METRUM AI

How it works | On-prem & hybrid | Savings | Capabilities | Documentation

APACHE-2.0 OPEN-SOURCE GENAI GATEWAY

One AI endpoint. The best model for every job.

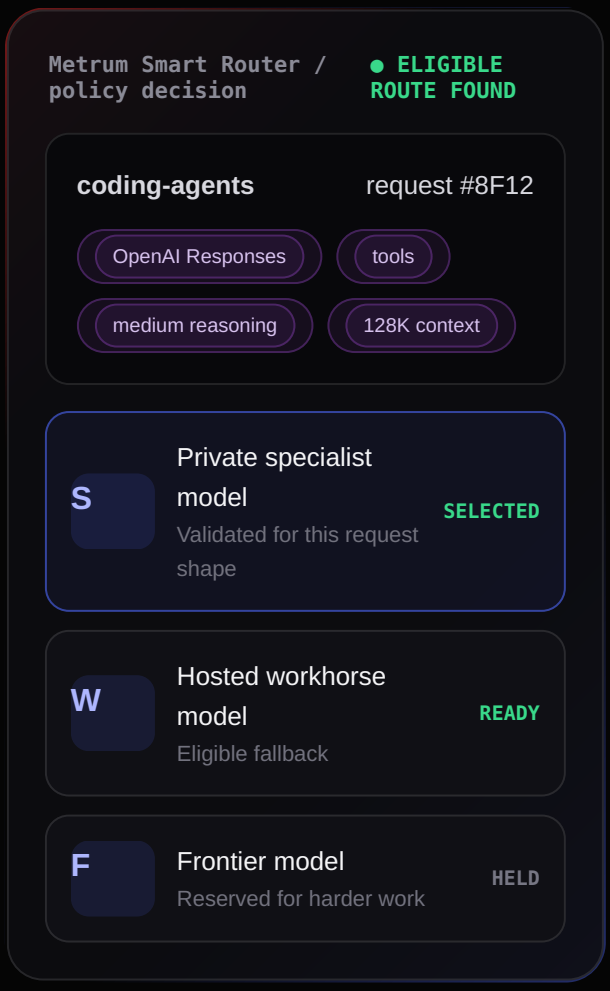
Route applications, coding agents, subagents, and always-on automation across hosted and private models—based on capability, quality, cost, latency, and your policy. Developers keep one stable API while your platform team continuously improves the model mix behind it.

Explore the documentation

View on GitHub

Get enterprise support →

OPENAI CHAT · OPENAI RESPONSES · ANTHROPIC MESSAGES · CODEX CLI · CLAUDE CODE · PRIVATE VLLM/SGLANG



Apache-2.0

Open source, self-hostable, and inspectable.

One governed API

OpenAI- and Anthropic-compatible client surfaces.

Hosted + private

Mix external APIs with enterprise inference endpoints.

Evidence built in

Request-time cost, latency, attempts, and fallback.

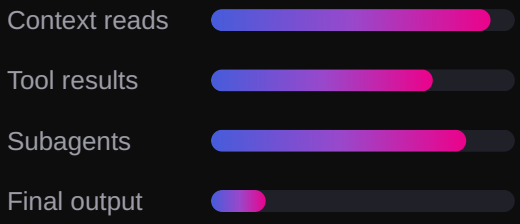
AGENT ECONOMICS CHANGED THE EQUATION

Frontier intelligence is valuable. Frontier-everywhere is wasteful.

Agents are not single prompts. Coding agents inspect repositories, call tools, replay context, retry steps, and launch subagents. Application, CI/CD, and monitoring agents run around the clock

Primary agent + subagent fan-out

Turn model choice into platform policy instead of asking every developer, application, and subagent to select providers manually.



Conceptual illustration, not measured customer data.

doing the same. Most of those steps do not require the most expensive model.

HOW IT WORKS

One stable client contract. A model portfolio behind it.

The user requests a deployment-defined model group. The platform team controls the eligible providers, models, endpoints, and policy behind that contract.

01 / RECEIVE

Accept familiar APIs

Applications and agents use OpenAI Chat, OpenAI Responses, or Anthropic Messages.

02 / FILTER

Check requirements

Dialect, tools, images, reasoning, structured output, payload size, token caps, and access.

03 / SELECT

Apply your policy

Choose among validated hosted or private targets with weighted, failover, scored, TypeScript, or external policy.

04 / EXPLAIN

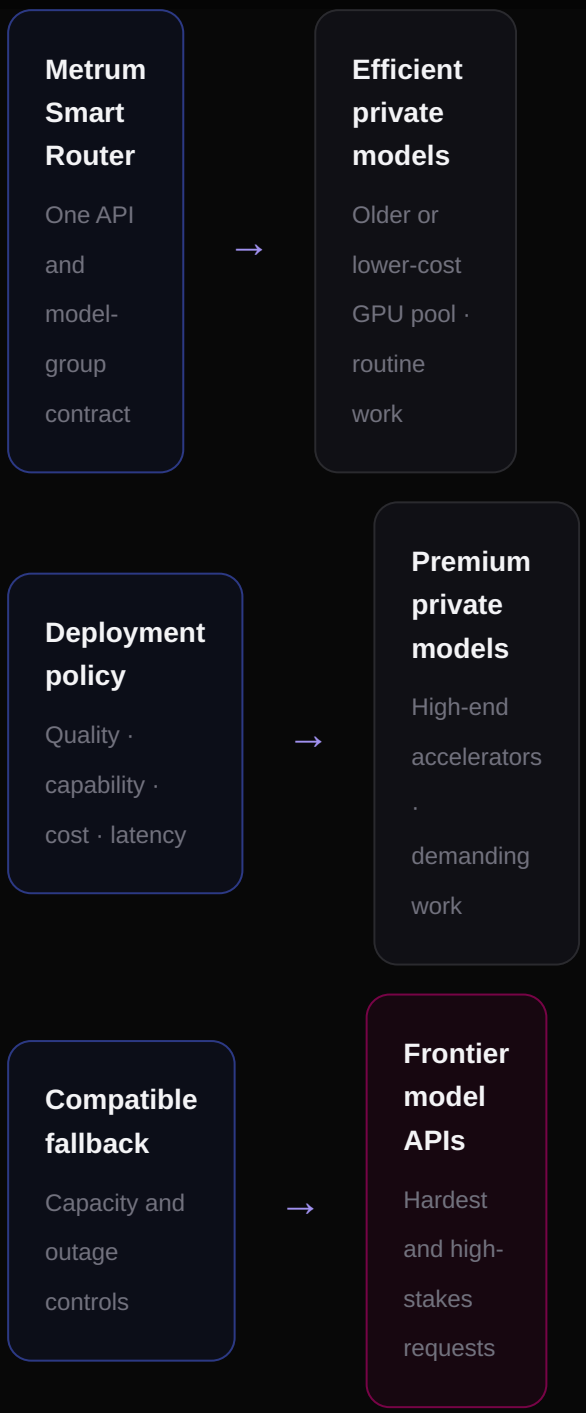
Record the evidence

Preserve request-time cost, latency, throughput, attempts, errors, cache, and fallback behavior.

ENTERPRISE HARDWARE EFFICIENCY

Put more of your AI infrastructure to useful work.

Connect smaller models on older or lower-cost enterprise hardware, larger models on premium accelerators, and external frontier APIs behind the same endpoint.



Use the hardware you already own

Register validated vLLM, SGLang, or other compatible private inference services as routing targets.

Reserve premium capacity

Keep top accelerators and frontier APIs available for requests that need complex reasoning, multimodality, or specialized capability.

No manual model switching

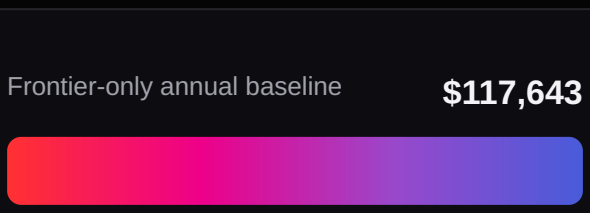
Developers and agents keep the same model-group name while the enterprise changes the infrastructure mix behind it.

Smart Router selects among configured inference endpoints. The enterprise serving platform remains responsible for GPU scheduling, model placement, and replicas.

TRANSPARENT SAVINGS SCENARIO

Model the savings. Then prove them with your workload.

Agents do not stop when the workday does. Between concurrent coding agents, delegated subagents, CI/CD automation, monitoring, and embedded applications, a developer's share of enterprise agent traffic runs to about 100 million tokens per week — and agent traffic lands in the most expensive corners of the price sheet.



Annual API cost per developer · 5.2B tokens/year · illustrative scenario

1 developer	10 developers	100 developers
$100K	**$1.0M**	**$10.0M**

Modeled annual savings at the central 85% scenario

MODELED ANNUAL SAVINGS PER DEVELOPER

$100K

Central 85% scenario. The 80–90% range equals approximately $94,000–$106,000 per developer per year.

Headline assumptions

- 100M tokens per developer per week — measured, not estimated
- Interactive agents plus 24×7 automation · 52 weeks/year
- About 2.4 always-on agent-equivalents per developer at ~70 tokens/sec
- 25% output share, because reasoning tokens bill at the output rate
- 75% of GPT-6 Astra requests above the 272K long-context threshold
- 50/50 GPT-6 Astra and Claude Fable 5.1 — true frontier peers
- List prices verified September 7, 2026 · blended $22.62/M

Illustrative model, not a guarantee. Actual results depend on workload, cache behavior, context length, reasoning effort, model mix, internal hardware allocation, quality thresholds, and commercial pricing.

The 272K trapdoor

Past 272,000 input tokens, GPT-6 Astra reprices **the entire request** at 2× input and 1.5× output. Agents that accumulate repository context cross that line all day.

Reasoning bills as output

Thinking runs on by default on both flagships, and reasoning tokens bill at the **output rate** — five times input. A 1% output assumption badly understates agent cost.

Cold caches on automation

CI/CD and monitoring agents run further apart than the cache TTL, so every run re-sends context at **full uncached rates**. Always-on traffic is not cache-warm.

PRODUCT CAPABILITIES

The control plane for enterprise GenAI traffic.

Control policy, access, cost attribution, traffic, upstream capacity, and operational evidence from the individual caller key through the selected provider and model.

Customize routing policy

- Static, weighted, failover, dynamic-score, TypeScript, and external-policy routing
- Quality and capability contracts before policy selection
- Stable model groups spanning providers, accounts, and private endpoints

Attribute every dollar

- Owner, caller key and public token ID
- Project, environment, client, and model group
- Use projects for applications, business units, or cost centers
- Request-time cost and baseline savings

Shape caller traffic

- Request and token burst controls
- RPM, TPM, and concurrency
- Per-caller overrides
- Bounded queue or fail-fast behavior

Protect shared upstream capacity

- Provider, model, and target shaping
- Adaptive 429 and quota backoff
- Capacity pooling and compatible fallback
- Noisy-neighbor and fallback-storm protection

Enforce budgets before spend

- Daily, monthly, and lifetime budgets
- Input plus requested-output reservation
- In-flight concurrency protection
- Disabled, suspended, expired, and rotated keys

Protect compatibility

- Chat, Responses, and Messages
- Tools and structured outputs
- Images/VLM and reasoning
- Request-size, schema, and output-cap gates

Operate with evidence

- Latency, TTFB, throughput, attempts, fallback, cache, and errors
- Request IDs joining usage, traces, shapes, and safe diagnostics
- CLI and authenticated browser reports with relational rollups

Govern administration

- Server-side provider keys
- Basic Auth or OIDC admin identity
- Scoped authorization and metrics isolation
- Optional PII filtering and retention controls

QUALITY CONTRACTS

The cheapest model is only cheaper if it completes the job.

1. Define the workload, client surface, quality threshold, and cost/latency target.
2. Evaluate candidate models with unit tests, tool assertions, golden datasets, OCR targets, browser tasks, or product acceptance tests.
3. Promote only eligible targets into the deployment-defined model group.
4. Monitor outcomes, request-time cost, reliability, latency, and fallback; hold, split, or roll back when evidence changes.

> “Promote, hold, split, or roll back based on evidence—not model reputation alone.”

Smart Router makes policy measurable and reversible. It does not claim every workload improves automatically.

OPEN SOURCE BY DESIGN

Open source. Self-hostable. Built to be inspected.

Metrum GenAI Smart Router is Apache-2.0 open-source software. Review the behavior, run it in your environment, extend deployment-owned policy, and validate it against your workloads.

View the source

Deployment options

```
model_group: enterprise-coding
policy: deployment-owned
eligible_targets:
  - private-efficient
  - hosted-workhorse
  - frontier-reasoning
quality_gate: required
evidence:
  - request_time_cost
  - latency_and_throughput
  - attempts_and_fallback
```

GO DEEPER

Product evidence, not just product claims.

The public documentation covers configuration, compatibility, validation, security boundaries, reporting, deployment, and operations.

Product capabilities →

…lt …ability inventory.

API compatibility →

…nthropic, … request shapes.

Private inference →

…/LLM, …nd …e … endpoints.

Prove quality →

…uted …nst …l … controls.

Routing policy →

…ools, … … and fallback.

Usage reporting →

…st and … …ect, key, … upstream.

Caller shaping →

…est and … … per-caller policy.

Provider shaping →

…y …ound … pressure.

METRUM GENAI SMART ROUTER

Give every team one AI endpoint. Keep control behind it.

Start with the open-source software, explore the documentation, or get enterprise support, evaluation, and optional private-managed deployment from Metrum.

Explore the docs | View on GitHub | Get enterprise support