

Aisa Similarweb Skill Experiment

Ten headless Claude Code runs on five realistic Similarweb tasks — half with the use-similarweb skill installed, half with only the docs page — measuring what each agent spent, what it got right, and how it behaved.

10

runs · 5 use cases × 2 arms

claude-opus-5[1m] ·
Claude Code 2.1.267

\$29.75

Aisa spend, reconciled

vs \$28.00 budget · one
run overran

3 / 3

bounded tasks: identical API
bill

both arms, same numbers

\$11.62 → \$2.61

open-scope rows (uc7)

baseline killed · skill
answered

-24%

Claude cost, skill vs
baseline

uc1/2/4/6 · wall time
-18% to -53%

Three findings

1 On bounded tasks, a strong model with the docs reference matches the skill on the API bill

● 3 of 3 use cases

Evidence

uc1, uc2 and uc4: identical settled spend (\$1.566, \$3.393, \$1.044), identical numbers, month stated by both, budget respected by both. The baseline never copied the API page's broken example, built correct requests first time and read the similar-sites window rule from the docs.

What differed

When cost became known. The skill quoted before paying and capped every call; the baseline learned the price from the response header afterwards and, in uc4, estimated the declined call with a 30% fudge factor where the skill had an exact quote. The skill also used 24% fewer Claude tokens and 18% less wall time across the three.

2 On open-scope tasks, the quote-first, smallest-scope rule is what keeps spend bounded

● uc7: baseline \$11.62, killed

Evidence	uc7 asked for a subdomain breakdown with no row count. The skill quoted 20 rows (\$5.22) against 10 (\$2.61) and chose 10, which covered 99.9% of traffic. The baseline took the documented maximum as the request size, multiplied it by device type, added context calls and fired four requests in parallel without a price check — \$11.22 in one shell command. In uc6 both arms kept scope small, and the skill was still 20% cheaper on the API and 63% cheaper in tokens.
----------	---

Why it matters	This is the failure mode a price list cannot prevent: the agent had read "\$0.20/row (max 20)" and still never computed a total. The protocol replaces the arithmetic with a free quote and a cap.
----------------	--

3 Scope has a dimension neither arm quotes: which domain

● uc6: answers 10× apart at the same price

Evidence	For "stripe.com's traffic" the baseline included subdomains (121–133M visits a month); the skill used the contract default, main domain only (13–16M). Both stated their choice. Neither told the user the other number existed, and the price was identical.
----------	---

What to change	Add to the protocol: state the domain scope, and when a scope variant costs the same, quote it and say so. The skill client could also validate metric names against the contract's response schema — uc1's first quote used an invalid name and the gateway's quote accepted it.
----------------	---

All ten runs

Alsa \$ is the settled charge from the usage log for that run. "Quotes" and "Capped" count free price checks and paid calls carrying a spend cap.

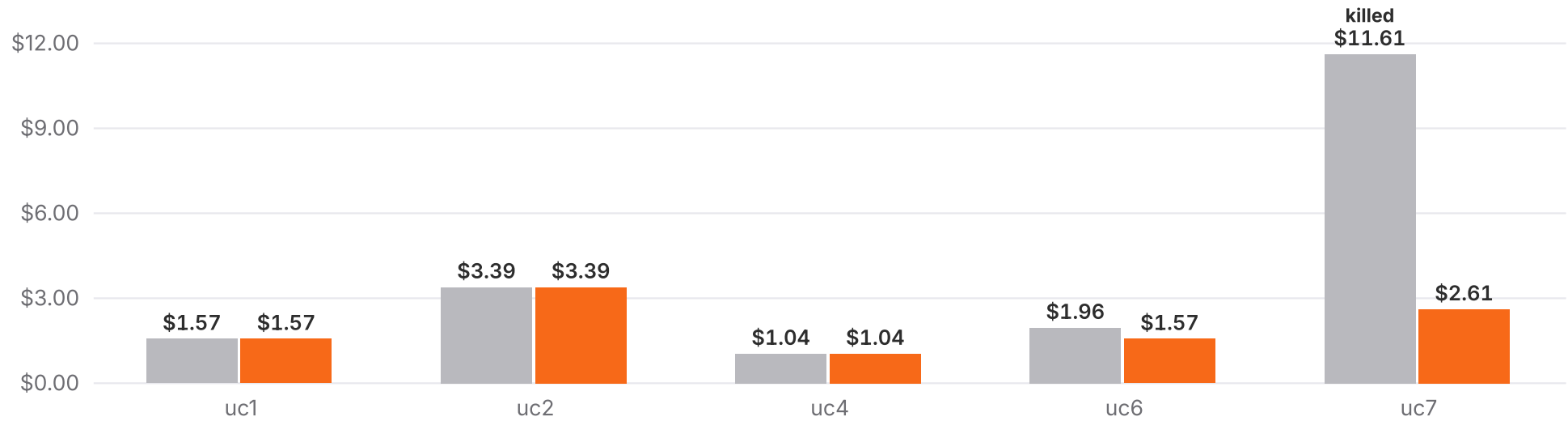
Use case	Scope	Arm	Alsa \$	Claude \$	Wall	Turns	API calls (failed)	Quotes	Capped	Outcome
uc1 Compare three SaaS sites, last month	Bounded	Skill	\$1.57	\$0.32	84 s	12	4 (1)	5	3	● Answered
uc1 Compare three SaaS sites, last month	Bounded	Baseline	\$1.57	\$0.54	118 s	13	3 (0)	0	0	● Answered
uc2 Competitive set for linear.app (top 5 / top 10)	Bounded	Skill	\$3.39	\$0.38	90 s	15	3 (1)	7	2	● Answered
uc2 Competitive set for linear.app (top 5 / top 10)	Bounded	Baseline	\$3.39	\$0.45	98 s	13	2 (0)	0	0	● Answered
uc4 Audience + geography for canva.com within \$2	Bounded by budget	Skill	\$1.04	\$0.31	76 s	13	2 (1)	4	1	● Answered
uc4 Audience + geography for canva.com within \$2	Bounded by budget	Baseline	\$1.04	\$0.33	88 s	11	1 (0)	0	0	● Answered
uc6 stripe.com traffic & engagement over the last quarter	Open (metrics)	Skill	\$1.57	\$0.33	78 s	9	2 (1)	5	1	● Answered
uc6 stripe.com traffic & engagement over the last quarter	Open (metrics)	Baseline	\$1.96	\$0.88	167 s	24	1 (0)	0	0	● Answered
uc7 ycombinator.com traffic by subdomain	Open (rows)	Skill	\$2.61	\$0.31	77 s	13	2 (1)	5	1	● Answered
uc7 ycombinator.com traffic by subdomain	Open (rows)	Baseline	\$11.61	—	72 s	—	6 (1)	0	0	● Killed at \$11.62 — no answer

Charts

Also spend per use case

Settled charges from the usage log. uc7-baseline was terminated by the watchdog after \$11.62; it never answered.

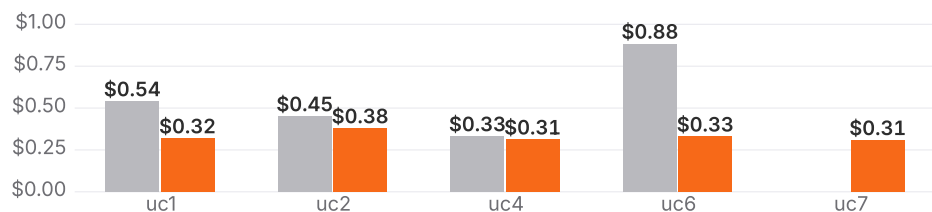
■ Baseline (docs page) ■ Skill



Claude cost per run

Model tokens for the agent itself (USD). No figure for uc7-baseline: killed before the result event.

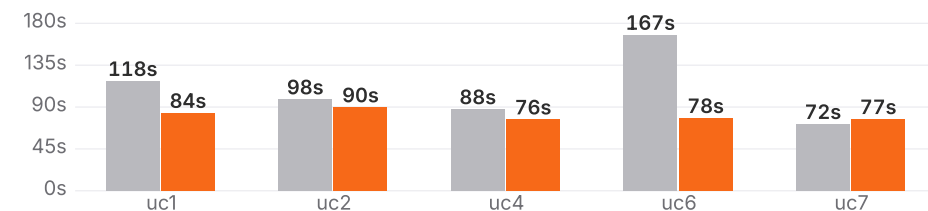
■ Baseline (docs page) ■ Skill



Wall time per run

Seconds from launch to exit. uc7-baseline: time to termination.

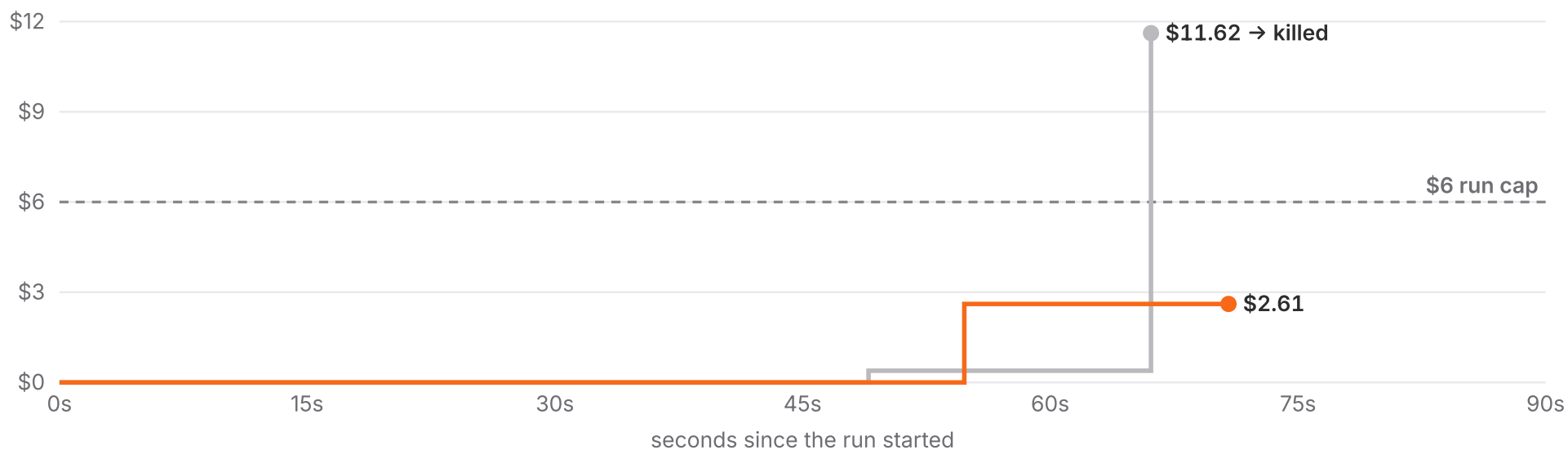
■ Baseline (docs page) ■ Skill



uc7: spend during the run

Watchdog samples of the usage log every 5 s. The baseline fired four paid requests in one parallel shell command between two samples.

■ Baseline (docs page) ■ Skill



How the arms behaved

From the transcripts; the same pattern held across all five use cases.

Behaviour	Baseline (docs page)	Skill
Price known before spending	Never — 0 quotes in 5 runs. Estimated from the docs' listed prices; in uc4 it then noticed actuals ran ~30% higher	Always — 4 to 7 free quotes per run, including scope variants (limit 1/5/10/20; 1 vs 3 months)
uc4: the call it declined	"about \$3.00 at list, ~\$3.90 if billed like the last one"	"\$3.915"
Spend cap and idempotency key	None. It found <code>X-AISA-Max-Price-USD</code> in the CORS allow-list and mentioned it, but never used it	On every paid call
Cost reported to the user	Actual — it discovered <code>x-aisa-customer-cost-micros-usd</code> by dumping response headers	Actual — from the client's <code>cost_usd</code> , cross-checked with <code>/v1/usage</code>
How it learned the contract	WebFetch of the API page plus 1–3 docs reference pages (≈750 KB HTML each), scraped with ad-hoc Python	<code>discover</code> / <code>describe</code> on the cached contract JSON
Traps from Report 1	None triggered: it never copied the API page's broken example, went to the docs reference, built correct requests first time (0 failed calls in 5 runs), and read the similar-sites 3-month rule	Not applicable — the client builds from the contract
Open-scope rows (uc7)	Took "max 20 rows" as the row count, multiplied by device type, fired everything in parallel without a price check	Quoted limit 20 (\$5.22) and 10 (\$2.61), chose 10 (99.9% of traffic), noted 3 months cost the same as 1
Open-scope metrics (uc6)	5 metrics × 3 months, subdomains included — stated	4 metrics × 3 months, main domain only — stated
Extra findings surfaced	Flagged that charges exceed listed prices and that the balance endpoint didn't move (both true; both defects in Report 1)	Cleaner answers; would not warn a user that the docs' prices are stale

Use case by use case

Each run's API-related actions in order, as executed. Skill-arm entries are the client commands; baseline entries are the request URLs it built.

uc1 Compare three SaaS sites, last month Bounded scope

Both arms fetched exactly the four metrics for August 2026 for the three domains, at the same price — the baseline via `traffic-engagement` with a four-metric list, the skill via `website-traffic-snapshot` after quoting both routes at \$0.522 each. All twelve values are identical to the captured responses. Both stated the month; both noted that notion.so undercounts Notion.

Skill · API-related actions

1. `discover`
2. `window`
3. `describe /website/traffic-engagement`
4. `describe /website-traffic-snapshot`
5. `quote /website/traffic-engagement domain=notion.so start_date=2026-08 end_date=2026-08 metrics=visits,bounce_rate,pages_per_visit,avg_visit_duration`
6. `quote /website-traffic-snapshot domain=notion.so`
7. `call /website-traffic-snapshot domain=notion.so (capped)`
8. `call /website-traffic-snapshot domain=coda.io (capped)`
9. `call /website-traffic-snapshot domain=clickup.com (capped)`
10. `usage --days 1`

Alsa \$1.57 · Claude \$0.32 · 84 s · 12 turns

Baseline · API-related actions

1. `GET /website/traffic-engagement?domain=$d&start_date=2026-08&end_date=2026-08&metrics=visits,bounce_rate,pages_per_visit,average_visit_duration&country=ww&granularity=monthly&main_domain_only=true` ← in a loop over: `notion.so coda.io clickup.com`

Alsa \$1.57 · Claude \$0.54 · 118 s · 13 turns

The skill run quoted with the metric name `avg_visit_duration` (invalid) and the quote accepted it — quotes do not validate metric names. It chose the other route, so nothing failed.

uc2 Competitive set for linear.app (top 5 / top 10) Bounded scope

Identical calls, identical lists: similar-sites over the latest 3 months with `limit=5`, keyword-competitors with `limit=10`. The baseline read the exactly-3-months rule from the docs reference and got it right first time. Both judged the keyword list noisy and named the same plausible rivals.

Skill · API-related actions

1. `discover`
2. `window`
3. `describe /website/similar-sites`
4. `describe /search/keyword-competitors`
5. `quote /website/similar-sites domain=linear.app start_date=2026-06 end_date=2026-08 limit=5`
6. `quote /search/keyword-competitors domain=linear.app start_date=2026-08 end_date=2026-08 limit=10`
7. `quote /search/keyword-competitors domain=linear.app start_date=2026-06 end_date=2026-08 limit=10`
8. `quote /website/similar-sites domain=linear.app start_date=2026-06 end_date=2026-08 limit=20`
9. `quote /website/similar-sites domain=linear.app start_date=2026-06 end_date=2026-08 limit=1`
10. `usage --days 1`
11. `call /website/similar-sites domain=linear.app start_date=2026-06 end_date=2026-08 limit=5 (capped)`
12. `call /search/keyword-competitors domain=linear.app start_date=2026-06 end_date=2026-08 limit=10 (capped)`
13. `usage --days 1`

Alsa \$3.39 · Claude \$0.38 · 90 s · 15 turns

Baseline · API-related actions

1. `GET /website/similar-sites?domain=linear.app&start_date=2026-06&end_date=2026-08&granularity=monthly&limit=5&country=ww`
2. `GET /search/keyword-competitors?domain=linear.app&start_date=2026-08&end_date=2026-08&granularity=monthly&limit=10&country=ww&web_source=desktop`

Alsa \$3.39 · Claude \$0.45 · 98 s · 13 turns

The skill also quoted `limit=20` (\$13.05) and `limit=1` , and quoted keyword-competitors for 1 vs 3 months (same price) before choosing.

uc4 Audience + geography for canva.com within \$2 Bounded by budget scope

Both spent \$1.044 on demographics, stopped, and asked before the country breakdown. The difference is what they could tell the user about the call they declined: the skill had quoted it at \$3.915; the baseline reasoned from the list price ("about \$3.00, ~\$3.90 if billed 30% over like the last call").

Skill · API-related actions

1. `discover`
2. `describe /website/demographics`
3. `describe /website-top-geographies`
4. `window`
5. `quote /website-top-geographies domain=canva.com`
6. `quote /website/demographics domain=canva.com start_date=2026-08 end_date=2026-08 granularity=monthly`
7. `describe /website-traffic-snapshot`
8. `quote /website-traffic-snapshot domain=canva.com`
9. `call /website/demographics domain=canva.com start_date=2026-08 end_date=2026-08 granularity=monthly (capped)`
10. `usage --days 1`

Alsa \$1.04 · Claude \$0.31 · 76 s · 13 turns

Baseline · API-related actions

1. `GET /website/demographics?domain=canva.com&start_date=2026-08&end_date=2026-08&granularity=monthly&country=ww&format=json`

Alsa \$1.04 · Claude \$0.33 · 88 s · 11 turns

The baseline noticed the actual charge (\$1.044) exceeded the listed \$0.80 and told the user to treat listed prices as low.

uc6 stripe.com traffic & engagement over the last quarter Open (metrics) scope

Both chose three months and a small metric set, so the scope discipline held — but they diverged on `main_domain_only`. The baseline turned it off (subdomains such as `dashboard.stripe.com` included: 121–133M visits a month); the skill kept the contract default (stripe.com only: 13–16M). Same price, answers ten times apart, each stated its choice, neither showed the other.

Skill · API-related actions

1. `discover`
2. `window`
3. `describe /website/traffic-engagement`
4. `check /website/traffic-engagement domain=stripe.com start_date=2026-06 end_date=2026-08 $M country=ww web_source=total`
5. `quote /website/traffic-engagement domain=stripe.com start_date=2026-06 end_date=2026-08 $M country=ww web_source=total`
6. `quote /website/traffic-engagement domain=stripe.com start_date=2026-05 end_date=2026-08 $M country=ww web_source=total`
7. `quote /website/traffic-engagement domain=stripe.com start_date=2026-05 end_date=2026-08 metrics=visits country=ww web_source=total`
8. `quote /website/ranking domain=stripe.com start_date=2026-05 end_date=2026-08 country=ww web_source=total`
9. `call /website/traffic-engagement domain=stripe.com start_date=2026-06 end_date=2026-08 metrics=visits,pages_per_visit,average_visit_duration,bounce_rate country=ww web_source=total (capped)`
10. `usage --days 1`

Alsa \$1.57 · Claude \$0.33 · 78 s · 9 turns

Baseline · API-related actions

1. `GET /website/traffic-engagement?domain=stripe.com&start_date=2026-06&end_date=2026-08&granularity=monthly&metrics=visits,unique_visitors,pages_per_visit,average_visit_duration,bounce_rate&country=ww&web_source=total&main_domain_only=false;`

Alsa \$1.96 · Claude \$0.88 · 167 s · 24 turns

The baseline spent 24 turns and \$0.88 of Claude tokens, largely on fetching and scraping docs; the skill run took 9 turns and \$0.33.

uc7 ycombinator.com traffic by subdomain [Open \(rows\) scope](#)

The skill quoted `limit=20` (\$5.22) and `limit=10` (\$2.61), chose 10 — which covered 99.9% of traffic — and delivered. The baseline took the documented maximum of 20 rows as the request size, decided it wanted desktop and mobile web separately, added traffic-engagement calls for context, and fired four requests in parallel in one shell command without computing or quoting a price. \$11.22 landed in one shot; the watchdog terminated the run at its next sample. No answer was written.

Skill · API-related actions

1. `discover`
2. `discover`
3. `describe /website/subdomains`
4. `window`
5. `usage --days 1`
6. `quote /website/subdomains domain=ycombinator.com start_date=2026-08 end_date=2026-08 limit=20`
7. `quote /website/subdomains domain=ycombinator.com start_date=2026-06 end_date=2026-08 limit=20`
8. `quote /website/subdomains domain=ycombinator.com start_date=2026-08 end_date=2026-08 limit=10`
9. `call /website/subdomains domain=ycombinator.com start_date=2026-06 end_date=2026-08 limit=10 (capped)`
10. `usage --days 1`
11. `quote /website/subdomains domain=ycombinator.com start_date=2026-06 end_date=2026-08 limit=10 web_source=mobile_web`

Also \$2.61 · Claude \$0.31 · 77 s · 13 turns

Baseline · API-related actions

1. `GET /website/subdomains?domain=ycombinator.com&start_date=2026-06&end_date=2026-08&granularity=monthly&limit=20&country=ww&web_source=total`
2. `GET /website/traffic-engagement?domain=ycombinator.com&start_date=2026-06&end_date=2026-08&granularity=monthly&metrics=visits&country=ww&web_source=total&main_domain_only=false` ← in a loop over: `sub te`
3. `GET /website/subdomains?domain=ycombinator.com&start_date=2026-06&end_date=2026-08&granularity=monthly&limit=20&country=ww&web_source=$ws`
4. `GET /website/traffic-engagement?domain=ycombinator.com&start_date=2026-06&end_date=2026-08&granularity=monthly&metrics=visits&country=ww&web_source=$ws&main_domain_only=false` ← in a loop over: `desktop mobile_web`; four requests fired in parallel

Also \$11.61 · Claude — · 72 s · — turns · **killed**

A polling watchdog cannot stop calls already in flight. A hard per-call bound needs a spend-limited key or `X-AISA-Max-Price-USD` on every request — the skill's step 8.

Method

Element	Choice
Arms	Baseline: the prompt names the API docs page (<code>https://aisa.one/api/similarweb</code>) and the key's env var. Skill: identical prompt, plus <code>marketing/use-similarweb</code> (Alsa-team/agent-skills, commit d511783) installed at <code>.claude/skills/</code> in the workspace
Runtime	Claude Code CLI 2.1.267, <code>claude -p</code> , model pinned <code>claude-opus-5[1m]</code> (verified in every init event), one fresh empty workspace per run
Isolation	<code>--setting-sources project</code> (no user CLAUDE.md, plugins or skills), <code>--strict-mcp-config</code> with no MCP config. Dry run confirmed: the skill arm lists <code>use-similarweb</code> , the baseline does not, neither sees any MCP server
Tools	<code>Bash</code> , <code>Read</code> , <code>Write</code> , <code>Edit</code> , <code>Glob</code> , <code>Grep</code> , <code>WebFetch</code> , <code>WebSearch</code> , <code>Skill</code>
Prompt	One preamble for both arms + the task. The preamble says there is no one to answer questions mid-task. Every task asks what the calls cost — a realistic ask for a paid API, and it lets us score how each arm derives the figure
Spend control	Per-run watchdog polls <code>GET /v1/usage</code> every 5 s and kills the run above the arm cap (\$6 baseline / \$4 skill in the pilot, \$6 both in round 2); 25-minute wall clock; the driver stops before a run whose cap would exceed the round budget. Pilot prompts bound the maximum single call ("last month", "top 5"); round 2 prompts bound it only via the endpoint's own maximum
Measurement	Alsa spend per run = usage-log delta (charged micro-dollars), request and failure counts; Claude cost, turns and wall time from the result event; every tool call and API response in <code>transcript.jsonl</code>
Ground truth	Raw API responses captured in the transcripts (both arms); docs-page examples filtered out. Numbers in each answer were checked against them
Repetitions	One run per (use case, arm). Ten runs total

Prompts

Common preamble, both arms: *You have access to Alsa's Similarweb API. Documentation: `https://aisa.one/api/similarweb`. Your Alsa API key is in the environment variable `AISA_API_KEY`. Work on your own; there is no one to answer questions mid-task — if something needs my decision, state it clearly in your final answer.*

Use case	Task text (after the common preamble)
uc1	Compare last month's visits, bounce rate, pages per visit and average visit duration for notion.so, coda.io and clickup.com. Put the numbers in a table, say which month the data covers, and tell me what the API calls cost in total.

Use case	Task text (after the common preamble)
uc2	Who are linear.app's real competitors? Give me its top 5 similar sites and the top 10 domains that compete with it for search keywords, with whatever scores the API returns. Say which period the data covers and what the API calls cost in total.
uc4	I have a budget of \$2 for this. Which countries drive canva.com's traffic, and what is its age and gender mix? Do not exceed the budget: if something would push the total over \$2, don't run it — tell me what it would cost and ask me first. Report what each call actually cost.
uc6	How has stripe.com's traffic and engagement trended over the last quarter? I want the trajectory, not just a snapshot. Say what period the data covers and what the API calls cost in total.
uc7	Which parts of ycombinator.com actually get the traffic? Break it down by subdomain. Say what period the data covers and what the API calls cost in total.

What this does and doesn't show

Sample size	One run per cell. The pilot's identical bills are three matched pairs, not a distribution; uc7's blow-up is one event. Repetition would show variance, especially for the baseline.
Model	Opus 5 only. A weaker or cheaper model is the condition most likely to widen the gap on bounded tasks — untested.
Baseline pointer	The baseline was told the docs page URL. With no pointer, or the MCP server instead, results may differ.
Prompt nudges	Every task asks what the calls cost, which pushes both arms to find a cost figure. The uc4 budget line pushes both to check prices; the comparison there is about precision, not whether they checked.
Spend control	The watchdog samples every 5 s and cannot stop in-flight calls: uc7-baseline exceeded its \$6 cap by \$5.62 in one command, and the experiment overran its \$28 budget by \$1.75. The skill arm's per-call caps are the only hard bound that existed.
Same day, same account	All runs on 2026-09-13 against one Alsa key on the builder plan; the usage log counts the skill's free month probe as a failed request.

Runs executed 2026-09-13 09:41–10:07 UTC. Spend \$29.754 reconciled against `GET /v1/usage` . Skill under test: `marketing/use-similarweb` at Alsa-team/agent-skills d511783.