



Bài 01

Giới thiệu về ASP.NET MVC 5



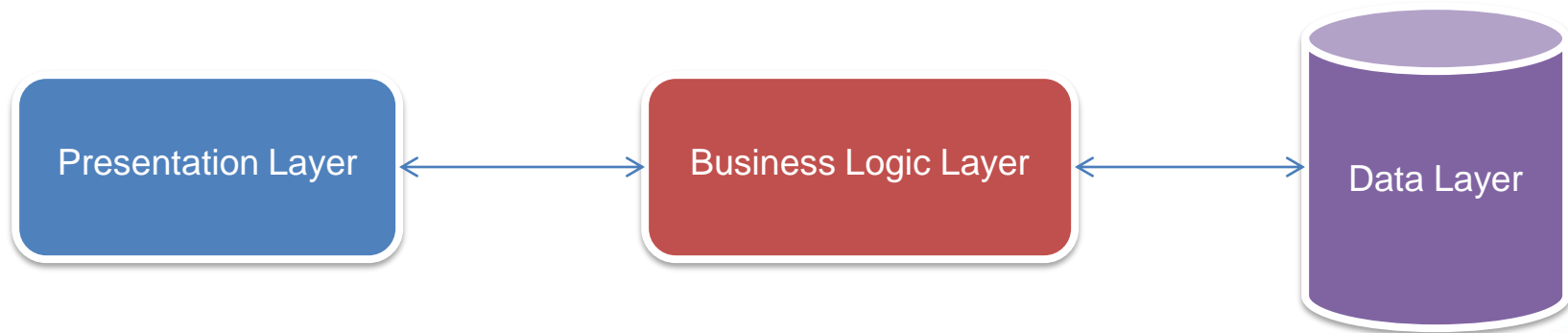
- ◆ Giới thiệu về các ứng dụng web
- ◆ Các lớp trong ứng dụng Web
- ◆ Kiến trúc ứng dụng web
- ◆ Các loại trang web
- ◆ Giới thiệu về mô hình MVC
- ◆ Giới thiệu về ASP.NET MVC
- ◆ Kiến trúc ứng dụng ASP.NET MVC 5
- ◆ Các công nghệ hỗ trợ
- ◆ Tạo ứng dụng ASP.NET MVC 5



- ◆ Các ứng dụng web là các chương trình chạy trên Web Server và truy cập thông qua trình duyệt.
- ◆ Cho phép bạn chia sẻ và truy cập thông tin qua internet toàn cầu tại bất kỳ thời điểm nào.
- ◆ Cho phép bạn thực hiện những giao dịch thương mại điện tử.



- ◆ **Presentation layer:** Cho phép người dùng tương tác với ứng dụng.
- ◆ **Business logic layer:** Cho phép điều khiển luồng thực thi và giao tiếp giữa lớp Presentation và lớp data.
- ◆ **Data layer:** Cung cấp dữ liệu được lưu trữ trong cơ sở dữ liệu cho tầng Business logic layer.



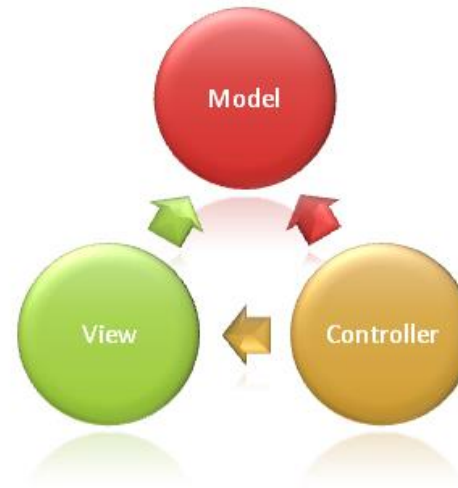
- ◆ Kiến trúc của ứng dụng phụ thuộc trên hệ thống mà các lớp của ứng dụng được phân phối và giao tiếp với nhau.
- ◆ Một ứng dụng có thể dựa trên một trong các loại kiến trúc sau:
 - ◆ **Single-tier:** 3 lớp được tích hợp cùng nhau và được cài trên 1 máy đơn.
 - ◆ **Two-tier:** 3 lớp được phân chia thành 2 tầng client và server.
 - ◆ **Three-tier:** 3 lớp được phân bổ trên các máy tính khác nhau.
 - ◆ **N-tier:** các thành phần của 3 lớp được tách ra nhiều hơn.

- ◆ **Static Web page:** bao gồm chỉ mã Hyper Text Markup Language (HTML) để hiển thị nội dung tới người dùng.
- ◆ **Dynamic Web page:** bao gồm mã HTML gộp với Server-side và client side scripts để đáp ứng các hành động của người dùng.



- ✓ Mô hình Model-View-Controller có nguồn gốc từ cộng đồng phát triển Smalltalk trong những năm 1970, và sau đó nó đã được phổ biến để sử dụng trên các trang web với sự ra đời của Ruby on Rails vào năm 2003.
- ✓ MVC Chia ứng dụng ra 3 thành phần chính:

1. Model
2. View
3. Controller



◆ Models

- ✓ Chứa data mà ứng dụng sử dụng
- ✓ Nhận yêu cầu từ Views
- ✓ Thực hiện các yêu cầu liên quan đến DB
- ✓ Trả về các giá trị tính toán cho Views
- ✓ Kiểm tra tính hợp lệ của data

◆ Views

- ✓ Hiển thị giao diện người dùng
- ✓ Được tạo ra từ dữ liệu của Models

◆ Controllers

- ✓ Xử lý các yêu cầu của user
- ✓ Chọn Views thích hợp hiển thị cho người dùng
- ✓ Đồng bộ hóa giữa khung nhìn và mô hình



ASP.NET MVC là gì?

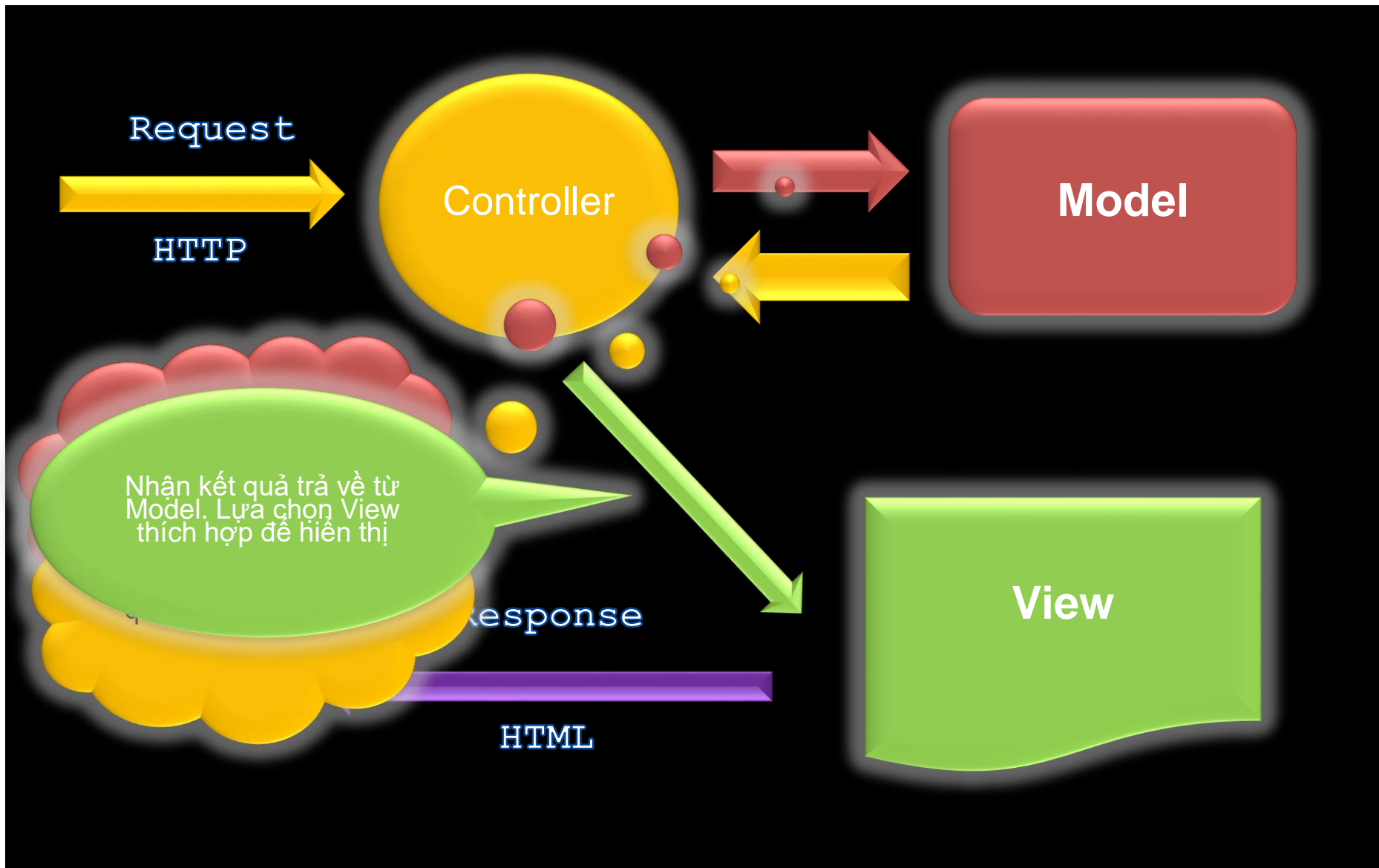
- ◆ ASP.NET MVC là một phần của Framework Appliaction ASP.NET Web.
- ◆ Nó mang đến một sự lựa chọn khác so với Web Form để xây dựng các ứng dụng Web trên nền tảng NET.
- ◆ Được Microsoft công bố lần đầu tiên vào tháng 10/2007 và kể từ đó đã có bốn phiên bản chính.

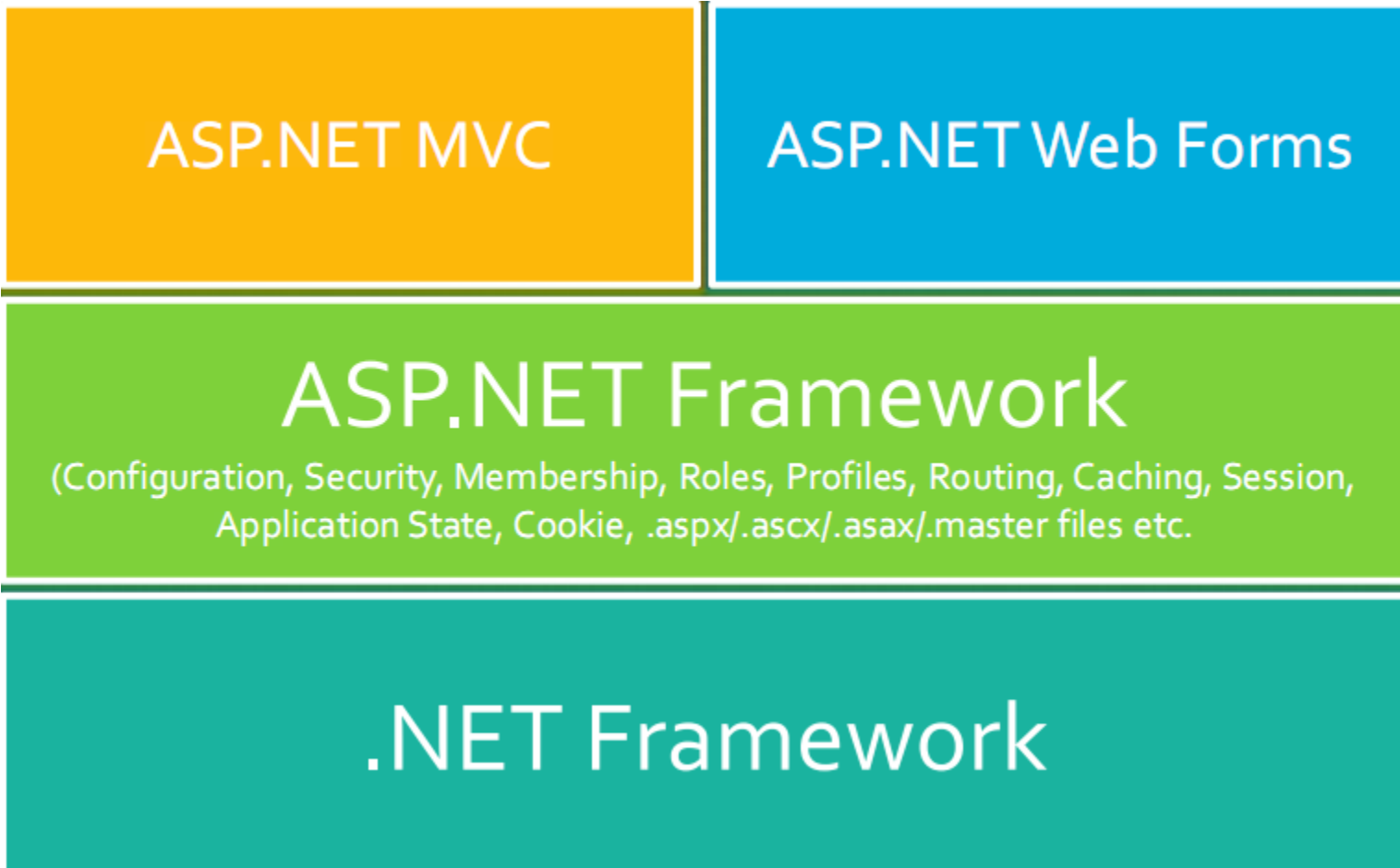


- ◆ ASP.NET **MVC 1** provides a new Model-View-Controller framework on top of the existing ASP.NET 3.5 runtime
- ◆ ASP.NET **MVC 2** provides a new Model-View-Controller framework on top of the existing ASP.NET 3.5 SP1 runtime
- ◆ ASP.NET **MVC 3** provides a Model-View-Controller framework on top of the ASP.NET 4 runtime
- ◆ ASP.NET **MVC 4** provides a Model-View-Controller framework for developing Web applications using Visual Studio 2010 SP1 or Visual Web Developer 2010 SP1
- ◆ ASP.NET **MVC 5** Release in 10/2013 on top of the existing Framework 4.5 and .Net Framework 4.5.1 distributed Visual Studio 2013



Hoạt động của ASP.NET MVC





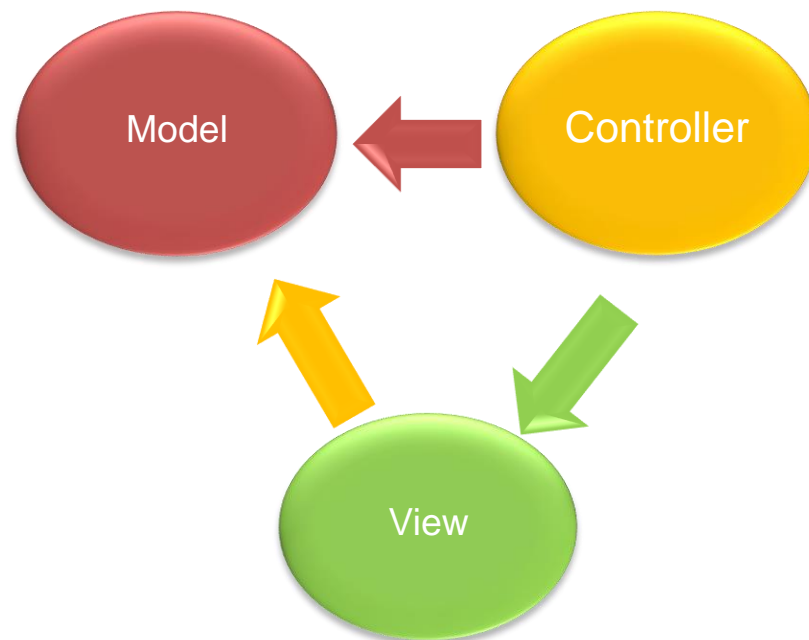
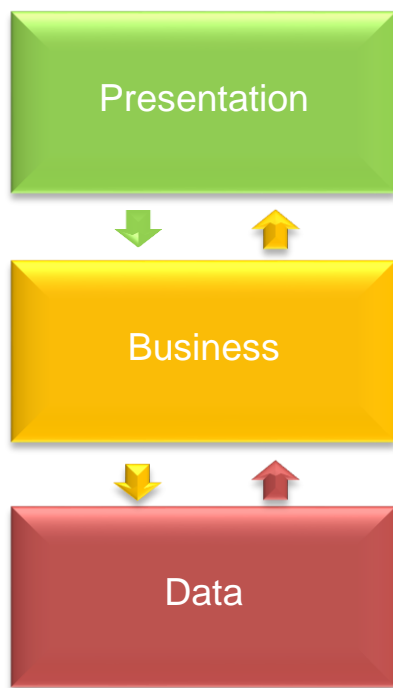
- ◆ Ưu điểm chính của **ASP.NET MVC** là:
 - ◆ Cho phép toàn quyền kiểm soát việc sinh ra mã HTML cho Client.
 - ◆ Cung cấp tách bạch những mối quan hệ.
 - ◆ Cho phép [Test Driven Development \(TDD\)](#).
 - ◆ Dễ dàng tích hợp với các JavaScript framework.
 - ◆ Theo thiết kế tự nhiên phi trạng thái của các trang web.
 - ◆ RESTful urls cho phép SEO.
 - ◆ Không có ViewState vàPostBack sự kiện.
- ◆ Ưu điểm chính của **ASP.NET Web Form** là:
 - ◆ Nó cung cấp phát triển RAD
 - ◆ Mô hình phát triển dễ dàng cho các nhà phát triển quen làm trên nền Winform.

Tham khảo



MVC và mô hình 3 Layer

- ✓ Giống nhau:
 - ✓ Cả hai đều tách rời các thành phần riêng biệt
 - ✓ Mỗi thành phần đảm nhiệm mỗi nhiệm vụ khác nhau



◆ Khác nhau

MVC

- ✓ View:
 - ✓ Chỉ hiển thị giao diện
- ✓ Controller:
 - ✓ Xử lý request, gọi thực hiện Model, View
- ✓ Model:
 - ✓ Bao gồm DTO
 - ✓ Xử lý dữ liệu
 - ✓ Kiểm tra tính hợp lệ data

3 Layers

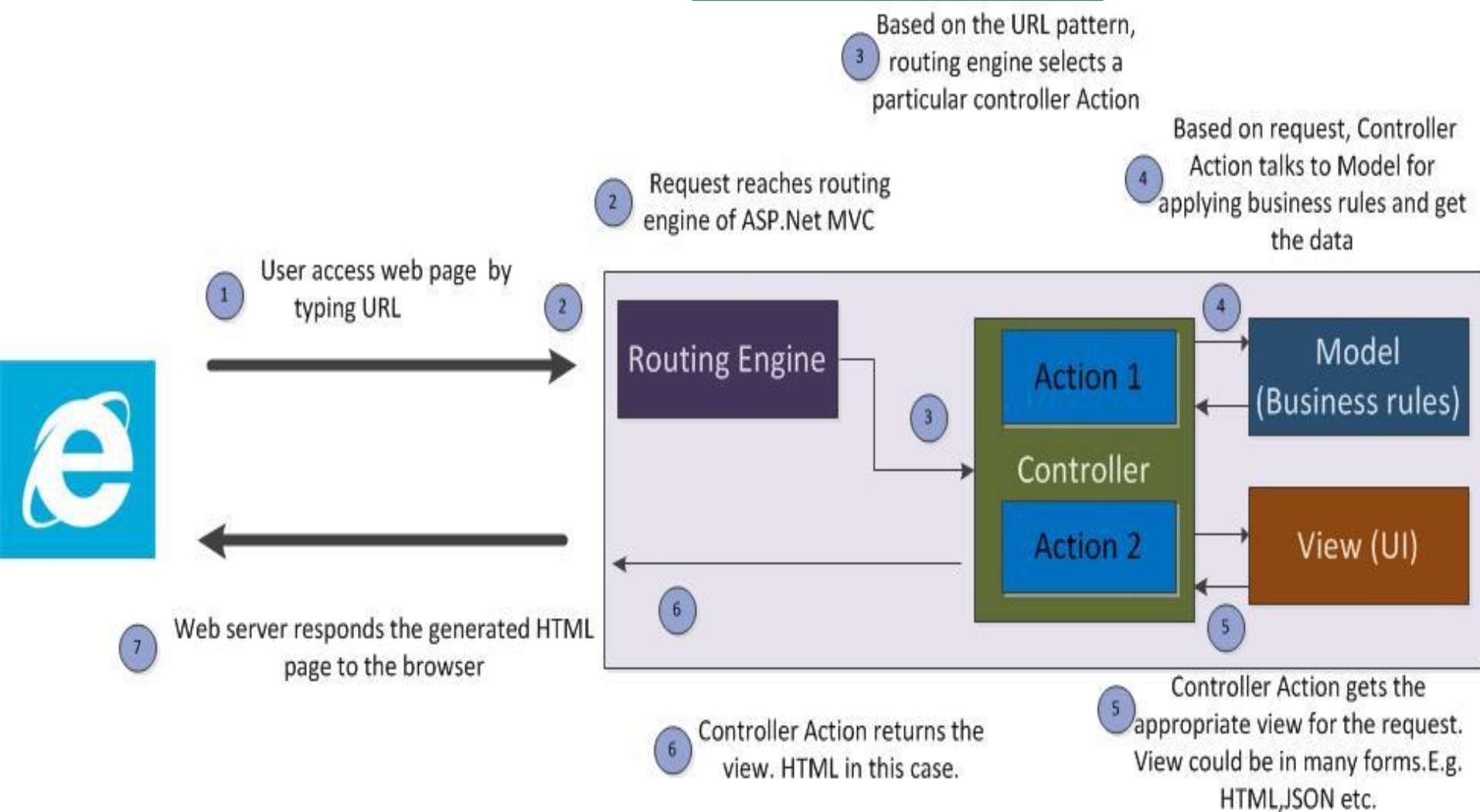
- ✓ Presentation:
 - ✓ Hiển thị giao diện
 - ✓ Xử lý sự kiện
- ✓ Business:
 - ✓ Xử lý nghiệp vụ
- ✓ Data Access:
 - ✓ Xử lý dữ liệu



- ◆ Các thành phần cơ bản của một ứng dụng ASP.NET MVC
 - ◆ MVC Framework
 - ◆ Route engine
 - ◆ Route configuration
 - ◆ Controller
 - ◆ Model
 - ◆ View engine
 - ◆ View



Kiến trúc của ứng dụng ASP.NET MVC5



Các công nghệ hỗ trợ

JavaScript

JQuery

Asynchronous JavaScript and XML (AJAX)

IIS







Windows Azure



Tạo ứng dụng ASP.NET MVC 1-2

New ASP.NET Project - Example1

Select a template:

 Empty	 Web Forms	 MVC	 Web API
 Single Page Application	 Facebook		

Add folders and core references for:

☐ Web Forms ☒ MVC ☐ Web API

☐ Add unit tests

Test project name:

A project template for creating ASP.NET MVC applications. ASP.NET MVC allows you to build applications using the Model-View-Controller architecture. ASP.NET MVC includes many features that enable fast, test-driven development for creating applications that use the latest standards.







[Learn more](#)

Authentication: **Individual User Accounts**

Tạo ứng dụng ASP.NET MVC 2-2

New ASP.NET Project - Example2

Select a template:

 Empty	 Web Forms	 MVC	 Web API
 Single Page Application	 Facebook		

Add folders and core references for:

☐ Web Forms ☒ MVC ☐ Web API

☐ Add unit tests

Test project name:

A project template for creating ASP.NET MVC applications. ASP.NET MVC allows you to build applications using the Model-View-Controller architecture. ASP.NET MVC includes many features that enable fast, test-driven development for creating applications that use the latest standards.

[Learn more](#)

Change Authentication

Authentication: **No Authentication**

Kích OK

OK Cancel

Cấu trúc của Project

- ◆ **App_Start:** chứa các class cấu hình để chạy ứng dụng
- ◆ **Content:** chứa các file css, theme, images cho ứng dụng
- ◆ **Controllers:** chứa các controller của ứng dụng, tên controller luôn kết thúc bằng từ Controller
- ◆ **Models:** chứa các models class, có thể là các entities map với các bảng trong cơ sở dữ liệu, hoặc có thể là class xử lý business logic trong ứng dụng
- ◆ **Scripts:** chứa các file javascript, jquery, ajax
- ◆ **Views:** mỗi controller trong ứng dụng phải có một thư mục tương ứng với tên controller đó nằm trong Views, bên trong thư mục sẽ có tên các View để hiển thị thông tin khi các action được gọi, thư mục **Shared** chứa các view được sử dụng ở nhiều controller khác nhau, hoặc chứa **MVC usercontrol** sử dụng ở nhiều nơi khác nhau, hoặc chứa **file master page** cho hệ thống
- ◆ **Global.asax**

