# Microsoft .NET Framework

7th Brazilian Symposium on Programming Languages
Ouro Preto, MG, Brazil - May 28-30, 2003

Alisson Sol
Software Design Engineer Lead
Microsoft Corporation

# Acknowledgements

- Redmond teams
  - Microsoft Business Solutions
  - Information Worker – New Markets
  - Part I based on presentation by Brad Abrams
- Microsoft Brasil
  - University Relations
- Legal Warning
  - Opinions in this presentation do not necessarily represent those of Microsoft Corporation

# Summary

- Part I – .NET Framework Overview
  - Concepts
  - Architecture
  - Features
- Part II – Application Development
  - Samples
  - Tools
- Note
  - Some concepts from C# language presented on another lecture
  - Will not explain programming language details

# .NET Framework Overview

Microsoft .NET Framework - Part I

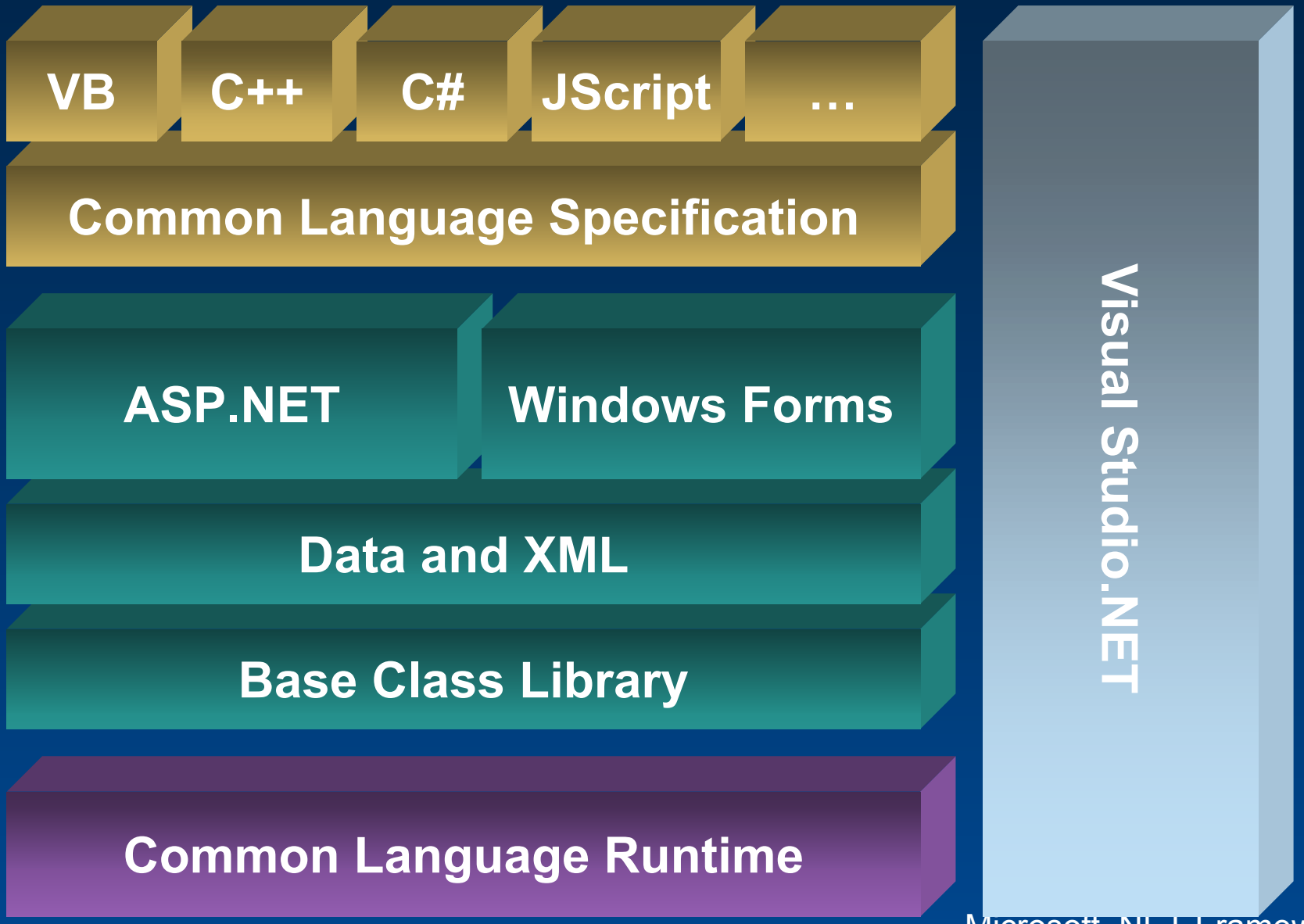
# Definition

- Old
  - .NET is the Microsoft platform for XML Web services.
- New
  - .NET is the set of Microsoft technologies for connecting your world of information, people, systems, and devices.
    .NET is infused into the Microsoft platform providing the ability to build, host, deploy and consume XML Web service connected solutions.

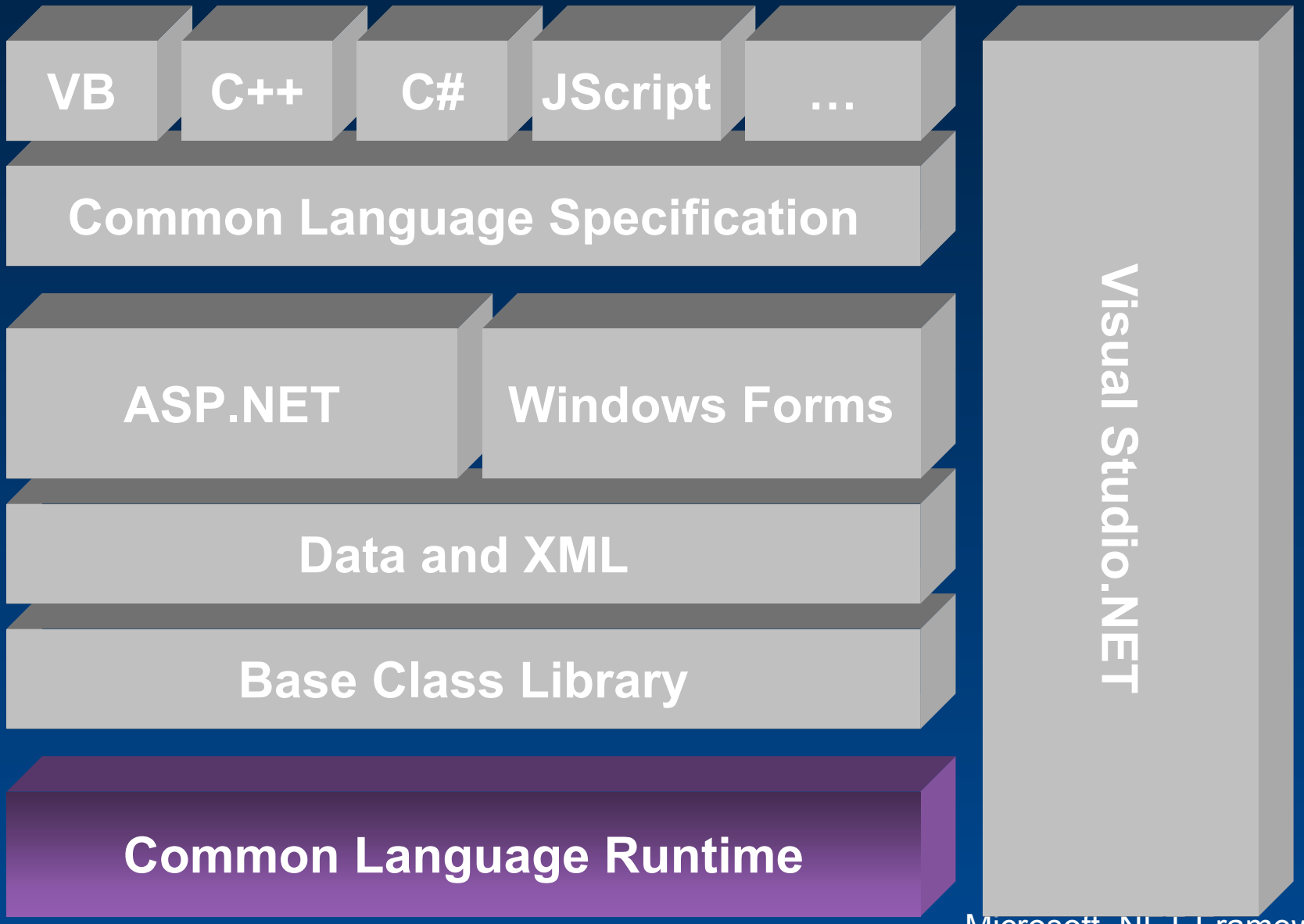
# Design Goals

- Simplified development
- Unified programming models
- Web standards and best practices
- Simple to deploy, run, & maintain

# Framework, Languages, And Tools



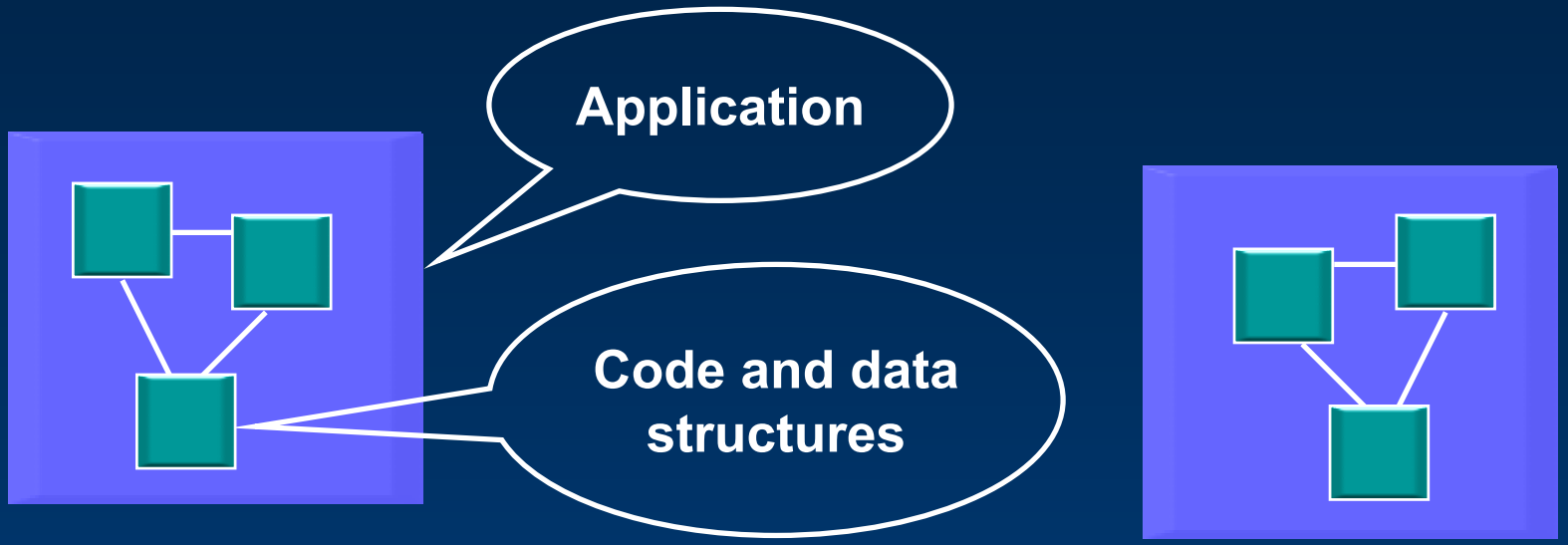
# Framework, Languages, And Tools

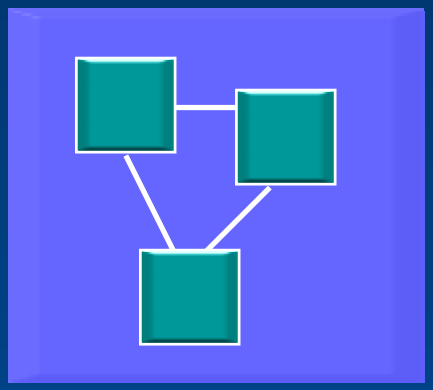


# CLR Design Goals

- Dramatically simplify application development
- Provide a robust and secure execution environment
- Support multiple programming languages
- Simplify deployment and management

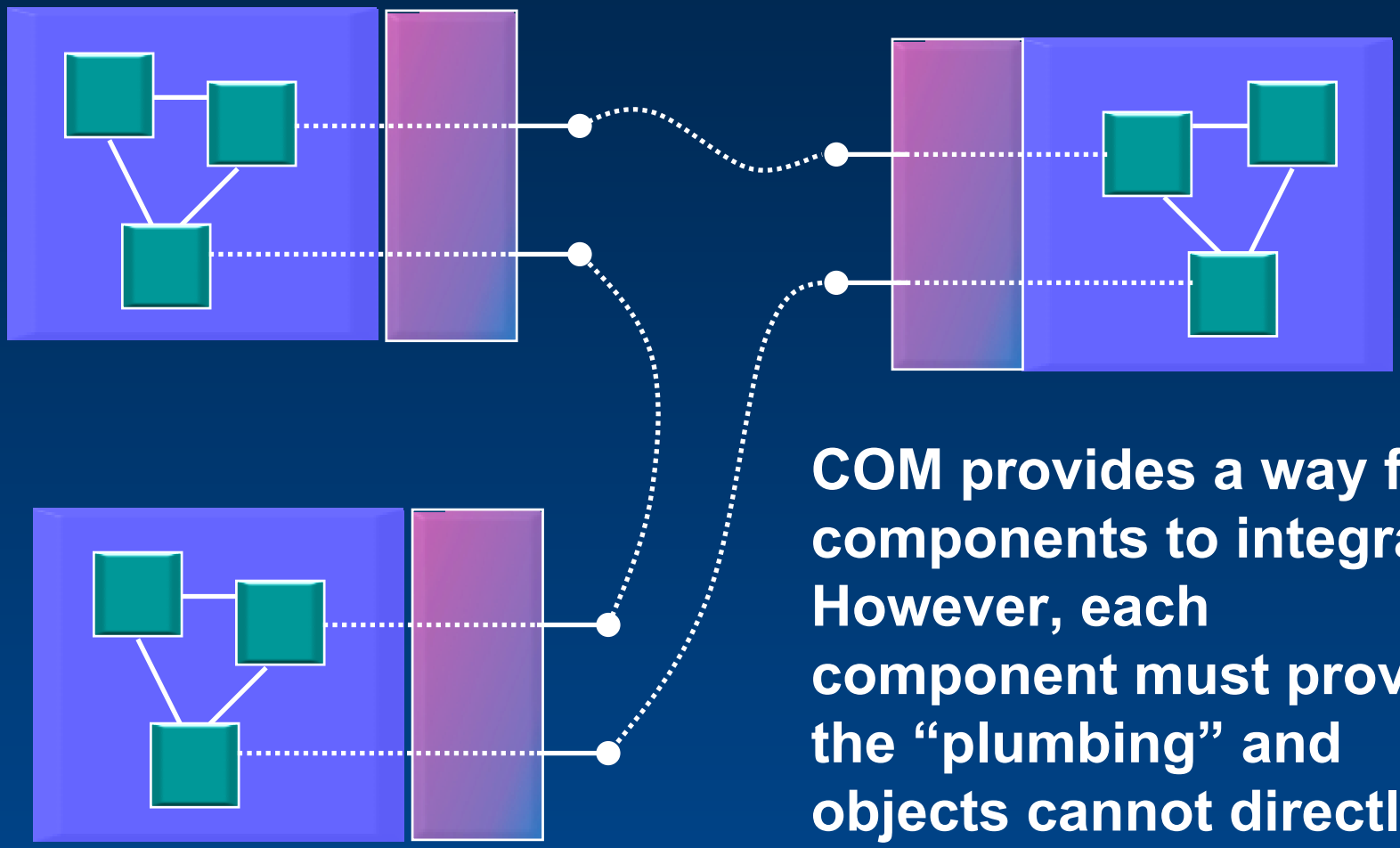
# The .NET Evolution



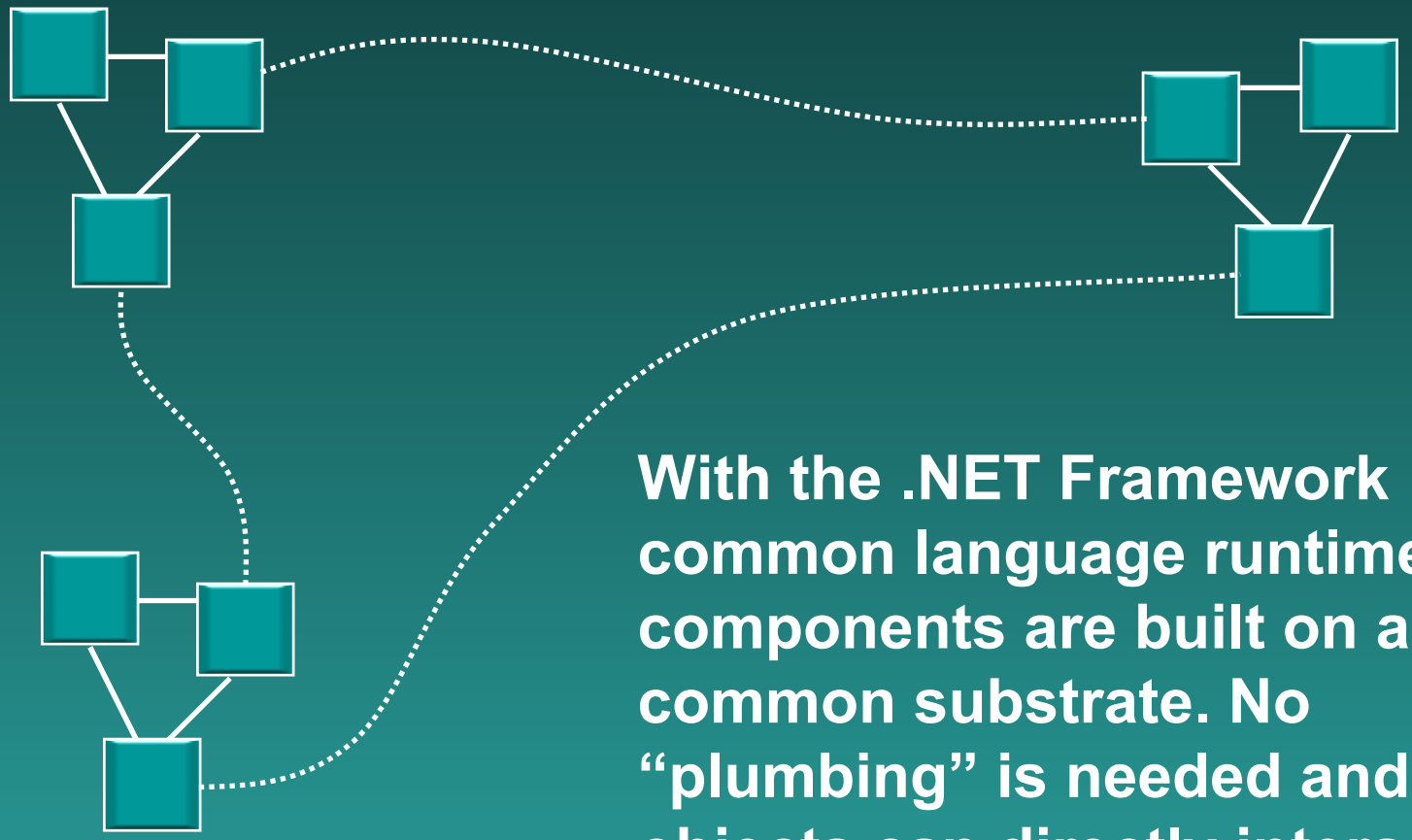
**Before COM, applications were completely separate entities with little or no integration**

# The .NET Evolution



**COM provides a way for components to integrate. However, each component must provide the “plumbing” and objects cannot directly interact.**

# The .NET Evolution



With the .NET Framework common language runtime, components are built on a common substrate. No “plumbing” is needed and objects can directly interact

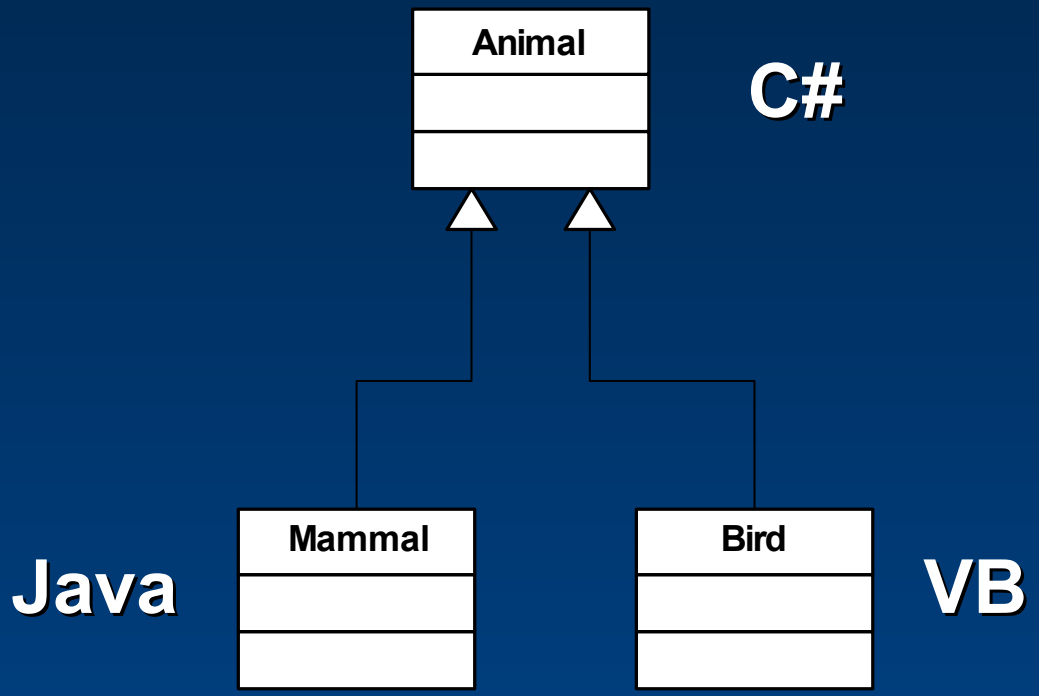
# Simplify Development

- Completely eliminates COM plumbing
- No more…
  - Registration **=>self described apps**
  - GUIDs **=>hierarchical namespaces**
  - .IDL files **=>unified object model**
  - HRESULTs **=>structured exceptions**
  - IUnknown **=>common root object**
  - AddRef/release **=>garbage collector**
  - CoCreateInstance **=>"new" operator**

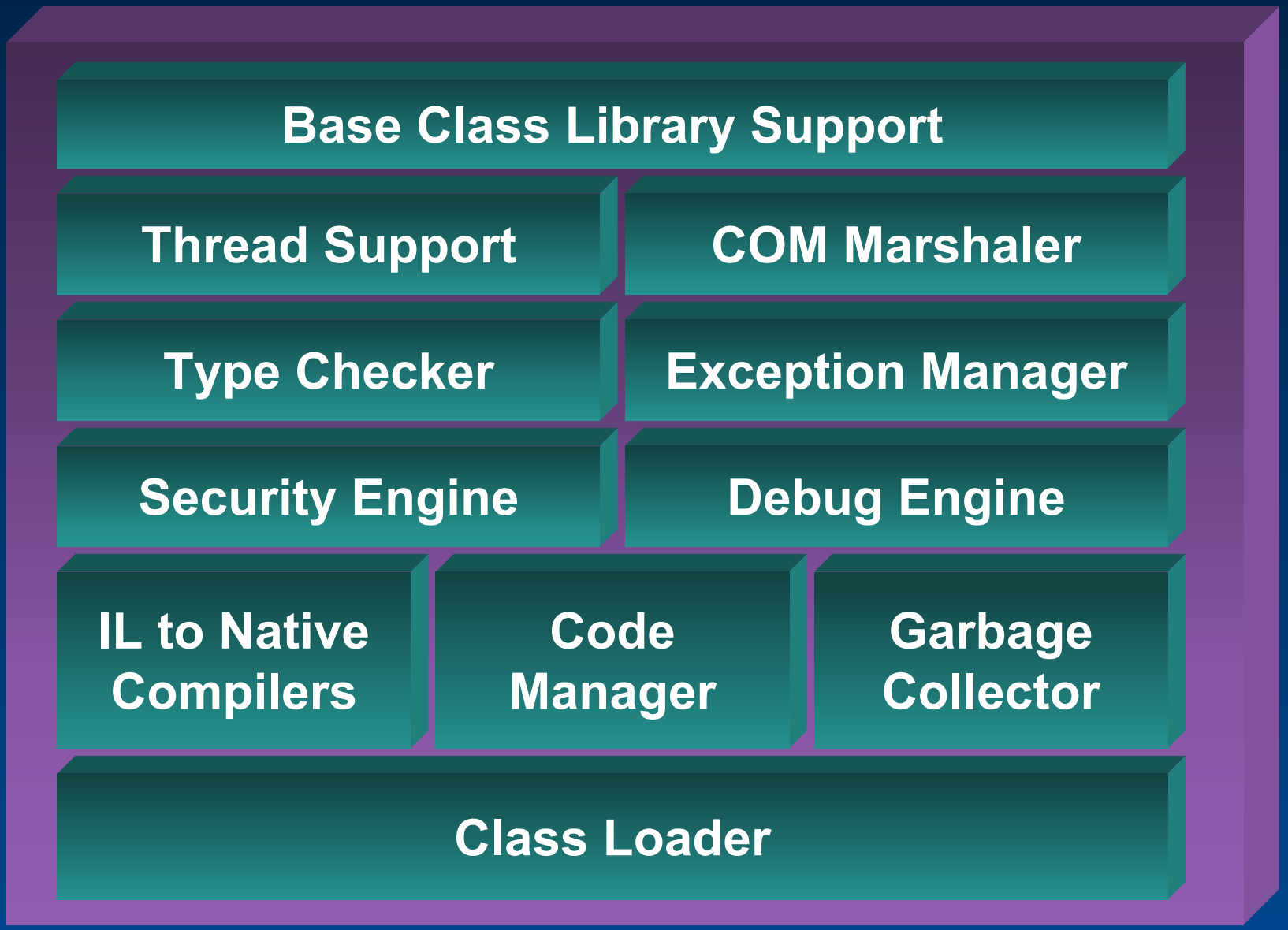
# Simplify Development

- Common Type System
  - Common instance and type definition
- Enables clean OO programming
  - Classes and Interfaces
  - Constructors, Properties, Methods
  - Cross language inheritance

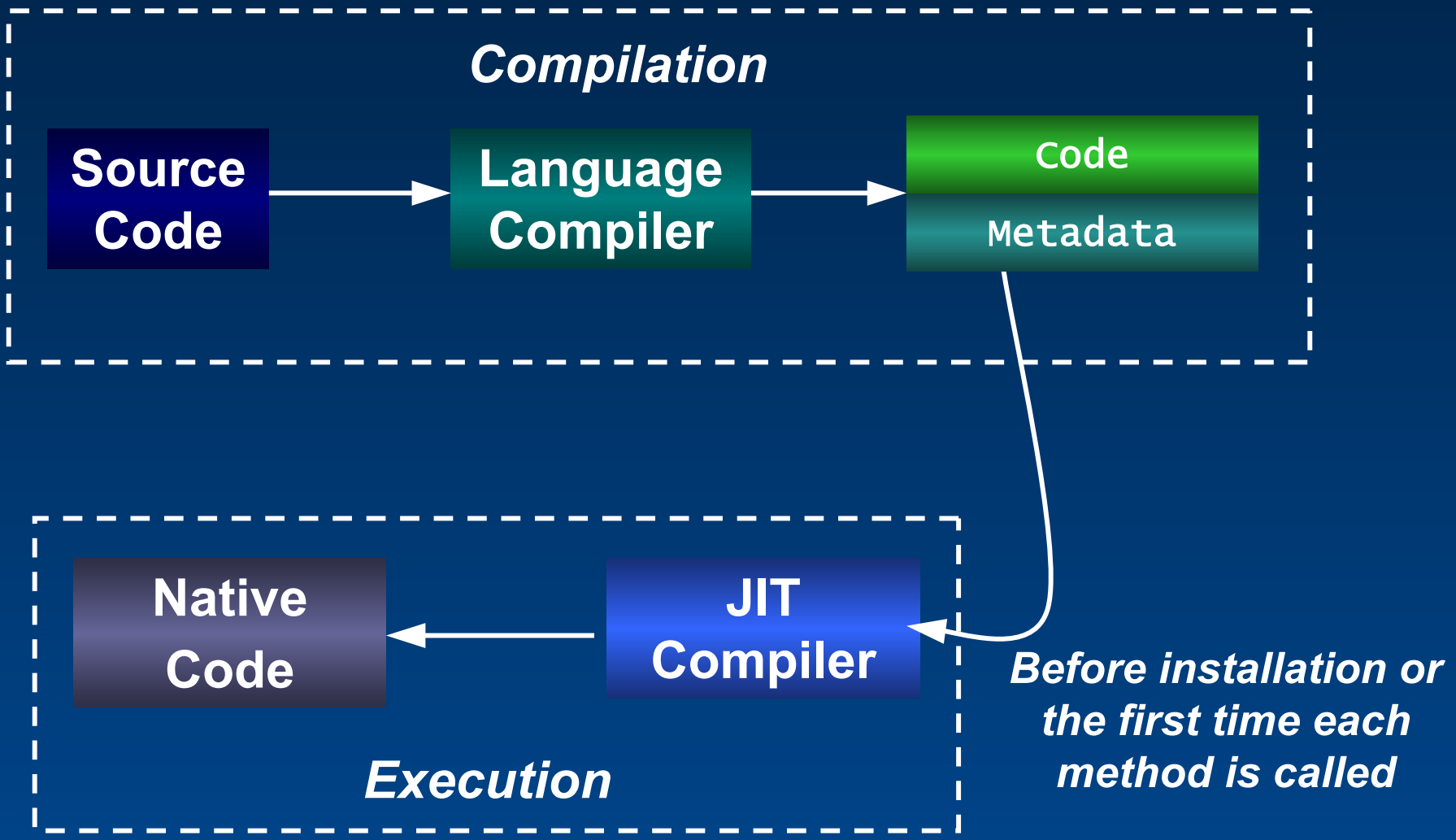
# Inheritance



# Common Language Runtime



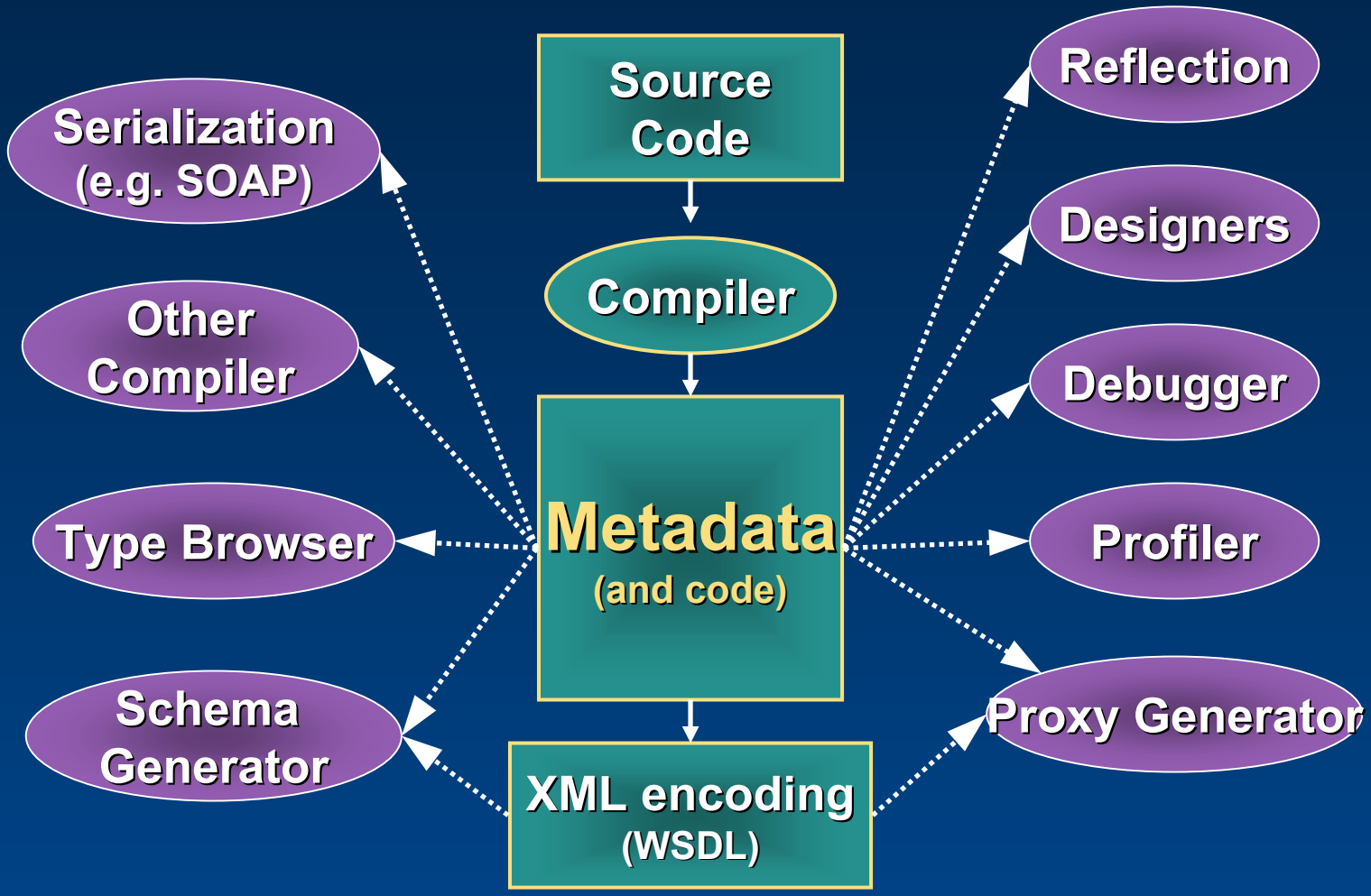
# Compilation and Execution



# Metadata

- Key to simpler programming model
- Generated automatically
  - Stored with code in executable file (.dll or .exe)
  - Uses existing COFF format
    - Via existing extension mechanism
  - Stored in binary format
- Convertible to/from XML Schema
- Convertible to/from COM type libraries

# Metadata: Creation And Use



# Robust And Secure

- Automatic lifetime management
  - All .NET objects are garbage collected
  - No stray pointers, no circular references
  - Multi-generational mark-and-compact GC
  - Self configuring, dynamically tuned
- Exception handling
  - Error handling is a 1st class concept (not bool or HRESULTs)
  - Dramatically improved error reporting
  - Integrated with windows SEH

# Robust And Secure

- Native code compilation
  - MSIL
  - No interpreter
  - Install-time or run-time IL to native compilation
  - C++ managed extensions
- Code correctness and type-safety
  - IL can be verified to guarantee type-safety
  - No unsafe casts, no uninitialized variables, no out-of-bounds array indexing
- Evidence-based security
  - Policy grants permissions based on evidence (signatures, origin)
  - Extensible permissions and policy

# Multi-language Platform

- The freedom to choose language
  - All features of .NET platform available to any .NET programming language
  - Application components can be written in multiple languages
- Highly leveraged tools
  - Debuggers, profilers, code coverage analyzers, etc. Work for all languages

# Deployment And Management

- Assemblies
  - The unit of deployment, versioning, and security
  - Like DLLs, but self-describing through manifest
- Zero-impact install
  - Applications and components can be shared or private
- Side-by-side execution
  - Multiple versions of the same component can co-exist, even in the same process

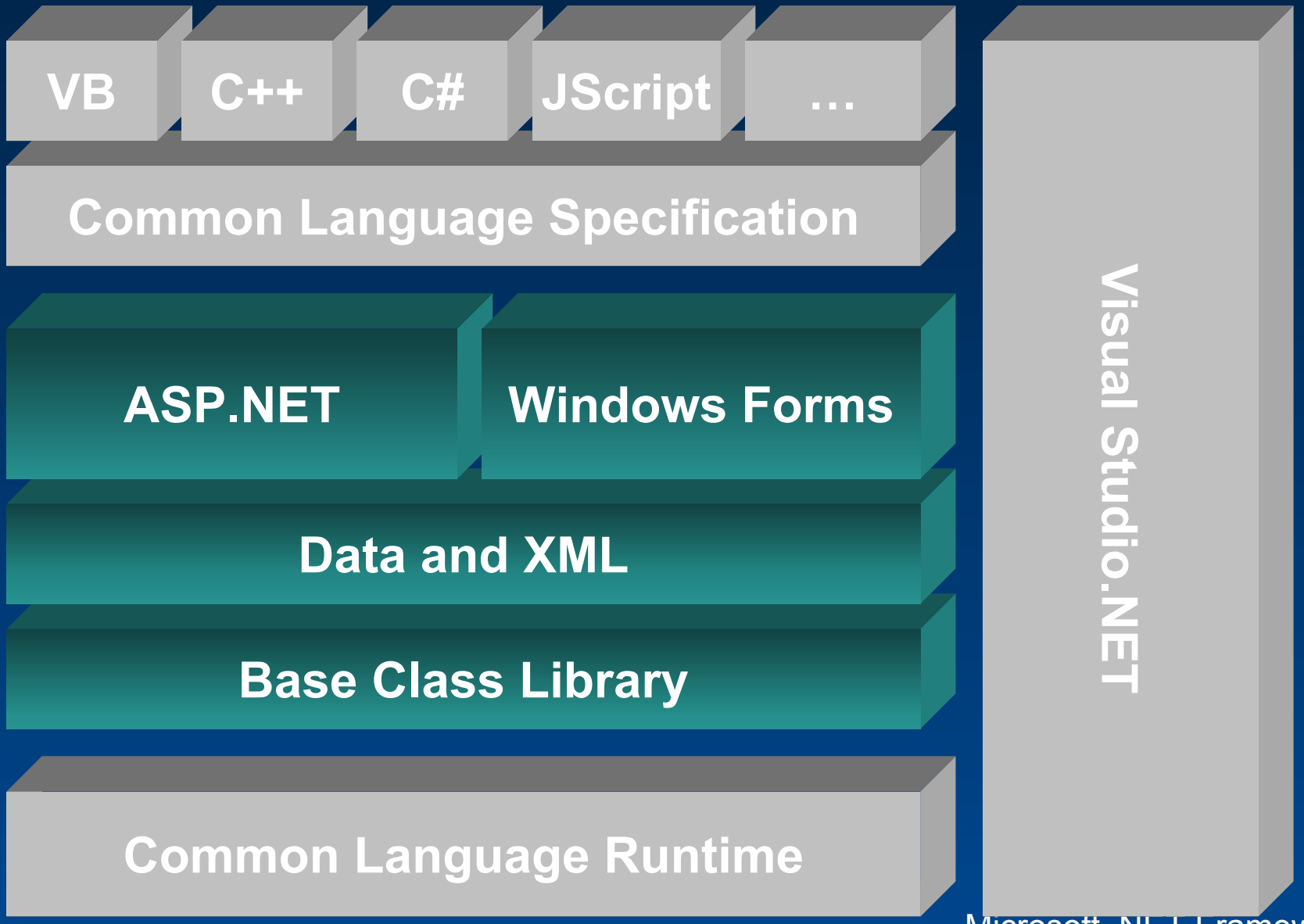
# Assemblies

- Unit of deployment
  - One or more files, independent of packaging
  - Self-describing via manifest
- Versioning
  - Provided by compiler use of attributes
  - Policy per-application as well as per-machine
- Security boundary
  - Assemblies are granted permissions
  - Methods can demand proof that a permission has been granted to entire call chain
- Types named relative to assembly

# Seamless Integration

- Any .NET class can be used as a COM class with zero extra work
- COM classes can be imported as .NET classes
- .NET classes utilize com+ services
  - Transactions
  - Object pooling
  - Etc…
- DLL entry points
- C++ managed extensions

# Framework, Languages, And Tools



# Framework Design Goals

- Make it simple to use
- Factored and extensible
- Web standards and practices as the foundation
- Unify application models

# Make It Simple To Use

- Organization
  - Code organized in hierarchical namespaces and classes
- Unified type system
  - Everything is an object, no variants, one string type, all character data is Unicode
- Component Oriented
  - Properties, methods, events, and attributes are first class constructs
  - Design-time functionality

# How Much Simpler?

## Windows API

```
HWND hwndMain = CreateWindowEx(
    0, "MainWClass", "Main Window",
    WS_OVERLAPPEDWINDOW | WS_HSCROLL | WS_VSCROLL,
    CW_USEDEFAULT, CW_USEDEFAULT,
    CW_USEDEFAULT, CW_USEDEFAULT,
    (HWND)NULL, (HMENU)NULL, hInstance, NULL);
ShowWindow(hwndMain, SW_SHOWDEFAULT);
UpdateWindow(hwndMain);
```

## .NET Framework

```
Form form = new Form();
form.Text = "Main Window";
form.Show();
```

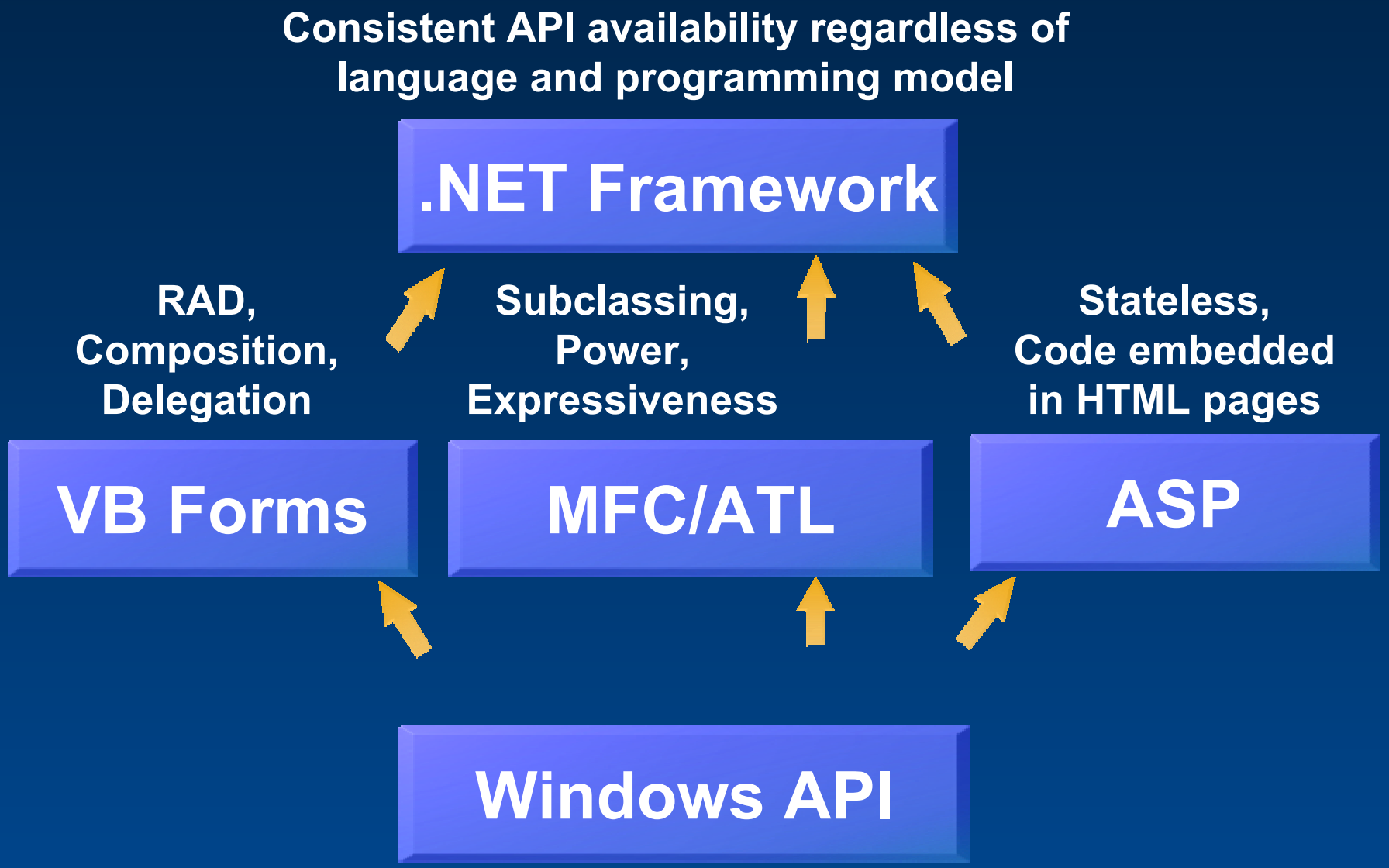
# Factored And Extensible

- The Framework is not a “black box”
- Any .NET class is available for you to extend through inheritance
  - Unlike COM, you’re using and extending the class itself, not a “wrapper”
- Plug-and-play components and subsystems
- Cross-language inheritance!

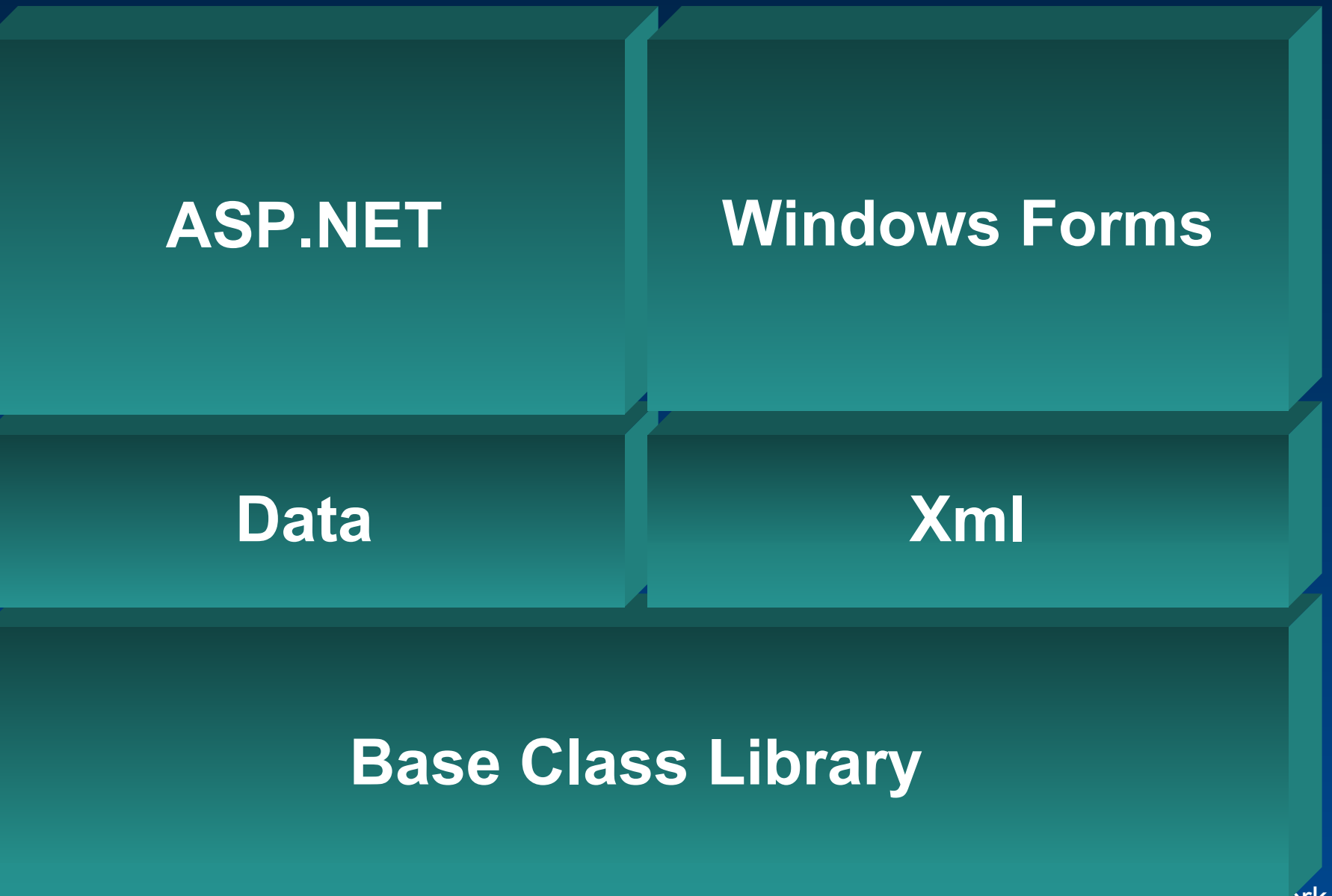
# Web Standards/Practices

- The .NET Framework supports
  - HTML, XML, SOAP, XSLT, XPath
- Web services enables internet scale distributed apps
  - State-full connected model does not work
  - Loosely connected web services enables scalable apps
- XML support is built in deeply
  - Data, remoting, serialization, documentation, config
- This affects everything
  - Base services, data access, UI, invocation/activation, programming model, tools

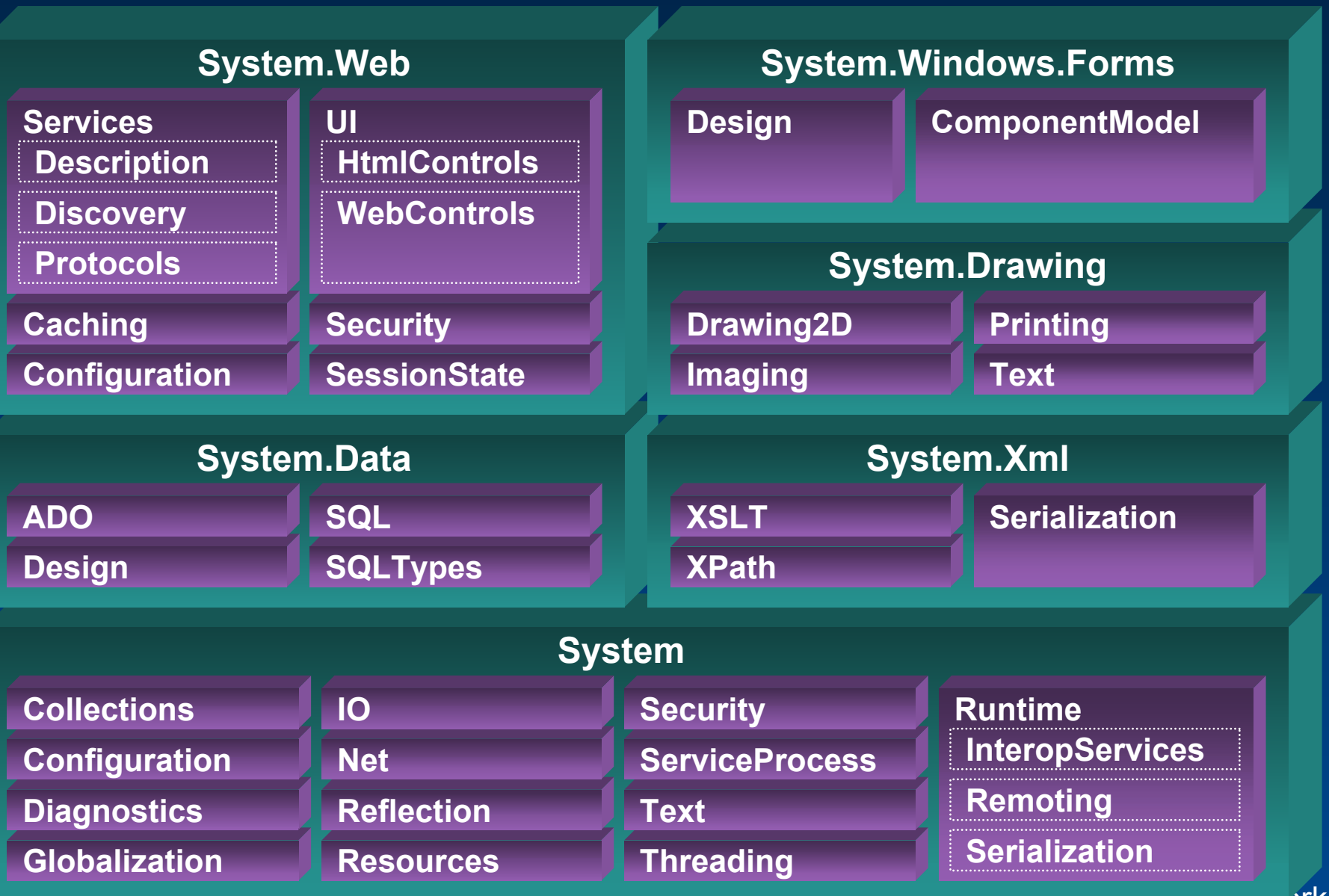
# Unify Programming Models



# The .NET Framework



# The .NET Framework



# Base Framework

# Data And XML

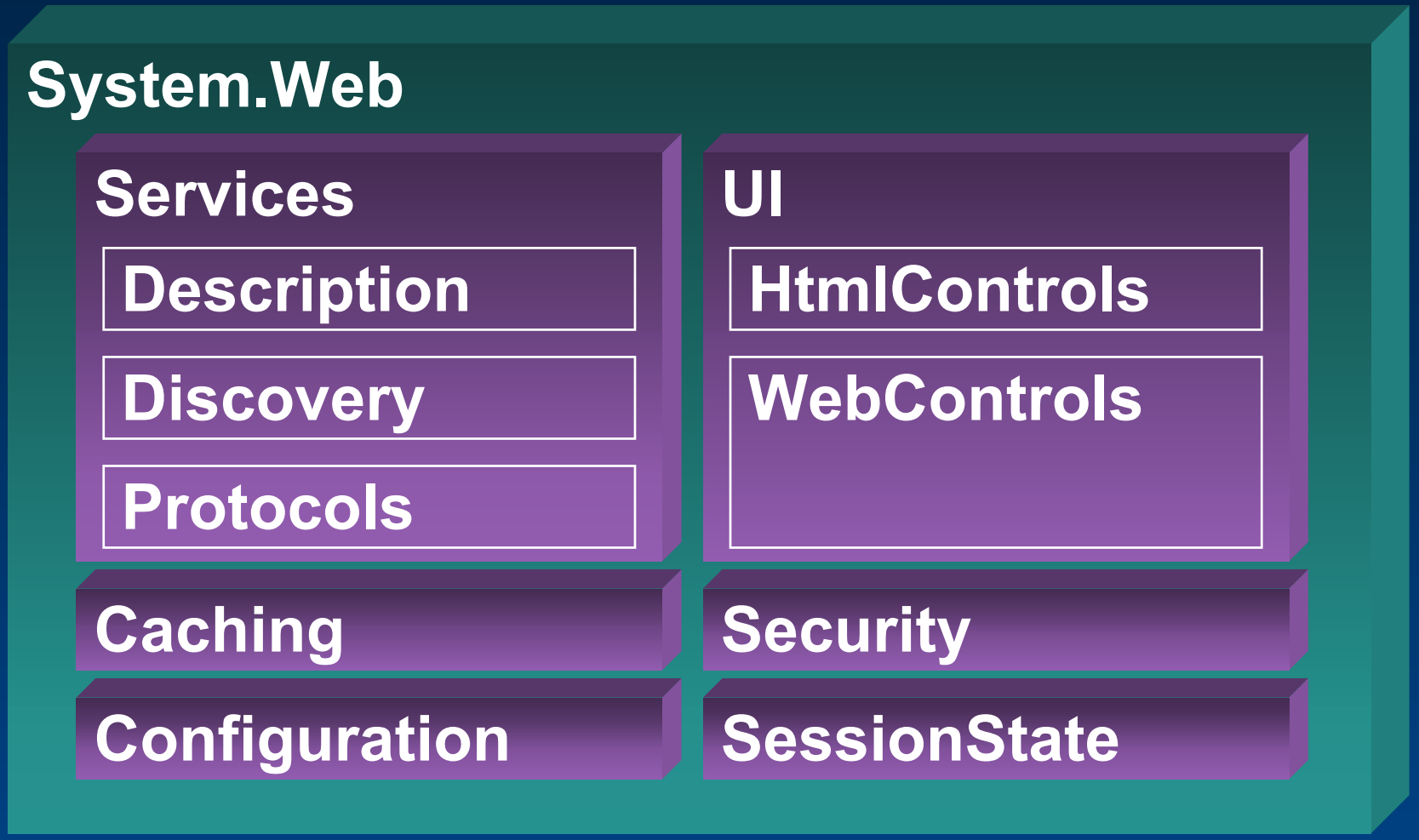
## System.Data

- ADO
- Design
- SQL
- SQLTypes

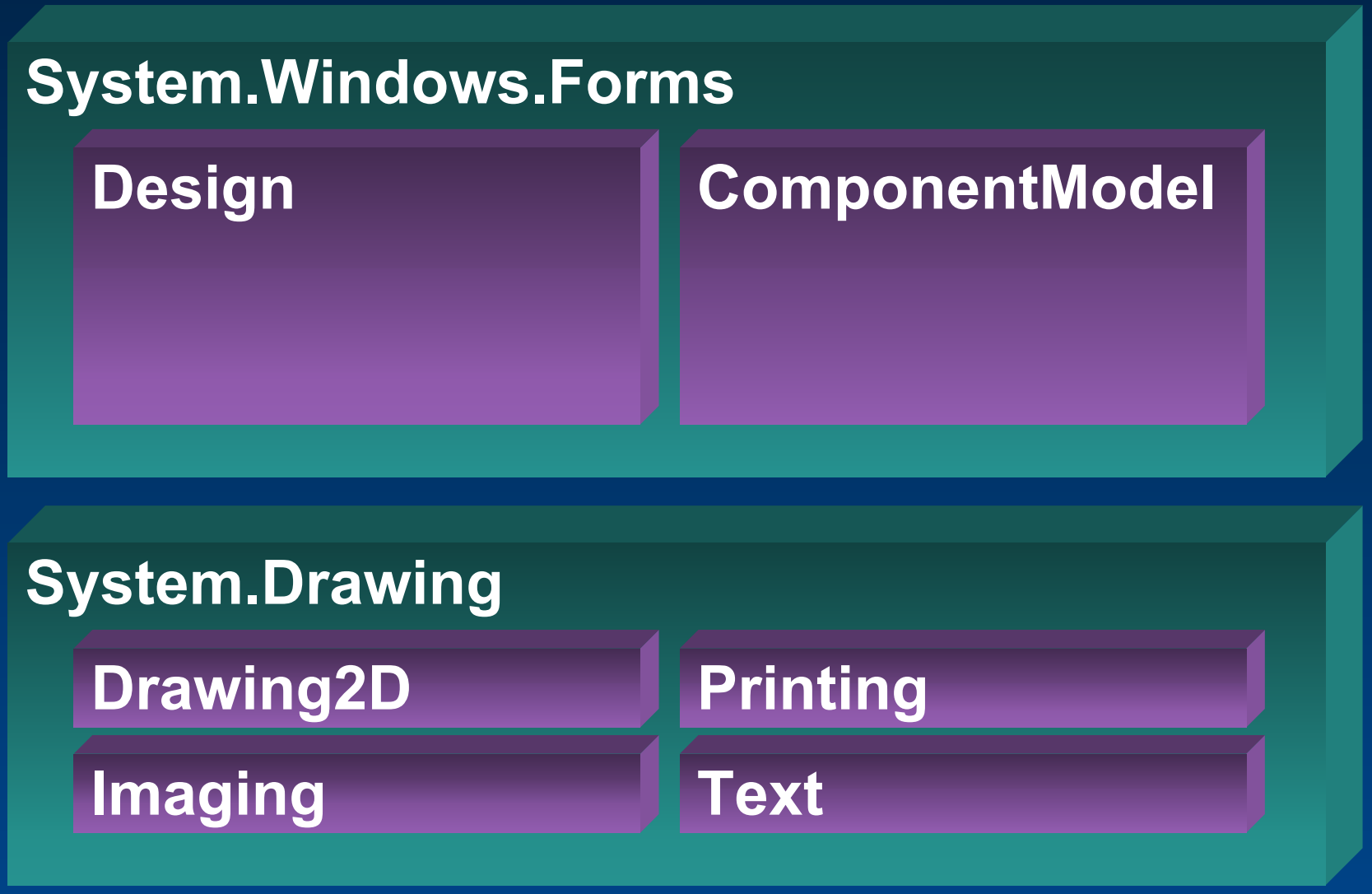
## System.Xml

- XSLT
- XPath
- Serialization

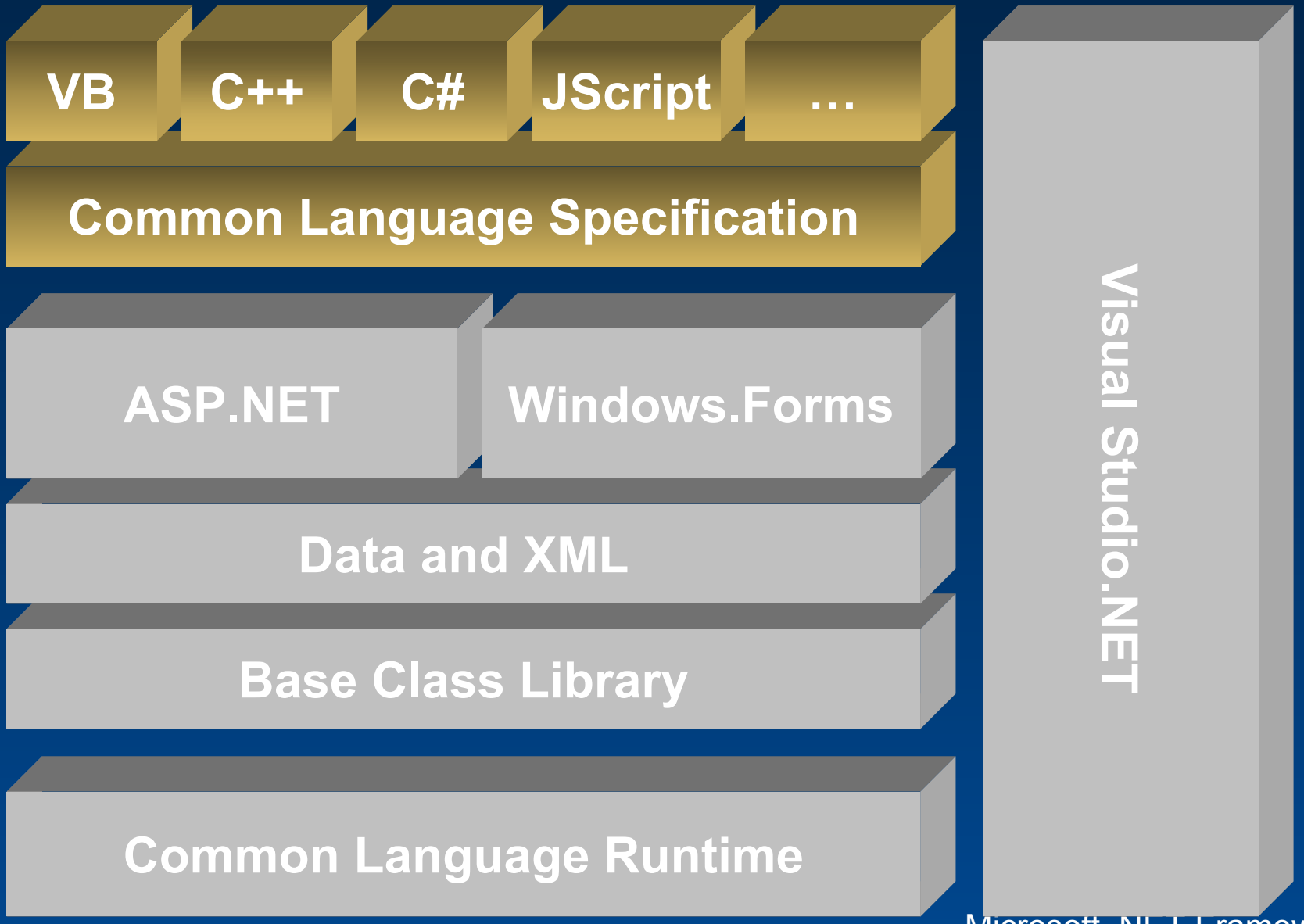
# ASP.NET


System.Web
Services
Description
Discovery
Protocols
UI
HtmlControls
WebControls
Caching
Security
Configuration
SessionState

# Windows Forms



# Framework, Languages, And Tools



# Languages

- The .NET Platform is Language Neutral
  - All .NET languages are first class players
  - You can leverage your existing skills
- Common Language Specification
  - Consumer: Can use the .NET Framework
  - Extender: Can extend the .NET Framework
- We are providing
  - VB, C++, C#, JScript
- Third-parties are building
  - APL, COBOL, Pascal, Eiffel, Haskell, ML, Oberon, Perl, Python, Scheme, Smalltalk…

# Summary

- Simplified development
- Unified programming models
- Multi-language, OO platform
- Web standards and best practices
- Simple to deploy, run, & maintain

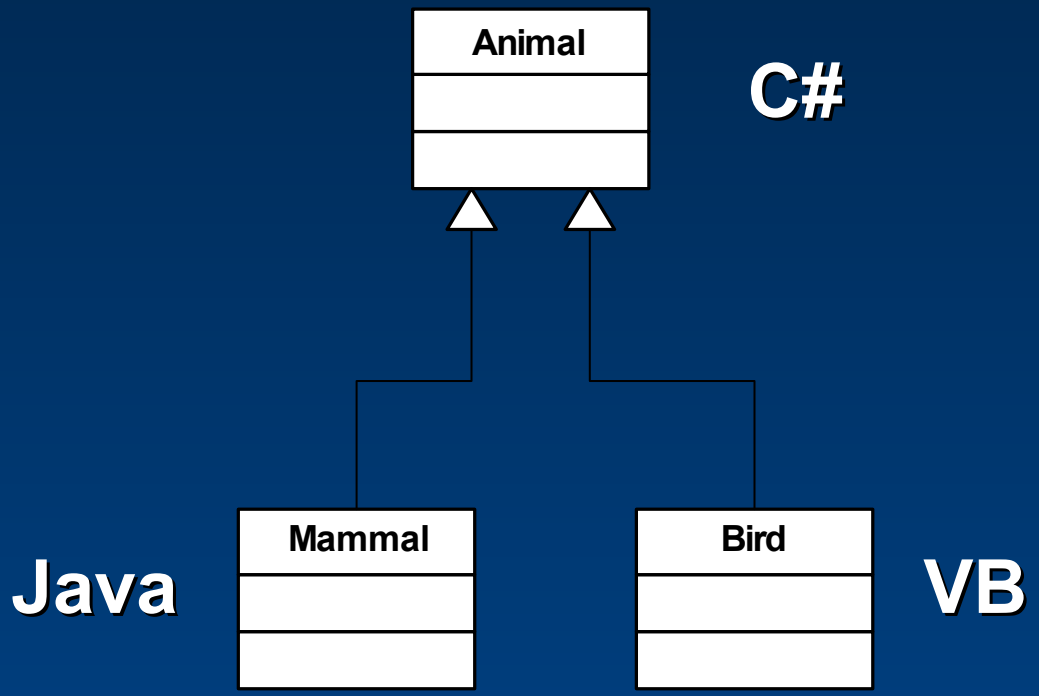
# Application Development

Microsoft .NET Framework - Part II

# The Tools

- Visual Studio .NET
- Command Line Tools

# Inheritance



# Sample Applications

- Focus on .NET Framework Concepts
  - .NET Application
  - .NET “Assembly”
  - Reuse class from C# in VB
  - ASP.NET application
  - Web Service

# Of Academic Interest

- Generics for the CLR (and C#)
- Shared Source Common Language Infrastructure
  - Runs on Windows XP, the FreeBSD operating system, and Mac OS X 10.2
  - Released under a shared source initiative

# New in .NET Framework 1.1

- Mobile Web Applications
- Granular Version Control: Side-by-Side Execution
- Execution of Windows Forms from the Internet
- Code Access Security for ASP.NET Applications
- Native Support for ODBC and Oracle Databases
- Unified Programming Model for Smart Client App.
- Support for IPv6
- Scalability, Performance, Documentation

# Links to More Information

- msdn.microsoft.com/netframework/
- www.gotdotnet.com
- msdn.microsoft.com/net/ecma/
- research.microsoft.com/projects/clrgen
- msdn.microsoft.com/net/sscli/

# Questions?


Microsoft®