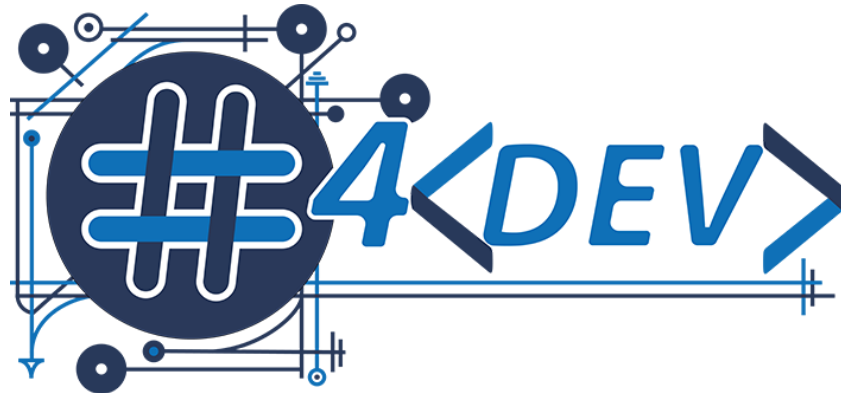




Coding

Refactoring Legacy Code



INTR3 25



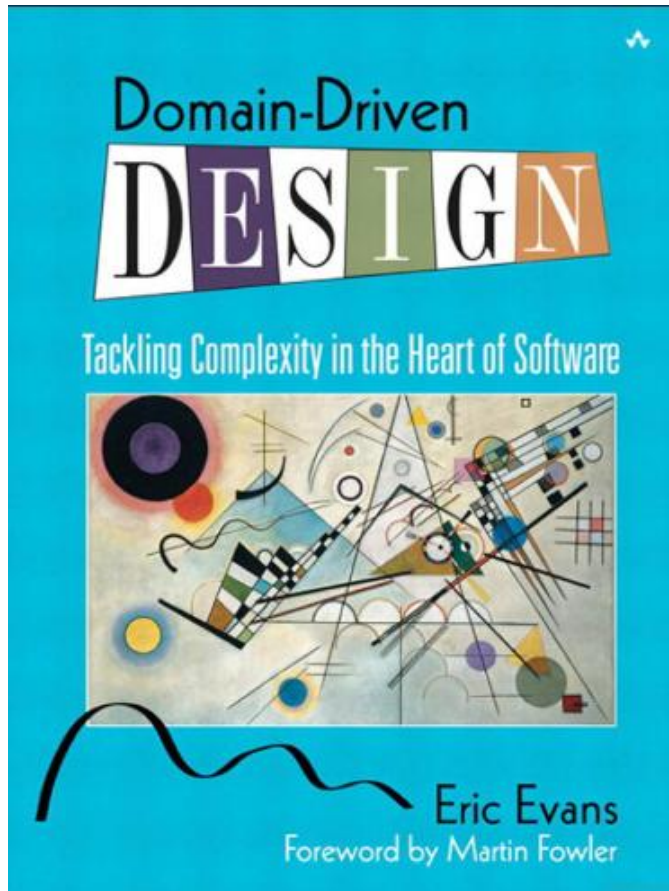
Alberto Acerbis



It was the year 2003



Eric,
we have got
a problem!



**Building great
software products
to solve business
problems**



**Your job is not to write working code,
your job is to design a working system.**

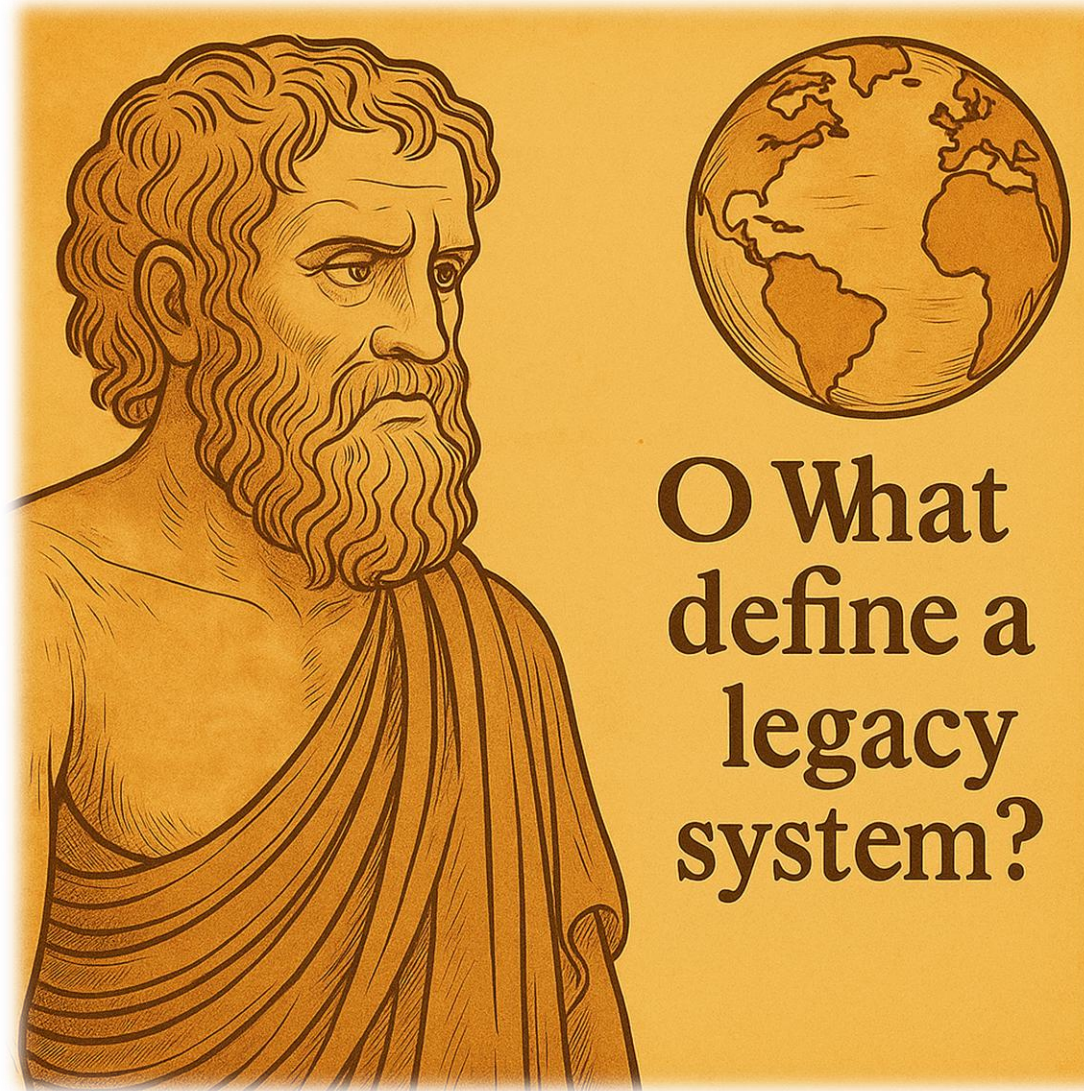


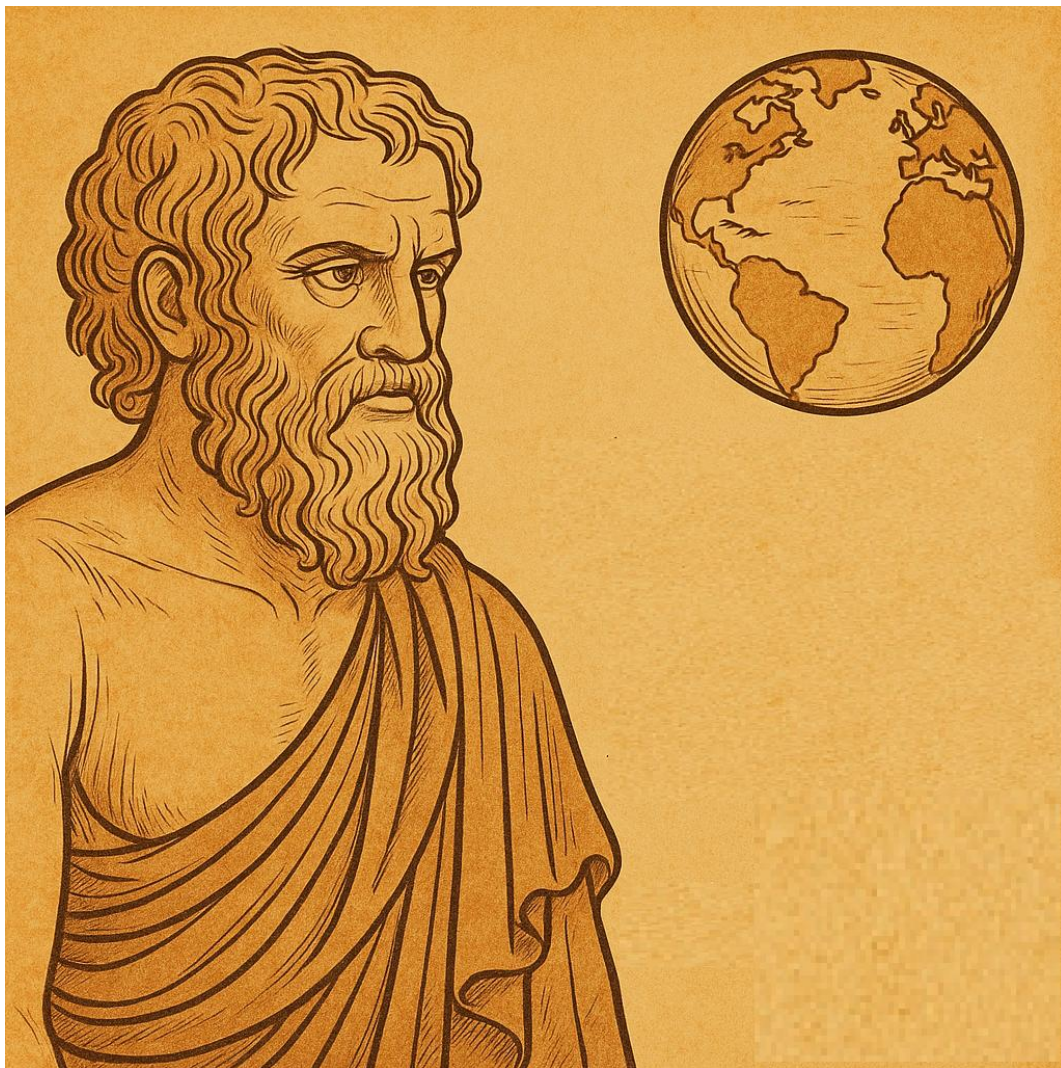
Greenfield

- Nessun vincolo da sistemi o processi esistenti.
- Libertà totale nelle scelte architetture e tecnologiche.
- Maggiore creatività, ma anche più incertezza.
- Rischio di reinventare ciò che già funziona.

Brownfield

- Si parte da sistemi, infrastrutture o processi già esistenti.
- Sfida principale: integrazione e compatibilità.
- Maggiori vincoli, ma anche meno incognite.
- Approccio pragmatico, spesso più rapido nell'apporto di valore.





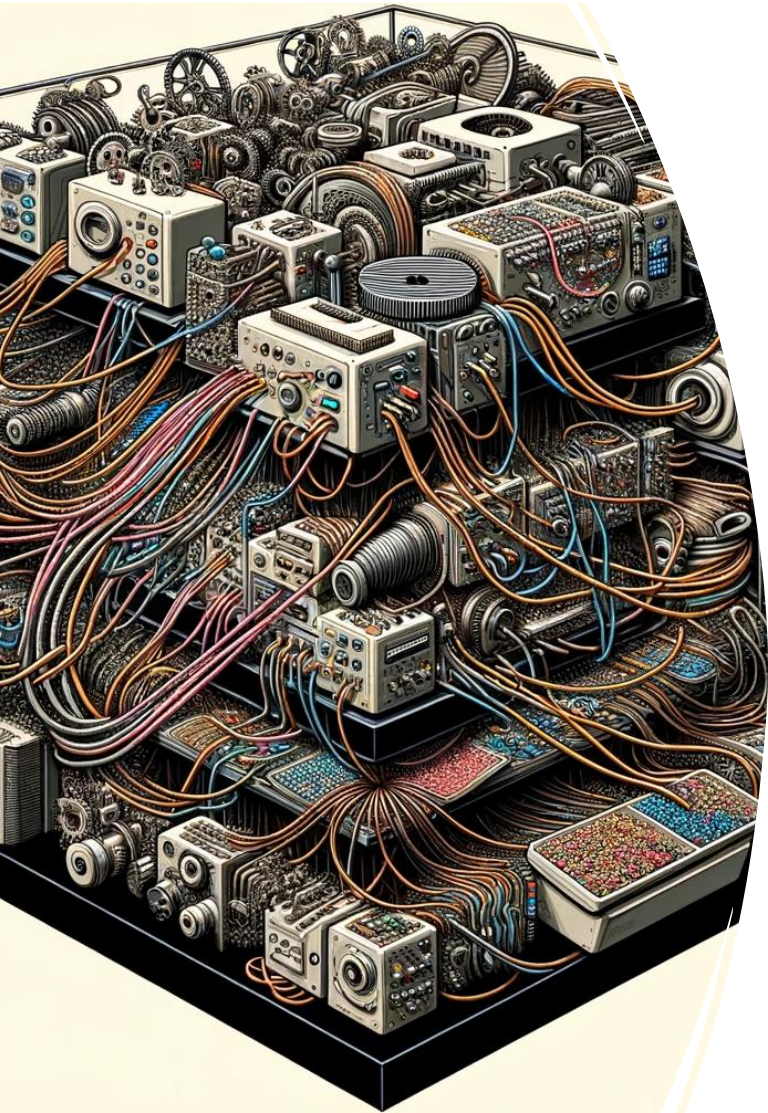
Panta Rhei

- Un Sistema Legacy è ciò che resiste al fluire, ciò che non scorre.
- E' il fiume che non cambia corso, mentre tutto intorno ad esso muta.
- Il nuovo che avanza e il vecchio che persiste.



Big ball of mud

Mancanza di un'architettura chiara, che porta a un sistema strutturato in modo casuale e difficile da comprendere o modificare.

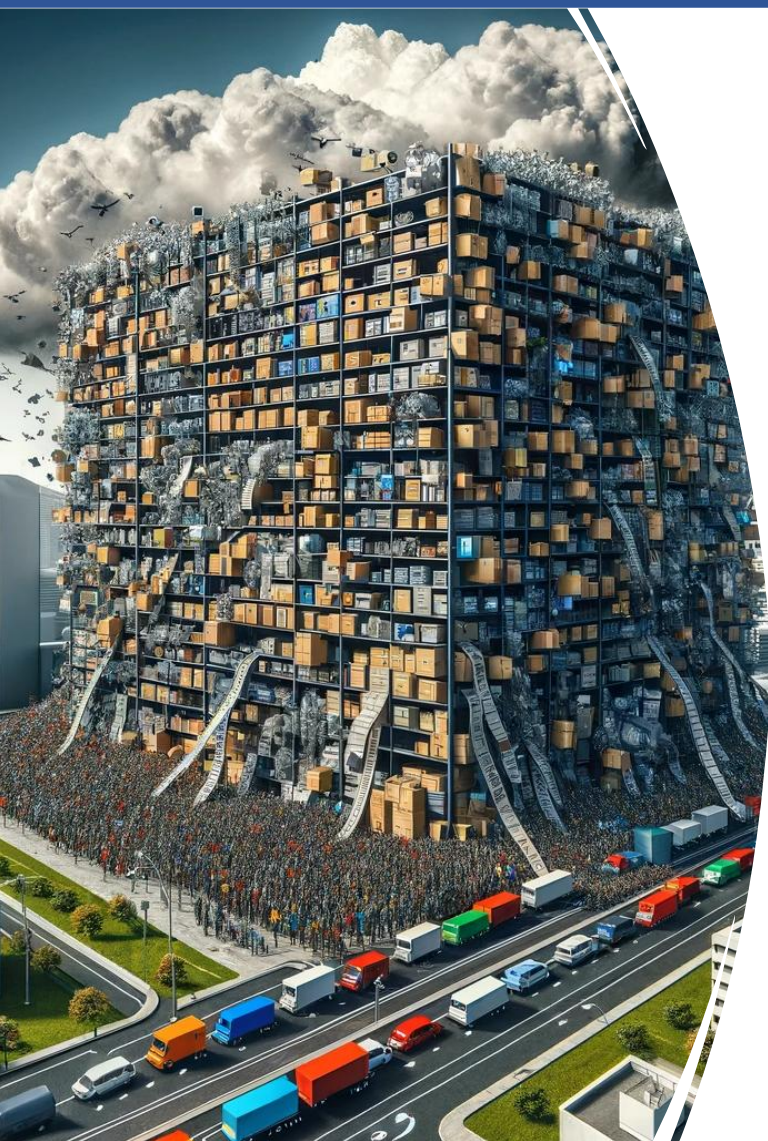


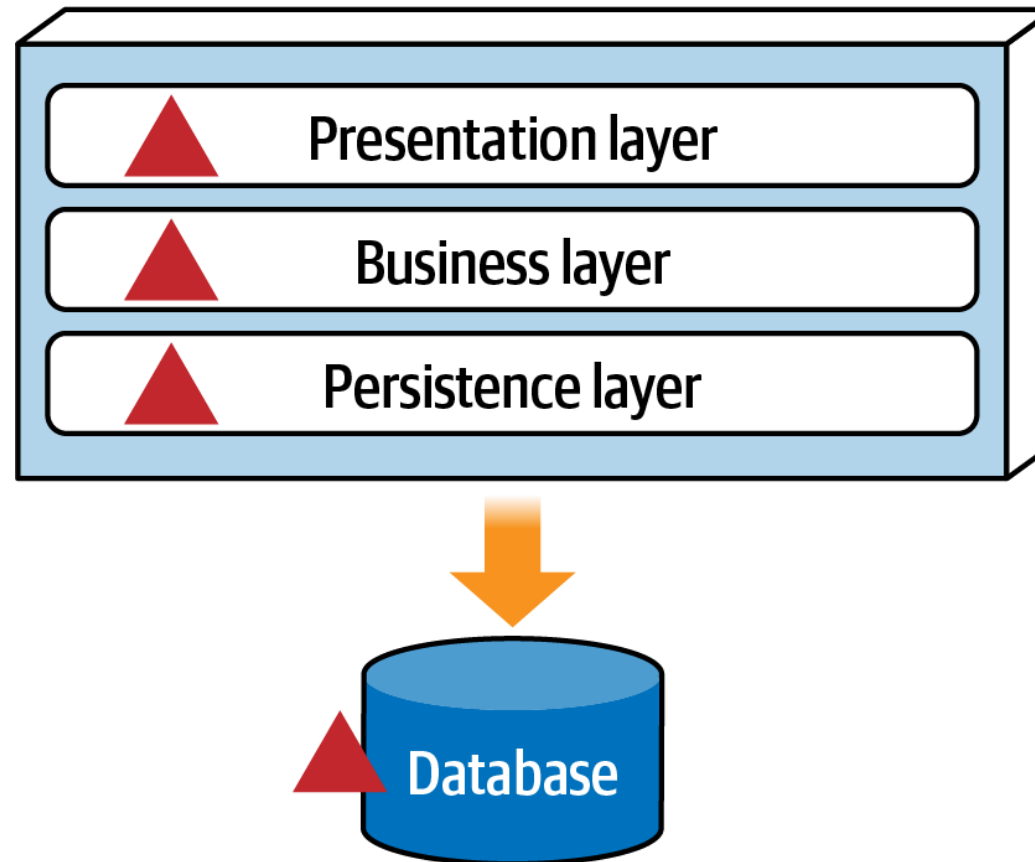
Lack of modularity

Mancata corretta incapsulazione delle diverse funzionalità, con conseguente creazione di un sistema in cui le modifiche apportate a un modulo si ripercuotono sugli altri.

Monolithic database design

Un unico database per tutte le esigenze applicative, che crea un collo di bottiglia e complica qualsiasi tentativo di scalabilità o separazione delle problematiche.

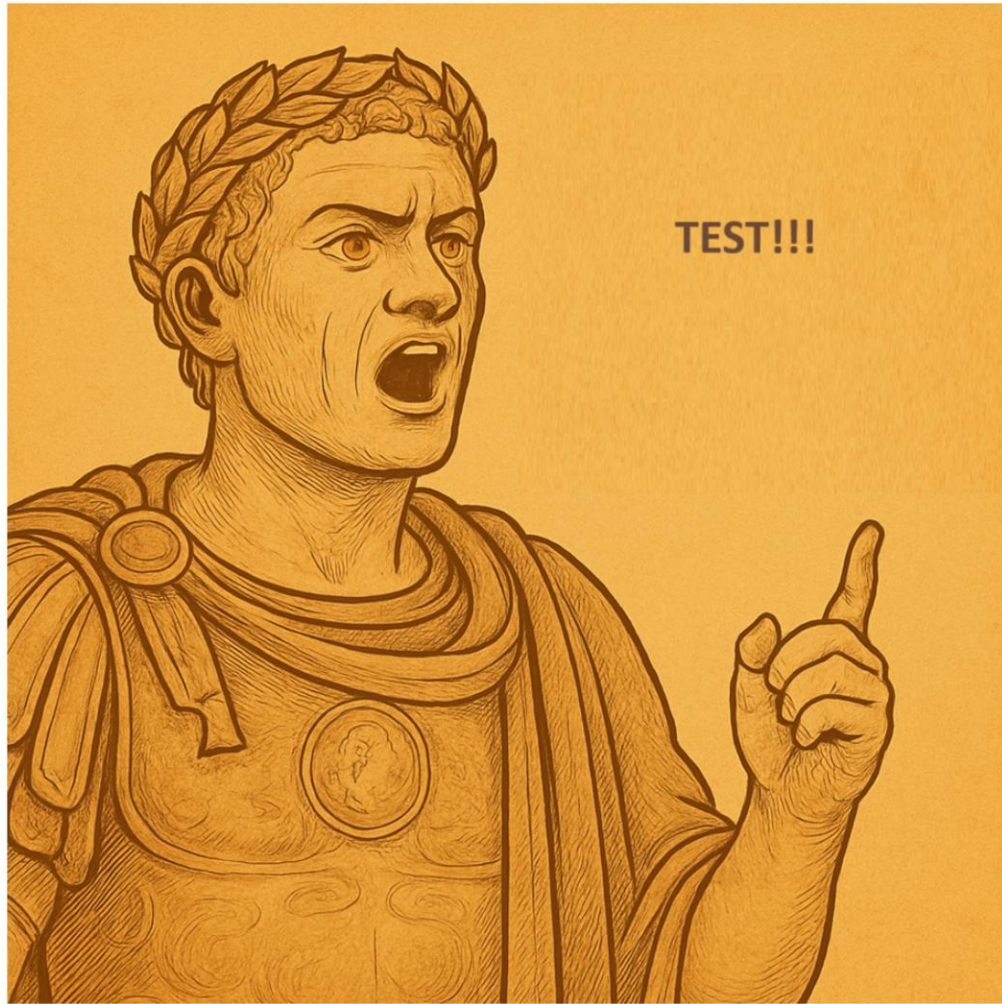




Application-level change scope

(Triangle represents where change occurs)



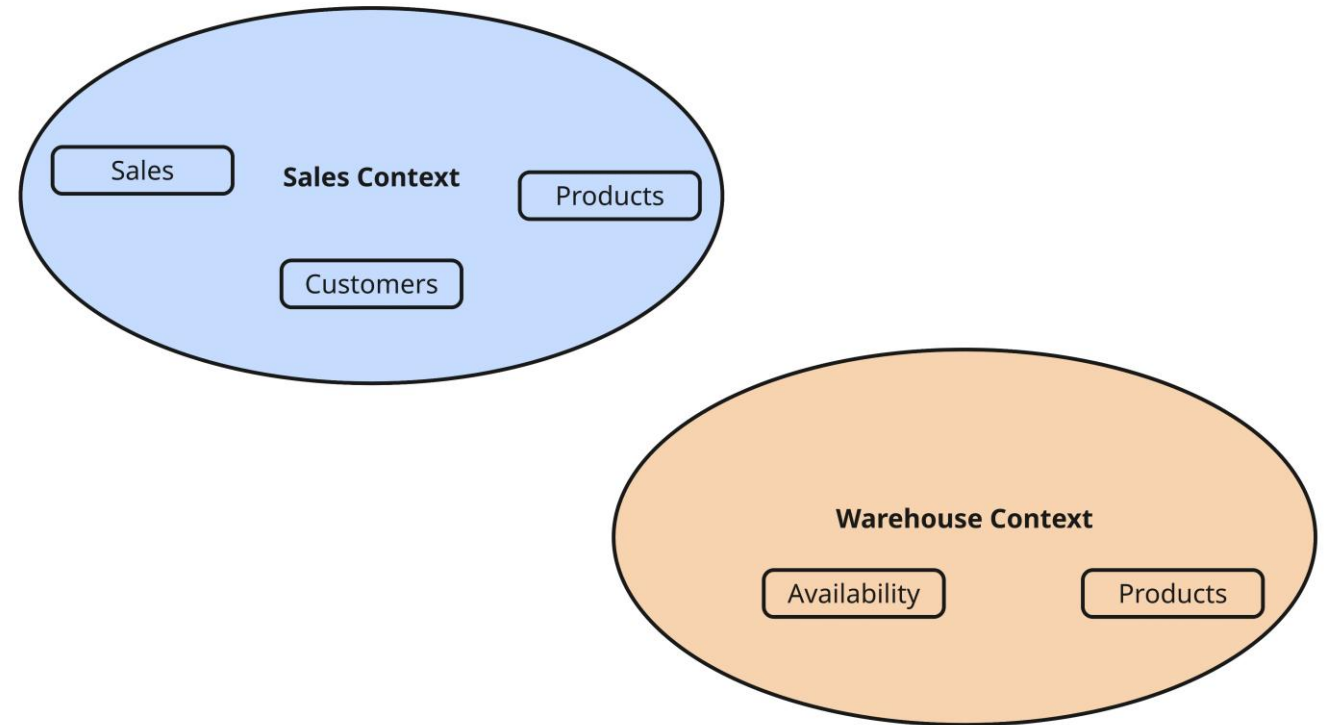


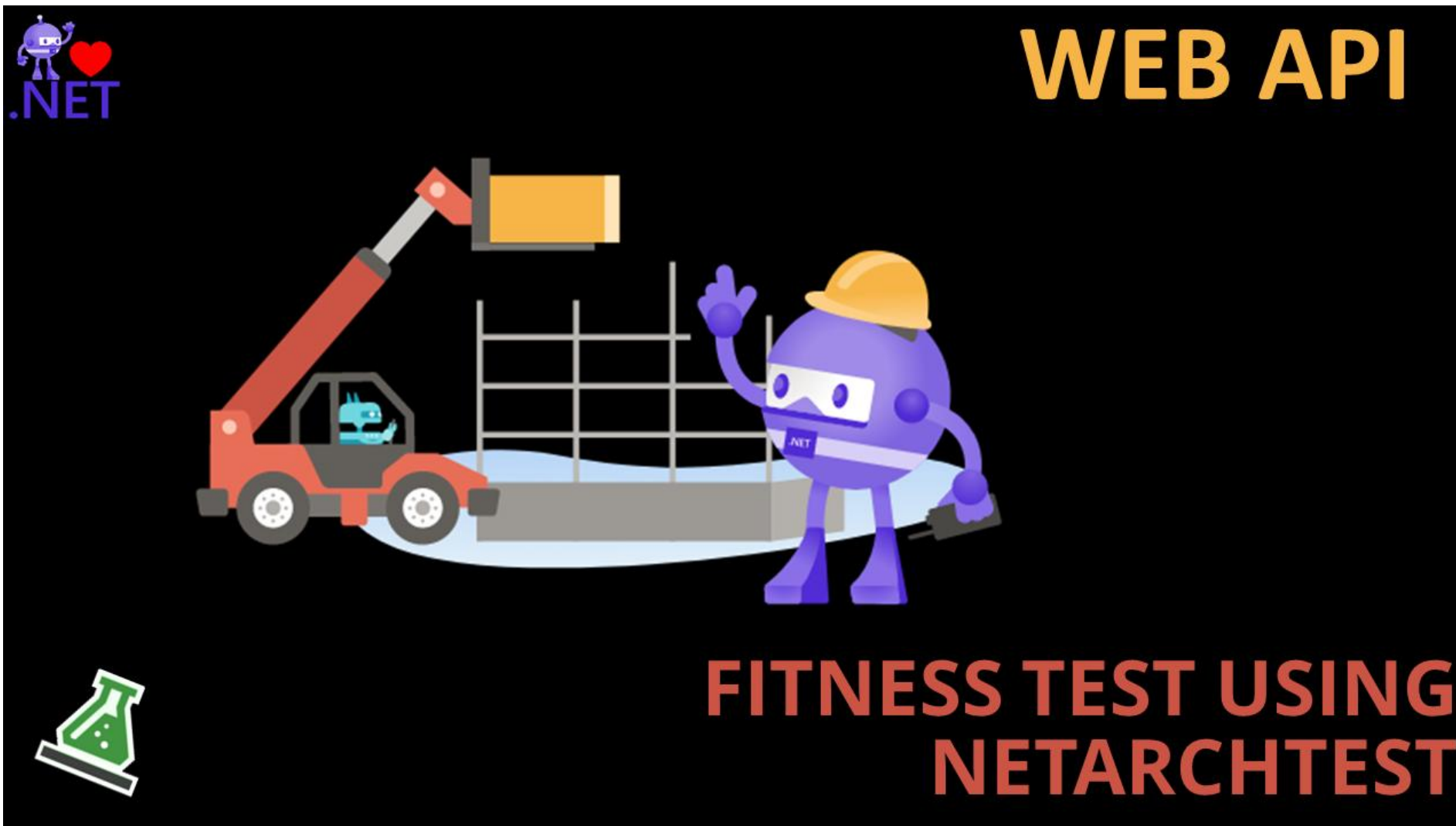
Test di Integrazione

- 🚦 *Il test di integrazione è il semaforo verde verso la produzione.*
- 🌉 *Un ponte sicuro tra codice e valore.*



Context Mapping

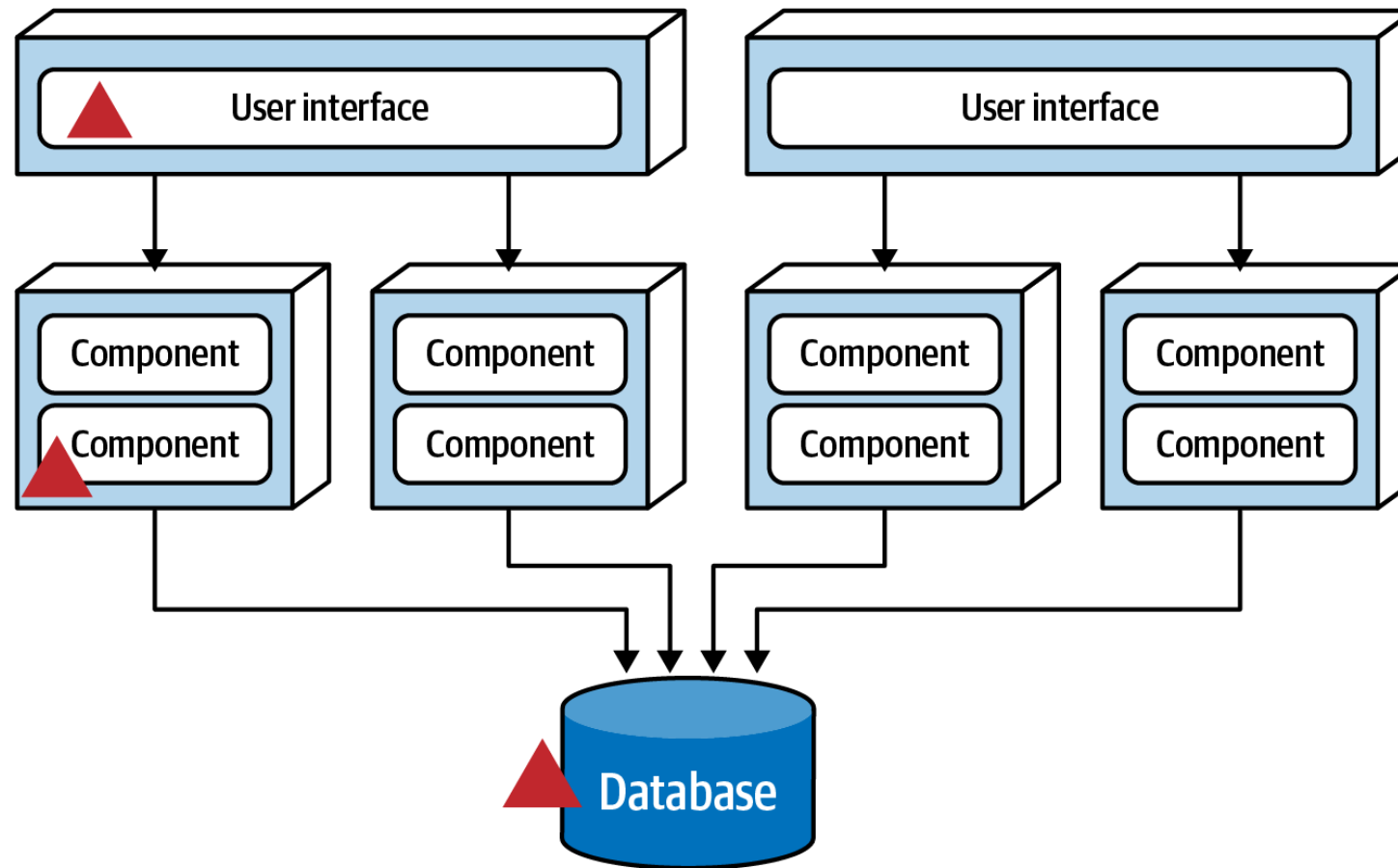


The illustration is set against a black background. On the left, a red crane with a yellow bucket is lifting a yellow rectangular block. A small blue robot is visible in the crane's cab. In the center, a purple robot wearing a yellow hard hat and safety goggles stands with one hand raised in a 'stop' or 'wait' gesture. The robot has a small blue tag with the text ".NET" on its chest. Behind the robot is a white grid structure. In the bottom left corner, there is a small green icon of a beaker with a white base. The text ".NET" is also visible in the top left corner of the illustration area.

WEB API

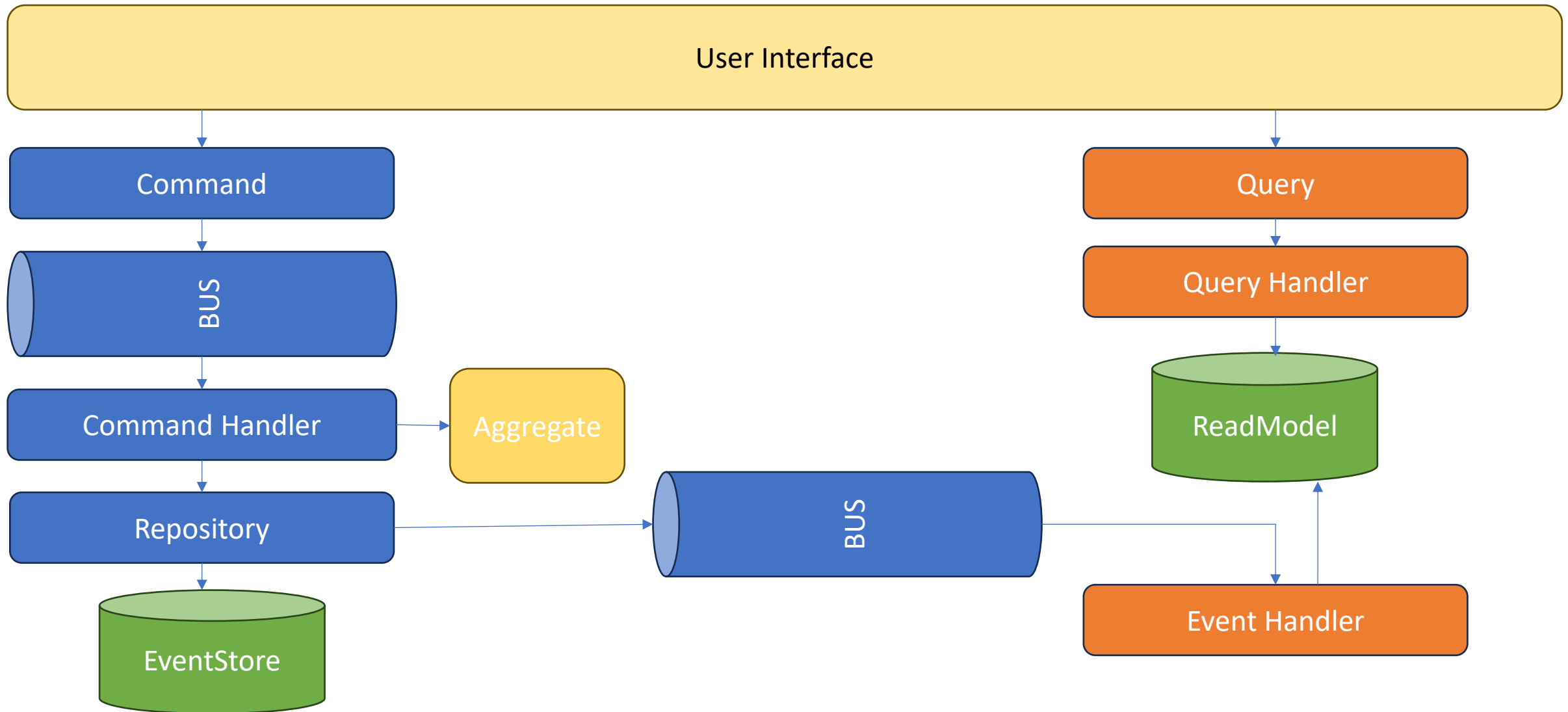
FITNESS TEST USING NETARCHTEST

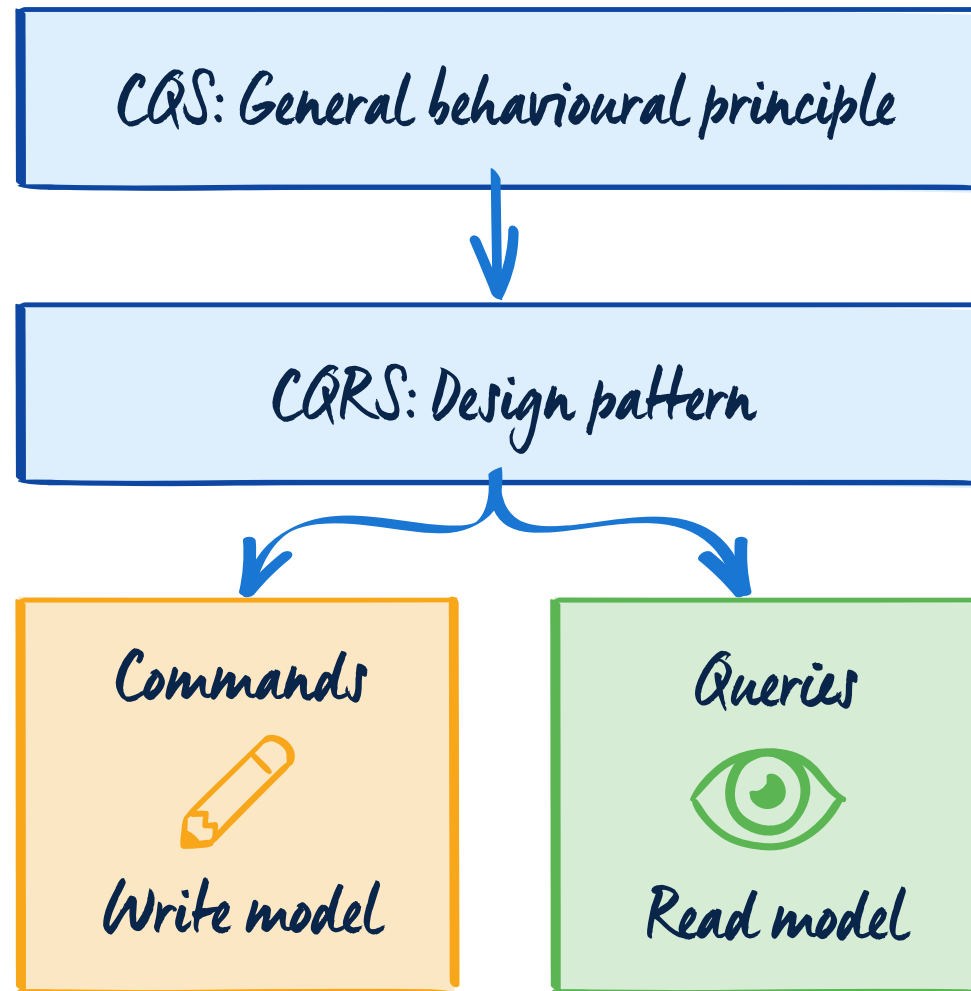


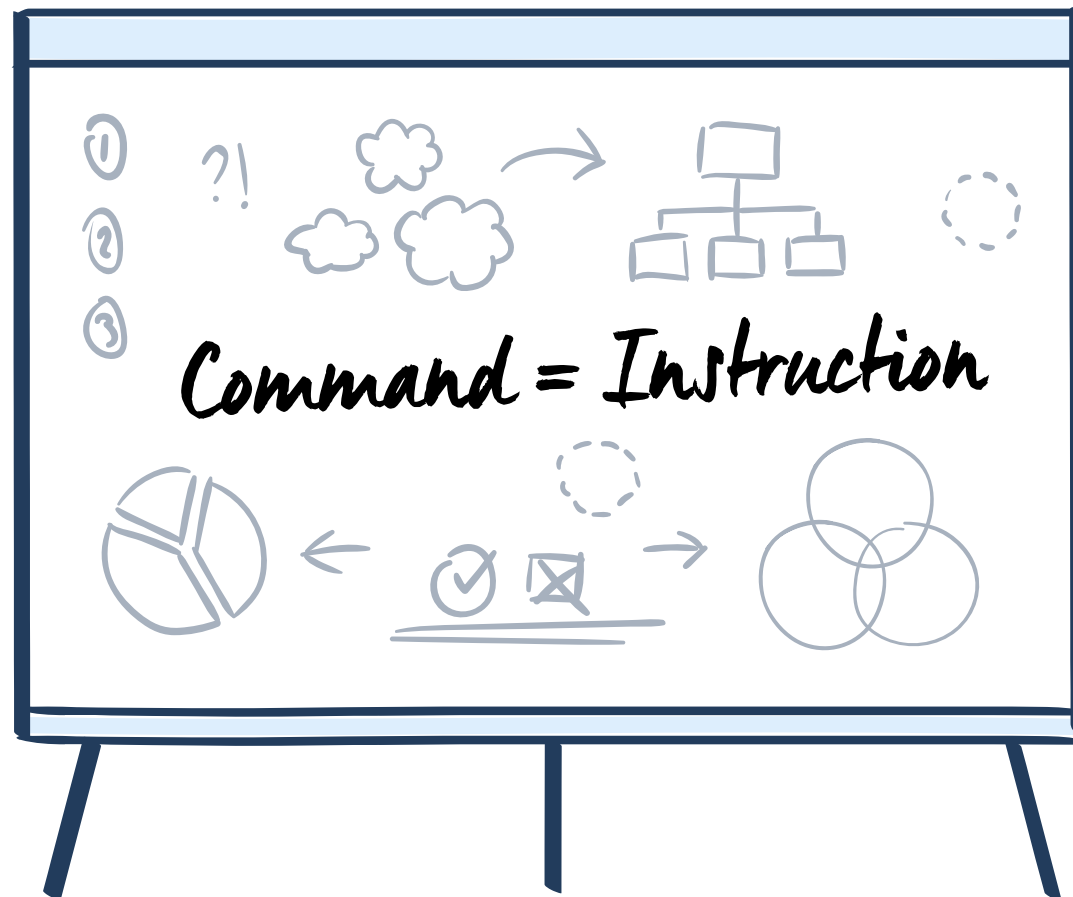


Domain-level change scope
(Triangle represents where change occurs)

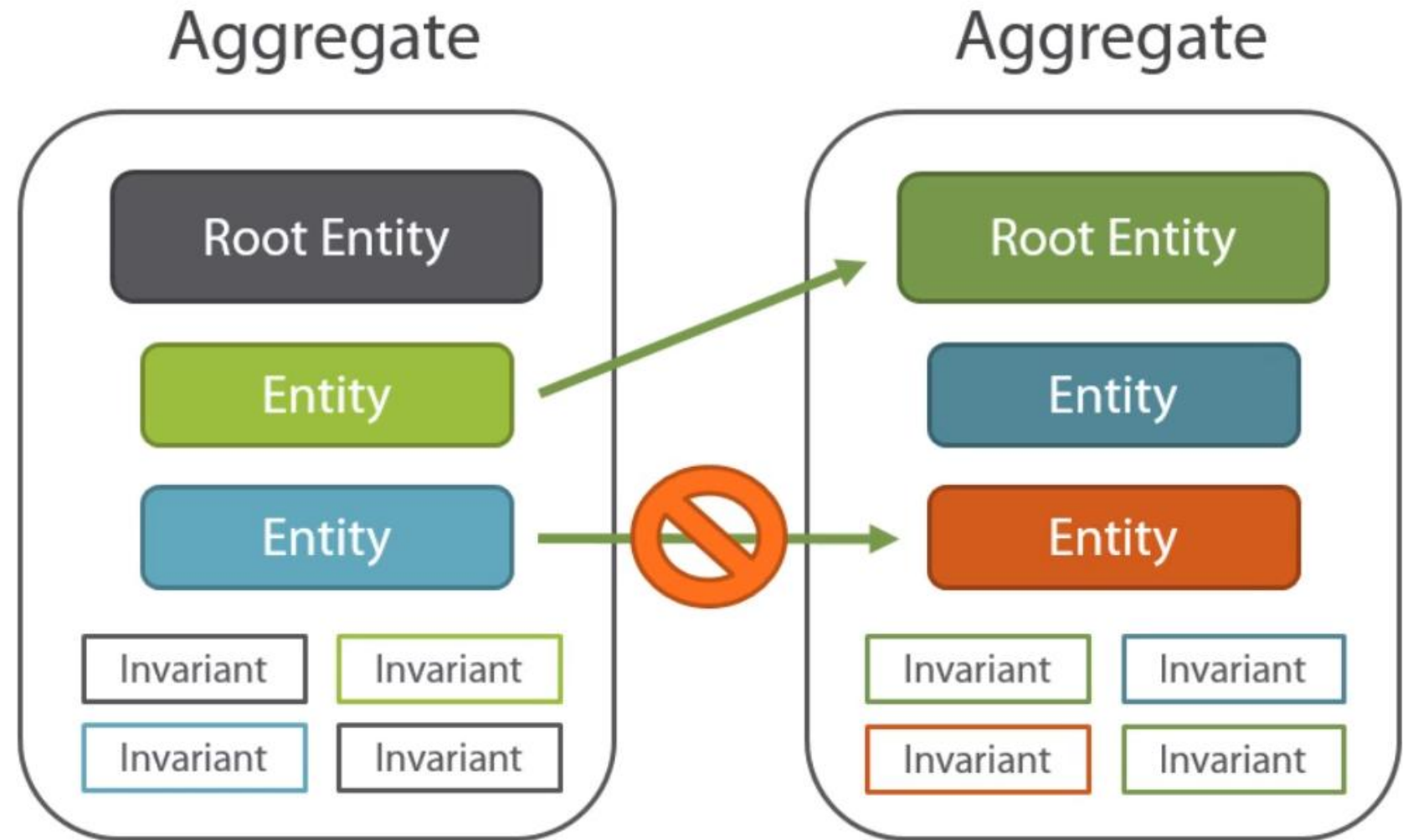


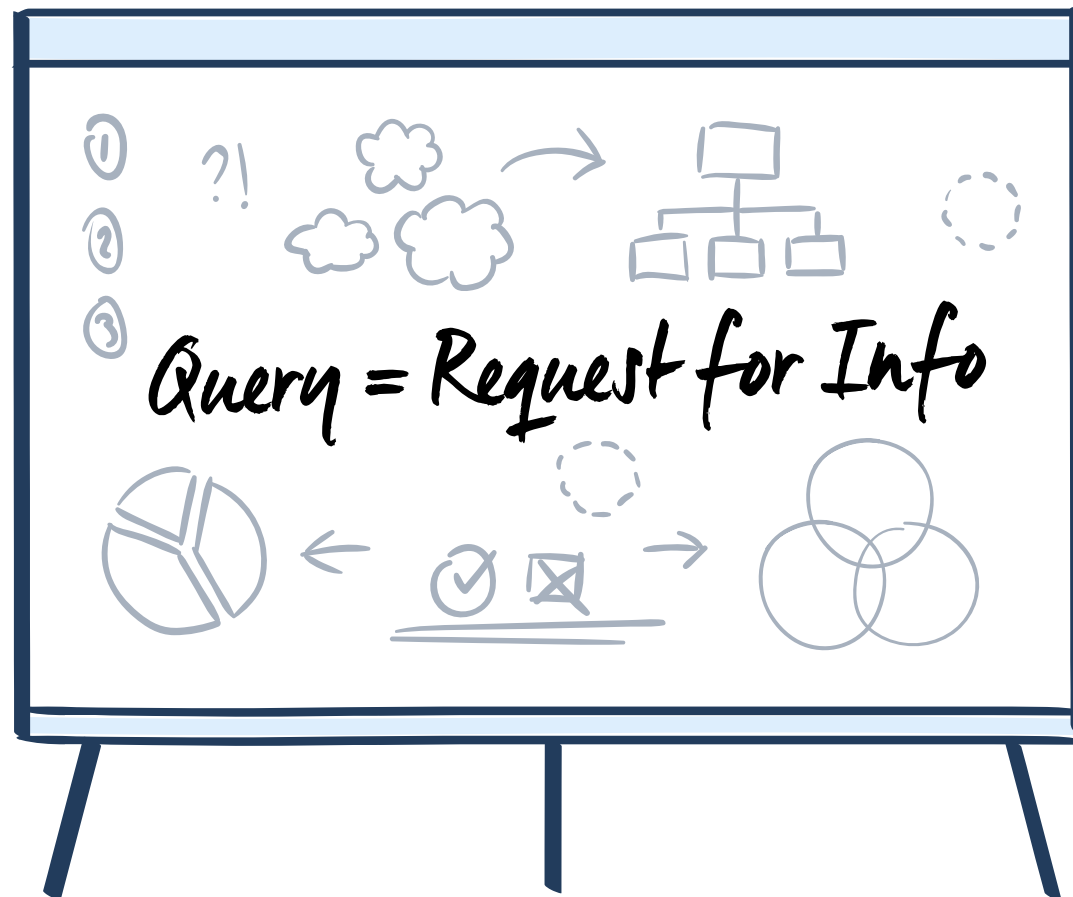


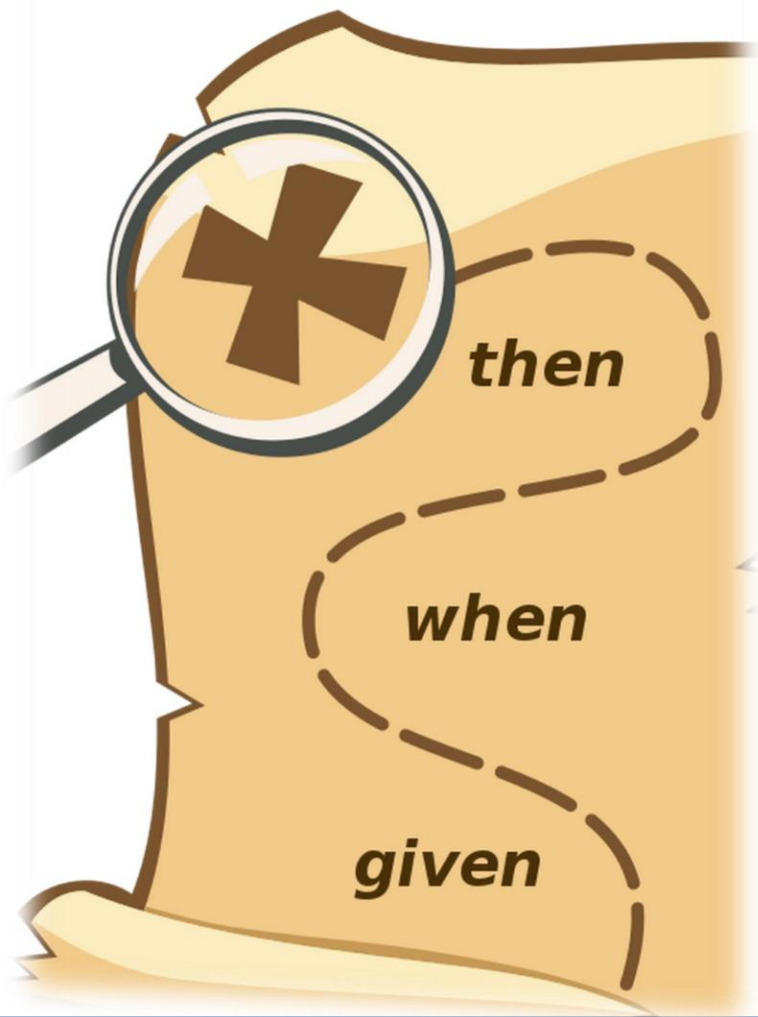




Aggregate

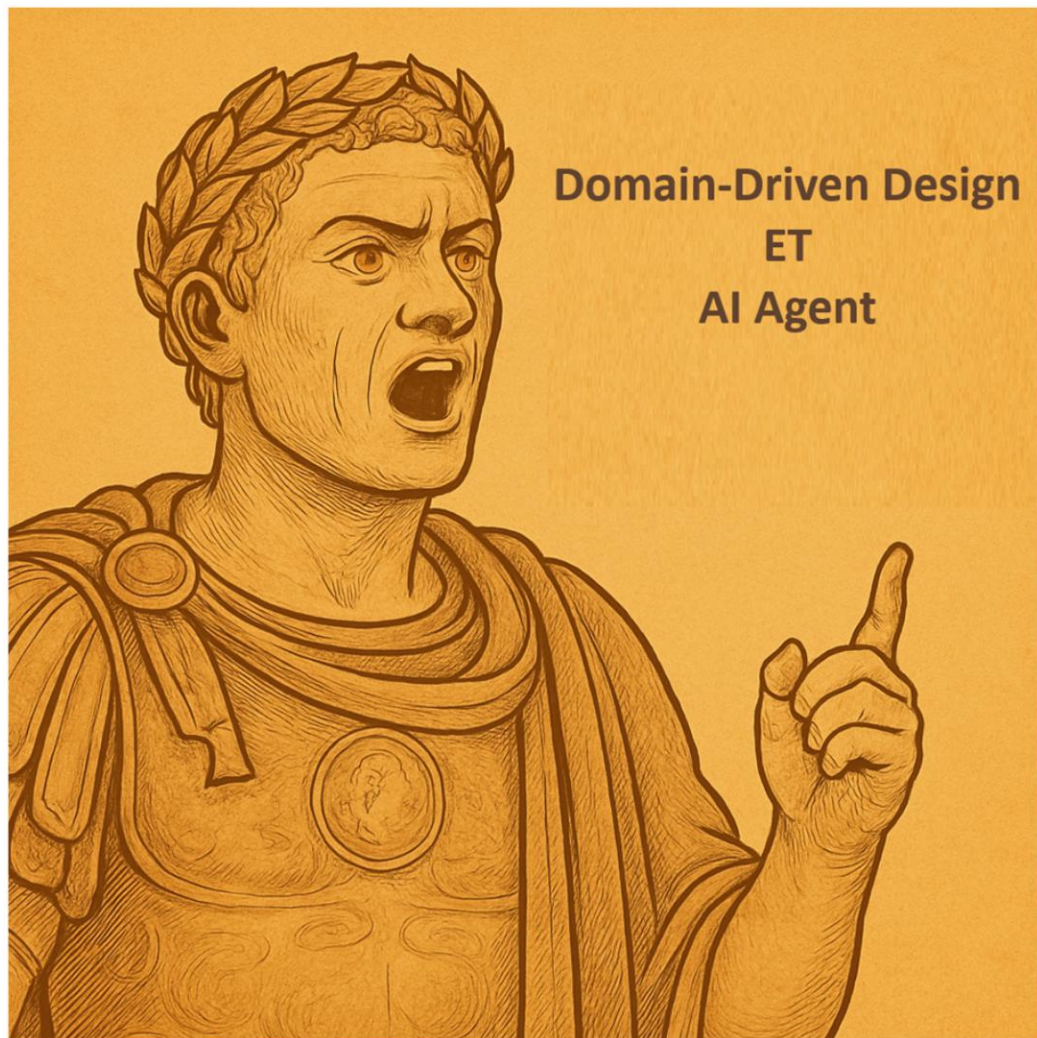






- L'idea fondamentale è quella di divider uno scenario in tre sezioni:
- **Given:** descrive lo stato dell'aggregato prima dell'invio del comando. Prerequisito.
- **When:** rappresenta il comando da inviare all'aggregato.
- **Then/Expect:** descrive i cambiamenti di stato previsti come risultato del comando.





Partendo dal contesto
#file:rest-api.prompt.md
implementa una solution
.NET che ne rispetti i requisiti
nella cartella src.

Not THE solution, but A solution

Disclaimer

Ho condiviso ciò che funzionato per me; non è un approccio universale.

Our Context

Le soluzioni si devono adattare alle dimensioni del Team, allo stack tecnologico...

Your Mileage

Prendete I principi, adattate l'implementazione alle vostre esigenze.







Alberto Acerbis



alberto.acerbis@intre.it



<https://github.com/BrewUp/DDD-Europe-2025>



Use **DDD20** for a 20% discount

