

Playing with Projections

by @michelgrootjans & @tcoopman

@DDDBER with @rradczewski

Please clone & set up this repository:

git.io/fpz6w

Agenda

1. About projections
2. About the domain
3. Matchmaking
4. Workshop
5. Retrospective

Projections

What is a projection?

A projection is a function that derives a
specific information from a sequence of *domain events*

Projections in EventSourcing

All data in an ES-system is stored as a *Domain Event*

```
CustomerPutItemInBasket(basketId, itemId)
CustomerChangedQuanityInBasket(basketId, itemId, quantity)
CustomerRemovedItemFromBasket(basketId, itemId)
CustomerOrderedBasket(orderId, basketId, deliveryAddress)

OrderWasShipped(orderId, trackingCode)
OrderItemWasPartiallyReturned(orderId, itemId, reason)
```

3-column-table-for-everything™

```
CREATE TABLE events (
    aggregateId UUID NOT NULL,
    aggregateVersion BIGINT NOT NULL,
    eventPayload JSON NOT NULL,
);
CREATE UNIQUE INDEX aggregate_consistency
    ON events (aggregateId, aggregateVersion);
```

Flashback to 2016: youtube.com/watch?v=b0X0akLqXeA

All data has to be derived from these events

- Login credentials
- Product catalogue
- Product availability
- List of orders for a customer

...and can then be stored in a database or cache for querying

That's pretty cool, because...

- projections are created for **exactly one** specific use-case
- they can be thrown away or changed at will, no data-loss
- they can be scaled independent of the core domain

 Example time 

Given this sequence of events,

```
[  
    PlayerHasRegistered(...), NewGameWasCreated(...),  
    PlayerHasJoined(...), PlayerHasJoined(...), NewGameWasCreated(...)  
]
```

how many games were ever started?

Procedural or functional

```
function countGames(events) {  
  let numberOfGames = 0;  
  for(let event of events) {  
    if(event.type === 'NewGameWasCreated') {  
      numberOfGames++;  
    }  
  }  
  return numberOfGames;  
}
```

```
const countGames = events =>  
  events.reduce(  
    (count, { type }) => count + (type === 'NewGameWasCreated' ? 1: 0),  
    0)
```

Asking the standard questions



Asking the standard questions



- What's the hashed password for the user with the given email address? *LoginTable*

Asking the standard questions 😴

- What's the hashed password for the user with the given email address? *LoginTable*
- Which quizzes are currently published and available to play? *GamesTable*

Asking the interesting questions



Asking the interesting questions



- How many players signed up but never played a game?

Asking the interesting questions



- How many players signed up but never played a game?
- How many once-active (⚠) players stopped playing (⚠)?

Asking the interesting questions



- How many players signed up but never played a game?
- How many once-active (⚠️) players stopped playing (⚠️)?
- Which quiz makes people so angry that they quit their session or never come back at all?

Domain

Quiz game



- Players can create new *Quizzes* and add Questions with the correct answer to them
`PlayerHasRegistered, QuizWasCreated, QuestionAddedToQuiz, QuizWasPublished`
- A new *Game* based on a *Quiz* can be started and *Players* can join the game
`GameWasOpened, PlayerJoinedGame, GameWasStarted`
- A *Player* needs to answer a *Question* in time before the timer runs out
`QuestionWasAsked, AnswerWasGiven, TimerHasExpired, QuestionWasCompleted`
- A *Game* is finished after the last *Question* was completed
`GameWasFinished, GameWasCancelled`

Our streams for today



0.json

4 events

Basic example

2.json

43727 events

Exhaustive example with
meaningful data

5_2015_01_2017_01.json

104712 events

Huge example with compressed
names

10_2015_01_2016_01.json

48960 events

Exhaustive example with
compressed names

Pairing



Tests make Raimo happy



Be mindful of each other



Share the keyboard 50:50



Give people a chance to learn

Matchmaking

What languages do we have in the room?

Matchmaking

What languages do we have in the room?

Join your programming language ambassador!

Matchmaking

What languages do we have in the room?

Join your programming language ambassador!

Split the group between those who have a notebook with them and those who don't!

Matchmaking

What languages do we have in the room?

Join your programming language ambassador!

Split the group between those who have a notebook with them and those who don't!

Pair up with someone of the other part of the group!

Workshop

Repo: <https://git.io/fpz6w>

Solutions: <https://git.io/fpgY2> (no cheating 😊)

Challenges:

- How many events does a stream have?
- How many registered players do we have?
- How many registered players do we have per month?
- How many players haven't played in a month?
- What is the most popular quiz?
- What's the hardest (or easiest) question we have?
- How many players signed up but never played a game?
- How many once-active players stopped playing?
- Which quiz makes people so angry that they quit their session or never come back at all?