

Enterprise Multi-Agent Architecture Framework

A Comprehensive Enterprise Architecture Approach for Coordinating Autonomous AI Agents, Shared Knowledge, Tools, Governance, Security, and Human Oversight

Sanjeeve Kumar Gajadi*

*(HCL Technologies, Bengaluru, Karnataka, India)

Email: sanjeevekumar.g@hcltech.com

Abstract:

Enterprises are rapidly moving from isolated copilots toward ecosystems of specialized AI agents that collaborate across business processes, data domains, applications, and cloud platforms. Multi-agent systems can improve decomposition, parallel execution, specialization, resilience, and decision quality, but they also introduce architectural challenges involving identity, delegation, shared memory, tool access, message routing, conflict resolution, security, observability, and accountability. This paper proposes an Enterprise Multi-Agent Architecture Framework that defines a layered architecture for agent discovery, orchestration, inter-agent communication, knowledge grounding, enterprise integration, policy enforcement, runtime isolation, and human oversight. The framework combines enterprise architecture principles, Zero Trust, Responsible AI, Retrieval-Augmented Generation, Knowledge Graphs, API governance, event-driven integration, and lifecycle management. It enables organizations to design scalable multi-agent environments in which specialized agents can collaborate safely while maintaining traceability, bounded authority, data protection, and measurable business outcomes across hybrid and multi-cloud landscapes.

Keywords — Multi-Agent Systems, Agentic AI, Enterprise AI, AI Orchestration, Agent Identity, Agent Communication, RAG, Knowledge Graph, SAP BTP, AI Governance, Zero Trust, Human Oversight, Responsible AI

I. INTRODUCTION

Enterprise AI is evolving from single-model assistants toward networks of specialized agents that can plan, reason, retrieve information, invoke tools, and collaborate to complete complex business objectives. A multi-agent architecture distributes work across agents with different responsibilities instead of expecting one component to perform every task.

This architectural shift increases flexibility and specialization, but it also creates a new enterprise control plane. Organizations must govern how agents discover one another, exchange context, delegate tasks, share memory, call business systems, resolve conflicting recommendations, and escalate decisions to people.

II. BUSINESS PROBLEM

Large enterprises contain complex processes that cross functional, application, and data boundaries. A single AI assistant may not have the domain expertise, permissions, context, or tools required to complete an end-to-end workflow safely. Multi-agent collaboration can address this limitation by

assigning focused roles such as planner, researcher, data analyst, compliance reviewer, transaction executor, and monitor.

Without a standardized architecture, however, independent agents can duplicate work, create circular delegation, leak sensitive context, invoke incompatible tools, exceed authority, or produce conflicting outcomes. Enterprise adoption therefore requires explicit coordination patterns and governance controls.

III. ARCHITECTURE OBJECTIVES

The framework aims to provide modularity, interoperability, bounded autonomy, resilience, security, explainability, and operational control. Agent capabilities should be reusable across business domains while remaining governed through standard contracts, identity, authorization, observability, and lifecycle policies.

The architecture should support centralized orchestration where appropriate, decentralized collaboration when latency or scale requires it, and hybrid patterns where supervisory agents coordinate specialized domain agents.

IV. AGENT ROLES AND CAPABILITY MODEL

Every agent should have a defined purpose, owner, capability profile, risk tier, approved tools, data scope, and decision boundary. Capability metadata enables discovery and routing while preventing arbitrary access to enterprise functions.

Common roles include planner agents, domain specialists, retrieval agents, verification agents, policy agents, execution agents, monitoring agents, and human-interface agents. Separating responsibilities supports least privilege and improves auditability.

V. MULTI-AGENT ORCHESTRATION

The orchestration layer decomposes goals, selects agents, manages dependencies, tracks task state, and combines results. It should support sequential, parallel, hierarchical, and event-driven collaboration patterns.

Supervisory agents may coordinate workflows, but critical policies must be enforced outside the model through deterministic services. Timeouts, retry limits, circuit breakers, and delegation depth controls prevent runaway execution.

VI. INTER-AGENT COMMUNICATION

Agents require standardized communication contracts containing identity, task intent, context references, confidence, provenance, deadlines, and authorization metadata. Messages should be authenticated, encrypted, and logged.

Enterprise messaging, APIs, and event services can provide reliable transport. Protocol design should distinguish instructions from untrusted content and prevent one agent from silently expanding another agent's authority.

VII. SHARED KNOWLEDGE AND MEMORY

Multi-agent systems require governed access to shared enterprise knowledge. RAG services, SAP Business Data Cloud, SAP Datasphere, Knowledge Graphs, and vector stores can provide common context while preserving source ownership and authorization.

Short-term working memory, long-term agent memory, and enterprise knowledge should be separated. Memory retention, isolation, encryption, deletion, and recertification policies are necessary to reduce privacy and contamination risks.

VIII. TOOL AND API ARCHITECTURE

Agents should access enterprise systems through narrow, registered tools and governed APIs rather than broad administrative interfaces. Each tool should define permitted actions, input schemas, output schemas, rate limits, and approval requirements.

SAP Integration Suite and API Management can mediate interactions with SAP S/4HANA, SuccessFactors, Ariba, Business Data Cloud, and external platforms while providing policy enforcement and auditability.

IX. IDENTITY, DELEGATION, AND TRUST

Every production agent requires a distinct machine identity. Delegation tokens should be scoped to the specific task, resource, action, and time window so privileges do not propagate indefinitely through the agent network.

Zero Trust principles require continuous verification of user identity, agent identity, tool identity, data sensitivity, and transaction risk. High-impact delegation should require human approval or additional policy checks.

X. CONFLICT RESOLUTION AND CONSENSUS

Multiple agents may produce different recommendations because they use different evidence, models, or objectives. The architecture should define mechanisms for ranking, voting, arbitration, verification, or escalation instead of assuming agreement.

Evidence provenance, confidence scores, business rules, and policy agents can support resolution. High-impact disagreements should be visible to human decision makers rather than automatically suppressed.

XI. SECURITY ARCHITECTURE

Multi-agent security must protect prompts, messages, memory, tools, APIs, model endpoints, credentials, and data. Threats include prompt injection, malicious delegation, agent impersonation, poisoned context, tool abuse, and cross-agent data leakage.

Controls include workload identity, least privilege, mTLS, encryption, content filtering, secret management, sandboxing, policy-as-code, network segmentation, and continuous threat detection.

XII. OBSERVABILITY AND OPERATIONS

Operational telemetry should capture agent identity, parent-child relationships, task state, messages, tool calls, model versions, retrieved evidence, policy decisions, latency, cost, errors, and human interventions.

Distributed tracing is especially important because one business request may involve many agents and systems. Enterprise observability should reconstruct the complete execution graph for audit and incident response.

XIII. GOVERNANCE AND RESPONSIBLE AI

Governance establishes agent registration, risk classification, testing, approval, deployment, recertification, and retirement. Architecture Review Boards and AI Governance Committees should review high-impact multi-agent use cases.

Responsible AI controls should address transparency, fairness, privacy, explainability, accountability, human oversight, and safe failure. Organizations should document which agent made or influenced each consequential decision.

XIV. IMPLEMENTATION METHODOLOGY

Implementation should begin with a bounded multi-agent use case and clearly defined agent roles, data sources, tools, policies, and success measures. Teams should validate coordination quality, security, latency, cost, and escalation behavior before scaling.

Reusable agent contracts, tool registries, policy services, knowledge services, observability, and deployment pipelines should then be standardized as an enterprise platform capability.

XV. INDUSTRY USE CASES

Manufacturing can use multi-agent systems for maintenance planning, production analysis, quality investigation, and supplier coordination. Financial services can use specialized agents for research, policy checks, risk review, and customer operations.

Enterprise IT can use agents for incident triage, architecture analysis, software engineering, cloud operations, and knowledge management while preserving separation of duties.

XVI. FUTURE TRENDS

Future multi-agent ecosystems will include richer agent discovery, standardized interoperability, graph-based collaboration, dynamic teams, autonomous policy enforcement, and secure agent marketplaces.

Enterprise architecture will increasingly manage agents as digital workforce components with explicit identities, capabilities, service-level objectives, risk profiles, and lifecycle governance.

XVII. CONCLUSION

Enterprise multi-agent systems can extend AI from isolated assistance to coordinated digital execution, but value depends on disciplined architecture and governance. The proposed framework combines agent specialization, orchestration, communication, shared knowledge, tool governance, identity, Zero Trust, observability, and human oversight.

By treating multi-agent collaboration as an enterprise architecture concern rather than a collection of prompts, organizations can scale Agentic AI while maintaining security, accountability, interoperability, and business 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

Sanjeeve 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 multi-agent architecture, Agentic AI, SAP BTP, enterprise knowledge, AI governance, and digital 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.