

Architecture

Three repositories and pins

🕒 1 min read · Updated 26 September 2026 · Verified against [a661527](#) on 2026-09-26

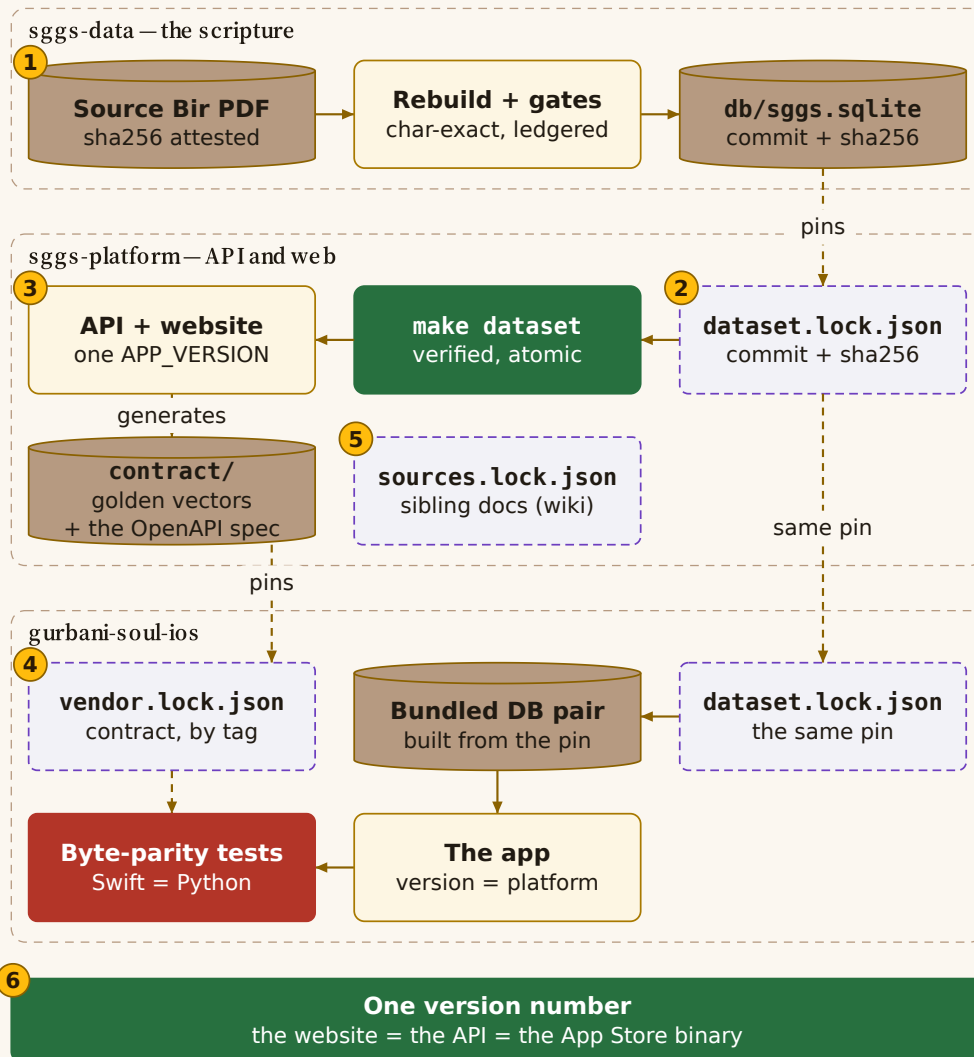
The hand-offs between sggs-data, sggs-platform and gurbani-soul-ios, each a reviewed lock file with a commit and a checksum, drawn as one poster you can step through.

<https://docs.gurbanisoul.com/architecture/three-repositories-and-pins/>

The project is three repositories joined by pins — small, reviewed files naming an exact commit and the sha256 of the artifact taken from it — so nothing is copied by hand and nothing can drift. The newcomer's version of this page is [How the repositories fit together](#); the decisions are [ADR-0007](#) and [ADR-0008](#).

02 · Three repositories and their pins

Nothing is copied by hand between data, platform and app — every hand-off is a reviewed lock file with a commit and a checksum



Poster 02 — three repositories and the pins that join them

1. **The scripture is built and proven in sggs-data.** The source PDF's checksum is attested. The pipeline extracts the corpus, reconciles it character for character against the PDF, runs the golden checks, builds the SQLite database and passes the integrity gate. Any editorial repair is registered in the editorial ledger and approved in a scholar review. The database is published by commit, sha256 and size.

2. **The platform pins it.** dataset.lock.json names the sggs-data commit and the object's sha256 and size. make dataset downloads exactly that object, verifies it and installs it atomically; make dataset-check proves sggs-data publishes it at that commit. A new dataset arrives only as a reviewed bump of this file.
3. **The platform serves and publishes.** webapp/ and frontend/ serve the API and the website at one version. From the running code, tools/gen_golden_vectors.py records the golden contract — thousands of input and output vectors for the Roman fold, search, verification and the reader — and tools/gen_openapi.py the OpenAPI spec.
4. **The app vendors and re-proves.** gurbani-soul-ios pins the contract at a platform release tag (vendor.lock.json, a sha256 per file) and the same dataset pin. It builds its own bundled database pair from that pin and runs byte-parity tests so its Swift port equals the Python source of truth. Its MARKETING_VERSION must equal the platform version it vendored.
5. **This wiki pins the sibling docs.** docs-site/sources.lock.json records, per sibling repository, a commit and the sha256 of every published documentation file; the build installs them under /data/ and /ios/ with a banner naming the canonical source. A pin bump is a small reviewed pull request, opened by the weekly docs-pins job.
6. **One version number.** The web footer, /api/meta, /api/health, the iOS About screen and the App Store all report the same X.Y.Z, and every platform release is re-archived as iOS X.Y.Z (1) from the release tag. A release is complete only when web, API and the binary report the same number from the same commit.

The four pins

Pin	Where	Names	Proved by
dataset.lock.json	platform	sggs-data commit · db/sggs.sqlite sha256 · size	make dataset-check (scripts/data/fetch_dataset.py -- check-pin), scripture-integrity
dataset.lock.json	app	the same object	the app's CI
vendor.lock.json	app	platform tag vX.Y.Z · sha256 per vendored contract/ file	the app's parity tests and release gate
docs-site/sources.lock.json	platform (wiki)	sibling commit · sha256 per published doc	tools/fetch_sibling_docs.py --check in the docs job

A bump of any of them is a small pull request that a reviewer can read in full. The wiki's own pin is proposed automatically: every Monday docs-pins opens that pull request when a sibling changed a file the wiki publishes, and moves nothing otherwise.