AIF Ingestion Framework - Analysis and Observations

# 1. Overview

The Allianz Ingestion Framework (AIF) is a Spark-based data ingestion pipeline designed to load data from various sources (CSV, TXT, XML, or RDBMS) into a Hadoop-based Hive environment. It supports full and incremental ingestion, handles metadata auditing, and tracks job failures for retry.

# 2. Required Inputs

- input\_feed.csv: Defines table-wise ingestion jobs.  
- wc\_connection.yaml: Environment-specific configs (HDFS, JDBC, etc.)  
- columns.json: Optional schema for each table.

# 3. Supported Flow Types

- fs\_hv: File-based CSV ingestion to Hive stage and lake.  
- txt\_hv: Delimited TXT/CSV ingestion.  
- xml\_hv: XML file processing (land, stage, lake).

# 4. Key Script Functions

- aif\_run.py: Master script that reads configs, sets up Spark, and dispatches ingestion jobs.  
- aif\_db\_stage.py: Handles CSV ingestion using Spark with schema, masking, incremental logic.  
- aif\_txt\_file\_stage.py: Deals with TXT/CSV files and supports metadata tracking.  
- aif\_xml\_land\_stage\_lake.py: Handles XML file staging and movement to Hive.  
- aif\_stage\_lake.py: Loads data from stage to lake (partitioning, bucketing, transformation logic).  
- aif\_meta.py: Metadata tracking class used for auditing.

# 5. Audit and Retry Mechanism

- meta\_file: Stores counts, statuses, file size, start/end time for auditing.  
- not\_loaded\_file: Captures failed jobs for rerun using the same framework.

# 6. Sample Execution Command

spark-submit aif\_run.py --env DEV --input\_feed conf/input\_feed.csv   
--conf\_file conf/wc\_connection.yaml --columns conf/table\_schema.json --mode yarn

# 7. Recommendations

- Modularize XML, TXT, and DB handlers for cleaner integration.  
- Enhance schema validation and logging for better debugging.  
- Consider adding S3/GCS support for cloud-native ingestion.  
- Document schema and partition logic in a central data dictionary.