RBNDT DATAFLOW

Overview

RBNDT data platform workflow Concept

The Proof of Concept consists of an automated pipeline that ingests health screening data from SharePoint, validates it, stores it cleanly in Azure, and presents it through Power BI dashboards, or any other means. When something goes wrong with a file, the right people get a Microsoft Teams notification within seconds and can act on it. It runs on free or near-zero-cost infrastructure outside of compute consumption.

How a single file moves through the system

A data steward exports screening data from RBA into an Excel file (the `RBA_EXPORT_SCREENING_END_<period>.xlsx` format) and uploads it to a specific SharePoint folder: `Data Sheets/HSDS/Tapologo Touch`. They don't trigger anything — they just save it.

Your Azure Data Factory pipeline `pl_sharepoint_to_lake` polls that SharePoint folder using Microsoft Graph API. It authenticates as the OPMO app registration in Entra ID, with the client secret stored in `kv-opmo` (Key Vault). It lists files in the folder, filters down to ones modified since the last run, and iterates through them.

For each file, three things happen inside the ForEach:

1. **DownloadRawFile** streams the Excel bytes from SharePoint into the raw zone of `storagerbndt` (Azure Data Lake Gen2), preserving the original as audit. Filename gets a `yyyy-MM-dd_` prefix so each run produces a dated copy.
2. **df_clean_and_rename** is a Spark-based ADF Data Flow that opens the Excel, validates that all 87 expected `Pat_` prefixed columns are present using Assert rules (one rule per column, fail-fast on first missing), and if validation passes, writes the result as a Parquet file to the clean zone. The "Clear the folder" sink option ensures each run overwrites the previous snapshot rather than accumulating.
3. **If anything fails** in the data flow, the red failure arrow triggers `Notify_DataFlow_Failure` — an Execute Pipeline call that invokes your separate `pl_notify_teams` pipeline.

How notifications work

`pl_notify_teams` is reusable across the whole platform. It does two things: reads the Teams webhook URL from `kv-opmo` (using ADF's Managed Identity), then HTTPS-POSTs a JSON payload to a Power Automate Workflow.

That Workflow receives the JSON, renders it into an Adaptive Card with five fact rows (Dataset, File, Source folder, Submitter, Pipeline run) and a Details section, and posts the card into the `rbndt-data-ops` Microsoft Teams channel. The data steward sees the failure, opens SharePoint via the Source folder link, fixes the file, and re-uploads it. The next pipeline run picks it up automatically.

How the data gets consumed

Once Parquet lands in the clean zone, the Synapse serverless SQL view `dbo.vw_health_screening` exposes it. The view uses `OPENROWSET BULK` against the clean zone path, applies type casting (`TRY_CAST` for nullable conversions, `CAST` for required ones), and normalises one piece of data quality logic — Phase 1/2 suburbs get rewritten to "Freedom Park". Column names match the source `Pat_` prefixed names rather than analyst-friendly aliases, so the same name appears in Excel, Parquet, Synapse, and Power BI throughout the stack.

Power BI reads from the Synapse view. The Tapologo report (6 pages, 87 visuals, custom DAX) was originally built with the v1 aliases and is currently being migrated to reference the new column names. Analysts open the report, refresh, see the latest screening data.