



# .NET Core Hintergrund Info's

Entwicklertag 2015 in Karlsruhe



# Andreas Bräsen



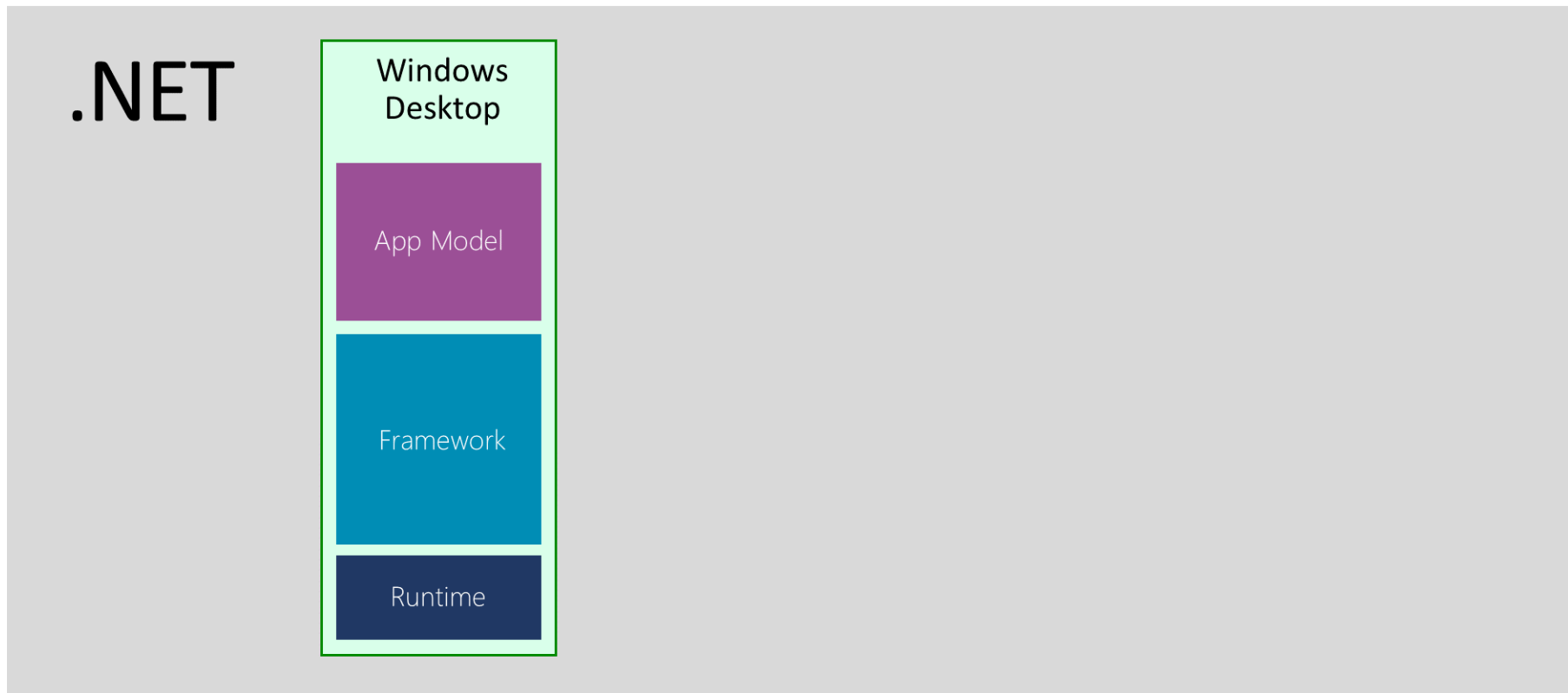
- Freiberuflicher Software Entwickler Mit dem Schwerpunkt auf .NET basierte pragmatischer Software Entwicklung
- Community
  - .NET User Group
  - #nossued
- [abraesen@bruke.de](mailto:abraesen@bruke.de)
- @abraesen



## Vier Probleme als Motivation für .NET Core

- Viele .NET Verticals
- Libraries für mehrere .NET Verticals
- Deployment von .NET
- Es gibt noch andere Betriebssysteme als Windows

# Problem: Viele .NET Verticals



## Problem: Libraries für mehrere Verticals - Die Geburt der PCL (Portable Class Libraries)

- Es gab bis dato kein Konzept, um übergreifend Libraries auszutauschen.
- Code Sharing mittels
  - Linked Files
  - Partial Classes
  - #if def
  - Abgeleitete Klassen
- Einführung von Portable Class Libraries (PCL)

# PCL

 net20

 net35

 net40

 net45

 netcore45

 portable-net40+sl4+wp7+win8

 portable-net45+wp80+win8

# Problem: Deployment von .NET

- Maschinenweit
  - Einige Vorteile
    - Zentralisierte Aktualisierung
    - Disk Space Reduzierung

# Problem: Deployment von .NET

- Maschinenweit
  - Einige Nachteile
    - Man benötigt als Entwickler auf der Zielumgebung ein installiertes .NET Framework oder man bringt die Installation mit.
    - Man muss sich mit dem begnügen was man auf der Zielumgebung findet.
    - Eine Aktualisierung durch meine Application kann andere Applikationen beeinträchtigen.



# Problem: Deployment von .NET

- Kompatible Änderungen...
  - Selbst kompatible Anpassungen können bestehende Applicationen beeinflussen.
    - Ein Interface hinzufügen => Die Serialisierung kann sich verändern
    - Ein Overload hinzufügen => Die Anzahl der Methoden ändert sich
    - Einen internen Type umbenennen => To String verändert sich
  - 99,9% Kompatibilität bei 1.8 Milliarden Maschinen => 1.8 Millionen nicht klappt

# Problem: Deployment von .NET

- Applikationslokal
  - Einige Vorteile
    - Man hat alles unter Kontrolle als Entwickler.
    - Man braucht nur noch seine Applikation installieren.

# Problem: Deployment von .NET

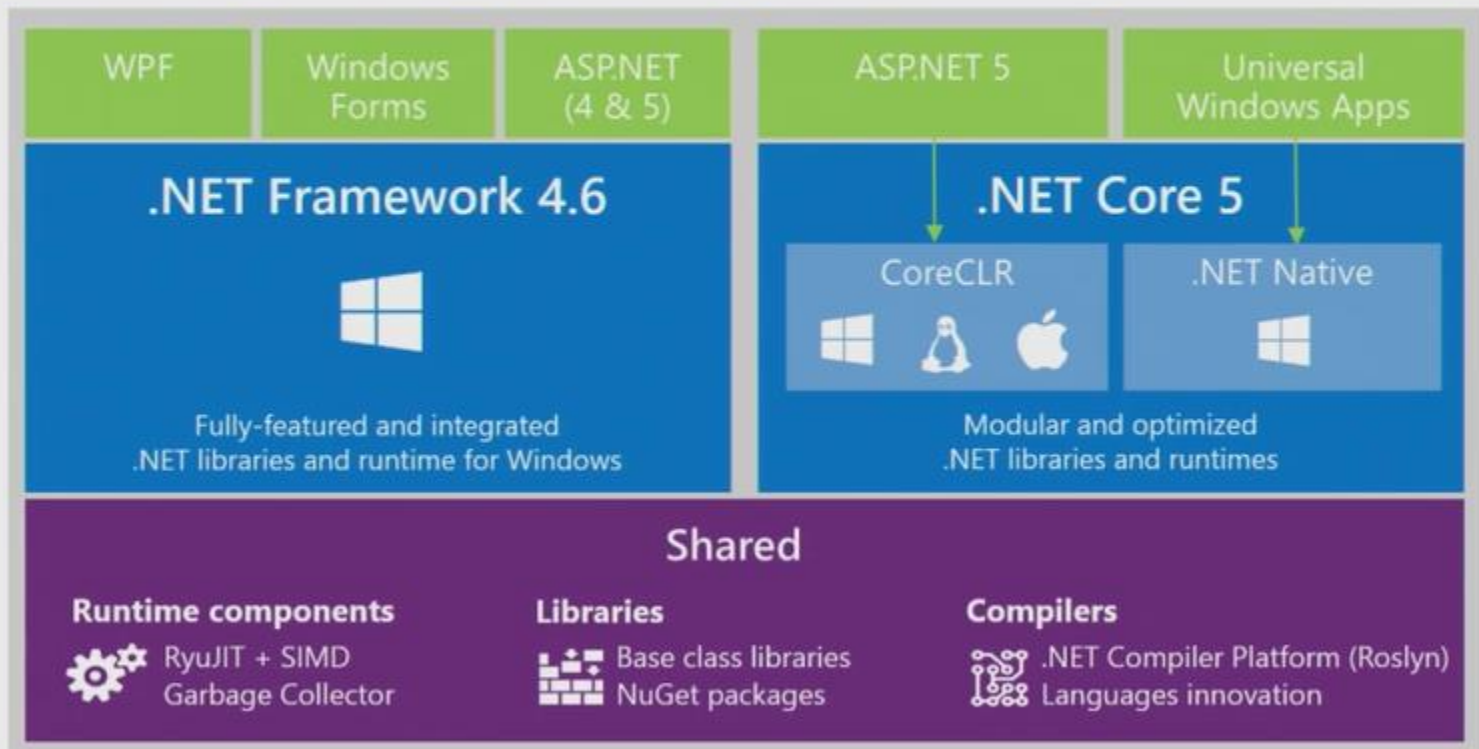
- Applikationslokal
  - Einige Nachteile
    - Der Disk Space, den man für seine Applikation hat wird mehr.
    - Teilung von Assemblies wird eine wenig komplizierter.
    - Man muss sich eventuell selber um die Aktualisierung der Microsoft Assemblies kümmern....
  - Wie läuft das mit dem Update ?
    - So wie es aktuell auch für NuGet Packages laufen würde, die in der Applikation verbaut sind.
      - Die Hersteller der Applikation ist dafür verantwortlich.

Problem: es gibt noch andere Betriebssysteme als Windows...

- Linux
- OSX
- Free BSD
- ...

# Ein Neuanfang... .NET Core

.NET 2015



## Was ist .NET Core ?

- .NET Core ist eine modulare Implementation, welche in vielen App Models benutzt werden kann.
- .NET Core ist im Grunde ein Fork des .NET Frameworks, dessen Implementierung auf das “Decomposition/Factoring” Gesichtspunkte optimiert ist.
- Es wird die Grundlage für alle zukünftigen App Models sein.

## Was ist .NET Core ?

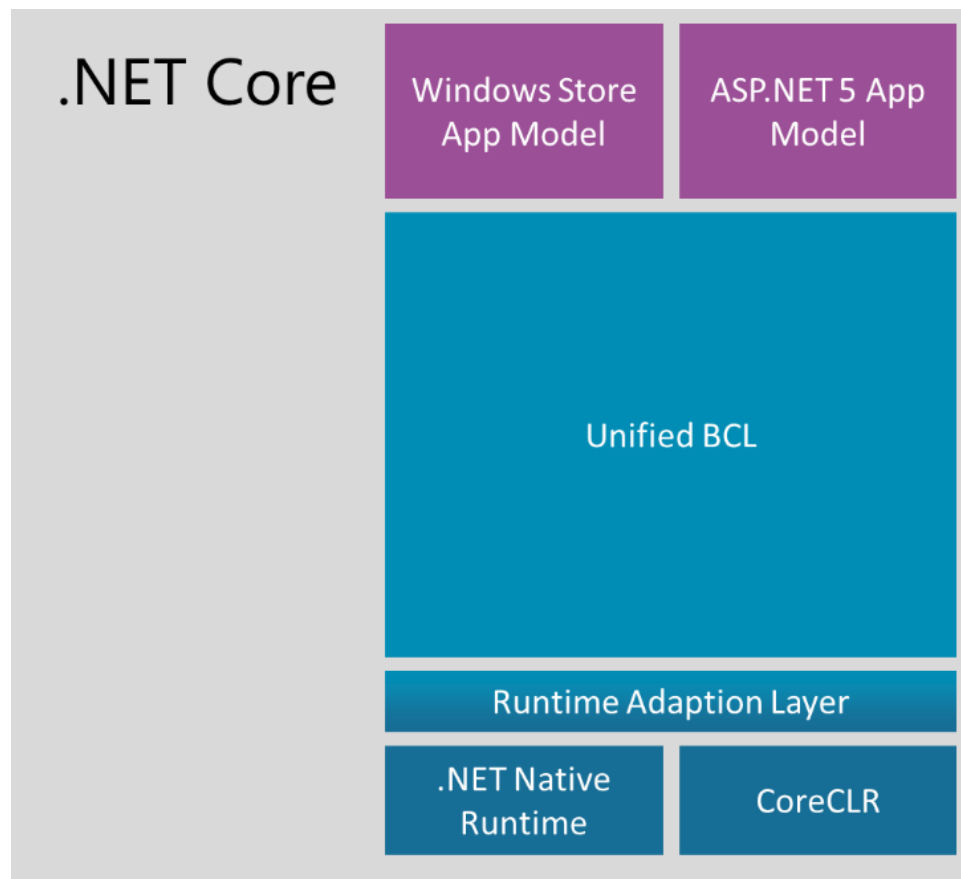
- Es skaliert vom Data Center zum Touch Based Device.
- Es wird von Microsoft für Windows, Linux/Unix und Max OSX supported.
- Es ist OpenSource.

# Wie löst .NET Core die Probleme ?

- Viele .NET Verticals
- Libraries für mehrere .NET Verticals
- Deployment von .NET
- Es gibt noch andere Betriebssysteme als Windows



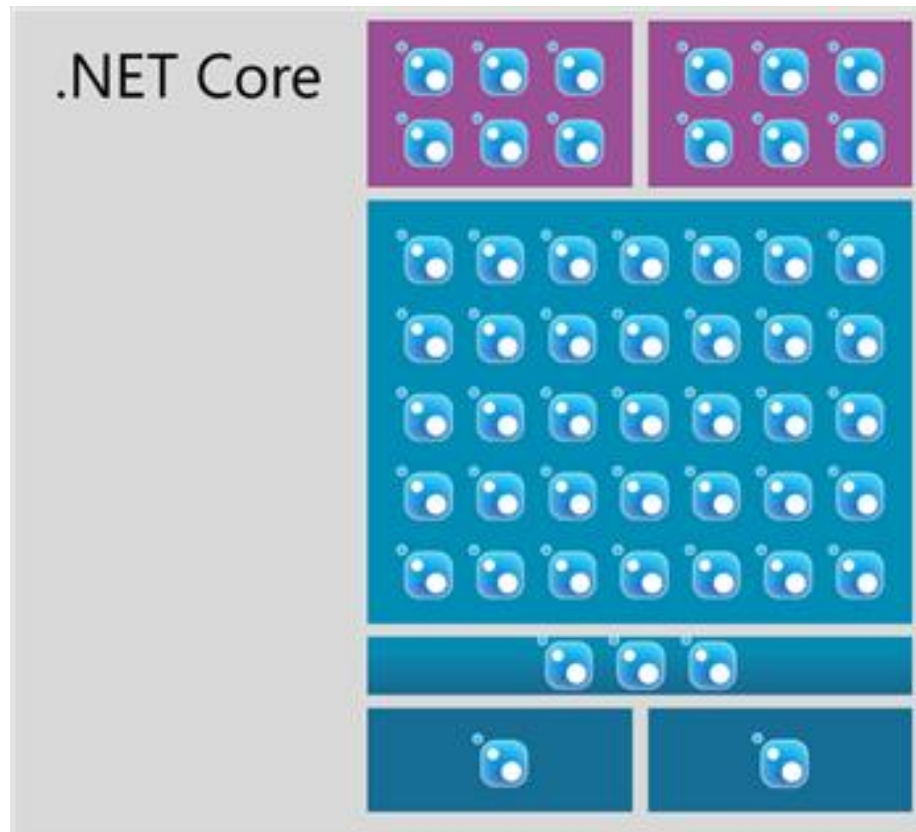
## Problem: Viele .NET Verticals - Erste Vereinheitlichung von .NET Native und ASP.NET



# Windows Store & Windows Phone

.NET Core wird dort Einzug halten... und man ist nicht mehr so abhängig von Framework Releases, was auch die Innovationsgeschwindigkeit erhöhen soll.....

## Problem: Libraries für mehrere Verticals - Verteilung über NuGet



# Nuget - .NET Core Pakete

- Für den BCL-Layer werden die NuGet Pakete so heißen wie die Komponenten.
  - z.B. wird das Paket [Microsoft.Bcl.Immutable](#) zu [System.Collections.Immutable](#)
- Die Assembly Nummern entsprechen den Nummer des NuGet Paketes (Semantic Versioning).
- Vorteil:
  - Man sieht gleich schon am NuGet Paket was drin ist und in welcher Version.
  - Der Upgrade der .NET Core Komponenten ist so einfach wie der Upgrade jeder anderen Komponente, die über NuGet verteilt wird...

## Problem: Deployment von .NET Application Lokales Framework

- Man braucht nur das nehmen, was man für seine Applikation braucht.
- Smart Sharing
  - Verschiedene Applikationen verwenden das selbe Framework (Sie arbeiten daran – Stand Nov 2014)
- Ziel war es, dass ein Upgrade eines lokalen Frameworks keine Auswirkungen auf andere Applikationen hat.... (Sieht gut aus ;o)

## Problem: Es gibt noch andere Betriebssysteme

- .NET Core läuft jetzt schon auf
  - Linux
  - OSX
  - Free BSD

Puh, einmal durchatmen... aber... ?

## Ist das Enterprise Ready ?

- Es wird einen “Snapshot” aller zusammenarbeitenden NuGet Packages geben, die wie heute als .NET Framework in einer bestimmten Version angesehen werden können. Das Testen übernimmt Microsoft....
- Es wird darüber nachgedacht 4 mal im Jahr solch einen Snapshot anzubieten.

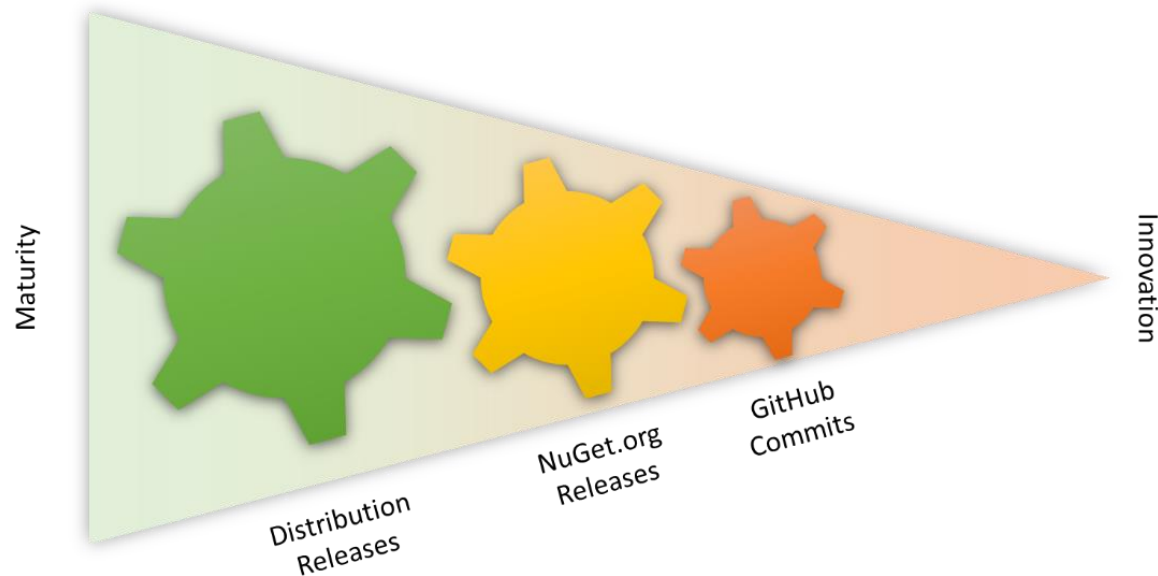


# Wie läuft das mit Updates ? Security und NuGet Packages ?

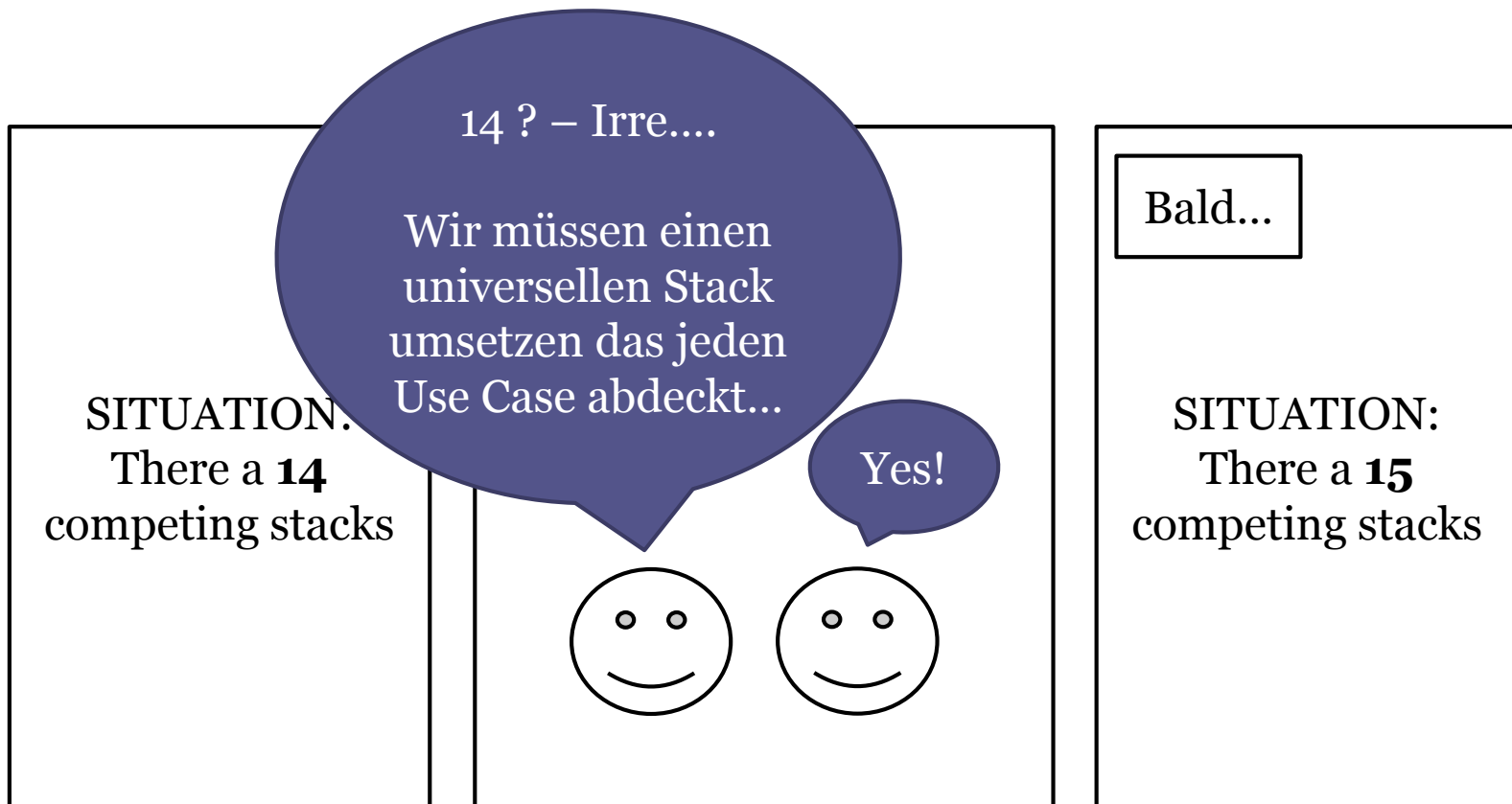
- Kein Problem, Microsoft wird Security Fixes bereitstellen, ob dass dann über Windows Update läuft.... Schauen wir mal.

# Warum jetzt Open Source ?

- .NET Core ist Open Source.
  - Eine Chance für .NET noch schneller zu wachsen auf allen Plattformen.



# Wie steht .NET Core zu den existierenden Plattformen ?



# Was ist mit dem .NET Framework ?

## .NET Framework 4.6

- Zuerst ist wohl das Ziel mit Visual Studio 2015, dass .NET Core ein Subset des neuen .NET Frameworks 4.6 ist.
- Wenn .NET Core released ist, wird es sich aber wahrscheinlich schneller entwickeln als das .NET Framework und einige Features werden dann wahrscheinlich auch nur dort enthalten sein. :o(

# Was ist mit dem .NET Framework ?

## .NET Framework 4.6

- Es wird weiterhin Updates für das .NET Framework geben....

Geplant ist ca. einmal im Jahr....

Innovationstransfer .NET Core => .NET Framework.

- Es wird aber auch Bestandteile geben, die es nur im .NET Framework geben wird z.B. WPF. - ???  
- Wie kriege ich das jetzt übereinander ?

## Was ist mit Mono ?

- Mono ist wie das .NET Framework mit all seinen Problemen und einer gewissen Komplexität.
- Die Komplexität ist auch das Problem, warum das .NET Framework nicht Open Source wird.....

## Was ist mit Mono ?

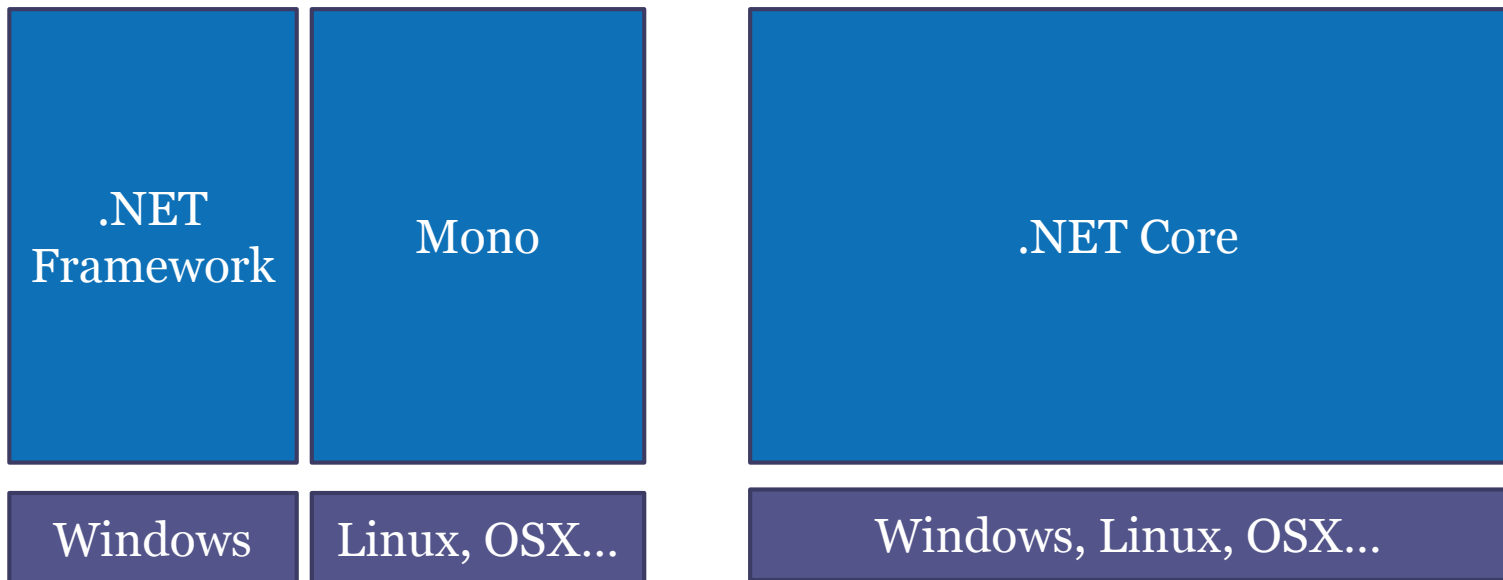
- Statement: “... Another way to look at it: The .NET Framework has essentially two forks. One fork is provided by Microsoft and is Windows only. The other fork is Mono which you can use on Linux and Mac. ....”
- Es wird mit der Mono Community zusammengearbeitet, um eine gute Plattform für Windows, Mac und Unix bereit zu stellen.

## Was ist mit Mono ?

- Wenn man das so betrachtet, dann wird Mono das selbe Schicksal erleiden wie das .NET Framework und .NET Core wird beide in Zukunft ablösen.. (So meine Einschätzung)
- Mal wieder ein harter Schritt.....den was empfehle ich meinem Kunden.... Wie spielt das Ganze mit WPF zusammen.....



# We wird die alte und neue Welt aussehen ?



# Kann ich Bestandteile des .NET Core im .NET Framework Apps benutzen ?

- Portable Class Libraries (Sharing Binaries)
  - Das wird bestimmt spannend, wenn man im .NET Framework .NET Core Sachen benutzen möchte.
- Shared project (Shared Code on Steroids)
  - Das `#if def` wird es noch eine Weile geben.....
  - Wird benutzt in Universal Apps.

# Wo bekommt man das .NET Core ?

- Visual Studio 2015 (für Windows)
- Runtime (CoreCLR) - <https://github.com/dotnet/coreclr>

	Linux	Windows	Mac OS X	FreeBSD
Debug	build passing	build passing	build passing	build passing
Release	build passing	build passing	build passing	build passing

- Framework (CoreFX) - <https://github.com/dotnet/corefx>

	Linux	Windows	Mac OSX
Debug	build passing	build running	build passing
Release	build passing	build passing	build passing
Coverage Report		coverage 46%	

# Wo stehen wir ?

- .NET Core on Linux/OS X verfügbar for Eval (64-bit only)
  - Download tar files for Linux and OS X
  - Evaluate using VM on Azure Marketplace
  - Evaluate using Dockerfile on Docker Hub
- ASP.NET 5 apps mit VS 2015 RC or Visual Studio Code
- Was als nächstes kommt
  - Remote Debugging from VS 2015
  - Local Debugging from Visual Studio Code
  - MSBuild support (aktuell erfordert es noch Mono)
  - Production readiness

# Microsofts Versprechen zur RTM...

## Plan of Record (Our Promise @ RTM)

- .NET Core apps can run in a production Linux environment, including Docker containers (both on-prem and in the cloud)
- Developers can edit, compile and debug their .NET code on Mac OS X using Visual Studio Code (or their favorite editor).
- Apps built using *platform-agnostic features* have identical behavior on Windows and x-platform
- .NET Core brings along existing .NET cloud ecosystem of libraries to Linux
- Microsoft will support, service and maintain .NET on Linux like any other Microsoft product

# Wie fange ich jetzt an ?

## Getting Ready for .NET on Linux

- If creating a brand new app on Linux
  - Start with Visual Studio 2015 RC ASP.NET 5 project templates
  - Evaluate *Parts Unlimited* ASP.NET sample on GitHub
- If moving an existing app to Linux
  - Get a head start by transitioning app to ASP.NET 5 on Windows now
  - Once app is on ASP.NET 5, moving to Linux is trivial as ASP.NET 5 on Windows and Linux are identical



API Portability Analyzer - Alpha

## Wrap Up...

- .NET Core ist ein neuer Stack, der auf die Open Source Entwicklung optimiert ist.
- Es wird mit der Mono Community zusammengearbeitet, um eine gute Plattform für Windows, Mac und Unix bereit zu stellen.
- Das Enterprise wurde nicht vergessen.
- Die Sicherheit wird nicht leiden.

# Links

- Der Blog Eintrag, auf den diese Folien basieren ist zu finden unter
  - <http://blogs.msdn.com/b/dotnet/archive/2014/12/04/introducing-net-core.aspx>
- Ein paar Informationen zu Shared Projects und Portable Class Libraries
  - <http://blogs.msdn.com/b/dotnet/archive/2014/04/21/sharing-code-across-platforms.aspx>

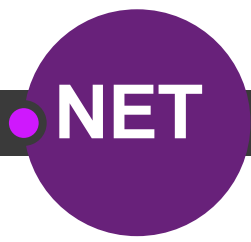


# Links

- .NET Core
  - .NET Core Runtime (CoreCLR)
    - <https://github.com/dotnet/coreclr>
  - .NET Core Libraries (CoreFX)
    - <https://github.com/dotnet/corefx>
- Build 2015 – Taking .NET Cross-Platform: Building .NET Applications on Linux and Mac
  - <http://channel9.msdn.com/Events/Build/2015/3-670>

# Links

- API Portability Analyzer (Alpha)
  - <http://www.microsoft.com/en-us/download/details.aspx?id=42678>
- Tutorial – Create Cross-Platform .NET Apps
  - <https://msdn.microsoft.com/en-us/library/f49c1308-6576-43c3-9dc7-db2eb2093aa4.aspx>



Fragen...

NET

**#NOSSUED**

**OPEN SPACE der .NET User Group Karlsruhe**

**18/19-Juli-2015 in Karlsruhe**

**[www.nossued.de](http://www.nossued.de)**