VT Academy
Thinking Ahead, Beyond Boundaries

Object Oriented Programming

Getting Started with C#

Session 1

Objectives

- Overview .NET Framework and .NET Core
- Introduction to C#
- Setup environment and run the first program
- Structure of C# program

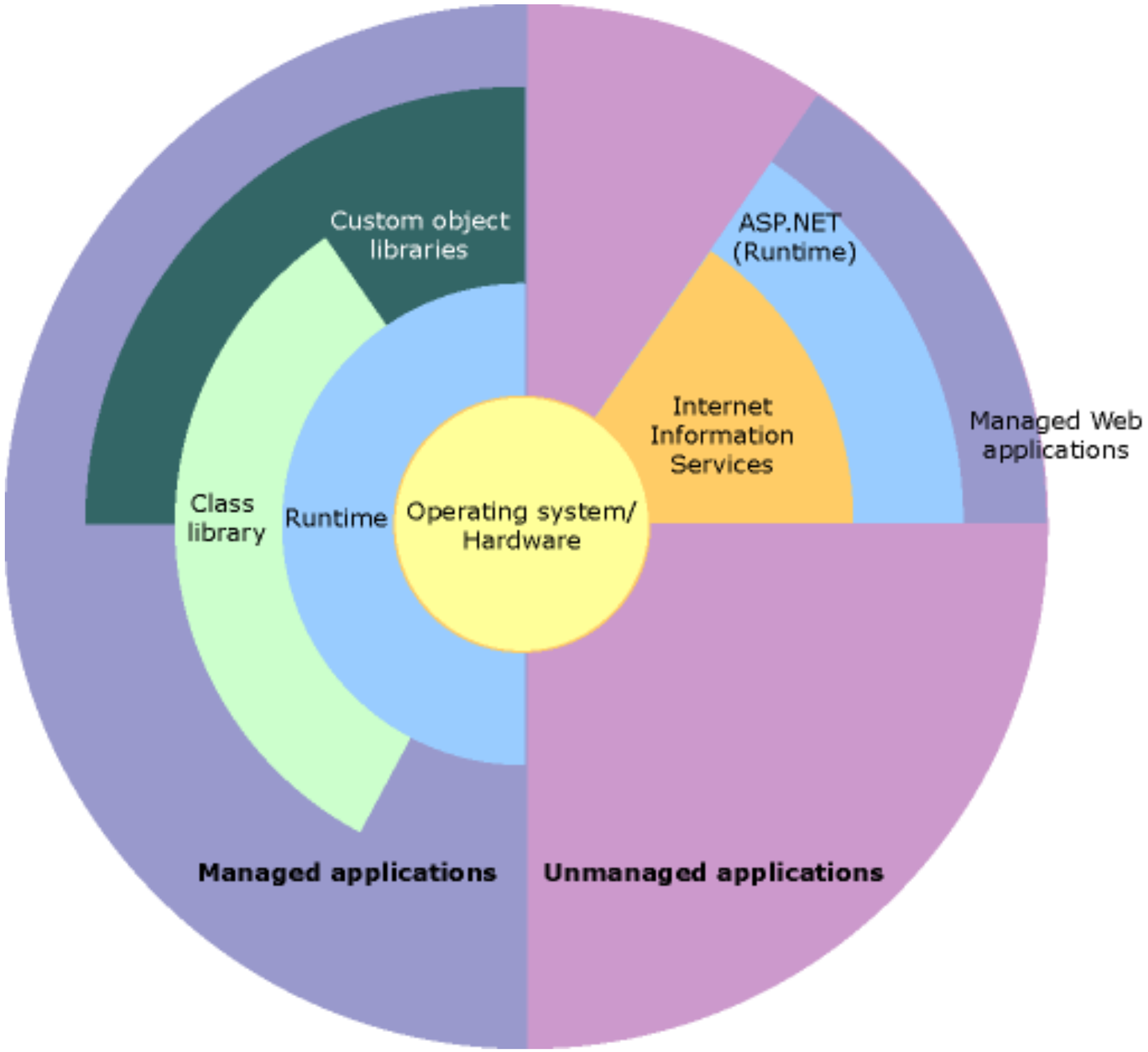
.NET Framework

- The .NET Framework is a technology that supports building and running the next generation of applications and XML Web services
- The .NET Framework is designed to fulfill the following objectives:
  - To provide a consistent object-oriented programming environment whether object code is stored and executed locally, executed locally but Internet-distributed, or executed remotely
  - To provide a code-execution environment that minimizes software deployment and versioning conflicts
  - To provide a code-execution environment that promotes safe execution of code, including code created by an unknown or semi-trusted third party

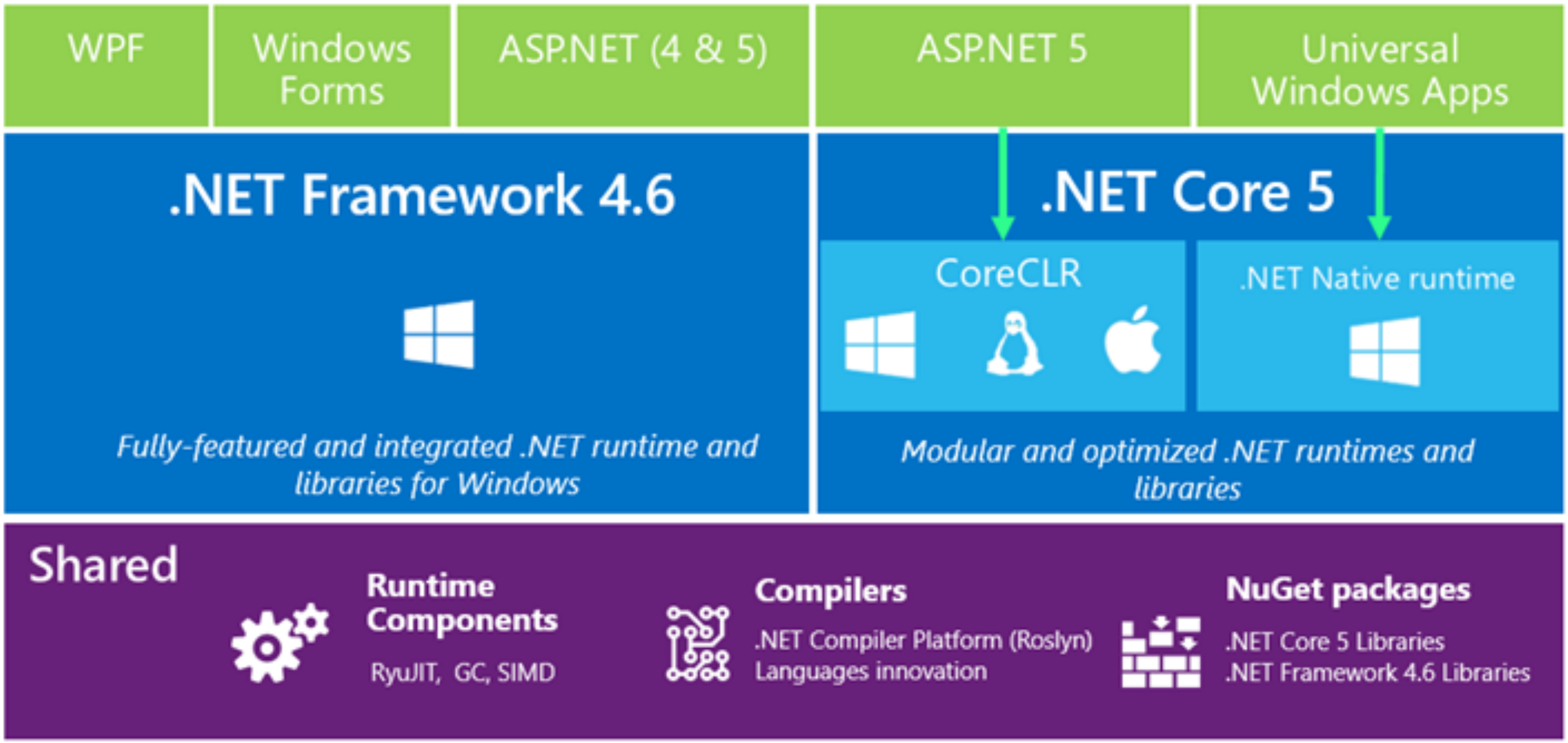
.NET Framework

- To provide a code-execution environment that eliminates the performance problems of scripted or interpreted environments
- To make the developer experience consistent across widely varying types of applications, such as Windows-based applications and Web-based applications
- To build all communication on industry standards to ensure that code based on the .NET Framework can integrate with any other code

.NET Framework



.NET Framework

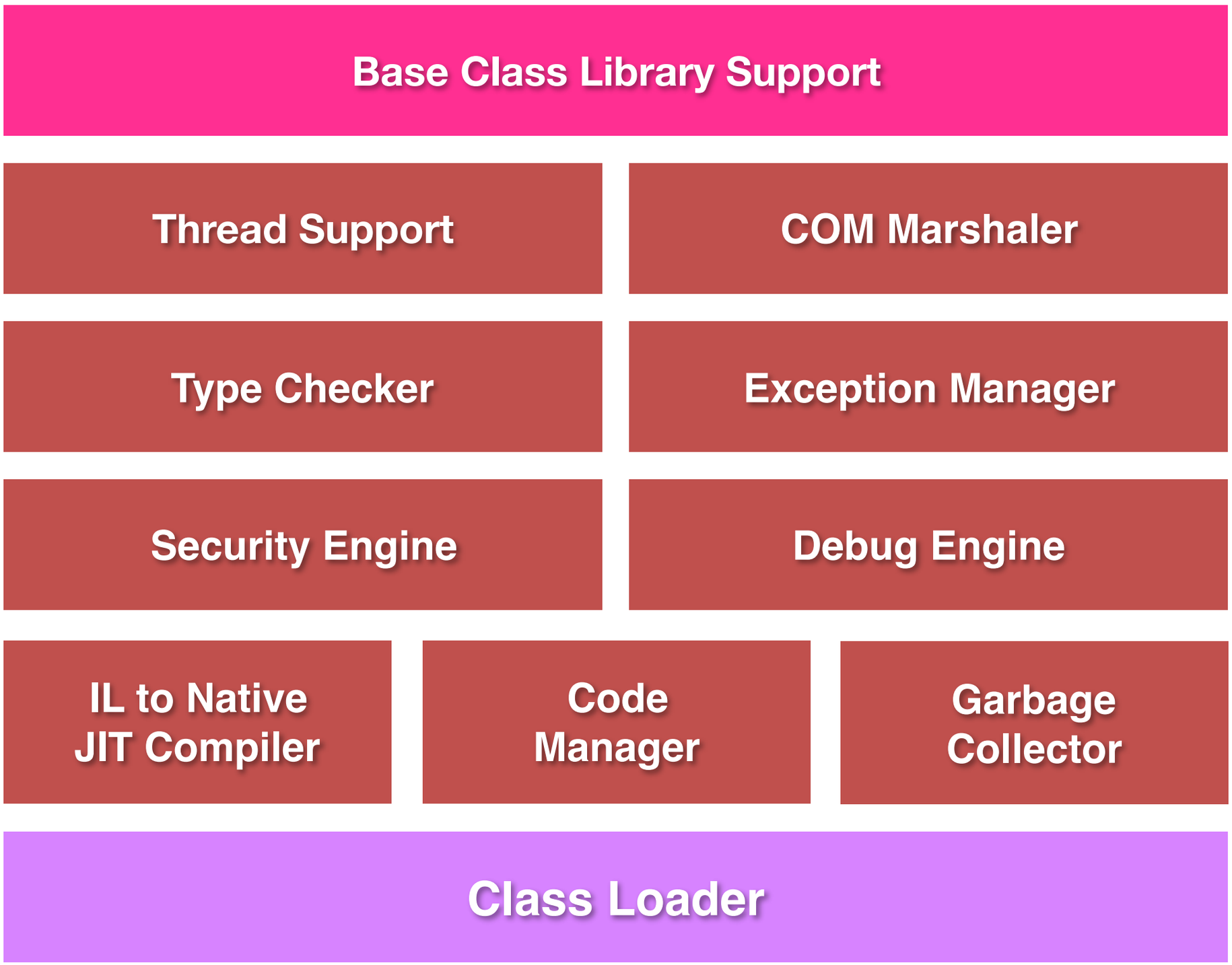


Common Language Runtime (CLR)

- All versions of the .NET framework include CLR
- CLR implements the Virtual Execution System (VES) as defined in the Common Language Infrastructure (CLI) standard, initially developed by Microsoft itself
- A public standard defines the Common Language Infrastructure specification

CLR Architecture

VTC Academy
Thinking Ahead, Beyond Boundaries



CLR Runtime Benefits

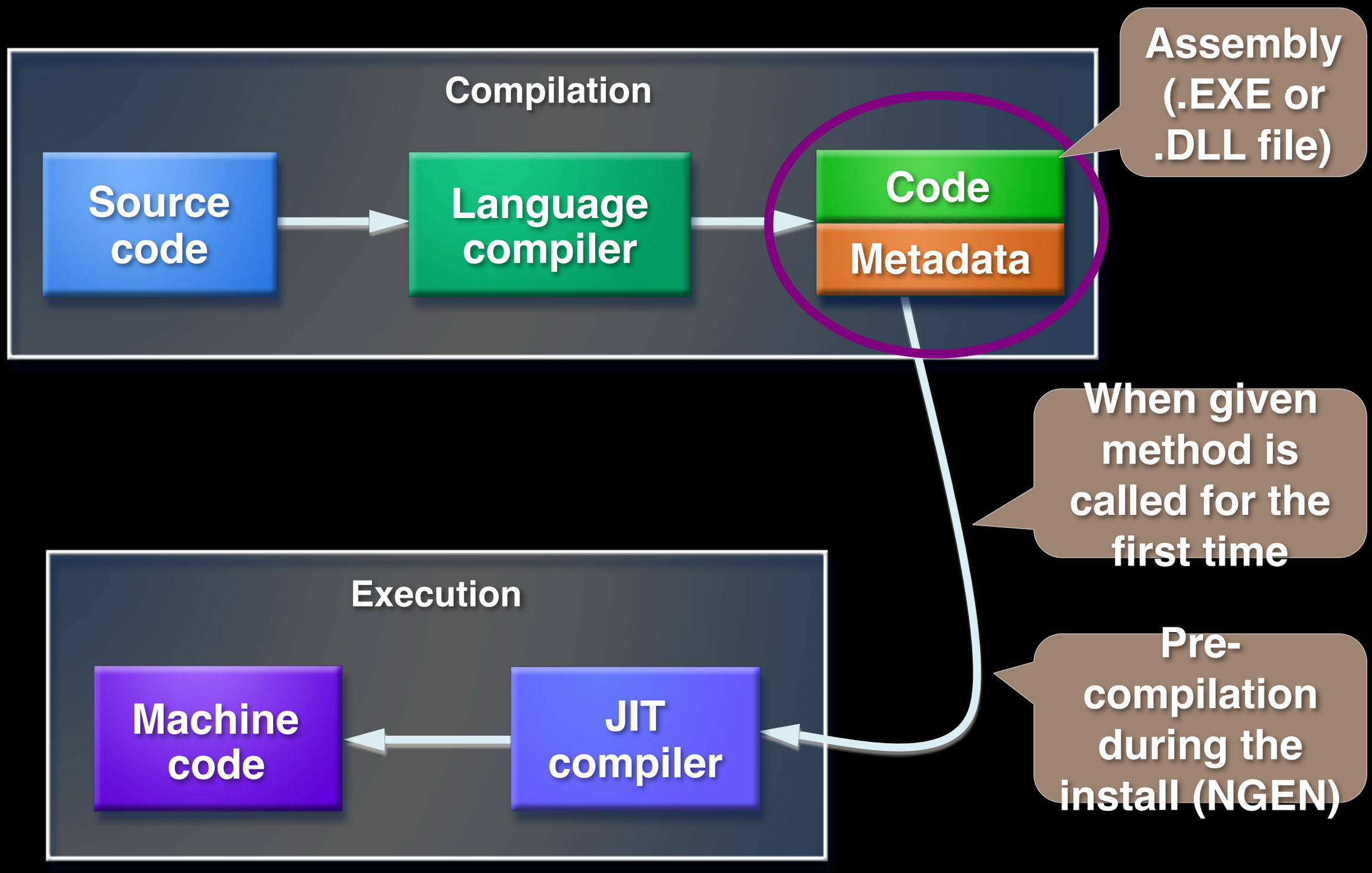
- Performance improvements
- The ability to easily use components developed in other languages
- Extensible types provided by a class library
- Language features such as inheritance, interfaces, and overloading for OOP
- Support for explicit free threading that allows creation of multithreaded, scalable apps
- Support for structured exception handling
- Support for custom attributes
- Garbage collection
- Use of delegates instead of function pointers for increased type safety and security

Intermediate Language (MSIL, IL, CIL)

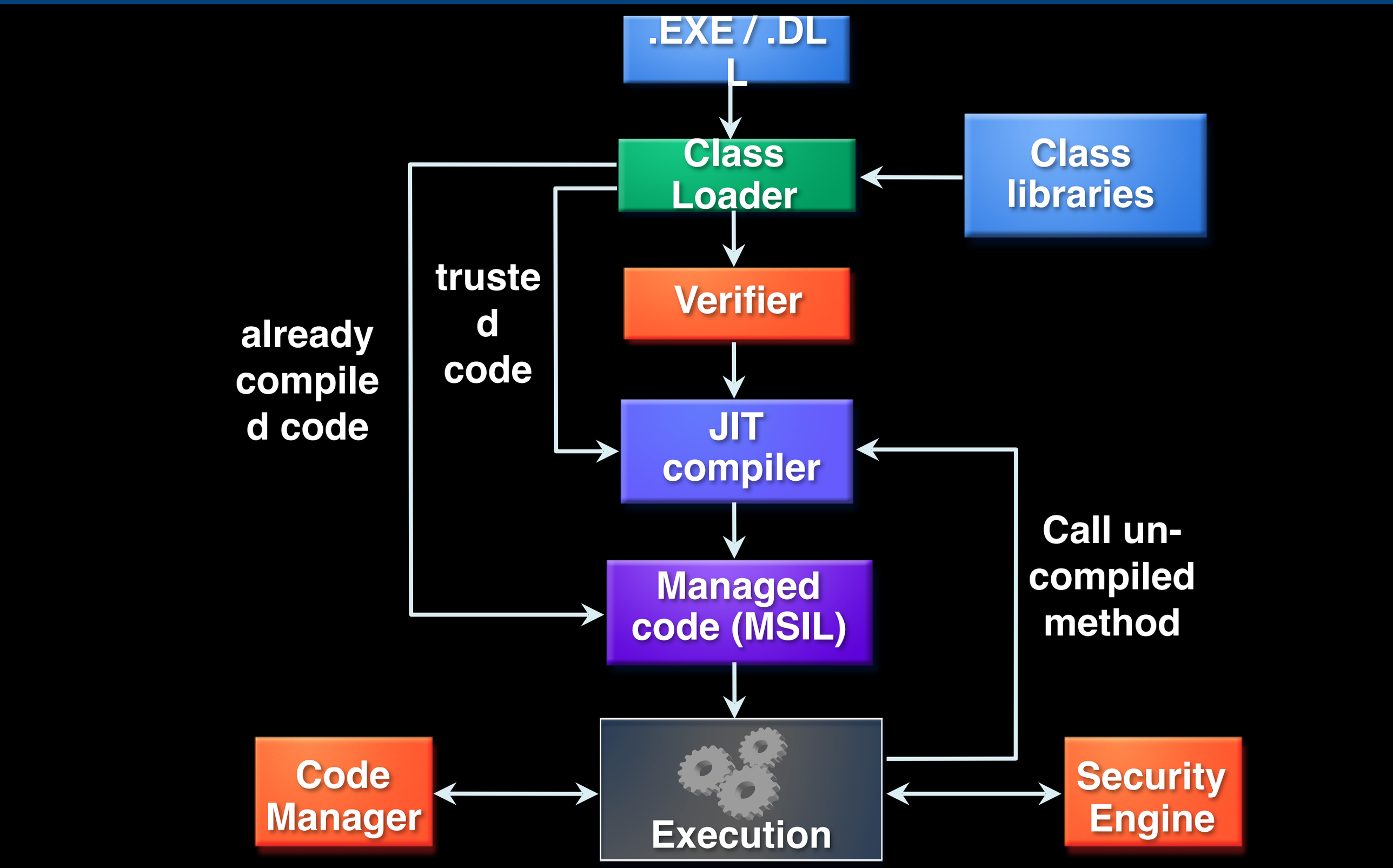
VTC Academy
Thinking Ahead, Beyond Boundaries

- Low level language (machine language) for the .NET CLR
- Has independent set of CPU instructions
  - Loading and storing data, calling methods
  - Arithmetic and logical operations
  - Exception handling …
- MSIL is converted to instructions for the current physical CPU by the JIT compiler

Compilation and Execution



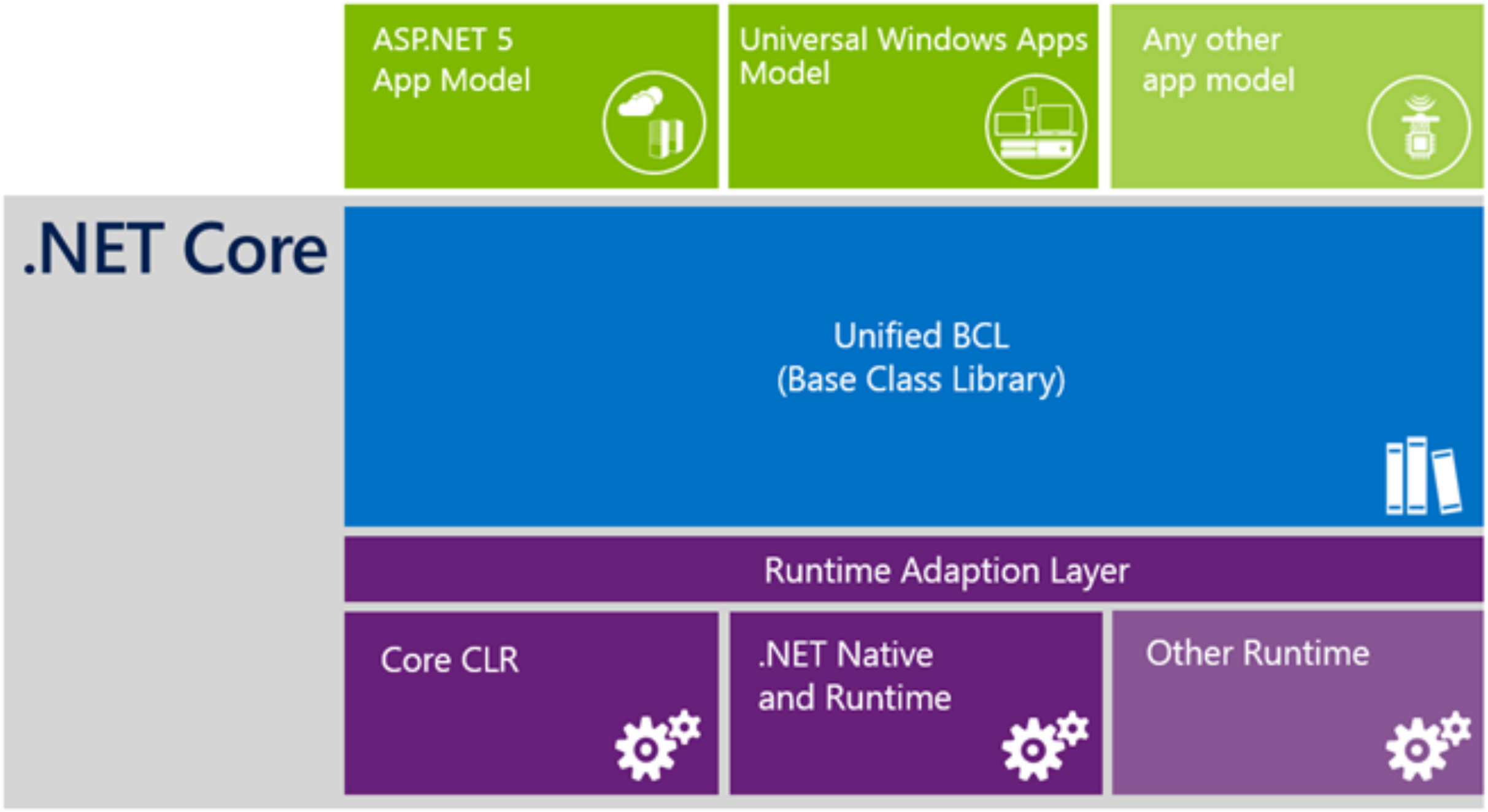
How CLR Execute MSIL



.NET Core

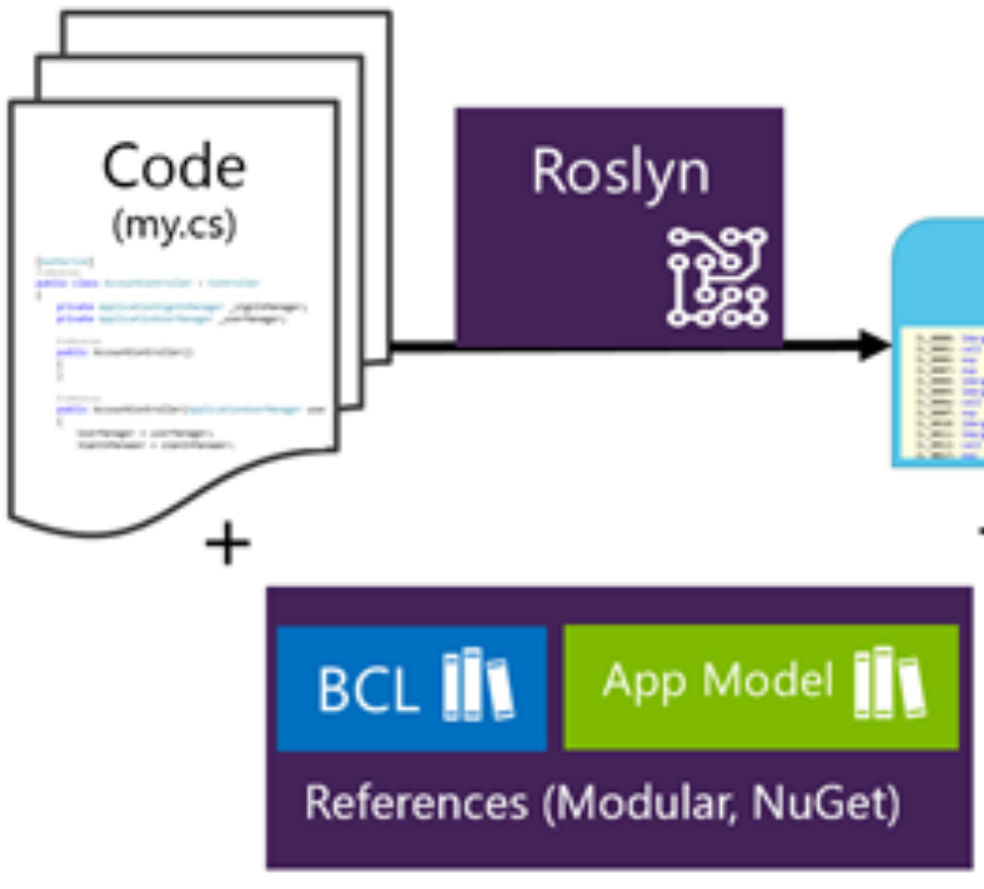
- .NET Core is something new in Microsoft technology families. In short .NET Core is cloud-optimized, cross-platform port of the .NET Framework described above (modern version of .NET Framework)
- The main target of .NET Core is to be accesible from three main operating systems – macOS, Linux & Windows
- .NET Core is small, optimized runtime that can be targeted by ASP.NET 5 Application
- It is a modular runtime & library implementation that includes a subset of the .NET Framework
- Libraries called as Core FX & runtime called as CoreCLR
- One of key benefits of .NET Core is its portability

.NET Core


ASP.NET 5
App Model
Universal Windows Apps
Model
Any other
app model
.NET Core
Unified BCL
(Base Class Library)
Runtime Adaption Layer
Core CLR
.NET Native
and Runtime
Other Runtime

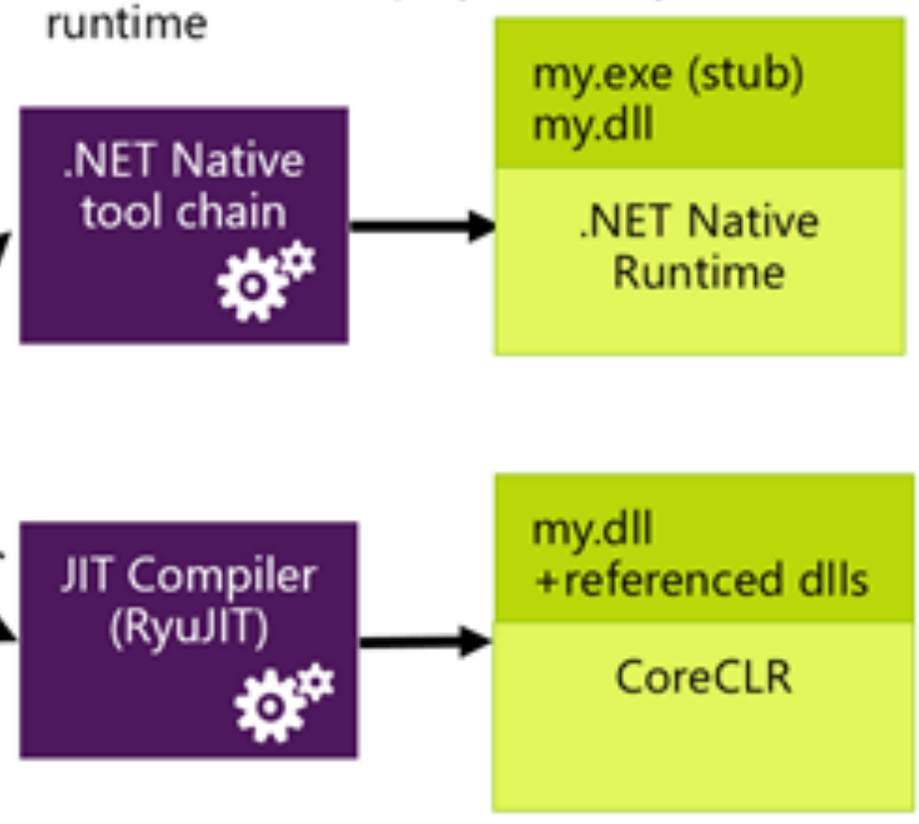
Code / Build / Debug

Roslyn takes your code and compiles it to IL. You have very modular references to the BCL and App Model you're targeting.



Deploy & Run

References are built with your app into one native dll deployed locally with runtime



References & CoreCLR are deployed with app locally, JIT compilation on start up

.NET Core Runtime (CoreCLR)

- The CoreCLR repo contains the complete runtime implementation for .NET Core
- It includes RyuJIT, the .NET GC, native interop and many other components. It is cross-platform, with multiple OS and CPU ports in progress
- Note that the library implementation CoreFX (System.Collections, System.IO, System.Xml and so on) lives in another repo dotnet/corefx.

.NET Core Libraries (CoreFX)

VTC Academy
Thinking Ahead, Beyond Boundaries

- The CoreFX repo contains the library implementation (called "CoreFX") for .NET Core
- It includes System.Collections, System.IO, System.Xml, and many other components
- The corresponding .NET Core Runtime repo contains the runtime implementation (called "CoreCLR") for .NET Core. It includes RyuJIT, the .NET GC, and many other components
- Runtime-specific library code - namely mscorlib - lives in the CoreCLR repo
- It needs to be built and versioned in tandem with the runtime
- The rest of CoreFX is agnostic of runtime-implementation and can be run on any compatible .NET runtime.

.NET Programming Language

- .NET languages by Microsoft
  - C#, VB.NET, Managed C++, J#, F#, JScript
- .NET languages by third parties
  - Object Pascal, Perl, Python, COBOL, Haskell, Oberon, Scheme, Smalltalk…
- Different languages can be mixed in a single application
- Cross-language inheritance of types and exception handling

C# Programming Language

- C# is an elegant and type-safe object-oriented language (OOP) that enables developers to build a variety of secure and robust apps that run on the .NET Framework
- You can use C# to create Windows client applications, XML Web services, distributed components, client-server applications, database applications, and much, much more
- Visual C# provides an advanced code editor, convenient user interface designers, integrated debugger, and many other tools to make it easier to develop applications based on the C# language and the .NET Framework

C# Programming Language History

- In January 1999, Anders Hejlsberg formed a team to build a new language at the time called Cool, which stood for “C-like Object Oriented Language”
- July 2000 Professional Developers Conference, the language had been renamed C#

Version	Date	.NET Framework	Visual Studio
C# 1.0	*January 2002*	*.NET Framework 1.0*	*Visual Studio 2002*
C# 2.0	November 2005	.NET Framework 2.0	Visual Studio 2005
C# 3.0	*November 2007*	*.NET Framework 2.0, 3.0, 3.5*	*Visual Studio 2008, 2010*
C# 4.0	April 2010	.NET Framework 4.0	Visual Studio 2010
C# 5.0	*August 2012*	*.NET Framework 4.5*	*Visual Studio 2012, 2013*
C# 6.0	July 2015	.NET Framework 4.6	Visual Studio 2015

Visual Studio Overview

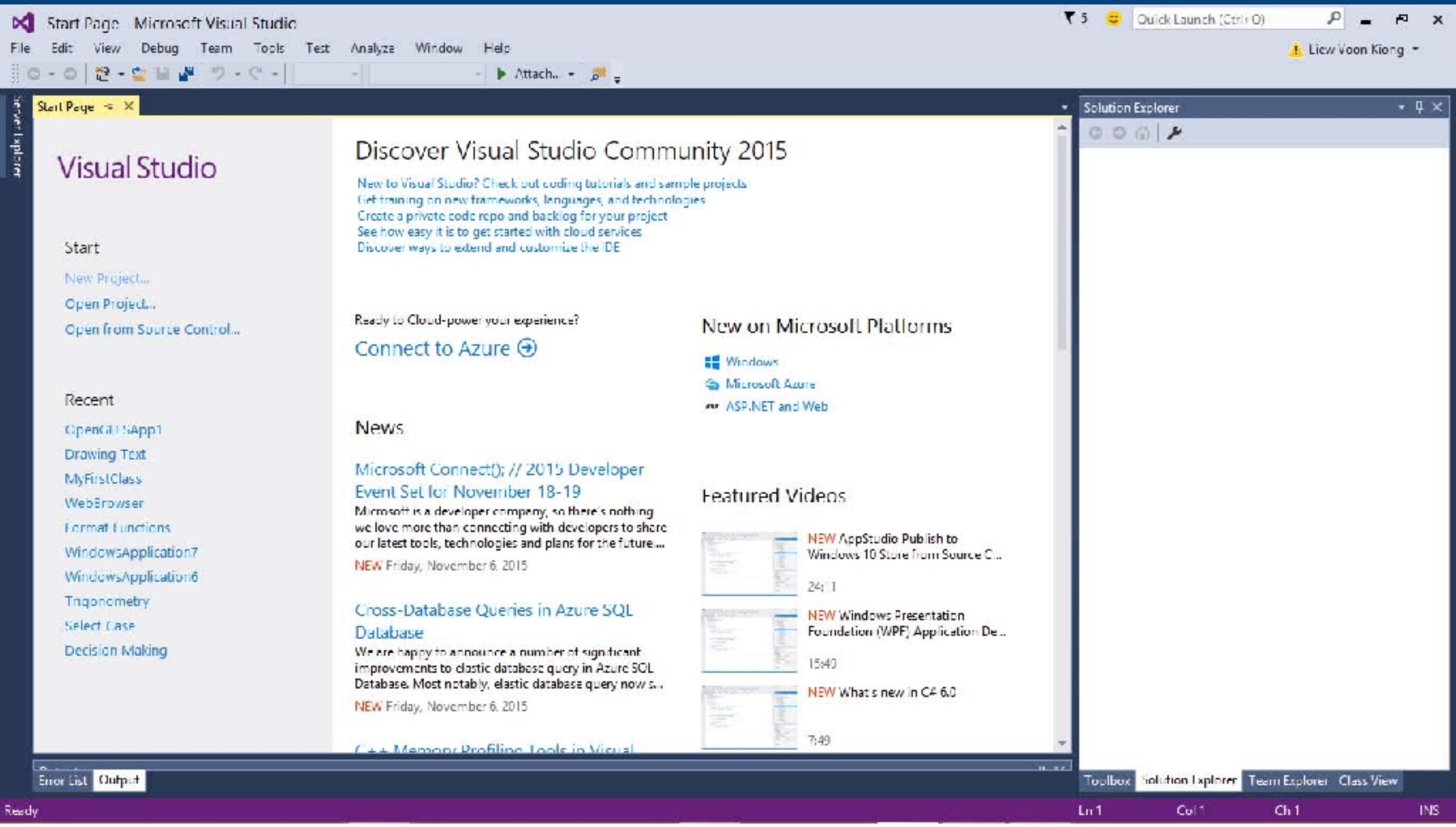
- Visual Studio is a rich, integrated development environment for creating stunning applications for Windows, Android, and iOS, as well as modern web applications and cloud services
  - Tools and services for projects of any size or complexity
  - C#, Visual Basic, F#, C++, Python, Node.js, and HTML/JavaScript
  - Sprint planning
  - Advanced debugging, automated and manual testing, and profiling
  - DevOps with automated deployments and continuous monitoring

Visual Studio Products

- Visual Studio Community
  - Free, full-featured, and extensible tool for developers building non-enterprise applications
- Visual Studio Professional
  - Professional developer tools and services for individual developers or small teams
  - $45 / month
- Visual Studio Enterprise
  - Enterprise grade solution with advanced capabilities for teams working on projects of any size or complexity, including advanced testing and DevOps
  - $250 / month

Visual Studio Community

VTC Academy
Thinking Ahead, Beyond Boundaries



Visual Studio Code

- Free, Open source, Runs everywhere
- Visual Studio Code is a lightweight but powerful source code editor which runs on your desktop and is available for Windows, macOS and Linux.
- Visual Studio Code comes with built-in support for JavaScript, TypeScript and Node.js and has a rich ecosystem of extensions for other languages (such as C++, C#, Python, PHP, Go) and runtimes (such as .NET and Unity).

Visual Studio Code

VTC Academy
Thinking Ahead, Beyond Boundaries

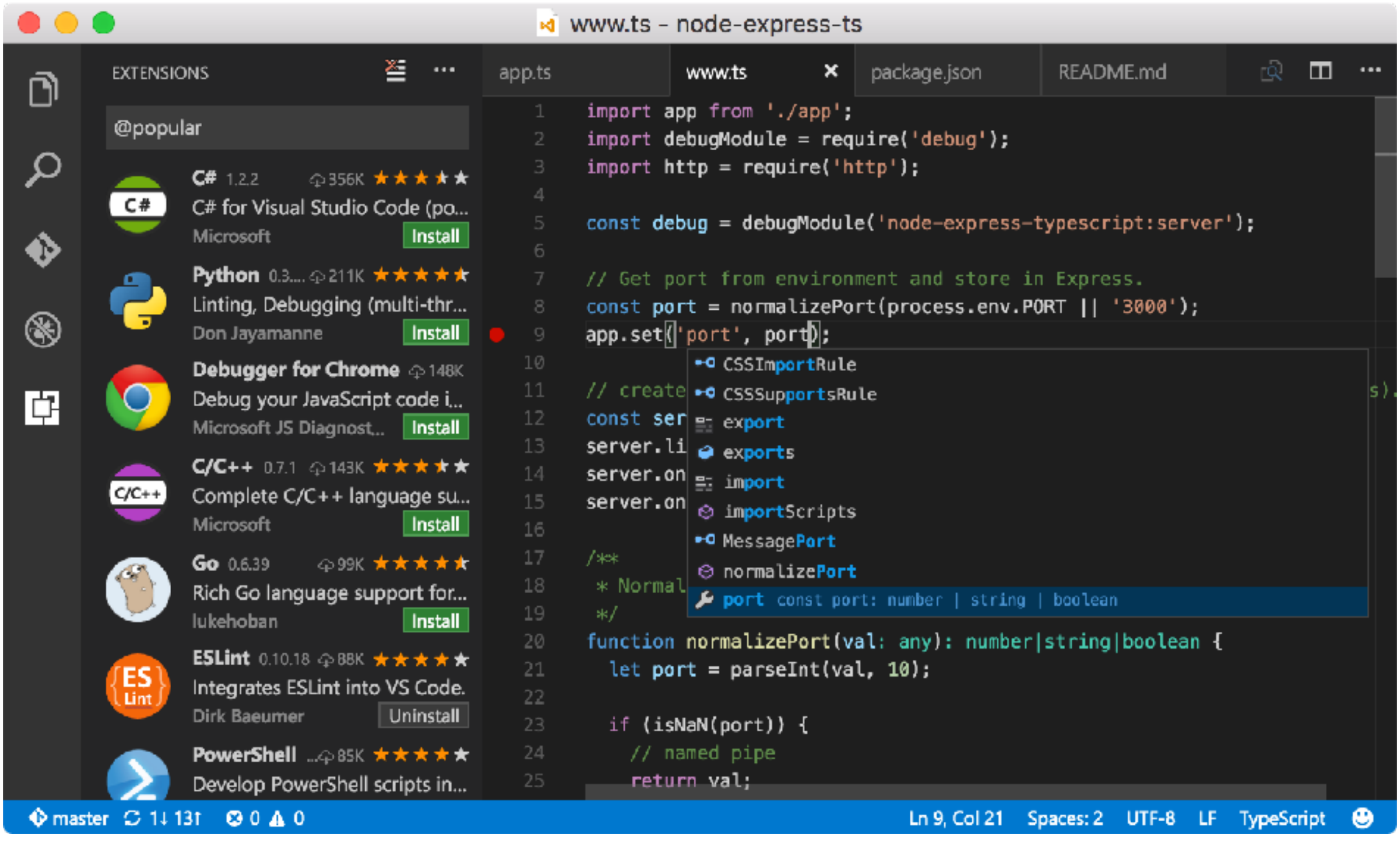
Download Visual Studio Code

Free and open source. Integrated Git, debugging and extensions.

Windows	.deb	.rpm	Mac
Windows 7, 8, 10	Debian, Ubuntu	Red Hat, Fedora, SUSE	macOS 10.9+
.zip | 32 bit versions	.tar.gz | 32 bit versions		

Visual Studio Code

VTC Academy
Thinking Ahead, Beyond Boundaries



The First C# Program

```
using System;

namespace ConsoleApplication
{
    public class Program
    {
        public static void Main(string[] args)
        {
                    // My first program in C#
                    Console.WriteLine("Hello C# Language");
                    Console.Readkey();
        }
    }
}
```

C# Program Structure

- A C# program consists of the following parts:
  - Namespace declaration
  - A class
  - Class methods
  - Class attributes
  - A Main method
  - Statements and Expressions
  - Comments

Comments & XML Documentation

- C# support three types of comments. There are:
  - Single-line comments: //
  - Multi-line comments: /*……..*/
  - XML comments: ///…
- XML source code documentation
- In C#, you can create an XML document that will contain all the XML comments

References

- https://msdn.microsoft.com/en-us/library/zw4w595w(v=vs.110).aspx
- https://msdn.microsoft.com/en-us/library/z1zx9t92.aspx
- https://msdn.microsoft.com/en-us/library/5a4x27ek(v=vs.110).aspx
- https://github.com/dotnet/core/blob/master/roadmap.md
- https://code.visualstudio.com/download

Summary

- The .NET Framework is a technology that supports building and running the next generation of applications and XML Web services
- The .NET Framework is a technology that supports building and running the next generation of applications and XML Web services
- MSIL is converted to instructions for the current physical CPU by the JIT compiler
- .NET Core is something new in Microsoft technology families. In short .NET Core is cloud-optimized, cross-platform port of the .NET Framework described above (modern version of .NET Framework)
- C# is an elegant and type-safe object-oriented language (OOP) that enables developers to build a variety of secure and robust apps that run on the .NET Framework