

Présentation de la Framework .NET

M.T MOHTADI

Plan

- Qu'est ce que .Net Framework?
- Quels sont ses objectifs?
- Quels sont ses services?
- Quelle est son architecture?
- Comment est structurée une application .Net?
- Quel est le processus de création d'un Application .Net

Qu'est ce que

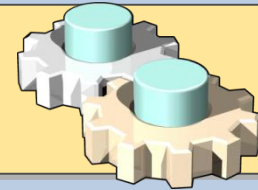
- .NET (prononcez «Dotnet») est un standard proposé par la société Microsoft, pour le développement d'applications d'entreprises multi-niveaux, basées sur des composants.
- C'est un environnement d'exécution qui gère les applications ciblant .NET Framework. Il se compose d'un Common Language Runtime, qui assure la gestion de la mémoire et d'autres services système, et d'une bibliothèque de classes étendue, qui permet aux programmeurs de tirer parti d'un code solide et fiable pour tous les domaines importants du développement d'applications.



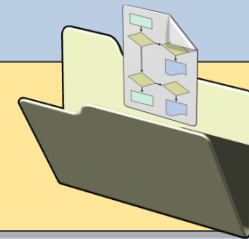
Qu'est-ce que le .NET Framework ?

Microsoft®
.net Framework

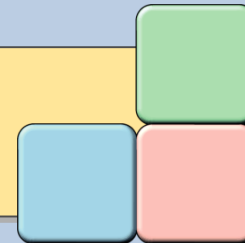
Common Language Runtime



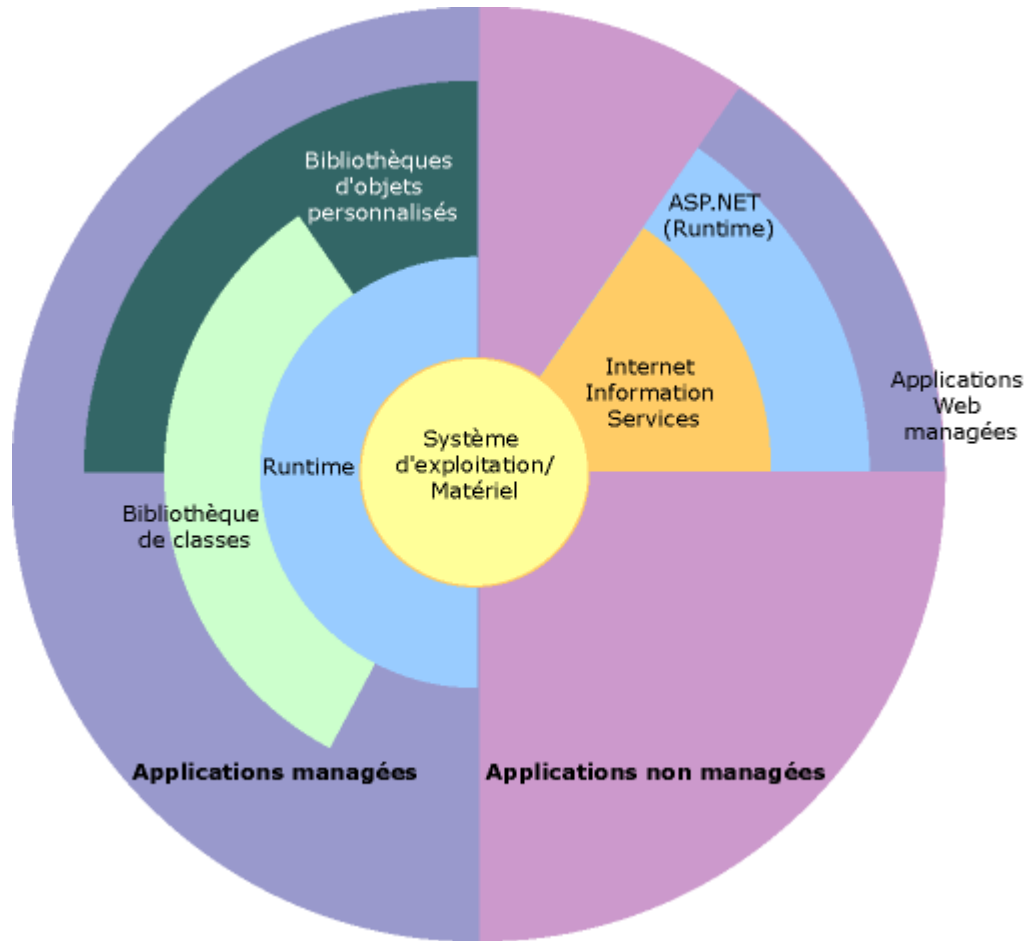
Class Library



Development Frameworks



Qu'est-ce que le .NET Framework ?



Objectifs

Fournir un Modèle de programmation unifié

- Différents types d'applications
 - Web, Windows, Mobile
- Le choix des langages :
 - Le choix des langages :
VB.NET, C#, C++, J#, ...
- Différents systèmes cibles
 - MS Windows, Windows Phone, Xbox360

Objectifs

Environnement d'exécution robuste et sécurisé

- **Common Language Runtime (CLR)**
- **Code managé : Garbage Collector, Type Safe..**
- **Runtime Security**

Déploiement et administration simplifiés

- **Xcopy, Versioning**

Intégration de l'existant

- **Multiples langages de programmation**
- **Interopérabilité COM et Win32**

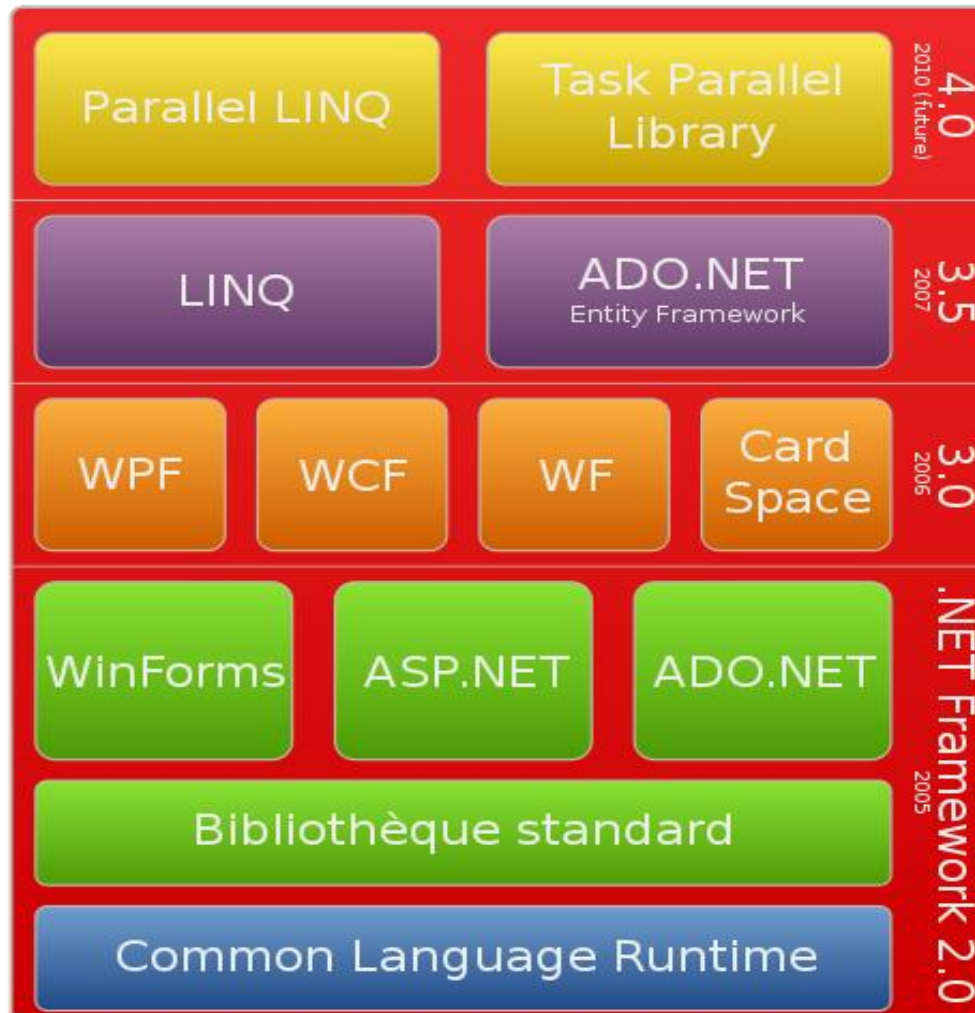
Développement d'applications simplifié

- **Elever le niveau d'abstraction**
 - **classe, interface, héritage, propriété, constructeur, exception...**
 - **Élimination complète de la plomberie COM**
- **Bibliothèque de classe riche et extensible**
- **Intégration native des standards Web (HTTP,XML, SOAP, ...)**

les services Offerts:

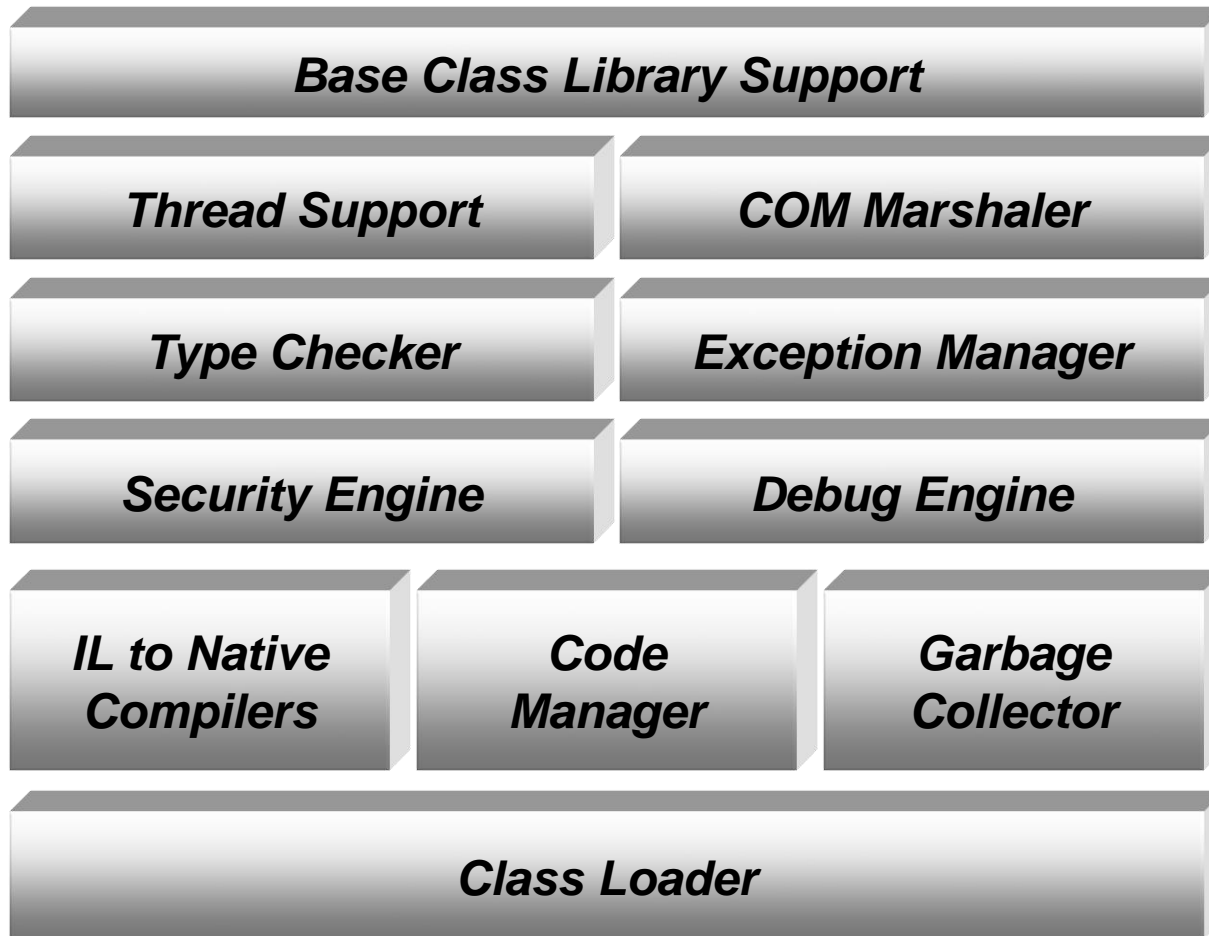
- Gestion de la mémoire.
- Système de types commun..
- Bibliothèque de classes étendue.
- Infrastructures et technologies de développement.
- Interopérabilité des langages.
- Compatibilité des versions.
- Exécution côte à côte.
- Multi-ciblage.

Architecture .NET Framework

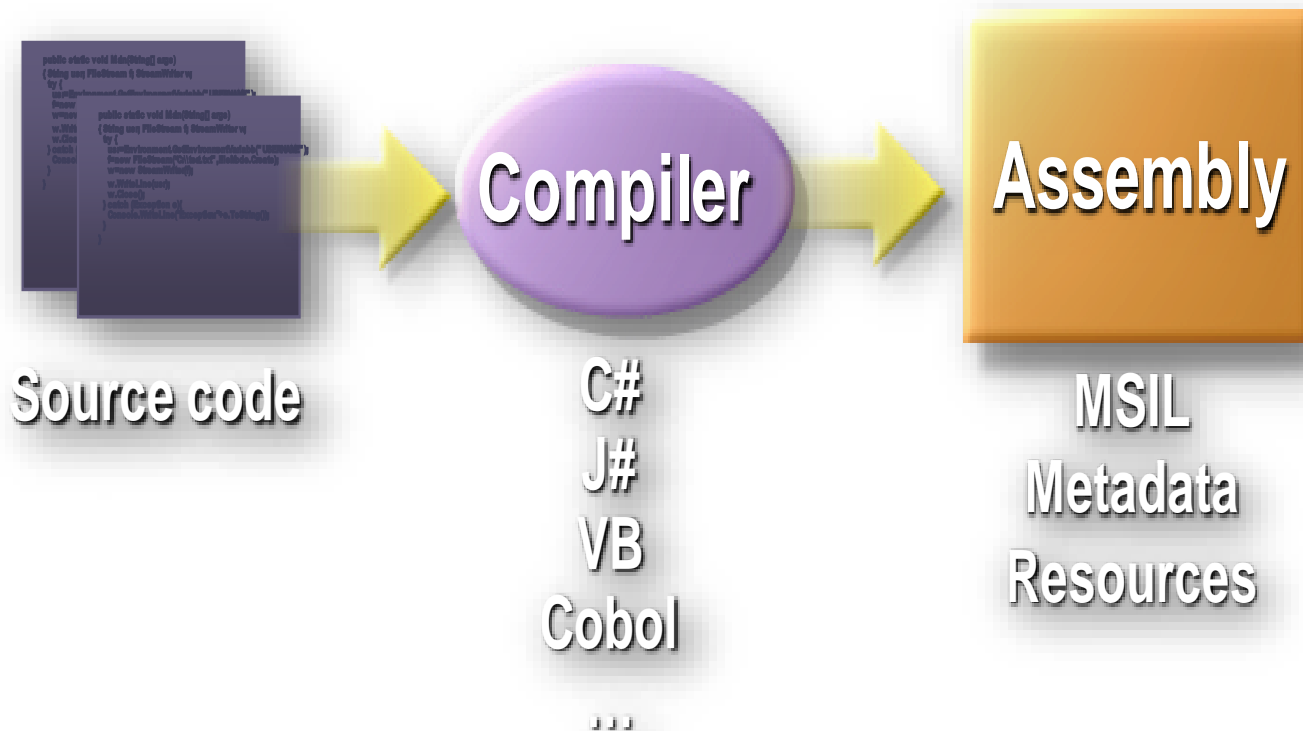


Pile de composants de .NET Framework

CLR : Vue d'ensemble



Structure d'une application .NET

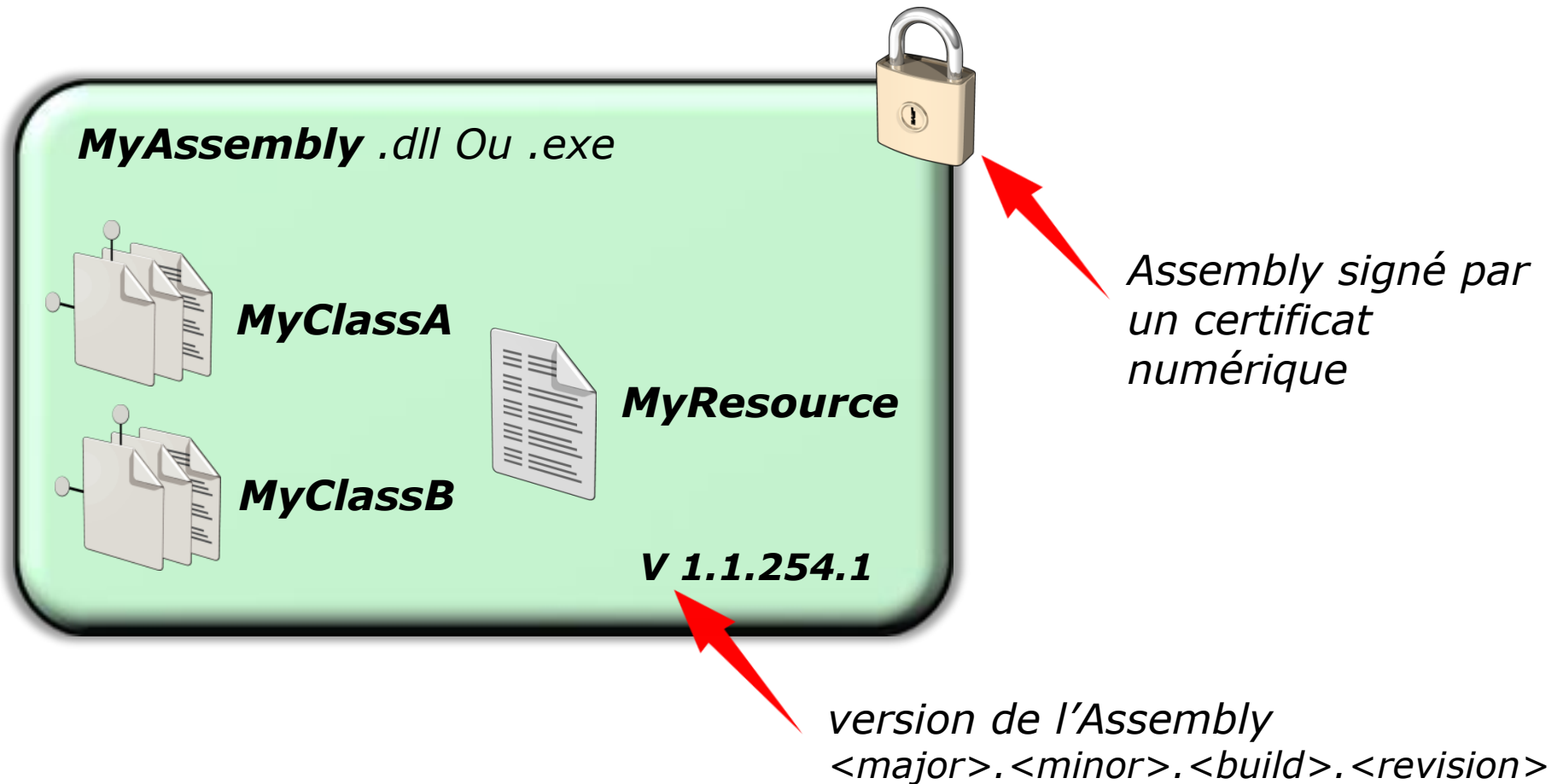


Structure d'une application .NET

- L'unité de base d'une application .NET est appelée un assemblage (*assembly*). Il s'agit d'un ensemble de code, de ressources et de [métadonnées](#).
- Un assemblage est toujours accompagné par un manifeste (*assembly manifest*) qui décrit ce qu'il contient : nom, version, types de données exposées, autres assemblages utilisés, instructions de sécurité.
- Un assemblage est composé d'un ou plusieurs modules qui contiennent le code.

Qu'est-ce qu'un Assembly?

Blocks de construction des applications .NET Framework qui forment une unité logique de fonctionnement

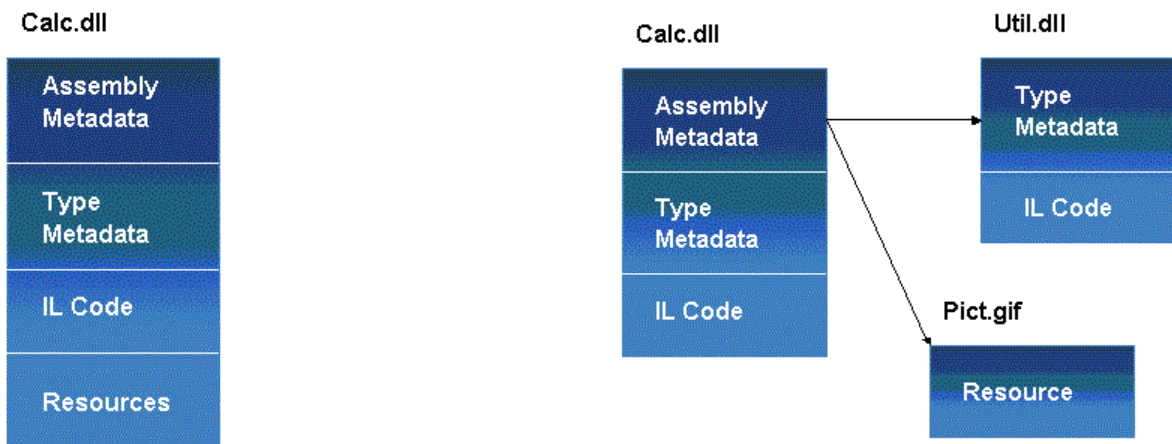


Assembly : Définition

- Unité de déploiement
- Point d'entrée d'exécution
- Unité de contrôle de sécurité
- Unité de contrôle de version
- Unité de réutilisation
- Unité de visibilité des types
- Unité d'exécution « cote à cote »

Assembly : contenu

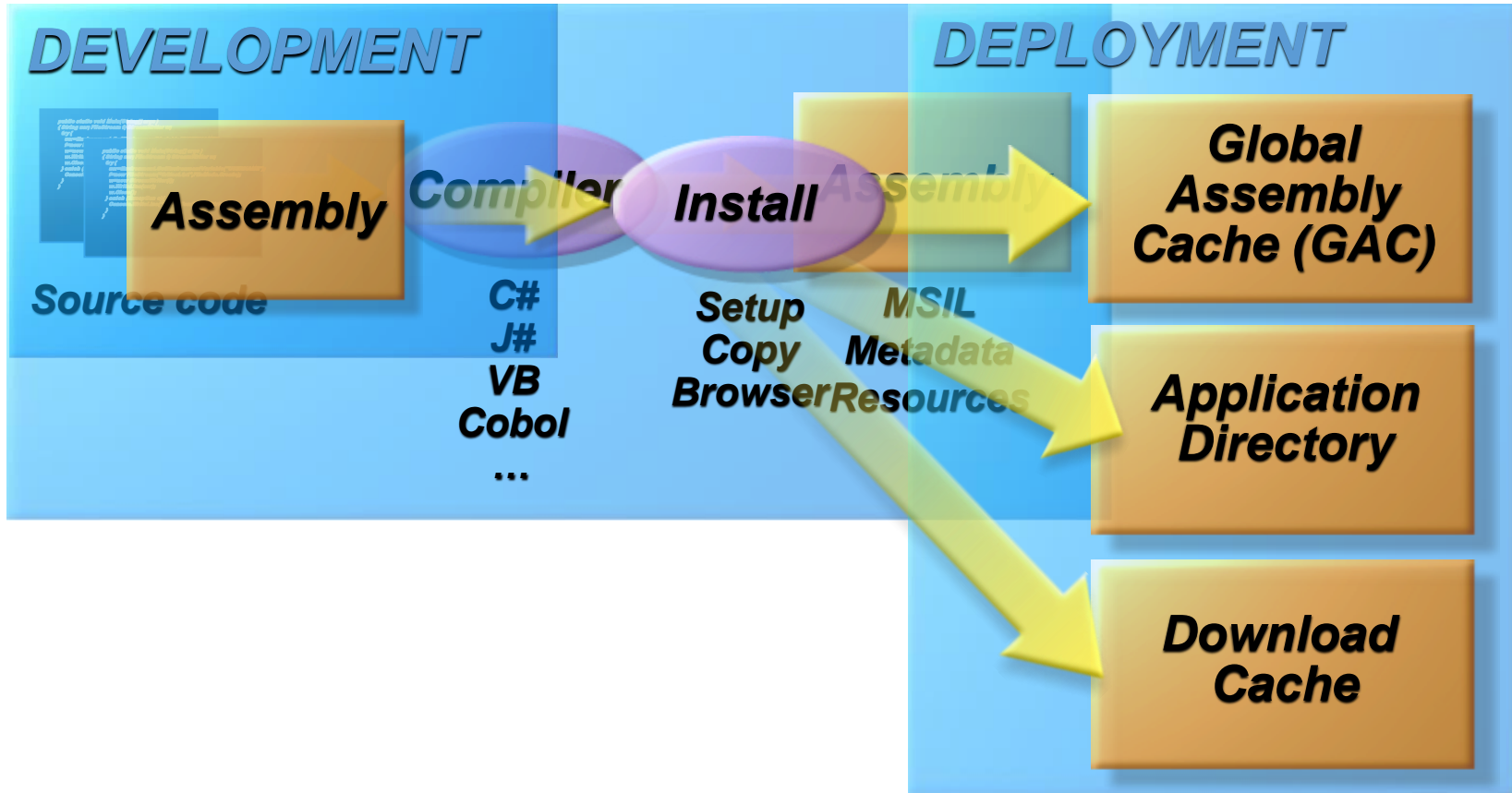
- Un manifest : metadonnées de l'assembly
- Les métadonnées décrivant les types utilisés
- Le code IL
- Les ressources utilisées (images,...)
- La liste des fichiers qui la compose



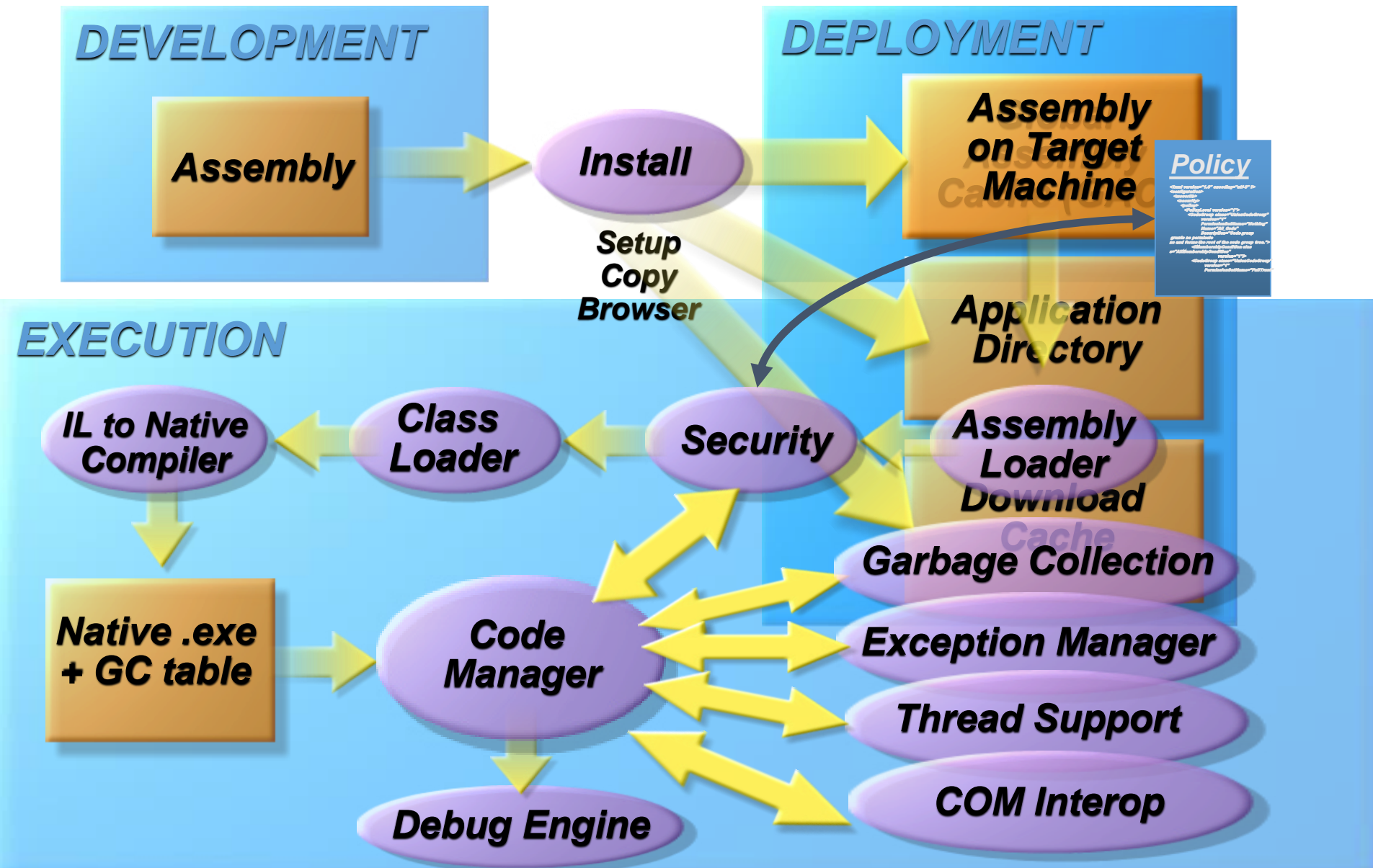
Assembly : 2 formes d'utilisation

- On considère 2 types d'assembly :
 - Privée : Dédiée à l'utilisation d'un programme particulier. Se situe généralement dans l'arborescence du programme qui l'utilise.
 - Partagée : Utilisée par un ensemble de programme différents. Se situe généralement dans le GAC (Global Assemblies Cache)

Processus : 2. Le Déploiement



Processus: 3. L'exécution



DEMO

DEMO

Première Application en .NET