Introduction to ASP.NET Core

Shailendra Chauhan

Microsoft MVP, Technical Consultant & Corporate Trainer

DotNetTricks

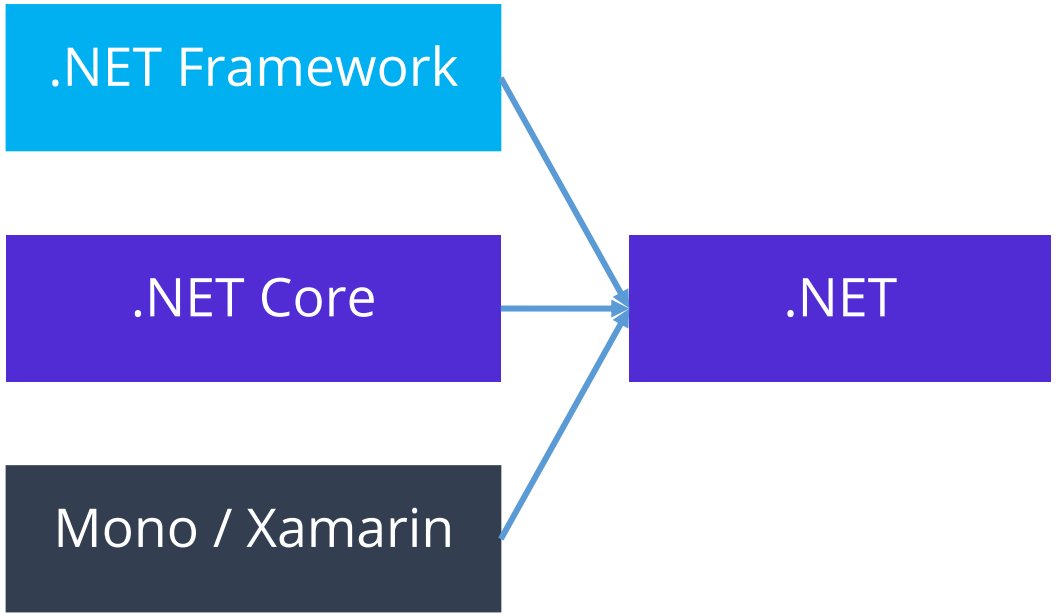
Agenda

- Evolution of ASP.NET Core
- What is ASP.NET Core?
- ASP.NET Core 5 Architecture
- ASP.NET Core Performance Benchmarks
- ASP.NET Core Application Anatomy
- ASP.NET and ASP.NET Core Request Processing

DotNetTricks

.NET 5: One .NET vision

- Single SDK, one BCL, unified toolchain
- Cross-platform native UI
- Cross-platform web UI
- Cloud native investments
- Continue improvements in speed, size, diagnostics, Azure services



DotNetTricks

Evolution of ASP.NET Core

Active Server Pages (Classic ASP)

ASP.NET Web Forms 1/2/3.5/4 /4.5/4.6/4.7

ASP.NET MVC 1/2/3/4/5

ASP.NET Web Pages 1/2/3

ASP.NET Web API 1/2

ASP.NET ~~MVC 6~~ Core MVC 1/1.1/2/2.1/ 2.2/3.0/3.1/ 5.0

Unified MVC, Web API and Web Pages

DotNetTricks

What is ASP.NET Core?

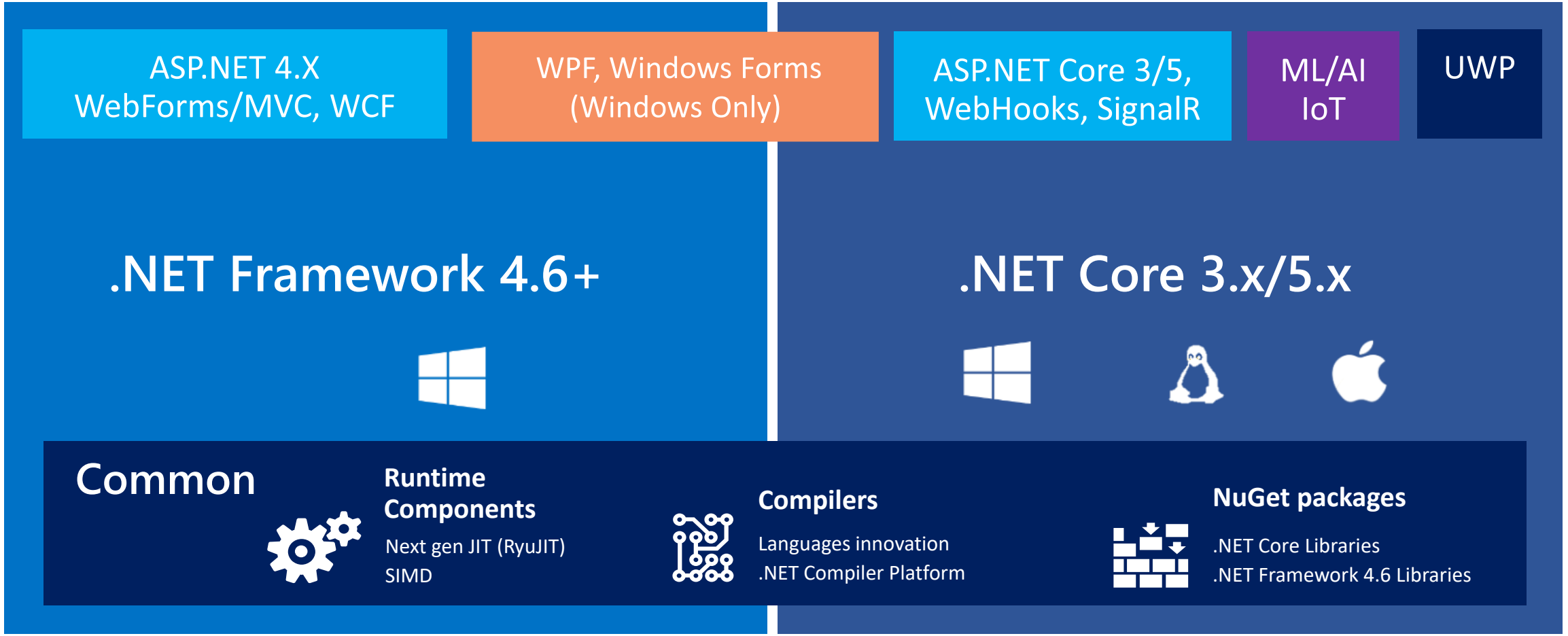
- An open-source and cross-platform framework
- Runs on .NET Core or on the full .NET Framework till ASP.NET Core 2.2
- A unified framework for building web UI and web APIs
- Built-in dependency injection
- New light-weight and modular HTTP request pipeline
- Ships entirely as NuGet packages
- Built-In support for SPA with client-side frameworks like Angular, React, Vue

DotNetTricks

What is ASP.NET Core?

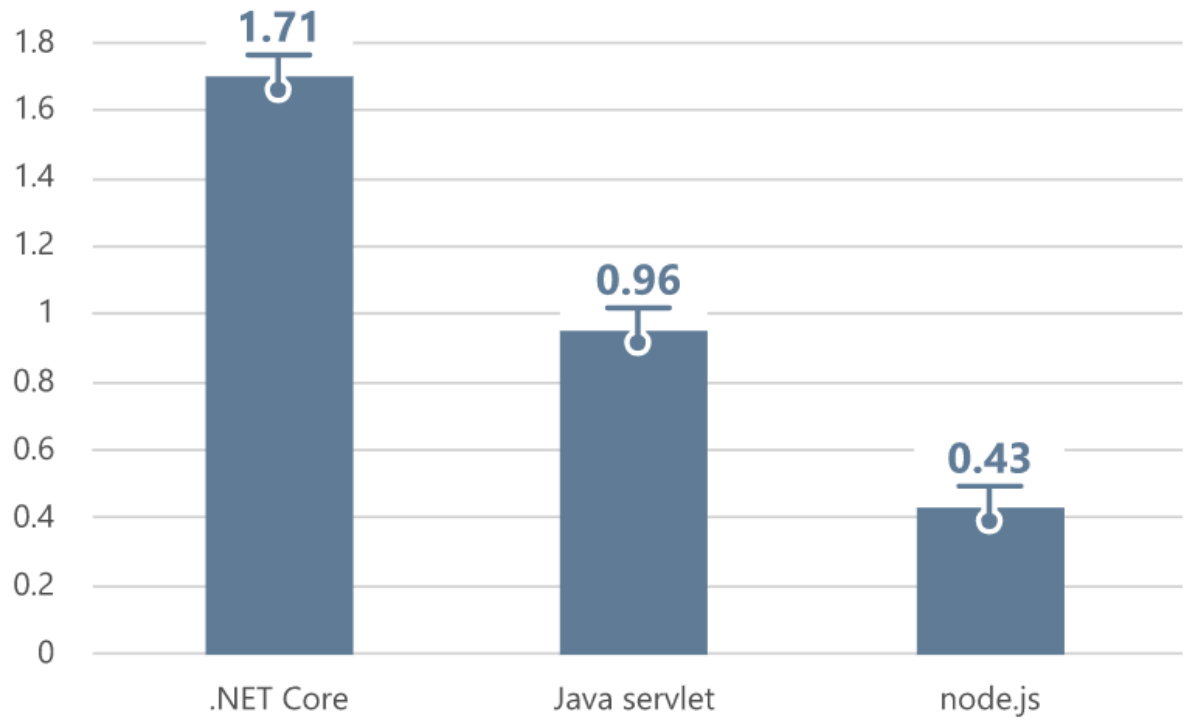
- Build modern cloud based applications, such as :
  - REST APIs or gRPC services
  - Single Page Applications with Blazor
  - Server Rendered web applications with Razor Pages
  - Real time web applications with SignalR
- Serve **2300%** more requests per second as compared to **ASP.NET 4.6**

ASP.NET Core 3.x/5.x Architecture



DotNetTricks

ASP.NET Core Performance Benchmarks



Request / second (millions) based on official tests available at TechEmpower Round 14.

Source : https://goo.gl/pgjPfm

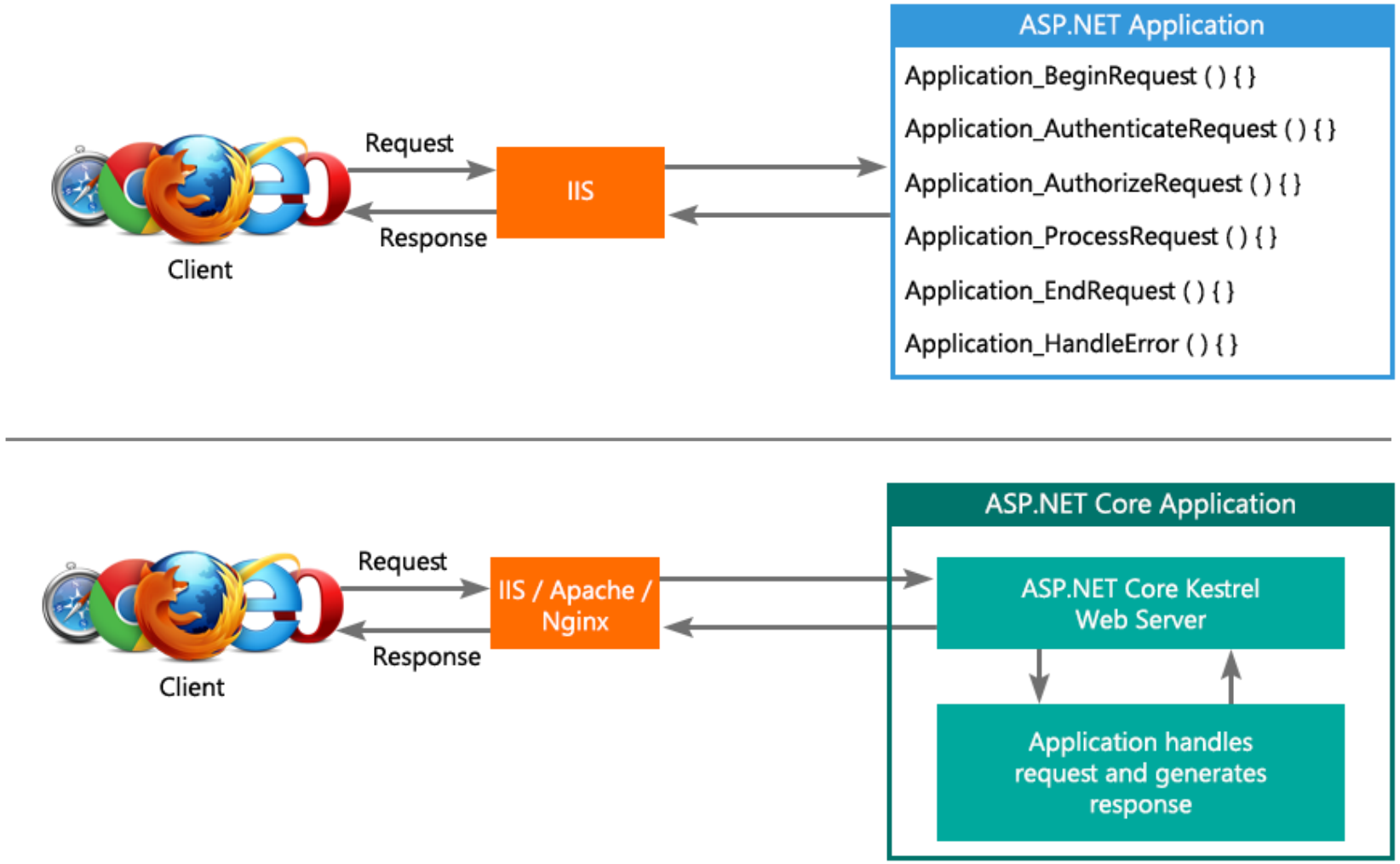
DotNetTricks

ASP.NET Core Application Anatomy

- Everything starts from Program.cs, Main Method
- ASP.NET Core apps require a Startup class
- No more Global.asax
- No more Web.Config requirement
- No more modules and handlers
- Cross-platform Kestrel web server

DotNetTricks

ASP.NET and ASP.NET Core Request Processing



DotNetTricks

Request Processing

Server

- Accepts the incoming request
- Produce an **IFeatureCollection** for the request execution

Request Processing

- Creates a **HttpContext** from the **IFeatureCollection**
- Calls into the middleware pipeline

Route Matcher

- Matches the incoming request against existing **endpoints**

Optional Middleware

- Middleware run here can observe the selected **endpoint** and make decisions (e.g. auth and CORS)

Endpoint Execution

- Execute the selected **endpoint**
- This may be a **Controller, gRPC service, SignalR Hub etc**

DotNetTricks