What’s New with ASP.NET Core in .NET 5.0

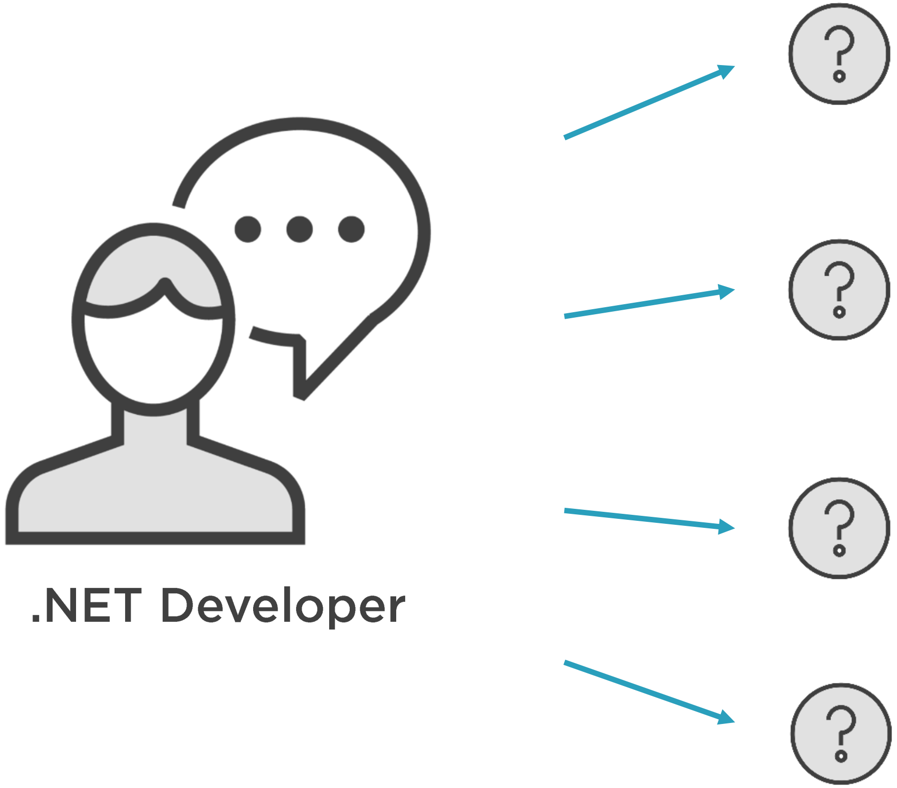
GET SETUP TO WORK WITH .NET 5.0

Alex Wolf

www.crywolfcode.com

.NET 5.0 provides a unified platform and vision.

Common Questions in the .NET Ecosystem



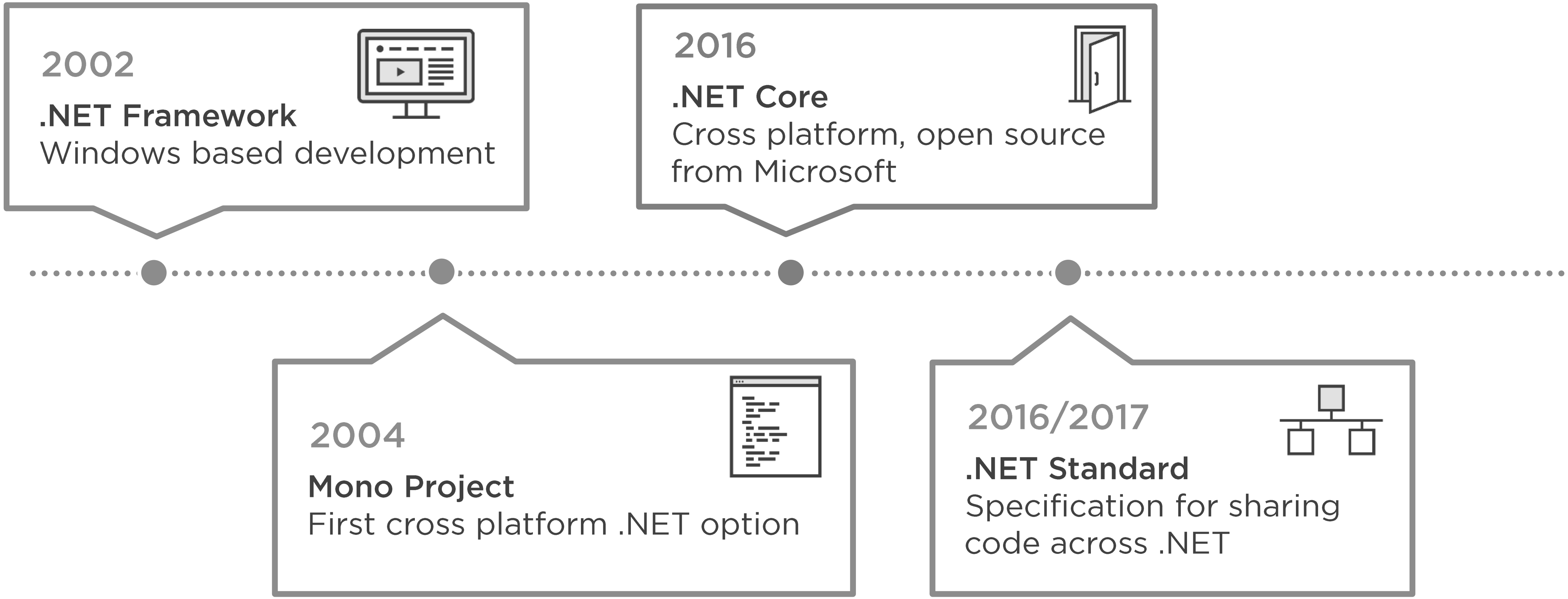
Should I use ASP.NET Framework or ASP.NET Core?

What is .NET Standard, and how does it's versioning system work?

What is Mono and how does it relate to the rest of .NET?

Should I use Entity Framework 6.0 or Entity Framework Core?

A Rapid History of .NET



2002

.NET Framework
Windows based development

2004

Mono Project
First cross platform .NET option

2016

.NET Core
Cross platform, open source from Microsoft

2016/2017

.NET Standard
Specification for sharing code across .NET

Challenges for Understanding the .NET Platform

.NET Developer

.NET Framework

.NET Core

Packages

Compatibility

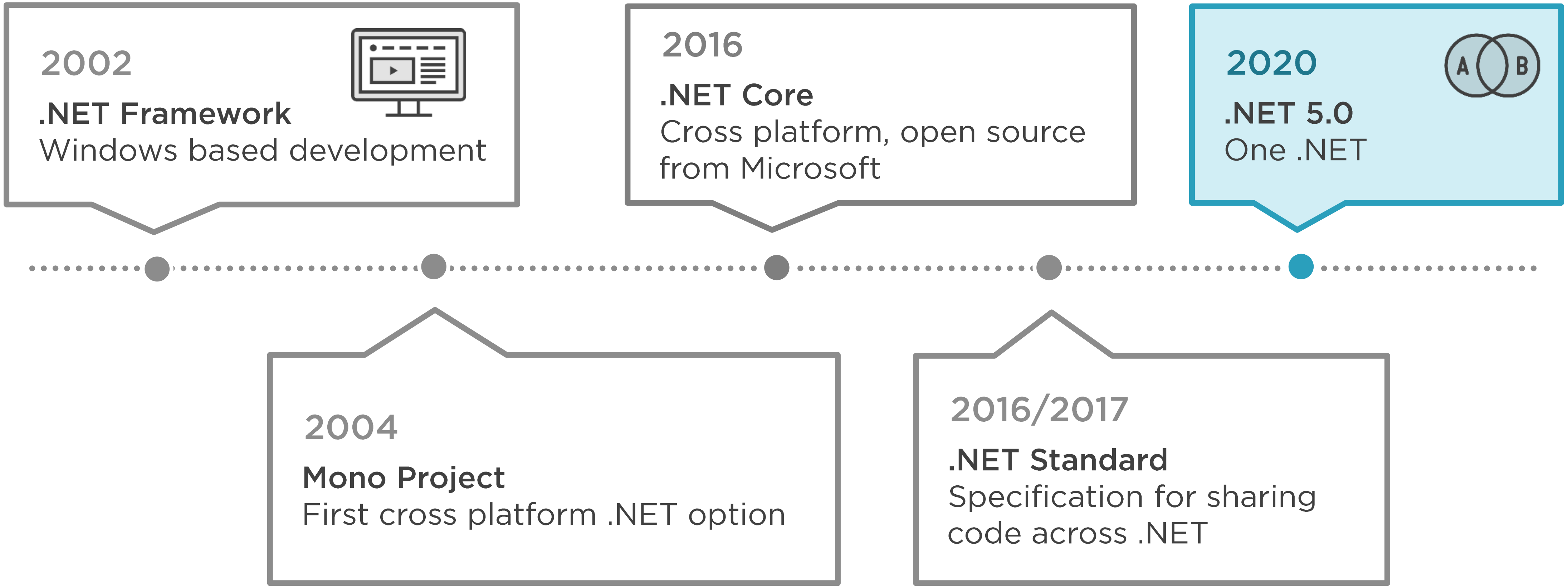
Data Access

Runtimes

Tooling

Cloud integration

A Rapid History of .NET



2002

.NET Framework

Windows based development

2004

Mono Project

First cross platform .NET option

2016

.NET Core

Cross platform, open source from Microsoft

2016/2017

.NET Standard

Specification for sharing code across .NET

2020

.NET 5.0

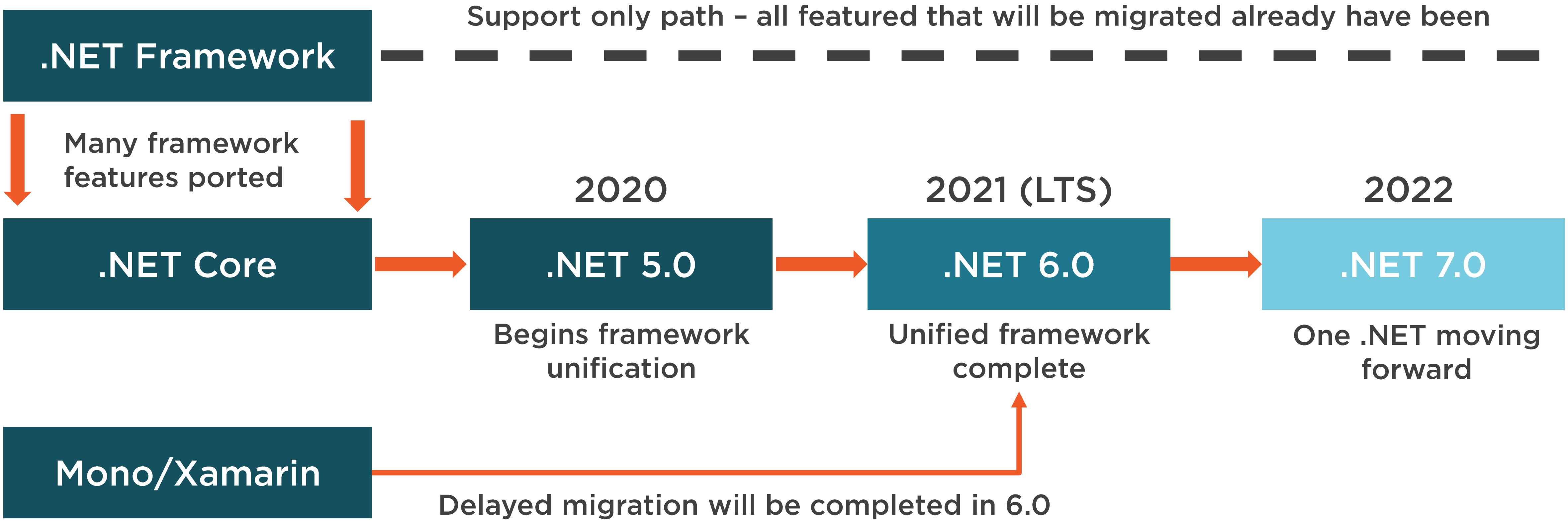
One .NET

Introducing .NET 5.0

One .NET

A unified .NET platform with a single runtime and set of APIs, languages and tools, starting in .NET 5.0 and completing with .NET 6.0.

The Future of One .NET



.NET Framework

Support only path – all featured that will be migrated already have been

Many framework features ported

.NET Core

2020

.NET 5.0

Begins framework unification

2021 (LTS)

.NET 6.0

Unified framework complete

2022

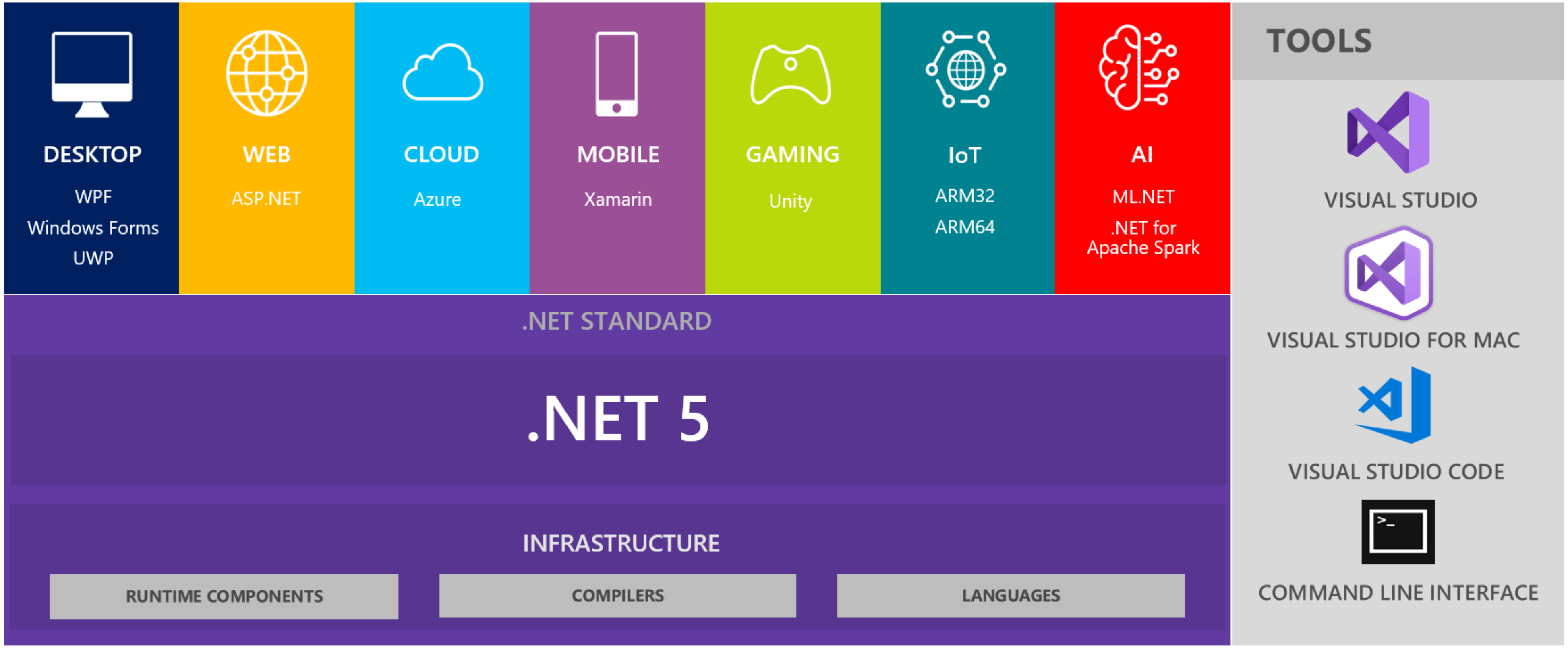
.NET 7.0

One .NET moving forward

Mono/Xamarin

Delayed migration will be completed in 6.0

One Platform, Endless Possibilities



Source: devblogs.microsoft.com/dotnet/introducing-net-5/

What does it really mean to unify .NET?

Consolidate the best features of .NET Core, .NET Framework and Xamarin

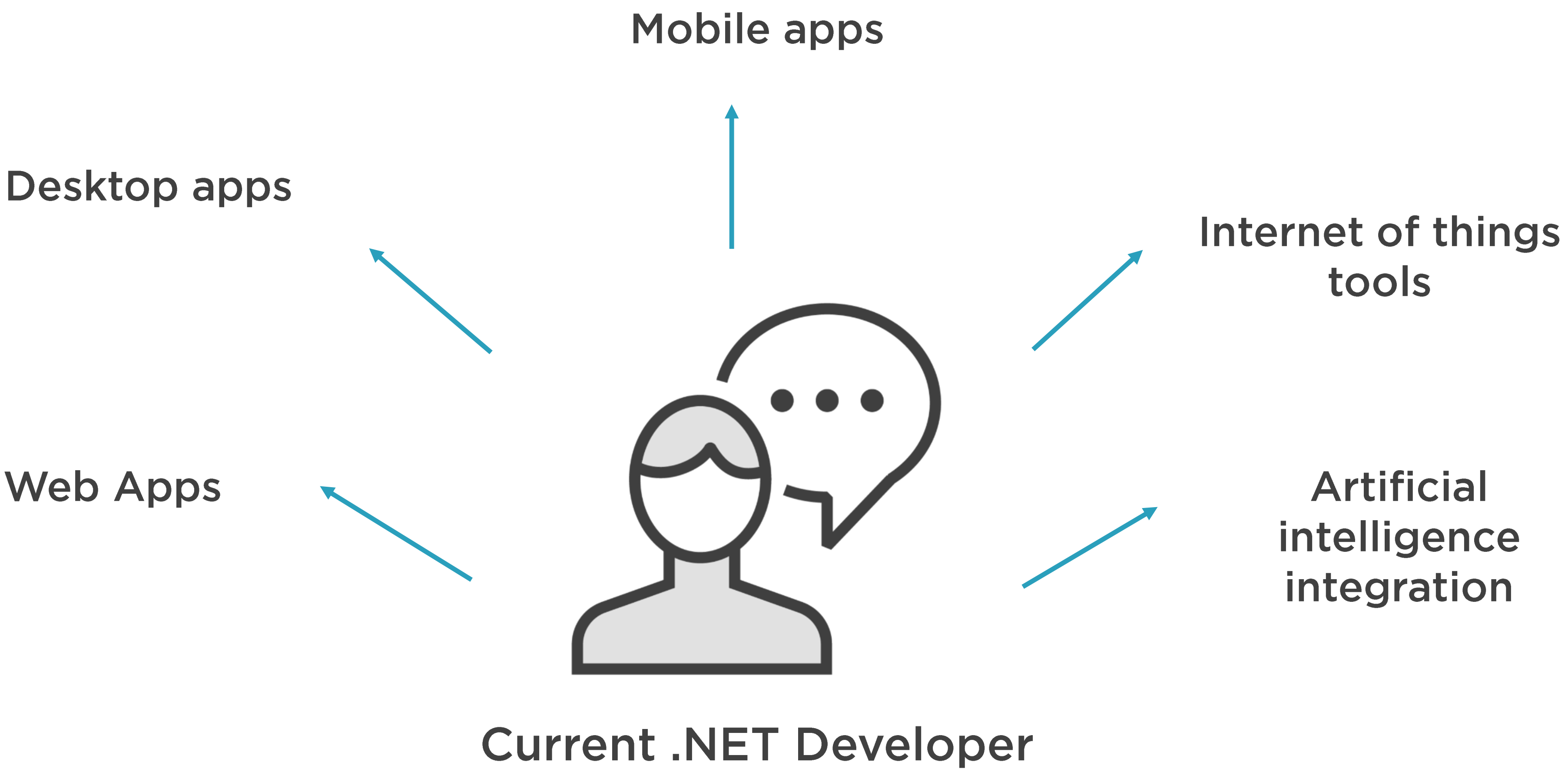
Produce a single runtime and framework APIs to be used anywhere

Improve cross platform tooling

More formally retire legacy features to support only

Standardizing versions and naming

New Horizons for Existing Developers



Mobile apps

Desktop apps

Internet of things tools

Web Apps

Artificial intelligence integration

Current .NET Developer

“So .NET 5.0 is just the next version of .NET Core right? ”

- Some developers

Understanding .NET Versioning

Framework	Major Versions						
One .NET	X	X	X	X	5.0	6.0	7.0
.NET Core	1.0	2.0	3.0	X	X	X	X
.NET Framework	1.0	2.0	3.0	4.0	X	X	X

Returning and Current Features in .NET 5.0

Returning Features

Cross platform support

Strong containerization features

Extensible tooling

Powerful cloud integration

New Features

Significant performance improvements

C# 9.0 and F# 5.0

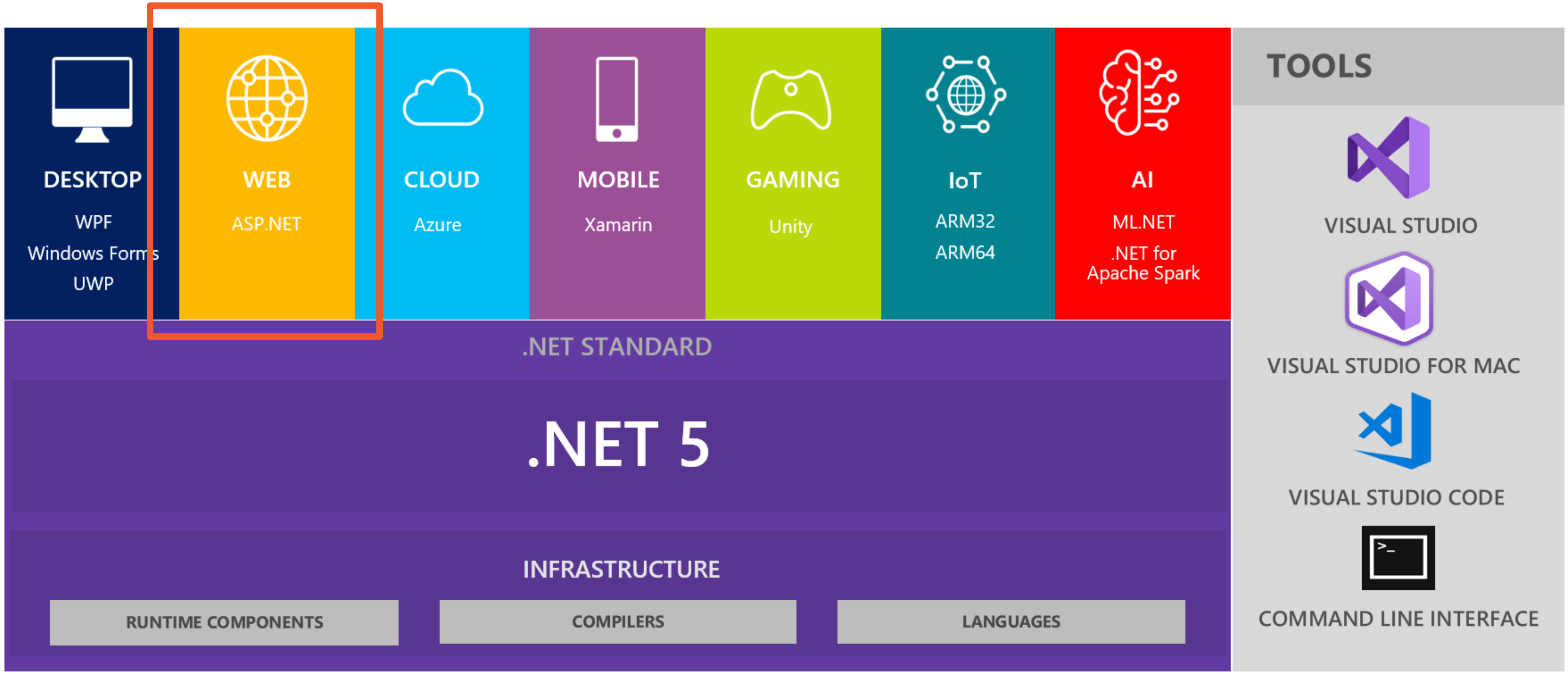
Simplified versioning and platform targets

New platform compatibility

Single file programs

Focusing on ASP.NET in .NET 5.0

Our focus



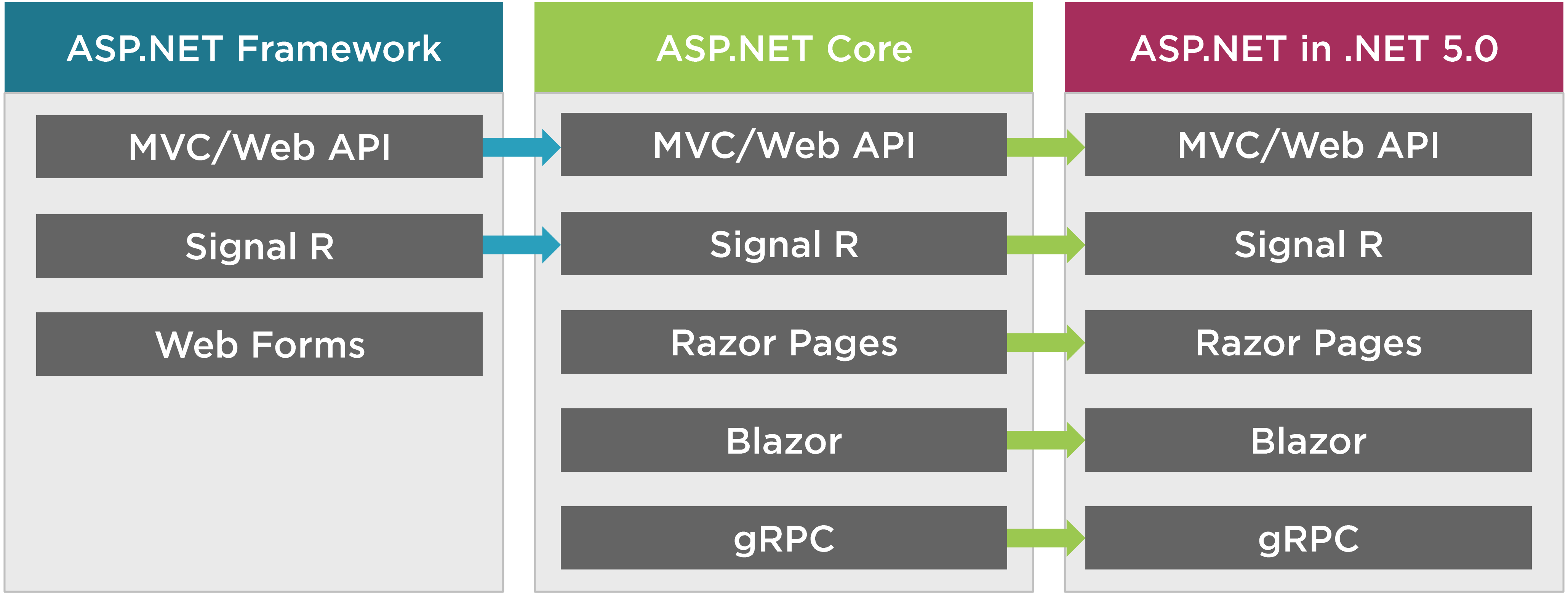
Source: devblogs.microsoft.com/dotnet/introducing-net-5/

Our Goal:

Explore the new features and enhancements ASP.NET in .NET 5.0 provides for building modern web apps.

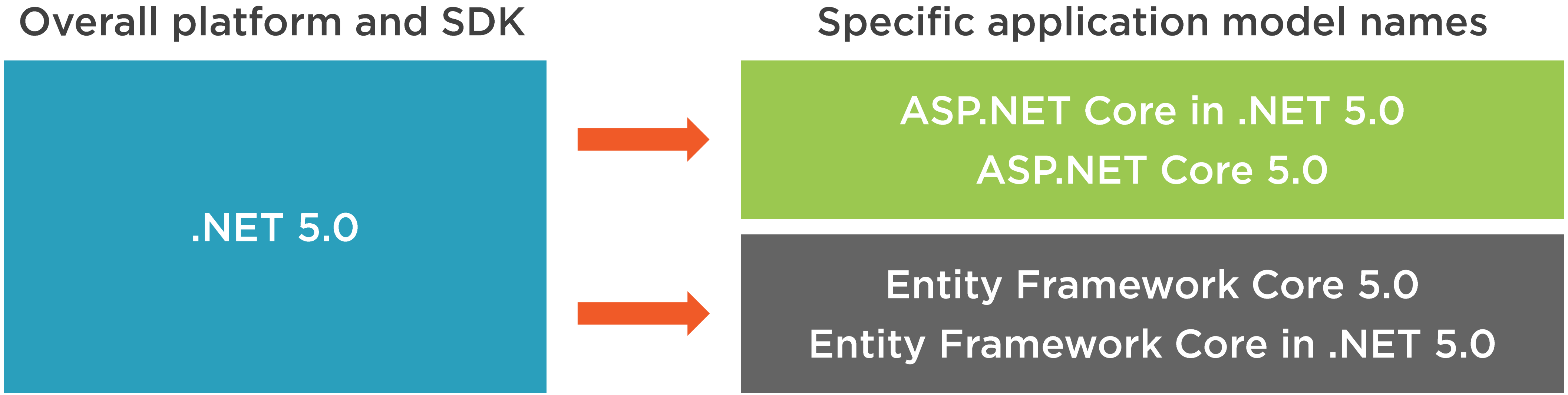
Exploring ASP.NET in .NET 5.0

Application Pattern Options



ASP.NET Framework	ASP.NET Core	ASP.NET in .NET 5.0
MVC/Web API	MVC/Web API	MVC/Web API
Signal R	Signal R	Signal R
Web Forms	Razor Pages	Razor Pages
	Blazor	Blazor
	gRPC	gRPC
Support only	Continued in .NET 5.0	The future standard

ASP.NET Naming Considerations



Overall platform and SDK

.NET 5.0

Specific application model names

ASP.NET Core in .NET 5.0

ASP.NET Core 5.0

Entity Framework Core 5.0

Entity Framework Core in .NET 5.0

Categorizing ASP.NET Enhancements

Blazor

Significant performance and feature enhancements

MVC Family and SignalR

Smaller feature enhancements

ASP.NET Platform

New features from .NET 5.0 that can benefit all applications

Performance Improvements in ASP.NET

Performance improvements across all of .NET simply by upgrading

Significant improvements to Blazor WebAssembly

.NET 5.0 is Supported on Azure

.NET 5.0 app

Azure App Services & more

Ready to go

Demo

Installing .NET 5.0

Demo

Setting up the Exercise Project

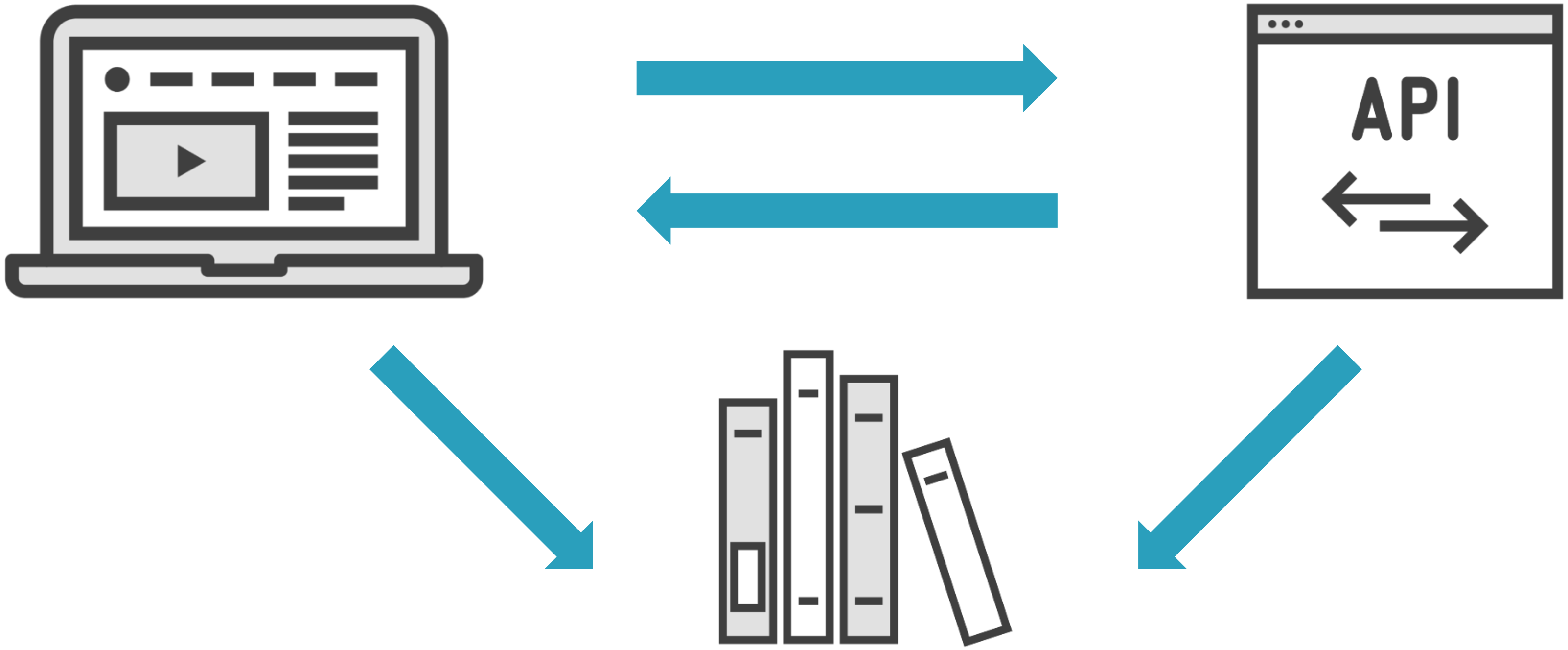
Introducing the Wired Brain Coffee app

- Online ordering
- Contact forms
- Promotions & info
- Expanding into admin features
- Dev team would like to utilize the latest features and benefits of .NET 5.0

GitHub Links

https://github.com/alex-wolf-ps/whatsnewdotnet5

The Wired Brain Coffee Architecture



Blazor UI project

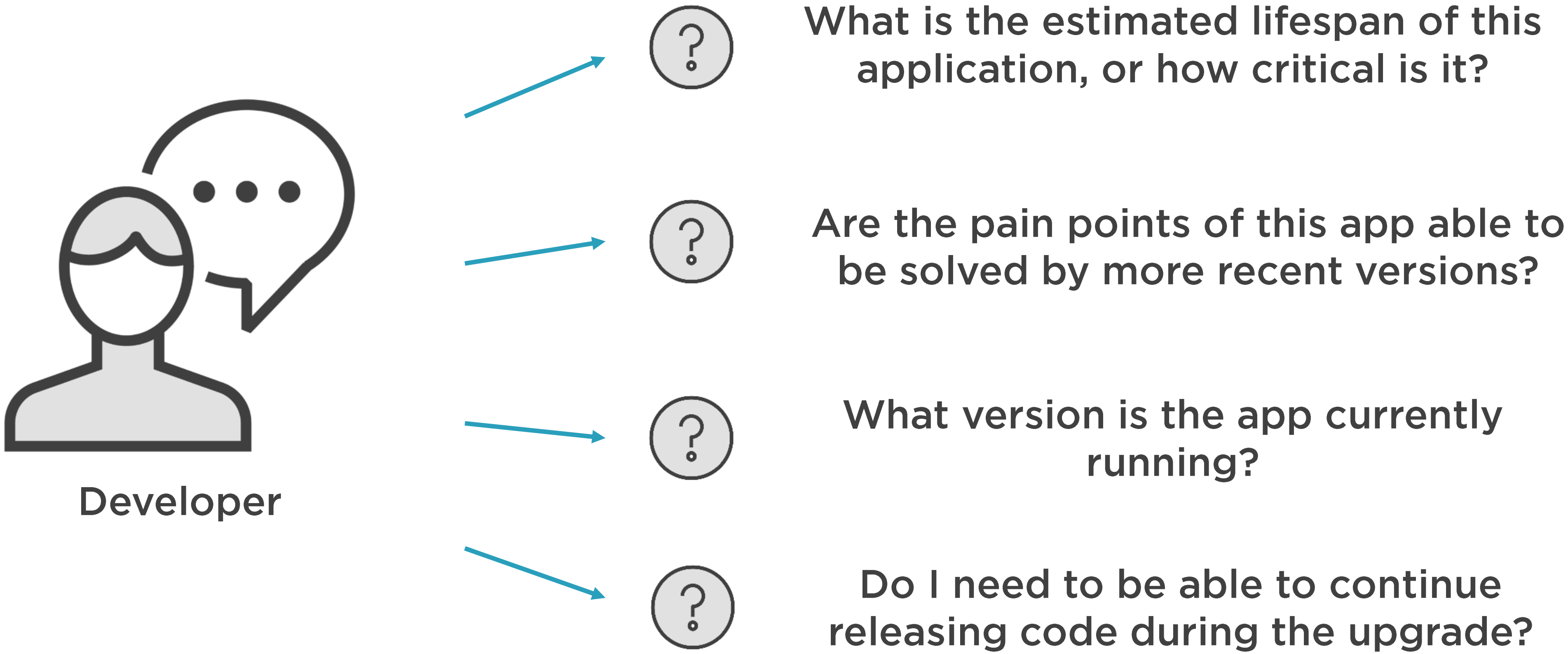
Web API project

Shared class library

Upgrade and Version Considerations

Upgrading to .NET 5.0 is painless
for projects built on recent versions.

Migration Questions to Ask



Developer

What is the estimated lifespan of this application, or how critical is it?

Are the pain points of this app able to be solved by more recent versions?

What version is the app currently running?

Do I need to be able to continue releasing code during the upgrade?

Effort Required to Migrate to .NET 5.0

Version	ASP.NET 2.0 – ASP.NET 3.1	ASP.NET 1.x	ASP.NET Framework
Level of Effort			

Migrating from ASP.NET Framework to .NET 5.0

MVC & Web API

Most application code can be ported, though framework complexities will still exist.

WebForms

Projects will require a near rewrite on top of framework migration complexities.

Upgrading from Recent Versions of ASP.NET Core

The more recent your ASP.NET Core version, the easier it will be.

Filename.code

```
<Project Sdk="Microsoft.NET.Sdk.BlazorWebAssembly">

    <PropertyGroup>

        <TargetFramework>net5.0</TargetFramework>

    </PropertyGroup>

    <!-- Package references and other code -->

</Project>
```

Update target Sdk

Update target framework

Update packages

Revisit startup and program files

A note about .NET Standard.

Guidelines for .NET Packages Going Forward

- ✓ Target .NET 5.0 to share code between only future projects in .NET 5.0

- ✓ Target .NET Standard 2.1 to share code between .NET 5.0, .NET Core 3+, recent versions of Xamarin, but *not* .NET Framework

- ✓ Use .NET Standard 2.0 to share code between .NET Framework and all other implementations

Version Specific Upgrade Details

https://docs.microsoft.com/aspnet/core/migration

Demo

Updating an Existing Project to .NET 5.0

Summary

.NET 5.0 is the first step towards One .NET

One .NET will provide a unified runtime, APIs, languages and tools

.NET 5.0 expands on .NET Core concepts and also offers plenty of new features

Performance is improved across the board in .NET 5.0 for nearly all app types

Blazor is a big focus of ASP.NET, with many performance and feature enhancements

MVC, Web API, Razor Pages and Signal R have also seen minor features updates

.NET 5.0 evolves beyond .NET Standard, while also remaining backwards compatible