

The 7 Most Powerful Moats For AI Startups

45 MIN · YOUTUBE · [HTTPS://WWW.YOUTUBE.COM/WATCH?V=BXBZSSSQAM](https://www.youtube.com/watch?v=BXBZSSSQAM)

<https://www.youtube.com/watch?v=bxBzsSsqQAM>

SUMMARY

The podcast episode discusses the concept of "motes" in the context of startups, particularly in the AI sector. It emphasizes the importance of having a defensible position against competition, especially as new AI technologies emerge that can easily replicate existing business models. The hosts explore the seven categories of motes from Hamilton Helmer's book, "The Seven Powers," and how these concepts can be applied to modern startups.

- *Motes are essential for startups to defend against infinite competition, particularly in the AI space.*
- *Early-stage founders should focus on solving real problems before worrying about long-term motes.*
- *Speed is highlighted as a crucial initial moat, allowing startups to outpace larger competitors.*
- *The seven powers (or motes) include process power, cornered resources, switching costs, counter positioning, network economies, and scale economies.*
- *Process power involves creating complex systems that are difficult for competitors to replicate.*
- *Cornered resources refer to unique assets or data that provide a competitive edge.*
- *Switching costs trap customers in existing solutions, making it costly for them to switch to competitors.*
- *Counter positioning allows new entrants to offer products that incumbents cannot replicate without harming their existing business models.*

This idea of motes is so pervasive and so important. >> It is interesting how motes have just become much more discussed by aspiring startup founders now than they were so pre-AI. What is going to prevent you from being basically subject to infinite competition? >> Like a mote is inherently a defensive thing and you have to have something to defend. Otherwise, like >> If you're nothing Yeah, yeah, you're nothing to defend. Don't worry about your motes. [Music] Welcome back to another episode of The Light Cone. Today, we're going to talk about motes. So, in your head, you might be thinking about barbarians storming your gate. You've got this little startup and you've got every other company out there who wants to come and eat your lunch.

Uh and, you know, right outside your castle is a mote that keeps them away. Jared, when you were going to college campuses, this isn't sort of this trivial thing that people are thinking about. It's actually uh something that keeps them >> from starting companies right now. Yeah, this is a question that we got from a lot of very smart college students on on our on our recent college trips. And basically, their question is like, they don't see how these new AI agent companies, like a lot of the ones that we've talked about on on this podcast, could have motes. Um it plays into this meme of like the ChatGPT wrapper that like all of these companies could be easily cloned. And so, they can see how you could build a business that makes some amount of revenue, but they don't really see how you can build a long enduring business. And so, I think >> it's actually not true. I actually think these businesses do have quite deep and interesting motes, but they're not totally obvious what they would be. So, I think this is an interesting topic for us to to explore. At our recent AI Startup School backstage, I had this exchange with Sam Altman that I thought was kind of funny. You know, we spend a lot of time thinking about,

you know, make something people want. Very simple maxims that are sort of anti-business school. And yet, this idea of motives is so pervasive and so important. We sort of remarked how funny it is that uh one of the more important books to read these days is actually business school fodder. Um this book called The Seven Powers. So, today we thought that we would actually go through those seven powers. What are they? What are some concrete examples and ways that a startup founder who's just starting out uh could or should be thinking about these things from real-world examples that we've seen. So, Diana, can you tell us a bit about this book? This book was written by Hamilton Helmer, who taught at Stanford Economic School and was published in 2016.

And the book title was The Seven Powers, The Foundations of Business Strategies. And a lot of the examples are more with the era of internet companies from the 2000s. So, a lot of the examples are like Oracle, Facebook, Netflix, which is a older generation. So, we want to do a bit of a reboot right now, how it applies now 2025 with AI. I think it's a little bit confusing the way he uses the terminology in the book. It's called The Seven Powers, but it would make a lot more sense if you just called the thing The Seven Motives, cuz that's really what he's talking about. He's really talking about seven categories of motives that a business can have. And it's true that the examples are out of date, but I think the framework is actually pretty timeless.

Like it turns out, there's just only so many kinds of motives that a business can have, and they don't really change. And so, like even though the specific like versions of these motives are different in the AI agent world, like the categories haven't changed. Thankfully, we live in a world where there's markets and there's free markets where there's lots and lots of competition. And these motives in a lot of ways are the only way, if you're running a business, you can sort of fight against all of the other people who might want to do exactly what you're doing. And um you know, famously, Peter Thiel talks about uh competition is for losers. And so, the profound view there is that given infinite competition, what is going to prevent you from being basically subject to infinite competition? And then, as a result, uh you know, your margins, how much you can actually profit off of what you're selling, goes down to zero. And what that means is like actually your business will die. And so, you know, having a motive is uh relatively existential eventually. You made a great point earlier, Gary, that like this is actually like you kind of have to worry about this at the right time of of a startup. Do you want to talk about like how like early-stage founders should think about motives? I mean, this is sort of why we generally tell people to go find a person with a real problem and then go solve that problem first.

It's um what's funny about the world uh that's a little surprising is that you can go almost anywhere and find some pain point, some problem that could be solved with software and especially with AI that frankly just isn't being solved. And if that and they're they're so numerous and so severe that if you find that thing and solve it, you literally can mint a billion-dollar or 10 billion or even hundred or hundreds of billions of dollars uh market cap business.

And it's just lying in plain sight. That's really the first thing that people should do. Like you should just find a problem and go solve it. And then, along the way, you will probably, as you work with customers, as you build the product itself and engineer it and figure out what data you need for it and all of these things, like you will stumble upon these seven powers. >> Yeah, the motives come later. Like it would be like pretty dumb for somebody to decide not to work on a startup idea because they can't see what the long-term motives of that idea

could be, right?

>> It is interesting how moats have just become um much more discussed by aspiring startup founders now than they were pre-AI. Seems like the main reason for that presumably is just that bigger the original ChatGPT wrapper meme and that the moat that most people are worried about is moat against the big model companies and how like are you not going to get crushed by one of the big labs when they decide this product you're working on is really valuable and they want to own it, too. And I think Varun uh from Winstorf, who we hosted some time ago, he said it himself, the early stages at the beginning, the only moat that startups have is really just speed.

Once you pass that and build something that people want, then you figure out and go deeper into these type of moats that we're going to discuss. I really like Varun's point that the only moat is speed. That is not one of the seven powers in the book, but I think it probably should be. I think it also comes with a lot of the SOs from PD because one of the tenants really at the beginning is, yes, you're a big company, let's say OpenAI at this OpenAI is a new Google. It's like, sure, OpenAI or Anthropic could build all these features, let's say like Claude Code, and then compete directly, let's say with Cursor or etc.

And for a startup like Cursor to really win even in the beginning is they had relentless execution because a larger company like a Google or Anthropic, it they just have a lot of more craft that they need to do in order to ship a product. They just have all these product managers, all the operations, it needs to go through a PRD, some spec doc, and it takes more a lot more time to ship a feature as opposed to Cursor. The incredible story about Cursor, when we hosted Michael Truel to come talk to the batch, he was sharing how his product development cycle for shipping features and sprint cycles were one day. One day.

So, one-day sprint. In the at the beginning during a 2023, 2024 around era, they would start the every day with restart the clock and try to ship things every day. I mean, that's like insane speed. Like there's no big company that could ship something at that speed. At most, weeks, couple weeks, and maybe that larger companies, I don't know if you're Google, maybe like multiple months or sometimes years. I mean, they had Google Bard or Gemini a long time ago. That took years to get out, right? I think Cursor and Winstorf are great examples of when you should start thinking about the moat because for the first few years, I don't think it really mattered that much. They just had to like they proved out that, hey, like code gen is going to be a really valuable application of AI.

The development environment is going to be very, very important to own. They like got rapid growth. And then, it's only when they were at scale that, you know, like they have to start thinking about how we're going to defend against like Claude Code or Codex or all the other things coming in. And so, like the mental model that's really stuck with me is when we spoke to Bob McGrew a couple of weeks ago. Um and how I think, Jared, you brought this up, actually. Was it one way you could think about it is that sort of all of these startups are kind of forward-deployed engineering teams like for the labs, maybe. Uh and so, like early on, actually, because this is all green field, we don't actually know what the valuable verticals and products to build are. So, in a sense, you don't Step one is to figure out what that is.

And it wasn't actually even 2 years ago, it wasn't actually clear it was code gen or um the IDE. Once you figure that out and you find any sort of struck gold, then you keep digging. That's when you have to probably assume you're at some point you're going to get more competition because people are going to realize, oh, this is really valuable. There's lots of money to be made here. And then, you have to start like defending like the treasure you found.

>> So, I mean, all the things that we're about to cover, aside from speed, are sort of one to a billion, one to 10 billion, one to a hundred billion, one to a trillion-dollar sort of problems. And then, uh the real stupid thing that people might do is watch this and look for this as a reason to not even get to one. Yes. So, that would be probably >> to use it to like pick between two different startup ideas because they're like trying to forecast 5 years in the future which one will have a greater mote. Which just isn't how it works. I mean, literally, you shouldn't do that.

Like a moat is inherently a defensive thing and you have to have something to defend. Otherwise, like Maybe you have nothing to Yeah, yeah, nothing to defend. Don't worry about your moat. Yeah. Otherwise, it's just like a puddle in a field. Yeah, yeah, exactly. Let's assume that someone has found something that's valuable that is worth defending. Should we talk through what some of the moats they they could think about are? Yeah, so process power. Again, like the terminology is kind of funky, but like basically it means you built something that's like you built a very complicated business with a lot of stuff that's just hard for people to replicate just because you like built all this stuff.

And so the example that he uses in his book is like the Toyota assembly line. And I think the AI version the AI agent version of this is just a really complicated AI agent that's been like finally honed over like multiple years to work really well under real-world conditions. We've we've talked about a bunch of these on this podcast. Like Jake Heller with Case Text is like the original example. A couple other ones I was thinking about from more recent companies, we have like a couple companies that sell AI agents to banks. We have Greenlight who worked with Tom. They do KYC for banks. And we have Casca which like does loan origination for banks. So it is essentially tells banks like which loans they should give. And I think these are interesting examples because for all of these AI agents you could build a version of Greenlight or Casca or Case Text like a like a demo version in like a weekend hackathon. And I think when college students are thinking about these AI agents, I think what they have in their mind is like the weekend hackathon version of the product and they're like like I could build that in a week. Like how could that be defensible? And like the reason is like the the version you build in a hackathon isn't useful to anyone. It's like like like like if Casca or or Greenlight fail, like the the banks will lose millions of dollars. This is like mission-critical infrastructure. It's it's it's more like a self-driving car.

One way to look at it is way better engineering is actually that's like the most profound form of process power. Like one example might be Plaid which you know, the surface area of the number of financial institutions that they have to support is so giant. It's you know, probably on the order of thousands to tens of thousands of different different websites, different crawlers. And then all of the different you know, can you imagine like Plaid's CICD structure? And then you know, this is pure speculation, but if I were Zach running Plaid, like I you know, know that I would want to be using code gen tool the latest code gen tools to be able to you know,

basically add every new financial institution on the planet quicker than anyone else. Like that's sort of a very profound form of process power in the modern AI age.

>> I think this is probably the main form of defensibility for the existing SaaS companies. Like if you look one generation before the AI agent companies, like why is Stripe or Rippling or Gusto defensible? I think it's mostly this, right? It's just like they've just built a lot of software and it'd be really expensive and hard to replicate all of it. And like you can't just copy it from their landing page. Like there's like like the back-end logic is like super deep. There's also I feel like kind of a schlepp blindness aspect to this going on too where like the the hackathon version of any AI tool is like quicker than ever to get to, but actually the last like 10% of getting it to work reliably across like tens of thousands of KYC requests like per day is sort of like a particular type of painstaking drudgery work in a way that I think like lots of engineers are just not excited to do. And then that is also kind of like the teams at OpenAI are going to experience this too, right?

Like if you're if you're working in one of the big model labs and there's teams of people trying to invent AGI, it's going to be hard to get jazzed about nailing the like final 5% consistency on your like KYC tool. Yeah, and so I I think this is especially true for like verticals like KYC that are require specialized knowledge to even know what to even to know what the edge cases are. Like if we had to pick from the seven powers, like I think speed and this these are probably like the two dominant ones that come up the most often.

And those are most related to execution. Is where the hardcore builders win. Having really good product taste and building the best product really matters. And I think it comes to a lot of the point maybe the the misconception is I think a lot of these products you can probably build the 80% solution with 20% of the effort. But for these solutions and products to work, you need the 99% accuracy one which then takes like 10 times or even sometimes 100 times the amount of effort, right? Sort of that Pareto principle type of thing. What about the the other power for cornered resources? I think the classic view is they're just coveted assets or things that you know, they're not arbitrageable.

They must be independently valuable. And then you sometimes they offer preferential access with you know, rates that are way lower. So the classic example that you know, you could look at is you pharma companies have these patents that are very hard to get. They have to come up with them and then prove them and get through regulatory approval. And the sheer fact that they have a patent plus you know getting through FDA approval is something that can be very durable. And it's you know, so powerful that patents have limited lifespan because you know, you don't want people to have that forever.

A more modern example I think you're on the regulatory side might be you Scale AI is doing a ton of work with the DOD. You know, Palantir as well. In order to even get there, it's you know, a painstaking process. You you got to hire the right people. You got to spend a lot of time in DC and Langley or wherever you know, you're trying to sell to. And you've got to literally build um skiffs like these like sort of you know, special data centers where you know, it's at great pain and expense.

You have to get embedded with the government. But then when you do like well, you've got it. You know, the

cornered resource in some sense is even the brain space in people who work in the government. Like you know, right now if you're working with AI, like you've got to go through a Palantir or a Scale and that's like literally written into their public documents around like how they're thinking about the nature of warfare and the nature of you everything that they want to do having to do with AI moving forward. So you know, the cornered resource doesn't have to be a diamond mine. It could be the diamond mine in your customer's heads. Those examples are sort of uh closer to like being way up in the sky having this like insane decacorn like worth hundreds of billions of dollars sort of situation. But what's relevant for startups that I think all of us sort of see every day is sort of what you were mentioning with this forward deployed engineer you know, FDE forward deployed engineer model that that is what a lot of startups that are extremely successful today are literally doing. Like they're going out and getting a cornered resource in the form of real data and real workflows.

Literally sitting with a customer who normally would never get access to good software. And then spotting okay, this is sort of the tailored time and motion. You know, first the you know, a request comes in by email. Then we take this and we enrich it in this way. Sometimes we have to have a call center call this person. Like you know, actually understanding what might be a very boring process. And then translating that into your own prompts, your own evals, eventually your own data sets to tune your own models.

Like those are all things that are incredibly valuable. And then clearly there are examples you know, earlier we're saying like Character AI for instance, you know, took LLMs. Obviously built some of the first LLMs. Then took many of them and then fine-tuned them in a way so that they could bring down the cost of serving those models by 10x. And so you know, that itself is also a form of a cornered resource. The best cornered resource to have is your own model that can like do the specific work Yeah. better, right? And for a while people thought that was the only moat that you could have in the space. If you didn't have your own model, like you were totally hosed. Turns out that's not true. Turns out that's just one of the possible moats. Partly that is a threat people are worried about in in the big picture. The 10,000 foot scary thing is if the labs at some point decide to treat their models as a cornered resource and they restrict access. I guess the interesting thing right now is like it may well be true that the you know, platonic ideal perfect manifestation of an AI system will require a lot of both you know, maybe pre-training, post-training, RLHF, like just so many different things that you have to throw at it to get it to like ChatGPT level. But we're also so early in the revolution that you know, even if just context engineering gets you 80 or 90% of the way there, that's plenty.

That's actually all people need to do for like the first two years of their startup almost always. You know, Cursor didn't start out by doing you know, full parameter fine-tunes of GPT-5 which they probably have access to now. They started just by making something people want. You know, earlier we were saying like don't use these frameworks to count yourself out prematurely. And this is a very profound version of that. So the third power we're going to discuss is switching costs. That is the concept where you get a moat when your customers are kind of trapped because it becomes very expensive for them to find other solution even if the other solution might be like a little bit better. It's just very painful for them to switch financially or in terms of the operations, time, or effort because they just have so much of it in the current solution. And examples that are given in the book are um like databases like Oracle. When you have all of your system of record and all your data in Oracle, it becomes incredibly hard to migrate. Like database migration is something that people don't do. Other example given is a Salesforce.

And because once you have all your customer records in Salesforce, you build all these workflows, the UI, and it's just a lot to retrain a lot of your sales team to use like a new software. You need to like migrate all the data, and then at that point for the company to switch to a new CRM is probably going to take I don't know, lose like a whole year of productivity or something. Even if the new solution is a little bit better. I think how AI companies are building moat with this has to do with a version of what Gary mentioned with the forward deploy engineer. We give an example to this with Happy Robot or Salient where they start with specific workflows that are very customized per company.

And they work with large enterprises. And part of it is actually with the forward deploy engineer, they may have actually very long pilot periods, which might last like 6 months to a year. But if they succeed, these convert into seven-figure contracts. And the reason why these pilots are so long is because they're very much building custom software for the specific operations in these companies. And the examples for Happy Robot, they got customers like DHL, where they went deep into integrating into a lot of the workflows for how all their logistic operations are done, which is very custom to the DHL operation. Or the example for Salient, who's building AI voice agent for the financial industry, they integrate with banks, and a lot of the banks have very different work flows on how they do a lot of the loan consolidation, how do you do the debt recovery, how they do a lot of the fraud monitoring, and risk, and compliance. And it's all a little bit different because all these companies have built kind of internal tools. And the whole part of bringing an AI company that builds these workflows, they build custom workflows and that work with them. But as a result, the trade-off is you do have very long pilot cycles.

But the protocol is worth it because you end up with this big contract. And once you're in, you're kind of minted. And the big enterprise is not going to do another big off because it's going to be a huge waste of time for them to let's try the other whatever cool AI voice agent company. At that point is like we just want to get the benefits. So that's how these AI companies are winning. I think it's like at once a moat, and it's also in it's interesting in the age of AI that uh simultaneously you could have see how AI brings down the cost of switching by a lot. And that's you know, sort of another lever that a startup could use.

Like if you can write um use code gen to basically extract data out of old ossified systems or your competitors, then you you know, there are things that might have really relied on switching costs that you could potentially bring it down to zero. Yeah, there's actually two different flavors of switching costs, right? There's the the old school ones from the SaaS era. All the system of records like Salesforce, but also ATs like like Lever and Ashby were where the switching cost was the painfulness of migrating data from one system to another. And I agree with Gary, LLMs might significantly reduce the switching cost cuz the LLMs can figure out how to like morph the data from the old schema into the new schema. You can use browser like use browser automation on both sides to like solve issues where like people don't let you export the data. But then there's this new form of switching costs that I think is pretty native to the AI era like you're talking about to Tatiana, which is like these these lengthy onboarding processes that lead to like deep customizations of the logic of the agent, not just the data, that didn't really exist in the SaaS era.

Like I guess you'd like customize your like your Zendesk implementation a little bit, but like not that much.

Yeah. I mean, and then for AI companies on the consumer side, I mean, this is all very nascent, but like I think memory is already becoming a bit of a switching cost for me. Like it actually blew me away that Claude was so behind on memory. And then, you know, my relationship with ChatGPT, I feel like has evolved very significantly in the last year where I'm like, oh, I actually just generally it seems to know, you know, what I'm into and what I care about. So, you know, that switching cost I think over time will only become greater and greater. And so personalization for consumer is actually a huge piece of that. What about the counter positioning, the other moat on the book? The definition of counter positioning is doing something that is difficult for the incumbent that you were competing with to copy because it would cannibalize their business. I think there's a couple of ways that this plays out. In every category, there is a Darwinian competition between the existing SaaS incumbents building their own AI agents, and the new AI native companies building AI agents on top of the existing SaaS companies. So like for customer support, the existing SaaS incumbents like Zendesk and Intercom and Front are all building their own AI agents. But then we have like a new wave of companies that grew up in the last couple years that are building AI agents that interface with with the system. I think it's like I don't know, this could be a topic of a whole like like cone episode, which like who will win in in in each of these fights. I think it's really interesting. Unstoppable force meets immovable object. One way where this is playing out in the counter positioning is that all almost all these SaaS companies, their pricing model is they charge per seat, i.e. per employee.

And this is I think a very big Achilles heel that they have strategically, which is that if their AI agents do a good job and actually work, those companies will need fewer employees doing this work because they're like the work will be automated by AI agents. And in a and in a simplistic way, that will just actually reduce the more successful they are, the more they will reduce the revenue. My guess is like some of them will be able to navigate this. Like especially if they're still founder controlled, I think like Intercom for example, like the I think the founder controlled versions of these companies are smart enough to recognize that this is existential, and they may be able to cannibalize themselves. I think the ones that are not founder controlled, I don't have a lot of hope for. It's super hard to cannibalize your own revenue. The alternative as we're seeing is so much of the startups pricing models are around sort of like work delivered or tasks completed. I think it's it's exactly what you said, but it's also that then switches the product towards having to actually be able to complete the work. And then something I actually repeat at the last YC batch the end as closing advice is that I wish the founders in the batch could just somehow go spend a month at some of the late stage companies cuz the top thing we hear from the founders running those companies is how hard a time they're having sort of resetting the engineering culture in their org to actually embrace AI, to use the tools, to want to do like context engineering and prompt engineering, and and the the net result of these teams not actually being able to be AI native, for lack of a better term, is they just can't deliver the products that work. Right? And so like they both don't want to switch from per seat pricing because like that's what they're used to um uh and in a world of AI being able to do the work, there's going to be less seats to sell to. But they also just cannot deliver on products that can do the work. And so they they they wouldn't that pricing model is not going to make any sense for them either. Yeah, it's it's like the process engineering part. They're not good at the process engineering part for this new kind of engineering. I mean, something sort of emerging that's very interesting in a bunch of YC startups like Avoca for instance, they're doing customer support software kind of like ServiceTitan, but for HVAC. So literally like people who help you with heating and air conditioning. And you know, I think ServiceTitan has something like 1% wallet share, 1% of the gross transaction value of like a given HVAC company. Um which is very small, right?

I mean, people don't spend that much money on software because these are relatively low margin service businesses. But the wild thing that Avoca discovered is that, you know, they can come in as software, but then over time they're actually getting a bigger and bigger chunk of the wallet share because they can get the HVAC people to pay them uh actually for the customer support piece, which is not 1% of their spend, but 4 to 10% of their spend. So what you may well find is that uh this new breed of AI startup will actually have more growth uh and uh higher wallet share. So, you know, actually we may well be all undervaluing how powerful and how big the vertical SaaS AI companies will actually be because you're not like 1% of wallet share, you can get to 10.

That's what we talked in that episode where vertical AI SaaS agents will be 10 times at least 10 times bigger than SaaS because it's really to your point, Gary, tapping into a whole different part of the spend of the companies. It's not the wallet of software where you're kind of at this point I suppose it's a bit of a uh finite budget, but it's really new space where with things that were not done possible, and it was mostly workflows from from people. And I you know I know that people are like pretty sensitive about uh workforce displacement, but you know customer support for an HVAC services company is not a fun job and you can tell because all of these customer support jobs actually have like 50 80% annual attrition rates. Like they're just such torturous not fun jobs that uh the companies themselves and the call centers themselves spend almost all of their time trying to vet and bring in more people to work on these terrible jobs. And so when you have better software, what's sort of happening is that instead of like people aren't losing their jobs. These people are quitting their jobs anyway because it's a terrible job. And then if anything uh what Avoca has told me is that many of the people who were in those customer support uh you know sort of roles, uh now they're actually having more fun jobs because instead of like managing a whole set of people who don't want to be there, uh they're actually managing AI agents and then handling the interesting weird cases. The coolest part of it is like they actually can go in and sometimes alter the prompts and sometimes you know actually have an impact a direct impact on uh both the experience of the customer, but then also their own day-to-day. And that immediately is like a 10 times more interesting job like wrangling a bunch of AI agents and making uh the support process better and better over time. Like that's you know as knowledge work goes like way more interesting than follow this script and read what the computer says.

So Harj, you you had a really interesting point about uh second form of counter positioning. The space has moved so quickly that in every vertical um or many verticals, there's sort of early on emerged one company that's seen as the early winner in the space and often it's actually like the second movers. At least in the YC context, we have seen over and over again that like advantage to being the second mover in the space. Like Stripe came after uh Braintree and authorized.net bunch of things and was able to like actually win by just building a better product. DoorDash came after Grubhub, Postmates, various other delivery services and eventually went on to win. And so I think it's interesting to sort of just consider about if you're entering a vertical where it's already feels competitive or there are already there's already seem to be like a early winner in the space, how do you counter position against them? One thing I think is really interesting here is Legora versus Harvey. Legora's obviously uh both in the legal AI space. Harvey was the early winner. The counter positioning that I see from Legora is Harvey came in early and maybe got early sales um but focused a lot on fine tuning as sort of like their product differentiation when over time it's seen that that was actually probably not the right move. You wanted to actually focus in on the application layer and actually sort of building a better product and and Legora has focused on that. That's what their branding and positioning is and it seems to be

working really well for them as a second mover into the space. A company that I've worked more closely with Giga ML entered the customer service space and they're competing with Ciera and Decacorn, like really well-known customer support companies. And from having seen their sales motion, how they've been able to sign up some big customers, their I think their counter positioning is their product fundamentally just works better out of the box and as a result they can have a much faster sales and onboarding process. So it's like their counter positioning is if you want to sort of get your customer support working as quickly as possible, um you should go through like the Giga ML onboarding process versus like the Decacorn. And I think that's actually worked quite well for them. Yeah, Giga ML is an interesting example of how to your point about like labor displacement, it's clear that an AI agent can do this job not just as well as a human, but actually much better than a human. Like the DoorDashers that the Giga ML agents are talking to, a lot of them don't speak very good English. They speak all kinds of languages. You can't hire a customer support person who's fluent in 200 languages. Um But but LLMs are actually out of box.

>> of the box. Um and they're infinitely patient if like there's a bad connection or so that's pretty interesting. I think you have other example where to your point of a superhuman abilities is where the AI version of the product actually works. I think Harj, you you had the example of a Duolingo versus Speak. Duolingo's obviously the biggest language learning app I think um most consumers know. The emerging criticism of it I would say is that um what it's actually just sort of like a gaming app versus a language learning app that like the way that app works is orthogonal to learning a true language. And then you have Speak um which is a uses LLM like uses voice to actually like help you practice and actually learn the language um and that counter positioning is working really well for them, right? It's sort of Speak has got explosive growth and it's not trying to compete with Duolingo on their we've we've got like lots of gamification and points and sort of like a great game mechanic. It's competing on hey, we're actually just the place you should come if you want to learn the language by speaking it. I I think the counter positioning mode is very um so close and overlaps with the branding moat idea. I think in the book he talks about you know like brand is it's essentially a moat when you become so well known that even if you have an equivalent product, um consumers will still choose you um because of like the brand effects. I think the example he used is like Coca-Cola. In the AI context, I think it's probably harder to apply brand as a moat directly to startups cuz it just takes time to acquire brand. Um but you can certainly see its effects.

Like the thing that still stuns me is OpenAI ChatGPT has more consumers using it per day than Google's Gemini. I think anyone who understands the models and uses them um daily would say that Gemini Pro 2.5 and Gemini Flash 2.5 are like equivalent models. And Google also had all the users. Like basically everyone in the world is a user of Google. OpenAI had no users initially. Google was already one of the biggest consumer brands on the planet. It was almost certainly the biggest consumer brand on the internet uh and yet somebody else came along and built the brand as the consumer AI app and Google is like playing catch-up.

>> If someone had tried had told me in 2022 that that's how it would play out, I would have been fairly incredulous. It's also a perfect example of counter positioning again. I mean this is Google had a uh a business model that required it to continue to support ads and an organization that uh they shipped. And so you have the greatest cash cow in the history of man. So why would you disrupt it um even at the cost of setting back uh human access to knowledge by a few years. Even if that's like the core stated goal of Google itself to organize the

world's information. There's also an untold story of how uh the origin story of ChatGPT, how it came to be.

Which is really the original moat for startups with speed. It shipped very quickly in a matter of months with a very small team of a couple engineers. I mean it required uh you know Sam Altman and YC research and Greg Brockman to go uh hire Ilya Sutskever out of DeepMind cuz he was there and you know he all the people a lot of the people who went on to help create OpenAI, uh they came from DeepMind. Like it was already in the right place. It's just that that place didn't nurture exactly the thing that society really needed. For speed. So that there's that moat again, speed.

Number one. Do you want to talk about network economies, Dan? Yeah, on the book uh network economy is described as uh where the value of the product increases as more users or customer get and use the product and everyone derives more value as a effect of more people using it. And examples that were given in the book are uh Facebook where as you use it and your friends use it, it's more fun for me to use Facebook because all my friends are in there. As more users come in, then is the social network becomes more valuable.

And this was very much the era of where people talked about uh network effects that came to be. And the other example he gives is like Visa, the Visa network where the more merchants are using Visa, the more value the consumer gets because you can swipe the Visa card more places. Then that becomes the the moat because it's harder to then acquire and amass this number and large number of uh users or merchants in order to to win. So that becomes very defensible. In the current era for AI, the shape of uh network effects is different. It really comes into the shape of data. I think a lot of uh the data that a lot of AI companies get access to becomes the moat where the more data they get, the custom models they build become better and the better models it becomes a better product for users. And there's lots of examples of these. And um besides like the big foundation lab companies where they probably use some of the data. I don't know. I mean they probably use some of the data from the users. They probably do.

>> ChatGPT almost certainly like feeds a lot of that back because you have a certain reward function for each training run, right? >> of every chat from ChatGPT 1 2 3 4 5 now goes fed into GPT-6 and then so on and so forth. Helps create the the next model version. And there's uh even smaller versions of this. For example, Cursor. They have probably one of the best uh tab tab auto complete because one the the free version of Cursor, they actually say it when you sign up that they they will use the data and they use that to train it. And the more users they get, the >> it's like all the data. Like I think it's like quite literally like every mouse click and every keystroke that you that you admit when you're using cursor like is fed into a model which is like kind of crazy.

Which then the more developers use cursor, the better the product gets and then they compound a lot of the a lot of the wins with that. And the version where this applies to AI startup is when they go work with enterprises and large companies, they get access to private data. I mentioned earlier Sillian or Happy Robot when the employees of the companies where they become customers as they use their product, they have a lot of that private data that makes a lot of the workflows better. And the way they improve that, which is a second way of having modes with networks is really evals. We we talked a lot about evals being the key mode for AI startups evals is where you get a lot of the this workflow work or they didn't work and then take that back and iterate

and improve your contact engineering. And that is a flywheel that you can only achieve when you get more and more usage of your product.

Whether it be in a consumer or a or a AI vertical SaaS agent. So now the last mode in the book is scale economies. Jared, do you want to tell us about it? Scale economies or economies of scale, you've invested a lot of money to build something that's really big and as a result you have a economies of scale and you can offer the service cheaper than anybody else. So like the classic example would be like UPS or FedEx or the Amazon delivery network they built like massive like physical infrastructure and as a result they have like a lower cost per unit um compared to a smaller competitor. Um I think the way this has played out in the AI world, I don't think it's actually played out that much at the application layer. It's really played out at the model layer, right? Like training a state-of-the-art LLM is very capital intensive. Only a few companies can afford to do it. Once you've done it, you can afford to like let people do inference on that model very inexpensively. This is why the Deep Seek announcement was so um was so earth-shattering last year because it seemed like it might be a lot cheaper than people previously thought to train a frontier LLM which would greatly diminish the power of this like economies of scale mode that people thought the the AI labs had.

>> The key thing about Deep Seek was they figured out and made public this new unlock for models which is how to do RL. They still build on top of a one of the large foundation models, so it's still expensive. That RL part is cheaper, but you still need the very expensive big foundation model. So that's one of the things that the media got wrong. There's a separate question that people talk about which is like how will the foundation model companies be defensible against each other? And like this is certainly one way, right? It's just like it's it's very hard to be a new entrant into that game now because of this economies of scale. And we were we were thinking earlier about like how this had played out with startups.

And there's not that many examples, but I think a couple of good ones, well one one good one is a is a company of yours Exa. Harsh, do you want to explain what what Exa does? Yeah, Exa is essentially search for AI agents. Um it provides an API for anyone building AI applications that wants to search the web. And the way I I think this is playing out for Exa is in order to provide that service, they need to crawl the web. Not the whole web like Google does, but a big chunk of it. And that's very expensive to do. It requires like a large fixed capital investment. And but then once once you crawl a big chunk of the web, you can reuse that same crawl for for many different customers. I think what's interesting about Exa, the the parallel to the model companies is that they they had invested in that like sort of before agents had really taken off. Like they were fairly early to this. I think their work on this actually even pre-chat GPT launching. So they made the investment early on to get better same way that the lab companies took a bet on like transformers and and scaling laws. Yeah. And there are two companies in just the most recent batch, uh Channel 3 and Orange Slice, that are both doing Exa.ai like plays where they crawl a big chunk of the web, have a big like static crawl on their own servers and then have agents that run on top of those of that crawl. So I think we're going to see more and more of this especially as the web agents work better. You need to mainly focus on uh the first mode that isn't even in the book, which is speed. Like you know, if you're really breaking your brain about like oh well, are we going to be a cornered resource or not? You're just thinking about it in the wrong way. Like you should not start there. You should start with do I have a specific person who has some sort of pain point and it's pretty painful. It's not like a oh, it'd be nice if I could do this. It's a oh, I am not going to get promoted this year.

Maybe I will get fired. Like this is so painful that I don't want to go to work today. Like that's sort of the type of pain that you're looking for. And if you can write software or build things that actually alleviate that pain. Like existential pain. Like the business is going to go out of business or oh my god, we could totally take over everything next year. Like that's sort of the feeling that you want in your customer. Uh if you can find things like that go go zero you know, go find that and go zero to one on that first. With that, see you guys next time.

[Music]