



Fluid Software Architektur

miro

lotharschulz.info

github.com/lotharschulz

speakerdeck.com/lothar

hachyderm.io/@lotharschulz

lnkd.in/lotharschulz

Lothar Schulz
Head of Engineering



Warum

Fluid Architektur

Was ist Software Architektur ?

The software architecture of a program or computing system is the **structure** or structures of the **system**, which **comprise software elements**, the **externally visible properties of those elements**, and the **relationships among them**.

Was ist Software Architektur ?



In most successful software projects, the expert developers working on that project have a shared understanding of the system design. This shared understanding is called architecture.

One architecture to rule them all

There is not one architectural pattern to rule them all.

How you choose to develop, deploy, and manage services will always be driven by the product you're designing, the skillset of the team building it, and the experience you want to deliver to customers (and of course things like cost, speed, and resiliency).

For example, a startup with five engineers may choose a monolithic architecture because it is easier to deploy and doesn't require their small team to learn multiple programming languages. Their needs are fundamentally different from an enterprise with dozens of engineering teams, each managing an individual subservice. And that's okay. It's about choosing the right tools for the job.

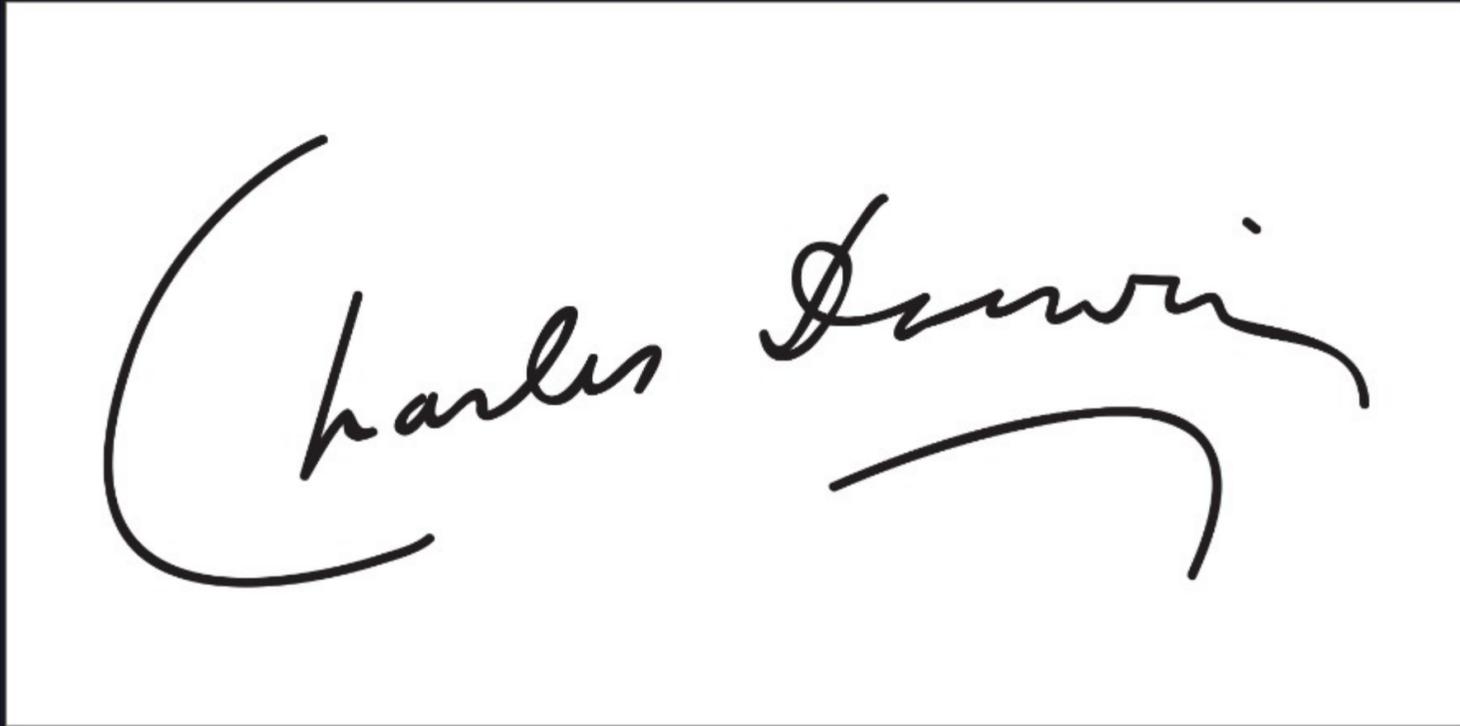
One architecture to rule them all

td;lr

Technologische Innovationen und neue Anforderungen machen Änderungen an der Architektur zwingend – sie kann nicht zukunftssicher sein.

Lösung: Dieser Realität in die Augen sehen und Architekturen anpassen, wenn notwendig.

Architektur anpassen, aber wie ?



“It is not the strongest of the species that survives, nor the most intelligent. It is the one that is most adaptable to change.” – often attributed to Charles Darwin

Wann ?

Wenn die meisten Informationen vorliegen

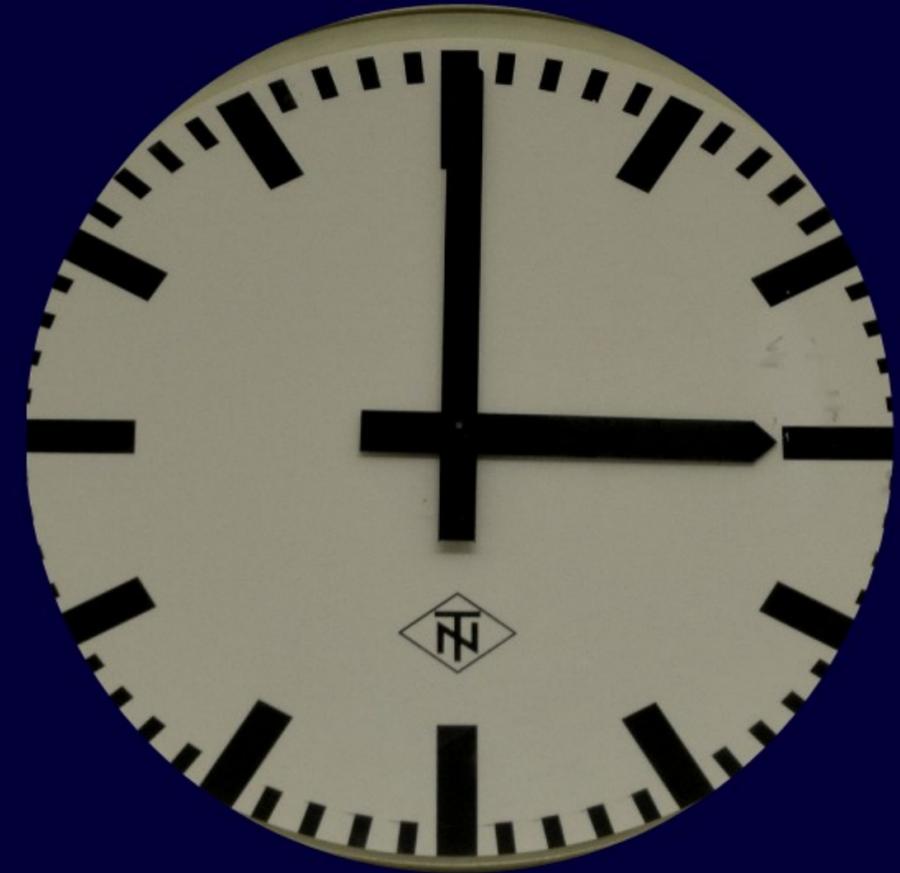
Last responsible moment (LRM)

The **last responsible moment (LRM)** is the strategy of delaying a decision until the moment when the cost of not making the decision is greater than the cost of making it.

LRM

Wann ist der
Last responsible moment

?



Architektur Fitness Funktionen



Architecture Fitness Funktionen

An architectural fitness function provides an objective integrity assessment of some architectural characteristic(s).

[Fitness Function Katas](#)

Fitness functions describe how close an architecture is to achieving an architectural aim

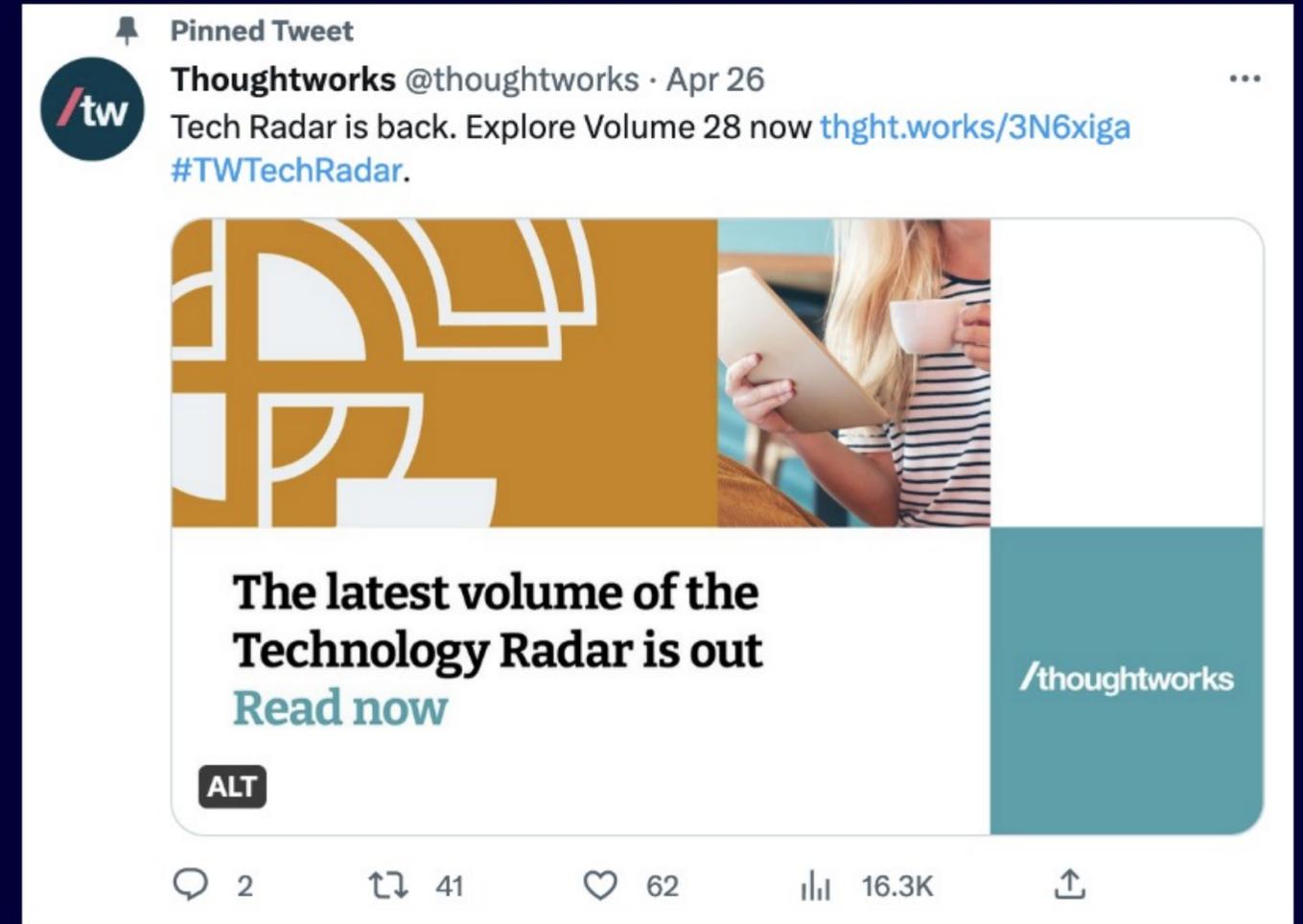
[Fitness function-driven development](#)

Architecture Fitness Funktionen

```
describe "Resiliency" do
  describe "New Deployment" do
    it "has less than 1% error rate for new deployment" do
      expect(new_deployment.get_error_rate()).to < .01
    end
  end
end
describe "Network Latency" do
  it "has less than 5% error rate even if there is network latency" do
    expect(network_tests.get_error_rate()).to < .05
  end
  it "completes a transaction under 10 seconds even if there is network latency" do
    expect(network_tests.get_transaction_time()).to < 10
  end
end
end
end
```

Architektur Fitness Funktionen

cost as architecture
fitness function



Fluid Architektur - Erfahrungen



Interfaces & Verträge

Zweideutigkeiten und unerfüllte Erwartungen

Interfaces & Verträge

request & response

events

request & response

REST Maturity Models

- Level 0 - remote interactions
- Level 1 - Resources
 - /books/123
 - /books/123/author
- Level 2 - HTTP Verbs
 - GET, POST, PUT, HEAD, PATCH, DELETE
- Level 4 - Hypermedia Controls
 - `<link rel="slots/book" uri="/slots/123">`

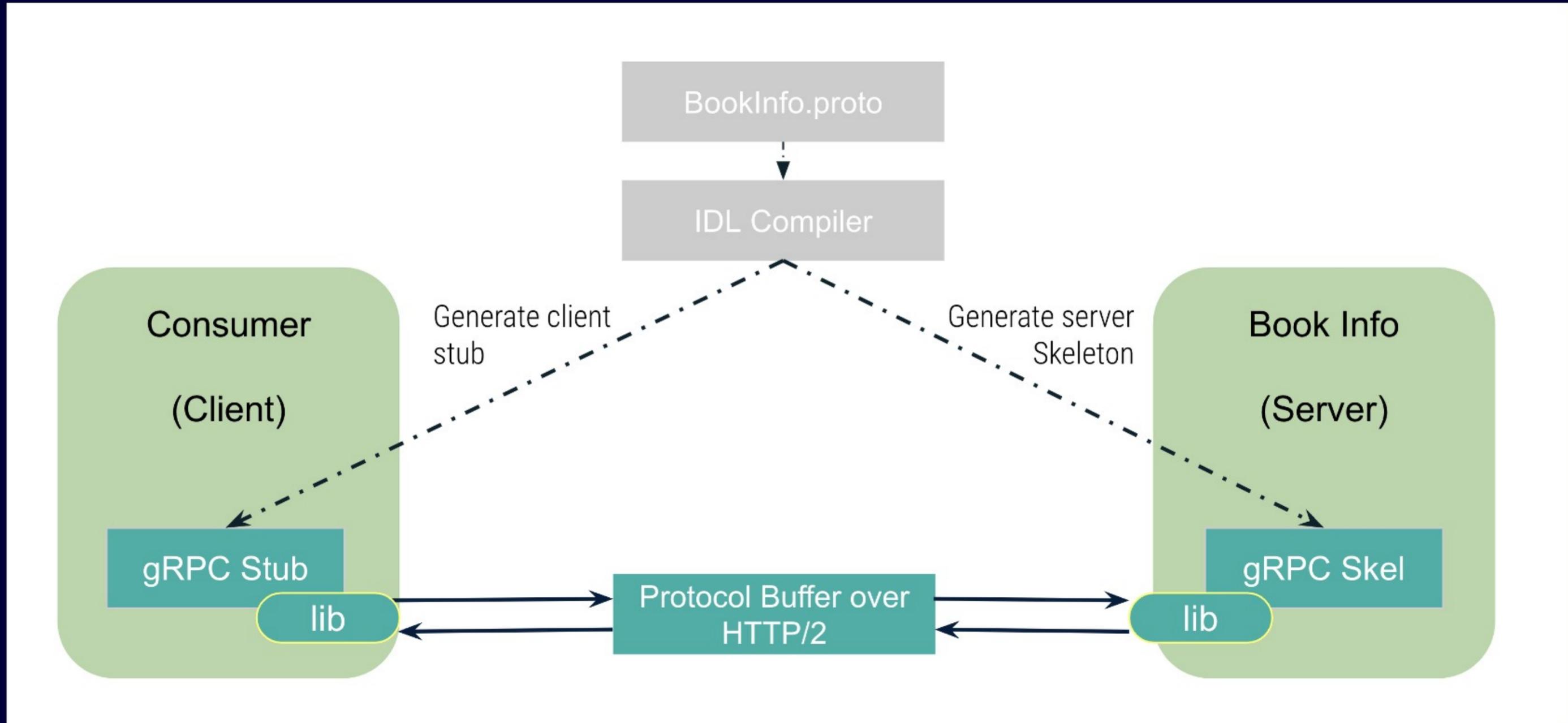
request & response

REST Maturity Models

- Level 0 - remote interactions
- Level 1 - Resources
 - /books/123
 - /books/123/author
- Level 2 - HTTP Verbs
 - GET, POST, PUT, HEAD, PATCH, DELETE
- Level 4 - Hypermedia Controls
 - `<link rel="slots/book" uri="/slots/123">`



request & response



request & response

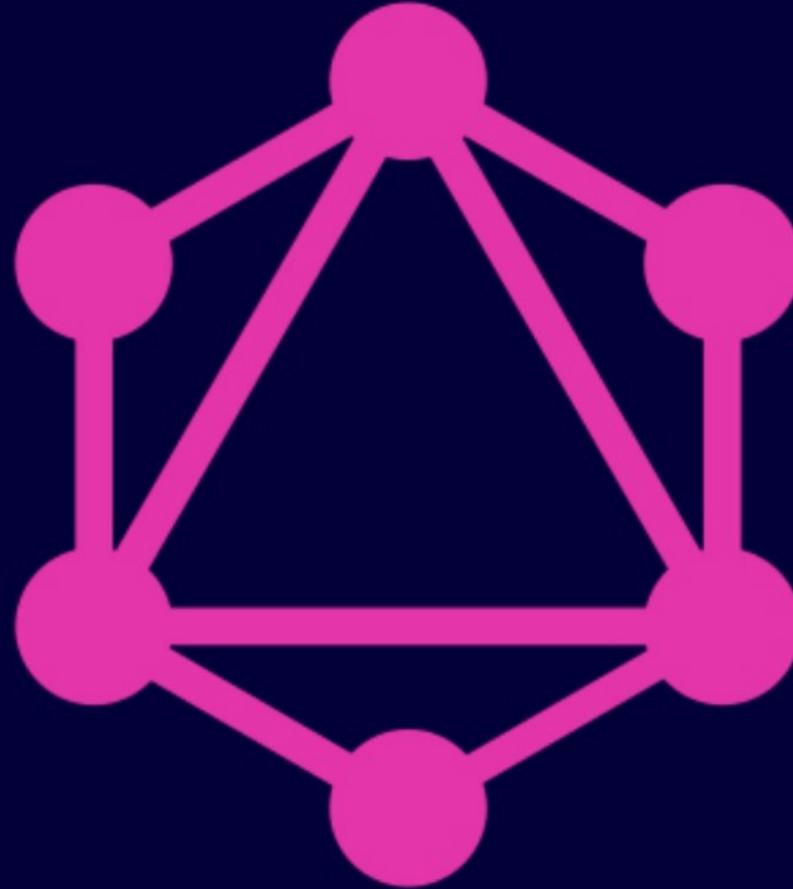
Protocol Buffers v3.15.0

 acozzette released this Feb 18, 2021 · [47 commits](#) to 3.15.x since this release  v3.15.0  ae50d9b

Protocol Compiler

- Optional fields for proto3 are enabled by default, and no longer require the `--experimental_allow_proto3_optional` flag.

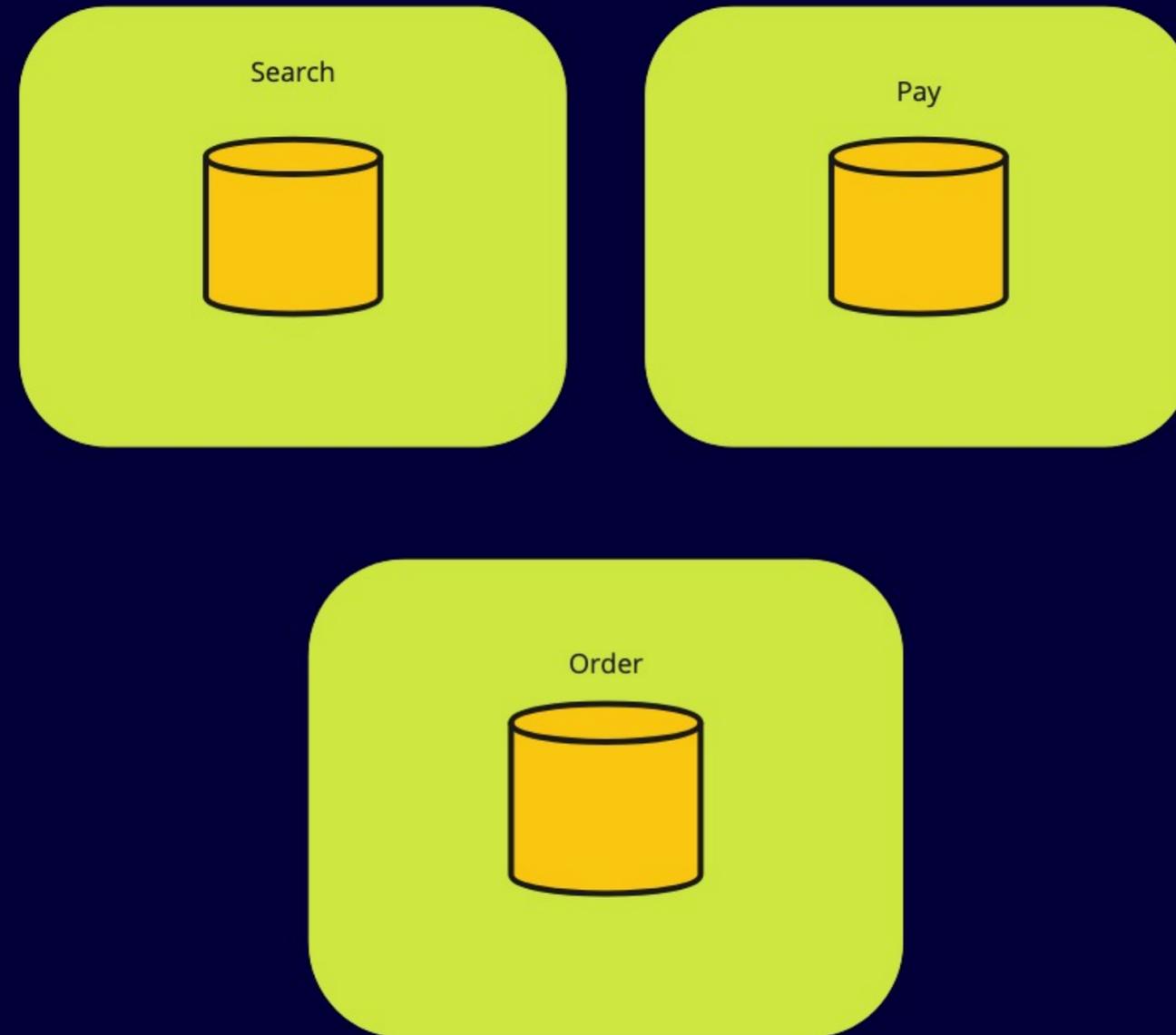
request & response



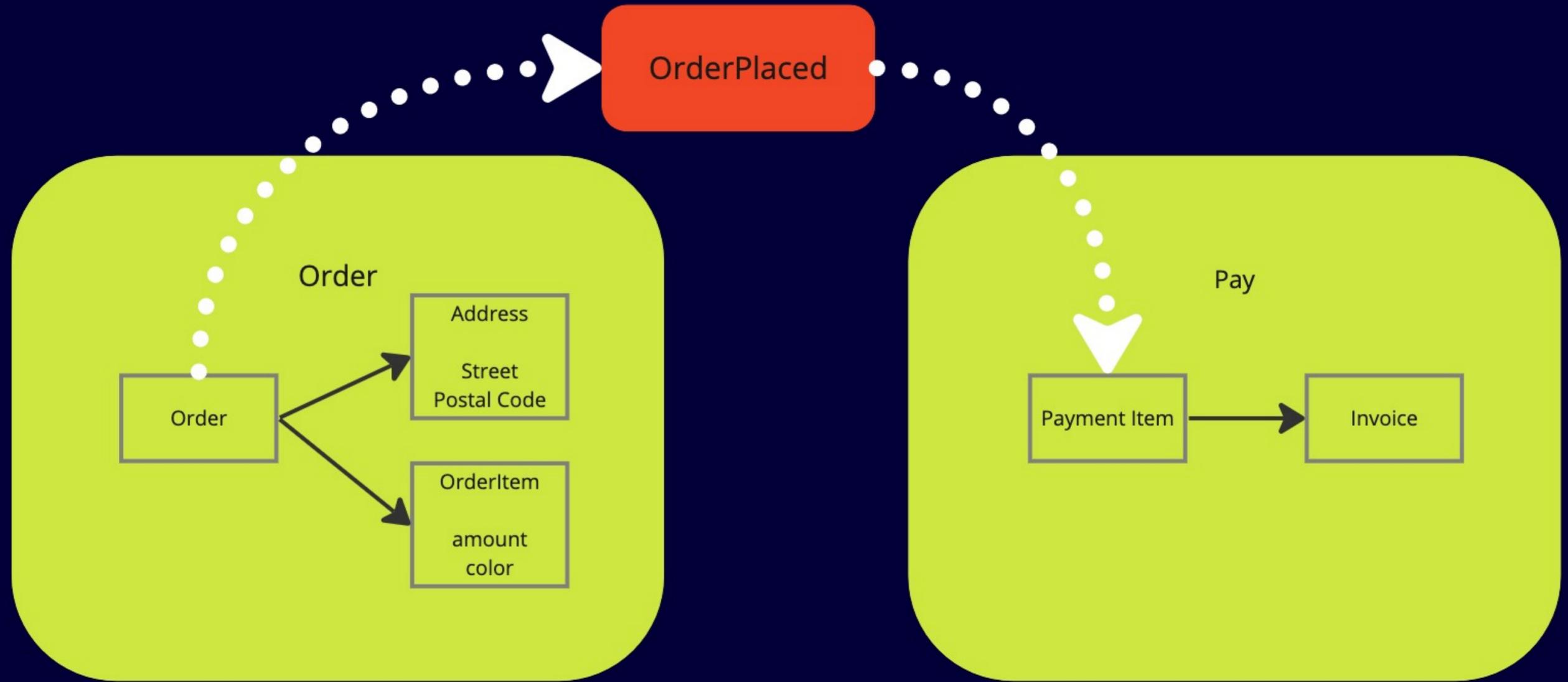
events



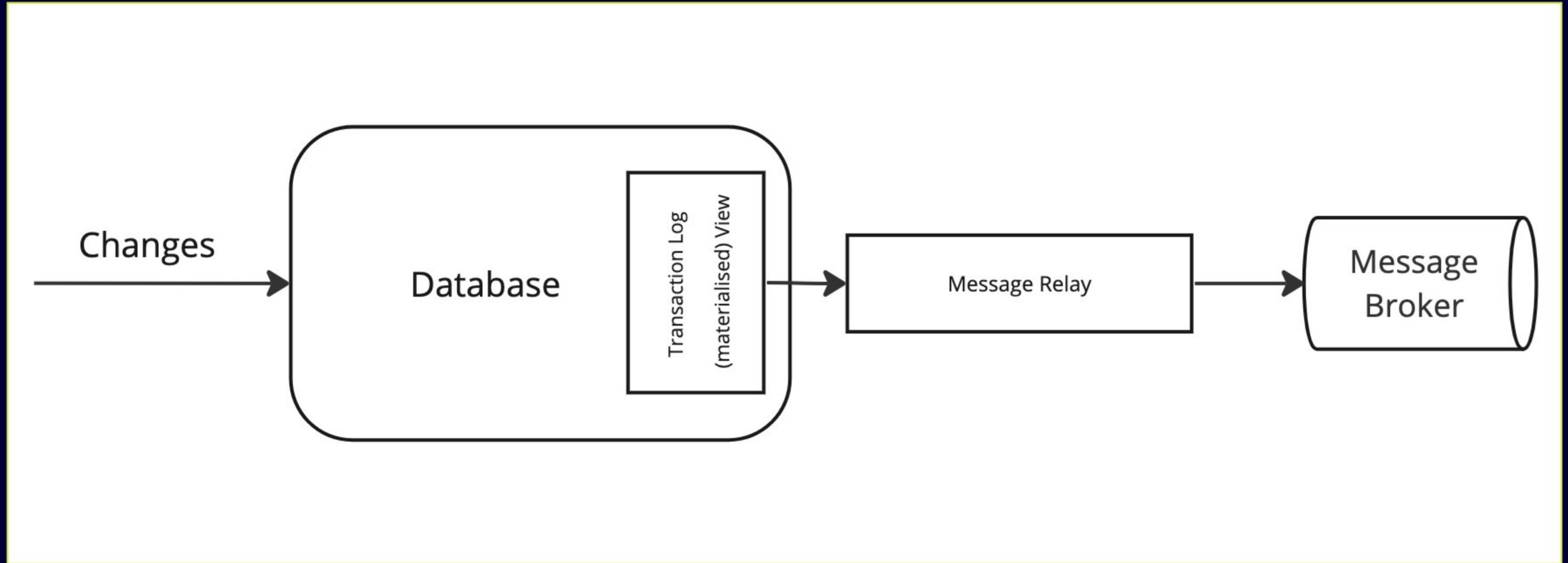
events



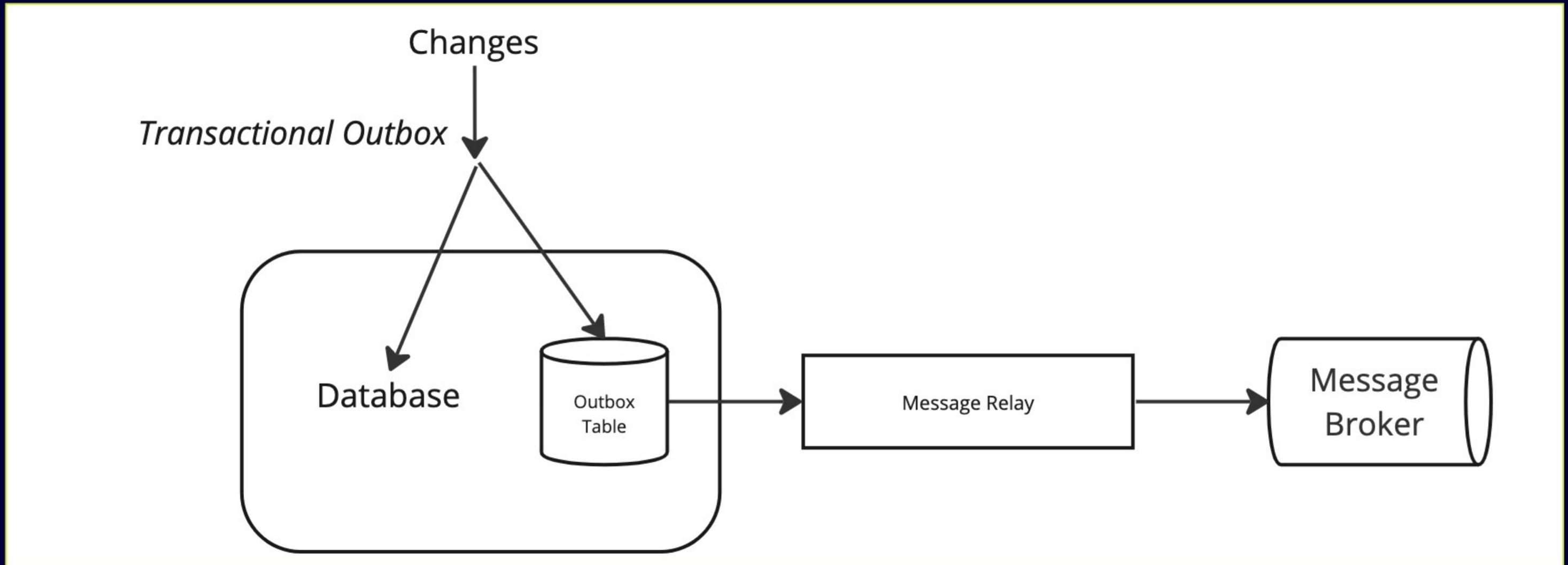
events



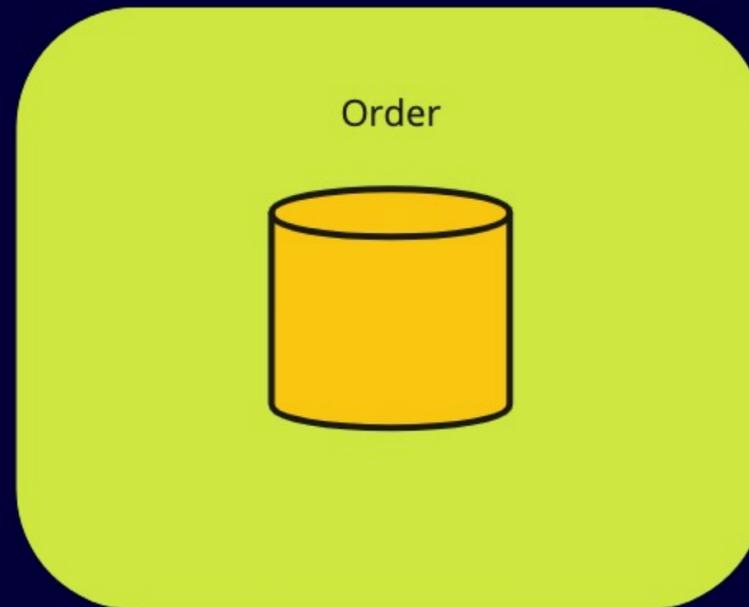
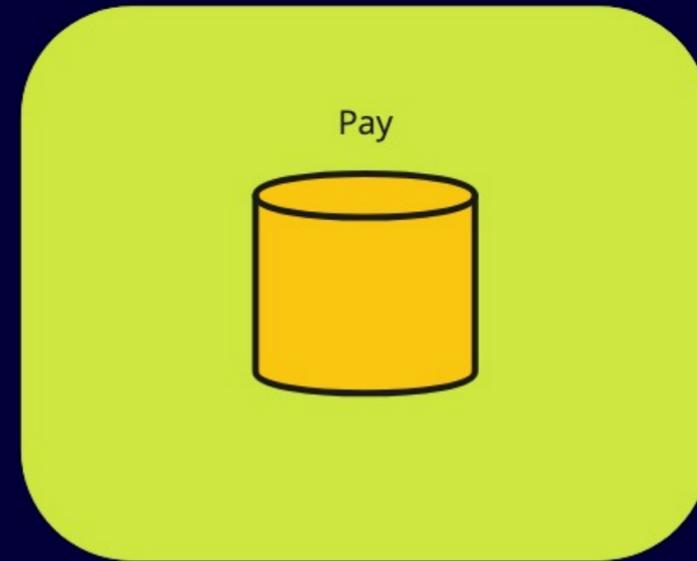
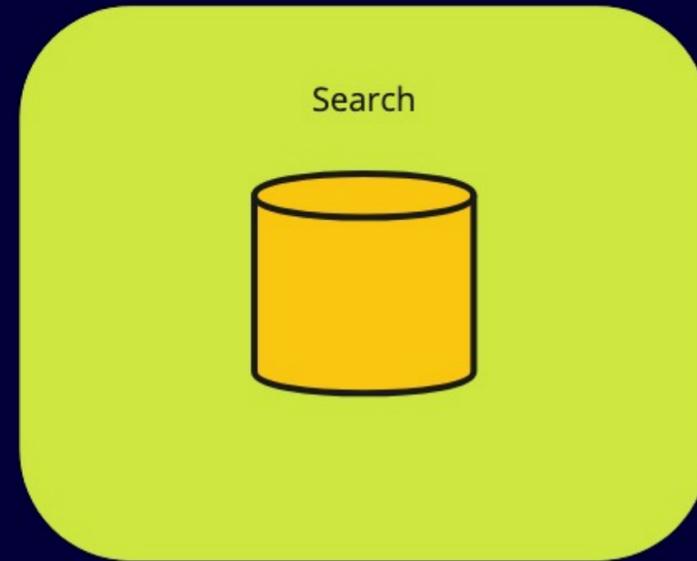
Domain events as a preferred way to trigger side effects across multiple aggregates within the same domain



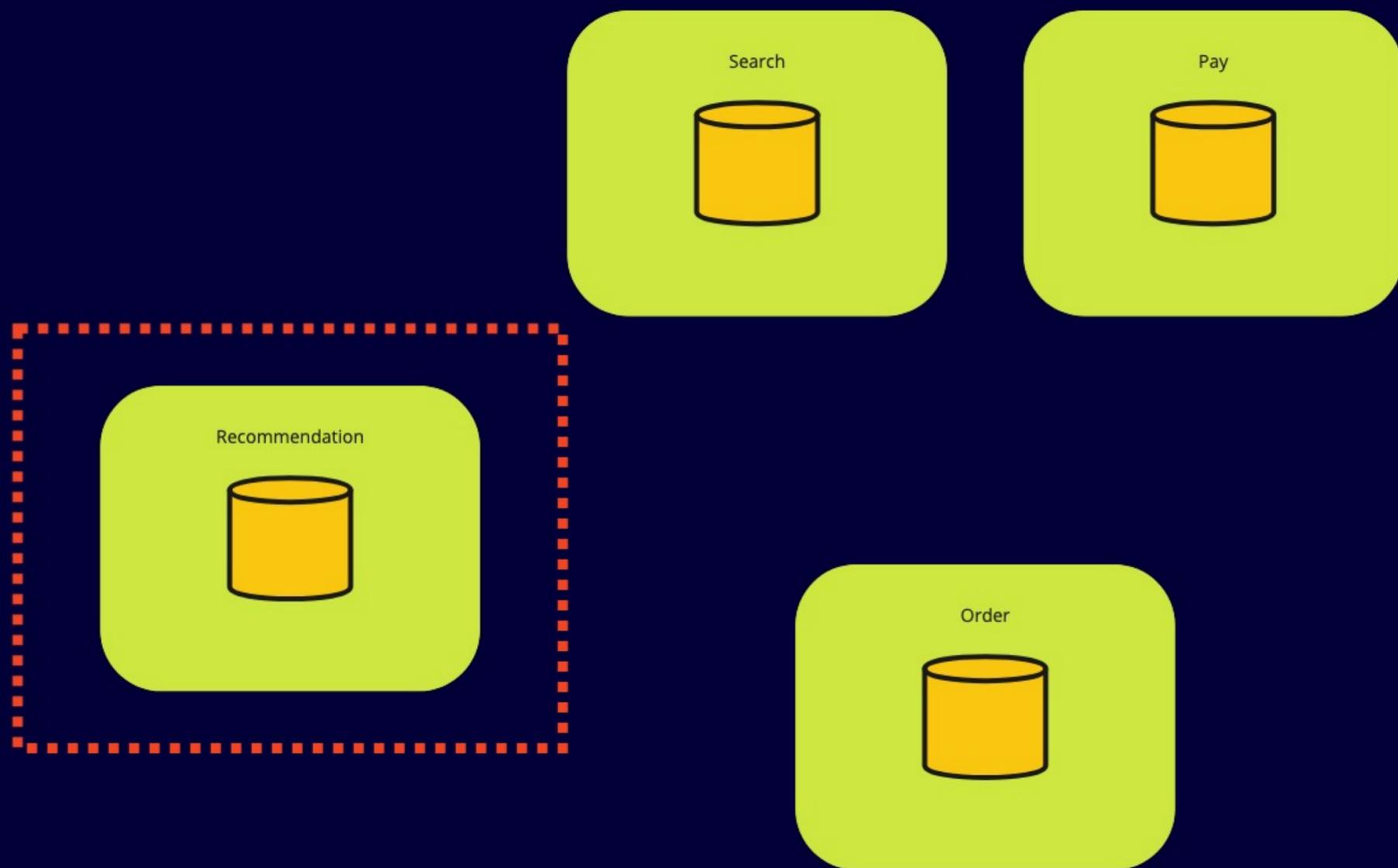
Outbox



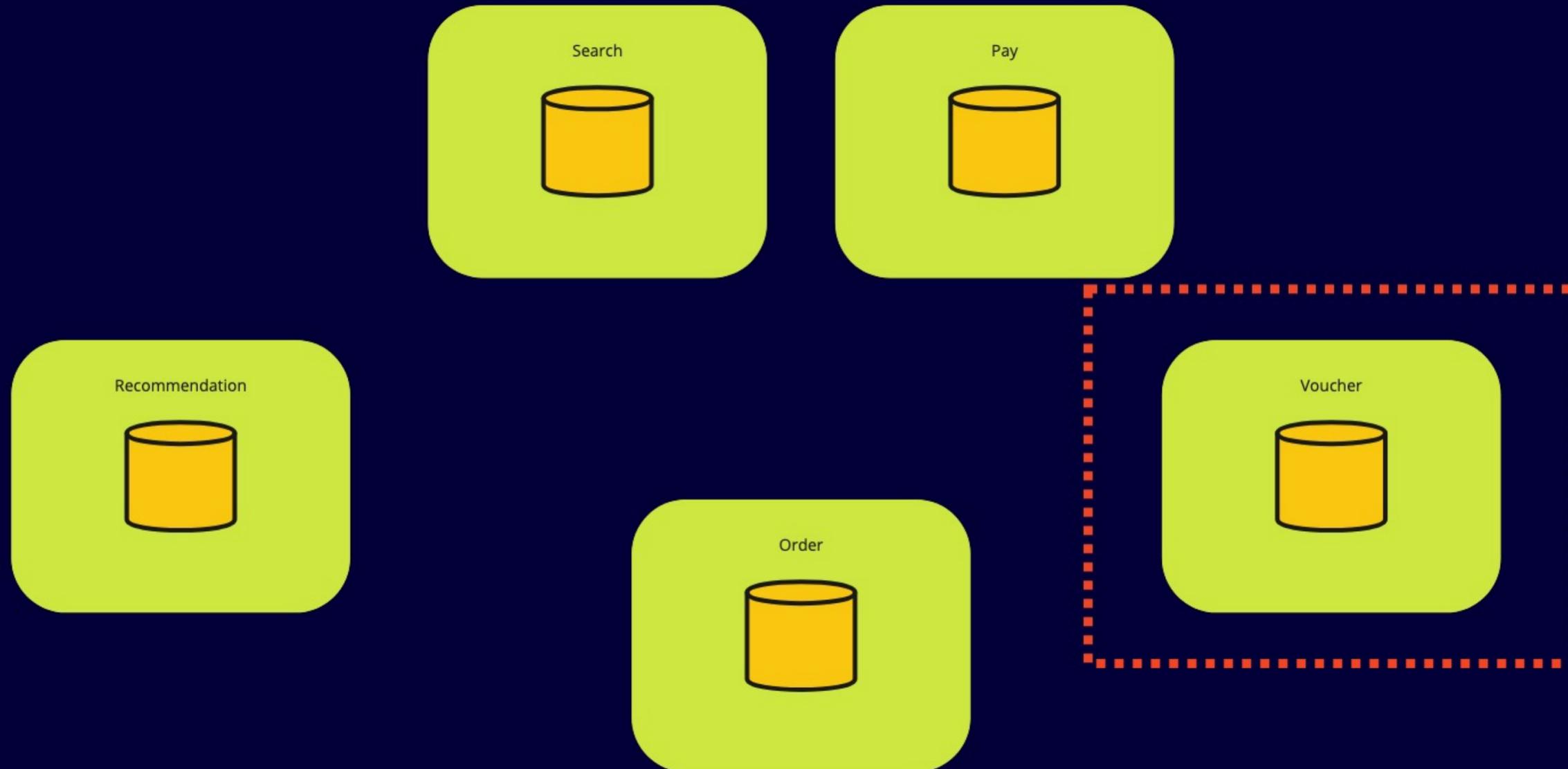
Domains



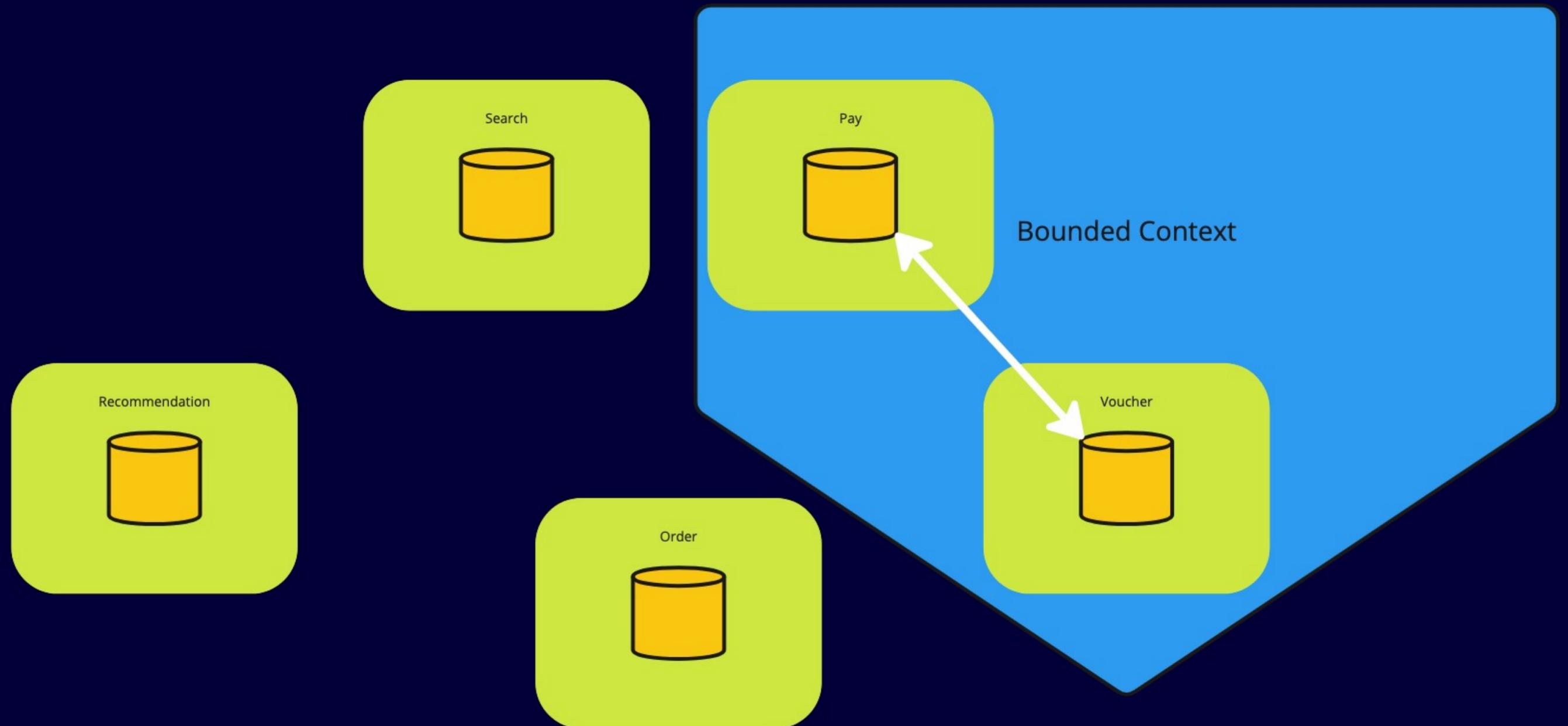
Domains



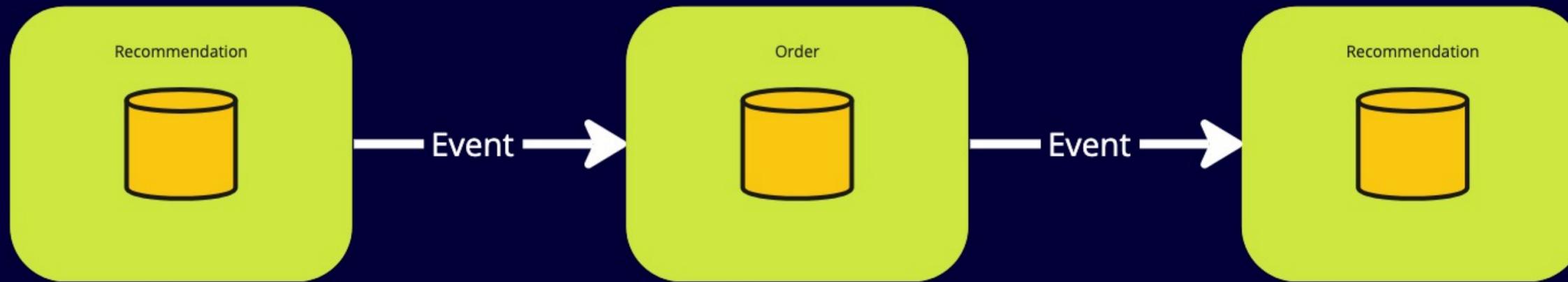
Domains



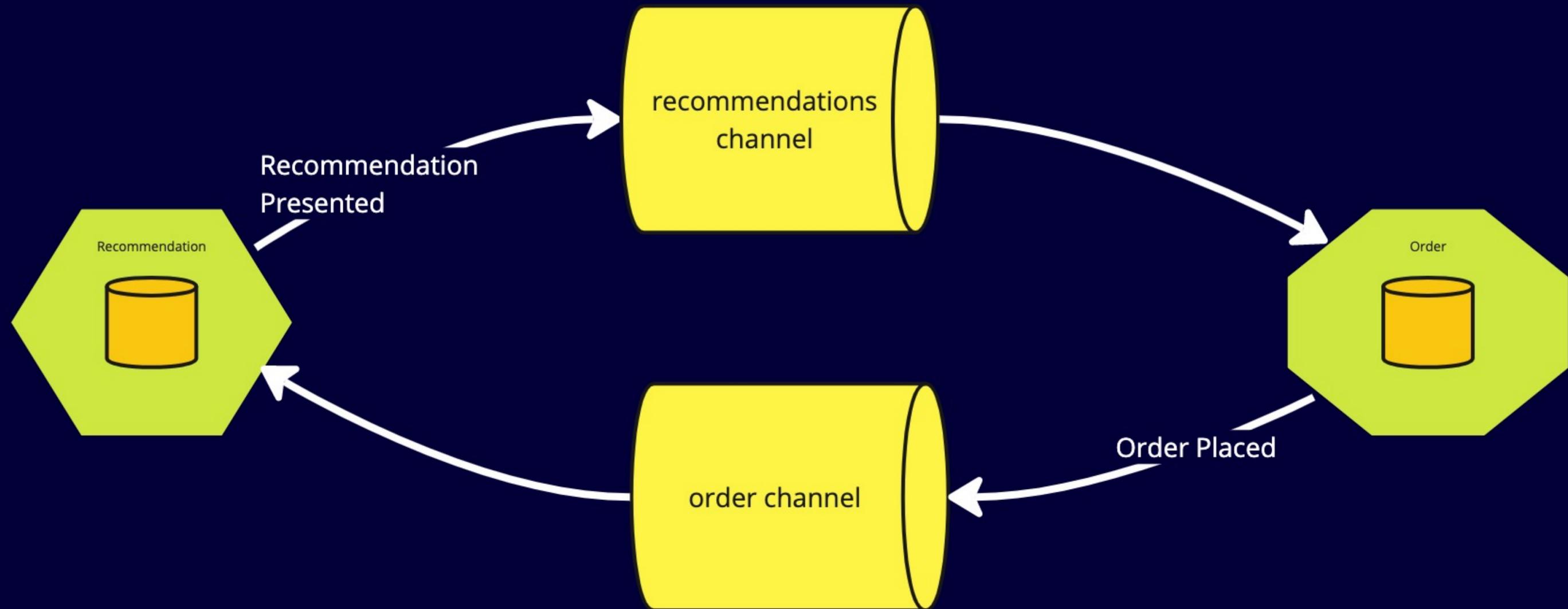
Domains



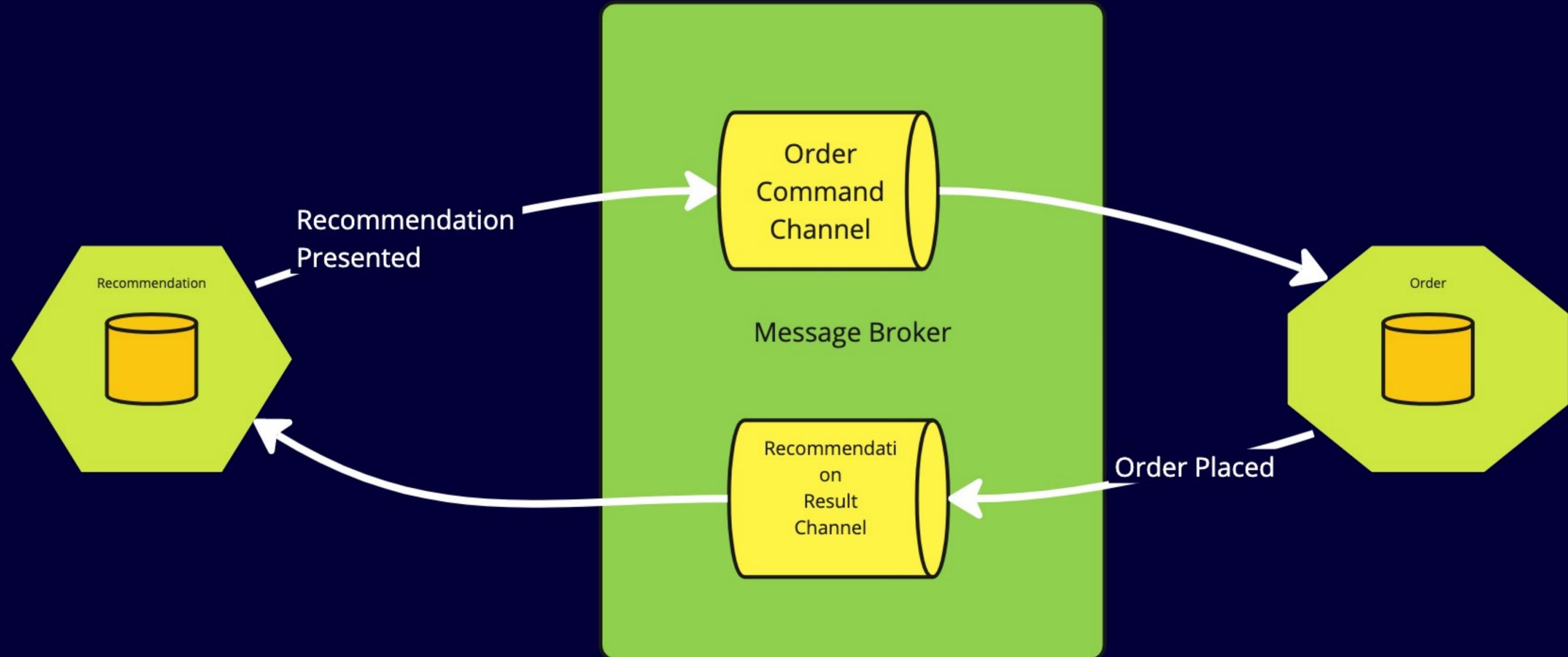
business transaction that spans multiple services - saga



Choreography-based saga



Orchestration-based saga



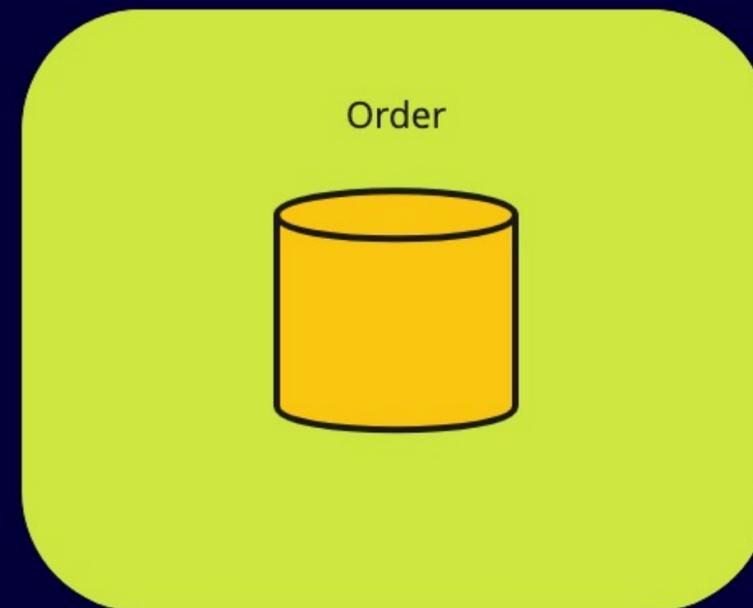
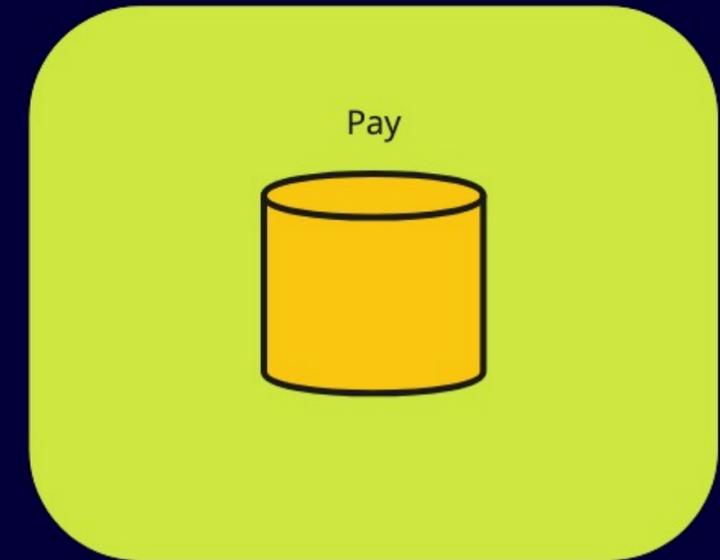
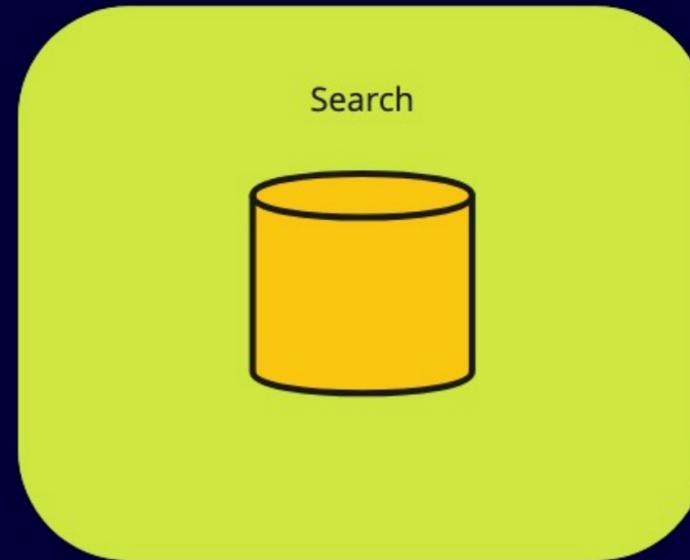
Storage



Storage

Scaling

- Vertical
- Horizontal
- Partitioning
 - Sharding



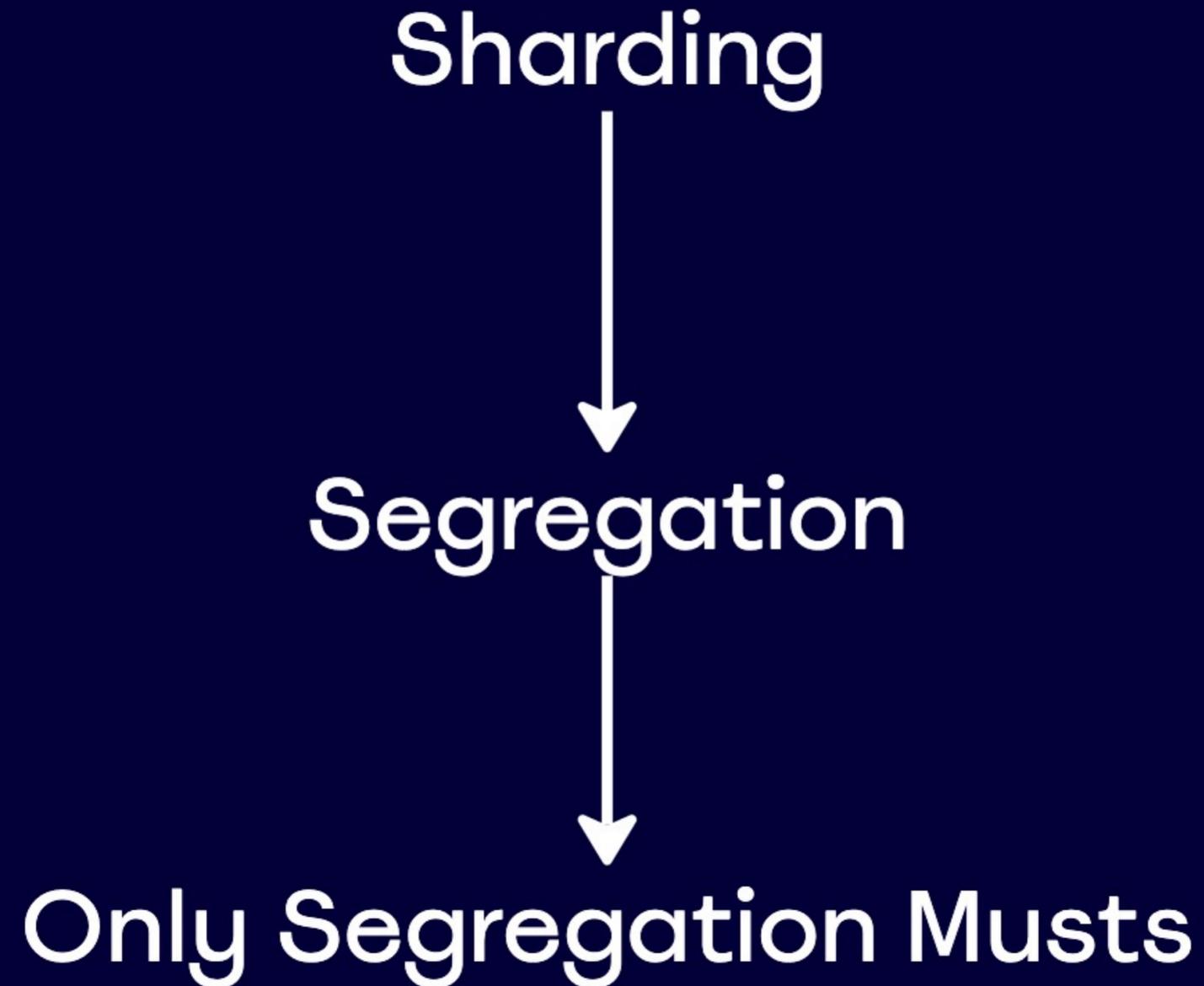
Storage

Sharding



Segregation

Storage



Events Registry
Schema Registry

Cluster Registry
Fare Registry
Canvas Registry

I am sure you have questions

