# DQ OBSERVATIONS

* For any time-aggregated reporting we need to define a time-based column per functional area, e.g. “claim\_closing\_date” for claims
* Aggregated KPIs to be defined
* From the plausibility checks, need to specify which ones would make a file submission invalid, e.g. if a column has null values we either don’t ingest the file, skip the record or just report?
* Claims potentially use different currencies, perhaps we should have KPIs in EUR, for which would need to convert the claim amounts to EUR, at end of month rate?
* Ideally a three step approach, but practically a two step (pre-ingestion (OE) + post on DB or pre-ingestion (OE) + file check only) might be a feasible start:
* Pre-ingestion (scripts ran on OE side) (file validity, CIM structure, …)
* Pre-ingestion (GDP) In-Stream or over the file (number format, …)
* Post-ingestion (GDP on DB) Business checks (GWP > 0, open after closed, …)
* Do we need and have reference data in the DB for DQ?
* For the conciliation we can load the provided aggregated csv files into tables and have a side-by-side view with the aggregated KPIs, or just one view showing the differences where the difference is more than 1% or similar
* Batch processing with Orchestration (attached Argo demo): for a proper DQ task orchestration we need a task manager available (e.g. Argo, ADF, Databricks), in Midcorp we used Argo Workflows as it is a Kubernetes native app and to use it didn’t have to go through all the usual Allianz’ hoops