Securing Intelligent Indexing and Search Systems: A Defense-in-Depth Architectural Blueprint

I. Strategic Risk Analysis and the Threat Landscape in Intelligent Indexing Systems

The implementation of intelligent indexing systems, particularly those leveraging Retrieval Augmented Generation (RAG) architectures, represents a paradigm shift in knowledge accessibility for enterprise, industrial, and maritime domains. However, this shift introduces complex, data-centric vulnerabilities that move far beyond traditional perimeter defenses. Securing these systems necessitates a proactive approach focused on securing the data flow, the derived indices (vector embeddings), and the retrieval pipeline itself.

1.1. Categorization of Vulnerabilities in Modern Knowledge Systems (RAG Focus)

The primary security challenge in modern RAG systems stems from the complexity and potential permissiveness of the retrieval pipeline, which frequently operates without the requisite granular access controls necessary for high-assurance, proprietary environments.[1]

A critical vulnerability is **Prompt Injection and Response Manipulation**. This occurs when adversarial content, originating either from user input or deliberately embedded within the source documents—a technique known as Index Poisoning—hijacks the underlying Large Language Model's (LLM) response generation. This manipulation can force the model to bypass established security controls, ignore policy limits, or generate output based on unauthorized source material.[1]

Furthermore, **Data Leakage via Unfiltered Retrieval** remains the most common operational threat. Without stringent and context-aware controls, the retrieval mechanisms may surface internal-only or highly sensitive data during the initial retrieval phase, even if the final LLM output is carefully censored.[1] This confirms that the retriever component, which fetches context from the knowledge base, is the primary point of failure for access control, regardless of the LLM's sophistication.[2] This challenge is compounded by the **Insufficient Granularity of Access Controls**. Traditional Role-Based Access Control (RBAC) [3] proves inadequate because enterprise data is multidimensional; access often depends not just on a user's role, but on contextual user attributes such as region, project scope, or document classification. This lack of precision necessitates a shift toward Attribute-Based Access Control (ABAC) for effective policy enforcement.[1]

1.2. Analysis of Advanced Retrieval Pipeline Attack Vectors

Threats targeting the core infrastructure—specifically the vector store and the embedding process—represent advanced attack vectors designed to bypass conventional network security layers.

Index Poisoning and Vector Database Manipulation involve malicious actors inserting or subtly manipulating content directly within the vector databases.[1] If successful, this can steer the LLM toward generating responses that are out-of-policy, factually incorrect (hallucination based on malicious context), or that inadvertently leak sensitive data. This manipulation threat requires robust integrity controls, compelling architects to integrate cryptographic hashing mechanisms during the ingestion phase to verify data lineage and prevent stealthy corruption.

Crucially, the vector store’s status has been elevated from a passive search index to a critical data protection surface due to the threat of **Embedding Inversion Attacks**. Research has demonstrated that attackers can potentially reconstruct sensitive portions of the original data from the vector embeddings themselves, especially if the embeddings are stored without encryption or adequate access controls.[1] This vulnerability means that simply removing PII from the source document is insufficient; the derivative data (the embedding) must also be secured with the same rigor as the original document.

1.3. Real-World Incidents and Sectoral Case Studies

The risks identified architecturally are manifest in real-world operational failures. **Enterprise Data Leakage through Unfiltered Retrieval** incidents generally underscore the recurring problem where the retriever component fails to adequately filter documents based on user permissions, confirming that the most effective point of attack targets the access control logic rather than the LLM's intelligence.[1]

A broader systemic concern, illustrated by incidents involving large-scale AI platforms like ChatGPT and Grok, relates to **Mis-Indexing, PII Persistence, and Hallucination Risks**. These cases highlight systemic privacy flaws where user conversations, once shared, can persist indefinitely and may be indexed by search engines, facilitating unauthorized misuse by malicious actors or data brokers.[4] This phenomenon reveals a fundamental flaw in confidentiality management: the convenience of user interaction directly compromises the fragility of sensitive information. When conversations expose full names, locations, or intimate insights into mental health or relationships, the risk of doxxing, social engineering, and fraud rises significantly.[4] The persistence of sensitive queries and contexts—for example, a query regarding a specific crew member's health record logged within an enterprise system—creates a permanent risk where confidentiality is systematically undermined by data retention policies, necessitating explicit warnings and strict logging governance for all conversational interfaces.

1.4. Operational Technology (OT) and Disconnected Environment Risks (Maritime/Industrial)

Deployment in maritime and industrial environments introduces unique security challenges centered around operational continuity, low bandwidth, and intermittent connectivity.

A primary challenge is the **Security Challenges of Low-Bandwidth Synchronization and Offline Indexing**. Secure indexing processes must operate reliably over unstable, high-latency networks typical of remote operations (VSAT, cellular, Wi-Fi, UHF).[5] This demands fault tolerance, requiring synchronization mechanisms to automatically resume failed file transfers precisely at the point of interruption, ensuring eventual consistency and data integrity without unnecessary re-transmissions.[5]

Furthermore, these environments demand **Robust Network Segmentation**. In shipboard or industrial control system (ICS) networks, physical and logical segmentation is paramount. Dividing the network into separate security domains is crucial for containing the impact of a potential breach, preventing lateral movement from a less-secured IT segment to critical Operational Technology (OT) systems.[6] Access controls, such as Role-Based Access (RBAC) and Multi-Factor Authentication (MFA), must be deployed locally and enforced strictly at the edge, ensuring that system access is restricted solely to authorized personnel (e.g., shipping parties).[6]

II. Foundational Architectural Mitigations: Implementing Zero Trust for Data Retrieval

To address the inherent access control deficiencies in intelligent indexing systems, a fundamental architectural pivot toward a Zero Trust Architecture (ZTA) is required. ZTA mandates that security controls focus on the data resource, rather than assuming trust based on network location.

2.1. Zero Trust Architecture (ZTA) in the RAG Context (NIST SP 800-207 Adaptation)

ZTA, as defined by NIST SP 800-207, redefines the security perimeter, shifting the focus from securing network segments to protecting individual resources such as data, services, and workflows.[7] For RAG systems, ZTA implies that every single request for information, whether originating from an internal employee or a remote asset (such as an onboard vessel system), must be continuously verified and authorized.

Defining the Protect Surface is the initial critical step. This surface includes the original indexed documents, the vector embeddings (which must be secured against inversion attacks [1]), the associated metadata tags (essential for ABAC policy enforcement), and the LLM service endpoints themselves. ZTA principles demand that every access attempt (i.e., every user query) is treated as potentially hostile. Access must adhere to the principle of least privilege, and authorization must be dynamic, relying on continuous authentication and authorization checks.

Policy Enforcement Points (PEPs) in the Retrieval Flow must be strategically inserted throughout the architecture. The most crucial PEPs reside at the orchestration layer, where they evaluate the ABAC policies and dynamically compile a security filter that is applied *before* the nearest-neighbor search executes within the vector database. This design ensures that sensitive data is prevented from ever being retrieved or even considered by the retriever, mitigating the risk of unfiltered retrieval.[8]

The enforcement of continuous authorization, essential to ZTA, introduces inherent complexity and potential latency. While caching access decisions is necessary for high-performance retrieval [8], the system must enforce tight integration between the Identity Provider (IdP), the ABAC policy engine, and the RAG orchestrator. This linkage is vital to ensure that entitlement caches are immediately invalidated upon any change to a user's attributes or a revocation of policy. Without this rapid invalidation mechanism, a time window exists during which unauthorized access based on outdated cached permissions could occur.

2.2. Granular Access Control: Shifting from RBAC to Policy-Aware ABAC

Traditional Role-Based Access Control (RBAC) [3] is structurally inadequate for managing the intricate, contextual permissions required for blended enterprise data (e.g., HR and technical content mixed within a single index). Attribute-Based Access Control (ABAC) provides the required multi-dimensional granularity.

Implementing Attribute-Based Policies (ABAC) for Document-Level Security allows the retrieval orchestrator to dynamically accept the user's identity and relevant attributes (e.g., location, clearance, department) and compare these against attributes securely tagged to the documents (e.g., classification, region, line of business).[8] This dynamic evaluation is essential for environments like global banks or maritime fleets, where visibility depends on department-based authorization or regional requirements.[8]

The design must incorporate **Policy-Aware Retrieval and Filter Injection**. The retriever component is specifically engineered to inject the derived security filter directly into the vector database query. This might involve utilizing metadata filtering capabilities inherent in the vector database to restrict the scope of the nearest-neighbor search itself, ensuring that only records matching both the query and the access policy are ever returned to the orchestrator.[8]

For maximum security assurance, **Entitlement Recheck and Post-Filtering** must be implemented. The orchestration layer performs a final entitlement check post-retrieval. This process removes any candidate documents that may have passed the initial vector search filter but fail a final, current policy verification, offering a crucial layer of defense against subtle policy changes or retrieval inaccuracies.[8]

2.3. Cryptographic Controls and Data Integrity (Hashing and Encryption)

Cryptographic measures are essential for protecting the index's integrity against Index Poisoning and for securing the vector embeddings themselves against reconstruction attacks.

The **Role of Hashing in Data Lineage and Tamper Detection** involves applying a strong cryptographic hash (e.g., SHA-256) to every source document before it is chunked and embedded. This hash creates an immutable digital fingerprint that must be securely linked to the generated embeddings and ingestion logs.[8] In an operational environment, this linkage provides a crucial audit trail: any discrepancy between the recorded hash and the source document signals potential Index Poisoning or data corruption. This check verifies that the indexed content accurately matches the source content, providing integrity guarantees necessary for audit compliance.

The requirement for **Secure Storage and Encryption of Vector Embeddings** arises directly from the threat of Embedding Inversion Attacks.[1] Because data can potentially be reconstructed from the vectors, the vector store must be treated as highly sensitive data storage. This demands the use of per-tenant encryption keys, the establishment of mutual TLS (mTLS) for all transport connections, and, ideally, the use of dedicated, isolated indices for the most sensitive data classifications.[8] This rigorous defense of the "data at rest" layer prevents unauthorized reconstruction of PII or confidential information from the index itself.

The vector database is therefore functionally reclassified from a mere index to a first-class security asset, requiring multi-layered defenses comparable to those applied to production relational databases (encryption, mTLS, anomaly detection).[8]

In environments with intermittent connectivity, such as maritime operations, the principles of ZTA—specifically continuous authorization—must be adapted. To maintain integrity while offline [5], the necessary ABAC policy rules and user attributes must be synchronized onto the local Policy Enforcement Point (PEP), often residing next to a localized vector database instance.[10] Access decisions made locally are logged as immutable, cryptographically **signed ingestion artifacts** [8], ensuring non-repudiation and guaranteeing that a complete audit trail is prioritized for synchronization back to the central policy engine once connectivity is restored. This shifts trust from "continuous verification" to "verified immutable state" during disconnected periods.

Zero Trust Enforcement Points (PEPs) in the RAG Pipeline
Component
Ingestion Pipeline

Identity Provider (IdP)
Policy Engine (PDP/PEP)
Retrieval Orchestrator
Vector Database (VDB)
LLM Service

III. Data Privacy and PII Protection for Blended Content

Indexing systems in regulated environments, particularly enterprise and maritime operations, must manage complex datasets that blend technical operational documents (e.g., vessel schematics, maintenance procedures) with sensitive Human Resources (HR) content (e.g., crew manifests, health assessments). Protecting Personal Identifiable Information (PII) contained in crew names and HR details is non-negotiable and requires active sanitization *before* indexing occurs.

3.1. Governance Framework for HR and Crew Data in Technical Indices

A robust governance framework provides the essential foundation for managing PII across mixed content types and international jurisdictions.[11]

Data Classification Taxonomy and Purpose Tags must be mandatory for all documents entering the ingestion pipeline. All incoming content—whether technical, operational, or HR—must be rigorously categorized and tagged (e.g., "Internal," "Confidential," "Contains PII," "HR-Only").[8] These structured tags form the bedrock of the ABAC policies, enabling granular access decisions based on the data's inherent sensitivity and classification. The governance framework must also define and strictly apply **Retention Policies for Embeddings and Logs**, ensuring that derivative data (vector embeddings and query logs) adhere to data lifecycle management requirements and are deleted when the purpose for processing is concluded.[8]

For organizations operating globally, such as maritime fleets, mandatory **Leveraging of Privacy Indices and Compliance Benchmarks** is necessary. These assessments evaluate the stringency of privacy legislation in every operational jurisdiction (e.g., GDPR, CCPA) based on factors like territorial scope and legal bases for processing.[11] This rigorous Privacy Index assessment informs the minimum level of PII sanitization required across the entire organization, ensuring that policies (e.g., mandatory pseudonymization for EU-based crew data) meet the most stringent relevant standard.

3.2. Technical PII Sanitization and Obfuscation Techniques (Pre-Embedding)

To prevent sensitive PII from ever residing in an index susceptible to retrieval or inversion attacks, data sanitization must occur *before* the document is chunked and passed to the embedding model.[2]

The fundamental risk mitigation tool is **Comprehensive Data Masking**. Techniques applied to PII fields include: **Redaction**, which replaces PII partially or completely with generic values; **Nulling Out**, which applies a null value to the data column; and **Shuffling**, which randomly inserts masked data from other records.[12] While offering maximum risk reduction, these methods can severely limit the utility of the indexed content.

For cases where data utility must be preserved alongside privacy, **Pseudonymization** is utilized. This involves swapping identifying PII (such as a crew member's name) with a random, secure token (a pseudonym) while securely storing the original link between the pseudonym and the real identity separately, often in an encrypted lookup table.[12] Pseudonymization is highly effective for both unstructured and structured data and solves a critical complexity in enterprise indexing: the Blended Content Challenge. Crew names often appear in both sensitive HR documents and less sensitive technical logbooks (e.g., "Engineer Smith authorized repair"). Pseudonymizing 'Smith' into 'Crew-ID-42' globally allows the index to unify technical retrieval (searching by 'Crew-ID-42') without allowing unauthorized access to the underlying sensitive HR data.[12] Access to the original PII remains strictly gated and audited.

For analytical or statistical use cases, **K-Anonymity** may be employed. This ensures that the released data cannot be uniquely distinguished from at least $k-1$ other individuals.[13] K-anonymity is achieved by suppressing direct identifiers (names) and generalizing quasi-identifiers (e.g., grouping age into ranges or locations into general states) to strike a necessary balance between data utility and privacy protection.[13] The choice of technique must align precisely with the data's defined purpose—strict redaction for diagnostics, or pseudonymization/K-anonymity where analytics are required.

3.3. Implementing Secure Pre-Processing and Ingestion Pipelines

The ingestion process must function as a fortified gateway, ensuring that all data meets security and governance standards before indexing.

The **Redaction/Pseudonymization Gate** is the critical security checkpoint where sophisticated Natural Language Processing (NLP) tools identify PII and actively mask or anonymize it. This process must occur *before* data chunking and embedding generation.[2] This is the primary defense against PII leakage into the vector space.

The system must also incorporate **Quarantine Zones for Untrusted or Unclassified Data**. Any data source lacking an established lineage, clear purpose tags, or appropriate classification taxonomy must be isolated. In industrial contexts, this is essential for documents sourced from third-party contractors or field equipment that may inadvertently contain unknown PII or proprietary content without the appropriate security metadata.[8]

Finally, the integrity of the process itself must be auditable through the use of **Signed Ingestion Artifacts**. The entire ingestion workflow, including classification, sanitization, and embedding operations, must produce cryptographically signed logs. This provides an immutable and verifiable record for compliance audits, detailing the processing steps taken for every piece of content indexed.[8]

IV. Best Practices for Operational Resilience and Deployment (Securing the Edge)

Industrial and maritime environments pose critical deployment constraints, requiring systems that are robust, isolated, and capable of maintaining security integrity even when disconnected from centralized resources.

4.1. Network Segmentation and Isolation for Industrial/Maritime

Systems

Containment via robust network architecture is the foundational layer of defense in environments operating complex physical assets. **Physical and Logical Isolation** is mandatory for vessels or industrial sites. Shipboard networks, for instance, must be segmented into distinct security domains—OT (control systems), Business IT, and the Crew Network—each with its own security controls.[6] The RAG indexing system must reside exclusively within the segmented IT domain, isolated entirely from mission-critical control systems (OT).

This segmentation must be paired with **Strict Access Controls at the Edge**. Multi-factor authentication (MFA) and strong access policies must be enforced even in localized or isolated environments. These controls ensure that only authorized shipping parties or personnel with the required clearance can access sensitive onboard systems.[6]

4.2. Secure Offline Mode and Data Synchronization Protocols

Data availability and integrity cannot rely solely on continuous, high-speed connection. Systems must be engineered for reliable operation during connectivity blackouts.

Data synchronization between shore and remote assets requires **Fault-Tolerant, Encrypted Data Transfer**. Protocols utilized must be capable of high efficiency over variable bandwidth (e.g., WAN acceleration) and provide robustness through fault tolerance, automatically resuming failed transfers at the point of interruption.[5] All synchronization activity must be secured using end-to-end encryption (mTLS is highly recommended) to protect the confidentiality and integrity of data packages while in transit. Critically, this synchronization mechanism must be protected against Index Poisoning by integrating the validation of cryptographic hashes and signed ingestion artifacts directly into the transfer handshake, ensuring the integrity of policy and data packages being pushed to the remote asset.

To support continuous, low-latency retrieval while disconnected, **Localized, Secure Vector Database Deployments** are necessary. A high-performance, deployable vector database (such as Qdrant [10]) must be instantiated locally on the vessel or industrial site. This local VDB holds a secure, replicated subset of the centralized index, complete with localized ABAC policies and a local Policy Enforcement Point (PEP).

Finally, **Patch Management in Disconnected Mode** must be handled with care. The regular application of security patches and software updates must be managed efficiently, often requiring staged, verified deployment bundles that are synchronized via the secure transfer mechanism and validated locally before installation.[6]

4.3. Secure Development and LLMOps Governance

The complexity and high-risk nature of RAG infrastructure require that it be managed under continuous security assessment and strict governance protocols.[2]

Auditability and Explainability in the Retrieval Chain are central to compliant operations. All retrieval activity must be immutably logged, capturing the user query, the specific ABAC policy filters applied, the actual documents retrieved, and the final LLM response along with its citations.[8] This meticulous logging provides complete traceability, ensuring "audit trails that map each response to the exact policy versions" used to derive the answer.[8]

RAG systems are structurally complex, making **Continuous Security Assessment** a necessity. Security teams must continuously test for gaps in the embedding models, retrieval logic, and vector store configurations that could unintentionally lead to sensitive data leakage.[2] Furthermore, data quality assurance [14] is directly linked to security; mis-indexed data or poor classification can cause the LLM to retrieve the wrong type of document (e.g., HR instead of technical), and the subsequent hallucination based on this retrieved but irrelevant sensitive data becomes a critical security leak.[4]

Architectural agility is achieved through **Modular Design for Adaptability**. The architecture must be constructed in a modular fashion, allowing for the easy replacement of individual components (LLMs, embedding models, vector databases) as new, more secure technologies or updated compliance standards emerge.[14]

V. Recommended High-Assurance Security Architecture Sketch

This section synthesizes the identified threats and mitigations into a prescriptive, defense-in-depth architectural blueprint suitable for securing intelligent indexing and search systems across global enterprise, industrial, and maritime operations. The core principle is policy-aware retrieval operating within a Zero Trust framework.

5.1. The Policy-Aware Retrieval Stack: Component Breakdown

The security of the RAG system relies on the collaborative enforcement of security measures across all layers of the retrieval stack:

- **Identity Provider (IdP):** Serves as the authoritative source for user identity, authentication, and core attributes (role, location, security clearance).
- **Policy Engine (PEP/PDP):** The brain of the ABAC implementation. It evaluates incoming user attributes against indexed document metadata to dynamically generate necessary security filters.
- **Ingestion Pipeline (The Secure Gate):** This component is responsible for governance enforcement, including data classification, PII sanitization (redaction/pseudonymization), cryptographic hashing of source content, chunking, and embedding generation. It includes quarantine zones for unclassified or untrusted data.[8]
- **Vector Database (VDB):** Stores encrypted embeddings and essential metadata (access control tags). It utilizes per-tenant/per-region isolation and mTLS for secure communication.[8]
- **Orchestrator/Retrieval Service:** Functions as the primary Policy Enforcement Point (PEP). It receives the user query, obtains identity attributes, requests filters from the Policy Engine, injects those filters into the VDB query, performs the critical entitlement recheck post-retrieval, and manages citation enforcement.[8]
- **LLM Service:** The generative component, operating behind guardrails that enforce policies such as requiring citations for all claims and actively suppressing PII in the final output.[8]

5.2. Defense-in-Depth Checklist for Vector Database Security

The vector database is a prime target for attack, necessitating specialized security controls to protect the indexed information against embedding inversion and unauthorized access.

Table V.1: Vector Database Security Checklist (Defense-in-Depth)

Security Layer	Control Mechanism	Mitigated Threat

Data at Rest	Per-tenant/Per-region Isolation & Encryption	Embedding Inversion Attacks [1], Unauthorized Bulk Access [8]
Data in Transit	Mutual TLS (mTLS) & Rate Limiting	Eavesdropping, Denial of Service [8]
Data Integrity	Hashing of Source Documents (Linked to Embeddings)	Index Poisoning, Data Corruption [1]
Access Enforcement	Document-level Metadata Filtering (ABAC)	Data Leakage via Unfiltered Retrieval [8]

5.3. Conceptual Architectural Diagram: Secure Enterprise/Maritime RAG Blueprint

The recommended high-assurance architecture operates across three domains: the centralized core, the transit layer, and the remote edge.

The **Core Enterprise Layer (Shore-Based Data Center/Cloud)** houses the centralized Policy Engine, the Master Vector Store (federated by region or tenant), and the Secure Ingestion Pipeline. All data must pass the PII Sanitization Gate and integrity checks (hashing) here before indexing. The Core maintains the golden copy of the ABAC policies and manages central identity governance.

The **Synchronization and Transit Layer** utilizes secure, fault-tolerant protocols.[5] This layer manages the secure transfer of synchronization packages, which include policy updates, new index segments, and cryptographically signed ingestion artifacts, all secured by mTLS.[8] This process is robust against intermittent connectivity, automatically resuming transfers and verifying the integrity of the package upon arrival.

The **Edge/Maritime Layer (Onboard System)** represents the remote operational environment. It contains a localized, secure VDB [10] and a localized Orchestrator/PEP. This local PEP maintains cached, immutable ABAC rules, allowing for secure, low-latency retrieval while operating in disconnected or low-bandwidth mode. All retrieval activities performed locally are logged immutably, generating audit trails that are prioritized for synchronization back to the core system once network stability permits.[6]

This comprehensive architecture ensures that security is applied at the source (PII removal), enforced at the retrieval layer (ABAC filtering), assured via cryptography (hashing and encryption), and maintained through operational resilience (secure offline mode).

VI. Conclusions and Recommendations

The proliferation of intelligent indexing systems presents systemic security challenges rooted in the complexity of RAG components and the blending of sensitive HR/crew data with technical content. Relying solely on perimeter security or traditional RBAC is insufficient.

The analysis confirms that the primary threat vector is **unfiltered retrieval** and **index manipulation**, demanding architectural controls focused on data-centric security. The Vector Database is confirmed as a first-class security asset requiring database-level defense measures, including encryption and strict isolation.

To achieve high assurance in enterprise, industrial, and maritime deployments, the following actionable recommendations are necessary:

1. **Mandate Zero Trust and ABAC:** Adopt a NIST SP 800-207-aligned ZTA model, positioning the retrieval orchestrator as the primary Policy Enforcement Point (PEP). Transition from Role-Based Access Control (RBAC) to Attribute-Based Access Control (ABAC) to ensure policy-aware retrieval via metadata filtering at the document level.[8]
2. **Enforce Pre-Embedding PII Sanitization:** Establish a mandatory PII Sanitization Gate within the ingestion pipeline. Utilize pseudonymization for crew and HR data to maintain index utility while segregating sensitive identity information, ensuring compliance with global privacy standards before data is vectorized.[2]
3. **Ensure Data Integrity via Hashing:** Integrate cryptographic hashing into the ingestion workflow and link these hashes to the embeddings (Signed Ingestion Artifacts). This is the only reliable mitigation against stealthy Index Poisoning and ensures an immutable audit trail for governance.[8]
4. **Architect for Disconnected Operations:** Deploy localized, secure vector databases and local Policy Enforcement Points (PEPs) on remote assets (vessels/sites) to maintain ZTA policy enforcement in offline mode. Utilize fault-tolerant synchronization protocols secured by mTLS to guarantee the integrity and consistency of data and policy updates across the fleet.[5]
5. **Treat Conversational Logs as Sensitive Data:** Implement mandatory dynamic masking or immediate quarantine for all conversational histories and query logs, recognizing the persistent nature of PII leakage demonstrated in real-world incidents.[4]

Works cited

1. Secure Retrieval-Augmented Generation (RAG) in Enterprise Environments - Daxa.ai, accessed October 17, 2025, https://www.daxa.ai/blogs/secure-retrieval-augmented-generation-rag-in-enterprise-environments
2. RAG Systems are Leaking Sensitive Data | we45 Blogs, accessed October 17, 2025, https://www.we45.com/post/rag-systems-are-leaking-sensitive-data
3. What Is Role-Based Access Control (RBAC)? - IBM, accessed October 17, 2025, https://www.ibm.com/think/topics/rbac
4. Sensitive Data Leaks From ChatGPT & Grok - Cyber Security Intelligence, accessed October 17, 2025, https://www.cybersecurityintelligence.com/blog/sensitive-data-leaks-from-chatgpt-and-grok-8680.html
5. Reliable, Secure & Fast Maritime File Transfer with Resilio Active Everywhere, accessed October 17, 2025, https://www.resilio.com/blog/maritime-file-transfer
6. Maritime Cybersecurity: Navigating the Challenges and Implementing Solutions, accessed October 17, 2025, https://www.virtuemarine.nl/post/maritime-cybersecurity-navigating-the-challenges-and-implementing-solutions
7. Zero Trust Architecture - NIST Technical Series Publications, accessed October 17, 2025, https://nvlpubs.nist.gov/nistpubs/specialpublications/NIST.SP.800-207.pdf
8. Securing Enterprise RAG: Governance, Vector DB Security & LLMOps for Compliant GenAI, accessed October 17, 2025, https://petronellatech.com/blog/securing-enterprise-rag-governance-vector-db-security-llmops-for-compliant-genai/
9. RAG & RBAC integration: Protect data and boost AI capabilities - Elasticsearch Labs, accessed October 17, 2025, https://www.elastic.co/search-labs/blog/rag-and-rbac-integration
10. Qdrant - Vector Database - Qdrant, accessed October 17, 2025, https://qdrant.tech/
11. DataGuidance Privacy Index | MyOneTrust, accessed October 17, 2025, https://my.onetrust.com/s/article/UUID-861ca6b9-8d3a-0fb8-705d-dad804c6ddf4?
12. What is Data Masking? A Practical Guide - K2view, accessed October 17, 2025, https://www.k2view.com/what-is-data-masking/
13. k-anonymity - Wikipedia, accessed October 17, 2025, https://en.wikipedia.org/wiki/K-anonymity
14. Insights into RAG Architecture for the Enterprise - Squirro, accessed October 17, 2025, https://squirro.com/squirro-blog/rag-architecture