Agentic WAN-Native LLM Orchestration: From User-Driven Inference to Geographically Distributed Model Instances Across Internet2, PNRP, and SCinet via NRP and SENSE

Submitted by Mohammad Firas Sada (mfsada@es.net)
on behalf of the involved parties listed herein

SC26 Network Research Exhibition

1 Abstract

The National Research Platform (NRP) AI Ecosystem provides a federated, Kubernetes-native environment for LLM inference and serving, hosting a rotating catalog of open-weights LLMs accessible via an OpenAI-compatible API and Envoy AI Gateway across 80+ member institutions. At SC26 we extend this ecosystem to the wide-area network by orchestrating dynamic, QoS-enabled optical Layer 2 paths between three distributed anchor points: the SCinet cluster on the exhibit floor, the Prototype National Research Platform (PNRP) distributed across SDSC, MGHPCC, and UNL, and Internet2's national backbone. Using the ESnet SENSE Orchestrator and the NRP's native SENSE Kubernetes Operator, this NRE provisions deterministic, high-bandwidth connectivity for live, multi-site LLM inference and agentic multi-turn workflows: cross-site generation, retrieval-augmented generation (RAG) over a shared embedding store, and pipeline-parallel experiments that measure how WAN latency shapes end-to-end serving performance. By integrating SCinet into the NRP's coast-to-coast AI research fabric, attendees interact with models that physically span three administrative domains.

2 Goals

- **WAN-orchestrated LLM serving**: route live inference requests from SCinet-attached Nautilus nodes to LLM backends on PNRP (SDSC, MGHPCC, UNL) and Internet2-connected caches, with real-time path control via SENSE.
- **Deterministic QoS for AI workloads**: dedicated 100 Gbps Layer 2 VLANs with guaranteed bandwidth between distributed inference endpoints, overcoming best-effort internet variability.
- **Multi-admin LLM serving**: NRP Envoy AI Gateway + vLLM/SGLang backends across administrative domains; per-model concurrency policies and scavenging replicas (priority −10) giving zero-reservation, low-preemption WAN access for AI experiments.
- **Live attendee-driven experiments**: cross-site model inference, multi-turn agent workflows, pipeline-parallelism and WAN-latency exploration, and RAG over a shared embedding store, spanning SCinet, PNRP, and Internet2 resources.
- **Validating the SENSE Kubernetes Operator**: building on prior cross-testbed integrations with FABRIC and AutoGOLE toward programmable, intent-based WAN provisioning for LLM services.
- **Quantifying the payoff**: collect and report measured bandwidth, latency, and jitter per circuit alongside end-to-end request latency, so attendees and reviewers can compare SENSE-provisioned paths against best-effort routing on the same workload.

3 Resources

- SCinet-connected Nautilus nodes on the exhibit floor, federated into the NRP production Kubernetes cluster.
- PNRP distributed infrastructure: GPU clusters at SDSC, MGHPCC, UNL, with data caches embedded in the Internet2 backbone.
- Managed LLM catalog (see §7) via https://ellm.nrp-nautilus.io/v1.
- ESnet SENSE Orchestrator + NRP SENSE Operator for on-demand multi-domain L2 provisioning (VLAN tagging, QoS guarantees).
- High-speed WAN circuits: dedicated 100 Gbps optical paths SCinet (Chicago) → StarLight → Internet2 → PNRP (San Diego, Boston, Lincoln).
- SENSE-native telemetry: circuit-level QoS visibility from the orchestrator (per-request path, jitter, utilization), mirrored into the cluster via the operator's status API.

4 Infrastructure

4.1 NRP AI ecosystem

Federated, Kubernetes-native LLM inference/serving across 80+ member institutions; a rotating catalog of open-weights models exposed via an OpenAI-compatible API and the Envoy AI Gateway; per-namespace API keys and fair-use concurrency (no per-token billing). Hosted Open WebUI at https://nrp-openwebui.nrp-nautilus.io for attendee access. Access is contributor-first: institutions contribute GPUs and keep priority on their hardware, and idle capacity is opportunistically available to the full federation. The WAN abstracts the same elasticity at 100 Gbps.

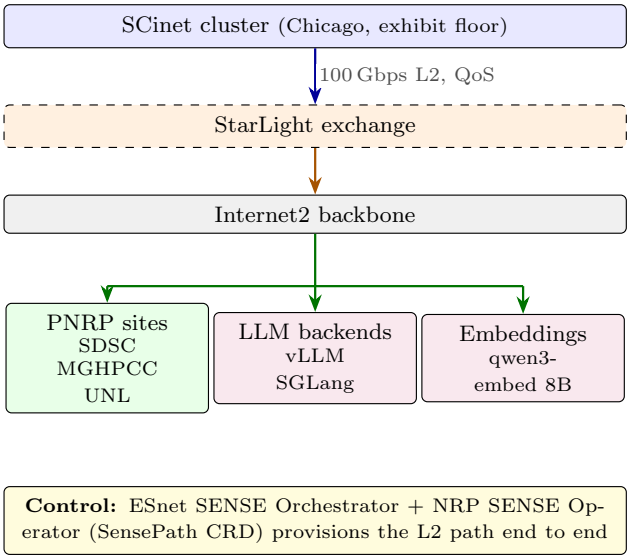


Figure 1: WAN-native LLM architecture: SCinet → StarLight/Internet2 → PNRP backends.

4.2 Serving stack

The managed service is OpenAI-compatible at `https://ellm.nrp-nautilus.io/v1`, with live Prometheus/vLLM metrics on the NRP LLM Status page. Per-model concurrency limits are enforced under NRP's Fair Use Policy (SDSC and Internet2 affiliates receive higher limits); the catalog rotates as the open-weights frontier moves. The NRP also supports self-hosting via vLLM or SGLang.

4.3 Agentic multi-turn workflows

Attendee prompts traverse reason–act–observe loops in which an agent scheduled across the NRP Kubernetes cluster decomposes a request, calls backend models over SENSE-provisioned paths, and synthesizes a response from distributed model instances. The NRP platform already runs this pattern for its own operations (node diagnosis, pod troubleshooting, and accounting queries) and exposes the same mechanisms to attendees.

4.4 RAG over a shared embedding store

Retrieval is served by the NRP embedding catalog (`qwen3-embedding`, 8B) and the NRP managed vector database (**Milvus**; `https://nrp.ai/documentation/userdocs/ai/vector-database`), so retrieval and generation can be placed on different sites and queried across the WAN, making retrieval latency a visible, measurable part of the end-to-end path.

5 Impact

First production LLM serving ecosystem directly orchestrated over a programmable, multi-domain WAN using ESnet SENSE. Unlike static VPNs or best-effort internet, SENSE provides deterministic bandwidth and latency, letting AI researchers treat the WAN as a schedulable resource. The intent-based model the application expresses ("I need 100 Gbps of guaranteed path between these two sites") is translated by the orchestrator into concrete switched circuits across independent domains, rather than negotiated over months with a NOC. Federating SCinet, PNRP, and Internet2 into a single logical AI cluster shows distributed inference and training can reach near-local performance across continental distances. For institutions without on-premise GPU clusters the NRP managed LLM service becomes a WAN-native capability: the compute is pooled and federated, and the bandwidth to reach it is now schedulable too. It directly addresses the SC26 themes of *network-intensive scientific workflows* and *AI-driven distributed applications*, and engages Chicago's FinTech and healthcare communities through ultra-low-latency, real-time AI demonstrations. The approach inherits the multi-domain lineage of AutoGOLE/SENSE (GNA-G), where service creation is delivered in minutes rather than days.

6 Interactive Experience at SC26

Attendees query LLMs via a live web interface or programmatic API; requests routed in real time over SENSE-provisioned paths to backends. Dashboards (Grafana, Prometheus, Observable) show per-request path selection and QoS metrics (bandwidth, latency, jitter), live vLLM metrics (GPU utilization, KV cache occupancy, request latency), and the SENSE Orchestrator topology view of active VLAN paths. Attendees can also issue their own `kubectl apply -f sensepath.yaml` to instantiate a dedicated circuit and watch it appear in the topology view in seconds, making the "network as a schedulable resource" claim tangible rather than descriptive.

7 Model Catalog (verified at `nrp.ai/llms`, Oct 2026)

Model	Params	Ctx	Notes
qwen3	180B (6B act)	1M	Flagship MoE, agentic coder
qwen3-small	27B	1M	Low-latency compact
gpt-oss	120B	131K	LTS candidate
gemma	31B	262K	Multimodal
gemma-small†	12B	262K	Audio/ASR
kimi†	1T	131K	Moonshot coding
glm-5†	753B	1M	NVFP4
deepseek-v4-flash†	552B (16B act)	1M	Top intelligence index
minimax†	428B (23B act)	1M	Sparse-attention MoE
qwen3-embedding	8B	262K	Embeddings only

Table 1: Rotating open-weights catalog served by the NRP AI ecosystem. †Under evaluation at the time of writing.

8 Involved Parties

ESnet / LBNL: Mohammad Firas Sada (`mfsada@es.net`); Tom Lehman (`tlehman@es.net`); Xi Yang (`xiyang@es.net`); Justas Balcas (`jbalcas@es.net`); Alberto Jimenez (`ajimenez@es.net`); Kiran Vasu (`kvasu@es.net`); Chin Guok (`chin@es.net`); Inder Monga (`inder@es.net`); Dmitry Mishin (`dimm@lbl.gov`); **UCSD / SDSC:** John Graham (`jjgraham@ucsd.edu`); Frank Würthwein (`fkw@ucsd.edu`); Tom DeFanti (`tdefanti@ucsd.edu`); Tom Hutton

(hutton@sdsc.edu); Mahidhar Tatineni (mahidhar@sdsc.edu); Daniel Diaz (d4diaz@ucsd.edu); Chris Hughes (crhughes@sdsc.edu); Aashay Arora (aaarora@ucsd.edu); Diego Davila (didavila@ucsd.edu); **RNP:** Marcos Felipe Schwarz (marcos.schwarz@rnp.br); **iCAIR / Northwestern:** Joe Mambretti (j-mambretti@northwestern.edu); Jim Hao Chen (jim-chen@northwestern.edu); Fei Yeh (fyeh@northwestern.edu); **Caltech:** Harvey Newman (newman@hep.caltech.edu); Preeti Bhat (preeti@caltech.edu); Andres Moya (amoya@caltech.edu); **Fermilab:** Phil Demar (demar@fnal.gov); Asif Shah (sshah@fnal.gov); Dmitry Litvintsev (litvinse@fnal.gov); **UMass:** Eduardo Bach (eduardobach@umass.edu); Rafael Coelho Lopes de Sa (rclsa@umass.edu); Jessa Westclark (jessicaw@umass.edu); **UNL:** Derek Weitzel (dweitzel@unl.edu); Huijun Zhu (huijun.zhu@unl.edu); Sam Albin (sam.albin@unl.edu); Ashton Graves (agraves10@unl.edu); **FIU:** Jeronimo Bezerra (jbezerra@fiu.edu); Julio Ibarra (julio@fiu.edu); **Internet2:** Chris Cox (ccox@internet2.edu); James Deaton (jdeaton@internet2.edu); Matt Zekauskas (matt@internet2.edu); Christian Hitchcock (chitchcock@internet2.edu); Karl Newell (knewell@internet2.edu); **Others:** Christopher Bruton (CENIC, cbruton@cenic.org); Neil McKee (inMon, neil.mckee@inmon.com); Heidi Morgan (USC ISI, hlmorgan@isi.edu); Everson Scherrer Borges (IFES, everson@ifes.edu.br); Seungmin Kim (Yonsei, ehf@yonsei.ac.kr).

References

https://nrp.ai/llms • https://ellm.nrp-nautilus.io/v1 • https://nrp.ai/llmtoken • https://nrp.ai/llm-status • https://nrp.ai/documentation/userdocs/ai/llm-managed • https://nrp.ai/documentation/userdocs/ai/vector-database • https://sense.es.net • https://www.gna-g.net/autogole/ • https://nrp.ai/documentation/userdocs/networks/fabric