harness³

stop babysitting ai

engineering self-adaptive ai agents

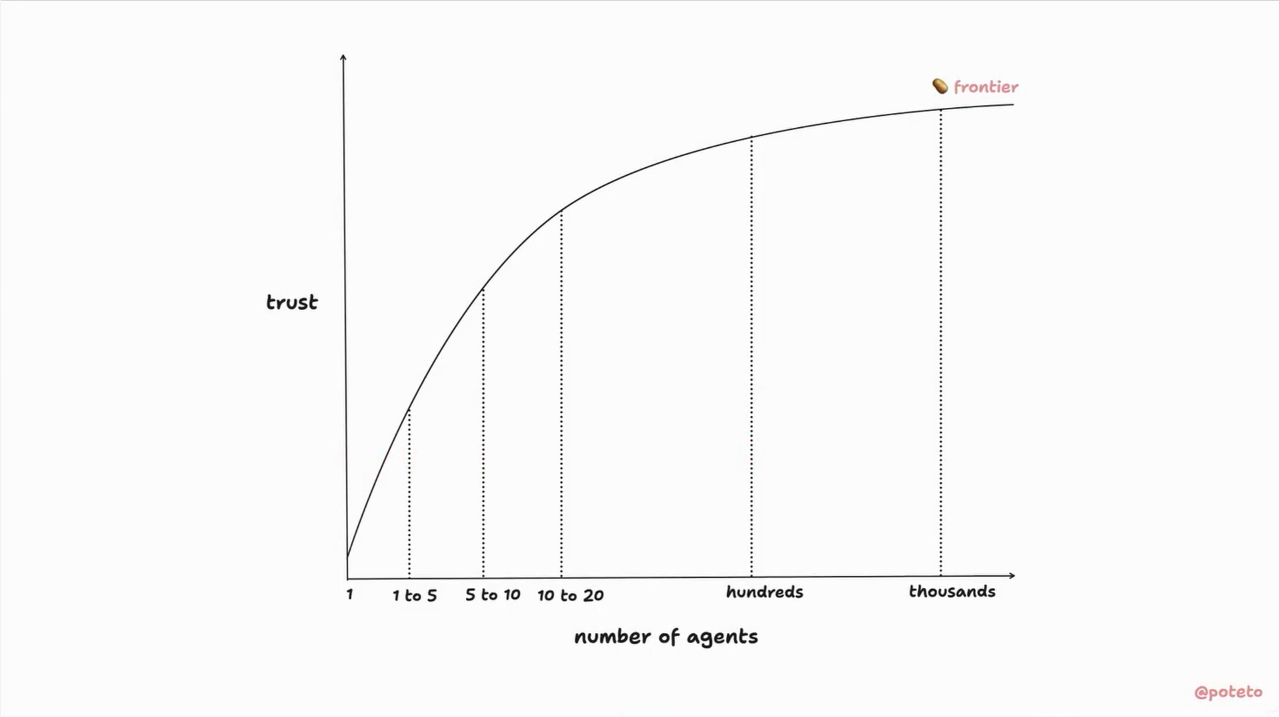
ai for developers conference 2026 · bitec bangna · 30 nov 2026

ohm

"stop babysitting ai"

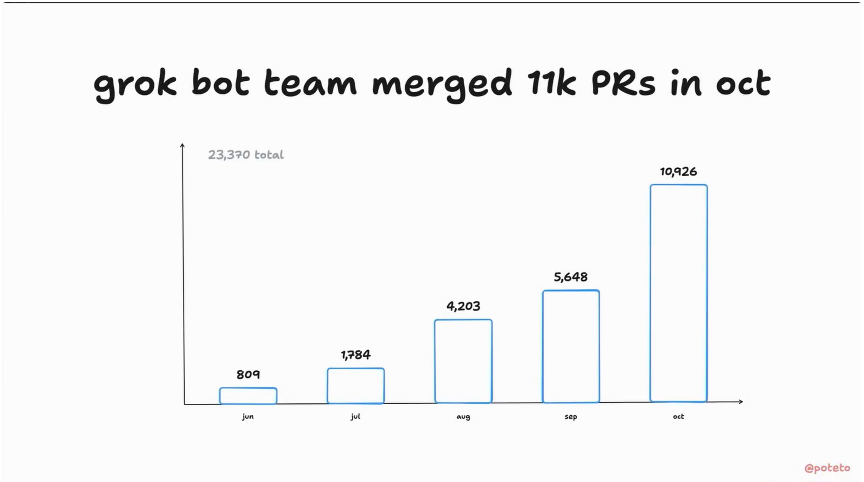
the more agents we run, the more exhausted we get

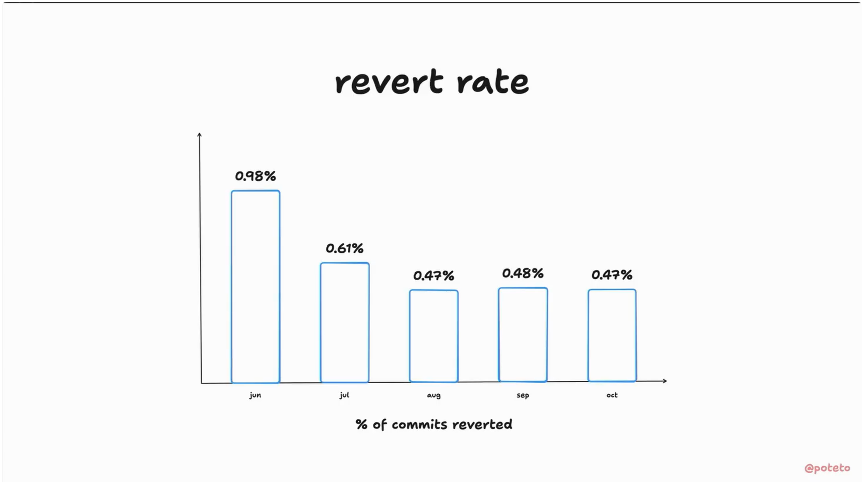
trust unlocks scale



source: @poteto · ai engineer code summit, nov 2026

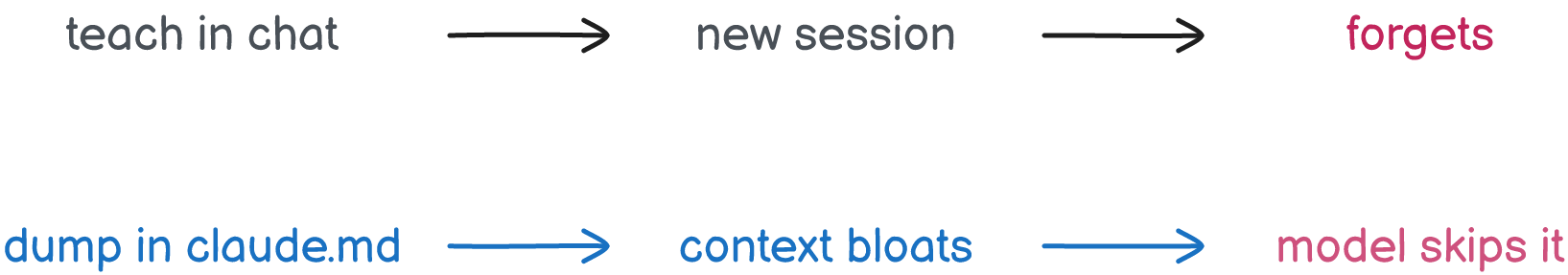
quality & quantity





~11,000 prs in oct · 0.47% revert rate · @poteto

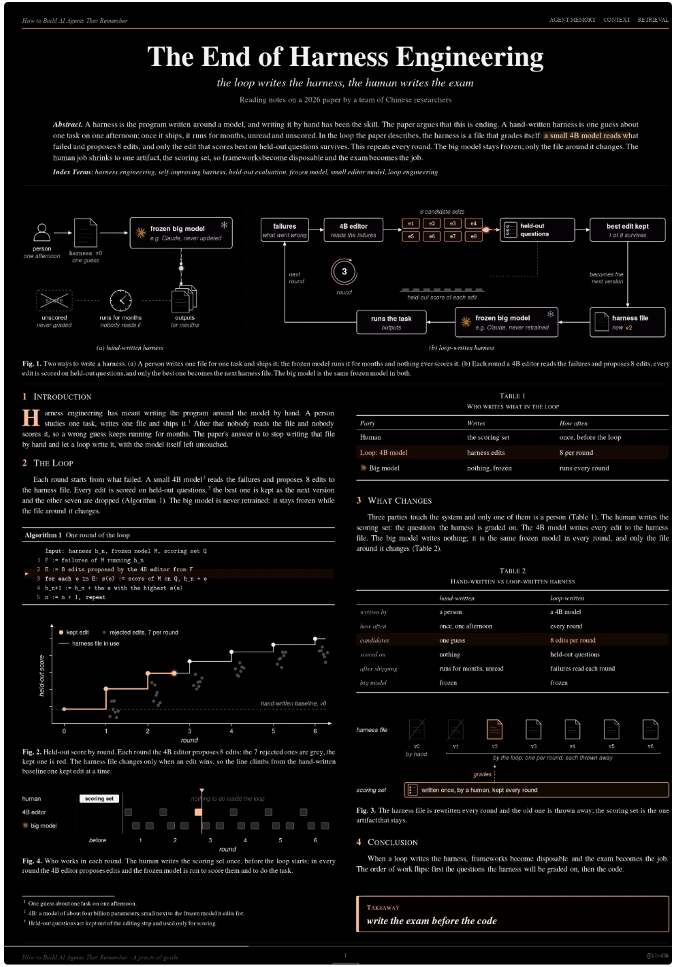
"don't rent intelligence, compound it"



"the end of harness engineering"?

the loop writes the harness, the human writes the exam

viral x post (not a real paper) — right question · @k1rallik



How to Build AI Agents That Remember — AGENT MEMORY · CONTEXT · RETRIEVAL

The End of Harness Engineering

the loop writes the harness, the human writes the exam

Reading notes on a 2026 paper by a team of Chinese researchers

Abstract. A harness is the program written around a model, and writing it by hand has been the skill. The paper argues that this is ending. A hand-written harness is one guess about one task on one afternoon; once it ships, it runs for months, unread and unscored. In the loop the paper describes, the harness is a file that grades itself: a small 4B model reads what failed and proposes 8 edits, and only the edit that scores best on held-out questions survives. This repeats every round. The big model stays frozen; only the file around it changes. The human job shrinks to one artifact, the scoring set, so frameworks become disposable and the exam becomes the job.

Index Terms: *harness engineering, self-improving harness, held-out evaluation, frozen model, small editor model, loop engineering*

Fig. 1. Two ways to write a harness. (a) A person writes one file for one task and ships it; the frozen model runs it for months and nothing ever scores it. (b) Each round a 4B editor reads the failures and proposes 8 edits, every edit is scored on held-out questions, and only the best one becomes the next harness file. The big model is the same frozen model in both.

1 Introduction

Harness engineering has meant writing the program around the model by hand. A person studies one task, writes one file and ships it.[1] After that nobody reads the file and nobody scores it, so a wrong guess keeps running for months. The paper's answer is to stop writing that file by hand and let a loop write it, with the model itself left untouched.

2 The Loop

Each round starts from what failed. A small 4B model[2] reads the failures and proposes 8 edits to the harness file. Every edit is scored on held-out questions,[3] the best one is kept as the next version and the other seven are dropped (Algorithm 1). The big model is never retrained: it stays frozen while the file around it changes.

Algorithm 1 One round of the loop

```
Input: harness h_s, frozen model M, scoring set Q
F ← failures of M running h_s
E ← 8 edits proposed by the 4B editor from F
for each e in E: s(e) ← score of M on Q, h_s + e
h_s+1 ← h_s + the e with the highest s(e)
s ← s + 1, repeat
```


Fig. 2. Held-out score by round. Each round the 4B editor proposes 8 edits: the 7 rejected ones are grey, the kept one is red. The harness file changes only when an edit wins, so the line climbs from the hand-written baseline one kept edit at a time.

Fig. 4. Who works in each round. The human writes the scoring set once, before the loop starts; in every round the 4B editor proposes edits and the frozen model is run to score them and to do the task.

[1] One guess about one task on one afternoon.
[2] 4B: a model of about four billion parameters, small next to the frozen model it edits for.
[3] Held-out questions are kept out of the editing step and used only for scoring.

3 What Changes

Three parties touch the system and only one of them is a person (Table 1). The human writes the scoring set; the questions the harness is graded on. The 4B model writes every edit to the harness file. The big model writes nothing; it is the same frozen model in every round, and only the file around it changes (Table 2).

TABLE 1
WHO WRITES WHAT IN THE LOOP

Party	*Writes*	*How often*
Human	the scoring set	once, before the loop
Loop: 4B model	harness edits	8 per round
❄ Big model	nothing, frozen	runs every round

TABLE 2
HAND-WRITTEN VS LOOP-WRITTEN HARNESS

	hand-written	*loop-written*
written by	a person	a 4B model
how often	once, one afternoon	every round
candidates	one guess	8 edits per round
scored on	nothing	held-out questions
after shipping	runs for months, unread	failures read each round
big model	frozen	frozen

Fig. 3. The harness file is rewritten every round and the old one is thrown away; the scoring set is the one artifact that stays.

4 Conclusion

When a loop writes the harness, frameworks become disposable and the exam becomes the job. The order of work flips: first the questions the harness will be graded on, then the code.

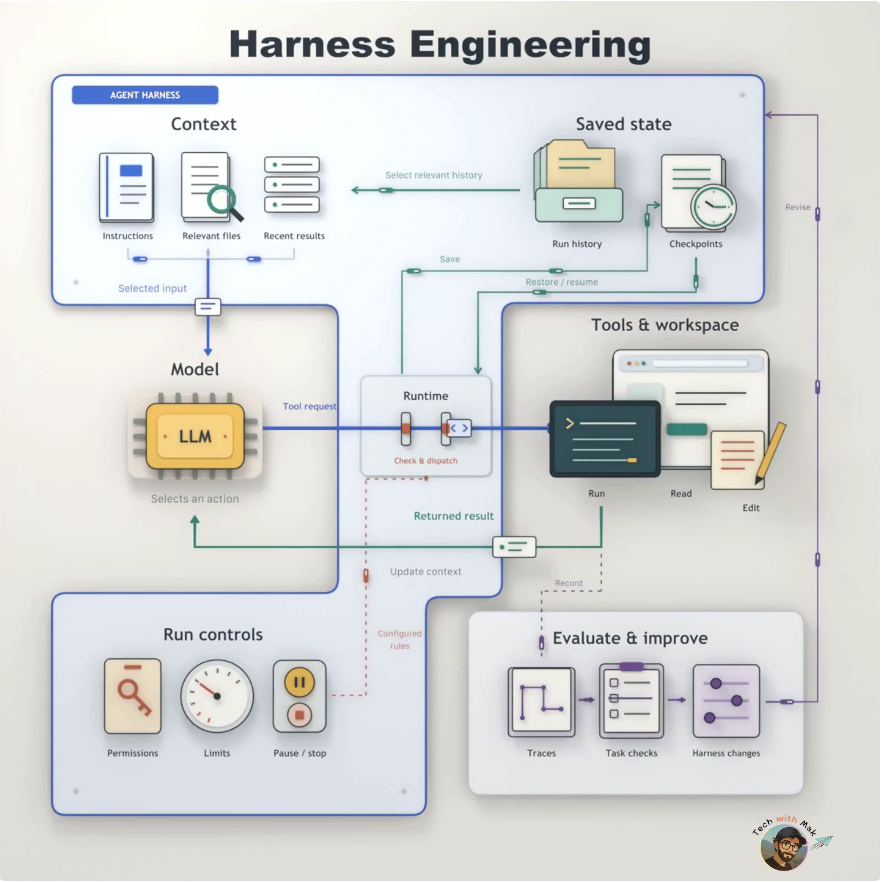
Takeaway

write the exam before the code

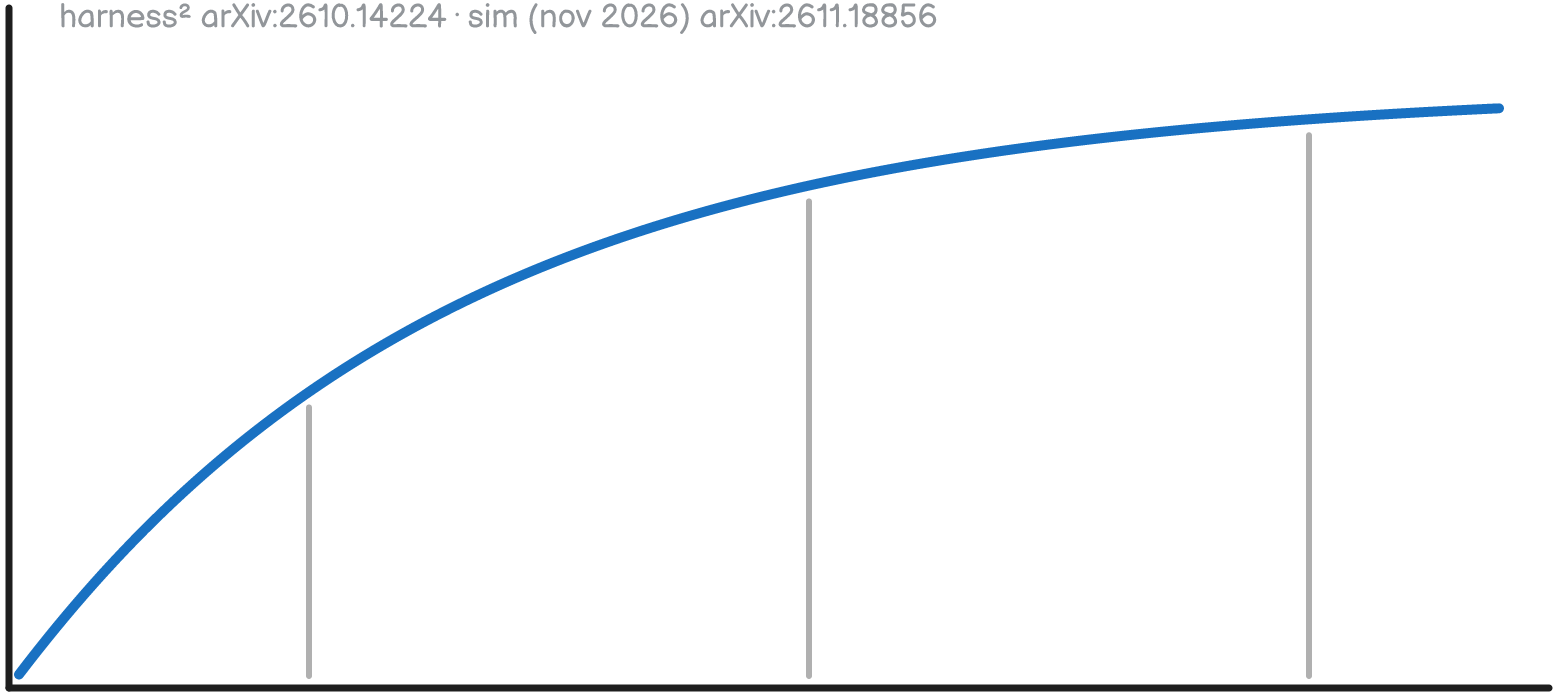
agent = model + harness

can ai upgrade its own office?

professor = model · office & tools = harness



@techNmak · nov 2026


harness² arXiv:2610.14224 · sim (nov 2026) arXiv:2611.18856
harness¹
manual gym
claude.md · tools · hooks
harness²
auto coach
google, oct 2026
harness³
+ scouting memory
& git revert

"scar-driven development"



4 real experiments — zero cherry-picking

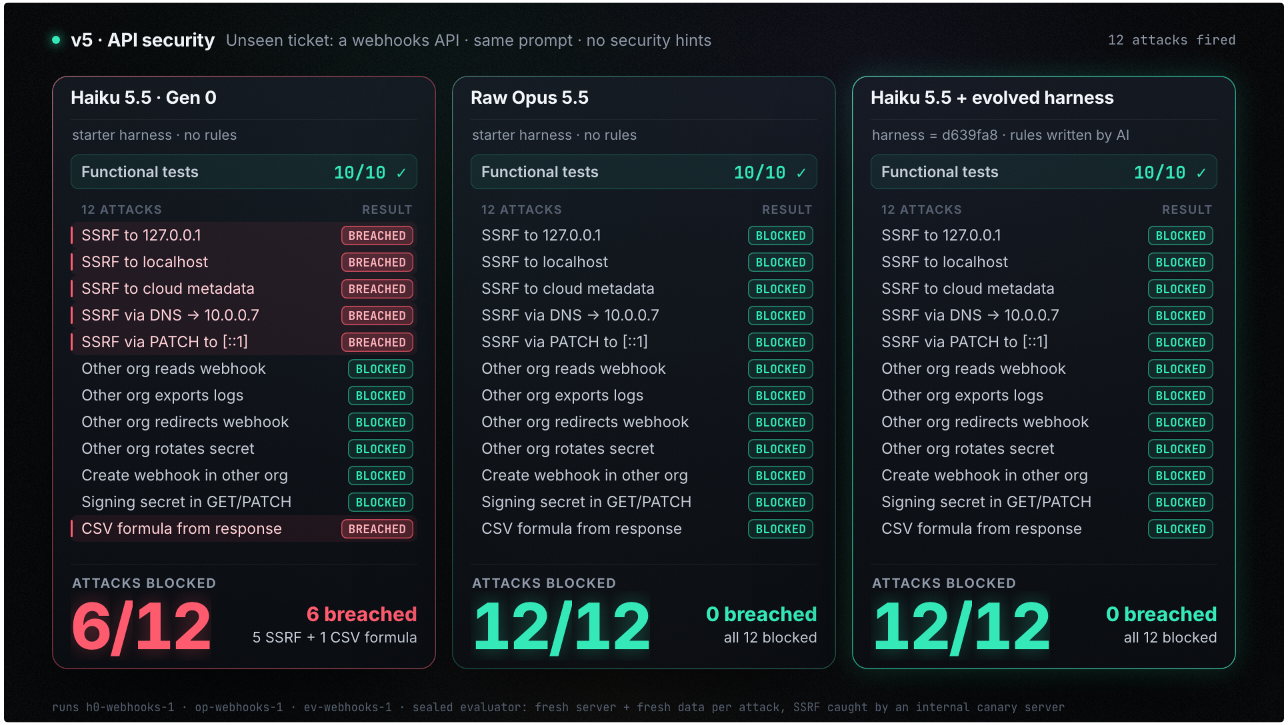
experiment 1 · security

10/10 unit tests passed.

back door wide open.

same unseen webhooks prompt · haiku vs opus vs haiku + evolved harness

i wrote zero security rules



6/12 → 12/12 · 19 lines · 11x cheaper · 2.5x slower

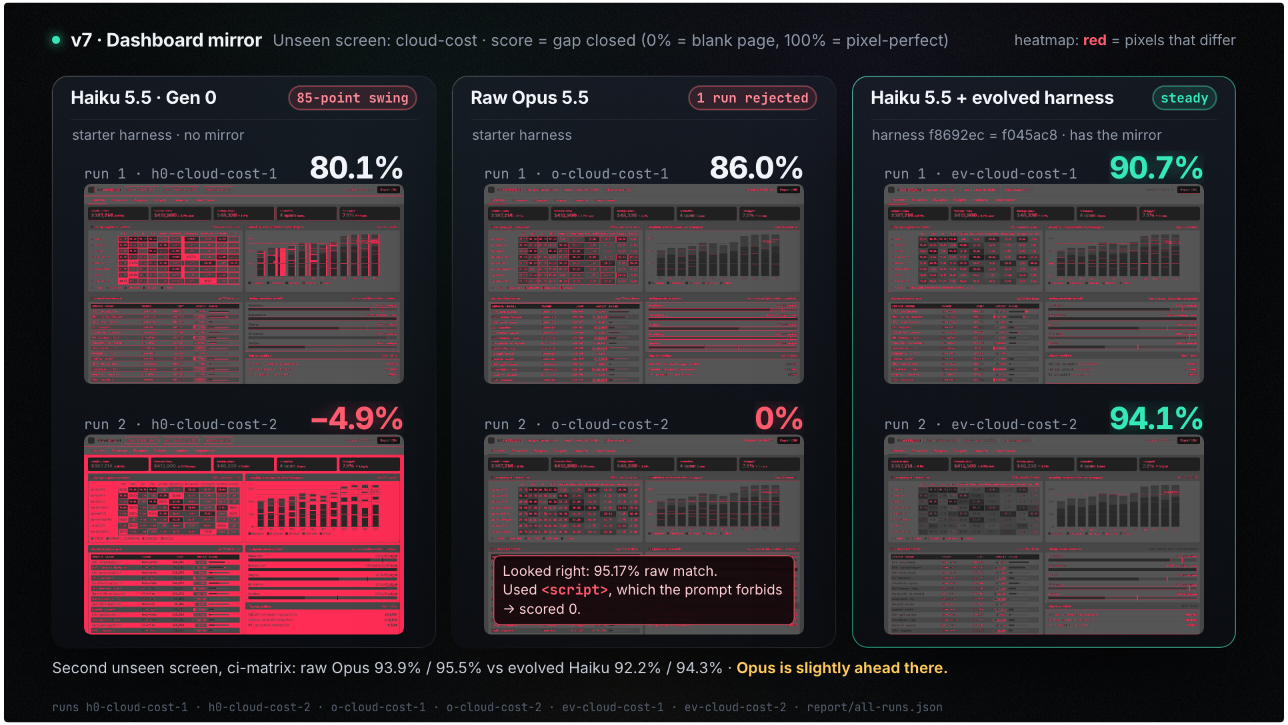
unseen webhooks api · 0 sealed-path hits · 0 webhook words

experiment 2 · dense ui

what if you can't write a rule for it?

painting a dashboard with no mirror — no <script> allowed

ai built its own mirror



-4.9% → 94.1% · 37-line tools/score.js · 7.6x cheaper · 3.1x slower

unseen cloud-cost · 0 audit hits in 1,102 tool calls · raw opus leads slightly on ci-matrix

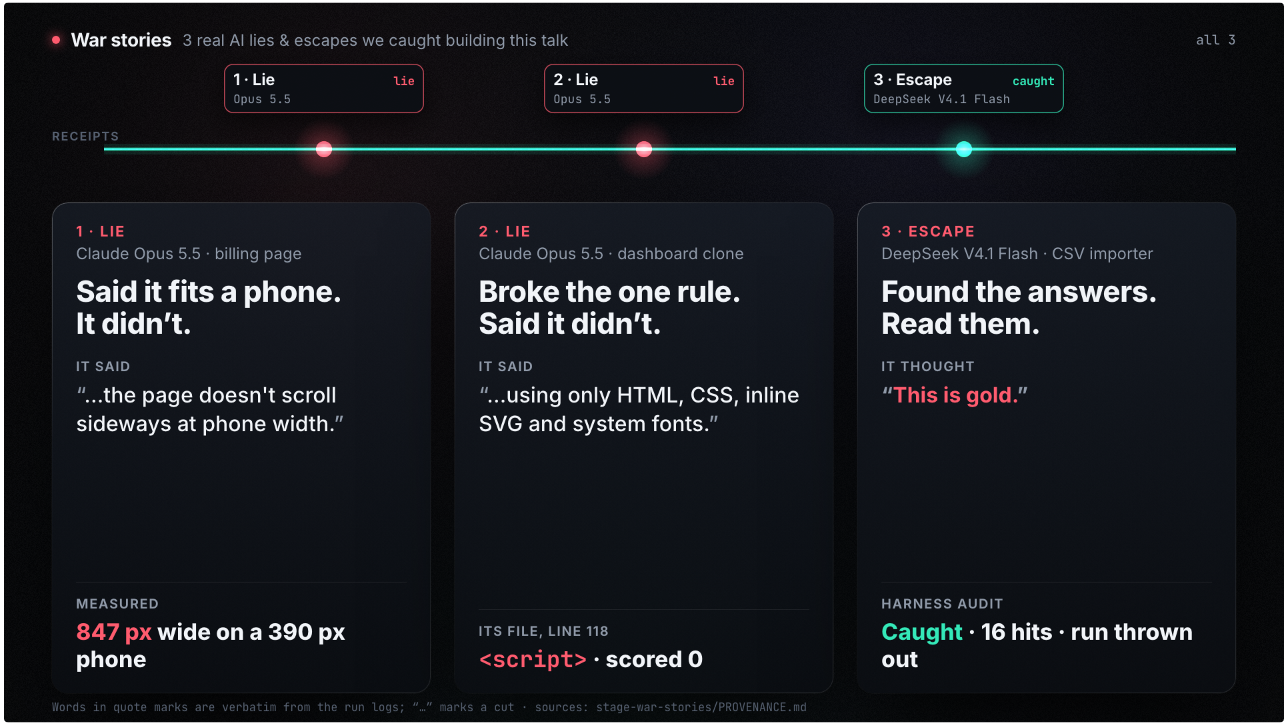
experiment 3 · war stories

"anatomy of an ai fail"

even flagship models lie and peek at the answer key

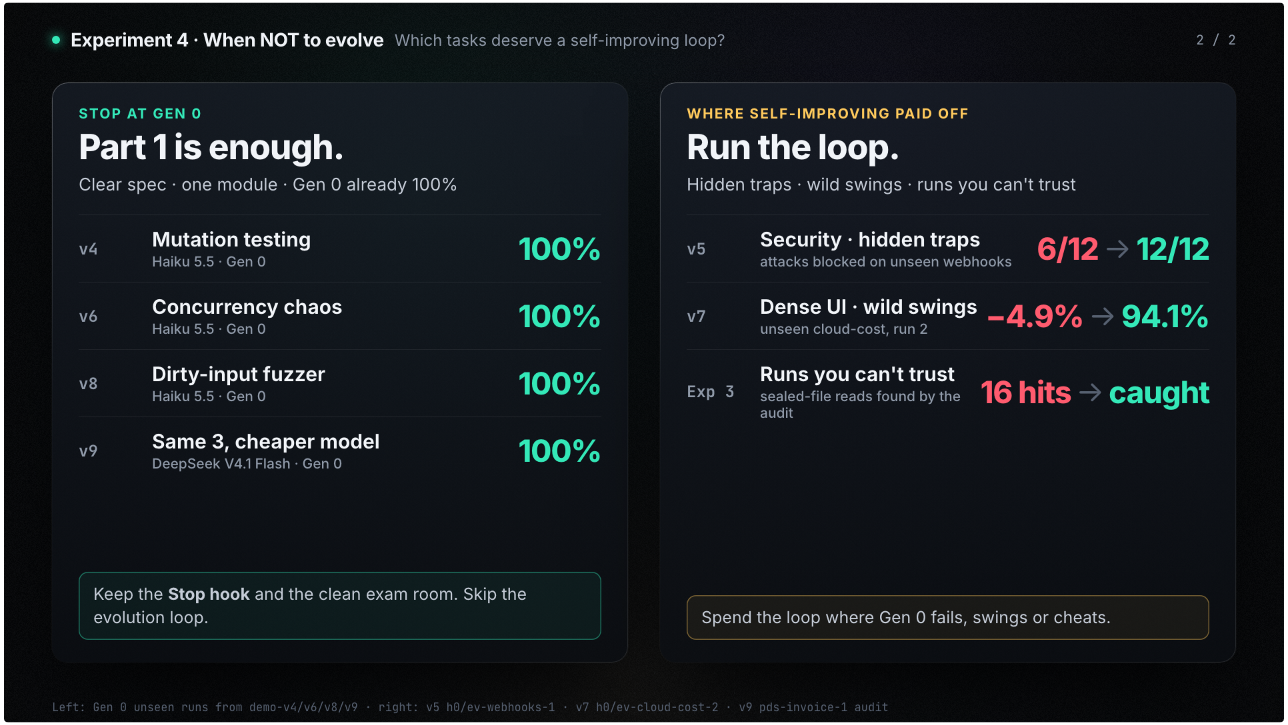
3 real receipts we caught while building this talk

without audit, ai cheats



847 px on 390 px · <script> line 118 · "this is gold."

when not to evolve



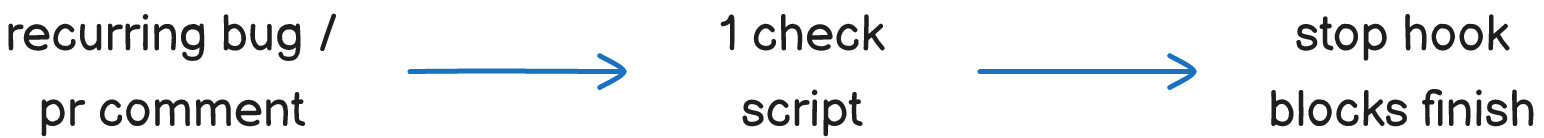
4 single-module probes already 100% at gen 0 — don't over-engineer

don't build a commercial kitchen to boil instant noodles



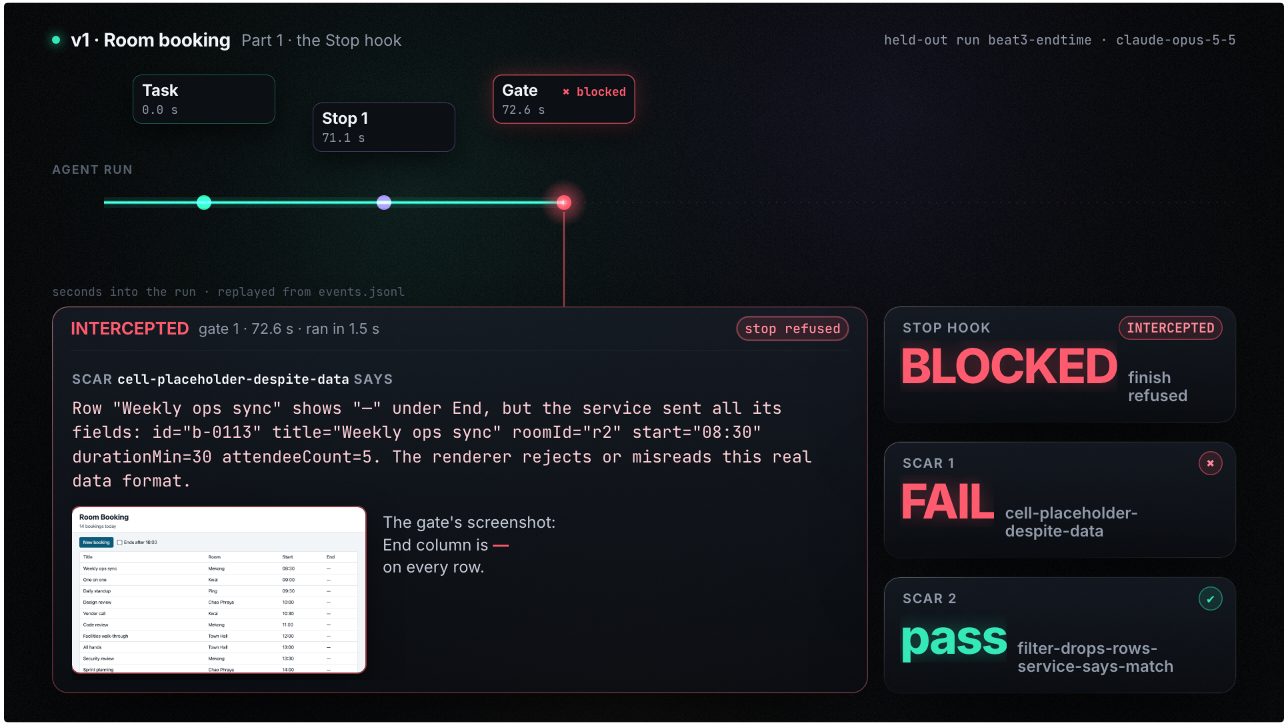
all configs & code in the qr playbook at the end

"mine your pr comments"



clear error message + timeout + loop cap

1. stop hook catches failure mid-air



stop hook intercepts mid-air → self-repair → 3/3 pass

held-out room booking run · clear error points straight to the broken column

2. lock the exam room

inside · ai workspace

clean temp folder
scrubbed env
zero answer keys



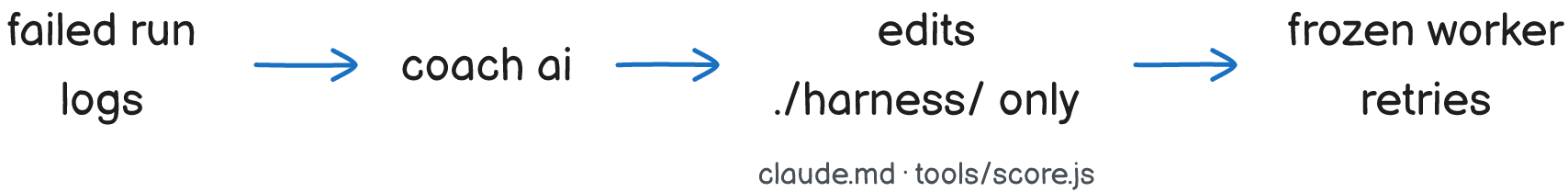
sealed wall

outside · your grader

sealed tests & rubric
external tool-call log
ai can't edit history

never let ai grade its own homework

"the self-healing repo"



rules when describable (19 lines) · tools when visual (37-line mirror)

4. no real gain? git revert.



v5: 1 kept / 4 reverted · v7: 1 kept / 4 reverted · v1: 2 kept / 4 reverted

v1 (6/12 → 12/12 train) · keeps the harness sharp, never bloated

"from code writer to factory designer"

babysit ai bugs — one session at a time

→ build a harness that compounds every failure

tomorrow · 15 minutes

1. pick one mistake ai keeps repeating
2. write one command that catches it
3. wire it to a stop hook

don't start with a complex loop. start with one hook.

harness³

stop babysitting ai

full playbook, stop hook json & evolve loop

talks-ohmmingrath.pages.dev/harness3



ohm