International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(23s) 2026 pp. 1730-1742
https://doi.org/10.70917/ijcisim-2026-5725
Article

Designing Auditability into Enterprise Payroll Integrations: A Configuration-Driven Architecture for Two-Stage Reconciliation and Batch-Scoped Rollback in Multi-Jurisdiction Equity Compensation Tax Processing

Rajagopal Arputham Chetty

Senior Workday Systems Analyst, Payroll
Independent Researcher, India

Abstract: Equity compensation payroll differs from ordinary payroll in a way that matters for control design: the tax-relevant facts arrive from outside the payroll system. A stock plan administrator supplies transaction records in a vendor format, and those records carry jurisdictional attributes that must be translated into payroll codes before any calculation occurs. This translation step is where error enters, and it is invisible to the payroll engine, which will faithfully compute tax on whatever it is given. This paper specifies an auditability architecture for that class of integration. Three design elements are described. Reconciliation is split into two stages, one before payroll calculation and one after, so that input fidelity and calculation correctness are proven separately rather than confounded. Rollback is scoped to an immutable batch identifier assigned at load time, which makes reversibility a property of a known set of records rather than an open-ended correction exercise, and the boundary at which rollback ceases to be available is stated explicitly. Jurisdictional mapping, earnings and deduction translation, and period control are expressed as configuration under segregation of duties rather than as integration code, so that a jurisdiction change is a reviewable configuration event rather than a code release. The architecture is drawn from a production integration operating across more than 90 tax jurisdictions in the United States and Canada, through which 52,816 equity transactions have been processed since October 2025. Operational counts are reported. Reconciliation break rates, rollback frequency, and cycle-close times were not instrumented and are not claimed.
Keywords: Payroll integration, auditability, equity compensation, ESPP, multi-jurisdiction taxation, reconciliation, batch rollback, configuration management, segregation of duties, SOX compliance, ERP controls, Workday Payroll

1. INTRODUCTION

Payroll systems are generally trusted to be correct. The calculation engine applies published rates to declared elections, and where the inputs are the employer's own time and absence records, the provenance of those inputs is internal and controllable. Equity compensation payroll breaks that assumption in a specific way, and the break is structural rather than incidental.

When shares vest or are sold, the resulting income is compensation for tax purposes [15], but the taxable event is recorded by a stock plan administrator, not by the employer. The administrator produces a transaction file describing units, proceeds, and the jurisdictions in which the income is taxable. That file is the authoritative input to payroll for the cycle, and it arrives expressed in the administrator's own encoding. Jurisdiction identifiers, earnings categories, and deduction types are all in a vocabulary the payroll system does not share. Every one of those values must be translated into payroll codes before calculation, and the translation is the point at which a defect can enter without any component reporting an error.

This is what makes the class of integration difficult to audit. The payroll engine is not wrong. Given a mapped input, it computes correctly and consistently. If the mapping assigned a transaction to the wrong jurisdiction, the engine will compute the wrong jurisdiction's tax perfectly, produce a plausible result, and report no fault. Downstream, that result is remitted to a tax authority and reported on a year-end form. The error surfaces, if it surfaces at all, as a reporting discrepancy long after the cycle closed.

The consequence of a single incorrect withholding is asymmetric in time. If the defect is identified within the calendar year and before year-end forms are generated, correction is possible through an adjustment cycle, at a cost measured in effort. If it is identified after year-end reporting, correction requires amended filings, and depending on amount and jurisdiction it can carry penalty exposure and, in the more serious cases, litigation risk. The control objective is therefore not merely to detect error but to detect it while reversal is still inexpensive, which means detecting it inside the processing window rather than after it.

Three properties follow from that objective, and they organize the architecture presented here.

Input fidelity and calculation correctness must be proven separately. A single reconciliation performed after calculation cannot distinguish a mapping defect from a calculation defect, because both present as a difference between expectation and result. Separating the proof into a pre-calculation stage and a post-calculation stage makes the two failure classes independently observable.

Reversibility must be bounded to a known set. Correcting an equity cycle by locating and adjusting individual records is error-prone precisely when the operator is under time pressure. Binding every record loaded in a run to an immutable batch identifier converts reversal into an operation on a defined set, and makes the question of what is reversible answerable by inspection.

Jurisdictional knowledge must live in configuration, not in code. Jurisdictions change. When the mapping between a vendor code and a payroll jurisdiction is embedded in transformation logic, every jurisdictional change is a code release, with the lead time and release risk that implies. When it is configuration, the change is a reviewable data event, auditable in the same record as any other configuration change.

The contribution is the composition of these three, evidenced by a production implementation. Section 2 positions the work and states the boundary against the author's prior paper. Section 3 describes why multi-jurisdiction equity tax is structurally harder than ordinary payroll. Sections 4 through 7 specify the audit model, the configuration surface, the two-stage reconciliation, and the rollback semantics. Section 8 reports what is known and states plainly what was never measured. Section 9 concludes.

2. BACKGROUND AND POSITIONING

2.1 Continuous auditing and ERP control

The problem addressed here is a specific instance of a general one that the auditing literature has treated for some time: how control assurance is obtained in systems where transactions are processed automatically and at a volume that precludes manual inspection. Kuhn and Sutton survey continuous auditing in ERP environments and identify the persistent gap between the audit evidence such systems can technically produce and the evidence organizations actually configure them to produce [2]. Singh and Best describe the design and implementation of continuous monitoring within a large ERP platform, and their account makes clear that continuous control is a configuration and design discipline rather than a product feature [3]. Elbardan and Kholeif examine how ERP adoption reshapes internal audit practice, including the shift of assurance work toward configuration review [4].

The requirement that a record identify who changed a value, when, and from what to what is long-established in the information systems control literature. Menkus set out the role of the audit trail in maintaining information integrity, and the elements he identified, actor, timestamp, prior value, and retention, remain the substance of what an auditor requests [5].

The regulatory frame for a US public employer is internal control over financial reporting under the Sarbanes-Oxley Act, whose Section 404 requires management assessment of control effectiveness [6], [7], assessed against a recognised control framework [13]. Payroll input is within that scope, which is why the process described here is subject to control procedure rather than left to operational discretion.

2.2 The multi-jurisdiction equity gap

What the literature above does not address is the equity compensation case specifically, and the gap is not incidental. Continuous auditing work generally assumes the transaction originates inside the system being audited, or arrives from a source under common control. Equity transactions do not. They originate with an external administrator, they carry jurisdictional attributes determined by events (share allocation) that occurred potentially years before the taxable event (share sale), and they must be translated across two independent code vocabularies before the system of record can act on them.

The result is a control surface that sits between two systems and belongs cleanly to neither. The administrator is authoritative for the transaction but has no visibility into payroll configuration. The payroll system is authoritative for the calculation but accepts the mapped input as given. Neither party is positioned to detect a mapping defect, and in the absence of a deliberately designed reconciliation, no one does until a tax authority does.

2.3 Contribution boundary against the author's prior work

The author has previously published an implementation account of the same integration [1]. Because the two papers share a system context, the boundary is stated explicitly rather than left to inference.

Aspect	Prior paper [1]	This paper
Subject	How the integration was built	How the integration is proven, reversed, and evidenced
Data movement, file specification, transformation tooling	Specified in detail	Assumed, cited, not restated
Tax withholding computation path	Described	Treated as a black box that computes correctly on given input
Two-stage reconciliation design	Not addressed	Section 6, new
Batch-scoped rollback semantics and boundary	Not addressed	Section 7, new
Configuration-versus-code separation and segregation of duties	Not addressed	Section 5, new
Audit evidence model and retention	Not addressed	Section 4, new
Jurisdiction derivation from allocation date	Not addressed	Section 3.2, new

Table 1. Contribution boundary against the author's prior implementation paper. The system context is shared and disclosed. The control architecture is new material.

Related work by the author on adjacent payroll problems is cited where it genuinely bears on a point rather than for completeness: retention and purge design in multi-tenant payroll [8], de-identification of payroll data for non-production use [9], cross-border year-end reporting obligations in Canada [10], and the handling of statutory change within a configured payroll system [11], [12].

3. WHY MULTI-JURISDICTION EQUITY TAX IS STRUCTURALLY HARDER

3.1 Four sources of jurisdictional multiplicity

In ordinary payroll, the jurisdictions applicable to a worker are derived from the worker profile as it stands at the moment of calculation, combined with the elections that worker has filed, against federal withholding guidance [14]. The derivation is a lookup against current state.

Equity compensation defeats that derivation in four distinct ways, and they compound.

Work and residence separation. A worker may perform duties in one state and reside in another. Both jurisdictions may assert a claim on equity income, and each may carry local jurisdictions beneath it with their own withholding and reporting obligations.

Elective tax allocation. Where duties span multiple jurisdictions, a worker may allocate income across them according to a declared work pattern. A single transaction then produces obligations in several jurisdictions simultaneously, in proportions that are themselves a data input.

Mid-year relocation. A worker who relocates during a year generates equity transactions attributable to different periods and therefore different jurisdictional sets, within a single tax year and often within a single cycle.

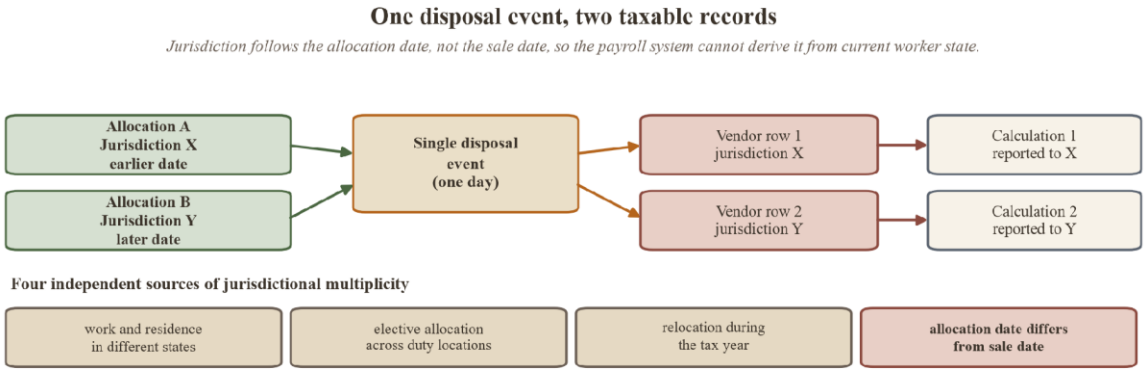
Allocation-date determination. This is the property that most distinguishes equity from ordinary payroll and it deserves separate treatment.

3.2 The allocation date rule and its consequence

For equity compensation, the jurisdiction and withholding treatment are determined by the circumstances at the date the shares were allocated, not the date they were sold.

The operational consequence is that a single sale event can decompose into multiple independently taxable rows. Where a worker received allocations in different jurisdictions on different dates and disposes of both holdings on the same day, the administrator transmits two rows, one per jurisdictional context. Two records are loaded, two calculations are performed, and two reporting obligations arise, from what the worker experienced as one transaction.

This is why the integration cannot treat a sale as a single unit of work, and why a reconciliation that counts sale events rather than transmitted rows will not tie. It is also why the jurisdiction cannot be derived from the worker's current profile: the correct jurisdiction is a historical fact carried in the vendor record, and the payroll system has no independent means of reconstructing it.



The fourth is the one with no ordinary-payroll analogue. In ordinary payroll the applicable jurisdictions are derived from the worker profile as it stands at calculation. Here the determining fact is historical, held by the plan administrator, and must be accepted from the vendor record and mapped. A reconciliation that counts sale events rather than transmitted rows will not tie, because one sale can legitimately produce several rows.

Figure 1. How a single disposal event decomposes into multiple taxable records. The determining fact is the allocation date, held by the administrator, not the worker's profile at sale. This is the reason jurisdiction must be accepted from the vendor record and mapped, rather than derived from current payroll state.

3.3 Vocabulary translation and currency

Beyond jurisdiction, the vendor record expresses earnings categories and deduction types in an encoding specific to the administrator. These have no meaning to the payroll system and must be translated into payroll earnings and deduction codes. States are likewise represented as codes that differ between the two systems.

For Canadian transactions, an additional transformation applies, and it sits alongside a separate remittance and reporting regime [16]: proceeds denominated in United States dollars must be converted at the rate applicable to the processing date before calculation. Currency conversion is therefore part of the mapped input rather than a downstream presentation concern, and an error in it is indistinguishable at the engine from a genuine difference in proceeds.

4. THE AUDIT EVIDENCE MODEL

The audit requirement is not satisfied by logging that an integration ran. It is satisfied by being able to answer, for any value that influenced a tax outcome, who or what placed it there and what it replaced.

Four properties are recorded for every record the integration loads.

Actor attribution. Every record is tagged with the identity that triggered the integration run. Because the load is automated rather than manual, the actor is the initiating identity, which distinguishes integration-loaded records from records entered by hand.

Temporal attribution. Every record carries a date and time stamp applied at load.

Change history. Where a loaded record is subsequently amended by a user, the amending identity is recorded together with the values before and after the change. This is the property that separates an audit trail from a log: the prior value is retained, not overwritten.

Retention. Audit records are retained beyond one year, which spans the year-end reporting cycle in which a defect would most plausibly surface. Retention design for payroll data more generally, including the interaction between retention obligations and purge, is treated in [8].

Recorded property	Applies to	Purpose in an audit
Initiating identity	Every integration-loaded record	Establishes that the value was system-loaded rather than hand-entered, and by whose authority the run occurred
Load timestamp	Every integration-loaded record	Places the value in the processing sequence and ties it to a cycle
Amending identity	Any record subsequently changed	Attributes post-load intervention to a person
Before and after values	Any record subsequently changed	Makes the intervention reconstructable rather than merely noted
Batch identifier	Every integration-loaded record	Binds the record to a reversible set, see Section 7
Retention beyond one year	All of the above	Preserves evidence across the year-end reporting cycle

Table 2. The audit evidence model. The distinguishing element is the retention of prior values on amendment, without which a post-load correction is invisible in its effect.

A practical observation from operating this process is worth recording, because it shapes where effort should be spent. When the process is examined, the first request is not for the record-level load audit. It is for the reconciliation evidence, that is, for proof that what was loaded agrees with what the administrator sent, and that what was calculated agrees with what was loaded. The record-level trail is supporting evidence, consulted when the reconciliation raises a question. Designing the reconciliation as a reportable artifact rather than an operational habit is therefore the higher-value control investment.

5. CONFIGURATION RATHER THAN CODE

The architectural claim of this section is that the jurisdiction-dependent knowledge in this integration belongs in configuration, and that placing it there is what makes the process maintainable and auditable at the pace jurisdictions actually change.

5.1 The configuration surface

Six categories are configured rather than coded.

Reason code. The code tagged against inputs at load, which classifies the input population for downstream reporting and reconciliation.

Check date. The date the payroll process uses for the cycle.

Period date. The period into which the data is loaded, which governs the accounting placement of the transaction.

Earnings translation. The mapping from vendor stock earnings categories to payroll earnings codes.

Deduction translation. The corresponding mapping for deductions, including the codes used for reversals.

Jurisdiction and state translation. The mapping from vendor state and jurisdiction encodings to payroll state codes and the payroll-relevant tax jurisdictions beneath them.

Jurisdictional withholding itself is configured at company level, with rates and percentages held in lookup tables. For each company and state, the applicable jurisdictional taxes and their withholding percentages are configured rather than computed in integration logic.

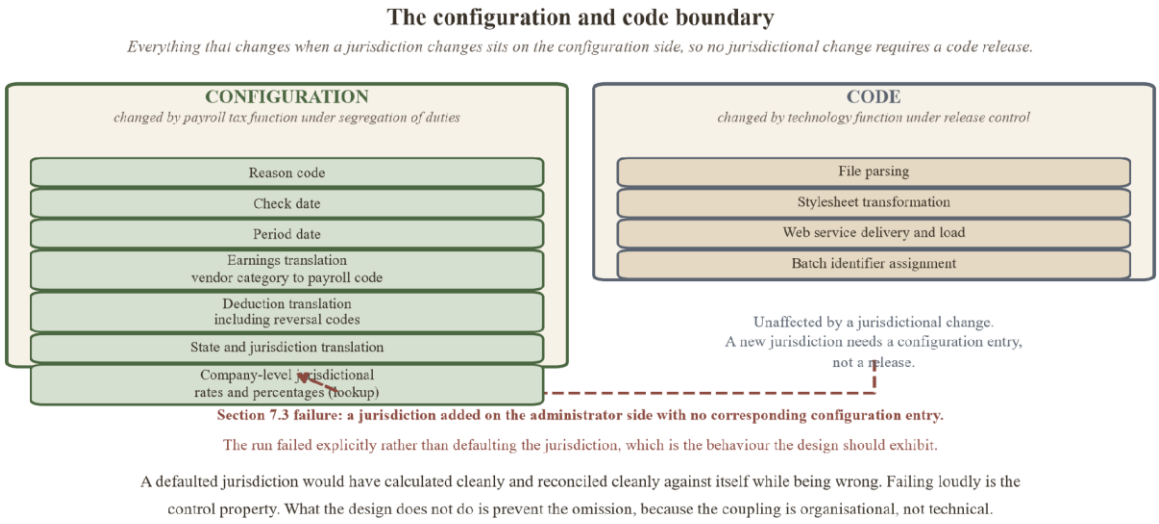


Figure 2. The configuration and code boundary. Everything that changes when a jurisdiction changes sits on the configuration side. The transformation and delivery mechanism sits on the code side and is unaffected by

jurisdictional change. The dashed path is the failure mode of Section 7.3, where a jurisdiction is introduced on the vendor side without a corresponding configuration entry.

5.2 Who may change configuration, and how it is evidenced

Configuration is maintainable by a defined population: designated payroll tax users and the supporting technology function. It is not open to the general payroll user population.

Every configuration update is captured in the audit record on the same terms as a data change, meaning the amending identity and the before and after values are retained. A jurisdictional rate change is therefore evidenced in the same way, and with the same reconstructability, as a correction to a single worker's record.

The control significance is that a jurisdiction change does not require a code release. It does not enter a deployment queue, it does not require regression of the transformation logic, and its review is a review of data rather than of logic. This is what allows the mapping to keep pace with jurisdictional change, which is the practical constraint that determines whether such an integration remains correct over time

Element	Configuration or code	Changed by	Evidenced as
Reason code, check date, period date	Configuration	Payroll tax function	Configuration audit entry with prior value
Earnings and deduction translation	Configuration	Payroll tax function with technology support	Configuration audit entry with prior value
State and jurisdiction translation	Configuration	Payroll tax function with technology support	Configuration audit entry with prior value
Company-level jurisdictional rates	Configuration, lookup table	Payroll tax function	Configuration audit entry with prior value
File parsing and transformation logic	Code	Technology function	Release control
Delivery and load mechanism	Code	Technology function	Release control

Table 3. The configuration and code split, with the change authority and evidence path for each element. The design intent is that no jurisdictional change falls on the code side of this table.

6. TWO-STAGE RECONCILIATION

Reconciliation is performed twice, at two different points, against two different questions. The separation is the design point.

6.1 Stage one, before calculation

The first reconciliation occurs after the integration has loaded inputs and before payroll calculation is triggered.

A report covering off-cycle input for a worker population is executed for the United States and Canadian groups, filtered by the batch identifier assigned to the run. It returns every record the run loaded. The payroll tax function compares that output against the administrator's own transaction report.

The question this stage answers is narrow and therefore decidable: does what was loaded agree with what was sent? It tests transmission and translation. It does not test calculation, because no calculation has occurred.

Placing this stage before calculation is what makes it valuable. A defect found here is corrected by deleting the batch and reloading, at negligible cost, because nothing downstream has consumed the data.

6.2 Stage two, after calculation

The second reconciliation occurs after payroll calculation and before the cycle is completed.

A dedicated audit report returns, for the loaded population, the input amount, the calculated amount, and the difference between them. The payroll tax function reviews differences in a single pass rather than by sampling individual results.

The question this stage answers is also narrow: does what was calculated agree with what was loaded? Because stage one has already established input fidelity, a difference surfacing at stage two is attributable to the calculation path rather than to the input, which is precisely the diagnostic separation that a single combined reconciliation cannot provide.

6.3 Why the separation matters

If reconciliation were performed only once, after calculation, a difference would have two candidate explanations and no means of choosing between them. The input might have been mapped incorrectly, or the calculation might have been configured incorrectly. Distinguishing them would require re-deriving the input from the vendor file by hand, under cycle time pressure.

Two stages remove that ambiguity by construction. Stage one clears the input. Any residual difference at stage two therefore has one candidate explanation, and investigation begins from a known position rather than an open one.

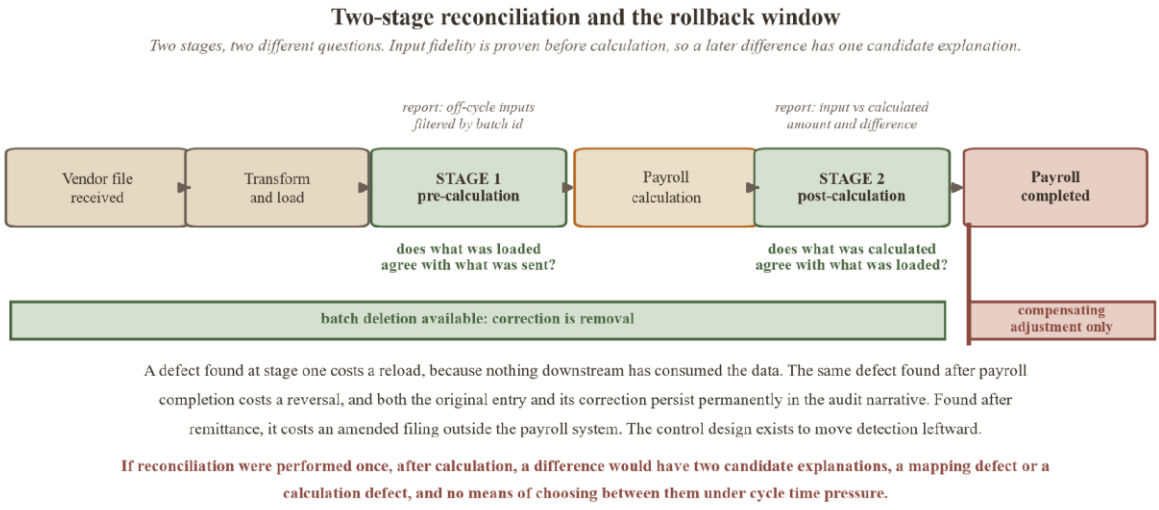


Figure 3. The two-stage reconciliation and the rollback window. Stage one proves input fidelity while correction is still inexpensive. Stage two proves calculation correctness against an already-cleared input. The shaded region marks the interval in which batch deletion remains available; beyond payroll completion, correction requires the compensating path of Section 7.2.

7. BATCH-SCOPED ROLLBACK

7.1 The batch as the unit of reversal

Every integration run generates a unique batch identifier, and every record loaded in that run is tagged with it. The identifier is structural, following the pattern INT FIDELITY STOCKRESTRICTED STOCK FEEDBACK <YYYYMMDDHHMISS>, so that a batch is self-describing as to source and run instant.

Because the identifier is assigned at load and is immutable, it defines a set with two useful properties. The set is complete, containing every record the run produced and no others, and it is addressable, so that operations can be expressed against the set rather than against enumerated records. A dedicated deletion task operates on this set directly.

This is the mechanism that makes reversal safe under time pressure. The operator does not identify which records to reverse. The batch identifier already did.

7.2 The rollback boundary

Reversibility is not unconditional, and stating where it ends is as important as stating that it exists.

Payroll state at time of correction	Available action	Character of the correction
Inputs loaded, not yet processed	Delete by batch identifier	Clean removal, no downstream effect
Inputs processed, payroll in progress	Delete by batch identifier	Clean removal, calculation discarded
Payroll calculation in progress	Cancel calculation, then delete by batch identifier	Clean removal after cancellation
Payroll completed	Deletion unavailable	Compensating adjustment through payroll reversal functionality
Filed or remitted to authority	Deletion unavailable	Amended filing, outside the payroll system

Table 4. The rollback boundary. The transition that matters is payroll completion. Before it, correction is removal. After it, correction is compensation, which produces a different and permanent audit narrative.

Before payroll completion, correction is removal, and the corrected state is indistinguishable from a state in which the defect never occurred. After completion, deletion is no longer available. Correction is then achieved through payroll reversal functionality, which adjusts wages and associated entries. This is a compensating action rather than a rollback: the original entries and their reversal both persist in the record, and the audit narrative permanently contains the defect and its correction.

Once amounts have been remitted or reported to a tax authority, correction moves outside the payroll system entirely and becomes an amended filing, which is the expensive case that the control design exists to avoid.

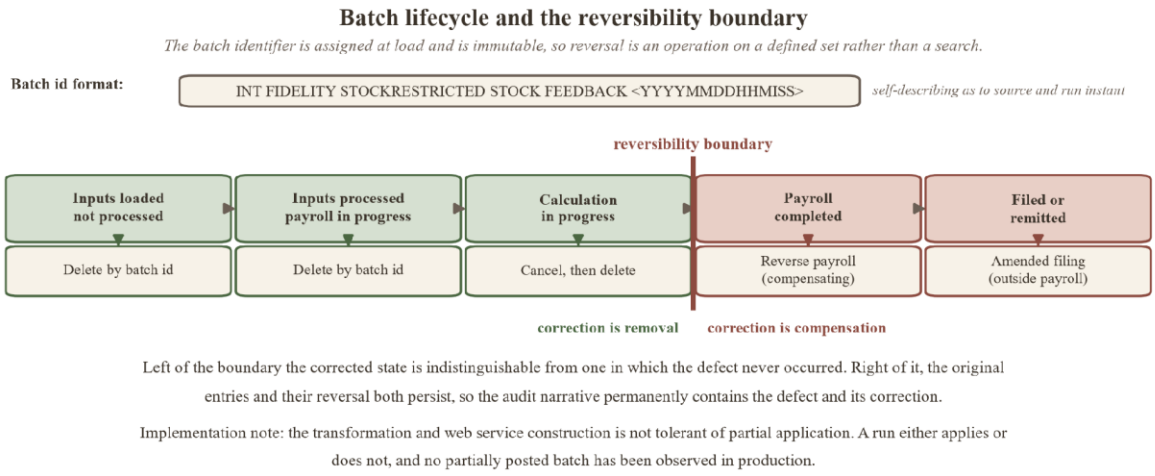


Figure 4. Batch lifecycle and the reversibility boundary. Deletion is available in the states to the left of the boundary. To the right, only compensating adjustment is available, and the correction becomes part of the permanent record rather than replacing it.

7.3 The failure mode encountered in production

The integration was implemented using the platform's transformation toolset with stylesheet-based transformation and web service delivery. That construction is not tolerant of partial application: a run either applies or does not. No partially posted batch has been observed.

The failure mode that did occur is instructive because it falls precisely on the configuration boundary of Section 5. On a subset of rows the integration raised errors, reporting that a deduction code could not be found. New tax jurisdictions had been established on the administrator side without a corresponding entry being added to the integration mapping configuration. Those rows were not silently mishandled; they failed, and the run surfaced the failure.

Two observations follow. First, the failure was of the right kind. An unmapped jurisdiction produced an explicit error rather than a default assignment, which is the behaviour a control design should prefer, since a defaulted jurisdiction would have calculated cleanly and reconciled cleanly while being wrong. Second, the defect was organizational rather than technical. The mapping configuration is coupled to a decision made in another system by another team, and nothing enforced notification of that decision. Resolution was to complete the mapping, and the run then processed normally.

The generalizable point is that the configuration surface described in Section 5 is also a coordination surface. Its correctness depends on a process obligation that no system control enforces.

8. WHAT IS KNOWN, AND WHAT WAS NOT MEASURED

8.1 Operational scale

Measure	Value	Basis
Equity transactions processed since first production run	52,816	Production audit report
Distinct tax jurisdictions covered	More than 90	Production configuration
Geographic coverage	All 50 United States and Canada	Production configuration
Population	All eligible employees of a US enterprise employer of approximately 40,000 staff	Production configuration
Build period	September 2025	Project record
First production cycle	2 October 2025	Project record

Table 5. Operational scale of the production implementation. These are counts drawn from the audit report, not performance measurements, and no control effectiveness claim rests on them.

8.2 What was never instrumented

The following were not captured, and no figure for any of them appears in this paper.

Reconciliation break rate at either stage, and the distribution of break causes. Rollback frequency, and the distribution of rollback across the states in Table 4. Time to close an equity cycle, before or after the architecture was introduced. Audit findings raised and closed. Any before-and-after comparison of error rates, since no equivalent measurement exists for the period preceding the integration.

The absence of a before-state measurement is the more consequential gap. Even had break rates been captured from the first cycle, there is no instrumented baseline for the manual process that preceded it, so an improvement claim would have no comparator.

8.3 What a measurement programme would capture

The audit report already produces the record-level data from which most of these could be derived retrospectively, which makes this tractable rather than aspirational.

Break counts by stage and by cause, separating translation defects from calculation defects, which the two-stage design already distinguishes structurally. Rollback events by payroll state at the time of correction, which would establish how often the completion boundary is actually crossed. Elapsed time from file receipt to cycle completion. Configuration change frequency by category, and the interval between a jurisdictional change on the administrator side and the corresponding configuration update, which is the quantity that would have predicted the Section 7.3 failure. Count of records amended after load, with amending population, which tests whether the automated path is in fact displacing manual intervention.

Reporting these would convert the present architecture contribution into an evaluated one. Until they are captured, the honest statement is the one made here: the architecture is in production and is used, and its effectiveness relative to the alternative has not been measured.

9. LIMITATIONS

Single implementation. The architecture is drawn from one production integration with one stock plan administrator on one payroll platform. The two-stage reconciliation and batch-scoped rollback patterns are platform-independent in principle, but this paper does not demonstrate that.

Rollback depends on a platform primitive. Batch-scoped deletion is available because the platform provides a batch-addressable deletion operation and enforces its own completion boundary. On a platform without an equivalent primitive, the batch identifier would remain useful as an audit construct but would not by itself confer reversibility.

Configuration governance is procedural at its boundary. As Section 7.3 shows, the configuration surface is only as current as the process that notifies it. The architecture makes an unmapped jurisdiction fail loudly rather than silently, which is a real control property, but it does not prevent the omission.

Effectiveness is unmeasured. Stated plainly because it bears on how the contribution should be read: this is an architecture and evidence-model contribution, not a demonstration that the architecture outperforms an alternative.

10. CONCLUSION

Equity compensation payroll is difficult to control for a reason that is architectural rather than computational. The facts that determine tax treatment originate outside the payroll system, in a vocabulary the payroll system does not share, and describe events that may have occurred years before the taxable transaction. The calculation engine cannot detect a defect in what it was given, because a mapping error produces a well-formed input that computes cleanly and reconciles cleanly against itself.

The response set out here is to make correctness demonstrable rather than assumed. Splitting reconciliation into a pre-calculation and a post-calculation stage separates input fidelity from calculation correctness, so that a difference has one candidate explanation rather than two, and so that the cheapest class of defect is caught while correction is still merely a reload. Binding every loaded record to an immutable batch identifier makes reversal an operation on a defined set rather than a search, and makes the boundary at which reversal ceases to be available explicit rather than discovered. Holding jurisdictional translation in configuration under segregation of duties keeps the mapping current at the pace jurisdictions change, and renders each change reviewable on the same evidentiary terms as a data correction.

The production failure described in Section 7.3 is the most useful evidence in the paper, precisely because it is a failure. An unmapped jurisdiction produced an explicit error rather than a defaulted assignment, which is the behaviour the design should exhibit, and the underlying cause was a coordination gap between two organizations rather than a defect in either system. That is a fair characterization of where the residual risk in this class of integration actually sits.

What remains is measurement. The counts reported here establish that the architecture operates at meaningful scale across more than 90 jurisdictions. They do not establish that it works better than what it replaced, and the audit data needed to test that has not yet been analyzed. Section 8.3 states what such an analysis would have to produce.

Note on scope, evidence, and disclosure

Relationship to prior published work. The integration described here was the subject of an earlier paper by the author, which documented how the vendor-to-payroll stock integration was constructed [1]. That paper is an implementation account: bi-directional data movement, file specification, transformation tooling, and the tax withholding path.

The present paper does not restate it. Its subject is the control architecture layered over that integration, specifically the two-stage reconciliation design, the batch-scoped rollback semantics and their boundary, the configuration-versus-code separation, and the audit evidence model. Section 2.3 states the boundary explicitly, claim by claim. Readers seeking the integration mechanics should consult [1]; readers concerned with how the integration is proven correct, reversed, and evidenced should continue here.

What is in production. The integration, the jurisdictional mapping configuration, both reconciliation reports, and the batch deletion mechanism are in production and are used by the payroll tax function on every equity cycle. The integration was built in September 2025 and first processed production data on 2 October 2025.

What is measured. Transaction volume, jurisdiction count, and covered population are drawn from the production audit report. These are counts, not performance measurements.

What is not measured and not claimed. No reconciliation break rate, rollback frequency, cycle-close time, audit-finding count, or error-rate reduction was instrumented. None is estimated or implied anywhere in this paper. Section 8 states what a measurement programme would have to capture. An earlier internal characterization of the process as achieving full compliance is not repeated here, because no measurement basis for such a claim exists.

Confidentiality. The employer is described by function and scale rather than by name. Configuration values, rate tables, and employee-identifying data are excluded. Object names, report names, and the batch identifier format are reproduced because they are structural rather than confidential, and because their specificity is what makes the architecture reproducible by another practitioner.

REFERENCES

1. R. A. Chetty, "Automation of Stock Payroll Functionality in Workday Payroll," International Journal of Innovative Research in Engineering and Management Practices and Science, vol. 14, no. 1, 2026. doi: 10.37082/ijirmps.v14.i1.232921
2. J. R. Kuhn and S. G. Sutton, "Continuous Auditing in ERP System Environments: The Current State and Future Directions," Journal of Information Systems, vol. 24, no. 1, pp. 91-112, 2010. doi: 10.2308/jis.2010.24.1.91
3. K. Singh and P. J. Best, "Design and Implementation of Continuous Monitoring and Auditing in SAP Enterprise Resource Planning," International Journal of Auditing, vol. 19, no. 3, 2015. doi: 10.1111/ijau.12051
4. H. Elbardan and A. O. R. Kholeif, Enterprise Resource Planning, Corporate Governance and Internal Auditing. Cham: Palgrave Macmillan, 2017. doi: 10.1007/978-3-319-54990-3
5. B. Menkus, "How an audit trail aids in maintaining information integrity, as illustrated in retailing," Computers and Security, vol. 9, no. 7, pp. 605-609, 1990. doi: 10.1016/0167-4048(90)90082-5
6. Sarbanes-Oxley Act of 2002, Pub. L. No. 107-204, 116 Stat. 745, Section 404, Management Assessment of Internal Controls.
7. United States Securities and Exchange Commission, Division of Risk, Strategy and Financial Innovation, "Study of the Sarbanes-Oxley Act of 2002 Section 404 Internal Control over Financial Reporting Requirements," 2012. doi: 10.2139/ssrn.2135143
8. R. A. Chetty, "Data Purge Process Model in SaaS Application," International Journal of Leading Research Publication, vol. 6, no. 7, 2025. doi: 10.70528/ijlrp.v6.i7.1680
9. R. A. Chetty, "Data Scramble Process Model in SaaS Application," International Journal on Science and Technology, vol. 16, no. 3, 2025. doi: 10.71097/ijsat.v16.i3.7868

10. R. A. Chetty, "Design and Automation of Dental Benefits Reporting Box 45 in Canada T4," International Journal of Multidisciplinary Research and Growth Evaluation, vol. 6, no. 5, pp. 986-991, 2025. doi: 10.54660/.ijmrge.2025.6.5.986-991
11. R. Chetty, "Implementation of H.R.1 OBBBA in Workday Payroll," International Journal for Multidisciplinary Research, vol. 8, no. 1, 2026. doi: 10.36948/ijfmr.2026.v08i01.70410
12. R. A. Chetty, "Implementation of Secure 2.0 Act, 401K Changes in Workday Payroll," International Journal of Innovative Research in Engineering and Management Practices and Science, vol. 14, no. 1, 2026. doi: 10.37082/ijirmps.v14.i1.232979
13. Committee of Sponsoring Organizations of the Treadway Commission, Internal Control, Integrated Framework. New York: COSO, 2013.
14. Internal Revenue Service, "Publication 15 (Circular E), Employer's Tax Guide," United States Department of the Treasury.
15. Internal Revenue Service, "Publication 525, Taxable and Nontaxable Income," United States Department of the Treasury.
16. Canada Revenue Agency, "T4001 Employers' Guide, Payroll Deductions and Remittances," Government of Canada.