Towards Unified Data Ingestion and Transfer: A Comparison of Apache Kafka and Rucio

Muhammad Arslan Tariq
arslan.tariq@uni.lu

UNIVERSITY OF LUXEMBOURG

Outline

- Introduction
- Objectives
- Related Work
- Data Ingestion and Transfer Use Cases
- Evaluation Methodology
- Results
- Conclusion
- Future Research Proposal

Introduction

Businesses need to process and analyse data to achieve growth and success.

Large-scale distributed systems process massive amounts of data in parallel.

Data ingestion and transfer are the process of collecting and transferring data to these systems.

The challenges of data ingestion and transfer include volume, velocity, variety, and veracity.

Addressing these challenges is critical for businesses to derive meaningful insights from data and stay competitive.

Objectives

- Introduction of Apache Kafka and Rucio
- Overview of Data transfer and Data ingestion platforms
- Conduct a benchmark comparison evaluation of data transfer and data ingestion
- Demonstrate the potential benefits of small records files of data transfer and data ingestion
- Analyse the potential benefits of unifying data transfer and data ingestion technologies

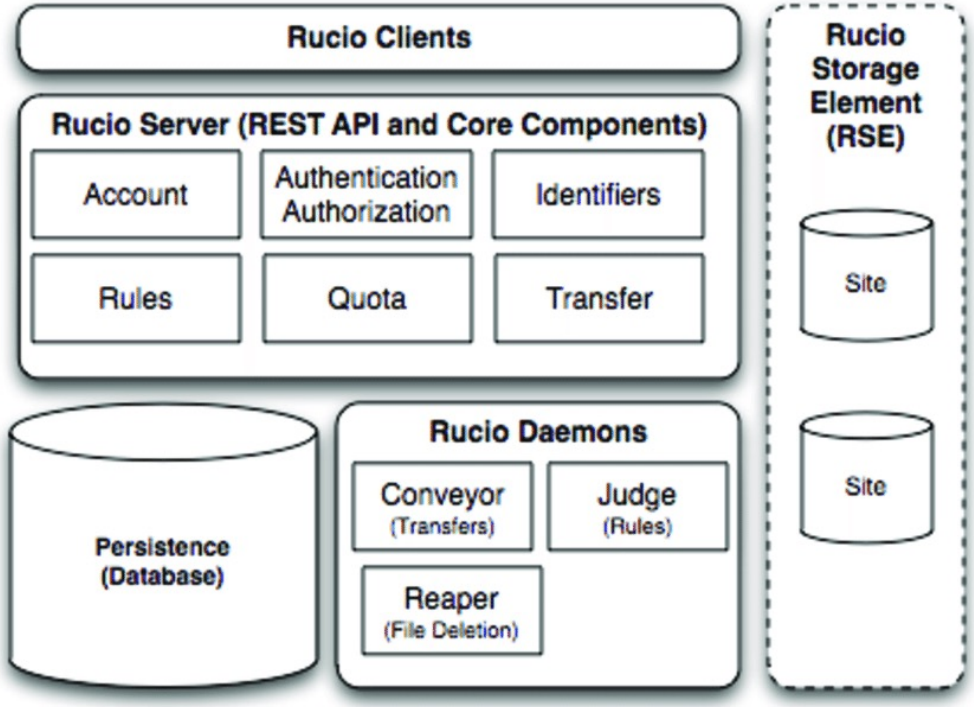
Related Work

- Data Ingestion tools like Apache Kafka, Apache Pulsar, Amazon Kinesis
- Data Transfer tools i.e. Globus, GridFTP, Rucio etc

- Our Focus:
  - Open Source
  - State-of-the-art
  - Easy to deploy
  - Containerized Environment
  - Interoperability

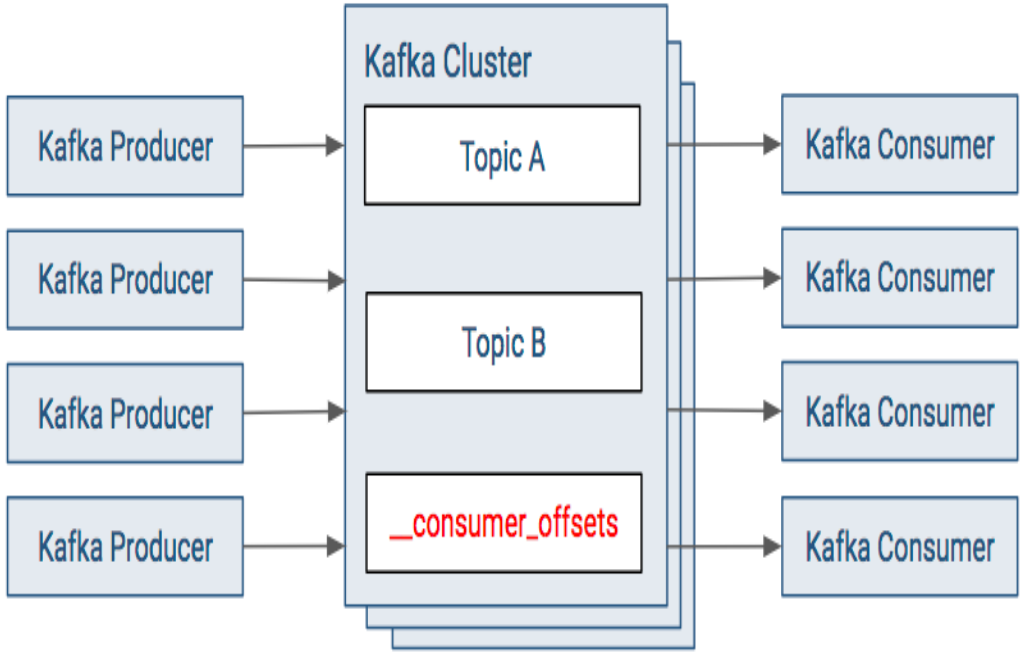
Overview of Rucio

- Rucio is a Distributed Data Management System
  - Unified interfacing of heterogenous network & storage infrastructure
  - Adaptive Replication
  - Dataflow autonomy
- Designed to handle metadata management, data replication and distribution, and data access control
- Scientific researchers can manage their data more effectively, reducing the time and effort required for data management



Overview of Apache Kafka

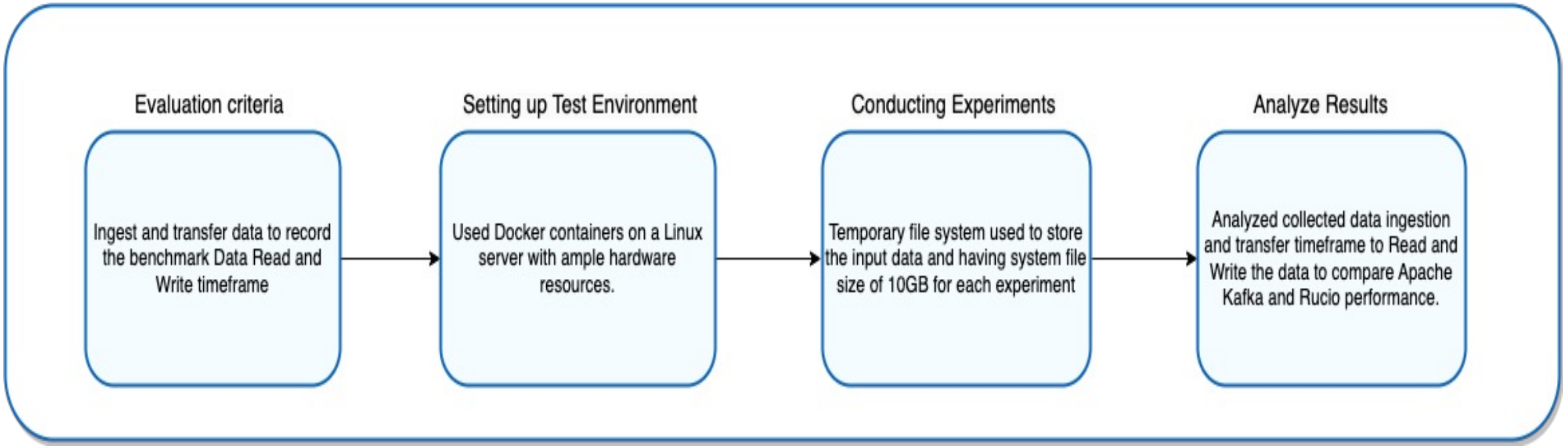
- Apache Kafka is a distributed streaming platform
  - High Scalable (partition)
  - Fault Tolerant (replication)
  - Allow high level of parallelism and decoupling between data producers and data consumers
- De facto standard for near real-time store, access and process data streams
- Critical component of most of the Big Data Platform and Hadoop ecosystem



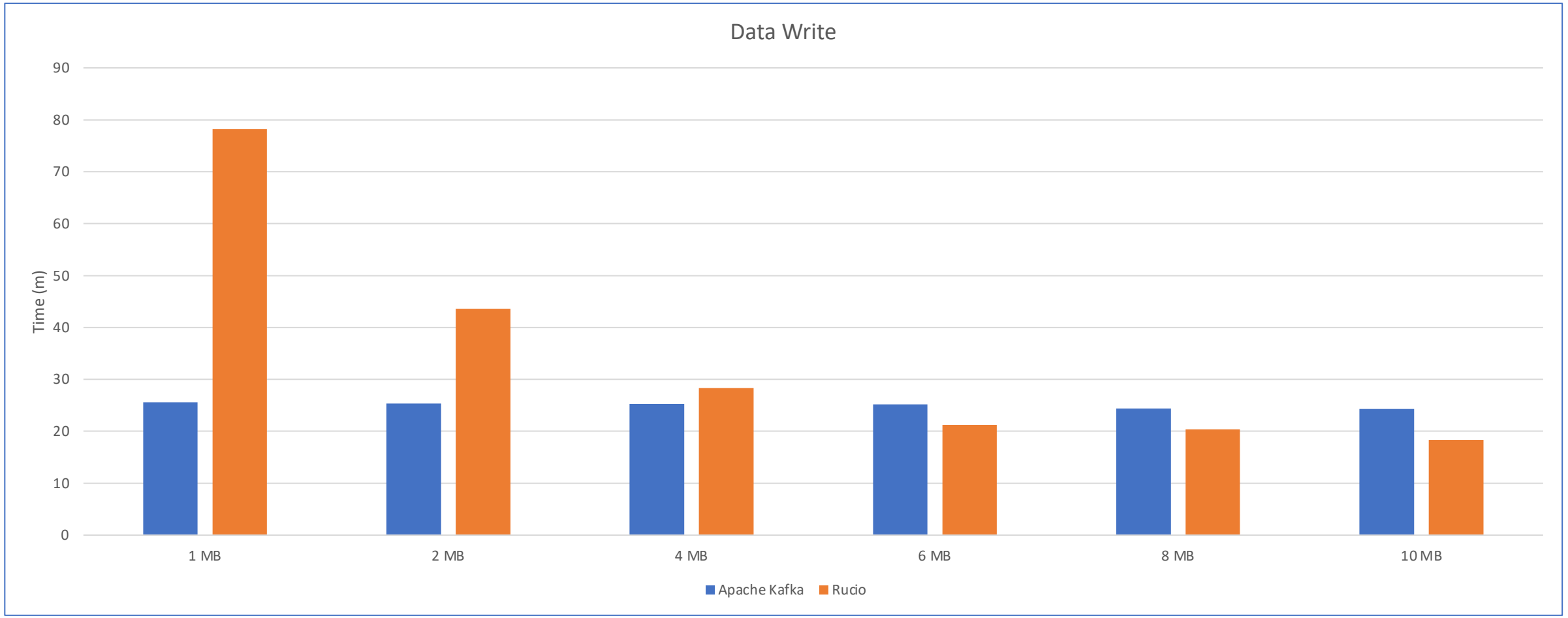
Data Ingestion and Transfer Use Cases

- Bridging Cloud and HPC
- Food Security
- Cloud data lake
- Data warehouse modernization
- Real-time analytics
- Logistics and IoT Management
- Real-time Alerts and Notifications
- Infrastructure Modernization

Evaluation Methodology

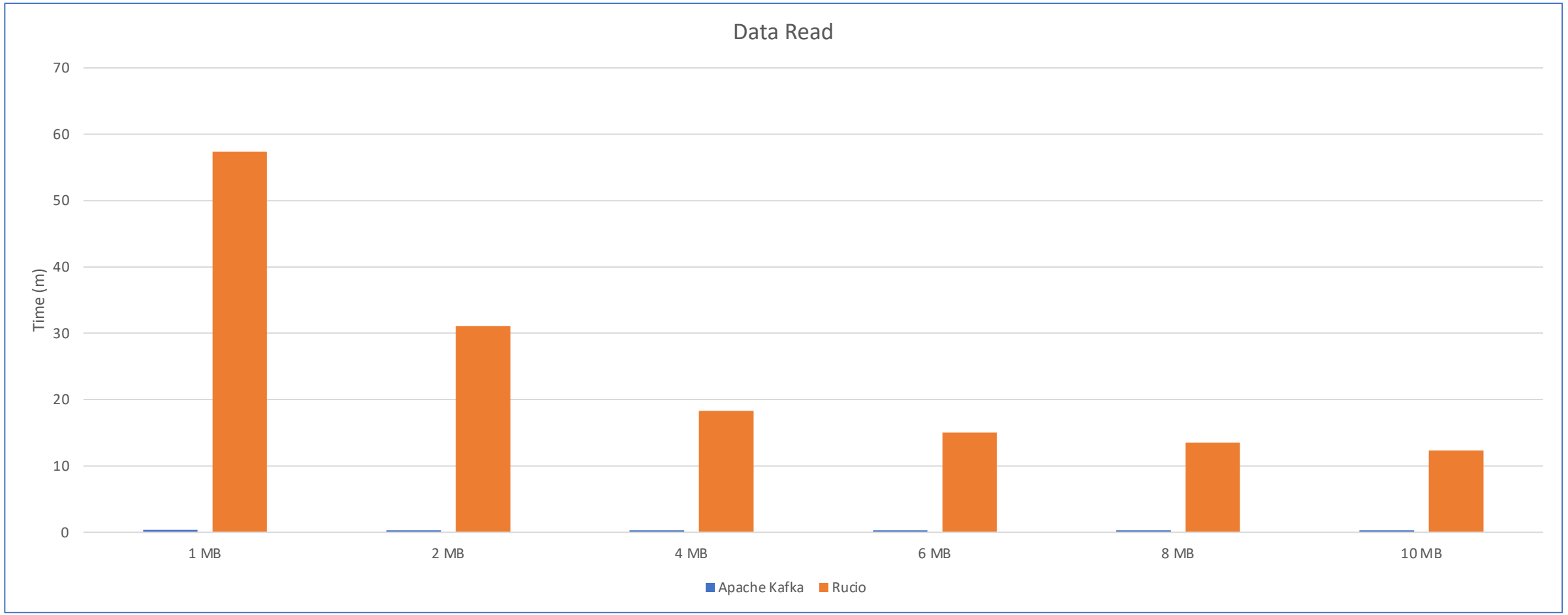


Timeframe for Writing Data in Apache Kafka vs Rucio



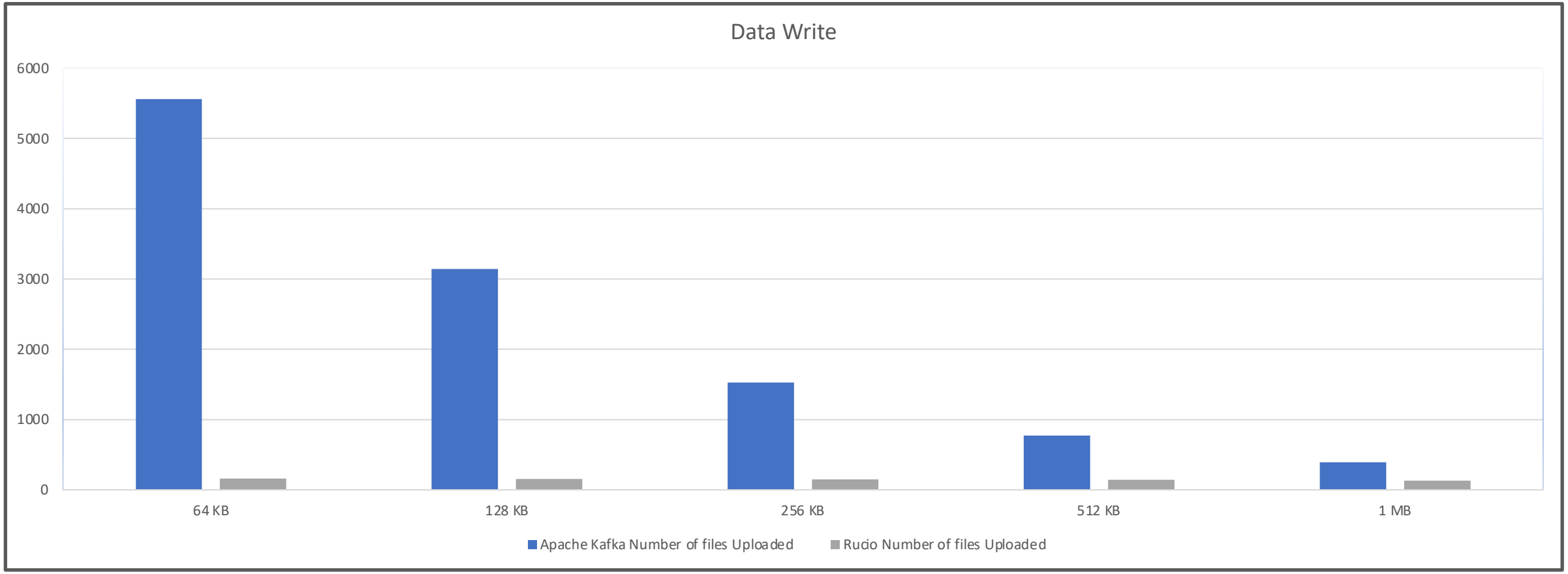
Lower is Better

Timeframe for Reading Data from Apache Kafka vs Rucio



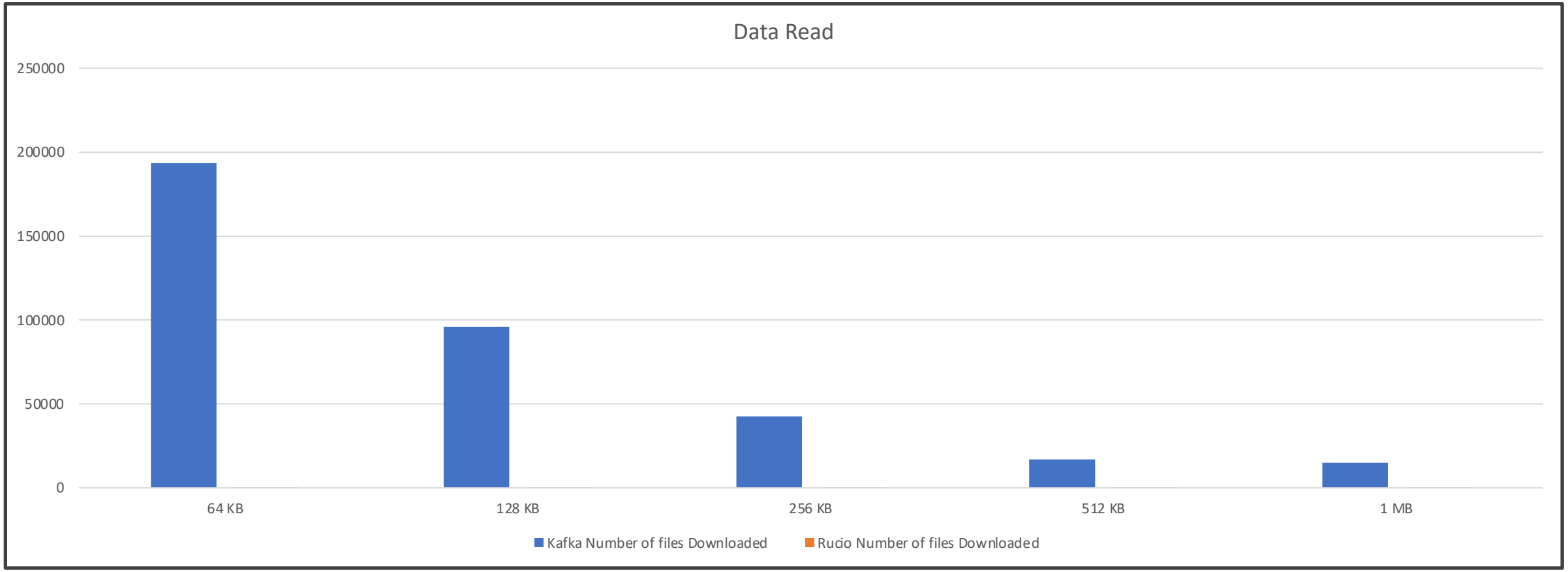
Lower is Better

Measuring Number of Files Written per Minute in Apache Kafka vs Rucio



Higher is Better

Measuring Number of Files Read per Minute in Apache Kafka vs Rucio



Higher is Better

Conclusion

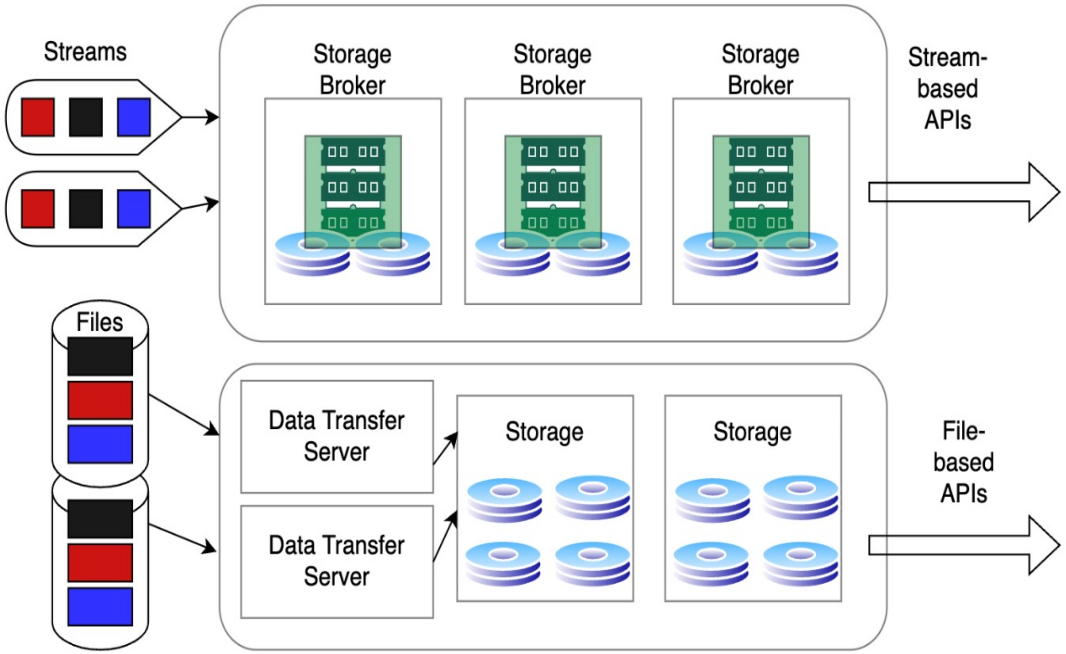
In conclusion, both Apache Kafka and Rucio are powerful technologies that enable unified data ingestion and transfer. After conducting a comprehensive evaluation of both technologies, we have identified some key takeaways:

- Apache Kafka excels in performance, as shown in results that it a great choice for smaller data file sizes. Even with few megabytes file size, it is still very competitive as compared to rucio.
- Rucio offers a high degree of reliability, and ease of use. But it shows better read and write results with large file sizes i.e. tens of megabytes to petabytes.
- When deciding between the two technologies, it's important to consider factors such as the nature of the data you're working with, the size and complexity of your data pipeline, and the specific requirements of your use case.

Future Work Proposal

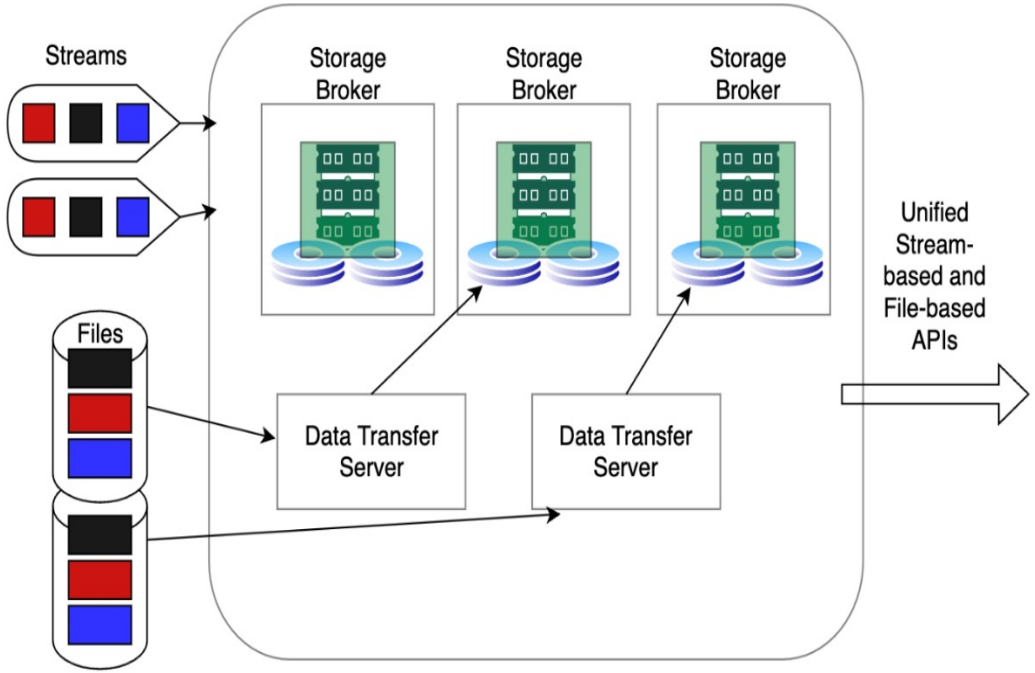
State-of-the-art Rucio and Kafka

To orchestrate stream and file data transfers, we can leverage two different deployments and APIs for stream ingestion (top) and files transfers (bottom)



Our proposal:

***Idea*: Unified Storage and APIs** *for stream-based and file-based data transfers*



Questions?