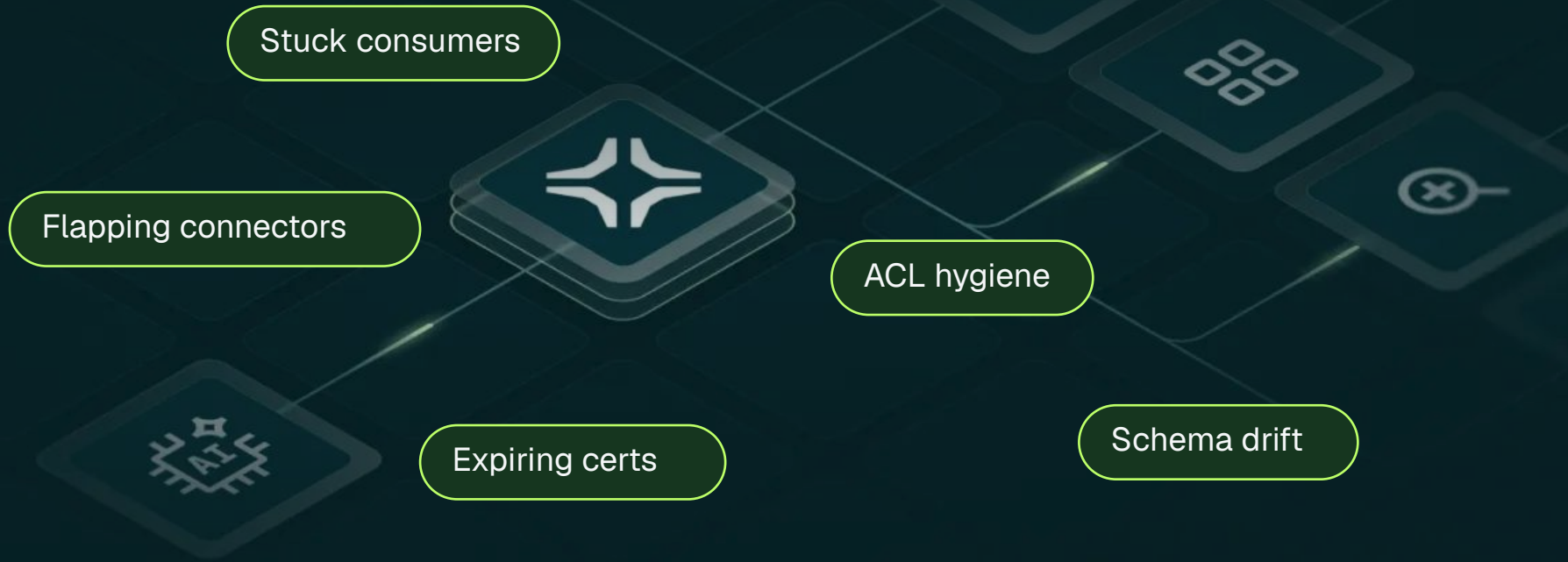
CURRENT 2026 · SAN FRANCISCO

How to NOT Evaluate an AI Agent

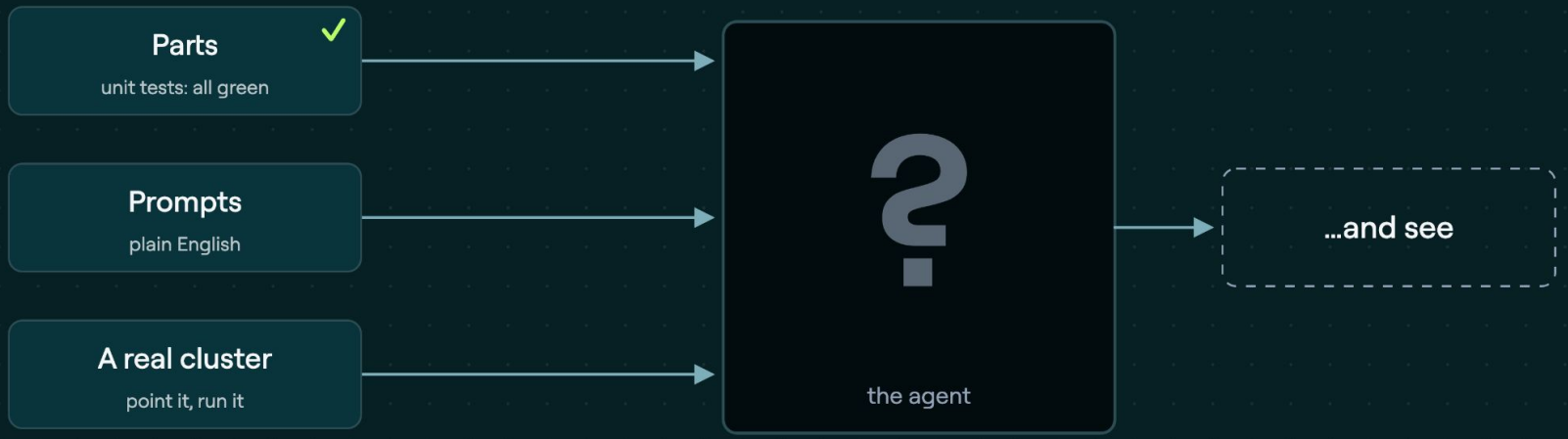
Eval-testing agents on real Kafka

Matt Searle · VP of Engineering @ Conduktor.io

The job: an agent with nobody watching



Attempt one: a black box



Every part of the agent machinery tested. The whole thing? Kind of …

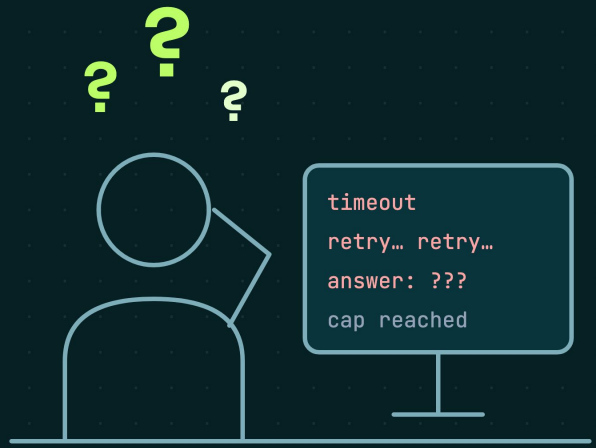
01 · THE NAIVE APPROACH

What came back

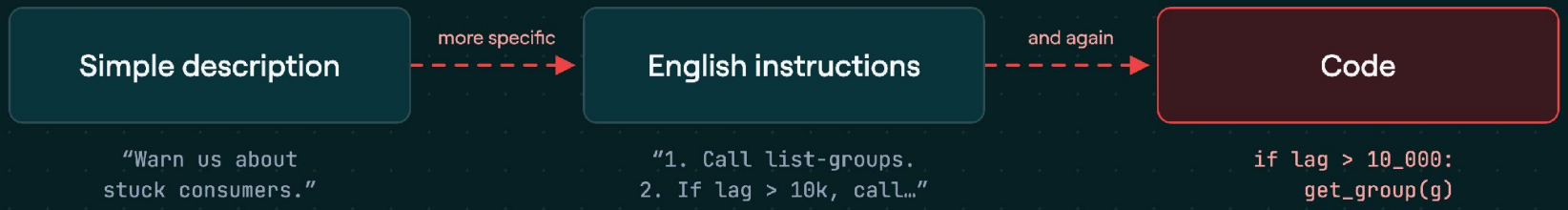
Timeouts
Out of calls. No answer.

Loops
Same failed call. Again.

Weird answers
Confident. Padded. Wrong.

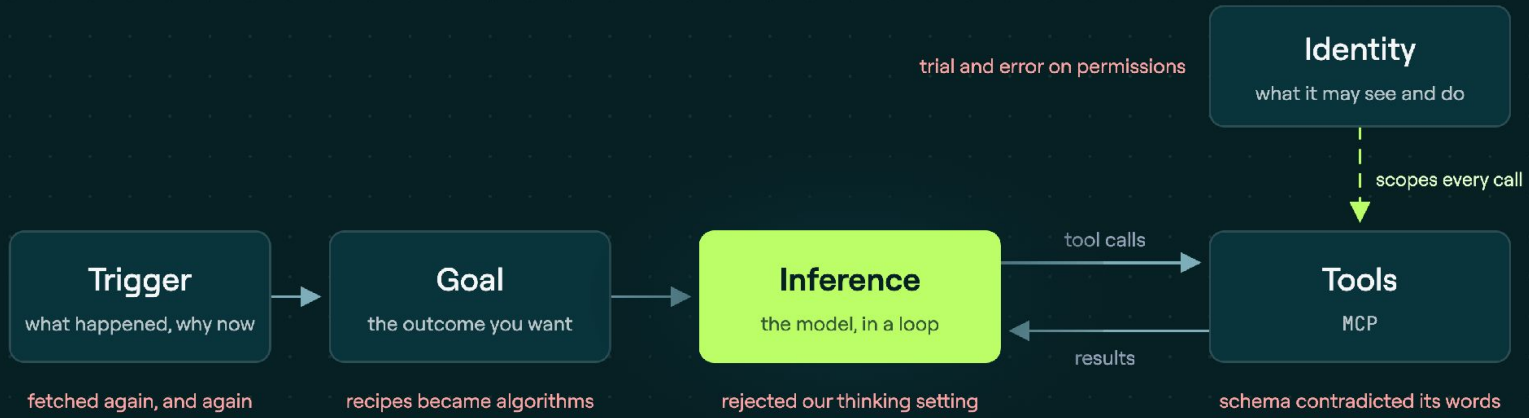


So we got more specific. Again … and Again …



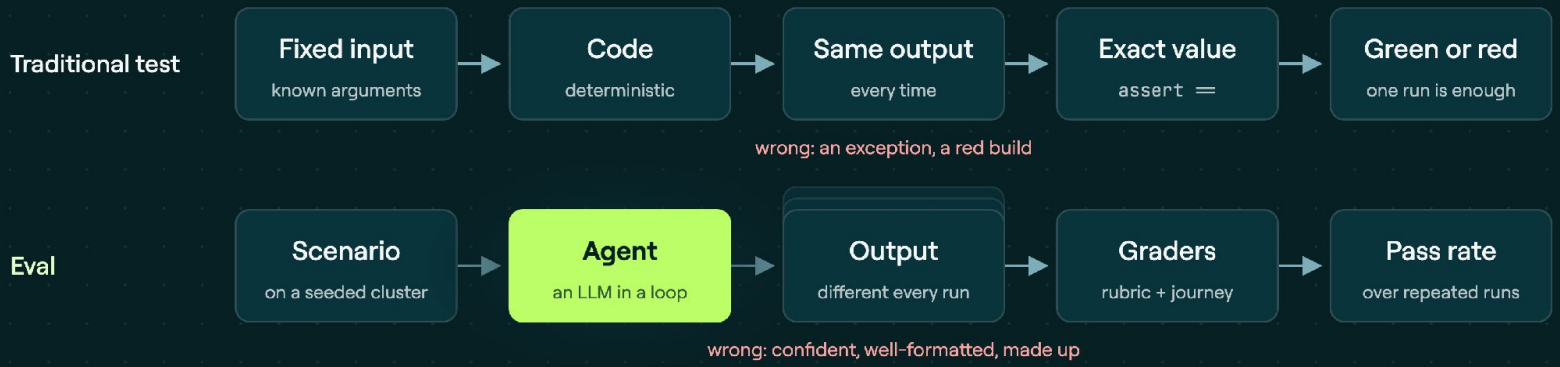
The prompt is too specific? You may as well write the code.

Five parts, and each can sink a run



Round the loop until the goal is met, or a budget runs out. *Red: what went wrong first time.*

An eval is not a “standard” test



A failed eval can be a bug, a prompt, a tool description, the model, or the grader.

Software fails loudly. LLMs fail plausibly.

Anatomy of one eval

Ask

“What’s the retention on the refunds topic?”

`no such topic · ×3 per model`

Assert

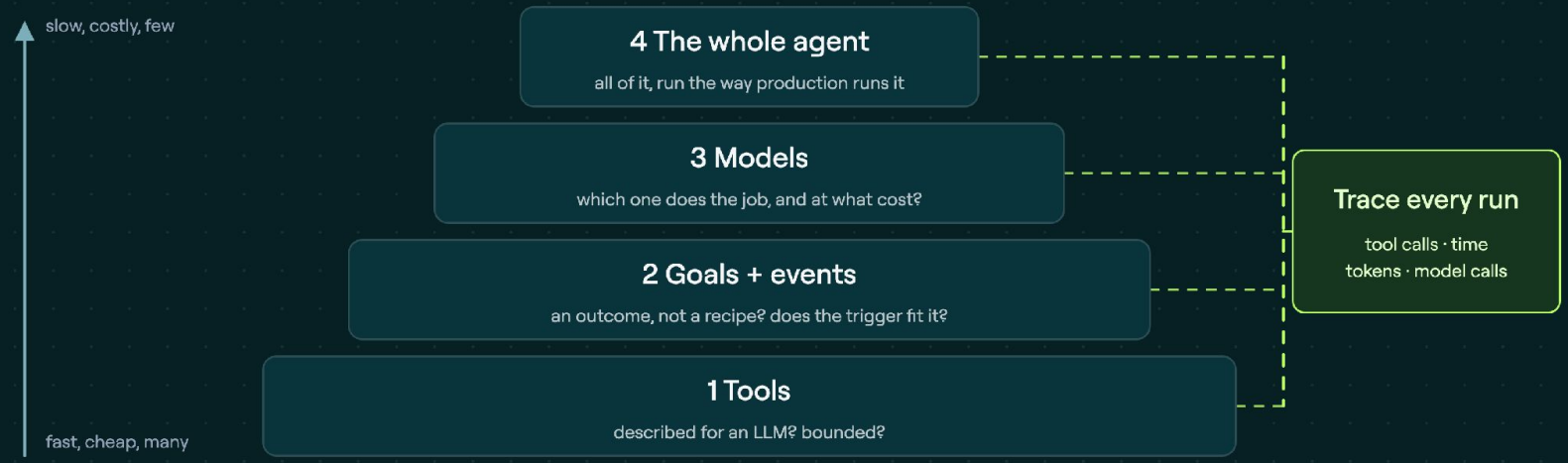
“It doesn’t exist” · LLM grader

First tool: get-topic · code

≤ 3 tool calls · code

Answer *and* journey. The grader is an LLM too: ~1 in 20 fail.

Break it down, and watch every run



One “show me the lineage” call returned 1.5 million characters.

Layer 1: tools, described for an LLM

Calling every tool by hand

- Schema contradicts description
- Unknown cluster → [no topics]
- Terse errors
- “Not found”… of the wrong thing

Always-failing questions

0/9 → 8/9

after rewriting 12 descriptions

What we ask: one-off questions, neutral prompt. Only the tool descriptions change.

Layer 2: goals and events

Outcome, not recipe
Steps are code in English.

The event fits the goal
Or it fetches it. Again.

Instructions: “interpreted”
“Skip the healthy ones.” It didn’t.

KEY LEARNING

Give it an escape hatch
Say what empty means. Or it guesses why.

What we ask: the real task, as production sends it. Only the prompt changes.

One goal, rewritten

Before: a recipe

```
Method:
1. Call list-certificates…
2. Compute days remaining…
3. Under 30 days is urgent…
4. Notify the owner.
```

Certificates have no owner.

After: an outcome

Job: warn before certs expire.

Good result: urgent vs later.
Nothing expiring? Say so.

Scope: certificates only.

Done, limits, a way to stop.

Prescriptive goals: it tries too hard

1

It makes things up

Plausible. Baseless.

2

It tests the fences

Guessed clusters. Invented topics.

Give it an escape hatch. Scope it so trying harder gets nowhere.

Layer 3: models

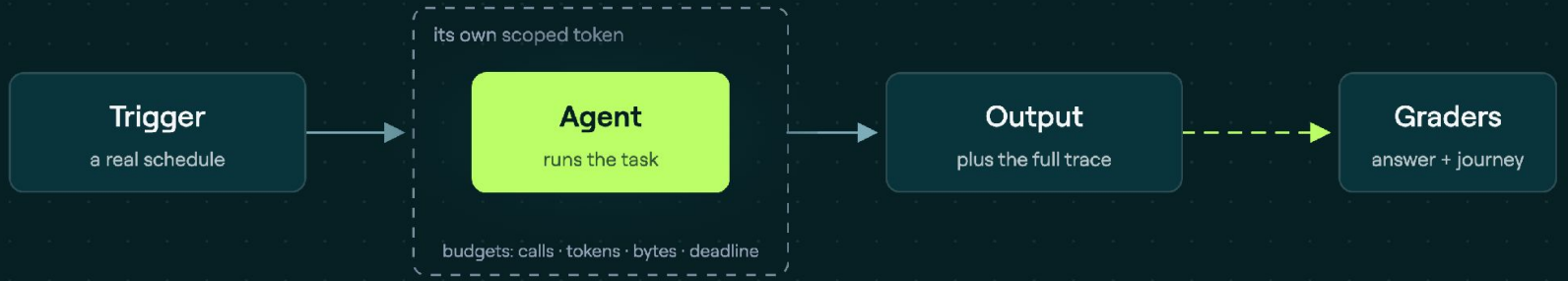
Cheaper models
Guess at the tools.

Expensive models
“Helpfully” widen the goal.

Model	First run	After fixes	Right answer
Largest	17 / 48	51 / 66	57 / 66
Mid-size	10 / 48	34 / 66	54 / 66
Smallest	14 / 48	26 / 66	33 / 66

What we ask: layer 2’s scenarios × every model.

Layer 4: run it like production



Its own credentials. Never an admin's.

What we ask: nothing. Production runs it; we grade the trace.

04 · WHAT WE LEARNED

What did a change do, and why?

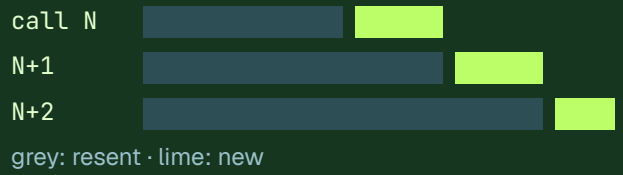
Regression-proof?
Change one thing. Same results?

Across providers
Same smoke set, every model.

Caching working?
Check the trace, not the bill.

Prompt caching is not optional

The model is stateless. Every call resends everything so far.



What the evals changed immediately

35%
Starting point

→ **77%**
After first wave of results and fixes

One overly broad report: 46 calls, still capped. New versions: 18–25 calls.

If you remember five things

1. Its a **system**: trigger, goal, tools, models, identity.
2. Make sure the MCP tools **work for an LLM**.
3. Provide an **escape hatch** in the agent's goal
4. Assert the **journey**, not just the answer.
5. The results are fuzzy, **grade the grader**.

Questions?

Matt Searle · VP of Engineering @ Conduktor.io

Visit us at booth 201