



# Ratatosk — Semantic Discovery and Manifest Generation

Mimir Labs Technical Publication

**Document:** ML-WP-009      **Version:** 1.0  
**Author:** Christopher Gaither      **Date:** March 2026  
**Classification:** Public

---

## 0.1 Overview

Ratatosk is the discovery layer of the Mimir Labs architecture stack. It is designed to extract, normalize, and formalize the semantic meaning of enterprise data before large-scale system transformation occurs. Like all tools in the Mimir Labs suite, Ratatosk is system-agnostic — it analyzes any enterprise database or data environment without requiring a specific target platform.

Most enterprise software initiatives fail not because of tooling limitations but because organizations cannot articulate the meaning of their own data structures. Definitions for fundamental business entities such as “customer”, “inventory”, or “available quantity” often differ across departments, systems, and reporting layers. Ratatosk addresses this problem by creating a structured, machine-readable description of enterprise meaning.

The output of Ratatosk is a canonical semantic manifest describing entities, attributes, relationships, and operational constraints across the enterprise data environment.

This manifest becomes the foundation for downstream systems in the Mimir Labs stack, including migration tooling, synchronization layers, and governance engines.

## 0.2 Role in the Mimir Labs Architecture

Ratatosk operates as the entry point into the broader architecture stack.

The stack is structured as follows:

1. **Ratatosk** — semantic discovery and manifest generation
2. **Mimisbrunnr** — canonical semantic reference model
3. **Ragnarok** — migration and transformation engine



4. **Bifrost** — cross-system synchronization layer
5. **Jormungandr** — governance and lifecycle enforcement
6. **Yggdrasil ERP** — operational execution environment

Ratatosk analyzes existing enterprise systems and produces a semantic manifest aligned to the canonical reference model provided by Mimisbrunnr.

Downstream systems use this manifest to guide data transformation, synchronization, and governance.

### 0.3 Problem: Semantic Drift

Enterprise environments accumulate multiple operational systems over time. ERP platforms, warehouse systems, CRM tools, analytics pipelines, and custom applications all maintain their own data definitions. Although the structures may appear compatible, the meaning of key fields frequently diverges.

Common examples include:

- “Customer” representing a billing entity in finance but a purchasing contact in CRM
- “Available inventory” meaning different things depending on reservations, quality holds, or warehouse states
- “Revenue” being recognized differently between operational reporting and accounting systems

These semantic inconsistencies propagate through integrations and reporting layers, eventually producing operational conflict.

ERP migrations frequently expose these conflicts only after implementation has begun, when correction becomes expensive and disruptive.

Ratatosk moves this discovery process earlier by making semantic inconsistencies visible before migration or integration begins.

### 0.4 Discovery Workflow

Ratatosk is typically deployed through a structured discovery engagement.

The discovery process proceeds through several stages.

#### 0.4.1 System Mapping

Existing systems are cataloged, including:

- ERP platforms
- Warehouse management systems



- CRM systems
- Manufacturing execution systems
- Analytics environments

Schema structures and operational data models are captured.

### 0.4.2 Entity Extraction

Core business entities are identified across systems. Examples include customers, suppliers, orders, products, inventory locations, and financial accounts.

Each entity definition is examined for structural and semantic differences.

### 0.4.3 Semantic Alignment

Entities are mapped to canonical definitions derived from the Mimisbrunnr reference model.

Conflicts in meaning, lifecycle state, or operational constraints are documented.

### 0.4.4 Manifest Generation

The discovery process produces a machine-readable manifest describing the enterprise semantic model.

This artifact becomes the authoritative description of enterprise meaning.

## 0.5 Artifact Portfolio

Ratatosk produces a layered set of deliverables designed for different audiences — from executive stakeholders evaluating data maturity to technical operators preparing for migration. Every artifact is deterministic and reproducible. None contain row-level data.

### 0.5.1 Canonical Manifest

The primary machine-readable output is the Ratatosk Manifest, a JSON document that captures the full semantic structure of an enterprise data environment. It is the artifact that downstream tools — Ragnarok, Bifrost, and Jormungandr — consume directly.

The manifest contains:

- **Source descriptors** — connection metadata, source classification (system of record, shadow system, legacy, reference), and profiling capabilities
- **Table annotations** — business labels, taxonomy group assignments, semantic notes, ontology concepts, and review status with full audit history
- **Field-level annotations** — per-column business labels, semantic notes, ontology mappings,



and provenance origin

- **Ragnarok mapping hints** — target table, per-column target mappings, WHERE clause filters, and confidence scores for every proposed mapping
- **Data steward assignments** — named accountability for taxonomy groups and individual tables
- **Policy and quality rule configurations** — governance thresholds and evaluation criteria
- **Auto-label metadata** — algorithm version, match thresholds, confidence caps, and dictionaries consulted, ensuring reproducibility
- **Data audit baselines** — aggregate profiling statistics (row counts, null rates, distinct counts, string lengths) captured under zero-knowledge constraints

This manifest is not a report. It is an operational artifact that drives migration planning, synchronization configuration, and governance enforcement across the entire Mimir Labs tool suite.

### 0.5.2 Executive Summary

A four-page PDF designed for leadership and steering committees. It presents the governance state of an organization's data in terms that support investment and prioritization decisions.

The summary includes a visual field coverage score, color-coded health indicators for definition consistency, ownership clarity, governance maturity, migration readiness, and schema completeness. A narrative section translates these metrics into plain-language findings: how many fields were assessed, where semantic collisions were detected, which areas lack clear data ownership, and how many remediation items the action plan contains. The report classifies overall governance maturity as Foundational, Developing, or Structured, with contextual explanation of what that classification means for operational risk.

### 0.5.3 Action Plan

A prioritized remediation roadmap that converts governance findings into executable work. Available as both PDF (for stakeholder review) and JSON (for project management tooling).

Action items are organized into four phases: Critical items requiring immediate attention, High-priority items to plan within weeks, Consolidation items for medium-term improvement, and Migration-prep items that establish the foundation for data movement. Each item includes a description, expected outcome, affected entities, effort estimate, and impact score. The plan also includes an operational risk inventory that identifies scenarios like duplicate order propagation or financial reconciliation failures that governance gaps could produce.

The plan computes an overall governance score and estimates the improvement achievable if all items are executed, giving stakeholders a concrete measure of return on governance investment.



### 0.5.4 Governance Summary

A comprehensive technical report covering the full depth of governance analysis. It includes field coverage breakdowns, per-source classification tables, taxonomy distribution charts, structural observations (collision counts, unclassified fields), schema quality profiles (type distributions, nullability coverage, FK density, orphan tables), annotation depth metrics, and migration readiness scoring with confidence distribution histograms.

### 0.5.5 Conflict and Coverage Report

A detailed collision analysis identifying where semantic definitions disagree across sources or within the same source. The report distinguishes label collisions (same business concept, different labels), name collisions (same field name, different semantic assignments), and ownership divergences. Priority analysis ranks collisions by severity and distinguishes cross-source conflicts from intra-source inconsistencies.

### 0.5.6 Data Quality Report

A rules-based quality evaluation that assesses primary key presence, foreign key coverage, nullable ratios, naming conventions, type consistency, and column count bounds. Each rule produces a pass, fail, or not-applicable result with the specific threshold, actual value, and list of violations. Quality rules are capability-gated — if profiling data is unavailable, the rule reports N/A with an explanation rather than failing incorrectly.

### 0.5.7 Policy Compliance Report

A governance policy evaluation covering organization-defined rules with pass/fail results, actual-vs-threshold comparisons, and per-rule violation details. This artifact supports compliance workflows where governance standards must be demonstrably met before migration proceeds.

### 0.5.8 Stewardship Matrix

A data accountability document that maps named stewards (with department, role, and contact information) to their assigned taxonomy groups and tables. This artifact establishes clear ownership before transformation work begins, preventing the common failure mode where no one owns the data definitions that migration depends on.

### 0.5.9 Visual Artifacts

Four SVG visualizations designed for presentations, workshops, and printed materials:

- **Coverage chart** — horizontal progress bar showing labeled vs. unlabeled field ratio
- **Taxonomy distribution** — stacked bar chart showing table counts per business domain



- **Collision summary** — bar chart comparing label collisions, name collisions, and ownership divergences
- **Data ecosystem map** — bipartite graph showing source systems connected to business domains via Bezier curves, with edge thickness proportional to table count, source nodes colored by classification (system of record, shadow system, legacy, reference), and a detailed cross-department FK path listing showing how data flows between organizational boundaries

### 0.5.10 Data Audit Baseline

When connected to a live source via ODBC, Ratatosk profiles every column to produce aggregate statistics: row counts, null percentages, distinct value counts, string length distributions, and empty-string counts. No actual row values are ever queried, computed, or stored — the engine operates exclusively on aggregate SQL functions (COUNT, AVG, MAX of lengths). The baseline enables data quality flagging: sparse columns, constant columns, all-empty text fields, and empty tables are identified automatically.

## 0.6 Integration with Mimisbrunnr

Mimisbrunnr provides the canonical semantic model used to normalize enterprise data structures. Ratatosk aligns discovered enterprise entities with this reference model.

This alignment allows organizations to translate disparate schemas into a shared semantic representation without forcing immediate structural changes to existing systems.

The canonical model therefore acts as a semantic intermediary between systems with incompatible schemas.

## 0.7 Integration with Migration and Synchronization Layers

Once a Ratatosk manifest has been produced, additional components of the architecture stack can operate against it.

### 0.7.1 Ragnarok

Ragnarok uses the manifest to generate transformation pipelines for ERP migrations or system replacements.

### 0.7.2 Bifrost

Bifrost uses the manifest to maintain semantic synchronization between multiple operational systems.



### 0.7.3 Jormungandr

Jormungandr uses the manifest to enforce governance policies and maintain semantic consistency over time.

## 0.8 Operational Architecture

Ratatosk is implemented as a standalone desktop application built using C++17 and Qt 6. It connects to enterprise systems through ODBC database connections and operates entirely locally — no cloud services, no external dependencies, no data transmitted off-premises.

The application provides a multi-window interface organized around discovery, annotation, and governance workflows:

- **Annotation Window** — the primary workspace where operators classify tables, assign business labels, review auto-generated mappings, and add semantic notes. Confidence-scored suggestions from the auto-label engine are presented for human review and override.
- **Lineage Window** — a visual data lineage graph that draws Bezier curves from source tables to target mappings, color-coded by confidence level. Clicking a connection reveals the column-level mapping detail. Filters by source, taxonomy group, and mapping status allow operators to focus on specific areas.
- **Workshop Display** — a projector-facing facilitation view designed for stakeholder workshops. It shows business-facing labels, taxonomy groups, and annotation progress without exposing technical schema details, confidence scores, or target identifiers.
- **Quality Dashboard** — rules-based data quality evaluation with configurable thresholds. Results display pass/fail status, actual values vs. thresholds, and violation details.
- **Policy Dashboard** — governance policy compliance evaluation with per-rule results and overall compliance scoring.
- **Catalog Panel** — a searchable data dictionary showing every source column with its enriched metadata: business label, taxonomy classification, semantic notes, annotation status, and assigned steward. Exportable to CSV.
- **Stewardship Panel** — data steward management and assignment interface that maps accountability to taxonomy groups and tables.
- **Export Dialogs** — pre-export validation (error/warning gating with override controls) and governance artifact generation with selective artifact toggles and engagement metadata.

The annotation engine uses a deterministic auto-label algorithm combining Levenshtein string distance, synonym normalization (50+ groups), and type compatibility scoring. All auto-generated labels are prefixed with SUGGESTED: and capped at 0.80 confidence to ensure that human operators remain authoritative. Every annotation carries a review status (Draft, Under Review, Approved, Locked) with a full audit trail of reviewer name, timestamp, and notes.



## 0.9 Deployment Model

Ratatosk can operate in several deployment configurations.

Typical modes include:

- Consultant-operated discovery environment
- Internal enterprise governance tool
- Pre-migration analysis platform

Because Ratatosk operates primarily on metadata and schema information, it can perform meaningful analysis without requiring full operational access to production systems.

## 0.10 Strategic Role

Ratatosk represents the first step in establishing coherent enterprise data architecture.

By formalizing the meaning of enterprise entities before transformation occurs, organizations gain a stable semantic foundation upon which migrations, integrations, and operational systems can be built. Because Ratatosk operates on schema metadata rather than requiring a specific target system, it delivers value to any organization regardless of whether they adopt Yggdrasil ERP or any other Mimir Labs product. The manifests it produces are consumed by the entire tool suite — Ragnarok, Bifrost, and Jormungandr — all of which share this system-agnostic design.

The Ratatosk manifest becomes a durable artifact describing enterprise meaning — one that can guide both current operations and future system evolution.

Ratatosk therefore functions not merely as a discovery tool, but as the mechanism through which organizations translate informal operational knowledge into a formal semantic architecture.

---

*Copyright 2026 Mimir Labs. All rights reserved.*

---

**Mimir Labs LLC** | Harrisburg, PA

© 2024–2026 Mimir Labs LLC. All rights reserved.

This document contains proprietary information belonging to Mimir Labs LLC. No part of this publication may be reproduced, distributed, or transmitted in any form without the prior written permission of Mimir Labs LLC. The information herein is provided “as is” without warranty of any kind. Product names and trademarks referenced in this document are the property of their respective owners.