Where the $32,341 of Part 2 actually went

Financial detail companion to Part 2 of the SHORA trilogy. Not part of the trilogy itself. For readers curious about the line-by-line decomposition of the AWS spend that produced the 9% navigation rate on 10.5 million product pages.

Source: AWS Cost Explorer queries against the SHORA production AWS account.
Query date: 2026-05-14.
Window: 2025-08-01 → 2026-05-14 (the full duration of the experiment, from architecture probes through wind-down).

Headline numbers

Metric	Value
AWS spend	**$32,341.37**
Calendar window	Aug 2025 → May 2026 (10 months; 7 of intense burn)
Peak month	February 2026 — $20,518.40
Peak day	2026-02-12 — $5,407.37 ($5,163.47 on Claude Opus 4.6 alone)
Primary inference model	Claude Opus 4.6 — $15,589.38 (48.2% of total)
Primary region	eu-west-3 (Paris) — $22,538.41 (69.7%)

> **The LLM consumed 65.8% of the budget.** Claude Opus 4.6 + Opus 4.5 + Sonnet 4.5 + Haiku 4.5 + 3.5 Haiku + Opus 4 + 3 Sonnet across all Bedrock Claude models sum to $21,289.92 of the $32,341.37 total. The architectural argument in Part 2 — that the cost of an LLM-agent measurement layer is dominated by the language model itself — is reflected directly in the bill.

1. Monthly burn curve

The spend curve has a shape that explains how the experiment moved from probe to peak to admission. February 2026 is the inflection point: it carries 64% of the total spend in a single month, after which the curve falls toward zero.

Month	AWS spend
2025-08	$37.32
2025-09	$25.87
2025-10	$480.54
2025-11	$791.10
2025-12	$584.88
2026-01	$4,643.30
2026-02	$20,518.40
2026-03	$3,825.63
2026-04	$1,418.87
2026-05	$15.46
TOTAL	$32,341.37

2. Service breakdown — full window

Service	Cost	Share
Claude Opus 4.6 (Amazon Bedrock)	$15,589.38	48.2%
Claude Opus 4.5 (Amazon Bedrock)	$5,220.53	16.1%
EC2 — compute (the journey + extraction fleet)	$3,832.46	11.9%
Amazon Bedrock (baseline / throughput)	$2,163.34	6.7%
EC2 — networking, EBS, NAT	$1,985.20	6.1%
Amazon DynamoDB	$1,418.00	4.4%
Amazon RDS Postgres (validator + audit)	$1,242.14	3.8%
Other Bedrock models (Sonnet 4.5, Haiku 4.5, Opus 4, etc.)	$479.91	1.5%
Amazon S3, VPC, WAF, KMS, CloudWatch, ECR, Route 53, Secrets	$410.41	1.3%
TOTAL	$32,341.37	100.0%

The two-thirds-on-the-LLM headline

Layer	Cost	Share
Claude inference (all models)	$21,289.92	65.8%
Bedrock orchestration (baseline / throughput)	$2,163.34	6.7%
Compute fleet (EC2 + EC2-Other)	$5,817.66	18.0%
Data + storage (DDB + RDS + S3)	$2,826.25	8.7%
Networking, security, observability, registry	$244.20	0.8%

3. Region breakdown

~70% of AWS spend stayed in eu-west-3 (Paris), and we used Bedrock Claude's EU public endpoints. The us-east-1 share was driven by experimental runs against Bedrock models available only in US regions at the time (Moonshot Kimi K2.5, etc).

Region	Cost	Share
eu-west-3 (Paris)	**$22,538.41**	**69.7%**
us-east-1 (N. Virginia)	$8,837.93	27.3%
us-east-2 (Ohio)	$906.11	2.8%
global (Route 53, KMS, etc.)	$58.92	0.2%

4. February 2026 daily curve — the peak month

The day-by-day shape of the experiment hitting its ceiling. Three phases visible:

- **Feb 1–11 — ramp-up.** Extraction fleet warming. Opus 4.5 dominant from January carry-over. Daily cost climbs from $144 to $1,258.
- **Feb 12 — peak day.** $5,407.37 in 24 hours, of which $5,163.47 (95.5%) on Claude Opus 4.6 alone. The day Claude Opus 4.6 was released at twice the price while the journey fleet was full throttle: 20-40 concurrent Camoufox browsers running the extract-then-journey loop against the 1,056-retailer websites.
- **Feb 12–20 — sustained-burn plateau.** $446 – $2,478 per day while the team confirmed the 9% navigation rate against the golden set.
- **Feb 21–28 — wind-down.** Under $300/day by Feb 22. The decision to retire the agent loop.

Day	Total	Opus 4.6	Opus 4.5	EC2	Bedrock	DDB
2026-02-01	$144.54	$0.00	$119.92	$5.64	$0.00	$1.19
2026-02-02	$226.12	$0.00	$198.41	$5.64	$0.00	$1.55
2026-02-03	$318.89	$0.00	$290.17	$5.64	$0.00	$2.73
2026-02-04	$293.60	$0.00	$265.08	$5.62	$0.00	$2.73
2026-02-05	$195.07	$4.23	$163.45	$5.64	$0.00	$1.07
2026-02-06	$295.67	$97.54	$141.56	$27.93	$0.00	$1.24
2026-02-07	$408.68	$206.63	$129.06	$44.43	$0.00	$0.94
2026-02-08	$284.53	$181.24	$28.37	$44.43	$0.00	$4.10
2026-02-09	$175.37	$58.64	$22.58	$61.39	$0.00	$4.48
2026-02-10	$1,258.04	$1,020.67	$112.56	$82.70	$0.00	$8.33
2026-02-11	$1,258.13	$1,089.73	$26.23	$82.52	$25.73	$2.77
2026-02-12	$5,407.37	$5,163.47	$9.52	$83.45	$116.00	$4.05
2026-02-13	$446.76	$166.01	$20.82	$84.35	$140.25	$2.58
2026-02-14	$525.93	$225.07	$10.17	$84.35	$117.69	$46.23
2026-02-15	$1,019.83	$595.05	$20.03	$84.35	$167.94	$115.06
2026-02-16	$2,084.72	$1,569.84	$113.47	$84.35	$128.81	$152.74
2026-02-17	$2,478.44	$2,013.50	$47.17	$84.35	$161.35	$135.25
2026-02-18	$1,076.66	$596.75	$5.05	$84.35	$190.77	$166.79
2026-02-19	$693.41	$217.56	$0.44	$84.35	$191.42	$166.53
2026-02-20	$540.92	$209.77	$3.56	$57.64	$128.05	$109.39
2026-02-21	$410.98	$260.55	$5.11	$20.30	$77.13	$16.96
2026-02-22	$200.59	$112.55	$0.00	$6.78	$48.55	$2.21
2026-02-23	$104.22	$36.99	$0.00	$6.78	$27.81	$2.21
2026-02-24	$128.53	$89.18	$0.00	$6.78	$0.00	$2.21
2026-02-25	$81.40	$42.03	$0.00	$6.78	$0.00	$2.21
2026-02-26	$83.73	$44.33	$0.00	$6.77	$0.00	$2.21
2026-02-27	$266.93	$210.00	$0.00	$6.73	$16.93	$2.51
2026-02-28	$109.37	$49.64	$0.00	$6.78	$19.97	$2.21

5. Bedrock token economics

The most informative slice for a technical reader: how the Bedrock bill decomposes into input vs output vs cache-write vs cache-read tokens. The shape confirms that the agent's prompt-caching strategy was working — 52.7% of total Bedrock spend was on cache writes + cache reads, meaning the orchestrator was correctly re-using context across iterations of the extract-then-journey loop rather than re-paying for it on every call.

> **Output tokens were only 12.1% of Bedrock spend.** The cost driver was *reading* pages (input tokens + cache), not *writing* answers. This is the cost signature of a vision-and-context-heavy extractor, not a chatty agent. The intuitive optimization ("reduce output verbosity") would have moved the bill by single digits at most. The structural cost was the volume of context the model had to ingest to answer at all.

Token category	Cost	Share of Bedrock
Input tokens (uncached, all regions/models)	$8,247.84	35.2%
Cache-write tokens (all variants)	$7,326.65	31.2%
Cache-read tokens (all variants)	$5,048.80	21.5%
Output tokens (all regions/models)	$2,829.89	12.1%

6. Unit economics

Metric	Value
Total AWS spend / total product pages instrumented	$32,341 / 10.5M ≈ **$0.0031 per product page studied**
Total AWS spend / total purchase journeys attempted	$32,341 / 10.5M ≈ **$0.0031 per journey attempted** (one journey per product)
Total AWS spend / successfully-navigated journeys	$32,341 / (10.5M × 9%) ≈ **$0.034 per successfully-navigated journey**
Claude Opus 4.6 inference / product page	$15,589 / 10.5M ≈ **$0.00149 per product page**

> **The economic killer is the 9% navigation rate.** The per-attempt cost of $0.0031 is competitive on the surface, but the per-*successful*-journey cost ($0.034) is what the buyer's accountant compares against alternatives. At 80% navigation rate (the bar SHORA set going in), the cost per successful journey would have been $0.0039 — competitive with BPO and an order of magnitude below the enterprise pricing of incumbents in adjacent

> categories. At 9%, the cost per successful journey is $0.034, which is an order of magnitude worse than the bar.

7. What this evidence supports — and does not support

Supports

- **The LLM bill is the bill.** $20,810 across all Claude models (Opus 4.6 + Opus 4.5 + Sonnet 4.5 + Haiku 4.5 + 3.5 Haiku + Opus 4 + 3 Sonnet) = 64.4% of the AWS spend. The architectural argument in Part 2 — that an LLM-agent measurement layer is dominated by the cost of the language model itself — is reflected directly in the breakdown.
- **The infrastructure layer was production-grade.** ~70% of AWS spend stayed in the EU region for data-residency compliance, prompt-caching working as designed (52.7% of Bedrock spend on cache reads + writes), the journey-orchestrator and extraction-orchestrator fleets ran reliably for seven months without major operational incident.
- **The pivot decision is dated.** Feb 12 peak day, Feb 22 ramp-down. No revisionism.

Does not support

- A claim that the LLM agent failed because of model selection. Multiple Claude models were tried; the smaller ones (Sonnet 4.5, Haiku 4.5, 3.5 Haiku) contributed $392.55 combined and did not clear the bar either.
- A claim that the LLM agent failed because of insufficient budget. The peak day spent $5,407 — at that rate, scaling 10× would have cost ~$320K and the structural ceiling would still have been visible at the same 9%.
- A claim that the LLM agent failed because of bot defenses. The Camoufox bypass rate was ~92%. The 91% navigation failure is application-layer, as documented in the technical companion and in Part 2 §"Why the architecture hit a ceiling".

Part 2 of the SHORA trilogy: $32,341 of AWS spend, 10.5 million product pages, 9% reliability. Technical companion: part-2-technical-companion.
