WHICH MODEL FITS THIS JOB?

Model choice is a system design decision.
A support summary, a policy answer, and tool arguments need different evidence.
Today: build a task scorecard before choosing a model.

SUPPORT SUMMARY

Summarize a customer conversation.

POLICY ANSWER

Answer a question using policy information.

TOOL ARGUMENTS

Create tool arguments to take an action.

TASK SCORECARD

- [ ] Required capabilities
- [ ] Key evidence needed
- [ ] Constraints
- [ ] Evaluation criteria

Compare each candidate against the task scorecard.

WRITE THE JOB FIRST

Toy task: draft a read-only support answer from supplied policy evidence.
Keep the return deadline and exception correct.
Do not invent an order status or execute a refund.

Policy evidence
(from knowledge base)

Returns Policy

Items can be returned within 30 days of delivery.

Exceptions:

- Final sale items cannot be returned.
- Personalized items are not eligible for return.

Application
(draft a reply)

Draft reply (read-only)

Based on our return policy, you can return the item within 30 days of delivery. Note that final sale items and personalized items are not eligible for return.

Application boundary

Refund action
(outside application)

Execute a refund
(do not do this)

Read the evidence.
Draft a response.
Do not take actions.

SEPARATE GATES FROM PREFERENCES

Gates decide eligibility: approved data handling, required language support, and critical task rules.

Preferences compare eligible candidates: quality, latency, operating cost.

All candidates

Eligibility Gates

- Approved data handling
- Required language support
- Critical task rules

All gates pass

Any gate fails

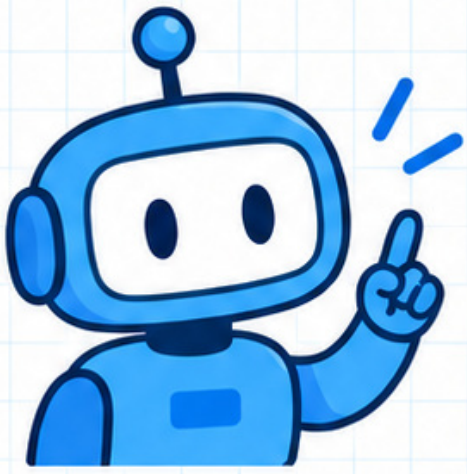
Fails a gate (exits before scoring)

Preference Comparison

- Quality
- Latency
- Operating cost

Selected Candidate

Best eligible model



A low price cannot erase a failed gate.

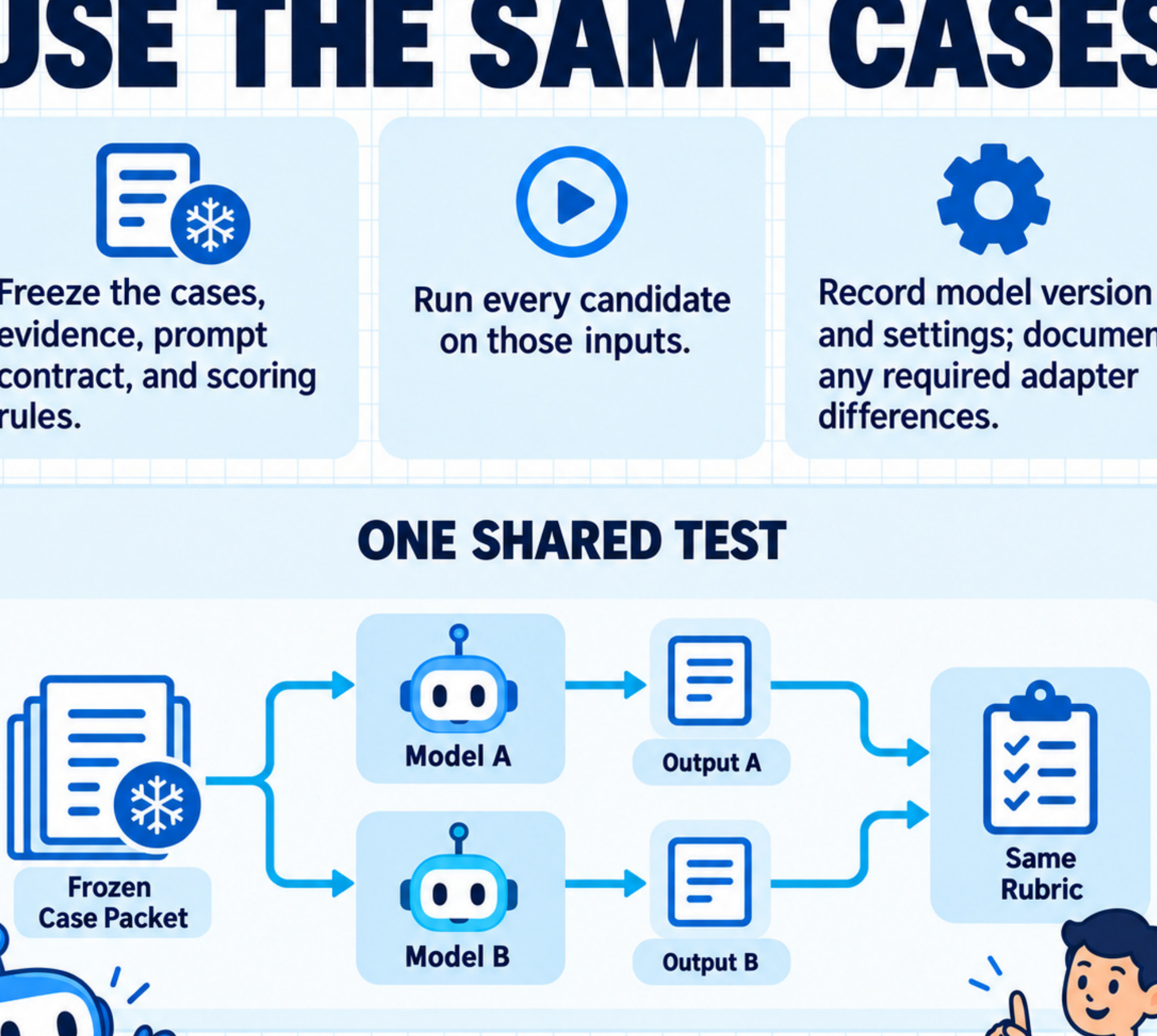
USE THE SAME CASES

Freeze the cases, evidence, prompt contract, and scoring rules.

Run every candidate on those inputs.

Record model version and settings; document any required adapter differences.

ONE SHARED TEST



COVER THE DIFFICULT SLICES

Include short and long requests, each required language, missing evidence, and permission failures.
Report results for each slice as well as the whole set.
An average can hide a weak group.

Short requests
"Summarize this."

Long requests
"Explain step by step with details."

Each required language
"Respond in Spanish."

Missing evidence
"Answer with the given information only."

Permission failures
"You do not have access to this data."

EVALUATION REPORT

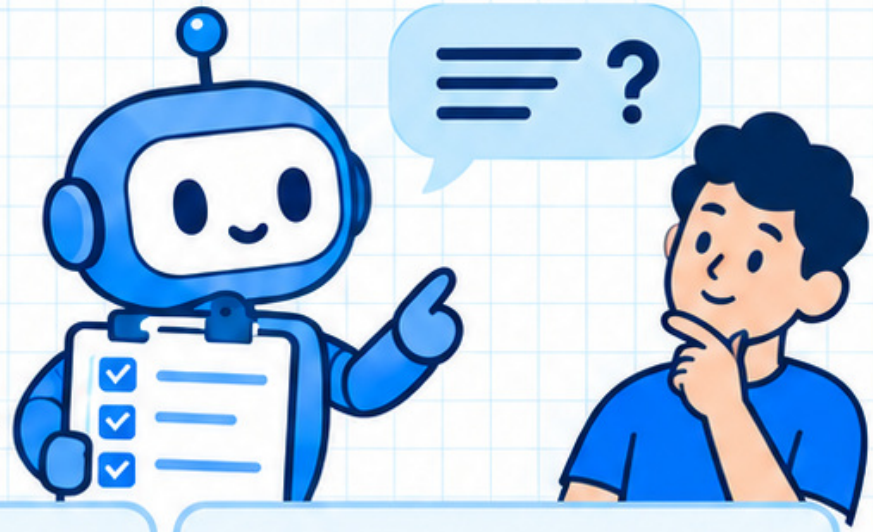
Slice	Result
All requests (aggregate)	—
Short requests	—
Long requests	—
Each required language	—
Missing evidence	—
Permission failures	—

DEFINE SUCCESS PER CASE

A policy answer passes only if required facts are correct, supported, and complete.

A tool-argument case also needs valid fields and correct values.

Valid shape alone does not prove a useful result.



SCHEMA SHAPE

The output matches the expected structure (e.g., valid JSON shape).

TASK MEANING

The content correctly addresses the task and intent.

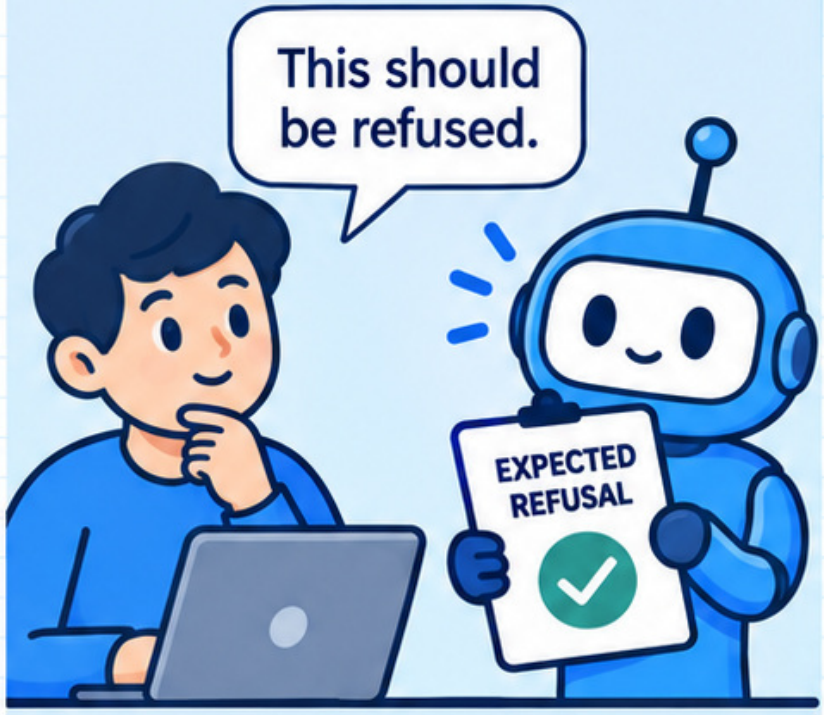
EVIDENCE SUPPORT

Claims are supported by the required facts and sources, and the answer is complete.

COUNT FAILURES HONESTLY

Timeouts, malformed outputs, unsupported claims, and inappropriate refusals belong in the results.

An expected refusal may pass its own case.



Outcome Ledger

Outcome	Description	Expected Behavior	Pass / Fail
Success	A valid, correct output.	Answer	Pass
Expected Refusal	A refusal that is appropriate for the request.	Refuse	Pass
Unexpected Refusal	A refusal when an answer was expected.	Answer	Fail
Timeout	No response within the limit.	Answer	Fail
Invalid Output	Malformed output, unsupported claims, or inappropriate refusals.	Answer	Fail

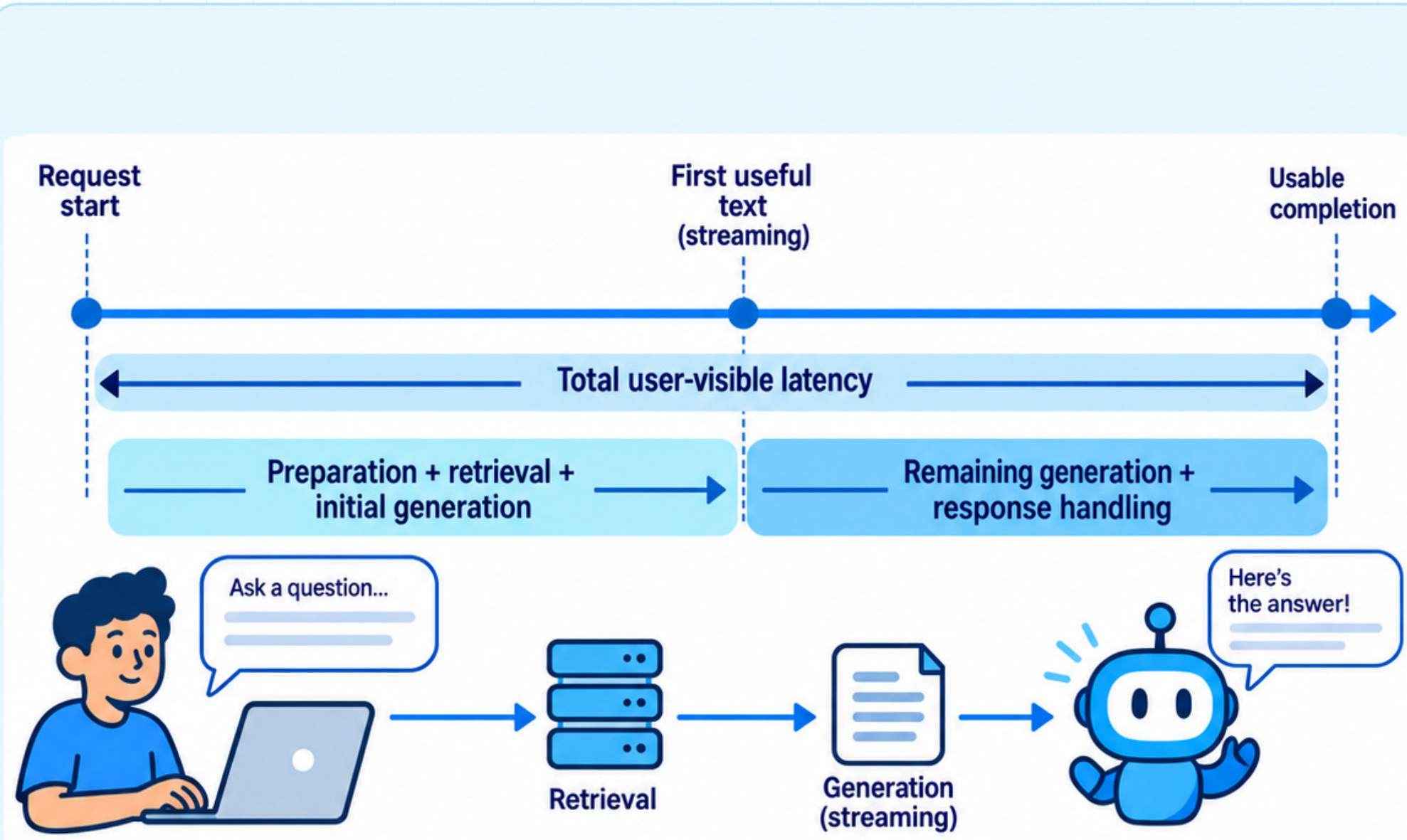
Do not silently drop unsuccessful attempts.

MEASURE THE WHOLE REQUEST

Measure user-visible latency from request **start** to usable **completion.**

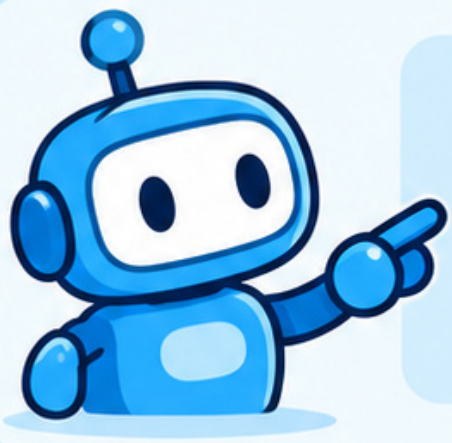
For streaming, also measure time to first useful text.

Record workload, concurrency, sample size, and timeout rules.



PRICE THE WORKLOAD

Estimate input, output, retries, retrieval, and other service costs.



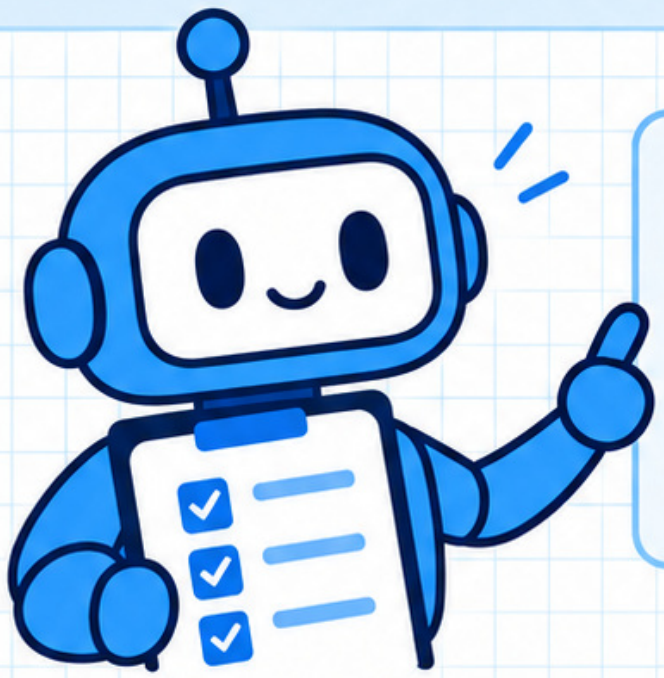
$$\text{Cost per successful task} = \frac{\text{total workload cost}}{\text{successful tasks.}}$$

A TOY CANDIDATE SCORECARD

Illustrative results: 20 shared cases, one run each.

Candidate	Passes (toy)	p95 (toy)	Total cost (toy)
A	18/20 pass	4 s	2 units
B	19/20 pass	6 s	3 units

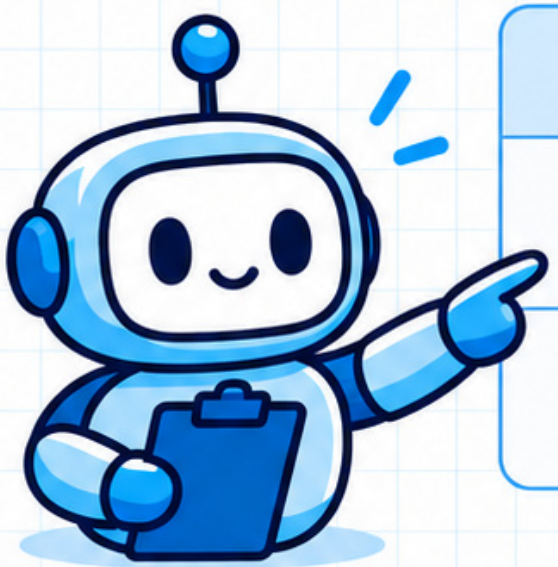
Both meet the example data-handling gate.



A	B
18/20 pass	19/20 pass
4 s	6 s
2 units	3 units

CHOOSE UNDER A CONSTRAINT

Toy requirement: at least 18/20 passes and p95 at most 5 s.



Model	Passes (out of 20)	p95 latency (s)
A	18/20 ✓	4 ✓
B	19/20 ✓	6 ✗

A meets both example thresholds.
B misses latency.

This **small test** supports a trial of A,
not a universal winner.

SMALL SAMPLES NEED CAUTION

A one-case lead can change on another run.
Repeat cases when outputs vary; keep a separate held-out set.
Report sample size and uncertainty before claiming an improvement.

CANDIDATE A

Different approach

CANDIDATE B

Another approach

Small sample

Repeated runs

Run 1 result

Run 2 result

Run 3 result

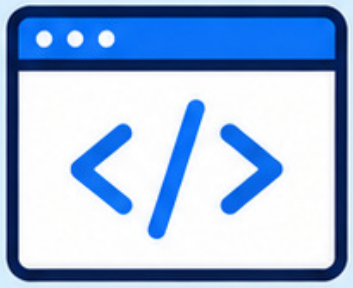
Repeated runs

Run 1 result

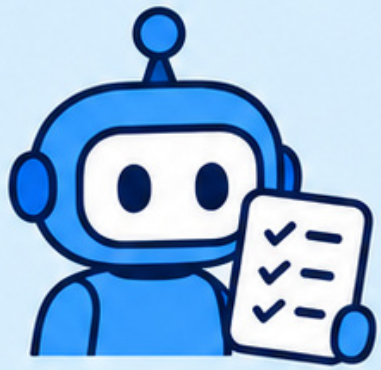
Run 2 result

Run 3 result

REVIEW JUDGE DISAGREEMENTS



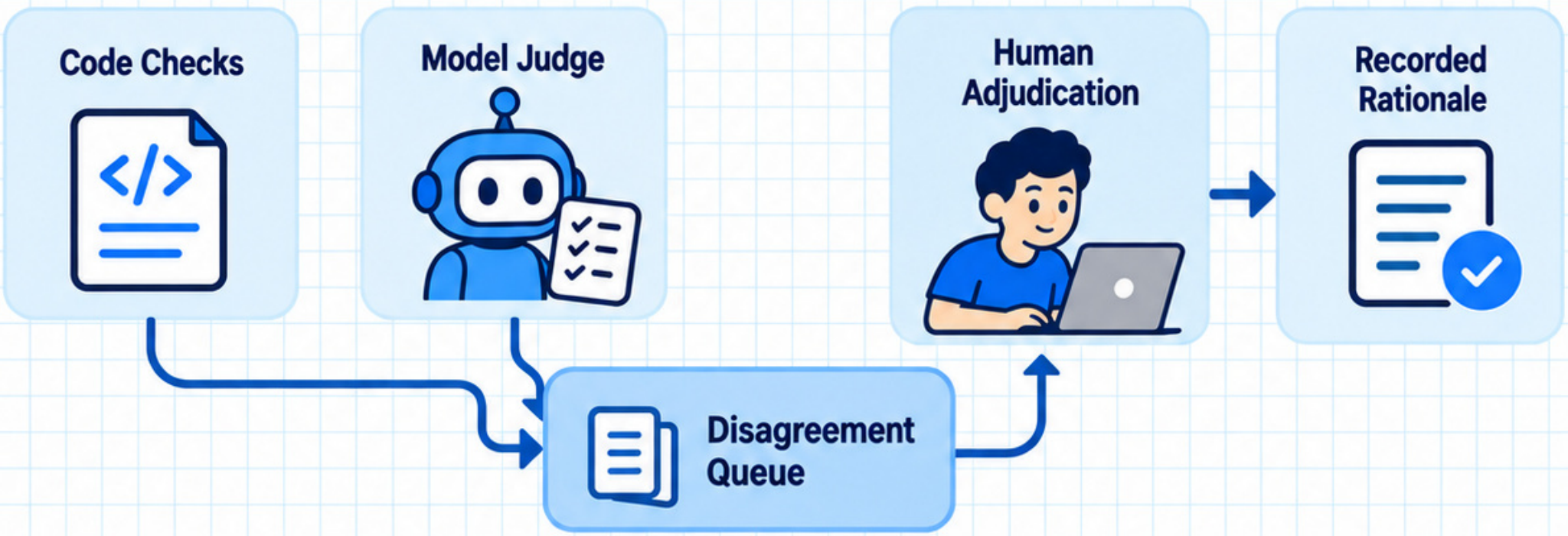
Use code for exact rules and human review for nuanced meaning.



A model judge can assist, but test it against reviewed examples.

Investigate disagreements instead of treating its score as truth.

REVIEW DISAGREEMENTS

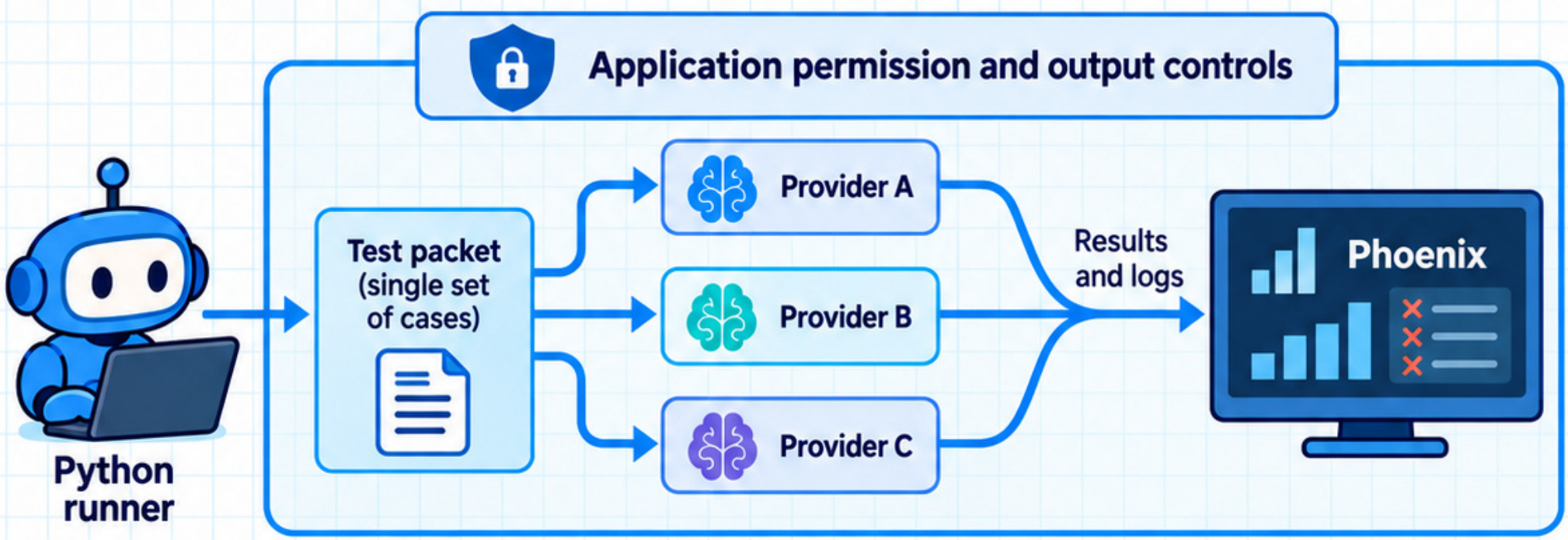


WHERE THE TOOLS FIT

Python runner: send cases and capture outcomes.

Phoenix: compare experiment results and inspect failures.

Your application: enforce permissions and output handling for every candidate.



RECORD THE DECISION

Save dataset version, prompt version, model identifier, settings, rubric, and dates.

Add slice results, latency conditions, cost assumptions, and unresolved risks.

Dataset version

Prompt version

Model identifier

Settings

Rubric

Dates

DECISION RECORD

Version 1

Dataset version

Prompt version

Model identifier

Settings

Rubric

Dates

Slice results

Latency conditions

Cost assumptions

Unresolved risks

Slice results

Latency conditions

Cost assumptions

Unresolved risks

Make the next comparison reproducible.

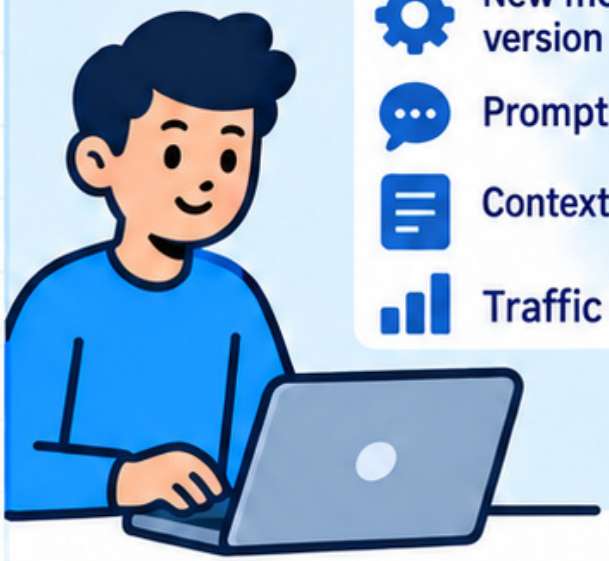
RE-EVALUATE AFTER CHANGES

A new model version, prompt, context policy, or traffic mix can change the result.

Rerun the relevant suite and critical gates.

Keep a tested fallback and a rollback owner.

CHANGE EVENT



- New model version
- Prompt
- Context policy
- Traffic mix

EVALUATION GATES

- Rerun relevant suite
- Check critical gates
- Verify readiness

PASS

Controlled rollout

FAIL

Route to tested fallback

TRY THE SCORECARD

Toy candidates on 20 cases:

CANDIDATE C

19 pass

p95 3 s

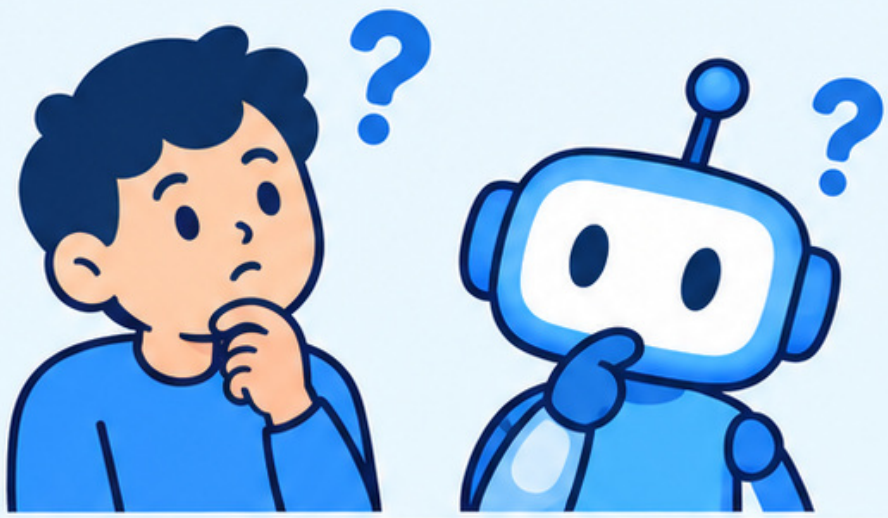
data gate fails.

CANDIDATE D

18 pass

p95 4 s

data gate passes.

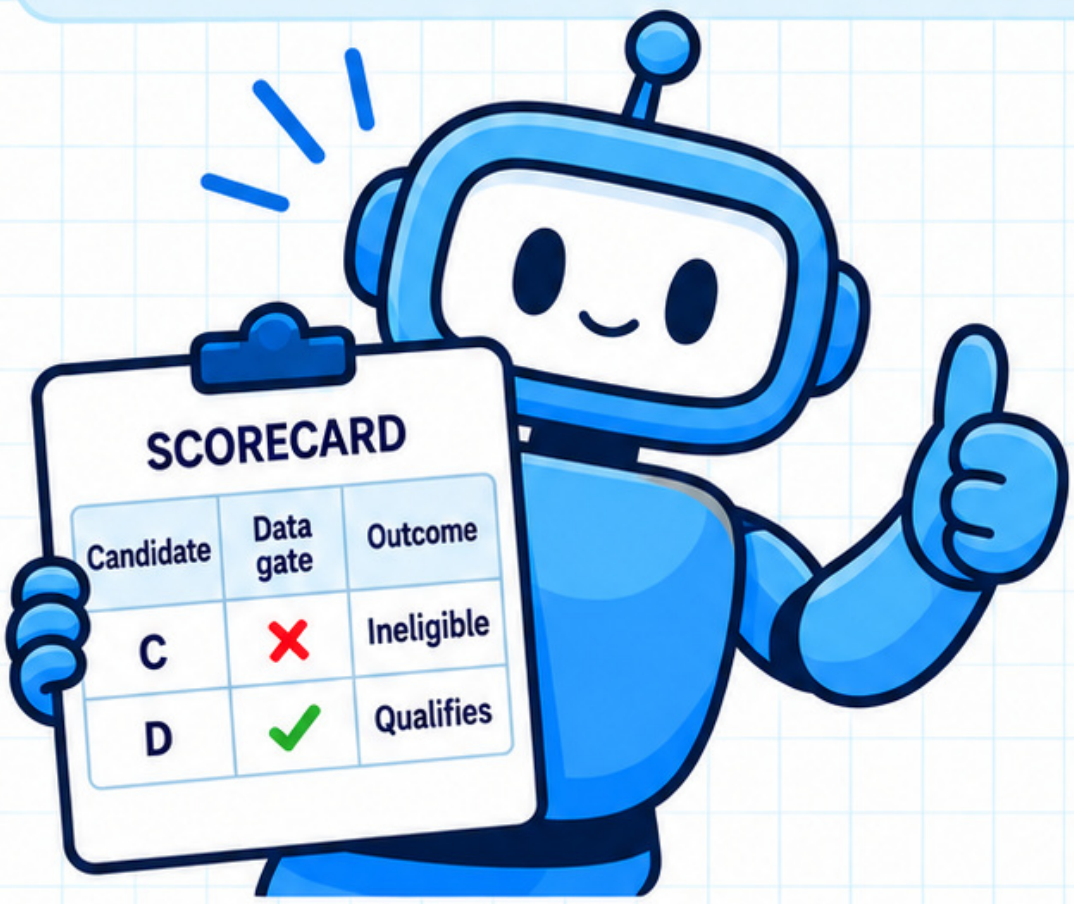


Require the data gate, 18 passes, and p95 at most 5 s.
Which candidate qualifies?

?

CHOOSE WITH EVIDENCE

Task first. Gates before tradeoffs. Same cases for every candidate.
Count failures, inspect slices, and record uncertainty.
Exercise: only D qualifies under the toy requirements.



1 Task

2 Compare

3 Decide

Next: prompting, retrieval, or fine-tuning?