HOCHSCHULE
DER MEDIEN

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

here i use API

1.1 Motivation

1.2 Research question(s)

1.3 Goals and non goals

1.4 Structure of the paper

2 Basics

2.1 WWW, Web APIs, REST

The World Wide Web (WWW) is a connected information network used to exchange data. Resources are can be accessed via URIs which are transferred using formats like JSON or HTML via protocols like HTTP. HTTP is a stateless protocol based on a request-response structure. It supports standardized request types (e.g. GET and POST) which convey a semantic meaning (Jacobs and Walsh 2004).

Web APIs are interfaces which enable applications to communicate. They use HTTP as network-based API (Fielding 2000, p. 138). Modern APIs typically follow REST principles. REST stands for "Representational State Transfer" and describes an architectural style for distributed hypermedia systems (Fielding 2000, p. 76).

REST APIs follow principles derived from a set of constraints (e.g. imposed by the HTTP protocol). One of them is "stateless communication": Communication between clients and the server must be *stateless*, meaning the client has to attach all necessary information for the server to fully understand the request.

Further, in REST applications, every resource must be addressable via a unique ID, which can then be used to derive URIs to access the resource. Below are some examples for resources and URIs which could be derived from them:

- Book; ID=1; URI=`http://example.com/books/1`
- Book; ID=2; URI=`http://example.com/books/2`
- Author; ID=100; URI=`http://example.com/authors/100`

The "Hypermedia as the engine of application state (HATEOAS)" principle describes that resources should be linked to each other. Clients should be able to control the application by following a series of links provided by the server (Tilkov 2007).

Every resource must support the same interface, typically that of HTTP methods (GET, POST, PUT, etc.) where operations on the resource are mapped to one method of the interface. A POST operation on a customer might map to the `createCustomer()` operation on a service.

Resources are decoupled from their representations. Client can request different representations of a resource, depending on their needs (Tilkov 2007): a web browser might request HTML, while another server or application might request XML or JSON.

2.2 Layered Architecture Foundations (CRUD)

2.3 DDD Architectural Foundations

2.4 Traceability and auditing in IT systems

2.4.1 Why is traceability a business requirement

2.4.2 Audit Logs

2.4.3 Event Streams

2.4.4 Rebuilding state from an audit log and an event stream

2.5 Event Sourcing and event-driven architectures

2.6 (Eventual) Consistency

2.7 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

7 Discussion

7.1 Analysis of results

7.2 Conclusion & Further work

Finally, I'm done.

Glossary

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

HATEOAS Hypermedia as the engine of application state. 4

HTML Hypertext Markup Language. 4

HTTP HTTP stands for *Hypertext Transfer Protocol.* It is a protocol used in internet communication and was defined in RFC 2616 (*RFC 2616: HTTP/1.1* 2025). 3

JSON JavaScript Object Notation. 4

REST REST stands for *Representational State Transfer.* It is an architectural style for distributed hypermedia systems.. 3

WWW World Wide Web. 3

XML Extensible Markup Language. 4

References

Fielding, Roy Thomas (2000). “Architectural Styles and the Design of Network-based Software Architectures”. en. In.

Jacobs, Ian and Norman Walsh (Dec. 2004). *Architecture of the World Wide Web, Volume One.* URL: `https://www.w3.org/TR/webarch/` (visited on 12/27/2025).

RFC 2616: HTTP/1.1 (2025). URL: `https://www.w3.org/Protocols/HTTP/1.1/rfc2616.pdf` (visited on 12/27/2025).

Tilkov, Stefan (2007). “A Brief Introduction to REST”. en. In.