Non-Human Identity Management: Designing and Governing Machine Actors

By Prithvi Poreddy

© 2025 IDPro, Prithvi Poreddy

To comment on this article, please visit our GitHub repository and submit an issue.

Table of Contents

Abstract

As automation, cloud computing, and artificial intelligence redefine enterprise operations, the majority of digital actors are now non-human. Containers, workloads, pipelines, and AI agents perform most system interactions—yet identity and governance frameworks remain designed for people.

This paper presents a modern framework for managing Non-Human Identities (NHIs) across lifecycle, authentication, authorization, and observability. It examines how traditional models such as role-based access control and HR-driven lifecycles fail to manage entities that are ephemeral, autonomous, and system-generated.

The framework emphasizes lifecycle automation, policy-driven authorization, and continuous assurance, while incorporating emerging standards for workload identity and contextual access. It also addresses ownership, provenance, and accountability—essential components for governing AI-driven and autonomous systems at scale.

By treating NHIs as first-class identities—with clear lifecycle controls, policy enforcement, and traceable accountability—organizations can reduce risk, meet compliance requirements, and safely accelerate automation. The goal is not only to manage machine identities but to establish continuous, verifiable control across all digital actors in the enterprise.

1. Executive Summary

In modern enterprises, non-human actors (NHIs) now outnumber human users by orders of magnitude. While traditional machine accounts, such as legacy service accounts, were governed by change management processes, this article focuses on the challenges posed by modern, dynamic NHIs, including containers, microservices, APIs, CI/CD pipelines, and autonomous AI agents. These ephemeral actors perform the majority of authentication and authorization events inside digital ecosystems. Yet, most identity frameworks were designed for people, not code.

Traditional identity management models, such as joiner/mover/leaver flows tied to HR systems, static role-based access control, and periodic manual reviews, cannot scale to ephemeral or autonomous identities. This failure stems from the sheer volume and velocity of NHI requests: While a human may authenticate a few times a day, a microservice authenticates thousands of times per second. If these processes required human review or periodic recertification, the responsible owner would encounter overwhelming access requests or consent requirements, leading to severe approval fatigue. NHIs are created dynamically, operate with privileged access, and often vanish within seconds. If unmanaged, they become one of the most underestimated sources of risk in enterprise environments.

This shift requires rethinking identity security from first principles. Humans and non-humans must coexist under a unified model that provides equivalent rigor, automation, and accountability. Emerging standards for workload and service identity have made this possible, but their adoption demands new approaches to lifecycle design, policy enforcement, and ownership clarity.

The next phase of identity practice must treat NHIs as first-class citizens. This requires:

- Lifecycle governance adapted to ephemeral and dynamic actors
- Authentication and federation that go beyond passwords and SSO, embracing workload identity standards and short-lived credentials
- Policy-driven authorization that can evaluate runtime context, not just static role assignments
- Automated governance controls to continuously monitor, review, and certify NHI entitlements
- Clear ownership and accountability for every machine identity, tied back to a responsible human or system-of-record

The enterprise identity perimeter has shifted. Non-human identities are no longer an exception; they are the rule. Organizations that recognize and govern them effectively will not only reduce security risk but also unlock the ability to scale cloud, AI, and digital transformation initiatives with confidence.

2. What is a Non-Human Identity?

An identity represents any actor that can be authenticated, authorized, and governed within a system. Historically, identity was synonymous with a human, an employee, a contractor, or a customer interacting with business systems. But in modern enterprises, most authenticated sessions don't come from humans at all. They come from code.

A Non-Human Identity (NHI) is any digital actor (workload, service, agent, or device) that interacts with systems. While often initiated by human guidance (such as a prompt or configuration), the NHI performs operations and authenticates without a human driving the step-by-step execution or session.

Aligned with foundational identity governance principles, every NHI has three defining elements that must be managed for authentication, authorization, and auditability:

- An identifier (certificate, key, token, Decentralized Identifier [DID], or ID)
- Attributes describing its purpose, environment, and level of trust
- A lifecycle governing when it is created, used, rotated, and retired

Where human identities are tied to HR or workforce systems, non-humans emerge from automation pipelines, runtime orchestrators, or AI-driven systems. They can exist for milliseconds, operate autonomously, and vanish without warning, often leaving behind stale credentials or dangling privileges if not properly governed.

Categories of Non-Human Identities

In 2025, NHIs span several distinct but overlapping domains:

1. Infrastructure Identities

Workloads such as virtual machines, containers, and serverless functions are now provisioned and destroyed continuously. Each instance needs its own identity to authenticate with APIs, databases, and peer services. Cloud providers and orchestration platforms typically issue and manage these identities using ephemeral certificates or tokens tied to short runtime windows.

2. Service and Integration Accounts

These accounts enable automated systems and integrations to authenticate and perform operations across applications without requiring a human session. Examples include API keys, automation scripts, and long-lived system accounts used by background processes. Because they often bypass human login flows, these accounts accumulate privileges silently over time, making them a frequent source of access risk if not rotated, reviewed, or linked to an accountable owner.

3. Workload and Microservice Identities

Modern microservices rely on identity frameworks like SPIFFE/SPIRE or Kubernetes-native service accounts to establish trust between workloads. These identities are short-lived, automatically issued, and renewed using mutual TLS or token-based exchange. This approach replaces hardcoded secrets with dynamic workload identity, a major step forward in machine-to-machine trust.

4. Autonomous AI Agents

This is the newest and fastest-evolving class of NHIs. AI agents, powered by large language models or autonomous decision systems, can now act independently, fetching data, executing transactions, or even calling other APIs. Each agent instance effectively becomes a temporary identity with delegated authority. Without governance, these can

lead to accountability gaps ("who authorized what?") or identity drift, where an agent exceeds its intended scope.

These agents increasingly rely on Decentralized Identifiers (DIDs) and Verifiable Credentials (VCs), as defined by W3C standards, to establish self-sovereign, cryptographically-bound identity and delegate specific, granular authority across different trust domains.

Governance Challenge: Agent Impersonation

As agents gain autonomy, new attack surfaces emerge. Agent impersonation, where a malicious system mimics a legitimate agent, can bypass traditional security controls. Mitigations include strong binding between agents and their code origins, behavioral baselines to detect deviations from expected activity, and attested delegation ensuring agents operate only within defined scopes.

5. IoT and Edge Devices

While not new, IoT identities remain part of the NHI family. Devices authenticate using embedded certificates or keys and often lack lifecycle hygiene credentials, rarely rotate, and device ownership is difficult to trace at scale. In industrial and healthcare environments, these identities now form part of the extended enterprise perimeter.

Non-humans differ from human identities not in capability, but in governance origin. Humans are onboarded through HR-driven processes (with explicit oversight). Non-human entities emerge automatically from system code pipelines, orchestration layers, or runtime environments. Without structured lifecycle management, they proliferate invisibly, creating thousands of unseen, unowned identities that persist long after their purpose ends.

The future of identity management depends on recognizing NHIs as peers to human identities, not exceptions. Only by treating them as governed entities, with full lifecycle and accountability, can organizations manage security and compliance at cloud and AI scale.

3. Lifecycle Management of Non-Human Identities

Identity governance has long centered on the joiner–mover–leaver lifecycle, an HR-driven model that mirrors the employment journey of an individual. But non-humans don't join a company, change departments, or resign. Their existence is tied to system events, not HR events.

In modern environments, these identities are established when pipelines execute, workloads are spun up, or agents are instantiated. Their lifespans can be measured in minutes, sometimes milliseconds. Traditional lifecycle concepts, therefore, fail to apply. A human-centric process that takes days to provision or deprovision an identity can't keep pace with infrastructure that auto-scales every few seconds.

3.1 The Lifecycle Phases

Creation ("Birth")

Non-human identities emerge from automated workflows, but it's essential to distinguish their origin. Traditional NHIs (such as service accounts or integration keys) were often provisioned manually or semi-automatically through Change Management (CM) processes, resulting in long-lived identities that were typically recorded in a Configuration Management Database (CMDB).

In modern automation environments, however, identities are often generated dynamically from workflows such as CI/CD pipelines, container orchestration systems, or agent registration systems. Their creation is triggered by machine events, not human approval flows.

- A Kubernetes cluster issues service account tokens when a new pod launches
- An AI orchestration system dynamically spawns an agent identity to execute a task

Because these are machine-triggered events, governance must focus on policy-based registration (what systems are allowed to create identities and under what conditions).

Active Use ("Operation")

During their lifetime, these identities continuously authenticate and authorize, often using certificates, tokens, or mutual TLS.

Traditional NHIs (system accounts, integration keys) require periodic, scheduled, and often manual reviews. Modern, ephemeral NHIs operate at machine speed, requiring a completely different governance model. This operational phase introduces several challenges:

- Credential lifespan: Long-lived secrets create persistence risk.
- Privilege creep: Permissions can accumulate if not automatically right-sized.
- Context drift: The runtime conditions under which the identity operates (host, network, workload state) may change, invalidating original assumptions.

Effective governance for modern NHIs requires continuous monitoring and runtime validation, ensuring each identity still aligns with its intended context and risk posture, rather than slow, manual attestation cycles.

Deactivation ("Death")

When workloads or agents terminate, associated credentials must be revoked or expired immediately.

For Traditional NHIs, deactivation relies on slow, human-triggered change management processes, often resulting in orphaned accounts or forgotten keys that persist long after their intended purpose has ended.

In modern environments, containers terminate more quickly than IAM systems can respond. Stale keys, dormant tokens, or orphaned service accounts often persist indefinitely. Automated teardown policies and short-lived credentials (minutes, not days) are essential to reducing this residual risk surface. This automated expiration and revocation is critical because reliance on manual processes guarantees failure at scale.

Drift and Shadow Identities

Perhaps the most insidious phase is the unacknowledged one when an NHI continues to exist after its purpose ends. For traditional NHIs (system accounts), drift often means that static, long-lived credentials are forgotten in a CMDB and persist indefinitely. For modern, ephemeral workloads, drift results in shadow identities: old tokens or cloned service accounts that may never appear in any inventory, yet still retain functional access.

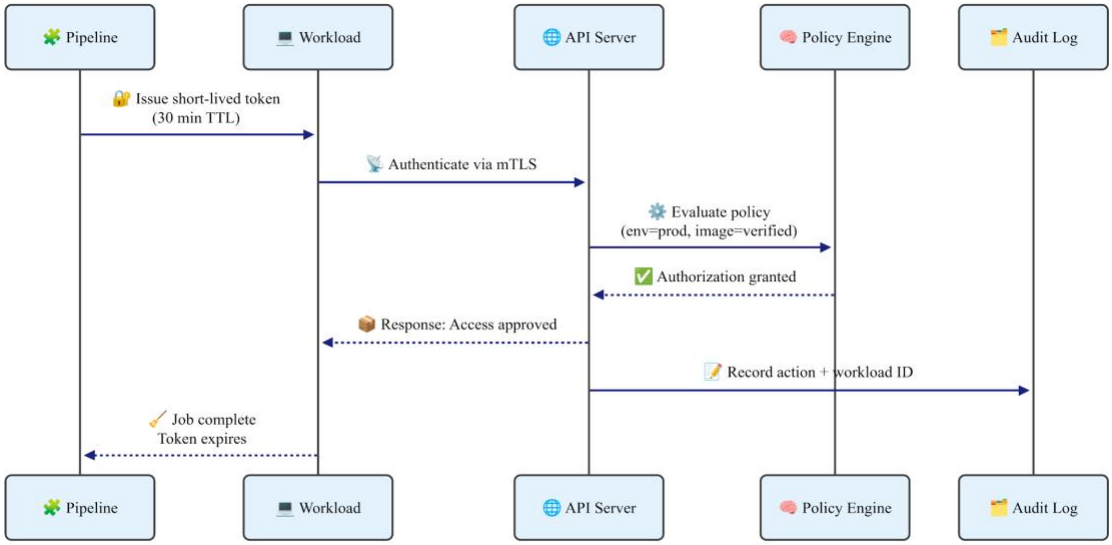
Detecting and eliminating drift requires continuous reconciliation between runtime inventories (e.g., Kubernetes, AWS IAM, CI/CD) and the identity governance system.

Example: CI/CD Pipeline Identity in Practice

Consider what happens when a Kubernetes deployment triggers:

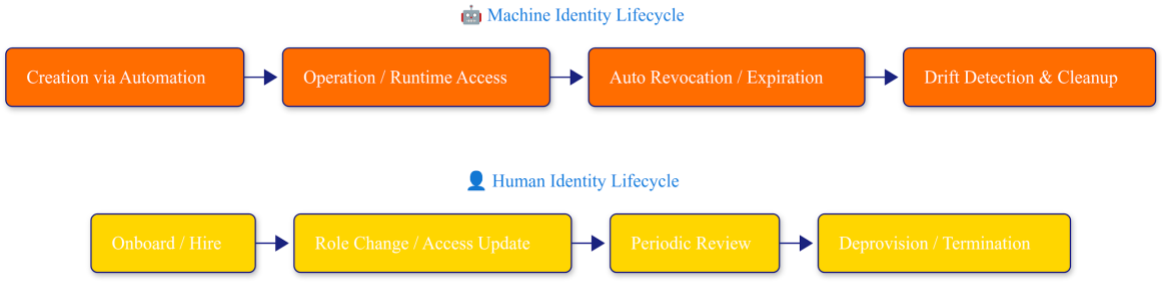
1. Creation: The cluster issues a service account token (30-minute TTL) bound to the pod specification and namespace
2. Authentication: The container presents this token via mTLS to authenticate to the database API
3. Authorization: Policy evaluates: `namespace=production AND image_signature=verified AND owner=platform-team` → ALLOW
4. Operation: All database queries are logged with the service account ID, pod metadata, and timestamp
5. Deactivation: Token expires automatically; pod termination triggers immediate credential cleanup
6. Ownership: Traced via deployment manifest to platform-team@company.com

This entire lifecycle completes in minutes far faster than any human-driven provisioning process could match. It demonstrates why NHI governance must operate at machine speed, with policy-driven automation replacing manual oversight.



3.2 Human vs. Non-Human Lifecycle

Lifecycle Phase	Human Identity	Non-Human Identity
Origin	Created via HR or onboarding system	Created dynamically by automation or orchestration
Operation	Continuous, interactive, session-based	Automated, event-driven, often ephemeral
Ownership	Explicitly tied to an employee or manager	Often lacks a clear owner; tied to system or code
Deactivation	Triggered by HR exit or role change	Triggered by system teardown or job completion
Common Failure Mode	Access not removed after departure	Orphaned secrets and long-lived credentials



3.3 Governance Implications

Manual processes and HR signals cannot govern machine identity lifecycles. This requires machine-speed enforcement, with the following design imperatives:

- Policy-based creation controls: Only trusted systems can mint new identities
- Ephemeral credentials: Default to short-lived certificates or tokens
- Automated reconciliation: Because NHIs are ephemeral and often leave behind shadow identities, continuous reconciliation is required to align the IAM inventory (what *should* exist) with the runtime inventory (what *actually* exists in orchestrators like Kubernetes or cloud IAM).
- Ownership linkage: Every NHI must have a responsible human or system-of-record anchor. This linkage is a foundational requirement across major security and compliance frameworks (e.g., NIST and ISO) to establish accountability for all system accounts and privileges, regardless of their ephemeral nature.

Without these measures, non-humans become unbounded by a growing sprawl of autonomous entities that operate outside formal control. The only sustainable path forward is automated, context-aware lifecycle governance that matches the pace of the systems it protects.

4. Authentication and Federation

Authentication is how an identity proves it is who (or what) it claims to be. For human identities, this typically involves something the person knows (a password), has (a device or token), or is (a biometric). Non-human identities, however, cannot perform any of these actions. Instead, they rely on cryptographic proof and system-issued credentials, which are issued and validated within a predefined Trust Framework (such as a centralized Certificate Authority or a federation boundary).

This difference fundamentally reshapes how authentication and federation are implemented across the enterprise. The methods used to verify human users (MFA, SSO, and browser redirects) do not scale to workloads or agents that operate without user interaction.

4.1 Authentication Mechanisms for NHIs

Certificates and Mutual TLS (mTLS)

Most modern NHI authentication hinges on asymmetric cryptography. Workloads present a private key–signed certificate to authenticate to other workloads or services. Mutual TLS (mTLS) ensures that both sides verify each other's identity before exchanging data.

- Advantage: Strong, verifiable, and fully automated once deployed
- Challenge: Certificate issuance, rotation, and revocation remain complex at scale, especially across hybrid environments

Workload Identity Standards

Frameworks such as SPIFFE/SVID and Kubernetes-native Service Accounts have introduced standardized ways for workloads to obtain and present short-lived identity credentials. SPIFFE IDs provide a uniform identity document for workloads across data centers and clouds, allowing federation between different trust domains. These frameworks effectively remove the need for long-lived secrets or hardcoded credentials.

- Advantage: Dynamic, short-lived credentials are automatically issued and renewed, drastically reducing the risk of orphaned or leaked long-lived secrets.
- Challenge: Requires specific infrastructure support (e.g., a SPIFFE/SPIRE deployment or cloud-native IAM configuration) and may add complexity when federating across diverse, legacy environments.

Token- and Key-Based Authentication

Legacy and SaaS integrations often still rely on API keys, OAuth2 client credentials, or shared secrets. These remain common due to simplicity but pose high risks if not rotated or scoped correctly. Transitioning these integrations toward short-lived tokens with bounded privileges is a core modernization goal for any NHI program.

- Advantage: Simple implementation and wide compatibility, making them easy to integrate with older applications and external SaaS platforms.
- Challenge: High risk due to long lifespan and static nature (especially API keys), making rotation and revocation a manual, high-friction process prone to error.

AI and Agent Authentication

For autonomous AI agents, the identity model is more fluid and relies on robust delegation. These agents often utilize profiles of OAuth 2.0 and OpenID Connect (OIDC) (such as OAuth 2.0 Token Exchange or Mutual TLS Client Authentication) to securely obtain and present scoped access tokens. Furthermore, as agents interact across independent domains, they increasingly leverage Decentralized Identifiers (DIDs) and Verifiable Credentials (VCs) to establish self-sovereign trust. Agents may need to authenticate both as themselves and on behalf of a parent system (e.g., an orchestration platform) or a responsible human owner. Each instance requires policy-defined delegation and binding, ensuring an agent cannot exceed its intended authority, and its actions remain attributable to a parent identity.

- Advantage: Supports fine-grained, verifiable delegation and enables secure interaction across independent, decentralized trust domains (using DIDs/VCs).
- Challenge: Introduces new complexity in policy tracing and auditability (who authorized what?), requiring advanced governance tools to manage fluid identity and

Non-human Credential Control Plane

At enterprise scale, organizations orchestrate these authentication mechanisms through a machine credential control plane, a service layer responsible for issuing, rotating, and validating credentials across clusters and clouds. This functions as the non-human equivalent of an identity provider, maintaining trust consistency between workload identity systems, certificate authorities, and policy engines.

- Advantage: Provides centralized visibility and automated lifecycle management (issuance, rotation, revocation) for all machine credentials across hybrid/multi-cloud environments.
- Challenge: Requires significant architectural commitment and potential integration with multiple proprietary cloud IAM systems and specialized credential vaults.
-

4.2 Federation of NHIs

Federation allows an identity from one domain to access resources in another without creating new credentials in each environment. For humans, this is implemented through SSO protocols like SAML or OIDC.

For non-human, federation operates at the infrastructure and trust-domain level, not at the browser or session level. In both cases, federation requires cryptographic trust anchors, not human redirection flows. Governance shifts from user experience to trust establishment, key distribution, and certificate policy management.

4.3 Governance Considerations

Managing NHI authentication is not simply a technical challenge; it's a governance challenge wrapped in cryptography. IAM teams must extend their purview to include:

- Credential lifecycle automation: auto-issuance, rotation, and expiration
- Cross-domain trust policies: defining which systems can federate and under what constraints
- Auditability: ensuring every certificate or token maps back to a legitimate workload, pipeline, or agent, and that every transaction or operation performed by that actor is chronologically recorded and attributable to that identity.

Without these controls, machine-to-machine trust devolves into unmanaged sprawl, a forest of secrets and certificates with no accountable owner.

5. Authorization and Access Control

If authentication answers "Who or what are you?", authorization answers "What are you allowed to do?"

For human identities, that question is typically governed by roles or groups defined around organizational structures. For Non-human identities, those assumptions

collapse. Non-humans appear and disappear in seconds, operate across multiple domains, and perform actions that are hard to predict in advance.

5.1 The Limits of RBAC

Role-Based Access Control (RBAC) assumes relatively static roles and slow-changing entitlements, which are valid for people, not machines.

Workloads and services are created dynamically by automation pipelines. Their function shifts with the environment or deployment stage. Fixed role assignments quickly become misaligned with reality, causing privilege sprawl and "role explosion."

RBAC still has value for coarse-grained control (for example, restricting namespaces or environments), but cannot express the fine-grained, runtime-aware decisions that NHIs require.

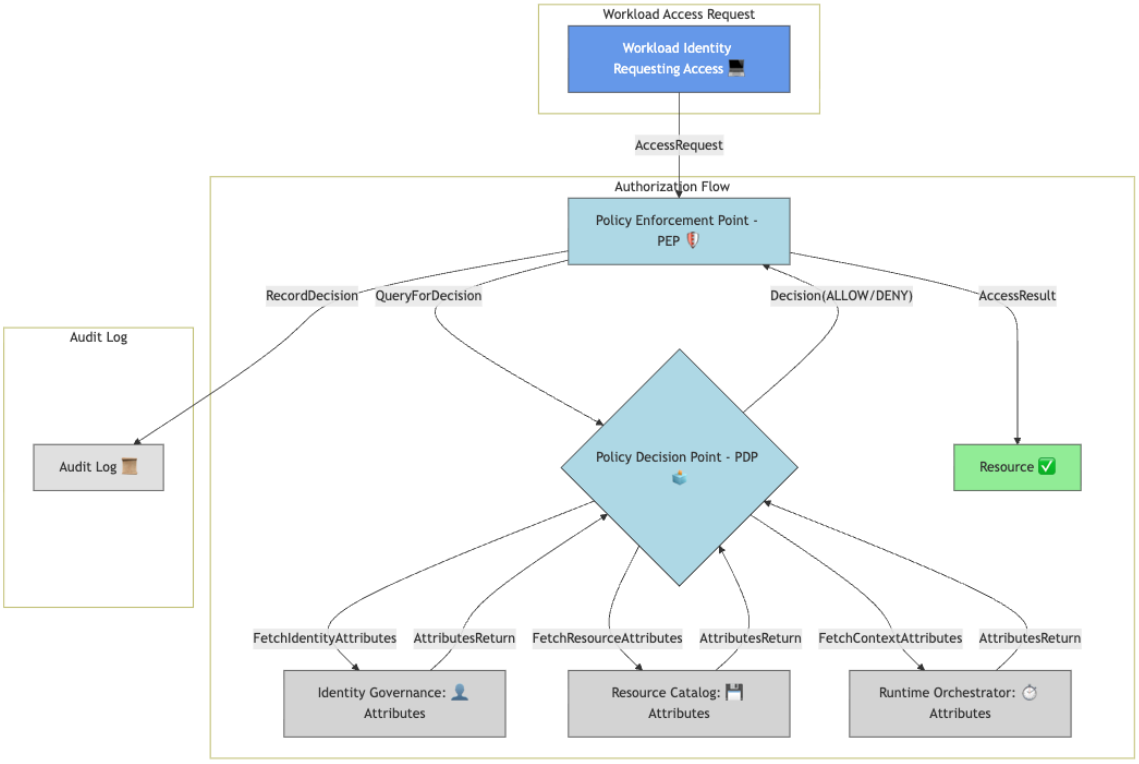
5.2 Modern Models for Non-human Authorization

To match the fluid behavior of non-human, modern IAM architectures increasingly rely on policy-based access control frameworks that evaluate attributes, relationships, and runtime context rather than static roles. A core tenet of this approach is Externalized Authorization Management (EAM), where the decision logic is decoupled from the application code itself. This separation allows policies to be managed, reviewed, and evolved independently and at machine speed, which is necessary for NHIs.

Key models include:

- Attribute-Based Access Control (ABAC): Evaluates identity and resource attributes to determine access based on the current context.
    - Example: Access is granted only if the identity's attributes state `environment = prod` AND the identity's attributes show `owner = devops`.
- Relationship-Based Access Control (ReBAC): Uses graph-style relationships between identities, services, and data objects to determine access.
    - Example (Delegated Authority): A data service can access the Customer Data Table (Resource) only if its immediate caller (Identity) is owned by the Billing Team and that team is linked to the Data Owner in the system graph.
- Policy-Based Access Control (PBAC): Separates authorization logic into declarative policies maintained independently of application code.
    - Example (Separation of Duties - SoD): A policy prevents a single CI/CD pipeline identity from both deploying application code (Task 1) and approving the production release (Task 2), ensuring that one machine identity cannot compromise the security chain.

For non-human entities, this approach aligns authorization with operational reality: access is granted based on the current context and relationships, not static assignment.



5.3 Delegated and Scoped Authorization

AI and autonomous agents extend this challenge further. They not only consume access but also initiate actions. Authorization, therefore, must account for both the agent's own identity and any delegated authority it exercises on behalf of a responsible system, workload, or originating human principle.

Scoped policies define those boundaries, but require advanced enforcement mechanisms to guarantee safe execution. For example, an AI agent may approve only low-value financial transactions, or a data-processing service may query sensitive fields but be unable to export them. Safe enforcement requires continuous runtime validation against policy constraints (e.g., verifying transaction value or data volume mid-action) and attested delegation models to prevent scope creep. Every action must remain traceable to a governing policy and to the originating principal that granted the authority.

5.4 Governance and Review

Authorization for non-human should be continuous and measurable, not event-based or quarterly. Effective programs emphasize:

- Automated entitlement review: Regularly verify that each NHI's privileges remain necessary
- Risk-weighted enforcement: Prioritize scrutiny of high-impact or high-privilege identities

- Policy observability: Record and analyze authorization decisions for audit and remediation

In this model, access governance evolves from periodic certification to continuous assurance, ensuring that machine-to-machine trust remains precise, contextual, and accountable.

6. Governance and Compliance

For two decades, identity governance has centered around human oversight of access requests, role certifications, and attestation campaigns designed for people and their entitlements. Non-Human Identities break nearly every assumption behind those models.

An engineer can attest to a colleague's access, but who attests to a container's? An HR system can trigger deprovisioning when an employee leaves, but what system triggers cleanup when an API key is no longer in use?

Traditional governance processes operate in manual time; non-human entities operate in machine time. Bridging that gap requires automation, continuous visibility, and context-aware policy enforcement.

6.1 Regulatory Context

Regulators and auditors have started catching up with the machine identity problem. While few frameworks explicitly name NHIs, most compliance frameworks now expect equivalent control for any entity with access to sensitive systems:

- SOX (Sarbanes–Oxley): Requires control over privileged accounts, including system and service accounts that can affect financial reporting systems. The key risk here is unowned Service Accounts allowing unauthorized changes to financial reporting systems.
- PCI DSS v4.0: Mandates periodic review of all accounts and secure key management, extending by necessity to machine credentials.
- Data Protection and Cybersecurity Mandates (GDPR/HIPAA/NIS2): Requirements around privileged access, data integrity, and accountability apply equally to machine accounts accessing sensitive data (Protected Health Information in the US, or general Personal Data/Critical Infrastructure data in the EU). These laws collectively mandate that NHI access to sensitive records must be strictly controlled, logged, and subject to audits showing minimal access (least privilege).
- NIST 800-53 and ISO 27001: Treat service accounts and automated processes as identities requiring governance and least-privilege controls.

In conclusion, these requirements demonstrate a universal principle: Every identity must be owned, monitored, and auditable, regardless of type. This extends the foundational principles of accountability and least privilege from human users directly to non-human actors.

6.2 Continuous Assurance

Traditional governance relies on scheduled review campaigns that validate access long after changes occur. Continuous assurance replaces this with automated, policy-driven validation that operates at runtime.

Core practices include:

- Real-time entitlement checks to detect excessive or unused privileges
- Automated ownership resolution to ensure every NHI maps to a responsible person or team
- Lifecycle-triggered reviews at creation, rotation, and decommissioning events
- Drift detection to identify credentials or accounts persisting beyond their intended lifespan

These controls maintain governance fidelity without human bottlenecks, turning compliance from a retrospective task into a proactive safeguard.

6.3 Measurable Accountability

Auditability for non-human requires verifiable evidence. This is because security and compliance frameworks (like SOX and GDPR) mandate non-repudiation: the ability to definitively prove which entity performed a specific action. Since machine identities operate autonomously, their actions must be perpetually traceable and linked to policy. Organizations must therefore be able to answer:

- Who owns this identity?
- When was its credential last rotated?
- What actions did it perform, and under what authority?

Machine-generated evidence logs, policy traces, and lifecycle data replaces manual attestations. Automated reporting pipelines provide provable assurance that governance policies are working as intended.

6.4 Governance by Design

Governance should be embedded directly into automation and deployment pipelines. Ownership metadata, policy APIs, and automated certification triggers can make compliance an inherent part of system design rather than an external process.

By integrating governance into creation workflows, organizations move from reactive oversight to built-in accountability, where every machine identity is governed from inception.

7. Observability and Auditability

Governance defines what should happen; observability reveals what actually does.

For Non-Human Identities, observability is not optional; it's the only way to maintain accountability for automated, ephemeral, and autonomous actions.

Human users leave visible trails through logins, approvals, or sessions. Non-human act programmatically, generating millions of API calls or process executions without direct oversight. Without identity-aware visibility, it becomes impossible to determine what was acted upon, when, or under whose authority.

7.1 Building Visibility

Effective observability begins by binding every machine action to an identity.

Key practices include:

- Identity-aware logging: Embed identifiers (certificates, tokens, or IDs) into system logs
- Session correlation: Group related actions into a single logical session for context
- Immutable audit pipelines: Store identity-linked events in tamper-evident logs for compliance and forensics
- Timestamp validation: Track creation and revocation times to confirm that no expired credentials executed actions

These controls transform raw event data into identity lineage, a verifiable record of who or what performed each operation.

7.2 From Logs to Accountability

Auditability requires more than collecting logs; it requires proving policy intent and execution.

Every recorded event should link back to:

- The governing policy that allowed the action
- The owner responsible for the identity
- The system or agent that executed it

Policy engines can produce decision traces showing why access was granted or denied, forming evidence for compliance and incident investigation.

7.3 Continuous Audit

Traditional auditing is retrospective; for non-human, it must be continuous.
Observability data should feed back into governance systems to enable:

- Real-time anomaly detection
- Automated revalidation when context or risk changes
- Continuous improvement of authorization and lifecycle policies

Audit becomes a living process, proving not just compliance but also active control.

8. Ownership and Accountability

Every identity must have an accountable owner. Ownership defines who is responsible for an identity's creation, behavior, and retirement. For Non-Human Identities, this principle is essential but often overlooked.

Human identities naturally map to individuals within HR systems. Non-human, however, originate from automation pipelines, orchestration tools, or AI frameworks and often lack direct linkage to a person or team. Without clear ownership, no one is accountable for their privileges, credentials, or security posture.

8.1 The Ownership Gap

In most organizations, NHI ownership is fragmented. Developers, DevOps teams, and security functions each manage parts of the lifecycle without unified accountability. This fragmentation leads to unowned or orphaned identities, stale credentials, and inconsistent governance.

8.2 Ownership Models

Organizations can assign responsibility using several complementary models:

Direct Ownership: Each identity is explicitly tied to a person, team, or service that created it.

Derived Ownership: Ownership is inherited from the system or pipeline that provisioned the identity.

Delegated Ownership: Higher-level systems, such as orchestration platforms or AI frameworks, assume responsibility for the identities they generate.

A blended model works best: derive ownership for scale, enforce direct accountability for oversight, and ensure delegated systems report lineage to human owners.

8.3 Ownership as a Governance Control

Ownership should function as an enforceable attribute, not a documentation field. Policies can enforce accountability by:

- Blocking identities without valid owner metadata
- Prioritizing reviews for identities whose owners have left the organization
- Routing certifications and alerts to assigned teams automatically

Embedding ownership in policy transforms accountability from a process to a control, ensuring no identity operates unowned.

8.4 Cultural Reinforcement

Technology establishes control, but culture sustains it. Embedding ownership tags in deployment manifests and pipeline templates normalizes accountability. Teams should internalize a simple principle:

"If you can create an identity, you own its risk."

This means the team or system that provisioned the NHI is directly responsible for setting its least-privilege scope, ensuring its credentials rotate, and guaranteeing its timely decommissioning.

Shared responsibility between engineering, IAM, and security teams ensures governance scales alongside automation.

Clear ownership connects every machine identity to a human. It is the anchor that ties automation back to oversight, ensuring that speed never comes at the expense of control.

9. Supply Chain and Provenance

For human identities, provenance is simple; their origin and history are recorded in HR systems. For Non-Human Identities, provenance extends across the digital supply chain: build pipelines, signed artifacts, workloads, and increasingly, AI models. Each stage introduces its own identities and dependencies, forming an interconnected trust chain.

Without verifiable provenance, a compromise anywhere in that chain (an altered build image, leaked key, or tampered model) undermines the integrity of every downstream identity.

9.1 Provenance as an Identity Attribute

Beyond "who has access," NHI governance must capture where an identity came from and how it was created.

Provenance attributes include:

- The system or pipeline that issued the credential
- Attestations verifying build or code integrity
- Dependency lineage across artifacts or models
- The runtime or environment in which the identity operates

Policies can then enforce trust conditions such as: "Allow deployment only if the workload originates from a verified pipeline with an approved attestation signature."

Treating provenance as an identity attribute links authentication to creation integrity.

9.2 Integrating Supply Chain Controls

Emerging supply-chain security frameworks already provide mechanisms for this verification. Concepts from initiatives such as SLSA, Sigstore, and software bills of materials (SBOMs) can be applied to identity. Together, they enable organizations to validate that every credential, artifact, or model is traceable and trusted before it gains access.

9.3 Cross-Domain Governance

Effective governance for NHIs relies on a defined Risk Governance Framework that establishes trust boundaries and liability between parties. This framework is often codified in agreements or contracts that define the Rules of Engagement between supply chain partners. These rules then drive the necessary technical enforcement across three integrated layers:

- Identity systems: Determine who or what can act.
- Supply-chain systems: Verify the integrity of what is executed.
- Policy systems: Define how trust is granted between them.

This cross-domain model ensures that access decisions consider both the actor and its origin. For example, even a valid certificate is rejected if it comes from an unverified build pipeline.

9.4 AI and Model Lineage

For AI and autonomous agents, provenance extends to model version, training source, and execution context. Recording this information provides accountability for agent decisions and supports explainability and audit requirements as AI systems act with increasing independence.

Provenance transforms identity from a static credential into a verifiable chain of creation. By embedding origin and lineage into governance, organizations extend zero-trust principles across the entire machine ecosystem from source code to running agent.

Zero Trust is a modern security strategy built on the principle of "never trust, always verify." It eliminates the traditional assumption that actors inside the network perimeter are safe. Instead, every access request, from any user, device, or workload, is treated as untrusted until its identity, context, and risk are dynamically verified.

For non-human identities, this framework traditionally focuses on continuous authentication (Section 4) and least-privileged access (Section 5). However, Provenance extends this idea by ensuring that the *origin* of the identity is also part of the verification check: an identity not only has to prove *who* it is, but also prove *how* it was built and *where* it came from (for example, from a signed, untampered CI/CD pipeline), thereby strengthening the very first step of the zero-trust chain.

10. Recommended Practices for Practitioners

Managing Non-Human Identities effectively requires both design discipline and operational consistency. The following practices summarize how to build scalable, secure, and auditable identity foundations for machine actors.

10.1 Treat NHIs as First-Class Identities

- Integrate non-human into the same identity inventory and governance processes as humans
- Apply least-privilege principles equally to workloads, APIs, and agents
- Establish unified visibility no separate or hidden credential stores

10.2 Automate the Lifecycle

- Authorize only trusted systems to create identities
- Issue short-lived credentials and enforce automatic rotation
- Continuously reconcile IAM records with runtime environments to detect orphaned or drifted identities

Lifecycle automation is not efficiency, it is the foundation of risk control at scale.

10.3 Modernize Authentication and Federation

- Adopt workload identity standards for mutual authentication
- Centralize credential management through automated issuance and rotation
- Define explicit trust boundaries for machine-to-machine federation

Authentication for non-human must be invisible, verifiable, and fully automated.

10.4 Shift to Policy-Driven Authorization

- Use ABAC, ReBAC, or PBAC models to enforce contextual and relationship-based policies
- Keep authorization logic external to application code for easier review and evolution
- Enforce attested delegation and runtime policy validation for agents, ensuring their actions remain strictly within the defined scope of the authority granted.

10.5 Embed Governance and Continuous Assurance

- Automate entitlement reviews and ownership validation
- Replace manual certification campaigns with policy-based triggers
- Treat compliance evidence as data collect, verify, and report automatically

10.6 Strengthen Observability and Ownership

- Include identity metadata in all logs and events
- Ensure every machine identity has a declared owner or team

- Block or alert on identities lacking ownership or valid purpose

10.7 Integrate Provenance and Supply Chain Controls

- Capture creation source, signing authority, and environment context for every identity
- Enforce policy checks that verify build and artifact integrity before granting access

Organizations that operationalize these practices move beyond account management toward continuous, automated governance where every human, service, and agent operates under verifiable identity, clear ownership, and real-time policy control.

11. Conclusion

Identity has always been the foundation of control defining who or what can act, under which conditions, and with what accountability. As automation and artificial intelligence transform enterprise systems, Non-Human Identities now represent the majority of digital actors. Managing them with frameworks built for people is no longer viable.

Traditional IAM approaches must evolve into identity security architectures that address machine speed, autonomy, and scale. This requires automated lifecycle management, policy-driven authorization, continuous observability, and enforceable ownership. Together, these form the basis of continuous governance, where identities are created, validated, and retired automatically while remaining fully accountable.

Identity is no longer limited to humans; it has become the control fabric of modern systems. Organizations that design for this reality will not only strengthen security and compliance but also enable safe, scalable automation across their digital ecosystems.

The future of identity lies in governing digital autonomy ensuring that every human, service, and agent operates under verifiable control.

Author Bio

Prithvi Poreddy is a Product Leader specializing in Identity Security, IAM, and AI-driven Governance. He works at the intersection of Identity, Risk, and Intelligent Automation, helping enterprises build secure and scalable identity foundations.

Prithvi has led IAM initiatives at organizations including Facebook, Lime, Deloitte, and World Bank, building scalable access models and advising C-suite teams on identity modernization. He is an active contributor to the Cloud Security Alliance's Identity Management working group and the MCP security group, where he focuses on AI agent security and authentication challenges.

His current focus is AI-driven identity governance, designing frameworks for autonomous agent identities, and aligning human and machine access models. He shares his thoughts on identity security through his blog at https://iam.ninja/ and

engages with the IAM community on LinkedIn. When he's not deep in security design, you'll find him playing tennis, writing about personal finance, stargazing, or playing tabletop board games.

References

IDPro Body of Knowledge

Williamson, G., Koot, A. & Lee, G., (2022) “Non-human Account Management (v4)”, IDPro Body of Knowledge 1(11). doi: https://doi.org/10.55621/idpro.52

McKee, M. K., (2021) “Introduction to Policy-Based Access Controls (v3)”, IDPro Body of Knowledge 1(12). doi: https://doi.org/10.55621/idpro.61

Standards and Frameworks

SPIFFE (Secure Production Identity Framework for Everyone). "SPIFFE Standards." Cloud Native Computing Foundation. https://spiffe.io/

National Institute of Standards and Technology (NIST). "NIST Special Publication 800-53: Security and Privacy Controls for Information Systems and Organizations." Revision 5, September 2020. https://doi.org/10.6028/NIST.SP.800-53r5

International Organization for Standardization. "ISO/IEC 27001:2022 Information Security Management." https://www.iso.org/standard/27001

World Wide Web Consortium (W3C). "Decentralized Identifiers (DIDs) v1.0." https://www.w3.org/TR/did-core/

Regulatory Frameworks

Payment Card Industry Security Standards Council. "PCI DSS Requirements and Testing Procedures Version 4.0." March 2022. https://www.pcisecuritystandards.org/

U.S. Department of Health and Human Services. "Health Insurance Portability and Accountability Act (HIPAA) Security Rule." 45 CFR Part 164. https://www.hhs.gov/hipaa/

European Parliament and Council of the European Union. "Regulation (EU) 2016/679 on the protection of natural persons with regard to the processing of personal data (General Data Protection Regulation - GDPR)." https://eur-lex.europa.eu/eli/reg/2016/679/oj

European Parliament and Council of the European Union. "Directive (EU) 2022/2555 on measures for a high common level of cybersecurity across the Union (NIS2)." https://eur-lex.europa.eu/eli/dir/2022/2555/oj

Cloud Provider Documentation

Amazon Web Services. "IAM Roles Anywhere." https://aws.amazon.com/iam/roles-anywhere/

Microsoft Azure. "What are managed identities for Azure resources?" https://learn.microsoft.com/en-us/azure/active-directory/managed-identities-azure-resources/

Google Cloud. "Workload Identity Federation." https://cloud.google.com/iam/docs/workload-identity-federation

Supply Chain Security

Linux Foundation. "Supply Chain Levels for Software Artifacts (SLSA)." https://slsa.dev/

Sigstore. "A New Standard for Signing, Verifying, and Protecting Software." https://www.sigstore.dev/