The Learning House
427 S 4th Street #300
Louisville KY 40202

Introduction to .NET

.NET Cohort

Windows Development Before .NET

- Fragmentation of development stacks (VB, C, C++ | Win32 API, MFC, COM, etc)
- Inconsistency between stacks, language constructs, a lot of time spent on “plumbing” tasks like memory management, etc.
- Not designed for internet development

Goals of .NET

Execution Environment

- Security
- Multiple platform support (desktop OS, server OS, mobile OS, etc.)
- Performance

Development Environment

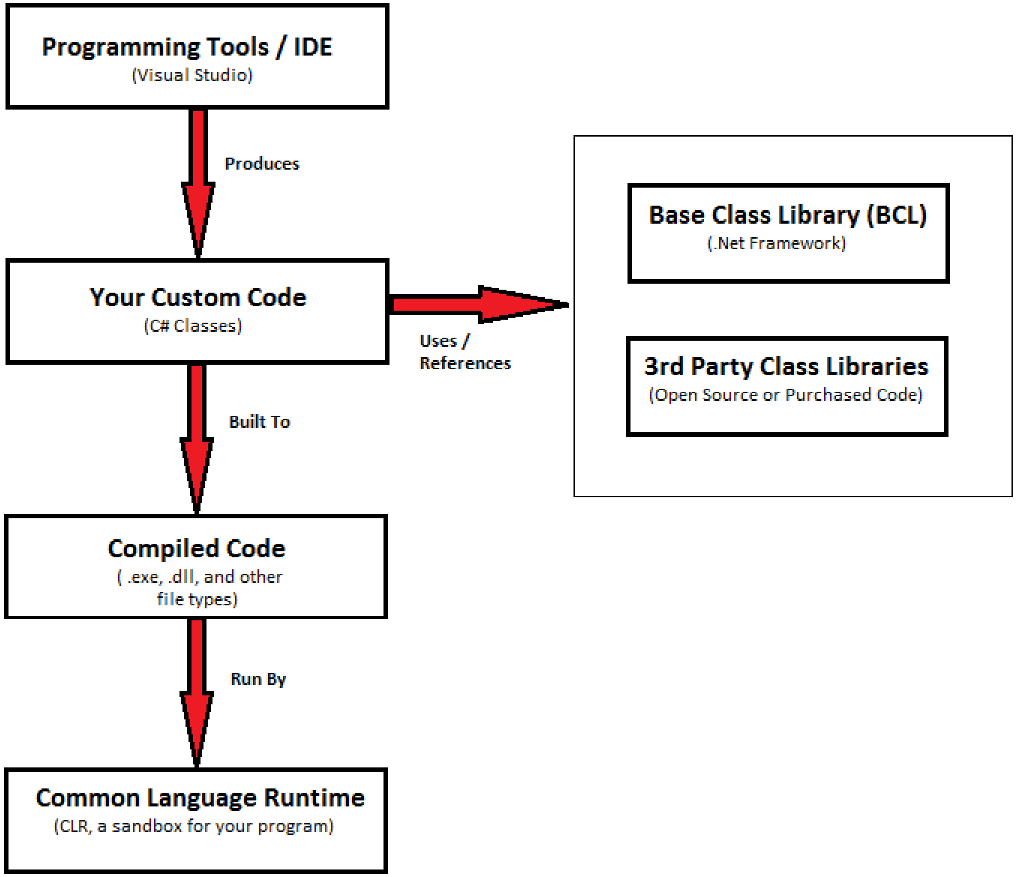
- Object-oriented
- Consistent programming experience
- Industry standard support (HTTP, XML, SOAP, JSON, WSDL)
- Language independence
- Interoperability

Components of .NET

- Programming Tools
  - Visual Studio, other IDEs (Integrated Development Environment) and debuggers
  - .NET compilers (C#, VB, F#, Iron Ruby, C++, etc.)
  - Server-side tech (ASP.NET, WCF, etc.)
- Base Class Library (BCL/FCL)
  - Built-in classes that expose common tasks methods (working with files, encryption, security, etc.)
- Common Language Runtime (CLR)
  - Memory management, garbage collection, code safety, exception handling, thread management

SOFTWARE GUILD

Overview: How .NET Apps Are Created



Programming Tools / IDE
(Visual Studio)

Produces

Your Custom Code
(C# Classes)

Uses / References

Base Class Library (BCL)
(.Net Framework)

3rd Party Class Libraries
(Open Source or Purchased Code)

Built To

Compiled Code
(.exe, .dll, and other file types)

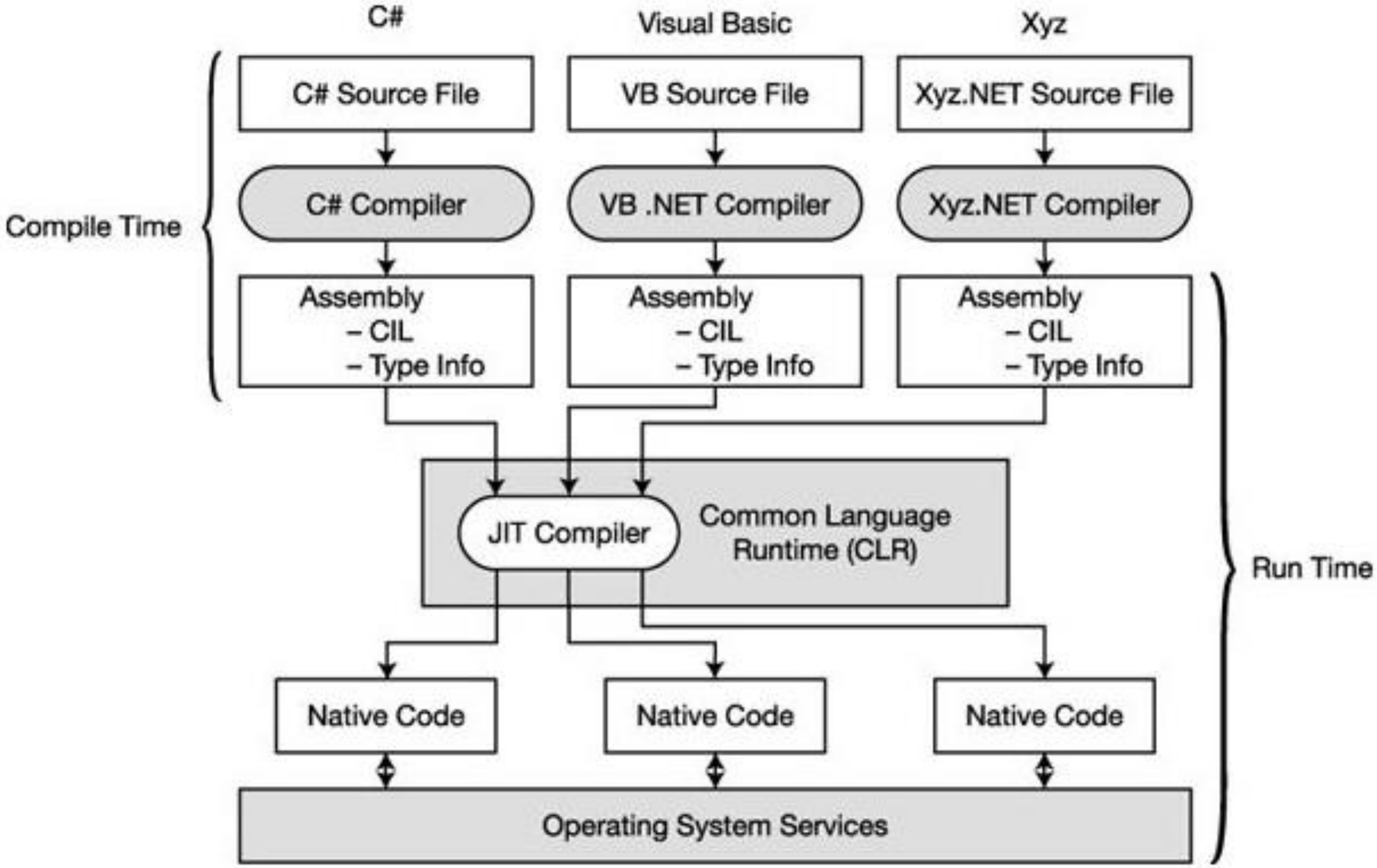
Run By

Common Language Runtime
(CLR, a sandbox for your program)

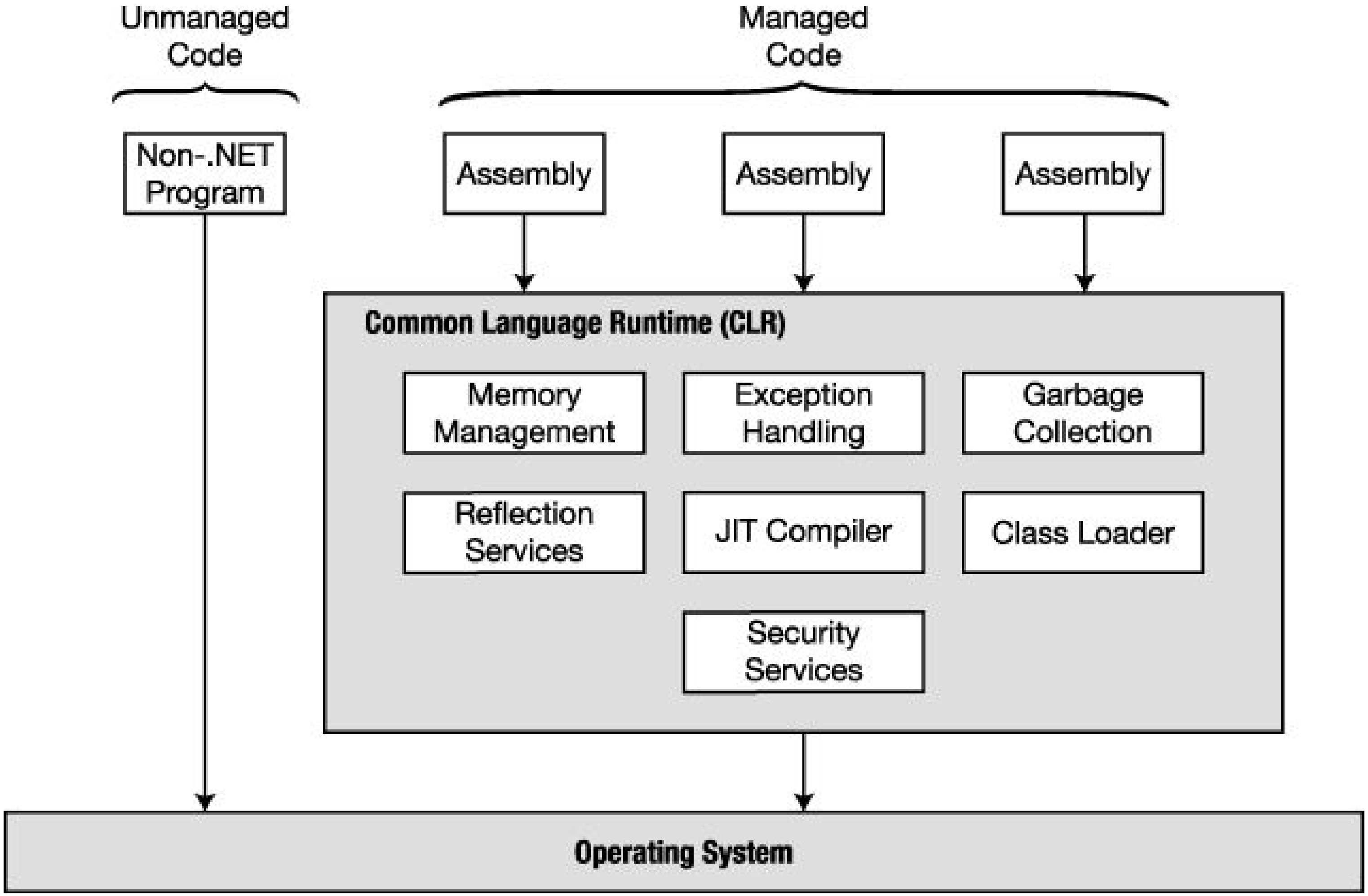
Additional Perks

- Automatic garbage collection
- Interoperability
  - Mix .NET language assemblies, pInvoke, and COM support
- Simplified deployment
- Type safety
- Rich framework library built-in

Detailed Overview of Application



The CLR and Why It's Awesome



CTS / CLI

- Common Type System and Common Language Infrastructure are ECMA open standards
- Contains all the type and data information to ensure that any .NET language compiles to the same IL

In Conclusion…

- The .NET Framework supports many languages besides C# and VB.NET.
- Applications and assemblies are compiled into the Intermediate Language and converted into native code at run time.
- Any computer or device running the .NET Framework version you target can run your code.
- .NET supports calls into unmanaged code and can interact directly with low-level APIs.

SOFTWARE GUILD