HOCHSCHULE
CAMPUS WIEN
UNIVERSITY OF
APPLIED SCIENCES

Bachelor's Thesis

Deletion and Anonymization of Personal Data in Apache Kafka Despite Log Immutability

Löschung und Anonymisierung personenbezogener Daten in Apache Kafka trotz Log-Immutabilität

Submitted by:	**Alexander Doubrava**
To partially obtain the academic degree of:	**Bachelor of Science in Engineering**

Vienna, 2025	
Study program:	Computer Science & Digital Communications
at:	Hochschule Campus Wien
supervised by:	FH-Prof. FH-Hon.Prof. Priv.-Doz. Mag. DI. DI. Dr. Karl Michael Göschka

Declaration of authorship:

I (Alexander Doubrava) declare that this Bachelor Thesis has been written by myself. I have not used any other than the listed sources, nor have I received any unauthorized help.

I hereby certify that I have not submitted this Bachelor Thesis in any form (to a reviewer for assessment) either in Austria or abroad.

Furthermore, I assure that the (printed and electronic) copies I have submitted are identical.

Date: December 25, 2025

Signature: Alexander Doubrava

Acknowledgements

I would like to express my sincere appreciation to my supervisor, Dr Göschka, for his valuable advice and feedback throughout the course of this thesis. Further thanks go to my colleague, Jasmin Bajrić, for reviewing the thesis and providing valuable feedback.

Abstract

Compact summary of the thesis in german (approx. 150–250 words).

Abstract

Compact summary of the thesis in English (approx. 150–250 words).

List of abbrevations

(optional)

Example::

AI	Artificial Intelligence
NLP	Natural Language Processing
LLM	Large Language Model

Contents

1. Introduction

Event streaming is becoming increasingly more relevant in modern microservice systems. Technologies such as Apache Kafka have become widely adopted due to enabling asynchronous communication, loose coupling, and scalable data distribution. Loose coupling applies to Kafka in two ways: Spatial decoupling and Temporal decoupling. Spatial decoupling is made possible by Kafka acting as a middleman between services. Temporal decoupling is enabled by Kafka holding a log of messages that can then be consumed by an interested service at a later date. Technologies like RabbitMQ offer this too, Kafka however adds an additional feature: Event Retention and Replayability. While RabbitMQ deletes a message from its queue once it has been consumed, Kafka however retains its messages in its topics even after they have been consumed. This allows consumers to repeatedly consume messages therefore enabling replayability.

The Apache Kafka ecosystem also offers the Kafka Streams framework. It offers an easy way to process, transform, join, group/aggregate and more directly on Kafka topics writing Java code. Shaving of the effort of having to deal with consumers, producers, state management, join logics and more all the while having to consider the partitioning of topics and ensuring scalability. Per default Kafka Streams applications materialize state locally in RocksDB state stores, caches, and changelog topics, potentially duplicating personal data across multiple components. To add to this, in complex environments Kafka Streams are often chained together causing many messages to be even further duplicated. Complicating any potential attempts at cleaning out personal data.

Overall, the Apache Kafka ecosystem is inherently stateful and strongly oriented around data retention.

At the same time, regulatory frameworks such as the General Data Protection Regulation (GDPR, DSGVO) impose strict requirements on the processing of personal data. In particular, the ”right to erasure” requires that personal data be deleted or irreversibly anonymized within a defined timeframe and that such deletion can be demonstrated.

This clashes with Kafka’s inherent architectural design considerations, as Apache Kafka’s log segments are not designed with fine grained message removal in mind. Instead focusing on time and size based retention, log compaction and tombstone records for the removal of data. The complex nature of Kafka Streams also makes it difficult to ascertain whether all information has been removed or not, as while data might have been removed from one topic, it might still be present in other topics or local state stores. As a result, ensuring timely, complete, and verifiable deletion of personal data in Kafka based systems is a non-trivial task.

Therefore, the research question this thesis intends to address is: **How can personal data be deleted or irreversibly anonymized in an Apache Kafka based event streaming system despite log immutability, while meeting GDPR requirements for timeliness, completeness, and demonstrable compliance?**

To answer this question, this thesis will first explore Apache Kafka’s existing deletion and anonymization mechanisms and investigate the capabilities of similar systems. These mechanisms are then analyzed with respect to their suitability for GDPR compliant deletion/anonymization, their limitations, and their impact on the system. Following this, the thesis will explore potential strategies for achieving data deletion and anonymization in Kafka, evaluating their effectiveness and suitability regarding the ”right to erasure” of the GDPR. Finally, coming together in one combined strategy for which a proof-of-concept implementation will be developed to explore the feasibility of the proposed approach.

The goal of this thesis is to provide practical insights for organizations using Apache Kafka who need to comply with GDPR data deletion requirements, while also contributing in general to the understanding of data and state/log management in event streaming systems.

2. Chapter heading

2.1. Subchapter heading

This is a sentence. [1]

2.1.1. Sub-subchapter heading

Like Muster (2020) shows, this is how you add a citation or reference.

2.2. Sub-subchapter heading

[1] and here is the explanation to the footnote

3. Chapter Heading

3.1. Subchapter heading

Abb. 1 Image of books

In the picture 1 you can see a stack of books.

3.1.1. Sub-subchapter heading

Tab. 1 Example Table

Name	Amount	Unit
Alpha	12.3	m/s
Beta	4.5	m/s
Gamma	7.8	m/s

See table 1

3.2. Subchapter heading

Bibliography

Muster, Max (2020). *Ein Beispielbuch*. Berlin: Beispielverlag.

Overview of AI technologies used

Describe the used AI-Tools, Models and Workflows. Use tables if necessary:

Technology	Use
OpenAI GPT	text generation, ideation, code examples
spaCy	NLP-Pipeline, Entity Recognition
scikit-learn	classification, model comparison

List of Figures

List of Tables

A. Appendix A

(optional) Only for additional documents such as circuit diagrams, program listings, interview transcripts, calculation examples, statistics, construction plans, sketches, etc.