# Building an Event Sourced System with Event Modeling and CQRS

•••

alexgemmell

agemmell

PATIENTPOP

no.fixed.abode.life

sudo_adventure_puppy

# Overview

★ Event Modeling

★ CQRS

★ Event Sourcing

# Event Modeling

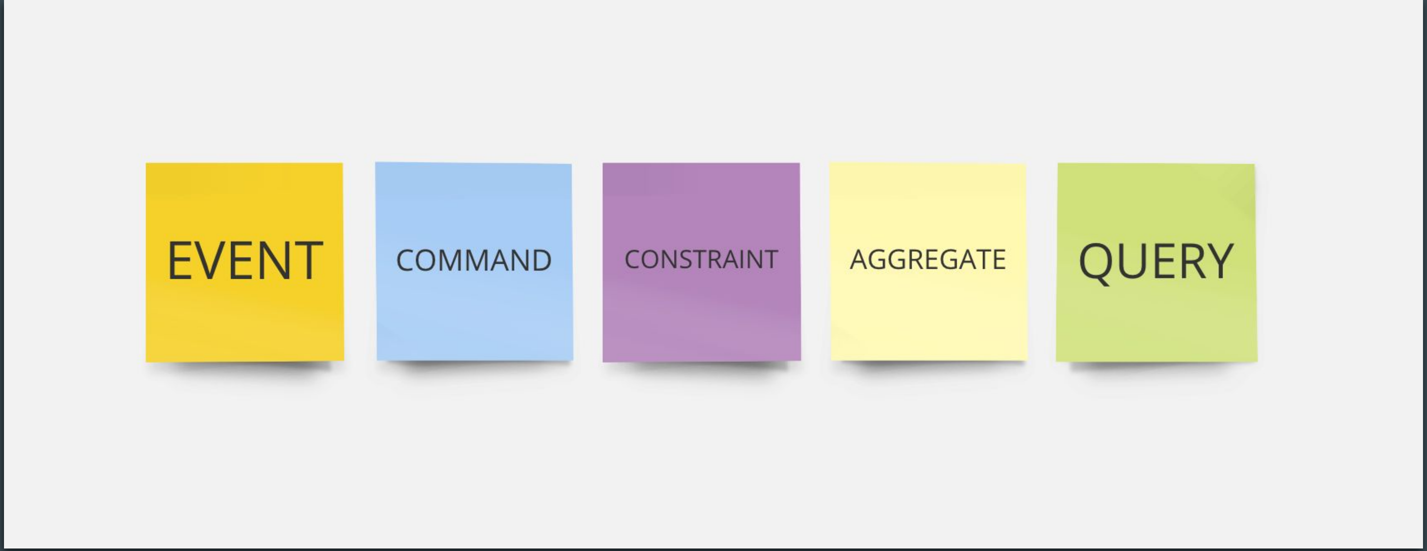
AKA Event Storming

Tool to gain shared understanding if the problem space and/or solution space.

Flavours:

- Big Picture (problem space)
- **Process Modeling (solution space)**
- Software Modeling (solution space)

# Process Modeling with stickies



# Event Sourcing

- Events are your source of truth `UserWasRegistered{lastname: “Gemmell”}`
- State is generated
- Event Store: an append-only database (past events are immutable)
- System Design:  Behaviour-first vs State-first

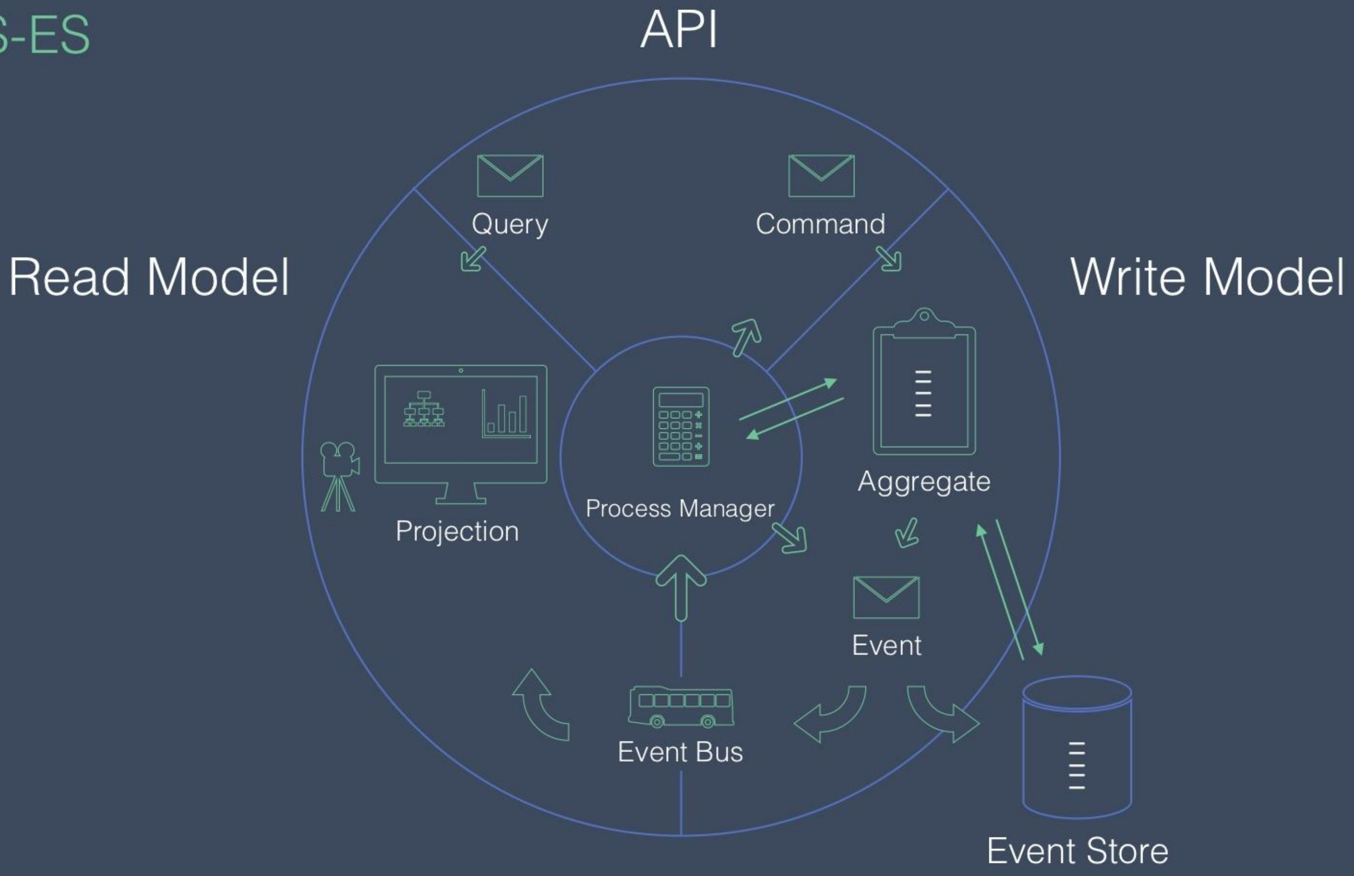
★ *How do you query an Event Store?*

E.g. `SELECT * FROM users WHERE lastname = “Gemmell”;`

# Command Query Responsibility Segregation (CQRS)

- Optimized Write Model
- Optimized Read Models
- Solves the Event Sourcing query problem with projections

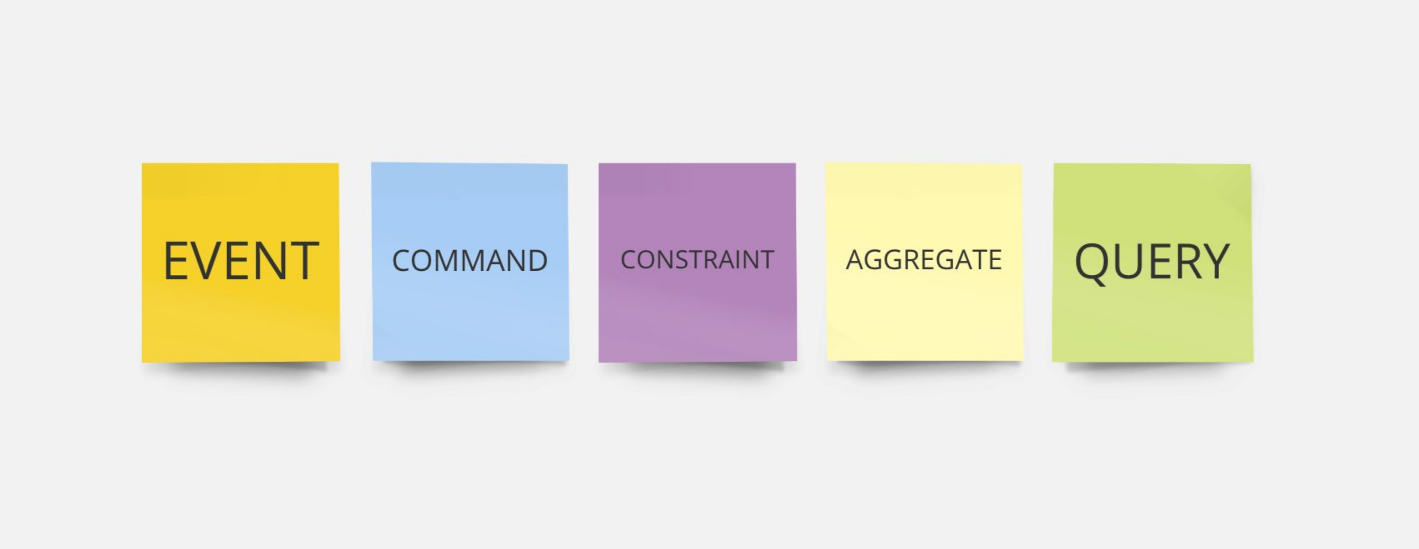
# CQRS-ES



*Source: Event Storming - From Sticky Notes to Code by Jamie Issacs*

# Interactive Time!

Practice Event Modeling



EVENT

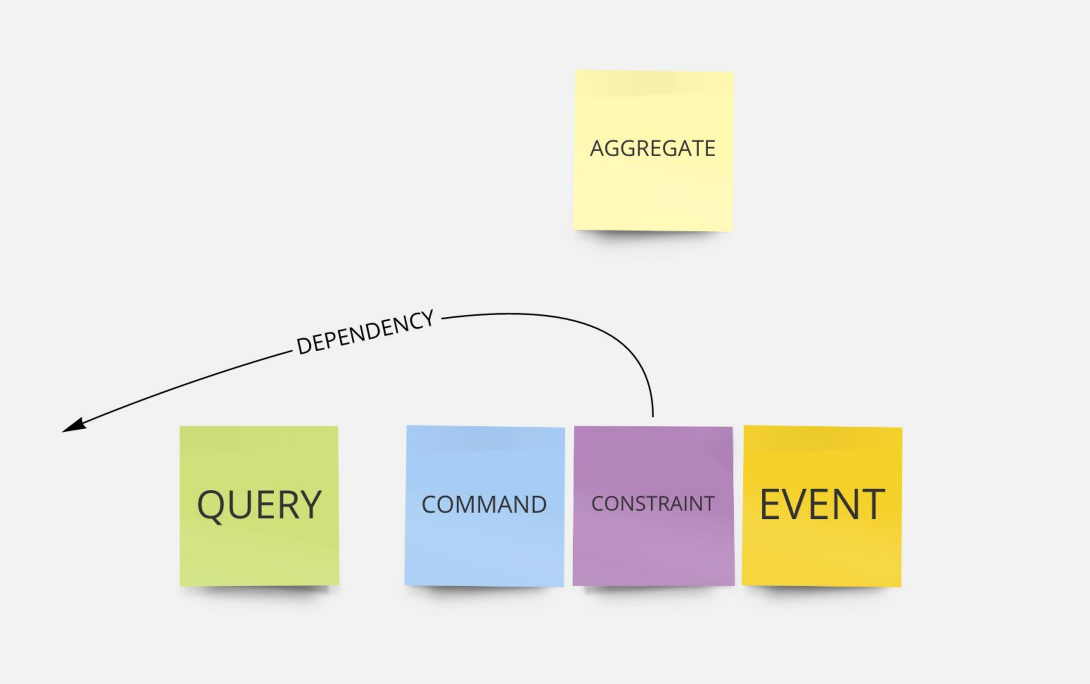
COMMAND

CONSTRAINT

AGGREGATE

QUERY

# Process Modeling



# Behavio(u)r Driven Development

Given that we model behaviour with Commands and Events

When we write tests

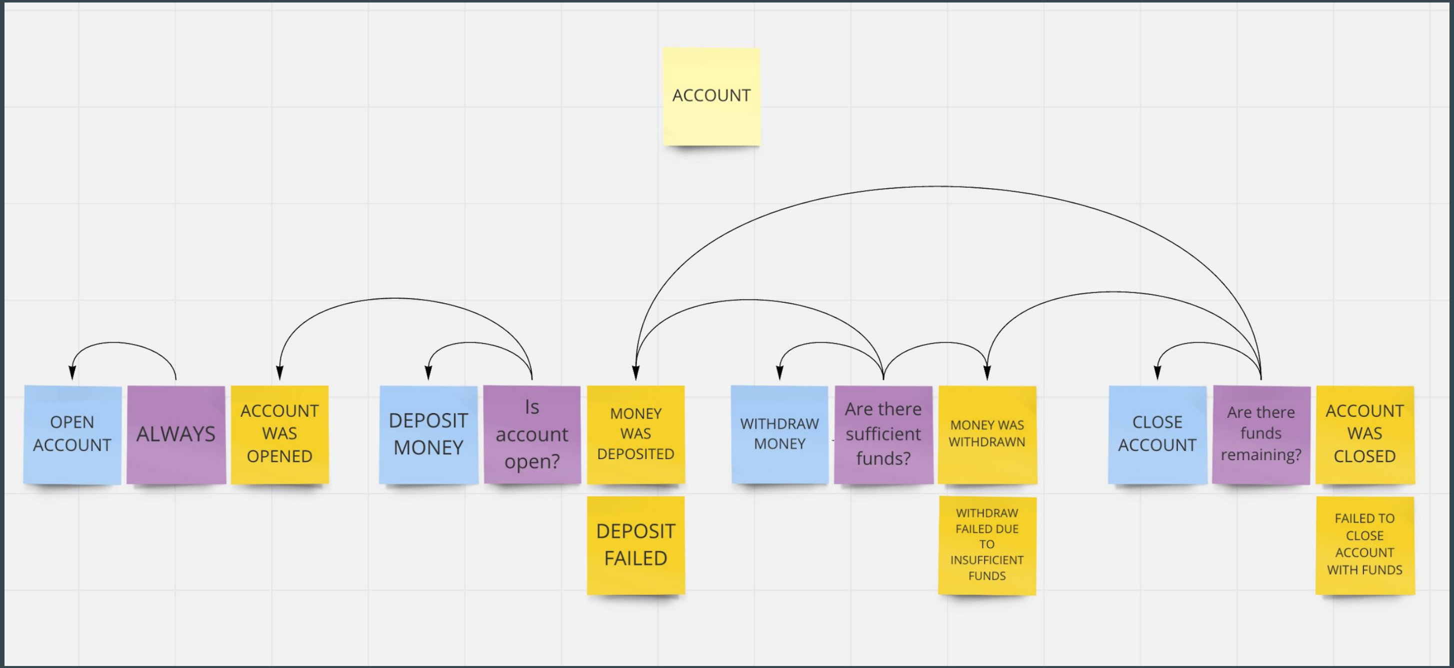
Then development is easy!

# BDD + TDD = Bestest Driven Development™

1. Model behaviour
2. Write your tests
3. Make the tests pass by implementing the code
4. …
5. Profit

# Interactive Time!

Bestest Driven Development™



# Wrap Up

Bestest Driven Development™

- Model Behaviour (problem space -> solution space)
- Involve experts (collaborate)
- CQRS + ES (solution architecture)
- Easy to write tests
- Easy to understand code
- Refactoring isn’t scary!

# What questions do you have for me?

# Thank you!

alexgemmell

agemmell