Design and Implementation of a Tool to Collect Execution- and Service-Data of Big Data Analytics Applications

Bachelor's Thesis

for obtaining the academic degree
Bachlor of Science (B.Sc.)

at

Beuth Hochschule für Technik Berlin
Department Informatics and Media VI
Degree Program Mediainformatics

1. Examiner and Supervisor:	Prof. Dr. Stefan Edlich
2. Examiner:	Prof. Dr. Elmar Böhler
Submitted by:	Markus Lamm
Matriculation number:	s786694
Date of submission:	06.09.2016

Ackknowledgements

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Abstract

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Contents

1 Introduction

1.1 Motivation

According to a survey in Germany, nine out of ten companies (89 percent) analyze large volumes of data from a variety of diffent sources for operational decision-making processes using modern Big Data Analytics Applications, where 48 percent of the respondents see the greatest potential of Big Data [Jör14]. The analysis of continuous data streams is taking up a growing importance for companies and therefore constitutes an important factor for business success.

Collecting, storing and analyzing system and operational data of Big Data Applications is therefore an essential tool in order to ensure successful operation. Even though logfiles are usefull for tracing problems in software systems, problems can be tracked and potential sources of error can be identified much earlier by collecting and storing execution and service data at runtime to describe the state of the system at a given point in time.

Due to the distributed character of Big Data Applications, where a system is composed of several interacting components, the examination of log data is not an adequate choice to gain insight into an entire system [Les14].

1.2 Objective

The main goal of the thesis is the design and implementation of a software system to ingest and store system and operational data of Big Data Analytics Applications on the example of the streaming frameworks Apache Flink and Apache Kafka. It should be examined which data is available and can be collected at all, what data is relevant and how to collect from source systems. Furhermore, the collected data must be stored in a persistence system to become available for possible consumers like visualization applications, analytical processes or as a data source for applications from the context of Machine Learning for example.

1.3 Structure of thesis

After a short introduction to the topics and the main goals of the present thesis in this chapter, the Chapter 2 covers the theoretical foundations of Big Data Analytics Applications, discusses the concept of "stream-processing" and introduces Apache Flink and Apache Kafka as representatives of widely used stream-processing frameworks.

Chapter 3 investigates which sources for collecting data exist for Apache Flink and Kafka and which data should be collected and stored in a persistence system regarding to its relevance and data quality.

The requirements and the target definition of the software-system will be introduced in Chapter 4, Chapter 5 describes the software solution by giving a detailed conceptional overview of the software components and providing implementation details for selected items.

In chapter 6 we'll see how to setup the technical environment for the usage of the prototype to verify the correct functionality related to the requirements defined in Chapter 4.

The last Chapter 7 covers a conclusion and summary of the present work.

2 Theoretical Foundations

This chapter will discuss the main characteristics of Big Data Analytics Applications and introduces the concept of stream processing, which is one of the main characteristics of the popular streaming frameworks Apache Flink and Apache Kafka. The underlying concepts both of these systems and how they're used in context of Big Data Analytics will be explained at the end of this chapter.

2.1 Big Data Analytics Applications

Big Data Analytics describes the process of collecting, organizing and analyzing large volumes of data with the aim to discover patterns and other useful information extracted from a incoming data streams [Nat15]. The process of analytics is typically performed using specialized software tools and applications for predictive analytics, data mining, text mining, forecasting and data optimization.

The areas of applications may be extremely diverse and ranges from analysis of financial flows or traffic data, processing sensor data or environmental monitoring.

Characteristics:

Robustness and fault tolerance TODO

Low latency reads and updates TODO

Generalization TODO

Ad hoc queries TODO

2.2 Stream-Processing

According to [Kle16], stream processing is the real-time processing of data continuously, concurrently, and in a record-by-record fashion in which data is treated not as static tables or files, but as a continuous infinite stream of data integrated from both live and historical sources.

Benefits:

- Accessibility: live data can be used while still in motion, before being stored.
- Completeness: historical data can be streamed and integrated with live data for more context.
- High throughput: high-velocity, high-volume data can be processed with minimal latency.

2.2.1 Apache Flink

2.2.2 Apache Kafka

3 Data Analysis

3.1 System data

3.2 Java Management Extensions (JMX)

3.3 Representational State Transfer (REST)

3.4 Data Quality

4 Requirements

4.1 Collection

4.2 Transport

4.3 Persistence

5 Architecture and Implementation

5.1 Collected data as time-series based stream

5.2 Microservices and Service-Discovery

5.3 System components

5.3.1 CollectorClient

The CollectorClient tier is our entry point for bringing data into the system...

5.3.2 Service-Discovery

Registraction for CollectorClients

5.3.3 CollectorManager

Gives overview, uses Consul as service-discovery

5.3.4 Message-Broker

Transport, "Event-Log"

5.3.5 Indexer

Receive messages from Kafka, roote data, create ES index

5.3.6 Persistence

ES as search index for time-series based data, easy vizualization with Kibana

6 Evaluation

7 Conclusion

List of Figures

List of Tables

List of Source Codes

Bibliography

[Kle16] Martin Kleppmann. *Making Sense of Stream Processing.* First edition. Sebastopol, CA 95472: O'Reilly Media, Inc., 2016. ISBN: 978-1-491-94010-5.

[Nat15] James Warren Nathan Marz. *Big Data - Principles and best practices of scalable real-time data systems.* Shelter Island, NY 11964: Manning Publications Co., 2015. ISBN: 978-1-617-29034-3.

Articles

[Les14] Tammo van Lessen. “Wissen, was läuft - Mit Laufzeitmetriken den Überblick behalten”. In: *Javamagazin* 10000.11 (2014), pp. 48–52.

Online resources

[Jör14] u.a Jörg Bartel Axel Mester. *Big Data Technologien – Wissen für Entscheider*. 2014. URL: https://www.bitkom.org/Publikationen/2014/Leitfaden/Big-Data-Technologien-Wissen-fuer-Entscheider/140228-Big-Data-Technologien-Wissen-fuer-Entscheider.pdf (visited on 08/06/2016).

A

A.1 Diagrams

A.1.1 Use Case diagram

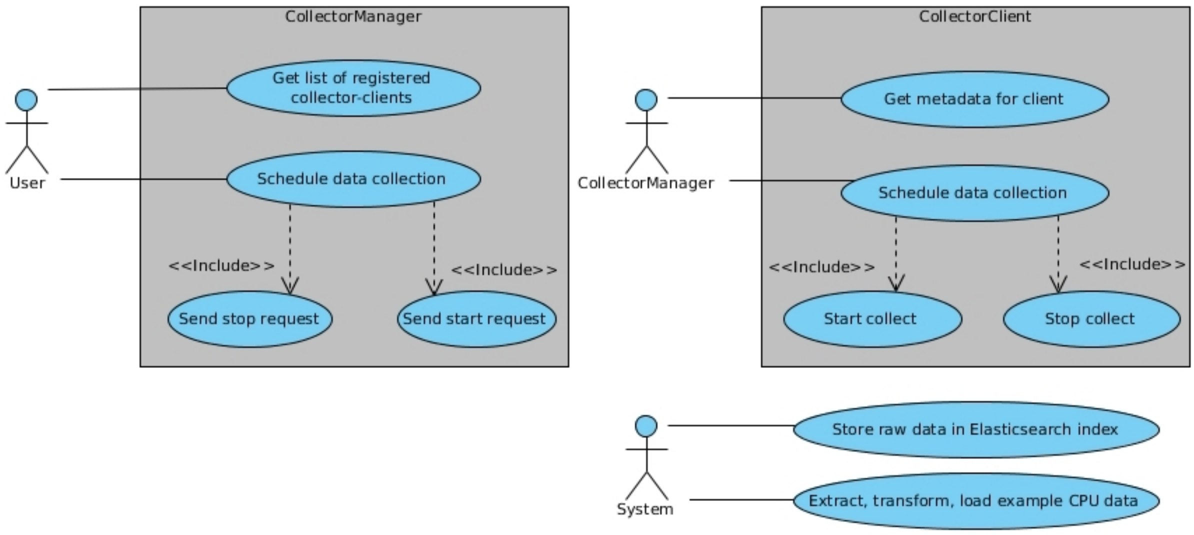

Figure A.1: Use Case Diagramm

A.1.2 Class diagrams

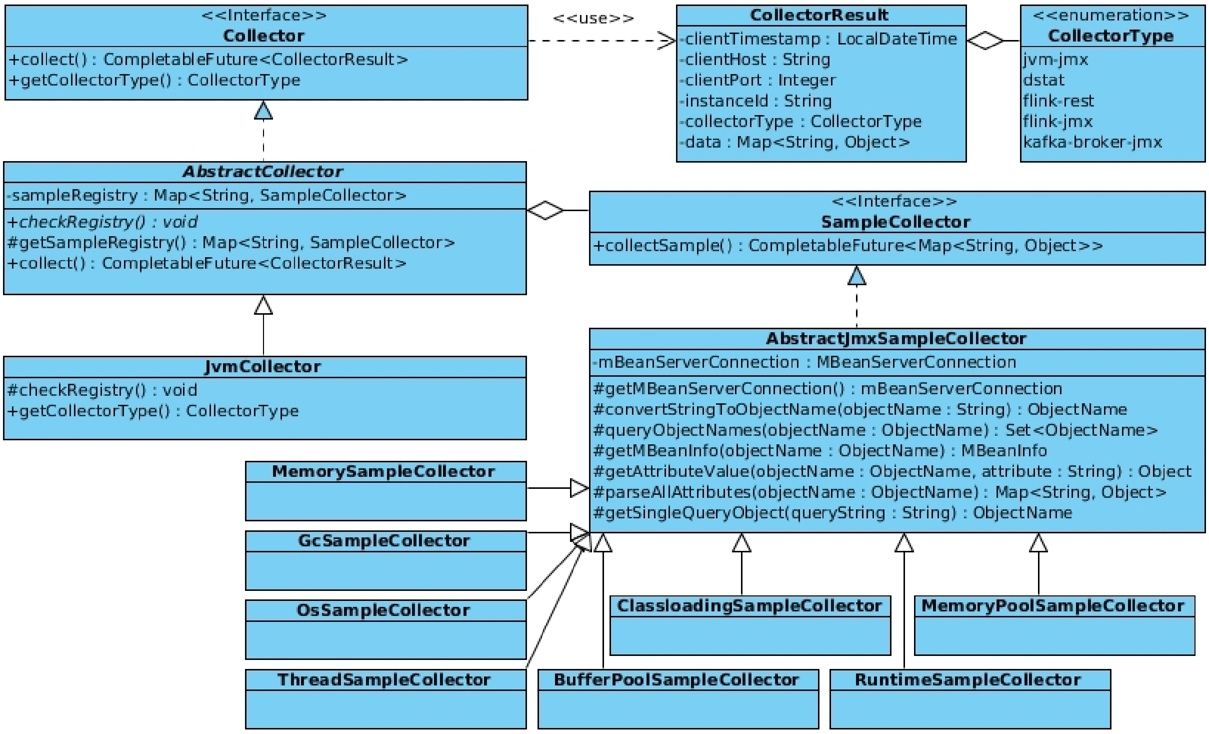

Figure A.2: Class diagram 'JvmCollector'

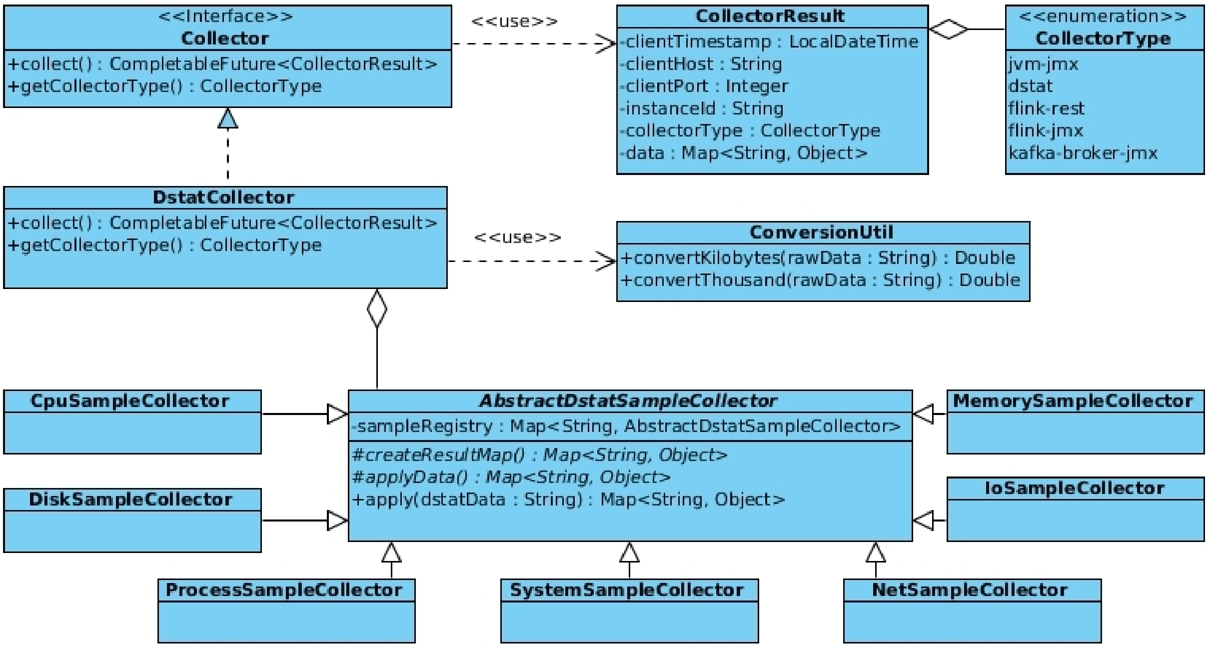

Figure A.3: Class diagram 'DStatCollector'

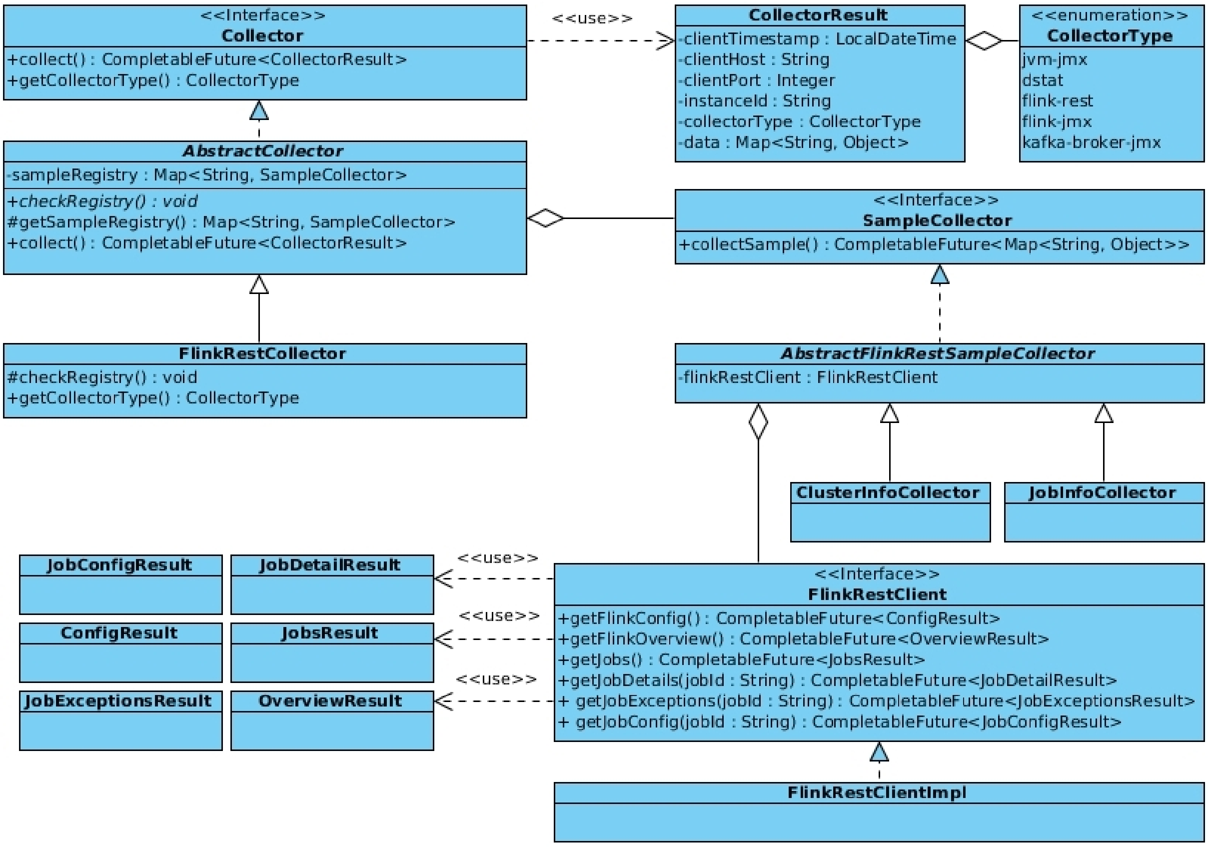

Figure A.4: Class diagram 'FlinkRestCollector'

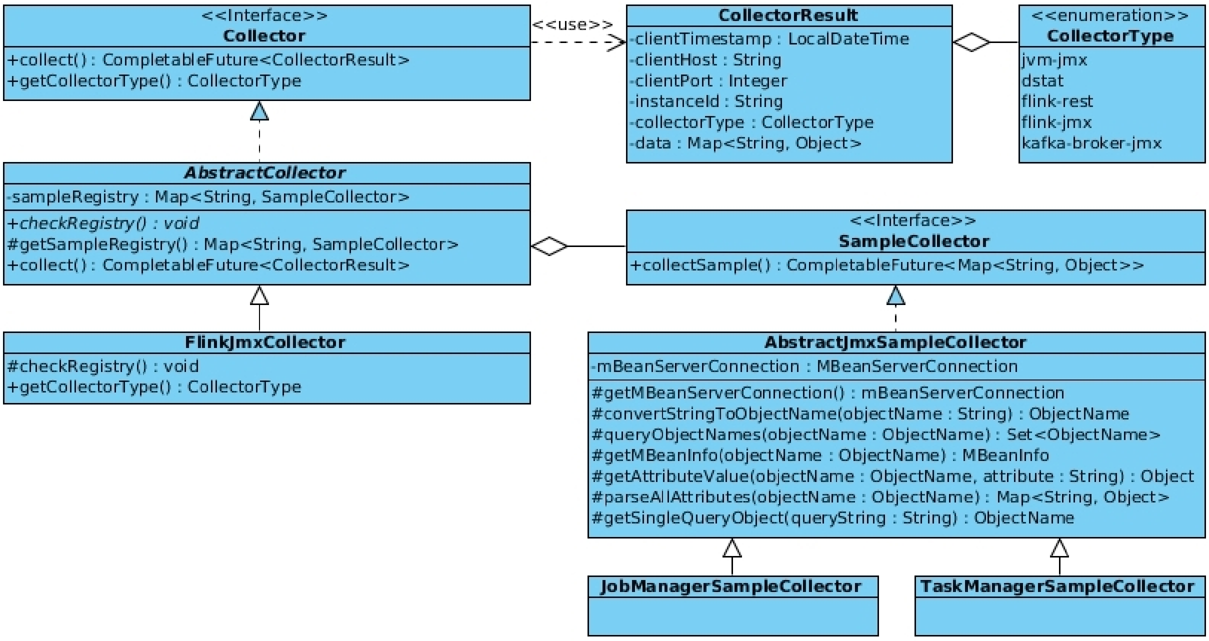

Figure A.5: Class diagram 'FlinkJmxCollector'

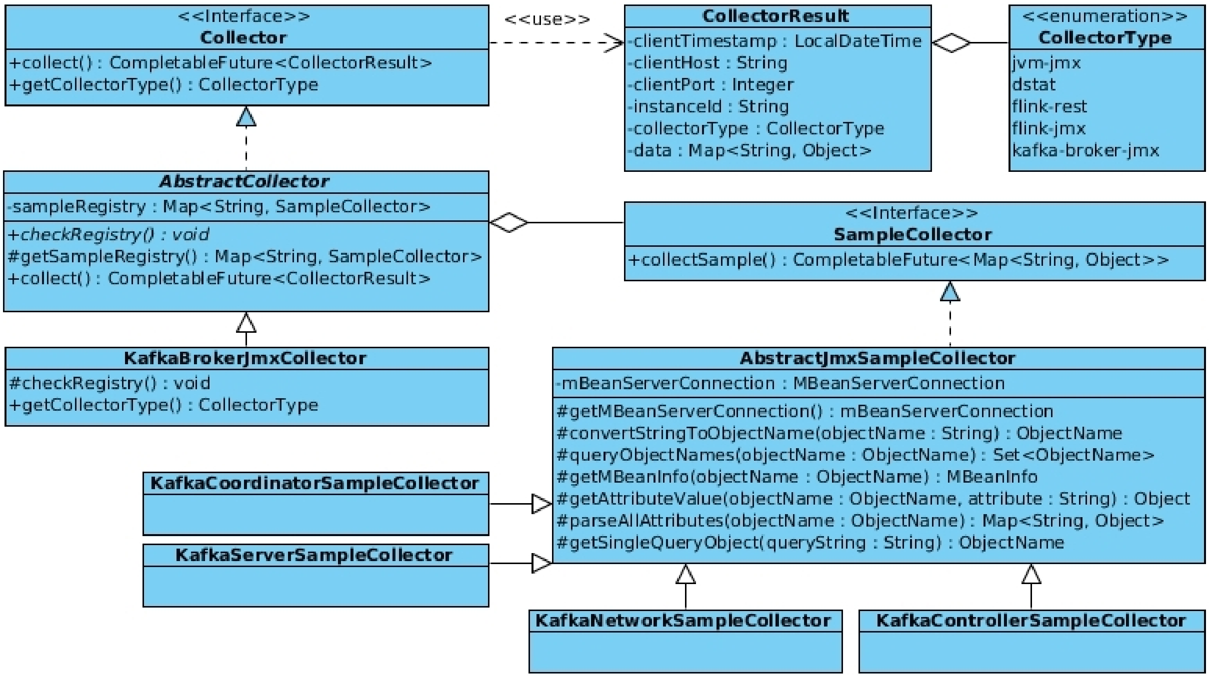

Figure A.6: Class diagram 'KafkaBrokerJmxCollector'

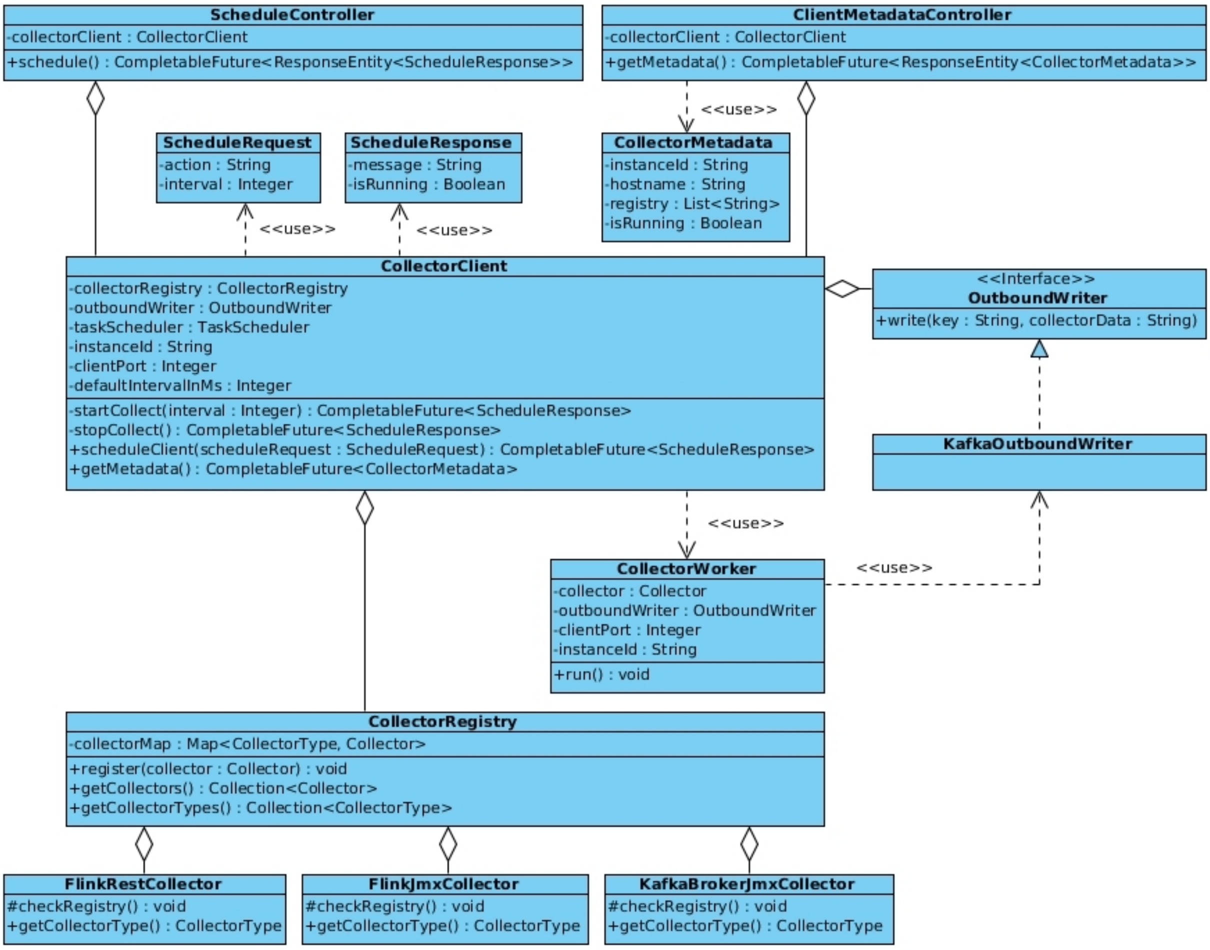

Figure A.7: Class diagram ’CollectorClient’

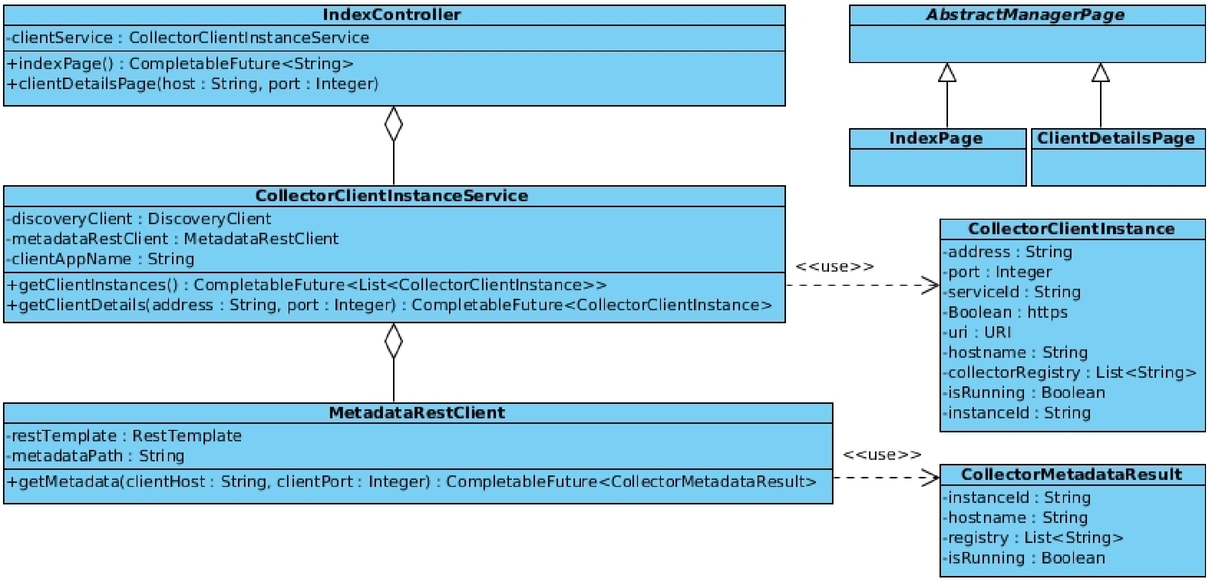

Figure A.8: Class diagram 'CollectorManager'

A.1.3 Sequence diagrams

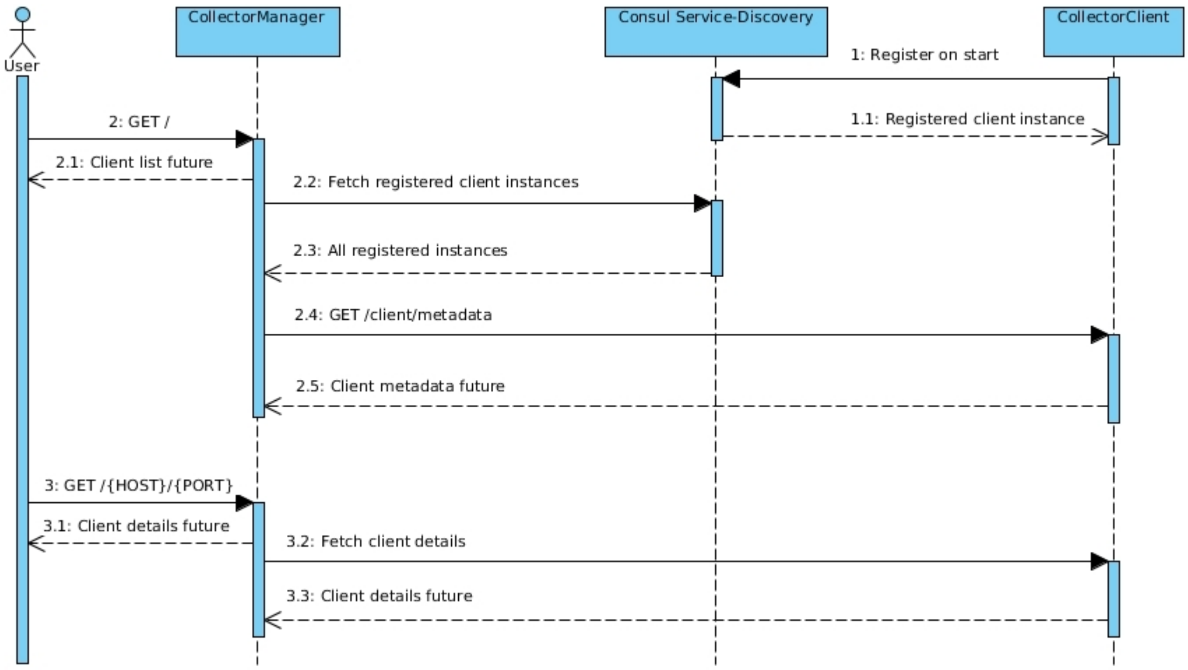

Figure A.9: Sequence diagram 'Client discovery'

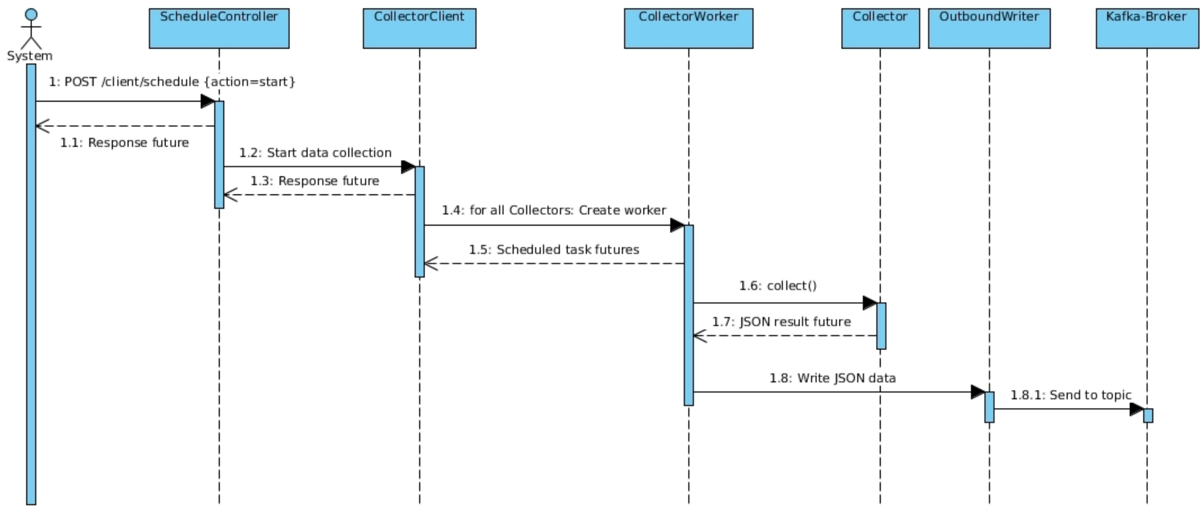

Figure A.10: Sequence diagram 'Client scheduling'

A.1.4 Component diagram

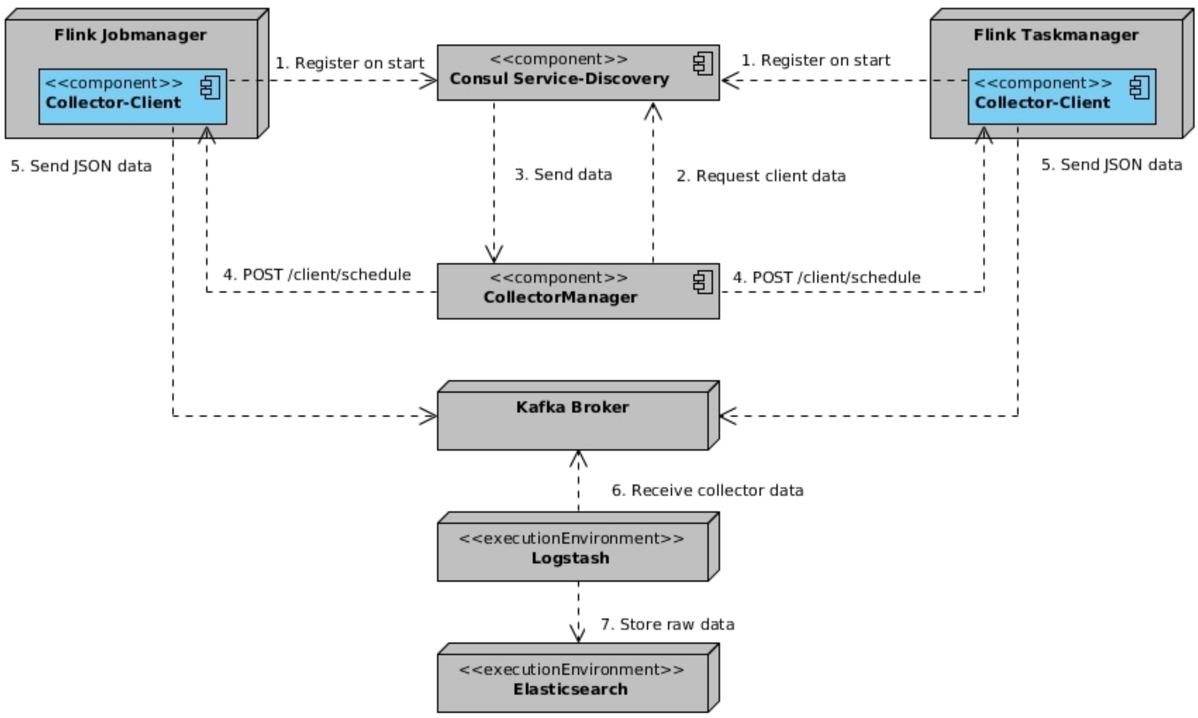

Figure A.11: Component diagram

A.1.5 Deployment diagram

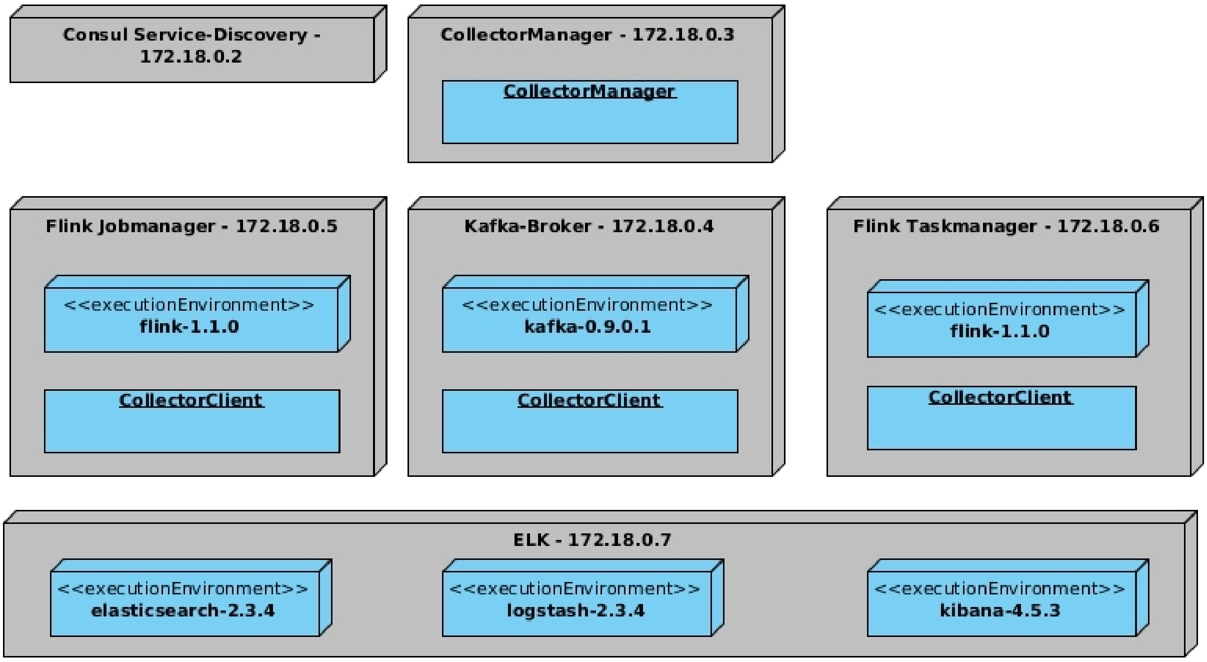

Figure A.12: Deployment diagram

A.2 Tabelle

A.3 Screenshot

A.4 Graph

Eigenständigkeitserklärung

Hiermit versichere ich, dass ich die vorliegende Masterarbeit selbstständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel verfasst habe. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt.

Stadt, den xx.xx.xxxx

Max Mustermann