

Enterprise Agentic AI Reference Architecture

A Comprehensive Reference Architecture for Planning, Reasoning, Retrieval, Tool Use, Memory, Integration, Security, Governance, and Enterprise Operations

Sanjeeve Kumar Gajadi*

*(HCL Technologies, Bengaluru, Karnataka, India)

Email: sanjeevekumar.g@hcltech.com

Abstract:

Agentic AI extends Generative AI from response generation to goal-oriented digital execution. Enterprise agents can interpret objectives, plan tasks, retrieve knowledge, invoke tools, call APIs, maintain memory, coordinate with other agents, and adapt execution based on intermediate results. These capabilities require a reference architecture that separates probabilistic reasoning from deterministic enterprise controls. This paper proposes an Enterprise Agentic AI Reference Architecture spanning experience, agent orchestration, planning and reasoning, enterprise knowledge, memory, model runtime, tool and integration services, identity, policy enforcement, observability, security, Responsible AI, and human oversight. The architecture integrates SAP Business Technology Platform, SAP Business Data Cloud, SAP Datasphere, SAP HANA Cloud, SAP Integration Suite, enterprise APIs, RAG, Knowledge Graphs, and cloud-native services. It provides patterns for secure and scalable agent deployment across hybrid and multi-cloud environments while preserving accountability, bounded autonomy, operational resilience, and business value.

Keywords — Agentic AI, Enterprise AI Architecture, AI Agents, SAP BTP, RAG, Knowledge Graph, AI Orchestration, Tool Use, Memory, Zero Trust, Responsible AI, Human Oversight, Enterprise Integration

I. INTRODUCTION

Agentic AI represents an architectural shift from conversational systems toward goal-oriented software capable of planning, reasoning, using tools, and executing multi-step work. Enterprises are exploring agents for service management, finance, procurement, software engineering, supply chain, analytics, and operational support.

A reference architecture is necessary because agentic solutions combine models, data, memory, APIs, security, human approvals, and business processes. Weak separation between these layers can create excessive autonomy and difficult-to-audit behavior.

II. ARCHITECTURE DRIVERS

Key drivers include automation of knowledge-intensive work, demand for natural-language interfaces, reusable enterprise APIs, richer business data, and advances in foundation models. At the same time, enterprises require predictable controls, privacy, resilience, and integration with existing systems.

The architecture must support rapid innovation without bypassing enterprise architecture, security, data governance, and compliance standards.

III. ARCHITECTURE PRINCIPLES

The reference architecture follows business-value-first, API-first, Zero Trust, least privilege, policy outside the model, reusable knowledge, human oversight, observable execution, secure-by-default, and lifecycle governance principles.

Autonomy should be intentionally selected rather than assumed. The default pattern should be the minimum autonomy required to achieve the business outcome.

IV. EXPERIENCE AND INTERACTION LAYER

Users may interact with agents through enterprise portals, mobile applications, collaboration tools, SAP interfaces, or APIs. The experience layer captures intent, user identity, business context, and consent.

It should display agent limitations, sources, approvals, and execution status so users understand when the system is recommending versus acting.

V. AGENT ORCHESTRATION LAYER

The orchestration layer manages goals, task decomposition, state, dependencies, agent selection, retries, timeouts, and completion criteria.

It can coordinate one agent, multiple specialized agents, or deterministic workflows.

Workflow engines remain valuable for predictable control while LLM-based planners provide flexibility. A hybrid pattern combines both capabilities.

VI. PLANNING AND REASONING LAYER

Planning transforms a goal into structured steps and selects appropriate knowledge and tools. Reasoning should be bounded by policy, available evidence, time limits, and cost budgets.

Critical business rules should not rely solely on model reasoning. Deterministic validation services should verify transaction constraints, compliance rules, and approval requirements.

VII. ENTERPRISE KNOWLEDGE LAYER

RAG, SAP Business Data Cloud, SAP Datasphere, Knowledge Graphs, metadata catalogs, and document repositories provide grounded enterprise context. Retrieval must preserve source authorization, provenance, freshness, and quality.

Combining structured business data with unstructured documents allows agents to reason with richer enterprise context while maintaining reusable governance.

VIII. MEMORY ARCHITECTURE

Agent memory can include session state, task memory, user preferences, learned summaries, and durable business context. Memory should be explicitly classified rather than treated as an unbounded transcript.

Retention, encryption, isolation, deletion, consent, and ownership controls are required because persistent memory can create privacy and contamination risks.

IX. MODEL RUNTIME LAYER

The architecture should support approved foundation models, specialized models, embedding models, and routing services. Model selection may consider task quality, latency, region, cost, context length, and risk.

Model gateways can centralize authentication, policy enforcement, logging, rate limits, and provider abstraction.

X. TOOL AND INTEGRATION LAYER

Agents execute work through tools and APIs. SAP Integration Suite, API Management, Event

Mesh, and governed microservices can expose enterprise capabilities while enforcing schemas, authorization, throttling, and monitoring.

Tools should be narrow, versioned, registered, and independently testable. The agent should never require unrestricted database or administrative access.

XI. IDENTITY AND POLICY LAYER

Users, agents, workloads, and tools require distinct identities. Authorization should combine RBAC, ABAC, task context, data classification, and transaction risk.

Policy engines should determine whether a requested action is allowed, requires approval, must be modified, or should be denied before execution.

XII. HUMAN OVERSIGHT LAYER

Human oversight includes approvals, exception review, escalation, override, and post-action audit. The architecture should place humans at control points proportional to consequence and uncertainty.

High-impact actions such as payments, legal commitments, regulated decisions, and master-data changes should not rely on uncontrolled autonomous execution.

XIII. SECURITY ARCHITECTURE

Agentic AI security covers prompt injection, tool abuse, credential theft, data exfiltration, model misuse, memory poisoning, malicious retrieved content, and supply-chain risk.

Controls include workload identity, secret management, secure gateways, input/output filtering, sandboxing, network segmentation, DevSecOps, threat detection, and incident response.

XIV. OBSERVABILITY AND FINOPS

Observability should capture prompts, plans, model calls, retrieval, tool calls, policy decisions, human approvals, latency, cost, errors, and outcomes. Distributed tracing enables end-to-end reconstruction of agent execution.

FinOps controls can enforce token budgets, model routing policies, concurrency limits, and cost allocation to business units or use cases.

XV. GOVERNANCE AND RESPONSIBLE AI

Governance covers use-case approval, risk classification, model selection, prompt

management, agent registration, testing, monitoring, recertification, and retirement.

Responsible AI controls should address fairness, transparency, explainability, privacy, accountability, and appropriate human control, particularly for consequential decisions.

XVI. HIGH AVAILABILITY AND RESILIENCE

Production agents should use resilient model endpoints, redundant integration services, durable task state, retry controls, and graceful degradation. If an AI service becomes unavailable, the architecture should fail safely rather than bypass controls.

Recovery plans should address model outages, data source failures, policy service failures, and compromised agent identities.

XVII. IMPLEMENTATION METHODOLOGY

Enterprises should start with a bounded use case, define the desired autonomy level, map required data and tools, identify governance controls, and implement measurable success criteria.

Reusable platform services for identity, knowledge, tools, model access, policy, observability, and deployment can then support multiple agent products consistently.

XVIII. USE CASES

Agentic AI can support incident investigation, procurement research, supplier analysis, customer service, financial reconciliation, architecture analysis, software engineering, and knowledge-intensive workflows.

The strongest candidates combine high manual effort with well-defined data sources, tool boundaries, measurable outcomes, and reversible actions.

XIX. CONCLUSION

The Enterprise Agentic AI Reference Architecture provides a structured foundation for moving from experimental agents to governed enterprise systems. It separates reasoning from policy, knowledge from memory, and autonomy from authority.

By integrating enterprise data, tools, identity, security, observability, human oversight, and lifecycle governance, organizations can scale Agentic AI while maintaining trust and operational control.

REFERENCES

- [1] ISO/IEC 42001:2023, Information technology — Artificial intelligence — Management system.
- [2] ISO/IEC 27001:2022, Information security, cybersecurity and privacy protection — Information security management systems.
- [3] ISO 31000:2018, Risk management — Guidelines.
- [4] National Institute of Standards and Technology, Artificial Intelligence Risk Management Framework (AI RMF 1.0), 2023.
- [5] National Institute of Standards and Technology, AI RMF Generative Artificial Intelligence Profile, 2024.
- [6] National Institute of Standards and Technology, SP 800-207: Zero Trust Architecture, 2020.
- [7] The Open Group, TOGAF Standard, 10th Edition, 2022.
- [8] ISACA, COBIT 2019 Framework: Governance and Management Objectives, 2019.
- [9] SAP SE, SAP Business Technology Platform Documentation.
- [10] SAP SE, SAP AI Core and SAP AI Launchpad Documentation.
- [11] SAP SE, SAP Integration Suite Documentation.
- [12] SAP SE, SAP HANA Cloud Documentation.
- [13] SAP SE, SAP Cloud Identity Services Documentation.
- [14] SAP SE, SAP Cloud ALM Documentation.
- [15] SAP SE, SAP Business Data Cloud and SAP Datasphere Documentation.
- [16] OWASP Foundation, OWASP Top 10 for Large Language Model Applications.
- [17] MITRE, ATLAS: Adversarial Threat Landscape for Artificial-Intelligence Systems.
- [18] OECD, Recommendation of the Council on Artificial Intelligence.
- [19] IEEE, Ethically Aligned Design: A Vision for Prioritizing Human Well-being with Autonomous and Intelligent Systems.
- [20] Cloud Native Computing Foundation, Kubernetes Documentation and Cloud Native Security guidance.

AUTHOR BIOGRAPHY

Sanjeev Kumar Gajadi is an SAP Enterprise Architect and Lead Consultant with extensive experience in enterprise architecture, SAP BTP, data and analytics, integration, cloud architecture, security, governance, and AI-enabled transformation. His professional interests include Agentic AI reference architectures, enterprise AI

platforms, SAP BTP, AI governance, and cloud-native transformation.

ACKNOWLEDGMENT

The author acknowledges the contributions of enterprise architecture practitioners, international standards bodies, AI governance communities, security researchers, and SAP documentation teams whose published guidance informed the concepts synthesized in this paper.