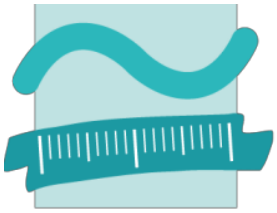

BEUTH HOCHSCHULE FÜR TECHNIK BERLIN
University of Applied Sciences

Bachelor Thesis

for obtaining the academic degree Bachelor of Science (B.Sc.) in Media Informatics,
Department VI of Beuth University of Applied Sciences Berlin

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

Markus Lamm
Matriculation number: 786697
markus.lamm@gmail.com

Berlin, 06.09.2016

Supervisor: Prof. Dr. Stefan Edlich
Consultant: Prof. Dr. Böhler

ACKNOWLEDGMENTS

ABSTRACT

Illustration Index

Index of Tables

Table of Contents

Introduction

According to a survey in Germany, nine out of ten companies (89 percent) analyze internal data for operational decision-making processes using modern Big Data Analytics Architectures, where 48 percent of respondents see the greatest potential of Big Data [BITK14]. 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 Architectures is therefore an essential tool in order to ensure successful operation. By analyzing the existing data, problems can be tracked and potential sources of error identified as early as possible.

Goals of work

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 frameworks. It should be investigated which data is available for Apache Flink and Apache Kafka . With regard to a possible user interface and further analytic-processes in the future, the measuring data must be collected on the source systems, transported and stored at a central data storage system to be accessible for other components.

Central questions of elaboration will be:

- What sources exist to gather data from Apache Kafka and Apache Flink?
- What data is relevant and shall be collected?
- How can the data be collected from existing sources
- How can the data be stored regarding to further analytics or display in a graphical user interface?

Structure of this thesis

First chapter introduction and goals

Second introduction to BDSA

Theoretical foundations

Big-Data-Analytics-Architectures

Stream-Processing

Apache Flink

Apache Kafka

Data Analysis

Java Management Extensions (JMX)

Representational State Transfer (REST)

(Dstat system utility)

Requirements and Target Definition

Architecture and Implementation

Microservices, Service Discovery

Event-Stream

System-Overview

CollectorClient

CollectorManager

CollectorDataProcessor

Message-Broker

Indexer

Persistence

Evaluation

Conclusion

Appendix

Use-Case diagram

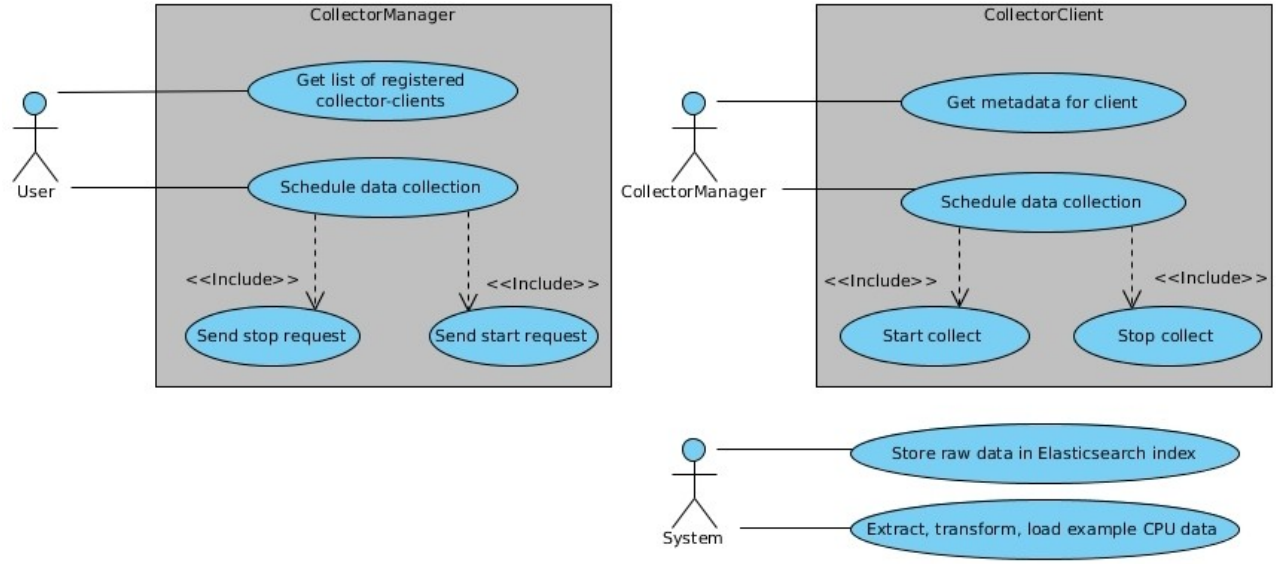

Illustration 1: Use-Case diagram

Class diagrams

CollectorClient

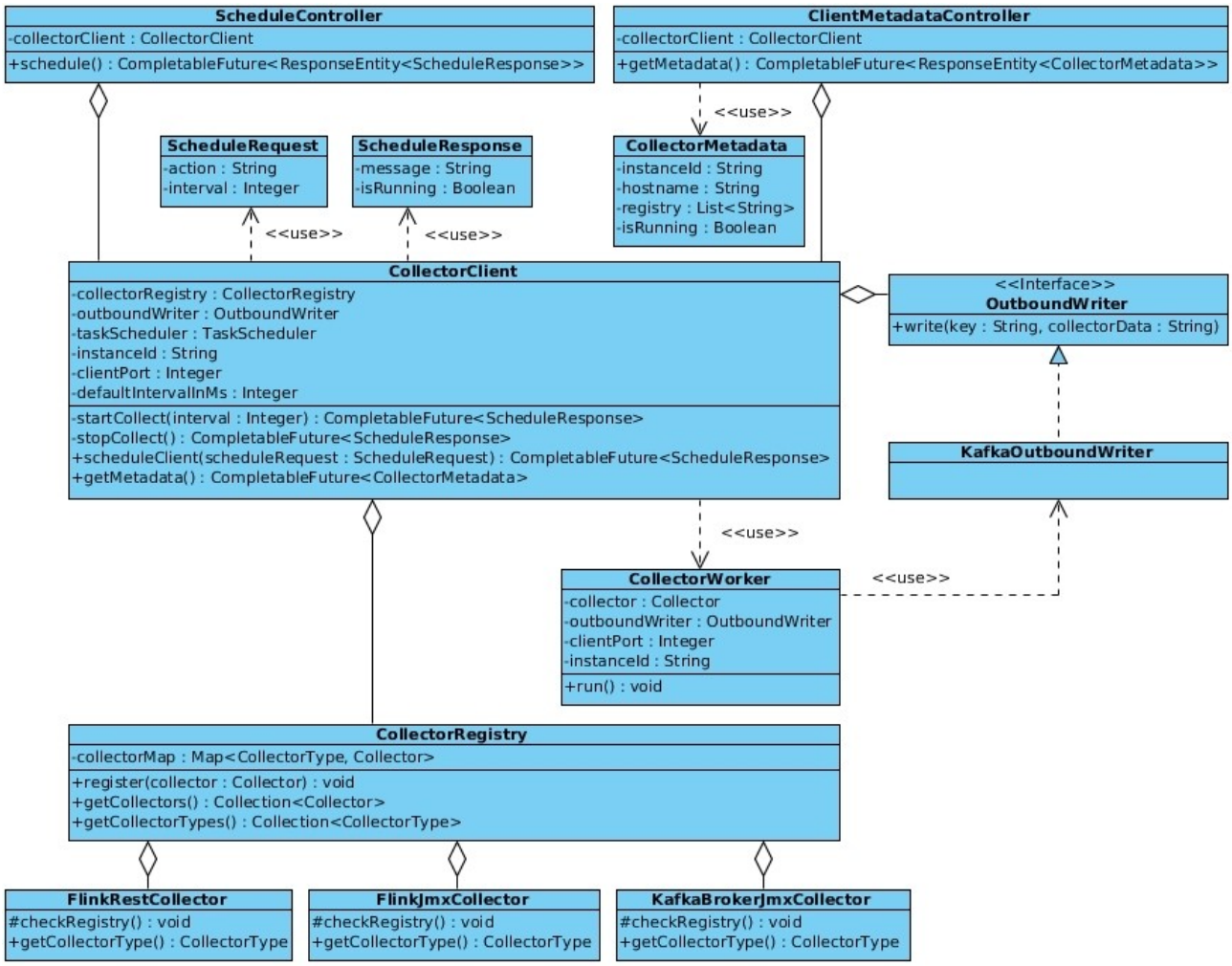

Illustration 2: Class diagram CollectorClient

CollectorManager

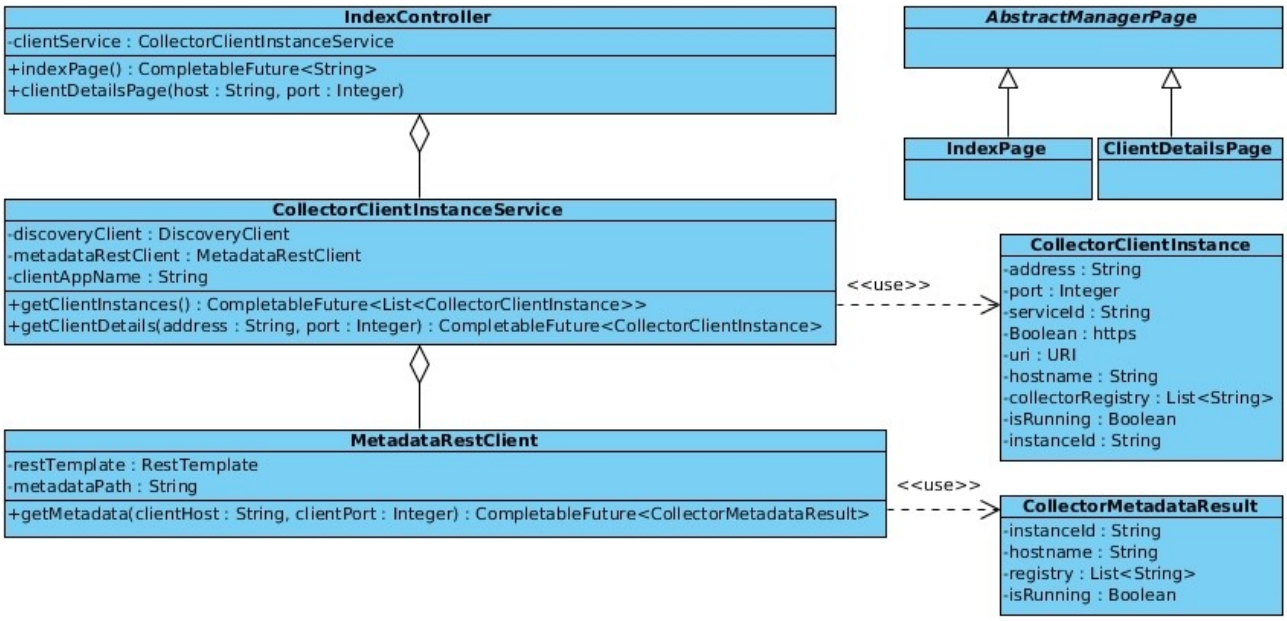

Illustration 3: Class diagram CollectorManager

JvmCollector

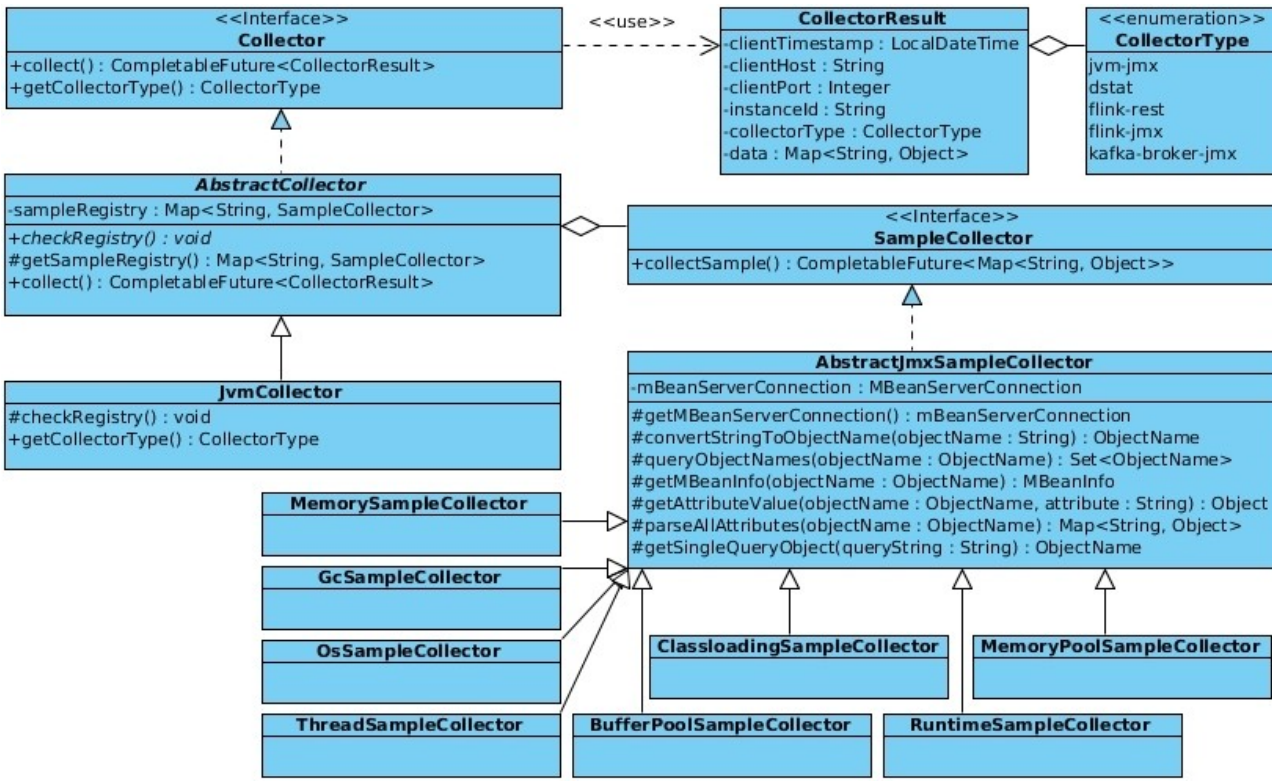

Illustration 4: Class diagram JvmCollector

DstatCollector

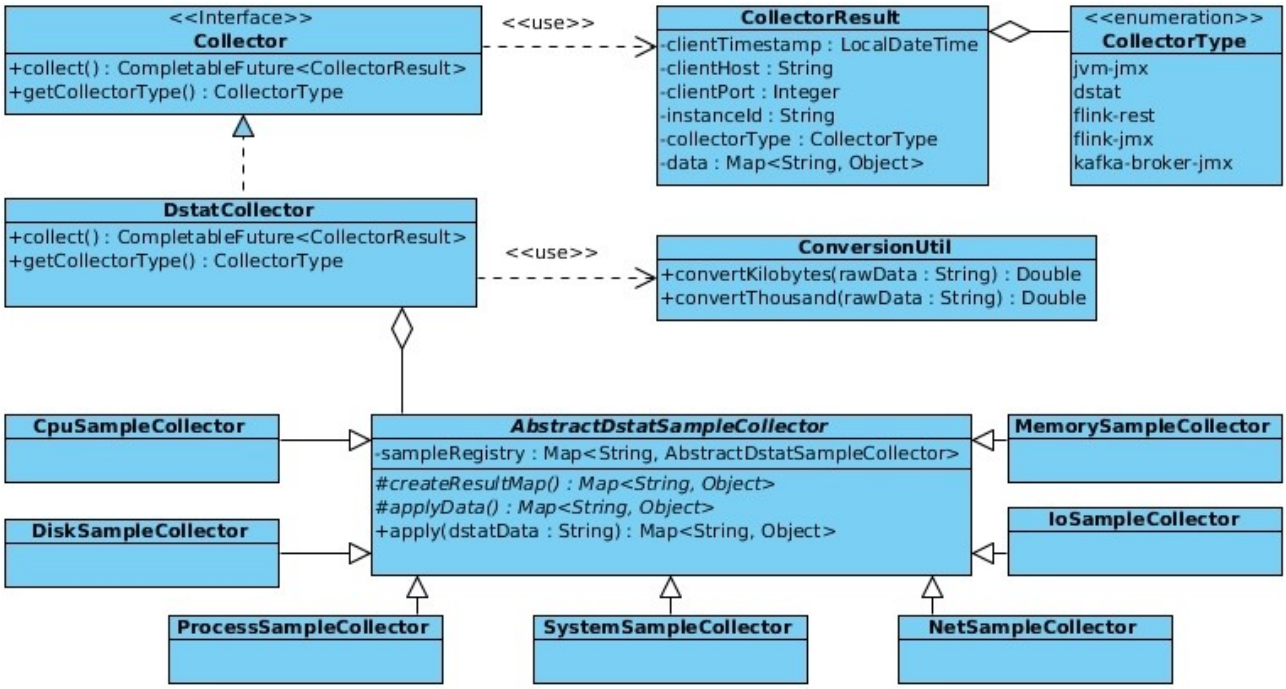

Illustration 5: Class diagram DstatCollector

FlinkJmxCollector

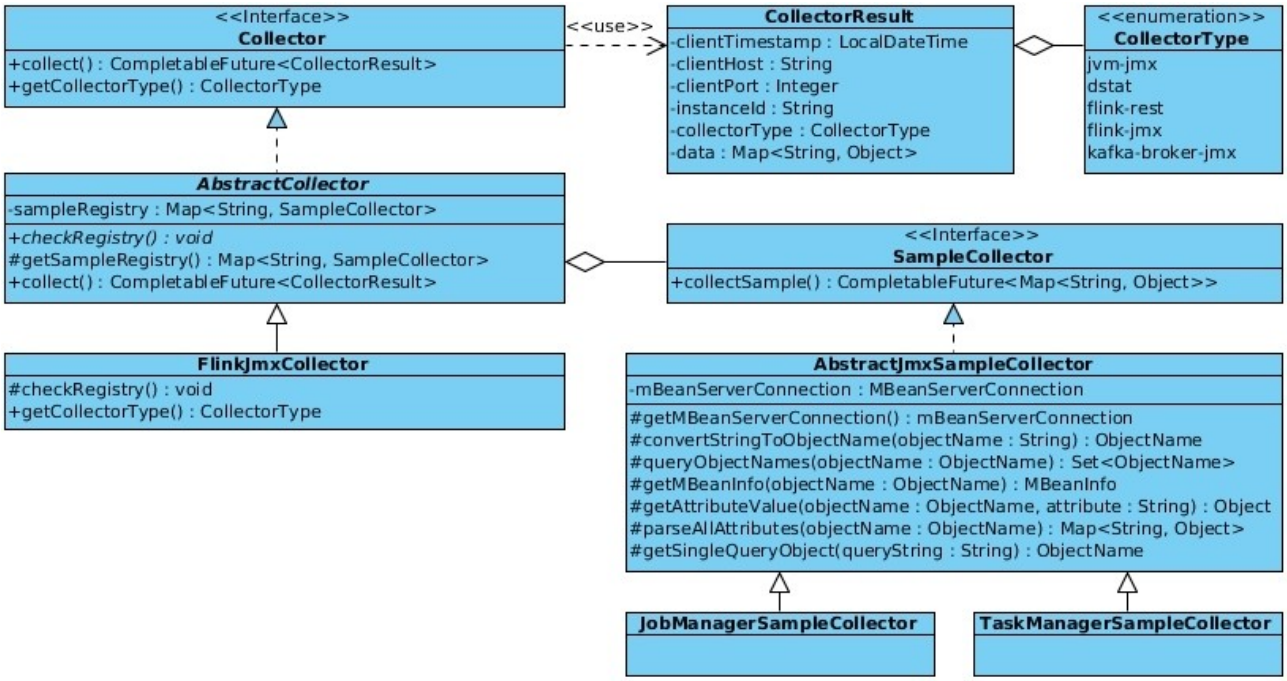

Illustration 6: Class diagram FlinkJmxCollector

FlinkRestCollector

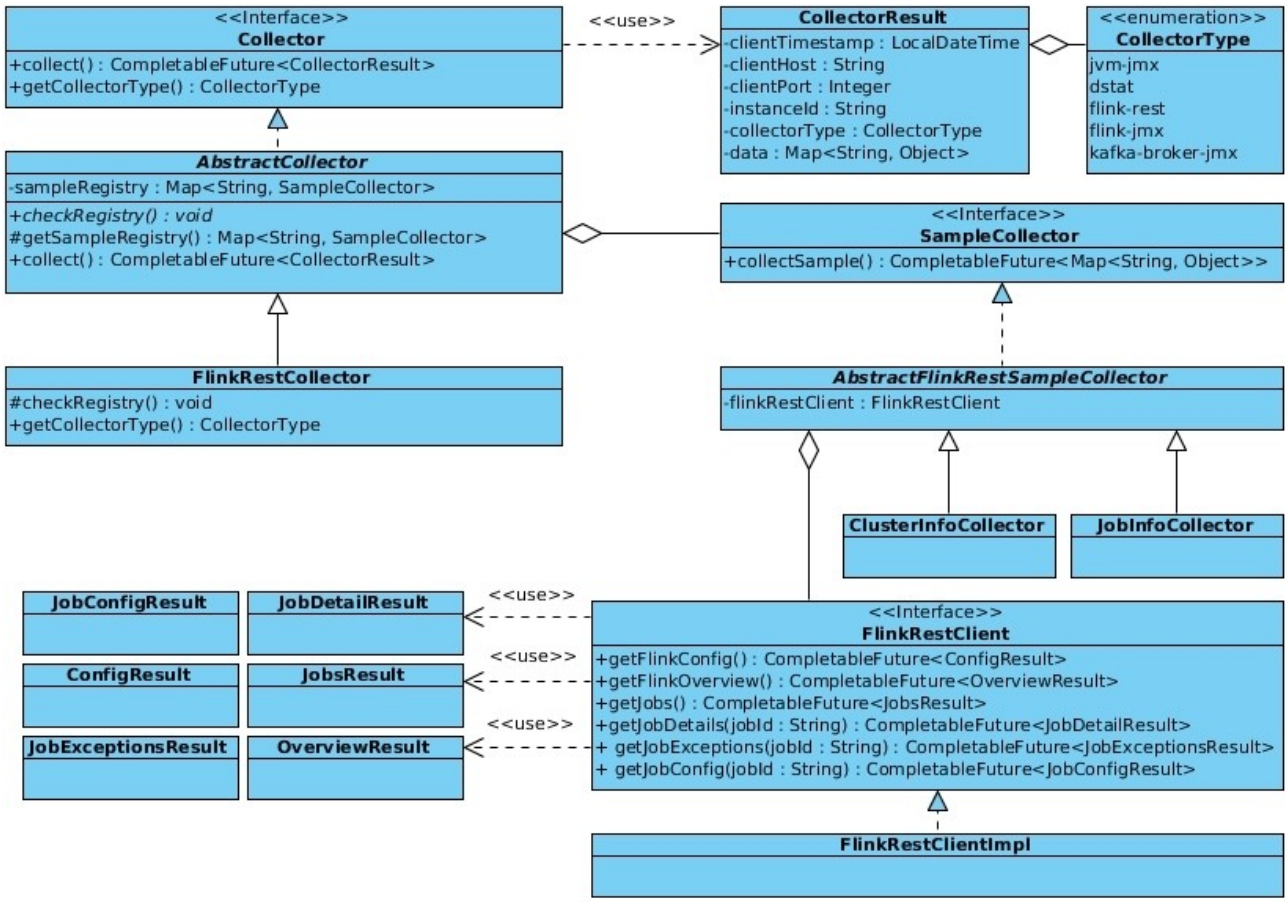

Illustration 7: Class diagram FlinkRestCollector

KafkaBrokerJmxCollector

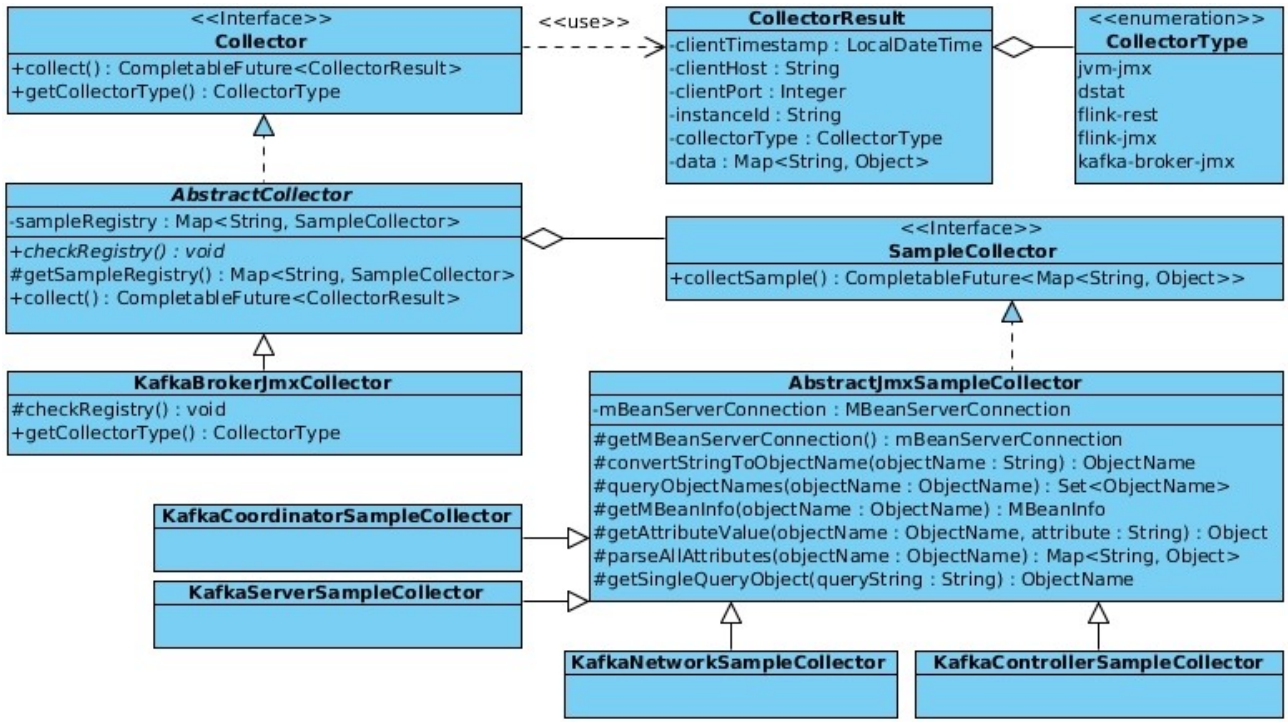

Illustration 8: Class diagram KafkaBrokerJmxCollector

Component diagram

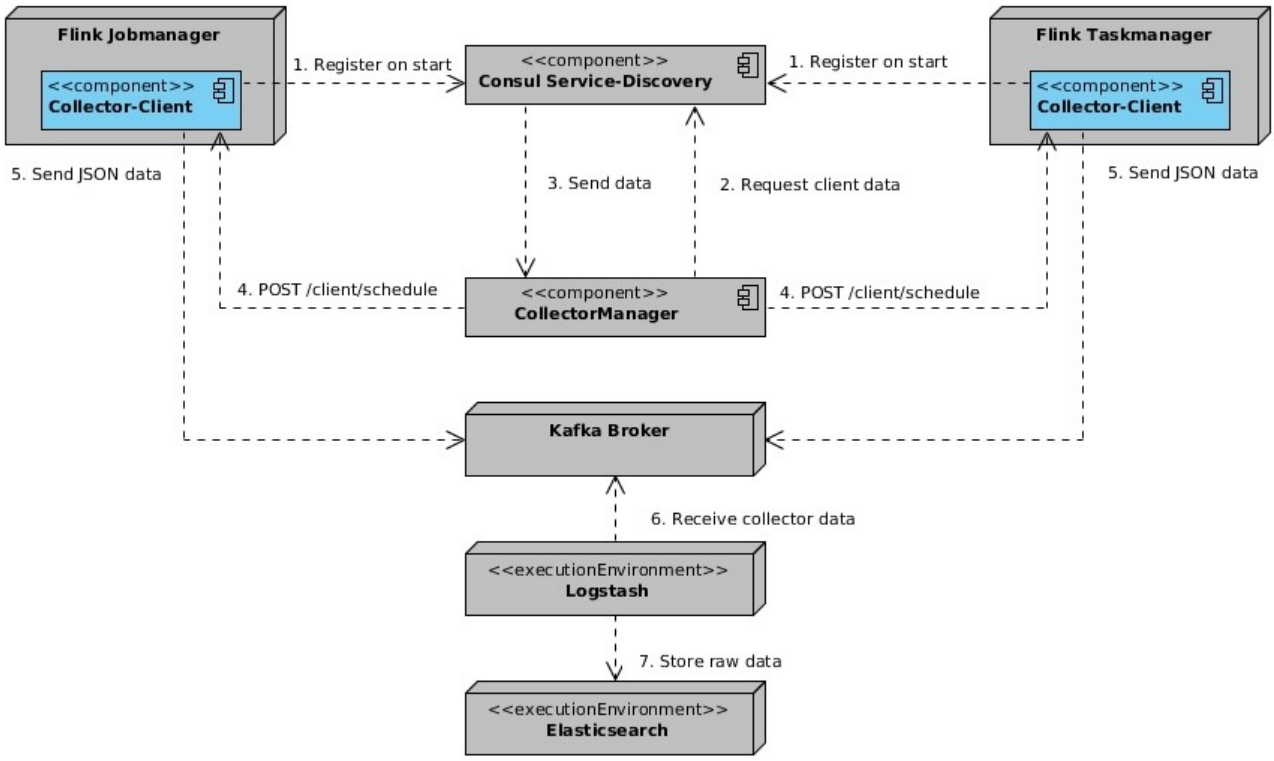

Illustration 9: Component diagram

Sequence diagrams

Client discovery

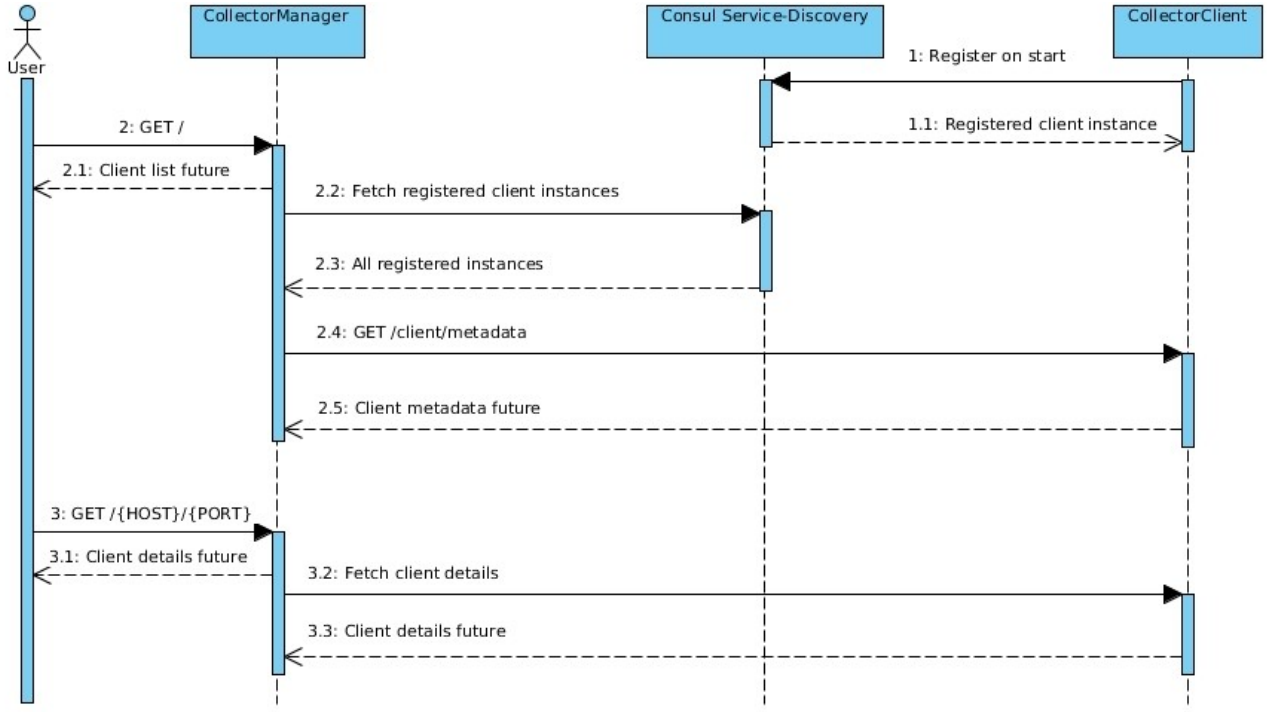

Illustration 10: Sequence diagram Client discovery

Client scheduling

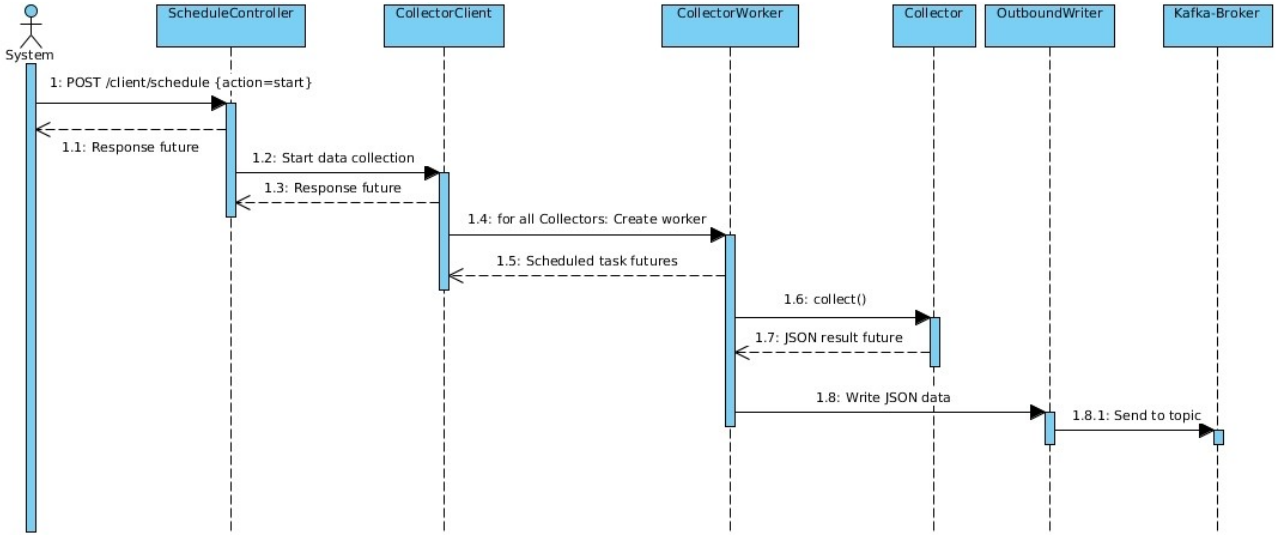

Illustration 11: Sequence diagram Client scheduling

Deployment-Diagram

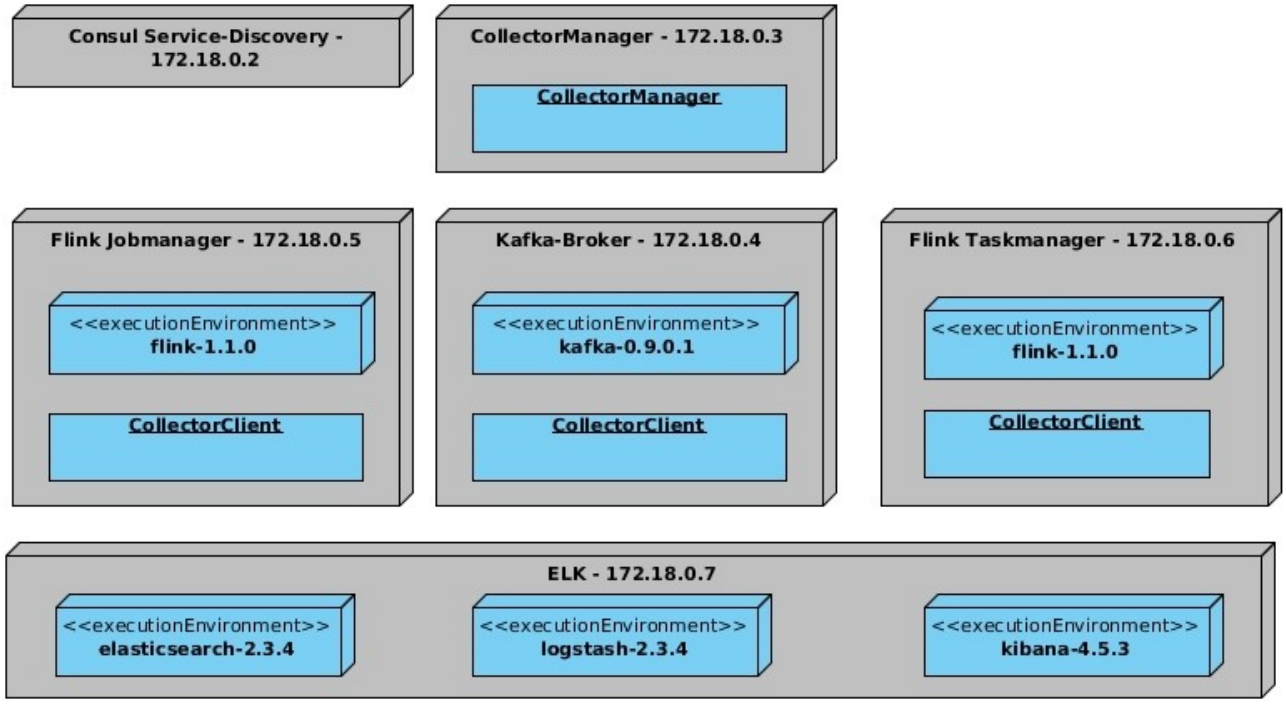

Illustration 12: Deployment diagram

Setup instructions for the prototype application