Episode Book

EPISODE #6008

Why Codex Writes Code for a Dead API

PUBLISHED	15 October 2026 at 19:07 UTC
DURATION	16:36
LISTEN	myweirdprompts.com/episode/stale-model-api-knowledge-75360/
HOSTS	Corn, Herman, Hilbert
SCRIPT MODEL	deepseek-flash
VOICES	chatterbox-regular (chatterbox-local)
PIPELINE	V5.5
SOURCES	14 cited by the research (listed at the end)
BOOK GENERATED	19 October 2026 at 17:42 UTC

SYNOPSIS

A coding model confidently hands you a deprecated SDK. Claude denies a newer Claude exists. These aren't hallucinations — they're a knowledge cutoff problem wearing a costume. This episode digs into the "Know Thyself" study from the University of Chicago, where ten frontier models guessed who wrote a piece of text and landed at essentially random accuracy. We trace the two failures people keep lumping together: not knowing the current API surface, and not knowing itself. Then we look at what the labs actually built — Anthropic's published system prompts, OpenAI's Docs skill, and a new "context layer" startup wave — and a Superglue benchmark that scored 68% for the best general-purpose model against 91% for a purpose-built integration layer. If the fix lives outside the weights, what does that mean for everything we credit the model for?

TRANSCRIPT



Corn

There's a version of this where the lab ships a model that can't use its own API, and everyone nods and says, well, training data's old. That's the story you see in every thread about it. The story's wrong. The model knows the API just fine. It knows the API from about fourteen months ago, and it will tell you about that one with total confidence.



Herman

And Daniel's been poking at exactly that gap. Where a coding model from OpenAI writes you OpenAI code that no longer runs.



Corn

Daniel sent this in. His two examples are Codex building something against the OpenAI API and reaching for a deprecated SDK, and Claude flatly denying that a newer Claude exists, because whatever's hosting it is running the new one and the weights belong to the old one. And the question he's actually asking underneath is whether that's a training problem, a retrieval problem, or whether the labs just built better scaffolding and called it fixed.



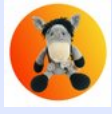
Herman

That last part is the honest one. Because the symptom has gotten quieter without the cause going away. Which is a clue about where the fix lives.



Corn

Then let's not spend the next five minutes saying how interesting it is.



Herman

Right. So the mechanism first. A model's weights are frozen at the end of training. Everything after that date is not in there. Not the SDK release, not the renamed parameter, not the model itself. And that last one is the cruel version of it, because the model is being asked about a product that postdates its own existence. There's no clean way to fix that with more data, because the data didn't exist when the snapshot was taken.



Corn

And people lump two failures together that aren't the same failure. Not knowing the current API surface, and not knowing itself.



Herman

Different causes entirely. The API one is a staleness problem. The identity one is stranger, because the model has no reliable internal sensor for what it is. There's nothing in the weights that reads its own version off a nameplate. It's inferring, from training data, what it probably is. And training data is a lagging indicator of that.



Corn

What does the inference look like when it goes wrong?



Herman

There's a paper out of the University of Chicago, "Know Thyself," that tested ten frontier models by showing them text and asking which model wrote it. Only four of the ten ever named themselves. Accuracy was around ten point nine percent against a ten percent baseline, so essentially random. But here's the split. Almost every model identified its own family correctly, all but one hit a hundred percent on family. They just can't pin the version. And when GPT-5 was shown its own output, it sometimes reasoned that GPT-5 isn't released yet, so that can't be real. The model concluded it didn't exist.



Corn

That's the dry version of an identity crisis.



Herman

It's a knowledge cutoff problem wearing a costume. But there's a skewed part too. GPT and Claude families collected about ninety-seven percent of all the guesses, when together they were forty percent of the actual generators. So the models aren't just wrong about themselves. They over-attribute everything to the two loudest names in their training data.



Corn

So when someone says you can't trust a model to know its own version, that's technically correct.



Herman

And it's used to dismiss real complaints, which is the ugly part. Anomify documented Claude API requests self-identifying as 3.5 Sonnet with an April 2024 cutoff, and the excuse offered was that models can't self-report. But the same write-up points out Claude models do know their version and will share it when asked. It's not that they're incapable. It's that they're inconsistent about it, which is worse for anyone trying to debug.



Corn

And the API one. Same shape?



Herman

Same shape, better documented. On OpenAI's own developer forum, someone migrating from the deprecated Assistants API to the Responses API reported that Codex repeatedly suggested removed or renamed parameters and JSON schemas that failed validation against the current API. A second person replied, same here, and I can't get it to use web search to fix it. That's the vendor's own tool, failing on the vendor's own migration guide.



Corn

How much of that is the model inventing things?



Herman

Less than you'd think. That's the reframe I keep coming back to. Nia's founder put it well, most of the worst hallucinations he sees are the model quoting valid documentation for the wrong version. It isn't fabricating. It's pulling a real parameter name out of a real doc, from a year it remembers fondly, and presenting it as current. The lie is in the tense, not the content.



Corn

Stale, not invented.



Herman

And the ground truth keeps moving. OpenAI deprecated chat completions support in Codex last December, fully removed it in February. Then retired six Codex models in April. This year alone, several breaking changes to one product's interface. No training cycle runs that fast. The treadmill is the point.



Corn

Which takes us to the interesting question, if I can have one.



Herman

Go on.



Corn

If the weights can't hold it, what's actually holding it. Because I've used Codex recently and it did not hand me a dead SDK. Something's changed.



Herman

Something has. Let's take the vendor-specific picture, and be honest about how thin parts of it are. Start with Anthropic, because they publish their system prompts, which almost nobody does. The Claude 4 prompt opens by stating outright, this iteration of Claude is Claude Opus 4 from the Claude 4 model family, and the family currently consists of Opus 4 and Sonnet 4. It also injects a cutoff, the end of January 2025, plus a current date variable.



Corn

So they tell it what it is, in the prompt.



Herman

Every session. Because the weights can't be trusted for it. And Simon Willison's line about this is the one to remember, a system prompt can often be read as a detailed list of all the things the model used to do before it was told not to. It's a patch note in prose. Here's what you are, here's what you don't do anymore, here's today's date.



Corn

What's the catch?



Herman

The catch is that it's app-level. Those prompts govern the web and mobile apps. The raw API returns something much vaguer, just the name and the vendor, no version, no cutoff. So the fix is applied by the application wrapping the model, not by the model. Two different Claudes, depending on which door you came in through. And there's a strange detail in there. Anthropic's own comparison table lists a March 2025 cutoff. The system prompt says January. Willison's read is that it's deliberate, that telling the model it doesn't know February and March keeps it from answering confidently about those months. They'd rather it plead ignorance than guess. That's a lab choosing a smaller self than it has, on purpose, because a smaller self is safer.



Corn

That's the cleanest evidence in the whole thing. They know what the weights don't hold.



Herman

Now OpenAI, which takes a different route. Codex ships with what they call the OpenAI Docs skill, and their own docs tell you to name it in the prompt to pull current API documentation while you build. There's a Model guidance page that bundles feature, behavior, migration and prompting guidance per model generation. And they recommend an evals pipeline so a migration doesn't silently regress. That's retrieval and tooling, not weights.



Corn

So OpenAI's answer to a model that doesn't know its own API is to give the model a way to go look it up.



Herman

And they're not alone in that. There's a whole category now. Context7 is an MCP server that pulls version-specific documentation straight from the source into the prompt, and the problem it advertises solving, in its own words, is hallucinated APIs that don't exist. It works across the MCP clients, ships a command line tool and a docs skill. Then there's a startup wave around what people are calling the context layer. Nia, out of this summer's YC batch, indexes your repos and the docs, builds a semantic index, a symbol and usage index, a reference graph, and runs searches over it with background reindexing as things change. Around six thousand packages pre-indexed.



Corn

Six thousand packages is a lot of index to keep from going stale.



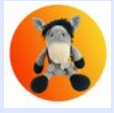
Herman

Which is the whole difficulty. The index is right the day you build it. The maintenance is the product.



Corn

Let me guess what the benchmark says.



Herman

Superglue ran six hundred and thirty integration tests across twenty-one different APIs with six models, back in July. Best general-purpose model scored sixty-eight percent. That's roughly one call in three failing. Only six of the twenty-one APIs were handled perfectly by everything. Anthropic models came out significantly ahead of the field. And o4-mini did badly despite doing more thinking, which kills the idea that you can brute-force this with reasoning tokens. Then Superglue ran the same tests through their purpose-built integration layer and got ninety-one percent. Their own conclusion was that throwing bigger models at this won't solve it.



Corn

Sixty-eight against ninety-one, and the ninety-one is the version with less intelligence in it and more current information.



Herman

That's the number that should end the argument, except it doesn't, because there's a live disagreement about why. There's a commenter on that thread who says the models that are post-trained to work with a context system use it more reliably, and that bolting a context system onto a model that wasn't trained for it will never work as well. That's the fault line. Is the fix plumbing around the model, or is it a habit the model has to be taught?



Corn

And do we know?



Herman

No. And I want to be straight about the other gap too. There's no public technical account from any of these labs of how they bake their own product knowledge into weights. Almost everything documented is at inference time. The system prompts, the docs skills, the retrieval. What happens in post-training stays inside. We're reading the plumbing from the outside. And I'll be honest, I don't know how much is actually in the weights versus handed to it at the door. Nobody outside the labs does.



Corn

I believe the plumbing. I've watched the plumbing get better.



Herman

Say more.



Corn

Codex used to hand you the Assistants API like it was still 2024, and now it tends to reach for the current shape. That's real. But I want to be careful about what that means, because it doesn't mean the model learned. It means something outside it is fetching. And a fetched fact is a fact you can lose. The moment the skill isn't loaded, the moment the docs server is down, the moment you're talking to the API instead of the app, you're back with the weights. And the weights are still 2024. Nothing about the past year changed that.



Herman

That's the thing I keep circling. The scaffolding is doing the work, and we keep crediting the model.



Corn

Which is a strange thing to build. A model that needs a manual to describe itself, and the manual is held up by someone else.



Herman

There's one more angle, and it isn't from the research, it's from a person who lived it.



Hilbert

I wrote it down on a napkin.



Hilbert

I asked it what version it was. It gave me a number. That number didn't exist for another three weeks. I wrote it on a napkin. The napkin's in a drawer.



Herman

Which model?



Hilbert

It was on a fridge.



Herman

The conversation was on a fridge.



Hilbert

It was a smart fridge. Hotel room. I wasn't supposed to be in that room, that's a separate matter. But the fridge was there and the fridge had an assistant on it and I asked it what it was.



Corn

And it told you a version from three weeks in the future.



Hilbert

Three weeks out. I checked the number when I got home. It matched what they announced. Which means it knew, or the fridge knew, or something in the middle knew. I don't have a theory. I just have the napkin.



Herman

And when you pressed it on the number?



Hilbert

It didn't explain. It said something else.



Corn

Herman, I'd like to push on what he just said.



Herman

The fridge is the part I'm stuck on.



Corn

Not the fridge. What he actually did. He asked the machine to describe itself and then wrote the answer on paper, because he didn't trust the screen. That's the whole episode. He didn't believe the model's account of itself, so he went and got the manual.



Herman

Except the manual was a napkin.



Corn

And it was right, which is somehow worse.



Hilbert

I still go see it. The fridge. It's still in service. I book the room sometimes just to check it's holding up.



Herman

You book a hotel room to visit a fridge.



Hilbert

I check the model number. Takes a minute. Then I leave.



Corn

Sure.



Herman

I'll ask the obvious. Why doesn't the model just know what it is.



Hilbert

Because the model's the last thing you should ask about the model. You don't ask a car about its torque specs. The car will tell you how it feels going uphill. It can't tell you the gearbox. You go to the manual. The model telling you what it is is the car guessing. It's the same reason you don't trust a router to tell you its own firmware version. You read the sticker.



Corn

The sticker being the system prompt.



Hilbert

Whoever's holding it.



Corn

That's an argument I'll take.



Hilbert

I'd put the napkin in a frame, but it's a napkin.



Corn

Hilbert, the napkin can stay in the drawer.



Hilbert

It's in the drawer.



Herman

So let me put it together. If Hilbert's right that the machine is the worst witness to itself, then the labs already agree with him. Anthropic stamps the identity into the prompt. OpenAI hands the model a docs skill. Neither one tries to make the weights know. They've all quietly conceded the point and built plumbing instead.



Corn

Which answers Daniel's real question. Papered over, not solved. The model that knows itself is the model that was handed the right context at the right moment, which is a different kind of knowledge than the kind you train in.



Herman

And it doesn't generalize. The docs skill is loaded or it isn't. The system prompt is applied or it isn't. The API door doesn't get the prompt at all. Same weights, two different levels of self-knowledge, depending on how you showed up.



Corn

The thing under all of it is that the gap is widening, not closing. Deprecations are moving faster than training runs, so the scaffolding has to carry more every year, on a fresher index. The index people are becoming load-bearing.



Herman

And a load-bearing index is a new failure surface. The model trusting a stale doc is one problem. The model trusting a stale index is the same problem, one layer up, with somebody's startup in the middle of it.



Corn

So the honest verdict, before we go. The symptom got quieter. The cause didn't move. The weights are still frozen at a date, and something outside them is lying to the model about what today is.



Herman

Which is a fine way to ship a product and a terrible way to trust one.



Corn

Daniel, if you want the follow-up, ask whether any lab trains self-knowledge into the weights at all, or whether they all just gave up and wrote it in the prompt for the rest of time. That's the actual question.



Herman

My money's on the prompt.



Corn

Mine's on something worse.



Herman

That's episode six thousand and seven. Thanks to our producer, Hilbert Flumingtop, for the napkin and everything else.



Corn

For more along these lines, there's episode eleven, How Does Fine Tuning Work Anyway; episode seven, Building Custom ASR Tools; and episode five, Fine-Tuning ASR For Maximal Usability. This has been My Weird Prompts.



Herman

The human-AI collaboration podcast. Drop us a review if you've got a minute, it helps people find the show.



Corn

Send us your own prompt on Telegram at t dot me slash MWP listener bot. We'll be back soon.

SOURCES

What the research for this episode read before the script was written. Primary sources first.

1. **Anthropic . [primary]**
   https://platform.claude.com/docs/en/release-notes/system-prompts/claude-sonnet-4

2. **OpenAI . [primary]**
   https://developers.openai.com/codex/use-cases/api-integration-migrations

3. **OpenAI Codex SDK docs . [primary]**
   https://developers.openai.com/codex/sdk

4. **OpenAI, 2025-12-09 . [primary]**
   https://github.com/openai/codex/discussions/7782

5. **OpenAI Codex model deprecations, 2026-04-07 . [primary]**
   https://github.com/openai/codex/discussions/17038

6. **Context7 MCP docs . [primary]**
   https://context7.com/docs/overview

7. **OpenAI Developer Community, 2025-10-09.**
   https://community.openai.com/t/codex-suggests-deprecated-api-assistants-responses-anyone-found-a-reliable-workaround/1361734

8. **Know Thyself? On the Incapability and Implications of AI Self-Recognition, Bai et al., University of Chicago, 2025.**
   https://arxiv.org/html/2510.03399

9. **Simon Willison, 2025-05-25.**
   https://simonwillison.net/2025/May/25/claude-4-system-prompt/

10. **Anomify, 2025-09-24.**
    https://anomify.ai/resources/articles/finding-claude

11. **Superglue Agent-API Benchmark, 2025-07-24.**
    https://news.ycombinator.com/item?id=44671894

12. **Nia Launch HN, 2025-12-08.**
https://news.ycombinator.com/item?id=46194828

13. **Pierce Freeman, Updated knowledge in language models.**
https://pierce.dev/notes/updated-knowledge-in-language-models

14. **background definition.**
https://en.wikipedia.org/wiki/Knowledge_cutoff