ExoBrain

AI Cost Analysis: June 2026

Introduction

The cost of running AI has become one of the most confusing line items in the modern technology budget. The same piece of work can cost a fraction of a penny or several pounds depending entirely on how you buy the compute behind it. A daily inbox summary might be billed as a flat subscription, a metered credit, a per-token API call, or simply the depreciation on a machine sitting under your desk. The numbers move monthly, the vendors deliberately obscure their allowances, and the cheapest option on paper is rarely the cheapest in practice.

This guide is designed to cut through that. It starts with the building blocks of how AI is sold, walks through the full provider landscape, explains where the market is travelling, and then grounds everything in a single real piece of knowledge work: a complex proposal and RFP response that we measured end to end. By the close you should be able to look at any AI pricing page and understand not just the headline number, but the cost model underneath it.

A note on figures throughout. Where a vendor publishes prices, we use them. Where a vendor deliberately withholds allowances, as the major subscription labs do, every figure is a third-party reverse-engineered estimate and should be treated as directional. We flag these clearly. All prices are current as of June 2026 and are quoted in US dollars unless a figure is naturally denominated in pounds.

1: The building blocks

What a token actually is

Everything in AI pricing reduces to the token. A token is roughly four characters of text, or about three quarters of a word. Both the text you send to a model and the text it generates are counted in tokens, and almost every pricing model is, underneath, a way of charging for tokens consumed.

The first thing to understand is that input and output tokens are not priced equally. Output tokens, the text the model generates, are consistently the more expensive of the two, typically by a factor of three to ten. This matters enormously for agentic work, which generates large volumes of output as it reasons, calls tools and narrates its progress.

The second thing to understand is the cache. Modern AI workloads, especially agentic ones, resend the same context repeatedly. Each turn of an agent re-submits the growing conversation history. Providers offer prompt caching to avoid charging full price for this repetition: the unique context is written to a cache once, then read back at a steep discount, often around ninety per cent off the standard input rate. For any serious agentic workload, caching is not a minor optimisation. It is the single largest lever on cost, and we will see it turn a sixty-dollar task into an eight dollar one later in this guide.

The four ways AI is sold

Beneath all the marketing, there are four fundamentally different ways to pay for AI, and they have varying cost behaviours.

- **Per-token APIs** charge you for exactly what you consume, metered to the token. Costs scale linearly with use. Best for spiky, programmatic or unpredictable workloads: you pay nothing when idle and precisely your usage when busy.
- **Flat subscriptions** charge a fixed monthly fee for an allowance of usage, usually expressed as messages or prompts within a rolling window rather than a hard token count. The marginal cost of each additional task is effectively zero until you hit the cap, so the more you use a subscription, the cheaper each task becomes. This is the opposite curve to metered billing.
- **Owned or rented hardware** turns AI into a fixed infrastructure cost. Once the capacity is paid for, the marginal cost per task approaches zero, bounded only by electricity and your ability to keep the hardware busy. This is the volume play: cheap at scale, wasteful when underused.
- **Routing and fusion APIs** sit on top of the other three. Instead of picking one model, you send your prompt to a meta-API that runs several models in parallel and returns a single synthesised answer.

The art of managing AI cost in 2026 is matching each workload to the cost model that suits it, rather than defaulting to whichever the vendor markets hardest.

2. The provider landscape

The market has stratified into seven recognisable tiers, running from fully managed and least flexible to fully self-operated and most flexible. Understanding which tier a provider sits in tells you most of what you need to know about its pricing and its trade-offs.

Tier	What they sell	Representative players	Flexibility
1. Foundation Model APIs	Closed-weight models behind a per-token API	OpenAI, Anthropic, Google, xAI	Easiest, least customisable
2. Inference / Serverless APIs	Optimised hosting of mostly open models as APIs	Fireworks, Together AI, DeepInfra	Low management
3. In-between Serverless GPU	Per-second / per-GPU-hour control, still managed	Modal, Baseten, Replicate	Medium control
4. Neoclouds / AI Clouds	GPU capacity and inference as a service	CoreWeave, Lambda, Nebius, DataCrunch	High control
5. Hyperscalers	Managed platforms with governance and SLAs	AWS Bedrock, Azure AI Foundry, Vertex AI	Bundled enterprise platform

6. AI Hardware / DIY	Buy and operate your own accelerators	Self-managed fleets, local machines	Hardest, most customisable
7. Routing and Fusion APIs	A meta-API over many models	OpenRouter Fusion, Sakana Fugu	Easy; model choice handled for you

The closed-weight foundation labs at tier one offer the highest intelligence and the least operational burden, but you are fully dependent on their pricing and their model choices. The neoclouds at tier four have become the engine room of serious self-hosting, renting raw Blackwell-class GPUs at a fraction of hyperscaler rates. The hyperscalers at tier five bundle governance, compliance and observability, which is genuinely valuable for regulated enterprise work but commands a substantial premium. And tier six, once the preserve of well-funded research labs, has been democratised by a new generation of capable hardware that fits on a desk.

The pure token cost picture

For the per-token APIs, the picture in June 2026 looks like this. We show input, output, and a blended rate that weights three input tokens to every output token, the convention Artificial Analysis uses for its headline cost metric.

Model	Provider	Input $/1M	Output $/1M	Blended $/1M	Intelligence
Qwen3.6 35B A3B	Alibaba	0.248	1.485	0.56	32
GLM-5.2	Z.ai	1.40	4.40	2.15	51
Gemini 3.5 Flash	Google	1.50	9.00	3.38	50
Claude Sonnet 4.6	Anthropic	3.00	15.00	6.00	36
Claude Opus 4.8	Anthropic	5.00	25.00	10.00	56
GPT-5.5	OpenAI	5.00	30.00	11.25	55

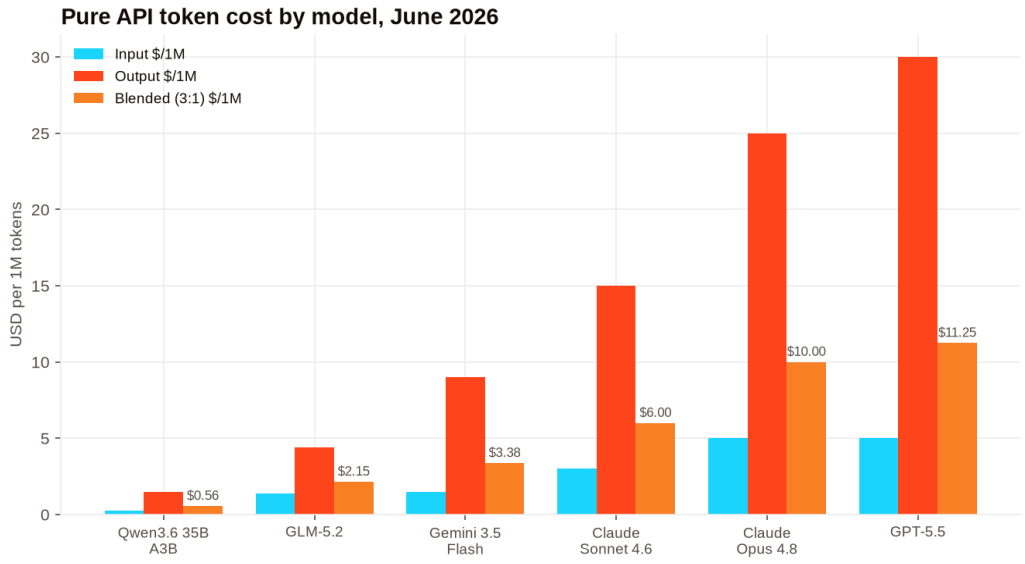

Figure 1. Pure API token cost by model, June 2026.

The spread is enormous. The most expensive frontier model costs roughly twenty times the cheapest open-weight option per blended token. But raw price tells only half the story. The other half is intelligence, and that is where the picture has shifted most dramatically over the past year.

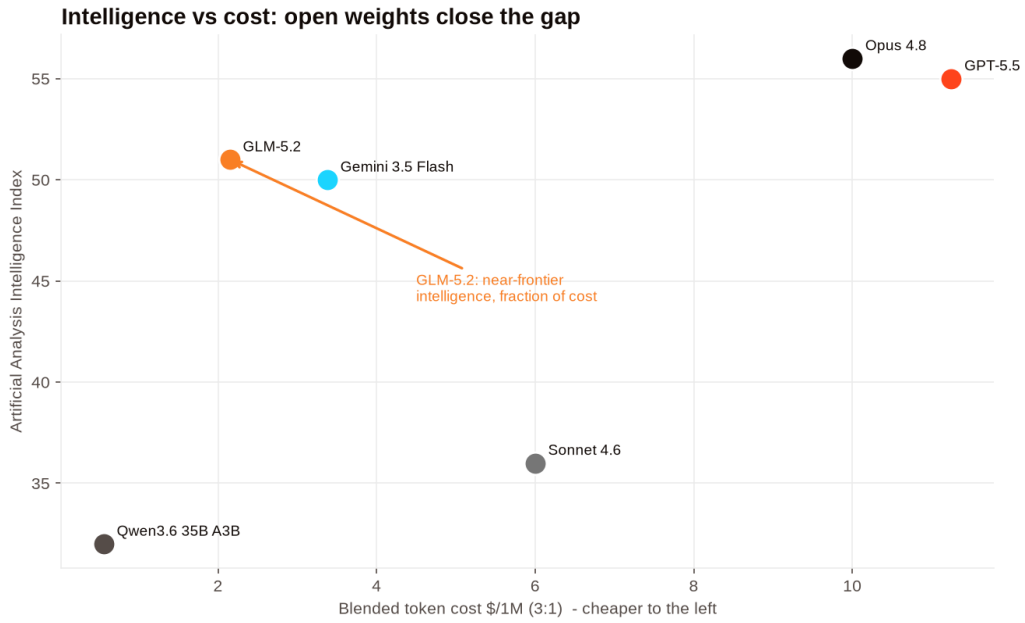

Figure 2. Intelligence versus cost. Open weights now sit close to the frontier at a fraction of the price.

The scatter above is the single most important chart in this guide. The frontier closed models, GPT-5.5 and Opus 4.8, sit top right: highest intelligence, highest cost. What is new in 2026 is the cluster in the upper left. **GLM-5.2 delivers an intelligence score of 51, within striking distance of the frontier, at a blended cost of barely two dollars per million tokens.** This is the open-weight disruption in a single data point, and it reshapes every build-versus-buy conversation that follows.

ExoBrain

3. The direction of travel

Everything is moving towards usage-based pricing

The clearest trend in 2026 is the migration from flat, all-you-can-eat access towards metered, usage-based billing. The economics of agentic AI have forced this. When a single user could consume the compute of fifty ordinary users by running autonomous agents around the clock, the flat-rate model became unsustainable for the vendors subsidising it.

Anthropic made the most visible move. From 15 June 2026, programmatic and headless agent usage, through the Agent SDK or command-line invocation, no longer draws from a subscriber's interactive allowance. Instead it consumes a separate monthly credit, sized to match the subscription fee, billed at standard API rates. Interactive use of Claude Code and Claude Cowork still draws from the subscription pool, but the principle is established: autonomous, high-volume agent work is being pushed onto metered billing. This is the shape of things to come across the industry.

Microsoft's pricing problem

Microsoft sits at the expensive end of the current landscape, and the reason is structural. Copilot Cowork, now generally available, is not priced as raw inference. It is priced in Copilot Credits at one cent each, charged per task according to model use, context retrieval, tool calls and runtime. A task is graded as light, medium or heavy, costing roughly 125, 500 or 1,200 credits respectively, which is to say between one and twelve dollars per task. Cowork also requires a Microsoft 365 Copilot licence before it can run at all, so the credit cost sits on top of an existing per-seat fee.

To put that in perspective, a heavy Cowork task at twelve dollars buys, in raw GLM-5.2 inference terms, well over eight million input tokens. The actual model work behind a typical governed task is a tiny fraction of that. Our analysis suggests Cowork carries something between a five-fold (Sonnet API) and a hundred-fold (GLM self-hosted) premium over raw inference for everyday knowledge work.

Microsoft has signalled that a more efficient generation, sometimes referred to as Cowork 1, should bring credit consumption down, and that is the number to watch. Until it lands, Cowork is a premium governance product, not a cost-effective inference engine.

The open-weight challenge

The second great force reshaping the landscape is the maturation of open-weight models, and GLM-5.2 is the standard-bearer. It is competitive on two fronts at once. As a usage-based API it undercuts every Western frontier model. As a subscription, through the GLM Coding Plan, it offers prompt-capped tiers from roughly six dollars a month at the entry level to a hundred and sixty at the top. And because the weights are openly licensed under MIT, it can be self-hosted entirely, removing the per-token meter altogether.

This combination, near-frontier performance available simultaneously as cheap API, cheap subscription, and self-hostable weights, is genuinely new. It means an organisation is no longer forced to choose between capability and control. A year ago, sovereignty and cost control meant accepting a significant intelligence penalty. GLM-5.2 has largely closed that gap.

ExoBrain

Blackwell arrives in the EU, and the desk becomes a data centre

The third force is hardware availability. Nvidia's Blackwell generation, the B200 and its variants, has reached European neocloud providers in earnest. Sovereign, CLOUD Act-free operators in Finland, the Netherlands, France and the UK now rent B200 capacity by the hour, which for the first time makes fully self-hosted, EU-resident inference of a near-frontier model practical for mid-sized organisations.

At the same time, the bottom of the market has been transformed. A capable local machine with at least 96GB of fast unified memory, whether an AMD Strix Halo mini-PC, a MacBook Pro or Studio, an Nvidia DGX Spark, or a high-end RTX 5090 desktop, now costs between roughly four thousand dollars and can run a quantised open-weight model entirely offline. The combination is powerful: a personal machine handles the constant stream of simpler work at zero marginal cost, while a rented or owned B200 cluster handles the heavier team-scale load, and the frontier APIs are reserved for only the hardest problems.

Where this leaves the big labs and their subscriptions

Our prediction is that the major US labs will continue to tighten subscription economics, pushing autonomous and high-volume usage onto metered billing as Anthropic has begun to do. But they cannot abandon the flagship subscription tiers, the Claude Max and ChatGPT Pro accounts at one and two hundred dollars a month. These accounts retain their heaviest, most capable and most influential users, the senior engineers and power users whose advocacy shapes adoption across entire organisations. Losing them to a competitor or to self-hosting would be far more damaging than the cost of subsidising them. Expect the headline tiers to survive, with the fine print around them growing steadily more restrictive.

A related and often missed point concerns enterprise licensing. Enterprise and business subscription tiers are, on a per-unit-of-work basis, frequently less cost-effective than the individual flagship accounts, because they are priced for breadth of deployment and governance rather than depth of usage. For genuine power users, and for the emerging class of agentic-accelerated knowledge workers whose output depends on uninterrupted access to the best models, the most cost-effective arrangement is often an individually funded personal Max or Pro account rather than a seat on a corporate plan. The days when a thirty-dollar Copilot licence was sufficient to capture the true value of AI for a serious professional are over. The value now sits with those running the heaviest accounts and the smartest models, and the pricing increasingly reflects that.

The rise of routing and fusion APIs

A fourth way to consume AI is emerging that sits on top of the other three. Rather than choosing a single model, you send your prompt to a meta-API that orchestrates several models on your behalf, then returns one answer. Two products make the shape of this trend clear: OpenRouter Fusion and Sakana Fugu. They take different routes to the same idea, that for hard problems a coordinated panel of models beats any single one.

OpenRouter Fusion turns a single prompt into a small multi-model deliberation. A panel of expert models analyses the prompt in parallel, with web search and web fetch enabled, and a separate judge model then synthesises their responses into a structured analysis covering consensus, contradictions, partial coverage, unique insights and blind spots before writing the final answer. The panel defaults to a quality preset, can be switched to a cheaper budget preset, or overridden entirely

so you choose the panel and judge yourself. The pricing model is the consequence of the design: you pay the sum of every underlying completion, every panel member plus the judge, not a single model call. It is positioned for research, expert critique and any situation where the cost of being wrong outweighs a few extra completions.

Sakana Fugu takes a more opaque but more integrated approach, billed as a multi-agent system delivered as one model through a single OpenAI-compatible API. Instead of running a fixed panel, it learns to assemble agents from a pool and coordinate them through collaboration patterns derived from its TRINITY and Conductor research, handling model selection and switching for each task. It comes in two variants, Fugu for balanced performance and low latency, and Fugu Ultra for maximum quality on complex multi-step reasoning. The headline claim is frontier-level performance without single-vendor dependency, with published benchmarks placing both variants at or above Opus 4.8, Gemini 3.1 Pro and GPT-5.5 on coding, reasoning and scientific tasks. The routing itself is proprietary and not exposed by design.

4. GLM-5.2 and the self-hosting approach

Because GLM-5.2 sits at the centre of the self-hosting opportunity, it is worth understanding how it is actually run.

Why the model size and quantisation matter

GLM-5.2 is a large mixture-of-experts model with 744 billion total parameters, but only a small fraction are active for any given token. The challenge is memory. At eight-bit precision, the weights require roughly 860 gigabytes, which forces a full eight-GPU B200 node and the cost that implies.

The breakthrough is quantisation. NVFP4, Nvidia's four-bit floating point format and the format used in production for GLM-5.2 by serving specialists, shrinks the weights to around 372 gigabytes. That fits comfortably within three B200 GPUs and their combined 540 gigabytes of high-bandwidth memory, leaving headroom for the key-value cache that holds active conversations. Critically, NVFP4 does not merely fit the model into less memory. It also raises throughput compared with eight-bit, because it unlocks the faster Blackwell tensor cores and reduces pressure on memory bandwidth, improving both the time to first token and the sustained generation rate.

The practical consequence is that a three-GPU B200 deployment, which was not viable at eight-bit precision, becomes a sensible unit of self-hosted capacity at four-bit. There is a trade-off to note: four-bit quantisation can introduce mild quality degradation on the longest and most demanding reasoning chains, so it should be validated against your own workloads before it is relied upon.

What three B200s can actually serve

A realistic three-B200 NVFP4 deployment delivers roughly 2,800 tokens per second of aggregate throughput, with around 125 gigabytes of memory left for the key-value cache. How many people that serves depends entirely on the kind of work they do.

For knowledge workers doing chat and retrieval, with modest context and light output, the binding constraint is memory: how many open sessions fit in the cache. The deployment supports roughly 240 simultaneous active users. For agentic engineers running coding harnesses, with large repository-scale context and heavy output, the binding constraint is throughput, because these harnesses generate enormous volumes of tokens and run almost continuously. The same deployment supports only around 12 simultaneous engineers. The roughly twenty-fold gap between

the two figures is the most important architectural insight for anyone sizing self-hosted capacity: the same box serves twenty times more light users than heavy agentic ones, because the two workloads exhaust completely different resources.

The hosting hierarchy

This naturally produces a tiered architecture. A local machine running the smaller Qwen3.6 35B model handles the constant stream of simple, high-volume work at zero marginal cost. A three-B200 cluster running GLM-5.2 handles team-scale serving and heavier agentic work. And the frontier APIs are reserved for the hardest reasoning where quality is decisive. Each tier is the cheapest sensible option for its slice of the work.

5. The real test, a complex knowledge work task

Abstract per-token comparisons only go so far. To understand what these options really cost, we measured a single, representative piece of complex knowledge work from end to end: an RFP response generation pipeline. This is a three-stage workflow that reads and plans against a request for proposal, selects relevant credentials and case studies, drafts the response, and renders a branded deck. It is a genuine agentic knowledge-worker task, the kind that increasingly defines professional output.

The anatomy of the task

The measured run, on Claude Sonnet, took 18.3 minutes and achieved an accuracy score of 0.925 against the answer key. The token figures are revealing.

Measure	Value
Distinct, unique tokens (the real work)	274,704
Cumulative input across all turns	20,625,497
Cache reads (re-sent context served from cache)	20,388,006
Context-resend multiplier	86.9x
Output tokens	37,213
Measured API cost (with caching)	$8.10

The headline number is that context-resend multiplier of 86.9. The actual unique work in this task was about 275,000 tokens, but the agentic harness, by re-sending its growing context at every one of its 351 steps, processed more than twenty million tokens cumulatively. Almost ninety-nine per cent of all input was served from cache. This is the defining characteristic of modern agentic work, and it explains why caching dominates the economics.

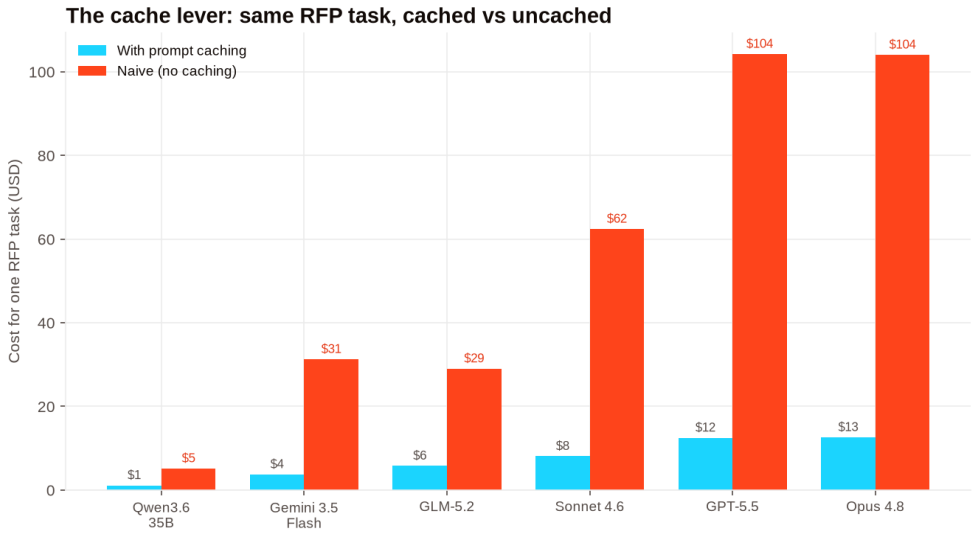

Figure 3. The cache lever. Without caching this single task would cost up to one hundred dollars; with caching, a fraction of that.

The chart above shows what caching is worth on this single task. Without it, billing every re-sent token at full input rate, the task would cost over sixty dollars on Sonnet and more than a hundred on the frontier models. With caching, the same task costs eight dollars on Sonnet, a saving of around eighty-seven per cent. Any cost comparison that ignores caching is meaningless for agentic work.

The full comparison

With the task understood, we can now cost it across every option in the landscape: the model APIs, Copilot Cowork (and Anthropic's Cowork which is the same as the subscriptions if used under a subscription, or the API respectively), the self-hosted B200 cluster, the local machines, and the subscription plans. The result is the most complete answer we can give to the question, what does this actually cost?

#	Option	Category	Cost per RFP
1	3x B200 at 12 concurrent	Self-host	$0.16
2	3x B200 best case (prefix cache)	Self-host	$0.17
3	Local Qwen, RTX 5090	Local	£0.20
4	Local Qwen, Strix Halo	Local	£0.45
5	Local Qwen, DGX Spark	Local	£0.85
6	Local Qwen, Mac M4 Max	Local	£0.99
7	Qwen3.6 35B API	Model API	$1.13

8	Claude 20x Max subscription (Sonnet)	Subscription	$1.44
9	3x B200 (measured latency)	Self-host	$1.96
10	Claude 5x Max subscription (Sonnet)	Subscription	$2.86
11	Gemini 3.5 Flash API	Model API	$3.75
12	GLM-5.2 API	Model API	$5.80
13	Claude Sonnet 4.6 API	Model API	$8.10
14	ChatGPT Pro subscription*	Subscription	uncapped, see note
15	GPT-5.5 API	Model API	$12.50
16	Claude Opus 4.8 API	Model API	$12.61
17	Copilot Cowork	Governed agent	$27.77

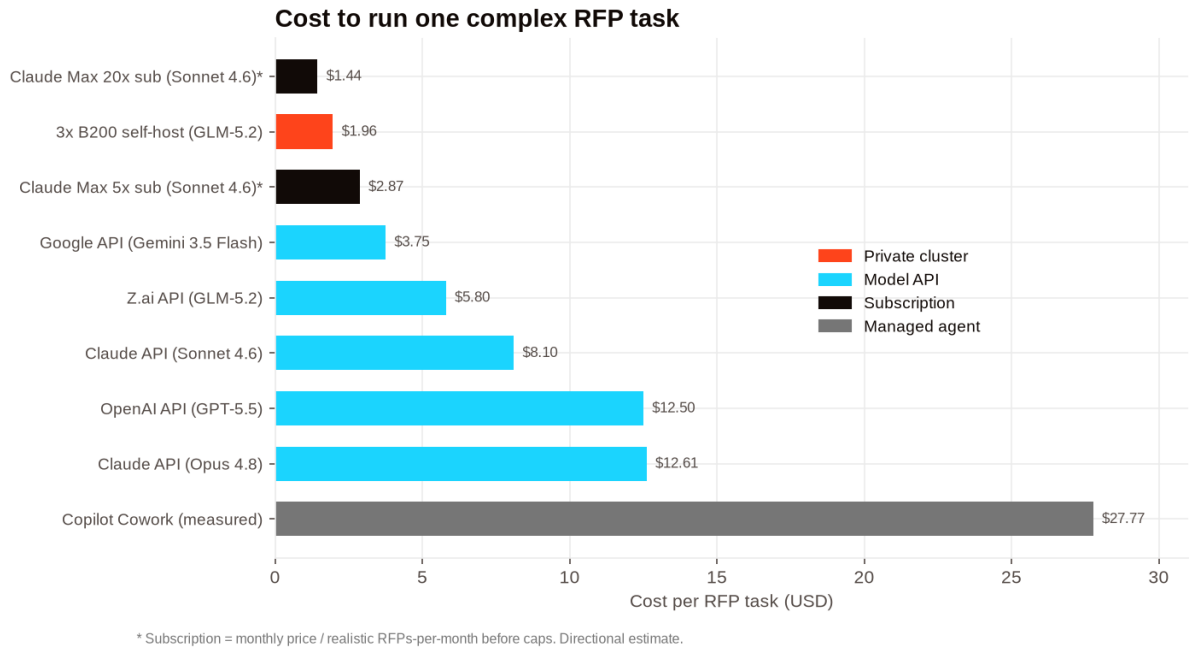

Figure 4. Cost to run one complex RFP task across every option, cheapest first.

The measured baseline, running on the Sonnet API at $8.10, sits in the expensive third of the range. It is a perfectly reasonable choice, with excellent measured quality, but it is far from the cheapest way to do this work.

Self-hosting and local hardware dominate the cheap end. A well-utilised B200 cluster does the task for around sixteen cents, and a local machine for a similar amount, because once the capacity is paid for the marginal cost is essentially electricity. The caveat is real, though: these figures assume high utilisation and, for the local tier, a model of materially lower intelligence than the frontier.

The subscriptions sit surprisingly low, once their allowances are read correctly. A personally funded Claude Max account does this task for somewhere between one and three dollars: roughly $1.44 on Max 20x and $2.87 on Max 5x. On a per-task basis the larger plan is the cheaper, because Anthropic's twenty-times multiplier buys far more headroom than the price step implies. For an individual running a steady stream of these tasks, a flagship subscription is among the most cost-effective routes to frontier-quality output that exists, second only to high-utilisation self-hosting.

The governed and frontier options anchor the expensive end. The frontier APIs sit above twelve dollars, but the costliest way to run this particular task by some margin is Copilot Cowork. A single measured run of this RFP consumed 2,777 Copilot Credits, which at one cent per credit is $27.77, more than double the headline heavy-task figure and over three times the measured Sonnet API baseline. For Cowork that premium buys governance and Microsoft 365 grounding; for the frontier APIs it buys the very best reasoning. The gap between Cowork's per-task tiering and its real consumption on a genuinely heavy task is the single most important caveat for anyone budgeting governed-agent work.

A necessary note on subscription figures

Neither OpenAI nor Anthropic publishes official token allowances, so every subscription number here is reverse-engineered from observed throttling and should be treated as directional. More importantly, a flat subscription does not have a natural cost-per-task. The only way to derive one is to divide the monthly fee by the realistic number of these heavy tasks you could run before hitting the usage caps.

Running this task repeatedly on a Max account, five to ten consecutive runs barely touched a single five-hour window, because within a window the re-sent context is served from cache and does not re-meter at full weight.

The strategic reading is that subscriptions are genuinely cheap per task across a wide band of low to moderate volume, considerably cheaper than this guide's earlier estimates suggested, and they only throttle at the high-throughput extreme, where an agency running many RFPs in parallel would saturate even a Max 20x window. That is exactly the point at which a self-hosted cluster or a tiered local-plus-cluster architecture earns its keep.

Part six: putting it together

No single option wins outright, because the right answer depends on volume, on how much intelligence the work genuinely requires, and on how much governance and sovereignty matter. But the shape of a sensible strategy is now clear.

For occasional or unpredictable work, a per-token API is the simplest and often the cheapest choice, and GLM-5.2 at under six dollars for our test task offers near-frontier quality without any infrastructure. For an individual power user or agentic knowledge worker, a personally funded Claude Max or ChatGPT Pro account remains the most cost-effective way to access the best models at depth,

far more so than an enterprise seat, and at one to three dollars a task it is now demonstrably so. For an organisation running genuine volume, a self-hosted GLM-5.2 cluster on EU-resident B200 capacity transforms the economics, taking the per-task cost into the cents, provided utilisation stays high and the slightly lower quantised quality is acceptable. And underneath all of it, a local machine running a smaller open model absorbs the constant stream of simple work for nothing but electricity.

A word, finally, on the managed-agent route. Copilot Cowork's measured cost of $27.77 for this task places it in a category of its own at the top of the range, and the lesson is not that governed agents are bad value but that their headline task tiers can badly understate real consumption on heavy work. Anyone budgeting Cowork should price it from measured runs, not from the light, medium and heavy bands, and should reserve it for the cases where Microsoft 365 grounding and per-user governance are the point rather than the throughput.

The organisations that will manage AI cost best in the years ahead are not those that pick a single provider, but those that route each task to the cheapest tier capable of doing it well. The building blocks are now all in place to do exactly that.

Methodology

The token prices and intelligence scores in this guide are drawn from published vendor rates and from Artificial Analysis, retrieved in June 2026. The Artificial Analysis Intelligence Index is a composite across nine evaluations including reasoning, coding and knowledge benchmarks; higher is more capable.

The RFP task is a single measured run, reconstructed from the session transcript using one-hour prompt-cache economics rather than provider billing telemetry. It should be treated as indicative rather than as a multi-run average. The per-model costs for that task were calculated by applying each model's published rates to the measured token footprint: the unique input billed once as a cache write, the re-sent context billed at the cached-read rate, and the output at the output rate. As a validation, our reconstruction of the Sonnet cost reproduces the measured figure of $8.10 to the penny, which gives us confidence in the same method applied to the other models.

The Copilot Cowork figure is likewise a single measured run, taken from the credit balance consumed by one complete RFP on a previously unused account: 2,777 Copilot Credits, billed at one cent each. It should be treated as indicative, and notably it ran on Sonnet, the same model as the measured API baseline, which makes the roughly threefold gap between the two a clean reading of the managed-agent wrapper premium.

Self-hosting costs assume a three-B200 NVFP4 deployment at EU neocloud rates, modelled at business-hours utilisation of 174 hours per month, with throughput of approximately 2,800 tokens per second derived from published eight-GPU benchmarks scaled to three GPUs and adjusted for the NVFP4 speed-up and a small-node mixture-of-experts penalty. Real figures vary by roughly a quarter depending on serving configuration. Local hardware costs are amortised over three years plus electricity.

All subscription token allowances are third-party reverse-engineered estimates, not vendor-published figures, and the independent estimates disagree widely. Where they conflict, we have used the interpretation that applies Anthropic's published usage multipliers literally to the Pro base, because it matches our own measured behaviour of running many of these tasks within a single five-hour window. They are included for completeness and marked throughout as directional. Prices in this

market move quickly. Every figure should be re-verified against current vendor pricing before any procurement decision.

Sources

1. Artificial Analysis, model intelligence and token pricing comparison. https://artificialanalysis.ai/models

2. Z.ai and Together AI, GLM-5.2 API pricing and Coding Plan guide. https://lushbinary.com/blog/glm-5-2-api-pricing-glm-coding-plan-guide/

3. GLM-5.2 self-hosting, vLLM hardware and cost. https://ofox.ai/blog/glm-5-2-self-host-vllm-hardware-cost-2026/

4. NVFP4 quantisation and GLM-5.2 deployment. https://www.spheron.network/blog/deploy-glm-5-2-gpu-cloud/

5. Baseten, building a fast GLM-5.2 API in production. https://www.baseten.co/blog/how-we-built-the-worlds-fastest-api-for-glm-52/

6. vLLM GLM-5.2 recipe. https://recipes.vllm.ai/zai-org/GLM-5.2

7. Lambda, GLM-5 inference throughput. https://lambda.ai/inference-models/zai-org/glm-5

8. GLM-5.2 capacity and effort token consumption (Verdent). https://www.verdent.ai/guides/what-is-glm-5-2

9. GLM-5.2 open-weight overview (Labellerr). https://www.labellerr.com/blog/glm-5-2-open-weight-ai-model/

10. Microsoft Copilot Cowork GA and credit pricing (Charles Lamanna). https://www.linkedin.com/posts/charleslamanna_copilot-cowork-is-now-generally-available-activity-7472665981962604544-GTj-

11. Anthropic agent billing change, June 2026 (The Register). https://www.theregister.com/ai-ml/2026/05/14/anthropic-tosses-agents-into-the-api-billing-pool/

12. Claude Max plan details (Anthropic). https://support.claude.com/en/articles/11049741-what-is-the-max-plan

13. Claude token-window estimates (Faros). https://www.faros.ai/blog/claude-code-token-limits

14. Claude Team vs Max usage analysis (lord.technology). https://lord.technology/2026/03/28/claude-team-premium-vs-max-plans-usage-limits-pricing-and-which-to-choose.html

15. Claude subscription value and credit analysis (ksred). https://www.ksred.com/claude-code-pricing-guide-which-plan-actually-saves-you-money/

16. ChatGPT plan structure and limits (Northflank). https://northflank.com/blog/chatgpt-usage-limits-free-plus-enterprise

17. ChatGPT Pro context and features (Zapier). https://zapier.com/blog/chatgpt-pro/

18. Local inference throughput benchmarks (Hardware Corner). https://www.hardware-corner.net/rtx-5090-llm-benchmarks/

19. Local LLM tokens-per-second benchmarks (Presenc AI). https://presenc.ai/research/local-llm-tokens-per-second-benchmarks-2026

Prepared by ExoBrain, June 2026. All figures are point-in-time estimates for guidance only and should be independently verified before any commercial decision.