International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(20s) 2026 pp. 1608-1617
https://doi.org/10.70917/ijcisim-2026-6021
Article

Cloud Migration of Enterprise Reporting Platforms with Automated Validation of Output Data Quality

Naga Ravindranatha Reddy Mallipeddi

Independent Researcher

Abstract: A reporting platform builds trust one report at a time, but in a migration, all of that trust can be lost in a single event. This paper treats cloud migration of enterprise reporting platforms and automated validation of report output data quality as a single engineering problem instead of build and testing phases. Output data quality is defined at the report boundary of a published figure as it matches or does not match what the incumbent platform produced. It is distinct from record-level data quality dimensions. Drawing on practitioner experience building source-to-target reconciliation frameworks for large reporting migrations in regulated financial services, the paper presents a reconciliation ladder mapping validation techniques to architectural layers and stages of migration. Cross-sector findings for finance, commerce and health system reporting include constraints that design validation architecture, with also harness-first sequencing, baseline-freeze discipline and anti-patterns for migration pitfalls. Taking into account machine-assisted migration, in the model LLMs compress the burden of mechanical translation and a named human retains adjudication authority over materiality, tolerance and cutover.

Keywords: cloud migration, reporting platform modernisation, data quality of output, source to target reconciliation, data observability, migration assurance, human and AI collaboration

1. Introduction

1.1 Why reporting platforms are a distinctive migration problem

Consider a controller who has closed the same set of books for eleven consecutive quarters and can detect a tenth-of-a-percentage-point discrepancy before it is formally reported. Faster queries cannot compensate for an incorrect result. This distinguishes reporting platforms from many other migration targets: business users judge correctness during every reporting cycle, not only at go-live. Yet many migration programs begin validation only after substantial target development, when defects are more expensive to isolate and correct.

A systematic review of 47 legacy-to-cloud migration studies found that migration models give limited attention to quality during the transition, allowing defects to surface after cutover [1]. The central problem is therefore not whether cloud infrastructure can compute an answer, but whether the organization can produce evidence that the answer is correct. Gartner estimates that poor data quality costs organizations at least USD 12.9 million annually [2]. Migration increases exposure because every remodeled table, translated rule, and changed schedule creates an opportunity for silent divergence: a pipeline can complete successfully while a published figure is wrong. Automated controls for schema, counts, duplicates, business rules, and model behavior are needed to detect that divergence before it reaches a report consumer.

1.2 Scope, position and contribution

Copyright: © 2026 by the authors

This paper addresses the engineering challenge of moving an enterprise reporting platform to cloud infrastructure while proving, through automated and repeatable evidence, that published outputs remain correct. The framework is vendor-neutral because durable migration controls should be defined as capabilities rather than tied to a product roadmap. Its contributions are fourfold: it defines output data quality at the report boundary as a testable construct; introduces a reconciliation ladder that links architectural layers to migration gates; synthesizes anonymized cross-sector patterns and practitioner guidance; and proposes a human-accountability model for machine-assisted migration and validation.

2. Materials and Methods

2.1 The reporting platform as a unit of migration

A reporting platform is not simply a database with a dashboard attached. It is a chain of source extracts, staged and conformed layers, business-rule transformations, semantic definitions, scheduling logic, entitlement rules, and report definitions. A change to one link can alter downstream behavior in ways that are difficult to predict. Migration scope should therefore be established by tracing dependencies backward from published outputs. This process exposes the hidden factory: logic embedded in stored procedures, scheduling scripts, spreadsheet macros, and manual adjustments. The most difficult elements are often not technically complex; they are undocumented. Cataloging dependencies, entitlement rules, and manual interventions before implementation is consequently one of the highest-leverage migration activities.

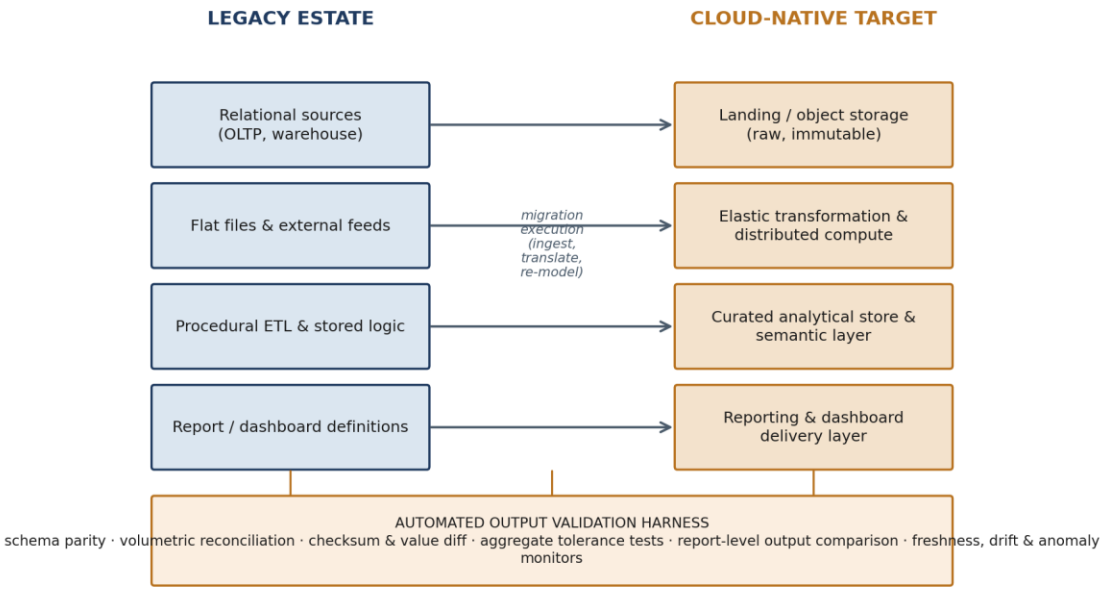


Figure 1. Reference architecture for a reporting platform migration. The validation harness spans both estates and is treated as a first-class component rather than a testing afterthought.

2.2 Correctness at the report boundary

Classical data quality frameworks use a dimensional approach and include accuracy, completeness, consistency, timeliness, validity and uniqueness as proposed in the DAMA framework. The ISO/IEC 25012 framework uses a more wide-ranging set of intrinsic and system-dependent characteristics [3]. More recent extensions of the taxonomy propose dimensions such as governance, usefulness and semantics [4] to increase comparability between vocabularies used by different companies/industries to describe the same concepts. These are relative to a record on its own terms. A migrated reporting platform needs a formulation applied to the report boundary, asking the question "does this published figure equal what the platform it replaces would have produced".

Restated at that boundary, each classical dimension is machine-checkable and completeness is row and population parity against the baseline, while accuracy assumes value equality within a defined tolerance. Consistency becomes agreement between the same measure across the estate, which is a higher bar than it might seem. It is not uncommon for organisations to find that "revenue" has actually meant three different things for years at migration time. Timeliness is conformance to freshness commitment. Validity means conformance with the business glossary. Traceability - not one of the six classical dimensions - means the ability to trace each figure back to its source records, and is highly prized in regulated industries.

Dimension	Assertion at the report boundary	Automated test	Typical failure signature
Completeness	Population and grain parity with baseline output	Row, distinct-key and null-rate comparison	Silent filter or join loss in migrated logic
Accuracy	Measure values equal within declared tolerance	Aggregate delta and row-level hash comparison	Rounding, data-type or precision divergence
Consistency	Same measure agrees across all reports using it	Cross-report measure reconciliation	Competing definitions inherited from silos
Timeliness	Output available within published freshness commitment	Freshness and pipeline latency monitors	Dependency or scheduling translation defects
Validity	Values conform to glossary definition and domain	Rule-based domain and referential checks	Reference data not migrated alongside fact data
Traceability	Every figure resolves to contributing source records	Column-level lineage capture and drill-through test	Lineage never recorded in the legacy estate

Table 1. Data quality dimensions restated as machine-checkable assertions at the report boundary.

2.3 Automated differencing and reconciliation methods

Manual inspection of sampled rows and summary totals is insufficient for business-critical reporting. A defensible approach uses automated, full-population differencing in layers: record-count comparison, field-level comparison for critical records, and derived-metric reconciliation to confirm that business calculations remain equivalent across platforms. Hash-based comparison is particularly useful when row counts agree but values have changed. It must, however, be paired with stable ordering, canonical null handling, and declared precision rules so that technical representation differences are not misclassified as business defects.

Cost and tolerance determine when full-population comparison is practical. Comparing every row of a large fact table across two platforms can be expensive, so teams may use partition-scoped differencing, risk-based stratification, or incremental comparison. Tolerance does not mean accepting unexplained differences. Some variances arise from documented deduplication changes, floating-point behavior, time-zone normalization, or corrected legacy defects. Each acceptable variance should therefore be quantified, approved by the business owner, versioned, and retained as cutover evidence.

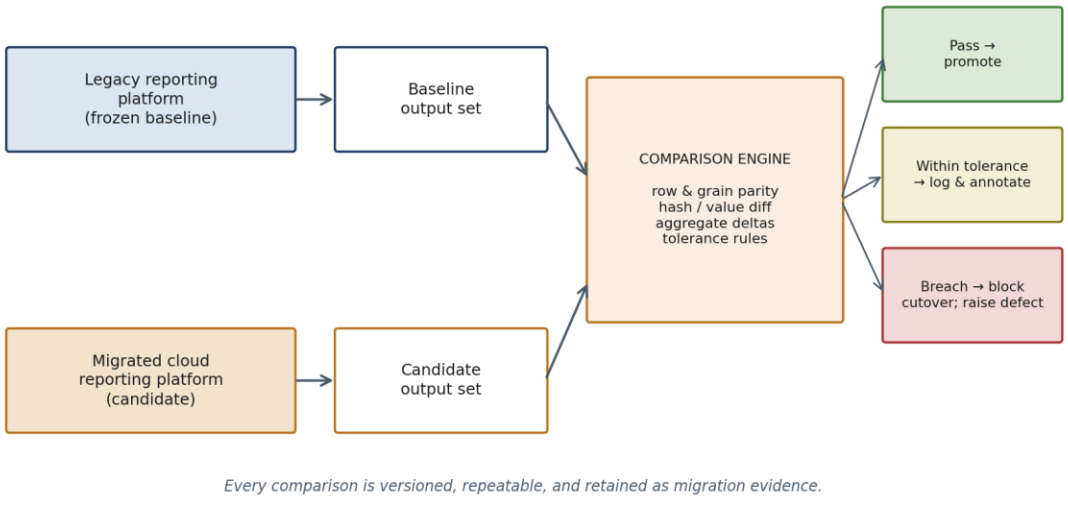


Figure 2. Dual-run reconciliation. Both platforms generate outputs from the same frozen source snapshot; the comparison engine, not human inspection, determines whether cutover proceeds.

2.4 Data contracts and post-cutover observability

Work in a second direction moves quality enforcement upstream of reconciliation. Data contracts define schema, grain, ownership and validation rules as a formal contract between producer and consumer, so that breaking changes are rejected at commit time, rather than discovered later in layers downstream [5]. After reconciliation is complete, on the day the legacy baseline is retired, observability takes over to track if the data freshness, volume, schema drift and lineage after cutover degrade in ways reconciliation is not set up to notice. Alert precision remains the binding constraint, because a detector that cries wolf trains its own team to stop listening within weeks.

3. Results

3.1 The reconciliation ladder

Another method of sequencing verification by architectural layer is to use a gated process. Each gate holds or halts further processing until the previous step is complete. Row counts and null-rate checks indicate truncated or missing extracts upon ingestion, and conformed-layer distinct-key constraints and data-type validation codes catch join fan-out and type coercion problems before they manifest as some subtly wrong total several transformations down the line. Curated layer aggregate deltas and measure-definition validation catches business-rule translation mistakes before end users ever see them in published report output. Full output diff by grain, period and entitlement with business sign-off catches filter and formatting divergence, including correctness proven only for an administrative user who sees everything, not who actually reads the report. After cutover, freshness and drift monitors can detect degradation since the comparison baseline is absent.

Architectural layer	Reconciliation test	Gate decision	Defect class detected
Ingestion / landing	Row counts, file and partition manifests, null rates	Block promotion to transformation	Truncated, duplicated or missing extracts

Conformed / staged	Distinct key counts, referential integrity, data types	Block promotion to curation	Join fan-out, type coercion, reference gaps
Curated / semantic	Aggregate deltas within tolerance, measure definitions	Block report rebuild	Business rule translation defects
Published report	Full output diff by grain, period and entitlement	Block cutover; require business sign-off	Filter, security and formatting divergence
Post-cutover run	Freshness, volume, schema drift, distribution monitors	Raise incident; trigger rollback assessment	Degradation after baseline is retired

Table 2. The reconciliation ladder: validation techniques mapped to architectural layers and gate decisions.

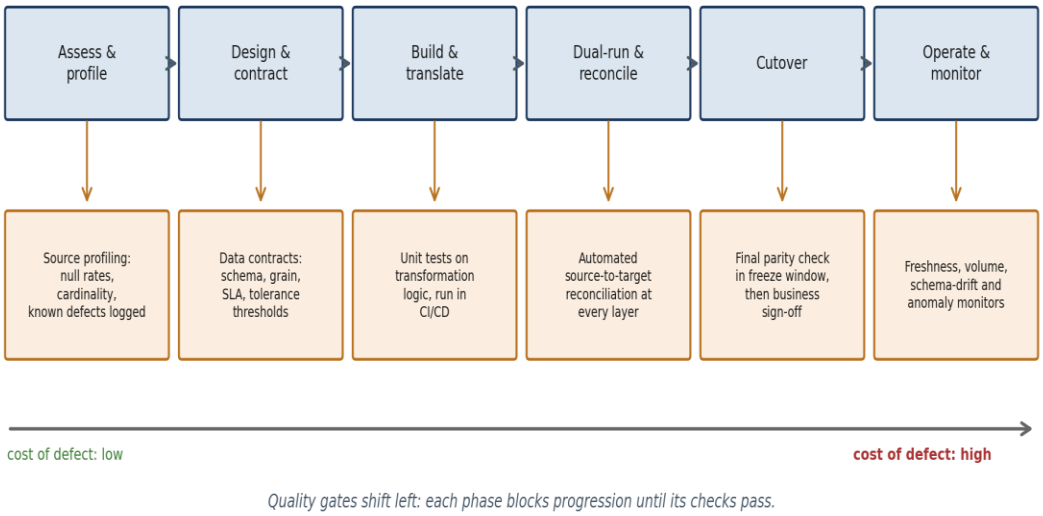


Figure 3. Quality gates positioned across the migration lifecycle. Each phase blocks progression until its checks pass, so that defect cost is incurred early rather than at cutover.

3.2 Cross-sector migration patterns

Three composite patterns, generalized from public descriptions of modernization practice and anonymized practitioner evidence, illustrate how the framework operates under different constraints. A multi-entity banking group migrating risk and regulatory reporting faced supervisory expectations that every aggregate be traceable to the transactions that produced it [6]. Lineage therefore became a prerequisite for acceptance rather than a post-implementation enhancement. The migration sequence was reversed: the validation harness was built first, and both platforms were run nightly against frozen source snapshots over several reporting cycles. Many residual differences were not translation errors but undocumented legacy adjustments. Converting those adjustments into coded, reviewable rules created a control improvement that remained valuable after migration.

A leading multi-channel retailer found its target headline metric had conflicting and disparate definitions across its regional reports that could not be reconciled side-by-side. Simplistically put, migration could only support a single definition for each certified measure in the target semantic layer. Therefore, technical migration was a governance challenge. The team sequenced migration of reports based on importance, first publishing a harness for a low-visibility report to prove the approach. Side-by-side row comparison during development does not work when a health system

cannot copy identifiable patient records to lower environments. The harness reconciled on salted hashes and distributional statistics, only doing the full-fidelity comparison inside the production boundary. This is a general lesson: the validation architecture must answer to the same governance regime it validates.

Pattern	Dominant constraint	Validation emphasis	Transferable lesson
Financial reporting	Supervisory traceability; audit evidence	Extended dual-run; column-level lineage	Variance mostly from legacy adjustment, not translation
Commerce reporting	Conflicting metric definitions	Cross-report consistency; certified measures	Sequence by criticality; prove harness on a thin slice
Health system reporting	Privacy limits on data movement	Hashed-key and aggregate reconciliation	Design validation under the governance regime

Table 3. Comparison of anonymised migration patterns and the lessons that transfer between them.

3.3 Sector constraints on validation design

Banking and insurance provide particularly concrete examples because risk-data aggregation policies require accurate consolidation across legal entities and geographies, together with drill-down from summary values to individual positions [6]. Lineage therefore becomes a formal deliverable, and validation evidence must be understandable to reviewers outside the delivery team. Healthcare reporting combines clinical, laboratory, pharmacy, and device data under privacy regimes that restrict where identifiable information may be processed. Validation can be performed with salted hashes, aggregates, and distributional statistics, limiting movement of identifiable records while reflecting privacy-preserving techniques formalized for constrained health-data sharing [7].

Manufacturing and telecommunications deliver volume and velocity, rendering governance constraints as telemetry that is not only legitimately late-arriving but also watermarked and windowed, converging over time, not instantly: as is often the case with retail, the semantic consolidation of Pattern B, with drifted metric definitions resolving into one certified version, was a social construct, albeit one framed by data mesh literature as domain-owned data products held to a single certified definition rather than an IT afterthought [8]. Public sector reporting carries its own overhead, since figures published to citizens enter the public record.

Sector	Dominant constraint	Validation emphasis
Financial services and insurance	Supervisory traceability; audit evidence	Column-level lineage; extended dual-run; independent re-execution
Healthcare and life sciences	Privacy regimes limiting data movement	Hashed-key, aggregate and distributional reconciliation
Manufacturing and telecommunications	Continuous, high-volume, late-arriving telemetry	Windowed and watermark-aware comparison; convergence tolerance
Retail and multi-channel commerce	Competing metric definitions across channels	Cross-report consistency; certified semantic definitions
Public sector	Published figures enter the public record	Extended parallel operation; retained restatement evidence

Table 4. Sector constraints and the validation emphasis each implies.

3.4 Practitioner guidance and anti-patterns

A recurring sequencing failure is to treat validation as a late testing activity that begins only when the first report is ready for inspection. By then, an ingestion-layer divergence may already have propagated through every downstream transformation. Building the validation harness first ensures that the earliest migrated object passes

through the same controls later applied to critical regulatory or executive reports. The harness should be implemented as version-controlled code, executable on demand, and designed to produce a dated evidence artifact for every run.

Some anti-patterns are also highly predictive of disaster: line by line transliteration instead of declaratively recomposing; checking the final report without checking layers in between; sampling for regulatory output instead of using the entire population, because the error population lives in the tail, and the tail is never reached if sampling; waiting to test entitlement until after cutover; and decommissioning the legacy platform before the full cycle has run on the replacement. Another would be optimizing cloud cost before proving correctness, because aggressive tuning will introduce divergence at exactly the point at which a team has variance and wants to mean "defect" and not "the only thing different is the customer".

4. Discussion

4.1 Societal stakes of trusted reporting migration

On the other hand, errors in reported outputs do not stop with that organisation. Reporting outputs can be used in regulatory filings, communications to investors, public health data, as well as be included in automated decisions without human oversight, so in this sense a migration defect is not a private engineering matter. Because of the escalating-cost models that show a defect found at entry costs an order of magnitude less to fix than later ones, late discovery is disproportionately damaging. The case for automated validation in social terms is that an organization that presents a dated, reproducible comparison can answer a challenge with the evidence of that comparison rather than with reassurance.

4.2 Environmental and economic stewardship of validation compute

Validation is a workload in itself, and full-population comparison within deep historical timelines can consume compute resources. Data centres in 2024 were estimated to account for around 1.5 per cent of global electricity consumption, rising at 12 per cent per year since 2017 [9]. It is estimated that a large percentage of cloud expenditure is spent not generating business value. Best practices for a responsible approach to cloud comparisons include changing partition comparisons, scheduling off-peak runs, saving evidence artifacts, and tiering historical data to keep the depth of comparisons without constant scans.

4.3 Workforce and the evolution of the reporting engineer's role

The reporting engineer's role has expanded during the past decade. Traditional strengths in SQL, dimensional modeling, scheduling, and report design remain necessary, but they are no longer sufficient. Cloud migration also requires test engineering discipline, declarative infrastructure, statistical reasoning about tolerances, and the ability to explain variances to business owners. Data stewardship is most effective when it is assigned to named individuals with authority to approve definitions and exceptions.

4.4 Machine assistance in legacy logic translation and validation

Existing reporting estates contain large amounts of undocumented procedural logic and sparse unit test coverage. Automated dialect translation remains a current research goal and the problem of source-to-target syntax mapping continues to be an open problem [10]. Case studies of foundation models applied to millions of lines of undocumented database code show that well-structured prompts can lead to syntactically correct and often functional code. However, the small number of evaluation examples and non-deterministic behavior remain limits to this work [11]. Related work on output-format bias (e. g. raw model output) found that much model output would require careful filtering or transformation to avoid unfair bias [12]. An evaluation of AI agents for project-level Java migrations found it necessary to build a multi-gate protocol checking compilation, test passing and preserved coverage together because agents can appear successful on a narrower metric without preserving correct behavior [13].

Validation is a particularly strong use case for machine assistance. Systems can propose test assertions from profiling statistics and perform exhaustive differencing at a scale that manual review cannot match, reducing thousands

of variances to a smaller set of likely causes. Learned anomaly detection complements rule-based checks: rules encode known conditions, while anomaly models can reveal previously unformalized drift. These benefits depend on trustworthy inputs, because data-quality defects in training and evaluation data can degrade model performance [14].

4.5 The proposal-and-adjudication accountability model

Machine-assisted migration requires explicit ownership. Responsibility should not default to whichever individual faces the greatest delivery pressure at the time of cutover. Automation and human cognitive bias can either compound or counterbalance one another, depending on how decision rights and review responsibilities are designed [15]. Data quality is also a prerequisite for reliable automation; poor-quality inputs can undermine both generated translations and anomaly-detection results [14].

A practical governance pattern is proposal and adjudication: machines propose translations, test assertions, and draft explanations, while a named human accepts, corrects, or rejects each proposal. Decisions are recorded as signed, replayable evidence rather than inferred from silence. This arrangement preserves accountability with a person who has the authority and subject-matter knowledge to justify the decision. Research on accountability in human-AI collaboration similarly indicates that responsibility can become blurred when automation is introduced without explicit decision rights [16].

Activity	**Machine contribution**	**Human authority**	**Evidence artefact**
Dependency discovery	Parse code and schedules; construct candidate graph	Confirm scope and exclusions	Approved scope inventory
Logic translation	Draft target-dialect equivalent	Review semantics; accept or rewrite	Reviewed, merged pull request
Test generation	Propose assertions from profiling statistics	Set materiality and tolerance	Versioned contract and test suite
Reconciliation	Execute full-population comparison	Interpret and adjudicate variance	Dated comparison report
Cutover decision	Summarise residual variance	Authorise or defer cutover	Signed sign-off record
Post-cutover monitoring	Detect drift and anomalies	Triage, escalate, remediate	Incident and resolution log

Table 5. Division of labour between machine assistance and human authority across the migration and assurance lifecycle.

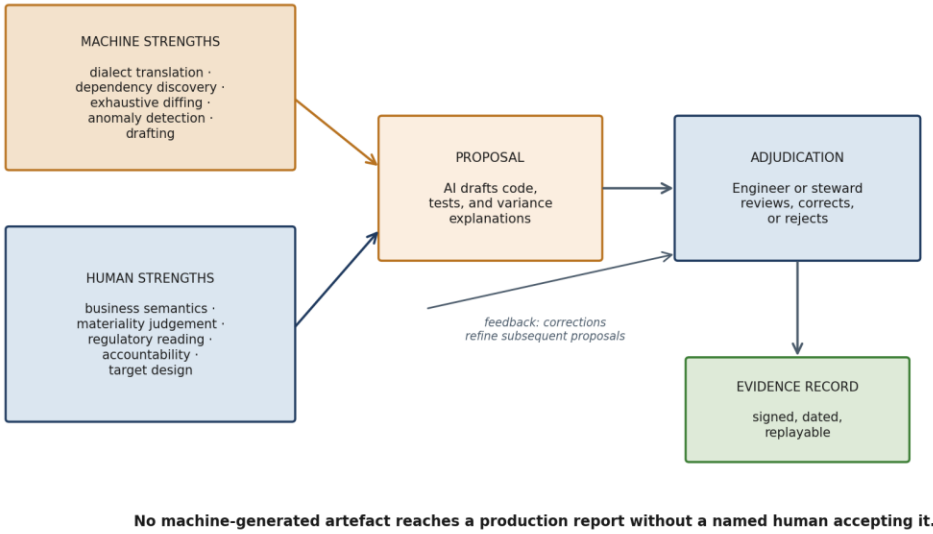


Figure 4. The proposal-and-adjudication model. Machine assistance compresses effort; human adjudication supplies the accountability that assistance cannot.

4.6 Limitations and future research directions

Beyond that, the computational difficulty of efficient full-population differencing over a, say, petabyte of historical depth remains an open problem. More broadly, there is a need for pushdown comparison and probabilistic methods better bounded on their error. Automatically capturing column-level lineage across heterogeneous engines is also an open problem and the binding constraint on regulatory traceability. Formal tolerance semantics, a machine-verified way of specifying and checking which perturbations are acceptable, could replace the emerging practice, often in spreadsheets or bespoke tools, in companies. Current evals for automated translation still rely on small datasets and non-deterministic models, and could therefore benefit from benchmarks which, like the test set, aim for behavioral rather than syntactic equivalence. The FreshBrew benchmark gestures in this direction, though not fully realized [13]. As more pipelines report outputs as input to downstream consumers or automated decisioning, assurance will need a confidence language that provides more information than a binary flag that the pipeline was executed without error.

5. Conclusions

Success or failure of the move to cloud of the reporting platform depends on whether the correctness of the reporting output can be shown to be preserved. This is an engineering artifact, not a management assertion taken on faith. Migration and verification are not two separate problems: the comparison harness is built before the first report syncs up, the reconciliation gates every architectural layer not just the last, tolerance is quantified and owned by the business team long before cutover, observability carries that assurance all the way through once the legacy baseline is flipped off. Migration programmes fail less often because the target platform cannot compute the right answer, and far more often because nobody built the means to prove the answer had not silently changed along the way. Machine assistance compresses a real share of the mechanical translation and testing burden this work requires, but does not remove the need for a named person who can look at a variance, decide what it actually means, and answer for that decision afterward.

6. Declarations

Funding: none. Author Contributions: sole author: conception, analysis and writing.

Conflict of Interest: None declared.

Data Availability: Not applicable; no original dataset was generated. All quantitative statements are based on cited external sources; sector patterns and career-context observations are based on de-identified composite practitioner experience.

REFERENCES

1. M. H. Hasan, M. H. Osman, N. I. Admodisastro, and M. S. Muhammad, "Legacy systems to cloud migration: A review from the architectural perspective," Journal of Systems and Software, vol. 202, art. 111702, 2023. Available: https://www. sciencedirect. com/science/article/abs/pii/S0164121223000973
2. Gartner, "Data Quality: Why It Matters and How to Achieve It." Available: https://www. gartner. com/en/data-analytics/topics/data-quality
3. ISO/IEC, "ISO/IEC 25012:2008 - Software engineering - Software product Quality Requirements and Evaluation (SQuaRE) - Data quality model," International Organization for Standardization, 2008. Available: https://www. iso. org/standard/35736. html
4. R. Miller, H. Whelan, M. Chrubasik, D. Whittaker, P. Duncan, and J. Gregório, "A Framework for Current and New Data Quality Dimensions: An Overview," Data, vol. 9, no. 12, art. 151, 2024. Available: https://www. mdpi. com/2306-5729/9/12/151
5. J. Wasser, I. Kumara, G. Monsieur, W.-J. Van Den Heuvel, and D. A. Tamburri, "Data Contracts in Data Mesh: A Systematic Gray Literature Review," in Business Modeling and Software Design (BMSD 2025), Lecture Notes in Business Information Processing, vol. 559, Springer, Cham, 2026, pp. 21-38. Available: https://link. springer. com/chapter/10.1007/978-3-031-98033-6_2
6. L. Prorokowski, "Risk data validation under BCBS 239," Journal of Risk Model Validation, vol. 13, no. 3, pp. 45-71, 2019. Available: https://www. risk. net/journal-of-risk-model-validation/6903226/risk-data-validation-under-bcbs-239
7. S. Karagiannis, C. Ntantogian, E. Magkos, A. Tsohou, and L. Landeiro Ribeiro, "Mastering data privacy: leveraging K-anonymity for robust health data sharing," International Journal of Information Security, vol. 23, no. 3, 2024. Available: https://link. springer. com/article/10.1007/s10207-024-00838-8
8. I. A. Machado, C. Costa, and M. Y. Santos, "Data Mesh: Concepts and Principles of a Paradigm Shift in Data Architectures," Procedia Computer Science, vol. 196, pp. 263-271, 2022. Available: https://www. sciencedirect. com/science/article/pii/S1877050921022365
9. International Energy Agency, "Energy and AI - Executive Summary," IEA, 2025. Available: https://www. iea. org/reports/energy-and-ai/executive-summary
10. "Translating between SQL Dialects for Cloud Migration," in Proceedings of the 46th International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP 2024), ACM, 2024. Available: https://dl. acm. org/doi/10.1145/3639477.3639727
11. L. Solovyeva, E. Carneiro Oliveira, S. Fan, A. Tuncay, S. Gareev, and A. Capiluppi, "Leveraging LLMs for Automated Translation of Legacy Code: A Case Study on PL/SQL to Java Transformation," arXiv:2508.19663, 2025. Available: https://arxiv. org/html/2508.19663
12. "Output Format Biases in the Evaluation of Large Language Models for Code Translation," in Proceedings of the 2024 IEEE/ACM First International Conference on AI Foundation Models and Software Engineering, 2024. Available: https://arxiv. org/abs/2403.17214
13. V. May, D. Misra, Y. Luo, A. Sridhar, J. Gehring, and S. S. Ribeiro Junior, "FreshBrew: A Benchmark for Evaluating AI Agents on Java Code Migration," arXiv:2510.04852, 2025. Available: https://arxiv. org/pdf/2510.04852
14. S. Mohammed, L. Budach, M. Feuerpfeil, N. Ihde, A. Nathansen, N. Noack, H. Patzlaff, F. Naumann, and H. Harmouch, "The effects of data quality on machine learning performance on tabular data," Information Systems, vol. 132, art. 102549, 2025. Available: https://www. sciencedirect. com/science/article/pii/S0306437925000341
15. D. Van Rooy, "Human-machine collaboration for enhanced decision-making in governance," Data & Policy, vol. 6, art. e72, 2024. Available: https://www. cambridge. org/core/journals/data-and-policy/article/humanmachine-collaboration-for-enhanced-decisionmaking-in-governance/0503282B5F04A6C52DDDCFAA883DEC31
16. Q. Jiang, "Accountability shift in human-AI collaborations," Artificial Intelligence and Law, 2026. Available: https://link. springer. com/article/10.1007/s10506-026-09536-3