AI Access Management Framework

A Comprehensive Approach to AI Identity and Access Security

Contents

How Different Audiences Should Use This Framework:

- **Security Leadership**: Focus on Executive Summary, Business Impact, and Risk Management sections for strategic planning

- **Technical Implementers**: Deep dive into the seven framework pillars and implementation guidelines
- **Compliance Teams**: Emphasize monitoring, audit requirements, and governance controls
- **AI Practitioners**: Concentrate on secure development practices and operational security measures

Executive Summary

As organizations rapidly adopt AI technologies, a new class of security challenge has emerged: managing the identities and credentials for AI agents. AI agents and services authenticate using service accounts, API tokens, certificates, and other methods—expanding an often overlooked machine-to-machine attack surface that traditional, human-centric IAM solutions were never designed to address.

This AI Access Management Framework provides a structured, seven-pillar approach to identifying, managing, and securing all forms of AI-related access throughout its lifecycle. This framework addresses the unique challenges posed by AI systems' autonomous behavior, resource consumption patterns, and data access requirements while maintaining the operational flexibility required for rapid iteration and innovation.

Organizations implementing this framework can expect to achieve comprehensive visibility into their AI identity landscape, establish accountability and governance structures, implement appropriate technical controls, and maintain continuous security posture improvement - all while enabling safe AI adoption at scale.

Unique Security Challenges

AI-related Non-Human Identities (NHIs) represent a new frontier in cybersecurity. They are often:

- **Invisible** to traditional identity governance systems
- **Long-lived** and rarely updated
- **Highly privileged** with broad access to sensitive resources
- **Difficult to monitor** due to autonomous, non-deterministic and ephemeral behavior patterns
- **Vulnerable to novel attack vectors,** including adversarial inputs and model manipulation

These unique properties and attributes bring about specific challenges, making it tougher for organizations to safely, securely, and scalably adopt AI:

Autonomous Behavior at Scale

Unlike deterministic services, AI agents are non-deterministic, e.g. they dynamically choose their tools and access paths, often taking exploratory approaches that involve trial-and-error,, and have higher rates of failed requests compared to human users. At scale, this autonomy and non-determinism make it difficult to distinguish legitimate AI activity from potential security incidents, overwhelming traditional user and entity behavior analytics (UEBA) systems. The challenge is compounded by the fact that many agents are ephemeral, making it difficult to establish stable behavioral baselines.

Resource Exhaustion and Cost Overruns

AI workloads can consume significant computational and financial resources. Compromised AI non-human identities (NHIs) can lead to resource exhaustion attacks, where malicious actors use AI systems to generate excessive costs or consume limited resources for ulterior purposes, impacting business operations.

Data Access and Privacy Implications

Data is the lifeblood of AI, and therefore, AI systems often require access to large amounts of data to function effectively. This broad data access, combined with AI's ability to process and potentially transmit information externally, creates unique data loss prevention (DLP) challenges that traditional DLP solutions may not adequately address.

Shadow AI and Governance Gaps

The ease of AI service consumption has led to rapid, often ungoverned adoption across organizations. Employees may integrate AI tools and services without proper security review, creating shadow AI deployments with unmanaged identities and unclear data handling practices.

Adversarial Risks and External Influence

AI systems can be manipulated through adversarial inputs designed to alter their behavior or extract sensitive information. When combined with privileged agentic access, these attacks can have far-reaching consequences beyond traditional system compromises.

The Business Impact

The risks associated with inadequate AI identity governance are both diverse and severe, with real-world consequences that can devastate organizations:

Financial Loss

Unauthorized access to foundation models or agents can result in catastrophic financial damage. A single leaked API key for a foundation model can be exploited to generate tens of thousands of dollars in costs per day through excessive consumption. Once rate limits or quotas are exceeded, attackers may trigger service disruptions, degrading availability for legitimate users. Bad actors frequently trade these compromised keys on the dark web, fueling attacks such as LLMJacking. Similarly, they can manipulate compromised agents or tools to perform costly unauthorized actions, rapidly draining organizational resources.

Data Exposure and Privacy Breaches

Sensitive data faces exposure across all AI components. Compromised agents, data sources, or tools can leak confidential customer information, employee data, or proprietary business intelligence. These breaches not only undermine organizational security but also erode stakeholder trust and can trigger cascading security incidents.

Compliance Violations

Unauthorized access to AI agents, data sources, or tools frequently results in violations of critical regulatory requirements such as GDPR, HIPAA, CCPA, and industry-specific standards. These violations can trigger substantial legal penalties, regulatory fines, and loss of essential certifications, fundamentally impacting organizational credibility and market positioning.

Operational Continuity Disruptions

Malicious actors can severely disrupt AI-driven operations through various attack vectors. For foundation models, attackers can exhaust account-level, subscription-level, or zone-level rate limits and quotas, degrading performance and affecting other AI-dependent components. Compromised agents or tools can impair automated workflows, causing significant delays and inefficiencies across the entire organization.

Reputational Damage

Security incidents involving AI systems can have lasting reputational consequences. Breaches affecting data sources, tools, or agent behaviors can erode customer confidence and damage hard-earned organizational reputation. Publicized incidents of unauthorized AI access often lead to permanent loss of trust among stakeholders, partners, and customers.

Organizations without proper AI NHI governance face this interconnected web of risks that can compound rapidly, making comprehensive security frameworks not just beneficial but essential for safe AI adoption.

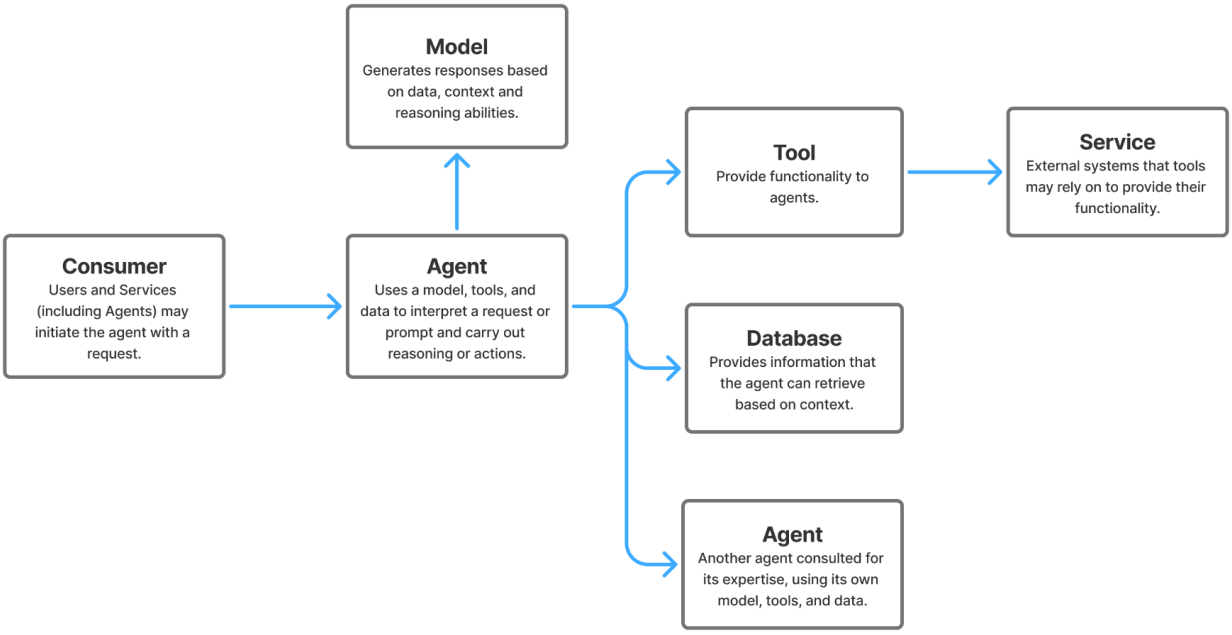
These challenges require a new approach to access security - one specifically designed for the unique characteristics of AI systems and their associated risks.

Access in the AI Ecosystem

Components of the AI Agent Ecosystem

Understanding the interconnected nature of AI systems is crucial for effective AI NHI security.

Each of the components introduced here can work separately from the rest of the system, can be hosted elsewhere and will need to be accessed in a secure manner.
The following describes the general case for a system:



Access Flow

Each blue edge in the graph represents an access requirement, where the arrow points from the authenticating unit to the resource that requires authentication.

Nodes can authenticate to other nodes in different ways - either by using **static, long-lived credentials** (e.g., access keys, API keys, client certificates) or by leveraging **short-lived, ephemeral credentials** (e.g.,. OAuth tokens, STS tokens, JWTs). These credentials, in turn, represent different identity types such as Roles, Service Accounts, Service Principals, Database Users, etc...

In relation to agents, use the terms *inbound* and *outbound access*:

Inbound Access refers to the Consumers of the agent, which can be human users, other agents, or standalone software. This flow is typically more controlled, as organizations can implement standard authentication mechanisms and access controls for known consumer types.

Outbound Access refers to resources utilized by agents. Those could be databases, APIs, tools, the internet and even other agents. Agent-to-Agent Authentication creates a complex scenario, as it requires the source agent to authenticate or authorize itself to the target agent, creating additional identity management complexity. These peer-to-peer connections require elevated privileges across diverse technology stacks and create cascading access relationships that are difficult to track and govern.

Each authentication flow generates its own set of NHIs with distinct lifecycle requirements, privilege levels, and security considerations. The challenge lies not just in managing individual identities, but in understanding how they interconnect to create potential attack paths across the entire AI ecosystem.

Understanding these access patterns is crucial for implementing appropriate security controls from our seven-pillar framework.

See real world implementation examples

The AI Access Management Framework

The framework consists of seven interconnected pillars that provide comprehensive coverage of AI access security concerns. Each pillar builds upon the others to create a robust security posture that evolves with your organization's AI adoption journey.

1. Discovery & Inventory

Identify and catalog all AI-related non-human identities and their contextual information.

1.1. Maintain Comprehensive Inventory of AI Related NHIs

Establish and maintain a comprehensive inventory of all AI-related non-human identities (NHIs) - for example, roles, service accounts, service principals, and database users. The inventory must document which agents and systems each NHI corresponds to.

The inventory must capture both inbound access used to interact with agents, and outbound NHIs used by agents to access external tools, resources, and data.

1.2. Document NHI Authentication Methods

For each NHI in the inventory, document the authentication methods employed, including authentication type (user access, delegated access, machine-to-machine access), secret storage location and security - when static credentials are used (e.g., secure key vault vs. hardcoded credentials) - OAuth configuration details including discovery endpoints where applicable, and principals who granted consent for delegated access scenarios.

1.3. Map NHI Consumer Relationships

For each NHI recorded in the inventory, identify and document its active usage status and potential, authorized, or known consumers (such as users, systems, or applications). Refer to Section 6.1 for requirements related to analyzing access patterns and frequency of use as part of ongoing monitoring.

1.4. Inventory AI System Resource Access

Document all resources accessible by AI-related NHIs, including but not limited to: (1) Database access (hosting location, sensitive data presence, access controls granularity, and read/write permissions); (2) SaaS integrations (access levels, service type, trust level and reputation); and (3) Internet access (general vs. restricted access and approved external destinations). See Section 4.2 for management of Data Exfilteration risks.

1.5. Detect and Manage Shadow AI Usage

Continuously identify unauthorized or unmanaged AI tools and services across environments using identity provider signals and endpoint telemetry. Implement automated detection for unapproved AI tools and establish remediation workflows for discovered shadow AI usage.

2. Ownership & Accountability

Ensure traceability and human responsibility for all AI NHIs.

2.1. Assign Ownership to All AI NHIs

Ensure every AI-related NHI is assigned to a responsible owner (individual or team) in accordance with organizational governance requirements. Ownership may be assigned directly to human owners or to CMDB items with designated human owners. For instance, each AI agent may have a clearly defined primary owner, and each associated identity may include an additional owner where appropriate, based on the resource type and scope.

2.2. List humans accountable for AI NHI changes

Track and retain logs of all humans who manage, or provision static credentials (such as tokens, API keys, certificates) used by AI NHIs or who modify AI NHI configurations and permissions.

2.3. Detect and Decommission Unused or Orphaned Identities

Establish processes for ongoing NHI management including regular access reviews and revalidation, automated detection of unused or orphaned identities, and periodic ownership verification. Enable automated decommissioning of unused identities based on contextual signals and customized workflows. See control 3.1 for initial provisioning security requirements and workflow customizations.

2.4. Revoke or Reassign Access During Offboarding and Personnel Changes

Upon employee offboarding or role changes, identify and remediate all AI-related access, including NHI ownership reassignment, credential revocation or rotation, and documentation of access transfer to new owners.

3. Credential Lifecycle & Hygiene

Manage AI NHI credentials throughout their entire lifecycle.

3.1. Enforce Secure Agent and NHI Provisioning

Require AI-related NHIs to be created and initially provisioned only through approved workflows that ensure scoped access based on least privilege principles, vault-backed credential storage, assigned ownership before activation, and documented business justification for the NHI request. Implement automated policy compliance checks during provisioning and enable workflow customization based on geographic, departmental, or risk profile requirements.

3.2. Prioritize Short-Lived Credentials

Use ephemeral or system-managed credentials where possible to reduce credential leakage risk and simplify lifecycle management.

3.3. Implement Secure Secret Storage

Store all AI NHI secrets and credentials in approved secure key vaults or secret management systems. Prohibit hardcoded credentials in configuration files or source code.

3.4. Rotate Long-Lived Credentials

When long-lived credentials are necessary, enforce regular rotation according to enterprise security policies and support automated rotation where technically feasible. Implement automated detection of credential rotation policy violations and establish escalation procedures for non-compliant credentials.

3.5. Prevent Human Credential Reuse

Detect and flag instances where human-issued credentials (session tokens, browser cookies) are inappropriately used by AI systems or automated processes.

4. Access Security & Risk Management

Implement technical access controls and manage AI-specific security risks.

4.1. Implement Resource Consumption Controls

For AI NHIs with access to consumable resources, implement usage controls including rate limiting mechanisms, quota enforcement, cost management boundaries, and LLM access limitations.

4.2. Manage Data Exfiltration Risks

For AI NHIs with sensitive data access and outbound connectivity, assess data exfiltration risks, evaluate potential for external influence through adversarial inputs, and implement data loss prevention controls appropriate for AI systems.

5. Vendor & Service Trust Management

Manage trust relationships with AI service providers and deployment models.

5.1. Classify AI Services and Vendors

Maintain a comprehensive catalog of all AI services and SaaS platforms that use NHIs to access organizational resources and data. Tag each AI service with comprehensive metadata including model vendor, hosting vendor, deployment model (SaaS, self-hosted, cloud-managed), service type classification, and vendor trust assessment.

5.2. Maintain Service Reputation Framework

Assign and regularly update reputation scores for AI services and SaaS platforms based on data handling practices, security incident history, geographic and regulatory compliance, and use reputation scores to prioritize and remediate access policies violations.

5.3. Enforce Geographic and Sovereignty Controls

Ensure AI services and LLMs operate only in pre-approved cloud regions that align with data residency and sovereignty requirements.

6. Monitoring & Threat Detection

Continuously monitor AI NHI usage patterns and behaviors.

6.1. Monitor AI Authentication Patterns

Log and analyze inbound and outbound access events, including authentication methods and frequency, consumer access patterns, unusual or anomalous authentication behavior, and cross-system access correlations. This ongoing monitoring complements the baseline consumer mapping established in control 1.3.

6.2. Track AI Resource Consumption

Monitor and log AI NHI outbound resource usage patterns including API call volumes and patterns, data access behaviors, resource consumption trends, and cost attribution and tracking.

6.3. Detect Behavioral Anomalies and Policy Violations

Implement automated detection of unusual AI NHI behaviors including access pattern deviations, unexpected resource consumption, suspicious data access or transfer activities, overprivileged AI NHI accounts, and violations of data access boundaries. Cross-reference behavior with known threat indicators and established organizational policies.

6.4. Maintain Immutable Audit Logs

Log all AI NHI-related activities in tamper-resistant audit logs, including access events, policy decisions, and lifecycle operations, with retention per organizational policy.

7. Risk Management & Continuous Improvement

Risk-based management and continuous improvement of AI NHI security.

7.1. Implement Risk-Based AI Access Management

Prioritize AI NHI security controls and management actions based on risk assessment considering access patterns and privileges, data sensitivity exposure, policy compliance status, and behavioral anomaly indicators.

7.2. Track Governance Maturity Metrics

Monitor organizational AI NHI security maturity through KPIs such as percentage of AI NHIs with assigned ownership, credential rotation compliance rates, policy coverage and exception rates, mean time to detect and remediate violations, and innovation enablement metrics (balance between security and operational flexibility).

7.3. Support Continuous Improvement and Remediation

Establish feedback mechanisms to assess policy effectiveness, identify gaps in coverage or control, measure impact on business operations and innovation, and drive iterative framework improvements. Support automated or semi-automated remediation actions for policy violations, including credential revocation, rotation, and ownership reassignment workflows.

Real-World Implementation Examples

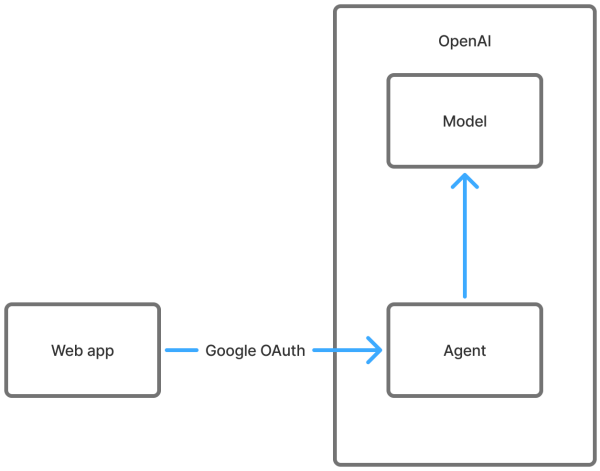
Example 1: Simple SaaS AI Usage

Scenario: ChatGPT Web App for Conversation Only

In this straightforward scenario, a user accesses ChatGPT through a web browser for basic conversation without external integrations.

Access chart:

Consumer	ChatGPT web app
Inbound to agent	Delegated OAuth (Google)
Agent to Models	Internal
Agent to Tools	—
Agent to DBs	—
Tools to Services	—



Framework Application:

- **Pillar 1** (Discovery): Identify the service account representing the app in GCP.
- **Pillar 2** (Ownership): Users or admins who provide Google OAuth consent to the application, along with vendor managers, are considered potential owners.
- **Pillar 3** (Credential Lifecycle): OAuth delegated access utilizing short-lived credentials.
- **Pillar 4** (Access Security): Corporate data is shared with OpenAI.
- **Pillar 5** (Vendor Trust): Evaluate OpenAI as trusted and sanctioned service provider.
- **Pillar 6** (Monitoring): Basic usage tracking and cost management.
- **Pillar 7** (Risk Management): Evaluating whether sensitive data-sharing risks with OpenAI outweigh chatbot benefits.

Key Security Considerations: visibility to usage, data residency with OpenAI.

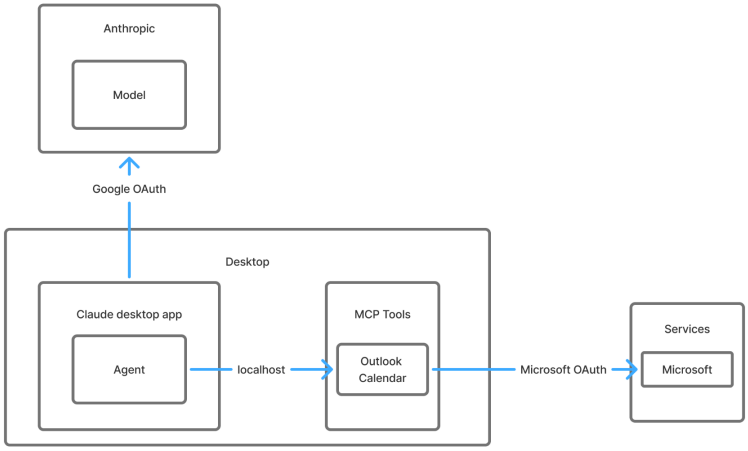
Example 2: Desktop AI with Local and Remote Components

Scenario: Claude Desktop App with MCP Integration

In this example a Claude desktop application interacts with the user's Outlook Calendar.

Access chart:

Consumer	Claude desktop app
Inbound to agent	Delegated OAuth (Google) to sign in to the app
Agent to Models	
Agent to Tools	Localhost with no authentication
Agent to DBs	—
Tools to Services	Delegated OAuth (Microsoft)



Framework Application:

- **Pillar 1** (Discovery): Scan for desktop apps, configured tools and credentials; scan for MCP tools and configured credentials; identify the Service Account representing the Claude Desktop in GCP and the MCP in Azure. Associate them all to the agent.
- **Pillar 2** (Ownership): Desktop users own their local apps and tools, along with any enterprise administrators.
- **Pillar 3** (Credential Lifecycle): OAuth delegated access utilizing short-lived credentials.
- **Pillar 4** (Access Security): Local MCP server security, limit internet outbound access to avoid calendar exfiltration.
- **Pillar 5** (Vendor Trust): Anthropic and Microsoft as service providers.
- **Pillar 6** (Monitoring): Desktop app usage, MCP server activity, calendar access patterns.
- **Pillar 7** (Risk Management): Evaluating sensitive calendar data-sharing risks with Anthropic. Evaluate potentially accidental calendar actions.

Key Security Considerations: local service security, OAuth scope management.

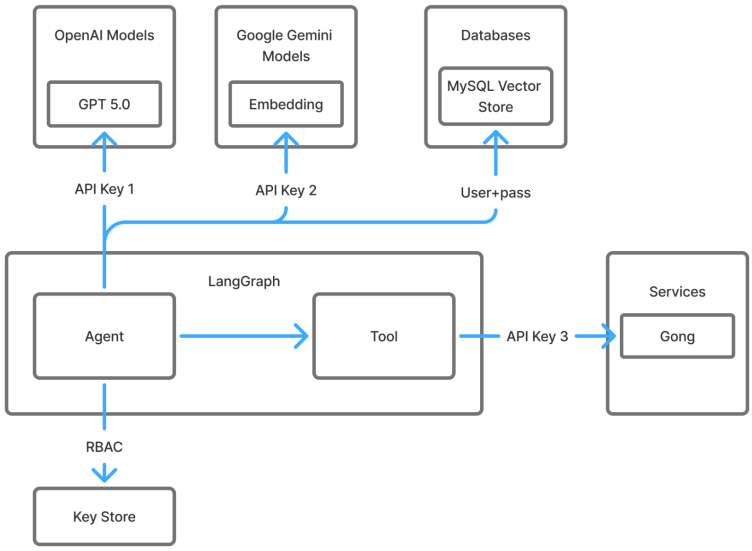
Example 3: Autonomous Agent with Multiple API Integrations

Scenario: LangGraph Agent for Sales Intelligence

This scenario shows a business intelligence autonomous agent that operates on a schedule. The agent accesses Gong to retrieve call transcripts, process them with OpenAI GPT model, then embeds them in an external vector database using a Google Gemini embedding model.

Access chart:

Consumer	—
Inbound to agent	—
Agent to Models	API key 1 with OpenAI API key 2 with Gemini
Agent to Tools	—
Agent to DBs	User+pass authentication to MySQL
Tools to Services	API key 3 to Gong
Other	Agent runs in the cloud and uses a form of managed identity, configured with RBAC to access a key store where it receives its keys and credentials.



Framework Application:

- **Pillar 1** (Discovery): Catalog all API keys (OpenAI, Google, Gong), database credentials, agent identity and its key store roles. Associate them all to the agent.
- **Pillar 2** (Ownership): Assign owners for each service integration and the agent.
- **Pillar 3** (Credential Lifecycle): Rotate API keys and database credentials, secure credential storage.
- **Pillar 4** (Access Security): Rate limiting for APIs - model providers and Gong. Proper database access management. Limit internet outbound access to avoid transcripts exfiltration.
- **Pillar 5** (Vendor Trust): Evaluate OpenAI, Google, and Gong as service providers.
- **Pillar 6** (Monitoring): Track API consumption, processing costs, data flow patterns, anomalous behavior.
- **Pillar 7** (Risk Management): Assess cumulative risk across the entire workflow.

Key Security Considerations: Long-lived API keys, automated execution risks, multi-vendor data flows, cost management across services.

Implementation Guidelines and Resources

Getting Started

Implementing the AI Access Management Framework requires a phased approach that balances immediate security improvements with long-term strategic goals. We recommend beginning with discovery and inventory (Pillar 1) to establish baseline visibility, followed by implementing ownership and accountability measures (Pillar 2) to establish governance foundations.

Cloud Platform Implementation Guides

Detailed implementation guides are available for major cloud platforms:

- **Azure Implementation Guide**: Comprehensive guidance for implementing AI access management using Azure Active Directory, Key Vault, and Azure Security Center
- **AWS Implementation Guide**: Step-by-step instructions leveraging IAM, Secrets Manager, and CloudTrail for AI access management
- **Google Cloud Implementation Guide**: Best practices using Cloud Identity, Secret Manager, and Cloud Security Command Center

Available Resources and Materials

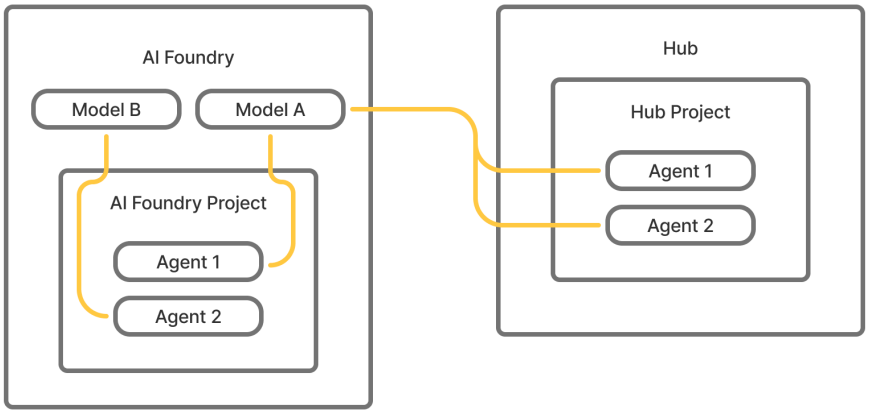
Cloud Platform Implementation Guides:

- **Azure**
- **AWS (TBD)**
- **Google Cloud (TBD)**

Azure AI Foundry - Background & Architecture

Overview

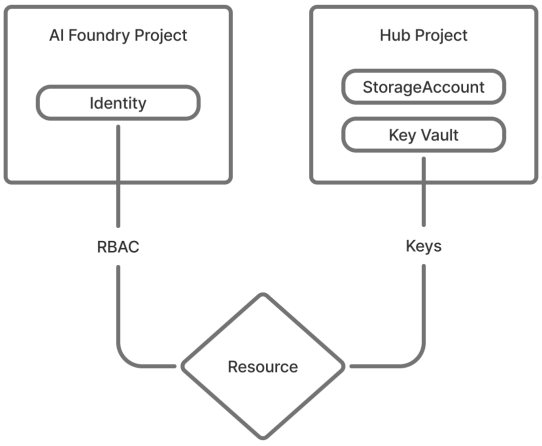
Azure AI Foundry is where language models are deployed.
Projects are where AI agents are created and configured to utilize these deployed models.
Projects can be organized under two different architectural patterns: directly under an AI Foundry resource, or within a Hub that connects to AI Foundry resources.



Agents of both types of projects utilize models deployed at the Foundry.

Project Architecture Patterns

Key Architectural Difference: Identity vs Key-Based Access



One key difference between project types lies in their access approach:

AI Foundry Projects use **managed identities and Role-based access controls (RBAC)** for secure access to Azure resources, reducing reliance on stored secrets and providing direct Entra ID integration.

Hub Projects use **key-based access** through secrets stored in associated KeyVaults, requiring explicit secret management but enabling resource sharing across multiple projects.

AI Foundry Projects

Created directly under an AI Foundry resource with built-in identity management capabilities. These projects can utilize their associated managed identities for RBAC-based access to external Azure resources.

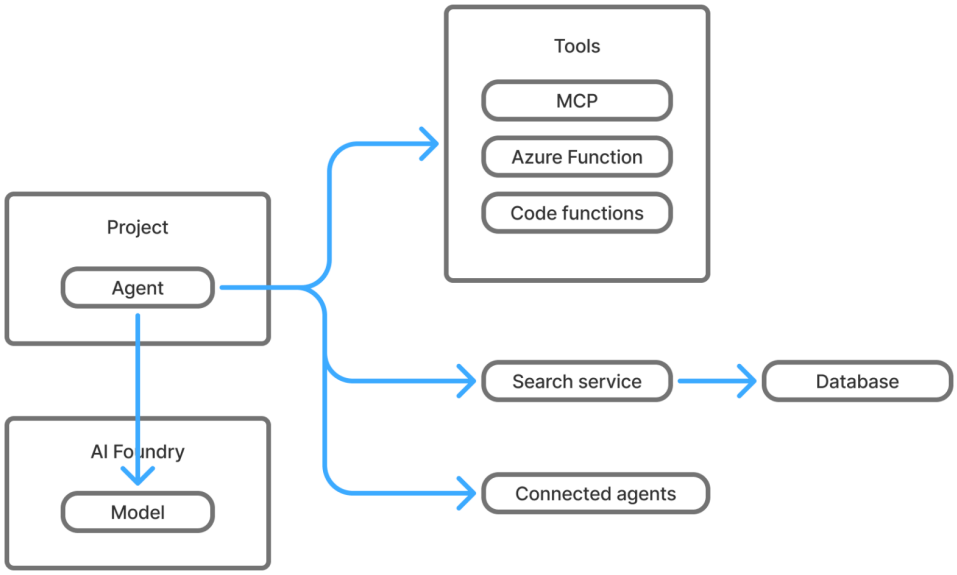
Hub Projects

Created within a Hub resource that provides shared infrastructure including Key Vault, StorageAccount, and compute resources. Hub projects can only access external resources using secrets stored in the associated Hub Key Vault. Hub projects are required for advanced features like **prompt flows**.

In addition, Hub projects are currently the only ones that can be set up with outbound network configuration to limit agent access to private resources.

Read more about Hub Project vs Foundry Project.

AI Agents



Agents are the primary AI entities created within projects and are configured with:

- **Model Assignment:** Each agent is assigned to a specific deployed model
- **Actions:** Integration with MCPs (Model Context Protocol), Azure Functions, or local code
- **Agent Associations:** Can be connected to other agents for complex workflows
- **Search Service Integration:** Can connect to AI Search services for contextual data retrieval, typically with API keys stored in associated Key Vaults

Permission Models

AI Foundry Project Permissions

- **IT Admin (Subscription Owner)** assigns “Azure AI Account Owner” role on resource groups to managers
- **Azure AI Account Owner** manages foundry infrastructure: deploys AI Foundry objects, deploys/decommissions AI models, applies guardrails (blocklist and content filtering) per deployed model, monitors model usage across all connected projects, and assigns “Azure AI Project Manager” roles
- **Azure AI Project Manager** creates projects under the AI Foundry and assigns "Azure AI User" roles for specific projects

- **Azure AI User** creates and configures agents at the project level

More information: Azure AI Foundry RBAC Setup

Hub Project Permissions

- **IT Admin (Owner)** assigns managers with “Contributor” role for creating new hubs or “Azure AI Developer” role for hub configuration
- **Contributor** manages hub infrastructure: compute resources, connected resources, and shared connections
- **Azure AI Developer** creates Projects and shared resources (at Hub level) and creates agents and utilizes connected resources (at Project level)

More information: Hub Project RBAC Setup

Core Components

AI Foundry Resource

The central resource for model deployment and management:

- **Model deployments:** Models selected from catalog, multiple deployments of the same model are possible (assigned with different names)
- **Quota management:** Per-deployment quota controls
- **Guardrails:** Blocklist and content filtering configured per deployed model
- **Monitoring:** Centralized usage monitoring for all connected projects

Hub Resource

Shared infrastructure platform, consisting of:

- One or more **Projects**
- **Key Vault** for secret management
- **StorageAccount** for data persistence
- **Compute resources** (required for non-serverless operations like prompt flow)
- **Connected Resources**
    - Model (deployed at either Azure OpenAI or Azure AI Foundry)
    - External resources accessible via Access Keys (stored in Key Vault)
- **Network Configuration** - In addition to inbound configuration, outbound can also be configured and limited to private networks and applied to all agents in related projects. See Network Configuration Guide.

Access Methods

API Key Authentication

Two API keys per AI Foundry resource with both keys visible in Azure Portal and only one visible in Foundry portal. Multiple API endpoints are available depending on specific API usage. Supports zero-downtime rotation workflows. See Key Rotation Guide.

RBAC Authentication

Role-based access leveraging Entra ID identities and the permission models outlined above.

External Tool Integration

Model Context Protocol (MCP)

Agents support MCP Tools requiring hostname or IP address specification. MCP servers can be deployed in public internet or Azure Virtual Networks. See MCP Integration Guide.

MCPs accessible through Azure Virtual Networks are reachable to Hub projects (requires configuring a proper outbound mode for the Hub).

Azure Functions

Functions can be invoked as agent tools through StorageAccount queue-based communication for input/output handling.

AI Search Service Integration

Core Functionality

The AI Search service provides vector database capabilities for contextual document embedding and retrieval:

- **Document embedding:** Converts records/documents from data sources into vector representations
- **Contextual proximity:** Documents embedded in close vector space proximity indicate contextual similarity
- **Similarity search:** Enables context-based retrieval of “most similar” items

Access Methods

- **RBAC:** Using built-in search roles

- **API Keys:** Direct endpoint access using service keys

Data Source Integration

Cloud Data Sources:

- Blob Storage, Cosmos DB, SQL Database, Azure Table Storage
- Authentication via managed identity or stored keys

External Data Sources:

- One of dozens of ADF connectors set as a Data source to an ADF pipeline
- A Search service set as a Linked service in the ADF pipeline
- The same Search service set as the pipeline's output destination

More information: ADF Search Connector

Agent Integration: Foundry agents can connect directly to Search services for context-based search capabilities.

Additional External Data Access

Beyond Search service integration, agents support direct access to files in storage, SharePoint, Bing, and other a continuously expanding list of external data sources.

Azure AI Foundry - Framework Controls Implementation

1. Discovery & Inventory (Controls 1.1-1.5)

Key Azure Resources to Inventory:

- Core resources: AI Foundry, AI Foundry Project, Hub Project, Hub, Search service
- Identity components: System and User assigned identities for all resources
- Credential stores: Key Vaults configured for Hub Projects and Hubs, or accessible to any other core resource.
- Access mechanisms: API Keys (AI Foundry, Search service), RBAC principals
- Tool identities: MCP Servers, Azure Functions, Search service data source keys

2. Ownership & Accountability (Controls 2.1-2.4)

Azure Implementation:

- Apply resource tags for ownership on all AI Foundry resources (AI Foundry, Projects, Hubs, Search services)
- Monitor human access through Azure Monitor resource logs and activity logs
- Implement automated detection of unused identities using dedicated monitoring tools
- Establish personnel change workflows that correlate ownership tags with HR systems for access transfer/revocation

3. Credential Lifecycle & Hygiene (Controls 3.1-3.5)

Best Practices for Azure AI Foundry:

- **Prefer identity-based access:** Use AI Foundry-based projects over Hub projects when possible for RBAC capabilities
- **Minimize secrets:** Configure connected resources and Search Service data sources with RBAC instead of API keys
- **Secure secret storage:** Store ADF linked service secrets in Key Vaults, maintain consumer API keys in approved secret management systems
- **Enable OAuth:** Use delegated OAuth for desktop applications instead of API keys where feasible
- **Implement key rotation:** Leverage AI Foundry's dual API key system for zero-downtime rotation workflows

4. Access Security & Risk Management (Controls 4.1-4.2)

Resource Consumption Controls:

- **Model quotas:** Set per-deployment quotas in AI Foundry Management Center
- **Usage monitoring:** Track token consumption through AI Foundry Management Center
- **Cost management:** Monitor expenses via Azure Portal Cost Analysis
- **Agent resource limits:** Control access through model deployment segregation

Data Exfiltration Risk Management:

- Assess risks for agents with sensitive data access and external connectivity
- Implement appropriate data loss prevention controls for AI-specific scenarios

5. Vendor & Service Trust Management (Controls 5.1-5.3)

Vendor Classification Sources:

- ADF linked services catalog
- Desktop applications using Azure AI Foundry access keys (usually done to configure LLM provider)
- Connected external services and data sources

Geographic Controls:

- Azure AI Foundry models are Microsoft-managed and deployed in your specified region
- Data residency aligns with existing Azure deployment regions

6. Monitoring & Behavioral Analysis (Controls 6.1-6.4)

Audit Log Configuration:

- **Activity logs:** 90-day default retention, export to Log Analytics/Storage/Event Hubs for extended retention
- **Sign-in logs:** 30-day Entra ID retention, forward to Log Analytics/SIEM for long-term storage
- **Resource logs:** Enable diagnostic logging per service, configure destination-specific retention
- **Immutable storage:** Implement for long-term audit log preservation in immutable storage accounts

Monitoring Requirements:

- Authentication pattern analysis using combined activity and sign-in logs

- Resource consumption tracking through service-specific diagnostic logs
- Behavioral anomaly detection via automated tooling
- Policy violation detection and alerting

7. Risk Management & Continuous Improvement (Controls 7.1-7.3)

Implementation Notes:

- Risk-based prioritization should consider access patterns, data sensitivity, compliance status, and behavioral indicators
- Establish KPIs for ownership assignment rates, credential rotation compliance, and policy coverage
- Implement automated remediation workflows for common violations
- Balance security controls with operational flexibility to support innovation

Glossary

Agent: An autonomous entity that performs tasks or interacts with other components within the AI ecosystem. Each agent is assigned with at least one model and can connect to other agents and access tools and services to carry out tasks.

Agent-to-agent connection: An interaction where one agent accesses or communicates with another agent, requiring the source agent to authenticate or authorize itself to the target agent.

AI Non-Human Identity (AI NHI): Digital credentials and authentication mechanisms used by AI systems to access resources, services, and data. This includes service accounts, API tokens, certificates, and other machine-to-machine authentication methods specifically associated with AI workloads.

Adversarial input: Malicious or carefully crafted input data designed to manipulate AI system behavior, extract sensitive information, or cause system failures.

Configuration Management Database (CMDB): A repository that stores information about IT assets and their relationships, used for tracking ownership and dependencies in IT environments.

Consumer: The entity that initiates actions or requests within the AI ecosystem. This can be a human user, another agent, or standalone software.

Database: Any storage or retrieval system that supports agent tasks, such as vector search engines, contextual search systems, or Retrieval-Augmented Generation (RAG) backends. Databases are typically accessed indirectly via tools or agents.

Data exfiltration: The unauthorized transfer of sensitive data from an organization's systems to external locations, often performed covertly by malicious actors.

Data Loss Prevention (DLP): Security technologies and processes designed to detect, monitor, and prevent unauthorized transmission of sensitive data.

Ephemeral credentials: Temporary authentication tokens with short lifespans, automatically generated and rotated to minimize security risks.

Inbound: In relation to agents, refers to consumers that authenticate and initiate requests or actions toward the agent.

Model: The intelligence component assigned to an agent that defines the agent’s behavior, capabilities, and reasoning logic. Models themselves do not act but serve as the decision-making foundation for agents.

Large Language Model (LLM): A specific type of model designed to process, understand, and generate human-like text. LLMs serve as the reasoning and communication engine for agents that interact through language.

LLMJacking: A type of attack where compromised API keys for language models are exploited to generate unauthorized costs and consume resources.

Machine-to-machine (M2M) authentication: Authentication processes that occur between automated systems without human intervention, typically using API keys, certificates, or service accounts.

OAuth: An authorization framework that enables applications to obtain limited access to user accounts or services without exposing user credentials.

Delegated OAuth: A specific use of OAuth where an application or agent is granted permission to act on behalf of a user, accessing only the resources and actions the user has authorized, without sharing the user's credentials.

Outbound: In relation to agents, refers to agent tools, databases, and other agents utilized by the agent to perform its functions.

Overprivileged account: An account or identity that has been granted more permissions or access rights than necessary to perform its intended function.

Service (in regard to the AI ecosystem presented): The backend system or provider that a tool interacts with, including SaaS platforms, internal services, or other persistent software systems. Tools provide agents controlled access to services.

Shadow AI: Unauthorized or unmanaged AI tools and services used within an organization without proper security review or governance oversight.

Tool (in regard to the AI ecosystem presented): A software component or interface that exposes functionality to agents, including MCP servers, APIs, or internal code modules. Tools act as intermediaries between agents and services.

User and Entity Behavior Analytics (UEBA): Security solutions that analyze patterns of user and entity behavior to detect anomalies and potential security threats.