4. Data Loading with Talend

Responsible & contact points:

- **Alessandro Mainardi - Project Owner**
- **Simon Bourguignon - Delivery Manager**
- **Alba Carrero/ Gaetan Frenoy - Product Owner**
- **Rui Ferraz - Project Manager**

4.1 - Talend Integration

Source data integration with Talend ETL tool

Connection details:

- **FTP Server (Meteologica):** ftp.meteologica.com (incremental load)
  - Talend connects to the FTP server where files containing current and future electricity prices for Spain, Italy, France, and Germany are located.
  - Talend retrieves these files from the FTP server.
  - The retrieved files are then loaded into Google Cloud Storage.
  - We need to use GCP RE on this source because in order to access Meteologica, it is required statistic IP in order to register the whitelist for data access.
- ~~**Postgres Database (Vendohm):** ses-gcp-sandbox.eu.vendohm -~~ **Decommission replaced by CSV from Dataiku**
  - ~~Talend establishes a direct connection with the Vendohm PostgreSQL database that stores sensor values identified by specific curve IDs.~~
- **Dataiku files (full load)**
  - Vendohm is decommissioned and IRM energy data need to calculated many rules. Therefore, project team decide to get the final calculation of energy from Dataiku.
    - Dataiku will generate energy files (Spot hour, Spot day and Forward combines to 1 consolidation file) as .gz file to DataOcean Industrial bucket (cs-ew1-prj-data-dm-industrial-[env]-staging/in/enr/irm-conso) twice a day
    - Dataiku will generate a file to replace IRM energy deal to DataOcean Industrial bucket (cs-ew1-prj-data-dm-industrial-[env]-staging/in/enr/irm-deals) once a day
  - Talend gets the the files considering GCS as data source
  - This extracted data is loaded into Google Cloud Storage as files again in the standard way.
  - Dataiku project
- ~~**Oracle Database (IRM):** acew1pirmdb01.prod.aws.cloud.solvay.com~~
  - ~~Talend establishes a direct connection with the IRM Oracle database that stores energy deals identified by specific deal IDs.~~
  - ~~It extracts relevant data from the Oracle database.~~
  - ~~This extracted data is loaded into Google Cloud Storage as files.~~
- **Google Sheets (Hedges, wap solid fuels, CO2): (full load)**
  - Talend integrates with Google Sheets, where solid fuel wap, CO2 emissions, and hedging information are stored.
  - It retrieves this data from Google Sheets.
  - Similar to the other sources, this data is also loaded into Google Cloud Storage as files.
    - CO2
    - WAP Solid Fuels
    - ~~Energy Deals Hubs~~
    - ~~Energy Deals Counter Parties~~
    - ~~Energy Deals Sites Hedges~~
    - ~~Vendohm Forwards~~
    - ~~Vendohm Meteo~~
    - ~~Vendohm Spot~~

Data Transformation and Loading to Google BigQuery:

- Once data from all four sources (FTP server,~~ PostgreSQL database~~, ~~Oracle database~~, Google Sheets, Dataiku file) is available in Google Cloud Storage as files,
  Talend proceeds with data transformation and loading.
- Talend performs data transformations as needed, including cleansing, mapping, and structuring the data for consistency.
- The transformed data is loaded into various stages, operational data stores (ODS), and data mart tables within Google BigQuery.
- These tables are organized to facilitate efficient querying and reporting for energy optimization purposes.

By utilizing Talend for data extraction, transformation, and loading (ETL), the web app ensures that data from diverse sources is collected, processed, and structured for analysis and reporting within Google BigQuery, enabling users to make informed decisions based on up-to-date and accurate data.

GCP Details:

Domain	GCP Project	Bucket
Industrial	prj-data-dm-industrial-[env]	cs-ew1-prj-data-dm-industrial-[env]-staging/ROBUSTIFY
Sustainability	prj-data-dm-sust-[env]	cs-ew1-prj-data-dm-sust-[env]-staging/ROBUSTIFY
~~Robustify~~	~~prj-data-robustify-[env]~~	~~N/A~~

Cloud Remote Engine = prj-talend-[dev] / ce-ew1-b-talend-gcp-remote-engine-[env]

4.2 - Source Data Extraction

Main jobs for source extraction	• **J001_FTP_to_GCS-METEOLOGICA_FRANCE** • **J002_FTP_to_GCS-METEOLOGICA_ITALY** • **J003_FTP_to_GCS-METEO**	**--to the top --** **GCP Remote Engine(RE)** **Industrial Data Ocean(DO) project**

	LOGICA_GERMANY • **J004_FTP_to_GCS-METEOLOGICA_SPAIN** • ~~**J005_FTP_to_GCS-METEOLOGICA_SPAIN_ENS**~~ • ~~**J006_FTP_to_GCS-METEOLOGICA_SPAIN_OBS**~~	
	Job description by steps	**Job design**

	1. FTP Server Connection:
	   a. Establishes a secure connection to the Meteologica FTP server.
	   b. Retrieves files from a predefined folder located on the FTP server.
	2. File Consolidation and Renaming:
	   a. After fetching	

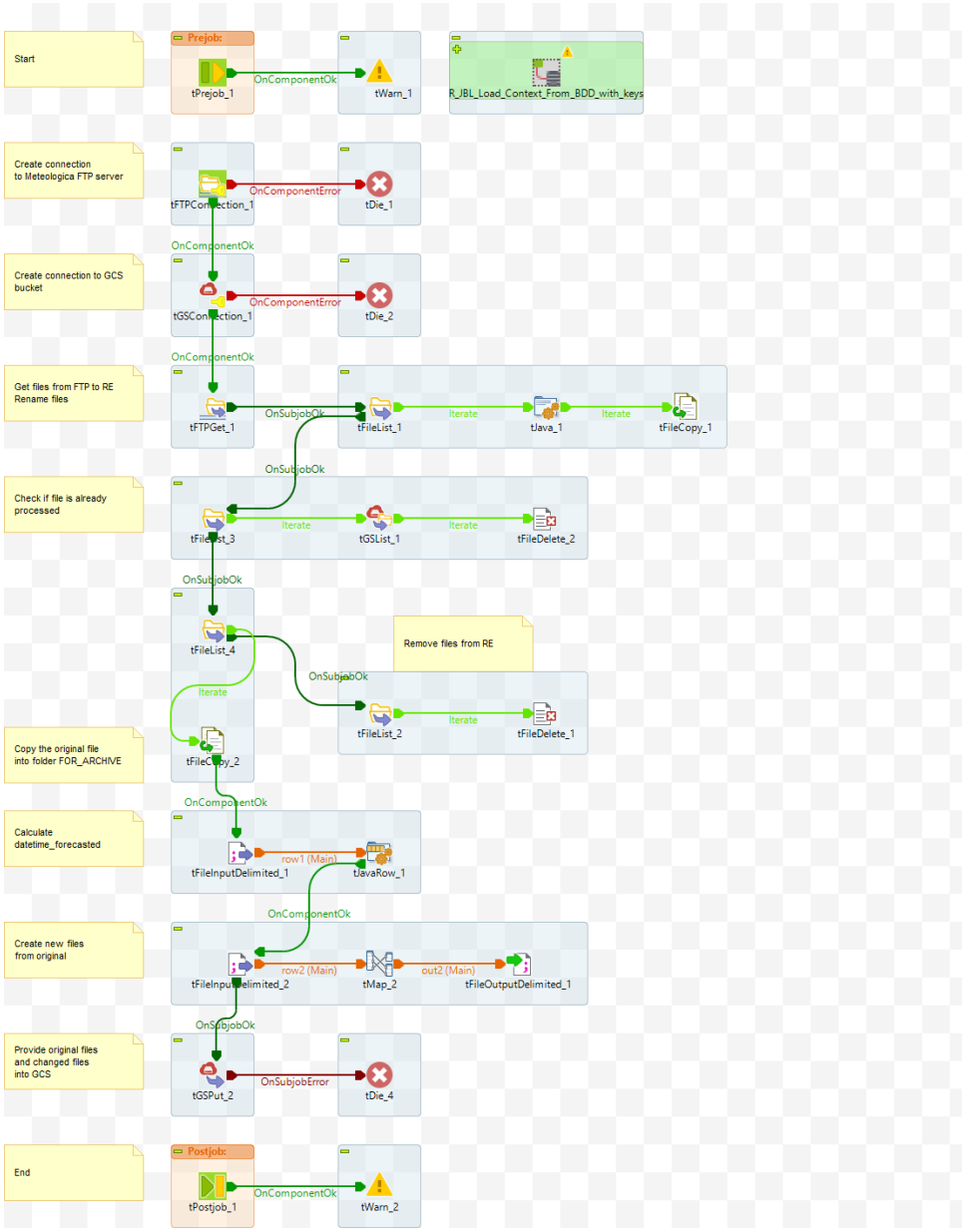
the files, the job consolidates and renames them according to predefined naming conventions.

b. This step ensures uniformity and consistency in file naming for further processing.

3. Creating File Backups:

a. As a best practice for data integrity, the job generates copies of the original files.

b. These copies are stored in an archive folder on the rem



ote engine for backup and historical reference.

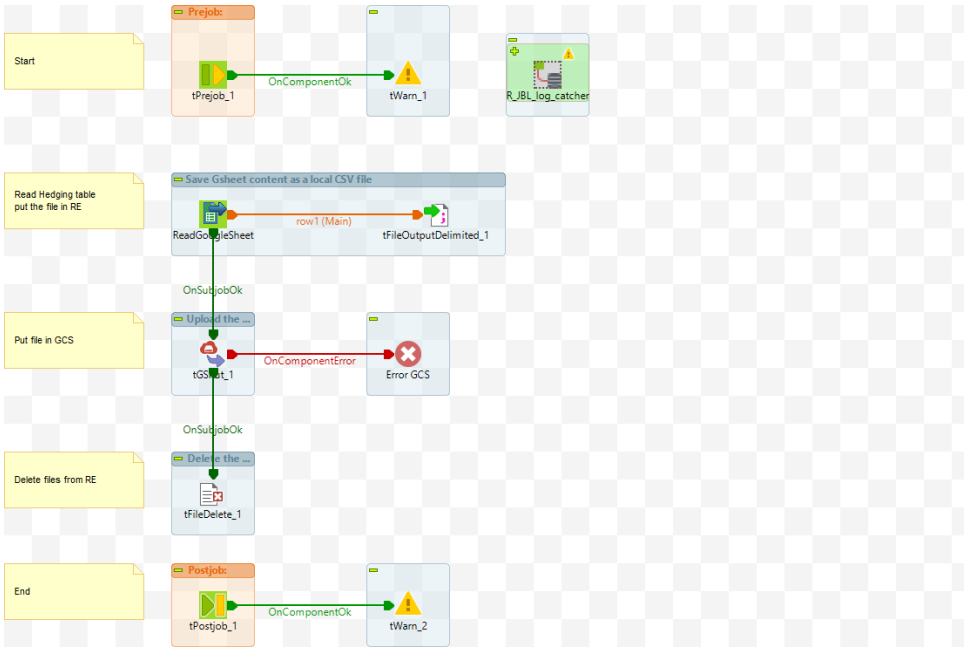
4. Checking for Previously Processed Files:
    a. The job performs a check to determine if the pulled files have already been processed.
    b. This verification is crucial to prevent the reprocessing of files from previous days, as older files may still exist in the folder.
5. Data Extraction and Transformation:
    a. For files that have not been processe

d earlier, the job proceeds with data analysis.
   b. It extracts a datetime value from each file, located at a specified position within the file.
   c. Using this datetime information, the job calculates a new column, enabling precise timestamping of the data.
6. CSV Output and Google Storage:
   a. The processed data, now enriched with the new timestamp colu

mn, is converted into a CSV format.
   b. The resulting CSV output files are then pushed to Google Cloud Storage for further storage and access.
7. File Deletion:
   a. Once the files have been successfully processed and their data stored in Google Cloud Storage, they are deleted from the remote engine.
   b. This deletion step helps manage

	storage resources efficiently and ensures that only processed data is retained. This Talend job ensures the efficient handling of data from the Meteologica FTP server, including consolidation, backup, and validation steps. It also extracts and enriches the data, making it ready for use in downstream processes, ultimately contributing to the accuracy and usability of the energy optimization project's data.	
Main jobs for source extraction	• **J008_GSheet_to_GCS_WAP_SOLID_FUELS** • ~~**J011_GSheet_to_GCS_VENDOHM_FORWARDS**~~ • ~~**J012_GSheet_to_GCS_VENDOHM_SPOT**~~ • ~~**J013_GSheet_to_GCS_VENDOHM_METEO**~~ • ~~**J015_GSheet_t**~~	**--to the top --** **GCP Remote Engine** **Industrial DO project**

	~~o_GCS_IRM_HUBS~~ • ~~J016_GSheet_to_GCS_IRM_COUNTER_PARTIES~~ • ~~J017_GSheet_to_GCS_IRM_SITES_HEDGES~~	
	Job description by steps	**Job design**

	1. Google Sheets Connection:
	   a. The job initiates a connection to a specified Google Sheets file.
	   b. It targets a specific sheet within the Google Sheets document, identified by its unique ID.
	2. Data Extraction and Formatting:
		

a. The job extracts data from the designated Google Sheets sheet.
b. The extracted data is converted into a CSV format with a predefined fixed schema.
c. The schema format is consistent and predefined to ensure data uniformity and structure.
d. The resulting CSV file includes a predefined file name that incorporates the date of

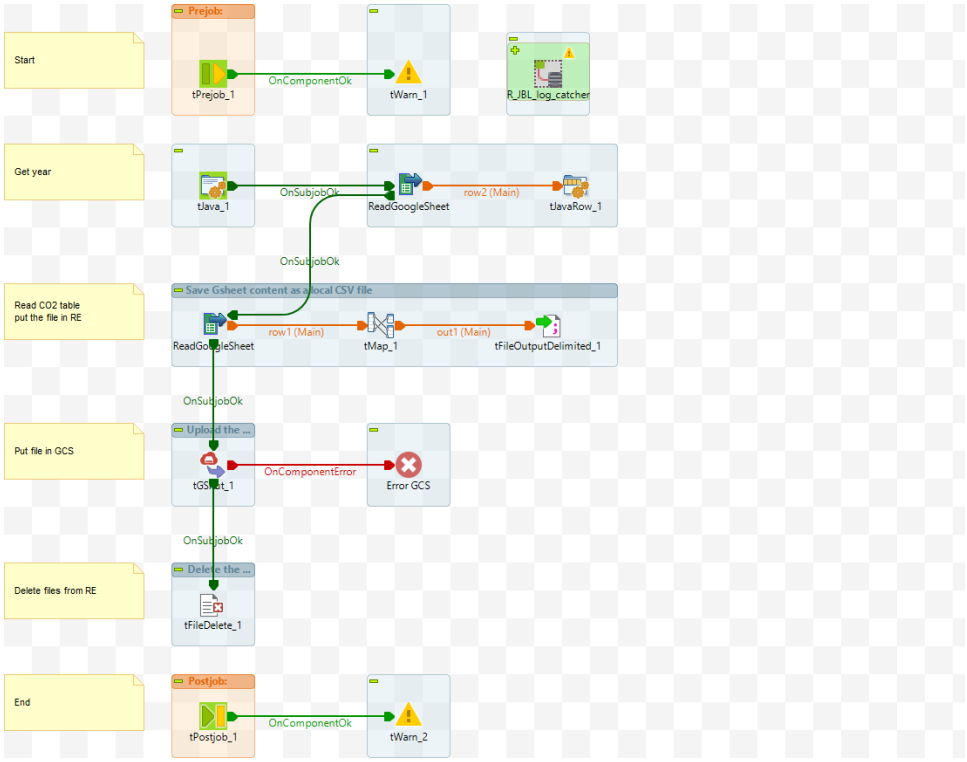
extraction.

3. Copying to Google Storage:
    a. After the successful extraction and formatting of the data, the job copies the CSV file to Google Cloud Storage.
    b. This step serves as a secure and reliable means of storing the processed data in the cloud.
4. File Deletion:
    a. Following the successful copying of the CSV file to Google Cloud Storage,

	the original file located on the remote engine is deleted. b. This cleanup step helps manage storage resources efficiently and maintains data integrity. This Talend job efficiently extracts, formats, and securely stores data from a Google Sheets document into Google Cloud Storage, ensuring that the data is readily available for further analysis and processing while adhering to a fixed schema and naming conventions.	

Main jobs for source extraction	• **J009_GSheet_to_GCS_CO2_EMISSIONS**	**--to the top --** **GCP Remote Engine** **Sustainability DO project**
	Job description by steps	**Job design**
	1. Google Sheets Connection:	

a. The job establishes a connection to a specified Google Sheets file.
b. It focuses on a particular sheet within the Google Sheets document, identified by its unique ID.



2. Data Extraction and Formatting:
   a. The job extracts data from the designated Google Sheets sheet.
   b. The extracted data is transformed into a CSV format, adhering to a predefined, fixed schema.
   c. The schema format is consistent and predefined

to ensure uniformity and data structure.

d. The resulting CSV file follows a predefined naming convention, which incorporates the date of extraction.

3. Data Reading:

a. The job involves two distinct data reading steps.

i. First Reading: The job reads only a single cell within the

e file, specifically the cell containing the year value. This is set to capture year-related data.

ii. Second Readin

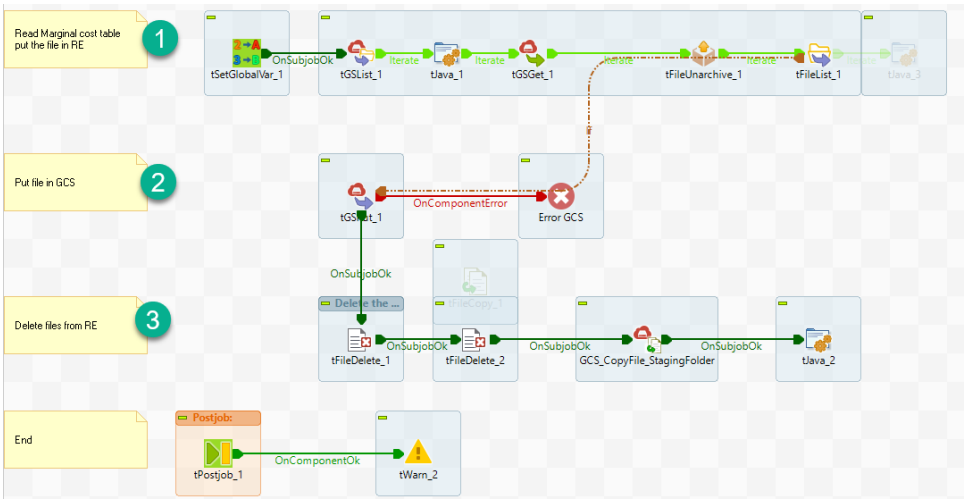
	g: The second read of the file is reserved for capturing the remaining data, defined by the schema and for

mat. This step comprehensively reads the data as specified in the schema.

4. Copying to Google Storage:
    a. After successfully extracting and formatting the data, the job copies the CSV file to Google Cloud Storage (inside Sustainability

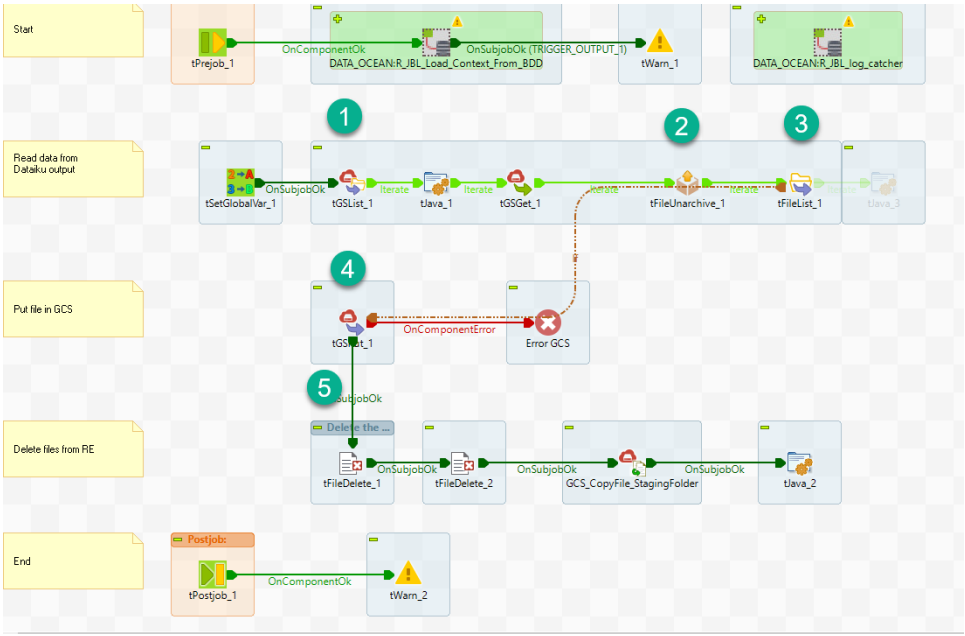
	domain). b. This step serves as a secure and reliable means of storing the processed data in the cloud. 5. File Deletion: a. Following the successful copying of the CSV file to Google Cloud Storage, the original file located on the remote engine is deleted. b. This cleanup step helps manage storage resources efficiently and maintains data integrity. This Talend job effectively extracts, formats, and securely stores data from a Google Sheets document into Google Cloud Storage. It features specialized data reading

	steps, ensuring the capture of year-related data separately, while adhering to predefined schemas and naming conventions. The deletion of the original file enhances data management and resource optimization.	

Main jobs for source extraction	• ~~**J010_Postgres_to_GCS_VENDOHM**~~ **Decommission** • **J010_FIL_Energy_F_O_TO_ODS**	**--to the top --** **AWS Remote Engine** **Industrial DO project**
	Job description by steps	**Job design**
	GCS Connection: ○ Dataiku will generate energy a file that calculated many rules from the Energy team and save to GCS at DataOcean Industrial bucket (cs-ew1-prj-data-dm-industrial-[env]-staging/in/enr) twice a day to get the historical data to future hourly. The mapping file and fields 1. First loading the data from cs-ew1-prj-data-dm-industrial-[env]-staging/in/enr to local drive and unzip the file 2. Upload the file to cs-ew1-prj-data-dm-industrial-[env]-	

	staging /ENERGY/irm-conso 3. Storage in Google Cloud Storage (GCS) : ○ After the successful extraction and formatting of the data, the job securely stores the CSV file in Google Cloud Storage. ○ The files are placed within a designated "ENERGY" folder in the GCS location for efficient storage and accessibility. ○ Also move the source file .gz to folder "Processed" ○ Following a successful data extraction and storage process, the original file located on the remote engine is deleted to manage storage resources efficiently. ○ The connection to the Oracle database is closed to ensure proper resource management and security.	
Main jobs for source extraction	• ~~**J014_Oracle_to_GCS_IRM**~~ **J014_GCS_to_GCS_IRM**	**--to the top --** **AWS Remote Engine** **Industrial DO project**

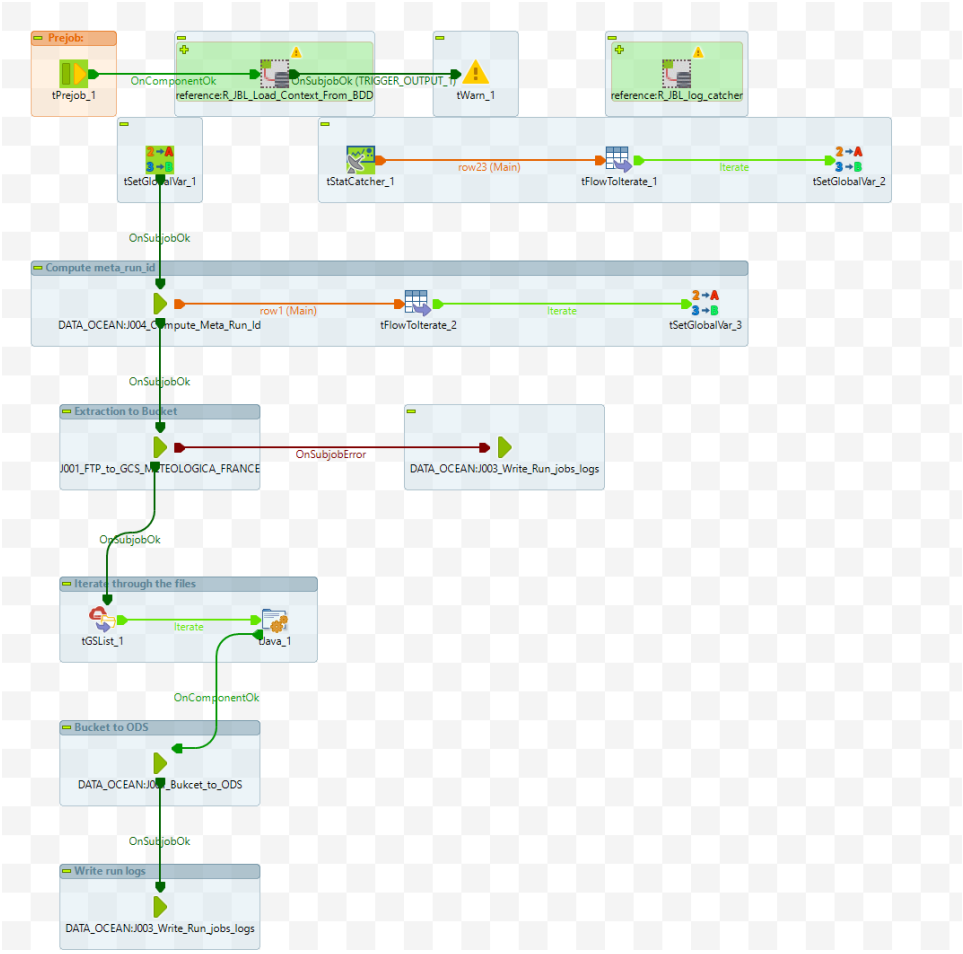
	_ENERGY_DEAL	
	Job description by steps	**Job design**

	1. Download file from GCS (output of Dataiku) cs-ew1-prj-data-dm-industrial-[env]-staging/in/enr/irm-deals
	2. Unzip the file and keep to the GCS cs-ew1-prj-data-dm-industrial-[env]-staging /ENERGY/irm-deals
	  • The data is structured according to a predefined and fixed schema to maintain data consistency.
	3. Data Formatting and CSV Output:
	  • The extracted data is formatted and transformed into CSV format.
	  • Each resulting CSV file includes the table name and the date of extraction for reference and organization.
	4. Storage in Google Cloud Storage (GCS):
	  • After the successful extraction and formatting of the data, the job securely stores the CSV file in Google Cloud Storage.
	  • The files are placed
		

	within a designated "IRM" folder in the GCS location for efficient storage and accessibility. 5. File Deletion and Connection Closure: • Following a successful data extraction and storage process, the original file located on the remote engine is deleted to manage storage resources efficiently. • Source file from Dataiku will move to folder Process and rename to the file + timestamps	

4.3 - Data Loading to Google BigQuery

Main jobs for source extraction	• **J001_Extraction_til_ODS_METEOLOGICA_FRANCE** • **J002_Extraction_til_ODS_METEOLOGICA_ITALY** • **J003_Extraction_til_ODS_METEOLOGICA_GERMANY** • **J004_Extractio**	**--to the top --** **GCP Remote Engine** **Industrial DO project**

	n_til_ODS_METEOLOGICA_SPAIN • ~~**J005_Extraction_til_ODS_METEOLOGICA_SPAIN_ENS**~~ • ~~**J006_Extraction_til_ODS_METEOLOGICA_SPAIN_OBS**~~	
	Job description by steps	**Job design**
	Job Initialization and Logging: 1. Job Run Initialization (Subjob 1): a. The job starts with the first subjob, where the job ID is retrieved, and the extraction date is calculated. b. This crucial step initializes the job and captures	

essential metadata.

Data Extraction and Processing:

1. Files Extraction from FTP Server (Subjob 2):
    a. In the next subjob, data files are extracted from the Meteologica FTP server.
2. Error Handling and Logging (Subjob 3 - On Failure):
    a. If the data extraction subjob (Subjob 2) encounters an issue or does not finish successfully, the job transitions to a subjob for writing error logs.

b. Logging is essential for tracking and diagnosing issues in the data extraction process.

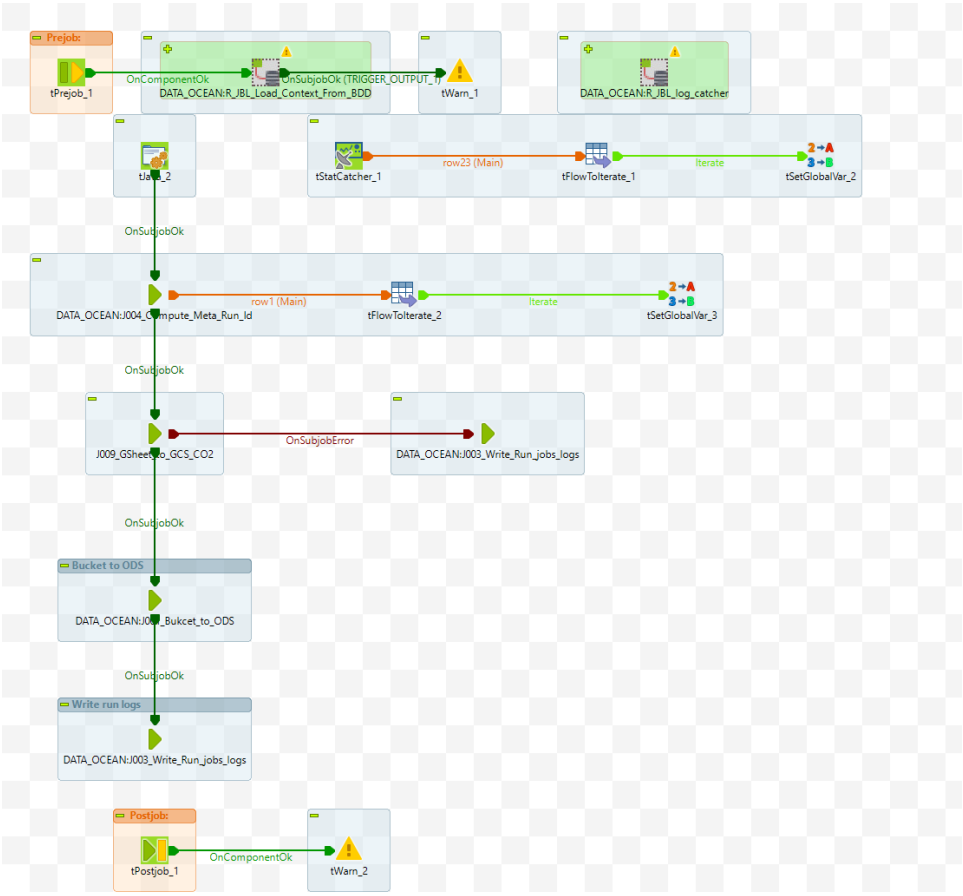
3. Data Loading and Transformation (Subjob 4 - On Success):
    a. When the data extraction subjob (Subjob 2) successfully completes, the job proceeds with data loading and transformation.
    b. In this stage, each extracted file is processed one by one.
    c. For each file, a subjob

is called to load data from the CSV file into a stage table, preparing it for further processing.

d. Subsequently, data from the stage table is loaded into an operational data store (ODS) table.
e. All necessary parameters, such as table names and connection parameters, are provided to ensure accurate data extraction and loading.

	Logging and Reporting: 1. Logging (Subjob 5): a. At the end of the data loading and transformation process, a subjob is called to write logs. b. Logging captures critical information about the data processing, ensuring transparency and traceability. This Talend job orchestrates the data extraction, loading, and transformation process. It begins with initialization and metadata capture, proceeds to extract data from the FTP server, handles errors if they occur, and performs data loading and transformation for energy optimization purposes. Detailed logging and

	error handling mechanisms enhance job monitoring and maintain data integrity.	
Main jobs for source extraction	• **J008_Extraction_til_ODS_GSHEET_WAP_SOLID_FUELS** • **J009_Extraction_til_ODS_GSHEET_CO2** • ~~**J011_Extraction_till_ODS_GSHEET_VENDOHM_FORWARDS**~~ • ~~**J012_Extraction_till_ODS_GSHEET_VENDOHM_SPOT**~~ • ~~**J013_Extraction_till_ODS_GSHEET_VENDOHM_METEO**~~ • ~~J014_Extraction_til_ODS_ORACLE_IRM~~ **J014_FIL_IRM_ENERGY_DEAL_F_O_TO_ODS** • ~~**J015_Extraction_till_ODS_GSHEET_IRM_HUBS**~~ • ~~**J016_Extraction_till_ODS_GSHEET_I**	**--to the top --** **GCP Remote Engine** **Industrial DO project** **Exception :** • **J009_Extraction_til_ODS_GSHEET_CO2 on GCP RE and on Sustainability DO project** • **J014_FIL_IRM_ENERGY_DEAL_F_O_TO_ODS on AWS RE and on Industrial DO project**

	~~RM_COUNTER_PARTIES~~ • ~~J017_Extraction_till_ODS_GSHEET_IRM_SITES_HEDGES~~	
	Job description by steps	**Job design**
	Job Initialization and Logging: 1. File Name Configuration (Global Variable): a. The job begins by configuring the file name as a global variable. b. This variable is used throughout the job to ensure consistency in file handling. 2. Job Run Initialization (Subjob 1): a. In the first subjob,	

the job ID is retrieved, and the extraction date is calculated.
b. This step sets the stage for the data processing and captures essential metadata.

Data Extraction from Google Sheets / IRM database:

1. Files Extraction from Google Sheets / IRM database (Subjob 2):
    a. The next subjob is responsible for extracting data from Google Sheets/ IRM database.
    b. It targets the specified file base

d on the global variable (filename) configured earlier.

2. Error Handling and Logging (Subjob 3 - On Failure):
    a. In case the data extraction subjob (Subjob 2) encounters any issues or doesn't finish successfully, the job transitions to a subjob for writing error logs.
    b. Logging is crucial for tracking and diagnosing problems during the data extraction

process.

Data Loading and Transformation:

1. Data Loading and Transformation (Subjob 4 - On Success):
    a. When the data extraction subjob (Subjob 2) successfully completes for the provided file name, the job proceeds with data loading and transformation.
    b. A subjob is called to load data from the CSV file into a stage table, preparing it for further processing.
    c. Subsequently

, data from the stage table is loaded into an operational data store (ODS) table.
   d. All necessary parameters, such as table names and connection parameters, are provided to ensure accurate data extraction and loading.

Logging and Reporting:

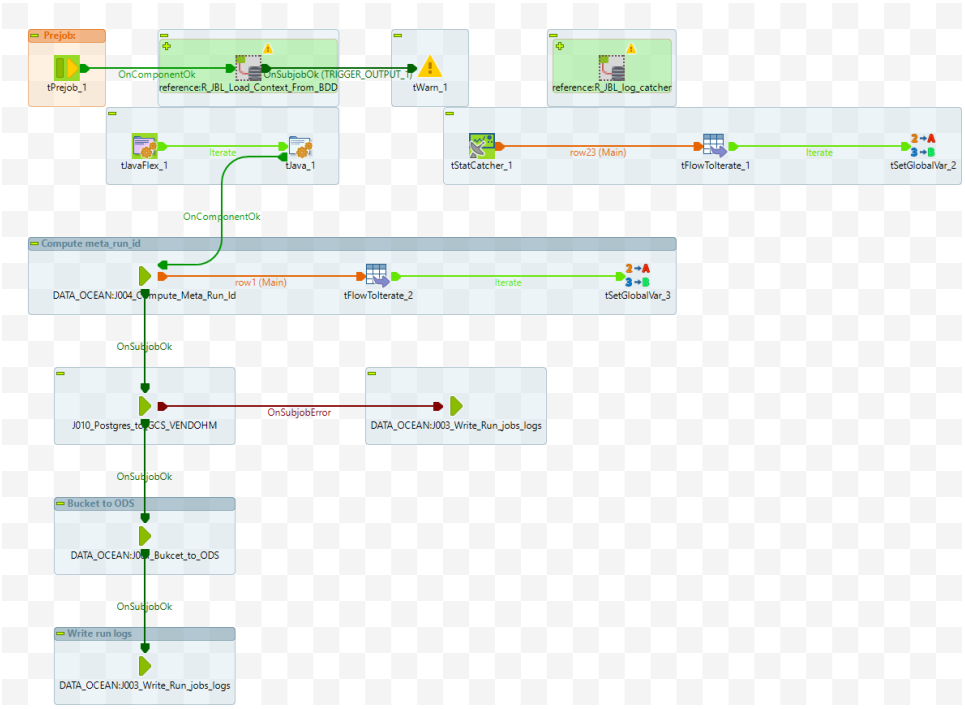
1. Logging (Subjob 5):
   a. At the end of the data loading and transformation process, a subjob is called to

	b. Logging captures crucial information about the data processing, enhancing transparency and traceability. This Talend job streamlines the data extraction, loading, and transformation process. It begins with file name configuration, proceeds with metadata capture, extracts data from Google Sheets, handles errors if they occur, and performs data loading and transformation, all while maintaining detailed logs for monitoring and maintaining data integrity.	

Main jobs for source extraction	• ~~**J010_Extraction_til_ODS_VENDOHM**~~	**--to the top --** **GCP Remote Engine** **Industrial DO project**
	Job description by steps	**Job design**
	~~Global Variable Configuration:~~ 1. ~~File Names~~	

~~and Table Names Configuration (Global Variables postgresTableNames):~~

a. ~~The job starts by configuring global variables for file names and table names.~~
b. ~~These variables are populated by reading a list of all tables and files that subjobs need to load.~~
c. ~~This dynamic configuration ensures that the job adapts to the specific files and tables required for processing.~~



~~Data Extraction and Processing in Iterations:~~

1. ~~Iteration Loop (One by One):~~
    a. ~~The job processes tables and files iteratively, one by one, based on the configured global variables.~~
2. ~~Job Run Initialization (Subjob 1):~~
    a. ~~Within each iteration, the first subjob is called to retrieve the job ID and calculate the extraction date.~~
    b. ~~This initialization step captures vital metadata for the ongo~~

ing data processing.

3. Files Extraction from PostgreSQL Database (Subjob 2):
    a. In the next subjob, data files are extracted from the PostgreSQL database.
    b. Each iteration targets a specific file based on the global variables (filenames) configured earlier.
4. Error Handling and Logging (Subjob 3 - On Failure):
    a. If the data extraction subjob (Subjob 2) encounters any issues

or doesn't finish successfully, the job transitions to a subjob for writing error logs.
   b. Logging is crucial for tracking and diagnosing problems during the data extraction process.
5. Data Loading and Transformation (Subjob 4 - On Success):
   a. When the data extraction subjob (Subjob 2) successfully completes for the provided file name, the job proceeds with

~~data loading and transformation.~~

b. ~~A subjob is called to load data from the CSV file into a stage table, preparing it for further processing.~~
c. ~~Subsequently, data from the stage table is loaded into an operational data store (ODS) table.~~
d. ~~All necessary parameters, such as table names and connection parameters, are provided to~~

ensure accurate data extraction and loading.

Logging and Reporting:

1. Logging (Subjob 5):
    a. At the end of each iteration, a subjob is called to write logs.
    b. Logging captures critical information about the data processing, ensuring transparency and traceability for each processed file and table.

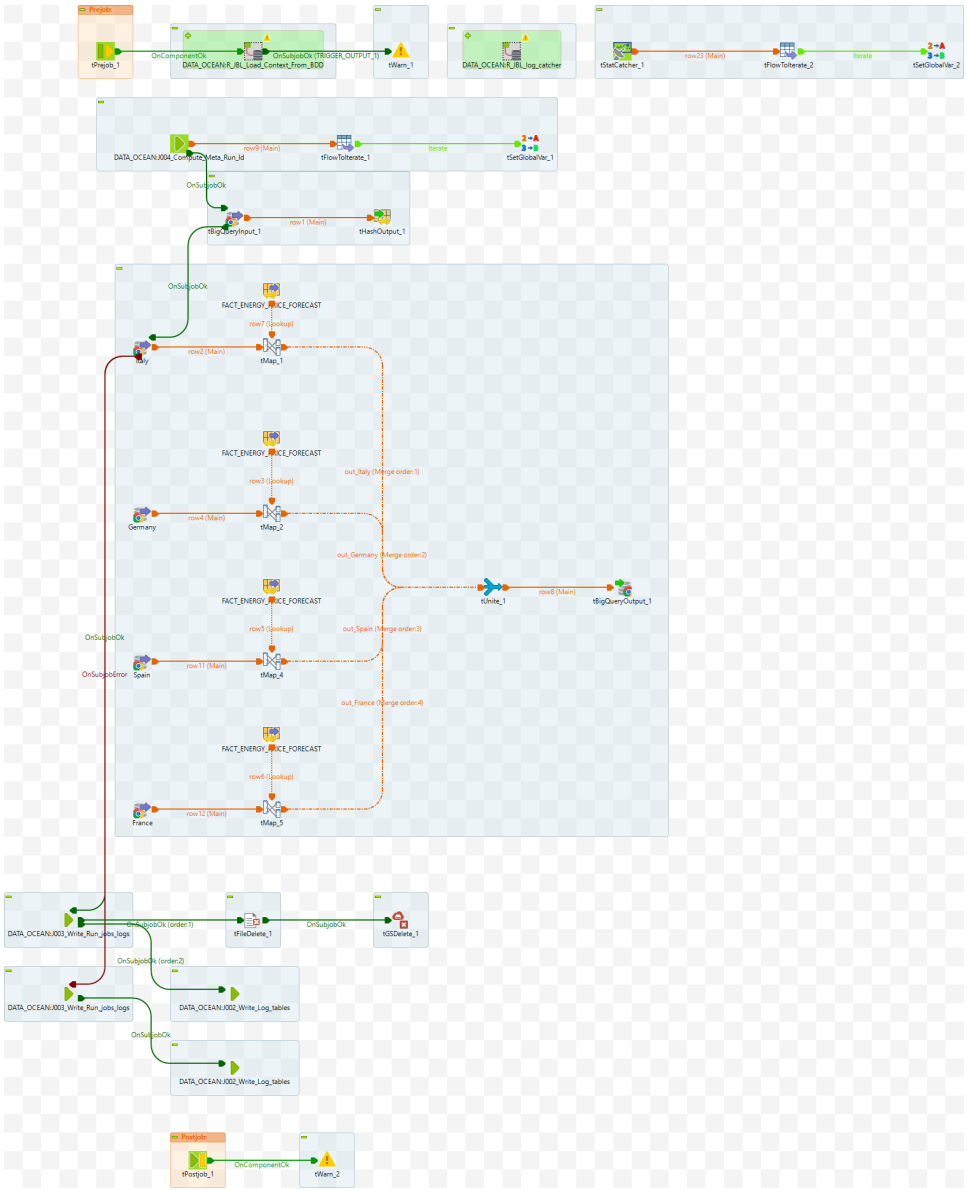
This Talend job is designed for dynamic data extraction and processing, adapting to different tables and files. It initializes each iteration with metadata capture, extracts data from the

	~~PostgreSQL database, handles errors when necessary, performs data loading and transformation with parameterized configurations, and maintains detailed logs for monitoring and data integrity across multiple files and tables.~~	

4.4 - Load to DM (calculations and transformations)

Main jobs for source extraction	• **J001_ODS_TO_DM_FACT_ENERGY_PRICE_FORECAST**	**--to the top --** **GCP Remote Engine** **Industrial DO project**
	Job description by steps	**Job design**
	Metadata Calculation and Cache Initialization: 1. Job Metadata Calculation: • The job begins by calculating essential job metadata, which include information such as job identifiers, time	

stamps, or configuration details.
    - This step sets the stage for the subsequent data processing.
2. Fact Table Keys Calculation and Caching:
    - Following metadata calculation, the job proceeds to compute and store keys associated with the current Fact table.
    - These keys are cached for efficient reference during data processing.



Data Extraction and Import:

1. Energy Forecast

Data Extraction:

- The job extracts energy forecast data from four Operational Data Store (ODS) tables, each containing information for a specific region (Italy, Germany, Spain, and France).

2. Data Comparison and Import:
   - The extracted energy forecast data is compared with the cached keys from the Fact table.
   - The primary purpose of this step

is to identify and import only new rows of data for the latest forecasted date.
- By importing only new data, the job ensures that the dataset remains up to date and avoids unnecessary duplication.

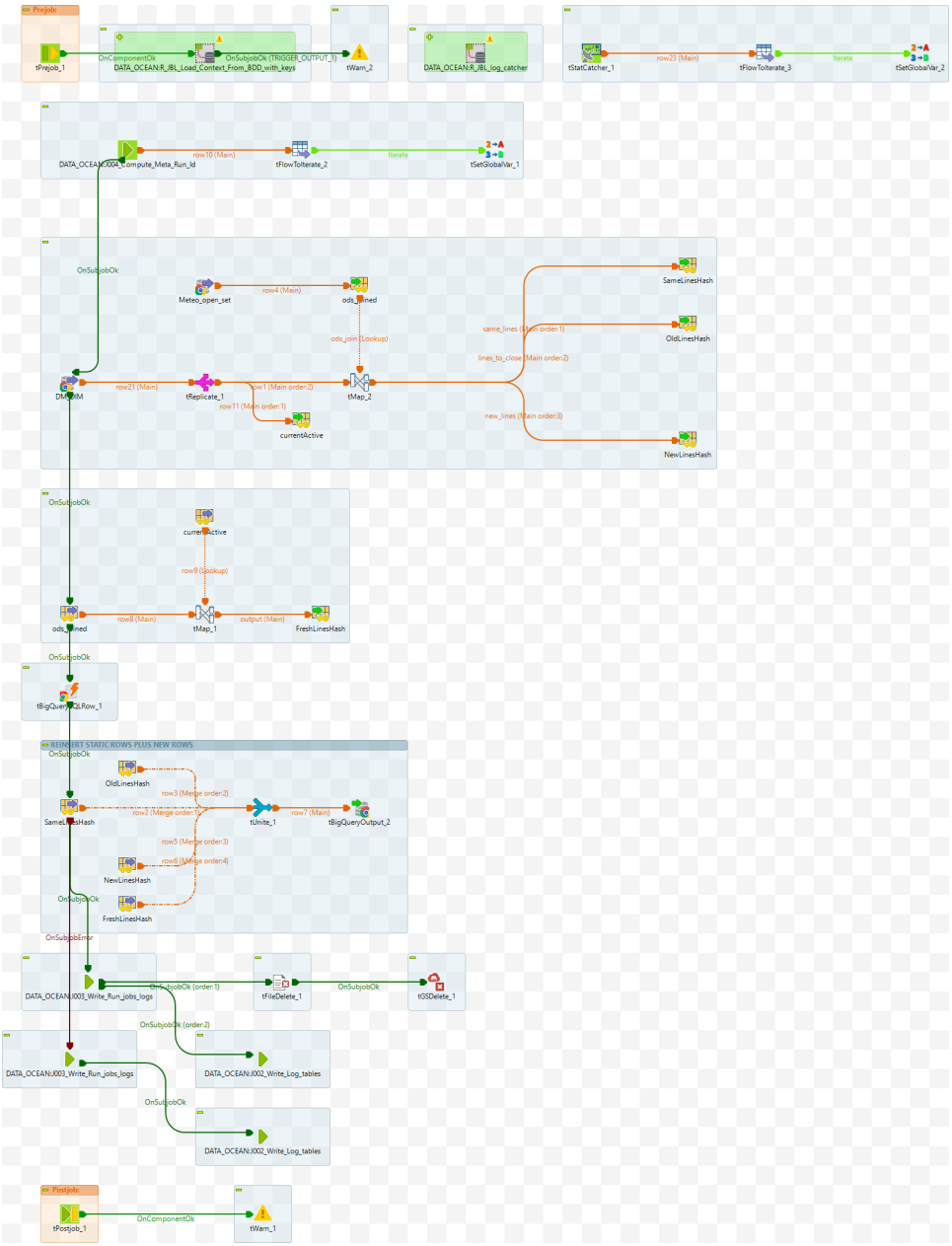
Logging and Cleanup:

1. Log Generation:
   - At the conclusion of data import and processing, the job generates logs.
   - These logs provide a record of the

	job's activities, facilitating monitoring and troubleshooting. 2. Temporary File Deletion: • As a final step, the job deletes any temporary files created during the data transfer process. • This cleanup helps manage storage resources and maintain data integrity.	
Main jobs for source extraction	• ~~**J002_ODS_TO_DM_DIM_METEO_INFO**~~	**--to the top --** **GCP Remote Engine** **Industrial DO project**
	~~**Job description by steps**~~	**Job design**
	~~Metadata Calculation:~~ 1. ~~Job Metadata~~	

~~Calculation:~~

- ~~The job initiates by calculating essential job metadata. This metadata include information like job identifiers, timestamps, or job specific configuration details.~~
- ~~This step sets the foundation for the subsequent data processing.~~



~~Slowly Changing Dimension (SCD2) Data Loading:~~

1. ~~SCD2 Logic for Dimension Data Loading:~~
   - ~~In this step, the job implements Slowly Changin~~

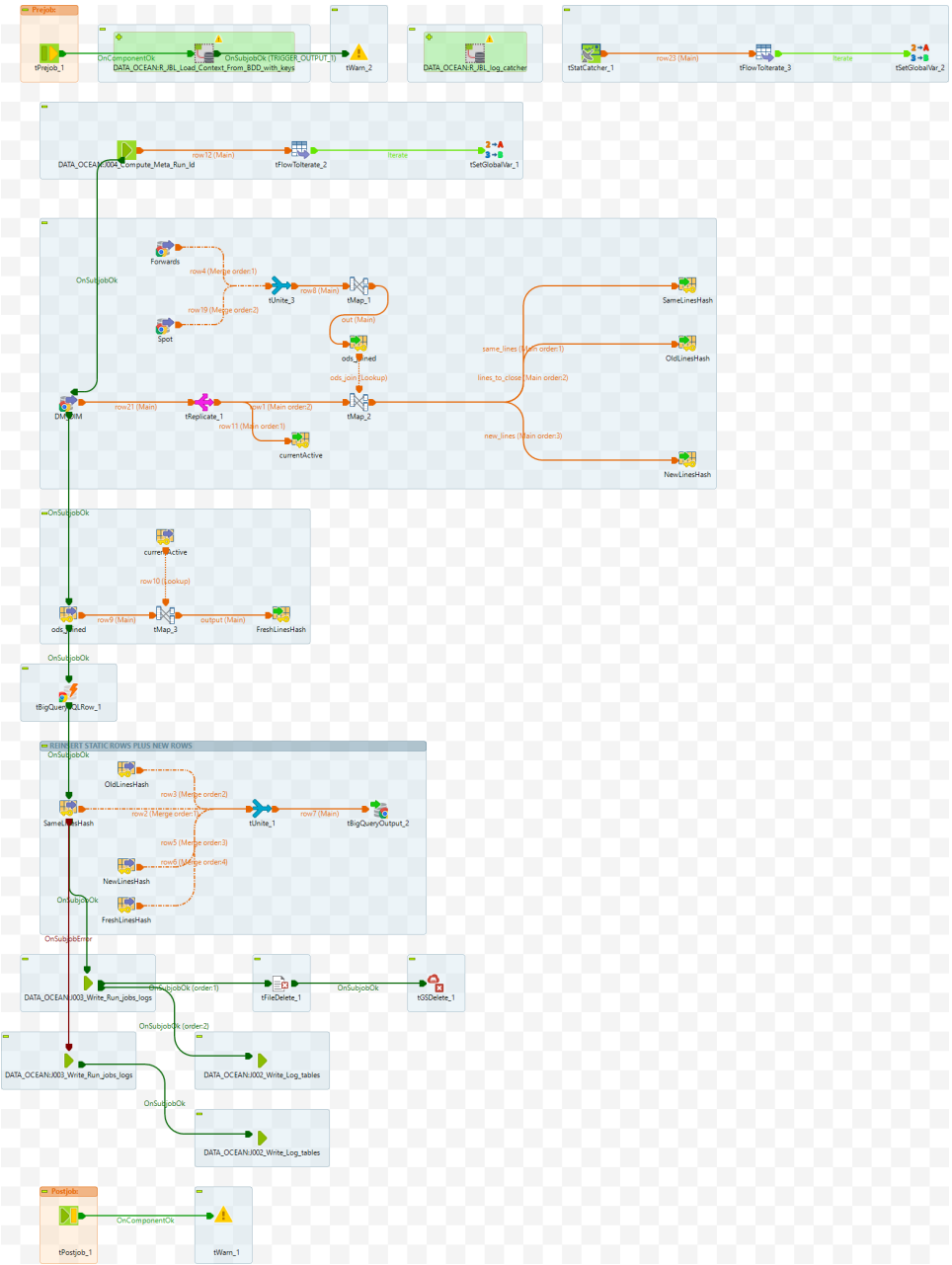
g Dimension Type 2 (SCD2) logic.
- The purpose of this logic is to load data into a dimension table, specifically from the Operational Data Store (ODS) meteo table into the Data Mart (DM) meteo table.
- The SCD 2 logic is applied by comparing multiple fields across the two tables, such as name, database name, lc grou

p, asset class, asset sub-class, asset type, capacity unit, commodity, curve publishing, curve type, hub, market, object type, owner, region, unit, and volume unit.

Logging and Cleanup:

1. Log Generation:
   - Upon completion of the data loading process, the job generates logs. These logs serve as a record of the job's activities, provi

	~~ding visibility into the data loading process.~~ 2. ~~Temporary File Deletion:~~ • ~~As a final step, the job takes care of deleting any temporary files created during the data transfer process. This cleanup ensures the efficient management of storage resources.~~ 3. ~~In case of error to save data in BQ, it the last data to save to BQ will save to /DATA/DEV/IND/ROBUSTIFY/InOut/DM/[meta_run_id].csv~~	
Main jobs for source	• ~~**J003_ODS_TO**~~	~~**- to the top -**~~ ~~**GCP Remote Engine**~~

extraction	~~_DM_DIM_ENERGY_PRICE_INFO~~	~~Industrial DO project~~
	~~Job description by steps~~	~~Job design~~
	~~Metadata Calculation:~~ 1. ~~Job Metadata Calculation:~~ • ~~The job initiates by calculating essential job metadata. This metadata include information like job identifiers, timestamps, or job specific configuration details.~~ • ~~This step sets the foundation for the subsequent data processing.~~ ~~Slowly Changing Dimension (SCD2) Data Loading:~~	

1. ~~SCD2 Logic for Dimension Data Loading:~~
    - ~~In this step, the job applies Slowly Changing Dimension Type 2 (SCD2) logic to load data into a dimension table.~~
    - ~~The source of the data is the Operational Data Store (ODS) energy price metadata table, which is a union of tables named "forwards" and "spot." The data is loaded into the~~

~~Data Mart (DM) energy price table using the SCD 2 methodology.~~

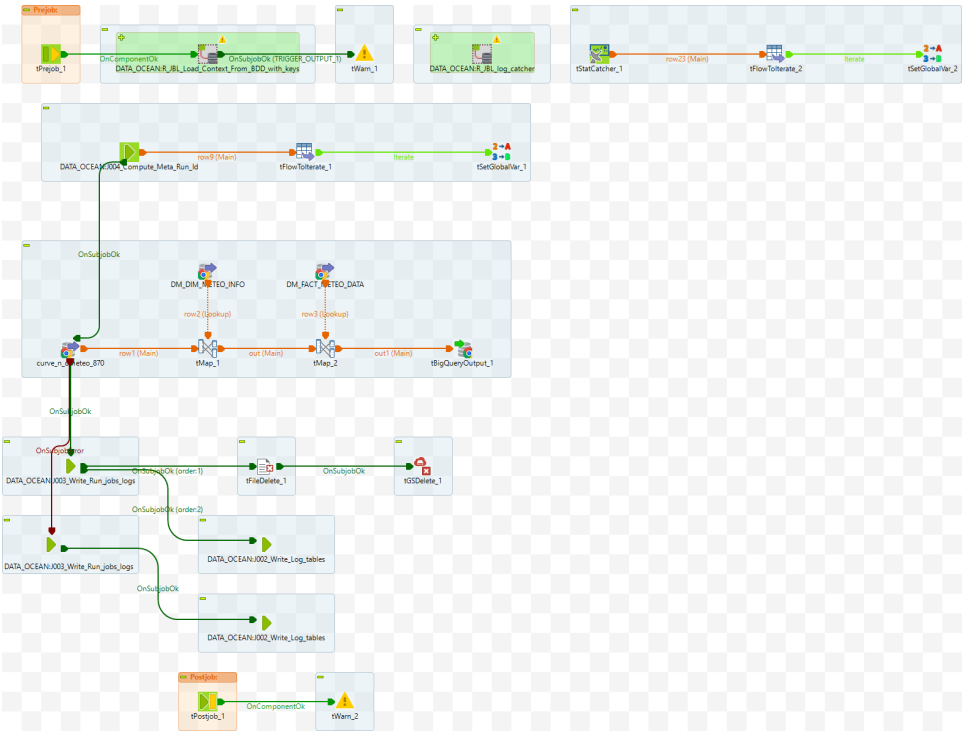
- ~~SCD 2 logic involves comparing specific fields, such as name, commodity, contract, proprietary, timespan, rolling, and timezone, to identify changes and maintain historical records in the DM table.~~

~~Logging and Cleanup:~~

1. ~~Log Generation:~~
   - ~~Upon the completion of the data loadi~~

ng process, the job generates logs. These logs serve as a record of the job's activities and provide visibility into the data loading process.

2. Temporary File Deletion:
   - As a final step, the job takes care of deleting any temporary files created during the data transfer process. This cleanup ensures the efficient management of storage

	~~resources.~~ 3. ~~In case of error to save data in BQ, it the last data to save to BQ will save to /DATA/DEV/IND/ROBUSTIFY/InOut/DM/[meta_run_id].csv~~	
Main jobs for source extraction	• ~~**J004_ODS_TO_DM_FACT_METEO_DATA**~~	~~**- to the top -**~~ ~~**GCP Remote Engine**~~ ~~**Industrial DO project**~~
	~~**Job description by steps**~~	~~**Job design**~~
	~~Metadata Calculation:~~ 1. ~~Job Metadata Calculation:~~ • ~~The job initiates by calculating essential job metadata. This metadata include information like job identifiers, timestamps, or job specific configuration~~	

details.
- This step establishes the foundation for the subsequent data processing.

Data Enrichment and Loading:

1. Data Enrichment from DIM_meteo_info:
   - In this step, the job enriches data from the Vendohm meteo table with attributes obtained from the DIM_meteo_info table.
2. Loading into FACT_meteo_data:
   - The enriched data is then loaded into the FACT_meteo_data table.

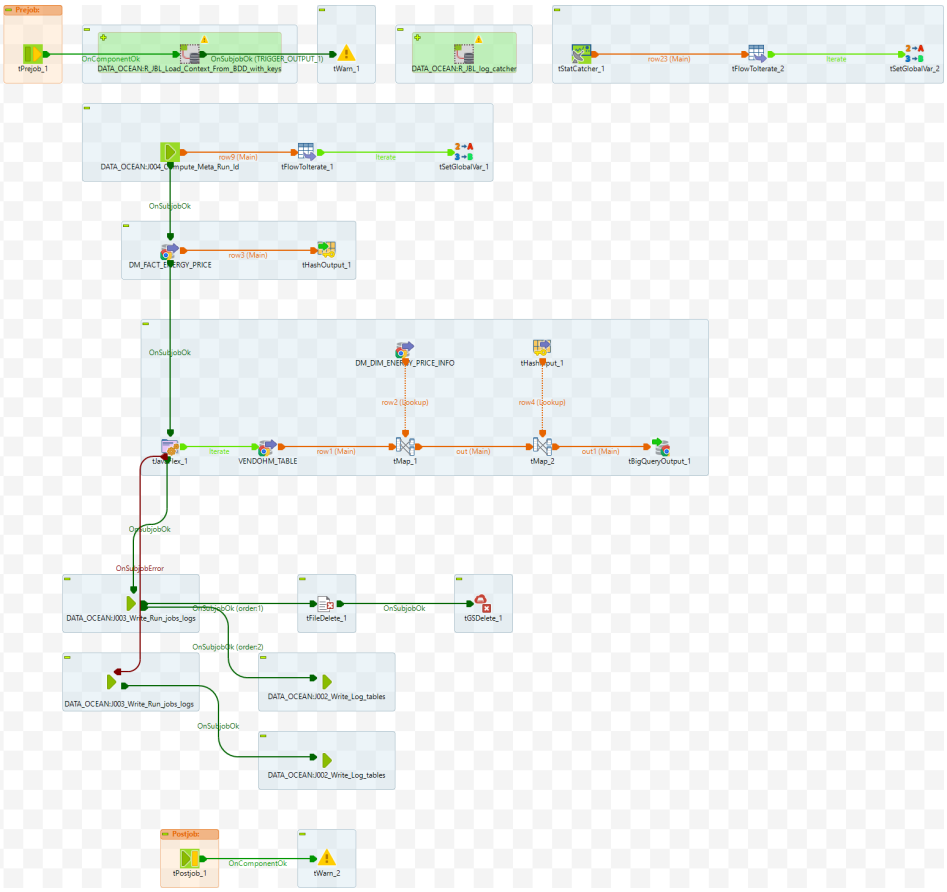
- ~~This table serves as a fact table containing data relevant to meteorological information.~~

~~Logging and Cleanup:~~

1. ~~Log Generation:~~
   - ~~Upon the completion of the data enrichment and loading process, the job generates logs. These logs serve as a record of the job's activities and provide visibility into the data processing steps.~~
2. ~~Temporary File Deletion:~~
   - ~~As a~~

	~~final step, the job takes care of deleting any temporary files created during the data transfor and enrichment process. This cleanup ensures efficient storage resource management.~~	
Main jobs for source extraction	• **~~J005_ODS_TO_DM_FACT_ENERGY_PRICE~~** **replace with view in Robustify prj-data-robustify-dev. DM. V_FACT_energy_price_hourly**	**--to the top --** **GCP Remote Engine** **Industrial DO project**
	Job description by steps	**Job design**
	~~Metadata Calculation:~~	

1. ~~Job Metadata Calculation:~~
   - ~~The job commences with the calculation of essential job metadata. This metadata includes information like job identifiers, timestamps, or specific configuration details.~~
   - ~~This initial step provides a foundation for subsequent data processing.~~



~~Data Enrichment and Loading:~~

1. ~~Data Enrichment from DIM_energy_price_info:~~
   - ~~In this step, the job enrie~~

hes data from the energy price table by incorporating attributes sourced from the DIM_energy_price_info table.

2. Loading into FACT_energy_price_data:
   - The enriched data is subsequently loaded into the FACT_energy_price_data table.
   - The FACT_energy_price_data table is designed to store data relevant to energy prices.

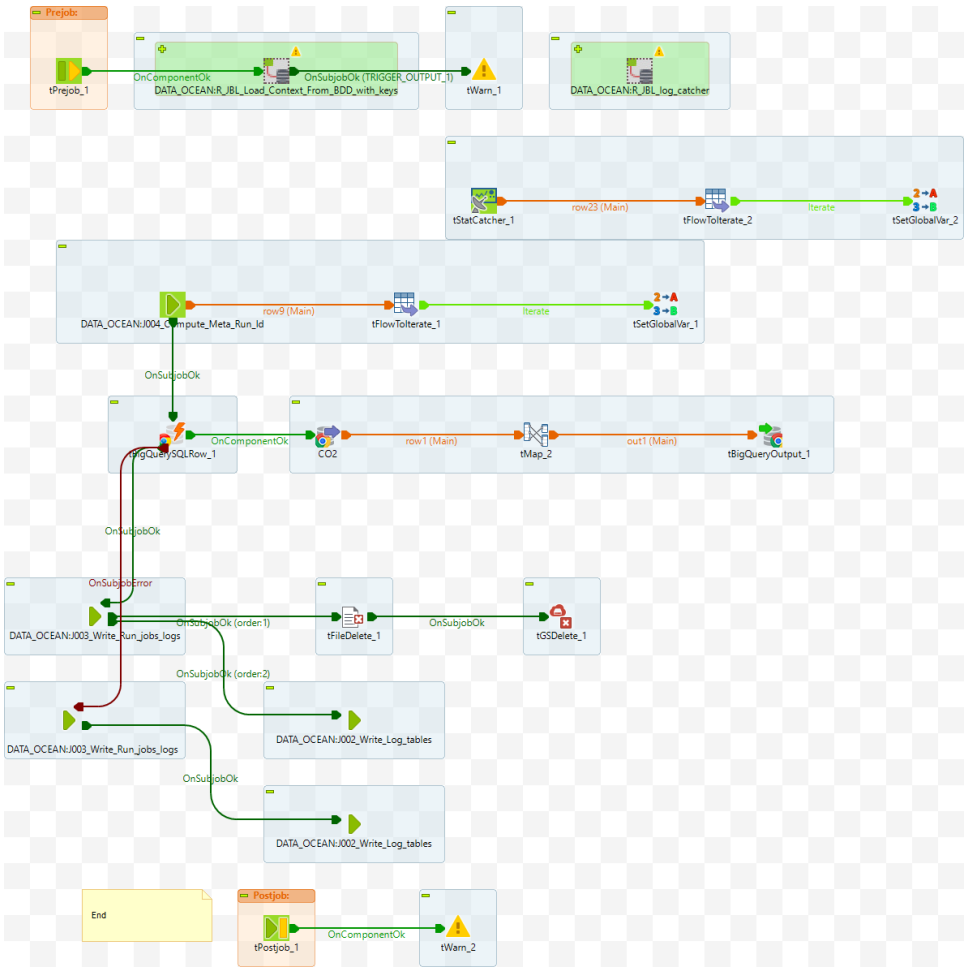
Logging and Cleanup:

1. Log Generation:
   - Following the com

pletion of the data enrichment and loading process, the job generates logs. These logs serve as a record of the job's activities and provide visibility into the data processing steps.

2. Temporary File Deletion:
    - As a final step, the job manages the deletion of any temporary files that were created during the data transfer and enrichment proc

	~~ess. This cleanup ensures efficient storage resource management.~~	

Main jobs for source extraction	• **J006_ODS_TO_DM_FACT_CO2_EMISSIONS**	**--to the top --** **GCP Remote Engine** **Industrial DO project**
	Job description by steps	**Job design**
	Metadata Calculation: 1. Job Metadata Calculation: • The job commences by calculating essential job metadata. This metadata includes information such as job identifiers, timestamps, or specific configuration details.	

- This initial step provides a foundation for subsequent data processing.

Latest CO2 Data Refresh:

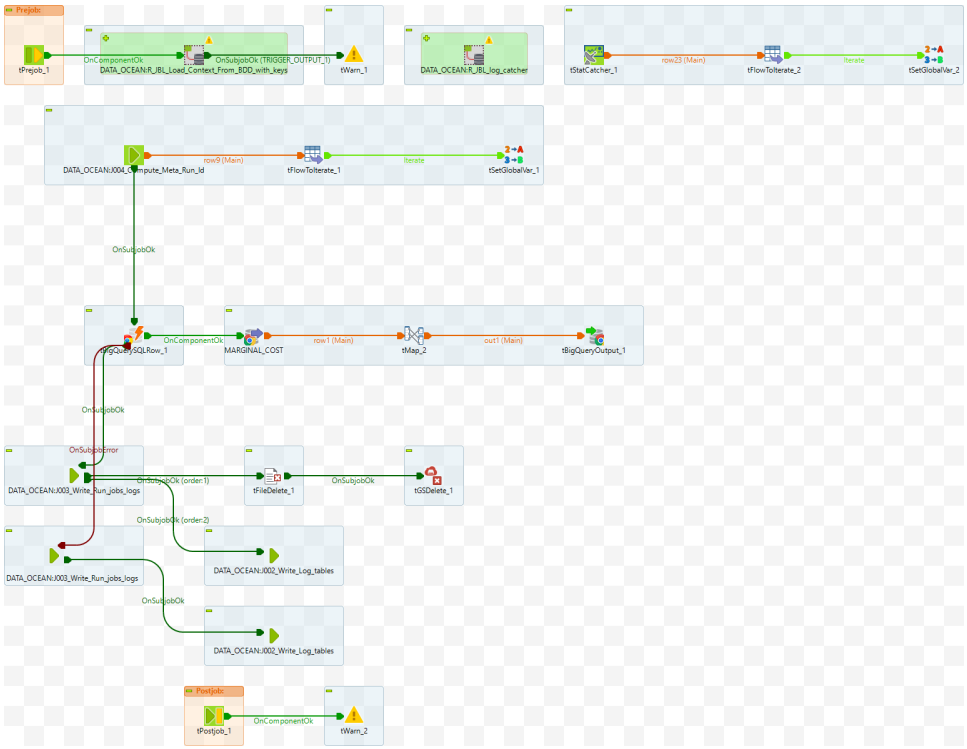
1. Refreshing /Providing Latest CO2 Data:
   - In this step, the job focuses on refreshing or providing the latest CO2 emissions data.
   - The source of this data is the Operational Data Store (ODS) table, which serves as a central repository for CO2-relat

ed information.

Logging and Cleanup:

1. Log Generation:
   - Following the completion of the CO2 data refresh, the job generates logs. These logs serve as a record of the job's activities and provide visibility into the data refresh process.
2. Temporary File Deletion
   - As a final step, the job manages the deletion of any temporary files that were created during the

	data transfer process. This cleanup ensures efficient storage resource management.	

Main jobs for source extraction	• **J007_ODS_TO_DM_FACT_SOLID_FUEL_WAP**	**--to the top --** **GCP Remote Engine** **Industrial DO project**
	Job description by steps	**Job design**
	Metadata Calculation: 1. Job Metadata Calculation: • The job commences by calculating essential job metadata. This metadata includes information such as job identifiers, timestamps, or specific confi	

guration details.
- This initial step provides a foundation for subsequent data processing.

Latest wap solid fuel data Refresh:

1. Refreshing /Providing Latest Cost Data:
   - In this step, the job focuses on refreshing or providing the latest weighted average fuel prices data.
   - The source of this data is the Operational Data Store (ODS) table, which serves as a

central repository for wap solid fuel - related information.

Logging and Cleanup:

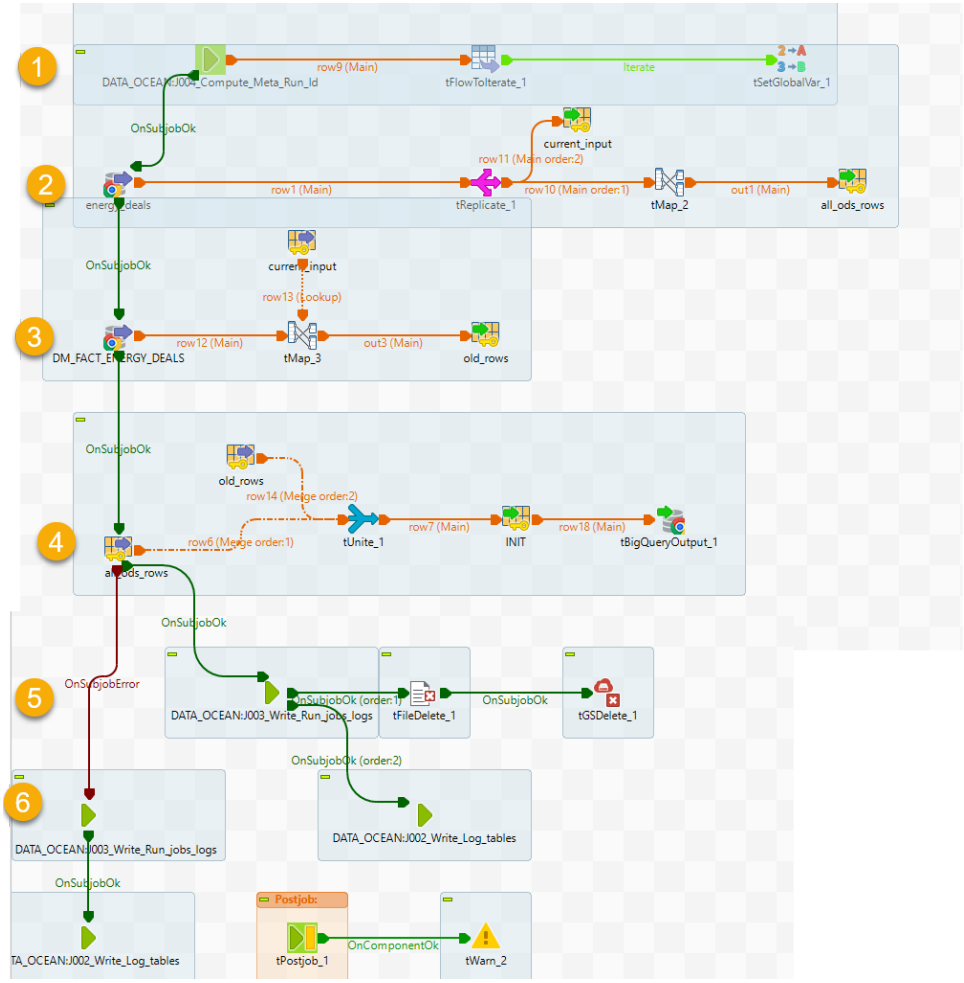
1. Log Generation:
   - Following the completion of the Wap solid fuel data refresh, the job generates logs. These logs serve as a record of the job's activities and provide visibility into the data refresh process.
2. Temporary File Deletion
   - As a final step, the job manages the deletion of any

	temporary files that were created during the data transfer process. This cleanup ensures efficient storage resource management.	

Main jobs for source extraction	• ~~**J009_ODS_TO_DM_FACT_ENERGY_DEALS**~~ **J009_ODS_TO_DM_FACT_IRM_DEALS** **(DATA_OCEAN_DOMAIN_SUSTAINABILITY)**	**--to the top --** **GCP Remote Engine** **Industrial DO project**
	Job description by steps	**Job design**
	1. Job Metadata Calculation: • The job initiates by calculating essential job metadata. This metadata	

includ
e
inform
ati
on
like
job
ident
ifiers
,
time
stam
ps,
or
job-
spec
ific
confi
gura
tion
detai
ls.
- This
step
esta
blish
es
the
foun
datio
n
for
the
subs
eque
nt
data
proc
essi
ng.

2. Get full load from ODS
3. Get data from DM. FACT_ir m_energy _deals_d aily in case there is some data, which not exist in ODS
4. Combine data from 2 and 3 to DM. FACT_ir m_energy _deals_d aily with truncate option
5. Log Generatio n:
   - Upo n the com pleti on of the data



enrichment and loading process, the job generates logs. These logs serve as a record of the job's activities and provide visibility into the data processing steps.

6. Temporary File Deletion:
    - As a final step, the job takes care of deleting any temporary files created during the data transfer and enrichment process. This cleanup ensures effici

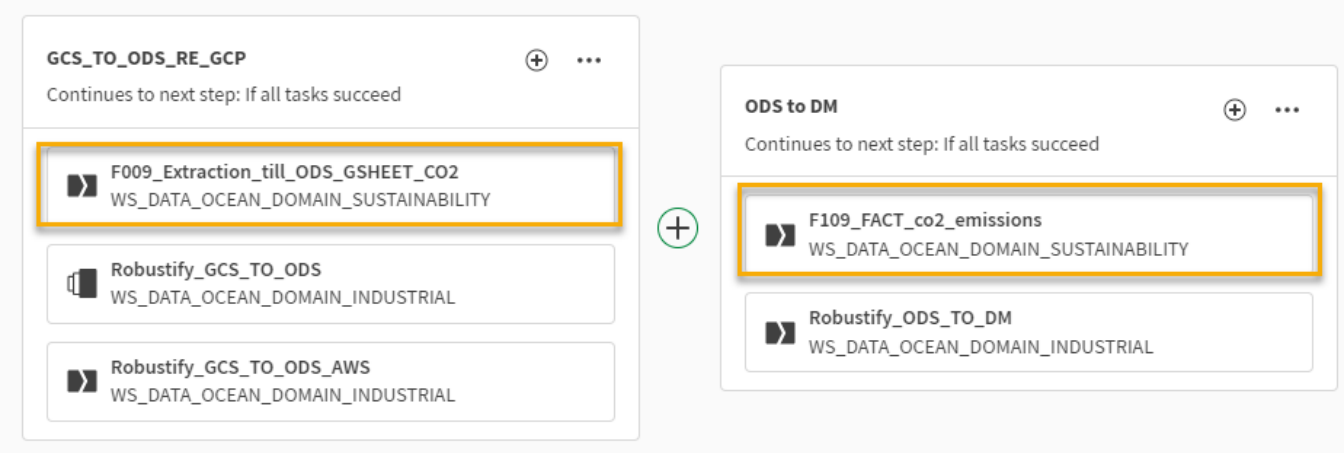
	ent storage resource management.

Main jobs for source extraction	• ~~**J001_DM_TO_DM_FACT_ENERGY_PRICE_HOURLY**~~ ~~**(ROBUSTIFY)**~~
	Job description by steps

	~~Data Extraction and Transformation:~~
	
	1. ~~The code extracts data related to energy prices and contracts from a database.~~
	   • ~~The source table is DataOcean.V_FACT_energy_irm~~
	   • ~~To get the raw data from IRM in ODS and transform to the format that Dataiku can consume~~
	      ○ ~~Split data into hourly~~
	      ○ ~~The past to current date will get from Spot~~
	      ○ ~~The current month will get from the first date of the next month~~
	      ○ ~~Next month data will be all the same within the month by the first date of each month~~
	
	~~Explanation the job:~~
	
	~~1. Setup global variable on output file / source table and define the value of gas = 0.0~~
	
	~~2. Get data from the source with format below~~
	
	delivery_date | elec_DE_EPEX | elec_FR_EPEX | elec_IT_PUN | elec_SP_OMEI | gas_DE_THE | gas_FR_PEG | gas_IT_PSV | gas_SP_PVB | gas_TTF
	2024-06-21 21:00:00 UTC | 105.26 | 105.14 | 135.3151 | 114.86 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0
	2024-06-21 22:00:00 UTC | 95.99 | 96.49 | 115.0116 | 115.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0
	2024-06-21 23:00:00 UTC | 82.14 | 83.26 | 112.0 | 105.74 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0
	2024-07-01 00:00:00 UTC | 73.15 | 50.94 | 106.9 | 74.56 | 34.5 | 34.31 | 36.15 | 34.45 | 34.48
	2024-08-01 00:00:00 UTC | 76.32 | 50.1 | 106.04 | 77.89 | 34.92 | 34.64 | 35.44 | 34.43 | 34.87
	
	~~3. Apply the business rule and save into the output file by tJavaFlex~~
	
	~~4. Transform to the Dataiku format~~
	
	date | elec_DE_EPEX | elec_FR_EPEX | elec_IT_PUN | elec_SP_OMEI | gas_DE_THE | gas_FR_PEG | gas_IT_PSV | gas_SP_PVB | gas_TTF | elec_IT_MGP | inserted_date
	2024-06-21 21:00:00 UTC | 105.26 | 105.14 | 135.3151 | 114.86 | 33.963 | 34.028 | 36.875 | 34.49 | 33.867 | null | 2024-06-21 10:09:55 UTC
	2024-06-21 22:00:00 UTC | 95.99 | 96.49 | 115.0116 | 115.0 | 33.963 | 34.028 | 36.875 | 34.49 | 33.867 | null | 2024-06-21 10:09:55 UTC
	2024-06-21 23:00:00 UTC | 82.14 | 83.26 | 112.0 | 105.74 | 33.963 | 34.028 | 36.875 | 34.49 | 33.867 | null | 2024-06-21 10:09:55 UTC
	2024-06-22 00:00:00 UTC | 73.15 | 50.94 | 106.9 | 74.56 | 34.5 | 34.31 | 36.15 | 34.45 | 34.48 | null | 2024-06-21 10:09:55 UTC
	2024-06-22 01:00:00 UTC | 73.15 | 50.94 | 106.9 | 74.56 | 34.5 | 34.31 | 36.15 | 34.45 | 34.48 | null | 2024-06-21 10:09:55 UTC
	2024-06-22 02:00:00 UTC | 73.15 | 50.94 | 106.9 | 74.56 | 34.5 | 34.31 | 36.15 | 34.45 | 34.48 | null | 2024-06-21 10:09:55 UTC
	2024-06-22 03:00:00 UTC | 73.15 | 50.94 | 106.9 | 74.56 | 34.5 | 34.31 | 36.15 | 34.45 | 34.48 | null | 2024-06-21 10:09:55 UTC
	2024-06-22 04:00:00 UTC | 73.15 | 50.94 | 106.9 | 74.56 | 34.5 | 34.31 | 36.15 | 34.45 | 34.48 | null | 2024-06-21 10:09:55 UTC
	2024-06-22 05:00:00 UTC | 73.15 | 50.94 | 106.9 | 74.56 | 34.5 | 34.31 | 36.15 | 34.45 | 34.48 | null | 2024-06-21 10:09:55 UTC
	
	~~5. Delete temporary file~~

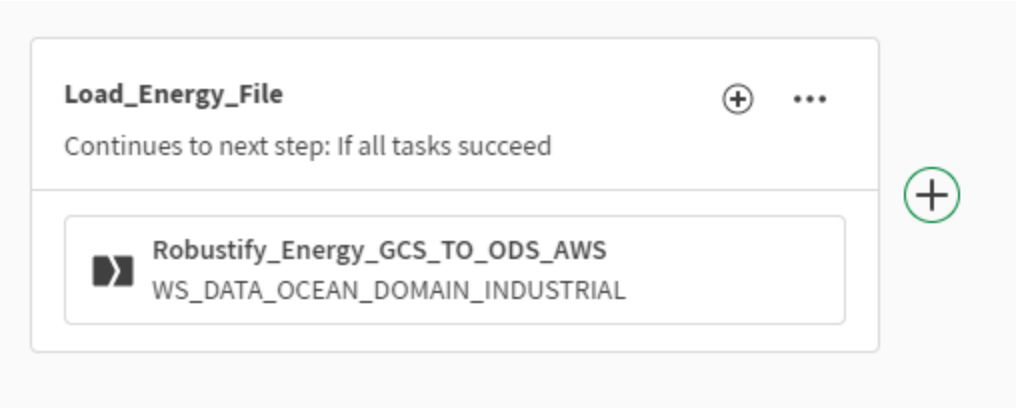
4.5 - Scheduling and Automation

Although there are 3 projects involve but main plan will be in TMC on WS_DATA_OCEAN_DOMAIN_INDUSTRIAL with plan PL_INDUS_ROBUSTIFY _LOAD load - Daily at 12:15 PM.



**Yellow box is the job in Sustainability domain.

PL_INDUS_ROBUSTIFY_ENERGY load hourly at xx:30



Monitor the loading in the log tables on prj-data-dm-industrial-[env] by

```
select job.job_name , job.meta_start_date , logs.meta_run_id , logs.meta_source_system , logs.meta_step , logs.meta_status , logs.meta_num_lines , logs.meta_error_lines from STG.log_tables logs join STG.run_jobs job on logs.meta_run_id = job.meta_run_id

where   logs.meta_run_id in ( SELECT meta_run_id FROM STG.run_jobs order by meta_start_date desc limit 1000 )

and  (

 job_name like '%METEOLOGICA%' or

 upper(job_name) like '%ENERGY%' or

 job_name like '%WAP%' or

 job_name like '%IRM%' )

and meta_start_date >   DATE_SUB ( CURRENT_TIMESTAMP () , INTERVAL 1 DAY )

order by job.meta_start_date desc
```

4.6 - Remark

1. Most of Talend jobs will be on project DATA_OCEAN_DOMAIN_INDUSTRIAL except:

 - Gsheet Co2 (J009_ODS_TO_DM_FACT_ENERGY_DEALS) will be in DATA_OCEAN_DOMAIN_SUSTAINABILITY prj-data-dm-sust-[env].STG.STG_FIL_0000_0000_F001_F_D_co2_emissions

 - DM to DM (J001_DM_TO_DM_FACT_ENERGY_PRICE_HOURLY) will be in ROBUSIFY project prj-data-robustify-[env].DM.FACT_energy_price_hourly

This is following data architecture

2. Most of Talend jobs are required to use remote engine on Cloud except J014_Extraction_till_ODS_ORACLE_IRM is required AWS. This is because of security reason