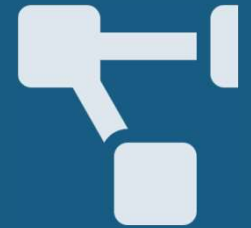


DMA UK – Webinar
Pally Singh – Accu Analytics
14th August 2026

Data Quality Governance in Complex Transformations: A Dual- Project Case Study

Why the two programs collide, what shared architecture should look like, and a real case study of what it costs to get governance wrong.

- 01 The Situation**
- 02 The Foundational Architecture**
- 03 Case Study**
- 04 Lessons & Recommendations**



01 • THE SITUATION

Two programs, one dataset, competing timelines

Organizations rarely get to modernize sequentially. A legacy platform is being retired at the same time a new warehouse needs to go live — so the same source tables get extracted, cleansed, remodeled, and loaded twice over, under two different plans, by two different teams.

These two programmes share significant common ground (source systems, business rules, reference data, data quality issues, and the Conceptual/Logical Data Models) while remaining distinct in purpose, technology and ownership.

Core interdependency

The cleansed, conformed migration dataset is frequently the primary (or sole) high-quality input into the new Data Warehouse. This creates a tight coupling: delays, quality shortfalls or reference-data changes in the migration stream directly impact the DW stream, and vice versa.

By the time anyone notices the two plans are working against each other, both are already mid-flight.



Data Migration

Move it once, correctly. Success = source and target reconcile, nothing is lost, lineage is provable.



Data Warehousing

Move it once, and reshape it for analysis. Success = trusted, query-ready, conformed to a shared model.

01 · THE SITUATION

Four challenges of running both at once



01

Competing Timelines

Migration cutover dates and warehouse release dates rarely align, so one program is always waiting on the other's extract.



02

Data Quality Debt, Doubled

Every defect in the source gets discovered twice — once during migration validation, once during warehouse load.



03

Reference Data Drift

Code sets and master values get remapped independently by each workstream, so the same value can have slightly different meaning



04

Governance Ambiguity, Duplicated Effort

No one workstream owns the data rules, data etc, so decisions on formats, keys, and definitions get made twice — inconsistently. Creating a great overhead on governance.

01 · THE SITUATION

What it costs to skip shared architecture

These aren't hypothetical — the case study ahead shows this exact pattern inside a real organization.



Rework

The same defect gets cleansed twice, in two different toolsets, by two different teams — instead of once, at the source.



Schedule slippage

Conflicts a shared design would have caught early instead surface at cutover or go-live, when they're most expensive to fix.



Budget overrun

Without a shared plan, remediation work is repeated rather than reused — inflating cost on both sides of the program.



Eroded trust

When migration and warehousing disagree on what a value means, business users stop trusting the numbers — and in regulated settings, that becomes a compliance exposure.

02 · THE FOUNDATIONAL ARCHITECTURE

Data Components both workstreams must share

Reference data, data quality rules, and data models can't be owned separately by migration and warehousing — they have to be built once and consumed by both.



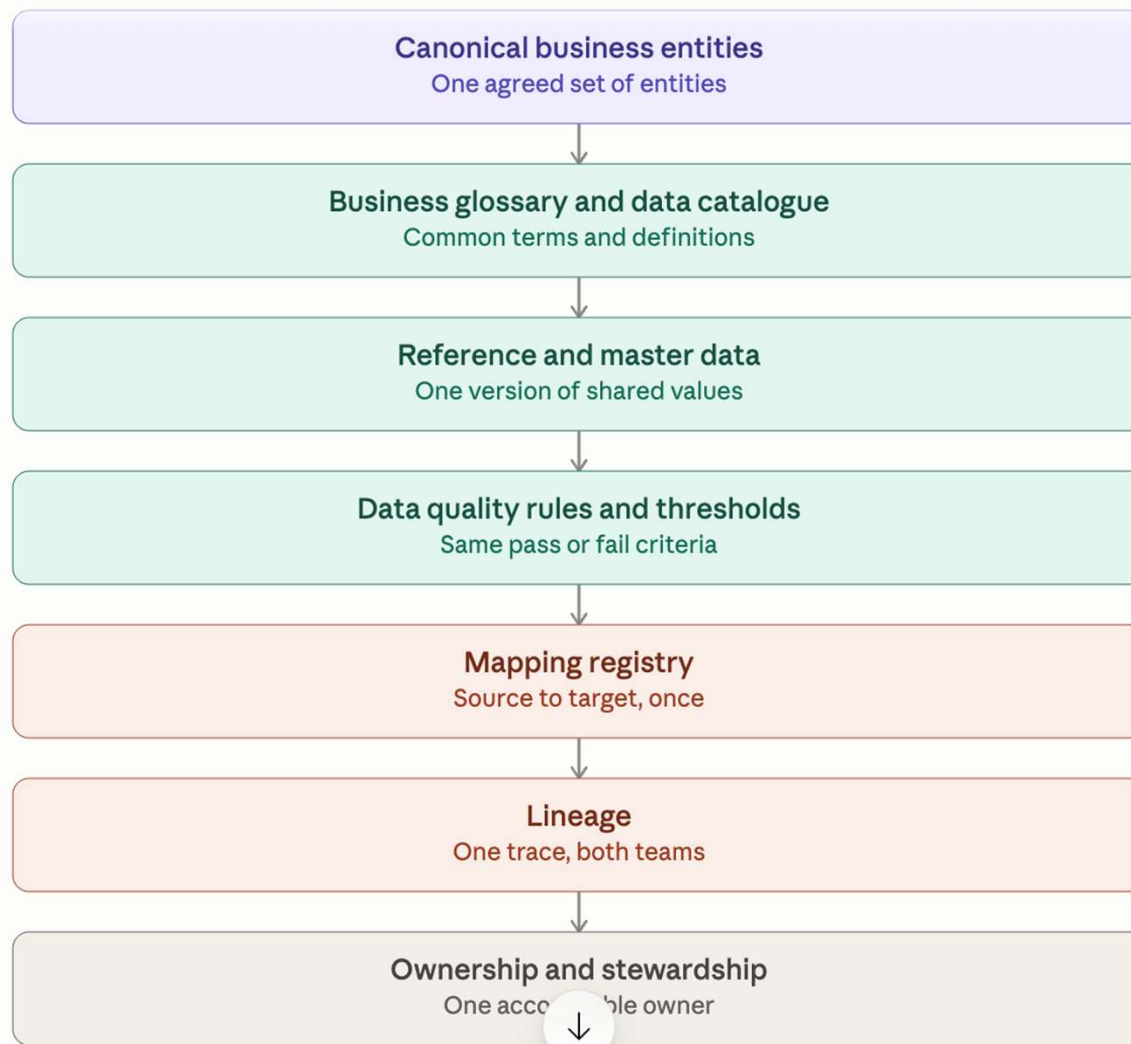
Reference
Data



Data
Quality



Data
Models



Define (purple) → govern (teal) → trace (coral) → own (gray)

Canonical business entities – the single agreed definition of core entities (customer, product, account) that both teams build against, so nobody re-derives their own version.

Business glossary and data catalogue – one place where terms mean the same thing to everyone, so "active customer" isn't defined three different ways.

Reference and master data – one authoritative set of shared values (country codes, product hierarchies, org units) both teams pull from rather than maintaining parallel copies.

Data quality rules and thresholds – one set of pass/fail criteria, so a record considered "clean" for migration is also considered clean for the warehouse.

Mapping registry – the single record of how legacy fields map to canonical entities, used by migration ETL and DW ingestion alike.

Lineage – one trace of how data moved and transformed, so both teams can answer "where did this number come from" the same way.

Ownership and stewardship – one accountable owner per shared element, so a rule change gets signed off once, not twice.

Shared Architecture

>> Migration >>

>> Data Warehouse >>

RDM / MDM

DQ

Models

Tooling

Catalogue

Governance (Processes)

Ownership (People)

02 · FOUNDATIONAL ARCHITECTURE

Reference data: the common anchor

Master, transactional, and reference data all have to move to the new platform — and reference data (code sets, hierarchies, lookup values) is what both migration and warehousing decode against. If it isn't governed as a single shared asset, each project builds its own version.

- **Single source of truth**

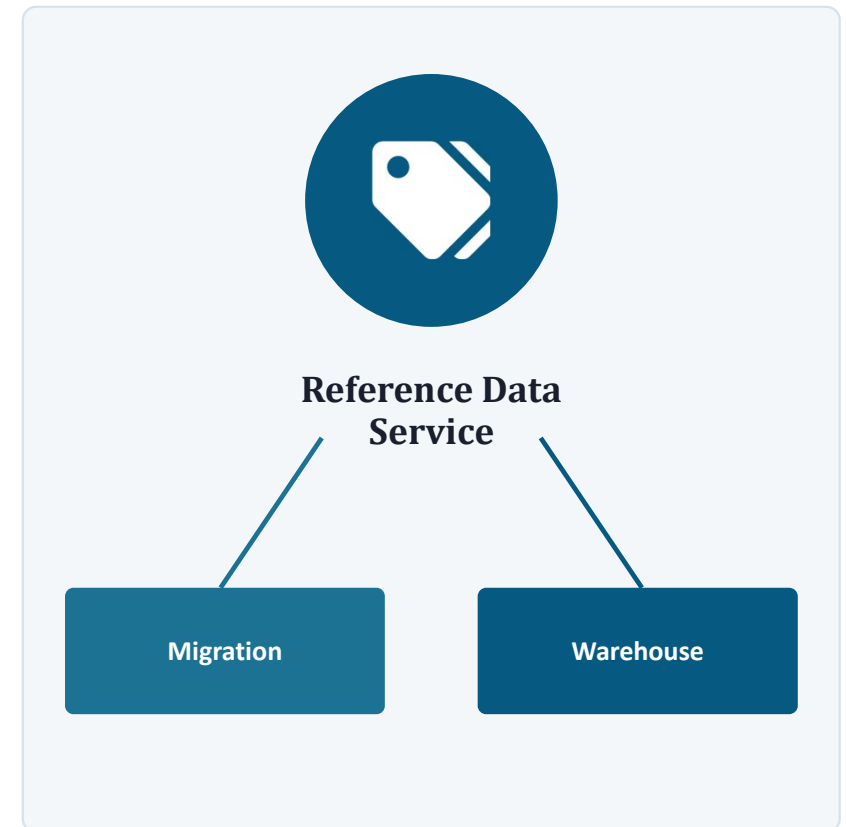
One governed reference-data set, versioned and published — not two spreadsheets maintained by two teams.

- **Shared ownership**

A named steward approves changes for both consumers, instead of each team resolving codes independently.

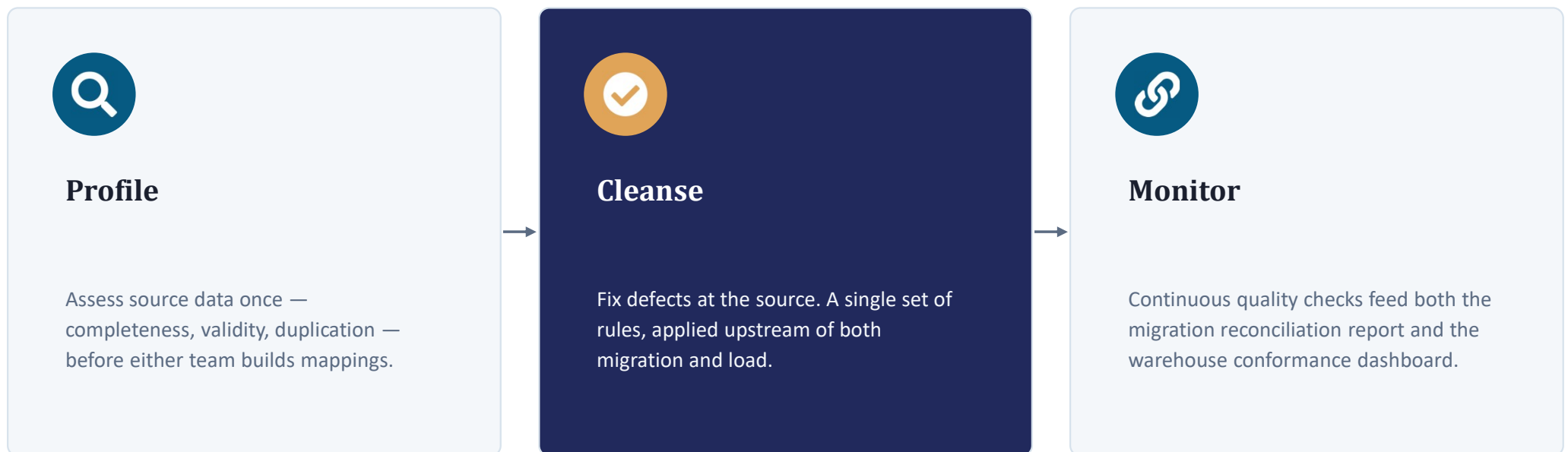
- **Used, not duplicated**

Migration mapping tables and warehouse conformed dimensions both pull from the same lookup source.



Data quality: one process, two consumers

Clean, well-structured, accurately labeled data is what makes both a migration trustworthy and a warehouse usable. Cleansing, deduplication, and enrichment done once at the source removes the need to repeat it downstream in either target.



02 · FOUNDATIONAL ARCHITECTURE

Data models: the blueprint both teams build to

Data models are foundational to data architecture — they provide a structured representation of data elements and relationships, guide database and warehouse design, and ensure consistency in how data is organized and accessed. Migration mapping and warehouse dimensional design should trace back to the same conceptual and logical models.

Conceptual

Business entities and relationships — the shared vocabulary both teams agree to before either writes a mapping.

Logical

Attributes, keys, and relationships defined independent of technology — what migration mapping and dimensional design both derive from.

Physical

Platform-specific implementation — legacy schema for migration, star/snowflake schema for the warehouse — diverging only here, by design.

Layer	What it contains	How migration uses it	How the warehouse uses it
Canonical business entities	The business's definition of Customer, Product, Account, Site, etc. — not a copy of any one system's model	Target operational schemas are designed to reflect these entities; migration mappings transform source data into canonical shape first, then into the target application model	Conformed dimensions are built directly from canonical entities; facts link to conformed keys so every report agrees on what a "customer" is
Business glossary – Data Catalogue	Plain-language definitions, business rules, and allowed values for each entity and attribute	Used to validate that migrated data meets the business definition before cutover	Used to drive report labels, KPI definitions, and metric logic so the warehouse speaks the business's language
Reference / master data	Shared code tables, lookup values, status codes, categorisations	Migration maps legacy codes to standard reference values so the operational system starts clean	Warehouse uses the same reference values for consistency in reporting and drill-downs
Data quality rules & thresholds	Validation rules, completeness %, accuracy targets, duplicate thresholds — agreed and signed off	Migration load reports and trial runs are measured against these thresholds; failures block cutover	Warehouse ingestion checks run the same rules; rejected records go to quarantine and are reported to data owners
Mapping registry	The authoritative source-to-target mapping spec for every field, transformation, and derivation	Migration ETL/load jobs are built directly from these specs	Warehouse ETL/ELT lineage traces every warehouse field back through the same mappings to source
Lineage	End-to-end traceability from source field → mapping → target operational field and/or warehouse field	Proves to auditors and business owners that data landed where it should	Lets analysts trace any KPI on a dashboard back to its source field and transformation
Ownership & stewardship	Named data owner, data steward, and SME for each canonical entity and critical attribute	The owner signs off migration data quality for their domain before cutover	The owner signs off warehouse dimensions and KPI logic for their domain before go-live

A UK Governments data program

A major government transformation programme replacing a fragmented, ageing legacy IT estate with standardised, modern systems — while running parallel migration and data warehouse workstreams against a legacy analytical landscape.



**50 legacy
source
systems**



**6 Key Data
Projects**



20TB data



**3 Cloud
Providers
18 Suppliers**

What happened without shared architecture

- **Reactive, ad hoc data management**

Data issues were handled after they surfaced rather than prevented — there was no proactive strategy for either the migration or ongoing quality.

- **Fragmented reference data**

Asset data was split across Oracle, Ingres, DB2, and IMS — and some IMS data wasn't even replicated into the systems the data team could reach.

- **Projects overran budget and schedule**

Data-improvement work was consistently over time and over budget, with no shared plan or dedicated coordination pulling it back on track.

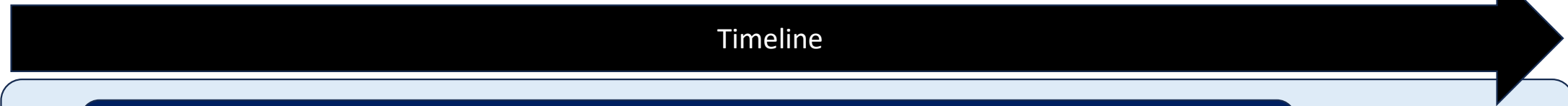
- **Standards existed but weren't adopted**

Data standards — needed for the Ellipse migration itself — had no mandate behind them, so different groups didn't follow them consistently.



Real-world stakes: incidents tied to insufficiently maintained network assets had already affected public safety and supply reliability — underlining exactly how much rode on getting this asset data right.

As-Is Time Line



Timeline

Data Migration (45 Legacy Systems into 3 COTS packages)

Data Warehouse (20TB on premise Oracle DW moving to OCI)



Foundational Services

>> Reference Data >>

>> Data Quality >>

>> Data Catalogue >>

Early outcomes and trajectory



Clear roles and accountability

A Data Governance Council, Data Custodians, and Data Stewards replaced ad hoc ownership — giving the data team an actual mandate to act.



Tools funded and adopted

With executive backing, the team purchased the metadata repository and profiling/cleansing tools it had wanted for years.



Quality metrics established

For the first time, the organization could measure and publish data-quality metrics tied to its improvement initiatives.



Built to scale

The Custodian/Steward/User Group model was designed to be adopted by other divisions, with the Governance Council as the connective layer.

The governance structure was newly introduced at the time of this study — its longer-term impact on migration and reporting outcomes was still being tracked.

The fix: govern data as a shared enterprise asset

Company A moved from informal “governance by stewardship” to a formal, mandated Data Governance structure spanning organizational bodies, standards, and technology — the same three layers of shared architecture from Section 2.



Organizational Bodies & Policies

- Governance structure & decision rights
- Data custodianship
- Issue escalation process



Standards & Processes

- Data definitions & metadata management
- Data profiling & cleansing
- Metrics development & monitoring



Governance Technology

- Metadata repository
- Data profiling tool
- Data cleansing tool

What this case confirms

- 1 Reactive data management guarantees the same defect gets found twice — profiling upstream is cheaper than remediating downstream in two targets.
- 2 Reference data spread across disparate databases with inconsistent access is a governance failure as much as a technical one.
- 3 Standards without a mandate don't get adopted — they need an accountable owner, not just a document.
- 4 Consensus across divisions doesn't happen on its own — it needs an escalation structure with real decision rights.
- 5 None of this is solvable by the migration team or the warehousing/reporting team alone — both need to report into one governance layer.

How to align the two workstreams



One governance body

A shared data-governance forum — Company A's Council/Custodian/Steward model is one proven pattern — owns reference data, quality rules, and model changes.



One shared plan

Sequence source freezes, profiling, and cutover windows on a single timeline both teams plan against, not two plans assuming exclusive access.



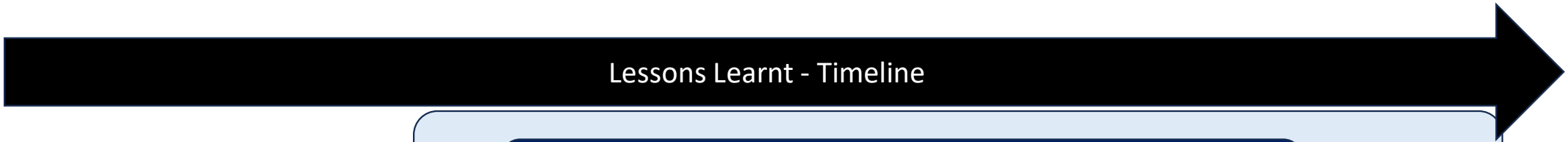
Fix once, at the source

Data quality issues get remediated upstream, before either mapping or modeling work begins — not patched separately in two targets.



One logical model

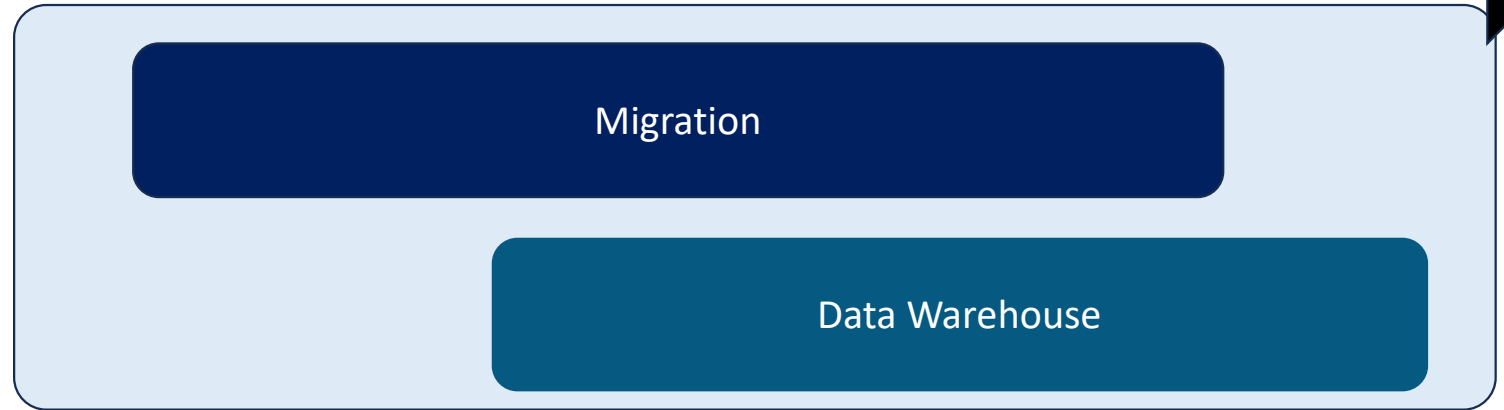
Both migration mapping and warehouse dimensional design trace back to the same conceptual and logical model, diverging only at the physical layer.



Lessons Learnt - Timeline



Key Projects



Migration

Data Warehouse

>> Reference Data >>

>> Data Quality >>

>> Data Catalogue >>

>> Shared Governance >>



Foundational Services

KEY TAKEAWAY

Treat migration and warehousing as one program with one architecture

Shared reference data, one quality process, and one logical model turn two competing projects into a single, coherent delivery — with risk found once, not twice.

Data Governance structure

Three tiers of accountability — strategic, tactical, and operational — connect business and IT around shared data.

