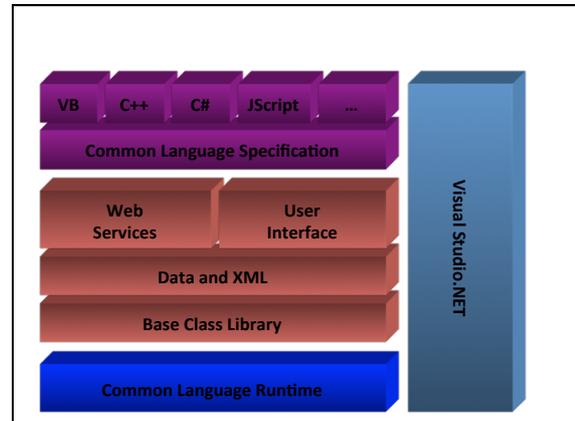


Introduzione al .NET Framework

A. Ferrari



Concetti chiave

- **CLS:** Common Language Specification
- **MSIL:** Microsoft Intermediate Language
- **CLR:** Common Language Runtime
 - gestisce il codice in esecuzione fornendo servizi quali la gestione della memoria, dei thread.
- **CTS:** Common Type System

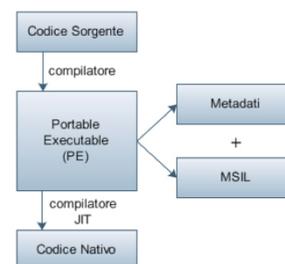
CLS

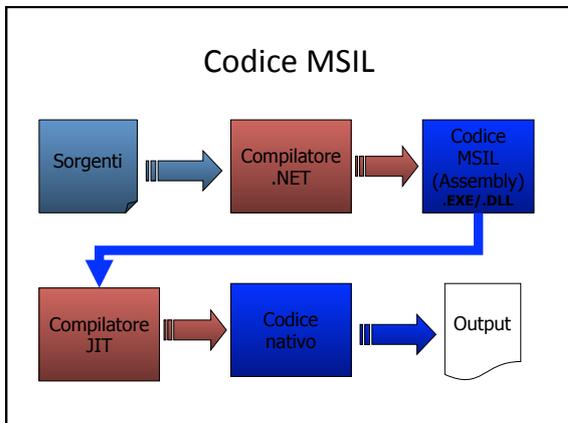
- Per essere certi che il proprio codice gestito sia accessibile ad altri indipendentemente dal linguaggio utilizzato .NET fornisce delle specifiche note col nome di Common Language Specification (CLS) che definiscono un set di funzionalità fondamentali dei linguaggi: se il codice sviluppato utilizza esclusivamente le funzionalità CLS nelle API che espone all'esterno allora l'accesso a tale codice è garantito a tutti i linguaggi che supportano le specifiche stesse.

Esecuzione in .NET

- Viene sviluppato il codice sorgente utilizzando un linguaggio in grado di supportare le funzionalità offerte dal CLR (ad esempio C#)
- il codice sorgente viene compilato in un Portable Executable (PE) contenente metadati e codice MSIL (Microsoft Intermediate Language)
- il codice MSIL viene progressivamente tradotto in codice nativo da un compilatore JIT (Just In Time)
- il codice nativo viene eseguito

Esecuzione





Compilazione

- In ambiente .NET dalla compilazione del codice sorgente non si ottiene subito il codice nativo ma si ottiene un linguaggio intermedio (Microsoft Intermediate Language) che essendo indipendente dall'architettura è assolutamente portabile da un sistema all'altro.

Esecuzione

- Il codice nativo viene ottenuto mediante una seconda compilazione detta JIT (Just In Time) che converte il linguaggio intermedio in codice nativo per l'architettura sulla quale il programma sta "girando".

JIT

- Il codice MSIL è costituito da istruzioni indipendenti dalla CPU non può essere eseguito direttamente ma occorre eseguire un'ulteriore compilazione in linguaggio nativo.
- I compilatori che effettuano la conversione da Microsoft Intermediate Language a codice nativo prendono il nome di compilatori JIT (Just In Time): il Common Language Runtime fornisce un compilatore JIT per ogni architettura supportata questo rende il codice MSIL altamente portabile da un sistema all'altro (è ciò che accade anche con Java e la Java Virtual Machine).
- I compilatori JIT prendono questo nome perché prendono in considerazione l'ipotesi che parte del codice MSIL non venga mai chiamato durante l'esecuzione pertanto piuttosto che convertire tutto il codice contenuto in un PE in codice nativo, viene convertito soltanto il codice MSIL necessario in fase di esecuzione ed il codice nativo risultante viene memorizzato per soddisfare le eventuali chiamate successive.

.NET e Java

- La compilazione in .NET è molto simile concettualmente alla compilazione che avviene in ambiente Java: il bytecode, risultato della compilazione Java, viene di volta in volta interpretato dalla Java Virtual Machine e tradotto nel codice nativo appropriato per l'architettura che si sta utilizzando.
- La compilazione Just In Time non fa altro che tradurre il codice espresso in Microsoft Intermediate Language (MSIL) in codice nativo eseguibile dall'architettura utilizzata.
- Per ottimizzare le prestazioni il codice compilato Just In Time viene memorizzato per far fronte a chiamate future dello stesso.



Interoperabilità

- Common Language Runtime fornisce il supporto per l'interoperabilità dei linguaggi ovvero la capacità di interagire con codice scritto in un linguaggio di programmazione differente.
- Fondamentalmente l'interoperabilità è garantita dall'utilizzo di un sistema di tipi comuni e dall'utilizzo dei metadati che definiscono un meccanismo unico e soprattutto indipendente dal linguaggio per l'archiviazione ed il recupero delle informazioni sui tipi utilizzati.