

Votal Shield x Rafay

Partner handoff: multi-tenant AI guardrails

Goal. Let a Rafay customer turn on AI guardrails for their workloads, with each customer's policy isolated to that customer.

What Votal Shield does

Inspects AI prompts and model outputs inline and returns an allow, block, or redact verdict (PII and secrets, prompt injection, unsafe content, and agent or tool misuse), governed by per-tenant policy. One API call before the model, one after.

How the integration works

- One Rafay customer = one Shield tenant, identified by its own API key.
- Rafay provisions a tenant and key per customer when they enable guardrails.
- In Rafay's serving path: call Shield PreCall on the user's input, PostCall on the model's output, passing that customer's key (header X-API-Key).
- Shield resolves the tenant from the key and applies that customer's policy. No cross-tenant leakage, and no model keys are shared with Shield.

customer app → Rafay serving → [PreCall /guardrails/input] → model → [PostCall /guardrails/output] → user
Deploys two ways, same API: hosted (api.guardrails.votal.ai) or in-cluster (Shield container in the customer's cluster for data residency; Kubernetes, Helm, and OpenShift references are included).

What is in this handoff

File	Purpose
00 START HERE (this page)	The one-page intro.
01 Integration guide	Full engineering detail: auth, onboarding, runtime calls with schemas, telemetry, in-cluster, OpenShift.
02 OpenAPI (openapi-partner.json)	The API contract to import for client codegen.
openshift/ manifests	Ready to apply: guardrail server with in-cluster vLLM (GPU), and Redis.
Sandbox tenant and key	Sent separately; integrate and validate before go-live.

Get started (about a day to first green test)

- 1. Import the OpenAPI and generate a client.
- 2. On enable guardrails, provision the customer's tenant and key, and store the key.
- 3. Wire PreCall before the model and PostCall after; honor the verdict and any sanitized output.
- 4. Validate against the sandbox: a benign prompt passes, a policy-violating prompt returns a block, and the decision appears in telemetry.

Who provides what

- Votal provides: the admin and provisioning key, a sandbox tenant and key, base URLs, the OpenAPI and integration guide, and (if in-cluster) the Helm and OpenShift manifests plus model sizing.
- Rafay builds: the per-customer onboarding hook, the PreCall and PostCall calls in the serving path, and either an embedded Shield policy portal or policy screens on the tenant API.

One deployment setting that matters

Run the Rafay-facing Shield with **SHIELD_GUARD_REQUIRE_KEY=enforce** so a missing or invalid customer key is refused rather than screened without policy. This is what keeps tenants isolated.

Next step: Votal issues the sandbox tenant and admin key and confirms hosted vs in-cluster; Rafay runs the four-step get-started against the sandbox.