Sageio inside your own infrastructure.

Self-Hosted Deployment Guide

For organizations that can't put meeting data in a multi-tenant cloud. Sageio runs in your VPC, your data center, or your private cloud — under your security boundary, on your audit trail.
This guide is written for the infrastructure, security, and compliance teams evaluating a self-hosted deployment. It covers what self-host is for, what runs where, what your environment needs, how a rollout proceeds, and the support that comes with it.

---

1. When self-host is the right answer

Self-hosting exists for organizations whose requirements rule out SaaS — not as a default.

- **Data residency.** Regulators, customer contracts, or internal policy require meeting data to stay inside a specific country or network. Self-hosting puts the data exactly where it has to be — and never anywhere it shouldn't.
- **Compliance obligations.** Industries under FINRA, HIPAA, ITAR, or equivalent local regimes face audit requirements that SaaS rarely satisfies. Running Sageio in your environment keeps you in control of the evidence — and the auditor's questions.
- **An existing security perimeter.** Organizations that have invested years in network segmentation, an internal CA, and zero-trust tooling can have Sageio inherit that perimeter rather than punch holes through it.

If none of these apply, the multi-tenant cloud is simpler and faster to adopt. Self-host is for the cases where it isn't optional.

---

2. Reference architecture

Self-hosted Sageio is a containerized application that runs end-to-end inside your environment. AI inference, transcription, and storage all stay within your network boundary.

The core deployment includes:

- **Sageio application server** — the API and web application.
- **PostgreSQL database** — application data and workspace state.
- **Object storage** — audio and transcript artifacts (S3-compatible).
- **Inference layer** — transcription (STT) and translation.

You may **bring your own STT and LLM endpoints** (Deepgram, Azure OpenAI, Google Vertex, or comparable) or run Sageio's reference inference stack on local GPUs. Outbound network access is required only for license validation and optional telemetry — both can be tunneled through a customer-controlled proxy.

> The reference architecture is adapted per customer environment during scoping (Step 1 below).

3. System requirements

Area	Requirement
Orchestration	Kubernetes 1.27+ with Helm 3.12+. Docker Compose supported for proof-of-concept and single-host deployments.
Compute — application tier	8 vCPU, 16 GB RAM minimum per node. Two nodes recommended for high availability.
Compute — inference tier (optional)	NVIDIA GPU with 24 GB VRAM (A10G, L4, or equivalent) for self-hosted STT. Not required if using managed STT / LLM endpoints.
Database	PostgreSQL 14+. Customer-managed or Sageio-deployed. Minimum 100 GB storage, scaled to retention policy.
Object storage	S3-compatible (AWS S3, MinIO, Ceph, Azure Blob, GCS). Used for transcript exports and audio retention windows.
Network egress	Outbound HTTPS to a Sageio license endpoint required. All other traffic configurable, including through a customer proxy.

4. Deployment process

A typical self-host deployment runs **four to six weeks** from contract signature to the first translated meeting in production. Sageio assigns a dedicated solutions engineer for the duration.

1. **Scoping & environment review.** A joint working session covering your infrastructure, identity provider, network architecture, compliance requirements, and rollout plan. *Output:* a deployment design document, signed off by both teams before any installation work begins.
2. **Staging install.** Sageio installed in your staging or pre-production environment. Helm values configured for your storage, identity provider, and inference endpoints. End-to-end test meeting validated.
3. **Security & compliance review.** Your security team runs penetration testing, SBOM review, and any internal change-management process. Sageio supplies documentation, attestations, and engineering responses to findings.
4. **Production deployment & cutover.** Production install, identity federation enabled, monitoring connected to your observability stack. First production meetings run alongside your existing tooling until your team is confident.

5. **Handover & enablement.** Admin training, runbook walkthrough, and documentation handover. Ongoing support transitions to your assigned customer success engineer.

---

5. Support

Self-hosted deployments include solutions engineering before, during, and after rollout — not a Slack channel and a wiki, but a person who knows your environment.

- A **named solutions engineer** for deployment, and a **customer success engineer** for the duration of the contract.
- Standard support: business-hours response on weekdays, with a **four-hour acknowledgment SLA** on production-down issues. Extended-hours and 24×7 coverage available as add-ons.
- **Quarterly review calls** covering platform updates, security advisories, and the upgrade roadmap.
- **Major version upgrades** scheduled jointly and run against your staging environment first.

---

6. Next steps

Tell us about your environment and we'll send a scoping questionnaire. Most evaluations complete in about two weeks, including a working proof-of-concept in your staging environment.

Contact: sales@sageio.net
