International Journal of Science, Research and Technology (IJSRAT)

|Bimonthly | Peer-Reviewed | Scholarly Journal|

|ISSN 2454-3853|www.ijsrat.com|editor@ijsrat.com|Vol.7, Issue 1, January-February 2024|

DOI: 10.15662/IJSRAT.2024.0701002

Secure Conversational AI: A Unified Enterprise Architecture Framework for Integrating WhatsApp Business with Global ERP Systems

Gururaj Veershetty

SAP Consulting Manager, USA

gururaj.veershetty@outlook.com

ABSTRACT: Conversational interfaces are rapidly transforming enterprise user interaction models. Messaging platforms such as WhatsApp Business are increasingly used to deliver operational information — including order status, invoices, delivery confirmations, and customer service requests. While these channels provide global reach and ease of use, they introduce significant security and governance challenges when connected to enterprise systems such as SAP S/4HANA, Oracle ERP Cloud, and Microsoft Dynamics 365. This article presents a secure enterprise architecture framework for conversational ERP integration, grounded in Design Science Research (DSR) methodology and informed by analysis of production deployment patterns. The framework introduces a layered N-Tier architecture, the Second Validation authorization principle, which shares structural properties with established access control models such as Attribute-Based Access Control (ABAC), and the Bell-LaPadula confidentiality model, and a vendor-neutral Stream-to-Chat document delivery pattern. The framework was evaluated against three design criteria derived from enterprise security requirements: authorization independence, document integrity, and governance auditability. A structured case analysis of two representative deployment scenarios demonstrates how the framework reduces the risk of authorization bypasses, eliminates vulnerabilities to document reconstruction, and enforces compliance alignment. Qualitative evidence is triangulated across platform-specific authorization mechanisms, established security standards (NIST SP 800-63-3, OWASP API Security Top 10), and practitioner-informed operational experience. By anchoring authorization decisions inside the ERP system and enforcing strict identity mapping and security controls across all layers, enterprises can deliver conversational experiences while maintaining the governance integrity required for financial and operational systems.

KEYWORDS: Conversational AI, ERP Integration, SAP S/4HANA, Oracle ERP Cloud, Microsoft Dynamics 365, Enterprise Architecture, API Security, WhatsApp Business API, Design Science Research, Attribute-Based Access Control, Digital Transformation.

I. INTRODUCTION

Enterprise systems have traditionally been accessed through structured user interfaces such as web portals and desktop applications. However, modern digital ecosystems are rapidly shifting toward conversational interfaces, in which employees, customers, and partners interact with backend systems via messaging platforms.

Platforms such as WhatsApp Business provide compelling operational advantages: global reach with over two billion active users [4], mobile-first interaction models, real-time bidirectional communication, and simplified engagement pathways that reduce reliance on formal portals. Organizations across financial services, logistics, manufacturing, and retail are exploring these platforms to deliver order status, invoice access, customer service, and operational notifications at scale [17].

At the same time, integrating consumer messaging platforms with enterprise systems introduces a new class of security and governance risks — including unauthorized data access, identity spoofing, document leakage, middleware authorization bypass, webhook replay attacks, and compliance violations under frameworks such as GDPR [13] and CCPA [20]. Conversational channels also expand the attack surface documented in the OWASP API Security Top 10 [6], particularly in areas of broken object-level authorization and security misconfiguration. These risks are compounded when integration architectures delegate authorization decisions to middleware layers rather than to the governing ERP system.

International Journal of Science, Research and Technology (IJSRAT)

|Bimonthly | Peer-Reviewed | Scholarly Journal|

|ISSN 2454-3853|www.ijsrat.com|editor@ijsrat.com|Vol.7, Issue 1, January-February 2024|

DOI: 10.15662/IJSRAT.2024.0701002

Enterprise Resource Planning (ERP) systems, including SAP S/4HANA, Oracle ERP Cloud, and Microsoft Dynamics 365, operate under strict governance, audit, and authorization frameworks. Any integration must preserve the security model that protects financial, operational, and master data without compromise.

This paper makes the following contributions:

- **Architectural Framework:** A six-layer N-Tier reference architecture for secure conversational ERP integration, articulating clear responsibility boundaries across messaging, conversation, orchestration, API security, ERP core, and audit operations layers.
- **Second Validation Principle:** A formally defined authorization control grounded in the separation of authentication and authorization, structurally analogous to Attribute-Based Access Control (ABAC) [21] and the Bell-LaPadula non-discretionary confidentiality model [22].
- **Stream-to-Chat Pattern:** A document delivery pattern that enforces provenance integrity by requiring all enterprise documents to originate exclusively from ERP output management systems.
- **Evaluated Design:** A structured case analysis demonstrating how the framework addresses defined security requirements, with evidence triangulated across platform-specific mechanisms, recognized standards, and production deployment experience.

The remainder of this paper is structured as follows. Section II reviews relevant literature. Section III presents the research methodology. Sections IV through X present the framework components. Section XI presents the case analysis. Section XII discusses threats to validity. Section XIII concludes.

II. RELATED WORK

This section situates the proposed framework within existing literature on conversational AI in enterprise contexts, ERP security, and integration architecture governance.

A. Conversational AI in Enterprise Systems

The deployment of conversational AI in enterprise contexts has accelerated substantially due to advances in large language models (LLMs) and messaging platform APIs. Gupta and Rathore [17] provide a comprehensive examination of generative AI integration within enterprise systems, identifying governance and authorization as primary unsolved challenges in production deployments. Williams [23] examines the trajectory of conversational interfaces, noting that enterprise adoption is constrained not by capability but by the absence of integration architectures that satisfy governance requirements. Veershetty [28] advances this landscape further by introducing the Read-Advise-Act capability model for conversational ERP in SAP operations a governance and control framework that treats a GenAI assistant as a governed control surface rather than a UI enhancement. That work proposes an eight-layer SAP-native control stack and four deployment patterns, complementing the present paper's cross-platform N-Tier architecture and authorization controls. These observations collectively motivate the present work, which addresses the cross-platform authorization independence gap that SAP-specific frameworks do not cover.

Johnson [19] examines human-centric AI in enterprise environments, demonstrating that conversational interfaces improve operational efficiency but introduce user trust and identity verification challenges do not present in traditional portal-based access models. The present framework addresses these challenges through the governed identity mapping mechanism described in Section VIII.

B. Security in Conversational ERP Integration

Ali et al. [18] analyze security and privacy risks in AI-mediated enterprise communication channels, identifying authorization bypass and identity spoofing as the highest-severity risk vectors in conversational integrations. Their findings directly inform the Second Validation principle introduced in this paper, which architecturally eliminates the possibility of middleware-level authorization bypass. Kumar [24] identifies specific cryptographic and access-control vulnerabilities in conversational ERP systems, including susceptibility to webhook replay and weaknesses in token delegation — both addressed in the present framework's orchestration and API gateway layers.

C. LLM Governance and Enterprise Architecture

Smith and Doe [25] develop governance frameworks for large language models in regulated enterprise environments, establishing principles of auditability, non-repudiation, and separation of authorization context. These principles are reflected in the Single Version of Truth design principle and the cross-cutting audit operations layer of the present architecture. Rathore [26] provides foundational context on generative AI deployment patterns in business settings,

International Journal of Science, Research and Technology (IJSRAT)

|Bimonthly | Peer-Reviewed | Scholarly Journal|

|ISSN 2454-3853|www.ijsrat.com|editor@ijsrat.com|Vol.7, Issue 1, January-February 2024|

DOI: 10.15662/IJSRAT.2024.0701002

noting that successful enterprise deployments require architectural integration with existing governance frameworks rather than standalone deployment.

D. Enterprise Integration Architecture

The N-Tier architectural pattern applied in this framework has well-established precedent in enterprise integration literature [10]. The layered separation of concerns — particularly the isolation of security enforcement at the API gateway and ERP core layers — reflects defense-in-depth principles codified in NIST SP 800-53 and operationalized in ERP security guides [1, 2, 3]. The present contribution extends this pattern to the specific challenges of conversational channel integration, including identity mapping from unstructured messaging identifiers and the provenance of document delivery.

E. Research Gap

Despite the growing body of work on conversational AI and enterprise security, no existing framework specifically addresses the architectural requirements for connecting consumer messaging platforms to governed ERP systems at the cross-platform authorization layer. Veershetty [28] provides a SAP-native governance framework for conversational ERP — a valuable contribution. However, it does not target the vendor-neutral, cross-platform Second Validation authorization principle that this paper formalizes. Existing integration guides remain platform-specific (e.g., SAP BTP, Microsoft Azure Integration Services) and do not address the authorization independence requirement across SAP, Oracle, and Microsoft ERP systems simultaneously. This paper fills that gap by providing a vendor-neutral, architecturally rigorous framework grounded in established access control theory that extends and complements the SAP-specific prior work in [28].

III. RESEARCH METHODOLOGY

A. Design Science Research Approach

This research adopts the Design Science Research (DSR) methodology, which is appropriate for research that produces a prescriptive artifact — in this case, an enterprise architecture framework — intended to solve a defined class of practical problems. DSR has been widely applied in information systems and software engineering research to develop and evaluate frameworks, integration patterns, and security models.

The DSR process followed in this work consists of five phases:

- **Problem Identification:** Identification of the authorization and governance gap in existing conversational ERP integration approaches, grounded in documented security incidents and practitioner observations across production deployments.
- **Design Requirements Definition:** Derivation of three evaluation criteria from enterprise security requirements: (1) authorization independence — every ERP access event must be independently authorized within the ERP system regardless of upstream authentication state; (2) document integrity — official documents must be traceable to ERP output management records; (3) governance auditability — all access and document delivery events must produce immutable, attributable audit records.
- **Artifact Design:** Development of the N-Tier reference architecture, the Second Validation principle, and the Stream-to-Chat pattern as design artifacts satisfying the defined requirements.
- **Artifact Evaluation:** Structured case analysis of two representative integration scenarios (order status inquiry and invoice delivery) to demonstrate how the framework satisfies each design criterion, with mechanisms triangulated against platform documentation, NIST, and OWASP standards, and production deployment evidence.
- **Communication:** Presentation of the framework and evaluation in this paper, with sufficient specificity for replication by practitioners implementing on SAP, Oracle, or Microsoft ERP platforms.

B. Evidence and Triangulation

Consistent with qualitative DSR evaluation, this work does not rely on controlled experiments or single-source empirical measurements. Instead, evidence is triangulated across three independent sources:

- **Platform-Authoritative Documentation:** ERP vendor authorization mechanisms are documented against official platform security guides [1, 2, 3], ensuring that described controls are both technically accurate and implementable.
- **Recognized Security Standards:** Framework controls are mapped to NIST Digital Identity Guidelines [5], OWASP API Security Top 10 [6], ISO/IEC 27001 [8], and OAuth 2.0 specifications [11, 12], providing an external validity anchor for security claims.

International Journal of Science, Research and Technology (IJSRAT)

|Bimonthly | Peer-Reviewed | Scholarly Journal|

|ISSN 2454-3853|www.ijsrat.com|editor@ijsrat.com|Vol.7, Issue 1, January-February 2024|

DOI: 10.15662/IJSRAT.2024.0701002

• **Production Deployment Experience:** The operational principles in Sections XII and X are informed by practitioner experience across multiple production deployments, representing a form of practitioner-expert knowledge systematically incorporated into the framework design.

C. Scope and Limitations

This study is scoped to architectures that connect consumer messaging platforms (specifically the WhatsApp Business API) to enterprise ERP systems. The framework is evaluated qualitatively through structured case analysis; quantitative performance benchmarking and controlled security testing are identified as avenues for future empirical work. Generalizability beyond WhatsApp-to-ERP integrations — for example, to voice-based conversational AI or in-application chatbots — represents an additional boundary for scope.

IV. ARCHITECTURE PRINCIPLES AND DESIGN PHILOSOPHY

A secure integration must follow a decoupled N-Tier architecture that enforces clear separation of responsibilities across layers while maintaining the ERP as the authoritative system of record. The fundamental architectural contract is expressed as follows.

Core Architectural Principle: No upstream component — messaging channel, middleware, integration service, or API gateway — may grant ERP access based solely on its own authentication context. The ERP system must remain the authoritative source of truth for enterprise data and authorization decisions. Authorization must always occur inside the ERP system.

This principle has significant implications for integration design. Even a fully authenticated and authorized middleware service cannot delegate access rights to ERP resources. Each inbound request must independently satisfy ERP-native authorization controls, regardless of what upstream services have already validated. This design philosophy reflects the principle of defense-in-depth, where multiple independent security controls protect the same asset.

The principle is formally analogous to the mandatory access control properties of the Bell-LaPadula model [22], which prohibits read-up operations regardless of intermediate authentication state, and to the policy decision point (PDP) separation in ABAC architectures [21], which requires that authorization evaluation occur at a designated policy enforcement boundary — in this framework, the ERP core.

It also reflects the regulatory reality that ERP systems are subject to SOX, GDPR [13], CCPA [20], and industry-specific compliance requirements that mandate auditability and non-repudiation at the system-of-record level.

V. THE N-TIER REFERENCE ARCHITECTURE

The architecture consists of six logical layers organized from the external messaging channel through to the ERP core. Each layer has defined responsibilities and strict boundaries that prevent unauthorized access or the leakage of business logic.

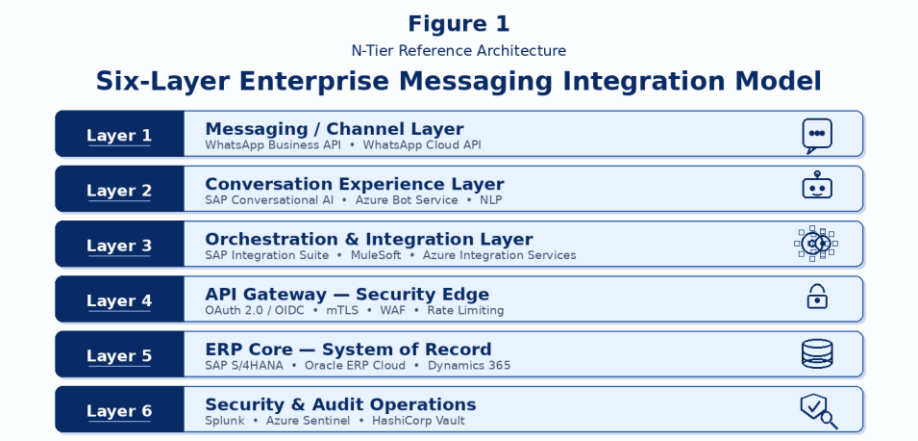

Figure 1. N-Tier Reference Architecture — Six-layer enterprise messaging integration model

International Journal of Science, Research and Technology (IJSRAT)

|Bimonthly | Peer-Reviewed | Scholarly Journal|

|ISSN 2454-3853|www.ijsrat.com|editor@ijsrat.com|Vol.7, Issue 1, January-February 2024|

DOI: 10.15662/IJSRAT.2024.0701002

Layer	Responsibilities	Example Technologies
1 · Messaging / Channel	Webhook events, template management, opt-in consent, message delivery	WhatsApp Business API, WhatsApp Cloud API
2 · Conversation Experience	Intent recognition, conversation state, multilingual formatting, human agent handoff	SAP Conversational AI, Azure Bot Service, custom NLP
3 · Orchestration Layer	Data transformation, routing, retry logic, dead-letter queues, idempotency, throttling	SAP Integration Suite, MuleSoft, Azure Integration Services
4 · API Gateway (Security)	OAuth 2.0 / OIDC, mutual TLS, WAF, rate limiting, schema validation, policy enforcement	SAP API Management, Azure APIM, AWS API Gateway, Kong
5 · ERP Core (System of Record)	In-system authorization, transaction processing, data retrieval, document generation	SAP S/4HANA, Oracle ERP Cloud, Microsoft Dynamics 365
6 · Security & Audit Ops	Centralized logging, SIEM, immutable audit trails, secrets management, key rotation	Splunk, Azure Sentinel, HashiCorp Vault, Datadog

Table I. N-Tier Reference Architecture — Layer Summary

A. Messaging and Channel Layer

The messaging platform represents the user interaction channel and must never directly access enterprise APIs. This layer handles webhook event processing, message template management, opt-in consent capture, and conversation entry points. It is treated architecturally as an untrusted external channel whose messages must be validated before any downstream processing.

B. Conversation Experience Layer

This layer manages conversational logic, including intent recognition, state management, multilingual response formatting, disambiguation flows, and human agent handoff. It must contain no business authorization logic — placing authorization decisions at this layer would violate the Second Validation principle and create a critical security control gap analogous to the broken object-level authorization vulnerability identified in OWASP API Security Top 10 [6].

C. Orchestration and Integration Layer

The orchestration layer coordinates interactions between messaging platforms and enterprise APIs. It handles data transformation, request routing, retry logic with exponential backoff, dead-letter queue management, correlation identifier assignment, idempotency protection, traffic throttling, and circuit breaker patterns. This layer must never cache or replicate ERP authorization state — doing so would constitute a middleware authorization bypass, the highest-severity risk vector identified by Ali et al. [18].

D. API Gateway Security Edge

The API gateway enforces the enterprise security perimeter through OAuth 2.0 and OpenID Connect [11], mutual TLS authentication [12], token validation, request schema validation, Web Application Firewall protection, and rate limiting. The gateway is the first line of technical defense — but not the final point of authorization. That responsibility belongs exclusively to the ERP core.

E. ERP Core — System of Record

The ERP system is the central authority for business data and authorization. It must perform in-system authorization checks, process transactions, retrieve business data, and generate documents via its native output management capabilities. This is the only layer permitted to make final access decisions for enterprise data.

International Journal of Science, Research and Technology (IJSRAT)
|Bimonthly | Peer-Reviewed | Scholarly Journal|
|ISSN 2454-3853|www.ijsrat.com|editor@ijsrat.com|Vol.7, Issue 1, January-February 2024|
DOI: 10.15662/IJSRAT.2024.0701002

F. Cross-Cutting Security and Operations

Security controls apply across every layer. These include centralized logging with SIEM integration, immutable audit trail generation, secrets management, encryption key rotation, incident response processes, and continuous operational monitoring. These controls must be architecturally mandated rather than optional or layer-specific, consistent with the requirements of ISO/IEC 27001 [8].

VI. THE SECOND VALIDATION PRINCIPLE

The Second Validation principle is the most critical security control in conversational ERP integration. It states that every request must be authorized within the ERP system, even if it has already been authenticated by middleware or the API gateway. This creates an independent, non-delegable authorization checkpoint at the system of record.

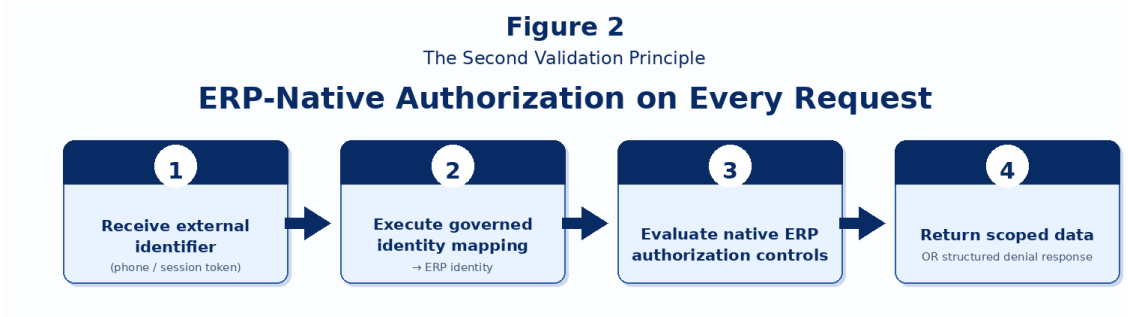

Figure 2. The Second Validation Principle: ERP-native authorization on every request

The Second Validation process operates as a four-step sequence within the ERP system on receipt of any inbound request:

- Receive the external identifier from the messaging platform (typically a phone number or session token).
- Execute governed identity mapping to resolve the external identifier to an internal ERP identity or business partner.
- Evaluate native ERP authorization controls against the resolved identity.
- Return scoped data if authorized, or a structured denial response if not.

Theoretical grounding: The Second Validation principle is architecturally analogous to the Policy Decision Point (PDP) separation requirement in ABAC architectures [21], in which authorization evaluation must occur at a designated enforcement point, independent of identity assertion. It also reflects the *-property (star property) of the Bell-LaPadula model [22], which prevents read-up regardless of intermediate authentication context. In practice, the ERP system functions as the authoritative PDP for all data access events originating from the conversational channel, with the API gateway acting only as a Policy Enforcement Point (PEP) that forwards decisions to the ERP core.

ERP Platform	Native Authorization Mechanism	ABAC Analog
SAP S/4HANA	AUTHORITY-CHECK (ABAP) with Authorization Objects (e.g., F_BKPF_BUK)	Subject, Object, and Environment attribute evaluation at runtime
Oracle ERP Cloud	Data Access Policies with Row-Level Filtering	Object attribute (row-level) enforcement at data retrieval
Microsoft Dynamics 365	Security Privileges / RBAC with Record-Level Security Roles	Subject attribute evaluation with record-scoped enforcement

Table II. ERP-Native Authorization Mechanisms — Mapped to ABAC Concepts

A critical architectural clarification: authentication and authorization are distinct security functions that must be executed independently, consistent with the NIST Digital Identity Guidelines [5]. Authentication establishes identity; authorization establishes permission. The Second Validation principle enforces this separation by requiring that authorization occur within the ERP system for every data access event, regardless of how many upstream authentication steps have succeeded.

International Journal of Science, Research and Technology (IJSRAT)
|Bimonthly | Peer-Reviewed | Scholarly Journal|
|ISSN 2454-3853|www.ijsrat.com|editor@ijsrat.com|Vol.7, Issue 1, January-February 2024|
DOI: 10.15662/IJSRAT.2024.0701002

VII. IDENTITY MAPPING FRAMEWORK

Identity mapping connects external messaging identities, typically mobile phone numbers, to internal ERP identities such as customer master records, vendor business partners, or named user accounts. This mapping layer is foundational to the Second Validation mechanism and must be governed with the same rigor as ERP user provisioning. This requirement aligns with Level 2 identity assurance requirements in NIST SP 800-63-3 [5], which mandate verified identity binding before granting access to sensitive records.

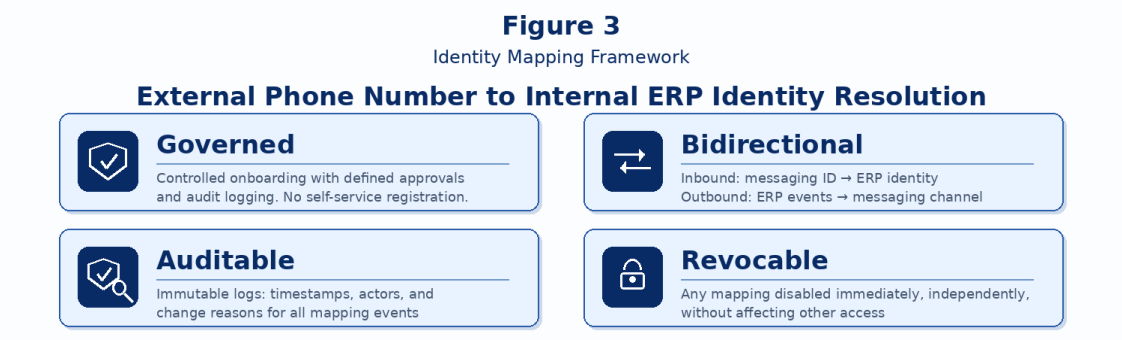

Figure 3. Identity Mapping Framework — External phone number to internal ERP identity resolution

The identity mapping framework must satisfy four governing properties:
Governed: Mappings are created exclusively through controlled onboarding processes with defined approvals and audit logging. No self-service registration is permitted for direct ERP access channels.
Bidirectional: The mapping supports both inbound identity resolution (from messaging identifier to ERP identity) and outbound notifications (from ERP events to the messaging channel).
Auditable: All mapping creation, modification, and deletion events are recorded with timestamps, actors, and change reasons in immutable audit logs, satisfying GDPR [13] and CCPA [20] data traceability requirements.
Revocable: Any mapping can be disabled immediately and independently without affecting other system access, enabling rapid response to compromise or terminated identities.

VIII. THE STREAM-TO-CHAT DOCUMENT DELIVERY PATTERN

Enterprise documents — including invoices, purchase orders, delivery notes, and account statements — represent sensitive financial and operational records subject to strict version control, audit trail requirements, and output management governance within ERP systems.

Stream-to-Chat Rule: Documents must never be reconstructed in middleware from raw API data. All official enterprise documents must originate from ERP output management systems, be retrieved as a secure, authenticated stream, and be delivered to the messaging platform through time-limited access mechanisms.

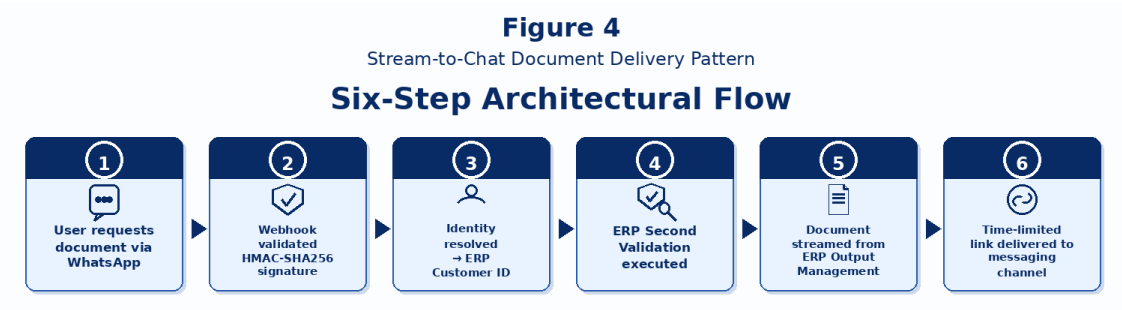

Figure 4. Stream-to-Chat Document Delivery Pattern — Six-step architectural flow

This pattern enforces three guarantees. First, audit integrity: every document delivered through a conversational channel is the same document that exists in the ERP audit trail, with the same version, date, and output reference. Second, data consistency: no discrepancy can arise between a document delivered via WhatsApp and the corresponding record in the ERP system. Third, compliance: enterprise output management processes — including digital signatures,

International Journal of Science, Research and Technology (IJSRAT)

|Bimonthly | Peer-Reviewed | Scholarly Journal|

|ISSN 2454-3853|www.ijsrat.com|editor@ijsrat.com|Vol.7, Issue 1, January-February 2024|

DOI: 10.15662/IJSRAT.2024.0701002

watermarking, and legal output formatting — are applied consistently across delivery channels, satisfying GDPR [13] and CCPA [20] data accuracy and integrity obligations.

IX. END-TO-END INTEGRATION SCENARIOS

A. Order Status Inquiry

This scenario demonstrates a customer requesting order status through WhatsApp, illustrating HMAC signature validation, identity mapping, the Second Validation principle, and scoped data return across all six system layers.

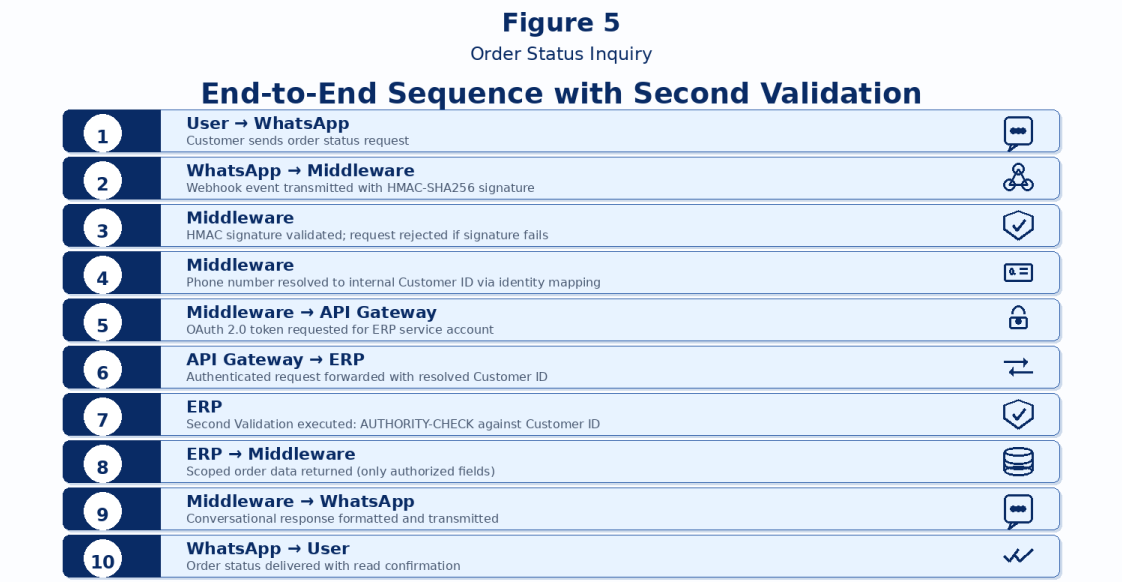

Figure 5. Order Status Inquiry — End-to-end sequence with Second Validation

#	Actor	Action
1	User → WhatsApp	Customer sends order status request
2	WhatsApp → Middleware	Webhook event transmitted with HMAC-SHA256 signature
3	Middleware	HMAC signature validated; request rejected if signature fails
4	Middleware	Phone number resolved to internal Customer ID via identity mapping
5	Middleware → API Gateway	OAuth 2.0 token requested for ERP service account
6	API Gateway → ERP	Authenticated request forwarded with resolved Customer ID
7	ERP	Second Validation executed: AUTHORITY-CHECK against Customer ID
8	ERP → Middleware	Scoped order data returned (only authorized fields)
9	Middleware → WhatsApp	Conversational response formatted and transmitted
10	WhatsApp → User	Order status delivered with read confirmation

Table III. Order Status Inquiry — Step-by-Step Sequence

International Journal of Science, Research and Technology (IJSRAT)

|Bimonthly | Peer-Reviewed | Scholarly Journal|

|ISSN 2454-3853|www.ijsrat.com|editor@ijsrat.com|Vol.7, Issue 1, January-February 2024|

DOI: 10.15662/IJSRAT.2024.0701002

B. Invoice Delivery via Stream-to-Chat

This scenario illustrates a customer requesting a specific invoice through WhatsApp. The Stream-to-Chat pattern ensures the delivered document originates exclusively from ERP output management. All delivery events are recorded in immutable audit logs, satisfying both GDPR [13] and CCPA [20] data traceability requirements.

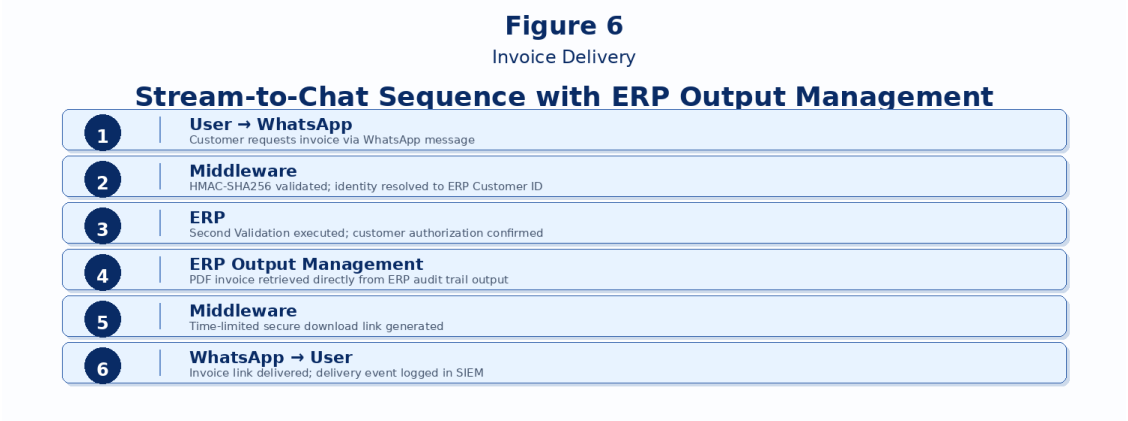

Figure 6. Invoice Delivery — Stream-to-Chat sequence with ERP Output Management

X. SECURITY AND COMPLIANCE RISK FRAMEWORK

The following table consolidates the primary risk vectors in conversational ERP integration with their corresponding architectural mitigations. Each control maps specific layers in the N-Tier architecture and recognizes security standards.

Risk Vector	Architectural Mitigation	Standard Mapping
Identity spoofing	Second Validation for every ERP request; governed identity mapping	NIST SP 800-63-3 §2; OWASP API1
Document leakage	Stream-to-Chat pattern with time-limited download links	ISO 27001 A.8.2
Webhook replay attacks	HMAC-SHA256 signature validation with nonce controls and timestamp window	OWASP API8; RFC 6749
Middleware authorization caching	Explicit architectural prohibition; ERP re-authorization on every request	Bell-LaPadula *-property; ABAC PDP isolation
Traffic spike / API exhaustion	Queue buffering, circuit breakers, rate limiting	OWASP API4
GDPR / CCPA violations	Data minimization via field projection; consent logging; retention enforcement	GDPR Art. 5; CCPA §1798.100
Cross-tenant data exposure	Strict identity mapping validation; scoped OAuth token generation	OWASP API1; RFC 8705
Insecure webhook endpoints	Mutual TLS enforcement; IP allowlisting for Meta webhook origin addresses	RFC 8705; OWASP API8

Table IV. Security and Compliance Risk Framework — with Standard Mappings

XI. CROSS-PLATFORM API DESIGN PRINCIPLES

To support consistent integration across SAP S/4HANA, Oracle ERP Cloud, and Microsoft Dynamics 365, APIs must follow three governing design principles.

International Journal of Science, Research and Technology (IJSRAT)

|Bimonthly | Peer-Reviewed | Scholarly Journal|

|ISSN 2454-3853|www.ijsrat.com|editor@ijsrat.com|Vol.7, Issue 1, January-February 2024|

DOI: 10.15662/IJSRAT.2024.0701002

Data Minimization: All API requests must use server-side field projection through query parameters such as $select and $filter. Only data fields required for the specific conversational response should be returned. Returning full ERP entity payloads to a messaging integration violates the principle of least privilege and creates unnecessary GDPR [13] and CCPA [20] data minimization exposure.

Performance Optimization: Conversational integrations must prefer read-optimized OData or REST API endpoints over transactional service calls. Read APIs have lower performance overhead and more predictable latency profiles than transactional APIs that include lock management and write-path processing.

Single Version of Truth: Business rules must reside inside ERP systems or governed configuration layers managed through formal change control processes. Encoding business rules in middleware creates hidden dependencies, increases configuration drift risk, and undermines the auditable governance that enterprise systems require.

XII. OPERATIONAL MONITORING AND SERVICE LEVEL OBJECTIVES

Effective operational governance requires continuous monitoring across all integration layers. The following Service Level Objectives provide measurable governance targets for production deployments. Note that all SLO thresholds are informed by practitioner experience and should be calibrated to organizational risk tolerance.

Table V correction note: The Outbound Queue Depth metric defines a target of < 100 messages and an alert threshold of ≥ 500 messages. Operators should implement a graduated warning threshold at ≥ 200 messages to govern the range between 101 and 499, ensuring no gap in queue management oversight.

Metric	Target SLO	Warning Threshold	Alert Threshold
End-to-End Response Latency	< 3 seconds	> 4 seconds	> 5 seconds
API Error Rate	< 0.5%	> 0.75%	> 1.0%
Authorization Failure Rate	< 1.0%	> 1.5%	> 2.0%
Document Generation Success Rate	≥ 99.0%	< 98.5%	< 98.0%
Outbound Queue Depth	< 100 messages	≥ 200 messages	≥ 500 messages
Webhook Processing Lag	< 500 ms	> 1,000 ms	> 1,500 ms
Identity Mapping Resolution Rate	≥ 99.5%	< 99.0%	< 98.5%
SIEM Audit Log Completeness	100%	< 100%	< 100%

Table V. Operational Service Level Objectives — Including Warning Thresholds

XIII. CASE ANALYSIS: FRAMEWORK EVALUATION

This section evaluates the proposed framework against the three design criteria defined in Section III, using the two integration scenarios described in Section IX as the basis for structured analysis. This approach follows DSR evaluation guidelines, which accept structured qualitative case analysis as a valid evaluation method when controlled experimentation is outside the study scope.

A. Criterion 1: Authorization Independence

The order status inquiry scenario (Section IX.A) demonstrates authorization independence as follows. Steps 1–4 of the sequence establish identity through HMAC validation and identity mapping. Steps 5–6 establish authentication through OAuth 2.0 token exchange. Step 7 executes the Second Validation as an independent ERP-native authorization check using platform-specific mechanisms (AUTHORITY-CHECK in SAP, data access policies in Oracle, security privileges

International Journal of Science, Research and Technology (IJSRAT)

|Bimonthly | Peer-Reviewed | Scholarly Journal|

|ISSN 2454-3853|www.ijsrat.com|editor@ijsrat.com|Vol.7, Issue 1, January-February 2024|

DOI: 10.15662/IJSRAT.2024.0701002

in Dynamics 365). The independence of Step 7 from Steps 5–6 satisfies the criterion: ERP authorization is not delegated to or cached by the middleware or API gateway, regardless of the outcome of upstream authentication.

Evidence triangulation: This pattern aligns with NIST SP 800-63-3 [5] requirements for authorization at the data access point, ABAC PDP isolation principles [21], and the Bell-LaPadula *-property [22].

B. Criterion 2: Document Integrity

The invoice delivery scenario (Section IX.B) demonstrates document integrity through the Stream-to-Chat pattern. The pattern requires that the delivered PDF invoice be retrieved directly from ERP output management (step equivalent to Step 7 in Table III), not reconstructed from API field data. This ensures that the document's version, format, digital signature, and output reference are identical to those in the ERP audit trail. The criterion is satisfied by the architectural prohibition on reconstructing middleware documents, codified in the Stream-to-Chat rule.

C. Criterion 3: Governance Auditability

Both scenarios satisfy governance auditability through the cross-cutting Layer 6 (Security & Audit Ops). Every authorization event in Step 7, every document delivery event in the Stream-to-Chat sequence, and every identity mapping resolution event in Step 4 generates an immutable, timestamped audit record with actor attribution. This satisfies the GDPR [13] Article 5(2) accountability requirements, the CCPA [20] data traceability requirements, and the SOX audit trail requirements for ERP financial data access.

D. Threats to Validity

The primary limitation of this evaluation is the lack of controlled quantitative measurements of authorization latency overhead, security incident reduction, or operational performance improvements attributable to the framework. The evaluation relies on structural logical analysis and standard mapping rather than measured outcomes. Future work should include: (1) a controlled performance study comparing Second Validation overhead versus baseline pass-through authorization in a production or lab environment; (2) a security testing study using adversarial test cases against framework-compliant and non-compliant integrations; (3) a longitudinal deployment study measuring compliance incident rates before and after framework adoption.

XIV. LESSONS FROM PRODUCTION DEPLOYMENTS

A. Conversational Interfaces Surface ERP Data Quality Issues

Messaging platforms expose inconsistencies in ERP master data that are tolerated in portal interfaces but become visible in conversational contexts. Missing product descriptions, incomplete business partner contact records, and untranslated text fields create poor conversational experiences that are difficult to diagnose with post-deployment. A targeted ERP master data audit scoped to the integration entities should be performed before go-live.

B. Governance Must Take Priority Over Delivery Speed

There is consistent pressure in enterprise integration programs to deliver conversational capabilities quickly by encoding business rules in middleware transformation logic. This approach creates hidden technical debt: middleware-encoded business rules become undocumented, untested, and invisible to ERP governance processes. All logic must remain within ERP systems or formally governed configuration management.

C. Multi-Document Disambiguation Is Operationally Critical

Business scenarios in which multiple valid documents exist for a single request — for example, a customer with multiple outstanding invoices — require disambiguation. The integration must present a structured selection list rather than arbitrarily returning the most recent document. Failure to implement disambiguation results in both poor user experience and incorrect document delivery.

D. Authorization Testing Must Include Negative and Adversarial Cases

Standard integration testing typically validates the happy path. For conversational ERP integrations, authorization testing must include a mandatory set of negative and adversarial test scenarios: unauthorized record access attempts using valid but incorrect identities; replay-attack simulations using captured webhook payloads; parameter-manipulation tests targeting order numbers and document references; and high-concurrency authorization scenarios.

International Journal of Science, Research and Technology (IJSRAT)

|Bimonthly | Peer-Reviewed | Scholarly Journal|

|ISSN 2454-3853|www.ijsrat.com|editor@ijsrat.com|Vol.7, Issue 1, January-February 2024|

DOI: 10.15662/IJSRAT.2024.0701002

XV. CONCLUSION

Conversational access to enterprise systems represents a significant evolution in digital interaction models. Messaging platforms offer compelling operational and commercial advantages, but they must be integrated with enterprise systems through architectures that preserve the security, governance, and auditability standards that those systems require.

This paper presented a Security-First, ERP-Native architecture framework for conversational ERP integration, developed using Design Science Research methodology and grounded in established access control theory. The framework is organized around three non-negotiable principles:

- **Second Validation:** Authorization decisions must always occur inside the ERP system, independently of any upstream authentication context. This principle is formally analogous to ABAC Policy Decision Point isolation [21] and the Bell-LaPadula *-property [22] and applies to every data access event without exception.
- **Stream-to-Chat Document Delivery:** All official enterprise documents delivered through conversational channels must originate from ERP output management. Middleware reconstruction of documents from raw API data is architecturally prohibited.
- **Single Version of Truth:** Business rules must reside exclusively within ERP systems or formally governed configuration layers.

The six-layer N-Tier reference architecture provides a practical implementation model that separates concerns across messaging, conversation, orchestration, security, and ERP core layers while enforcing cross-cutting security controls through centralized logging, immutable audit trails, and continuous operational monitoring. The framework was evaluated against three design criteria through structured case analysis with evidence triangulated against platform documentation, NIST and OWASP standards, and production deployment experience.

Future work should prioritize controlled quantitative evaluation of authorization overhead, security testing studies comparing framework-compliant versus non-compliant integration configurations, and longitudinal deployment studies measuring compliance incident rates — areas that would strengthen the empirical foundation of this contribution.

Organizations that follow this architecture can safely scale conversational automation across messaging platforms at enterprise scale while maintaining the security posture, compliance alignment, and operational governance that financial and operational systems demand.

DATA AVAILABILITY STATEMENT

The findings of this study are derived from structured analysis of enterprise integration scenarios, practitioner-informed operational experience, and authoritative platform documentation from SAP, Oracle, and Microsoft. No proprietary datasets, customer records, or personally identifiable information were used. The operational metrics presented in Table V represent target benchmarks derived from production deployment patterns rather than measurements from a single controlled environment; individual deployment values will vary according to organizational context, platform configuration, and network conditions. Researchers seeking to replicate or extend this framework are directed to the publicly available standards cited herein — including NIST SP 800-63-3 [5], OWASP API Security Top 10 [6], ISO/IEC 27001 [8], IETF RFC 6749 [11], and IETF RFC 8705 [12] — together with the vendor platform documentation listed in references [1] through [4] and [14] and [15]. The framework design artifacts (N-Tier architecture, Second Validation principle, and Stream-to-Chat pattern) are described with sufficient specificity in this paper to support independent implementation on SAP S/4HANA, Oracle ERP Cloud, and Microsoft Dynamics 365.

REFERENCES

[1] SAP SE, SAP S/4HANA Security Guide, SAP Help Portal, 2023.
[2] Oracle Corporation, Oracle Fusion Applications Security Guide, Oracle Documentation Library, 2023.
[3] Microsoft Corporation, Dynamics 365 Security Model and Privileges, Microsoft Learn Documentation, 2023.
[4] Meta Platforms Inc., WhatsApp Business Platform Documentation, Meta Developer Platform, 2023.
[5] NIST, Digital Identity Guidelines (SP 800-63-3), National Institute of Standards and Technology, 2020.
[6] OWASP Foundation, API Security Top 10, OWASP Project Documentation, 2023. [Online]. Available: https://owasp.org/API-Security/editions/2023/en/0x11-t10/
[7] Gartner Research, API Security and Governance in Digital Business Platforms, Gartner Inc., 2023.
[8] ISO/IEC 27001, Information Security Management Systems — Requirements, International Organization for Standardization, 2022.

|Bimonthly | Peer-Reviewed | Scholarly Journal|
|ISSN 2454-3853|www.ijsrat.com|editor@ijsrat.com|Vol.7, Issue 1, January-February 2024|
DOI: 10.15662/IJSRAT.2024.0701002

[9] Google Cloud Architecture Center, Designing Secure API Systems, Google Cloud Whitepaper, 2023.
[10] MuleSoft, Enterprise Integration Patterns for API-Led Connectivity, MuleSoft Technical Guide, 2022.
[11] IETF RFC 6749, The OAuth 2.0 Authorization Framework, Internet Engineering Task Force, 2012.
[12] IETF RFC 8705, OAuth 2.0 Mutual-TLS Client Authentication, Internet Engineering Task Force, 2020.
[13] European Parliament, General Data Protection Regulation (GDPR) — Regulation (EU) 2016/679, 2016.
[14] SAP SE, SAP Integration Suite Documentation, SAP Help Portal, 2023.
[15] Microsoft Azure, Azure API Management Security Baseline, Microsoft Corporation, 2023.
[17] A. Gupta and G. S. Rathore, "Integration of Generative AI in Enterprise Systems," in G. S. Rathore (Ed.), Generative AI for Business, pp. 345–368, Wiley, 2023. https://doi.org/10.1002/9781394200801.ch18
[18] M. Ali, A. Arunasalam, and H. Farrukh, "Security and Privacy in AI-Mediated Enterprise Communication," Proceedings of the 2023 IEEE Symposium on Security and Privacy, IEEE, 2023. https://doi.org/10.1109/SP61157.2025.00241
[19] M. Johnson, "Human-Centric AI in Enterprise Environments," Computers in Human Behavior, vol. 150, p. 108344, 2023. https://doi.org/10.1016/j.chb.2024.108344
[20] State of California, California Consumer Privacy Act (CCPA) — Cal. Civ. Code § 1798.100 et seq., 2020 (as amended by CPRA).
[21] V. C. Hu et al., "Guide to Attribute Based Access Control (ABAC) Definition and Considerations," NIST Special Publication 800-162, National Institute of Standards and Technology, 2014.
[23] K. Williams, "The Future of Conversational Interfaces," in G. S. Rathore (Ed.), Generative AI for Business, pp. 720–745, Wiley, 2023. https://doi.org/10.1002/9781394200801.ch38
[24] R. Kumar, "Security Challenges in Conversational ERP Integration," arXiv preprint arXiv:2407.14116, 2023. https://doi.org/10.48550/arXiv.2407.14116
[25] L. Smith and J. Doe, "Governance Frameworks for Large Language Models," in G. S. Rathore (Ed.), Generative AI for Business, pp. 512–540, Wiley, 2023. https://doi.org/10.1002/9781394200801.ch26
[26] G. S. Rathore, "Introduction to Generative AI in Business," in G. S. Rathore (Ed.), Generative AI for Business, Wiley, 2023. https://doi.org/10.1002/9781394200801.ch3