Programming dotNet

Paolo Burgio
paolo.burgio@unimore.it

UNIMORE
UNIVERSITÀ DEGLI STUDI DI
MODENA E REGGIO EMILIA

High Performance
Real Time Lab


JAVA
He does exactly what I do
C#
But better.
imgflip.com

Before .NET framework (and dotNET core)

Windows GUI development

- Win32 API, MFC, Visual Basic, COM

Web (with MS technologies)

- ASP / ASP.NET

C, C++...

- Embedded

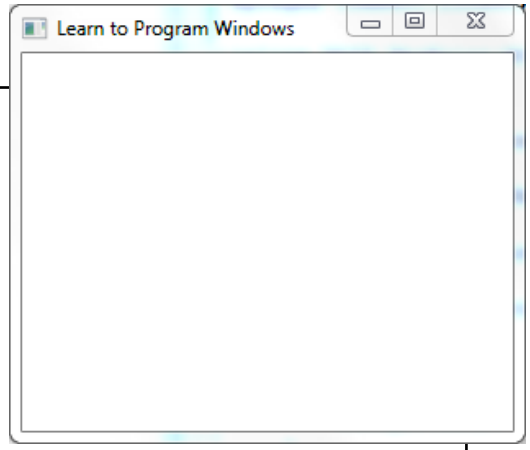
Java

- Nearly everything

Today, also

- Python for prototyping
- Flutter/react/JS derivatives for Web
- RUST for concurrent programming..

Win32 API example




```
#include <windows.h>

int WINAPI wWinMain(HINSTANCE hInstance, HINSTANCE, PWSTR pCmdLine,
                    int nCmdShow) {
  // Register the window class.
  const wchar_t CLASS_NAME[] = L"Sample Window Class";
  WNDCLASS wc = { };
  wc.lpfnWndProc = WindowProc;
  wc.hInstance = hInstance;
  wc.lpszClassName = CLASS_NAME;
  RegisterClass(&wc);
  // Create the window.
  HWND hwnd = CreateWindowEx( 0, // Optional window styles.
                              CLASS_NAME, // Window class
                              L"Learn to Program Windows", // Window text
                              WS_OVERLAPPEDWINDOW, // Window style
                              // Size and position
                              CW_USEDEFAULT, CW_USEDEFAULT, CW_USEDEFAULT, CW_USEDEFAULT,
                              NULL, // Parent window
                              NULL, // Menu
                              hInstance, // Instance handle
                              NULL // Additional application data );
  if (hwnd == NULL) { return 0; }
  ShowWindow(hwnd, nCmdShow);
  // Run the message loop.
  MSG msg = { };
  while (GetMessage(&msg, NULL, 0, 0))
   { TranslateMessage(&msg); DispatchMessage(&msg); }

  return 0;
}
```

C Win32 API

"Traditional SW development", with all its limitations we know

- Explicit memory management, pointers, malloc/free
- Functional/procedural approach, no object-oriented
- Win32 API has 1000s of global functions and datatypes

C++/MFC

C++ is an object-oriented layer built on top of C

› Can mix the two!!

Microsoft Foundation Classes (MFC) is a set of C++ classes to build Win32 applications

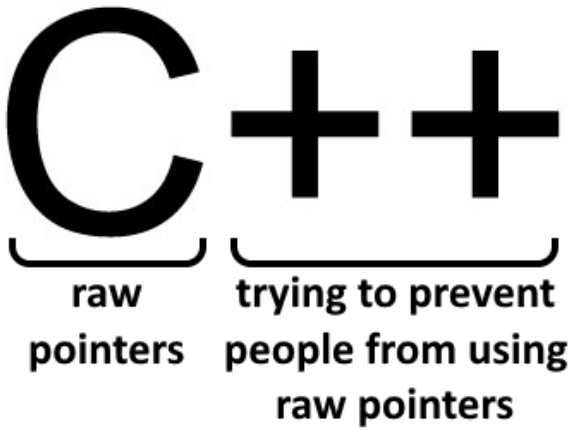
› With a lot of tools, e.g., for code-generation

› GUI?

Greatly helps, but…

› Still, enormous

› Error-prone



Java

Pros

- Solves many issues of C++
- Provides "packages" with helpful classes/functionalities
- Can embed hightly-optimized C code with JNI

Cons

- Limited capability of accessing non-Java APIs
- Little support for real cross-language

Microsoft COM

Application development framework

- Reusable binary code, e.g., between different languages
- C++ can be used in VB6
- C can be used in Delphi

Cons

- Limited language independence
- High degree of complexity
- Active Template Library (ATL) to provide C++ support classes, templates, macros..

.NET, finally

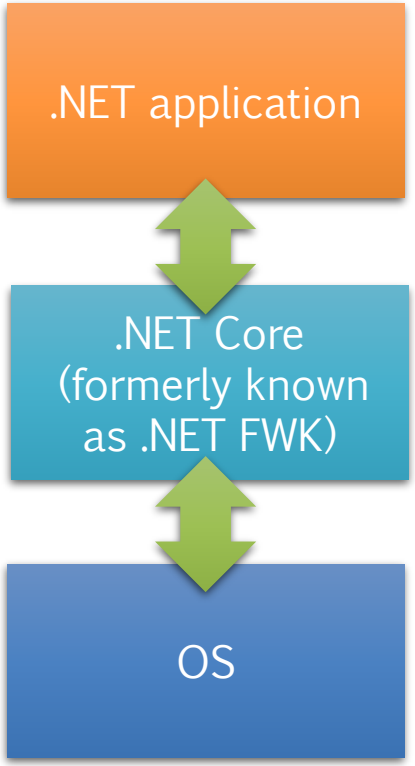
Built "after ~~failures~~ experience"

Key aspects

- OOP
- Runtime environment (MS "JVM")
- GUI
- Web
- Component-based design
- Multi-tier design
- Cross-language interoperability

What is .NET

- A framework
- A programming methodology
- Platform independent
- Language-insensitive



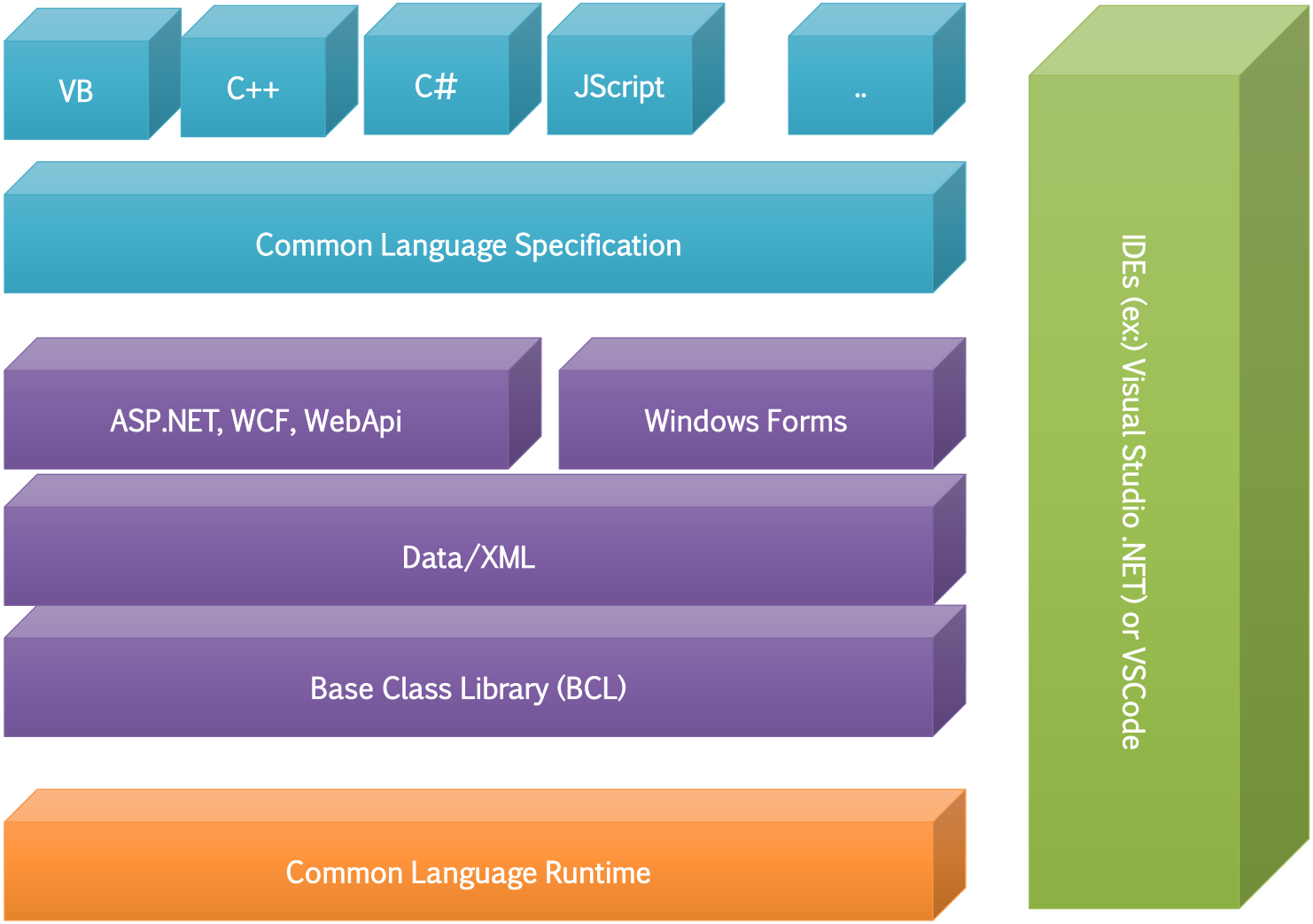
What .NET provides

- An integrated environment
- For internet | desktop | mobile app development
- Object-oriented
- Portable
  - dotNet Core runs on GNU/Linux, and provides shell tools
- Managed (???)

.NET architecture

- Multi-language, cross-platform
- "JVM-like" environment called Common Language Runtime (CLR)
- Framework Class Library (FCL) providing a rich set of classes, templates…
- Just-in-Time conversion between CLR and "native" binary
- .NET components are packaged in assemblies

.NET technical architecture



.NET ecosystem

Language interoperability

- Common Language Specification (CLS)
- Common Language Runtime (CLR)
- Language-specific compilers (C#, VB, Managed C++...)

Frameworks & libraries

- WebAPI, WCF, ASP.NET, Windows Forms, MAUI, Framework Class Libs (FCL)...

Design patterns

- Heavily relies on highly scalable code.
- It's nearly impossible to program it, if you don't adopt SOLID and good

IDE

- Visual Studio, VSCode,...
- Shell

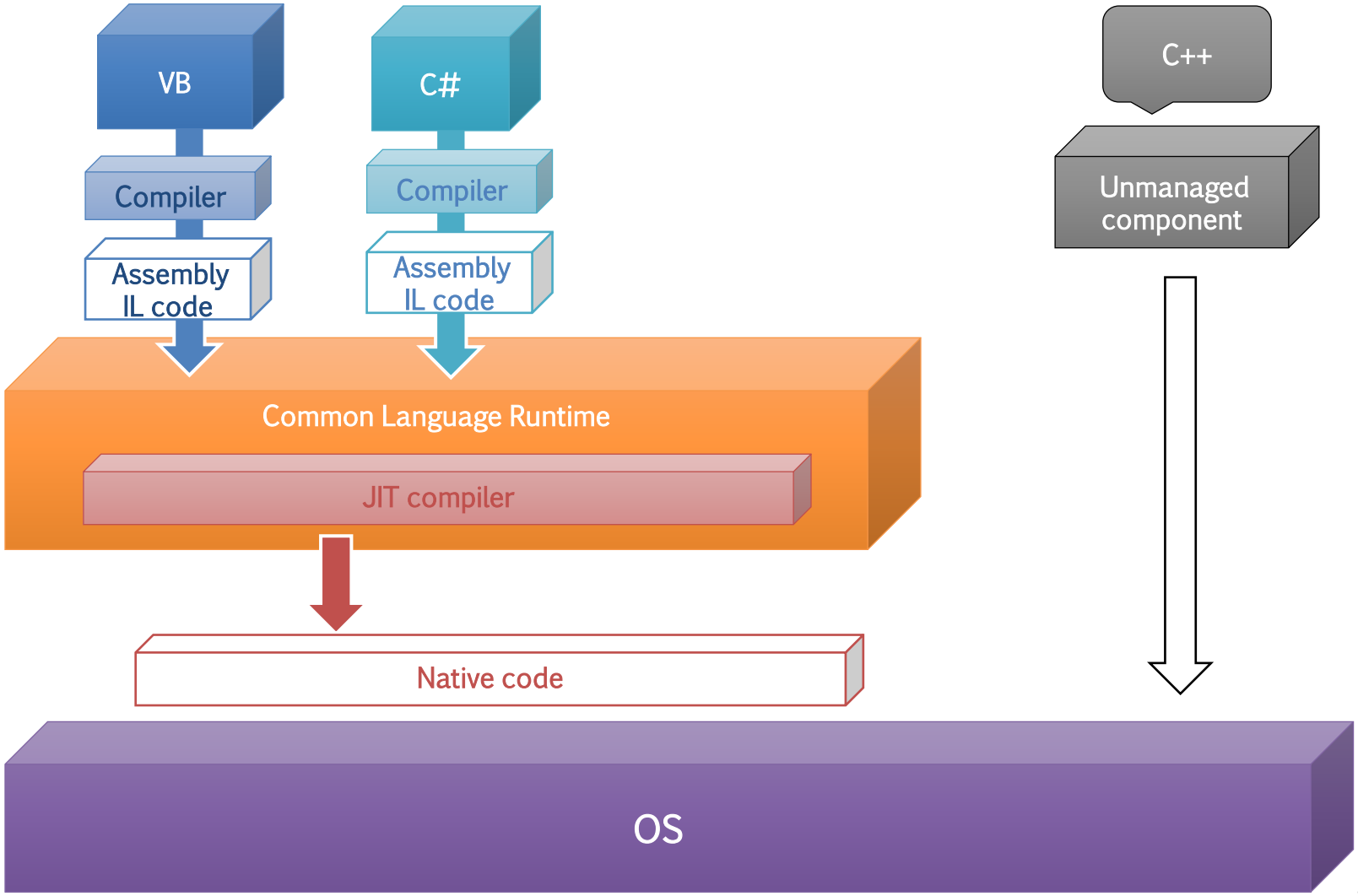
Common Language Runtime

A common runtime for all .NET languages

- Common type system
- Common metadata
- Intermediate Language (IL) to native code compilers
- Memory allocation and garbage collection

CLR Execution Model

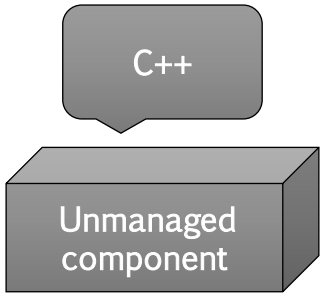
«Toolkit-based» programming



The problem with C++ compliancy

C++ has no features such as those!

- It's a «traditional» language, with syntax for classes/objects
- We call it Unmanaged, it «only» runs on OS + runtime libs
- …remember the *Toolkit* coding paradigm?



However, it is (was) still widely adopted!

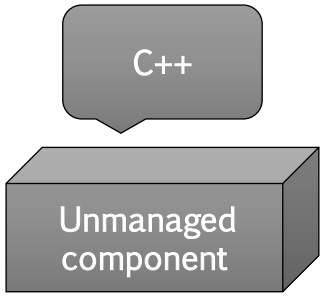
Microsoft introduced Managed C++

- Makes it possible to integrate C++ into the Framework
- They changed the C++ syntax by adding multiple keywords
- Especially with respect to…..what? ☺

The problem with C++ compliancy

C++ has no features such as those!

- It's a «traditional» language, with syntax for classes/objects
- We call it Unmanaged, it «only» runs on OS + runtime libs
- …remember the *Toolkit* coding paradigm?



However, it is (was) still widely adopted!

Microsoft introduced Managed C++

- Makes it possible to integrate C++ into the Framework
- They changed the C++ syntax by adding multiple keywords
- Especially with respect to **pointers and references**

Managed C++ - an example

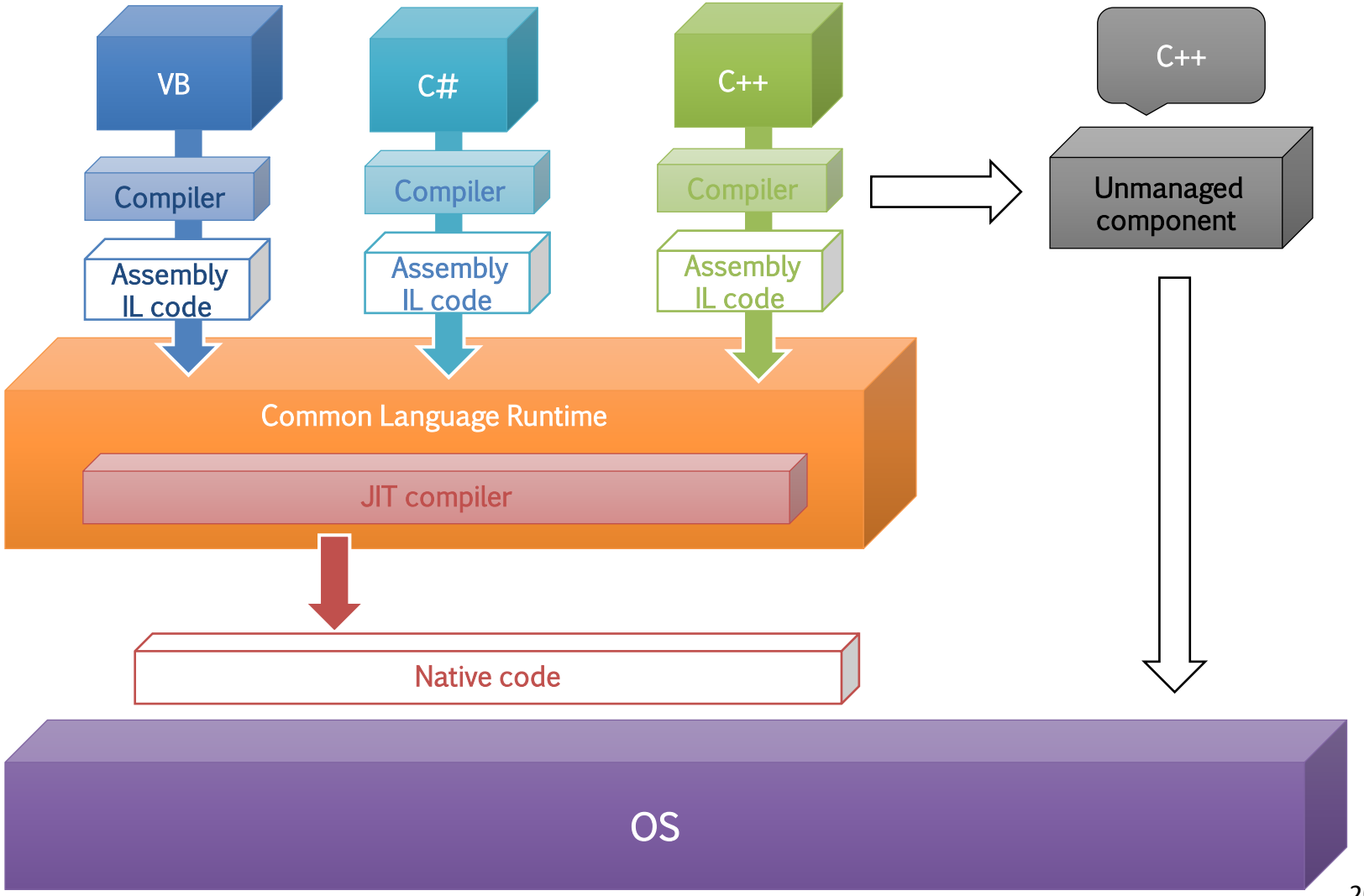
Scenario: Modena Automotive Smart Area

- Where we test&deploy autonomous driving, and smart city
- A windows server, implemented in dotNet: https://github.com/pburgio/Masa.Windows
- High-perf, custom communication protocol: https://github.com/HiPeRT/MASA_protocol

I wanted to include this protocol in the server

- It's C++ based, so, I needed to make it compliant with dotNet via Managed C++
- https://git.hipert.unimore.it/rcavicchioli/masa_protocol
- `feature/dotNet` branch

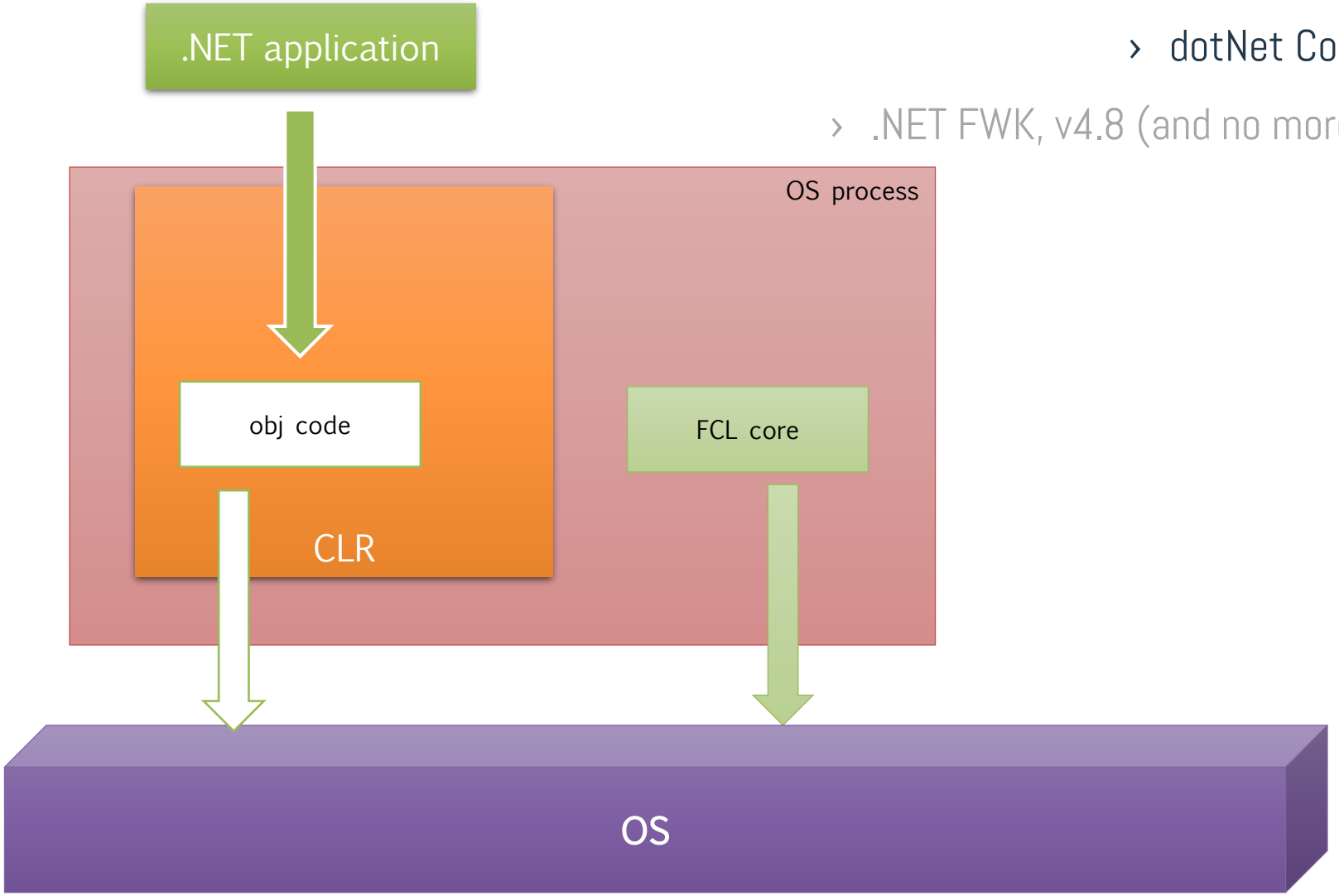
CLR Execution Model



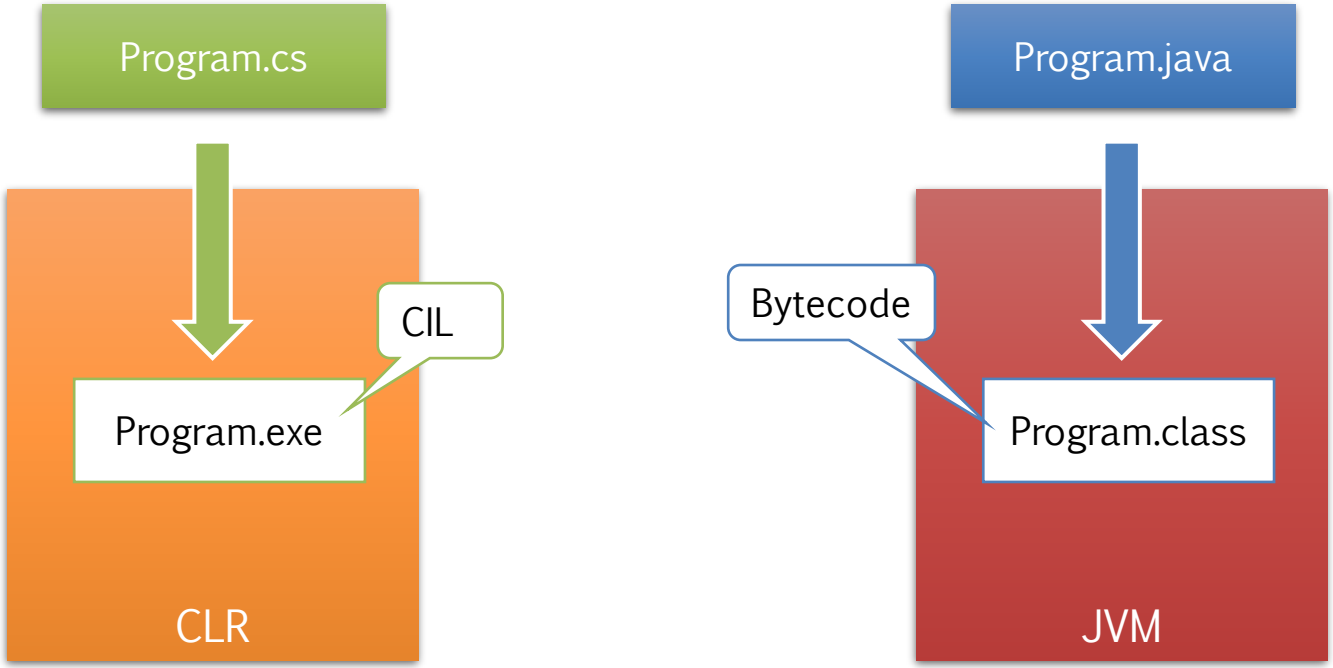
CLR Execution Model

Distributed and open source

- dotNet Core
- .NET FWK, v4.8 (and no more)



Comparison to java



CLR and JIT compilation

"Just in time"

- All .NET languages compile to the same CIL
- The CLR transforms CIL to a given architecture ASM
- Highly-optimized process

Pros

- Support for developer services (debugging)
- Interoperation between un/managed code
- Memory management and garbage collection
- Protected "sandbox" execution
- Access to metadata (type information - reflection)

Base Class Library

- Similar to Java's System namespace
- "Core" subset of FCL used by all .NET applications
- Provides IO, Threading, DB, text, GUI, console, sockets, web…

Java vs .NET platforms

.NET is 100% proprietary (although some components are open-source), while Java is more "open"

- Mono project to provide open .NET platformn, dotNet Core was born!
- Visual Studio Code

In desktop app, Java struggled to establish its presence

- Swing, AWT (Abstract Window Toolkit), SWT (Standard Widget Toolkit)
- Also, Sun failed in promoting it properly as appealing choice for desktop products

.NET is the most popular choice for developing Windows apps

- Visual Studio Express and Community are free-of cost
- Windows doesn't ship with Java JRE; it ships with .NET

Server applications

.NET is highly integrated with Windows Server infrastructure

- E.g., integrated **Security**, Debugging, Deployment…

Server applications

- Java EE vs. ASP.NET (which is however not part of standard CLI)
- Of the top 1,000 websites, approximately 24% use ASP.NET and also 24% use Java, whereas of all the websites approximately 17% use ASP.NET and 3% use Java (http://w3techs.com/technologies/cross/programming_language/ranking)

..and mobile?

Android is based on Java

› It is also possible to develop in C (NDK)

Microsoft recently acquired Xamarin

› Cross-(mobile)platform IDE + libraries for c#

› Multi-platform APP UI «MAUI» - https://github.com/dotnet/maui

› Integrated in VS

C#

See sharp?

A bit of history…

- Born during the development of the .NET Framework, in 1999
- Team lead by Anders Hejlsberg
- Initially called "C-like Object Oriented Language" (Cool)
- When MS announced .NET project, the language had been renamed C#, and the class libraries and ASP.NET runtime had been ported to it

- Hejlsberg is C#'s principal designer and lead architect at BigM
- "Flaws in most major programming languages (e.g. C++, Java, Delphi, and Smalltalk) drove the fundamentals of the Common Language Runtime (CLR), which, in turn, drove the design of the C# language itself"

Design goals

"Simple, modern, general-purpose, object-oriented"

"Provide support for software engineering principles such as strong type checking, array bounds checking, detection of attempts to use uninitialized variables, and automatic garbage collection. Software robustness, durability, and programmer productivity are important."

And also

- Should support software for distributed environments.
- Portability (C/C++), internationalization
- No need to compete directly on performance and size with C or assembly language.

I've been talking too much...

› ...let's see it in action!

Development flow

Devtools

- Build/run based on command line
- Visual Studio vs. Visual Studio Code
- ...but basically, any editor would work

Code/folders structure

- Namespaces and assemblies (vs. Java Packages)
- Free file contents / folder structure compared to Java

Dependencies

- Nu-Get tool

Development flow (cont'd)

Solutions & projects

- References, shared projects
- Can specialize projects, e.g., for shared code/DLL, frontend, unit tests…

Config files

- Publishing, transformations
- Implement the Abstract Factory pattern

Versioning

- Git (github / DevOps)

Language: unique features

Strongly typed - more type safe than C++

- No non-safe implicit conversions
- Object-oriented (Like Java)

Local variables cannot shadow variables in enclosing non-classes blocks

- Unlike any other lang.

Arithmetic overflow automatically handled

- `checked`/`unchecked` (**statically** check blocks for arithmetic overflow)

Why are arithmetic overflow not nice?

ARIANE 5 FAILURE

- BACKGROUND:-
- European space agency's re-useable launch vehicle.
- Ariane-4 was a major success
- Ariane -5 was developed for the larger payloads

- LAUNCHED:-on June 4 1996
- MISSION was to delivered $500 million payloads to the orbit
- THE MAIDEN FLIGHT OF THE ARIANE 5 ENDED IN A FAILURE.
- ONLY AFTER 40 SECONDS THE FLIGHT VEERED OFF ITS PATH AND BROKE UP AND EXPLODED
- CAUSE: Unhandled floating point exception in code
- ENGINEERS FROM THE ARIANE PROJECT STARTED TO INVESTIGATE THE CAUSES OF LAUNCH FAILURE.

Language: basic features

- Base types and unified type system (unlike Java)
- Arrays and collections
- Pointers and references
- Implicit types (`var`)

- Boxing/unboxing
- Nullables
- Dynamic types

Object Oriented Programming

- Classes, interfaces, abstract classes, partial classes
- Properties as opposite to `get` & `set` properties
- ~~`throws`~~ `try/catch/finally` (no unchecked exceptions in Java)
- Expression bodies `<DEFINITION> => <EXPR>`

- Dynamic types
- Operator overloading (!!)
- Generic are not a type, but they are supported inside the runtime (reification)

Language: advanced features

Attributes to classes/methods

- Like Java

Functional programming

- Closures
- Delegates =>lambda functions

Language-Integrated Query (LINQ)

- Query expression (SQL-like)
- Fluent expression

Fluent interface coding pattern

Not really anything new…

› An Object-Oriented API / **coding pattern** that relies on method chaining

TL;DR: always "`return this`" in methods

What is a *context*, in this sense?

› The return value of method, creates a context

› In fluent interface, the context is always the same ("this")

› Ended when a method returns void

Example: ??

Pros/cons

Pros

- Increase code readability
- Designs an interface that specifies a DSL (Domain Specific Language), identified by the context

Cons

- Brings some errors at run-time, instead of compile time
- Harder to debug/log
- Issues when subclassing/ovrerriding methods

References

Course website

› http://hipert.unimore.it/people/paolob/pub/ProgSW/index.html

How to run the examples

```
$ dotnet new [webapi|console] [--use-controllers] [-o NAME]
$ dotnet build
$ dotnet run
```

My contacts

› paolo.burgio@unimore.it

› http://hipert.mat.unimore.it/people/paolob/