

# Introduction au .NET Framework

Stéphane RZETELNY - Ingénieur 2000 - Informatique Réseaux  
Octobre 2002

# PLAN

- Qu'est-ce que le .NET Framework ?
- Les objectifs du .Net Framework
- Notion d'assemblée
- Comment ça fonctionne
- Interopérabilité
- Standardisation du .Net Framework
- .Net Framework et Linux

# Qu'est-ce que le .NET Framework ?

- Un nouvel environnement de développement et d'exécution
- L'introduction d'un nouveau langage : C#
- Une bibliothèque de classes extensibles, regroupées par namespace en C#

# Objectifs du .NET Framework

## 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
- Compatibilité « side by side »

## Développement d'applications simplifié

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

# CLR : Vue d'ensemble



# Assembly : Définition

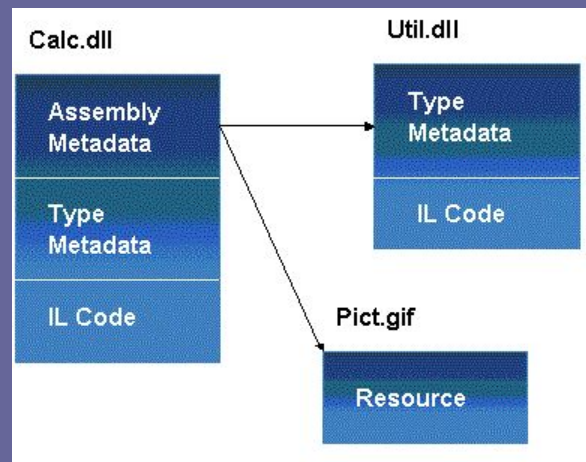
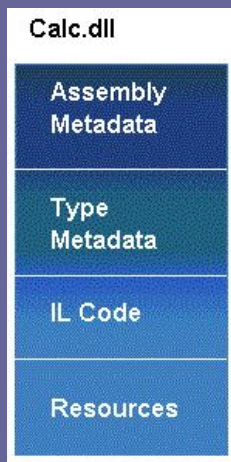
- Unité de déploiement
- Point d'entrée d'exécution
- Unité de contrôle de sécurité
- Unité de contrôle de version
- Unité de réutilisation
- Unité de visibilité des types
- Unité d'exécution « cote à cote »

# Assembly : 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)

# Assembly : contenu

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



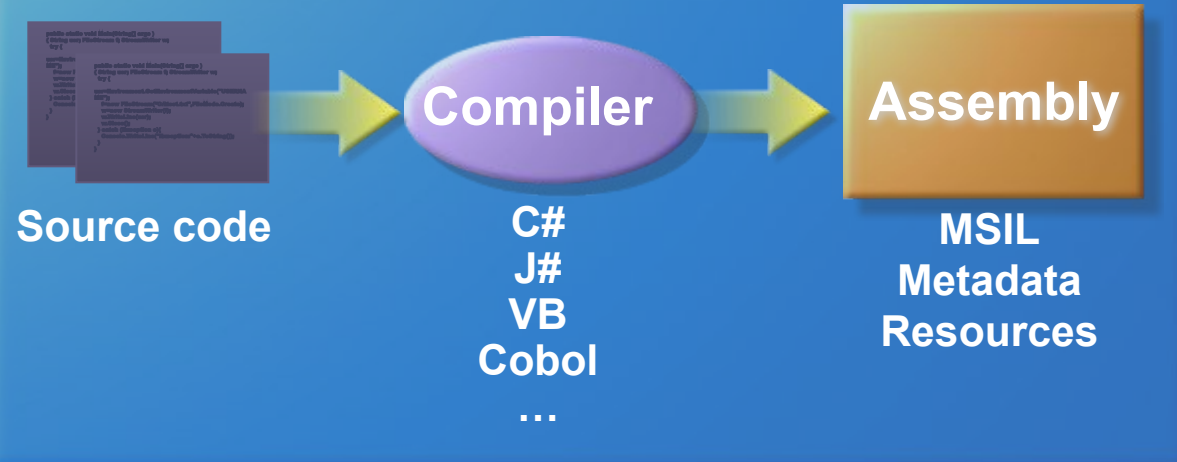


# Global Assemblies Cache (GAC)

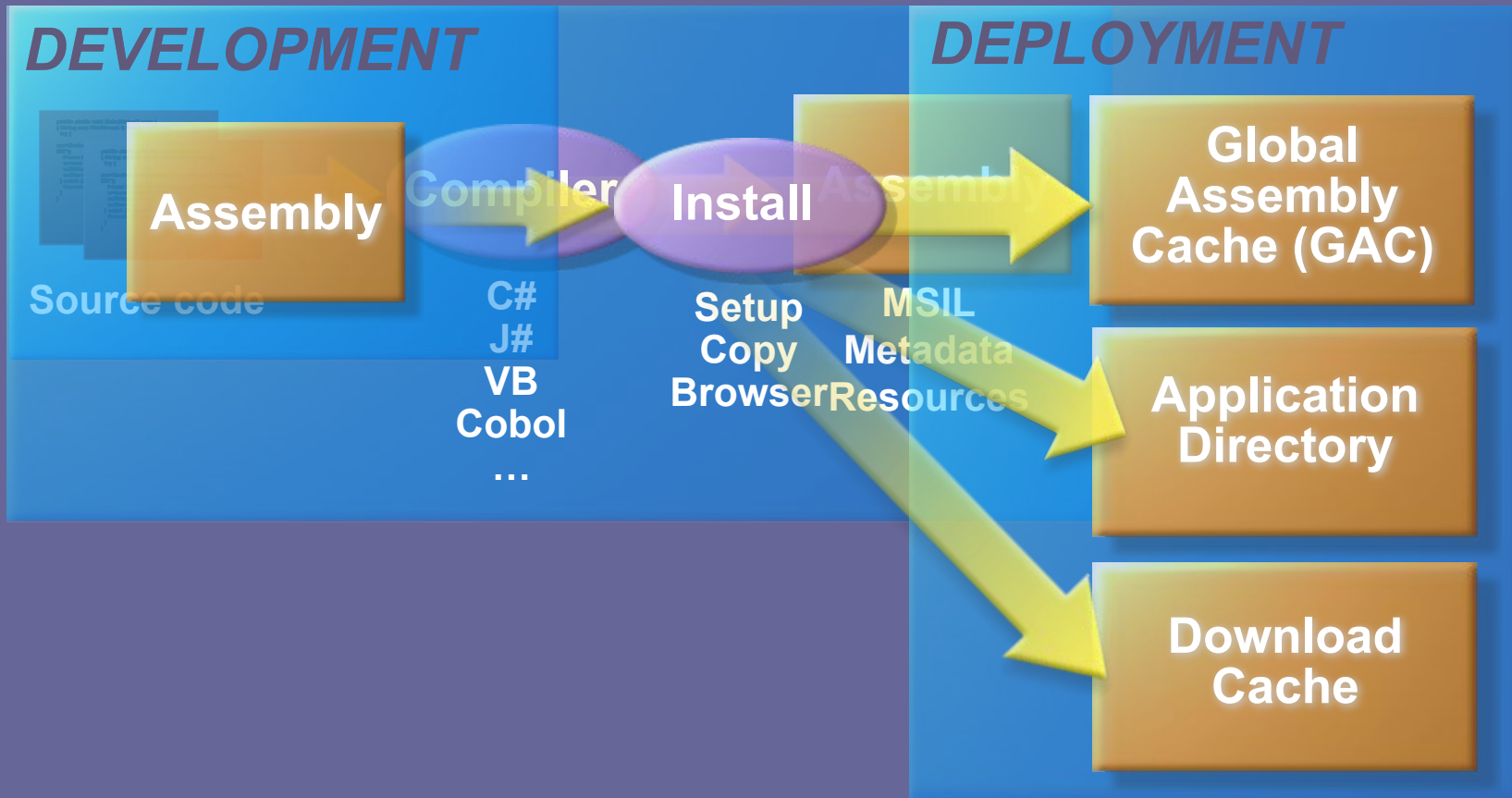
- Un cache accueillant les assemblées partagées par plusieurs applications
- Doivent être définies avec des « noms forts »
  - Grâce au mécanisme de clé privée, clé publique
  - Pour garantir l'unicité d'une version d'une assemblée
  - Pour garantir l'intégrité de l'assemblée
- Doivent faire référence à des assemblées fortement nommées !

# Comment ça marche : 1. Le Développement

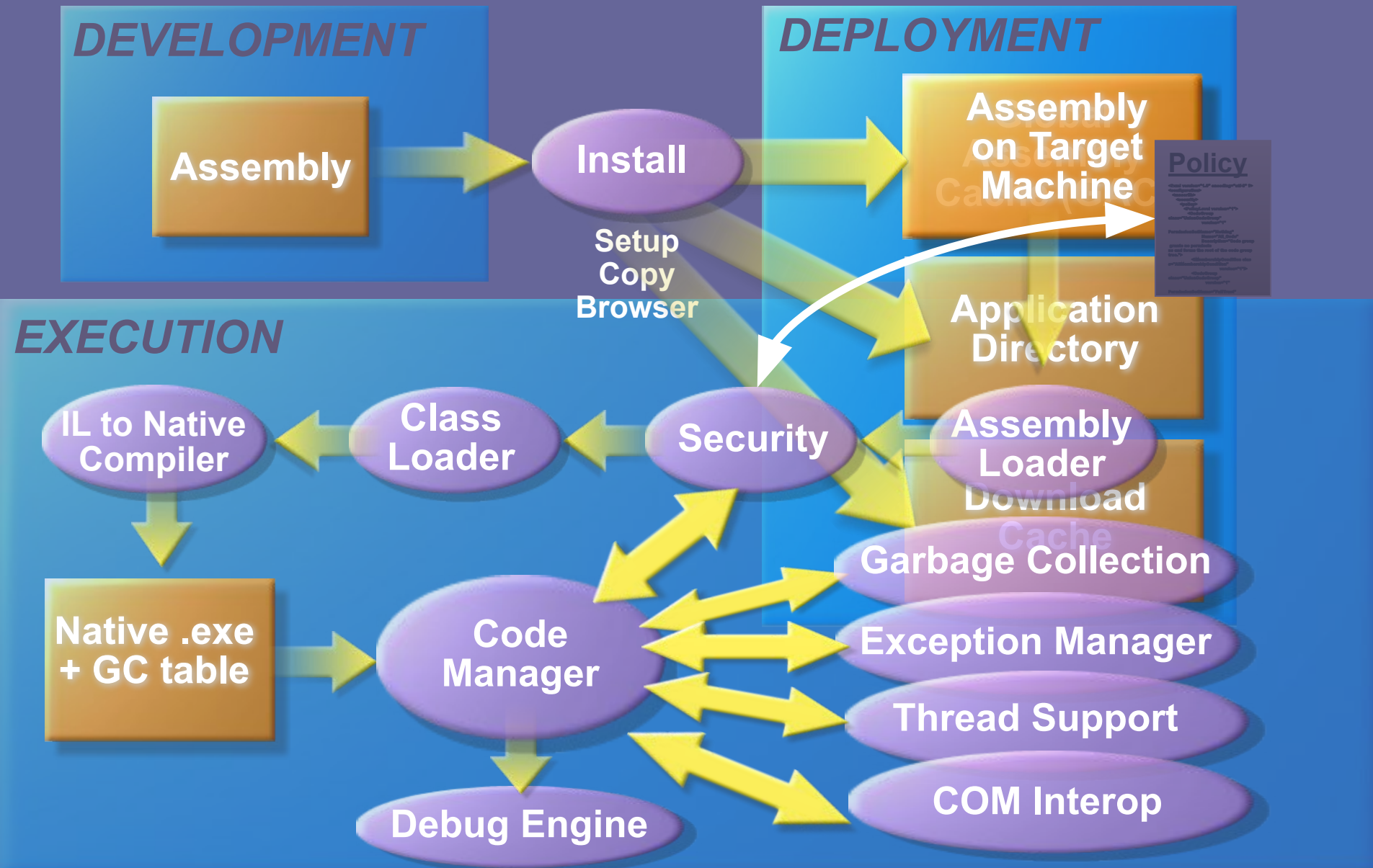
## *DEVELOPMENT*



# Comment ça marche : 2.Le Déploiement



# Comment ça marche : 3. L'exécution



# Interopérabilité

- Interopérabilité des langages, grâce au langage intermédiaire (IL)
- Interopérabilité avec COM
- Interopérabilité avec Win32

# Standardisation

- **C# et CLI qui sont déposés à l'ECMA**

CLI (Common Language Infrastructure) est très précis quand à ce que l'on doit trouver dans le "framework " et le "runtime " et laisse champ libre sur la manière de l'implémenter

- **ECMA-334**, C# (C sharp) Language Specification.
- **ECMA-335**, Common Language Infrastructure, approved by the ECMA General Assembly of 13th December 2001.
- **ECMA TR/84**, related to Standard ECMA-335

# .Net Framework et Linux !

- Début du « portage » du framework .Net
- Différents projets :
  - Mono par le créateur de gnome
  - GNU Portable DotNet
  - Rotor (FreeBSD)

# Sites Internet

- <http://www.gotdotnet.com>
- <http://www.dotnetguru.org>
- <http://www.dotnet-fr.org/>
- <http://msdn.microsoft.com/netframework/>
- <http://www.go-mono.com/>
- <http://www.southern-storm.com.au/>