

ASP.NET Core Web API Fundamentals

Getting Acquainted with ASP.NET Core



Kevin Dockx

Architect

@Kevindockx | www.kevindockx.com

Version Check



This version was created by using:

- ASP.NET Core 8
- Entity Framework Core 8
- .NET 8
- C#12
- Visual Studio 2022



Version Check



This course is 100% applicable to:

- ASP.NET Core 8
- Entity Framework Core 8
- .NET 8
- C#12



Version Check



New course versions are regularly released:

- <https://app.pluralsight.com/profile/author/kevin-dockx>



Coming Up



Course structure

Prerequisites, frameworks and tooling

ASP.NET Core: the big picture

Starting an ASP.NET Core project

Middleware and the request pipeline



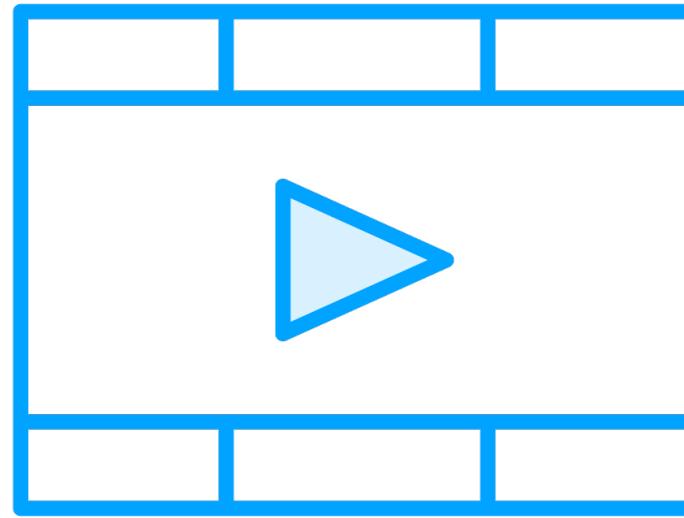
Discussion tab on the course page

X: @KevinDockx

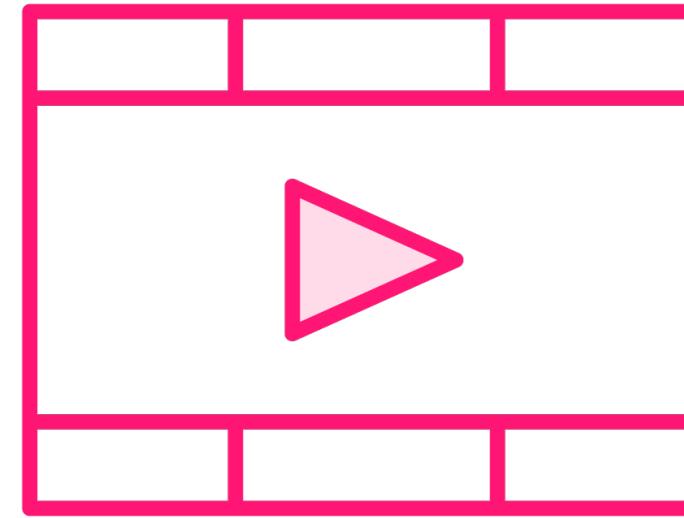
The screenshot shows a course page for "Accessing APIs Using HttpClient in .NET 6" by Kevin Dockx. The page features a dark background with a person's hands on a keyboard. At the top, there's a code editor window showing C# code for creating a movie using HttpClient. Below the code, the course title and author are displayed. A descriptive text states: "This course will teach you how to integrate with APIs using HttpClient in .NET." The bottom of the page has a navigation bar with several tabs: "Table of contents" (which is underlined in blue), "Description", "Transcript", "Exercise files", "Discussion", and "Related Courses". A pink arrow points to the "Discussion" tab.

(course shown is one of my other courses, not this one)

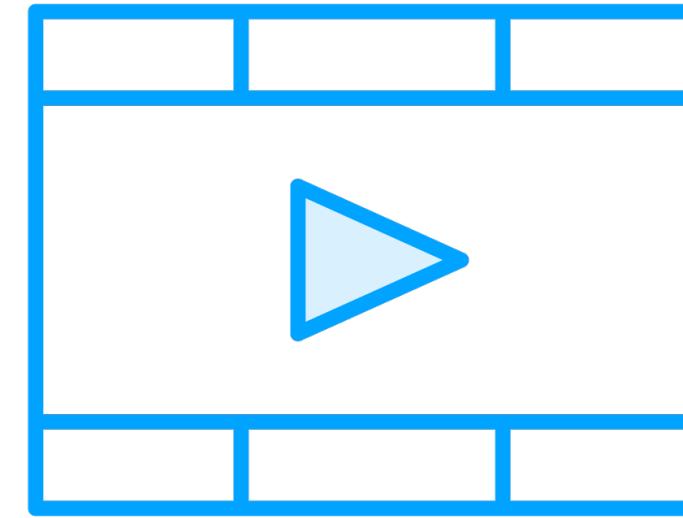
About the Course Structure



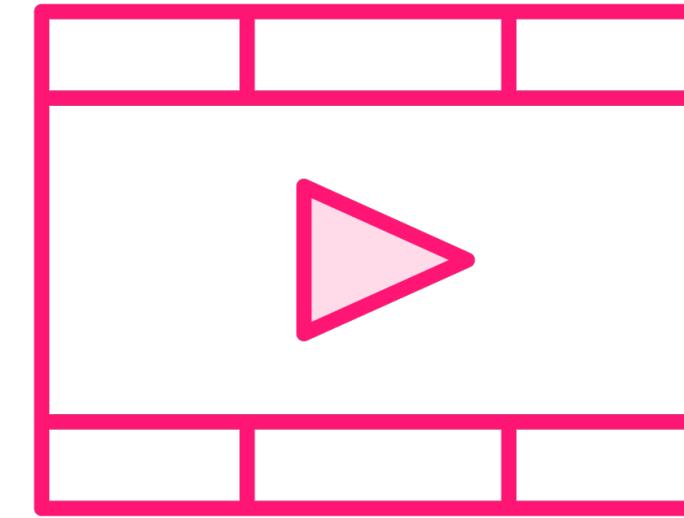
**ASP.NET Core
Web API
Fundamentals**



**ASP.NET Core
Web API Deep
Dive**



**Developing an
Asynchronous
ASP.NET Core
Web API**



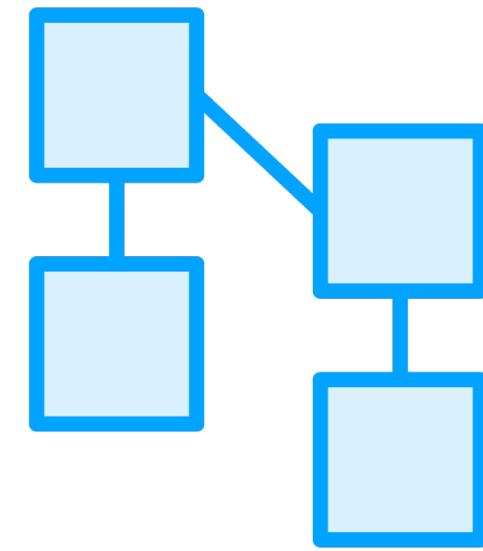
**Building
ASP.NET Core
Minimal APIs**



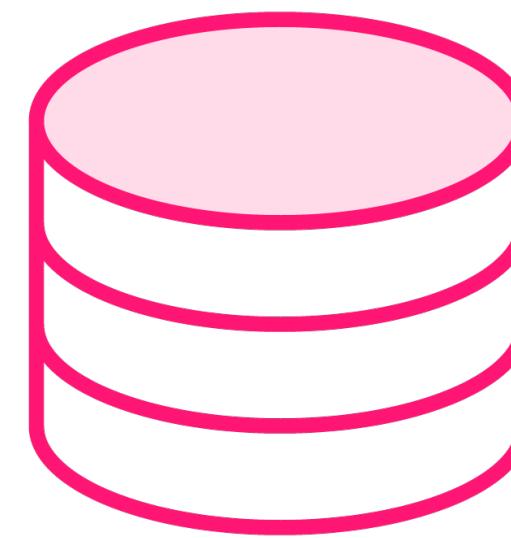
About the Course Structure



From
introducing
ASP.NET Core...



...over building
the API...



...and using
Entity
Framework
Core...



...to common
resource concerns,
security, versioning,
documentation and
deployment



Course Prerequisites

Beginner course, no specific knowledge required regarding

- ASP.NET Core
- Entity Framework Core

Knowledge of C# is required

- C# 12 is used in this course



Installing Visual Studio

Workloads Individual components Language packs Installation locations

Web & Cloud (4)

-  **ASP.NET and web development**
Build web applications using ASP.NET Core, ASP.NET, HTML/JavaScript, and Containers including Docker supp...
-  **Azure development**
Azure SDKs, tools, and projects for developing cloud apps and creating resources using .NET and .NET Framework....
-  **Python development**
Editing, debugging, interactive development and source control for Python.
-  **Node.js development**
Build scalable network applications using Node.js, an asynchronous event-driven JavaScript runtime.

Desktop & Mobile (5)

-  **Mobile development with .NET**
Build cross-platform applications for iOS, Android or Windows using Xamarin.
-  **.NET desktop development**
Build WPF, Windows Forms, and console applications using C#, Visual Basic, and F# with .NET and .NET Frame...



Exercise files tab on the course page

The screenshot shows a course page for "Securing ASP.NET Core 6 with OAuth2 and OpenID Connect" by Kevin Dockx. The page includes a large image of a computer screen displaying code in a code editor, the course title, author information, a brief description, and several action buttons: Start Course, Bookmark, Add to Channel, Download Course, and Schedule Reminder. At the bottom, there is a navigation bar with tabs for Table of contents, Description, Transcript, Exercise files, Discussion, and Related Courses. The "Exercise files" tab is highlighted with a pink arrow pointing to it from the left.

(course shown is one of my other courses, not this one)

ASP.NET Core

A cross-platform, high-performance, open-source framework for building modern, cloud-enabled, internet-connected apps



ASP.NET Core: The Big Picture



Build web apps and services, IoT apps and mobile backends



Develop on Windows, Linux and macOS



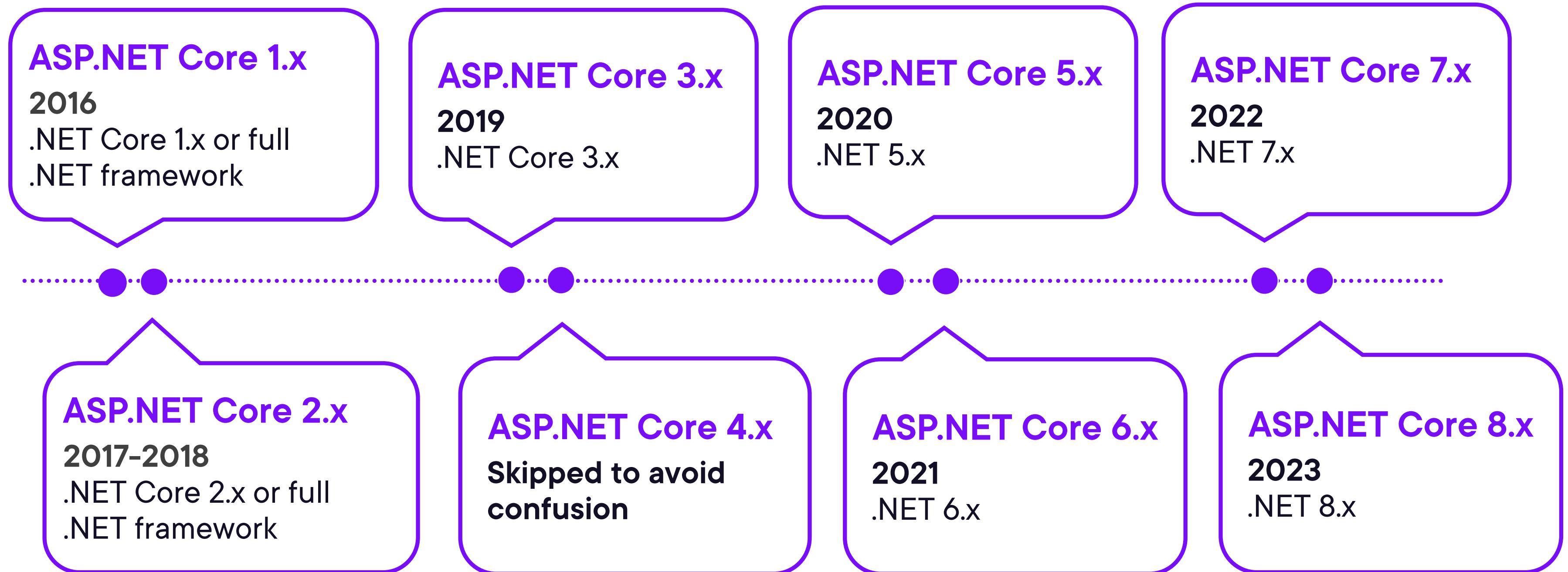
Deploy to the cloud or on-premises



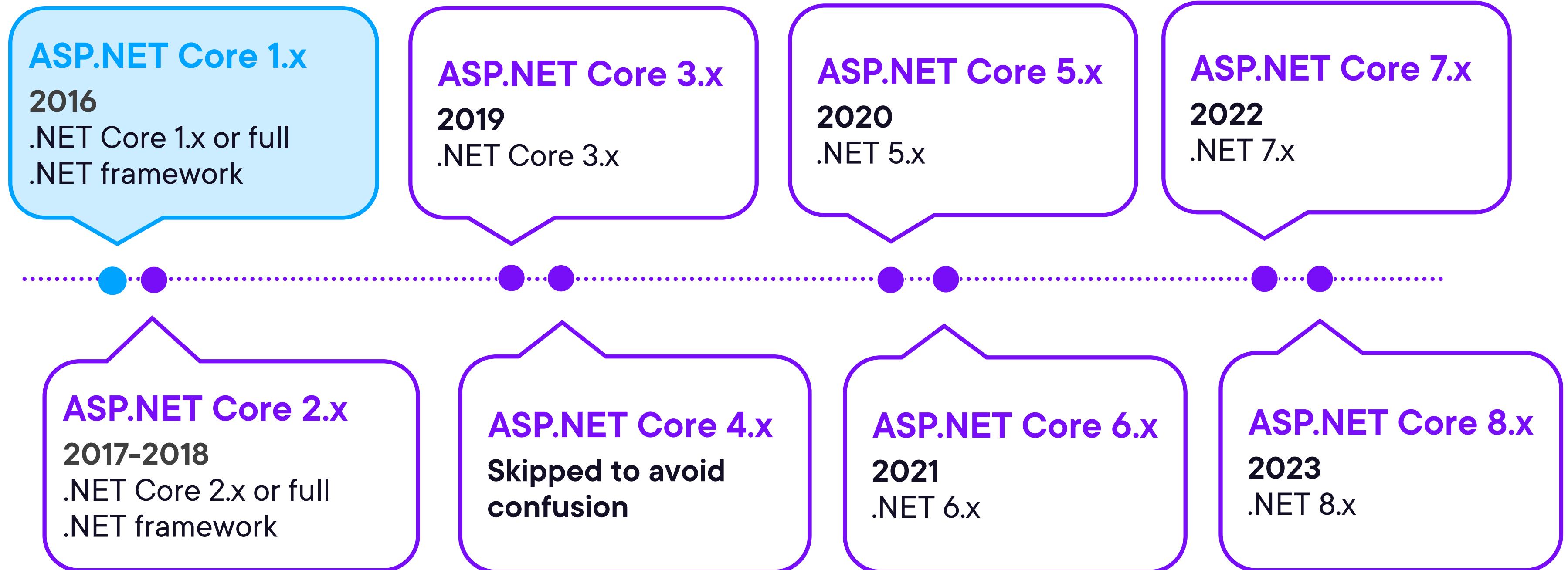
Run on .NET (or .NET Core, depending on the version)



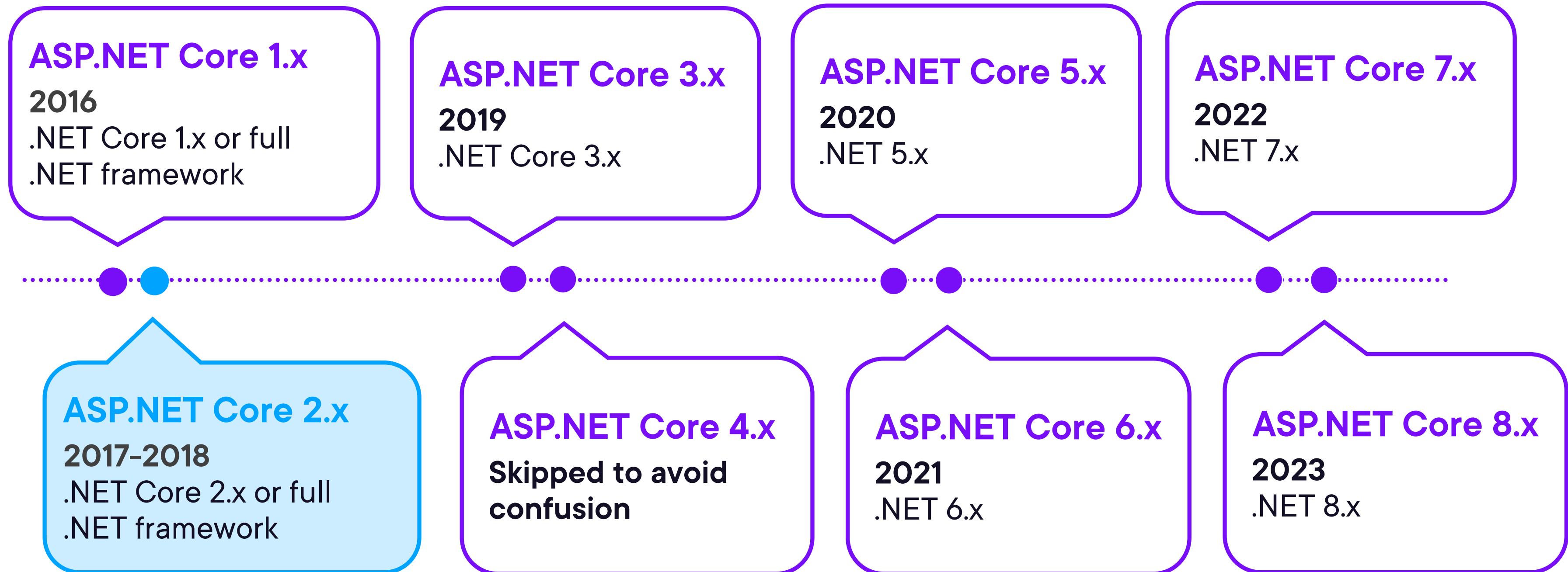
ASP.NET Core History



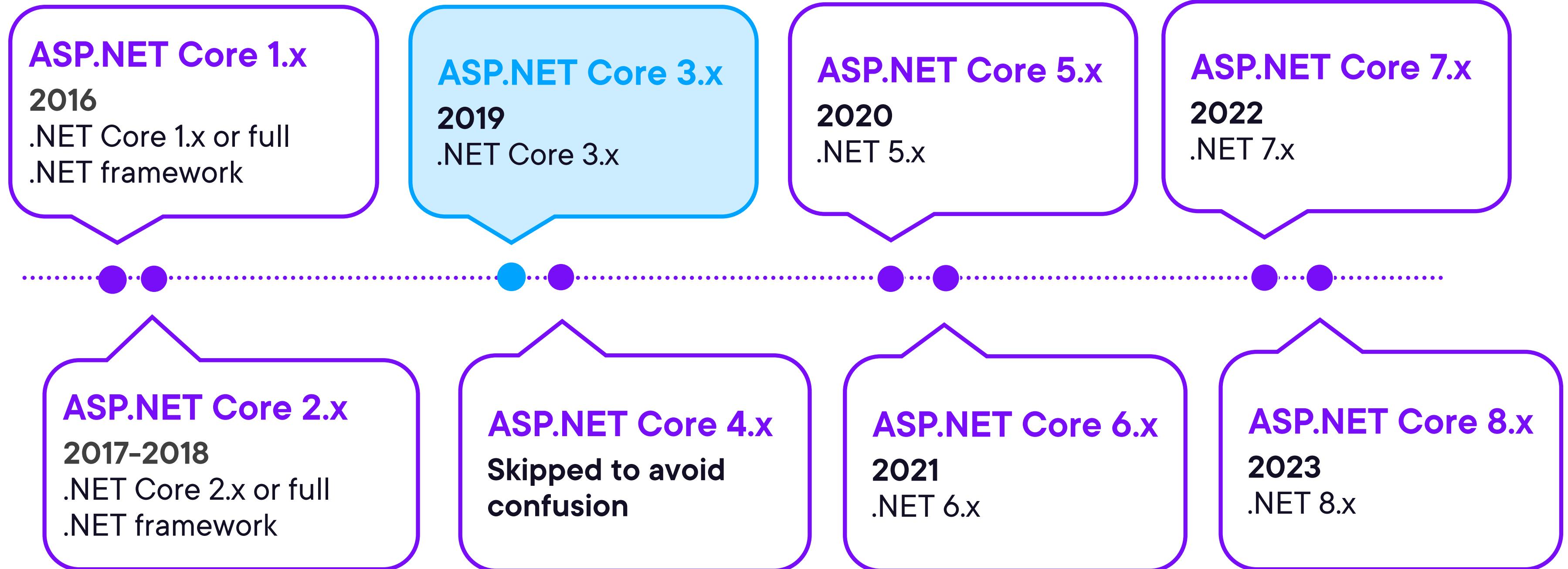
ASP.NET Core History



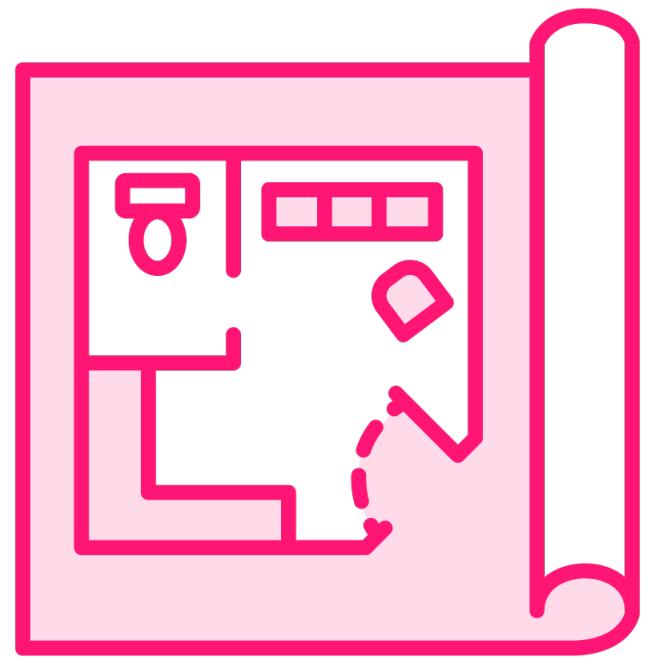
ASP.NET Core History



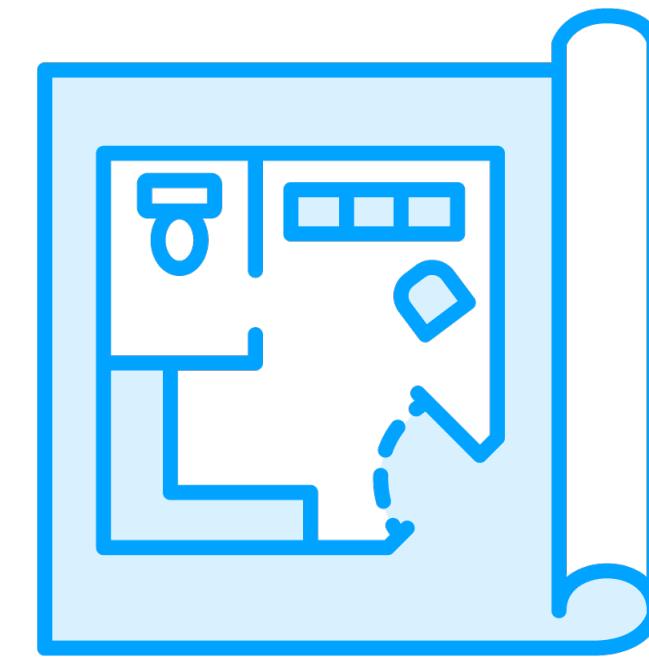
ASP.NET Core History



ASP.NET Core History



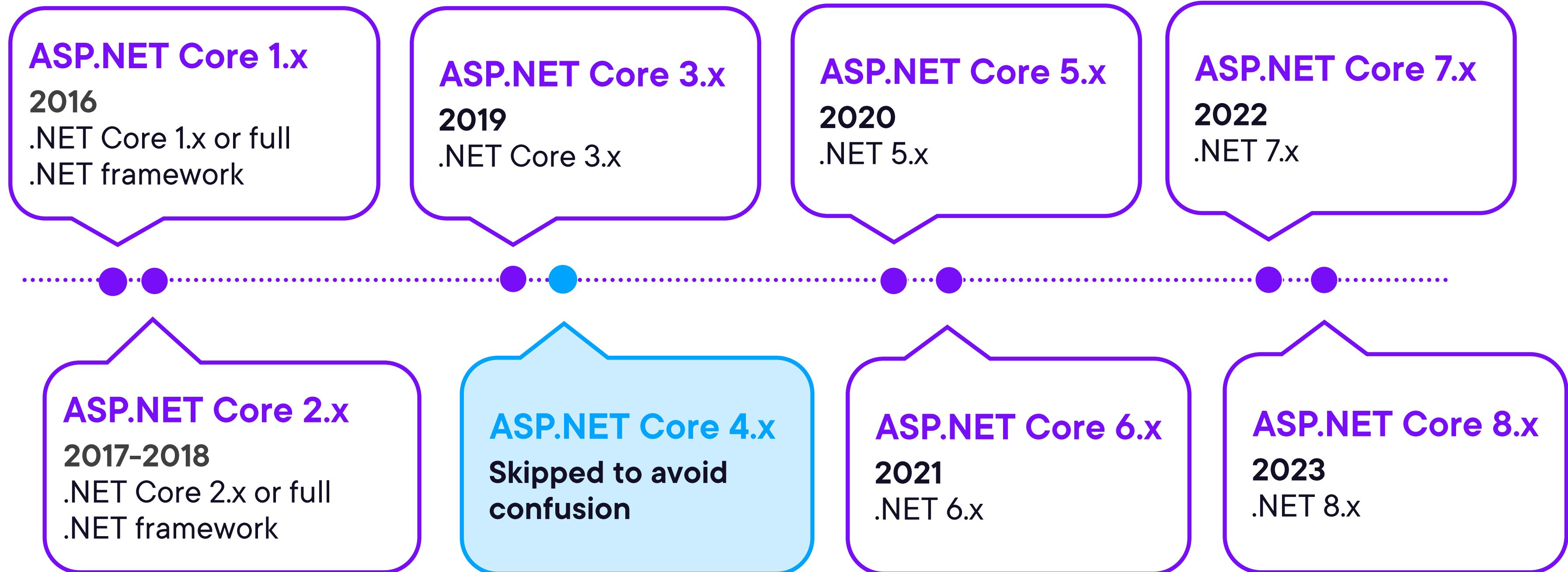
Full .NET framework (4.8)
No new development



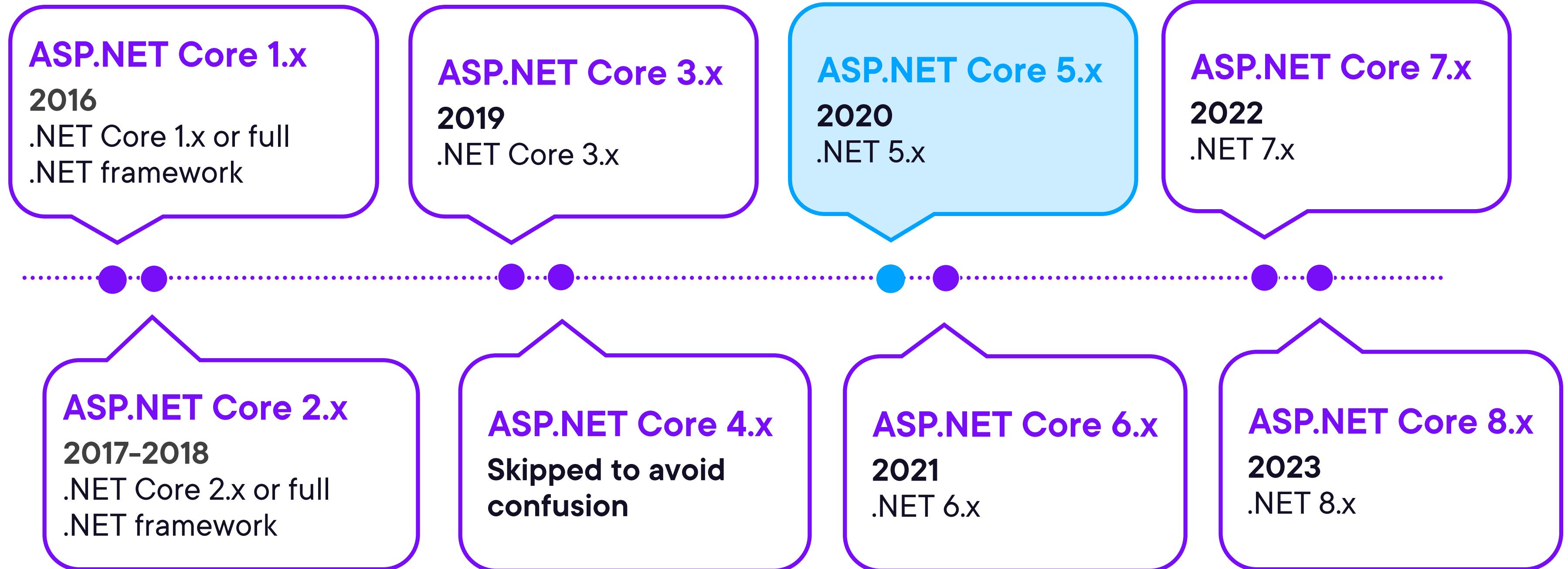
.NET Core vNext (“4.x”)
Useful functionality from the full .NET framework ported to increase performance and support cross-platform development



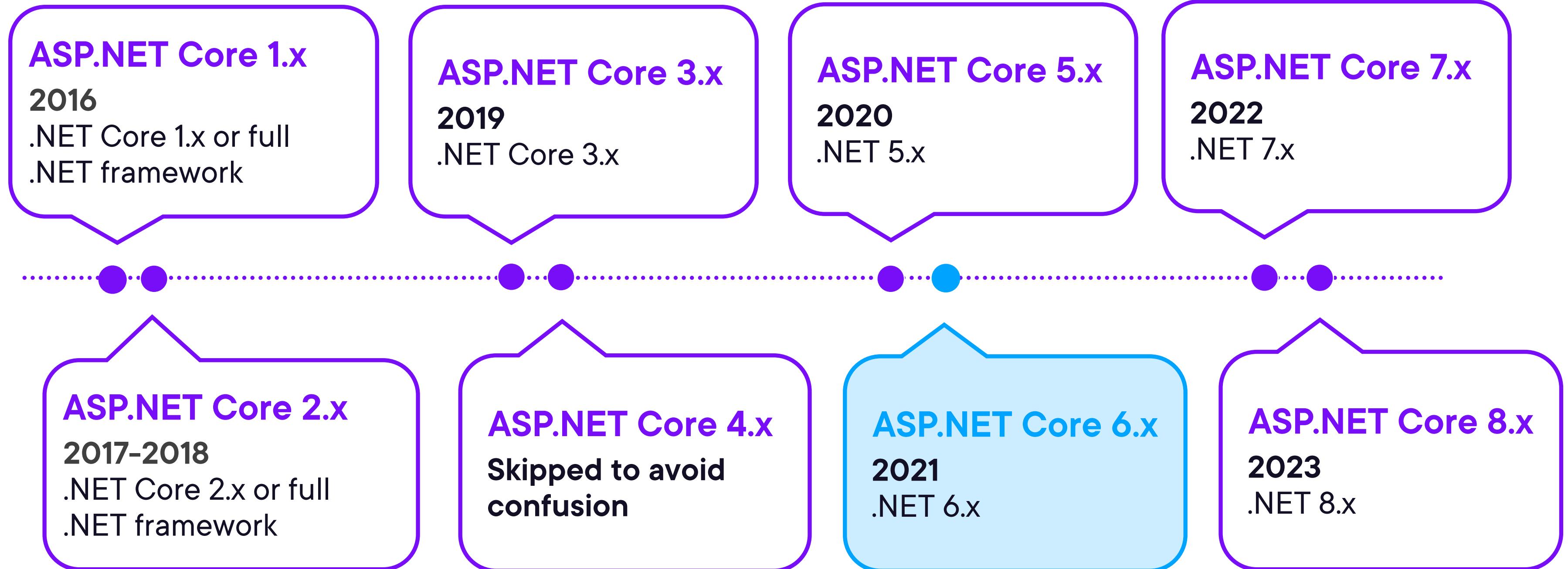
ASP.NET Core History



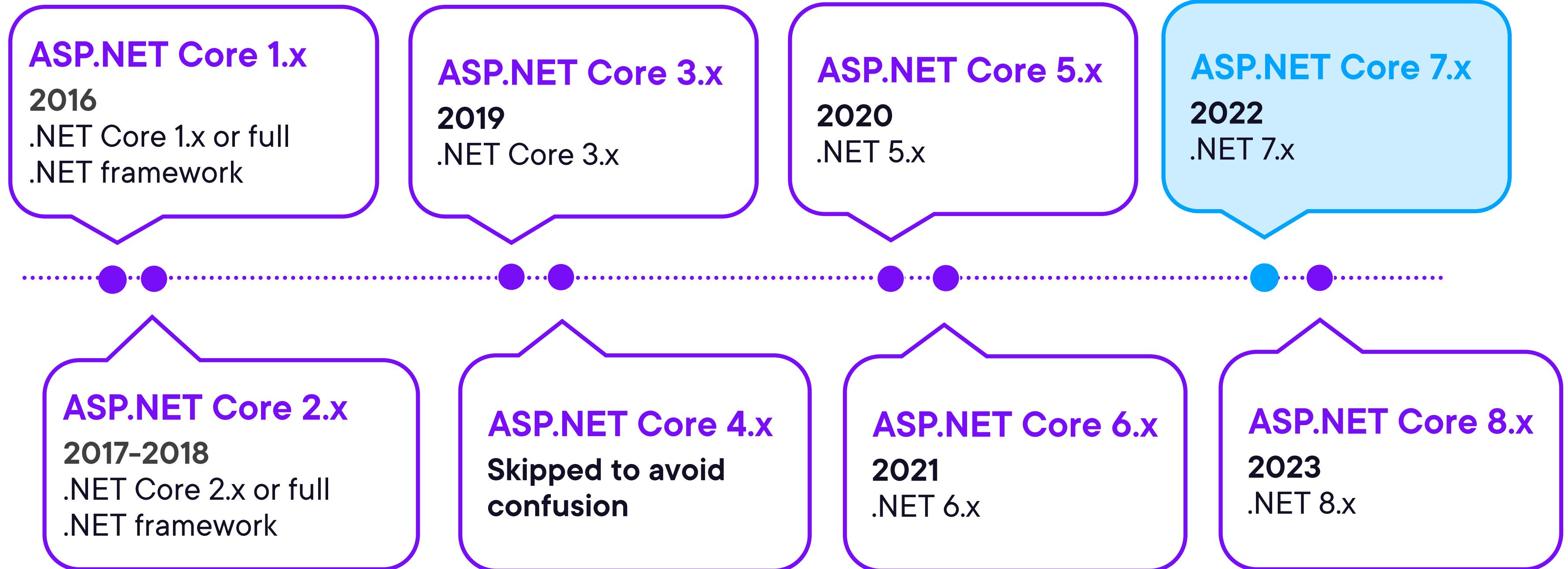
ASP.NET Core History



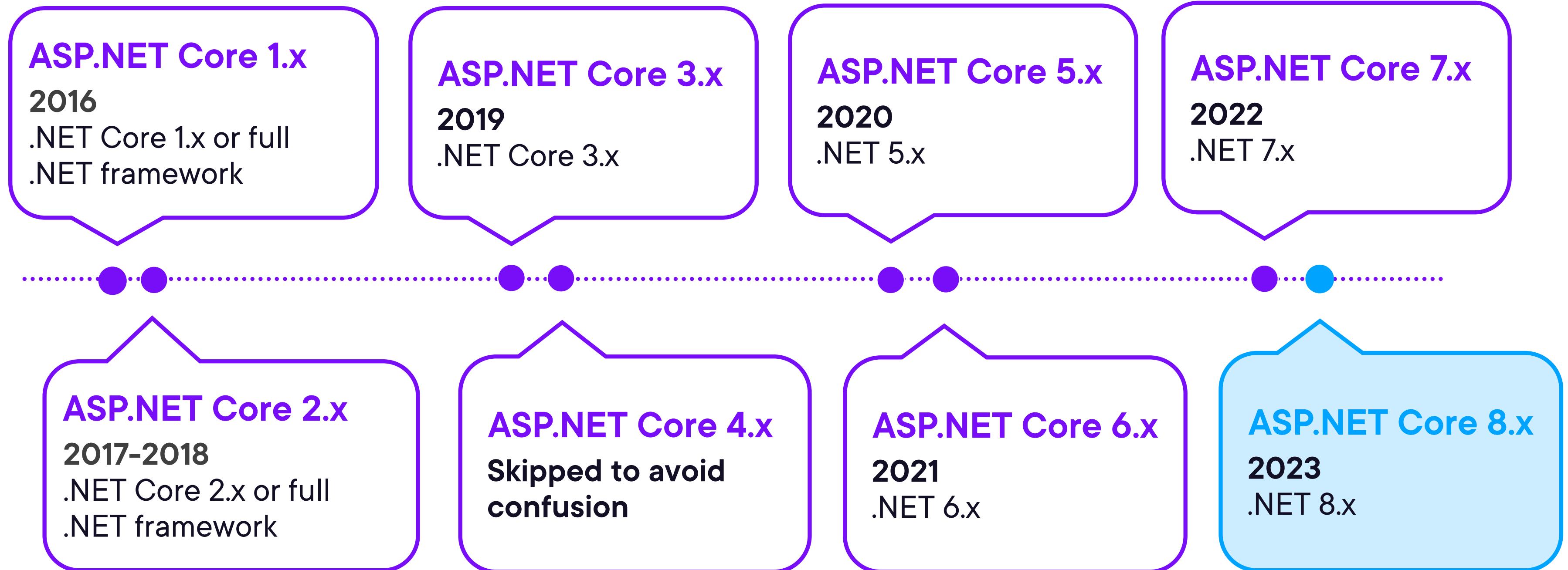
ASP.NET Core History



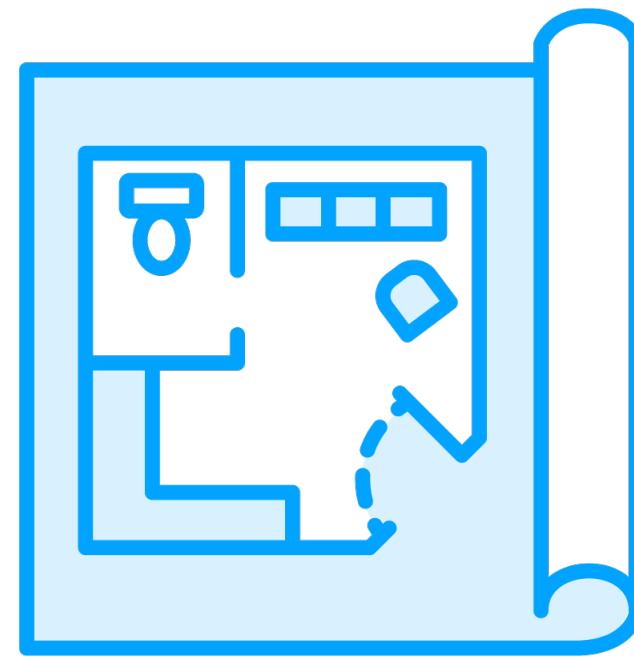
ASP.NET Core History



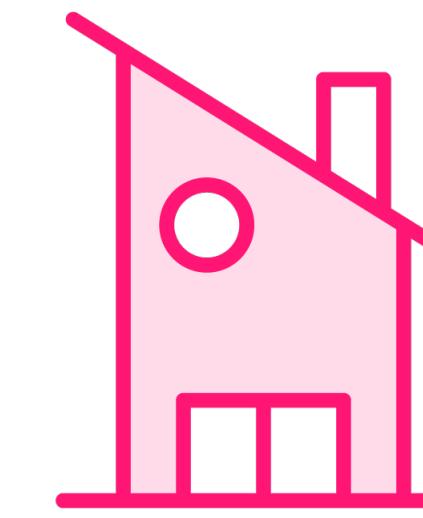
ASP.NET Core History



Approaches to Building APIs with ASP.NET Core



ASP.NET Core MVC



ASP.NET Core Minimal API



Demo



**Creating and running a new
ASP.NET Core project**



Demo



**Running an ASP.NET Core project
using the CLI**



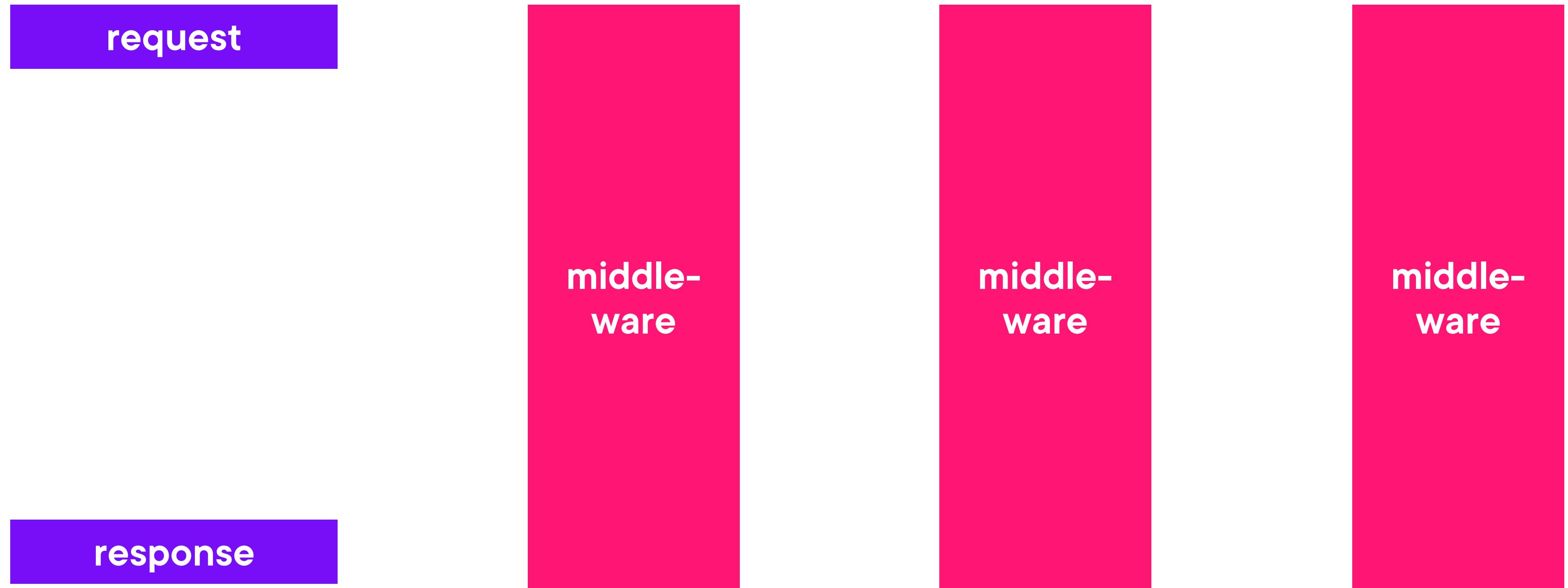
Demo



**Inspecting the ASP.NET Core API project
code structure**



The ASP.NET Core Request Pipeline & Middleware

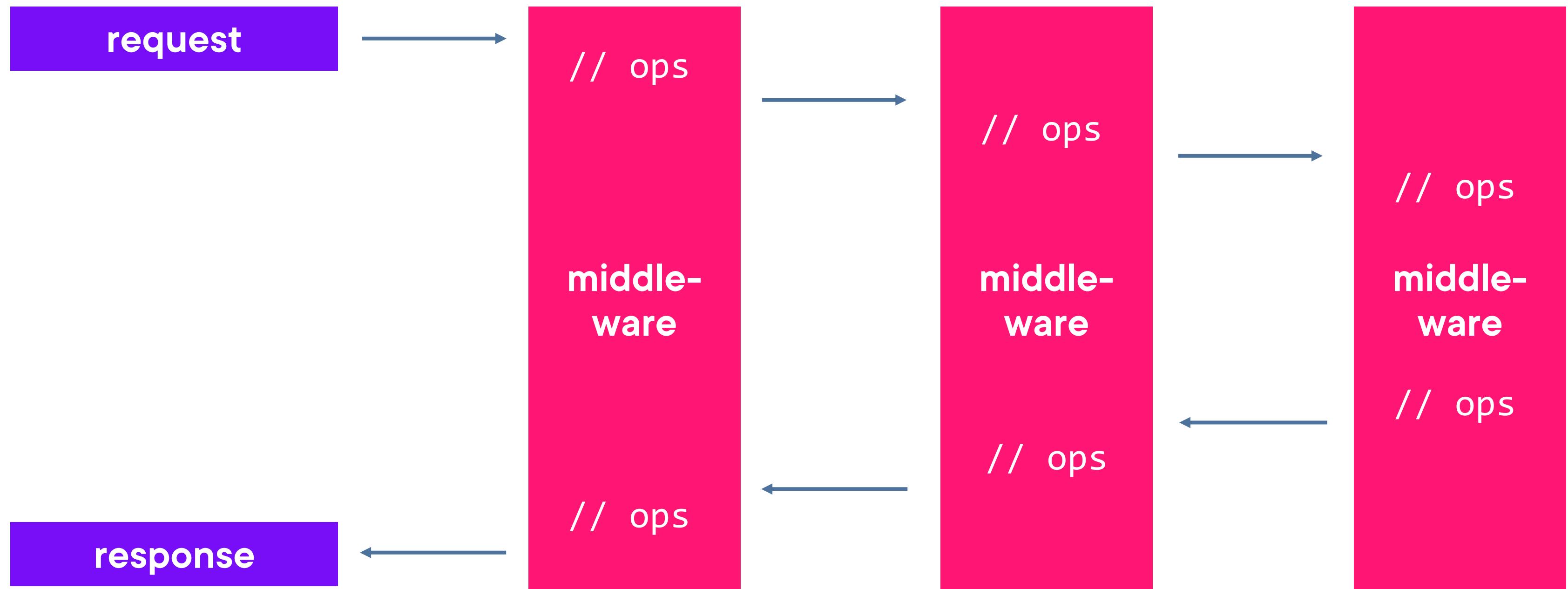


Middleware

Software components that are assembled into an application pipeline to handle requests and responses



The ASP.NET Core Request Pipeline & Middleware



The ASP.NET Core Request Pipeline & Middleware



Demo



Configuring the ASP.NET Core request pipeline



Demo



Working with different environments



Summary



ASP.NET Core

- Cross-platform
- High-performance
- Open-source
- Modern, cloud-enabled, Internet-connected apps

Current version: ASP.NET Core 8, on .NET 8



Summary



Program class is the starting point of the application

Generated Main method is responsible for configuring and running the application

**Configure services by adding them to the services collection on the builder
(WebApplicationBuilder)**



Summary



After building the web application via the WebApplicationBuilder, configure the request pipeline by adding middleware to it

- Potentially, scope this middleware to different environments



Up Next:

Creating the API and Returning Resources

