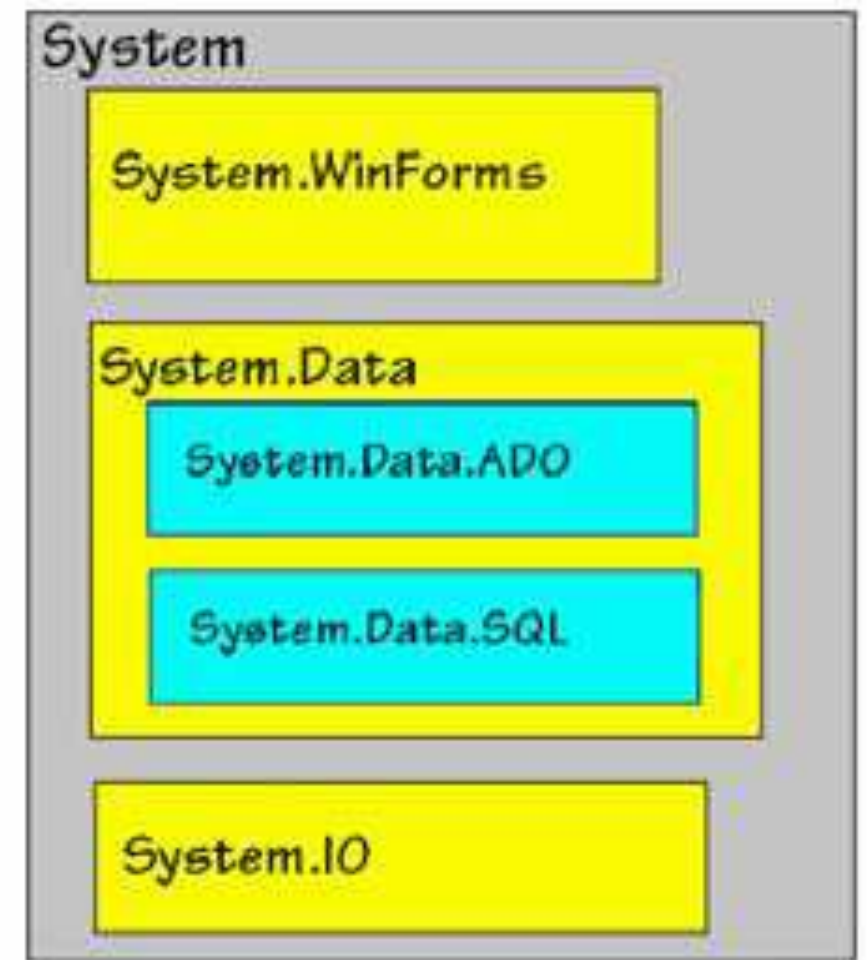
- Exemples d'accès à une classe de la BCL

- La classe « **DataSet** » faisant partie de l'espace de noms « **System.Data.Ado** » se déclare :

✓ **System.Data.Ado.DataSet**

- La classe « **Console** » faisant partie de l'espace de noms « **System** » se déclare :

✓ **System.Console**





05

# DÉFINITION DU CIL

Comprendre le « Common Intermediate Language »

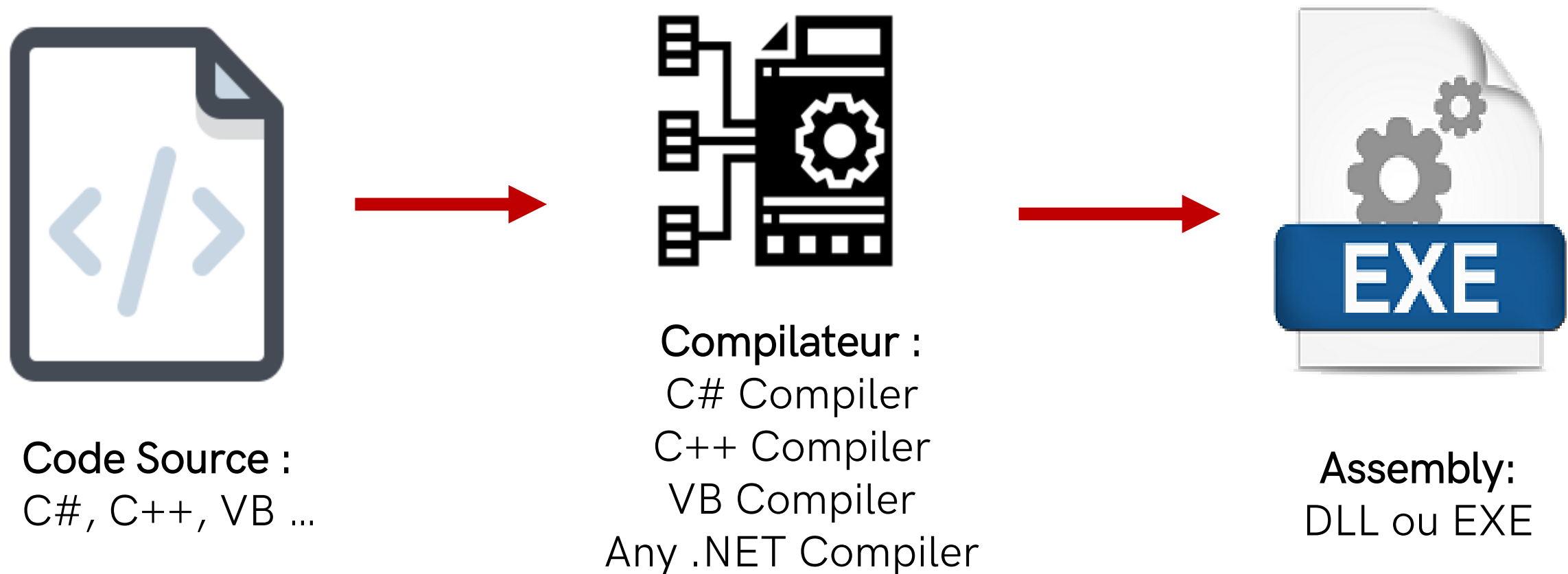
# DÉFINITION DU CIL

- Le Common Intermediate Language (CIL anciennement MSIL)
  - La compilation d'un programme écrit en .NET conduit vers la création d'un fichier exécutable (fichier .exe).
    - ✓ Il n'est pas écrit en code machine mais en CIL
    - ✓ L'exécution de fichier compilé en CIL ne peut pas être assuré par les service du système d'exploitation



# DÉFINITION DU CIL

- Le Common Intermediate Language (CIL anciennement MSIL)



06

# ÉTAPE D'ASSEMBLAGE

Comprendre l'assembly, sa composition et  
métadonnées

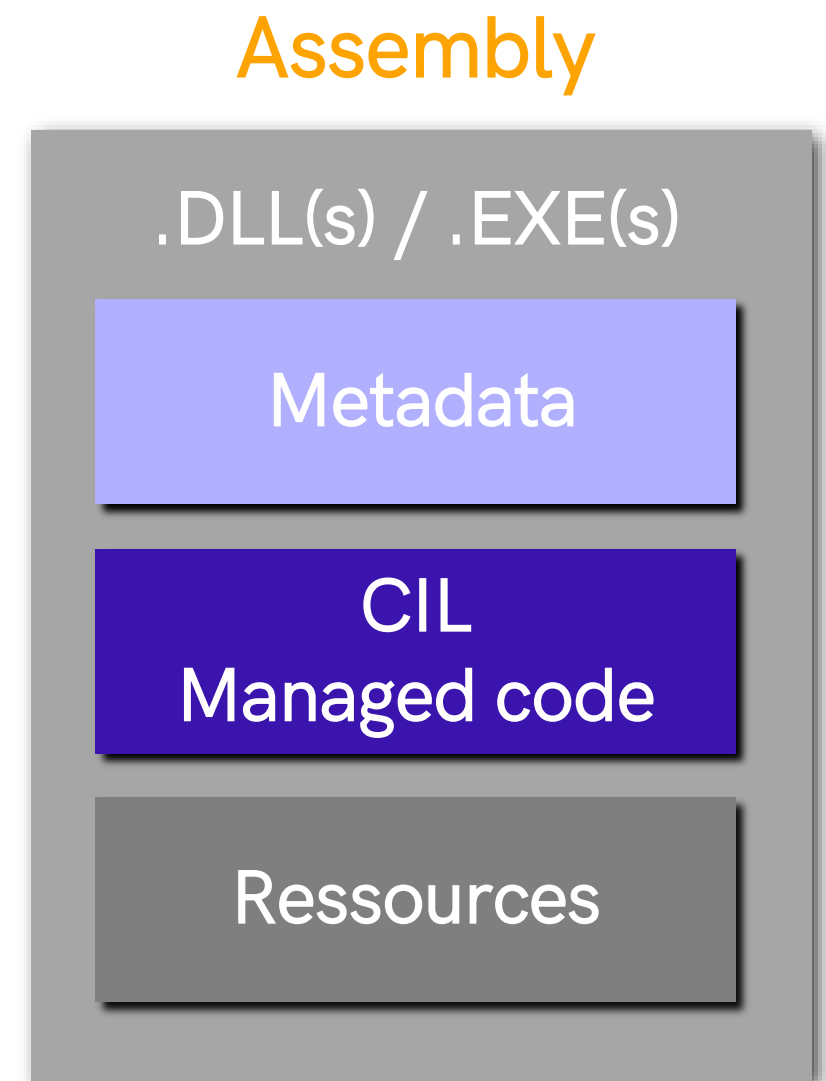


# ÉTAPE D'ASSEMBLAGE

- L'assemblage (**Assembly**) est le fichier **exe** ou **dll** produit par la compilation d'un code **.NET**
  - C'est lui qui est exécuté par la **CLR**
- Un **Assembly** contient trois types de données :
  - Le code **CIL** qui résulte de la compilation.
  - Les méta données.
  - Les ressources.

# ÉTAPE D'ASSEMBLAGE

- Le CIL dans un Assembly
  - Peut comporter plusieurs classes
  - Ne peut avoir qu'un seul point d'entrée (une seule classe Main)
- Un Assembly peut contenir
  - Des ressources utilisées par le CIL : icônes, bitmap...





# ÉTAPE D'ASSEMBLAGE

Les **méta données** dans un **Assembly**

- Les données qui accompagnent le code **CIL** dans un **Assembly** sont de deux catégories
  - Des données de **description des types**
  - L'**Assembly** manifest

## Metadata

### Type Descriptions

Classes  
Base classes  
Implemented interfaces  
Attributes  
Methods

### Assembly Manifest

Name  
Version  
Culture

Other assemblies  
Security permissions  
Exported types

07

# L'ENVIRONNEMENT CLR

Utilisation de la « Common Language Runtime » en .NET

# L'ENVIRONNEMENT CLR

- La Common Language Runtime est un environnement qui assure l'exécution des programmes .NET
  - Elle joue le rôle de la machine virtuelle de Java mais pour les programmes écrits en .NET
  - Elle s'adapte aux ressources du système d'exploitation sur lequel elle est installée
- La CLR interprète les fichiers exécutables compilés en CIL
  - Elle permet d'exécuter le même Assembly sur différents système d'exploitation

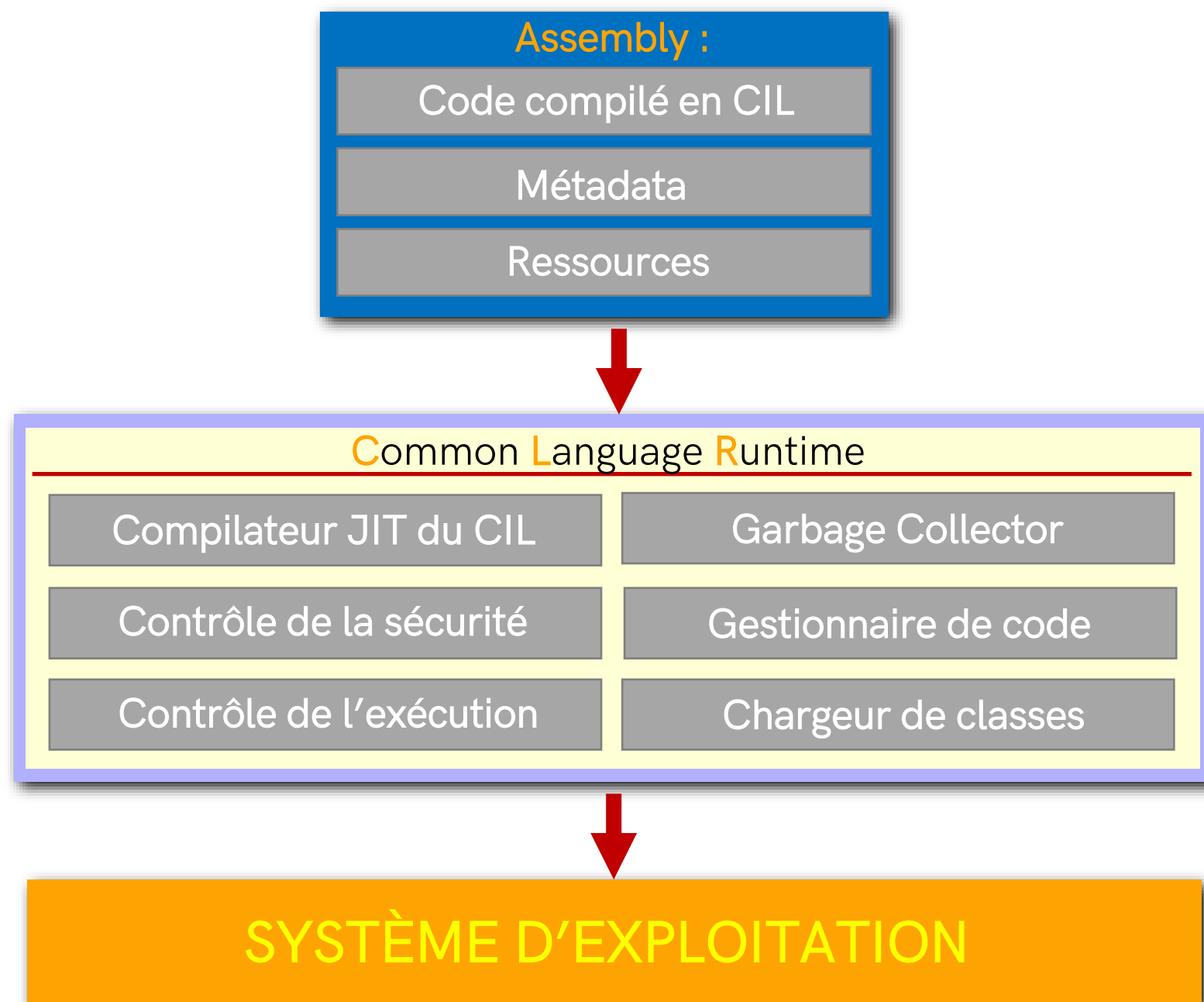


# L'ENVIRONNEMENT CLR

- La Common Language Runtime fournit des services tels que :
  - La gestion de la mémoire (avec le Garbage Collector)
  - La gestion des exceptions
  - La gestion des threads
  - L'interopérabilité entre plusieurs langages
  - Le chargement dynamique des modules à exécuter
  - La compilation du CIL en code machine natif (propre à l'OS sur lequel le programme s'exécute)
  - Le contrôle de l'exécution des programmes

# L'ENVIRONNEMENT CLR

- L'exécution d'un **Assembly** avec la **Common Language Runtime**



# L'ENVIRONNEMENT CLR

- Code managé... Code natif... Quelle différence ?
  - Un programme managé (Managed program) est un programme compilé en **CIL**, son exécution est gérée par la **CLR**
  - Un programme **non managé** est un programme compilé en **code natif**. Son **exécution** est **directement prise en charge** par les services du **système d'exploitation**
  - **VB.NET** et **C#.NET** ne permettent de créer que des programmes managés. **C++.NET** permet de créer des programmes **managés** et **non managés**

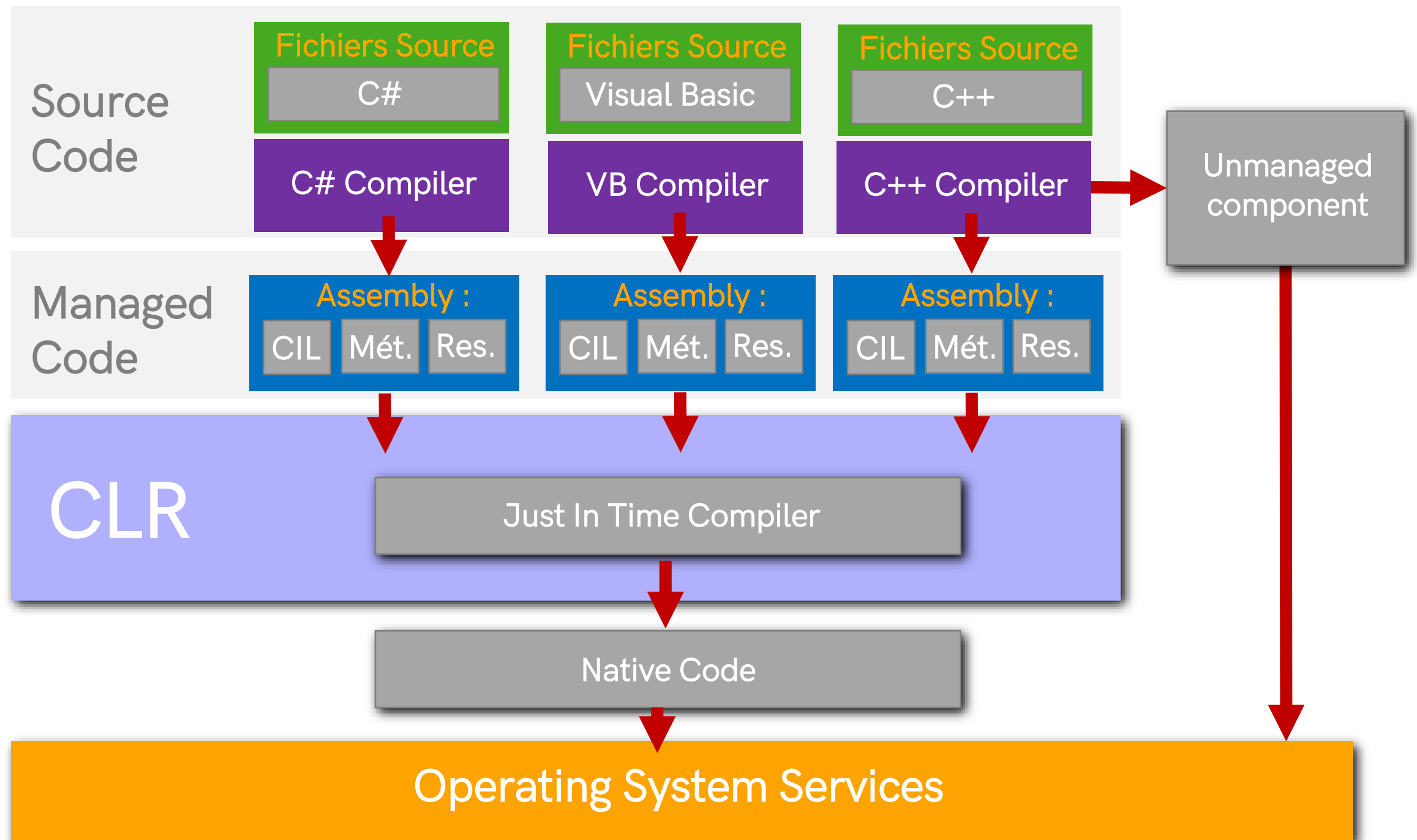


# L'ENVIRONNEMENT CLR

- Les avantages de la Common Language Runtime
  - Sécurité de l'exécution des programmes
    - ✓ Grâce à la gestion des exceptions et à la gestion automatique de la mémoire.
  - Interopérabilité de programmes écrits dans différents langages
    - ✓ Tous les langages qui supportent le .NET compilent vers un même code intermédiaire (CIL)
    - ✓ Possibilité de faire communiquer des programmes écrits dans des langages différents

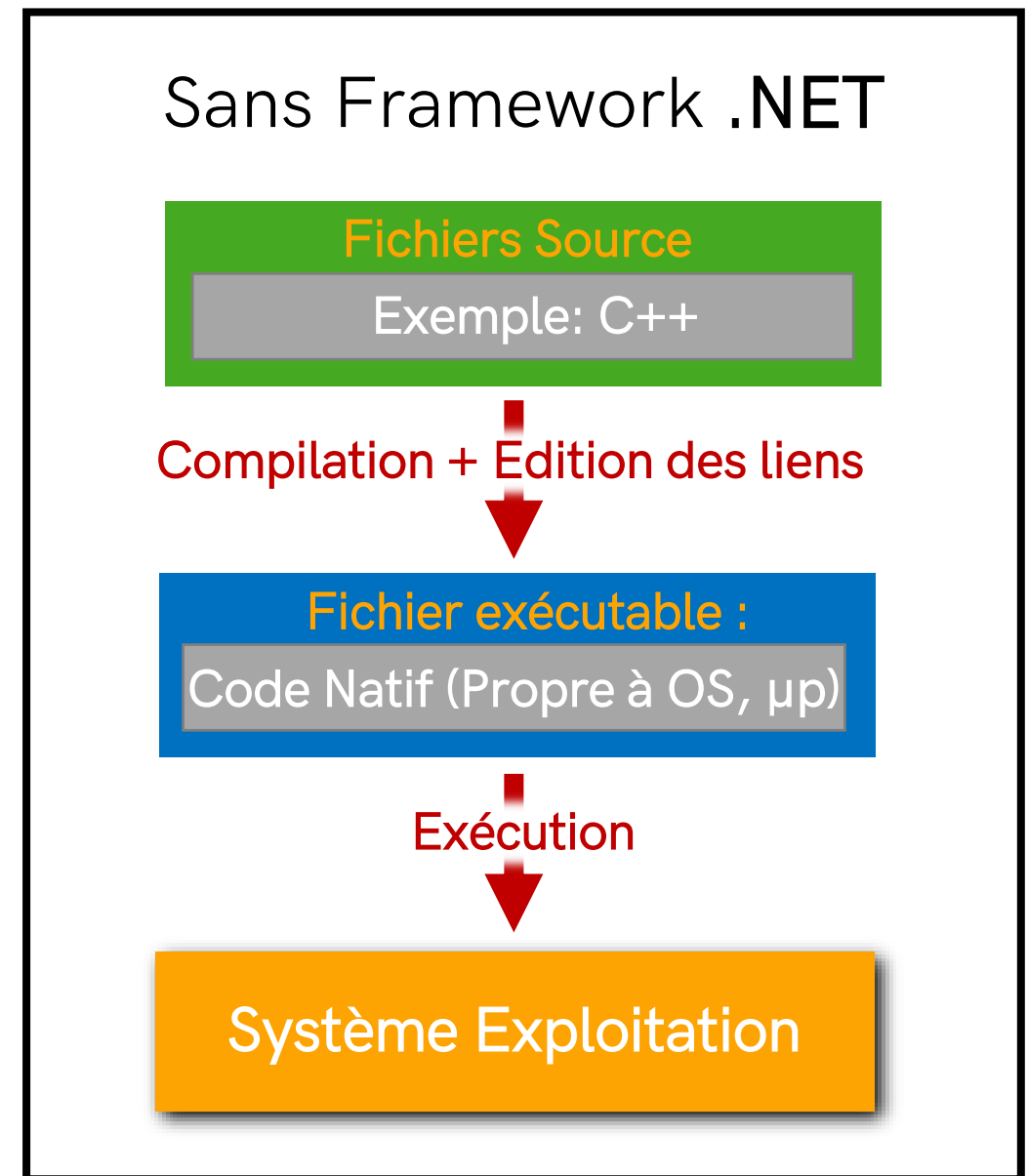
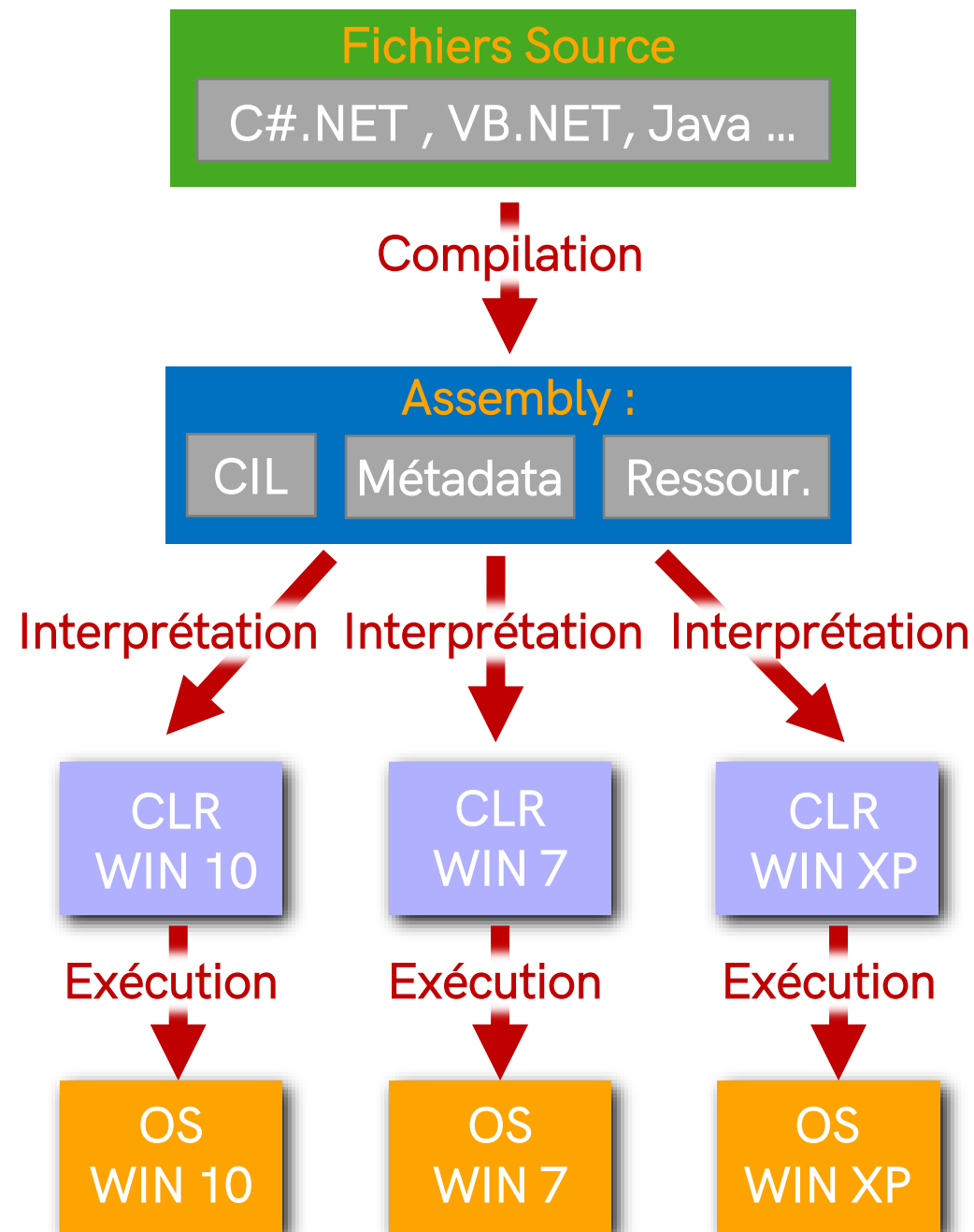
# L'ENVIRONNEMENT CLR

- Interopérabilité de programmes écrits dans différents langages



# L'ENVIRONNEMENT CLR

- Portabilité d'un programme avec la Common Language Runtime





# L'ENVIRONNEMENT CLR

- Comparaison entre .NET et Java

.NET	JAVA
Code source	Code source
Common Intermediate Language	Bytecode
Common Language Runtime	Machine Virtuelle (JVM)

08

# LES SPÉCIFICATIONS .NET

Comprendre les spécifications .NET et .NET Core

# LES SPÉCIFICATIONS .NET

- .NET Framework en tant que spécification est une spécification publique (proposée par MS)
- Il existe deux principales implémentations de cette spécification





# LES SPÉCIFICATIONS .NET

- Toutefois une dernière implémentation viens de voir le jour
  - Appelée initialement « Mono », c'est la spécification Xamarin



# LES SPÉCIFICATIONS .NET

- La spécification .NET Framework est aussi appelée le « .NET Standard ».
  - On y retrouve les App Models suivante
    - ✓ Windows Présentation Foundation (WPF)
    - ✓ Windows Forms
    - ✓ Console
    - ✓ ASP.NET
  - La Base Class Library du .NET Standard
    - ✓ .NET Framework BCL

# LES SPÉCIFICATIONS .NET

- La spécification .NET Core
  - On y retrouve les App Models suivante
    - ✓ Console
    - ✓ ASP.NET Core
  - La Base Class Library du .NET Core
    - ✓ .NET Core BCL

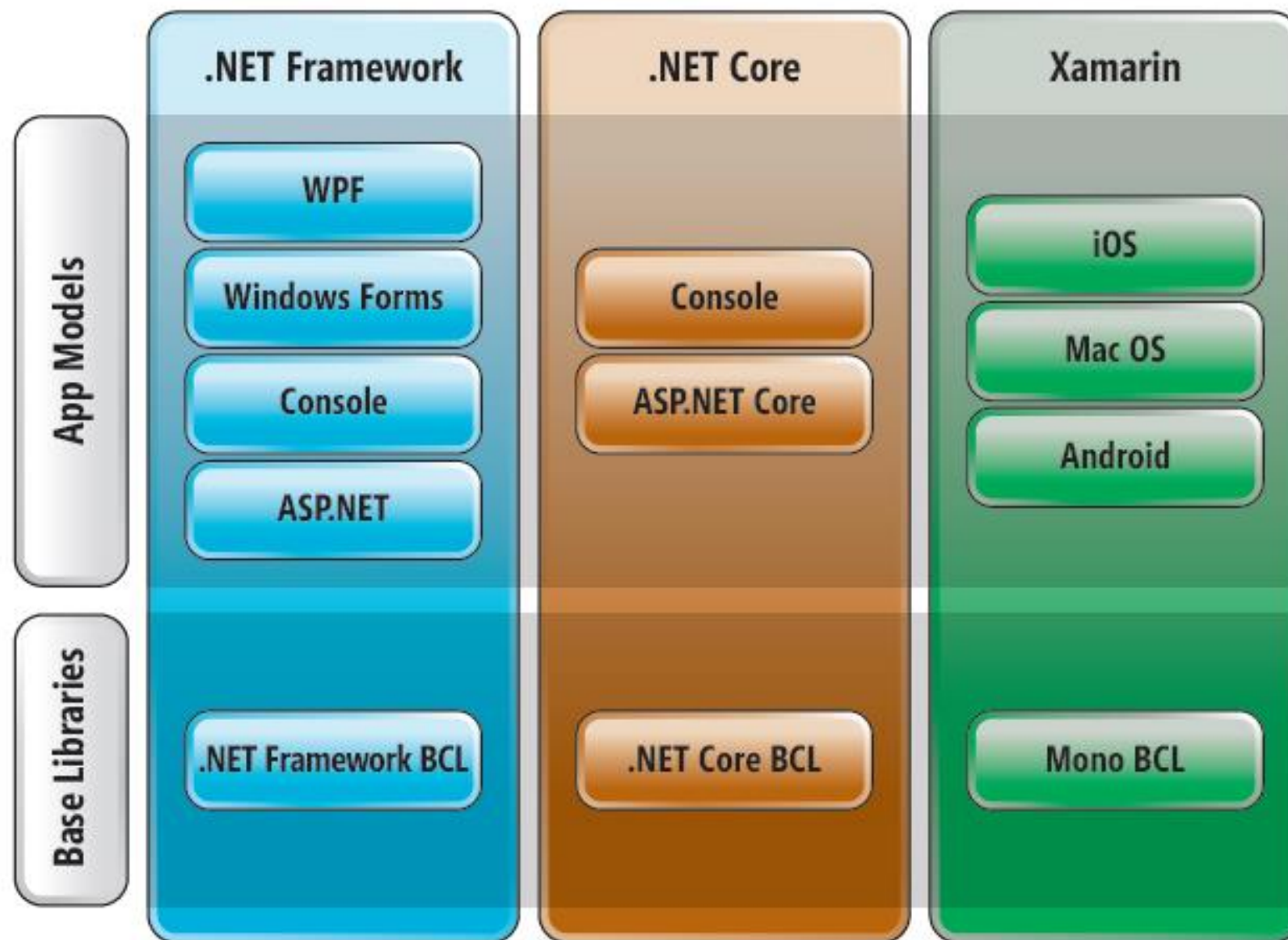


# LES SPÉCIFICATIONS .NET

- La spécification **Xamarin**
  - On y retrouve les App Models suivante
    - ✓ iOS
    - ✓ Mac OS
    - ✓ Android
  - La **B**ase **C**lass **L**ibrary de **Xamarin**
    - ✓ Mono BCL

# LES SPÉCIFICATIONS .NET

- Synthèse des spécifications .NET Standard, .NET Core et Xamarin





# MERCI DE VOTRE ATTENTION

Des questions...?