

A decorative graphic on the left side of the slide consisting of two overlapping parallelograms. The front one is blue and the back one is a light green color. They are positioned diagonally, with the blue one partially covering the green one.

Event Driven Architecture

Huy Nguyen - GHTK Tech Center



Who am I

- PHP lover, Laravel enthusiast
- Over 6 years experience
- Currently working at GHTK Tech Center

My main interests: Software Architecture, Distributed system

Love sharing, writing



Problems



Problems

- Mutable state
- Scaling your system
- Tight coupling
- Debugging

You think you know your system ?. Absolute NO !



What is Event Driven Architecture ?



What is event ?



Event sourcing



Application State
(Domain Models)

Replay of All
Events

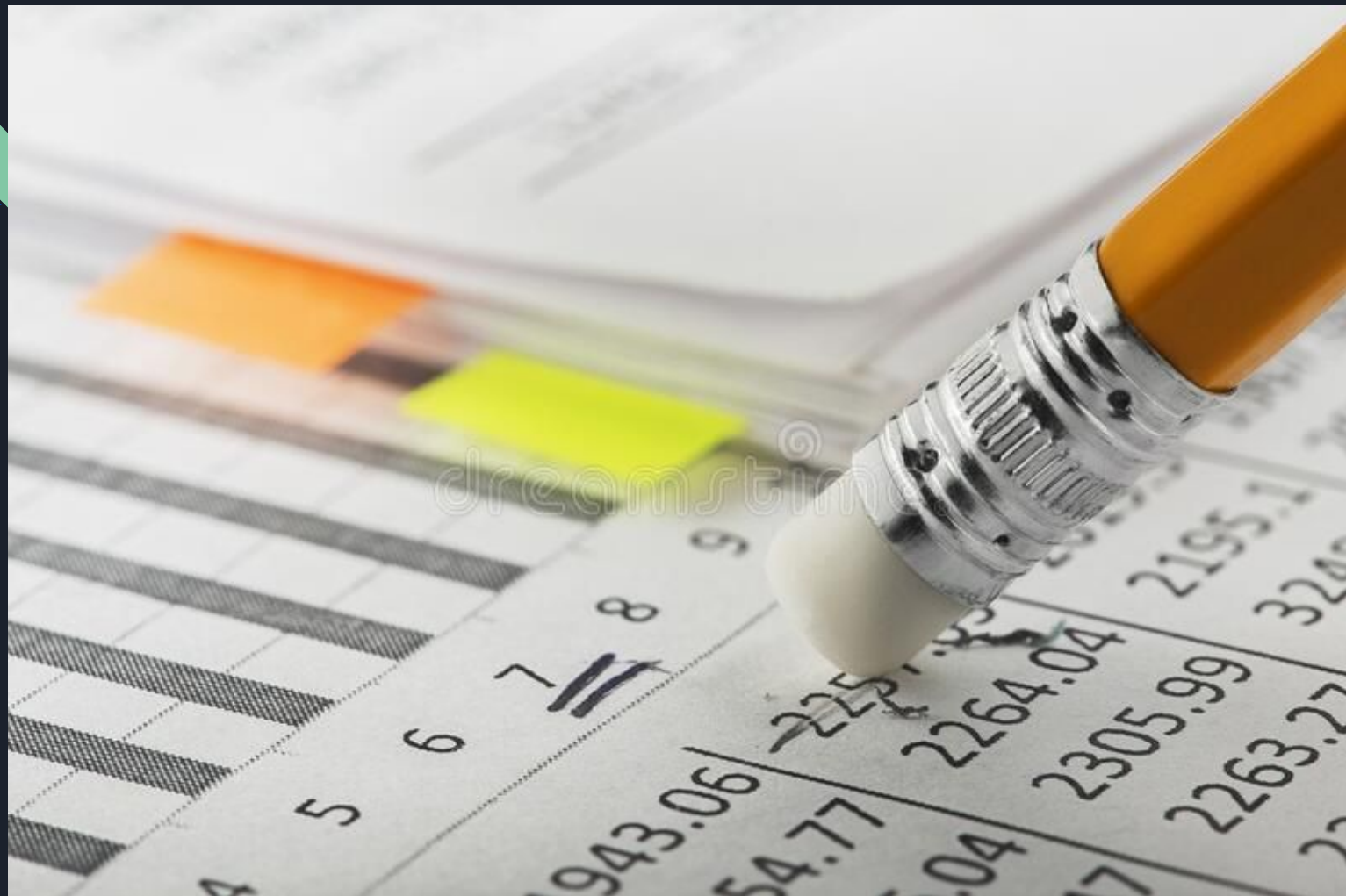
Event Store
(Persistence)

Event



```
graph LR; ES[Event Store Persistence] -- "Replay of All Events" --> AS[Application State Domain Models];
```

The diagram illustrates the Event Sourcing pattern. On the right, a large light blue rectangle represents the 'Event Store (Persistence)'. Inside this rectangle, a series of blue rounded rectangles are stacked diagonally from the top-left towards the bottom-right, representing a sequence of events. The bottom-most event is labeled 'Event'. A blue line originates from the bottom of the Event Store, extends horizontally to the left, and then turns vertically upwards with an arrowhead pointing to the 'Application State (Domain Models)' box on the left. The text 'Replay of All Events' is placed next to the horizontal segment of this line.





Event sourcing says all state is transient and you only store facts



Order Created

GHTK takes the
package

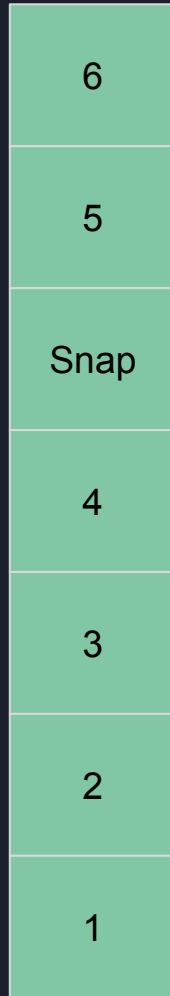
Shipping

Money !





I can't query a series of events
And I wanna delete it





mongoDB



CouchDB



Cassandra

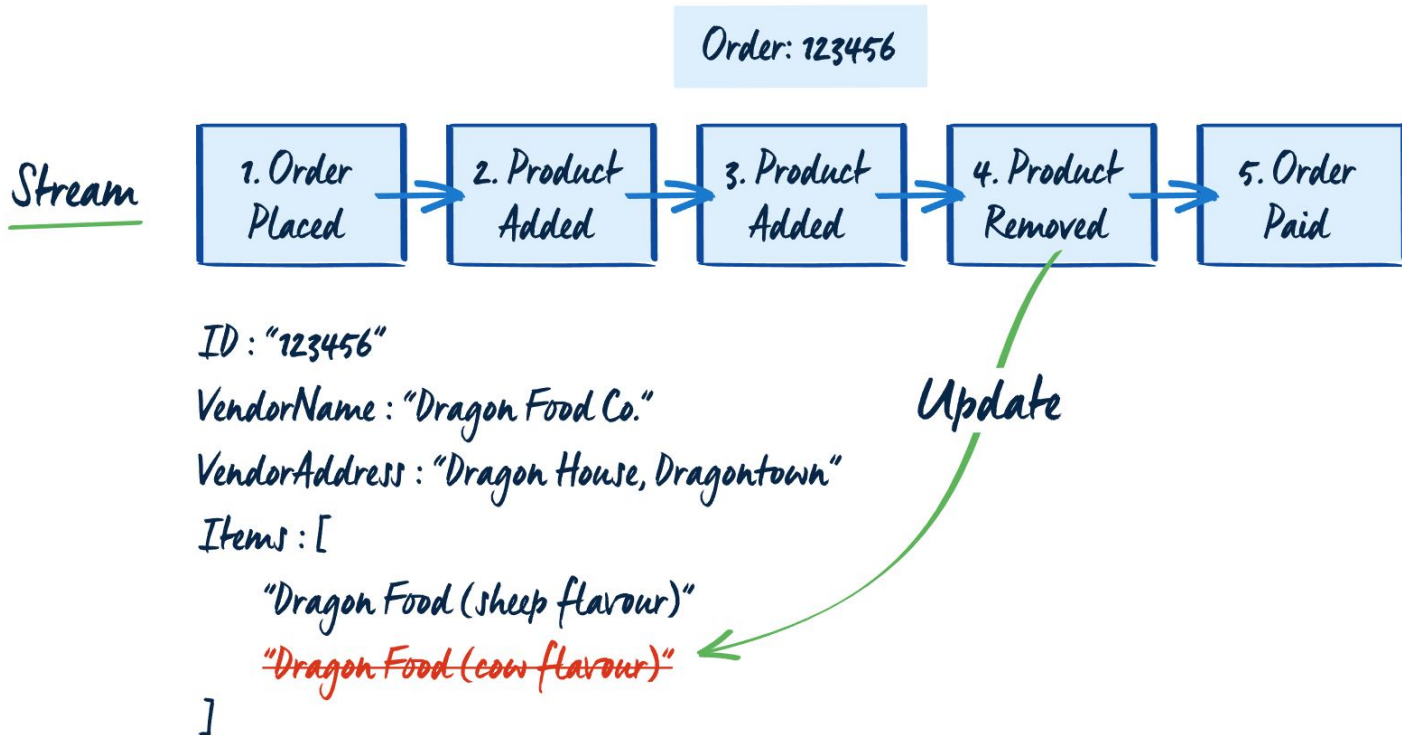


EVENT STORE

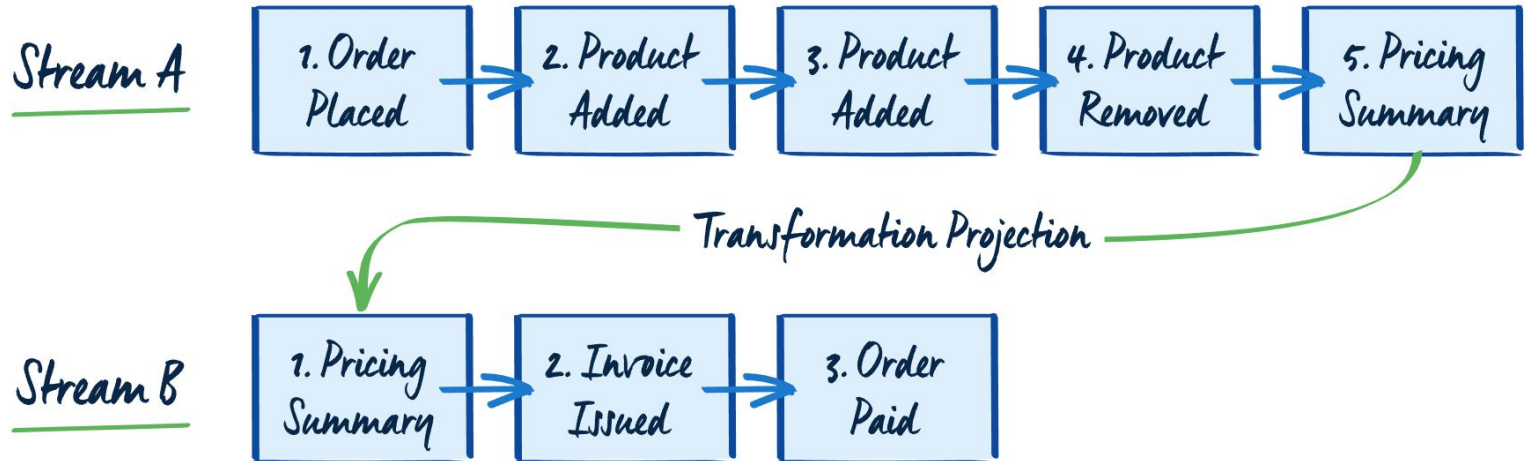


Event Sourcing components

Projector



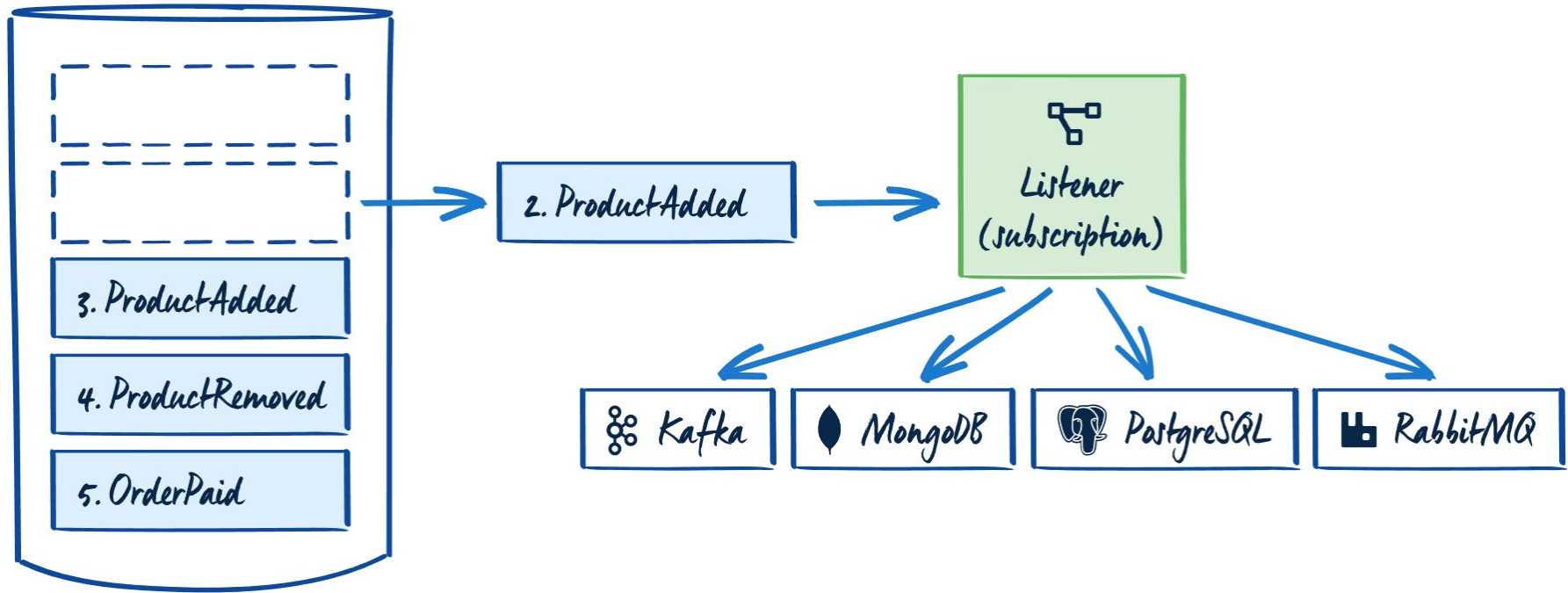
Projector





Reactor

Subscriber

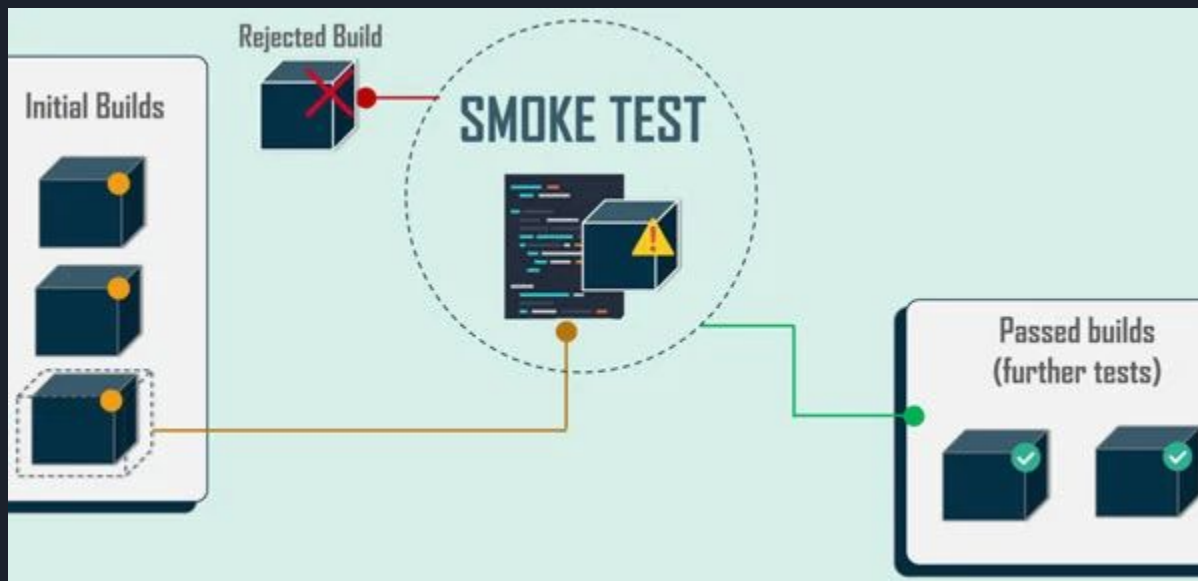




Use cases



Smoke testing



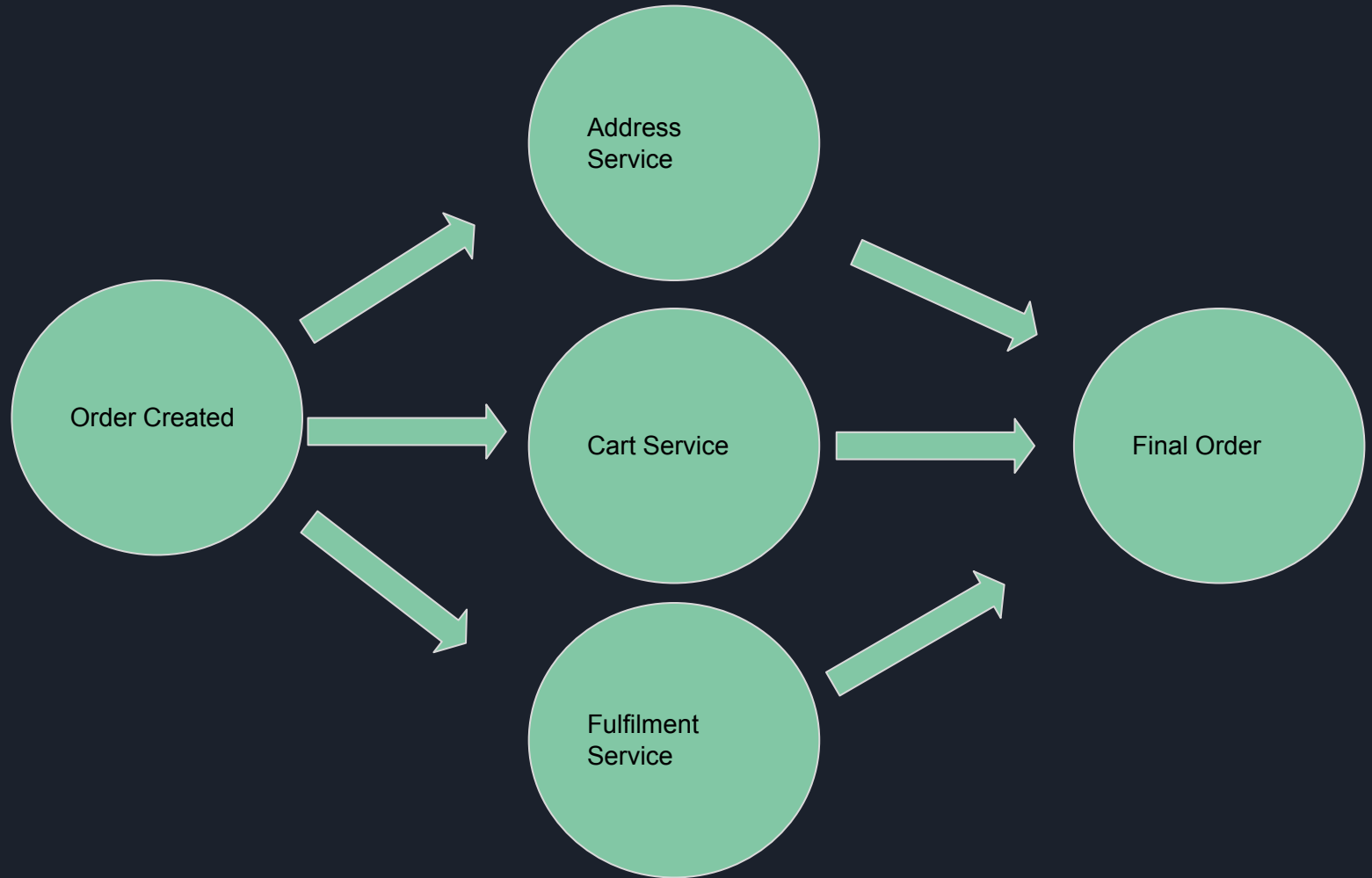


Super User Attack





Parallel Processing High Performance





Debugging your production ?
Release on weekend ?



What if my events are huge ?

If it fits on your RAM, there are no problems

Or on a Micro-SD



CQRS

Command and Query Responsibility Segregation

Write (Command)

Domain
Object

Domain
Object

Application Services

Remote Facade

Message Sent
Ack / Nak
Response



Client

Read (Query)

Thin Read Layer (Direct to DTO)

Remote Facade

Request DTO
DTO Returned





Scaling your system



Dealing with eventual consistency



A close-up shot of Thanos from the movie Avengers: Endgame. He is shown from the chest up, wearing his iconic purple armor which appears heavily worn and cracked. He has a purple, wrinkled face with a thin, grey beard. His right hand, clad in a gauntlet, is raised towards his face. The background is a dark, smoky, and hazy battlefield scene. The text "eventual consistency" is overlaid in a white, italicized font across his face.

eventual consistency

I am inevitable



Solutions

UI solution

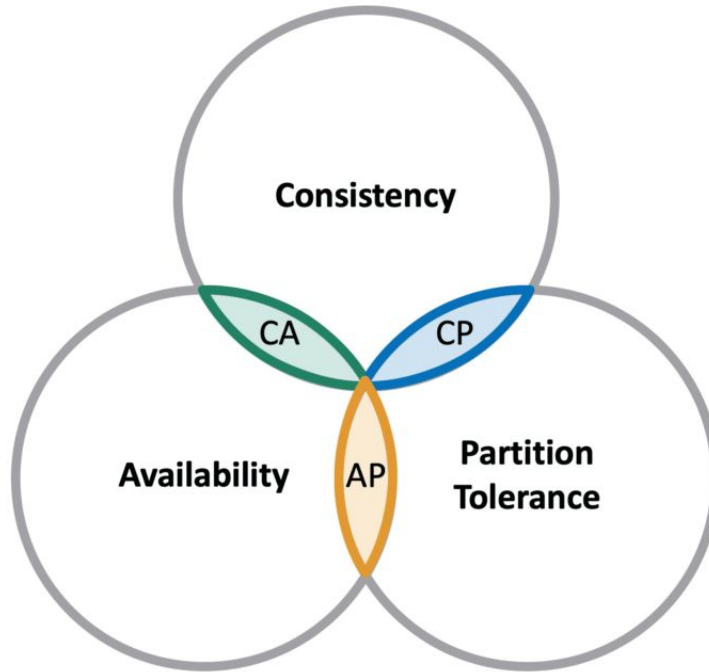
Aggregates are

boundaries



Distributed Transaction

CAP Theorem





2PC or Saga ?



Why event sourced system fail ?



Versioning



Domain Driven Design



Eventual consistency
Yes, again !!!!!

Together !





Thank you
Q&A