Proceedings of GICITE, Berlin, Germany (6-7th November, 2026)

WINS *Proceedings*

ISSN: 3153-8304

DOI
10.XXXX/ICXXX.2027.000

Proceeding Paper

The Agentic Network

Vishwanath Hiremath

Hewlett Packard Enterprise (HPE)

Spring, Texas, United States of America

hiremath1@ieee.org

***Abstract*—**This keynote manuscript presents the slides and speaker notes of a 12-minute GICITE 2026 keynote on agentic networks. It argues that the human decision loop between detection and action has become the bottleneck in network operations, as scale, streaming telemetry and the speed of faults and attacks outpace tickets and change windows. It places agents on a path from scripted automation through intent-based networking to agentic operation, describes a network agent as a closed loop that perceives, reasons, acts and verifies, and outlines a five-layer reference architecture of intent and policy, specialist agents, typed tools over the Model Context Protocol, telemetry with a digital twin, and the network itself. It then applies zero-trust principles to the agents themselves, matches autonomy to risk, and proposes a crawl, walk, run adoption path.

***Keywords*—**agentic networking; AI agents; zero trust; Model Context Protocol; intent-based networking; digital twin

Published: 6 November 2026

From automated to autonomous: AI agents that operate networks safely
Keynote — GICITE 2026, Technische Universität Berlin, November 6–7, 2026

Format: 12 slides, 12-minute keynote. Each section lists the slide content followed by the speaker notes.

I. Slide 1: GICITE 2026 cover

A. On the slide

- **Conference cover page:** Germany India Conference on Innovation, Technology, and Engineering (GICITE) 2026, November 6–7, 2026, Technische Universität Berlin

B. Speaker notes

No speaker notes. This is the conference cover slide shown while you are introduced.

II. Slide 2: The Agentic Network

A. On the slide

- **Subtitle:** From automated to autonomous — AI agents that operate networks safely
- Keynote — Vishwanath Hiremath, Principal System Engineer, Hewlett Packard Enterprise (Juniper Networks)
- IEEE Senior Member · hiremath1@ieee.org

B. Speaker notes

Good morning, everyone, and thank you to the GICITE organizers for having me. I'm Vishwanath Hiremath, and I work at Hewlett Packard Enterprise in the Juniper Networks business. I validate data center and service provider networks for some of the largest operators in the world.

Today's topic is the Agentic Network. I want to cover three things in the next twelve minutes. First, why the way we operate networks is hitting a wall. Second, what an agentic network actually looks like inside. Third, and most important, how we make it safe enough to trust with production traffic.

The short version: agents can close the loop between what the network is telling us and what we do about it. But only if we treat them with the same zero-trust discipline we apply to any other identity on the network.

III. Slide 3: Networks now change faster than people can respond

A. On the slide

- **Scale:** AI fabrics, multi-cloud and SD-WAN multiply the devices, paths and policies to manage
- **Signal:** Streaming telemetry produces more data per second than any team can read
- **Speed:** Faults and attacks unfold in seconds; tickets and change windows take hours
- **Key message:** The bottleneck is no longer bandwidth. It is the human decision loop.

B. Speaker notes

Let's start with the problem. Three forces are pushing networks past what human-paced operations can handle.

Scale: AI training fabrics, multi-cloud, and SD-WAN have multiplied the number of devices, paths, and policies an operations team owns.

Signal: we did the right thing and moved to streaming telemetry — gNMI, sFlow, BGP monitoring. But now we're drowning in it. The data exists; nobody has time to read it.

Speed: a fiber degrading, a BGP leak, or a DDoS unfolds in seconds. Our response still runs through tickets, chat threads, and change windows measured in hours.

So the bottleneck isn't bandwidth or hardware anymore. It's the human decision loop sitting between detection and action. That is exactly the gap agents are designed to fill.

© 2026 The Authors. Published in the Proceedings of GICITE 2026.

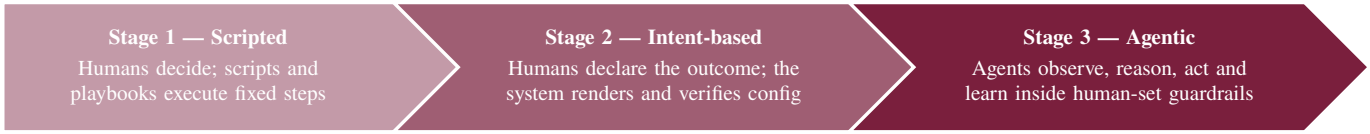


Fig. 1. From automation to agency (Slide 4).

IV. Slide 4: From automation to agency

A. On the slide

- **Stage 1 — Scripted:** Humans decide; scripts and playbooks execute fixed steps
- **Stage 2 — Intent-based:** Humans declare the outcome; the system renders and verifies config
- **Stage 3 — Agentic:** Agents observe, reason, act and learn inside human-set guardrails
- **Key message:** Automation follows instructions. An agent pursues a goal and chooses its own next step.

B. Speaker notes

It helps to place agents on a timeline.

Stage one is scripted automation: Ansible, Python, playbooks. Very useful, but a human still makes every decision; the script just types faster.

Stage two is intent-based networking. We declare the outcome we want — "these two tenants never talk" — and the system renders and verifies the configuration. Most mature operators are here today.

Stage three is agentic. The agent doesn't just execute an instruction. It's given a goal, it watches the network, reasons about what's wrong, picks the next action, checks the result, and adjusts.

Fig. 1 places the three stages on one timeline.

The key distinction: automation follows instructions; an agent pursues a goal. That shift is powerful — and it's also exactly why the trust model has to change, which we'll come to.

V. Slide 5: Anatomy of a network agent

A. On the slide

- **Perceive:** gNMI, sFlow, BMP, syslog, flow and topology state
- **Reason:** LLM planner plus domain models correlate and diagnose
- **Act:** Calls approved tools — query, simulate, change, roll back
- **Verify:** Confirms the intent holds and learns from the outcome
- **Key message:** A closed loop that runs continuously, and every action is checked against intent.

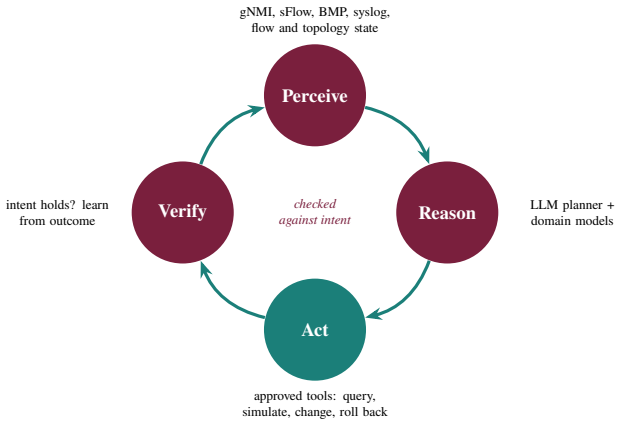


Fig. 2. Anatomy of a network agent (Slide 5): a closed loop that runs continuously.

B. Speaker notes

So what is a network agent, concretely? It's a closed loop with four stages.

Perceive: the agent consumes the telemetry we already stream — gNMI counters, sFlow samples, BGP monitoring, syslog, and topology state.

Reason: a language-model planner, combined with domain models that understand routing protocols and fabric behavior, correlates those signals and forms a diagnosis. For example: these FEC errors on this aggregate bundle line up with that RSVP-TE reoptimization and those IS-IS flaps.

Act: the agent doesn't get raw CLI access. It calls a defined set of tools — query state, run a what-if simulation, push a scoped change, or roll back.

Verify: it checks that the intent still holds after the action, and feeds the outcome back so the next decision is better.

The important part is the last stage. Every action is validated against intent before the loop closes.

Fig. 2 draws the four stages as a loop.

VI. Slide 6: A layered reference architecture

A. On the slide

- **Intent & policy:** Operator goals, SLOs, guardrails and approval rules
- **Agent layer:** Planner agent coordinating specialists — fabric, routing, security, capacity
- **Tool layer (MCP):** Typed, permissioned tools over NETCONF, gNMI and controller APIs
- **Telemetry & twin:** Streaming data lake plus a digital twin for safe what-if runs

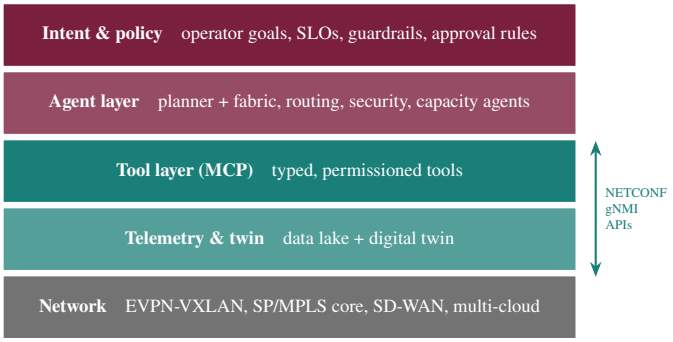


Fig. 3. The layered reference architecture of Slide 6. The agent never touches a device directly.

- **Network:** EVPN-VXLAN fabrics, SP/MPLS core, SD-WAN edge, multi-cloud

B. *Speaker notes*

Here is how we structure it in practice, as five layers.

At the top, intent and policy: what the operator wants, the service-level objectives, and the guardrails — what an agent may and may not do without approval.

Below that, the agent layer. Rather than one giant agent, a planner coordinates specialists: a fabric agent, a routing agent, a security agent, a capacity agent. Smaller scopes are easier to test and easier to constrain.

Then the tool layer. This is where the Model Context Protocol is becoming the standard. Every capability is exposed as a typed, permissioned tool sitting on top of NETCONF, gNMI, or controller APIs. The agent never touches a device directly.

Under that, telemetry and a digital twin, so an agent can simulate a change before it happens.

And at the bottom, the real network: EVPN-VXLAN data centers, the MPLS core, SD-WAN, and cloud.

Fig. 3 shows the five layers. A review of digital twins and their implementation, in the context of supply chains, is given in [1].

VII. Slide 7: Where agents earn their place first

A. *On the slide*

- **Incident triage (read-heavy):** Correlate IGP flaps, LSP reoptimization and optics errors into one root cause
- **Application-aware segmentation (read-heavy):** Learn workload behavior and propose EVPN-VXLAN segments automatically
- **DDoS response (acts on network):** Detect attack traffic and divert it at the MPLS core in seconds
- **Change validation (acts on network):** Simulate, test and verify every migration before and after it lands

B. *Speaker notes*

Where should you start? Four use cases stand out, and all four come from problems I see in real customer networks.

Incident triage. In a large backbone, one fault shows up as IS-IS adjacency drops, RSVP-TE reoptimization, and FEC errors on a bundle — three alarms, three teams. An agent correlates them into one root cause in minutes.

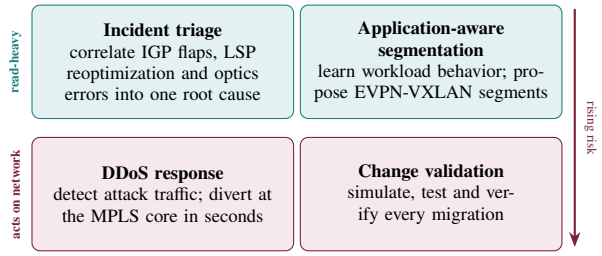


Fig. 4. Where agents earn their place first (Slide 7): trust is built on diagnosis before control is handed over.

Application-aware segmentation. Instead of hand-writing micro-segments, an agent learns workload behavior and proposes EVPN-VXLAN segments for approval.

DDoS response. Detect the attack and divert traffic at the MPLS core in seconds, not after a bridge call.

Change validation. Run the change against a digital twin, execute it, then verify the intent still holds.

Notice the pattern: the first two are read-heavy and low-risk. That's deliberate. You build trust on diagnosis before you hand over the controls.

Fig. 4 sorts the four use cases by risk.

VIII. Slide 8: Zero trust, applied to agents

A. *On the slide*

- **Identity:** Every agent and tool call is authenticated and attributable
- **Least privilege:** Tools scoped per agent, per task and per time window
- **Policy gate:** Each action checked against intent and blast radius before it runs
- **Audit & rollback:** Every decision logged with its reasoning, every change reversible

B. *Speaker notes*

Now the most important slide. An agent that can change your network is a new kind of privileged identity, and it introduces new attack surfaces: prompt injection through a log line, a poisoned tool description, an over-permissioned token.

Security issues and digital forensics for industrial IoT are discussed in [2].

My answer, and the subject of my recent research on zero-trust security for MCP-based agentic systems, is to apply zero trust to the agents themselves.

Identity: every agent and every tool call is authenticated and attributable. No shared service accounts.

Least privilege: tools are scoped per agent, per task, and per time window. The triage agent can read everything and change nothing.

Policy gate: before any action executes, an independent policy engine checks it against intent and estimated blast radius.

Audit and rollback: every decision is logged with the reasoning behind it, and every change has a tested rollback.

TABLE I
ZERO TRUST APPLIED TO AGENTS

Control	What it means for an agent
Identity	Every agent and tool call authenticated and attributable; no shared service accounts
Least privilege	Tools scoped per agent, per task and per time window
Policy gate	Independent policy engine checks each action against intent and blast radius
Audit & rollback	Every decision logged with its reasoning; every change reversible

TABLE II
AUTONOMY SHOULD MATCH RISK

Action class	Examples	Agent may
Observe	Query state, correlate alarms	Act freely
Simulate	What-if runs on the digital twin	Act freely
Low-risk change	Drain a link, adjust a QoS queue	Act, then notify
High-impact change	Routing policy, segmentation, core reroute	Propose; human approves

Actions move up a row only as an agent builds an evidenced track record.

Never trust, always verify — including your own AI.

A structured review of zero-trust architecture, its effectiveness and its implementation challenges is given in [3]. Table I lists the four controls of Slide 8.

IX. Slide 9: Autonomy should match risk

A. On the slide

- **Observe (query state, correlate alarms):** Act freely
- **Simulate (what-if runs on the digital twin):** Act freely
- **Low-risk change (drain a link, adjust a QoS queue):** Act, then notify
- High-impact change (routing policy, segmentation, core reroute): Propose; human approves
- Move actions up a row only as an agent builds an evidenced track record.

B. Speaker notes

Zero trust tells us who an agent is and what it can touch. This slide is about how much freedom it gets, and the rule is simple: autonomy should match risk.

Observing — querying state, correlating alarms — the agent can do freely. There's no blast radius.

Simulating on a digital twin is also free. Let it try a hundred options.

Low-risk, easily reversible changes — draining a link, adjusting a QoS queue — the agent can execute and then notify the team.

High-impact changes — routing policy, segmentation, rerouting the core — the agent prepares the change, the evidence, and the rollback, and a human approves.

Over time, as an agent builds a track record, you move specific actions up a row. That's how trust is earned: action by action, with evidence, not by flipping a switch.

Table II summarises the rule.

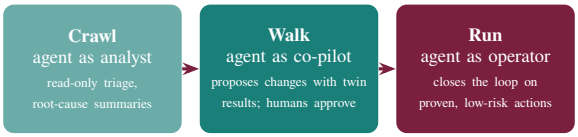


Fig. 5. A practical adoption path (Slide 10).

X. Slide 10: A practical adoption path

A. On the slide

- **Crawl — Agent as analyst:** Read-only triage and root-cause summaries for the NOC
- **Walk — Agent as co-pilot:** Proposes changes with twin results; humans approve
- **Run — Agent as operator:** Closes the loop on proven, low-risk actions on its own
- **Key message:** Start where the data already is — your telemetry pipeline and your incident history.

B. Speaker notes

How do you get there without betting the network? Crawl, walk, run.

Crawl: the agent as analyst. It's read-only. It watches telemetry, triages incidents, and writes root-cause summaries for the NOC. Zero risk, and immediately useful — mean time to diagnose drops.

Walk: the agent as co-pilot. It proposes changes, attaches the digital-twin results and the rollback plan, and an engineer approves. This is where you build the evidence base.

Run: the agent as operator, closing the loop by itself — but only on the specific, proven, low-risk actions it has earned.

And my practical advice: start where your data already is. If you have a streaming telemetry pipeline and a history of incidents, you already have what you need to begin the crawl phase next quarter.

Fig. 5 shows the three phases.

XI. Slide 11: Three things to take home

A. On the slide

- The human decision loop is now the bottleneck; agents close it
- **Give agents tools, not keys:** typed, scoped and policy-gated
- Earn autonomy action by action, with evidence and rollback
- Thank you — Vishwanath Hiremath, hiremath1@ieee.org

B. Speaker notes

Let me close with three things to take home.

One: the network's bottleneck is now the human decision loop. Agents are how we close it.

Two: give agents tools, not keys. Typed, scoped, policy-gated tools over MCP — never raw device access.

Three: earn autonomy one action at a time, with evidence and a rollback for everything.

The agentic network isn't about replacing network engineers. It's about letting engineers spend their time on design and intent, while agents handle the second-by-second work, safely.

Thank you very much. I'm happy to take questions, and you can reach me at hiremath1@ieee.org.

XII. Slide 12: Contact details

A. On the slide

- Vishwanath Hiremath
- Independent Researcher
- Principal System/Software Engineer (HPE)
- IEEE Senior Member | IETE Fellow | SCRS Distinguished Fellow
- **Email:** hiremath1@ieee.org
- **LinkedIn:** www.linkedin.com/in/vishwanath-hiremath-198a918

B. Speaker notes

Leave this slide up during Q&A. Invite attendees to connect on LinkedIn or reach out by email to continue the conversation on agentic networking and zero-trust security for AI agents.

References

[1] K. Soliman, "Digital Twins for Sustainable Supply Chains in the Gulf Cooperation Council: A Critical Review and Implementation Roadmap," J. Comput. Mech. Manag., vol. 5, no. 4, pp. 176–189, Aug. 2026.

[2] V. V. R. G. Saigopal and V. Raju, "IIoT Digital Forensics and Major Security issues," in 2020 International Conference on Computational Intelligence, ICCI 2020, 2020. doi: 10.1109/ICCI51257.2020.9247685.

[3] M. Shahnawaj, H. I. Hellol, D. Biswas, M. A. Ridwan, R. Uddin, and S. Akhtar, "Zero Trust Architecture: A Structured Review of Effectiveness, Implementation Challenges, and Component-Level Security Contributions," J. Comput. Mech. Manag., vol. 5, no. 4, pp. 190–203, Aug. 2026.