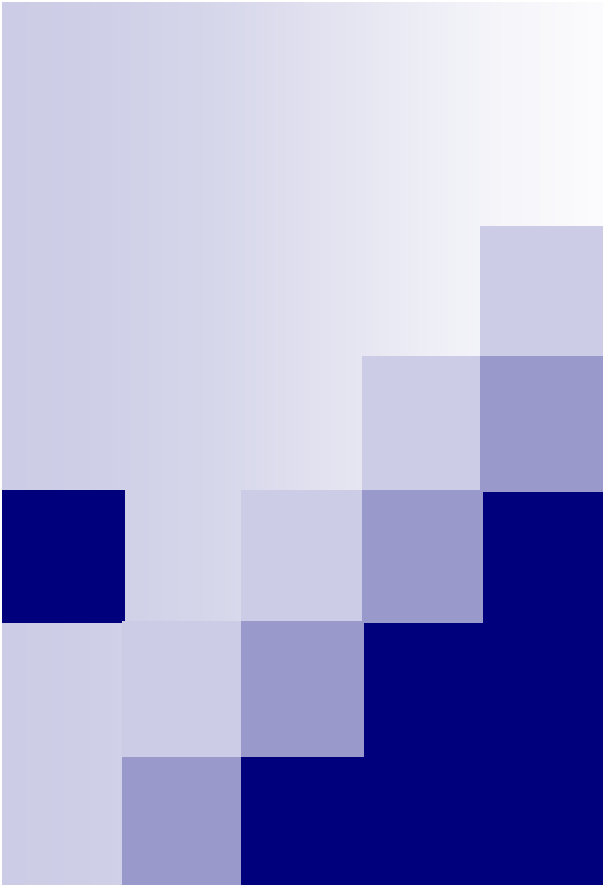




.NET Framework

Konzepte von Betriebssystem-Komponenten

11.01.2005

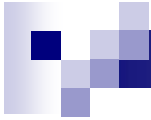
Wilhelm Haas

wilhelm.haas@informatik.stud.uni-erlangen.de



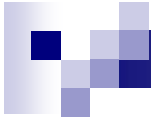
Gliederung

- Geschichte
- Ziele
- .NET Framework
 - Struktur
 - Freie Wahl der Sprache
 - Common Language Specification (CLS)
 - Microsoft Intermediate Language (MSIL / IL / CIL)
 - Assemblies
 - Framework Class Library (FCL)
 - Common Language Runtime (CLR)
- Gegenwart & Zukunft
- Literatur



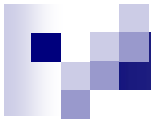
Geschichte

- 1998 bekannt gabe der neuen Entwicklung
- „COM Version 3.0“ -> „NGWS“
- Seit Juli 2000 „.NET Framework“
- Entwicklungsplattform für das Internet und gleichzeitig eine neue API
- Einfluss von herkömmlichen Ansätzen

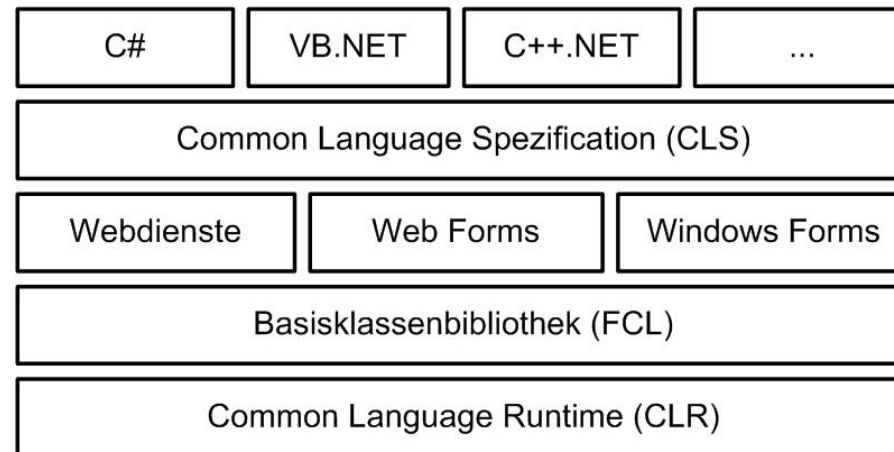


Ziele

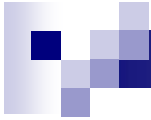
- Lösung aktueller Probleme
 - Spezielle und unterschiedliche Bibliotheken
 - Betriebssystemabhängigkeit
 - Fehleranfälligkeit
 - Interoperabilität



Struktur

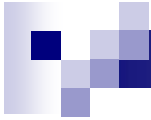


Viele Programmiersprachen deren Compiler, durch die CLS standardisierten, Intermediate Language Code erzeugen. Unterstützt werden die Programmierer durch viele Klassenbibliotheken, die aus der FCL aufgebaut sind. Die VM für .NET-Anwendungen ist die CLR.



Freie Wahl der Sprache

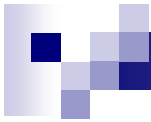
- Idee der „freien Wahl des passenden Werkzeugs“
- Wirtschaftliche und programmiertechnische Aspekte
- Jederzeit erweiterbar
- Zur Zeit etwa 20 Programmiersprachen



Common Language Specification (CLS)

- Regelwerk für das Compilieren in die MSIL
- Ermöglicht die Erweiterung der Sprachfamilie
- Common Type Specification (CTS)

Bringt die Vielzahl verschiedener Datentypen auf einen gemeinsamen Nenner. Beschreibt sozusagen die Struktur der Basisdatentypen.



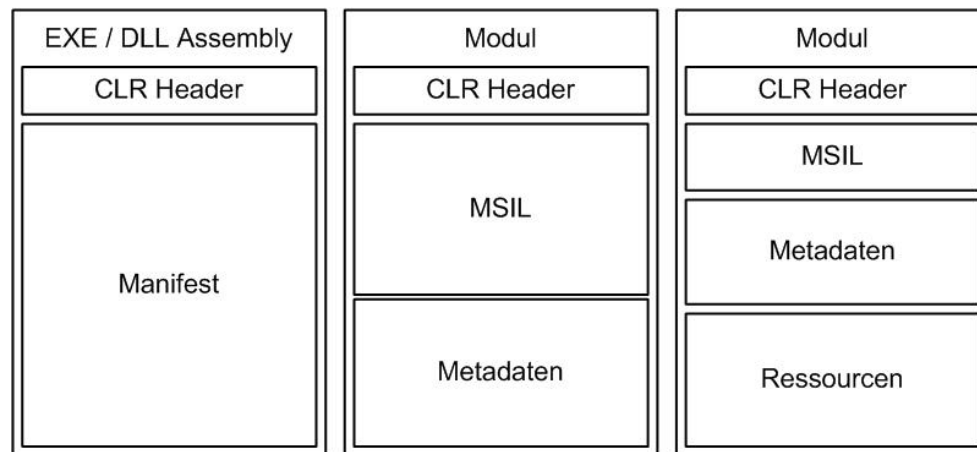
Microsoft Intermediate Language (MSIL / IL /CIL)

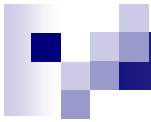
- Streng typisierte, prozessorunabhängige objektorientierte Assembler-Sprache
- Besitzt alle Konzepte der modernen Programmiersprachen
- Übersetzung in MSIL erfolgt nach den Regeln der CLS

Assemblies

- Logischer Verbund einer oder mehreren EXE- oder DLL-Dateien
- Enthalten

- ☐ CLR Header
- ☐ Manifest
- ☐ MSIL-Code
- ☐ Metadaten
- ☐ Ressourcen



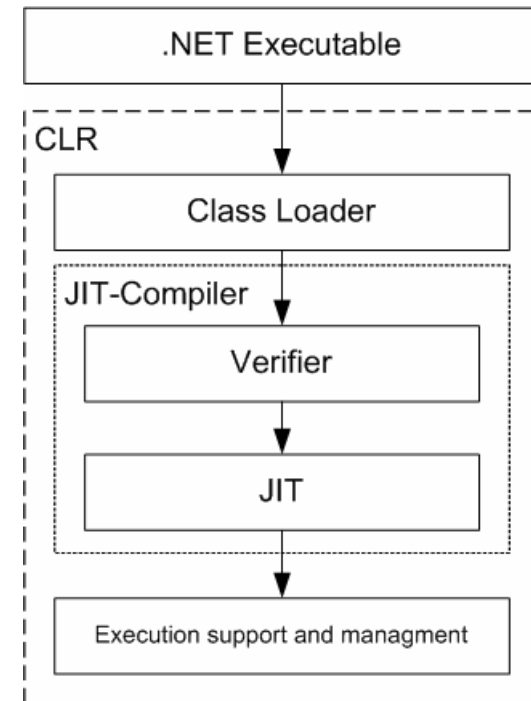


Framework Class Library (FCL)

- Reihe von verschiedenen Klassen
 - Datenverwaltung
 - XML-Manipulation
- Oberklasse ist die `System.Object`
- Version 1.0 enthält 2246 Klassen
- Aufgebaut aus den Basisklassen (BCL)

Common Language Runtime (CLR)

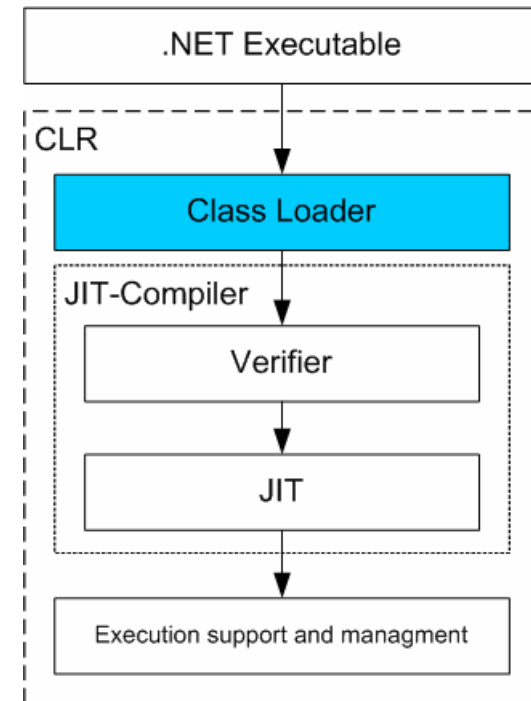
- VM bzw. VES des .NET-Frameworks
- Class Loader lädt benötigte Klassen
- Verifier ist für die Typsicherheit zuständig
- Just In Time - Compiler
- Execution support and management unterstützt den Ablauf einer Anwendung



Common Language Runtime (CLR)

Class Loader

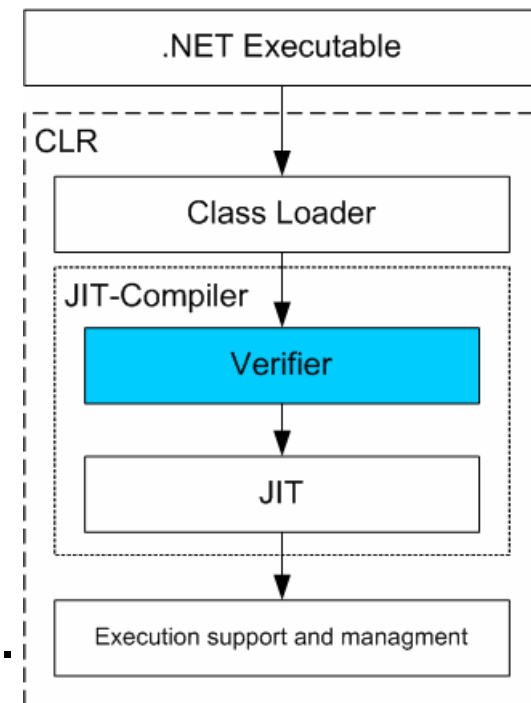
Lädt die für die Ausführung benötigten Klassen. Fügt zusätzlich zu jeder Methode eine Informationsstruktur hinzu.



Common Language Runtime (CLR)

Verifier

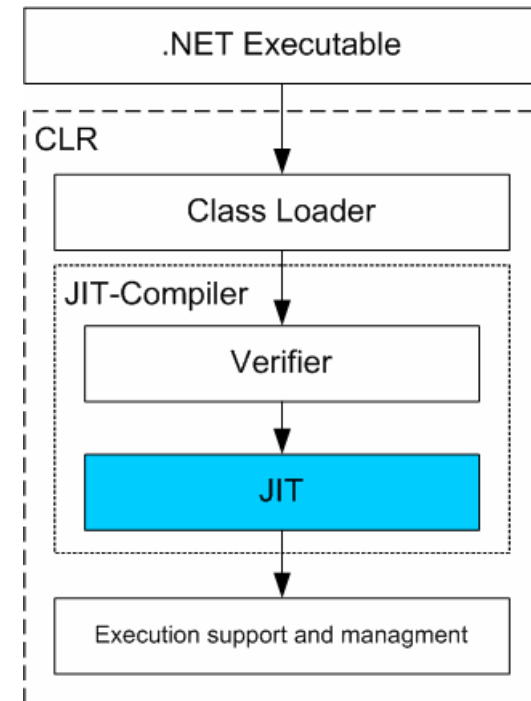
Ist ein Schutzmechanismus der für die Typsicherheit sorgt. Da das zur Laufzeit geschieht hat das Auswirkungen auf die Performance. Der Verifier kann durch einen Erwerb eines Zertifikats ausgeschaltet werden.



Common Language Runtime (CLR)

JIT – Just in Time

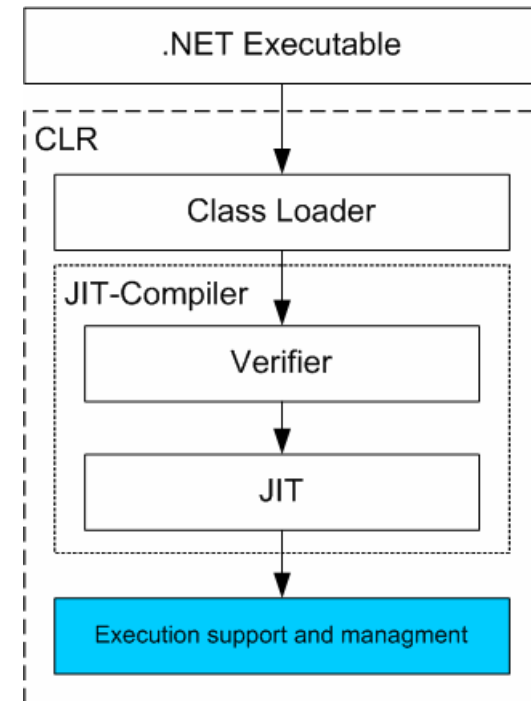
Compiliert zur Laufzeit den MSIL Code. Ein Vorteil ist die dynamische Compilierung das aber auch gleichzeitig ein Nachteil ist wegen der teuren Übersetzungszeit.

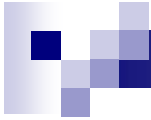


Common Language Runtime (CLR)

Execution support & management

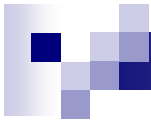
Unterstützt das Programm zur Laufzeit, z. B. mit einem Garbage collector.





Gegenwart & Zukunft

- .NET Framework als Version 2.0 erhältlich
- Viele Kinderkrankheiten sind beseitigt worden
- In der Wirtschaft schon weit verbreitet
- Wird mit dem Betriebssystem Windows Longhorn ausgeliefert



Literatur

- **MSDN .NET Framework Technologie**
Link: <http://msdn.microsoft.com/netframework/technologyinfo/default.aspx>
- **O'Reilly .NET Framework Essentials, 2nd Edition**
Autor: Thuan Thai, Hoang Q. Lain
ISBN: 0-596-00302-1
- **Introducing Microsoft .NET, 2nd Edition**
Autor: David S. Platt
ISBN: 0-735-61571-3