



Bachelor's Thesis in Computer Science and Media

How does an Event Sourcing architecture compare to CRUD systems with an independent audit log, when it comes to scalability, performance and traceability?

Lukas Karsch

45259

Hochschule der Medien Stuttgart

Submitted on 2026/03/02

to obtain the degree of Bachelor of Science

Main Supervisor: Prof. Dr. Tobias Jordine

Secondary Supervisor: Felix Messner

Ehrenwörtliche Erklärung

Contents

1	Introduction	3
1.1	Motivation	4
1.2	Research question(s)	4
1.3	Goals and non goals	4
1.4	Structure of the paper	4
2	Basics	4
2.1	Layered Architecture Foundation (CRUD)	4
2.2	DDD Architectural Foundations	4
2.3	Traceability and auditing in IT systems	4
2.3.1	Why is traceability a business requirement	4
2.3.2	Audit Logs	4
2.3.3	Event Streams	4
2.3.4	Rebuilding state from an audit log and an event stream	4
2.4	Event Sourcing and event-driven architectures	4
2.5	(Eventual) Consistency	4
2.6	Scalability of systems	4
3	Related Work	4
4	Proposed Method	4
4.1	Project requirements	4
4.2	Performance	4
4.3	Scalability or flexibility (TODO)	4
4.4	Traceability	4
4.5	Tech Stack	4
5	Implementation	4
5.1	CRUD implementation	4
5.2	ES/CQRS implementation	4
5.3	Infrastructure	4
6	Results	4
7	Discussion	4
7.1	Analysis of results	4
7.2	Conclusion & Further work	4
8	Glossary	5

1 Introduction

I will build an API using two different architectural approaches.

- 1.1 Motivation
- 1.2 Research question(s)
- 1.3 Goals and non goals
- 1.4 Structure of the paper

2 Basics

- 2.1 Layered Architecture Foundation (CRUD)
- 2.2 DDD Architectural Foundations
- 2.3 Traceability and auditing in IT systems
 - 2.3.1 Why is traceability a business requirement
 - 2.3.2 Audit Logs
 - 2.3.3 Event Streams
 - 2.3.4 Rebuilding state from an audit log and an event stream
- 2.4 Event Sourcing and event-driven architectures
- 2.5 (Eventual) Consistency
- 2.6 Scalability of systems

3 Related Work

4 Proposed Method

- 4.1 Project requirements
- 4.2 Performance
- 4.3 Scalability or flexibility (TODO)
- 4.4 Traceability
- 4.5 Tech Stack

5 Implementation

- 5.1 CRUD implementation
- 5.2 ES/CQRS implementation
- 5.3 Infrastructure

6 Results

8 Glossary

Glossary

API API stands for *Application Programming Interface*. It describes the public interface of a module or service, often exposed over a network.
3