



CMJ251-Manajemen Jaringan Mobile

www.esaunggul.ac.id

Dosen Pengampu :

5165-Kundang K Juman, Ir, MMSI
Prodi Teknik Informatika Fakultas Ilmu Komputer

Application Platforms Today

Browser
Apps

Web Services
Apps

Local
Apps

Other
Apps

GUI
Services

Transaction
Services

Web
Scripting

Data
Access

More

Standard Library

Runtime Environment

Operating System

What is Microsoft .Net?

▶ Visi:

- The platform for the **digital future**
- ‘.NET is Microsoft’s platform for a new computing model built around XML Web Services’

Microsoft Corporation Annual Report, 2001

▶ Suatu brand name

- Applied to many things

▶ Diperkenalkan oleh Microsoft pada (Juni 2000)

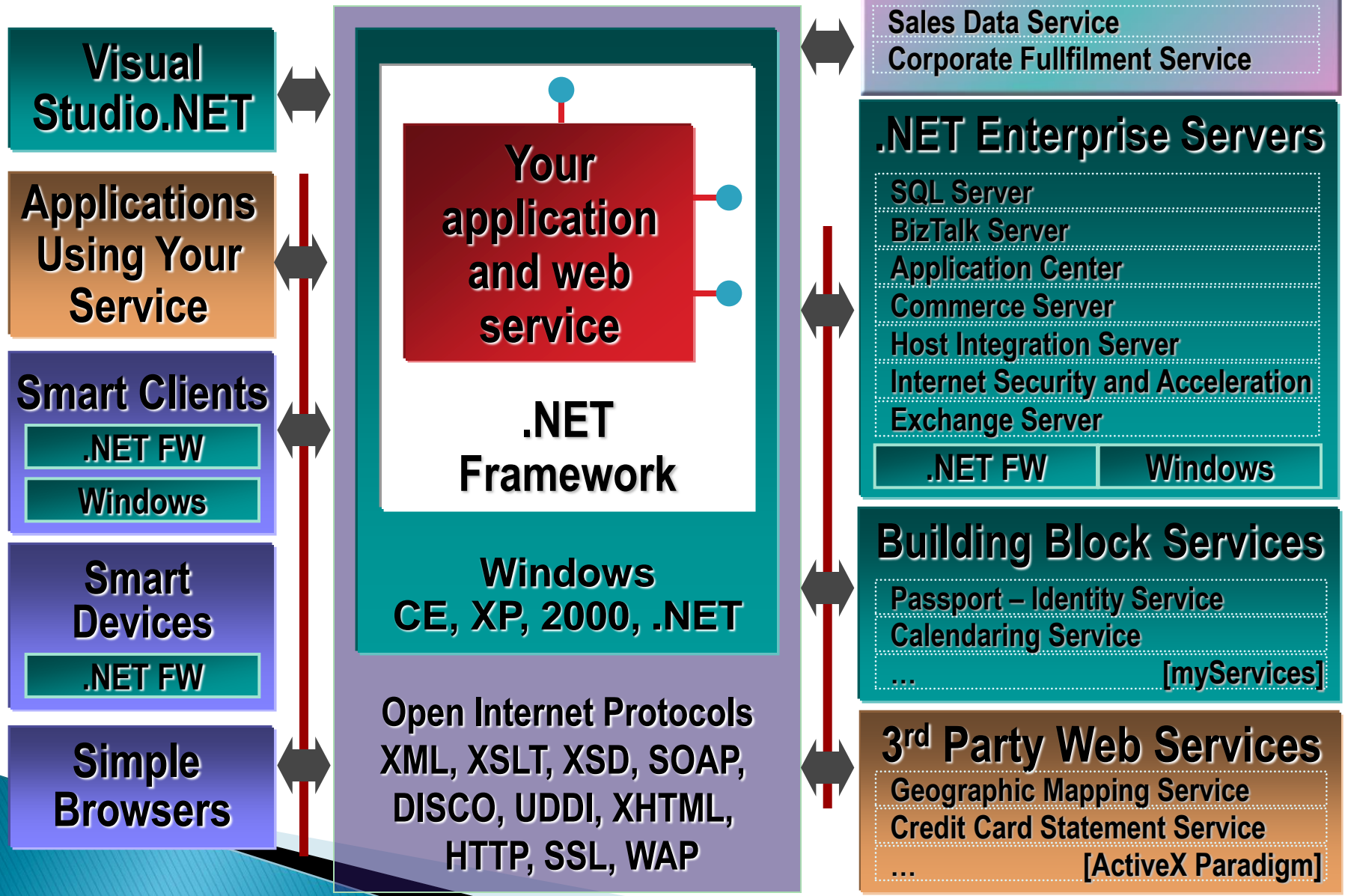
- Sekarang sudah sampai .net framework 3.5

.NET FRAMEWORK

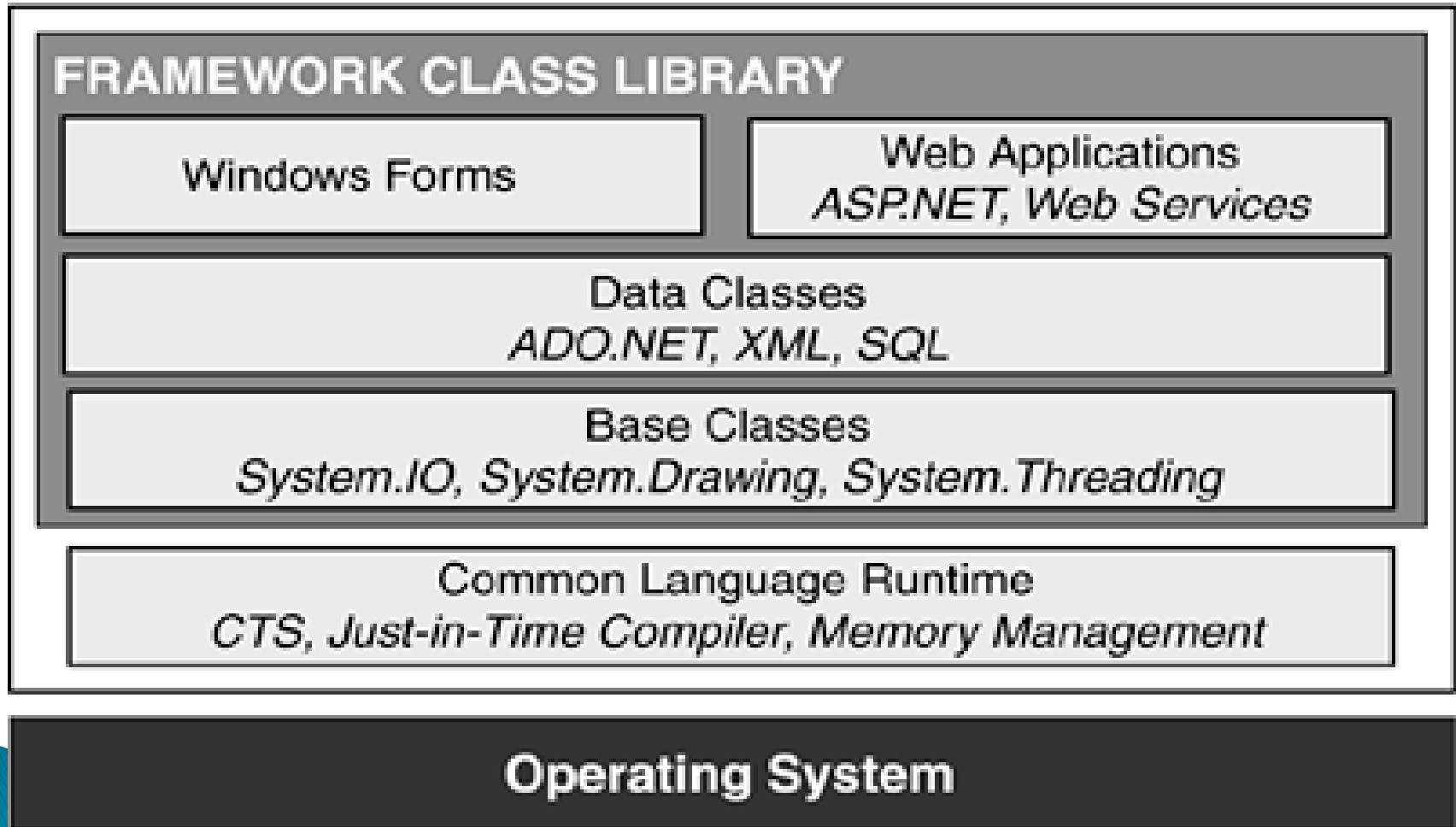


- ▶ Bersifat independen dalam hal bahasa
 - Aplikasi dapat dikembangkan di beberapa bahasa yang didukung oleh .NET: Visual Basic .NET, Visual C++ .NET, C# dan J#
 - Programmer dapat memilih salah satu bahasa yang paling dikuasainya.
- ▶ Mampu dijalankan di semua platform

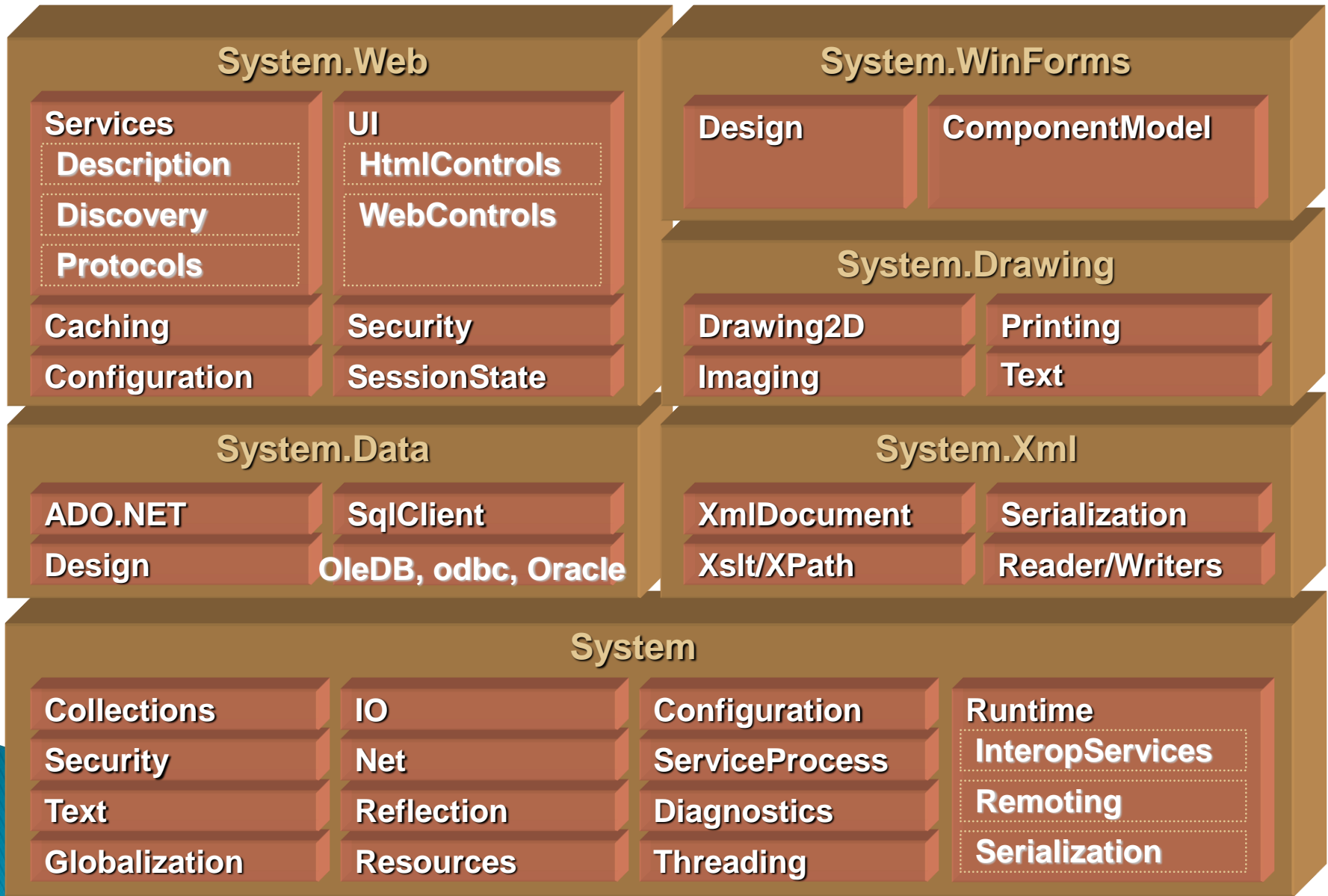
.NET Blueprint



.NET Framework (2)



.NET Framework (3)



Common Language Runtime

- ▶ Bagian utama dari framework yang akan mengkompilasi dan mengeksekusi program yang ditulis dengan bahasa yang mendukung .NET
- ▶ Compilation process
 - Dua Langkah Kompilasi
 - Pertama program di compile menjadi Microsoft Intermediate Language (MSIL) / Common Intermediate Language (CIL)
 - Mendefinisikan instruksi untuk CLR
 - Kemudian MSIL code diterjemahkan kedalam machine code
 - Machine code untuk platform yang khusus
 - Mengapa dua proses kompilasi ?
 - Agar platformnya bersifat independence
 - .NET Framework dapat diinstal di platform berbeda
 - Mengeksekusi program .NET tanpa modifikasi kode-nya
 - Language independence
 - .NET program tidak terikat dengan bahasa tertentu
 - Komponen yang lama dan baru dapat terintegrasi

Common Language Runtime (2)

- ▶ Keuntungan lain dari CLR
 - Execution-management features
 - Manages memory, security
 - Membebaskan programmer dari banyak tanggung jawab
 - Programmer lebih berkonsentrasi pada program logic
- ▶ CLR menyediakan JIT compilation.
- ▶ JIT hanya akan mengkompilasi method-method yang memang digunakan dalam **suatu bagian** aplikasi pada saat tertentu, dan hasil kompilasi ini sendiri di *cache* di dalam mesin dan akan dikompilasi kembali jika memang ada perubahan pada kode aplikasi kita.
 - Jadi tidak semua method diload ke-memori pada saat yang sama

Common Language Runtime (3)

Base Class Library Support

Thread Support

COM Marshaler

Type Checker

Exception Manager

Security Engine

Debug Engine

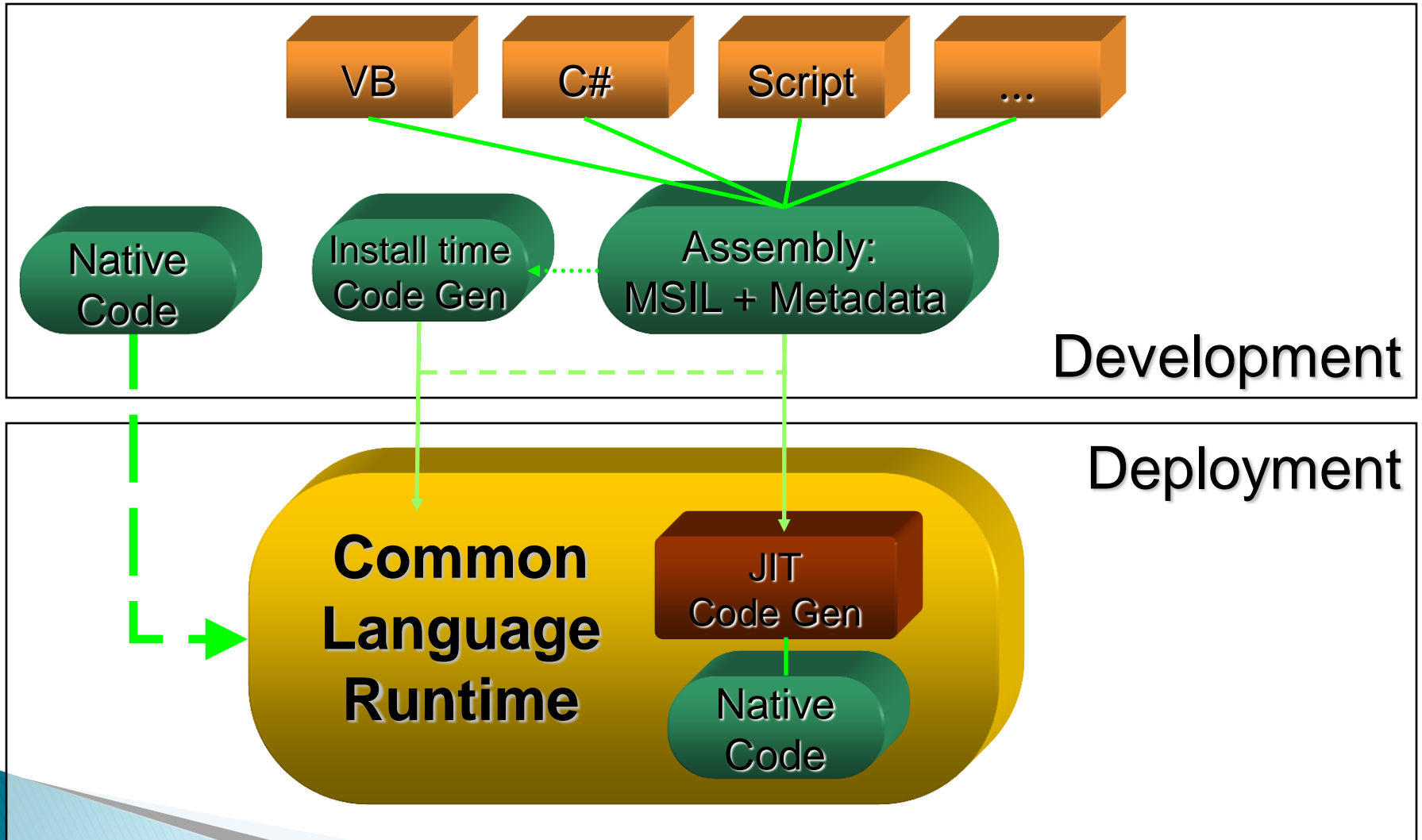
**IL to Native
Compilers**

**Code
Manager**

**Garbage
Collector**

Class Loader

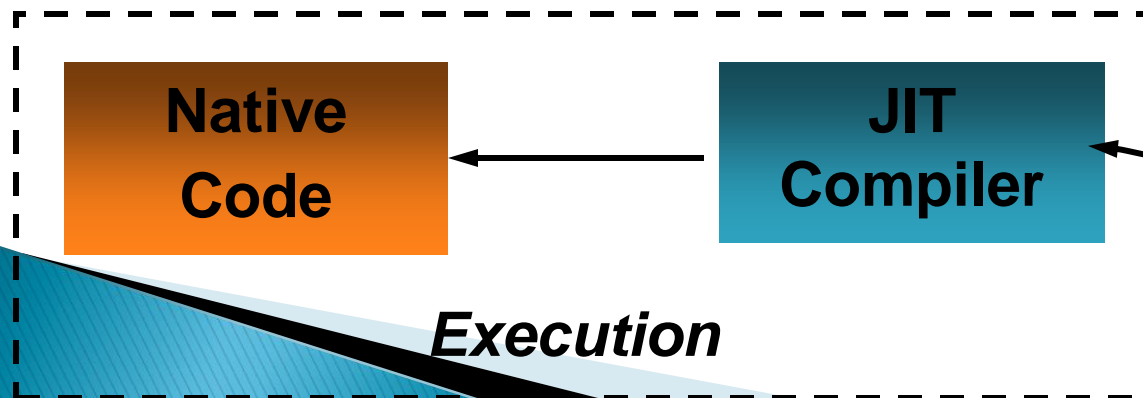
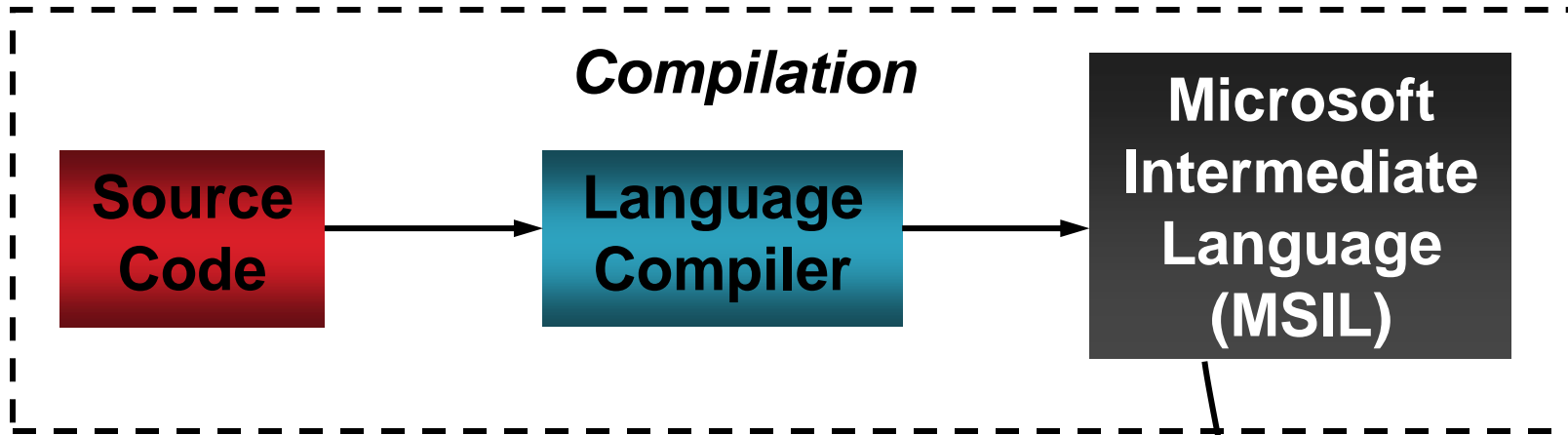
CLR Execution Model



Data Types in the CLR

- ▶ The CLR mendefinisikan Common Type System (CTS)
 - All languages built on the CLR use the CTS (managed code)
- ▶ Ada 2 kategori:
 - Value types:
 - Relatively simple types
 - Allocated on the stack
 - Reference types:
 - More complex types
 - Allocated on the heap
 - Destroyed through garbage collection

Compiling & Executing Managed Code



The first time each method is called

.NET Class Library

- ▶ Sering disebut sebagai Base Class Library.
- ▶ Adalah koleksi dari *reusable types* yang sangat terintegrasi secara melekat dengan CLR.
- ▶ *Class library* bersifat berorientasi objek yang akan menyediakan *types* dari fungsi-fungsi *managed code*.
- ▶ Dengan BCL kita dapat membuat:
 - Aplikasi *console*
 - Aplikasi berbasis windowd (*Windows Form*)
 - Aplikasi ASP.NET (berbasis web)
 - Aplikasi Web Services XML
- ▶ Aplikasi berbasis *Windows Services*

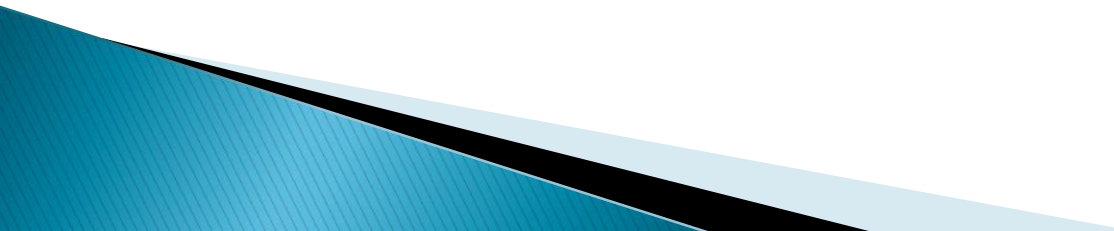
A Brief History of Windows Mobile Development

- ▶ Embedded Tools for Visual Studio
 - VS 6
 - Tools for VB, C++
- ▶ eMBedded Tools
 - Standalone and free
 - eVB
 - eC++
- ▶ .NET Compact Framework (First Pass)
 - Visual Studio.NET
 - Smart Device Extensions

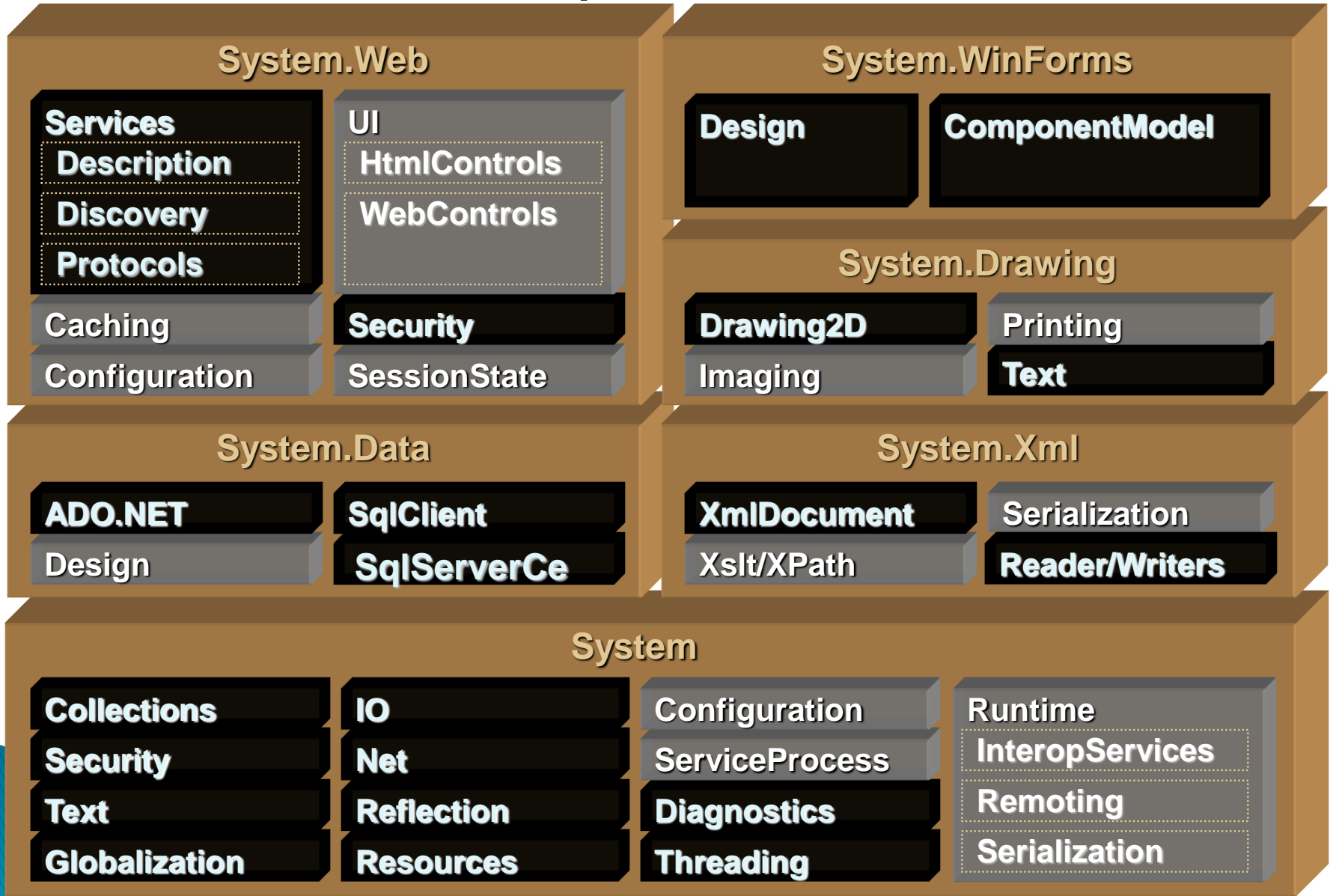
.NET Compact Framework Design Goals

- ▶ Portable and small .NET CLR for devices
 - Enable Visual Basic® and C#
- ▶ Integrated in Visual Studio® .NET
 - Run managed .EXEs and .DLLs directly
 - Debug with Visual Studio .NET
- ▶ Target devices:
 - Compaq, HP, Dell, Toshiba, ...
 - PocketPC 2000, 2002
 - Windows Mobile (alias PPC 2003)
 - .Net CF in ROM!
 - Windows CE
 - SmartPhone
 - No Palm/Handspring

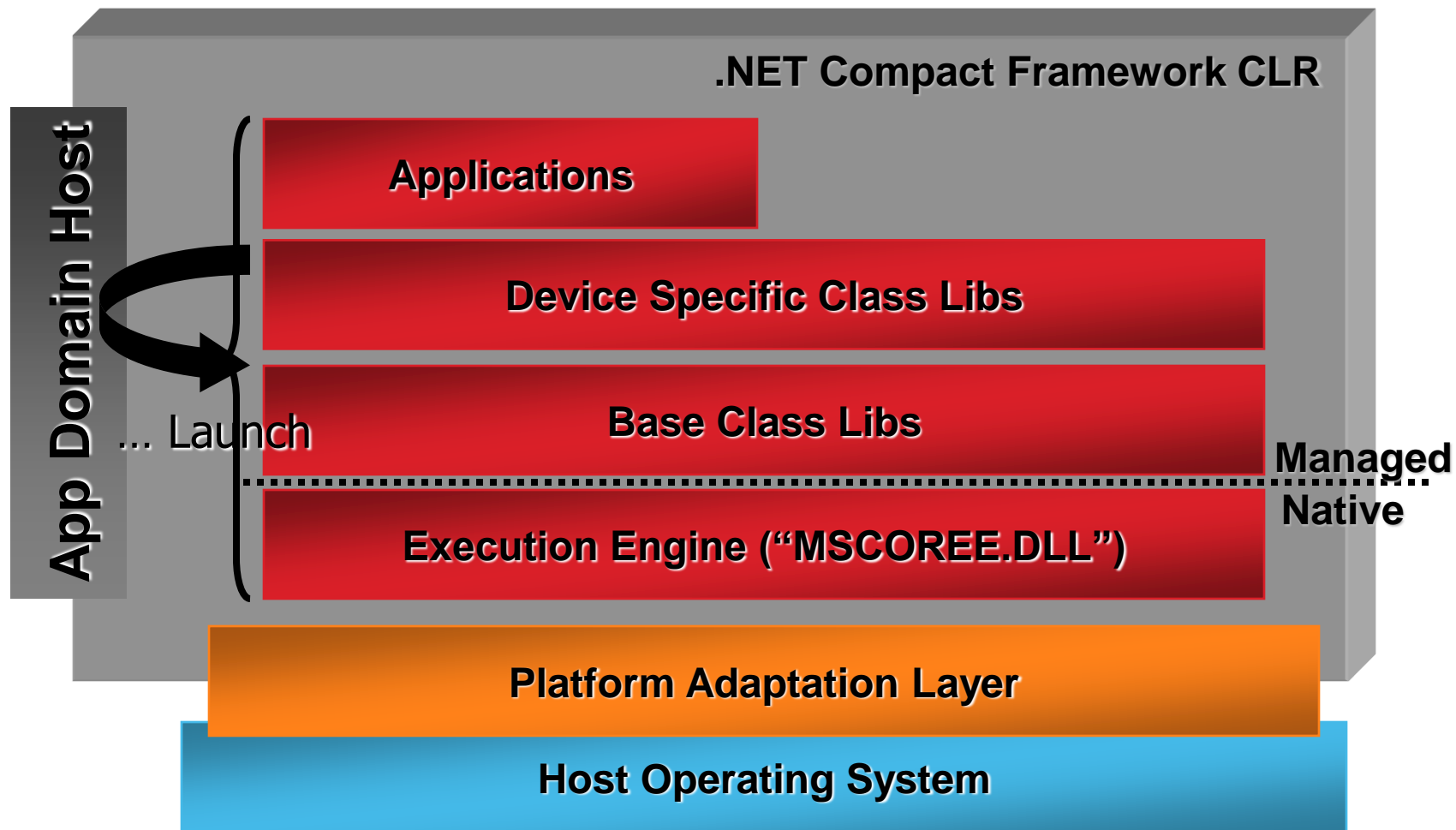
The Framework – “Optimized”

- ▶ 20mb → 1.5mb
 - ▶ Same languages, same code, same MSIL
 - ▶ Multiple target CPUs / Mobile device
- 

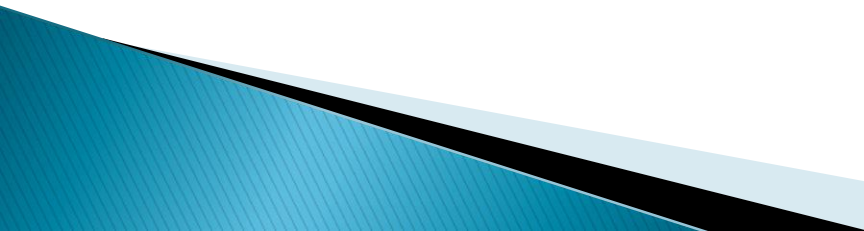
.NET Compact Framework



.NET CF Architecture



Platform Adaptation Layer

- ▶ **Base Services**
 - Threads, sync, timers, memory, math
 - ▶ **Storage**
 - Open, close, read/write, directory
 - Direct addressing mode
 - ▶ **Networking**
 - Sockets
 - Connection Control
 - ▶ **GUI**
 - Framebuffer, char and pointer I/O, Fonts
- 

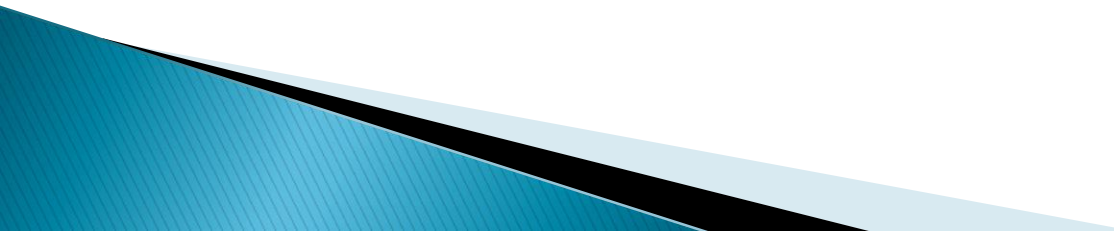
Execution Engine

- ▶ Basic services for managed execution
 - Loader
 - Meta-data engine/cache
 - Representation of class hierarchy/layout
 - Reflection
 - JIT-Compiler/Verifier
 - Execution Infrastructure
 - Exceptions, Native Code InterOp, Security
 - Garbage Collector
 - Debugging Support
 - “Native” implementation of some managed APIs (class libraries)

Class Libs

- ▶ Base Types
 - Type conversion, display formatting
 - String manipulation
 - Array.Sort
- ▶ Threading & Sync (System.Threading)
- ▶ Storage (System.IO)
- ▶ Resources (System.Resources)
- ▶ Collections (System.Collections)
- ▶ Reflection (System.Reflection)
- ▶ XML
 - DOM and Reader/Writer
- ▶ Networking, Forms, Data Access

Developing in a Compact World

- ▶ VS.Net 2003/2005 installation option
 - ▶ Only C#, VB.Net
 - ▶ True on-device debugging
 - If you can get it connected...
 - ▶ Emulator
 - ▶ Deployment
- 

Developing in a Compact World

VS.NET Integration

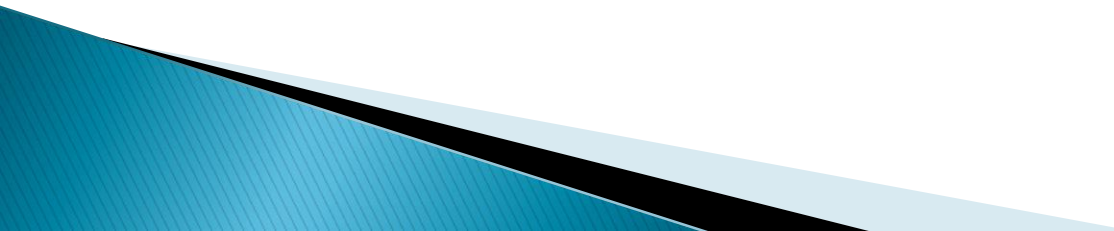
- ▶ Templates for devices in New Project dialog
 - Template sets device and project type
- ▶ Template types
 - Pocket PC Application
 - Pocket PC Class Library
 - Pocket PC Control Library
 - Windows CE .NET Application
 - Mobile Phone Application
- ▶ Default set of references that are appropriate for your platform

Develop Symbian in VS 2005

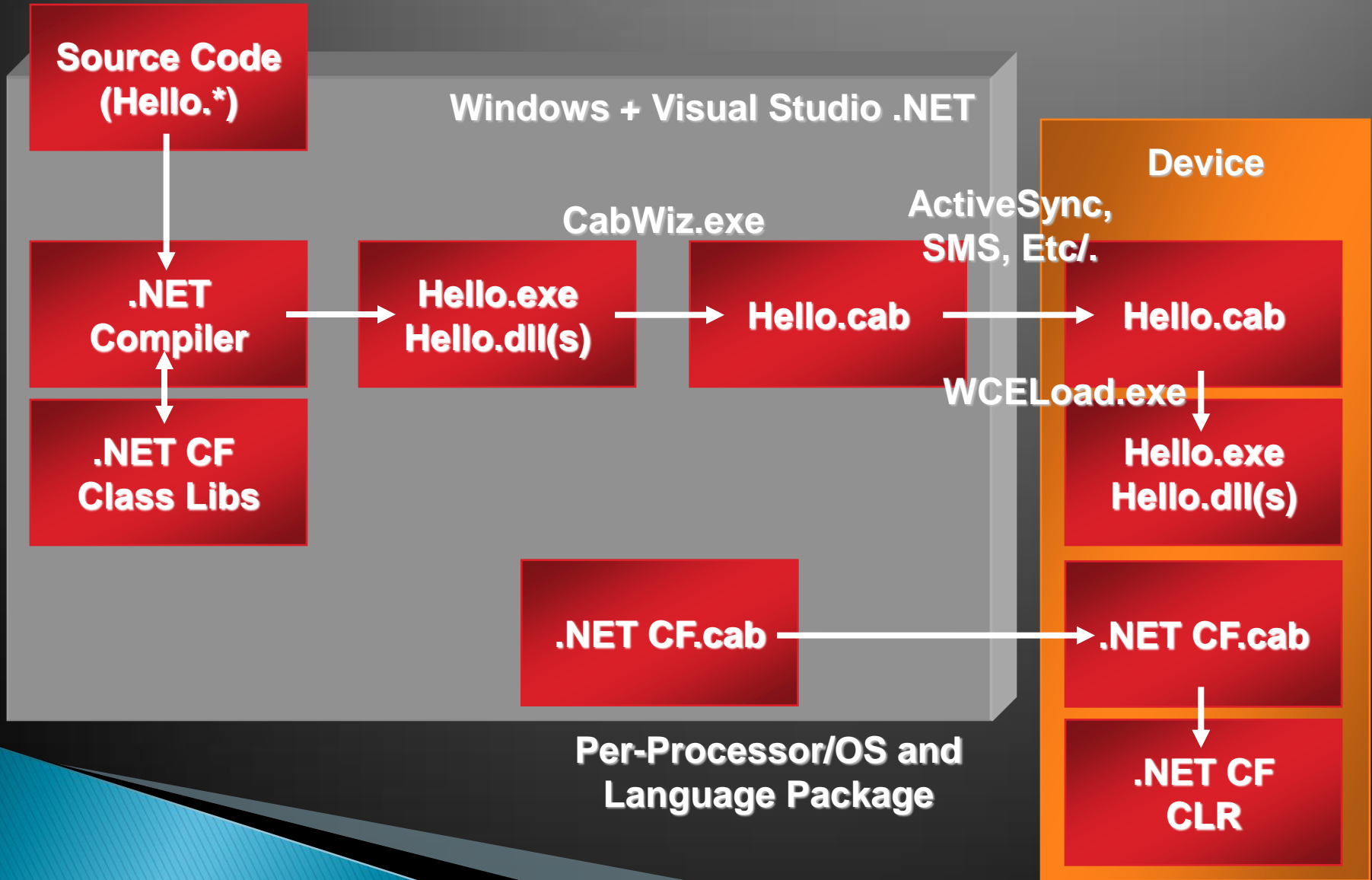
- ▶ You can get it! You can download **AppForge** plugin!

Developing in a Compact World

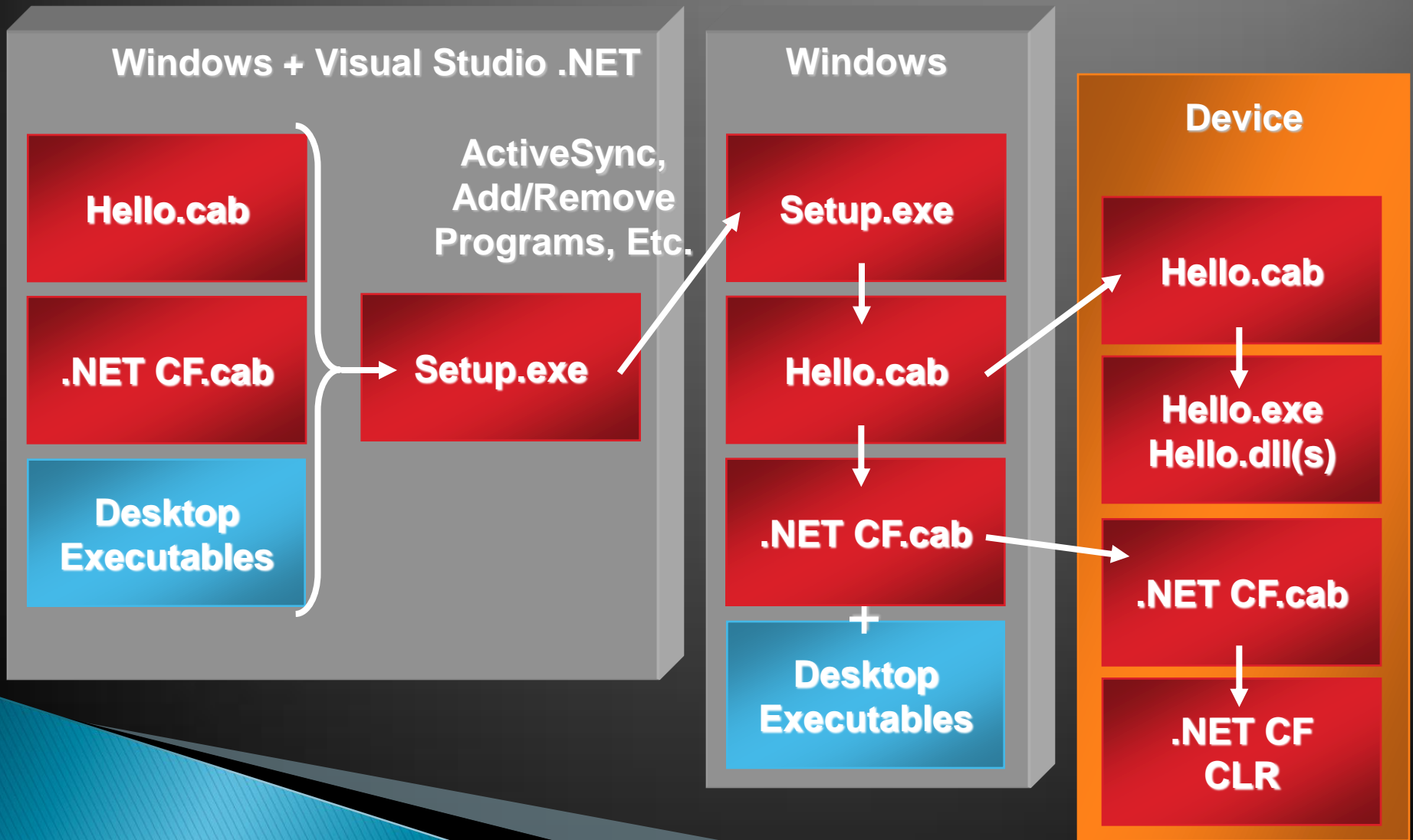
Deployment

- ▶ Same user experience as desktop
 - ▶ Compiles application for device
 - ▶ Output Window used for deployment messages
 - ▶ Visual Studio deploys the framework to device if needed
- 

Packaging and Deployment



More Deployment Options



Working with data: SQL Server CE

- ▶ Limited subset of SQL Server functionality
- ▶ Good for:
 - Persistent storage locally
 - Persistent storage for occasionally-connected applications and large sets of data
 - Merge Replication
- ▶ Currently, SQL Server CE requires an additional installation

Working with data: SQLite

- ▶ SQLite adalah database bukan client-server, dia bersifat stand-alone
- ▶ Hanya terdiri dari 1 file
 - Berekstensi: .db3
- ▶ Perlu download file DLL tambahan agar bisa koneksi:
 - System.Data.SQLite
- ▶ Secure, bisa menggunakan bahasa SQL
- ▶ Bersifat lokal

Working with data: XML

- ▶ Good for:
 - Persistent storage of smaller sets of data
 - Local storage
 - Occasionally-connected computing
 - Not as secure as SQL Server CE

How Compact ?

	<i>File Size</i>			<i>Classes</i>			<i>Methods</i>		
	<i>NETCF</i>	<i>Desktop</i>	<i>%</i>	<i>NETCF</i>	<i>Desktop</i>	<i>%</i>	<i>NETCF</i>	<i>Desktop</i>	<i>%</i>
"MSCorEE"	400K*	2.2 M	18%	N/A	N/A	N/A	N/A	N/A	N/A
MSCorLib	200K	2M	10%	364	1286	28%	3989	13817	29%
System	100K	1.2M	8%	140	765	18%	1090	6953	16%
System.Drawing	20K	458K	5%	41	254	16%	385	3509	11%
System.Web.Services	67K	503K	13%	54	274	20%	302	2083	14%
System.Windows.Forms	56+55K	2M	5%	43	823	5%	393	11337	3%
System.XML	138K	1.2M	12%	100	724	14%	927	7227	13%

* .NETCF file size of MScorEE is for Win32/x86 version

Framework Size

- ▶ Framework size (RAM or ROM)
 - ~1.5 MB
- ▶ Running RAM needs
 - 1 MB+ (depends on app)
- ▶ Typical application sizes
 - 5 – 100 KB
 - Apps often smaller due to use of platform features in the framework