

# .NET Conf 2022

Da WebAssembly a Blazor... e ritorno

Le novità di Blazor 7

Giampaolo TUCCI

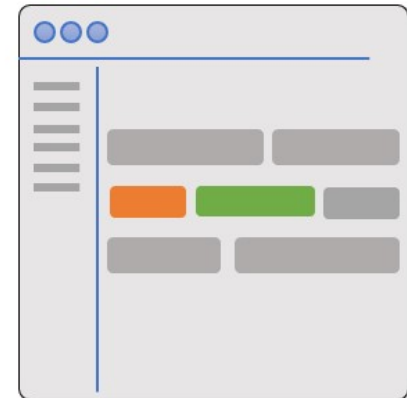
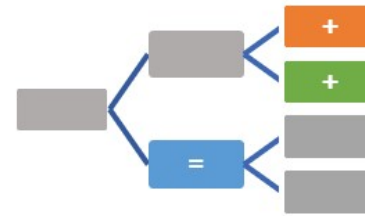
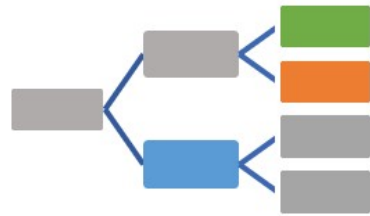
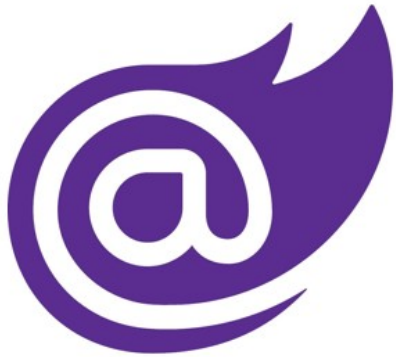


*DotNet  
Liguria*

# Con Blazor è possibile...

- Scrivere logica client che interagisce con la UI della web-app usando .Net/C# in luogo di javascript. In altri termini: è possibile creare webapp SPA senza alcun utilizzo di javascript.
- Il framework favorisce il riutilizzo del codice
  - Utilizzo di componenti
  - Condividere codice lato client e lato server (esempio: POCO class)
- Può riutilizzare le competenze C#/Desktop nell'ambito delle webapp.
- E' possibile riutilizzare molte librerie scritte per .Net standard
- Per chi non riesce a vivere senza javascript si può ! Per questo Blazor interagisce perfettamente con javascript (Javascript Interop)
  - E' possibile richiamare funzioni javascript da codice Blazor C#
  - Da codice javascript è possibile utilizzare classi Blazor/C#

# Blazor in pillole....

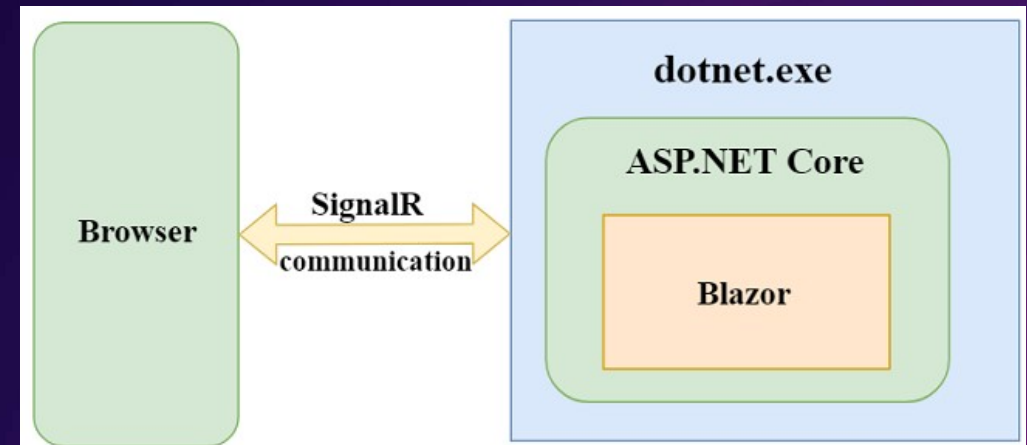
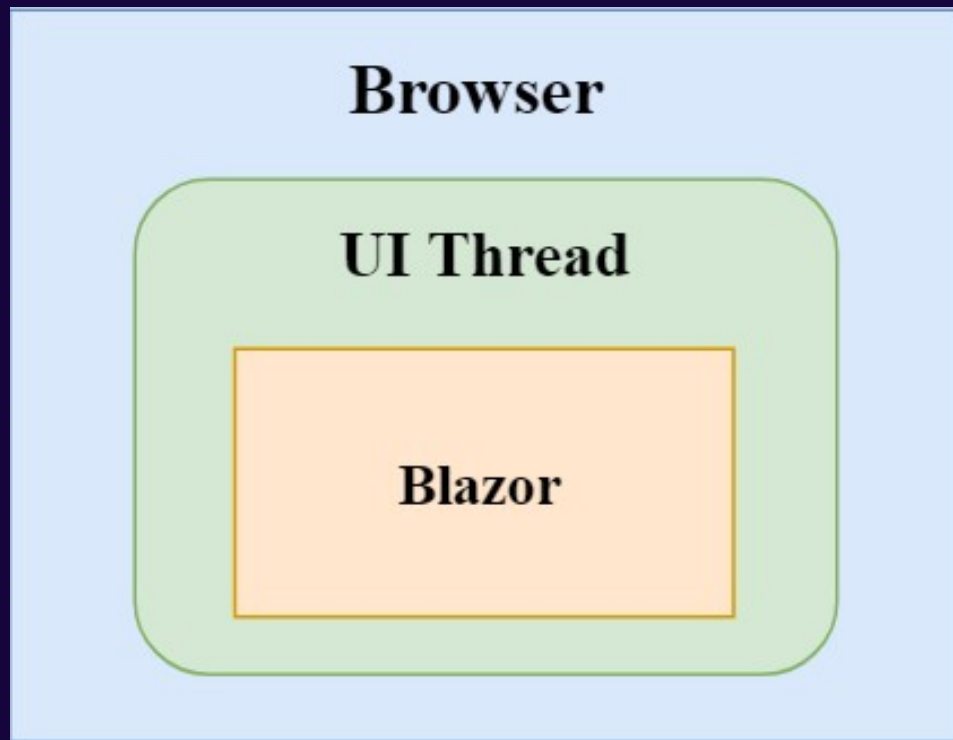


doWork();  
doWork();  
doWork();

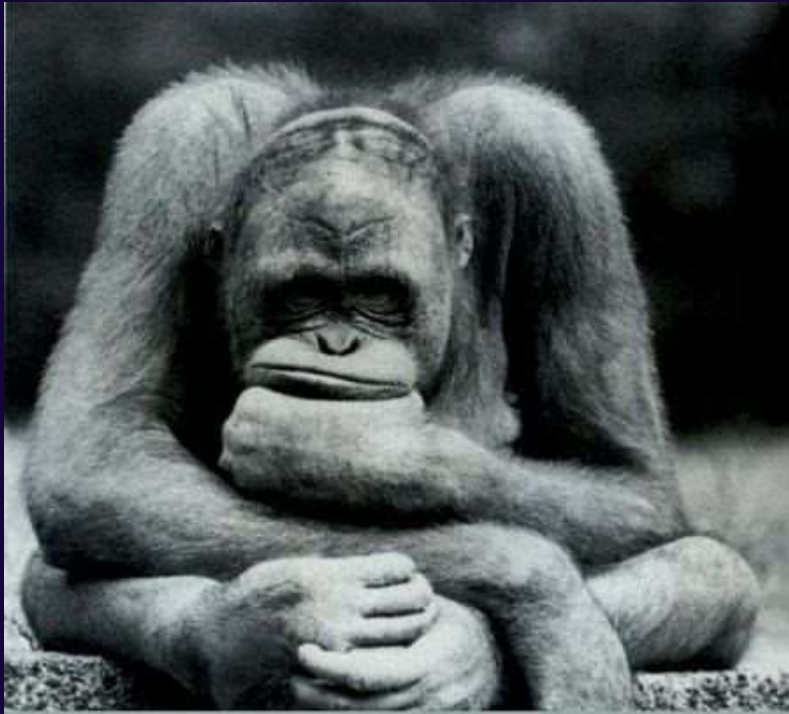
Diff

Update

# Blazor Web Assembly vs Server-Side



# Server-Side vs Client-Side: Pros & Cons



Oh what to do, what to dooo?

## Pros

- Migliore compatibilità con browser poichè al client NON arriva WASM
- Accesso completo all'ecosistema .NET (utilizzo completo delle librerie .NET, tool di debug avanzati)
- Accesso diretto alla base dati
- Il download iniziale della app è molto minore rispetto al client-side (che deve per forza scaricare ad esempio mono.wasm)

## Cons

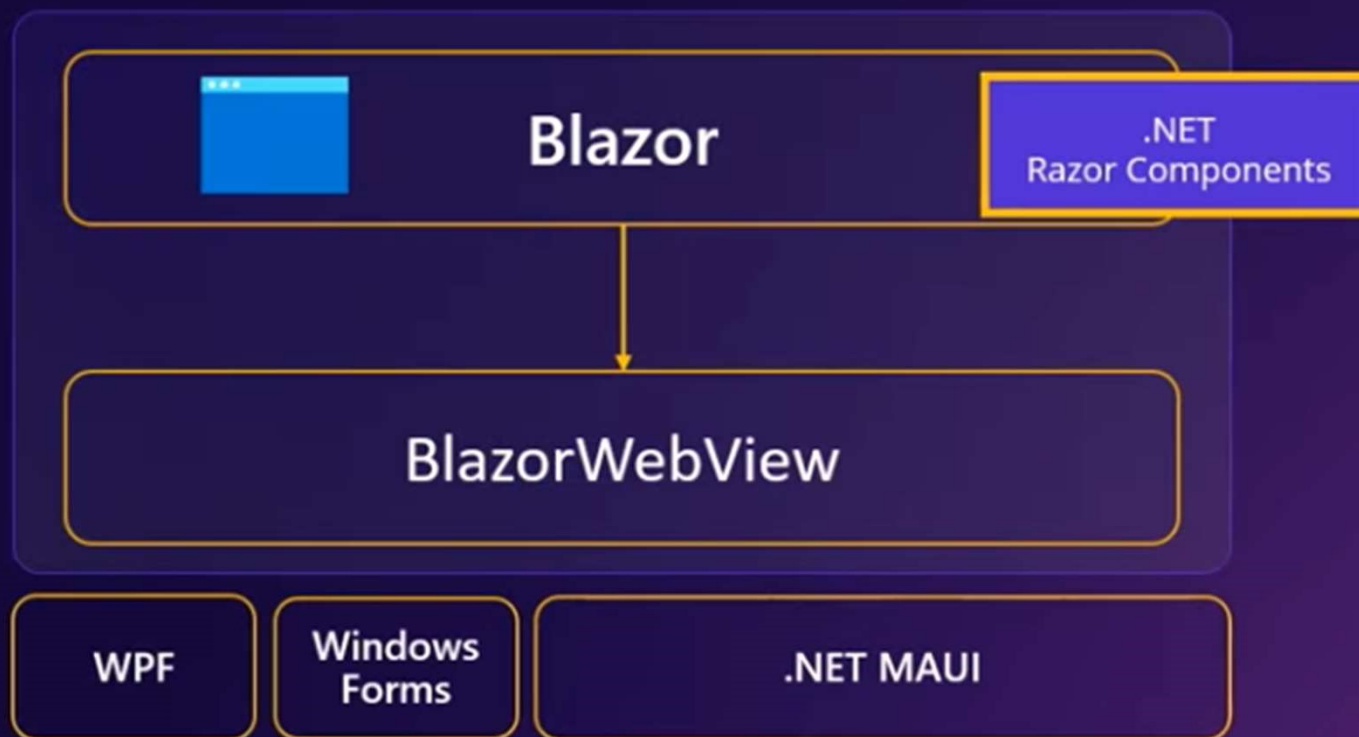
- Qualità e velocità dipendente dalla connessione.
- Scalabilità: dipende dal server.
- Costo hosting più alto (.NET Core)

# Il futuro (?): Blazor Hybrid

.NET offers choice for developers



# Il futuro (?): Blazor Hybrid





# Alcune novità introdotte in Blazor 7

- Quick Grid: advanced grid for advanced interaction with user
- Handle location changing events: Handle change location event in Blazor
- Custom elements: Build standard HTML custom elements with Blazor to integrate Blazor components with any JavaScript-based app.
- Bind after/get/set modifiers: Run async logic after data binding and independently control how data binding gets and sets the data (Att.: Not working ! As of today presents several BUGS!).



# Alcune novità introdotte in Blazor 7

- WebAssembly SIMD & exception handling: Improve performance with .NET WebAssembly ahead-of-time (AOT) compilation using WebAssembly SIMD and exception handling support.
- Wasm-tools: breaking changes
- Blazor WebAssembly: Multithreading via web workers (experimental)
- Improved JavaScript interop on WebAssembly: Optimize JavaScript interop call when running on WebAssembly using new attributes
- Lista complete delle novità

<https://devblogs.microsoft.com/dotnet/announcing-asp-net-core-in-dotnet-7/>