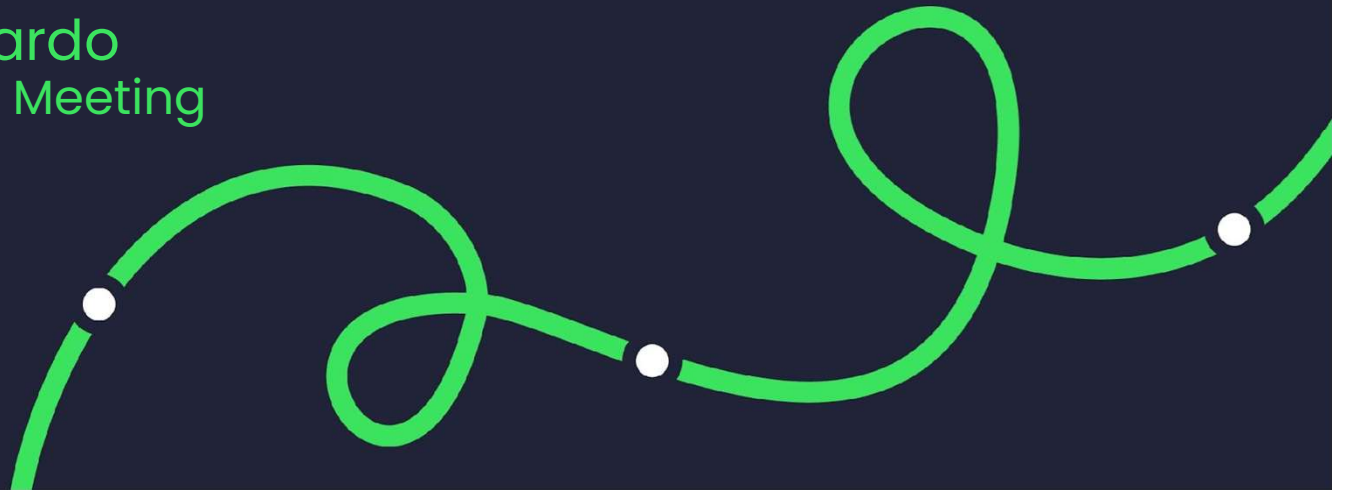


AI agents for GARR network analysis and troubleshooting

Francesco Lombardo
6th SIG-AI for NRENs Meeting



Agenda



Context & Motivation

Why AI agents for network operations?



Technical Implementation

Tech stack, observability



Architecture & Concepts

Multi-agent system, tools, skills, progressive disclosure.



Impact & Lessons Learned

Real-world results, what worked, what didn't



Capabilities & Demos

The actual capabilities



Context & Motivation

Context & Motivation



Strategic Context



Curiosity about emerging AI tech



Can it help beyond automation?



DC Network → GARR Backbone



Environment

GARR DC Network:

- 8 data centers
- Multi-vendor
- Already automated



The Question

Can AI agents add intelligence to existing automation?

Why Antares?



Classic Automation

Rigid & Imperative

Is **powerful** but it doesn't understand *intent*

Limited structured input

Unable to process unstructured data

Limited scenarios

Incapable of handling unexpected scenarios

Fixed workflows

No adaptability or learning ability



AI Agent

Flexible

We can state the goal , not the command

Autonomous Reasoning

It decides which tools to run. It synthesizes the results into a human-readable answer

Unified Conversational Interface

A single "co-pilot" that orchestrates all underlying tools (automation, RAG, APIs).



Our Goals



ENHANCE our existing automation with a powerful intelligence layer



SHIFT network troubleshooting operations from imperative commands to intent-based conversations.



BUILD a true "Network Co-pilot" that assists and reasons, not just executes.

Architecture & Concepts

The Building Blocks of Antares



What is an AI Agent?

It's more than just an LLM. It's a system that **reasons (the brain)** and **acts (the hands)**.

- **Brain (LLM):** Understands intent, plans steps, analyzes data.
- **Hands (Tools):** Executes code, calls APIs, runs scripts, searches docs.

This moves us from **imperative execution** ("Run this exact script") to **intent-based reasoning** ("Figure out why this BGP session is down").



What is RAG?

A technique to "ground" an LLM in specific, private knowledge. It's like giving the agent a private library to consult before answering.

It ensures Antares' answers are based on **GARR's reality** (our topology, our procedures), not the public internet. It reduces "hallucinations".



Why LangGraph?

A framework for building complex, stateful agents. Network troubleshooting isn't a simple line ($A \rightarrow B \rightarrow C$)

It allows Antares to **reason step-by-step**:

1. Run a tool (e.g., `check_port_status`).
2. Analyze the result.
3. **Decide the next tool** to run (e.g., `check_acls`).

Antares: A Team of Specialized Agents



The Supervisor:

- It acts as the "project manager".
- It understands your **intent**.
- It **plans** the task and **routes** the query to the correct specialist agent.
- It uses tools for the inventory (e.g. Netbox)



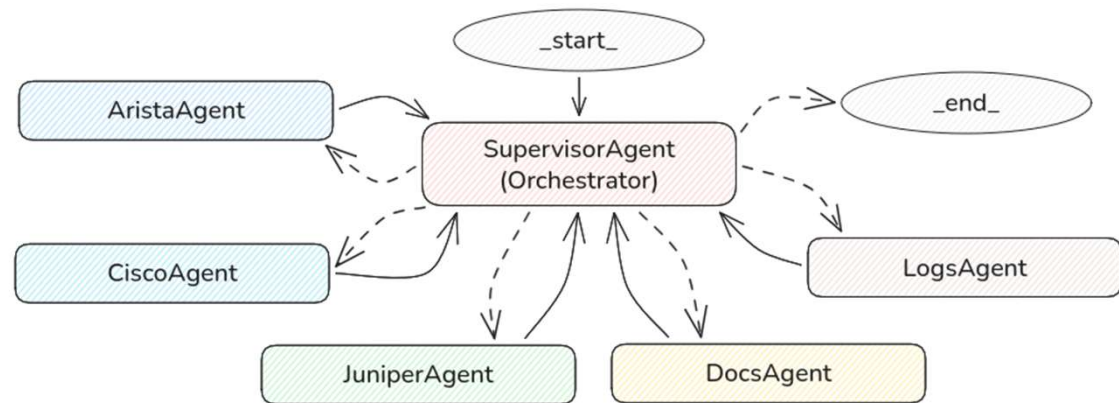
The Specialists:

- **Network Agents** (Arista, Cisco, Juniper)
- **Docs Agent** (RAG for GARR docs)
- **Logs Agent** (OpenSearch)



The Process:

- **LangGraph** enables complex, step-by-step diagnosis
- It's a **cycle**, not a simple script.



Tool Hierarchy & Progressive Disclosure

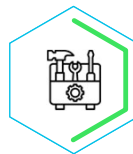


Atomic Tools

Single command, single device

Example: **show_bgp_summary**

- Execute: "show bgp summary"
- Parse JSON/XML
- Return structured data



Composite Tools

Orchestrate multiple atomic tools

Parallel multi-device execution

Example: **bgp_health_check**

- show_bgp_summary (all devices)
- show_bgp_neighbor_detail (if down)
- Aggregate analysis



Skills

Multi-step conditional logic

Domain expertise encoded

Example: **firewall-filter-consistency**

1. Get config (all devices)
2. Resolve prefix-lists
3. Compare term-by-term
4. Severity assessment
5. Generate recommendations

Hide complexity, expose capabilities

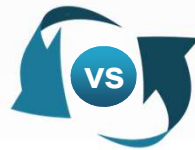
Why Skills? Encoding Expert Workflows

Composite Tool vs. Skill



Composite Tool

- Fixed sequence: $A \rightarrow B \rightarrow C$
- No branching logic
- Good for: "Always do these 3 things"



Skill

- Conditional flow: If X then Y, else Z
- Loop constructs: For each device
- State tracking: Remember what we learned
- Good for: "Expert diagnosis workflow"

Dynamic Discovery

Skills stored as
markdown



Agent loads at
startup



Executes when
query matches

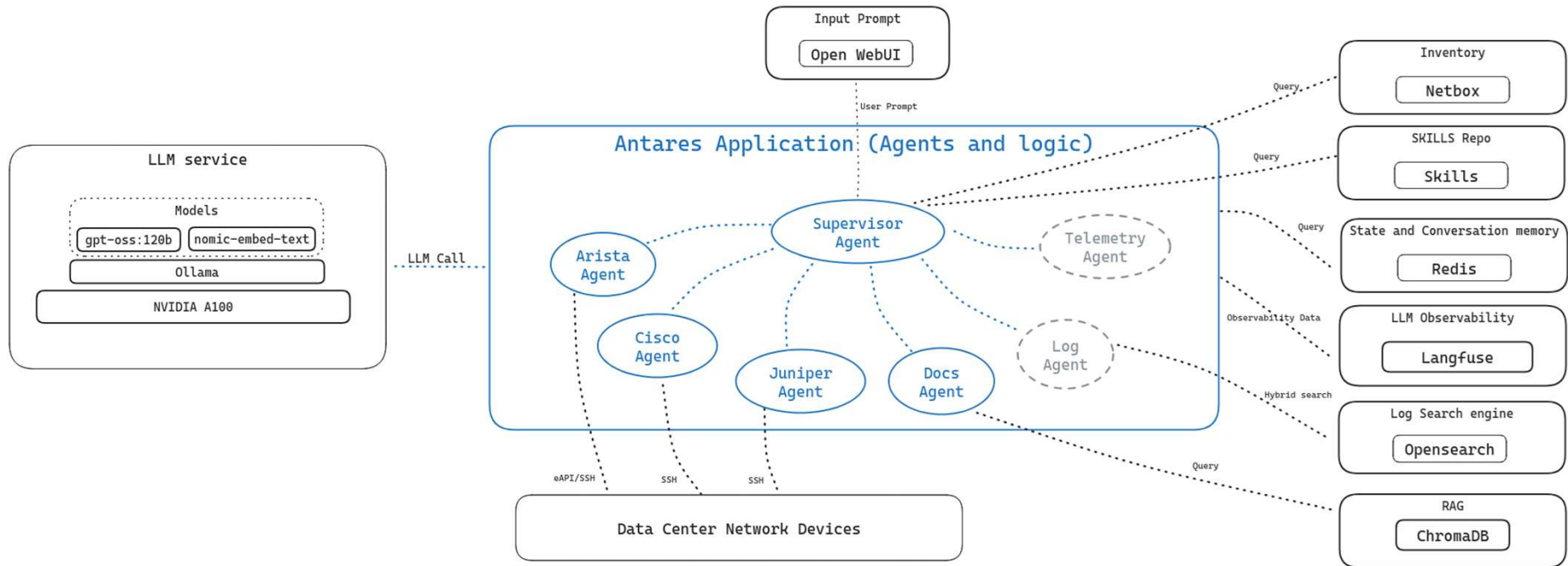


No code
redeployment
needed

Key Innovation:

- Skills are documentation that executes
- Network engineers write workflows in markdown
- Agent interprets and executes
- Version controlled, testable, shareable

Deployment Architecture



Capabilities & Demos

What Antares Can Do Today

Architecture & Coverage



DC Fabric
Arista: 24 Tools



DC Edge
Juniper: 11 Tools



Management
Cisco: 2 Tools

Capability	Arista	Juniper	Cisco
 BGP & Routing	✓	✓	X
 LACP / Port-Channels	✓	⌚	X
 Security Policy	✓	✓	X
 VXLAN Overlay	✓	⌚	X
 Interface Health	✓	✓	✓
 VRF Management	✓	✓	X

Demo | Interface Health & Alarm Analysis

I received alarms on the Bologna edge1 interfaces, can you check?

Antares Multi Agent System
Thought for a minute

Summary of the check on BO1_EDGE1 (10.8.100.26 - Juniper dc-edge, Bologna06)

INTERFACE	ADMIN / OPER	ALARMS / ERRORS	REMARKS
ae70 (aggregate)	up / up	No alarms	Healthy 100 Gbps link
et-0/0/0	up / up	Bit-Error-Seconds = 7 482 (only error counter)	Within normal limits, but worth monitoring
et-0/0/2	up / up	No alarms	100 Gbps, low traffic
et-0/0/3	up / up	No alarms	100 Gbps, low traffic
fxp0 (OOB)	up / up	No alarms	Management interface
lo0 (loopback)	up / up	No alarms	-
xe-0/1/0 - xe-0/1/7 (8 x 10 Gbps)	down / down (admin-disabled)	Alarm flag (device-health check marks disabled ports as "alarm")	Ports are intentionally disabled; no traffic or error counters. This flag can be ignored unless you plan to bring a port into service.

System-wide alarms: None reported (no PSU, fan, temperature, or chassis alarms).

Key take-aways

- The device is **healthy** – all active interfaces are up and show **no active alarms**.
- The only non-zero error metric is the modest **bit-error-seconds** on **et-0/0/0**. It is not yet critical, but you may want to keep an eye on it; a rapid increase could indicate a fiber or transceiver issue.
- The disabled **xe-0/1/** ports are flagged by the health-check tool simply because they are administratively down. This is expected and can be ignored unless you intend to enable those ports.

Next steps / Options

- Monitor et-0/0/0** – watch the bit-error-seconds counter; if it climbs sharply, consider checking the fiber patch panel, transceiver, or cleaning the connector.
- Ignore the disabled-port alarm flag** – it's normal for spare ports that are intentionally shut down.
- Check the second edge device (BO1_EDGE2)** – if you suspect the alarms might be on that device as well, I can run the same diagnostics there.
- Get raw alarm details** – I can pull the exact alarm list from the device if you need the full text of any active alarms.

Let me know which of the above you'd like to proceed with (e.g., run the same check on BO1_EDGE2, fetch raw alarm messages, or anything else).

Thought for less than a second

Send a Message

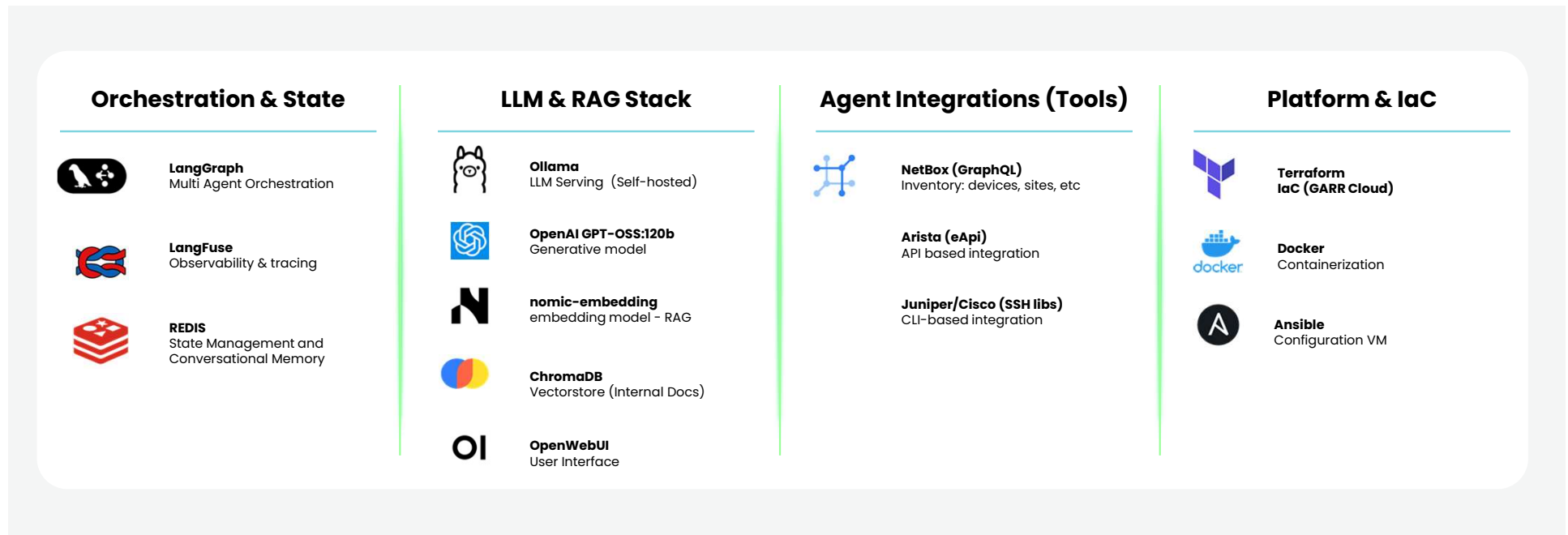
Engineer query:
I received alarms on Bologna edge1 interfaces, can you check?

Antares Response:

- Checked all interfaces autonomously
- Correlated alarms with admin state
- Identified bit-error-seconds for monitoring
- Distinguished spare ports from real issues
- Provided next steps and recommendations

Technical Implementation

Tech Stack



Observability with Langfuse

Why it matters?

Multi-agent execution is complex

- Multiple LLM calls
- Tool chains
- Conditional routing
- Parallel execution

Without visibility → impossible debug

What We Track?



Latency
Every tool, every agent, every LLM



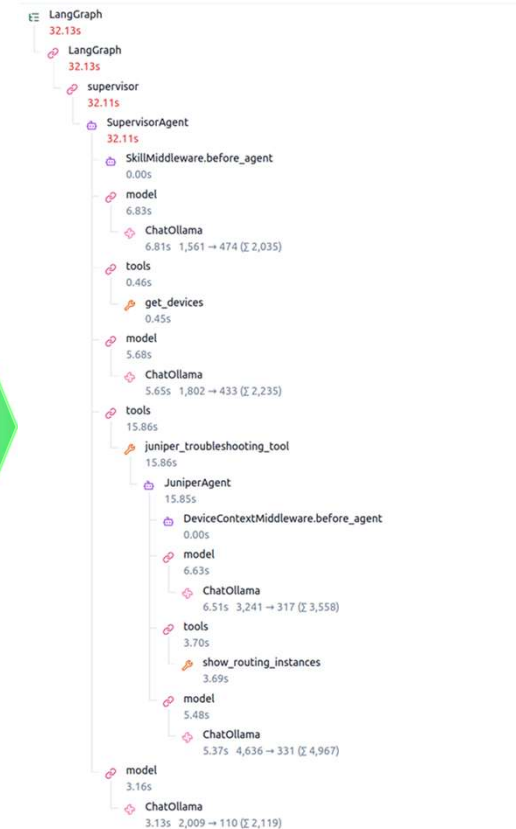
Cost
Token usage, API calls



Success/Failure
Error correlation, stack traces

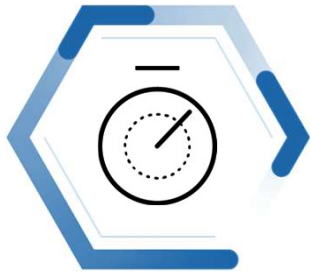


Flow
Agent routing, tool selection



Impact & Lessons Learned

Impact & Benefits



Reduction MTTR

The agent performs in seconds an analysis that would take a human much longer.



From Automation to AIOps

Moving from simply running scripts (automation) to delegating reasoning (intelligence).



Empowerment & Consistency

Makes advanced diagnostics accessible to the entire team (not just senior engineers) while ensuring consistent, repeatable analysis.



Dynamic Knowledge Base

The RAG-powered DocsAgent turns static documentation into an interactive, always up-to-date expert.

Lessons Learned



Architecture

The right architecture/pattern of AI agents has a strong impact, sometimes greater than prompt engineering alone



Observability is not Optional

Without Langfuse: "It failed somewhere" → slow debugging
With observability tools (eg. Langfuse): Exact tool, exact step, exact error → minutes



Hybrid Beats Pure (Code + AI > Either Alone)

Pure LLM: Flexible but unreliable
Pure code: Reliable but rigid
Skills (structured + AI): Best of both worlds



Tool Design is Critical

Parse & structure data; don't dump raw output



Domain Experts Must Drive

They define "good"; AI enables it
Periodic sessions with network engineers = game changer

Thank You!

francesco.lombardo@garr.it

www.garr.it

