

Zwischen ACID, Eventual und Chaos

Konsistenz in verteilten Systemen

Frank Scheffler
Senior Solution Architect

frank.scheffler@digitalfrontiers.de
github.com/maverick1601

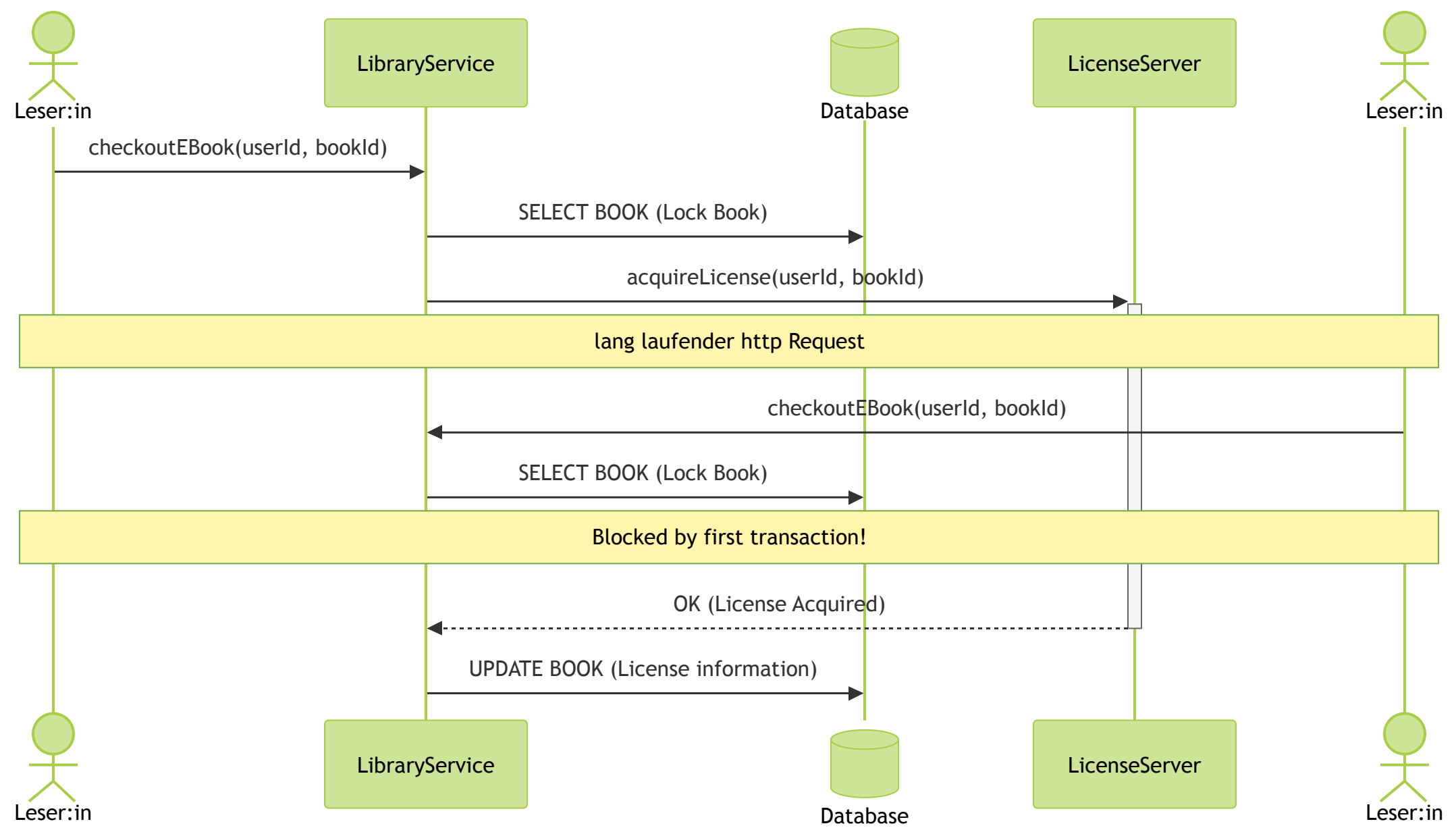
Nikolai Neugebauer
Principal Consultant

nikolai.neugebauer@digitalfrontiers.de
github.com/nik101010

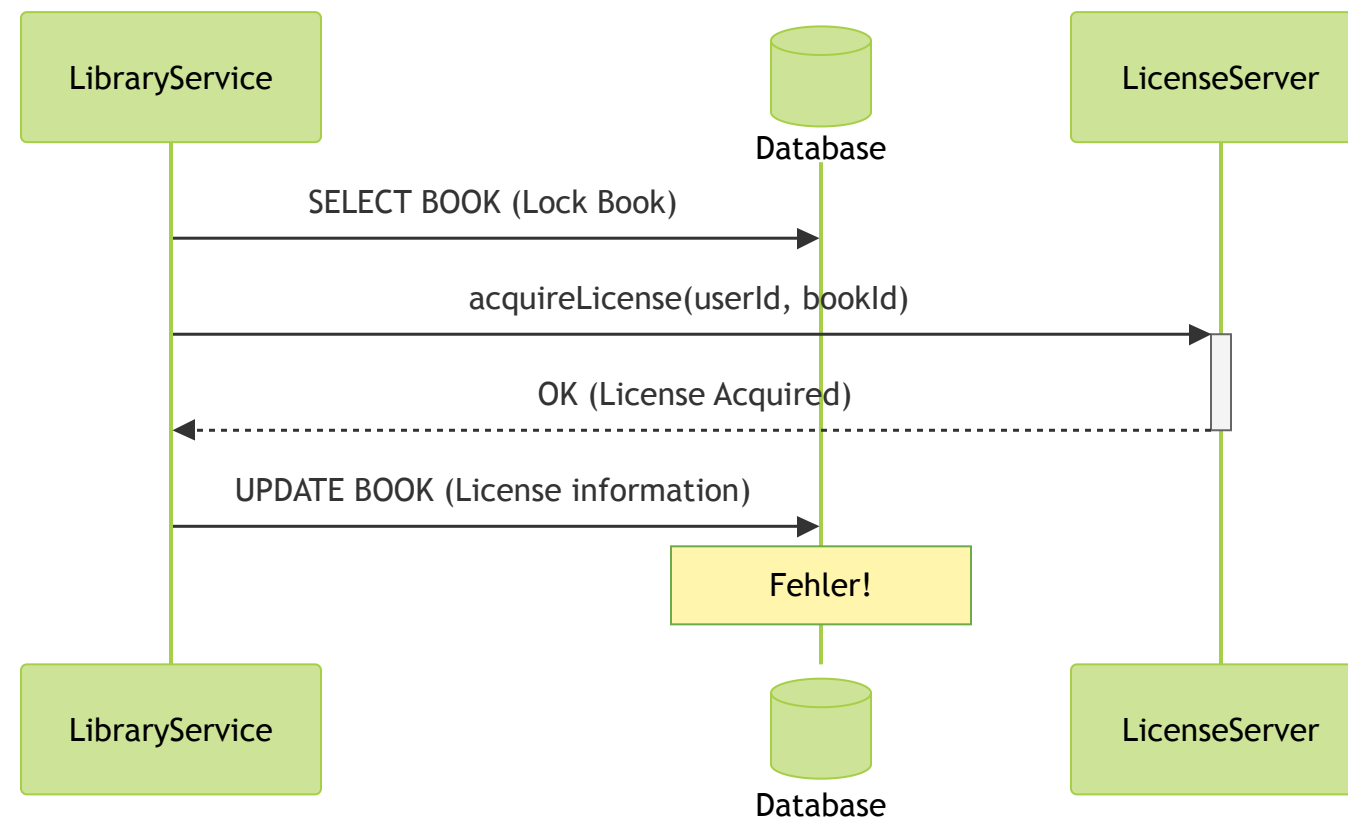
E-Book ausleihen

```
1 @Service
2 public class LibraryService {
3
4     private final BookRepository bookRepository;
5     private final LicenseServerClient licenseClient;
6     private final NotificationService notificationService;
7
8     public LibraryService(BookRepository bookRepository,
9                           LicenseServerClient licenseClient,
10                          NotificationService notificationService) {
11         this.bookRepository = bookRepository;
12         this.licenseClient = licenseClient;
13         this.notificationService = notificationService;
14     }
15
16     /**
17      * Leiht ein eBook an einen Leser aus.
18      */
19     @Transactional
20     public void checkoutEBook(String readerId, String bookId) {
21         Book book = bookRepository.findById(bookId)
22             .orElseThrow(() -> new BookNotFoundException(bookId));
23
24         if (book.licensesAvailable() == 0) {
25             // ...
26         }
27     }
28 }
```

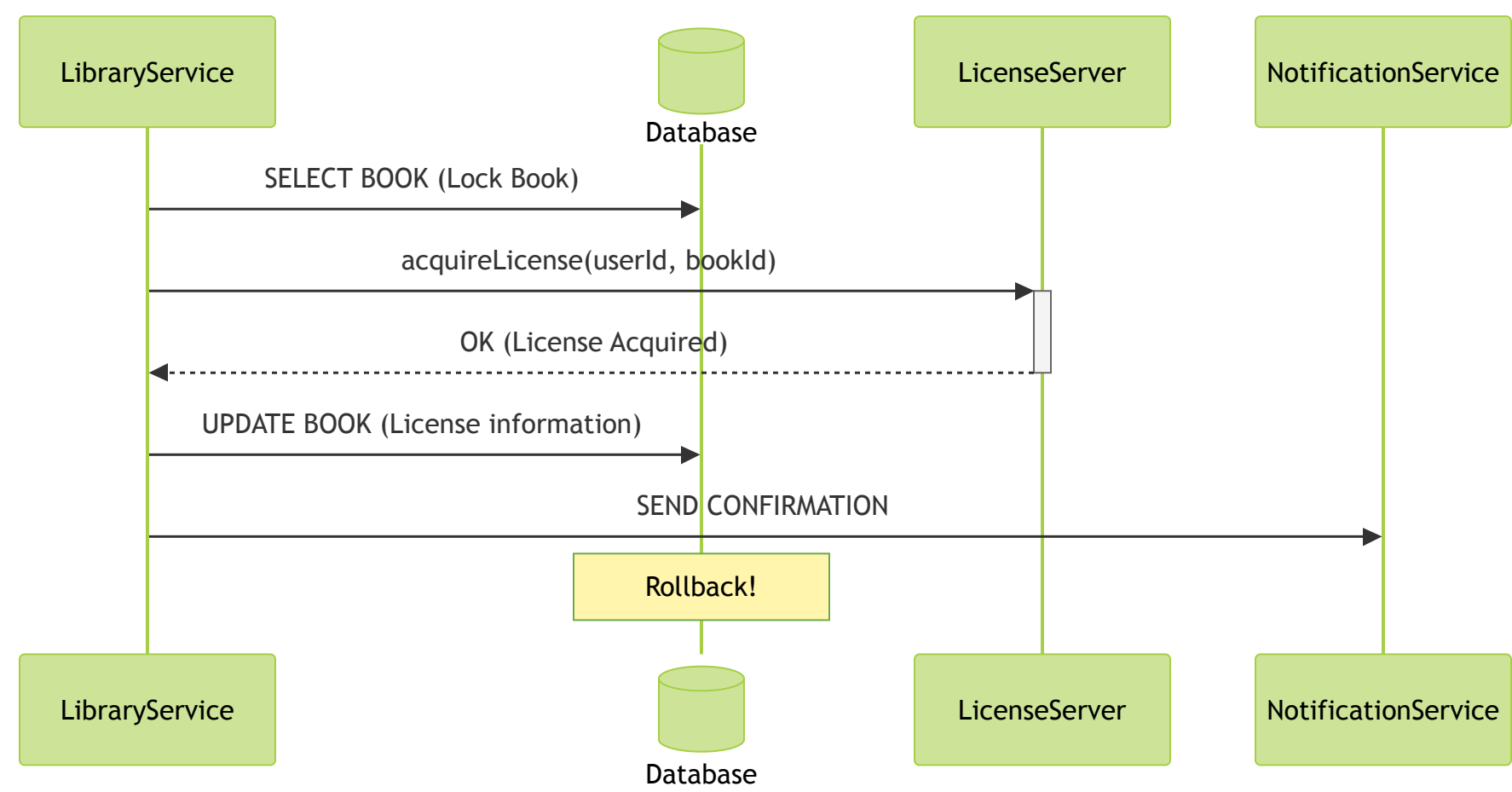
Wenn zwei sich streiten ...



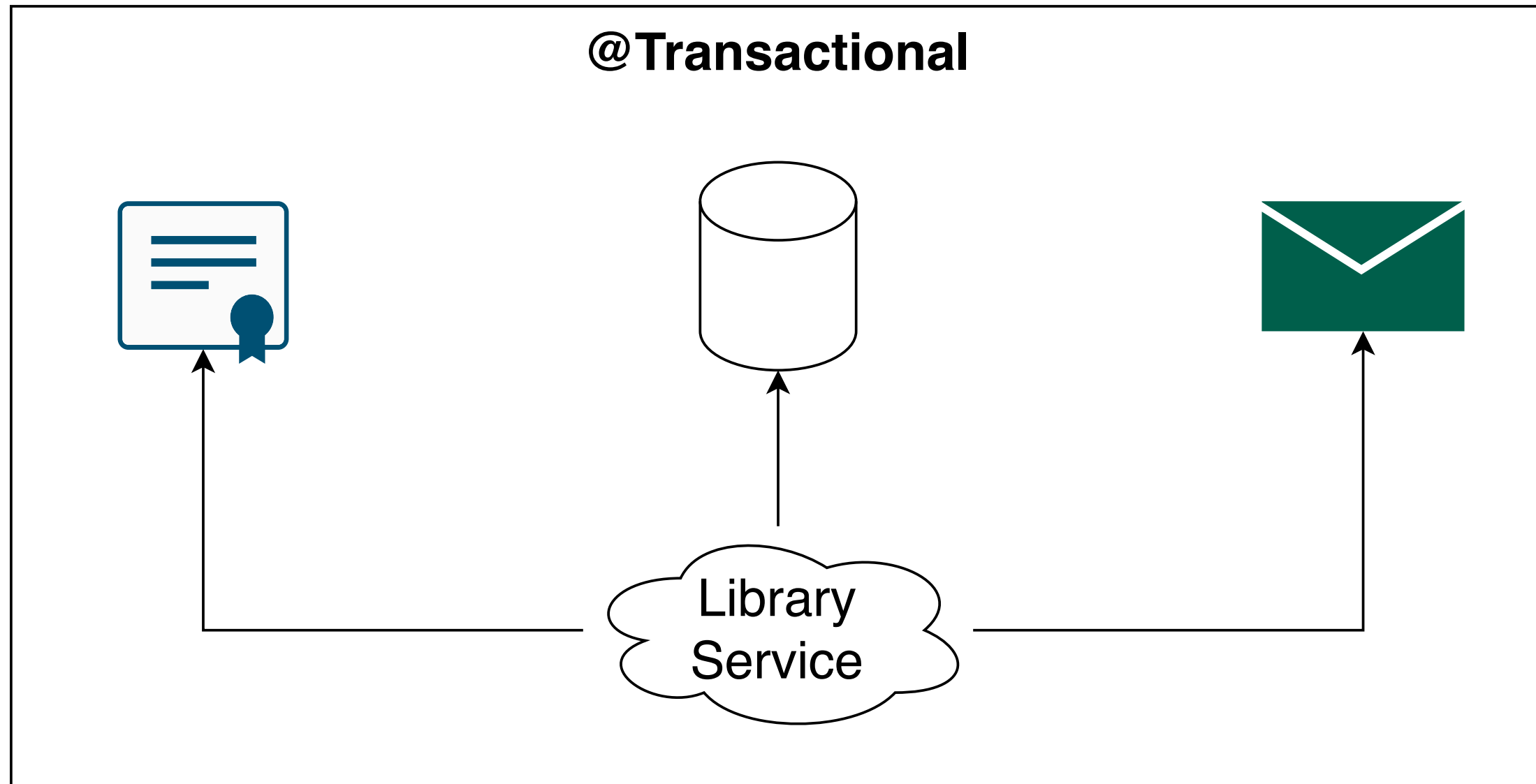
Die Lizenz zum Scheitern



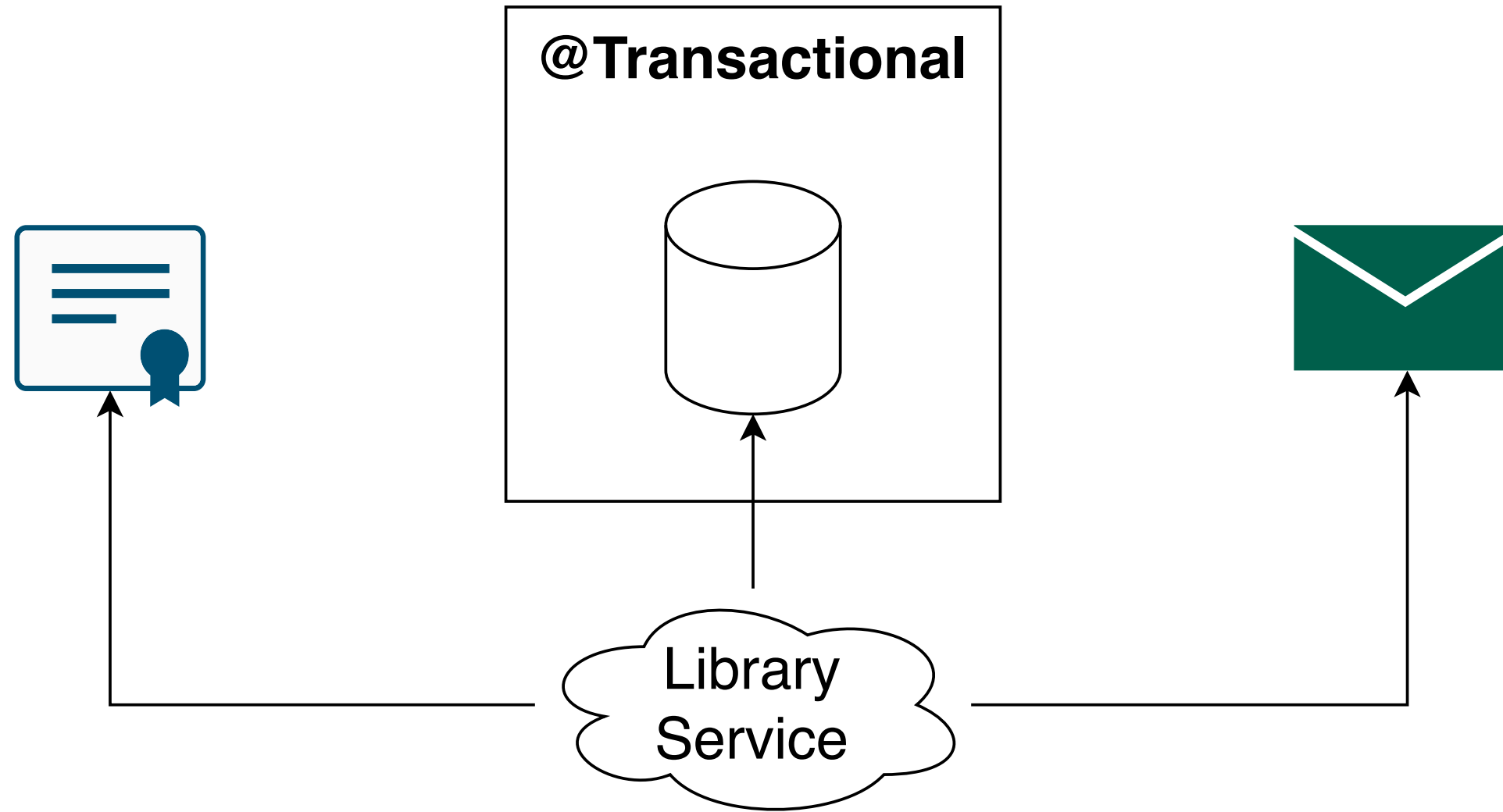
Schöne Grüße aus dem Nichts



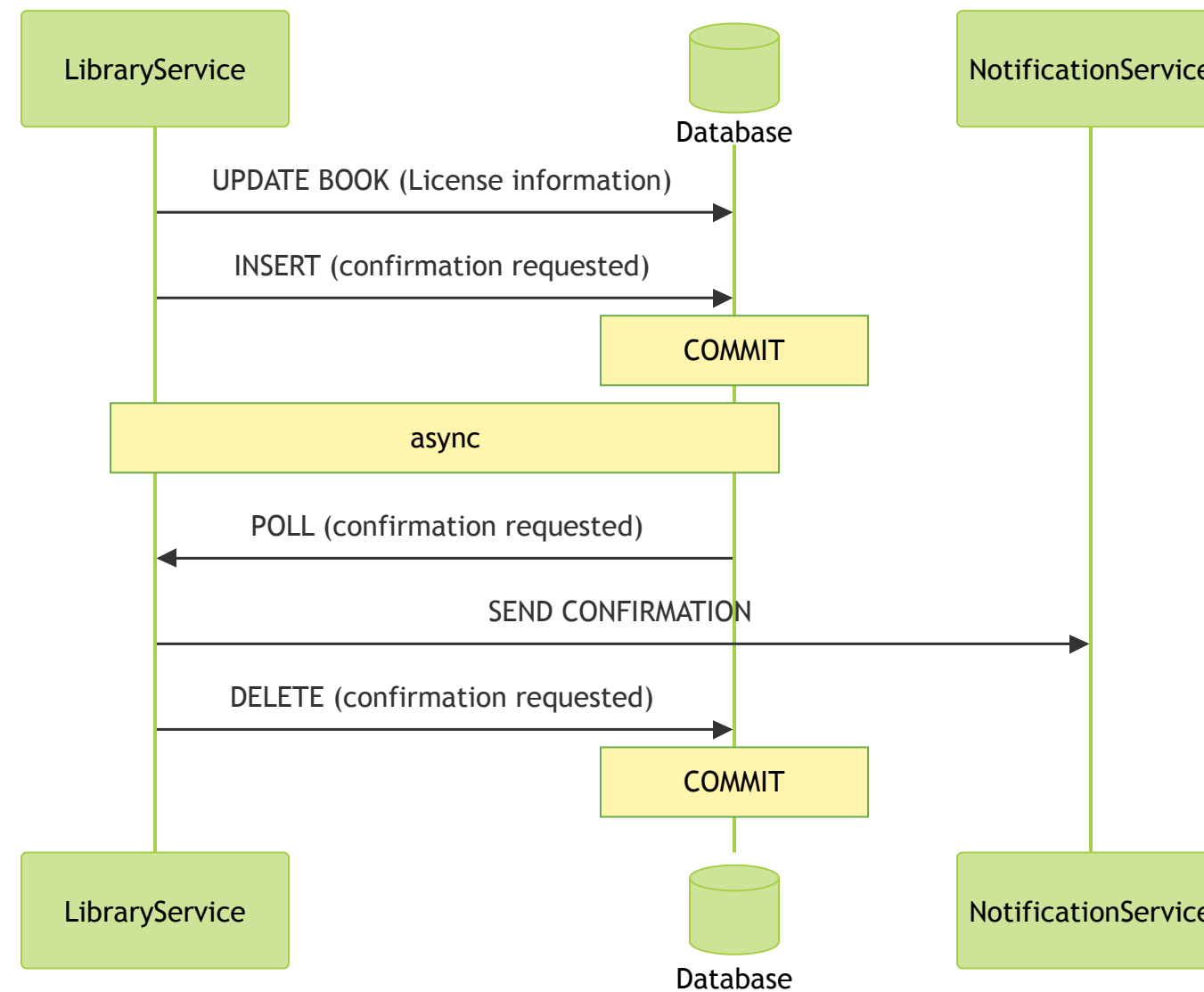
Lügen ...



... und Wahrheiten

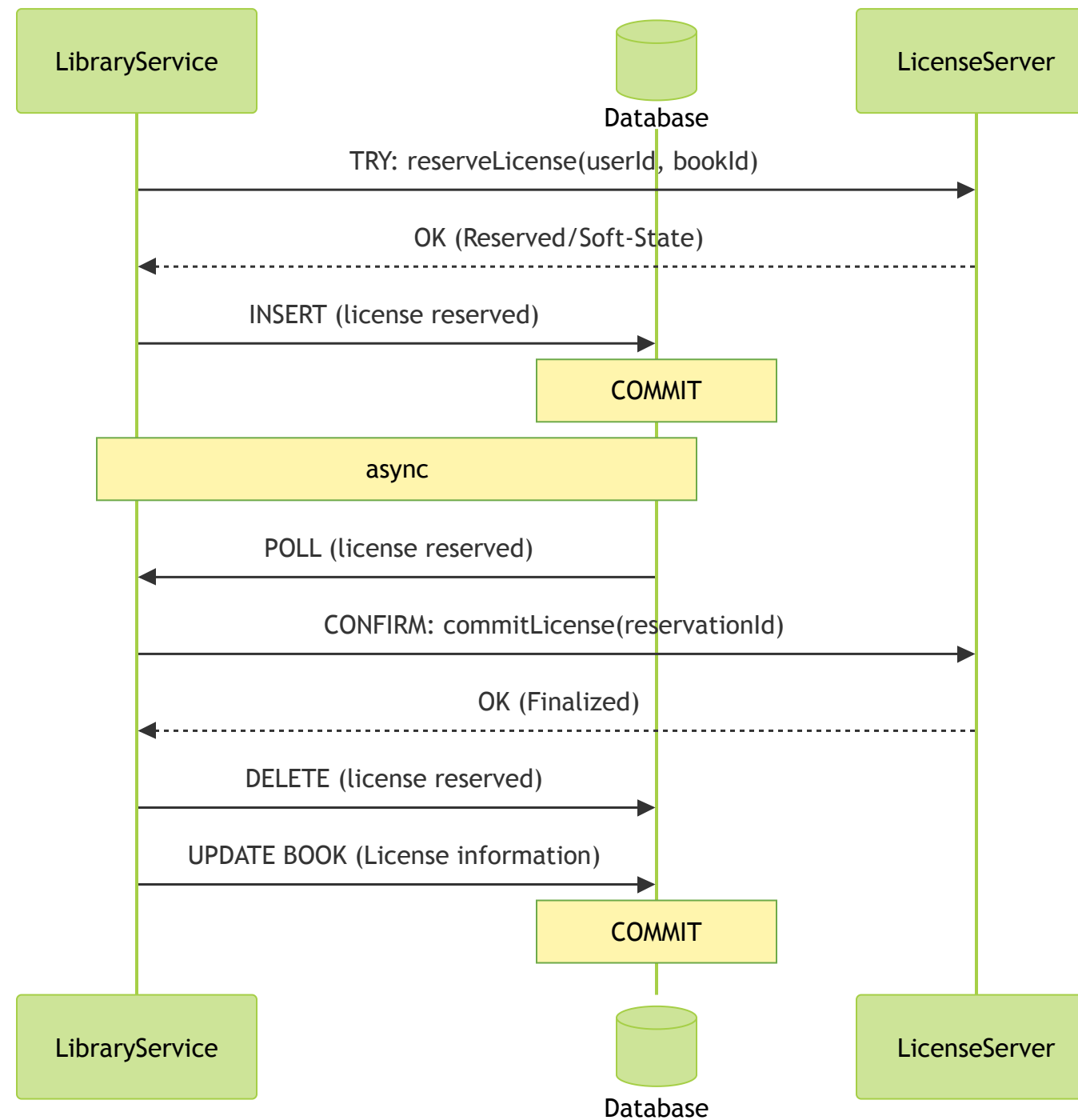


Transactional Outbox Pattern

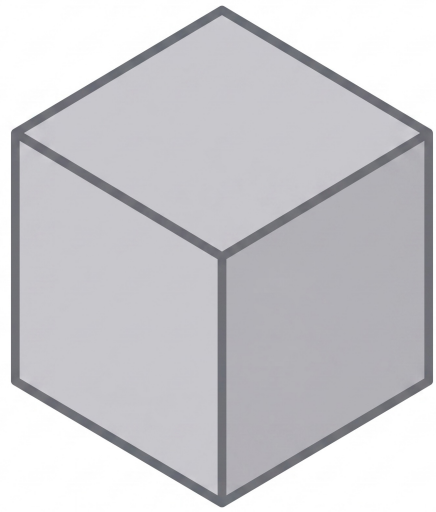


At Least Once Delivery

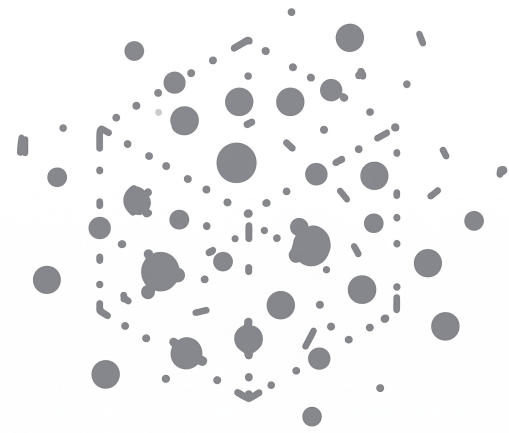
Reservation Pattern



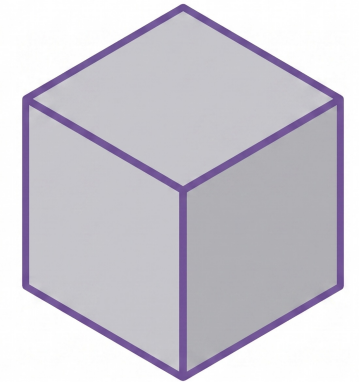
Von ACID zu Eventual Consistency



Konsistenter Zustand

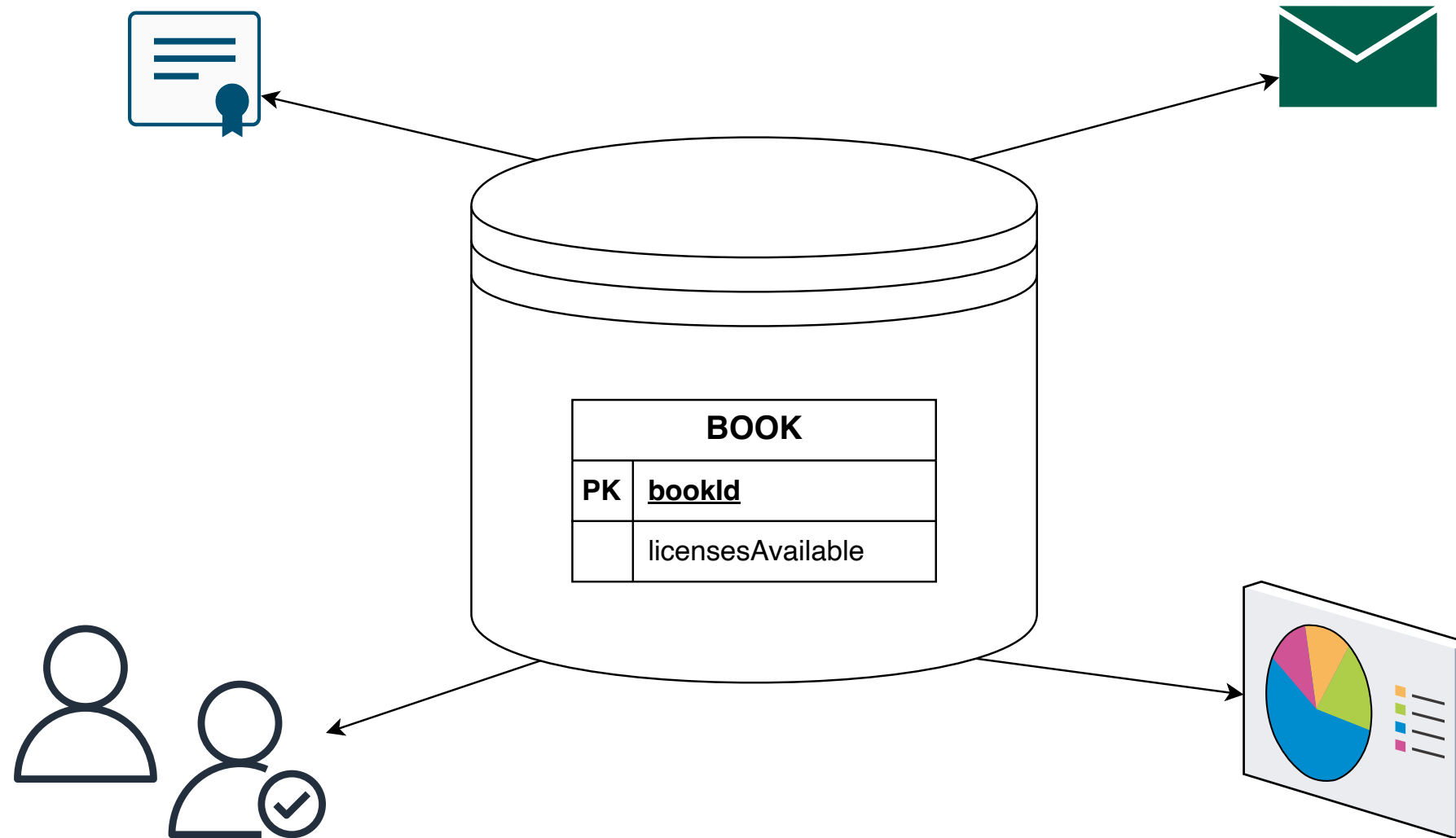


Soft State



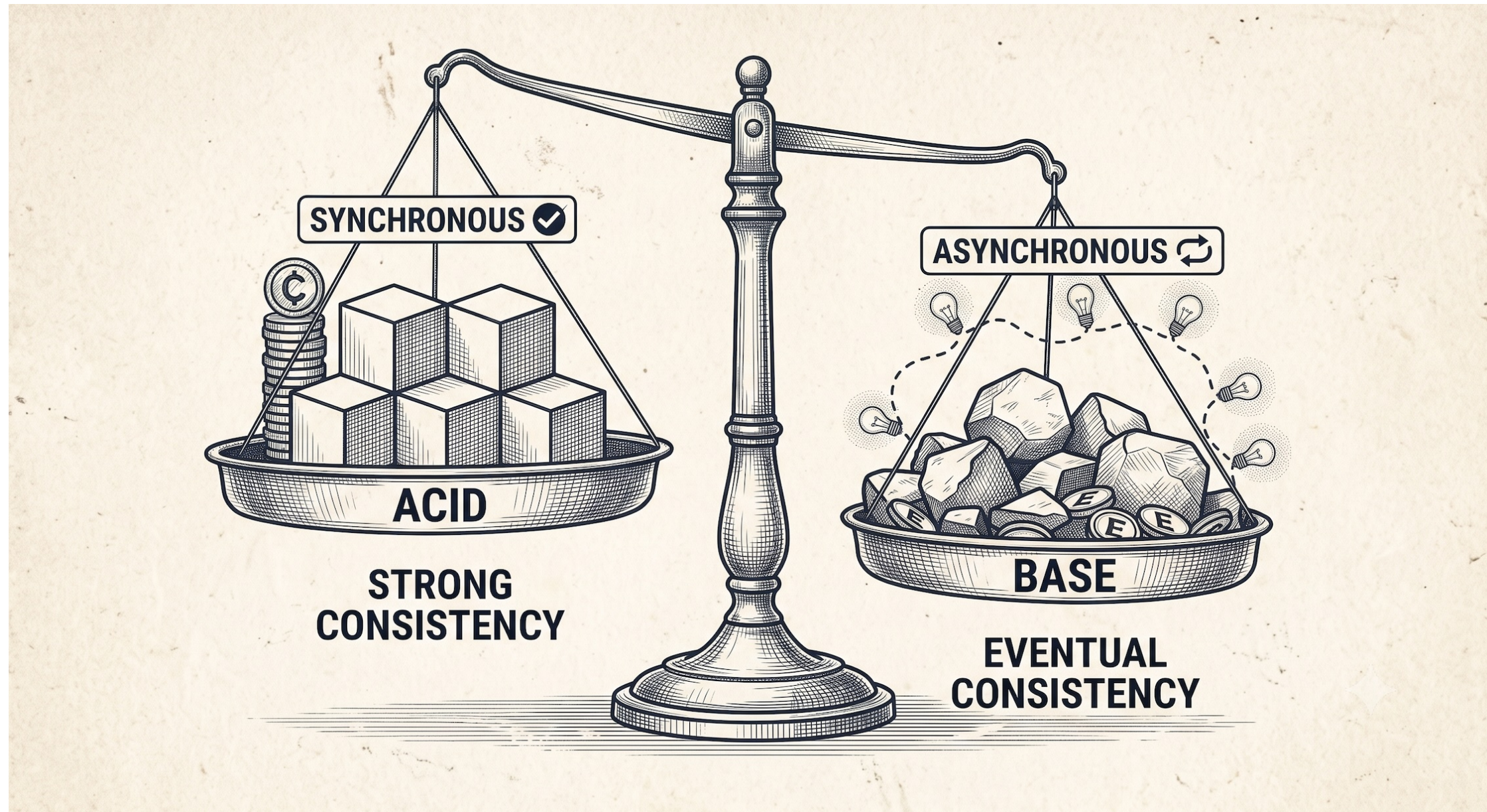
Letztendlich konsistenter Zustand

Anforderungen ans Datenmodell wachsen



Deckt unser Datenmodell jetzt **wirklich** alles ab?

Hör auf alles festzunageln

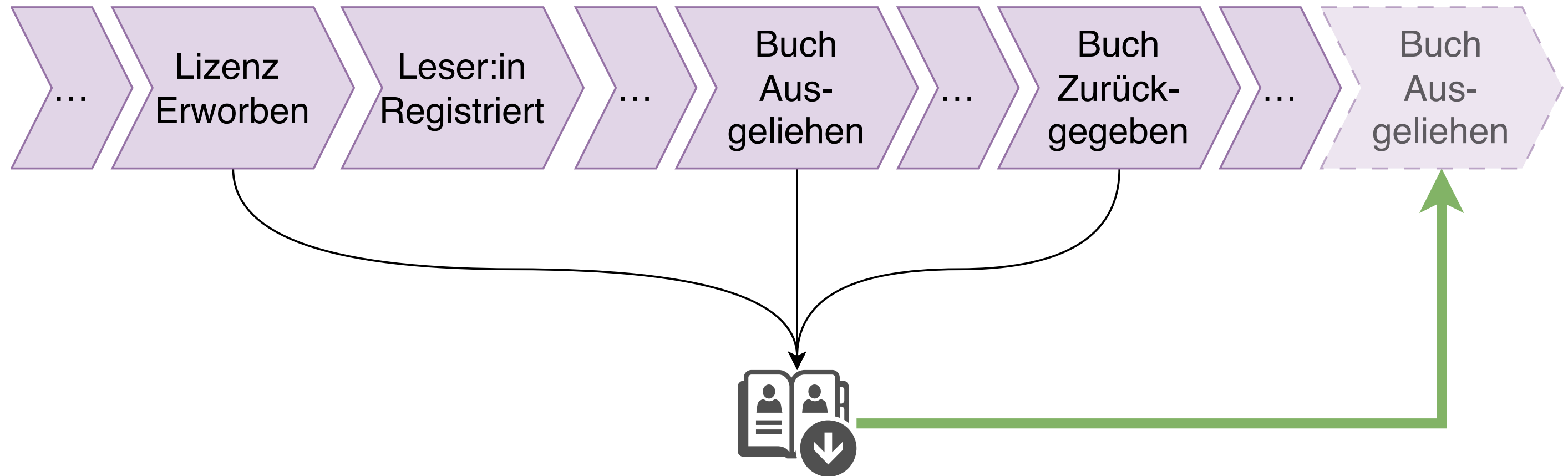


Was wäre, wenn wir statt Zuständen **Änderungen** speichern?

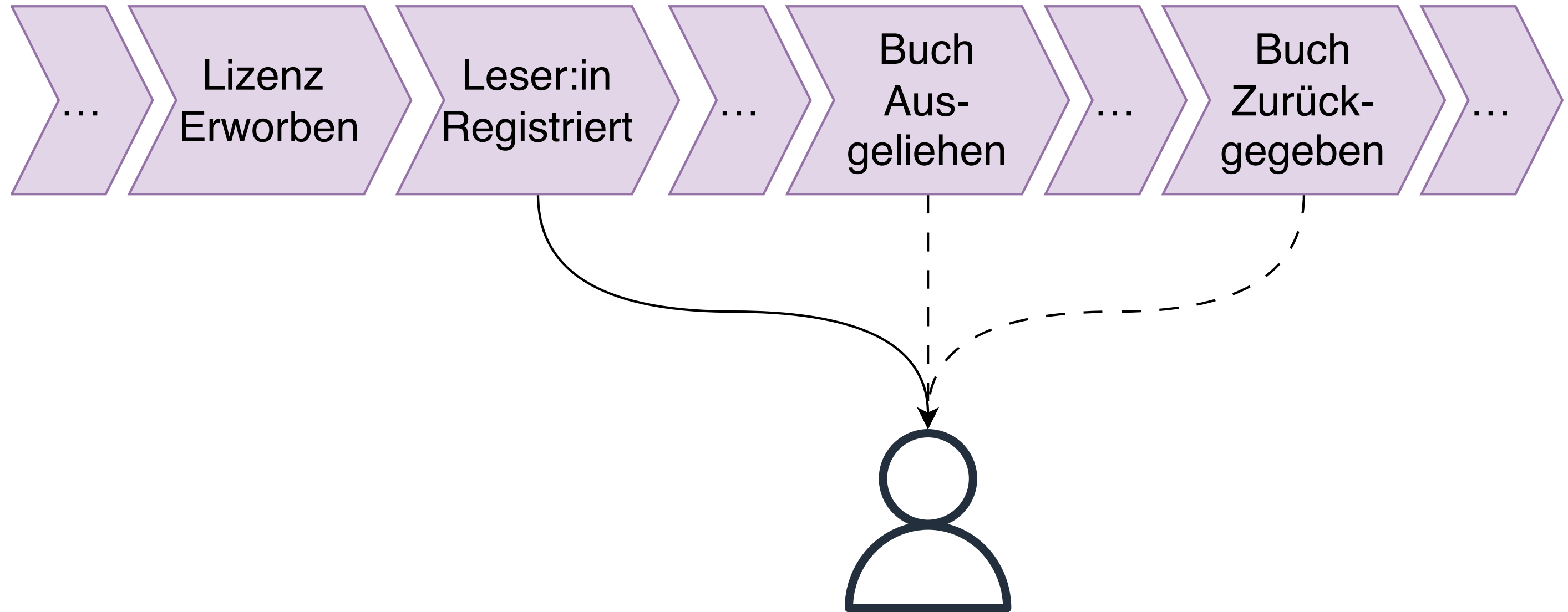
Das unveränderliche Log



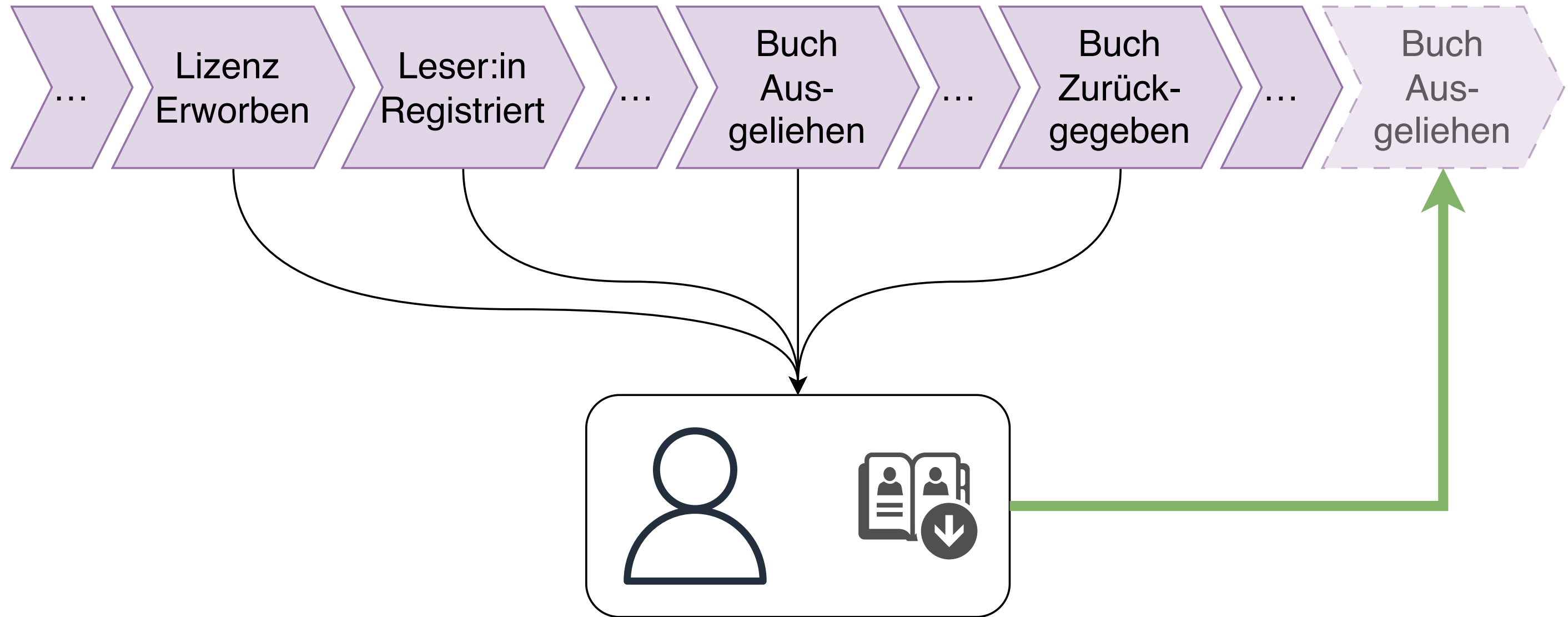
Buchausleihe als Events



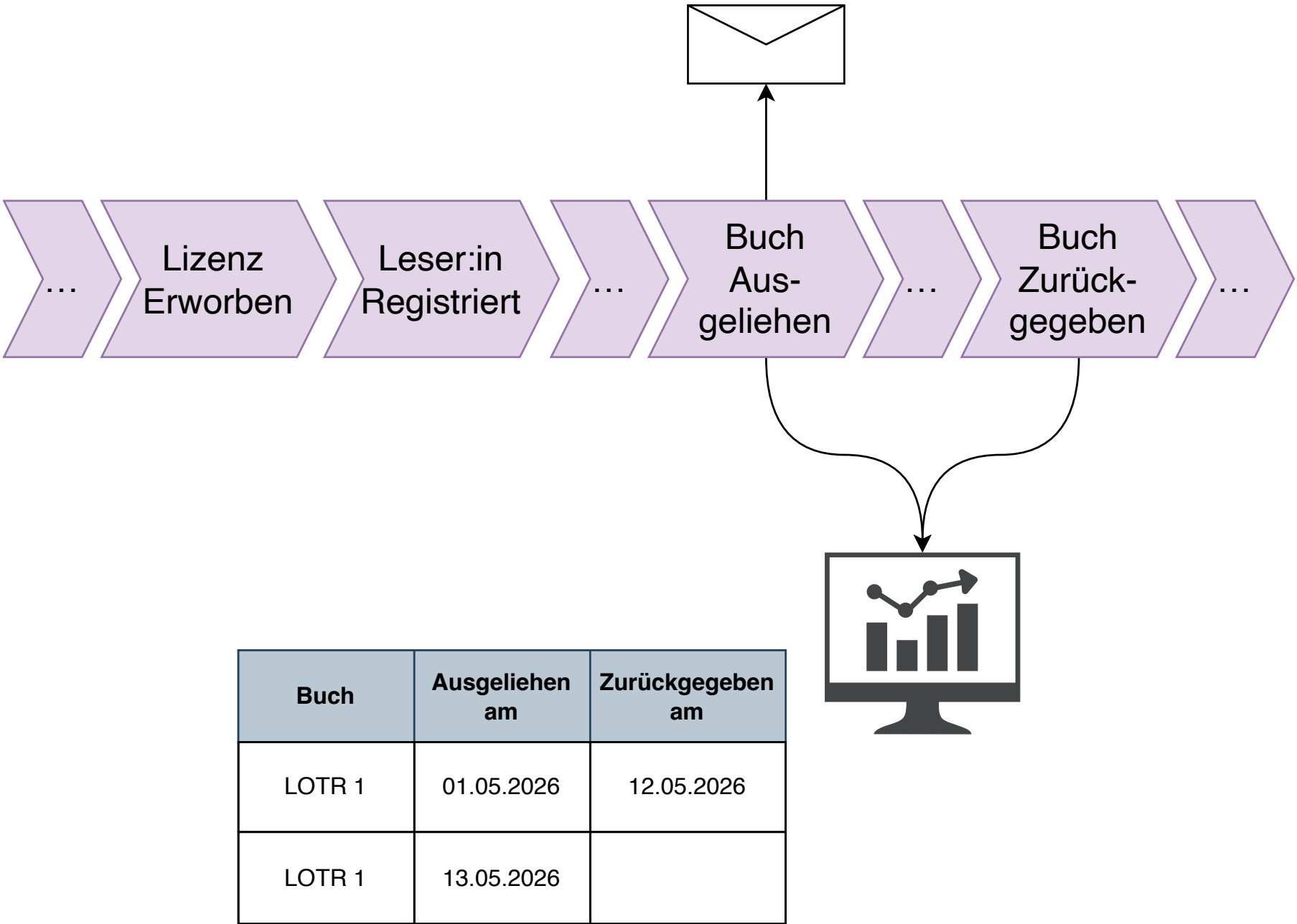
Leser:in als Events



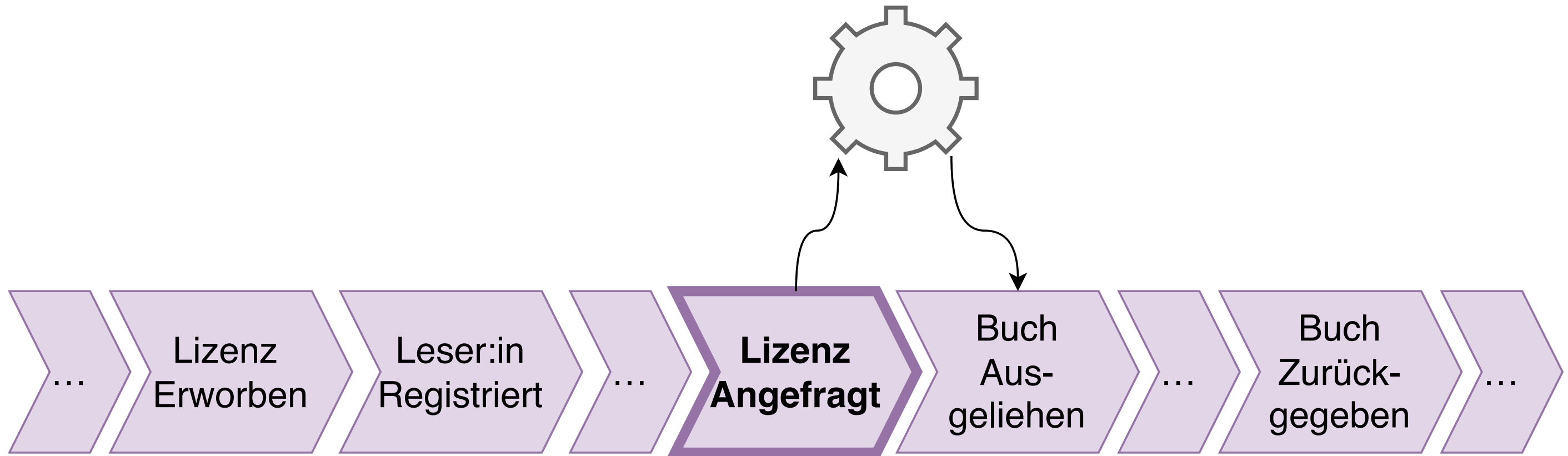
Dynamische Konsistenzgrenzen

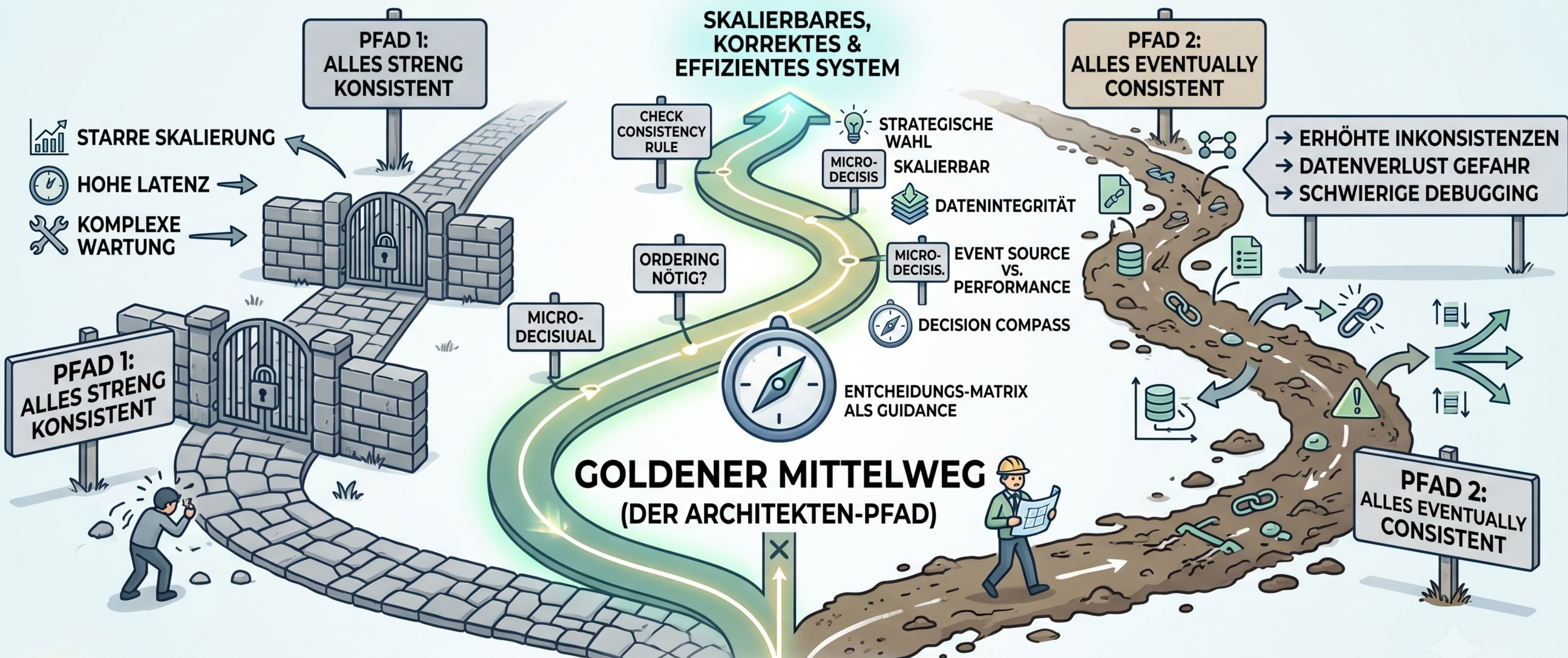


Transactional Outbox mit Events



Reservation Pattern mit Events

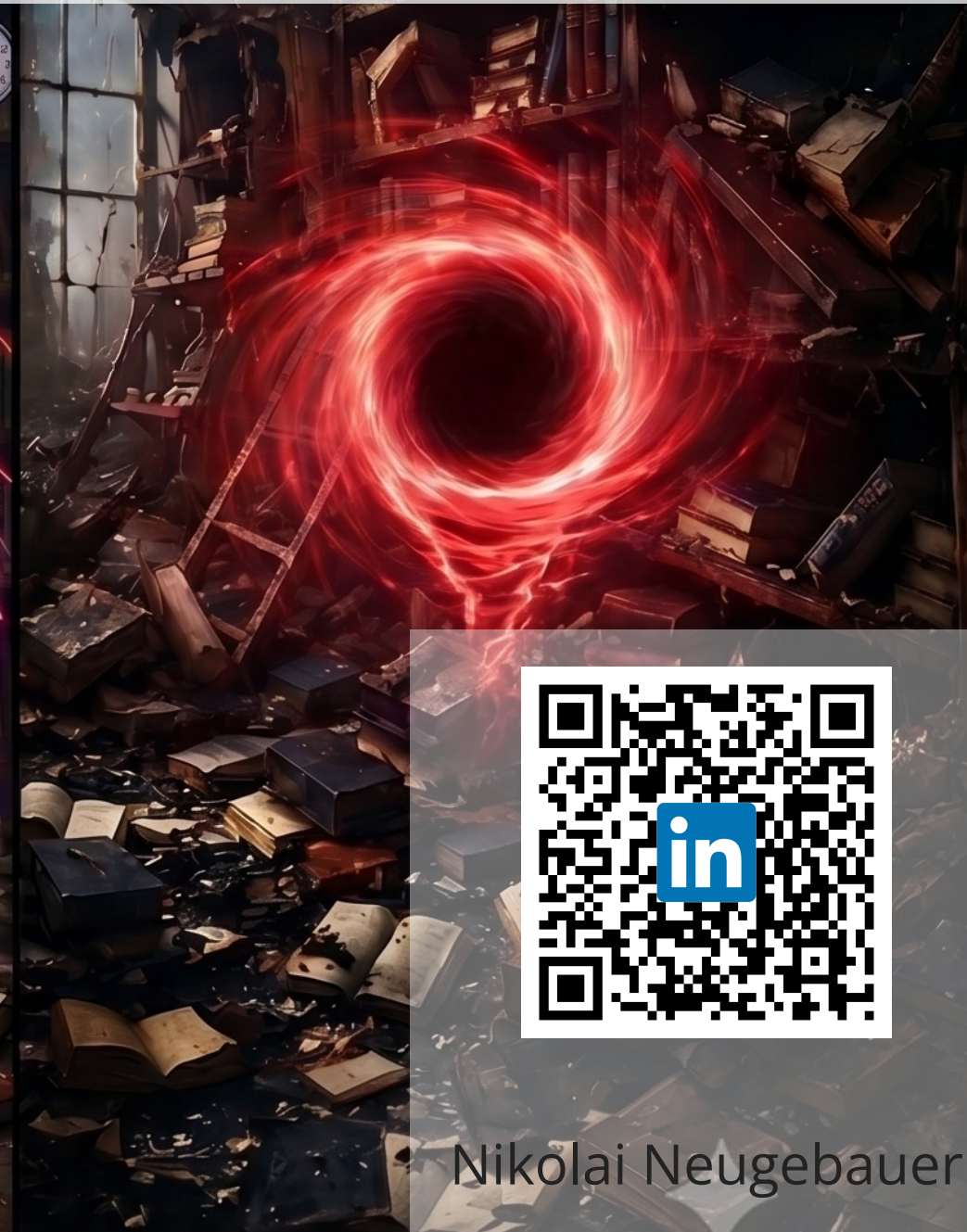




Wer definiert eure Konsistenzgrenzen? Die Technik oder das Business?



Frank Scheffler



Nikolai Neugebauer