.NET Core

by

Dr. Günter Kolousek

Quellen

- https://docs.microsoft.com
- *Professional C# and .NET Core 2.0*
  - Christian Nagel
  - John Wiley & Sons, Inc.
  - 2018
  - ISBN: 978-1-119-44927-0
- https://de.wikipedia.org/wiki/.NET
- https://de.wikipedia.org/wiki/.NET_Framework
- https://github.com/dotnet/core
- ...

Einführung

- .NET ist eine Familie von Frameworks von Microsoft
  - gewachsen seit 2002
  - unübersichtliche Geschichte
- ursprüngliche Idee: ein Framework für mehrere Programmiersprachen unter Windows
- heute: mehrere .NET-basierende Implementierungen!
- eine .NET App wird ausgeführt in einer Implementierung
- .NET Core Guide

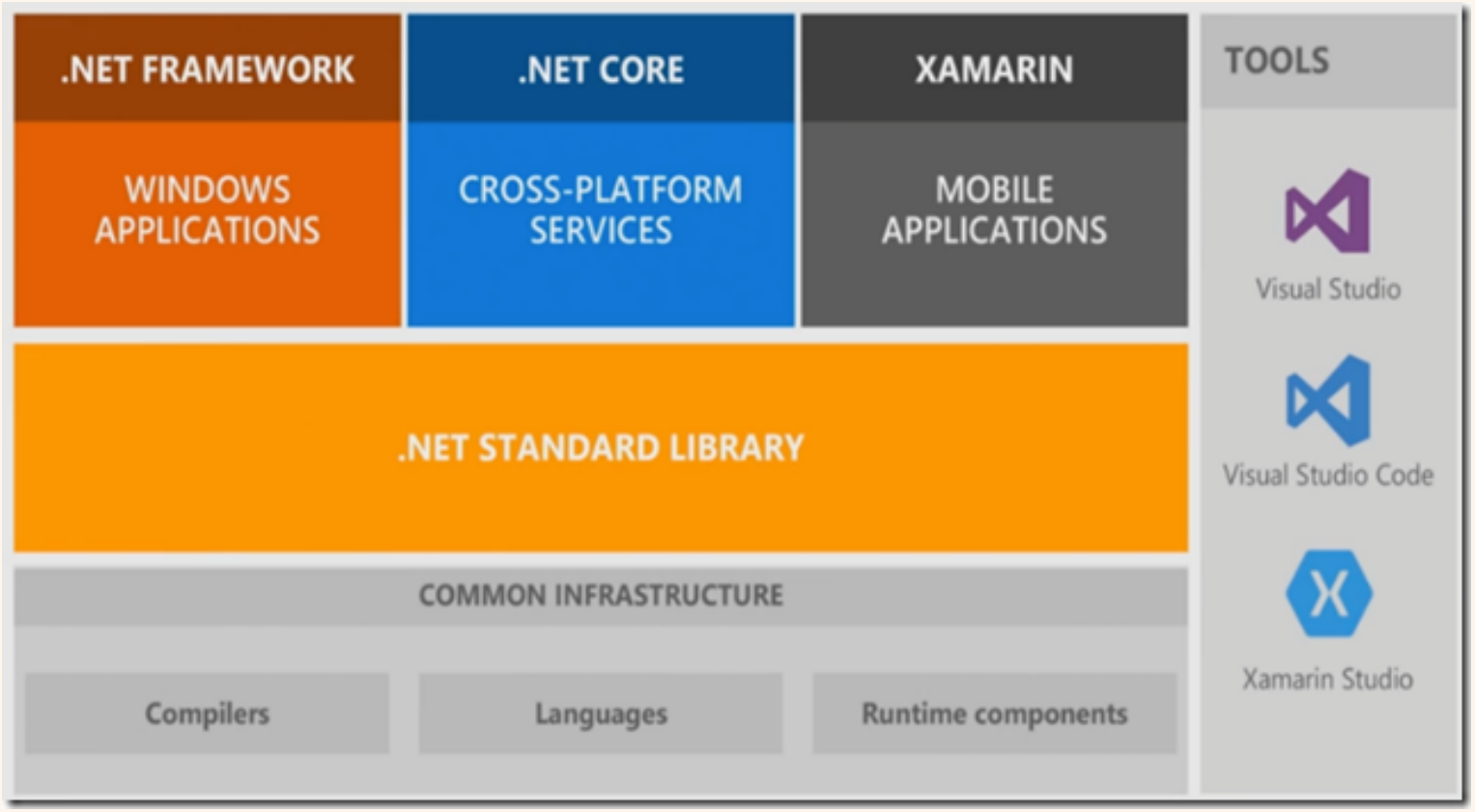
.NET Implementierungen

- .NET Framework
- .NET Core
- Xamarin (für Mac, iOS, Android)
- Mono
- UWP (Universal Windows Platform)

  für Touchscreengeräte und Software für das Internet der Dinge (Internet of Things, IoT) … PCs, Tablets, Phablets über Smartphones bis hin zur Xbox – https://docs.microsoft.com/de-de/dotnet/standard/components

  Programmable in C#, C++, Visual Basic, and Javascript. For UI, use XAML, HTML, or DirectX. – https://docs.microsoft.com/en-us/windows/uwp/get-started/universal-application-platform-guide

.NET Überblick



Quelle: https://blogs.msdn.microsoft.com/premier_developer/2017/04/12/net-core-overview/

.NET Standard

keine Implementierung, sondern eine Spezifikation

- liegt allen Implementierungen zugrunde
- .NET Standard 2.0
  - .NET Core 2.0 (dzt. .NET Core in Version 2.1)
  - .NET Framework 4.6.1
  - Xamarin.iOS 10, Xamarin.Android 8.0, Xamarin Mac 3.8
  - Mono 5.4 … Open Source von Xamarin (Microsoft)
    - ist Grundlage von Xamarin
    - implementiert einen größeren Teil der APIs von .NET
  - UWP 10.0.1

Common Language Infrastructure

- CLI, internationaler Standard (ISO 23271 und ECMA-335)
  - für sprach- und plattformneutrale Anwendungsentwicklung
- besteht aus
  - Common Type System (CTS)
    - Hierarchie von Datentypen
  - Metadaten
    - Metadaten, die Klassen und Methoden beschreiben
    - werden vom CLI Compiler erzeugt und zu CIL in Assembly gespeichert
  - Common Intermediate Language (CIL)
    - Zwischencode
    - objektorientierte stack-basierte Sprache
  - Virtual Execution System (VES)
    - $\rightarrow$ Common Language Runtime (CLR) in .NET
  - Common Language Specification (CLS)
    - Regeln, die alle kompatiblen Sprachen erfüllen müssen
    - Sprachen: C#, C++/CLI, VB.NET, F#

.NET Implementierungen – 2

bestehen aus

- mind. einer Runtime (basiert auf CLI)
  - CLR für .NET Framework
  - CoreCLR für .NET Core
- einer Base Class Library (BCL, implementiert .NET Standard)
  - z.B.: .NET Framework Base Class Library
  - z.B.: .NET Core Base Class Library (CoreFx)
- Optional: Anwendungsframeworks
  - z.B.: ASP.NET oder ASP.NET Core
  - z.B.: Windows Forms oder WPF
- Optional: Entwicklungstools
  - Compiler
  - Build-Management
  - Paket-Management
  - z.B. `dotnet` von .NET Core

Common Language Runtime (CLR)

- Laufzeitumgebung (analog zu JRE)
- Abarbeitung von IL (Intermediate Language) Zwischencode (analog zu Java Opcodes)
  - auch CIL genannt (Common IL)
    - objektorientierte stack-basierte Sprache
  - d.h. ist eine virtuelle Maschine
  - wird von CLR verwaltet ($\rightarrow$ managed code)
  - IL wird von Compiler erzeugt und in Assembly gespeichert
- enthält einen JIT-Compiler
- Managed vs. Unmanaged
  - Managed: Programme, die von CLR verwaltet werden
  - Unmanaged: nicht in CLR (nativer Code, meist Treiberprogramme)

.NET Standard 2.0 – BCL Überblick

- System
- System.Collections
- System.Collections.Generic
- System.Data
- System.IO
- System.Linq
- System.Net
- System.Numerics
- System.Reflection
- System.Runtime
- System.Security
- System.Text
- System.Threading
- System.XML

Features von .NET Core

- Open Source
- verwendet moderne Patterns
- unterstützt mehrere Plattformen
  - Windows, macOS, Linux und Docker-Images
- ASP.NET Core
- Entity Framework Core (ORM)
- modular – im Gegensatz zu .NET Framework

Anwendungsgebiete

Wahl zwischen .NET Core und .NET Framework für Server-Apps:

- .NET Core
  - plattformübergreifende Anwendungen
  - Erstellung von Microservices
  - verwenden von Docker-Container
  - für skalierbare Hochleistungssysteme
    - 20.8.2018: M$ stellt bing.com auf .NET Core 2.1 um $\rightarrow$ 34% Reduktion der Server-Latenz
  - pro Anwendung verschiedene parallele .NET Versionen notwendig
- .NET
  - vorhandene .NET Anwendungen
  - Bibliotheken von Drittanbietern
  - wenn .NET Technologien notwendig, die .NET Core nicht unterstützt (z.B. WPF)

Ausblick auf .NET Core 3

- Unter .NET Core 3 and Support for Windows Desktop Applications findet man die folgenden Zitate:

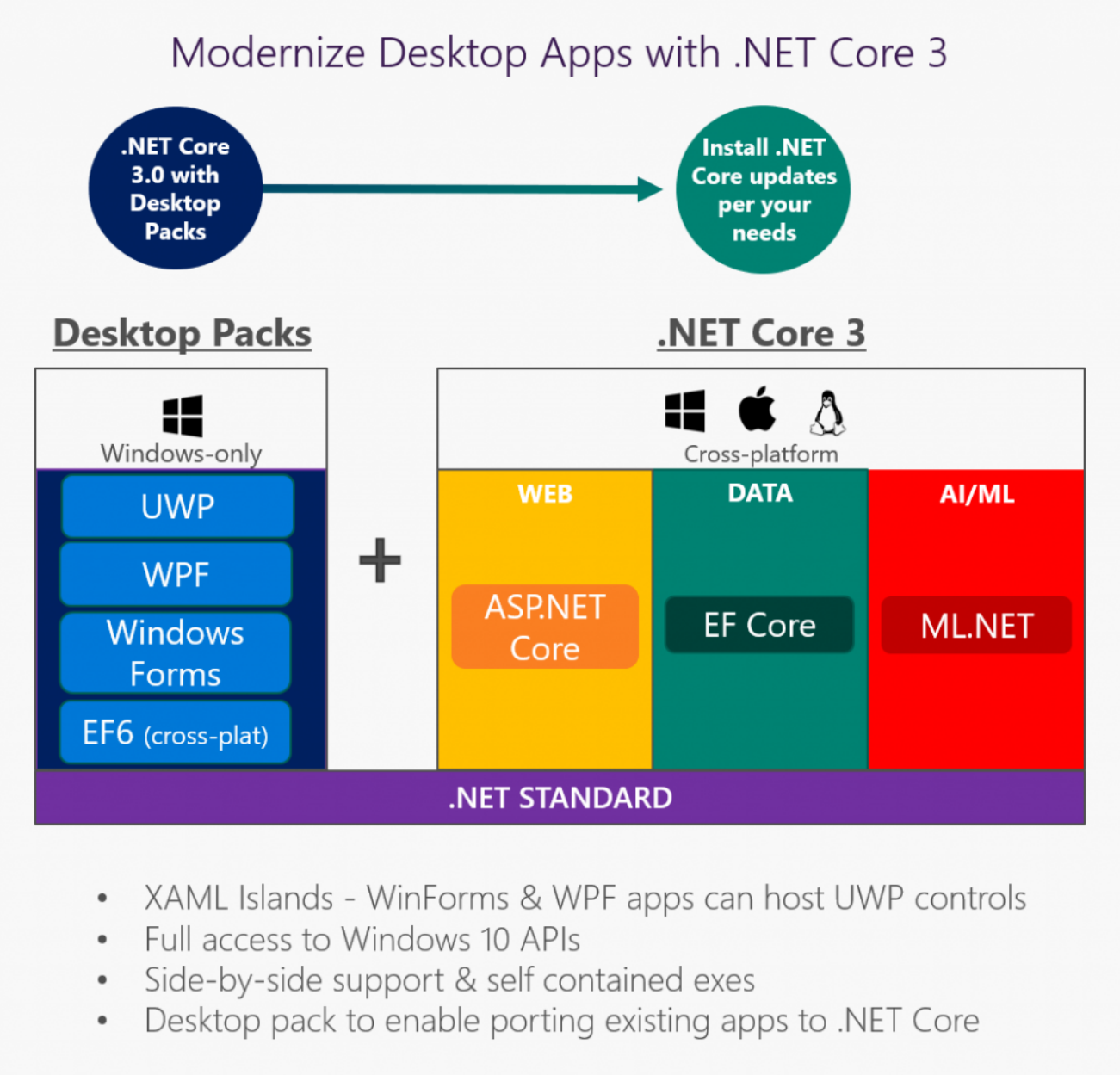
  We think that .NET Core 3.0 will be one of the most exciting .NET releases we've ever released.

  For new desktop applications, we'll guide everyone to start with .NET Core 3.

  There are many benefits with .NET Core that are great for desktop apps. … Performance improvements and other runtime updates that will delight your users

- Unter .NET Core Roadmap: Milestone == Q1 2019

Ausblick auf .NET Core 3 – 2



- 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