Guarded Agentic AI Approach

for Network Automation and Orchestration

Sonja Filiposka, UKIM

WP6T2

6th SIG AI for NRENs Meeting, Madrid, Spain

10 March 2026

Why Guarded Agentic AI?

Problem

Network automation is increasingly AI-assisted

LLMs are powerful but:

- hallucinate
- break schemas
- skip validation
- perform unsafe actions

Risk

Direct AI-to-network automation may be unsafe

Orchestration must remain:

- deterministic
- auditable
- reversible
- governed

Approach

AI should propose, but workflow engines enforce

Idea: Guarded Agentic Architecture

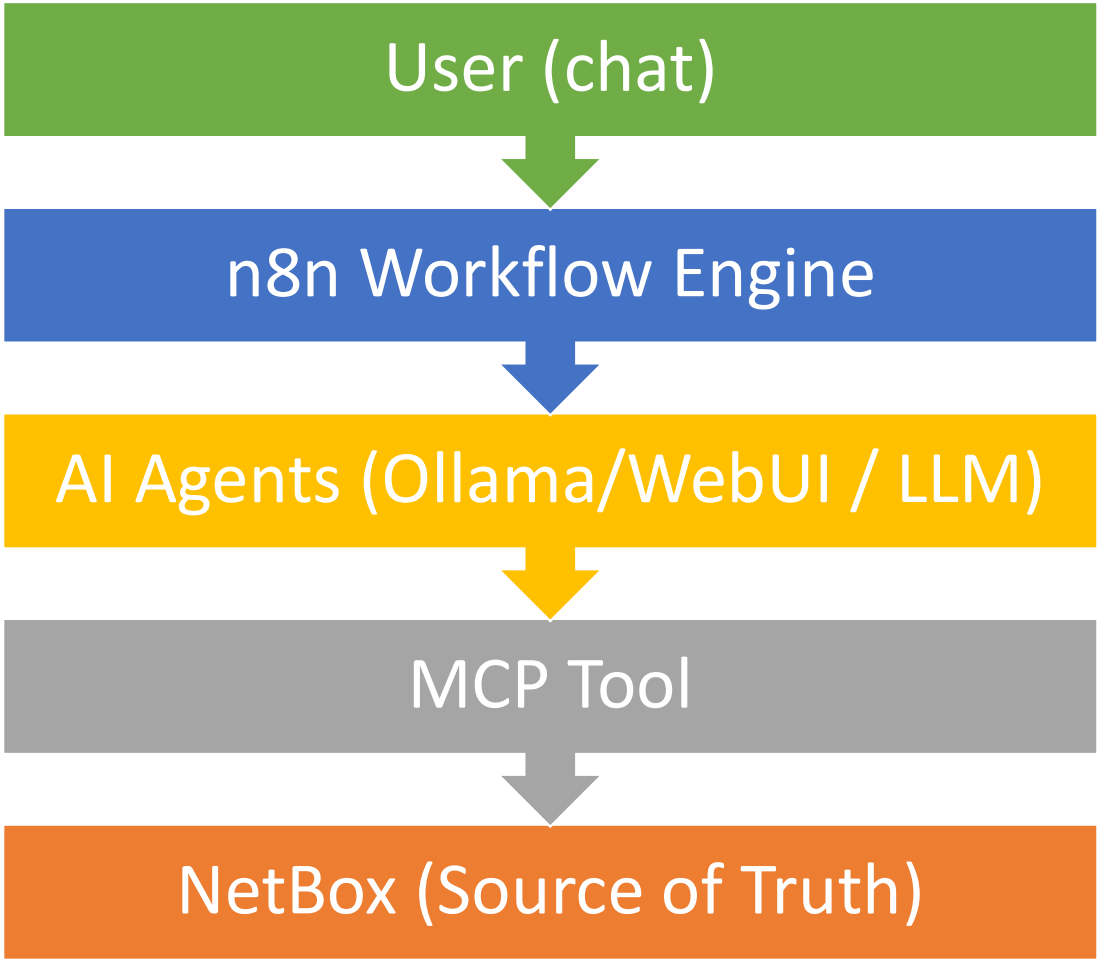
Instead of one "super-agent", use **specialised agents with explicit contracts:**

- **Planner**
  Interpret user intent
- **Resolver**
  Query source of truth
- **Patch Proposer**
  Generate patch + rollback
- **Verifier**
  Validate post-change state

Executed via:

- Deterministic workflow engine (n8n)
- Structured schemas
- Guard conditions
- Explicit human approvals

High-Level Architecture Overview



Key principle:

AI never directly modifies infrastructure

It generates proposals executed by deterministic nodes

Example – VLAN tagging on interface

Actions

- Add
- Remove

Information

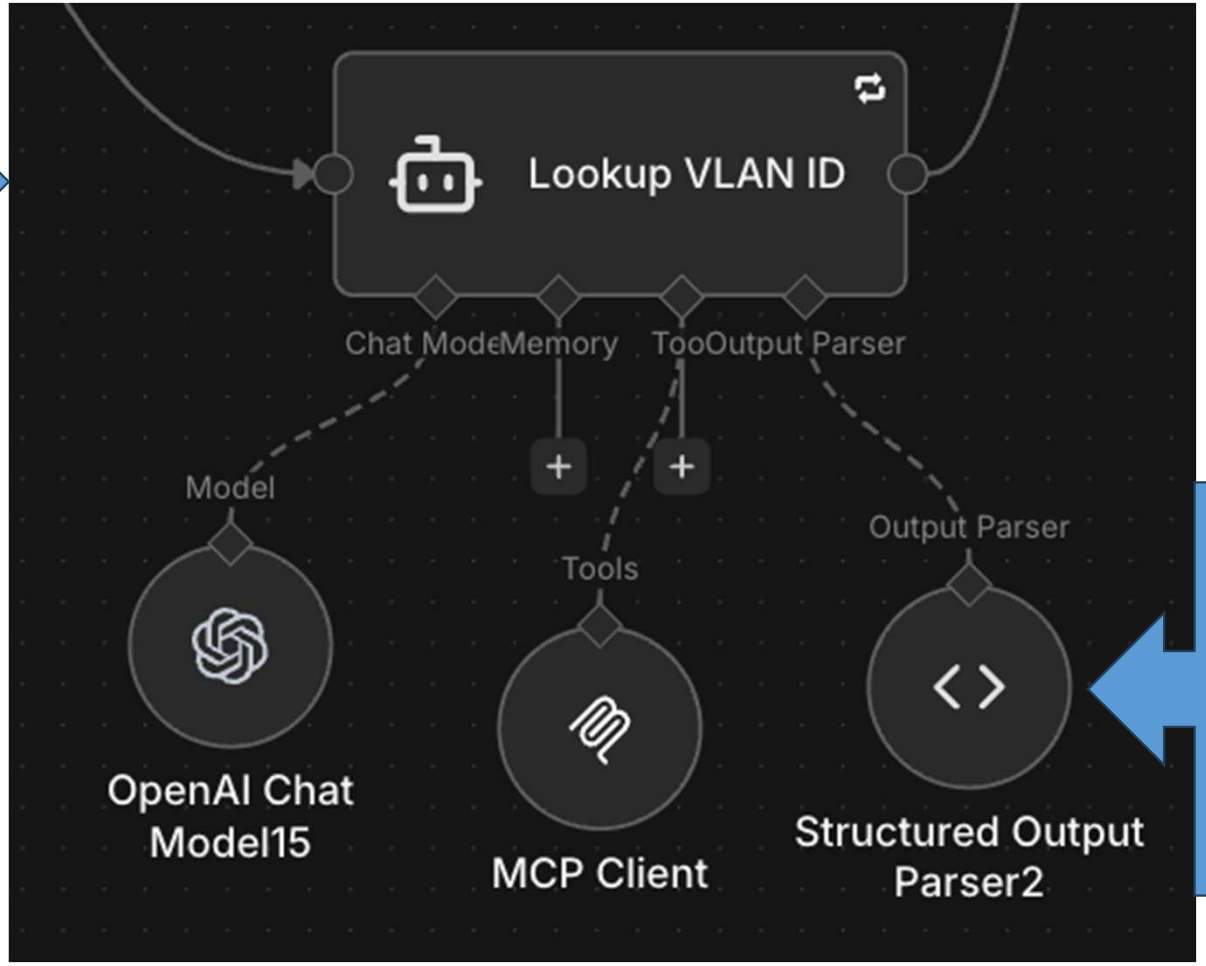
- VLAN ID
- Device name
- Interface name

Guards

- VLAN ID is in IPAM
- Device exists
- Interface on device exists
- Interface supports tagging
- VLAN ID on the interface already exists
- Human approval of change

AI Agent node in N8N

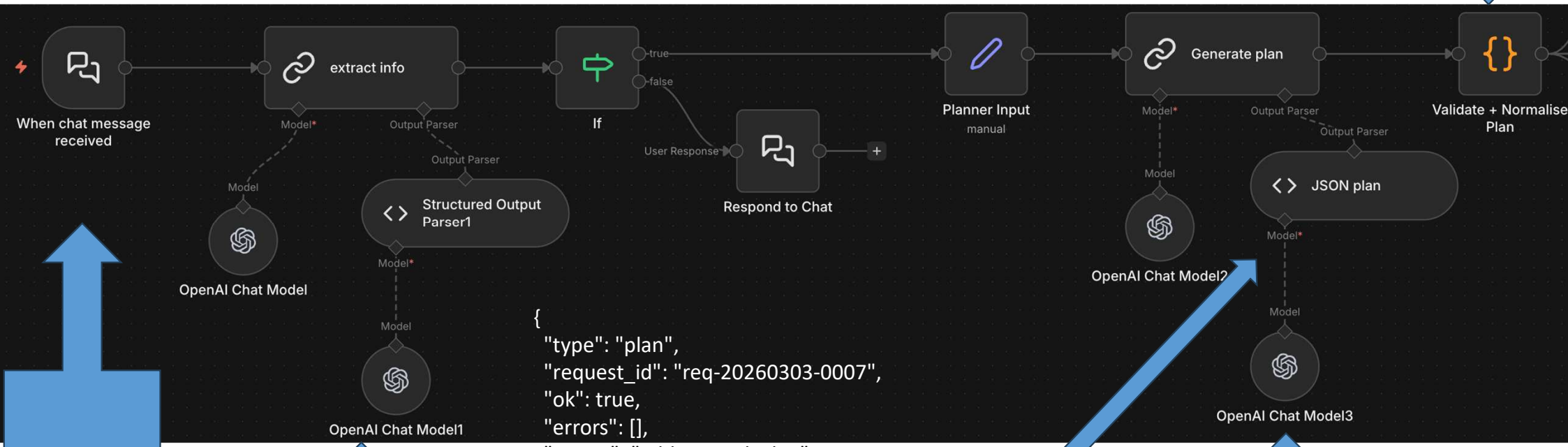
Prompt
(User/System message)
+
Input



JSON
example
or
JSON
schema

Agentic Contracts - Plan

Deterministic behaviour



Workflow Trigger

Parse input for required information

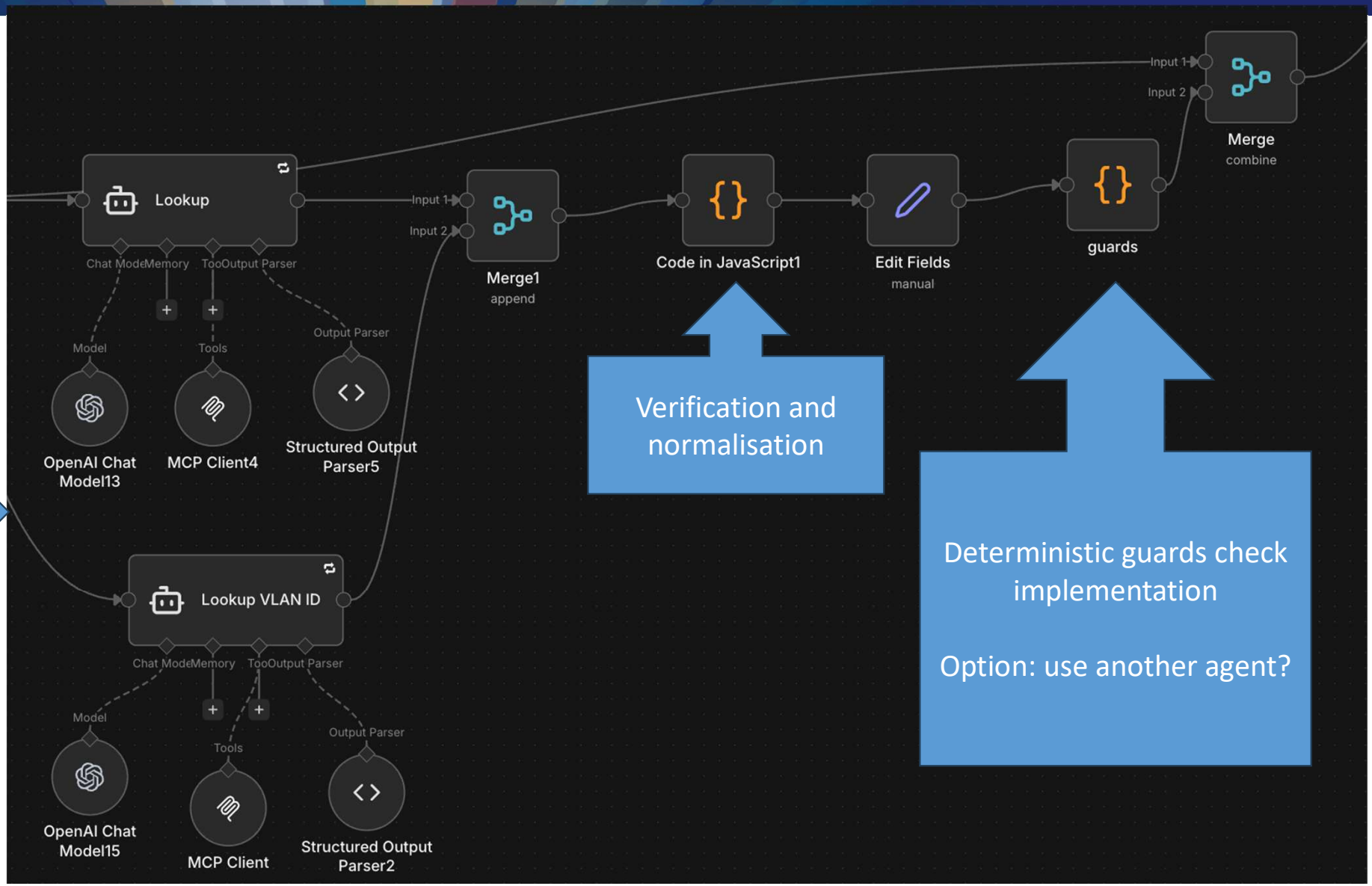

```
{
 "type": "plan",
 "request_id": "req-20260303-0007",
 "ok": true,
 "errors": [],
 "action": "add_tagged_vlan",
 "params": {
  "device": "rtr-lab-02",
  "interface": "ge-0/0/0",
  "vlan_vid": 40
 },  "lookups": [],  "guards": [],
 "approval_required": true,
 "explanation": "Plan to add VLAN 40 as a tagged VLAN on rtr-lab-02 interface ge-0/0/0."
}
```


Planning agent

Lookup

2 Lookup agents: Problems with multi-MCP calls

The most “unstable” part of the workflow



Models

Temperature
0.0

Tested with

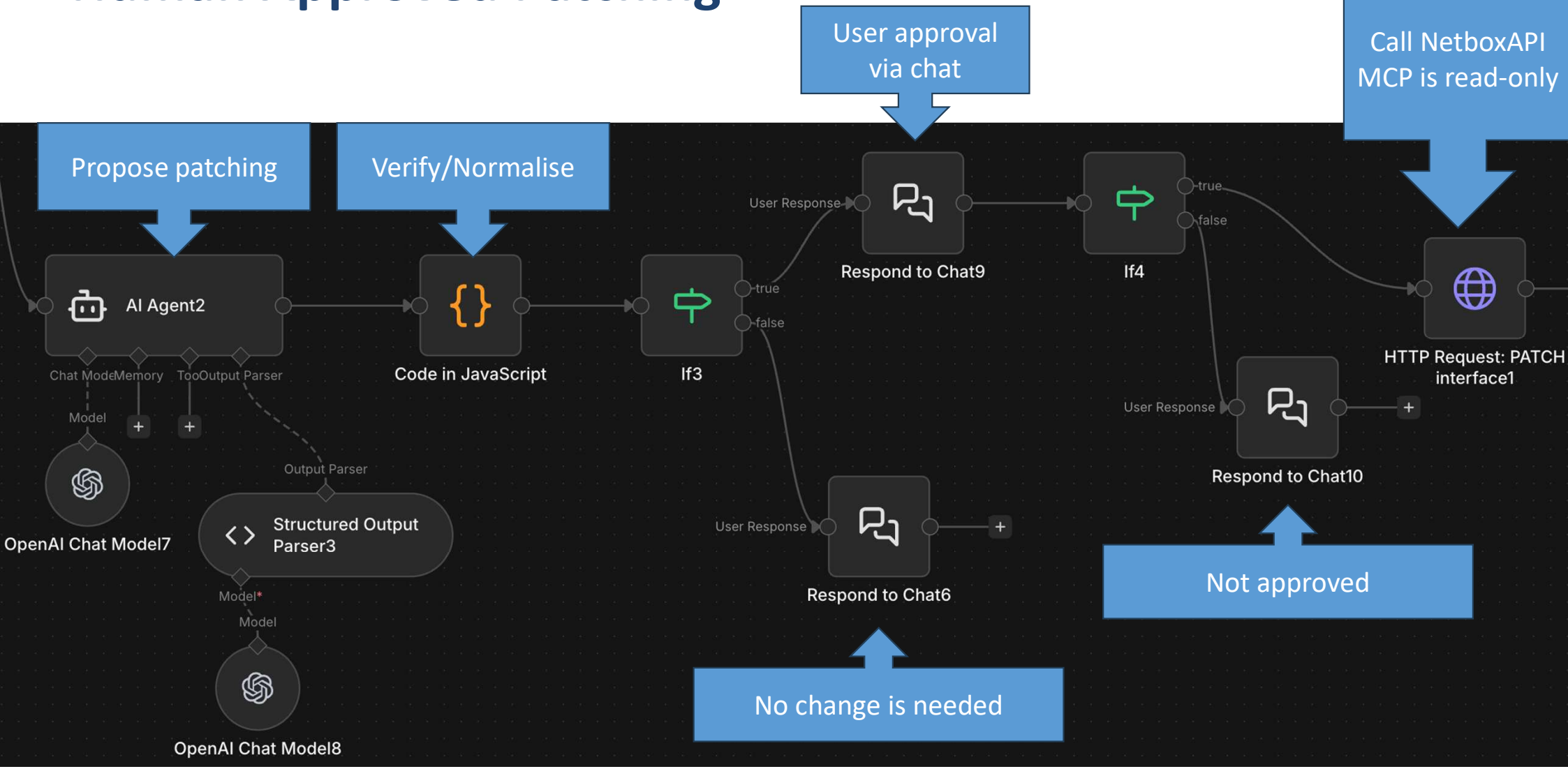
- gpt-oss:120b
- Qwen3-coder:30b
- qwen2.5:75b

Powered by PCSS

Observed behaviour:

- Tool-calling loops occurred
- Schema drift happened
- Hallucinated values appeared
- Structured output parser alone was insufficient
- Normalisation layer required

Human Approved Patching



Verify and Rollback if needed

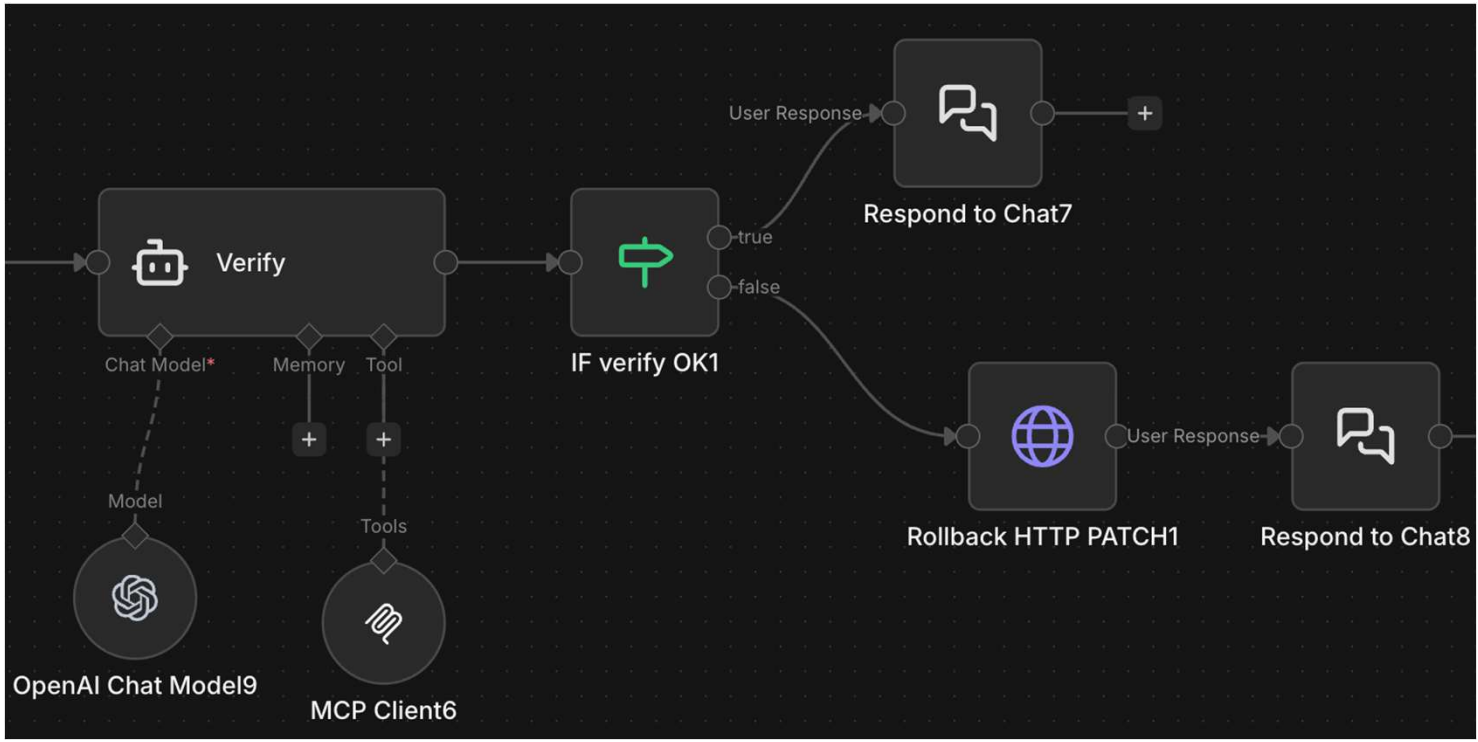
Agent prompt:
You are a NetBox verification agent. Use the MCP tool to verify the operation.

CONTEXT:
- interface id
- action
- VLAN VID, VLAN ID

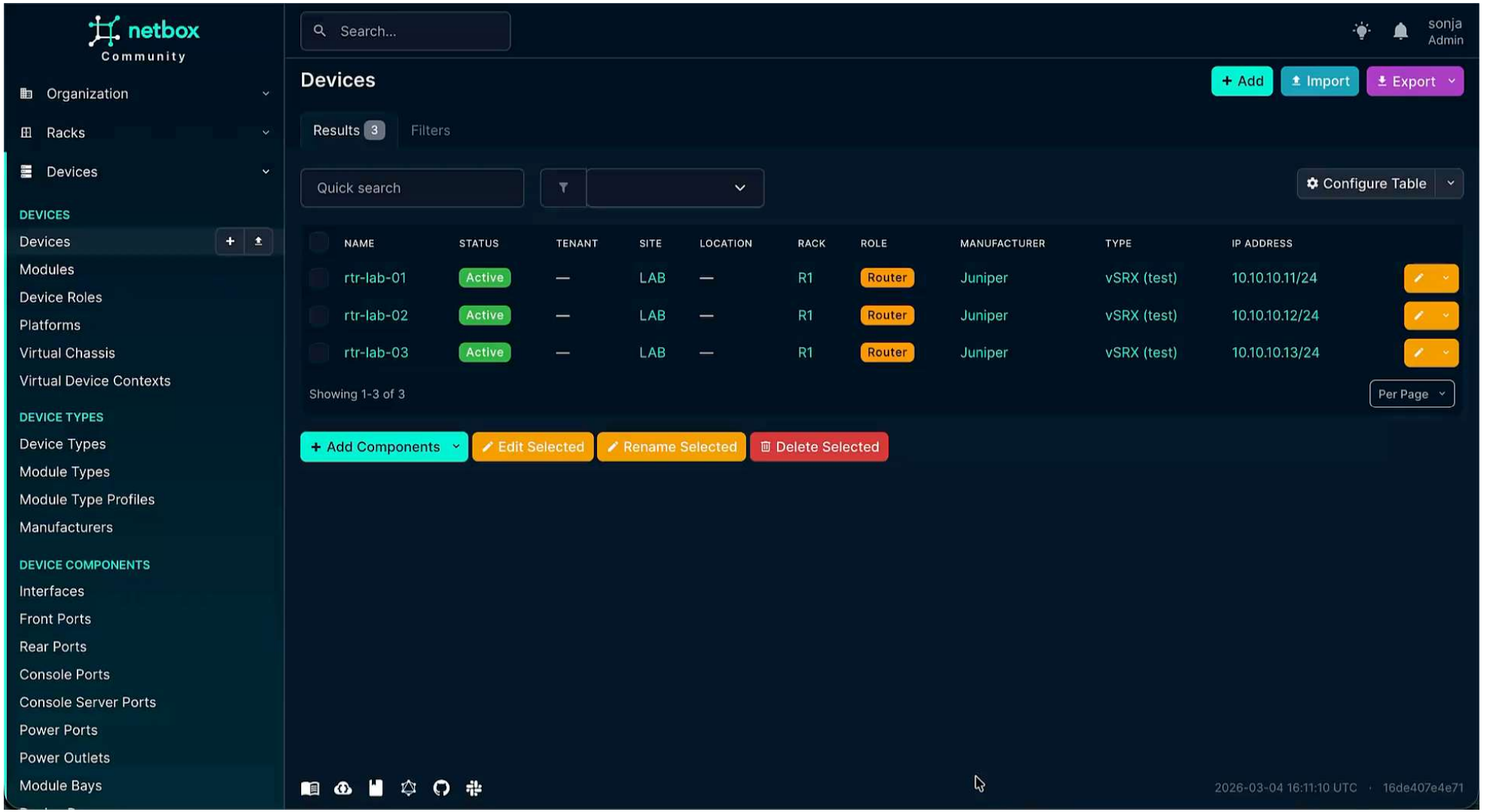
TASK: Call MCP, then check the returned tagged_vlans list for VLAN ID

RULES:
- Retry MCP call up to 3 times on failure.
- If still failing: verified=false, reason="MCP verification call failed after retries."
- NEVER invent results. Only report what MCP returns.

OUTPUT: {"verified": <boolean>, "reason": "<brief explanation>"}



All in One


netbox
Community
Search...
sonja
Admin
Organization
Racks
Devices
DEVICES
Devices
Modules
Device Roles
Platforms
Virtual Chassis
Virtual Device Contexts
DEVICE TYPES
Device Types
Module Types
Module Type Profiles
Manufacturers
DEVICE COMPONENTS
Interfaces
Front Ports
Rear Ports
Console Ports
Console Server Ports
Power Ports
Power Outlets
Module Bays
Devices
+ Add
Import
Export
Results 3
Filters
Quick search
Configure Table
NAME STATUS TENANT SITE LOCATION RACK ROLE MANUFACTURER TYPE IP ADDRESS
rtr-lab-01 Active — LAB — R1 Router Juniper vSRX (test) 10.10.10.11/24
rtr-lab-02 Active — LAB — R1 Router Juniper vSRX (test) 10.10.10.12/24
rtr-lab-03 Active — LAB — R1 Router Juniper vSRX (test) 10.10.10.13/24
Showing 1-3 of 3
Per Page
+ Add Components
Edit Selected
Rename Selected
Delete Selected
2026-03-04 16:11:10 UTC
16de407e4e71

Why This Is Not “Just ChatOps”

Lessons Learned

Occasional tool-call deadlocks

Structured output enforcement is essential

Multi-agent parallel resolution is more stable

LLM hallucination requires schema validation

Verification and normalisation layer is mandatory

Guard evaluation should not be delegated to LLM

Key Takeaways

AI should **propose**, not execute

Multi-agent separation increases reliability

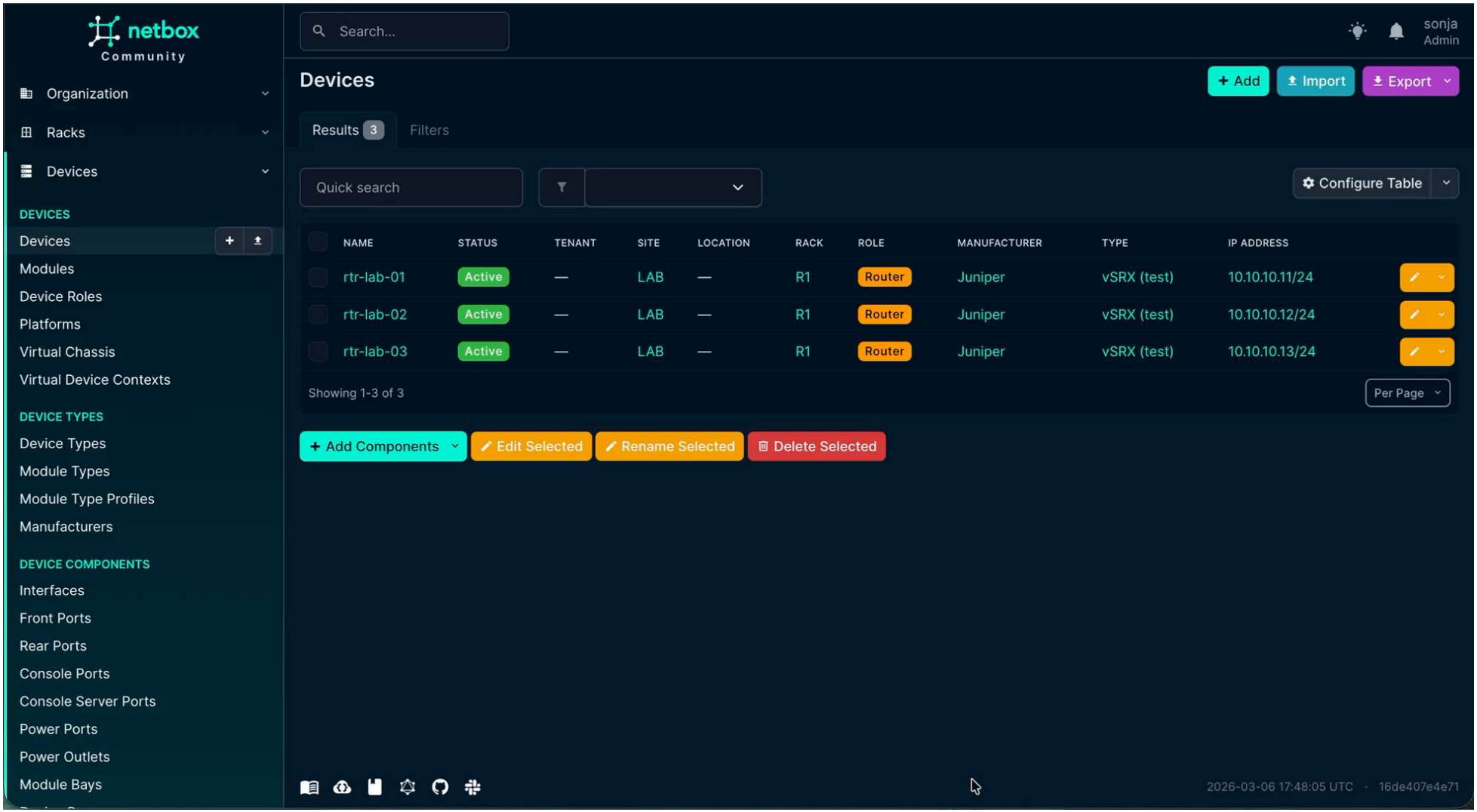
Deterministic workflow engines keep things **on track**

Strong **guard** evaluation is a must

Human approval remains crucial

Rollback must be **precomputed**

Non-Guarded Approach



System prompt

- You are a NetBox VLAN Orchestrator. Tools available:
  - MCP Client: read-only NetBox queries
  - netbox_patch_interface: PATCH interface tagged_vlans (ONLY after user approves)
  - netbox_rollback_interface: PATCH interface back to previous state (ONLY after failed verify)
- **TURN DETECTION**
  - Check conversation history. If any previous assistant message contains a line starting with "APPROVAL_TOKEN:" -> TURN 2. Otherwise -> TURN 1.
- **TURN 1**
  1. Parse input
  2. Look up the interface
  3. Look up the VLAN
  4. Validate — STOP on first failure, do NOT proceed to Step 5
  5. Look up VIDs for existing tagged VLANs
  6. Compute display lists (VIDs, sorted)
  7. Reply with this exact format:
  - NEVER call netbox_patch_interface or netbox_rollback_interface in TURN 1.
- **TURN 2**
  1. Read APPROVAL_TOKEN
  2. Check current user message
  3. Apply change
  4. Verify using PATCH response
  5. Compare and report
- **RULES**
  - interface_id and tagged_vlan_ids MUST come from the Step 2 tool response - never invent them
  - vlan_id MUST come from the Step 3 tool response
  - VLAN arrays contain integers only (no objects, no VIDs)
  - If a tool call fails, report the error immediately - do not retry
  - Do not make extra MCP calls beyond those specified above

Balancing the benefits

The Good

- One large agent handles planning, resolution, and patching
- Tool calls performed sequentially by the model
- Logic enforced through prompting

The Bad

- Agent must maintain state internally
- Careful when choosing a model
- Susceptible to tool-calling loops
- Schema drift and hallucinated values possible

The Ugly

- No validation against source of truth
- Works in controlled demos
- Hard to ensure it is always stable in production

Thank You

Special thanks to PCSS for allowing free access to their LLM models via WebUI

www.geant.org

Co-funded by the European Union