

Real-Time Smart Bank Data Streaming Capture

Gourp 03

Team Members

Sindhu Nagesha - 017419987

Prayag Nikul Purani - 017416737

Syed Faraaz Ahmed - 017428619

Sai Vivek Chunduri- 017435301

Overview


01 Abstract
02 Introduction &
Motivation
03 Literature Survey

04 Tech Stack &
System Design
05 Source 1 (Transaction data)
06 Source 2 (Application data)
07 Source 3 (Sensor Data)

08 Result
09 Conclusion
10 Code demo



Abstract

- Developing a real-time data analytics pipeline for the banking sector.
 - Utilizing Docker, Kafka, and Apache Spark.
 - Gathering data from sensors, transactions, and applications.
 - Enabling swift processing and integration with Flask and Elastic search.
 - Providing actionable insights and enhancing customer satisfaction.
- 

A decorative, light gray wavy line starts from the top left corner and curves towards the right, ending under the 'I' of the title 'Introduction'.

Introduction

- This project aims to redefine how banks leverage data through two workflows: one involving sensor data and the other application and transactional data.
- Sensor data is processed via Amazon IoT Core, Timestream, Python, and Grafana for fraud detection and visualization.
- Application and transactional data are streamed through Python, Kafka, MongoDB, and Kibana, utilizing Apache sessions and complex logic for fraud detection.
- Real-time data processing empowers banks to detect patterns, anticipate trends, and make precise decisions swiftly.
- Smart bank data streaming captures the potential to transform customer experiences, enabling hyper-personalized services and enhancing engagement and loyalty.



Motivation

- Real-time data analytics in banking enhances customer experience.
- It enables prompt identification and prevention of fraud.
- Improves operational efficiency by adapting to market dynamics.
- Provides a competitive edge through innovative services.
- Represents a strategic shift towards a data-centric approach, transforming the industry landscape.

Literature Survey

- 01** Apache Spark: A Big Data Processing Engine
Eman Shaikh, Iman Ahmed Mohiuddin,
Yasmeen Alufaisan and Irum Nahvi (2019).
Apache Spark: A Big Data Processing Engine.
- 02** Information Security in Big Data: Privacy and
Data Mining Xu, L., Jiang, C., Wang, J., Yuan, J.
and Ren, Y. (2014). Information Security in Big
Data: Privacy and Data Mining.
- 03** Beyond Batch Processing: Towards Real-Time
and Streaming Big Data. Shahrivari, S. (2014).
Beyond Batch Processing: Towards Real-Time
and Streaming Big Data. Computers
- 04** Real-Time Processing of Big Data Streams:
Lifecycle, Tools, Tasks, and Challenges. F.
Gürcan and M. Berigel, "Real-Time Processing of
Big Data Streams: Lifecycle, Tools, Tasks, and
Challenges," 2018 2nd International Symposium
on Multidisciplinary Studies and Innovative
Technologies (ISMSIT), Ankara, Turkey, 2018
- 05** KAFKA: The modern platform for data
management and analysis in big data domain. R.
Shree, T. Choudhury, S. C. Gupta and P. Kumar,
"KAFKA: The modern platform for data
management and analysis in big data domain,"
- 06** Event-Based Sensor Data Scheduling: Trade-
Off Between Communication Rate and
Estimation Quality J. Wu, Q. -S. Jia, K. H.
Johansson and L. Shi, "Event-Based Sensor Data
Scheduling: Trade-Off Between
Communication Rate and Estimation Quality,"

TECHNOLOGY STACK



Python



Flask



AWS
IoT core



AWS
Lambda

Streaming



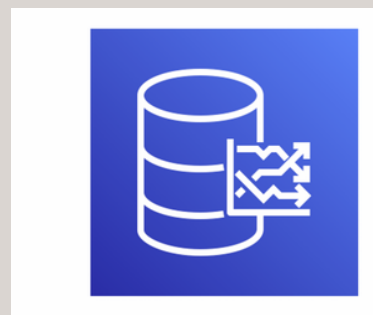
Storage



Co-ordination



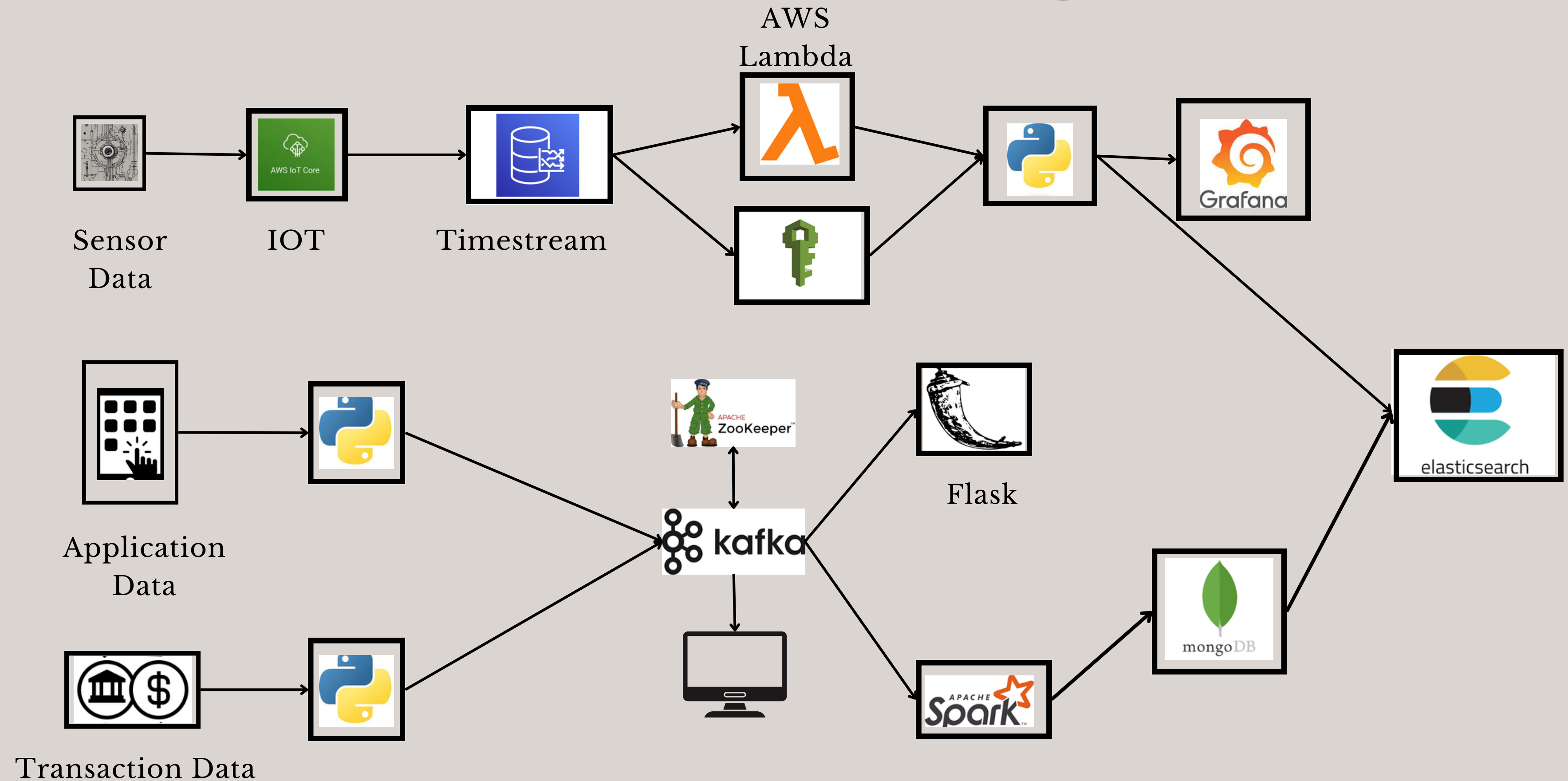
Timestream



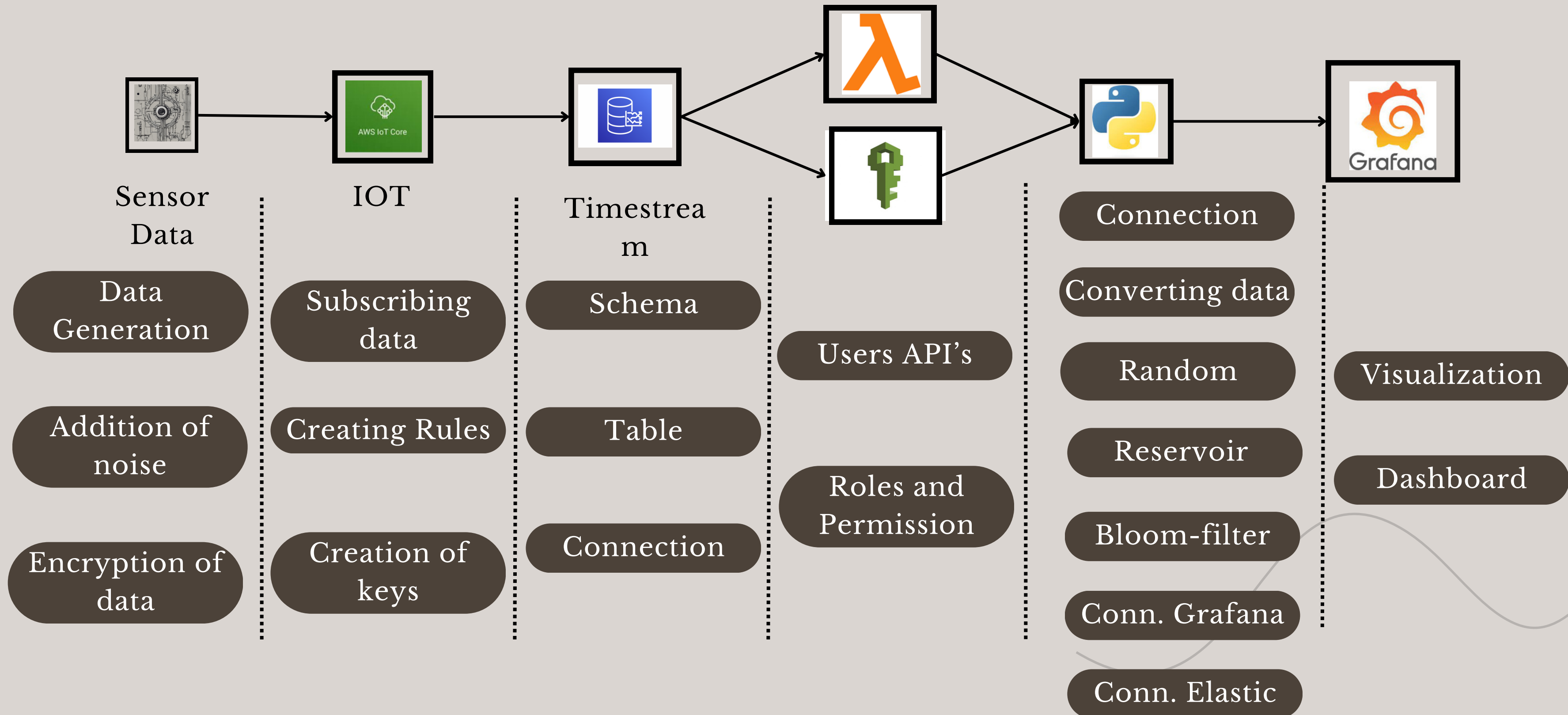
Analytics



Architecture Diagram



Work-flow 1 (Sensor)



Result

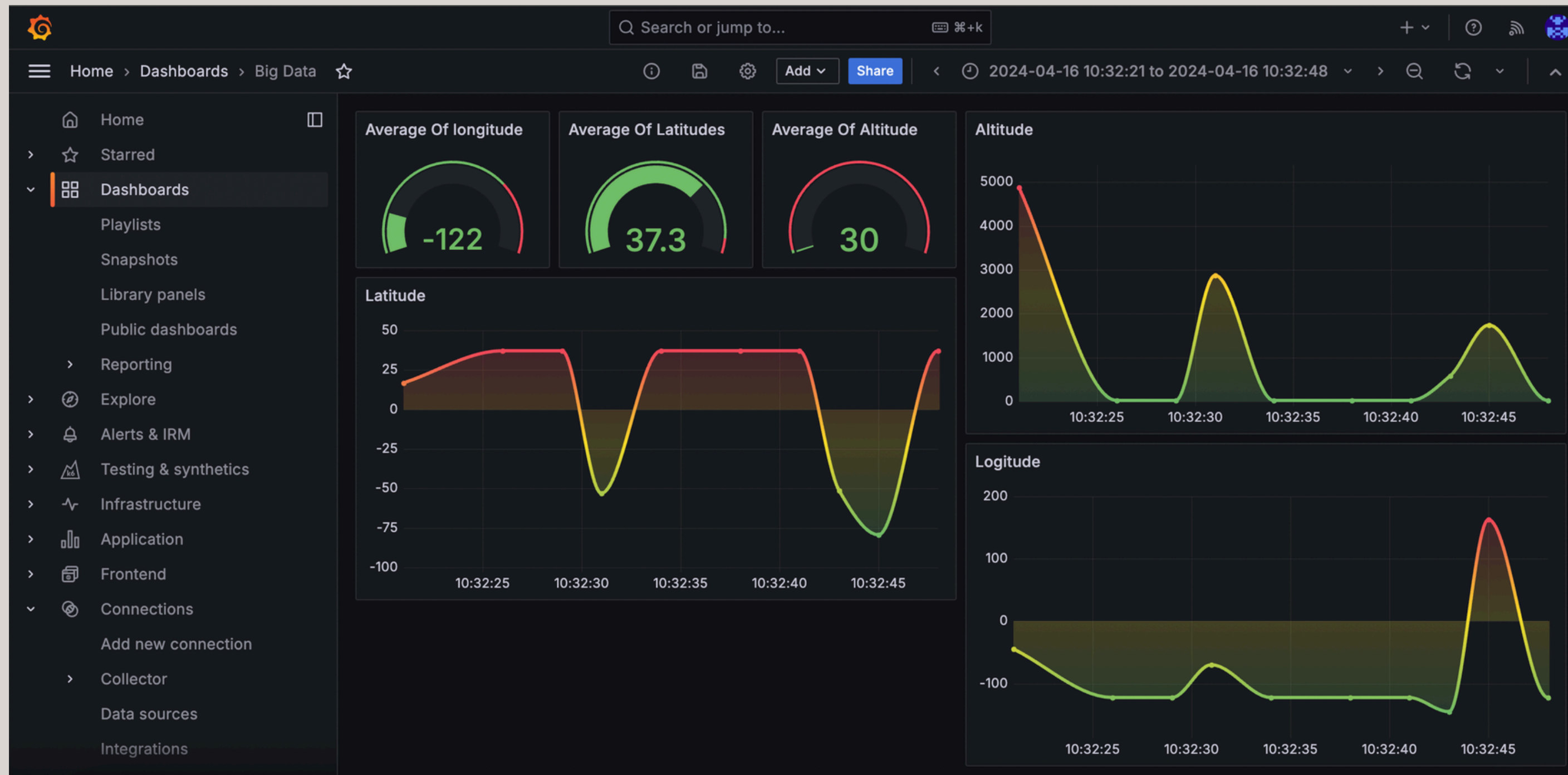


Fig: Dashboard using Grafana For Workflow 1

Result

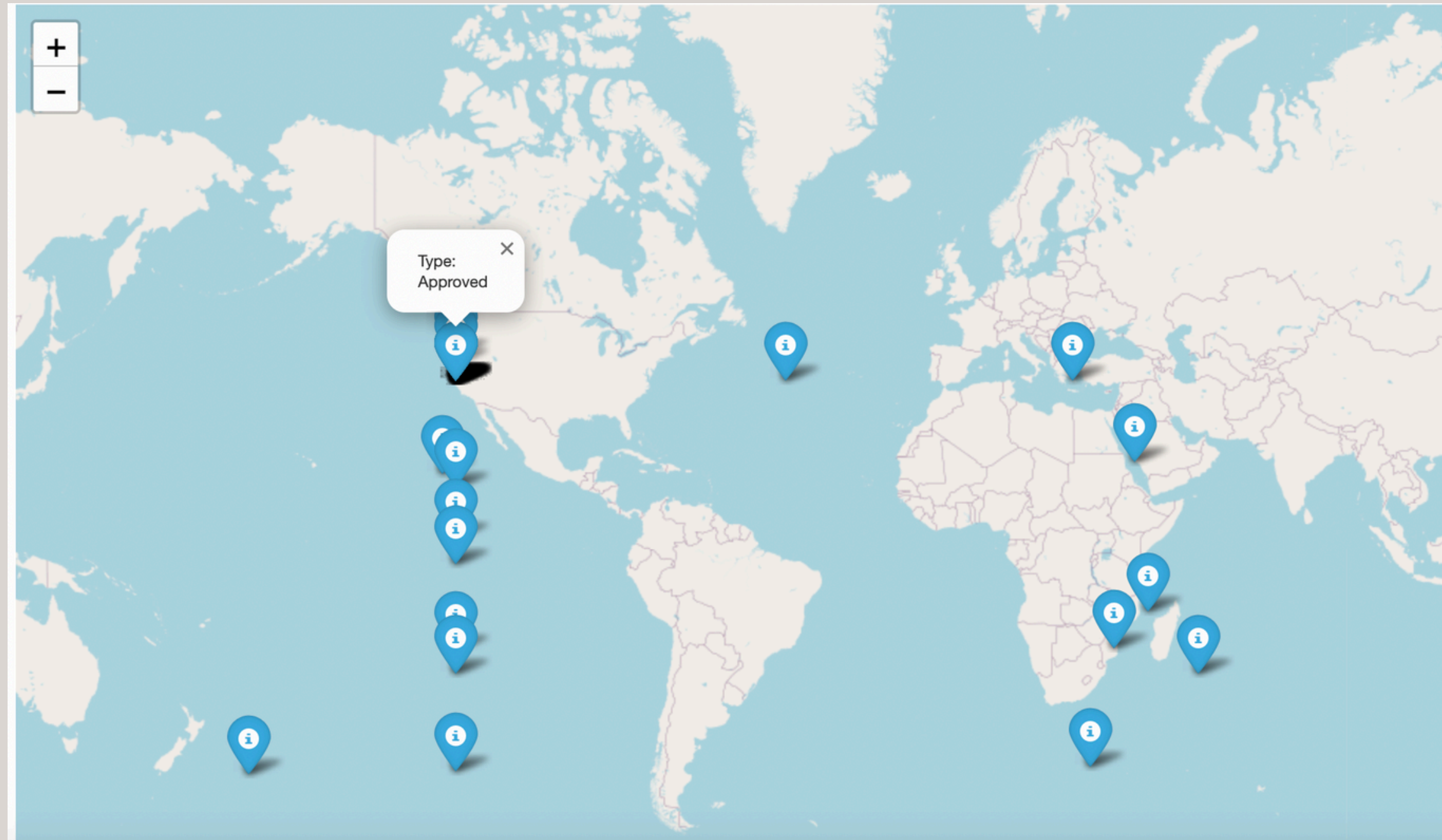
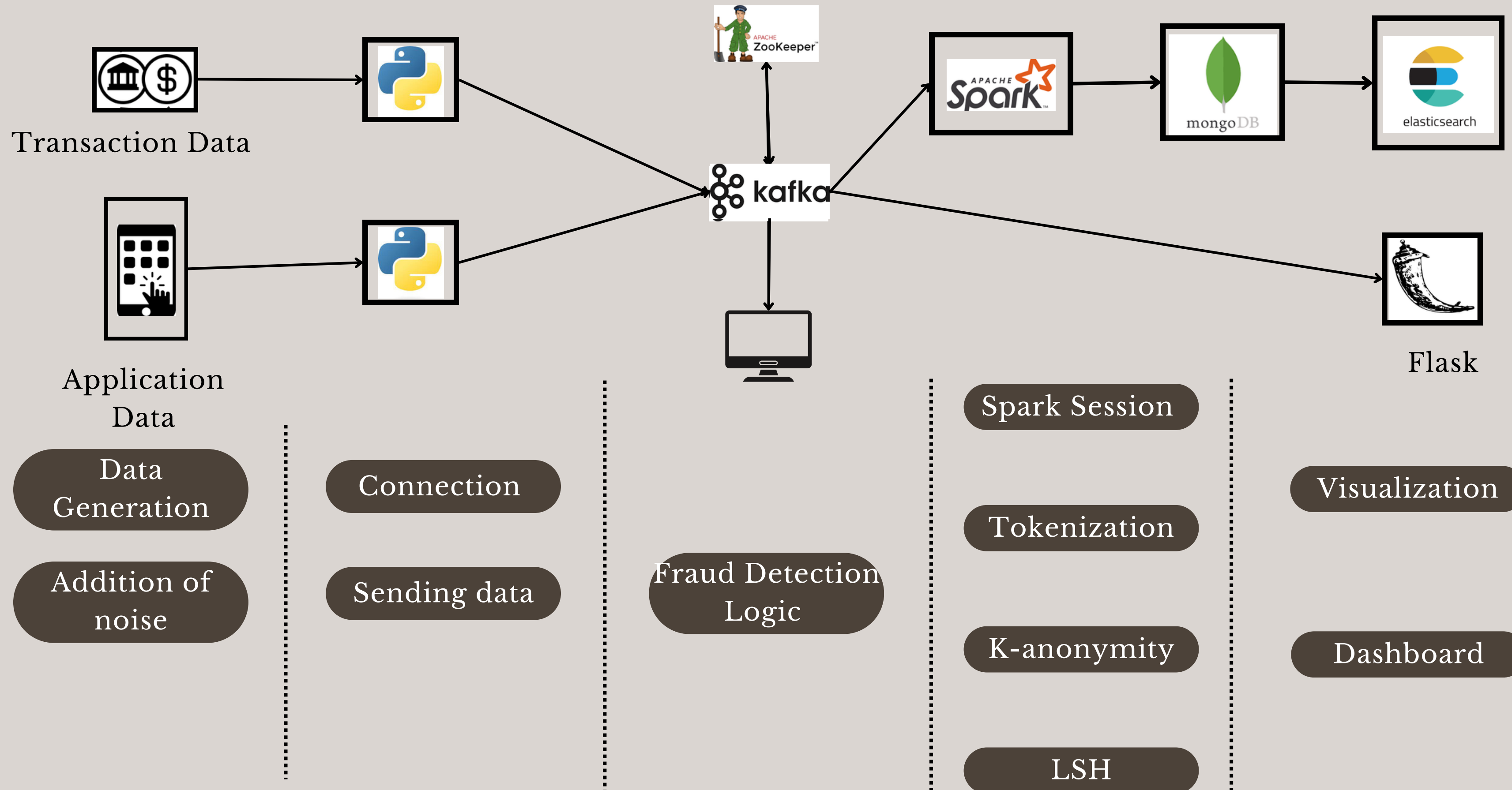


Fig: Ploting the Samples

Work-flow 2 (Transaction & App)



Result

Live Location Data Streamed from Sensor Logger

Live Location Tracking

Map

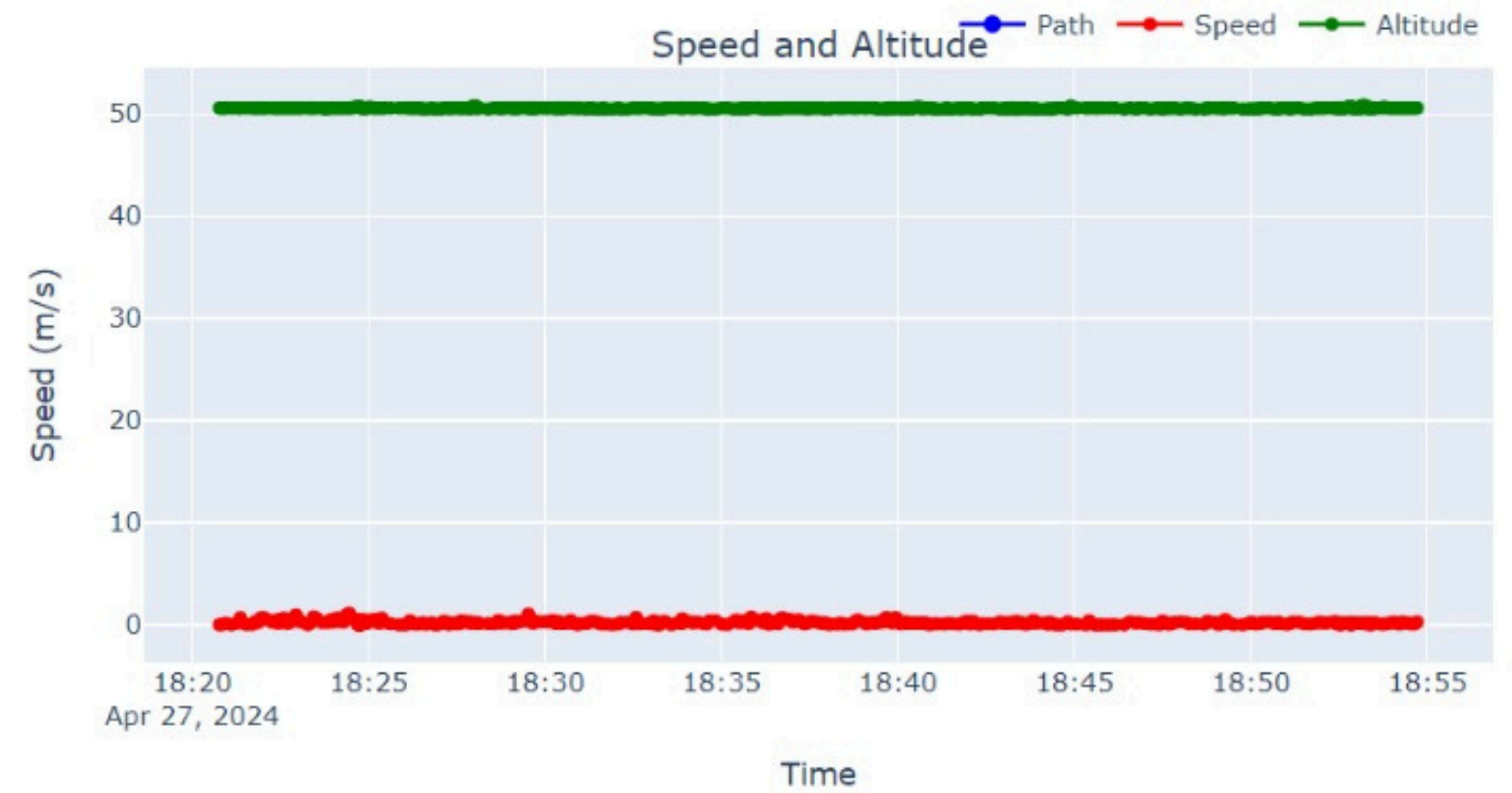


Fig: Visual using Flask

Result

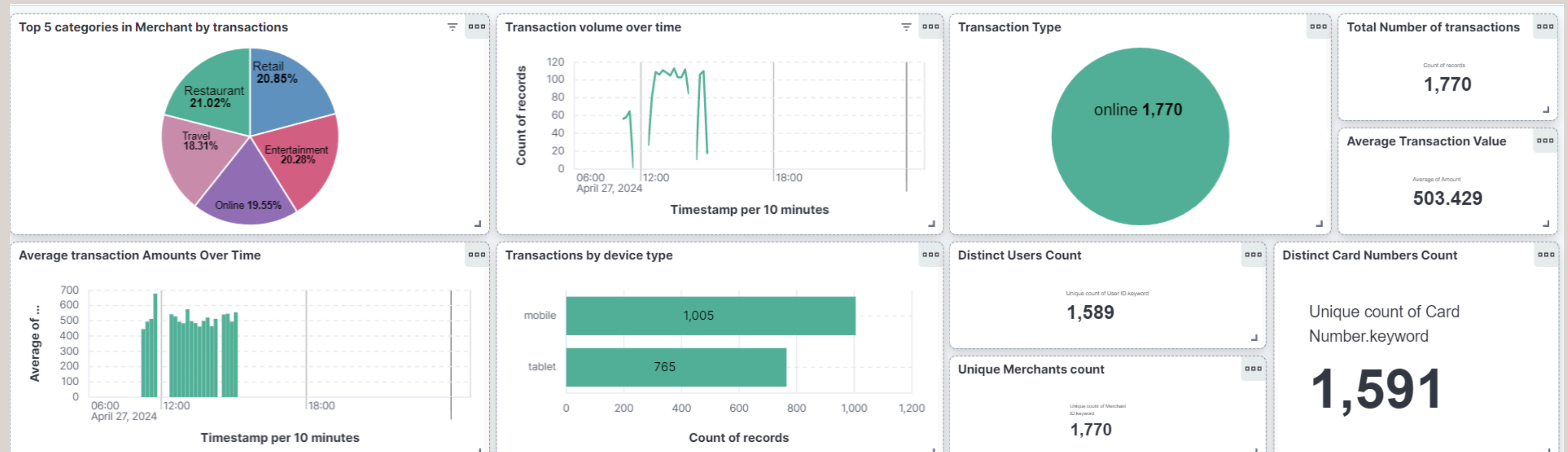
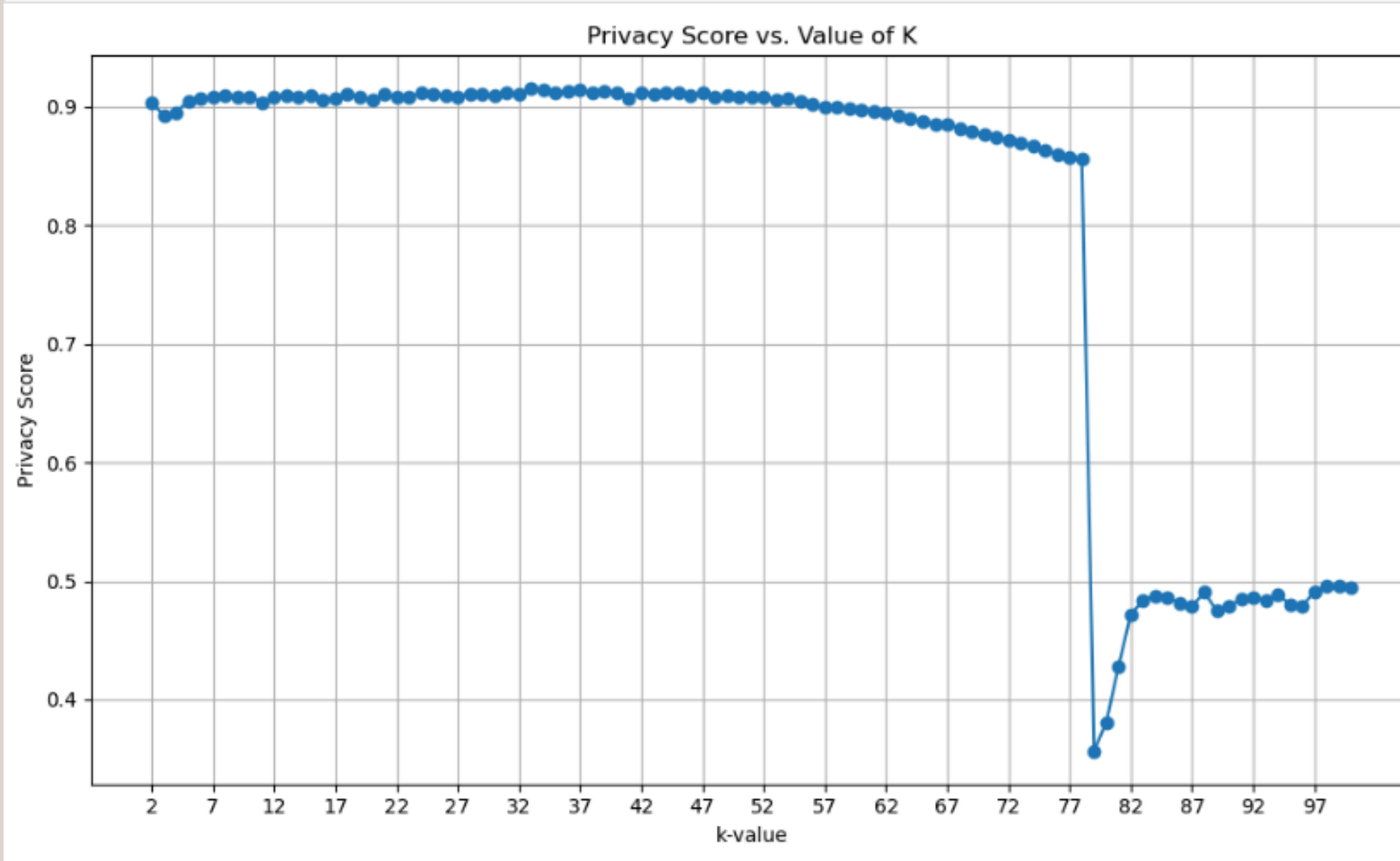


Fig: Kibana Dashboard for Workflow 2

Result



k-value	Privacy Score
2	0.90
7	0.91
12	0.91
17	0.91
22	0.91
27	0.91
32	0.91
37	0.91
42	0.91
47	0.91
52	0.91
57	0.90
62	0.89
67	0.88
72	0.87
77	0.86
78	0.35
82	0.47
87	0.48
92	0.48
97	0.49

Fig: k-value Vs Privacy and cluster from LSH

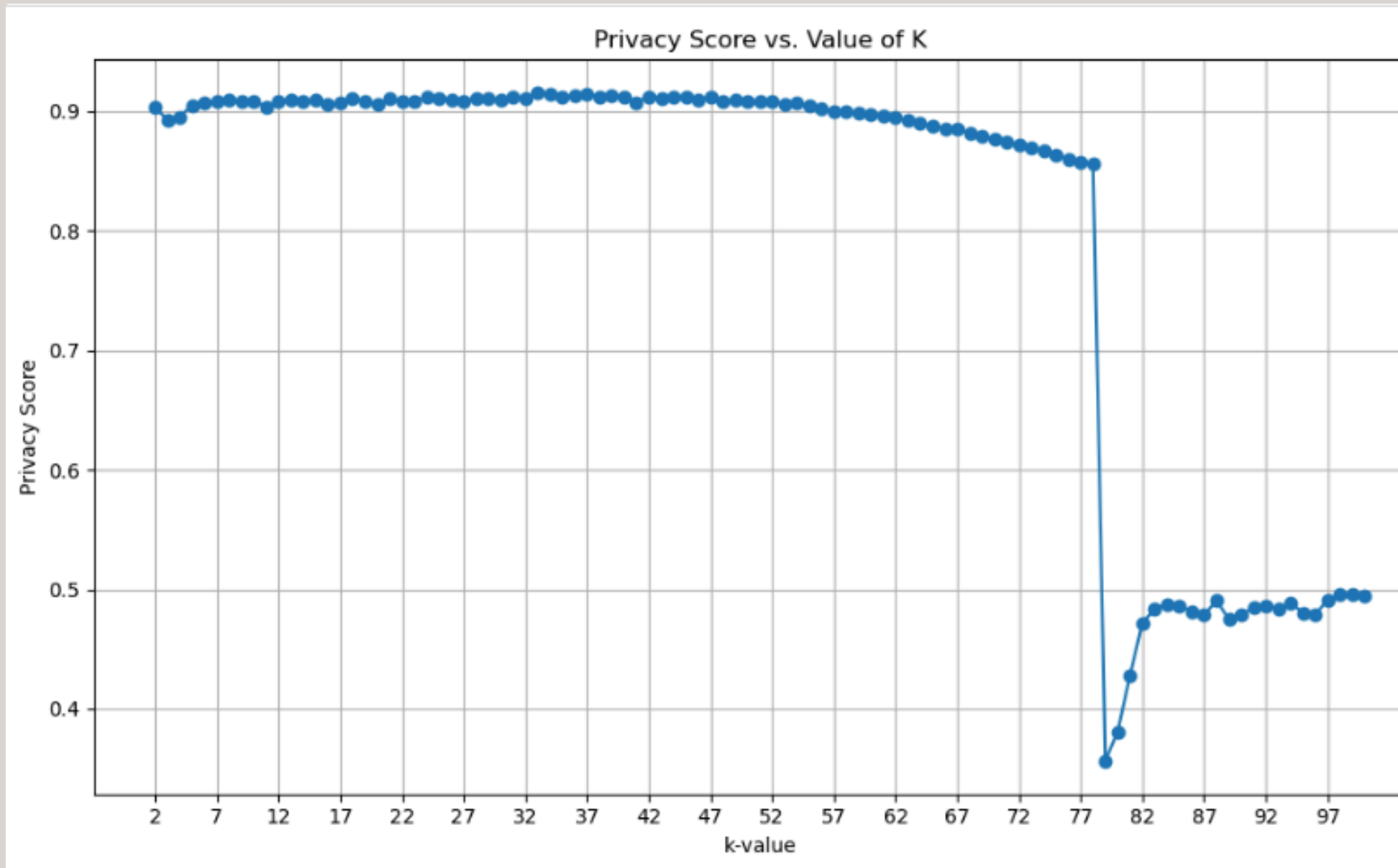
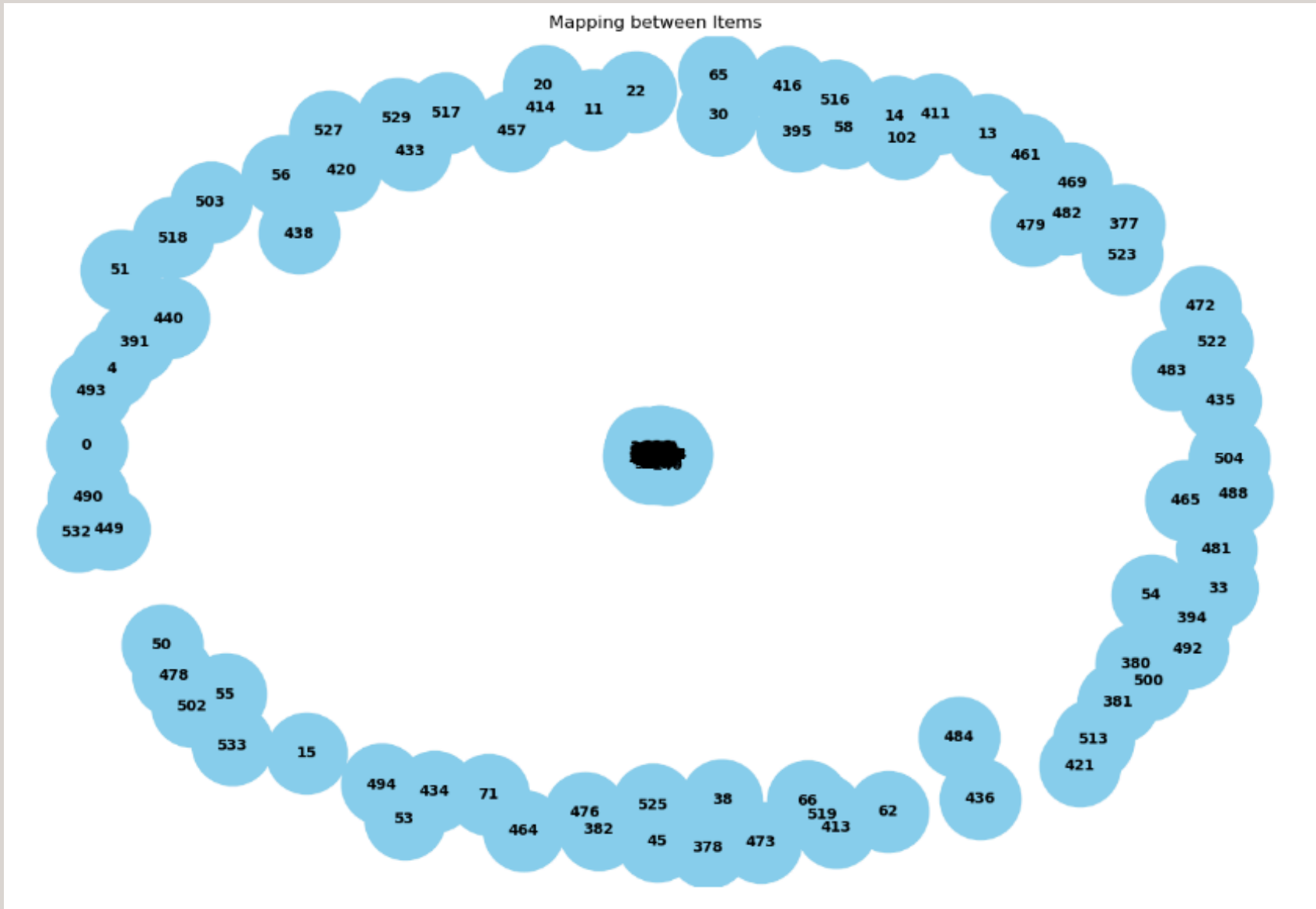
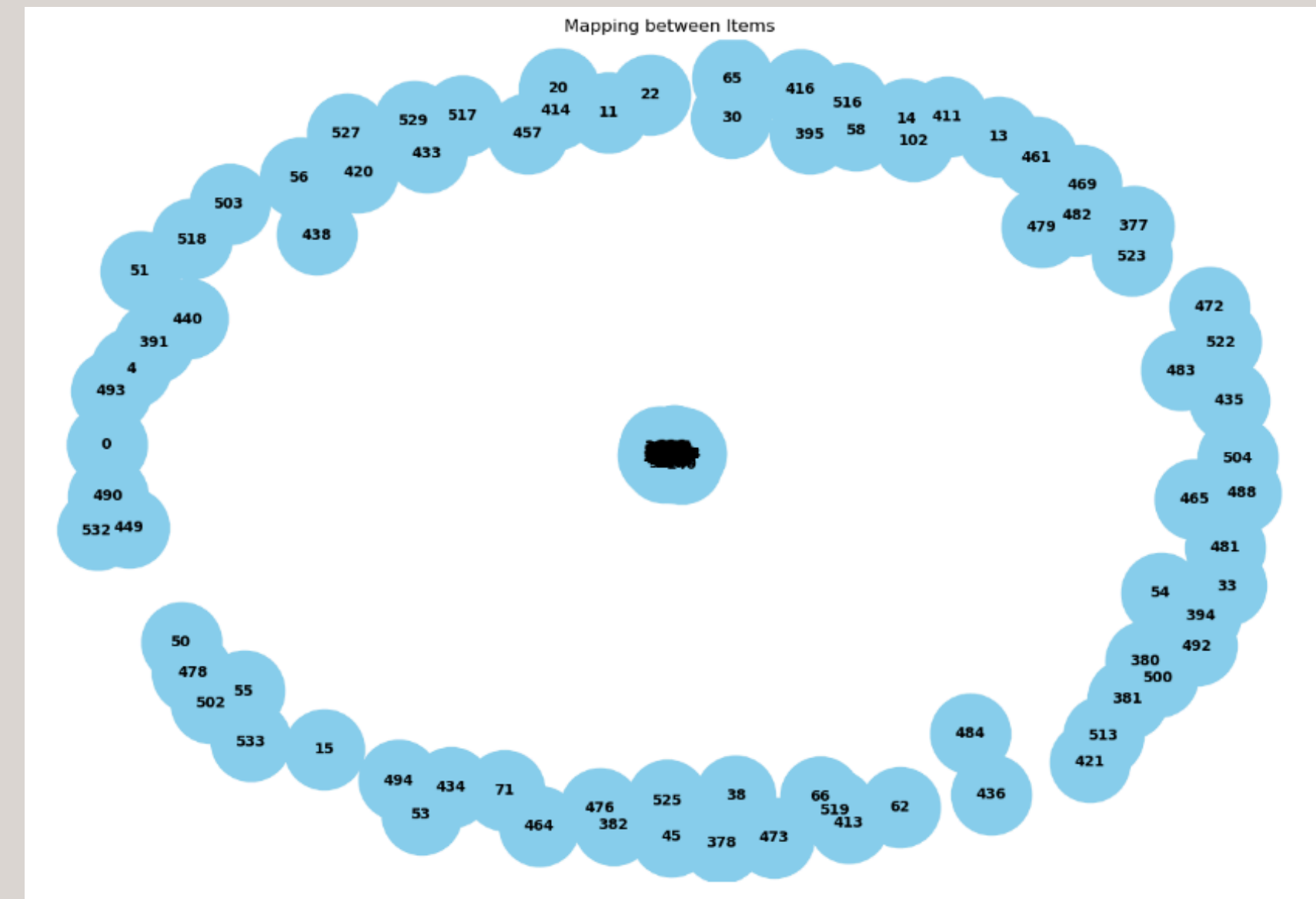


Fig: k-value Vs Privacy and cluster from LSH



Technical Difficulties

- Scalability and fault tolerance: Ensuring infrastructure can handle large volumes of streaming data while maintaining high availability.
- Data quality and consistency: Addressing issues like duplication, out-of-order arrival, and data skew to ensure accurate analytics results.
- Data privacy and security: Implementing encryption, access control, and anonymization techniques to protect sensitive financial information.
- Interoperability: Integrating disparate data sources and systems within the banking environment to achieve seamless data exchange.

Key Learnings

- AWS Integration: Leveraging AWS IoT and Timestream ensures scalable and reliable infrastructure for real-time analytics.
- Synthesizing Data: Introducing fake data alongside original sensor data enriches datasets for more representative analysis.
- Real-time Streaming: Kafka facilitates seamless communication and data flow for efficient processing and analysis.
- Efficient Data Handling: Bloom filters optimize large data volumes for fast queries and enhanced analysis.
- Reservoir Sampling: Ensures creation of representative data subsets crucial for accurate analysis and Bloom filter implementation.
- Data Anonymization: Protects user privacy while preserving data integrity, crucial for regulatory compliance.
- Locality-Sensitive Hashing (LSH): Enables efficient data clustering for fraud detection and anomaly identification.
- Web Application Development: Flask along with Dash empowers development of web applications and APIs for user interaction.
- Analytics with Elasticsearch: Elasticsearch provides robust analytics for diverse analyses and data visualization.

Conclusion

- The implementation of a real-time data analytics pipeline in the banking sector marks a significant advancement in meeting digital banking demands, leveraging robust technologies and integrations for efficient processing and analysis.
- Leveraging Docker, Kafka, and Apache Spark alongside innovative integrations like IOT and Timestream enables streamlined data processing.
- Deployment of Flask, Elastic search, and Grafana enhances pipeline capabilities, providing prompt visuals and alerts for critical scenarios
- Real-time data analytics offer invaluable insights into customer behavior, fraud detection, and personalized banking experiences, enhancing satisfaction and loyalty
- This project represents a strategic move towards a data-driven approach, positioning itself to transform the industry landscape and foster innovation and competitiveness in the ever-evolving market.



**Thank
You**