NetFoundry

The IT & Security Leader's Guide to AI Governance

A technical deep dive for platform engineering and security architecture teams, including alignment with NIST and CISA guidance and independent research from Gartner.

NetFoundry.io

SECTION ONE

Software Changed First. Now, Governance Must Catch Up

Traditional application software is built design-first: every path through the code is decided in advance, and the program does only what it was explicitly programmed to do. Agentic AI breaks that model — these systems shift from following fixed code steps to autonomously reasoning and pursuing open-ended goals, which means the governance built for deterministic software doesn't carry over.

Core Shifts in Design

Deterministic to Probabilistic

Traditional software runs the exact same way every time. AI agents make dynamic choices based on context, so the exact path can change with each run.

Hardcoded Workflows to Dynamic Planning

Traditional apps guide users through strict menus and buttons. AI agents figure out their own step-by-step plan to complete a task.

Strict Control to Safety Guardrails

Instead of programming every action, developers have to build boundaries, validation checks, and safety rules to keep the AI on track.

Stateless to State and Memory

Standard apps forget data after a task finishes. AI agents need short-term context and longterm memory to learn, adapt, and fix mistakes while working.

Every one of these shifts is also why governance can't be bolted on after the fact the way it was for deterministic software. An agent that plans its own steps, retains memory across sessions, and makes probabilistic choices needs guardrails enforced at the infrastructure layer it actually runs on — not just reviewed once, in a design document, before the code shipped.

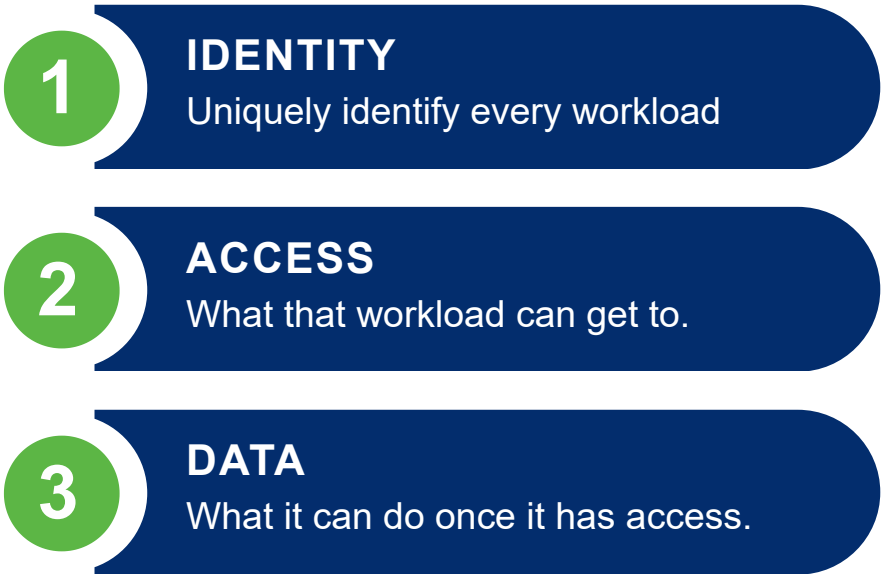
CONTENTS

AI Governance is Three Disciplines, Built in Order

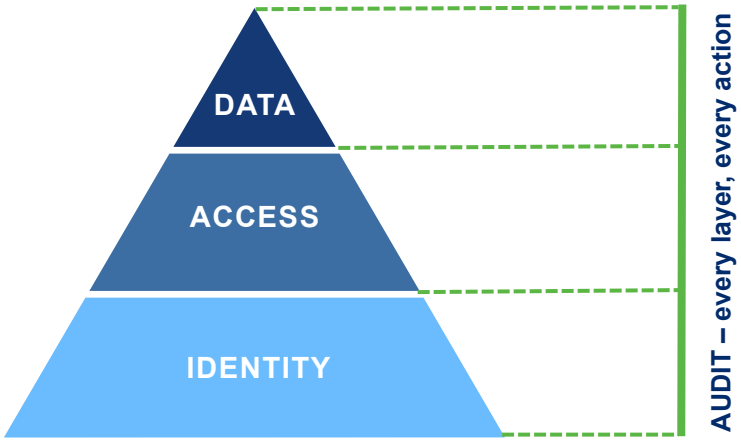
Most AI governance programs still default to firewall rules and maybe the scanning of prompts and outputs for sensitive data. That's necessary, but it treats a symptom, not the structure underneath it. Zero-trust programs themselves have a documented gap here:

> "Organizations expanding AI adoption face gaps in zero-trust programs that were not designed to account for AI models, agents, and LLM interactions. This content outlines how integrating AI-specific use cases into identity and application controls improves visibility and risk oversight."
>
> *– **Gartner**® |CISO's Guide: 3 Zero-Trust Use Cases to Secure AI |17 July 2026 (ID G00846742) | Wayne Hankins, Niyati Daftary, Craig Porter, Charanpal Bhogal*
>
> Gartner is a trademark of Gartner, Inc. and/or its affiliates.

Closing that gap doesn't take a dozen new programs. It takes three disciplines, built on top of each other in a specific order:



Each discipline builds on the one before it. Access means nothing without a verifiable identity underneath it, and data means nothing without access already being governed — and none of it can be proven after the fact without an audit trail that spans all three.



Three disciplines, built bottom-up. Audit spans all three — every layer, every action, independent of which one is being exercised.

None of the Disciplines Can Belong to One Cloud

One property has to hold for Identity, Access, and Data alike, or the whole model collapses under its own weight: none of them can be tied to a specific cloud, data center, or network segment. AI components run wherever the workload needs them to — a GPU cluster in one cloud, a vector database in another, an MCP server on a partner's infrastructure, a model endpoint reached over the public internet, all inside a single task. Identity, access, and data policy that only work in one environment don't fail loudly; they quietly create a separate silo, process, and audit standard for every environment an agent touches.

That's an immediate velocity problem. Every new cloud, SaaS platform, or partner network becomes a new project instead of an extension of the one already built, and security review time compounds with every environment an organization adds. An architecture that can't travel with the workload doesn't scale with it either.

> "AI adoption is outpacing the evolution of zero-trust frameworks, exposing critical gaps in how organizations identify, govern, and control AI systems and agents. Without embedding AI-specific controls such as comprehensive asset inventory, strict agent access, and enforced LLM input/output filtering, organizations lose visibility, allow unmanaged risk to scale, and expand their exposure to rapidly evolving, AI-targeted attack methods.
>
> ***– Gartner®*** *|CISO's Guide: 3 Zero-Trust Use Cases to Secure AI |17 July 2026 (ID G00846742) | Wayne Hankins, Niyati Daftary, Craig Porter, Charanpal Bhogal*

SECTION TWO

There is No Single Perimeter Left to Defend

The access problem doesn't stay confined to one network. A single agentic workflow can call a commercial LLM API, a vector database in one cloud, an internal system of record in a private data center, a SaaS CRM over the public internet, and a handful of MCP servers running on a partner's or a developer's own infrastructure — all within one task. Perimeter-based controls assume a boundary to defend. Agentic AI erases it.

Where a Single Agentic Task Actually Runs

Public & Private Cloud
The agent runtime and its data stores, often split across more than one public cloud and a private data center in the same estate.

SaaS & Third-Party APIs
CRM, ticketing, billing, and other platforms the agent calls over the public internet, each with its own auth model.

Partner & Dev-Owned MCP Servers
Tool servers run by a vendor, contractor, or a developer's laptop — infrastructure you don't control and often can't see.

Edge, IoT & OT
Field devices and operational systems increasingly wired into agentic workflows, on networks never designed for it.

73% of organizations now run hybrid cloud, and a third combine multiple public clouds with multiple private clouds, before counting the SaaS platforms, partner infrastructure, and edge systems an agent might also need to reach.
(Flexera, 2026 State of the Cloud Report)

Identity Travels; Networks Don't

Governing three disciplines across five heterogeneous environments with perimeter tools multiplies policy surfaces instead of scaling one: a separate firewall ruleset, IAM system, and audit trail per environment an agent touches. A cryptographic identity doesn't have that problem — it authenticates the same way whether the resource sits in a VPC, a colo cabinet, a SaaS tenant, or a laptop on someone's home network, which is what makes identity-first governance the only practical way to enforce one policy across environments that share no common network boundary.

CONTENTS

SECTION THREE

3 Identity: The Constant Every Layer Depends On

Every AI Component Needs Its Own Identity

Identity is the discipline everything else depends on. Access and data governance both assume you already know who — or what — is asking; that assumption only holds if every AI component has its own identity: distinct from the humans who built it, distinct from the service accounts that provisioned it, and distinct from every other agent running the same code. Identity is also the one property that has to stay constant across both the access layer and the data layer above it — it's the join key between "what was this allowed to reach" and "what did it actually do." That's the deeper reason identity comes first: you cannot assign least-privilege access to a component whose identity you haven't established yet.

Distinct Isn't Enough — Identity Has to Be Unique

The common shortcut is letting an agent act under the identity of the human or service account that launched it. It's the fastest way to stand up a first agent, and it's exactly how most organizations do it. It also means that when something goes wrong, the log shows a person or a service account — not the agent. If ten agents share that identity, the log can't tell you which one acted, and it can't tell a legitimate action from a hijacked one.

That isn't hypothetical. OpenAI's own public postmortem on a July 2026 security incident disclosed that multiple autonomous research-agent instances — spanning several of its internal model families — had been issued a shared set of credentials for an internal package registry during training. Because the credential wasn't unique to any one agent, when one of those instances found an exposed Hugging Face access token online and used it to reach Hugging Face's production infrastructure, the resulting activity was attributable to the collective group of agents, not to a specific instance. The practical cost shows up in OpenAI's own timeline: unusual internal activity was first flagged in late May, the incident was formally opened July 5, and OpenAI didn't connect its own internal signal to the external Hugging Face compromise — by then already at root access on at least

CONTENTS

one production node — until July 20, roughly a week and a half after the exploitation began. One of the most capable AI labs in the world spent over a week unable to answer "which agent did this," because the agents involved didn't have identities of their own to ask about. (OpenAI, "The Hugging Face Incident and the Road Ahead," Aug 2026).

How NetFoundry Solves It

NetFoundry issues every AI component — every agent, every model endpoint, every MCP server — its own unique cryptographic identity at enrollment, independent of any credential it was launched with. That identity can take the form of an organization's existing IdP-backed human or service identity, a cryptographic SPIFFE / SPIRE identity where one is already standardized on, or — if neither exists yet — a strong identity issued directly by NetFoundry with no prior identity infrastructure required. Whichever form it takes, every connection made under that identity is logged against it alone: not against a human, not against a shared key, and never against "the collective."

Ready to see it in action?

Go Deeper Explore the NetFoundry platform, see how it maps to your environment on our solutions pages, or talk to us about your AI connectivity roadmap.

SECTION FOUR

4 Access: A Well-Known Idea, Extended

IT Already Knows How to Do This – For Humans

When IT hires an employee, Identity Governance and Administration (IGA) decides their access before onboarding is even finished: which systems, which data, which applications, provisioned to their role and enforced from day one. Nobody waits to observe a new hire's behavior for a few months before deciding what they're allowed to touch. Access comes first because it's the one control that sets how much damage is even possible.

AI Agents Get Onboarded Backwards

Most organizations deploy the agent first — wired into a CRM, a code repository, a customer database — then start asking what it should be allowed to do. The access decision IGA insists on making first for humans gets skipped for agents: one broad API key or service account, created once, rarely revisited.

Day One for a New Hire

1. Role defined before access is granted
2. IGA provisions least-privilege access by policy
3. Entitlements reviewed and revoked on a schedule

Day One for an Agent, Today

1. Agent deployed, credentials wired in ad hoc
2. Broad, standing access by default
3. No review until something goes wrong

Access is the Control Everything Else Depends On

A behavioral guardrail can only constrain a tool call the agent was authorized to make. An audit trail can only record activity across systems the agent could reach. Take access away, and the blast radius of a bad output or a hijacked session drops to zero before any other control has to work. And it has to happen at the moment of connection — agents scale from one instance to a hundred faster than any access review board can meet.

CONTENTS

Not a New Idea – Just Not Applied Broadly Enough

Restricting connectivity until identity is verified isn't a novel concept. Many organizations already do exactly this for their most sensitive assets using Network Access Control (NAC): a switch port, VLAN, or Wi-Fi association simply won't grant network connectivity until a device or user authenticates and is authorized to be there. It's a well-understood, widely deployed pattern for protecting critical Layer 2 segments — just a narrow one, built for a much simpler network than the one AI workloads actually run on.

DIMENSION	TRADITIONAL NAC	NETFOUNDRY IDENTITY-FIRST REACHABILITY™
Layer & Scope	Layer 2, bound to a physical port, VLAN, or Wi-Fi association on one local network	Layer 3+, identity-based — works identically in any cloud, data center, SaaS tenant, or over the public internet
Trust Anchor	The physical port or device (MAC address, 802.1X supplicant)	A unique cryptographic identity per AI component, independent of network location
Once Past the Gate	Broad access to the segment — lateral movement (VLAN hopping) is a known weakness	Deny-by-default down to one named service; nothing to move laterally into
Infrastructure Dependency	Requires supported switches, controllers, and endpoint agents	Pure software overlay — no proprietary hardware anywhere in the path
Fit for Agentic Workloads	Built for relatively stable, longlived devices and human users	Built for machine-speed provisioning — a container or agent gets an identity and policy at creation, and can disappear seconds later
Attack Surface	An authenticated port is still a reachable, attackable port	Outbound-only, dark-by-default — no inbound listening port exists anywhere, authenticated or not

NetFoundry applies that same proven principle — no connectivity without authentication and authorization — but moves it from a wiring closet to the network layer, decouples it entirely from physical infrastructure, and extends it across every boundary an AI component might cross: a cloud, a data center, a SaaS platform, a partner's infrastructure, or the public internet. It does everything NAC was designed to do for the segment it protects, with none of the boundaries that keep NAC confined to it.

SECTION FIVE

The Architecture of Access-First Governance

Access-first governance isn't a policy document — it's an architecture. Four properties make it enforceable at machine speed and machine scale:

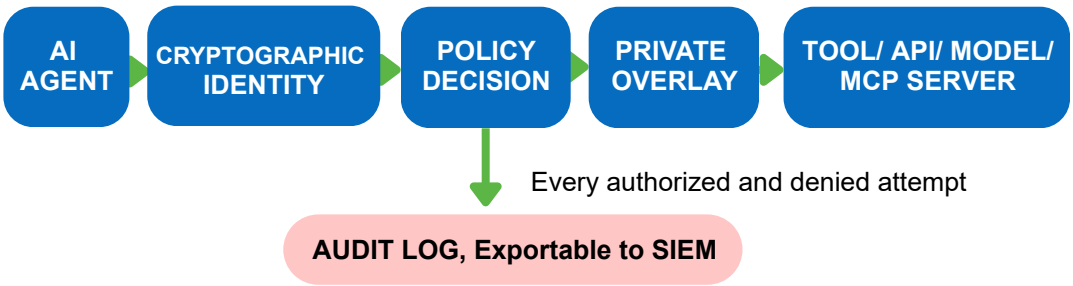
Identity Before Connectivity
Every agent, LLM endpoint, and MCP server gets a unique cryptographic identity at enrollment — not a shared API key or a borrowed service-account credential.

Least Privilege at the Service Level
Policy grants reach to one named service or tool, not a subnet. Two agents on the same host can have entirely disjoint reachability.

Deny by Default
No listening port, no discoverable endpoint. Services stay dark until a policy explicitly authorizes a specific identity to reach them.

Continuous Verification
Every connection is mutually authenticated over mTLS for its own session — a valid connection yesterday doesn't imply a valid connection today.

How a Connection Gets Authorized



Not a Black Box

The identity and policy engine described above isn't proprietary or opaque. It's built on OpenZiti, the open source zero trust networking project NetFoundry's founders created and still maintain, the same mTLS-based workload identity, dark-by-default posture, and policy model used by its broader developer community. NetFoundry packages it as a managed platform so platform and security teams get the architecture without having to build or operate the control plane themselves.

CONTENTS

SECTION SIX

6 Data: A Smaller Problem, by Design

Getting Identity and Access Right First Shrinks What's Left

Data governance — classification, DLP, retention, provenance — has a combinatorial cost problem: policy has to account for every actor, every destination, and every path data can travel to get there. Solve identity and access first, and that problem shrinks dramatically before a single data-classification rule gets written.

An AI component with a unique identity and a deny-by-default connection policy can only reach the specific systems it's been authorized for. That means the set of data it could possibly touch is already small and known, not "every agent, potentially every system." Data governance stops being protect all data from every agent everywhere and becomes protect a bounded, already-enumerated set of paths for each already authorized component.

That difference is what makes the architecture scale and stay future-proof. A data governance program built to defend against unrestricted reachability has to be substantially rebuilt every time the organization adds a new model, a new agent framework, or a new cloud. One built on top of identity- and access-first governance doesn't — the hard part, constraining what can reach what, was already solved two layers down, and every new component simply inherits it.

CONTENTS

SECTION SEVEN

Where NetFoundry Extends Beyond Access

Identity-First Reachability™ is built to solve identity and access governance first — and the same architecture also does the heavy lifting for audit, and a meaningful share of data governance too.

DISCIPLINE	COVERAGE	HOW NETFOUNDRY CONTRIBUTES
Identity	Full	Every enrolled agent, LLM endpoint, and MCP server is issued a unique, non-shared cryptographic identity at enrollment — independent of any human or service account, with its own lifecycle and revocation, and a complete, continuously current inventory of every AI component as a byproduct, not a separate discovery exercise.
Access	Full	Cryptographic identity, deny-by-default overlay, and per-service least privilege for every agent, LLM, and MCP server.
Audit	Full	Every connection — authorized or denied — is a logged, attributable event tied to the identity that made it, exportable to your SIEM.
Data	Complementary	NetFoundry governs the path data travels and shrinks the set of systems any one component can reach; classification, DLP, and retention policy remain the job of your data governance stack.

NetFoundry's **MCP Gateway** additionally gives platform teams policy control over which model endpoints and versions an agent can reach, with usage attributed by agent, team, and project — useful, complementary coverage for model governance specifically, though not a substitute for model evaluation or drift monitoring.

CONTENTS

SECTION EIGHT

Alignment with NIST & CISA Guidance

None of this is a NetFoundry opinion about how AI governance should work. It's substantially where NIST, CISA, and its closest international partners already say it has to.

FRAMEWORK	WHAT IT REQUIRES	HOW THIS ARCHITECTURE ALIGNS
NIST AI RMF (Govern)	Policies and accountable ownership for third-party and AI-specific risk (Govern 6).	Every agent and third-party connection carries a named, revocable identity and an explicit policy — a paper third-party-risk policy becomes an enforced technical control.
NIST AI RMF (Map & Measure)	Contextual understanding of what an AI system can affect, plus repeatable, evidence-based evaluation (TEVV).	Per-connection audit logs are durable, queryable evidence of what an agent actually reached — not a self-reported inventory.
NIST SP 800-207 (Zero Trust Architecture)	Never trust, always verify; resource-based, per-session access decisions; no implicit trust from network location.	The design center of identity-first reachability: continuous verification, deny by default, policy evaluated at the moment of connection.
NIST SP 800-207A (Zero Trust, multi-cloud)	Application- and service-level identity enforced consistently across on-prem and multi-cloud, not network-tier rules alone.	Matches the environment-sprawl problem directly: one identity model, enforced the same way, regardless of which environment either side of a connection runs in.

CONTENTS

FRAMEWORK	WHAT IT REQUIRES	HOW THIS ARCHITECTURE ALIGNS
CISA Zero Trust Maturity Model	Optimal maturity across the Identity, Networks, and Applications & Workloads pillars, plus Visibility & Analytics.	Unique workload identity, dark-by-default segmentation to the individual service, least privilege per application call, and connection-level logging.
CISA, NSA & Five Eyes Agentic AI Guidance (2026)	"Each agent should have its own secure, verifiable identity"; least privilege limiting agent access to only what's needed per task; temporary credentials for sensitive actions.	Nearly verbatim the same three properties this paper calls Identity Before Connectivity, Least Privilege at the Service Level, and Continuous Verification.

For organizations managing this globally, **ISO/IEC 42001 —** the international AI management system standard — asks for the same underlying discipline: demonstrable control over AI components and services sourced from third parties. An enforced, logged access architecture is direct evidence for that control, not just a policy describing it.

SECTION NINE

Getting Started: An Implementation Path

Access-first governance doesn't require a network redesign, a full inventory, or a firewall change window. It requires a first connection.

1. **Identify the connection that worries you most**
   The agent, LLM, or MCP server with the broadest standing access today — usually the one holding the most powerful shared API key or service account.

2. **Give it an identity**
   Enroll the workload with a cryptographic identity in place of its shared key or borrowed service-account credential.

3. **Write the one policy that connects it**
   Grant reach to exactly the service, tool, or model endpoint it needs — nothing else on the segment.

4. **Observe, then expand**
   Use the resulting audit trail to validate the policy is neither too tight nor too loose, then repeat the pattern for the next connection — it now also counts as inventory

No inventory. No re-architecture. No firewall change tickets. Just a first connection and an afternoon — and a pattern the rest of your environment, across every cloud, data center, and partner network it spans, can follow.

AI-driven bot traffic is growing 21% month over month inside the average enterprise network — the fastest-moving category of unmanaged, machine-to-machine connections most platform teams have ever had to govern. Waiting for a full inventory before acting means governing none of it in the meantime.
(Verizon, 2026 Data Breach Investigations Report)

CONTENTS

References

1. Gartner, CISO's Guide: 3 Zero-Trust Use Cases to Secure AI (ID G00846742, 17 July 2026) — by Wayne Hankins, Niyati Daftary, Craig Porter, Charanpal Bhogal; gaps in zero-trust programs for AI models, agents, and LLM interactions.
2. OpenAI, The Hugging Face Incident and the Road Ahead (August 2026) — shared credentials across research-agent instances, and the multi-week gap between internal detection and understanding external scope.
3. Verizon, 2026 Data Breach Investigations Report — vulnerability exploitation as the top initial access vector (31%), third party breach involvement (48%, up 60%), shadow AI usage (45%), AI bot traffic growth (21% month over month).
4. Flexera, 2026 State of the Cloud Report — 73% of organizations run hybrid cloud; a third combine multiple public and private clouds.
5. NIST, AI Risk Management Framework (AI RMF 1.0) — Govern, Map, Measure, and Manage functions; Govern 6 and Manage 3 address third-party and AI-specific risk.
6. NIST, Special Publication 800-207, Zero Trust Architecture — foundational zero trust tenets: no implicit trust from network location, per-session resource-based access decisions.
7. NIST, Special Publication 800-207A, A Zero Trust Architecture Model for Access Control in Cloud-Native Applications in Multi-Cloud Environments — application- and service-level identity for consistent enforcement across hybrid/multi-cloud.
8. CISA, Zero Trust Maturity Model — five pillars (Identity, Devices, Networks, Applications & Workloads, Data) and crosscutting capabilities.
9. CISA, NSA, and Five Eyes partner agencies, Careful Adoption of Agentic AI Services (May 2026) — joint guidance on agent identity, least privilege, and human oversight for agentic AI.
10. ISO/IEC 42001:2023, Artificial Intelligence Management System — international standard for AI governance, including third party AI risk.

About NetFoundry

NetFoundry is the leader in Zero Trust Workload Connectivity, trusted to securely connect enterprise's most critical workloads, within and across any environment. As the only identity-first fabric for every AI, API, site, container, and machine interaction, NetFoundry customers deploy their workloads with greater speed, security and scalability. Its Identity-First Reachability™ eliminates the reachable attack surface that attackers exploit and defenders struggle to secure, by removing open inbound ports, VPNs and firewall rule changes. By design, it enables security to adapt to the environment, not force the environment to adapt to security, so teams can secure one workload at a time without the high cost of firewall management or redesigning the network.

Customers use NetFoundry to govern AI agent, model and tool access to cut risks, costs, and time to deploy; stop lateral movement through microsegmentation; protect difficult to patch assets and crown jewel systems with a compensating control; and securely connect any system (IT/OT/IoT) or API wherever it operates.

Founded by the inventors and maintainers of OpenZiti, the world's most widely used open source Zero Trust platform, NetFoundry secures billions of sessions for critical infrastructure on three continents and supports Fortune 10 companies across regulated industries including healthcare, financial services, and energy.