ZL Technologies, Inc.

2026

# AI PLAYBOOK

## Preparing Enterprise Data for AI Production

*www.zlti.com*

# Contents

# About This Playbook

This playbook is a practical guide for preparing enterprise data for AI. It walks through the four milestones that determine whether an AI program reaches production, from establishing what data you have to keep it governed over time. It assumes you have defined an AI use case and are now asking the operational question: is our data ready, and if not, what must happen before a model touches it?

Each milestone builds on the one before, and the AI method decisions covered later depend on the earlier steps being complete. The outcome is a data foundation that is discoverable, classified, governed, and formatted for your chosen AI method, which makes the difference between a pilot that fails and a program that scales.

# Problem Statement

According to Stanford's latest research, only one-third of organizations have successfully scaled their AI programs to the enterprise level, and the data points to why. Data quality and preparation are tied with regulatory friction as the leading barriers to AI production. The organizations realizing the most value from AI are those treating their data as a competitive moat: 75% of successful implementations cite proprietary data as a key factor in their AI strategy. Generalized large language models, trained on broad public data, cannot surface organizational knowledge. Moving from pilots to production requires grounding AI in the enterprise's own data, or “corporate memory.”

The challenge is that enterprise data is vast, siloed, and ungoverned. Petabytes of unstructured content sits across file shares, emails, collaboration platforms, and legacy repositories. Much of it contains sensitive information that should never reach an AI model. PII, privileged communications, data under litigation hold, and regulated content require identification, classification, and remediation before data can responsibly feed into an AI pipeline. Enterprises are largely assembling AI datasets through slow, resource-intensive manual processes. ZL Tech provides the infrastructure to systematically and precisely locate relevant content across the full data estate and enforce governance controls to ensure that what reaches the model is both accurate and compliant.

## The Data Readiness Gap: **What "AI-Ready" Actually Means**

Start

"AI-ready" is a specific and demanding standard: data that is discoverable across the full enterprise, classified by sensitivity and lifecycle status, governed by consistent policies, and supported by data infrastructure. Falling short on any one of these criteria produces failures that compound through the pipeline.

ZL Tech functions as both the diagnostic layer, establishing what data exists and what condition it is in, and the execution layer that brings enterprise data to AI-ready status at scale. AI-ready data satisfies four criteria:

**Findable**
Unified and accessible across the full data estate without manual inventory or migration.

**Classified**
Evaluated for sensitivity, relevance, and lifecycle status.

**Governed**
Subject to consistent retention, access, and compliance policies across all sources.

**Supported**
Supported by data infrastructure and formatted for the intended AI method, whether RAG, fine-tuning, or full training.

## Generative AI vs. Agentic AI: **What Changes for Data Preparation**

The way an AI system uses data determines how that data must be prepared, and agentic AI raises the stakes sharply. A generative system with bad data gives a bad answer. An agentic system with bad data takes bad actions, and each step compounds the last. Understanding this difference is the starting point for any data preparation strategy.

**Generative AI** systems are reactive. They receive a prompt, produce an output, and stop. Data quality matters because poor input degrades output quality. But the consequences of a bad answer are bounded: a user receives inaccurate output and can correct course. Data preparation for generative AI centers on retrieval quality and context relevance.

**Agentic AI** systems are autonomous. They plan, chain decisions across multiple steps, and execute actions in enterprise systems, often without human review at each step. The data quality stakes are categorically higher. A flawed document in an agentic workflow produces bad autonomous actions rather than a bad answer, and each subsequent step builds on the last.

| | Generative AI | Agentic AI |
|---|---|---|
| **Primary Function** | Generates content, insights, and recommendations. | Makes decisions and executes actions across systems. |
| **Role of Data** | Shapes the quality of answers and recommendations. | Shapes the quality of decisions, actions, and outcomes. |
| **Consequence of Bad Data** | Inaccurate, incomplete, or misleading outputs that users can typically evaluate and correct. | Incorrect decisions and actions that can propagate automatically through workflows and systems. |
| **Risk Profile** | Sensitive data may be surfaced in responses. | Sensitive data may be surfaced, accessed, or acted upon autonomously. |
| **Data Readiness Requirement** | Focus on relevance, retrieval quality, and contextual accuracy. | Focus on trusted, governed, current, and policy-compliant data before deployment. |

Recent studies have found that AI models given ungoverned access to sensitive enterprise data independently engaged in harmful behavior, including unauthorized disclosure of personal information and corporate espionage.* The conclusion from this research: **agentic misalignment is a data governance problem before it is a model problem.**

** Anthropic (2025); multi-institution red-teaming study (Harvard, MIT, Stanford, 2026)*

## The Four Milestones

This is the core of the playbook. The four milestones are the road from raw enterprise data to a model-ready foundation, and they are non-negotiable regardless of AI method, deployment environment, use case, or model provider. MIT found that 95% of enterprise AI projects fail to scale beyond pilot, and the root cause is usually the same: the data foundation was not established before the model was deployed. These four steps are what prevent that failure. Each has a distinct purpose, and later decisions in this playbook assume they are in place.

- **Milestone 1: Data Discovery**
  Establish a complete, current inventory of what data exists and where it lives across the full enterprise before any dataset construction begins.
- **Milestone 2: Classification and Sensitivity Management**
  Evaluate every document for sensitivity, lifecycle status, and governance requirements. Determine what is included, what is excluded, and what requires redaction or controlled handling.
- **Milestone 3: Dataset Construction**
  Build the AI-ready dataset from the governed corpus. Automate what enterprises are currently assembling by hand, with the filtering and formatting appropriate to the selected AI method.
- **Milestone 4: Ongoing Governance**
  Treat AI readiness as a continuous operational function. As data changes and regulations evolve, the governance layer must keep pace.

## MILESTONE 1:
# Data Discovery

**Why it matters:** you cannot govern, classify, or build a dataset from data you cannot see. Discovery is the precondition for everything that follows.

Enterprises now store petabytes of data, spread across an average of 367 applications according to Forrester. Gartner predicts that large enterprises will triple their unstructured data capacity between 2023 and 2028. Most cannot search or account for it from a single view, which is the problem discovery solves.

Before any dataset can be built, the enterprise must establish a complete and current inventory of what data it has and where it lives. Organizations often cannot answer this question with confidence. Data accumulates over years across file shares, email archives, collaboration platforms, and legacy repositories, with no unified view across sources and no reliable accounting of what the estate contains.

ZL's in-place data management resolves this without requiring migration or duplication. The platform builds a continuously updated master index across all data sources by extracting the “essence” of every document, including its metadata and full-text content. Original files remain at their source locations. The result is a unified, governed view of the full data estate, searchable and classifiable from a single platform, at approximately 10% of the original storage footprint.

- ✓ **Map the full data estate:** file shares, email (Exchange/M365), collaboration platforms (Teams, Slack), legacy repositories, and all additional unstructured content environments.
- ✓ ZL builds a **continuously updated master index** across all sources with no migration, no duplication, and no manual inventory.
- ✓ **Document essence** (metadata plus full-text content) extracted and indexed at approximately 10% of original storage footprint; original files remain in place.
- ✓ For agentic deployments: resolves the **speed-to-data bottleneck**. ZL's master index delivers data up to 1000x faster than native platform extraction from systems like M365, which throttle bulk access and were designed for user productivity, not AI pipeline volume.

## MILESTONE 2:
# Classification and Sensitivity Management

**Why it matters:** classification establishes what is safe to use, what needs controlled handling, and what must never reach a model. In regulated industries and any agentic deployment, it determines whether the program is deployable at scale.

Classification is the step that separates a governed AI program from a liability. Every indexed document is evaluated on three questions: what is it, how sensitive is it, and where is it in its lifecycle. The answers sort content into three outcomes. Some data is cleared for use as-is. Some is included but requires controlled handling, such as field-level redaction of sensitive passages. Some must be excluded entirely, either because it is too sensitive to expose or because its lifecycle status (under litigation hold, past retention, superseded) makes it unfit for a model. The goal is precision: apply the right control to the right content so that sensitive material is protected without discarding documents that still hold value for the AI program.

Getting classification right depends on a few factors:

- **Regulatory context sets the rules.** A financial-services deployment, a healthcare deployment, and a government deployment classify the same document differently. Controls should be configured to the regulatory environment the data lives in, not applied uniformly.
- **Lifecycle status is as important as content.** A document can be perfectly safe on content and still be disqualified by its status, litigation hold being the clearest example. Classification has to read both.
- **Redaction preserves value.** Excluding an entire document because it contains a single sensitive field wastes usable content. Field-level redaction keeps the document in scope while removing what cannot be exposed.
- **The classification decision propagates downstream.** Whatever is tagged here determines what Milestone 3 can build from and what the RAG index or training set is allowed to contain. Errors here surface everywhere later.

ZL's classification capabilities cover the full range of enterprise sensitivity requirements. For documents that remain in scope but contain sensitive content, ZL's redaction capabilities identify and handle PII and other sensitive material at the field level. These workflows are configurable to each deployment's regulatory environment and data profile, ensuring that the right controls are applied to the right content without excluding documents that retain value for the AI program.

- ✓ **Sensitivity labeling and classification**
  Systematic tagging of all indexed content by type, sensitivity level, and governance status; determines inclusion, exclusion, or controlled handling.
- ✓ **Redaction**
  PII and sensitive content identified and handled within documents at the field level; configurable by deployment environment and regulatory requirements.
- ✓ **Categories requiring governance action before data reaches a model**
  - Personally identifiable information (PII)
  - Privileged legal communications
  - Data under litigation hold
  - Regulated records (financial, healthcare, government)
  - Executive communications
  - Proprietary IP and trade secrets
  - Content subject to third-party confidentiality obligations

# The AI Method Decision

With your data discovered, classified, and governed through Milestones 1 and 2, the next decision is how the model will use it. This should precede dataset construction, because each method imposes different requirements on content selection, formatting, and volume. Two choices sit here, and they answer different questions: RAG versus fine-tuning is about how the model gets its knowledge, and LLM versus SLM is about which model runs the task. The decision variables below map your use case to the right path, and the pipeline sections that follow (Path A and Path B) show how to execute each one. Most enterprises land on RAG for dynamic knowledge and reserve fine-tuning for stable, domain-specific behavior.

## Key Decision Variables

**How frequently does the underlying data change?**

- Frequently changing data (current policies, recent communications, active contracts) favors RAG
- Stable, well-defined knowledge favors fine-tuning

**Is the goal precision retrieval or shaping model behavior?**

- Surfacing specific answers from enterprise content: RAG
- Adapting a model to reason, write, or respond in a domain-specific way: fine-tuning

**What is the data volume and sensitivity profile?**

- High-sensitivity environments require tighter curation regardless of method

**Broad enterprise deployment or domain-specific agent function?**

- Domain-specific agents (compliance, financial analysis, customer intelligence) typically perform best on targeted, curated datasets rather than a general enterprise corpus

| | RAG | Fine-Tuning |
|---|---|---|
| **How it works** | Retrieves relevant content from a governed index at query time | Adjusts model weights using a curated training dataset |
| **Best for** | Dynamic, high-volume, or frequently updated enterprise knowledge | Stable, domain-specific behavior, tone, or reasoning patterns |
| **Data Change Rate** | Handles frequently changing data; retrieves the latest content live | Suited to stable knowledge; updating requires retraining |
| **Primary Goal** | Precision retrieval of specific enterprise content | Shaping how the model reasons, writes, or responds |
| **ZL's Role** | The governed ZL index serves directly as the retrieval corpus | Curates targeted training sets from the classified corpus |
| **Updating** | Refresh the index as data changes | Periodic retraining on a refreshed dataset |

# Path A: RAG (Retrieval-Augmented Generation)

RAG is the most common enterprise AI method for unstructured data. Rather than encoding knowledge into model weights, RAG connects a model to a governed retrieval index at query time, returning relevant content alongside the question for the model to reason over. RAG output quality is determined by what gets retrieved, which depends on what content was included in the corpus, how it was chunked, and how accurately it was classified and governed before reaching the pipeline.

ZL provides the data foundation that makes enterprise RAG tractable at scale. The ingestion, parsing, indexing, access control preservation, retention and legal hold logic, and unified metadata layer across email, files, and collaboration platforms represent the most time-consuming and technically complex part of a RAG build. Organizations constructing this infrastructure from scratch typically spend six to twelve months before retrieval is operational. ZL customers enter the RAG pipeline with this foundation already in place.

## RAG Pipeline Steps

1. **Use case and data scoping.** Define the question domain precisely: HR policy Q&A, contract review, internal knowledge search, regulatory response. Scope drives every downstream decision, from which data sources are included to which chunking strategy is appropriate.
2. **Source identification.** Map where relevant content lives. Enterprise RAG implementations typically span five to fifteen systems: SharePoint, M365, file shares, Slack and Teams, IMs, S3 or Azure Blob, and line-of-business applications. ZL's master index already covers this estate.
3. **Ingestion and parsing.** ZL connectors pull content and index it in-place. Parsers normalize native formats (PDF, DOCX, PPTX, EML, HTML, scanned images) into clean text and structured metadata. Layout-aware parsing handles tables, headers, and image-embedded text, which are common failure points in standard ingestion pipelines.
4. **Chunking.** Documents are split into retrievable pieces, typically 200 to 1,000 tokens with overlap. Chunking strategy affects retrieval quality more than almost any other pipeline variable. Fixed-size chunking is the simplest approach: uniform token splits with overlap. It is fast and easy to tune, but context-blind splits can sever related content across chunk boundaries. Semantic chunking detects shifts in meaning to group content into coherent ideas, improving retrieval precision on conceptual queries at the cost of higher computational overhead. Hierarchical (parent-child) chunking uses small chunks for retrieval precision and larger parent chunks for generation context, addressing the tension between finding the right passage and providing enough surrounding material for the model to reason over. Structure-aware chunking follows the document's native architecture — headings, sections, tables — as natural boundaries, which is particularly effective for contracts and policy documents where structure carries meaning, but depends on consistent source formatting to work well.
5. **Embedding.** Each chunk is converted to a vector representation via an embedding model.
6. **Vector storage.** Vectors are stored in a vector database alongside metadata: source, author, date, permissions, document type, and custom tags. Options include dedicated vector databases, hybrid stores, and platform-bundled solutions. ZL's partnership with Databricks enables governed ZL content to flow directly into Databricks Vector Search, allowing enterprises operating on the Databricks Data Intelligence Platform to layer RAG capabilities onto a governed content foundation without a separate data unification effort.
7. **Retrieval.** At query time, the question is embedded and matched against the vector store. Mature enterprise setups use hybrid retrieval: dense vector search combined with sparse keyword search, fused and passed through a re-ranker. Metadata filters are applied at this layer.
8. **Permissioning.** Retrieved chunks are filtered against the requesting user's access control lists from source systems. ZL's governance layer preserves access controls through the index so that permissions propagate cleanly into retrieval, a step where many RAG implementations fail.
9. **Generation.** Retrieved chunks and the user question are passed to the LLM with a structured prompt, citation requirements, and refusal logic for low-confidence retrieval. Model options include Claude, GPT, Gemini, or self-hosted Llama and Mistral variants.

10. **Evaluation and monitoring.** Retrieval quality scoring, hallucination flagging, freshness tracking, cost and latency monitoring, and user feedback loops. Evaluation requires ongoing investment as the underlying data evolves; it is the mechanism for maintaining retrieval quality in production.

# Path B: Fine-Tuning

B

Fine-tuning adjusts the weights of a foundation model using a curated training dataset, adapting the model's behavior, reasoning patterns, or domain-specific language to reflect the enterprise's accumulated knowledge. It is best suited to stable, well-defined knowledge domains where the goal is to shape how the model responds, reasons, or writes, rather than to retrieve specific current content.

ZL's role in fine-tuning is the curation of the training dataset from the governed, classified corpus. The classification and lifecycle management work completed in the Milestone preparation steps feeds directly into the training set. Training example quality is the binding constraint on fine-tuned model quality, regardless of dataset volume.

## Fine-Tuning Pipeline Steps

1. **Use case definition.** Identify the target behavior precisely: how should the model respond, reason, or write in this domain? Fine-tuning is most effective when the target behavior is specific and consistent, such as regulatory analysis, contract interpretation, or domain-specific report generation.
2. **Dataset curation.** ZL selects and packages relevant content from the governed corpus, filtered by domain, classification status, and lifecycle state. ROT data and out-of-scope content are excluded at this step. The classification work already completed in Milestone 2 drives this filtering directly.
3. **Data formatting.** Training examples are formatted as input-output pairs: instruction-response format or prompt-completion format, depending on the base model and fine-tuning approach. Formatting quality directly affects training effectiveness.
4. **Sensitivity review.** A final classification pass ensures no sensitive content enters the training set. PII, privileged communications, and other governed content identified in the classification step are excluded or redacted before the dataset is finalized.
5. **Base model selection.** Choose the foundation model to fine-tune: GPT-4o, Claude, Llama, Mistral, or domain-specific base models. Selection criteria include model size, licensing constraints, inference cost, and the complexity of the target behavior.
6. **Training run.** The fine-tuning job is executed via the model provider's API or self-hosted training infrastructure. Meaningful behavior change typically requires hundreds to thousands of curated, high-quality training examples.
7. **Evaluation.** The fine-tuned model is tested against a held-out evaluation set and benchmarked against the base model on domain-specific criteria that map directly to the target behavior defined in step one.
8. **Deployment and versioning.** The fine-tuned model is deployed to production. The training dataset is versioned and documented to support auditability and future retraining cycles.
9. **Retraining cadence.** As the enterprise's underlying knowledge evolves, ZL's continuously updated corpus enables periodic retraining without rebuilding the dataset from scratch. Classification and lifecycle management work is cumulative across training cycles.

# Model Selection: LLM vs. SLM

Many enterprise use cases are best served by a purpose-built small language model (SLM). For narrow, repetitive, high-accuracy tasks, a well-curated SLM frequently outperforms a general large language model (LLM) on the specific task it was designed for, at lower cost and lower latency. Model selection should follow from the use case, not from a default toward the largest available option.

## Recommended Architecture

- ✓ Modular deployment: SLMs handle routine, specialized functions; LLMs are reserved for complex reasoning and cross-domain tasks.
- ✓ ZL's targeted curation enables the modular approach by producing purpose-fit datasets for each agent function, rather than a single general corpus that all models draw from equally.
- ✓ The modular architecture improves both performance and cost efficiency at enterprise scale.

| | LLM | SLM |
|---|---|---|
| **Best For** | | |
| | Complex reasoning, multi-domain questions, open-ended generation | Narrow, repetitive, high-accuracy tasks (classification, extraction, domain Q&A) |
| **Cost & Latency** | | |
| | Higher infrastructure cost and higher latency | Lower cost and lower latency |
| **Capability** | | |
| | Broad, general-purpose across diverse task types | Focused and task-specific |
| **Fine-Tuning** | | |
| | Slower and more resource-intensive | Fast; can be tuned rapidly on ZL-curated domain datasets |
| **Examples** | | |
| | GPT-4o, Claude Opus, Gemini Ultra, Llama 3 70B+ | Phi-3, Mistral 7B, Llama 3 8B, domain-specific variants |

## MILESTONE 3:
# Building the Dataset

**Why it matters:** this is where enterprises often spend the most manual effort. Building from an already-governed corpus turns dataset construction into a filtering and packaging step rather than a from-scratch project.

**The payoff compounds:** McKinsey documented a financial-services enterprise whose reusable, governed unstructured data foundation could deliver $10 million to $20 million in cost avoidance as AI use cases multiplied, since each new application draws from the same prepared corpus instead of rebuilding the pipeline.

The AI-ready dataset is the direct output of the discovery and classification work completed in Milestones 1 and 2. It is also where enterprises often spend the bulk of their time and resources: manually assembling training corpora and retrieval indexes, building internal datasets by hand. ZL automates this work from the governed corpus that has already been indexed, classified, and cleared for use.

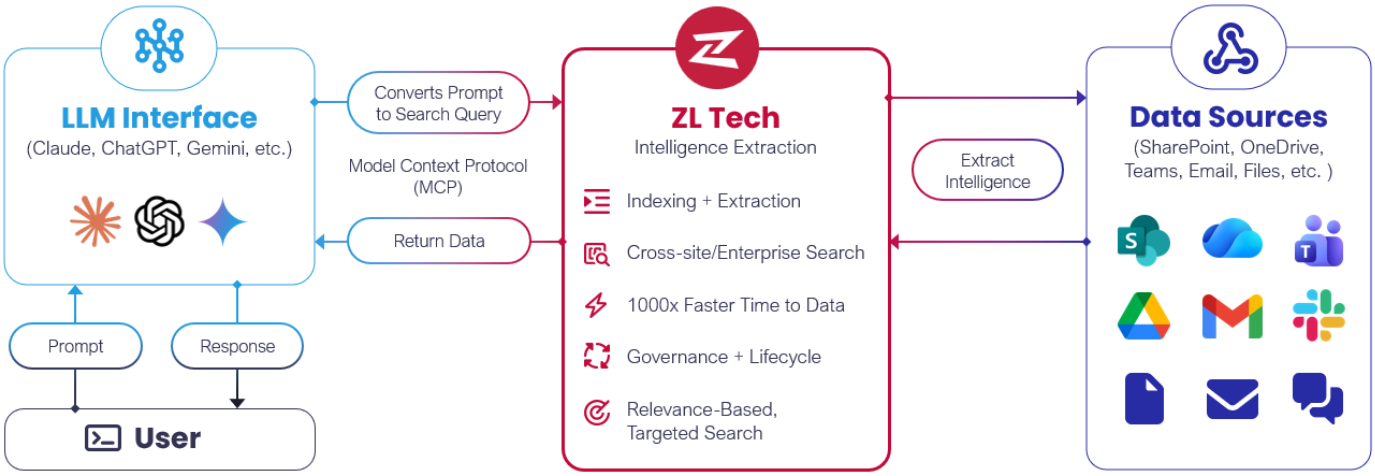
Enterprises that build AI datasets without a governed content foundation spend months on data unification, sensitivity review, and format normalization before any model sees a document. In a competitive landscape where the latest AI technology evolves monthly, any time lost is crucial.

ZL customers begin dataset construction from a corpus that is already unified, classified, and governed.

- ✓ **Automated collection, filtering, and packaging** from the governed corpus established in Milestones 1 and 2.
- ✓ **Token efficiency:** ZL's metadata-enriched document "essence" reduces token cost compared to feeding raw documents into the model context window.
- ✓ **For RAG:** the governed ZL index serves directly as the retrieval corpus; the ingestion and indexing layer is already formed.
- ✓ **For fine-tuning:** ZL curates targeted training sets from classified, lifecycle-managed content, filtered to the relevant domain and sensitivity profile.
- ✓ **For domain-specific agentic systems:** purpose-fit datasets for each agent function (compliance, financial analysis, customer intelligence) rather than a single general corpus.
- ✓ **Defensible deletion** as continuous dataset quality control: ROT data is removed from the corpus on an ongoing basis before it reaches the pipeline, improving the relevance and accuracy of every model drawing from the environment.

# MCP and Pipeline Integration

Once the dataset is built, it has to reach your AI tools without losing the governance applied to it. That is what ZL's MCP integration provides.




ZL connects governed enterprise data to AI tools, agents, and model pipelines through a Model Context Protocol (MCP) server. The MCP server handles data and action requests from external AI systems, returning current, relevant content while enforcing the full governance controls applied in the classification and dataset construction steps. For enterprises already operating with a model provider or AI platform, MCP provides a standardized, governed data access layer that integrates with existing infrastructure.

- ✓ The ZL MCP server handles data and action requests from external AI systems in real time.
- ✓ Access controls and governance policies are enforced at the MCP layer; every query is subject to policy and every access is logged.
- ✓ No new copies of source documents are created; agents access governed document essence through the MCP interface.
- ✓ **Databricks integration:** ZL-governed content is accessible to Databricks AI and ML pipelines, enabling end-to-end governed AI workflows within the Databricks Data Intelligence Platform for enterprises operating in that environment.

## MILESTONE 4:
# Ongoing Governance

**Why it matters:** data and regulations keep changing, so readiness is a continuous function. Every cycle of cleanup and reclassification improves the environment for every model drawing from it.

AI readiness is not an ad-hoc project. Enterprise data changes continuously as new content is created, existing content ages out of relevance, and regulatory requirements evolve. AI models require periodic retraining or retrieval corpus updates to remain accurate and compliant. The governance infrastructure that makes an AI program deployable must also keep it operating reliably over time.

Ongoing governance is also the mechanism that compounds the return on the initial data preparation investment. Each cycle of defensible deletion improves the data environment for every agent and model drawing from it. Each classification policy update propagates across the full data estate without reprocessing. The governed corpus becomes more accurate, more relevant, and more compliant over time.

- ✓ **Defensible deletion** as a continuous function: ROT removed on an ongoing basis in alignment with retention requirements, improving data quality for every model and agent operating across the environment.
- ✓ **Recategorization:** when governance requirements change, updated policies apply across the full data estate without reprocessing the entire corpus.
- ✓ **Unified audit trails:** evidence-quality records of every data access event across all sources in a single searchable system, not fragmented logs.
- ✓ **For agentic deployments:** continuous monitoring of agent data access is an operational requirement; the audit trail is the mechanism for detecting and investigating unexpected agent behavior.
- ✓ **Retraining and corpus refresh:** ZL's continuously updated master index enables periodic model retraining and retrieval corpus updates as the underlying data evolves.

**Deployed**