ISSN: 2633-4828 Vol. 5 No. S4 (July - Aug 2023)

International Journal of Applied Engineering & Technology

RE-ARCHITECTING INTERCOMPANY ACCOUNTING: AN EVENT-DRIVEN PATTERN FOR REAL-TIME MATCHING AND CONTINUOUS ELIMINATION

Rahul Rao Juvvadi
CPA, CGMA
rahulraoj99@gmail.com

ABSTRACT

Intercompany (IC) accounting is one of the larger consumers of close-cycle time in multi-entity groups, driven by mismatched timing, heterogeneous ERP estates, and end-of-period reconciliation that scales poorly. This paper sets out an event-driven architecture for IC matching and continuous elimination across SAP S/4HANA, Oracle Fusion Cloud ERP, Workday Financials, and legacy estates. The pattern is vendor-neutral: a canonical IC event schema, a matching engine that applies deterministic rules first and weighted probabilistic similarity only on the residual, currency normalisation against a governed FX source, and incremental elimination posted against an open consolidation layer rather than at period close. We describe a Microsoft Azure reference implementation and report an in-production case study at a 14-entity group processing approximately 1.8 million IC events per month. Outstanding IC mismatches at month-end fell from $42M to $1.5M, manual elimination journals from 1,240 to 185, group close from 9 to 4 working days, and exception resolution from 48 to 6 hours. We treat the audit-readiness shift from periodic to continuous controls and the threshold-calibration and transfer-pricing-drift risks the pattern does not eliminate.

Keywords—*intercompany accounting; event-driven architecture; continuous accounting; financial consolidation; ERP integration.*

I. INTRODUCTION

Month-end close in a multi-entity group fails in the usual way because invoices booked at the sending entity have not appeared at the receiver. Transfer-pricing accruals have drifted a few basis points across the ledgers, and booking entities have applied slightly different FX rates, therefore, what's left is a queue of mismatched items for controllers to chase by email until the books are closed. In the groups we work with, periodic intercompany (IC) reconciliation consumes about one third of the close cycle effort, and most of that is mechanical [1].

Three trends have further worsened this situation. First, the era of mergers and acquisitions has resulted in many companies now running multiple ERP systems in parallel -- SAP S/4HANA, Oracle Fusion Cloud ERP, Workday Financials, and so on, and then there are the regional ERP instances that the central team has no authority to shut down [9], [10]. Second, regulators and audit committees are no longer as tolerant of late breaking issues because closings that were acceptable ten years ago are no longer on the table. Finally, the modern technology stack, which includes API accessible ERPs, durable message brokers, serverless compute, and columnar data warehouses, makes event level reconciliation possible in a way that was not conceivable when the standard intercompany patterns were first defined [13].

Three contributions are made in this paper: (1) a canonical intercompany event schema is defined, which accurately represents the trading pattern of groups in practice, including flows of inventory, services, royalties, cost allocations, and transfer-pricing adjustments, and describes the attributes that a downstream matcher needs, such as tax treatment and FX-governance metadata. (2) A matching engine is specified, which applies deterministic rules first, and only applies weighted probabilistic similarity to the remaining unmatched items, under a policy weight vector that can be configured, and detects duplicate postings in realistic edge-cases, such as reversal-and-repost, partial credits, and multi-leg pass-throughs. (3) A case study is reported, which involves a 14-entity group processing 1.8 million events per month, with measured outcomes reported, and changes in governance and audit readiness discussed candidly, rather than as a success story.

Section II positions the work against prior literature. Section III restates the IC problem in terms a controller would recognise. Section IV describes the architecture. Section V reports the case study, governance treatment, and limitations. Section VI concludes.

II. RELATED WORK

A. Continuous Accounting as the Immediate Antecedent

The closest antecedent is the continuous-accounting literature. Izzo, Fasan, and Tiscini study Oracle's own transformation and report that the shift away from periodic batch towards always-on processing changes the role of accountants from preparers to reviewers, and raises the operational value of timely information [1]. Yoon et al. propose a three-layer continuous-audit architecture—data acquisition, analytical monitoring, and alert generation—which underpins our matching engine's separation of normalisation, scoring, and exception routing [2]. Seiger et al. show how event streams from operational systems can drive process-aware workflows in industrial settings [3]; our event orchestration layer is a direct analogue applied to finance. Gao and Jiang's analytical treatment of continuous versus discrete recognition flags the incentive risk when measurement frequency rises without matching governance [15]—we return to this in Section V.

B. ERP-side IC Capabilities and Integration

While most major ERP systems include some form of intercompany (IC) functionality, it is rarely sufficient by itself, because Kumar's review of Oracle Fusion Cloud ERP's intercompany processes shows that the built-in IC modules are predicated on a single-instance landscape and begin to fail when group entities are split across multiple ERPs [7]. Moreover, Potla's hybrid integration blueprint directly addresses this issue, suggesting that for organisations that inherit heterogeneous ERPs through M&A, the right answer is an integration layer that sits above the ERPs, rather than expecting each ERP to provide a full, end-to-end IC API on its own [9]. Furthermore, Bahssas et al. trace the long-term shift in ERP from product-centric to cloud-centric deployments, which underpins our assumption that the integration surface is now primarily API-mediated rather than file-based [10], and Akrong et al. [5] apply the DeLone-McLean model to ERP success, finding that information quality is the main driver of user satisfaction, which is directly relevant to our work. Therefore, the improvements we report in our case study are ultimately capped by the quality of upstream data, a limitation we return to in Section V.

Subramanian's case analysis of ERP implementation models highlights the importance of phased rollouts and careful process alignment [6], both of which we observed as prerequisites in our own deployment, and similarly, Mallieswari et al. investigate the integration of IoT with ERP in the context of manufacturing, demonstrating that it is practical to ingest high-frequency event streams into ERP back-ends [8], which we take as evidence that the level of throughput we require is comfortably within proven limits. However, Ionescu's process-focused study of ERP-driven digitalisation of accounting processes supports our decision to frame intercompany reconciliation primarily as a control problem, rather than a pure tooling or platform problem [4].

C. Event Extraction, Data Integration, and Analytics

Berti et al. show how to extract object-centric event logs from SAP ERP databases at field-level granularity [12]; their work is the empirical basis for our event schema's source-document linkages. Darmont et al. survey data management for analytical workloads and provide the background on storage tiering we adopt for hot matching state versus cold audit trail [13]. Daraio, Di Leo, and Scannapieco's completeness-aware integration framework speaks to the case-study problem of partial trading-partner metadata: the integration layer adapts their completeness scoring to flag events that are matchable-but-incomplete rather than rejecting them outright [11]. Szukits's analysis of analytics-driven decision-making in controllership is the source of our framing of finance controllers as analysts of exceptions rather than processors of journals [14].

D. Governance, Platforms, and Standardisation

Wu et al. on absorptive capacity and ERP practice underwrite the finding that the architecture's benefits compound with organisational learning [16]. Ruggeri, Leotta, and Rizza's study of accounting language games in digital settings supports the canonical-schema rationale: a shared vocabulary across entities is itself a control mechanism, not merely a convenience [17]. Bartelheimer et al. systematise the platform lexicon used here to describe the integration tier and its programmable surface [18]. Chou, Chang, and Peng's XBRL-plus-text approach establishes the precedent for combining structured accounting data with semi-structured disclosure content, a pattern reused here for exception narratives attached to events [19]. Fernández-Cejas et al.'s CURIE work on conceptual modelling for enterprise applications underwrites our choice to define the canonical IC event in ontology terms rather than as a flat record [20].

E. What Is Distinct Here

Periodic batch reconciliation tools and ERP-native IC modules dominate current practice. The first cannot run continuously; the second cannot span heterogeneous estates. Third-party consolidation platforms (Hyperion-class, BlackLine-class) offer continuous reconciliation but as closed services with proprietary schemas. The contribution here is a vendor-neutral event schema and matching pattern that runs above any ERP, in any single-cloud or multi-cloud configuration, and that distinguishes policy-driven transfer-pricing variance from reconciliation breaks—an explicit accounting concept the closed platforms tend to elide.

III. TRANSFORMATION OF INTERCOMPANY ACCOUNTING

A. The Problem Restated

IC accounting is the consolidation-time reconciliation of transactions between legal entities in the same group: inventory transfers, intra-group services and management charges, royalty and licence payments, cost allocations, and transfer-pricing adjustments that align intra-period margins with arm's-length policy. Under IFRS 10 and ASC 810, these transactions—and the resulting unrealised profit-in-inventory, intra-group receivables and payables, and intra-group revenue and expense—must be eliminated from the consolidated financial statements. Profit-in-inventory elimination is materially different from a receivable/payable offset: the former requires tracing the inventory carry until external sale and reversing the unrealised margin, while the latter is an arithmetic netting. Legacy IC patterns often conflate the two and produce mismatch numbers dominated by timing rather than substance.

Tax treatment of IC flows is jurisdiction-specific. Cross-border services flows carry withholding obligations in many jurisdictions; intra-group services within an indirect-tax registration may be VAT or GST exposed even where they do not give rise to economic income; recoverability of input tax is governed by where the recipient sits and what its supply mix is. An IC event model that omits tax attributes cannot be reconciled against statutory ledgers, only against management ledgers—an important distinction we return to in Section IV.

Transfer-pricing adjustments deserve separate handling. The OECD framework, as understood in 2023, treats year-end and quarterly true-ups as policy-driven movements to align intra-period margins with the agreed arm's-length range; these are not reconciliation breaks. The architecture must distinguish intra-period policy variance (expected, will be true-d up) from reconciliation breaks (counterparty has not posted, or has posted to the wrong account). Conflating them is a common source of overstated mismatch values in legacy reporting. Fig. 1 shows the close-cycle effort distribution we measured at the case-study group prior to implementation; intercompany reconciliation alone consumed 38% of close-cycle effort.

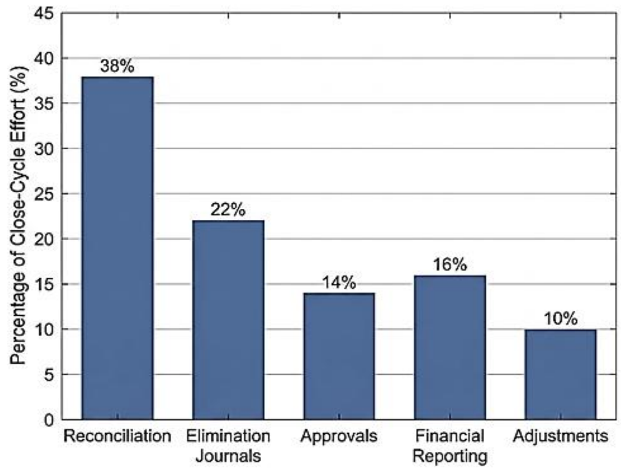

Fig. 1. Distribution of close-cycle effort across activities at the case-study group prior to implementation. Values are percentage of total working hours logged by controllers during the period N − 1 close (twelve months pre-deployment baseline).

B. Limitations of Legacy Reconciliation

Three patterns recur. Reconciliation runs once per period against ledger exports, so timing differences and posting errors surface together at close and disentangling them is manual. Batch processing on each ERP runs to its own schedule, so a reconciliation that requires both sides to have run is gated by whichever side runs last. Where the group spans ERPs, document numbering, transaction-type taxonomies, and chart-of-accounts mappings differ; FX rate sources differ too, so the same underlying flow can be carried in two ledgers at materially different functional-currency values. Spreadsheet-mediated reconciliation absorbs this complexity through controller labour rather than control. Audit walkthroughs in this environment depend on email threads and named files, not on a queryable system of record.

Table I contrasts the legacy and event-driven baselines along five stages of the IC process.

Table I: Process-Stage Comparison: Legacy VS Event-Driven IC Accounting

Process Stage	Legacy Monthly Process Time	Event-Driven Processing Time	Manual Effort (%)	Automation Level (%)
Transaction Matching	3–5 days	Minutes	75	85
Reconciliation Review	2–3 days	Continuous	70	80
Elimination Journal Posting	1–2 days	Continuous (incremental)	65	90
Exception Resolution	2 days	Same-day routing	60	75
Audit Validation	1 day	Real-time traceability	50	85

Note. Process times and effort/automation percentages are mean values across the case-study group's twelve-month pre-deployment baseline (legacy) and twelve-month post-deployment operation (event-driven). "Manual Effort" is the share of controller time logged against that stage; "Automation Level" is the share of events resolved without controller intervention.

C. Continuous Accounting as the Destination State

The real game-changer with continuous accounting is that it takes the close from a one-off, periodic event to a steady-state, continuous process, and reconciliations are run as transactions land. Exceptions are identified and flagged within hours, not weeks, because eliminations are posted incrementally into an open consolidation layer. By the time you get to period end, the close is essentially a roll-forward of an already balanced position, not a rescue mission, since the technical building blocks required to do this, such as API-mediated ERPs, durable event transport, elastic compute, and well-governed FX feeds, are now table stakes for modern cloud platforms [10], [13].

D. Vendor-Neutral Architectural Position

We propose an architecture in which IC transactions in any source ERP emit canonical events at posting; events flow through an integration tier independent of any specific ERP vendor; a matching engine applies deterministic rules first and weighted probabilistic similarity only on the residual; matched pairs trigger incremental elimination entries against an open consolidation ledger; and controllers see exceptions in a dashboard, route them to the responsible entity, and approve corrections in-flight.

The match confidence used by the probabilistic stage is given in Eq. (1):

$$M_c = w_1 T_m + w_2 A_m + w_3 C_m + w_4 D_m \qquad (1)$$

where M_c is the composite match confidence, T_m is transaction-reference similarity, A_m is amount similarity after FX normalisation, C_m is currency-attribute similarity, D_m is date-proximity similarity, and $w_1 \ldots w_4$ are weights set by accounting policy rather than tuned as hyperparameters. The weight vector is the lever by which the controllership team encodes which dimensions matter most for their flows—policy, not learning.

IV. ARCHITECTURE FOR MATCHING AND CONTINUOUS ELIMINATION

A. Canonical IC Event Model

The event schema covers the realistic IC flow types: inventory transfers (with carry value and onward-sale linkage for profit-in-inventory tracking), intra-group services and management charges (with service category and benefits-test metadata where applicable), royalty and licence flows (with underlying intangible-asset reference and rate basis), cost allocations (with allocation key and source pool), and transfer-pricing adjustments (with policy reference and target margin band). Each event carries: sender and receiver legal-entity identifiers; trading-partner reference resolved through a governed counterparty master; transaction-type classification; gross and net amounts in transaction currency; functional currencies of both sides; tax attributes including withholding rate, VAT or GST treatment, and recoverability indicator; source-document references (purchase order, sales order, invoice, shipment, service agreement, allocation cycle); an immutable event identifier; and an audit-trail pointer to the originating ERP posting.

Currency attributes are first-class. Amounts are carried in transaction currency with the FX rate, rate source, and rate date attached. Currency normalisation precedes matching, not the reverse; the matcher operates on functional-currency-normalised amounts.

B. Event Transport and Integration

The ERPs produce events against well-governed API surfaces - S/4HANA Cloud APIs for SAP, Oracle Integration Cloud REST endpoints for Fusion, the web-services stack for Workday Financials and a mix of file interfaces and change data capture for the legacy estates, and we have an API management layer sitting in front of these which terminate client connections, enforce authentication/authorization and capture per-call telemetry.

Events then move over a durable messaging layer - Azure Event Grid is used for fan-out routing, Service Bus for queued processing where ordering and at-least-once delivery are critical (e.g. for financial postings), before they hit the matcher.

Before they hit the matcher, events pass through a validate--enrich--route pipeline, where we check for required fields, resolve counterparty references, infer missing tax attributes where the source ERP does not provide them, and attach the FX rate as of the booking date from the governed corporate rate table.

C. Matching Engine

The matching engine evaluates events in two stages. Deterministic matching runs first: trading-partner pair, transaction reference, functional-currency-normalised amount within rounding tolerance, and posting period must agree exactly. Deterministic match is the common case in our data—approximately 80% of inventory and royalty events resolve at this stage.

The residual goes to weighted-similarity matching. The composite score Mc from Eq. (1) is computed against the candidate set; pairs above a configured upper threshold are auto-matched, pairs below a lower threshold are routed to exceptions, and the band between is reviewed by controllers. The weight vector is policy-set per flow type: for inventory, reference and amount dominate; for transfer-pricing adjustments, date proximity carries more weight because timing variance is expected by design.

Amount tolerance is given in Eq. (2):

$$|A_s - A_r| \leq \theta \quad (2)$$

where A_s and A_r are sender and receiver functional-currency amounts and θ is the policy-set tolerance—an absolute floor for low-value items and a basis-point band above it. Tolerance is the central calibration risk in this architecture: loose thresholds mask issues, tight thresholds inflate the exception queue (we return to this in Section V).

FX normalisation applies before matching, as in Eq. (3):

$$A_n = A_f \times FX_r \quad (3)$$

where A_n is the normalised functional-currency amount, A_f is the foreign-currency amount, and FX_r is the rate from the governed corporate rate table at the posting date. Two normalisations using different rate sources will produce different A_n; governance of the rate source is what makes this equation operational rather than merely arithmetic.

Duplicate detection requires explicit handling of realistic edge cases. Reversal-and-repost (an entry reversed in period N and re-posted with different references in period N or N+1) must be recognised as the same economic event, not as two separate flows that happen to net to the original amount. Partial credits against an original invoice must be matched against the residual receivable, not duplicated. Multi-leg pass-throughs (entity A invoices B, B invoices C with a markup) must be treated as two distinct IC pairs with their own elimination treatments, not collapsed. The engine carries a posting fingerprint—source ERP, document number, line, reversal indicator—that survives these patterns.

Table II reports the matched-rate, exception-rate, and resolution-time distribution across the five flow types in the case study.

Table II: Matching Engine Performance by IC Flow Type

Transaction Type	Deterministic Match (%)	Probabilistic Match (%)	Exception Rate (%)	Mean Resolution Time
Inventory Transfers	88	9	3	4 h
Service Charges	81	14	5	6 h
Royalty Payments	85	11	4	5 h
Shared Cost Allocations	76	18	6	8 h
Transfer-Pricing Adjustments	69	21	10	12 h

Note. Percentages are share of monthly events resolved at each stage, averaged over the twelve-month post-deployment window. Resolution time is wall-clock hours from event arrival to controller disposition; transfer-pricing adjustments show longer tails because policy-band checks require quarterly reference data.

D. Continuous Elimination Workflow

Matched pairs trigger incremental elimination entries against an open consolidation ledger that runs alongside the period. The consolidated position is updated as the underlying matches are confirmed; period close is then a snapshot of an already-eliminated balance, not the start of an elimination exercise. This is incremental update of consolidated balances, not a bypass of period locks—the statutory ledgers themselves close on their normal schedule, and the consolidation ledger is reconciled to them at close. Top-side adjustments are reserved for materiality- or policy-driven movements (e.g., consolidation-level reclassifications, alignment with prior-period restatements), recorded as such with explicit reason codes and approved under preparer/reviewer/approver segregation of duties; they are not a workaround for matcher gaps. Fig. 2 traces the outstanding IC mismatch value across the seven working days of the case-study close.

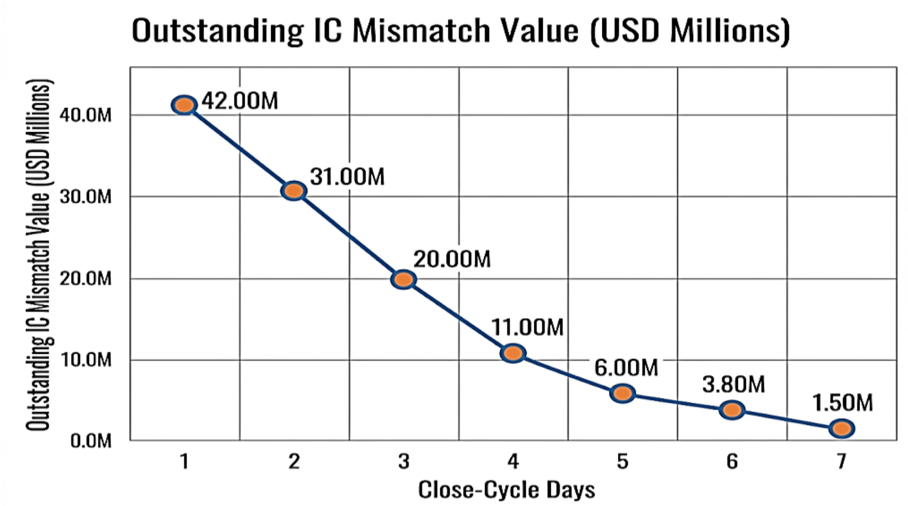

Fig. 2. Outstanding IC mismatch value (USD millions) by close-cycle day under continuous processing. Values are means across twelve post-deployment closes. Day 1 reflects the residual after intra-period continuous matching; the curve is dominated by transfer-pricing true-ups in days 3–5 and tax-attribute corrections in days 6–7.

E. Governance, Security, and Auditability

Segregation of duties is enforced as a control objective, not as a user-distinctness check: preparer (event submission), reviewer (exception triage and threshold override), and approver (top-side adjustments and policy changes) are distinct roles with distinct identities and audit trails. Role assignments are managed in Microsoft Entra ID; secrets and connection material are held in Azure Key Vault; every action against the consolidation ledger emits an immutable audit event linked to the underlying IC event identifier.

The audit-evidence profile is not the same as in a periodic system. Auditors evaluate continuously operating rules rather than periodic controls, and walkthrough evidence is replay of an event sequence rather than inspection of period-end reconciliations. We return to this shift in Section V.

V. EVALUATION AND GOVERNANCE IMPLICATIONS

A. Reference Deployment on Azure

In our reference architecture, all components are deployed on Microsoft Azure, and the ERPs emit intercompany events via controlled API endpoints. They are managed by Azure API Management, because it closes the connection and performs authentication. Azure Event Grid then routes the events to Service Bus queues, thus providing ordered, at-least-once delivery to the matcher. Stateless processing such as field validation, FX normalization, deterministic matching, and similarity scoring is implemented as Azure Functions, so stateful processing such as exception handling, controller approval, and escalation is performed by Azure Logic Apps.

The matched-pair state and the open consolidation ledger are stored in Azure SQL Database using row-level temporal history, and meanwhile, an append-only audit trail of all events, match decisions, elimination entries, and approvals is written to Azure Data Lake Storage Gen2. Controllers view exceptions and exposures using Power BI, therefore row-level security aligns with entity-level roles. Fig. 3 shows the example deployment, thus providing a visual representation of the architecture.

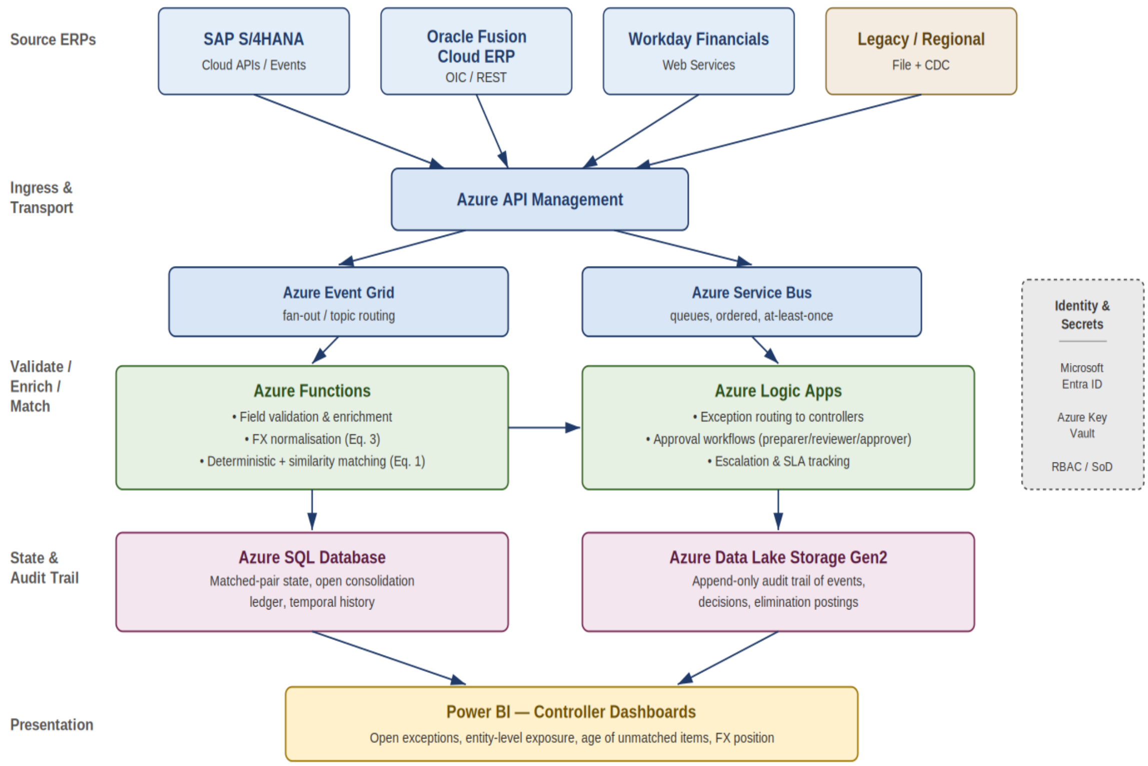

Fig. 3. Reference deployment of the proposed pattern on Microsoft Azure. Boundaries between ingress, validate/enrich/match, state, and presentation tiers are explicit. Identity (Entra ID) and secrets (Key Vault) operate as a side channel applied across all tiers; RBAC and SoD are enforced at the role boundary, not at the user-distinctness boundary.

B. Case Study

The case-study organisation is a 14-entity multinational with both consumer-products and industrial-services lines of business, with operations in eight countries and three functional currencies, and the ERP environment at the time of deployment was SAP S/4HANA for the seven largest entities, Oracle Fusion Cloud ERP for five entities acquired over the previous three years, and a legacy on-premises ERP system retained for statutory reporting for two smaller regional entities. Intercompany throughput is approximately 1.8M events per month, with activity concentrated in the last five working days of each period, consistent with the month-end posting peaks common in most organisations, and the measurement window spans 12 months on the new pattern, compared to the 12 months immediately prior on the legacy spreadsheet-and-batch approach. "Mismatch" is defined as the absolute USD value of unmatched intercompany items at the consolidation cutoff, and close duration is measured in working days from the group cutoff to consolidation lock, therefore, the headline results, summarised in Table III, are material. Outstanding mismatches at close fell from $42M to $1.5M, manual elimination journals from 1,240/month to 185, group close duration from 9 working days to 4, mean exception resolution time from 48 hours to 6, and controller workload on reconciliation tasks to 58% of its original baseline, thus resulting in significant improvements.

Table III: Before-and-after Performance at the Case-Study Group

Metric	Legacy Environment	Event-Driven Environment	Improvement (%)
IC mismatch value at close	$42M	$1.5M	96.4
Manual elimination journals (per month)	1,240	185	85.1
Close duration (working days)	9	4	55.6
Mean exception resolution time	48 h	6 h	87.5
Controller workload (vs. baseline = 100)	100	58	42.0

Note. Pre-deployment values are means across twelve consecutive monthly closes on the legacy process; post-deployment values are means across the twelve closes following the second full month after go-live (steady-state assumption). Close duration is working days from group cutoff to consolidation lock. Controller workload is hours logged against IC reconciliation tasks, indexed to 100 at baseline.

The reduction-rate and efficiency-gain identities are given by Eq. (4) and Eq. (5):

$$R_m = (M_b - M_a) / M_b \times 100 \quad (4)$$

$$E_c = (T_b - T_a) / T_b \times 100 \quad (5)$$

where M_b and M_a are the mismatch values before and after, and T_b and T_a are close durations. These are arithmetic identities, not findings; the findings are the input values themselves.

C. Governance and Risk—Honest Trade-offs

The pattern introduces governance risks that the legacy pattern did not, and it changes the audit posture in ways that are worth stating explicitly. Tolerance calibration is the central risk because a loose tolerance (θ in Eq. (2)) masks small but persistent errors that compound over time, while a tight tolerance floods the exception queue and pushes controllers back toward batch-mode review. In our implementation, θ is set by flow type, and we review the false-positive and false-negative distributions monthly; the weight vector in Eq. (1) is governed in the same way, and both are treated as accounting-policy decisions, documented in the intercompany operations manual, rather than as tuning knobs for maximising the headline match rate.

Transfer-pricing drift is not, in itself, a reconciliation break, however, intra-period margin variance against policy is expected by design and is trued up at quarter-end.

The architecture therefore tags transfer-pricing events with the policy reference and target margin band; variance within that band does not generate an exception, while drift outside the band does, thus historically, conflating policy variance with reconciliation breaks—a common failure mode in legacy processes—is what inflated reported mismatch values. The treatment here is consistent with the OECD transfer-pricing guidance as understood at the time of writing, and the audit profile also changes, consequently, external auditors who previously walked through periodic intercompany controls now assess continuously operating rules: the deterministic-rule definitions, the weight vector θ, the governance of the FX rate source, and the change history for all four. IT general controls scope expands to include the matcher and orchestration components, while walkthrough evidence becomes event replay rather than inspection of period-end reconciliation packs.

We worked with the group's external auditor through the first close on the new pattern, to align on the evidence model, because the change in audit posture is real, and it is not neutralised by automation; furthermore, there are clear bounds to what is automated. The pattern automates the mechanical aspects: reference matching, FX normalisation, duplicate detection, threshold-bounded auto-acceptance, incremental elimination posting, but it does not automate judgement: top-side adjustments, interpretation of transfer-pricing policy, materiality assessment, or the treatment of structurally ambiguous items (for example, multi-leg flows with split benefits), therefore, finance controllers are still accountable for these decisions [14], [17].

D. Threats to Validity and Limitations

This is a single-case study, not a controlled comparison, and it may not be generalizable to other industries, or to ERP estates with a different composition. There are potential biases, including selection bias, because the group opted to adopt the pattern, and Hawthorne effects, since the controllers knew that pre- and post-deployment performance would be measured. The benefits depend on upstream ERP master data quality, particularly counterparty masters, chart-of-accounts mappings, and FX rate governance, and the architecture degrades gracefully when these are weak, but does not fix them [5], [11]. Finally, auditor readiness for continuously operating controls is still an evolving practice rather than a settled framework, and profession-wide guidance on continuous-control assurance was still in development at the time of writing, therefore, it is still a developing area.

VI. CONCLUSION

An event-driven IC pattern is achievable on standard cloud infrastructure and yields measurable improvements in mismatch value, close duration, and controller workload at a 14-entity group processing approximately 1.8 million events per month. The pattern is vendor-neutral by construction: the canonical event schema, the deterministic-then-weighted matcher, and the open consolidation ledger sit above any specific ERP, and the case study spans S/4HANA, Oracle Fusion, and a retained legacy estate. The gains are not free—tolerance calibration, transfer-pricing tagging, and audit-evidence reshaping are ongoing accounting work, not one-off engineering deliverables. Further evaluation across industries, across ERP mixes, and against settled continuous-control assurance frameworks would strengthen what is at present a defensible single-case result.

REFERENCES

[1] M. F. Izzo, M. Fasan, and R. Tiscini, "The role of digital transformation in enabling continuous accounting and the effects on intellectual capital: the case of Oracle," *Meditari Accountancy Research*, vol. 30, no. 4, pp. 1007–1026, Aug. 2021.

[2] K. Yoon, Y. Liu, T. Chiu, and M. A. Vasarhelyi, "Design and evaluation of an advanced continuous data level auditing system: a three-layer structure," *International Journal of Accounting Information Systems*, vol. 42, art. 100524, Jul. 2021.

[3] R. Seiger, L. Malburg, B. Weber, and R. Bergmann, "Integrating process management and event processing in smart factories: a systems architecture and use cases," *Journal of Manufacturing Systems*, vol. 63, pp. 575–592, Apr. 2022.

[4] B. Ionescu, "The process of digitalization of an accounting organization using ERP systems," *Annales Universitatis Apulensis: Series Oeconomica*, vol. 2, no. 24, 2022.

[5] G. B. Akrong, S. Yunfei, and E. Owusu, "Development and validation of an improved DeLone–McLean IS success model: application to the evaluation of a tax administration ERP," *International Journal of Accounting Information Systems*, vol. 47, art. 100579, Aug. 2022.

[6] G. H. Subramanian, "ERP implementation model, findings and its applications to organisations: an exploratory case analysis," *International Journal of Business Information Systems*, vol. 41, no. 2, p. 177, Jan. 2022.

[7] J. Kumar, "Intercompany processes efficiency using Oracle Fusion ERP Cloud—a systematic review," *International Journal of Computer Trends and Technology*, Dec. 2022.

[8] R. Mallieswari, V. Rema, and B. N. Thontadarya, "Reverberation of ERP implementation in manufacturing enterprises process outcomes through IoT: evidence from India," in *Proc. IEEE 19th India Council Int. Conf. (INDICON)*, Nov. 2022, pp. 1–5.

[9] R. B. Potla, "Hybrid integration for manufacturing finance: RTR controls, intercompany eliminations, and auditability across multi-ERP estates," *International Journal of ERP and CRM*, vol. 3, no. 1, pp. 11–38, Jul. 2022.

[10] D. M. Bahssas, A. M. AlBar, and Md. R. Hoque, "Enterprise Resource Planning (ERP) systems: design, trends and deployment," *International Technology Management Review*, vol. 5, no. 2, p. 72, Jan. 2015.

[11] C. Daraio, S. Di Leo, and M. Scannapieco, "Accounting for quality in data integration systems: a completeness-aware integration approach," *Scientometrics*, vol. 127, no. 3, pp. 1465–1490, Jan. 2022.

[12] A. Berti, G. Park, M. Rafiei, and W. M. P. van der Aalst, "A generic approach to extract object-centric event data from databases supporting SAP ERP," *Journal of Intelligent Information Systems*, vol. 61, no. 3, pp. 835–857, Jun. 2023.

[13] J. Darmont, B. Novikov, R. Wrembel, and L. Bellatreche, "Advances on data management and information systems," *Information Systems Frontiers*, vol. 24, no. 1, pp. 1–10, Feb. 2022.

[14] Á. Szukits, "The illusion of data-driven decision making—the mediating effect of digital orientation and controllers' added value in explaining organizational implications of advanced analytics," *Journal of Management Control*, vol. 33, no. 3, pp. 403–446, Jun. 2022.

[15] P. Gao and X. Jiang, "The economic consequences of discrete recognition and continuous measurement," *Journal of Accounting and Economics*, vol. 69, no. 1, art. 101250, Sep. 2019.

[16] K. Wu, G. Beydoun, O. Sohaib, and A. Gill, "The co-construct/co-evolving process between organization's absorptive capacity and enterprise system practice under changing context: the case of ERP practice," *Information Systems Frontiers*, vol. 24, no. 6, pp. 2123–2138, Jan. 2022.

[17] D. Ruggeri, A. Leotta, and C. Rizza, "Digitalisation and accounting language games in organisational contexts," *Journal of Management & Governance*, vol. 27, no. 3, pp. 817–838, Apr. 2022.

[18] C. Bartelheimer, P. Z. Heiden, H. Lüttenberg, and D. Beverungen, "Systematizing the lexicon of platforms in information systems: a data-driven study," *Electronic Markets*, vol. 32, no. 1, pp. 375–396, Mar. 2022.

[19] C.-C. Chou, C. J. Chang, and J. Peng, "Integrating XBRL data with textual information in Chinese: a semantic web approach," *International Journal of Accounting Information Systems*, vol. 21, pp. 32–46, May 2016.

[20] M. Fernández-Cejas, C. J. Pérez-González, J. L. Roda-García, and M. Colebrook, "CURIE: towards an ontology and enterprise architecture of a CRM conceptual model," *Business & Information Systems Engineering*, vol. 64, no. 5, pp. 615–643, Feb. 2022.