White paper 2 of 4

Multi-model AI with bring-your-own-keys:

Why a stable model abstraction, routing and fallback matter, and how BYOK keeps the AI bill with the provider, at cost.

About 6 min read · 1,293 words Print-friendly

CONTENTS

Summary

Why a stable model abstraction, routing and fallback matter, and how BYOK keeps the AI bill with the provider, at cost.

Why one model is rarely enough

Model providers release new versions often, and their strengths differ: some are better at long documents, some at fast conversational turns, some at code, some at low cost for high-volume work. A platform that binds every workflow to a single model family inherits that family's limits and its pricing.

AgenTorQ treats models as interchangeable engines behind a stable abstraction. Workmates are built once; the engine behind them can change. This paper explains the pieces — the lineup, routing, fallback, keys and policy — and how to pick a sensible starting setup.

A stable model abstraction

A model abstraction sits between your Workmates and the providers. Workmates, prompts and actions talk to the abstraction; the abstraction talks to whichever provider is selected. That separation is what makes it possible to route by task, fall back on error and swap providers without rebuilding anything, on an abstraction that is versioned and outlives any single provider.

Hundreds of models are available via live fetch from the connected providers, so new models can be selected as providers publish them.

The provider lineup

The platform supports 14+ providers on your own keys, plus self-hosted and any OpenAI-compatible endpoint. The descriptions below are the roles the site assigns to each family.

Provider family	Typical role
OpenAI	GPT-class models for reasoning, drafting and general enterprise workloads.
Anthropic Claude	Long-context, safety-focused models for careful reasoning and analysis.
Google Gemini	Multimodal models with strong grounding and large context windows.
xAI Grok	Fast, capable models for real-time reasoning and conversational workloads.

Provider family	Typical role
DeepSeek	Cost-efficient reasoning and coding models for high-volume tasks.
Mistral	Open-weight, efficient models for latency-sensitive and private deployments.
Groq	Ultra-low-latency inference for interactive, high-throughput workflows.
Perplexity	Answer models with web grounding and dated, sourced results.
SambaNova, Cerebras, MiniMax, AI21, Hugging Face	Additional model families available on your keys.
Ollama and OpenAI-compatible endpoints	Self-hosted or private models that keep sensitive workloads inside your network.

Routing by task, cost, latency and policy

The Intent Router grades each request and decides what it needs — a live data lookup, a knowledge answer, pure reasoning or a real action — and re-grades on every turn instead of following hardcoded routing. Multi-model routing then picks the engine by task, cost, latency and policy.

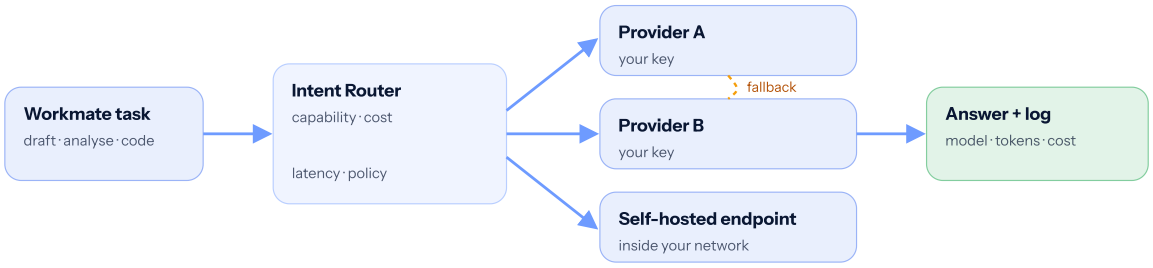

Figure 1. A task is routed to an approved provider, with fallback and logging.

A practical way to think about routing is by the shape of the work: a quick classification or extraction does not need the largest model; a long contract review benefits from long

context; code changes benefit from a model that is strong at code. Routing lets each of those go to a suitable engine without users having to choose.

Fallback without lock-in

Providers have outages, rate limits and regional restrictions. Automatic fallback sends the request to another approved provider when the first one fails, within the latency and cost limits you set. Because Workmates are built against the abstraction, fallback does not change their behaviour or their tools — only the engine.

The same property removes lock-in: if a provider's pricing or terms change, you can move a workflow to another provider without rebuilding the Workmate.

BYOK: the AI bill stays with the provider, at cost

With bring-your-own-key, you add your own API keys for the providers you want and pay them directly. There is no per-message metering and no per-seat AI tax on top; the platform price and the model bill are separate lines. The Pricing page describes the platform side, including a 30-day free trial on every plan.

Keys are held in the credential vault, isolated per organization and kept out of prompts, logs and client storage. Shared engine keys can be made available to a team WorkSpace while personal tokens stay private.

> **Transparency:** model, token and cost tracking is recorded per call, with per-user and per-team attribution, so spend is visible by workflow rather than hidden inside a bundled price.

Model access is a policy decision

Security teams can allow or restrict specific providers and models per role, team or workflow, apply data-handling rules per provider and per route, and see every model call in the audit trail. Named personal data is masked before any prompt leaves, and only the masked prompt reaches the model you chose — whichever model that is.

- Per-role and per-workflow model allowlists.
- Data-handling rules applied per provider and per route.
- Automatic fallback within latency and cost limits.
- Every model call logged.

Self-hosted and private endpoints

Some workloads should not leave your network at all. Ollama and any OpenAI-compatible endpoint can be added as providers, so sensitive work can be routed to private or region-locked models while other work uses hosted providers. The routing and policy layer treats them the same way.

Matching engines to Workmates

Model choice is easiest to reason about per Workmate, because each Workmate has a clear job. The platform ships role-aware Workmates — Sales, Service, Ops, Admin and Developer — and a library of templates across divisions, and each one runs on the model you choose.

- **Sales Workmate.** Briefs reps on live pipeline and drafts follow-ups: a strong general model for drafting, with a faster one for quick lookups.
- **Service Workmate.** Answers from records and knowledge with citations: long context helps when case threads are long.
- **Ops Workmate.** Runs multi-step busywork across connectors: a cost-efficient model suits high-volume, well-defined steps.
- **Admin and Developer Workmates.** Explain schema and propose code: a model that is strong at code and careful reasoning.

Auto-Pilot crews (in early access) follow the same idea: each crew picks its own engine and model from the AgenTorQ Engine Console. The suggestions above are starting points, not rules; the logs should drive later changes.

Files, vision and web grounding

Model choice also interacts with the content a Workmate reads. AgenTorQ reads PDF, Word, Excel and CSV natively and runs OCR on scans and screenshots, then summarizes and reasons over them with any model — including models without native vision. That removes one common reason to standardize on a single multimodal provider.

For questions that depend on public information, model-agnostic web search brings back current, sourced results with dates, and answer models with web grounding are available as a

provider family. Either way, the answer cites where and when the facts came from, next to your internal data.

Evaluating providers in practice

Provider comparisons in the abstract are rarely useful; what matters is how a model performs on your prompts, with your data, at your volume. The platform includes latency testing, a prompt library and streaming responses, which together make it practical to try the same task on two providers and compare speed and output before changing a route.

Three habits help. Keep the prompt and grounding constant when comparing, so only the engine changes. Compare on a handful of real, representative tasks rather than a single example. And record the decision — which workflow moved to which engine, and why — so the next review starts from evidence rather than memory.

Choosing a starting setup

1. Pick two hosted providers you already have agreements with, so fallback is available from day one.
2. Add a self-hosted or OpenAI-compatible endpoint if you have sensitive workloads.
3. Set allowlists per role before inviting users; start narrow.
4. Let the router choose by task, and review model, token and cost tracking after the first weeks.
5. Move specific workflows to a different engine only when the logs show a reason.

For the security side of these choices, see the companion paper on governed enterprise AI.

Sources

Product statements in this paper restate the AgenTorQ website:

- agentorq.com/platform
- agentorq.com/connectors
- agentorq.com/security
- agentorq.com/standalone
- agentorq.com/pricing

General context links point to the public pages named in the text. This paper contains no market statistics.