DATA MIGRATION FOR THE STEP-BY-STEP MODERNISATION

OF ORGANICALLY GROWN SYSTEM LANDSCAPES

Oracle Forms is one of the pioneering technologies of rapid application development. In many companies it is a core part of data processing and inseparable from how data is stored.

That deep integration has often turned into a major strategic hurdle – the Java applet technology Forms relied on for years, and which is often still in use, is outdated, and experts who can develop Oracle Forms are increasingly hard to find. On top of that, business logic frequently lives partly in Forms code and partly in various stored procedures called by triggers or jobs. This makes a replacement considerably more complex. A migration in stages is the sensible path – and we support it in several areas.

CHALLENGE

Metro Digital engaged us for the data migration of exactly this kind of project. Legacy and target systems had to run in parallel during a coexistence phase to avoid a risky big-bang switchover.

To achieve this, we synchronised two complex Oracle landscapes bidirectionally and supported the migration from Teradata to BigQuery on Google Cloud. Requirements emerged alongside our solution and changed constantly.

The transfer had to be correct and verifiable for the financial audit. Because the two systems modelled processes differently, data quality had to satisfy the rules and validation of both. Records needing correction were detected automatically so that data enrichment could run in parallel with the migration.

APPROACH

We built an ETL platform to implement and orchestrate migration pipelines quickly and to isolate errors for analysis.

The foundation was Apache Spark with Java in a serverless Google Cloud environment. The original abstractions stayed almost unchanged while the pipelines grew with the requirements: new validation rules, cross-system lookups and changed mappings. More than 400 versions of the mapping rules were rolled out via CI/CD with minimal effort.

A quarantine system tagged faulty records with descriptive messages for targeted fixing. Apache Airflow orchestrated the independent pipelines, and extras such as email triggers were pure configuration.

IMPLEMENTATION

1 CLOUD INFRA-STRUCTURE

Using infrastructure as code, we set up a spoke project in Google Cloud Platform on the client's infrastructure, including network connectivity to the OCI systems (Oracle Cloud Infrastructure) and encryption in transit and at rest. This ensured data security and compliance.

2 SCALING ON DEMAND

With serverless Apache Spark on Google Cloud Platform, individual workloads scaled automatically. Large volumes from a data warehouse, for example, could be migrated much faster in a bigger cluster. Since we didn't have to run any servers of our own, operating costs in this project dropped considerably.

METRO digital × POSEDIO

Do it RIGHT.

3 MIGRATION CONCEPT

In line with the ETL process, we defined two environments: the landing zone as the result of extraction and the curated zone as the result of transformation, including validation. Depending on data volume, we implemented them in AlloyDB or as Parquet files in Google Cloud Storage to keep costs under control.

4 FLEXIBLE ARCHITECTURE

A well-designed Java architecture following composition over inheritance, and a test concept using Testcontainers for real Spark integration tests, carried the ETL platform. Whether CSV, Teradata, Oracle or static data – it was just a matter of configuration.

5 BUILDING THE PIPELINES

Through GitHub CI/CD, domain experts could submit mapping rules as SQL files that were loaded straight into the platform. Automated release management ensured that every pipeline run was pinned to a specific version of the mapping rules. A few parameters were usually enough to deploy a new, working pipeline.

6 ERROR ANALYSIS

The descriptive messages from the quarantine system exposed contradictions and gaps in the migration concepts. We resolved them together with the client's domain experts.

7 OBSERVABILITY & LOGGING

On failure, Airflow sent notifications to Microsoft Teams, so we spotted pipeline problems early and could react quickly. Searchable log files in Parquet format showed the migration history of every record and led to the root cause of a problem much faster.

RESULTS

CENTRAL COMPONENT

Our ETL platform is the heart of a large migration project. Beyond the Oracle migration, it versions the mapping rules and serves as evidence for the financial audit.

SAFE COEXISTENCE

During the coexistence phase, the platform orchestrates the regular data reconciliation between legacy and target systems until every Metro store in Austria is connected to the target system.

FLEXIBILITY

Flexibility was one of the key design goals from early on. This let us support the domain experts technically and develop new requirements together, often taking them to production, tests included, in under an hour.

EFFICIENCY GAIN

The bottom line is a clear efficiency gain: the platform and its reusable ETL building blocks form the basis for around 300 ETL pipelines and a repeatable migration process across all Austrian stores.

This project showed once again that there is no one-size-fits-all solution. Understanding client requirements and building them into system and application design early saves cost and effort later. Complex projects demand agility – and our ETL platform had us well equipped for it.

MORE INFO

www.posedio.com
sales@posedio.com
+43 660 915 44 19

METRO digital × POSEDIO

Do it RIGHT.