

Do the cheap agents pay for themselves? Seven delegations, measured

Seven delegations from one working session, instrumented. What the cost routing actually bought, and the numbers I refuse to publish.

I run a set of cost-routing subagents for Claude Code. Each is pinned to the cheapest model tier that can do its job, and the orchestrating session is supposed to keep only the work that needs judgement. I had never checked whether the arrangement actually pays.

So I instrumented one working session. Seven delegations across two repositories, all of it real work — reviewing pull requests, writing agent definitions, translating a blog post, mapping a corpus — not a benchmark built to be measured.

The headline numbers

	Value
Delegations	7
Tokens, Haiku tier	231,369
Tokens, Sonnet tier	92,457
Tokens, total	323,826
Tool calls made by agents	122
Agent wall-clock	22.9 min
Cost as delegated (upper bound)	\$2.08
Same tokens at Opus 5 rates (upper bound)	\$8.10
Delegated cost as a share of that	26%

COST OF THE SAME 323,826 TOKENS

As delegated		\$2.08
All at Opus 5 rates		\$8.10

Both cost figures are upper bounds, because the runtime reports one token total per agent rather than an input/output split, and I charged every token at the higher output rate. The true figures are lower and the ratio between them is roughly stable, since the input:output price ratio is 1:5 on all three tiers.

The price gap is narrower than the premise it was built on

The repository's own pitch is "stop paying Opus prices for work a cheaper model does just as well." That framing dates from Opus 4.1 at \$15/\$75 per million tokens, where Haiku was one fifteenth the price. It is no longer true.

Model	Input	Output	Share of Opus 5
Claude Opus 5	\$5	\$25	—
Claude Sonnet 5	\$2	\$10	2/5
Claude Haiku 4.5	\$1	\$5	1/5

The ceiling on delegation savings is now 5x, and only for the Haiku tier. Sonnet work saves 60%, not 87%. Any claim resting on "roughly fifteen times cheaper" is quoting retired pricing.

There is a second-order effect pointing the other way. Opus 5 and Sonnet 5 use a newer tokenizer that, per Anthropic's own documentation, "produces approximately 30% more tokens for the same text"; Haiku 4.5 predates it. So a Haiku agent's token count and an Opus orchestrator's are not in the same units, and the same work costs more tokens on Opus. That widens the real gap above 5x. I am not going to publish a combined multiple: multiplying a vendor approximation by a price ratio and quoting it to three figures would be exactly the false precision this kind of report exists to avoid.

What I cannot measure, and will not estimate

The number everyone wants is "how much did delegation save." I cannot produce it honestly.

To compute a saving I would need to know what the same task would have cost had I done it inline, and I have no instrument for my own token consumption. Worse, the two paths are not equivalent in the direction people assume: a subagent starts cold and re-reads context the orchestrator already holds, so it can spend *more* tokens on the same task, not fewer. "Tokens at a cheaper rate" is therefore a comparison between a measured number and an imagined one.

What I can state is narrower and true: **doing this work through cheap agents cost \$2.08 at most**. Whether that beats the alternative is unmeasured.

Quality is the real question, and it splits three ways

Cost only matters if the output is usable. Six of the seven outputs were used: the three that found something, and the three that produced usable work without an independent find. One of those three usable outputs — the A/B rewrite described below — was used as a merge rather than wholesale, but it did reach the shipped document. The seventh, an accessibility review, is the only one whose recommendation I rejected outright.

Here is the breakdown that actually decides whether the routing earns its place:

OUTCOME OF EACH DELEGATION (N = 7)

Found something I had missed		3
Usable output, no independent find		3
Net negative — correct observation, rejected recommendation		1

The three finds are the case for delegation, and none of them were things I was going to catch:

- **A stale cached measurement.** A correctness review of a footer-positioning fix noticed that the footer's document position was measured at mount and refreshed only on `resize` — so a late-loading image or a swapped webfont would move the footer without the viewport ever changing size, aiming the correction at where the footer used to be. I had written that code and verified it. I had not thought about reflow.
- **A false positive in my own detector.** A review of a hook I had written — one that scans workflow scripts for unpinned agent calls — found that a regex literal containing `agentC` was scanned as a real call, because the masking covered strings and comments but not regexes.
- **A wrong premise in the task itself.** Reconnaissance on the corpus revealed that the site's blog directory is not indexed at all; the corpus reads a different tree entirely. I had been about to update the wrong files.

The one that cost me

An accessibility review of the same footer change reported that `prefers-reduced-motion` failed to suppress the new transform, exposing motion-sensitive users to repositioning on scroll.

I rejected it, and rejecting it took a full verification pass: reading three stylesheets to establish that the transform is untransitioned in all three places it appears — deliberately, so it cannot lag scroll — which makes it 1:1 scroll-linked positioning of the same class as `position: sticky`, not the animation that exemption targets. Suppressing it would have restored the original bug for precisely the users the exemption serves.

Verified under reduced-motion. Publishing that reasoning as settled would have been the same error this report exists to avoid, so I went and measured it. Driving the real page in Chrome with `prefers-reduced-motion: reduce` emulated, the transform is *not* suppressed: the chrome still lifts by the footer's full intrusion, 0 to -90px at 884×900 , identical to a session with no preference set. The agent's factual observation was correct. What the measurement also shows is that the movement is not animation: computed `transition-duration` for the transform is `0s` in both places, and stepping the last 200px of the document 20px at a time gives exactly -20px of counter-translation per 20px of scroll, starting only when the footer enters the viewport and stopping when it parks. No easing, no overshoot, no motion of its own.

So the observation was right and my refutation's premise holds. What sits between them is a judgement call: whether 1:1 scroll-linked repositioning is "motion" that a reduced-motion user should be spared. I read it as positioning rather than animation, in the same class as `position: sticky`, and I still think suppressing it would hurt the users it claims to protect. That reading is interpretive, not a settled WCAG boundary, and I am not going to present it as one.

That is why the outcome table calls this delegation a correct observation with a rejected recommendation rather than a false finding — the earlier label overstated what I had established. It was still net negative: the finding cost more to verify than it returned, and the verification work landed on the expensive model. At 1 in 7, that is affordable. It is not free.

The A/B: neither version won

For one task I did the work twice — a rewrite of a corpus document — once delegated to Sonnet and once myself, from an identical brief.

Neither was better. The delegated version chose two section headings that retrieve better than mine (`The workflow-inheritance gap` beats my prose-shaped alternative for a document that will be chunked and embedded) and worked a concept into the opening paragraph that I had forgotten. Mine chunked better, having split two new agent descriptions into their own subsection where the delegated version crammed them into single oversized bullets, and it was more precise on one identifier.

The shipped document is a merge, which is why this delegation counts as used in the table above even though none of it shipped unedited. Delegated cost for that draft: 23,746 tokens, three tool calls, 27 seconds. As a way to obtain a second complete draft to argue with, that is cheap. As a way to obtain a finished document, it was not — and the framing that treats delegation as "get the output back and use it" is the wrong model.

Where the routing rule breaks down

The rule I work under names bright-line triggers: a search across three or more files goes to a scout, the same edit repeated across three or more files goes to a mechanic. During this session I hit a case that satisfied the trigger and where following it would have been wrong — three one-line corrections across three corpus files, where writing a brief precise enough to delegate would have taken longer than making the edits.

There is a floor below which delegation costs more than it saves, and the current rule does not express it. The trigger should probably be a conjunction: three or more files **and** enough specification that the brief is shorter than the work.

Limits

- **n = 7.** One session, one orchestrator model, two repositories, one author. Nothing here is a rate; the 1-in-7 rejected-recommendation figure could as easily be 1-in-3 or 1-in-20.
- **One boundary in the accessibility case is interpretive,** not measured. The behaviour is measured; whether it constitutes a defect is my reading.
- **The counterfactual is absent by construction,** as described above. No saving is claimed.
- **Outcome grading is mine,** and I graded work I had commissioned. "Found something I had missed" is checkable against the commits it produced; "usable output" is a judgement call.
- **Costs are upper bounds,** charging input tokens at output rates.

A side finding that outweighs the report

Everything above measures the delegations I made deliberately. While measuring them I found a second code path where the pins had never applied at all. A workflow script fans work out with its own `agent()` calls, and a call that names no model does not get the pinned agent — it gets a generic worker that inherits the orchestrating session's model and effort. My session model is Opus at high effort, because that is what I keep the main thread on. Every unpinned fan-out had been running there.

Nothing announced it. It shows in one place only, in each agent's own metadata, where the type reads `workflow-subagent` instead of the agent's name — which is where I happened to look while gathering the per-delegation token counts for this report.

The bill dwarfs the one this report set out to measure. Across this repository's workflow subagents, orchestrator-tier calls account for **3,755,242 tokens** on the same input-plus-output basis used everywhere else here. At Opus 5 rates and the same upper-bound convention — every token charged at the \$25/M output rate — that is **\$93.88, upper bound**. The seven deliberate delegations this report is about cost \$2.08. The path I was not watching cost roughly forty-five times the one I was.

The repair is three things: the gap is written down in the repository README, two review-shaped agents give a fan-out something cheap to point at, and an opt-in hook reads a workflow script before it runs and names the calls that pin nothing.

What I changed as a result

Nothing about the tiering. The evidence supports it: three independent catches for \$2.08 is a good trade even before any cost comparison, and the one rejected recommendation was cheap to reject.

One thing did change beyond the side finding above. The pricing claim in the repository documentation is being corrected, because "Opus prices" now means a 5x gap and the old framing quietly overstates it.