12: Costs: subscriptions vs API

What an hour with a coding agent really costs, and how to pay for it

Why this matters

- Agents re-send the whole conversation on every turn
- One agent hour at frontier API prices: **$24.30** uncached, **$3.73** cached
- Heavy users can burn hundreds of dollars a month on the API
- Subscription caps and terms changed several times in one year
- Leads need a cost number they can defend

Three ways to pay

Option	Who you pay	How it's billed	Watch out for
Chat subscription (Claude, ChatGPT)	A lab	Flat monthly fee	Caps: a few-hour window plus a weekly limit
Tool subscription (Cursor, GitHub Copilot)	A tool vendor	Fee with an allowance, then pay-as-you-go	Some models billed at API list price
API	A lab	Metered, per token	Uncapped, so the bill is too

What a token bill is made of

Line item	Relative price
Input: everything you send, history and tool results included	Base price
Output, including thinking tokens	Usually 4–6x input
Cache write (a premium at Anthropic, none at OpenAI)	1.25x for 5 minutes, 2x for 1 hour
Cache read: re-sending a prefix that's already cached	2.5–10% of input
Batch: async jobs that finish within hours	50% off
Fast / priority modes	About 2x

Tokenizers differ. Thinking bills as output.

Why agents eat tokens

The model is stateless, so the agent pays for its own history

Every turn re-sends everything

```
call 1    [prompt · tools · docs]                  20,000 tokens
call 2    [prompt · tools · docs] + 1 result       22,500 tokens
  …
call 60   [prompt · tools · docs] + 59 results  167,500 tokens
```

- Total input grows with the **square** of the number of turns
- Double the session, roughly quadruple the input tokens
- Big tool outputs, retries, subagents, and parallel agents multiply it

Prompt caching makes it affordable

- Each call starts with the exact prefix of the last one
- The provider keeps that prefix's processed state
- Re-sending it costs a small fraction of the input price
- Claude Code and Codex cache automatically
- In your own agent, put stable content first and conversation last
- Changing the system prompt or tools mid-session breaks the cache

Worked example: one hour, the assumptions

Assumption	Value
Model calls in the hour	60, about one a minute
First call	20,000 tokens
Context added per call	2,500 tokens
Output per call, including thinking	1,500 tokens
Compaction, or breaks over 5 minutes	None
Claude Opus 5.5 prices (as of November 2026)	$4 in · $5 cache write · $0.20 cache read · $20 out

Totals: **5,625,000** input tokens and **90,000** output tokens.

One hour on Opus 5.5: $24.30 or $3.73

	Without caching	With caching
Input	5.625 MTok × $4 = $22.50	—
Cache writes	—	0.1675 MTok × $5 = $0.84
Cache reads	—	5.4575 MTok × $0.20 = $1.09
Output	0.09 MTok × $20 = $1.80	$1.80
Total per hour	**$24.30**	**$3.73** (85% less)

API list prices as of November 2026.

The same hour at other price points

Tier (input / output)	Uncached	Cached	168 hours a month, cached
Top tier ($10 / $50)	$60.75	$7.96	~$1,340
Frontier ($4 / $20)	$24.30	$3.73	~$630
Mid ($2 / $10)	$12.15	$1.86	~$310

168 hours = 8 hours a day for 21 working days, one agent at a time.

What the numbers tell you

- Caching turns "unaffordable" into "plausible"
- With caching on, output becomes the biggest line
- Heavy frontier use costs hundreds of dollars a month at API prices
- Three agents at once ≈ three times the bill
- Light users are often cheaper on the API than a $20 plan
- A cache miss on a 150,000-token context costs $0.75

Paying for it

Subscriptions, cheaper options, and cost per task

Subscriptions (as of November 2026)

Vendor	Plan	USD a month	What the vendor says you get
Anthropic	Pro	$20	Claude Code included
	Max	$100	5x Pro usage; $100 a month in API credits
	Ultra	$200	10x Pro per week; no 5-hour limit (verified accounts)
OpenAI	Plus	$20	Codex on web, CLI, and IDE
	Pro	$100, $200, or $500	Higher limits; $500 adds "Ultrafast" mode

Individual tool plans: GitHub Copilot $10–$100, Cursor $8–$200.

A subscription is a flat fee with a cap

- Use less than the cap, and you overpay versus the API
- Heavy users report API-equivalent usage several times their fee
- In Aug 2025 Anthropic added weekly limits on top of 5-hour windows
- 2026 brought new tiers, renamed plans, smaller allowances for new subscribers
- "5x Pro" is a multiple, not a token count
- Rules for using a plan in third-party tools differ by vendor

Cheaper alternatives

Option	How it saves	Watch out for
Open-weight coding plans (GLM from $18/mo, Kimi, MiniMax)	Flat rates inside Claude Code, Codex, OpenCode, Cline	Trail the frontier on the hardest work; another provider and jurisdiction
Routing	Small model for easy work, frontier for hard	Small models cost 20–100x less per token than top tier
Local models	No per-token cost	Hardware and power; well behind for agentic coding

Judge cost per finished task, not per token

Same task, both cached	Frontier model	Budget model
Prices (input / output per MTok)	$4 / $20	$1 / $5
Calls to finish	30	90
Total input tokens	1.69M	11.81M
Cost with caching	**$1.68**	**$2.14**
Your attention	1x	3x

A quarter of the price per token, about seven times the input tokens.

When the API is the right choice

- Production features your users trigger
- CI and automation: review bots, triage, nightly test fixes
- Batch jobs like migrations, labeling, and evals, at 50% off
- Multi-user apps, with usage attributed and limited per user
- Light or bursty individual use
- Data terms: no training by default; zero data retention available

Cost controls for day one

- Spend limits and alerts per workspace, project, or key
- Separate keys per app, environment, and team
- Caching on by default, stable content first
- Small model by default; escalate on failure
- Cap `max_tokens`, effort, and turns per agent run
- Trim tool outputs, batch what can wait, review costs weekly

Teams and trade-offs

Pricing it in engineer-hours, and Bryan's take

Team math: price it in engineer-hours

Assumption	Value
US median developer pay, May 2025 (BLS)	$135,980
Loaded cost, about 1.4x salary	~$190,000 a year
~1,800 productive hours a year	~$100 per engineer-hour
A $200-a-month seat	~2 engineer-hours a month
Heavy API user, ~$630 a month	~6 engineer-hours a month
10 people on $200 seats	$24,000 a year; a 2% gain ≈ $38,000

The weak link is the "if": self-reported gains are unreliable.

Subscriptions vs API

> **Bryan's take:** API pricing is too expensive for heavy agentic use. Subscriptions are what make it affordable to work this way every day.

The strongest counterpoints

- Cheap until you hit the weekly limit on a Thursday
- Terms change; API prices are published per token
- You can't build on it; production and CI need the API
- Subscription value is tied to the vendor's harness
- Light users overpay; open-weight plans start under $20
- For teams, pooled API, caching, and routing can beat top-tier seats

Try it

On your own, measure your real cost:

```
npx ccusage@latest daily         # Claude Code, by day
npx ccusage@latest codex daily   # Codex, by day
```

1. Use your agent normally for a week, then run ccusage
2. Compare the API-equivalent cost with what you pay
3. Find your most expensive session. What made it expensive?

As a class or team, budget a 12-person team plus a PR-review bot, three ways.

Discussion

1. Why does doubling a session more than double its cost? How would you flatten it?
2. 75% cheaper per token, three times the turns: when is it still right?
3. Should a course give students API keys, subscriptions, or neither?
4. What would you measure to decide whether a $200 seat is worth it?

Read more

- Module 12: Costs: subscriptions vs API
- Anthropic API pricing docs: caching, batch, fast mode
- ccusage: local usage and cost reports for agent CLIs
- Becker et al., 2025: METR's study of AI and developer speed