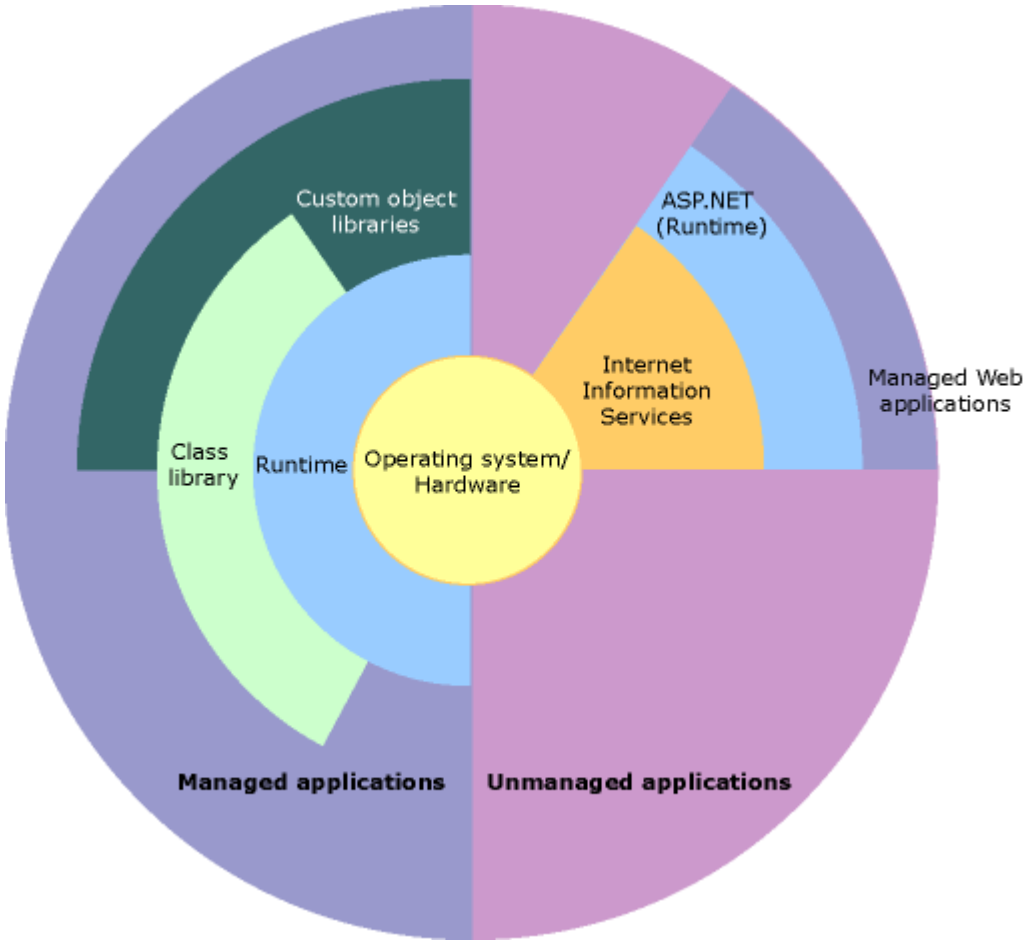
ASP.NET Core MVC

- Steve Smith. Architecting Modern Web Applications with ASP.NET Core and Azure. EDITION v5.0 - Updated to ASP.NET Core 5.0 -- Redmond, Washington, 98052-6399 – 2021, Microsoft Corporation.
- Adam Freeman. Pro ASP.NET Core MVC 2. -- Seventh Edition -- 2017, London, UK. -- 1017 pp.
- https://docs.microsoft.com/en-us/dotnet/architecture/modern-web-apps-azure/test-asp-net-core-mvc-apps

.NET


Custom object libraries
Class library
Runtime
Operating system/ Hardware
ASP.NET (Runtime)
Internet Information Services
Managed Web applications
Managed applications
Unmanaged applications

ASP.NET

ASP.NET – unified Web development **model** that includes the services necessary to build **enterprise-class Web applications** with a minimum of coding:

- server-side web application framework
- designed to produce dynamic web pages for dynamic web sites, web applications and web services

.NET and ASP.NET Core History

- **.NET Framework 1.0** (first beta)-- late 2001
  - Microsoft, Hewlett-Packard and Intel standardized Common Language Infrastructure (CLI) and C#

.....

- **.NET Core 1.0** -- June 2016
  - Visual Studio 2015 Update 3
  - **13k** APIs in .NET Standard 1.6
  - high performance and scalability
- **.NET Core 2.0** -- August 2017 (++ **2.1, 2.2** in 2018)
  - Visual Studio 2017 15.3
  - **ASP.NET Core 2.0**
  - **EF Core 2.0**
  - **32k** APIs in .NET Standard 2.0
  - new configuration system and many new other features added to make building web apps easier
  - performance have been more improved
- **.NET Core 3.0**, 3.1 – September, December 2019
  - Visual Studio 2019, C# 8
- **.NET 5.0** – November 2020 -- the next major release of .NET Core following 3.1
  - Visual Studio 2019, **C# 9**, ASP.NET Core 5.0, 6.0
- **.NET 6.0** – November 2021 **-- C# 10**
- **.NET 7.0** – November, 2022 -- Visual Studio 2022**, C# 11**,
  - **ASP.NET Core 7.0**

Modern Web Applications

- Traditional web applications:
  - have involved little client-side behavior, but instead have relied on the server for all navigation, queries
  - each new operation made by the user would be translated into a new web request, with the result being a full page reload in the end user's browser

- ASP.NET Core MVC (also supports Razor Pages) follows traditional approach:
  - each new request corresponding to a different controller action, which in turn would work with a model and return a view
  - some individual operations on a given page might be enhanced with **AJAX** (Asynchronous JavaScript and XML) functionality
  - the overall architecture of the app used many different MVC views and URL endpoints
  - Razor Pages is a simpler way to organize MVC-style pages

- Single Page Applications (**SPAs**) involve very few dynamically generated server-side page loads (if any)
  - may be initialized within a static HTML file that loads the necessary **JavaScript** libraries to start and run the app
  - make heavy usage of **web APIs** for their data needs and can provide much richer user experiences.

- Web applications may involve a combination of traditional web application behavior (typically for content) and SPAs (for interactivity)

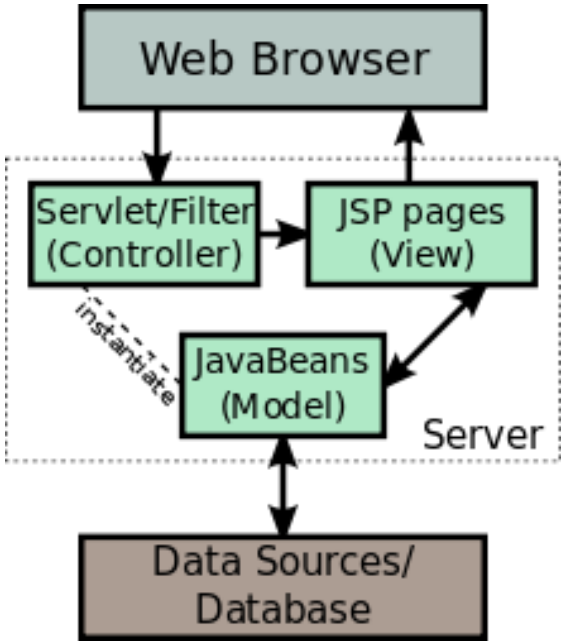
- ASP.NET Core supports both MVC (Views or Page based) and web APIs in the same application, using the same set of tools and underlying framework libraries.

ASP.NET Core MVC

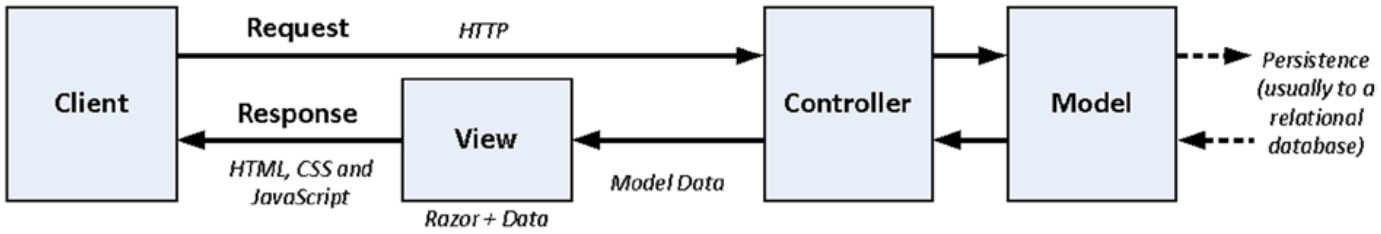
- framework designed to build the web applications according to MVC pattern
- built on the top of ASP.NET Core: ASP.NET Core MVC still has **almost all** features of ASP.NET Core
- follows the stateless pattern: there will be no illusion of state at all
- instead of using any server side controls (Grid Controls or Menu Controls) – writing and crafting the HTML completely from scratch
- Razor view engine
- testability is the one of the most important factor of using MVC -- it maintains complete separation of concerns, and hence it's easy to write and test for each module explicitly

JavaServer Pages

- **JSP** - JavaServer Pages - a technology that helps create dynamically generated web pages based on HTML, XML, or other document types.
- Released in **1999** by Sun Microsystems
- JSP is similar to PHP and ASP, but it uses the **Java**
- To deploy and run JSPs, a compatible web server with a servlet container, such as Apache Tomcat or Jetty, is required.



Model - View - Controller



Controller – application logic (functionality):

- handles browser requests
- retrieve model data
- calls View templates that return a response

Model – contains classes that represent data of the app

- performs operations with DBs
- organizes relations between data-classes
- uses validation logic to enforce business rules for that data
- typically, model objects retrieve and store model state in a database

View – works with the components that display the app's UI

- generally, this UI displays the model data

class HomeController

```
public class HomeController : Controller
{
    // GET: /Home/
    public ActionResult Index()
    {
        return View();
    }
}
```

```
public string Index()
{
    return "Hello World";
}
```

Rendering Web Pages by View

```
public class HomeController : Controller
{
          public ViewResult Index()
         {
             return View();
         }
}
```

Index.cshtml after Add View

```
@{
    Layout = null;
}
<!DOCTYPE html>
<html>
<head>
    <meta name="viewport" content="width=device-width" />
    <title>Index</title>
</head>
<body>
    <div>
    </div>
</body>     </html>
```

PartyInvites

MVC Conventions

- create controllers in Controllers folder
- naming: **`XXXXController`**
- scaffolding according to patterns
- **`HomeController`** – default starting point for MVC application
- **`Index()`** – default action method (any public)
- Routing System maps URLs to controllers and actions
  default </>, </Home>, </Home/Index>
- views are associated with action methods by naming convention
  `/Views/Home/Index.cshtml`

ASP.NET Core Facilities

- **Security Infrastructure**
  - user authentication
  - authorization
- **State Management**
  - save and manage information between page requests: application-specific, session-specific, page-specific, user-specific information
  - information can be independent of any controls on the page
- **ASP.NET Configuration System**
  - enables to define configuration settings for Web server, Web site or for individual applications at time applications are deployed
  - editing configuration settings at any time
  - configuration settings are stored in XML-based files
- **Extensible Hosting Environment and Application Life-Cycle Management**
  - application life cycle control from a user first accesses a resource to the point the application is shut down
  - architecture of ASP.NET enables to respond to application events and create custom HTTP handlers and HTTP modules

Dynamic Output by ViewBag

- **`ViewBag.Greeting`** property doesn't exist until the moment assigning value
- using in Razor expression

```
<div>
    @ViewBag.Greeting World (from the view)
</div>
```

- in **`View()`** call (in controller's **`Index`** method) MVC framework locates the **`Index.cshtml`** view file and Razor view engine parses the file's content
- passing multiple data values from controller to the view by assigning values to more than one dynamic property

PartyInvites