065: Composio Schema Injection & Guardrails Pipeline

Date: January 15, 2026
Status: Draft (Discovery Notes)

---

1. Overview

Composio tool calls are guarded by schema validation and tool-name hygiene checks prior to execution. These guardrails are designed to prevent malformed tool names, invalid *COMPOSIO_MULTI_EXECUTE_TOOL* payloads, and schema mismatches.

---

2. Pre-Tool Guardrail (Schema)

pre_tool_use_schema_guardrail executes before tools run and performs:
• **Malformed tool name detection** (XML-style arg fragments)
• **Name normalization via** *parse_tool_identity*
• **Schema validation for known tools**
• **Special validation for** *COMPOSIO_MULTI_EXECUTE_TOOL* **inner tool entries**

If validation fails, the hook blocks the tool call and returns a corrective system message.

---

3. Schema Sources

Schema matching happens in this order:
1. Curated schemas in *_TOOL_SCHEMAS* (guardrails/tool_schema.py)
2. Composio live schema fetcher (*_fetch_composio_tool_schema* in main.py)

The Composio fetcher uses caching (*_COMPOSIO_SCHEMA_CACHE*) and an allowlist to reduce overhead. Cache is cleared if it grows beyond 200 entries.

---

4. Post-Tool Validation Nudge

If a tool fails schema validation, *post_tool_use_schema_nudge* provides guidance to the model for corrected retries.

---

5. Guardrail Notes

• Multi-execute calls are capped at 4 tools per request.
• XML-fragment tool names are blocked immediately.

• Tool names are sanitized to strip hallucinated suffixes (e.g., `...TOOLtools`).

---

6. Related Files

• `guardrails/tool_schema.py`
• `durable/tool_gateway.py`
• `main.py` (schema fetcher + hook wiring)