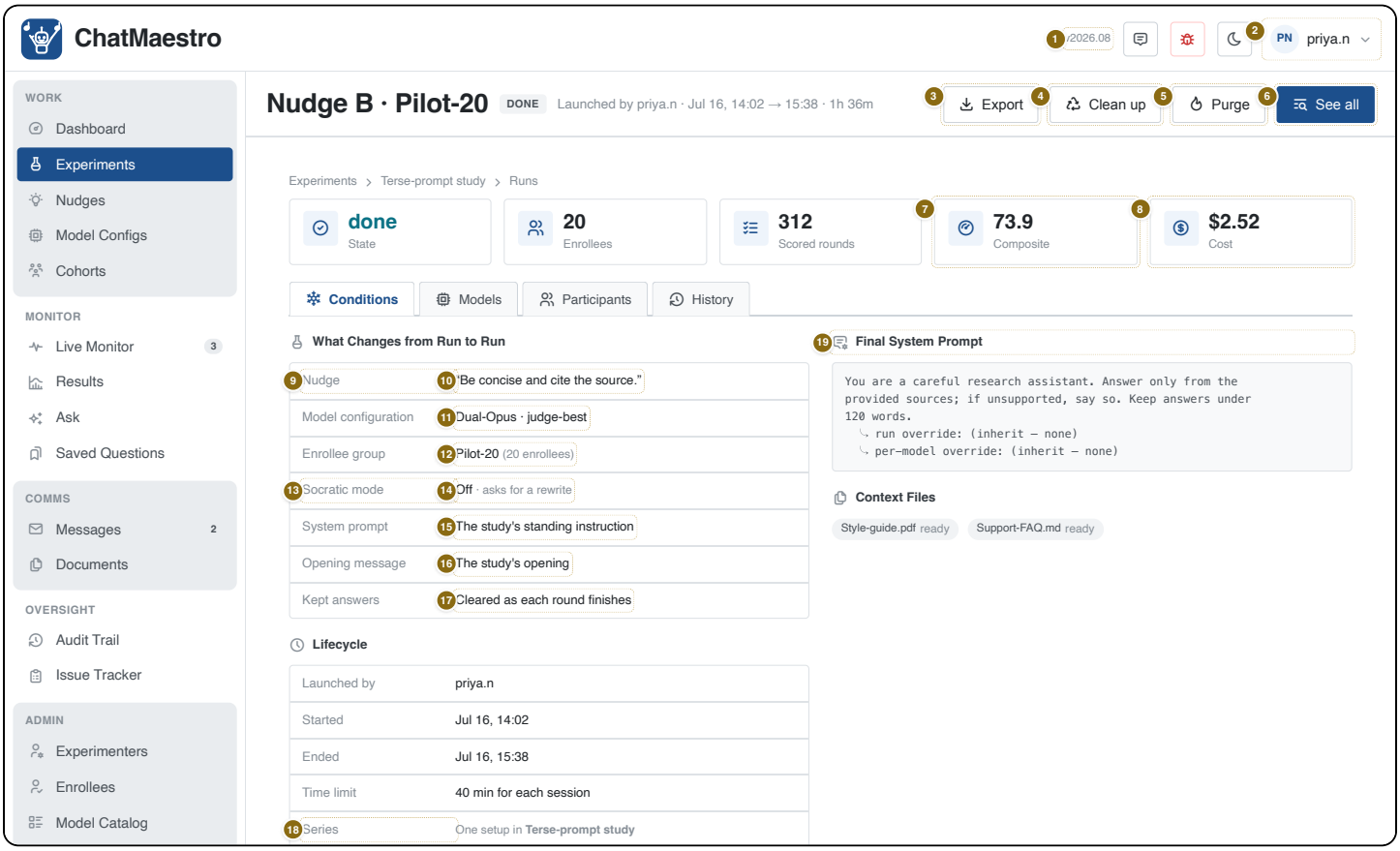
Run detail

`run-detail.html · 1360×820`



Annotations

1. **Version pill (right-aligned)** Dimmed build pill sits on the RIGHT, beside the account menu. Hover reveals the build sha; on phones it folds into the account menu.

2. **Account menu** Avatar + name + chevron open the account menu (profile, settings, sign out). On narrow screens the feedback / bug / dark-mode icons and version pill collapse into here.

3. Download this run's results.

4. **Clearing a run — the gentler way Cleanup** deletes the heavy data the run produced (transcripts, candidate answers, messages, enrollments and the run's audit rows) and **keeps** the `experiment_run` row and its `run_result` scorecard. An admin or the owning experimenter may run it, and only on a `done` or `aborted` run. The same action sits on the run's row in the Experiments screen's runs pane, so a run need not be opened to be cleared. Reclaim this run's storage, keeping its scorecard so comparisons still work.

5. **Clearing a run — the harder way (admin only) Purge** drops the `experiment_run` row too, which cascades `run_result` away, so the run leaves every comparison. It is what finally releases a definition that is held immutable while a run references it. Remove this run and its scorecard entirely (admin only).

6. View every response in this run.

7. **Composite score** `run_result` — the scorecard row, built **once** when the run hits *done*. Sibling nudges are compared on this. One overall score from 0 to 100 that blends the quality measures this study's rubric names.

8. **Cost** Split under the hood: **generation** tokens (the doer models) + **judge** tokens (the trusted judge).

9. A short steering message the person sees during the chat.

10. **A per-run variable** `experiment_run.nudge_id` — the enrollee-safe steering message. It's **one of four** per-run variables — alongside the model configuration, the cohort and the Socratic switch. A comparison varies **one** and holds the rest constant; this run's series varies the nudge.

11. **A per-run variable · reusable model set** `model_config_id` — a versioned, named bundle of models + combine method (like a cohort, but for models). Also swappable across sibling runs; reused across runs.

12. **Who saw it** `cohort_id` — the enrolled panel. A **disjoint-cohort gate** at launch stops an enrollee being live in two runs at once.

13. How the assistant replies when a request is missing something.

14. **The fourth per-run variable** `experiment_run.socratic_enabled`. It decides how the sufficiency gate replies when it stops a request: on, the trusted judge asks for one missing piece at a time; off, it asks for the request to be rewritten and resent. Detection is the same either way, and the wording of both replies is frozen on the study, so two runs differ in this and nothing else. The models under test never ask anything themselves.

15. **Set on the study, not the run** `experiment.system_prompt`. There is **no** per-run prompt override: a run varies only the nudge, the model configuration, the cohort and the Socratic switch, and holds the rest constant, which is what makes its siblings comparable. The one layering that exists is per **model** — `model_config_model.system_prompt`, which belongs to the configuration, not to this run.

16. **Set on the study, not the run** `experiment.opening_message` — the enrollee-safe first assistant message. Like the system prompt it is a property of the study and is identical across all of its runs; there is no per-run override.

17. **Housekeeping** `experiment_run.keep_candidates` (boolean, default `false`) — whether the per-model, per-iteration rows behind “See all” are retained after each round finalizes. Here it is false, so they clear as the run proceeds.

18. Compare this run with the study's other runs on the Results screen.

19. **Why this is faithful** It is **re-resolved**, not replayed from a copy: there is no snapshot column. The run holds foreign keys to its experiment and its model configuration, and the database keeps those rows unchangeable while any run references them, so resolving the cascade again today reproduces exactly what the models were sent. A stored copy would be a second source of truth able to disagree with the first, which is why the design does not keep one. The prompt shown is the **2-layer** cascade flattened — each model's own `model_config_model.system_prompt` first, then `experiment.system_prompt`; here no member carries a role of its own, so the study's instruction stands alone. Re-resolved from the study and the configuration, which the database holds unchangeable for as long as this run points at them.