OASIS

The next phase of PAM: Governing non-human identity at scale

Complete your PAM strategy with visibility, control, and lifecycle enforcement

Privileged Access Management (PAM) protects high-risk credentials, but it was never designed to discover every machine identity consuming them or to govern how those identities are created, used, and retired across your environment. This paper outlines what it takes to **complete your PAM program** with full visibility into non-human identities, fewer vault gaps, and enforceable ownership and rotation policies. The goal is not to replace PAM, it is to extend its value into the identity landscape it was never designed to reach.

Learn

1.

How to **securely manage a multi-vault environment with centralized oversight and consistent controls**, without forcing consolidation or disrupting existing systems.

2.

Why **PAM platforms designed for human-privileged access cannot effectively discover, contextualize, or manage the lifecycle of non-human identities** at enterprise scale.

3.

What a **modern approach to NHI governance looks like** when it works with your existing PAM and vault investments to reduce costs and accelerate your success.

OASIS

PAM's role, and its structural gap

Privileged Access Management is a cornerstone of enterprise security. It protects the high-value credentials that control infrastructure, production systems, and sensitive data. For more than two decades, PAM has reduced human privilege risk and strengthened access control across the enterprise.

What has expanded dramatically is the population of identities using those credentials.

In modern enterprises, most privileged access paths are no longer traversed by people. Applications, cloud services, CI/CD pipelines, APIs, containers, SaaS integrations, and emerging AI-driven agents continuously create and use non-human identities at machine speed and cloud scale. These identities proliferate automatically, operate across distributed environments, and often exist outside centralized vaulting programs.

The result is not a failure of PAM, but **a structural gap between what PAM was built to manage and how privileged access is now created and consumed.**

Organizations today face four core challenges in the PAM program:

1

Limited visibility into unmanaged non-human identities and credentials that should be protected, but never enter the vault

2

Fragmentation and compliance/audit complexities when different teams use multiple vaults and secrets systems in hybrid and cloud environments

3

A missing lifecycle governance model for non-human identities where credential storage exists, but ownership, policy enforcement, and decommissioning controls do not

4

Rapid growth of AI agents and autonomous systems, which introduce dynamic, self-operating identities that no existing vault or PAM model was designed to govern

OASIS

Why Non-Human Identities outgrow the PAM model

PAM is excellent at what it was designed to do: governing privileged human accounts through credential vaulting, session management, and access approvals. PAM has always governed certain non-human accounts, such as shared admin accounts, service accounts, and database credentials. **What has changed is the scale, distribution, and automation of machine identities across modern infrastructure.**

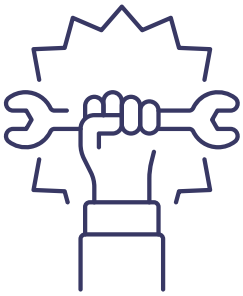
But even where PAM and vaulting are deployed, a subtler gap persists.

Identity and security teams often mistake "secrets management" for "access management." They assume that because a secret is vaulted, the access it grants is governed. But vaulting solves storage, not context. It tells you where a secret lives, not why it exists, what consumes it, or when it should expire. PAM layers critical controls on top of those vaulted credentials: session management, access approvals, and audit logging. These controls were architected for human-scale privileged access, where the identity is known, the consumer is singular, and the lifecycle is managed through HR and directory systems. Non-human identities operate outside those assumptions.

	Human Privileged Access	Non-Human Identity Access
Scale	Hundreds to thousands	Tens of thousands to millions
Source of truth	HR / Active Directory	None, decentralized creation
Credential type	Password + MFA	Secrets, tokens, certificates; no MFA
Lifecycle	Joiner → mover → leaver	Created ad hoc, often never decommissioned
Vault coverage	High, mandated, and enforced	Partial: multi-vault, many ungoverned
Rotation	Straightforward, single consumer	Risky without dependency mapping
Ownership	Clear: the human	Unclear: who created it? Who uses it now? Who is accountable when it's compromised?

Scale Mismatch and Manual Effort

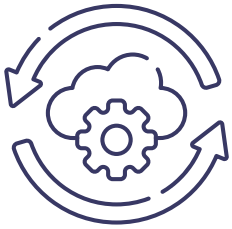
PAM was designed to manage hundreds or thousands of privileged human accounts. Enterprises now have tens of thousands to millions of NHIs. Onboarding each one into a PAM vault (with per-user licensing, manual configuration, and approval workflows) does not scale. And attempting to force this model introduces the very friction that drives teams to create identities outside governance entirely.



Even in traditional environments, service accounts were often the most operationally burdensome part of a PAM program. Manually creating accounts, onboarding them into the vault, configuring rotation policies, and coordinating safe password changes required careful planning to avoid outages. That **model was tolerable when service accounts were relatively static and limited in number. It becomes unworkable in cloud-native environments where identities are created dynamically by pipelines and infrastructure code.**

Lack of an Authoritative Source

PAM for human identities works because it integrates with HR systems and Active Directory as authoritative sources of identity, driving permission management and context from the beginning of the identity lifecycle. **NHIs** have no equivalent. They are **created by developers, automation tools, cloud platforms, and third-party integrations, often without any centralized record of their existence.**



This absence is foundational. Organizations cannot govern what they cannot enumerate, and they cannot enumerate NHIs because no system of record exists for them. Where human identity governance starts with a directory, non-human identity governance starts with a gap.

Context Blindness

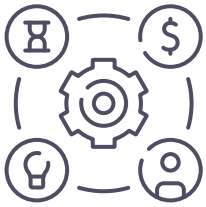
PAM sees credentials stored in the vault. It does not see the full chain of relationships that give those credentials meaning: which application consumes the secret, what the identity does at runtime, what systems depend on it, or what would break if the credential were rotated or revoked.

This context gap is not a minor inconvenience. It is the reason most organizations are afraid to rotate secrets. A PAM vault can execute a password change. But if it does not know every system, application, or pipeline that uses that credential, rotating it causes outages. This fear is why secrets become long-lived, and long-lived secrets are among the most exploited attack vectors in enterprise environments.

No NHI Lifecycle Management

PAM manages credentials. It does not manage identities. There is an important distinction between the two.

A credential is one component of a non-human identity's existence. But governing NHIs requires managing the full lifecycle: provisioning with appropriate access and ownership, monitoring usage patterns over time, rotating credentials safely, and decommissioning identities when they are no longer needed. PAM was never designed to own this lifecycle.



Limited Multi-Vault Visibility

PAM platforms govern what is inside their vault. They have **no visibility into secrets stored in cloud-native managers, CI/CD systems, SaaS platform credential stores, or environment variables**.

In a multi-vault world, PAM governs a fraction of the total secret landscape, and has no mechanism to even quantify the fraction it is missing.

The Multi-Vault Reality: How We Got Here

Enterprises rarely choose to be multi-vault. They arrive there through rational, independent decisions. Cloud-native integrations, developer workflows, cost considerations, and operational autonomy all drive teams toward the tools that fit their environments. When organizations attempt to mandate a single vault, friction increases. Migration disrupts automation. Licensing costs rise. Developer velocity slows. The predictable outcome is partial adoption of the mandated vault and continued secret storage outside governance controls.

Multi-vault sprawl therefore reveals a deeper limitation: **vault-centric governance cannot deliver unified identity visibility, consistent policy enforcement, or lifecycle management across distributed systems.**

The path is remarkably consistent across organizations. A cloud team adopts AWS Secrets Manager because it's native to their environment, tightly integrated, and has negligible incremental cost. An Azure team uses Azure Key Vault for the same reasons. A central security team mandates the use of CyberArk or Delinea for on-premises systems and regulated workloads. DevOops teams standardize on HashiCorp Vault for dynamic secrets in CI/CD pipelines. SaaS platforms (Snowflake, Databricks, GitHub) each maintain their own credential stores. And when an acquisition closes, the acquired company brings its own vault stack entirely.

Each of these decisions is defensible in isolation. Each team chose the tool that made the most sense for their context, their workloads, and their operational cadence. The problem is what happens at the organizational level.

Why Consolidation Fails

When security and identity teams recognize the sprawl, the instinct is to mandate: pick one vault, migrate everything into it. In practice, this rarely works.

Cloud-native vaults are deeply integrated with their ecosystems. Moving secrets out of AWS Secrets Manager breaks IAM role assumptions and deployment automations that teams have built over the years. Enterprise PAM vaults add per-secret licensing costs and operational overhead, discouraging broad adoption.

Developers will route around friction, not through it. And the vault that central security mandates is almost never the vault that developers prefer to use.

The result is predictable: partial adoption of the "official" vault, with the majority of secrets continuing to live in cloud-native stores, CI/CD pipeline configurations, environment variables, and, in too many cases, hardcoded in application code.

The Adoption Trap

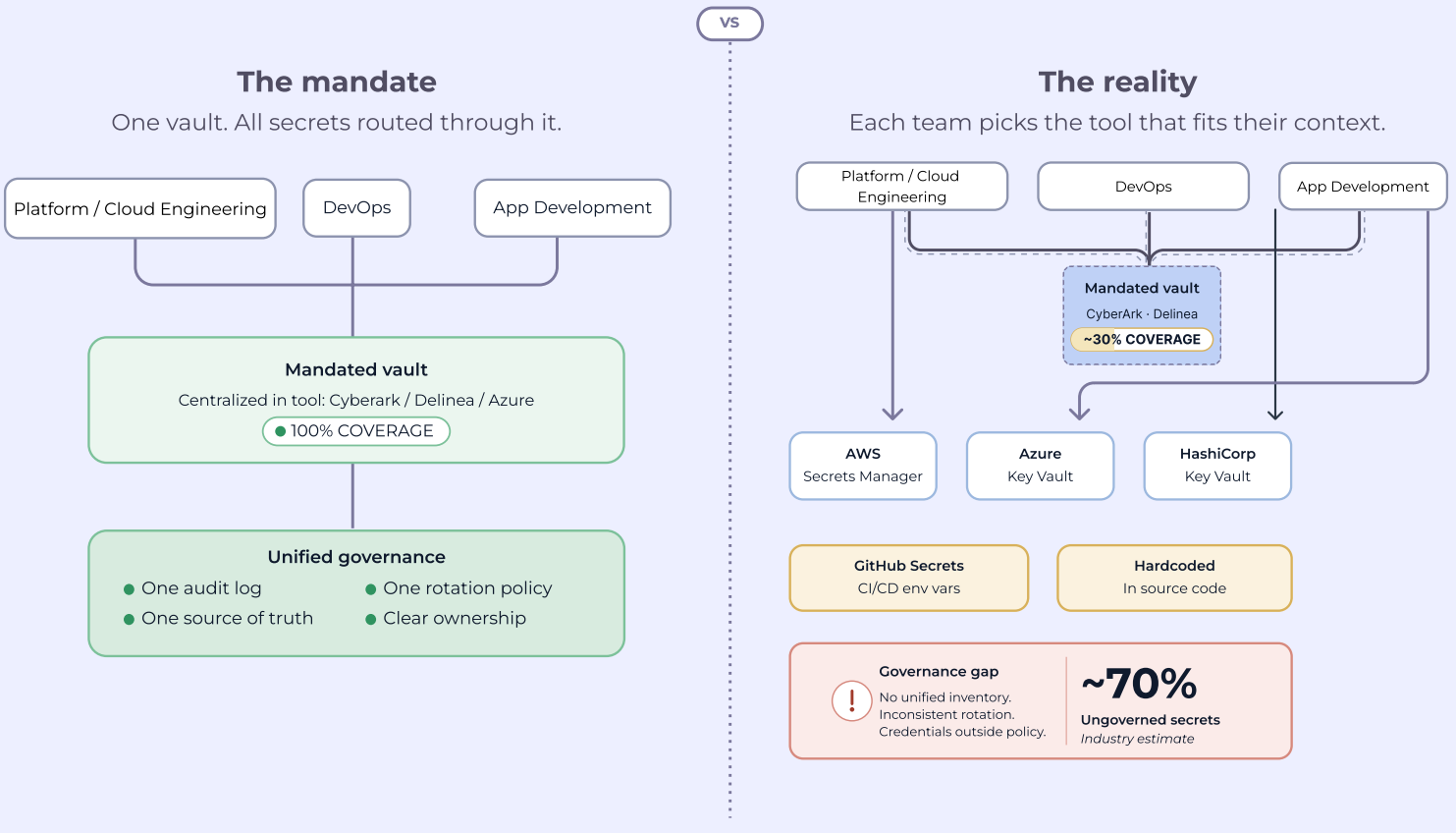
GitGuardian's 2025 State of Secrets Sprawl report found that even 5.1% of repositories already using a secrets manager still leaked secrets. Industry research consistently shows that the vast majority of organizations struggle with secrets sprawl, credentials scattered across code repositories, configuration files, and deployment scripts. **Mandating a single vault without addressing developer workflows and cloud-native integration patterns does not eliminate sprawl. It simply makes the ungoverned portion invisible.**

The Consequence: Vault Sprawl Equals a Governance Gap

When secrets are distributed across five, ten, or more storage systems, security teams lose the ability to answer fundamental questions: How many secrets do we have? Where do they live? Are they being rotated? Who owns them? Which ones are still in use?

Rotation policies become inconsistent or nonexistent across vaults. Secrets outside the mandated vault are effectively ungoverned. And no single tool in the stack can provide a unified view of the organization's total secret landscape.

This is not a failure of discipline. It is the natural outcome of building modern infrastructure across multiple clouds and platforms, each with its own native credential management approach.



The Cost of Inaction

The gap between what PAM governs and what NHIs actually require is not abstract. It produces specific, measurable consequences.

Operational Risk

Unmonitored secrets expire without warning, causing production outages that can affect critical business systems for hours or days. When secrets are scattered across multiple vaults without dependency mapping, teams cannot rotate proactively; they discover the problem only after services fail.

Security Risk

Long-lived, overprivileged, unmonitored non-human identities have become a primary attack vector. The OWASP Non-Human Identities Top 10 for 2025 identifies the most critical risks: improper offboarding of NHIs that remain active long after they should have been decommissioned, secret leakage into code repositories and collaboration tools, overprivileged identities with far more access than they need, and long-lived secrets that give attackers an unlimited window of exploitation.

The data reinforces the urgency. The CSA's State of NHI Security survey found that **only 15% of organizations feel highly confident in their ability to prevent NHI-related attacks.**

Compliance Risk

Auditors increasingly ask for evidence of NHI governance. PAM audit logs that only cover a fraction of the organization's secrets will not satisfy SOC 2, HIPAA, PCI-DSS, or GDPR requirements. The question auditors ask is not *do you have a vault?* but ***can you demonstrate governance over all privileged access, including machine identities?***

The Workaround Paradox

Perhaps the most counterintuitive risk is that heavy-handed vault mandates actively worsen the problem. When security teams impose rigid vault requirements without accommodating developer workflows, teams route around the friction. They hardcode secrets. They store credentials in CI/CD variables. They share tokens in Slack messages. GitGuardian's 2025 State of Secrets Sprawl report found 23.8 million secrets leaked on public GitHub repositories in 2024, a 25% year-over-year increase. And private repositories contained secrets at an eightfold rate compared to public ones. **Vault mandates that ignoring developer experience does not reduce sprawl. They create it.**

OASIS

What Modern NHI Governance Requires, on Top of PAM

PAM is a necessary layer. But it is not sufficient for securing the non-human identity attack surface.
Governing non-human identities at enterprise scale requires additional capabilities that sit above individual vaults and PAM platforms, providing the unified visibility, context, and lifecycle automation that no single vault can deliver on its own.

Accept Multi-Vault as the Reality

The starting point is acknowledging that **multi-vault is the permanent condition, not a temporary state to be resolved through migration**. A modern governance approach builds a policy layer on top of existing vaults rather than fighting for consolidation. This preserves team autonomy, respects cloud-native integrations, and avoids the adoption failures that consolidation mandates consistently produce.

Discover NHIs Everywhere

Governance begins with complete visibility. This means discovering every non-human identity and every secret, not just what is in the vault, but across all clouds, SaaS platforms, on-premises environments, developer tooling, and CI/CD pipelines. Discovery must be continuous and agentless, capable of surfacing identities that no team knows about.

Map the Full Context Chain

This is the capability that separates governance from inventory.

Effective governance requires understanding the complete chain of relationships: which consumer (application, service, pipeline) uses which secret (key, token, certificate) to authenticate as which identity (service account, role, principal) to access which resource (API, database, cloud service). Without this context, teams cannot rotate safely, review meaningfully, or decommission confidently.

Without this context, every downstream governance action is either impossible or dangerous. Teams cannot rotate safely because they do not know what will break. They cannot review access meaningfully because they do not know what the identity actually does. They cannot decommission confidently because they cannot confirm whether the identity is truly unused or simply not recently observed.

Discovery tells you what exists. Context mapping tells you what it means. The difference between the two is the difference between an inventory and a governance program.

Organizations that have invested in discovery without context mapping consistently report the same outcome: they know they have a problem, they can quantify its scale, but they still cannot act on it. Visibility without context is awareness without the ability to govern.

Enforce Consistent Policies Across All Vaults

Rotation cadence, ownership requirements, least-privilege enforcement, expiration limits, these policies must be consistent regardless of which vault stores the credential. A secret in Azure Key Vault should be governed with the same rigor as a secret in CyberArk. This requires a governance layer that operates across vaults, not within any single one.

Enable Safe, Dependency-Aware Lifecycle Operations

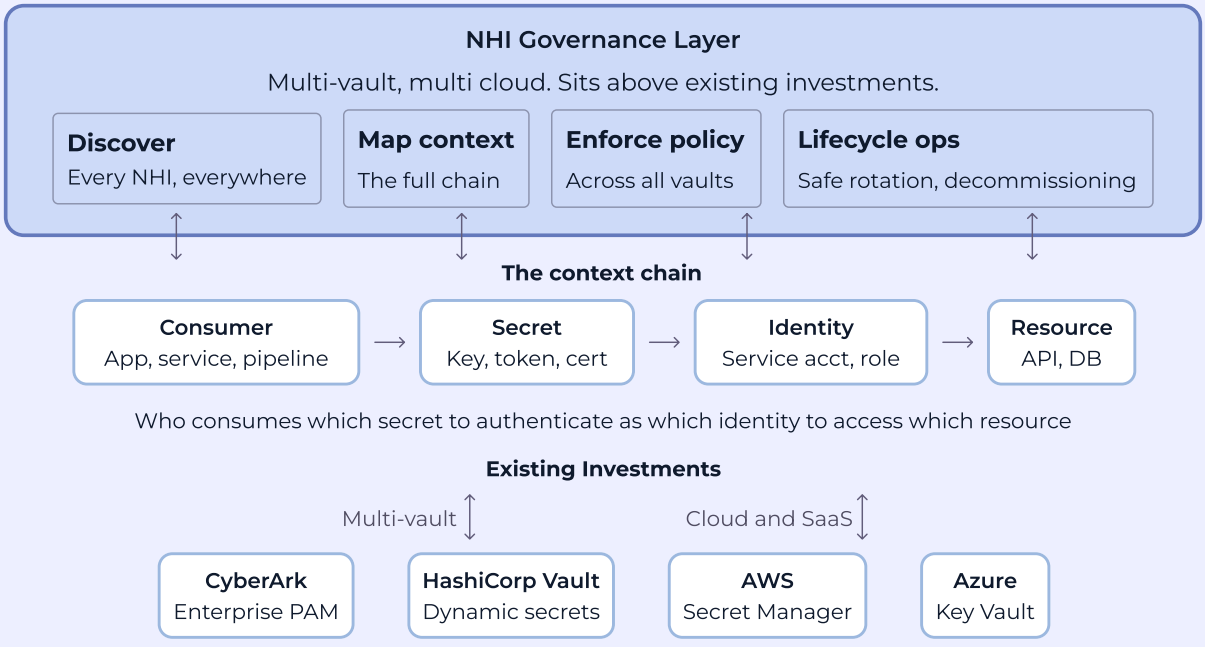
Rotation and decommissioning must account for every consumer of a secret before execution. Safe rotation means knowing what will break and coordinating the update across all dependent systems. **Safe decommissioning means confirming that an identity is truly unused, not just that it hasn't been seen recently.**

This continuous lifecycle approach stands in contrast to traditional PAM models that rely on periodic reviews. For NHIs, where access changes continuously through deployments and automation, waiting for a quarterly review cycle is not just inefficient, it is dangerous.

Complement PAM: Don't Replace It

The governance layer should not duplicate PAM's strengths. It should extend them. Route secrets into CyberArk, Delinea, or cloud-native vaults as appropriate. Leverage PAM's session management for human privileged access. But add the NHI-specific capabilities (discovery, context, lifecycle automation, multi-vault visibility) that PAM was never designed to deliver.

PAM platforms remain the right answer for credential vaulting, session management, and human privileged access. What organizations need is a governance layer purpose-built for NHIs that works across their entire vault and cloud landscape, extending PAM's value rather than duplicating it.



Next Steps

The challenge facing organizations today is not whether PAM is important; it is whether PAM alone is sufficient for the identity reality enterprises now operate in.

Non-human identities already represent the majority of privileged access. They authenticate through secrets scattered across multiple vaults and cloud platforms. And they operate at a scale and speed that human-centric PAM workflows were never designed to govern.

Effective NHI governance does not require abandoning your PAM investment. It requires augmenting it with a purpose-built layer that provides the visibility, context, and lifecycle automation PAM was never designed to deliver, across every vault and every cloud.

Start by asking you team three questions:

1. **Can we enumerate every non-human identity in our environment, not just what's in the vault?**
2. **For any given secret, do we know every consumer, system, and dependency chain attached to it?**
3. **Could we rotate or decommission any NHI today without risk of an outage?**

If the answer to any of these is no, your PAM program has room to grow.