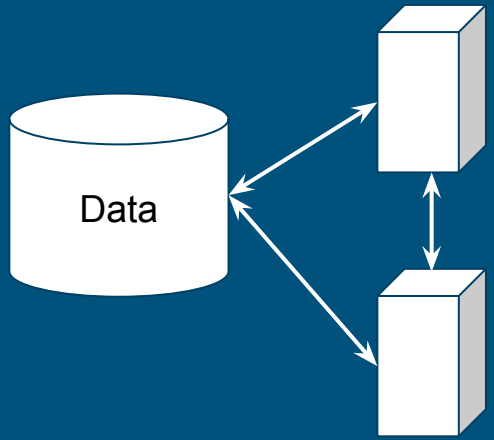
Event Sourcing

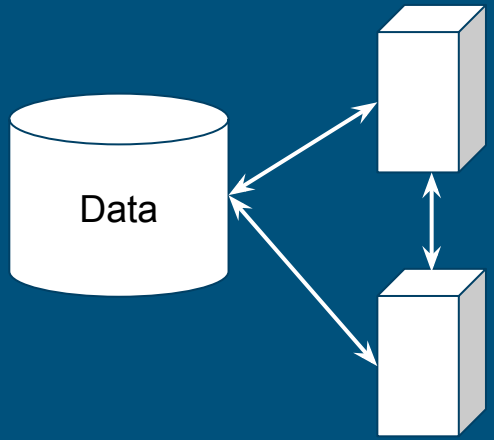
A short introduction to the architecture

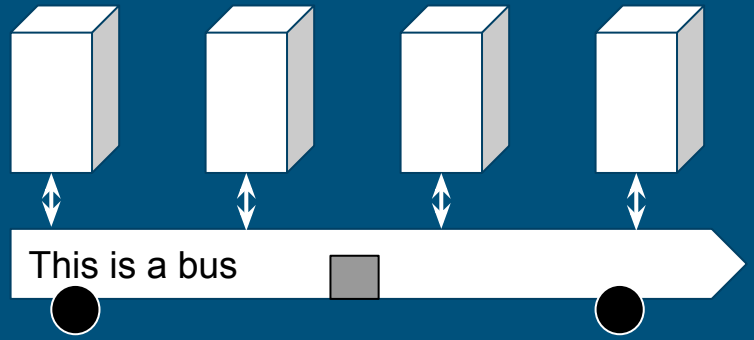
Hannes Rabo & Julius Celik

Events?



Events?





One step further - Event sourcing

- The **source** of information

Example of Event sourcing (Bank)

Julius	110kr
Hannes	70kr
...	...

Julius	+10	110
Julius	+90	100
Hannes	-5	70
Julius	+10	10
Hannes	+75	75

Major advantages

- No information is lost
- Ease of use
- Replayability

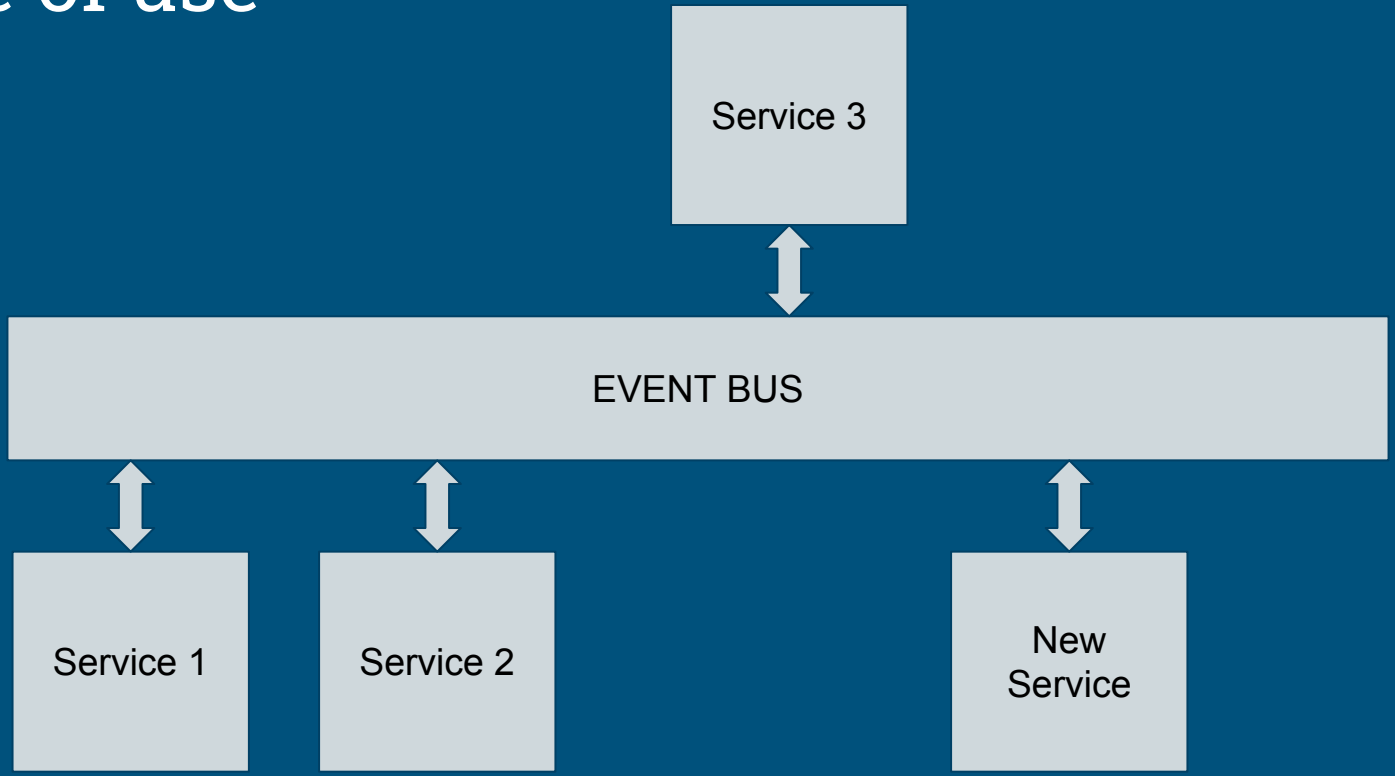
No information is lost

“Could you tell me how many people added an item to their cart, then removed it, then bought that item a month later?”

No information is lost

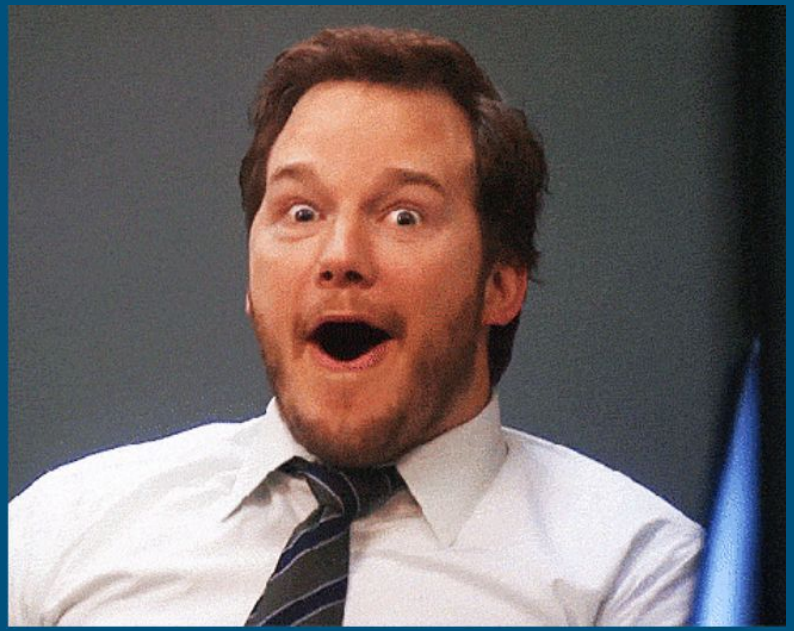
“Could you tell me how many people added an item to their cart, then removed it, then bought that item a month later?”

Ease of use



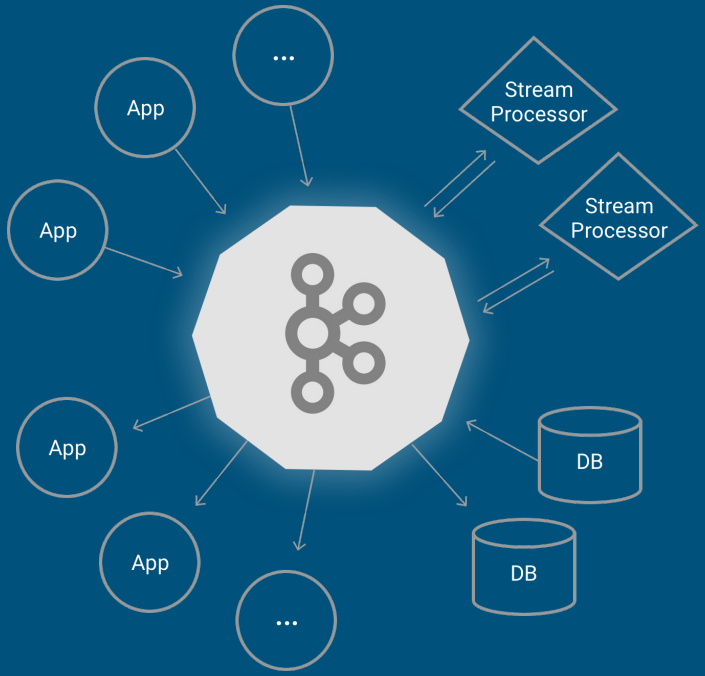
Replayability

Replayability



In practice!

- Distributed transaction log like Kafka
- Frameworks like AxonIQ and prooph
- Regular SQL Databases



The dark side

The dark side

Some people when confronted with a problem think: “I know, i will just break it down and create some nicely contained microservices”, now they have 15 problems located on servers they never even knew existed.

Don’t

(The curse of) distributed systems

- Extreme case of micro services

Data lake

Current Lake

Data lake (overflow)

Proposed Lake


Deleting the lake?

Take home message

Think not in terms of system state, but system events, and you will be able to be prepared for any future changes.