

Event Sourcing & CQRS

NPRG024 – Návrhové vzory

3. 4. 2021

Úvod

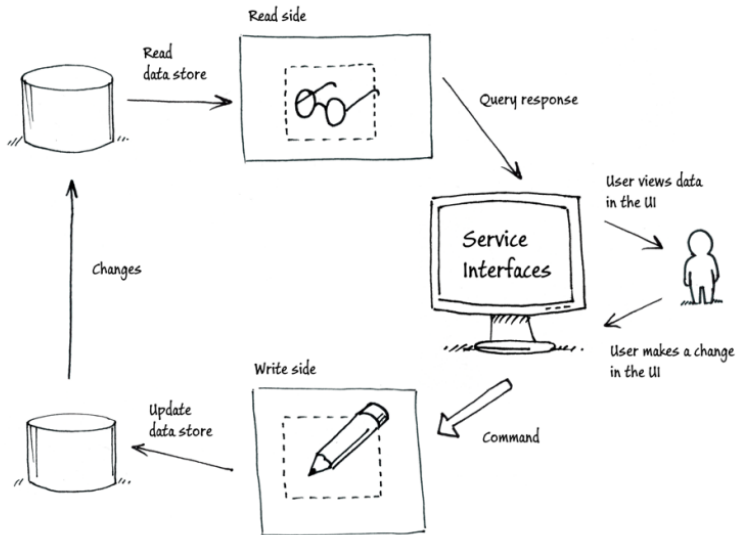
- ▶ Dva návrhové vzory, často se používají dohromady.
- ▶ Zabývají se čtením a zapisováním dat.
- ▶ Škálovatelnost, přizpůsobitelnost konkrétní doméně.
- ▶ Využití ve velkých cloudových aplikacích.
- ▶ Zdroje:
 - ▶ Cloud Design Patterns: Prescriptive Architecture Guidance for Cloud Applications
 - ▶ Exploring CQRS and Event Sourcing

CQRS – Problém

- ▶ Typicky jeden model (resp. jedna databáze) zajišťuje čtení i zápis.
 - ▶ DTO (Data transfer object), ORM (Object-relational mapping).
- ▶ Výhodou je snadná (často automatizovaná) implementace.
- ▶ Nevýhody:
 - ▶ Nepoměr mezi počtem čtení a zápisů – špatná škálovatelnost.
 - ▶ Optimalizace databáze na čtení (denormalizace) vs. zápis (3NF).
 - ▶ Záписы бývají výrazně složitější než čtení. Jeden modul jen stěží zvládne oboje dobře.

CQRS – Řešení

- ▶ Command and Query Responsibility Segregation.
- ▶ Různé modely pro čtení (*query*) a zápis (*command*).
- ▶ Další krok – oddělení databází.



CQRS – Výhody a nevýhody

- ▶ Škálovatelnost.
- ▶ Specializace a optimalizace na konkrétní činnost (Materialized View).
- ▶ Snížení složitosti operací.
- ▶ Náročné na implementaci.

CQRS – Souvislosti

- ▶ Není vhodné použití na celý systém jako celek.
- ▶ Důležité je identifikovat části (*bounded context*), pro které se vyplatí.
 - ▶ Složitost, široká spolupráce, často se měnící doménová logika ...
- ▶ Domain-driven design – podobné důvody použití.
- ▶ Data Partitioning – jak rozdělit data na části.
- ▶ Materialized View.

Event Sourcing – Problém

- ▶ Klasický přístup k datům (například CRUD) vychází z udržování aktuálního stavu systému, což má svá omezení.
- ▶ Transakce limitují škálovatelnost a výkon.
- ▶ Uživatelé, upravující najednou stejnou entitu, se navzájem blokují.
- ▶ Není k dispozici historie úprav.
- ▶ Příklad – prodej zboží těsně po zahájení.

Event Sourcing – Řešení

- ▶ Každý zápis je samostatný event.
- ▶ Eventy jsou velmi jednoduché a immutable – vysoká propustnost.
- ▶ Databáze je append-only posloupnost eventů.
- ▶ Současný stav lze zjistit přehráním všech eventů.
- ▶ Optimalizace – snapshoty, databáze pro čtení.
- ▶ Příklad – bankovní transakce, BTC.

Event Sourcing – Výhody a nevýhody

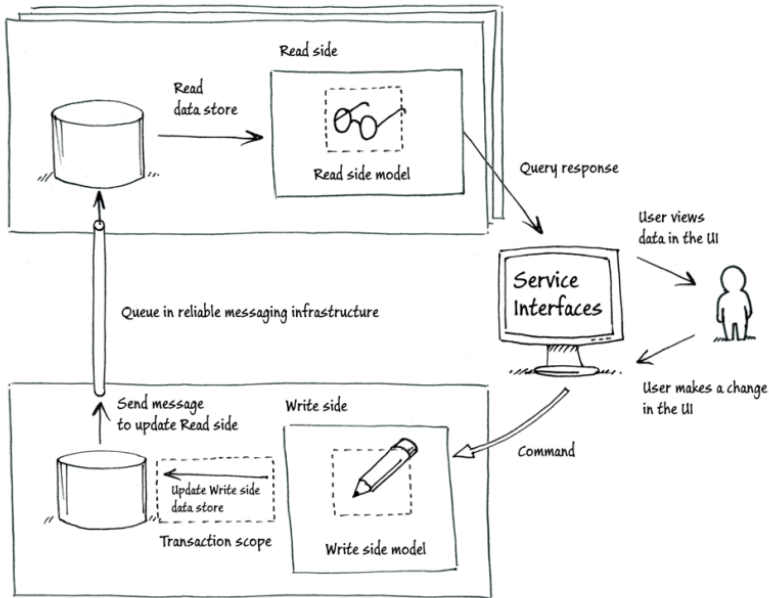
- ▶ Kompletní historie.
- ▶ Jednoznačnost operací.
- ▶ Škálovatelnost, výkon.
- ▶ Pouze eventuální konzistence.
- ▶ Zpětná kompatibilita eventů.

Event Sourcing – Souvislosti

- ▶ Podobně jako u CQRS se hodí jen v některých případech.
 - ▶ Historie, řešení konfliktů, oddělení zpracování událostí od jejich zaznamenání, ...
- ▶ Nevhodné pro real-time aplikace.
- ▶ Compensating Transaction – místo úpravy eventu vytvoříme nový event.
- ▶ Data Partitioning, Materialized View – jako u CQRS.

Event Sourcing & CQRS

- ▶ Velmi přirozená kombinace.
- ▶ Event Sourcing bez CQRS je možný, ale jen málokdy dává smysl.



Event Sourcing & CQRS – Čtení a zápis

- ▶ Zápis – přidávání immutable eventů do append-only databáze.
- ▶ Po uložení jsou hned odesílány odběratelům.
 - ▶ Těch může být i více, ale také nemusí existovat žádný.
- ▶ Čtení – v podstatě denormalizovaná cache zachycující aktuální stav, upravovaný podle příchozích eventů.
- ▶ Možná obnova z historie.