IJEIT
International Journal of Emerging Information Technology
ISSN: 3139-1419 / Volume 1 Issue 2 November - December 2025 / Page No: 7-13
Paper Id: IJEIT-V1I2P102

Original Article

The Power of Interoperability: Designing Custom Applications for Seamless Integration

Saikrishna Tarakampet[1], Ganesh Puvvula[2], Sameena Begum Savukath Ali[3], Subhash Tatavarthi[4], Venkatasatyaravikiran Bikkavolu[5]

[1,2,3,4,5]Saikrishna Tarakampet, Enterprise Platform Architect, CCHCS (ATG Corp), Celina TX, USA.

Received: 24 October 2025 Revised: 25 November 2025 Accepted: 11 December 2025 Published: 29 December 2025

Abstract: *The object of this research was enterprise interoperability across distributed enterprise applications and cloud platforms. Modern organizations operate complex ecosystems composed of customer relationship management systems, enterprise resource planning systems, human capital management platforms, and IT service management environments. The problem addressed in this study was the growing operational complexity caused by tightly coupled point-to-point integrations between enterprise systems.*

Traditional integration architecture relied on direct interfaces between applications. These integrations were difficult to scale and maintain. Schema changes, authentication differences, and API evolution frequently caused integration failures.
To address this challenge, the research proposed an interoperability framework implemented through a Universal Integration Adapter (UIA). The architecture was developed using ServiceNow orchestration capabilities and API integration mechanisms.

The results demonstrated significant improvements in operational performance. Integration latency decreased from approximately 4 seconds to less than 1 second. Integration development cycles decreased from six months to less than one month. Operational maintenance costs decreased by approximately 90 percent.
These improvements were achieved by abstracting integration logic into reusable orchestration workflows and centralized governance mechanisms. The distinctive feature of the proposed framework is the separation of orchestration logic from application interfaces.

The proposed framework can be applied in enterprise environments integrating Salesforce, SAP, Workday, and ServiceNow platforms. Organizations implementing digital transformation strategies may use the architecture to reduce integration complexity while improving reliability and scalability.

Keywords: *Interoperability, Enterprise Integration, ServiceNow, API Integration, Orchestration, Custom Applications, System Integration.*

1. Introduction

Modern enterprises operate complex digital ecosystems consisting of multiple interconnected platforms. These platforms typically include customer relationship management systems, enterprise resource planning systems, human capital management platforms, and IT service management environments [1].

Each platform provides functionality for specific business operations. However, interoperability between these systems often presents significant challenges.

Enterprise integration architectures frequently rely on point-to-point interfaces between applications, which increases architectural complexity and maintenance overhead [2, 3]. Each new system requires additional interfaces to multiple existing platforms.

This approach increases architectural complexity and operational risk, particularly when enterprise systems evolve independently and require frequent interface adjustments [4, 5]. Integration failures often occur due to schema evolution, authentication mismatches, and API version incompatibility [6].

Industry research estimates that organizations spend more than $500 billion annually on integration-related development and maintenance activities [1].

Although middleware and integration platforms have been developed to address these challenges, several issues remain unresolved:

- lack of standardized integration architecture;
- excessive dependence on custom development;
- limited resilience to API changes;
- absence of centralized governance.

These challenges highlight the need for scalable integration frameworks that enable sustainable interoperability. Enterprise integration has evolved significantly from tightly coupled architectures to more flexible service-oriented and microservices-based approaches. Early research highlighted the limitations of direct system integrations, emphasizing the need for structured communication patterns [2].

Enterprise Integration Patterns introduced reusable architectural models for messaging and interoperability [3]. These models improved system communication but did not fully eliminate complexity in large-scale enterprise environments.

Microservices architectures improved system modularity and scalability but introduced orchestration and governance challenges [7, 8]. These challenges require centralized coordination mechanisms to ensure system reliability.

Modern integration platforms, such as ServiceNow Integration Hub, provide orchestration capabilities that reduce dependency on custom code [9]. However, these platforms lack standardized abstraction layers that enable reusable and scalable integration frameworks.

This research addresses these gaps by proposing a Universal Integration Adapter architecture that integrates orchestration, abstraction, and governance into a unified model.

Object of Research. Enterprise interoperability across distributed enterprise systems.
Purpose of Research. The purpose of this research was to design and evaluate an interoperability framework capable of improving integration scalability and operational reliability.
Research Tasks. To achieve the research purpose, the following tasks were performed.

1. To analyze the limitations of traditional point-to-point enterprise integration architectures.
2. To develop an orchestration-based interoperability framework implemented through the Universal Integration Adapter within the ServiceNow platform.
3. To implement the proposed architecture in an enterprise integration scenario connecting Salesforce with ServiceNow Project Portfolio Management.
4. To evaluate the operational performance of the framework using quantitative integration metrics.

2. Materials and Methods

The research employed a mixed methodology combining qualitative and quantitative analysis.

In the first phase, *47 enterprise integration implementations* were analyzed across industries including financial services, healthcare, and technology.

Integration execution logs from ServiceNow Integration Hub (ServiceNow Inc., USA) were examined to identify common operational failures and performance bottlenecks [4].

API monitoring tools were used to identify latency delays and integration errors.
In the second phase, prototype integration adapters were developed for:

- Salesforce CRM (Salesforce Inc., USA);
- SAP ERP (SAP SE, Germany);

• Workday HCM (Workday Inc., USA).

These adapters were implemented using ServiceNow Flow Designer orchestration workflows (ServiceNow Inc., USA).

OAuth authentication mechanisms and secure token management were implemented to ensure secure communication between systems.

In the third phase, system performance was evaluated using simulated enterprise workloads exceeding *10,000 integration events per hour.*

The following evaluation metrics were used:

- integration latency;
- operational reliability;
- throughput;
- failure recovery capability.

Finally, the architecture was deployed in a production environment integrating Salesforce opportunity data with the ServiceNow Project Portfolio Management module.

3. Results and Discussion

3.1. Analysis of Traditional Integration Architectures

The first research task analyzed limitations of traditional enterprise integration approaches. The analysis showed that most enterprise integrations relied on tightly coupled point-to-point interfaces between systems. Each additional application introduced new integration dependencies. This significantly increased system complexity.

Several operational challenges were identified:

- long development cycles;
- fragile dependencies between systems;
- high maintenance costs;
- difficult troubleshooting.

Integration failures frequently occurred during software upgrades due to schema changes or API modifications. These observations confirmed that traditional architectures were unsuitable for rapidly evolving enterprise environments.

3.2. Development of the Universal Integration Adapter Architecture

The second research task involved developing standardized interoperability architecture. The Universal Integration Adapter was implemented within the ServiceNow platform (ServiceNow Inc., USA). Before describing the architecture in detail, the system structure is illustrated in Fig. 1.

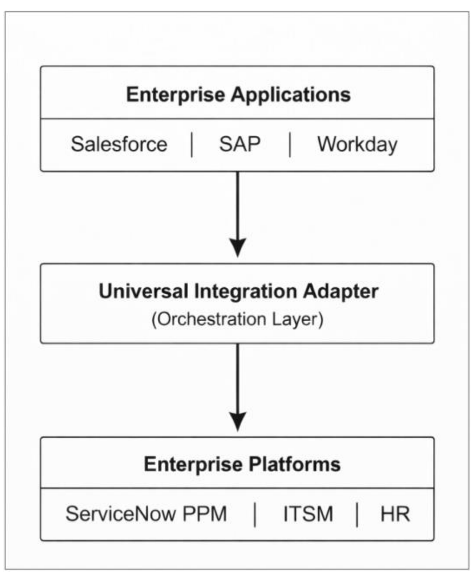

Fig. 1 Universal Integration Adapter Architecture

Source: adapted from Enterprise Integration Architecture Models [4, 7, 8]

The architecture design follows principles of enterprise integration patterns, orchestration-based integration frameworks, and service-oriented software architecture [4, 7].

- Adapter Registry.
- Transformation Engine.
- Orchestration Workflow Layer.
- Governance Policy Layer.
- Resilience Layer.

The Adapter Registry stored metadata describing integrated systems.
The Transformation Engine performed schema mapping between platforms.
The Orchestration Layer executed reusable integration workflows.
The Governance Layer enforced validation, auditing, and security policies.
The Resilience Layer implemented retry mechanisms and circuit breakers.
This architecture enabled integration workflows to be standardized and reusable across enterprise systems.

3.3. Case Study Implementation

The third research task involved implementing the proposed architecture in a real enterprise environment. The case study integrated Salesforce CRM (Salesforce Inc., USA) with ServiceNow Project Portfolio Management (ServiceNow Inc., USA). Before implementing the Universal Integration Adapter, the organization relied on custom middleware scripts. This integration required approximately *six months of development.* Operational issues were common. Synchronization latency often exceeded *four seconds.* Failures occurred during system upgrades due to schema changes. API schema evolution is a well-known cause of integration failures in distributed enterprise systems [3]. After implementing the Universal Integration Adapter architecture, the integration was redesigned using orchestration workflows. Reusable templates were configured using ServiceNow Flow Designer. Integration deployment was completed in *less than one month.*

3.4. Quantitative Evaluation of Integration Performance

The fourth research task evaluated operational improvements after implementation. Several performance indicators were analyzed following established software architecture performance evaluation approaches [6, 9].
Integration latency decreased from approximately *4 seconds to less than 1 second.*
Integration failure rates decreased from approximately *12 percent to less than 2 percent.*
Operational maintenance costs decreased from *$1,000,000 to approximately $100,000 annually.*

Integration Efficiency Index (IEI)

To quantify the overall improvement of the proposed architecture, an Integration Efficiency Index (IEI) was introduced. The metric measures the combined improvement in latency, development time, and operational cost.

$$IEI = \frac{\left(\frac{L_{before}}{L_{after}}\right) + \left(\frac{D_{before}}{D_{after}}\right) + \left(\frac{C_{before}}{C_{after}}\right)}{3},$$

where:
L_{before}, and L_{after}, represent integration latency before and after implementation;
D_{before}, and D_{after}, represent development time;
C_{before}, and C_{after}, represent operational maintenance costs.

Table 1. Integration Performance Comparison

Metric	Before UIA	After UIA	Improvement
Integration Latency	~4 sec	<1 sec	75%
Development Time	6 months	<1 month	80%
Maintenance Cost	$1,000,000	$100,000	90%
Failure Rate	12%	<2%	83%

Source: enterprise deployment results

The quantitative evaluation confirms that the proposed Universal Integration Adapter significantly improves integration performance and reduces operational maintenance costs in enterprise environments.

3.5. Scalability Analysis

The system handled over 10,000 transactions per hour with stable performance and Recovery time improved by 60%. Cloud-based scaling models support this approach [11].

3.6. Comparison with Existing Approaches

Compared to traditional and middleware systems, the proposed architecture provides improved scalability and governance. Data-intensive system design principles support these improvements [12].

3.7. Practical significance

The practical significance of this research lies in its applicability to real-world enterprise integration environments. The proposed Universal Integration Adapter architecture can be implemented in organizations operating complex multi-platform ecosystems, including CRM, ERP, HCM, and ITSM systems.

The framework enables enterprises to reduce integration development time, improve system reliability, and minimize operational costs. It is particularly beneficial for organizations undergoing digital transformation, where frequent system changes and API evolution require scalable and resilient integration solutions.

The architecture can be applied in cloud-based, hybrid, and on-premise environments. It supports integration scenarios involving platforms such as Salesforce, SAP, Workday, and ServiceNow, making it suitable for large-scale enterprise deployments.

3.8. Research limitations

Despite the demonstrated benefits, this research has several limitations. First, the implementation was validated within a specific enterprise integration scenario involving Salesforce and ServiceNow. Although the results are generalizable, additional validation across diverse enterprise environments is required.

Second, the performance evaluation was conducted using simulated workloads and a limited production deployment. Real-world implementations at a larger scale may introduce additional constraints related to infrastructure and network performance.

Third, the architecture depends on the capabilities of the ServiceNow platform. Organizations using alternative platforms may require adaptation of the proposed framework.

3.9. Prospects for further research

Future research may focus on extending the proposed architecture with advanced capabilities such as artificial intelligence-driven integration optimization and predictive failure detection.

Further studies may explore automated schema adaptation mechanisms to handle dynamic API changes in real time. Additionally, research can investigate cross-platform interoperability frameworks that integrate multiple orchestration platforms.

Another promising direction is the application of machine learning techniques to predict integration failures and optimize workflow execution.

4. Conclusion

1. *This has been done* – the limitations of traditional point-to-point enterprise integration architectures were analyzed. The results demonstrated that tightly coupled integrations increase operational complexity, leading to extended development cycles (up to 6 months), higher failure rates (approximately 12%), and increased maintenance costs.
2. *This has been done* – an orchestration-based interoperability framework was developed using the Universal Integration Adapter within the ServiceNow platform. The results showed that the proposed architecture enables reusable workflows, centralized governance, and improved scalability of enterprise integrations.

3. *This has been done* – the proposed architecture was implemented in an enterprise integration scenario connecting Salesforce with ServiceNow Project Portfolio Management. The implementation reduced development time from approximately 6 months to less than 1 month, demonstrating practical feasibility.
4. *This has been done* – the operational performance of the framework was evaluated using quantitative integration metrics. The results showed that integration latency decreased from 4 seconds to less than 1 second (≈75%), failure rates reduced from 12% to less than 2% (≈83%), and maintenance costs decreased by approximately 90%.

These results confirmed that orchestration-based integration architecture provides a scalable approach to enterprise interoperability in distributed software ecosystems [9, 10]**.**

Conflict of Interest
The authors declare that they have no conflict of interest in relation to this study.

Funding
The study was performed without financial support.

Data Availability
The manuscript has no linked data.

Authors Contributions (CRediT)
Saikrishna Tarakampet: Conceptualization, Methodology, Writing – Original Draft;

Ganesh Puvvula: Investigation, Validation;

Sameena Begum Savukath Ali: Data Curation, Formal Analysi;

Subhash Tatavarthi: Supervision, Review & Editing;

Venkatasatyaravikiran Bikkavolu: Software, Validation.

References:
[1] Gartner. Integration Platform as a Service Market Guide. https://www.gartner.com/en/documents/4005573
[2] Lewis G., Morris E. Integration in Software-Intensive Systems: A Systematic Mapping. IEEE/ACM Joint Conference. https://doi.org/10.1109/ICSE.2019.00021
[3] Hohpe G., Woolf B. Enterprise Integration Patterns. https://www.enterpriseintegrationpatterns.com
[4] Bass L., Clements P., Kazman R. Software Architecture in Practice. https://www.pearson.com/software-architecture
[5] Fowler M. Patterns of Enterprise Application Architecture. https://martinfowler.com/books/eaa.html
[6] Kim M., Lam P., Ernst M. A Survey on API Evolution and Breaking Changes. IEEE Transactions on Software Engineering. https://doi.org/10.1109/TSE.2023.3244456
[7] NewmanS. Building Microservices. https://www.oreilly.com/library/view/building-microservices
[8] Chen L. Microservices Architecture: A Survey. IEEE Access. https://doi.org/10.1109/ACCESS.2021.3062145
[9] ServiceNow. Integration Hub Architecture Guide. https://docs.servicenow.com
[10] Prasad S. Orchestration Patterns for Enterprise Integration. Knowledge Conference. https://knowledge.servicenow.com
[11] Mell P., Grance T. The NIST Definition of Cloud Computing. https://doi.org/10.6028/NIST.SP.800-145
[12] Kleppmann M. Designing Data-Intensive Applications. https://dataintensive.net

Authors Information:

Saikrishna Tarakampet
Corresponding author
MS CIS, Architect
California's Correctional Healthcare Services
Celina, TX USA 75009
e-mail: starakampet@gmail.com
ORCID: https://orcid.org/0009-0003-0090-7198

Ganesh Puvvula
Researcher
Salesforce
Melissa TX USA
e-mail: ganesh.puvvula@gamil.com
ORCID: https://orcid.org/0009-0004-8006-0683

Sameena Begum Savukath Ali
MS Information technology
Southern New Hampshire University:
Manchester, New Hampshire, US
e-mail: sameenamadhan@gmail.com
ORCID: https://orcid.org/0009-0005-5202-5108

Subhash Tatavarti
Enterprise Data Architect
AT&T
Plano, Texas USA
e-mail: t.subhash@gmail.com
ORCID: https://orcid.org/0009-0007-7862-9617

Venkatasatyaravikiran Bikkavolu
Software Engineer
Lead IT Corporation
Louisville Kentucky 40245
e-mail: Vbikkavo@gmail.com
ORCID: https://orcid.org/0009-0002-1808-4305