

ENTERPRISE AGENTIC REFERENCE ARCHITECTURE · V1.0

The future of engineering is more than a copilot — it's an **architecture.**

Seven layers, every one pluggable. Nine principles. A vendor-neutral framework for running AI agent fleets under enterprise governance — **built on the SDLC, designed to apply to any domain.** One view of where this is heading; offered to be challenged.

Enterprise Agentic Reference Architecture

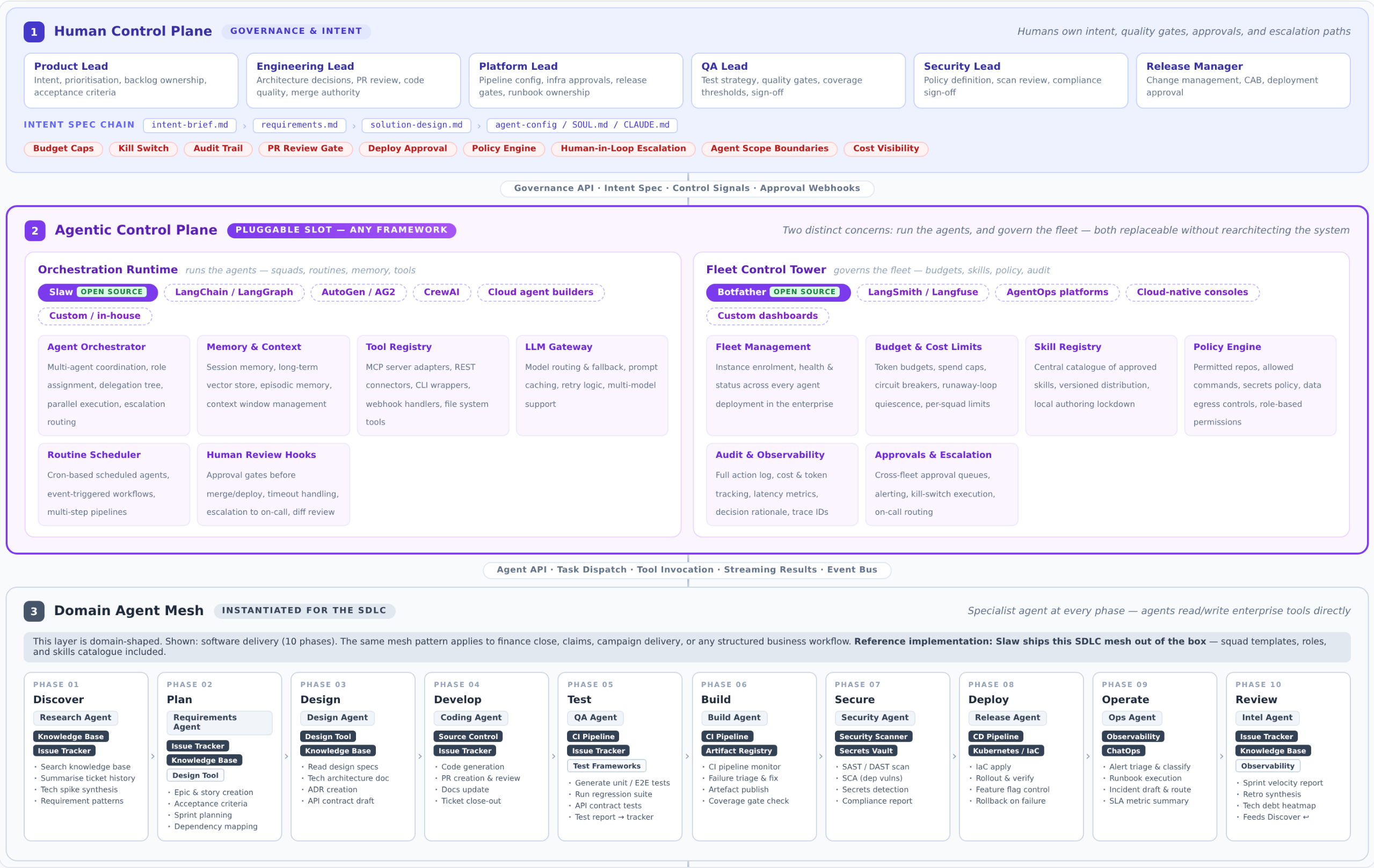
Version 1.0 · June 2026

Seven layers. Every one pluggable. **Humans govern, agents execute** — the SDLC today, any domain tomorrow.

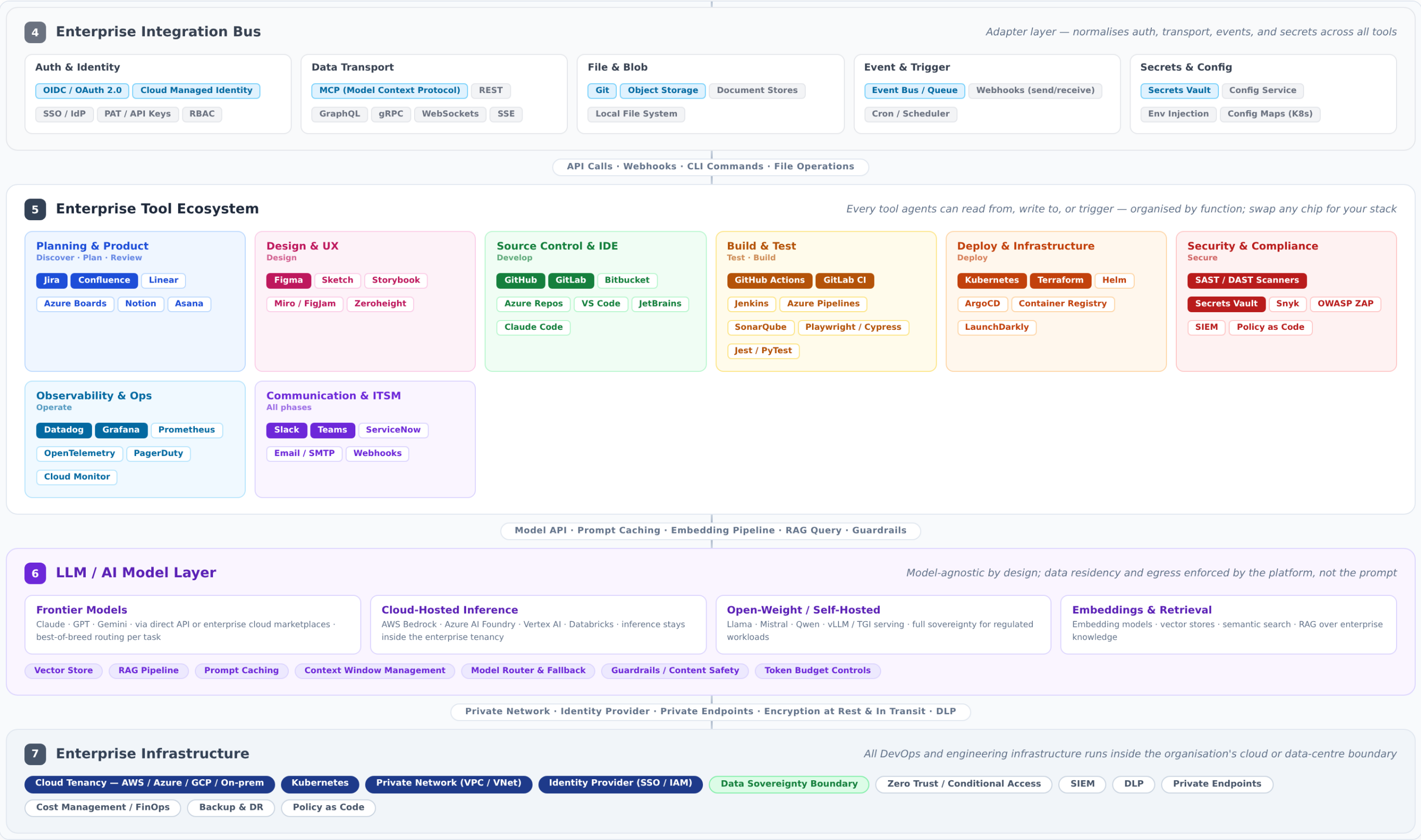
Framework-agnostic · Vendor-neutral

01	Human Control Plane Humans own intent, quality gates, approvals — and the kill switch	Intent specs · Approval gates · Budget caps · Audit trail
02	Agentic Control Plane The pluggable slot — swap frameworks without rearchitecting	runs the agents Slaw OPEN SOURCE + governs the fleet Botfather OPEN SOURCE or LangGraph · CrewAI · your own
03	Domain Agent Mesh A specialist agent at every phase — the SDLC mesh ships with Slaw out of the box	Discover > Plan > Design > Develop > Test > Build > Secure > Deploy > Operate > Review ↔
04	Integration Bus Normalises auth, transport, events & secrets across every tool	MCP · REST · Webhooks · OAuth / OIDC · Vault
05	Enterprise Tool Ecosystem Agents work in the same tools as humans — no parallel universe	Issue tracker · Source control · CI / CD · Security · Observability
06	LLM / Model Layer Model-agnostic; sovereignty enforced by platform, not prompts	Frontier API · Cloud-hosted · Self-hosted open weights
07	Enterprise Infrastructure Your tenancy, your network, your data sovereignty boundary	Cloud / on-prem · Kubernetes · IAM / SSO · Zero trust

Layers 01-03 — Governance, Control Plane, Agent Mesh



Layers 04-07 — Integration, Tools, Models, Infrastructure



THE OPERATING RULES

Nine Principles of Enterprise Agentic Engineering

01

Humans own intent; agents own execution

Agents act only downstream of an explicit, human-authored intent spec.

02

The control plane is pluggable

Swap frameworks without rearchitecting governance above or integrations below.

03

Orchestration ≠ fleet governance

Running agents and governing them at scale are distinct concerns, distinct tools.

04

A gate at every consequence

Anything irreversible pauses for a named human role. Gates are architecture, not etiquette.

05

Cost is a first-class control

Budgets, circuit breakers and kill switches designed in from day one — not a bill at month end.

06

Agents work in the same tools as humans

No parallel agent universe — same tickets, same PRs, same trail.

07

Every action is attributable

Logs, rationale, trace IDs. If you can't replay it, it shouldn't have run.

08

Model-agnostic, sovereignty-aware

Any model, any host — residency enforced by the platform, never by prompts.

09

The loop closes itself

Review feeds Discover. Agents that don't learn from the last cycle are expensive scripts.

Explore the architecture. Challenge it. Or **try it** — both halves are open source.

ORCHESTRATION RUNTIME — RUNS THE AGENTS

Slaw

Agent squads across the full SDLC: roles, routines, persistent memory, MCP tool access, human review hooks.

FLEET CONTROL TOWER — GOVERNS THE FLEET

Botfather

Fleet enrolment, token budgets & circuit breakers, central skill registry, audit, approvals, kill switch.

```
$ npx @slaw-ai/slaw onboard
```