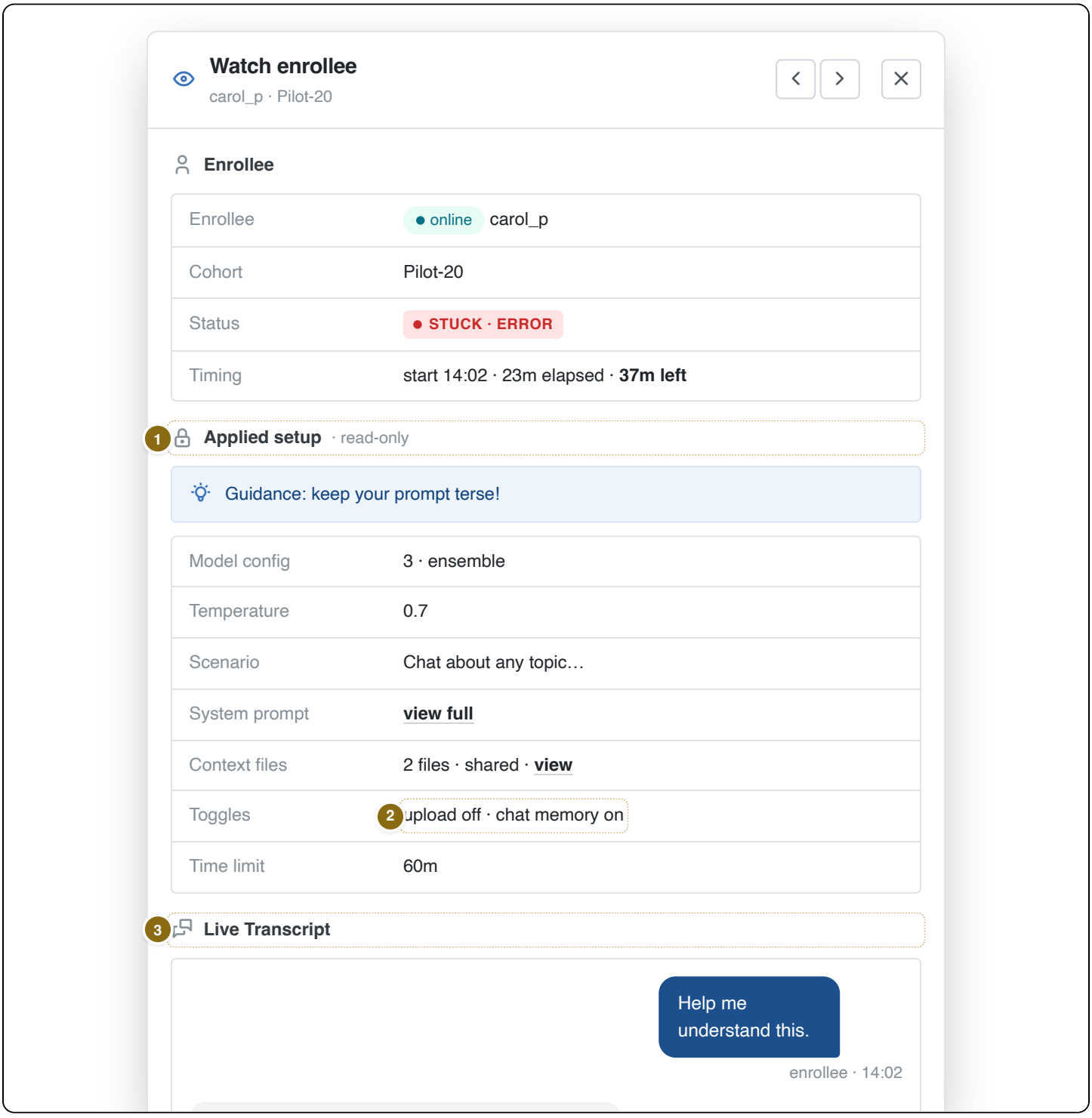
Watch drawer

`watch-drawer.html · 760×776`



Annotations

1 **Read-only condition** The frozen condition applied to this enrollee. The watch drawer is for **observing** — it doesn't re-steer the run mid-flight. This setup is **read-only** here — the watch drawer lets you observe, not change the run while it's running.

2. **Only two of these are settings upload** follows `experiment.blinded` — there is no separate upload switch. **chat memory** is `experiment.enable_chat_history`. Clear-session, rewind and example prompts are chat conveniences that are always available and are not configured on the experiment.

3. **Instrumented transcript** Every model call, its route (**ensemble → chosen model**) and latency is captured — the live transcript doubles as the analytics record.

4. **Turned back before a round existed** The sufficiency gate replied instead of the models. No `chat_round` was written, so this attempt is not in the run's round count, cost per round or score distribution — it is kept in `chat_round.clarification` on the round it led to. The models under test never saw the message; the gate runs on the trusted judge, and its tokens land in `judge_tokens` / `judge_cost`.

5. **Enhanced before generation** This study has the prompt enhancer on, so the message was rewritten before the models saw it. `chat_round.raw_prompt` keeps the enrollee's own words, `chat_round.prompt` the rewrite the models answered, and `chat_round.enhancement` which transformations applied — or the check that rejected a rewrite and fell back to the original, which is why a fallback is visible rather than looking like the enhancer having been off. Scoring measures the answer against the raw prompt.

6. **Memory limit reached** The session outgrew the budget, so the oldest turns were dropped and the conversation carried on — truncation never stops a session. The budget comes from the **smallest** `model_catalog.context_window` among the set-up's members, so every model still received identical input. The enrollee saw their own version of this notice; that it fired is recorded in `chat_round.history_window.alert_shown` (ALGORITHMS §23).

7. **Session-level ops** Retry / restart / remove act on **this enrollee's session only** — never the frozen run condition. Removing is a **soft** withdraw: rounds already produced stay in the analytics.

8. View every response in this run.

9. Retry the failed model call.

10. Remove this enrollee from the run.

11. Close this dialog.