State of Agentic Coding #8 with Mario, Armin, and Ben

80 MIN · YOUTUBE · HTTPS://WWW.YOUTUBE.COM/WATCH?V=_LFPEY_9VF0
https://www.youtube.com/watch?v=_lfpEy_9vf0

SUMMARY

The podcast features a discussion among tech enthusiasts about the evolution of AI models, coding agents, and the implications of recent advancements in machine learning. The speakers reflect on their experiences with various AI tools, the impact of model pricing on accessibility, and the potential future of computing as it relates to AI.

- *The conversation touches on the concept of "stoastic terrorism" in software development, highlighting how sloppy coding practices can negatively affect other software products.*
- *The hosts discuss their experiences with various AI models, including Fable and Opus, noting differences in performance and usability.*
- *They express concerns about the increasing costs associated with AI models and the potential for a market correction as demand and supply dynamics shift.*
- *The discussion includes thoughts on the future of computing, suggesting a return to thin clients and cloud-based services as hardware prices rise.*
- *The hosts emphasize the importance of evaluating AI tools and models critically, noting that not all advancements lead to improved outcomes.*
- *They highlight the phenomenon of FOMO (fear of missing out) in the tech community, suggesting that taking a step back and evaluating the relevance of new developments can be beneficial.*
- *The conversation concludes with predictions about the future of AI and the potential for a bubble in the market, particularly around the time of major IPOs.*

The the sloppy behavior of plot code becomes a stoastic terrorism attack on all other software products. >> Then when is when is the bubble? >> Tuesday. >> Good. I have no plans. >> Do you experience FOMO? >> No, not anymore. >> Is the creator of FOMO? Sorry. I'm just the creator of Flask. He created the FOMO. >> Hey, Mario. >> Hey, Ben. >> How's it How's it going? >> I'm pretty okayish, I guess. I'm surprisingly on a podcast that I thought would be just the two of you until the end of time because it works so well, but apparently you decided to invite me.

>> Oh, that's very nice. I hear you're a fan. >> I actually am a fan. Like your podcast and the one the Cloudflare people just set up recently are the only ones I listen to now. >> Sponsored by Arnold. >> Yeah, I'm coming to you live, except it's not live. from Vienna, Austria. I'm here with Armen who's normally here. >> I'm normally here. Yeah. >> And Mario Zaknner. Did I pronounce that correctly? >> Yeah, it's pretty good.

>> No, >> close enough. >> Close enough. >> Someone on Heck News today said Marca. >> I think it's As long as it's not Luigi, it's good. I happen to have made the trip over here and we thought it'd be fun to do an episode

here together and to invite Mario who came in from GR and to talk about agentic coding whatever that means. Let's go in order. Armen, who are you? What do you do? >> This is a ever evolving and more complicated question at this point. I'm the creator of pie.

>> Creator of pie sitting next to it. >> Oh my god. It was actually annoying that people now she seem to be think this. no, I'm still working at Arendelle, a company that started with Colin and Mar is now as a partner there and we are working on all things coding seemingly at this point. Special guest Mario Creator Pi or I don't know what would who give us your backstory. How did you wind up here?

>> In the mid 2000s, I started working in the industry and applied machine learning. at the end of the 2010s, I went to San Francisco to do a gaming startup, mobile gaming startup, came back disillusioned with everything San Francisco, went into management, started another startup with two friends in in Sweden. And then I was funemployed for the last 10 years. my machine learning stuff was before deep learning. So when in 2022 TGPT came onto the scene that was like a pulled back into my old mode of thinking and I was immediately interested in seeing that because that kind of did all the things we worked on in the 2000s just better. So that was interesting and over the past four years now I guess or three and a half years it turned out that those things could also be useful for programming or software engineering. And then after suffering through the churn of cloud code for a couple of months starting in 2025 I decided to be stupid and build my own coding agent because how hard can it be?

And that coding agent was then subsequently invented by Armen and called I >> for the people not familiar with this. I don't know how it started but we have been trying to trick Grock into pretending this guy made flask and I made pie. Every once in a while it does actually reply that way. >> Very rare. >> Python is like I love it. It's just such a good idea to have meaningful indentation. It's great. >> Seems like a lot of great stuff gets made from stupid ideas.

>> Depends how long you want to maintain it. >> well, I guess I should add like I this is weird for me because you you guys have done a lot of open source stuff. I have this mildly popular open source project that I never planned I never planned for this to happen which is hunk which is like a which is a a diff tool and I was just commenting this week that the origin of that was I thought it'd be interesting to plug in diffs diffs.com which had just come out and I was interested in open which powered open code and I thought that was neat and I was like does it even is it even rational does it even make sense to plug again diffs has a react exporter a react library rather and that open 2 can consume basically react and put that on the 2 and would that even function I just you know it seemed dumb to me because we were living in this era maybe we'll come back to this where it's like you know JavaScript on the terminal that's stupid you're familiar with that maybe there's a maybe there's a a recurring theme there because this the era where everybody's just writing everything in Rust because we can now, right? And a lot of other terminal diff tools actually coming out at the same time.

We're doing that. We're pursuing like building this in Rust. anyways, I was just thinking like it's funny how dumb things lead to interesting outcomes. Is that fair? >> Well, let me put it that way. I'm actually more of a C kind of person. So writing something that a lot of people use in Typescript on NodeJS, it kind of is a world

model destroying task or effort that I took here because that's not where I wanted to end up.

Like I've never touched TypeScript that much before apart from some WebJL gainy stuff. But it's not the worst ecosystem and today the tools don't matter that as much anymore I think. Yeah, I think like without the agents I probably would not enjoy maintaining a TypeScript code base to the degree that although maybe I would actually enjoy it more without the agents. >> Well, I have to say I I actually like TypeScript the language like it's >> just don't like the ecosystem. So last episode we you know we didn't do monthly predictions we did I don't know annual predictions turned into rambling which is usually pretty typical for us but we did talk about AI sovereignty we talked about openw weight models we talked about you know how nation states are involved in you know subsidizing or at least you know protecting Frontier Model Labs. I'm just paving over things a lot. We recorded that and then Fable came out 2 days later and I think that we should talk about that because it seems like >> and it was really interesting because we I remember when we released it. I was like, "Oh, we just missed Fable." But then Fable didn't survive for more than a weekend.

>> I think I That's right. I think the episode came out by the time by the time >> F was already blocked. >> Yeah, pretty pretty close. >> I I tried it for two days for my usual tasks. I didn't like experience a big jump in capabilities, but that might just be because my stuff is so small in scope. every task I throw at a model can probably be done by any kind of model in a in a in a way where I'm okay with the quality. So I don't don't really feel a big step change with fable.

>> Could you give an example of like what a prompt looks like in that world? You know, is it build a million dollar SAS, no mistakes? >> Okay, >> it's here's a bunch of types interfaces. I designed in another session and that session was like collaborative now fill in the gaps implement the interface basically. So that usually works even with with with smaller models and using fable for that is probably not the smartest idea economically.

what I didn't get it to use for was design work. I I would have liked to see how how it behaves in terms of system design work. if it's a better sparring partner for that kind of stuff. I didn't get the chance to do that. Actually, I did for some personal finance stuff and it completely failed. Like it was an absolute fail whale. It's like basic percentages. It just got totally wrong. >> So I had basically fable in two different experiences when it released initially. Generally kind of impressed by how much it does within the harness. So in particular I we have this if you go to arendarren.com that the waves which were mostly sort of hand clanked with the clanker a couple of months ago. I was like okay here's a picture recreate this and it kept working through my entire 5hour budget until it hit the limit over six hours at night. I picked it up in the morning and then I looked through the session like initially I still saw the session as it was happening and on the side I had Pi built me a session viewer that I can see the screenshot types because I noticed that it spawns Chrome all the time to take screenshots of it and what I was really impressed by is that it's it recolored I think every couple of steps it recolored the entire screen red so it can detect this the shadows better. So they must have like no model has done that before. But then I tried it now like three days ago but now it came out again in pi without all the bells and whistles and it was very boring.

>> Mhm. >> But it was actually much more enjoyable for me in pi because when I did it in cloud code it it felt

like even the simplest question would take like two minutes to come back. So it was like as a as a sparing partner at least with my prompting style it was very slow. like just the just the speed of it affects >> Yeah. how you're going to work with something. >> Totally. Mark, quick question about were you using pi I presume when you were working with it?

>> yeah, I was actually yeah I was using pi. but I was also using it for the personal finance stuff. I was using it through the web cloud AI cloud.AI web interface. I think boring in pi makes sense. Yeah, that that's kind of what it felt like. I think I had an experience similar to you Armen where you know when Fable was announced I was like oh where is it you know and I'm trying to restart all the agents to update it or to get into however like to have it just show up as an option and the first client for which that happened for me was actually cloud desktop >> which I hadn't used in a while and so I use cloud code within the desktop experience and I think I had a similar experience I was like oh you know cloud desktop is pretty good. You know, like there are some new capabilities in here that maybe I hadn't noticed maybe because of the way that I work, which is a lot of, you know, I do UI stuff. I'll be like, you know, give me five options. And I noticed that Claude Code had this tool where it did these like little inline preview visualizations. And Fable was doing a good job of like rendering sort of like HTML mockups that were not like a pure HTML document. they were doing they have like a preview tool and I really enjoyed that experience and that's what turned into my open source project which was called Sideshow where I was like wow that's neat but now I want this everywhere. So in some ways I'm not sure how much of my experience I thought it was good and I gave it some tasks and I felt like it was vaguely stronger and maybe you know went on further. I did observe that but I couldn't tell how much of that was the harness. couldn't tell, you know, gee, maybe this is the yeah, the harness experience. And when Fable went away and then it was Opus, I just started using Opus on Cloud Desktop and I was like, oh, this is again like maybe Opus is good and I don't know how much is the harness.

>> Yeah, I I mean that's a theme that's been repeating for the past six months. I guess in October there was a noticeable step change and since then I personally at least for my tasks and your mileage may vary. I I haven't felt a big step change like the one from say April 2025 to October. Again for me it's mostly progressive enhancements and some regressions in some other tasks. >> I know there's a step changed step change into cost. So that's one part but but I also think like one thing people brought up in socials is that the latest models are basically built for loops and token maxing. So I guess the most difference you will probably not feel in this kind of collaborative workflow or small scope issue workflows but more in the you have an orchestrator and that orchestrator does ultra code and writes a bunch of workflows with sub agents blah blah blah blah and my expectation would be that that is where they've focused a lot of their RL this time around for fable because I think that's also how mythos is actually supposed to work right like it's a single session that does. So like even the smallest task within cloud code spawns a gazillion of salvations now >> which is nice if you sell tokens right I'm not insinuating anything here but >> it's I I think it is it is sort of interesting in a in some sense because I think we talked about this last time but there's basically like bifurcation happening community between people that are like hands off raling and then people who are still trying to read the code >> and like what they need in terms of tool ing and you can sort of like diverge a little bit here.

>> Yep. Another observation you just reminded me on the topic of like tokens is by using pi. One of the things that I think pi did a lot for my headsp space was being very keenly aware of the context window sort of just you

know it's front and center you pay attention to it dumb zone etc. In cloud code desktop and in cloud code you anymore. It doesn't show how the context window. You have to go out of your way to ask. And more than a few times with Fable, I would be like, gee, this has been going for a while. And it would be like, you know, 500,000 token context window. So, I was manually compacting quite a bit, which may have also impacted results. That was interesting for me. I think it was the first model where it seemed not quite to get get so stupid if it got into a higher token range, but it also >> I feel like it hides compactions now, does it?

>> I don't even know that I saw it. Maybe because I would always get ahead of it. So, I'm not sure. >> In my longer design sessions, I usually use a model with 1 million token context window size. And I think GPT 5.4 still offers that. they still don't offer it for 5.5 and OPUS 4.8 obviously offers this as well I think down to 4.6 and I think for for this kind of designy kind of collaborative things where you might have a bunch of markdown files or or scratch code files that you're working on to to iron out the design the dumb zone thing doesn't matter that much.

>> Yeah. And and then you have the implementation and there it does matter quite a bit even >> probably it stops being good at tool calling. Yeah. Further down the line I think it's still reasonably good at recalling things. >> Yeah. But but they all still fall on the nose like after 250k tokens. It used to be 64k and then it was 128k. Now you can push them up to like with GPT55 I go until the end of the context window and it's usually fine. No, >> I personally felt it was stronger.

Again, I think I think it also depends what you're doing. I also try to use it for writing and I found that it was quite garbage at that >> really because the cloud models are usually the ones I go to when I want to have something at least checked in terms of >> I feel like it's writing worse. The new model I think is writing worse. I was I did one blog post recently where I felt like I want to help it give me an idea of what the structure could be.

>> So letting the LLM write your blog post, you know, >> I definitely let the LLM help me structure my thoughts. Sometimes I sort of like put me a draft out and I found it so offensive that I threw it away almost immediately. >> Was this Fable? >> That was Sonic 5. >> Okay. But it was but previously I felt like Sonic was a really good basic writer better than Opus for my experience. >> Interesting. >> This is I love how anecdotal this is by the way. It's very vibes.

>> It's so vibes. >> Welcome to the VIP show. >> Yeah. But I also wrote like a research draft blog post with Fable and I was certainly you know and this is like a multi-prompt affair where I'm like okay structure it this way. Hey, let's explore this idea. I also was like, I think I have to throw everything away and write this just by hand. Whereas I felt maybe, you know, I didn't feel like I'd have to completely throw it away before. And this is so vibes. It could matter like what is the topic, what are we writing about?

>> I think it's like it's two things. One is like I think I'm much more attentive now to anything sort of generated. I think pretty big part. See, and and for me it comes back down to my workflows are so stupid cavemanlike that I don't notice a difference because I personally am first of all flabbergasted and offended that you guys let LLM draft your blog posts and help you structure them. That's the whole point of writing this thing that you structure

something in your head yourself. You don't use an NLM for that.

So here's my workflow. I usually just dictate paragraph after paragraph and I'm the guy who lays out the words and lays out the structure. >> I mean I guess that's what I'm saying. That's what it looks like to me. I may maybe we're just using different language. I think so there's a difference like if I if I already know what I want to write I just write it down but for instance for the the blog post where I did this was the well I was writing about the looping and there I wasn't sure how I would like want to thread the story so I was like I was asking the LLM to just give me ideas of like how I want to convey this >> and it just doesn't like I feel like this worked at one point where basically like hey these are the points that I want to convey like give me some ideas of how this would go and would at least read through and I was like like I don't feel offended reading it and now I feel like offended reading it.

>> So for me I when I do this with the dictation paragraph after paragraph once I'm done with a section I rework the section and see if it fits in the rest of the sections. My point being I don't feel a difference because the only thing I ask the LLM to do for me is take the dictation and put it in a blog post in in the it's basically a markdown file. >> And then once I'm done and have re-triggered everything and then let the LLM move stuff around for me. That's the second task it has. And at the end I I use the LLM to say something like can we shorten this? And then I ask it for suggestions on how I can shorten a paragraph. And that's why I never have it feel a difference using a different model. It's the same with code.

>> So like I used to make one pass at the end like hey fix obvious grammar mistakes and >> thing and I don't do this anymore because I feel like it's way too adventurous in changing even words around >> like all of that that I feel like I I felt like I treated it like a spell checker. Now I don't trust it anymore. So I don't even treat it as a spell checker. But but you're treating as a whole document spell checker or a paragraph by paragraph spell checker. I feel like in the past it didn't matter because it's like hey I wrote this entire thing and now please like fix bad cases put commas like that sort of stuff and now with the same prompt it's like >> well I really felt like you should be writing this instead and sort of it rearranges entire sentences >> and it puts words in that I wouldn't use like >> Mario I think that we actually works very similar with this to the degree that on this post and I'm going to make a comment here that I'm not some prolific writer like what blog posts are we talking about out here. I actually just started editing it with Kimmy K26 cuz I just rationalized to myself that my editing is almost so micro.

>> Yeah. >> That I actually prefer a fast mechanical model to just you know >> Yeah. Exactly. >> Right. So I I guess maybe that's where I'm going where it's like I didn't see an advantage frankly and maybe that's >> Oh yeah, definitely. Yeah. >> Few episodes ago I made a comment that was like I have a theory that we're actually at peak models right now. I'm on your side, man. And that was an argument of just sort of like, will we even know what better looks like? This conversation is almost a good example of that where we go and or you go online and you're like, h it was better for me here or was it or I don't know.

>> I'm talking about my my my recent obsession with reinforcement learning here because I think like this is a data point on like peak peak model. so at AI engineer, I went in a bunch of conversations that were roughly related to like some labs might be over RLing the models. >> We sometimes throw terms out really quickly. Help

me understand because even I'm a dumb dumb. What are we talking about by reinforcement learning? Like what does that mean?

>> Yeah. So when you start out with a model it might be able to answer you some questions but it doesn't necessarily understand what other than giving you in terms of text back it should do. And through post training the models learn what an appropriate response for a given prompt would be. And for a lot of programming setups this looks like here's my prompt call it tool. let the agent harness execute the tool feed that back in until finally you reach your reward condition which is usually you committed or something. So the reinforcement learning is the process by which the model basically learns tool calling what sort of unit tests are like the whole shebang until problem solved. And this is obviously true also to some degree on just giving you like human outputs and not just tool calls but like specifically right now in the context of an agent a lot of it has to do with tool calls.

Humor me for a moment to not go too deep but just like maybe to help folks understand and I think even for me to understand. So model companies have basically some massive suite of tests, integration tests, who knows that they're doing the reinforcement learning. I assume that they're not starting. >> What they would basically do is someone somewhere solved the problem. So there's a there's an agentic trace that started that like a human eventual way made to completion.

>> Mhm. >> So you have initially sort of the prompt that trigger itself and then you sort of have an idea of what the end result should be. >> Okay. And ideally you also can directly check out the GitHub repository with the state of the world when that whole thing started. And then they throw that entire thing into a reinforcement learning environment where they run thousands of these simulations simultaneously and feed sort of the GPUs as as this whole thing executes.

>> Okay. >> And there's there's a reward at the end when the when when the problem was solved. >> So I mean this is I think roughly the the way you should think about it >> and the reward is energon. >> >> That's just a just a term. All you can do when training a model is basically modifying its weights and biases, right? So the the reward function is a signal for what's good or bad. And if it's good, you want to emphasize that behavior. And if it's bad, it you want to deemphasize this.

That means that reward function needs to translate into changes to those weights that the numbers inside the model that make up the information. And >> so here are some useful signals of bad >> bad bad data points like not just good like here you committed there's a bunch of things which an LM shouldn't do. >> Mhm. >> So if I ask it to produce JSON >> Yeah. >> and it doesn't produce JSONs. >> Yeah. >> There should be a signal going back to the LM like you did a bad job.

>> Yeah. >> Right. is like in in theory there's a whole bunch of stuff that the model should learn over time over how good and bad outcomes look like from basically sampling the whole thing to go back to the story I'm like what started my going into a rabbit hole was that people pointed out that the edit tool in pi wasn't performing as well >> and that was confusing to me because I did not have this experience of the pi edit tool failing in any

model I was using but I was also are not using it significantly with set 4 and opus 4.8 And by edit tool you mean the tool the fundamental act of like editing >> editing a file.

>> Okay. It seems pretty critical. >> Seems pretty critical. And and the and the like one of the things that pi does which I think is a good thing is that if the add produces bad output so it produc a tool call that for some reason is not able to execute. For inance it wants to make a change to a file but the reference string is actually not in the file. So it hallucinated something. Pi will be relatively strict with regards to what it accepts and if the model gets it wrong it will error back to the model and say like hey you did a poor job try it again or differently or whatever it will tell the LLM what went wrong and so you have this in context learning thing going on where hopefully it gets better at calling this tool as its own going forward so is it will to some degree fix up some things because models are terrible at understanding whites space characters so it will allow different whites space characters to appear within an edit match to replace it. So that that's a that's a thing that fixes up.

Seemingly newer entropic models are worse at doing these tool calls and I a didn't expect that also couldn't reproduce it for a while until I found someone give me some data to reproduce it and then I was kind of shocked how it works because what seemingly is happening with and this is a little bit hypothesis a little bit is sort of a thing you can measure. So there's a there's a there's a concept in LLMs called I guess grammar constraint sampling which is if you pull tokens from the GPU >> there are certain probability tokens that you can get and the highest probability token might not be a token that's valid for the position of the text you want to pull. So the simplest version of this is like if you want to produce a JSON object and you know there has to be a JSON object then the first character first token you have to pull has to be an opening brace any other character would be rejected.

>> Mhm. >> And so GBT models for inance represent all the two calls as JSON objects. In entropic models, it's a little bit more complicated like top level tool calls are strings. And then if you have arrays of like if parameter is an array, then it's represent as a chase object. It's a long story short, they're using grammar constraint sampling for a lot of these parameters in in in in function calls which are complex objects.

And so what what is very interesting is that Pi's edit tool is an array of possible edits that it can do. There's like an array of objects. First parameter is usually old string. Second parameter is new string and then there can be a third parameter. I think pi is the third one. No, in pi there's no third parameter. Actually there's only two. but more importantly in plot code there can be a third parameter which is optional which is replace all.

And when you do grammar constraint encoding decoding it means that it samples one token. So that advances it. So let's say you have old string colon the value of the old string and then you sample a comma. That means the only possible valid other string that you can do in a JSON dictionary is another string which is the next key. >> Okay. >> Does it make sense so far? >> I think so. So if I if I pull the comma I have to produce another key >> because I cannot close because Jason ups do not support trading commas.

>> Yes. >> Okay. So that is that is what happens. >> That is the deepest technical I think talk that we've done on this show. >> But now imagine it samples this comma by accident. >> Mhm. >> so now it has to make the option

fit which is no longer valid in pi zel tool. And so we have found out that if you bring the session in a certain state, 20% of all edits that continue fail and it makes up completely random keys that are completely invalid in the tool. So it it says like okay your string is now old string value of the old string new string value of the new string and require unique or tools or it just makes up complete random strings at this point.

>> It's forced to do it. >> It is forced to do that. the sampler from from the output tokens must conform to the grammar of the output language in this case JSON for the tool call. >> Yeah. >> And then you're And the real interesting part is once it has emitted that wrong tool call that stays within the context. So that explains why you can't easily reproduce it because you need a session where it is in context most likely because that then poisons subsequent tool calls because the model then sees in context there was a tool call that looked like this. So I'm just going to do it again or I do I try it with a different kind of additional property that's not allowed.

Now pi can be lenient when it comes to to checking valinity of the output the tool call with respect to the tool schema the input schema. I'm not sure if you have implemented that. >> So it's just not turned on for this particular tool. So if you make this tool sort of accept additional arguments and it would sort of not do anything. >> Yeah. >> But I think the interesting part here is a old models didn't do that.

>> Yeah. So this is a regression on your models. >> Mhm. >> The second thing is what changed from the old models to the new ones. Like what actually changed? One is their training on their own harness. So cloud code is the reference harness now for these models where the older ones trained with some sort of native harness that they had. And I don't know if it's exactly clot code, but at least it's sort of clearly trained on clot codes behavior. But the second thing is clot code at this point is incredibly lenient and I would dare to say a little bit sloppy. So there's no signal in the training process that you did a bad thing here because clot code like I I looked a little bit at a de deminified code just to see what it accepts.

>> Did you do an illegal? >> No, it's not illegal. The fable did not fail. >> Is the US government coming after you now? >> But so like there's no negative signal in the training process anymore because like the cloud harness is so willing to accept a whole bunch of nonsense. >> Yeah. And the cloud code harness has an edit tool obviously as well. Yeah. >> but it only takes file path old text new text and replace all or something.

>> But for instance the cloud code harness if >> used to have a multi-edit. >> But for instance it it clearly says like your parameters are called new old string new string and yet it also accepts new str new old string new like it is very lenient in what it accepts. So they know the deficiency of the model or at least the agent who wrote that tool in the cloud code harness was told there's a problem, right? And then it fixed it by being super lenient and accepting any old garbage the model outputs.

>> I think it could have been you who highlighted this for me, but I'm not sure >> months ago. you know, cloud code will write markdown header metadata with extra parameters that technically might be like out of the spec, but because it does it now, people are modifying, you know, their own applications to support this sort of alternative cloud code spec. >> Yeah, that was a while ago. So, skills, right? Who invented skills? Anthropic obviously invented skills. and there's even a spec and that spec says the the header in your skills file which is >>

yl >> should be yl. So you as an engineer like okay which yl which version not specified. So you go with whatever's the latest and implement that >> and then you have users who bring in their cloud code skills to pi and say my skills don't work.

>> Yes >> why don't they work? Oh, there's a new line in the description field which by YAML's grammar isn't allowed, but cloud code happily swallows that garbage. So now we have a spec, but the company that created that spec, and that spec is wrong, and the the software implements it leniently and doesn't adhere to the spec. So now everybody else has to do has to adhere to the same slop. >> I'm I'm thinking of a parallel to almost like old Windows applications. Mhm.

>> infinite compatibility, backwards compatibility in, >> you know, and and maybe young people don't know this, >> but at some point, you know, you're a games developer and you, you know, the Microsoft DirectX or whatever says it's doing one, you know, one thing, but everybody in the industry knows that you basically have to code around it in a certain way. >> then you have to send the signal via the keyboard controller to get more than one megabyte of my RAM.

>> >> many many because we all very old have seen some >> a little bit. I guess there is a history obviously of programmers you know writing around these problems and certainly I know when we were doing ST century SDK development lots of that I guess what's interesting or different about this time is it's not that anthropic or anybody has designed a spec and you know we're working around deterministic code which is what we were doing in the past. It's that the non-deterministic solutions we have built have just sort of decided to go and do it this way and as a result we are all >> that the sloppy behavior of cloud code becomes a stoastic terrorism attack on all other software products that want to infest skills. Why this is why this I think it reaches a new level to me is that first of all we we have a little bit of evidence now that this is actually sort of getting a little bit worse because like if an if an agent does not fully align with like plot call view of the world then you're going to be punished by that.

>> Yeah. >> But also there's no documentation on it like there's there's no spec for it to begin with but I also don't even know if the people at Entropic necessarily know where the thing is. is fine. Like I nobody ever probably thought about whether the metadata header in a skill file by parsed as parsed by cloud code is valid YAML or not. Nobody gives a So just try code. But I I want to say something to that effect too.

what does it mean for API consumers that send requests to Enthropic with their own custom tools that are nothing like cloud code because they're building agents that are not coding agents but do something else? What does it mean for the error rates of these agents and their tool calls? If Entropic is now basically going all in on cloud code as the RL harness, that's bad. And I think you alluded to earlier that MCP might also have a problem with that approach.

>> If the only consumer of that model would be cloud code, even in that world, cloud code doesn't control all the tools because you can load MCP stuff in there with registers as tools. And so the the quality of cloud code to be able to invoke MCP tools is also I mean I don't have a particularly good statistics on it but I remember at the

conference we were talking that people were exposed challenges with modern models ability to actually also reliably invoke MCP tools because if if your training data does not incorporate your MCP tools but it has crazy amounts of go to the internet and do web searches or launch Chrome or do something like it the the likelihood of the model doing that is going to be higher than the model picking your MCP tool. So there's a within sort of the more you do this reinforcement learning the the harder you're going to make the the the life for everybody who is trying to use it as it sort of is advertised which is like it extends it calls anyone you want and you can build your own future on this model but simultaneously plot makes or entropic makes most money with clot code at this point so the incentives within the company are like >> a little bit I don't know intention at the very least >> so I have I have a related question that I want to ask which seems highly relevant to this which is to go back like Mario said when we he was trying Fable it's like boy it's really bad at this financial stuff right or budgeting >> pretty sure >> I I one of the first thing I did on Opus 48 was someone on Twitter said like you asked it how many which days of the week have the letter D in it and I said like yeah three Monday to Wednesday and I was like are you sure I was like oh yeah Thursday also has a D or like a variation there it's because it doesn't invoke >> it does not invoke for that specific kind of prompt like a trivial prompt but for the prompts I gave it and especially the the data files I gave it a bunch of CSVs and it obviously wrote code in the background that it executed in a small little container somewhere so that wasn't the problem the problem was just that it was a pretty long planning session I I can't say how many tokens because they don't tell you I I regularly ask it for a summary of what's been discussed so far so I keep things fresh so to and eventually it just started deviating from the percentages we we calculated previously and it just couldn't remember them anymore.

That might be an artifact of the the web harness if you want but I don't think it was because I then also took that data and put it into PI with H that wasn't Fable then it was Opus I think. Yeah. Anyways I I wasn't impressed at least if I think about a nontechnical user using Fable through the web interface or the desktop app. I'm not sure about coork but the desktop app is basically the web app.

>> Yeah. So my question was did you see that as a regression from previous models? >> No. >> Okay. I I expect to be there to be regressions. >> Okay. Well, then maybe this question won't make sense anymore, but I'll ask it anyways. Which was, you know, let's say that you are anthropic or open AI and you've identified, you know, some valuable set of operations, i.e. writing code right now seems to be one of them, you know, but there's only so much intelligence, let's say, that you can pack into this. Do you you know, almost like you you let some parts go, right, so that you can reinforce harder on the more valuable parts. And so when you brought up the financing, I was wondering like, you know, maybe there's a humorously an accountant at topic who is like, you know, the math just doesn't work on finance, so we're just not going to focus on that, you know, just put all the GPUs on code. Is that like a is that a thing that happens? That's like my genuine question. I mean we would like if we said we had any in insights there because we don't know about the training machines. We don't know about the mixture between what goes into pre-training, what goes into supervised finetuning, what goes into RL. We we have no idea but we can make some guesses, some educated guesses. The first guess is model size. there's probably still room training bigger and bigger models, but eventually you need to serve that model, right? And then economics of scale hit you. So that means even if you have a humongous model with I don't know how many trillion gazillion parameters, you probably have to distill it down for it to become economical to serve it. Even if you manage to scram more information, more abilities into the big fat model, once you serve

it, you got to distill it down to something that you can actually serve at economical prices. So I think for that reason, it points back to your earlier point. We might be at the top of the S-curve in terms of capabilities.

Yeah, I I guess I was wondering, let's say there's only so much you can you can pack in there, but over the course of deploying this model and talking with customers and doing your forward deployer, >> I I don't think it's knowing how big organizations are, I don't think anyone actually has the power to sort of steer something in one way or another. I think what might actually overwhelm is like what do I have access to in terms of data? Where I'm going with this is like let's say you have a SAS product. Okay, old school deterministic SAS product.

Lots of products sunset features all the time. Not economically viable. We couldn't get enough customers on this, right? But when that happens hopefully and usually or a good company sends you an email, lets you know, hey, you know what? Google product 4000 is shutting down, right? And you're like nuts. But at least you know you can make an educated decision about what to do next. Are we possibly entering a world where you can you know you can basically you know these models serve different purposes I guess is what I'm trying to say right but you you know you could sort of adopt a model for a purpose like finance and a new version of that model could come out and then we just sort of you know vibes identify that it's sort of ambiguous you know like we're not getting a release notes that says by the way we just made this aspect effect of its intelligence 20% lower.

>> So actually we sort of had the opposite problem but for exactly the same outcome which is imagine for a second you have this model. So I think coding is sort of like it's fine we've understood this but then imagine in December and I think we were talking about this in around December was like hey people start figuring out they're really good at security research. It was long before Memphis. >> Yes. Yeah. And then it got so good in security research that if you are now going to get to the latest models, you cannot use it at all because it basically says like screw you. This is getting too risky.

>> I'm not going to serve you. You need to now be one of the hundred companies that the US government gives you. >> Yeah, you're right. >> Yeah. But I just want to come back to the abilities of the model and regressions, right? like how can they evaluate that basically how can I ensure that once I train the newest line of models that they don't regress in specific verticals right and obviously they have a metric ton of evaluations running there and ablation studies and blah blah blah blah but ultimately whatever they have can probably not cover all the real world examples or all the real world usage that's out there even though they have those traces probably as well and I I think it's going to become even harder in the future if they add more capabilities to ensure that old capabilities that we started relying on in our workflows are still there. and I think that's also why they are pushing for you have to use our harness because that's the only controllable thing that that kind of can keep this in b in check and that's a rabbit hole I don't want to go down to.

>> Yeah. Yeah, but I think it's it's it is quite likely that if the models are this is why like I think this this reinforcement thing is like it's really on my in on my mind because like if they're doing more and more reinforcement learning on the harness like they are going to increase the capabilities of that but at the cost of seemingly deviating from this path and so if you're if you have a less capable model that's not reinforced learned to death then you can actually maybe solve some sort of adjacent use cases with a quality that sort of stays within

more or less the bounds of like what the expected path was. Whereas now seemingly if you're getting a little bit too close to where they really trained on but not quite your experience is going to be worse. And my suspicion is that this is actually totally okay with the model providers because they don't have to sell you that model because you want to buy it for whatever reason >> like obviously we have all these you know humanity's last exam and these benchmarks that exist right and I think you were alluding to this which is there you know there's there's a ton of benchmarks however if you think about the multitude of tasks that people are being pushed upon for every random bespoke way that they're using LLMs, they could have this real deviation and you know from model to model and they may not understand it and that and I was just chuckling to myself because the answer is like most other things everyone's going to have to run evals even end users and you know what that's going to cost.

>> That's great. >> Yeah, I I don't see a future where this changes to be honest. obviously all the emails we have available at the moment are basically just pro proxies for capabilities. they will not necessarily translate to your own use cases but they're a good kind of like there's ignoring all the benchmaxing that's going on and most of the coding ben benchmarks are basically useless now as a kind of >> I I think that the coding benchmarks are even worse now in my head because the they basically just really care like did you solve the problem and so like I think that the Sonet 5 release was an interesting one because Sonic 5 actually scores really well on some of those benchmarks but if you look at how much it cost you to actually complete the benchmark >> Yeah, >> it's more expensive than fable because the other thing is like the cost of solving problems seems to be going up rather than down.

>> Huh. Funny how it works. >> That's a good point. We could we we could get into that which is the benchmarks are only capturing one dimension. >> Not all like I think artificial analysis also takes into account cost or at least number of turns or something like that. There there's definitely benchmarks that try to be more truthy, let's say. but but like terminal bench or something like that, that doesn't take anything into account. Not not cost, not runtime, not anything. So, >> we should address the fact that Fable went away. We mentioned it did. It went it went away. It's backed. It It could be gone again.

>> Like, Jesus. >> And it will be gone from the subscription. Don't forget that. So, >> most people goes away. Even that's an interesting topic which is we have been so you know like model provider subscriptions have been one of the biggest ways that people have experienced these models and this feels like that's for the first time that you might not have access to the sort of like frontier model on these subscriptions. Seems that way to me.

>> I think they're price testing really. >> We'll find out. But regardless, when Fable went away, you know, everyone was like, "This is concerning." Concerning period, can people just take my models away? What about open weight models? And then GLM 5.2 came out. A lot of people got really excited about that. What I saw for this window of time was a range of opinions on GLM 5.2. Obviously, there's the benchmarks. It looked pretty good, but people saying, "Hey, anywhere from this is Opus to this is fable quality." Let's let's talk about that.

Is it that good? Is it, you know, should everybody >> You give your vibes first. >> I'm kind of tired of that garbage. Can it make 3GS scenes? Yes, it can make very nice 3Ts scenes. The question is, does it does it work

with my tasks? And just like the models of the past three point generations, yeah, it works. It's I I don't feel a lot of difference, but again, that might be down to my workflow, my code bases, and so on and so forth. your mileage may vary if you do more complex things than I'm doing. ultimately, I think Armen's always saying open source wins, always wins. models aren't really open source, they're open weights. That's quite a bit of a difference. so it's really hard to reproduce them. It is also very costly. Even if you had all the training process and data in hand, you could probably not afford reproducing that. So it's far cry from open source. Ultimately we are still dependent on a bunch of labs and their kindness or their let's put it that way shenanigans with respect to market by throwing those out for free basically. but again I'm a big proponent of open weight models and I think a lot of people not only in Europe but also the US have started to wake up to the fact that it's not healthy.

>> Okay. And Alex Karp now >> goes and CNBC to complain about closed weight models. I have not a lot of opinion on GLM. I think like there two things that I noticed. one is like it likes to I think it has maybe sort of similar kind of vibes to some other models with regards to it just likes to succeed no matter the cost. So it will force push, it will delete file, it will do a whole bunch of stuff that's maybe a little bit out of my comfort zone of what I want a model to be. And so it's actually probably a really good programmer. But I also found DeepSeek to be pretty good. Like Deepseek pros actually to me felt really good. And from that experience GLM, I didn't see a huge difference. But I also don't loop like I don't really do any of those things.

It's fine. Yeah, I I think for the workflows that at least we have established all optical models of right now which is kind of good. >> I think an interesting thing is that KLM is significantly more expensive than deepse pro. This might be that deepse pro is also sort of subsidized a little bit right now on the providers but clear is seemingly like between three and four times as expensive in practice than deepseek prosec pro. So is it is it four times as good?

I think I got an open code go subscription just to use it because for a brief period it was only on go and then I eclipsed that limit very fast and then conveniently by the next day it was on the pay peruse on Zen and so then I was just paying per token and I felt like it was going pretty well. I guess this is just more vibe stuff. I didn't think it was the second coming of Christ. like this industry needs a story every week, you know, and conveniently GLM had, I guess, like a moment where this story was really good for them. And not to take anything away from the model, but it is not the second coming. It's it's a good model, right?

>> in a way, not much has happened the entire last year other than the models got a little bit better. We figured out that we can put the >> agents into like different setups like can put them into >> Discord or Slack >> Discord or Slack and they can do a bunch of different things. But like on a very fundamental level even this discussion about looping and orchestration whatever was actually there pretty much exactly one year ago >> sort of >> just not not to the same amount of people being involved not to the same amount of quality of what they can do with the model. We learned a little bit in the process but actually somewhat somewhere we have been there >> nuance. what what ch is that the correct pronunciation of his name? Choff Jeff Joff >> Joffrey >> Joffrey what he did with Ralph loop was just basically create a pattern that has its uses like all the research the thing called parsy and then later Toby from Shopify codified that kind of came from the rail loop I would say basically set up a goal for the thing and let it do it the thing for as long as it needs to with some verification at the end that that it

reached the goal and that's a super useful pattern like I've used this quite a bunch now for performance optimization ations for I don't even remember anymore that that's one style of loop right and that to me is known good I would use this pattern I have actually used for that pattern interesting enough I think the Ralph loop came out when in June last year >> I think >> and then it had a renaissance in November because >> I think because they put into the marketplace >> yeah something like that like >> also with the original name >> right and and and then Now you have you have Boris and you have Peter talking about loops and now everybody's going crazy about loops and if I understood correctly I engineer in San Francisco last week was basically all about loops right there was a lot of loops.

>> So was there ever a good definition of what a loop is? >> I mean I think they're basically two loops. There's the inent loop that sort of naturally ends and then there's the harness level loop that keeps the damn thing going until some externally defined condition arises >> which is basically still what the raw focus I don't think has that dramat like I think the main thing now is like is actually cues in a way people put a bunch of tasks into some sort of system where it picks it up and then starts looping >> right >> so what Peter was presenting at the AI engineer I think it was like a 15minute sequence where he kind of kind of sort of went into how his loops work and it's basically an orchestrator that then distributes work to other sub agents possibly based on an external trigger an issue coming in a PR coming in and stuff like that and that to me is the new style loop you have external events that trigger a predefined thing basically >> but I will say did you ever look at trophy's programming language that he built in August last year >> I think I looked at the website was discussed it and went away >> but as far as I remember it was basically one Ralph loop hurting issues over and over and another Ralph Luke busy picking up issues and troing them over >> which is I think in a way sort of maybe a like then you have beats which is also basically >> people are talking about this seriously now and I think like even Peter like that you didn't expect to do what like sort of say like Steve Yagi was right but that's literally what he said and I think it's like it is obviously performative art to some degree but like the underlying principles are not completely wrong it's just >> I am not a looper name is Ben. I'm not a looper.

>> I Yeah, I am a insane ADHD, you know, multiple thread conductor. I think objectively I do produce a lot. I also take, you know, people use the software. I think that's worth something. But and the way that I achieve it is mostly like, you know, it's almost like spinning plates, right? Like, how you doing over here? How you doing over here? Okay. All right. All right. Oh What? What's going on? Let's fix you. Right. Okay. it's not healthy.

There's certainly some world where, you know, if I could relax more and use loops, I was interested in that. But I just don't know how to achieve it other than for the problem that you highlighted, which is with some deterministic outcome. Right. So I do have for almost all my open source projects and closed source projects performance benchmarks, unit tests, right? Punk, for example, has like 13 different tests tracking memory growth and, you know, startup time and scroll time, right? And part of the reason why it's even mildly fast is that there was a whole bunch of auto research loops in pi that was like improve this your go you're right totally works what I've yet to see or what I don't understand is like I understand how to construct loops I just don't know for what kind of tasks or for what kind of outcomes other than like a deterministic numeric one you am I missing out on?

>> Maybe I'm misparrasing Peter here, but the way I understood for example is something like you have your issue tracker. You might go into the issue tracker or you might tell your agent go into the issue tracker, identify a bunch of issues that are valid, open them back up and distribute that work to some other agent to then process that issue and come up with a PR for the issue. So I the human at the end can just go through the PRs see if they're good and say okay merge this don't >> right >> I I have tried this a long time ago and the models might have gotten better but I don't trust that this leads to any kind of maintainable software at all. I understand this better though because what he's doing is he's saying I don't trust what you're creating but I put some you know some value in that you may have identified something so I'm just going to regenerate everything in the tools that I do trust >> yeah but I don't trust the tools that he trust and I think that is I think a little bit of the the delta here because so one one thing friends is that I think I don't know we already discussed this but like this the simplest loop that I can imagine that they can conjure up is that task comes in, generate now start from scratch, review it, feed all the review findings back in the generator until eventually the thing achieves unstable equilibrium where the other satisfying >> because it always finds something that needs improvement >> and that produces the most complex and ungodly code possibly imaginable >> from my experience so far. And yet that is a loop that a lot of people are running.

>> Yeah. So, there's probably a version of this loop that's not quite as crazy. And I think one of the suggestions that I heard is like you should be running the review but without thinking because if it has lower reasoning budget then it doesn't hang itself as much. but like it's all kind of >> like like I don't really yet understand which tool I can actually use that I trust so much that because you said earlier like it's a I don't you didn't say anxiety inducing but it sounded a little bit the description of like doing orchestration on that level is a little bit like you wish you had more free time and maybe the loops give you the free time or like >> was you were thinking last year >> this literally like the more people started doing all of this the more I felt like all this great free time that I had a little bit is like completely gone at this point. It's just pure anxiety and so the loops are not in any way a future. It feels like fills me with the idea that it's going to get back to like being in control. I don't know. It's just >> I think there's another another side of this and that is cost >> like the more you have running in the background the less controller you spend basically you have. And the question is people promoting this kind of workflow they obviously don't have any limitations in terms of tokens.

>> Maybe open AI is >> I did like Peter's Peter's talk was literally like the first was swap I solved this by joining OpenAI. >> Yeah. And then he went to the next thing which is like now my only problem is orchestration. >> And then and then the question becomes the like Peter obviously his system works for him because openclaw is still running as it should and it's super big in China still. I I just recently had the talk with two native Chinese people and they were like oh my god it just changed the entire landscape in China.

So I I'm not disputing that it works for Peter. I just wonder does it scale down to us? Does it scale to your run-of-the-mill enterprise company that may have a software department or which is mostly software? And given today's pricing, I would say it probably doesn't. Specifically when you then compare it to the actual return on investment you get like does this produce things that increase my revenue stream or reduce churn or whatever, right? And I question whether that's the case. I mean, for an open source project like Open Claw, awesome. If OpenAI basically sponsors the tokens to use to keep Open Claw running and becoming better and more secure specifically, which is what they've been working on, that's great.

Love it. I just wonder OpenClaw itself doesn't need Roy per se. Like they don't need to make revenue and they don't have any liabilities either. which doesn't mean that Open Claw is garbage or anything because they've put a lot of work into security over the past couple of months and that paid off. It's just the incentive structure is different here. And and I don't know, I kind of feel like we're being sold a version of how things should work that are not to our benefit, but to the benefit of people who sell us tokens.

Let's put it that way. >> Mhm. A common theme of this show. but so far it hasn't been completely wrong. I think >> and and I think this this one of the ways in which this would sold on the token factory right now the dark factory is that well we have reached a point now where the the the we have reached a point a couple of months ago but the the general point is like now we are sort of piling up on reviews. It's like and and so the the answer is like review less, right? That seems to be sort of the model right now. Like we have we haven't solved this problem with the reviews.

>> So now the recommendation is just like not everything that you're shipping needs to be reviewed. And I don't think that's actually a solution. That just happens to be >> I I personally do think though that a lot of reviews in the past before agents were purely performative and weren't actually reviews. Here's my nitpick. >> Yeah. And I I also agree that you don't need to review every single line of code an agent generates. I think that's not that cannot work, right? I usually give the the example of PI as an HTML export. I have not read a single line of how that export works. I just look at the export and it looks correct or it doesn't. That's it. I don't care.

It's not mission critical, right? So, and and some companies are actually now going full in on the dark factory pattern like factory AI. It's in the name obviously and you will have their marketing material and then on a Twitter thread you will have Eno who I really like. He's a great guy. I had a call with him a while ago where he's like yeah obviously it needs a little bit of setup first it work out of the box and then I wonder what does this entail and what are my results after the setup and I don't know I I'm not ruling out that this is going to be a part of the future or the future. I wouldn't want it to be the future for for for various reasons, but I don't see it at the moment.

>> Just had a random thought on like reviews and maybe maybe it's ambiguous because sometimes we're talking like human review but also agentic review. randomly thinking I've actually had quite a bit of result results using dumb models to do reviews and my argument. I don't know. Sometimes I like to think about what would this look like with humans and you know sometimes having junior employees look at stuff is half of it wrong half of their suggestions make no sense do they occasionally maybe because they're just you know adultled on something else do they identify something that maybe you wouldn't have noticed also true >> it's also a way for them to learn >> yes this is also something I experimented with this with fable I would explicitly say like run a haiku sub agent review. Boy, did that find a lot of bugs, which also made me think, what the heck is going on? Fable, right? I don't know what the point of this was. I guess I do want to come back just to punctuate last episode, we talked a lot about this isn't 7D chess.

We have a company that benefits a lot from you spending more tokens. isn't it convenient that a lot of these products seem to come out that consume more tokens and that the solution for all of all of your ills is spend

more tokens? And we have two things that came out actually that do this, which is Fable, of course, but also Sonet 5. We touched on this earlier, but we didn't connect it in this way, which is it's better purportedly. I mean, it probably is >> than what Sonnet 47 at least on some benchmark, but also surprise. It also seems to, you know, use more turns or >> it also cost as much as GPD 5.5, I think. And it's definitely worse than GPD 5.5. Who is going to >> Yeah, I mean, but it's not the only thing that actually cost more what tokens now. Like so the Swbench as a good example there's websites where you can sort of see people running them independently and also give you like token cost. So basically cheap models are falling away because they're being replaced by slightly more expensive cheap models.

So like the the cost point that they got at 1.4 for haiku 3.5 has not of course there's like deepse flash so there's a cheap model appearing but the flagship cheap models are the cost to solving a problem actually get more expensive and so >> yeah you've seen this with Gemini flesh as well >> yeah or just SAS products >> like you can't go back in time and use the version of Netflix that was totally great that cost half as much >> Yeah I guess >> we have a little bit of an inflation problem right now.

>> Just you know, yeah, tying it back that these are not even some grand conspiracy. This is just you know traditional marketplace dynamics. >> I remember in the early beginnings of LLMs available to us perhaps you had things like model checkpoints. you you could select a specific version of a specific model from a specific date and use that in your production software or backend system whatever and I think they're still kind of doing this but it kind of slowly goes away and you kind of push towards just use the latest whatever right it's going to be fine and we know it's not we see regressions >> and another interesting bit was 20 I think around the end of 2023 and beginning of before there was a large push towards do your own supervised fine-tuning on our models and then just a couple weeks ago OpenAI actually turned it off. So you can also not customize them anymore probably because SFT is really hard to do well and the outputs are usually garbage anyways and but but still it just points in a direction where they're trying to become vertically integrated. They want to they want to own the entire stack probably because they know that just selling tokens is not enough. Our friend David Kramer highlighted the fact that if you're working with agents, you benefit a lot from determinism. We we talked about, you know, hey, loops with a deterministic output, you can have actually really great results because you're just smashing tokens till you get the answer. what that what that also mean so that can be unit tests, that can be benchmarks I alluded to earlier. However, to drive all those things, you also need compute. They're not tokens. And we've historically thought about at least in the last few years is compute as being increasingly cheap. However, compute is also going also.

>> We talked about we shouldn't mention the price point of like how much this laptop is. >> You know how much this laptop is now? >> Yeah. Yeah. We we've talked so much about computers being expensive. I don't even think we were prepared for literally devices in your hand. It's almost like used cars when used cars went through the roof. I don't know if they did here, but >> I don't know if we discussed this on this podcast, but I discuss I did talk with someone else about recently. We just I made this sort of so in our bubble clearly this tech is amazing.

>> Yeah. How much? >> Well, this is 10 grand now. Can you imagine? Right. >> But I mean like the tech of AI is

amazing. Like there's no there's no doubt, right? You have no doubt about it. >> You know what my son sort of found out via listening to news on the internet that AI makes the switch to more expensive. >> That's where comes from. >> And and I think this is sort of the fact that you have a child who doesn't really care about AI in any sort of meaningful capacity.

>> Yeah. but he learns that his friend is not going to buy a switch to now because it's more expensive because of AI. Like the like there's a there's a sort of negative association with this thing now. >> I think that's great. I mean, if you look in the future, man versus machine. It's great that our youth now built up this internal hate towards the things. They also had this they had this like I don't know I think was it Forbes or is it like Financial Times where they had like a bunch of surveys of like age groups by like how much they like AI.

It's actually the older ones that like it and the younger generation >> I think that makes intrinsic sense because you think about it the older generation white collar workers they love this because now they can can let the AI AI build all their useless slide decks and chop materials. That makes total sense to me and I I actually talking to people in our in our neighborhood that that's what they're telling me basically. It helps them be lazy at the job which is great for them.

That makes total sense for me. I think it's it's interesting that this like we are obviously a bubble but like our little bubble sort of destroys the world economy right now in in not in a way it's like oh AI makes your jobs use less but like we're building out so much capex for data centers that all of a sudden like your computer costs more educational computers cost more consoles cost more electricity costs more >> and the market crashes >> and your pension's gone >> so I mean this is this is weird >> I may have bought a switch too as a response response to in cool scene prices and when I was looking at getting an SD card to install in it, they are also more expensive. So, here's a question for you and I guess you know now we're now we're turning into predictions because we've been talking for a while, but is this kind of the beginning or is this the peak or is this sort of on the way up for you know just compute as a commodity, right?

We've talked about sovereign AI and I'm going to bring it back to what Mario brought up which is what does it matter that you have an openweight model if you actually don't you can't afford the hardware to run it doesn't matter right does that trend continue because the value of what now people are realizing with computers is going so high up that is just going to like never mind sovereign GPUs frankly people should be you know should I be hoarding a bunch of AMD Ryzen's in my computer, you know, in my basement because I don't know where that's going to be. Or is that just contributing to the problem?

>> I mean, hoarding definitely contributes to the problem, right? >> The moment I said the word hoarding, I think that that answer the question. >> Big product out of the market, prices increase, so it makes sense. but I I'm convinced that the wet dream of all of these people at the moment is that we go back to the 70s where you had just a terminal and compute was basically not something you own own but something you rent and I have a feeling it's going to go that way. The only problem with that is that everybody's addicted to those little rectangles we have in our pants and they are also affected by the increase in DRAM prices for example.

and that's not going to play well. Like you're not gonna going to be able to take away phones from people, but I have a feeling that everything else is going to be >> That's a great insight. I want to say they won't because the value derived from this compute depends on the fundamental availability of the rectangle and this machine being distributed in people's homes. So there's some fundamental, >> but here's the thing like this is a production machine, right? I can do stuff with this. My phone anytime I have to do any actual work on it, I hate it. But I think what we're going to get, this is now conspiracy tin head for tin tin foil head territory. But my guess is the wishes of the grandmasters.

The wish is you're only going to use your phone. We're going to make your phone real cheap, but you're going to sign up for our AI services that can then do your work for you by just giving them prompts on your phone. And you will never need this again. Yeah, we're g we're gonna turn this thing that you can buy for a onetime payment. We're going to turn it into a subscription service because you now need a computer in the cloud where your agent can run and do all your work. And I think that's where we got to go.

>> Thin clients feel like they're coming back. I want to give one quick example which is X the everything app. >> Oh yeah. introduced they've had a live streaming component for a little while, but the way that you did that is you ran sovereign software like OBS and you streamed that to their platform. Now, this is actually being the actual sort of like recording capabilities are now part of that platform and you're going to go on the web on a thin client and you can use it that way.

>> Yeah, I'm not over. I I so I had a conversation with Idan Gazit >> with that >> who works GitHub next I think he's head of GitHub next and he he brought up an interesting thought there are maybe a little bit as an anti-pieces to the whole thing so he was reminiscent on a time when computer game developers basically paid for incredib so incredibility is a thing I also used which is it took us too long to compile Unreal Engine So every machine in our gaming studio had incredib build installed and when someone want to compile it distributed the build across everybody else.

And so if you do the math it actually turns out that in some office there are enough probably sufficiently large Mac chip is sitting around that you could sort of cluster up and actually have a local weight model do pretty decently if you get it to run decently well. And so in terms of like random compute sitting around in different places, take all the Tesla cars standing around full packed with GPUs for self-driving, there's actually enough deployed infrastructure outside of data centers that you could actually run a bunch of models and eventually that like even so in your version of this where the computer goes away that's a problem. But in a world where there's still going to be a bunch of fuel computers, there will be a surplus of deployed GPUs and maybe eventually someone is going to build a distributed inference engine that will harness all of this stuff.

>> I like the limits of physics will throw a wrench in that. I think >> I know I think like maybe we're not all going to pluck them in but there's not that much like for like the anti thing where he distributed deepse flash I think pro across two machines >> all with lightning connectors >> no no no >> because the actual on the way he probed it the actual data that he had to exchange between the sort of the checkpoints that he did was megabytes not meabytes >> interesting >> >> I mean more power to that kind of effort I there they still I can still see a

world that's not completely making us dependent on this. I have been thinking this during this conversation about like Cyberpunk 2077 where we're all just gonna start killing each other for like the you know the newest you know hardware upgrade or >> so there's like on this sort of thing is like there's like everybody like prices go up because like obviously there's a demand and then hopefully production goes up and then there's a glut and then the prices collapse. So presumably it's going to happen here too, right?

It doesn't seem like we're actually going to need as much stuff as we are probably in the process of building out. >> There could be a correction. Okay, so we've just been kind of rambling about random topics and maybe coming to some of our favorite things like tokens are expensive. we have one question here from Chewy Distraction Q. How to handle FOMO? Do you experience FOMO?

>> No, not anymore. I got my eye psychosis out of my way last year while being funemployed while everybody else was ignoring agents. I think the easiest way is to just look back at the last six months or so and ask yourself what really changed in your day-to-day and then ask yourself if it makes sense to be kind of like this and getting all news every minute into your system and whether it has any impact on your day-to-day if you do this and my prediction would be it does not. So, I think checking in with the state of things every month is probably enough.

>> That's all well and good for you, Mario, because you're literally at the at the you know, you are the creator of FOMO. You are leaving a trail. >> I'm the creator of FOMO. I'm sorry. I'm just the creator of Flask. He created the FOMO. No, I think I mean ultimately it's still a while loop with an LM request and then a tool execution. Nothing changed about that. So, >> yeah, I don't know how to handle one. I think is the problem. I think that to some degree is like it's a question of like how do you deal with your own emotions.

I think the suggestion is the correct one which is I also have given which is just don't read the news all the time. >> What >> if something is interesting it's going to be there in two weeks and then if you're two weeks late it's not going to be a massive problem. but despite me saying that how how good am I actually managing my own emotions is a completely different question right. I think when I went to AI engineer like on the first day I was like I I felt like I don't know what I wrote. I think I wrote something in the discourse. I'm like >> oh yeah yeah I I don't want to reproduce it.

>> But it was like because like you're you're sort of you're showing up in a place where everybody sort of lives the future and it's like crazy disorienting and then like a day and a half and it's like okay they're just pretending to live in the future. But but there's I think that actually I think Dax the guy from open codes not the one from human layer actually say this reason which is like we all live in a in a in a world of like uncertainty and a good way to feel better about it is like you're pretending to know the future and telling it to everybody else. So you build up yourself there. Maybe that's one way to deal with FOMO but also just recognizing that probably a lot of people are doing that. It's like nobody really knows what's going on. Some hunches maybe. But I think we'll find out at the end of the year. I think by that >> So you think by the end of the year we're ready.

>> That's my prediction by the way for the end of the thing. My prediction for the end of the year is we will have found out what the end state of all of this carnage is. >> We should predict when the bubble bursts. I think

that would be that would be a good >> as soon as Entropico may Io. I I think the IPO is basically like the good thing is like I think it's relatively safe point to say like a little bit after the IPO is going to be when this all plays out. But >> on the other hand, we have SpaceX who still haven't gone down to the actual dollar amounts they should go down to.

>> But they only issued 5% of the stocks. Ben, when is when is the bubble gone the worst? >> Tuesday. >> Good. I have no plans. Well, should I shut down here? >> Yeah. >> All right. Thanks for being here. >> Thanks for having me. >> Okay. >> Bye.