# Generic Workflow Design

## Objective

The generic workflow automates data ingestion by dynamically orchestrating metadata retrieval, ingestion, and processing. It efficiently handles multiple data sources using API-driven triggers and integrates with Delta Live Tables (DLT) for seamless data processing.

## Workflow and DLT Orchestration

The workflow is triggered by an API call, which initiates a series of actions within a workflow environment.

A diagram of a diagram

AI-generated content may be incorrect.

## Orchestration Components

### 1. Third party API Calling

**Description:** This serves as the starting point of the process. It contains API logic that points to the main workflow with different parameters and configurations to orchestrate the workflow dynamically based on different data sources.

**Input parameters:**

* **Datasource name** – Data source names such as 'PTS\_JET', 'Boost\_Retail', etc.
* **Raw config path** – Locations where all the raw configuration files are stored.
* **Bronze config path** - Locations where all the bronze configuration files are stored.

**Functionality:**

* Contains API logic to trigger the workflow with specific parameters.
* Dynamically orchestrates the workflow based on different data sources.

### 2. Metadata Fetcher

**Description:** The Metadata fetcher is a component designed to retrieve metadata necessary for the subsequent steps in the workflow.

**Input parameters:**

* **Raw config path –** This input is dynamically received from the API trigger for each data source.
* **Utils path –** This is configured as a task parameter that contains the workspace path.

**Functionality:**

* Fetches all the JSON configuration files present at a location based on different data sources.
* Compiles a list of all JSON configuration file locations for the Ingest notebook, allowing it to ingest tables in parallel based on the defined concurrency limit.

### 3. Ingest Notebook

**Description:** The Ingest notebook is responsible for ingesting the data from the source into the system based on the metadata fetched.

**Input parameters:**

* **utils\_path** - This is configured as a task parameter that contains the workspace path.
* **config\_file -** This input is dynamically received from the Metadata fetcher which is a list of all JSON configuration file locations for each data source
* **job\_run\_id –** This is the run ID of the job, assigned through task parameters.
* **task\_run\_id –** This is the run ID of the current task, set through task parameters.

**Functionality:**

* Extracts data from the source, applies Change Data Capture (CDC) by comparing it with bronze data, performs transformations, and writes the results to the target Delta table.

### 4. Bronze Notebook

**Description:** The Bronze notebook handles two scenarios based on the presence of a DLT pipeline.

**Input parameters:**

* **Datasource name** – This input is dynamically received from the API trigger for each data source.
* **Bronze config path** - This input is dynamically received from the API trigger for each data source.

**Functionality:**

* **Start DLT Pipeline**: If the DLT pipeline is already present (same data source running daily), it starts the DLT pipeline for the respective data source.
* **Create DLT Pipeline**: If the DLT pipeline is not present (new data source), it creates a new DLT pipeline by calling an API for the respective data source and then it starts this created DLT pipeline.