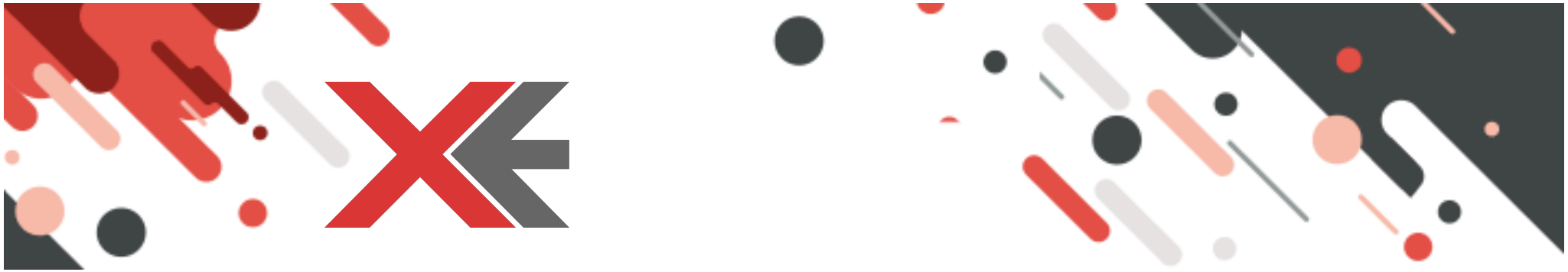


# Modernizing Enterprise Apps

**Mirco Vanini**

#XeOneDay

Evento realizzato grazie
al supporto di

DBA LAB
A DBA Group Company

MVP
Microsoft®
Most Valuable
Professional

www.xedotnet.org

# Agenda

Windows desktop market and opportunity
One “Windows” Platform
Why Windows Desktop on .NET Core?
.NET Core 3 Desktop Improvements
.NET Core road map
.NET FW technologies unavailable on .NET Core 3
Migrating a WinForm / WPF app to Core

# Survey

- Windows Forms ?
- WPF ?
- UWP ?

# Windows desktop market and opportunity

**2.4 Million**

Developers building desktop apps in Visual Studio every month

**700 Million**

Windows 10 Desktops

**+50%**

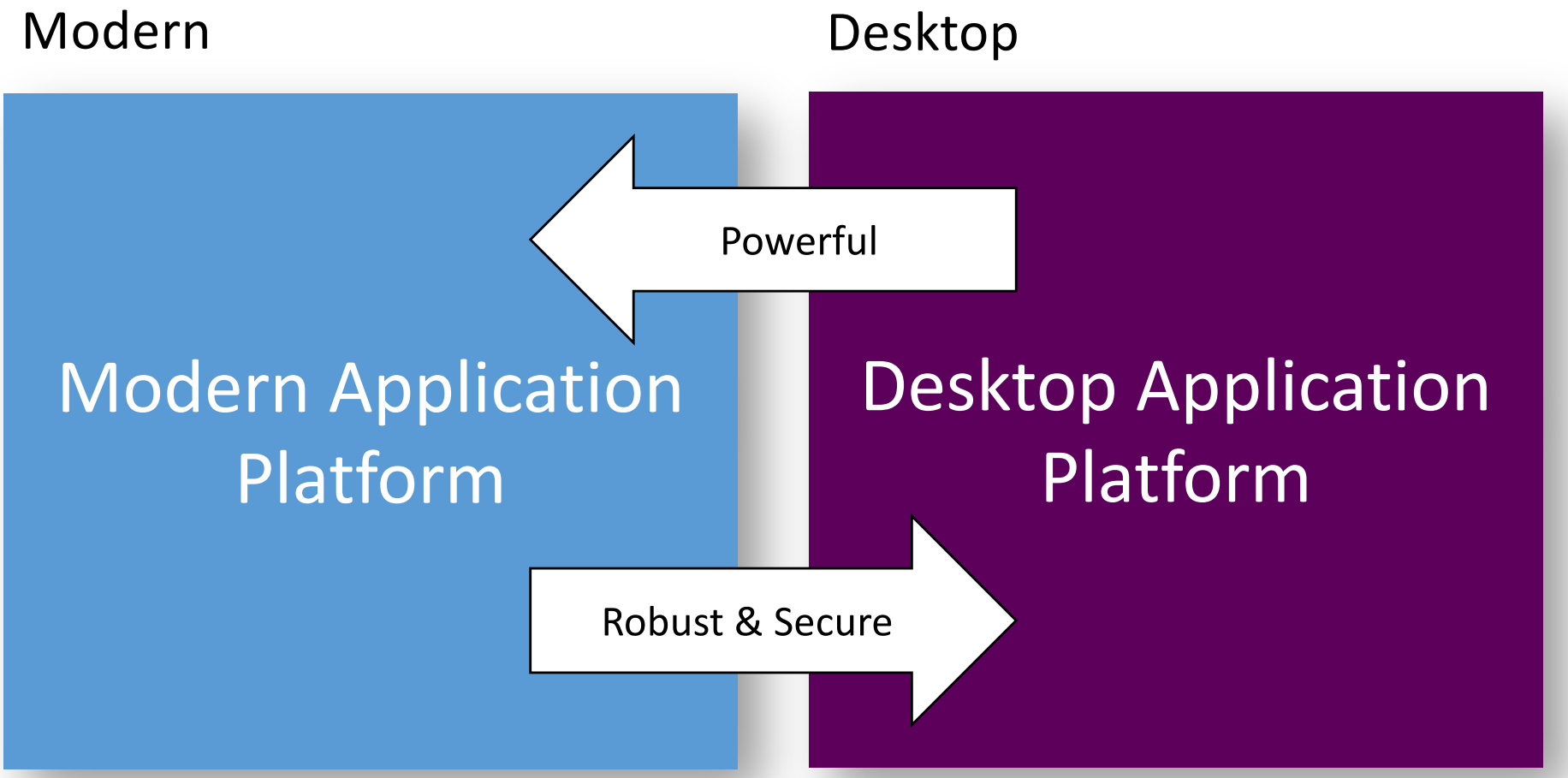
In the last 18 months

Source: Ignite 2018

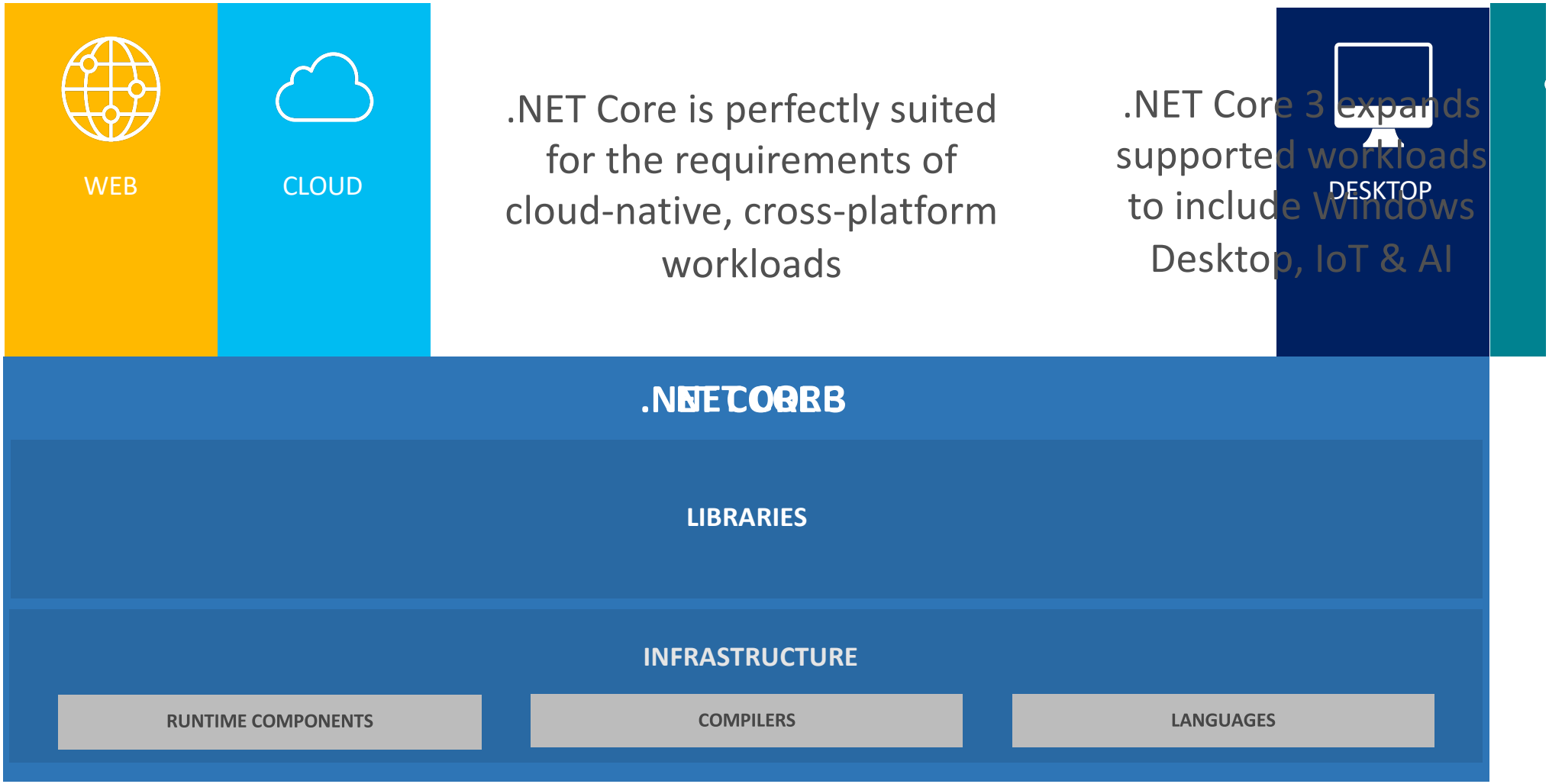
# Windows 10

- How many of your companies have moved to Windows 10 ?
- Remain on Windows 7 ?
- Plans to move by … ?

# One "Windows" Platform



# .NET Core 3


WEB
CLOUD
.NET Core is perfectly suited for the requirements of cloud-native, cross-platform workloads
DESKTOP
.NET Core 3 expands supported workloads to include Windows Desktop, IoT & AI
LIBRARIES
INFRASTRUCTURE
RUNTIME COMPONENTS
COMPILERS
LANGUAGES

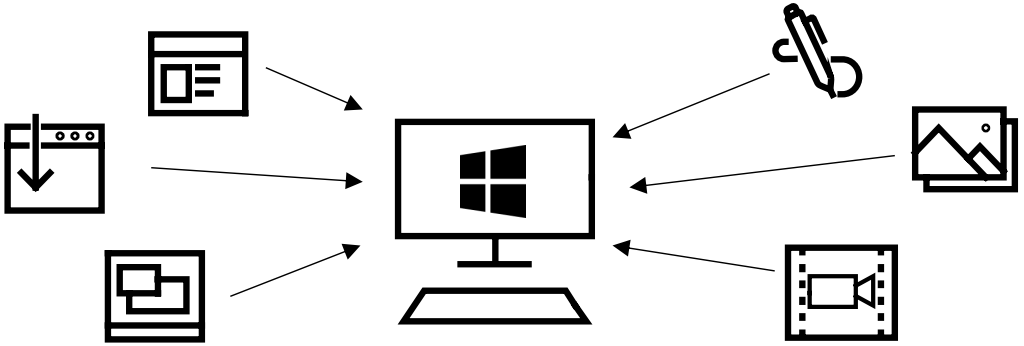
# Why Windows Desktop on .NET Core?

- .NET Core is the fast innovating .NET
- Deployment Flexibility
  - Side-by-side support
  - Machine global or app local framework
  - Self-contained EXEs
- Core runtime and API improvements
  - BCL
  - Language features
- Performance
- Open Source
  - Windows Form
  - Windows Presentation Foundation
  - Windows UI XAML Library

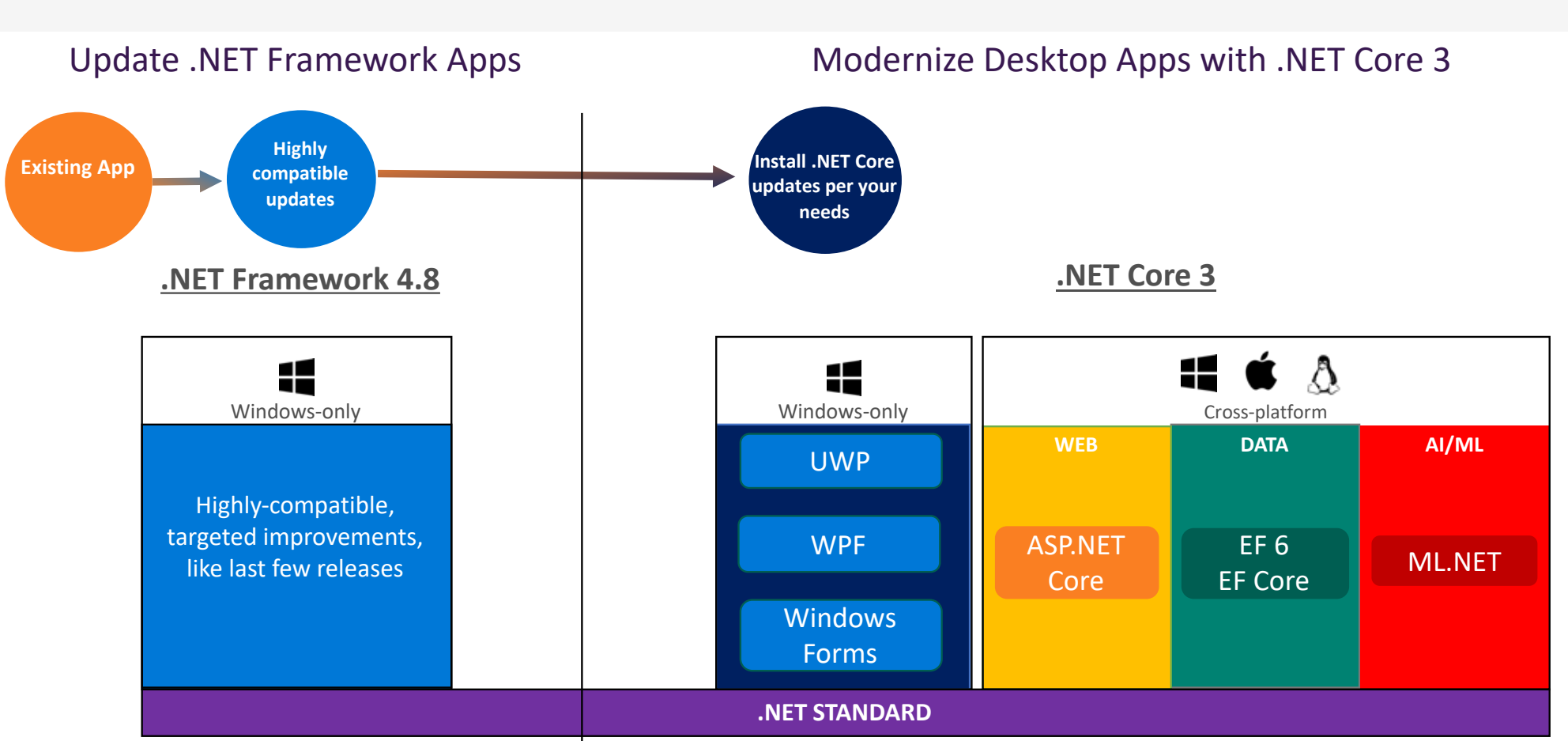
# .NET Core 3 Desktop Improvements

- Support for Windows Forms and WPF
  - XAML Islands – WinForms & WPF can host UWP
  - XAML Controls – WinForms & WPF browser and media UWP controls
  - High DPI fixes for WinForms
- Access to all the Windows 10 (AKA “WinRT”) APIs
- .NET Core App Bundle
  - Precompiled, fast startup
  - Small app by removing unused dependencies, link away unused IL
  - Self-contained .exe

(WPF + WinForms) * .NET Core = Modern Desktop



# Big Picture




- .NET Framework support **unchanged** (supported for life of Windows)

- Side-by-side support
- Self-contained EXEs

- XAML Islands - WinForms & WPF apps can host UWP controls
- Full access to Windows 10 APIs

FEATURES IN BOTH FXs

# Microsoft blogs quotes

*"Async streams, indexers and ranges all rely on new framework types that will be part of .NET Standard 2.1 […]NET Core 3.0 as well as Xamarin, Unity and Mono* ***will all implement .NET Standard 2.1, but .NET Framework 4.8 will not****"*

*".NET Core will get new APIs and language features over time* ***that .NET Framework cannot****"*

*"Most of the C# 8.0 language features will run on any version of .NET. However,* ***a few of them have platform dependencies****."*

*".NET Framework is* ***going to see less innovation*** *in the future, instead focusing on stability and reliability"*

*"to remain as compatible as possible […] .NET Framework* ***moves at a slower pace*** *than .NET Core"*

*"Default interface member implementations rely on new runtime enhancements, and we will not make those in the .NET Runtime 4.8 either. So this feature simply* ***will not work on .NET Framework 4.8*** *and on older versions of .NET."*

*"…continue to make it easier to move applications to .NET Core […] by adding WPF, WinForms support […] we will keep porting APIs and features to help close the gap and make migration easier for those who chose to do so."*

*"…the file APIs were faster on .NET Core. If we put those same changes into .NET Framework we could break existing applications, and we don't want to do that."*

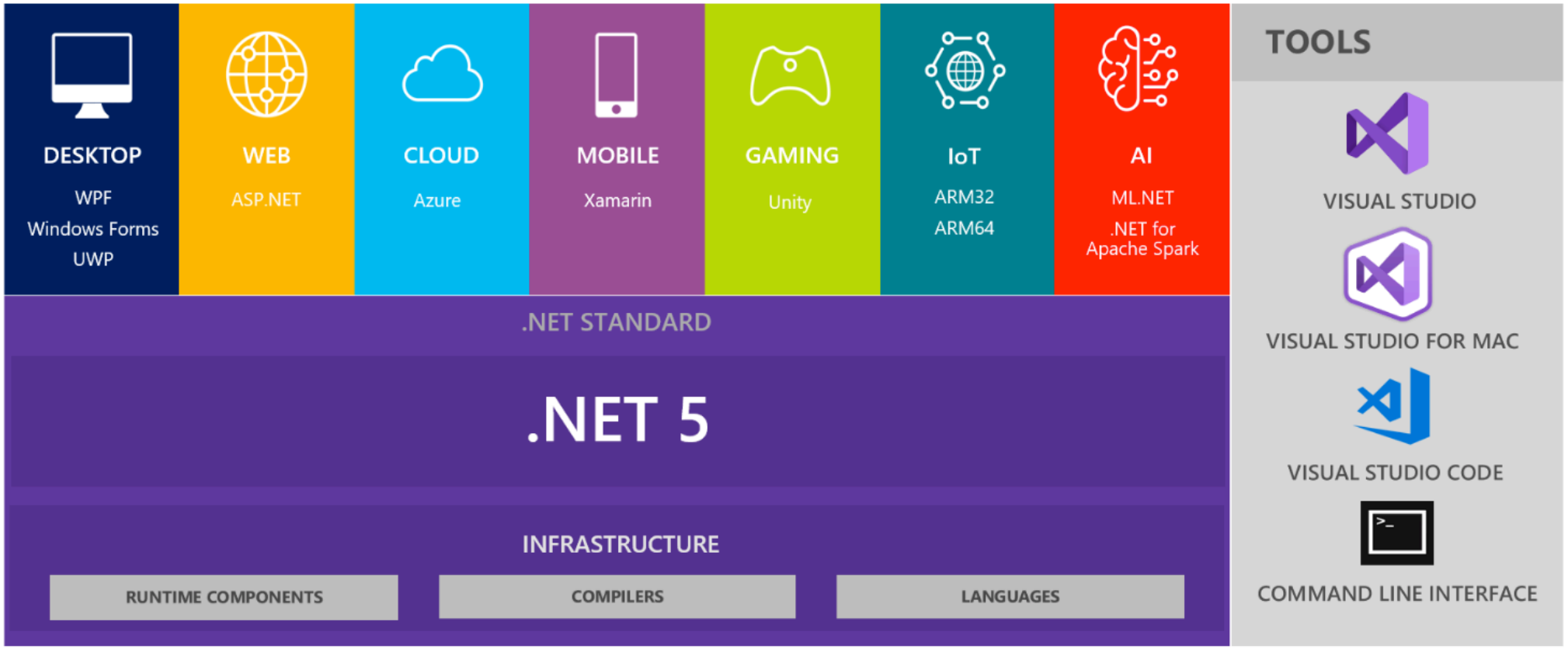
*"While we do recommend that* ***new desktop apps should consider targeting .NET Core****, the .NET Framework will keep the high compatibility bar and will provide support for your apps for a very long time to come"*

*"…moving forward* ***they will contain somewhat different features****"*

Sources:

https://blogs.msdn.microsoft.com/dotnet/2018/10/04/update-on-net-core-3-0-and-net-framework-4-8/
https://blogs.msdn.microsoft.com/dotnet/2018/12/04/announcing-net-core-3-preview-1-and-open-sourcing-windows-desktop-frameworks/
https://blogs.msdn.microsoft.com/dotnet/2018/11/12/building-c-8-0/

# .NET 5 = .NET Core vNext – A Unified platform




# .NET Core 3 current status

- .NET Core 3 Preview 5 SDK's WinForms & WPF
  - VS templates
  - CLI command
- Visual Studio 2019 supports building and debugging
  - Designers not available yet (16.04)
  - Designers is available start 16.1 preview 3
- WPF, Windows Forms, Windows UI Library are OSS
- WPF, Windows Form complete covered on .NET Core Preview 7

# .NET FW technologies unavailable on .NET Core 3

- AppDomains
- Remoting
- Code Access Security (CAS)
- Security Transparency
- COM* (any COM environment (e.g. C/C++) other than .NET.)

| Server | Client | Current Support |
|---|---|---|
| COM* | .NET Core | Yes |
| .NET Core | COM* | Yes |
| .NET Core | .NET Core | Yes |
| .NET Framework | .NET Core | No |
| .NET Core | .NET Framework | No |

https://docs.microsoft.com/en-us/dotnet/core/porting/net-framework-tech-unavailable
https://github.com/dotnet/core-setup/blob/master/Documentation/design-docs/COM-activation.md

# .NET Core 3 road map

- .NET Core 3.0: September 2019
  - Includes Windows forms and WPF
  - Open Source on “dotnet” GitHub
- .NET Core 3.1: November 2019
  - Long Term Support (LTS)

# Migrating a WinForm / WPF app to Core

- Identify dependencies
- Retarget to .NET Framework 4.7.2
- .NET Portability Analyzer
- Packages.config -> PackageReference
- Libraries first (.NET Standard or multi-target)
- Convert your old project files to the new 2017 format
- Test everything thoroughly (use the app)
- Consider self-contained (beware risk)

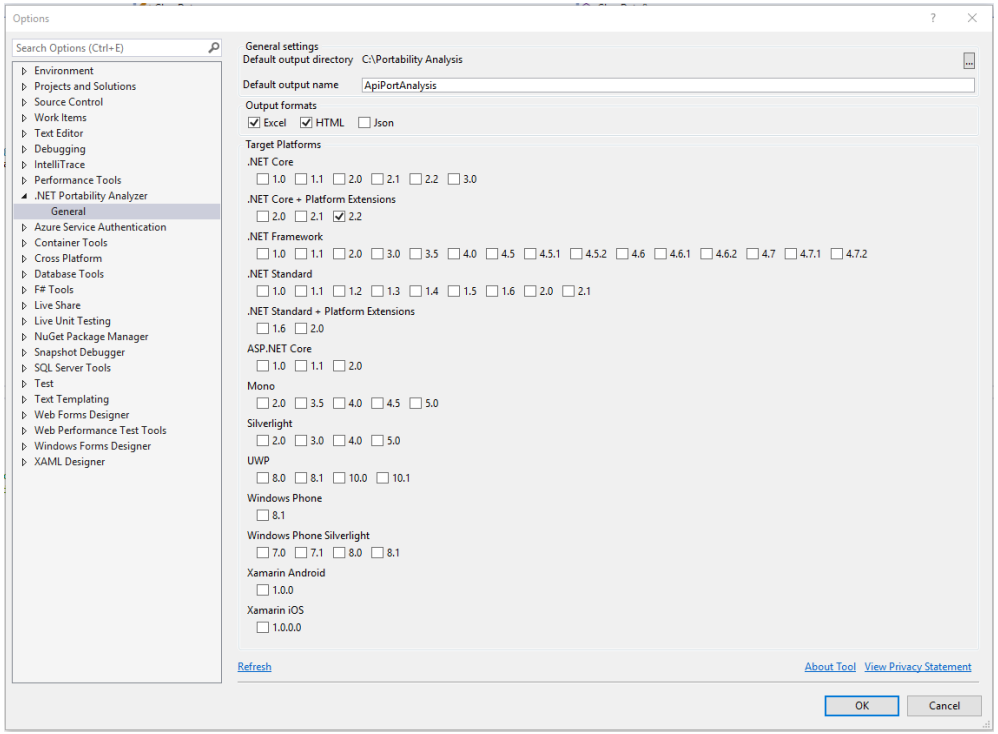
https://docs.microsoft.com/en-us/dotnet/core/porting/
https://devblogs.microsoft.com/dotnet/how-to-port-desktop-applications-to-net-core-3-0/
https://github.com/hvanbakel/CsprojToVs2017
https://www.michaeltaylorp3.net/migrating-to-sdk-project-format/

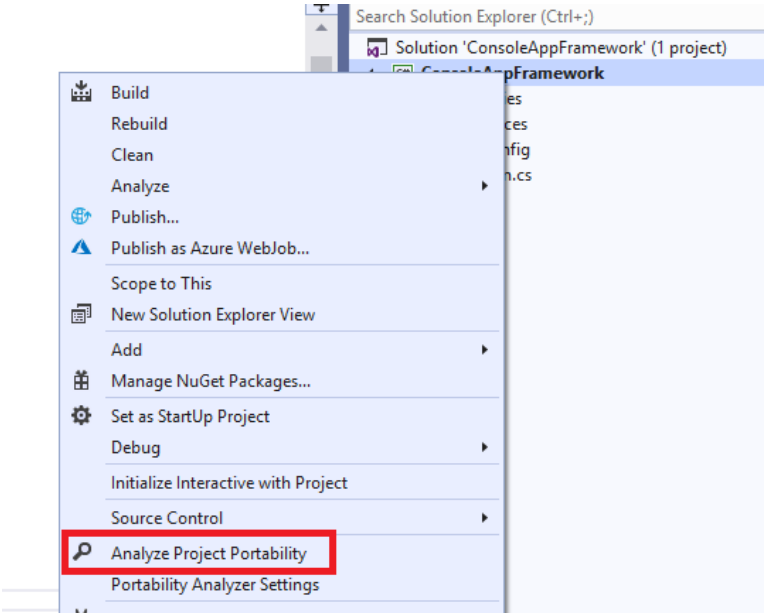
# Migrate Windows Form / WPF Apps

DEMO

# Detection Platform Compatibility Issues

- API Analyzer
  - Roslyn analyzer that notifier you about the usage of APIs that don't work across all platforms and about deprecate APIs




https://docs.microsoft.com/en-us/dotnet/core/porting/windows-compat-pack

# Windows Compatibility Pack

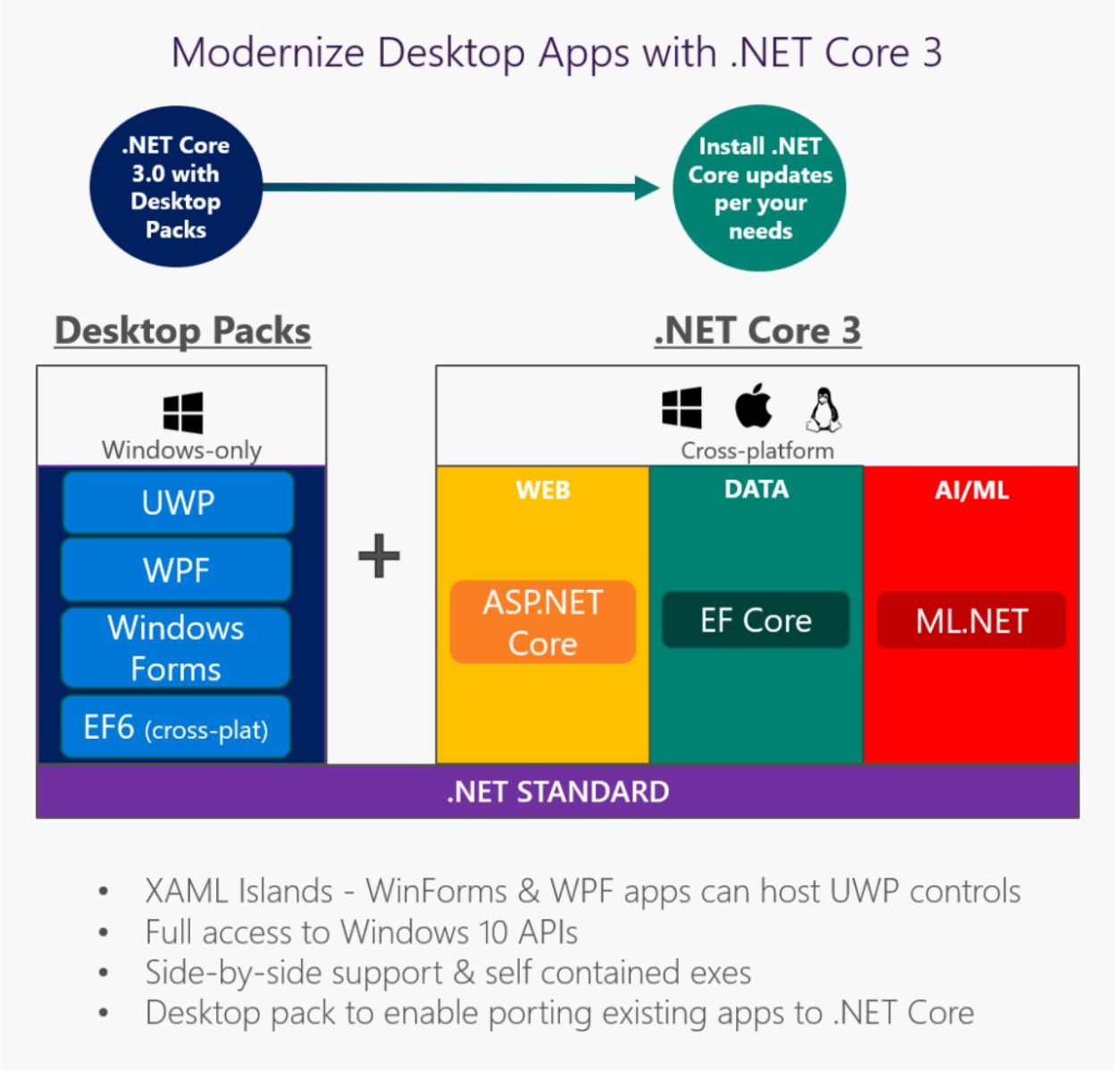
- Provided as NuGet Package
  - Microsoft.Windows.Compatibility
  - Can be referenced from .NET Core & .NET Standard
  - Has more 21K APIs (Windows-only as well as cross-platform)

- Code Pages
- CodeDom
- Configuration
- Directory Services
- Drawing
- ODBC
- Permissions
- Ports

- Windows Access Control Lists (ACL)
- Windows Communication Foundation (WCF)
- Windows Cryptography
- Windows EventLog
- Windows Management Instrumentation (WMI)
- Windows Performance Counters
- Windows Registry
- Windows Runtime Caching
- Windows Services

https://docs.microsoft.com/en-us/dotnet/core/porting/windows-compat-pack

# Recap


Modernize Desktop Apps with .NET Core 3
.NET Core 3.0 with Desktop Packs
Install .NET Core updates per your needs
Desktop Packs
.NET Core 3
Windows-only
UWP
WPF
Windows Forms
EF6 (cross-plat)
Cross-platform
WEB
DATA
AI/ML
ASP.NET Core
EF Core
ML.NET
.NET STANDARD
XAML Islands - WinForms & WPF apps can host UWP controls
Full access to Windows 10 APIs
Side-by-side support & self contained exes
Desktop pack to enable porting existing apps to .NET Core

# Contatti


## Mirco Vanini

www.proxsoft.it

info@proxsoft.it

@MircoVanini


Microsoft® MVP Windows Development
AllSeen Alliance - AllJoyn® Ambassador
Open Connectivity Foundation - OCF® Ambassador