

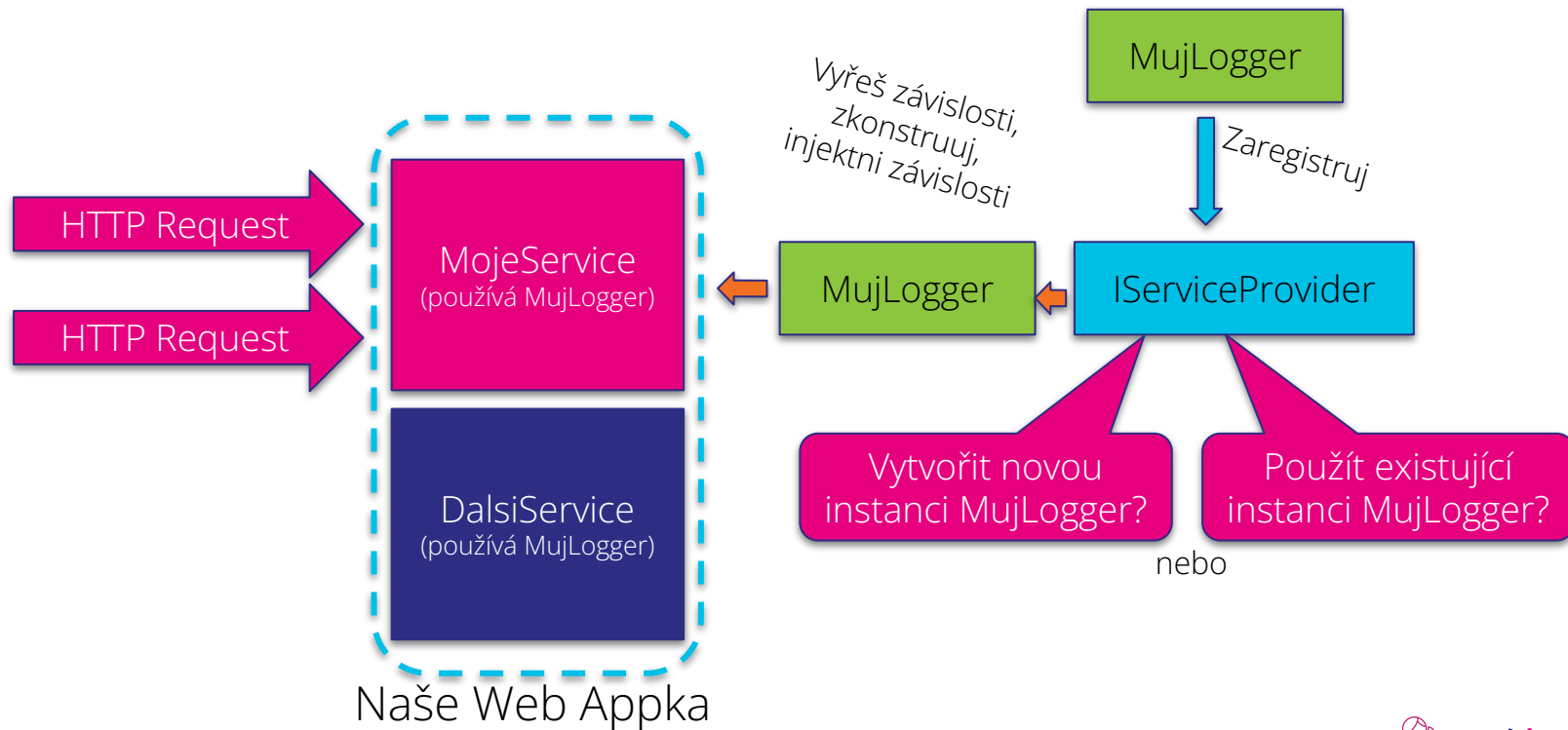
# LEKCE 08

## Blazor Introduction

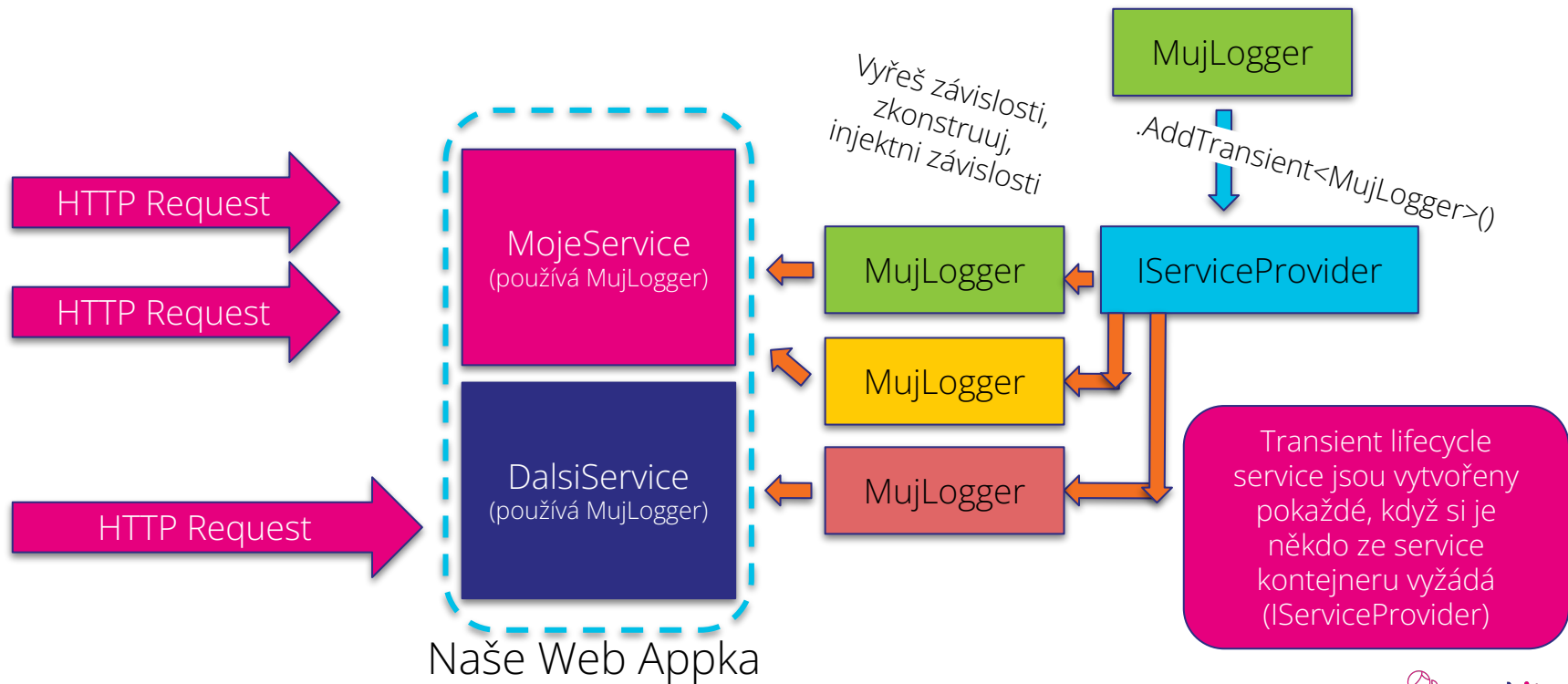


# Service Lifecycle

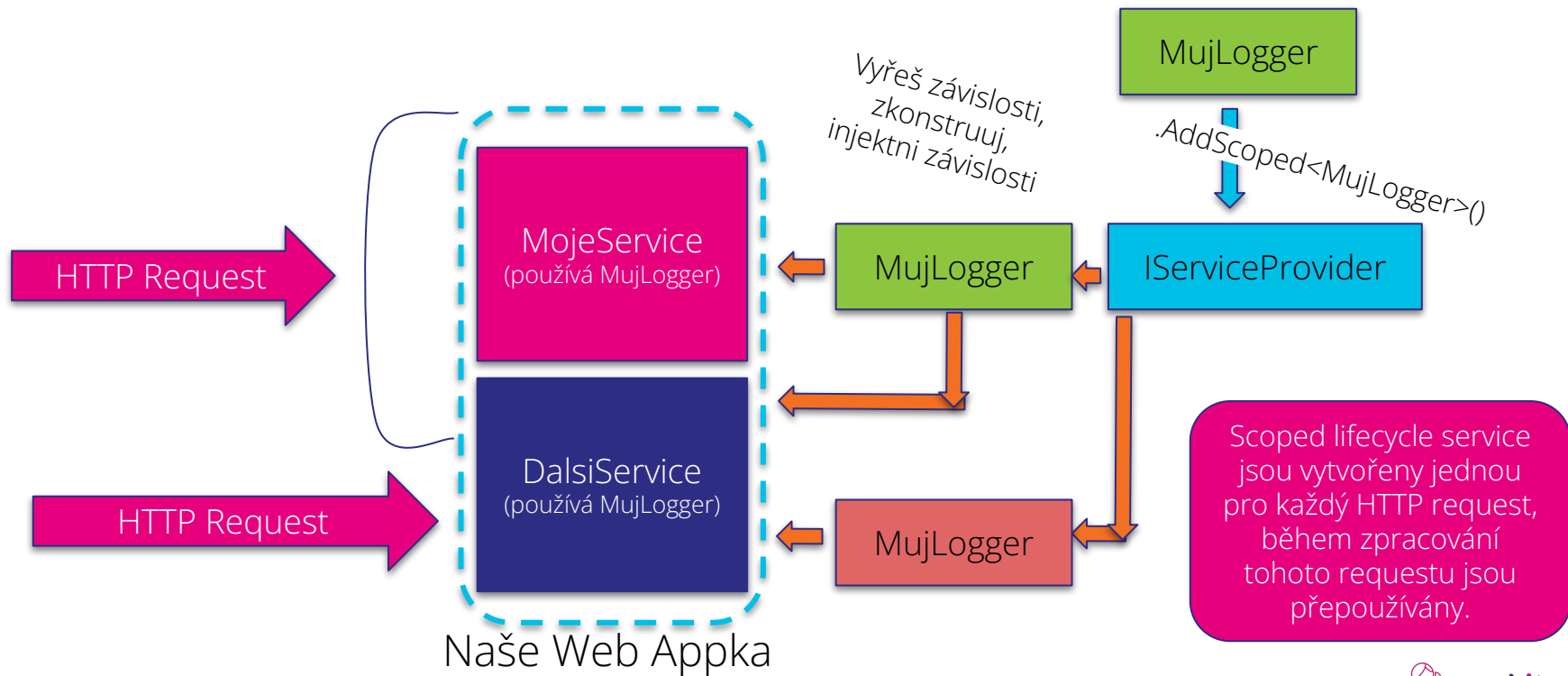
# Kdy by se měly třídy instanciovat?



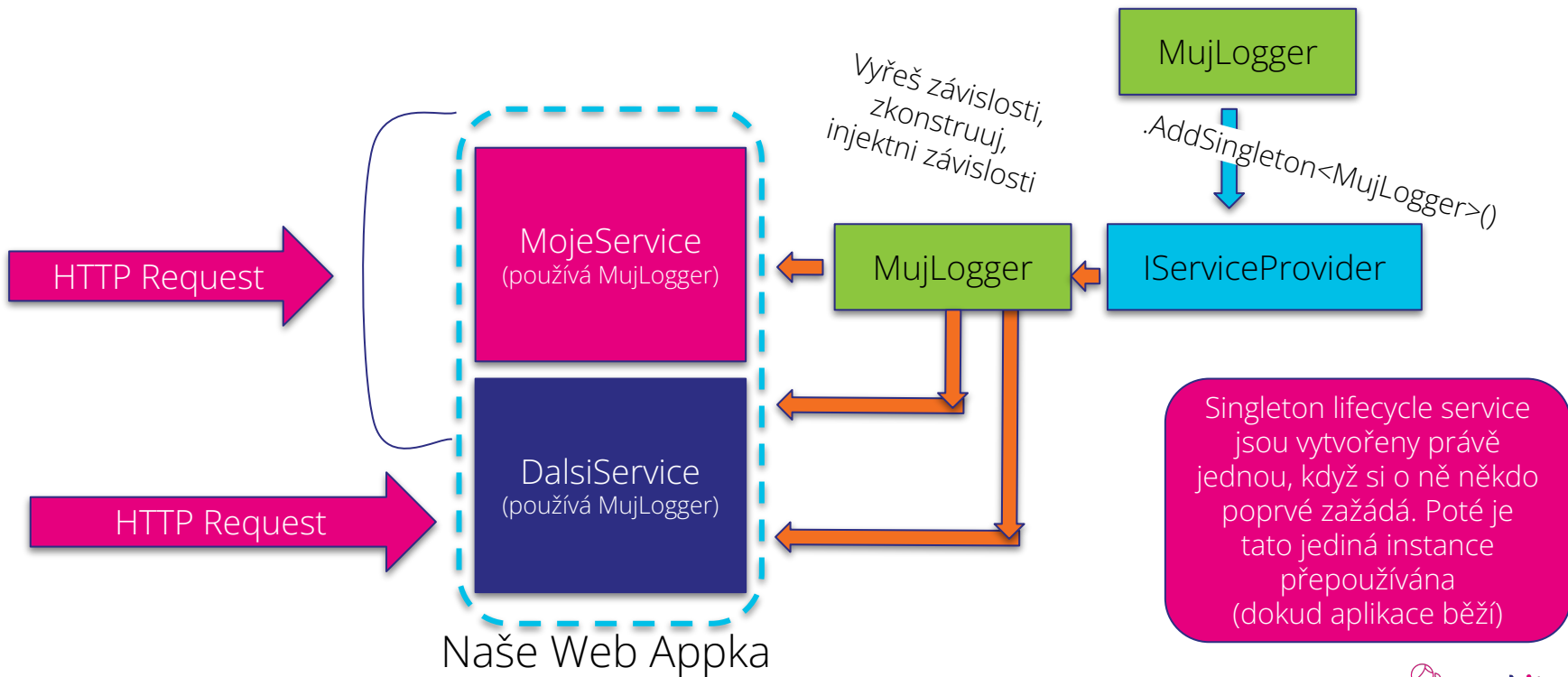
# Transient Service Lifecycle



# Scoped Service Lifecycle



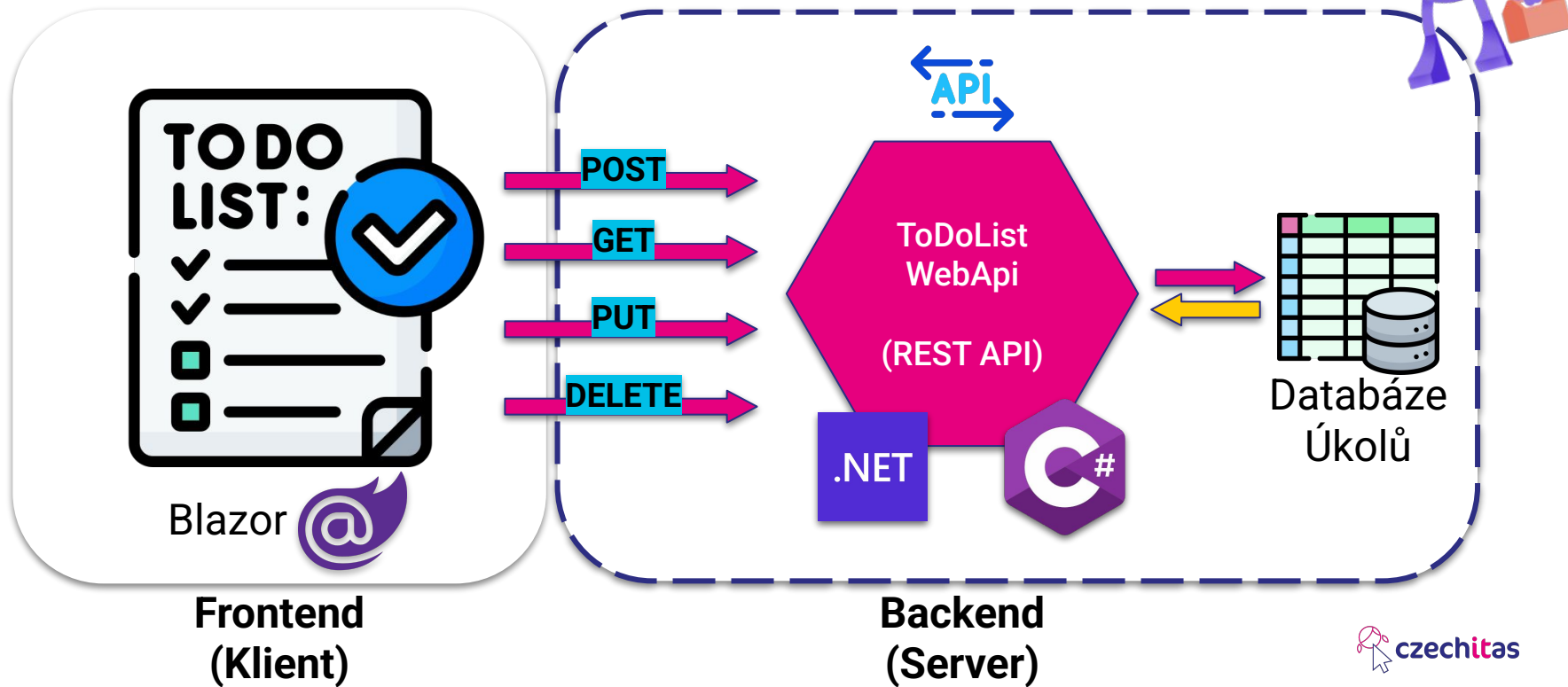
# Singleton Service Lifecycle



## Lekce 08

1. Frontend
2. Co je to Blazor?
3. Interaktivita
4. Komponenty

# Co už máme





# K čemu frontend

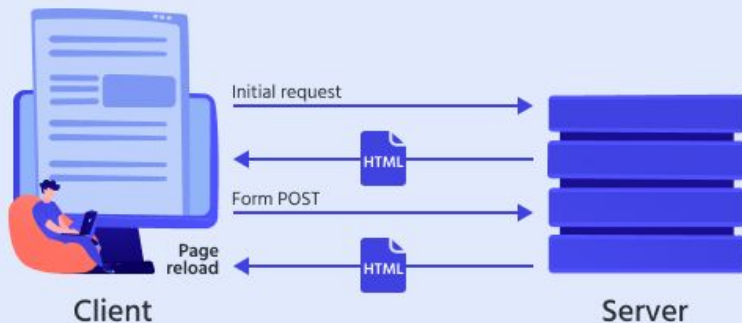
- Sami dokončíme celý projekt
- “Full-stack” porozumění
- Uvidíme, jak se používá naše API

# Co je Blazor

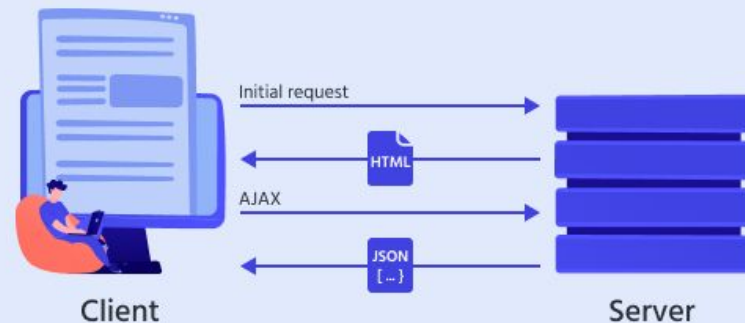
- **Razor** ≈ syntax, který kombinuje HTML a C#
- **Blazor** ≈ framework na tvorbu webových aplikací (pomocí Razoru)
  - Browser + Razor
  - Běží na serveru i na clientovi
  - **S**ingle **P**age **A**pplication

# SPA vs MPA

## Traditional page lifecycle



## SPA lifecycle



# Proč Blazor

- Eliminuje nutnost používat JavaScript<sup>\*</sup>
- Sdílení logiky a knihoven mezi BE a FE
- Open-source, podpora od Microsoftu
- .NET ecosystem (MAUI)

**dotnet new blazor**

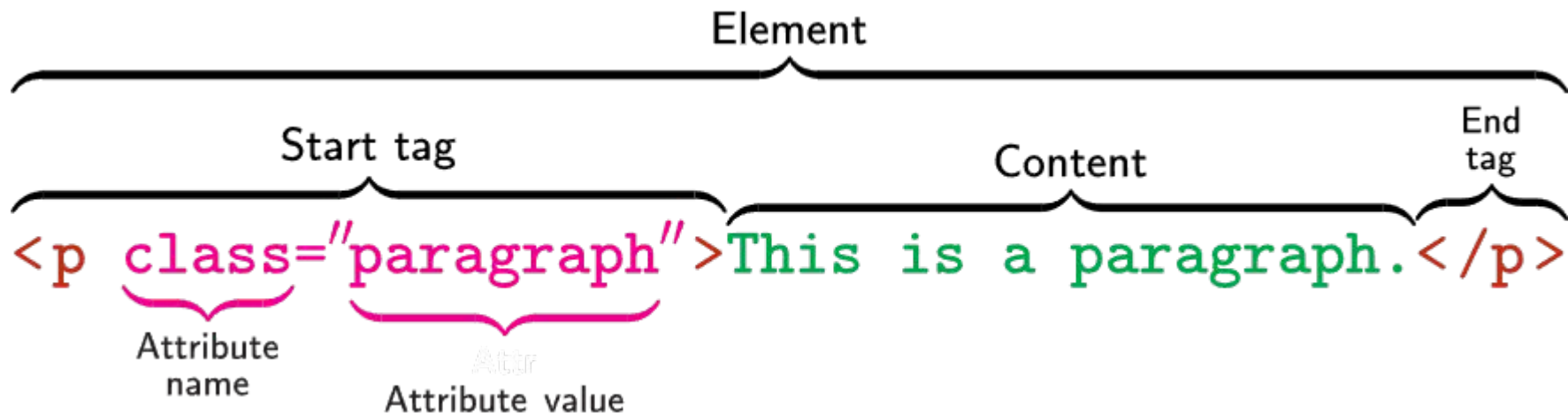


# HTML

# HTML

- **H**yper**T**ext **M**arkup **L**anguage
- Definuje obsah a strukturu
- `<head>` a `<body>`
- Skládá se z HTML elementů
- Elementy vytváříme pomocí otevíracích a zavíracích tagů

# HTML syntax





# Základní HTML tagy

- **<h1>**Titulek**</h1>**
- **<p>**Paragraf**</p>**
- **<br>** - Nový řádek





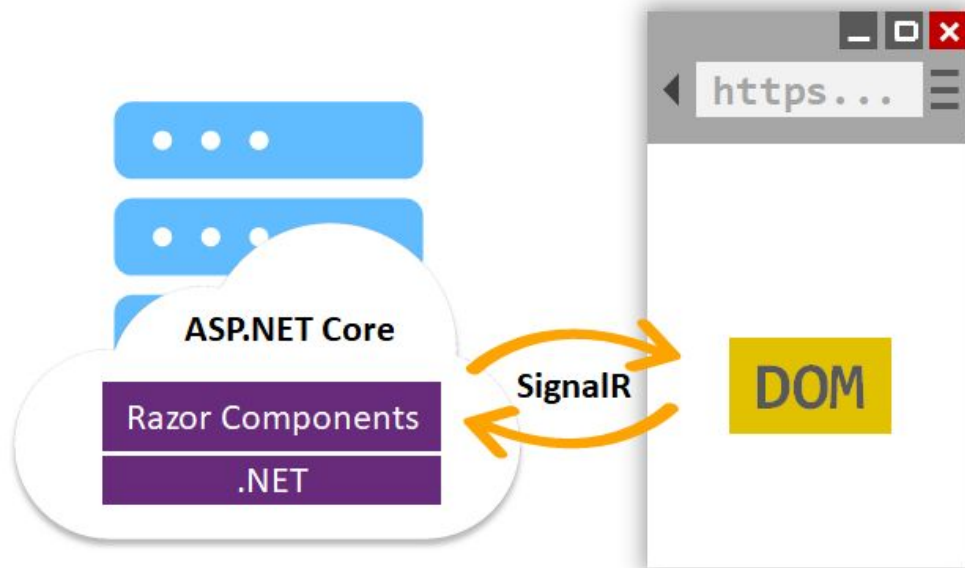
# Interaktivita

# Renderovací typy

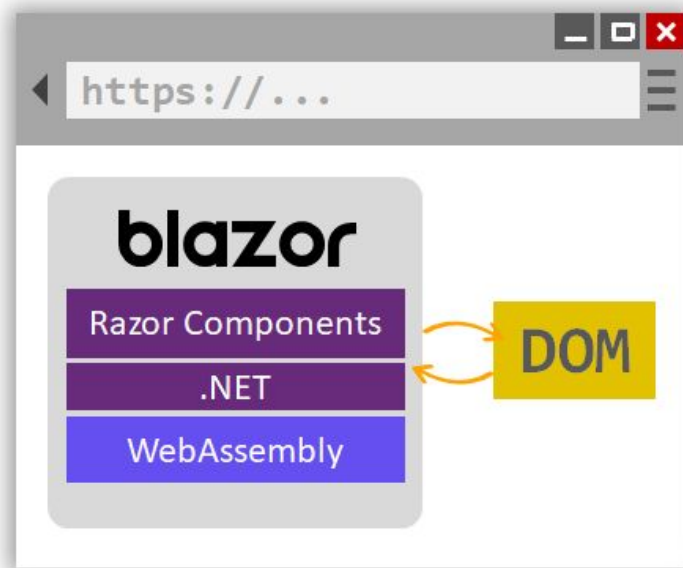
Name	Description	Render location	Interactive
Static Server	Static server-side rendering (static SSR)	Server	✗
Interactive Server	Interactive server-side rendering (interactive SSR) using Blazor Server.	Server	✓
Interactive WebAssembly	Client-side rendering (CSR) using Blazor WebAssembly <sup>†</sup> .	Client	✓
Interactive Auto	Interactive SSR using Blazor Server initially and then CSR on subsequent visits after the Blazor bundle is downloaded.	Server, then client	✓

# Jak propíšeme změny?

## Server-side Blazor



## Client-Side Blazor



**<button>**

```
builder.Services.AddRazorComponents()  
    .AddInteractiveServerComponents();
```

```
app.MapRazorComponents<App>()  
    .AddInteractiveServerRenderMode();
```

**@rendermode InteractiveServer**



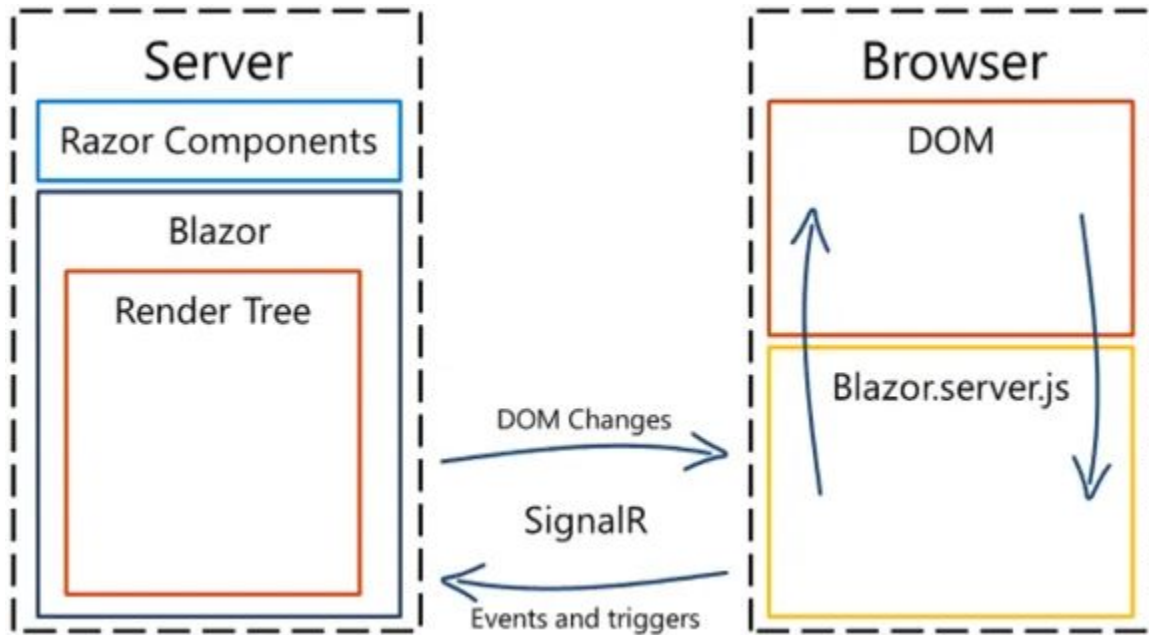


**<Komponenty>**

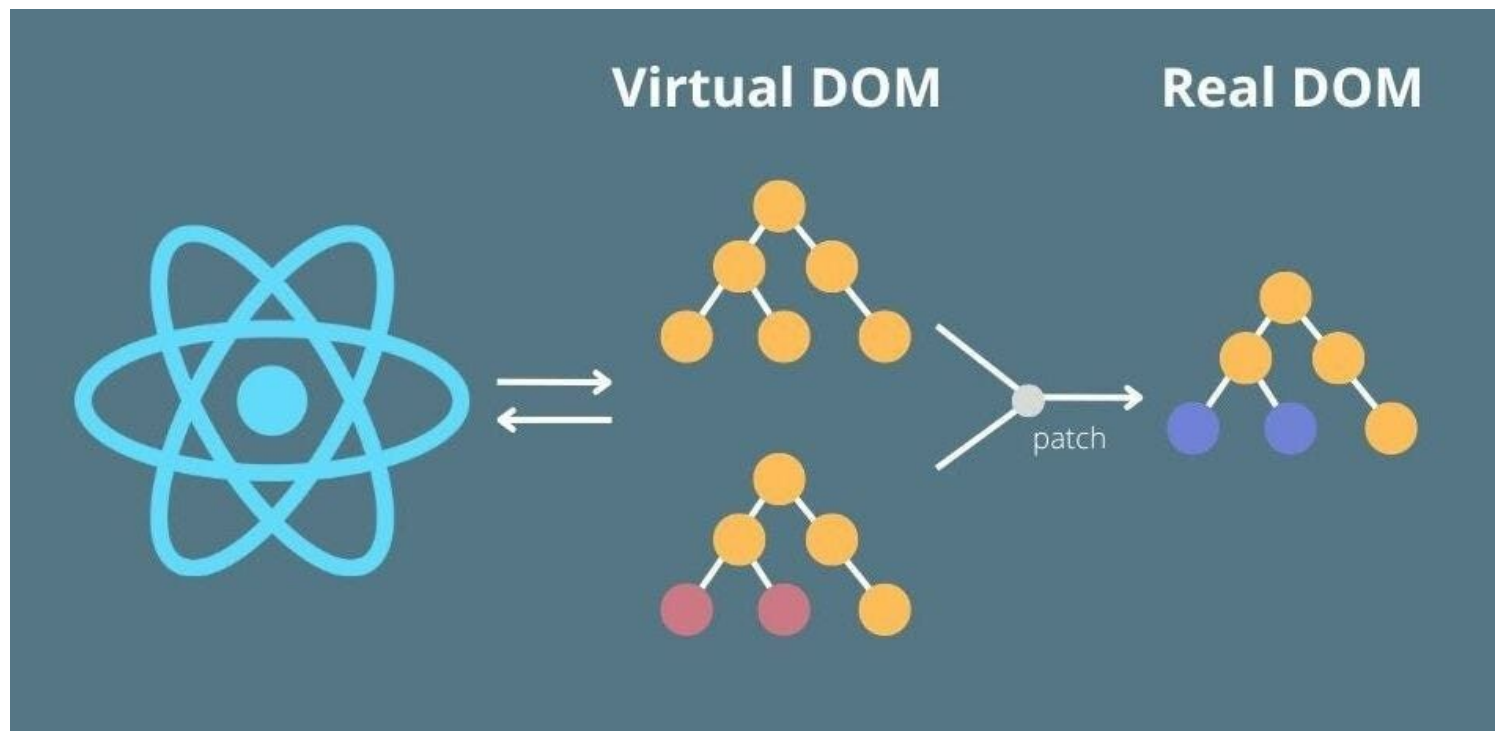
# Komponenty

- Přepoužitelný kus UI
- Zahrnuje HTML, CSS i C# na jednom místě
- Nemusíme manuálně upravovat DOM
- Použití pomocí HTML syntaxi
- Můžeme nakoupit komponentové knihovny

# Komponenty



# Komponenty



# Stránka vs komponenta

- Stránka je komponenta, která má svojí cestu
- @page "/" direktiva zaregistruje cestu ke komponentě

# Další HTML tagy

- `<button>Tlačítko</button>`
- `<table>Tabulka</table>`
- `<tr>Tabulkový řádek</tr>`
- `<td>Tabulková data</td>`

# Extra zdroje na závěr

- [Interaktivní HTML editor](#)
- [Awesome Blazor](#)
- [Blazor school](#)