KAMIWAZA.AI

The inference firewall:

Why enterprise AI demands relationship-based access control (ReBAC)

Operationalizing trust in the age of autonomous agents

The shift from passive chatbots to autonomous "digital co-workers" changes the enterprise security problem. Traditional access controls were designed to govern human access at human speed. Autonomous agents operate at machine speed and can synthesize meaning across large volumes of information. This change introduces a new requirement: it's no longer enough to govern access — enterprises must govern inference.

This whitepaper outlines why role-based access control (RBAC) and many real-world attribute-based access control (ABAC) deployments often struggle to contain inference risk in agentic workflows. It introduces relationship-based access control (ReBAC) as an architectural layer for enforcing granular, context-aware governance. We also describe how Kamiwaza enforces permissions before retrieval occurs, reducing the likelihood that sensitive information (or sensitive relational context) enters an agent's working context in the first place.

The inference gap in traditional security

For decades, enterprise security has relied on RBAC, where permissions are granted based on job role — for example, the HR Manager can view employee records.

RBAC can be effective when users navigate structured applications and access is mediated through well-defined interfaces.

Generative AI changes the access pattern. In order to be useful, an LLM-powered copilot or agent is often connected to broad repositories, like data lakes, file shares, wikis, ticketing systems, and collaboration tools. In that environment, object-level authorization alone can become context-blind.

It checks whether a user can access a given item, but it doesn't rationalize how many permitted items can be combined to derive restricted conclusions.

This is the inference problem sometimes described as the mosaic effect. Sensitive outcomes can be inferred by synthesizing many individually permitted fragments.

The mosaic effect: How agents bypass file-level controls

Even when a user or agent is blocked from a restricted document, they may still have access to surrounding signals: calendars, travel, vendor onboarding, project updates, or operational notes. LLMs excel at pattern recognition and synthesis; they can connect disparate fragments to reach a high-confidence conclusion without ever opening the restricted file.

Risk example: An agent may not need to read Merger Agreement.pdf to infer that a merger is imminent. Access to enough related fragments, such as flight logs, calendar invites, and news snippets can be sufficient to connect the dots.

Core limitation: File-level authorization governs objects. It doesn't govern inference paths across objects.

Why ReBAC?: Governing access through relationships

Kamiwaza applies relationship-based access control (ReBAC). With ReBAC, authorization is based on how users and resources are connected through real enterprise relationships, like team, project, workspace, folder hierarchy, deal room, and data domain. This relationship model is represented in a context graph (often ontology-backed) that captures business structure and the relationships that matter for governance.

Unlike role-only decisions ("who you are") or policies that rely heavily on attributes in isolation ("what labels you carry"), ReBAC answers, "What is your relationship to this data in this context?" That's the unit of control enterprises need when agents traverse large graphs of related information.

How Kamiwaza enforcement works: Pre-retrieval controls

In many AI architectures, security is applied after retrieval, or after content enters an agent's context window. That pattern can be fragile: once sensitive content is retrieved, it may influence the model's behavior even if downstream output filtering suppresses it.

Kamiwaza enforces permissions before retrieval by gating graph traversal. Conceptually:

1. **User-scoped execution:** An agent operates as an extension of the requesting user's permissions (rather than a broad, shared service identity).
2. **Graph traversal:** The system navigates relationships across relevant nodes to identify candidate information sources.
3. **Policy gate:** ReBAC checks required relationships along the traversal path in real time.

4. **The "void":** When access isn't permitted, traversal halts. Restricted nodes and tightly coupled relational context aren't retrieved and aren't provided to the model.

This approach doesn't claim to eliminate inference risk in every scenario. It materially reduces risk by limiting exposure. If sensitive nodes and their immediate relational context aren't retrieved, the amount of usable signal, or "clues," available for inference shrinks significantly.

Mitigating inference risk in retrieval-augmented generation workflows

A common retrieval-augmented generation (RAG) failure mode relies on post-retrieval filtering — retrieve broadly, then filter responses. The safer pattern is to enforce authorization before retrieval so content never enters the model context unless it's authorized.

Kamiwaza enforces security at the graph level before retrieval occurs. Because the ontology-backed context graph encodes relationships — for example, how Public Document A relates to Restricted Document B, or how a document belongs to a restricted workspace — the system can apply governance across related nodes, not only at the single-document level.

ReBAC in action: A financial services scenario

A large investment bank uses autonomous agents for M&A due diligence. The workflow requires access to both restricted MNPI (deal models, internal valuation analyses) and non-sensitive operational documents (compliance checklists, public filings, routine business documentation) from a shared repository.

How Kamiwaza ReBAC works

- **Context graph:** Data, users, and governance boundaries (client, deal room, deal team, restricted models) are represented as live relationships.
- **Enforcement:** When a non-deal-team compliance user prompts an agent, the ReBAC layer evaluates whether traversal paths intersect restricted nodes and whether the user has the required relationships.
- **Outcome:** Lacking the Deal Team relationship, traversal paths into restricted deal artifacts terminate at the policy gate. The agent returns a compliant summary based only on authorized sources, like public filings and permitted internal documents, with an auditable record of what was accessed.

Business impact: Innovation without compromise

Moving from static RBAC to dynamic ReBAC helps resolve the security deadlock that keeps many AI initiatives trapped in pilot mode.
For the CISO (governance and auditability)

- **Auditable, user-scoped enforcement:** Actions taken by agents can be traced back to a user identity and a specific authorization decision.
- **Stronger posture for regulated environments:** Supports compliance programs and audits where least privilege, access reviews, and evidence trails are required.

For the CIO (velocity and architecture)

- **Reduced centralization pressure:** Security is enforced through context and relationships, enabling teams to connect to data where it lives — on-prem, cloud, legacy systems — while maintaining policy guardrails.
- **Clear separation of concerns:** Governance is enforced consistently at the authorization layer rather than scattered across application logic.

For the enterprise (scalable agent deployment)

- **Autonomous workflows, built-in boundaries:** Agents operate independently while staying within contextual constraints aligned to organizational structure.

Conclusion

Agentic AI changes the security question. Instead of only "Who can open this file?" its "What can be inferred from what an agent is allowed to traverse and retrieve?" ReBAC, enforced pre-retrieval through a context graph, provides a practical foundation for governing inference risk at enterprise scale. So organizations can operationalize AI without compromising security.

To learn more about ReBAC and Kamiwaza, visit kamiwaza.ai.