.NET Core 2.0 and Docker

MIKE BENKOVICH

mike@benko.com

Goals for today
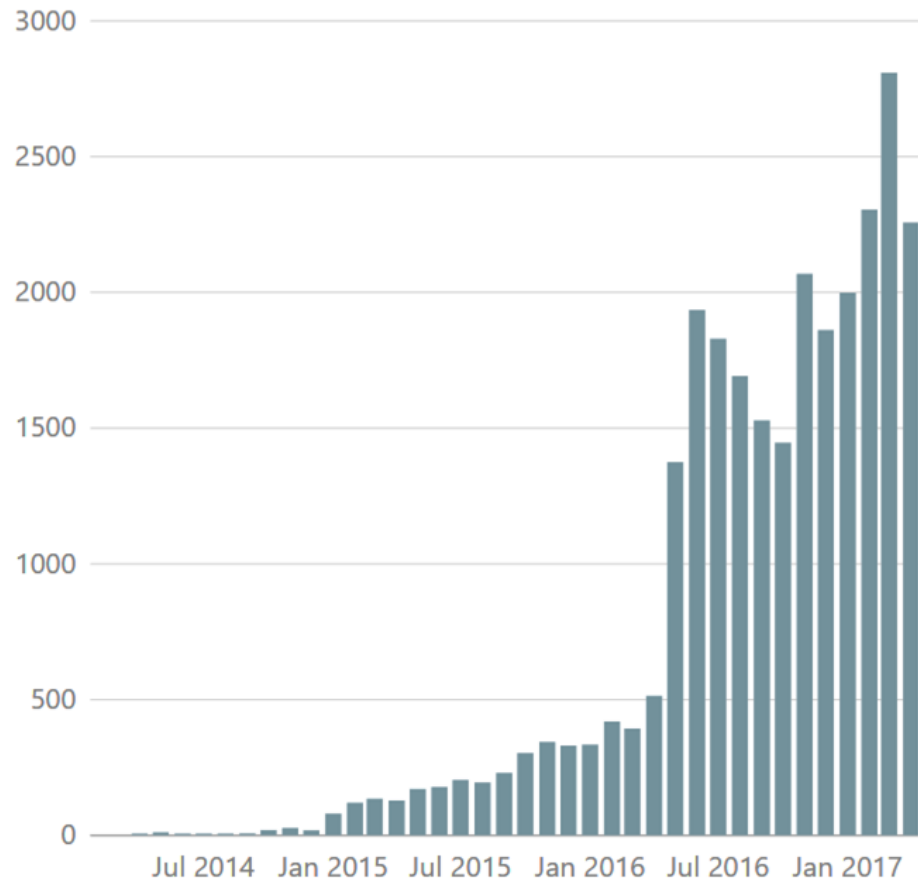
Why .NET Core?

History & roadmap

Demos

Open source momentum

"Samsung is embracing .NET because it is a completely open source project."

"ASP.NET is open source, that allows us to contribute back to it if we have any performance issues which Microsoft review and together we make a better product."

— Illyriad Games

@mbenko

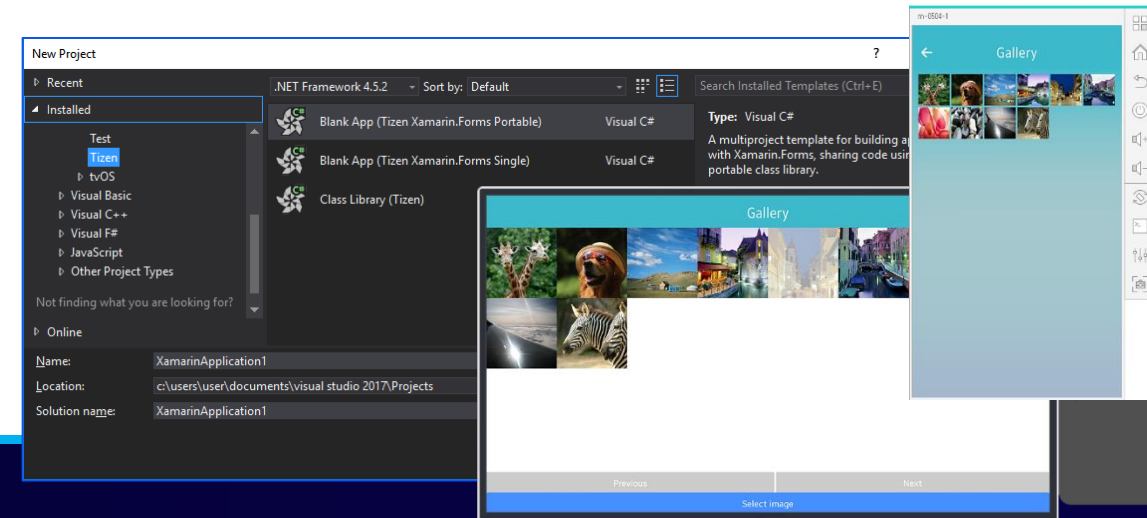# Build Tizen apps for mobile devices and TVs with .NET Core and Xamarin.Forms

Preview #1	Preview #2	Preview #3	Preview #4	Official Release
Mobile	Mobile/TV	**Mobile/TV**	Mobile/TV	Mobile/TV
Nov 2016	Jan 2017	**Mar 2017**	June 2017	End of 2017

Visual Studio Tools for Tizen
Supports everything you need to develop Tizen apps with .NET

https://developer.tizen.org/development/tizen-.net-preview/

@mbenko

.NET Core 2.0

.NET Core
.NET Standard
Containers
Orchestration
DevOps

.NET Core

A cross platform, high performance version of the .net framework rewritten to deliver best of breed characteristics to developers everywhere

.NET Core is optimized for "cloud-native" apps

Cross-platform

Lightweight

Fast

Open source

.NET Core 2.0

More APIs

More project templates

More distros

Simplified packaging

New and improved tooling in Visual Studio 2017

Visual Studio for Mac support

.NET Standard

.NET Standard library

One library to rule them all

Learn one API **across ALL .NET platforms**

Reuse your code **across ALL .NET platforms**

Easily consume third-party/OSS libraries **across ALL .NET platforms**
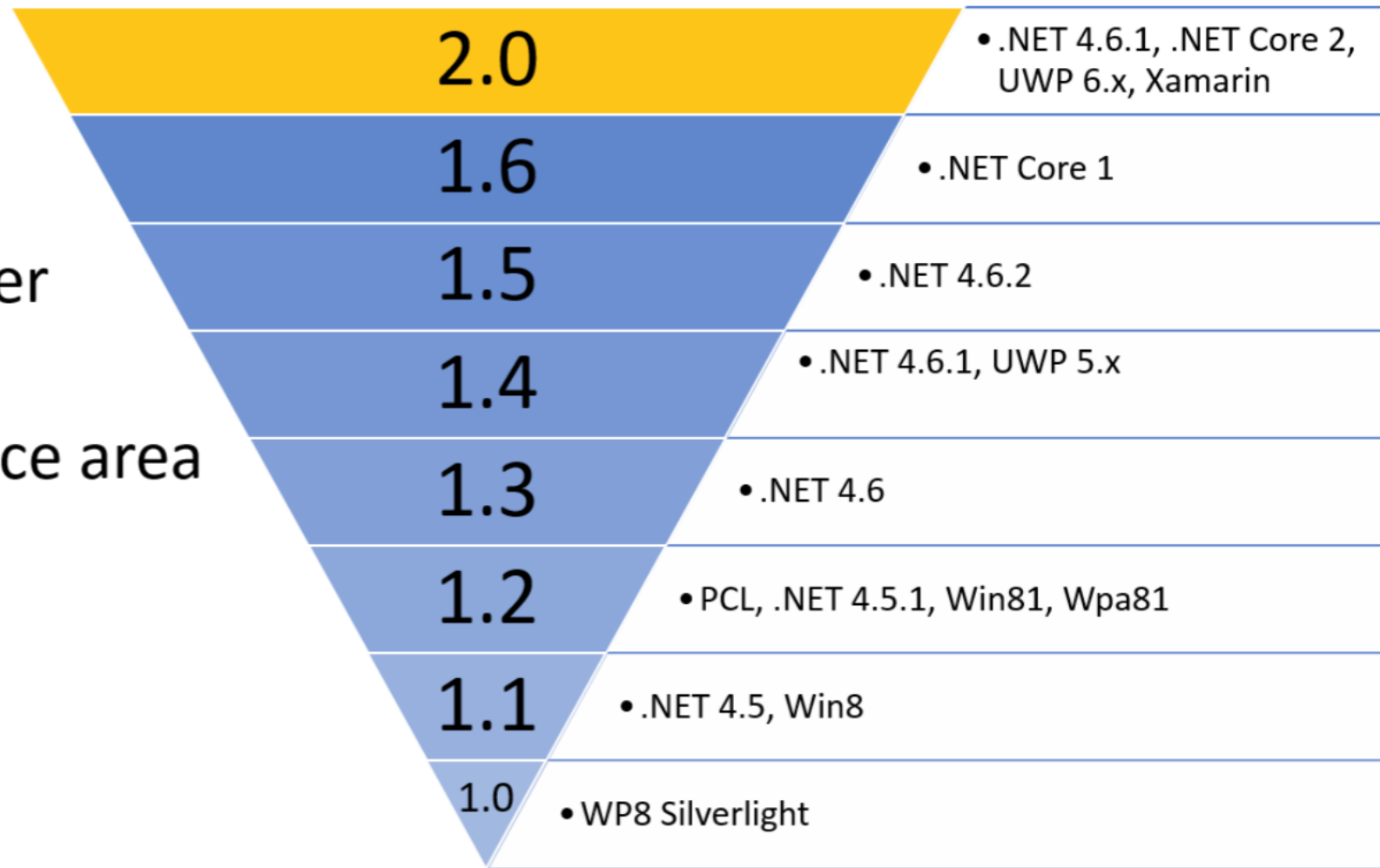
.NET Standard 1.x

Evolution of Portable Class Libraries (PCLs) with a much simpler model

Two constraints:

- API surface relatively small, because .NET Core was small
- Can't reference the vast number of libraries already available for .NET Framework

.NET Standard versions

- Currently 1.0 – 1.6

- 2.0 is in preview

- Each version defines a larger surface area

- Platforms implement surface area up to version

Version	Platforms
2.0	• .NET 4.6.1, .NET Core 2, UWP 6.x, Xamarin
1.6	• .NET Core 1
1.5	• .NET 4.6.2
1.4	• .NET 4.6.1, UWP 5.x
1.3	• .NET 4.6
1.2	• PCL, .NET 4.5.1, Win81, Wpa81
1.1	• .NET 4.5, Win8
1.0	• WP8 Silverlight

DEMO

.NET Standard Library

APIs in .NET Standard 2.0

XML	XLinq • XML Document • XPath • Schema • XSL
SERIALIZATION	BinaryFormatter • Data Contract • XML
NETWORKING	Sockets • HTTP • Mail • WebSockets
IO	Files • Compression • MMF
THREADING	Threads • Thread Pool • Tasks
CORE	Primitives • Collections • Reflection • Interop • Linq

Announcing .NET Standard 2.0

Has much bigger API surface

Extended to cover intersection between .NET Framework and Xamarin

Also makes .NET Core 2.0 bigger as it implements .NET Standard 2.0

+20K
More APIs than
.NET Standard 1.x

Can reference .NET Framework libraries

Compatibility shim allows referencing existing .NET Framework binaries

No recompile required – also covers existing NuGet packages

Limited to libraries that only use APIs that are available for .NET Standard

~70%
of NuGet packages
are API compatible

ASP.NET Core 2.0

ASP.NET Core 2.0

Faster (20% faster in Tech Empower benchmark)

Razor Pages

Razor support for C# 7.1

Smaller Publish (2.6MB vs 16.8MB)

Azure Diagnostics & Live Analytics

Angular and React templates

Authenticator App support in templates

ASP.NET Core 2.0

DEMO

ASP.NET Web App – Razor Pages

Windows Compatibility Pack for .NET Core

Microsoft.Win32.Registry
System.CodeDom
System.Configuration.ConfigurationManager
System.Drawing
System.Runtime.Caching
And many more...

PREVIEW AVIALABLE SOON

Visual Studio 2017 and Azure

Azure Functions

CI/CD support for containers & App Services

Azure Diagnostics

Application Insights

Snap Points

Cloud and Container Diagnostics

Simple
No modifications to your app
No republishing of your app

From the Azure Portal
Enable/Disable analytics and diagnostics
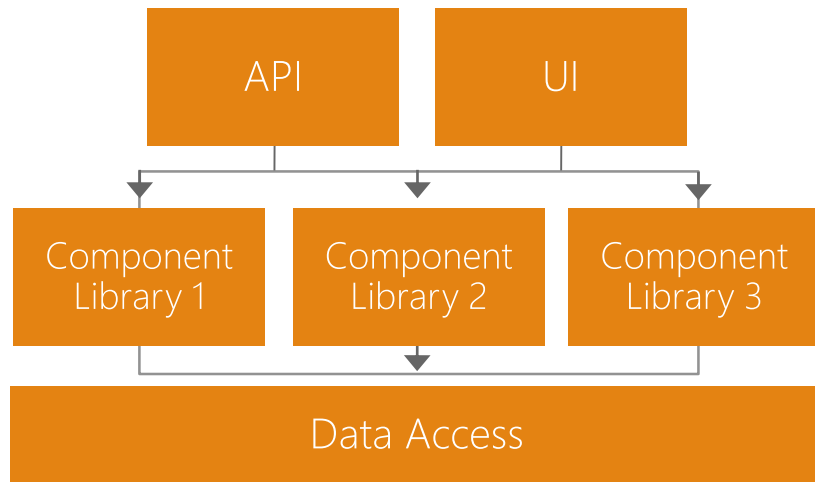See common crashes and performance problems including seeing code

From Visual Studio
See analytics data from portal in real time
Use "snap points" to debug a running application

Docker

Why microservices?

Traditional approach

Microservices approach

.NET Core and MicroServices

.NET Core Demo

.NET Architecture Guides

Available at https://dot.net/architecture

eBooks & Samples: Web, Containers, Mobile

Patterns: Health checks, queueing

More books and patterns coming

Where can I get more info?

email me – mike@benko.com

    - or -

Twitter @mbenko

Visit my site www.BenkoTIPS.com

- Resources from today's talk
- Webcasts
- Downloads
- More!