

ASP.NET Core 2 *Preview 2*

Tomáš Jecha

Microsoft Most Valuable Professional (MVP), MCSD

.NET team lead, software architect, lector

www.jecha.net

tomas@jecha.net

What is .NET Core?



Cross-platform

- Windows
- Linux
- macOS



Modular

- Only packages you need
- Lightweight deployment



Modern

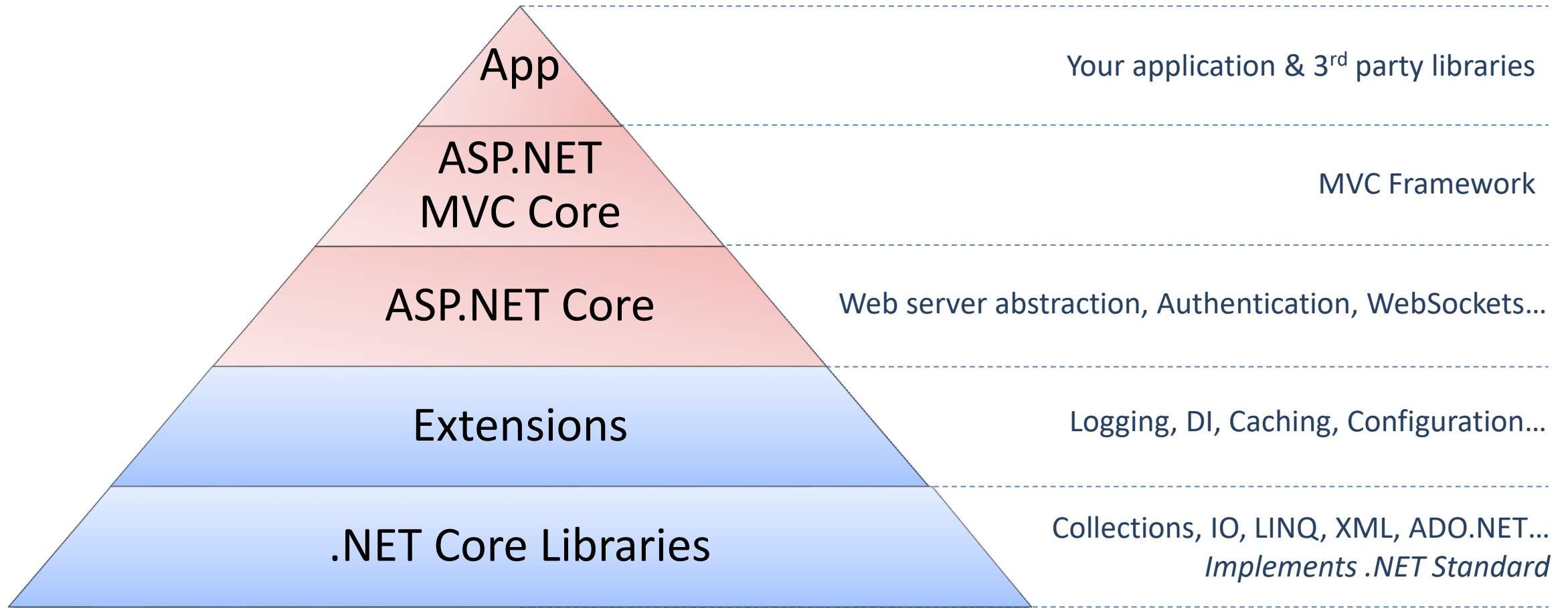
- C#
- LINQ
- Async support
- Much more!



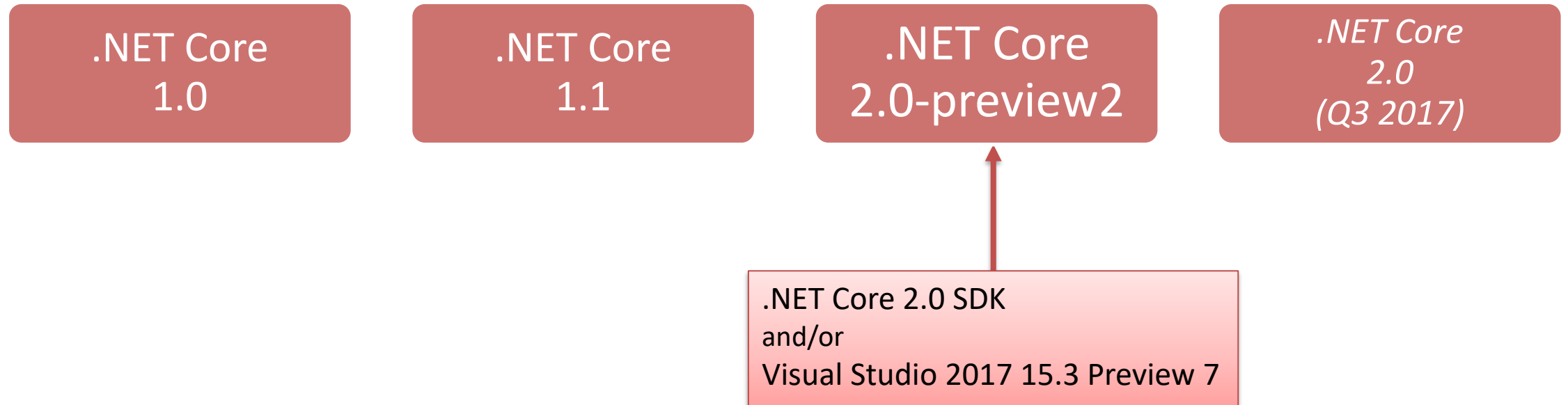
Open Source

- Runtime, libraries, compiler, languages and tools are all open source on GitHub.
- Contributions are accepted, tested and fully supported.

.NET Core Web App Stack



Current .NET Core Versions



<https://github.com/dotnet/core/blob/master/roadmap.md>

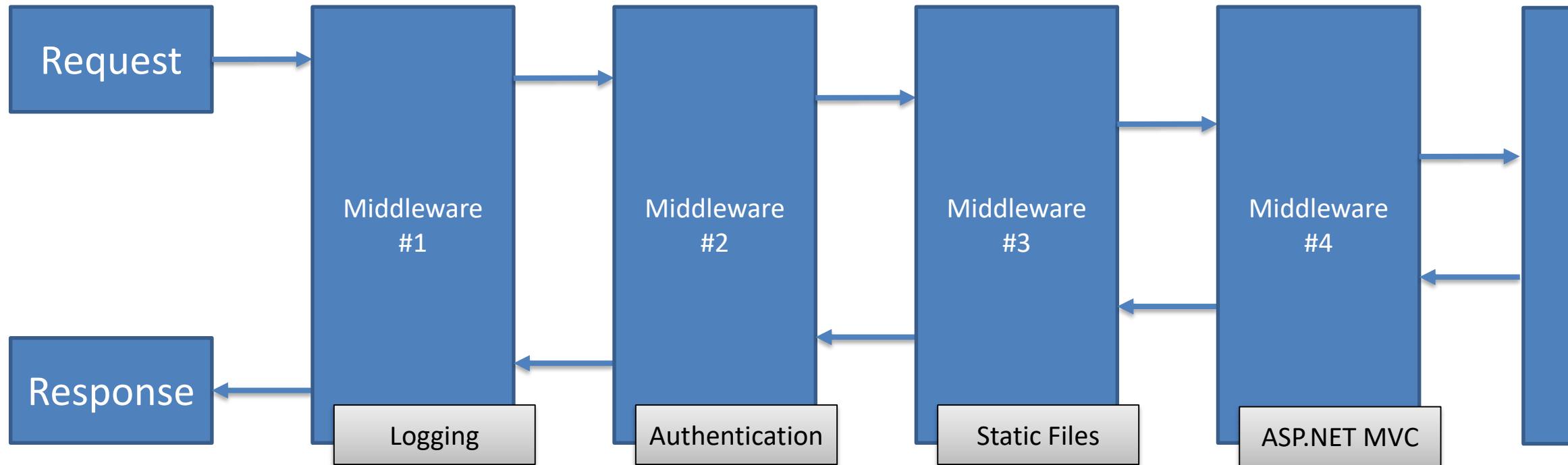
.NET Core App



ASP.NET Core

- .NET Core App project type
- Set of packages Microsoft.AspNetCore.All
- Not based on IIS – starting with clean HTTP pipeline
- HTTP pipeline is configured via Startup class
- ASP.NET MVC Core is optional part of ASP.NET Core

ASP.NET Core HTTP pipeline: Middlewares Chain



ASP.NET Core App



Microsoft Extensions

- Namespace Microsoft.Extensions.*
- Used by ASP.NET Core
- Important parts:

Logging

DI

Config

Caching

File
Providers

Identity

Localization

Configuration

- No static configuration accessors
- Configuration is optional
- Build-in sources:
 - JSON
 - XML
 - INI
 - Command line
 - Environment vars
 - Custom

Configuration & Options



Logging



Caching, file providers, default documents



Front-end features

- Visual Studio Bower integration (bower.json)
- Visual Studio NPM integration (packages.json)
- Task runner integrations (gulp, WebPack)
- Templates with CSPROJ integrations
- Bundling & minification
- SPA integrations (Angular, React)

Front-end features



Cross-platform .NET Standard Deployment

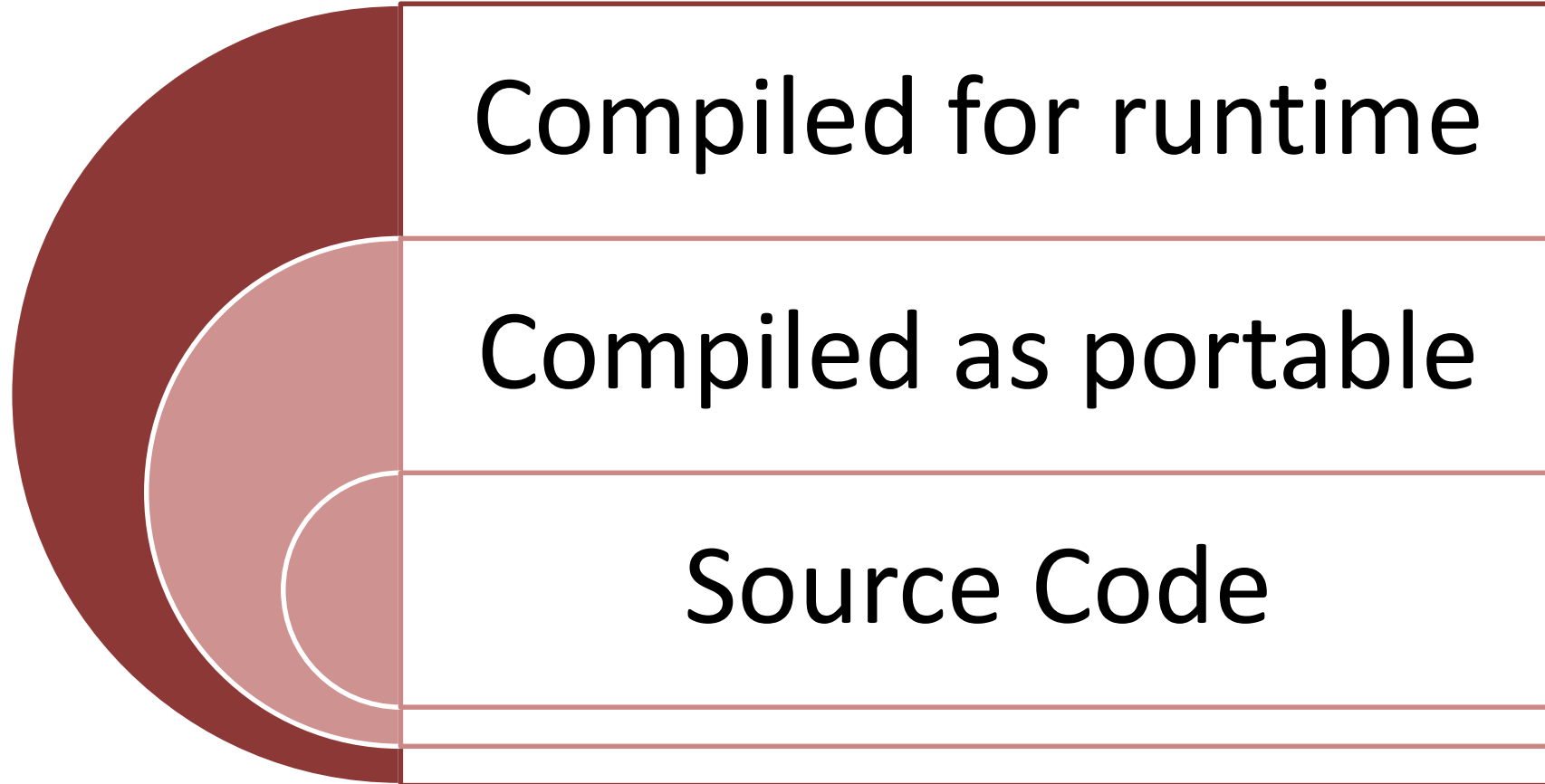
.NET Core Runtime vs .NET Core SDK

- .NET Core SDK
 - Allows you to compile
 - Installation package or binaries
 - Supports side-by-side SDK versions and runtimes
- .NET Core Runtime
 - Allows to run compiled code
 - Delivered with SDK or with application
 - Identified by:
 - Version (1.0, 1.1, 2.0-preview2...)
 - And Runtime Identifier (win10-x64, centos.7-x64...)
 - Some runtimes have prerequisites (libunwind & libc on CentOS 7 etc.)

SDK installation and versions



Deployment Strategies

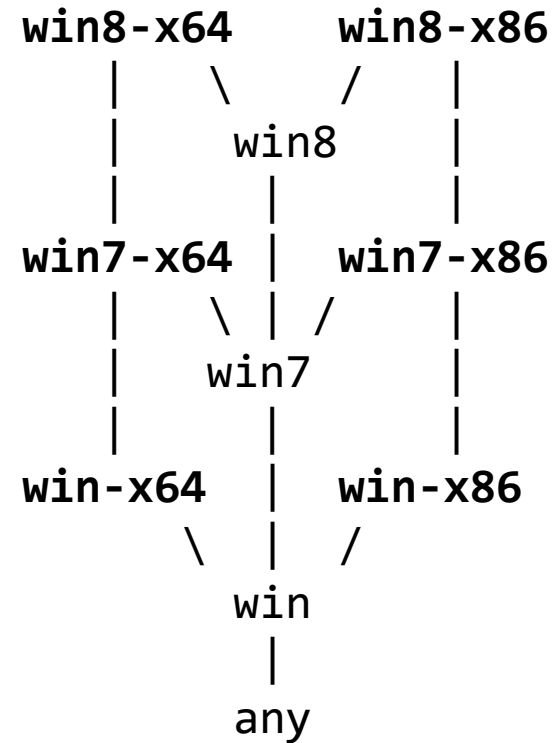


Publishing



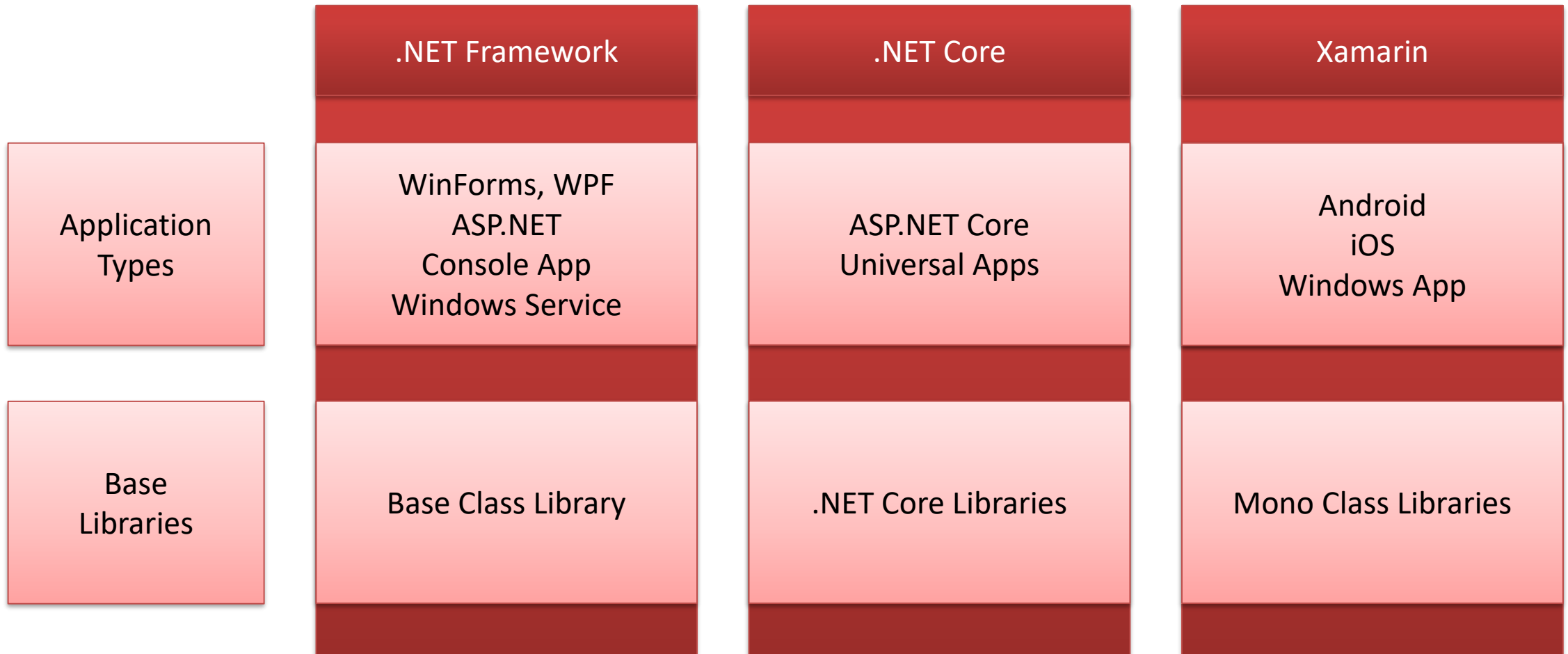
.NET Core Runtime Identifier (RID)

- RIDs are used to identify target operating systems
- [os].[version]-[architecture]
- Examples:
 - **win10-x64**
 - **win10-arm**
 - **win7-x86**
 - **centos.7-x64**
 - **ubuntu.16.10-x64**
 - **osx.10.12-x64**

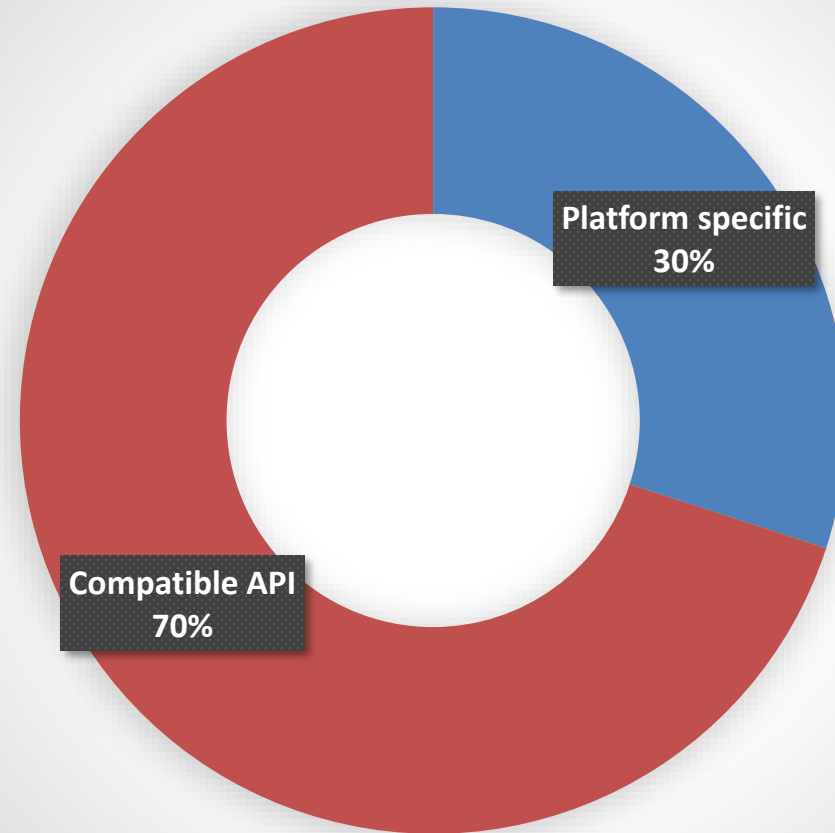


- Full catalog: <https://docs.microsoft.com/en-us/dotnet/core/rid-catalog>

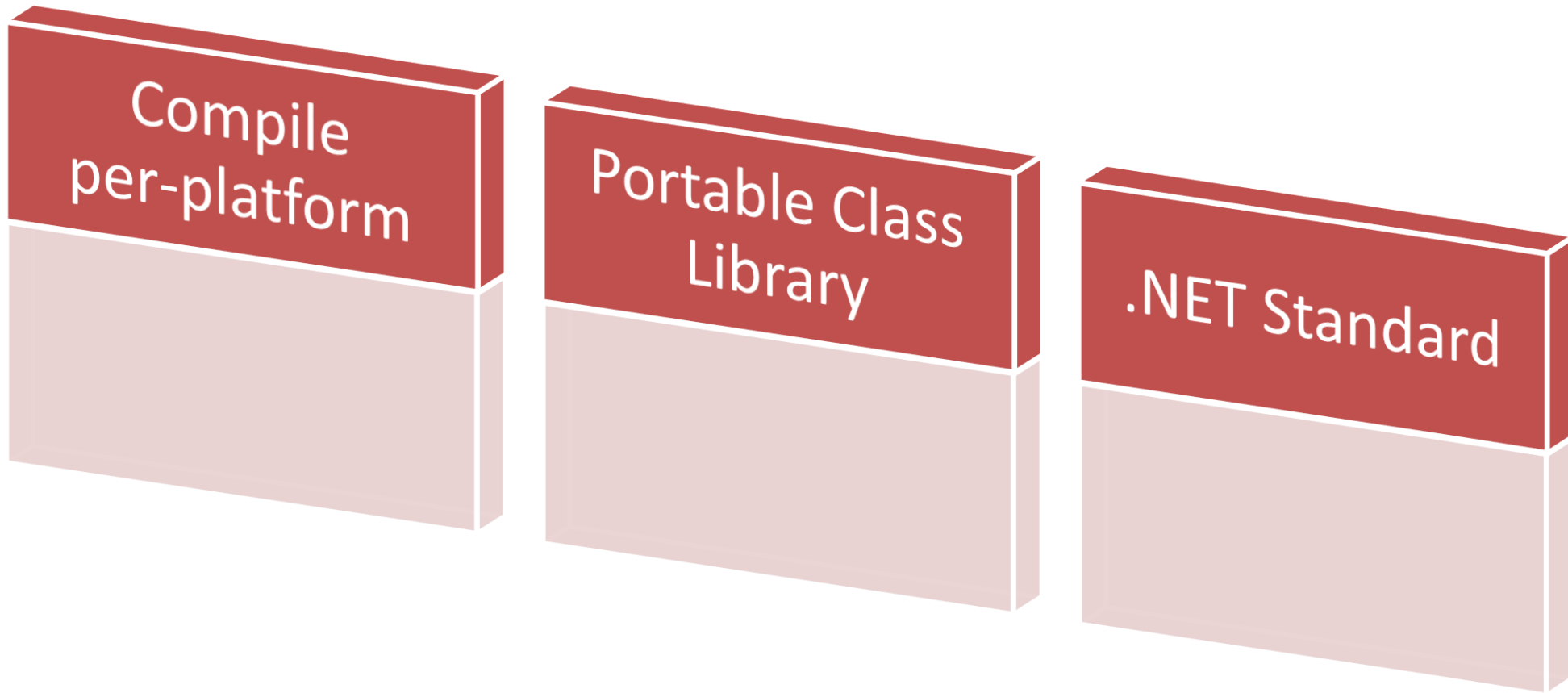
.NET platforms



NuGet packages using API available on all platforms



Sharing code with multiple runtimes?



What is .NET Standard?

- Set of APIs that all .NET platforms have to implement
- Unifies the .NET platforms and prevents future fragmentation
- .NET Standard 2.0 is implemented by .NET Framework, .NET Core, and Xamarin
- Use .NET Standard to write and share universal libraries.
- <https://github.com/dotnet/standard>
- Easy to use!

.NET Standard Libraries

Portability analyzer



.NET Standard Versioning

.NET Standard	1.0	1.1	1.2	1.3	1.4	1.5	1.6	2.0
.NET Core	1.0	1.0	1.0	1.0	1.0	1.0	1.0	2.0
.NET Framework	4.5	4.5	4.5.1	4.6	4.6.1	4.6.1 4.6.2	4.6.1	4.6.1
Mono	4.6	4.6	4.6	4.6	4.6	4.6	4.6	vNext
Xamarin.iOS	10.0	10.0	10.0	10.0	10.0	10.0	10.0	vNext
Xamarin.Mac	3.0	3.0	3.0	3.0	3.0	3.0	3.0	vNext
Xamarin.Android	7.0	7.0	7.0	7.0	7.0	7.0	7.0	vNext
Universal Windows Platform	10.0	10.0	10.0	10.0	10.0	vNext	vNext	vNext

Full compatibility matrix: <https://github.com/dotnet/standard/blob/master/docs/versions.md>

Compatibility



What's next?

- Learn .NET Core 2 & ASP.NET Core 2 when released
- Consider using .NET Standard as target for your libraries
- Install Visual Studio 2017 Preview (side-by-side)

Tomáš Jecha

www.jecha.net

tomas@jecha.net