An Industry-Agnostic Agent-Assist Decision System: RAG + Calibrated Routing with Reject-Option Handoff

Hemanth Sai Danaboina

Abstract

I propose an industry-agnostic agent-assist decision system that unifies retrieval-augmented generation (RAG) with supervised routing and calibrated abstention. The system consumes a fixed two-file data contract: (i) extttdocs.jsonl, a chunked knowledge corpus with mandatory multi-tenant and section metadata, and (ii) exttttickets.parquet, historical tickets labeled with extttresolution_path and extttescalated. Offline, I normalize documents, chunk with overlap, deduplicate near-duplicates, embed, and build sparse (PostgreSQL full-text) and dense (pgvector HNSW) indices. Online, I perform hybrid retrieval, fuse ranks with reciprocal rank fusion, rerank into an evidence pack, and pass this evidence to both (a) a LightGBM routing model that predicts extttresolution_path and escalation risk and (b) a generator constrained to cite chunk identifiers. I calibrate routing probabilities with temperature scaling and apply a reject-option policy based on a single confidence score that combines calibrated routing probability with evidence quality, triggering structured human handoff under uncertainty. I evaluate retrieval quality, routing accuracy, calibration, risk–coverage, citation faithfulness, and end-to-end latency.

1 Introduction

Support agents spend substantial time searching internal documentation, interpreting prior cases, and deciding whether a case is safe to resolve or should be escalated. RAG improves factual grounding by conditioning generation on retrieved evidence rather than relying solely on parametric recall [1]. However, in enterprise support settings, a system must also *route* an issue into a resolution path and *avoid harmful false positives*. I address this by coupling RAG with supervised routing and an explicit abstention policy.

I target the practical constraint that organizations differ widely in tooling, taxonomy, and documentation quality. My approach is therefore *industry-agnostic by construction*: the pipeline and code are fixed, and all domain differences are represented by the data contract and metadata (tenant and section). Sections represent sub-units inside one organization (e.g., departments); the same tenant can have many sections simultaneously.

System overview. Figure 1 shows the full pipeline. Offline, I ingest `docs.jsonl` and `tickets.parquet`, normalize and chunk documents, deduplicate near-duplicates, embed and index chunks, and train routing models. Online, a new issue triggers hybrid retrieval (sparse + dense), reciprocal rank fusion (RRF) [2], reranking, routing with calibrated confidence [3], and a reject-option decision rule [4]. If confidence is sufficient, the system returns a recommendation and a grounded draft response; otherwise it abstains and emits a structured handoff payload.

Preprint.

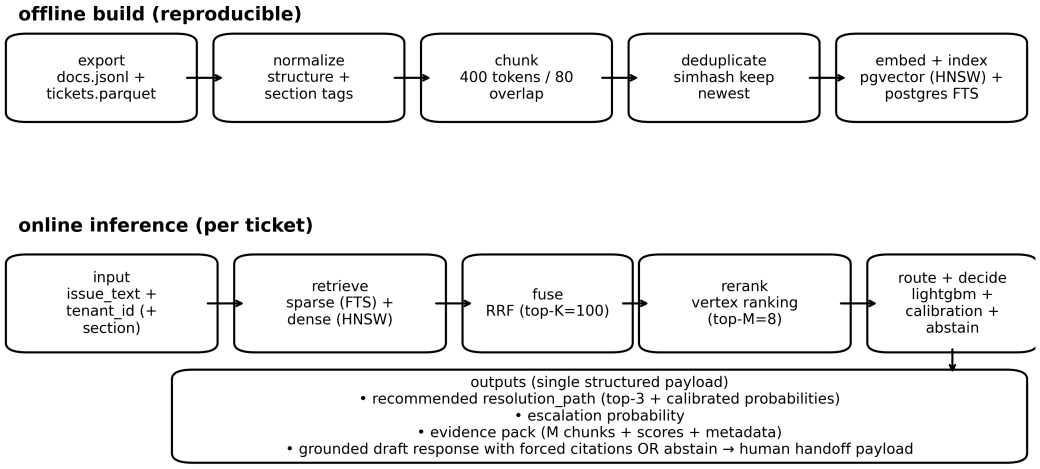

Figure 1: End-to-end system flow. Offline build produces indexed evidence and versioned artifacts. Online inference performs hybrid retrieval, fuses ranks via RRF, reranks evidence, routes the issue, and either answers with forced citations or abstains and hands off with an evidence pack.

2 Contributions

I claim two contributions, stated without ambiguity.

(C1) Reproducible end-to-end blueprint (portable across industries). I specify a complete and reproducible blueprint for an industry-agnostic agent-assist decision system that unifies RAG with supervised routing. The system consumes a canonical two-file data contract and produces a single structured output payload for each issue. Portability is achieved by pushing domain variation into metadata (`tenant_id`, `domain`, `section`) rather than code.

(C2) Calibrated confidence + abstention to reduce false positives. I make calibrated confidence and reject-option behavior first-class outputs. I calibrate routing probabilities and enforce a selective classification policy that abstains under uncertainty. Confidence is computed jointly from calibrated routing probabilities and retrieval evidence quality, and abstention triggers a structured handoff payload rather than an ungrounded answer.

3 Problem setting

Each organization (tenant) provides:

- **Knowledge documents** describing policies, SOPs, and troubleshooting steps.
- **Historical tickets** with labeled outcomes: a discrete `resolution_path` and a binary `escalated`.

At inference time, given `issue_text` and `tenant_id` (and optionally `section`), the system must output:

1. a recommended `resolution_path` with calibrated probabilities,
2. an escalation probability,
3. an evidence pack (retrieved chunks + scores + metadata),
4. a grounded draft response that cites chunk identifiers, or a handoff payload.

4 Data contract

I enforce a minimal, portable schema.

Field	Type	Role
tenant_id	string	Multi-tenant isolation (mandatory filter).
section	string	Sub-unit routing and retrieval constraint.
text	string	Retrieval corpus content (documents).
issue_text	string	Query text for retrieval and routing (tickets).
resolution_path	class	Supervised routing target.
escalated	bool	Supervised escalation target.

Table 1: Minimal fields used by the blueprint. Domain and source are optional descriptors; tenant and section are operational.

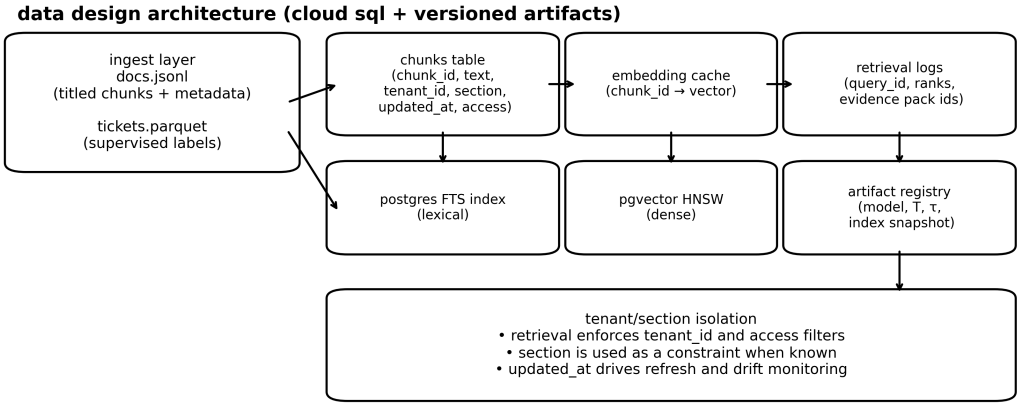

Figure 2: Data design architecture. The blueprint uses PostgreSQL for both sparse and dense retrieval, with explicit tenant and section isolation and a versioned artifact registry.

Documents (`docs.jsonl`). Each record is one chunk with mandatory metadata for isolation and filtering:

```
{"doc_id":"...","title":"...","text":"...","tenant_id":"orgA",
 "domain":"...","section":"...","source":"...","updated_at":"YYYY-MM-DD",
 "access":"internal"}
```

Tickets (`tickets.parquet`). Each record is one labeled support event:

```
ticket_id,tenant_id,domain,section,issue_text,resolution_path,escalated
```

5 System architecture

5.1 Data design architecture

I implement storage and indexing on PostgreSQL (Cloud SQL). I store chunks and metadata, maintain an embedding cache, and create two retrieval surfaces: (i) PostgreSQL full-text search for sparse lexical retrieval and (ii) pgvector HNSW for dense retrieval. I also persist retrieval logs and a versioned artifact registry (model weights, calibration temperature T, and abstention threshold τ).

5.2 Retrieval: chunking, deduplication, and hybrid search

Chunking. I chunk documents into coherent spans with small overlaps. This reduces retrieval granularity and improves citation specificity.

Deduplication. I remove near-duplicate chunks using SimHash [5] so retrieval does not repeatedly surface the same evidence with slightly different formatting.

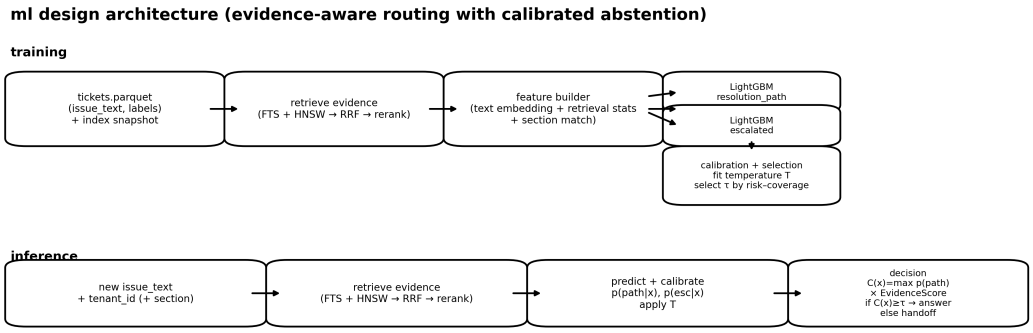

Figure 3: ML design architecture. Training simulates retrieval for historical tickets to produce evidence-aware features, then fits routing models and a calibration + abstention policy. Inference applies calibrated probabilities and a reject option.

Hybrid retrieval and fusion. Given a query, I compute two ranked lists:

- Sparse lexical ranking from PostgreSQL full-text search.
- Dense semantic ranking from pgvector HNSW [6].

I merge the two lists with reciprocal rank fusion (RRF) [2], which is robust to score-scale mismatch across retrieval systems. I then rerank fused candidates using a learned ranker (Vertex AI Ranking) and select the top-M chunks as the *evidence pack*.

5.3 ML design architecture: routing + calibration + abstention

I train two supervised models:

- **Resolution routing:** multi-class classifier predicting `resolution_path`.
- **Escalation risk:** binary classifier predicting `escalated`.

Both models use LightGBM [7] with evidence-aware features derived from the issue text and the evidence pack (e.g., dense embedding of issue text, evidence relevance statistics, section match indicators).

Temperature scaling. I calibrate the routing probabilities using temperature scaling [3]. For multiclass routing logits z, calibrated probabilities are:

$$p_T(y \mid x) = \mathrm{softmax}(z/T), \tag{1}$$

where T is fit on a held-out validation set by minimizing negative log-likelihood.

Evidence score. Let $s_1, \ldots, s_M$ be the reranker scores of the top-M evidence chunks (normalized to $[0, 1]$). I define:

$$\mathrm{EvidenceScore}(x) = \frac{1}{M} \sum_{i=1}^{M} s_i. \tag{2}$$

Single system confidence and reject option. I define a single confidence:

$$C(x) = \max_y p_T(y \mid x) \times \mathrm{EvidenceScore}(x). \tag{3}$$

Decision rule:

- If $C(x) \geq \tau$ and $p(\text{escalated} \mid x) < 0.7$, I output a recommendation and grounded draft response.

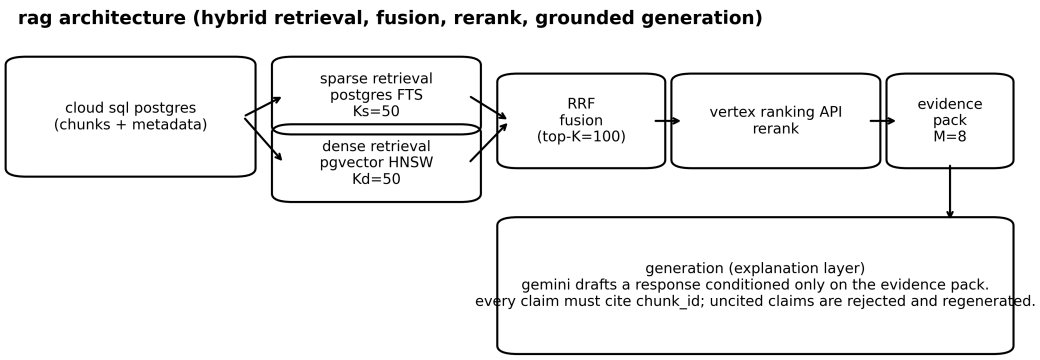

Figure 4: RAG architecture. Hybrid retrieval (sparse + dense) is fused by RRF and reranked into an evidence pack. Generation is conditioned on the evidence pack and constrained to cite chunk identifiers.

- Otherwise, I abstain and output a handoff payload with the evidence pack and model scores.

This is selective classification (reject option) [4] instantiated for a coupled RAG + routing system.

5.4 Grounded generation with forced citations

The generator consumes only the evidence pack plus the issue. I constrain generation to cite chunk identifiers for every actionable claim. Outputs are parsed; if claims lack citations, generation is repeated with an explicit constraint. This enforces traceability and reduces hallucinated resolution steps.

6 Evaluation protocol and metrics

I evaluate three coupled components: retrieval quality, routing quality, and end-to-end safety.

6.1 Retrieval metrics

I report Recall@k and MRR@k using ticket-linked evidence when available. I also report section-constrained retrieval performance to verify that the same tenant can support many sections concurrently.

6.2 Routing metrics

For `resolution_path`: top-1 accuracy, macro-F1, and top-3 accuracy. For `escalated`: AUROC and AUPRC.

6.3 Calibration and selective classification metrics

I report Expected Calibration Error (ECE) for routing probabilities and evaluate abstention via the risk–coverage curve [4]. I select τ by minimizing false positives subject to an application-level coverage target.

6.4 Grounded response metrics

I report citation coverage (fraction of sentences with at least one chunk citation) and evidence precision (fraction of cited chunks ranked in top-M). I also audit abstained cases to confirm that low confidence correlates with ambiguous evidence or distribution shift.

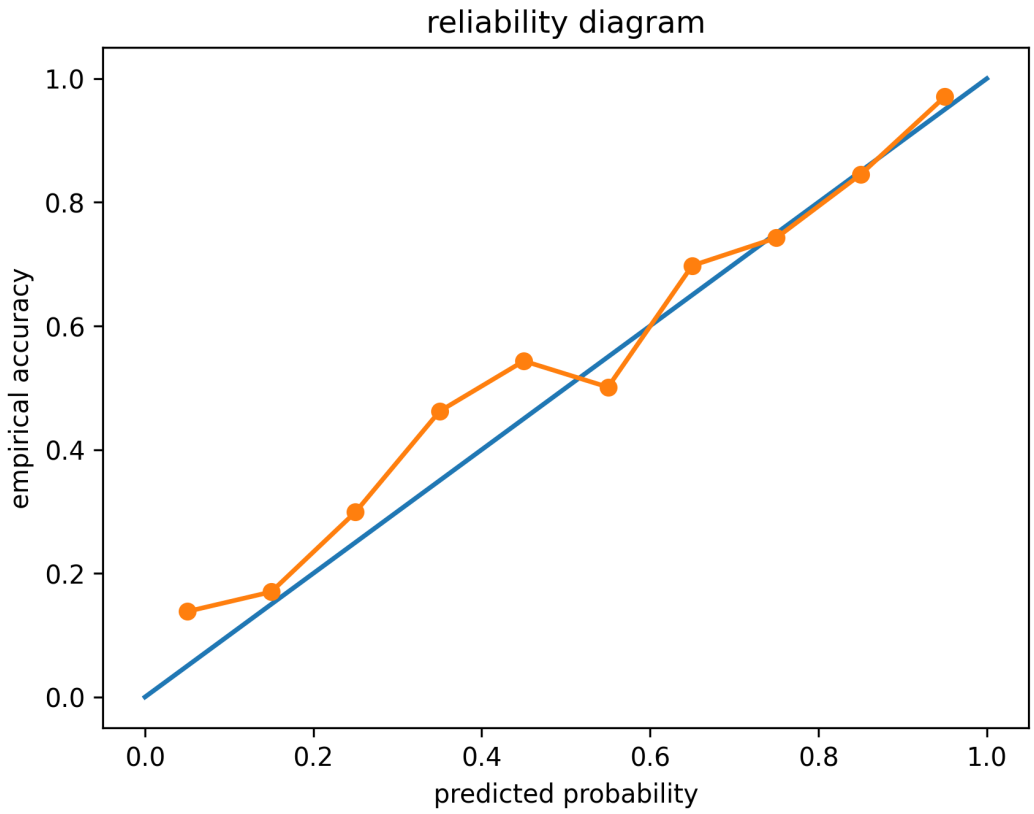

Figure 5: Illustrative reliability diagram for routing probability calibration (temperature scaling), computed on a held-out validation set in experiments.

6.5 System metrics

I report end-to-end latency (P50/P95), throughput, and cost per ticket. Figure 7 shows the latency distribution plot used in experiments.

7 Implementation details

I implement the pipeline as deterministic stages: ingest+normalize, chunk+deduplicate, embed+index, retrieve+fuse+rerank, and route+generate with forced citations and handoff. I version the index snapshot and the routing artifacts (T, τ) to support reproducibility. Multi-tenancy is enforced at retrieval time by mandatory `tenant_id` filtering; sections are used as a retrieval constraint when known.

8 Limitations and responsible deployment

My design assumes organizations provide sufficient labeled tickets to train routing. When labels are sparse, abstention coverage will increase, shifting more volume to human support. I treat abstention as the default safety mechanism: the system prefers handoff over low-confidence automation.

References

References

[1] Patrick Lewis, Ethan Perez, Aleksandra Piktus, et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

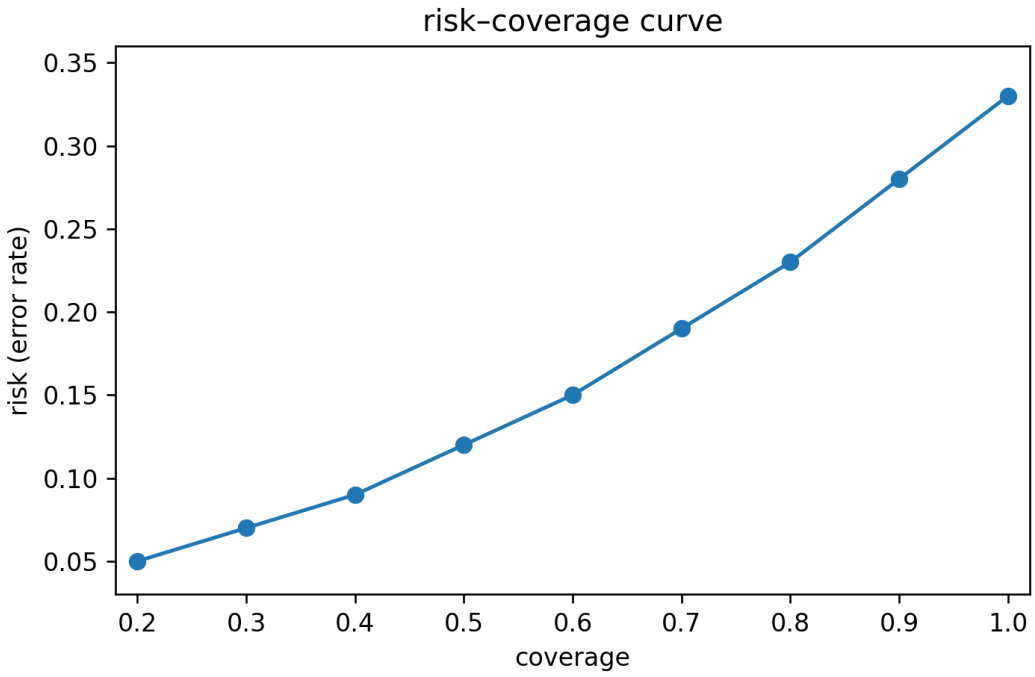

Figure 6: Illustrative risk–coverage curve for reject-option behavior: as the threshold is relaxed, coverage rises and risk typically increases [4].

[2] Gordon V. Cormack, Charles L. A. Clarke, and Stefan Buettcher. Reciprocal rank fusion outperforms Condorcet and individual rank learning methods. In *Proceedings of SIGIR*, 2009.

[3] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of ICML*, 2017.

[4] Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[5] Moses S. Charikar. Similarity estimation techniques from rounding algorithms. In *Proceedings of STOC*, 2002.

[6] Yu. A. Malkov and D. A. Yashunin. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[7] Guolin Ke, Qi Meng, Thomas Finley, et al. LightGBM: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[8] Andrew Kane. pgvector: Open-source vector similarity search for Postgres. Project documentation, 2024.

[9] Google Cloud. Vertex AI Ranking API documentation. Product documentation, 2024.

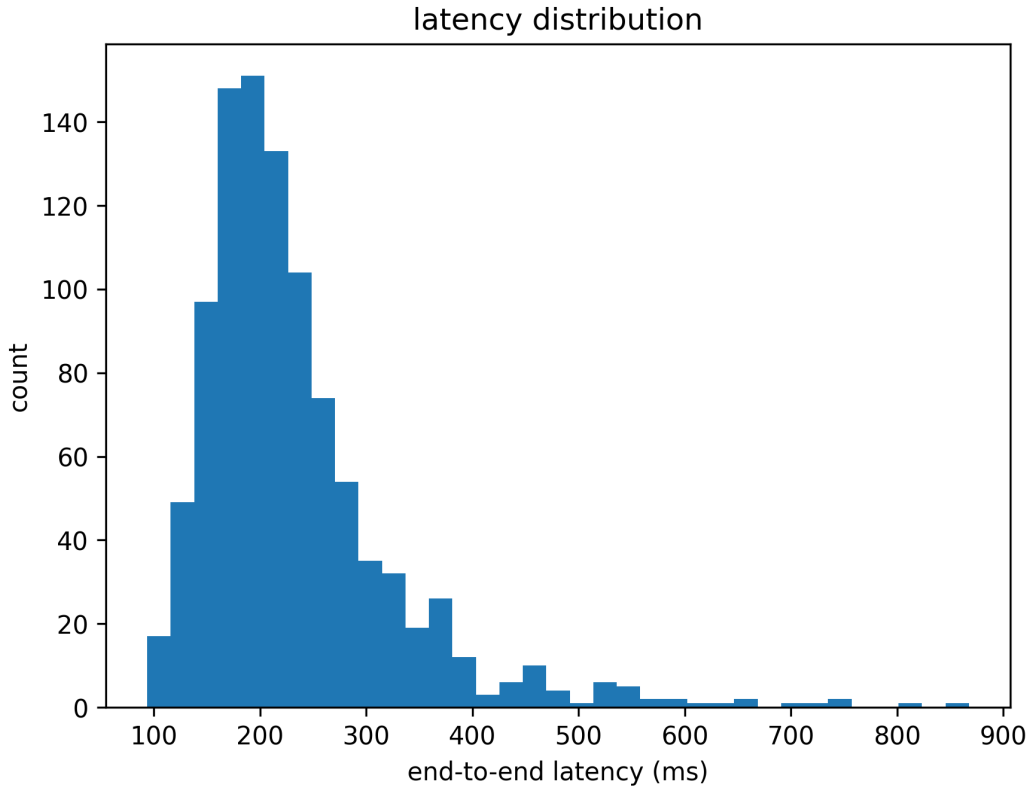

Figure 7: Illustrative end-to-end latency distribution (P50/P95) reported in evaluation.