Unified Agent Management at Scale: A White Paper

Leveraging an OpAMP-Based Supervisor Proxy Architecture for Enterprise Fleet Management

Publication Date: December 5, 2025 **Author:** Manus AI

1. Executive Summary

1.1. The Challenge

Enterprise-scale observability and security require managing a massive and complex fleet of software agents. With over **200,000 endpoints**—including workstations, servers, and cloud infrastructure—and an average of six agents per endpoint, the organization is tasked with coordinating approximately **1.2 million agent instances**. The current model, which relies on baking agents into Amazon Machine Images (AMIs), creates a slow, cumbersome, and high-risk deployment process. Release cycles stretch from **1 to 9 months**, severely limiting the organization's ability to respond to security threats, deliver new features, and maintain compliance.

1.2. The Proposed Solution

This document proposes the adoption of a **Unified Agent Management Platform** built on the **Open Agent Management Protocol (OpAMP)**. This open-source standard provides a modern, centralized control plane to manage the entire lifecycle of every agent on every endpoint. The architecture consists of a central **OpAMP Server** for orchestration, a lightweight **Supervisor Proxy** on each endpoint, and the **managed agents** themselves.

1.3. Strategic Models Compared

We analyze three potential models for agent management:

1. **Unified OpAMP Model (Recommended):** A single, open-source platform to manage all agents (observability and security), providing maximum control, operational simplicity, and vendor independence.
2. **CrowdStrike Fleet Management:** A vendor-native tool that manages only CrowdStrike agents, creating management silos and vendor lock-in.

3. **Current State (AMI-Based):** The existing slow, disruptive, and high-risk process of baking agents into machine images.

1.4. Recommendation

We strongly recommend adopting the **Unified OpAMP Model**. It is the only solution that provides the required level of control, agility, and extensibility necessary for a world-class enterprise observability and security posture. The initial investment in this platform will yield significant long-term returns in speed, safety, and efficiency.

---

2. High-Level Architecture

The proposed architecture establishes a clear separation between the control plane and the data plane, enabling centralized management without creating a data bottleneck.

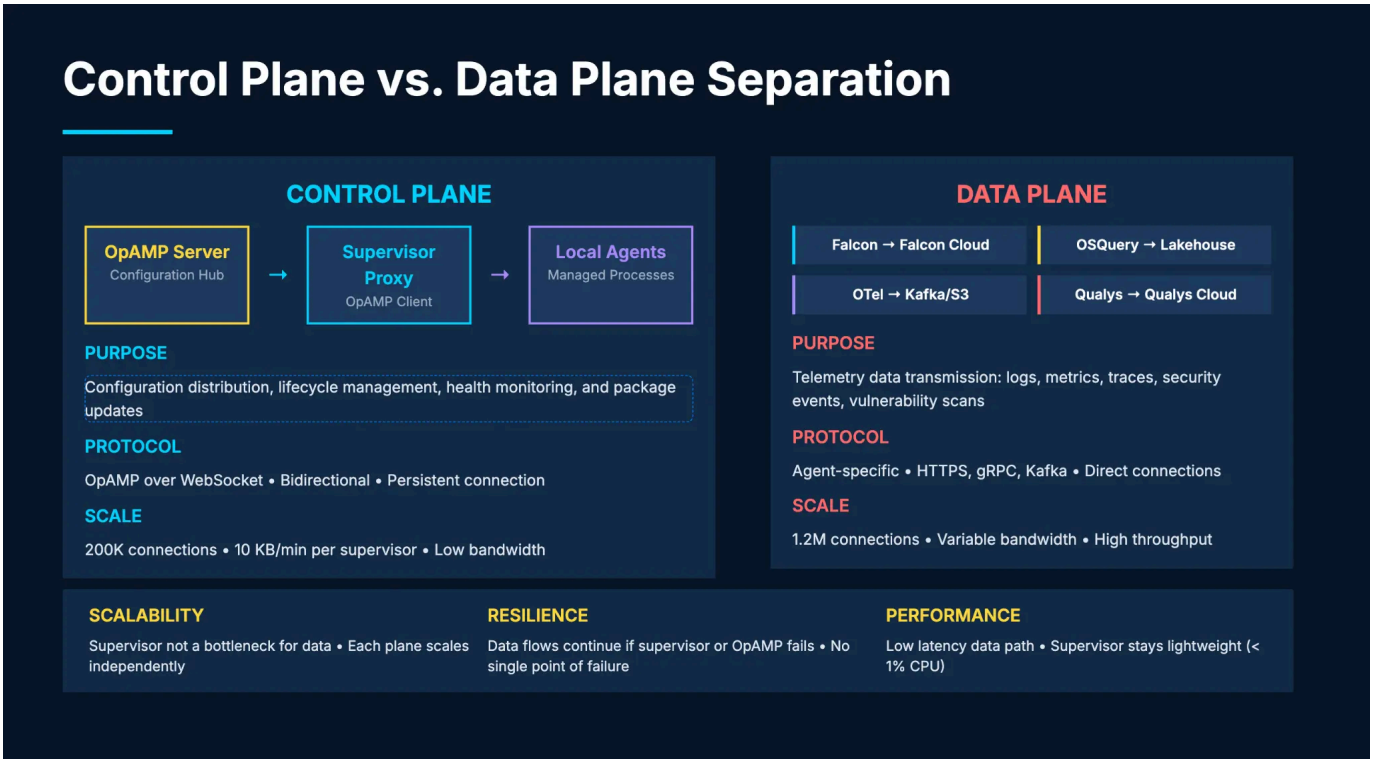
- **OpAMP Control Servers:** A multi-region, active-active cluster of servers that form the control plane. They are responsible for configuration management, health monitoring, and package distribution.
- **Supervisor Proxies:** A lightweight proxy (< 50 MB RAM) runs on each of the 200,000+ endpoints. Each supervisor acts as a single OpAMP client, receiving commands from the control servers.
- **Managed Agents:** The 1.2 million agents (CrowdStrike, OSQuery, OpenTelemetry, etc.) are managed locally by the supervisor via IPC/REST.

- **Data Plane:** The agents continue to send their telemetry data directly to their respective destinations (Falcon Cloud, Lakehouse, Kafka, etc.), ensuring the supervisor does not become a bottleneck.

---

3. Control Plane vs. Data Plane Separation

A core principle of this architecture is the strict separation of the control plane (how agents are managed) from the data plane (the telemetry data the agents produce).

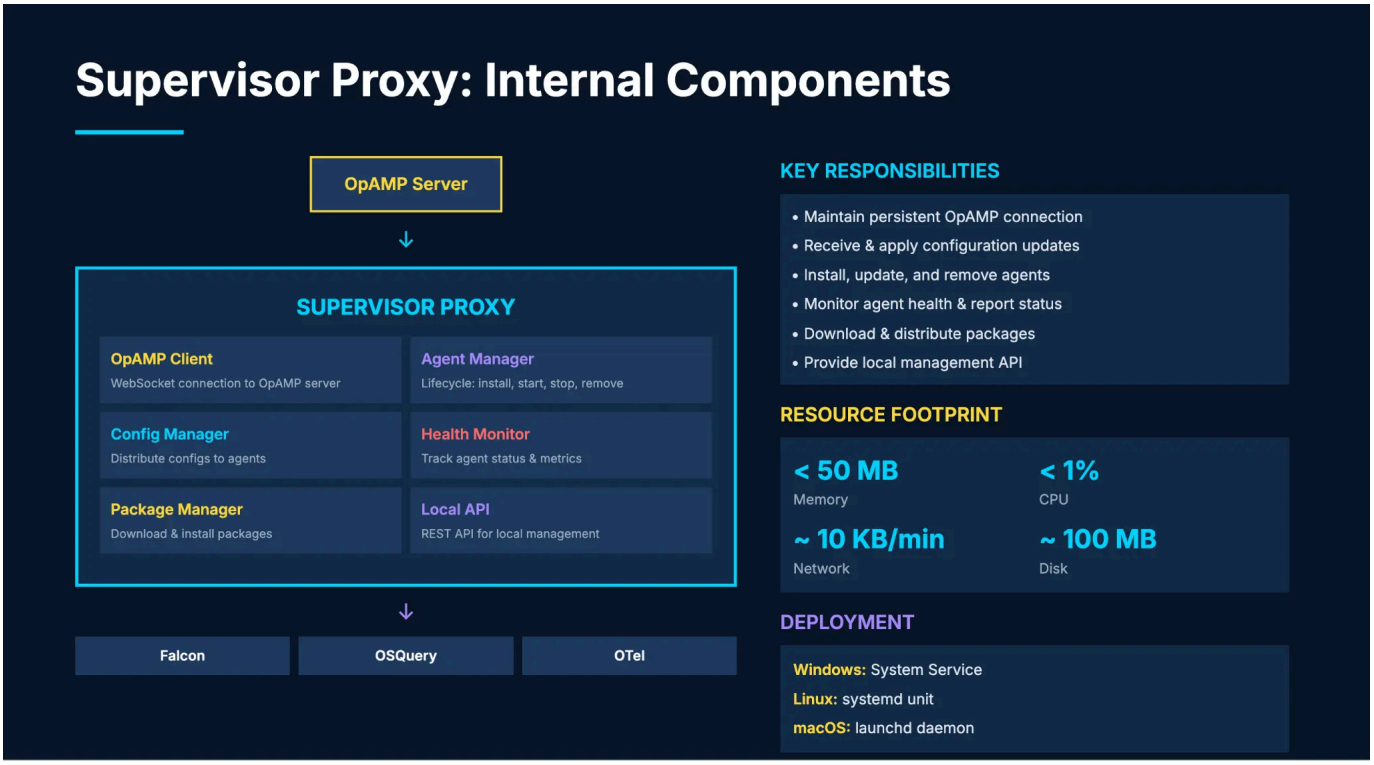


Plane	Control Plane	Data Plane
Purpose	Configuration distribution, lifecycle management, health monitoring, and package updates.	Telemetry data transmission: logs, metrics, traces, security events, vulnerability scans.
Protocol	OpAMP over WebSocket (Bidirectional, Persistent)	Agent-specific (HTTPS, gRPC, Kafka, Direct connections)
Scale	200K connections • 10 KB/min per supervisor • Low bandwidth	1.2M connections • Variable bandwidth • High throughput
Benefits	**Scalability:** Supervisor is not a data bottleneck. **Resilience:** Data flows	

	continue if supervisor or OpAMP fails. **Performance:** Low latency data path; supervisor stays lightweight.	

4. Supervisor Proxy: Internal Components

The Supervisor Proxy is a purpose-built, lightweight service with several key internal components designed for efficient local agent management.

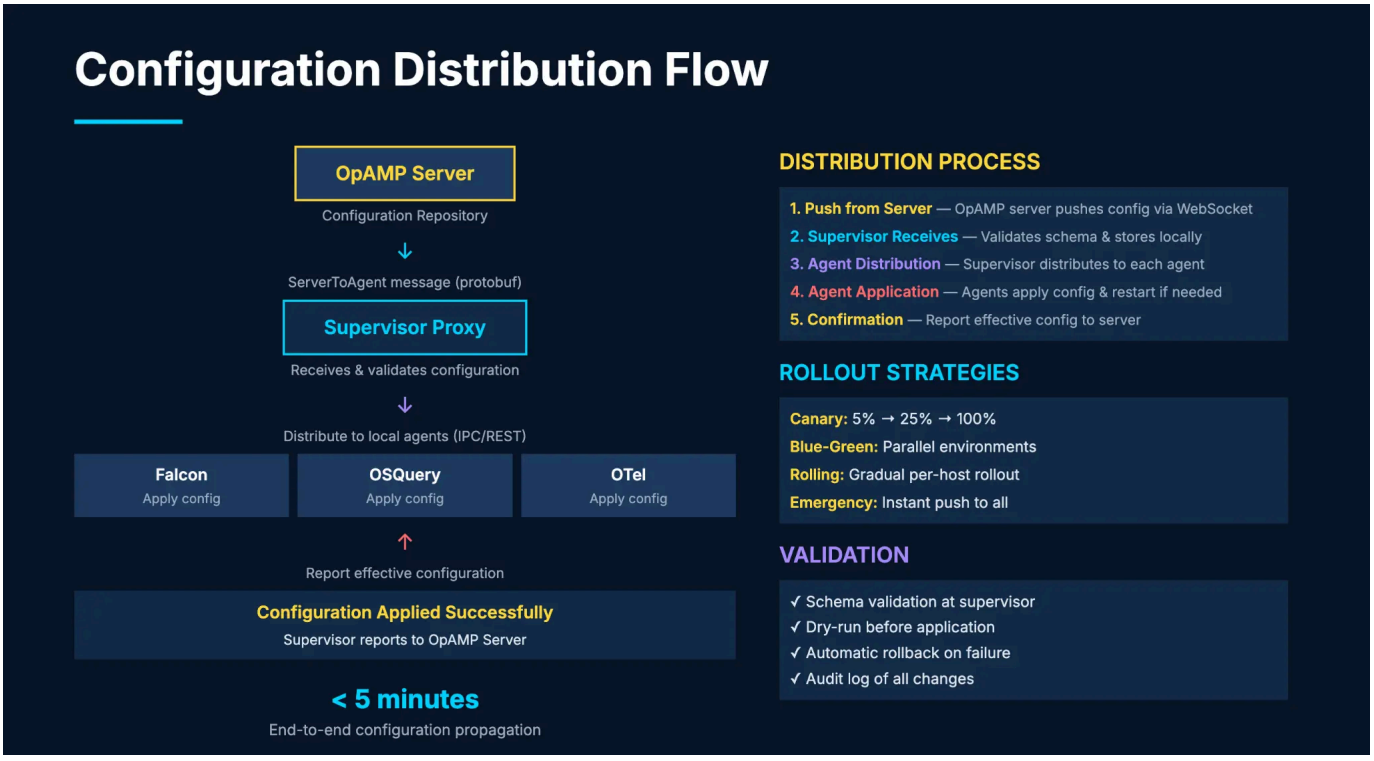


- **OpAMP Client:** Maintains the persistent WebSocket connection to the OpAMP server.
- **Config Manager:** Receives and distributes configurations to the managed agents.
- **Package Manager:** Downloads and installs new agent binaries.
- **Agent Manager:** Manages the lifecycle of local agents (start, stop, restart).
- **Health Monitor:** Tracks the status and resource usage of each agent.
- **Local API:** Provides a local REST API for on-host diagnostics and management.

5. Configuration and Package Distribution

The platform provides robust, centrally managed workflows for distributing both configurations and software packages to the entire fleet.

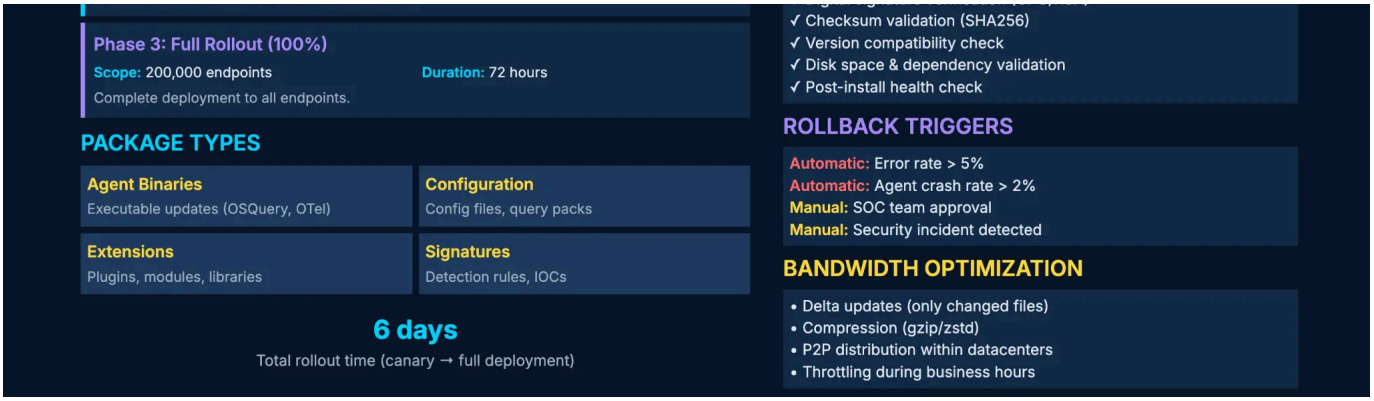
5.1. Configuration Distribution Flow



1. **Push from Server:** The OpAMP server pushes a new configuration via WebSocket.
2. **Supervisor Receives:** The supervisor validates the schema and stores the new configuration.
3. **Agent Distribution:** The supervisor distributes the configuration to each managed agent.
4. **Agent Application:** The agent applies the configuration, restarting if necessary.
5. **Confirmation:** The supervisor reports the effective configuration back to the server.

This entire process completes in **< 5 minutes** end-to-end.

5.2. Package Distribution & Rollout



- **Canary Rollout Process:** A phased approach (e.g., 5% → 25% → 100%) allows for safe, validated rollouts of new agent binaries over a period of days.
- **Distribution Mechanism:** The OpAMP server notifies supervisors of a new package URL. The supervisor downloads, verifies the signature, installs, and reports status.
- **Validation & Rollback:** A rich set of validation checks (digital signature, checksum, disk space) and automatic rollback triggers (error rate > 5%, crash rate > 2%) ensure deployment safety.

---

6. Conclusion

The Unified OpAMP Model provides a comprehensive, modern solution for managing a large and diverse fleet of agents. By separating the control and data planes and leveraging a lightweight supervisor proxy, this architecture delivers centralized control without compromising performance or creating data bottlenecks. It enables fast, safe, and efficient management of agent configurations and binaries, reducing deployment times from months to days and significantly lowering operational risk.

---

7. Capabilities Roadmap: From Current State to Target State

The following table outlines the key capabilities of the Unified Agent Management Platform, along with their current status, priority, and estimated effort to reach the target state. This roadmap provides a clear view of the work required to build the platform.

Capability Group	Capability	Description	Status	Priority	Effort (Phase 1)	Comments

Core Platform	OpAMP Server Cluster (AWS)	A multi-region, active-active cluster of servers that form the control plane.	**Planned**	P0	High	Requires infrastructure setup in AWS (EKS, RDS, etc.) and high-availability design.
Core Platform	Supervisor Proxy (Linux, Win, macOS)	A lightweight proxy (< 50 MB RAM) that runs on each endpoint as a single OpAMP client.	**Planned**	P0	High	Requires building and packaging for all three major operating systems.
Agent Inventory	Automatic Registration	Agents automatically register with the OpAMP server on first boot.	**Planned**	P0	Low	A simple bootstrap process using a one-time token.
Agent Inventory	Dynamic Fleet Inventory	A real-time, searchable inventory of all agents, their versions, and their status.	**Planned**	P0	Medium	Requires a database (e.g., RDS) to store and query agent metadata.
Configuration	Centralized Push Updates	The ability to push	**Planned**	P0	Medium	The core of the OpAMP

		configuration changes to the entire fleet in near real-time.				protocol; requires robust WebSocket handling.
Configuration	Git-Based Versioning	All configurations are stored in a Git repository, providing versioning, auditing, and rollback.	**Planned**	P0	Medium	Integrates with existing CI/CD pipelines for configuration management.
Health Monitoring	Continuous Health Checks	The supervisor continuously monitors the health of each managed agent.	**Planned**	P1	Medium	Health checks include CPU, memory, and restart counts.
Health Monitoring	Real-Time Metrics & Dashboards	A centralized dashboard showing the health and performance of the entire agent fleet.	**Planned**	P1	High	Requires a time-series database (e.g., Prometheus) and a visualization tool (e.g., Grafana).

Package Management	Remote Updates & Upgrades	The ability to remotely update and upgrade agent binaries.	**Planned**	P1	High	A critical feature for maintaining security and delivering new features.
Package Management	Artifactory Integration	Agent binaries are stored in Artifactory, providing a secure and reliable distribution point.	**Planned**	P1	Medium	Integrates with existing Artifactory infrastructure.
Deployment Strategies	Canary Deployments	The ability to roll out changes to a small subset of the fleet before a full rollout.	**Planned**	P1	High	A key feature for reducing deployment risk.
Deployment Strategies	Tag-Based Targeting	The ability to target deployments to specific groups of endpoints based on tags (e.g., region, OS).	**Planned**	P1	Medium	Requires integration with a tag management system.

Deployment Strategies	Instant Rollback	The ability to instantly roll back a failed deployment.	**Planned**	P1	Medium	A critical feature for maintaining service availability.
Security & Compliance	mTLS Authentication	All communication between the supervisor and the OpAMP server is authenticated using mutual TLS.	**Planned**	P2	Medium	Requires a PKI infrastructure for issuing and managing certificates.
Security & Compliance	RBAC & Audit Trail	Role-based access control and a complete audit trail for all changes.	**Planned**	P2	Medium	Ensures that only authorized users can make changes and that all changes are tracked.
Agent Wrappers	CrowdStrike Falcon/LogScale	A wrapper that allows the supervisor to manage CrowdStrike agents.	**Planned**	P2	Medium	Requires reverse-engineering the CrowdStrike agent's configuration and management APIs.

Priority Levels:

- **P0:** Foundational capabilities required for the platform to function.
- **P1:** High-priority features that deliver major business value.
- **P2:** Important features for enhancing security, compliance, and extensibility.

Effort Levels:

- **High:** Requires significant design and engineering effort (multiple engineers, 1-2 quarters).
- **Medium:** Requires moderate engineering effort (1-2 engineers, 1 quarter).
- **Low:** Can be implemented with minimal engineering effort (1 engineer, a few sprints).