Introduction Framework .Net

m2iformation.fr

UTOPIOS
TRANSFORMATION DIGITALE INTELLIGENTE

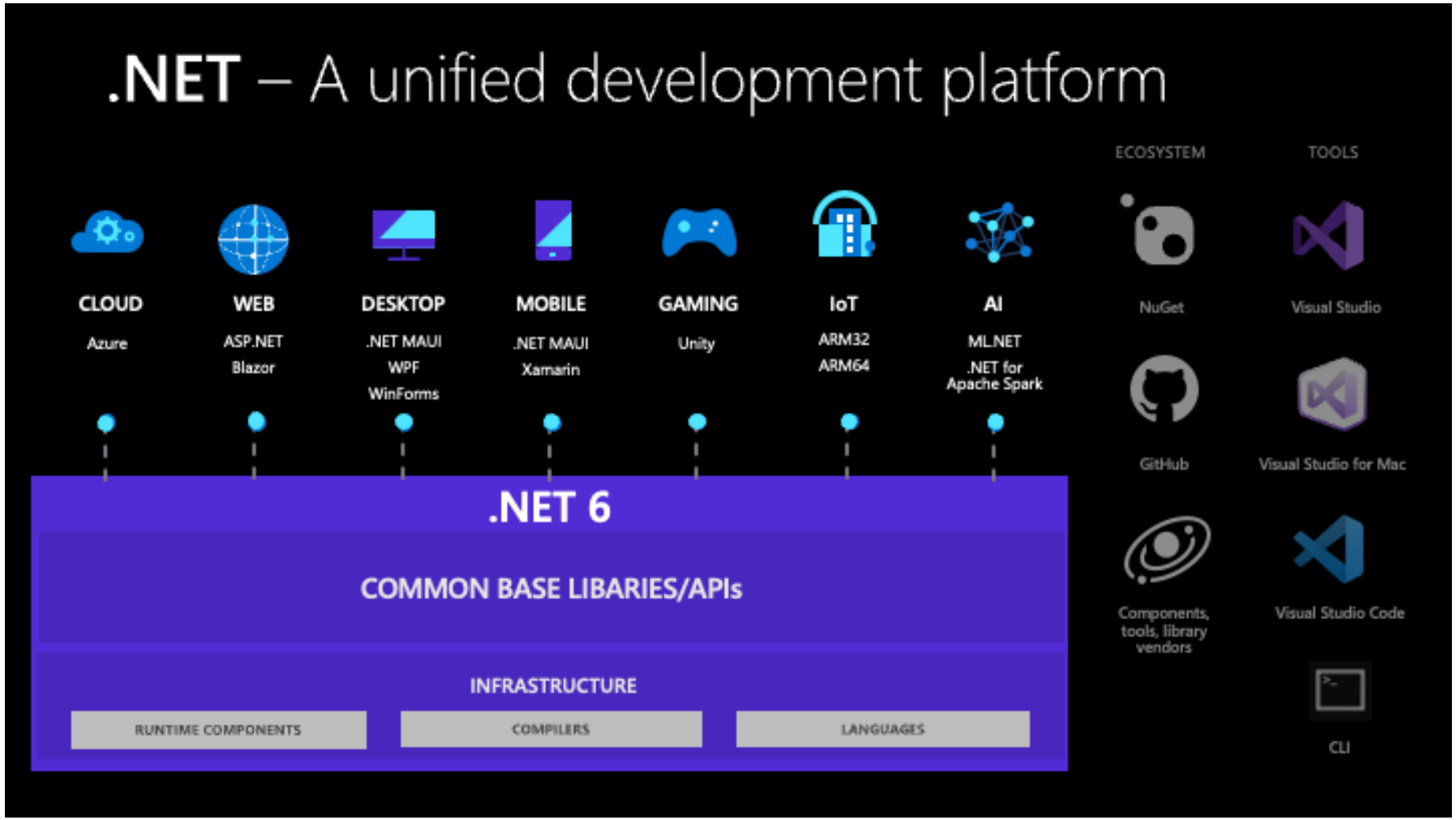
Introduction Framework .Net

Sommaire

UTOPIOS
TRANSFORMATION DIGITALE INTELLIGENTE

01 DECOUVERTE du .Net

.NET Framework est une technologie développée par Microsoft



- 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

- **.NET Framework** est une technologie qui prend en charge la création et l'exécution d'**applications** et de **services Web**
- Il fournit un **environnement de programmation orienté objet**
- Il rend l'expérience du développeur cohérente entre les différents types d'applications, telles que les **applications Windows**, les **applications Web** ou **mobiles**
- Il crée toutes les communications sur les normes du secteur afin d'assurer que le code basé sur .NET Framework s'intègre à tout autre code.

.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.

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

Application de bureau

Applications Mob. Et Tab

Application web et site web

Microsoft .NET

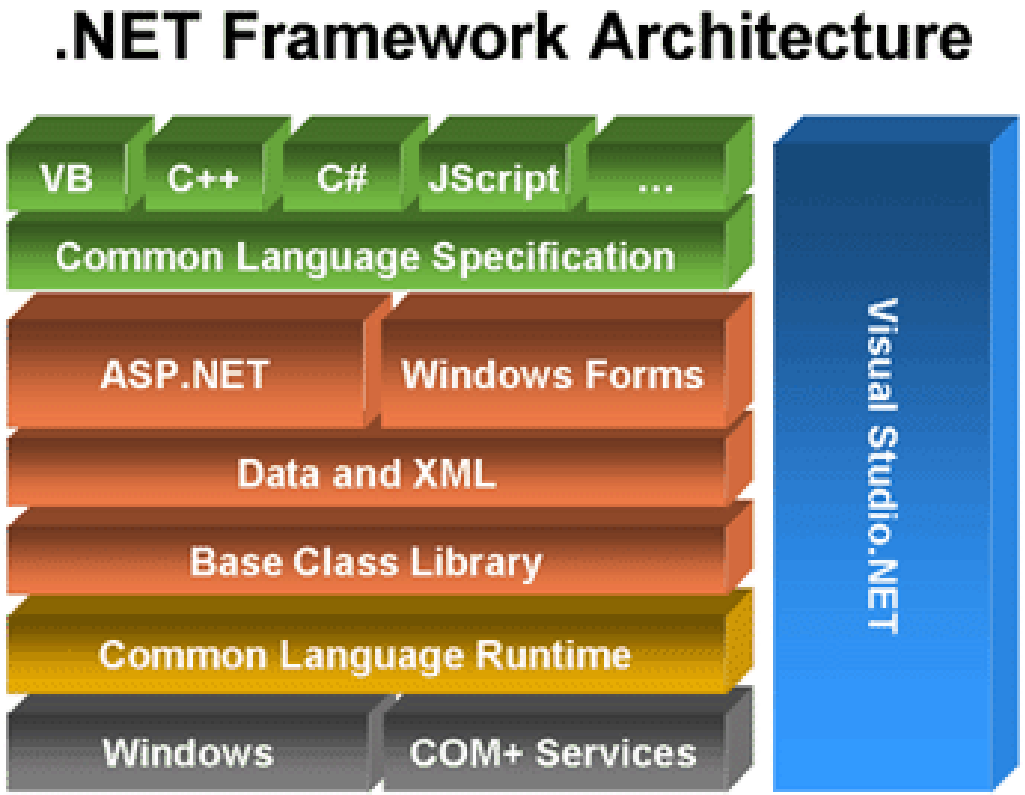
Enterprise Applications

Microsoft SharePoint

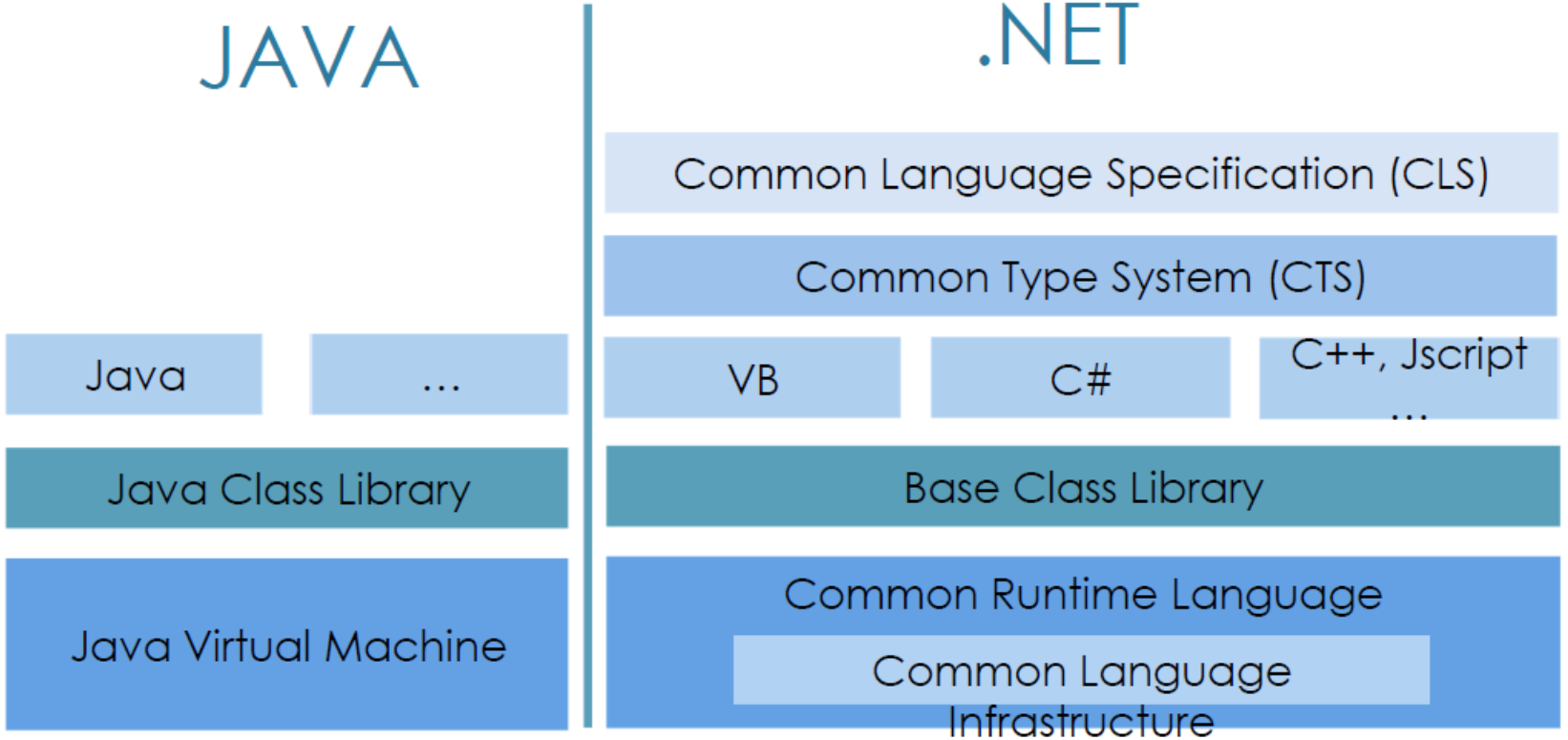
Services web

Systèmes embarqués
Voiture, avion, machines etc…

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



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



02 Qu’est ce que la CLS ?

La **C**ommon **L**anguage **S**pecification (**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 la CTS ?

Le Common Type System (CTS)

- Le Système de Type Commun fourni 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)

UTOPIOS
TRANSFORMATION DIGITALE INTELLIGENTE

04 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 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 : C#, VB, C++
- Chaque langage conforme au CLS peut instancier les classes de la BCL et utiliser les méthodes

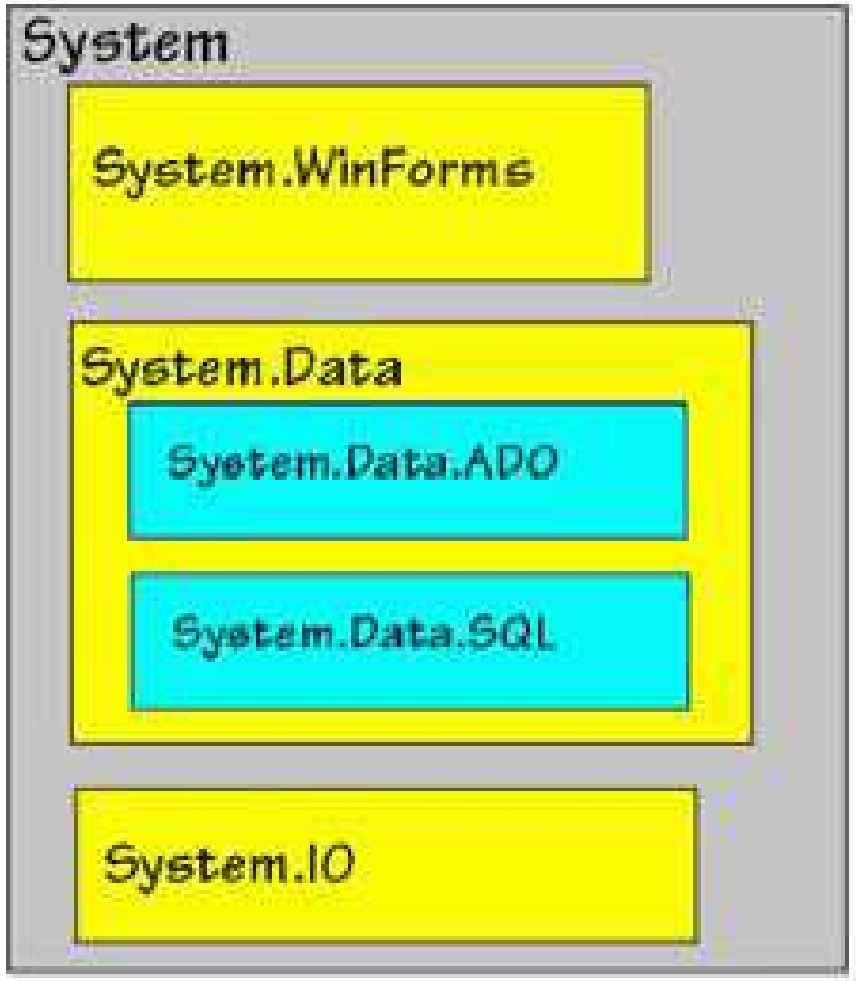
La BCL, 3 couches de classes

Les classes de la BCL sont organisée 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

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



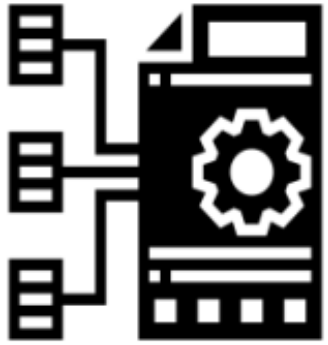
05 Définition du CIL

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 ou .dll).
  - 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

Le Common Intermediate Language (CIL anciennement MSIL)

Code Source :
C#, C++, VB ...



Compilateur :
C# Compiler
C++ Compiler
VB Compiler
Any .NET Compiler

Assembly:
DLL ou EXE

06 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

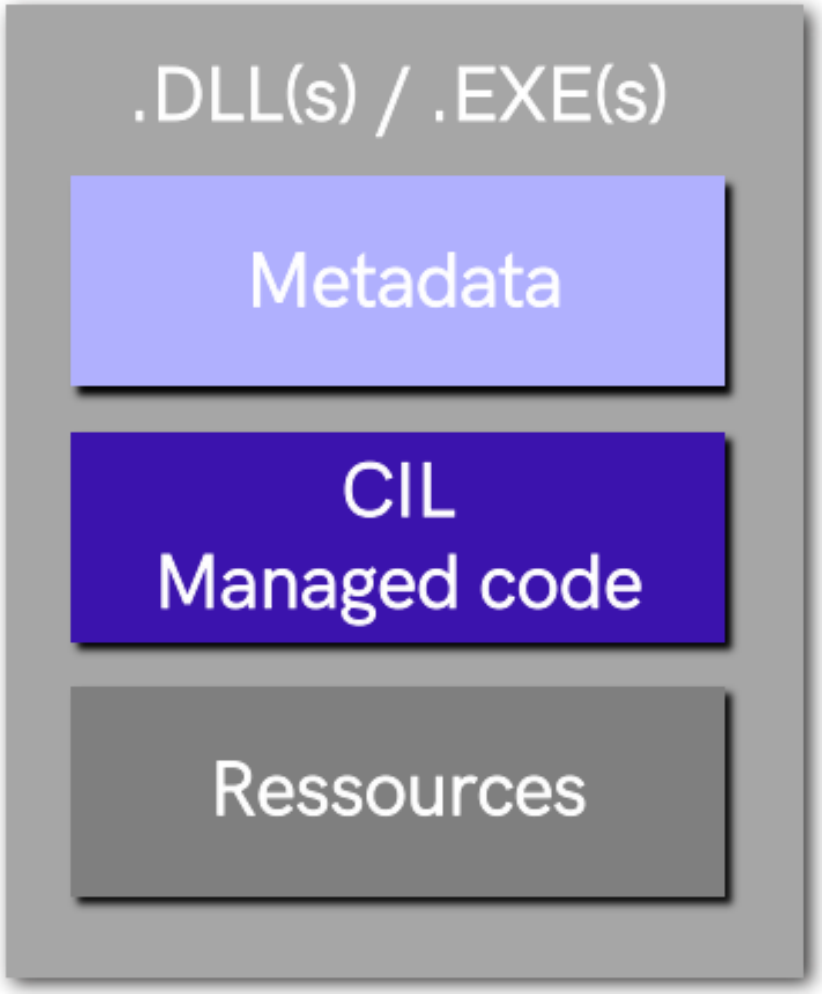
Un **Assembly** contient trois types de données :

- Le code **CIL** qui résulte de la compilation.
- Les méta données.
- Les ressources.

Le CIL dans un Assembly

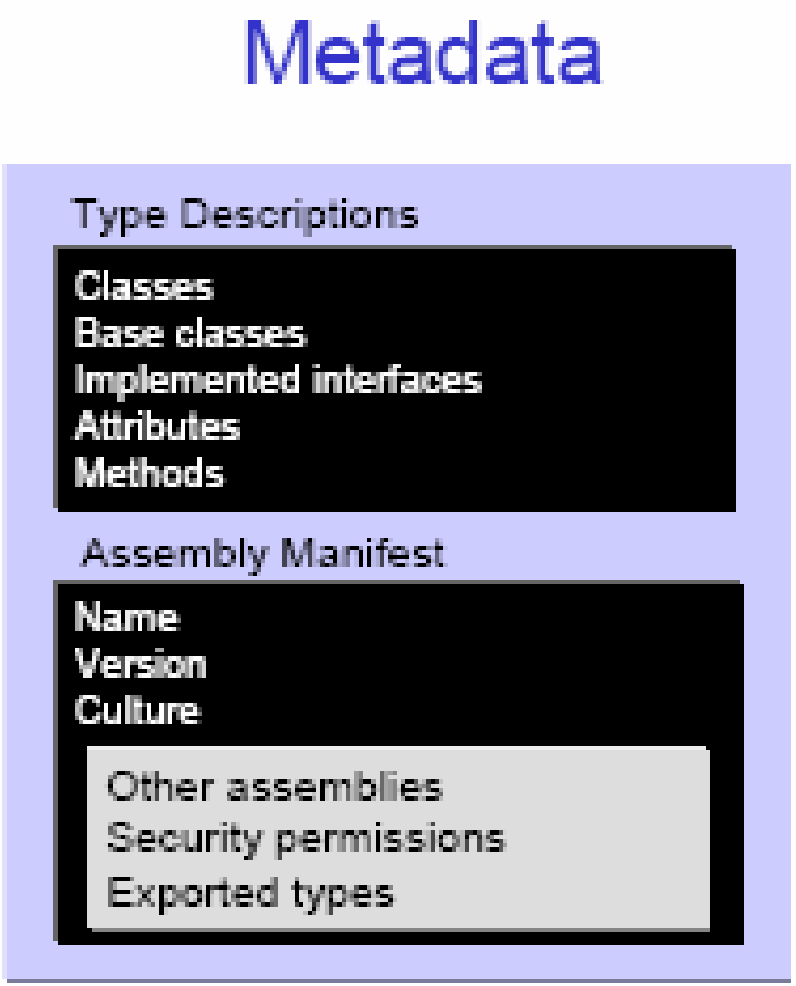
- Peut comporter plusieurs classes
- Ne peut avoir qu'un seul point d'entrée (une seule classe Main)

Un **Assembly** peut contenir des ressources utilisées par le **CIL** : icônes, bitmap…



Les métadonnées dans un Assembly

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



UTOPIOS
TRANSFORMATION DIGITALE INTELLIGENTE

07 L’environnement CLR

Le **C**ommon **L**anguage **R**untime est un environnement qui assure l'exécution des programmes **.NET**

- Il joue le rôle de la machine virtuelle de Java mais pour les programmes écrits en **.NET**
- Il s'adapte aux ressources du système d'exploitation sur lequel elle est installée

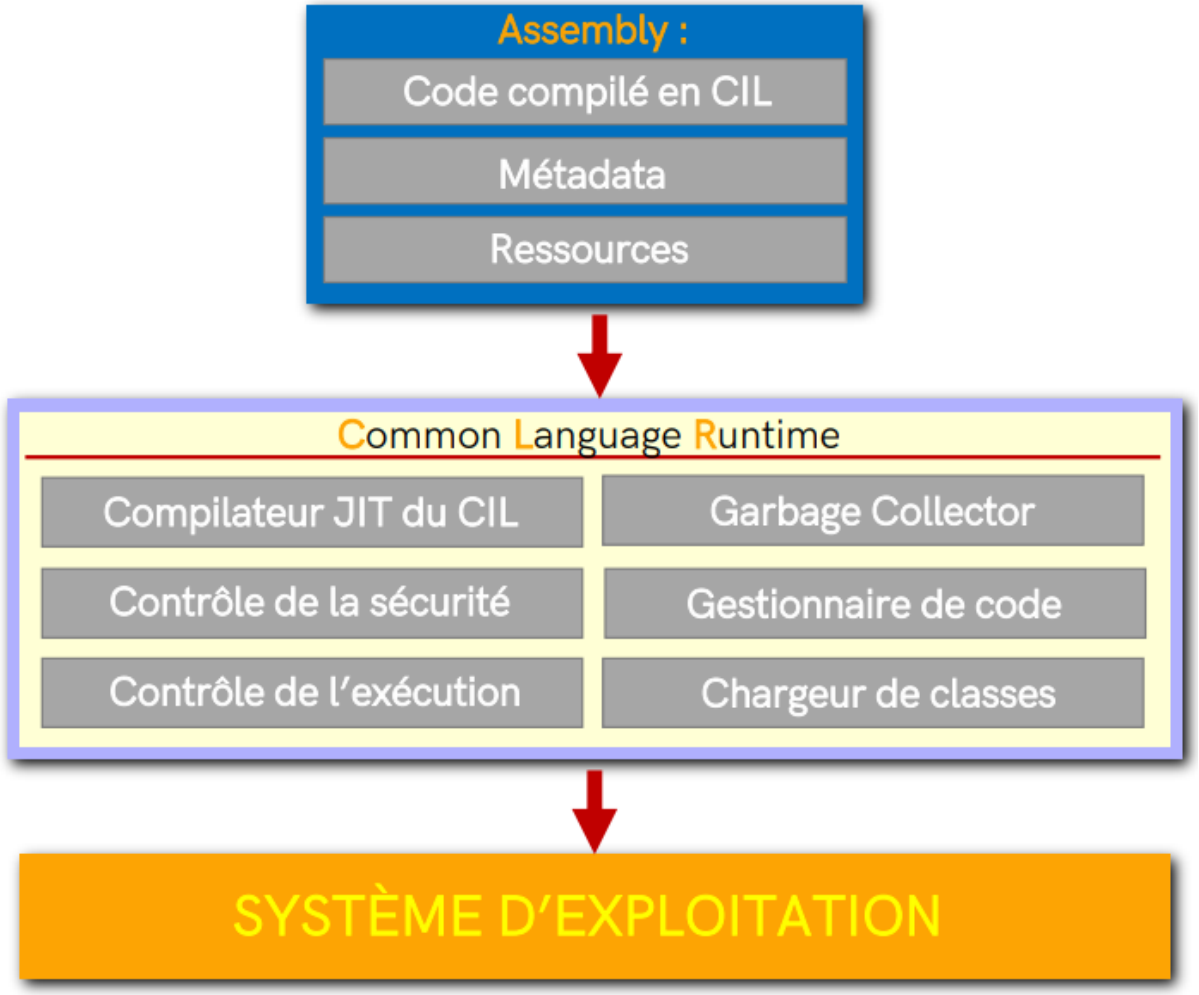
Le **CLR** interprète les fichiers exécutables compilés en **CIL**

- Il permet d'exécuter le même **Assembly** sur différents système d'exploitation

Le **C**ommon **L**anguage **R**untime 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'exécution d'un **Assembly** avec le **C**ommon **L**anguage **R**untime



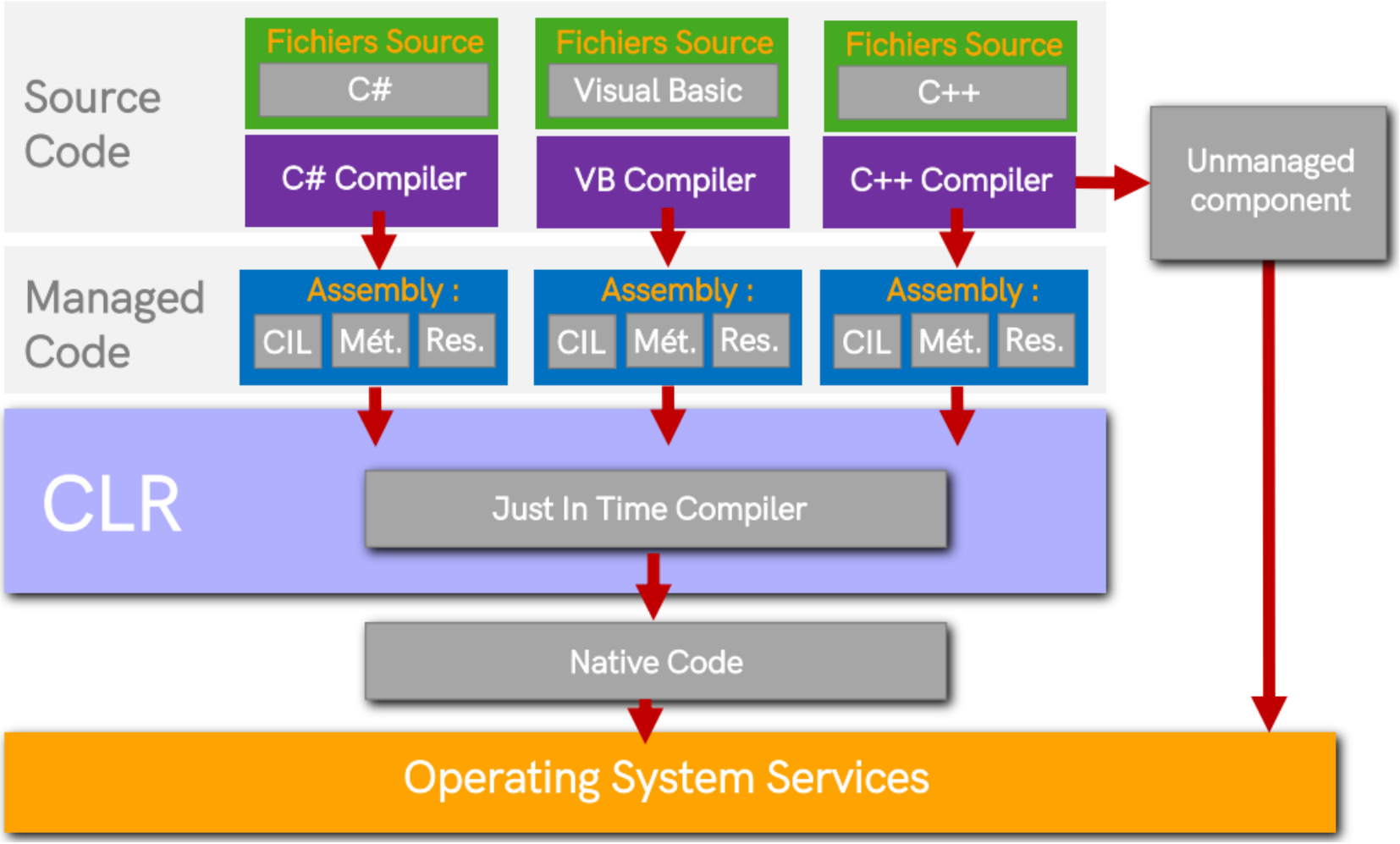
Code managé… Code natif… Quelle différence ?

- 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**

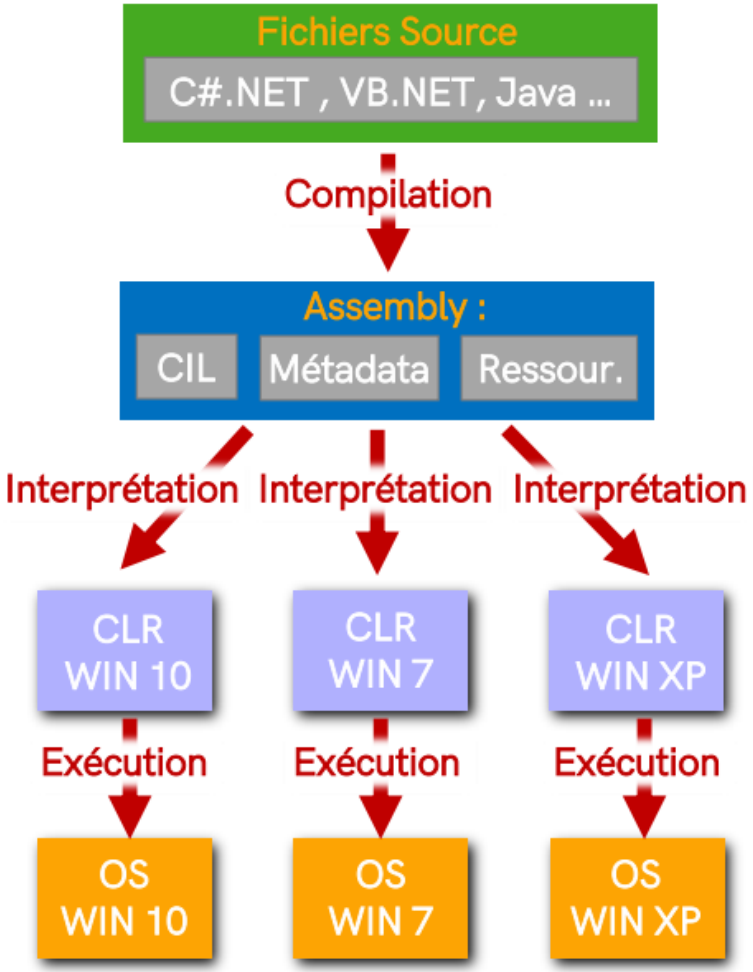
Les avantages du Common Language Runtime

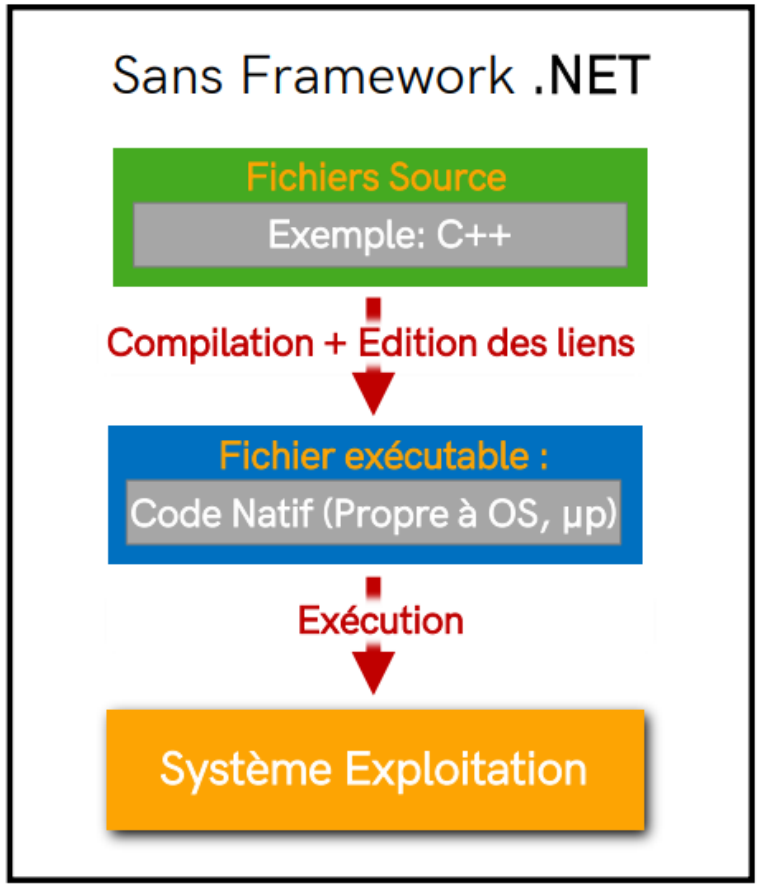
- Sécurité de l'exécution des programmes
  - Grâce à la gestion des exceptions et à la gestion automatique de la mémoire.
- Interopérabilité de programmes écrits dans différents langages
  - Tous les langages qui supportent le **.NET** compilent vers un même code intermédiaire (**CIL**)
  - Possibilité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

Interopérabilité de programmes écrits dans différents langages



Portabilité d'un programme avec la **C**ommon **L**anguage **R**untime





Comparaison entre .NET et Java

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

UTOPIOS
TRANSFORMATION DIGITALE INTELLIGENTE

08 Les Spécifications .NET

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

Toutefois une dernière implémentation a vu le jour :

- Appelée initialement « Mono », c'est la spécification Xamarin

La spécification .NET Framework (aussi appelée le «.NET Standard»)

- On y retrouve les App Models suivante
  - Console
  - **W**indows **P**resentation **F**oundation (**WPF**)
  - Windows Forms
  - ASP.NET
- La Base Class Library du .NET Standard
  - .NET Framework BCL

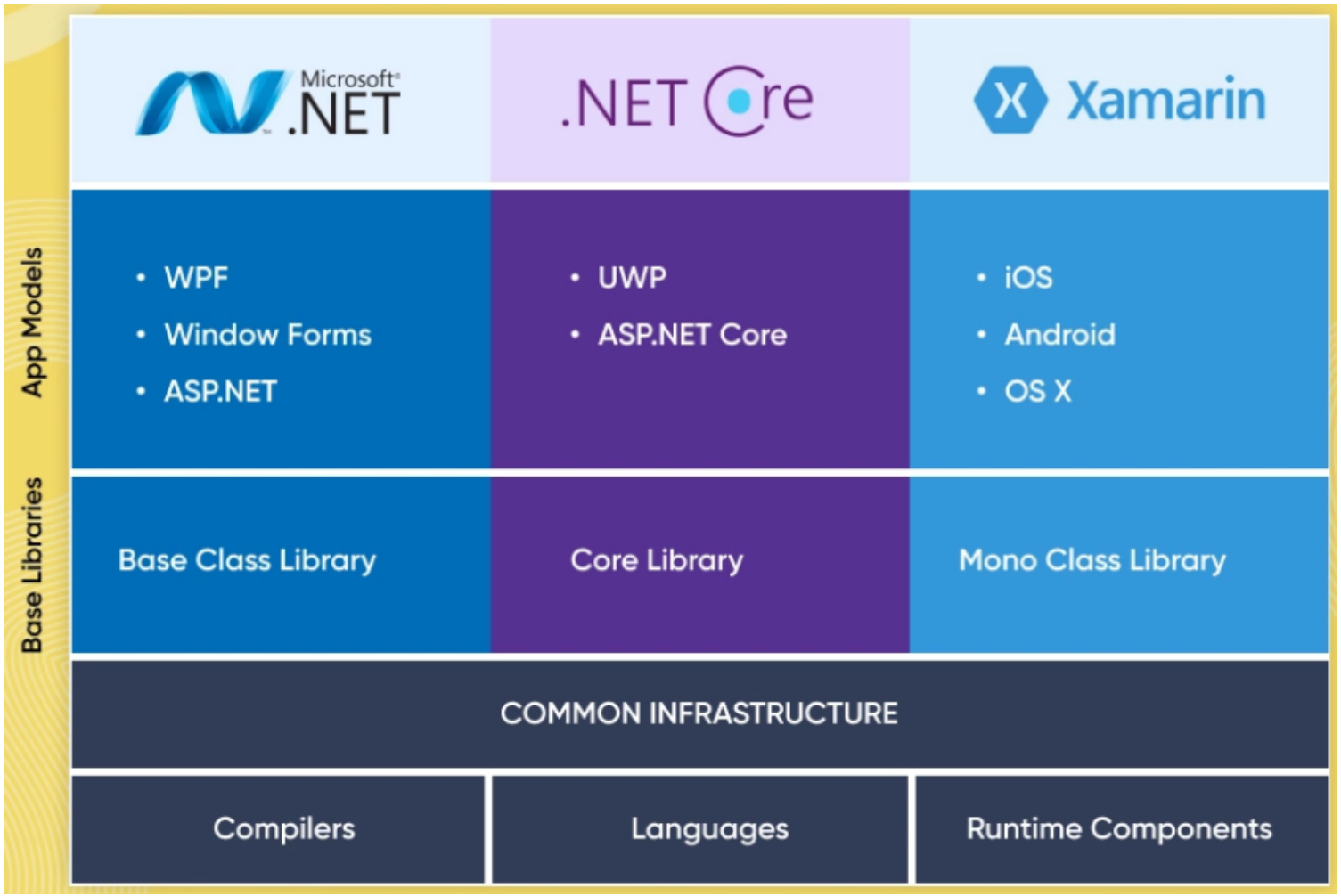
La spécification .NET Core

- On y retrouve les App Models suivante
  - Console
  - ASP.NET Core
  - Universal Windows Platform (UWP)
- La Base Class Library du .NET Core
  - .NET Core BCL

La spécification Xamarin

- On y retrouve les App Models suivante
  - iOS
  - Android
  - Mac OS
- La Base Class Library de Xamarin
  - Mono BCL

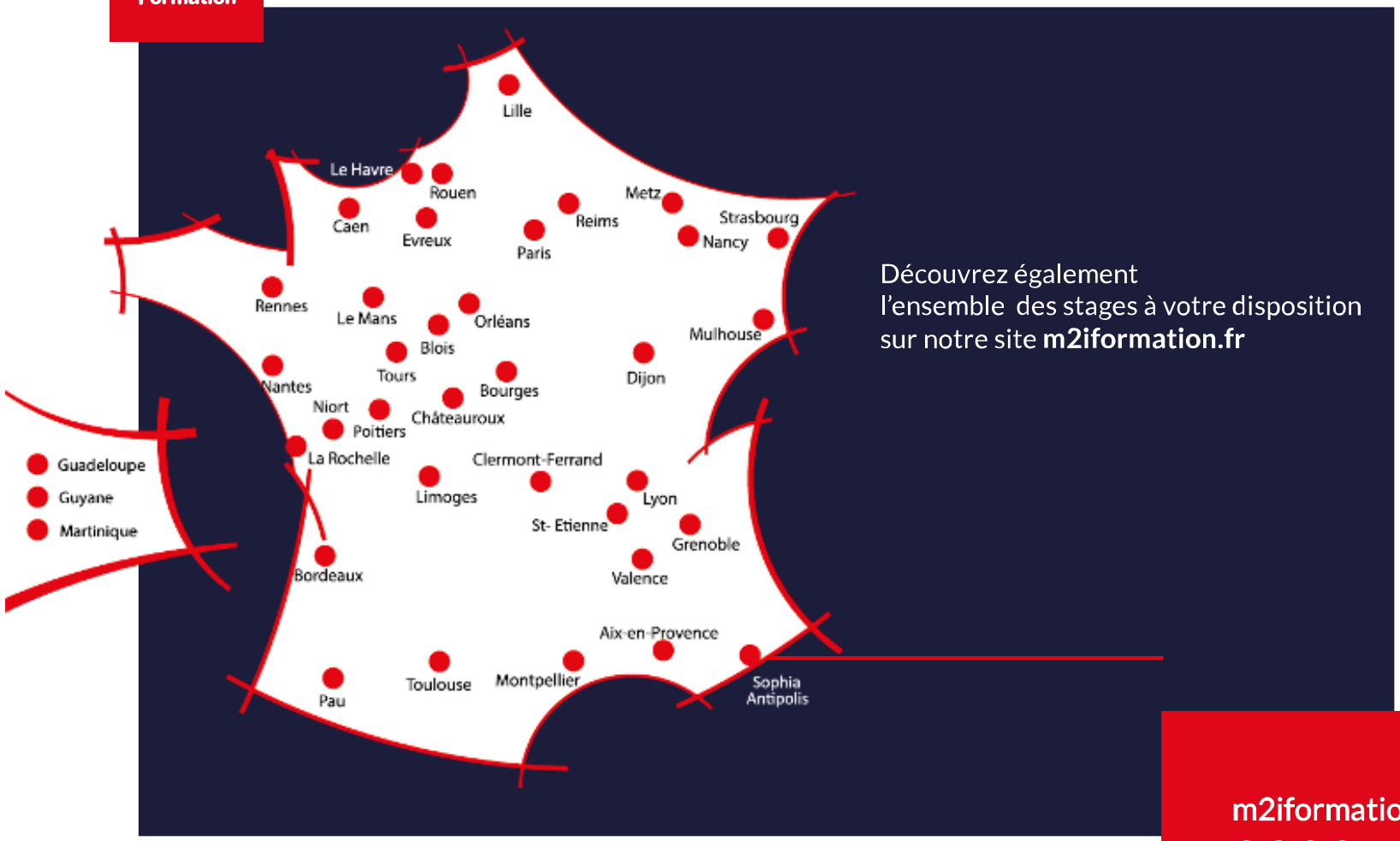
Synthèse des spécifications



UTOPIOS
TRANSFORMATION DIGITALE INTELLIGENTE

Merci pour votre attention

Des questions ?



Découvrez également
l'ensemble des stages à votre disposition
sur notre site **m2iformation.fr**

m2iformation.fr