



Hochschule Karlsruhe  
University of Applied Sciences

Fakultät für Informatik und Wirtschaftsinformatik  
Wirtschaftsinformatik

## BACHELORTHESIS

Performance-Optimierung einer asynchronen Marketing Event-API

von	Drilon Jashari
Matrikelnr.	74861
Arbeitsplatz	SAP SE, Walldorf
Erstbetreuer*in	Prof. Dr. rer. nat. Udo Müller
Zweitbetreuer*in	Prof. Dr. Ingo Stengel
Abgabetermin	30.06.2025

Karlsruhe, 30.06.2025

Vorsitzender des Prüfungsausschusses

# Inhaltsverzeichnis

<b>Inhaltsverzeichnis</b>	<b>2</b>
<b>1 Einleitung</b>	<b>4</b>
1.1 Problemstellung und Motivation . . . . .	4
1.2 Forschungsfrage und Hypothesen . . . . .	4
1.3 Zielsetzung und Abgrenzung . . . . .	4
1.4 Aufbau der Arbeit . . . . .	4
<b>2 Theoretische Grundlagen</b>	<b>5</b>
2.1 Marketing-Automatisierung und Event-APIs . . . . .	5
2.2 Java: Architektur, Speicherverwaltung und Threading-Modell . . . . .	5
2.3 Node.js: Event-Loop, asynchrone Programmierung . . . . .	5
2.4 Leistungsmessung von Webdiensten: Methoden und Metriken . . . . .	5
<b>3 Stand der Forschung</b>	<b>6</b>
3.1 Bestehende Vergleiche zwischen Java und Node.js . . . . .	6
3.2 Performance-Optimierung von Web-APIs . . . . .	6
3.3 Spezifische Herausforderungen bei Marketing-Event-APIs . . . . .	6
<b>4 Methodik</b>	<b>7</b>
4.1 Beschreibung der untersuchten Systeme . . . . .	7
4.2 Entwicklungs- und Testumgebung . . . . .	7
4.3 Erhebungsmethoden und Metriken . . . . .	7
4.4 Teststrategie und Lastprofile . . . . .	7
<b>5 Implementierung</b>	<b>8</b>
5.1 Java-Implementierung: Architektur und Besonderheiten . . . . .	8
5.2 Node.js-Migration: Anpassungen und Herausforderungen . . . . .	8
5.3 Optimierungsstrategien . . . . .	8
<b>6 Ergebnisse</b>	<b>9</b>
6.1 Darstellung der Messergebnisse . . . . .	9
6.2 Vergleich beider Implementierungen . . . . .	9

6.3	Analyse der Leistungsunterschiede . . . . .	9
<b>7</b>	<b>Diskussion</b>	<b>10</b>
7.1	Interpretation der Ergebnisse . . . . .	10
7.2	Beantwortung der Forschungsfrage . . . . .	10
7.3	Kritische Würdigung der Methodik . . . . .	10
7.4	Praktische Implikationen für SAP Emarsys . . . . .	10
<b>8</b>	<b>Fazit und Ausblick</b>	<b>11</b>
8.1	Zusammenfassung der Ergebnisse . . . . .	11
8.2	Praktische Empfehlungen . . . . .	11
8.3	Weiterführende Forschungsfragen . . . . .	11
<b>A</b>	<b>Anhang</b>	<b>12</b>
A.1	Quellcode-Auszüge . . . . .	12
A.2	Detaillierte Messergebnisse . . . . .	12
A.3	Sonstige relevante Materialien . . . . .	12

# Kapitel 1

## Einleitung

- 1.1 Problemstellung und Motivation
- 1.2 Forschungsfrage und Hypothesen
- 1.3 Zielsetzung und Abgrenzung
- 1.4 Aufbau der Arbeit

# Kapitel 2

## Theoretische Grundlagen

- 2.1 Marketing-Automatisierung und Event-APIs
- 2.2 Java: Architektur, Speicherverwaltung und Threading-Modell
- 2.3 Node.js: Event-Loop, asynchrone Programmierung
- 2.4 Leistungsmessung von Webdiensten: Methoden und Metriken

# Kapitel 3

## Stand der Forschung

- 3.1 Bestehende Vergleiche zwischen Java und Node.js
- 3.2 Performance-Optimierung von Web-APIs
- 3.3 Spezifische Herausforderungen bei Marketing-Event-APIs

# Kapitel 4

## Methodik

4.1 Beschreibung der untersuchten Systeme

4.2 Entwicklungs- und Testumgebung

4.3 Erhebungsmethoden und Metriken

4.4 Teststrategie und Lastprofile

# Kapitel 5

## Implementierung

- 5.1 Java-Implementierung: Architektur und Besonderheiten
- 5.2 Node.js-Migration: Anpassungen und Herausforderungen
- 5.3 Optimierungsstrategien



# Kapitel 6

## Ergebnisse

6.1 Darstellung der Messergebnisse

6.2 Vergleich beider Implementierungen

6.3 Analyse der Leistungsunterschiede

# Kapitel 7

## Diskussion

7.1 Interpretation der Ergebnisse

7.2 Beantwortung der Forschungsfrage

7.3 Kritische Würdigung der Methodik

7.4 Praktische Implikationen für SAP Emarsys

# Kapitel 8

## Fazit und Ausblick

8.1 Zusammenfassung der Ergebnisse

8.2 Praktische Empfehlungen

8.3 Weiterführende Forschungsfragen

# Anhang A

## Anhang

A.1 Quellcode-Auszüge

A.2 Detaillierte Messergebnisse

A.3 Sonstige relevante Materialien