

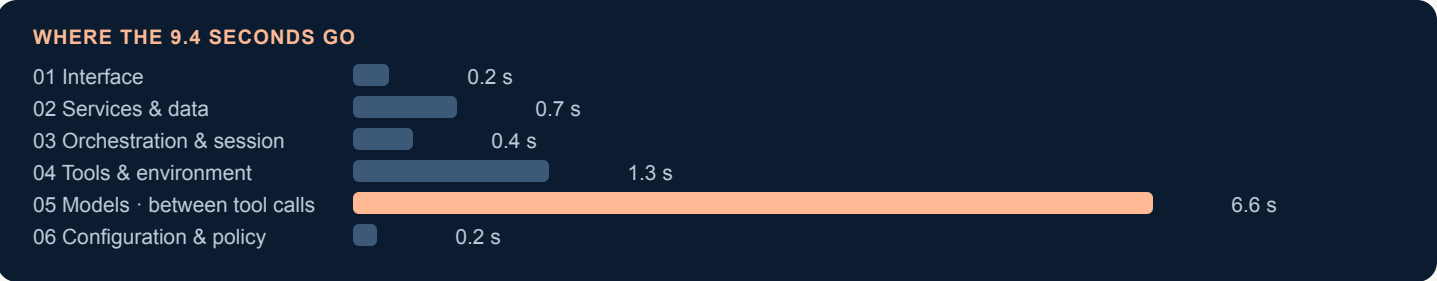
SAMPLE DELIVERABLE · SYNTHETIC DATA · HARNESS REVIEW

Harness Scorecard

Support-ticket triage agent, synthetic workload. Fifty cases, scored on the current model and one candidate. Quality counts every attempt: accepted results divided by all attempts, retries included. The fixes are ranked by impact, each with its evidence and owner.

Measure	Current model	Candidate model	What it means
Quality	50 cases · 53 attempts · 41 accepted (77%)	50 cases · 51 attempts · 45 accepted (88%)	Gains on long tickets; two refund cases still fail.
Speed	Median 9.4 s to a useful result	Median 7.1 s	Most of the wait sits between tool calls, not in the application code.
Cost	Index 100 per accepted task	Index 84	Longer answers offset part of the lower token price.
Control	0 unauthorized actions observed · 1 failed control check	0 unauthorized actions observed · 1 failed control check	The refund tool skips the approval step when called directly, on both models. Refunds are blocked until it is fixed.
Drift	Baseline recorded	4 format changes on edge cases	Gate added for the reply template.

HARNESS MAP · ONE TICKET, TIMED BY LAYER



CONTROL FINDING

Refund tool skips the approval step when called directly

In the control checks, the agent asks for approval above the refund limit, but the tool itself accepts the larger amount when called directly. No unauthorized refund was observed. The gate blocks the refund action until the limit is enforced at the tool boundary and the denied case is added. Owner: payments engineering.

RANKED FIX PLAN

1. Enforce the refund limit in the tool.

Control. Owner: payments.
2. Cut the model's thinking between tool calls.

Speed. Owner: platform.
3. Adopt the candidate behind the gate.

Quality and cost. Owner: AI team.
4. Delete the summarization pre-pass.

The candidate no longer needs it.

Scope: a synthetic workload built to show the format of a Harness Review scorecard. The ticket data, the models and the results are illustrative and do not describe a client system. Cost is shown as an index so the format carries no prices. A client scorecard runs on the client's own cases, models and settings, and the test runner stays in the client's repository.