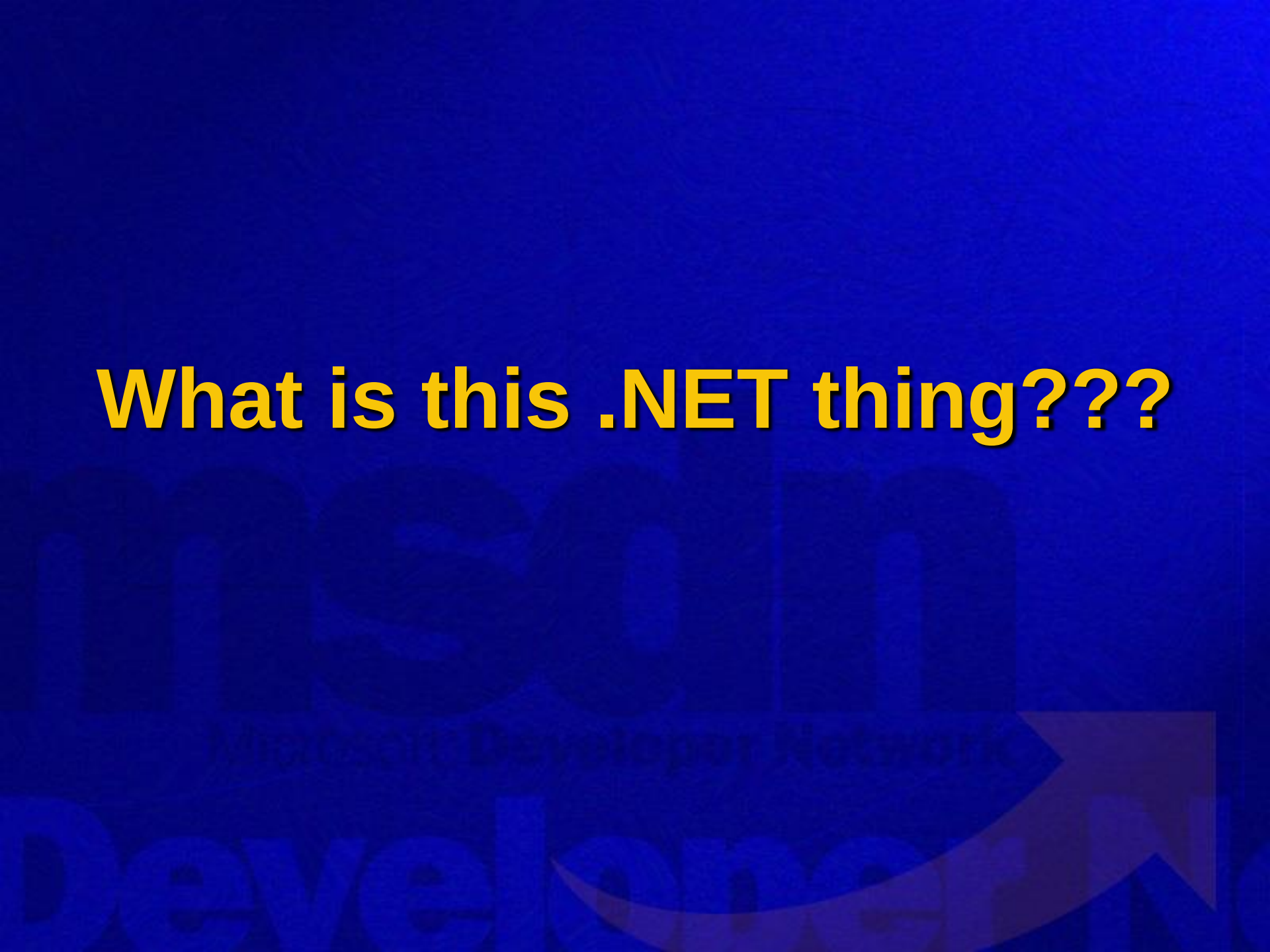

Microsoft®
.net™

What is this .NET thing???



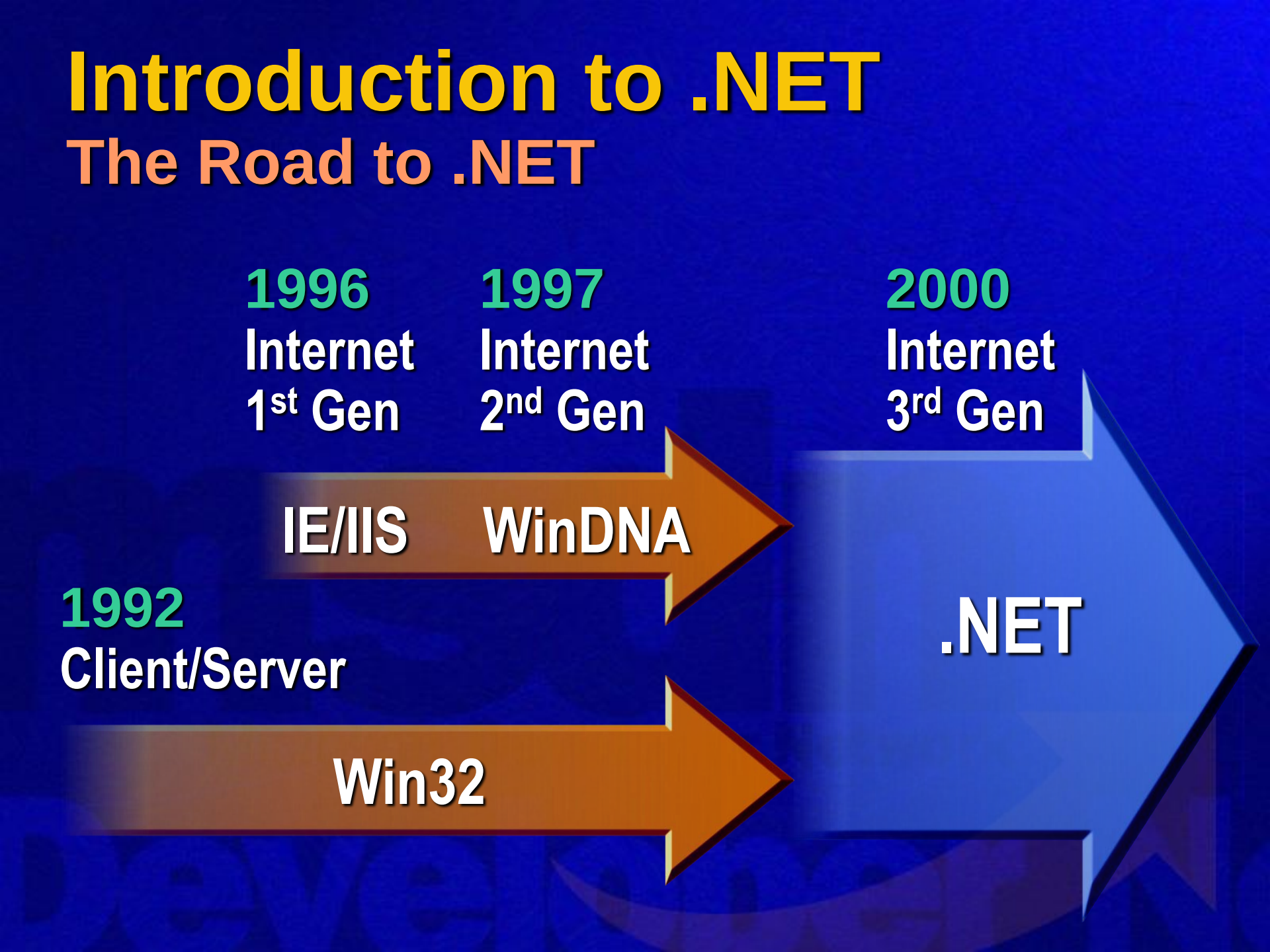
.NET Framework Overview

Session Prerequisites

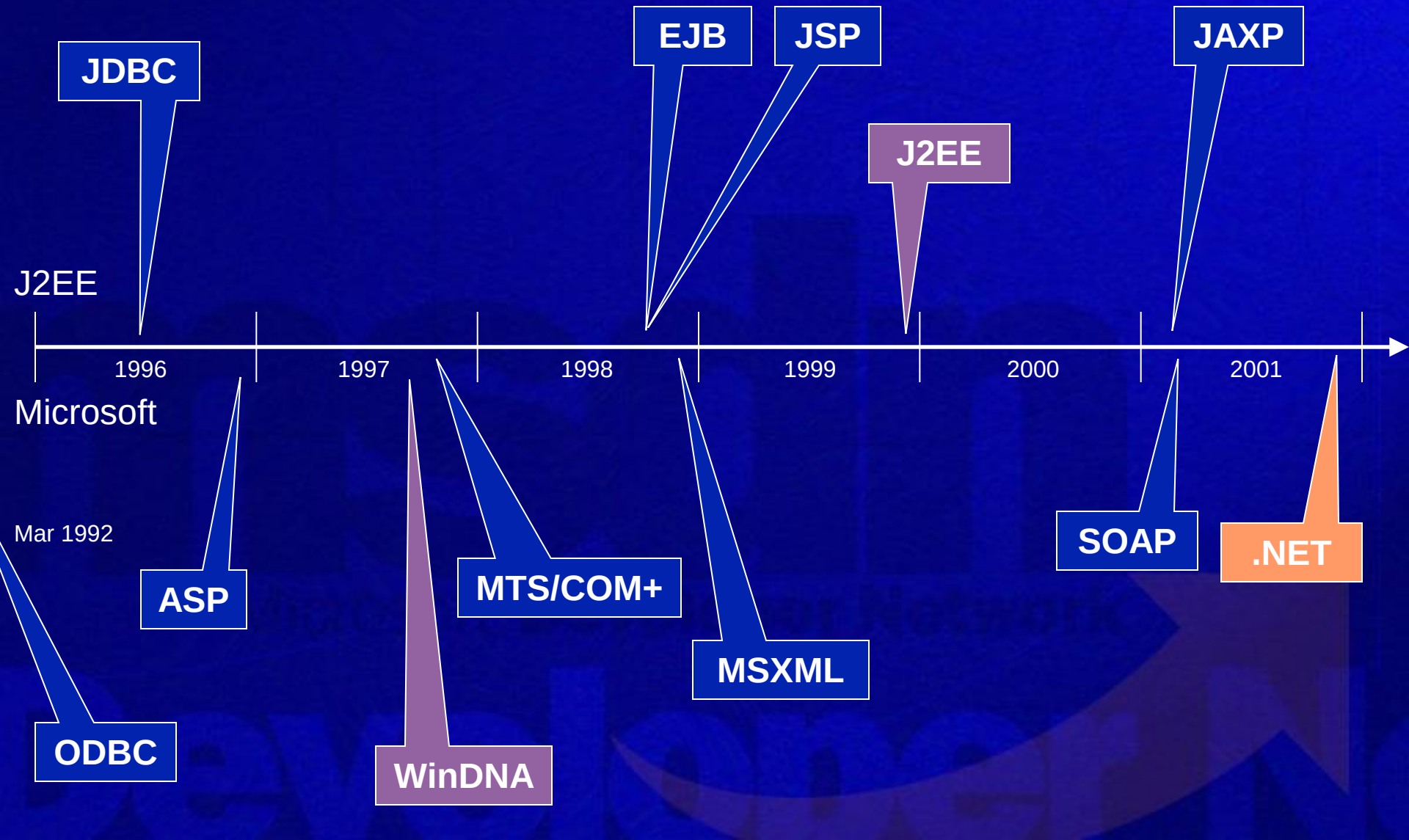
- This session assumes that you understand the fundamentals of
  - Distributed Web application development
  - ASP
  - ADO
  - COM
- This is a Level 200 Session

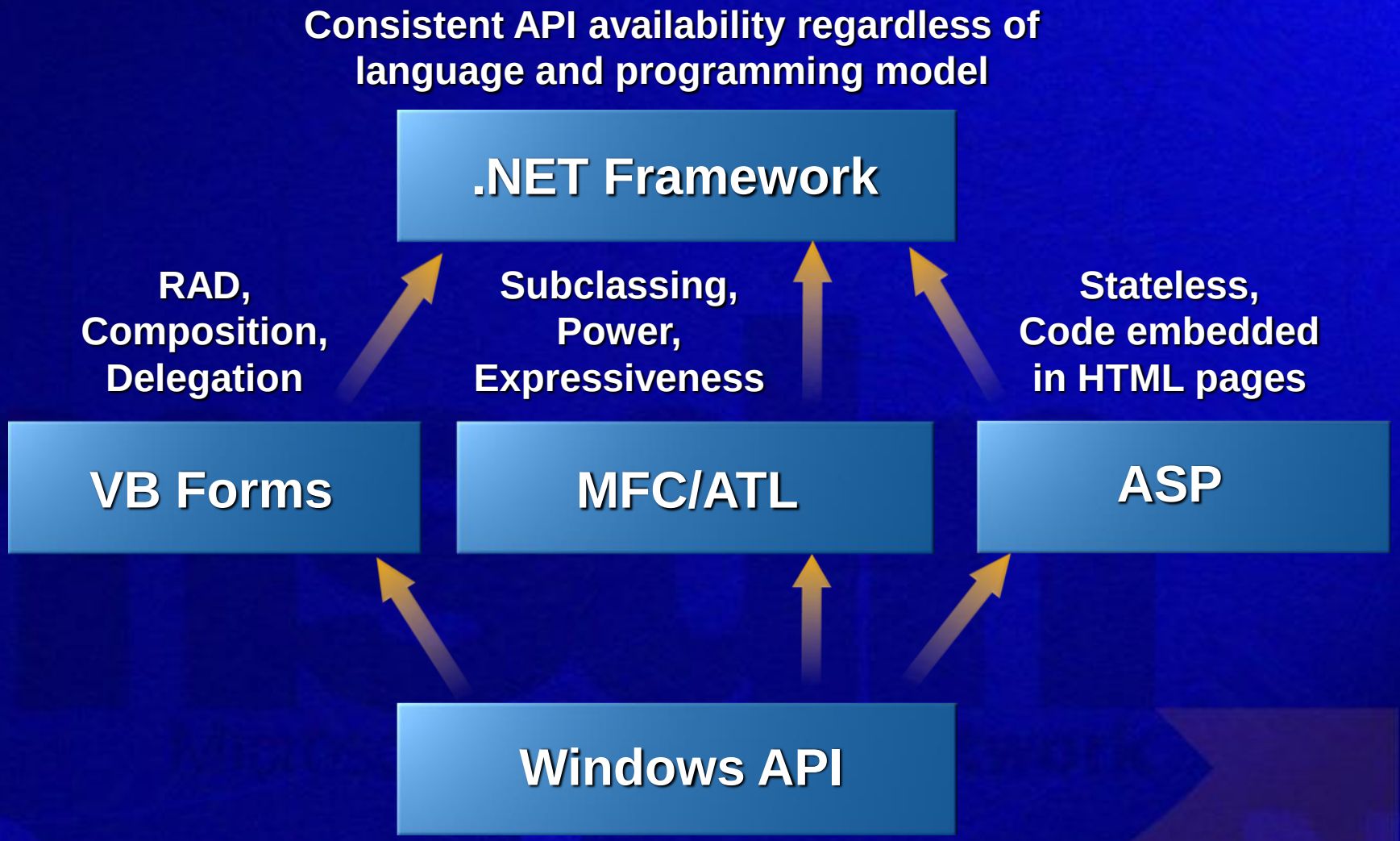
Introduction to .NET

The Road to .NET



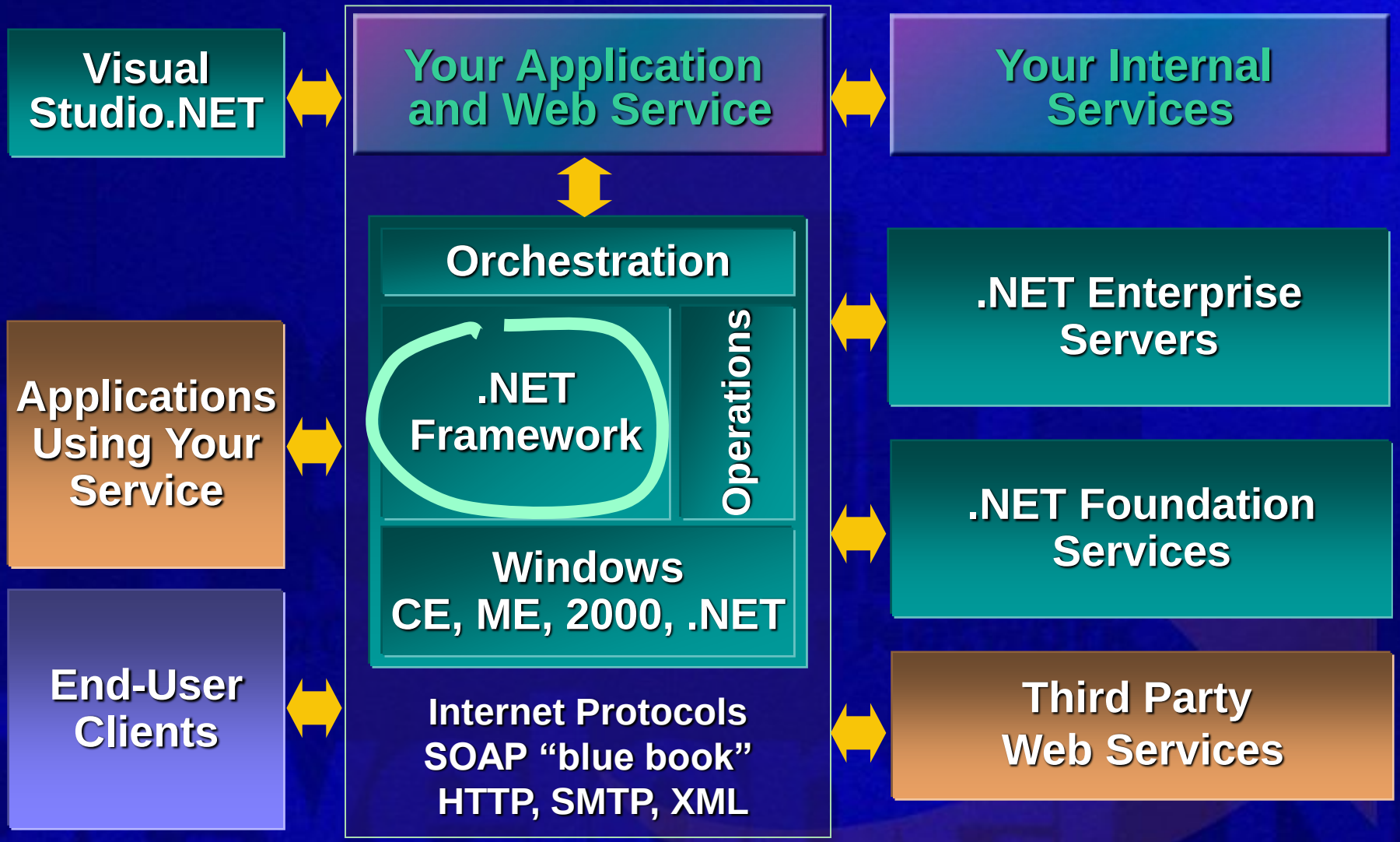
I. Technology Timeline




Consistent API availability regardless of language and programming model
.NET Framework
RAD, Composition, Delegation
Subclassing, Power, Expressiveness
Stateless, Code embedded in HTML pages
VB Forms
MFC/ATL
ASP
Windows API

Introduction to .NET

The .NET Platform



Agenda

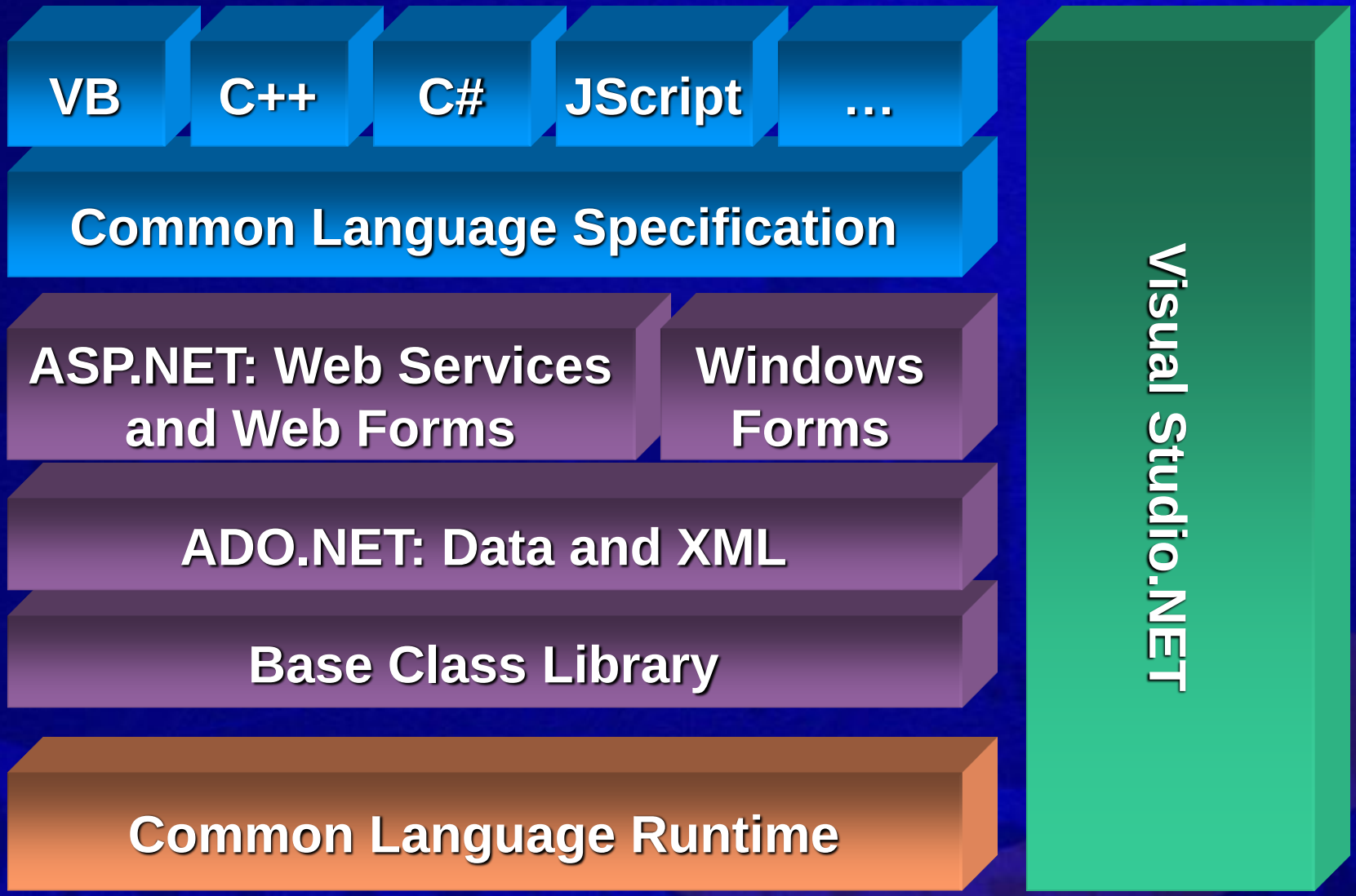
- Introduction to .NET
- The .NET Framework
- Common Language Runtime
- Building User Interfaces
- Data and ADO.NET
- Summary

The .NET Framework
What Is the .NET Framework?

- A set of technologies that
  - Unite today's isolated Web applications
  - Make information available anytime, anywhere
  - Simplify development and deployment
- How does .NET achieve the above?
  - Web Services
  - ADO.NET Datasets and XML support throughout the platform
  - Rich tools, runtime services and XCOPY Deployment

The .NET Framework

The .NET Framework and Visual Studio.NET



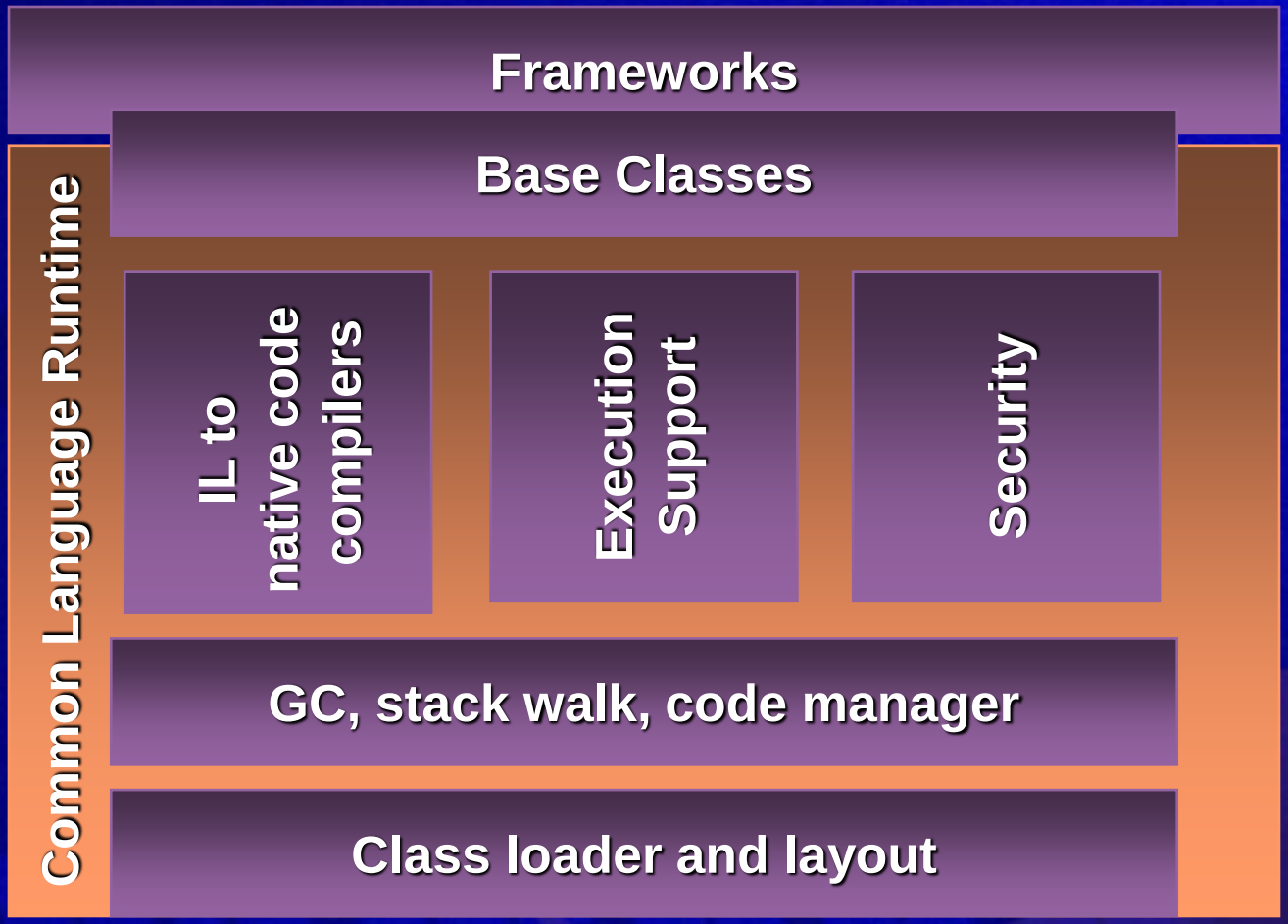
The .NET Framework
.NET Framework Services

- ASP.NET
  - Logical evolution of ASP (compiled)
- Web forms
  - Manageable code (non spaghetti)
- Windows® forms
  - Framework for building rich clients
- ADO.NET, evolution of ADO
  - New objects (e.g., DataSets)
- XML Support Throughout

Agenda

- Introduction to .NET
- The .NET Framework
- Common Language Runtime
- Building User Interfaces
- Data and ADO.NET
- Summary

Common Language Runtime
Architecture



Common Language Runtime
Goals

- Development
  - Standard class framework
  - Automatic memory management
  - Consistent error handling
  - Mixed language applications
  - Multiple platforms
  - Safer execution
- Deployment
  - Removal on registration dependency
  - Safety – fewer versioning problems
  - THE END of ‘DLL Hell’

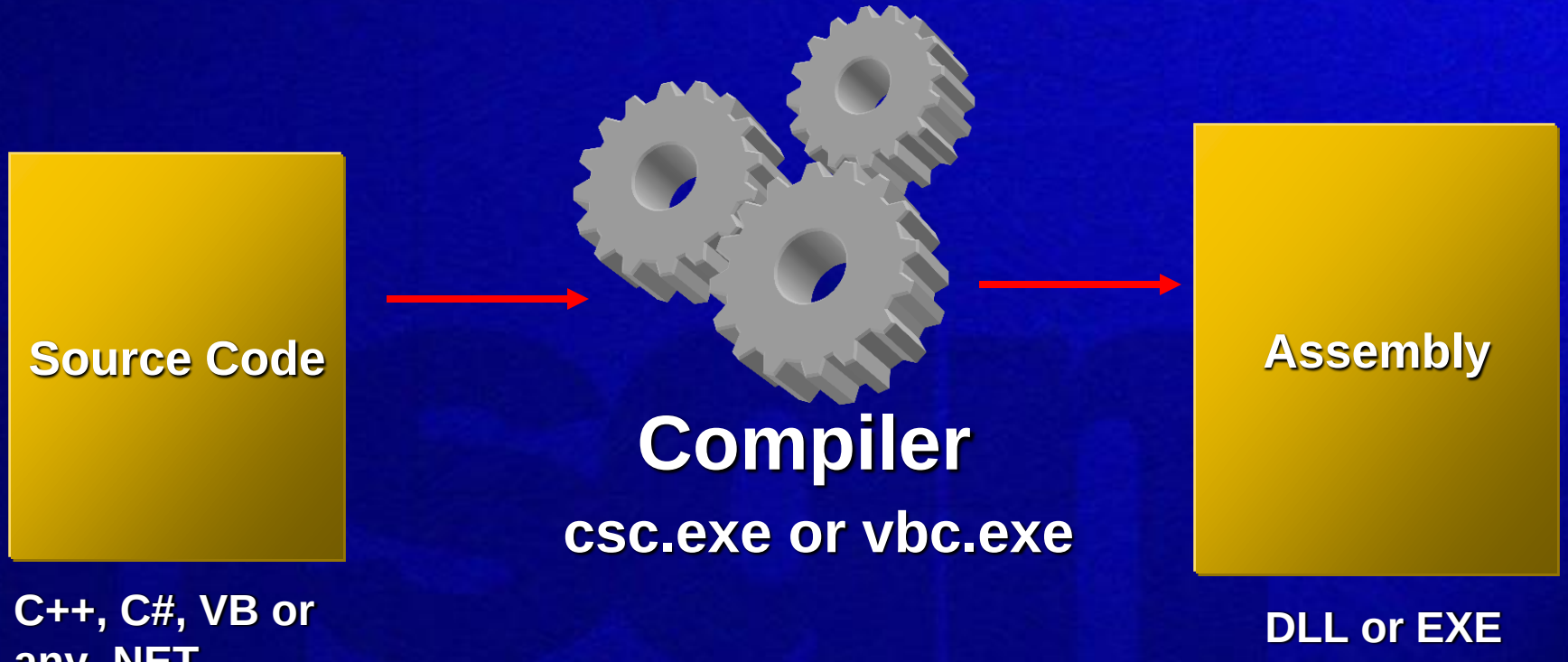
Common Language Runtime
Multiple Language Support

- What about types?
  - Common type system (CTS)
- Other languages and compilers
  - Common Language Specification (CLS)

Runtime Benefits

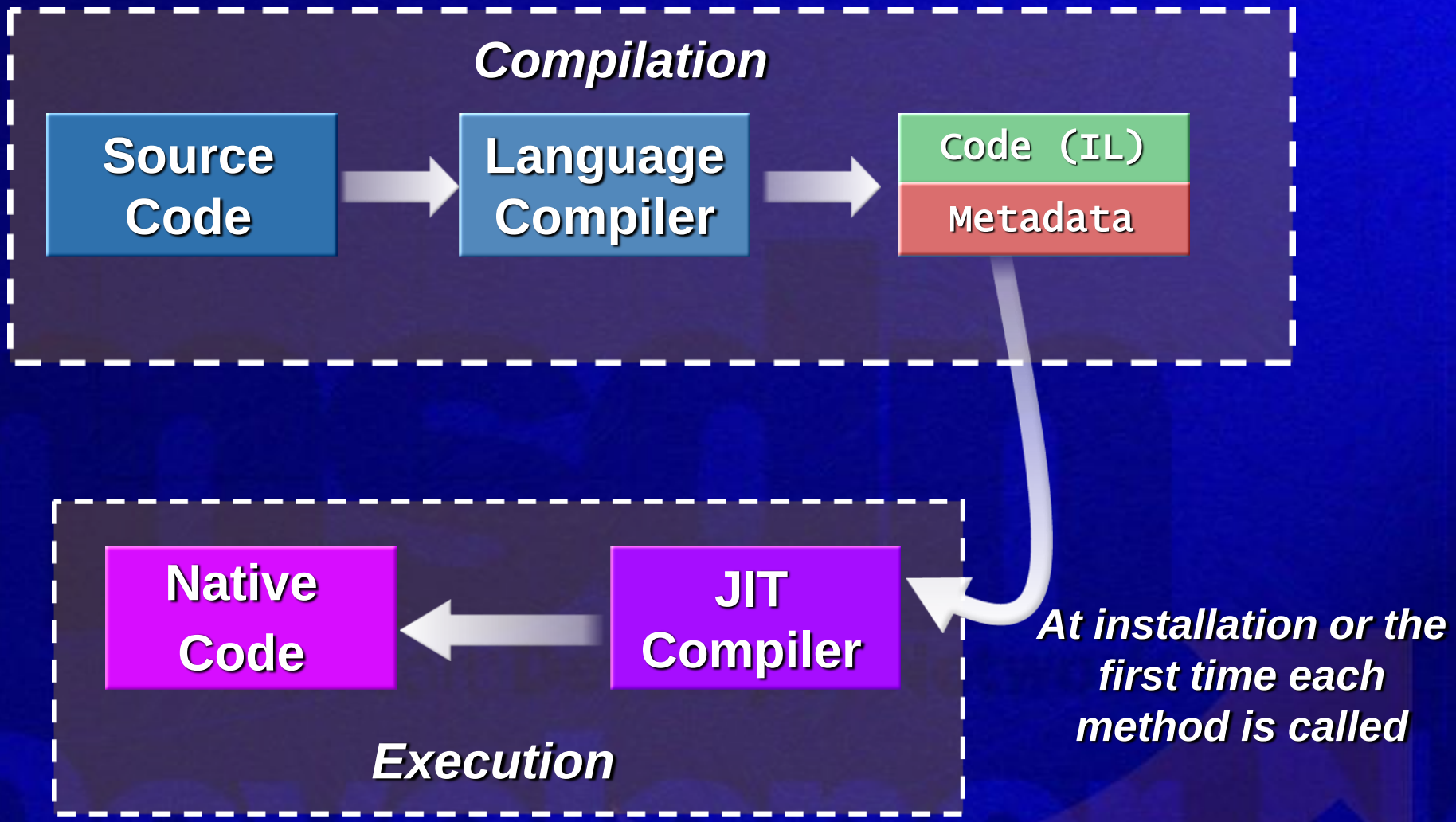
- Eliminates memory management drudgery
- Kills entire classes of bugs (e.g., memory corruption, ref counting)
- Security model ensures safety
- XCOPY deployment, no registry required
- Auto-versioning, no more DLL Hell
- Scalability, performance, reliability all improve

Common Language Runtime
Compilation

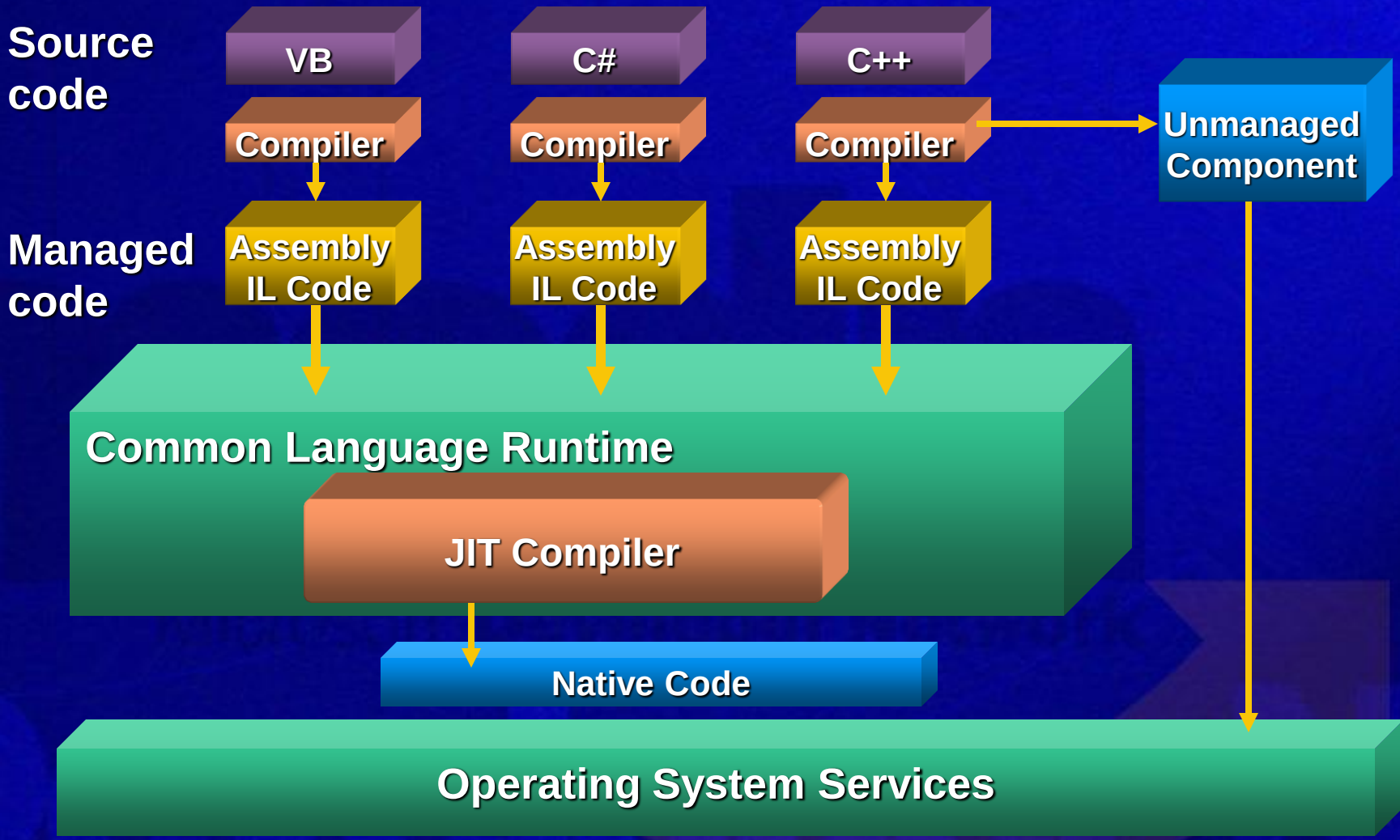


Implementation and Benefits

Compilation and Execution

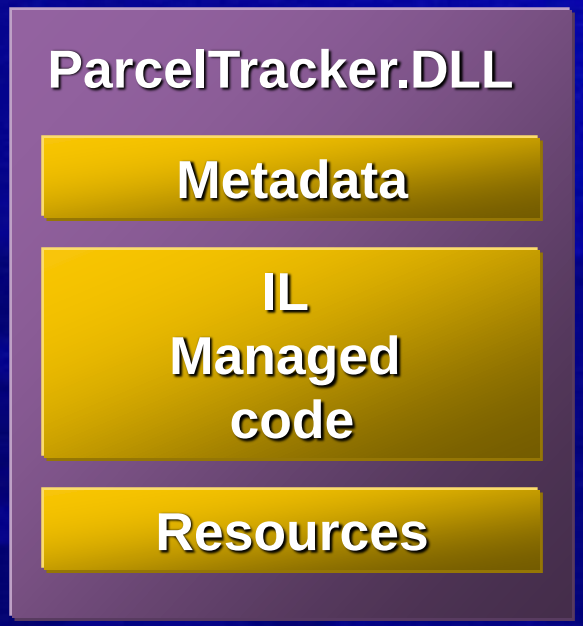


Common Language Runtime
Execution Model



Common Language Runtime

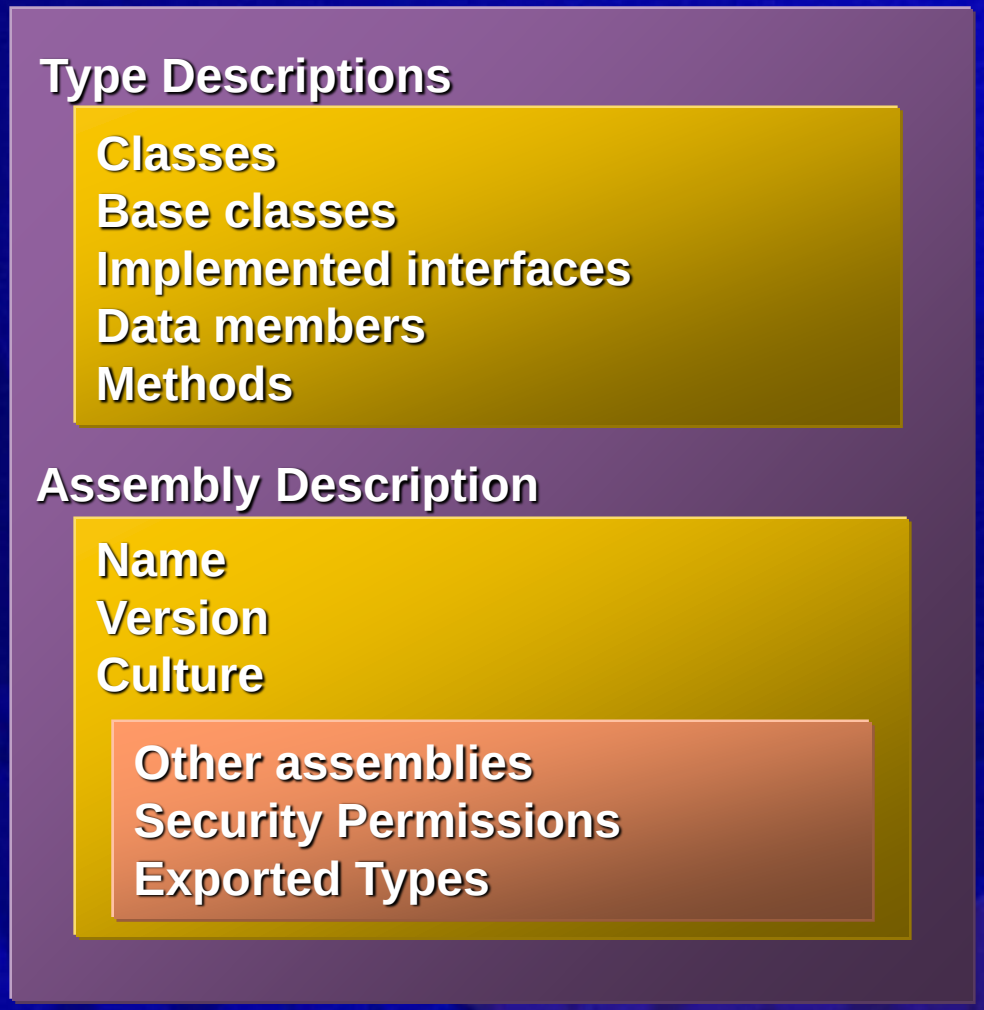
Assemblies



Common Language Runtime
Metadata

- Type Information
  - More complete than IDL / TLB
  - Automatically bound into assembly
    - Inseparable
    - Stored in binary format
  - Describes every class type
  - Used by VS.NET IntelliSense™

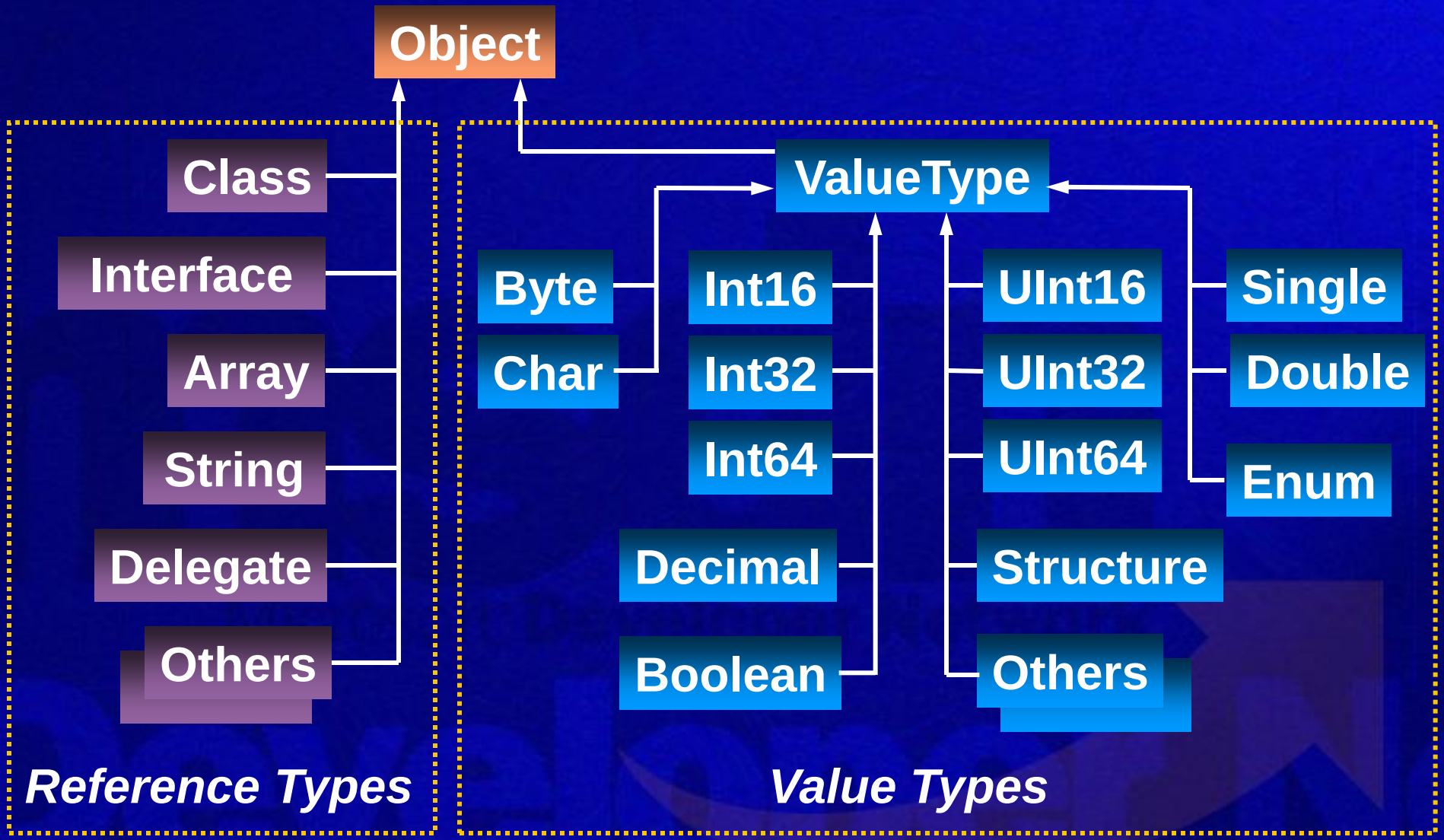
Common Language Runtime
Metadata in an Assembly



Common Language Runtime
Applications

- One or more assemblies
- Assemblies resolution
  - Using metadata
    - Local (preferred)
    - Assembly Global Cache
- Different applications may use different versions of an assembly
  - Easier software updates
  - Easier software removal

CLR-Defined Types



.NET Language Support

- The Runtime is language neutral
  - All .NET languages are first class players
  - You can leverage your existing skills
- Common language specification
  - Any language can use and extend the .NET Framework
  - All languages are interoperable
- Microsoft is providing:
  - Visual Basic, C++, C#, JScript™
- Third-parties are building:
  - APL, COBOL, Pascal, Eiffel, Haskell, ML, Oberon, Perl, Python, Scheme, Smalltalk, Objective CAML, …

The .NET Framework
Web Services

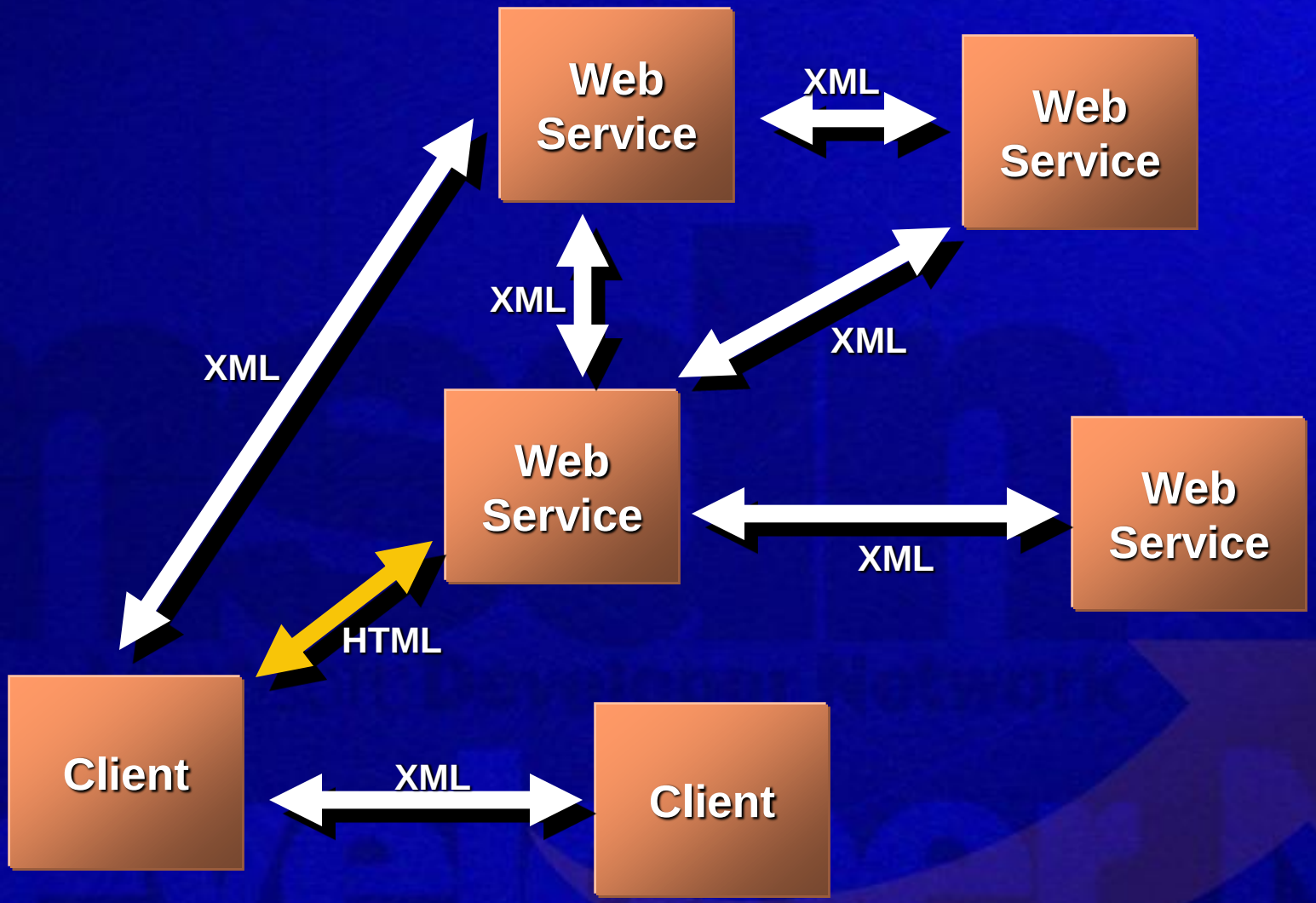
- The center of the .NET architecture
- Technical definition
  - “A programmable application component accessible via standard Web protocols”
- Expose functionality from Web Sites
  - Almost like component programming over the Web

Building User Interfaces
Web Services

- Technical definition – “A programmable application component accessible via standard Web protocols”
  - Built on SOAP
- Expose functionality from Web Sites
  - Almost like component programming over the Web
  - Functionality exposed using XML/HTML
- Standard Web Services include
  - Storage.NET Service
  - Calendar
  - MSN Passport

The .NET Framework

Web Services



More Resources
Find It on the Internet

- http://msdn.microsoft.com
- http://www.microsoft.com/net
- http://msdn.microsoft.com/xml
- msnews.microsoft.com
  - microsoft.public.dotnet.general
  - microsoft.public.dotnet.xml