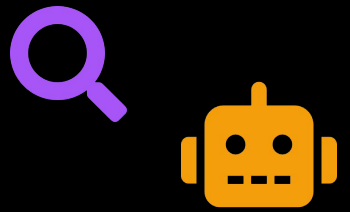
shutterstock

From Boolean Search to Agentic Generative Discovery



Orchestrating 25 Years of IR Innovation

Optimized AI Conference 2026

Rajani Maski
Staff Software Engineer, AI · Shutterstock

Shutterstock : Search at the core

The scale that demands this architecture

500M+

Creative Assets

Images · Video · Music · Sound FX

2M+

Contributors Worldwide

Content grows daily

500K+

Daily Search Sessions

From professionals globally

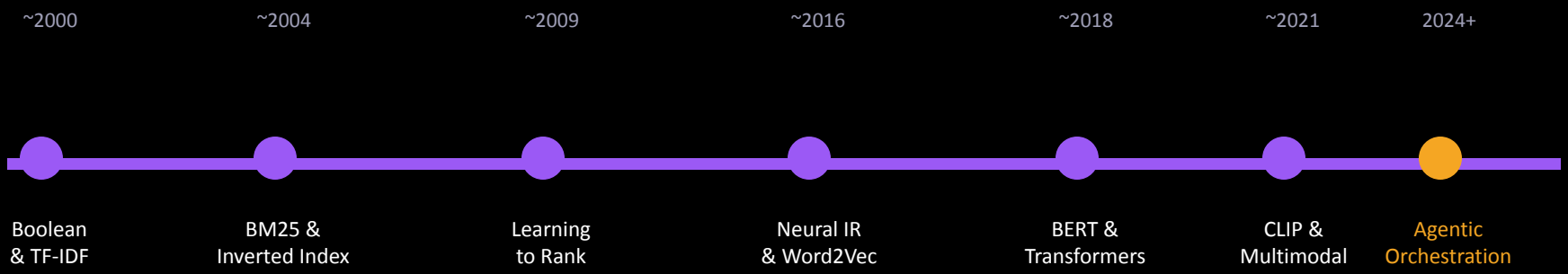
25+

Languages & Modalities

Text · Voice · Image · Audio

Stack: AWS · OpenSearch · S3 · SageMaker · Lambda · Bedrock

25 Years of IR Innovation: Still Running in Production



None of these are deprecated. They are all tools in the agent's toolkit.

Lexical(BM25) Search: The Load-Bearing Infrastructure

Why term-based retrieval still outperforms neural models in critical cases

Exact Match Precision

SKU IDs, proper nouns, file names. Vectors hallucinate; lexical retrieval nails it.

Latency Profile

Inverted index lookups are sub-millisecond. Dense retrieval is 10-50x slower.

Cold Start Reliability

New assets have no embedding neighborhood. Lexical retrieval works day one.

Explainability

Every BM25 score is auditable. Critical for debugging and editorial trust.

OpenSearch: Lexical (BM25) and vector (k-NN) in one engine — one ops team, full retrieval stack.

Multimodal Search: The Shutterstock Reality

Users do not think in modalities — our retrieval stack has to bridge that gap

Image

CLIP embeddings + visual similarity

Reference image ⟶ semantic matches

Video

Frame-level CLIP + temporal signals

Motion, scene, mood detection

Music

Audio embeddings + BPM/mood taxonomy

‘Upbeat corporate’ ⟶ audio retrieval

Voice

ASR ⟶ intent ⟶ structured query

Natural expression, no query syntax

All modalities indexed in OpenSearch — unified retrieval surface for the agent.

Hybrid Fusion: When 1 + 1 > 2

Three patterns — each optimal for different query types

RRF

Reciprocal Rank Fusion

Merge BM25 + vector rankings by reciprocal rank. No score normalization needed.

Best for: General discovery queries

Weighted Combo

Score Combination

α·BM25 + β·semantic

Tune α/β per query class.

Best for: Known-item + semantic mix

Cascade Re-rank

Cross-Encoder Reranking

BM25 retrieves candidates then cross-encoder re-ranks top-K.

Best for: Precision-critical editorial

Agent selects the fusion pattern at query time based on detected intent class.

Agentic Search Patterns

Four behaviors that distinguish an agentic IR system from a search API

Reflection

Agent evaluates its own retrieval output, detects gaps (low confidence, missing modality coverage, ambiguous results), and decides to re-query with refined parameters.

Planning

Before executing retrieval, the agent decomposes a complex query into sub-goals: what entities to resolve, which modalities to search, what order to execute in.

Tool Use

BM25, dense retrieval, knowledge graph, CLIP, cross-encoder — each exposed as a discrete tool. The agent selects and composes tools dynamically per query.

Multi-Agent

Specialized sub-agents handle modality-specific retrieval (image agent, audio agent, text agent). A coordinator agent fuses and re-ranks across all sub-agent outputs.

Agentic Workflow: From Intent to Discovery

How an LLM turns natural language into structured retrieval — via MCP and OpenSearch

01

Search Intent Detection

LLM classifies query intent:
Discovery · Known-item · Similarity
Refine · Editorial browse

02

Entity Extraction & Focus

Extract: subject, style, mood, modality, license type, era, brand. Build structured attribute map.

03

LLM → Query Strategy

LLM decides: which retrieval tools, which fusion pattern, which filters. Outputs a query execution plan.

04

OpenSearch via MCP

MCP server exposes BM25 full-text, k-NN vector, and filter endpoints. Agent executes and fuses results.

Architecture: LLM Agent → calls tool → OpenSearch MCP Server → executes against OpenSearch index → results back to agent

MCP: OpenSearch as a Search Tool for the Agent

```
// Tool exposed by the OpenSearch MCP Server
// The LLM agent calls this tool — OpenSearch executes it
{
   "name": "opensearch_search",
   "description": "Hybrid lexical + vector search over
   Shutterstock assets",
   "inputSchema": {
      "query": { "type": "string" },
      "mode": { "enum": ["bm25", "knn", "hybrid"] },
      "filters": { "type": "object" },
      "top_k": { "type": "integer", "default": 20 }
   }
}
```

Reflection

Agent checks result quality; retries with refined params if confidence is low

Tool Use

Each mode (bm25/knn/hybrid) is a discrete capability the LLM can reason about

Planning

Filters + mode selection happens before execution — structured query plan

Live Demo: Agent Search in Action

From natural language to structured asset discovery

Demo Architecture

Frontend

Node.js / React — Chat UI with streaming reasoning panel

Backend

Python / FastAPI — Agent orchestration layer

Agent Squad Router

GPT-4o-mini — Routes to specialist sub-agents

MCP Server

OpenSearch MCP Server — Exposes BM25, k-NN, hybrid endpoints

OpenSearch (AWS)

Execution engine — Lexical + vector retrieval at scale

Agents in Workflow

Project Manager Agent

Parses creative brief, decomposes into retrieval sub-goals, coordinates results

Search Specialist Agent

Calls OpenSearch Tool via MCP — selects BM25 / k-NN / hybrid per intent

Reflection Reranking Agent

Evaluates result confidence, detects gaps, re-queries with refined parameters

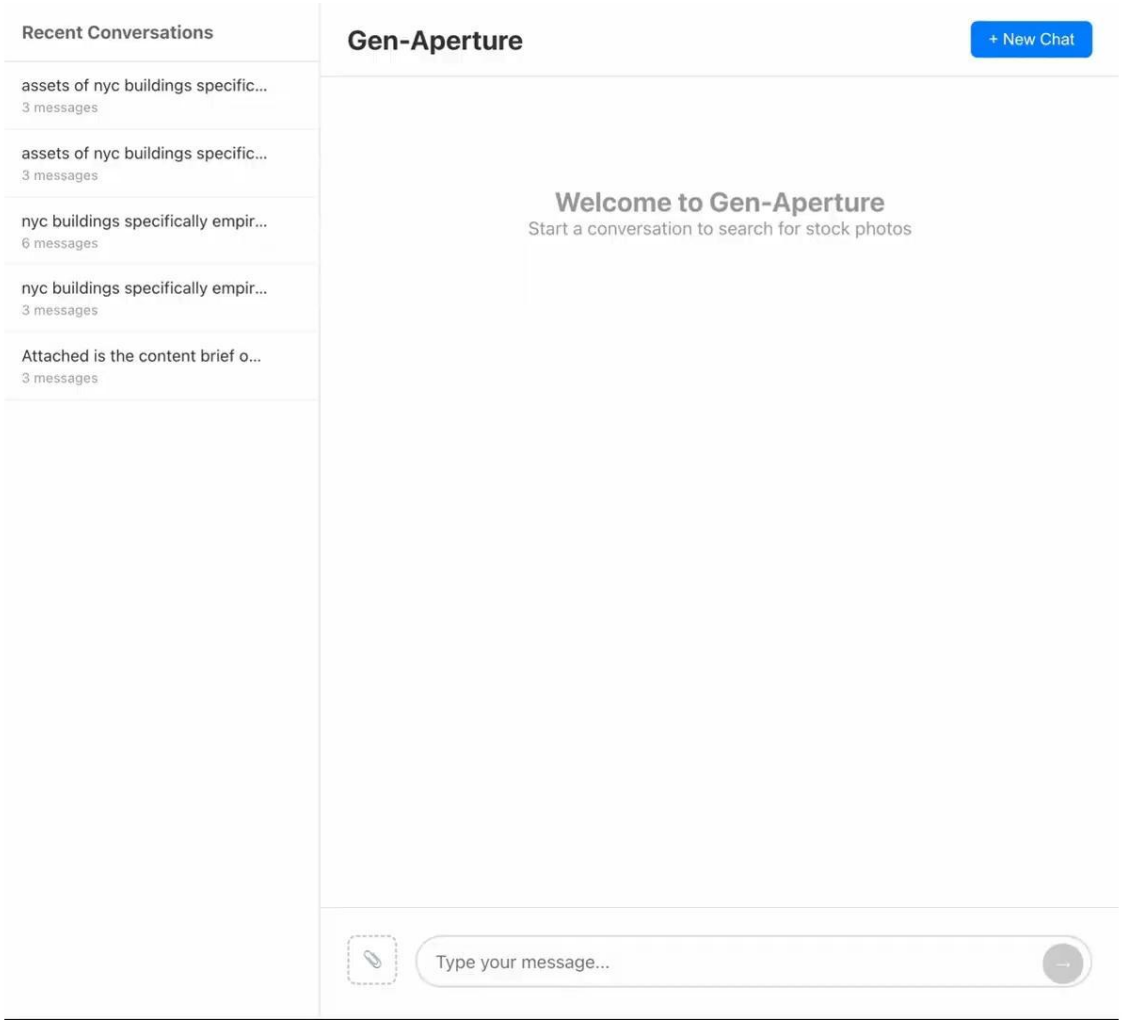
File Analysis Tool

Reads uploaded reference images or docs to extract style/mood signals

Demo Scenarios

1. Planning agent pattern: a content brief workflow for asset discovery.
2. Reflection-based agent pattern: evaluate asset quality and uniqueness.
3. Tool use: multimodal search and aggregation

Stack: Node.js/React · Python/FastAPI · GPT-4o-mini · OpenSearch MCP Server · OpenSearch (AWS)


Recent Conversations
assets of nyc buildings specific...
3 messages
assets of nyc buildings specific...
3 messages
nyc buildings specifically empir...
6 messages
nyc buildings specifically empir...
3 messages
Attached is the content brief o...
3 messages
Gen-Aperture
+ New Chat
Welcome to Gen-Aperture
Start a conversation to search for stock photos
Type your message...

The Orchestrated Future — Takeaways by Role

Search Engineers

- Expose BM25, k-NN, and filters as discrete MCP tools
- OpenSearch handles both workloads: one ops surface
- Query routing table: intent class to retrieval strategy
- Instrument every tool call; non-determinism requires logging

AI Engineers

- RAG pipelines need lexical anchors: pure vector retrieval drifts
- Use Reflection and Planning before any retrieval tool call
- Multi-agent fan-out for modalities; coordinator fuses results
- Bedrock and SageMaker on AWS: LLM inference and model serving unified

Engineering Managers

- Hybrid search ROI: measurable lift on precision@K and session depth
- Agentic architecture: existing infra becomes composable tools
- AWS-native stack reduces operational surface area
- Invest in eval frameworks: non-determinism hides regressions

The orchestra was always here. We finally have a conductor.