AVIATRIX®

SOLUTION BRIEF

Contain Enterprise MCP Infrastructure

Powered by Aviatrix + Obot

Aviatrix Validated Containment Architectures are lab-tested containment deployment blueprints for the AI platforms enterprises are actually running. Ship-ready, policy-included, validated before they arrive. This Validated Containment Architecture covers Enterprise MCP Infrastructure with Obot.

The Threat

MCP servers often operate as egress paths that existing controls weren't designed to govern. Kubernetes NetworkPolicy, service meshes, and audit logs typically reason about identity and traffic shape rather than per-server destination policy, which leaves a visibility gap worth closing.

MCP servers commonly combine three properties that warrant attention together: access to private data, exposure to untrusted content, and outbound network reach. Constraining the third at the network layer materially reduces the blast radius of the other two. Recent supply-chain incidents including the March 2026 wave affecting Axios (70 million weekly downloads), LiteLLM, Trivy, KICS, and Telnyx illustrate how a compromised dependency can inherit a server's egress permissions. Domain-based enforcement is well-suited to this case because it evaluates the destination rather than the caller's identity.

A SCENARIO WORTH PLANNING FOR

An MCP server is permitted to send sensitive business data to an approved internal system of record. A poisoned npm dependency attempts to exfiltrate the same data to an attacker-controlled domain. The server has permission for the action, not the destination. Aviatrix sees the destination, applies the FirewallPolicy CRD, and blocks the egress before it leaves the environment.

The Architecture

Obot governs which MCP servers run and how they're configured. Aviatrix governs where those servers can reach. Two decisions. Two enforcement points. One posture.

Component	What It Does
Obot controller	Deploys MCP servers via Helm. Sets egress permissions per server. Triggers automatic cleanup when a server is deleted.
FirewallPolicy CRDs	One per MCP server. Permits declared domains; default-denies everything else. Lives in git. Ships in the same PR as the deployment manifest.
Aviatrix Spoke Gateway	Evaluates policy inline at the eBPF dataplane. Drops disallowed egress before it leaves the environment. No sidecar. No per-pod overhead. Validated on AKS and EKS (GKE validation in progress). Extends to on-premises Kubernetes via Aviatrix Edge.
Default-deny posture	When deployed with the reference architecture, a server with no declared domains has no egress. Containment is the default, not the exception.

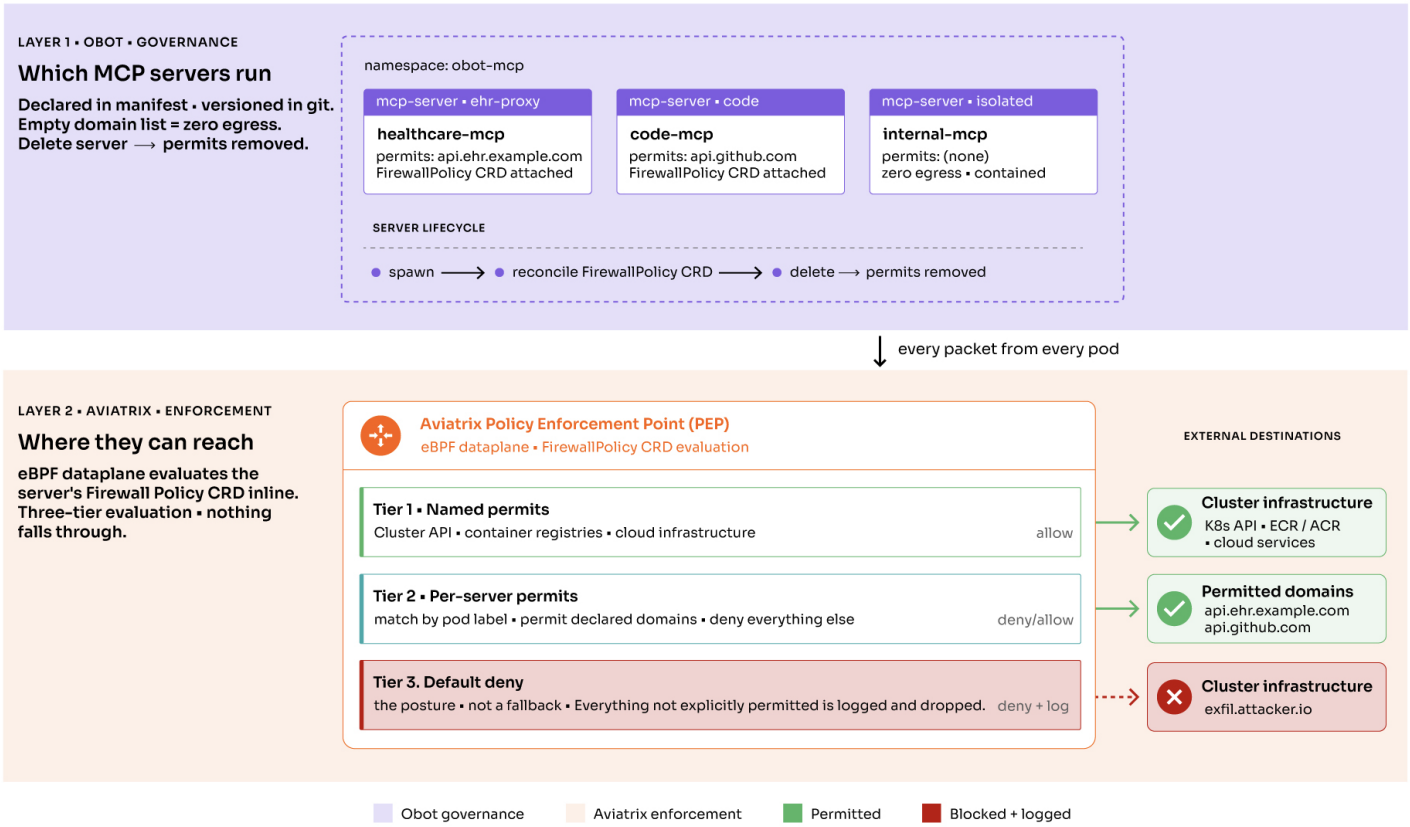
How it works in practice

REFERENCE ARCHITECTURE • CONCEPTUAL

Two-layer enforcement Obot governs what runs, Aviatrix governs where they reach

Solution Brief • Security Architecture Brief

AVIATRIX + obot



Three Things Your Stack Can't Do

- Granular containment per server: two MCP servers in the same Kubernetes namespace can carry different egress policies based on pod labels, a level of granularity beyond what NetworkPolicy was designed for.
- Zero policy debt: delete an MCP server and its egress permissions are automatically removed. No orphaned permits accumulate.
- Multicloud portability: the same policy enforces egress across AKS and EKS without per-cloud rewrite. GKE validation in progress; extends to on-premises Kubernetes via Aviatrix Edge.

Compliance Evidence

For HIPAA, PCI-DSS, SOC 2, and ISO 27001 environments, post-incident reviews now expect architectural evidence of enforcement alongside written policy.

FirewallPolicy CRDs are Kubernetes objects. They live in git, inspected with kubectl, and tracked in the Kubernetes audit log. Security policy ships in the same repository as the deployment manifest, reviewed in the same PR, deployed in the same pipeline. That is your proof of enforcement.

Questions Worth Asking

- How many MCP servers are running in your environment right now, and how many are fully inventoried?
- If a compromised dependency inside an MCP server attempted to exfiltrate data to an external domain, what in your stack would catch it?
- When you delete an MCP server today, do you know whether its network permits are also removed?
- How are you currently proving to auditors that AI workload egress is controlled, not just described in policy?

What's Included

Deliverable	Detail
Insertion pattern	Transparent to applications and developers: no code changes required
FirewallPolicy CRDs	As code: ships in GitHub repo, deploys via Helm
SmartGroup model	AI-aware tagging keyed to MCP server pod labels
Default-deny policy pack	Baseline egress posture for AKS and EKS

Deployment guide	Lab-validated reference architecture
30-day free trial	Trial environment sized for your Obot deployment

Get Started

Request a 30-minute architecture review. We walk through the policy model in your environment, help inventory the MCP servers running today, and surface the egress paths worth bringing under explicit policy.

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 plane. One rule. Universal propagation. Enforced at the workload, not at a chokepoint. Trusted by more than 500 of the world's leading enterprises. For more information, visit aviatrix.ai.