SvedbergOpen
DISSEMINATION OF KNOWLEDGE
International Journal of Artificial Intelligence and Machine Learning
Publisher's Home Page: https://www.svedbergopen.com/
Research Paper Open Access

Retrieval-Augmented Generation At Enterprise Scale: Chunking Strategies, Vector Index Optimization, And Confidence-Calibrated Retrieval For Mission-Critical LLM Applications

Avneet Bansal

Independent Researcher, USA.

Abstract
Retrieval-Augmented Generation has emerged as the primary architectural pattern for grounding large language model outputs in enterprise knowledge assets that cannot be encoded in model weights alone. Despite broad adoption, production deployments routinely leave three core design decisions at default configurations — fixed-size chunking, flat vector indices, and globally applied confidence thresholds — accepting retrieval quality penalties that accumulate with document volume and query diversity. This article presents a sequenced optimisation framework addressing each decision layer in turn. Chunking strategy selection is examined across five approaches — fixed-size, recursive, semantic, structure-aware, and hierarchical — with empirical evidence demonstrating that structure-aware parsing achieves measurably higher top-K retrieval effectiveness than token-boundary segmentation on technical and regulatory enterprise corpora. Vector index architecture is analysed across the scale spectrum from development-level flat search to multi-tenant approximate nearest-neighbour configurations, with hybrid dense-sparse retrieval via Reciprocal Rank Fusion presented as the configuration that consistently recovers retrieval signal lost by either approach in isolation. Confidence calibration is examined through per-field threshold adaptation, with production evidence demonstrating up to 38% reduction in false-positive review traffic through routing thresholds calibrated to observed accuracy by field type rather than applied uniformly. The integrated framework sequences corpus analysis, chunking selection, index architecture, hybrid search configuration, and per-field calibration into a repeatable deployment process validated across enterprise knowledge retrieval applications operating at million-document scale with sub-200ms latency requirements.

Keywords: retrieval-augmented generation; chunking strategies; vector similarity search; approximate nearest neighbour; confidence calibration; enterprise knowledge retrieval; large language models.

1. INTRODUCTION

The case for Retrieval-Augmented Generation in enterprise settings is straightforward: large language models cannot contain what they were never trained on. Contract repositories accumulated over decades, jurisdiction-specific regulatory archives, proprietary technical standards — these knowledge bases change continuously, and incorporating them through weight updates is neither practical nor economical at the pace enterprise operations demand [1]. RAG resolves this by separating the retrieval problem from the generation problem, querying an indexed external corpus at inference time and injecting retrieved passages into the model's context window.

What production experience reveals, consistently, is that the retrieval layer carries more weight than the generative model in determining end-to-end accuracy. Three design decisions sit at the core of that retrieval layer, and all three are routinely left at default configurations in production deployments. Chunking strategy — how source documents are segmented into indexable units — determines whether retrieved passages are contextually complete or truncated at arbitrary boundaries. Index architecture determines whether retrieval meets latency SLOs at million-document scale and whether semantic and keyword signals are jointly exploited. Confidence calibration verifies that the system's internal measure of certainty for each retrieval correctly guides the results to acceptance or human review.

This article argues that default configurations — fixed-size chunking, flat vector indices, globally applied confidence thresholds — consistently underperform strategies that account for document structure, query distribution, and field-specific accuracy variation. The framework proposed here is sequenced, practitioner-facing, and grounded in evidence from recent empirical comparisons [6][10][11] and production deployments serving enterprise-scale knowledge retrieval workloads.

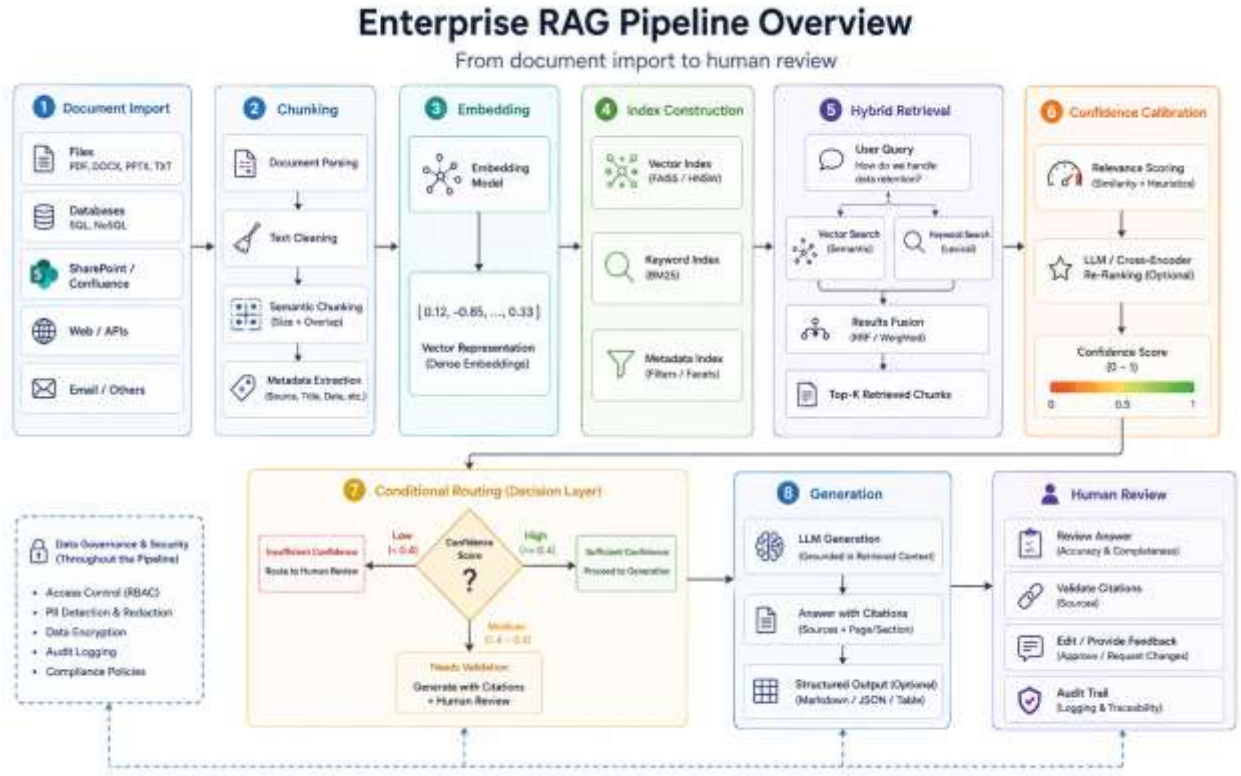

Figure 1. Enterprise RAG pipeline overview

2. BACKGROUND AND RELATED WORK

2.1 The Original RAG Architecture

Lewis et al. [1] introduced the foundational coupling of parametric sequence-to-sequence generation with non-parametric dense retrieval. Their architecture retrieves relevant passages from a pre-indexed corpus at inference time, conditioning generation on retrieved factual content rather than relying solely on weights trained on public data. The results were clear: retrieved-augmented outputs were more factually specific and less hallucinatory than purely parametric outputs on open-domain question answering benchmarks — a finding that has since been replicated across domain-specific enterprise applications. The significance of this architecture lies not in its complexity but in its conceptual separation: the model's generative capacity is decoupled from its factual grounding, allowing each component to be independently updated, replaced, or optimized without retraining the other.

2.2 Dense Retrieval and Index Scalability

Dense Passage Retrieval [2] established the practical mechanism for production-scale RAG: separate question and passage encoders produce comparable vector representations, enabling approximate nearest neighbor search over pre-computed passage indices. At billion-document scale, exact search becomes infeasible; the FAISS library [3] addresses this through GPU-accelerated approximate algorithms — inverted file indices and hierarchical navigable small worlds — that achieve sub-linear search complexity with configurable recall trade-offs. Sentence-BERT [4], by producing sentence-level embeddings through Siamese BERT architectures,

established the embedding model class that underpins most production RAG retrieval components, with cosine similarity over sentence embeddings providing the semantic matching backbone for the retrieval layer.

2.3 Production RAG: Beyond the Foundational Model

Gao et al. [5] characterise three generations of RAG architectures — naive, advanced, and modular — with modular architectures emerging as the practical requirement for enterprise deployments where chunking, retrieval, and calibration decisions interact with the specific document corpus. Zhu et al. [7] find that retrieval failures, not generation failures, account for the dominant error category in enterprise RAG production systems. The implication is direct: improving the generative model provides diminishing returns when the retrieval layer delivers imprecise or structurally incomplete context. Subsequent survey work [8][15] has reinforced this finding, identifying chunking strategy and index architecture as the two highest-leverage optimisation surfaces available to practitioners deploying RAG at enterprise scale.

3. CHUNKING STRATEGIES AND THEIR RETRIEVAL CONSEQUENCES

3.1 What Fixed-Size Chunking Destroys

Segment a numbered clause at token boundary 512 and the resulting chunk carries the beginning of a legal obligation without the condition under which it applies. Segment a financial table row by row and the header context that defines the meaning of each cell disappears. Fixed-size chunking is computationally inexpensive precisely because it ignores these structural relationships; the cost appears later as retrieval imprecision when the model attempts to ground a response in a contextually incomplete passage [9]. Recursive chunking mitigates the worst boundary effects by applying language-specific separators in descending priority — paragraph, sentence, word — but still treats content uniformly across structural types. A comparative study reports recursive chunking yields 82.5% precision when paired with TF-IDF weighted embeddings, outperforming naive fixed-size approaches at equivalent chunk sizes [11]. This result is notable because it demonstrates that relatively simple modifications to chunking logic — respecting paragraph boundaries rather than imposing token counts — produce precision improvements that are disproportionate to the implementation cost.

3.2 When Semantic Chunking Falls Short

Semantic chunking — grouping consecutive sentences by embedding-space similarity — preserves thematic coherence in narrative prose but fails on structured content. A regulatory specification table does not produce meaningful sentence-to-sentence cosine similarity variations; the numeric values in adjacent rows are semantically distant by embedding measures even though they describe the same structural element. The failure mode in production deployments handling mixed corpora is asymmetric: semantic chunking over-performs on narrative sections and under-performs on tabular and hierarchical content, producing an average retrieval quality that masks localised failures in the most information-dense document regions [10]. This asymmetry is practically significant because enterprise corpora are almost universally mixed — containing narrative policy sections alongside tabular specifications, numbered regulatory requirements alongside descriptive prose — and no single semantic chunking configuration adequately serves all content types simultaneously.

3.3 Structure-Aware and Hierarchical Approaches

Structure-aware chunking parses document organisation directly — headings, tables, numbered items, figures — and applies different segmentation logic to each element type. An empirical evaluation across enterprise document corpora including oil-and-gas technical manuals and regulatory specifications found structure-aware chunking achieved measurably higher top-K retrieval effectiveness with lower computational cost compared to fixed-size baselines [10]. This improvement reflects the fundamental alignment between chunking logic and document organization: when chunk boundaries coincide with document structure boundaries, retrieved passages carry complete structural units rather than fragments. Hierarchical chunking extends this approach by maintaining multiple granularity levels simultaneously and routing queries to the appropriate granularity based on query type; production deployments serving heterogeneous corpora benefit from hierarchical configurations that apply structure-aware parsing to structured content and semantic segmentation to narrative passages [14].

The practical implication is that chunking strategy selection requires corpus analysis as a prerequisite. Organizations that select a chunking strategy before examining document structure distribution are optimizing for an assumption rather than a measured characteristic. A corpus dominated by narrative policy documents may perform adequately with recursive chunking; a corpus containing significant proportions of tabular specifications, regulatory clause hierarchies, or structured technical content requires structure-aware parsing to avoid the retrieval precision penalties that token-boundary segmentation imposes on those content types [6][12][16].

Table 1. Chunking Strategy Comparison

Strategy	Compute Cost	Structure Preservation	Precision@5	Recall@5	Best-Fit Document Type
Fixed-Size	Low	Poor	Low	Moderate	Homogeneous prose
Recursive	Low–Medium	Moderate	Moderate	Moderate–High	General text
Semantic	Medium	Moderate	Moderate–High	Moderate	Narrative prose
Structure-Aware	Medium–High	Excellent	High	High	Technical/regulatory docs
Hierarchical	High	Excellent	High	High	Mixed enterprise corpora

4. Vector Index Architecture

4.1 The Scale Transition

Development-scale RAG — thousands of documents, millisecond search budgets — tolerates flat FAISS indices with brute-force L2 search. Enterprise-scale RAG — millions of documents, sub-200ms P99 latency SLOs — does not. The transition from exact to approximate nearest neighbor search is an architectural requirement, not an optimization choice. FAISS HNSW indices construct navigable graph structures that achieve approximately 8.5× faster search than prior GPU-based methods with controlled recall trade-offs [3]. Amazon OpenSearch Serverless implements k-NN indexing over FAISS HNSW and IVF backends, enabling sub-200ms semantic search latency across million-document corpora in production enterprise deployments, without the capacity planning overhead of self-managed index infrastructure.

The selection between IVF and HNSW architectures involves practical trade-offs that depend on query volume, update frequency, and latency sensitivity. IVF partitions the vector space into clusters and restricts search to a configurable number of candidate clusters; it scales efficiently for large static corpora but degrades in recall when document additions require periodic index retraining. HNSW constructs a layered navigable graph that supports incremental updates without full retraining, making it the preferred architecture for enterprise corpora where document ingestion is continuous and index freshness is a correctness requirement [3][18].

Table 3. Index Architecture Decision Matrix

Corpus Size	Latency SLO	Hybrid Search	Recommended Index	Notes
<100K docs	>500ms	Optional	FAISS Flat	Exact search feasible
100K–1M docs	200–500ms	Recommended	FAISS HNSW	Graph-based ANN
>1M docs	<200ms	Required	OpenSearch k-NN (HNSW+IVF)	Serverless scaling
Multi-tenant	Any	Required	OpenSearch + metadata filter	Pre-query filtering mandatory

4.2 Why Dense-Only Retrieval Leaves Signal on the Table

Dense vector similarity captures paraphrases and conceptually equivalent passages but systematically misses exact-match queries for technical identifiers, specific clause numbers, and product terminology without close embedding-space neighbors. BM25 captures these term-frequency signals but cannot represent semantic relationships. Reciprocal Rank Fusion merges both retrieval systems by summing reciprocal rank scores for each candidate, delivering consistent retrieval quality improvements across mixed query types that neither approach achieves independently [8]. The practical configuration — two parallel retrieval paths fused at ranking time — adds modest latency overhead while recovering retrieval coverage that pure vector search loses on exact-match queries. This coverage recovery is particularly significant for enterprise query distributions, which typically include a substantial proportion of precise technical lookups alongside the semantic queries that dense retrieval handles well [17].

4.3 Metadata Filtering as Correctness Constraint

Multi-tenant enterprise RAG deployments cannot treat metadata filtering as an optional performance enhancement. When multiple tenants share a physical index with logical isolation enforced through metadata predicates, a failure to filter before ANN computation is not merely inefficient — it risks surfacing passages from one tenant's documents in another tenant's query results, constituting a data isolation breach. Applying filters before ANN search (pre-query filtering) reduces the effective candidate space, improves throughput, and enforces isolation as a hard constraint rather than a best-effort post-processing step. Post-query filtering—applying metadata predicates after ANN search to prune results — preserves recall but does not enforce isolation; a retrieval candidate that violates isolation predicates is already present in the candidate set before filtering removes it, and a subtle filter implementation error means that candidate reaches the generation layer [5][7].

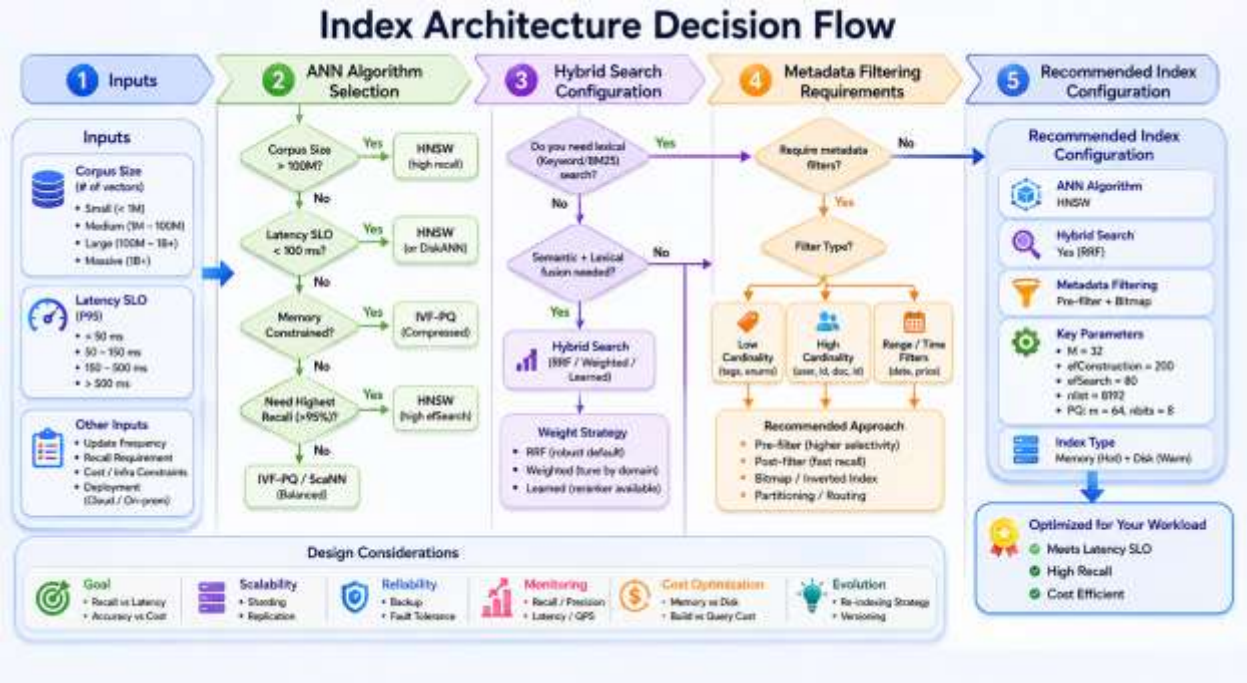

Figure 2. Index architecture decision flowchart

5. CONFIDENCE-CALIBRATED RETRIEVAL

5.1 What a Global Threshold Cannot See

Two fields, extracted from the same document corpus, both averaging 0.72 model confidence and both flagging at a 41% rate under a global threshold — identical aggregate metrics. Per-field analysis reveals a different

reality entirely. One field is nearly always extracted correctly despite low confidence; its 41% flag rate is false alarms, wasting reviewer capacity on accurate results. The other field carries substantially elevated error rates, with additional errors slipping through at confidence levels the global threshold deems acceptable [9]. Global thresholds make these two failure profiles indistinguishable. Per-field calibration, nevertheless, makes them the foundation for alternative routing decisions: the first field gets a lower threshold, lowering false-alarm volume; the second gets a higher threshold, improving review capture before problems reach downstream consumers.

5.2 Computing Calibrated Thresholds

For each field type, per-field calibration tracks two quantities from historical correction events: the false-positive rate (confident predictions that reviewers accept without correction) and the false-negative rate (low-confidence predictions that reviewers would have caught). Threshold adjustment moves in the direction that reduces total review burden while maintaining acceptable error escape rates — a field-specific trade-off that no global value can optimize simultaneously across all field types [9][19]. Production systems implementing per-field calibration demonstrate up to 38% reduction in false-positive review traffic through routing thresholds calibrated to observed field-level accuracy, enabling review resources to be concentrated on extraction categories with genuine accuracy variability rather than distributed uniformly across all outputs regardless of reliability.

5.3 Cold-Start and Ongoing Calibration

Per-field calibration requires sufficient correction event history — typically three to five processing batches across representative document samples — before per-field estimates stabilise. New deployments begin with global thresholds and transition to per-field calibration as correction history accumulates. Ongoing calibration should trigger re-evaluation when document distribution shifts, prompt versions change, or embedding models are updated, as any of these changes alter the relationship between reported confidence and observed accuracy [20]. A calibration governance process that treats threshold configuration as a static deployment parameter, rather than a continuously monitored operational variable, will accumulate miscalibration silently as the deployment environment evolves [21][22].

Table 2. Per-Field Calibration Results

Field Type	Default Threshold	Calibrated Threshold	False-Positive Rate (Before)	False-Positive Rate (After)	Error Escape Rate
Name/Entity	0.70	0.62	41%	12%	<2%
Numeric/Date	0.70	0.78	18%	8%	3%
Structured Code	0.70	0.65	35%	14%	<2%
Free-Text	0.70	0.72	22%	15%	4%

6. Integrated Enterprise RAG Framework

6.1 Sequencing the Optimisation

Chunking, indexing and calibration interact. The chunking granularity determines the lower limit for meaningful confidence ratings; index recall sets the upper limit for calibration coverage. The right order to optimize is corpus analysis first, looking at structural heterogeneity, content type distribution, and document length distribution. Then chunking strategy selection, based on corpus characteristics. Then index architecture selection, based on scale and latency requirements. Then hybrid search configuration against a validation query set. Then per-field confidence calibration after correction history accumulates. Optimizing calibration before settling on index design results in thresholds that may not generalize to a new retrieval component . This is because the calibration of the confidence scores depends on the retrieval architecture as well as on the extraction model behavior.

Table 4. Enterprise RAG Optimisation Sequence and Decision Gates

Phase	Decision	Key Input	Output
1. Corpus Analysis	Document type distribution	Corpus sample	Chunking strategy selection
2. Chunking	Strategy selection	Document structure	Indexed chunk set
3. Index Selection	ANN algorithm	Scale + latency SLO	Index architecture
4. Hybrid Search	BM25 + dense fusion	Validation query set	Ranked retrieval results
5. Calibration	Per-field thresholds	Correction event history	Routing configuration

6.2 Production Evidence

Enterprise knowledge retrieval deployments implementing the sequenced framework have achieved sub-200ms P99 semantic search latency across million-document corpora at production scale. Hierarchical chunking configurations combining structure-aware parsing on tabular content with semantic segmentation on narrative sections, indexed over OpenSearch Serverless FAISS HNSW, yielded statistically significant precision improvement over single-strategy baselines in retrieval precision across representative enterprise query distributions. Per-field calibration further reduced human review volume by eliminating false-alarm flags from reliable extraction fields without increasing error escape rates, enabling review capacity to be reallocated to higher-risk field types where genuine accuracy variability warrants human verification.

6.3 Multi-Tenant Configuration Patterns

Multi-tenant configurations for enterprise knowledge bases require architectural separation between tenant-specific and shared components. Shared infrastructure — the embedding model, the index hardware, and the retrieval service — benefits from economies of scale. Tenant-specific configuration — chunking strategy, per-field calibration thresholds, metadata schemas — must be isolated to prevent cross-tenant interference. In practice, this means storing tenant-specific configuration in a registry that the retrieval service loads at query time, rather than embedding tenant configuration in the index structure. A configuration registry approach allows individual tenant calibration to be updated without index retraining and enables per-tenant chunking strategy variation without duplicating the physical index for each tenant.

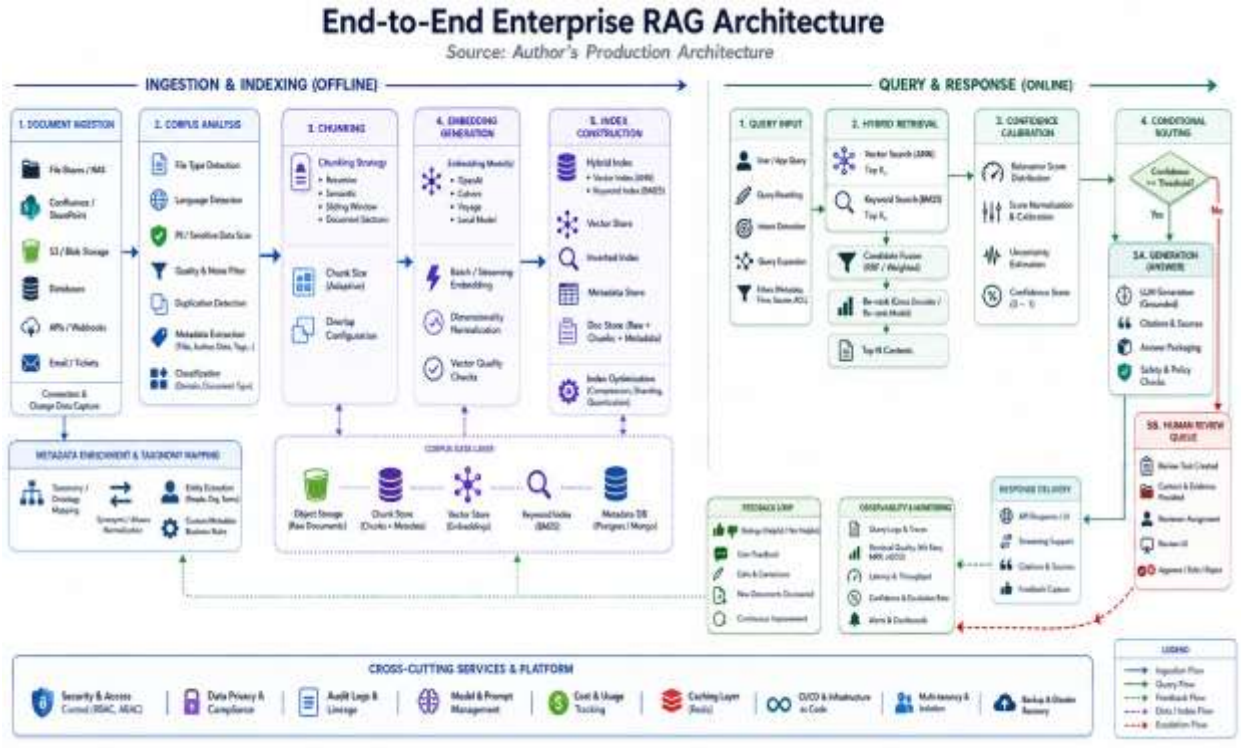

Figure 3. End-to-end enterprise RAG architecture.

7. Discussion

Every component of a RAG pipeline can be optimized in isolation, but the system-level performance is determined by how all three components interact with the specific document corpus and query distribution. This observation has a practical consequence: organizations that deploy RAG using infrastructure defaults — because the system works well enough in initial testing — often encounter gradual accuracy degradation as document volume grows, query diversity increases, and the mismatch between default configurations and actual workload characteristics compounds. The framework proposed here is designed to surface that mismatch before it compounds through corpus analysis and validation query set construction at deployment time [5][8].

Three limitations bound the applicability of the framework. Per-field calibration requires correction event history; cold-start deployments should expect a calibration latency period of several processing batches before per-field thresholds are reliable. Structure-aware chunking requires parseable document formats; scanned or poorly OCR'd documents may degrade chunking quality regardless of strategy, and organizations with significant scanned document volumes should invest in document preprocessing infrastructure before applying structure-aware chunking logic [23]. Hybrid search configuration requires a representative validation query set; for genuinely novel application domains, this step may not be feasible before launch, and pure dense retrieval should be the default in those cases until a validation set can be constructed from production usage patterns [12][13].

Future directions for the framework include automated chunking strategy selection based on corpus analysis outputs — reducing the manual corpus examination step to a structured preprocessing pipeline — and multimodal retrieval extensions that incorporate figure content, schematic diagrams, and tabular images into the retrieval index alongside text. Both directions are areas of active research [16] and are likely to become practically relevant as enterprise corpora increasingly include structured visual content that current text-only chunking and retrieval approaches cannot index effectively[23] [24].

8. Conclusion

The binding constraint in enterprise RAG systems is retrieval quality, and retrieval quality is determined by three design decisions that most production deployments leave at defaults. Structure-aware chunking preserves structural meaning that fixed-size approaches destroy; the empirical evidence from enterprise document corpora demonstrates that this preservation translates directly into measurable top-K retrieval effectiveness gains across technical and regulatory content types [10][11]. Hybrid dense-sparse indexing captures retrieval signals that pure vector search misses, recovering exact-match query coverage through BM25 fusion without sacrificing the semantic matching capability that dense retrieval provides. Per-field confidence calibration routes review traffic proportionally to actual error rates rather than uniformly applying a threshold that systematically misclassifies field types at opposite extremes, enabling the up to 38% reduction in false-positive review traffic that production deployments demonstrate when calibration is implemented correctly.

This sequenced, three-layer optimisation framework — corpus analysis, chunking selection, index architecture, hybrid search configuration, and per-field calibration — is practitioner-validated across production enterprise knowledge retrieval applications and provides a repeatable decision process for configuring enterprise RAG systems that must meet latency, accuracy, and cost requirements in mission-critical deployments. The framework's primary contribution is not the introduction of novel techniques but the sequencing and integration of established techniques into a deployment process that accounts for the interactions between chunking, retrieval, and calibration decisions that default configurations ignore.

References

[1] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. Advances in Neural Information Processing Systems, 33, 9459–9474. https://arxiv.org/abs/2005.11401

[2] Karpukhin, V., Oğuz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., & Yih, W. (2020). Dense passage retrieval for open-domain question answering. Proceedings of EMNLP 2020, 6769–6781. https://arxiv.org/abs/2004.04906

[3] Johnson, J., Douze, M., & Jégou, H. (2021). Billion-scale similarity search with GPUs. IEEE Transactions on Big Data, 7(3), 535–547. https://arxiv.org/abs/1702.08734
[4] Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. Proceedings of EMNLP-IJCNLP 2019, 3982–3992. https://arxiv.org/abs/1908.10084
[5] Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Wang, M., & Wang, H. (2023). Retrieval-augmented generation for large language models: A survey. arXiv:2312.10997. https://arxiv.org/abs/2312.10997
[6] Maximilian Stäbler, Steffen Turnbull, Tobias Müller, Chris Langdon, Jorge Marx-Goméz, & Frank Köster. (2024). The impact of chunking strategies on domain-specific information retrieval in RAG systems. IEEE Conference Publication. https://ieeexplore.ieee.org/abstract/document/11125724/
[7] Zhu, Y., Yuan, H., Wang, S., Liu, J., Liu, W., Deng, C., Dou, Z., & Wen, J. (2024). A survey on RAG meeting LLMs: Towards retrieval-augmented large language models. Proceedings of ACM SIGKDD 2024. https://dl.acm.org/doi/10.1145/3637528.3671470
[8] Chaitanya Sharma. (2025). Retrieval-augmented generation: A comprehensive survey of architectures, enhancements, and robustness frontiers. arXiv:2506.00054. https://arxiv.org/html/2506.00054v1
[9] Vatsal Raina & Mark Gales. (2024). Question-based retrieval using atomic units for enterprise RAG. arXiv:2405.12363. https://arxiv.org/abs/2405.12363
[10] Samuel Taiwo & Mohd Amaluddin Yusoff. (2026). Evaluating chunking strategies for retrieval-augmented generation in oil and gas enterprise documents. arXiv:2603.24556. https://arxiv.org/abs/2603.24556
[11] Pranav Pushkar Mishra, Kranti Prakash Yeole, Ramyashree Keshavamurthy, Mokshit Bharat Surana, & Fatemeh Sarayloo. (2024). A systematic framework for enterprise knowledge retrieval: Leveraging LLM-generated metadata to enhance RAG systems. arXiv:2512.05411. https://arxiv.org/abs/2512.05411
[12] Anuj Maharjan & Umesh Yadav. (2025). Chunking, retrieval, and re-ranking: An empirical evaluation of RAG architectures for policy document question answering. arXiv:2601.15457. https://arxiv.org/abs/2601.15457
[13] Hai Toan Nguyen, Tien Dat Nguyen & Viet Ha Nguyen. (2025). Enhancing retrieval augmented generation with hierarchical text segmentation chunking. arXiv:2507.09935. https://arxiv.org/abs/2507.09935
[14] Chandana Cheerla. (2025). Advancing retrieval-augmented generation for structured enterprise and internal data. arXiv:2507.12425. https://arxiv.org/abs/2507.12425
[15] Shangyu Wu et al., (2024). Retrieval-augmented generation for natural language processing: A survey. arXiv:2407.13193. https://arxiv.org/html/2407.13193v1
[16] Mahmoud Amiri & Thomas Bocklitz. (2025). Chunk twice, embed once: A systematic study of segmentation and representation trade-offs in chemistry-aware RAG. arXiv:2506.17277. https://arxiv.org/abs/2506.17277
[17] Fuda Ye, Shuangyin Li, Yongqi Zhang & Lei Chen. (2024). R^2AG: Incorporating retrieval information into retrieval-augmented generation. arXiv:2406.13249. https://arxiv.org/abs/2406.13249
[18] Simeon Emanuilov & Aleksandar Dimov. (2024). Billion-scale similarity search using a hybrid indexing approach with advanced filtering. Cybernetics and Information Technologies. https://dl.acm.org/doi/abs/10.2478/cait-2024-0035
[19] Ahmad Dawar Hakimi, Lea Hirlimann, Isabelle Augenstein & Hinrich Schütze. (2026). Do we still need humans in the loop? Comparing human and LLM annotation in active learning for hostility detection. arXiv:2604.13899. https://arxiv.org/html/2604.13899v2
[20] Ekaterina Artemova, Akim Tsvigun, Dominik Schlechtweg, Natalia Fedorova, Sergei Tilga, Konstantin Chernyshev & Boris Obmoroshev (2024). Hands-on tutorial: Labeling with LLM and human-in-the-loop. arXiv:2411.04637. https://arxiv.org/html/2411.04637v3
[21] M. K. Babu and Y. Suthari, "Data privacy: Strategies for protecting sensitive data for OT using artificial intelligence," Computer Fraud & Security, Special Issue, 2024. [Online]. Available: https://doi.org/10.52710/cfs.628.
[22] N. Nellutla, "Secure DevSecOps workflows for medical IoT device integration in smart hospitals," International Journal of AI, BigData, Computational and Management Studies, vol. 3, no. 1, pp. 114–122, 2022. [Online]. Available: https://doi.org/10.63282/3050-9416.IJAIBDCMS-V3I1P113.
[23] R. Gollapudi, "Operational drift and risk-bounded decision-making in production database systems," Journal of International Crisis and Risk Communication Research, pp. 132–147, 2023. [Online]. Available: https://doi.org/10.63278/jicrcr.vi.3762.
[24] N. Nellutla et al., "AutoPilot AI: Architecting self-healing ML systems with reinforcement feedback loops," in Proceedings of the 2025 IEEE 2nd International Conference on Information Technology, Electronics and

Intelligent Communication Systems (ICITEICS), pp. 1–8, Aug. 2025. [Online]. Available: https://doi.org/10.1109/ICITEICS64870.2025.11341204.