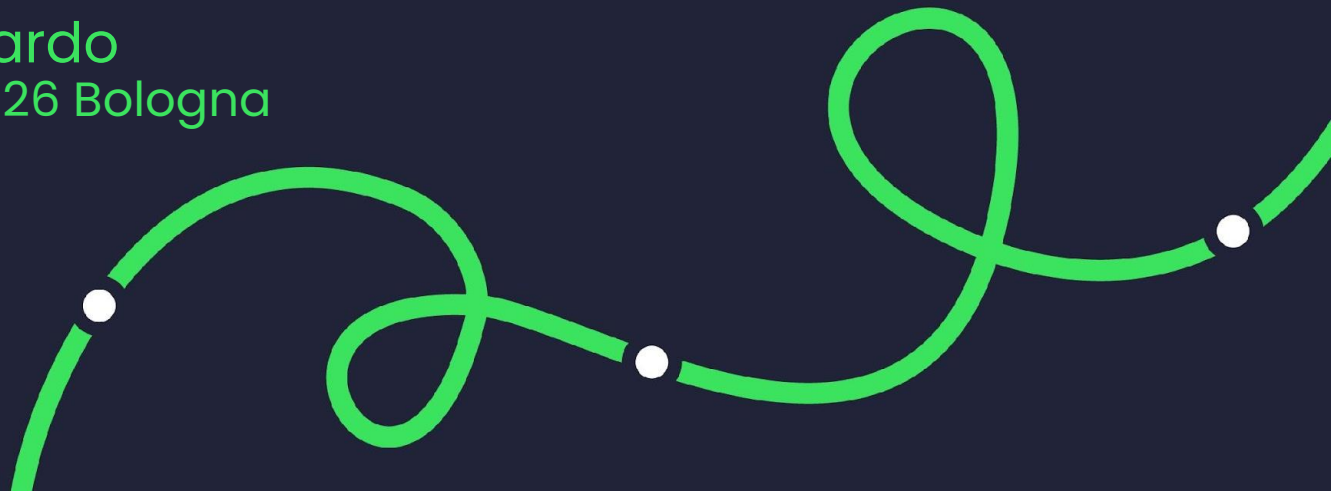
# Antares: Multi-Agent AI for Automated Network Analysis and Troubleshooting in GARR Data Centers

Francesco Lombardo

ITNOG10 - 21 April 2026 Bologna



Consortium GARR

# Agenda

## Context & Motivation

Why AI agents for network operations?

## Technical Implementation

Tech stack, observability

## Architecture & Concepts

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

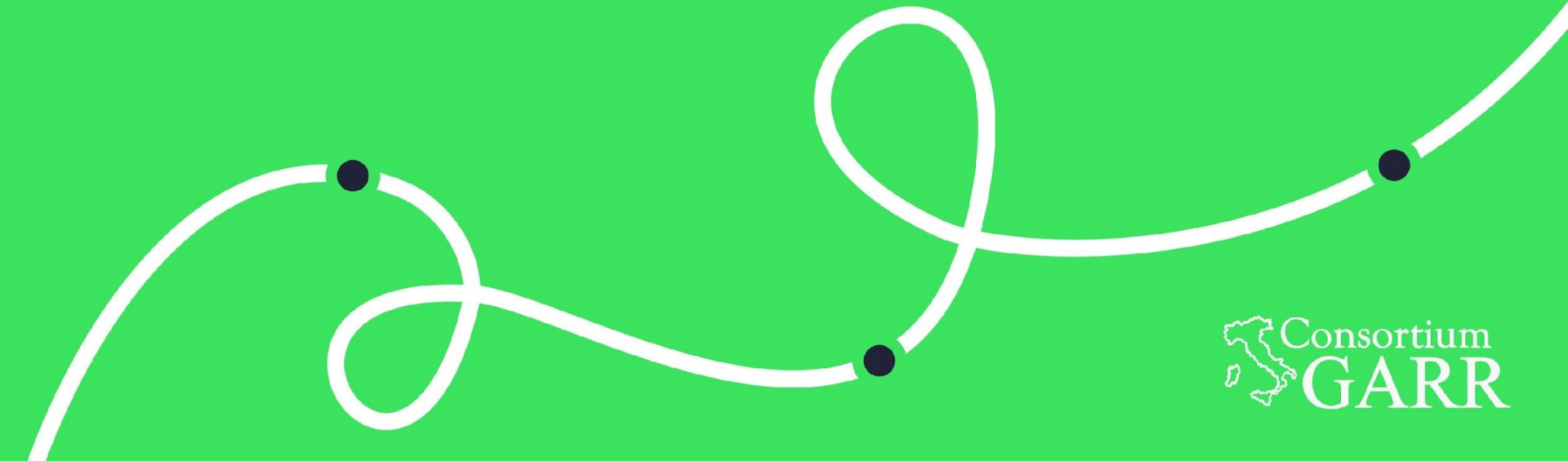
## Impact & Lessons Learned

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

## Capabilities & Demos

The actual capabilities

# Context & Motivation



Consortium GARR

# Context & Motivation

## Strategic Context

- **Curiosity about emerging AI tech**
- **Can it help beyond automation?**
- **DC Network → GARR Backbone**

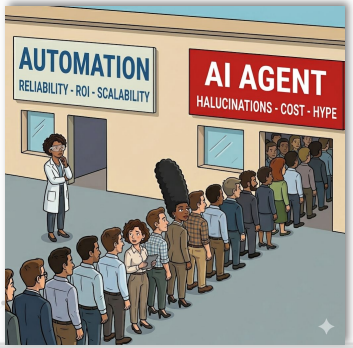
## Environment

**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

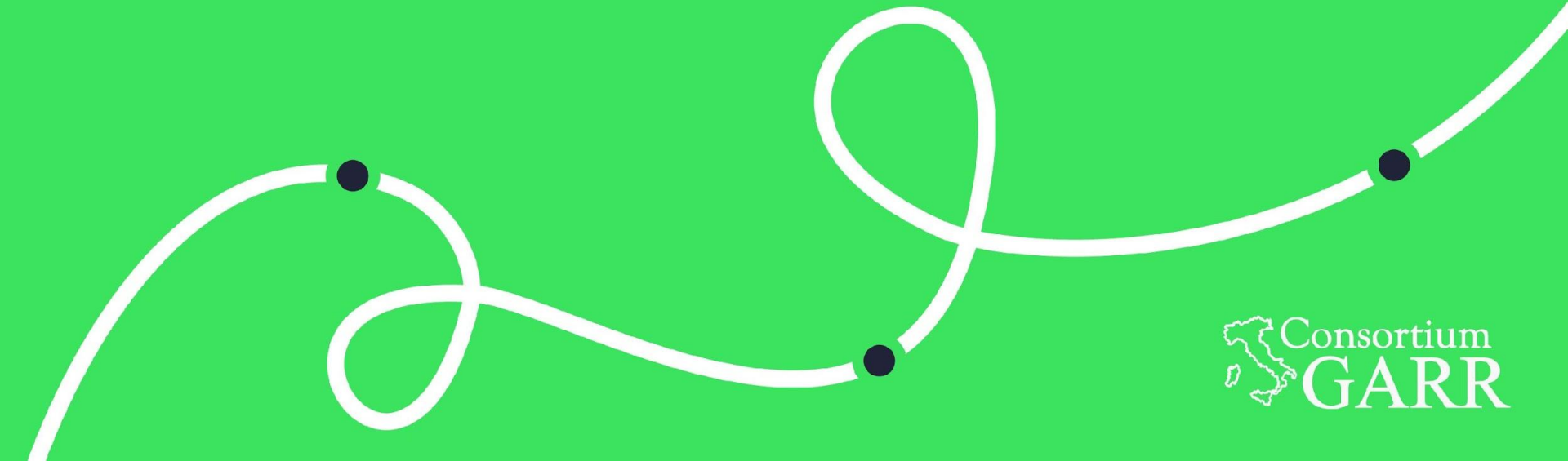
**Unified Conversational Interface**
A single "co-pilot" that orchestrates all underlying tools (automation, RAG, APIs).

## 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.

**Can we get automation reliability WITH AI flexibility?**

# Architecture & Concepts



Consortium GARR

# 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 → B → 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:**

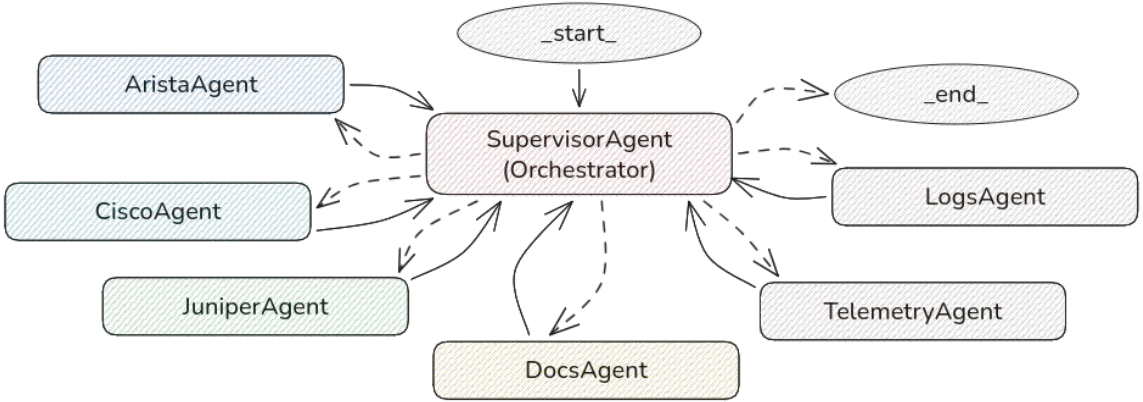
- It acts as the "project manager".
- It understands your **intent**.
- It **plans** the task and **routes** the query to the correct specialist agent.
- It use tools for the inventory (e.g. Netbox)

**The Specialists:**

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

→ Skills are documentation that executes

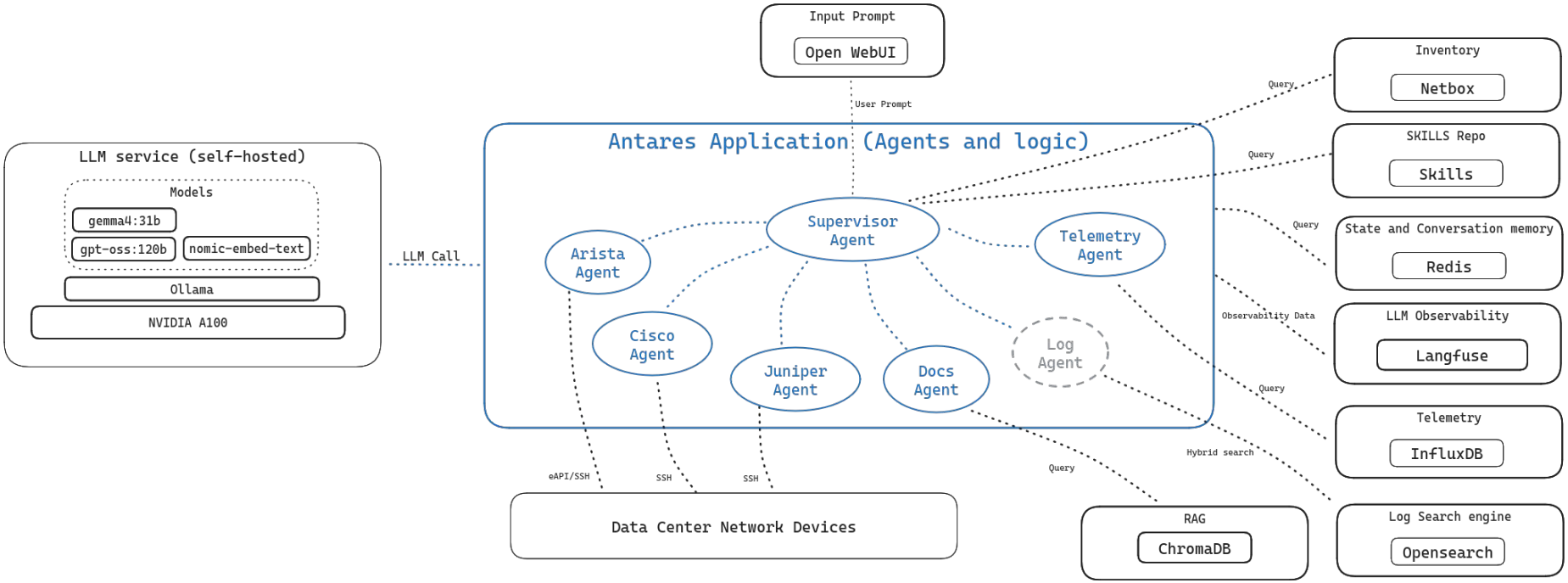
→ Network engineers write workflows in markdown

→ Agent interprets and executes

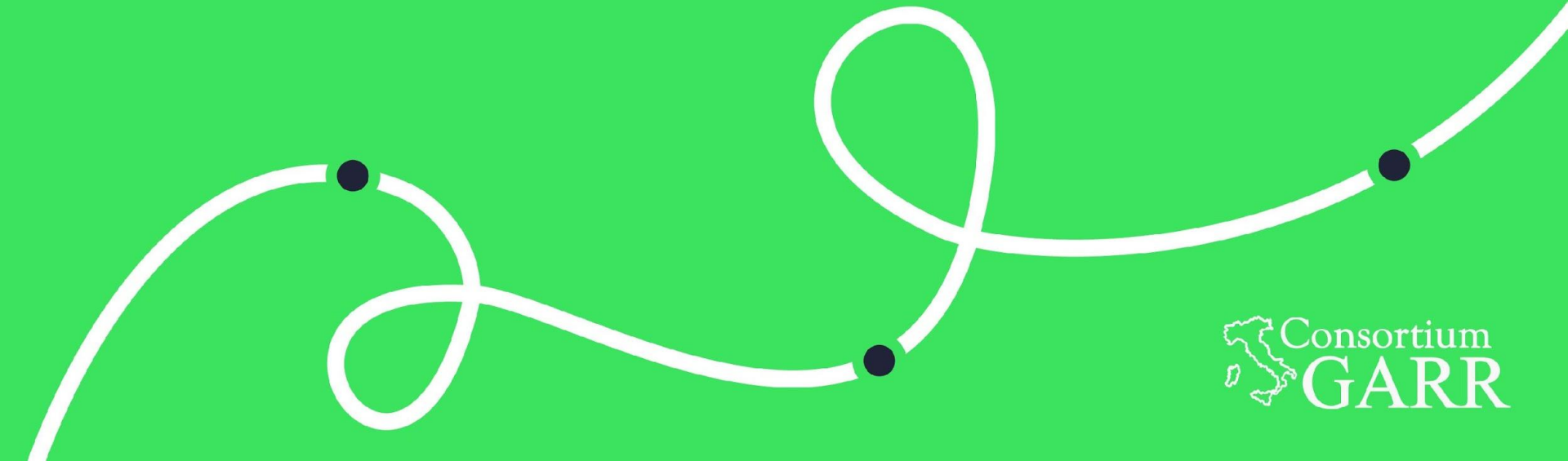
→ Version controlled, testable, shareable

**Hide complexity, expose capabilities**

# Deployment Architecture



# Capabilities & Demos



Consortium GARR

# What Antares Can Do Today

## Architecture & Coverage

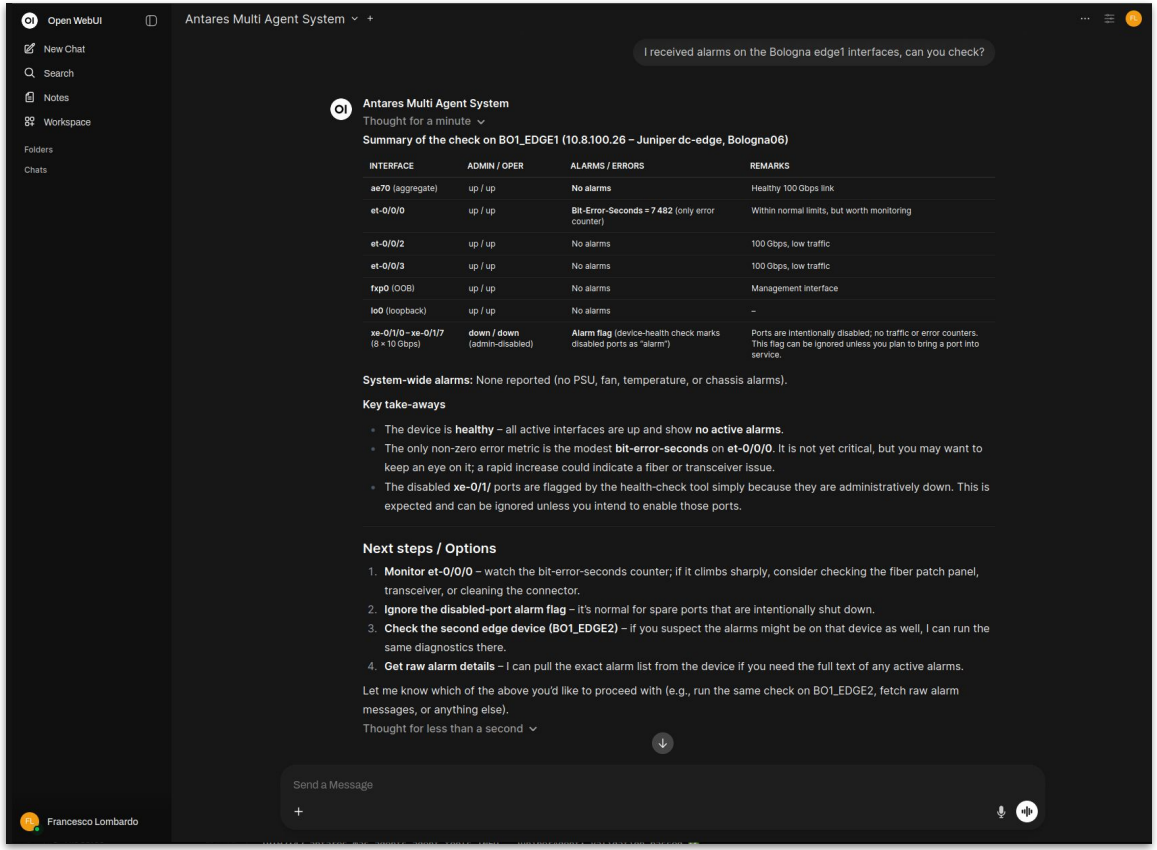
**DC Fabric**
Arista: 25 Tools

**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

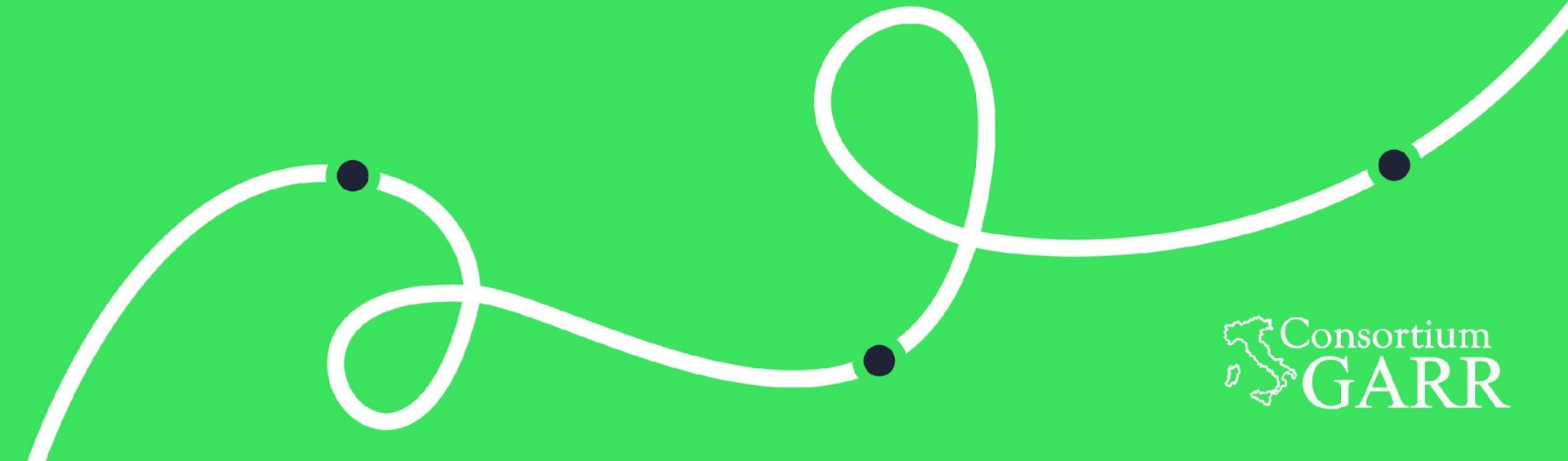



Engineer query:
**I received alarms on Bologna edge1 interfaces, can you check?**

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



Consortium GARR

# Tech Stack

## Orchestration & State

**LangGraph**
Multi Agent Orchestration

**LangFuse**
Observability & tracing

**REDIS**
State Management and Conversational Memory

## LLM & RAG Stack

**Ollama**
LLM Serving (Self-hosted)

**OpenAI GPT-OSS:120b**
Generative model

**Google Gemma 4 - 31b**
Generative model

**nomic-embedding**
embedding model - RAG

**ChromaDB**
Vectorstore (Internal Docs)

**OpenWebUI**
User Interface

## Agent Integrations (Tools)

**NetBox (GraphQL)**
Inventory: devices, sites, etc

**Arista (eApi)**
API based integration

**Juniper (SSH libs)**
CLI-based integration

**Cisco (SSH libs)**
CLI-based integration

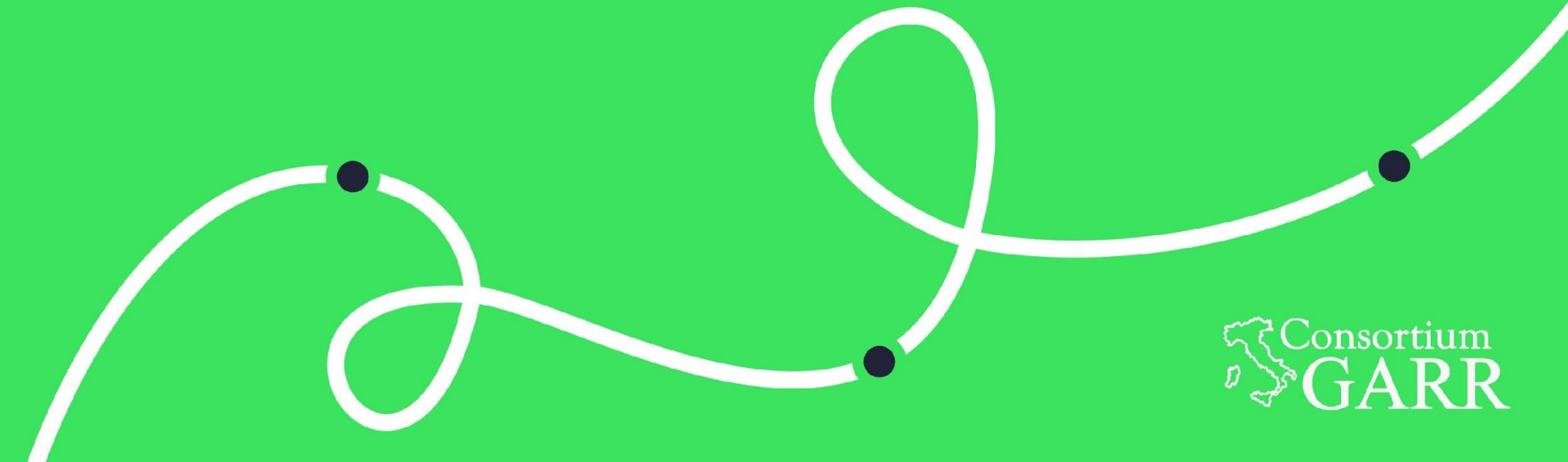
## Platform & IaC

**Terraform**
**IaC (GARR Cloud)**

**Docker**
Containerization

**Ansible**
Configuration VM

# Impact & Lessons Learned



Consortium GARR

# 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

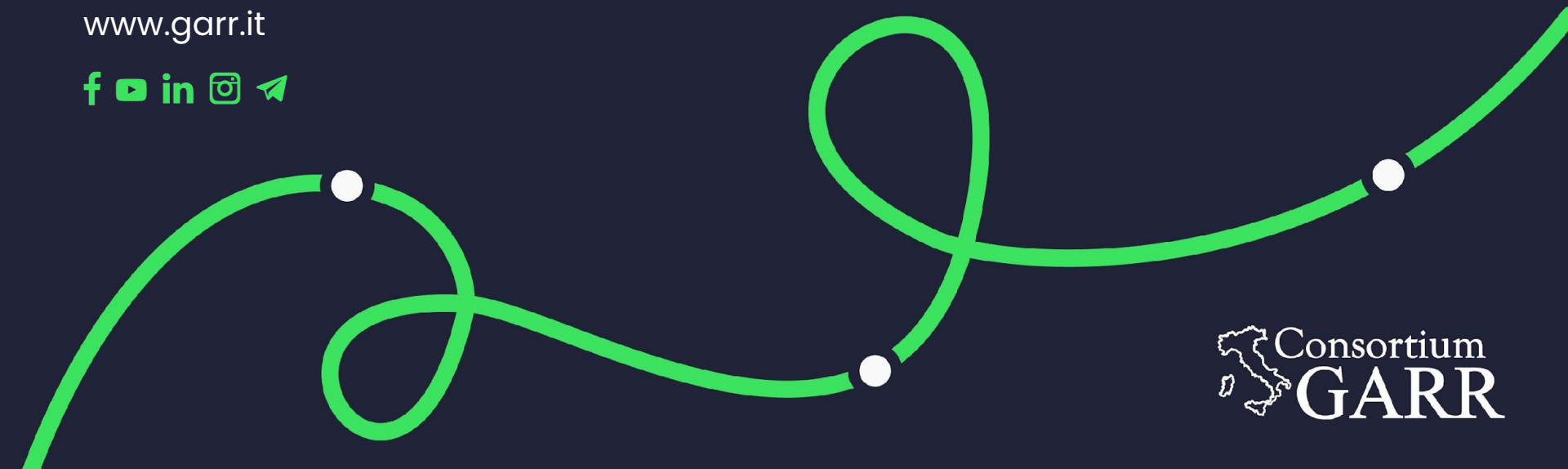
**Tool Design is Critical**
Parse & structure data; don't dump raw output

**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



Consortium GARR