Introduzione a .NET

Raffaele Cappelli
raffaele.cappelli@unibo.it

.NET Framework

- .NET Framework
  - Architettura
- Common Language Runtime (CLR)
  - Compilazione ed esecuzione del codice
  - Assembly
- Class Library
  - Classi di base
  - Windows Forms
- Linguaggi per .NET

.NET Framework – Che cos'è?

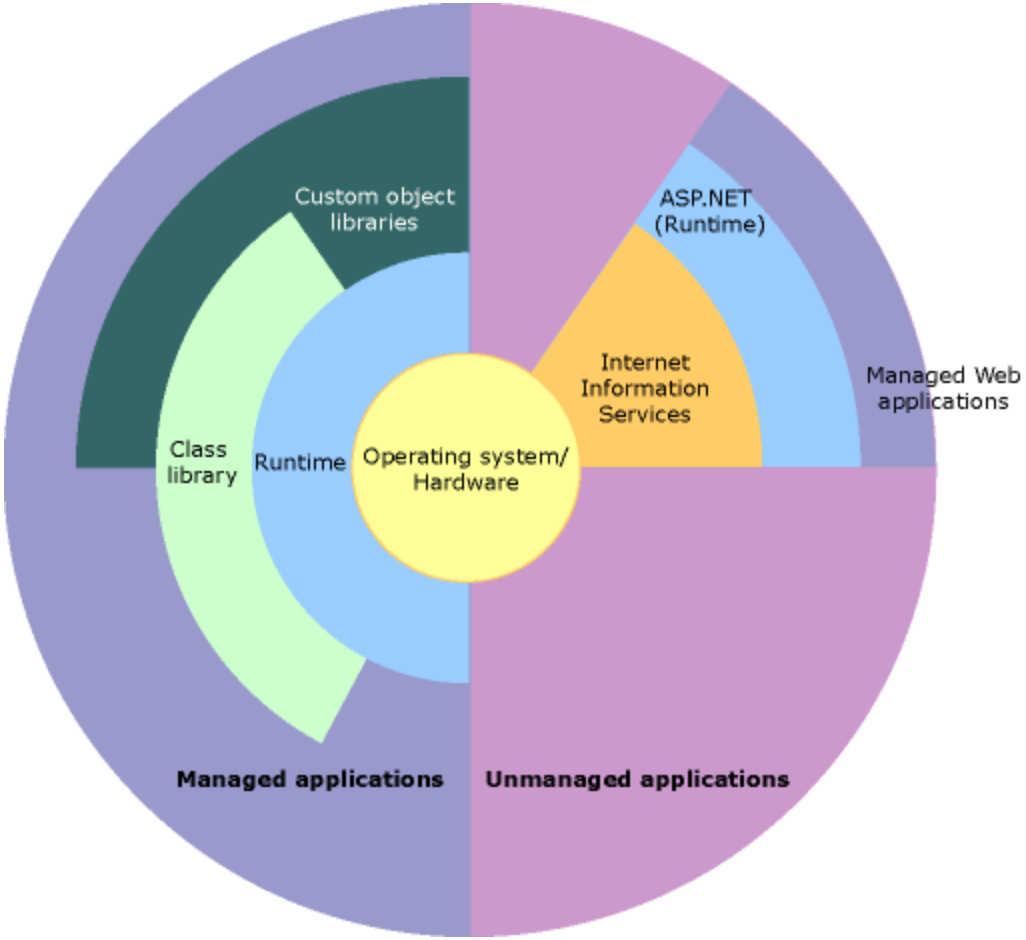
- Un componente di Windows che permette di sviluppare, eseguire e distribuire applicazioni e servizi web.
- Obiettivi:
  - Fornire un unico ambiente di sviluppo object-oriented sia per applicazioni eseguite localmente che in remoto
  - Mettere a disposizione un ambiente di esecuzione dei programmi che riduca problematiche di deployment e conflitti fra versioni diverse
  - Aumentare la sicurezza e affidabilità del codice
  - Fornire agli sviluppatori strumenti analoghi in applicazioni Windows, Web, Windows Phone.

.NET Framework – Struttura

- Si compone di due elementi principali: CLR e Class Library.
- Common Language Runtime (CLR)
  - Si occupa dell'esecuzione dei programmi
  - Fornisce servizi base quali gestione della memoria e degli thread
  - È responsabile della sicurezza e affidabilità dei programmi
  - I programmi eseguiti dal CLR sono detti "managed applications"
- Class Library
  - Una vasta collezione, gerarchica ed estendibile, di classi
  - Sia funzionalità di base (file, stringhe, strutture dati, accesso a database), che per specifiche tipologie di applicazioni (Console applications, Windows GUI applications, Web services, ...)

.NET Framework – Esecuzione delle applicazioni

- Managed applications: programmi eseguiti dal CLR
- Unmanaged applications: applicazioni “tradizionali”
- Applicazioni unmanaged (esempio un DBMS, o un web server) possono “ospitare” al loro interno il .NET Framework, chiedendo al CLR di eseguire “componenti managed”



CLR e CLI: non solo Windows

- CLR è l'implementazione Microsoft di CLI (Common Language Infrastructure)
  - CLI è uno standard ISO (ISO/IEC 23271:2003)
    - "ISO/IEC 23271:2003 defines the Common Language Infrastructure (CLI) in which applications written in multiple high-level languages may be executed in different system environments without the need to rewrite the applications to take into consideration the unique characteristics of those environments." [http://www.iso.org]
  - Esistono già altre implementazioni di CLI:
    - SSCLI (Shared Source Common Language Infrastructure): disponibile per Windows, FreeBSD e Macintosh
    - .NET Compact Framework: per dispositivi PocketPC, SmartPhone, ...
    - Mono: implementazione Open Source per Linux
    - ...

CLR – Terminologia

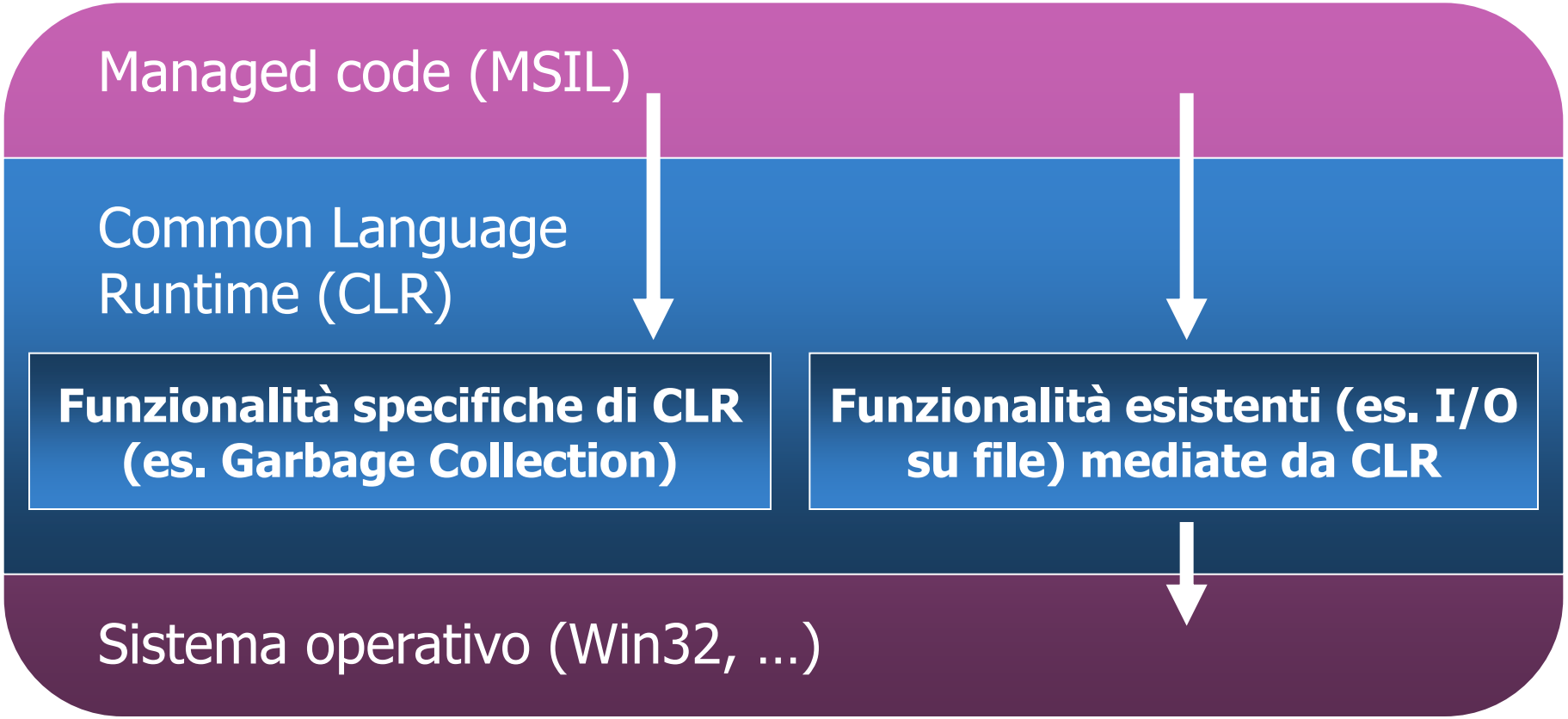
- CTS - Common Type System
  - Sistema di tipi unificato e inter-linguaggio
  - Due categorie di tipi (Value Type e Reference Type)
- CLS - Common Language Specification
  - Uno standard a cui qualsiasi linguaggio per .NET deve aderire; prevede un sottoinsieme minimo del CTS (utile per garantire interoperabilità fra linguaggi differenti)
  - In questo modo tutti i linguaggi .NET possono beneficiare del Class Library
- CIL - Common Intermediate Language (MSIL nell'implementazione Microsoft)
  - Un linguaggio indipendente dalla CPU che può essere efficientemente tradotto nel linguaggio macchina di una data CPU
- JIT- Just In Time Compiler
  - Non tutto il codice CIL di un programma viene sempre eseguito: solo la parte necessaria viene compilata un istante prima della sua esecuzione
  - Il codice compilato viene memorizzato per successive esecuzioni
- VES – Virtual Execution System
  - L'ambiente di esecuzione (macchina virtuale)

CLR – Terminologia (2)

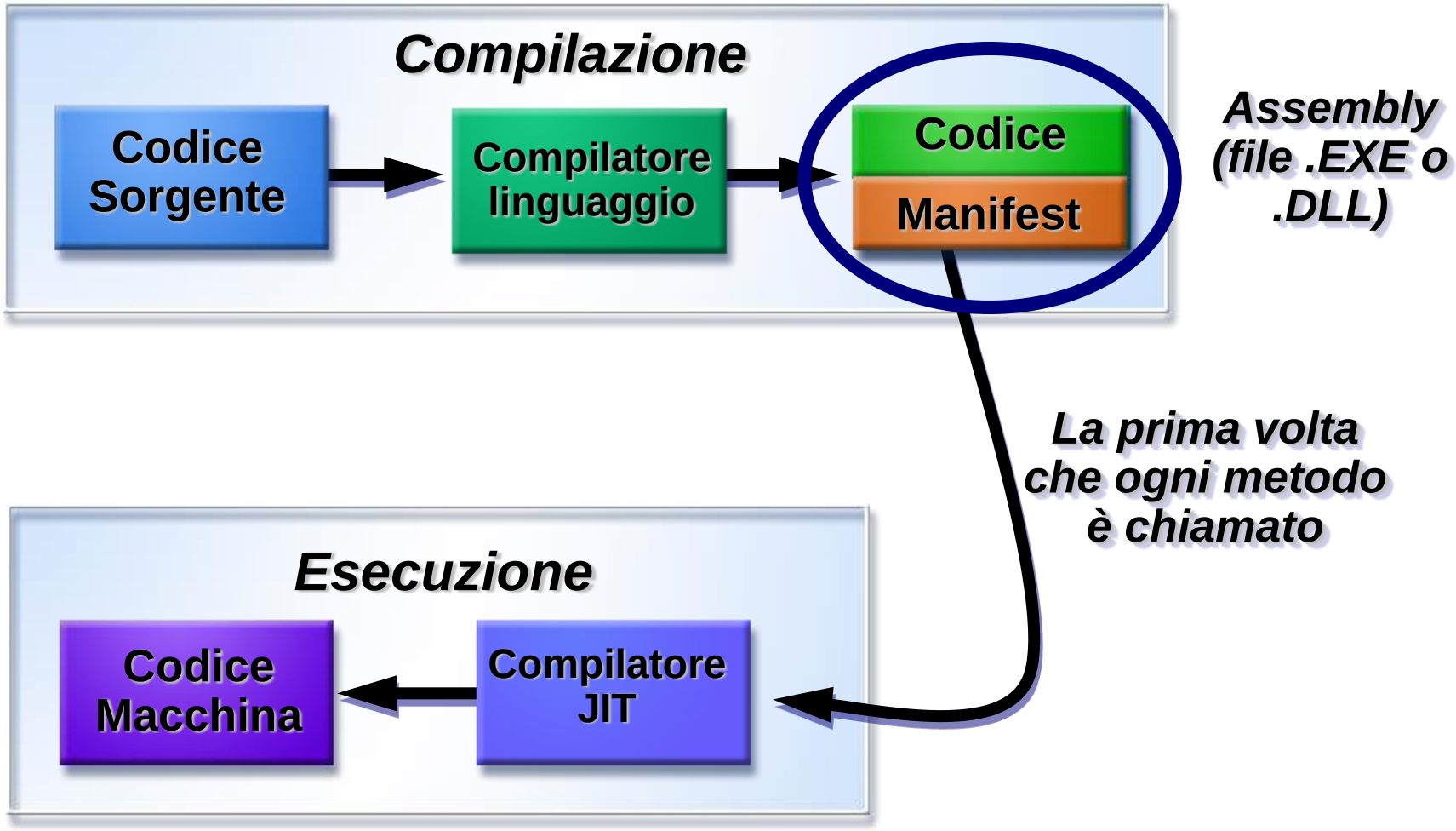
- Assembly
  - Insieme di funzionalità sviluppate e distribuite come una singola unità applicativa, composta da uno o più file
  - Completamente auto-descrittivo grazie al suo *manifest*
- Manifest
  - Stabilisce l'identità dell'assembly in termini di nome, versione, livello di condivisione tra applicazioni diverse, firma digitale, ...
  - Definisce quali file costituiscono l'implementazione dell'assembly
  - Specifica le dipendenze in fase di compilazione da altri assembly
  - ...
- Application Domain
  - Unità di elaborazione .NET (un assembly deve essere caricato in un Application Domain per poter essere eseguito)
  - Più "leggero" di un processo (più Application Domain possono risiedere nello stesso processo, ma vi sono meccanismi di sicurezza e isolamento)

CLR – Esecuzione managed applications

- Le managed applications sono scritte in MSIL, che il CLR è in grado di eseguire, offrendo vari servizi



CLR, codice MSIL e compilatore JIT



Un esempio: Sorgente – MSIL – ASM

```
class Program
{
    static void Main(string[] args)
    {
        Console.WriteLine("Hello World!");
    }
}
```

Sorgente (C#)

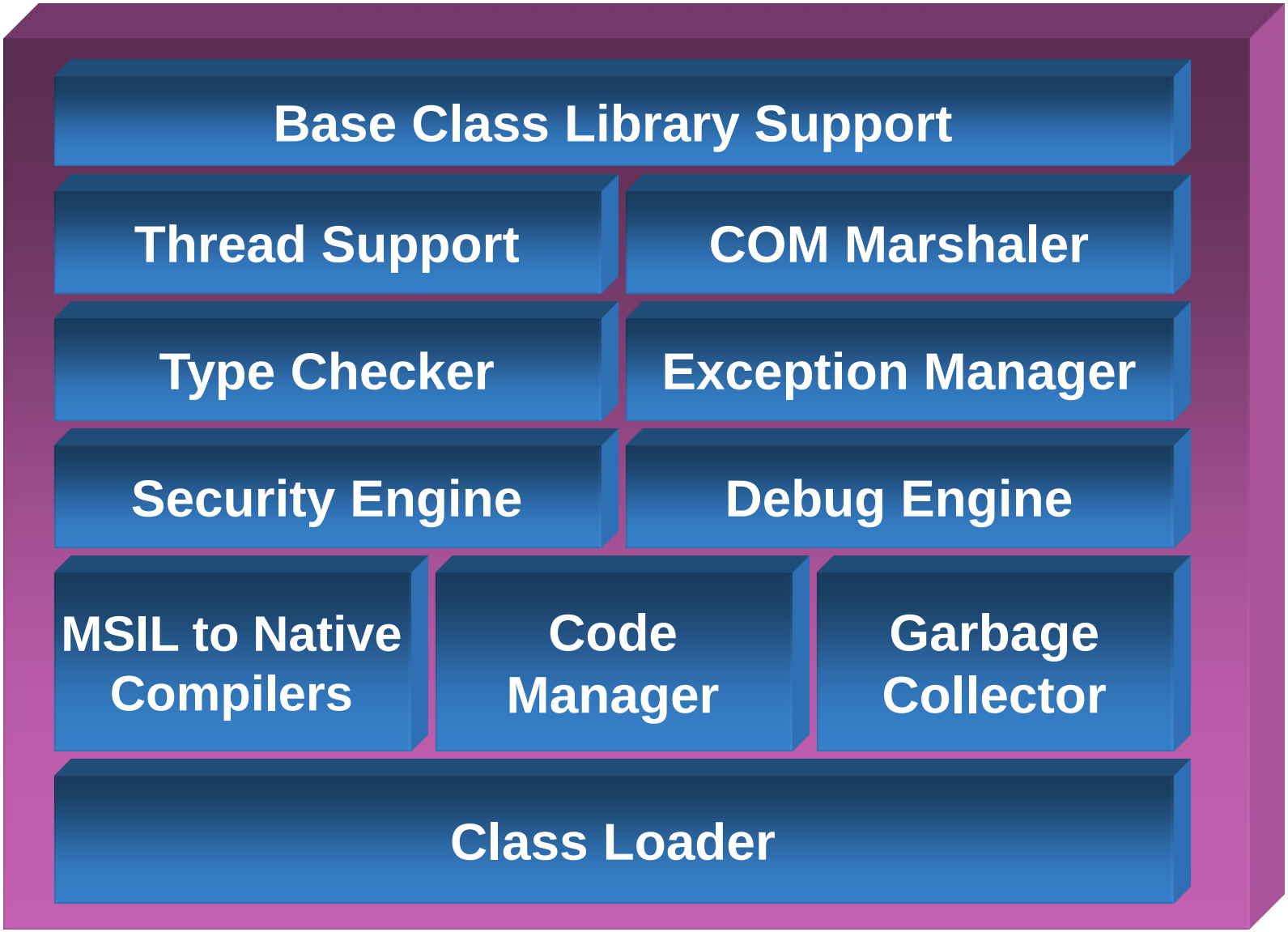
```
.method private hidebysig static void  Main(string[] args) cil managed
{
  .entrypoint
  // Code size       13 (0xd)
  .maxstack  8
  IL_0000:  nop
  IL_0001:  ldstr      "Hello World!"
  IL_0006:  call       void [mscorlib]System.Console::WriteLine(string)
  IL_000b:  nop
  IL_000c:  ret
} // end of method Program::Main
```

MSIL

```
00000000 55              push    ebp
00000001 8B EC           mov     ebp,esp
00000003 57              push    edi
00000004 56              push    esi
00000005 53              push    ebx
00000006 83 EC 30        sub     esp,30h
00000009 33 C0           xor     eax,eax
0000000b 89 45 F0        mov     dword ptr [ebp-10h],eax
0000000e 33 C0           xor     eax,eax
00000010 89 45 E4        mov     dword ptr [ebp-1Ch],eax
00000013 89 4D C4        mov     dword ptr [ebp-3Ch],ecx
00000016 83 3D 80 88 A1 00 00 cmp        dword ptr ds:[00A18880h],0
0000001d 74 05           je      00000024
0000001f E8 22 11 34 79  call    79341146
00000024 90              nop
00000025 8B 0D 7C 30 37 02 mov    ecx,dword ptr ds:[0237307Ch]
0000002b E8 FC 7E 66 78  call    78667F2C
00000030 90              nop
00000031 90              nop
00000032 8D 65 F4        lea     esp,[ebp-0Ch]
00000035 5B              pop     ebx
00000036 5E              pop     esi
00000037 5F              pop     edi
00000038 5D              pop     ebp
00000039 C3              ret
```

Codice Macchina e ASM (Intel 80386)

CLR – Struttura

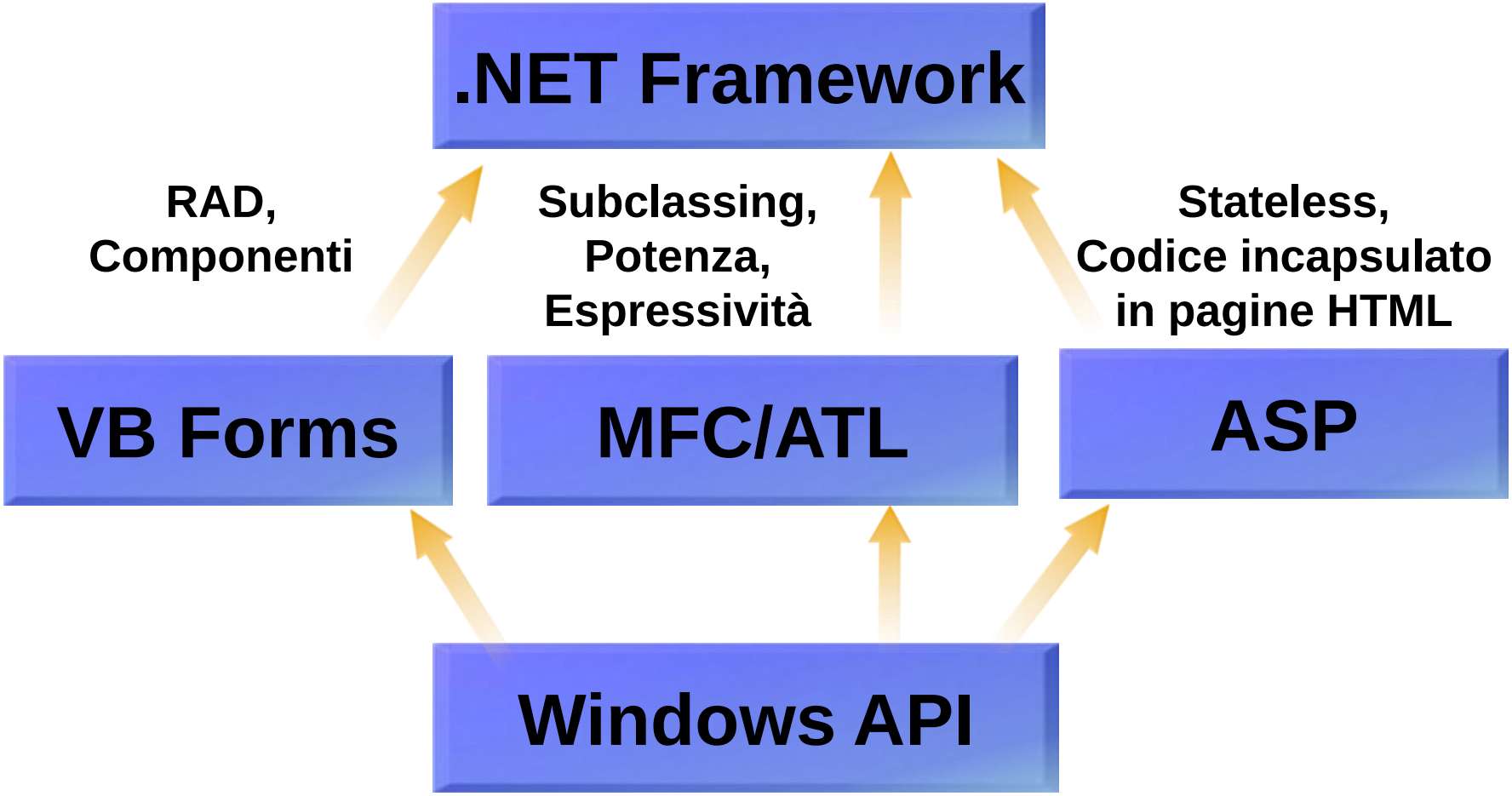


CLR – Vantaggi

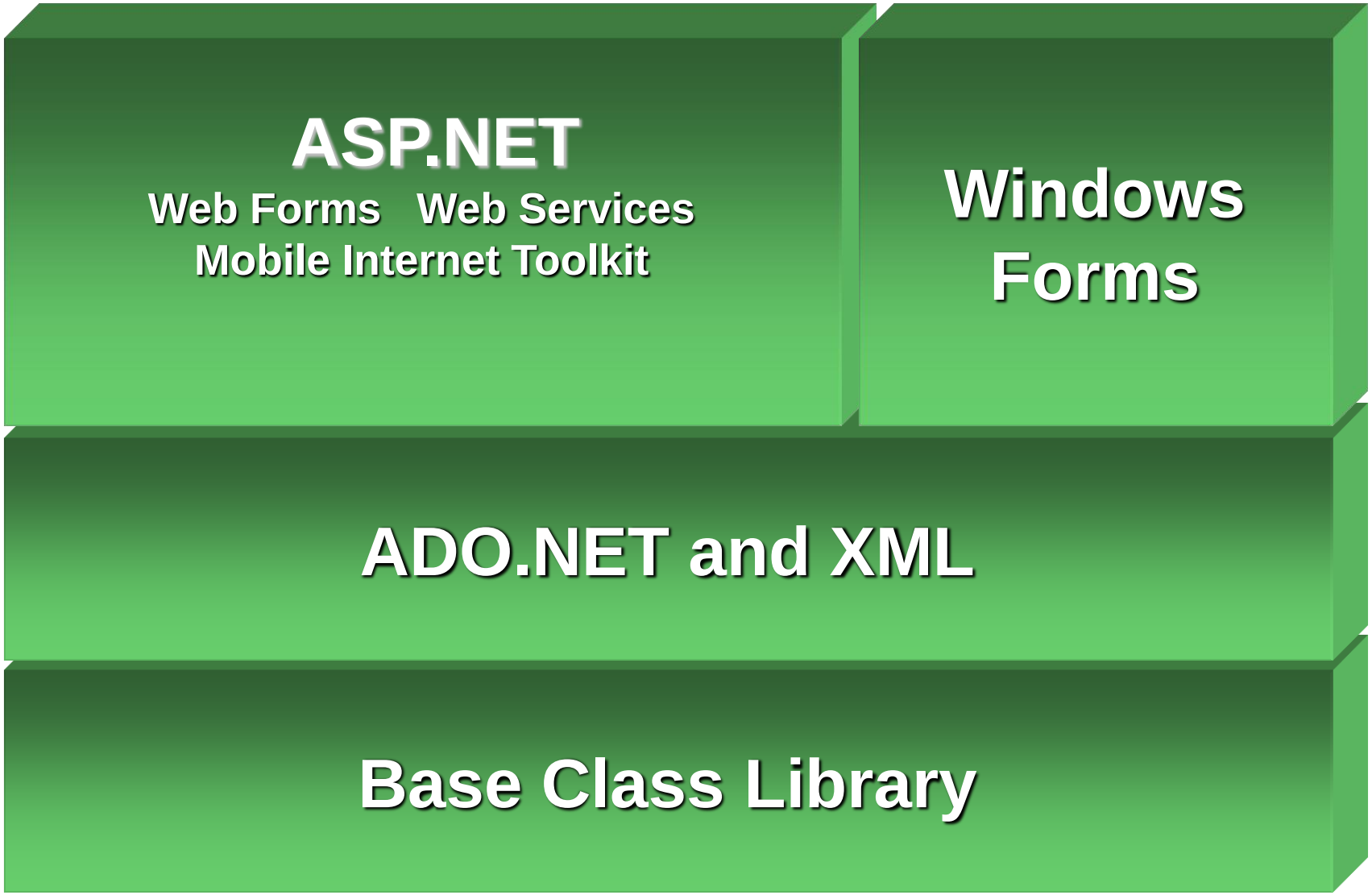
- Ambiente object-oriented
  - Qualsiasi entità è un oggetto
  - Classi ed ereditarietà pienamente supportati
    - Anche tra linguaggi diversi
- Riduzione errori comuni di programmazione
  - Linguaggi fortemente tipizzati
  - Gestione eccezioni
  - Prevenzione dei memory leak: Garbage Collection
- Indipendenza dal sistema operativo
  - Senza perdere troppa efficienza grazie al JIT che può ottimizzare il codice per la specifica piattaforma
- Piattaforma multi-linguaggio
  - I componenti di un'applicazione possono essere scritti con linguaggi diversi

.NET Framework Class Library

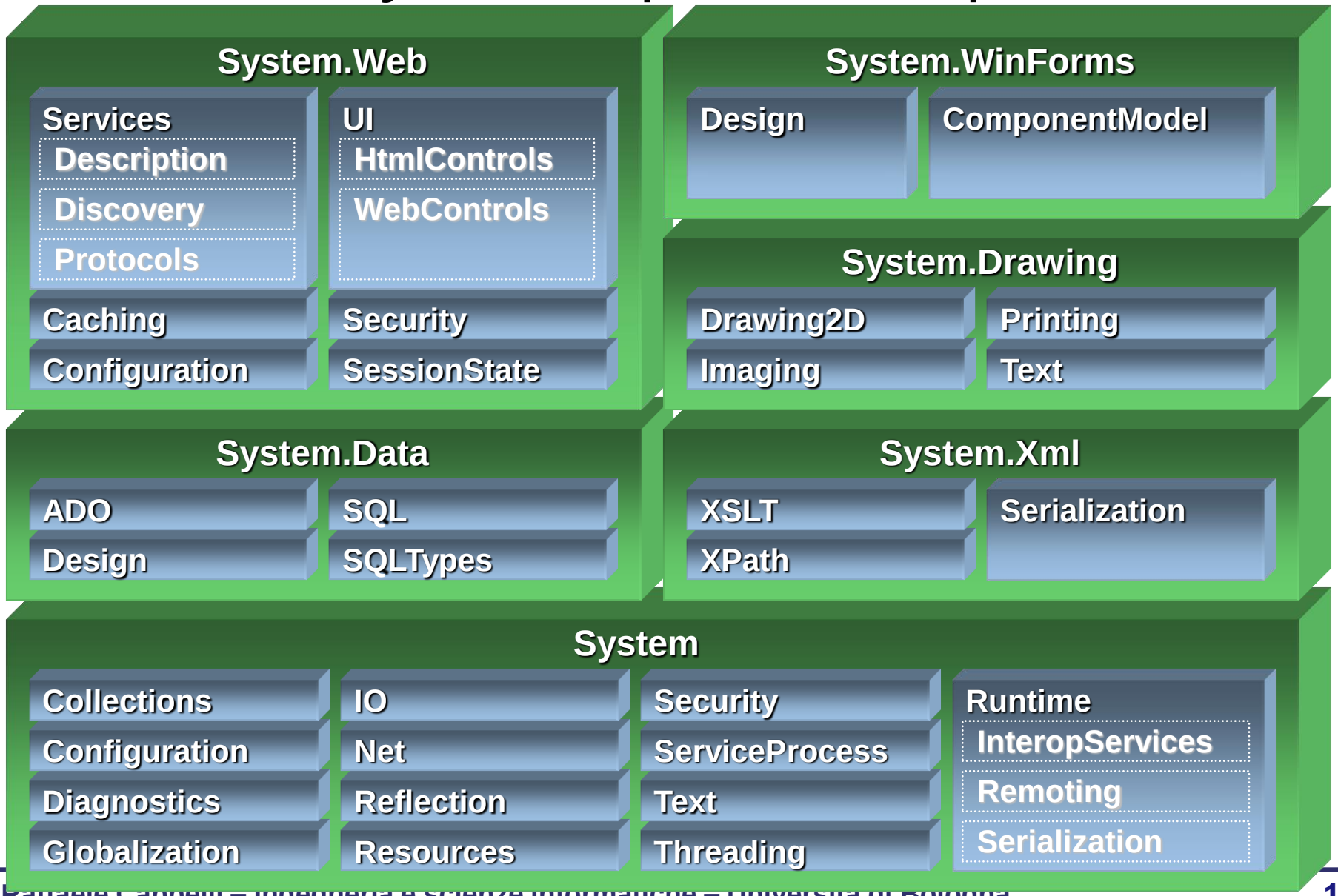
La Class Library è indipendente
dal linguaggio e dal modello di programmazione



Class Library



Class Library – Principali “namespace”

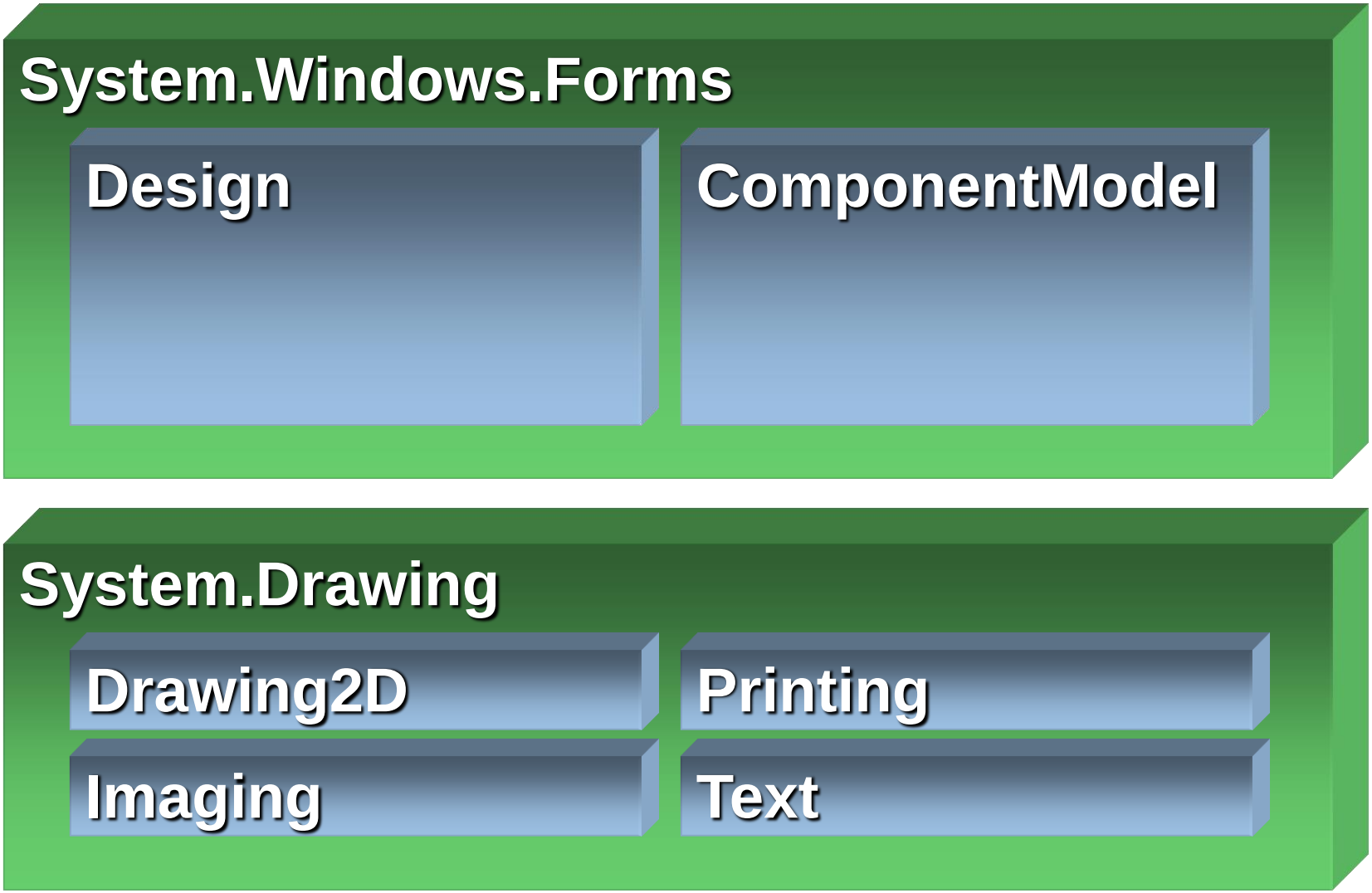


System namespace

Class Library – Classi di base

- Tipi di dati, conversioni, formattazione
- Strutture dati: Array, Liste, Hash, ...
- I/O: file di testo e binari, compressione, ...
- Rete: HTTP, TCP/IP socket, ...
- Sicurezza: Permessi, crittografia, ...
- Testo: Codifiche, espressioni regolari, ...
- Supporto per la localizzazione (multi-lingua)
- ...

Class Library – Programmazione Windows



Windows Forms

- Classi per realizzare interfacce utente grafiche (GUI)
  - Coniugano la semplicità del Visual Basic con la potenza delle MFC
  - Basate su componenti ed eventi
  - Layout automatico dei controlli
  - Supporto grafico avanzato (GDI+)
  - Un insieme di controlli predefiniti molto ricco
  - Componenti per l'accesso a database
  - Supporto ActiveX
  - Supporto per la stampa
  - Unicode
  - ...

Linguaggi per .NET

- Qualsiasi linguaggio conforme al CLS
- Forniti da Microsoft
  - C++, C#, F#, VB.NET, JScript
- Forniti da terze parti
  - Perl, Python, Pascal, APL, COBOL, Eiffel, Haskell, ML, Oberon, Scheme, Smalltalk, ...

- Tutti i linguaggi .NET possono utilizzare la Class Library e le funzionalità del framework, ma il linguaggio “principe” è il C#!

Linguaggi per .NET – Esempi

Visual Basic .NET

```
Class HelloWorldApp
  Shared Sub Main()
     System.Console.WriteLine("Hello, world!")
  End Sub
End Class
```

C#

```
class HelloWorldApp
{ static void Main()
  {
     System.Console.WriteLine("Hello, world!");
  }
}
```

COBOL.NET

```
000330 IDENTIFICATION DIVISION.
000340 PROGRAM-ID. MAIN.
000350
000360 ENVIRONMENT DIVISION.
000370
000380 DATA DIVISION.
000390 WORKING-STORAGE SECTION.
000400
000410 PROCEDURE DIVISION.
000420     DISPLAY "Hello, World!"
000430 END PROGRAM MAIN.
```

.NET Framework – Risorse

- Siti web
  - http://msdn.microsoft.com/netframework
  - http://code.msdn.microsoft.com
  - http://windowsclient.net
  - http://www.mono-project.com
  - http://www.codeplex.com
  - http://dotnetkicks.com