Hybrid Infrastructure

Questions this doc answers

- How do you mix hosted connectors with private on-prem models?
- What happens when one team needs cloud AI (ChatGPT/Claude) and another uses internal models?
- How do you keep AI egress under control across those boundaries?

Shared memory across hybrid stacks

Reflect Memory abstracts the infrastructure tier from the AI tier. A hosted deployment can still connect to your on-prem models via `RM_ALLOWED_MODEL_HOSTS`, while self-host deployments can still serve ChatGPT/Claude via MCP if agent keys are available.

- `disableModelEgress` (default `true` in self-host) prevents outbound LLM calls unless explicit.
- `requireInternalModelBaseUrl` ensures every hosted model call resolves to `allowedModelHosts`.
- `allowedModelHosts` also gates scheduled webhooks and automation triggers that otherwise would reach public endpoints.

Model connectivity patterns

1. **Cloud-first + self-host complement:** Use hosted API for general reads/writes, and self-hosted connectors for highly regulated data. Both sides read/write the same memory store, so context replicates naturally.
2. **On-prem model bridging:** Install Reflect Memory inside your VPC. Set `RM_ALLOWED_MODEL_HOSTS=llama.local,llama.next`, `RM_REQUIRE_INTERNAL_MODEL_BASE_URL=true`, and `RM_DISABLE_MODEL_EGRESS=true`. Your LLMs call the local REST API; the same environment also accepts MCP keys from corporate-approved tools.
3. **Hybrid autop-run:** Use the MCP server to let GPUs in your datacenter talk to Reflect Memory. Agents like Claude or Grok can reach internal memory via the same `RM_PUBLIC_URL` referencing your reverse proxy.

Connectivity guardrails

- `enforceModelHostPolicy(baseUrl)` compares URLs to `RM_ALLOWED_MODEL_HOSTS`.
- Private deployments still support optional `RM_ALLOW_PUBLIC_WEBHOOKS` for integration with approved partners.
- Telemetry and billing events track cross-boundary usage so you can see if a self-hosted team is hitting cloud connectors or vice versa.