

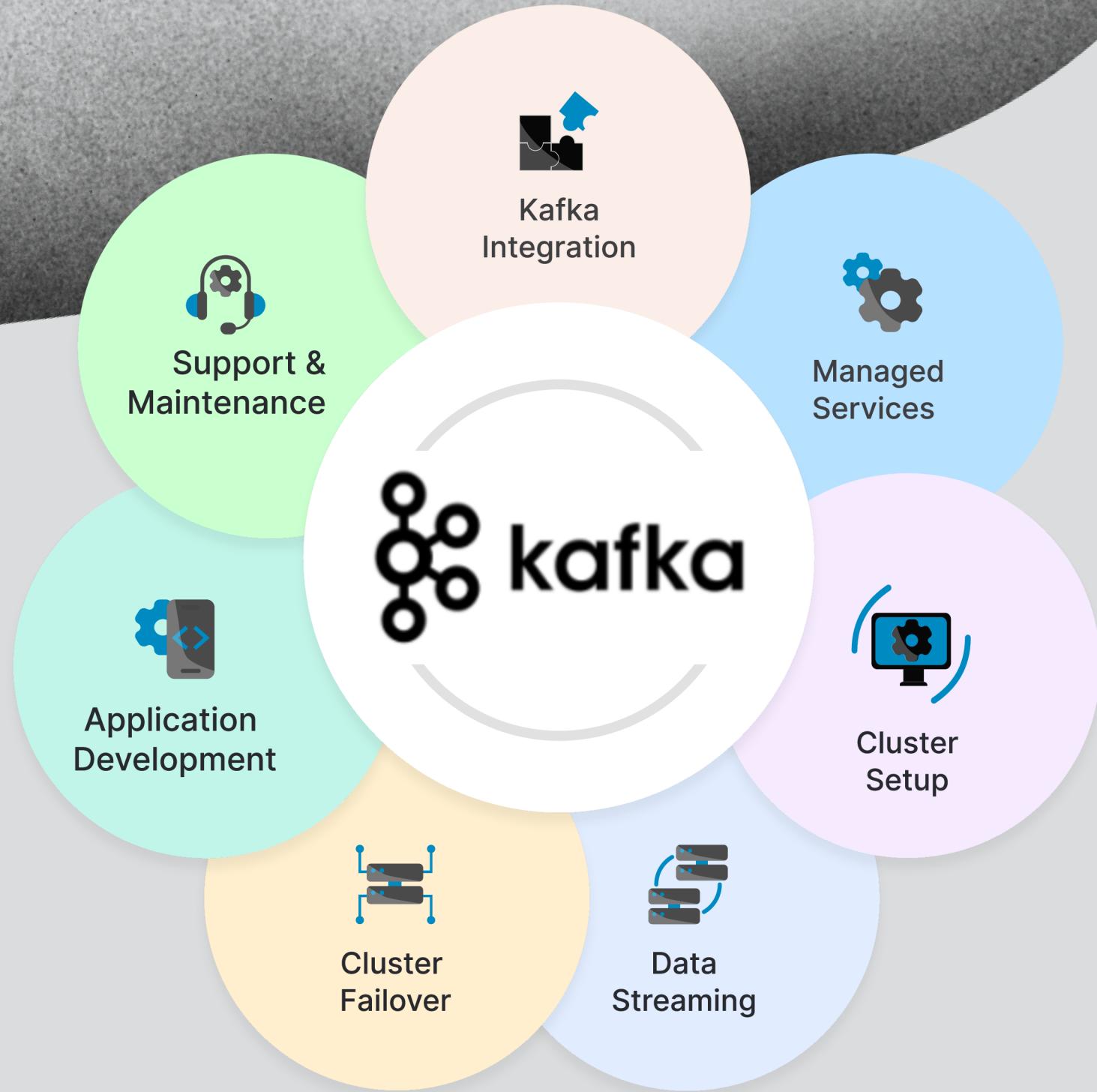


F A U S T

REAL-TIME MONITORING OF ENVIRONMENTAL CONDITIONS

Using Apache Kafka, Faust & Pandas

Author: Hognogi Ana-Maria Cristina



APACHE KAFKA IS A DISTRIBUTED SYSTEM USED TO SEND, STORE, AND PROCESS REAL-TIME DATA STREAMS BETWEEN DIFFERENT SYSTEMS.

WHAT IS APACHE KAFKA?

EXAMPLES:

- In banking: handles live transactions to detect fraud instantly.
- In e-commerce: powers product recommendation engines based on your live activity.
- In transportation: streams GPS data to track delivery vehicles in real time.

WHAT IS FAUST?

Faust is a Python library for real-time stream processing, built to work with Kafka. It allows you to write stream logic using Python in a simple, readable way.

01

A smart home system can use Faust to detect high CO₂ levels and trigger ventilation.

03

In offices, it can detect overcrowded or poorly lit rooms and adjust lighting or air quality automatically.

02

In factories, Faust can monitor machines and alert when noise or temperature exceed safe thresholds.

WHAT IS PANDAS?

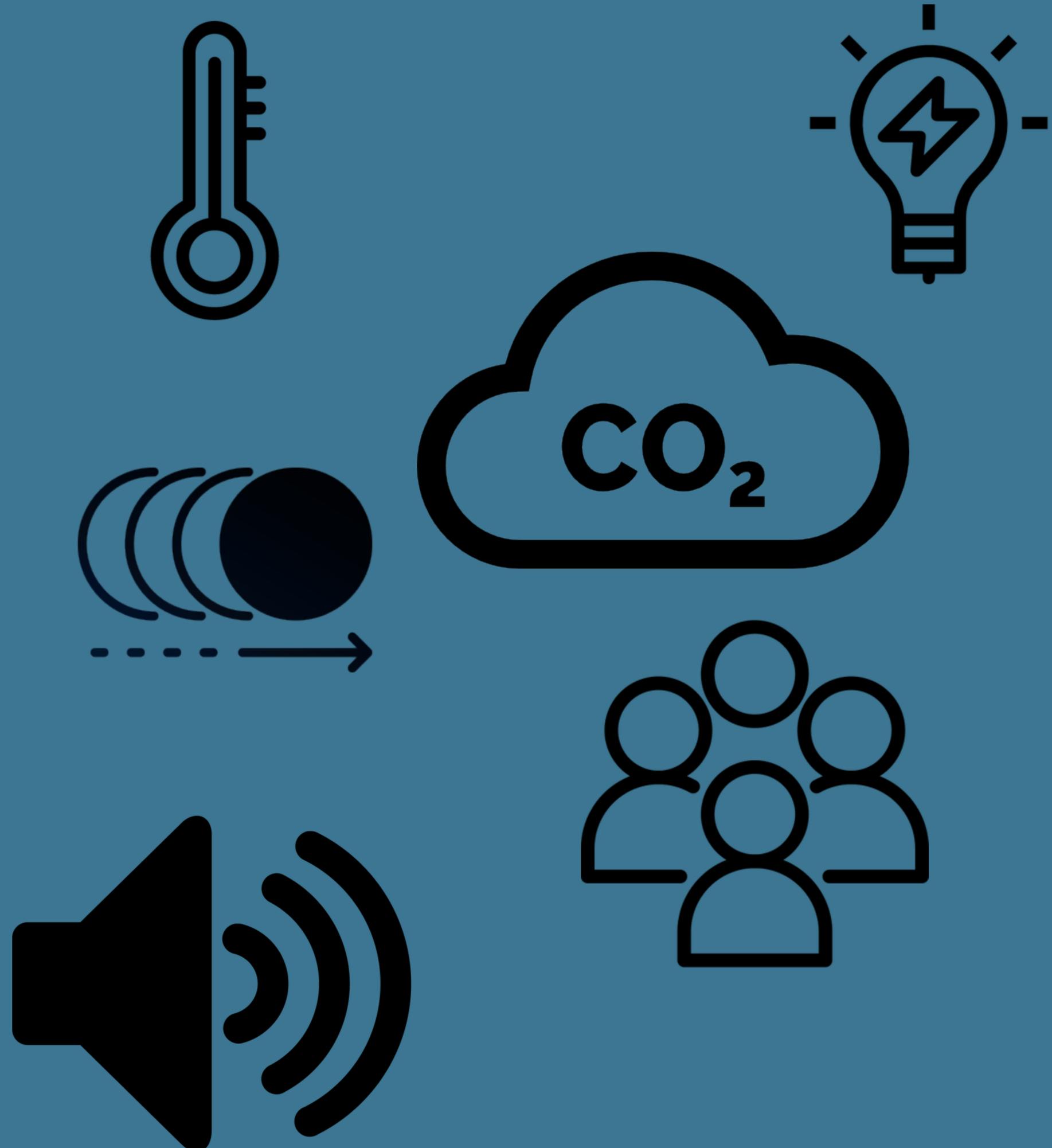
Pandas is a fast, flexible, and easy-to-use open-source data analysis and manipulation library for Python. It provides data structures like DataFrame and Series, making it ideal for cleaning, transforming, and analyzing structured data. Widely used in data science, finance, and machine learning.

Pandas as Python Library



WHY MONITOR INDOOR ENVIRONMENTS IN REAL TIME?

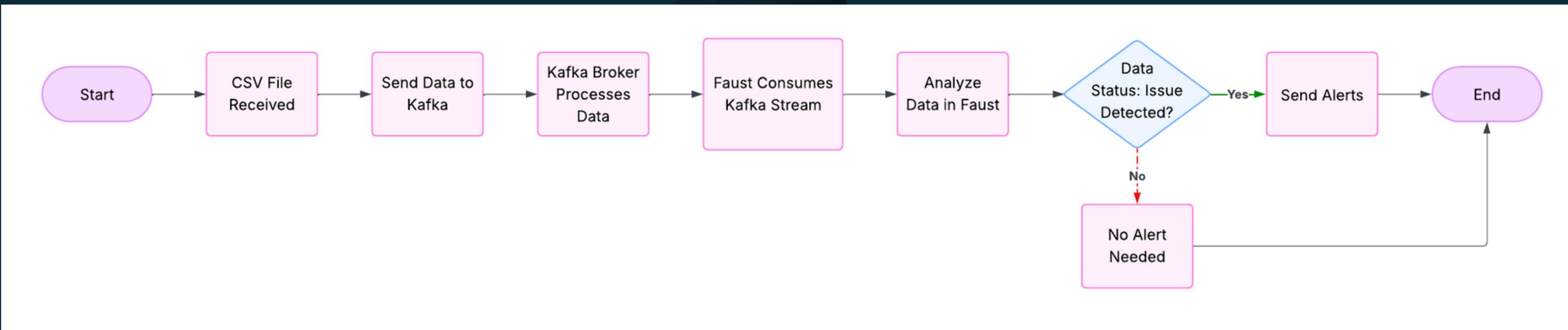
Monitoring environmental parameters like temperature, light, or CO₂ is critical in smart buildings. This project provides a technical solution to do that efficiently and automatically.



OBJECTIVES AND FUNCTIONAL SCOPE

- Process 6 environmental parameters
- Compute averages and trigger alerts
- Use real data for realistic simulation
- Ensure scalability and modularity

The application handles six data streams. It interprets data, not just collects it. This means it can react to unhealthy or uncomfortable conditions automatically.



SIMULATING REAL SENSOR DATA

The architecture is composed of a producer, Kafka as a streaming layer, and Faust which processes the data in Python. Each row of the CSV (the dataset) mimics a sensor snapshot. The producer extracts the data and sends it to Kafka, room by room, in a loop.

1	Date	Time	S1_Temp	S2_Temp	S3_Temp	S4_Temp	S1_Light	S2_Light	S3_Light	S4_Light	S1_Sound	S2_Sound	S3_Sound	S4_Sound	S5_CO2	S5_CO2_Slope	S6_PIR	S7_PIR	Room_Occupancy_Count
2	12/22/17	10:49:41	24.94	24.75	24.56	25.38	121	34	53	40	0.08	0.19	0.06	0.06	875	0.76923077	0	0	4
3	12/22/17	10:50:12	24.94	24.75	24.56	25.44	121	33	53	40	0.93	0.05	0.06	0.06	875	0.64615385	0	0	4
4	12/22/17	10:50:42	25	24.75	24.5	25.44	121	34	53	40	0.43	0.11	0.08	0.06	880	0.51923077	0	0	4
5	12/22/17	10:51:13	25	24.75	24.56	25.44	121	34	53	40	0.41	0.1	0.1	0.09	880	0.38846154	0	0	4
6	12/22/17	10:51:44	25	24.75	24.56	25.44	121	34	54	40	0.18	0.06	0.06	0.06	890	0.25384615	0	0	4
7	12/22/17	10:52:14	25	24.81	24.56	25.44	121	34	54	40	0.13	0.06	0.06	0.07	900	0.16538462	0	0	4
8	12/22/17	10:52:45	25	24.75	24.56	25.44	120	34	54	40	1.39	0.32	0.43	0.06	905	0.07692308	1	0	4
9	12/22/17	10:53:15	25	24.81	24.56	25.44	121	34	54	41	0.09	0.06	0.09	0.05	910	-0.0115385	0	0	4
10	12/22/17	10:53:46	25	24.81	24.56	25.5	122	35	56	43	0.09	0.05	0.06	0.13	910	-0.1	0	0	4
11	12/22/17	10:54:17	25	24.81	24.56	25.5	101	34	57	43	3.84	0.64	0.48	0.39	910	-0.1884615	1	1	4
12	12/22/17	10:54:47	25.06	24.81	24.56	25.44	122	35	57	43	2.2	0.31	0.33	0.21	910	-0.2769231	1	1	1
13	12/22/17	10:55:18	25.06	24.81	24.56	25.5	123	35	57	44	0.42	0.13	0.14	0.09	910	-0.3653846	1	0	1
14	12/22/17	10:55:49	25.06	24.88	24.63	25.5	123	35	57	43	0.21	0.15	0.07	0.06	910	-0.4538462	1	0	1
15	12/22/17	10:56:19	25.06	24.81	24.63	25.56	123	35	57	44	1.66	0.21	0.12	0.09	910	-0.5423077	1	0	1
16	12/22/17	10:56:50	25.06	24.88	24.63	25.56	123	35	58	44	0.57	0.17	0.21	0.13	910	-0.6307692	1	0	1
17	12/22/17	10:57:21	25.06	24.88	24.63	25.56	123	35	58	44	0.14	0.17	0.15	0.06	915	-0.7192308	1	0	1
18	12/22/17	10:57:51	25.06	24.88	24.63	25.56	122	35	58	44	0.24	0.05	0.1	0.05	910	-0.7115385	0	0	1
19	12/22/17	10:58:22	25.06	24.81	24.63	25.56	122	35	57	44	0.25	0.07	0.07	0.05	915	-0.6538462	1	0	1

Kafka Topics

Topic Name
temperature-readings
co2-readings
motion-readings
light-readings
sound-readings
occupancy-readings

KAFKA TOPICS FOR EACH DATA STREAM

Each environmental parameter has its own Kafka topic. This ensures decoupled and scalable data flow.

In Faust, each agent corresponds to a type of data. They run concurrently and respond instantly to incoming messages.

Average temperature in Living Room: 21.35°C

ALERT: High CO2 level in Bedroom: 1240 ppm

ALERT: Cold room in Office - 16.80°C

Low light level in Kitchen: 42 lux

ALERT: High noise level in Living Room: 1.23 dB

Overcrowded room in Bedroom: 5 people

PROCESSING RULES AND ALERT CONDITIONS

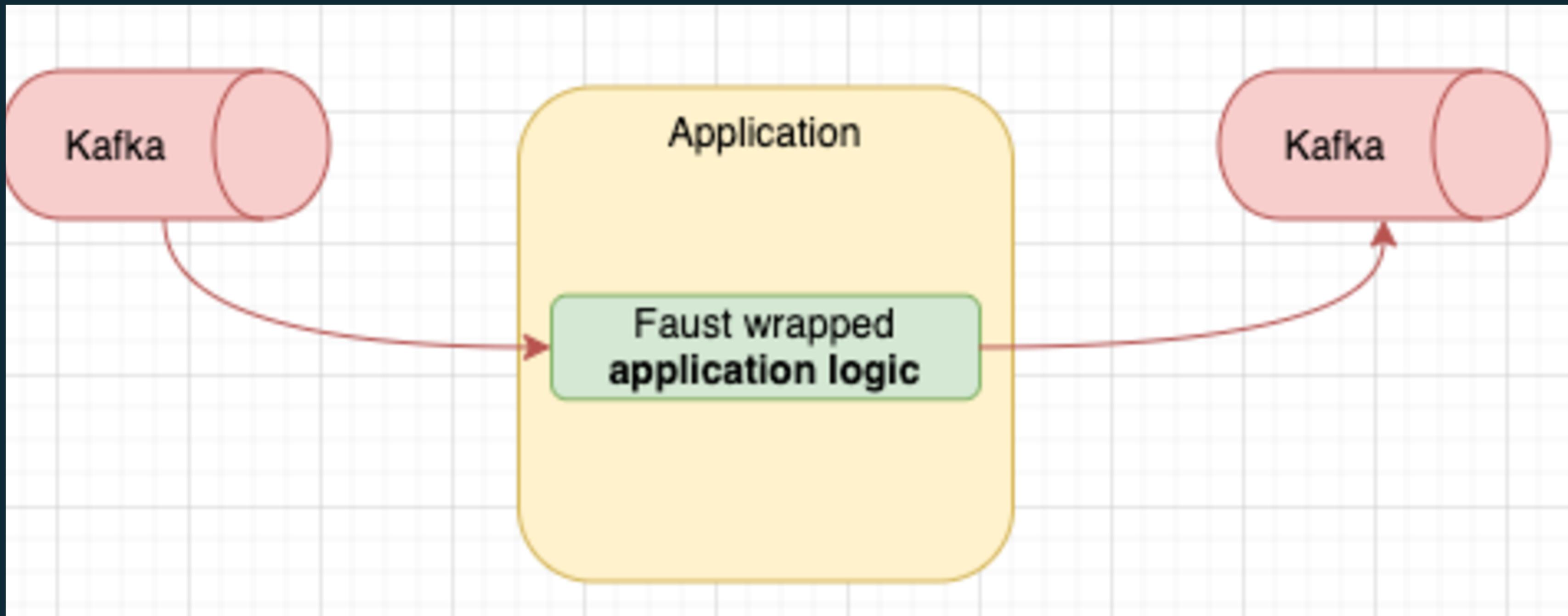
- Average temperature per room
- CO₂ > 1000 ppm alert
- Temperature < 18°C cold room
- Light < 50 lux low lighting
- Sound > 1.0 dB noise alert
- Occupancy > 3 overcrowding

The background features a dark teal gradient. Overlaid on it are several abstract shapes: a large circle in the upper left containing a black line-art globe with latitude and longitude lines; a large, irregularly shaped circle in the center; and a dark, rounded shape in the lower left. A diagonal band of lighter teal runs from the bottom left towards the center.

DEMO

CONCLUSIONS

This project proves how stream processing can help monitor and react to environmental conditions effectively. It's a foundation for future smart systems.



BIBLIOGRAPHY

Apache Kafka Documentation

Faust Stream Processing for Python

Kafka Python Client – kafka-python

Python Virtual Environments (venv).

Pandas Documentation

THANK YOU!

