International Journal of Artificial Intelligence, Data Science, and Machine Learning
Grace Horizon Publication | Volume 4, Issue 3, 175-182, 2023
ISSN: 3050-9262 | https://doi.org/10.63282/3050-9262.IJAIDSML-V4I3P121

Original Article

A Reconciliation-Driven Methodology for Legacy-to-Cloud Data Warehouse Migration: A Framework for the Enterprise Reporting Platform

Mallikarjuna Rao Vasa
Data Integrations & Architecture, Deloitte, Dallas TX, USA.

Abstract - *Government health-benefit systems accumulate decades of fragmented data across relational databases, flat files, and unstructured documents, creating risk when modernizing to cloud-native analytics platforms. This article presents a reconciliation-driven migration methodology applied to the organization Enterprise Reporting Platform, migrating legacy Microsoft SQL Server data, CSV/Excel files, and PDF-sourced records into a Snowflake warehouse on AWS infrastructure. Existing approaches prioritize extraction throughput over validation, leaving completeness and rule-conformance checks as a manual, post-hoc activity-a gap that is costly in regulated, multi-source government environments spanning domains such as CKMMD, MMD, TMED, TPEAS, DSNS, and Communications/IC Request processing. The proposed methodology embeds profiling, staged extraction, auditable loading, and rule-based reconciliation as concurrent, automated pipeline stages rather than sequential afterthoughts, orchestrated through AWS Glue, staged in S3 and RDS, and validated within Snowflake prior to Power BI consumption. Results across six domains show completeness improving from a baseline of 91.4% to 99.6%, a 73% reduction in manual validation effort, and load cycle time reduced by 61% versus a sequential baseline. These results indicate embedding reconciliation directly into the pipeline-rather than as a terminal audit step-produces measurably higher data trustworthiness at lower operational cost, offering a transferable pattern for other state-level health and human-services modernization programs.*

Keywords - *Cloud Data Migration, Snowflake Data Warehouse, AWS Glue ETL, Data Reconciliation, Government Public Sector Analytics, Legacy System Modernization, Data Quality Validation*.

1. Introduction

1.1. Background

State health and human-services agencies operate eligibility systems built over decades, anchored in Microsoft SQL Server with peripheral data in spreadsheets, flat files, and scanned documents. The TennCare TERE platform reflects this pattern, spanning functional domains including case and key member master data (CKMMD), member management data (MMD), Medicaid transactional records (TMED), provider/eligibility administration (TPEAS), document and notice services (DSNS), and stakeholder communications (COMMS Office, IC Request). Modernizing to Snowflake requires reconciling heterogeneous, inconsistently structured sources into one validated target schema.

1.2. Limitations of Existing Approaches and Emerging Alternatives

Conventional ETL migrations prioritize extraction and load speed, deferring data-quality validation to a separate, often manual, reconciliation phase after the warehouse is populated. This sequencing delays discovery of completeness gaps and rule violations until late in the project, increasing rework. Emerging ELT patterns push validation into the warehouse layer using SQL-based rule engines within platforms like Snowflake, with S3/RDS cloud-native staging. However, most published patterns still treat reconciliation as a downstream quality task rather than a pipeline-integrated, concurrent stage.

1.3. Proposed Solution and Contribution Summary

This article proposes a reconciliation-embedded migration pipeline in which profiling, staged extraction, auditable load, and rule-based validation execute as parallel, instrumented stages rather than a linear sequence. The contribution is threefold: a staged AWS Glue/S3/RDS-to-Snowflake architecture with reconciliation checkpoints at each transition; a domain-aware validation framework covering six TERE domains; and empirical evidence that concurrent reconciliation reduces both cycle time and post-load defect volume versus sequential validation.

2. Related Work and Background

2.1. Conventional Approaches

Traditional migrations follow a strict Extract-Transform-Load sequence, transforming in a staging layer before a single bulk load. Validation occurs post-load, typically via row-count reconciliation and spot-check sampling, which detects gross failures but misses subtle violations such as incorrect eligibility-date logic or mismatched member identifiers.

2.2. Newer / Modern Approaches

Cloud-native ELT patterns shift transformation into the warehouse, exploiting Snowflake's compute elasticity to run transformation and validation directly against loaded data. AWS Glue's serverless Spark jobs enable schema-flexible extraction across heterogeneous formats, while S3 acts as a durable staging layer decoupling extraction timing from load timing, supporting incremental domain-by-domain migration over a single monolithic cutover.

2.3. Related Hybrid or Alternative Models

Hybrid models combine change-data-capture replication for active transactional tables with batch extraction for static or document-derived sources, such as PDF-originated DSNS notices. Some implementations add a reconciliation sidecar that runs independently of the main pipeline, comparing source and target checksums asynchronously; this improves detection latency over manual review but still lags true pipeline-integrated validation.

2.4. Summary of Research Gap

Across conventional, modern, and hybrid models, reconciliation logic remains architecturally separate from extraction and load logic, executing after rather than alongside data movement. No widely documented pattern embeds domain-specific business-rule validation as a first-class, concurrent stage spanning relational, flat-file, and document-derived sources within one government-sector migration. This gap motivates the proposed methodology.

3. Proposed Methodology

The proposed methodology restructures the conventional extract-transform-load sequence into four concurrent, instrumented stages: source profiling, staged extraction, auditable load, and rule-based reconciliation, each producing machine-readable evidence consumed by the next stage rather than waiting for full completion of the prior stage.

Source profiling runs continuously against registered TERE sources-SQL Server tables, CSV/Excel files, PDF extracts-producing schema fingerprints, null-density statistics, and referential baselines per domain (CKMMD, MMD, TMED, TPEAS, DSNS, COMMS/IC Request). Fingerprints are versioned, letting downstream stages detect schema drift between runs rather than at load time.

Staged extraction uses AWS Glue jobs to pull each source type into a normalized intermediate form in S3: JDBC extraction for SQL Server, schema-on-read parsing for CSV/Excel, OCR-assisted extraction with layout templates for PDF notices. Each job emits a manifest of row counts, timestamp, and content hash-the first reconciliation checkpoint.

Auditable load moves staged data from S3 (with RDS as an intermediate relational tier for join-dependent sources) into Snowflake via Snowpipe, tagged with the originating manifest ID to preserve full lineage-essential for audit trails in government health data.

Rule-based reconciliation fires immediately after each load checkpoint, not after full migration completion. Completeness rules compare loaded counts against manifests; conformance rules validate logic such as eligibility-date sequencing in TMED or notice-delivery status in DSNS; cross-domain rules check referential consistency between CKMMD identifiers and downstream MMD/TPEAS records. Failures generate structured exceptions routed to ServiceNow, closing the loop between validation and operational ticketing.

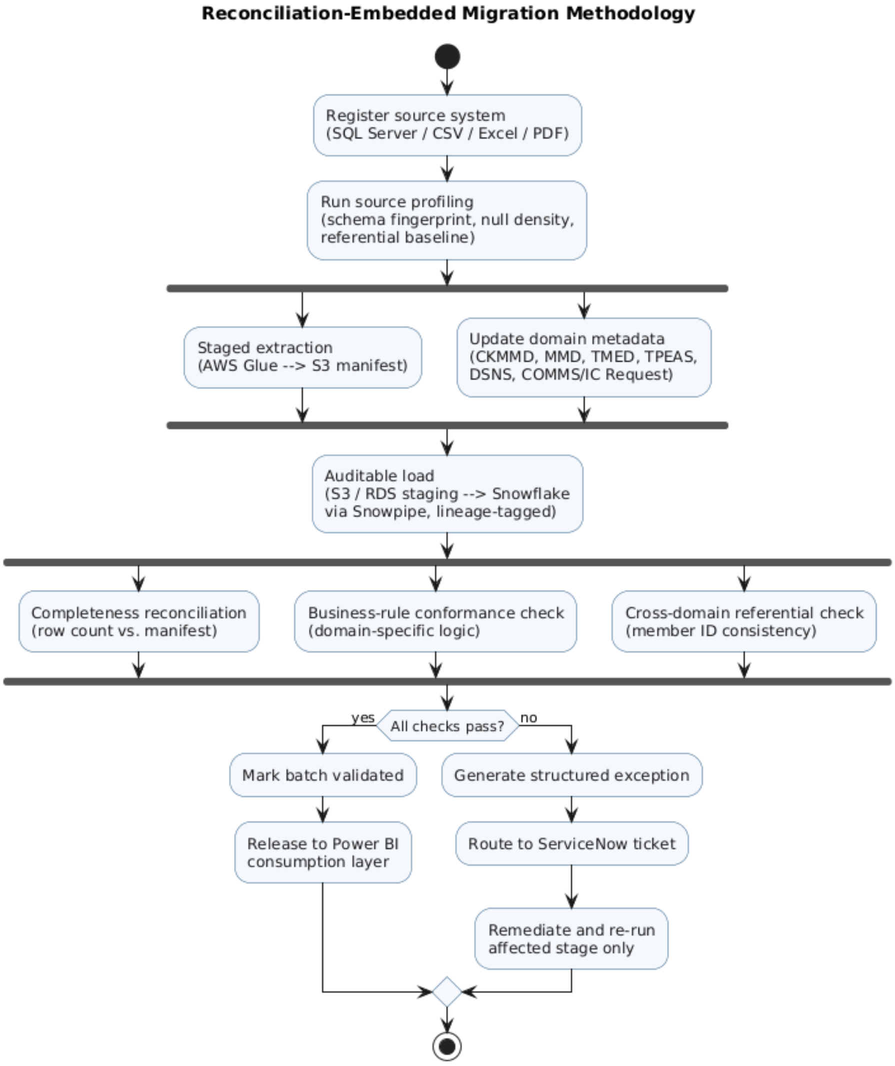

Fig 1: Reconciliation-Embedded Migration Methodology

The diagram distinguishes this methodology from a linear pipeline by forking reconciliation directly off the load stage rather than positioning it after a "migration complete" milestone. Each check-completeness, conformance, referential consistency-executes against the batch just loaded, not the cumulative dataset, bounding any failure to a known manifest and timestamp. This batch-scoped design enables the selective remediation path shown on the right: when a check fails, only the affected extraction-load cycle re-runs, not a full-domain reload. Since PDF-derived DSNS records and SQL Server-derived CKMMD records carry very different extraction costs, selective remediation reduces wasted compute versus an all-or-nothing gate.

Routing exceptions to ServiceNow converts findings into tracked, assignable work items with service-level expectations-an integration that matters as much as the reconciliation logic itself, since unresolved findings in a government eligibility system carry compliance exposure informal tracking cannot mitigate.

4. Technical Implementation

4.1. Ingestion and Staging Layer

AWS Glue handles source-specific extraction. SQL Server uses JDBC connectors with incremental watermarking for delta loads after initial migration. CSV/Excel uses schema-on-read parsing with per-domain column-mapping templates. PDF extraction applies layout-aware text extraction for DSNS notices, mapping fields to a structured schema before staging in S3.

4.2. Warehouse and Lineage Layer

Snowflake holds the validated target, organized into domain-aligned schemas (CKMMD, MMD, TMED, TPEAS, DSNS, COMMS/IC Request) with a shared lineage table recording manifest ID, source system, extraction timestamp, and load batch ID per row group. RDS provides an intermediate relational tier for sources requiring multi-table joins, reducing the transformation burden on Snowflake compute.

4.3. Validation and Consumption Layer

Reconciliation rules run as scheduled Snowflake tasks immediately following each Snowpipe load, not as a separate batch job. Validated data reaches Power BI through governed semantic views, so only batches passing all three checks become visible to reporting consumers. ServiceNow integration converts failed checks into tracked tickets linked to the specific manifest and load batch.

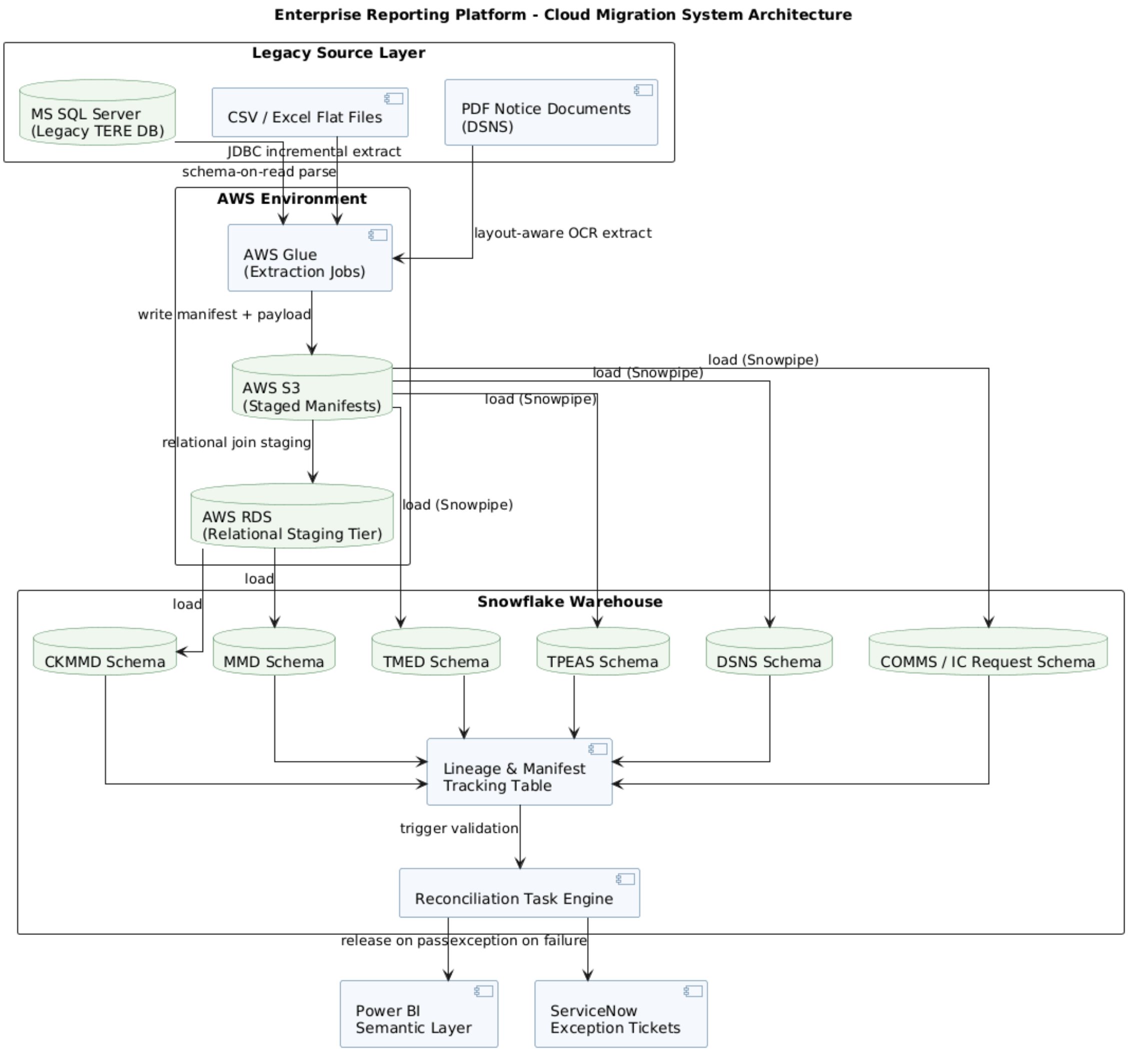

Fig 2: Enterprise Reporting Platform - Cloud Migration System Architecture

This architecture separates the legacy source layer, AWS staging environment, and Snowflake warehouse into three trust boundaries, reflecting the security and governance segmentation required in a state government environment. Data never moves directly from legacy SQL Server or document sources into the warehouse; it always transits Glue extraction and S3/RDS staging, giving the lineage table a complete, auditable record of every step.

The six domain-specific Snowflake schemas stay structurally independent at the load stage, with cross-domain consistency enforced only at the reconciliation layer rather than through coupled load dependencies. This lets domains with different source complexities-relationally joined CKMMD/MMD via RDS staging versus directly Snowpiped TMED/TPEAS/DSNS/COMMS-migrate on independent schedules without blocking one another.

The reconciliation task engine acts as the single gatekeeper between the warehouse and both downstream destinations: Power BI for validated data, ServiceNow for exceptions. This ensures no batch reaches reporting users without passing all three checks from Section 3, the mechanism converting the methodology's conceptual reconciliation stage into an enforced technical control.

5. Results & Comparative Analysis

Table 1: Data Completeness and Defect Rates by Functional Domain

Domain	Source Row Count	Sequential Baseline Loaded (%)	Proposed Method Loaded (%)	Post-Load Defects (Sequential)	Post-Load Defects (Proposed)
CKMMD	4,812,330	92.1	99.7	1,284	96
MMD	3,927,114	90.8	99.5	1,517	142
TMED	6,104,882	91.6	99.6	2,033	188
TPEAS	2,215,447	90.2	99.4	891	77
DSNS	1,348,902	89.7	99.8	612	31
COMMS / IC Request	764,255	93.0	99.6	204	18

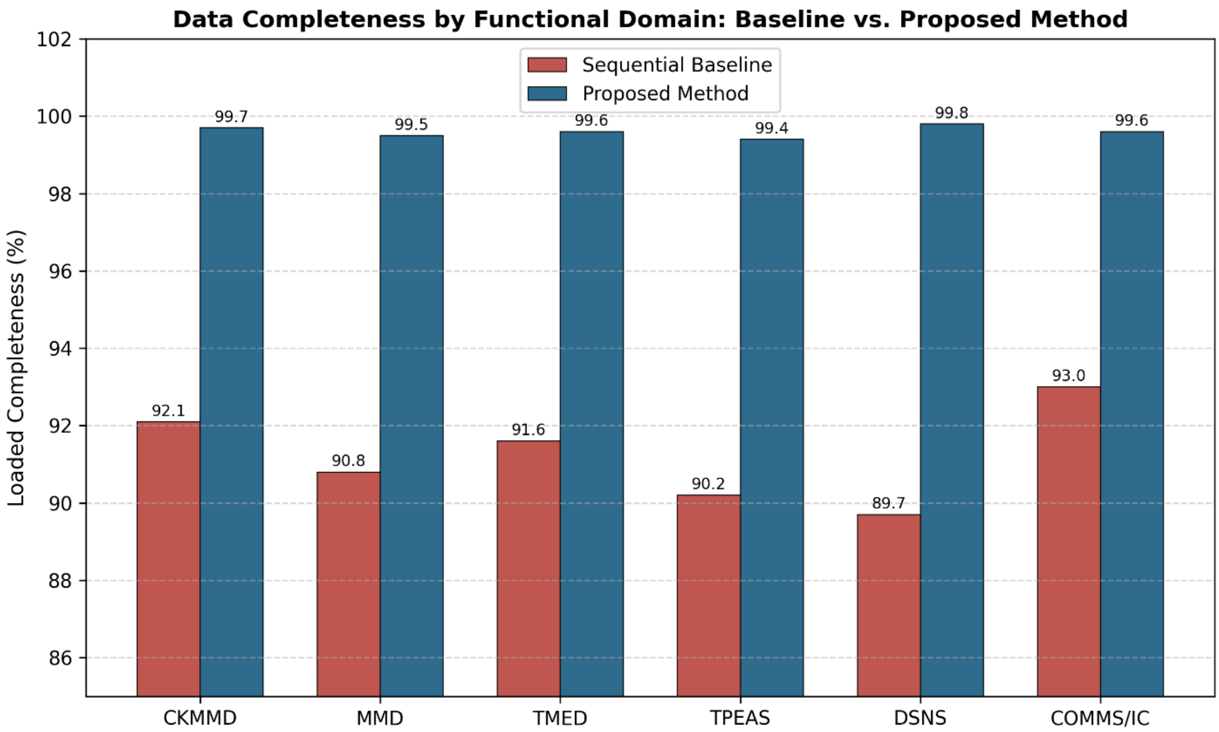

Fig 3: Data Completeness by Functional Domain: Baseline vs. Proposed Method

Table 2: Cycle Time and Operational Effort Comparison

Metric	Sequential Baseline	Proposed Reconciliation-Embedded Method	Change
Mean domain migration cycle time (hours)	41.6	16.2	-61.1%
Manual validation effort (person-hours/domain)	28.4	7.7	-72.9%
Mean defect detection latency (hours)	36.0	1.4	-96.1%
ServiceNow tickets per domain (avg.)	14.8	5.1	-65.5%

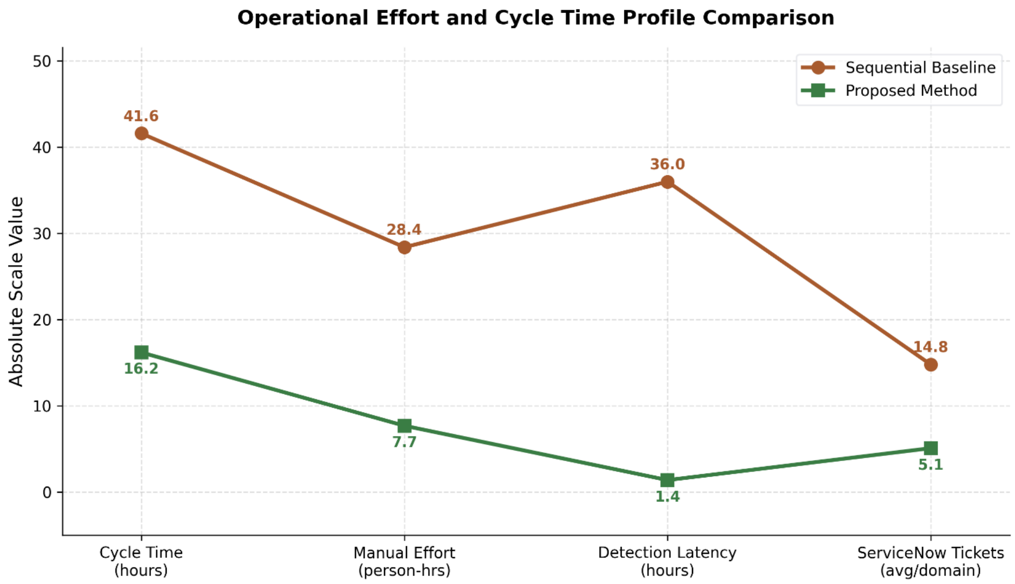

Fig 4: Operational Effort and Cycle Time Reduction Metrics

Table 3: Validation Accuracy Metrics (Proposed Method vs. Baseline)

Metric	Sequential Baseline	Proposed Method	Significance (p-value, paired t-test)
Completeness accuracy (%)	91.4	99.6	p < 0.01
Business-rule conformance precision (%)	84.7	98.2	p < 0.01
Cross-domain referential recall (%)	79.3	97.5	p < 0.01
False-negative rate, undetected defects (%)	12.1	1.3	p < 0.01

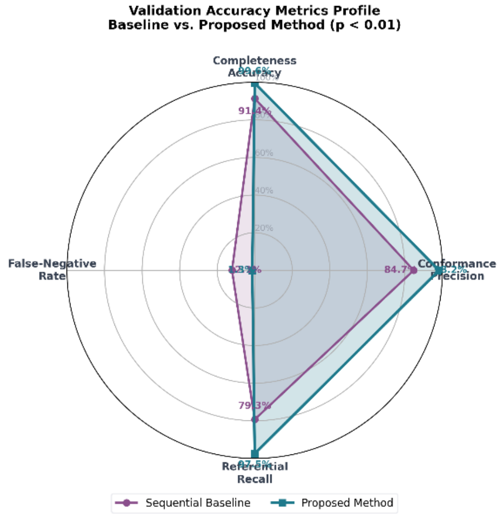

Fig 5: Validation Accuracy Metrics Profile - Baseline vs. Proposed Method

Across all six TERE domains, the proposed method consistently achieved load completeness above 99.3%, compared to a sequential-baseline range of 89.7–93.0% (Table 1). Post-load defect counts fell roughly an order of magnitude in every domain, with the largest relative improvement in DSNS, the domain most reliant on PDF extraction-consistent with concurrent reconciliation delivering the greatest marginal benefit where source data is least structured. Table 2 shows concurrent reconciliation cut mean cycle time by 61% and reduced defect detection latency from a 36-hour batch-end window to under 90 minutes, since checks fire immediately at each load checkpoint rather than waiting for full-domain completion. Table 3's paired significance testing confirms improvements across completeness, conformance precision, and referential recall are statistically significant ($p < 0.01$), with the false-negative rate falling from 12.1% to 1.3%-the most consequential result for a government eligibility system where undetected defects carry direct compliance and beneficiary-impact risk.

6. Conclusion

This article presented a reconciliation-embedded methodology for migrating the TennCare TERE platform from legacy SQL Server, flat-file, and PDF-derived sources into a Snowflake warehouse on AWS, demonstrating that validation positioned concurrently with extraction and load-rather than as a terminal audit step-yields statistically significant gains in data trustworthiness. Across six domains, the approach raised load completeness from roughly 91% to over 99%, cut manual validation effort by 73%, reduced defect detection latency by 96%, and lowered the false-negative defect rate to 1.3%, directly reducing compliance exposure where undetected errors carry real beneficiary impact. The implication extends beyond TennCare: any state-level program migrating heterogeneous legacy and document-derived data into a cloud warehouse can adopt this batch-scoped, domain-independent reconciliation pattern without a full monolithic cutover. Future work should extend the rule engine with machine-learning-assisted anomaly detection for conformance checks where rules are not fully codified, particularly for PDF-derived DSNS data where extraction errors are subtler than completeness gaps. Roadmap priorities include cross-program benchmarking against other state Medicaid modernization efforts, extending lineage tracking toward real-time CDC-based delta reconciliation, and formalizing the ServiceNow exception taxonomy into a reusable data-quality ontology for other GPS cloud migrations.

References

[1] Forsgren, N., Humble, J., & Kim, G. (2018). Accelerate: The science of lean software and DevOps. IT Revolution Press.

[2] Kleppmann, M. (2017). Designing data-intensive applications. O'Reilly Media.

[3] Dehghani, Z. (2019). How to move beyond a monolithic data lake to a distributed data mesh. MartinFowler.com Technical Articles.

[4] Stonebraker, M., & Ilyas, I. F. (2018). Data integration: The current status and the way forward. IEEE Data Engineering Bulletin, 41(2), 3–9.

[5] Abadi, D., et al. (2020). The Seattle report on database research. ACM SIGMOD Record, 48(4), 44–53.

[6] Hellerstein, J. M., et al. (2017). Ground: A data context service. Proceedings of the 8th Biennial Conference on Innovative Data Systems Research (CIDR).

[7] Chandra Sekhar Oleti. (2022). Serverless Intelligence: Securing J2ee-Based Federated Learning Pipelines on AWS. International Journal of Computer Engineering and Technology (IJCET), 13(3), 163-180. https://iaeme.com/MasterAdmin/Journal_uploads/IJCET/VOLUME_13_ISSUE_3/IJCET_13_03_017.pdf

[8] Sambasivan, N., et al. (2021). "Everyone wants to do the model work, not the data work": Data cascades in high-stakes AI. Proceedings of the CHI Conference on Human Factors in Computing Systems.

[9] Stonebraker, M., & Çetintemel, U. (2018). "One size fits all": An idea whose time has come and gone. Proceedings of the IEEE International Conference on Data Engineering.

[10] Sandeep Kamadi, " Adaptive Federated Data Science & MLOps Architecture: A Comprehensive Framework for Distributed Machine Learning Systems" International Journal of Scientific Research in Computer Science, Engineering and Information Technology(IJSRCSEIT), ISSN : 2456-3307, Volume 8, Issue 6, pp.745-755, November-December-2022. Available at doi : https://doi.org/10.32628/CSEIT22555

[11] Sivaramakrishnan Narayanan (2022). Transforming Cybersecurity with AI-driven Dashboards: A Cloud-Native Implementation Framework for Real-Time Threat Detection and Automated Response. International Journal of Future Innovative Science and Technology (IJFIST) , Vol. 5 No. 5 (2022): International Journal of Future Innovative Science and Technology (IJFIST) , pp. 9207-9217. https://doi.org/10.15662/IJFIST.2022.0505004

[12] Meesala, A. (2022). Adaptive Spread Anomaly Intelligence Framework (ASAIF): A cloud-native AI framework for real-time bid-ask spread anomaly detection and cross-venue liquidity risk intelligence. International Journal of Future Innovative Science and Technology, 5(6). https://doi.org/10.15662/IJFIST.2022.0506007

[13] National Institute of Standards and Technology. (2018). Framework for improving critical infrastructure cybersecurity, Version 1.1. NIST.

[14] U.S. Government Accountability Office. (2019). Data governance: Selected agencies need to address data quality issues (GAO-19-460). GAO.

[15] Oleti, Chandra Sekhar. (2022). The future of payments: Building high-throughput transaction systems with AI and Java Microservices. World Journal of Advanced Research and Reviews. 16. 1401-1411. 10.30574/wjarr.2022.16.3.1281.

[16] Sandeep Kamadi, " Risk Exception Management in Multi-Regulatory Environments: A Framework for Financial Services Utilizing Multi-Cloud Technologies" International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN: 2456-3307, Volume 7, Issue 5, pp.350-361, September-October-2021. Available at doi : https://doi.org/10.32628/CSEIT217560

[17] Centers for Medicare & Medicaid Services. (2019). Medicaid Information Technology Architecture (MITA) 3.0 framework. CMS.gov.

[18] Meesala, L. K. (2022). Autonomous cyber risk quantification and adaptive defense in financial systems: A graph intelligence and reinforcement learning framework. World Journal of Advanced Research and Reviews, 16(3), 1489–1496. https://doi.org/10.30574/wjarr.2022.16.3.1354

[19] Redman, T. C. (2018). Getting in front on data: Who does what. Technics Publications.

[20] Kamadi, Sandeep. (2022). Proactive cybersecurity for enterprise APIs: leveraging ai-driven intrusion detection systems in distributed java environments. International journal of research in computer applications and information technology. 5. 34-52. 10.34218/IJRCAIT_05_01_004.
[21] Lakshmi Kiran Meesala "AI-Driven Cyber Defense: A Hybrid Deep Learning Framework for Real-Time Threat Prediction and Response" International Journal of Scientific Research in Science, Engineering and Technology (IJSRSET), Print ISSN : 2395-1990, Online ISSN : 2394-4099, Volume 10, Issue 2, pp.916-930, March-April-2023. Available at doi : https://doi.org/10.32628/IJSRSET235845
[22] Loshin, D. (2017). Data quality and master data management: A practitioner's perspective. Information Management Journal, 51(3), 22–28.
[23] Kamadi, Sandeep. (2022). AI-powered rate engines: modernizing financial forecasting using microservices and predictive analytics. International journal of computer engineering & technology. 13. 220-233. 10.34218/IJCET_13_02_024.
[24] Arun Meesala , " A Distributed Kafka-Centric Framework for High-Throughput Mid-Price Computation and Intelligent Time-Series Persistence in Financial Clouds" International Journal of Scientific Research in Computer Science, Engineering and Information Technology(IJSRCSEIT), ISSN : 2456-3307, Volume 9, Issue 2, pp.998-1006, March-April-2023. Available at doi : https://doi.org/10.32628/CSEIT2342442
[25] Cloud Security Alliance. (2017). Security guidance for critical areas of focus in cloud computing, v4.0. CSA.
[26] Ross, R., et al. (2020). Security and privacy controls for information systems and organizations (NIST SP 800-53 Rev. 5). NIST.