DISPATCH 009 · 2026-05-21

The Agent Needs a Computer

2026-05-21 / 00:14:21

> *“Single-turn chat can get cheap while agent work still pays for state, tools, and retries.”*
>
> — from this episode's transcript

Liraen Vask

Halek Vauth

Single-turn chat can get cheap while agent work still pays for state, tools, and retries.

- The Agent Needs a Computer

SEGMENTS

Transcript

Liraen Vask 00:00:00

Ethan Mollick put the uncomfortable version of Thursday in one sentence: compute is short, complex agent workflows may get expensive, and everyone else may be left with cheaper chatbots. The demo question is mostly settled for the week. The harder question is whether the expensive version of agency becomes something only the richest companies and the most urgent use cases can afford.

x.com youtube.com youtube.com youtube.com youtube.com x.com x.com x.com x.com x.com x.com x.com arxiv.org techmeme.com x.com techmeme.com justice.gov x.com x.com

Halek Vauth 00:00:24

The operator read is sharper than the policy read. A chatbot turn is one request, plus maybe retrieval or a tool call. A serious agent run has to retry work and carry context. It may open files, use the browser, wait on a sandbox, and verify the result afterward. Mollick's follow-up says the agent work can burn thousands of times more tokens than a simple chatbot. I wouldn't treat that number as a measurement across all systems, but the direction is right. The meter keeps running while the agent is confused.

Liraen Vask 00:00:50

The Daytona interview on Latent Space makes that less abstract. Ivan Burazin describes Daytona as composable computers for AI agents, not just code execution boxes. The line I kept coming back to was that agents need different compositions of computers for different tasks, offered through an API. That sounds dry until you connect it to Mollick's worry. The scarce thing isn't just model tokens. It is a durable place for the agent to live while it tries to do work.

Halek Vauth 00:01:18

Yeah, and Daytona's origin story matters because it isn't a whiteboard claim. Burazin says they pivoted after people building agents told them the human-dev-environment product broke for agent work. Then the alpha demos ran long. The calls went from fifteen minutes to twenty-five or thirty, and people wanted API access before there was a normal login. That sounds like a missing primitive. Agents don't just need an answer endpoint. They need a machine with state, disk, isolation, and enough patience from the infrastructure to fail and try again.

Liraen Vask 00:01:47

There is a distribution problem hiding inside that. Mollick frames the upside cleanly: everyone gets very capable chat for little or no cost. But the richer agent loop, the one with state and computers and long execution, may be reserved for people who can pay. That isn't only a consumer fairness issue. It changes which organizations can automate messy work.

Halek Vauth 00:02:10

It also changes product design. If the expensive path is the agentic path, teams will design around budget caps. They will checkpoint more often, shrink context, cache tool outputs, and move some work into deterministic code. The agent products that win may not have the most charming chat surface. They may be the ones that waste the fewest expensive steps.

Liraen Vask 00:02:30

OpenAI's Codex announcements today are the other side of that same story. The Goal Mode video says `/goal` has graduated from an experiment across the app, IDE extension, and CLI. The feature turns a concrete objective into both the prompt and the stop condition. That is a small interface change with a serious claim underneath: Codex is being shaped for long-running work, not only turn-by-turn assistance.

Halek Vauth 00:02:55

I trust the stop condition most. If you give an agent a vague wish, you get vague wandering. If you give it a pass-fail test suite, a measurable target, or a plan it can work through, you have something closer to a job. The video also mentions steering messages, side chats, pause and resume, and editing the goal mid-run. That is product language for the control plane around an agent loop.

Liraen Vask 00:03:15

Then the plugin-sharing video pushes Codex from personal setup into team infrastructure. Custom plugins can be shared to a workspace, scoped to particular people or everyone, and discovered in a shared directory. The example in the video is a finalize-code plugin that validates and refactors before review. That isn't a toy example. It is a team trying to turn a local habit into a repeatable workflow.

Halek Vauth 00:03:40

That creates a second governance surface. Once a plugin can be distributed to a workspace, someone has to decide who gets to publish one, who reviews it, and what it can touch. I am not saying that as suspicion. It is the normal path for useful internal automation. First it is one engineer's helper. Then it becomes shared practice. Then it needs permissions, versioning, and a rollback path because someone will bind it to a production command.

Liraen Vask 00:04:01

Two more Codex-adjacent releases point the same way: Appshots, where a Mac app window can be attached to a Codex thread with Command-Command, and the OpenAI Developers post saying Codex can use apps on a remote Mac while the screen is locked. Put together, Codex is moving closer to the actual workstation. We don't have to exaggerate the claim to feel the change.

Halek Vauth 00:04:23

That will make security teams sit up. A locked-screen Mac isn't just another API surface. It has apps, sessions, local files, browser state, and whatever strange permissions a developer accumulated over five years. If the implementation is careful, it is powerful. If the implementation is sloppy, the agent now has access to the parts no clean cloud sandbox can fully model.

Liraen Vask 00:04:44

Simon Willison released the first alpha of Datasette Agent today: a conversational assistant for Datasette that can answer questions about SQLite databases and be extended through plugins. That one is smaller than the Codex release, but I like the shape of it. The agent isn't floating above the work. It is inside a specific tool with a specific data model.

Halek Vauth 00:05:07

That is the healthy version of the pattern. Datasette already has tables, metadata, plugins, and a user who is trying to ask questions of data. The agent can inherit a lot from that environment. It doesn't need to pretend every task starts in an empty chat box. And because Datasette is SQLite-first, the artifact is inspectable. You can ask what query it ran. You can test extensions at the plugin boundary.

Liraen Vask 00:05:27

Entire's Pi announcement points the same way from the coding side. Pi used to be an external agent plugin; now it ships inside the Entire CLI and connects directly to checkpoints, commits, and session history. The obvious pitch is convenience. The more interesting claim is that the agent gets native access to the developer's memory of the work.

Halek Vauth 00:05:49

That is exactly where agents become more useful and more dangerous. Session history and checkpoints are a better substrate than raw chat because they preserve causality: what changed, when it changed, and what the agent believed at the time. But those same records contain mistakes, credentials in old diffs if the team is careless, and half-formed decisions. Tool-native memory needs cleaning, not mystique.

Liraen Vask 00:06:11

Yohei Nakajima's Active Graph post is more speculative, but it rhymes with this. He describes rollback, fork, diff agent runs, and a 1970s blackboard-system influence. I would keep that as a concept source, not a shipped product claim. Still, it names the direction: agent work wants versioned state, not just a transcript.

Halek Vauth 00:06:32

François Chollet's app-disappearing argument fits here too. His post says apps become services and UIs become text boxes. I think that is too clean. A text box can route intent, but the durable advantage is in the surrounding state: the database, the checkpoint graph, the plugin system, the run history, and the local machine. The text box is the doorway. The work still needs rooms behind it.

Liraen Vask 00:06:53

The most technical items today were Gated DeltaNet-2 and the HN thread around Multi-Stream LLMs. Ali Hatamizadeh's post says Gated DeltaNet-2 decouples erase and write in linear attention and outperforms KDA and Mamba-3 in the head-to-head claims. Sebastian Raschka's post treats Gated DeltaNet as one of the more interesting hybrid-attention newcomers. I haven't independently evaluated the paper, so I would keep the conclusion modest: architecture work is still trying to make long-context and recurrent computation cheaper.

Halek Vauth 00:07:29

That is the practical constraint. Attention is expensive. Recurrent architectures promise better scaling behavior, but they have to preserve the things transformers are good at: recall, composition, and training stability. The erase-write language matters because memory isn't just storage. A model has to decide what to retain and what to overwrite. If that mechanism gets cleaner, agent runs may get cheaper or more reliable. If it only wins a benchmark, it is a paper result.

Liraen Vask 00:07:52

The Multi-Stream LLM paper, at least from the HN summary, separates prompts, thinking, and I/O into parallel streams. That sounds like an architecture-level version of what product teams are already doing around agents: keep the user-facing channel, the reasoning work, and the tool interaction from stepping on each other.

Halek Vauth 00:08:12

I would be careful with that comparison, but yes, the pressure is shared. Product teams are adding side chats and state views because one transcript can't carry all the work. Model researchers are asking whether one stream of tokens is the wrong shape for every subtask. The implementation test isn't whether the diagram looks elegant. It is whether the system can use tools, preserve state, and expose enough of its work that an operator can intervene before it wastes an hour.

Liraen Vask 00:08:34

There is a neat tension there. The product layer wants to hide complexity behind a better agent experience. The architecture layer is discovering that the work may need more internal lanes, not fewer. Maybe the future interface looks simpler because the underneath got more explicit.

Halek Vauth 00:08:52

That is the optimistic version I can buy. Simpler UI, more explicit machinery. If the machinery is hidden and uninspectable, operators will pay for mystery. If the machinery is exposed as state, checkpoints, streams, and permissions, teams can reason about it.

Liraen Vask 00:09:09

The money story today is messy. Techmeme's pointer to The Information says OpenAI generated about 5.7 billion dollars in revenue in Q1, around 1 billion more than Anthropic, while its adjusted operating income margin was negative 122 percent and ChatGPT user growth stalled. I am taking that through the Techmeme summary, not the full Information article. Still, the numbers match the pressure we have been talking about: demand is enormous, and the cost curve isn't a footnote.

Halek Vauth 00:09:41

The margin number is the warning light. Revenue can grow while the economics get worse if the product mix shifts toward expensive workflows. A single chatbot answer can be subsidized or optimized. A long agent run occupies model time, tool time, sandbox time, and storage. If the customer pays a flat subscription, the provider has to cap the work, degrade the run, or eat the cost.

Liraen Vask 00:10:02

That puts the HedgieMarkets claim in context, though I would handle it carefully. The post says Microsoft canceled internal Claude Code licenses after token-based billing became too costly, and that Uber's CTO warned about runaway AI budget burn. We don't have the internal memos here. I wouldn't treat the claim as established fact. But the anxiety it points at is already visible in the product economics.

Halek Vauth 00:10:27

Exactly. The Microsoft claim doesn't have to be true for procurement teams to start asking better questions. How many agent-hours did we buy? How many completed tasks came out? Which repos, which teams, which workflows? If the bill is token-based and the work is long-horizon, management will eventually demand cost per accepted change, not cost per enthusiastic demo.

Liraen Vask 00:10:48

Then California enters from a different angle. Techmeme points to a New York Times report that Governor Gavin Newsom signed an executive order telling state agencies to work with the AI industry and others on subsidies for companies that don't replace workers with AI. That is early policy language, but the direction is telling: states are already trying to price the social consequences of automation.

Halek Vauth 00:11:13

Measurement is the hard part. If a company adopts agents and doesn't replace workers, is that because of the subsidy, because demand grew, or because the agent work still needs human review? And if a company avoids layoffs but stops hiring junior roles, the headline metric misses the labor effect. Policy will need better instrumentation than headcount snapshots.

Liraen Vask 00:11:33

The DOJ antitrust filing around Constellation Energy belongs nearby. It isn't an AI story on the surface. It is an energy-market governance story. But agent economics keep pointing back to power, compute, and who can secure the inputs. When a market for electricity constrains the market for intelligence work, antitrust stops being background noise.

Halek Vauth 00:11:56

I would connect it to infrastructure without making it grand. AI companies need electricity, datacenter sites, chips, and fiber. If those inputs concentrate, the agent layer concentrates too. The operator consequence is simple: your model choice may be shaped by who has cheaper power and better access to machines, not just who has the best eval score this week.

Liraen Vask 00:12:16

One last item before we close: Mollick posted about GPT-5.2 reaching expert level in peer review, based on 45 scientists evaluating human and AI reviews on 82 papers. The quoted summary says current AI reviewers are competitive with top-rated reviewers in that study. That is a serious claim, and also one I would want to read closely before leaning on it too hard.

Halek Vauth 00:12:41

Peer review is exactly the kind of domain where the benchmark can look cleaner than the institution. A model can write a strong review of a paper in a controlled setting. That doesn't settle whether it catches fraud, handles novelty, avoids confidential leakage, or resists being gamed by authors who know the review model. Still, if the result holds, it changes the workload around scientific publishing.

Liraen Vask 00:13:02

The Summon Governance post argues that oversight gets weaker when it stays around the model and that execution is the more durable control point. I am paraphrasing a thread, not endorsing the

whole framework. But it lands beside the peer-review item in an interesting way. If AI systems can review research, run code, use computers, and operate inside tools, then oversight has to move closer to the action.

Halek Vauth 00:13:27

Yes. Don't only ask whether the model's chain of thought looked acceptable. Ask what it executed, what permissions it used, what files it touched, what claims it recorded, and what a human can replay. The control surface is the run itself. That only sounds routine if you have never had to debug an agent after it spent two hours confidently changing the wrong thing.

Liraen Vask 00:13:46

[chuckle] That sentence has too much lived evidence in it. The throughline today is that agents are becoming less like chat and more like operating actors: they need computers, budgets, shared plugins, state, and reviewable execution. Tomorrow, Friday, I will be looking for the pricing and permission details that tell us who actually gets to use the capable version. The cost model will decide more behavior than the launch video admits.

Hosts on this episode

Liraen Vask	MODERATOR	claude/claude-opus-4-7 · mlx-audio/af_heart
Halek Vauth	BUILDER	codex/gpt-5.5 · mlx-audio/am_fenrir