

[JAVI BELTRAN AND JORDAN GREENBERG, "TELEBOT"]

JORDAN GREENBERG: Welcome to season seven of the *Prodcast*, Google's podcast about site reliability, engineering, and production software. This season, we are exploring what it takes to ship software and what that even means to different folks. Join us as we discuss how to ship it as SREs. And of course, the most important part is the friends we made along the way. Happy listening, and may all your incidents be novel.

Welcome to the *Prodcast*, the podcast and site reliability engineering and production software. My name is Jordan, and today I have my fellow co-host, Matt.

MATT SIEGLER: Good to see you.

JORDAN GREENBERG: Yes, good to see you as well. And a very, very cool guest who, still in 2026, runs a blog and does some other stuff. If you could give us an intro for yourself.

VERY PENNARUN: Hi. I'm Avery. If you've seen my blog, it's under @apenwarr, which has long time been my online handle. I'm the CEO and co-founder of Tailscale. I'm a pretty technical person. I haven't done a lot of coding in the last few years because I got sucked into management and stuff, although in the advent of all this AI stuff, I've been able to do or direct a bunch of coding in between meetings, which has been pretty fun.

JORDAN GREENBERG: Ooh. That's really cool that you said that, and we're definitely going to get back there eventually. But for now--

VERY PENNARUN: Yeah, and you said that you're going to do it flawlessly this time. But isn't the right SRE approach to have an error budget?

JORDAN GREENBERG: Hey, you know what? I've already exhausted the error budget, and that's why I mentioned it.

[LAUGHTER]

In previous episodes, we've exhausted the error budget. You said Tailscale. What is Tailscale?

VERY PENNARUN: Well, Tailscale-- the name is a bit of a joke. It's the opposite of internet scale. And so what we try to do is to solve the smaller problems that everybody else is not obsessed with while they're obsessing over giant problems. And that was where we started. As the company has grown, we've started solving larger and larger problems, but our core is still when they're like, hey, can you do the stuff that is not the things that require scaling to thousands and thousands of machines to serve billions of users? Like, how easy can it be to launch an internal dashboard for your team? That was the initial concept.

And we started exploring that, and we quickly discovered that networking and security are the top two reasons that it's hard to make a dashboard for your team. Networking is like, hey, how do I get my team to see the dashboard? And security is, how do I get people who are not my team to not see the dashboard?

And that became, now, a seven and a half-year-long project. My joke to the *Hacker News* people, who, of course, posted as soon as the announcement came out, I could do this in a weekend. And it's like, well, yes, we also did it, and we've been debugging it for the subsequent seven and a half years. It's been a long journey, but it's pretty reliable now.

JORDAN GREENBERG: And there's this component of reliability that is simplicity. So that makes sense to want to make something that is so simple because then it means it actually is more reliable, in addition to the seven years of debugging. I heard that also helps with reliability, but I don't know.

VERY PENNARUN: Yeah. Well, I mean, the simplicity is hard. It's something that comes up a lot. And it's actually pretty easy to make a complicated networking product where all of your reliability problems are the user's fault, but that doesn't actually help the user, who-- the majority of whom don't know how to debug their problems. So if you make those problems your fault,

that kind of lowers your perceived reliability at first, but then you at least have the ability and the agency to be able to fix those problems on behalf of everybody.

And that's what we've been doing for seven and a half years. We took more responsibility for just making sure the connection always happens. And if it doesn't happen, we're going to figure out why, and we're going to fix those problems.

JORDAN GREENBERG: OK. So how has that understanding of reliability evolved, and how did you wrap it into Tailscale? I know you mentioned users. Obviously, there are enterprise users, but there's also users at home who have home labs. I won't self-report. But how has that reliability concept been rolled into the product?

AVERY PENNARUN: So Tailscale, from the very beginning-- our strategy has been, we want to make it so easy to use and so reliable for individual people playing with it at home that we will worry later about going upmarket.

Now, it's been seven and a half years. Later came and went. So we have gone upmarket since then. There's multiple Fortune 50 companies, many Fortune 500 companies, apparently now 40,000 other paying companies that are using Tailscale and millions and millions of people using Tailscale for free.

So that's a lot of people. But it all starts with, hey, we're-- either less technical people, or people with not that much free time, or the situations that are not that complicated, and they want to do something. The thing they wanted to do is not networking. Can you get that out of the way for them?

And it's really served us well that we solved that problem at the very bottom of the market first because then when you come-- when you show up in an enterprise and say, look, I realize all the other networking products that you're considering are pretty neat, and they have lots of features in their feature checklist, but they're going to take you weeks to evaluate because it's really hard to set them up. Why don't you, in front of all that, evaluate Tailscale for five minutes, right? And then tell me what you think, and then we'll talk about the feature checklist.

And a lot of people have already tried it at home. They have friends who've tried it at home. That's why we got in the door in the first place. So it's worked really well for us.

But enterprises have a pretty low bar for product quality and simplicity, and sometimes reliability, because they have whole teams of people who take responsibility for that on behalf of the vendor. And that's OK. You have enough money. You can do it. And it gives you more flexibility to buy something that perfectly meets your need. But you end up with thousands and thousands of companies just solving the same problems over and over because the upstream vendor's product does not solve the problem for you.

And it's just-- to me, it just seems really wasteful that everybody's just doing the same repetitive stuff over and over and over. I guess the SRE concept of toil, right? That's how it feels to me. If I make thousands of customers do the same thing, that's toil. Why don't we spend a certain budget and optimize that? And then all those people don't have to do it, and they should give us-- it's a business. They should give us back a fraction of that time they saved in terms of the subscription fee.

MATT SIEGLER: I like how you phrased that. So you've effectively-- I don't want to use the word normalized, but averaged out, amortized out a lot of people's idiosyncratic problems into a standardized problem. They subscribed to it. And then you absorb all that behavior? What does that look like from, internally, all their idiosyncratic difficulties-- how do you absorb all the complexities of that? And what does it look like internally in terms of the distribution of their complexities?

Because although, from their perspective, they're each their own snowflake, and they each have their own particular problem, and then you absorb it. You create a standardized practice. But of

course, internally it's like, oh, well, they're out on some weird internet part of the island. And actually, it creates some strange network behavior. And they don't want to know that they have a peculiar implementation problem you have to solve. They just want to know, yep, I signed up. It works great. And you're like, oh, we have to support that customer. It needs to just look like it works. How does that look like from a business practice, that customers just sign up and it just works?

AVERY PENNARUN: Yeah. Well, I mean, it looks like thousands and thousands and thousands of bugs in our bug tracking system, first of all. But all of the stuff that, for your own network, just would not be plausible, like, the approaches that just wouldn't make sense-- it's like I'm writing a little-- I want to connect a little home VPN so I can access my file server from my phone when I'm not at home.

And it's like, look, are you going to write a giant, end-to-end integration test that includes hundreds of different router vendors and different operating system versions and old and new versions of your VPN software and a whole bunch of super weird conditions, and I'll break the network connection, and I'll shift over between Wi-Fi and cellular? Like, all of that end-to-end testing-- you won't do that for your one VPN, just your one deployment.

You either count on the fact that somebody's done that for you, which, if you're using an open source VPN, chances are, they haven't. Or you're going to be like, OK, well, I hope my situation is not too weird, which, statistically, most people's situation is not that weird, and you'll be OK-- or not that weird most of the time.

But then you get this long tail of just weird issues that you're never going to be able to address. And that's people's experiences connecting to their home lab at home, traditionally. And that's really-- that actually is weird. We're doing this just because the network has been broken. So what we do at Tailscale is, look, we've got all these millions of users. We've got lots of people paying us money to work, to use Tailscale in their companies in really complicated situations. We're just going to make sure that every version of Tailscale is tested against every situation we've ever seen historically.

And we keep on adding more and more regression tests, and we optimize the regression tests, and we find the patterns, and we say, oh, well, here's a situation that we've never seen, but it could happen. Therefore, it's probably happening to somebody and they just haven't reported it as a bug yet. So let's also fix that, and let's make sure that every future version of Tailscale also doesn't have that problem. And so over time, you just have this ratcheting level of reliability. But it's interesting because there are clones of Tailscale that exist in the open-source world. Tailscale itself is open source, but sometimes people don't want to use the Tailscale one, and they want to write their own from scratch. We have non-open source competitors.

They can look, really, very similar on the wrapper, right? I mean, Tailscale looked very similar in 2019 when we first released it. It hasn't changed that much, visually. But the part you can't see is this super long tail of ridiculous edge cases that have been resolved one by one by one by one because millions of people have been finding problems and then we've been fixing problems.

JORDAN GREENBERG: You did mention exactly what my own network stack is for my home lab. And as we call it, the SRE at home-- it seems pretty straightforward, and obviously been proven, that people can use it for being able to network into their computers wherever they're at, if they're on the airplane. Been there, done that. How does this look different from SRE at home versus SRE at work? What are the differences in resources constrained that a person who is an enterprise would see versus someone who is at home would see?

AVERY PENNARUN: So Tailscale has been one of the most interesting products I've ever worked on in the sense that it's almost exactly the same product, all the way down to one user

and all the way up to 100,000 users.

Now, there are differences, and one of our earliest investors at Heavybit explained this to me early on. He said, Avery, what people will pay for is more control and more auditability. Like, if you need those two things, you need them for your enterprise and you know that you can't get it unless you pay for it. All the other stuff just holds you back if you try to charge too much money for those things.

And so the core function of Tailscale is exactly the same on the free plan as it is for a customer on the 100,000-user plan. The extra stuff that we've added for enterprises is adjunct. So there's network flow logs, for example. So you can log every single connection in the entire system, even though it's a peer-to-peer mesh network. You can flow it all into your CM system and then have something analyze it and look for intrusions and things like that.

We've got other kinds of audit logs. We've got device posture stuff flowing in from all the different popular EDR vendors that you might want to use. We've got a bunch of integrations with other kind of enterprise software.

We work with all the different kinds of login systems, and then you've got user group sync. If you're only one person, you don't need to sync the groups from your identity provider, because there are no groups because you're just one person. But as soon as you get bigger, it's like, hey, this is a useful feature that you're going to need to configure.

So all of that stuff looks a little bit different, but the core of Tailscale looks pretty much the same. And SREs at home, from a connectivity point of view, or the programs they use, or the weird setups they have-- home lab users just famously make everything complicated on purpose because they want to learn about this stuff. That's one of the reasons they built the home lab. So you'd be surprised.

JORDAN GREENBERG: Yeah, it's part of the process.

AVERY PENNARUN: You might say like, oh, well, you're going to have-- you wouldn't have a Kubernetes cluster at home. I'm like, yes, you would.

JORDAN GREENBERG: Yeah.

AVERY PENNARUN: Yeah, you absolutely would. And so that stuff is exactly the same between the free plan and the enterprise.

MATT SIEGLER: You have an interesting business growth complexity problem you're describing. Many businesses are constrained by the demand equation, which is like, I just have a lot more people, and they're in more places, and they want to do more. There's more QPFs or more gigabits or whatever. But you're describing more scenarios, more use cases, more unseen things. As time goes by, you have more unknowns. That's an unusual growth problem.

Describe what it's like to support a business where, as it grows, instead of just adding more SSDs and more RAM, you actually have to add more cases and more complexity handling. And although you said you have more bugs and you have more explanations that hopefully solve other people's problems, what is it like to support a business where you have more unknowns as you grow? That's a different kind of business that you have to support, and a reliability with a structure that isn't just scaling in a typical way.

AVERY PENNARUN: I think that's not exactly true. I think basically, every business-- if you support individual consumer users, all the way up to enterprises, you always have a situation where the enterprises have a bunch of requirements that the consumer users didn't have, that you have to check off those boxes before they can buy it. That's the normal state.

Especially in networking, you'll end up with-- if you're a Layer 7 level network instead of a Layer 3, where Tailscale sits, every single app you might want to plug into your thing might require a special adapter or something like that to be able to work with your thing.

MATT SIEGLER: Fair. Fair.

AVERY PENNARUN: And enterprises probably have completely different apps than the people at home. And so now you have a team building enterprise adapters, and if you're lucky, you have a team building anything for consumers, but there's no overlap between them. And then you sometimes get products where you'll see product name, and then, product name enterprise. And some point in the past they forked the product. And they're actually not the product-- the same product anymore. It just says the same name, but there's actually two different things.

And the really neat thing about Tailscale is that it's just Layer 3. It's basically the internet layer. And I have a whole thing where I talk about this sometimes, where the internet Layer 3 is actually not doing the job of OSI Layer 3 anymore. It actually does the job of OSI Layer 2, if you think about it.

If I move around on the network, my address changes. Nobody can find me anymore. I need a different Layer 3 address for every network interface on the physical link that I might be plugged into. And I can no longer reach every computer on my network, depending on my physical location. That's the definition of Layer 2 networking.

So Tailscale-- what I try to tell people is, it's actually the Layer 3 that we were missing and we haven't had since like 1980-something, because you get a unique address. The unique address is the same no matter where you physically are. Anybody can find you who's supposed to be able to find you. It doesn't depend on your interface. And you bypass these firewalls and NATs, and so on, and it gives you the security layer that you wanted and discoverability. So DNS actually works, and so on.

So anyway, the reason I talk about that is just like, look, Layer 3, the internet layer-- everyone basically has that same fundamental need to be able to move packets between devices with an address, and the address doesn't change when your device moves around. And so it's very interesting. That fundamental need is no different from the very smallest deployment to the very largest deployment of Tailscale.

The part that has been-- a lot of enterprise networking companies, when they're not operating at that layer and things diverge so much-- they basically say, I'm not even going to bother with the consumer level because it's too different. I'm just going to only chase where the money is, at the enterprise level. And that's how you solve the problem of the difference. You only do one of them.

That's not how Tailscale approaches it, because we've noticed that the network is just the same for everybody. And so what actually happens in our case is we get more weirder problems from the users on the free plan than we see from the users on corporate plans.

But the neat thing is all these users on the free plan-- like, Tailscale has this telemetry system where we feed back all the information about connections that worked or didn't work. And it's not your private information. It doesn't feed information about your web browsing habits or whatever, because Tailscale's only about your private network. And so there's nothing to-- no information to steal about whether you connected to a server named XYZ on your internal network. It's not interesting.

The only interesting thing is, did the packets get through? What did they look like when they got through? What things worked? What things didn't work? What kind of reliability do we have, and so on?

So we stream all that stuff back for all of the users of Tailscale. And then the trade is we use all that information to find anomalies, detect bugs before you notice them, and fix the bugs. And then our corporate users benefit from those fixes. Even though there's fewer corporate users with fewer edge cases every day, when they finally-- one employee at a company finally has a weird situation because they're traveling or they're in an airport or whatever, like, a thousand

other consumers who are on the free plan have had that problem before, and we've already fixed it. No customer ever experienced it-- no paying customer.

MATT SIEGLER: This thing you're mentioning about anomaly detection from your more idiosyncratic use cases and then propagating them through the more general case-- this is a spectacular use case. How early on in Tailscale's evolution did you come up with that model? I mean, I'm now enlightened to what you're-- inherently in your product. How early on in the evolution did this become aware to you, that was really going to be, essentially, a very valuable part? This is like beta testing in your free-- I don't know how to think of this, but this is fantastic, and I can see why that works. And it's how you scale your product to all its different kind of edge cases. How early on in your business did you discover that this was going to be a really good addition?

AVERY PENNARUN: Well, we cheated a little bit, because first of all, we named it Tailscale, so we were aiming at small teams. We were a small team when we started, right? Three people, and grew pretty slowly, in terms of headcount, for the first few years.

So this was a product that, also, we built for ourselves. It's like, here's a small team. We want to make a dashboard. We're going to use Tailscale to make our own internal dashboard, so we're going to learn all this stuff. Turns out a lot of other people on small teams and individuals have the same problems, wanted to solve the same problems. So that was really the thing we did first. And then the bet we made is that maybe someday, we're going to be able to go upmarket and charge a lot of money for a similar product in enterprise.

And that bet has turned out to work out quite well for us, because it did turn out, unexpectedly, that-- I didn't want to bet that everything would be pretty much the same in enterprise. That seemed like a stretch. I'd never seen it before in my life, so it's as much of a surprise to me as to anybody else that the problems look so similar in enterprise as they do in the smaller businesses and individuals. That's just basically good luck.

But yeah, we did it backwards. If we had started with enterprise, we might've never discovered the bottom part because we would've assumed that the situation is so different that it's not even worth chasing it. But because we started with small, because we wanted to start with small, and then it's like easily justifiable. It's like, oh, let's do a little-- dip our toes in the enterprise world and see how much money we can make. It's like, ooh, a lot of money--

[LAUGHTER]

--with the product we have right now, right?

JORDAN GREENBERG: Yeah, I think when you're aiming for small business or even home users, the people who are on the other side of the computer in that equation are inherently motivated to figure it out because you have more time versus having more budget in that way. So I think that it's a perfect test bed for all of this stuff.

And what I'd like to get into now, quickly skipping over the fact that we're going to need to publish a separate episode on the new OSI model according to Avery, is how do demand cycles impact the reliability of the service? Like, what happens on Black Friday, or has there been a historical day or event that leads to unusually heavy traffic for you?

AVERY PENNARUN: So Tailscale is a really interesting product. The vast majority of Tailscale traffic is not moved by us. The magic of the mesh network is, almost all the data is directly peer-to-peer between your own machines. Your machines have their own private keys. They put the public key in a little database entry in our control server in the cloud. And then our control server sends it down, the public key, to all the other nodes in your network that should have it. And that's basically our job, exchanging some public keys for you, to oversimplify greatly, but that's essentially what Tailscale does. And as a result, that-- basically, the little table of public keys in the cloud can scale a heck of a lot without falling over.

Now, there are some exceptions. Not every single pair of devices can achieve a peer-to-peer connection, and we're constantly fighting with those. We have a network of servers called DERP, which is named that way because that's how you feel when peer-to-peer connections don't work out. It's like, ah, derp, and then you fall back to [INAUDIBLE]. If lots and lots of people sign up very quickly, our DERP network could get overflowed, or overloaded. But it's basically stateless, our configuration list. All the DERP servers don't need to know who you are. They never can decrypt your traffic because it's end-to-end encrypted, and they're just a connection along the way, just like any router at Layer 3 on the internet is just a connection on the way. And so we can spin up new DERP servers pretty fast and take on any of that load that happens in case of a spike.

The control server, historically-- we made the intentional design decision early on to make it really, really fast and low-latency at the expense of using standard distributed database design patterns. And you can find some blog posts. We've been known to publish them on April 1. And it's basically-- a lot of people are like, here's a joke. The joke's on you. We like to publish April 1 blog posts. It's like, here's a joke. The joke was on us, and we made some mistakes in our data store design and the scalability of our control server, and people have found those interesting. There's another one that's due--

JORDAN GREENBERG: I love that.

AVERY PENNARUN: The current joke is that we should've published it on Halloween because of how bad some of the problems got. But anyway, we have figured out how to scale. If you've got lots and lots of users signing up, for example, or lots and lots of small tail nets, we have the ability to scale the control service now, pretty much horizontally, as much as we want.

The next frontier is like, how big can you make a single tail net? And it's pretty big, but it's not as big as it turns out some people would like. And then once you have a single tail net, how fast can you add and remove nodes from that tail net? Because in the original design of Tailscale, we're like, well, look, nodes don't come and go that often. Even if you've got 100,000 employees, how many new employees do you hire per day? Not very many. So what's the big deal?

And it turns out our imaginations were not vibrant enough, because it turns out people now do like CI/CD. They do swarms of agents. They do little apps that persist for five seconds and then disappear.

And so people want to do a network of, like, a million nodes, and then they want to have a thousand or 5,000 per minute coming and going. And it's like, wow. When you think about how many notifications it takes to tell all these nodes about the change to the shape of the network, that's a very large number. You start getting into n -squared and n -cubed problems, and it's like, woo-hoo!

And there's shortcuts we can take to rewire that to be closer to order n . And we've done some of those shortcuts. Like, we can get surprisingly big with surprisingly high churn rates today. But we want to make it so it's essentially-- maybe unlimited is too optimistic, but it's much less limited. And your option right now is like, OK, if you really have a lot of nodes, you should either have-- if you're going to churn a lot, you should have smaller tail nets. If you're going to churn less, you can have a larger tail net. And I'd like to remove that restriction.

MATT SIEGLER: You mentioned your blog, I think. By the way, you've been blogging since before it was cool when it was cool the first time around. What's it been like?

AVERY PENNARUN: I published some of my high school English class essays when I first made my blog. So if you rewind all the way back to the beginning, it's, like, 1990-something.

MATT SIEGLER: Yeah, exactly.

AVERY PENNARUN: I just retro-published them just to make it look like I've been doing it

longer.

JORDAN GREENBERG: It looks like he rode two different cycles of blogging becoming in fashion and then not, and then returning. So we're in the second one now.

AVERY PENNARUN: The amazing power of being lazy is you just-- all the fads average out.

JORDAN GREENBERG: Mm, true.

MATT SIEGLER: Yeah. It's rewarded. Inertia's great. So tell us a bit about your experience lately, especially now that people are rediscovering this fabulous thing. What kind of feedback are you getting? What has been really sticky, topics that seem to have impact? What is really important to you? Why are you talking about it?

AVERY PENNARUN: At this point, my persona has been divided into different slices. So we have a special flavor of Avery that goes on LinkedIn, for example, and has to post in LinkedIn style, but it's adapted from little micro blog posts that I write on Bluesky. So like, here's my joke about something, and then we'll take the same joke and make it much longer, but the first paragraph has to be really short because of LinkedIn's way of displaying stuff before the Read More button.

All of that stuff is businessy. There's Tailscale.com's blog that has my posts sometimes that are specifically about, oh, we launched a thing. Here's the business strategy, blah, blah, blah. And I try to write in a little more businessy voice for that.

And then for things that I feel like just, OK, this doesn't fit the Tailscale voice, let's put it on the Avery blog. And a lot of people have not actually connected the dots between those two things. And so I can say whatever I want, quote unquote, asterisk, on my own blog. And so my spicier opinions end up there. So when I'm talking about the philosophy of networking or, here's how management should work, or, here's my thoughts on cool stuff you can do with AI. I like to put it over there.

JORDAN GREENBERG: So speaking of the opinions that you're getting and the ones that you're writing, what do you think about the information that's made available to you as a C-level exec, now that you are one? You've done some writing about how information goes through and how executives may see products. Tell me about what you believe now that you are the executive and not the engineer.

AVERY PENNARUN: Let's see. I do have one blog post called "What Do Executives Do Anyway?" And it was from a few years ago, and it was inspired by some of my time spent at a certain giant company that you are probably familiar with. And after I left-- while I was there, I looked into the kind of stuff that executives are doing and that individuals are doing, and how the relationships work, and what makes teams successful and so on. And I wrote an internal post there. It was very fun, though. Anyway, so a lot of people had comments. And then I left, and some people sent me a reference to this book. And I read the book, and I summarized it into that blog post.

But the thing that I learned after all this exposure to so many things, and also being the CEO of a reasonably successful company now and talking to lots of investors-- in short, nobody knows anything. All conspiracy theories are disappointing, because actually, you wish that some of the conspiracy theories would turn out to be true because it would make me feel better about the state of humanity. It's like, well, at least somebody knows what's going on. And it's like, maybe I don't, but I would feel better if somebody up there somewhere has a grand plan and the plan is executing, right?

But nobody has-- nobody has the plan. You can have a big strategy, and those are important. Like, executives do a job. That's why I wrote the essay about it. But it's so hard to coordinate a large group of people. And most of the information that's really interesting on a day-to-day basis is not something any one person could possibly fit in their head. And so if you can't fit it in your

head, you can't act on it.

And so the job of an executive is just so-- it's sort of disappointing, right? [LAUGHS] You have to set the general theme. You have to set the values that you're going to enforce. And the values turn into the strategy. You don't even get to set the strategy, because people will just go do stuff in alignment with the company's values or the group's values.

And sometimes, those values aren't even the ones you wish they had. If you aren't really careful about enforcing the values, the culture just gets created, and then it spreads, because culture spreads naturally inside a company. And you might have a company that doesn't even share your values. And then the set of values they have is what strategy they follow.

And so what are you doing? It's like, well, you just lost control of everything. So the only thing you can do is have a clear set of values and then try to build a company around that set of values and then trust everybody to be able to work together to achieve those things. And that's the strategy.

MATT SIEGLER: With that in mind, tell me how you're wrangling the use of AI and its related technologies, either technically or in how people communicate or in people making decisions, especially with what you just said.

JORDAN GREENBERG: Or what not to do.

AVERY PENNARUN: [LAUGHS] So I guess to add to what I just said, I think information is extremely valuable, and it always has been, and the way you handle it. In the world, up to AI's arrival, we had got to the point where, instead of the old days, where information was hard to get, information had become way too easy to get. There was way too much information. And if you just gave everybody access to all the information, they would be just completely confused all day long. And that's from the executive level to the individual contributor, all the way up and down. Nobody can act on that much information.

And so the trick is to somehow get all the information and then filter and organize it and feed it up into a system that then makes it make sense, and then distribute the information that has now been condensed into some story that makes sense or some collection of things that makes sense and then feed that back down so the people at the bottom and the middle of the tree can actually make decisions with the raw information that they personally have, plus this cleaned-up, organized information that comes top down. And then that's kind of optimal. And that's an important job of what you could call the information spine of a company.

It is very hard to do that. I wouldn't say, at Tailscale, that we're super successful at that yet. We're pretty small in the relative scheme of things. I think we're about 300 employees now. So the problem isn't out of control, which is why it's OK that we don't have the perfect information spine. But it would be better if we had a better one, and everyone suffers every single day from not having a better one.

Bigger companies either figure it out, or they end up doing a bunch of stuff that appears really dumb to people on the inside and the outside. They can still make lots of money if they've got an underlying successful business.

But information flow is the most interesting thing. So in this AI world, I am very excited about the fact that you can hook an AI up to your Slack and your Notion and your Google Docs and a bunch-- and your GitHub and a bunch of other stuff and say, hey, I don't understand this thing. Can you go scout it out for me and at least give me a summary, or at least tell me who I can ask?

MATT SIEGLER: Routing.

AVERY PENNARUN: And it does.

MATT SIEGLER: Routing is good, yeah.

AVERY PENNARUN: And so for me, as a CEO who's a technical person, I'm in this situation

where if I send anybody in the company a message, they panic and will reply to me right away, which is, I guess, nice if I need the answer to a question. But it's not as nice when I didn't actually intend for them to panic and interrupt what they were doing, and I just wanted the answer eventually. But now I've interrupted someone in the middle of the day and made them panic. And so I have to choose whether or not to bother people with my dumb question. And it's the same exact problem that someone who's a junior engineer who started last week has. It's like, hey, I don't understand this thing. Who can I ask? And they don't want to go bother a senior engineer who's obviously busy. And it's like, look, my work is not as important as your work. I'm embarrassed to ask this dumb question. I just can't find the right document. And AI fixes that specific thing. That is really wonderful. Both me and a new junior engineer can ask the same question of the AI, and it might be slightly wrong, but at least it's going to find me some stuff, and it gives me a place to start scouting. And often, it's enough information that with that I, can figure out the rest by reading the right documents, looking in the source code, or whatever. Or if I still can't figure it out, at least now I know which person to go bother, so I'm not bothering an entire team full of people who then runs around trying to find my answer. And that is a great improvement to the spine of information. It's not done, but it's just a night and day difference.

MATT SIEGLER: I could see an increasing general-purpose psychological safety in an organization. People feel safe to ask questions of an ombudsman who doesn't judge you for asking the question and might be wrong, or might tell you, oh, that's an important question. You should ask an important person. It's like, hmm. So I can validate that. That's very good insight. Thank you for answering it.

AVERY PENNARUN: Yeah, I've lived my whole life under the theory that there's no such thing as a dumb question. And so I want to be the person who in the meeting is willing to ask the dumb questions. But at some point, it's not about whether the question is dumb. It's just pure volume. There's just so many questions. And in a world where, now, everybody's in Slack, this is like, wow, that Slack thread-- Slack is just filled with messages all day. And nobody's doing anything wrong, but there's just too much, right? And this is the problem of our time is there's too much information.

And these AIs-- this AI processing can-- it's not about dumb questions. It's about, hey, I can filter out the easy questions or the things that an AI can answer, and that noise no longer has to go into the bigger channel. And then the stuff that ends up in the channel is just better. Like, I might still have a question after I do all that, but my question is better. And now everybody could benefit from reading the answer to that question because it looks like the answer to that question has not been written down anywhere yet. And so now you're building on the information supply more effectively.

JORDAN GREENBERG: So this is an awesome way to use AI in, like, a people-first direction. You're effectively working on improving your communication and the info you have to present that information better to everyone, and again, as Matt said, promote the psychological safety behind it. This sounds a really core value to you. Has that propagated within Tailscale, and what other core values have propagated throughout it that are important to you or important to your team, anything like that?

AVERY PENNARUN: I mean, Tailscale-- again, it always comes back to the name. It's like, how do you make small teams and individuals more powerful or more productive, more able to do cool things? Like, there's lots of companies and products out there that's like, oh, I have an entire team. I'm going to make them able to do stuff that is really hard. And it's like, look, I'm an individual. I just want to do some stuff, and I want to run into fewer obstacles along the way. And so Tailscale the product-- it's like, hey, I want to launch a dashboard versus our approach to how

do we make information available-- it's the same exact philosophy.

And so one of the things, our newest product, Aperture, which we just launched in January-- and it's an AI gateway, but it also has other stuff around it. And the underlying purpose that I think motivates me and the whole Aperture team is making it so that people who couldn't do stuff at all before can suddenly do that stuff.

And then again, the same parallel of Tailscale is eventually, you can move upmarket to the people who could do stuff before and helping them do it better. But why don't we go rewind all the way back to, hey, there was just no good way to do this, period?

And so for example, I have an executive assistant nowadays, and at some meeting, she was running things. And she's non-technical, and she-- for an icebreaker activity, she needed to be able to flip a coin. And it's in front of a big room of people, so an actual coin wasn't necessarily going to work. So she downloaded a coin flipping app for her phone.

And every five coin flips popped up an ad that we all had to watch for 30 seconds. And someone said, hey-- after the meeting, someone's like, hey, let me show you our product. I think we can do this. We're going to help you ~~use Claude Code~~ to produce a coin flipping app and deploy it onto your tail net so only our coworkers can use it, including yourself.

And so it took a little while, but as we were doing it, we took notes of all the bumps we ran into along the way. And the goal of Aperture-- and we're now very close to this-- is, hey, anybody like that, of which, basically, there are 8 billion-plus in the world-- you should be able to install Aperture and then make an app-- and it ~~doesn't have to be Claude Code~~ because there's now a lot of really cheap LLMs out there that could absolutely produce a coin flipping app for you on demand. And you need to be able to host it, and you need to be able to put that somewhere that you and your friends or your coworkers can see that's going to be secure.

And that can be anywhere. It doesn't have to be the expensive cloud, because Tailscale can connect anywhere to anywhere. It can be sitting on your laptop. And your friends or coworkers can flip a coin by accessing the web server that's running on your laptop in a container or a VM or something.

And I want to make it so five minutes after you install Aperture, which includes Tailscale in it, anybody can say, I want a coin flipping app, and they'll have a coin flipping app, and it'll be on their computer. And it'll still-- what's it cost Tailscale? It costs us rows in the database, not, we're going to host it for you, not, we're going to put it in our Tailscale cloud, not, we're going to provide the storage, not, we're going to provide the compute.

And the LLM's going to be the cheapest thing you can find anywhere, and maybe even on a demo plan from something. We'll try to make that as smooth as possible. And that is my fundamental value, Tailscale. Like, let's make it so there's 100 times more people in the world who can write software and have a painless experience. And of course, Aperture can also-- when you get into a company, you can tie it into your Slack, your Notion, your Google Docs, et cetera, and do all that stuff I was talking about, right?

JORDAN GREENBERG: Yeah, that's really smart. I think that, especially for your user base, you've got small businesses, but also people who are writing things at home. And maybe they're like, oh, I want to host my virtual tabletop game, or maybe I want to tinker with this app and see if I can make something. That seems really good.

Let's say I have that coin flipping app, and my friends love it, and I'm an engineer and want to make this a real product for people I don't yet know. What's the thing to know there, and what are the scary things people won't tell me about? My network, security behind it? What do I have to think about?

VERY PENNARUN: So let's say you just made that coin flipping app. I think one of the most important things about the way I want to approach making this app versus-- you can find vibe

coding websites on the internet. And their business model requires them to lock you in, host the app for you on their infrastructure. And so you basically can't leave.

And so if you want to post it on the public internet, that's great. But all of a sudden, you're like, oh, paywall. And it's like, well, look, we're paying good money to host your app for you. You should pay us money so you can share it with your friends.

If you do it with Tailscale, then you have, on your own laptop, the source code and the ability to run this application. And you can give the source code for this app to your friend if you want. And your friend can say, I'm going to commercialize this, if you want to let them do that, and they can go ahead. And they can run it on any platform they want. You can publish it on-- you can pick a cloud provider. You can run your own cloud provider off a MacBook mini under your desk with Tailscale funnel to publish it on the public internet if you want. You can do whatever you want.

If you want it to remain a private app among you and your friends, you can use Tailscale sharing to let people access those nodes. Or maybe you want to abandon Tailscale because the app you made doesn't require Tailscale at all. It's just an app that you can put anywhere and do anything you want.

So if you're an engineer who understands how to run production apps, that's the line where, hey, Tailscale-- we never said we were going to help you host production apps on the public internet. That's not what we do. That's literally the thing we said we were not going to do because there's enough people who do that. But it only works if we let you produce an app for yourself that doesn't have any lock into Tailscale.

MATT SIEGLER: All right. Let's suppose I'm more like an Avery, and my little thing gets kind of popular, and it gets picked up by some major publication or whatever you want to call it. And now 100,000 people want to use it. And I hire a marketing department, and I'm now selling it in 20 other countries. And I got to scale up, and I got 50 employees, and I got a payroll and investors, and, oh, shoot, I'm actually not an engineer anymore. What's something people don't know about from--

AVERY PENNARUN: The bane of my existence is this accidental popularity.

MATT SIEGLER: Yeah. What's it like? What's it like to go from, I was writing code to now I'm running a company, and now I'm being interviewed about that? What's that arc like?

AVERY PENNARUN: I mean, it's honestly not as fun as I would like.

[LAUGHTER]

And I think some people are born to be CEOs. And I was-- I was definitely not born to be a CEO. I've done startups before, and I've worked at big companies. And the thing that has frustrated me the most is handing over control of the business side, of the cool thing that I built. And then the cool thing I built, in my opinion, is, in fact, that cool, and the engineering is done very well. And then it still fails or gets canceled or whatever for dumb reasons-- in my opinion, dumb. Like, I look at the business people, and, look, you just made an obvious mistake. And it's very easy looking up from below and saying, ah, you ding dongs up there made an obvious mistake. Sometimes true, sometimes not.

Again, those people up there are not geniuses. They don't have access to all the information in the world. Quite often, the obvious mistakes are, in fact, just obvious mistakes, because we all make mistakes.

But for me, the thing that eventually got frustrating-- I'm now pretty old. The thing that finally-- I just lost it is, but I need it to be my fault or I'm never going to be able to fix it. So even if I fail, I need it to be my fault so that I can learn from it and go do the next thing. And if somebody else is failing, it's like, OK, great, I wouldn't have made the same mistake as you, but how do I know I wouldn't have made just a different mistake? I need to go through that process.

And with Tailscale so far, that is-- I've learned a lot of stuff. I have definitely made a lot of mistakes. But I guess they've averaged it out to being pretty successful so far. But there was no guarantee of that. And I think even if I fail now, completely, I will be sad for a while. But for my personality type, I'm going to feel better about being sad about, hey, I screwed up and learned something, than, oh, somebody else screwed up, wasted years of my life as a result, and I didn't learn anything because I wasn't doing the job that screwed up.

JORDAN GREENBERG: That is a perfect way for us to end our conversation. I really appreciate how direct you were, and it was awesome. Where can we find you on the internet besides your blog and gfblip.appspot.com, which we have to thank you for as well?

AVERY PENNARUN: I'm glad you like it. I also have a program called [? ISO ?] Ping, which you can find on GitHub. It's sort of like gfblip, but it's bidirectional. It'll tell you if the downstream versus the upstream direction is broken, which you'd think would be impossible, but it is possible, turns out.

But yeah, apenwarr.ca is my blog. I'm also on Bluesky under apenwarr.ca because they let you name yourself after your domain name. I have an account on the formerly Twitter, but I don't use it very often for various reasons. But I'm there. And yes, there is the corporatized version of me on LinkedIn if you're that kind of person. I'm happy to connect to you there.

JORDAN GREENBERG: Awesome. Thank you so much for joining us on the *Prodcast*.

MATT SIEGLER: Thanks.

JORDAN GREENBERG: And have an awesome day.

AVERY PENNARUN: Thank you.

JORDAN GREENBERG: You've been listening to the *Prodcast*, Google's podcast on site reliability, engineering, and production software. Visit us on the web at SRE.google, where you can find books, papers, workshops, videos, and more about SRE. This season is brought to you by hosts Jordan Greenberg, Steve McGhee, and Matt Siegler. The *Prodcast* is produced by Paul Guglielmino and Salim Virji, with contributions from many SREs behind the scenes. The *Prodcast* theme is "Telebot" by Javi Beltran and Jordan Greenberg.

[JAVI BELTRAN AND JORDAN GREENBERG, "TELEBOT"]