

Which Meter Is Running?

A field guide to the invisible billing layer behind AI agents and automations.

The model name does not tell you who gets billed.

Ask where the work runs, which credential authorizes it, and which tools it calls. That route - not the logo in the chat window - determines the meter.

Coverage: xAI/Grok, X API, Cursor, Google, Anthropic, and OpenAI. Pricing and product boundaries checked against public sources on September 15, 2026.

Seat 01
subscription or workspace

Model 02
tokens, credits, or requests

Tools 03
search, storage, runtime

Platform 04
the app that routes the work

The invoice follows the route, not the logo

A model can appear inside a consumer app, a coding editor, an agent platform, or a custom API workflow. Those surfaces may use the same underlying model while charging four different ways.

The operating rule: name the execution path before you estimate cost.

1. SURFACE

Where did you click?

Consumer app, IDE, cloud console, or custom automation.



2. CREDENTIAL

What opened the door?

Subscription login, workspace seat, API key, or cloud project.



3. CHARGE

What unit moves?

Messages, credits, tokens, tool calls, storage, or runtime.

Four questions identify most meters

1

Which product surface?

Do not stop at the model name. “Claude in Cursor” is a Cursor workload. “Claude through an Anthropic API key” is an Anthropic Platform workload.

2

Which credential?

A login usually points to a seat or subscription. An API key points to a developer account, project, or cloud bill.

3

Who hosts the request?

An intermediary can bundle several providers behind its own usage pool. The upstream logo may not own the bill.

4

Which extras fire?

Search, code execution, vector storage, voice, and long-running containers can add meters beyond model tokens.

“I already pay for the model” is not a billing diagnosis.

xAI and X can create three separate ledgers

The brands sit close together. The meters do not.

Route	Meter owner	Typical unit	Where to verify
Grok app	Consumer plan	Plan limits	Subscription settings
xAI API	xAI developer	Model tokens	xAI API console
xAI tools	xAI developer	Tool invocations	xAI usage detail
X platform API	X developer	Endpoint usage	X developer console

XAI DEVELOPER METER

Tokens are only the first line

xAI documents model-token charges and separate server-side tool charges. Web search, X search, code execution, attachment search, and collection search can each add their own usage line.

The durable signal is authentication: a workload using an xAI API key belongs on the xAI developer ledger, even if the operator also pays for a Grok consumer plan.

X PLATFORM METER

Reading and writing X is a different service

The X API handles platform data and actions such as reading posts or publishing through an integration. Public reporting describes a pay-per-use credit model with rates that vary by endpoint.

Do not treat an X subscription or a Grok subscription as an API allowance. Verify the live credit balance and endpoint rate in the X developer console before scaling.

Operator example: An automation can ask Grok to research a topic through xAI, then publish a result through the X API. One workflow; two developer meters. If it is launched from a paid automation platform, that platform may be a third.

The xAI pricing page announces a change to X Search billing effective September 21, 2026. This is exactly why cost estimates should carry a date and point to the live dashboard.

Inside Cursor, Cursor is usually the meter

Seeing “Claude,” “Gemini,” or “Grok” in the model picker does not automatically mean the corresponding provider account is being charged.

Default read: if the request starts and finishes inside Cursor using Cursor-managed access, it draws from Cursor's plan pools or overage rules.

CURSOR MODELS

First-party pool

Cursor documents a dedicated pool for its own model lineup. The pool resets with the monthly billing cycle.

OTHER MODELS

Third-party pool

Selected provider models draw from a separate included pool at the model's published API rate, with additional usage available depending on plan settings.

BYOK

Provider key

If a user deliberately supplies their own API key, the provider account behind that key may become the meter. Treat this as an explicit routing change.

The useful accounting label

Bad: “Used Anthropic for the coding task.”

Better: “Ran Claude inside Cursor; charged against Cursor's Other Models pool.”

Best: “Cursor / Other Models / Claude / overage disabled.”

The model is the engine. Cursor is the toll road.

What to watch

- **Auto routing:** the model can change between turns, so the unit economics can change without a new workflow.
- **Agent length:** a whole pull request costs more than a short edit because the agent reads, reasons, writes, and often retries.
- **Overage settings:** included usage and paid continuation are different states. Make the transition visible.

Google's brand is one umbrella; its billing is not

“Gemini” can mean a consumer app, AI features inside Google products, an AI Studio project, a Gemini API key, or a Vertex AI workload.

SUBSCRIPTION SURFACE

Google AI plans

Google AI memberships provide access and higher limits across named Google experiences, including the Gemini app and selected productivity and creative tools.

Those benefits belong to the products listed in the plan. They are not a blanket promise that every API or cloud workload is prepaid.

DEVELOPER SURFACE

Gemini API and Google Cloud

Gemini API billing attaches to projects and billing accounts. Free-tier use may exist, but paid tiers, rate limits, and spend are managed through the developer billing path.

Vertex AI usage belongs on the Google Cloud bill. Additional services - storage, search, grounding, hosting, or data processing - can create adjacent line items.

A safer way to ask

Question	Why it matters
Gemini app or API key?	Separates subscription access from project billing.
AI Studio or Vertex?	Locates usage in developer billing or Cloud.
Free or paid project?	Determines whether calls consume free quota or money.
Which project ID?	Finds the actual budget, logs, and billing account.

Same prompt, different meter: Gemini app use draws on plan limits; an API script draws on its Google project.

Claude and ChatGPT each have a product side and a platform side

Anthropic

CLAUDE PLANS

Paid Claude plans cover the Claude product and named features. Anthropic's public plan page now includes Claude Code on paid consumer plans.

CLAUDE PLATFORM

API workloads use model-token pricing. Managed agents add runtime pricing, while web search and code execution can add tool or container charges.

Boundary test: a CLI or agent can be subscription-backed or API-backed depending on how it is authenticated and configured. Check the active credential and usage page; do not infer from the interface alone.

OpenAI

CHATGPT PLANS

ChatGPT subscriptions cover the ChatGPT product and its plan limits. OpenAI states directly that API usage is not included.

OPENAI API

API charges depend on models and features. Tokens are the base meter; applicable tools, storage, and execution environments can add separate charges.

Boundary test: if the workflow sends requests with an OpenAI API key, it belongs to the API organization and project - not the ChatGPT subscription attached to a browser login.

The chat product is a destination. The API is infrastructure.

Common trap: Teams buy a premium chat seat, then assume the same payment covers a background agent, Slack bot, or nightly batch job. If that automation authenticates through an API or cloud project, it has crossed into another billing system.

Put the meter label next to the workflow

The cheapest cost-control system is a line of metadata that survives handoffs.

MINIMUM VIABLE METER LABEL

platform / billing account / pool / model / tools / overage state

Before launch

- ✓ Name the product surface and owning account.
- ✓ Record whether access is subscription, platform, or BYOK.
- ✓ List model, search, storage, runtime, and external-action meters.
- ✓ Set a budget cap or disable automatic overage where possible.
- ✓ Run a small job and match it to the usage dashboard.

After launch

- ✓ Log cost by run, not only by month.
- ✓ Separate retries and background tasks from human-initiated work.
- ✓ Flag route changes when a model, key, or platform changes.
- ✓ Recheck official pricing before increasing volume.
- ✓ Review dormant keys, prepaid balances, and overlapping seats.

Three generic workflows

Research brief

Model tokens + web search + document storage. Email delivery itself is not an AI meter, though the mail service may have its own limits.

Coding agent

Editor usage pool + possible overage. Provider API charges appear only if the route uses provider credentials or a separately billed backend.

Social monitor

Platform data API + model inference + scheduler/hosting. One business outcome can cross three independent ledgers.

Scale rule: do not multiply a prototype cost by 10,000 until one real run has been reconciled against every relevant dashboard.

This map is meant to change

Provider plans, included pools, promotional rates, and tool fees move quickly. The durable asset is not a frozen price table. It is a repeatable way to prove which ledger moved.

The standing update

Future entries in *Which Meter Is Running?* should answer five questions:

- What changed?
- Which execution path is affected?
- What billing unit now moves?
- Where can an operator verify it?
- What should be re-tested before scaling?

Current as of September 15, 2026. Any price or product boundary should be checked again at the point of purchase or deployment.

Public sources

1. xAI API Pricing - tokens and server-side tools.
2. MediaNama: X API pay-per-use model - endpoint-priced credit structure.
3. Cursor Models & Pricing - included usage pools and third-party model rates.
4. Google One Help: Google AI Pro - subscription product benefits.
5. Google AI for Developers: Gemini API billing - projects, tiers, and billing accounts.
6. Anthropic Plans & Pricing - Claude plans, API models, managed agents, and tool charges.
7. OpenAI Help: ChatGPT and API billing - separate billing systems and feature charges.

When the bill looks mysterious, draw the route.