

DRAFT — UNOFFICIAL — NOT FOR OPERATIONAL USE

ARCHITECTURE REFERENCE

ODT-GDAP



LlamaIndex Persistence Strategy for GDAP

Goals

HEADQUARTERS
UNITED STATES ARMY EUROPE AND AFRICA
(USAREUR-AF)
Wiesbaden, Germany

DRAFT — NOT FOR OFFICIAL USE. FOR TRAINING PLANNING PURPOSES ONLY.

20 MARCH 2026

DRAFT — UNOFFICIAL — NOT FOR OPERATIONAL USE

LLAMAINDEX PERSISTENCE STRATEGY FOR GDAP

Date: 2026-03-07

GOALS

- Keep DuckDB as canonical truth
- Persist enough index state for fast startup
- Guarantee reproducible rebuilds after model/version changes
- Support rollback without data ambiguity

PERSISTENCE LAYERS

1) Canonical Layer (DuckDB)

- Path: `GDAP_DB` (`gdap.duckdb` by default)
- Content: doctrine atoms + temporal + analytics tables
- Role: authoritative source for all rebuilds and audits

2) LlamaIndex Storage Layer (Derived)

- `docstore` : node documents and metadata
- `index_store` : index metadata and index graph structures
- `vector_store` : embedding vectors and nearest-neighbor data

Recommended default directories (local/dev): - `data/llamaindex/docstore/` -
`data/llamaindex/index_store/` - `data/llamaindex/vector_store/`

VERSIONING AND REBUILD POLICY

Build Key

Every index build must record: - `source_snapshot_id` (DuckDB snapshot or timestamp) - `embedding_model_id` - `chunking_profile` (`chunk_size`, `chunk_overlap`, transformations) - `index_schema_version`

Rebuild Triggers

- Doctrine content/version changes in DuckDB
- Embedding model changes
- Chunking/transformation changes
- Index schema version changes

Rebuild Mode

- **Incremental:** when document-level diff is available
- **Full rebuild:** on model/schema/profile change

ROLLBACK STRATEGY

- Persist immutable snapshots of index artifacts per build key
- Promote snapshot pointer atomically on deploy
- Rollback by pointer swap + health checks

INTEGRITY CHECKS

- Node count parity against expected ingest set
- Metadata completeness for required identity fields
- Retrieval smoke checks on canary queries
- Embedding/version mismatch guard must block serving

OPERATIONAL NOTES

- Treat vector/doc/index stores as disposable caches with controlled lifecycle
- **Never use LlamaIndex stores as audit evidence** — DuckDB is the audit record
- Recovery runbook: restore DuckDB → rebuild or restore index snapshot → run canary gate

SEE ALSO

- [GDAP_ADR_0001_LLAMAINDEX.md](#) — Architectural decision: DuckDB owns canonical truth, LlamaIndex owns serving artifacts
- [GDAP_ACCEPTANCE_TESTS.md](#) — Section H: Operations and Lifecycle acceptance gates

DRAFT