

TNC26 · HELSINKI · JUNE 2026

Beyond Configuration

Rethinking Network Orchestration

With Generative AI

Sonja Filiposka · UKIM

Roman Łapacz · PCSS



Networks are complex. Teams are small. AI changes this.

ASSISTED

1



Config Management

AI helps engineers generate and validate configurations

AUTOMATED

2



Workflow Orchestration

AI generates full workflow pipelines from intent alone

AUTONOMOUS

3



Agentic Orchestration

AI reasons, decides and acts on the live network

Three Forces Converging

LLM CAPABILITY



Models that actually work

- Production-grade models now widely available
- High accuracy on non-complex network config
- RAG + function calling close the domain gap
- Cost dropping — inference now viable at scale

NREN PRESSURE



More network, fewer engineers

- On-demand services expected as standard
- Multi-vendor, multi-domain complexity growing
- Smaller NRENs can't staff deep automation teams
- Manual config is the bottleneck, not the network

OPEN STANDARDS



The API surface is finally ready

- YANG + NETCONF + RESTCONF widely deployed
- TM Forum ODA providing service model layer
- A2A, MCP — agent interoperability emerging
- CMDB providing AI a grounded source of truth

What GenAI Can Do Today

Intent → Config



Natural language to vendor-specific device configurations (NETCONF, CLI, YANG)

AI Co-pilots



Interactive assistants that guide engineers step by step, reducing cognitive load

Validation & Fix



AI flags inconsistencies, suggests alternatives, and validates against policy

Multi-vendor Bridge



Unified intent layer across heterogeneous infrastructure — Cisco, Juniper, Nokia...

92%

zero-touch config accuracy
(LLM-NetCFG, small network, non-complex tasks)

***Strong results when tasks are structured,
scoped and verified.***

Complexity still breaks naive autonomy.

100%

Static NAT success (narrow task)
LLM + RAG + Function Calling

10x / 6x

Config translation / policy impl.
(verified prompt programming)

5-7 min

Non-complex config intents
handled (LLM-NetCFG)



Complex tasks still need
decomposition, verification
and human approval

Configuration generation is impressive.

**But networks do not
need more snippets.**

**They need safe, auditable,
end-to-end change workflows.**

"Vibe-Coding" Orchestration

Express intent. AI drafts the workflow.
You review, validate, deploy.

Phase 1

Holistic Prompting

Full-context prompt → standards-aligned workflow blueprint

Phase 2

Granular Refinement

One function at a time → auditable incremental patches, engineer approves each step

Human oversight at every step: AI suggests, engineers decide and validate

Create a network service
orchestration workflow...

Here's your workflow scaffold:
12 steps, custom operators OK

Add rollback on service fail

Guard task + auto-revert
operator added. Ready to test.

Generate service contract schema

Schema generated. ODA compliant.

Engineer

AI Copilot

What AI Actually Built



TM Forum Data Models

- TMF-compliant service schemas
- Dynamic attribute mapping
- Reusable across environments
- ODA-aligned blueprints



Orchestrated Workflows

- Complete workflow generation
- Custom operators & tasks
- API interaction code
- Error handling & rollback



DAYS → HOURS

- Development time slashed
- Standards-aligned output
- Consistent structure
- Lower barrier to entry

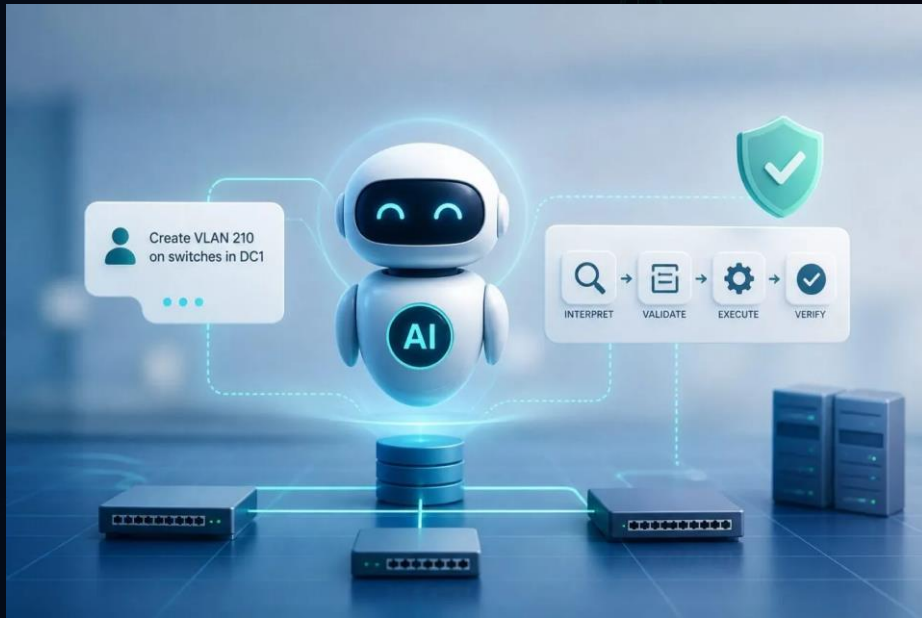
THE NEXT LEVEL

AI generates configs

AI generates workflows

AI acts on the network.

Observes • Plans • Decides • Acts



The question is not:

"Can an AI agent change the network?"

The real question is:

"Can we design the workflow so that when the AI is wrong, the network is still safe?"

Why Guarding Matters

approach to agentic network ops

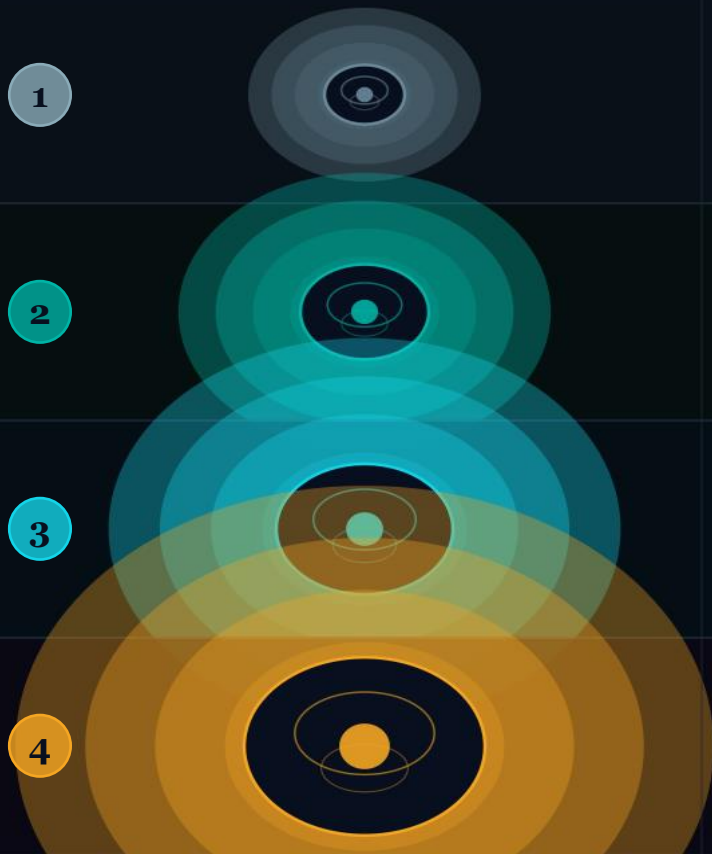
1 Just let the LLM call the network API directly

2 Add a try/catch and retry on failure

3 Validate every output — guard before every write

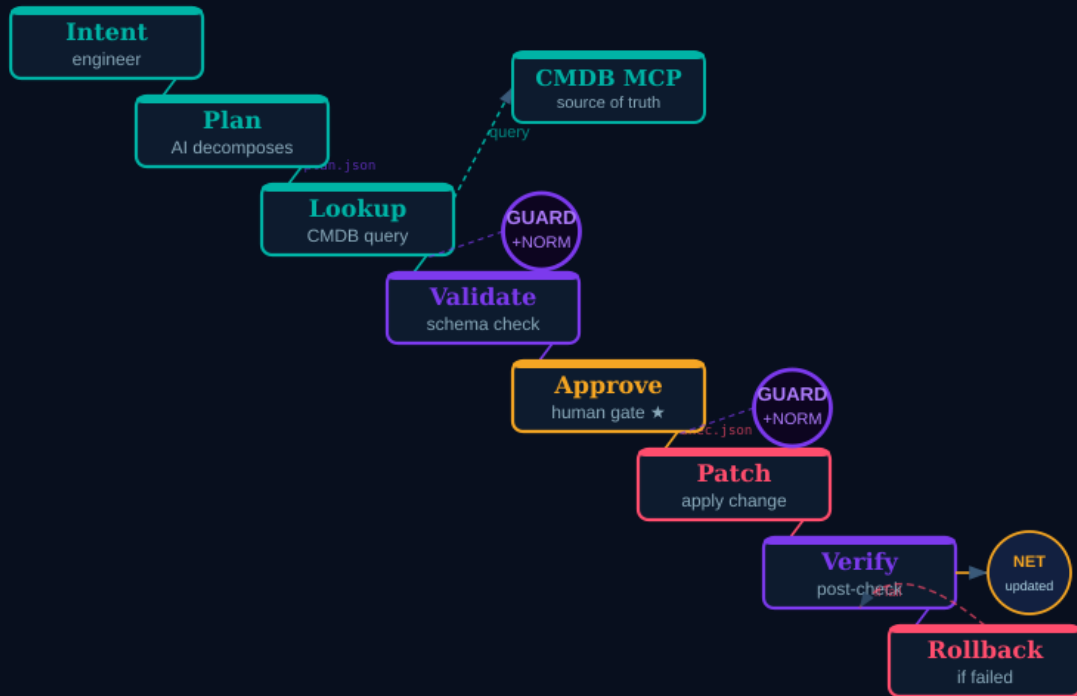
4 Normalise • Validate • Guard • Approve • Verify

at every agentic step — automatically



Multi-Agent Guarded

Apply a service change on a router — how agents collaborate to get it done safely



Agent Contracts

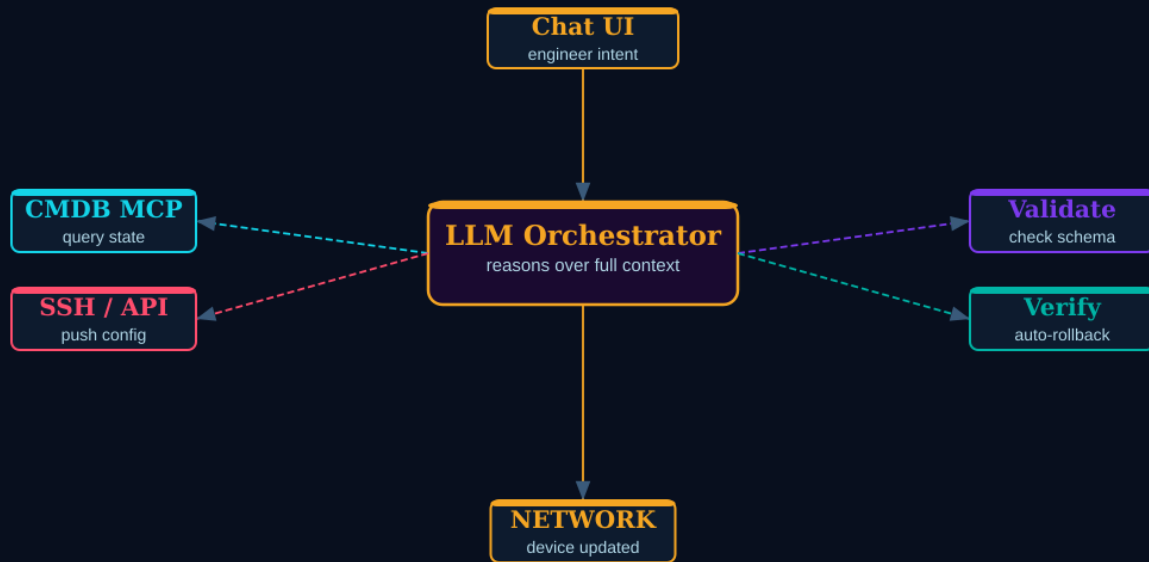
Agents speak structured JSON —
not freeform text

plan.json

```
{
  "action": "add_vlan",
  "device": "core-rtr-dc1",
  "vlan_vid": 40,
  "source": "cmdb://dc1",
  "ok": true,
  "approval": true
}
```

Single Orchestrator

One LLM reasons over the full context and decides the next tool call each step



The Good

Fast, simple, impressive demo value.
Great for rapid prototyping.

The Bad

Harder to audit — more hidden reasoning, more dependence on prompt discipline.

The Ugly

The model may skip a step, call tools in the wrong order, or produce a convincing but unsafe answer.

The Engine Matters

n8n

VS

Langflow

Graph-based routing

LangChain agent loop

Wait nodes & deterministic flow

LLM decides next tool call

~40 nodes for guarded pipeline

~8 nodes for orchestrator

Explicit routing between agents

Emergent agent behaviour

Production rigour

Rapid prototyping

Six Things We Learned

1 Works for blueprinting

Design & scaffolding phases — not yet for fully autonomous production deployment.

2 Guarding is essential

LLMs are probabilistic — architecture must catch failures visibly, never silently.

3 Human stays in loop

Every PATCH to the network requires explicit approval. A deliberate design choice.

4 Preprocessing is key

Structured inputs + RAG + vendor context dramatically improve AI output quality.

5 Build skills, not just workflows

Agentic workflow knowledge can be packaged as a skill and shared across the community.

6 Barrier lowered

Days → Hours for workflow generation. Any NREN can now participate in orchestration.

Adopt • Experiment • Lead

Today

- AI config generation
- Workflow scaffolding in hours
- Standards-aligned blueprints
- AI co-pilot for engineers

Near Future

- Guarded agentic network ops
- AI-driven fault remediation
- Dynamic service adaptation
- Multi-platform agent chains

Horizon

- Internet of Agents for NRENs
- Cross-domain orchestration
- Autonomous service lifecycle
- Open agent marketplaces

How to Start

Do not start with autonomous networks. Start with reusable, guarded, standards-aligned workflow patterns.

1

ASSISTED

Start with copilots.

AI suggests. Engineer decides. Every output reviewed before anything touches the network. Zero operational risk.

Risk: none · Reversibility: immediate

2

AUTOMATED

Move to workflow scaffolding.

Express intent. AI generates the workflow scaffold. Review and validate every output before deployment.

Risk: low · Reversibility: review before apply

3

AUTONOMOUS

Test agents only with contracts, guards, approval and rollback.

Guard before every write. Human approval for every change. Auto-rollback on failure. If you cannot explain what the agent did — do not deploy it.

Risk: contained · Reversibility: by design

Thank You

Questions & Discussion

Documentation: <https://geant-netdev.gitlab-pages.pcss.pl/gp4ldocs/guides/playground/>

Agentic AI tests conducted using open-source LLMs provided by



Poznańskie Centrum Superkomputerowo-Sieciowe
Poznan Supercomputing and Networking Center

The presentation was prepared with the assistance of Generative AI

— in the spirit of the topic —