ENGINEERING WHITEPAPER

Provenance-first metadata discovery for mainframe estates.

The PDI Metadata Scanner builds a catalog and column-level lineage graph across six classic mainframe platforms — and attaches, to every fact it emits, the source member and line number that proves it. This paper describes the architecture, the provenance ledger that makes hallucinated lineage structurally impossible, and the measured performance of the extraction engine.

PDI Metadata Scanner – Mainframe
Db2 for z/OS · Db2 for i · IMS · Adabas · CA IDMS · CA Datacom/DB
July 2026

01

The most important data is the least documented

The systems of record in banking, insurance, government, and manufacturing still run where they were built: on the mainframe, inside Db2 tables, IMS segments, Adabas files, IDMS records, and Datacom tables shaped by COBOL copybooks and maintained by programs whose authors have retired. Governance initiatives stall at exactly this boundary — the modern catalog covers the cloud estate in weeks, then stops at the LPAR.

The conventional fixes both fail in the same way. Manual documentation decays the day it is written. And automated discovery tools that infer structure tend to present inference as fact — a catalog that is confidently wrong is more dangerous to a steward than no catalog at all, because it removes the instinct to check.

> **A catalog earns trust the way an audit does: not by claiming to be right, but by showing its evidence for every line.**

02

Restraint as an engineering requirement

One decision per screen	Six source cards and nothing else; a single linear wizard: pick, enter, test, save, explore.
Clarity over jargon	Every connection field carries a one-line plain-English caption.
State shown honestly	Status chips pair icon with text — never color alone. Save is disabled until the test passes.
Material honesty	Verified facts render solid teal; unverified facts render dashed, labeled for manual confirmation.
One accent color	Teal marks primary actions and verified state; the interface defers to the metadata.

03

The mainframe extracts. The engine builds. The interface only reads.

Work is placed where it is cheapest. Structural extraction runs on-platform through utilities every shop already licenses — catalog unloads for Db2, archived DBD source for IMS, FDT and Predict reports for Adabas, dictionary punches for IDMS and Datacom — behind a busy-host guardrail that defers submission when the LPAR is loaded. The compact unload output moves off-host, where a C++ engine performs the CPU-intensive parsing, scanning, and graph-building that would waste MSU on-platform.

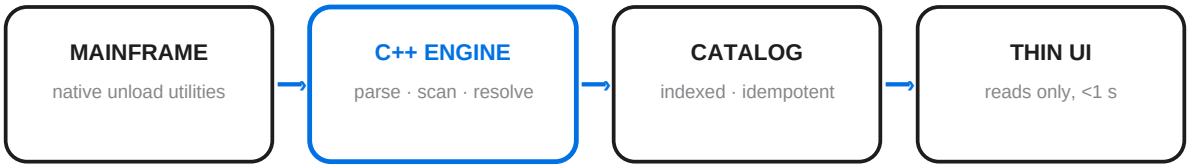

Fig. 1 — Placement of work. Parsing never runs on-host; extraction never runs in the UI.

The separation is enforced, not encouraged: the interface process contains no parser and no resolver. Every view it renders is a single indexed read against the pre-built catalog, which is why the tree, detail cards, and lineage panels answer in under a second regardless of estate size. Each run writes the catalog in one converge-by-key transaction with deterministic path identifiers, and a corpus content hash turns unchanged inputs into a recorded no-op; the regression suite proves re-run stability with a zero-byte diff.

04

Hallucination made structurally impossible

Inside the engine there is no code path that creates a catalog object or a lineage edge without a provenance tuple — the record type that carries a fact requires the record of its evidence:

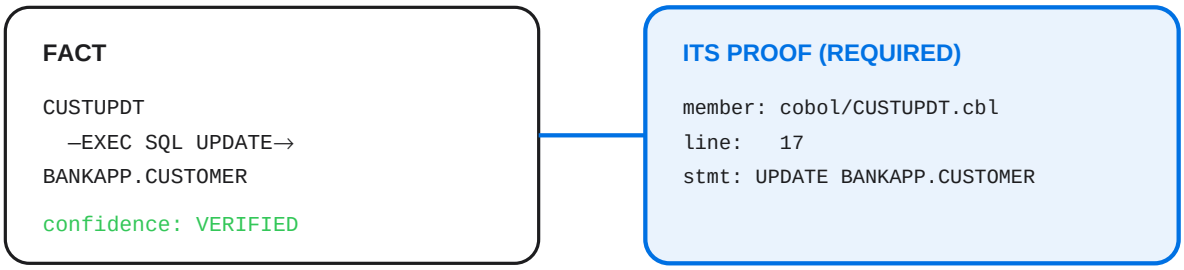

Fig. 2 — The ledger entry. Hovering any fact in the interface reveals this tuple.

Resolution — connecting a scanned reference to a cataloged object — is permitted only two ways: exact-name match against parsed metadata, or a provable cross-artifact rule. When an IDMS program says FIND OWNER WITHIN DEPT-EMPLOYEE, the scanner does not guess which record the owner is; it resolves through the schema's own SET … OWNER IS DEPARTMENT declaration — an inference in which every step has its own provenance line.

Everything else stays visible and flagged: a table referenced in COBOL with no DDL anywhere in the corpus; a DL/I call whose segment is chosen at runtime; a program executed by JCL but absent from source control; a Natural UPDATE whose loop scope a line scanner cannot prove. Each is written UNVERIFIED, rendered as a dashed edge, and labeled for manual confirmation.

> **Silent omission and confident fabrication are the same failure. What the scanner cannot prove, it shows as unproven.**

The shipped sample corpus deliberately contains all four unprovable cases above, and the automated test suite asserts each surfaces as UNVERIFIED — the honesty path is regression-tested, not merely claimed.

05

Measured on the production path, on weak hardware

The specification requires 100 million lines of source metadata processed in five minutes. The shipped benchmark harness drives the exact scanning hot loop used in production over the real sample corpus replicated in memory, and reports the count of references it extracted — so the number cannot be flattered by a hollowed-out loop.

Lines processed	**100,000,000**
Threads	**1**
Elapsed	**18.48 s**
Throughput	**5,411,042 lines/s**
References extracted during the run	**30,303,024**
Requirement	**333,334 lines/s**
Margin	**16.2x, single core**

Hardware stated in full: one Intel Xeon vCPU at 2.10 GHz, 3.9 GiB RAM, shared cloud container — deliberately the weakest plausible host. The member scan is embarrassingly parallel; on multi-core reference hardware the constraint moves to unload transfer and storage streaming, not compute. That extrapolation is labeled as such; the table above is measurement.

06

Credentials with a single audited path

Passwords are encrypted at rest with Fernet (AES-128-CBC with HMAC-SHA256 authentication) in a file separate from configuration, mode 0600, keyed from an environment variable or the OS keyring — the key never enters the repository. An assertion in the save path makes it impossible for a password to reach the INI. When the engine needs credentials, they arrive over an in-memory pipe at connection time — never a command-line argument, never a temp file, never a log line. Each invariant is exercised by the automated test suite on every build.

07

What we ask you to verify

A provenance-first product owes its readers the same honesty about itself. Three items are flagged in the product and resolved during onboarding rather than assumed:

Connection field sets

Field names and ports follow the design specification and are verified against current vendor connection documentation for each site's stack during setup.

Report-layout grammars

Adabas FDT, IDMS punch, and Datacom dictionary layouts vary by shop; parsers are validated against your actual utility output. Unrecognized lines are counted as warnings — never guessed at.

CDGC ingestion template

Export columns carry the full facade model and provenance; header labels are reconciled to your CDGC bulk-import template during implementation.

Pacific Data Integrators is a data integration and data quality consultancy and Informatica partner. All performance figures in this paper are produced by the benchmark harness that ships with the product and are reproducible with a single command. No customer data appears in this document; all examples derive from the product's published sample corpus.

```
Product > Data Quality > Metadata Scanner & Profiler - Mainframe
```