

Ship & operate

Release Process

🕒 2 min read · Updated 13 October 2026 · Verified against [a661527](#) on 2026-09-26

The steps that take a green integration branch to a tagged production release.

<https://docs.gurbanisoul.com/process/release/>

1. On integration, ensure everything is green on staging.
2. Bump the unified version: `make release VERSION=1.2.0` (runs `bump.py` + `check_versions.py`). Fill in the CHANGELOG section it stubs.
3. If the dataset changed: it was rebuilt, proven and published in sggs-data first; here the release carries the reviewed `dataset.lock.json` bump and the regenerated contract (`make contract`).
4. Open a **release PR integration** → **main**, labelled release. All gates must pass.
5. Merge (a merge commit). `deploy-production` deploys the exact commit, verifies it, and only then tags `vX.Y.Z` and cuts a GitHub Release from the CHANGELOG section. The tag names the commit, and that commit's `dataset.lock.json` names the exact database it serves.
6. Confirm with `make verify-prod` (commit identity, health, Ang 712 heading, web deployment). The release that moves production's API contexts onto their own functions (a later one, after the app is live; [services-production](#) steps 2–3) also owes the performance gate, from the same vantage as the baseline: `python3 tools/perf_baseline.py --base https://gurbanisoul.com --from "<same vantage>" --compare docs/perf/baseline-2026-09.json` must PASS (p95 within +10 % per context), and each context's X-Service must name itself.
7. **iOS is part of every release.** After **main** is tagged `vX.Y.Z`, the app repository ([Algorythmos-AI/gurbani-soul-ios](#)¹) vendors this release (`make vendor-sync-platform REF=vX.Y.Z`), sets `MARKETING_VERSION` to `X.Y.Z`, tags its own `vX.Y.Z` and uploads the App Store binary (`make testflight TEAM_ID=... BUILD=1 CHANNEL=appstore UPLOAD=1`; the archive refuses an appstore upload not built against platform `vX.Y.Z`). It commits its ledger. The release is **complete** only when `scripts/release/check_release_complete.py X.Y.Z` passes.

Versioning

SemVer, **one unified version number across web, API and iOS** — the footer, `/api/meta`, `/api/health`, the iOS About screen and the App Store all read the same `XYZ`, and they are never allowed to differ. **Every release is re-archived at the same version** and uploaded as iOS `X.Y.Z` (1) from the release tag, even when the change is web-only — so the number in the App Store always equals the live site. The iOS **build**

number restarts at 1 per version and only increments for a re-upload of the *same* version (e.g. an App Review fix). Never ship a one-sided hotfix: web or iOS fixes bump the **patch** everywhere. **MAJOR** = breaking API/data-model (e.g. a `comp_id` renumber); **MINOR** = features or data corrections; **PATCH** = fixes. RCs on integration are `vX.Y.Z-rc.N`. The DB keeps its own decoupled `meta.version`. See [ADR-0004](#).

Links

1. <https://github.com/Algorythmos-AI/gurbani-soul-ios>

Sri Guru Granth Sahib Ji — KnowledgeBase wiki. Generated from commit c3d9360. Updated 13 October 2026. Verified against a661527 on 2026-09-26.
This copy may be out of date; the page is at <https://docs.gurbanisoul.com/process/release/>