



Carl von Ossietzky Universität Oldenburg

**Microservice-Architekturen im Kontext
von Data Science Workflows**

Konzeption, Experimente, unternehmensrelevante Implikationen

Von der
Fakultät für Informatik, Wirtschafts- und Rechtswissenschaften der
Carl von Ossietzky Universität Oldenburg zur Erlangung des
Grades und Titels eines

Doktors der Ingenieurwissenschaften (**Dr.-Ing.**)

angenommene Dissertation
von

Christoph Schröder

geboren am
12.05.1991 in Lingen

Gutachter: **Prof. Dr.-Ing. habil. Jorge Marx Gómez**
Weiterer Gutachter: **Prof. Dr.-Ing. Hergen Pargmann**

Tag der Disputation: 04. Juli 2023 in Oldenburg

Danksagung

Ich bedanke mich ganz herzlich bei meinem Doktorvater Prof. Dr.-Ing. Jorge Marx Gómez, Professor für Wirtschaftsinformatik und Leiter der Abteilung Very Large Business Applications an der Carl von Ossietzky Universität Oldenburg. Jorges Unterstützung von Anfang an sowie sein Feedback zur Dissertation und zu gemeinsamen wissenschaftlichen Publikationen haben wesentlich zum Erfolg des Promotionsverfahrens beigetragen.

Für die Übernahme der Zweitbegutachtung bedanke ich mich bei Prof. Dr.-Ing. Hergen Pargmann, der durch praxisnahe Fragen und Kommentare wichtige Impulse gesetzt hat. Außerdem möchte ich mich beim Vorsitzenden der Prüfungskommission, Prof. Dr. Philipp Staudt, sowie beim Prüfungskommissionsmitglied, Dr.-Ing. Sven Rosinger bedanken.

Die Dissertation ist größtenteils während meiner Tätigkeit in der Future Research Abteilung der Volkswagen AG entstanden. Ich bedanke mich daher bei Wolfgang Müller-Pietralla, Dr. Sven Wittfoth und Dr. Jonas Frischkorn für die optimalen Rahmenbedingungen, gemeinsamen wissenschaftlichen und praxisrelevanten Diskussionen sowie die gute Stimmung im Büro.

Finalisiert habe ich die Dissertation während des Aufbaus eines eigenen Unternehmens. Dafür bedanke ich mich ganz herzlich bei meinen beiden Mitgründern Felix und Raphael, die nicht nur über die vielen Jahre davor mich ausgezeichnet unterstützt haben, sondern Verständnis für die Arbeit an der Dissertation gezeigt haben. Auch bedanke ich mich bei den Kollegen und Kolleginnen der Abteilung Very Large Business Applications für die vielen Diskussionen und Feedbackrunden während der Promotion.

Last but not least, bedanke ich mich bei meinem privaten Umfeld und meinen Freunden. Ganz besonderer Dank gilt meiner Freundin Lara, meinem Vater Gerd und meinen Geschwistern Verena und Jochen, die mich in den letzten Jahren mehr als großartig unterstützt haben und mir die Zeit und den Raum gegeben haben, meinen wissenschaftlichen Zielen erfolgreich nachgehen zu können.

Zusammenfassung

Im Zuge der Digitalisierung von Unternehmen gewinnen Unternehmensdaten und deren Analyse zunehmend an Bedeutung. Dazu hat sich mit der Data Science eine eigenständige Forschungsdisziplin entwickelt. Data Science zielt darauf ab, Unternehmensdaten explorativ zu erkunden und für datengetriebene Entscheidungsprozesse zu analysieren (vgl. Long und Talbot, 2015). Die wachsende Bedeutung zeigt sich unter anderem auch in dem Marktvolumen, das für Data Science Plattformen beispielsweise bis 2026 um jährlich ca. 27 Prozent auf über 320 Mrd. USD anwachsen soll (vgl. Markets and Markets, 2021; Redgate, 2022).

Data Science etabliert sich in Unternehmen zunehmend. Für die Umsetzung und Implementierung von Data Science Vorhaben braucht es effiziente Softwarearchitekturen. Dabei ist nicht nur die reine technische Funktionalität erforderlich, um hohe und schnell aufkommende Datenvolumen verarbeiten zu können. Auch die schnelle und effiziente Entwicklung von Data Science Vorhaben inkl. der Bereitstellung in produktive Umgebungen ist wichtig.

In transaktionsorientierten Applikationen haben sich Microservice-Architekturen etabliert. Diese adressieren zusammen mit Continuous Integration/Continuous Delivery-Ansätzen und Cloud-nativer Entwicklung schnelle Entwicklungs- und Bereitstellungszyklen. Nichtsdestotrotz hat sich einschlägige Forschung noch nicht ausführlich damit beschäftigt, inwiefern Data Science Workflows auch mit Microservices entworfen und implementiert werden können.

Das Ziel der vorliegenden Arbeit ist es, ein Konzept und ein Prozessmodell zu entwerfen, mit denen Microservice-Architekturen für Data Science Workflows abgeleitet werden. Weiterhin wird eine prototypische Implementierung dieses Konzeptes vorgeschlagen, mit der ausgewählte Schritte automatisiert werden können, um so die Anwendung des Konzepts zu vereinfachen und zu beschleunigen. Mit diesem Konzept und der prototypischen Implementierung können Softwarearchitekten effizient beim Entwurf Microservice-orientierter Data Science Workflows unterstützt werden. Forschungsfokus liegt auch auf der adäquaten Identifizierung von Microservices und der Aufteilung der Workflow-Funktionen.

Im Ergebnis berücksichtigt das Konzept die Modellierung von Data Science Workflows und Microservices, ein Prozessmodell, Microservice-Entwurfsmuster sowie die Microservice-orientierte Datenintegration. Das Prozessmodell besteht aus sieben Schritten, bei denen Designphase und Laufzeitphase essentiell miteinander verknüpft werden. In Bezug auf die Designphase wurden mehrere Designmetriken und Verfahren exploriert, die für Data Science Workflows angewendet werden und mit denen zur Designphase Microservice-Architektur-Kandidaten erstellt werden können. In Bezug auf die Laufzeitphase wird eine Experimentierarchitektur vorgeschlagen, mit der mehrere Architekturen parallel getestet werden können. Insbesondere werden dazu auch Data Science spezifische Workloads definiert. Das Konzept wurde über Experimente entwickelt sowie mit ausgewählten Experten über Interviews validiert.

Die Expertenmeinungen lassen darauf schließen, dass das entwickelte Prozessmodell eine externalisierbare Vorgehensweise erlaubt, mit der Entscheidungen nachvollzogen werden und ein Bewusstsein für Architekturentscheidungen geschaffen wird. Auch Themen wie die Reduzierung des manuellen Aufwands und somit Entgegnung des Fachkräftemangels konnten in den Interviews adressiert werden. Allgemein zeigt sich, dass die Relevanz für Microservices und ein Prozessmodell für die Data Science zukünftig noch ansteigen wird.

Abstract

Due to the growth of companies' use of digitalization strategies, corporate data and its analysis are becoming increasingly important. Therefore, data science has become an entire research discipline. The field of data science aims to explore and analyze corporate data for data-driven decision-making processes (cf. Long und Talbot, 2015). To this end, both data volume and market volume are growing continuously. The market volume for data science platforms, for example, is expected to grow by around 27 percent annually to over USD 320 billion by 2026 (cf. Markets and Markets, 2021; Redgate, 2022).

These changes show that data science is becoming increasingly important for companies. Efficient software architectures are needed for the implementation of data science projects. These projects require technical functionality for processing huge and fast emerging data volumes and the fast and efficient development of such projects, including deployment in production environments.

Microservice architectures have become established in transaction-oriented applications. Based on continuous integration/continuous delivery approaches and cloud-native development, microservices address fast development and deployment cycles. Nevertheless, relevant research regarding which data science workflows can also be designed and implemented with microservices is lacking.

This dissertation project aims to address the aforementioned gap in research. The goal is to design a concept and process model that can be used to derive microservice architectures for data science workflows. Furthermore, this dissertation proposes a prototypical implementation of this concept for automating and reducing manual efforts. With this concept and the prototypical implementation, software architects can be efficiently supported in the design of microservice-oriented data science workflow. This dissertation's research also focuses on adequately identifying microservices and partitioning workflow functions.

As a result, this dissertation's concept considers the modelling of data science workflows and microservices, a process model, microservice design patterns, and microservice-oriented data integration. The process model consists of seven steps in which the design and runtime phase are essentially linked. Concerning the design phase, several design metrics and procedures that can be applied to data science workflows and used to create microservice-architecture candidates for the design phase were explored. Concerning the runtime phase, an experimental architecture that can be used to test multiple architectures in parallel is proposed. In particular, data science-specific workloads are also defined. The concept was developed via experiments and validated by selected experts via interviews.

The experts stated that the developed process model allows for an externalizable approach to track decisions and create awareness for architecture decisions. Topics such as reducing manual effort and, thus, countering the shortage of skilled workers were also addressed in the interviews. In general, the relevance of microservices and a process model that concerns various perspectives should continue to increase in future.

Inhaltsverzeichnis

Abkürzungen	XI
Abbildungen	XV
Tabellen	XVII
Listings	XIX
1 Einleitung und Motivation	1
1.1 Problemstellung des Forschungsvorhabens	4
1.2 Ziele, Forschungsfragen und Abgrenzung der Arbeit	6
1.3 Forschungsdesign der Dissertation	8
1.4 Struktur der Arbeit	9
2 Grundlagen zu Data Science und Microservices	13
2.1 Data Science als Forschungsdisziplin	13
2.2 Data Science Architekturen	13
2.2.1 Organisatorische Data Science Vorgehensmodelle	14
2.2.2 Data Science Workflows	15
2.2.3 Workflow-Management-Systeme	17
2.2.4 Entwicklung technischer Architekturen für Data Science Workflows . .	17
2.3 Software- und Microservice-Architekturen	18
2.3.1 Softwarearchitekturen	18
2.3.2 Microservice-Architektur zur Gestaltung von IT-Systemen	19
2.3.3 Charakteristika von Microservices	21
3 Kontextbestimmung von Microservices	23
3.1 Literaturreviews über Kontextabgrenzungen von Microservices	24
3.2 Methodik systematischer Literaturreviews	25
3.2.1 Vergleich und Auswahl der Methode	25
3.2.2 Qualitative Inhaltsanalyse als Forschungsmethode	26
3.3 Artefakte zur Identifizierung von Microservices während der Anforderungsana- lyse	28
3.4 Identifizierung von Microservices während der Entwurfsphase	29
3.4.1 Atomare Einheit	29
3.4.2 Modellierung der Applikation	30
3.4.3 Identifikation von Microservice-Kandidaten	32
3.4.4 Einflussfaktoren auf die Identifikation von Microservices	32
3.5 Test und Evaluation von Identifizierungsansätzen	34
3.6 Interpretation der Ergebnisse	37

3.7	Zusammenfassung zur ersten Teilforschungsfrage	38
4	Allgemeines Prozessmodell zur Kontextbestimmung in Microservice-Architekturen	39
4.1	Existierende Prozessmodelle	39
4.2	Methode zur Entwicklung eines Prozessmodells	41
4.3	Schritte des allgemeinen Prozessmodells	42
4.3.1	Schritt 1: Wahl einer atomaren Einheit	43
4.3.2	Schritt 2: Modellierung der Beziehungen zwischen atomaren Einheiten	44
4.3.3	Schritt 3: Charakterisierung atomarer Einheiten und deren Beziehungen mit Qualitätsattributen	44
4.3.4	Schritt 4: Anwendung von Methoden und Algorithmen zur Identifizierung von Microservices	45
4.3.5	Schritt 5: Evaluation der Microservice-Kandidaten und -Architektur .	46
4.4	Evaluation des Prozessmodells	46
4.4.1	Evaluation anhand von Identifikationsansätzen aus dem Jahr 2020 . .	47
4.4.2	Fallstudie - Microservices für eine Kreditwürdigkeitsanwendung	47
4.4.3	Vergleich von Microservice-Architekturen und Diskussion des Prozessmodells	52
4.5	Zusammenfassung zur zweiten Teilforschungsfrage	53
5	Konzept zur Microservice-Kontextbestimmung für Data Science Workflows	55
5.1	Prozessmodell zur Ableitung einer Microservice-Architektur für Data Science Workflows	56
5.2	Modellierung von Data Science Workflows und Microservice-Architekturen . .	57
5.3	Einfluss von Microservice-Entwurfsmustern	59
5.3.1	Herausforderungen der Implementierung von Data Science Workflows	59
5.3.2	Auswahl von Microservice-Entwurfsmustern	61
5.4	Microservice-orientierte Datenintegration	63
5.4.1	Herausforderungen der Datenintegration	63
5.4.2	Einsatz von Microservices für die Datenintegration	65
6	Prototypische Umsetzung der Microservice-Kontextbestimmung für Data Science Workflows	69
6.1	Tools zur Identifikation von Microservices	69
6.2	Anforderungen an den Tool-Support	70
6.3	Designphase	72
6.4	Experimentierumgebung	73
6.4.1	Konzeption der Experimentierumgebung	74
6.4.2	Konzeption der automatisierten Bereitstellung einer Experimentierumgebung	75

6.5	Laufzeitmetriken	80
6.6	Abschließende Bewertung	82
6.7	Zusammenfassung der prototypischen Umsetzung	83
7	Experimente zur Kontextbestimmung von Microservices für Data Science Workflows	85
7.1	Methodik und Vorgehen	85
7.2	Multi-Metrik-Ansatz	86
7.2.1	Anwendungsfall und Data Science Workflow	87
7.2.2	Designphase des Prozessmodells	89
7.2.3	Laufzeitevaluation der Microservice-Architekturen	91
7.2.3.1	Laufzeitmetriken	91
7.2.3.2	Experimentierumgebung	94
7.2.3.3	Laufzeitergebnisse	96
7.2.4	Abschließende Bewertung der Microservice-Architekturen	98
7.3	Datenfluss-getriebener Ansatz	100
7.3.1	Anwendungsfall und Data Science Workflow	101
7.3.1.1	Data Science Workflow-System	101
7.3.1.2	Kreditwürdigkeitsprüfung als Data Science Workflow	102
7.3.2	Designphase des Prozessmodells	102
7.3.2.1	Datenfluss-getriebener Ansatz	103
7.3.2.2	Anwendung des Ansatzes auf ein Data Science Workflow-System	104
7.3.2.3	Microservice-Architektur für Anwendungsfall	108
7.3.3	Laufzeitphase der Microservice-Architekturen	109
7.3.3.1	Laufzeit- und Experimentierumgebung	109
7.3.3.2	Ergebnisse der Laufzeitevaluation	110
7.3.4	Abschließende Bewertung	113
7.4	Experimente zu Microservice-Entwurfsmustern	115
7.4.1	Anwendungsfall und Microservice-Architektur	115
7.4.2	Experimentierumgebung	117
7.4.3	Ergebnisse der Laufzeitevaluation	120
7.4.4	Diskussion und Auswahl passender Microservice-Entwurfsmuster	121
7.5	Zusammenfassung und Implikationen der Experimente	122
8	Evaluation	125
8.1	Grundlagen und Auswahl der Evaluationsmethode	125
8.2	Vorbereitung und Interviewleitfaden	127
8.3	Durchführung	128
8.4	Ergebnisse der Experteninterviews	129
8.4.1	Entwicklung von Data Science Workflows	129
8.4.2	Relevanz von Microservice-Architekturen	130

8.4.3	Beurteilung der Phasen des Prozessmodells	131
8.4.3.1	Schritte während der Designphase	131
8.4.3.2	Schritte der Laufzeitevaluation	131
8.4.3.3	Schritt der abschließenden Bewertung	133
8.4.4	Metriken und Einflussfaktoren auf die Architekturentwicklung	133
8.4.5	Reduzierung des manuellen Aufwands	134
8.4.6	Standardisierung der Architekturentwicklung	136
8.5	Implikationen der Experteninterviews	136
9	Zusammenfassung und kritische Betrachtung	139
9.1	Theoretische und praktische Implikationen	142
9.2	Limitationen	143
9.3	Weiterer Forschungsbedarf	143
	Anhang	144
	A Ausgewählte Publikationen für das Literaturreview	145
	B Mockups des Tool-Supports	147
	C Interviewleitfaden	149
	Literaturverzeichnis	150

Abkürzungen

ACID Atomicity, Consistency, Isolation, Durability

AI Artificial Intelligence

API Application Programming Interface

AWS Amazon Web Services

BPMN Business Process Model and Notation

BI Business Intelligence

CAP Consistency, Availability und Partition-tolerance

CD Continuous Delivery

CI Continuous Integration

CLI Command Line Interface

CPU Central Processing Unit

CQRS Command-Query-Responsibility-Segregation

CRISP-DM CRoss Industry Standard Process for Data Mining

DAG Directed acyclic graph

DBMS Datenbankmanagementsystem

DDD Domain Driven Design

EC2 Amazon Elastic Compute Cloud

EDA Event driven Architecture

ESB Enterprise Service Bus

RAMA CLI RESTful API Metric Analyzer CLI nach Bogner (2020)

gRPC gRPC Remote Procedure Calls

HTTP Hypertext Transfer Protocol

HTTPS Hypertext Transfer Protocol Secure

HV High Volume

IP Internet Protocol

IT Information Technology

JAR Java Archiv

JMX Jmeter XML

JSON JavaScript Object Notation

KI Künstliche Intelligenz

LV Low Volume

MAAT Microservice Architecture Analysis Tool nach Engel et al. (2018)

ML Machine Learning

MLOps Machine Learning Operations

MQTT Message Queuing Telemetry Transport

MSA Microservice-Architektur(en)

OASIS Organization for the Advancement of Structured Information Standards

PMML Predictive Model Markup Language

Pub Publishing Interface

PY Veröffentlichungsjahr (Publication Year)

RDS Amazon Relational Database Service

SaaS Software-as-a-Service

SOA Service-orientierte Architektur

SQS Amazon Simple Queue Service

Sub Subscribing Interface

SVM Support Vector Machine

TF Teilforschungsfrage

TLS Transport Layer Security

TOSCA Topology and Orchestration Specification for Cloud Applications

VM Virtuelle Maschinen

WDL Workflow Description Language

WF Workflow

XML Extensible Markup Language

Abbildungen

1.1	Mehrere Alternativen von Microservice-Architekturen.	3
1.2	Data Science Workflow (vgl. Blazquez und Domenech, 2018) und mögliche Abgrenzungen.	5
1.3	Darstellung der Ziele und mögliche Ergebnisse der Arbeit.	7
2.1	Data Analysis Layer nach Blazquez und Domenech (2018).	15
2.2	Microservices, deren Beziehungen und mögliche Kommunikationsmuster. Notation in Anlehnung an Richardson (2019).	21
3.1	Einfluss der Anzahl an Microservices auf Kostenkriterien (nach Gouigoux und Tamzalit (2017)).	23
3.2	Suchergebnisse.	27
3.3	Anzahl betrachteter Publikationen nach Veröffentlichungsjahr.	28
4.1	Prozessmodell für services-orientiertes Design und Entwicklung nach Papazoglou und van den Heuvel (2006).	40
4.2	Bei der Entwicklung des abstrakten Prozessmodells berücksichtigte Prozessmodelle und schritte.	42
4.3	Entwickeltes Prozessmodell für das Design und Kontextbestimmung von Microservices.	43
4.4	Atomare Einheiten und ihre Beziehungen, angereichert um Qualitätsattribute.	45
4.5	Funktionaler Workflow zur Berechnung der Kreditwürdigkeit basierend auf Berthold et al. (2008).	48
4.6	Datenflussdiagramm nach Durchlaufen des Schritts 2 des Prozessmodells (erste Iteration). Notation basiert auf Li et al. (2019).	50
4.7	Funktions-orientiertes Diagramm nach Schritt 2 des Prozessmodells (zweite Iteration). Notation nach Tserpes (2019).	51
4.8	Entity-Relationship Modell nach Schritt 2 des Prozessmodells (dritte Iteration).	51
5.1	Konzept zur Microservice-Kontextbestimmung für Data Science Workflows.	55
5.2	Ausschnitt aus dem Konzept zur Formalisierung des Zuordnungsproblems von Data Science Workflows zu Microservices.	58
5.3	Ausschnitt aus dem Konzept zur Darstellung des Einflusses von Microservice-Entwurfsmustern auf die Implementierung.	59
5.4	Ausschnitt aus dem Konzept zur Darstellung der Berücksichtigung der Microservice-orientierten Datenintegration.	63
5.5	Dezentrale Datenbereitstellung auf Basis von Microservices für zentrale Datenteams zur Bearbeitung analytischer Fragestellungen.	65
6.1	Komponentendiagramm des Tool-Supports.	72

6.2	Allgemeine Architektur des Prototyps zur Ableitung von Microservice-Architektur-Kandidaten während der Designphase auf Basis unterschiedlicher Workflow-Formate.	73
6.3	Prototypische Umsetzung zur Ableitung von Microservice-Architektur-Kandidaten während der Designphase.	73
6.4	Architektur für eine Experimentierumgebung (Symbole nach AWS (2022)) . .	74
6.5	Allgemeine Architektur des Prototyps zum automatisierten Deployment mehrerer Microservice-Architektur-Alternativen und zur Ausführung der Laufzeitevaluation.	76
6.6	Prototypische Umsetzung des Deployments und der Ausführung der Laufzeitevaluation.	76
6.7	Allgemeine Architektur des Prototyps zur Log-Zusammenführung und Berechnung von Laufzeitmetriken.	81
6.8	Prototypische Umsetzung der Log-Aggregation und Berechnung relevanter Laufzeitmetriken.	81
6.9	Allgemeine Architektur des Prototyps zur Ableitung einer Microservice-Architektur auf Basis von Design- und Laufzeitmetriken.	83
6.10	Prototypische Umsetzung der abschließenden Bewertung und Erstellung eines Reports.	83
6.11	Gesamtübersicht über die prototypische Umsetzung der automatisierten Ableitung und Evaluation Microservice-orientierter Data Science Workflows. . .	84
7.1	Einordnung der Experiments für Microservice-orientierte Data Science Workflows in das Gesamtkonzept.	87
7.2	Kohäsionsmetriken der Microservice-Architektur-Alternativen des Anwendungsfalls in Abhängigkeit von der unterschiedlichen Clusteranzahl.	91
7.3	Bereitstellung der Microservice-Architektur-Alternativen und Experimentierumgebung in der Cloud (AWS-Symbole nach AWS (2022)).	95
7.4	Average Change of Workflow Execution Time verglichen mit dem Basisworkload λ_0	96
7.5	Changes of Total Service Activation Time verglichen mit der Baseline-Architektur MSA_1	97
7.6	Ergebnisse der Domain Metric abhängig von verschiedenen Workloads.	97
7.7	Ausfälle nach Microservices.	98
7.8	Ausfälle nach Funktionen.	98
7.9	Kombinierte Metriken in Abhängigkeit verschiedener Metrikgewichte und Workload-Häufigkeiten.	100
7.10	Einordnung des Experiments für Microservice-orientierte Data Science Workflows in das Gesamtkonzept.	101
7.11	Fachlicher Datenfluss innerhalb eines Data Science Workflow-Systems.	105
7.12	Teilbares Datenflussdiagramm auf Basis des Level-1-Datenflussdiagramms für ein Data Science Workflow-System.	106

7.13	Datenflussdiagramm für ein Data Science Workflow-Systems mit abgeleiteten Microservice-Architektur-Alternativen.	108
7.14	Für den konkreten Anwendungsfall abgeleitete Microservice-Architektur.	109
7.15	Implementierung des Data Science Workflows sowohl als Microservice-Architektur als auch mit KNIME.	110
7.16	Ausführungszeit der Microservice-Architekturen in Abhängigkeit der Anzahl von Datensätzen.	111
7.17	Ausführungszeit MSA bei 100 000 Datensätzen.	112
7.18	Ausführungszeit MSA bei 250 000 Datensätzen.	112
7.19	Ausführungszeiten MSA und KNIME bei 1 000 Datensätzen.	112
7.20	Ausführungszeiten MSA und KNIME bei 25 000 Datensätzen.	112
7.21	Microservice-Architektur-Alternativen desselben Data Science Workflows.	115
7.22	Anwendung des Orchestrator Entwurfsmusters zur Umsetzung des Workflows.	117
7.23	Anwendung des Choreografie Entwurfsmusters zur Umsetzung des Workflows.	117
7.24	AWS Architektur mit Microservice-Architekturen und Infrastrukturkompo- nenten (AWS-Symbole nach AWS (2022)).	119
7.25	Mittlerer End-to-End-Delay $\overline{WED}(\lambda, \alpha)$ der Workflow-Ausführungen je Workload und Microservice-Architektur.	120
7.26	Mittlere Veränderung der End-to-End-Delays $\overline{\Delta WED}(\lambda, \alpha)$ von Workflow- Ausführungen verglichen mit den Baseline Workloads LV-1 und HV-1 je Ar- chitektur.	121
8.1	Evaluation über Experteninterviews und resultierte Ergebniskategorien.	127
B.1	Grafische Darstellung von Data Science Workflows im Confluence-Report.	147
B.2	Grafische Darstellung von Designmetriken und Microservice-Architektur-Al- ternativen im Confluence-Report.	147
B.3	Grafische Darstellung der bereitgestellten Microservice-Architekturen in der Experimentierumgebung im Confluence-Report.	147
B.4	Grafische Darstellung berechneter Laufzeitmetriken im Confluence-Report.	148
B.5	Grafische Darstellung der abschließenden Bewertung der Microservice-Architektur- Alternativen im Confluence-Report.	148

Tabellen

3.1	Artefakte für die Identifizierung von Microservices.	28
3.2	Art der Applikationen.	29
3.3	Atomare Einheiten.	30
3.4	Ansätze zur Modellierung der Applikation als Basis für die Identifizierung von Microservices im Rahmen der Entwurfsphase.	31
3.5	Algorithmen und Methoden zur Identifizierung von Microservice-Kandidaten.	32
3.6	Kohäsion und Kopplung.	33
3.7	Weitere Qualitätskriterien als Einflussfaktoren.	34
3.8	Evaluationsansätze.	35
3.9	Verwendete Applikationen für die Evaluation.	36
3.10	Vergleich der resultierenden Microservice-Architektur.	36
4.1	Anwendung des Prozessmodells zur Einordnung der Studie „A Complexity Metric for Microservices Architecture Migration“ (vgl. Santos und Rito Silva, 2020).	47
4.2	Fallstudie mit angewandtem Prozessmodell und drei Identifikationsansätzen.	49
5.1	Zuordnung von Workflow-Herausforderungen zu Microservice-Entwurfsmustern.	61
5.2	Charakteristika der Schnittstellenmethoden.	66
6.1	Auswahl an Attributen inkl. Beispielwerte, die über die Logs gesammelt werden.	82
7.1	Übersicht über die zur Konkretisierung des Konzeptes zur Microservice-Kontextbestimmung für Data Science Workflows durchgeführten Experimente.	85
7.2	Auswahl von Funktionen des Data Science Workflows.	88
7.3	Verwendete mathematische Symbole.	90
7.4	Übersicht über Metriken und Architektur-Alternativen bei gleichwahrscheinlichen Workloads.	99
7.5	Prozesse, Datenspeicher und Sätze auf Basis der Datenflussdiagramme in Abbildung 7.11.	106
7.6	Zuordnung der Sätze aus Tabelle 7.5 zu Microservices.	107
7.7	Zuordnung der Sätze aus Tabelle 7.5 zu Microservices auf Basis des Level-1-Datenflussdiagramms aus Abbildung 7.12.	107

Listings

- 6.1 Auszug aus dem erzeugten Cloud Formation Template. Auszutauschen ist lediglich das Docker Image und ggf. der Port des Listeners. 77
- 6.2 Auszug aus dem Script zum Starten eines Laufzeitexperiments. 79
- 6.3 Source Code zur Integration der Messung von Laufzeitdaten. 80

