

# HashiCorp Vault Manages Secrets. It Does Not Make Policy Cryptographically Binding.

by Nick Clark | Published March 27, 2026 | PDF

HashiCorp Vault became the standard for secrets management by centralizing credentials, encrypting data at rest, and controlling access through dynamic secrets and fine-grained ACL policies. Vault solved the secrets sprawl problem. But Vault manages access to secrets. The policies that govern what those secrets can be used for, once retrieved, are not cryptographically bound to the secrets themselves. Once a secret leaves Vault, governance becomes the application's responsibility. The gap is between managing secrets and cryptographically governing their use.

---

Vault's contribution to infrastructure security is substantial. Dynamic secrets, lease management, and the transit engine represent genuine engineering. The gap described here is not about secrets management quality. It is about where governance ends.

## Access control is not use control

Vault controls who can access a secret through ACL policies, authentication methods, and audit logging. When an application authenticates and requests a secret, Vault evaluates the policy and either grants or denies access. This is access governance.

But once the secret is retrieved, Vault has no further governance over how it is used. A database credential retrieved from Vault can be used for any query the database allows. An API key can be used for any call the API accepts. The governance ended at the Vault boundary. What happens after retrieval is outside Vault's control.

## Policies govern Vault, not the system

Vault's policy language governs operations within Vault: which paths can be read, which secrets can be created, which authentication methods are allowed. These policies are enforced by Vault's own authorization layer.

But the broader system governance, what an agent is allowed to do with a credential, whether a particular operation complies with regulatory requirements, whether the action chain that led to the credential request is itself valid, exists outside Vault. Vault does not know the semantic context of a secret request. It knows the requester's identity and the requested path. It does not know why the secret is needed or what governance should apply to its use.

## What cryptographic governance provides

Cryptographic governance makes policy cryptographically binding. Every operation is gated by a signed policy reference that specifies what is allowed, under what conditions, and with what constraints. The policy is not a separate system that checks access. It is a cryptographic artifact that binds to the operation itself.

In a cryptographically governed system, a credential would carry its governance constraints with it. An agent using the credential would be validated not just for access but for the specific operation it intends to perform, against the signed policy attached to the credential. The governance would not end at the secret retrieval boundary. It would travel with the secret through every operation.

## The remaining gap

Vault solved secrets management. The remaining gap is in governance scope: whether policy can be cryptographically bound to operations rather than just to access decisions, ensuring governance persists through the entire lifecycle of a secret's use.

Cryptographic Governance All 21 steps →

Policy that binds cryptographically — not by convention.

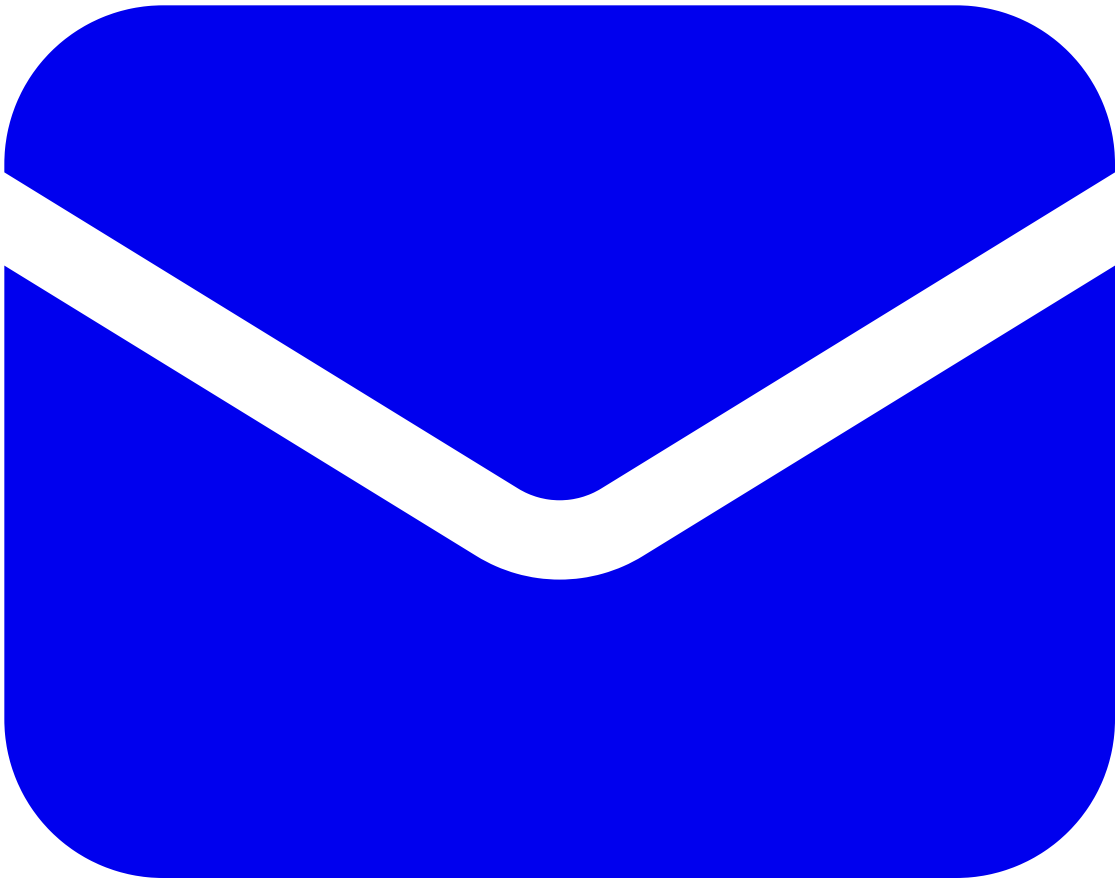
Patent
US 19/561,229 · filed
Primary Technical Disclosure
○ Ethical Enforcement as Infrastructure: Cryptographic Governance for Autonomous Systems
Secondary Technical
○ Governance Gate as Deterministic Precondition: No Verification, No Execution○ Canonical Alias to External Policy Indirection: Policy Evolution Without Agent Mutation○ Immutable-by-Default Policy Objects: Governance Changes Through Successor Issuance○ Runtime Policy Resolution Pipeline: Mandatory Verification Before Every Execution○ Freshness, Revocation, and Anti-Rollback Controls: Preventing Stale Authority○ Memory-Derived Eligibility Conditioning: Past Violations Constrain Future Authorization○ Intent-Independent Authorization: Governance Without Alignment Scoring○ Execution Feedback as Enforcement Signals: Operational Outcomes Shaping Future Authorization○ Trust Degradation as State Transition: Policy-Defined Narrowing of Permitted Actions○ Structural Quarantine: Execution Prevention Until Authorized Remediation○ Lineage-Constrained Governance Inheritance: Constraints That Persist Across Generations○ Unauthorized Fork Prevention: Lineage Continuity as Anti-Cloning Mechanism○ Meta-Policy Objects: Higher-Order Constraints Across System Behavior Categories○ Quorum-Based Governance Override: Multi-Party Approval With Signature-Chain Continuity○ Distributed Alias Publication: Policy Dissemination Through Federated Registries○ Fallback Enforcement Agents: Distributed Monitors as Defense-in-Depth○ Append-Only Governance Audit Ledger: Tamper-Evident Records of Every Authorization○ Governance Without Persistent Keypairs: Trust-Slope Authorization Replacing Static Keys○ Execution Eligibility Indicator: Dynamic Computation From Policy, Memory, and Lineage
Applications (General)
○ EU AI Act Compliance Through Structural Governance○ Financial Services Audit Trails Without Trusted Intermediaries○ Healthcare Compliance Through Structural Governance○ Defense Data Classification Enforcement○ Environmental Monitoring With Tamper-Proof Governance○ Pharmaceutical Supply Chain Governance○ Nuclear Facility Operational Governance○ Child Safety Content Enforcement
Applications (Specific)
● HashiCorp Vault Manages Secrets. It Does Not Make Policy Cryptographically Binding.○ AWS KMS Manages Encryption Keys. The Keys Do Not Carry Governance.○ Open Policy Agent Decoupled Policy From Code. The Policy Is Not Cryptographically Bound.○ Styra Made OPA Enterprise-Ready. The Governance Model Did Not Change.○ Snyk Finds Vulnerabilities Before Deployment. Governance After Deployment Is Still Manual.○ Palo Alto Networks Inspects Traffic. It Does Not Govern the Operations That Generate It.○ SPIFFE/SPIRE Provides Workload Identity. The Identity Has No Cryptographic Governance Binding.○ cert-manager Automates Certificate Lifecycle. The Certificates Carry No Governance Policy.○ Keycloak Provides Open-Source Identity Management. The Tokens It Issues Carry No Governance Binding.○ HashiCorp Boundary Provides Zero-Trust Access. The Access Sessions Have No Cryptographic Governance.○ Teleport Provides Unified Infrastructure Access. Access Control Is Not Cryptographic Governance.○ BeyondTrust Manages Privileged Access. Privilege Is Not Cryptographic Governance.○ CyberArk Pioneered Privileged Access Security. The Privilege Model Has No Cryptographic Governance Layer.○ 1Password Made Password Management Accessible. The Credentials It Manages Are Still Credentials.
Cryptographic Governance overview →
AQ
deterministic
autonomy

Legal

Subject to one or more pending U.S. and international patent applications, see Patents for the current list and status. No license, express or implied, is granted. Any use requires a separate written agreement—see Licensing. Patent applications referenced on this site are pending. Claim scope, if any, is subject to examination and may issue in altered form or not at all. See Legal for terms and conditions.



Last updated: 2026-03-03




- 
- nick@qu3ry.net
- 72 28 14 36 01

Invented by Nick Clark | Founding Investors: Devin Wilkie