

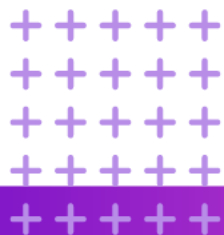


Blazor
Conf 2022



Realizzare un gestionale con Blazor

Andrea Saltarello



Un grazie agli sponsor



E alle community che ci hanno supportato





Premesse

Analisi del testo

Realizzare un gestionale con Blazor

Gestionale

Useremo **Merp** (*My Micro ERP*) un progetto reale, open source (AGPL 3) e disponibile su GitHub:

<https://github.com/mastreeno/Merp>

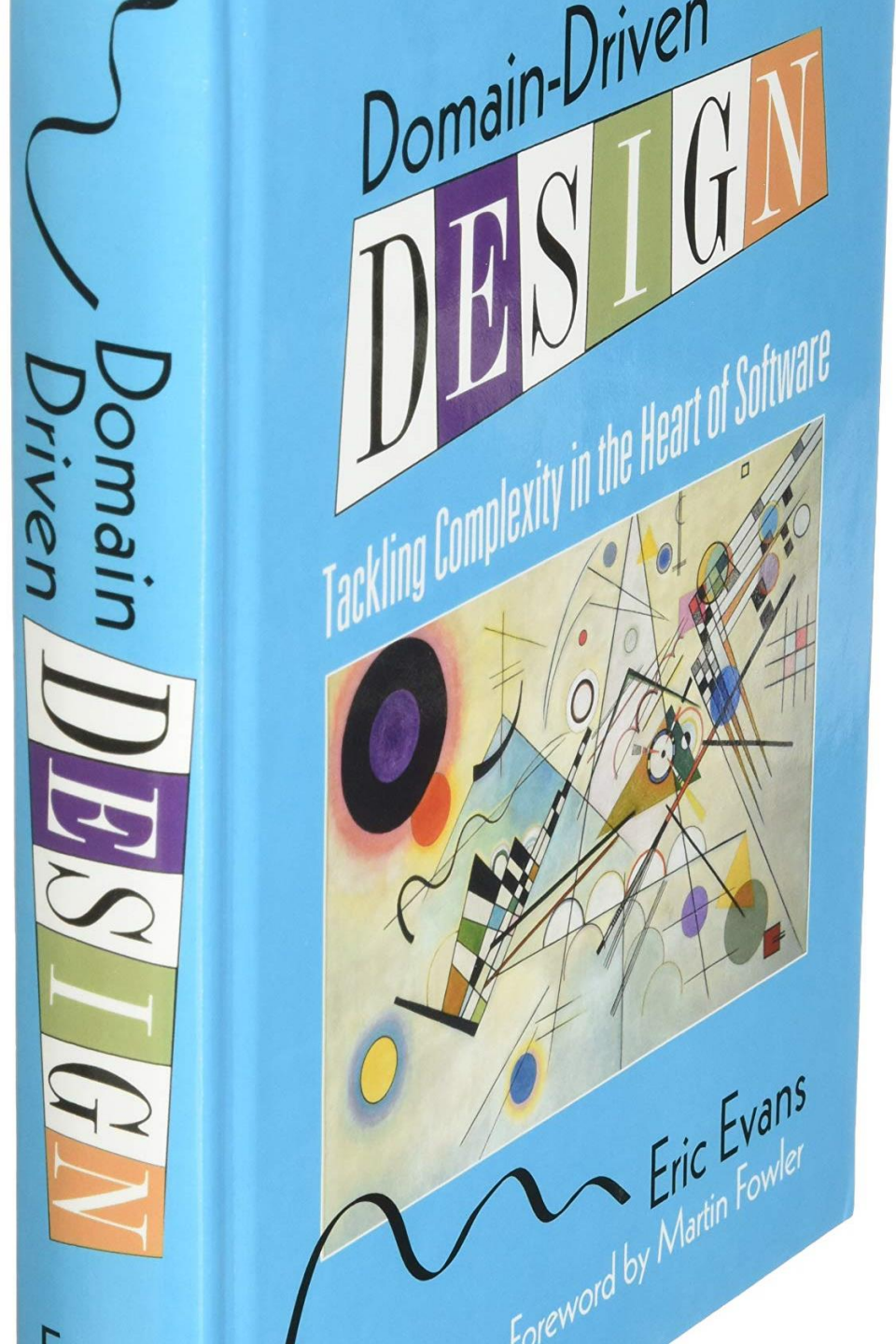
Lavori in corso, oggi useremo la branch «onering» basata su .NET 6



Blazor

Context is King: questa sessione non è una introduzione a Blazor, ci focalizzeremo sugli aspetti peculiari dello scenario in oggetto





Merp 1-2-3

Merp utilizza:

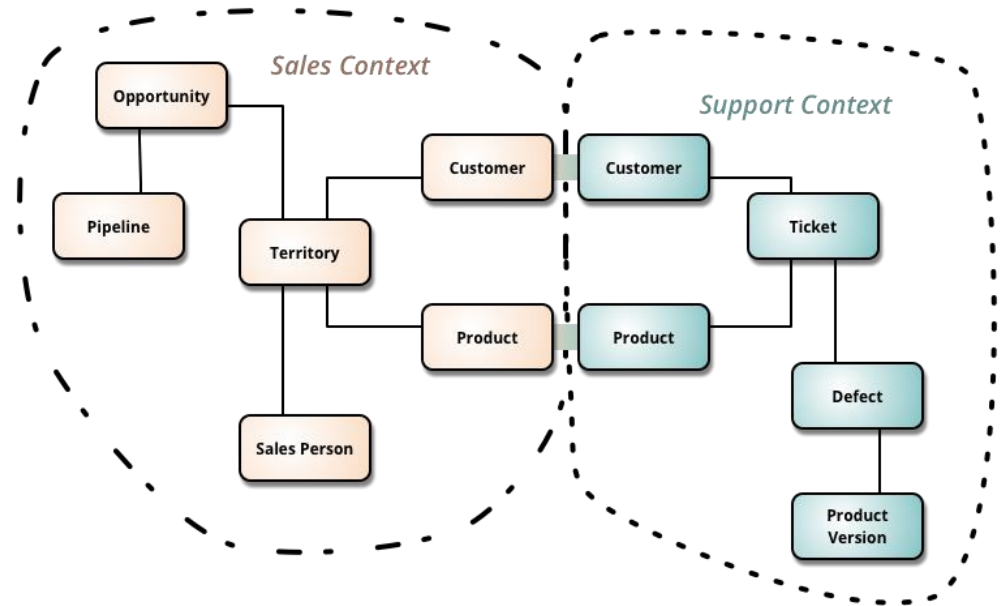
1. DDD come strategia di modularizzazione
2. CQRS come architettura applicativa
3. [OT] Event Sourcing come principale strategia di persistenza

Domain Driven Design

DDD approccia le difficoltà tipiche dei progetti «grandi» mediante una strategia di partizionamento basata su **Bounded Context**.

Ogni BC:

- È autocontenuto e realizzato con scelte ad hoc
- Comunica con gli altri mediante qualche API





demo

Merp project structure

Episode IV

A NEW HOPE

***On an unexpected basis, Microsoft released Blazor
C# developers rebelled against Darth JavaScript***

Blazor <3 Bounded Contexts

E' possibile partizionare i moduli usando:

- Razor Class Library
- Client asset embedded

.NET 7 dovrebbe aiutare ulteriormente in ottica *micro-frontend*



demo

Merp <3 Blazor

Localizzazione 1-2-3

1. Creare un file RESX per ogni lingua
2. Registrare le lingue supportate
3. Usare **Microsoft.Extensions.Localization.IStringLocalizer**

Maggiori info qui: <https://blazordev.it/>



demo

Localizzazione

The background of the slide features the Blazor mascot, a red, blob-like creature with large eyes and a wide, toothy grin, wearing a white shirt. It is positioned on the left side of the frame. The background is a dark space with a white, curved structure, possibly part of a spaceship or a futuristic building, and a starry sky.

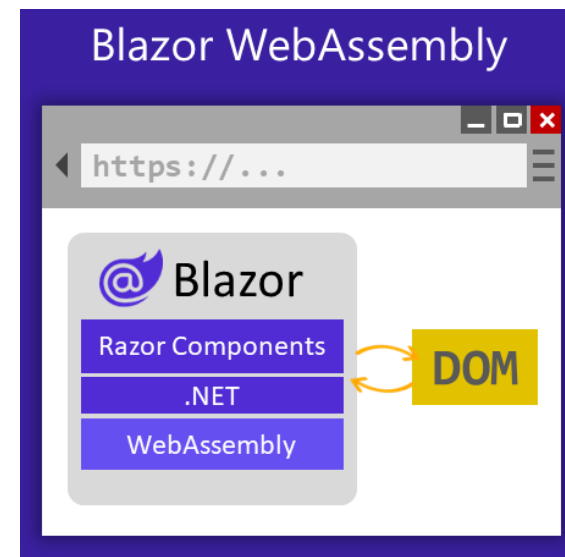
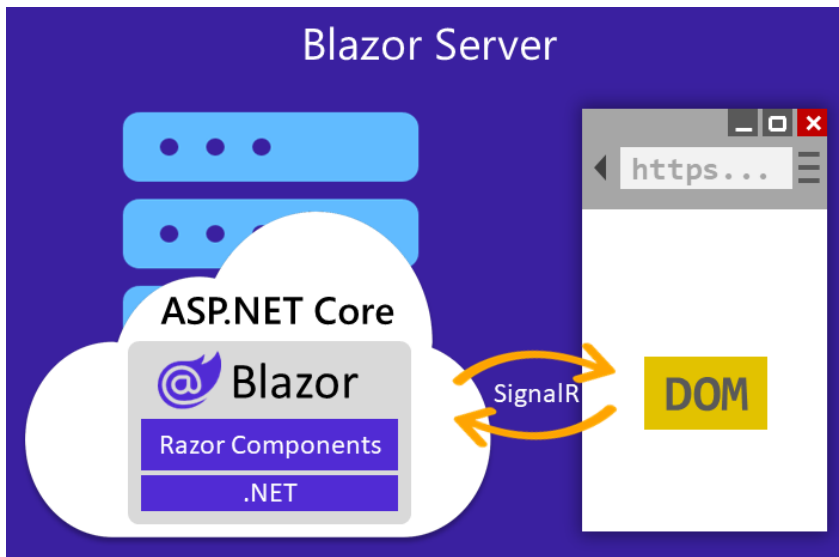
Quale Blazor?

**IT'S A
TRAP!**

Forti differenze architetture tra Server e WASM, che influenzano sia l'architettura complessiva sia le scelte tecnologiche

Blazor: Server vs. WASM

- WASM é *tiered*, Server é *layered*
- WASM é JS-friendly, Server un po' meno





demo

layered vs. tiered

Accessibilità

Blazor fa molto *out of the box*, ma qualcosa dobbiamo mettercelo noi.

Per approfondire:

- [Accessibilità Web, un esempio con Blazor – UGIdotNET](#)
- [Episodio L'accessibilità: perché riguarda anche te, ma ancora non lo sai - Small Talks \(ugidotnet.org\)](#)
- [Episodio Accessibilità, from a software developer's perspective - Ask Me Anything \(ugidotnet.org\)](#)

Me.About();



CTO @ Managed Designs

<https://www.linkedin.com/in/andysal/>

<https://github.com/andysal>

<https://www.ugidotnet.org>

<https://t.me/CloudChampions>

