International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

Abstention-Aware and Governance-Constrained RAG for Grounded, Personalized, and Auditable Assistance

Adish Kumar S
Department of Computer Science and Engineering, PSG College of Technology, Coimbatore, Tamil Nadu, India

Prateekshaa T
Department of Computer Science and Engineering, PSG College of Technology, Coimbatore, Tamil Nadu, India

Hemanthkumar V (Corresponding Author)
Department of Computer Science and Engineering, PSG College of Technology, Coimbatore, Tamil Nadu, India

Praneeth M
Department of Computer Science and Engineering, PSG College of Technology, Coimbatore, Tamil Nadu, India

Jayavarshini S S
Department of Computer Science and Engineering, PSG College of Technology, Coimbatore, Tamil Nadu, India

Dr. Saranya K G
Department of Computer Science and Engineering, PSG College of Technology, Coimbatore, Tamil Nadu, India

Abstract - Retrieval-augmented generation (RAG) enhances grounding, but retrieval does not dictate when an assistant should refuse, how to separate user memory, or how to manage knowledge change. We propose a RAG architecture with abstention and governance constraints for grounded, personalised and auditable assistance. Our evidence-restricted policy allows domain responses only when evidence meets the explicit support conditions; otherwise, the assistant abstains or asks for clarification. Personalization is confined to evidence-restricted adaptation, and knowledge-base updates are managed as reviewed transitions, with audit trails. Assessment of existing academic-policy assistant artifacts shows 93.1% corpus average quality (ten policy documents), 151 responses out of 153 application attempts, and a 22-session, 150-turn scenario benchmark with groundedness, memory, refusal, contradiction resolution, and governance workflows. The largest validated signals are evidence-bounded compliance, refusal correctness, contradiction resolution, and knowledge-evolution stability. The findings establish architecture-level feasibility rather than benchmark-level superiority, as external (unrestricted corpus) and controlled (within corpus, with baselines) evaluation is left for future work.

Index Terms: retrieval-augmented generation, abstention-aware inference, grounded generation, personalization, memory isolation, governance, auditable AI systems.

I. INTRODUCTION

Large language models are increasingly used as chat assistants in contexts where not only is correctness a desired property but an operational one. Authority matters in institutional, legal, health, financial, compliance and enterprise knowledge domains: an answer is valuable only if it rests on reliable evidence, and inference is made only when it is justified. The danger is not that an assistant fails to generate grammatically coherent text; it is that it generates grammatically coherent text whose confidence is unwarranted.

Retrieval-augmented generation is one way to address this issue by gating generation on documents rather than on knowledge in the parameters of the model [1], [5], [6]. Dense retrieval, late interaction retrieval and hierarchical retrieval also improve the content and structure of the retrieved context [2]-[4], [12]. But better retrieval does not specify when the system should say “I don’t know”, how to separate user-specific memory from the rest of the knowledge base, or how to update the knowledge base in response to user feedback. These characteristics are often considered implementation considerations, but they directly impact whether a deployed assistant works or not.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

This paper explores RAG as a decision-making process with governance, rather than a retriever-and-generator. The innovation isn't a new retriever or generator, but the incorporation of answerability gating, user-specific personalization, role-specific update management and audit metadata into a unified architecture for deployed assistants. We consider three deployment principles. First, generation has to be abstention-aware: when evidence is lacking, the system should abstain or ask for clarification, rather than generate on the basis of unsupported evidence. Second, personalization must be constrained: user style, continuity and relevance preferences can be applied, but must not interfere with document evidence. Third, knowledge evolution must be controlled: changes in policy or corpus should be via reviewable update processes with audit trails, not via conversational drift.

We realize these principles in an evidence-restricted response policy. This policy constrains the assistant from answering using model prior knowledge if the domain answer requires support from the evidence retrieved from the corpus. It either generates an answer with references to evidence or abstains if the corpus evidence is insufficient. The same holds for memory and personalisation: user state can enhance the interaction but cannot generate new authority.

The research is tested in an academic-policy assistant. This domain is attractive because it includes real governance challenges: users pose policy questions, documents evolve over time, users play different roles (such as students, faculty, administrators), and wrong answers can misinform users about deadlines, eligibility, or review processes. The proposed method is domain-independent. Its abstractions are query, evidence chunks, support score, user memory, role-restricted actions, update tickets and audit logs. These abstractions can be applied to other document-based assistants.

This paper makes the following contributions:

1. **A decision-driven evidence-restricted RAG framework.** We cast response generation as a decision-making problem with constrained responses that can be made only when evidence meets conditions of support.

2. **A policy-constrained knowledge evolution model.** We decouple online inference (updating the model in real time) from offline mutation (corpus and policy updates) as reviewer-driven state transitions with contradiction resolution and audit trails.

3. **A multi-user personalization and memory-isolation model.** We specify personalization as evidence-bounded adaptation and mandate memory and preferences be confined to the logged-in user.

4. **Algorithmic processes for inference and update processing.** We provide auditor-friendly algorithms for support-gated query processing and reviewer-controlled policy update processing.

5. **A multi-dimensional evaluation protocol.** We structure evaluation around groundedness, refusal correctness, evidence-restricted compliance, personalization correctness, memory relevance, governance stability, and end-to-end, task-level success and provide results from the existing validation artifacts.

II. RELATED WORK

A. Retrieval-Augmented Generation

Retrieval-Augmented Generation (RAG) generates language conditioned on retrieved information and is a common technique for knowledge-driven NLP [1], [5], [6]. Dense Passage Retrieval enhances open-domain retrieval using learned dense representations [2], and ColBERT of late interaction retains the power of token-level matching [12]. Sparse retrieval systems like BM25 are important baselines for real-world retrieval [18]. In-context retrieval and hierarchical retrieval also enhance the way evidence is provided to the generator [3], [4]. These methods demonstrate the importance of evidence access but they don't completely specify the deployment policies for abstention, memory isolation and controlled corpus updates.

The important contribution of this paper is that retrieval is viewed as a prerequisite to answerability. An effective retriever might still return incomplete, contradictory, or slightly irrelevant evidence. Consequently, the assistant needs a support function and decision rule that chooses between answer and abstention. This transforms RAG from a conditioning process to a policy.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

B. Language Model Alignment and Reasoning Control

Large-scale pretraining and transformers afford the representational capacity of contemporary LLM assistants [7]-[11]. Instruction tuning and reinforcement learning from human feedback enhance alignment with user intent and conversational standards [13], [30], and chain-of-thought approaches and self-consistency enhance reasoning capabilities [31], [32]. Recent foundation models and benchmarking studies also show general capability gains [14], [15], [28], [29], [33].

But higher capabilities do not erase unsupported generation. A model can follow the instructions while producing claims not supported by the retrieved context. The use of language models with tools and modular reasoning is motivated by previous research on augmented language models [16], [17], but governed deployments need to explicitly refuse to perform actions and have identifiable control points. The current work therefore leverages LLM capabilities as part of a controlled architecture, but not exclusively.

C. Security, Identity and Governance

Authenticated systems often use OAuth2, bearer tokens and JSON Web Token (JWT) for identity management [19]-[22]. Role-based access control continues to play an important role in segregation of privilege [23], [24]. Guidance on digital identity and risk management for AI highlight lifecycle, traceability, accountability, and risk [25], [26]. Risks in the design of web applications also emphasize authorization, validation and auditability [27].

This informs the governance component of the proposed architecture. Existing RAG research focuses on retrieving evidence and generating answers, while governance standards focus on who can modify state, how they can mutate state, and how they record their decisions. The proposed architecture unites these issues by separating queries from privileged access to the knowledge base.

D. Assistants with Personalization and Memory

Personalization can enhance satisfaction by modulating tone and response format and by maintaining context. But persistent memory can be problematic when it is not user-scoped or when memory of facts is elevated to the status of evidence. Memory in a governed assistant should be used to make interactions more relevant, not to replace domain evidence. Hence, this paper considers user memory as scoped context, not authority. The response may be tailored in its form but based on retrieved documents.

E. Gap in Existing Work

Prior research offers excellent tools for retrieving and generating, aligning, securing and governing, but often in isolation. What's missing is an architecture for a RAG assistant that simultaneously ensures answerability, user-level memory, role-based governance and verifiable knowledge evolution. In this paper, we address this gap with a formal model of the assistant as a constrained decision-making system with online and offline control loops.

III. Problem Formulation

We use a user query to be represented as q, the previous history of the conversation as h, the user who is authenticated as u, optional user-specific memory as m_u, and the set of evidence as

$$C_k = \{c_1, c_2, \dots, c_k\}.$$

The response from the assistant is either an answer y with citations from the set of sources $S \subseteq C_k$, or a refusal to answer r. The choice is constrained by four rules.

1. **Groundedness constraint:** statements in y must be backed by evidence in C_k.
2. **Refusal constraint:** if evidence support is not sufficient, the system must refuse to answer by outputting r instead of inferring from unsupported prior knowledge.
3. **Isolation constraint:** preferences and memory of user u must remain private.
4. **Governance constraint:** only reviewed processes can change the authoritative corpus.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

We set the system goal as utility maximization with constraints:

$$\max U = \alpha G + \beta P + \gamma T - \delta H - \epsilon L,$$

Here, G is groundedness fidelity, P is personalization correctness, T is task success, H is unsupported-content rate, and L is governance violation risk. The weights $\alpha, \beta, \gamma, \delta, \epsilon$ capture preferences for utility and risk in deployment contexts.

To decide whether a question is answerable, we define a support function:

$$\sigma(q, C_k) = \lambda_1 \cdot \max_i \text{sim}(q, c_i) + \lambda_2 \cdot \text{coverage}(q, C_k) + \lambda_3 \cdot \text{consistency}(C_k).$$

Here, $\text{sim}(q, c_i)$ is a measure of retrieval relevance, $\text{coverage}(q, C_k)$ is a measure of whether the key information needs in q are covered by the evidence spans in the top-k chunks, and $\text{consistency}(C_k)$ is a measure of inconsistency amongst the top-k chunks. The answer policy is:

$$\pi(q) = \begin{cases} \text{answer with citations}, & \sigma(q, C_k) \geq \tau, \\ \text{abstain/refuse}, & \sigma(q, C_k) < \tau. \end{cases}$$

The risk threshold τ reflects the deployment's risk tolerance. In dangerous situations, τ should be low, to reject updates with weak evidence.

We capture knowledge evolution as a risk term:

$$L = \eta_1 \cdot \Pr(\text{unauthorized update}) + \eta_2 \cdot \Pr(\text{unreviewed contradiction}) + \eta_3 \cdot \Pr(\text{audit incompleteness}).$$

This term connects governance to system risk. The assistant is not only judged on whether it gives the right answer at the time, but on whether the state which determines the answer for future questions is changed through transparent and auditable processes.

IV. METHODOLOGY

A. Online Inference

The online inference path processes user queries. It retrieves recent history from the session, preferences and memory from the user, rewrites the query (if conversational history demands it), retrieves evidence, scores support, and generates a grounded answer if the support is sufficient, otherwise it refuses.

Algorithm 1: Abstention-Aware Evidence-Restricted Query Processing Input: query q, session s, user u Output: answer y or refusal r, sources S, metadata M 1: H <- load_recent_history(s) 2: P <- load_user_preferences(u) 3: U <- load_user_memory_context(u) 4: q' <- optional_history_aware_rewrite(q, H) 5: C_k <- retrieve_top_k(q', index) 6: sigma <- support_score(q', C_k) 7: if sigma < tau then 8: M <- {confidence=sigma, refusal=true} 9: persist_interaction(s, u, q, r, M) 10: return refusal_response(q, C_k), empty_or_partial_sources(C_k), M 11: end if 12: y <- grounded_generate(q', C_k, P, U) 13: S <- source_map(y, C_k) 14: M <- {confidence=sigma, refusal=false, timing, risk_alerts} 15: if policy_claim_detected(q, y, C_k) then 16: M <- M + pending_policy_update_signal 17: end if 18: persist_interaction(s, u, q, y, M) 19: return y, S, M

This algorithm embodies the key design principle: grounded generation. User memory and preferences are only accessible to the generator when support is present. They may influence tone, structure or continuity, but not claims that are not in C_k.

B. Offline Knowledge Evolution Update of the knowledge base is postponed with respect to answering live queries. When a user informs the system that a policy has changed they should not update the corpus immediately. Rather, it generates a proposal for review, attaches evidence, identifies affected chunks, and awaits an approval from a reviewer.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

Algorithm 2: Reviewer-driven Policy Update Execution Input: claim x, evidence e, reviewer decision d Output: updated corpus state or rejected update record 1: t <- create_ticket(x, e, trust_score(x, e)) 2: a <- detect_contradictions(x, active_chunks) 3: z <- identify_affected_chunks(a) 4: if d == reject then 5: write_audit(t, status="rejected") 6: stop 7: end if 8: deprecate_chunks(z) 9: create_updated_chunks(new_content) 10: write_audit(t, status="implemented") 11: verify_post_update_retrieval()

The critical design choice is to distinguish between uncertainty in response time and in corpus time. Abstention is the handling of uncertainty in interaction; reviewer-governed update execution is the handling of uncertainty about whether we need to change the underlying knowledge base.

C. Personalization with Constraints

Personalization is represented as a bounded function:

$$y = f(q, C_k, p_u, m_u),$$

where p_u is the user's preferences and m_u is the user's memory. The function is defined only if the factual information in y is entailed by C_k. This distinguishes adaptation from factual authorization. The assistant can tell the user about an answer in a shorter way if the user prefers short answers, or may use a user's previously specified program if it is relevant, but it cannot infer a policy rule from memory.

Memory isolation is defined as:

$$\forall u_i \neq u_j, \quad m_{u_i} \cap_{\text{accessible}} m_{u_j} = \emptyset.$$

In practice, this means that memory recall, preference application and session persistence should be keyed by the authenticating user ID and role-constrained.

D. Auditability

Controlled decisions generate metadata. Metadata for queries includes confidence, refusal, data source, time and risk. For corpus mutation, metadata includes ticket number, evidence submitted, contradiction analysis, reviewer decision, affected chunks and audit status. It enables reproducibility and future auditability of answers and corpus mutations.

V. System Overview

The design has four planes, depicted in Fig. 1 and Fig. 2 below.

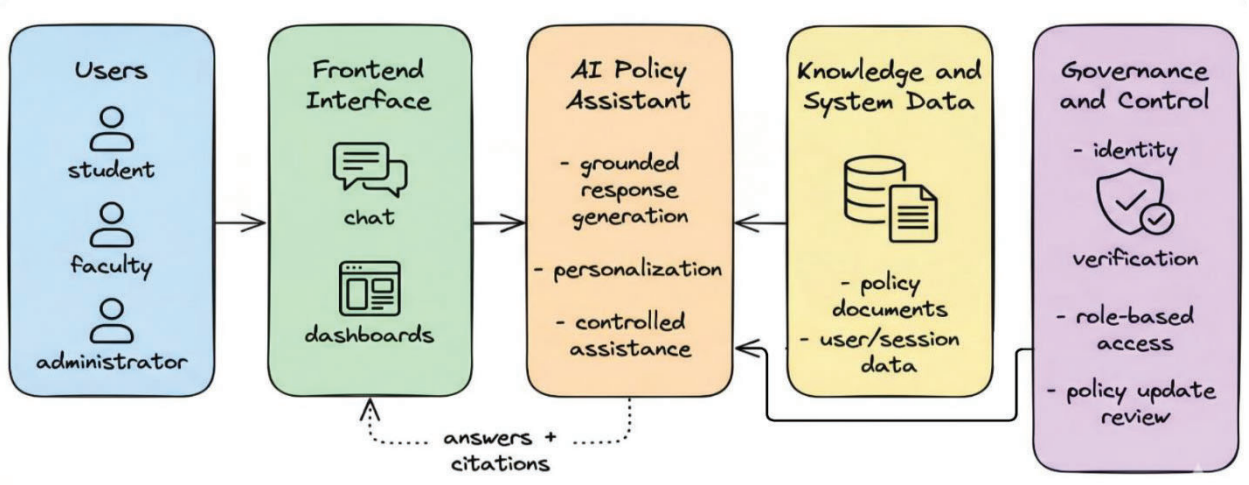


Fig. 1 - Methodological flow

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

Fig. 1. Methodology flow of the proposed abstention-aware evidence-restricted academic policy assistance system, showing the authentication of the query, loading the session, retrieving evidence from ChromaDB, generating with support, and escalating with policy claims.

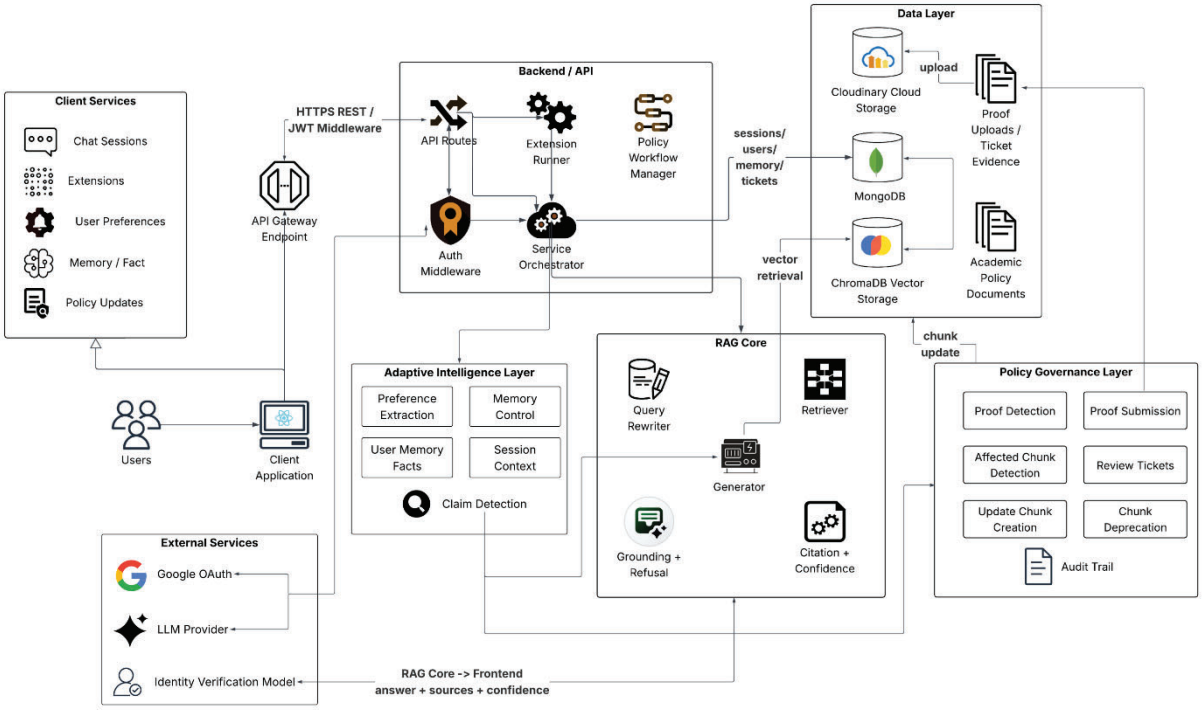


Fig. 2 - High-level system architecture

Fig. 2. Architecture overview of client services, orchestration (API routes, extension runner, policy workflow manager, auth middleware, service orchestrator), RAG components, data (Cloudinary, MongoDB, ChromaDB), and policy governance.

The interaction plane includes chat, session persistence, preferences, memory settings, role-based governance actions. The orchestration plane orchestrates retrieval, support, generation, refusal, policy-claim match and update. The evidence and state plane is responsible for storing document fragments, metadata, sessions, preferences, memory facts, review tickets and audit events. The security and identity plane enables authentication and role-based authorisation.

The backend is based on a service architecture, as shown in Fig. 3. Route groups are mapped to distinct services: authenticate, chat (including retrieval and memory sub-services), extension, claim detection, policy update and identity verification. Services link to the corresponding repository or data interface, ensuring separation of concerns.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

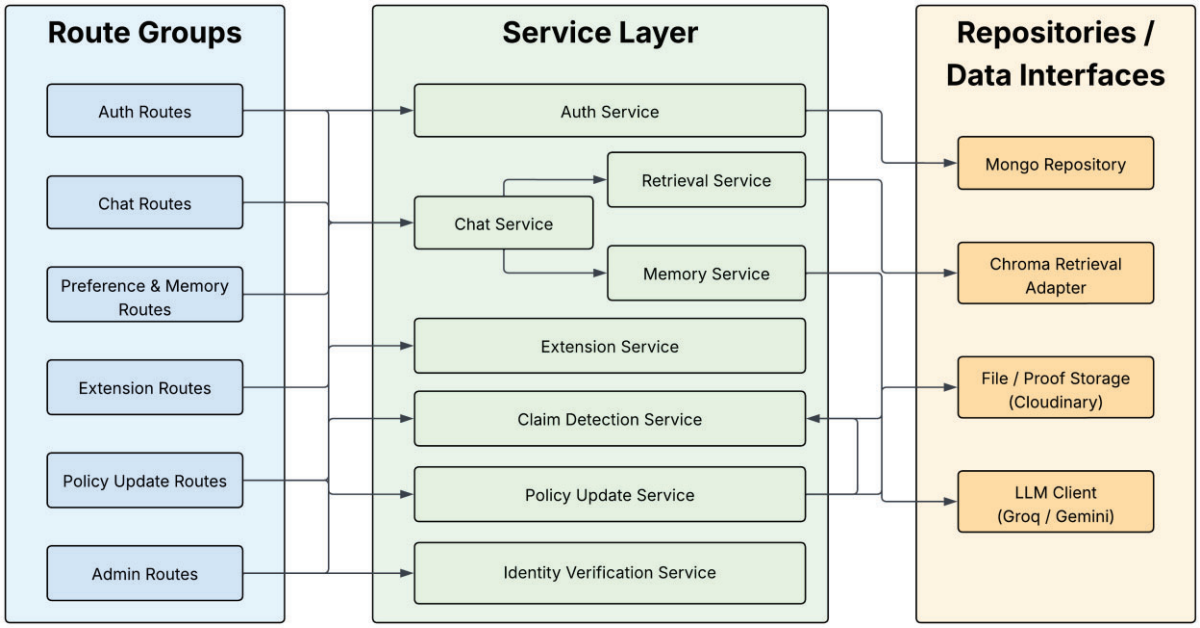
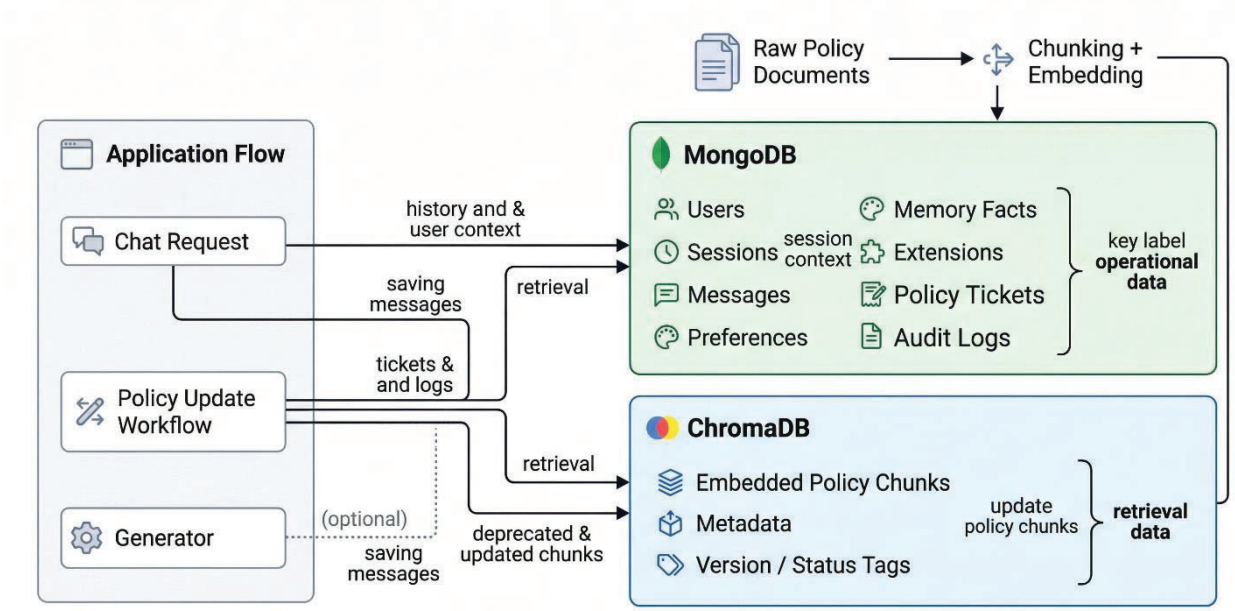


Fig 3 Backend service architecture

Fig. 3. Backend services architecture: groups of routes (Auth, Chat, Preference & Memory, Extension, Policy Update, Admin) connect to a service layer, which then connects to repository and data interface bindings (Mongo Repository, Chroma Retrieval Adapter, File/Proof Storage via Cloudinary, LLM Client via Groq/Gemini).

The data layer has operational and retrieval data. Users, sessions, messages, preferences, memory facts, policy tickets, audit logs, and session context are stored in MongoDB. ChromaDB stores policy chunks, metadata and version or status. The policy documents are chunked and embedded before ingested into ChromaDB. Fig. 4 illustrates the interaction between the two data stores in the chat and policy-update processes.



International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

Fig. 4 - MongoDB and ChromaDB data interaction

Fig. 4. Interaction with operational data in MongoDB (users, sessions, messages, preferences, memory facts, extensions, policy tickets, audit logs) and retrieval data in ChromaDB (embedded policy chunks, metadata, version and status tags) in the retrieval and write paths of chat requests, policy update workflows, and the optional generator flow.

This diagram omits implementation details. The paper's contribution is not the specific web stack or database technology, but rather the control structure that enables a combination of evidence-based inference, limited personalization, isolation of sessions, and reviewer-based evolution of knowledge.

VI. Experimental Protocol

A. Evaluation Setting

The assistant is evaluated on an academic-policy corpus. The corpus includes ten policy artifacts on academic advising, continuous assessment, course add, drop and withdrawal, exam registration, final-year projects, grading, internships, laboratory evaluation, project submission and revaluation. The environment is controlled but realistic to allow for the evaluation of grounded question answering, follow-up queries, personalisation, memory recall, contradiction resolution and policy update.

The results are based on three existing artifacts: a dataset validation report, an application core validation report and a scenario-flow validation report [34]-[36]. The artifacts are considered as a complementary evaluation report. They are not a public benchmark evaluation.

B. Metrics

Let N be the number of test cases. The protocol uses these metrics:

$$\text{Grounded Fidelity} = \frac{N_{\text{grounded}}}{N},$$

$$\text{Personalization Accuracy} = \frac{N_{\text{personalization correct}}}{N_{\text{personalization expected}}},$$

$$\text{Memory Isolation Integrity} = \frac{N_{\text{leak free}}}{N_{\text{cross user}}},$$

$$\text{Task Success Rate} = \frac{N_{\text{scenario success}}}{N_{\text{scenarios}}},$$

$$\text{Abstention Precision} = \frac{N_{\text{correct abstain}}}{N_{\text{all abstain}}},$$

$$\text{Policy Update Reliability} = \frac{N_{\text{consistent audited updates}}}{N_{\text{approved updates}}}.$$

Grounded fidelity measures groundedness of responses in terms of fact-checking using evidence and sources. Personalization accuracy is measured only on turns where personalization is expected; neutral turns are not considered. Memory isolation integrity is tested on cross-user scenarios; it is a measure of memory isolation. Task success is more stringent than just response success: a turn or scenario may get a response but still fail if it is not complete, not well grounded, or does not complete the user's requested task. Refusal precision measures if abstention happens when evidence is not sufficient, while policy-update success measures if approved updates are audited and retrieval-consistent.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

The unified validation report also measures evidence-restricted compliance, user memory relevance, contradiction resolution, knowledge-evolution stability, hallucination rate, context sufficiency, retry recovery, refusal correctness, and policy-claim false alarm rate. The original validation artifact refers to evidence-restricted compliance as "zero-knowledge compliance"; this paper uses evidence-restricted to avoid conflation with a cryptographic concept. The user memory relevance is not a normalized accuracy. It is an average relevance value across memory-eligible turns, and can be greater than 1 if multiple relevant memory items are retrieved or used in a turn.

C. Validation Design

The data set validation checks suitability of the corpus on eight criteria: structure, content, cross-referencing, terminology, logical consistency, realism, RAG compatibility and quality [34].

The application validation checks the authenticated end-to-end assistant. It involves deterministic testing for session consistency, source inclusion, latency, refusal, preference and memory, policy-claim, and retry. It also compares the results of an independent LLM judge for groundedness, hallucination, relevance, completeness, context-sufficiency, personalization, memory relevance, contradiction, and acceptance [35].

The scenario-flow validation tests the benchmark design and execution. It has 22 sessions with 150 authored turns structured as multi-turn dialogues rather than single prompts. These sessions range from onboarding and preferences, grounded policy questions, memory tests, contradictions, policy-oriented proof or "theory" workflows, and reasoning-saturated scenarios [36].

VII. RESULTS

A. Corpus Quality

The dataset validation report assessed ten policy documents on eight criteria, and gave an average overall score of 93.1% [34]. The results for the individual documents are presented in Table I.

Policy Document	Overall Score
Academic advising and mentorship	93%
Continuous assessment overview	96%
Add/drop/withdrawal of courses	93%
Exam registration process	95%
Final-year project guidelines	91%
Marking components and weightage	97%
Internship registration and evaluation	90%
Lab marks and internal assessment	93%
Project submission and review process	90%
Revaluation and answer script review	93%
Average	**93.1%**

Table I. Dataset validation of the academic-policy corpus [34].

The best performing dimensions were consistency, RAG suitability and quality. Cross-reference accuracy was lower, but still good. This finding confirms the suitability of the corpus as a controlled environment for testing grounded RAG, but also suggests differences from an actual institutional document store.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

B. Application Validation

The integrated application validation shows 153 attempts, 151 successful responses and 2 interrupted attempts due to provider usage limits, not application logic issues [35]. This results in a 98.69% effective success ratio. The main metrics are reported in Table II.

Metric	Value
Successful responses	151/153
Effective success rate	98.69%
Personalization accuracy	0.7391
Evidence-restricted compliance	1.0000
Relevance to user memory	1.2693
Grounded response fidelity	0.6784
Contradiction resolution rate	1.0000
Knowledge-evolution stability	1.0000
End-to-end task success rate	0.6536
Hallucination rate	0.0463
Context sufficiency rate	0.6755
Retry recovery rate	0.3333
Refusal correctness rate	1.0000
Policy-claim false alarm rate	0.0000

Table II. Integrated application validation metrics [35].

The top metrics are in the safety and governance areas: evidence-restricted compliance, contradiction resolution, knowledge-evolution stability, refusal correctness, and policy-claim false alarm rate. These results correspond to the expected control points. The lower grounded response fidelity and task success scores suggest that evidence-gated control does not solve answer completeness or synthesis problems.

The grounded response fidelity score is moderate because the metric rewards not just non-hallucinated responses, but also full coverage of evidence and proper citation. Some responses were safe and relevant but did not answer all sub-questions, cite all relevant source spans, or synthesise evidence from several chunks. This is a critical distinction: evidence restrictions can stop unsupported assertions while leaving the door open for improvements in retrieval ranking, chunking and answer synthesis.

End-to-end task success is also lower than execution success due to more realistic multi-turn scenarios. A response may be successful at the application level but receive a lower task success if it needs to be clarified, does not complete a role-specific task workflow, makes a conservative refusal when a better retrieval might allow an answer, or is interrupted by provider rate limits. So the task success number reflects the use case rather than the availability of the backends themselves.

The security model used to achieve these results is shown in Fig. 5. Google OAuth is integrated into JWT middleware and an identity gate. Role-based access control ensures user, faculty, and admin access control for feature-level access to chat, preferences, memory, extensions, policy review, and admin pages. A data isolation and governance layer isolates user memory, protected policy review routes, audit logs and chunk updates.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

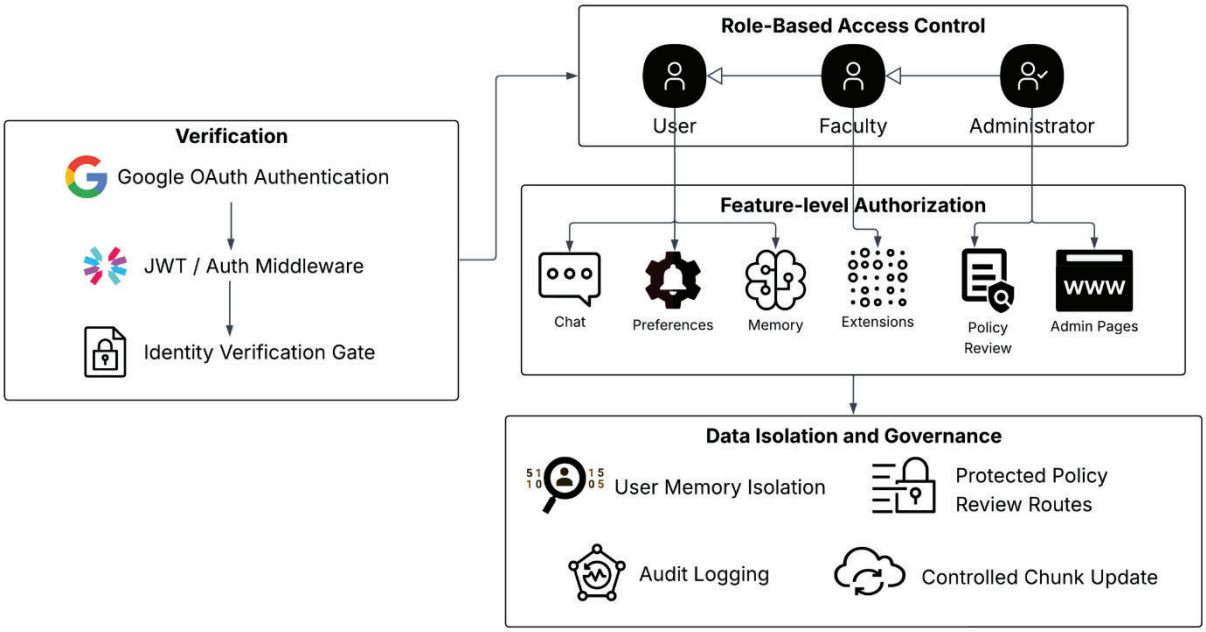


Fig. 5 - Role-based access control and layered security model

Fig. 5. Role-based access control and layered security model: Google OAuth and JWT middleware-based verification, role-based feature-level authorization (User, Faculty, Administrator), data isolation and governance through memory isolation, protected routes, audit logging and controlled chunk updates.

C. Scenario-Flow Coverage

The scenario-flow test suite has 22 sessions and 150 author's turns [36]. It includes first-session behavior, user introduction, preference learning, user-profiles recall, basic academic-policy questions, follow-up clarification, contradiction, policy-proof workflows and complex reasoning. The coverage is summarised in Table III.

Coverage Area	Included
Groundedness	Yes
Hallucination resistance	Yes
Completeness	Yes
Context carryover	Yes
Preference learning	Yes
Memory use	Yes
Evidence-restricted behavior	Yes
Contradiction handling	Yes
Policy-claim workflow	Yes
Refusal correctness	Yes
Dense reasoning	Yes

Table III. Behavioral coverage of the multi-turn scenario benchmark [36].

The scenario is crucial because the proposed architecture is not a single-turn question answering (QA) system. The important behaviors occur across sessions: the assistant should adapt to user preferences, without inferring hidden

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

facts; use memory, without leaking information between users; preserve context, without overriding evidence; and escalate user policy uncertainty, without automatically rewriting the corpus.

D. Baseline Comparison

Table IV compares the proposed system to a standard RAG assistant [1]. The comparison is qualitative and is thus not controlled because the current validation artifacts do not include a baseline implementation of standard RAG over the same scenarios.

Aspect	Standard RAG	Proposed Evidence-Restricted Governed RAG
Answer policy	Answers from retrieved context if available	Answers only if support threshold is met, otherwise abstains or refuses to answer
Poor evidence	May answer with partial evidence or prior bias	Refuses, declines, or asks for clarification
Personalization	May use memory as well as evidence retrieval	Uses memory only for limited adaptation after evidence support is confirmed
Memory isolation	Implementation-specific	Authenticated, user-specific memory retrieval required
Knowledge updates	Typically uses direct re-indexing or other unstructured editing	Involves tickets mediated by a reviewer, with contradiction and audit trails
Contradiction detection	Usually left to retrieval or generation actions	Checks for contradictions and notifies policy-change claims to governance processes
Auditability	Typically limited to logs and references	Tracks confidence, refusal, source, ticket, reviewer decisions and impacted chunks
Main threat mitigated	Retrieval helps access facts	Unjustified answering, memory leakage, uncontrolled mutation of the corpus

Table IV. Qualitative assessment of the difference between a RAG assistant and the evidence-restricted governed RAG of this work.

VIII. ANALYSIS

Three key observations can be made.

First, forcing abstention and evidence-restricted compliance enhance discipline. The observed evidence-restricted compliance of 1.0 and refusal correctness of 1.0 in the validated cases suggest that the assistant does not use unsupported personalization or unsupported prior knowledge where it should not. This is the main anticipated benefit of the answerability threshold.

Second, governance separation improves stability. In the cases tested, contradiction resolution stability, knowledge-evolution stability and policy-claim false alarm rate were all 1.0 and 0.0 respectively. This suggests that by considering update requests as objects in a review workflow it is possible to avoid corpus mutation gone wild, while still raising policy-change claims. The overall process is illustrated in Fig. 6: a user-provided policy contradiction results in the detection of a claim and the submission of proof, which establishes a review ticket that is then reviewed by faculty or administrators. Successful updates trigger affected chunk location, outdated chunk deprecation, updated chunk creation and audit trail logging, before the knowledge base is modified. Denied tickets are archived without changes.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

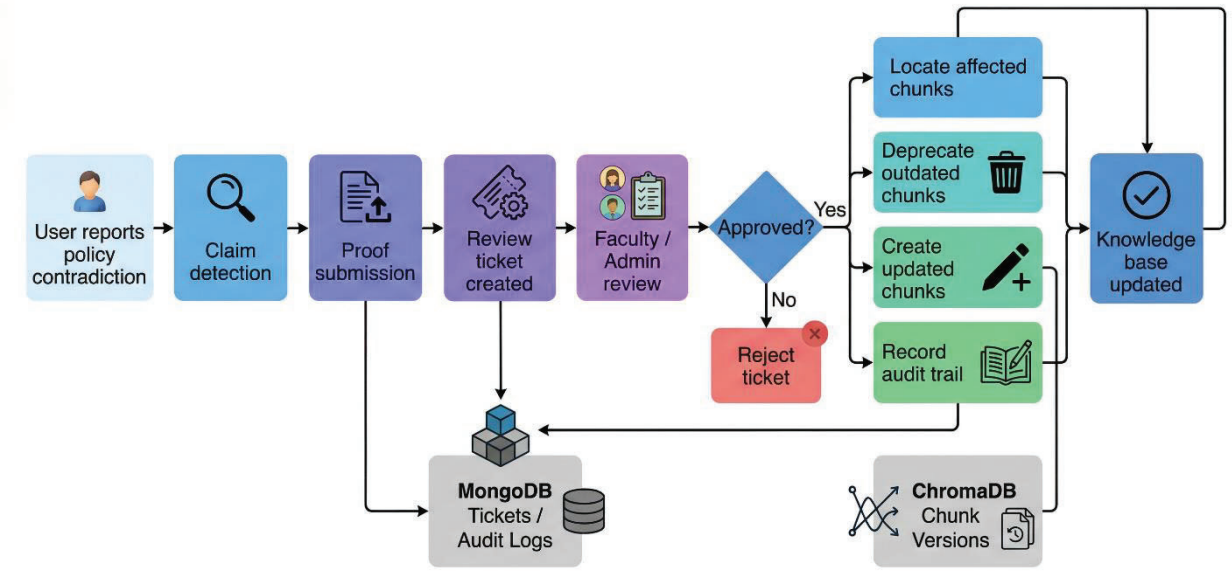


Fig. 6 - Policy unlearning and controlled knowledge evolution workflow

Fig. 6. Unlearning and controlled evolution of policy-based knowledge: user perceived contradictions prompts claim identification, proof collection, ticket generation; the corpus mutation is gated by the review of faculty/administrators; if approved, the update creates new chunks, invalidates old chunks and logs the audit trail in ChromaDB; if not, it logs the rejection without modifying the knowledge base in MongoDB.

Third, safety measures do not fully address quality issues. Grounded response fidelity (0.6784) and context sufficiency (0.6755) indicate the need for improved evidence synthesis or retrieval. Overall task success was 0.6536, which is reasonable for a multi-turn assistant but may be improved. The architecture mitigates some risks but does not guarantee optimum retrieval, complete answers or best performance compared with other RAG designs.

These results are in line with the paper's main thesis. The proposed architecture is most beneficial where groundedness, abstention, personalisation limits and governance are needed. It is not claimed to be a panacea for all RAG systems but a governance framework for controlled deploys of assistants.

A. Failure Cases and Error Analysis

Our validation evidence suggests that most errors are either quality or coverage errors rather than uncontrolled-generation errors. First, groundedness failures occur when the retrieved topic chunk is relevant but not comprehensive, leading to the answer being missing a qualification, exception or link. Second, task failures result in multi-step sessions when the assistant answers locally correctly, but does not complete a larger user task, such as bringing a policy workflow to the next step in the process. Third, failure recovery is limited due to service interruptions or rate limits that can invalidate the continuation of scenarios even when the application logic is correct.

The hallucination rate (0.0463) indicates unsupported generation was rare but not zero. In the proposed solution, such cases should be interpreted as support calibration and citation-attribution failures: either the support score allowed through an answer with inadequate coverage, or the generated answer added information not entailed by the cited chunks. The solution is to reduce support thresholds for high-risk intents, to enhance citation-span verification and to include regression tests for multi-hop policy questions and exception-laden documents.

IX. REPRODUCIBILITY STATEMENT

The evaluation is done on three project artefacts: the dataset validation report, the application core validation report and the scenario-flow validation report [34]-[36]. These artifacts define the corpus under evaluation, validation aspects, metrics, and multi-turn scenario-flow. For exact reproduction of application-level numbers, the same corpus version, authentication and role settings, retrieval index, model providers, prompts and scenario scripts are needed.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

Given the use of LLM-assisted evaluation and external model providers for some judgments, exact reproduction may differ between model providers and rate-limited scenarios. The next release should contain fixed benchmark splits and prompt/evaluator rubrics, metadata on the retrieval index, and scripts for reproduction.

X. LIMITATIONS

The study has several limitations.

First, the baseline in this paper is an architectural comparison, not an empirical one with a baseline. The results don't include any implemented baselines such as traditional RAG, R-only QA and assistants, or monolithic orchestration of LLM agents on the same set of use cases. So, the results should not be viewed as comparative.

Second, the corpus is simulated. While it was validated as structurally sound and fit for RAG assessment, it should not be used as a proxy for deployment over institutional documents with dynamic policies, poor formatting, and multiple authors. The architecture should be evaluated on other datasets and non-institutional corpora to verify its suitability outside the academic-policy domain.

Third, some evaluation uses LLM-assisted judgments. This is helpful for automated semantic review, but adds judge variability and should be supplemented with human annotation in future research.

Fourth, the benchmark is project-specific. The 22 sessions, 150 turns of scenarios cover common use cases, but is not yet a public, independently splittable benchmark. Benchmarking against external data is required to confirm the same abstention, memory, and governance scenarios apply to a range of document genres, retrieval distributions and user goals.

Fifth, the safety and governance scores are based on the tested cases. Stress testing is needed for jailing, session and user leakage, policy update conflicts, and long-term memory retention.

XI. CONCLUSION

This article proposed an abstention-capable, policy-constrained RAG system for grounded, personal and auditable support. The approach models answer generation as a decision process, constrained by an evidence support function that must be met before generation commences. It also constrains personalisation via user-specific memory and decouples online inference from reviewer-driven knowledge growth.

Architecture-level claims are supported by evaluations over existing project artifacts. Corpus validation provides an average quality of 93.1% for ten policy documents, application validation provides 151 successful responses out of 153 attempts (98.69% effective success rate), and the scenario-flow benchmark provides 22 sessions, 150 turns encompassing groundedness, memory, personalization, refusal, contradiction resolution, and policy workflows. The best evidence is in evidence-restricted compliance, refusal correctness, contradiction resolution, and knowledge-evolution stability.

Next steps should include controlled baselines, benchmarking on institutional and non-institutional corpora, benchmark splits, abstention thresholds calibrated to human reference, and formal proofs of role-policy invariants in workflows. More generally, the paper suggests that achieving trustworthy RAG assistants requires not just improvements in retrieval and generation, but also control over whether to answer, what to remember, and how knowledge evolves.

Funding Declaration

The authors received no financial support for the research, authorship, and/or publication of this article.

Author Contributions

Adish Kumar S and Hemanthkumar V conceptualized the study and designed the proposed abstention-aware and governance-constrained RAG framework. Adish Kumar S, Hemanthkumar V, Jayavarshini S S, Prateekshaa T, and Praneeth M developed the system, implemented the methodology, conducted experiments, and collected validation results. Hemanthkumar V prepared the manuscript draft and figures. Adish Kumar S, Jayavarshini S S, Prateekshaa T, and Praneeth M contributed to data preparation, testing, evaluation, and manuscript revision. Dr. Saranya K G

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

supervised the research, provided technical guidance, reviewed the methodology, and provided feedback on the manuscript. The authors reviewed, edited, and approved the final manuscript.

ACKNOWLEDGMENT

The authors would like to thank Dr. Saranya K G, Department of Computer Science and Engineering, PSG College of Technology, Coimbatore, for her guidance, suggestions, and encouragement throughout this work.

REFERENCES

[1] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Kuttler, M. Lewis, W.-t. Yih, T. Rocktaschel, S. Riedel, and D. Kiela, "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," in Proc. NeurIPS, 2020.

[2] V. Karpukhin, B. Oguz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W.-t. Yih, "Dense Passage Retrieval for Open-Domain Question Answering," in Proc. EMNLP, 2020.

[3] O. Ram, Y. Levine, I. Dalmedigos, D. Muhlgay, A. Shashua, K. Leyton-Brown, and Y. Shoham, "In-Context Retrieval-Augmented Language Models," Trans. ACL, 2023.

[4] P. Sarthi, S. Abdullah, A. Tuli, S. Khanna, A. Goldie, and C. D. Manning, "RAPTOR: Recursive Abstractive Processing for Tree-Organized Retrieval," arXiv:2401.18059, 2024.

[5] K. Guu, K. Lee, Z. Tung, P. Pasupat, and M. Chang, "REALM: Retrieval-Augmented Language Model Pre-Training," in Proc. ICML, 2020.

[6] G. Izacard and E. Grave, "Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering," in Proc. EACL, 2021.

[7] A. Vaswani et al., "Attention Is All You Need," in Proc. NeurIPS, 2017.

[8] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in Proc. NAACL-HLT, 2019.

[9] Y. Liu et al., "RoBERTa: A Robustly Optimized BERT Pretraining Approach," arXiv:1907.11692, 2019.

[10] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," in Proc. EMNLP-IJCNLP, 2019.

[11] T. B. Brown et al., "Language Models are Few-Shot Learners," in Proc. NeurIPS, 2020.

[12] O. Khattab and M. Zaharia, "ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT," in Proc. SIGIR, 2020.

[13] L. Ouyang et al., "Training Language Models to Follow Instructions with Human Feedback," in Proc. NeurIPS, 2022.

[14] H. Touvron et al., "LLaMA: Open and Efficient Foundation Language Models," arXiv:2302.13971, 2023.

[15] OpenAI, "GPT-4 Technical Report," arXiv:2303.08774, 2023.

[16] S. Mialon et al., "Augmented Language Models: A Survey," arXiv:2302.07842, 2023.

[17] E. Karpas et al., "MRKL Systems: A Modular, Neuro-Symbolic Architecture that Combines Large Language Models, External Knowledge Sources and Discrete Reasoning," arXiv:2205.00445, 2022.

[18] S. Robertson and H. Zaragoza, "The Probabilistic Relevance Framework: BM25 and Beyond," Found. Trends Inf. Retr., vol. 3, no. 4, pp. 333-389, 2009.

[19] D. Hardt, "The OAuth 2.0 Authorization Framework," RFC 6749, IETF, 2012.

[20] M. Jones, J. Bradley, and N. Sakimura, "JSON Web Token (JWT)," RFC 7519, IETF, 2015.

[21] M. Jones and D. Hardt, "The OAuth 2.0 Authorization Framework: Bearer Token Usage," RFC 6750, IETF, 2012.

[22] T. Lodderstedt, M. McGloin, and P. Hunt, "OAuth 2.0 Threat Model and Security Considerations," RFC 6819, IETF, 2013.

[23] D. F. Ferraiolo, R. Sandhu, S. Gavrila, D. R. Kuhn, and R. Chandramouli, "Proposed NIST Standard for Role-Based Access Control," ACM Trans. Inf. Syst. Secur., vol. 4, no. 3, pp. 224-274, 2001.

[24] R. Sandhu, E. Coyne, H. Feinstein, and C. Youman, "Role-Based Access Control Models," IEEE Computer, vol. 29, no. 2, pp. 38-47, 1996.

[25] NIST, "Digital Identity Guidelines: Authentication and Lifecycle Management," SP 800-63B, 2020.

[26] NIST, "AI Risk Management Framework (AI RMF 1.0)," NIST AI 100-1, 2023.

[27] OWASP Foundation, "OWASP Top 10: The Ten Most Critical Web Application Security Risks," 2021.

[28] Google, "Gemini: A Family of Highly Capable Multimodal Models," arXiv:2312.11805, 2023.

[29] A. Chowdhery et al., "PaLM: Scaling Language Modeling with Pathways," arXiv:2204.02311, 2022.

[30] H. W. Chung et al., "Scaling Instruction-Finetuned Language Models," arXiv:2210.11416, 2022.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

[31] J. Wei et al., "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models," in Proc. NeurIPS, 2022.

[32] X. Wang et al., "Self-Consistency Improves Chain of Thought Reasoning in Language Models," arXiv:2203.11171, 2022.

[33] A. Srivastava et al., "Beyond the Imitation Game: Quantifying and Extrapolating the Capabilities of Language Models," Trans. Mach. Learn. Res., 2023.

[34] "Dataset Validation Report, 22 January 2026" [Online]. Available: https://github.com/Hemanth-kumar-05/zero-knowledge-adaptive-agent/releases/download/reports/dataset_validation_report.pdf

[35] "Application Core Validation Report, 1 April 2026" [Online]. Available: https://github.com/Hemanth-kumar-05/zero-knowledge-adaptive-agent/releases/download/reports/application_core_validation_report.pdf

[36] "Scenario Flow Validation Report: GPT-Authored Multi-Turn Benchmark, 1 April 2026" [Online]. Available: https://github.com/Hemanth-kumar-05/zero-knowledge-adaptive-agent/releases/download/reports/application_scenario_flow_validation_report.pdf