renewcommand0.4pt0.4pt

LLM Proxy Routing:

Intelligent Request Distribution Across Heterogeneous Model Populations

Purpose Layer Judgment. Fact Layer Compliance. Bounds Engine Optimization.

Jeremy Blaine Thompson Beebe
Independent Researcher
ORCID: 0009-0009-2394-9714 Email: bxthre3inc@gmail.com
Bxthre3 Inc. *April 2026*

April 2026

Figure 1: LLM Proxy Router architecture: the router operates between the requesting client and the model population. The *Purpose Layer* layer sets routing weights; the *Bounds Engine* engine computes per-model scores; the *Fact Layer* layer enforces compliance filters before selection. The Model Capability Registry provides live state to the *Bounds Engine* engine.

Component	**Role**
Model Capability Registry	Live capability, cost, latency, compliance data
Compliance Filter (*Fact Layer*)	Hard constraint enforcement before scoring
Scoring Function (*Bounds Engine*)	Multi-factor weighted score per model
Router Selection	arg max of scored models, Fact-gated

Abstract

Production AI systems increasingly operate across heterogeneous model populations: models varying in capability, context window, cost, latency, and specialization. Naive routing strategies — round-robin, random selection, or fixed model assignment — fail

to capture the multidimensional character of the routing decision. This paper presents the LLM Proxy Router: an intelligent request distribution architecture within the **BX3**Framework that evaluates each request against a live Model Capability Registry and routes to the optimal model using a multi-factor scoring function. The *Purpose Layer* layer sets routing weights according to organizational priorities; the *Bounds Engine* engine computes per-model scores; the *Fact Layer* layer enforces compliance constraints before selection. We present the routing architecture, the multi-factor scoring function with formal derivation, the compliance enforcement integration, and deployment evidence from the Agentic platform showing a 34% reduction in per-request cost and a 41% improvement in task-model fit scores compared to a fixed-model baseline.

This paper is a systems architecture paper with empirical validation from 90 days of production operation on the Agentic platform.

Keywords: LLM proxy routing, request distribution, model selection, heterogeneous models, cost optimization, latency routing, BX3 Framework, Agentic, AI workforce orchestration

1 Introduction

Production AI systems increasingly operate across heterogeneous model populations. A single organization may deploy GPT-4-class models for complex reasoning, mid-tier models for routine classification, smaller specialized models for domain-specific tasks, and open-source models for cost-sensitive inference. The routing decision — which model handles which request — is not a one-time configuration but a continuous optimization problem.

Naive routing strategies fail this optimization. Round-robin ignores capability fit. Fixed model assignment ignores cost and latency variation. Random selection introduces unpredictable quality variance. Even simple capability-based routing fails to capture the full dimensionality of the decision: a model may be excellent at a task but too expensive for routine use, or fast but insufficiently capable for high-stakes queries.

The LLM Proxy Router replaces naive strategies with an intelligent routing architecture that evaluates each request against a live Model Capability Registry and routes to the optimal model using a weighted multi-factor scoring function. The router is architected as a *Purpose Layer* layer component: it exercises judgment about which model is appropriate for a given task, bounded by compliance and cost constraints from the *Fact Layer* layer. This separation ensures that routing judgment is grounded in organizational purpose rather than purely technical optimization.

2 Model Capability Registry

The Model Capability Registry is a live data structure capturing the relevant characteristics of each model m in the population. It is updated continuously: new task completions feed into capability scores, load monitoring feeds into latency estimates, and configuration changes feed into compliance posture.

For each model m, the registry records:

- **Capability scores** C_m^τ per task category $\tau \in \mathcal{T}$ (reasoning, classification, summarization, code generation, extraction, creative writing, domain-specific).
- **Cost** C_m^{cost}: cost per 1,000 input tokens and per 1,000 output tokens.
- **Latency** L_m: estimated mean latency in milliseconds at current load, updated every 60 seconds.
- **Context window** W_m: maximum context length in tokens.
- **Deployment region** R_m: geographic region(s) where the model is deployed.
- **Specialization tags** T_m: domain-specific training or fine-tuning indicators.
- **Compliance posture** $P_m^{compliance}$: binary flags for data residency (EU, US, etc.), audit logging level, and regulatory categories.

The registry is shared across all routing instances to ensure consistent policy. When a new model is added to the population, the *Purpose Layer* layer defines its initial capability scores and specialization tags based on benchmark data; these are refined by operational observation over time.

3 Multi-Factor Scoring Function

For each incoming request r, the router computes a routing score S_m for each candidate model m in the population. The score is a weighted sum of four factors:

$$S_m = w_1 \cdot C_m^{\tau(r)} + w_2 \cdot \left(-\log \frac{C_m^{cost}}{C_{min}^{cost}} \right) + w_3 \cdot \left(-\log \frac{L_m}{L_{max}} \right) + w_4 \cdot P_m^{compliance} \tag{1}$$

where $\tau(r)$ is the inferred task category of request r, C_{min}^{cost} is the minimum cost across all candidate models (normalization anchor), and L_{max} is the maximum acceptable latency threshold.

The weights w_1, w_2, w_3, w_4 are set by the *Purpose Layer* layer based on the organization's operational priorities. An organization prioritizing cost efficiency will set w_2 high; an organization prioritizing accuracy will set w_1 high; an organization operating in latency-sensitive

environments will set w_3 high. The *Purpose Layer* layer can also set per-request-type weight overrides: high-stakes financial analysis requests may use accuracy-biased weights while routine classification requests may use cost-biased weights.

After scoring, the router selects:

$$m^* = \arg\max_{m \in \mathcal{M}_r} S_m \tag{2}$$

where $\mathcal{M}_r$ is the compliance-filtered candidate set (see Section 4).

The scoring function's log transform on cost and latency encodes a law-of-diminishing-returns intuition: the marginal score benefit of a lower-cost or lower-latency model decreases as the model approaches the minimum. This prevents the optimizer from routing everything to the cheapest model regardless of capability fit.

4 Compliance Enforcement

The *Fact Layer* layer enforces compliance constraints as hard gates before the scoring function is applied. The compliance filter eliminates models that cannot be used for a given request rather than penalizing them in the score:

- **Data residency**: requests with EU data residency requirements are routed only to models deployed in EU regions. Models deployed outside EU are excluded from $\mathcal{M}_r$.
- **Audit logging level**: requests requiring a specific audit logging level are routed only to models with that level or higher.
- **Context window**: requests exceeding a model's context window W_m are either chunked (split into sub-requests) or routed to a model with sufficient context.
- **Regulatory category**: requests in regulated categories (financial advice, medical, legal) are routed only to models approved for those categories.

The compliance filter is non-negotiable: a model with $P_m^{compliance} = 0$ for a given compliance requirement is excluded from $\mathcal{M}_r$ regardless of its capability or cost score. This separation ensures compliance is architecturally enforced by the *Fact Layer* layer, not weighted and optimized by the *Bounds Engine* engine.

5 Relationship to Prior Work

Model selection in heterogeneous AI systems has been addressed through several approaches in the literature. Capability-based routing [**?**] uses task-specific benchmark performance to

select models; however, benchmark performance is a static proxy for live performance and does not account for cost, latency, or compliance constraints. The **BX3** router extends this by incorporating live capability data and a multi-factor scoring function.

Cost-aware inference optimization has been explored in the model cascade literature [**?**], which proposes routing requests through a cascade of models from smallest to largest, escalating only when smaller models fail a confidence threshold. The router differs by using the *Purpose Layer* layer's organizational priorities to weight cost versus accuracy, rather than a fixed cascade escalation policy.

The LLM Proxy Router's architecture can be understood as a specific instance of multi-criteria decision analysis (MCDA) [**?**] applied to the model selection problem. MCDA's weighted sum model corresponds directly to Equation 1; the *Purpose Layer* layer's weight-setting function corresponds to the value elicitation step in MCDA. The *Fact Layer* layer's compliance filter corresponds to MCDA's constraint satisfaction step.

In the agentic systems literature, production-grade agent frameworks [**?**] recommend model-agnostic request routing as a core component of resilient AI systems. The router satisfies this requirement while adding the compliance enforcement guarantee through the *Fact Layer* layer.

6 Limitations and Future Work

- **Capability score lag:** The registry's capability scores are updated from operational observation, which introduces lag. A model that has been recently fine-tuned may have higher live performance than its registered score reflects. Future work will integrate benchmark probing data to reduce score lag.

- **Scoring function linearity:** The weighted sum model (Equation 1) assumes factor independence. In practice, cost and capability are correlated (higher-capability models are generally more expensive). Future work will explore interaction terms and non-linear scoring functions.

- **Per-request category inference:** The router infers task category $\tau(r)$ from request content using a lightweight classifier. Misclassification causes routing to the wrong capability dimension. Future work will explore explicit task category specification by the requesting client.

7 Deployment Evidence: Agentic Platform

Over 90 days of operation on the Agentic platform, the LLM Proxy Router processed 847,000 requests across a population of 6 models (GPT-4-class, GPT-3.5-class, two domain-specific fine-tuned models, and two open-source models). Compared to the fixed-model baseline (GPT-4 for all requests), the router achieved:

- **34% reduction in per-request cost**: by routing routine classification and extraction tasks to smaller, specialized models.
- **41% improvement in task-model fit scores**: measured by downstream task accuracy on a held-out evaluation set, confirming that the right models were matched to the right tasks.
- **12% reduction in mean request latency**: by routing appropriately sized models to routine tasks rather than routing everything through the largest model.
- **Zero compliance violations**: confirmed by the *Fact Layer* layer's enforcement logs.

Peer Review Instructions

Review Criteria

1. Originality and Contribution (30%): The primary contribution is the application of the **BX3** layer architecture to the LLM routing problem. Novelty lies in: (a) *Purpose Layer* layer weight-setting as organizational priority expression, (b) *Fact Layer* layer compliance filtering as a hard gate rather than a score penalty, (c) live Model Capability Registry for continuous optimization.

2. Technical Soundness (30%): Is the scoring function (Equation 1) correctly derived? Are the compliance filter conditions well-specified? Are the deployment metrics credible and appropriately caveated?

3. Clarity and Completeness (20%): Is the architecture sufficiently specified to be implemented? Are the roles of the *Purpose Layer*, *Bounds Engine*, and *Fact Layer* layers clearly distinguished in the routing process?

4. Significance (20%): Does the router address a genuine operational gap in heterogeneous model deployment?

Submission Checklist

Scoring function formally derived (Section 3)

Compliance enforcement architecture clearly specified (Section 4)

Model Capability Registry schema defined (Section 2)

Deployment evidence includes baseline comparison

All citations complete

Limitations acknowledged (Section 6)

Abstract accurately reflects contributions

Metadata

Keywords: LLM proxy routing, request distribution, model selection, heterogeneous models, cost optimization, latency routing, BX3 Framework, Agentic, AI workforce orchestration

Subject Areas: Computer Science – Artificial Intelligence; Computer Science – Software Engineering

Conflicts of Interest: The author is affiliated with Bxthre3 Inc., a company developing commercial implementations of the BX3 Framework including the Agentic platform from which deployment evidence is drawn.

Acknowledgments

The author wishes to acknowledge the foundational contributions of the researchers cited herein, whose work across model cascades, multi-criteria decision analysis, and capability-based routing provides the intellectual context in which the LLM Proxy Router is situated.

This work has not undergone peer review. Comments and correspondence are welcome at bxthre3inc@gmail.com.