Control Surfaces in Enterprise GenAI Security

A Comparative Analysis of Enforcement Boundaries and Operational Tradeoffs

Version: 1.0

Date: January 2026

Classification: Public / Enterprise Buyer Evaluation

Executive Summary

Enterprise adoption of Large Language Models presents a governance challenge that traditional security tools were not designed to address: sensitive data flows through conversational interfaces where the boundary between "input" and "output" is fluid, context-dependent, and often invisible to conventional controls.

This whitepaper examines how different GenAI security solutions define and enforce their **control surfaces**—the boundaries at which security enforcement begins, operates, and ends. Rather than comparing feature counts, this analysis focuses on what each solution's architecture *can* and *cannot* control, and where silent failures may occur despite correct deployment.

The Problem: LLM Adoption Creates New Data Control Risks

When employees use LLMs for enterprise work, sensitive data—customer records, financial information, protected health information—enters systems that may:

- Reside outside enterprise security boundaries (third-party LLM providers)
- Retain data for training or operational purposes
- Expose information through model outputs to unauthorized users
- Lack the granular access controls enterprises require for compliance

Traditional Data Loss Prevention (DLP) tools monitor egress points. Traditional access controls gate entry to systems. Neither addresses the fundamental challenge of LLM workflows: the same data may need to be accessible to some users in some contexts while remaining protected from the same users in other contexts—all within a single conversational interface.

Three Dominant Control Philosophies

This analysis identifies three distinct approaches to GenAI data security, each optimizing for different tradeoffs:

Philosophy	Representative Solution	Core Tradeoff
Lifecycle Tokenization	Protecto	Operational complexity for comprehensive coverage
Zero-Trust Prevention	ZeroTrusted.ai	Workflow degradation for threat prevention
Privacy-by-Removal	Private AI	Loss of data utility for irreversible privacy

A fourth approach—**Governed Access**—is represented by PromptVault, which prioritizes enabling authorized users to work with sensitive data while maintaining audit trails and policy enforcement. This analysis examines PromptVault's beta functionality only; capabilities not documented in the beta release are explicitly excluded.

Why Control Surfaces Matter More Than Features

A security solution may advertise 50 detection capabilities, but if its control surface ends before the LLM processes data, those capabilities cannot prevent exposure at the model layer. Conversely, a solution with fewer features but a control surface that spans the entire prompt-response lifecycle may provide stronger operational security for specific use cases.

This whitepaper equips security leaders to evaluate: *Where does control actually begin and end? What happens when control fails? Can I prove to auditors what was protected?*

Methodology: Control-Surface-First Analysis

Defining Control Surface

A **control surface** is the set of points in a data flow where a security solution can observe, evaluate, and act on data. It encompasses:

- **Entry points**: Where the solution first gains visibility into data
- **Processing scope**: What transformations or evaluations the solution can perform
- **Exit points**: Where the solution's visibility and control ends
- **Feedback mechanisms**: Whether the solution can observe outcomes and adjust

A solution's control surface is distinct from its feature set. Features describe *what* a solution can do; the control surface describes *where* and *when* those capabilities apply.

Why Feature Lists Are Insufficient

Feature comparisons create a false equivalence between solutions with fundamentally different architectures. Consider two hypothetical solutions:

- **Solution A**: Detects 100 PII types, operates only at data ingestion
- **Solution B**: Detects 20 PII types, operates inline during LLM interactions

For an enterprise concerned about sensitive data in LLM prompts, Solution B's smaller feature set may provide stronger protection because its control surface intersects the actual risk point.

Key Terms Used in This Analysis

Control surface: The boundary within which a security solution can observe and act on data flows.

Enforcement boundary: The specific point at which a control decision (allow, deny, transform) is executed. A solution may have visibility beyond its enforcement boundary but cannot act there.

Workflow continuity: The degree to which security controls preserve the user's ability to complete their intended task. High continuity means authorized users experience minimal friction; low continuity means workflows frequently break or degrade.

Silent failure: A failure mode where neither the user nor the system is immediately aware that sensitive data has escaped the control boundary or that enforcement has failed. Characterized by: no alert at the moment of failure, discovery only via forensics or breach notification, and normal workflow continuation despite protection failure.

Audit evidence: The artifacts (logs, records, attestations) a solution produces that can demonstrate to auditors what controls were enforced, when, and with what outcomes.

The Prompt Lifecycle and Enforcement Points

Understanding where controls can exist requires mapping the typical LLM interaction lifecycle.

Lifecycle Stages

1. Pre-Prompt (User Intent → Submission) User formulates a question or task, potentially including sensitive data. At this stage, data exists in the user's context (browser, application, clipboard) but has not yet entered the LLM workflow.

Control opportunities: Client-side detection, input validation, user warnings *Limitations*: Relies on endpoint controls; difficult to enforce centrally

2. In-Flight (Submission → LLM Processing) Data travels from user interface through middleware to the LLM provider. This is the primary opportunity for inline interception, transformation, or blocking.

Control opportunities: Gateway interception, tokenization, sanitization, policy evaluation *Limitations*: Adds latency; requires architectural integration

3. LLM Processing (Model Inference) Data is processed by the LLM. For third-party providers, this occurs outside enterprise control boundaries.

Control opportunities: Limited to provider-side controls (if any) *Limitations*: Enterprise has no direct visibility or control

4. Post-Response (LLM Output → User Delivery) Model output returns through middleware to the user. Opportunity to scan responses for data leakage or policy violations.

Control opportunities: Response scanning, output filtering, detokenization gating
Limitations: Cannot undo what the model has already processed

5. Outside Prompt Flow (Data at Rest, Training, Analytics) Data may be retained by LLM providers, used for model improvement, or stored in enterprise systems for analytics.

Control opportunities: Data retention policies, contractual controls *Limitations*: Often outside technical enforcement; relies on trust

Why Placement Matters More Than Feature Count

A solution that operates at Stage 2 (in-flight) can prevent sensitive data from reaching the LLM. A solution that operates only at Stage 1 (pre-prompt) cannot prevent a determined user from bypassing client-side controls. A solution with no presence at Stage 4 (post-response) cannot detect if the model reconstructed sensitive data from context.

The following sections analyze where each solution's control surface begins and ends.

Why Governed Access Is the Right Model for Enterprise LLM Security

Before examining individual solutions, it's worth asking a more fundamental question: **Which control philosophy is actually right for enterprise LLM workflows?**

The market has converged on three approaches—sanitization, anonymization, and lifecycle tokenization—largely because vendors adapted existing tools for the LLM problem. But each carries structural flaws when applied to conversational AI workflows.

The Flaw in Sanitization (Zero-Trust Prevention)

Sanitization assumes sensitive data should never reach an LLM under any circumstances. This sounds secure, but it misunderstands how enterprises actually use LLMs.

The reality: Employees use LLMs to work with sensitive data—analyzing customer complaints, investigating fraud, drafting responses to clients. Sanitization doesn't distinguish between "analyst investigating fraud with customer SSN" and "attacker exfiltrating customer SSN." Both are blocked.

The consequence: Workflows break. Legitimate work stops. Users either abandon the LLM (wasted investment) or find workarounds (shadow AI, copy-paste into personal ChatGPT). The security tool becomes a productivity obstacle, and the obstacle gets routed around.

The failure mode: Sanitization optimizes for a threat model (exfiltration to untrusted LLMs) that may not match the enterprise's actual deployment. If you're using Azure OpenAI within your tenant, with data processing agreements in place, the "untrusted LLM" threat model is already addressed contractually. Sanitization solves a problem you may not have—at the cost of breaking workflows you definitely need.

The Flaw in Anonymization (Privacy-by-Removal)

Anonymization assumes that if data can't be re-identified, it's safe. This is true for external data sharing—but catastrophically wrong for internal workflows.

The reality: When a compliance officer anonymizes audit data and discovers issues with "Person A," they need to know who Person A actually is to fix the problem. When a fraud analyst anonymizes transaction data and identifies suspicious patterns, they need original account numbers to take action.

The consequence: Anonymization creates a one-way door. Data goes in, anonymized data comes out, and the link is severed. For workflows requiring action on real entities—which is most enterprise workflows—anonymization is a dead end.

The failure mode: Anonymization is the right tool for the wrong job. It's designed for external sharing, research, and statistical analysis. Applying it to internal operational workflows forces users to maintain parallel systems: anonymized data for LLM analysis, original data for action. This defeats the efficiency gains LLMs are supposed to provide.

The Flaw in Lifecycle Tokenization (Data Governance Platforms)

Lifecycle tokenization—Protecto's approach—assumes LLM security is a subset of enterprise data governance. Tokenize data at source, maintain tokens across systems, gate detokenization with sophisticated policies. This sounds comprehensive, but it conflates two different problems.

The reality: Most enterprises adopting LLMs don't need to tokenize data across their entire data lifecycle. They need to protect sensitive data specifically in LLM interactions—a much

narrower problem. Lifecycle tokenization solves the broader problem at the cost of the narrower one.

The consequence: Implementation complexity. Protecto requires integration with databases, APIs, data lakes, event streams, and RAG pipelines—plus policy configuration for purpose-limitation, temporal scoping, and role hierarchies. For an enterprise that just wants to let employees use Copilot safely, this is overengineered.

The failure mode: Lifecycle tokenization optimizes for enterprises with mature data governance programs and dedicated teams to manage policy complexity. For the majority of enterprises—those with simpler needs and smaller security teams—it's the right solution at the wrong scale.

The Case for Governed Access

Governed access takes a different stance: **the goal is not to prevent all access, but to ensure the right people access the right data with the right audit trail.**

This philosophy accepts three premises that other approaches reject:

1. **Authorized users need sensitive data to do their jobs.** Fraud analysts need SSNs. Compliance officers need transaction records. Customer support needs account details. A security model that blocks authorized access is a failed security model.
2. **The LLM prompt layer is the right enforcement point.** Enterprises don't need to tokenize data across their entire lifecycle—they need to control what enters LLM workflows specifically. Prompt-layer enforcement is sufficient for LLM-specific risks.
3. **Workflow continuity is a security requirement, not a nice-to-have.** When security tools break workflows, users find workarounds. Shadow AI is a bigger risk than controlled access with audit trails.

Governed access in practice:

- Sensitive data is tokenized before reaching the LLM (protection)
- Authorized roles can detokenize when needed (workflow continuity)
- All access is logged (audit trail)
- Unauthorized users see tokens, not data (enforcement)

This is not a weaker security model—it's a *right-sized* security model for the actual enterprise LLM threat.

The Enterprise LLM Threat Model: What Are You Actually Protecting Against?

Different solutions are optimized for different threats. Choosing the right solution requires understanding which threat model matches your reality.

Threat Model A: Malicious Exfiltration to Untrusted LLMs

"Attackers or malicious insiders will deliberately send sensitive data to external LLMs to exfiltrate it."

Best-fit solution: ZeroTrusted.ai (sanitization)

Why: If you assume hostile intent and untrusted destinations, blocking all sensitive data makes sense. Workflow degradation is acceptable because you're preventing deliberate harm.

When this model applies: Organizations using public LLMs (consumer ChatGPT) with no data processing agreements. Government/defense contexts with classified data. High-threat environments with known insider risk.

When this model doesn't apply: Organizations using enterprise LLM deployments (Azure OpenAI, Anthropic API with DPA) where the "untrusted destination" concern is already addressed contractually.

Threat Model B: External Data Sharing Risk

"We need to share data with external parties (auditors, researchers, partners) without exposing PII."

Best-fit solution: Private AI (anonymization)

Why: If data leaves your organization permanently, irreversible anonymization is the appropriate control. You don't need to re-identify; you need to ensure others can't identify.

When this model applies: Healthcare research data sharing. Academic collaborations. Third-party audits where auditors don't need individual-level data.

When this model doesn't apply: Internal workflows where the same team needs to act on findings. Any use case requiring correlation back to original entities.

Threat Model C: Enterprise Data Lifecycle Governance

"We need unified tokenization across all systems—databases, APIs, data lakes, LLMs—with sophisticated policy controls."

Best-fit solution: Protecto (lifecycle tokenization)

Why: If you're solving enterprise-wide data governance and LLMs are one application among many, a comprehensive platform makes sense.

When this model applies: Fortune 500 enterprises with mature data governance programs. Organizations with dedicated data protection teams. Multi-system environments requiring consistent tokenization.

When this model doesn't apply: Organizations whose primary concern is LLM-specific. Enterprises without existing data governance infrastructure. Teams that need fast deployment without extensive policy configuration.

Threat Model D: Accidental Exposure in LLM Workflows

"Employees will accidentally include sensitive data in LLM prompts. We need to protect against mistakes without blocking legitimate work."

Best-fit solution: PromptVault (governed access)

Why: The threat is accidental, not malicious. The goal is protection with workflow continuity. Users are mostly trusted; the system catches mistakes and enforces access control.

When this model applies: Enterprises deploying LLMs to productivity workers (not just developers). Organizations using enterprise LLM providers with data processing agreements. Microsoft-centric environments with existing Entra ID/Purview infrastructure. Teams that need fast deployment and simple administration.

Matching Threat Model to Solution

If your primary threat is	The right control philosophy is	Because

Deliberate exfiltration to untrusted LLMs	Sanitization (ZeroTrusted.ai)	Block everything; accept workflow loss
External sharing of sensitive datasets	Anonymization (Private AI)	Irreversible de-identification; no retrieval needed
Enterprise-wide data lifecycle risk	Lifecycle tokenization (Protecto)	Comprehensive coverage; accept complexity
Accidental exposure in LLM workflows	Governed access (PromptVault)	Right-sized protection; preserve workflows

For this threat model, sanitization is overkill (breaks workflows unnecessarily), anonymization is wrong-fit (users need to act on real data), and lifecycle tokenization is overengineered (solves a bigger problem than you have).

Governed access is the right-sized solution: protect against accidents, enable authorized access, maintain audit trails, deploy quickly.

Comparative Control Models

Protecto: Lifecycle Tokenization

Control Philosophy

Protecto approaches data security as a lifecycle problem: sensitive data should be tokenized at its source and remain tokenized throughout its journey across systems, including LLM interactions. The philosophy is *govern data everywhere it travels* rather than *protect data at specific chokepoints*.

Where Control Starts and Ends

Entry Points (per vendor documentation):

- Database connectors (tokenize sensitive columns at source)
- API gateways (intercept payloads before application processing)
- Event streams (pre-process streaming data)

- RAG pipelines (tokenize documents before embedding)
- LLM prompts (SDK or API integration)

Exit Points:

- Detokenization gate (authorized users retrieve original values)
- Data deletion (token mapping removal makes tokens meaningless)

Control ends when detokenized data is delivered to an authorized user or application. Post-delivery use is outside Protecto's visibility.

Enforcement Type

Tokenization (reversible): Sensitive values replaced with tokens that map to originals in an encrypted vault. Deterministic tokens enable joins and analytics; non-deterministic tokens provide higher entropy.

Detokenization (policy-gated): Reversal requires role authorization, purpose justification, and temporal validity. Short-lived approvals (hours, not days) per documentation.

Impact on Workflow Continuity

High continuity for authorized users: Workflows proceed with tokens in transit; authorized users see original values when policy permits.

Explicit denial for unauthorized users: Users without authorization see tokens, not sensitive values. Denial is visible (user knows access was blocked).

Audit Evidence Produced

- Tokenization events: element detected, token generated, timestamp, source system
- Detokenization events: requesting user, role, purpose claimed, approval decision, timestamp
- Policy decisions: rule evaluated, parameters, outcome, rationale

Audit trail is documented as immutable and tamper-evident.

Control Gaps That Remain

Even when correctly deployed, Protecto does not control:

- **Vault compromise**: If the token vault is breached, all tokens become reversible by the attacker
- **Key compromise**: If encryption keys are stolen, all tokenized data is exposed
- **LLM provider security**: Once data reaches the LLM (even tokenized), it is subject to the provider's controls
- **Post-detokenization use**: After authorized delivery, data use is outside Protecto's visibility
- **Policy misconfiguration**: Over-broad policies (e.g., "all staff can detokenize SSN") render controls ineffective

ZeroTrusted.ai: Zero-Trust Prevention

Control Philosophy

ZeroTrusted.ai operates on the assumption that any data sent to an LLM may be compromised. The philosophy is *prevent sensitive data from ever reaching untrusted systems* rather than *enable controlled access*. This is a binary model: sanitize or block.

Where Control Starts and Ends

Entry Points (per vendor documentation):

- Web interface (vendor-provided ChatGPT-like UI)
- API integration (developers route prompts through ZeroTrusted.ai)
- LangChain integration (middleware intercept)
- Zapier integration (workflow automation)

Exit Points:

- Sanitized prompt delivery to LLM
- Optional response scanning before user delivery

Control ends at sanitization. Once data is masked, redacted, or replaced, the original is discarded. There is no retrieval mechanism.

Enforcement Type

Sanitization (irreversible): Sensitive data is masked ("[REDACTED]"), replaced with generic values ("John Doe"), or removed entirely.

Blocking: Prompts containing injection attempts, jailbreak patterns, or anomalous content are denied entirely.

Detection-only modes: Some deployments flag rather than block, but the dominant model is prevention.

Impact on Workflow Continuity

Degraded continuity: When sensitive data is sanitized, context is lost. The LLM receives incomplete information and may produce less useful or incorrect responses.

Binary outcomes: Prompts are either sanitized (workflow proceeds with reduced context) or blocked (workflow stops entirely). There is no "allow authorized access" mode.

False positives impact legitimate users: Unusual but legitimate prompts may trigger blocking or excessive sanitization.

Audit Evidence Produced

- Detection events: PII type identified, pattern matched, action taken
- Sanitization events: original prompt (hashed or omitted), sanitized version, transformations applied
- Injection/jailbreak events: pattern detected, user, action taken
- Anomaly events: score, confidence, action

Audit shows what was sanitized but typically does not retain original prompts (for privacy reasons).

Control Gaps That Remain

Even when correctly deployed, ZeroTrusted.ai does not control:

- **Authorized access**: No mechanism for authorized users to bypass sanitization for legitimate purposes
- **Detection false negatives**: Undetected sensitive data passes through silently

- **Model reconstruction**: Sophisticated LLMs may infer sensitive data from remaining context
- **Workflow degradation**: Legitimate work requiring sensitive data cannot proceed
- **LLM provider security**: Once sanitized data reaches the LLM, provider controls apply

Private AI: Privacy-by-Removal

Control Philosophy

Private AI prioritizes irreversible de-identification. The philosophy is *eliminate identifiability entirely* rather than *control access to identifiable data*. Once anonymized, data cannot be re-identified even by authorized users.

Where Control Starts and Ends

Entry Points (per vendor documentation):

- Text input (API, direct paste)
- Image upload (OCR + NLP processing)
- Audio upload (transcription + NLP)
- PDF upload (document processing)
- Batch processing (bulk document sets)

Exit Points:

- Anonymized data output
- No detokenization or re-identification pathway (by design)

Control ends at anonymization. The original data is not retained; the mapping is not preserved.

Enforcement Type

Anonymization (irreversible): Multiple methods documented— masking (replace with placeholder), synthetic replacement (realistic fake data), generalization (convert to ranges), pseudonymization (consistent codes).

Detection spans 50+ PII entity types across 49+ languages, including multi-format support (text, images, audio, PDFs).

Impact on Workflow Continuity

Reduced but functional continuity: Users work with anonymized data. Workflows proceed, but with altered context and reduced fidelity.

No retrieval path: Even authorized users cannot recover original values. Re-identification requires returning to original (non-anonymized) source data.

External sharing enabled: Anonymized data can be shared with auditors, researchers, or partners without privacy risk.

Audit Evidence Produced

- Anonymization events: batch processed, entity types detected, methods applied
- Detection confidence: per-entity scores
- No detokenization events (none occur)

Audit shows what was anonymized but cannot prove completeness (false negatives are silent).

Control Gaps That Remain

Even when correctly deployed, Private AI does not control:

- **Authorized retrieval**: Anonymization is permanent; no role-based access to original data
- **Detection false negatives**: Undetected PII remains in "anonymized" output
- **Residual identifiability**: Sufficient context may allow statistical re-identification
- **External linkage attacks**: Anonymized data combined with external datasets may enable re-identification
- **Data utility loss**: Anonymized data has reduced value for workflows requiring precise information

PromptVault: Governed Access (Beta)

Control Philosophy

PromptVault positions between prevention-only and retrieval-enabled models, prioritizing *controlled access with audit trails* for enterprises using Microsoft identity infrastructure. The philosophy is *enable authorized work while maintaining governance*.

Note: This section reflects only capabilities documented in the PromptVault beta release. Roadmap items and inferred capabilities are excluded.

Where Control Starts and Ends (Beta)

Entry Points (documented):

- Inline interception of user prompts before LLM delivery
- Source data processing (documents for RAG)
- Browser plugin (deployed via Microsoft Intune)
- Microsoft Teams/Copilot extension

Exit Points:

- Detokenization gate (RBAC-controlled)
- Audit logging (all events captured)

Control ends when detokenized data is delivered to an authorized user through an approved context.

Enforcement Type (Beta)

Detection: Layered approach combining NLP-based entity recognition, rule-based pattern matching, and context-aware ML models. Applied to source data and user prompts.

PromptVault treats sensitivity as customer-defined rather than vendor-assumed. Enterprises configure detection for the data types that matter to their business—including proprietary identifiers, internal codes, and domain-specific entities that generic PII libraries typically miss. This design philosophy acknowledges that "sensitive" varies by industry, regulatory context, and business function. The tradeoff is upfront configuration effort; enterprises must define their sensitivity types before detection operates.

Two configuration models are supported:

- Without Purview: Administrators upload a CSV defining sensitivity information types, which become detection "Elements" within PromptVault
- With Purview: Sensitivity information types are automatically retrieved from Microsoft Purview, eliminating manual definition for organizations with existing classification infrastructure

Tokenization (reversible): Detected sensitive data is tokenized immediately. Raw sensitive data is never stored in plaintext. Tokenized values secured in encrypted storage with Azure Key Vault managing encryption keys.

Detokenization (RBAC-gated): Access requires role authorization. Roles are derived from Microsoft Entra ID groups and mapped to Elements (sensitivity types). When a user requests access to tokenized data, PromptVault evaluates whether their Entra ID group membership grants access to the relevant Elements.

Current beta limitations: Detokenization is role-based only. Purpose-limitation (e.g., "fraud investigation" vs. "customer support" as separate approval contexts) and temporal scoping (time-bound approvals) are not currently supported. All detokenization events are audited, though audit logs record the approval decision rather than the specific policy rule that triggered approval.

Impact on Workflow Continuity (Beta)

High continuity for authorized users: Authorized roles see original values; unauthorized users see tokens.

Microsoft ecosystem integration: Single sign-on via Entra ID; no additional authentication layers.

Policy-driven outcomes: Access decisions based on role-element mappings configured by administrators.

Deployment constraints: Browser plugin deployment requires Microsoft Intune-managed devices (Entra ID-joined or Hybrid Entra ID-joined). BYOD and contractor access scenarios are not currently supported. Microsoft Teams/Copilot extension provides an additional access path within the Microsoft ecosystem.

Audit Evidence Produced (Beta)

- Tokenization and detokenization events (what was tokenized/detokenized, when, by whom)
- Prompt execution records
- Role-based access decisions (approval or denial)
- Integration capability with SIEM tools

Current limitation: Audit logs record that detokenization was approved or denied, but do not capture the specific policy rule or rationale that triggered the decision. Log retention periods are not yet defined in beta documentation.

Logs support search, filtering, and export.

What Is Outside Beta Scope

The following capabilities are outside the current beta scope. They are documented here for transparency with evaluators conducting due diligence:

Capability	Beta Status	Context
On-premises deployment	Not in beta	Cloud-native architecture; Azure-hosted
Multi-format support (images, audio, PDFs)	Not in beta	Text processing only
Detection accuracy benchmarks	Testing in progress	No published metrics yet
DSAR workflow	Not in beta	No published playbook
Multi-language support	Not specified	Language coverage not documented
Purpose-limited detokenization	Not in beta	Role-based only; purpose-scoping on roadmap
Time-bound access approvals	Not in beta	Temporal limits on roadmap
BYOD/contractor support	Not in beta	Intune-managed devices only
Policy rule audit logging	Not in beta	Logs show decision outcome, not rule rationale
Audit log retention periods	Not defined	Retention policy pending

Buyers requiring these capabilities today should evaluate Protecto (purpose-scoping, temporal limits, on-premises, multi-cloud) or Private AI (multi-format support) depending on their primary use case.

Control Gaps That Remain (Beta)

Even when correctly deployed, PromptVault does not control:

- **Policy misconfiguration**: Over-broad role assignments render controls ineffective
- **Post-detokenization use**: After authorized delivery, data use is outside visibility
- **LLM provider security**: Data sent to LLMs is subject to provider controls
- **Identity system compromise**: If Entra ID is compromised, role assignments may be bypassed

Where PromptVault Is Uniquely Positioned

Despite beta-stage limitations, PromptVault occupies a distinct position in the market that no competitor currently addresses:

1. Native Microsoft Identity Integration

Protecto, ZeroTrusted.ai, and Private AI require separate identity mapping—either custom IDAM integration or manual role configuration. PromptVault consumes Entra ID groups directly. For enterprises where Entra ID already governs Microsoft 365, Azure resources, and line-of-business applications, this eliminates an entire integration layer.

Implication: Time-to-value for Microsoft-centric enterprises is significantly faster. No identity infrastructure to build; existing group memberships translate directly to PromptVault access controls.

2. Purview Alignment for Existing Classification Investments

Organizations using Microsoft Purview for data classification have already defined sensitivity information types and labels. PromptVault ingests these definitions automatically—Purview sensitivity labels become PromptVault roles; sensitivity information types become detection Elements.

Implication: Enterprises with mature Purview deployments achieve consistency between data classification policy and LLM access control without manual synchronization. Configuration drift between systems is eliminated.

3. Customer-Defined Sensitivity (Intentional Design)

Where competitors ship generic PII libraries (SSN, credit card, email), PromptVault treats sensitivity as customer-defined. This requires upfront configuration but solves a real problem: generic libraries miss industry-specific sensitive data (internal customer IDs, proprietary codes, domain terminology) while flagging data that isn't sensitive in a given context.

Implication: Higher detection precision for organizations with non-standard sensitive data. Lower false-positive rates for organizations where generic PII patterns appear in non-sensitive contexts.

4. Reversible Tokenization with Workflow Continuity

Unlike ZeroTrusted.ai (sanitization destroys data) and Private AI (anonymization is irreversible), PromptVault's tokenization preserves the ability for authorized users to access original values. Unlike Protecto's data-lifecycle approach, PromptVault operates inline at the prompt layer—purpose-built for LLM workflows rather than adapted from broader data governance.

Implication: Authorized users experience workflow continuity; unauthorized users see tokens. The solution is optimized for conversational AI patterns, not retrofitted from traditional data protection.

Buyer fit summary: PromptVault is the strongest fit for Microsoft-centric enterprises that need reversible tokenization, already use (or plan to use) Entra ID and Purview, and can operate within managed-device deployment constraints. Organizations requiring multi-cloud parity, on-premises deployment, or advanced governance controls (purpose-scoping, temporal limits) may find Protecto a better fit today.

User Journey Comparisons

The following scenarios illustrate how each control model behaves when processing identical enterprise prompts. These are textual walkthroughs based on documented architectures, not UI assumptions.

Scenario: Fraud Analyst Investigating Customer Complaint

Context: An analyst in a financial services firm investigates a potential fraud case. The investigation requires accessing customer SSN, account details, and transaction history through an LLM-assisted workflow.

Prompt submitted: "Analyze the payment history for customer John Smith, SSN 123-45-6789, email john@example.com. What patterns suggest fraud?"

Protecto Response:

1. Prompt intercepted at API gateway
2. Detection identifies: name (John Smith), SSN (123-45-6789), email (john@example.com)
3. Elements tokenized: `tok_name_abc`, `tok_ssn_def`, `tok_email_ghi`
4. Tokenized prompt sent to LLM
5. LLM requests payment data via tool call; tool call intercepted
6. Detokenization gate evaluates: analyst role authorized for SSN? Yes. Purpose (fraud investigation) valid? Yes.
7. Original SSN provided to payment system; history retrieved
8. LLM generates analysis; analyst receives complete response
9. Audit trail records: detection event, tokenization event, detokenization approval, role, purpose, timestamp

Outcome: Workflow completes. Analyst sees full analysis. Sensitive data protected in transit. Access logged.

ZeroTrusted.ai Response:

1. Prompt intercepted at gateway
2. Detection identifies: name (PII), SSN (PII)
3. Sanitization applied: "John Smith" → "John Doe"; SSN → "[REDACTED]"
4. Sanitized prompt sent to LLM: "Analyze the payment history for customer John Doe, SSN [REDACTED]..."
5. LLM cannot retrieve actual payment history (no valid SSN)
6. LLM responds with generic fraud indicators, not customer-specific analysis
7. Analyst receives partial response; must look up customer data separately to correlate

Outcome: Workflow degraded. Analyst receives generic guidance, not specific investigation support. Sensitive data prevented from reaching LLM. No authorized access pathway exists.

Private AI Response:

1. Prompt processed for anonymization
2. Detection identifies: name, SSN, email
3. Anonymization applied: "John Smith" → "Margaret Chen"; SSN → synthetic SSN; email → synthetic email
4. Anonymized prompt sent to LLM
5. LLM analyzes "Margaret Chen's" payment patterns (using synthetic identifiers)
6. Response returned with anonymized identifiers
7. Analyst cannot correlate response to actual customer without returning to original data source

Outcome: Workflow proceeds with anonymized data. Analyst must manually correlate findings to real customer. No re-identification possible through Private AI.

PromptVault Response (Beta):

1. Prompt intercepted inline before LLM
2. Detection identifies: name, SSN, email (via NLP + pattern + ML layers)
3. Elements tokenized; tokens stored in encrypted vault
4. Tokenized prompt sent to LLM
5. When response requires customer-specific data, detokenization evaluated
6. RBAC check: analyst's Entra ID group mapped to role with SSN access? (Depends on configuration)
7. If authorized: original values provided; workflow completes with full context
8. If not authorized: tokens remain; analyst sees limited response
9. All events logged for audit

Outcome: Workflow continuity depends on role configuration. Authorized analysts see full data; unauthorized users see tokens. All access audited.

Scenario: Unauthorized Access Attempt

Context: A junior customer service representative attempts to retrieve SSN for a customer, despite not being authorized for SSN access.

Prompt submitted: "What is the SSN for customer John Smith?"

Protecto: SSN tokenized; CSR receives token string, not SSN. Escalation path exists (manager approval for time-limited access). Denial is explicit.

ZeroTrusted.ai: SSN sanitized to "[REDACTED]"; CSR receives "[REDACTED]". No escalation path; binary outcome.

Private AI: SSN anonymized; CSR receives synthetic SSN (unusable). No retrieval path.

PromptVault (Beta): SSN tokenized; CSR sees token. If role not mapped to SSN element, detokenization denied. Denial logged.

Risk and Threat Implications

Threat Mitigation Comparison

Threat	Protecto	ZeroTrusted.ai	Private AI	PromptVault (Beta)
Data exfiltration (plaintext)	Mitigated (data tokenized in transit)	Mitigated (data sanitized before LLM)	Mitigated (data anonymized)	Mitigated (data tokenized)
Unauthorized access	Mitigated (RBAC detokenization)	Partially (binary sanitization for all)	N/A (no retrieval exists)	Mitigated (RBAC detokenization)
Insider misuse	Partially (audit trail; doesn't prevent authorized access)	Partially (sanitization applies to all)	Partially (anonymization prevents access)	Partially (audit trail; doesn't prevent authorized access)

Over-blocking / DoS	Low risk (authorized users proceed)	High risk (false positives block legitimate work)	Moderate (anonymization reduces utility)	Low risk (authorized users proceed)
Audit failure	Low (comprehensive logging)	Moderate (original prompts may not be retained)	Moderate (no detokenization events)	Low (comprehensive logging documented)

Failure Mode Visibility

Solution	Detection False Negative	Vault/Key Compromise	Policy Misconfiguration
Protecto	Silent (undetected data passes through)	Initially silent; discovered via forensics	Detectable via audit review
ZeroTrusted.ai	Silent (undetected data reaches LLM)	N/A (no vault)	N/A (no RBAC)
Private AI	Silent (PII remains in "anonymized" output)	N/A (no vault)	N/A (no RBAC)
PromptVault (Beta)	Silent (undetected data passes through)	Not documented	Detectable via audit review

Critical Risk: Silent Failures

All solutions share one fundamental limitation: detection false negatives are silent. If sensitive data is not recognized by the detection layer, it passes through without protection, without logging, and without visibility. No audit trail exists for data that was never detected.

This makes detection accuracy a critical—but often undisclosed—variable in security posture. Buyers should request false-negative rates and red-team testing results.

Auditability and Compliance Evidence

Evidence Quality Comparison

Dimension	Protecto	ZeroTrusted.ai	Private AI	PromptVault (Beta)
Tokenization events	Comprehensive	N/A (sanitization instead)	N/A (anonymization instead)	Documented
Detokenization events	Comprehensive (user, role, purpose, time)	N/A (no retrieval)	N/A (no retrieval)	Documented
Original prompt retention	Not specified	Often omitted (privacy)	Not retained	Not specified
Policy decision logging	Documented	Documented	Limited	Documented
SIEM integration	Documented	Not publicly documented	Not publicly documented	Documented
DSAR support	Documented (token deletion)	Not publicly documented	Impossible (irreversible)	Not documented in beta

What Can Be Proven to Auditors

Protecto: "User X with role Y requested access to element Z for purpose P at time T. Decision was ALLOW/DENY."

ZeroTrusted.ai: "Prompt from user X was sanitized at time T. Elements [types] were masked/redacted."

Private AI: "Data batch X was anonymized at time T. [N] entities of types [types] were transformed."

PromptVault (Beta): "User X with Entra ID role Y accessed element Z at time T. Tokenization and detokenization events recorded." (Note: specific policy rule rationale not captured in current beta; logs show decision outcome only.)

Evidence Gaps

All solutions share a critical evidence gap: **proving completeness**. Audit logs show what was detected and controlled but cannot prove that all sensitive data was detected. Undetected data leaves no audit trail.

Buyer Decision Matrix

Buyer Profile	Primary Concern	Best-Fit Philosophy	Recommended Solutions	When Insufficient
Healthcare (HIPAA)	PHI protection + authorized clinical access	Governed Access	Protecto, PromptVault	ZeroTrusted.ai (blocks clinical workflows); Private AI (no retrieval for patient care)
Financial Services (SOX, PCI)	Transaction data protection + fraud investigation access	Governed Access	Protecto, PromptVault	ZeroTrusted.ai (blocks investigation workflows); Private AI (no retrieval for audits)
Research/Academic	Data sharing with external parties	Privacy-by-Removal	Private AI	Protecto, PromptVault (require vault management for external sharing)
Government/Defense	Zero exposure to third-party LLMs	Zero-Trust Prevention	ZeroTrusted.ai, on-prem solutions	PromptVault beta (cloud-native only)
Microsoft-centric Enterprise	Entra ID integration + Purview alignment	Governed Access	PromptVault	Protecto (requires separate

				identity mapping)
Multi-cloud Enterprise	Vendor-neutral deployment	Lifecycle Governance	Protecto	PromptVault (Azure-first)

When Each Solution Is Sufficient

Protecto is sufficient when: Organization requires comprehensive data lifecycle governance across multiple systems, has mature policy management capabilities, and can absorb integration complexity.

ZeroTrusted.ai is sufficient when: Organization uses untrusted/public LLMs, accepts workflow degradation as acceptable cost, and does not require role-based access to sensitive data.

Private AI is sufficient when: Organization shares data externally, prioritizes irreversible privacy, and does not require internal authorized retrieval.

PromptVault (Beta) is sufficient when: Organization uses Microsoft identity infrastructure, requires role-based detokenization, operates with Intune-managed devices, and can accept current beta limitations (no purpose-scoping, no temporal limits, no BYOD support).

When Each Solution Is NOT Sufficient

Protecto is not sufficient when: Organization needs rapid deployment, has no existing data governance infrastructure, or requires simple point-solution protection.

ZeroTrusted.ai is not sufficient when: Legitimate workflows require sensitive data access, false positives are unacceptable, or role-based governance is required.

Private AI is not sufficient when: Authorized internal users need access to original data, re-identification is required for compliance, or data utility must be preserved.

PromptVault (Beta) is not sufficient when: Organization requires on-premises deployment, multi-format (image/audio/PDF) processing, non-Microsoft identity systems, BYOD/contractor access, purpose-scoped approvals, or time-bound access controls.

Conclusion

Control Boundaries Define Security Posture

The central finding of this analysis is that GenAI security solutions differ more fundamentally in *where their control surfaces end* than in *what features they offer*. A solution with extensive detection capabilities but a control surface that ends before the LLM provides different protection than a solution with fewer features but enforcement that spans the full prompt-response lifecycle.

Different Solutions Optimize for Different Threats

No solution examined eliminates all risks. But the more important insight is that each solution optimizes for a *different threat model*:

- **ZeroTrusted.ai** optimizes malicious exfiltration to untrusted LLMs—a real threat, but not the primary threat most enterprises face when deploying managed LLM services with data processing agreements.
- **Private AI** optimizes external data sharing—the right solution when data must leave the organization, but the wrong solution when users need to act on findings.
- **Protecto** optimizes enterprise-wide data lifecycle governance—comprehensive but overengineered for organizations whose concern is specifically LLM workflows.
- **PromptVault** optimizes accidental exposure in LLM workflows with authorized access for legitimate work—the threat model that matches most enterprise LLM deployments today.

The Right-Sized Security Argument

Many enterprises adopting LLMs in 2026 are deploying productivity tools: Copilot, internal GPT interfaces, AI-assisted workflows for knowledge workers. The users are not attackers. The LLMs are not untrusted. The primary risk is accidental exposure, not deliberate exfiltration.

For this threat model:

- **Sanitization is overkill**: It blocks legitimate work to prevent a threat (untrusted LLM exfiltration) that's already addressed by enterprise LLM contracts.

- **Anonymization is wrong-fit**: It destroys the link between analysis and action that makes LLMs useful for operational work.
- **Lifecycle tokenization is overengineered**: It solves enterprise-wide data governance when the need is prompt-layer protection.

Governed access—tokenization with RBAC-gated detokenization and audit trails—is the right-sized solution for the actual enterprise LLM threat.

Recommendations for Evaluation

Security leaders evaluating GenAI protection should:

1. **Identify your actual threat model**: Are you protecting against malicious exfiltration, external sharing, lifecycle governance gaps, or accidental exposure? The answer determines the right control philosophy.
2. **Match philosophy to threat**: Don't buy a data lifecycle platform when you need prompt-layer protection. Don't buy sanitization when your users need to work with sensitive data.
3. **Evaluate control surfaces, not features**: Where does each solution's enforcement actually begin and end? Features that don't apply at your risk points don't provide protection at your risk points.
4. **Assess workflow impact honestly**: Will security controls break legitimate work? If so, users will find workarounds. Shadow AI is a bigger risk than controlled access.
5. **Verify audit evidence quality**: What can actually be proven to regulators and auditors? Audit trails that don't exist can't demonstrate compliance.

Final Note on PromptVault

PromptVault is purpose-built for enterprises whose primary concern is protecting sensitive data in LLM workflows—specifically, Microsoft-centric enterprises that want fast deployment, native Entra ID integration, and workflow continuity for authorized users.

If your threat model is malicious exfiltration to untrusted LLMs, evaluate ZeroTrusted.ai. If your need is external data sharing, evaluate Private AI. If you require enterprise-wide data lifecycle governance, evaluate Protecto.

But if your reality is knowledge workers using Copilot and Azure OpenAI, and your goal is protecting against accidental exposure without breaking their workflows—PromptVault is the right-sized solution.

Appendix: PromptVault Beta Scope Summary

This analysis reflects PromptVault's documented beta functionality as of January 2026. The table below summarizes what is in scope, what is outside current scope, and where PromptVault offers unique positioning.

In Beta Scope (Verified Capabilities)

Capability	Description
Inline prompt interception	Detection and tokenization before LLM delivery
Customer-defined sensitivity	Enterprises configure detection for their specific data types
NLP + rule + ML detection	Layered detection approach
Reversible tokenization	Encrypted vault storage; Azure Key Vault integration
RBAC detokenization	Role-based access tied to Entra ID groups
Purview integration (optional)	Auto-ingestion of sensitivity types and labels
Audit logging	Tokenization and detokenization events captured
SIEM integration	Export and integration capability
Browser plugin (Intune)	Managed deployment via Microsoft Intune
Teams/Copilot extension	Access within Microsoft collaboration tools

Outside Beta Scope

Capability	Status
On-premises deployment	Cloud-native only
Multi-format (image/audio/PDF)	Text only
Purpose-scoped detokenization	Role-based only
Time-bound approvals	Not supported
BYOD/contractor devices	Intune-managed only
Policy rule audit logging	Decision only, not rationale

Unique Positioning

Differentiator	Why It Matters
Native Entra ID integration	No identity mapping overhead for Microsoft shops

Purview sensitivity alignment	Existing classification investments apply automatically
Customer-defined sensitivity	Higher precision than generic PII libraries
LLM-native design	Purpose-built for prompt workflows, not adapted from data lifecycle tools

Document Classification: Public / Enterprise Buyer Evaluation

Methodology: Control-surface-first analysis based on public documentation

Sourcing: Vendor documentation, public specifications, PromptVault beta features document

References

Protecto

- Protecto Product Documentation. "Data Tokenization for AI & Analytics." https://www.protecto.ai/
- Protecto Technical Overview. "Privacy-Preserving Data Protection for LLMs."
- Protecto Compliance Documentation. "HIPAA, GDPR, PCI-DSS Compliance with Tokenization."

ZeroTrusted.ai

- ZeroTrusted.ai Product Documentation. "LLM Firewall & Security Gateway." https://www.zerotrusted.ai/
- ZeroTrusted.ai Technical Specifications. "AI Judge Anomaly Detection and Prompt Sanitization."

Private AI

- Private AI Product Documentation. "Multi-Format PII Detection and Anonymization." https://www.private-ai.com/
- Private AI Technical Overview. "50+ Entity Types, 49+ Languages, Multi-Format Support."

Industry Context

- Gartner. "Market Guide for Data Loss Prevention." 2025.
- NIST. "AI Risk Management Framework." 2024.
- Microsoft. "Microsoft Purview Information Protection Documentation." https://learn.microsoft.com/en-us/purview/
- Microsoft. "Microsoft Entra ID Documentation." https://learn.microsoft.com/en-us/entra/

Note: Competitor capabilities described in this analysis are based on publicly available documentation as of January 2026. Vendors may have updated their offerings since publication. Buyers should verify current capabilities directly with vendors.