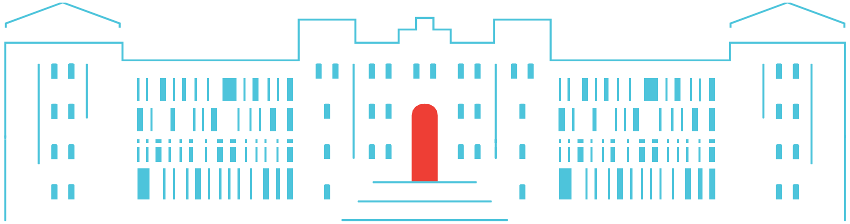
institute for data engineering

Big Data Final Presentation – Group A9

Team Members:
Nikhil Phadke
Viveksingh Jadon
Riya Joshi
Radhika Ray

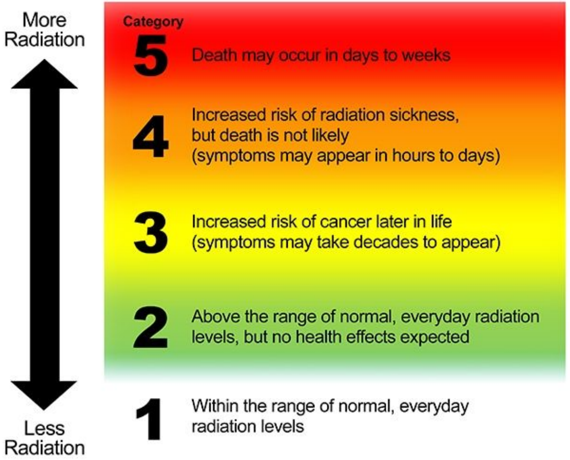
TUHH
Hamburg
University of
Technology



NEED FOR RADIATION TRACKING

TUHH

- Public Health Monitoring
- Early detection of radiation leaks or spills.
- Enables response to radiation emergencies or incidents.



CHALLENGES FACED

- Reliable transmission of data in real-time
- Storage of large data volumes
- Timely reporting and documentation
- Scaling the monitoring system to cover large areas or multiple sites without losing efficiency

INTRODUCTION

The rise of Internet of Things (IoT)

- Increases volume and speed of data.
- Enables constant monitoring of human activities, locations, health states, and communication patterns.
- Example: Sensor networks. (These generate continuous data streams from diverse locations).

However,

- Large data volume demands extensive processing.
- High velocity requires real-time processing to prevent data from being outdated.

PROJECT OVERVIEW

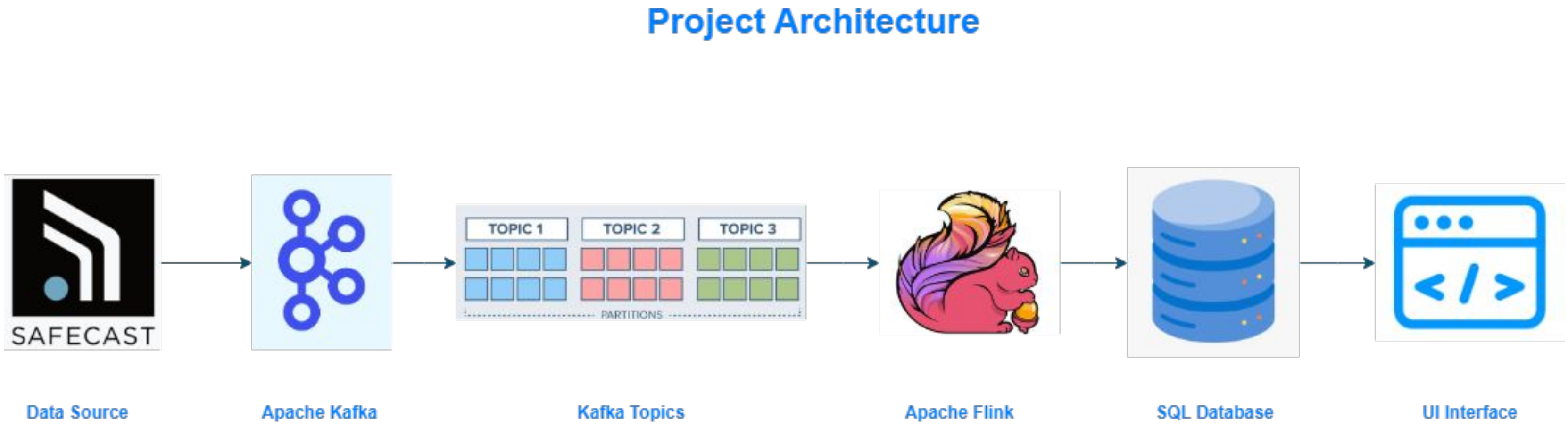
1. Implement and utilize a stream processing framework to handle large volumes of sensor data.

2. Use the latest version of the Safecast Radiation Measurements and populate a world map using the data from it.

3. Provide various operations on this data:

- Summarize minimum, maximum, average and total radiation values.
- Generate alerts when radiation exceeds a user-defined threshold.
- Use different colors to indicate safe and dangerous radiation levels.

ARCHITECTURE



KEY FEATURES

1. Real time overview of radiation levels all over the world.
2. Colour coded indication of safe and dangerous radiation levels
3. Continent-based filtering
4. Overview of minimum, maximum, average and total radiation
5. Storing the data points having dangerous radiation levels

SETUP – DATA SOURCE

Safecast Radiation Measurements Dataset

1. The dataset contains millions of radiation readings, summing up to approximately 25.8 GB of uncompressed data.
2. Radiation data is recorded in counts per minute (CPM), indicating the number of radioactive particles detected per minute by sensors.
3. Measurements include location coordinates (latitude and longitude) and specific times, ensuring detailed spatial and temporal information.
4. The dataset naturally includes noise and potential inaccuracies, requiring careful data processing to filter out erroneous readings.

SETUP – KAFKA PRODUCER

1. Retrieve radiation data from the Safecast dataset.
2. Use the “Geopy” library within the Kafka producer to add continent information to each data point based on the latitude and longitude.
3. Extract and prepare the relevant data fields: captured time, latitude, longitude, radiation value, and the newly added continent information.
4. Create and send messages to a Kafka topic, each containing the enriched data point.

SETUP – KAFKA TOPIC

1. For real-time processing topics
2. Enable producers to publish data and consumers to read it
3. Support immediate data handling.
4. Ideal for real-time analytics and monitoring applications.
5. Data divided into different Kafka topics based on geographic locations(continents)
6. Improves data organization and retrieval efficiency.

SETUP – APACHE FLINK

Utilized for subsequent processing of the data stream generated from Kafka:

1. Initially, the data streams are read from Kafka and directed to the processing job.

2. Flink Job:

   - Within this job, data from Kafka topics are read.
   - Various operations are performed on the incoming data.
   - After processing, the data streams are forwarded to a database sink.

SETUP – SQL DATABASE

TUHH

1.Processed data received from Apache Flink is stored

2. Set up separate tables for each continent along with a consolidated table containing all data points, regardless of location.

3. In total, there are 5 tables

- 3 for individual continents
- 1 aggregate table
- 1 table containing average and total radiation for each continent

4. These tables used to populate the UI map

SETUP – UI

TUHH

1. Data queried from SQL tables using python and mysql connector
2. Javascript used for simulating real-time plotting
3. HTML-CSS used for designing the interface
4. Flask to render the web application

DEMO

Thank You Very Much!

TUHH
Hamburg University of Technology