

Decisions · Platform

ADR-0010: One image per context, behind a gateway generated from the code (staging first)

🕒 2 min read · Updated 25 September 2026

Why each bounded context can run as its own image behind a gateway generated from the route table.

<https://docs.gurbanisoul.com/adr/0010-services-behind-a-generated-gateway/>

Status: accepted (2026-09-25); hosting superseded by ADR-0011 — the contexts now run as Vercel functions in the web project, not as Render services. The slicing, the generated routing and the proofs below stand.

Context

The API was split into five bounded contexts in code (reader, search, verify, insights, knowledge), each declaring the tables it reads. Running them as separate services must not create a second source of truth about which path belongs where, must not duplicate the database by hand, and must stay reversible one context at a time.

Decision

- **One image, any service.** The API image takes `SGGS_MODULES.all` (the default) is the single API. A single context cuts its database slice at build time with `tools/slice_db.py` — the declared tables plus foreign-key targets, FTS5 shadows and content tables — proves every kept table equal in content to the pinned database and every kept FTS index whole, and serves only its routes (`/readyz` refuses to start without every declared table; other contexts' routes return 404).
- **The gateway is generated from the route table.** `tools/gen_gateway.py` writes the Vercel rewrites from `serve.ROUTES` (every route is tagged with its context) and `gateway/routes.json` (which contexts have their own service, per environment). A test fails if `vercel.json` drifts or a prefix could belong to two contexts. Staging rules are host-conditioned and precede the catch-all.
- **Proven on every staging deploy.** Each service is deployed at the exact commit through the Render API and must report it on `/readyz`; each routed context must answer through the real gateway with its own `X-Service`; the whole golden contract (266 records) replays through the gateway.
- **Reversible.** The single API keeps serving the catch-all; moving a context back is one line in `gateway/routes.json`.

Consequences

Staging runs five services on free instances (they sleep when idle; the pipeline waits for them). Slices: search 74 MB, insights 66, reader 56, verify 44, knowledge 0.7 of the 108 MB database. Moving production is the same mechanism with paid always-on services (docs/process/runbooks/services-production.md), gated on the performance baseline (docs/perf/baseline-2026-09.json, +10 % p95 budget) and the data canary.

Sri Guru Granth Sahib Ji — Knowledge Base wiki. Generated from commit c3d9360. Updated 25 September 2026. This copy may be out of date; the page is at <https://docs.gurbanisoul.com/adr/0010-services-behind-a-generated-gateway/>