

RESEARCH ARTICLE

Governance-driven enterprise integration architecture for modern HCM and legacy ERP synchronization: a Workday–COINS ERP reference framework**Dipak Makhija***Southland Industries, Enterprise Integration Architect, Agentic AI, AWS, Workato, USA***Corresponding Author:** Dipak Makhija, **E-mail:** deepmh@gmail.com

ABSTRACT

Cloud-based human capital management (HCM) platforms and legacy enterprise resource planning (ERP) systems frequently operate with different data models, identifiers, processing capabilities, and governance responsibilities. These differences create challenges in maintaining consistent workforce information, coordinating employee lifecycle changes, and establishing traceable integration processes. This study develops a governance-driven enterprise integration architecture for synchronizing a modern HCM platform with a legacy ERP system. The proposed framework explicitly defines system-of-record responsibilities, cross-system identity management, data validation and transformation, synchronization and orchestration, exception handling, auditability, and extensibility. The architecture is examined through a case-based scenario evaluation using representative employee lifecycle scenarios, including onboarding, workforce updates, organizational changes, payroll-related changes, and termination. Each scenario is evaluated against predefined architectural criteria covering identity consistency, data integrity, synchronization reliability, governance, auditability, and extensibility. The analysis indicates how explicit governance mechanisms can be incorporated into the integration architecture rather than treated as separate operational controls. The study contributes a reusable reference framework for integrating modern HCM platforms with legacy ERP environments while maintaining clear data ownership, controlled synchronization, and operational traceability. The study also identifies limitations associated with single-case, architecture-focused evaluation and suggest directions for multi-case and empirical validation.

KEYWORDS

enterprise integration architecture; information systems governance; human capital management; legacy ERP; enterprise architecture; data governance; systems integration

ARTICLE INFORMATION

ACCEPTED: 09 August 2026**PUBLISHED:** 30 September 2026**DOI:** 10.32996/fcsai.2026.5.10.3

1. Introduction

Enterprise systems integration has become an important enterprise architecture and information systems management concern as organizations combine cloud-based applications with established enterprise systems. ERP and information-systems integration involves more than technical connectivity; prior research identifies technical, business, and socio-organizational dimensions that influence integration design and implementation (Tobón Valencia et al., 2018). These dimensions become particularly important when applications have different data models, processing capabilities, identifiers, ownership structures, and synchronization requirements.

Enterprise architecture provides a means of representing and coordinating business, information, application, and technology concerns, while information technology governance establishes decision rights, accountability, and alignment between technology and organizational objectives (Hazen et al., 2017; De Haes et al., 2020). Recent research likewise identifies the complementary role of enterprise architecture and IT governance in supporting organizational alignment and integrated technology management (Fuentes-Quijada et al., 2025). Consequently, enterprise integration cannot be treated solely as an interface or message-routing problem; system-of-record responsibilities, information ownership, identity management, data quality, exception handling, and auditability are also architectural and governance concerns.

The problem becomes more significant when a modern HCM platform is integrated with a legacy ERP system that has different technical and functional constraints. A cloud HCM platform may support richer worker lifecycle information and effective-dated business events, while the legacy ERP may continue to provide specialized payroll or financial capabilities. In such environments,

replacing the legacy system may not be practical or desirable. The architectural challenge is therefore to establish controlled interoperability while preserving the responsibilities of each system.

Existing research addresses enterprise architecture, information systems governance, data governance, enterprise integration, and ERP integration as related areas. However, there is a need for an architecture-oriented framework that explicitly connects governance decisions with the operational mechanisms required to synchronize employee lifecycle information between modern HCM and legacy ERP platforms. In particular, system-of-record definitions, cross-system employee identity, validation and transformation, synchronization orchestration, exception handling, and auditability are often addressed as individual concerns rather than as interconnected architectural dimensions.

This study addresses this gap by developing a governance-driven enterprise integration architecture for a Workday–COINS ERP integration context. The architecture establishes Workday as the authoritative source for defined workforce information while preserving COINS responsibilities for payroll and financial processing. It introduces explicit mechanisms for cross-system employee identity mapping, validation and transformation, controlled synchronization, exception handling, monitoring, and auditability.

The study addresses the following research questions:

RQ1. How can governance mechanisms be incorporated into an enterprise integration architecture to support synchronization between modern cloud HCM platforms and legacy ERP systems?

RQ2. How does the proposed framework address identity consistency, data integrity, synchronization reliability, and operational governance in a Workday–COINS integration context?

RQ3. What architectural capabilities and limitations emerge from applying the proposed framework to representative HCM–ERP integration scenarios?

The study makes three contributions. First, it develops a governance-driven reference architecture that integrates data ownership and governance mechanisms with technical integration capabilities. Second, it conceptualizes governed HCM–ERP integration through six architectural dimensions: identity consistency, data integrity, synchronization reliability, governance, auditability, and extensibility. Third, it demonstrates the application of the framework to representative employee lifecycle scenarios in a Workday–COINS integration context.

The remainder of the paper is organized as follows. Section 2 reviews relevant literature concerning enterprise integration architecture, enterprise architecture and information systems governance, HCM–ERP integration, data governance and identity management, and event-driven integration and orchestration. Section 3 describes the research approach, case context, integration problem, evaluation procedure, and evaluation criteria. Section 4 presents the results of the scenario-based architecture evaluation. Section 5 discusses the findings and their implications for enterprise architecture and information systems management, followed by limitations and future research directions. Section 6 concludes the study.

2. Background

2.1 Enterprise integration architecture

Enterprise integration concerns the coordination of information, applications, processes, and technology components across organizational boundaries. Research on ERP and information-systems integration identifies technical, business, and socio-organizational dimensions of integration, indicating that successful integration cannot be reduced to interface connectivity alone (Tobón Valencia et al., 2018). Technical integration concerns the mechanisms through which systems exchange and process information, while business and socio-organizational dimensions concern the processes, responsibilities, and organizational relationships affected by integration.

From an architectural perspective, integration therefore requires explicit decisions about application relationships, information flows, data ownership, transformation, synchronization, and operational responsibilities. Enterprise architecture provides a structured means of representing these relationships and coordinating business and technology concerns. Hazen et al. (2017) describe enterprise architecture as encompassing business processes, information, applications, infrastructure, and management considerations and link architectural capability with organizational agility and performance.

These perspectives suggest that enterprise integration architecture should address both connectivity and responsibility. A technically functional interface may still create architectural problems if the systems involved have ambiguous ownership, inconsistent identifiers, incompatible data semantics, or unclear exception responsibilities. This view is consistent with research that treats enterprise integration as a combination of technical, business, and organizational concerns rather than as a purely technical connectivity problem (Tobón Valencia et al., 2018). The present study extends this perspective by examining how these concerns can be incorporated into a governance-driven architecture for synchronization between cloud HCM and legacy ERP systems.

2.2 Enterprise architecture and information systems governance

Enterprise architecture and information technology governance address related but distinct aspects of enterprise technology management. Enterprise architecture provides representations and relationships across business, information, application, and technology domains, whereas governance establishes structures for decision making, accountability, alignment, and oversight. De Haes et al. (2020) position IT governance as a mechanism for achieving alignment and value from information technology, emphasizing the relationship between governance structures, decision rights, and organizational objectives.

The relationship between enterprise architecture and governance has also received attention in recent research. Fuentes-Quijada et al. (2025), through a systematic mapping study, examined the relationship between enterprise architecture and IT governance and identified business-IT alignment as a central concern. Their review also identified limitations in the empirical validation of proposals in this area. This observation is relevant to the present study because the proposed architecture is not evaluated as an abstract architectural model alone; it is examined through representative integration scenarios.

Within an integration environment, governance decisions become architectural decisions when they determine which system is authoritative for a particular information domain, who owns a data element, which identifier establishes cross-system identity, who is responsible for exceptions, and which system retains responsibility for downstream business processing. These concerns connect governance decision rights with the architectural representation of applications, information flows, and responsibilities (De Haes et al., 2020; Fuentes-Quijada et al., 2025). Accordingly, the present study treats governance as an architectural concern that is expressed through system-of-record definitions, data ownership, identity relationships, processing responsibilities, exception management, and auditability.

2.3 Cloud HCM and legacy ERP integration

The integration of cloud applications with established enterprise systems creates architectural challenges because the systems may have different data models, interface mechanisms, processing capabilities, data ownership arrangements, and synchronization requirements. Research on ERP and information-systems integration demonstrates that integration challenges extend across technical, business, and socio-organizational dimensions (Tobón Valencia et al., 2018). These dimensions are particularly relevant when a modern cloud application and a legacy ERP system perform different business functions and therefore cannot simply be treated as equivalent data repositories.

Cloud HCM platforms commonly support workforce-oriented processes such as worker lifecycle management, organizational information, employment information, and other human-resources processes, whereas ERP environments may retain responsibility for payroll, financial processing, or other downstream enterprise functions. When responsibilities are distributed in this manner, integration architecture must establish which system is authoritative for each information domain rather than assuming that one platform should become the universal source of all data.

Synchronization also introduces temporal and semantic considerations. A change that is represented as an effective-dated business event in one system may need to be represented differently in another system that has different processing or historical-data capabilities. Consequently, integration requires explicit treatment of effective dates, transformation rules, identity relationships, validation, reconciliation, and exception handling.

The present study applies these integration concerns to a case involving a cloud HCM platform and a legacy ERP system. The specific system responsibilities and technical configuration are described in Section 3.2 rather than treated as general properties of HCM or ERP platforms.

2.4 Data governance and identity consistency

Data governance establishes organizational structures and responsibilities for managing information as an enterprise asset. Khatri and Brown (2010) describe data governance in terms of decision domains and accountability for data-related decisions, emphasizing the importance of establishing governance mechanisms for organizational data. Kooper et al. (2011) similarly distinguish information governance as a management concern involving the control and use of information within organizational contexts.

These perspectives are directly relevant to cross-system integration because the movement of information between applications does not by itself establish ownership. A data element may originate in one system, be transformed by an integration platform, and be consumed by several downstream applications. Without explicit ownership and system-of-record decisions, the integration layer can become an implicit source of truth or create uncertainty about which system should be used when discrepancies occur.

Identity consistency is a related concern. When two enterprise systems maintain different identifiers for the same worker or business entity, integration requires an explicit relationship between those identifiers. Such a relationship is not merely a technical mapping; it establishes how the identity of an entity is recognized across system boundaries. The architecture therefore needs to define identifier ownership, mapping rules, validation, and exception handling.

In the framework developed in this study, data governance is operationalized through system-of-record definitions, attribute ownership, cross-system identity mapping, validation and transformation rules, exception responsibilities, and auditability. This

operationalization connects the decision domains identified in data-governance research with concrete architectural mechanisms for managing information across application boundaries (Khatri & Brown, 2010; Kooper et al., 2011). It therefore provides a direct connection between governance principles and integration architecture.

2.5 Integration orchestration, synchronization, and exception handling

Enterprise integration requires more than the transmission of messages between applications. Integration processing must also determine how information is transformed, validated, routed, processed, monitored, and reconciled. Research on enterprise integration emphasizes that integration involves the coordination of technical mechanisms with business processes and organizational responsibilities rather than connectivity alone (Tobón Valencia et al., 2018). These concerns become important when a source system and a downstream system have different processing capabilities or when a business event requires coordinated processing across multiple applications.

Orchestration provides a mechanism for coordinating the sequence of integration activities, while validation and transformation establish whether incoming information can be processed by the receiving system. Exception handling provides a complementary control mechanism for situations in which validation fails, a downstream system rejects information, or processing does not produce the expected business result. Consequently, an integration architecture should distinguish successful message transmission from successful business processing. This distinction is particularly relevant in heterogeneous enterprise environments where technical delivery does not necessarily imply successful completion of the corresponding business transaction.

Within a governance-driven architecture, synchronization and exception handling also have accountability implications. The architecture should identify which system owns the business information, which component performs transformation and routing, which stakeholders are responsible for resolving exceptions, and what processing evidence is retained for audit and reconciliation. The present study incorporates these concerns into the evaluation of the proposed architecture rather than treating orchestration as an isolated technical capability.

2.6 Research gap

The literature reviewed above establishes several relevant perspectives. Research on ERP and information-systems integration identifies technical, business, and socio-organizational dimensions of integration (Tobón Valencia et al., 2018). Enterprise architecture provides a means of coordinating business, information, application, and technology concerns (Hazen et al., 2017), while IT governance addresses alignment, accountability, decision making, and oversight (De Haes et al., 2020; Fuentes-Quijada et al., 2025). Data governance further establishes the importance of ownership and accountability for enterprise information (Khatri & Brown, 2010; Kooper et al., 2011).

However, the literature reviewed addresses these concerns primarily from related but distinct perspectives, including enterprise integration, enterprise architecture, IT governance, and data governance (Tobón Valencia et al., 2018; Hazen et al., 2017; De Haes et al., 2020; Khatri & Brown, 2010; Kooper et al., 2011). Based on the literature reviewed in this study, less explicit attention is given to their combined operationalization within an enterprise integration architecture for synchronizing a modern cloud HCM platform with a legacy ERP system. In particular, the relationship among system-of-record responsibilities, cross-system identity, data validation and transformation, synchronization, exception handling, governance, and auditability is not explicitly represented as a single architectural evaluation framework. This study addresses this gap by developing and applying such a framework within a Workday-COINS integration context.

This study addresses this gap by developing a governance-driven reference framework and evaluating it through five representative integration scenarios: employee onboarding, workforce updates, organizational changes, payroll-related changes, and termination. The objective is not to establish statistical generalizability or quantitative performance improvement, but to examine whether governance concerns can be explicitly represented and evaluated within the architecture.

3. Method

3.1 Research approach

This study uses a case-based architecture evaluation using scenario-based qualitative analysis. The primary research artifact is the governance-driven enterprise integration architecture developed for the case environment. The study therefore evaluates an architectural artifact and its application to representative integration scenarios rather than measuring statistical relationships among variables (Hevner et al., 2004).

The research follows two related activities. First, the architecture was developed from the integration problem, system responsibilities, data ownership requirements, identity requirements, synchronization requirements, and governance considerations identified in the case context and relevant literature. Second, the resulting architecture was evaluated against representative employee lifecycle scenarios.

The architecture was structured around the identified governance and integration requirements and then represented through system-of-record definitions, identity relationships, integration flows, validation and transformation mechanisms, orchestration, exception handling, and auditability.

The evaluation is qualitative and architecture-focused. It does not attempt to estimate statistical performance or establish causal relationships. Instead, it examines whether the architecture provides explicit mechanisms for addressing predefined integration and governance requirements. The unit of analysis is therefore the architectural scenario and its associated integration mechanisms rather than an individual participant or survey response.

This approach enables examination of how architectural mechanisms address a complex integration problem in a specific organizational context. It also makes the distinction between architectural capability and measured operational performance explicit. The study does not claim that the architecture quantitatively improves processing time, failure rates, or financial outcomes; such measures are identified as opportunities for future empirical validation.

3.2 Case context

The case involves an enterprise environment in which a cloud-based HCM platform and a legacy ERP system perform complementary business functions. In the case environment, Workday HCM was designated as the authoritative source for defined workforce information, while COINS retained responsibility for payroll and financial processing. An integration platform was used to validate, transform, route, orchestrate, monitor, and manage exceptions associated with information exchanged between the systems. The integration layer was not designated as a system of record.

Workday serves as the authoritative source for defined workforce information, including worker identity and relevant employment and organizational attributes. COINS remains responsible for payroll and financial processing. An integration platform provides the intermediary capabilities required for data validation, transformation, routing, orchestration, exception handling, and operational monitoring.

The distinction between system ownership and integration responsibility is fundamental to the case. The integration platform does not become the system of record for workforce or financial information. Instead, it provides controlled movement and transformation of information between systems while preserving the functional responsibilities assigned to Workday and COINS.

A further architectural constraint in the case concerns effective-dated information. In the case environment, COINS did not provide the same effective-date handling capabilities available from the HCM environment. Consequently, the integration design had to account for the temporal characteristics of workforce changes when determining how information was extracted, transformed, and delivered to downstream processing.

The case therefore provides a relevant setting for examining governance-driven integration because it combines modern cloud HCM capabilities, legacy ERP constraints, multiple identifiers, employee lifecycle changes, and distinct system-of-record responsibilities.

3.3 Integration problem

The central integration problem is the controlled synchronization of workforce information between systems that have different data models, identifiers, processing responsibilities, and temporal capabilities.

Five architectural problems were identified:

1. **System-of-record ambiguity:** different systems may contain overlapping workforce attributes, creating uncertainty concerning which value is authoritative.
2. **Cross-system identity:** the HCM and ERP platforms maintain different representations of employee identity, requiring an explicit relationship between the authoritative worker record and the corresponding ERP employee identifier.
3. **Data transformation and validation:** workforce attributes must be mapped and validated before being processed by the receiving system because the source and target systems do not necessarily share the same data structures or business rules.
4. **Synchronization and exception handling:** successful transmission of an integration message does not guarantee successful business processing. The architecture must therefore distinguish normal processing from rejected or incomplete transactions.
5. **Governance and auditability:** ownership, processing responsibility, exceptions, and integration status must be sufficiently explicit to support operational accountability and traceability.

These problems form the basis for the architecture evaluation described in the following sections.

3.4 Evaluation procedure

The architecture was evaluated using a structured scenario-based procedure. Five representative employee lifecycle scenarios were examined: employee onboarding, workforce updates, organizational changes, payroll-related changes, and termination. The scenarios were selected as representative architectural event classes relevant to the integration problem examined in the case and involving synchronization between HCM and ERP environments.

For each scenario, the analysis followed a consistent sequence:

1. identify the initiating business event;
2. identify the authoritative source for the affected information;
3. identify the relevant employee identity relationship;
4. identify the data elements requiring synchronization;
5. trace validation and transformation requirements;
6. trace the synchronization and orchestration path;
7. identify exception and failure-handling mechanisms;
8. identify governance responsibilities and ownership boundaries; and
9. identify the audit or processing evidence produced by the architecture.

Each scenario was then assessed against the evaluation criteria defined in Section 3.5.

The evaluation focuses on architectural evidence contained in the system-of-record definitions, integration flows, identity mappings, data mappings, validation and transformation rules, orchestration mechanisms, exception-handling paths, governance responsibilities, and representative scenarios. The analysis determines whether each architectural concern is explicitly addressed, partially addressed, or not demonstrated by the available architecture evidence.

No interviews, surveys, statistical sampling, or experimental performance measurements are claimed as part of this evaluation. The purpose of the evaluation is to determine how the proposed architecture addresses the identified integration and governance requirements within the case context.

The evaluation criteria were applied as qualitative architectural assessment dimensions rather than quantitative performance indicators. No participant ratings, survey scores, statistical tests, or production performance percentages were generated as part of the evaluation.

3.5 Evaluation criteria

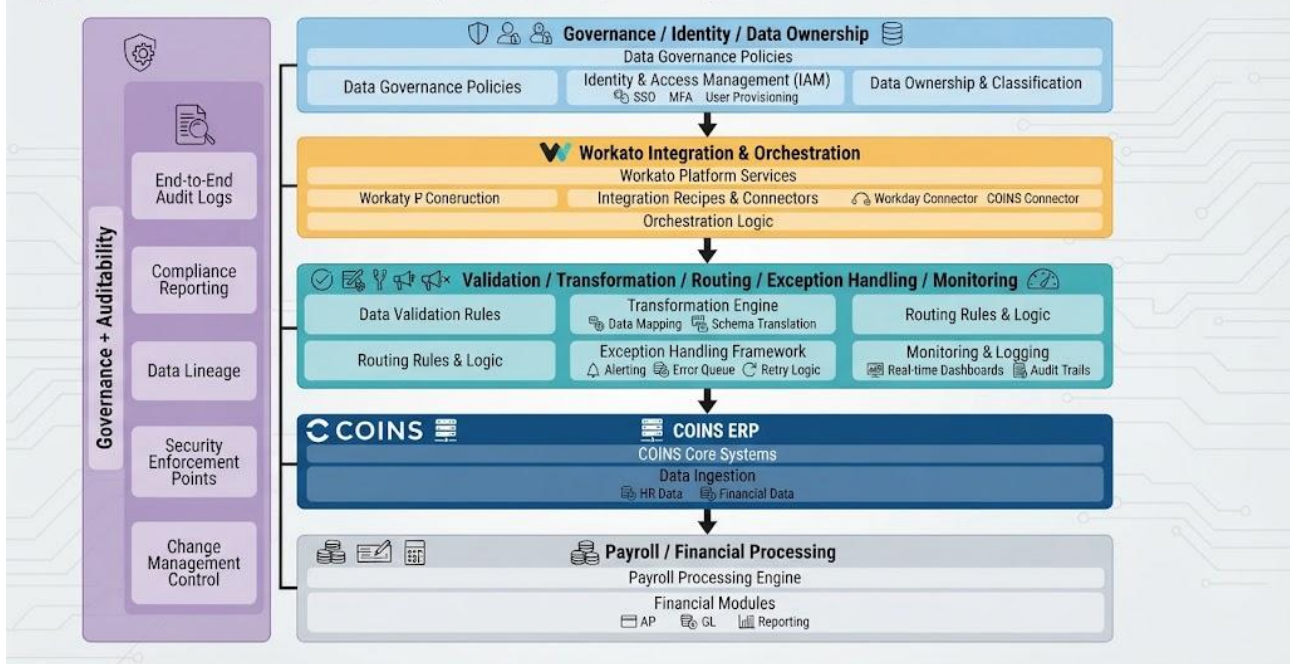
The architecture was evaluated using six qualitative criteria derived from the integration and governance concerns identified in the case: identity consistency, data integrity, synchronization reliability, governance, auditability, and extensibility. The criteria were used to determine whether the architecture explicitly addressed the relevant architectural concern in each scenario. They were not treated as quantitative key performance indicators, and no numerical performance score was assigned.

Criterion	Evaluation question
Identity consistency	Does the architecture establish an explicit relationship between worker identity in the HCM platform and employee identity in the ERP system?
Data integrity	Does the architecture provide mechanisms for validation, transformation, and controlled handling of workforce information?
Synchronization reliability	Does the architecture define controlled processing, orchestration, exception handling, and reconciliation mechanisms?
Governance	Are system-of-record responsibilities, ownership boundaries, and processing responsibilities explicitly defined?
Auditability	Does the architecture provide mechanisms for tracing processing status, exceptions, and integration activity?
Extensibility	Can the architectural mechanisms be reused or extended for additional employee lifecycle scenarios or downstream systems?

These criteria were derived from the integration problems identified in Section 3.3 and the governance and architecture concepts discussed in Section 2. The criteria are used as qualitative analytical dimensions rather than as a numerical scoring system.

3.6 Evidence used

Figure 1. Governance-driven Workday–COINS enterprise integration architecture



The evaluation uses architecture-level evidence from the case, including system-of-record definitions, application and integration architecture, employee identity mappings, data mappings, validation and transformation requirements, integration flows, representative lifecycle scenarios, and exception-handling mechanisms.

The evidence was analyzed by tracing each scenario through the proposed architecture and determining whether the six evaluation criteria were explicitly represented in the architecture. The analysis emphasizes architectural mechanisms and responsibilities rather than subjective assessments of stakeholder satisfaction.

The evidence has two important limitations. First, the study represents a single organizational case and therefore does not establish statistical generalizability. Second, the evaluation is architecture-focused and does not include longitudinal production measurements of processing time, failure rates, reconciliation effort, or operational cost. These limitations are considered when interpreting the findings.

4. Results

4.1 Evaluation results by scenario

The scenario-based evaluation examined five representative employee-lifecycle event classes: onboarding, workforce updates, organizational changes, payroll-related changes, and termination. These scenarios were selected as architectural event classes that exercise the major integration mechanisms identified in the Method section rather than as a statistically sampled population of business events. Across the scenarios, the evaluation identified a consistent architectural pattern: Workday served as the authoritative source for defined workforce information, COINS retained responsibility for payroll and financial processing, and the integration layer performed controlled validation, transformation, routing, orchestration, and exception handling.

Table 1 summarizes the architectural mechanisms identified through the scenario evaluation. The classifications describe the presence of architectural mechanisms in the case evidence and should not be interpreted as quantitative performance scores.

Table 1. Evaluation of representative employee lifecycle scenarios

Scenario	Identity consistency	Data integrity	Synchronization reliability	Governance	Auditability	Extensibility
Employee onboarding	Explicit identity mapping	Mandatory-field validation and transformation	Controlled orchestration and exception handling	Workday/COINS responsibility boundaries	Processing status and exception traceability	Reusable lifecycle pattern
Workforce update	Employee identity	Attribute-level validation/transformation	Change propagation	Attribute ownership defined	Change processing traceability	Applicable to additional

Scenario	Identity consistency	Data integrity	Synchronization reliability	Governance	Auditability	Extensibility
	maintained across systems		and exception handling			workforce attributes
Organizational change	Worker identity retained	Organizational/effective-date validation	Controlled propagation	Workday ownership of workforce organization data	Change traceability	Reusable for organizational lifecycle events
Payroll-related change	Worker/employee relationship retained	Validation of payroll-relevant workforce attributes	Controlled delivery to COINS	COINS payroll responsibility retained	Processing/reconciliation evidence	Applicable to additional payroll-relevant attributes
Termination	Identity retained through lifecycle	Lifecycle/status validation	Controlled termination propagation	Ownership boundaries maintained	Exception and processing traceability	Reusable for other lifecycle closure events

4.2 Identity management results

The scenario analysis identified cross-system employee identity as a distinct architectural concern. The proposed architecture separates the authoritative worker identity maintained by Workday from the corresponding employee identifier used by COINS. Rather than treating the identifiers as interchangeable values, the architecture establishes an explicit relationship between them. This distinction is relevant across the employee lifecycle because the same worker may be represented in multiple systems using different identifiers. Maintaining an explicit identity relationship provides a consistent basis for associating onboarding, workforce changes, organizational changes, payroll-related information, and termination events with the corresponding ERP employee record.

The analysis therefore indicates that identity mapping functions as both a technical integration mechanism and a governance mechanism. It establishes which system owns the authoritative workforce identity while defining how that identity is represented in downstream processing. The evaluation did not measure identity-related error rates; consequently, the finding is limited to the architectural capability demonstrated by the case.

4.3 Data validation and effective-dated synchronization

The evaluation identified data validation and temporal synchronization as important architectural concerns in the Workday–COINS integration. Workforce information originating in Workday must be transformed into structures and values that can be processed by COINS. The integration layer therefore provides a controlled location for validation, transformation, and routing before information is delivered to the ERP.

A specific case constraint concerns the handling of effective-dated workforce changes. The legacy ERP does not provide the same effective-date capabilities available in the HCM environment. Consequently, the integration design must preserve the temporal meaning of workforce changes while delivering information in a form that can be processed by the receiving system.

A specific architectural decision concerned the selection of the Workday integration mechanism for effective-dated workforce changes. In the case configuration, PECO was selected for the relevant integration requirement because its output characteristics aligned more closely with the requirement to communicate effective-dated workforce information to downstream processing. PECL was not selected for this integration because its change-oriented exchange pattern would have required additional downstream coordination in the case environment to satisfy the required processing behavior. The selection was therefore treated as a context-specific architectural decision based on the processing requirements and capabilities of the participating systems rather than as a general assertion that one Workday integration mechanism is universally preferable to another.

This finding illustrates an important role for architecture governance. The interface mechanism is not selected solely on the basis of technical availability. It is evaluated against the business requirement, target-system capability, data semantics, and operational

processing model. The evaluation therefore treats the PECO/PECI decision as an architectural decision governed by the characteristics of the integration problem rather than as a simple interface configuration choice.

4.4 Synchronization and exception-handling results

The scenario evaluation distinguishes message transmission from successful business processing. For each lifecycle scenario, the architecture identifies the processing path from the originating HCM event through validation, transformation, orchestration, and delivery to the ERP environment.

The integration layer provides a controlled location for coordinating these activities. Where a transaction cannot be processed successfully, the architecture provides an exception path through which the issue can be identified and managed rather than treating the failed transaction as equivalent to successful synchronization.

This distinction is particularly important for employee lifecycle events because a technically successful message exchange may still result in a business-rule or data-validation failure in the receiving system. The evaluation therefore considers exception handling part of synchronization reliability rather than an independent operational concern.

The analysis indicates that synchronization reliability in the proposed framework is defined architecturally through controlled processing, exception handling, and traceability. No production failure-rate data were available for this evaluation, so the results do not establish a quantitative improvement in reliability.

4.5 Governance and auditability results

The evaluation identified explicit system-of-record definitions as a central governance mechanism. Workday is responsible for authoritative workforce information within the defined scope, while COINS retains responsibility for payroll and financial processing. This separation helps prevent the integration layer from implicitly becoming the owner of information simply because it performs transformation or routing.

The architecture also associates processing responsibilities with the relevant system and integration components. Data ownership remains with the designated system of record, while the integration platform provides the mechanisms required to transport, transform, validate, and coordinate information.

Auditability is addressed through the ability to associate integration processing with the originating business event, target processing, and exception state. This provides an architectural mechanism for tracing the status of a synchronization process. The finding concerns the existence of traceability mechanisms in the architecture and does not represent a measured assessment of operational audit performance.

The results therefore indicate that governance and auditability are embedded within the integration design through system-of-record definitions, responsibility boundaries, processing visibility, and exception traceability.

4.6 Consolidated evaluation results

The scenario analysis identified six recurring architectural dimensions across the evaluated employee lifecycle scenarios.

Evaluation dimension	Architectural mechanism	Evaluation finding
Identity consistency	Cross-system employee identity mapping	The architecture explicitly represents the relationship between Workday worker identity and the COINS employee identifier.
Data integrity	Validation and transformation	The integration layer provides a controlled mechanism for validating and transforming workforce information before downstream processing.
Synchronization reliability	Orchestration and exception handling	The architecture distinguishes normal processing from exception conditions and provides controlled processing paths.
Governance	System-of-record definitions and responsibility boundaries	Workday and COINS responsibilities are explicitly separated according to their designated business roles.
Auditability	Processing status and exception traceability	The architecture provides mechanisms for tracing integration processing and exceptions.
Extensibility	Reusable integration and governance patterns	The architecture is structured so that the identified mechanisms can be reused across multiple employee lifecycle scenarios and considered for additional integration requirements.

The consolidated evaluation indicates that the proposed architecture addresses the six evaluation dimensions across the representative scenarios. Identity consistency is addressed through explicit cross-system identity relationships; data integrity through validation and transformation; synchronization through controlled orchestration and exception handling; governance

through explicit system-of-record and ownership responsibilities; auditability through processing and exception traceability; and extensibility through reusable integration mechanisms. These findings represent qualitative architectural observations within the case and should not be interpreted as quantitative evidence of production performance improvement.

At the same time, the evaluation does not establish quantitative operational improvements. The results indicate the architectural mechanisms present in the case and their applicability to representative scenarios. Measurement of production performance, including failure rates, reconciliation effort, processing time, and operational cost, remains outside the scope of the present study.

5. Discussion

5.1 Interpretation of findings

The findings indicate that governance and integration architecture are closely connected when information is distributed across systems with different business responsibilities. The architecture does not treat governance as a separate policy layer applied after integration design. Instead, governance is expressed through architectural decisions concerning system-of-record responsibility, identity relationships, data ownership, validation, synchronization, exception handling, and auditability.

This finding is consistent with the literature on data governance, which emphasizes explicit decision domains and accountability for organizational data (Khatri & Brown, 2010), and with information governance research emphasizing the management and control of information across organizational contexts (Kooper et al., 2011). It also aligns with the broader enterprise governance perspective described by De Haes et al. (2020), in which governance provides structures for alignment, decision making, and oversight.

The findings further extend the integration perspective identified by Tobón Valencia et al. (2018). Their literature review demonstrates that ERP and information-systems integration has technical, business, and socio-organizational dimensions. The present framework connects these dimensions within a specific architectural structure by associating technical integration mechanisms with system ownership, identity, governance responsibilities, and operational controls.

5.2 Comparison with conventional integration approaches

Conventional integration designs frequently emphasize connectivity, interface contracts, data transformation, and message delivery. These mechanisms remain necessary, but the findings from the present study indicate that they do not by themselves establish responsibility for the information being exchanged. The governance-driven approach developed in this study places additional emphasis on the relationship between technical integration and organizational responsibility.

The distinction can be summarized as follows. Conventional integration emphasizes moving information between systems, whereas the proposed framework additionally defines which system owns the information, how identities are related, which validation rules apply, who is responsible for exceptions, and what processing evidence should be retained. Similarly, transformation is treated not simply as a technical mapping activity but as an activity constrained by data ownership and system-of-record responsibilities.

This perspective is consistent with enterprise architecture research emphasizing the coordination of business and technology concerns (Hazen et al., 2017) and with research connecting enterprise architecture and IT governance to business-IT alignment (Fuentes-Quijada et al., 2025). The proposed framework therefore extends conventional integration design by making governance relationships explicit within the architecture rather than treating them as external management activities.

5.3 Implications for enterprise architecture

The study has several implications for enterprise architecture practice. First, system-of-record decisions should be treated as explicit architectural decisions rather than informal application-level assumptions. Explicit ownership definitions provide a foundation for determining which information should be propagated, which system is authoritative when values conflict, and which application is responsible for maintaining the authoritative value (Hazen et al., 2017; De Haes et al., 2020).

Second, cross-system identity should be represented as an architectural concern. Employee identifiers affect the ability to maintain lifecycle continuity across applications, and identity mappings should therefore be governed alongside application interfaces and data mappings.

Third, integration architecture should incorporate operational controls. Validation, exception handling, monitoring, and auditability are not merely implementation details when integration processes support critical workforce and payroll-related activities. They influence accountability and operational governance.

Finally, the case illustrates the importance of aligning integration mechanisms with target-system constraints. The PECO/PECI decision demonstrates that interface selection can depend on temporal data requirements and target-system capabilities rather than on interface technology alone.

5.4 Implications for information systems management

For information systems management, the findings emphasize the importance of clearly assigning responsibilities across applications and integration components. When multiple systems participate in a business process, ambiguity concerning ownership can result in duplicated data maintenance, inconsistent values, or unclear responsibility for exceptions.

The proposed framework provides a structured basis for defining these responsibilities. Business and technology stakeholders can use the system-of-record model to determine which application owns particular data, while the integration architecture defines how that information is validated, transformed, and propagated.

The framework also supports operational coordination by making exception paths explicit. Integration failures can involve data quality, business rules, application availability, or configuration conditions, and responsibility for resolving these conditions may differ. Associating exception handling with defined ownership and traceability can therefore support clearer operational accountability.

These implications may also be considered in other heterogeneous enterprise integration settings where applications maintain overlapping information and different processing responsibilities. However, validating the transferability of the framework beyond HCM–ERP integration requires additional cases and empirical investigation.

5.5 Generalizability and limitations

Several limitations should be considered when interpreting the findings. First, the study examines a single enterprise integration context involving a cloud HCM platform and legacy ERP environment. The findings therefore should not be interpreted as statistically generalizable across organizations or HCM–ERP implementations.

Second, the evaluation is architecture-focused and scenario-based. The study does not report participant-based qualitative evaluation, statistical analysis, production reliability measurements, transaction-volume analysis, or longitudinal performance measurements. The six evaluation criteria were used as qualitative architectural assessment dimensions rather than quantitative performance indicators.

Third, the five scenarios represent selected employee-lifecycle event classes and are not intended to exhaustively represent all workforce, payroll, organizational, or downstream integration events. Additional scenarios may reveal architectural requirements that are not captured by the present evaluation.

The framework is therefore intended to provide analytically transferable architectural principles rather than statistically generalizable findings. Future research should evaluate the framework across multiple organizations and integration architectures and should incorporate quantitative production measures, longitudinal observations, and comparative evaluations of alternative integration designs.

6. Conclusion

This study examined the architectural challenges associated with synchronizing workforce information between a modern cloud HCM platform and a legacy ERP system. The case demonstrates that such integration involves more than technical connectivity. Differences in system responsibilities, data models, identifiers, effective-date capabilities, and processing rules require explicit architectural governance.

The study developed a governance-driven enterprise integration architecture that connects system-of-record definitions with cross-system identity management, data validation and transformation, synchronization and orchestration, exception handling, auditability, and extensibility. The architecture was evaluated through representative employee lifecycle scenarios covering onboarding, workforce updates, organizational changes, payroll-related changes, and termination.

The evaluation identified six recurring architectural dimensions: identity consistency, data integrity, synchronization reliability, governance, auditability, and extensibility. The findings indicate that these dimensions are interconnected. System-of-record decisions establish ownership; identity management maintains relationships between representations of the same worker; validation and transformation control information exchanged between systems; orchestration and exception handling coordinate processing; and auditability provides traceability of integration activity.

The Workday–COINS case also illustrates the importance of aligning integration design with the capabilities and constraints of participating systems. In particular, the treatment of effective-dated workforce information and the selection of an appropriate Workday integration mechanism demonstrate that interface decisions should be evaluated against business requirements, data semantics, target-system capabilities, and governance responsibilities.

The study contributes a reference framework for analyzing governed HCM–ERP integration and provides an architecture-oriented perspective that connects enterprise integration with information systems governance. For practitioners, the framework provides a structured basis for defining system ownership, employee identity relationships, validation responsibilities, synchronization mechanisms, and exception-handling processes.

The study is limited by its single-case and architecture-focused evaluation. It does not establish quantitative improvements in operational performance and does not claim statistical generalizability. Future research should therefore validate the framework across multiple organizations and technology environments and examine measurable outcomes such as synchronization failures, processing time, reconciliation effort, exception volumes, and operational workload.

Overall, the study indicates how governance considerations can be embedded within enterprise integration architecture to provide a structured approach to synchronizing modern HCM and legacy ERP environments while preserving clear system

responsibilities and operational traceability. The contribution is therefore an architecture-oriented reference framework that can be examined and further validated in other enterprise integration contexts.

Funding: Funding information was not provided by the author.

Conflicts of Interest: Conflict-of-interest information was not provided by the author.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers.

References

- [1] Tobón Valencia, E., Lamouri, S., Pellerin, R., Dubois, P., & Moeuf, A. (2018). The integration of ERP and inter-intra organizational information systems: A literature review. *IFAC-PapersOnLine*, 51(11), 1212–1217. <https://doi.org/10.1016/j.ifacol.2018.08.425>
- [2] De Haes, S., Van Grembergen, W., Joshi, A., & Huygh, T. (2020). *Enterprise governance of information technology: Achieving alignment and value in digital organizations* (3rd ed.). Springer. <https://doi.org/10.1007/978-3-030-25918-1>
- [3] Fuentes-Quijada, G., Ruiz-González, F., & Caro, A. (2025). Enterprise architecture and IT governance to support the BizDevOps approach: A systematic mapping study. *Information Systems Frontiers*, 27, 865–888. <https://doi.org/10.1007/s10796-024-10473-2>
- [4] Hazen, B. T., Bradley, R. V., Bell, J. E., In, J., & Byrd, T. A. (2017). Enterprise architecture: A competence-based approach to achieving agility and firm performance. *International Journal of Production Economics*, 193, 566–577. <https://doi.org/10.1016/j.ijpe.2017.08.022>
- [5] Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Quarterly*, 28(1), 75–105.
- [6] Khatri, V., & Brown, C. V. (2010). Designing data governance. *Communications of the ACM*, 53(1), 148–152. <https://doi.org/10.1145/1629175.1629210>
- [7] Kooper, M. N., Maes, R., & Roos Lindgreen, E. E. O. (2011). On the governance of information: Introducing a new concept of governance to support the management of information. *International Journal of Information Management*, 31(3), 195–200. <https://doi.org/10.1016/j.ijinfomgt.2010.05.009>

Appendix A – Representative HCM–ERP integration scenarios

The scenarios in this appendix represent architectural event classes used to evaluate the proposed integration framework. They were selected to exercise the principal identity, data, synchronization, governance, auditability, and extensibility mechanisms identified in the Method section. They are not statistically sampled business events and are not intended to represent the complete population of employee-lifecycle transactions.

Table A1. Representative employee lifecycle integration scenarios

Scenario	Business event	Primary source system	Target system	Key integration concern	Governance/control consideration
A1	Employee onboarding	Workday HCM	COINS ERP and downstream systems	Creation and propagation of a consistent employee identity	Employee identifier mapping, mandatory-field validation, exception handling and auditability
A2	Workforce information update	Workday HCM	COINS ERP and downstream systems	Synchronization of changes to worker attributes	Change detection, transformation, validation and exception handling
A3	Organizational change	Workday HCM	COINS ERP and related systems	Synchronization of organizational, supervisory or job-related changes	Effective-date handling, data ownership and controlled propagation
A4	Payroll-related change	Workday HCM for workforce attributes defined as	COINS ERP	Accurate transmission of payroll-relevant workforce information	System-of-record definition, validation and reconciliation

Scenario	Business event	Primary source system	Target system	Key integration concern	Governance/control consideration
		authoritative in the case			
A5	Employee termination	Workday HCM	COINS ERP and downstream systems	Consistent termination status across dependent systems	Lifecycle-state validation, controlled propagation and exception management

The scenarios were evaluated against the six architectural criteria defined in Section 3.5: identity consistency, data integrity, synchronization reliability, governance, auditability and extensibility.

The evaluation focused on whether the architecture provides explicit mechanisms for identifying the authoritative source, maintaining cross-system identity relationships, validating data before transmission, handling integration exceptions, and maintaining traceability of integration activity.

The scenarios do not represent a statistically sampled population of employees. They are representative architectural scenarios selected to examine the behavior and governance requirements of common HCM–ERP workforce lifecycle events.

Appendix B – Data ownership and integration governance matrix

The following matrix summarizes the principal data-governance responsibilities considered in the Workday–COINS integration architecture.

Table B1. Data ownership and governance responsibilities

Data / information domain	Authoritative source	Primary consumers	Integration governance mechanism
Worker identity	Workday	COINS and downstream systems	Cross-system employee ID mapping
Worker personal information	Workday	COINS/downstream systems	Attribute-level validation and transformation
Organizational information	Workday	COINS/downstream systems	Controlled propagation and effective-date handling
Job and employment information	Workday	COINS/downstream systems	Attribute ownership and validation
Payroll-processing information	Preserve COINS payroll-processing responsibility while controlling and validating the upstream workforce attributes supplied to payroll processing	Payroll and financial processes	Preserve COINS processing responsibility while controlling the upstream workforce attributes supplied to payroll processing
Financial information	COINS	Financial processes	COINS remains authoritative
Integration status and exceptions	Integration platform	Architecture and operations stakeholders	Processing status, exception tracking, auditability

The matrix illustrates that integration does not require a single system to own all workforce-related information. Instead, the architecture establishes explicit ownership boundaries and defines how information is exchanged between systems while preserving the responsibilities of each platform.

These ownership boundaries are particularly relevant in hybrid environments where a modern cloud HCM platform and a legacy ERP system continue to coexist. The matrix illustrates that integration does not require a single system to own all information involved in a business process. Instead, the architecture establishes explicit ownership boundaries and defines how information is exchanged between systems while preserving the processing responsibilities assigned to each platform. These boundaries are treated as case-specific architectural decisions and should be reassessed when business responsibilities, application capabilities, or integration requirements change.