NOMA

Secure Agentic Identity 101

Agentic identity environments

Employee Use AI
Locally uses end-user's identity via API keys or end-user delegated OAuth

Homegrown Apps
Developed using agentic SDKs, assuming native cloud identities and consuming tokens in cloud key vaults

Coding Agent
Using the developer's identity via API keys or developer's delegated OAuth

No-code/Low-code Agents
Connectors authenticated via API keys or end-user delegated OAuth

The new control plane for AI security

Agentic identity encompasses the collection of identities AI agents use to authenticate, authorize and act across tools, data and APIs often autonomously without human intervention.

The goal: Make every agent known, owned, least-privileged, context-aware, and auditable; then enforce at runtime.

Agentic identity includes:

AI Agent Identity

Non-human identities for tool calling
Long-lived tokens: Bearer tokens, API keys, service accounts
Short-lived tokens: OAuth app credentials
Native cloud identities: IAM roles, service principals

Delegated User Identity

Agents scoped to end user access
Browser session tokens
Short-lived tokens: Delegated OAuth tokens
User-provided credentials: Supplied in prompts/ configs

Agent Editors & Owners

Users authorized to edit, configure or clone the agent and its tools

Communicating Identities

Users or agents using A2A

Agentic identity vs. NHI

Agentic identity adds contextual execution linking every agent action to the delegating user's intent, permissions and runtime guardrails. Because agent behavior is goal-driven and non-deterministic, access must be evaluated dynamically, not pre-granted.

Traditional NHIs (API keys, service accounts) authenticate what a system is, not why it acts, operating without understanding user intent, context or runtime conditions.

	Agentic identity	Non-human identity
Scope	Cross domain	Platform-bound
Capabilities	Non-deterministic, context-driven	Static, predefined
Provenance	Runtime context + user intent + A2A chains	Credential issuer only
Accountability and traceability	Traceable to both agent and human principal	Identity-only (no user attribution)

Top agentic identity risks

Excessive Agency

Overprivileged or inherited permissions enabling destructive or sensitive operations beyond intended scope (e.g., data deletion, production writes).

Data Leakage

Compromised tokens, overprivileged agents or absent user-scoping creates unauthorized data exposure.

Identity Sprawl & Shadow Agents

Unregistered agents and orphaned NHIs operating without ownership, creating visibility gaps and uncontrolled access.

Misconfigurations

Maker's Identity
Shared agents operate under the creator's credentials. All users inherit the maker's privileges instead of executing with their own scoped permissions.

Ungoverned Long-Lived Tokens
Often provided in plaintext in MCP configs without rotation policies or expiration creating a persistent attack surface. Short-lived, vault-issued tokens used when possible.

Overprivileged Agents
Excessive capabilities far beyond operational requirements granted to agent directly or via A2A inheritance (violates least privilege).

Auditability Failure
Agent actions are indistinguishable from human users due to session impersonation or shared credentials, eliminating forensic traceability.

Agentic identity security checklist

- ☐ **Discover all AI agents** across environments, cataloging identity and federated roles.
- ☐ **Assign ownership** by designating a clear human owner for every agent to ensure accountability and lifecycle management.
- ☐ **Enforce traceability** by requiring every agent action to be authenticated and logged, capturing who (user or agent), what (action or tool), and where (target resource).
- ☐ **Prohibit shared or maker identities** by blocking use of developer or organization-wide credentials and enforcing unique, scoped identities per agent.
- ☐ **Audit and enforce least privilege** by blocking use of developer or organization-wide credentials and enforcing unique, scoped identities per agent.
- ☐ **Enforce contextual access** by validating at runtime that agent X can access resource Y only through its authorized identity Z and under proper user context.
- ☐ **Harden token hygiene** by identifying long-lived or hard-coded credentials and replacing them with short-lived, vault-issued credentials.
- ☐ **Govern the lifecycle** by standardizing agent creation, review and decommission to prevent orphaned identities or un-monitored access.