From EventStorming To EventStore

Passing through BDD

Alberto Acerbis

Connect the dots …

1	EventStorming
2	Event Sourcing
3	CQRS
4	Behavior-Driven Development

Disclaimer

- Event Sourcing and CQRS are orthogonal concepts.
- You can apply DDD patterns without implementing CQRS

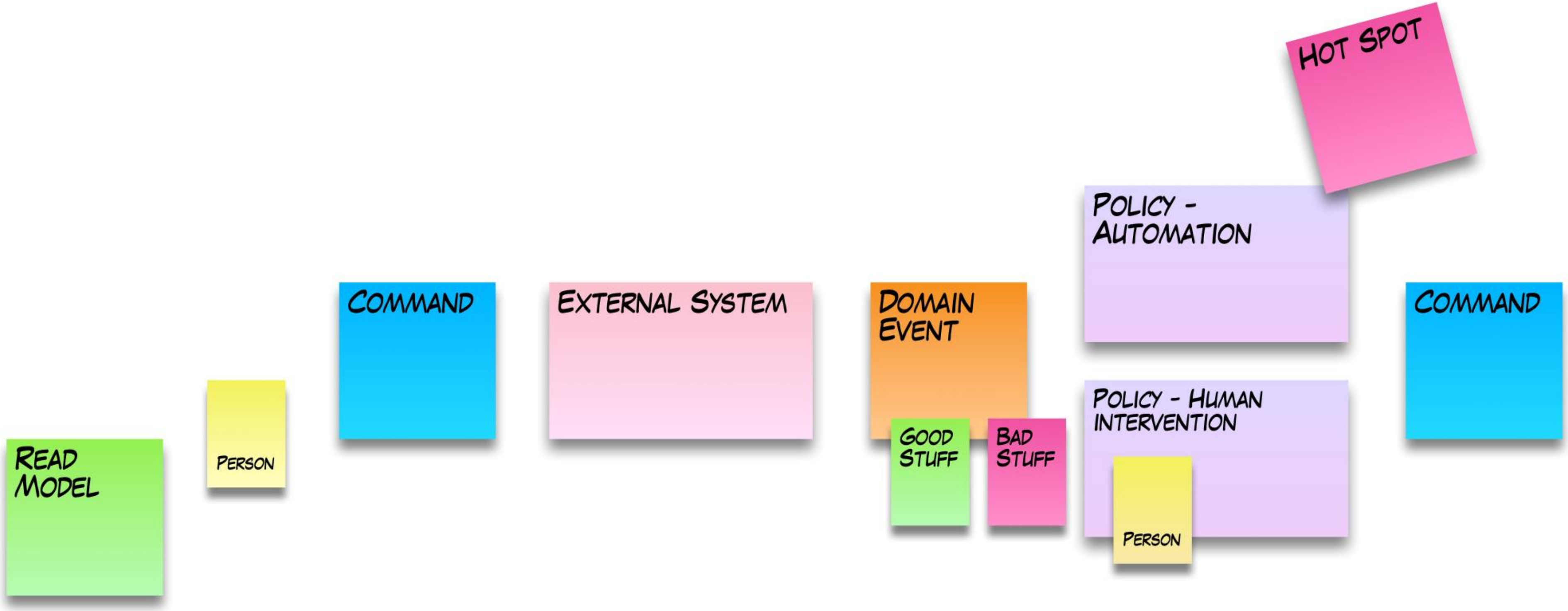
Why EventStorming?

If the goal is to learn the complexity of the domain in its different facets and possible inconsistencies, then why not involve all the key experts and build with them a model of the behaviour of the domain as a whole?

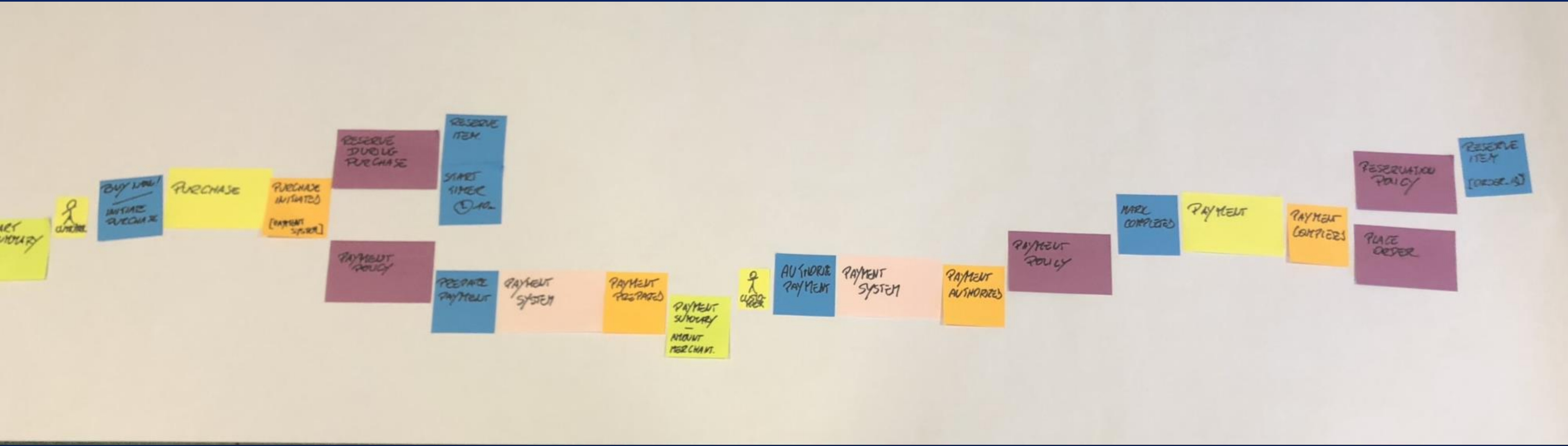
What is EventStorming?

1. Big Picture
2. Process Modelling
3. Software Design

Process Modelling Notation



Software Design



Focus on …

Command / Action - written in first person like ‘Create Sales Order’ it represents a user or a system intention.

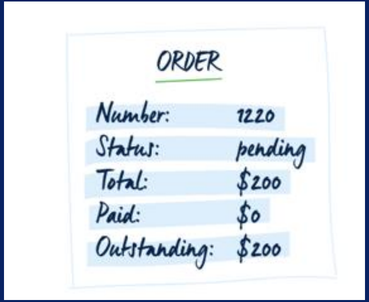
Event - something that happened, a state transition, usually the result of executing some action / command.

Aggregate – the name is not important, bit the role is. A software component whose internals and responsibilities we’re designing.

What is Event Sourcing?

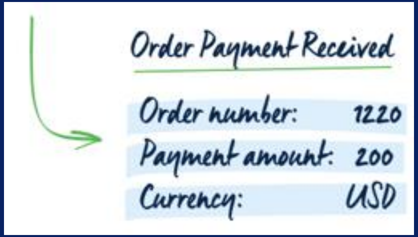
Event Sourcing is an architectural pattern where changes that occur in a domain are immutably stored as events in an append-only log.

How it Works?



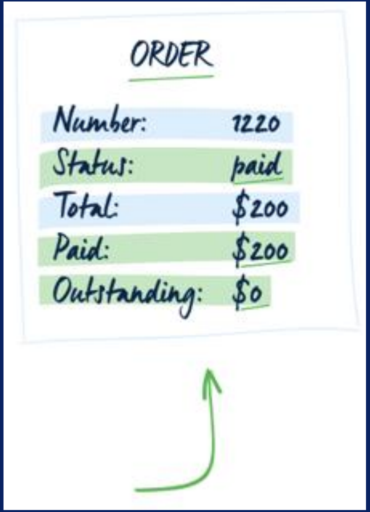
ORDER

Number:	1220
Status:	pending
Total:	$200
Paid:	$0
Outstanding:	$200



Order Payment Received

Order number:	1220
Payment amount:	200
Currency:	USD



ORDER

Number:	1220
Status:	paid
Total:	$200
Paid:	$200
Outstanding:	$0

In an event sourced system, the status change would be captured as an event «OrderPaymentReceived» and stored in the append-only log in the order in which it occurred

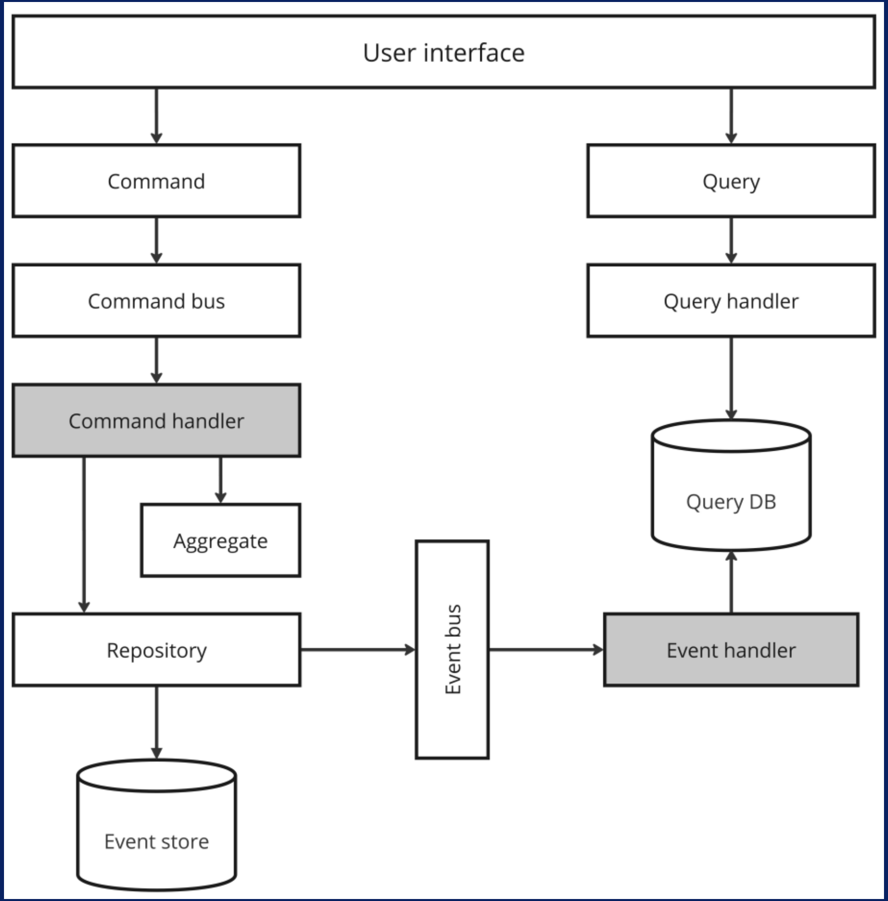
EventStoreDB

EventStoreDB is an open-source event-driven database, built to support Event Sourcing.

Command **Q**uery **R**esponsability **S**egregation - **CQRS**

Command Query Responsability Segregation (CQRS) is the segregation of the responsabilities of the commands and queries in a system. That means that we're slicing our application logic vertically.

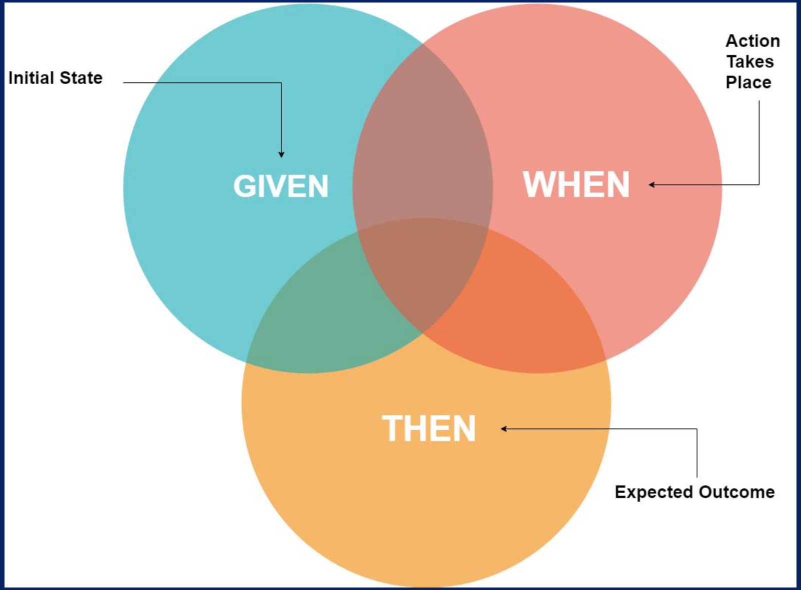
CQRS Flow


User interface
Command
Query
Command bus
Query handler
Command handler
Aggregate
Query DB
Event bus
Repository
Event handler
Event store

Show me the Code!

Behavior-Driven Development (BDD)

- The **Given** statement sets up the initial context for the behavior and defines the starting point of the system.
- The **When** statement describes the trigger that brings about a change or behavior in the system.
- The **Then** statement defines the expected outcome that should be observed after the mentioned in the When statement.



Show me the Code!

Thank You!

alberto.acerbis@intre.it

https://github.com/BrewUp/EventStoreDB-Course-2024

https://github.com/ace68