DDD in salsa Cloud

- Minimal API & Blazor mixed by Azure ServiceBus

Sabato 24 settembre 2022

Alberto Acerbis

Grazie ai nostri sponsor

altitudo

fluentis

the ERP ready to go live!

beanTech

IT moves your business

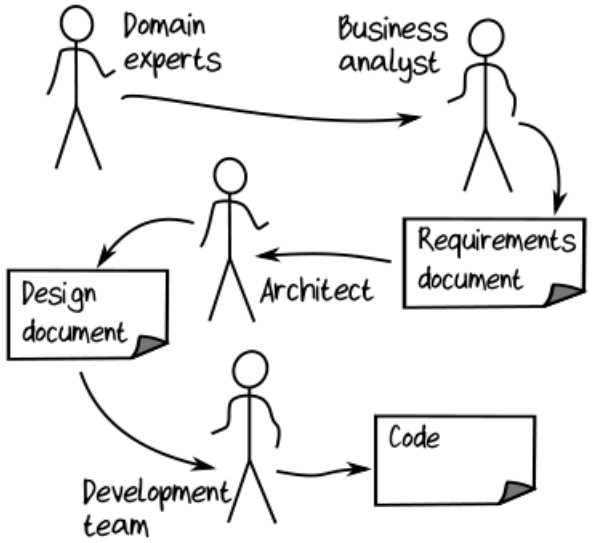
[stesi]

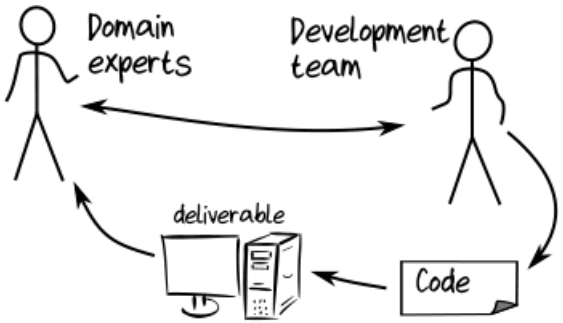
Powered by Innovation

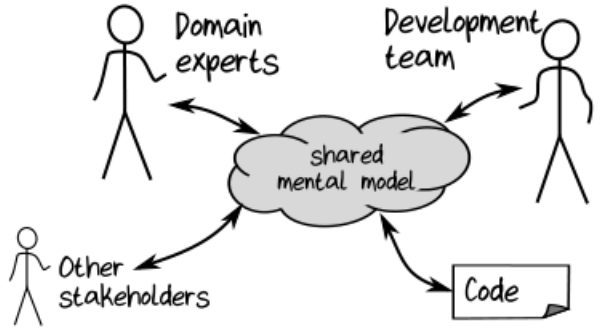
Perchè DDD?

- La tecnologia è destinata a diventare, inevitabilmente, obsoleta!
- La comprensione del Dominio diventa il punto centrale nello sviluppo software
- I Domain Expert sono la nostra risorsa principale
- Non parlano il nostro linguaggio
- Ci serve un Linguaggio Comune
- NON si parte dal database!

Shared Model



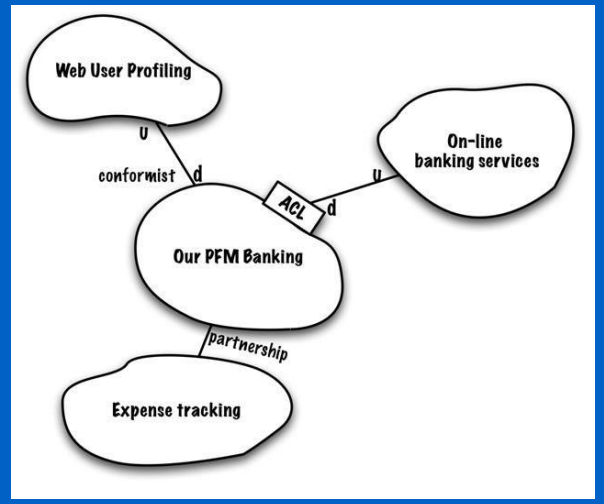




Scott Wlaschin: Domain Modelling made Functional

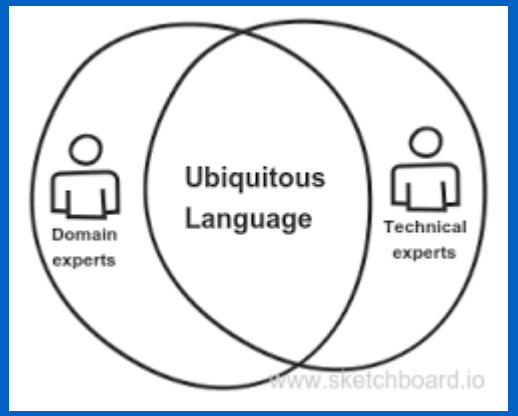
Cosa Propone DDD?

Context Mapping



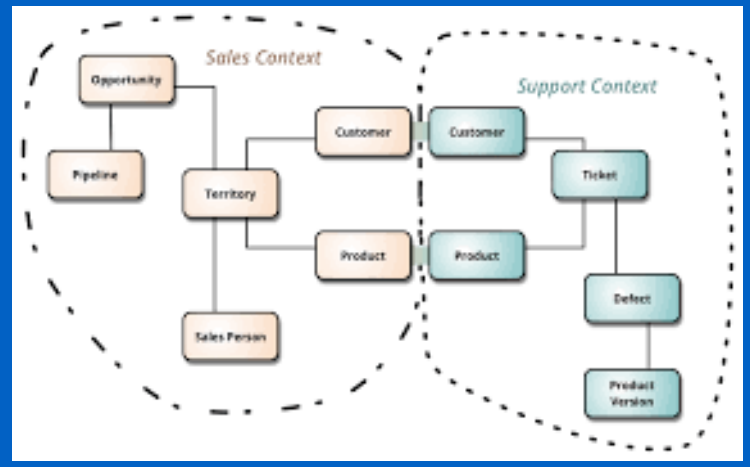
https://www.infoq.com/articles/ddd-contextmapping/

Ubiquitous Language



https://blog.carbonfive.com/ubiquitous-language-the-joy-of-naming/

Bounded Context



https://martinfowler.com/bliki/BoundedContext.html

Come Risolve il Problema DDD?

Feature	Bounded Context	Microservices	Microfrontends
Organized around Business Capabilities	It is implicitly understood in the very concept of Ubiquitous Language, which is the main pattern for identifying a Bounded Context	Cross-Functional Teams specific to a business functionality	Each SCS is owned by One Team
Decentralized Governance	A shared model for each purpose	Local choices, which must be independent, are favored/encouraged.	Autonomous WebApp
Decentralized Data Management	Private persistence is critical for language consistency, but especially necessary for the safe and independent evolution of the model	Each microservice must persist its data in a private database! Otherwise, it will be unable to evolve independently from others	Each SCS has its own API Include Data and Logic
Evolutionary Design	Each model can, and must, evolve independent of the others	Key feature	No Shared UI No Shared Business Code Shared infrastructure can be minimized
Smart endpoints and dumb pipes	Recommended as a strategic model	Key feature. SOA docet!	Asynchronous Communication
Language Consistency	Ubiquitous Language	Key feature	Private Logica and Data

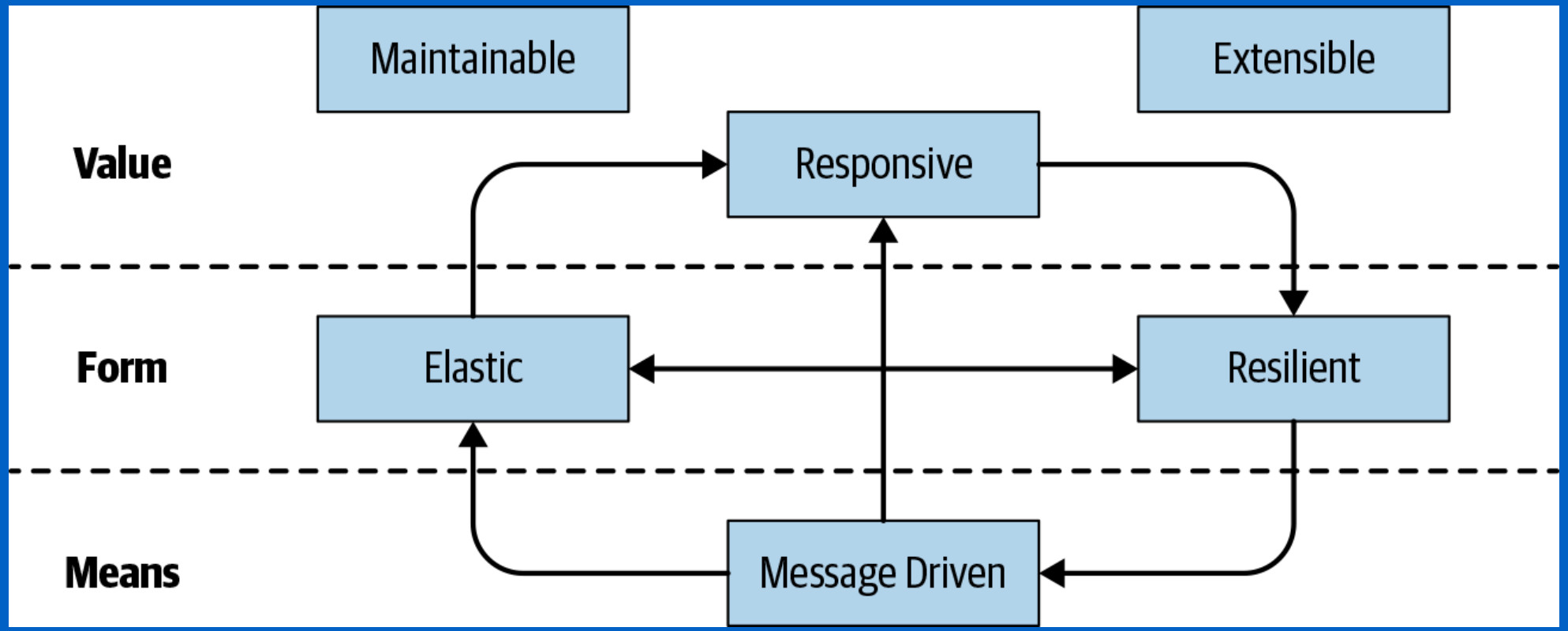
Minimal API & Microservices

- No supporto nativo per content negotiation. Solo JSON
- No supporto nativo per il versioning
- No supporto nativo per la validazione
  - Ma noi abbiamo FluentValidation
- No forzature sulla struttura del progetto
  - E questo ci piace molto!

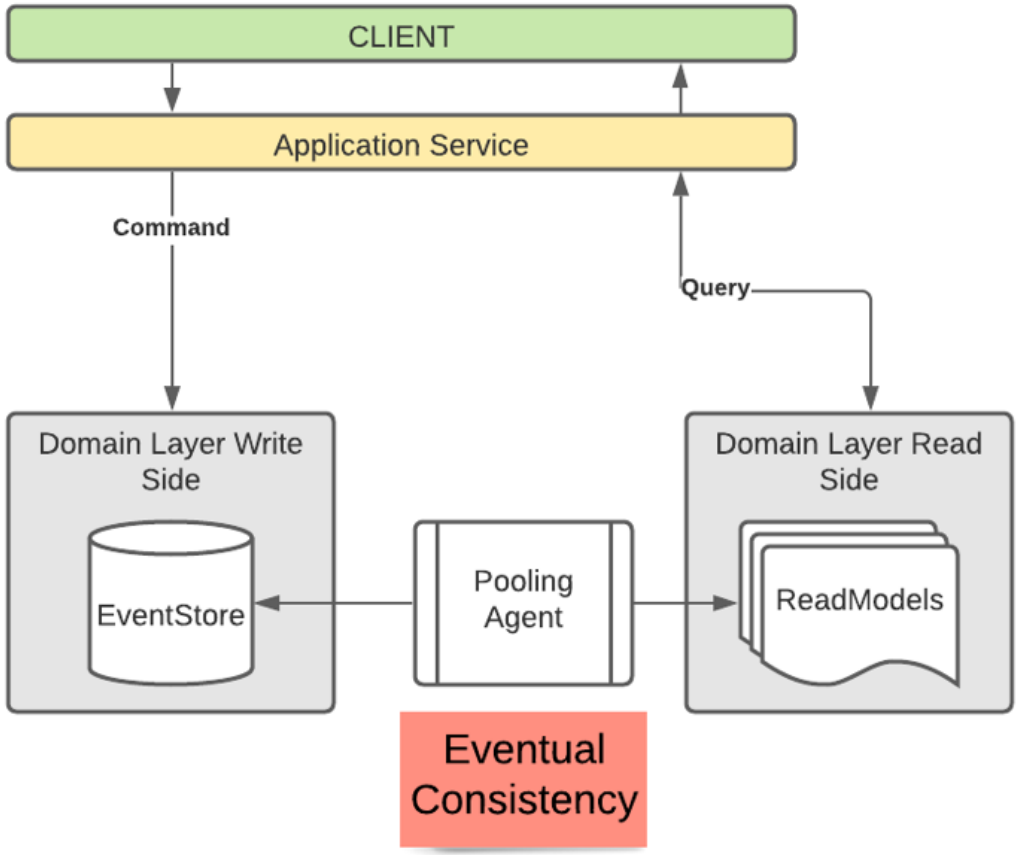
Talk is Cheap … Show me the code

Reactive Manifesto

1. Jones Boner, Dave Farley, Roland Kuhn, Martin Thompson – 16.01.2014
2. The absolute, most important thing is it to be responsive.
   This means that a reactive system needs to remain responsive event when a failure occurs.



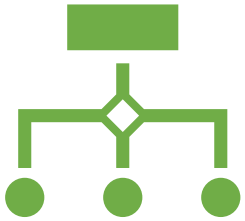
Eventual Consistency



Talk is Cheap … Show me the code

Microfrontend

ThoughtWorks Technology Radar 2016



Self-Contained Systems

- Autonomous Web Application
- Each SCS is owned by One Team
- Asynchronous Communication
- Each SCS has its own API
- Each SCS include Data and Logic
- No Shared UI
- No Shared Business Code
- Shared Infrastructure can be minimized

Cosa sono i Microfrontend?

Microfrontends are the technical representation of a business subdomain, they allow independent implementations with the same, or different, technology choices.

Finally they should avoid sharing logic with other subdomains and they are own by a sigle team.

Microfrontend in Pratica

- Micro-apps with a shared session and parameters
- Routing
- Blazor as a component in an existing project
- Shared components or a Razor class library

Talk is Cheap … Show me the code

Grazie ai nostri sponsor

altitudo

fluentis

the ERP ready to go live!

beanTech

IT moves your business

[stesi]

Powered by Innovation

Saturday ghuva community

Thanks!

Saturday ChNOVA community

alberto.acerbis@intre.it

https://github.com/brewup

https://github.com/cqrs-muflone

Microsoft CERTIFIED AZURE SOLUTIONS ARCHITECT EXPERT

Microsoft CERTIFIED AZURE IOT DEVELOPER SPECIALTY

Sabato 24 settembre 2022