



CQRS és Mediátor tervezési minták használata ASP.NET Core-ral

Tóth Tibor
AutSoft Zrt., BME-VIK-AUT
toth.tibor@autsoft.hu

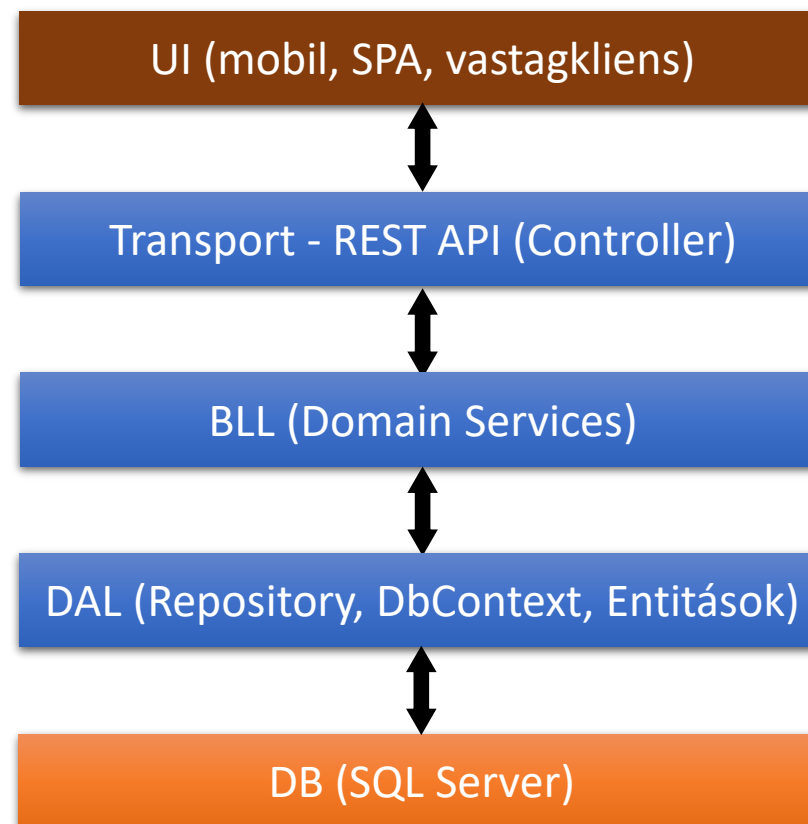
 @Tibcsi_
 @tibitoth



Minták

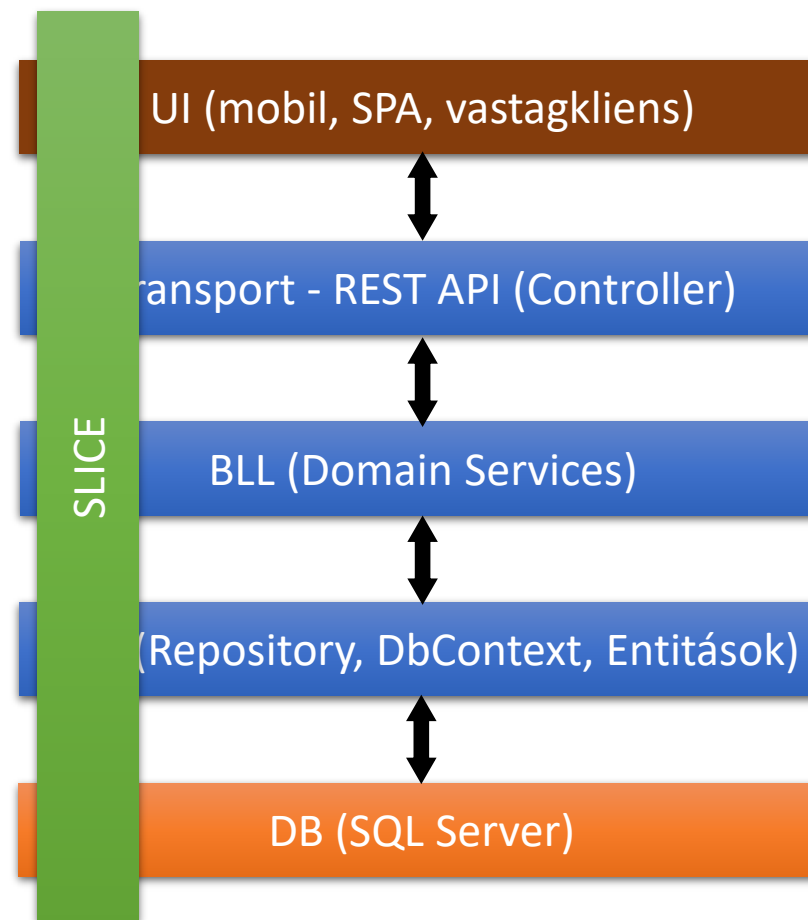
Többrétegű architektúra

- Klasszikus, már egyfajta minimum
- Rétegek között jó felelősség szétválasztás
- Rétegen belül nem strukturált (alapból)
- Fejlesztés logikáját kevésbé követi
 - Feature/bugfix fejlesztés vertikális



Vertical slices

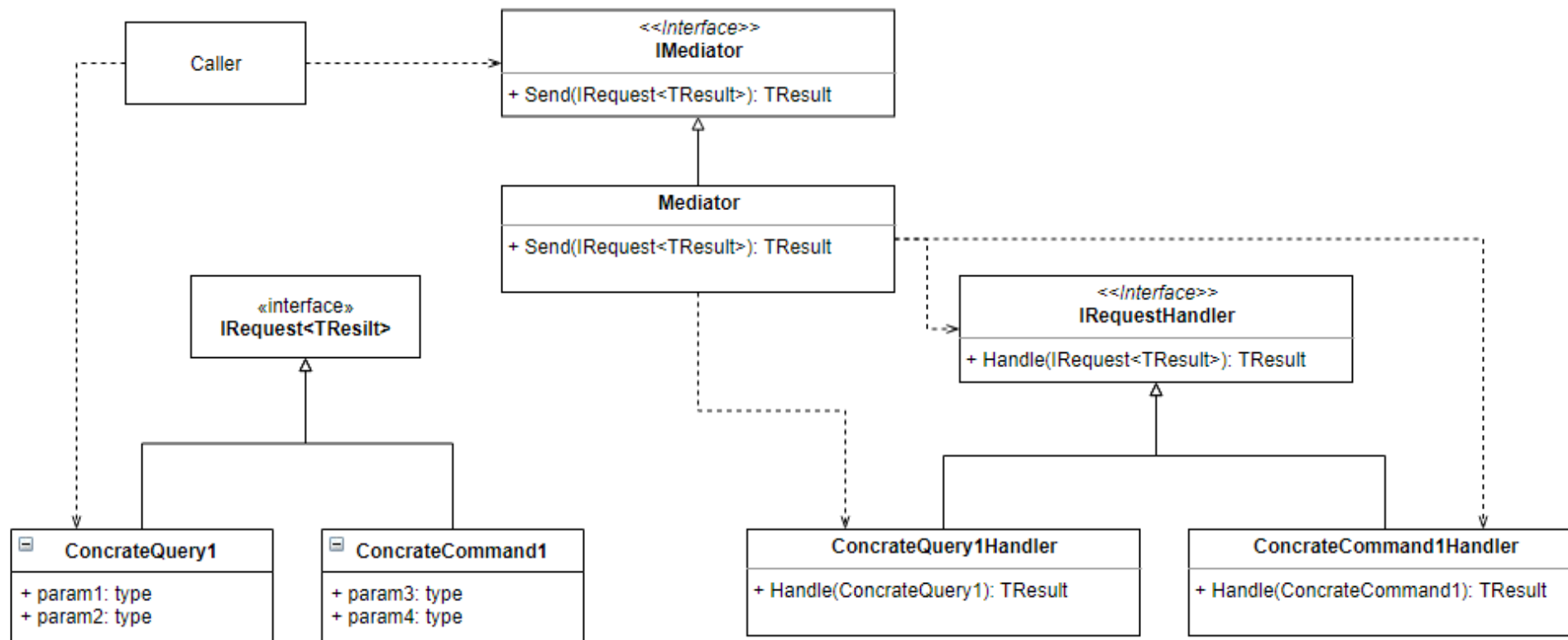
- Vertikális logikai csoportosítás funkciók szerint
- Egyszerűbb megtalálható kód funkció szerint



CQRS

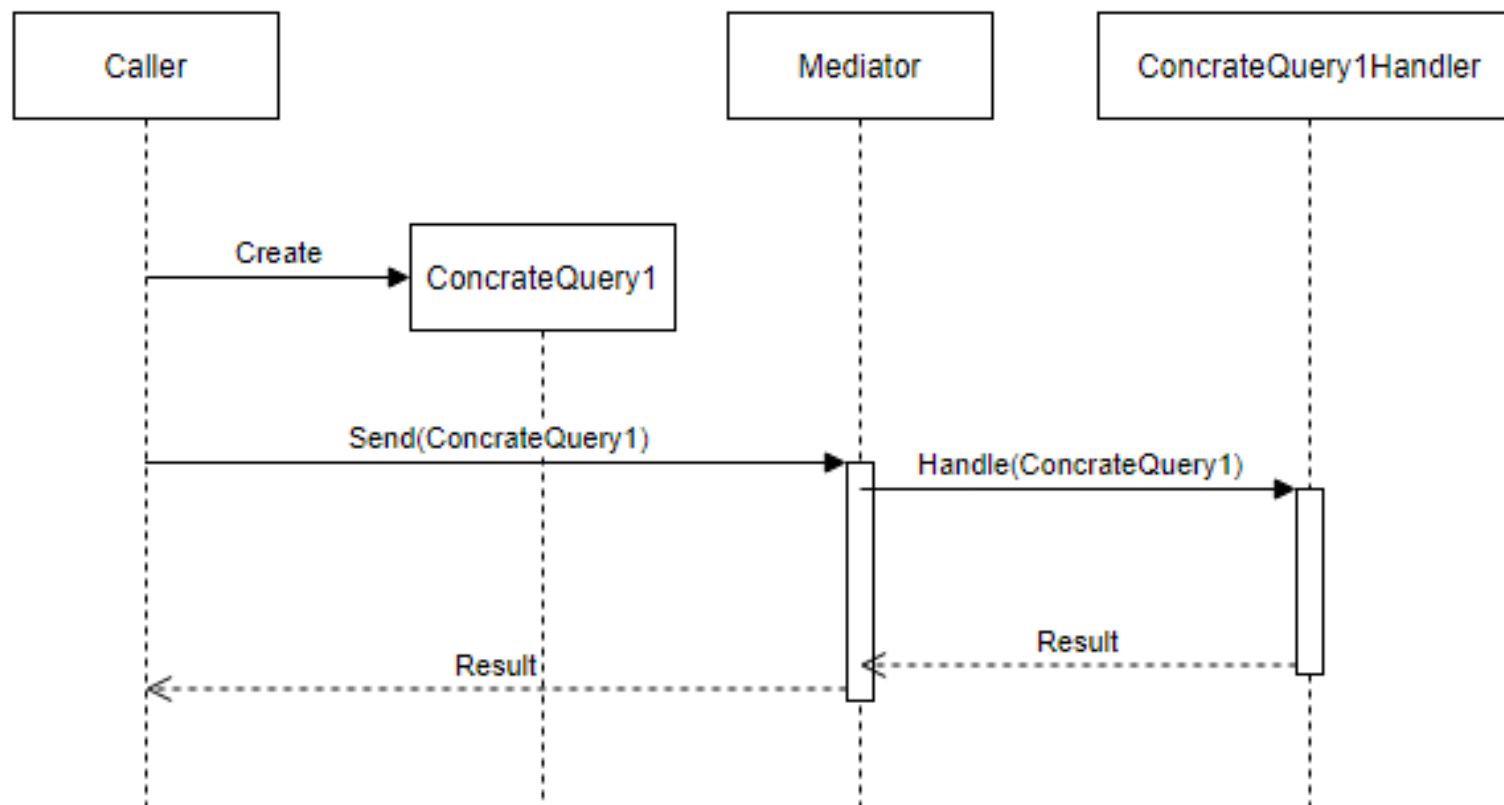
- Command Query Responsibility Segregation
- CRUD mentális modell új aspektusú felosztása
 - Módosító (Command) és lekérdező (Query) műveletek szétválasztása
- DDD-ben a műveletek egyik tervezés módszere
- Vertical Slices gondolattal egybevág
 - Feature
 - Command
 - Query
 - stb

Mediátor



- Eseménykezelésért felelős minta
- Központi objektum, ami elosztja a parancsokat a cél objektumokhoz

Mediátor





MediatR

MediatR

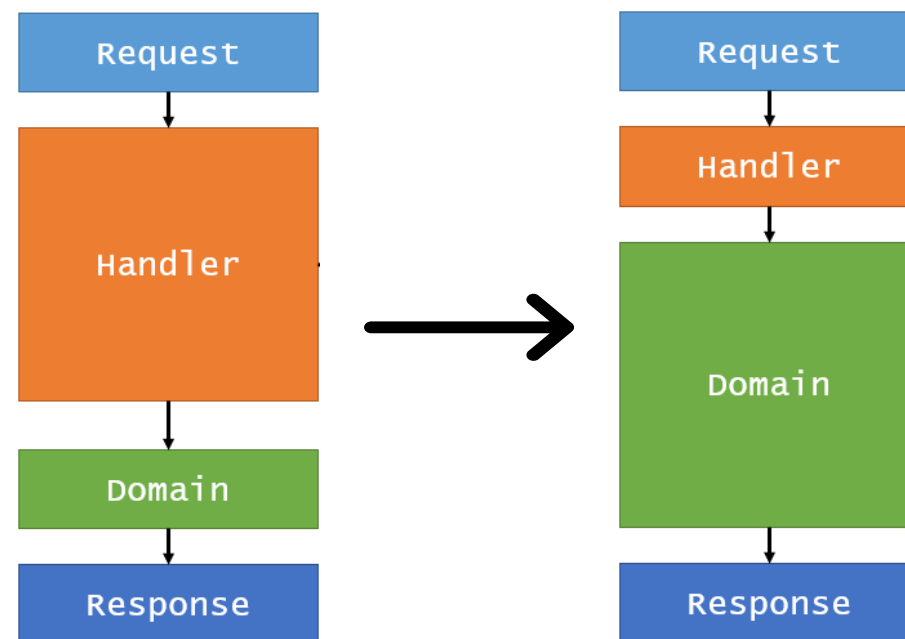
- Jimmy Bogard ~ 2015
- .NET környezetben az egyik leggyakoribb CQRS + Mediátor keretrendszer
- DI támogatás
- Request pipeline
- Event Collaboration (Domain Event) támogatás
 - Notification mechanizmuson keresztül



Red Green Refactor

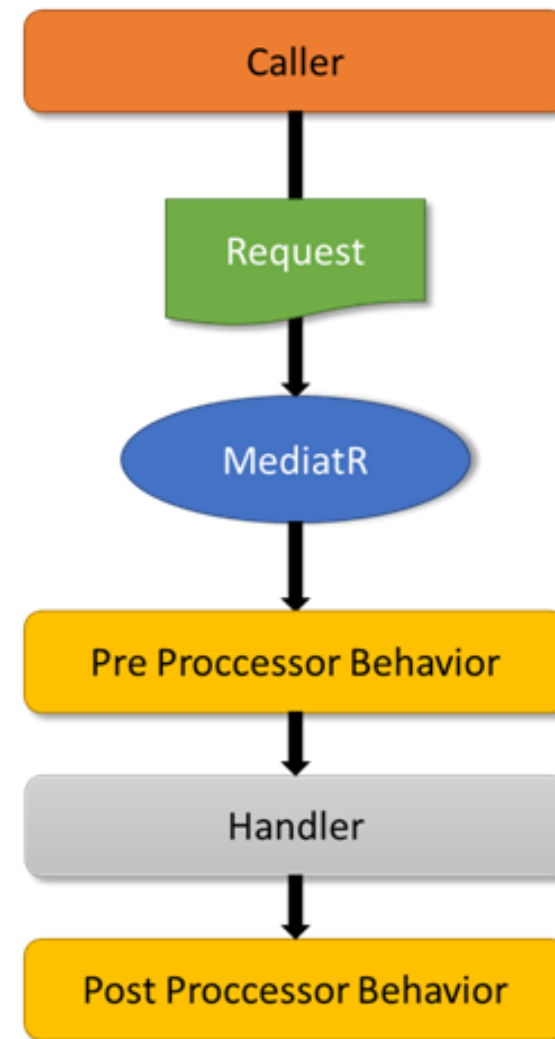
Handler mechanizmus

- Művelet leíró osztály (Command, Query)
 - `IRequest<TResult>`
- Művelet elvégző osztály (Handler)
 - `IRequestHandler< IRequest<TResult>, TResult >`
- Direkt üzenet küldés
 - `IMediator.Send`
- Egy Requesthez egy handler
 - Ha nincs, vagy több is van, akkor futási idejű hiba
- Üzleti logika Handlerben vagy alacsonyabb BLL / Domain rétegben (DDD)



Request pipeline

- ~Decorator tervezési minta
 - `IPipelineBehavior<IRequest<TResult>, TResult>`
 - `IRequest[Pre | Post]Processor`
 - `IRequestExceptionHandler`
- Hasonlóan az MVC filterekhez, vagy ASP.NET Core middleware pipelinehoz
- BLL szintű pipeline
- Sok mindenre használható, pl.:
 - Exception handling
 - Logolás
 - Validálás
 - Retry, Cache
 - Tranzakciókezelés
 - Stb.



Notifications

- Domain események küldése és lekezelése, Event Collaboration (Domain Event)
 - BLL rétegek ne közvetlenül egymást hívják meg
 - Pl.: Megrendelés „modul” módosítja a készletet vs. OrderCreatedEvent és majd a Készlet „modul” lekezeli
 - Mikroszolgáltatásoknál gyakori, de monlit appon belül is hasznos
- Broadcast üzenetküldés
 - INotification
 - INotificationHandler<INotification>
- Az üzenet küldés sikeres lesz, akkor is ha 0 handler van

Tippek

- RequestHandler ne indítson másik Request-et
 - Ajánlás, de indokolt esetben mégis hasznos lehet
 - Nem kell, hogy minden egy request legyen
- IMediator : ISender, IPublisher
 - Új verzióban már szeparált
- Használjuk az IRequestHandler<,> interfészt
 - Specializált változatai inkább haszontalanok és növelik a komplexitást
- Tesztelhetjük, hogy van-e handler!
- Pattern matching DI konténerrel .NET 5-től
- Kliens alkalmazásoknál: MediatR.Courier
 - Dinamikus fel- és leiratkozás értesítésekre

MediatR DEMÓ

<https://github.com/tibitoth/AspNetCore-CQRS-MediatR-webinarium>

Köszönöm a figyelmet! Kérdések?

Tóth Tibor

AutSoft Zrt., BME-VIK-AUT

toth.tibor@autsoft.hu

 @Tibcsi_

 @tibitoth