AVIATRIX

SECURITY ARCHITECTURE BRIEF

Contain OpenClaw, NemoClaw, and Hermes Agents

Powered by Aviatrix Cloud Native Security Fabric

Threat model, control boundaries, audit evidence, and rollout model for enterprise agent runtimes

EXECUTIVE SUMMARY

Executive summary. Agent Harness technologies like OpenClaw, Hermes, and NemoClaw create persistent, tool-using, credential-bearing agent runtimes. This architecture assumes one runtime may be manipulated. Aviatrix reduces Blast Radius by enforcing destination policy outside the runtime and recording independent per-connection evidence.

Threat Model

The Containment Era operating assumption: something in the environment is already compromised, and the Blast Radius of that compromise is decided by architecture, not by detection speed. Agent Harness technologies like OpenClaw, Hermes, and NemoClaw create new risks.

Threat class	Enterprise scenario	Containment response
Prompt injection	Untrusted content tells the agent to send retrieved data to a hostile URL.	Default-deny egress blocks the unlisted destination and logs the rule hit.
Supply-chain / skill compromise	A package, plugin, or skill phones home or downloads a payload.	Unknown host is denied; package access is limited to approved classes and endpoints.
Sensitive data disclosure	The agent combines private data with an unapproved model or SaaS call.	Only sanctioned model gateways, SaaS APIs, and MCP gateways are reachable.
Excessive agency / shadow model	Agent code or manipulated instructions call an unapproved model provider.	A named deny rule for unapproved model providers fires before allow rules.

DNS exfiltration	Runtime sends encoded data to an external DNS resolver.	Allow VPC resolver first, then deny external UDP/TCP 53.
Lateral movement	Compromised agent reaches adjacent workloads or production systems.	Optional SmartGroup east-west microsegmentation; not required for the first egress blueprint.

This review covers the threat model, control boundaries, kill chain, and evidence model for the OpenClaw Agent Runtime Containment on AWS. The trusted enforcement boundary is the Aviatrix Spoke Gateway and Distributed Cloud Firewall (DCF) policy, not the agent runtime. The agent VM is treated as a potentially compromised non-human operator with delegated access.

Attack Scenario and Kill Chain

Stage	What happens	Control evidence
Initial manipulation	Prompt, web page, ticket, email, or document changes agent intent.	Prompt guardrails and app logs may flag it; containment assumes they can miss.
Tool execution	Agent reads data, opens a shell, calls a browser, invokes a skill, or installs a package.	Runtime sandbox and host telemetry provide local context.
Network attempt	Agent tries to connect to an unapproved host, shadow model, external resolver, or adjacent workload.	DCF permits or denies by ordered policy; FlowIQ records source, destination, rule, action, time.
Impact path	Data would leave the VPC or the agent would pivot east-west.	Blocked egress and optional microsegmentation limit Blast Radius before detection speed matters.

Enforcement Architecture

The Validated Containment Architecture is one layer in a defense-in-depth model. It does not replace the controls above it; it guarantees that a bypass of any of them still cannot reach an arbitrary destination.

Control	Runs where	Why Aviatrix still matters
Prompt / model guardrails	Model and application layer	A successful bypass still should not be able to reach arbitrary destinations.

OpenClaw / NemoClaw sandbox	Agent runtime boundary	Enterprise teams still need independent VPC routing, FlowIQ evidence, and fabric-wide consistency.
IAM and SaaS permissions	Identity providers and APIs	They do not stop traffic to attacker infrastructure or unsanctioned endpoints.
Aviatrix DCF	Cloud network path	Outside the compromised runtime and visible to platform and security teams.

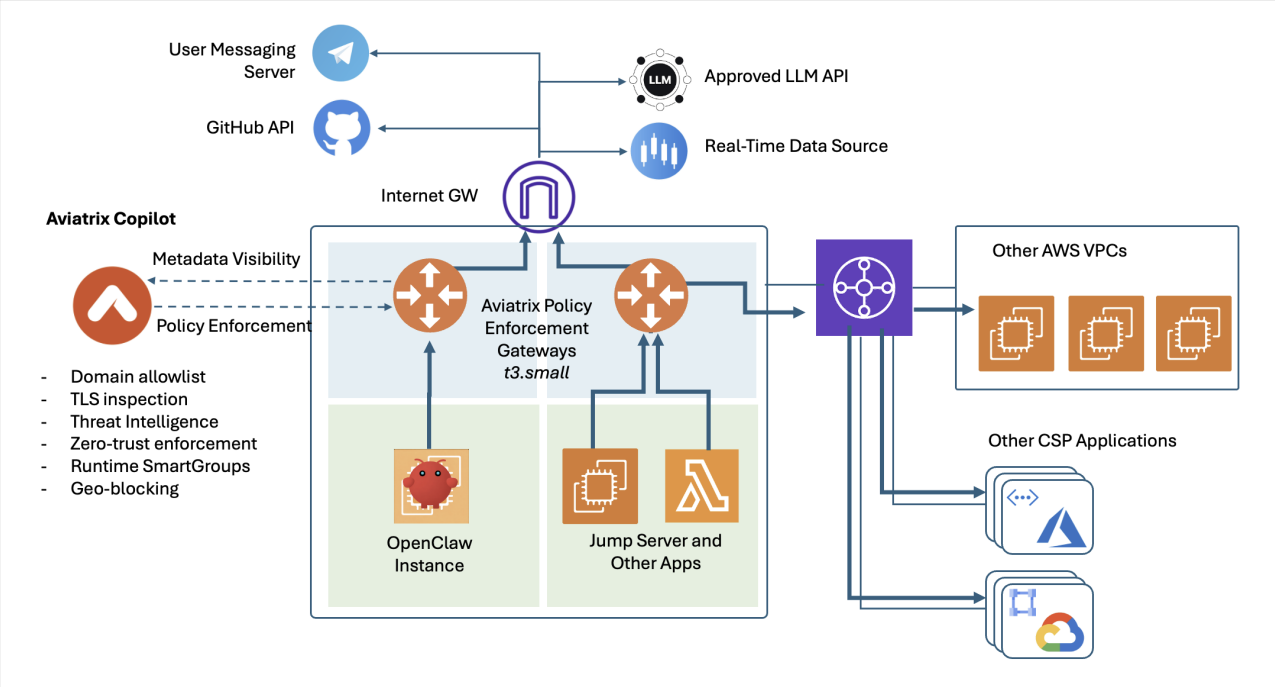

Figure 1. Reference insertion pattern for AWS agent runtime egress containment.

The goal of this Validated Containment Architecture is not to replace prompt guardrails, sandboxing, model safety, IAM, or MCP/tool authorization. The goal is to add an independent network control the agent cannot bypass from inside the harness. If a prompt injection, malicious skill, or compromised dependency attempts to connect to an unapproved destination, the socket is blocked before data leaves the VPC – and the attempt is logged with an attributable rule name.

Architectural Boundaries

Out of Scope	What Governs it Instead
Prompt content safety	Handled by application guardrails, model safety, red teaming, and prompt/tool policy. This blueprint does not inspect prompt semantics.
Tool authorization	Handled by MCP gateway policy, SaaS scopes, and IAM. Aviatrix governs network reachability to those systems.

Local sandbox escape	Handled by host hardening, OpenClaw/NemoClaw sandbox controls, patching, EDR, and least-privilege host config.
Developer workstations	Strongest containment applies to cloud VPCs and Kubernetes where routing is deterministic; workstation use needs a managed device and enterprise egress path.
Full east-west design	An Aviatrix fabric extension. The AWS blueprint only needs egress control to be useful on day one.

Compliance Evidence

Compliance reviewers need proof that the control operated, not just a policy statement. CoPilot FlowIQ provides runtime-independent evidence that can support reviews for SOC 2, HIPAA, PCI-DSS, FedRAMP, ISO 27001, DORA, NIST AI RMF, and EU AI Act governance programs.

Beyond log tables, the same data renders as a real-time topology map. CoPilot draws the live fabric – the agent spoke, the Aviatrix Spoke Gateway, and each destination – and overlays per-connection flows as they occur. Permitted flows resolve to declared WebGroups; denied attempts remain on the map attributed to a named rule and the default-deny action. Reviewers can watch enforcement operate during a test window and export the view as point-in-time evidence.

Evidence artifact	What it proves
Allowed-flow logs	Approved workflow destinations were reached through the declared SmartGroup / WebGroup policy.
Named deny logs	Shadow-model, DNS-exfiltration, or default-deny attempts were blocked and attributed to a readable rule name.
Real-time topology map	A live CoPilot fabric view showing the agentic application egressing only through the Aviatrix Spoke Gateway, with permitted flows resolving to declared WebGroups and denied attempts attributed to a named rule, in real time and exportable as point-in-time evidence. (East-west segmentation between workloads is out of scope for this egress blueprint.)
Terraform policy history	Destination changes were peer-reviewed, versioned, and tied to an agent-class owner.
Route-table evidence	The agent subnet default route sends traffic through the Aviatrix Spoke Gateway, not directly to the internet.
Optional east-west logs	If added later, SmartGroup segmentation can prove the agent VPC could not pivot to other spokes.

Request a 30-minute architecture review.

We walk through the enforcement model in your environment, map your current your agent runtime / agent harness, and identify the paths you currently do not see.

About Aviatrix

Aviatrix® is pioneering the Cloud Native Security Fabric – the architecture the Containment Era requires. The Cloud Native Security Fabric governs every workload communication path across every cloud, every VPC, every Kubernetes cluster, and every serverless function, from a single policy engine. Universal propagation. Enforced at the workload during runtime. Trusted by more than 500 of the world's leading enterprises. For more information, visit aviatrix.ai.