

CQRS (Command Query Responsibility Segregation)

CQRS (Command Query Responsibility Segregation) - це архітектурний підхід, який полягає в розділенні команд та запитів на окремі моделі (тобто розділення коду який змінює стан і коду просто читає данні). Це дозволяє покращити масштабованість додатку, покращити зручність розробки, оскільки дозволяє розділити команди, що спрощує розробку та підтримку додатку. Також дозволяє покращити швидкість та продуктивність додатку, оскільки дозволяє оптимізувати кожен тип операції окремо.

В базі підходу лежить принцип Command-query separation (CQS). **Головною ідеєю** є те що методи об'єкта можуть бути двох типів:

- **Queries:** Методи які просто повернають результат не змінюючи стан. Повертає DTO.
- **Commands:** Методи які змінюють стан не повертаючи результат.

Перевагами такого підходу є:

- **Швидкість:** Можливість оптимізувати окремо запити та команди.
- **Масштабованість:** Можливість масштабувати окремо запити та команди.
- **Розширюваність:** Можливість додавати нові команди та запити без зміни існуючого коду.
- **Тестування:** Можливість тестувати окремо команди та запити.

Під масштабованістю маємо на увазі, наприклад нас сервіс стає популярним і ми маємо багато запитів, і нам треба додати нові потужності (хочемо додати ще один сервер). Класична [багатошарова архітектура](#) не дозволяє нам так просто це зробити. Але якщо ми використовуємо CQRS, ми можемо просто додати новий сервер для запитів, і все буде працювати. Або якщо навпаки в нас буде велика кількість операцій на запис ми також можемо распаралелити ці операції.

CQRS може бути складним для розуміння та реалізації, особливо для початківців. Це може призвести до збільшення часу розробки та вартості проекту. Крім того, використання CQRS може призвести до збільшення складності коду, оскільки він вимагає розділення команд та запитів. Це може зробити код менш зрозумілим та важким для підтримки.

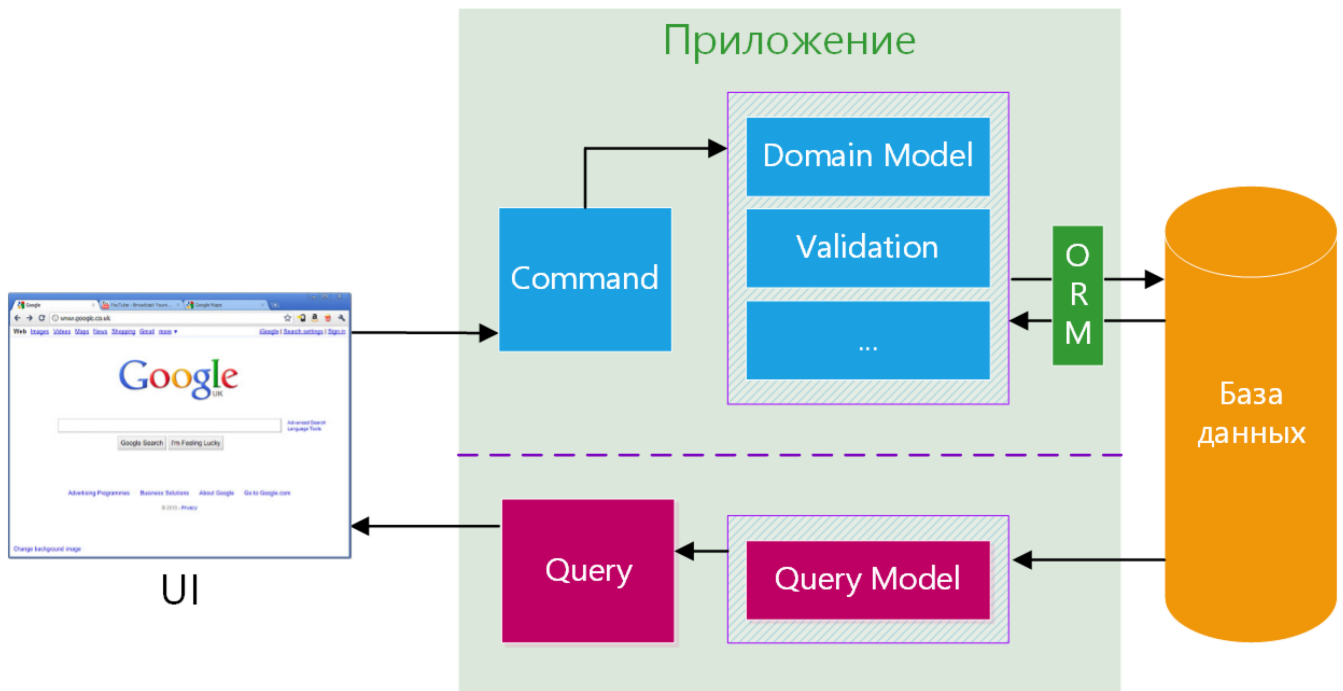


Figure 1. Схема архитектуры CQRS