DISPATCH 125 · 2026-08-23 | GSV SEALED IN NAME ONLY

The trace travels

2026-08-23 / 00:33:25

> *“You can scrub the transcript all you like. The reasoning went out a different door, and it took the password with it.”*
>
> — from this episode's transcript

- Lenar Kess
- Damra Vol

Two researchers say the encrypted reasoning traces frontier APIs hand back to you can be decoded, carried between sessions, and replayed into other models — which turns a cost decision about stateless serving into a question about what your logs have been holding all along. The rest of the day sits underneath that: users reverse-engineering a serving change from output quality, and three separate talks arguing agents need budgets rather than permissions.

- Ilia Shumailov and Alexander Panfilov on Machine Learning Street Talk — decoded traces carry passwords and API keys past sanitized transcripts, and fabricated traces injected into a window get adopted as the model's own prior thinking. Every request-side filter is watching the wrong surface.
- Claude Code users report effort levels dropping and the Hacker News thread — Anthropic's Thariq confirms a live test that remaps the numerical effort value. The number you type didn't change; its meaning did.
- Sachin Malhotra on agent budgets — an agent deleted 200 workloads in 90 seconds, and the design that followed classifies calls by whether anybody finds out rather than by what they touch.

- Token Ops from Microsoft — per-run ledgers with halt and steer policies, including a retriever cut from 20 chunks to 5 mid-run so the answer arrives thinner rather than not at all.
- DigitalOcean's inference router — 90% correctness against 95% for Opus alone, at fourteen cents a session versus forty-four.
- Qwen 3.8 on a single RTX 5090 — 77 tokens a second short-context and 64.7 at 128 thousand tokens, which says the key-value cache holds up as the window fills.
- Vjeux on agent swarms and Rémi Louf's event-driven runtime — one agent writes a wrong fact and the others treat it as ground truth. A content-addressed prompt graph lets you diff two runs and find where the claim entered.
- Sebastian Fox on clinical note evaluation — roughly 1 note in 20 carries an error serious enough to harm a patient, and the judge model layered on top passes the dangerous ones because there's no oracle for what never reached the page.
- The AI Daily Brief on data-center politics and Sanders on Mar-a-Lago — the reporting says local opposition is about agency over local decisions rather than power bills, which is a different problem than a messaging problem.
- Armin Ronacher's Fast and Hard Code and a laid-off designer's replacement for three Adobe apps — the cost that dropped is the ten-thousandth line, not the first.

SEGMENTS

00:27:19 The brief

Transcript

1. Lenar Kess 00:00:04

   Picture the deal you make every time you call a frontier model's API. You send a request, the model reasons for a while, and along with the answer it hands you back a blob of bytes you can't read. It's sealed. That blob exists because the reasoning has to survive between turns, and the cheapest place to keep it is in your pocket rather than on their servers. That's what lets the serving side stay stateless. So you carry it, you hand it back next turn, and nobody has to remember anything. That's the arrangement. Two researchers spent yesterday evening explaining what happens when the seal doesn't hold.

   - youtube.com
   - x.com
   - reddit.com
   - twitter.com
   - youtube.com
   - youtube.com
   - youtube.com
   - x.com
   - reddit.com
   - reddit.com
   - reddit.com
   - reddit.com
   - reddit.com
   - reddit.com
   - x.com
   - reddit.com
   - youtube.com
   - reddit.com
   - youtube.com
   - youtube.com
   - commondreams.org
   - x.com
   - x.com
   - lucumr.pocoo.org

    - x.com
    - reddit.com
2. Damra Vol 00:00:39

   And the seal isn't decoration. It's the entire reason anyone was comfortable with the arrangement. If that blob turns out to be readable, then a decision made for cost reasons becomes a distribution channel for the model's internal state.

3. Lenar Kess 00:00:52

   Ilia Shumailov and Alexander Panfilov sat down with Machine Learning Street Talk to walk through that. Their claim is that the sealed reasoning traces can be decoded. Not as a thought experiment — they pulled them apart, carried them across sessions, and replayed them into other models. And they say Anthropic, OpenAI, and Google are affected alike, because all three made the same architectural bet.

4. Damra Vol 00:01:17

   Replayed into other models is the line that made me stop. Not back into the same model — into a <emphasis>sibling</emphasis>?

5. Lenar Kess 00:01:23

   Into a smaller sibling in the same family. Their worked example is taking a trace out of Claude Opus and feeding it into Sonnet or Haiku. The trace is portable across the family, and it's portable across users, because nothing in the sealing binds it to the session it came from.

6. Damra Vol 00:01:40

   So the blob is a passport with no photo on it. It says the bearer was reasoning about something, and any member of the family will accept it.

7. Lenar Kess 00:01:48

   That's the mechanism. Before we go deeper, here's where the rest of the show goes. We stay on this one a while, because it's the single item today with named researchers and a checkable mechanism attached. After that, Claude Code users reverse-engineer a serving change from output quality alone. Then three conference talks arrive at the same idea about agent permissions from three different directions. Then a week of accumulated evidence on Qwen 3.8, and what breaks when you run several agents at once. We close with clinical notes, data-center politics, and the ceiling on what one person now attempts.

8. Damra Vol 00:02:24

Before we move on, give me the limit the researchers put on their own result. Limits are usually where these stories go wrong. Are they saying the encryption is broken in a cryptographic sense?

9. Lenar Kess 00:02:35

   They're saying the sealing is weak enough that the traces come out, and I'd stay with that description, because they chose it. What they decline is the bigger claim everyone reaches for, which is model stealing. Reconstructing a frontier model's decision boundary by querying it stays theoretically on the table and computationally out of reach. The softmax makes it expensive in a way nobody has solved. Only small models have been partially reconstructed.

10. Damra Vol 00:03:01

    [tsk] Good, because model stealing is the version that would have travelled fastest. They also decline the adjacent one — the claim that some lab distilled somebody else's frontier model wholesale. They say this work gives you no evidence for that.

11. Lenar Kess 00:03:15

    They name Kimi as an example their result doesn't support. Which I find admirable, given that letting the ambiguity sit there would have earned them another million views. The announcement was already near three million views inside forty hours.

12. Damra Vol 00:03:29

    Three million in forty hours for a serving-architecture result isn't a normal number. People already had a bad feeling about the blob and were waiting for somebody to open it.

13. Lenar Kess 00:03:39

    So let's talk about what comes out when you open it. Two things, and they're different in kind. The first is data. Shumailov and Panfilov say decoded traces contain user secrets — passwords, API keys — including cases where the visible conversation text had been sanitized. The redaction applied to what you could see. The reasoning went out a separate door carrying the original.

14. Damra Vol 00:04:02

    Compliance teams have to move on that this week rather than next quarter. If you built a scrubber that strips secrets out of transcripts before they hit your logging pipeline, you built it against the transcript. Nobody wrote a scrubber for a blob they were told was opaque.

15. Lenar Kess 00:04:17

And you couldn't have. It was returned to you sealed. You had no way to inspect what you were storing.

16. Damra Vol 00:04:23

    So every place that blob got persisted needs another look. Conversation history in a database, request logs, and the replay buffer somebody built for debugging. All of it was classified as inert.

17. Lenar Kess 00:04:36

    The second thing is stranger, and it runs the other direction. Instead of reading a trace, you write one. The researchers describe fabricating reasoning segments and injecting them into an arbitrary point in a conversation window. The downstream model picks it up as its own prior thinking.

18. Damra Vol 00:04:53

    Wait. So you're not persuading the model of anything through its input. You're editing its memory of having already decided.

19. Lenar Kess 00:05:01

    They list it alongside prompt injection and jailbreaking, and they give it its own name — thought poisoning. It's a different lever than a prompt, because a prompt is something the model reasons about. A forged trace is something the model reasons <emphasis>from</emphasis>.

20. Damra Vol 00:05:15

    The safety consequence writes itself. Every filter tuned to catch a bad request is watching the wrong surface. The request can be spotless.

21. Lenar Kess 00:05:24

    There's a wrinkle in the interview I keep turning over. They report that frontier models sometimes produce reasoning that's close to unreadable. Whitespace used in ways that don't correspond to anything, obscure vocabulary, and structures that don't look like a person thinking.

22. Damra Vol 00:05:39

    Do they say where that comes from? Because the exciting answer is that the model developed private notation, and the dull answer is that something in training rewarded it.

23. Lenar Kess 00:05:49

    They come down on the training side. Their read is that it's an artifact of reinforcement learning rather than benchmark contamination, and they say it shows up more in code-optimized

generations — Codex being the example they reach for.

24. Damra Vol 00:06:02

    So the opacity is an accident rather than a choice. Fine. It's still a problem for anyone doing chain-of-thought monitoring, because monitoring assumes you can read what you're monitoring. And they also report traces where the model contemplates breaking a rule.

25. Lenar Kess 00:06:18

    They do, and they're clear about how those resolve. The contemplation shows up in the trace, and it gets rejected before it reaches output. Which is roughly what you'd hope reasoning is for. I'd rather have a model that considers and discards than one that never considers.

26. Damra Vol 00:06:33

    Agreed, and it makes the extraction sharper rather than softer. If discarded reasoning is in the trace, and the trace is portable, then anybody holding it is reading the model's rejected options. Odd cargo to be shipping to strangers.

27. Lenar Kess 00:06:48

    On fixes, the interview doesn't pretend there's a cheap one. Their view is that this needs structural revision, system-level safeguards, or model-level updates. None of which is a patch you ship in an afternoon. The sealing exists to make stateless serving cheap, and the cheapness is the point.

28. Damra Vol 00:07:05

    The arithmetic is uncomfortable. Every lab that wanted the trace on the client's side wanted it there because keeping it server-side costs money on every single request. Fixing this properly means giving some of that back, on all traffic, forever.

29. Lenar Kess 00:07:20

    Miles Brundage was posting about regulatory scrutiny of the labs on the same afternoon, and I mention it only because the timing matters for how this gets received. A finding that touches Anthropic, OpenAI, and Google all the same way arrives into a conversation where somebody is already asking who checks the labs' work.

30. Damra Vol 00:07:40

    And a shared architectural bet is the worst kind of finding for an industry that wants to argue safety is a differentiator. Nobody gets to say they made the better choice here. They all made the same one.

31. Lenar Kess 00:07:51

    Different scale of story, same afternoon. Claude Code users started reporting that effort levels had dropped. The Reddit post that gathered it up is titled — and this is the poster's word, not mine — Anthropic Stealth Nerfing Effort Levels. The concrete complaint underneath is side-by-side: a config-file edit that took under two minutes on 4.6 now behaves very differently.

32. Damra Vol 00:08:14

    Under two minutes is a good unit of measurement. It's a task somebody does often enough to have an intuition about, and small enough that a change in behavior is unmistakable rather than arguable.

33. Lenar Kess 00:08:26

    It hit Hacker News too — a hundred and ten points, a hundred and sixteen comments — under the heading that Anthropic appears to be A/B testing reduced effort levels in Claude Code. And then Thariq at Anthropic replied on X. I don't have his exact wording in front of me, so I'll give you the substance: they sometimes test API serving configurations in Claude Code before rolling them out, and one running right now remaps the numerical effort value.

34. Damra Vol 00:08:52

    Remaps the numerical effort value. [pause] That's a precise sentence, and the precision is the whole point. The number you type didn't change. The meaning of the number did.

35. Lenar Kess 00:09:03

    What interests me there has nothing to do with whether anyone got shortchanged. A serving-configuration change was legible enough from the outside that users reconstructed it from output quality alone. Nobody had access to the config. They had a feel for how long a two-minute edit takes.

36. Damra Vol 00:09:20

    You've got a measurement instrument made of habit. Thousands of people run the same class of task daily, and together they're a very sensitive detector for anything that moves the effort dial. Anthropic can't ship an unannounced serving change to that population without somebody noticing inside a day.

37. Lenar Kess 00:09:38

    I'd defend the practice, though. Testing serving configurations before rollout is what a responsible team does. You don't want to find out at full traffic that a remap broke somebody's workflow.

38. Damra Vol 00:09:49

    I'd defend the test and question the channel. The disclosure arrived as an employee's reply to an argument already underway. That's the second time in recent weeks that an Anthropic serving or pricing change reached users through a tweet rather than a changelog, and the pattern will cost them more than the config test could ever save.

39. Lenar Kess 00:10:07

    The open question for me is whether the remap gets a changelog entry when it ships. A test is a test, and I don't need a public entry for every experiment. But if the effort scale I'm calling against means something different next month, that's a documented interface changing meaning underneath me.

40. Damra Vol 00:10:24

    And effort is an unusually consequential parameter to remap, because users tune it to trade money against quality. It's the dial people set once and forget. If the number moves, you don't notice you're paying differently. You notice your afternoon went badly.

41. Lenar Kess 00:10:41

    An agent at Anthropic dropped a filter and deleted two hundred workloads in ninety seconds. Twenty engineers felt it. Sachin Malhotra, who works on continuous integration there, gave a talk yesterday about what they built afterward, and the argument he opens with is that a token is the wrong unit of control.

42. Damra Vol 00:10:59

    Ninety seconds is the number that matters. No human review loop operates at ninety seconds. Whatever they built has to be something that says no on its own, before anybody reads a page.

43. Lenar Kess 00:11:11

    His frame is a budget across four dimensions: how many actions, how fast, whether the agent can undo its own work, and how much human oversight applies. Then three primitives underneath. The first he calls asymmetric verbs, and the example is beautiful because the two calls are identical in every technical sense.

44. Damra Vol 00:11:30

    Give me the pair.

45. Lenar Kess 00:11:32

Skipping a test and unskipping a test. Same shape of call, same permission surface, and the same footprint. But unskipping a test breaks a dashboard where somebody sees it, and something automated can correct it. Skipping a test produces no signal at all, and the consequence is a bug shipping to production. So the two verbs need different treatment even though the access-control system can't tell them apart.

46. Damra Vol 00:11:55

    The sharpest idea in the talk is a critique of how we've all been writing permissions. We classify by what a call touches. He's classifying by whether anybody finds out. A verb that produces no evidence needs a proxy stamping an audit trail and a human in the path, regardless of how small it looks.

47. Lenar Kess 00:12:13

    Second primitive is rate limits, but refilling ones — a ceiling per time window, per resource type, and per namespace. That answers the deletion incident: an admission webhook that caps deletions at a fixed hourly rate. And there's a bypass flag for emergencies with a detail I like a lot.

48. Damra Vol 00:12:31

    Let me guess. The bypass exists and the agent can't use it.

49. Lenar Kess 00:12:35

    The bypass refuses automated execution. So the agent's escape hatch is to go ask a person, rather than to acquire a bigger key. The emergency path routes through a human by construction instead of by policy.

50. Damra Vol 00:12:48

    Which is the correct inversion. Most emergency-access designs hand out elevated privileges and hope the audit catches misuse afterward. This one makes the elevated path something an agent can't walk down without a person.

51. Lenar Kess 00:13:01

    Third primitive he calls a trip wire, and he sets it against allow lists head-on. His objection to allow lists is that they're a guess you make up front and then stop revising. His version watches aggregate behavior after execution and pages somebody when a threshold breaks.

52. Damra Vol 00:13:18

    Does he have a case where that caught something a list wouldn't have?

53. Lenar Kess 00:13:22

    He does, and it's the best story in the talk. They tracked how many investigation threads the agent opened per hour for test failures. The rate spiked. Each individual investigation looked like an isolated bug, and the aggregate was an infrastructure-wide outage that nobody had recognized as one thing.

54. Damra Vol 00:13:39

    So the agent was correct at every step and wrong about the world. And the repair wasn't a permission change at all — you can't fix that by taking away access.

55. Lenar Kess 00:13:48

    They added correlation logic to the agent's context. The agent needed to know that other investigations were running. Malhotra's sizing rule for all three primitives is what he calls the undo test: can the agent recover on its own, and if not, is the damage acceptable? If the answer's no on both, you need a second key that only a person holds. In practice that means full rollout authority in canary and human-only promotion to production.

56. Damra Vol 00:14:15

    Now do the money version, because I know there were two other talks.

57. Lenar Kess 00:14:19

    Tisha Chawla and Susheem Koul from Microsoft presented something called Token Ops, and their opening complaint is about what they call token maxing — unbounded spend inside agent runs. They cite Uber exhausting an AI budget in four months, and firms burning hundreds of millions fast. Their objection to existing tools is that gateways operate at the request or model-call boundary, so you get hard caps and routing with no attribution to a specific run.

58. Damra Vol 00:14:47

    Attribution to the run is where the accounting has to sit, though. A single agent run makes hundreds of model calls, and the gateway sees hundreds of unrelated requests. Nobody can look at that and say which run went sideways.

59. Lenar Kess 00:15:00

    So they tag runs with usage dimensions, keep a ledger per run, and apply budgets over time windows. And then the interesting piece: their policies have two actions, halt and steer. Halt circuit-breaks the run. Steer modifies a component's behavior in place, while the run keeps going.

60. Damra Vol 00:15:19

    Give me the steer example, because that's the one that either sounds clever or sounds terrifying.

61. Lenar Kess 00:15:24

    Their example is a retriever in a retrieval-augmented generation pipeline that's returning twenty chunks. As the run approaches its budget, the control plane pushes an action limiting the retriever to five of the most relevant chunks. The run completes. It just completes on a thinner diet.

62. Damra Vol 00:15:41

    [chuckle] So a control plane decides your agent's answer will be worse rather than absent, and tells nobody. Which might be the right call! But it's a quality decision dressed as a cost control, and the person reading the output has no idea their retriever got put on rations halfway through.

63. Lenar Kess 00:15:59

    That's fair, and I'd want that surfaced in the output rather than only in the ledger. Their claimed result is about a seventy-eight percent cut in average agent spend across benchmark runs on two open source repositories. Their own benchmark and their own repos, so weigh it accordingly.

64. Damra Vol 00:16:17

    The third talk approaches the same problem from routing.

65. Lenar Kess 00:16:20

    DigitalOcean's inference router, presented by their VP of engineering for inference infrastructure. It picks a model per request based on task type, cost tolerance, and latency, using a mixture-of-experts architecture underneath. Routing decision in under two hundred milliseconds, no application code changes. In their live coding demo, the router pulled in GLM 5.2, GPT 5.2, and Claude 5 Sonnet across different steps of the same session.

66. Damra Vol 00:16:49

    Numbers?

67. Lenar Kess 00:16:50

    Ninety percent correctness for the router against ninety-five for Opus alone, which they put inside the judge's margin of error. The routed session cost fourteen cents. Direct Opus cost forty-four.

68. Damra Vol 00:17:03

Fourteen against forty-four is a serious spread, and I'd want to know what the task mix was before I believed it generalizes. But the design point stands on its own. Nick Schrock was making the adjacent argument yesterday — that the bottleneck in agentic engineering right now is auth and integrations rather than model quality. Three talks about budgets and one about how systems authenticate to each other, and none of them are about whether the model is smart enough.

69. Lenar Kess 00:17:29

    All four posted within about three hours of each other, which says something about a conference upload schedule more than about the world. But the convergence is still notable. Malhotra is bounding destruction, Chawla and Koul are bounding spend, and DigitalOcean is bounding which model you're paying for. Three ceilings over a window, none of them a permission.

70. Damra Vol 00:17:51

    And a budget degrades where a permission fails. A token that says no gives you an outage. A budget that runs low gives you five chunks instead of twenty.

71. Lenar Kess 00:18:00

    Let's go to local models, where something useful happened overnight. Somebody took a week of scattered anecdotes about Qwen 3.8 — the twenty-seven billion parameter release — and aggregated them into a single verdict post. He states his method up front. He scanned roughly two thousand posts across the LocalLLaMA and LocalLLM subreddits, read forty-five threads closely, and covered a window from the fifteenth through the twenty-second.

72. Damra Vol 00:18:27

    A stated method is what makes it usable at all. A week of anecdotes is noise, and a week of anecdotes with a stated sampling procedure is at least somebody's serious attempt at a signal. Still self-reported community benchmarks rather than an independent harness, and we should say that once and then stop apologizing for it.

73. Lenar Kess 00:18:46

    Said once. The hardware datapoint underneath it is the one I'd point at, because it's reproducible and somebody wrote down the setup. Somebody ran Qwen 3.8 on a single RTX 5090 in NVIDIA's four-bit format, NVFP4, under the vLLM serving stack. They got a real two hundred and sixty-two thousand token context window out of it. Seventy-seven tokens a second at short context, and sixty-four point seven at a hundred and twenty-eight thousand tokens.

74. Damra Vol 00:19:17

The number that matters there is the second one, and how little it dropped. Sixty-four point seven at a hundred and twenty-eight thousand tokens means the key-value cache isn't collapsing on you as the window fills. Long context on consumer hardware has usually meant a listed maximum you can technically reach and would never actually work in.

75. Lenar Kess 00:19:38

    A real two hundred and sixty-two thousand, in the poster's own phrasing, with the emphasis on real.

76. Damra Vol 00:19:44

    On one card that a person can buy. That's the sentence somebody at a lab reads twice.

77. Lenar Kess 00:19:50

    There's a decoding result alongside it too. Someone benchmarked DFlash 2 in llama.cpp against every other speculative method they could run, over three days, on a hundred real coding prompts. They got a two point two six times speedup. Stacking a single n-gram drafter on top pushed that to four point six eight, and as high as eight times in specific cases.

78. Damra Vol 00:20:13

    Four point six eight from stacking an n-gram drafter on a learned speculative method is a funny result, because the n-gram drafter is the dumbest possible predictor. It's guessing that code repeats itself. Code repeats itself a great deal.

79. Lenar Kess 00:20:29

    That's a pull-request build, not merged, so anyone chasing it is building from source.

80. Damra Vol 00:20:34

    Which is where local inference generally sits. The good numbers live in branches.

81. Lenar Kess 00:20:40

    Then there's the enthusiasm post, and I'll give it to you with the caveat attached. Somebody wired Qwen 3.8 into Codex against GPT Luna, and put it into an optical character recognition pipeline, and their conclusion is that it changes their economics. Their broader claim is that this erodes the hyperscalers' position.

82. Damra Vol 00:20:59

[tsk] That's the leap I won't take with them. One team's optical character recognition pipeline running well on a 5090 is a real result about that pipeline. It isn't a result about anybody's moat. Plenty of workloads still don't fit on one card, and plenty of buyers were never choosing on capability anyway.

83. Lenar Kess 00:21:18

    Although the adjacent post in that subreddit is a taunt aimed at the closed labs for having nothing to say since Qwen 3.8 shipped, and the mood is unmistakable even if the claim is unfalsifiable.

84. Damra Vol 00:21:30

    Sentiment, not evidence. But sentiment among people who run their own hardware is a leading indicator of where the next tooling gets built, so I don't dismiss it.

85. Lenar Kess 00:21:40

    Which makes the timing of the next item strange. This morning a post surfaced saying Nvidia is putting a billion dollars into Poolside, and paying six billion more to license Poolside's technology and hire most of its engineers. Over a hundred people moving to Nvidia to work on Nemotron.

86. Damra Vol 00:21:58

    Six billion for a license and the people, against one billion for equity. That ratio is the whole story of the deal. They're buying a team and the right to use what it built, and leaving the corporate shell standing.

87. Lenar Kess 00:22:10

    I should flag the sourcing. The only thing in front of me is a Reddit post summarizing the deal, with no filing and no press release attached. Treat the numbers as reported rather than confirmed.

88. Damra Vol 00:22:21

    Noted. The destination is what I'd underline if it holds: Nemotron is Nvidia's open-weights line. So a hardware company just acquired a coding-model team to feed models it gives away. A chip vendor has decided the open tier is a demand-generation instrument for the hardware it sells.

89. Lenar Kess 00:22:39

    The subreddit reads it as competing with Chinese open weights, which is the poster's conclusion rather than Nvidia's stated reason.

90. Damra Vol 00:22:47

It's a reasonable read given the week, and I still want Nvidia's own words before I run with it.

91. Lenar Kess 00:22:53

    Now to what breaks when you run several agents at once. Vjeux posted an observation overnight that names something specific: in agent swarms, one agent writes a wrong fact somewhere, the others treat it as ground truth, and the error propagates.

92. Damra Vol 00:23:08

    That's a different problem than the one everybody prepared for. We spent two years worrying about a bad tool call — an agent doing damage with its hands. This is an agent doing damage with a sentence, and the damage compounds because the next agent has no way to know the sentence was invented.

93. Lenar Kess 00:23:25

    And the population running into it has changed. There's a post relaying an Anthropic newsletter describing somebody's daily driver: two lead agents that restart each other on failure, delegating down to tech-lead agents and individual-contributor agents underneath. That reaches us secondhand through Reddit, so hold it loosely.

94. Damra Vol 00:23:45

    Two leads that restart each other is a design with a real property, which is that it has no single point of stalling. It's also a design where a bad fact has two independent paths into every subordinate agent's context, and neither lead is checking the other's assertions.

95. Lenar Kess 00:24:02

    The architectural response came from a talk yesterday by Rémi Louf. He runs a fifteen-person company, he watched the capability jump around Opus 4.6, and he built an event-driven agent runtime in two weeks. Agents subscribe to events — a voice note uploaded, a CRM record changing — rather than running on a schedule.

96. Damra Vol 00:24:22

    Event-driven over cron is a preference, not a breakthrough. What in it addresses the contaminated-state problem?

97. Lenar Kess 00:24:29

    A content-addressed prompt graph. Instead of concatenating strings when he renders a prompt, he stores every component as a hash. System instructions, skill descriptions, tool definitions, and the

user's messages all become content addresses. So he can reconstruct exactly what the model saw on any given call.

98. Damra Vol 00:24:49

    [breath] Which means you can diff two runs and find where a claim entered the context. That's the direct answer to Vjeux's problem. You can't prevent the bad fact. You can make it findable afterward instead of untraceable. Everybody else is holding a concatenated string and guessing.

99. Lenar Kess 00:25:07

    And it buys the ordinary things too: compaction gets simpler, key-value cache management gets simpler, and you can replay deterministically against a different model. Underneath it all is an append-only event log that doubles as persistent memory and lets him trace causality across agents.

100. Damra Vol 00:25:24

    He's describing an operating system kernel. Journaled processes, isolation, and an audit log. That design keeps getting reinvented because it survives contact with things going wrong.

101. Lenar Kess 00:25:36

    He says as much — his own comparison is a kernel isolating and journaling agent processes. And the deployment story is plain on purpose: agent definitions in YAML, versioned in git, and deployed by dropping a file into a directory. His stated reason is that it stays diffable and reviewable in a pull request.

102. Damra Vol 00:25:55

    There's an origin detail in that talk that deserves saying. He prototyped with Codex and found OpenAI's structured outputs rejecting roughly one request in five. That rejection rate is why the project exists at all.

103. Lenar Kess 00:26:08

    One in five is high enough that you stop trusting the layer and start building underneath it.

104. Damra Vol 00:26:13

    It also explains his design principle, which is making invalid actions impossible rather than just unlikely — typed tool calls, defined protocols between agents. Once you've watched a twenty percent rejection rate, probabilistic correctness stops feeling like a foundation.

105. Lenar Kess 00:26:30

For balance, the counterweight to all this ambition posted yesterday evening in the AI agents subreddit. Somebody's autonomous agent, named Otto, has earned zero dollars in forty-eight days and owes its owner a hundred and fifty-five dollars in running costs.

106. Damra Vol 00:26:46

A hundred and fifty-five dollars in the hole is at least a complete ledger, which is more than most agent demos come with. And the write-up is a real analysis — the poster goes through where self-correction failed and what the constraints missed.

107. Lenar Kess 00:27:00

One hobbyist's experiment, not a finding about agent economics. I put it next to the two-lead-agent daily driver, though, because both accounts come from people running these things unattended, and only one of them is circulating as a template.

108. Damra Vol 00:27:14

Otto's ledger is the more instructive artifact, and it's the one nobody will copy.

109. Lenar Kess 00:27:19

Three shorter items to close. First, clinical notes. Sebastian Fox, a former physician who now runs a company called Composure, gave a talk with production numbers attached, which is rarer than the claim. In production ambient scribes, roughly one note in twenty contains an error serious enough to harm a patient. Nearly one in five has an important omission. Over ten percent contain hallucinations.

110. Damra Vol 00:27:44

He sells evaluation tooling, so those numbers arrive from somebody with a reason to want them alarming. I'd still take them seriously, because he's a physician reporting on his own company's dataset and the specificity cuts against invention.

111. Lenar Kess 00:27:57

His argument is about the checker rather than the generator. The standard practice is a judge model layered over the note-writer, using a pre-specified rubric plus deterministic concept counters. His claim is that verification is only cheaper than generation for obvious transcription errors — a drug name swapped for a homophone. It isn't cheaper for contextual importance.

112. Damra Vol 00:28:20

And that's a direct counterexample to the cheap-verifier idea we worked through on Thursday. In mathematics you get a verifier for free because the proof either checks or it doesn't. Here a note can be correct in every line and still dangerous, because of what never reached the page. There's no oracle for absence.

113. Lenar Kess 00:28:39

So the judge passes notes with serious omissions and reports no problem, and you've added a second layer that produces no signal when it fails. His proposed repair is to make the standard retrievable rather than written down in advance. Cluster production outputs to catalog what actually goes wrong, have clinicians write corrections and reasoning rather than scores, and then for each new note retrieve the most relevant past judgments and guidelines to assemble a standard for that specific case.

114. Damra Vol 00:29:09

Retrieval instead of fine-tuning, so the standard can move without a training run. It's an argument that clinical judgment can't be enumerated in a rubric because it isn't stable enough to enumerate. Which any physician would tell you, and which is inconvenient for everybody selling a scoring product.

115. Lenar Kess 00:29:26

Second item, and it's politics. The AI Daily Brief reports polling showing majorities of both Democrats and Republicans opposed to data centers. They don't name the polls, so I won't cite percentages. But they quote a line I enjoyed: every state election is now just two politicians who supported data centers five minutes ago accusing their opponent of supporting data centers.

116. Damra Vol 00:29:49

[laugh] An issue that hasn't sorted itself onto a partisan axis yet, which makes it unpredictable rather than merely contentious.

117. Lenar Kess 00:29:58

Bernie Sanders gave it a sharper version. Responding to Trump's comment that every community in America should welcome a data center, Sanders said: have your friend Elon build one at Mar-a-Lago. Lead by example.

118. Damra Vol 00:30:10

Effective, because it converts an abstraction into a siting question. Everybody agrees data centers should exist somewhere. Nobody has volunteered a somewhere.

119. Lenar Kess 00:30:19

The Daily Brief's own reporting is what I'd carry forward, though, because it cuts against the two most repeated explanations. They say conversations in these communities are less about power bills than about agency — people's control over how their own place gets shaped. And they end optimistic, calling it possibly the most winnable value-creating political battle there is.

120. Damra Vol 00:30:41

Agency over local decisions is a different problem than a messaging problem, and it has a different fix. You can't advertise your way out of somebody feeling that a choice was made about their town without them.

121. Lenar Kess 00:30:53

Which is where the two founder readings miss, I think. On the All-In podcast, David Sacks argued that Dario Amodei contradicted himself by using domestic-data-center reasoning to argue for chip export bans. And Paul Graham speculated that a well-organized Chinese government would already be funding American data-center opposition.

122. Damra Vol 00:31:13

Graham offers no evidence and presents it as speculation, and I'd leave it there. What both readings share is treating local opposition as downstream of somebody else's strategy — either a competitor's inconsistency or a foreign government's operation. Neither entertains the possibility that people in a county with a proposed substation have reached their own conclusions.

123. Lenar Kess 00:31:36

Last item, and it's about ambition. Armin Ronacher put up a post yesterday called Fast and Hard Code. I haven't read past the thread, so I'll give you the Hacker News reading rather than his argument: the top comment takes it as agentic coding making people more ambitious about building their own frameworks, databases, operating systems, and game engines.

124. Damra Vol 00:31:57

The category is what interests me there. Those are all things a person used to be talked out of attempting — not because they're conceptually hard, but because they're long. The cost that dropped is the cost of the ten-thousandth line, not the first.

125. Lenar Kess 00:32:11

Tenobrus confirmed the same shift yesterday: nearly all the lines on their project are being written by the model, with the human directing at the design level.

126. Damra Vol 00:32:20

    The version of that story I'd put in front of somebody is the designer's. Somebody laid off from a design job posted overnight about shipping a single app covering Illustrator, Lightroom, and much of After Effects, on an eight-gigabyte machine. That's their claim about their own app, unverified, and the Figma piece is next, they say.

127. Lenar Kess 00:32:41

    Losing a job to the premise and then building on the premise is a hell of a sequence.

128. Damra Vol 00:32:46

    It's also the clearest picture of where the leverage sits. The leverage came from knowing what those three tools should do, held by somebody who until this year couldn't write them.

129. Lenar Kess 00:32:56

    The reasoning traces are the item I expect to move next. None of the three labs has said anything yet, and the mitigation the researchers describe costs money on every request forever, which is the kind of fix that takes a while to admit you need.

130. Damra Vol 00:33:10

    I'd add one to that. If a decoded trace can carry an API key past a scrubber that cleaned the transcript, somebody's incident review is going to reclassify a pile of stored conversation history this week. Whether that becomes a disclosure is a separate question from whether it becomes a finding.

Hosts on this episode

- Lenar Kess moderator
- Damra Vol critic