ProjectX

GOVERNED GENAI OVER YOUR OWN DATA · USE-CASE BRIEF

An assistant over your documents that reads only what each person may.

Corpora, embeddings and a per-tenant vector index, retrieval filtered by the person asking, inference routed by policy or kept entirely in-house, guardrails and token budgets on every call, and an evaluation harness to prove it works.

The situation

The first assistant pilot sent documents to an external model, answered one employee with another team's files, and nobody could say what it cost or whether its answers were grounded.

How it works, step by step

01 **Build the corpus** · ML & GenAI
Chunk documents with a strategy per corpus, embed them with a model from the registry, and index them in a per-tenant HNSW index.

02 **Retrieve as the person asking** · ML & GenAI
Hybrid retrieval with reranking, filtered by the requesting user at query time.

03 **Decide where inference runs** · ML & GenAI
Route to self-hosted models or external providers by policy, or turn on sovereign mode and forbid external inference.

04 **Guard every call** · ML & GenAI
PII redaction, prompt-injection defence and toxicity filtering, output schema validation and token budgets per tenant, user and agent.

05 **Measure it** · ML & GenAI
An evaluation harness scores groundedness, relevance, latency and cost, and prompts are versioned and A/B tested.

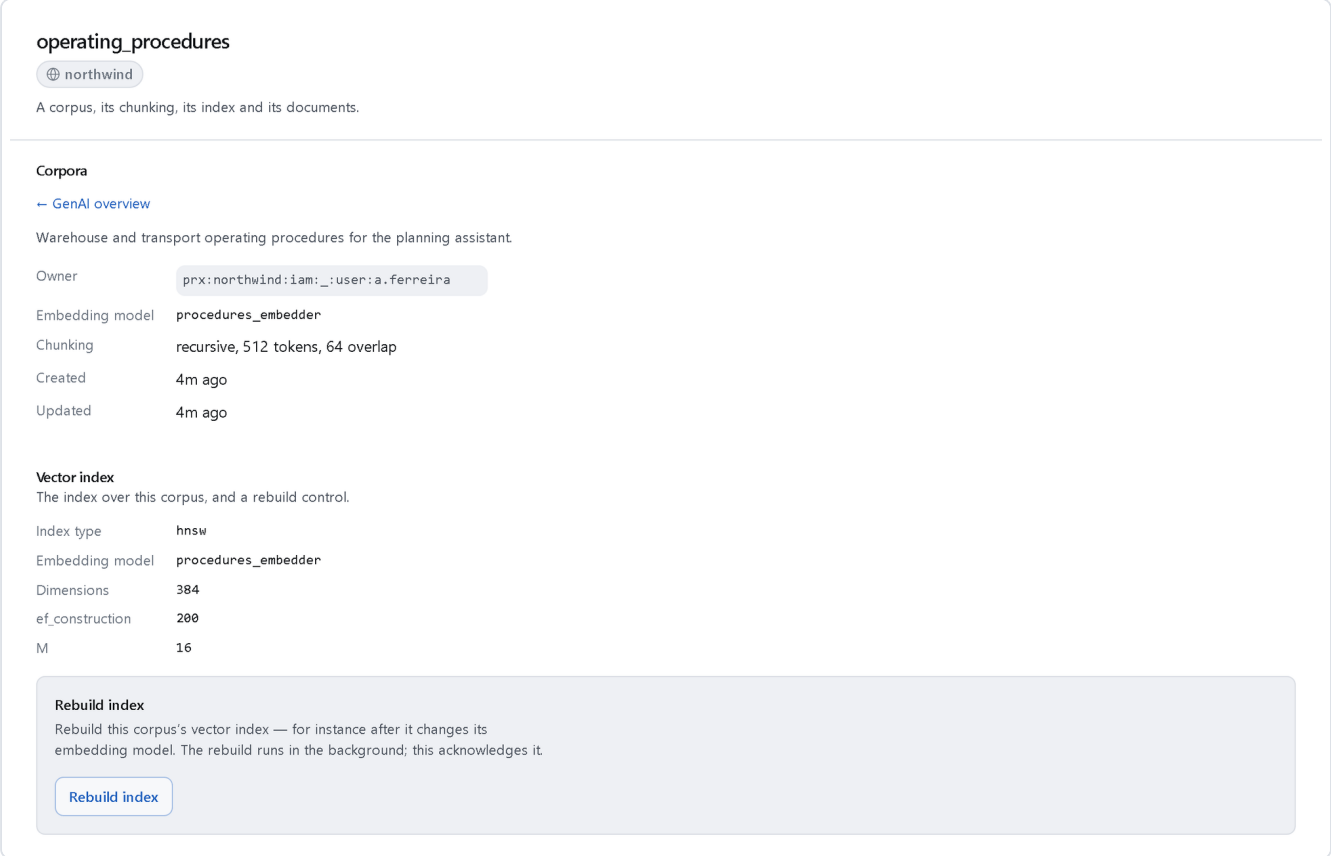
What you get

01
No document reaches the wrong person
Retrieval is filtered by the asker's own permissions at query time.

02
Inference stays where you decide
A sovereign tenant can forbid external inference entirely.

03
Spend with a ceiling
Token budgets cap use per tenant, per user and per agent.

In the console



A corpus, its chunking and the vector index over it. Screens shown with demo data.

Questions buyers ask

Which models can we use?
Self-hosted models served with vLLM or external providers, chosen per request by policy through the model gateway.

How do we know the answers are good?
The evaluation harness measures groundedness, relevance, latency and cost on your own question sets.