

Decisions · Platform

ADR-0012: The wiki is rendered as a docs site over the Markdown in docs/

🕒 3 min read · Updated 25 September 2026

Why the engineering wiki gets a rendered, searchable site at docs.gurbanisoul.com while every page stays PR-reviewed Markdown in docs/, and how sibling docs and diagrams are brought in.

<https://docs.gurbanisoul.com/adr/0012-docs-site/>

Status: accepted (2026-09-25). Extends ADR-0003; does not replace it. Amended 2026-09-26: the Content-Security-Policy allows inline scripts by sha256 hash, not 'unsafe-inline', and the e2e suite runs under it (see *Amendments*).

Context

ADR-0003 put the wiki in docs/ as Markdown + Mermaid so it is versioned with the code and reviewed in pull requests. Read raw on GitHub it stays a set of files: no search, no sidebar, small diagrams at page width, plain URLs to the two sibling repositories (ADR-0007), and nothing interactive. Interns and students need one place that reads like a product — a site with instant search, dark mode, large step-through diagrams and live playgrounds against the read-only API — without giving up review, versioning, or GitHub rendering.

Decision

- **One source, two renderings.** Every page stays a Markdown file in docs/ (frontmatter title/description, the body keeps its # Title, relative links written for GitHub). A separate Astro Starlight project, docs-site/, reads ../docs in place and publishes it at **docs.gurbanisoul.com**. Nothing is copied; docs/reports/archive and docs/design are history and are linked, not published.
- **Sibling docs by pin.** docs-site/sources.lock.json records, per sibling repository, a commit and the sha256 of every published file; tools/fetch_sibling_docs.py installs them at build under /data/... and /ios/... with a “canonical source” banner and edit links into that repository. A bump is a reviewed PR, exactly as dataset.lock.json (ADR-0008).
- **Diagrams render at build.** Mermaid fences become inline, accessible SVG on the site (one brand theme from docs/brand/tokens.json; GitHub keeps rendering the fence itself); large process posters are hand-built SVGs under

docs/diagrams/posters/ with a steps sidecar that the site turns into a walkthrough. A diagram that fails to render fails the build.

- **Interactive elements degrade.** Widgets are embedded from plain Markdown as `<!-- sggs:<name> -->` comments (invisible on GitHub) and must be followed by static fallback text. They call `/api` same-origin; the site rewrites it to the public API on the product host. No analytics, no third-party scripts, no cookies.
- **Gated like the product.** `tools/docs_check.py` (frontmatter, links and anchors, widgets, Mermaid palette, the scripture-quotation rule verified against the pinned database, poster spec, site configuration) runs in the required python check and in the docs job, which also builds the site with links validated, runs unit, end-to-end and accessibility tests and the performance budgets. Deploys are CI-only (ADR-0005): `integration` → the docs staging alias, `main` → build, deploy unaliased, smoke by commit, promote, smoke the public domain, roll back on failure.

Consequences

Authors write Markdown as before plus a two-line frontmatter block (GitHub shows it as a small table). Relative links work on both renderings; a broken one fails CI. The site adds a second Vercel project and a docs bypass secret (owner setup in the deploy runbook). Scripture never appears typed into a page: only as an API-fetched verbatim line with its Ang, or a cited blockquote the gate verifies. Starlight is used with Astro's unified Markdown processor because the wiki's plugins are remark plugins; a move to Astro's Sättteri processor is possible later without touching any page.

Amendments

- **2026-09-26 — inline scripts by hash.** The launch CSP allowed `script-src 'unsafe-inline'`. The build's inline scripts are few and fixed per package version (13 on 149 pages: Starlight's theme, sidebar and search dialog, and the OpenAPI reference's tab pickers), so `script-src` now lists their sha256 and nothing else runs inline. `docs-site/scripts/csp.mjs --check` fails the docs job when the build and the policy disagree, and `scripts/serve-dist.mjs` serves the production headers so the e2e suite runs under the real policy.