

Copyright © 2014 by Software Craftsmanship Guild.

All rights reserved. No part of these materials may be reproduced, distributed, or transmitted in any form or by any means, including photocopying, recording, or other electronic or mechanical methods, without the prior written permission of the Software Craftsmanship Guild. For permission requests, write to the Software Craftsmanship Guild, addressed “Attention: Permissions Coordinator,” at the address below.

Software Craftsmanship Guild

526 S. Main St, Suite 609

Akron, OH 44311

# Introduction to .NET

Software Craftsmanship Guild

# 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.
- Wasn’t designed for internet development

# Goals of .NET

## Execution Environment

- Security
- Multiple Platform Support (desktop os, server os, phone os, etc)
- Performance

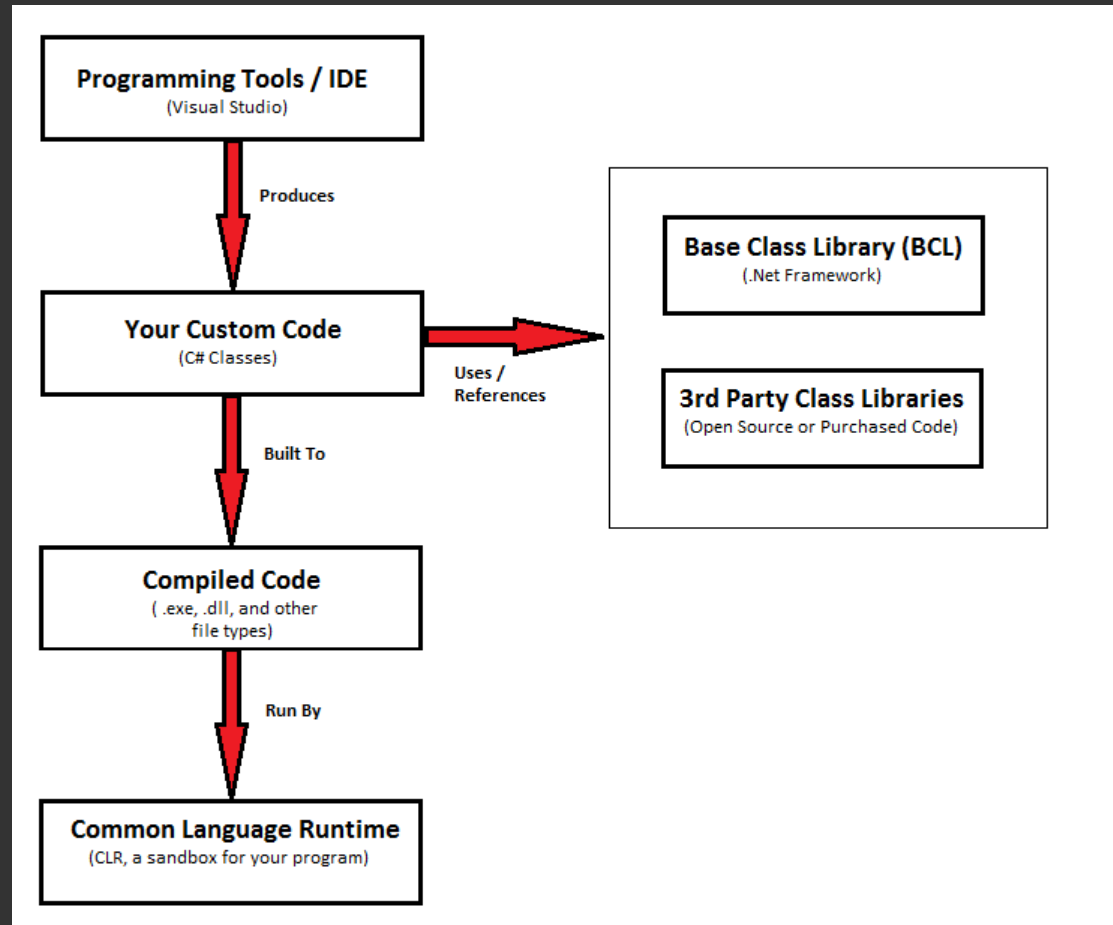
## Development Environment

- 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

# Overview: How .NET Apps Are Created

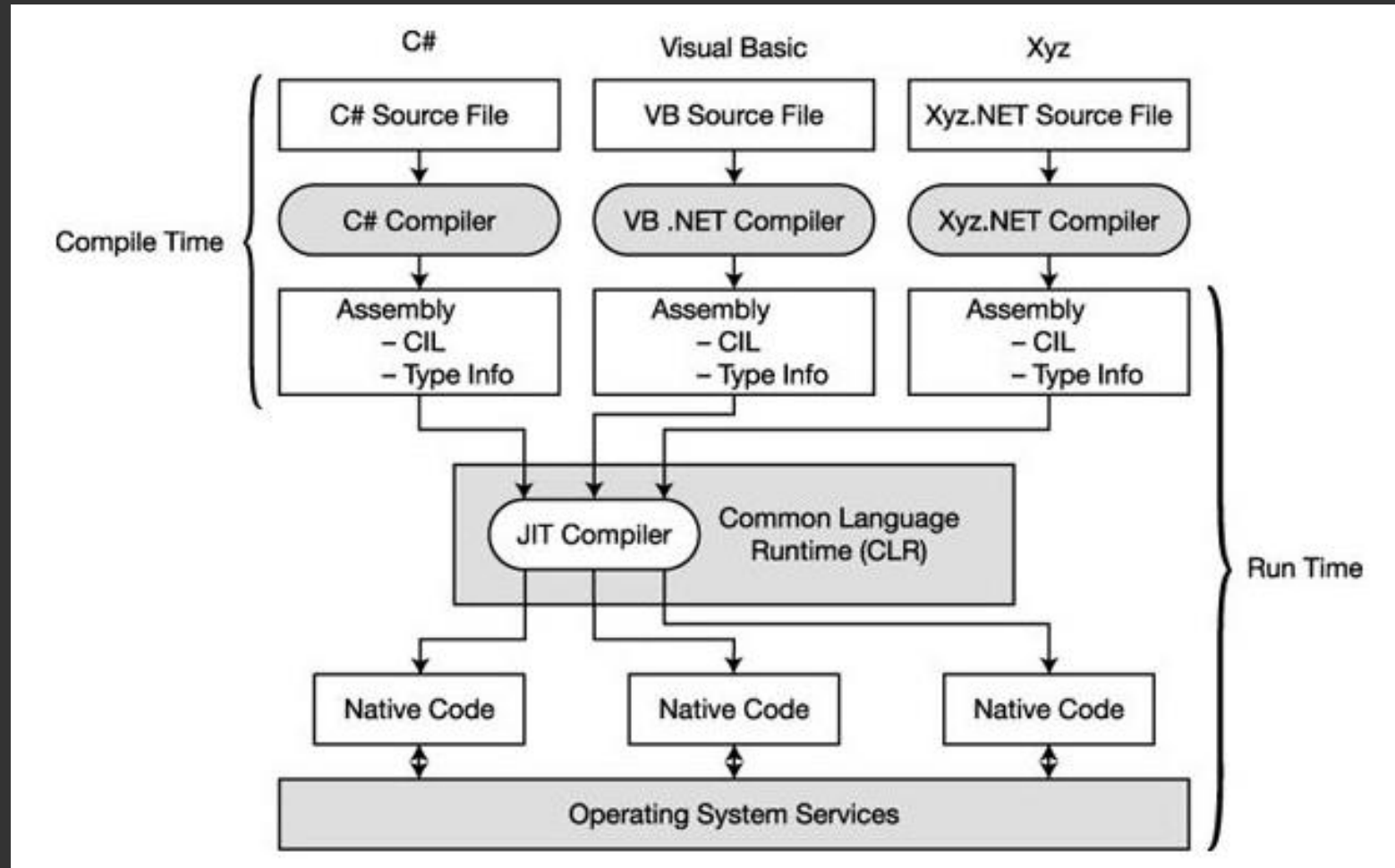


**SOFTWARE**  
CRAFTSMANSHIP GUILD

# Additional Perks

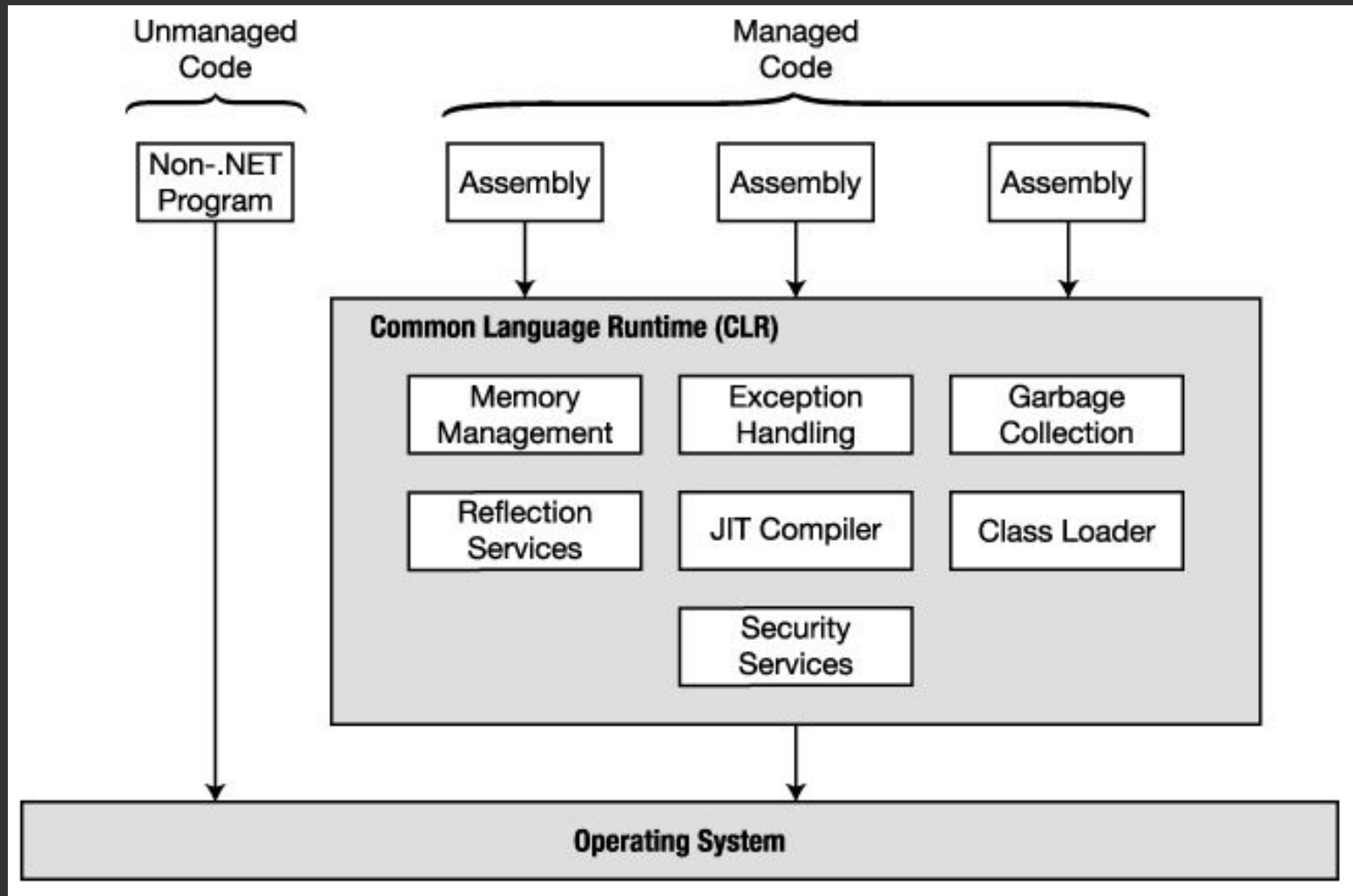
- Automatic Garbage Collection
- Interoperability
  - Mix .NET Language Assemblies, plnvoke, and COM support
- Simplified Deployment
- Type Safety
- Rich Framework Library Built-In

# Detailed Overview of Application





# The CLR and Why It's Awesome



**SOFTWARE**  
CRAFTSMANSHIP GUILD

# 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 at run-time converted to native code
- 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