**CDD**

**Source System:**

SAP MDR , which contains structured business data related to materials, customers, and vendors. This data is periodically extracted and moved into (ADLS) for further processing.

**Source to Landing Zone:**

Once the data is extracted from SAP MDR, it is first pushed into the Landing Zone. The landing zone serves as a temporary storage area where raw, unprocessed data is collected before it moves through various processing stages

**how data is ingested (loaded) into Azure Data Lake Storage (ADLS):**

The configuration file helps automate data processing in an Ingestion Framework using Apache Airflow.

**Things to Remember When Updating the Configuration File**

**Mandatory Fields:**

* TriggerName,
* SourceName,
* SourceFilePath,
* FileFormat,
* WriteMode,
* ClusterNodeType,
* ClusterNumOfWorker,
* ClusterVersion,
* Tags,
* ActiveFlag.

**Data Relationships Across Layers:**

Some tables follow a one-to-one mapping, meaning they retain the same structure across all layers with minimal transformation.

Other tables follow a one-to-many relationship, where one record in a source table generates multiple records in downstream layers due to data enrichment or normalization.

Certain tables are dependent on others, requiring complex joins or lookups before they can be processed further.

Some tables are independent, meaning they do not rely on other data sources and can be processed in isolation.

**We have three modules: Material, Customer, and Vendor** – each follow different data flow

**Automating Data Ingestion with Apache Airflow**

* Apache Airflow automates and orchestrates data pipelines.
* The ingestion framework uses DAGs to schedule and manage data processing.
* Each DAG defines dependencies, scheduling logic, and execution steps.
* DAGs are stored and managed in GitHub, allowing version control and automated deployments.

**Cdc techniques:**

**1. Timestamp-Based CDC**

Uses a last\_updated timestamp column to track changes.

Extracts only records that have changed since the last run.

Efficient for sources with reliable timestamp tracking.

**2. Log-Based CDC**

Monitors database transaction logs to detect changes.

Captures inserts, updates, and deletes in real time.

Provides high accuracy but requires database log access.

**3. Snapshot-Based CDC**

Compares the latest dataset with the previous version.

Identifies new, modified, and deleted records.

Useful when timestamp or log-based CDC is unavailable but can be resource-intensive.