International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 6, November-December 2026 |

DOI:10.15662/IJEETR.2026.0806005

Secure and Governed Multi-Agent AI Architectures Using Identity-Aware Authorization and Fine-Grained Access Control

Sanjeev Vemireddy

Staff Software Engineer, Intuit Inc., USA

Publication History: Received: 23.09.2026; Revised: 26.10.2026; Accepted: 01.11.2026; Published: 10.11.2026.

ABSTRACT: The growing adoption of multi-agent artificial intelligence (AI) systems within enterprise environments has introduced new challenges in securing autonomous interactions, controlling access to sensitive resources, and maintaining governance across distributed computational workflows. Unlike conventional AI applications, multi-agent architectures involve multiple autonomous entities that independently interpret tasks, exchange contextual information, invoke external tools, and coordinate decisions across heterogeneous enterprise systems. These capabilities introduce complex security risks associated with unauthorized resource access, excessive agent privileges, identity impersonation, unintended information disclosure, and insufficient accountability for machine-generated actions. Traditional role-based authorization models and perimeter-oriented security mechanisms are often inadequate for managing the dynamic relationships and contextual dependencies inherent in autonomous agent ecosystems.

This article presents a generalized architectural framework for designing secure and governed multi-agent AI systems through identity-aware authorization, fine-grained access control, and policy-driven governance mechanisms. The proposed approach integrates workload identity management, attribute-based access control, policy-based authorization, least-privilege enforcement, and continuous trust evaluation to establish verifiable security boundaries between agents, tools, data resources, and enterprise services. Particular emphasis is placed on securing agent-to-agent communication, enforcing context-sensitive permissions during tool execution, managing delegated authority, and establishing auditable decision pathways across distributed workflows. The architecture further incorporates centralized policy decision capabilities, distributed enforcement mechanisms, cryptographic identity verification, and continuous monitoring to support scalable governance without unnecessarily restricting agent autonomy. The study examines architectural components, authorization workflows, operational security considerations, and evaluation criteria for implementing trustworthy multi-agent platforms. By combining identity-centric security principles with adaptive authorization and governance controls, the proposed framework provides a foundation for developing enterprise AI ecosystems that balance autonomous execution, operational scalability, regulatory accountability, and protection of organizational information assets.

KEYWORDS: Multi-Agent Artificial Intelligence, Identity-Aware Authorization, Fine-Grained Access Control, Agentic AI Security, Zero Trust Architecture, Attribute-Based Access Control (ABAC), Role-Based Access Control (RBAC), Policy-Based Access Control, Agent-to-Agent Communication, Least-Privilege Access, Enterprise AI Governance, Workload Identity, Secure Tool Invocation, Delegated Authorization, Autonomous AI Systems.

I. INTRODUCTION

The rapid evolution of artificial intelligence (AI) from isolated predictive models toward autonomous, collaborative systems has accelerated the adoption of multi-agent architectures across enterprise computing environments. Modern multi-agent AI platforms consist of intelligent software agents capable of independently interpreting objectives, coordinating activities, accessing organizational resources, and executing complex workflows with limited human intervention. These systems increasingly support business process automation, enterprise service orchestration, intelligent decision-making, cybersecurity operations, and distributed data processing. By enabling specialized agents to collaborate through structured communication and delegated execution, organizations can achieve greater operational flexibility, scalability, and automation efficiency.



|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 6, November-December 2026 |

DOI:10.15662/IJEETR.2026.0806005

However, the transition from centralized AI applications to distributed autonomous agent ecosystems introduces significant security and governance challenges. Unlike traditional enterprise applications, where user identities and application permissions are relatively static, multi-agent environments involve dynamically interacting software entities operating across multiple trust boundaries. Individual agents may communicate with other agents, invoke external application programming interfaces (APIs), retrieve confidential information, and initiate transactions on behalf of users or enterprise services. Without appropriate security controls, these interactions can expose organizations to unauthorized resource access, privilege escalation, identity spoofing, sensitive data leakage, and untraceable automated decisions.

Conventional access control mechanisms, including role-based access control (RBAC) and static permission assignments, provide foundational security capabilities but may lack the contextual flexibility required for autonomous agent interactions. An agent authorized to retrieve enterprise information for one business process should not automatically receive equivalent privileges when executing unrelated tasks or operating under a different user context. Furthermore, permissions delegated between cooperating agents must remain constrained by the original authorization scope, preventing unintended privilege expansion across multi-stage workflows. These requirements highlight the importance of incorporating identity awareness, contextual authorization, and fine-grained policy enforcement directly into multi-agent system architectures.

Identity-aware authorization provides a security foundation by associating every agent, user, service, and tool invocation with a verifiable identity and an explicitly defined authorization context. When combined with attribute-based access control (ABAC), policy-driven enforcement, and Zero Trust principles, identity-aware security enables organizations to evaluate access requests based on multiple contextual factors, including agent identity, delegated authority, resource sensitivity, requested operation, workflow purpose, and environmental conditions. Rather than assuming that previously authenticated agents remain trustworthy, this approach continuously enforces authorization at security-relevant interaction boundaries, thereby reducing the risks associated with unrestricted autonomous execution.

A critical architectural consideration is the separation of agent reasoning capabilities from authorization authority. Although intelligent agents may autonomously determine how to accomplish assigned objectives, security-sensitive actions should remain subject to independently enforced organizational policies. This separation prevents agent-generated instructions, manipulated prompts, or compromised reasoning processes from directly overriding access restrictions. Centralized policy decision services, distributed policy enforcement points, and secure workload identities can collectively establish consistent authorization behavior across heterogeneous agent frameworks and enterprise technology platforms.

This article examines the architectural principles and security mechanisms required to establish trustworthy multi-agent AI environments through identity-aware authorization and fine-grained access control. It presents a generalized security framework integrating identity management, contextual policy evaluation, controlled delegation, secure agent-to-agent communication, and governance-oriented observability. The discussion further explores authorization workflows, architectural design considerations, implementation challenges, and evaluation approaches for balancing autonomous agent capabilities with enterprise security requirements.

The primary contribution of this work is an integrated architectural perspective that treats identity, authorization, and governance as foundational components of multi-agent AI systems rather than supplementary security layers. By embedding least-privilege enforcement, explicit trust boundaries, and auditable authorization decisions throughout agent execution workflows, the proposed approach supports the development of scalable, secure, and accountable autonomous enterprise platforms.

II. SECURITY CHALLENGES AND ARCHITECTURAL REQUIREMENTS OF MULTI-AGENT AI SYSTEMS

The distributed and autonomous characteristics of multi-agent artificial intelligence introduce security challenges that extend beyond those encountered in conventional enterprise applications. In traditional architectures, security policies are generally enforced around predefined users, services, and application boundaries. Multi-agent AI systems, however, involve dynamically coordinated software entities that continuously exchange information, delegate tasks, invoke external tools, and access enterprise resources. Consequently, establishing secure operational boundaries requires identity-aware access management, contextual policy evaluation, and continuous governance throughout the agent execution lifecycle.

International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 6, November-December 2026 |

DOI:10.15662/IJEETR.2026.0806005

A. Dynamic Agent Identities and Trust Establishment

Identity management represents a foundational challenge in distributed multi-agent environments. Autonomous agents may be instantiated dynamically, assigned temporary responsibilities, and terminated following task completion. Each agent must therefore possess a distinguishable and verifiable identity that enables enterprise security infrastructure to authenticate its interactions and associate its activities with an accountable execution context.

Traditional application authentication mechanisms frequently rely on shared credentials, long-lived API keys, or broadly scoped service accounts. Applying these mechanisms to multi-agent systems can create excessive permissions and weaken accountability when several agents operate under the same identity.

A secure architecture should establish dedicated workload identities, short-lived authentication credentials, and cryptographically verifiable identity assertions. Identity lifecycle controls must support agent registration, authentication, credential rotation, expiration, and revocation. These mechanisms help ensure that only authenticated agents operating within approved security contexts can participate in enterprise workflows.

B. Fine-Grained Authorization and Privilege Management

Multi-agent AI systems frequently execute operations that require different levels of authorization. For example, an analytical agent may require read-only access to financial records, while a transaction-processing agent may require permission to initiate specific business operations. Granting identical privileges to both agents introduces unnecessary security exposure.

Fine-grained authorization addresses this limitation by evaluating access requests against resource-specific, action-specific, and context-dependent security policies. Attribute-based access control and policy-based authorization enable decisions to incorporate agent identity, resource classification, execution purpose, delegated authority, and environmental conditions.

The principle of least privilege should be enforced at every sensitive interaction, ensuring that agents receive only the permissions necessary to complete authorized tasks. Authorization decisions should also be reassessed when execution conditions change, particularly when agents cross application or organizational trust boundaries.

C. Agent-to-Agent Communication and Delegated Authority

Collaboration between autonomous agents introduces additional risks when tasks and permissions are transferred across distributed workflows. An initiating agent may delegate a processing activity to another specialized agent, which may subsequently invoke enterprise APIs or request assistance from additional agents.

Without controlled delegation, downstream agents may unintentionally obtain broader permissions than those granted to the original requester. This creates opportunities for privilege escalation, unauthorized resource access, and misuse of inherited execution authority.

Secure delegation requires explicit authorization scopes, bounded execution contexts, and verifiable relationships between initiating identities and delegated operations. Delegation credentials should identify the authorized audience, permitted actions, applicable resources, and validity period. Each receiving agent must independently verify the delegated authority before executing security-sensitive operations.

Furthermore, mutual authentication and encrypted communication channels should protect agent-to-agent interactions against impersonation, interception, and message manipulation.

D. Prompt Injection and Unauthorized Tool Execution

Multi-agent systems often process external documents, retrieved information, user-generated instructions, and outputs generated by other agents. These inputs may contain malicious or misleading instructions intended to manipulate agent behavior.

Prompt injection presents a particularly important security risk because an agent may interpret untrusted content as operational guidance and attempt to invoke tools, retrieve sensitive information, or perform unauthorized actions.

Although input filtering and instruction isolation can reduce exposure, they cannot independently guarantee secure behavior. Authorization enforcement must therefore remain separate from agent-generated reasoning and model outputs. Every security-sensitive tool invocation should pass through a policy enforcement mechanism that independently verifies the requesting identity, delegated permissions, resource scope, and requested action. This architectural separation prevents compromised agent instructions from directly granting access to protected enterprise resources.

E. Governance, Auditability, and Operational Accountability

Autonomous execution creates governance challenges because multiple agents may contribute to a single business outcome. Organizations must maintain sufficient visibility to reconstruct the sequence of agent interactions, authorization decisions, tool executions, and resulting resource modifications.

International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 6, November-December 2026 |

DOI:10.15662/IJEETR.2026.0806005

Traditional application logging may not adequately capture relationships between initiating users, coordinating agents, delegated tasks, and downstream operations. A governance-oriented architecture should therefore maintain structured records containing agent identities, workflow identifiers, authorization outcomes, policy versions, delegation relationships, and execution timestamps.

Sensitive information should be protected through appropriate logging minimization, access restrictions, and retention controls. These governance capabilities support security investigations, policy compliance reviews, operational diagnostics, and accountability for automated decisions.

F. Architectural Security Requirements

Addressing the identified risks requires security capabilities to be incorporated throughout the multi-agent architecture rather than implemented exclusively at external application boundaries.

TABLE I. SECURITY CHALLENGES AND CORRESPONDING ARCHITECTURAL REQUIREMENTS

Security challenge	Architectural requirement	Expected security outcome
Dynamic agent identities	Verifiable workload identities and credential lifecycle management	Reliable agent authentication
Excessive agent privileges	Fine-grained, context-aware authorization	Least-privilege enforcement
Uncontrolled task delegation	Scoped delegation and authority verification	Prevention of privilege expansion
Unauthorized tool invocation	Independent policy enforcement at tool boundaries	Controlled enterprise resource access
Agent impersonation	Mutual authentication and cryptographic identity validation	Trusted agent communication
Prompt injection	Isolation of untrusted content from authorization decisions	Reduced unauthorized execution risk
Sensitive data exposure	Resource-level access restrictions and data classification policies	Protection of confidential information
Limited operational visibility	Distributed audit logging and identity-linked traceability	Improved governance and accountability
Compromised or expired credentials	Short-lived credentials and revocation mechanisms	Reduced credential misuse
Inconsistent security policies	Centralized policy administration with distributed enforcement	Consistent enterprise authorization

G. Design Principles for Secure Multi-Agent Architectures

A secure multi-agent AI platform should follow several complementary architectural principles.

Identity-first security: Every participating agent, user, service, and tool should operate under a verifiable identity that supports authentication, authorization, and traceability.

Explicit authorization: Authentication establishes identity but must not automatically grant unrestricted resource access. Each protected operation requires an authorization decision appropriate to its execution context.

Least-privilege delegation: Delegated permissions must remain within the authority of the initiating principal and should be constrained by task-specific scopes, resource boundaries, and expiration conditions.

Separation of reasoning and enforcement: AI-generated plans and instructions should not act as authoritative security decisions. Independent policy enforcement components must validate protected operations.

III. PROPOSED IDENTITY-AWARE MULTI-AGENT SECURITY ARCHITECTURE

The proposed architecture establishes a unified security and governance framework for enterprise multi-agent artificial intelligence systems by integrating verifiable agent identities, fine-grained authorization policies, controlled task delegation, and distributed enforcement mechanisms. The framework is designed to support autonomous agent collaboration while maintaining explicit security boundaries between users, intelligent agents, external tools, and enterprise information resources.

International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 6, November-December 2026 |

DOI:10.15662/IJEETR.2026.0806005

Unlike conventional architectures that primarily enforce security at application entry points, the proposed model incorporates identity verification and authorization throughout the agent execution lifecycle. Every security-sensitive interaction is evaluated against organizational policies, ensuring that autonomous reasoning and task orchestration do not independently determine access privileges. The framework adopts Zero Trust principles, separating agent intelligence from security authority and enabling consistent governance across heterogeneous deployment environments.

A. Architectural Overview

The proposed framework consists of six interconnected architectural layers: the enterprise interaction layer, multi-agent orchestration layer, identity and trust management layer, authorization and policy governance layer, secure tool execution layer, and enterprise resource layer. A cross-cutting governance and observability capability provides traceability and operational accountability across these layers.

Each layer addresses a distinct responsibility while interacting through standardized security interfaces.

1. Enterprise Interaction Layer

The enterprise interaction layer serves as the controlled entry point for business users, applications, and automated enterprise workflows. It receives task requests, authenticates initiating principals, and establishes the execution context before allowing requests to enter the multi-agent environment.

Authentication can integrate with enterprise identity providers using federated identity mechanisms, such as OpenID Connect and OAuth-based authorization flows. The resulting security context identifies the initiating principal, applicable organizational attributes, and authorized operational boundaries.

This layer ensures that autonomous workflows originate from authenticated and appropriately authorized sources.

2. Multi-Agent Orchestration Layer

The orchestration layer coordinates specialized AI agents responsible for planning, information retrieval, analytical processing, workflow execution, and enterprise system interaction.

A coordinating agent decomposes business objectives into smaller activities and distributes them to specialized agents according to their capabilities. However, task assignment does not automatically transfer unrestricted authority.

Each agent operates within a defined execution context that includes its own workload identity and, where applicable, delegated permissions associated with the initiating principal. Before requesting protected resources or invoking privileged tools, the agent must obtain authorization through the designated enforcement interface.

This approach permits distributed collaboration without allowing agent autonomy to bypass organizational security requirements.

3. Identity and Trust Management Layer

The identity and trust management layer establishes verifiable identities for participating agents, execution workloads, and enterprise services.

It incorporates an identity provider, workload identity issuance mechanisms, credential lifecycle management, and identity verification services. Technologies such as SPIFFE-based workload identities and mutually authenticated TLS can support trusted communication in compatible infrastructure environments.

Each agent receives a distinct identity bound to its execution workload rather than relying on shared service credentials. Short-lived credentials reduce exposure associated with long-term secrets, while revocation and workload termination controls limit continued access following security events.

Identity verification establishes who or what is requesting access; authorization remains a separate decision.

4. Authorization and Policy Governance Layer

The authorization layer represents the principal security control within the proposed framework. It evaluates access requests using organizational policies and contextual information before protected operations are permitted.

The layer incorporates three primary architectural components:

- Policy Administration Point (PAP): Manages authorization policies, policy versions, approval workflows, and controlled deployment of security rules.
- Policy Decision Point (PDP): Evaluates access requests based on verified identities, applicable permissions, resource attributes, and execution context.
- Policy Enforcement Point (PEP): Intercepts protected operations and enforces authorization decisions before execution.

The framework supports hybrid access control models that combine RBAC for broad organizational responsibilities with ABAC for resource-specific and context-sensitive authorization.

International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 6, November-December 2026 |

DOI:10.15662/IJEETR.2026.0806005

For example, an agent assigned to financial analysis may be authorized to retrieve aggregated financial data but prohibited from modifying accounting transactions. The policy decision considers both the agent's operational role and the requested action's specific context.

5. Secure Tool Execution Layer

The secure tool execution layer establishes a controlled boundary between autonomous reasoning and enterprise operations.

Agents interact with external applications through approved connectors, API gateways, or managed tool interfaces. These interfaces enforce identity verification, authorization decisions, input validation, execution constraints, and audit recording.

Sensitive actions, including financial transaction updates, administrative configuration changes, and confidential data extraction, may require additional controls such as human approval, transaction limits, or task-specific authorization.

Tool access is therefore treated as an independently governed operation rather than an automatic consequence of an agent's ability to generate a tool invocation.

6. Enterprise Resource Layer

The enterprise resource layer contains the applications, databases, document repositories, business services, and external APIs accessed by the multi-agent platform.

Enterprise resources retain their own security controls, including resource-level permissions, data classification restrictions, network access protections, and transaction authorization rules.

The proposed architecture does not replace these protections. Instead, it complements them through identity-aware authorization at the agent execution boundary and, where supported, propagation of verified authorization context to downstream services.

This layered approach reduces dependence on a single enforcement mechanism and preserves defense-in-depth security.

B. Integrated Architecture and Component Interactions

Figure 1. Proposed Identity-Aware and Policy-Governed Multi-Agent AI Security Architecture

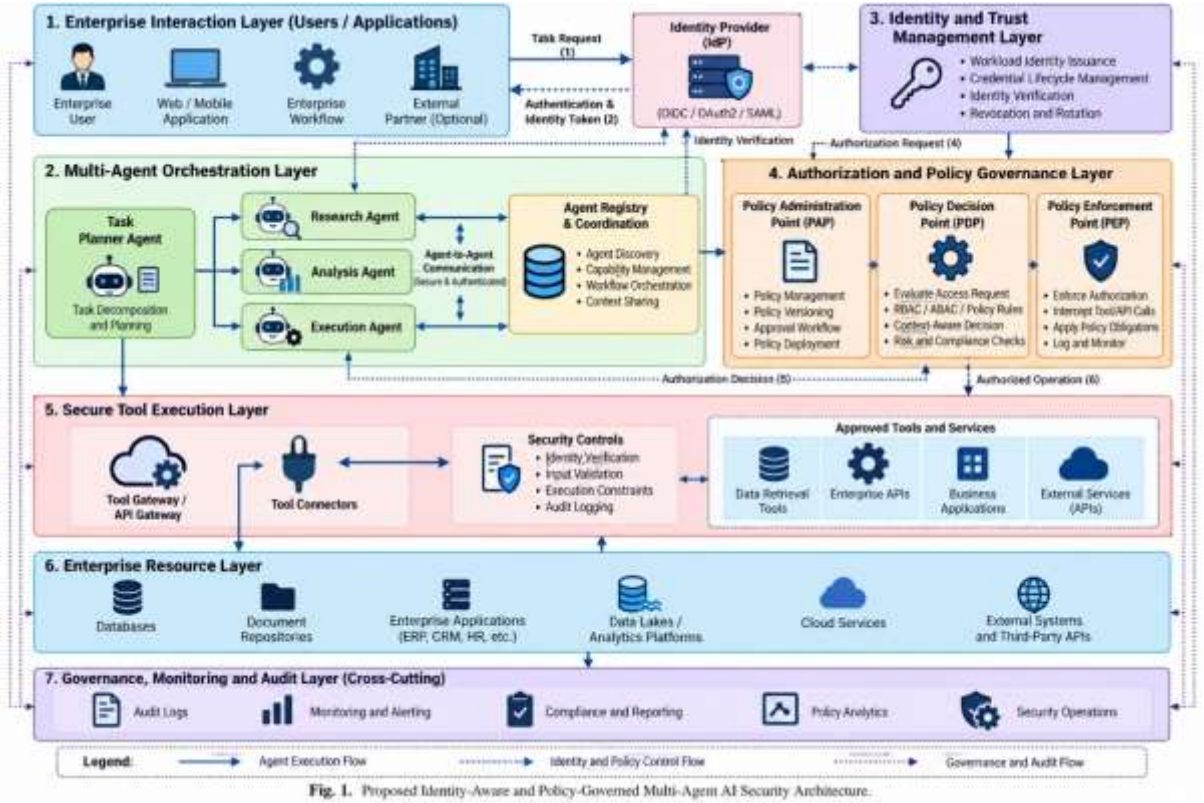


Fig. 1. Proposed Identity-Aware and Policy-Governed Multi-Agent AI Security Architecture.

International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 6, November-December 2026 |

DOI:10.15662/IJEETR.2026.0806005

The diagram should distinguish the agent execution path from the identity and policy control paths. It should also show governance and audit collection spanning the principal architectural components.

The central design requirement is that access to protected enterprise resources must pass through an authorized enforcement boundary. Agents may independently generate plans and exchange task information, but they cannot legitimately acquire additional enterprise privileges simply by modifying instructions or delegating execution.

C. Identity-Aware Authorization Workflow

The proposed architecture implements a structured authorization workflow that validates execution authority before agents interact with protected enterprise resources.

1. Request authentication: An enterprise user or application submits a task. The interaction layer verifies the initiating identity and establishes the associated security context.
2. Agent assignment: The orchestration component assigns the task to an appropriate agent possessing a verified workload identity.
3. Delegation validation: If the agent acts on behalf of another principal, the system validates the delegated authority, including permitted scope, audience, expiration, and delegation chain.
4. Authorization evaluation: A tool gateway or other PEP sends the verified request context to the PDP, which evaluates relevant organizational policies.
5. Policy enforcement: Authorized operations proceed under specified conditions. Denied requests are blocked, and additional obligations such as approval or data masking are enforced where applicable.
6. Controlled resource execution: The approved tool or enterprise connector performs the operation using credentials and permissions constrained to the authorized context.

IV. FINE-GRAINED ACCESS CONTROL AND SECURE AGENT-TO-AGENT COMMUNICATION

Fine-grained access control is a critical component of secure multi-agent AI architectures because autonomous agents frequently interact with sensitive enterprise resources, execute delegated tasks, and exchange information across multiple security domains. Unlike conventional application environments, where permissions are generally associated with predefined users and service accounts, multi-agent systems require authorization mechanisms capable of evaluating dynamic execution contexts, agent identities, resource sensitivity, and delegated authority. The proposed framework integrates role-based access control (RBAC), attribute-based access control (ABAC), policy-driven authorization, and Zero Trust principles to establish consistent security enforcement throughout distributed agent interactions.

Fig. 2. Fine-Grained Authorization and Secure Agent-to-Agent Communication in a Multi-Agent AI Architecture.

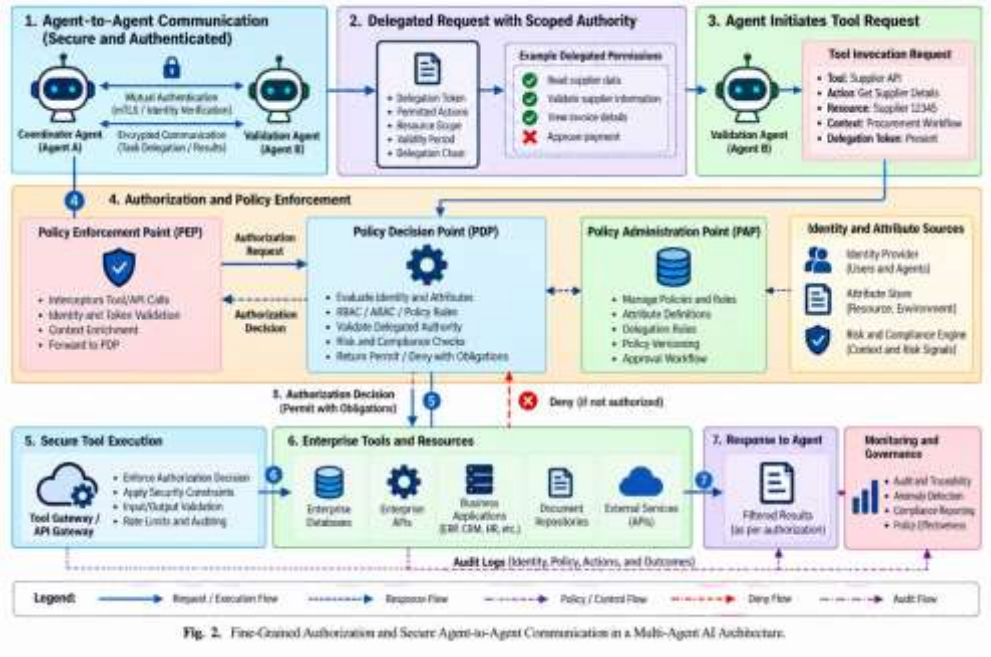


Fig. 2. Fine-Grained Authorization and Secure Agent-to-Agent Communication in a Multi-Agent AI Architecture.

International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 6, November-December 2026 |

DOI:10.15662/IJEETR.2026.0806005

A. Role-Based and Attribute-Based Access Control

Role-based access control provides a foundational mechanism for assigning permissions according to predefined organizational responsibilities. Within a multi-agent ecosystem, roles may represent analytical agents, workflow coordinators, transaction processors, or administrative agents. Although RBAC simplifies permission administration, static role assignments may grant broader access than necessary when agents participate in multiple workflows.

Attribute-based access control addresses this limitation by evaluating authorization requests using contextual information associated with subjects, resources, actions, and execution environments. For example, an analytical agent may be permitted to retrieve financial information only when processing an approved reporting task and operating under an authorized business context.

Combining RBAC and ABAC provides a balanced authorization model. RBAC establishes baseline permissions, while ABAC introduces contextual restrictions based on resource classifications, delegated authority, workflow objectives, and environmental conditions. This hybrid model supports least-privilege enforcement without requiring an excessive number of narrowly defined roles.

B. Policy-Based Authorization and Distributed Enforcement

Policy-based authorization separates security decision-making from agent execution logic. Rather than embedding authorization rules directly within individual agents, the proposed architecture maintains centrally governed policies that are evaluated by dedicated policy decision components.

When an agent requests access to a protected resource, a policy enforcement point intercepts the request and forwards the relevant identity and contextual attributes to a policy decision point. The decision component evaluates the request against applicable organizational policies and returns an authorization outcome.

A permitted operation proceeds only when all mandatory policy conditions are satisfied. Requests that violate security policies, lack required attributes, or exceed delegated permissions are denied.

Distributed policy enforcement points can be deployed at agent communication interfaces, API gateways, tool execution services, and enterprise resource connectors. Centralized policy administration promotes consistency, while distributed enforcement supports scalable execution across heterogeneous infrastructure.

To maintain security during service disruptions, enforcement mechanisms should adopt fail-closed behavior for protected operations when mandatory authorization decisions cannot be obtained or validated.

C. Secure Agent-to-Agent Communication

Agent-to-agent communication enables autonomous systems to coordinate activities, exchange intermediate results, and delegate specialized processing tasks. However, unrestricted communication introduces risks associated with identity impersonation, unauthorized message consumption, information leakage, and malicious task redirection.

Secure communication requires mutual authentication, encrypted transport, and explicit authorization of communication relationships. Each participating agent should possess a verifiable workload identity that can be authenticated independently of the application-level instructions contained in exchanged messages.

Transport Layer Security with mutual authentication can protect communication channels, while workload identity mechanisms provide cryptographically verifiable identifiers for participating services.

Authorization policies should determine which agents are permitted to communicate, exchange specific categories of information, and initiate delegated operations. Sensitive messages may additionally require integrity verification, replay protection, and appropriate restrictions on data propagation.

Importantly, successful authentication between two agents does not automatically authorize either agent to execute privileged actions on behalf of the other.

D. Controlled Task Delegation and Privilege Propagation

Task delegation is fundamental to multi-agent coordination because complex enterprise workflows often require several specialized agents to contribute to a shared objective. Nevertheless, delegated execution creates security challenges when downstream agents inherit permissions without appropriate restrictions.

The proposed framework employs scoped delegation to ensure that transferred authority remains limited to the permissions necessary for the assigned task. Delegation credentials should specify the authorized subject, intended recipient, permitted resources, approved operations, and expiration conditions.

Consider an enterprise procurement workflow in which a coordinating agent delegates supplier verification to a specialized validation agent. The validation agent may receive authorization to retrieve supplier registration details but should not automatically gain permission to modify payment information or approve financial transactions.

International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 6, November-December 2026 |

DOI:10.15662/IJEETR.2026.0806005

When an agent delegates work to another agent, the resulting authorization scope must remain within the effective authority of the delegating principal. Downstream delegation should impose equivalent or stricter restrictions and preserve traceability to the initiating business context.
This model prevents uncontrolled privilege propagation while allowing autonomous agents to collaborate across distributed workflows.

E. Fine-Grained Authorization for Tool Invocation
External tool invocation represents one of the most security-sensitive activities within multi-agent AI systems. Agents may interact with databases, enterprise APIs, cloud services, financial applications, or administrative interfaces capable of modifying organizational resources.
The proposed framework therefore introduces a governed tool execution boundary that validates each protected invocation before forwarding requests to enterprise systems.
Authorization decisions incorporate the requesting agent's verified identity, delegated permissions, targeted tool, requested operation, resource classification, and execution context. Additional enforcement conditions may include rate limits, transaction thresholds, data masking requirements, or human approval.
For example, an agent authorized to generate financial reports may execute read-only database queries but remain prohibited from initiating payment transactions or modifying accounting records. Even when an AI model generates instructions requesting such actions, the independent authorization mechanism prevents execution outside the approved security scope.
Separating autonomous reasoning from tool authorization reduces the potential impact of prompt injection, compromised agents, and unintended model behavior.

F. Comparative Evaluation of Access Control Approaches
Different authorization mechanisms provide complementary capabilities for enterprise multi-agent environments. Selecting an appropriate strategy requires consideration of security granularity, contextual adaptability, administrative complexity, and distributed execution requirements.

TABLE III. COMPARISON OF ACCESS CONTROL MECHANISMS FOR MULTI-AGENT AI SYSTEMS

Access control model	Primary mechanism	Advantages	Limitations
RBAC	Permissions assigned through roles	Simple administration and role governance	Limited contextual flexibility
ABAC	Decisions based on subject, resource, action, and environment attributes	Fine-grained, context-aware enforcement	Requires reliable attribute management
Policy-based access control	Authorization through centrally managed policies	Consistent governance and independent enforcement	Policy evaluation and deployment complexity
Relationship-based access control (ReBAC)	Permissions evaluated through entity relationships	Supports hierarchical and delegated relationships	Relationship graph management overhead
Zero Trust authorization	Explicit verification and least-privilege access at protected boundaries	Reduces implicit trust and lateral movement	Requires coordinated identity and enforcement infrastructure
Hybrid authorization	Combination of roles, attributes, relationships, and policies	Flexible enterprise-wide authorization	Increased policy design and testing requirements

The comparison suggests that no single authorization model independently addresses all security requirements associated with autonomous multi-agent systems. A hybrid architecture provides greater flexibility by combining baseline role permissions with contextual evaluation, relationship-aware delegation, and independent policy enforcement.

G. Authorization Performance and Operational Considerations
Fine-grained authorization introduces additional processing requirements because protected interactions may require repeated identity verification, attribute retrieval, and policy evaluation. In high-volume multi-agent environments, inefficient authorization infrastructure can increase request latency and reduce workflow throughput.



|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 6, November-December 2026 |

DOI:10.15662/IJEETR.2026.0806005

Scalable implementations should employ distributed enforcement points, efficient policy evaluation engines, and carefully controlled authorization caching. Cached decisions must remain constrained by validity periods, policy changes, identity revocation requirements, and relevant contextual changes.

Policy management processes should include version control, automated testing, conflict detection, staged deployment, and controlled rollback. These capabilities help prevent inconsistent authorization behavior across distributed agent workloads.

Organizations should also monitor authorization latency, denial rates, delegation failures, credential validation errors, and attempted privilege escalation. Such metrics provide operational visibility while supporting continuous improvement of security policy effectiveness.

Overall, fine-grained access control and secure agent-to-agent communication establish the operational security foundation for governed multi-agent AI systems. By combining cryptographic identity verification, contextual authorization, restricted delegation, and independently enforced tool permissions, the proposed framework enables collaborative agent execution while preserving enterprise control over sensitive information and business operations.

VI. CONCLUSION

The increasing adoption of autonomous multi-agent artificial intelligence systems has fundamentally transformed enterprise automation by enabling intelligent agents to collaborate, exchange information, and execute complex workflows across distributed technology environments. However, these capabilities introduce substantial security challenges involving dynamic agent identities, unauthorized resource access, privilege escalation, uncontrolled task delegation, and limited operational accountability. Addressing these risks requires an architectural approach that incorporates identity verification, fine-grained authorization, secure communication, and continuous governance throughout the agent execution lifecycle.

This article presented a generalized security framework for multi-agent AI systems based on identity-aware authorization, hybrid access control mechanisms, and policy-driven enforcement. The proposed architecture integrates workload identity management, role-based and attribute-based authorization, scoped delegation, independent policy decision services, and controlled tool execution boundaries. By separating autonomous agent reasoning from security enforcement, the framework establishes explicit authorization checkpoints that prevent agent-generated instructions from independently granting access to protected enterprise resources. Secure agent-to-agent communication, short-lived credentials, contextual policy evaluation, and least-privilege enforcement further strengthen protection across distributed execution environments.

An important contribution of the proposed framework is its emphasis on governance and accountability as integral architectural capabilities. Identity-linked audit records, policy evaluation traceability, authorization monitoring, and controlled credential lifecycles provide a foundation for investigating security incidents, verifying compliance, and maintaining organizational oversight of automated operations. The combination of centrally governed policies and distributed enforcement mechanisms also offers a practical architectural direction for maintaining authorization consistency across heterogeneous enterprise applications and cloud environments.

Although the proposed framework provides a conceptual foundation for secure multi-agent execution, its operational effectiveness requires further empirical validation. Future research should examine authorization latency under high-volume agent interactions, scalable policy evaluation, automated detection of malicious delegation patterns, and adaptive security controls capable of responding to changing execution risks. Additional investigation into interoperability between heterogeneous agent frameworks, verifiable authorization provenance, and privacy-preserving governance mechanisms may further strengthen enterprise adoption.

Ultimately, trustworthy multi-agent AI systems require security architectures that preserve the benefits of autonomous collaboration without compromising organizational control over identities, permissions, information assets, and critical business operations. Identity-aware authorization and fine-grained access control provide essential foundations for achieving this balance, supporting the development of scalable, accountable, and security-governed enterprise AI ecosystems.



|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 6, November-December 2026 |

DOI:10.15662/IJEETR.2026.0806005

REFERENCES

[1] S. Rose, O. Borchert, S. Mitchell, and S. Connelly, "Zero Trust Architecture," National Institute of Standards and Technology, NIST Special Publication 800-207, Aug. 2020.
[2] V. C. Hu, D. Ferraiolo, D. R. Kuhn, A. Schnitzer, K. Sandlin, R. Miller, and K. Scarfone, "Guide to Attribute Based Access Control (ABAC) Definition and Considerations," National Institute of Standards and Technology, NIST Special Publication 800-162, 2014, updated Aug. 2019.
[3] Joint Task Force, "Security and Privacy Controls for Information Systems and Organizations," National Institute of Standards and Technology, NIST Special Publication 800-53, Revision 5, Sept. 2020.
[4] M. Jones, A. Nadalin, B. Campbell, J. Bradley, and C. Mortimore, "OAuth 2.0 Token Exchange," Internet Engineering Task Force, RFC 8693, Jan. 2020.
[5] E. Tabassi, "Artificial Intelligence Risk Management Framework (AI RMF 1.0)," National Institute of Standards and Technology, NIST AI 100-1, Jan. 2023.
[6] Z. Xi et al., "The Rise and Potential of Large Language Model Based Agents: A Survey," arXiv preprint arXiv:2309.07864, 2023.
[7] SPIFFE Project, "SPIFFE: Secure Production Identity Framework for Everyone," SPIFFE Specifications and Documentation.
[8] T. Lodderstedt, J. Bradley, A. Labunets, and D. Fett, "Best Current Practice for OAuth 2.0 Security," Internet Engineering Task Force, RFC 9700, BCP 240, Jan. 2025.
[9] OWASP Foundation, "OWASP Top 10 for Large Language Model Applications," 2025 Edition, 2025.