UNIT-2 PART-1

AUDIT , BALANCE AND CONTROL LAYER

- This layer is the engine that ensures that each processing request is completed by the ecosystem as planned.

- The audit, balance, and control layer is the single area in which you can observe what is currently running within your data scientist environment.

AUDIT

- A systematic and independent examination of the ecosystem.
- Audit sublayer keeps track of the processes that are running at any specific point within the environment.
- Used by data scientists and engineers to plan the future improvements to the processing.

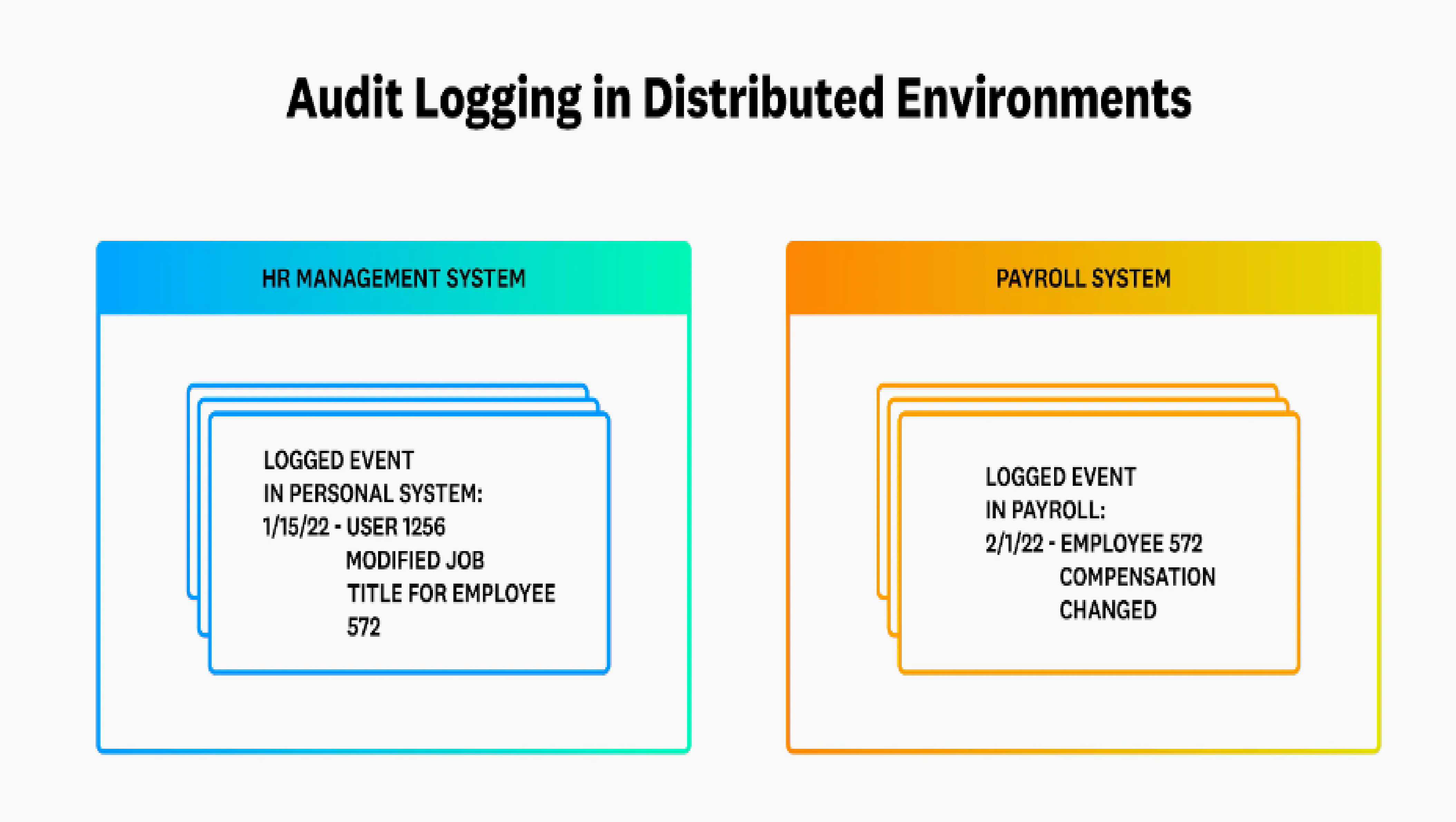
Good Indicators for audit purposes

- Built-In logging
- **Layers of Logging**
- Debug watcher
- Information Watcher
- Warning watcher
- Error Watcher
- Fatal Watcher

What Is Audit Logging?

- Audit logging is the process of documenting activity within the software systems used across your organization.

- Audit logs record the occurrence of an event, the time at which it occurred, the responsible user or service, and the impacted entity. All of the devices in your network, your cloud services, and your applications emit logs that may be used for auditing purposes.



Audit Logging in Distributed Environments

HR MANAGEMENT SYSTEM

LOGGED EVENT
IN PERSONAL SYSTEM:
1/15/22 - USER 1256
MODIFIED JOB
TITLE FOR EMPLOYEE
572

PAYROLL SYSTEM

LOGGED EVENT
IN PAYROLL:
2/1/22 - EMPLOYEE 572
COMPENSATION
CHANGED

What is Audit Trail?

- A series of audit logs is called an audit trail because it shows a sequential record of all the activity on a specific system. By reviewing audit logs, systems administrators can track user activity, and security teams can investigate breaches and ensure compliance with regulatory requirements.

Audit logs capture the following types of information:

- Event name as identified in the system
- Easy-to-understand description of the event
- Event timestamp
- Actor or service that created, edited, or deleted the event (user ID or API ID)
- Application, device, system, or object that was impacted (IP address, device ID, etc.)
- Source from where the actor or service originated (country, host name, IP address, device ID, etc.)

Audit Logs vs. Regular System Logs

- Regular system logs are designed to help developers troubleshoot errors, audit logs help organizations document a historical record of activity for compliance purposes and other business policy enforcement.

Process Tracking,Data Provenance, data lineage

- There is numerous server-based software that monitors temperature sensors, voltage, fan speeds, and load and clock speeds of a computer system.

- Keep records for every data entity in the data lake, by tracking it through all the transformations in the system. This ensures that you can reproduce the data, if needed, in the future and supplies a detailed history of the data’s source in the system.

- Keep records of every change that happens to the individual data values in the data lake. This enables you to know what the exact value of any data record was in the past.

Balance

- The balance sublayer has the responsibility to make sure that the data science environment is balanced between the available processing capability against the required processing capability or has the ability to upgrade processing capability during periods of extreme processing.

- In such cases the on-demand processing capability of a cloud environment becomes highly desirable.

Control

- The execution of the current active data science processes is controlled by the control sublayer.

- The control sublayer also ensures that when processing experiences an error, it can try a recovery, as per your requirements, or schedule a clean-up utility to undo the error. The cause-and-effect analysis system is the core data source for the distributed control system in the ecosystem.

Yoke Solution

- The yoke system is used to control the processing.

- The yoke system ensures that the distributed tasks are completed, even if it loses contact with the central services. The yoke solution is extremely useful in the Internet of things environment, as you are not always able to communicate directly with the data source.

Contt…

Kafka provides a publish-subscribe solution that can handle all activity-stream data and processing. The Kafka environment enables you to send messages between producers and consumers that enable you to transfer control between different parts of your ecosystem while ensuring a stable process.

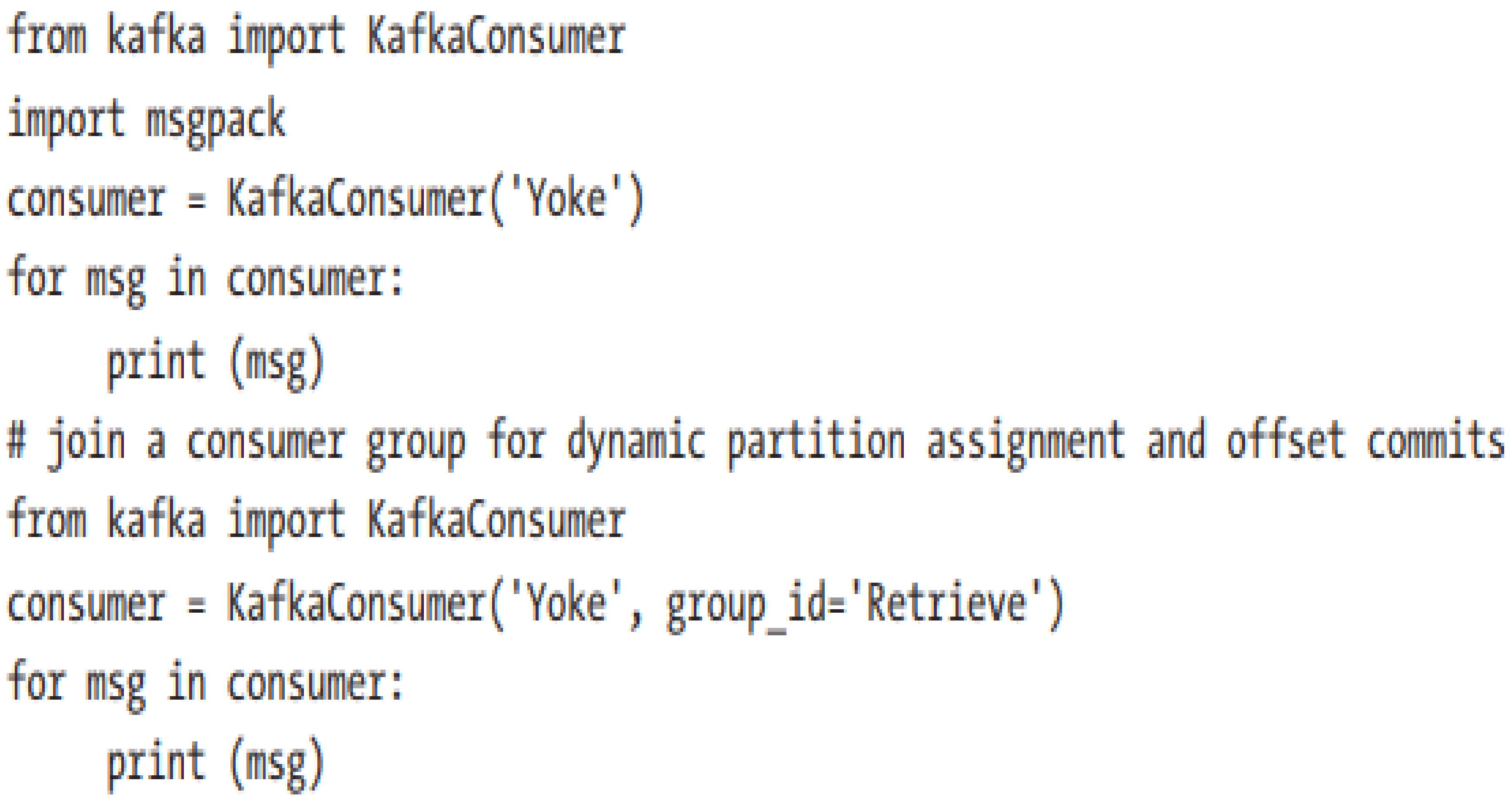
Producer

The producer is the part of the system that generates the requests for data science processing, by creating structures messages for each type of data science process it requires. The producer is the end point of the pipeline that loads messages into Kafka.

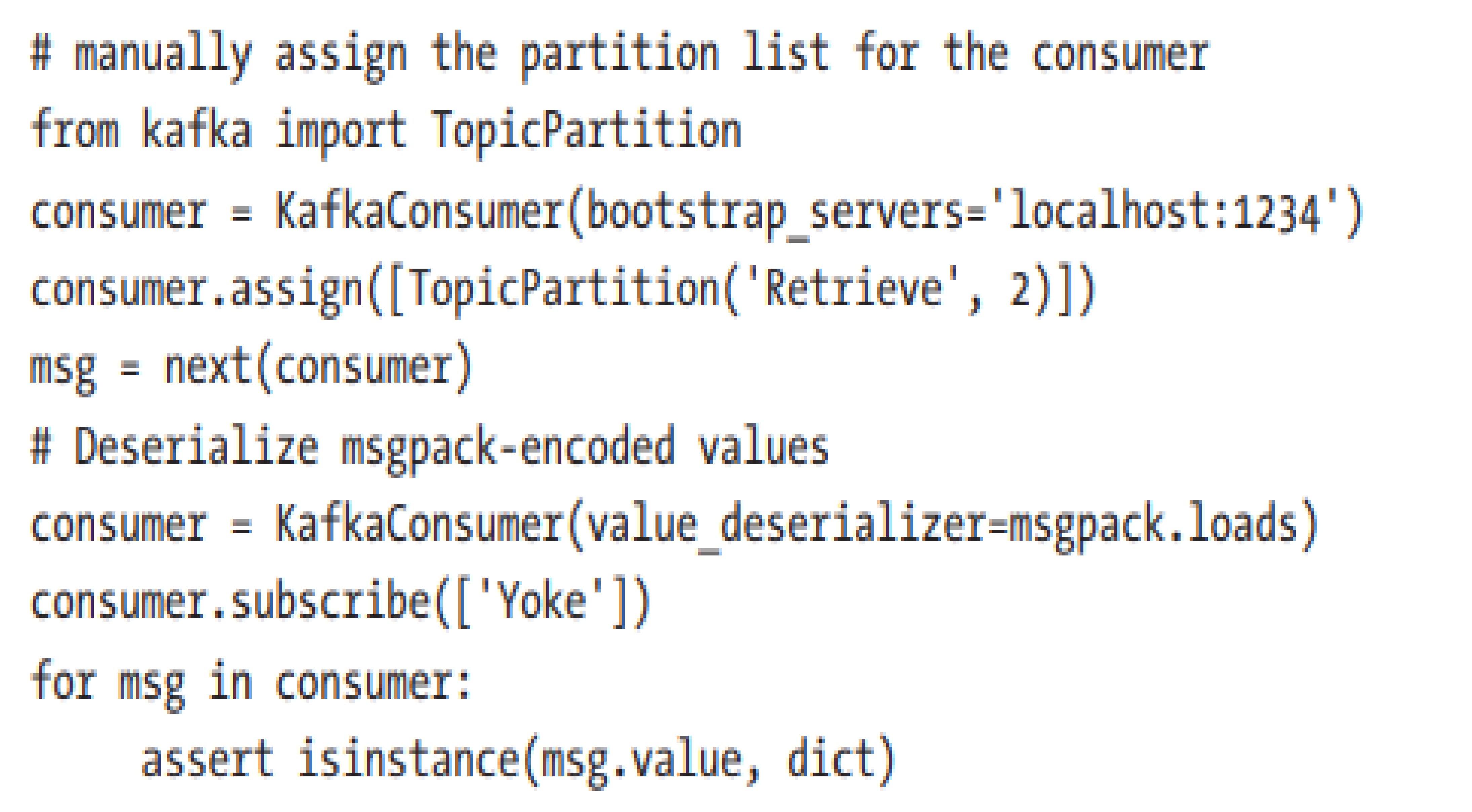
```
from kafka import KafkaProducer
```

Consumer

The consumer is the part of the process that takes in messages and organizes them for processing by the data science tools. The consumer is the end point of the pipeline that offloads the messages from Kafka.



```
from kafka import KafkaConsumer
import msgpack
consumer = KafkaConsumer('Yoke')
for msg in consumer:
     print (msg)
# join a consumer group for dynamic partition assignment and offset commits
from kafka import KafkaConsumer
consumer = KafkaConsumer('Yoke', group_id='Retrieve')
for msg in consumer:
     print (msg)
```



```
# manually assign the partition list for the consumer
from kafka import TopicPartition
consumer = KafkaConsumer(bootstrap_servers='localhost:1234')
consumer.assign([TopicPartition('Retrieve', 2)])
msg = next(consumer)
# Deserialize msgpack-encoded values
consumer = KafkaConsumer(value_deserializer=msgpack.loads)
consumer.subscribe(['Yoke'])
for msg in consumer:
     assert isinstance(msg.value, dict)
```

Directed Acyclic graph scheduling

This solution uses a combination of graph theory and publish-subscribe stream data processing to enable scheduling.

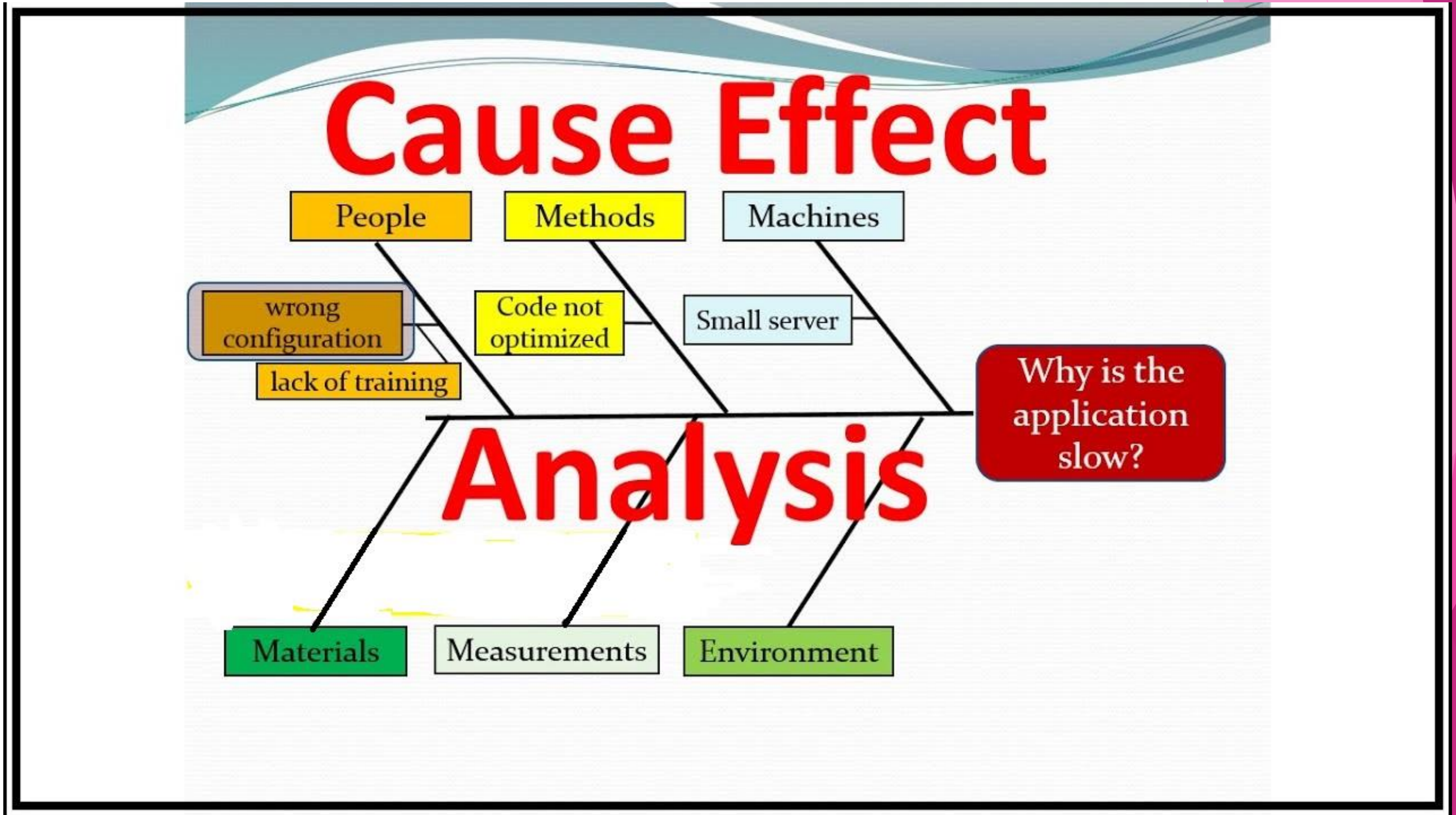
You can use the Python NetworkX library to resolve any conflicts, by simply formulating the graph into a specific point before or after you send or receive messages via Kafka.

That way, you ensure an effective and an efficient processing pipeline.

Cause and Effect Analysis System

- The cause-and-effect analysis system is the part of the ecosystem that collects all the logs, schedules, and other ecosystem-related information and enables data scientists to evaluate the quality of their system

Example


Cause Effect
Analysis
People
Methods
Machines
wrong configuration
lack of training
Code not optimized
Small server
Why is the application slow?
Materials
Measurements
Environment

Functional Layer

- The functional layer of the data science ecosystem is the largest and most essential layer for programming and modeling.

- Any data science project must have processing elements in this layer. The layer performs all the data processing chains for the practical data science.

DATA SCIENCE PROCESS

- Following are the five fundamental data science process steps.
- o Begin process by asking a What if question
- o Attempt to guess at a probably potential pattern
- o Create a hypothesis by putting together observations
- o Verify the hypothesis using real-world evidence
- o Promptly and regularly collaborate with subject matter experts and customers as and when you gain insights .

Begin process by asking a What if question

- Decide what you want to know, even if it is only the subset of the data lake you want to use for your data science, which is a good start.

Take a Guess at a Potential Pattern

- Use your experience or insights to guess a pattern you want to discover, to uncover additional insights from the data you already have.

Gather Observations and Use Them to Produce a Hypothesis

- A hypothesis, it is a proposed explanation, prepared on the basis of limited evidence, as a starting point for further investigation.

Use Real-World Evidence to Verify the Hypothesis

- Now, we verify our hypothesis by comparing it with real-world evidence .

Promptly and regularly collaborate with subject matter experts and customers as and when you gain insights

- Things that are communicated with experts may include technical aspects like workflows or more specifically data formats & data schemas.

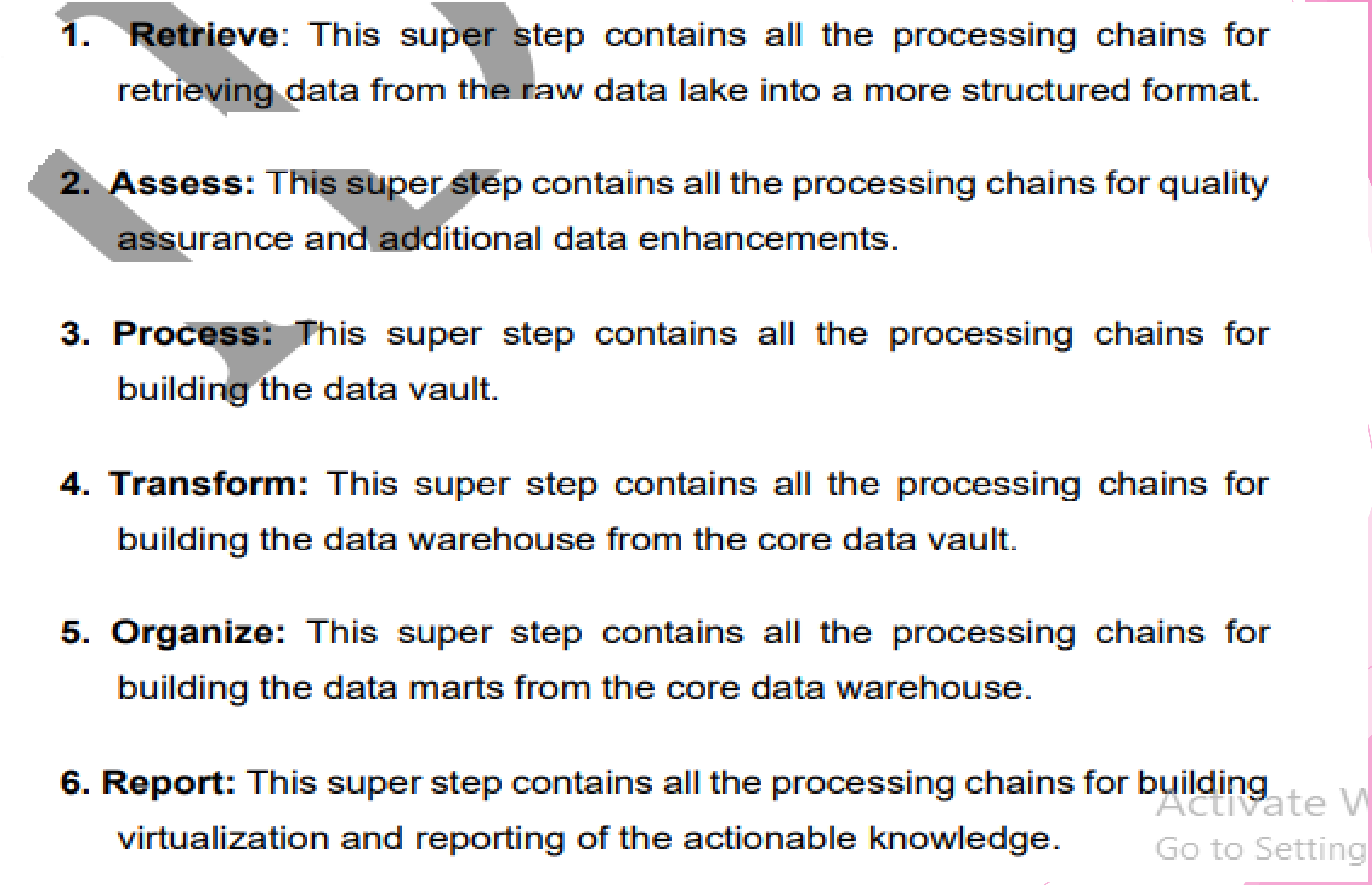
DATA STRUCTURES USED IN FUNCTIONAL LAYER

- Data schemas and data formats
- Data models:
- Processing algorithms
- Provisioning of infrastructure

SUPERSTEPS FOR PROCESSING DATA LAKES

- A superstep consists of a unit of generic programming, which through a global communication component, makes thousands of parallel processing on a mass of data and sends it to a “meeting” called synchronization barrier. At this point, the data are grouped, and passed on to the next superstep chain.

SUPERSTEPS FOR PROCESSING DATA LAKES



1. **Retrieve**: This super step contains all the processing chains for retrieving data from the raw data lake into a more structured format.

2. **Assess:** This super step contains all the processing chains for quality assurance and additional data enhancements.

3. **Process:** This super step contains all the processing chains for building the data vault.

4. **Transform:** This super step contains all the processing chains for building the data warehouse from the core data vault.

5. **Organize:** This super step contains all the processing chains for building the data marts from the core data warehouse.

6. **Report:** This super step contains all the processing chains for building virtualization and reporting of the actionable knowledge.

THANKS