L'infrastruttura .NET

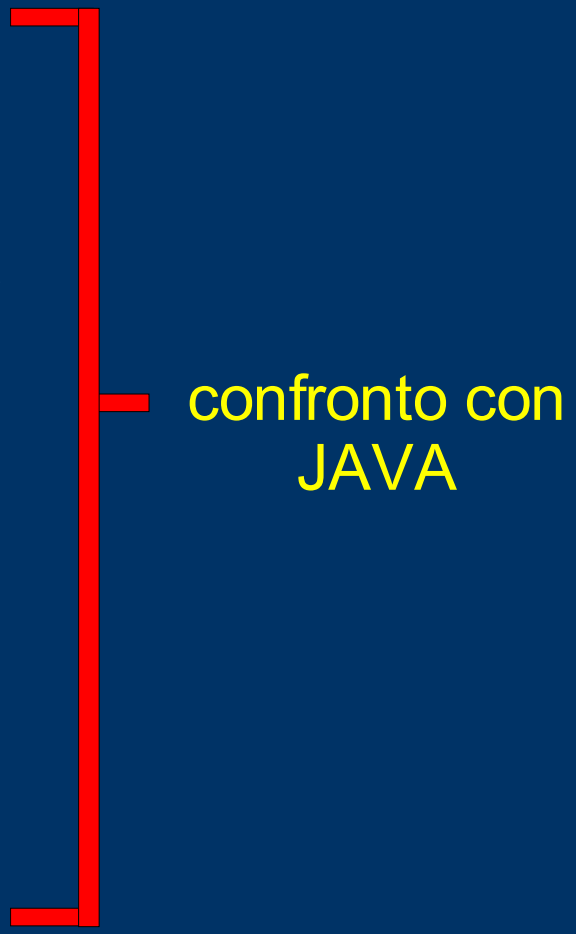
Breve presentazione dell'infrastruttura .NET di Microsoft

Dott. Mario Di Raimondo
Dipartimento di Matematica e Informatica
Università degli Studi di Catania

contenuti presentati al
Microsoft Research Academic Days 2003
“.NET Mobile and Distributed Technologies”
(Torino, 29-30 Settembre, 1 Ottobre)

Sommario

- Motivazioni e Obiettivi
- Uno sguardo all'architettura .NET
- I linguaggi supportati
- Accenno a C#
- Microsoft e gli standard
- Conclusioni



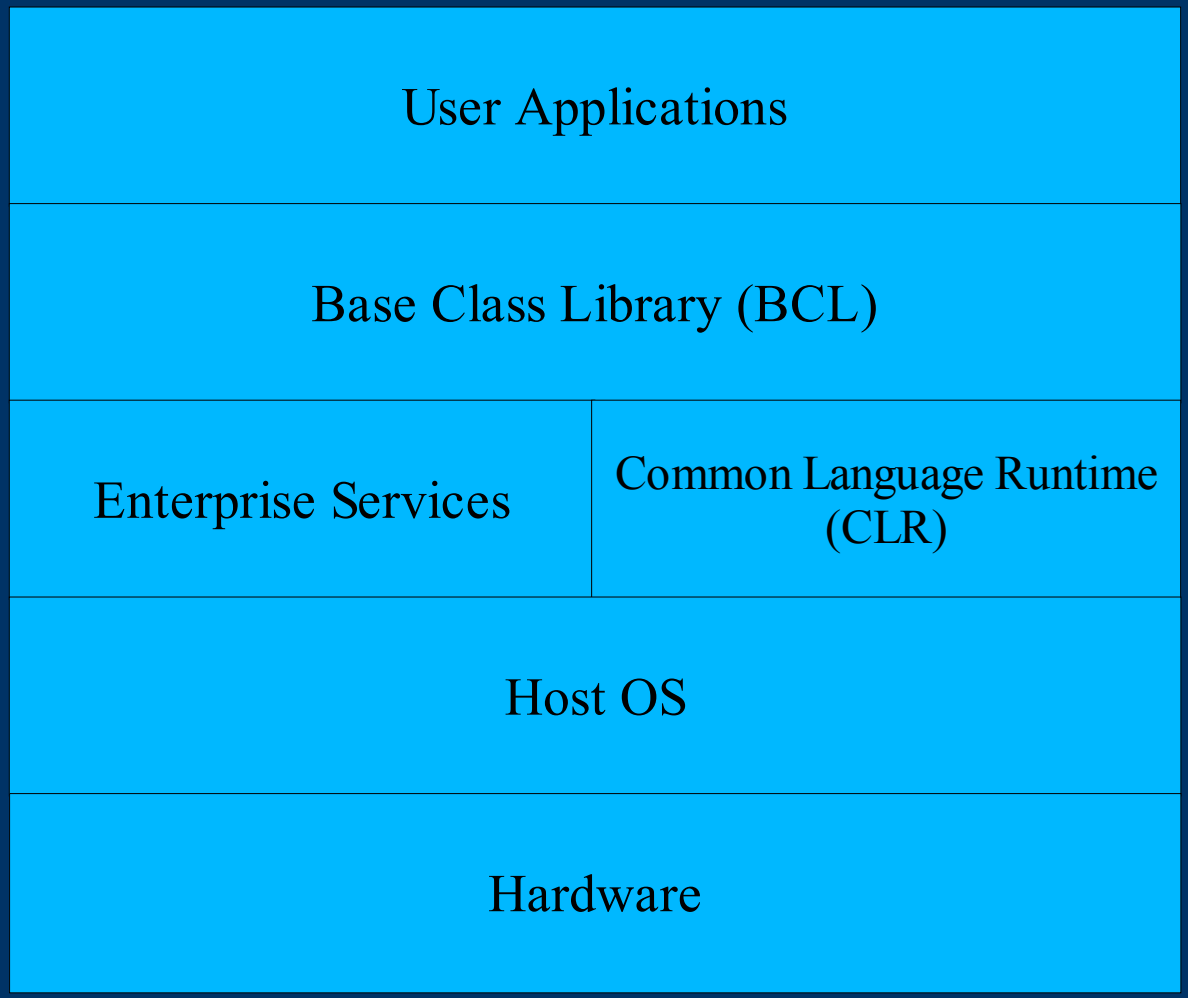
.NET

- si tratta di una “piattaforma software” su cui sviluppare applicazioni
- caratteristiche:
  - pensato per le applicazioni distribuite attraverso la rete
  - multilinguaggio
  - multipiattaforma (?!?)
  - semi-standardizzato

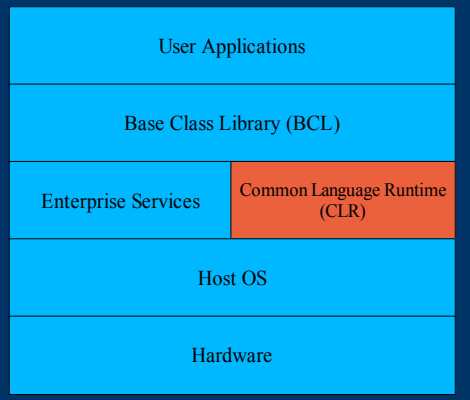
Motivazioni

- se non lo si può definire un clone di Java...
- ... si può sicuramente considerare la risposta Microsoft a Java! (o più correttamente, all'infrastruttura J2EE di Sun)
  - insinuazioni...
- semplificare la vita a chi sviluppa applicazioni complesse
  - Visual-Basic: poco espressivo
  - Visual-C: troppo complesso
- rendere le applicazioni portatili (come in Java)
- riassettare le immense (e disordinate) librerie MS

Architettura



Common Language Runtime (CLR)

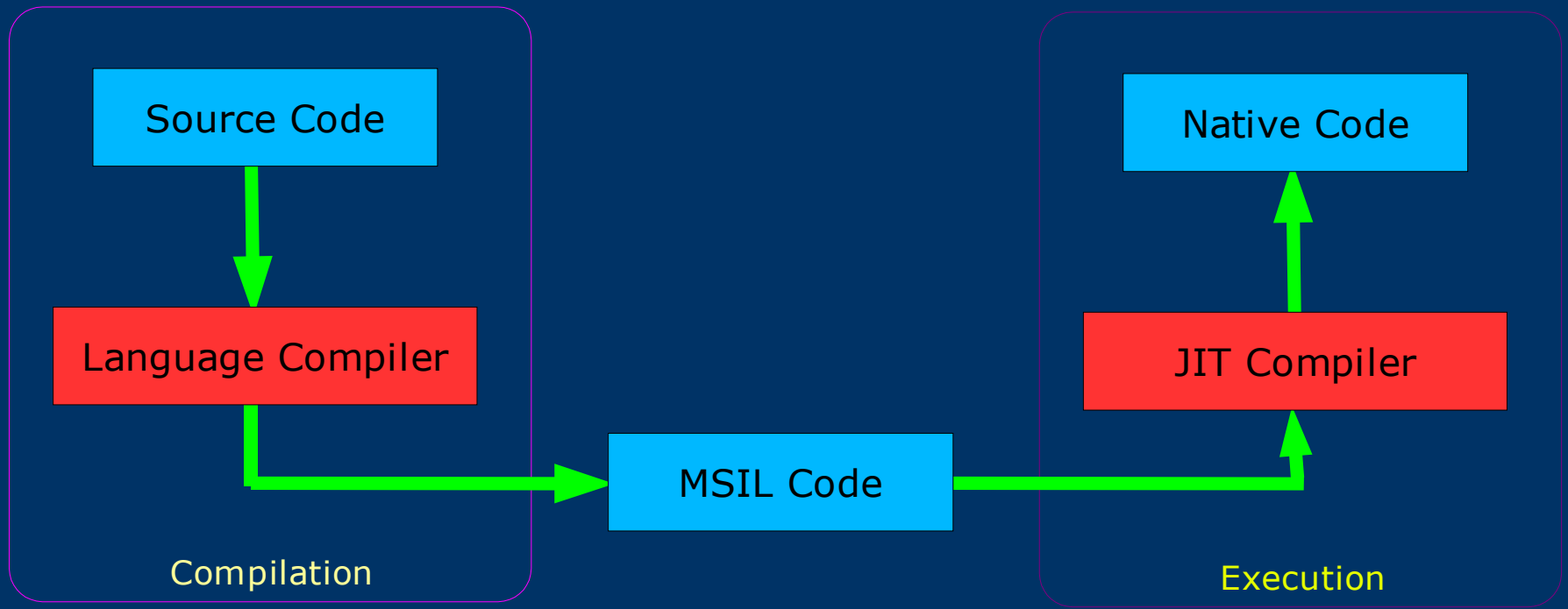


- cardine dell'architettura .NET
- ambiente indipendente dalla piattaforma per l'esecuzione delle applicazioni
- l'equivalente della Java Virtual Machina (JVM) nell'architettura Java
- offre una serie di funzionalità di base:
  - gestione della memoria (Garbage collection)
  - gestione della concorrenza
  - verifica sui tipi dei dati
  - controlli di sicurezza al momento del caricamento e dell'esecuzione dei moduli (disabilitabili)
  - controlli sui diritti di lettura/scrittura delle aree di memoria (si può disabilitare o gestire con permessi specifici)
  - controllo rigido delle versioni e gestione del “multi-versioning”... fine dell'inferno delle DLL?!
- a differenza di Java, non dipende da uno specifico linguaggio

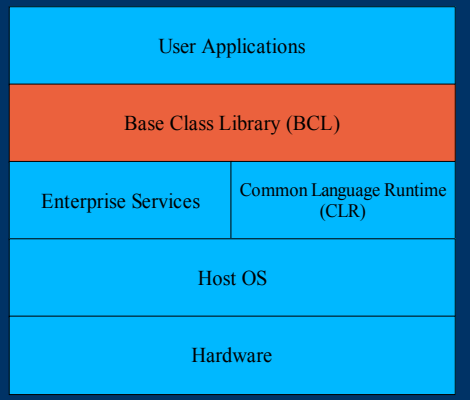
Microsoft Intermediate Language (MSIL) e il JIT

- si tratta del linguaggio intermedio in cui vengono compilate le applicazioni .NET
- è l'equivalente del byte-code della JVM
- è più ad alto livello rispetto al byte-code (esempio: operazioni senza tipi), quindi:
  - non è prevista l'interpretazione
  - viene compilato in codice nativo al momento dell'esecuzione (JIT)
    - **JIT "normale"**: traduce il codice MSIL in codice nativo e applica varie ottimizzazioni; richiede parecchie risorse (computazionali e di memoria)
    - **EconoJIT**: traduce il codice MSIL in codice nativo in modo basilare e veloce, senza ottimizzazione; specifico per sistemi embedded e con poche risorse
    - **PreJIT** (o **NGen** = "Native [code] GENeration"): insieme alla copia del codice in MSIL viene mantenuta anche una copia con codice nativo ottimizzato; tecnica parzialmente utilizzata per le classi della BCL
    - **OptJIT** (solo un idea per il futuro): ottimizzare la compilazione on-the-fly utilizzando informazioni addizionali inserite nel MSIL in modo da velocizzare anche la compilazione
  - al momento le tecnologie di esecuzione JIT di Java sono più avanzate
    - JVM profilanti e ottimizzazioni incremantali

Esecuzione del codice in .NET

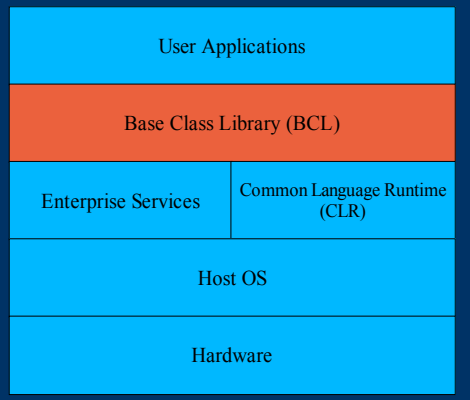


Base Class Library (BCL)



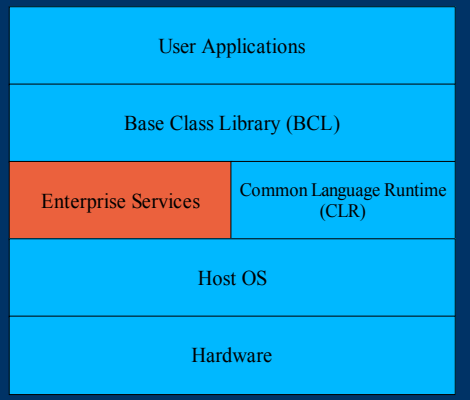
- insieme delle classi di base della piattaforma
  - analoghe alla Standard Library di C/C++ e ai package java.* di Java
- incapsulano le principali funzionalità di un sistema di elaborazione:
  - tipi dato complessi (collection)
  - networking
  - accesso al File System
  - interfaccia utente
  - sicurezza
  - programmazione concorrente
  - XML
  - etc...

Base Class Library (BCL)



- la BCL è suddivisa in namespace (come Java) a partire dal namespace System
- sono presenti, inoltre, funzionalità tipiche di Windows:
  - accesso al registro di sistema
  - interoperabilità con COM (riuso dei vecchi componenti)
  - funzionalità di basso livello
- **Aspetto positivo:** in questo modo hanno riassettato l'intero parco librerie di Windows che così risultano accessibili attraverso una gerarchia omogenea e coerente

Enterprise Services



- si tratta di servizi base per la realizzazione di applicazioni .NET
- a rigore non sono parte integrante dello standard, ma sono essenziali per lo sviluppo di applicazioni enterprise
- corrispondono alle applicazioni server MS (SQL server, IIS, BizTalk, etc...) e non sono applicazioni .NET (quindi non sono portatili)
  - GUI (WinForms)
  - Applicazioni Web (WebForms e ASP.NET)
  - Accesso ai DB (ADO.NET)
  - Transazioni (MTS)
  - Scripting (VBScript e Jscript)
  - Web Services

.NET e i linguaggi

- l'architettura .NET, a differenza di J2EE, non è legata ad un linguaggio specifico: è **multilinguaggio**
- si può scrivere un programma in un qualunque linguaggio supportato, compilarlo in MSIL ed eseguirlo su di una piattaforma .NET
- oggetti scritti in un linguaggio possono essere compilati e riutilizzati come moduli in programmi scritti in un altro linguaggio
  - esistono varie limitazioni:
    - codice unmanaged
    - dipendenza da librerie specifiche di Windows
    - ....

Common Type System (CTS)

- per permettere di utilizzare diversi linguaggi bisogna stabilire un massimo comune divisore tra tutti:
  - **Common Type System (CTS)**
- specifica dettagli come:
  - i tipi primitivi
  - cosa sono le classi (o nel gergo usato in .NET “tipi”) e le loro caratteristiche:
    - metodi
    - campi dati
    - proprietà
  - politiche di visibilità
  - ereditarietà
    - la multi-ereditarietà non è prevista...
    - si possono usare le interfacce

Linguaggi supportati

Microsoft supporta direttamente:

- C# (un nuovo linguaggio...)
- C
- C++
- Visual-Basic
- JScript

Terze parti hanno scritto compilatori .NET per:

- APL, CAML, Cobol
- Haskell, Mercury
- ML, Oberon, Oz
- Pascal, Perl, Python
- Scheme, Smaltak
- ... e forse Java!

Sono utili tutti questi linguaggi?

- PRO:
  - sicuramente permettono di “riciclare” competenze acquisite negli anni da parte dei programmatori
- CONTRO:
  - parecchi compromessi e limitazioni (esempio: tutto case-insensitive)
  - hanno snaturato del tutto alcuni linguaggi:
    - C++.... che ERA uno standard (addio multi-ereditarietà)!
    - Visual-Basic... praticamente si tratta di un altro linguaggio... meglio passare a C#!
- anche su Java si possono utilizzare altri linguaggi per produrre il bytecode ma non è una funzionalità basilare come viene presentata su .NET
- Java è un unico linguaggio che si può utilizzare in vari ambiti
  - applicazioni desktop, applicazioni server, web (applet)
  - desktop, server, dispositivi embedded

Il linguaggio C#

Java:JVM = C#:CLR

- si tratta del linguaggio principe dell'architettura .NET
- linguaggio ad oggetti
- derivato dal C++ (più semplice)
- è farcito da parecchio “zucchero sintattico”
- le specifiche riguardano semplicemente la grammatica del linguaggio
- non ha una libreria propria ma si appoggia sulla BCL di .NET
- .NET è indipendente da C#, ma C# non serve a nulla senza .NET

Standardizzazione di .NET

- .NET è una tecnologia interamente sviluppata da Microsoft
- la ECMA ha standardizzato il **Common Language Infrastructure** (**CLI**) che:
  - può essere considerato un “sottoinsieme” delle specifiche del CLR
  - comprende solo un insieme minimo (core) delle librerie di sistema
- sembra che altre parti siano in via di standardizzazione
- .NET è uno standard aperto?!
  - No!
  - .NET è una implementazione di alcuni standard aperti
  - molte parti dell'infrastruttura (come la maggior parte delle BCL) sono del tutto proprietarie e non standardizzate.

Conclusioni

- .NET è un progetto con grandi ambizioni
- Microsoft utilizzerà la propria posizione monopolista per imporre tale tecnologia sul mercato
- le similitudini con Java non possono passare inosservate
  - ci sono delle differenze
  - ma a volte poco sostanziali e prevalenti
- si nutrono fortissimi dubbi sulla portabilità della applicazione .NET su piattaforme diverse da quelle Windows (nonostante progetti coraggiosi come Mono su ambienti Linux/Unix...)
- staremo a vedere se il professionisti del settore alla fine rigetteranno o accetteranno questi profondi stravolgimenti