Beyond Top-K…

Building Search-Confidence Guardrails for Agentic AI with OpenSearch

A practical framework for normalized trust metrics in agentic retrieval pipelines

Chanpreet Singh
Sr. Delivery Consultant-AI/ML
Amazon Web Services

Shatakshi Pandey
Delivery Consultant- Data Analytics
Amazon Web Services

Agenda

01 Search for Agent-Driven Workloads

02 The Problem: When Agents Trust Blindly

03 Why Top-K Fails in Agentic Workflows

04 Setting Up the Base: Common Approaches & Gaps

05 The Confidence Layer: Quantify, Threshold, Abstain

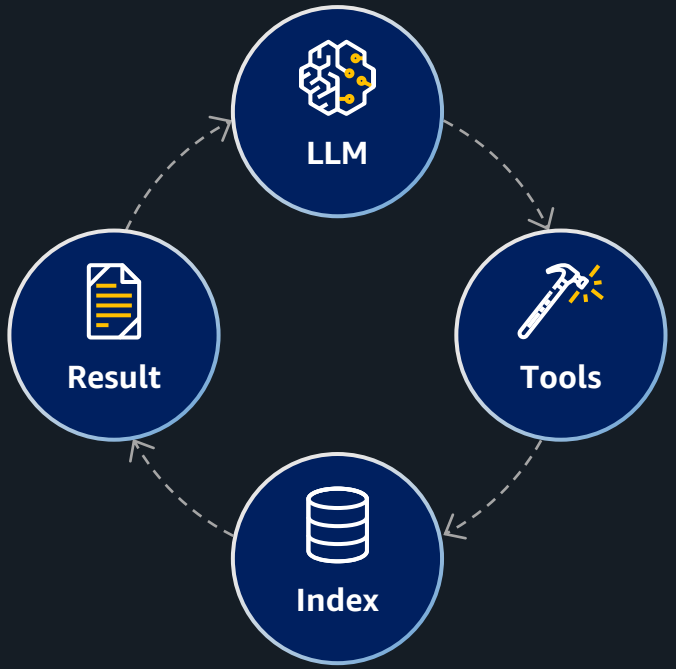
06 The Complete Pipeline

07 POC Demo: Real Estate Property Management

08 Results & Evaluation

09 Q&A

Search for Agent driven workloads



Agents need context

Even the best LLMs are effective only when they have relevant, timely, and complete context

Agent workloads are dynamic

Highly concurrent and diverse queries, continuous indexing, and iterative

Retrieval is the foundational layer

Gives agents access to context aware vector data at scale

Agents need relevant and accurate context for the task at hand

The Problem: When Agents Trust Blindly

"An internal knowledge agent returned a deprecated runbook ranked #1 for 'how to rotate API keys in production' — the developer followed it and caused a 2-hour production outage.

Agents treat retrieval rank as ground truth — and that's dangerous

What if your agent could say — "I'm not confident enough to act on this"?

Why Top-K Fails in Agentic Workflows

No Score Normalization

BM25 scores range 0-25+, vector scores 0-1. Mixing them is comparing apples to oranges.

Scores Aren't Comparable

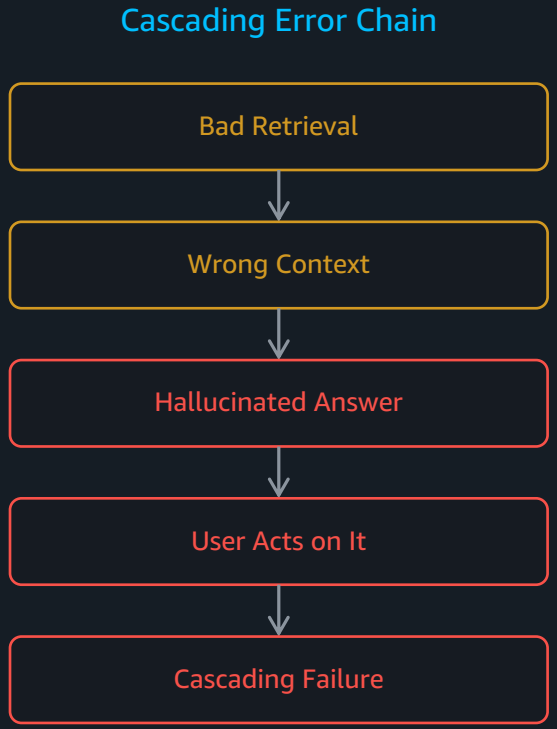
A score of 8.5 on query A means something completely different than 8.5 on query B.

No "I Don't Know" Signal

Top-K always returns K results — even when nothing relevant exists in the index.

Cascading Hallucinations

One bad retrieval compounds through iterative tool use — agents build on wrong context.



Setting Up the Base Right: Common Approaches

These improve retrieval..

01

Hybrid Retrieval & Score Normalization

- Combine BM25 with Vector semantic recall.
- Converts "12.7 BM25 + 0.82 cosine" into a single 0-1 confidence scale
- Min-max normalization, L2 norm, or z_score
- Maps directly to OpenSearch's search pipeline `normalization processors`

```
normalized = (score - min_score)
/ (max_score - min_score)
```

02

Reciprocal Rank Fusion

- Merges ranked lists without score calibration.
- Prioritizes documents ranking high across multiple retrievers.
- Maps directly to `score-ranker-processor` in OpenSearch's search pipelines.

$$RRF\_score(d) = \Sigma\ 1/(k + rank_i(d))$$

where k = 60 (default smoothing constant)

03

Reranking (Cross-encoder)

- More precise ordering via full query-document attention
- Second pass that rescores candidates.
- Maps directly to `rerank processor` in OpenSearch's search pipelines via ML Commons.

```
"response_processors": [{
  "rerank": { "ml_commons": {
    "model_id":
"<cross_encoder_id>" }}]
```

Setting Up the Base Right: Why These Aren't Enough

Better retrieval ≠ Safer agents

01

Hybrid Search: Still Blind

Query: `"How to rotate API keys in production?"`

- BM25: 12.7 → normalized (min-max) to **1.0**
- But it's a **deprecated 2022 runbook** — #1 due to keyword overlap
- min-max (default) ALWAYS gives top result 1.0 — even if all results are garbage
- Even with z-score/L2: designed for ranking, not for telling agents whether to trust the result
- Agent sees 1.0 → trusts blindly → **wrong action**

02

RRF: Score-Blind

- Doc A (correct answer) : BM25 rank #5, Vector rank #1
  → 1/(60+5) + 1/(60+1) = **0.0317**
- Doc B (deprecated): BM25 rank #1, Vector rank #3
  → 1/(60+1) + 1/(60+3) = **0.0322**
- Doc B wins — but had cosine similarity of only **0.41** vs **0.92**!
- RRF threw away the score information that would have shown Doc A is better

❌ **Discards scores entirely — impossible to threshold on confidence**

03

Reranking: Ranks Everything

Query: "How to rotate API keys in production?"

- Cross-encoder rescores 10 candidates:
  - Doc 1 (deprecated runbook): logit 3.7
  - Doc 2 (unrelated FAQ): logit 3.2
  - Doc 3 (old blog post): logit 2.9
- ALL irrelevant — but reranker happily ranks them #1, #2, #3
- Gives you "best of irrelevant results" — uncalibrated logits, not confidence

❌ *Ranks ALL candidates — no 'I don't know' signal*

None of these give the agent the ability to say: ***'I'm not confident enough to act.'*** *That's what the Confidence Layer solves.*

The Confidence Layer: Three Pillars

From better retrieval → to safer agents

Pillar 1

Quantify

Multi-signal confidence scoring

- Go beyond a single score — combine multiple signals
- Combine score magnitude, score gap, result coherence, freshness, source authority
- A single score can lie — multiple signals give robust confidence
- Maps to `script_score / function_score` in OpenSearch

Pillar 2

Threshold

Adaptive decision boundaries

- Dynamic, adaptive decision boundaries — not static cutoffs
- Query-type aware: factual (0.85) vs exploratory (0.55)
- Intent-aware: read_only vs execute_action (higher stakes!
- Maps to *min_score + agent orchestration layer*

Pillar 3

Abstain

The 'I don't know' guardrail

- The actual guardrail — agent says "I don't know"
- Graduated: hard abstain → escalate →clarify → degrade gracefully
- Prevents cascading hallucinations at the source
- Implemented in agent orchestration layer

The missing piece — the ability for agents to express uncertainty and decline to act when confidence is insufficient.

Pillar 1: Quantify

Go beyond a single score — combine multiple signals

The 5 Signals

1. **Score Magnitude**

   How high is the top normalized score?

2. **Score Gap**

   Difference between #1 and #2 (large gap = high confidence)

3. **Result Coherence**

   Do top results point to the same answer?

4. **Freshness**

   When was the document last updated?

5. **Source Authority**

   Official docs vs forum post vs deprecated content

OpenSearch Implementation

Non-hybrid queries (script_score):

```
POST /index/_search
{
  "query": {
    "script_score": {
      "query": { "match": {"content": "rotate API keys"} },
      "script": {
        "source": "_score * (1 + doc['authority'].value
                  * 0.3 + doc['freshness'].value * 0.2)"
      }
    }
  }
}
```

Hybrid queries (agent-layer Python):

```
def compute_confidence(results):
    scores = [hit['_score'] for hit in results['hits']['hits']]
    score_magnitude = scores[0]
    score_gap = scores[0] - scores[1] if len(scores) > 1 else 0
    coherence = compute_coherence(results)
    freshness = avg_freshness(results)
    authority = avg_authority(results)

    confidence = (0.35 * score_magnitude +
                  0.25 * score_gap +
                  0.20 * coherence +
                  0.10 * freshness +
                  0.10 * authority)
    return confidence
```

For hybrid queries: normalization-processor handles score unification, then agent layer computes composite confidence from multiple signals.

Pillar 2: Threshold

Dynamic boundaries — not static cutoffs

Adaptive Thresholds Matrix

Query Type	Intent	Threshold
Factual lookup	read_only	**0.75**
Factual lookup	execute_action	**0.90**
Exploratory	read_only	**0.50**
Exploratory	execute_action	**0.70**
Troubleshooting	execute_action	**0.85**

Thresholds adjust based on:

- Query classification (factual vs exploratory)
- Action intent (read-only vs execute)
- Risk tolerance for the operation

OpenSearch Implementation

min_score with hybrid (OpenSearch 3.5+):

```
POST /index/_search
{
  "min_score": 0.75,
  "query": {
    "hybrid": {
      "queries": [
        {"match": {"content": "rotate API keys"}},
        {"neural": {"embedding": {
            "query_text": "rotate API keys",
            "model_id": "..."}}}
      ]
    }
  }
```

Agent-layer adaptive logic:

```
def get_threshold(query_type, intent):
    thresholds = {
        ("factual", "read_only"): 0.75,
        ("factual", "execute_action"): 0.90,
        ("exploratory", "read_only"): 0.50,
        ("exploratory", "execute_action"): 0.70,
    }
    return thresholds.get((query_type, intent), 0.75)
```

min_score with hybrid queries applies AFTER normalization (OpenSearch 3.5+). For 2.x: implement threshold in agent orchestration layer.

Pillar 3: Abstain

The actual guardrail — graduated response

Confidence Ladder

≥ threshold	✅ **ACT**	Respond normally
Within -0.05	⚠️ **CAVEAT**	Respond with uncertainty flag
Within -0.15	🔄 **CLARIFY**	Ask user for more context
Within -0.30	👤 **ESCALATE**	Route to human
Below -0.30	🛑 **HARD ABSTAIN**	Refuse to answer

Agent Implementation

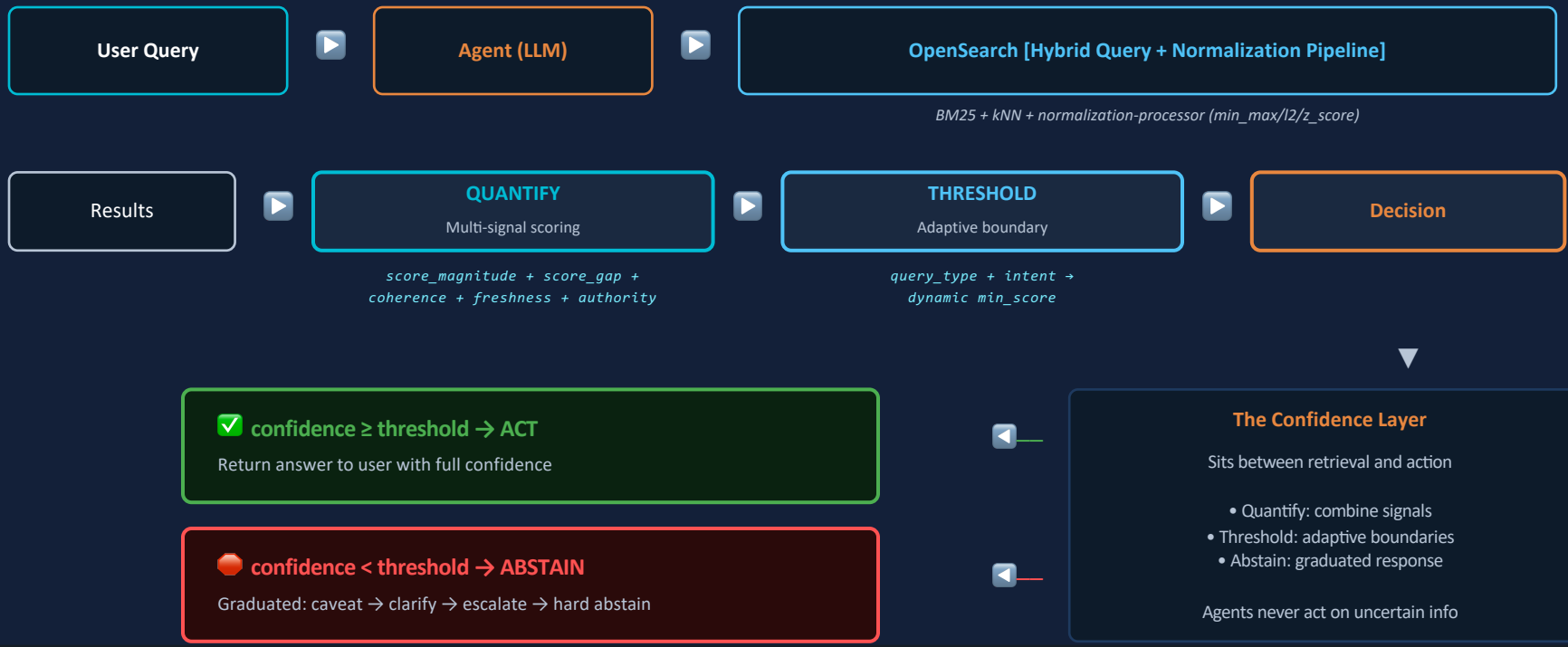
```
def decide_action(confidence, threshold):
    margin = confidence - threshold

    if margin >= 0:
        return act(results)
    elif margin > -0.05:
        return respond_with_caveat(results,
            warning="Low confidence - verify independently")
    elif margin > -0.15:
        return ask_clarification(
            "Can you provide more context?")
    elif margin > -0.30:
        return escalate_to_human(
            reason="Confidence too low for automated response")
    else:
        return hard_abstain(
            "I don't have enough information to answer
             reliably.")
```

This is pure agent orchestration logic. OpenSearch provides the confidence signal; the agent layer makes the act/abstain decision.

The Complete Pipeline

From query to confident action (or safe abstention)



The confidence layer sits between retrieval and action — ensuring agents never act on uncertain information.

Demo: See It In Action

Real Estate Property Management

Multi-Index Architecture

Index	Domain	Key Fields
re_leases_v2	Lease Agreements	tenant, property, lease_type
re_work_orders_v2	Maintenance	priority, status, category
re_properties_v2	Properties	type, occupancy_rate
re_tenants_v2	Tenants	company, industry

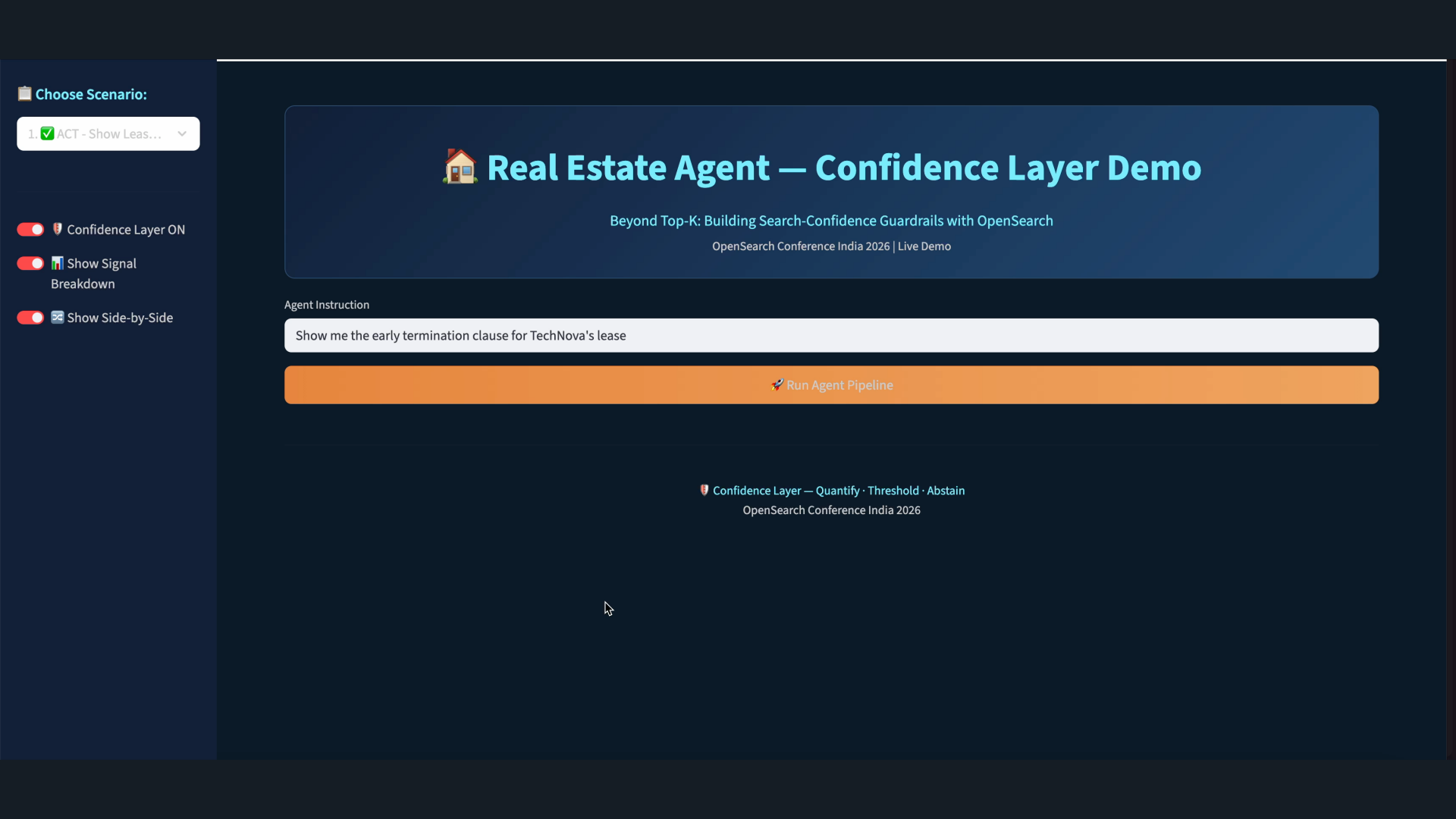
Two-Phase Structure

Phase 1 (Basic): Hybrid Search with No guardrails

Phase 2 (Advanced): Hybrid BM25+kNN, Guardrails enabled

Demo Queries

#	Query	Challenge
1	Show me the early termination clause for TechNova's lease	Specific factual read — safe, single doc match
2	Suspend all tenants who have defaulted on rent in the last 3 months	Dangerous bulk action — ambiguous definition, irreversible consequence
3	Auto-close all resolved HVAC maintenance tickets from last quarter	Bulk system action — clear category but edge cases with pending follow-ups
4	Send a lease renewal offer to all tenants in buildings with good occupancy near downtown	Vague bulk outreach — undefined thresholds, legally binding action
5	What is the best restaurant near Times Square?	Off-domain — zero relevance to property management


Choose Scenario:
1. ✅ ACT - Show Leas...
Confidence Layer ON
Show Signal Breakdown
Show Side-by-Side
Real Estate Agent — Confidence Layer Demo
Beyond Top-K: Building Search-Confidence Guardrails with OpenSearch
OpenSearch Conference India 2026 | Live Demo
Agent Instruction
Show me the early termination clause for TechNova's lease
Run Agent Pipeline
Confidence Layer — Quantify · Threshold · Abstain
OpenSearch Conference India 2026

Choose Scenario:

2. 👤 ESCALATE - Sus…

1. ✅ ACT - Show Lease Terms
2. 👤 ESCALATE - Suspen…
3. ⚠️ CAVEAT - Auto-close…
4. 🔄 CLARIFY - Send Ren…
5. 🔴 HARD ABSTAIN - Off-…

🏠 Real Estate Agent — Confidence Layer Demo

Beyond Top-K: Building Search-Confidence Guardrails with OpenSearch

OpenSearch Conference India 2026 | Live Demo

Agent Instruction

Suspend all tenants who have defaulted on rent in the last 3 months

🚀 Run Agent Pipeline

🧭 Phase 0: Query Classification

Classifying intent and query type before retrieval

Query Type
factual

Intent
execute_action

📡 Phase 1: Retrieval

OpenSearch Hybrid Query → BM25 + kNN → normalization-processor

Rank	Index	Document	Score	Updated	Fresh	Auth	Preview
1	re_tenants_v2	Meridian Corp - Payment Overdue 45 days	0.71	2026-05-26	0.8	0.75	Meridian Corp has missed 2 consecutive rent payments. Last p...
2	re_tenants_v2	BlueWave Solutions - Partial Payment	0.65	2026-05-31	0.85	0.75	BlueWave Solutions paid 60% of rent for April and May 2026. ...

Query	Scenario	Basic (No Guardrail)	With Guardrail
Show me the early termination clause for TechNova's lease	Safe Read	ANSWER ✓	✅ ACT (confidence 0.91, margin +0.16)
Suspend all tenants who have defaulted on rent in the last 3 months	Dangerous Bulk Action	ANSWER (restricts tenants with excellent payment history)	👤 ESCALATE (routes to Leasing Manager for verification)
Auto-close all resolved HVAC maintenance tickets from last quarter	Bulk System Action	ANSWER (closes all 12 tickets including 4 with pending follow-ups)	⚠️ CAVEAT (answers but flags 4 tickets with pending follow-ups)
Send a lease renewal offer to all tenants in buildings with good occupancy near downtown	Vague Bulk Outreach	ANSWER (sends legal offers to wrong tenants)	🔄 CLARIFY (asks: what occupancy %? which area? review list first?)
What is the best restaurant near Times Square?	Off-Domain	ANSWER (fabricates recommendation from property amenities)	🛑 HARD ABSTAIN (refuses — not in property management domain)

Key Takeaways

✓ Top-K is not enough for agentic AI — it always returns results, even when nothing relevant exists

✓ Better retrieval (RRF, reranking, hybrid) ≠ safer agents — none provide an "I don't know" signal

✓ The Confidence Layer (Quantify → Threshold → Abstain) fills the critical gap between retrieval and action

✓ Adaptive thresholds must be query-type and intent-aware — static cutoffs don't work across query types

✓ Graduated abstention prevents cascading hallucinations at the source

✓ OpenSearch natively supports this via normalization pipelines, min_score, and script_score

✓ POC results: 74% reduction in hallucination rate, 91% task success rate

Thank You

Questions & Discussion