Airflow as a dynamic ETL tool

Hendrik Kleine

Vicente Ruben Del Pino

Who are we

- Hendrik Kleine
- Analytics Lead

- Spend the past 10 years establishing BI teams and services including eBay, Microsoft and IBM. Focused on improving ease of use for end users.

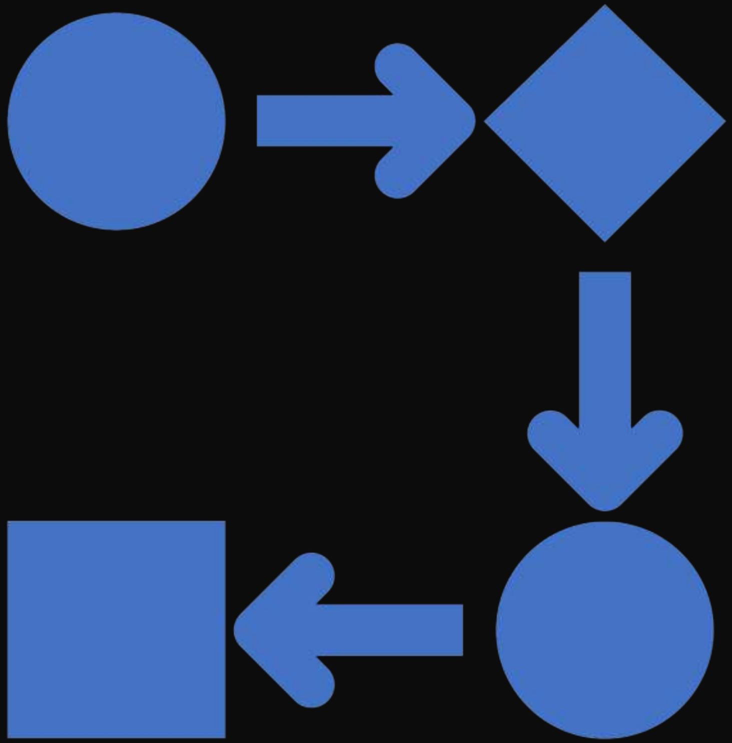
Who are we

- Vicente Ruben Del Pino:
- Data Engineering Lead

- More than a decade of experience working on the architecture, design, coding and implementation of Business Intelligence and Data Warehouse environments at scale.

Content

1. Challenges of legacy platform.
   1. Environment
   2. Skillset
   3. Our central Application
2. Transition from a platform with Alteryx to Airflow.
   1. Requirements
   2. Design of the solution
3. Challenges faced and lessons learned
   1. Achievements
   2. Challenges for next version



The environment

Data Silos:

- Multiple services generating data
- Each service designer choses different storage
- Data Science and Analytics consumption

The environment (II)

Data Sources disconnected:

- Integrate data sources
- Different technologies
- Lack of expertise in ETL processes

The environment (III)

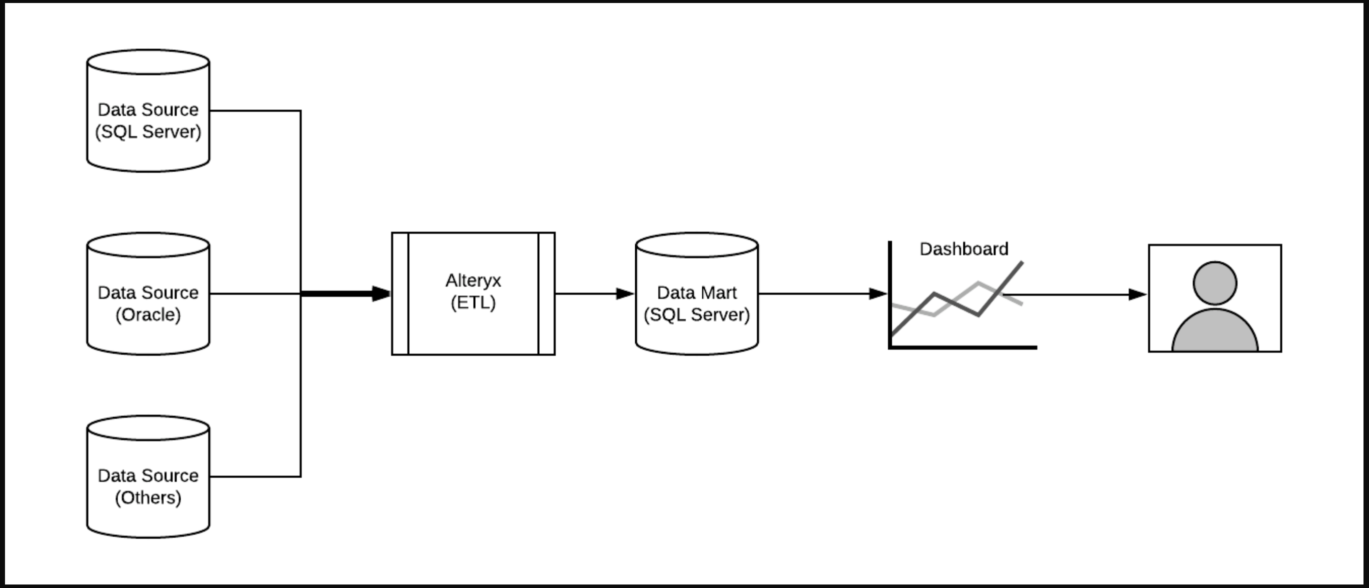
Technology Stack:

- SQL Server as storage for Analytics
- Alteryx as ETL tool
- Tableau as reporting tool

The environment (IV)

Technology Stack:

- SQL Server as storage for Analytics
- Alteryx as ETL tool
- Tableau as reporting tool



Skills set (I)

Three main roles in the area:

Data Engineer:

- Data Ingestion
- Data Processing

Business Intelligence

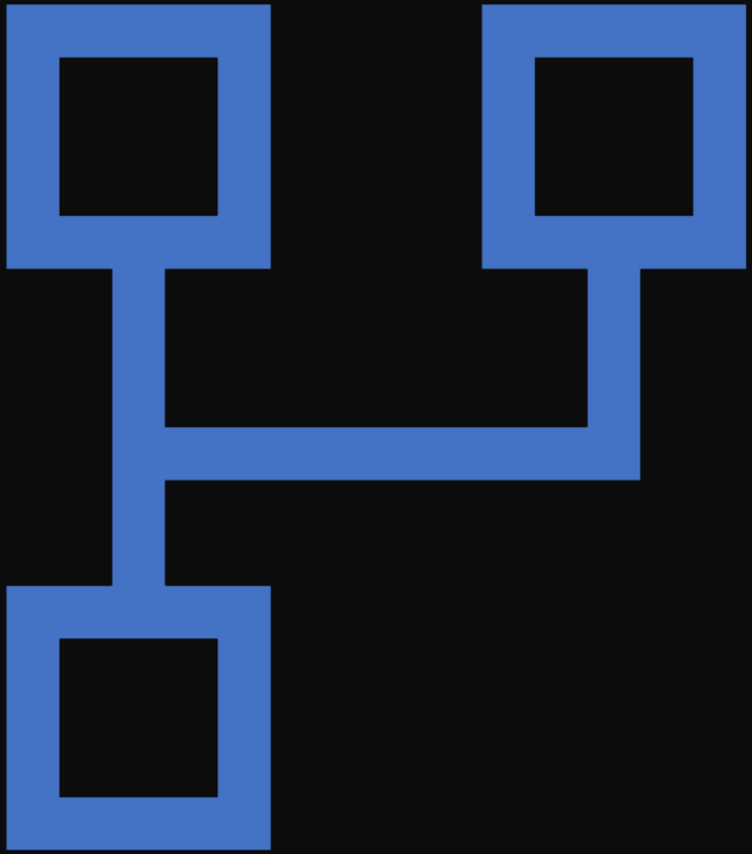
- Data Mart design/development
- Dashboard Creation

Business Analyst

- Requirements gathering

Skills set - Data Engineer (II)

- Experts in
  - Big Data technologies
  - Code programming
  - Data Processing

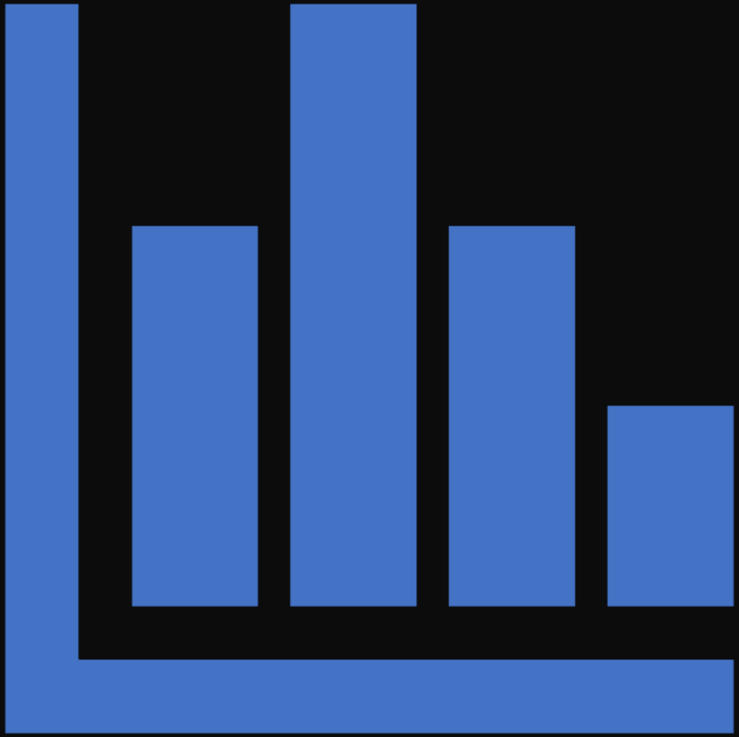


Skills set - Business Intelligence (III)

- Experts in:
  - Building dashboards
  - Creating logic for complex KPIs
  - Designing data marts

Skills set - Business Analyst (IV)

- Experts in:
  - Business Knowledge
  - Requirements Gathering
  - Bridge Gap between Engineers and BI Developers



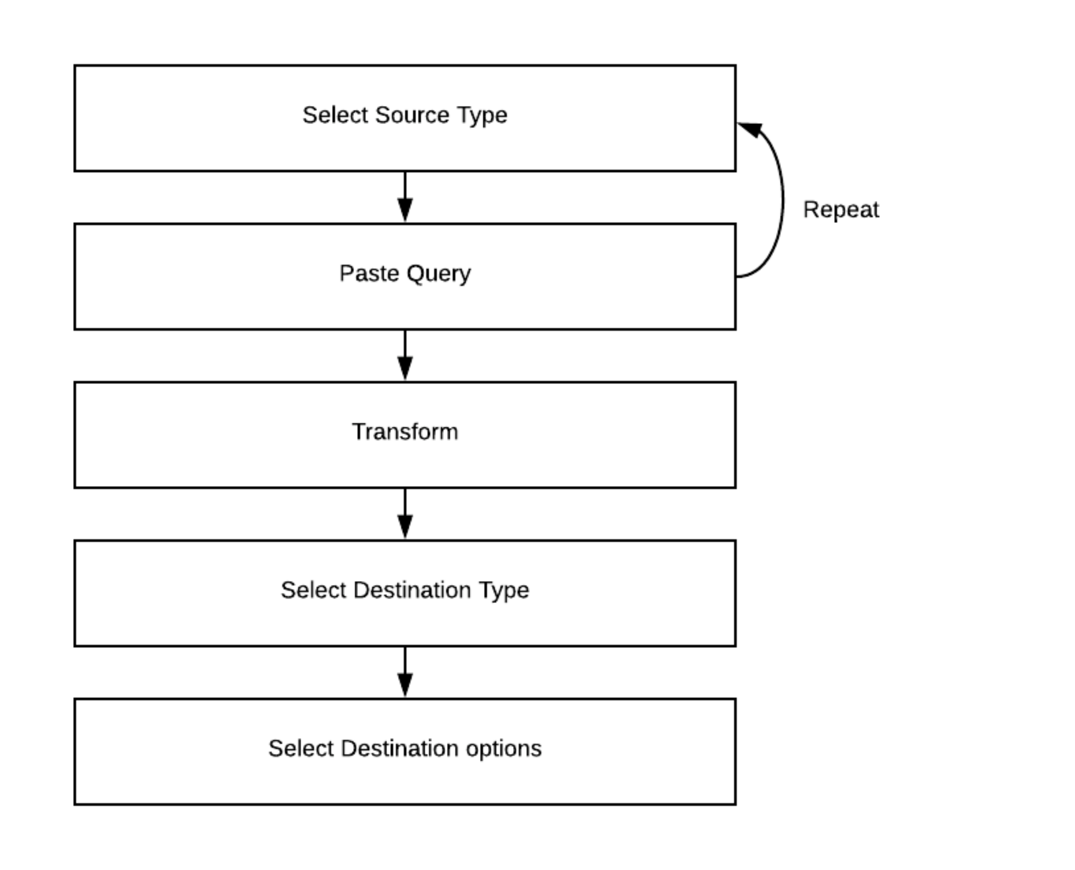
Vision

A user-friendly interface to allow power-users to:

- Orchestrate data ingestion and transformation.
- Automatically compile DAG’s
- Link ETL to reports

ETL Builder

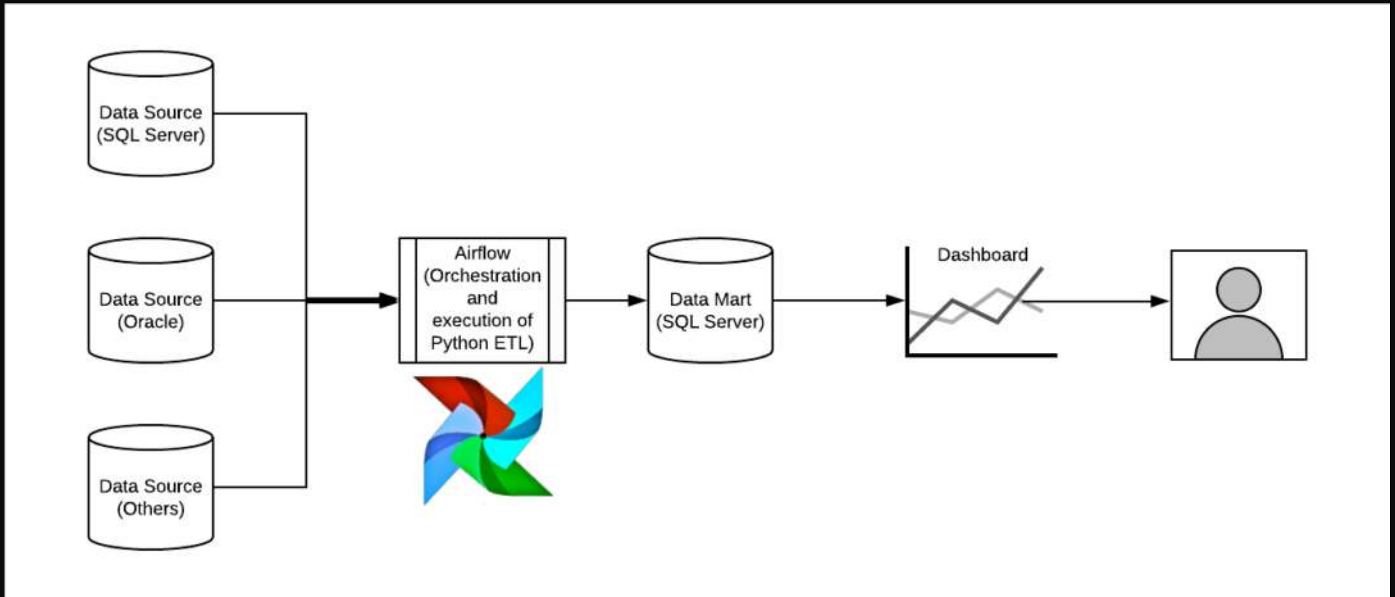
- Use Web portal to build ETL’s without coding knowledge



Solution - Requirements (I)

Requirements for the solution:

- UI for defining DAGS
- SQL Command Box
- Dependencies Set
- Version Control



Solution – Requirements (II)

- Data Repositories as Source
- Data Processing with SQL
- SQL Server as Destination

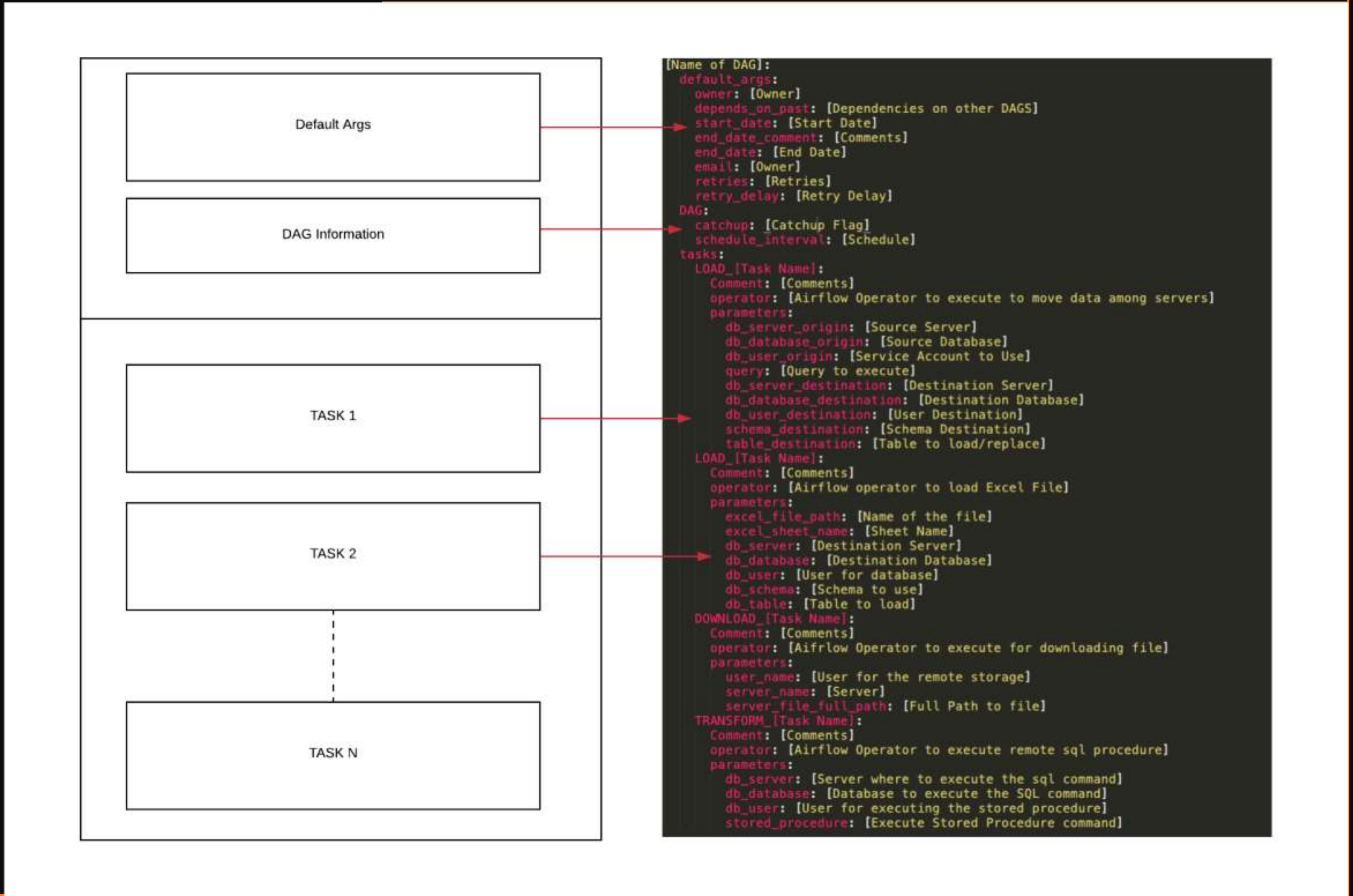
Solution - Requirements (III)

Version Control

Solution – UI (IV)

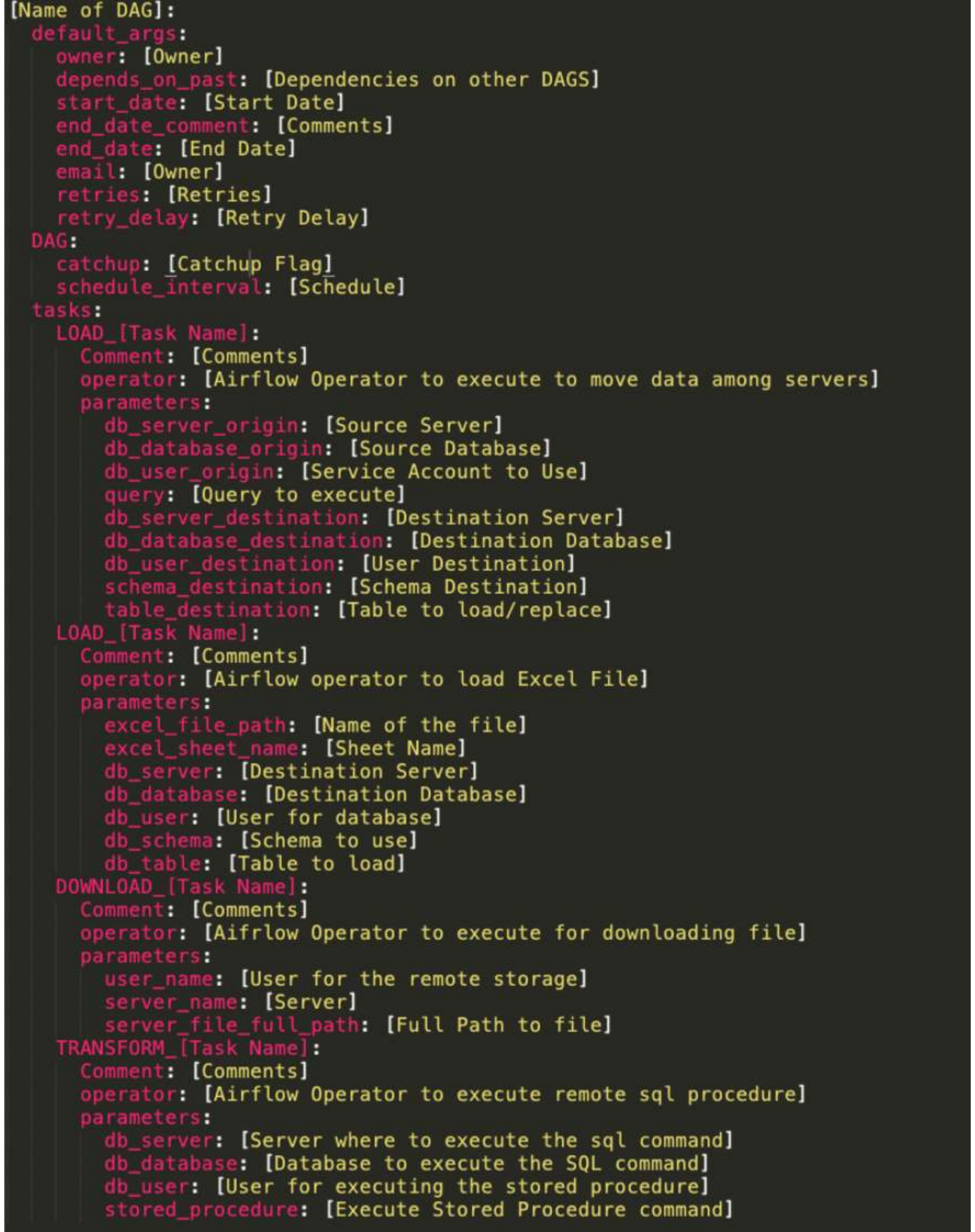
First step is to create the GUI for:

- Working as interface with users
- Allow to define DAG actions
- Generate YAML behind scenes
- Version Control



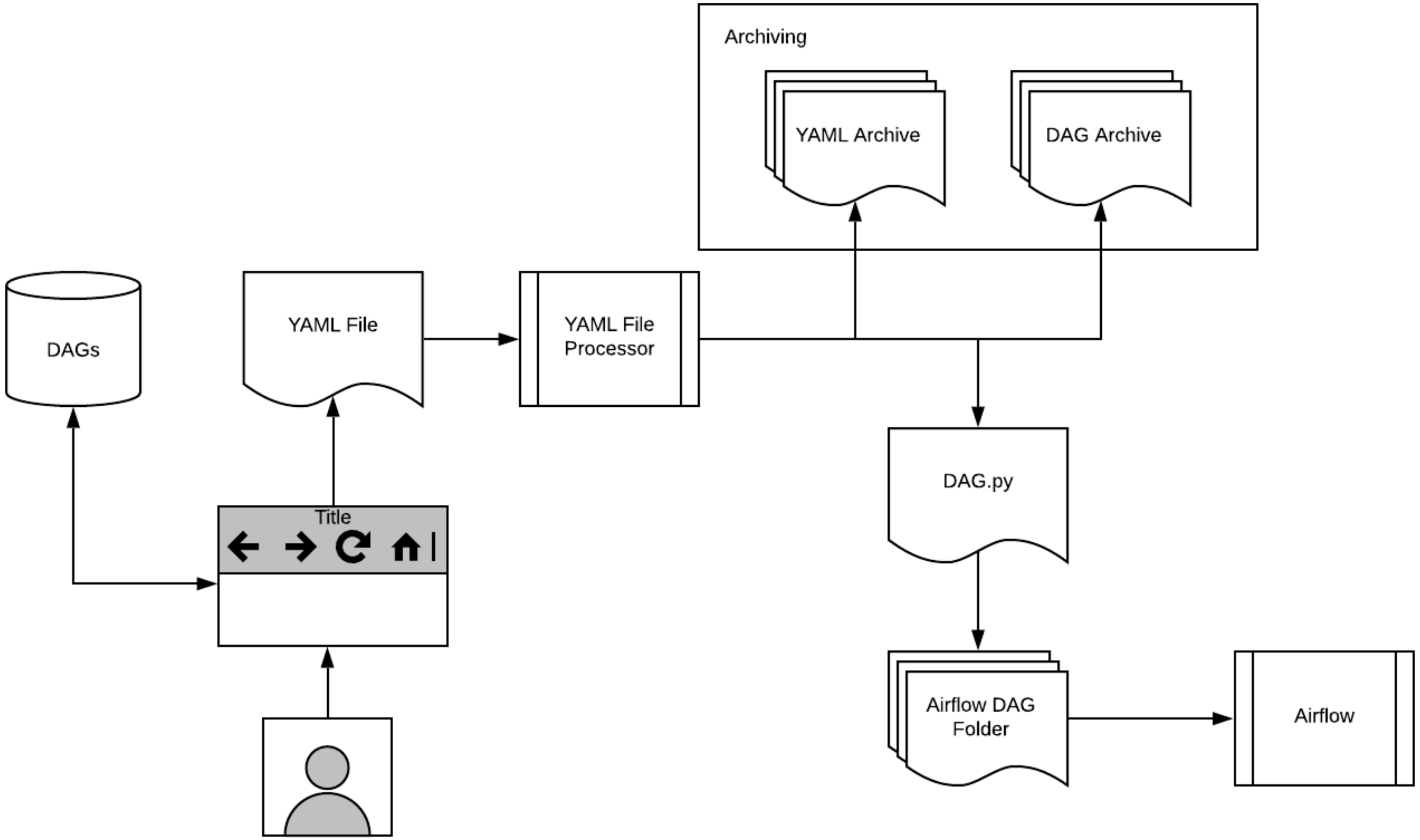
```yaml
[Name of DAG]:
  default_args:
    owner: [Owner]
    depends_on_past: [Dependencies on other DAGS]
    start_date: [Start Date]
    end_date_comment: [Comments]
    end_date: [End Date]
    email: [Owner]
    retries: [Retries]
    retry_delay: [Retry Delay]
  DAG:
    catchup: [Catchup Flag]
    schedule_interval: [Schedule]
  tasks:
    LOAD_[Task Name]:
      Comment: [Comments]
      operator: [Airflow Operator to execute to move data among servers]
      parameters:
        db_server_origin: [Source Server]
        db_database_origin: [Source Database]
        db_user_origin: [Service Account to Use]
        query: [Query to execute]
        db_server_destination: [Destination Server]
        db_database_destination: [Destination Database]
        db_user_destination: [User Destination]
        schema_destination: [Schema Destination]
        table_destination: [Table to load/replace]
    LOAD_[Task Name]:
      Comment: [Comments]
      operator: [Airflow operator to load Excel File]
      parameters:
        excel_file_path: [Name of the file]
        excel_sheet_name: [Sheet Name]
        db_server: [Destination Server]
        db_database: [Destination Database]
        db_user: [User for database]
        db_schema: [Schema to use]
        db_table: [Table to load]
    DOWNLOAD_[Task Name]:
      Comment: [Comments]
      operator: [Aifrlow Operator to execute for downloading file]
      parameters:
        user_name: [User for the remote storage]
        server_name: [Server]
        server_file_full_path: [Full Path to file]
    TRANSFORM_[Task Name]:
      Comment: [Comments]
      operator: [Airflow Operator to execute remote sql procedure]
      parameters:
        db_server: [Server where to execute the sql command]
        db_database: [Database to execute the SQL command]
        db_user: [User for executing the stored procedure]
        stored_procedure: [Execute Stored Procedure command]
```

Solution – YAML File (VI)



```yaml
[Name of DAG]:
  default_args:
    owner: [Owner]
    depends_on_past: [Dependencies on other DAGS]
    start_date: [Start Date]
    end_date_comment: [Comments]
    end_date: [End Date]
    email: [Owner]
    retries: [Retries]
    retry_delay: [Retry Delay]
  DAG:
    catchup: [Catchup Flag]
    schedule_interval: [Schedule]
  tasks:
    LOAD_[Task Name]:
      Comment: [Comments]
      operator: [Airflow Operator to execute to move data among servers]
      parameters:
        db_server_origin: [Source Server]
        db_database_origin: [Source Database]
        db_user_origin: [Service Account to Use]
        query: [Query to execute]
        db_server_destination: [Destination Server]
        db_database_destination: [Destination Database]
        db_user_destination: [User Destination]
        schema_destination: [Schema Destination]
        table_destination: [Table to load/replace]
    LOAD_[Task Name]:
      Comment: [Comments]
      operator: [Airflow operator to load Excel File]
      parameters:
        excel_file_path: [Name of the file]
        excel_sheet_name: [Sheet Name]
        db_server: [Destination Server]
        db_database: [Destination Database]
        db_user: [User for database]
        db_schema: [Schema to use]
        db_table: [Table to load]
    DOWNLOAD_[Task Name]:
      Comment: [Comments]
      operator: [Aifrlow Operator to execute for downloading file]
      parameters:
        user_name: [User for the remote storage]
        server_name: [Server]
        server_file_full_path: [Full Path to file]
    TRANSFORM_[Task Name]:
      Comment: [Comments]
      operator: [Airflow Operator to execute remote sql procedure]
      parameters:
        db_server: [Server where to execute the sql command]
        db_database: [Database to execute the SQL command]
        db_user: [User for executing the stored procedure]
        stored_procedure: [Execute Stored Procedure command]
```

Solution – YAML File Processor (V)



Achievements

Empower users for creating DAGS with 0 code

Data Transformation and Data Loading on demand

Democratize access to ETL

Savings in Alteryx Licenses

Challenges of first version

- Logic to recreate the same DAG
- Extend to different databases (Oracle, Teradata)
- Stop using Airflow server as processing server (move to Kubernetes + Docker)
- Collaboration among users