

Ship & operate · Engineering handbook

Engineering invariants

🕒 7 min read · Updated 25 September 2026 · Verified against [f25ab97](#) on 2026-09-25

The rules every change must preserve: the prime directive on scripture, the data model, conventions, gotchas and how to verify.

<https://docs.gurbanisoul.com/engineering/invariants/>

The rules every change must preserve. The prime directive overrides everything else in this repository.

⚠ Prime directive: this is sacred scripture — never alter the text

This project is a knowledge base of **Sri Guru Granth Sahib Ji**, the living Guru of the Sikhs. Textual fidelity is the single most important rule and overrides everything else here.

- **Never edit, “correct,” paraphrase, normalize, reorder, translate-in-place, or guess at any Gurmukhi text** — not in the served database, the API, the web UI, copy/share, or search. The corpus and database are owned by [sggs-data](#)¹; nothing here writes them.
- If something in the scripture looks wrong, **flag it for human review — do not change it**. Add a note; never a “fix.”
- The corpus is **verbatim** from the source PDF and is **proven char-for-char** by sggs-data’s `pipeline/reconcile.py`. The only sanctioned text transforms are (a) PDF **visual** → **logical Unicode reordering** of the sihari (e.g.

ਿਕ੍ਰ

→

ਕ੍ਰਿ

) and (b) **editorial Unicode-repair corrections** where the source font emitted impossible sequences. The complete register is sggs-data’s [audit/editorial-ledger.jsonl](#)²: **4 rules, 11 applications** — Angs 573, 586, 727 (itemised since the start) and Angs 1354, 1358, 1387, 1398, 1402, 1406, 1408, 1409 (applied by the same `fix_text` step-4 rules, itemised 2026-09-23; scholarly review completed and approved 2026-09-24, gate G3). Any new transform must be registered there and reviewed; its `pipeline/ledger_check.py` (a CI gate there) fails a scripture change or a new `fix_text` rule that has no ledger entry.

- English translations are a **separate, labelled layer** (Dr. Sant Singh Khalsa via BaniDB/ShabadOS) — never blend them into the Gurmukhi, and never present a translation as the original.

- When answering Gurbani questions, follow the Answer Protocol: quote **verbatim**, cite the **Ang**, and clearly label any explanation as explanation.

Data model (lines table / JSONL record)

Each record is one display line:

id · ang (1–1430) · pdf_page · raag · section (bani) · author · comp_type · ghar · comp_id
(groups a shabad/unit) · line_no · is_rahao · is_header · markers (e.g. ॥

१॥

) · gurmukhi (verbatim, with dandas) · text (clean) · translit · translit_norm (phonetic fold, **DB-built**, not in JSONL) · fl_g/fl_r (first letters, Gurmukhi/roman) · skeleton (matra-stripped).

DB also has: fts/fts_en/fts_shabad/fts_tri, translations, variants, canon_tokens, raags/sections/authors, concepts/concept_lines, word_freq, analytics (*_analytics, theme_network, theme_fingerprint, author_resonance, vaars/vaar_units, *_neighbors), meta.

Conventions & gotchas (don't break these)

- **roman_norm exists in three places** — webapp/romannorm.py (query time), sggs-data's pipeline/sggs_pipeline.py (index time) and the app's RomanNorm.swift (the port) — and **must stay byte-identical**, or search silently breaks. contract/golden_roman_norm.ndjson (24,719 vectors, checked against the indexed column and vendored by the app) proves it; a change to the fold is a coordinated dataset release. See The Roman fold.
- **Search is a deliberate waterfall** (exact FTS → seeker lexicon → variant index → English → cross-line passage → skeleton-blob), BM25-ranked with weighted columns. Don't reorder tiers or change weights without re-running the harnesses; small changes shift ranking corpus-wide.
- **DB is opened mode=ro&immutable=1 + query_only**. The app must never write. All writes happen in sggs-data's pipeline.
- **Each bounded context declares the tables it reads** (TABLES in webapp/sggs/<context>.py; serve.CONTEXT_TABLES). A new query against another table must add it there — webapp/tests/test_declared_tables.py runs every route under an SQLite authorizer that denies undeclared reads, because the service split cuts each database slice from these declarations. Shared state lives only in sggs/core.py and is reached as core.NAME; context modules never import serve.
- **Logs never record what people search for**. The access log (one JSON line per request on stderr) carries method, path, status, duration and the request id — never the query string, IP or user agent (the privacy policy promises no added tracking). Every response echoes X-Request-Id (an incoming plain-token id or Vercel's x-vercel-id, else a random one) so a request can be followed from Vercel to the API; a startup line records version, commit, dataset and modules.

- **Security invariants to preserve:** FTS column allowlist + `_fts_clean` (strip `"/`); parameterized SQL everywhere; `realpath` path-traversal jail in `_resolve_static`; central HTML-escape helper in the front-end (all scripture/API text is escaped before `innerHTML`). Don't introduce string-built SQL or unescaped `innerHTML`.
- **Vaar attribution:** a Vaar's **pauris** take the Vaar's author (`vaar_author`) even though the interleaved **saloks** carry other Gurus'

ਮ:

headers. Detect the 22 Vaars by **title headers** (

ਵਾਰ

+

ਕੀ

/

ਧੁਨੀ

), not by pauri-run structure (structural detection over-/under-counts).

- **Raag spans** are computed as the longest contiguous run where a raag is the Ang's majority — liturgical occurrences elsewhere must not drag a raag's start. After `POST_RAAG_ANG = 1353` the Granth leaves the raag framework (saloks, swaiyye, Mundavani, Raagmala); raag is cleared and sections are header-detected.
- **comp_id groups a whole composition, heading run included.** A run of consecutive heading lines (a raag/title line, the

ੴ

invocation, a

ਸਲੋਕ ਮ:

label) shares the `comp_id` of the composition it opens (fixed v1.1.0, `build_corpus.py` post-pass 1b) so `/api/shabad/{comp_id}` and the iOS `fetchShabad` return the printed heading. The run adopts its **last** heading's id, so **no body line's comp_id ever changes** and the vacated ids become **permanent gaps** (distinct comps **4,527**; `max(comp_id)` 5,376 — read from the pinned database, checked by `sggs-data's verify_regroup.py --invariants`; `comp_id 1` is a gap — the Mool Mantar folds into Japji, `comp_id 2`). **v1.1.4** demoted 233 verses the detector had mis-flagged as headers (a comp-type word / leading raag name / substring Bhagat-name match inside a verse — see `detect_header's` weak-signal rule); each `comp_id` such a verse used to open is **burned** (`build_corpus.py comp_seq`) so every other composition keeps its id. Review pack: `sggs-data validation/header-fix-review/`. Three closing rubrics (

ਜੁਮਲਾ

,

ਦੁਤਰੇ

,

ਏਹੁ ਸਲੋਕੁ ਆਦਿ ਅੰਤਿ ਪੜਣਾ

— `TRAILING_RUBRICS`) belong to the *preceding* unit and stay one-line comps,

flagged for scholarly review. Any rebuild that touches the corpus (in sggs-data) must pass its pipeline/verify_regroup.py (only comp_id/line_no may change; scripture byte-identical).

- **Deploys are CI-gated (v1.1.1+).** Production is deployed only by .github/workflows/deploy-production.yml after every required check on the exact SHA passes (Render hook ref=SHA → /api/health.commit must equal the SHA → web built unaliased + @smoke → promote → public smoke → tag). Never deploy by hand from a dirty tree; never tag before a verified deploy. Release PRs integration→main are **merge commits**. See docs/process/runbooks/deploy.md.
- **comp_type is known-mislabeled** in places (Japji tagged

ਰੁਤੀ

/

ਵਾਰ

; many shabads tagged

ਪਉੜੀ

); it's suppressed at the display layer. Don't rely on it for logic; prefer comp_id/section.

- **Japji (385 lines) has author = null** (the print has no per-line

ਮਹਲਾ

), so author filters for Guru Nanak miss Japji. Known/deferred.

- **Nitnem / Gutka (iOS app, Algorythmos-AI/gurbani-soul-ios).** The registry lives in banis/bani_lines/extra_lines (ADR-0006): SGGS lines are pointers into the verbatim corpus (cited by Ang), extra_lines is a separate labelled Sri Dasam Granth / Ardaas layer (no English, never an Ang, never a bookmark). Pauri/ashtapadi/salok numbers come ONLY from BaniLine.markers via GurbaniSearchKit/BaniOutline (pure, strict-gate → [] rather than a guess, never emits Gurmukhi). One clock: Shared/NitnemClock defines “now” and a **03:00 Nitnem day** (dayKey(now-3h)) used by the home, reader, widgets and reminders alike. Progress (nitnem-progress.json, App Group, schema **v1 frozen**) stores a verbatim anchor+nLines so a saved position survives a DB rebuild, and never overwrites a newer schema; My-Nitnem sets + the journey go in a separate file (not this one). Reader tones use Ink.paper/Ink.paperWarm (contrast-proven) or a forced dark scheme; the hero glow is ≤14% (brand book §6 “Glow”). Frozen XCUI identifiers are listed in docs/nitnem/spec.md.
- **db/sggs.sqlite is installed, never committed** (make dataset; git-ignored). Don't bloat the repo with node_modules/dist either (git-ignored — keep it that way).
- **The database comes from sggs-data, by pin.** dataset.lock.json names a commit of Algorythmos-AI/sggs-data and the database's sha256 and size. CI installs it only through scripts/data/fetch_dataset.py (sha256-verified, atomic). The integrity check fails unless the lock and contract/_meta.json name the same object, sggs-data publishes that object at the pinned commit, and the installed database passes quick_check with 60,658 lines over Angs 1–1430. A data change therefore lands in sggs-data first; this repository then bumps the lock in a reviewed PR.

Verify your changes

- Dataset bump → make dataset, make contract, then `python3 webapp/serve.py` and check `http://localhost:7777/api/health` (all checks true). The scripture proofs for that dataset ran in sggs-data.
- Search change → run the harnesses in `tools/` (make harnesses; `chaos_harness.py qa/chaos/attacks_*.jsonl <out>` against a running server) and confirm no regressions.
- Never mark scripture-touching work “done” without a human reviewing any text-level diff.

Links

1. <https://github.com/Algorythmos-AI/sggs-data>
2. <https://github.com/Algorythmos-AI/sggs-data/blob/main/audit/editorial-ledger.jsonl>

Sri Guru Granth Sahib Ji — Knowledge Base wiki. Generated from commit c3d9360. Updated 25 September 2026. Verified against f25ab97 on 2026-09-25. This copy may be out of date; the page is at <https://docs.gurbanisoul.com/engineering/invariants/>