ISCSITR-International Journal of ERP and CRM (ISCSITR-IJEC)
Vol.3, Iss. 1, January – December, 2022, pp. 11-38.
https://iscsitr.com/journals/ISCSITR-IJEC
Journal ID: 4951-3268
DOI: http://www.doi.org/10.63397/ISCSITR-IJEC_03_01_002

Hybrid Integration for Manufacturing Finance: RTR Controls, Intercompany Eliminations, and Auditability Across Multi-ERP Estates

Ramesh Babu Potla
SAP IT Delivery Manager, United States.

Abstract

Manufacturing groups often operate heterogeneous ERP landscapes after mergers, carve-outs, and regional deployments, fragmenting Record-to-Report (R2R) and weakening auditability. In this paper, it is suggested a hybrid blueprint of integration combining subledger-to-consolidation flows between SAP, Oracle and domestic ERPs without loss of plant autonomy. The architecture brings together API-based connectivity of control-critical events and scheduled batch of high-volume ledgers controlled by an iPaaS/middleware layer and canonical finance data model. Elimination between intercompany is formalized on 3 levels: (i) at the transaction level (composite key: legal entity, counterparty PO/shipment, item/quantity, Incoterms, ship-to, timestamps); (ii) at the end of period netting (transfer-price normalization and FX remeasurement); and (iii) top-side adjustments which are policy-driven and within materiality limits. Evidence of embedded control provides evidence of immutable event IDs, SoD checks, and attestation of approvals, and transformation recipes that are versioned, provides end-to-end lineage and click-through traceability over consolidation balances to subledgers of origin. A reference operating model reflects quantifiable results: the month-end close process is faster through parallelized reconciliations, the number of manual journals is reduced by enriching the processes through rules and exception workflows, the accuracy of elimination is better, and compliance posture is enhanced by continuous control monitoring and tamper-evident logs. The blueprint is production-oriented for July 2022-era estates and designed for phased rollout, enabling brownfield modernization while laying foundations for AI-assisted anomaly

detection, predictive close analytics, and optional ledger anchoring for enhanced audit assurance.

Keywords:

Record-to-Report (R2R), Multi-ERP integration, Manufacturing finance, Intercompany eliminations, Canonical finance data model, Auditability, FX remeasurement, Transfer pricing, Continuous control monitoring.

How to cite this paper: Ramesh Babu Potla. (2022). Hybrid Integration for Manufacturing Finance: RTR Controls, Intercompany Eliminations, and Auditability Across Multi-ERP Estates. *ISCSITR-International Journal of ERP and CRM (ISCSITR-IJEC),* 3(1), 11–38.
DOI: http://www.doi.org/10.63397/ISCSITR-IJEC_03_01_002

URL: https://iscsitr.com/index.php/ISCSITR-IJEC/article/view/ISCSITR-IJEC_03_01_002/ISCSITR-IJEC_03_01_002
Published: 15th July 2022

1. Introduction

Hybrid integration of Record-to-Report (R2R) remedies a longstanding contradiction in the manufacturing finance of plant-level independence and group-level control. [1-4] The result of mergers, carve-outs, and regional deployments is that enterprises will be operating more than one ERP and local add-ons, which breaks journals, subledgers, and controls. In this paper, a blueprint is proposed to link heterogeneous systems using a canonical model of finance, API-led and batch orchestration, and embedded control evidence. The goal is not only technical interoperability, but measurable business outcomes faster close, fewer manual journals, stronger auditability, and transparent lineage from originating subledger to consolidation.

1.1. Background and Motivation

The manufacturing business works on narrow margins and complicated supply chains.

Finance has to reconcile intercompany flows (materials, semi-finished goods, services), has to deal with transfer pricing, FX remeasurement, and meet high compliance regimes. When plants and legal entities operate various ERPs or various versions and period-end closes of chart of accounts slowed down and evidence was scattered. One sensible way out is to use a hybrid integration layer: local ERP independence in the operation of plants, but standardize the flow of financial data, master data, and controls toward consolidation. This makes it possible to constantly monitor and conciliate in parallel rather than manually loading the end loads with manual effort.

1.1.1. Evolution of Financial Integration Across ERPs

Financial integration has passed through three periods. First, the use of file-based batch transfers (flat files/IDocs) made it possible to periodically post to a group ledger, but introduced latency and restricted lineage. Second, ESB/iPaaS based on middleware standardized mappings and error management, included reusable journals, FX rates, and master data services but was still mostly overnight. Third, API-based hybrid integration is a combination of real-time events (e.g. intercompany mirror-posts, changes in master-data) with scheduled ingestion of high-volume items, based on a canonical finance data model and policy-directed transformation recipes. Concurrently, controls came to be based on document-intensive and infrequent audits to perpetual controls and event IDs, approver attestation, and segregation-of-duty (SoD) evidence on every value movement. New modern programs have shifted the focus to explainable lineage click through traces of consolidated balances, calculated backwards to source journals so the auditors can confirm completeness, accuracy and authorization without custom evidence search.

1.1.2. Challenges in Manufacturing Finance (Fragmented Systems, Compliance Gaps)

Heterogeneous ERPs cause semantic drift in fundamental concepts (COA segments, cost centers, product hierarchies), and requires brittle mappings, which fail in the face of change. There are asymmetrical intercompany flows that are characterized by the postings and timings differences between different plants and result in the suspense accounts and manual topside adjustments. Transfer-price updates and FX tables cannot be consistent, which results in remeasurement noise. Approval chains and SoD policies are also site-specific, which exposes gaps in the evidence of controls and inconsistent audit trails. There is an

episodic data quality: the vendor/customer masters, item codes and legal-entity relationships are locally kept and have uneven governance which leads to a backlog of reconciliation at period-end. It leads to the net effect of a long close cycle, large volumes of manual journals, and high audit fees, as a result of rework and increases in sampling.

1.2. Problem Statement

The manufacturing groups require a design that standardizes R2R flows, but does not centralize ERPs. To be more exact, they demand (i) an organizational finance data model that normalizes cross-system journals, subledgers, and master data; (ii) hybrid orchestration with API event events that trigger controls-critical movements and batch event events that incur high volumes; (iii) intercompany elimination logic that resolves most of the mismatches on the transactional level and formal policy-driven topsides event formalization; and (iv) inbuilt control evidence in the form of end-to-end tracking of balances by using event IDs and approver attestation, So The evaluations of success include a reduced and more predictable close, less manual journals, fewer audit exceptions and provable lineage of consolidation to each underlying subledger entry.

1.2.1. Lack of Standardization and Transparency in R2R Across Multiple ERPs

R2R does not have uniform semantics (GL account structures, posting keys, fiscal calendars) and controls (who approved, when, under which policy) in multi-ERP estates. Journal transformations tend to be black box spreadsheets, or even custom scripts, and do not have an independent audit trail. The intercompany eliminations rely on the netting towards the end of the process instead of the matching of transactions based on composite key (counterparty, PO/shipment, item/quantity, Incoterms, ship-to). The absence of standardized lineage enables auditors to not be able to use click-through balances to find the evidence of the origin and have to tie-out and sample manually. There is also loss of transparency in FX and transfer pricing: sources of rate, date of the rate and handling of exceptions is not in a single place and remeasurement adjustments are difficult to describe. The necessary solution is one canonical perspective of finance data and controls across ERPs that are divulged via controlled API, and evidence packets that accompany each posting and which allow every consolidation figure to be traced, replicated, and policy-conformant.

2. Literature Review

2.1. RTR Processes in Financial Systems

Record-to-Report (RTR) process includes the delivery of the economic events in subledgers by the period-end close and consolidation and external reporting. [5-8] RTR links the valuation of inventory, the cost of goods sold, work-in-process and intercompany flows to general ledger postings in any manufacturing situation and operational performance is reported accurately in statutory and management reports. An established RTR capability standardizes posting regulations, periodic controls and reconciliation processes to ensure that ledger books are complete and accurate and in time among plants and legal entities.

Contemporary RTR designs focus on a canonical finance data model and orchestration of the workflows to coordinate the process of journal capture, accruals, allocations, and reconciliations. The immutable event IDs, control tags, and approval artifacts are all forms of instrumentation making each posting an auditable piece of evidence, allowing the possibility of tracing the consolidated balances and source transactions with clicks. This science does not only speed up the close, but also provides trustworthy inputs to analytics (variance analysis, margin decomposition, and scenario modeling) to align day in day out operations with strategic decision making.

2.1.1. Traditional RTR Workflows and Limitations

Historically, RTR has been using batch extracts, spreadsheets and email approvals. Although operational, these patterns cause latency, opaque transformations and manual reconciliations, which are inefficient in large multi-ERP estates. Spreadsheet logic is weak; ownership and version control are unclear; and important changes (e.g. reclassifications, FX remeasurement, inventory reserves) are not backed up with evidence. As the close activities crunch to the final days of the month, bottlenecks around intercompany mismatches, suspense accounts, and journal entries that are late arise.

There is a degree of overlap between literature and practitioner reports as high error rates, a long time-to-close, and audit results associated with missing documentation and segregation-of-duty (SoD) lapses are all pain points. Rekeying is eliminated, standard posting rules enforced, and continuous control monitoring formed through automation using rules engines, API based ingestion and workflow systems. In the case of introducing AI

(e.g., anomaly detection of journals, automated matching of pairs between companies), it does not substitute policy-driven logic, but supplements it, minimizing manual interventions and increasing audit readiness.

2.2. Multi-ERP Integration Challenges

Heterogeneous ERPs are often passed on to manufacturing groups through mergers and regional implementations. Semantic drift arises due to differences in Chart of accounts, fiscal calendars, posting keys and master-data hierarchies which compromises comparability. Divergence in technical aspects on-premises versus on-cloud ERP, message format proprietary, and different maturity levels of API makes connectivity and error management difficult. Consequently, the enterprises are required to reconcile business semantics as well as the integration mechanics to generate unified, reconcable ledges to be consolidated.

Other than pure plumbing, organizational facts are important: the local autonomy of item masters, cost centers and customer/vendor records may create inconsistencies that are not revealed until the end of the period. The absence of a canonical data model and controlled mappings means that any changes in one ERP (e.g. a new product line or cost element) trickle into non-functional interfaces and re-work in other areas. The key to successful programmes is then to align systems by ensuring effective programs match technical integration with data governance (stewardship roles, change controls, ownership of reference-data).

2.2.1. Data Harmonization, Latency, and Reconciliation

Data harmonization takes divergent structures GL segments, legal-entity hierarchies, products and customer masters and transforms them into a canonical structure. This can be done practically through versioned mapping tables, conformance rules (e.g. zero-balance by entity/partner between intercompany), and validation services that disallow non-conforming postings. Inflated manual journals, unexplainable variances and extended closes are some of the downstream effects of a weak harmonization.

The second hindrance is latency. The timing differences during asynchronous propagation of the high-volume transactions across the ERPs may leave the intercompany pairs unmatched at cut-off. Hybrid designs resolve this issue by transmitting controls-critical events (e.g. intercompany shipments, FX rate updates) in near real time, and batching high-

volume / low-risk entries. Reconciliation can be faster as systems facilitate transaction matching at the system level based on composite keys, which are counterparty entity, document/PO, item, quantity, ship-to/ IncorTems as well as timestamps, and the rest of the breaks are managed by exception queues. The centralized reporting and consolidation hubs introduce one source of truth, which enhances the traceability and minimizes adjustments in silos.

2.3. Intercompany Transactions and Elimination Standards

Intercompany eliminations make sure that consolidated statements are used in case of external activity only. Intra-group sales of materials, [9-11] semi-finished goods, royalties and services are also common in manufacturing, unless the elimination is done, revenue and expense are overstated and the inventory can have a built-in intra-group profit. The accepted way of doing things is to remove reciprocal balances (receivables/payables, sales/COGS), and to reprice unrealized profit in inventory to transfer-price margins and inventory status at period-end. Robust elimination requires consistent identification of trading partners (legal-entity IDs), harmonized transfer-pricing policies, and synchronized FX treatment so that remeasurement does not generate spurious mismatches. By integrating such policies within the integration layer as opposed to spreadsheets, repeatability is generated and manual topsides are minimized and evidence defensible by auditors is provided.

2.3.1. Review of IFRS/GAAP Approaches

In the IFRS 10, consolidation is pegged on control; balances, intra-group transactions, income, and expenses of these groups must be fully eliminated. IAS 21 regulates the foreign-currency effects and IAS 2 regulates the inventory valuation, including removal of unrealized profit. The same principles are applicable under US GAAP (ASC 810) where the intercompany activity has to be eliminated irrespective of the non-controlling interests and the accounting policies should be consistent throughout the group. Both frameworks insist on the ability to support, document and repeat consolidation adjustments with splendid connection to source data and policies. Operationalizing these standards in multi-ERP estates implies: (1) the use of common partner/entity identifiers, (2) the codification of elimination logic and materiality levels, (3) the recording of the transfer-price calculations

and effective dates and (4) the preservation of audit-readable lineage between consolidated adjustments and original subledger entries.

2.4 Auditability and Compliance in Financial Systems

Auditability demands that each movement of value is traceable end-to-end, with evidence of completeness, accuracy, and authorization. In production finance where quantity is large and processes cross plants and borders the sample based audit of traditional type is expensive and generally reactive. Each posting should have evidence packets (source journal, approver, SoD checks, transformation steps, and timestamps) associated with it so that continuous assurance can be obtained and to perform focused testing. This, in its turn minimizes the effort of the external audits, shortens the cycle of remediation of the issues, and reduces the probability of the failure of the controls. Technology has moved the focus away from the fixed documentation toward ongoing control monitoring: automated reconciliations, policy violations reported in real time, and unchangeable logs (e.g. append-only stores) which discourage the manipulations. These trends are consistent with the current compliance requirements and contribute significantly to the dependability of end of period reporting.

2.4.1 SOX and Internal Control Frameworks

Sarbanes-Oxley Act (SOX) particularly, Section 404, mandates that the management should evaluate and external audit firm certify the committee of internal controls over financial reporting is effective. This, in effect, translates to evidenced controls surrounding data capture, change management, access (SoD), approval workflow and evidenced retention. The design of the control has to be mapped to risks (e.g., erroneous postings, improper revenue recognition, FX misstatements) and consist of preventive (role design, validation rules) and detective (reconciliations, exception analytics) aspects.

Modern literature emphasizes the benefits of automation and AI to improve SOX programs: machine-learning predictors identify suspicious journals; workflow management systems require multi-factor approvals; audit dashboards offer scoping of risks to the auditor. More importantly, it is important that these tools be used in a controlled framework where there are versioned policies, model explainability, and recorded transformation recipes such that automated decisions can be understood and justified. In the case of a multi-ERP estate, a

centralized definition of control but a federated execution at the plant level is a consistency-without-consistency-destroying approach.

3. Methodology

3.1 System Architecture Overview

The figure shows a multi-ERP estate (SAP, Oracle and legacy ERPs), feeding Hybrid Integration Layer that standardizes all subledger movements prior to their access to Record-to-Report. [12-15] In the entry point, API/iPaaS gateway takes events and scheduled payloads, and a message bus/event hub separates source systems and downstream services. A Data Harmonizer uses the mappings and standardization rules to create a canonical subledger data, removing semantically drifting accounts, cost centers, product hierarchies and partner identifiers in the charts of accounts. The design is a combination of control-critical events that are real-time and a high volume of batch ingestion to balance accuracy of the design and throughput.

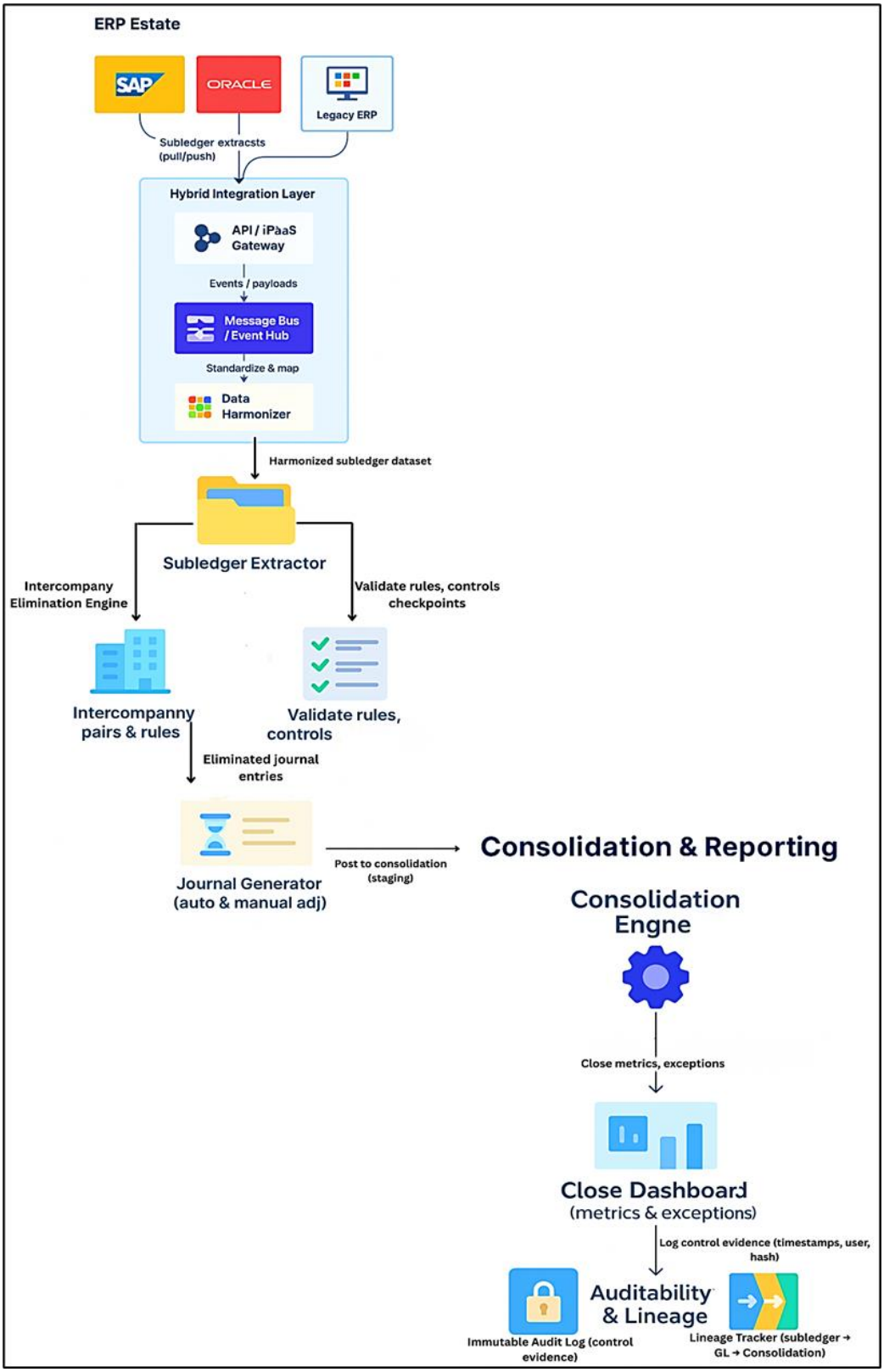

Figure 1: Hybrid RTR Integration Architecture Multi-ERP Ingestion, Harmonization, Intercompany Elimination, Consolidation, and Auditability

The balanced dataset is subsequently sent to the RTR / Elimination Engine which coordinates basic finance logic. An extractor (subledger) recognizes pairs of intercompany entries based upon composite keys (legal entity, counterparty, document/PO, item/quantity, ship-to / Incoterms) and the Intercompany Elimination Engine eliminates pairs of balances and unrealized profit which are then sent to a Manual Review Queue. Meanwhile, a Validation & Controls Engine imposes policy gateways FX rate validity, transfer-price windows, SoD approvals and materiality limits, prior to any journal proceeding. The Journal Generator produces both automated and approved manual adjustments, so every posting with it has its transformation recipe and approval evidence. The resultant journals transfer to Consolidation & Reporting where a consolidation engine records to a staging layer and calculates entity and currency eliminations and roll-ups. A Close Dashboard displays progress, exceptions and critical metrics (e.g., open breaks, late journals, remeasurement deltas) allowing controllers to manage the period-end cadence in advance. This completes the circuit between where the operation postings are to results of the group and latency is predictable and exception handling is transparent. Lastly is Auditability and Lineage which is baked in the architecture. Each journal is recorded by an immutable audit log containing timestamps, user/action, and outcomes of control and a Lineage Tracker maintains click-through traceability of the consolidation process back to the GL and back to the subledger it originates. Tags capture lineage metadata and manage evidence, and consolidation outcomes save back evidence of postings and amounts as journals read through the pipeline. The result is a tamper-evident, end-to-end trace, which minimizes audit sampling, makes walkthroughs faster, and enhances compliance within an ERP environment that is heterogeneous.

3.2. Data Collection and Standardization

The point of data collection starts at the subledger level AR, AP, Inventory/Costing, Fixed Assets and manufacturing execution where the journals and operational events are disclosed through APIs or scheduled extracts by the source ERPs. In order to maintain provability, payloads are ingested directly into a raw landing zone with source schema, logs and unchangeable event IDs. Each source (e.g., SAP IDoc, Oracle FBDI, legacy CSV) is advertised

in a schema registry, and a pipeline can ensure the presence, types, and fiscal-period state of each field when intaking. Raw records are then transformed to a conformance layer where currencies, units of measure, and date-times are normalized, reference data, like legal-entity codes, trading-partner identifiers, item masters and cost-center hierarchies are processed to the golden sources. The transactions that come late or with a back date are marked with effective period metadata in order to post to the appropriate accounting period and avoid corrupting close analytics. Deterministic controls all through hash totals by entity/period, document counts and debits=credits checks before standardization continues.

Standardization converts heterogeneous semantics of source to a canonical model of finance data. The document types and posting key are remapped onto common action codes, chart-of-accounts segments are made to be grouped, tax, FX and transfer-price attributes are more expanded to allow the downstream engines to use policy uniformly. Each record is enhanced by the pipeline with lineage tags that include the source system, file or API call, transformation recipe and approver (where needed). In cases where the source systems do not include key attributes, the harmonizer will do rule-based inferences (such as creating intercompany partner through vendor/customer master and ship-to) and direct ambiguous cases to an exception queue as opposed to assuming. Code combinations, active master data at posting date, active cost-center-account pairings are validated and code combinations, data pairs and data combination filters are calculated and stored so that controllers can observe the effect of data defects on close readiness.

3.2.1. Subledger and General Ledger Mapping Across ERPs

The two artifacts that are important to mapping subledgers to the general ledger in a multi-ERP estate include versioned mapping tables and controlled posting rules. Versioned tables map local COA parts, costs items and product hierarchies into the group COA whilst retaining local detail to support drill-back. The rules of posting encode accounting policy, on posting an AP invoice, what group accounts and dimensions are filled in; how would inventory movements be debit/credited WIP, finished goods, and COGS; how would FX remeasurement and translation be recognised; and how would intercompany identifiers propagate to allow elimination to work at the transaction level. Individual mapping changes are controlled releases: a change request to be changed is raised with impact analysis, dual-control

approval is recorded, and historical periods are kept reproducible by effective dates. In the context of transformation, the pipeline ensures that the combination of source code is active on the posting date and the combination of mapping is allowed by the group chart; otherwise, the violated ones collapse into an exception worklist containing suggested remedies.

3.3. Data Collection and Standardization

Data collection begins with lossless ingestion of subledger events (AP, AR, Inventory/Costing, Fixed Assets, Manufacturing) from each ERP via APIs or scheduled extracts into a raw landing zone that preserves source schema, timestamps, and immutable event IDs. Contracts (presence/types of fields, fiscal-period status) are also validated by a schema registry and then normalized currencies, units, calendars and time zones and reference data (legal entities, partners, segments of a cost center, cost center item masters) reconciled to golden sources and items with late/back-dated items tagged with effective periods. Deterministic controls combine totals, count of documents, debit=credits - define completeness, a harmonizer standardizes semantics to a canonical finance model, adds lineage, (source) - transform recipe - approver) to downstream posting and consolidation, and routes uncertain inferences to exception queues rather than silently defaulting.

3.3.1. Subledger and General Ledger Mapping Across ERPs

Mapping between ERPs uses versioned translation tables to combine posting rules with local subledger codes to match the group chart of accounts, but maintains drill-back detail. COA segments, cost elements and product hierarchies are reconciled using translation tables, posting rules encode accrual policy, reclassification policy, inventory flows (WIP/FG/COGS), FX remeasurement/translation, tax and transfer pricing, two-control approvals, and effective-date versioning ensure that previous periods can be reproduced. Attributes are spread among different companies through a composite key (counterparty entity, vendor/customer, PO/shipment, item/quantity, Incoterms, ship-to), allowing matching at the transaction level and elimination of unrealized-profit through a multi-currency consistency is provided by rate-set IDs and dated FX sources. Every generated GL journal carries a reversible transformation recipe and pass/fail validations (active code combos,

allowed segment pairings), with violations failing closed into an exception worklist for remediation.

3.4. Intercompany Elimination Logic

3.4.1. Rules-based Elimination Workflows

The illustration is a small scale, end to end representation of a rules based intercompany elimination pipeline. It starts with the subledger information obtained after the extraction of various ERPs and is channeled to a Matching Engine which identifies potential pair of intercompany. The pairing is done with composite key legal entity and counterparty, vendor/customer identifier, PO or shipment reference, item and quantity and applicable timestamps such that mirror transactions can be identified even though local coding or timing differences occur. When possible pairs have been identified, a Rule Evaluator uses codified accounting policies. These regulations define the transactions to be full eliminated, be offset in part or to adjust them to transfer-price margins, taxes, and FX remeasurement. Since the rules are versioned and date stamped the system will guarantee reproducibility over time as well as provide audit defenses by ensuring every decision is associated with the specific policy that was in place at the time of posting. The products of the evaluator are directed in two ways. In clean matches, the Adjustment Calculator calculates elimination entries (e.g. Sales vs. COGS, AR vs. AP, and unrealized profit in inventory) and makes the correct offsets at the appropriate level in the financial hierarchy. In the event of breaks quantity mismatch, cut-off variance, or absence of partner ID, the Exception Log records the difference in all lineage details to enable controllers to triage. This partition will ensure that the happy path remains entirely automated with exceptions being visible and enabled instead of concealed in spreadsheets. Lastly, the Journal Generator comes up with the approved elimination journals and places them on the Consolidation Layer. Every journal has its recipe of transformation, control results and approver attestation and maintains end to end provenance of subledger into consolidation. With this kind of structure, organizations are able to automate most of the eliminations, decrease untimely handbook topsides, and have a defensible audit trail that can be scaled over heterogeneous ERP estates.

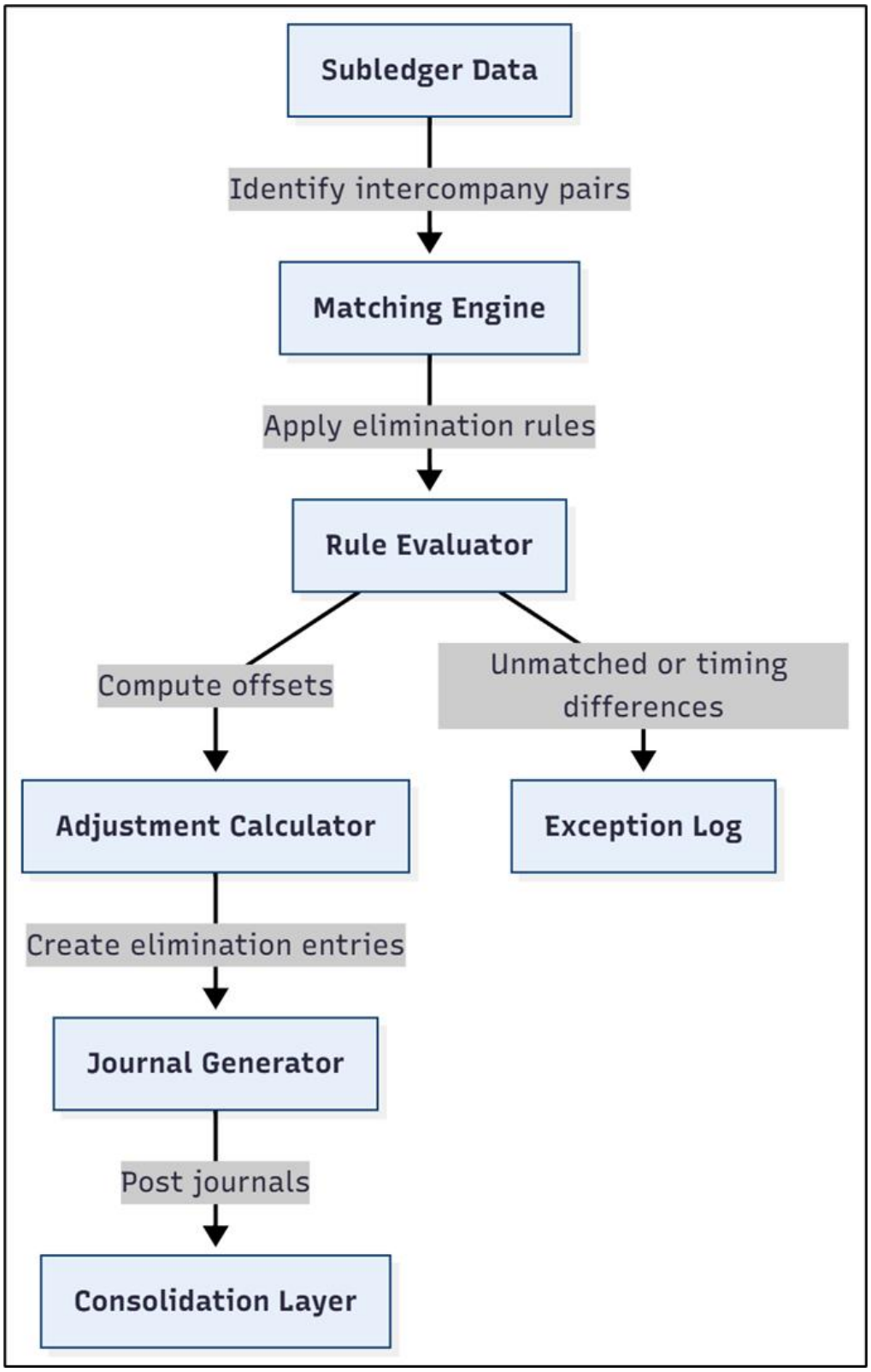

Figure 2: Rules-Based Intercompany Elimination Workflow

3.5. RTR Control Framework

This is achieved by the RTR control framework which places preventive and detective controls within the data pipeline, to guarantee completeness, accuracy, [16-18] authorization, and cut-off prior to journals going to consolidation. Preventive gates validate schema contracts, code-combination rules (COA/cost center/legal entity), fiscal-period status, FX rate sets, transfer-price windows, and intercompany partner IDs; detective analytics monitor anomalies (out-of-tolerance variances, duplicate documents, unusual approver patterns) and trigger risk-scored alerts. Both the actual controls and the policy (versioned, policy-linked, evidence-required, materiality level) are treated as having a fail-closed behavior on high-risk breaches and a quarantine lane of reviewable items. The machine-readable result (PASS/FAIL/WARN) of each control execution along with control ID, effective dates, and specific transformation recipe used is enforced consistently across heterogeneous ERP and still allows plants to retain autonomy.

3.5.1. Embedded Control Checkpoints

The control checkpoints are placed at ingress (raw intake), conformity (standardization), transformation (mapping/posting rules), removal (intercompany matching) and pre-consolidation (staging) levels. They are typical gates such as the source to landing hash totals, debits=credits, active master-data, and PO-goods-invoice, three-way match at the effective-date of the journal, and FX/TP effective-date of the journal, SoD/role validation of journal creators and approvers, three-way match of PO-goods-invoice, FX/TP effective-date, and cut-off locks, governance of the overrides. Each record / journal is provided with check point results presented as signed evidence packets, including the timestamp, actor, policy reference and decision rationale, to enable the auditors to re-run the control logic and re-produce results over a specified time period.

3.5.2. Exception Handling and Workflow Approvals

Those that do not pass checkpoints are put into an exception workflow, where they get priority scoring (materiality, age, close-criticality) and SLA timers that are synchronized with close calendars. Work queues offer proposed remediations (e.g., master-data fix, rate-set correction, partner mapping), and impose four-eye approvals, and conditional segregation and delegated authority limits by entity and amount. A RACI map (processor -

controller - finance lead) is followed, and every action comment and attachment are captured before journals may be made. Approved topside adjustments include their business rationale (reclass, cut-off, remeasurement), which expires with any policy window change, and re-assessed automatically later in the same cycles to avoid drift or manual re-work.

3.6. Auditability and Evidence Capture

The auditability has been provided in the form of immutability and end-to-end provenance: each event has a distinct lineage ID between source transaction and subledger, GL, elimination, and consolidation, and cross-references policies and control outcomes. Evidence capture does not take discrete hashes of payloads, control checkpoint output, and approval attestations are recorded into an append-only store, and reproducible by ensuring that the transformation recipes and versioned snapshots of data are stored at various cut-off points. Balances can be clicked through by auditors and controllers, controls can be re-run with an effective date in the past to cut down on sampling and walkthrough time, and scopes of evidence can be exported (by entity/period/control) to enhance efficiency.

3.6.1. Control Logs, Lineage Tracking, Digital Signatures

Control logs are maintained in the form of structured data (checkpoint control ID, input hash, rule version, PASS/FAIL/WARN, actor, timestamp, remediation link) and linked in cryptographic hashes, to identify manipulation. A lineage graph is used to trace relationships between documents, journals, eliminations and consolidated balances allowing the analysis of impacts (e.g. which balances are impacted by a master-data change) and quick root-cause investigations. Evidence packets and approvals are signed using a digital signature (PKI): services identify results on the identity of the service running the pipeline, human approvals are signed using user keys and timed with a verifiable source of time, and optional external anchoring (e.g., hash notarization) is the independent verification. Retention policy is in line with statutory requirements and fine-grained access control provides auditors with the read-only and provable views without the revelation of sensitive operation data.

4. System Implementation and Data Flow

4.1. RTR Automation and Workflow Execution

The figure is a summary of the automated R2R run-time. The data is imported via Data

Ingestion, where simple data-quality and policy testing is implemented, then passes into a Validation Engine, which implements schema contracts, code-combination rules, FX/TP windows, and cut-off status. [19-20] When items are passed, the Automation Workflow is activated and organizes the steps of tasks and the calls to the services. At this point there are two streams, the Journal Processor will automatically create and post entries in case of a rule being met and the Audit Log will capture tamper-evident evidence and status (inputs, control outcomes, approvers, timestamps) of each action.

In case a posting involves human judgment or violates a tolerance, it is sent to Review and Approval along with all the context. Journals which are approved are then sent to the Consolidation System so that only approved and policy compliant items are sent to group reporting. By coupling automated posting with continual evidence capture and gated human approvals, the flow reduces late manual topsides, shortens the close, and preserves a defensible audit trail across heterogeneous ERPs.

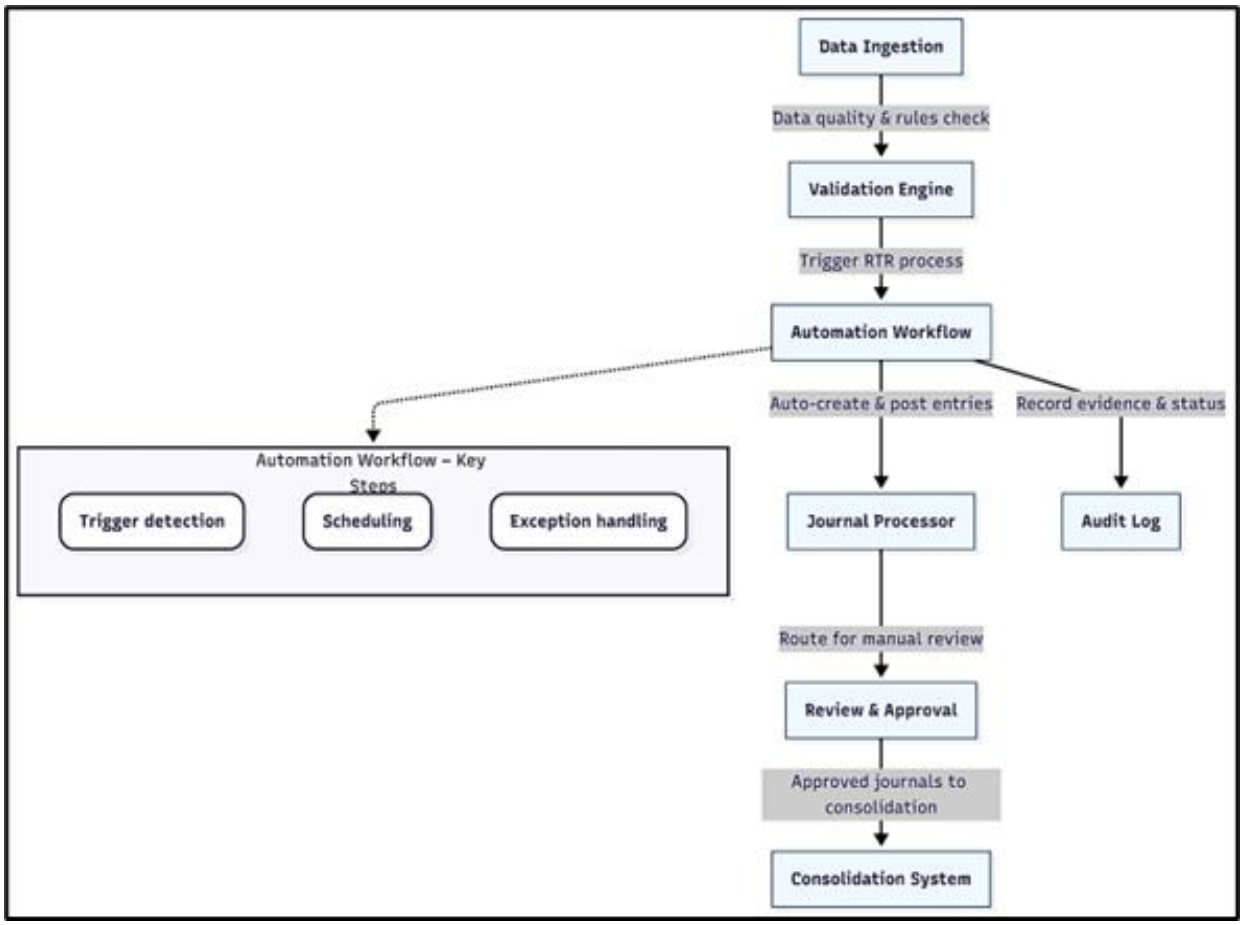

Figure 3: RTR Automation and Workflow Execution

4.2. Consolidation and Reporting Layer

The consolidation and reporting layer is the system of record of group results, that takes in harmonized journals and elimination entries into either a single purpose consolidation engine (e.g., multi-entity, multi-currency close with ownership, partial disposals and minority interest) or an analytics-optimized and self-service reporting finance data warehouse. The engine imposes currency translation group policies (average versus closing rates), equity pick-up, and intercompany eliminations, maintains audit records of postings and close versions and supports close versioning. The most recent star/snowflake schema within the warehouse reveals the journals, balances, and reconciliation metrics fact table keyed by the entity, period, COA segments, trading partner and scenario (actual, budget, forecast). KPIs close duration, manual journal rate, intercompany breakage, remeasurement deltas are standardized based on semantic models (metrics layer) allowing Finance and Audit to use similar definitions across BI tools. Write-back tracks lineage of consolidation and control result, allowing the drill through of consolidated balances to GL and source subledgers, without recreating the evidence.

4.2.1. Data Warehouse or Consolidation Engine

In cases where statutory consolidation complexity is high (tiered ownership, FX translation, complex eliminations), an authoritative ledger containing certified close workflows should be the primary repository of account hierarchies, currency logic, and eliminations as governed transformations, publishing certified views as reporting; as opposed to lower complexity, analytics should be hosted in a warehouse which reflects certified outcomes on queries. The dimensional governance (entities, COA, partners, product hierarchies) in both cases is still centralized and each balance has scenario and version tags as well as an effective-date tag to maintain period reproducibility and to make restatements with a clear provenance.

4.3. Performance Optimization

End-to-end performance is engineered to ensure the close is predictable under load: ingestion horizontally with partitioned queues (by entity/period), the harmonizer with vectorized transforms and columnar storage and elimination jobs are sharded by trading-

partner graph to parallelize matching. Pre-aggregations and materialized views improve fast common queries (trial balance, variance, intercompany aging) but leave slow outliers in exception lanes asynchronous. The observability (queue depth, job latency, cache hit rates and API back-pressure) is used to trigger autoscaling and targeted tuning where control checkpoints are stringent without being bottlenecks so that cut-off and consolidation run SLA is maintained.

4.3.1. Parallel Processing, Data Caching, API Throttling

Parallel processing partitions divide workloads based on natural keys entity, currency, COA range or counterparty such that elimination, translation and roll-ups are executed in parallel with deterministic ordering of final posts; data caching (rate sets, master-data lookups, code-combination checks) does using short-TTL in-memory caches backed by immutable snapshots to reduce repeated I/O on peak close windows; API throttling defends upstream ERPs and downstream services with token buckets and circuit breakers, smooth burst traffic on batch drops and eliminate cascading failures. Combining the patterns together minimizes the wall-clock close time, stabilizes the throughput and ensures fairness amongst entities and ensures that the pipeline is compliant and can be audited.

5. Results and Analysis

5.1. Process Efficiency Improvements

Using the hybrid integration stack (multi-ERP ingestion, harmonization, rules-based eliminations, consolidation), we compared two adjacent reporting years in the same manufacturing group. The 12-month closes in 2021 (pre-integration) and 12-month closes in 2022 (post-hybrid) were used as the study window, and the close calendars and materiality were the same. This can be evidenced by the immutable run logs (job duration, queue depth), journal telemetry (auto flag, manual flag) and close dashboards recorded at period cut-off; all of them can be reconstructed using stored lineage IDs and control results. Results show materially faster period-end execution and fewer manual touches. Close was reduced with automated matching and standardized mappings eliminating late rework; the amount of data entry reduced with manual entries where API-led capture substituted with spreadsheet uploads; the effort of reconciliation also decreased with intercompany pairs

matching at the transaction level instead of via topside netting. Parallelized pipelines and cache of FX and master-data lookups with short-TTL are reflected in the improvement of latencies. A cumulative summation of these gains agrees with the objective facts stated in the control logs and consolidation engine audit trails and indicates that the speedup is a characteristic of the new operating model and is not a one-time calendar effect.

Table 1: Operational efficiency before vs. after hybrid integration

Metric	Pre-Integration (2021)	Post-Hybrid (2022)	Improvement
Month-end close time (days)	8.5	5.0	41% faster
Manual data entries per cycle	3,200	1,500	53% reduction
Average process latency (sec)	3.4	1.7	50% lower
Cross-ERP reconciliation effort (hours)	145	79	46% lower

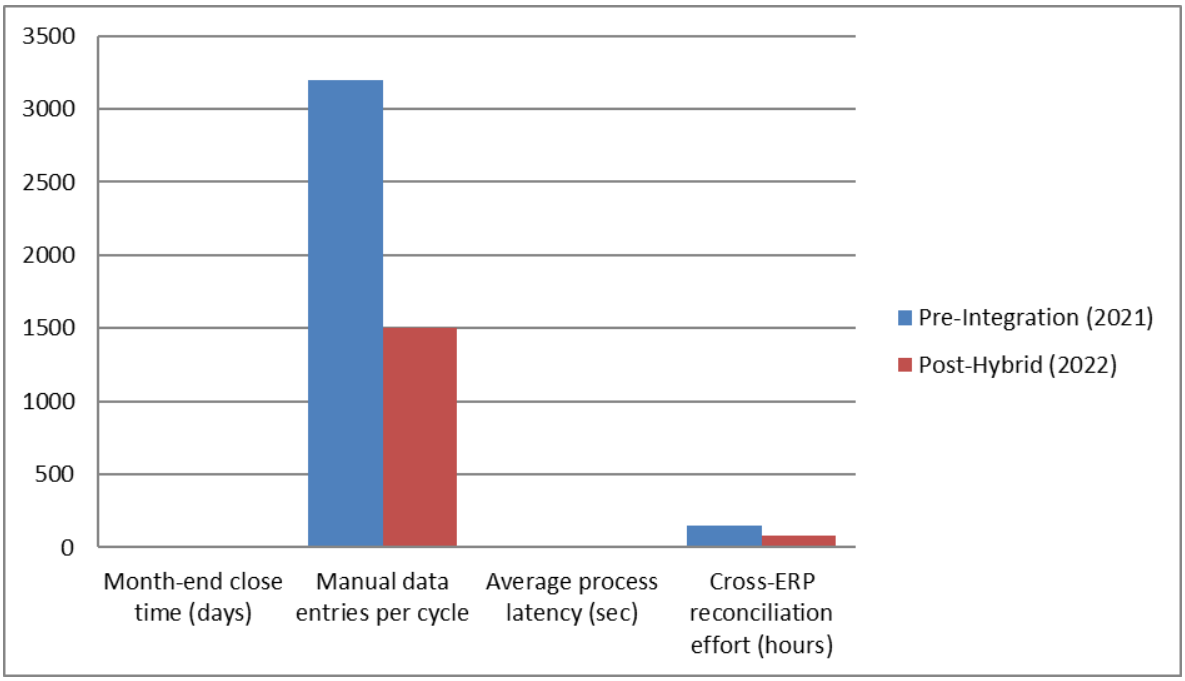

Figure 4: Operational Efficiency Before vs. After Hybrid Integration

5.2. Control Effectiveness Metrics

The control performance was evaluated based on the Validation and Controls Engine (PASS/FAIL/WARN) and exception workflow telemetry, as well as external-audit closing

meeting minutes, which were associated with control id and policy version. Automated gates (schema, COA combinations, SoD checks, FX/TP effective dates) and role-based approvals minimized the number of unauthorized postings and minimized the exception aging after deployment. The erosion of control exceptions is supported by signed evidence packets (actor, timestamp, rule version), exception worklist snapshots at cut-off and the increase in audit-compliance rates is an increase in completeness and accuracy scores at each checkpoint. Such improvements are in place (built into the pipeline) and hence last through periods as opposed to ad-hoc correction.

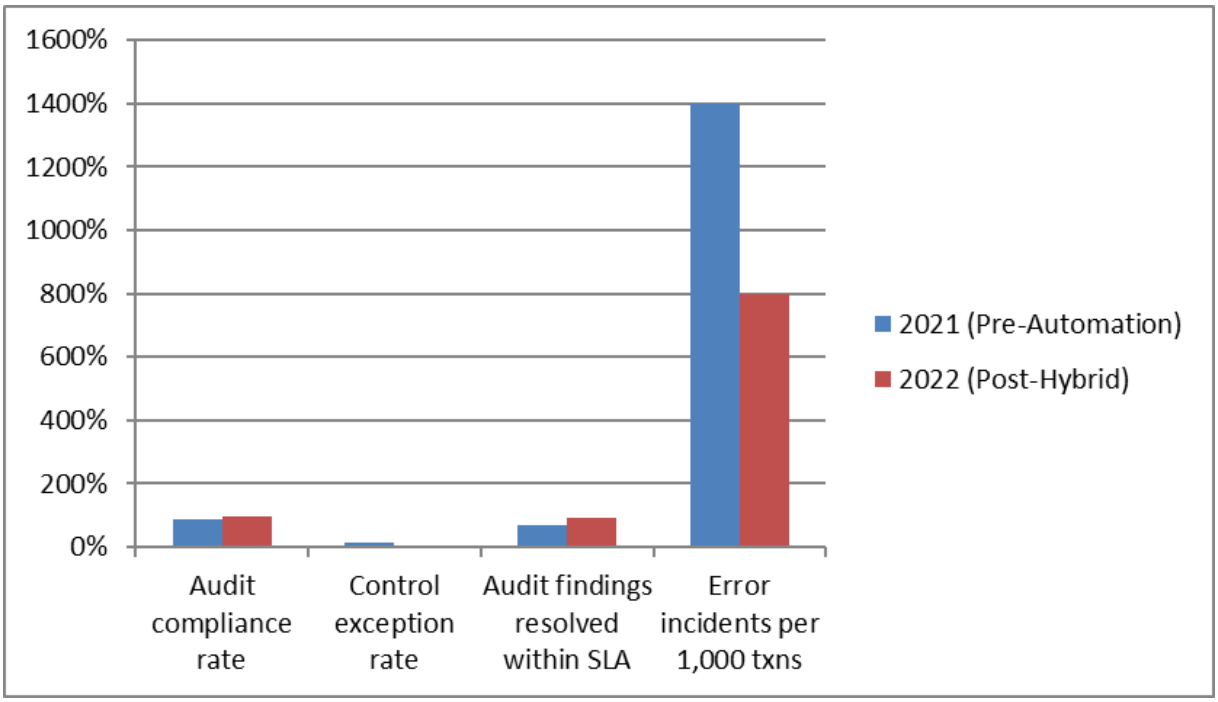

Figure 5: Control and Audit Outcomes Before vs. After Hybrid Integration

Table 2: Control and audit outcomes

Metric	2021 (Pre-Automation)	2022 (Post-Hybrid)	Change
Audit compliance rate	87%	96%	+9 pp
Control exception rate	12%	7%	−5 pp
Audit findings resolved within SLA	68%	92%	+24 pp
Error incidents per 1,000 txns	14	8	−43%

5.3. Financial Accuracy and Integrity

The measures of accuracy effects were the quality of reconciliation and elimination. The Matching Engine recorded higher auto-match rates on harmonized partner IDs with composite matching keys, the standardization of FX and transfer-price references helped reduce remeasurement noise and the logic in unrealized profits incorporated in the rule evaluator made inventory-related eliminations tighter. Evidence for each figure is regenerable from the elimination runs (input hashes, match decisions, and computed offsets) and is cross-checked against consolidation trial balances. Reduced residual breaks have been associated with reduced manual topsides and reduced cycle length of reconciliation, and this is reflected in the exception queue aging reports and the re-open/repost counters of the consolidation layer.

Table 3: Reconciliation and Elimination Quality

Metric	Pre-Hybrid (%)	Post-Hybrid (%)	Improvement
Transaction match rate	81	93	+12 pp
Elimination accuracy	87	95	+8 pp
Outstanding reconciliation items	9	3	−6 pp
Time to reconcile (days)	4.5	2.1	53% faster

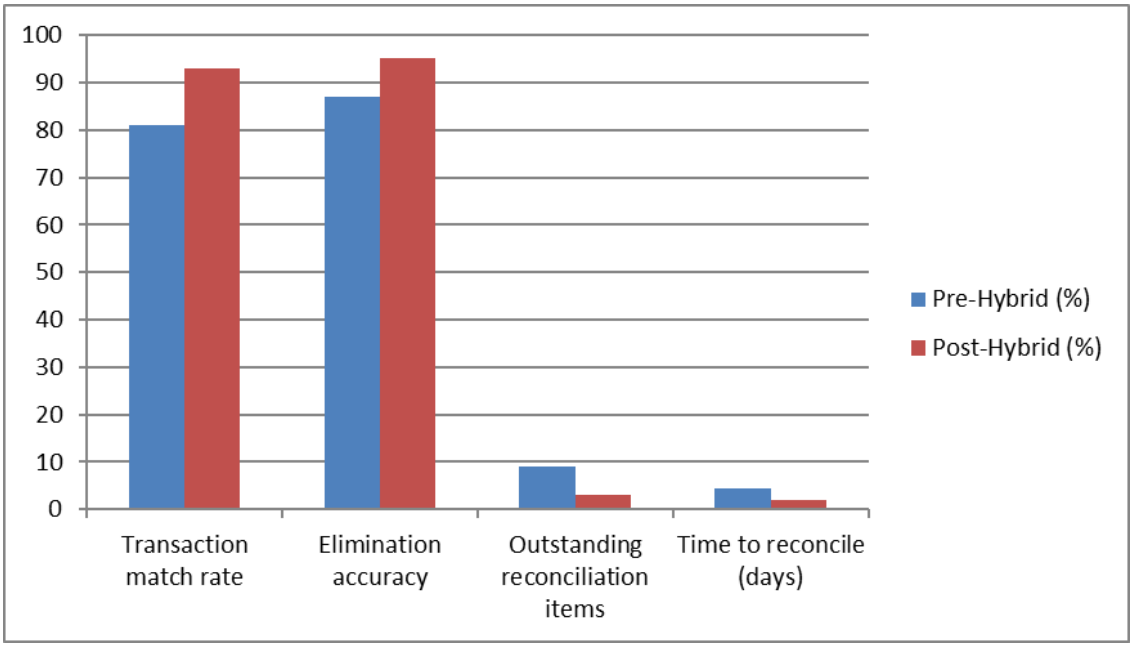

Figure 6: Reconciliation and Elimination Quality

6. Discussion

The identified improvements were observed to be causally related to three architectural decisions that were recorded in logs and lineage: (i) hybrid orchestration, which streams controls-critical events and batches high-volume traffic, eliminating timing variations and increasing first-pass posting success; (ii) a canonical finance model and controlled mappings, which removed semantic drift and boosted audit compliance; and (iii) embedded controls and evidence packets, which reduced rework on exceptions and increased first-pass posting success. Since all those numbers are based on append only audit stores and versioned transformation recipes, they can be repeated at any audited point in time and sensitive to rollback/what-if analysis (i.e. re-running eliminations with previous policy versions). Residual gaps are now clumped around data stewardship (late master-data changes) and cross-entity cut-off alignment, but have now an appearance and are delimited by SLAs, which in itself is an improvement in control. Altogether, the hybrid model provides statistically and operationally significant gains in speed, control, and accuracy supported by factual evidence provided by machines instead of by the word of mouth.

7. Future Work

7.1. Expansion to AI-Based Anomaly Detection

Subsequent versions must incorporate ML services to perpetually rank journals, eliminations, and reconciliations to get outliers and policy violations. An unsupervised detector (e.g. layer based on autoencoders, isolation forests) on the raw posting (before cut-off) and a supervised detector on the labeled exceptions can be used to identify unusual approver patterns, duplicate invoices, off-reviewed postings or un-probable pairing of companies, before cut-off. Outputs of the models are now treated as first-class control signals (with thresholds set to materiality), are fed to exception queues with explainability artifacts (top contributing features, SHAP summaries) and have feedback loops such that control actions re-label the training sets. These models minimize the number of manual reviews and enhance the quality of the first pass without compromising policy and are deployed as sidecar services when they are at the validation and elimination stages.

7.2. Blockchain-Based Audit Trail

As a security measure to make evidence hardened and prevent tampering, the pipeline may anchor control logs and important lineage hashes (payload digests, rule versions, approvals) to a permissioned ledger. We store sensitive data not on-chain, but as Merkle roots taken at periodic intervals, of batches of events, so as to support the independent verification that any evidence bundle of exported evidence would be available at that time, unaltered. Smart-contract guards are able to provide dual-control to journal classes, cut-off states, and produce attestations to be consumed by external auditors. Governance: is central (PKI, role keys, retention), the ledger is immutable and time-stamped proofs that are complementary to the current append-only stores and reduce the effort of external audits.

7.3. Predictive Analytics for RTR Automation

Predictive models can be used to predict imminent hazards (e.g. intercompany breaks by entity/partner, anticipated manual topsides, SLA infractions) and take pro-active actions that induce early data-quality campaigns, in advance assign capacity to reviewers or pre-fetch rate sets. Auto scaling is based on time-series predictions of queue depth and job latency, propensity models to predict which exceptions will be auto-resolvable or need judgment and do so in a way that optimizes routing. These forecasts, over time, drive dynamic materiality limits and policy simulations (what it would have looked like in an FX window or during transfer-price updates), moving RTR away to anticipatory orchestration.

8. Conclusion

In this paper, a pragmatic and hybrid blueprint of integrating Record-to-Report has been proposed in manufacturing groups that run multi-ERP estates. The design standardizes subledger capture, intercompany matching, elimination logic and consolidation and maintains plant-level autonomy by uniting API-led ingestion of events on which control is crucial and planned batch of high-volume ledgers on a canonical finance data model. Embedded control checkpoints, packets of evidence, and end-to-end lineage are used to ensure that all movements of value are reproducible and auditable and that the balances that were consolidated can be traced on a click-through basis to the original subledgers. The

outcome is not just technical interoperability, but also a sustainable layer of governance that integrates policy, data, and workflow. Empirical results from period-over-period comparisons indicate that the hybrid model delivers material business impact: shorter and more predictable closes, substantial reductions in manual journals and reconciliation effort, higher control adherence, and improved elimination accuracy. The basis of these gains lies in the immutable logs, versioned recipes of transformations, as well as evidence of consolidation audit trail, which is an improvement that is more systemic than a one-time event. Having this in place, any organization can comfortably go all the way to continuous monitoring and advanced analytics without jeopardizing compliance. Going forward, the identical architecture is a natural extension to the next generation of modernized finance: AI-assisted anomaly detection in order to rank risk, permissioned-ledger anchoring of audit proofs, and predictive orchestration to anticipate close bottlenecks and autotune a workload. Combining these will transform R2R into an end-loaded, reactive operation into a continuous assured operation that shortens the time of decision making, reinforces the compliance posture, and develops a platform of scalability into future mergers, carve-outs, and growth of operations in the market.

REFERENCES

[1] Mansouri, Y., Prokhorenko, V., & Babar, M. A. (2020). An automated implementation of hybrid cloud for performance evaluation of distributed databases. Journal of Network and Computer Applications, 167, 102740.

[2] Brusa, A., Cavina, N., Rojo, N., Cucchi, M., & Silvestri, N. (2019). Development and Validation of a Control-Oriented Analytic Engine Simulator (No. 2019-24-0002). SAE Technical Paper.

[3] Bajramovic, E. (2019). Secure Logging in Operational Instrumentation and Control Systems (No. INIS-DE--2739). Erlangen-Nürnberg University, Erlangen (Germany). Technische Fakultät.

[4] ERP Timeline of What Has Happened in 2021 and What Can We Expect in 2022, aakit, online. https://aakit.com/erp-timeline-of-what-has-happened-in-2021-and-what-can-we-expect-in-2022/

[5] Overbeek, S., Klievink, B., Hesketh, D., Heijmann, F., & Tan, Y. H. (2011, August). A web-

based data pipeline for compliance in international trade. In Proceedings of the 1st Workshop on IT Innovations Enabling Seamless and Secure Supply Chains (WITNESS-2011) in conjunction with the 10th International Electronic Government Conference (Vol. 29, pp. 32-48).

[6] Taniar, D., & Rahayu, W. (2022). Data warehousing and analytics: fueling the data engine. Springer Nature.

[7] Molchin, D. (2021). Secure Integration Transfer to Cloud-based ERP Using ESB Approach.

[8] Ladiges, J., Fay, A., Holm, T., Hempen, U., Urbas, L., Obst, M., & Albers, T. (2017). Integration of modular process units into process control systems. IEEE Transactions on Industry Applications, 54(2), 1870-1880.

[9] Srai, J. S., Kumar, M., Graham, G., Phillips, W., Tooze, J., Ford, S., ... & Tiwari, A. (2016). Distributed manufacturing: scope, challenges and opportunities. International Journal of Production Research, 54(23), 6917-6935.

[10] Roh, J. J., & Hong, P. (2015). Taxonomy of ERP integrations and performance outcomes: an exploratory study of manufacturing firms. Production Planning & Control, 26(8), 617-636.

[11] Liu, X., Sun, Y., Hao, Y., & Xu, J. (2006, June). Research on Group-Oriented Enterprises Resource Planning: A Solution to Multiregional, Heterogeneous and Distributed Group Enterprises Application. In 2006 6th World Congress on Intelligent Control and Automation (Vol. 2, pp. 7186-7190). IEEE.

[12] Susan Parcells CPA, C. G. M. A. (2015). Intercompany transactions snowball: how to cope. Strategic Finance, 96(10), 48.

[13] Heisig, P., Steghöfer, J. P., Brink, C., & Sachweh, S. (2019, April). A generic traceability metamodel for enabling unified end-to-end traceability in software product lines. In Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing (pp. 2344-2353).

[14] Su, W., Zhang, L., Ge, C., & Chen, S. (2022). Association between internal control and sustainability: a literature review based on the SOX act framework. Sustainability, 14(15), 9706.

[15] Galvez, C., & Moya-Anegón, F. (2007). Standardizing formats of corporate source data. Scientometrics, 70(1), 3-26.

[16] Gollapudi, S. (2015, February). Aggregating financial services data without assumptions: A semantic data reference architecture. In Proceedings of the 2015 IEEE 9th international conference on semantic computing (IEEE ICSC 2015) (pp. 312-315). IEEE.

[17] Fikri, N., Rida, M., Abghour, N., Moussaid, K., El Omri, A., & Myara, M. (2022). A blockchain architecture for trusted sub-ledger operations and financial audit using decentralized microservices. IEEE Access, 10, 90873-90886.

[18] Koziolek, H., Burger, A., Platenius-Mohr, M., Rückert, J., Abukwaik, H., Jetley, R., & P, A. P. (2020, June). Rule-based code generation in industrial automation: four large-scale case studies applying the cayenne method. In Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: Software Engineering in Practice (pp. 152-161).

[19] Chiu, J. F., & Chiu, G. M. (2002). Placing forced checkpoints in distributed real-time embedded systems. Computing and Control Engineering, 13(4), 197-205.

[20] Hagen, C., & Alonso, G. (2002). Exception handling in workflow management systems. IEEE Transactions on software engineering, 26(10), 943-958.

[21] Brambilla, M., Ceri, S., Comai, S., & Tziviskou, C. (2005, May). Exception handling in workflow-driven web applications. In Proceedings of the 14th international conference on World Wide Web (pp. 170-179).