DEYIRVIEL · PROJECT OVERVIEW

M&A Clean Room — Diligence Without Ever Touching the Raw Data

Project overview — a 2-minute read before you open the repository

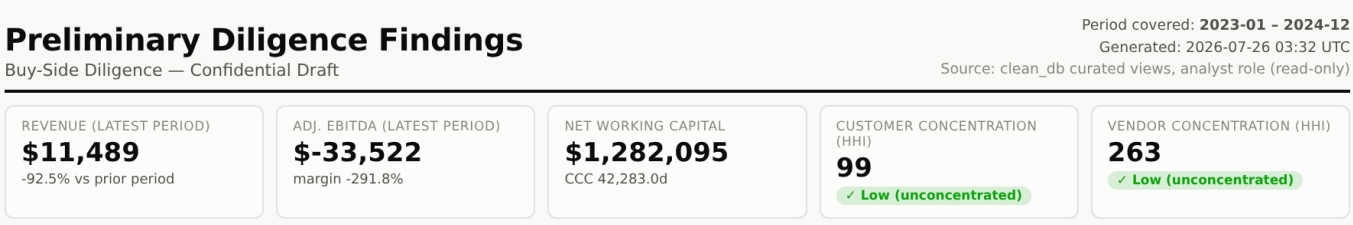
RAW ROWS AN ANALYST SEES	HARD GATES BEFORE ACCESS	CUSTOMER HHI (THIS RUN)	COMMANDS TO FULL REPORT
0	2	99 · Low	1

The project, in brief

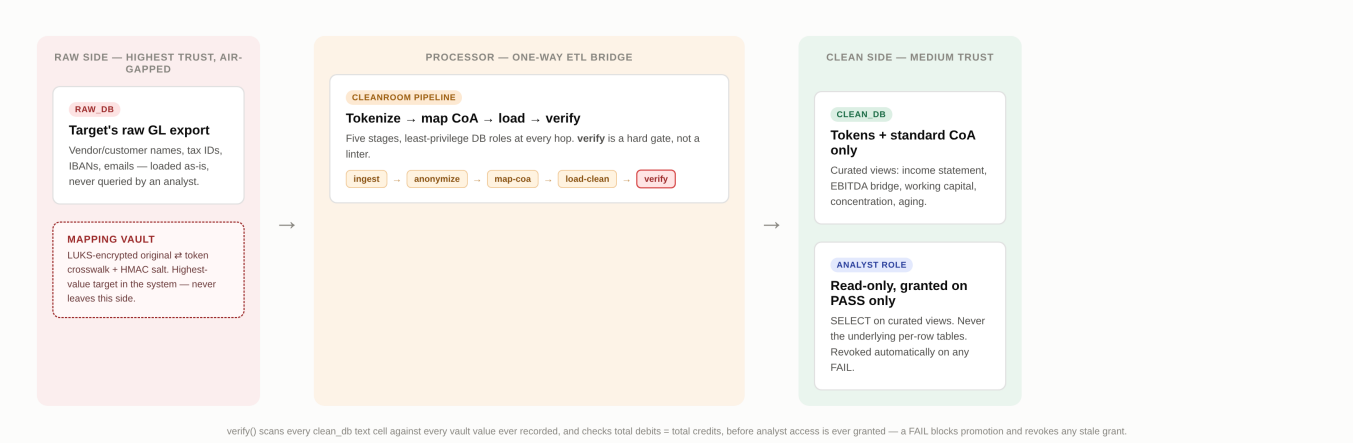
Before a deal closes, antitrust law and confidentiality agreements often mean a buy-side M&A team legally cannot see the target's raw general ledger — yet diligence still needs real revenue, margin, and working-capital numbers. The standard fix is an expensive, slow outside clean-room engagement. This project replaces it: a self-hosted pipeline that tokenizes a target's GL, gates analyst access behind a leak-scan and a trial-balance check, and renders a one-page diligence tear sheet — reproducible on the next deal for the cost of a server instead of a consulting invoice.

What it does

- Tokenizes every vendor/customer name, tax ID, and IBAN with HMAC-SHA256 before data reaches the analyst side
- Two independent hard gates — a trial-balance check and a leak scanner — either failing blocks all access
- Auto-generates a one-page tear sheet: EBITDA bridge, DSO/DPO/DIO, customer & vendor concentration (HHI)
- Raw and clean data live on structurally separate, air-gapped databases — not just a permissions setting
- Every run writes an append-only audit record: PASS/FAIL, violations, and a table-hash manifest



The tear sheet's header block: five KPI tiles, generated entirely from a read-only analyst role that has never touched a raw value.



Data flows one direction only: raw_db (air-gapped) through a five-stage pipeline into clean_db, which grants analyst access only after both gates pass.

What I found

The tokenization, access control, and audit trail are the expensive, hard-to-differentiate part of a bespoke clean-room engagement — this project is that work, done once, reviewed like any other code change, and reused instead of re-bought on every deal. The full repository, including a committed sample run and a documented threat model, is public on GitHub.

Want the full detail? The complete write-up — the tokenization mechanism, the finance explained line by line, and the honest list of limitations — is on the project page, with a link to the public repository.

Repository: github.com/yirvisom/M-A-Clean-Room — all figures are real, unedited output run against a synthetic target company.