

# How to Build AI-Native Semantic Discovery Without a Central Index

You want agents and services to find each other by meaning and provenance, not by looking up rows in one central registry that every deployment has to trust and synchronize against. This guide walks through an architecture that does discovery as a governed walk over a self-describing index, with no global lookup table. It describes an approach disclosed in United States Patent Application 19/647,395, not a shipping library, and its home is the Semantic Discovery inventive step.

---

## What You Are Building

You are building a discovery layer where an agent, a query, or a reasoning task finds the resource it needs by describing what it means, not by knowing where it lives. The thing it is looking for might be a knowledge node, a user identity, an execution agent, a service endpoint, or a dataset. You want three properties at once: resolution by meaning and lineage rather than by exact key; no single central index or registry that every participant must query and keep in sync; and a governance record for every step, so that how a result was reached is auditable rather than opaque.

This is the problem faced by anyone assembling a multi-agent system, a federated knowledge platform, or an AI search substrate where the corpus and the participants change constantly and no one party owns the global map. The approach below is

disclosed in United States Patent Application 19/647,395. It is an architecture you implement yourself, not a package you install.

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

The default is a central index or registry: one service that maps names or embeddings to locations, and that every participant queries. This works and is well understood. Its structural cost is that the lookup table is an external data structure that must be kept in synchronization with the underlying data. DNS, URL resolution, and database key lookups all share this shape: the address is resolved by consulting a table that is maintained separately from the thing it points at. When the underlying content moves, splits, or is re-scoped, the table and the data can disagree until reconciliation catches up.

The second common approach is retrieval-augmented generation: a separate search engine returns candidate documents and a language model synthesizes an answer from them. Here the index is a passive retrieval target. There is no persistent query entity that carries state across steps, no structured semantic state maintained between retrieval operations, and no governance evaluation of the intermediate retrieval or reasoning transitions. Provenance of the final answer is difficult to reconstruct because the search, ranking, and generation subsystems each hold their own state and hand off through serialized interfaces that drop semantic context at every boundary.

The third is multi-hop knowledge graph traversal, where each hop retrieves connected facts from a graph database. The graph is a data structure that is queried; a hop is a lookup that returns connected entities. Nothing at the hop evaluates whether the traversing entity is authorized to make that hop, whether it contradicts what the traversal has already committed to, or whether the transition is admissible under policy.

None of these is wrong. The gap they share is that the index is passive: an external process poses queries to it, and governance, if any, lives outside the traversal. The architecture below moves both the description and the governance into the index itself.

## **The Architecture**

The disclosed approach rests on three ideas: the index is an active substrate of governance-enabled anchors, the query is a persistent object that walks it, and every step is a single fused operation.

**Anchors, not a lookup table.** Every addressable object 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 also maintains its own description of what is reachable from it, called a neighborhood publication, and it maintains this itself rather than reporting to a central authority. The publication is not a static document list or precomputed inverted index. It comprises a semantic content descriptor (an abstracted description of the semantic territory the container covers, not an enumeration of objects), a reachability graph (the sub-anchors and peer anchors directly navigable from here, with the semantic relationship to each), a policy envelope (the governance constraints on traversing this container), a freshness indicator, and an entropy summary describing the semantic diversity and update frequency of the contents. Because each anchor computes its own publication from its own container state, updates propagate without coordination with any global index management service.

**The discovery object.** A query, search, or reasoning task is instantiated as a discovery object: a persistent, memory-resident, typed entity, not a query string, keyword list, embedding, or prompt. It carries an intent field (a structured objective with a goal type, domain scope, resolution criterion, and specificity constraints), a context block, a memory field (the accumulated semantic commitments of the traversal), a policy reference field, a lineage field (the ordered sequence of admitted

transitions), an affective state field, and a confidence field. It persists across every step, accumulating state at each anchor, and is structurally isomorphic to the platform's semantic agent schema, so the same governance and lineage machinery that applies to agents applies to it without modification.

**The three-in-one traversal step.** At each anchor, the discovery object undergoes one atomic step with three coupled phases, and no transition is possible without completing all three:

- *Search* is local, not global. The anchor evaluates the discovery object's current state against its published reachable neighborhood and produces a candidate transition set: the next transitions that are both structurally reachable and semantically relevant. It does not score the object against the whole corpus; it considers only what this anchor advertises. This bounds the cost of each step and narrows the space monotonically as the walk goes deeper.
- *Inference* scores or selects among candidates and updates the object's state (refined intent, extended memory, updated confidence). The inference engine is model-agnostic: it may be an embedding-similarity scorer, a rule-based matcher, a probabilistic model, a neural ranker, or a language model. It proposes; it does not commit.
- *Execution* decides admissibility against deterministic criteria: the policy constraints in the object's policy reference field and the anchor's governance configuration, lineage continuity, entropy bounds, and temporal validity. It returns admit, reject, or decompose, and it records the determination in lineage whether or not the transition is taken. Authority to commit resides only here, which is what lets an untrusted inference engine participate without compromising governance: the model proposes, the substrate decides.

**Resolution by walking, not by table.** Addresses are structured aliases of the form

`type@domain.subdomain/path`, for example `agent@net.qu3ry/nest/alpha`.

Resolving one is navigational: it begins at the domain anchor and proceeds one path segment at a time, each anchor routing the request to the appropriate sub-anchor, until

the terminal segment identifies the object. There is no lookup table; the address is the path through the index and resolution is the act of walking it, so the result always reflects the live index. Alias resolution runs under the same governance as any traversal: possessing the full alias string does not bypass the per-anchor authorization checks, so alias knowledge does not confer access.

**Anchor resolution with three outcomes.** When a transition references an external concept, entity, or fact not already established in the object's memory, that reference is not committed directly. It is submitted to an anchor resolution step that attempts to bind it against available semantic infrastructure, and that produces one of three outcomes: *resolved* (a verified referent is found and incorporated), *unresolvable* (no verified referent exists, so the mutation is rejected to keep ungrounded content out), or *ambiguous* (multiple candidate referents exist and the step cannot deterministically choose, so the mutation may be decomposed into alternatives, each evaluated independently). This is the mechanism that keeps discovery grounded rather than confabulated.

**Self-organization under load.** Anchors monitor their own telemetry (mutation throughput, resolution request rate, entropy load, trust slope signals) and reshape the index without external coordination: splitting a high-entropy container into sub-containers, merging under-used sibling containers, migrating a container nearer to where it is actually accessed, and rekeying aliases when structure changes. Every such operation runs under deterministic policy, is recorded in lineage, and is transparent to active traversals; rekeyed aliases resolve through a policy-bounded redirect chain so references survive structural change.

## How to Approach the Build

The following is an ordered path a developer can take to implement this architecture. The sketches are illustrative and faithful to the disclosure, not runnable code.

1. **Model the anchor and its container.** Give each anchor a mutation policy, an alias mapping, lineage metadata, and a governance configuration. Decide your container hierarchy semantics before anything else, because everything downstream navigates it.
2. **Implement the neighborhood publication as a computed artifact.** Have each anchor derive its publication from its own container state and recompute on mutation events (objects added or removed, sub-anchors split or merged, policy changes). Include all five parts. Make it requester-scoped: the same anchor returns a narrower publication to a discovery object whose policy profile authorizes less.

```
publication(anchor, requester) = {  
  content_descriptor, # abstracted semantic territory, not an object list  
  reachability_graph, # navigable sub/peer anchors + relationships  
  policy_envelope, # constraints, filtered to requester  
  freshness, # epoch of last recompute  
  entropy_summary # diversity / update frequency / density  
} # illustrative only
```

3. **Define the discovery object schema** with the seven typed fields (intent, context, memory, policy, lineage, affect, confidence). Populate intent and policy at initialization from the originating query and the user's governance profile. Treat memory and lineage as append-through-transitions, not free text.
4. **Build the three-in-one step as one indivisible transition.** Do not implement search, inference, and execution as three services with interfaces between them; the whole point is to remove those boundaries. Search must read only the local publication. Inference must be swappable behind a proposal interface. Execution must be deterministic: given the same object state, anchor configuration, and proposed transition, it returns the same admit/reject/decompose outcome, and it writes to lineage every time.

5. **Implement navigational alias resolution** as a special case of traversal that steps the `type@domain/path` segments, enforcing governance at each hop. Do not add a shortcut lookup table; that would reintroduce the synchronization problem you are avoiding.
6. **Add anchor resolution for external references** with the resolved / unresolvable / ambiguous trichotomy, so ungrounded references are rejected and ambiguous ones decompose rather than silently resolving to a guess.
7. **Add self-organization last**, driven by anchor telemetry and gated by deterministic policy, with lineage recording and redirect chains so it is transparent to in-flight traversals.

Keep proposal authority (inference) and commitment authority (execution) strictly separated throughout. That separation is the load-bearing property.

## What This Does Not Give You

This is an architecture, not a drop-in library. There is no package to install and nothing here "just works" out of the box; you implement the anchors, the publications, the discovery object, the traversal step, and the governance evaluators yourself, and you make the design choices the disclosure leaves open (clustering criteria for splits, thresholds, the specific inference engine). The method is disclosed in a patent filing. It is not presented here as a benchmarked or production-proven system, and this guide states no performance numbers, because the filing does not turn on any.

The disclosure describes admissibility evaluation as operating on typed fields rather than unstructured content, and characterizes its per-step overhead as bounded relative to the number of governance constraints rather than the size of the index; that is a described property of the design, and you should measure your own implementation rather than assume it. The approach is a fit where discovery must be by meaning and

lineage, where no party owns a global map, and where per-step provenance matters. It is heavier than a plain key-value registry, and if you genuinely have a small, stable, centrally-owned catalog, a central index may be the simpler and correct choice.

## Disclosure Scope

The architecture described in this guide is disclosed in United States Patent Application 19/647,395. This guide is educational: it explains an architectural approach to semantic discovery so that a skilled developer can understand and build it themselves. It is not a warranty, a specification of any product, or an offer of software, and nothing here should be read as a claim that a downloadable implementation exists. Every mechanism described above is drawn from that filing; design decisions the filing leaves open are yours to make.

---

## **Semantic Discovery** (</semantic-discovery>)

[All 40 steps → \(/inventive-steps\)](/inventive-steps)

Search, inference, and execution as one governed step.

### **Chapter 10** (</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 \(/articles/semantic-discovery/credential-reader-activation\)](/articles/semantic-discovery/credential-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 \(/articles/semantic-discovery/google-search\)](/articles/semantic-discovery/google-search).
- [Perplexity Alternative: Governed Semantic Discovery Beyond the Answer Engine \(/articles/semantic-discovery/perplexity\)](/articles/semantic-discovery/perplexity).
- [Elasticsearch Alternative: Governed Semantic Discovery Beyond Index-Centric Search \(/articles/semantic-discovery/elasticsearch\)](/articles/semantic-discovery/elasticsearch)
- [Algolia Alternative: Governed Semantic Discovery Beyond Per-Query Search \(/articles/semantic-discovery/algolia\)](/articles/semantic-discovery/algolia).
- [Pinecone Alternative for Governed Semantic Discovery \(/articles/semantic-discovery/pinecone\)](/articles/semantic-discovery/pinecone).
- [Weaviate Alternative: Governed Semantic Discovery Beyond the Vector Database \(/articles/semantic-discovery/weaviate\)](/articles/semantic-discovery/weaviate).
- [You.com Alternative: Governed Semantic Discovery Beyond AI Search \(/articles/semantic-discovery/you-com\)](/articles/semantic-discovery/you-com)
- [Brave Search Alternative: Governed Semantic Discovery Beyond an Independent Index \(/articles/semantic-discovery/brave-search\)](/articles/semantic-discovery/brave-search).
- [Kagi Alternative for Governed Semantic Discovery Beyond Query-Response Search \(/articles/semantic-discovery/kagi\)](/articles/semantic-discovery/kagi)

- [Exa \(Metaphor Systems\) Alternative: Governed Semantic Discovery Beyond Neural Link Prediction \(/articles/semantic-discovery/metaphor-systems\)](/articles/semantic-discovery/metaphor-systems)
- [Glean Alternative for Governed Enterprise Discovery: Beyond Stateless Workplace Search \(/articles/semantic-discovery/glean\)](/articles/semantic-discovery/glean)
- [Coveo Alternative: Governed Semantic Discovery Beyond Personalized Ranking \(/articles/semantic-discovery/coveo\)](/articles/semantic-discovery/coveo)
- [Apple Find My vs Governed Semantic Discovery: Two Meanings of Multi-Authority \(/articles/semantic-discovery/apple-find-my\)](/articles/semantic-discovery/apple-find-my)
- [Google Find My Device Alternative: Governed Cross-Network Reader Activation \(/articles/semantic-discovery/google-find-my\)](/articles/semantic-discovery/google-find-my)
- [Governed Discovery Beyond IETF DULT: Behavior Standards vs Architectural Governance \(/articles/semantic-discovery/ietf-dult\)](/articles/semantic-discovery/ietf-dult)
- [Glean Alternative: Governed Semantic Traversal Beyond Enterprise Search \(/articles/semantic-discovery/glean-enterprise-search\)](/articles/semantic-discovery/glean-enterprise-search)
- [Microsoft GraphRAG vs Governed Traversal: Adding Per-Step Governance to Graph Retrieval \(/articles/semantic-discovery/microsoft-graphrag\)](/articles/semantic-discovery/microsoft-graphrag)
- [Governed Agent Memory Beyond Mem0, Zep, and Letta \(/articles/semantic-discovery/memory-for-agents\)](/articles/semantic-discovery/memory-for-agents)

---

[Semantic Discovery overview → \(/semantic-discovery\)](/semantic-discovery)