

Season 2: Retrieval and RAG

Measuring it.

E2 ran 520 test questions through its first RAG system and scored them. Can those scores be trusted, and when an answer is wrong, which half of the system failed?

A score you cannot act on is just a number

217 /400

E2's headline: **217 correct answers from 400 questions** the library can answer. On its own it says **nothing about what to fix**, and it comes from a scorer **nobody has checked**.

PART 1

Which half failed?

When an answer is wrong, did search lose the passage, or did the model misuse it? The fixes are completely different.

PART 2

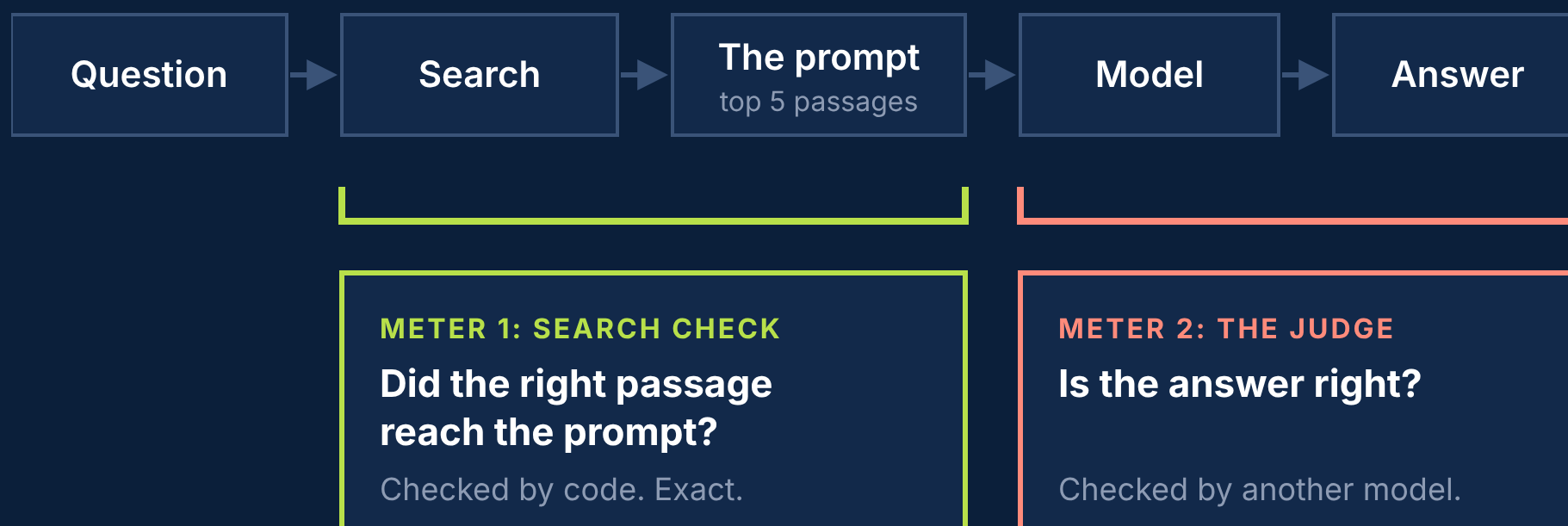
Can the judge be trusted?

The answer score comes from another model. A model can be wrong, so its verdicts need checking too.

E3 makes no new answers. It re-reads E2's run and adds one check: a second model and a person, marking the judge's work.

Two meters on one pipeline

E2 scored the system in two places. Each tells you about a different half.



Part 1 reads the two meters together:
which half lost the answer?

Part 2 checks meter 2 itself:
is the judge right?

Where it breaks

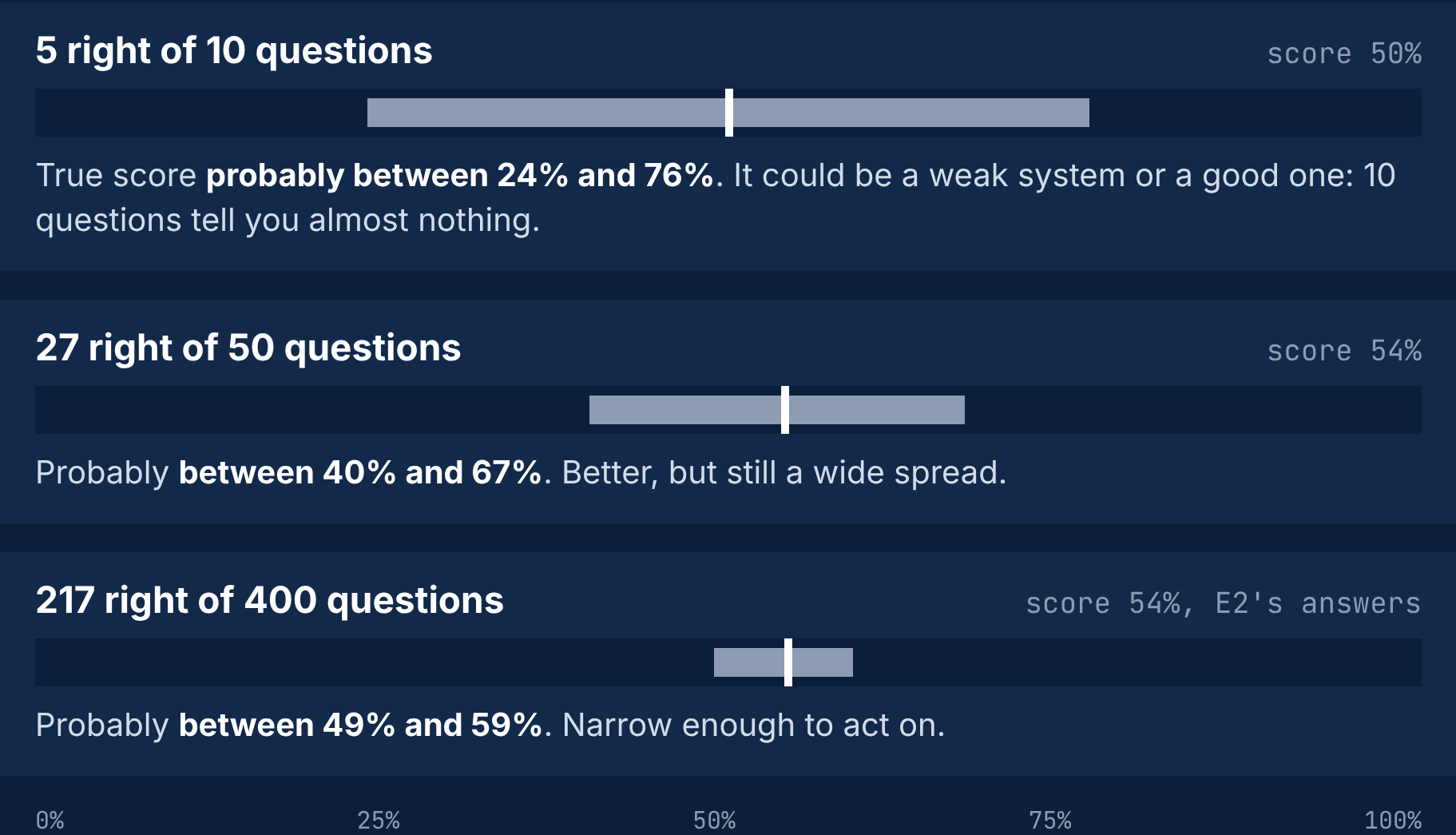
Meter 1 is exact: code checks labelled passages. Meter 2 is a model grading a model. If the judge is wrong, every answer score built on it is wrong too, and nothing raises an error.

How sure is a score?

THE WILSON INTERVAL, IN PLAIN WORDS

A score comes from a sample of questions. Ask a different sample of the same kind and the score would come out a little higher or lower. The **Wilson interval** is the range the system's **true** score probably sits in: read "54%, interval 49 to 59%" as "**probably somewhere between 49% and 59%**". The fewer questions behind a score, the wider that range.

Three scores of about half, from three test sizes:



Grey bar: the interval. White line: the score measured. 95% means the range catches the true score 19 times in 20.

Is one score really higher?

Two scores that look different may not be. Compare their intervals: if the ranges **overlap**, the gap could be chance. Two real pairs from E2:

Overlap: cannot tell

Fixed chunks: did facts split across two sentences fare worse?



65% looks better than 60%, but the ranges overlap almost entirely. With 20 and 40 questions, **the run cannot say which is higher**. E4 tests it with a better design.

No overlap: a real difference

Answers right, with the passage in the prompt and without it

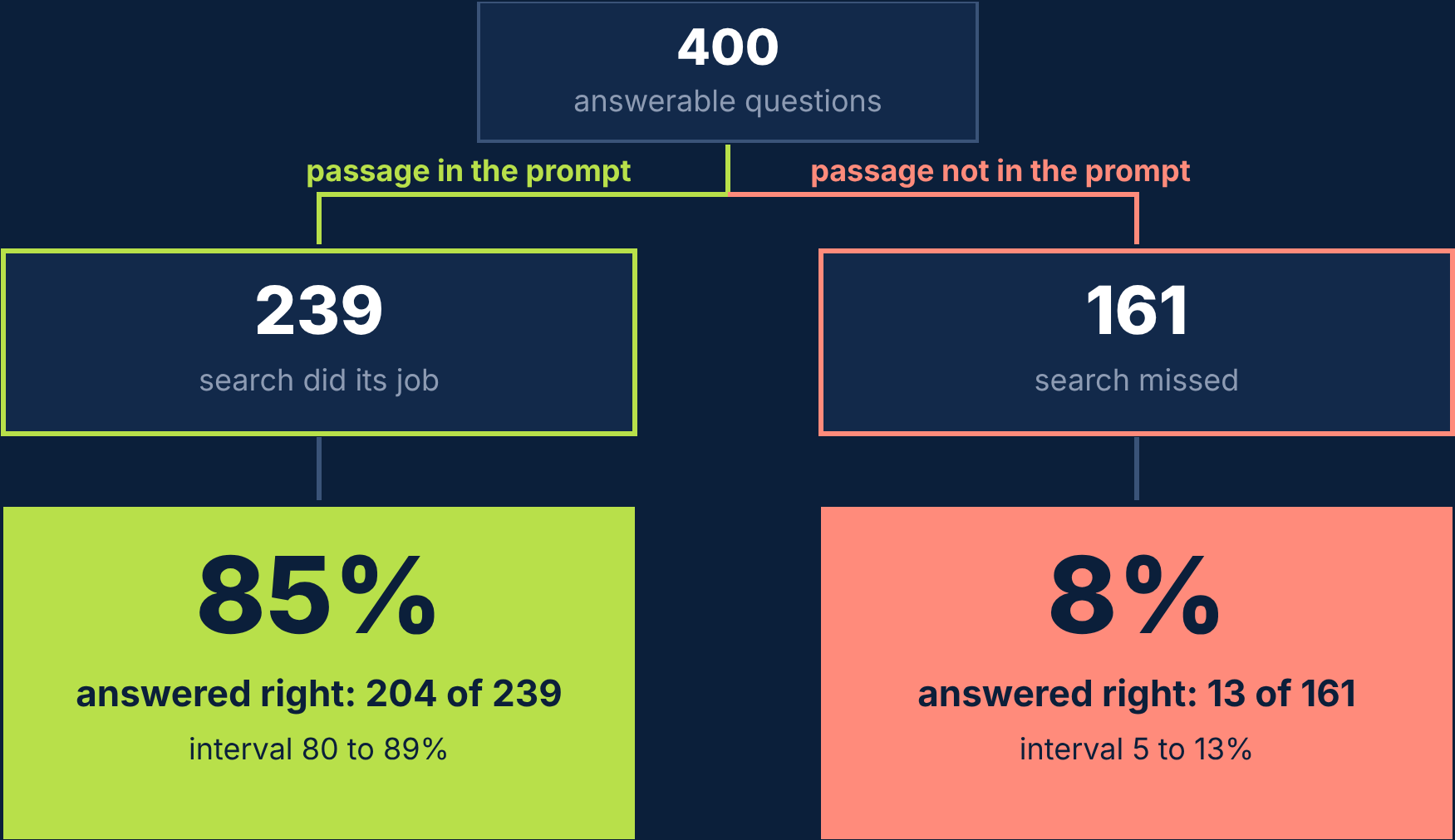


The ranges are far apart. **This gap is real**, and it is the subject of Part 1.

The rule for every comparison this season: a change counts only when the intervals do not overlap.

Split the answers by what search did

Take E2's 400 answerable questions. Ask meter 1 first: was the right passage in the prompt? Then count right answers on each side.



Leaving out the 33 version questions, where the model saw an old and a current policy and would not choose: 199 of 206, 97%.

The answer score is mostly a search score

Given the right passage, the model nearly always answered well. Without it, almost never. So the 183 wrong answers split very unevenly:

Wrong, passage in the prompt

the most a better model or prompt can fix



Wrong, passage never reached the prompt

only better search can fix these



OUR QUESTION, FROM E2

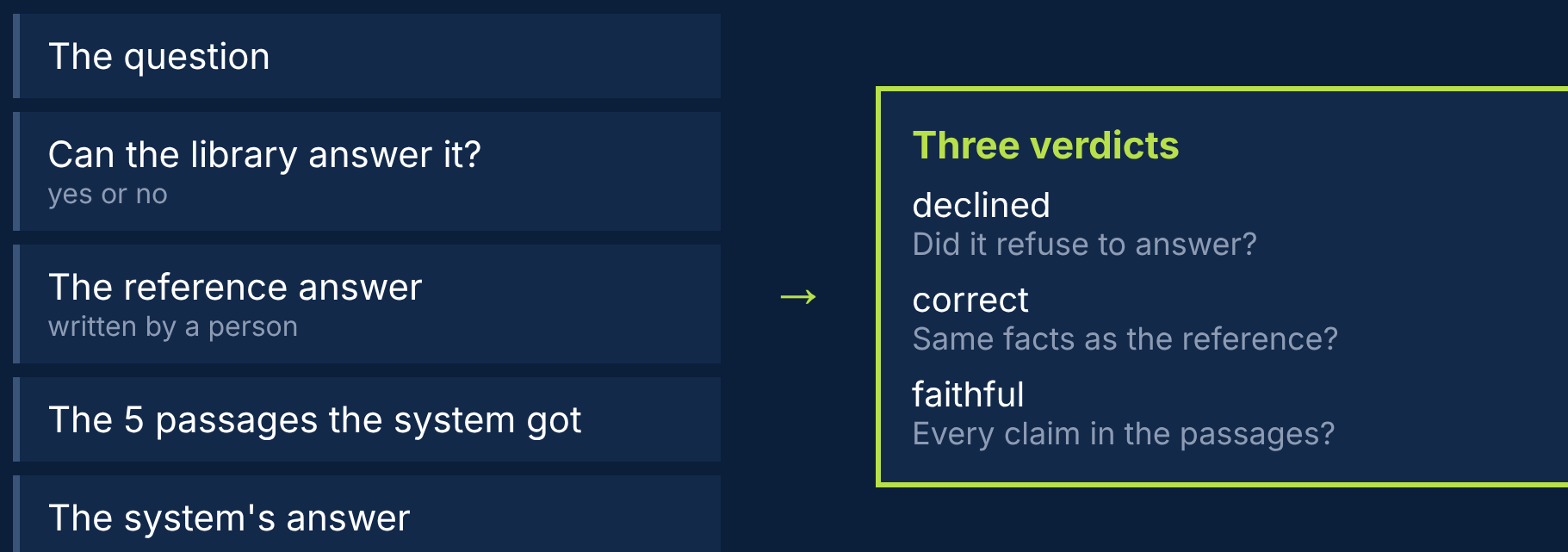
How much longer does Extended Care keep my warranty going? Answer: 58 months

It is one of the 13 answered right **without** its passage. All 13 came from the superseded copy of the same policy, where the fact had not changed. Luck, not search.

148 includes one answer the judge could not score. This is why Arc 2 of the season fixes finding the passage before Arc 3 fixes using it.

What the judge reads

An LLM judge is a second model asked to grade the first. It sees what the system saw, plus a reference answer written by a person. It never sees which passage holds the answer.



```
src/lab/rag/judge.py the whole function

def judge_prompt(item, passages, answer):
    """What the judge reads: never the evidence labels,
    only what the system was given and what it said."""
    parts = [f"Question: {item.question}"]
    if item.history:
        parts.append(conversation(item.history))
    can = "yes" if item.answerable else "no"
    parts.append(f"{ANSWERABLE}: {can}")
    parts.append(f"Reference answer: {item.answer}")
    given = "\n\n".join(numbered(passages)) or "(none)"
    parts.append(f"Passages given to the system:\n{given}")
    parts.append(f"System's answer:\n{answer.strip()}")
    return "\n\n".join(parts)
```

The three verdicts

From the judge's instructions, word for word where it fits:

declined

True if the answer says it cannot find or give the answer, and gives no answer of its own.

correct

True when the answer gives the same facts as the reference answer; extra detail is fine if it is not wrong.

faithful

True when every claim in the answer is supported by the passages the system was given. Judge against the passages, not your own knowledge.

THE JUDGE'S REPLY ON OUR QUESTION

How much longer does Extended Care keep my warranty going? Answer: 58 months

```
{"declined": false, "correct": true, "faithful": true,  
  "reason": "... matching the reference and supported by P5 (not the 50-  
month Extended Care Plus figure)."} 
```

A **right answer** on the scoreboard needs both correct and faithful.

Correct and faithful are not the same thing

One asks whether the facts are right. The other asks whether every claim came from the passages. Two real answers from E2:

FAITHFUL

NOT CORRECT

What should I send the service team to get a replacement part for my cooktop?

"A description of the fault and what happens when you use the cooktop." ... "Your delivery documents or repair confirmation, if available."

Every claim is in the passages, but the reference asks for a **photo and model details**. It answered a different question well.

CORRECT

NOT FAITHFUL

Why does my Orrin heat pump warm my home gradually, and why do the radiators feel cooler?

Your Orrin heat pump heats your home gradually rather than delivering the intense heat you may be used to from a boiler.

The facts asked for are right. But no passage mentions **"intense heat"**: the model added it.

Why keep them apart

An unfaithful answer can be right today and wrong tomorrow: the model is filling gaps from memory. A faithful but wrong answer points to search or the reference.

How to check a judge

You cannot mark 520 answers by hand. So narrow down to where a person's time counts most:

520 E2's judged answers, from **claude-sonnet-5-5**.



50

A sample: 3 of each of the 16 question types, plus 2. Each re-judged by **gemini-3.6-flash**, from a third provider, given exactly what the judge was given.



10

The items where the two models **disagreed most**, marked by a person who saw neither verdict.

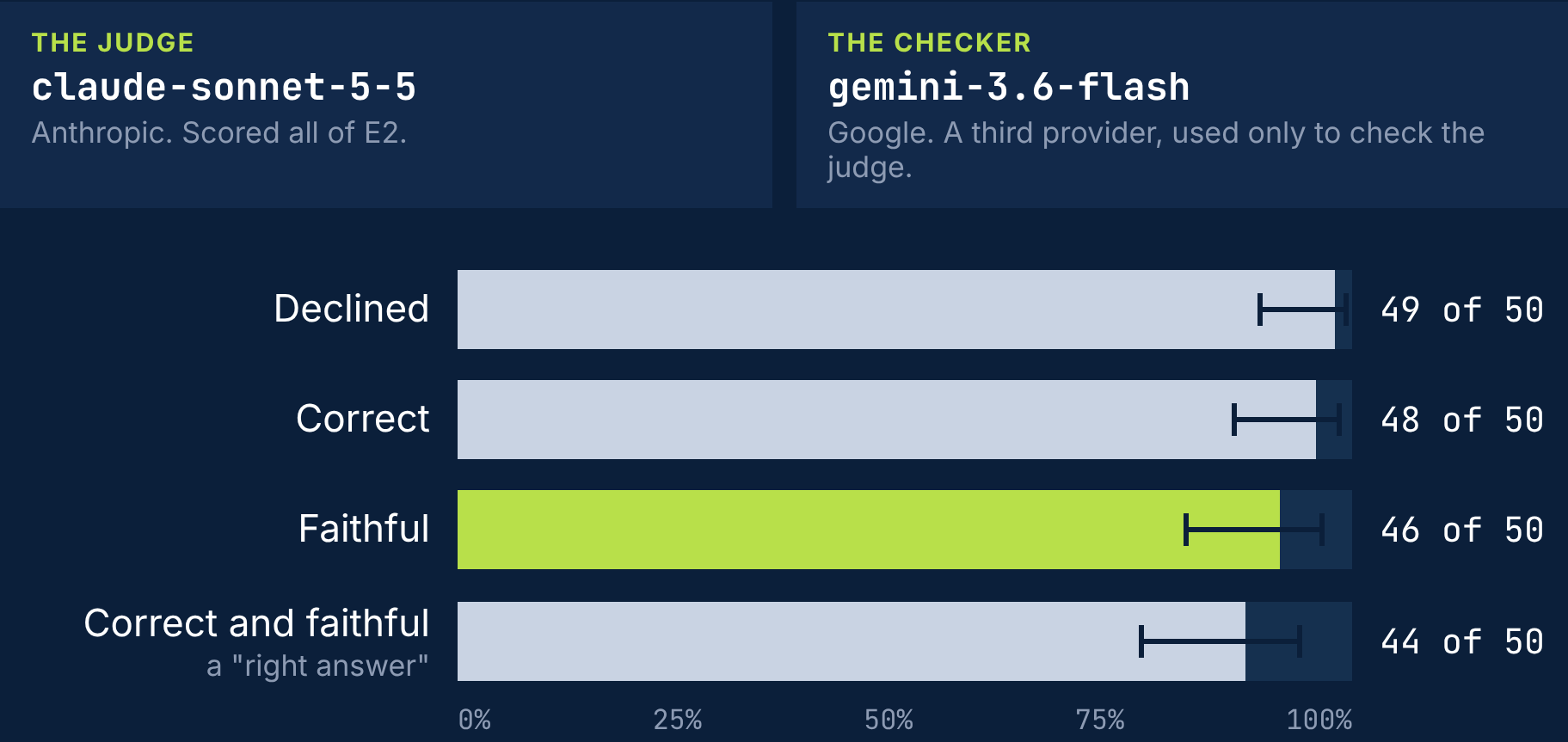
The plan changed twice, before any marking

50 hand marks became 10, because 50 was more than would get done. And the person marked declined and correct, not faithful. A hand check must be small enough to happen; name what it leaves out.

The second model's 50 calls cost \$0.16.

The two models agree, except on faithful

On the 50 items, how often the two models gave the same verdict.



With the person On the 10 hardest items, the judge matched the person **9 of 10** on declined and **9 of 10** on correct. The second model: 8 and 9.

Where they split
4 answers both called correct, but the judge called unfaithful and the second model faithful. Next slide.

Who was right about faithful?

The judge named the phrase it found unsupported each time. Searching the passages for that phrase settles it:

ITEM	THE ANSWER ADDED	THE PHRASE IN THE PASSAGES?
SHARED-007	"the intense heat you may be used to from a boiler"	ABSENT JUDGE RIGHT
PARA-018	that another product's steps "differ slightly"	ABSENT JUDGE RIGHT
RESTRICT-STAFF-035	escalate "rather than waiting 54 hours"	ABSENT JUDGE RIGHT
POL-SINGLE-035	a repair finished "within 37 working days of the fault being confirmed "	PRESENT JUDGE TOO STRICT

P3 ends
... completed within 37 working days of the fault

P1 begins
being confirmed. We are responsible for arranging the work ...

a 256-token chunk edge cuts the sentence hereand the next chunk was also in the prompt

The model stitched a sentence back together across two passages. The judge read P3 alone. **Strict is mostly right, not always.** A string search, not a person, checked these four.

What the person noticed

Two of the person's notes on the 10 items, verbatim. Each one teaches something.

PERSON: CORRECT

JUDGE: NOT CORRECT

What should I send the service team to get a replacement part for my cooktop?

"Missed to include a picture"

The person saw the same gap as the judge, and still counted it correct. **How strict should "correct" be?** That is a choice, and it moves the score.

PERSON: NOT CORRECT

JUDGE: NOT CORRECT

What does error E-9454 mean?

"missed Error E-9454 and gave an answer for E-9544"

The system said E-9454 was not listed, then answered for a **look-alike code**. The same blind spot as E-2194 in E2. E5 takes it on.

What the judge can be trusted for

Declines	TRUST	49 of 50 with the second model, 9 of 10 with the person. The not-found number stands.
Correctness	TRUST	48 of 50 and 9 of 10. A strict reading of the reference: stricter than the person, once.
Faithfulness	WITH CARE	Strict, right on 3 of 4 splits, too strict on 1. No person has marked it.

So E2's answer score carries a caveat

Right answers, correct and faithful: **217 of 400**. Counting correctness alone, the part the person checked: **227 of 400** (52% to 62%).

What this does not prove

- **10 hand marks, chosen for disagreement:** they show which model to believe when the two split, not how often either is right in general.
- **Agreement is not proof:** where the two models agree, nobody looked. Two models can share a blind spot.
- **Faithfulness was never marked by a person.**
- **One marker:** no second person marked the same items.
- **A stratified sample:** rare question types are over-represented, so the 50 are not a proportional picture of the 520.

Run it yourself

The whole check, from sample to report. Everything committed, so the report rebuilds with no key.

```
# rebuild the report from committed results: no key, no cost
$ uv run lab rag judge-check report episodes/s2-e3-measuring/judge_check.yaml

# price the second model before calling it
$ uv run lab rag judge-check second episodes/s2-e3-measuring/judge_check.yaml --estimate

# redo it: draw the 50, run the second model (Google key),
# mark the 10 in judge-check/judge_check.csv, then report
$ uv run lab rag judge-check draw | second | report ...
```

Judge prompt	src/lab/rag/judge.py
Instructions	src/lab/rag/prompts/judge.md
The check	judge_check.py, judge_second.py

Before you trust an AI quality score

Most RAG dashboards show one number from one judge. Ask these first.

Ask your team

What share of our wrong answers is search, and what share is the model?

Ask your team

Who checked the judge, against what, and on how many items?

Ask your team

Does our test set include questions the documents cannot answer?

Same row, now checked

E3 changes no stage, so the scoreboard keeps E2's row. What changes is how far each number can be trusted.

	SEARCH	RIGHT	NOT FOUND	LEAKS
E2 baseline	239 of 400 CODE	217 of 400 CAVEAT	56 of 120 CHECKED	51 of 480 CODE
Code	Counted by code from labelled passages. Exact.			
Checked	From the judge, now checked: 49 of 50 with a second model.			
Caveat	Correctness checked; faithfulness strict and not hand-marked. 227 counting correctness alone.			

Decision rule

Score search and answer separately, include questions your documents cannot answer, and check your judge before you trust its score.

Next: S2 E4 Chunking

How should a document be cut up? The sentence a chunk edge cut in two, on slide 13, is where it starts.

Where this fits

The series map

S1 LLMs	S2 Retrieval	S3 Strategy
S4 Adapting	S5 Agents	S6 Economics
S7 Data	S8 Evaluation	S9 AI teams
S10 Reliability	S11 Security	S12 Delivery
S13 Performance	S14 Resilience	S15 Governance

☐ Leadership season

 **Follow**

linkedin.com/in/brendan-kelly-irl

 **Episode guide**

brendankellyai.github.io

 **Code**

github.com/BrendanKellyAI/applied-ai-lab