ASP.NET Core

Day 1

Introduction to ASP.NET Core

Session Overview

- About .Net Core
- Asp.Net Core & MVC 5
- Visual Studio Project Templates

Part 1: About .Net Core

- Introduction to Microsoft Web Technologies
- Overview of ASP.NET
- Client-Side Web Technologies
- Hosting Technologies
- Introduction to .Net Core
- .Net Core Features
- .Net Core Framework Architectures
- .Net Core 5.0 vs .Net Core 4.5
- .Net Core Supports
- Advantages of .Net Core

Introduction to Microsoft Web Technologies

Develop

- Visual Studio
- Visual Studio Code

Host

- IIS
- Microsoft Azure

Execute

Server-Side

- ASP.NET Core
- ASP.NET 4.x

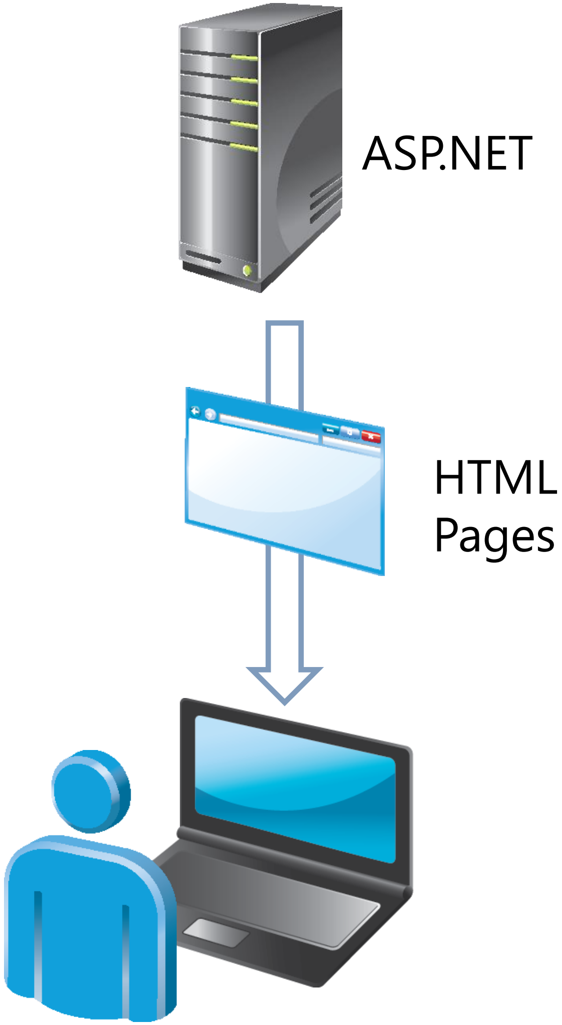
Client-Side

- JavaScript
- jQuery
- Angular
- React
- AJAX

Overview of ASP.NET

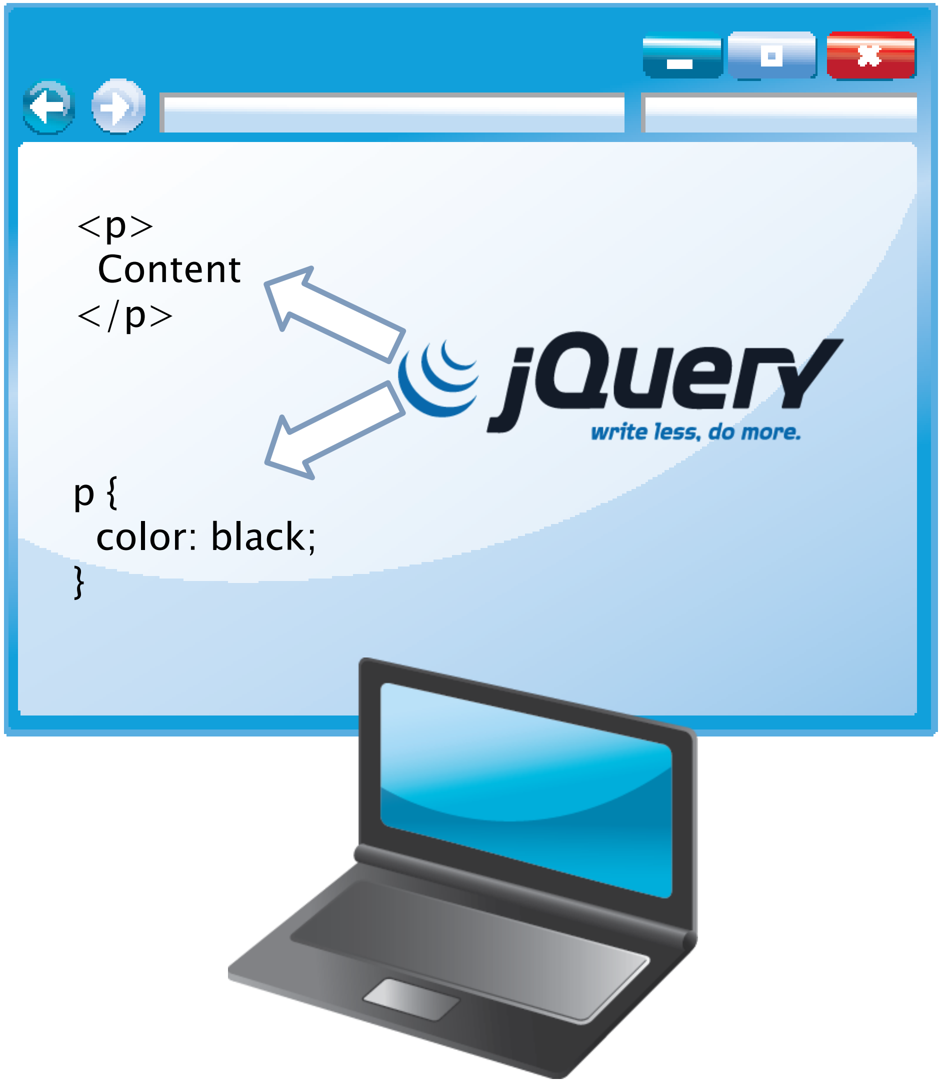
Programming Models:

- ASP.NET 4.x:
  - Web Pages
  - Web Forms
- ASP.NET 4.x and ASP.NET Core:
  - MVC
  - Web API
- ASP.NET Core:
  - Razor Pages



Client-Side Web Technologies

- JavaScript
- jQuery
- AJAX
- Angular
- React
- And more



```
<p>
  Content
</p>

p {
  color: black;
}
```

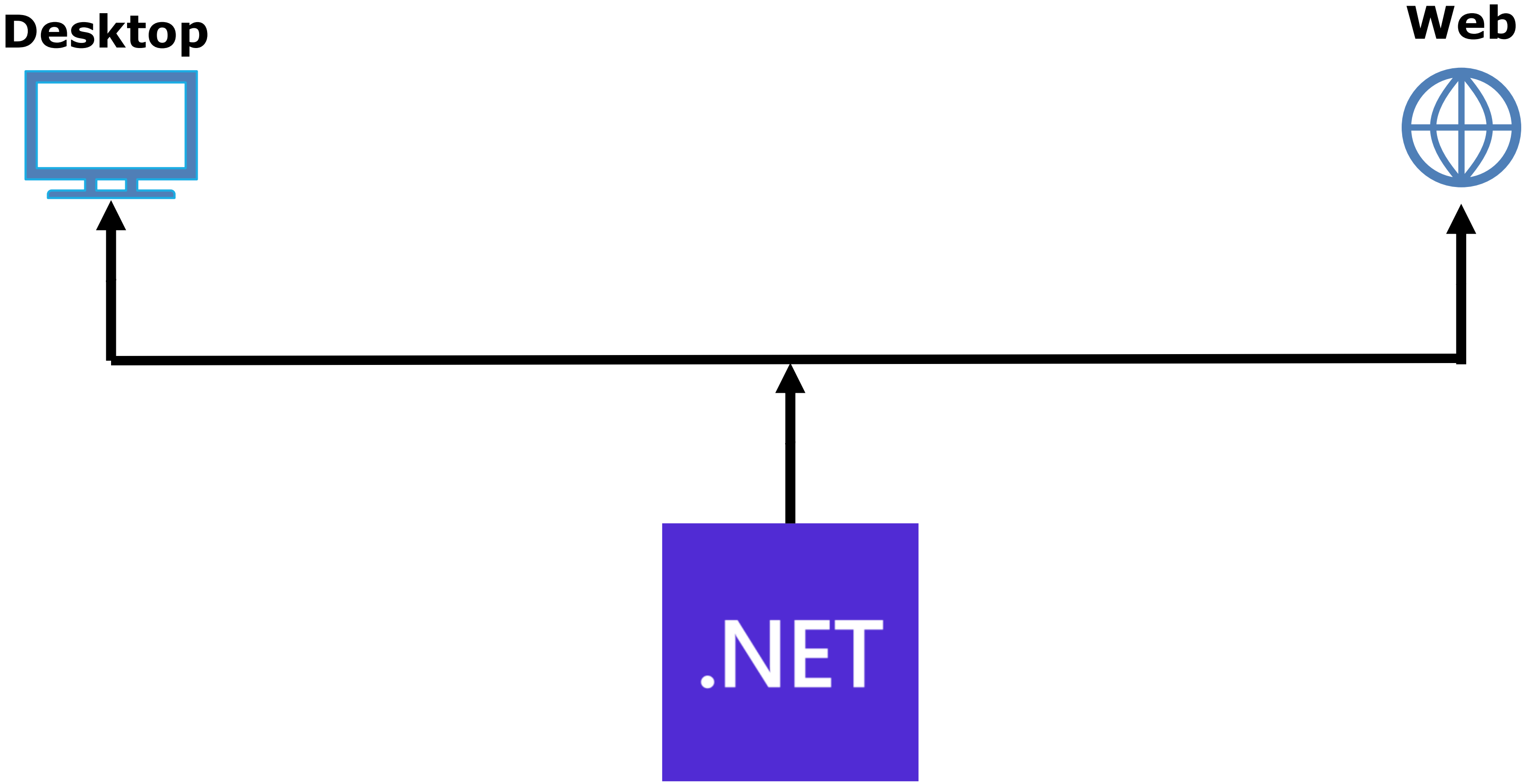
Hosting Technologies

- IIS
  - Features
  - Scaling
  - Perimeter Networks
- IIS Express
- Other Web Servers

Microsoft Azure

- Cloud computing provides scalability, flexibility, security, and reliability
- The Microsoft Azure platform includes:
  - Web Apps
  - Databases (Azure SQL Database, Cosmos DB)
  - Virtual Machines
  - Mobile Apps
  - Media Services

.NET Framework (Earlier)

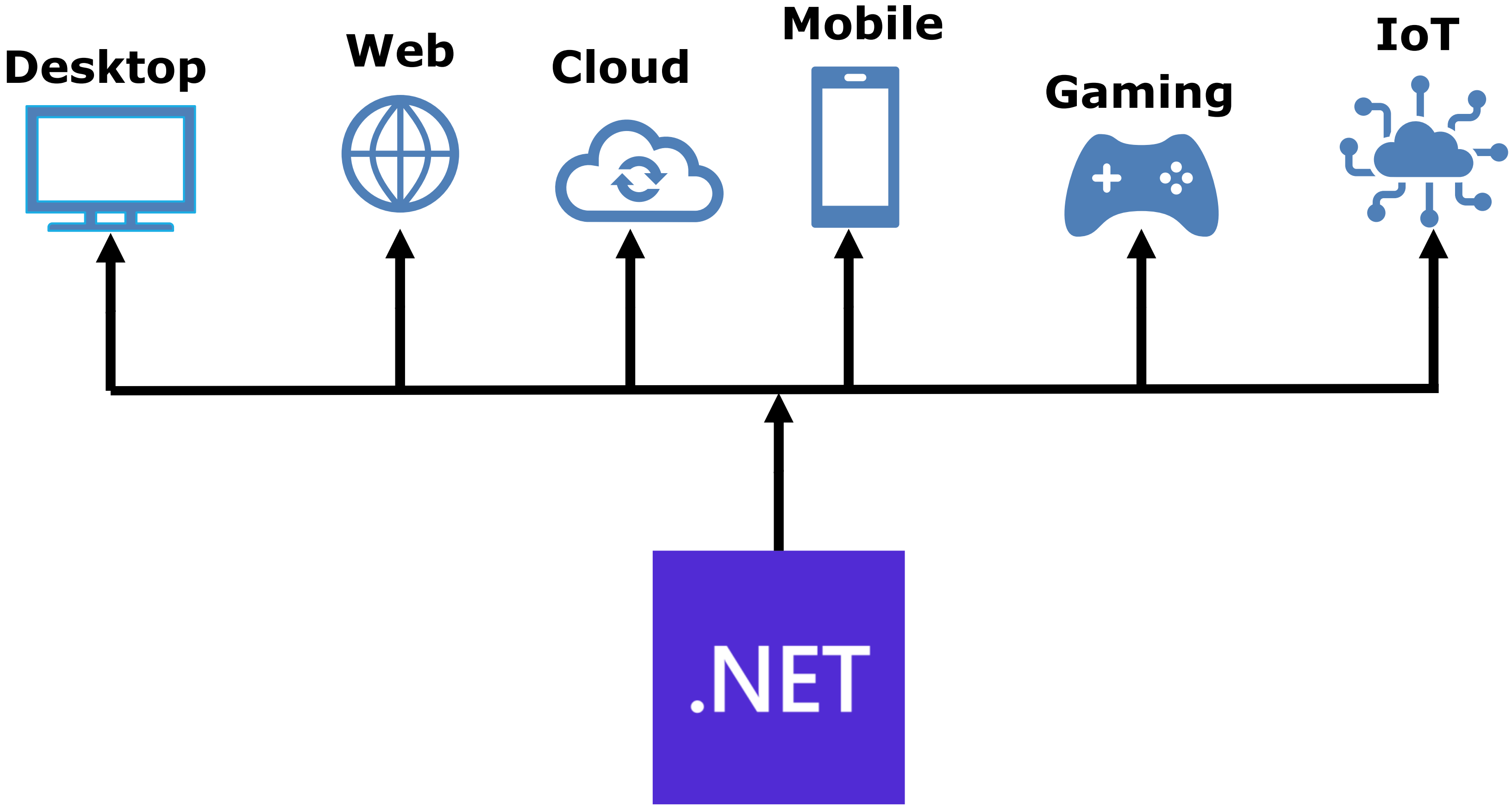


Desktop

Web

.NET

.NET Framework (Current)



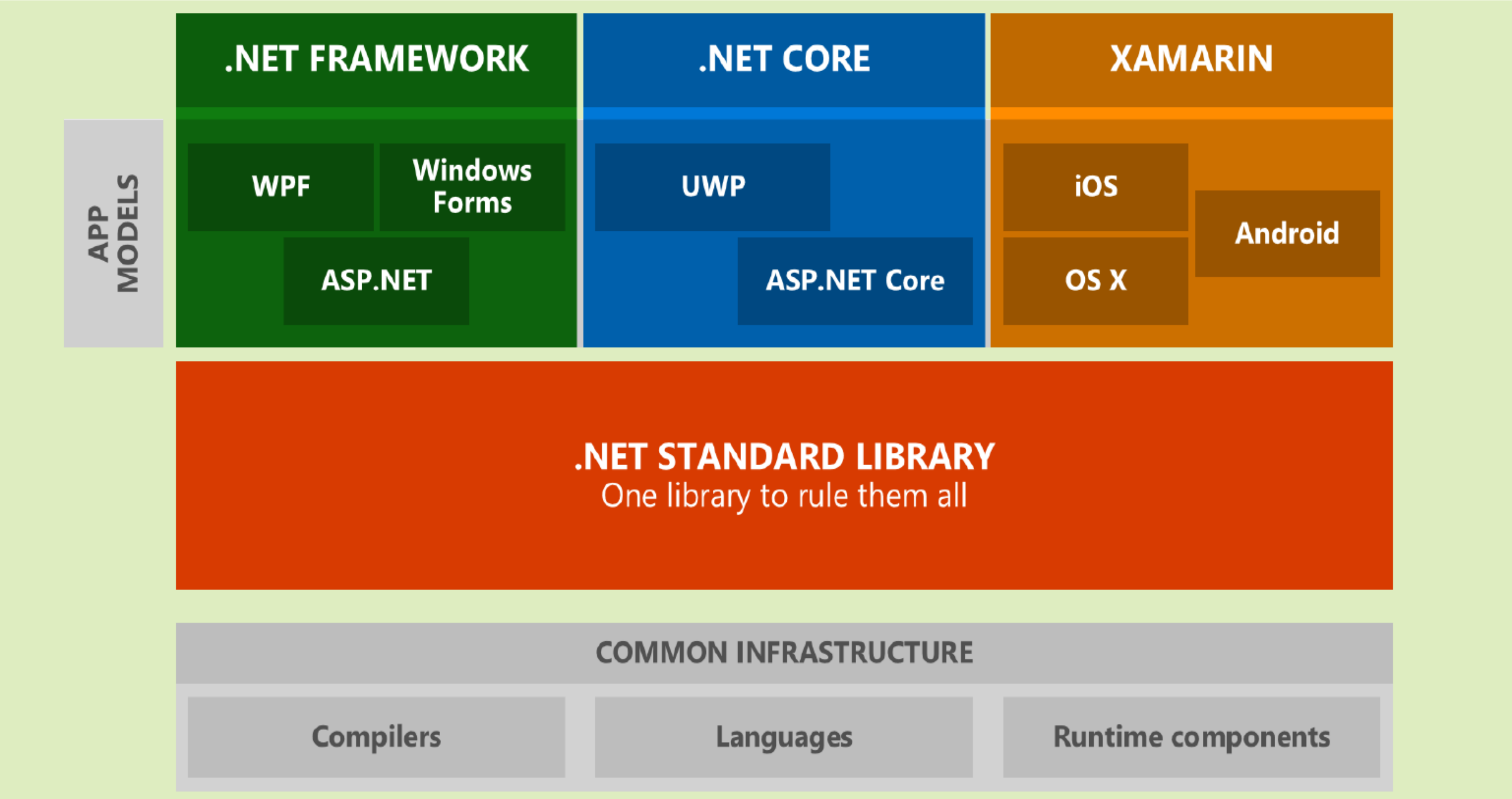
Features of .Net Core Technology

- Open Source
- Cross Platform
- Micro Service Architecture
- Containers
- Modular Design
- Asp.Net Core MVC
- In Build Dependency Injection
- Ships Entirely in NuGet Packages
- Support Client Side Framework like Angular,React, Backbone etc.

.NET Framework vs .NET Core

.NET Framework	.NET Core
Developer Already Familiar with It.	It require some training or searching or development
Support Windows Only.	**Support Windows, MacOS, Linux**
Deeply ties with IIS Server	**Can be host in Non IIS Server also**
WPF, Windows Forms, Asp.Net (Web Form, MVC, Pages)	**UWP, ASP.NET Core, Razor Pages, CLI**
Speed is an important issue	**.NET Core is much more Faster**
Can be Run app in Visual Studio Only	**Can be run App using Docker Container, Visual Code etc.**
Need current code along with 3rd Party Libraries	**No need of SignalR, WCF Client Library, Workflow**

.NET Framework vs .NET Core



Understanding .Net Core Technology

- An open-source and Cross Platform Framework
- Runs on .NET Core or full .NET Framework
- An Unified Framework for building Web UIs and Web APIs
- New Light Weight and modular HTTP Request Pipeline
- Build modern cloud based applications
    - Web Apps or
    - Mobile Backends / IOT Apps
    - Machine Learning and AI
- Serve 2300% request/Seconds as Compare Asp.Net 5.0

Choose between .NET Core and .NET Framework

- You should use .NET Core when:
  - You want your code to run cross-platform
  - You want to create microservices
  - You want to use Docker containers
  - You want to achieve a high-preforming scalable system
- You should use .NET Framework when:
  - You want to extend an existing application that uses .NET Framework
  - You want to use NuGet packages or third-party .NET libraries that are not supported in .NET Core
  - You want to use .NET technologies that aren't supported in .NET Core
  - You want to use a platform that doesn’t support .NET Core

Part 2: Asp.Net Core & MVC

- Introduction to Asp.Net Core
- Compare Asp.Net Core 3.1 and Asp.Net Core 5.0
- Asp.Net Core 3.1 vs Asp.Net MVC5
- Advantages of Asp.Net Core

Introduction to ASP.NET Core

- ASP.NET Core is:
  - Modern redesign of ASP.NET 4.x
  - Cross-platform and open source
  - Lean, high-performance and modular framework
  - Cloud-ready
- Supports the following programming models:
  - MVC
  - Razor pages
  - Web API

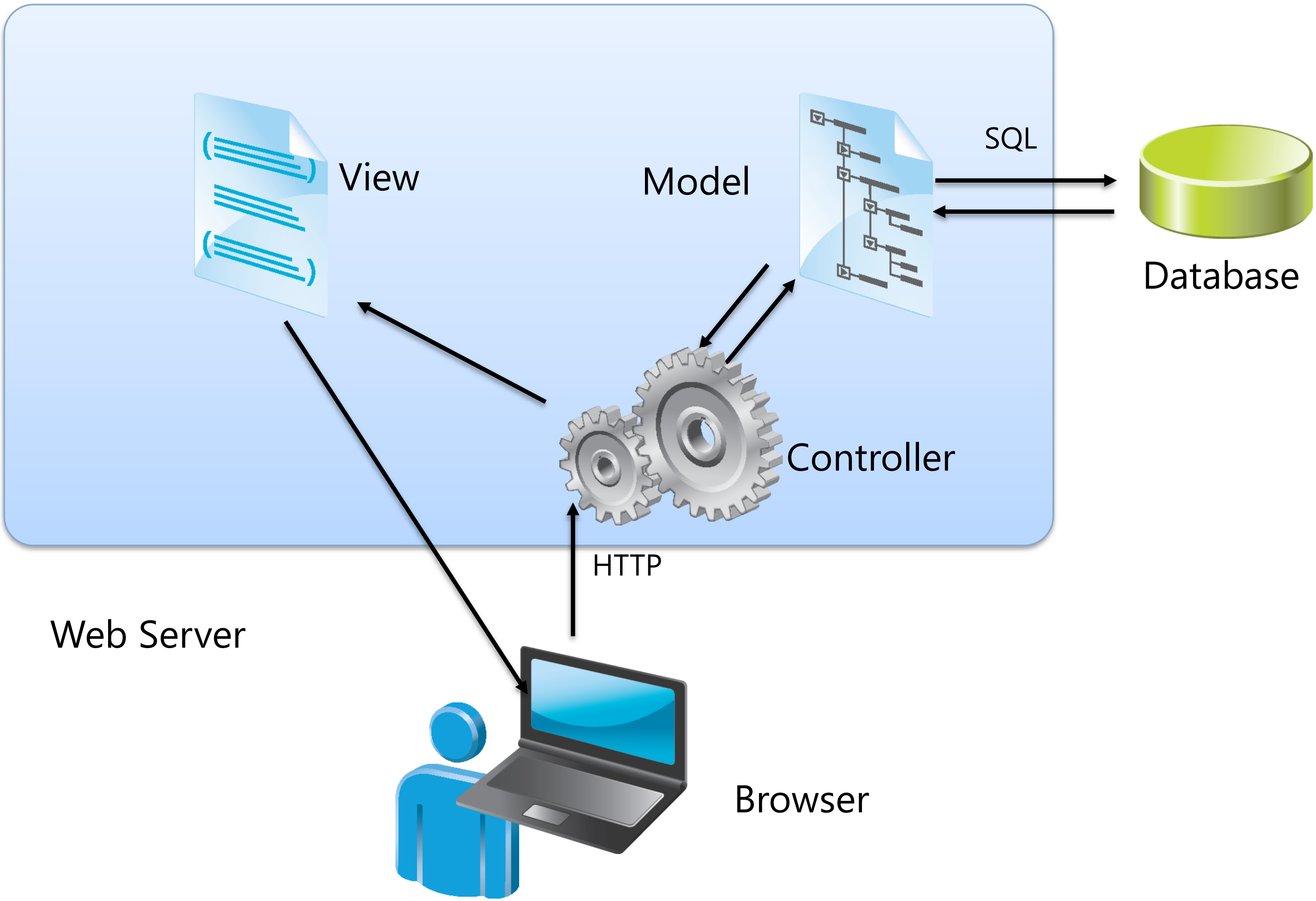
Razor Pages

- Alternative to the MVC programming model
- Starts with the **@page** directive

```
 @page
@model HomePageModel

<h1>@Model.Title</h1>
<h2>@Model.Description</li>
```

Models, Views, and Controllers



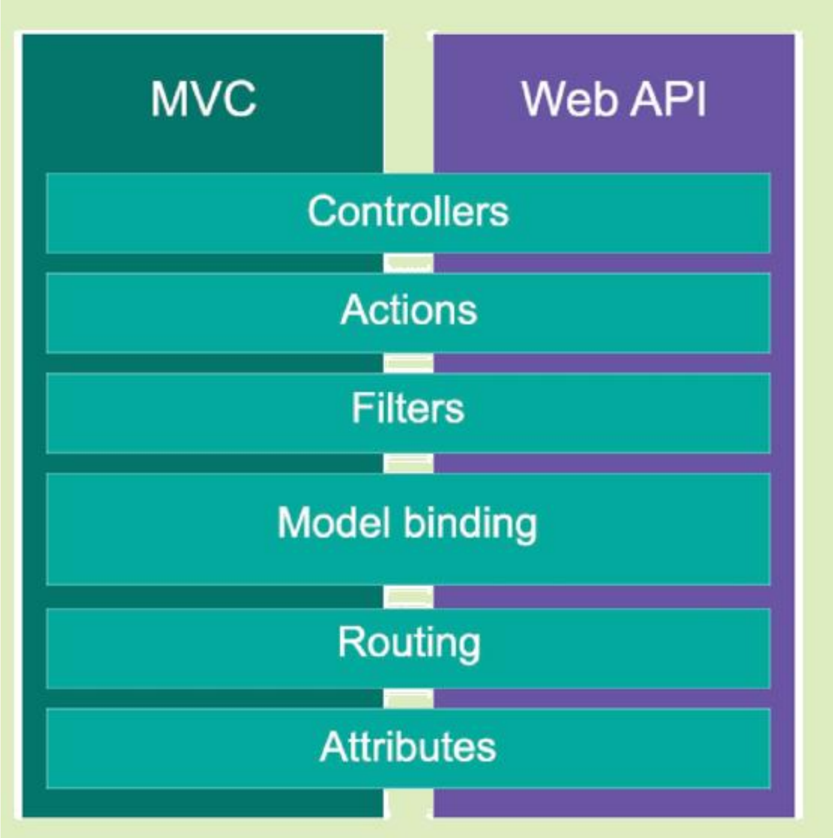
About Asp.Net Core

- ❑ ASP.NET Core is no longer support System.Web.dll
- ❑ It is based on set of granular and web factored NuGet Packages
- ❑ Benefits of Smaller Apps
  - ❑ Tighter Security
  - ❑ Reduced Servicing
  - ❑ Improved Performance
  - ❑ Decreased costs in Pay-for-what-you-use model

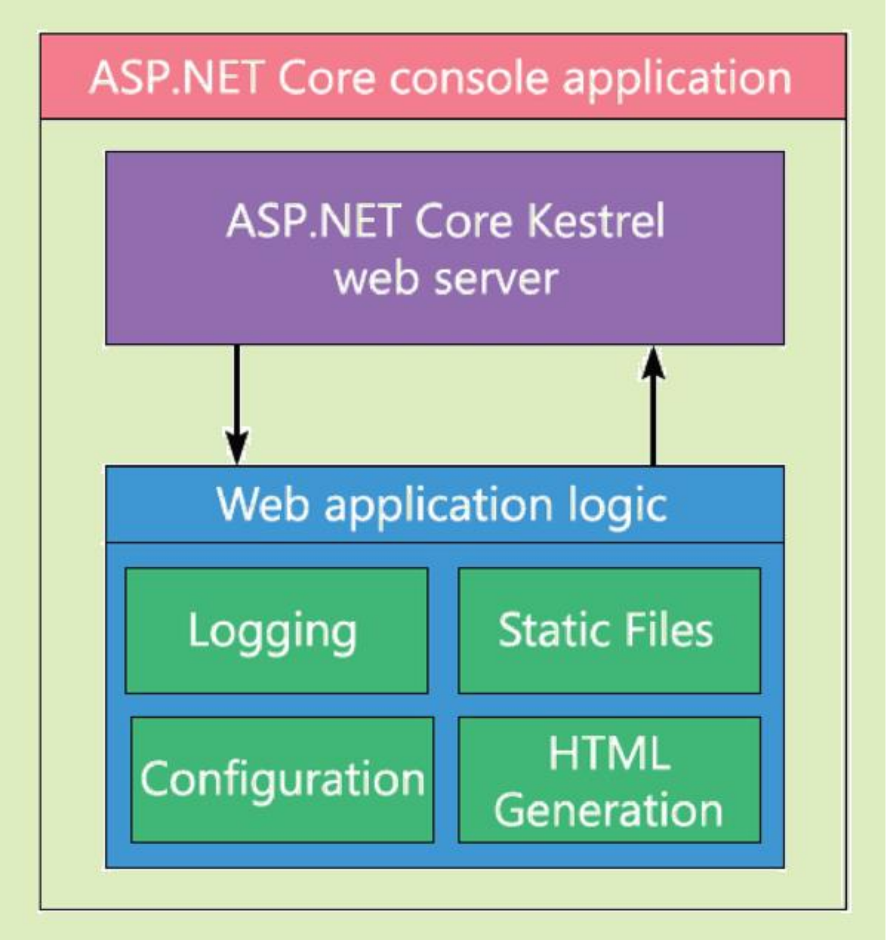
About Asp.Net Core Anatomy

- ❑ Everything starts from Program.cs, Main Method
- ❑ ASP.Net Core required Startup.cs Class
- ❑ No use of Global.asax
- ❑ No more requirement of web.config
- ❑ No More modules and handlers
- ❑ Cross Platform Kestral Web Server

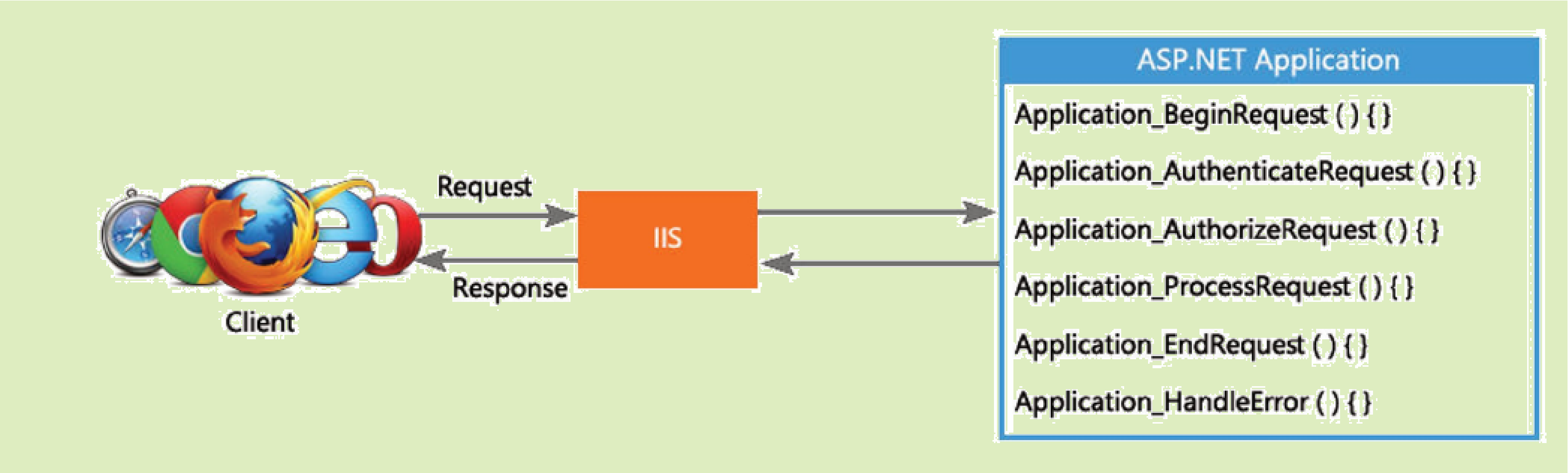
About Asp.Net Core Anatomy

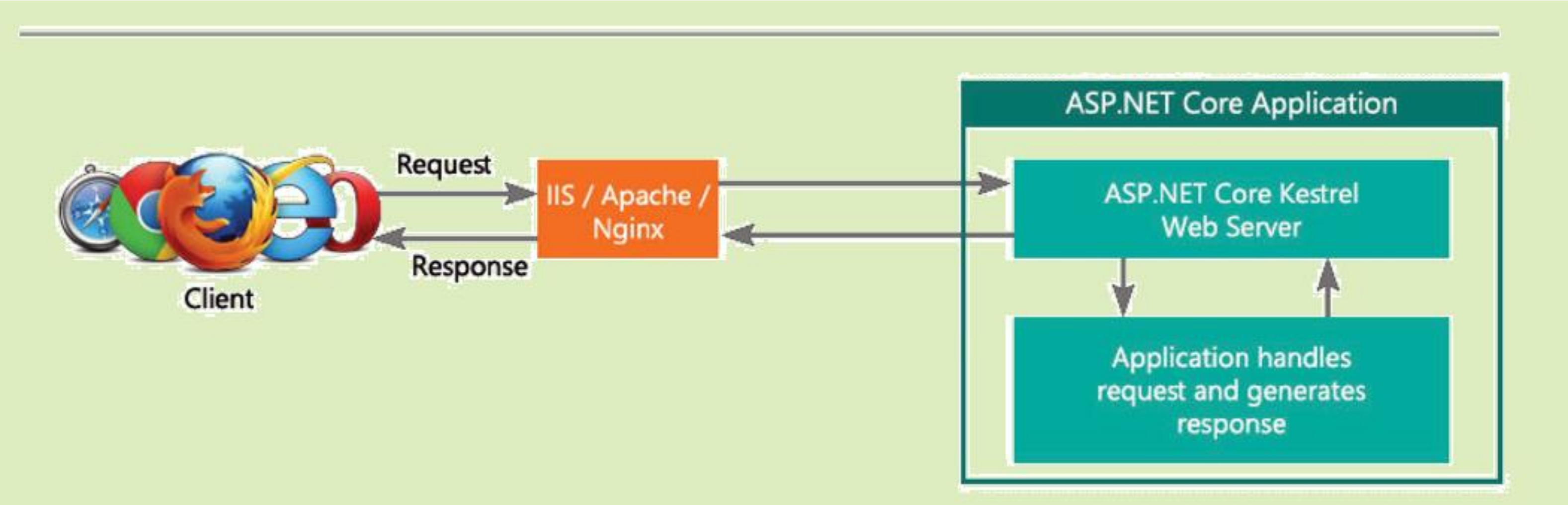


Asp.Net Core Application Model



Asp.Net & Asp.Net Core Request Processing


Client
Request
Response
IIS
ASP.NET Application
Application_BeginRequest ( ) { }
Application_AuthenticateRequest ( ) { }
Application_AuthorizeRequest ( ) { }
Application_ProcessRequest ( ) { }
Application_EndRequest ( ) { }
Application_HandleError ( ) { }


Client
Request
Response
IIS / Apache / Nginx
ASP.NET Core Application
ASP.NET Core Kestrel Web Server
Application handles request and generates response

Asp.Net Core 5.0 New Features

- ❑ Cross Platform Functionality
- ❑ Enhanced Performance
- ❑ **Unification of Development Model.** The MVC and Web API development models are unified and use the same base class Controller.
- ❑ New Health Checks API
- ❑ Integrated Dependency Injection
- ❑ Self Hosted Web Apps
- ❑ Swagger Open APIs
- ❑ Easy Compilation

Part 3: Visual Studio Project Templates

- Understanding Visual Studio Asp.Net Core Templates
- Creating an Asp.Net Core Project
- Understanding Asp.Net Core Project Folder Structure
- Understanding Configuration Files

Visual Studio Project Templates

- Console Application
- Class Library
- Unit Test Project
- xUnit Test Project
- ASP.NET Core Empty
- ASP.NET Core Web App MVC
- ASP.NET Core Web App Razor Pages