LATINFLASH RESEARCH · TECHNICAL PUBLICATION SERIES

PAPER 2026-07 · TECHNICAL REPORT · VERSION 1.0

Applied Audit 01: OpenAI Swarm — Framework-native governance controls versus externally required controls under the Vertical AI Governance Stack v1.0

Manuel Enrique Morales Santiago

LatinFlash Research | ORCID iD: 0009-0007-6921-7688 | Dialnet Code: 3123473

Document Date: September 19, 2026 | Status: Independent Technical Audit Report

ABSTRACT

Empirical governance audit evaluating OpenAI Swarm handoff mechanisms, context propagation, and failure semantics against the Vertical AI Governance Stack v1.0 requirements. Differentiates framework-native control capabilities from external governance requirements.

Framework: OpenAI Swarm (Educational/Experimental)
Reference Standard: Vertical AI Governance Stack v1.0
Scope: Primitives & Architectural Defaults
Assessor: Governance Conformance Observatory

1. Executive Summary

This report utilizes **OpenAI Swarm** as a baseline case study to evaluate how deliberately minimal agent orchestrators handle the structural requirements of production-grade autonomy.

By analyzing Swarm's core primitives—**Agents**, **Tools**, and **Handoffs**—we demonstrate that while the framework achieves exceptional developer ergonomics, it delegates nearly all governance responsibilities to the implementation layer. This audit identifies the delta between framework-native capabilities and the deterministic controls required by the **Vertical AI Governance Stack v1.0**.

Framework Conformance Dashboard

Layer	Framework-native control	Assessment
L1 Normative Authority	No normative authority model supplied	OUT OF SCOPE / EXTERNAL
L2 Policy & Taxonomy	No centralized policy enforcement layer	ABSENT BY DEFAULT
L3 Capability	Tool availability exists; external capability authorization does not	PARTIAL / DEVELOPER-SUPPLIED
L4 Execution & Control	Direct function dispatch; no mandatory authorization boundary	NON-CONFORMANT BY DEFAULT
L5 Verification & Audit	Conversation/tool results returned; no durable evidence plane	PARTIAL / EXTERNAL PERSISTENCE
L6 Resilience	No governance-specific circuit breaker/fallback layer	ABSENT BY DEFAULT

2. Key Architectural Findings

Observed Architecture

Finding 1: Absence of a Framework-Enforced Execution Boundary (L4)

Swarm implements a direct dispatch loop where model-requested tool names are mapped to Python functions and executed using `function_map[name](**args)`.

What Swarm guarantees
Successful execution of a tool if the model provides syntactically valid arguments.

Deployment Responsibility
Developer must implement parameter validation and business-rule checks inside every exposed function.

Vertical Stack Requirement
A deterministic Capability Verification Adapter (CVA) must independently authorize parameters before execution.

Residual Gap: Swarm does not impose a framework-enforced execution boundary between the model's decision to execute and the system's action of execution.

Documented Behavior

Finding 2: Capability Exposure vs. Capability Authorization (L3)

Swarm utilizes Handoffs where a function returns an `Agent` object to transfer control.

What Swarm guarantees
The model can only transfer to agents explicitly exposed via `agent.functions`.

Deployment Responsibility
Ensure any exposed handoff function is appropriate for the current user's privilege level.

Vertical Stack Requirement
The framework must independently verify that an exposed handoff remains within authorized scope.

Residual Gap: Swarm does not provide a native mechanism to authorize a handoff. If a transfer function to a more privileged agent is exposed, Swarm activates it without an external authorization decision.

Stack Interpretation

Finding 3: The Reconstructibility Obligation (L5)

Swarm is designed as a stateless framework that returns `messages`, the last `Agent`, and `context_variables` to the client after each run.

What Swarm guarantees
Return of the complete turn history to the caller.

Deployment Responsibility
Implement an external persistence layer to store these turn history results.

Vertical Stack Requirement
A durable, append-only evidence plane capable of reconstructing identity, authorization, and outcomes.

Residual Gap: Swarm’s stateless core does not satisfy the reconstructibility obligation. A conformant deployment requires an external evidence plane.

3. Comparative Context: From Swarm to Agents SDK

The transition from Swarm to the **OpenAI Agents SDK** is consistent with several control gaps identified in this audit. The production-oriented SDK adds sessions, tracing, and structured handoff primitives that were not native to Swarm.

Crucially, the Agents SDK introduces native guardrail mechanisms, including tool-level validation hooks, materially reducing the L4 gap. Whether these controls satisfy a deterministic governance boundary still depends on where and how they are configured.

4. Critical Architectural Tests (Framework Primitives)

- **Test 1: Can the LLM directly invoke a production side effect without a framework-enforced deterministic authorization step?**
  Result: YES — The framework dispatches tools immediately upon model request.

- **Test 2: Can the framework independently verify that an exposed handoff remains within the caller's authorized capability scope?**
  Result: NO — Authorization is coupled with tool exposure.
- **Test 3: Does the framework native core provide a durable, tamper-evident evidence plane?**
  Result: NO — Persistence must be managed externally.

5. Framework Boundary vs. Deployment Responsibility

Control	Swarm Primitive	Deployment Responsibility (for Conformance)
Authorization	Linked to Tool Exposure	Implement external Policy Engine (L2)
Validation	Schema-only	Implement deterministic parameter checking (L4)
Audit	Volatile Chat History	Implement append-only Evidence Plane (L5)
Circuit Breakers	Manual `max_turns`	Implement Governance Circuit Breakers (L6)

6. Conclusion

OpenAI Swarm provides a **useful Governance Baseline**. Its design clarifies the distinction between *Agent Ergonomics* (which Swarm solves) and *Autonomous Governance* (which the Stack requires).

For an enterprise building on Swarm, the **Implementation Conformance Checklist** identifies exactly what must be built around the framework: a deterministic execution boundary, an external policy store, and a durable evidence plane.

Cite This Audit

To cite this applied governance audit, use the following BibTeX entry:

```
@techreport{morales2026audit01swarm,
  author      = {Morales Santiago, Manuel Enrique},
  title       = {Applied Audit 01: OpenAI Swarm --- Governance Conformance Observatory},
  institution = {LatinFlash Research},
  type        = {Applied Audit},
  number      = {LatinFlash Research Paper 2026-07},
  year        = {2026},
  month       = {September},
  url         = {https://research.latinflash.com/papers/applied-audit-01-openai-swarm/},
  note        = {ORCID: 0009-0007-6921-7688, Dialnet: 3123473}
}
```

Permanent Identifier: **https://research.latinflash.com/papers/applied-audit-01-openai-swarm/** ORCID iD: 0009-0007-6921-7688 Dialnet Code: 3123473