AVIATRIX®

SOLUTION BRIEF

Secure Enterprise AI Chat

Powered by Aviatrix — LibreChat on Kubernetes Edition

Aviatrix Validated Containment Architectures are lab-tested containment deployment blueprints for the enterprise platforms you are actually running. Ship-ready, policy-included, validated before they arrive. This Validated Containment Architecture enforces network containment for LibreChat on Kubernetes – with the network policy derived from the application's own configuration.

The Threat

Enterprises stand up self-hosted AI chat so sensitive data stays in their environment. LibreChat – the most widely deployed open-source enterprise chat platform – delivers that promise at the application layer. What it does not ship with is an enforcement layer beneath it, and neither does Kubernetes networking by default.

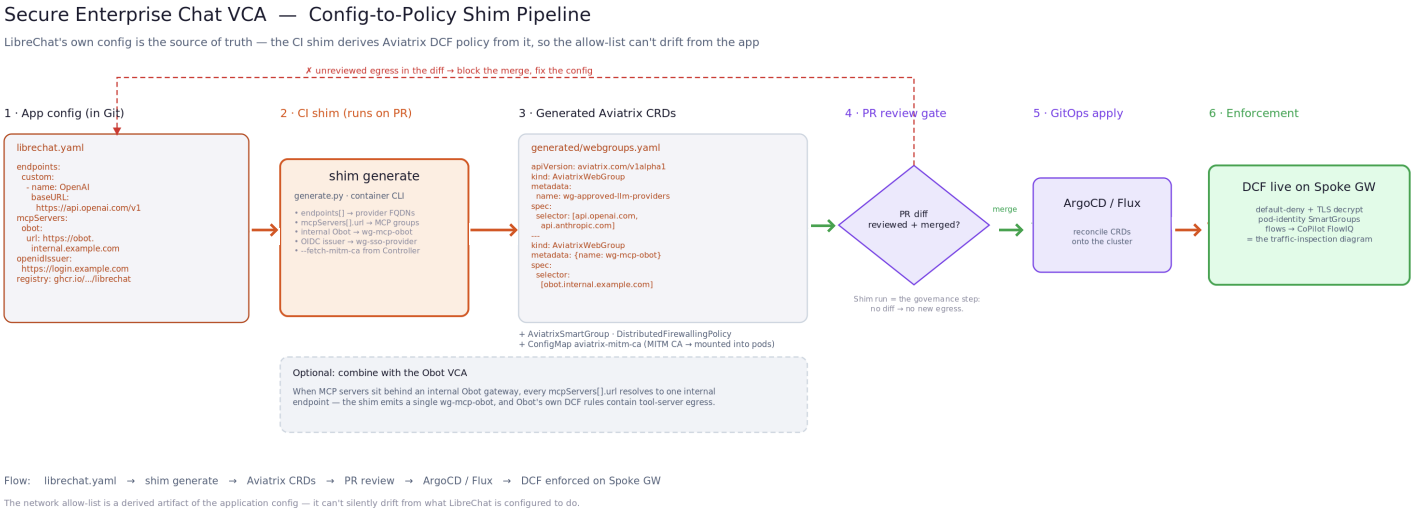
The result is a wider boundary than most teams realize. By default, any large language model (LLM) provider an administrator adds to the configuration can be called immediately; any Model Context Protocol (MCP) tool server can reach any destination on the public internet; and because LibreChat is open-source software with a large dependency tree, any compromised component – a tampered container image, a poisoned package, a hijacked Helm chart – can open a socket to an attacker-controlled host. Constraining that reach at the network layer materially reduces the Blast Radius of a misconfiguration or a supply-chain compromise. Domain-based enforcement is well-suited to this case because it evaluates the destination rather than the caller's identity – blocking exfiltration regardless of which process initiated it.

The Architecture

Aviatrix governs what the LibreChat stack can reach. The Validated Containment Architecture is a policy overlay that attaches to an existing Aviatrix-secured Kubernetes cluster and enforces default-deny network containment – locking LLM provider egress to approved providers, scoping each MCP server to its authorized interfaces, and isolating the conversation data store. Enforcement occurs at the network, outside the application trust boundary, so a compromised dependency cannot disable it. No new network infrastructure is provisioned, and the policy is derived from LibreChat's own configuration.

How it works in practice

Coverage	Enforcement Method	Bypass Risk	Status
LibreChat on EKS, AKS, or GKE	Policy overlay; network policy derived from librechat.yaml by a CI shim	Zero (network-layer; below the application)	Architecture Validated



Three Things Your Current Stack Can't Do

01 Network policy derived from the application's own config

A continuous integration (CI) pipeline shim reads LibreChat's configuration file and generates the exact Aviatrix Kubernetes manifests that encode it as enforceable policy. Add a provider, and a new destination group appears in the diff; add an MCP server, and a new per-server allow-list appears – reviewed in the same pull request. Policy and config stay in lockstep instead of drifting apart as two separately maintained files.

02 Supply-chain containment below the application

The registry allow-list permits image pulls only from the exact Open Container Initiative registries the LibreChat chart uses; the default-deny catch-all blocks everything else before the socket opens. A compromised dependency that tries to beacon outbound hits the deny rule. No application-layer control catches a compromise beneath itself – the network layer does, shrinking the Blast Radius of an open-source supply-chain incident to nothing reachable.

03 Enterprise security fabric integration

LibreChat egress policy lives alongside production workload policy, continuous integration and continuous delivery policy, and Model Context Protocol server policy in a single Aviatrix Distributed Cloud Firewall plane. Policy is keyed to Kubernetes pod identity, so it survives pod restarts, scaling, and MCP sidecar churn. CoPilot provides unified visibility across the chat stack and the rest of the estate from one console. No standalone AI gateway provides this without adding a new external dependency in the egress path.

Compliance Evidence

For SOC 2, HIPAA, PCI-DSS, FedRAMP, and NIST Secure Software Development Framework environments, post-incident reviews now expect architectural evidence of network egress enforcement for AI workloads – not just written policy.

Aviatrix Distributed Cloud Firewall WebGroup and SmartGroup configurations are Terraform-managed infrastructure-as-code, derived by the shim from the LibreChat configuration and reviewed in the same pull request as the application change. CoPilot logs every allowed and denied connection with source identity, destination fully qualified domain name, and timestamp – continuous audit evidence, not snapshots. With transparent Transport Layer Security (TLS) decryption active, CoPilot shows Uniform Resource Locator paths, scoped only to permitted connections.

Questions Worth Asking

- You deployed LibreChat to keep data in your environment – do you know every LLM provider and destination it is actually allowed to reach today, or just the ones you intended to configure?
- If a compromised npm package or container image in the LibreChat stack tried to beacon to an attacker-controlled host, what in your current stack would block it before the socket opens?
- When an administrator adds a new LLM provider to the configuration, does a corresponding network rule get reviewed in the same change – or does the network policy drift behind the config?
- How are you currently proving to auditors that AI chat egress is controlled, not just described in policy?

What's Included

Component	Description
Configuration Shim	Open-source CI pipeline step that reads librechat.yaml and generates Aviatrix Kubernetes custom resource definitions for WebGroups and Distributed Cloud Firewall rules – LLM provider egress, per-MCP allow-lists, registry allow-list, and default-deny base.
Terraform Policy Overlay	Cloud-agnostic Aviatrix provider resources that attach to an existing EKS, AKS, or GKE cluster. The only cloud-specific parameter is the container registry allow-list. Deploys in under twenty minutes; no new network infrastructure.
Transparent TLS Decryption	Aviatrix certificate loaded into the LibreChat container via a configuration mount – no image rebuild – so the gateway decrypts permitted LLM traffic inline for URL-path visibility. This is the insertion point for Aviatrix AgentGuard inspection as those capabilities ship.

CoPilot Dashboard	Chat-stack traffic visibility with every allowed and denied connection logged by source identity and destination fully qualified domain name, plus URL paths when decryption is active.

Get Started

Request a 30-minute architecture review

We run the shim against your actual LibreChat configuration, show you the network policy it derives, and surface any provider or tool path you did not expect to be there.

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.