



International Journal of Artificial Intelligence and Machine Learning

Publisher's Home Page: <https://www.svedbergopen.com/>



Research Paper

Open Access

Retrieval-Augmented Generation as a Foundation for Trusted Enterprise Artificial Intelligence: Architectural Principles for Accuracy, Explainability, and Operational Adoption

Ravindra Patil

Independent Researcher, USA.

ABSTRACT

A majority of organizations now report regular use of generative artificial intelligence in at least one business function, and enterprise adoption has accelerated accordingly. Trust in large language model outputs has not kept pace. Hallucination, weak provenance, and reasoning chains that resist inspection continue to limit generative systems in regulated and high-stakes workflows, even as pressure to deploy them grows. Retrieval-Augmented Generation (RAG) has become the leading architectural response to this problem: rather than depending solely on parametric memory, RAG grounds model outputs in retrieved, verifiable evidence. This paper treats RAG not as a single technique but as an architectural foundation for enterprise trust, tracing its path from naive retrieve-then-generate pipelines to governed, modular architectures that build in evaluation, attribution, and auditability from the outset. Drawing on the enterprise data architecture literature, the RAG evaluation literature, and recent industry survey evidence, the paper organizes governance-relevant RAG design choices into a maturity framing, compares two leading automated evaluation frameworks, and examines the organizational barriers that determine whether technically sound RAG systems achieve real operational adoption. It contributes a structured account of how retrieval architecture, evaluation practice, and enterprise trust relate to one another, intended for practitioners building production RAG systems and researchers studying enterprise AI governance alike.

Keywords: Retrieval-Augmented Generation, Enterprise Artificial Intelligence, Hallucination Mitigation, AI Governance, Explainable AI, Data Platform Architecture

I. Introduction

Generative artificial intelligence moved from experimental pilots to embedded enterprise infrastructure in a remarkably short span. Survey evidence indicates that 88 percent of organizations now report regular AI use in at least one business function, up from 78 percent a year earlier, and industry forecasts project that more than 80 percent of enterprises will have used generative AI application programming interfaces or deployed generative AI-enabled applications by 2026, compared with less than 5 percent in 2023 [1], [2]. These numbers reflect real enterprise appetite for the productivity and decision-support gains large language models (LLMs) promise. They do not, by themselves, show that enterprises trust these systems the way trust is normally understood in regulated, auditable, or safety-relevant settings.

The same survey data that describes adoption also exposes the gap between adoption and trust. Among AI-using organizations, 51 percent report at least one negative consequence arising from AI use, while only 39 percent report enterprise-wide EBIT impact from their AI investments [1]. Adoption has outpaced governance. Enterprises are pushing generative systems into workflows faster than the mechanisms needed to verify, explain, and audit those systems' outputs have been able to mature.

Retrieval-Augmented Generation is the most widely adopted architectural answer to that gap. Instead of relying only on knowledge encoded in model parameters during pretraining, RAG systems retrieve relevant evidence from an external corpus at inference time and condition generation on what they retrieve [3]. The approach was introduced as a way to combine parametric and non-parametric memory for knowledge-intensive tasks, and later work extended the idea to retrieval-augmented pretraining and dense retrieval methods that improved open-domain question answering by a wide margin [3], [4], [5]. For enterprise practitioners, RAG's appeal goes beyond accuracy: it offers a plausible route to provenance, since retrieved passages can in principle be shown to end users and auditors as evidence behind a generated claim.

Adopting a RAG architecture does not automatically deliver on that promise. Retrieval alone does not guarantee faithful grounding, and citation-style attribution in RAG outputs has been shown to diverge from genuine

evidential support in a substantial share of cases [6]. An enterprise that treats RAG as a drop-in fix for hallucination, without attending to retrieval quality, evaluation, and auditability, risks reproducing the same trust deficit in a more convincing disguise.

This paper argues that RAG works best as an architectural foundation for enterprise AI trust, not a single technique, and that building on that foundation requires attention to retrieval design, evaluation methodology, explainability mechanisms, and organizational governance together. Four contributions follow:

1. The paper synthesizes the enterprise trust problem in generative AI, connecting adoption statistics, hallucination evidence, and governance literature into a single account of why grounding matters operationally and not only technically.
2. It presents a maturity framing distinguishing naive, advanced, and modular RAG architectures along governance-relevant dimensions, including retrieval strategy, grounding mechanism, evaluation approach, and characteristic failure modes.
3. It compares the two leading automated RAG evaluation frameworks, RAGAS and ARES, treating them as governance instruments rather than purely as research benchmarks, and clarifies what each metric measures and how it is computed.
4. It examines explainability, auditability, and organizational adoption barriers together, arguing that technical grounding and institutional trust depend on each other rather than arriving in sequence.

Section II develops the enterprise trust problem in more detail. Section III describes the architectural principles underlying RAG. Section IV presents the naive-to-governed maturity framing with an accompanying comparison table, and Section IV-A applies that framing as a secondary analytical lens to industry interview evidence on enterprise RAG deployments. Section V examines evaluation as a governance mechanism, including a comparison of RAGAS and ARES. Section VI addresses explainability and auditability in enterprise deployments. Section VII considers operational adoption barriers. Section VIII discusses a governed RAG adoption model and its limitations, and Section IX concludes.

A methodological note on scope is warranted before proceeding. This paper is a narrative synthesis of the enterprise RAG literature rather than a systematic review; sources were selected for their direct relevance to the trust, evaluation, and governance themes developed here rather than through a formal systematic search protocol (e.g., predefined search strings, database coverage criteria, or PRISMA-style screening). This choice suits the paper's goal of building an integrated conceptual account across technical and organizational literatures that have largely developed on separate tracks, but it also means the coverage claims that follow should be read as illustrative of documented patterns rather than as an exhaustive census of the field.

II. THE ENTERPRISE TRUST PROBLEM IN GENERATIVE AI

Enterprise software has historically earned trust through deterministic behavior: a given input produces a predictable, traceable output, and errors can be reproduced and root-caused. Generative AI systems built on LLMs break this expectation, and not incidentally. Language models generate text by sampling from a learned probability distribution over tokens, and the same capability that produces fluent, contextually appropriate responses also produces fluent, contextually appropriate fabrications. Enterprises evaluating these systems for production use have found hallucination to be a measurable, recurring failure mode with real consequences, not a rare edge case.

The clearest evidence comes from domains where factual precision carries direct professional or financial risk. An assessment of leading AI legal research tools found that RAG-based legal AI products from LexisNexis and Thomson Reuters each hallucinate between 17 percent and 33 percent of the time, despite being marketed specifically on the basis of retrieval-grounded reliability [7]. These are not naive LLM deployments. They are commercial products built explicitly around retrieval augmentation, and the residual hallucination rate shows that retrieval mitigates the underlying trust problem without eliminating it.

A related finding concerns the reliability of citations themselves. Even when a RAG system produces an answer with source attribution attached, the cited passages do not reliably support the claims they accompany. Research examining citation faithfulness in attributed RAG outputs found that up to 57 percent of citations lacked genuine faithfulness to the claims they were meant to support [6]. The distinction between correctness and faithfulness matters for enterprise governance: an answer can be factually correct while its stated evidence trail is fabricated or mismatched, and an answer can cite real passages that do not actually support the generated claim. Neither failure is visible to an end user who is not independently checking the underlying source documents, which is often the exact burden RAG was supposed to remove.

Enterprise data leadership has typically approached trust through data governance frameworks built around lineage, access control, and quality validation applied to structured data pipelines. Generative AI introduces a parallel set of concerns: the trustworthiness of unstructured retrieval corpora, the integrity of embedding

pipelines, and the fidelity of the language model's synthesis step. Recent survey work on trustworthy RAG systems has organized these concerns into a broader set of dimensions spanning reliability, safety, fairness, and accountability [8], while broader RAG surveys have documented how quickly architectural variants have proliferated to address different points of failure in the retrieval-generation pipeline [9], [10]. For enterprise architects, the practical implication is that trust cannot be delegated to the language model vendor, nor can it be treated as something that emerges automatically once a retrieval step is added. It has to be engineered, measured, and governed across the whole pipeline, from source data ingestion through retrieval, generation, and presentation to the end user.

The rest of this paper treats that observation as its organizing premise: RAG is necessary but not sufficient for enterprise trust, and the sections that follow examine what additional architectural, evaluative, and organizational structure is needed to close the gap.

III. ARCHITECTURAL PRINCIPLES OF RETRIEVAL-AUGMENTED GENERATION

RAG architectures combine two components that were historically developed separately: a retriever, which identifies relevant evidence from an external corpus, and a generator, which produces fluent output conditioned on that evidence. The original formulation paired a pretrained sequence-to-sequence generator with a dense vector index over Wikipedia passages, retrieving the top-k most relevant passages for a given query and marginalizing over them during generation [3]. This design showed that non-parametric memory, refreshed by retrieval rather than baked into model weights, could improve performance on knowledge-intensive tasks while keeping the model's evidence base inspectable in principle.

A parallel line of work built retrieval directly into the language model pretraining objective. REALM augmented masked language model pretraining with a learned retriever, letting the model attend to retrieved documents when predicting masked tokens, and demonstrated accuracy improvements of 4 to 16 percentage points absolute over prior open-domain question answering methods [4]. This result established that retrieval quality and generation quality are not independent: a model trained to use retrieved evidence performs measurably differently from one that only receives retrieved context as an inference-time addition.

Retrieval quality itself depends heavily on how passages are represented and indexed. Dense Passage Retrieval showed that dense vector embeddings, trained with a dual-encoder architecture to place semantically related questions and passages close together in embedding space, outperformed traditional sparse retrieval methods such as BM25 by 9 to 19 percentage points absolute in top-20 passage retrieval accuracy [5]. This finding underlies the semantic similarity scoring most enterprise RAG deployments now rely on. Rather than matching keywords, the retriever computes a similarity score, typically cosine similarity or dot product, between a query embedding and a corpus of document embeddings stored in a vector database.

Standardized benchmarking of these components matters for enterprise adoption because retrieval and generation improvements do not always transfer across task types. The KILT benchmark was introduced to unify evaluation across a range of knowledge-intensive tasks, including open-domain question answering, fact-checking, and dialogue, using a single underlying knowledge source. This allows retrieval and generation components to be compared on consistent grounds rather than through task-specific leaderboards that obscure whether gains come from better retrieval or better generation [11].

At the architectural level, an enterprise-grade RAG pipeline can be broken into five stages: ingestion, embedding, retrieval, augmentation, and generation. Ingestion covers acquisition and preprocessing of source documents, including chunking strategy, metadata tagging, and access control propagation from the source system. Embedding converts document chunks into vector representations using an embedding model; the choice of chunk size and embedding model materially affects downstream retrieval quality. Retrieval runs similarity search against the vector index, often combined with metadata filters that enforce data governance constraints such as document sensitivity classification or business unit ownership. Augmentation assembles the retrieved passages into a prompt context, frequently applying re-ranking, deduplication, or context compression before passing the assembled context to the generator. Generation produces the final response, ideally with references back to the retrieved source passages that supported it.

Each stage introduces its own failure surface. Because the stages are coupled, a well-designed generator cannot compensate for a poorly designed retriever, and the reverse holds too. This structural interdependence, established across the foundational RAG literature, motivates the maturity framing developed in the next section, where governance is treated as a property distributed across the pipeline rather than concentrated in any single component.

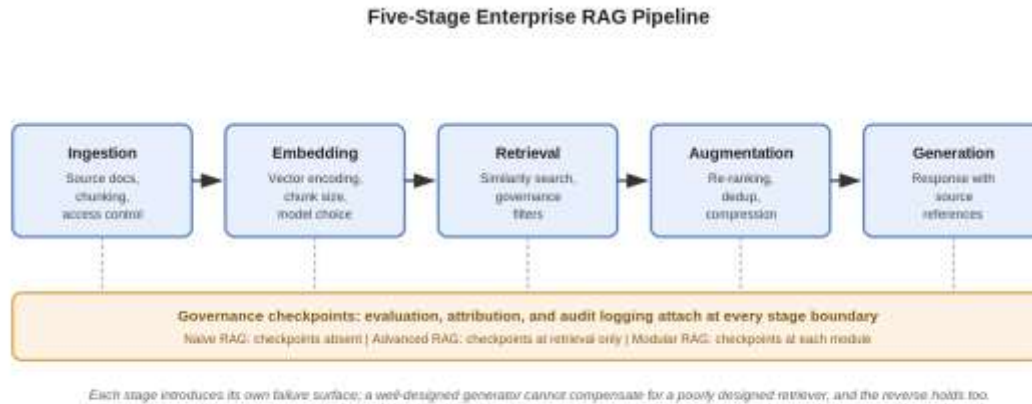


Fig. 1. The five-stage RAG pipeline (ingestion, embedding, retrieval, augmentation, generation) with governance checkpoints distributed across stage boundaries rather than concentrated in one component.

Fig. 1. The five-stage RAG pipeline (ingestion, embedding, retrieval, augmentation, generation) with governance checkpoints distributed across stage boundaries rather than concentrated in one component.

IV. FROM NAIVE TO GOVERNED RAG: A MATURITY FRAMING

Not all systems described as RAG are architecturally equivalent, and the differences among them matter directly for enterprise trust. One useful way to organize this variation, consistent with how the broader RAG survey literature has categorized the field, is a three-stage maturity progression: naive RAG, advanced RAG, and modular RAG [9], [10].

Naive RAG describes the original retrieve-then-generate pattern. A query is embedded, the top-k nearest passages are retrieved by similarity search, and those passages are concatenated into the generation prompt with little further processing. This pattern is straightforward to implement, which is why most enterprise proof-of-concept RAG projects start here. Its failure modes are well documented as a result. Retrieval can return passages that are topically similar to the query but do not contain the information needed to answer it. Generation can ignore retrieved context in favor of parametric knowledge. And there is typically no mechanism to catch either failure before the response reaches an end user.

Advanced RAG introduces optimization at the retrieval and pre-generation stages without changing the underlying linear pipeline structure. Techniques at this stage include query rewriting or expansion prior to retrieval, re-ranking of retrieved passages with a secondary cross-encoder model, and context compression to strip irrelevant content before it reaches the generator. These address specific weaknesses of naive retrieval, particularly the tendency of pure dense similarity search to surface passages that are semantically adjacent but not evidentially sufficient. Evaluation in advanced RAG systems still tends to focus on retrieval metrics, measured through passage-level precision and recall, with generation-side faithfulness checked only informally or through spot-checking.

Modular RAG generalizes the pipeline into a set of interchangeable, independently governable components, often organized as a graph rather than a linear chain. Modular architectures support routing logic that selects among multiple retrieval sources, iterative or multi-hop retrieval where an initial retrieval and generation step informs a subsequent retrieval query, and explicit verification modules that check generated claims against retrieved evidence before finalizing a response. Recent work on agentic RAG pushes this modularity further, framing retrieval and generation steps as actions available to an autonomous agent that plans, retrieves, and revises across multiple steps rather than executing one fixed pipeline [12]. Modular and agentic designs bring new failure surfaces too, including compounding retrieval errors across multi-hop steps and added latency and cost, but they also let evaluation and governance attach at each module boundary rather than only at the final output.

Table I summarizes this progression across the dimensions most relevant to enterprise governance: retrieval strategy, grounding mechanism, evaluation approach, and characteristic failure modes.

TABLE I. FROM NAIVE TO GOVERNED RAG: A MATURITY FRAMING

Dimension	Naive RAG	Advanced RAG	Modular RAG
-----------	-----------	--------------	-------------

Retrieval strategy	Single-pass top-k similarity search	Query rewriting/expansion; re-ranking with cross-encoder	Multi-source routing; iterative/multi-hop retrieval
Grounding mechanism	Concatenated passages with minimal processing	Re-ranked, compressed context prior to generation	Explicit verification modules checking claims against evidence
Evaluation approach	Little to no systematic evaluation	Retrieval-focused metrics (precision/recall); informal generation checks	Component-level evaluation at each module boundary
Characteristic failure modes	Typically similar but insufficient passages; generation ignoring context	Residual semantic-adjacency gaps; limited generation-side faithfulness checks	Compounding multi-hop errors; added latency and cost

The progression from naive to modular RAG is not just a technical refinement exercise. Each additional layer of architectural sophistication adds a point where governance controls, evaluation checkpoints, and audit logging can be inserted. Enterprises that adopt naive RAG because it is fastest to deploy often discover the governance cost later, when incident review or regulatory inquiry demands evidence of why a particular output was generated and no such trail exists. Adversarial robustness becomes more tractable as architectures mature, too: research on RAG systems under adversarial poisoning attacks has shown that retrieval corpora can themselves be manipulated to induce targeted hallucinations, a risk that modular architectures with source verification are better positioned to catch than naive pipelines [13].

Treating this progression as a maturity model, rather than a menu of interchangeable options, gives enterprise architects a basis for sequencing investment: retrieval quality and evaluation infrastructure first, then modular governance controls, rather than deploying agentic capability before the underlying grounding and measurement foundation is in place.

IV-A. APPLYING THE MATURITY FRAMING: A SECONDARY ANALYSIS OF INDUSTRY INTERVIEW EVIDENCE

Table I organizes naive, advanced, and modular RAG as a taxonomy. Its value increases if the dimensions it defines, retrieval strategy, grounding mechanism, evaluation approach, and characteristic failure modes, can also function as a scoring instrument applied to real deployments rather than only as a descriptive classification. This subsection applies that instrument to the enterprise RAG deployments described in Brehme et al.'s interview study of practitioners across industry sectors [14]. It is important to state plainly what this is and is not. This is a secondary, qualitative analysis of findings already published in [14] and already paraphrased elsewhere in this paper (Sections VI and VII), read through the lens of Table I's four dimensions. It is not new primary data collection, and no companies, quotes, or deployment details beyond what [14] reports and this paper has already cited are introduced here. Where the interview evidence available through the existing citations does not speak directly to a Table I dimension, that gap is noted rather than filled by inference.

On retrieval strategy and grounding mechanism, the practitioner interviews summarized in [14] describe systems oriented toward domain-specific question-answering tasks, with deployments that the interviewees themselves characterize as still at prototype or early-production stages rather than fully matured pipelines. That description is more consistent with the naive-to-advanced range of Table I, single-pass or lightly optimized retrieval feeding a generation step, than with the multi-source routing, iterative retrieval, or explicit claim-verification modules that define modular RAG in this framing. Nothing in the existing citations of [14] used in this paper describes interviewees running multi-hop retrieval or agentic verification loops in production; the reported concerns cluster instead around getting retrieval and grounding to work reliably on a single domain corpus.

On evaluation approach, the pattern is similarly consistent with the earlier maturity stages in Table I. The interview findings already cited in Section VII describe a persistent gap between the evaluation metrics researchers favor and the criteria business stakeholders actually use, together with difficulty defining acceptance criteria that satisfy both groups [14]. A gap of that kind is what Table I predicts for naive and advanced RAG, where evaluation is described as little to no systematic evaluation or informal, retrieval-focused checks, rather than the component-level, per-module evaluation regime that characterizes modular RAG. The interview evidence cited here does not describe interviewees running automated frameworks comparable to RAGAS or ARES (Section V) as a routine practice; acceptance criteria appear to be negotiated case by case rather than instrumented continuously.

On characteristic failure modes, the recurring practitioner concerns already cited in this paper, data quality and freshness in the retrieval corpus and unclear ownership of corpus maintenance once a system moves past pilot stage [14], map most directly onto the naive and advanced RAG failure modes in Table I: topically similar but insufficient retrieved passages, and residual gaps that ad hoc re-ranking or compression do not fully close. The compounding multi-hop errors and added latency that Table I associates with modular RAG are not features the cited interview findings describe, which is itself consistent with deployments that have not yet reached that architectural stage.

Read together, this qualitative application of Table I to the industry interview evidence in [14] suggests that a meaningful share of real enterprise RAG deployments described by practitioners cluster in the naive-to-advanced range of the maturity framing, with evaluation practice lagging architecture, rather than at the modular, continuously governed end this paper argues enterprises should be sequencing toward. This is offered as a modest, transparent strengthening of Table I from a pure taxonomy into an applied analytical lens, not as a validated empirical test of the framework; the sample underlying [14] is a single interview study, and a scoring exercise of this kind would benefit considerably from replication against deployments observed directly rather than through a secondary reading of published interview findings.

V. EVALUATION AS A GOVERNANCE MECHANISM

Retrieval architecture sets the boundaries of what evidence a RAG system can draw on. Evaluation is what tells an enterprise, in a documented and repeatable way, whether that evidence was actually used correctly. Manual review of RAG outputs does not scale to production query volumes, and human review alone cannot supply the continuous monitoring that governance frameworks in regulated industries tend to require. That gap is what has pushed development of automated RAG evaluation frameworks capable of scoring retrieval and generation quality without a human-labeled reference answer for every query.

RAGAS (Retrieval Augmented Generation Assessment) was introduced as a framework for evaluating RAG pipelines along dimensions that skip ground-truth human annotation, relying instead on an auxiliary LLM to judge properties of the generated answer against the retrieved context and the original question [15]. The framework breaks evaluation into component metrics: faithfulness, which checks whether claims in the generated answer are backed by the retrieved context, and answer relevance, which checks whether the generated answer actually addresses the question asked, regardless of whether it is factually grounded.

ARES (Automated Evaluation of RAG Systems) takes a related but methodologically different path. It trains lightweight classifiers on a small set of human-annotated examples combined with synthetic data to predict evaluation scores, then applies prediction-powered inference to attach statistical confidence bounds to those predictions [16]. The intent was to cut reliance on a general-purpose LLM as judge for every single evaluation call, swapping in a task-specific classifier that can be calibrated to a particular domain corpus.

What is striking, for enterprise governance purposes, is that both frameworks converge on a similar set of core evaluation dimensions despite computing them differently. Faithfulness, answer relevance, and context precision or recall have become something close to a shared vocabulary for RAG quality assessment across the literature [15], [16], [17]. Table II lays out how each framework defines and computes these shared metrics.

TABLE II. RAGAS AND ARES: SHARED EVALUATION DIMENSIONS

Metric	RAGAS Definition/Method	ARES Definition/Method
Faithfulness	LLM-judge checks whether generated claims are backed by retrieved context	Lightweight classifier trained on human-annotated + synthetic data, calibrated per domain

Answer relevance	LLM-judge checks whether the answer addresses the question, independent of grounding	Classifier-based relevance prediction with prediction-powered inference confidence bounds
Context precision/recall	Reference-free scoring against retrieved context via auxiliary LLM	Classifier-based scoring calibrated to task-specific corpus
Underlying method	General-purpose LLM as judge for every evaluation call	Task-specific classifier reducing reliance on per-call LLM judgment

Neither framework fully solves RAG evaluation, and both carry the limitations of the underlying LLM-as-judge or classifier-based approach they rest on. A recent survey of RAG evaluation in the era of large language models catalogs a much wider range of metric families beyond faithfulness and relevance, covering factual consistency, answer completeness, and robustness under distribution shift, and observes that no single framework currently handles all of these with equal reliability [17]. Explainability-oriented extensions of automated evaluation are also starting to appear, pushing past a bare numeric score toward actionable diagnosis of which pipeline component, retrieval or generation, caused a given failure. That kind of output is considerably more useful for governance than an aggregate quality score [18].

For enterprise governance, the real value of RAGAS and ARES has less to do with their absolute metric values, which shift depending on domain and prompt design, and more to do with their role as continuous, repeatable instrumentation. Wiring these frameworks into a production monitoring pipeline lets an enterprise catch faithfulness degradation after a corpus update, a model version change, or a retrieval configuration change, ideally before a customer-facing incident surfaces it. Seen this way, evaluation acts as a governance control not unlike data quality monitoring in a conventional data platform, rather than a one-time benchmarking exercise run before launch and then forgotten.

VI. EXPLAINABILITY AND AUDITABILITY IN ENTERPRISE DEPLOYMENTS

Evaluation metrics tell you how well a RAG system performs in aggregate. Explainability and auditability answer a different, more consequential question for enterprise deployment: what evidence backs a specific output, and can the trail be reconstructed after the fact. Regulated industries tend to treat this as a precondition for production use, separate from aggregate accuracy figures.

Attribution, in the RAG context, means linking specific claims in a generated answer to the retrieved passages that support them, not simply listing retrieved documents alongside it. A recent survey catalogs attribution techniques for mitigating hallucinated information in RAG systems, from post-hoc citation generation to attention-based methods tracing which tokens most influenced generated output [19]. Its central observation is a design constraint, not a footnote: attribution mechanisms vary widely in reliability. A citation being present does not mean it is faithful, a distinction the citation faithfulness research in Section II already established [6], [19].

Auditability takes attribution from a single response to the level of the system over time. A properly auditable RAG deployment records which corpus, embedding model, retrieval configuration, and generation model version produced an output, along with retrieved passages and evaluation scores. This is a data lineage problem borrowed from structured data governance, extended to cover the artifacts a RAG pipeline introduces: embedding vectors, retrieval logs, prompt templates, model version identifiers. Metadata management is therefore core to surviving post-incident review or audit.

Interview-based industry research on RAG deployment confirms these auditability requirements are not abstract concerns invented by researchers. Traceability, evaluation, and evidence-quality requirements recur across use cases and sectors, alongside practitioner worries about the reliability of retrieved evidence [14]. That finding lines up with the trustworthy RAG survey literature, which places accountability among the core dimensions for judging enterprise RAG systems [8].

Explainability also has a presentation dimension. Surfacing retrieved passages only helps if the user can quickly tell whether a passage supports the claim attributed to it. That argues for interface choices, such as highlighting the relevant span, over dumping an undifferentiated document list. Explainable evaluation frameworks that move toward actionable, component-level diagnosis, rather than a single aggregate score, represent one step toward operational explainability. They identify not just that a failure occurred, but which stage of the pipeline is responsible [18]. Enterprises building RAG systems for auditable domains should treat that diagnostic capability, not the mere presence of a citation, as the bar for explainability.

VII. OPERATIONAL ADOPTION: GOVERNANCE, RISK, AND ORGANIZATIONAL BARRIERS

Sound architecture does not automatically translate into operational adoption. Enterprise RAG initiatives frequently stall before scaling past pilot stage for organizational, not architectural, reasons. The maturity framing and evaluation instrumentation covered earlier address only part of what determines sustained production adoption.

Interview-based research into industry RAG deployments identifies recurring cross-sector challenges: data quality and freshness in the retrieval corpus, and unclear corpus-maintenance ownership after a system moves to production [14]. A further challenge is defining acceptance criteria that satisfy both technical and business stakeholders, since a gap remains between researcher-favored metrics and stakeholder trustworthiness criteria. Improving retrieval precision or fluency does not fix this; these are governance problems adjacent to the pipeline, not inside it.

The negative-consequence data cited earlier matters here. Slightly more than half of AI-using organizations report at least one negative consequence from AI use, while a much smaller share report enterprise-wide financial impact [1]. These organizations are not a small, separable subset of careless adopters; they make up a substantial share of the adopting population. Risk and value realization run in parallel, not confined to those that failed to capture value.

Governance structures that would normally catch this, such as model risk management, AI ethics review, and data governance, are typically built around traditional statistical models or structured pipelines. Neither maps cleanly onto a RAG system combining a vector store, a retrieval service, and an LLM. Assigning clear ownership for corpus curation, embedding version control, and evaluation monitoring requires roles many enterprises have not formalized. The absence of a named owner for corpus quality is a recurring failure independent of any technical shortfall.

Adversarial and security considerations make the organizational barrier worse. RAG systems retrieve from large, evolving corpora, inheriting a supply chain risk traditional deployment does not carry. A poisoned document introduced into the retrieval corpus can bias generated outputs, a vulnerability demonstrated in controlled studies of RAG systems under adversarial poisoning attack [13]. Enterprises with mature access control for structured data often lack an equivalent process for unstructured documents feeding a vector store, echoing the gaps discussed in Section VI.

Reviews of hallucination mitigation techniques for retrieval-augmented language models reach a similar conclusion from the technical side. Mitigation works best when treated as a property of the full pipeline and its operational controls, not any single model or retrieval component [20]. Organizational adoption barriers are thus not separate from the hallucination and faithfulness issues discussed earlier; they are the institutional expression of the same gap between deploying and governing a capability.



Fig. 2. Organizational adoption barriers separating a technically governed RAG pilot from sustained production adoption, resolved through named ownership rather than further architectural refinement.

Fig. 2. Organizational adoption barriers separating a technically governed RAG pilot from sustained production adoption, resolved through named ownership rather than further architectural refinement.

Viii. Discussion: toward a governed rag adoption model

Taken together, the material reviewed across Sections II through VII points to a common conclusion: enterprise trust in generative AI is not delivered by any single fix. It comes from aligning retrieval architecture, evaluation, explainability, and governance around a shared standard of evidence. A governed RAG adoption model treats the maturity progression from naive to modular RAG (Section IV) as a technical prerequisite. It treats RAGAS and ARES (Section V) as continuous instrumentation, and attribution and lineage mechanisms (Section VI) as the bridge to auditability. Named organizational ownership (Section VII) keeps the first three from decaying after deployment.

The strongest evidence for this synthesis is consistency across otherwise separate literatures. Hallucination and faithfulness research [6], [19], [20], [7], evaluation methodology research [15], [16], [17], [18], and industry interview research [14] all converge: retrieval alone is an incomplete guarantee of trustworthiness. Measurement, attribution, and accountability must be engineered on purpose. That convergence strengthens confidence in the maturity framing proposed here, though it stops short of experimental validation.

Several limitations deserve stating directly, along with what would be needed to resolve each. First, the maturity framing in Section IV is a synthesis of patterns documented across the cited literature, not a framework validated through a controlled comparative study across the three maturity stages on a common enterprise workload; resolving this would require a multi-organization field study that scores matched deployments against Table I and tracks governance outcomes over time. Second, its practical value lies in giving practitioners vocabulary and sequencing guidance rather than quantitative benchmarks; closing that gap would require benchmark datasets and shared scoring rubrics that let independent teams apply the framework consistently and compare results. Third, the evaluation comparison in Section V reflects the published design of RAGAS and ARES at the time of the cited studies, and both frameworks, along with the broader evaluation landscape they sit inside, keep evolving; a living, version-tracked comparison, updated as these tools release new metrics, would be needed to keep the comparison current rather than a one-time snapshot. Fourth, the organizational barriers discussed in Section VII draw substantially on a single interview-based industry study [14], supplemented by adoption survey data [1], [2]; broader or longitudinal organizational research, ideally replicating the interview protocol across additional sectors and geographies, would strengthen this part of the argument considerably and test whether the reported patterns generalize beyond the practitioners already sampled. This paper also stops short of domain-specific regulatory requirements, which vary by sector and jurisdiction and would need separate treatment beyond the general architectural and governance principles addressed here.

Even with those limitations, the practical implication for enterprise architects and data leaders is direct. RAG investment should be sequenced around governance readiness rather than the fastest path to a demonstration. The gap between a functioning pilot and a trusted, auditable production system is a governance and evaluation gap, not a retrieval or generation capability gap.

IX. Conclusion

Retrieval-Augmented Generation has become the default architectural pattern for enterprises trying to ground generative AI outputs in verifiable evidence. The evidence reviewed in this paper supports treating that pattern as necessary infrastructure for enterprise trust rather than a complete solution on its own. The distance between a retrieval-augmented system and a trusted enterprise system is filled with specific, addressable design decisions: how retrieval, augmentation, and generation get decomposed and governed at each stage; whether evaluation is instrumented continuously through frameworks capable of measuring faithfulness and relevance rather than checked once before launch; whether attribution mechanisms are held to a standard of genuine evidential support rather than surface-level citation; and whether organizational ownership for corpus quality, evaluation monitoring, and incident review has actually been assigned to someone.

This paper offers four contributions as a structured vocabulary for practitioners and researchers working on this problem: a synthesis of the enterprise trust problem, a maturity framing distinguishing naive, advanced, and modular RAG, a comparison of RAGAS and ARES as governance instruments, and an integrated account of explainability and organizational adoption barriers. The technical and organizational literatures on RAG have largely developed on separate tracks rather than in conversation with each other, which is part of why that vocabulary is offered together here. Enterprise adoption statistics show that generative AI deployment is proceeding regardless of whether this vocabulary gets adopted, which raises the stakes for closing the gap between adoption and governance rather than lowering them.

Future work building on this account would benefit from controlled, longitudinal comparison of RAG maturity stages within single enterprise deployments, from evaluation frameworks that unify the current fragmented metric landscape documented across recent surveys, and from organizational research examining how

governance ownership for RAG systems gets assigned and sustained over the multi-year lifecycle these systems increasingly require. Architecture, evaluation, and governance are not sequential phases of a RAG program. On the evidence reviewed here, they are interdependent design commitments that together determine whether an enterprise generative AI system earns operational trust or merely operational deployment.

CREDIT AUTHOR CONTRIBUTIONS

Ravindra Patil: Conceptualization, Methodology, Investigation, Formal Analysis, Writing - Original Draft, Writing - Review & Editing, Visualization, Project Administration.

ETHICS STATEMENT

Conflict of Interest: The author declares no conflict of interest.

Statement of Human and Animal Rights: This study did not involve human participants, animal subjects, or identifiable personal data requiring ethics board review.

Informed Consent: Not applicable.

Funding: This research received no external funding.

Data Availability: Not applicable, this study synthesizes previously published literature and introduces no new primary dataset.

ACKNOWLEDGMENTS

The authors used Claude (Anthropic, claude-sonnet model) to assist with manuscript drafting and language refinement. The authors take full responsibility for the accuracy, originality, and integrity of all content presented in this manuscript, and confirm that all data, analysis, and conclusions reflect their own independent work and verified sources.

References

- [1] McKinsey & Company, "The State of AI in 2025: Agents, Innovation, and Transformation," Nov. 2025
- [2] Gartner, Inc., press release, Oct. 11, 2023
- [3] P. Lewis et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," NeurIPS 33, 2020. arXiv:2005.11401
- [4] K. Guu, K. Lee, Z. Tung, P. Pasupat, M.-W. Chang, "REALM: Retrieval-Augmented Language Model Pre-Training," ICML, 2020. arXiv:2002.08909
- [5] V. Karpukhin et al., "Dense Passage Retrieval for Open-Domain Question Answering," EMNLP, 2020, pp. 6769-6781. arXiv:2004.04906
- [6] J. Wallat, M. Heuss, M. de Rijke, A. Anand, "Correctness is not Faithfulness in Retrieval Augmented Generation Attributions," arXiv:2412.18004, 2024
- [7] V. Magesh, F. Surani, M. Dahl, M. Suzgun, C. D. Manning, D. E. Ho, "Hallucination-Free? Assessing the Reliability of Leading AI Legal Research Tools," arXiv:2405.20362, 2024
- [8] B. Ni et al., "Towards Trustworthy Retrieval Augmented Generation for Large Language Models: A Survey," arXiv:2502.06872, 2025
- [9] Y. Gao et al., "Retrieval-Augmented Generation for Large Language Models: A Survey," arXiv:2312.10997, 2023/2024
- [10] W. Fan et al., "A Survey on RAG Meeting LLMs: Towards Retrieval-Augmented Large Language Models," KDD, 2024. arXiv:2405.06211
- [11] F. Petroni et al., "KILT: A Benchmark for Knowledge Intensive Language Tasks," NAACL, 2021. arXiv:2009.02252
- [12] A. Singh, A. Ehtesham, S. Kumar, T. T. Khoei, A. V. Vasilakos, "Agentic Retrieval-Augmented Generation: A Survey on Agentic RAG," arXiv:2501.09136, 2025
- [13] J. Su et al., "Towards More Robust Retrieval-Augmented Generation: Evaluating RAG Under Adversarial Poisoning Attacks," arXiv:2412.16708, 2024
- [14] L. Brehme, B. Dornauer, T. Ströhle, M. Ehrhart, R. Breu, "Retrieval-Augmented Generation in Industry: An Interview Study on Use Cases, Requirements, Challenges, and Evaluation," KDIR, 2025. arXiv:2508.14066
- [15] S. Es, J. James, L. Espinosa-Anke, S. Schockaert, "RAGAS: Automated Evaluation of Retrieval Augmented Generation," EACL 2024. arXiv:2309.15217
- [16] J. Saad-Falcon, O. Khattab, C. Potts, M. Zaharia, "ARES: An Automated Evaluation Framework for Retrieval-Augmented Generation Systems," NAACL-HLT, 2024. arXiv:2311.09476

- [17] A. Gan et al., "Retrieval Augmented Generation Evaluation in the Era of Large Language Models: A Comprehensive Survey," arXiv:2504.14891, 2025
- [18] "RAGXplain: From Explainable Evaluation to Actionable Guidance of RAG Pipelines," arXiv:2505.13538, 2025
- [19] Y. Zhao, Z. Liu, Y. Zheng, K.-Y. Lam, "Attribution Techniques for Mitigating Hallucinated Information in RAG Systems: A Survey," ICAIIC 2026, arXiv:2601.19927
- [20] W. Zhang, J. Zhang, "Hallucination Mitigation for Retrieval-Augmented Large Language Models: A Review," Mathematics, vol. 13, no. 5, art. 856, 2025. doi:10.3390/math13050856