MindRouter: Open-Source LLM Inference Gateway for Institutional AI Sovereignty

Fair-Share Scheduling, Protocol Translation, and Observability for On-Premises AI in Higher Education

Lucas Sheneman*

University of Idaho, sheneman@uidaho.edu

As large language models become more integrated into research, teaching, and administration, universities must decide which AI workloads belong in the cloud and which require stronger institutional control. This is not an either-or choice: an effective AI strategy combines cloud and on-premises systems to preserve AI sovereignty where it matters most. In this context, AI sovereignty means retaining control over infrastructure, data, models, access policies, guardrails, monitoring, and governance so that sensitive or strategic workloads remain aligned with institutional requirements while enabling transparent energy accounting and lower costs at scale.

This paper presents **MindRouter**, an open-source LLM inference gateway designed to help universities operate self-hosted AI services on institutional GPU infrastructure as part of a balanced AI strategy. MindRouter is designed to support sensitive workloads involving FERPA-regulated student data, controlled unclassified information (CUI), export-controlled data, and institutionally managed intellectual property (IP). The system provides a unified API layer that translates among OpenAI, Ollama, and Anthropic client protocols, routes requests across heterogeneous GPU backends using a Weighted Deficit Round Robin (WDRR) fair-share scheduler, and incorporates role-based access control, audit logging, configurable guardrails, and operational observability through real-time GPU telemetry and system-level monitoring. Deployed in production at the University of Idaho on a cluster of 40 GPUs serving more than 64 open-weight models, MindRouter shows that mid-sized institutions can operate institutionally managed AI services while retaining direct control over data, infrastructure, and operational policy. MindRouter is released under the Apache 2.0 license and is available at **MindRouter.ai**

CCS CONCEPTS: Computer systems organization → Distributed architectures; • Computing methodologies → Artificial intelligence; • Information systems → Web services

Additional Keywords and Phrases: LLM inference, AI sovereignty, load balancing, fair-share scheduling, open source, GPU cluster, compliance, agentic computing, observability

1 INTRODUCTION

The rapid adoption of large language models (LLMs) has created substantial demand for AI inference services across higher education. Faculty are incorporating LLMs into research and teaching, students are using them for learning and coding assistance, and administrators are exploring their potential to improve operational efficiency. Many institutions currently meet this demand through commercial API subscriptions, such as OpenAI, Anthropic, and Google, which route prompts, research queries, and student interactions through third-party cloud infrastructure.

* Director of Research Computing and Data Services (RCDS) within the Institute for Interdisciplinary Data Sciences (IIDS)

This reliance raises important concerns related to data sovereignty, regulatory compliance, and institutional autonomy. Under FERPA, student educational records require strong access controls that may be more difficult to ensure when data is processed by commercial cloud services. Research data subject to export controls, IRB protections, or sponsor restrictions may also be prohibited from leaving institutional infrastructure. The EU AI Act [1] introduces additional compliance and governance considerations for institutions engaged in international collaboration. More broadly, as AI capabilities become strategically important to universities, dependence on a small number of commercial providers creates institutional risk through pricing changes, service interruptions, policy changes, and reduced flexibility.

Several technical barriers have limited the development of sovereign institutional AI infrastructure. Operating a heterogeneous GPU environment across multiple inference engines, including vLLM [2] and Ollama [3], requires managing incompatible API dialects and deployment patterns. Although the OpenAI API has emerged as a *de facto* client standard, self-hosted inference engines often expose different interfaces, and the Anthropic Messages API introduces an additional API dialect. Fairly scheduling shared GPU resources across large and diverse user populations with different institutional roles and priorities also remains a practical challenge. Building the observability, auditability, and security controls needed for compliance review, operational monitoring, and capacity planning adds significant complexity.

MindRouter addresses these barriers as an open-source LLM inference gateway designed for institutional deployment. It makes three primary contributions: ***(1) a bidirectional protocol translation layer*** built around a canonical intermediate representation, ***(2) a fair-share scheduler*** for shared heterogeneous GPU inference, and ***(3) an institutional deployment model*** that integrates authentication, audit logging, telemetry, and cost-aware operations into a unified campus AI service.

The remainder of this paper describes the system architecture, scheduling approach, security model, and production deployment experience of operating MindRouter at the University of Idaho.

2 RELATED WORK

Several systems address aspects of the LLM serving stack, but each targets a narrower layer than MindRouter. vLLM delivers high-throughput inference with PagedAttention but is designed for single-engine serving rather than multi-engine orchestration. Ollama simplifies local deployment, but lacks multi-user scheduling, enterprise authentication, and governance features, and is not designed to consistently sustain high-throughput institutional workloads. LiteLLM [4] provides API translation and unified access to providers but focuses on proxying rather than GPU-aware scheduling, access control, or audit logging. NVIDIA Triton Inference Server [5] is a mature serving platform but targets general ML inference rather than LLM-specific workflows such as chat, function calling, structured outputs, and streaming.

Frameworks such as LangChain [6] address yet another layer of the stack. These systems help developers build LLM-enabled applications, retrieval workflows, and agentic pipelines, but they assume the existence of an underlying inference service and do not solve the infrastructure-level problems of protocol interoperability, fair resource allocation, backend routing, or institutional observability. In practice, universities deploying campus-wide LLM services must support both application-facing integration and infrastructure-facing control.

Institutional LLM services therefore face requirements that extend beyond model serving alone. These include real-time request routing, fair sharing of limited GPU resources across many users, protocol interoperability across clients and backends, strong authentication and authorization, auditability for compliance review, and operational observability for capacity planning and system management. Existing systems do not combine protocol interoperability, heterogeneous backend routing, fair-share scheduling, and institutional governance in a form designed for campus-operated LLM infrastructure. MindRouter is designed to address this combination of requirements by providing an LLM-specific

inference gateway that integrates protocol translation, GPU-aware scheduling, and institutional governance capabilities in a form suitable for university-operated infrastructure.

3 SYSTEM ARCHITECTURE

MindRouter is implemented as a containerized Python application built on the FastAPI ASGI framework. The architecture comprises five principal components: the API gateway, the protocol translation layer, the fair-share scheduler, the backend registry with GPU sidecar agents, and the observability stack.

3.1 API Gateway

The gateway exposes three protocol-compatible API surfaces: OpenAI-compatible endpoints (*/v1/chat/completions*, */v1/embeddings*, */v1/models*), Ollama-compatible endpoints *(/api/chat*, */api/generate*, */api/tags*), and an Anthropic Messages API endpoint *(/anthropic/v1/messages*). Each endpoint authenticates requests using Argon2-hashed API keys, enforces per-user quotas, and delegates requests to the translation layer. The gateway provides a built-in web chat interface and locally hosted voice endpoints for text-to-speech (TTS) and speech-to-text (STT) using Kokoro and Whisper [7].

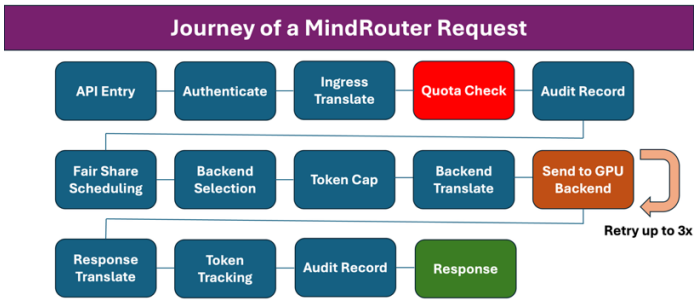

Figure 1: MindRouter processes each API request through authentication, translation, quota enforcement, scheduling, retries, and response formatting, enabling multi-engine support with high throughput, low latency, and strong security and observability.

3.2 Protocol Translation Layer

Rather than implementing $N \times M$ point-to-point translators between each client protocol and backend engine, MindRouter uses a canonical intermediate schema with $N + M$ translators: one inbound translator per client protocol and one outbound translator per backend engine. The canonical schema normalizes chat requests, messages, tool definitions, embedding requests, and streaming responses, including tool-call deltas.

This design allows any supported client protocol to access any supported model hosted on any supported backend engine. For example, an application using the Anthropic SDK can target models hosted on Ollama, while an OpenAI-configured LangChain application can route requests to vLLM. The translation layer preserves structured outputs and tool-calling semantics across protocols, enabling agentic workflows [8, 9] that span heterogeneous inference backends.

3.3 Fair-Share Scheduler

MindRouter implements a ***Weighted Deficit Round Robin (WDRR)*** scheduler [10] adapted for LLM inference workloads. Each user is associated with a group and corresponding scheduling weight, for example administrator=10, faculty=5, staff=2, and student=1. The scheduler maintains per-user deficit counters and computes a composite priority score as:

$$\text{priority} = \frac{\text{deficit}+\text{burst\_credits}}{\text{weight}} \times \text{deprioritization\_factor} + \text{wait\_bonus} \qquad \textbf{(1)}$$

Burst credits accumulate during idle periods in proportion to user weight, allowing a single user to consume otherwise idle cluster capacity. When contention arises, credits decay rapidly to restore fair-sharing. A configurable fairness window tracks recent usage and reduces effective priority for users that have consumed a disproportionate share of cluster capacity.

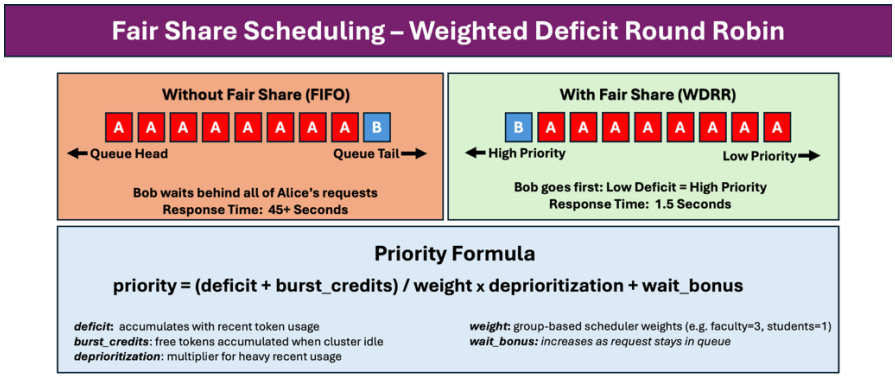

Figure 2: MindRouter's fair-share scheduler allows the GPU cluster to remain fully utilized by large, automated workflows while still being responsive (i.e., low latency) to interactive users and smaller workloads.

3.4 Backend Scoring

Backend selection uses a multi-factor scoring function to choose among eligible backends for each request. Hard constraints, including model availability, modality support, capacity, and memory fit, filter out ineligible backends. The remaining candidates are ranked using the soft-scoring factors shown in *Table 1*.

***Table 1**: Backend soft scoring factors.*

Factor	Score	Description
Model Loaded	+100	Model already resident in GPU memory
Low Utilization	+50	GPU utilization below 50%
Low Latency	+40	Low observed latency (exponential moving average)
Short Queue	+30	Few pending requests
High Throughput	+20	Recent tokens/second processed
Priority	+N x 10	Admin-configured backend preferences based on GPU capability

3.5 GPU Sidecar Agents and Node/Backend Model

MindRouter models cluster topology using a separation between physical nodes and logical backends. A node represents a physical server, while a backend represents an inference endpoint running on that node. Multiple backends may exist on a node, with each backend assigned to at least part of a physical GPU. Each server node runs a lightweight containerized ***sidecar agent*** implemented as a minimal FastAPI service using *pynvml*, exposing per-GPU metrics such as utilization, memory usage, temperature, power draw, clock speeds, and running processes. The MindRouter gateway collects telemetry

in two phases: it first polls each node-sidecar once to cache device-level GPU state, then polls each backend's inference API for health and model information and derives backend-level GPU utilization from cached node data filtered by GPU indices. This design reduces redundant network calls while preserving accurate backend-level visibility.

3.6 Security Architecture

The security model was validated through institutional CISO review covering authentication, secrets management, encryption, network architecture, container security, database security, and input validation. Key features include: Argon2-hashed API keys and passwords; Azure AD SSO with JIT user provisioning; RBAC across all surfaces; non-root Docker containers behind nginx TLS termination; sidecar authentication via shared secrets with constant-time comparison; localhost-bound internal services; and full request/response audit logging for compliance. All data remains on campus.

4 DEPLOYMENT EXPERIENCE

MindRouter has been in production at the University of Idaho since early 2026 (Figure 3). The deployment spans 13 GPU nodes with 40 GPUs, including NVIDIA H200, H100, RTX A8000, and A4000 cards, and serves 64 open-weight models across Ollama and vLLM backends. User provisioning is integrated with Azure AD SSO through Microsoft Entra ID, with users assigned to institutional groups that determine scheduler weights and quota allocations. API keys are issued through a self-service dashboard, enabling programmatic access from tools such as the OpenAI Python SDK, LangChain, LlamaIndex, IDE extensions, and agentic harnesses such as Claude Code or ForgeCode.

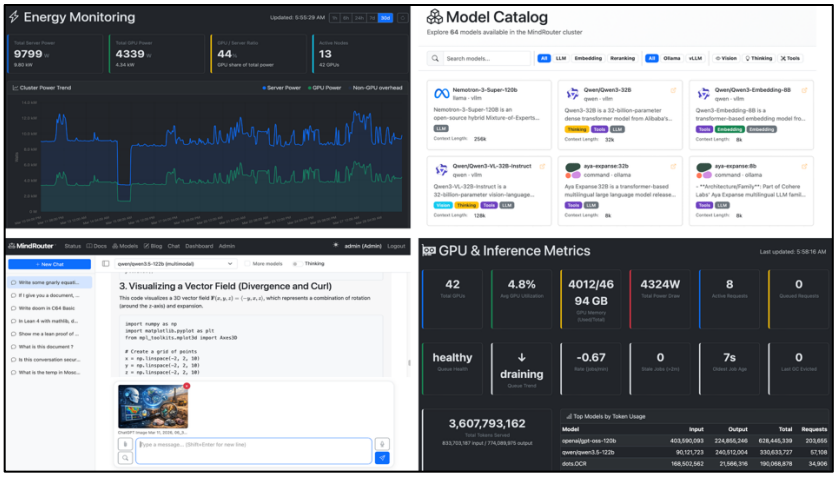

Figure 3: Screenshots of the MindRouter web interface showing energy consumption telemetry, the hosted model catalog, a full-featured interactive multimodal chat interface with voice integration, and GPU monitoring metrics.

Several operational lessons emerged. Node-backend separation enabled efficient use of heterogeneous GPUs and proper placement of large multi-GPU models. Automatic context-length capping for vLLM and Ollama prevented GPU memory oversubscription. The WDRR scheduler maintained full utilization during idle periods and fair-sharing under contention. A built-in web chat interface accelerated adoption with several non-technical user groups.

Telemetry shows steady growth: since early 2026, MindRouter processed **3.7B tokens** across **64 models**, with **OpenAI gpt-oss-120b and Qwen3.5 400B/122B** among the most used. Integrated power telemetry showed normal cluster operation

at about **9.4 kW**, with an observed **peak of 14.1 kW** and a **theoretical maximum of 22.1 kW**; including facility overhead, this corresponds to about **275 kWh/day** under normal operation. Over a **17-day** observed period in March 2026, the system processed **872,997,430** tokens using campus electricity, compared with an estimated **$4,027 to $11,896** for comparable commercial API usage, implying **annual savings of $82,843 to $251,806** depending on provider and model assumptions.

These results show that MindRouter made both power and cost first-class operational metrics, giving the University of Idaho a practical basis for energy accounting, capacity planning, and evaluation of on-prem AI relative to external services.

5 CONCLUSION

MindRouter demonstrates that greater institutional control over AI infrastructure and services is achievable for mid-sized universities. By providing a unified, open-source inference gateway with protocol translation, fair-share scheduling, and institutional security controls, it enables universities to deliver campus AI services while keeping data and operations under institutional management. Its production deployment shows that a small IT team can operate a heterogeneous, university-managed AI inference service that supports real institutional workloads while improving visibility into utilization, energy use, and operating cost. As AI becomes more central to research, teaching, and administration, MindRouter offers a practical open-source blueprint for universities seeking greater control over their AI infrastructure and services.

MindRouter is released under the Apache License 2.0 at **MindRouter.ai**, with documentation covering architecture, APIs, scheduling, and security as well as a link to the public GitHub repository.

ACKNOWLEDGMENTS

This work was supported by the National Science Foundation under NSF Award #2427549

REFERENCES

[1] European Parliament and Council of the European Union. 2024. Regulation (EU) 2024/1689 of the European Parliament and of the Council laying down harmonised rules on artificial intelligence (Artificial Intelligence Act). Official Journal of the European Union L, 2024/1689.

[2] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient Memory Management for Large Language Model Serving with PagedAttention. In Proceedings of the 29th ACM Symposium on Operating Systems Principles (SOSP '23), October 23–26, 2023, Koblenz, Germany. ACM, New York, NY, USA, 611-626. https://doi.org/10.1145/3600006.3613165

[3] Ollama. 2024. Ollama: Get up and running with large language models locally. Retrieved March 15, 2026 from https://ollama.com

[4] BerriAI. 2024. LiteLLM: Call 100+ LLM APIs in OpenAI format. Retrieved March 15, 2026 from https://github.com/BerriAI/litellm

[5] NVIDIA Corporation. 2024. *Triton Inference Server*. Retrieved March 15, 2026 from https://github.com/triton-inference-server/server

[6] Harrison Chase. 2022. LangChain: Building applications with LLMs through composability. Retrieved March 15 2026 from https://github.com/langchain-ai/langchain

[7] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust Speech Recognition via Large-Scale Weak Supervision. In Proceedings of 40th Intl. Conference on Machine Learning (ICML '23), July 23–29, 2023, Honolulu, Hawaii. PMLR, 28492–28518.

[8] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. ReAct: Synergizing Reasoning and Acting in Language Models. In Proceedings of the 11th International Conference on Learning Representations (ICLR '23), May 1–5, 2023, Kigali, Rwanda.

[9] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2024. Toolformer: Language Models Can Teach Themselves to Use Tools. In Advances in Neural Information Processing Systems 36 (NeurIPS 2023), December 10–16, 2023, New Orleans, LA. Curran Associates, Inc.

[10] Abhijit Shreedhar and George Varma. 1995. Efficient fair queuing using deficit round-robin. IEEE/ACM Transactions on Networking 3, 3 (June 1995), 375–385. https://doi.org/10.1109/90.392383