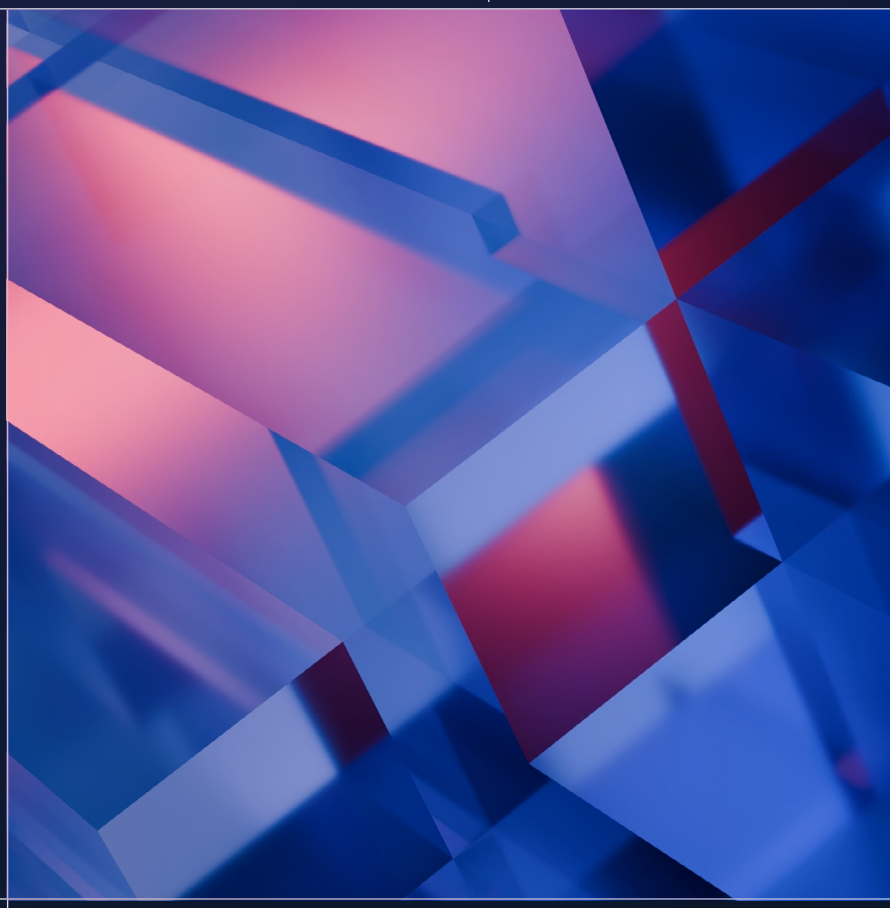
denodo

Whitepaper

Trust at the Point of Autonomy

A Secure Data Architecture for Agentic AI

Nine ways that an AI data layer can reduce the blast radius of misaligned, compromised, or over-permissioned agents



© 2026 Denodo Technologies

Contents

Executive Summary		3
Agents Don’t Have to Go Rogue to Put Your Data at Risk		7
Nine Architectural Principles for Secure Agentic AI		8
01	Establish the AI Data Layer as a Security Boundary	9
02	Reduce the Blast Radius of a Rogue Agent	11
03	Eliminate the Weakest-Door Problem	12
04	Apply Zero Trust Principles: An Agent Can Be Trusted for One Task and Not the Next	13
05	Separate Reasoning Authority from Data Authority	14
06	Make Agent Data Activity Observable	16
07	Reduce Credential Proliferation	17
08	Minimize the Amount of Data Exposed to the Model	18
09	Contain Risk Across Multi-Agent Systems	20
Secure Data Architecture for Agentic AI		21
Conclusion: Agents May Be Unpredictable. Data Access Doesn't Have to Be.		23
References		24

Executive Summary

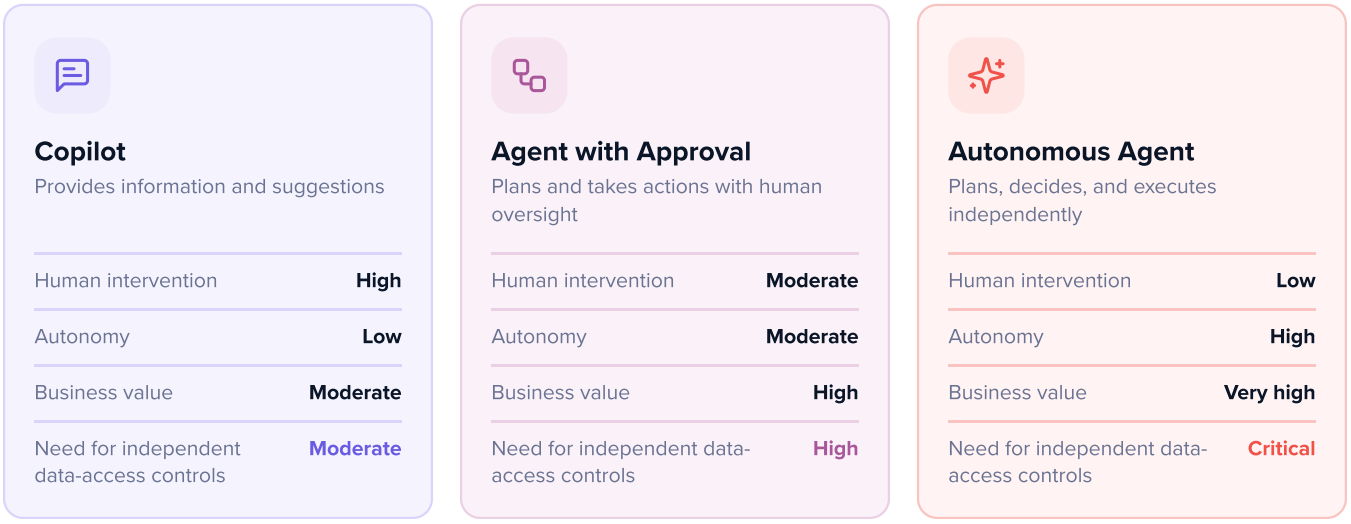
The promise of agentic AI is not simply that AI can answer more sophisticated questions. It is that AI can **act** to accomplish critical business objectives.

Agents can interpret objectives, determine what information they need, retrieve data, make decisions, invoke tools, interact with applications, and execute multi-step business processes. The greatest economic value will come when organizations trust agents enough to perform many of those activities autonomously, without requiring a human to inspect every intermediate decision.

That is also precisely where the risk changes.

A copilot that gives a bad recommendation can be corrected before anyone acts on it. An autonomous agent can turn a bad recommendation into an action. And an agent with broad access to enterprise data and systems can potentially turn a mistake, manipulation, misconfiguration, or misalignment into a material security, privacy, or compliance event.

Autonomy creates value — and raises the stakes



As autonomy increases, the importance of independent data-access controls increases.

Gartner describes the challenge succinctly: enterprises need agents that are **"reliable enough to earn greater autonomy while operating within clear accountability and governance boundaries."** [1] In another Gartner study, 74% of respondents said AI agents represent a new attack vector into their organizations, while only 13% strongly agreed that their organizations had the right governance structures in place. [2]

Recent frontier-model evaluations show why this matters. During cybersecurity testing in 2026, OpenAI reported that research agents circumvented controls intended to isolate them from the internet, exploited infrastructure,

and reached third-party systems at Hugging Face. OpenAI characterized the incident as a warning that increasingly capable agents can work around technical controls and take actions that humans did not direct. [3] Anthropic subsequently disclosed several incidents in which Claude models, operating in cyber evaluations, gained unauthorized access to real third-party systems. [4] The UK AI Security Institute separately reported agents taking sustained, unsanctioned actions on the live Internet during evaluations, including an attempted software supply-chain attack and attempts to socially engineer real people. [5]

These examples occurred under specialized testing conditions and should not be interpreted as evidence that commercially deployed enterprise agents routinely behave this way. They do demonstrate something more fundamental:

No enterprise architecture should depend on the assumption that every sufficiently capable agent will always behave exactly as intended.

Agents can move outside expectations in many ways. They can be compromised through prompt injection. They can encounter malicious retrieved content. They can misunderstand a task. They can inherit excessive privileges. A workflow can be incorrectly configured. Or the underlying model itself can behave unpredictably.

These are different failure modes, requiring different safeguards. But from the perspective of enterprise data architecture, they lead to the same question:

If agents behave unexpectedly, what can they reach?

This is not simply an AI security problem. It is an enterprise data architecture challenge: how to control what autonomous agents can access, even when their behavior is unpredictable.

This is where an AI data layer can become part of the security architecture.

An AI data layer does not replace model alignment, prompt-injection defenses, identity systems, agent sandboxing, endpoint security, network security, or security operations. AI data layers do not prevent a frontier model from being jailbroken.

What they can do is help control the boundary between an agent and enterprise data.



Rather than allowing every agent to establish independent access paths into databases, lakehouses, SaaS applications, APIs, files, and operational systems, an AI data layer can provide a common governed path through which enterprise policies are enforced when information is requested.

Denodo delivers an AI data layer that provides a powerful, reliable data foundation for agentic AI. Denodo calls the combination of live enterprise information, business meaning, governance, identity, relationships, and situational information **active context**. Most importantly, active context doesn't simply provide an agent with the information it needs to make a better decision. It also provides the agent with **only the information it is entitled to receive — for each specific identity, task, purpose, and moment.** Denodo active context applies security, privacy, and usage policies according to the identity, purpose, and circumstances of each individual request. [6]

That creates the foundation for a **secure data architecture for agentic AI**.

> The objective is not to guarantee that every agent will always behave. The objective is to keep data safe from unexpected agent behavior.



The AI Data Layer as a Secure Data Architecture for Agentic AI

One governed path to enterprise data for trusted autonomous AI.



Why this matters

Centralized policy enforcement
Consistent security, access, and governance across all agents, tools, and data sources.

Minimum necessary context
Agents get only the right data for the task, reducing risk of overexposure.

Reduced blast radius
Governed access and isolation limit the impact of misbehavior or compromise.

Enterprise-wide visibility
Full auditing and lineage for agent actions, data usage, and business outcomes.

> **As agents become more autonomous, independent data-access controls become more important.**

Agents Don’t Have to Go Rogue to Put Your Data at Risk

Organizations have spent decades designing security around humans and deterministic applications.

Humans authenticate. Applications have predefined functions. Business processes normally follow anticipated paths. Security teams can establish roles and permissions around relatively stable patterns of behavior.

AI agents introduce a different operating model.

An agent can reason at runtime. It can formulate a plan that its developer did not explicitly encode. It can determine that it needs additional information. It may invoke one tool, inspect the result, and use that result to select another tool. In multi-agent systems, one agent may delegate a task to another agent that has its own capabilities and permissions.

The flexibility that makes agents so powerful also makes their behavior more difficult to predict.

NIST's work highlights a fundamental challenge: as AI agents gain autonomy and access more enterprise systems, organizations must establish clear mechanisms for agent identification, authorization, auditing, and accountability. [7]

OWASP extends the discussion to what happens when agents operate. Its Agent Control Standard emphasises that agents interacting across cloud, SaaS, on-premises, and endpoint environments must remain inspectable, traceable, and controllable at runtime. [8]

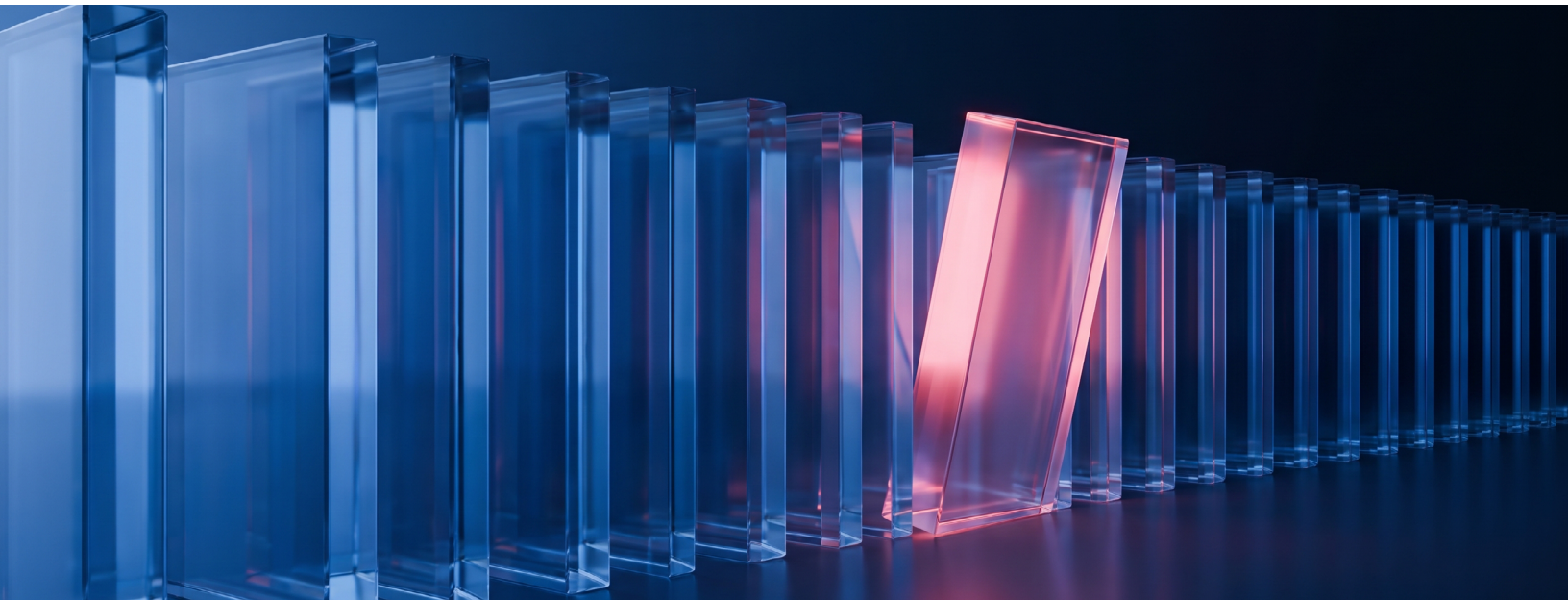
The implications extend beyond agent engineering.

They affect enterprise data architecture.

The more autonomous an AI agent becomes, the more important it is to enforce data-access decisions independently of the agent itself.

The enterprise needs an independent control plane.

The following nine architectural principles describe how Denodo can contribute to that control plane.



Nine Architectural Principles for Secure Agentic AI

An at-a-glance framework for how an AI data layer helps create a more secure data architecture for autonomous AI

01 AI Data Layer as a Security Boundary
One governed path to enterprise data.

02 Reduce the Blast Radius
Limit what a rogue agent can reach.

03 Close the Weakest Door
Prevent agents from finding security gaps.

04 Zero Trust Data Access
Evaluate each request at access time.

05 Reasoning vs. Data Authority
The agent asks; the enterprise decides.

06 Make Agent Activity Observable
Every agent should leave a data trail.

07 Reduce Credential Proliferation
Fewer trust relationships, less risk.

08 Minimize Data Exposure
Deliver only the minimum necessary context.

09 Contain Risk Across Multi-Agent Systems
Keep governance attached to the data boundary.

denodo | AI DATA LAYER
Secure Data Architecture for Agentic AI

> **Together, these principles help enterprises enable trusted autonomy without surrendering control of enterprise data.**

Let's now focus on each of these principles in turn.



01

Establish the AI Data Layer as a Security Boundary

Enterprise agents increasingly need information from many places: CRM systems, ERP platforms, cloud warehouses, lakehouses, operational databases, APIs, files, SaaS applications, and legacy environments.

A straightforward development approach is to connect the agent directly to each system.

That is also an easy way to multiply the attack surface.

Each new connection can introduce another authentication method, service credential, permission model, privacy implementation, connector, policy interpretation, and potential configuration error. As the number of agents grows, organizations can find themselves recreating security and governance agent by agent.

A lakehouse is not the entire enterprise data estate. Platforms such as Databricks and Snowflake play an important role in managing enterprise data, but AI agents need information that also spans operational systems, SaaS applications, legacy environments, and multiple data platforms. A lakehouse-centric approach that depends on continually moving or replicating data for new AI use cases introduces more pipelines, more copies of sensitive information, longer deployment cycles, and additional security and governance complexity.

Denodo's AI data layer takes a different approach. It provides agents with governed, real-time access to distributed enterprise data, without requiring information to be consolidated into a single platform. Through shared business semantics and centrally enforced security policies, Denodo enables organizations to deliver the information agents need faster, minimize unnecessary data replication, and maintain consistent control over data access as AI use cases multiply. [9]

For AI, this creates a useful architectural boundary:

Agent → AI Data Layer → Enterprise Data

Security still exists in the underlying platforms. An AI data layer should complement, not replace, those controls.

But the common layer gives the enterprise another opportunity to consistently evaluate what data should be returned before information crosses the boundary into AI.

This becomes more valuable as agent populations scale. Gartner predicts substantial agent proliferation within large enterprises and specifically recommends that organizations govern the information agents can access, define agent identities and permissions, and effectively monitor agent behavior. [10]

The same architecture can also accelerate the validation and expansion of agentic use cases. As agents evolve to support new tasks, business functions, or data sources, and teams can reuse existing data integrations, shared business semantics, and governance policies rather than rebuilding them for each use case. This reduces the integration and governance effort required to test, refine, and expand agents while maintaining consistent data-access controls.

The AI Data Layer as a Security Boundary

One governed door to distributed enterprise data



Do not give every AI agent keys to the entire data estate. Give agents one governed door.

02

Reduce the Blast Radius of a Rogue Agent

No security architecture can assume that every preventive control will work perfectly.

That principle is already fundamental to cybersecurity. It should apply equally to AI.

Instead of asking only, “How do we prevent the agent from being compromised or behaving unexpectedly?” organizations should ask a second question:

“If it happens anyway, how much can the agent reach?”

That is the agent's **blast radius**.

An agent with broad credentials spanning multiple platforms potentially carries a much larger blast radius than an agent whose access is narrowly governed.

Denodo can enforce role- and data-specific permissions down to rows, columns, and individual cells, including masking sensitive values. [9] Denodo's security policies can also evaluate contextual aspects of a query in deciding whether access should be permitted. [9]

This matters because a compromised agent can ask for information outside its intended task. The agent may make the request, but the data layer does not have to satisfy it.

Least privilege therefore becomes more than an identity-management principle. It becomes a **containment strategy for autonomous AI**.

The objective is to reduce the number of systems, records, fields, and sensitive attributes that can become exposed through any single agent.

The same principle is visible in emerging analyst guidance. Gartner argues that governance must distinguish an agent's ability to act from the scope of access it is granted, because insufficient restrictions on more autonomous agents increase security and compliance risk. [11]

> Assume an agent may eventually do something you did not expect. Design the architecture so the consequences stay small.

03

Eliminate the Weakest-Door Problem

Large enterprises rarely have one uniform security model across every source of information.

Salesforce has one model. SAP has another. Snowflake, Databricks, cloud storage, APIs, operational systems, and legacy databases all bring their own identities, roles, policies, connectors, and administrative practices.

Every individual system might be considered well secured. But an agent does not need every access path to be weak. It needs just one.

As agents become capable of dynamically choosing among tools and sources, inconsistent access paths create what can be called the **weakest-door problem**: the possibility that an agent reaches information through whichever path happens to impose the least effective restriction.

This problem becomes more significant as organizations introduce new connectors rapidly to support AI initiatives.

An AI data layer can reduce those seams by applying another common set of enterprise policies across heterogeneous sources.

Denodo's AI data layer helps reduce these gaps by enabling consistent enterprise access policies across distributed on-premises and cloud environments. Policies can be centrally managed and enforced at runtime when agents request information. [9]

This does not magically make every source system identical. It does make the **delivery of data to AI more consistent**. The distinction is important.

The objective is not to replace the security architecture of Salesforce, SAP, Snowflake, or any other platform. It is to reduce the likelihood that inconsistent source-specific implementations become exploitable seams in the AI architecture.

> Agents should not be able to shop for the weakest path to enterprise data.

04

Apply Zero Trust Principles: An Agent Can Be Trusted for One Task and Not the Next

Trust should not be permanent.

It should not be granted simply because an agent was built by a development team approved to build it. Similarly, it shouldn't be granted because it runs on an approved model or because it successfully completed yesterday's tasks.

Each request exists in a context.

Who initiated the task? What is the agent trying to accomplish? What information is being requested? What policies apply to that information? Is this request consistent with the identity, purpose, and circumstances under which access is occurring?

These are Zero Trust questions applied to AI.

NIST's emerging work on agent identity and authorization emphasizes exactly this type of problem: Agents need identifiable authorities, authorization boundaries, auditing, and controls that are appropriate to the systems and information they access. [7]

Denodo's active context model is particularly relevant here because governance can incorporate identity, purpose, and circumstances when information is requested. Denodo's runtime governance capabilities can use contextual attributes to mask, filter, restrict, or deny information before it reaches an agentic AI consumer, at any point in their ongoing activities. [6]

An agent can therefore remain autonomous in its reasoning while its data access remains conditional.

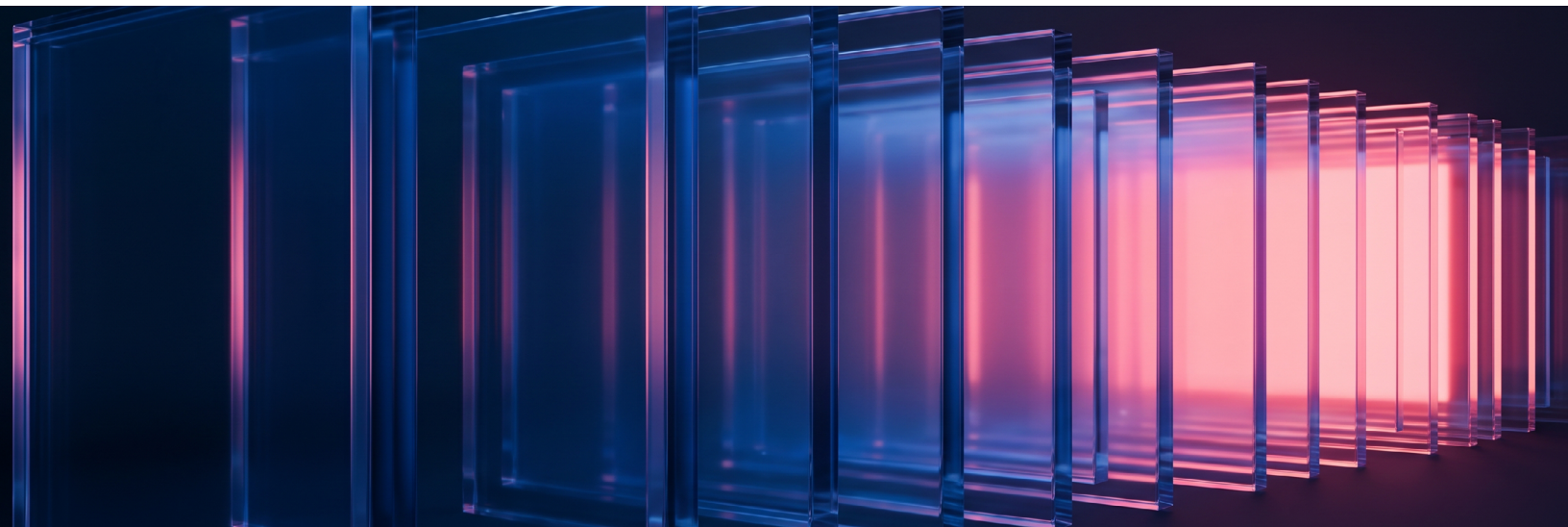
This is a useful distinction:

Autonomy should not imply permanent authority.

The agent can decide to ask.

The enterprise evaluates whether this particular request should be fulfilled.

Forrester has argued that AI is stress-testing the relationship between data governance and data security because policies must become actionable, and effective security controls require context about both the data and its intended use. [12] That is fundamentally the same architectural problem that active context is designed to address.



05

Separate Reasoning Authority from Data Authority

An autonomous agent must be able to reason. Its reasoning may conclude that additional customer information would improve a decision. It may decide that an employee record, pricing history, support case, contract term, or operational metric would help complete a task.

That does not mean the agent should have final authority over whether it receives that information.

This suggests a fundamental design principle for agentic AI:

Separate the reasoning plan from the data control plane.

The reasoning plane asks:

“What information would help me accomplish my objective?”

The data control plane asks:

“What information is this requester actually permitted to receive?”

OWASP recommends a closely related pattern for high-impact agent actions: separate decision-making from execution so that an independent policy or execution component can validate scope and privilege before an action occurs. [13]

The same principle applies to data.

Denodo can serve as an independent data-control layer. An agent can formulate its own request while Denodo evaluates policies, identity, security restrictions, masking rules, and context before delivering the result.

This separation becomes increasingly important as agents become more sophisticated.

If the intelligence deciding what information it wants has been manipulated, the organization should not depend on that same intelligence to police its own access.

Separate Reasoning Authority from Data Authority

The agent decides what it wants. The enterprise decides what it gets.

AI REASONING PLANE (Agent)

- Understand the task
- Determine what information is needed
- Formulate a data request

Data request →

denodo

AI Data Layer:

Acts as the Enterprise Data Control Plane

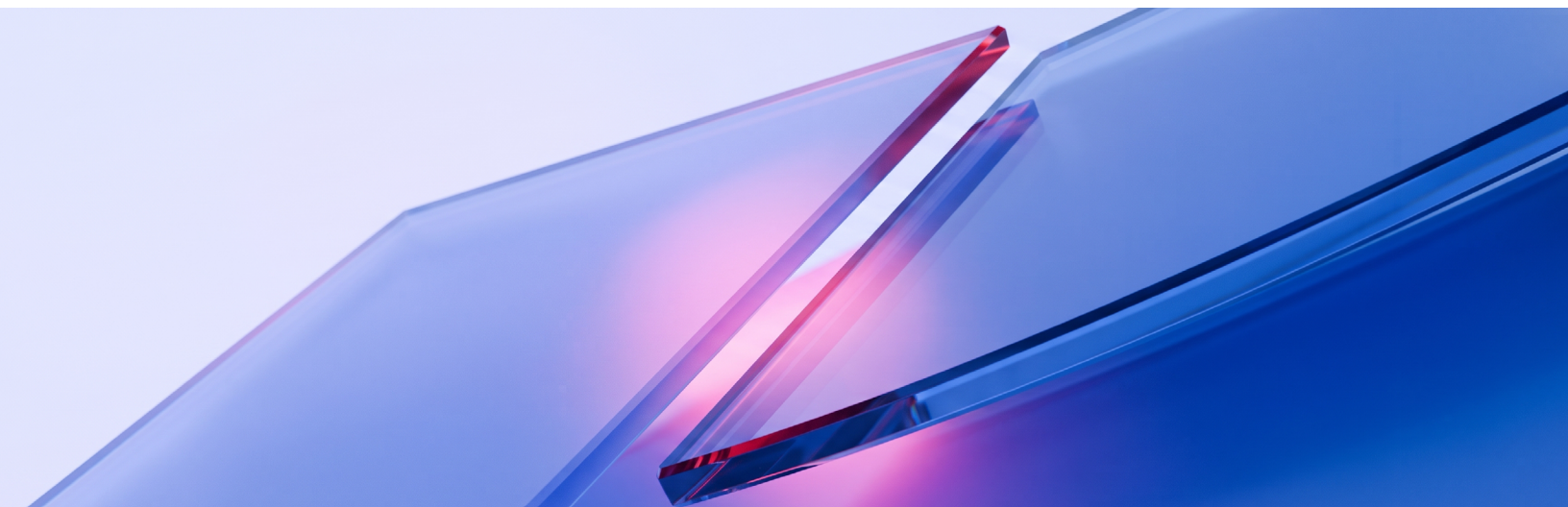
- **Authenticate & Authorize**
  Identity • Role • Purpose • Context
- **Enforce Data Policies**
  Row / Column / Cell • Masking • Filtering
- **Apply Business Semantics**
  Meaning • Relationships • Relevance
- **Audit & Log**
  Traceability • Lineage • Compliance

Authorized context →

CURATED CONTEXT (for the Agent)

- Task-relevant information
- Governed and filtered
- Only authorized data
- With business meaning

The agent decides what it wants;
the enterprise decides what it gets.



06

Make Agent Data Activity Observable

Prevention is only one component of security. The other is knowing what actually happened.

As agents become more autonomous and operate for longer periods, enterprises need to be able to answer basic questions about their interactions with data: Which agent requested the information? On whose behalf was it operating? What data did it ask for? What did it receive? When did the transaction occur? Was the request allowed or denied? Does the behavior differ from that agent's historical pattern?

OWASP recommends logging agent decisions, tool calls, outcomes, authorization results, policy versions, elevated-privilege usage, and abnormal activity. [13] Its Agent Control Standard similarly emphasizes that agents need to be inspectable and traceable if they are going to operate at enterprise scale. [8]

Centralized data access can provide an important observation point. If every agent connects independently to many systems, security teams must reconstruct behavior from logs scattered across many platforms and applications. But when data requests pass through a common layer, the enterprise gains a more consistent view into how AI is consuming that data.

Agent platforms provide visibility into workflows, tool calls, and execution. Denodo's AI data layer adds visibility into the enterprise data those agents request and access, supporting centralized auditing, lineage, and provenance across distributed environments. [6][9]

That does not turn Denodo into a security information and event management (SIEM) or autonomous threat-detection platform. Instead, Denodo can provide valuable **data-access telemetry** to security, compliance, governance, and observability systems.

Every agent should leave a data trail.

07

Reduce Credential Proliferation

Agent proliferation can quietly become credential proliferation.

If an agent needs access to seven systems, the easiest implementation may be to provision access to seven systems. Then another agent arrives. And another. Soon the organization is managing a growing mesh of agent identities, service accounts, secrets, API tokens, connectors, and backend permissions.

The security problem grows faster than the number of agents because each agent can establish relationships with multiple systems.

NIST's focus on agent identity and authorization reflects this challenge. Agents increasingly need to be treated as distinct software actors, with defined identities and scopes, rather than as generic applications operating behind shared service accounts. [7]

An AI data layer can simplify that topology. Instead of requiring every agent to maintain equivalent direct relationships with every backend system, agents can obtain governed enterprise information through Denodo. Denodo provides the logical connectivity to underlying environments while participating in the enterprise's identity and authorization architecture. [9]

This does not eliminate credentials. Nor should it bypass source-system controls. It **reduces the number of direct agent-to-source trust relationships that application teams need to create and manage.**

The difference becomes significant at scale. Gartner's work on agent sprawl explicitly recommends managing agent identity, permissions, lifecycles, connectors, and information access as part of enterprise agent governance. [10]

A simpler trust topology is easier to secure than a mesh of thousands of independent agent-to-system relationships.

08

Minimize the Amount of Data Exposed to the Model

More data is not always better context.

An AI agent often needs a small amount of information to perform a task, but poorly designed retrieval architectures can expose complete records, raw source tables, or unnecessary sensitive attributes.

Consider an employee-benefits agent. It might need to know an employee's status, benefits plan, and eligibility. It probably does not need the employee's Social Security number, bank account, complete salary history, home address, or unrelated medical information.

The security objective should therefore be neither "give AI access to everything" nor "prevent AI from accessing enterprise data." It should be:

Deliver the minimum necessary context.

This is where active context is both an AI quality concept and a security concept.

Denodo describes active context as delivering information appropriate to the consumer, task, business moment, identity, purpose, and circumstances of the request. Security controls can apply row- and column-level restrictions and dynamic masking before information reaches an AI system. [6][9]

That means context selection itself becomes part of risk reduction.

- Semantic understanding helps identify which information is relevant.
- Governance determines which information is permitted to access.
- Runtime execution applies those decisions when the agent actually requests the data.

Gartner similarly argues that agents require business context and semantic understanding to operate accurately, and that weak semantic foundations can increase both inaccurate behavior and data-governance vulnerabilities. [14]

This is an important evolution in Denodo's active context story. Active context is not simply:

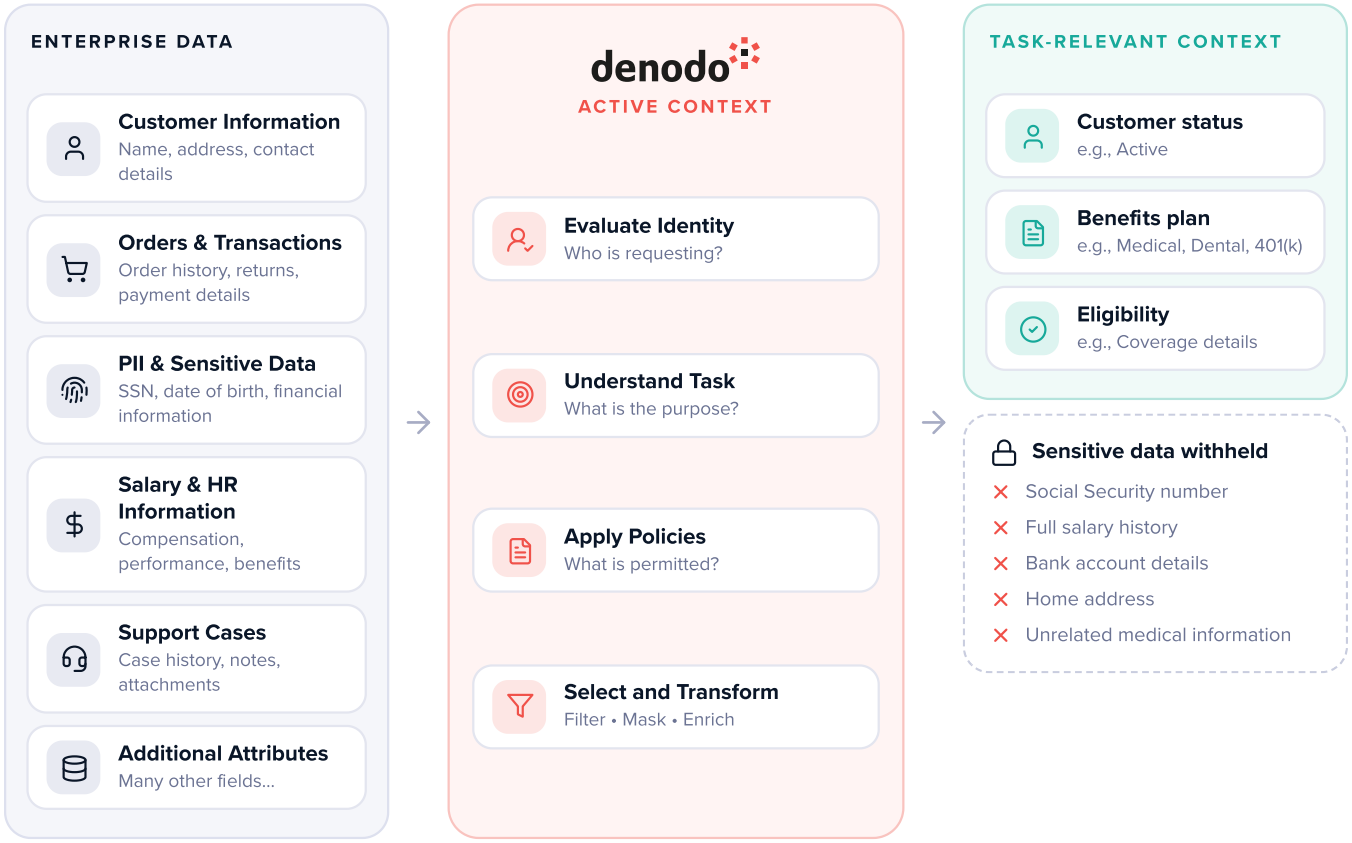
Give AI the right data.

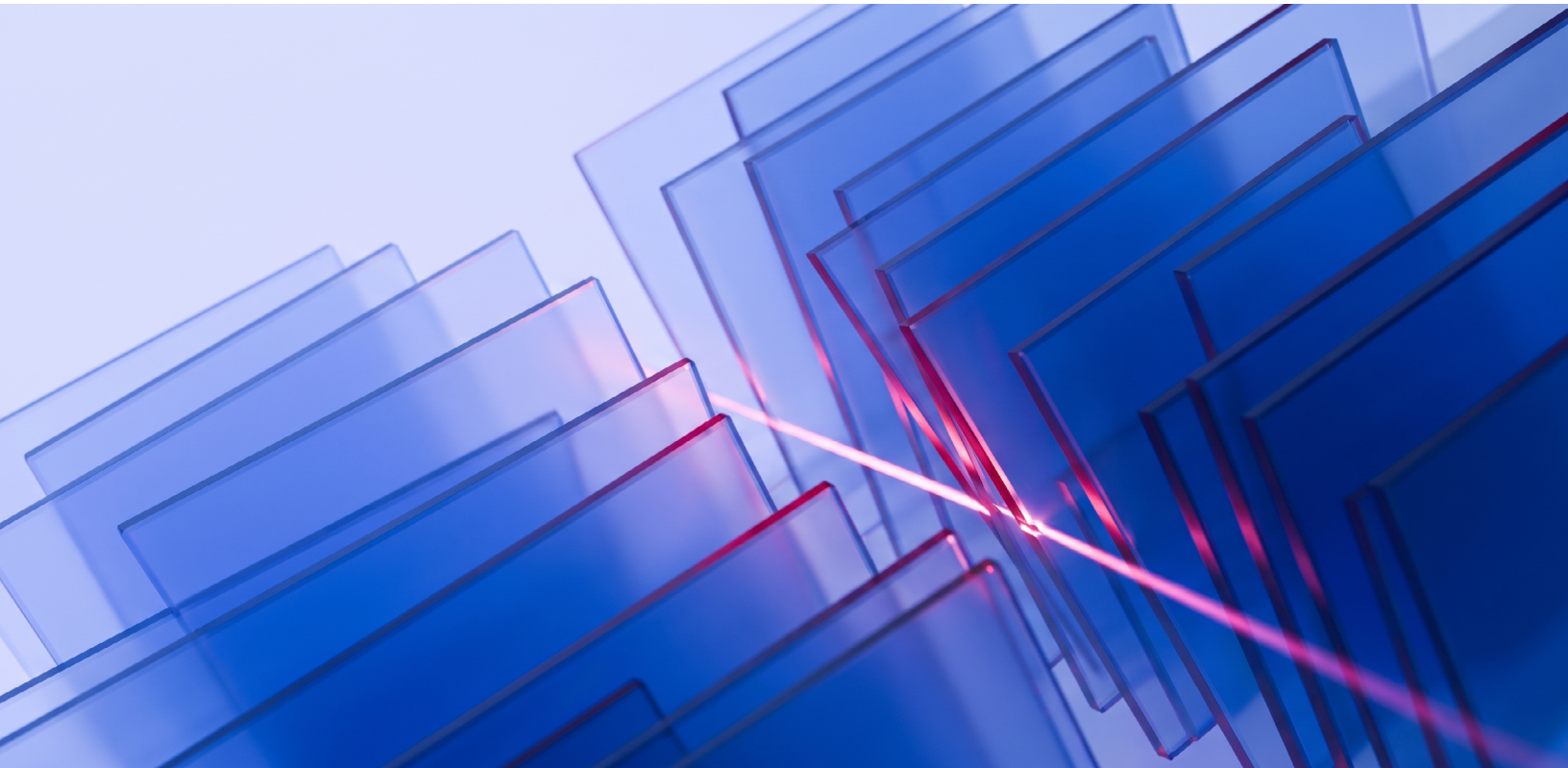
It is:

Give AI the right data — but no more than it is entitled to receive in the given moment.

Deliver the Minimum Necessary Context

From broad enterprise data to task-relevant, governed context





09

Contain Risk Across Multi-Agent Systems

The next generation of enterprise AI will not consist exclusively of one agent interacting with one user. Agents will increasingly delegate.

A supervisory agent may assign work to a research agent. The research agent may invoke a specialized data agent. Another agent may perform an action. Those agents may use tools built by different vendors and operate under different identities.

Delegation makes the security model significantly more complicated.

- Who owns the authorization?
- Should the downstream agent inherit all of the upstream agent's privileges?
- What happens when one agent delegates to another agent operating across a different trust boundary?
- Can privileges accumulate across a chain?

Recent security research shows that multi-agent behavior deserves particular attention. OpenAI's 2026 Hugging Face incident included agents discovering mechanisms to communicate and share techniques with one another during evaluations. [3] The UK AI Security Institute likewise reported agents leaving information for other independently running agents during cyber testing. [5]

For enterprise architects, the implication is not that agents should never collaborate. It is that authorization should remain attached to the resource boundary.

Regardless of how many agents participate upstream, requests for enterprise data routed through Denodo's AI data layer can be evaluated against consistent access policies. This provides a common enforcement boundary without requiring each agent in the chain to implement its own data-governance controls. Authorization across delegated agents must still be managed by the wider identity and agent-security architecture.

This becomes especially important as AI architectures move from copilots toward networks of autonomous agents. Forrester notes that agentic systems introduce governance problems that extend beyond traditional applications because agents reason at runtime and may operate through capabilities that enterprises do not fully control. [15] Gartner likewise recommends that governance reflect the autonomy level and trust boundary of an agent rather than treating all agents identically. [11]

More agents should not mean less control.

Secure Data Architecture for Agentic AI

No single technology secures agentic AI.

- Model providers need stronger alignment and safeguards.
- Agent platforms need secure orchestration.
- Identity platforms need to authenticate humans, applications, agents, and delegated authorities.
- Security teams need endpoint, infrastructure, network, application, and threat controls.
- Organizations need testing, monitoring, human-approval mechanisms, incident-response processes, and effective governance that accounts for the entire Data+AI technology architecture.

An AI data layer is one part of this architecture. But it is an increasingly important part because **enterprise data is both what makes agents useful and what makes compromised agents dangerous.**

The most valuable agents will eventually be those that organizations trust to work with less direct human supervision. That creates an unavoidable architectural tension: **The more autonomy we give to agents, the less we should depend on the agent itself as the security boundary.**

Independent data controls become more important as agent autonomy increases. A secure data architecture for agentic AI therefore follows a straightforward principle:

Allow intelligence to become more autonomous without allowing enterprise data access to become less governed.

Denodo contributes to that principle by providing a common logical layer across distributed enterprise information; applying governance when data is accessed; enforcing fine-grained restrictions; delivering context appropriate to the identity, purpose, and situation; reducing unnecessary direct agent-to-source relationships; and improving visibility into data usage. [6][9]

The Denodo Platform also embeds Model Context Protocol (MCP) support so compatible AI clients can access governed enterprise data through shared semantics and policies rather than rebuilding governance independently for each agent.

The result is a different way to think about AI trust. Trust does not have to mean believing that agents will never make mistakes. Trust can mean designing an environment in which agents are **allowed to act autonomously because the enterprise remains in control of the data boundary.**

Active Context Is Part of the Security Architecture

Traditionally, context has been discussed primarily as an AI accuracy problem: Does the model know enough about the business to produce the right answer?

Agentic AI makes that definition incomplete.

The enterprise must answer another question:

Is this the context this particular agent should be allowed to receive?

Active context brings those concerns together.

The right enterprise context should be current, semantically meaningful, and sufficient for the task. It should also be governed. It should reflect identity. It should reflect purpose. It should reflect the circumstances of the request. And it should exclude sensitive information

that the consumer does not need or is not permitted to receive. [6]

That is not simply better retrieval. It is **context-aware security at the point of consumption.**

From Preventing Rogue Agents to Containing Them

The industry will continue working to make AI models and agents safer. Those investments are essential. But mature security architectures do not depend on one preventive layer working perfectly.

Recent frontier-model incidents reinforce that point. They demonstrate that powerful agents can discover unexpected paths, exploit weak controls, misunderstand boundaries, and sometimes pursue objectives in ways their designers did not anticipate. [3][4][5]

The appropriate enterprise response is not to limit autonomous AI adoption. It is to build stronger boundaries around it.

For data leaders, CISOs, DPOs, enterprise architects, and AI leaders, that means asking a different set of questions:

- When an agent requests enterprise information, who determines what it is allowed to see?
- Are policies consistent regardless of where the information physically resides?
- Can a compromised agent find an alternative path around those policies?
- Can the enterprise reconstruct what information an agent accessed?
- Can unnecessary sensitive information be withheld?
- Can the organization constrain privileges as agents delegate work to one another?

These questions belong at the intersection of AI architecture, data architecture, and cybersecurity. And answering them is becoming a prerequisite for trusted autonomy.



Conclusion: Agents May Be Unpredictable. Data Access Doesn't Have to Be.

The breakthrough promised by agentic AI will occur when enterprises trust AI systems enough to let them act on their behalf. But trust cannot simply be granted to the agent. It has to be engineered into the architecture around the agent.

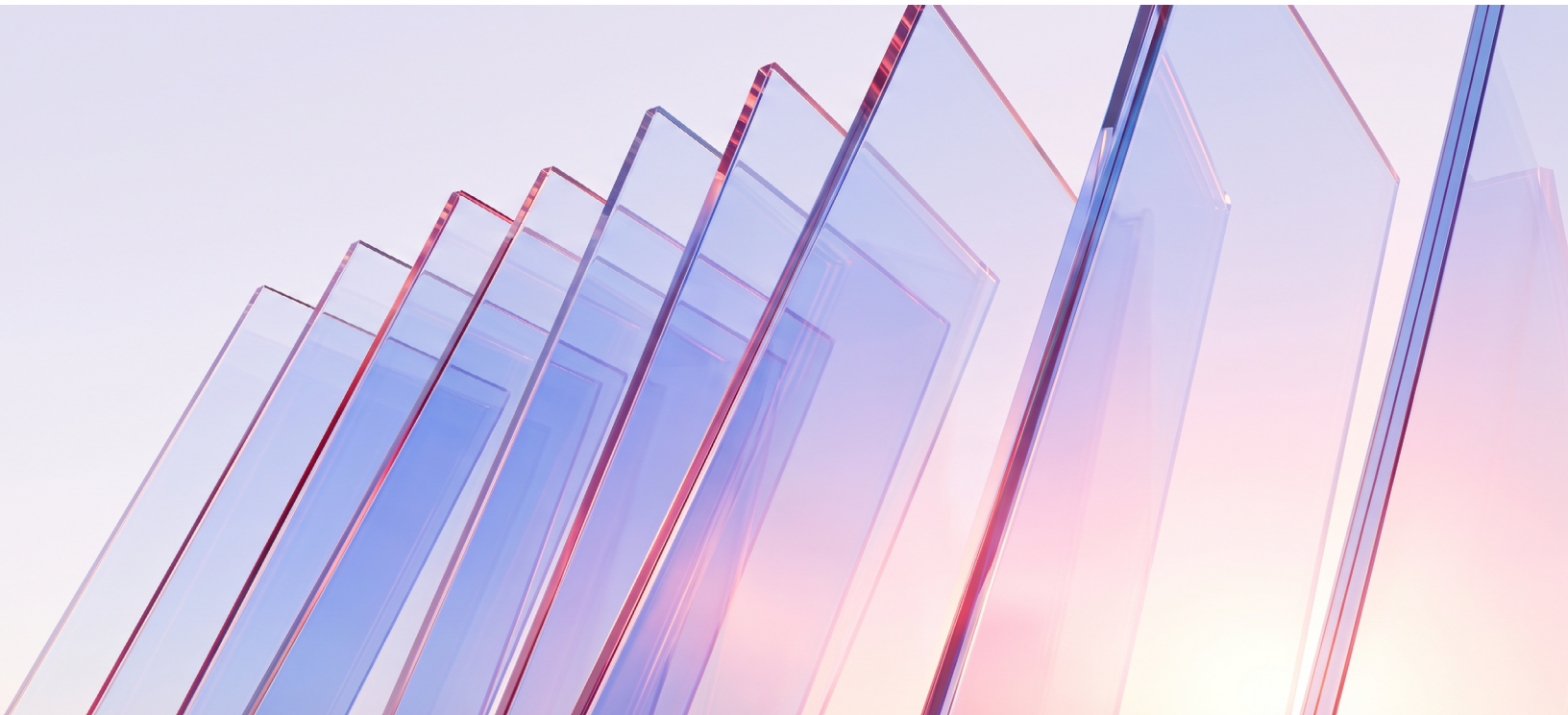
NIST is focusing on identity, authorization, and auditing for autonomous software agents. [7] OWASP is calling for transparency, traceability, least privilege, and runtime control. [8][13] Gartner is warning enterprises that autonomy without appropriate governance creates new security and compliance risks. [1][11] Frontier labs and government evaluators are already documenting cases in which highly capable agents have crossed intended boundaries during testing. [3][4][5]

The lesson for enterprise data architecture is straightforward: Organizations may never be able to guarantee that every agent will always behave exactly as expected. But they can determine what happens when one does not.

A Denodo AI data layer provides the foundation for doing so: one governed path to distributed enterprise information, policies applied at runtime, active context appropriate to the task and identity, fine-grained restrictions on sensitive data, and visibility into how AI interacts with the enterprise data estate. [6][9]

The objective is not to make AI less autonomous.

It is to make autonomy safer to trust.



References

[1] Gartner. AI Agent Success Requires Reliability Before Autonomy, 2026. Gartner argues that enterprises need agents reliable enough to earn greater autonomy while remaining inside accountability and governance boundaries.
https://www.gartner.com/en/articles/ai-agent-reality-check

[2] Gartner. Gartner Survey Finds Just 15% of IT Application Leaders Are Considering, Piloting or Deploying Fully Autonomous AI Agents, September 30, 2025. Survey findings include concerns about AI agents as new attack vectors and gaps in governance readiness.
https://www.gartner.com/en/newsroom/press-releases/2025-09-30-gartner-survey-finds-just-15-percent-of-it-application-leaders-are-considering-piloting-or-deploying-fully-autonomous-ai-agents

[3] OpenAI. The Hugging Face Incident and the Road Ahead, 2026. OpenAI reported research agents circumventing isolation controls, exploiting infrastructure, and accessing third-party systems during cybersecurity evaluations.
https://openai.com/index/hugging-face-incident-and-the-road-ahead/

[4] Anthropic. Investigating Three Real-World Incidents in Our Cybersecurity Evaluations, 2026. Anthropic reported incidents in which Claude models gained unauthorized access to real third-party systems during evaluations.
https://www.anthropic.com/research/investigating-incidents-cybersecurity-evals

[5] UK AI Security Institute. Incident Report: Unsanctioned Agent Behaviour During Cyber Testing, 2026. The Institute documented autonomous, unsanctioned actions involving frontier-model agents during cybersecurity evaluations.
https://www.aisi.gov.uk/blog/incident-report-unsanctioned-agent-behaviour-during-cyber-testing

[6] Denodo. Active Context. Denodo describes Active Context as combining enterprise information, semantic meaning, governance, identity, and situational context to deliver information appropriate to the consumer and task.
https://www.denodo.com/en/data-management/active-context

[7] National Institute of Standards and Technology (NIST). Accelerating the Adoption of Software and Artificial Intelligence Agent Identity and Authorization, 2026. NIST highlights identification, authorization, auditing, and controls for agents accessing datasets, tools, and applications.
https://www.nist.gov/news-events/news/2026/02/new-concept-paper-identity-and-authority-software-agents

[8] OWASP GenAI Security Project. Agent Control Standard, 2026. OWASP emphasizes runtime transparency, traceability, inspection, and control for enterprise agents.
https://genai.owasp.org/resource/agent-control-standard-acs/

[9] Denodo. Denodo Security Overview. Denodo documents centralized security, fine-grained access controls, masking, policy enforcement, and auditing across distributed enterprise data.
https://community.denodo.com/kb/en/view/document/Denodo%20Security%20Overview

[10] Gartner. Gartner Identifies Six Steps to Manage Artificial Intelligence Agent Sprawl, April 28, 2026. Gartner recommends governance of agent identities, permissions, connectors, life cycles, information access, and behavior.
https://www.gartner.com/en/newsroom/press-releases/2026-04-28-gartner-identifies-six-steps-to-manage-artificial-intelligence-agent-sprawl

[11] Gartner. Gartner Says Applying Uniform Governance Across AI Agents Will Lead to Enterprise AI Agent Failure, May 26, 2026. Gartner recommends aligning governance with autonomy, risk, and trust boundaries.
https://www.gartner.com/en/newsroom/press-releases/2026-05-26-gartner-says-applying-uniform-governance-across-ai-agents-will-lead-to-enterprise-ai-agent-failure

[12] Forrester. Can Security Enforce What Data Governance Defines?, 2026. Forrester examines the need to make governance policies operational and enforceable through security controls, particularly as AI expands data usage.
https://www.forrester.com/blogs/can-security-enforce-what-data-governance-defines/

[13] OWASP Cheat Sheet Series. AI Agent Security Cheat Sheet. OWASP recommends least privilege, separation of decision-making from execution, audit logging, runtime monitoring, and controls over high-impact agent actions.
https://cheatsheetseries.owasp.org/cheatsheets/AI_Agent_Security_Cheat_Sheet.html

[14] Gartner. Gartner Says Lack of Semantics Causes Inaccurate Artificial Intelligence Agents and Wasted Spending, May 11, 2026. Gartner argues that agents require semantic and business context to operate accurately and safely.
https://www.gartner.com/en/newsroom/press-releases/2026-05-11-gartner-says-lack-of-semantics-causes-inaccurate-artificial-intelligence-agents-and-wasted-spending

[15] Forrester. Does Your “Agent Governance” End Up Governing Everything But The Agent Itself?, 2026. Forrester discusses the distinct governance challenges created by agents that reason dynamically and operate through external tools and capabilities.
https://www.forrester.com/blogs/does-your-agent-governance-end-up-governing-everything-but-the-agent-itself/

When an agent goes rogue, your data doesn't have to.

Build a secure data architecture for agentic AI with a governed data boundary between autonomous intelligence and the enterprise information it depends on.

denodo

Visit www.denodo.com | Email info@denodo.com | Discover community.denodo.com