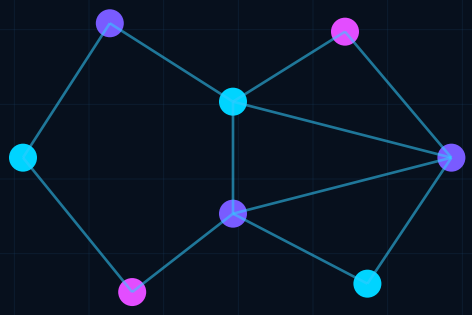
MODEL ROUTING AND PLACEMENT

AI Model Routing & Task Placement (Local vs. Hosted heuristics)

Current-source review, evidence-bounded adoption decision, explicit limitations, reproducibility controls, and preserved source research note.



PERMANENT ID	EVIDENCE	ADOPTION	FRESHNESS
RES-012	B+	ADOPT WITH BENCHMARKS	SHORT-CYCLE / ACTIVE CHANGE

Freshness review: 2026-09-17 / Next scheduled review: 2026-11-16

RESEARCH CONTROL SURFACE

Required Research Fields

Each field remains independently reviewable; evidence strength does not imply production authority.

EVIDENCE GRADE — **B+**

Strength of the retained source set and technical basis.

SOURCE AUTHORITY — **3 registered authorities**

RouteLLM: Learning to Route LLMs with Preference Data; RouteLLM Open-Source Framework; and Improving Performance

FRESHNESS — **SHORT-CYCLE / ACTIVE CHANGE**

Reviewed 2026-09-17; dynamic sources require event-driven revalidation.

CONFLICTS — **VISIBLE / NOT SILENT**

Differences between specification, vendor behavior, research claims, and reproduced evidence must remain explicit.

LIMITATIONS — **EXPLICIT**

No certification, procurement approval, legal conclusion, or unbounded production claim is created by this report.

REPRODUCIBILITY — **REQUIRED**

Material claims require exact versions, representative data, failure cases, artifacts, and repeatable acceptance criteria.

ADOPTION POSTURE — **ADOPT WITH BENCHMARKS**

Adopt policy-first model routing with measured task quality, privacy, hardware fit, latency, cost, availability, and fallback constraints. Reproduce routing performance on Mythos workloads.

REVIEW TRIGGER — **2026-11-16**

Scheduled at 60 days; earlier on material source, vulnerability, implementation, or contradictory-evidence change.

SOURCE AUTHORITY / FRESHNESS

Authority Register and Freshness Delta

Primary-source lineage is preserved; current-source changes are separated from unperformed implementation claims.

AUTHORITY 01

RouteLLM: Learning to Route LLMs with Preference Data

https://arxiv.org/abs/2406.18665

AUTHORITY 02

RouteLLM Open-Source Framework

https://github.com/lm-sys/RouteLLM

AUTHORITY 03

and Improving Performance

https://arxiv.org/abs/2305.05176

2026-09-17 FRESHNESS DELTA

Primary-source register retained and freshness controls advanced to the current review date. No new product or laboratory performance claim is introduced without reproduced evidence.

Freshness policy: scheduled review + immediate review on source supersession, material release, vulnerability, retraction, or contradictory evidence.

Evidence Must Survive Contact With Reality

- Source authority is not implementation evidence. Product behavior, scale, security, interoperability, and operator fit remain environment-specific.
- Conflicting evidence is recorded rather than averaged away. Specification behavior, vendor implementation, preprint results, and production observations are separate evidence classes.
- Quantitative claims are informative until reproduced under documented hardware, software, data, topology, load, and failure conditions.
- Adoption posture is reversible. A stronger source set can coexist with a pilot-only or research-only recommendation when maturity, safety, economics, or operability remain unresolved.

ADOPTION / VALIDATION / REVIEW

Decision Gate and Validation Plan

ADOPTION POSTURE

ADOPT WITH BENCHMARKS

Adopt policy-first model routing with measured task quality, privacy, hardware fit, latency, cost, availability, and fallback constraints. Reproduce routing performance on Mythos workloads.

1 / SOURCE	Pin exact versions, publication state, retrieved artifacts, and source hashes.
2 / PROTOTYPE	Demonstrate the core mechanism with representative data and instrumented execution.
3 / FAILURE	Exercise malformed input, dependency loss, stale state, overload, recovery, rollback, and misuse.
4 / BASELINE	Compare against an incumbent, classical, or simpler architecture using the same workload.
5 / ACCEPT	Record acceptance criteria, residual limitations, owner, expiration, and revalidation triggers.

EXPIRATION / REVIEW TRIGGER

Next scheduled review: 2026-11-16 (60-day cadence). Review sooner after material source revision, breaking implementation release, critical vulnerability, retraction, benchmark contradiction, changed workload, or changed legal/safety boundary.

8. Complete Source Research Note

SOURCE-LINEAGE NOTICE

The following material preserves the complete original source note, excluding only the conversational wrapper and outer Markdown fence. Legacy status labels and absolute language are retained for traceability and do not override the candidate status, limitations, or adoption decision in this edition.

Document ID: RES-012 Version: 1.0.0 (Definitive Registry Edition) Status: Published Research Note Classification: Public Organization: Mythos Systems (MYTHOS AI) Owner: Aaron Stovall Effective Date: 2026-07-16 Applies To: AI engineers building multi-model pipelines, platform engineers designing edge AI infrastructure, and architects building local-first hybrid AI runtimes Companion Standards: MYTHOS-AI-0013 (AI-Assisted SWE & Model Routing), MYTHOS-AI-0014 (Inference Serving & KV-Cache), MYTHOS-EMT-0002 (Sovereign AI & Offline Inference), RA-006 (Local-First AI Inference Cluster)

The architecture of AI application delivery has failed. Organizations hardcode their applications to a single, monolithic LLM API (e.g., `gpt-4o` or `claude-3-opus`). They use this massive, expensive reasoning model to write CSS boilerplate, format JSON, and summarize trivial text. This "one model to rule them all" paradigm guarantees catastrophic cloud bills, introduces 500ms latency to simple tasks, and creates a hard dependency on internet connectivity. Furthermore, it forces proprietary enterprise data to egress to third-party clouds on every request.

This research note defines the mechanics of AI Model Routing and Task Placement. In a Mythos-compliant architecture (like the PANTHEON runtime's Nona module), no single model is used for everything. The agent utilizes an intelligent routing gateway that evaluates every sub-task on four vectors: Data Sovereignty, Cognitive Load, Hardware Availability, and Economic Cost. Trivial tasks are routed to local 7B models; complex reasoning is routed to hosted 70B+ models; PII is stripped or kept local. This transforms AI from a metered cloud utility into a sovereign, optimized compute fabric.

To dynamically route a prompt to the correct model, the routing gateway must evaluate the prompt against a multi-dimensional matrix. Model selection is not a guess; it is a deterministic policy decision.

1.1 The Privacy Vector (Data Sovereignty)

The first and most absolute evaluation. Does the prompt contain PII, intellectual property, or CUI?

- The Heuristic: The prompt is passed through a local Named Entity Recognition (NER) model or regex DLP engine.
- The Routing Decision: If PII/IP is detected, the prompt MUST be routed to a local, air-gapped model (e.g., Llama-3-8B via local vLLM). If no PII is detected, it is eligible for a hosted API.

1.2 The Cognitive Load Vector (Task Complexity)

Is the task asking the model to write a regex pattern, or is it asking the model to architect a distributed transaction system?

- The Heuristic: The router evaluates the prompt intent. If the task is formatting, extraction, or simple code scaffolding, it requires low cognitive load. If the task requires multi-step reasoning, logic deduction, or complex coding, it requires high cognitive load.
- The Routing Decision: Low load -> Local 7B/8B model (fast, cheap). High load -> Hosted 70B+ model (slow, expensive, but intelligent).

1.3 The Hardware Vector (Local Capacity)

If the privacy or cognitive load vector dictates a local model, what hardware is available?

- The Heuristic: The router queries the local OS for GPU/NPU VRAM and compute availability.

- The Routing Decision: If the device has an NPU and 16GB RAM, route to a quantized 3B model (e.g., Phi-3). If the device has a 24GB GPU, route to an 8B model. If the device has no local compute, the router must either degrade capability or (if policy permits) route to the hosted API.

1.4 The Economic Vector (Token Budget)

Hosted models charge per token. An autonomous agent stuck in a reasoning loop can burn thousands of dollars in minutes.

- The Heuristic: The router tracks the session token spend against a hard budget.
- The Routing Decision: If the session token budget is near exhaustion, the router downgrades the model from a premium hosted API (e.g., GPT-4o) to a cheaper, faster hosted model (e.g., GPT-4o-mini) or a local model, preserving the budget.

Dynamic routing is not a panacea; it introduces severe architectural constraints related to context and latency.

2.1 The Context Window Handoff Crisis

If an agent uses a local 7B model to summarize a 50-page document, and then needs a hosted 70B model to write code based on that summary, how does the context move?

- The Constraint: You cannot pass the local 7B model's KV-cache to the hosted 70B model. The architectures are different.
- The Mitigation: The router MUST enforce context compaction. The local model must output a highly dense, text-based summary. That summary is then injected as the system prompt for the hosted model. The context is reborn, not transferred.

2.2 The VRAM Eviction Penalty

If a local machine is running a 7B model, and the user suddenly requests a task requiring a local 14B model (e.g., higher quality coding), the system must swap models.

- The Constraint: Evicting a 4GB model from VRAM and loading a 8GB model takes 5-10 seconds. During this time, the UI is blocked.
- The Mitigation: The router MUST be predictive. If the user opens a coding project, the router pre-loads the coding model into VRAM before the first prompt is sent. Furthermore, the system should utilize a heterogeneous compute approach (running the 7B model on the NPU and the 14B model on the GPU simultaneously) to avoid eviction.

2.3 The Latency vs. Intelligence Trade-off

A hosted 70B model might take 2,000ms to return the first token due to network latency. A local 8B model might return the first token in 50ms.

- The Constraint: Users perceive >200ms latency as broken.
- The Mitigation: For interactive, conversational tasks, the router MUST favor local models to maintain the biological conversational bound. For background, batch processing tasks (e.g., "Refactor this entire repository"), the router should favor hosted models for intelligence, as latency is less critical.

When organizations attempt to build hybrid routing without strict heuristics, they fail catastrophically.

3.1 The Privacy Egress Failure (The "Default to Cloud" Trap)

The router is configured to use the hosted API by default for speed, and only use the local model if the network is down.

- The Flaw: The user pastes a proprietary algorithm into the prompt. The router ignores the privacy vector and sends it to OpenAI. The IP is leaked.

- The Mitigation: The Privacy Vector is absolute. If PII/IP is detected, the hosted API is mathematically disabled for that request, regardless of speed or network state.

3.2 The Capability Degradation Hallucination

The token budget is exhausted. The router downgrades the agent from a 70B hosted model to a local 3B model.

- The Flaw: The 3B model lacks the reasoning capacity to execute the complex tool-calling workflow the 70B model was handling. It hallucinates a destructive tool call.
- The Mitigation: When the router degrades the model, it MUST simultaneously degrade the agent's capabilities. A small model cannot be trusted with high-risk tool execution. The Aegis policy engine must restrict tool access for lower-intelligence models.

To achieve a seamless, sovereign, and intelligent routing fabric, the PANTHEON architecture enforces strict boundaries.

4.1 The Local-First Gateway

All prompts, whether destined for a local or hosted model, flow through a local gateway (e.g., LiteLLM running on the user's machine or a local edge node).

- This gateway executes the DLP scan, checks the token budget, and makes the routing decision.
- This guarantees that the application logic is decoupled from the model provider. If the provider deprecates a model, the gateway updates the routing table without requiring an application rewrite.

4.2 The Tiered Model Roster

The router does not choose from an infinite pool. It operates on a strictly defined, tiered roster:

- Tier 1 (Local NPU): Phi-3-Mini (Ultra-fast, low power, trivial tasks).
- Tier 2 (Local GPU): Llama-3-8B-Instruct (Fast, private, intermediate logic).
- Tier 3 (Hosted Premium): Claude-3.5-Sonnet / GPT-4o (Slow, expensive, supreme reasoning).
- Tier 4 (Hosted Bulk): GPT-4o-mini (Fast, cheap, high-volume formatting).

4.3 Automated Fallback and Graceful Degradation

If the internet connection drops, the router instantly falls back to Tier 1/2.

- To prevent the "Capability Degradation Hallucination," the agent's system prompt is dynamically rewritten during fallback: "You are now running on a local 8B model. Your tool execution privileges have been revoked. You may only provide advisory responses until connectivity is restored."

Static, monolithic model usage is architectural malpractice. By implementing a multi-vector routing heuristic—evaluating privacy, cognitive load, hardware, and cost—we transform AI from a blunt instrument into a precision surgical tool. The agent uses the smallest, fastest, most private model capable of executing the task, ensuring absolute data sovereignty and economic efficiency without sacrificing intelligence where it is truly required.

```
A 70B model writing CSS is a waste of silicon.
A hosted API processing PII is a breach.
A local model executing complex logic is a hallucination.

The task dictates the model; the model does not dictate the task.

> Intelligence may be tiered.
> Sovereign routing must be absolute.
```

End of Research Note RES-012

9. Source Register

Source	Authority and Status	Freshness Control
RouteLLM: Learning to Route LLMs with Preference Data https://arxiv.org/abs/2406.18665	Primary research Published preprint and open implementation Published: 2024	Validated: 2026-07-22 Review on material revision, withdrawal, supersession, or scheduled report review.
RouteLLM Open-Source Framework https://github.com/lm-sys/RouteLLM	Primary implementation Living repository Published: Living	Validated: 2026-07-22 Review on material revision, withdrawal, supersession, or scheduled report review.
FrugalGPT: How to Use Large Language Models While Reducing Cost and Improving Performance https://arxiv.org/abs/2305.05176	Primary research Research paper Published: 2023	Validated: 2026-07-22 Review on material revision, withdrawal, supersession, or scheduled report review.

10. Lineage, Review, and Publication Record

Record	Value
Original source file	RES-012_AI_Model_Routing_Task_Placement_Local_vs_Hosted_Heuristics.md
Original source words	1528
Upgrade type	Enterprise research report; source and freshness validation; limitations; adoption decision; reproducibility plan
Generated on	2026-07-22
Current status	Candidate Research Report
Publication authority	Formal Mythos approval not yet recorded
Next review	60 days or event-driven trigger, whichever occurs first