Deployment Architecture

Questions this doc answers

- Which `RM_DEPLOYMENT_MODE` should I choose?
- How does Reflect Memory enforce private network boundaries and model egress policies?
- What guardrails exist for allowed model hosts, webhooks, and SSO in self-hosted deployments?

Deployment modes table

Mode	Ownership	Network boundary	Model egress	Public webhooks	Typical guardrails
hosted	Reflect Memory cloud	public	optional	allowed	Standard rate limits, telemetry, SOC 2 API
isolated-hosted	Dedicated runtime/DB per tenant	managed/public	configurable	restricted (per tenant)	Same infra plus tenant-level logging
self-host	Your VPC/air-gapped network	private	disabled by default	blocked	`RM_ALLOWED_MODEL_HOSTS`, `RM_REQUIRE_INTERNAL_MODEL_BASE_URL`, `RM_DISABLE_MODEL_EGRESS`

Resolve deployment config

`resolveDeploymentConfig()` centralizes the runtime flags:

- `RM_DEPLOYMENT_MODE` → `"hosted" | "isolated-hosted" | "self-host"` (default `hosted`)
- `RM_DISABLE_MODEL_EGRESS` → favors `true` in `self-host`
- `RM_REQUIRE_INTERNAL_MODEL_BASE_URL` → ensures internal models are reachable
- `RM_ALLOWED_MODEL_HOSTS` → comma-delimited whitelist enforced via `enforceModelHostPolicy`
- `RM_ALLOW_PUBLIC_WEBHOOKS` → defaults to `false` in air-gapped mode
- `RM_SSO_*` → toggles OIDC authentication; missing `JWKS`, `ISSUER`, or `AUDIENCE` ✅ fails startup

Validation is strict: self-host mode with `requireInternalModelBaseUrl` requires a non-empty `allowedModelHosts`. SSO enables per-tenant JWKS validation and email claim resolution for user lookups.

Network boundary enforcement

Every deployment config exposes `networkBoundary` (`"public"` vs `"private"`). Self-host defaults to `"private"`. In addition:

- `requireInternalModelBaseUrl` forces every external LLM call to be proxied through your internal gateway.
- `allowedModelHosts` ensures only approved models (e.g., `llama.local`, `ollama.company`, `vicuna.private`) can be reached.
- `disableModelEgress` defaults to true for `self-host`, preventing outbound connections unless explicitly lifted.

Production hosting (Reflect Memory cloud)

Reflect Memory production runs on **self-managed dedicated infrastructure** (Vultr), not third-party PaaS platforms like Railway or Vercel. GitHub Actions rsyncs the dashboard and API to `/opt/reflect/` on the host and runs `deploy.sh`; Caddy terminates TLS for `reflectmemory.com`, `dev.reflectmemory.com`, and `api.reflectmemory.com`. Enterprise buyers evaluating data residency should treat this as Reflect-owned cloud — the same deployment modes (`hosted`, `isolated-hosted`, `self-host`) apply whether you use our cloud or run the container inside your network.

Pilot & upgrade flow

1. Scope call → determine `mode`, SSO/JWKS, `allowedModelHosts`, compliance requirements.
2. Deploy pilot container/Helm chart with `RM_TENANT_ID`, `RM_SSO_*`, `RM_ALLOWED_MODEL_HOSTS`, `RM_AGENT_KEY_*`.
3. Connect AI tools via MCP (agent keys) or REST (API key).
4. After pilot, flip `RM_REQUIRE_INTERNAL_MODEL_BASE_URL` / `RM_DISABLE_MODEL_EGRESS` toggles as needed, then monitor audit trail + usage events for compliance.