Data Quality Monitoring Tool - Technical Documentation

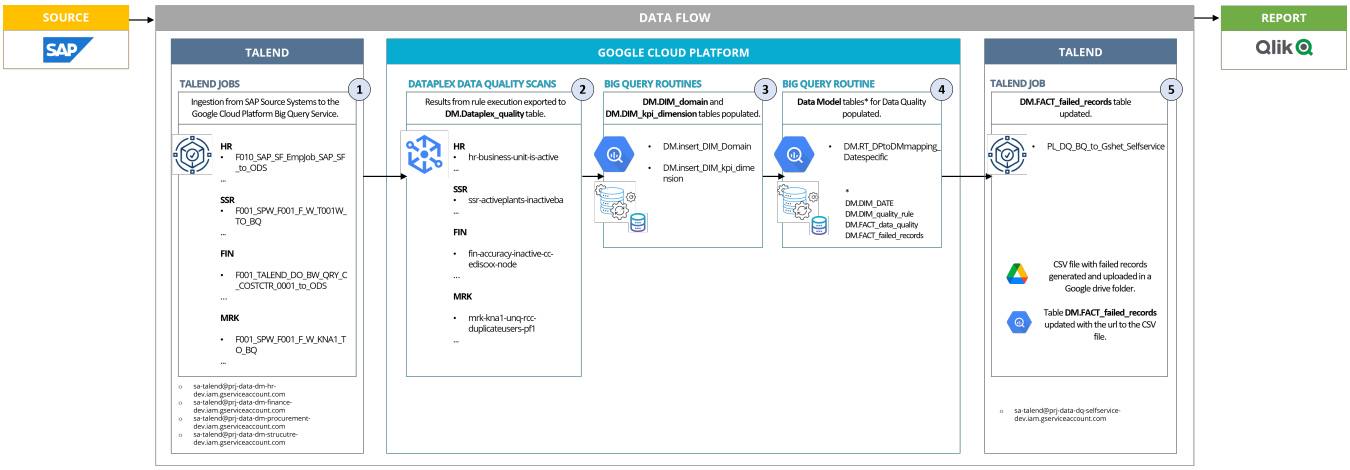
This document provides a **technical overview** of the **Data Quality Monitoring Tool**, detailing its architecture, data processing pipeline, monitoring mechanisms, and deployment strategy within **Google Cloud Platform (GCP)**.

The tool ensures data quality and integrity by leveraging **Talend for data ingestion**, **BigQuery for storage and processing**, and **Dataplex for rule execution**. The processed results are then visualized in **Qlik Sense** .

This documentation serves as a reference for understanding the system components, workflows, and operational best practices.

Architecture

Below is the high level architecture for the Data Quality KPIs monitoring tool.



1. Data Ingestion, Processing and Transformation

The process begins with the ingestion of data for each domain from various SAP source systems:

- HR data is sourced from SAP SuccessFactors
- SSR data is sourced from SAP WP1 and SAP PF1
- FIN data is sourced from SAP BW, WP1, and SAP PF1
- MRK data is sourced from SAP BW, WP1, and SAP PF1

Each table from each domain has its own dedicated Talend job responsible for ingesting and loading the data into the GCP BigQuery Data Ocean, specifically the following datasets:

- *prj-data-dm-hr-[env].ODS*
- *prj-data-dm-structure-[env].ODS*
- *prj-data-dm-finance-[env].ODS*
- *prj-data-dm-marketing-[env].ODS*
- *prj-data-dm-procurement-[env].ODS*

Also, views including only the necessary data are created in the following datasets:

- *prj-data-dm-hr-[env].DS_prj_dqkpi*
- *prj-data-dm-structure-[env].DS_prj_sls_dataquality_kpi*
- *prj-data-dm-finance-[env].DS_prj_sls_dataquality_kpi*
- *prj-data-dm-marketing-[env].DS_prj_sls_dataquality_kpi*
- *prj-data-dm-procurement-[env].DS_prj_sls_dataquality_kpi*

This views are the sole source for the for the quality checks performed by Dataplex.

2. Data Quality Execution in Dataplex

Once the views are created, the data quality rules are executed using GCP Dataplex Service and the validation results are stored in the following BigQuery table:

- *prj-data-dq-selfservice-[env].DM.Dataplex_quality*

3. Data Model Dimension Tables Population

After ingestion, two routines are executed to populate the Data Model (DM) Dimension tables:

- Routine *prj-data-dq-selfservice-[env].DM.insert_DIM_Domain* populates *prj-data-dq-selfservice-[env]. DM.DIM_domain* table

- Routine *prj-data-dq-selfservice-[env]. DM.insert_DIM_kpi_dimension* populates *prj-data-dq-selfservice-[env]. DM.DIM_kpi_dimension* table

4. Data Model Fact Tables Population

A routine is executed to populate the DM Fact tables:

- Routine *prj-data-dq-selfservice-[env].DM.RT_DPtoDMmapping_Datespecific* populates the following tables:
    - *prj-data-dq-selfservice-[env].DM.DIM_DATE*
    - *prj-data-dq-selfservice-[env].DM.DIM_quality_rule*
    - *prj-data-dq-selfservice-[env].DM.FACT_data_quality*
    - *prj-data-dq-selfservice-[env].DM.FACT_failed_records*

5. Failed Records Handling & Export

A final Talend job - *PL_DQ_BQ_to_Gshet_Selfservice* - handles failed records:

- Generates a CSV file with failed records.
- Uploads the CSV file to a Google Drive folder.
- Updates *prj-data-dq-selfservice-[env].DM.FACT_failed_records* with the URL to the CSV file, associated with the quality_rule_key.

[env] is one of the following: dev, test, ppd, prod

6. Visualization in Qlik Sense

The processed and validated data is available for visualization and analysis in Qlik Sense.

Data Model

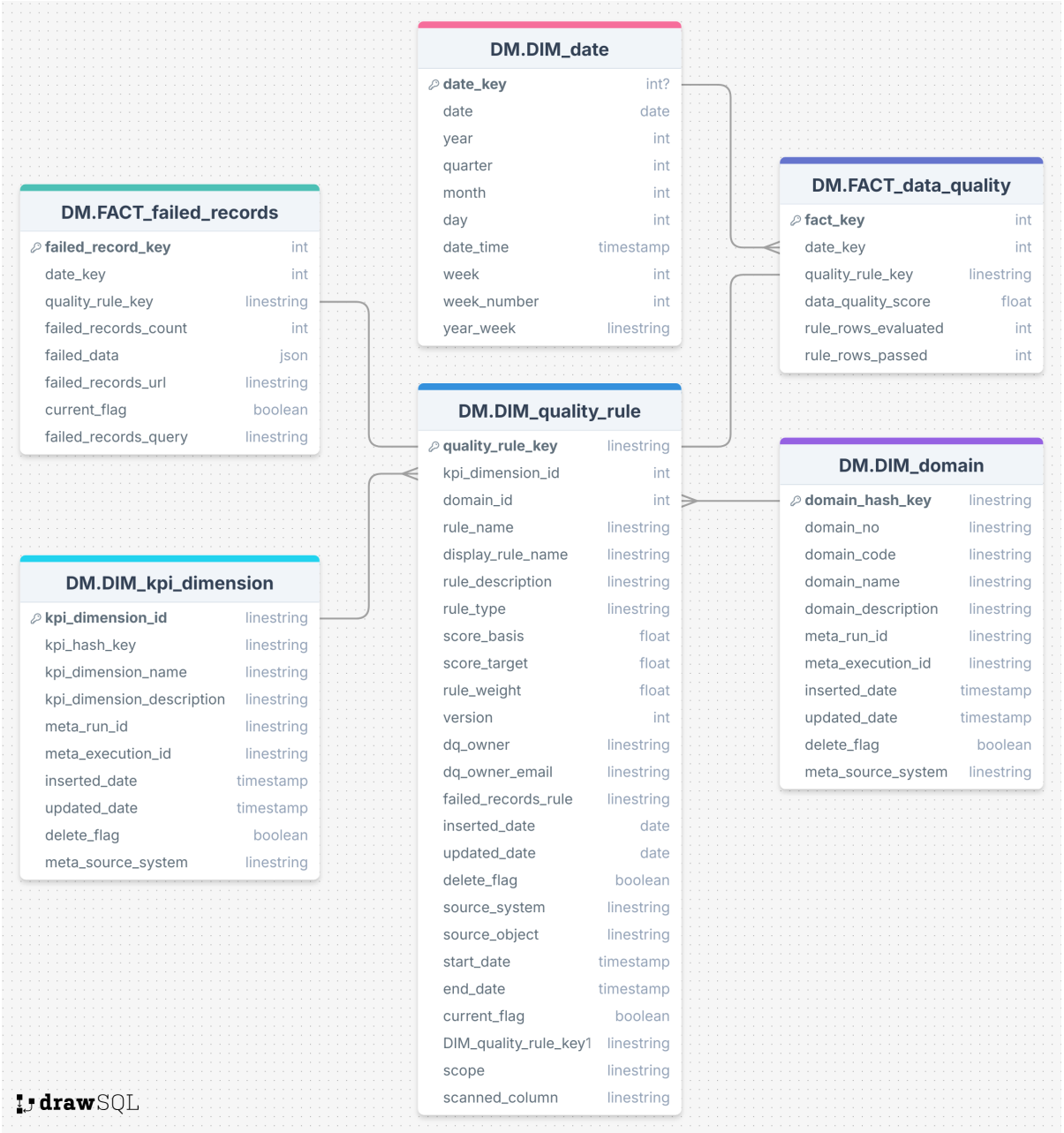
The **Data Model** consists of a set of structured tables within BigQuery that store processed and validated data. These tables are designed to support efficient querying, data quality monitoring, and reporting.

The following tables are part of the Data Model schema:

- *prj-data-dq-selfservice-[env].DM.DIM_DATE*
- *prj-data-dq-selfservice-[env].DM.DIM_quality_rule*
- *prj-data-dq-selfservice-[env].DM.FACT_data_quality*
- *prj-data-dq-selfservice-[env].DM.FACT_failed_records*

[env] is one of the following: dev, test, ppd, prod

Each table plays a crucial role in storing metadata, data quality rules, validation results, and failed records for further analysis, you can find the schema showing the relationships between the DM tables below:



Scheduling

The following process are scheduled on a weekly basis.

1. Talend Ingestion Jobs

The Ingestion Jobs are scheduled to run within Talend every week, at the beginning of the process.

2. Data Quality Scans

Initially "On Demand" for testing purposes and then "Scheduled" to run every week within Dataplex.

3. Routines Execution - Data Model Dimension Tables

The 2 routines are triggered using scheduled queries on a weekly basis within BigQuery.

- *prj-data-dq-selfservice-[env].DM.insert_DIM_Domain*
- *prj-data-dq-selfservice-[env].DM.insert_DIM_kpi_dimension*

[env] is one of the following: dev, test, ppd, prod

4. Routines Execution - Data Model Fact Tables

The routine is triggered using scheduled queries on a weekly basis within BigQuery.

- *prj-data-dq-selfservice-[env].DM.RT_DPtoDMmapping_Datespecific*

[env] is one of the following: dev, test, ppd, prod

5. Talend Report Job

The Talend Job *PL_DQ_BQ_to_Gshet_Selfservice* is scheduled to run within Talend every week, at the end of the process.

6. QlikSense Refresh

The QlikSense refresh schedule is set by the Visualization Engineer within QlikSense.

Process Scheduling Details

Bellow you can find a table that summarizes the processes, their frequency, duration window and average duration.

Process	Frequency	Duration Window	Average Duration (min)
Talend Ingestion Jobs	Every Sunday	21:00 CET	
Dataplex Data Quality Scans	Every Monday	4:00 - 5:00 CET	1
BigQuery Routine *insert_DIM_Domain*	Every Monday	5:00 - 5:30 CET	1
BigQuery Routine *insert_DIM_kpi_dimension*	Every Monday	5:30 - 6:00 CET	1
BigQuery Routine *RT_DPtoDMmapping_Datespecific*	Every Monday	6:00 - 6:30 CET	1
Talend Report Job *PL_DQ_BQ_to_Gshet_Selfservice*	Every Monday	6:30 - 7:00 CET	5
QlikSense	Every Monday	7:30 CET	1

Error Handling

To maintain the reliability of the data quality pipeline, a structured error handling procedure is in place for each scheduled process.

In the event of a failure, it's crucial not only to resolve and rerun the failed step, but also to re-execute all subsequent steps in the pipeline — as they may have run on incomplete or outdated data.

For a full overview of how the data and processes flow together, please refer to the Architecture & Data Flow Diagram.

1. Talend Ingestion Jobs

What to check:

- Verify the Talend execution logs to identify the root cause.
- Confirm SAP source system connectivity.
- Check for schema changes in source systems that may have caused mapping errors.
- If a prior process failed, ensure all upstream steps have been rerun.

Next steps:

- Rerun the failed Talend job manually after resolving the issue.
- Re-execute all downstream processes: Data Quality Scans, Routines, Report Job, and QlikSense Refresh.
- Inform the Data Engineer in case further support is needed.

2. Data Quality Scans

What to check:

- Access Dataplex logs to locate the failed rule or asset.
- Ensure that the source views in BigQuery are available and not empty.
- Confirm rule syntax and metadata configurations.
- If a prior process failed, ensure all upstream steps have been rerun.

Next steps:

- Re-execute the failed scan via the Dataplex UI or using a scheduled query.
- Re-run subsequent routines and the Talend Report Job to reflect updated quality results.
- If multiple rules fail, check if a shared dependency is broken.
- Contact the Data Architect for review if the failure is rule-related.

3. Routines Execution - Data Model Dimension Tables

What to check:

- Review scheduled query logs in BigQuery for error messages.
- Validate that the input tables contain data for the current cycle.
- If a prior process failed, ensure all upstream steps have been rerun.

Next steps:

- Rerun the failed query manually.
- Re-execute the Talend Report Job and QlikSense Refresh to align downstream outputs.
- Fix any reference issues or update logic if the schema has changed.
- Escalate to the Data Engineer if the issue persists.

4. Routines Execution - Data Model Fact Tables

What to check:

- Review scheduled query logs in BigQuery for error messages.
- Validate that the input tables contain data for the current cycle.
- If a prior process failed, ensure all upstream steps have been rerun.

Next steps:

- Rerun the failed query manually.
- Re-execute the Talend Report Job and QlikSense Refresh to align downstream outputs.
- Fix any reference issues or update logic if the schema has changed.
- Escalate to the Data Engineer if the issue persists.

5. Talend Report Job

What to check:

- Review Talend logs to determine if the issue was during query execution, file generation, or upload to Google Drive.
- Confirm the existence and access permissions of the target Google Drive folder.
- If a prior process failed, ensure all upstream steps have been rerun.

Next steps:

- Manually generate and upload the failed records file if needed.
- Update the DM.FACT_failed_records table with the file URL manually if automation fails.
- Ensure the DM.FACT_failed_records table is updated with the correct file URL.
- Manually trigger the QlikSense Refresh afterward.
- Coordinate with the Talend support team.

6. QlikSense Refresh

What to check:

- Check QlikSense dashboard status and refresh logs.
- Ensure the data sources (BigQuery tables) are accessible.
- If a prior process failed, ensure all upstream steps have been rerun.

Next steps:

- Notify the Visualization Engineer to manually trigger the refresh.
- In case of missing data, trace the issue upstream (Talend, BigQuery, or Dataplex).

Monitoring

To ensure the smooth operation of the data pipeline, monitoring is implemented using **Google Cloud Platform monitoring tools**.

These tools help track system performance, identify issues, and ensure data integrity throughout the process. The key monitoring tools used are:

- **Dataplex Logs** : Provides insights into data quality execution and rule application within Dataplex.
- **BigQuery Logs** : Tracks data ingestion, query execution, and table updates in BigQuery.
- **Cloud Monitoring Dashboard** : Offers a centralized view of system performance, resource utilization, and potential errors.

Environments and Deployment

The data processing pipeline is deployed across four different **Google Cloud Platform environments** to ensure a structured and controlled rollout:

- **Development (*dev*)** : Used for initial development and testing.
- **Testing (*test*)** : Serves as a staging environment for QA and validation.
- **Pre-production (*ppd*)** : A near-production environment for final validation before release.
- **Production (*prod*)** : The live environment where data processing occurs in real-time.

The deployment process involves migrating updates between these environments and is managed by the **DataOps Team** in collaboration with **D ata Engineers** .

As of now, **four deployments** have been completed. Detailed documentation related to these deployments is available in the following **Google Drive folder** :

Known Bugs

Currently, no bugs have been identified in the system.

Roadmap

FSD

TSD