Srasta — Platform Architecture & Data-Flow

Version: 1.0 · **Last updated:** 2026-04-29 · **Audience:** prospective customers, design-partners, security reviewers, auditors

One-page summary

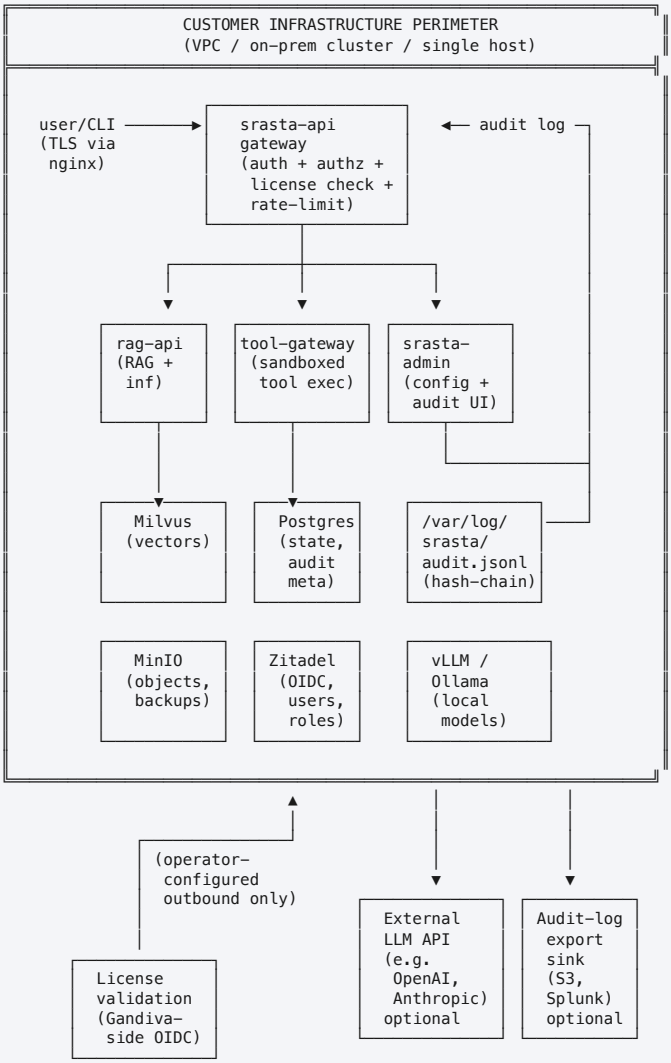
Srasta is a **single-perimeter governed AI platform**. Every component runs inside the customer's infrastructure boundary — their VPC, their cluster, their on-prem hosts. Customer data (documents, prompts, responses, audit records, identity records) **never leaves that perimeter** in normal operation. The only outbound traffic is the channels the customer explicitly enables (model providers, audit-log export sinks, license validation — all listed below).

Architectural invariant

> **Customer data lives where the customer says it lives. Srasta does not mediate, terminate, or store any customer data inside Gandiva-operated infrastructure.**

This invariant drives every other decision in this document. It is the most-asked-about property in security reviews, and it is the property auditors check first.

Trust boundary diagram



What runs where

Inside the customer perimeter (always)

These components ship as Docker images (or Helm charts) and run on the customer's infrastructure. Data they hold is the customer's data; it never leaves the perimeter unless an explicit outbound channel below is configured.

Component	Image	Role	What it stores
`srasta-api`	`srasta/srasta-api`	Single external entry point. Validates JWT (OIDC), enforces RBAC + per-role model whitelist, gates on license posture, rate-limits, signs forwarded principal headers, proxies to internal services.	Nothing on disk except its share of `/var/log/srasta/audit.jsonl` (gateway audit).
`srasta-admin`	`srasta/srasta-admin`	Operator UI + Postgres-backed config (`platform_config`). License management, model-access RBAC, audit-feed live view, security panel.	Postgres-resident config + admin-action entries on the shared `audit.jsonl`.
`rag-api`	`srasta/rag-api`	RAG pipeline: ingest, chunk, embed, retrieve, rerank, compress, route to LLM, stream back.	Audit entries on `audit.jsonl`. Stateless across requests.
`tool-gateway`	`srasta/tool-gateway`	Persona-scoped tool execution (git, file ops, sandboxed exec, MCP servers).	Workspace at `/workspace` (per-session); audit entries.
`Milvus`	`milvusdb/milvus`	Vector store for embeddings.	Customer documents → embeddings. Stays in customer perimeter.
`Postgres`	`postgres` (multi-DB host)	Relational state for langfuse, srasta-admin, infisical, litellm, n8n, dify.	Operator config, model-access grants, hardware inventory, ingest history, langfuse traces.
`MinIO`	`minio/minio`	S3-compatible object store for audit-log archive, backup bundles, langfuse media.	Backup bundles + audit archives. Customer's MinIO buckets, customer's data.
`Zitadel`	`ghcr.io/zitadel/zitadel:v4.13.1`	OIDC identity provider. Issues + verifies JWTs.	User records, roles, sessions.
`nginx`	`srasta/nginx` (thin)	TLS termination via Let's Encrypt wildcard or operator-provided cert.	TLS state, ACME accounts.
`vLLM` / `Ollama` / `TEI`	`srasta/vllm-gb10`, `ollama/ollama`, `ghcr.io/huggingface/text-embeddings-inference`	Local model inference + embedding (when the customer runs models in their own infra).	Compiled torch graphs, model weights cache. No prompt/response storage.
`Vector` (`audit-forwarder`)	`srasta/vector`	Optional audit-log shipping to S3 / SIEM.	In-flight buffer only.

Outside the customer perimeter (operator-controlled, optional)

These are the **only** ways data leaves the perimeter. Each is explicitly enabled by the operator; each can be turned off without breaking core platform functionality.

Outbound channel	When	What flows	How to disable
External LLM API (OpenAI, Anthropic, Bedrock, etc.)	Only when `platform_config.INFERENCE_HOST` is set to an external endpoint, or per-persona routing references one.	Prompt + completion text for the requests routed externally. The operator chooses which models route externally vs. locally; per-role model whitelist (#149) controls which users can hit external models.	Set `INFERENCE_HOST=http://litellm:4000` (local LiteLLM). Use only persona-routes that reference local-vLLM / Ollama models.
Audit-log export sink (S3 / Splunk / external SIEM)	Only when `--profile audit` is enabled and `AUDIT_S3_*` / `AUDIT_HTTP_*` env vars are set.	Hash-chained audit JSONL entries (auth, inference metadata, tool calls, admin actions, license events). No prompt/response payloads.	Disable the `audit` profile. Audit log stays local at `/var/log/srasta/audit.jsonl`.
License validation (Gandiva-side license-server)	Only when the operator runs `scripts/license-check.sh` or similar against the Gandiva-operated server. License JWTs are RSA-verified **offline** against the embedded public key — the runtime does NOT contact the license-server on every request. Operators only contact license-server at issuance/renewal time.	Operator-initiated request: license ID lookup, JWT issuance. No runtime customer data.	The runtime license check is offline-only by design.
Operational telemetry (Gandiva-operated endpoint)	Daily POST from the Srasta install (default ON for both community + enterprise editions).	Operational metadata only: install events, version, cluster size, feature-touch counts, error class counts. **Never** prompt content, response content, document content, audit-log content, or end-user identifiers. Full schema in `docs/legal/privacy-policy.md` Section 3.4.	`SRASTA_TELEMETRY=off` env var; admin UI License-page toggle (enterprise edition). Disabling does not affect platform functionality.
Brand / PWA assets (CDN-served on the website)	Marketing site at `srasta.ai`. Not part of any customer install.	None — not in the customer-install codepath.	N/A.

Where customer data lives

Every byte of customer data has a single, customer-controlled home.

Data class	Storage	Format	Retention default	Operator can change?
Source documents (RAG ingest)	Customer's filesystem (workspace), MinIO bucket, or external blob via configured ingest source.	Original format (PDF, MD, text, etc.).	Customer-controlled.	Yes — `platform_config.AUDIT_LOG_RETENTION_DAYS`, ingest-sources config.
Document embeddings	Milvus, in customer's deployment.	Float vectors + metadata.	Indefinite.	Yes — manual collection drop.
Prompts + completions	**Not persisted by default.** Held in memory by `rag-api` for the duration of the request only. Optional Langfuse trace storage in customer's Postgres if Langfuse is enabled.	Plain text.	Per-request only (default).	Yes — Langfuse is opt-in via `--profile langfuse`.
Audit records	Hash-chained JSONL at `/var/log/srasta/audit.jsonl` in customer's volume + optional MinIO archive + optional external SIEM.	JSONL (one event per line, SHA-256 chain).	90 days default, configurable via `AUDIT_LOG_RETENTION_DAYS`.	Yes — env var.
User records (identity)	Zitadel's Postgres database in customer's deployment.	Zitadel native schema.	Indefinite.	Yes — Zitadel admin actions.
Operator config (secrets, model whitelist, ingest sources)	Postgres `platform_config` table + `model_access` + `ingest_sources` in customer's deployment. Secrets marked `is_secret=true` are returned redacted by listing endpoints.	Key-value rows.	Indefinite.	Yes — admin UI.
Tool-execution sandbox state	`tool-gateway-workspace` named volume. Per-request workdirs.	Customer's filesystem.	Cleared between sessions.	Yes — workspace TTL.
Backup bundles	MinIO `srasta-backups` bucket in customer's deployment, or customer-configured external S3.	Postgres dump + .env snapshot + Milvus metadata, all bundled.	5 most recent upgrades by default.	Yes — `BACKUP_RETENTION` config.
License key (the JWT itself)	Postgres `platform_config.SRASTA_LICENSE_KEY` (is_secret=true) + optional `.env` fallback.	RSA-signed JWT.	Until operator clears it.	Yes — admin UI License page.

Trust boundaries

Each boundary below is an enforcement point — a place where data or control crosses a security domain. All boundaries are explicit; none is implicit.

Boundary	What's enforced	Mechanism
External user → srasta-api	TLS + OIDC JWT verification + role-based authorization + rate limit + license posture	nginx (TLS termination) → srasta-api (`auth.py` + `authz.py` + `_check_license_block` + `rate_limit.py`). Inputs that fail any check return 401/402/403/429 and never reach internal services.
srasta-api → internal services (rag-api / tool-gateway / admin)	Forwarded principal headers signed with HMAC (`X-Srasta-Signature`). Internal services reject requests without a valid signature. Direct external traffic to internal services is blocked at the network layer (Docker bridge, no exposed ports for internal services in customer mode).	`srasta-api/auth.py:sign_forwarded_headers` + downstream signature verification. `docker-compose.yml` does not publish internal-service ports.
srasta-api → external LLM provider	Per-role model whitelist (#149). Audit emission of every external call. TLS to provider via system trust store.	`srasta-api/model_rbac.py` + LiteLLM (which carries the call out).
rag-api / tool-gateway / admin → audit log	UID 1000 (`srasta`) for all four services (#164). Append-only writes. Hash-chain integrity (`audit/audit_log.py:verify_chain`).	Dockerfile `USER srasta` , K8s `securityContext.runAsUser: 1000` , compose `srasta-audit-init` chowns the volume.
tool-gateway → user workspace	Per-persona tool whitelist + path denylist + command allowlist + sandbox. No persona can run tools outside its policy.	`setup/tool_policy.yaml` (operator-configurable) → `tool-gateway/srasta_compliance.py` .
Operator (admin UI) → mutating operations	License-write block: hard-expired license or enterprise+trial blocks all non-safe HTTP methods (#169 Phase 4). /api/license itself is exempt for recovery.	`admin/main.py:_license_write_middleware` .
Audit log → external SIEM	Operator-configured; off by default. Vector-based forwarding with retry + DLQ.	`setup/vector/vector.toml` + `--profile audit` .
License JWT issuance (Gandiva-side)	Issued only by Gandiva ops via `license-server` admin endpoint with `X-Admin-Key` auth. Customer's only interaction is paste-key in admin UI.	`license-server/main.py` (Gandiva infrastructure, not in customer install).

Failure modes (what happens when...)

Scenario	Behavior
External LLM provider unreachable	LiteLLM returns upstream timeout; srasta-api surfaces 502 with audit entry. Customer's other models keep working. No platform-level failure.
Zitadel down	New OIDC logins fail with 503. Existing JWT-bearing requests continue (JWTs verify against cached JWKS). Admin UI surfaces a "managed-auth degraded" banner.
Audit-log export sink down	Vector buffers locally up to disk-budget limit; resumes on sink recovery. No platform-level failure. Hash-chain on disk preserved.
License hard-expired	srasta-api 402s on `/api/rag/` , `/api/tools/` , `/api/mcp/` , `/api/enterprise` . Admin UI write-block engages. `/api/license` exempt for paste-recovery. 60s revalidation picks up new key without restart.
Gandiva license-server unreachable	**No effect on running deployments** — license verification is RSA-offline. Only affects new license issuance, which is an out-of-band Gandiva-ops action.
Postgres unavailable	srasta-admin readiness probe fails; Zitadel can't authenticate. rag-api + tool-gateway continue to serve in-flight requests. Audit log keeps writing (file-based).
MinIO unavailable	Backups and audit-archive shipping pause. Live audit log keeps writing locally. Inference + tool calls unaffected.

Customer perimeter — what Srasta does NOT do

These are explicit non-properties of the architecture, listed so a security reviewer doesn't have to ask:

- **No customer data leaves the perimeter.** Operational telemetry (install events, feature-touch counts, error class counts — see the outbound-channels table above) is sent to Gandiva by default. Telemetry contains **counts and shapes only** — never prompt content, response content, document content, audit-log content, or end-user identifiers. Operators can disable telemetry entirely via `SRASTA_TELEMETRY=off` .
- **No SaaS multi-tenancy.** Each customer install is its own deployment. We don't share infrastructure across customers.
- **No data-plane routing through Gandiva.** Even if a customer uses an external LLM provider, the request flows customer-network → provider directly; not customer-network → Gandiva → provider.
- **No backup of customer data on Gandiva infrastructure.** Backups land in the customer's MinIO bucket or an operator- configured external S3 bucket. Gandiva has no path to the data.
- **No automatic update channel.** Updates are operator-initiated via `scripts/upgrade.sh` or Helm chart upgrade; the customer decides when (and whether) to apply each release.

Compliance posture references

- **CC6 — Logical Access** of the SOC 2 Common Criteria controls matrix (`docs/security/controls-matrix.md`) — every row in this diagram has a matching control.
- **CC6.6 — External-use access controls** is the single-gateway invariant (Mandate 1 of `docs/strategy/seed-momentum-plan.md`).
- **CC6.7 — Restricts data movement** maps to the trust boundary table above.
- **CC7.2 — Monitors components/operation** maps to the audit-log flow + canonical event taxonomy.

Roadmap dependencies

This document is a contract for the seed-stage architecture. As the following ship, this doc gets updated, not invalidated:

- **Phase H single-gateway authz** (post-seed) — collapses the three-copies-of-the-RBAC-matrix bug class without changing the external trust boundary.
- **Audit infrastructure Phase 1 schema migration** (post-seed) — consolidates the JSONL writer into a Postgres-backed canonical store; the trust boundary stays the same.
- **SSO federation wizard** (Milestone 4) — adds an in-product flow for operators to wire their IdP to Zitadel. No change to the data-flow diagram.
- **Embeddings TEI migration** (deferred) — replaces Ollama embeddings with TEI service. Same perimeter.

Discipline

Every change to the customer-data-flow surface must be reflected in this document **before** it ships. If the diagram and the running system disagree, the diagram is wrong and a security review will surface that gap. Mandate: **the architecture diagram matches reality, or the change does not ship**.

Related documents

- `docs/security/controls-matrix.md` — SOC 2 CC mapping that references this diagram.
- `docs/security/caiq-lite.md` — CAIQ Lite responses that reference this diagram. Lands with #158.
- `docs/legal/privacy-policy.md` — privacy policy that mirrors the data-flow assertions. Lands with #158.
- `docs/roadmap/platform-architecture-roadmap.md` — long-form architecture roadmap; this doc is the seed-stage one-pager that distills the customer-facing surface from that.
- `docs/strategy/seed-momentum-plan.md` — the anchor. This document closes seed-momentum item #5 (architecture + data-flow one-pager).