Glean Indexes Enterprise Knowledge Without Governing Its Discovery

by Nick Clark | Published March 28, 2026 | PDF

Glean connects to dozens of enterprise applications and builds a unified search index across an organization's Slack messages, Google Drive documents, Confluence pages, Jira tickets, and more. The platform makes enterprise knowledge findable from a single search box. But indexing content and governing how it is discovered are structurally different operations. Each query retrieves relevant documents without maintaining a persistent discovery process that accumulates understanding of the organization's knowledge landscape. The gap is between finding content and governing discovery.

What Glean built

Glean ingests content from enterprise applications through connectors, respects access permissions from the source systems, and provides search that spans the entire organization's content landscape. The system understands organizational context: who created content, when it was last modified, who

interacts with it, and how it relates to other content. AI-generated answers synthesize information from multiple sources into coherent responses.

The unification is valuable. Before enterprise search platforms, knowledge was trapped in application silos. Finding the right Slack thread, the relevant Confluence page, and the associated Jira ticket required searching three different systems. Glean makes all of it searchable from one place. But the search interaction is still query-response. Each search is independent. The system does not maintain a governed discovery process that tracks what an employee has explored, what knowledge gaps remain, or how findings across queries relate to each other.

The gap between unified indexing and governed discovery

Unified indexing solves the findability problem: content across enterprise tools is searchable from one place. Governed discovery solves the understanding problem: how does an employee navigate from not knowing to knowing within the organization's knowledge landscape? These are different problems. Finding a document is a retrieval task. Building understanding of a domain within organizational knowledge is a traversal task that requires accumulated state.

A new employee onboarding to a project needs governed discovery, not just better search. They need a process that tracks what they have read, identifies what they have not yet encountered, detects contradictions between documents, and directs them toward the knowledge gaps that remain. Glean can help them find any individual document. It cannot govern the process through which they build comprehensive understanding.

The organizational knowledge problem is particularly acute because enterprise knowledge is distributed, contradictory, and evolving. A policy documented in Confluence may have been superseded by a Slack discussion that was never formalized. Governed discovery detects this: the discovery object encounters both the policy and the discussion, identifies the contradiction, and flags it for resolution. Stateless retrieval returns whichever document ranks higher for the query, with no awareness of the contradiction.

What governed semantic discovery enables for enterprise knowledge

With persistent discovery objects, Glean's unified index becomes navigable rather than merely searchable. An employee conducting a discovery process across organizational knowledge maintains a discovery object that tracks which documents have been visited, what relationships have been identified, and what contradictions are pending. The discovery object governs the traversal strategy: directing the employee toward unvisited areas of the knowledge landscape rather than returning them to content they have already absorbed.

The three-in-one traversal model integrates search, inference, and execution within each discovery step. A step that retrieves a document, infers its relationship to prior findings, and updates the traversal strategy functions as one governed operation. The enterprise context enriches this: organizational relationships, access patterns, and content provenance inform the inference step.

Traversal lineage provides organizational value beyond individual discovery. When an employee completes a governed discovery process, the traversal lineage represents a documented path through organizational knowledge. This lineage can be shared with colleagues, used for onboarding, or referenced during audits. The discovery process itself becomes a reusable organizational artifact.

The structural requirement

Glean solved unified enterprise content indexing. The structural gap is between finding content across enterprise tools and governing the discovery process through organizational knowledge. Semantic discovery provides persistent discovery objects that accumulate understanding, governed traversal that directs exploration toward knowledge gaps, and traversal lineage that makes the discovery process itself a shareable organizational asset.

Semantic Discovery All 21 steps →

Search, inference, and execution as one governed step.

AQ
deterministic
autonomy

Legal

Subject to one or more pending U.S. and international patent applications, see [Patents](#) for the current list and status. No license, express or implied, is granted. Any use requires a separate written agreement—see [Licensing](#). Patent applications referenced on this site are pending. Claim scope, if any, is subject to examination and may issue in altered form or not at all. See [Legal](#) for terms and conditions.

Adaptive Query™ is a trademark of Nicholas Clark. U.S. federal registration is pending. federal registration. AQ™ , AQ Inside™ , Adaptive Index™ , Adaptive Network™ , Semantic Agent™ , @AQ™ , AQID™ , and Adaptive Coin™ are used as trademarks in connection with the Adaptive Query platform and brand. Other names may be trademarks of their respective owners.

Last updated: 2026-03-03

-
- [Inventive Steps](#)
- [Licensing](#)
- [Patents](#)
- [Articles](#)
- [Legal](#)
- [Opportunities](#)
- [Sitemap](#)

-
- nick@qu3ry.net
- 72 28 14 36 01

[Invented by Nick Clark](#) | Founding Investors: Devin Wilkie