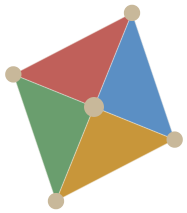


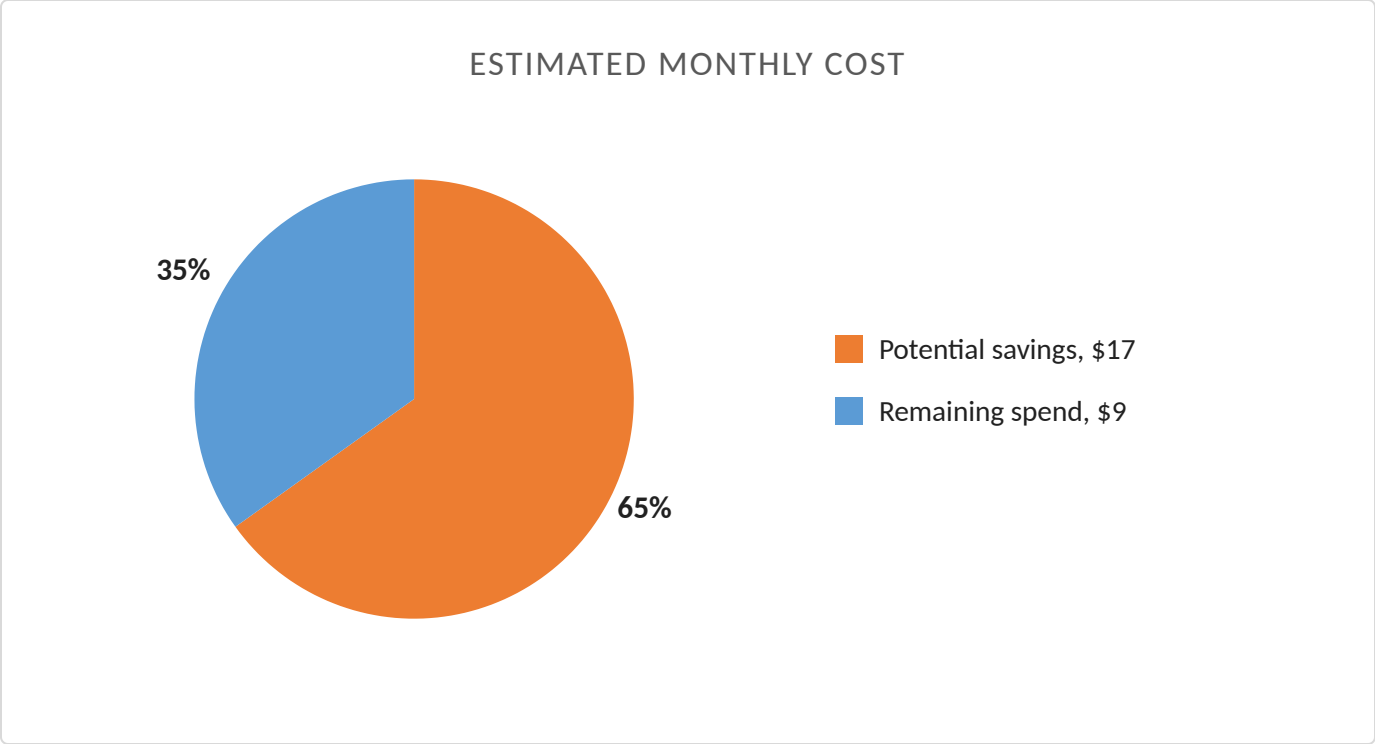
Cost Leak Report

Sample report, generated from demo usage data

Prepared by an independent engineer. TraceMinder turns a sanitized usage export into a ranked cost leak report. Your file is used only to generate this report and is deleted after delivery.

Executive Summary

The TraceMinder cost leak audit analyzed **426 logged API calls** from your usage export, covering roughly 0.9 day(s) and projected to a 30-day month. It found potential monthly savings of **$16.60** out of **$25.49** projected monthly spend, a **65% reduction**, by removing five common cost-leak patterns. The chart below shows the share of spend that is recoverable and how the recoverable amount breaks down by leak type.



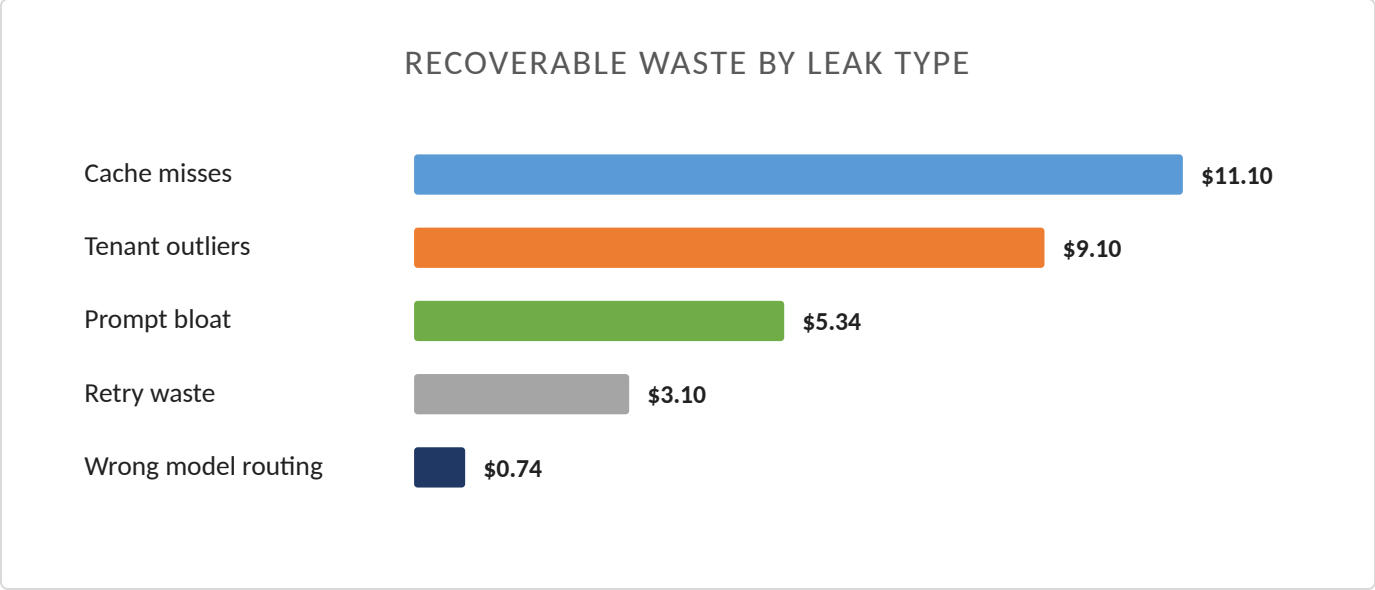
Every figure comes from sanitized usage metadata only. No prompt or completion text was read. Each billed call is attributed to a single leak, so the headline figure is never inflated.

Cost Leak Summary

The findings below are ranked by estimated monthly waste.

Leak	Monthly waste	Share of spend	Confidence
Cache misses	$11.10	43.5%	medium
Tenant outliers	$9.10	35.7%	high
Prompt bloat	$5.34	21.0%	medium
Retry waste	$3.10	12.2%	high
Wrong model routing	$0.74	2.9%	medium-high
Recoverable (de-duplicated)	**$16.60**	**65%**	

Per-finding figures are standalone and can overlap, so they add up to more than the de-duplicated recoverable total on the last row.



Findings

Cache misses

Finding

Identical requests repeat but are never served from cache, so each repeat is paid for.

Evidence

/answer: 66 calls share only 4 input sizes, cache-hit rate 0%.

Estimated monthly waste

$11.10 per month.

Likely cause

No response or prompt caching for repeated identical requests.

Recommended fix

Add caching keyed on the normalized request; enable provider prompt caching; set sensible TTLs.

Confidence

Medium.

Tenant outliers

Finding

A small number of tenants drive a disproportionate share of spend.

Evidence

822b33ad = 55% of spend ($0.41); median tenant $0.10.

Estimated monthly waste

$9.10 per month.

Likely cause

One heavy workload or customer; possibly mispriced or running the inefficient patterns above at scale.

Recommended fix

Review limits/pricing for these tenants and apply the fixes above to their traffic first; add per-tenant budget alerts.

Confidence

High.

Prompt bloat

Finding

A large, near-constant prompt prefix is sent on every call to some endpoints.

Evidence

/answer: 66 calls avg 1315 in / 249 out (floor 1314); /chat: 28 calls avg 1365 in / 173 out (floor 1336).

Estimated monthly waste

$5.34 per month.

Likely cause

Full knowledge base or verbose system prompt resent every request; no prompt caching or context trimming.

Recommended fix

Trim the system prompt, move static context to provider prompt caching, retrieve only relevant chunks.

Confidence

Medium.

Retry waste

Finding

Calls are billed multiple times because failed or rejected responses are retried.

Evidence

108 billed retry attempts (retry_count > 0); 32 rate-limited/error attempts. Most retry spend is on /extract.

Estimated monthly waste

$3.10 per month.

Likely cause

App-level retries on strict validation or transient errors, without idempotency or backoff.

Recommended fix

Cap retries, add exponential backoff, fix the validation that rejects good responses, alert on retry spikes.

Confidence

High.

Wrong model routing

Finding

Trivial tasks run on an expensive model where a smaller model is typically sufficient.

Evidence

/classify: 110 calls on gpt-4o producing only 6 output tokens.

Estimated monthly waste

$0.74 per month.

Likely cause

Everything routed to a flagship model by default, regardless of task difficulty.

Recommended fix

Route classification and short tasks to a mini model behind a quality check; add a model-selection policy.

Confidence

Medium-high.

Method and Assumptions

- Analysis uses only sanitized usage metadata. No prompt or completion text is read.
- Prompt bloat assumes 50% of a static input floor (600 tokens or more) is reducible.
- Cache savings assume a hit avoids 90% of a repeated call's cost.
- Routing savings recompute trivial-task calls at the cheaper model's published rate.
- Per-finding numbers are standalone and may overlap; the de-duplicated total does not.