

# How to Search Your Knowledge Base by Meaning Without a Vector Database Vendor

You want search that understands meaning, not just keywords, but you do not want to hand your corpus and your query path to a proprietary vector database. This guide describes an architecture for resolving and discovering resources by meaning and lineage through scoped anchors instead of a central embedding index. It is an architecture disclosed in United States Patent Application 19/647,395, not a shipping library, and its home inventive step is the Semantic Discovery inventive step. You implement it yourself.

---

## What You Are Building

You have a knowledge base: documents, records, agents, endpoints, whatever your organization addresses and reasons over. Keyword search misses anything phrased differently from the query. The usual fix is to embed everything into vectors, push those vectors into a hosted vector database, and query by nearest neighbor. That works, but it centralizes your corpus, your access decisions, and your query traffic inside one index owned by one vendor, and the index itself does not know or enforce who is allowed to see what.

This guide describes a different approach: search by meaning as a governed walk through a decentralized index of scoped anchors, where each anchor describes its own semantic neighborhood and enforces its own access policy, and where a query is a

persistent object that accumulates state and lineage as it traverses. The goal is meaning-aware discovery, per-anchor governance, and an auditable trail of how each result was reached, without a central embedding store as the single source of truth. This is the Semantic Discovery inventive step disclosed in United States Patent Application 19/647,395.

## **Why the Obvious Approaches Fall Short**

The standard meaning-based search stack is retrieval-augmented: a separate embedding model turns content into vectors, a vector database returns nearest neighbors for a query vector, and often a language model summarizes what came back. This is a real and effective pattern. Nearest-neighbor vector search is accurate and fast for its intended job. The gap is structural, not a performance defect, and it is worth stating fairly.

First, in that pattern the index is passive. It answers a query posed to it by an external process and returns candidates; it does not itself evaluate whether the requester is authorized, whether the transition is consistent with what was already retrieved, or whether an intermediate step should have been taken at all. Governance, if present, is a filter applied before or after retrieval, not a property of the index. The specification describes this plainly: conventional systems treat search, inference, and execution as distinct subsystems connected by interfaces that lose semantic context at each boundary crossing, and it distinguishes itself from retrieval-augmented generation in which the search index is a passive retrieval target and no governance evaluation is applied to intermediate retrieval or reasoning transitions.

Second, the query is stateless. A query string is evaluated once and discarded; a query embedding is a static vector that does not evolve during retrieval. Neither carries the accumulated context of a multi-step search, so there is no structured memory to check a new candidate against the trajectory so far.

Third, a hosted vector database concentrates the corpus and the access surface in one place. Possessing a reference or being able to reach the endpoint tends to confer the ability to retrieve, because the store's job is to return neighbors, not to decide eligibility per hop.

The architecture below targets exactly these three gaps: an active index, a stateful query, and governance carried at every step.

## **The Architecture**

The approach has four parts, all traced to the filed specification.

**A decentralized anchor index instead of a central embedding store.** Every addressable object (content, an identity, a knowledge node, an agent, a service endpoint) is assigned to a nested container governed by an anchor object. Each anchor encodes a mutation policy, a quorum threshold, an alias mapping, and historical lineage metadata. Crucially, each anchor maintains its own description of its reachable semantic neighborhood, called the neighborhood publication, and computes it itself rather than deferring to a central authority. That publication includes a semantic content descriptor (an abstracted description of the semantic territory the container covers, not an enumeration of every object), a reachability graph of directly navigable sub-anchors and peer anchors, a policy envelope of governance constraints for entities traversing the container, a freshness indicator, and an entropy summary describing the diversity and update frequency of the container. This is how the index becomes traversable by meaning without any traverser needing prior knowledge of the whole structure.

**The query as a persistent discovery object.** A query is not a string, a keyword list, a vector embedding, or a prompt. It is instantiated as a discovery object: a persistent, memory-resident semantic entity carrying the full context of the traversal as typed fields. The specification lists these fields: an intent field (a structured objective with a

goal type, domain scope, resolution criterion, and specificity constraints, not a natural-language string), a context field (situational parameters), a memory field (accumulated semantic commitments from admitted steps), a policy field (governance constraints the traversal must respect), a lineage field (the ordered record of admitted transitions), an affect field (modulation parameters), and a confidence field (whether the traversal is making adequate progress). Because it carries a policy field and a memory field, the query itself can be checked against both what it is allowed to do and where it has already been.

**The three-in-one traversal step.** At each anchor boundary the discovery object performs three coupled phases in sequence: a search step evaluates the object's semantic state against the anchor's published neighborhood to produce candidate transitions; an inference step scores, ranks, or selects among those candidates, using the neighborhood's entropy summary as one input; and an execution step evaluates the selected transition for admissibility under deterministic policy before it is committed. No transition happens without all three. The object then advances to the next anchor and repeats. This is how a single walk simultaneously narrows the search, updates semantic state, and enforces governance, which is the substance of the inventive step.

**Meaning-and-lineage resolution through scoped anchors.** Two mechanisms make the walk meaning-aware and auditable. The neighborhood publication is scoped to the requester: two discovery objects with different policy profiles can receive different publications from the same anchor, so restricted callers simply do not see neighborhoods they are not authorized to enter. And addressing is by structured alias of the form `type@domain.subdomain/path` (for example, `article@org.wikipedia/computing`). An alias is resolved by walking the index one path segment at a time, not by consulting a lookup table, so the resolution always reflects the live index and, because each hop is governed, possessing an alias does not by itself grant access. Every admitted step, and every structural redirect when anchors reorganize, is written to the lineage field, giving each result an auditable provenance chain rather than an opaque similarity score.

For grounded generation, the same substrate supports anchored semantic resolution: when a transition references an external concept, that reference must resolve to a verified referent before it is committed, producing one of three outcomes described in the spec (resolved, unresolvable, or ambiguous), where an unresolvable reference is rejected rather than fabricated.

## How to Approach the Build

You are implementing this yourself; the specification describes the architecture, not a package you install. A reasonable order:

1. **Model the anchor and its container.** Give each anchor an identity, a governed set of member objects, a mutation policy, and a policy envelope. Start with a static hierarchy; you can add self-organization later.
2. **Define the discovery object schema.** Encode the seven fields as typed structures. The most important early decision is making intent structured (goal type, domain scope, resolution criterion, specificity constraints) rather than a raw string, because the whole search step depends on comparing intent to neighborhood descriptors.
3. **Implement the neighborhood publication.** Have each anchor compute its own publication from its current container state. An illustrative interface sketch, faithful to the spec's listed components and clearly not production code:

```
publication(anchor, requester_policy) -> {  
  semantic_content_descriptor, // abstracted territory description  
  reachability_graph,         // navigable sub/peer anchors + relations  
  policy_envelope,           // constraints for traversing entities  
  freshness_indicator,       // epoch of last update  
  entropy_summary            // diversity / update-frequency measure  
}
```

Note the `requester_policy` argument: scoping the publication to the caller is what enforces access at the meaning layer.

- 4. Implement the three-in-one step as one indivisible operation.** search produces candidates, inference selects, execution admits or rejects under deterministic policy, and only an admitted transition mutates the memory and lineage fields. Do not let a candidate influence state before the execution phase clears it.
- 5. Implement navigational alias resolution.** Resolve `type@domain.subdomain/path` by stepwise traversal from the domain anchor, evaluating requester policy at each hop. Do not build a central alias-to-location table; the path through the index is the address.
- 6. Record lineage and check drift.** Append each admitted transition to lineage. The spec also describes a traversal integrity mechanism that measures semantic drift between the accumulated state and the original intent and can re-anchor, branch, or halt; add this once the basic walk works, so long searches do not wander off intent.
- 7. Add self-organization last.** Splitting, merging, migration, and alias rekeying let anchors rebalance under load, all under deterministic policy per the spec. This is an optimization; the walk is correct without it.

Pick one operating mode to start. The spec describes three over the same substrate: human search mode (return source-grounded objects satisfying intent), agent reasoning mode (build an admissibility-verified reasoning chain), and answer synthesis. Human search mode is the simplest to validate.

## What This Does Not Give You

This is an architecture, not a drop-in library. There is no package to install, no SDK to import, and no promise that it "just works." You implement the anchors, the discovery object, the traversal step, alias resolution, and the governance checks yourself, and the

quality of your semantic content descriptors and intent modeling will determine whether meaning-based matching actually performs well for your corpus.

The approach is disclosed in a patent filing. It has not been presented here as a benchmarked or production-proven product, and this guide states no throughput, latency, recall, or accuracy numbers, because the specification does not. Do not expect published performance figures from this document.

It is also not always the right tool. If you have a small corpus, no per-item access control, and no need for an auditable trail of how a result was reached, a single vector index is simpler and this architecture is overkill. The design earns its cost where per-anchor governance, requester-scoped visibility, live navigational addressing, and lineage provenance matter. It complements rather than forbids embeddings: nothing here prevents an anchor from using vector similarity internally to compute its neighborhood, and the architectural claim is about where meaning, governance, and lineage live, not about banning nearest-neighbor math.

## **Disclosure Scope**

The approach described in this guide, semantic resolution and discovery of resources by meaning and lineage through scoped anchors rather than a central index or proprietary vector database, is disclosed in United States Patent Application 19/647,395. This guide is educational: it explains an architecture a skilled developer can build, and every mechanism described above is traced to that filing. It is not a warranty, not a specification of a released product, and not an offer of software. You are responsible for your own implementation, testing, and compliance.

Search, inference, and execution as one governed step.

[Chapter 10 \(/patents/19-647395/chapters/discovery\)](/patents/19-647395/chapters/discovery)

## **PRIMARY TECHNICAL DISCLOSURE**

- [Governed Semantic Discovery: Search, Inference, and Execution Through Adaptive Traversal \(/articles/governed-semantic-discovery-search-inference-and-execution-through-adaptive-traversal\)](/articles/governed-semantic-discovery-search-inference-and-execution-through-adaptive-traversal)

## **SECONDARY TECHNICAL**

- [The Adaptive Index as Unified Search-Inference-Execution Substrate \(/articles/semantic-discovery/unified-substrate\)](/articles/semantic-discovery/unified-substrate)
- [Three-in-One Traversal: Search, Inference, and Execution in a Single Bounded Step \(/articles/semantic-discovery/three-in-one-traversal\)](/articles/semantic-discovery/three-in-one-traversal)
- [The Discovery Object: A Traversal-Native Semantic Agent \(/articles/semantic-discovery/discovery-object\)](/articles/semantic-discovery/discovery-object)
- [Post-PageRank Semantic Ranking: Relevance Through Governed Traversal \(/articles/semantic-discovery/post-pagerank\)](/articles/semantic-discovery/post-pagerank)
- [Persistent Semantic State: Eliminating Prompt Reconstruction \(/articles/semantic-discovery/persistent-state\)](/articles/semantic-discovery/persistent-state)
- [Traversal Lineage as Index Evolution Signal \(/articles/semantic-discovery/traversal-lineage\)](/articles/semantic-discovery/traversal-lineage)
- [Anchor Semantic Neighborhood Publication \(/articles/semantic-discovery/semantic-neighborhoods\)](/articles/semantic-discovery/semantic-neighborhoods)
- [Inference-Time Execution Control as Traversal Primitive \(/articles/semantic-discovery/inference-governance\)](/articles/semantic-discovery/inference-governance)
- [Anchor Self-Organization Under Entropy and Load Pressure \(/articles/semantic-discovery/anchor-self-organization\)](/articles/semantic-discovery/anchor-self-organization)
- [Alias Resolution as Navigational Traversal \(/articles/semantic-discovery/alias-resolution\)](/articles/semantic-discovery/alias-resolution)
- [Three Discovery Operating Modes: Human Search, Agent Reasoning, Answer Synthesis \(/articles/semantic-discovery/operating-modes\)](/articles/semantic-discovery/operating-modes)
- [Model-Agnostic Semantic Discovery \(/articles/semantic-discovery/model-agnostic\)](/articles/semantic-discovery/model-agnostic)
- [Affect-Modulated Discovery Traversal \(/articles/semantic-discovery/affect-modulated-traversal\)](/articles/semantic-discovery/affect-modulated-traversal)
- [Confidence-Gated Discovery Traversal \(/articles/semantic-discovery/confidence-gated-traversal\)](/articles/semantic-discovery/confidence-gated-traversal)
- [Integrity-Tracked Traversal Drift Detection \(/articles/semantic-discovery/integrity-tracked-drift\)](/articles/semantic-discovery/integrity-tracked-drift)
- [Biological Identity-Scoped Access During Discovery \(/articles/semantic-discovery/biological-access\)](/articles/semantic-discovery/biological-access)

- [Rights-Grade Anchor Governance for Content Discovery \(/articles/semantic-discovery/rights-grade-anchors\)](/articles/semantic-discovery/rights-grade-anchors).
- [Forecasting-Shaped Discovery Traversal \(/articles/semantic-discovery/forecasting-shaped\)](/articles/semantic-discovery/forecasting-shaped).
- [Capability-Constrained Anchor Accessibility \(/articles/semantic-discovery/capability-constrained\)](/articles/semantic-discovery/capability-constrained).
- [Collaborative Multi-Object Discovery Traversal \(/articles/semantic-discovery/collaborative-traversal\)](/articles/semantic-discovery/collaborative-traversal).
- [Credentialed Reader Activation in the Discovery Substrate \(/articles/semantic-discovery/credentialed-reader-activation\)](/articles/semantic-discovery/credentialed-reader-activation).
- [Personal Cognitive Asset: How Per-User Lineage Re-Weights the Same Substrate \(/articles/semantic-discovery/personal-lineage-layer\)](/articles/semantic-discovery/personal-lineage-layer).

## **APPLICATIONS · GENERAL**

- [Enterprise Knowledge Management Through Governed Traversal \(/articles/semantic-discovery/enterprise-knowledge-management\)](/articles/semantic-discovery/enterprise-knowledge-management).
- [AI-Native Search That Replaces PageRank With Governed Contextual Relevance \(/articles/semantic-discovery/ai-native-search\)](/articles/semantic-discovery/ai-native-search).
- [Semantic Discovery for Scientific Research \(/articles/semantic-discovery/scientific-research-discovery\)](/articles/semantic-discovery/scientific-research-discovery).
- [Verifiable AI Legal Case Research: Governed Case Law Search with Citation Lineage \(/articles/semantic-discovery/legal-case-research\)](/articles/semantic-discovery/legal-case-research).
- [Patent Landscape Analysis Across Classification Boundaries: Governed Semantic Discovery for Prior Art and Freedom-to-Operate \(/articles/semantic-discovery/patent-landscape-analysis\)](/articles/semantic-discovery/patent-landscape-analysis).
- [Governed Medical Literature Search: Evidence-Grade Traversal for Clinical Questions \(/articles/semantic-discovery/medical-literature-search\)](/articles/semantic-discovery/medical-literature-search).
- [Governed Competitive Intelligence Search Across Patents, Filings, and Hiring Signals \(/articles/semantic-discovery/competitive-intelligence\)](/articles/semantic-discovery/competitive-intelligence).
- [Governed Semantic Search for Regulatory Compliance: Defensible Requirement Discovery Across Fragmented Corpora \(/articles/semantic-discovery/regulatory-compliance-search\)](/articles/semantic-discovery/regulatory-compliance-search).
- [Cross-Vendor Credentialed Sensor Coordination for Multi-Sensor Perception \(/articles/semantic-discovery/coordinated-perception\)](/articles/semantic-discovery/coordinated-perception).
- [Anti-Stalking Architecture for Cross-Vendor Bluetooth Tracker Networks \(/articles/semantic-discovery/post-airtag-tracking\)](/articles/semantic-discovery/post-airtag-tracking).
- [How Small Language Models Beat Large Ones: Use the World as Memory \(/articles/semantic-discovery/world-as-memory\)](/articles/semantic-discovery/world-as-memory).

## APPLICATIONS · SPECIFIC

- [Google Search Alternative: Governed Semantic Discovery Beyond Retrieval \(/articles/semantic-discovery/google-search\)](#)
- [Perplexity Alternative: Governed Semantic Discovery Beyond the Answer Engine \(/articles/semantic-discovery/perplexity\)](#)
- [Elasticsearch Alternative: Governed Semantic Discovery Beyond Index-Centric Search \(/articles/semantic-discovery/elasticsearch\)](#)
- [Algolia Alternative: Governed Semantic Discovery Beyond Per-Query Search \(/articles/semantic-discovery/algolia\)](#)
- [Pinecone Alternative for Governed Semantic Discovery \(/articles/semantic-discovery/pinecone\)](#)
- [Weaviate Alternative: Governed Semantic Discovery Beyond the Vector Database \(/articles/semantic-discovery/weaviate\)](#)
- [You.com Alternative: Governed Semantic Discovery Beyond AI Search \(/articles/semantic-discovery/you-com\)](#)
- [Brave Search Alternative: Governed Semantic Discovery Beyond an Independent Index \(/articles/semantic-discovery/brave-search\)](#)
- [Kagi Alternative for Governed Semantic Discovery Beyond Query-Response Search \(/articles/semantic-discovery/kagi\)](#)
- [Exa \(Metaphor Systems\) Alternative: Governed Semantic Discovery Beyond Neural Link Prediction \(/articles/semantic-discovery/metaphor-systems\)](#)
- [Glean Alternative for Governed Enterprise Discovery: Beyond Stateless Workplace Search \(/articles/semantic-discovery/glean\)](#)
- [Coveo Alternative: Governed Semantic Discovery Beyond Personalized Ranking \(/articles/semantic-discovery/coveo\)](#)
- [Apple Find My vs Governed Semantic Discovery: Two Meanings of Multi-Authority \(/articles/semantic-discovery/apple-find-my\)](#)
- [Google Find My Device Alternative: Governed Cross-Network Reader Activation \(/articles/semantic-discovery/google-find-my\)](#)
- [Governed Discovery Beyond IETF DULT: Behavior Standards vs Architectural Governance \(/articles/semantic-discovery/ietf-dult\)](#)
- [Glean Alternative: Governed Semantic Traversal Beyond Enterprise Search \(/articles/semantic-discovery/glean-enterprise-search\)](#)
- [Microsoft GraphRAG vs Governed Traversal: Adding Per-Step Governance to Graph Retrieval \(/articles/semantic-discovery/microsoft-graphrag\)](#)
- [Governed Agent Memory Beyond Mem0, Zep, and Letta \(/articles/semantic-discovery/memory-for-agents\)](#)

---

Semantic Discovery overview → (/semantic-discovery)