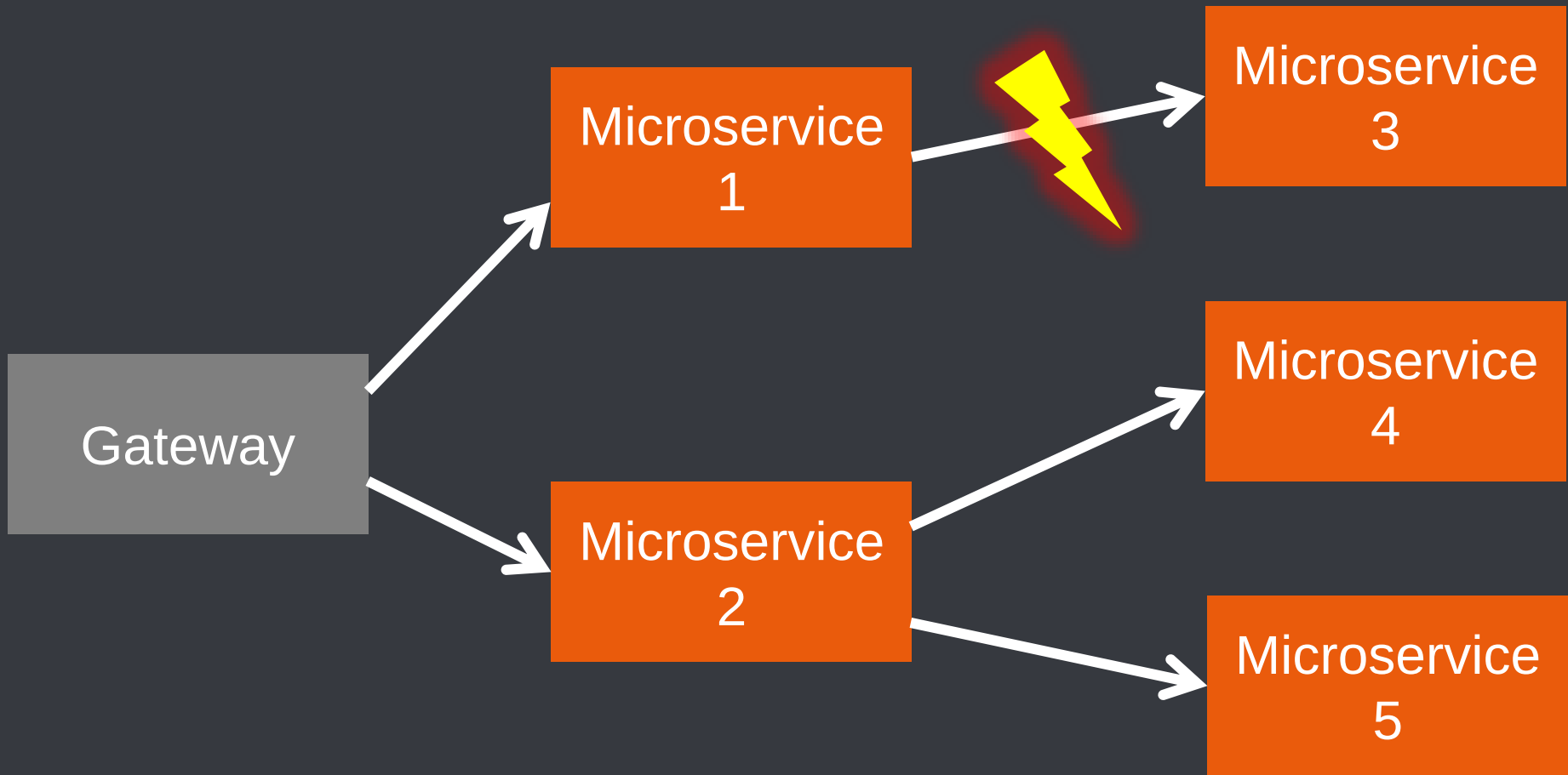


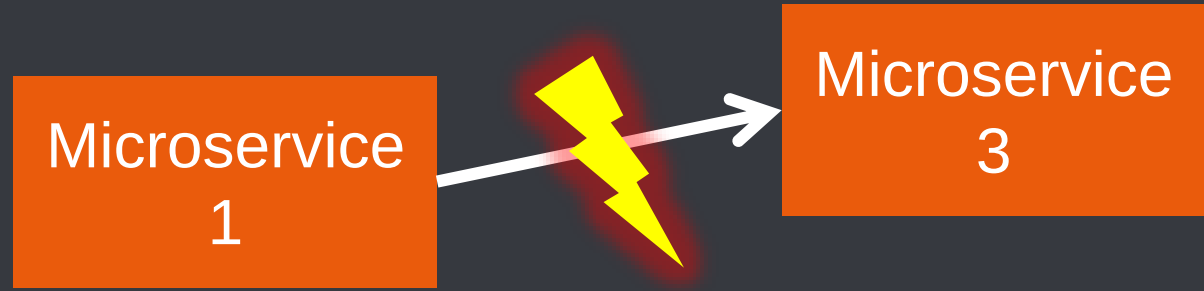


Bild: Sebastian Gauder

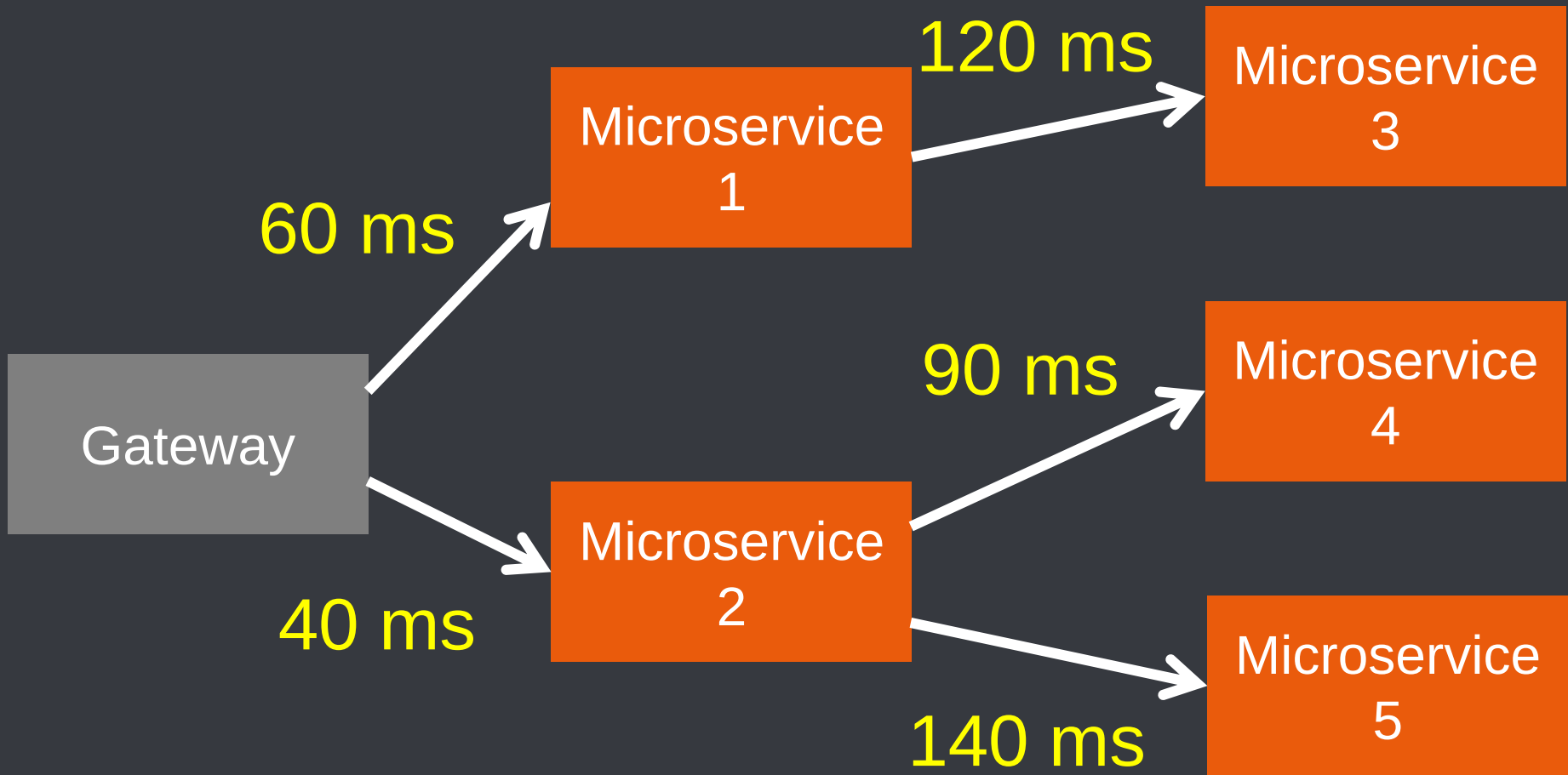
Sebastian Gauder REWE Digital

[youtube.com/watch?
v=1-BaWhwyN00](https://youtube.com/watch?v=1-BaWhwyN00)

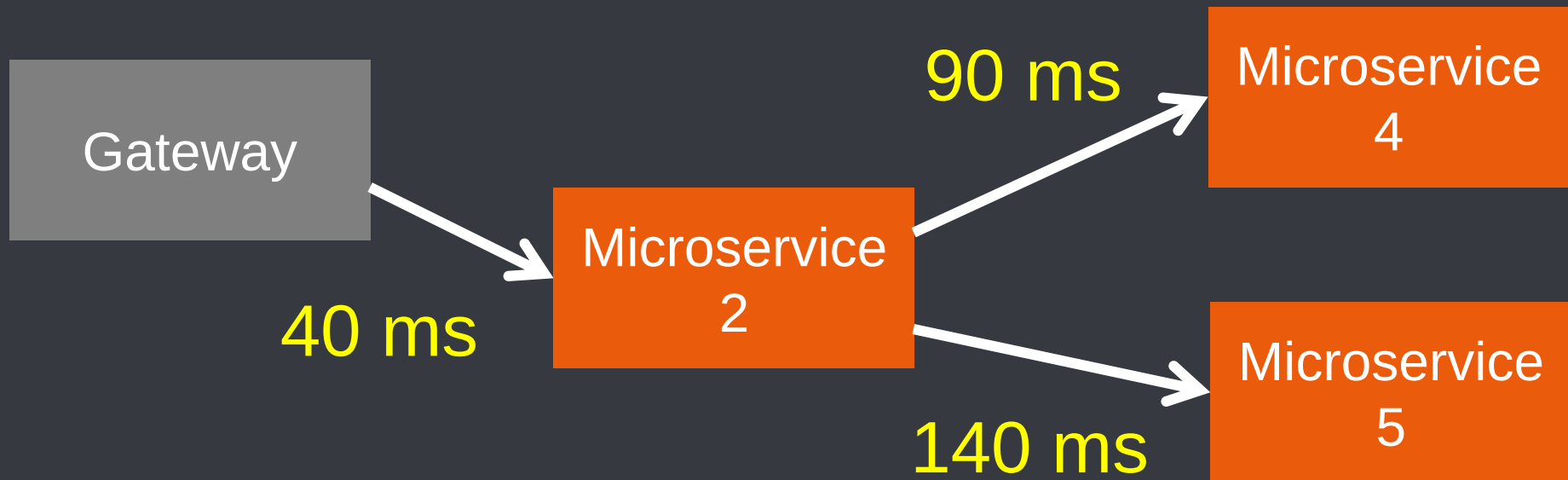




- Timeouts
- Retries
- Circuit Breaker

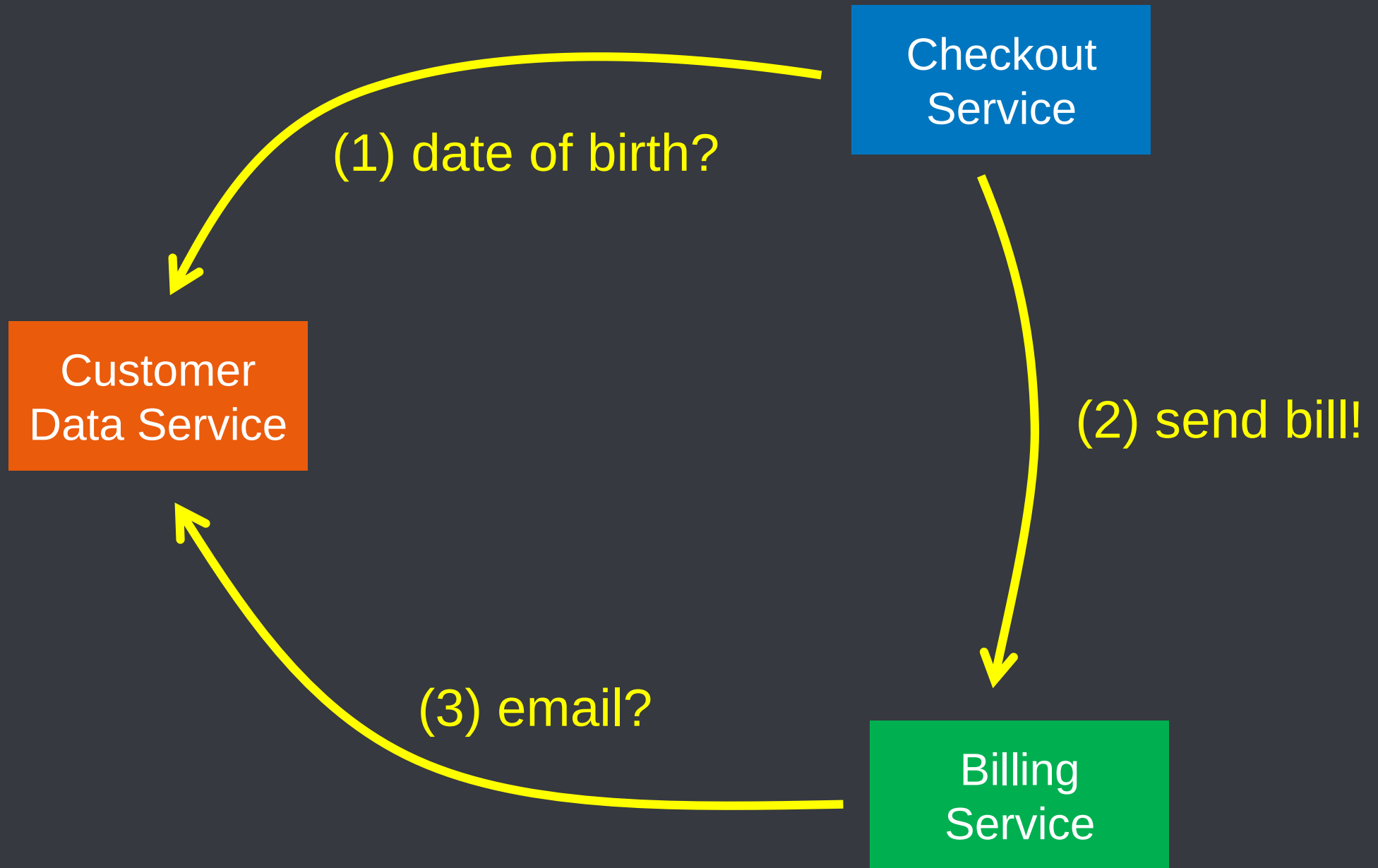


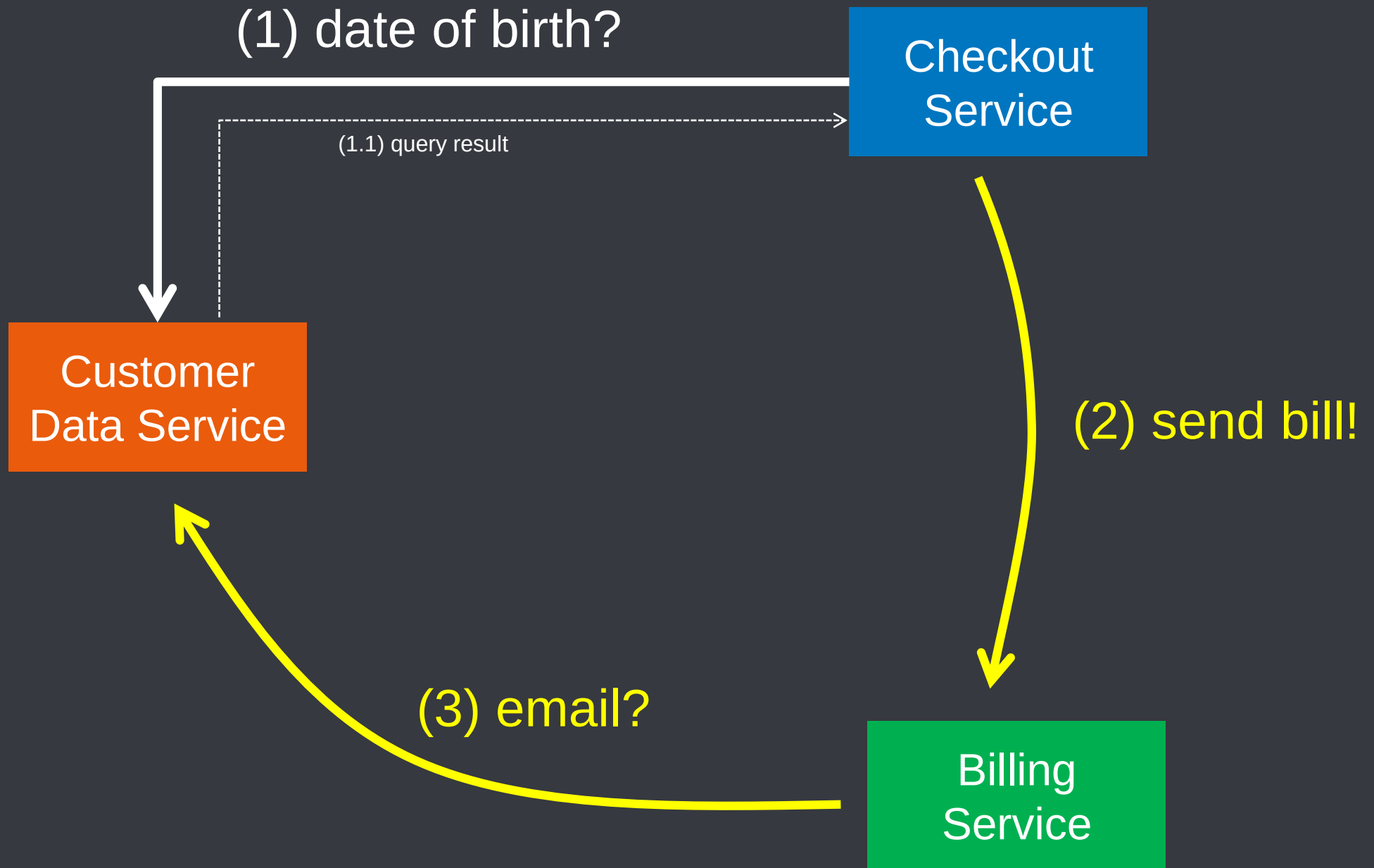
- Caching
- Parallelization

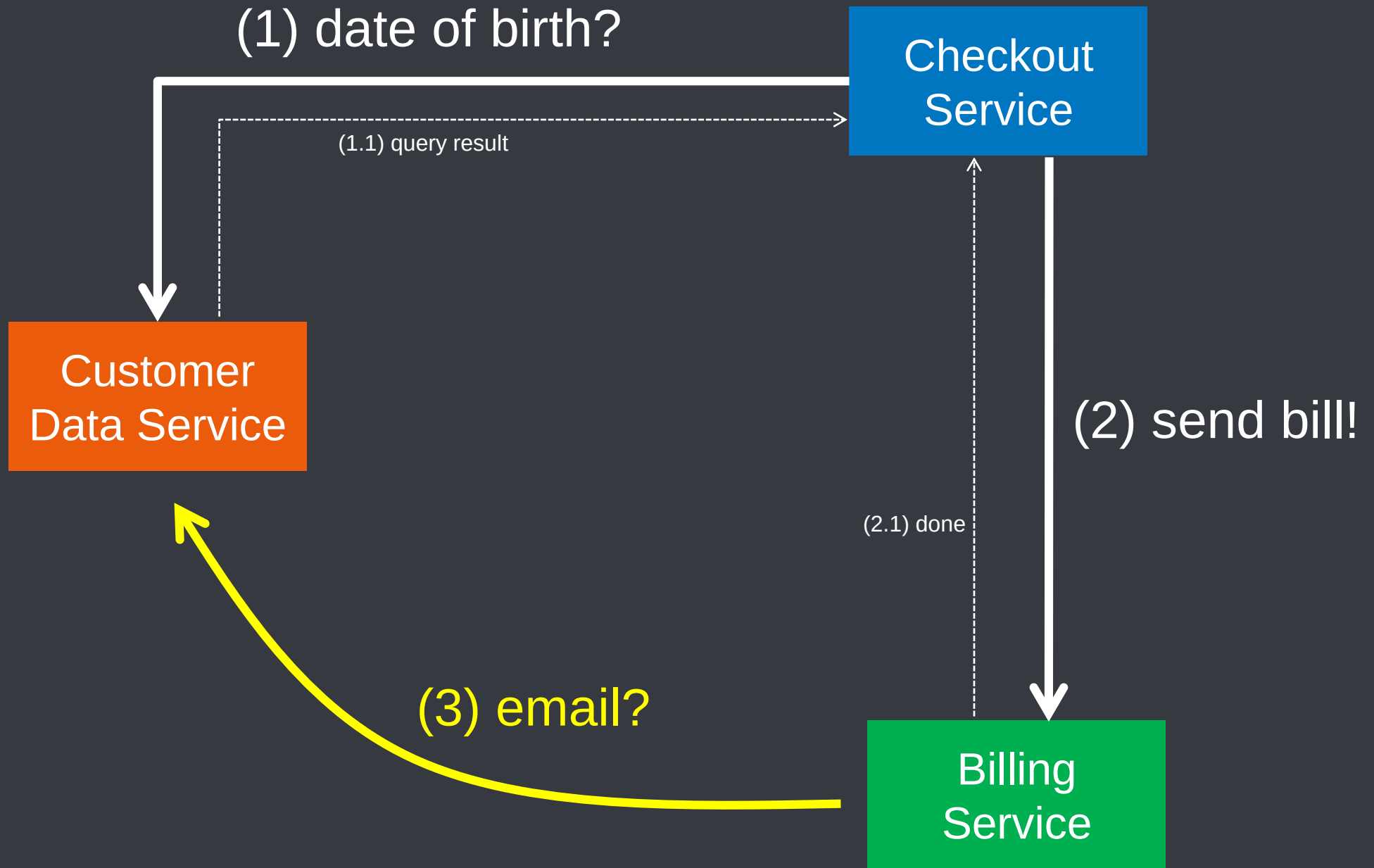


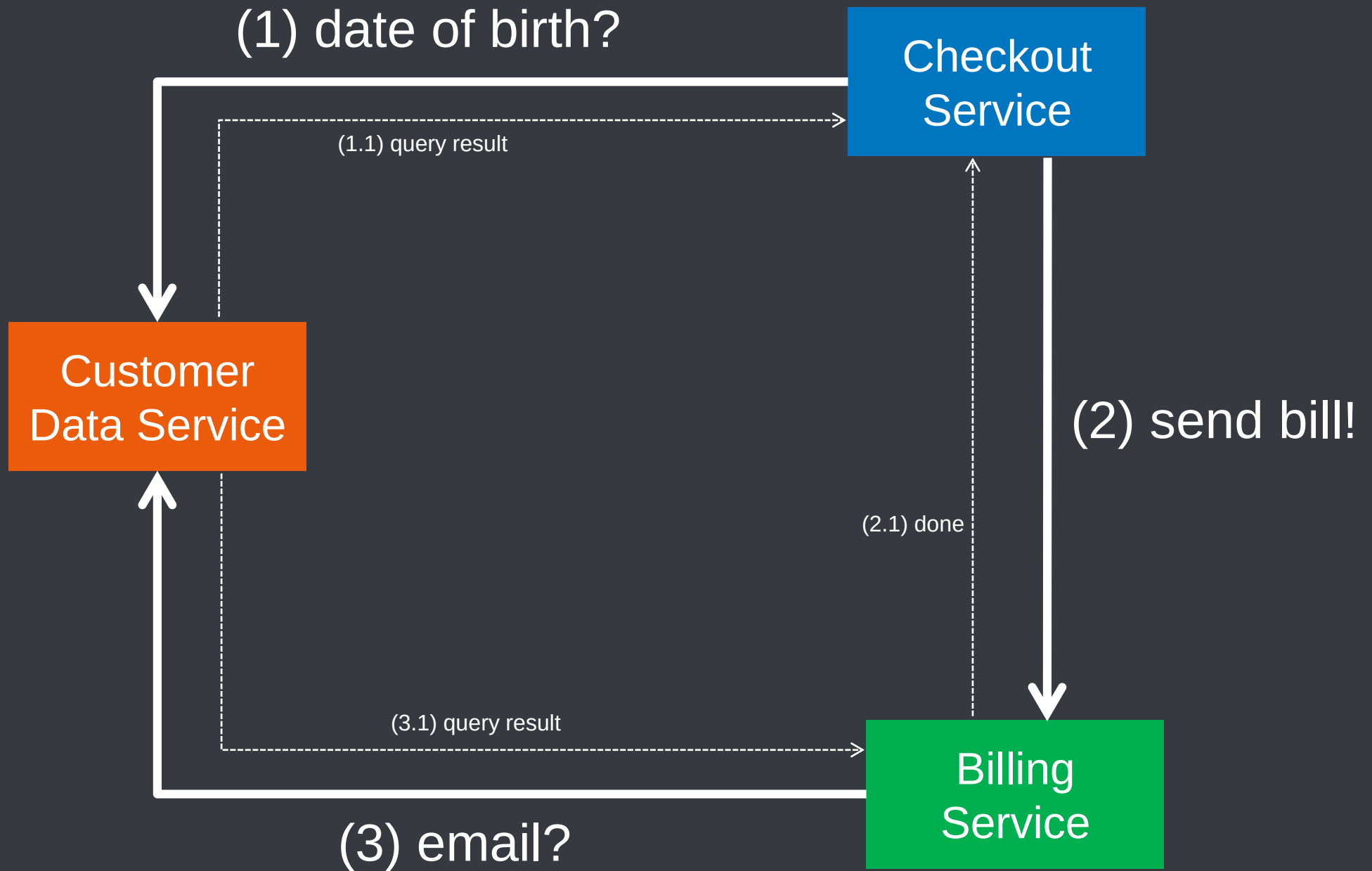
REWE digital

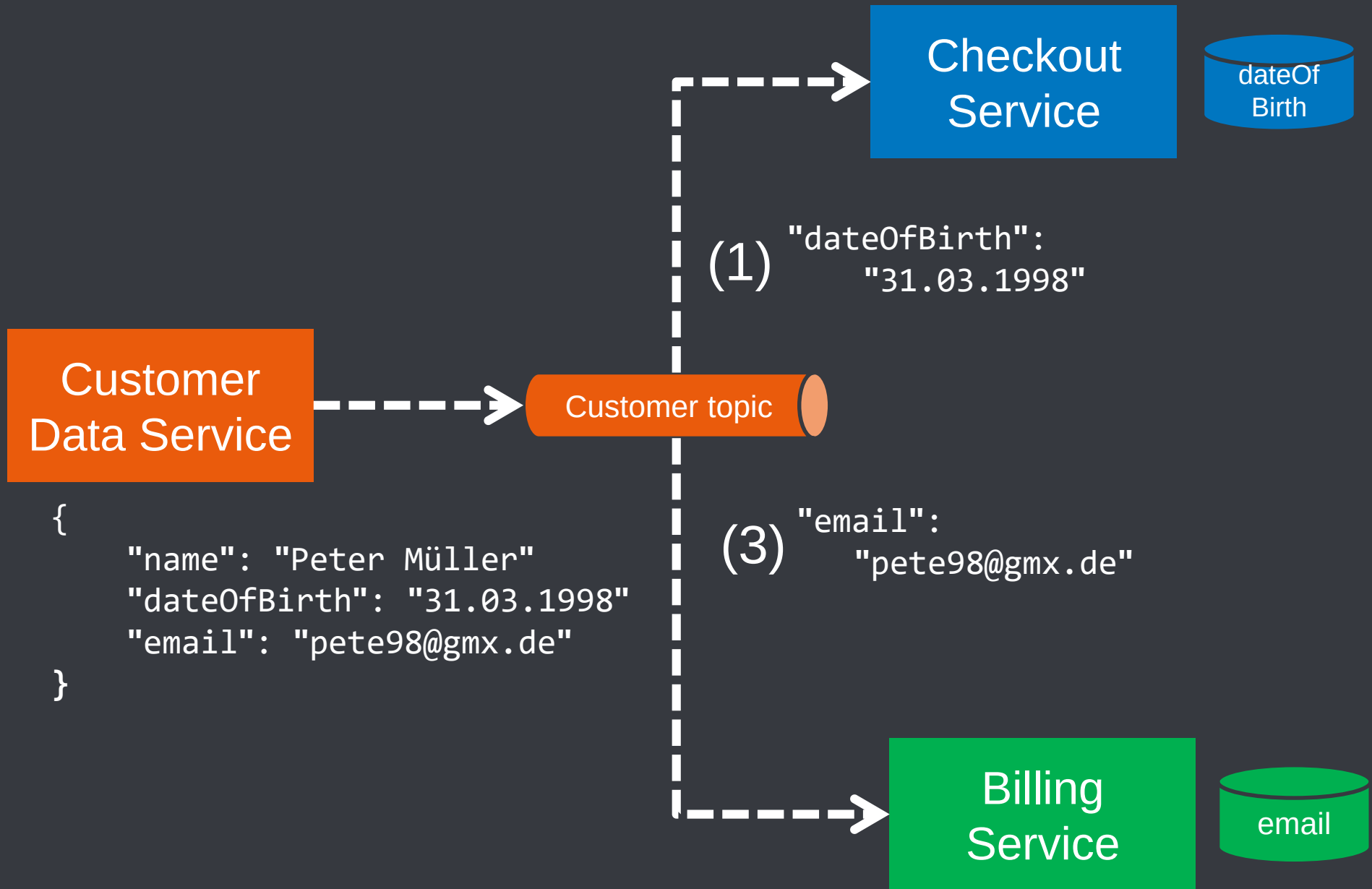
having data
is better than
needing data











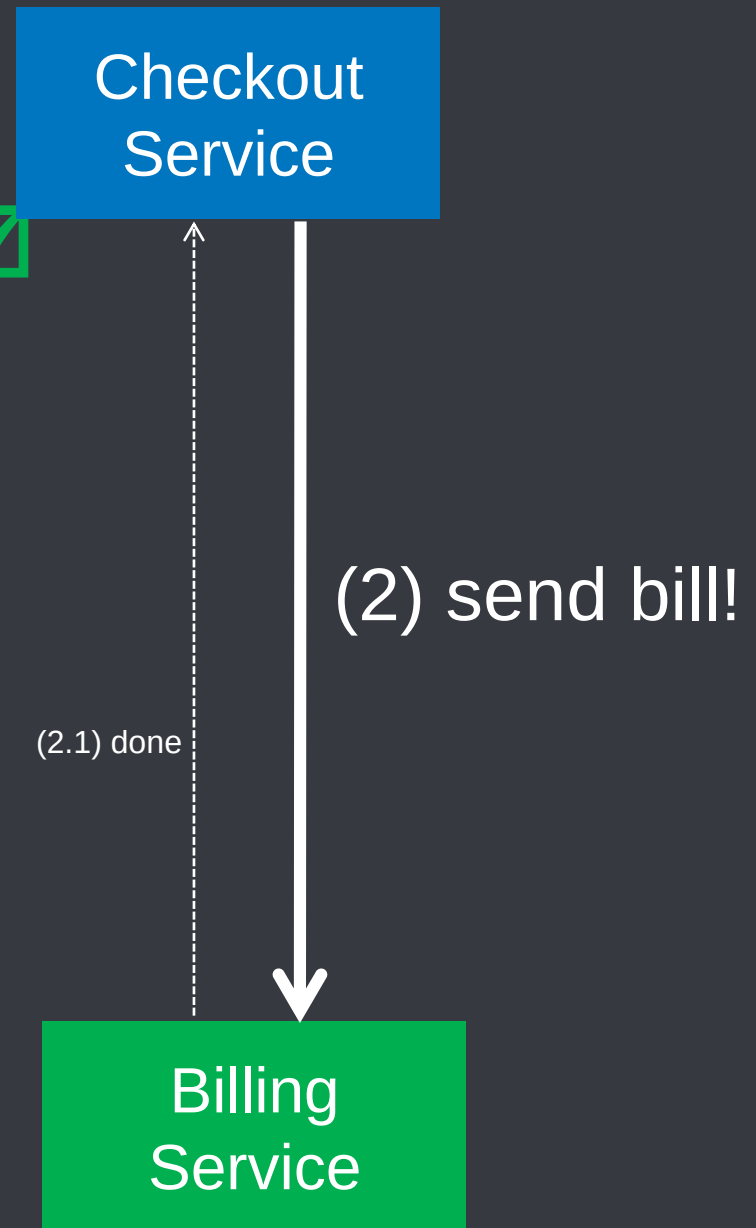
Queries vs. Commands

- Queries: asynchronous ☒

- Commands?

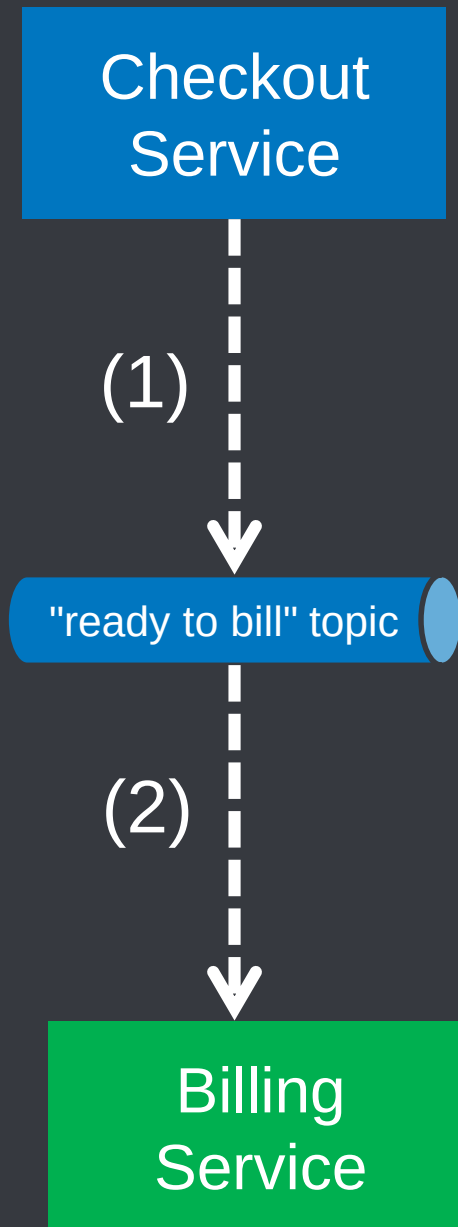
- Synchronous:

- + Acknowledgement
- + Fail & retry simple
- - tight coupling
- - quickly turns into „spaghetti“



Asynchrone Commands

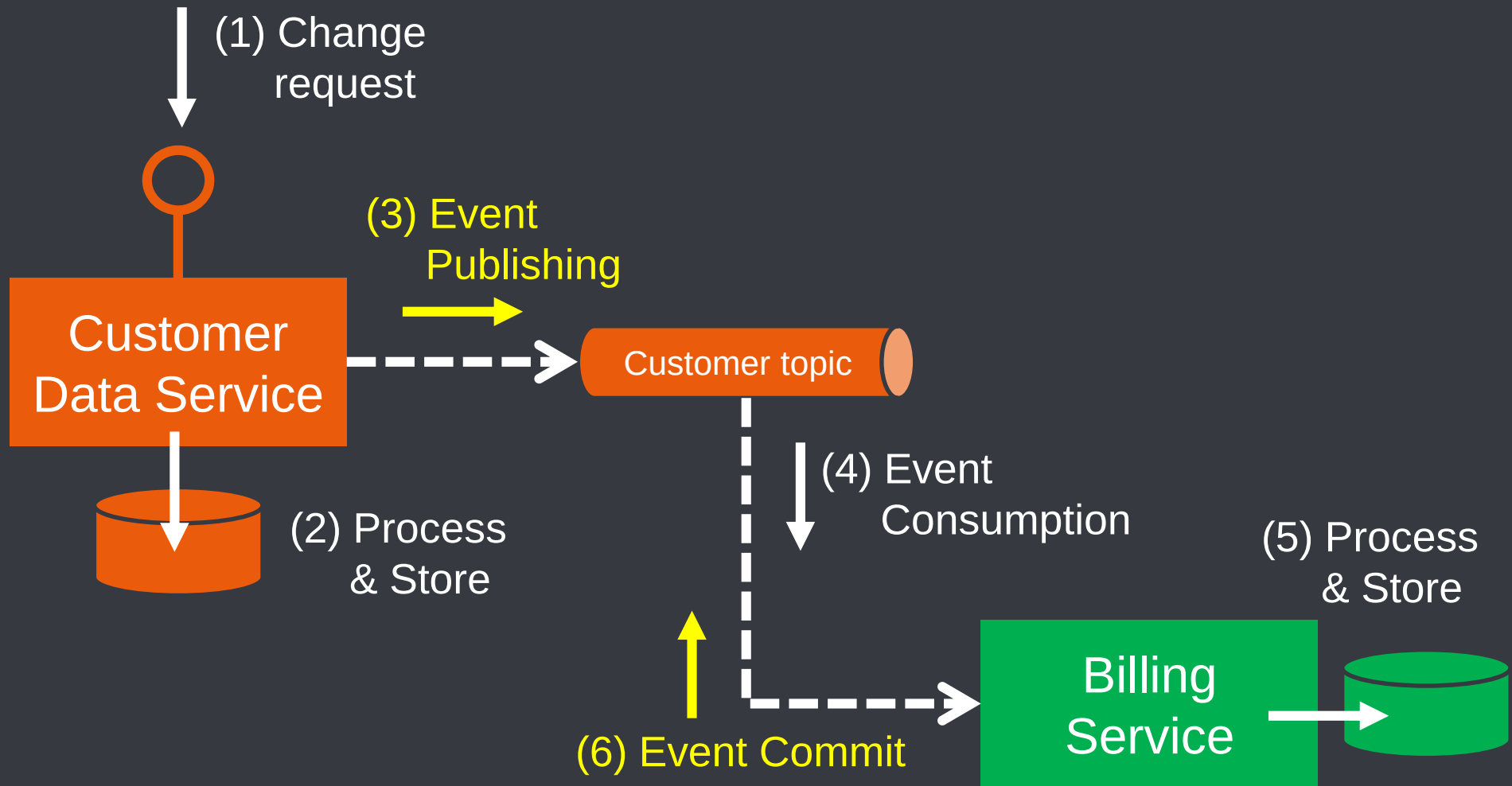
- + loose coupling
- + consistent communication
- – „Fail“ only obvious by monitoring
- – Acknowledgement needs own event
- – Orchestration of overall process intransparent



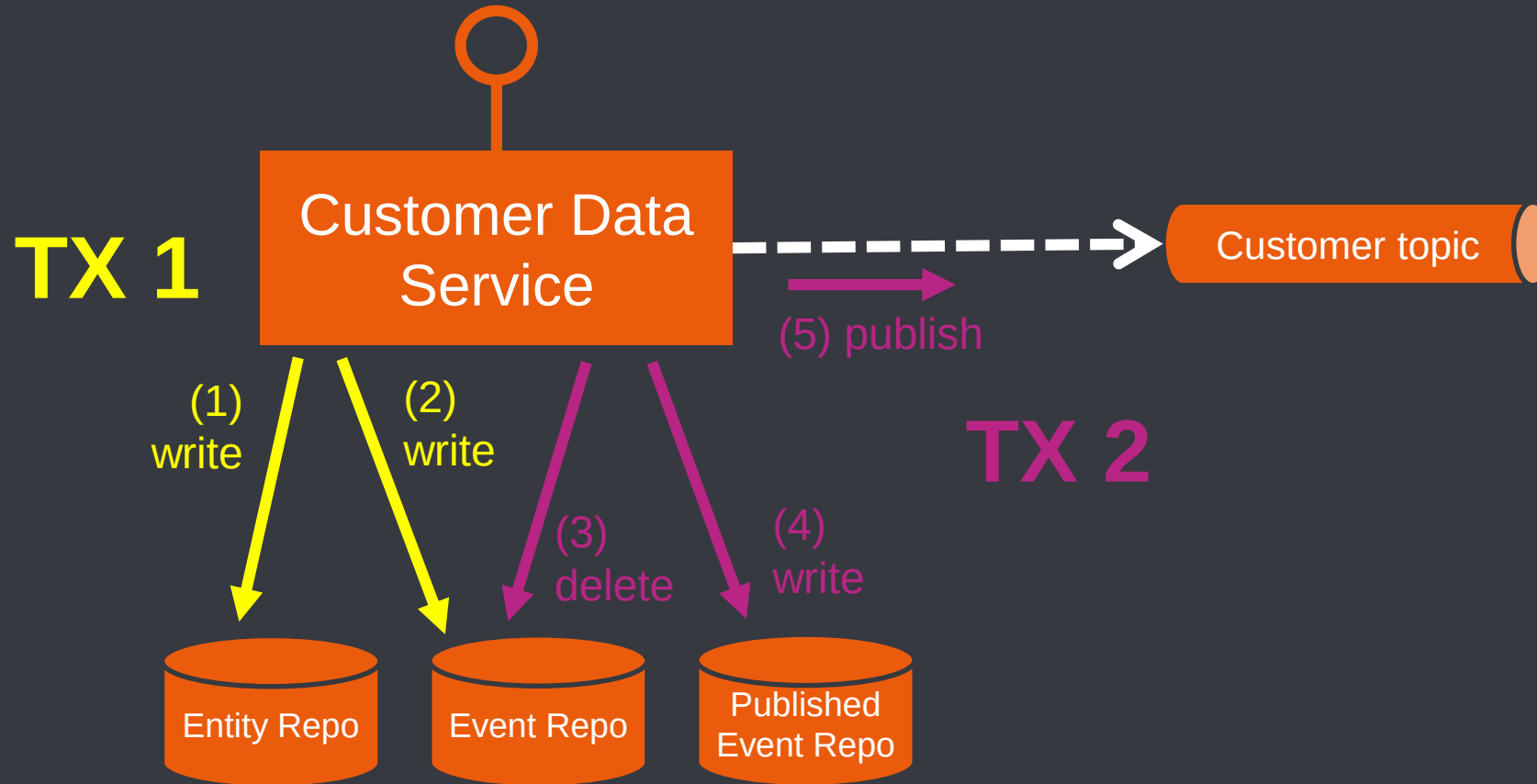
Rules for Events

1. Events = True Facts
2. No CRUD!
3. One dedicated owner per topic

1. Events = True Facts



REWE Digital Best Practice



2. No CRUD!

~~customer_
created~~

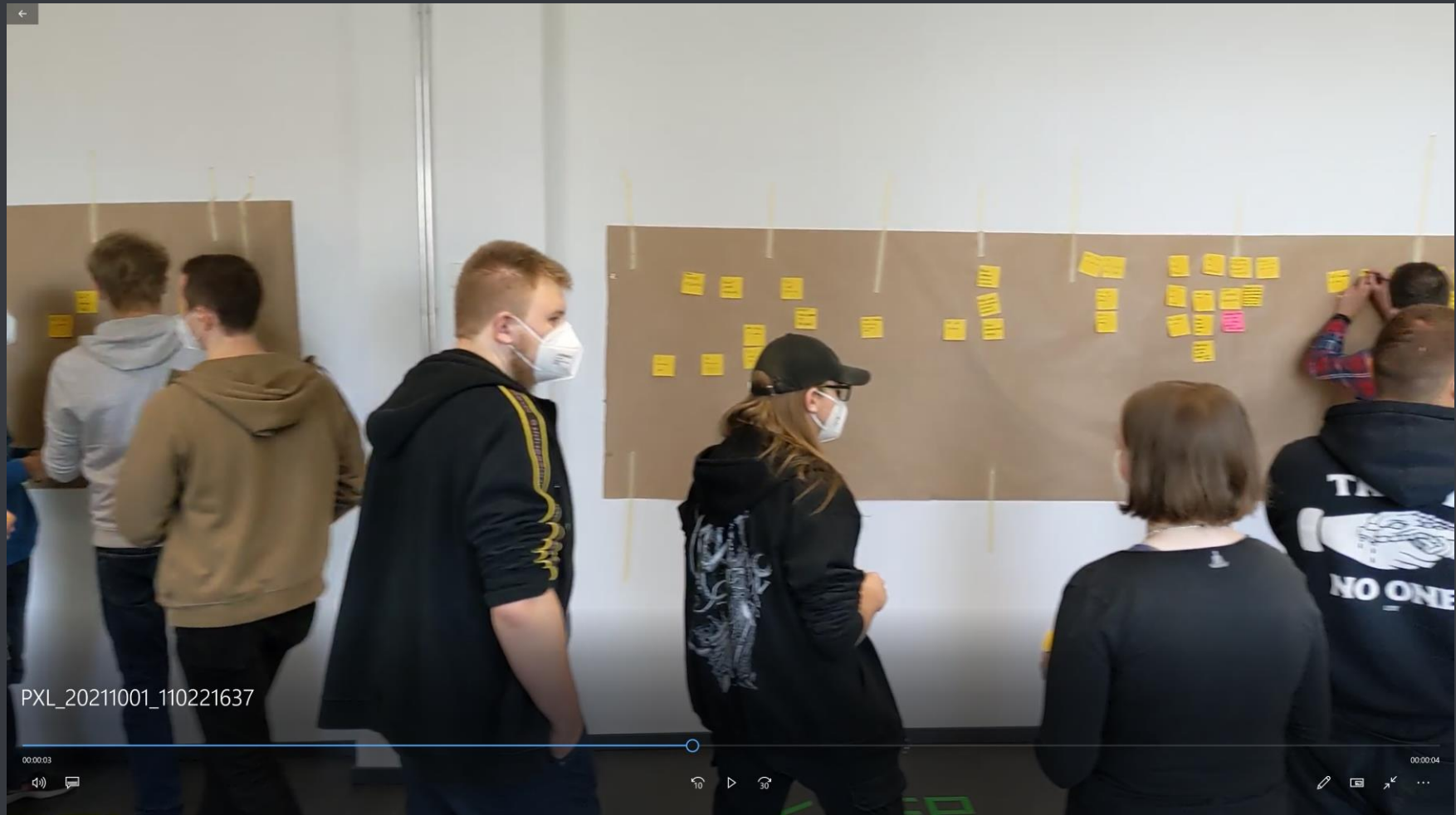
~~customer_
deleted~~

account_
registered_
by_customer

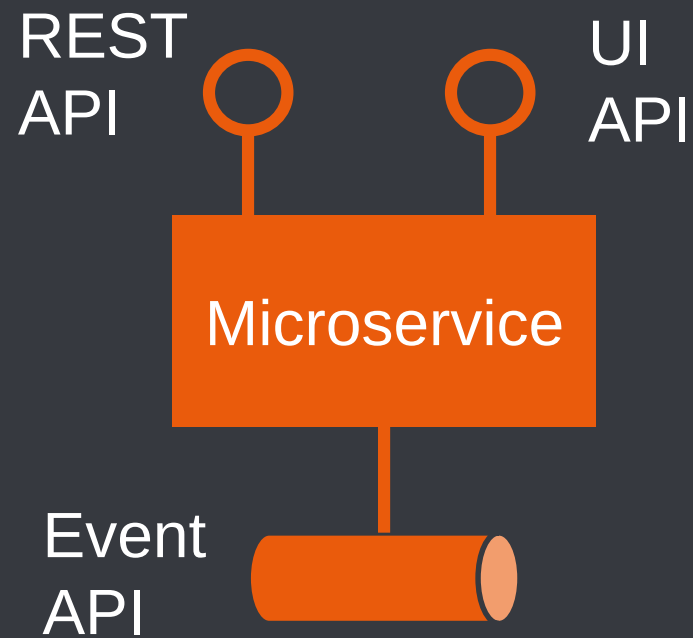
account_
temporarily_
blocked

account_
cancelled_
by_customer

Source: e.g. Event Storming



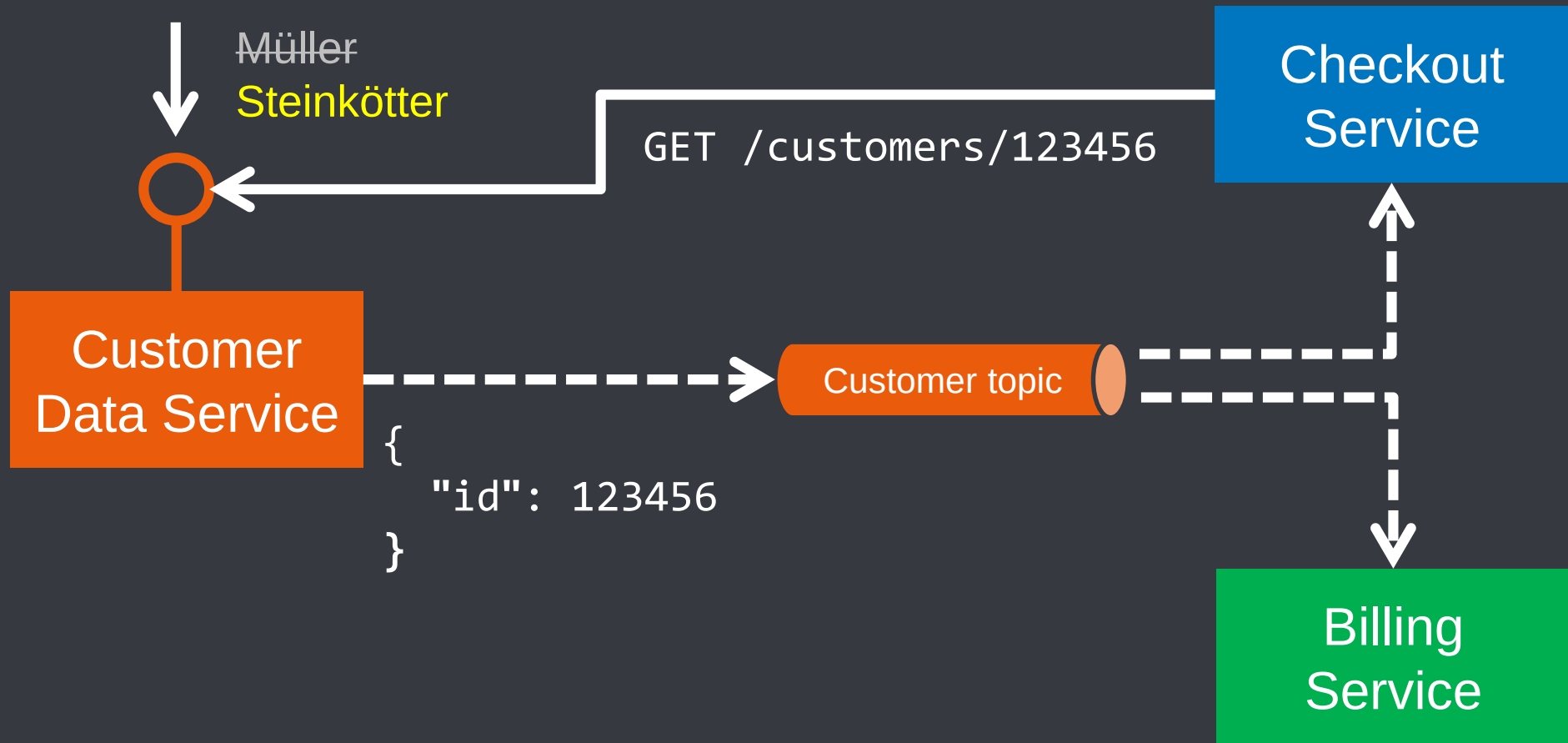
3. One owner per topic



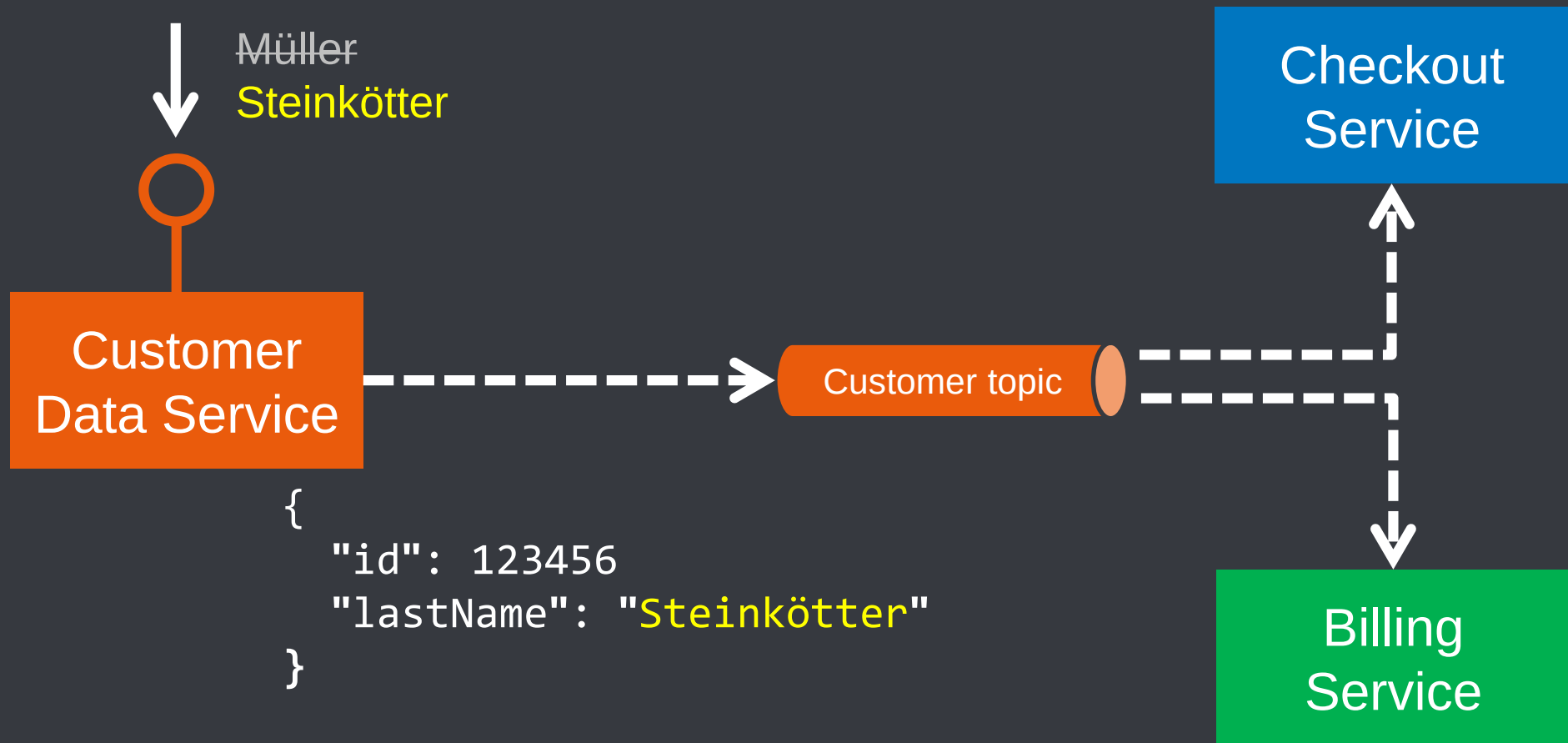
3 Options for payload

1. Only ID, with REST call
2. Event Sourcing (just delta)
3. Event-carried State Transfer
(full payload)

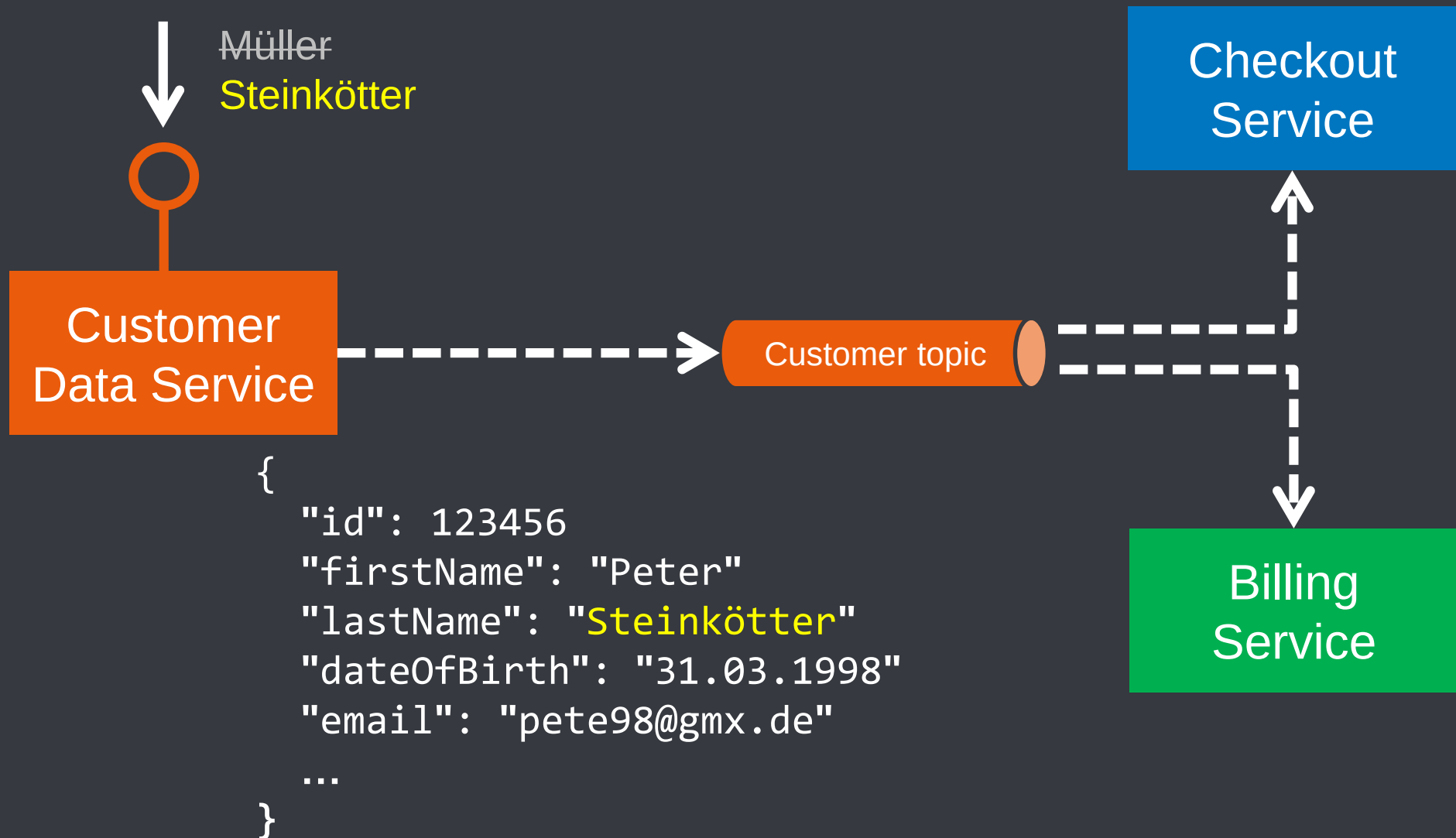
1. Only ID, with REST call



2. Event Sourcing (just delta)



3. Full Payload



When to use what?

1. Only ID, with REST call

- Many events, rarely relevant for business logic

2. Event Sourcing (just delta)

- Big aggregates, many events

3. ECST - full payload

- Transactional completeness important
- Log compaction possible



- Full payload
- Log compaction
- Goal: Transactional Completeness
- Event = complete aggregate



www.archi-lab.io

Videoserie „Softwaretechnik“ - © 2021 Prof. Dr. Stefan Bente