Commands and Queries

Matthew Renze

SOFTWARE CONSULTANT

@matthewrenze   www.matthewrenze.com

Overview

Commands and Queries

CQRS Architectures

Pros and Cons

Demo

Command-Query Separation

Command

Does something

Should modify state

Should not return a value

Command-query Separation

Command	Query
Does something	Answers a question
Should modify state	Should not modify state
Should not return a value	Should return a value

Command-query Separation Exceptions

Pop a stack

Remove item (command)

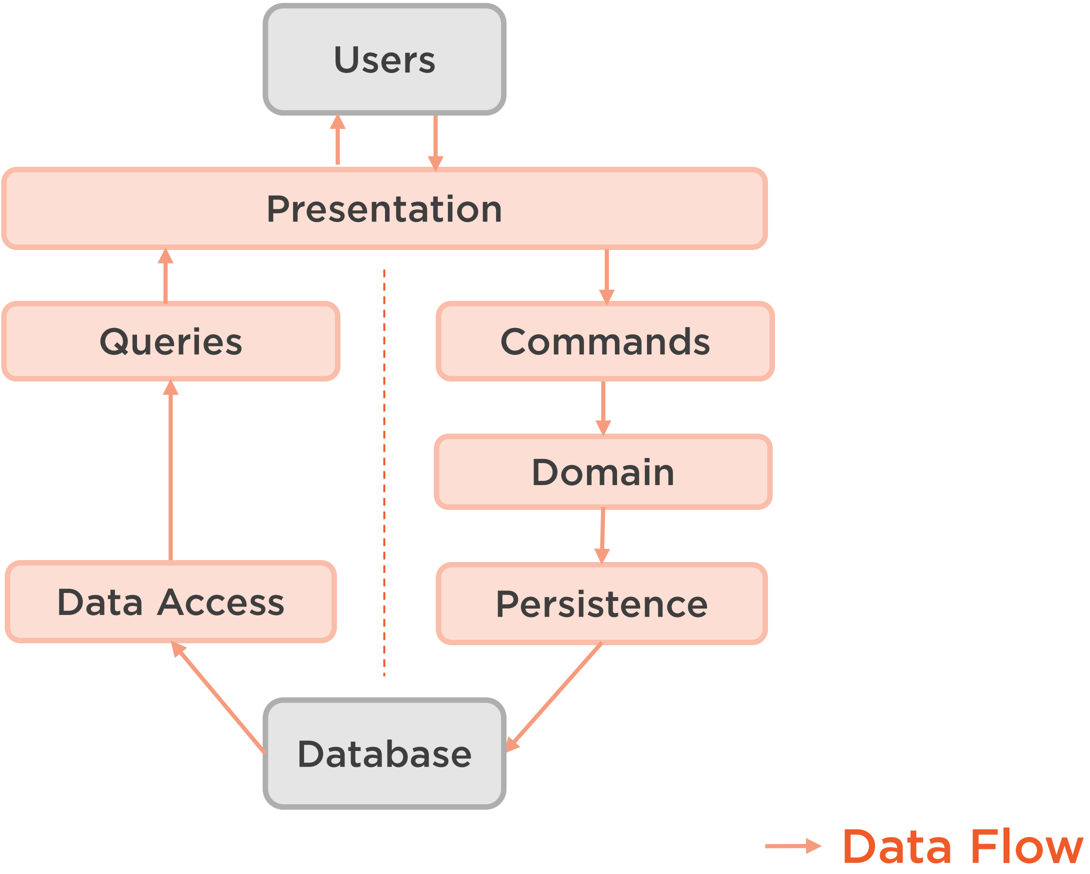
Return top item (query)

Create a new record

Create record (command)

Return new ID (query)

CQRS Architectures



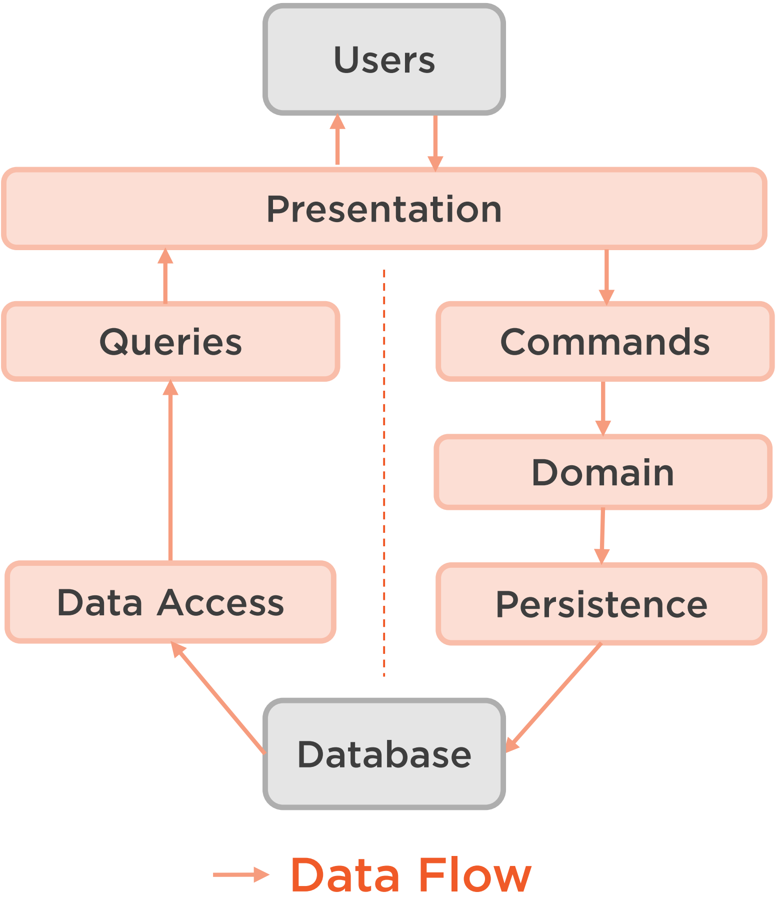
Single-database CQRS

Single database

Commands use domain

Queries use database

Simplest of the three



Two-database CQRS

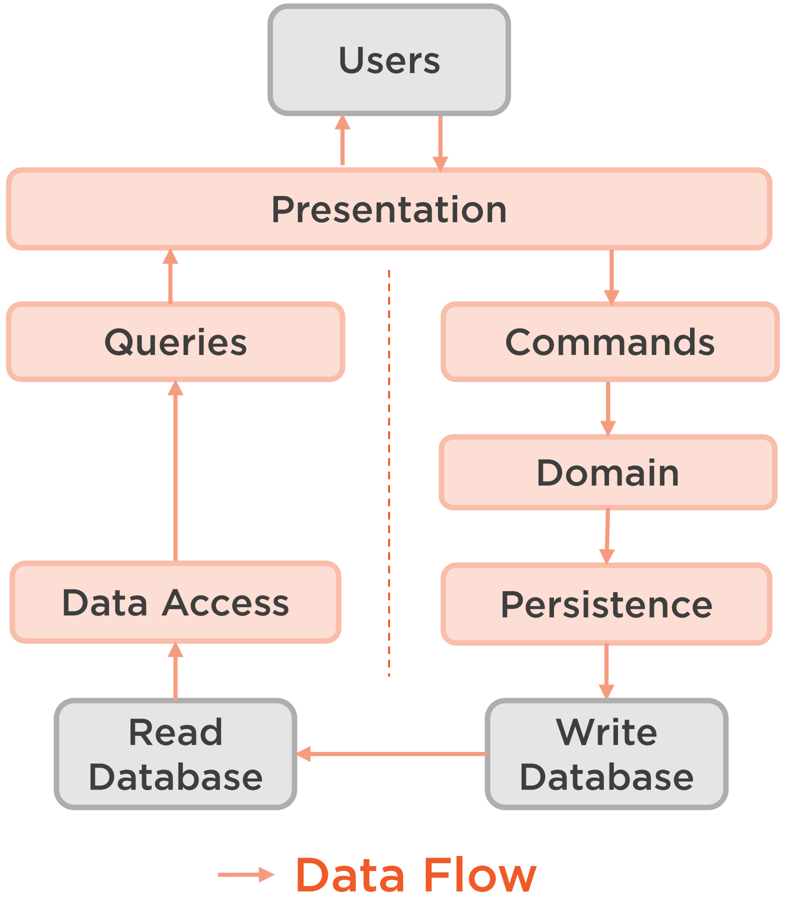
Read and write databases

Commands use write DB

Queries use read DB

Eventual consistency

Orders of magnitude faster



Event Sourcing CQRS

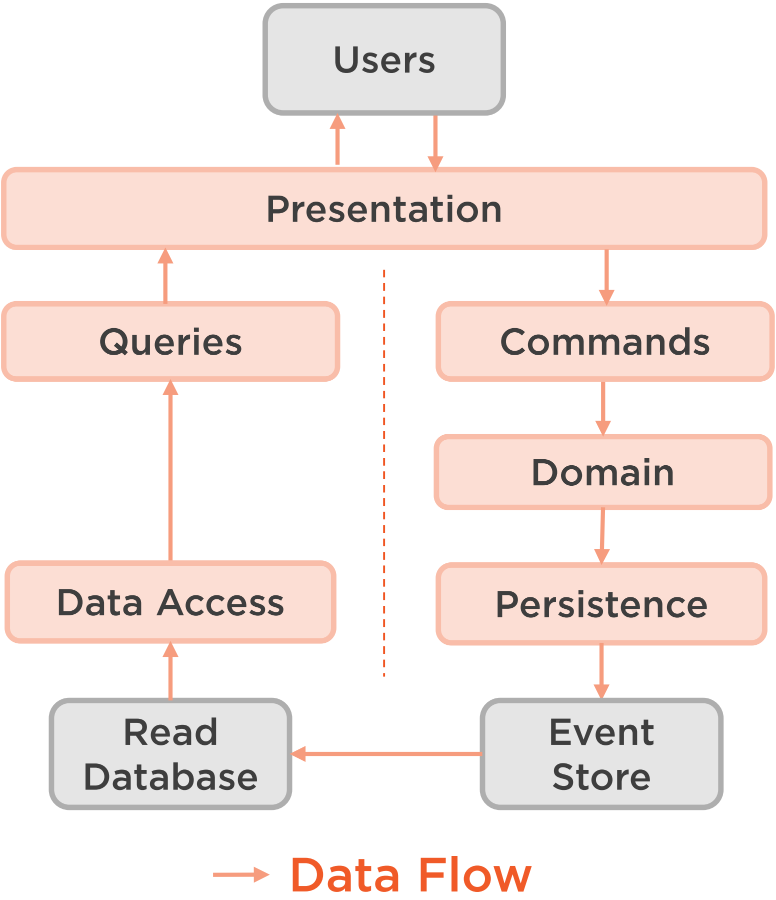
Store events

Replay events

Modify entity

Store new event

Update read database



Event Sourcing CQRS

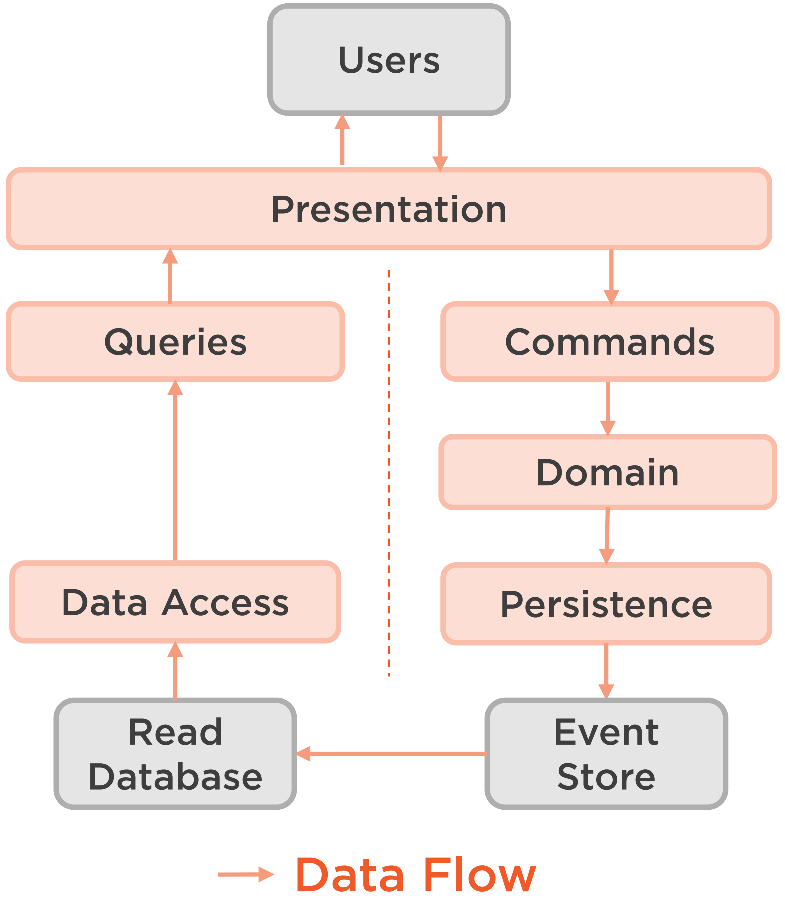
Complete audit trail

Point-in-time reconstruction

Replay events

Multiple read database

Rebuild production database



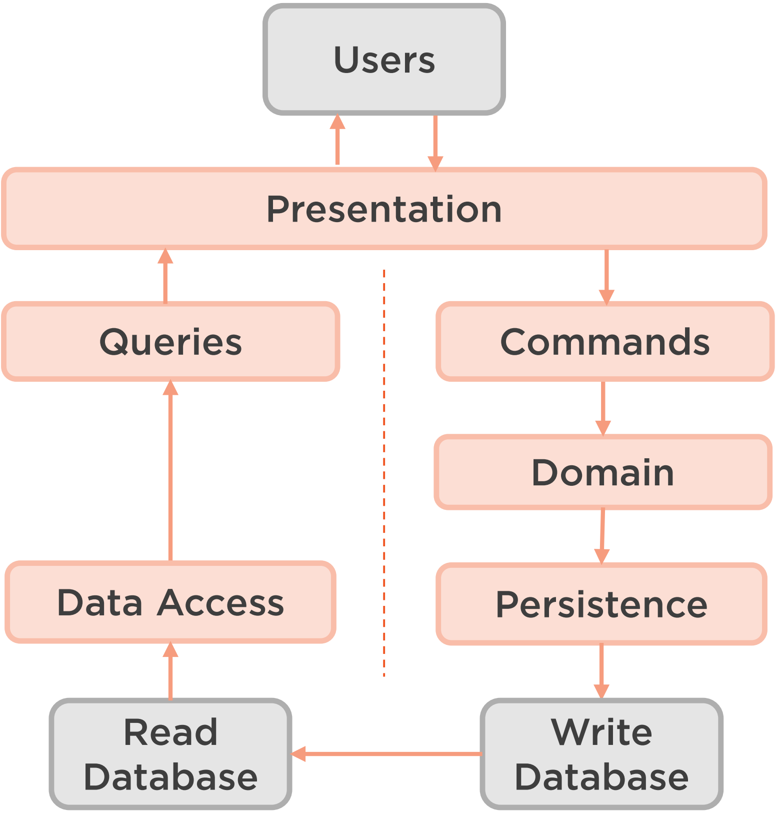
Why Use CQRS?

Pros

More efficient design

Optimized performance

Event sourcing benefits



→ Data Flow

Why Use CQRS?

Pros

More efficient design

Optimized performance

Event sourcing benefits

Cons

Inconsistent across stacks

More complex

Event sourcing costs

Set Up

Show SalesController Class

Show IGetSalesListQuery Interface

Show SalesListItemModel

Show GetSalesListQuery Class

Show Unprojected Lazy Query

Show Unprojected Lazy Query

Show Projected Query

Show GetSaleDetailsQuery

Wrap Up

Summary

Commands and Queries

CQRS Architectures

Pros and Cons

Demo