AVIATRIX

TECHNICAL BRIEF

Contain Enterprise AI Agents

Practitioner reference for compiling Microsoft Agent Control Specification policy into network enforcement

Aviatrix Distributed Cloud Firewall · Reference Architecture for Microsoft Agent Control Specification

Threat Context

An AI agent reaches Model Context Protocol (MCP) servers, large language model (LLM) APIs, and internal endpoints over the network. Agent governance frameworks define what an agent may reach in code, but cannot guarantee every agent process loads and honors that policy – vendor-shipped, legacy, and compromised agents bypass it. Microsoft's Agent Control Specification (ACS) provides a portable policy contract; this architecture compiles that contract into Aviatrix Distributed Cloud Firewall (DCF) rules so the policy is enforced at the network, where every agent's traffic must pass.

LAB-VALIDATED THREAT SCENARIO

An agent workload with no ACS integration attempts a tool call to an MCP server not on its permitted list. With the agent scoped by a SmartGroup and its permitted endpoints defined as WebGroups under a default-deny posture, DCF evaluates the connection, matches no allow rule, blocks it, and logs the attempt in CoPilot – with no awareness or cooperation required from the agent process.

Insertion Pattern

Enforcement is inserted at the network boundary, in the agent workload's egress path, not inside the agent. The same model applies across runtimes; only the gateway attachment point differs.

1. Deploy an Aviatrix Spoke Gateway into the VPC or virtual network where agent workloads run, so their egress routes through it.
2. Run the plugin (agent-shield-to-dcf) against the ACS guardrails file to generate SmartGroups, WebGroups, allow/deny rules, and inline pattern guards.
3. Apply the generated policy through the Aviatrix REST Application Programming Interface (API) or the Terraform provider.
4. DCF evaluates every outbound agent connection against the default-deny posture before it leaves the environment.

No sidecar, no mesh dependency, no agent code changes. Because enforcement is a property of the network path, there is no process inside the agent for a compromise to disable.

Coverage across runtimes

- Kubernetes (any cloud) – cluster VPC or virtual network attaches to the Spoke Gateway; pod egress routes through it at the node subnet level.
- Virtual machines – workloads in a private subnet whose default route points to the gateway.
- Serverless functions (AWS Lambda, Azure Functions, Google Cloud Run) and managed cloud agent platforms (Amazon Bedrock AgentCore, Azure AI Foundry, Vertex AI Agent Engine) – egress governed at the gateway serving their network.
- On-premises workloads served by an Aviatrix gateway.

Prerequisites

Before compiling and applying policy, verify the following are in place:

- Aviatrix Controller and CoPilot deployed and reachable from the target cloud accounts; a standard DCF deployment (no additional Aviatrix products required).
- An Aviatrix Spoke Gateway in each VPC or virtual network hosting agent workloads, with agent egress routed through it.
- A valid ACS guardrails policy file for each agent (or agent group) to be governed.
- A tool-to-host registry (operator-supplied) or gateway inspection enabled, so tool names in the ACS resource catalog can be bound to destination hosts for WebGroup construction – ACS does not carry FQDNs.
- Workload tagging conventions agreed in advance so SmartGroups can resolve agent identity by tag, label, or namespace.
- REST API access configured for the plugin to apply generated policy as infrastructure-as-code.

> **STDIO TRANSPORT NOTE**
>
> Agents that talk to a local MCP server over standard input/output (stdio) generate no network traffic and are invisible to DCF. This is an architectural limitation of stdio transport. Recommendation: deprecate stdio MCP in governed estates so all tool calls traverse the network and are subject to policy.

SmartGroup and WebGroup Layout

Policy follows agent identity, so the grouping model is the foundation of the deployment. Define one egress profile per agent or agent group so each policy subject maps to a minimal set of permitted endpoints.

Object	Type / scope / purpose
Agent source SmartGroup	Matches the agent workload by cloud tag, Kubernetes label, or namespace. One SmartGroup per agent or agent group; policy follows the workload as it scales or moves.
MCP server WebGroup	Permitted MCP server destinations for the agent, classified by fully qualified domain name and Uniform Resource Locator (URL) path from the ACS resource catalog (via the tool-to-host registry).
LLM API WebGroup	Permitted LLM endpoints for the agent (for example, the provider APIs the agent is authorized to call).
Endpoint WebGroup	Permitted internal or external Hypertext Transfer Protocol (HTTP) endpoints from the resource catalog.
Default-deny rule	The base posture for the agent SmartGroup: any destination not matched by a permit WebGroup is denied and logged.

Policy Evaluation Tiers

DCF evaluates agent egress in tiers, from coarse default-deny to deeper inspection. Apply progressively – baseline in monitor mode, then enforce.

Tier	What it evaluates / status
Tier 1 – Default-deny	Any destination not matched by an explicit permit is denied and logged. Removes every unguarded path to a tool endpoint. (Today.)
Tier 2 – Domain / path permit	WebGroups permit MCP, LLM, and HTTP destinations by fully qualified domain name and URL path using Layer 7 classification. (Today.)
Tier 3 – Inline pattern guards	ACS input/output pattern guards applied inline using traditional intrusion prevention system (IPS) rules – pattern-based detection of PII, confidential markings, and structured data. (Today.)

Tier 4 – Guardrail Profile	MCP and LLM protocol decoding, and semantic detection and redaction. (Coming soon.)

Coverage report. The plugin emits a report distinguishing rules enforced today (Tiers 1–3) from those pending the Guardrail Profile (Tier 4). Treat the report as the authoritative statement of what is live versus roadmap for a given policy file.

How the Controller Drives Enforcement

The Aviatrix Controller is the policy plane; the Spoke Gateways are the enforcement points. The shim translates the ACS contract into DCF objects, which the Controller programs onto the gateways without touching agent code.

Coverage across runtimes

1. The plugin reads the ACS guardrails file and resolves tool names to hosts via the tool-to-host registry.
2. It emits DCF objects: source SmartGroup, destination WebGroups, allow/deny rules, and inline IPS guards.
3. It applies them via the REST API
4. The Controller programs the Spoke Gateways; enforcement takes effect at the network boundary on the next connection.

Kubernetes custom resource definitions

On Kubernetes, per-agent-group policy can be expressed as custom resource definitions (CRDs) applied in-cluster, while SmartGroups resolve agent identity by namespace and label. The Controller reconciles the CRD-defined intent with gateway enforcement at the node subnet boundary. CRD-defined WebGroups live in the cluster and are distinct from controller-managed WebGroups; both are evaluated under the same default-deny model.

Audit correlation

When ACS policy events carry a session identifier, CoPilot correlates those framework-level decisions with gateway flow logs, giving one view across the agent's intended verdict and the network's actual enforcement.

Known Constraints

Constraint	Workaround / Notes
MCP-over-stdio agents generate no network traffic and are invisible to DCF.	Deprecate stdio MCP in governed estates so all tool calls traverse the network. Where stdio is unavoidable, treat those agents as out of network scope and govern them by other means.

Stateful ACS predicates (sensitivity ratchets, session variables, conditional rules, approval gates) are not enforced at the gateway today.	Enforce the unconditional subset now (Tiers 1–3); track the pending predicates via the coverage report; adopt the Guardrail Profile when available (coming soon).
ACS resource catalog carries tool names and path patterns, not hostnames.	Supply a tool-to-host registry to the plugin, or enable gateway MCP/TLS inspection, to bind tool names to FQDNs for WebGroup construction.
Semantic PII detection and redaction are not available with traditional IPS alone.	Use IPS pattern guards for structured and marked data today; enable semantic detection and redaction with the Guardrail Profile.
Bare wildcard destinations are not accepted as a permit target.	Use exact hostnames or leading wildcards (for example, *.example.com). A bare * is rejected by the Controller.

Conclusion

The integration compiles a portable Microsoft ACS policy file into enforced DCF network policy on a standard Aviatrix deployment, with no agent code changes. Layer 4 default-deny enforcement and inline IPS pattern guards are validated today. The same DCF fabric governs production workloads, AI agents, and MCP servers – one policy model, one audit log, one control plane.

Appendix — Cloud-Specific Domain

Permit each agent only the destinations its policy requires. The lists below are representative starting points; baseline against monitor-mode data before enforcing. Express destinations by fully qualified domain name; use leading wildcards where a provider rotates subdomains. Bind tool names to hosts via the tool-to-host registry.

Representative LLM and agent-platform destinations

Provider / platform	Representative permitted domains
OpenAI APIs	api.openai.com
Anthropic APIs	api.anthropic.com
Amazon Bedrock AgentCore	bedrock*.<region>.amazonaws.com scoped to the regions in use
Azure AI Foundry / Azure OpenAI	*.openai.azure.com, *.cognitiveservices.azure.com scoped to the resources in use
Google Vertex AI Agent Engine	*.googleapis.com scoped to the Vertex AI services in use

Cloud control-plane destinations (per runtime)

Cloud	Representative permitted domains
Amazon Web Services	*.amazonaws.com scoped to the regions and services the agent uses (for example, sts.<region>.amazonaws.com)
Microsoft Azure	login.microsoftonline.com, management.azure.com scoped to the resources in use
Google Cloud Platform	oauth2.googleapis.com, *.googleapis.com scoped to the services in use

Domain syntax: exact hostnames or leading wildcards (*.example.com). A bare * is rejected by the controller. MCP server hosts are environment-specific and must be supplied through the tool-to-host registry or resolved by gateway inspection.

Aviatrix Containment Plugin for Microsoft Agent Control Specification removes the unknown from agentic deployment by enforcing policy at the network layer. **Ask your Aviatrix account team** for a guided deployment.

Explore Validated Containment Architectures for other AI platforms.

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.