EPISODE 1123

[INTRODUCTION]

[00:00:00] JM: Anduril is a technology defense company with a focus on drones, computer vision and other problems related to national security. It's a full stack company that builds its own hardware and software leading to a great many interesting questions about cloud services, engineering workflows and management.

Gokul Subramanian is an engineer at Anduril and he joins the show to share his knowledge of how Anduril operates and what the company has built.

[SPONSOR MESSAGE]

[00:00:30] JM: Today's show is sponsored by StrongDM. Managing your remote team as they work from home can be difficult. You might be managing a gazillion SSH keys and database passwords and Kubernetes certs. So meet StrongDM. Manage and audit access to servers, databases and Kubernetes clusters no matter where your employees are. With StrongDM, you can easily extend your identity provider to manage infrastructure access. Automate onboarding, off-boarding and moving people within roles. These are annoying problems. You can grant temporary access that automatically expires to your on-call teams. Admins get full auditability into anything anyone does. When they connect, what queries they run, what commands are typed? It's full visibility into everything. For SSH and RDP and Kubernetes, that means video replays. For databases, it's a single unified query log across all database management systems. StrongDM is used by companies like Hurst, Peloton, Betterment, Greenhouse and SoFi to manage access. It's more control and less hassle. StrongDM allows you to manage and audit remote access to infrastructure. Start your free 14-day trial today at strongdm.com/sedaily. That's strongdm.com/sedaily to start your free 14-day trial.

[INTERVIEW]

[00:02:00] JM: Gokul, welcome to the show.

[00:02:01] GS: Thank you. Thanks for having me.

[00:02:04] JM: You work at Anduril, which is a modern defense company. Describe the canonical engineering problems of Anduril.

[00:02:11] GS: Yeah, absolutely. Anduril is really – I'll just take a step back and say, at a high-level, Anduril is really here to solve the hardest and most complex problems in the defense space. And, really, if you think about the defense, it's a combination of hardware and software and bringing those two things together in the real-world for real-world applications. At the highest level, those are the canonical problems that we end up solving.

If I think about Anduril as a company, we're really software-first company. So things like how to we do computer vision, machine learning, autonomy on hardware platforms. Those are the challenges that we're looking to solve. And those are the problems that we're actively working on.

[00:02:57] JM: You mentioned computer vision. How is computer vision used at Anduril?

[00:03:01] GS: Yeah, absolutely. For us, computer vision is one of the many sensor modalities that we are integrating in real-time on both our own platforms that we build. For example, our Sentry Tower, our Ghost UAS, our Anvil Interceptor Drone. All of those systems, these are airplanes, helicopters, towers, have cameras on them. And we are processing those members in real-time to detect objects of interests utilizing modern computer vision techniques.

[00:03:35] JM: Okay. And we should say at this point that Anduril makes lots of use of drones. Could we talk a little bit about the drones and where are these drones from? Are they built within Anduril? What kinds of roles are drones fulfilling?

[00:03:54] GS: Yeah. Great question. We've got a number of different products. In terms of drones, we have two major drone platforms. We've got one called our Ghost UAS, and that's roughly a 30-pound drone that we built in-house. It's a helicopter drone. So, vertical takeoff. Can fly for little over an hour. And it's really used for what's called intelligence surveillance reconnaissance. If I am in a new area that I know little about or I'm trying to defend an area, the drone's purpose is to be able to surveill that area with very little operator input without taking up a lot of manpower and be able to alert you if there are things of interest going on. So that's our Ghost UAS.

And the second one is Anvil Interceptor Drone, and that's a much smaller drone. Roughly about 5 to 10 pounds. And its job is to defend airspace. So the idea here is if we have an adversary sending a drone at a base and we know this has happened to the United States multiple times. The Anvil's job is to fly out, intercept that adversary drone or potentially an adversary weapon and smash to it. Deter the weapon from getting to the US space.

So these are just two of the drones that we built. We got a number of other products. Some of them are not drones. Our Sentry Tower is another good example. And all of these, I would say the really interesting thing going on here is that we do all of our processing in real-time on these systems. So they're constantly sensing their world through computer vision as we discussed in other modalities. And we're processing and making decisions in real-time on these systems.

[00:05:32] JM: Real-time is of course a loaded word. Nothing happens in real-time. There's always some kind of loop or series of steps that must be taken in a real-time system. Can you talk about latency sensitivity and how much latency you can afford in these kinds of applications?

[00:05:52] GS: Yeah, absolutely. Let me take one of use cases and we can dive in deep on it. So with our counter UAS system, which is a combination of our Anvil Interceptor Drone and our Sentry Tower, those who have to work together in "real-time" to defeat a threat. As you said, real-time is a loaded word, but we're talking on the order of milliseconds that those two are communicating back and forth with Sentry Tower detecting threat and telling the Anvil drone where to go to intercept it. And we are – So things specific to our parameters I can give you, our tower is detecting and tracking targets at 20 Hz. So 20 times a second. It's constantly updating the position of where a target might be, and our Anvil drone is going after that position at that same speed.

On board, the Anvil drone is able to detect with its own cameras and sensors at upwards of 30 Hz. So 30 times a second the position of a threat drone, and it's constantly updating its own internal navigation and guidance system based on those inputs.

[00:07:03] JM: Computer vision is a very broad field. There're a lot of off-the-shelf models that you can take these days. You can also train your own machine learning models. What kinds of machine learning models have you built for computer vision? And are there things that you can take off-the-shelf?

[00:07:19] GS: Yeah. Definitely, we leverage a lot of what's been done in academia and in industries that are off-the-shelf or open source. So we can broadly take two different classes of machine learning that we're doing. We're doing computer vision. In that case, we're looking at mostly optical data, electro-optical data and we end up using off-the-shelf models and then retraining those models with our own adjustments, parameter optimization, hyperparameter optimization and data that we collect on our own.

So a good example of this might be our Sentry Tower has the ability to detect drones in the sky. We start with an off-the-shelf model. We might use like YOLO v4 or Faster R-CNN, two of to the most popular object-detection models. And then we've done significant adjustment of those models, because in the open source community, there's not a lot of drone training data available. And so we refine those models with our own data and our own training techniques to get really high performance of the threats that we're interested in.

We also do a lot of machine learning on sensors that are not optical. And so a good example of this might be radar. And so we've got our own machine learning models that are looking at radar's measurements coming in from one or more of our sensors and doing things like target classification. So I see radar returns out in the environment. Is that a drone, or a bird, or is that car driving down? We use a lot of machine learning techniques to make those decisions or make those inferences.

[00:08:50] JM: Let's talk more broadly about things you can take off-the-shelf. You obviously use some infrastructure as a service or some platform as a service. Are you heavy users of AWS? Google Cloud? What's your cloud provider of choice?

[00:09:06] GS: Yeah. We are roughly cloud provider agnostic. So our customers have asked to us use AWS certainly. We also use the Microsoft Gov Cloud to a high degree, especially that's the direction that our customers are moving in due to kind of security requirements.

The other thing is that – And the reason we're cloud provider agnostic is we end up having to build our own clouds often at the edge, so to speak. At a remote outposts, let's say a forward-operating base that has no Internet connectivity or very weak Internet connectivity. We end up effectively building our own data infrastructure out there. And so we have to basically be able to spin up all of our infrastructure on just standard server racks, if you will, or a ground control station. Basically, whatever we're given.

So our infrastructure, one of the key details that we thought through when we designed it from the beginning was the ability to scale up and down based on the resources that we have available. So the same infrastructure can run at a remote forward-operating as what we can use when we have the AWS Cloud at our disposal. And then how do you synchronize data between these two systems when connectivity is present?

[00:10:18] JM: Can you use the AWS Outpost, the on-prem AWS instances?

[00:10:25] GS: I'm not actually familiar with the specific term you're describing whether an AWS has on-prem instances. I'm certainly not familiar with us using that. But I could figure that out or I could get you in contact with the right person there. But for us, most of the time, we are essentially just spinning up either Docker containers or Kubernetes clusters right on our own hardware or customer hardware that are provided to us at these forward-operating bases.

[00:10:52] JM: What's the continuous integration and continuous deployment system like for software that's getting deployed to drones?

[00:11:01] GS: Yeah. This has been one of the biggest challenges that we've had to solve as we’ve scaled as a company. It's one thing to be able to hack on a drone when you're a 10-person company, or a 50-person company and everyone can kind of say, “Hey, this is my drone. It’s going to sit on my desk and I’m going to a flash code on it today.” And it's another thing to do that, like you’re a 250-person company, a 300-person company and beyond as we continue to scale. So we’ve really invested a lot in the CI/CD process. The ability to check out hardware. The ability to do testing in simulation, software in the loop and hardware in the loop simulation.

So we built a lot of that recently. We leveraged CircleCI. And we have GitHub enterprise as kind of the backend. And then what our systems allow us to do is we can check-in code into GitHub and have that automatically statically checked. Run through software simulation. And then we create basically a binary that can get flashed on to any of the drones. And use a package manager called Nix as the kind of baseline to load our software on to these drones. And using Nix, we can control the drone all the way down the device firmware level. So we’re very vertically integrated. So if a new build of our drone software requires a change to a kernel level driver, we can do that all the way through our build system and just get that flashed on to the drone and then take it for a flight test.

[00:12:27] JM: The static analysis that you run in that process, is that your own static analysis tooling that you’ve built?

[00:12:34] GS: Yeah. I think it's a combination. And this is again where this is not my personal area of expertise. But it's a combination of tooling that we’ve built and also standard tooling that exists in the environment. All the way from just linking the code to looking for very specific types of vulnerabilities, and then finally doing software in the loop testing to validate that this doesn't break any of our baseline functionality.

[00:12:58] JM: What about machine learning models? Does the deployment process differ at all for the machine learning training pipeline?

[00:13:05] GS: Yes, absolutely. So the challenge with the machine learning models, and this is the area that we're investing in heavily right now instead is that it's very data-driven, which means it's a statistical process. And so the idea of a unit test or a pass-fail test is hard to describe in a machine learning model, because by definition when we update the model, it will not do the exact same thing it used to do before.

Let's take a simple example. If we have a set of baseline images and we have a model that could detect drones in those images. The version 1.0 of that model might have generated a specific bounding boxer on the drone on a specific image. And when we updated to version 2.0, it may not generate that exact same bounding box. It might generate a bounding box that slightly smaller, or slightly bigger, or slightly offset. It doesn't mean we fail the test. It doesn't mean the machine learning model is not working. It's just doing something slightly different.

And so how do you capture that in the sense of a unit test? How do you use metrics to start capturing these things picking at higher levels? If we have a tracker, it's generating all sorts of tracks through a machine learning process. What is it look like for that tracker to be getting better? What are the statistical metrics that we can capture? That's the area that we're investing in heavily. We still model this as unit tests to us. We have datasets that we captured in our machine learning pipeline that are used as reference datasets. And then we run new models against those reference datasets to validate that we're getting the same precision, the same recall or better. Are we getting the same track heuristics, track quality, track accuracy? Things like that. But this is an area where it's very statistically- driven instead of black and white.

[SPONSOR MESSAGE]

[INTERVIEW CONTINUED]

[00:16:16] JM: The customer interactions with the Anduril technology, what have you learned from actually deploying these pieces of hardware and seeing them being used in the wild?

[00:16:29] GS: Absolutely. I think the really cool thing about working in our industry is that our customers also end up being our partners. So I've worked in this industry for about a decade. And every customer that I've worked with, they're not just buying your technology and going to just expect the world of it. But they also want to work with you to make it better. I would say that that's different from, say, like the self-serving car industry where you don't buy cars so that you can go like work with Elon Musk to build a better Tesla. Or when the car gets there, you're not sending your ideas back to Tesla and saying, "Hey, can you do this? Can you add this feature?"

What's really cool about working with our customers is they're excited to go down this journey with us. So some stories I can share here are, first, Anvil interceptor. Before we even had a name for it, we that built that drone in six weeks and we entered it into a government competition, and we ended up winning that competition outright. We did better than many of the other companies that have been in that industry for many, many years.

And what happened after that was really amazing, and that customer said, "You've got a really great product. It doesn't do everything, but it does something that's really special. We want to invest in you and we want to work with you to make this a better product." All the way from telling us, "Here are the specific ways we want to use this. Here are some ideas for improvements that could be made." Our original version of our interceptor drone did not have a radar onboard. And that was suggestion that our customer gave to say, "We think you guys could make this better by doing that." And so I think the feedback loop is this really amazing in the defense space, and I think it's one of the differentiated between the defense space and maybe commercial product development.

[00:18:10] JM: The stack for software the gets deployed to a drone, can you talk at all about the software architecture of what is on a drone? We've done a few shows about drones. I remember we did one about Airware a while ago, and Airware was a company that they used I an operating system, RAS. They took a lot of stuff off-the-shelf. But tell me what you see as the architecture of the Anduril stack.

[00:18:41] GS: Yeah. So our software stack is, in large part, again as I mentioned earlier, we use something called NixOS, which is not only the operating system layer, but it's also our package manager, which enables us to control everything from the kernel level all the way up to the exact pieces of runtime that are going to run on our hardware. On the software side, all of our code is written in C++ that runs in real-time on the drones. Some major parts to callout are communications is one of the most important things when you're flying drones, or any kind of thing in real-time. So how do you make sure that the data is getting to the drone and how do you route it around to all the other assets that are in the sky. We can have 5 to 6 helicopters flying around at the same time and 2 or 3 interceptor drones in the air. We can have towers on the ground. All sorts of stuff, and they're all communicating seamlessly. And they're sharing data with each other and helping each other do their mission.

And so how we communicate around the stack is very, very important to us and another area that we've invested. And we have a piece of technology called Flux, which is our mesh routing layer. And it allows us to basically move data around using very simple pub/sub mechanics, published/subscribed mechanics, and we take care of how we move it around the mesh. So an example of this might be a tower might want to tell a helicopter drone where it sees a threat. And it might not have direct line of sight or even a communication link with that heli drone. The ghost drone might only be able to see another ghost drone, which can see a ground control station, which can then see the tower based on antenna links, signal strength, all that stuff.

Flux is the layer that we've built to move data around without having engineers to think about the logistics of moving that data. So that's a part of our Lattice system that gets deployed on every single drone is this Flux layer. Another layer that is part of Lattice is the autonomy layer. So how do our systems think and can make sense of the world? And in the final part of our system is the perception layer. And so I would say those three layers; communications, autonomy and perception make up the major parts of our stack.

[00:20:50] JM: What about programming language choices? When do you need to use C++? When can you use a higher level language?

[00:20:58] GS: Yeah, this is I think one of the central questions for us, which is that research ends up being done in a high-level language. Our idea almost always ends up being time done in Python, or JavaScript, or MATLAB, for example. But in production, we want to run everything in C++, because we want to run time speed. We want the guarantees that C++ give us, right? It's a compiled language. And so you get a lot of benefits in terms of access to lower level memory logic and control over the runtime. But at the same time when you're doing machine learning model development, for example, or you're doing autonomy algorithm development, you want end up doing that in Python.

And so we end up supporting both of those. We've built libraries so that we – For example, our vision library enables us to do our model development in Python and then transpile that model into codes that's optimized for running at the edge and had guarantees that what I developed in Python and the metrics I'm getting there will work when I get it on to a drone and it's running in C++. And so that takes away a lot of the guesswork for engineers and it gives us a platform to build upon. So I would say that, for us, we do a lot of the R&D in Python or MATLAB, and then we have a team that's focused around how do we ensure that the C++ runtime will match what we're doing in Python.

[00:22:30] JM: What about the team structures? This is not a straightforward SaaS company. Are there any ways in which structuring teams or the management department is unique that would be informative?

[00:22:44] GS: Again, we think of ourselves mainly as a software company. The Lattice platform is really the thing that Anduril brings to the table and whether that platform is running on our hardware or another customer's hardware, or another company that we're partnering with its hardware. Our company organization is really built around Lattice.

I would say that the leadership in the company and the way that the company decisions are being made is really communal in the sense that the strong partnership across the perception team, for example, which I lead versus the autonomy team, versus our platform teams that are building, the tooling and infrastructure for the engineers and also through all the way down to our hardware teams. We're very much a matrix organization. And so you've got your areas of expertise, for example, perception, autonomy platform, UI, UX. And then you've got the programs and projects that you're working on, for example, counter UAS, or Ghost Helicopter UAS, and so we matrix our teams on to those specific projects. And we have program leads that are technical running those projects working closely with product managers.

[00:23:56] JM: How do you benchmark the quality of your machine learning models in your sensor processing models?

[00:24:03] GS: Yeah. Again, this is an area where we think of this as building blocks, or is a pyramid, right? You've got to make sure that your algorithms work in isolation. That's the first parts. So each of your sensors are running sensor processing often with machine learning. The simple example would be a camera on a drone. It can detect things on the ground. And so start to look at classic computer vision metrics, for example, precision, or recall. Precision being of the things that the model is detecting, what percentage of them are accurate? And recall being of the days that we wanted to detect, did we detect what percentage of those did we detect?

Those are kind of model-specific metrics. But then a big part of what the Lattice system does that we're building is the sensor fusion. How do you put all of those things together and go from sensing to sense making? And so that's like saying I have my camera detecting things. I have a radar detecting something. I have a sonar detecting something. I have a LIDAR detecting something. How do I put all that together and say, "That's a car and it's driving right at me. And that car might have ammunition on it. I got to be worried about that."

For that, we need higher level metrics. And so we do a combination of things. We have track level metrics, track purity, track accuracy, things like that when we're tracking objects. But then we also do – One of the big strengths of Anduril is that we do test events very, very regularly. We're testing every other week, our entire Counter UAS system. And so then we're measuring metrics that a customer might care about. So we're flying drones at our towers. Every other week, we fly two dozen drones and we say, "How many of those drones did we detect? Were we able to detect it early enough? Were we able to respond to that drone? Were we able to take drone out of the sky that’s coming to attack us?”

And getting engineers be able to just go out to the test site and see that progress happening and work directly against the end metrics that we’re trying to solve, that's another way we keep ourselves honest and make sure there we’re building against the right thing.

[00:26:13] JM: Are there any kinds of restrictions or additional strictures that come into place when you're building software for national security needs? Are there any kinds of tooling that you need to be aware of? Any kinds of additional testing that needs to take place?

[00:26:29] GS: Yeah, absolutely. I think that when you're building things for the national security needs, definitely, security becomes paramount and it's something our customers expect of us and what we expect of ourselves. And so from the simplest things like ensuring that we’re not putting any location information into our binaries or into our source code that could identify where something was built or could be reverse engineered. That’s your base layer. But then also thinking about how do we ensure that our communications lengths are secure. What happens if an adversary gets a hold of one of our drones? Can they hack into the rest of the network?

And so thinking of how do you build a trustless network where you can revoke authority has been something that we’ve designed from the ground-up in our system. And the Lattice system has the ability to identify rogue actors for users to take those out and to react to malicious threats entering the network.

The other thing that we’ve done is that we focus very heavily on ensuring that all of our data does not have to go back to a set of cloud. Again, if you think about classic Silicon Valley samples; Facebook, Google, Tesla who are doing machine learning, all of them are designed around this concept of I’m going to collect data from the environment, then I’m going to bring it all back home. And I’m going to get it into my cloud and I’m going to start doing my machine learning at scale on my supercomputer cloud centers.

For us, we know that when our hardware and our systems end up in the real world, often that data can't come back to us for security reasons. And so we have to think about how do we do machine learning at the forward edge? How do we update our models at the edge? How do we give that power to our end users so that they can train our models or they can refine our models? And so that's been something that has been core to our system from the day we designed it.

[00:28:22] JM: When you say perception of detecting perception around a drone, what does that actually mean? Do you need cameras all around the drone? Do you need thermometers? Do you need accelerometers? What do you need in order for a drone to have what we call perception?

[00:28:40] GS: Great question. So I think the list of sensors is always endless. We could always add in additional sensor. But at the end of the day, it's the outcomes that we're looking towards. So, what would it take for our drone to be able to detect a threat approach a base? That's the way we think about it. Not that the drone has a sense of everything at all times, but it has to solve specific problems for our customers who are in some of the most dangerous parts of the world. For that, again, we work backwards from what is the mission the we have to solve and then what are the sensors that we need to put in there?

I can tell you, for a Ghost UAS, it's got a combination of electro-optical and medium-wave infrared cameras. It can optionally be shipped with a radar payload. And that actually just constitutes the majority of the sensors that we put on that UAS just to sort of weight and power reasons. On our power, we have any number of sensors, because we know that we'll have more power available to us. And so we've got thermometers, we've got wind sensors, we've got a bevy of cameras. We've got multiple radars. You name it.

The other thing that I think is important to describe here is that for the system to achieve its mission, and again if it's detecting a threat coming towards a base or identifying a hostile actor. It's a combination of sensing and perception and also autonomy, right? So in order to sense something, you have to be at the right spot at the right time or you get to come around from the right angle. Even if you have the best camera, the camera can't look through a wall. So the drone has to be able to recognize that there's a wall on my way. I need to move around to this corner. Or there's a tree blocking my line of sight. I need to move so that I can get the right look angle. And we've designed our system to be able to do that. Not just autonomously. So our system and do things like scan and track. Scan and area and our drone can identify something and track it. But also take cue from operators. High-level playbooks. And you turn those into missions sets that it can run.

[00:30:43] JM: Well, tell me more about synthesizing all the different sources of perception.

[00:30:49] GS: Sure. Again, the way I think about this is, the first layer, you're just doing sensing. At the next layer, what you're starting to think about it sense making. I'll give you a specific example. So, going back to detecting enemy drones that might be approaching a US base, and we do that through a combination of our Sentry Tower and our Interceptor UAS.

At the first level, you've got all these sensors, and each of them alone might be very weak. So a radar can send anything that's moving in the environment, but it's littered with false alarms. A tree flows in the way and a paper bag flies around. A bird flies around. A radar is going to pick that up. So it's got it strengths, but it also got its weaknesses. A camera can give you positive identification. You could look at something and you can say with some confidence that might be a drone. But the camera can only look at one thing at one time, right? The camera's field of view is very narrow. You might have a couple of different cameras. And so you have to figure out when you want to use each one.

The next domain might be the radio frequency signature. Drones are communicating back to their operator, or they might be communicating back to their operator. So we can pick up on that signature. Combining all of these things requires what we think about as sense making. We use a combination of tracking technology. So common filters and the like, and we also use machine learning to start to say, "The radar is telling you there's something moving over there. I can get camera on top of it," and the camera is telling me with some confidence I think that might be a drone or a bird. And I've got RF signature emanating from that area. I can now put something together where the whole is greater than some of its parts. I can tell a user, "Hey, we think with very high confidence based on these three different sources, each of which is not sure on its own, that we think there's a drone approaching your base."

[00:32:42] JM: Has there been anything that is harder than you expected or easier than you expected? Surprises that have come up as you've been developing this technology?

[00:32:53] GS: Sure. Putting things together is always much harder than it seems at least on the outside. We're software engineers at heart at Anduril. And so when we built the Lattice platform we thought we're going to get all this data coming in and we will be able to generate this very exquisite understanding of the world. But actually getting that to run in the real-world and getting your drone to fly and do the thing you ask it to do, getting that last mile, especially when you're touching hardware, is exceptionally difficult and exceptionally fun at the same time.

And so I think that has been the biggest challenge, but also the most fun and rewarding at working at Anduril. I can think of many other companies where you'd get to go out to a test site in beautiful Southern California weather and you can watch your drones fly around. And then in real-time, make some software updates and figure out, "Oh, okay. Now I can get the perception really dialed in."

And so it's been both a blessing and a challenge. But getting things to work in the real world, not just on our systems, but on our government customer systems and in our partner systems has been a challenge. Whenever you're working with objects in the real world, they don't always behave what you want them to.

[00:34:10] JM: Can you say more about that

[00:34:11] GS: Yeah. I mean, just simple things like it's very hard to simulate real data from the real world. So when you're designing your algorithms, you might get a small tranche of data for, let's say, radar. We collect hours 'worth of radar data. And then when we go to the new environment there we're going to test in or it gets deployed into the customer environment, it turns out the radar signature looks nothing like you expect it to. The radar is now mounted a little higher than you thought it was. It's getting multi-path from the ground. And so your radar reflections are nothing like you thought they would be.

How do you adapt to that? You have the best of plans in software, in your simulation, in your software in the loop simulation, in your hardware in the loop simulation. When you really get out to the real world, the world doesn't behave as cleanly as you thought it would. Or tomorrow it starts raining. And it turns out when it's raining, your camera can't see nearly as far as you thought you could. It's just fogging up your camera. So how do you adapt to that? Those are always challenging things.

And then, certainly, when you're building drones, things are going to break. We've had our drones crashed, dozens of drone crash I the sky. We've had algorithms that we think were going to work, and they've worked in sim. Dive down straight into the ground. So building not just a very safe test environment for engineers, but also building the freedom for engineers to get out there and make mistakes and iterate from those mistakes just like you would at any other software company. I think that that's the challenge and also the strength of Anduril.

[00:35:41] JM: How big is the Anduril team at this point?

[00:35:44] GS: We're closing in on 300 people at this point. So we're so growing pretty rapidly. I can't give you an exact number, but we're probably two 270 to 300.

[00:35:52] JM: What are the challenges of onboarding that many people that quickly?

[00:35:56] GS: Yeah, great question. Everything is easier when you're very small. Communication is very easy when you're small, because everybody knows each other by first name basis. Everybody knows kind of knows each other's life stories. They're friends of each other outside of work, and they cannot understand. They can empathize with each other in where they're coming from.

When you scale, you start to build. You have team structures. And I know this is my team and that's their team, and our team does this. If your team does that, stay out of my turf and I'll stay out of yours. That's the thing that we've explicitly tried to avoid as we scale this company, is how do we grow and still feel like we're a family and we're still working together and we're all in it for the common good and the common missions that this company shares?

I think being a matrix organization really helps with this, because you've got your homeroom, your home base in terms of the area of expertise that you have. For example, perception engineer. But you end up touching all sorts of different parts of the company when you work on a project together. For example, our Ghost UAS or our Sentry Tower, and keeping people moving across projects and touching different parts of the organization helps everyone still feel connected even at the scale 300 employees and as we grow out to 3,000 employees one day.

[SPONSOR MESSAGE]

[00:37:20] JM: If you are a retailer, a big sales day like Cyber Monday can make or break your business. If you sell accounting software, the tax deadline day is like your Super Bowl. And if you're a sports broadcaster, then the Super Bowl is your Super Bowl. Every company has days or seasons that are more critical than the rest. If your systems are ready for the moments that matter most to you, then you're going to be doing much better. And that's the theme of this year's Chaos Conf, the world's largest chaos engineering conference. Chaos Conf is a free online conference taking place on October 6th through 8th, and you can register at chaosconf.io.

Ever since the first Chaos Conf in 2018, the objective has been to create a community around resilience and SRE best practices. Attendees range from having a decade of experience to those who are totally new to chaos engineer. And this year's keynote speakers are Gene Kim, and has been a guest on the show several times; and Adrian Cockroft, who's the VP of cloud architecture strategy at AWS. There will be 20 sessions for all experienced levels focusing on the practice of reliability, completing the DevOps loop and how to build a data-driven culture of reliability. You can register for free at chaosconf.io and the first 1,000 registrants will receive a limited edition swag pack. Claim yours at chaosconf.io and be prepared for the moments that matter.

[INTERVIEW CONTINUED]

[00:38:51] JM: There are several different products that you have at Anduril. You have this UAS system. You have the Sentry Tower and you have the Ghost Drone. Could we go through each of these? Let's start with the UAS system.

[00:39:06] GS: Sure. Yeah. I think before we touch any of our hardware systems, it's worth mentioning the thing that every single one of our hardware systems and any system we deploy on with one of our partners shares, and that's the Lattice system that we've build. That's the Lattice software systems that we built. That is the brains behind everything we deploy. In fact, the code running on any of our systems is actually more similar than different, because they're all running the same Lattice codebase. Whether it's our tiny dust sensor that I'll talk about in a second, to our giant tower, they're all running Lattice. They're all running effectively the same perception modules, the same autonomy modules, and basically the same communication modules. Everything is shared at that pool.

So if you think about Anduril as a company, what we build is Lattice, and these hardware products are just mechanisms for us to get Lattice out into the world. In terms of the products that we build, we do build the Sentry Tower. The Sentry Tower is basically a fixed site defense solution. So if you've got a base, and we've deployed over 100 of these towers at this point in multiple continents. If you've got a base you want to defend or a perimeter you're trying to secure, the Sentry Tower's job is to sense that perimeter and detect anything that might be moving out there, or stationary, actually. And we can do that for both ground targets, so vehicles, people, objects of interest, but also air targets. So that includes missiles that might be in the sky, or drones that might be in the sky. Our Sentry Tower can detect all of those things and provide persistent awareness.

Another product that we've build is the Ghost UAS. The Ghost UAS is kind of the forward arm of the Sentry Tower, you can imagine. Because the Sentry Tower can't move, it has a fixed area it can protect. The Ghost UAS can go out and search and it's got just over an hour of battery life. It's a 40-pound drone. And so that means that man portable. A single person can carry this drone around. Set it up with a single laptops, fly it. And it's basically a point-and-click interface.

And the drone can go out and do things like search for anything moving in the environment, or hover in this position and look for objects of interest. Or work together with the Sentry Tower to go and positively identify anything that the Sentry Tower detects. Those are the kinds of things that we do our Ghost UAS.

The next product that we build is the Anvil Interceptor System. Again, this works with our Sentry Tower or Ghost UAS, and that is the kinetic defeat component what Anduril builds. And again, this works because we have Lattice running on all of our systems. And so the same code can basically talk to each other and a tower can say, “I think there's a drone out there. That drones is bee-lining towards our face. It might have ammunition on it. We need to defend the base. We can launch our Anvil UAS.” Again, this still requires no operator control. The operator just gives it the go button and says, “Go out there and take that thing out.” The Anvil opens up a launch box. Fire out of it, very similar to like Iron Man launching – The Iron Man suits launching out of a box. And it goes and gets underneath an enemy drone. Gives the user a chance to confirm whether or not that this is an actual threat. And then it can ram into it to take it out of the sky.

And then the last product that we build is our dust sensor. And so these are very small perimeter security sensors. It’s basically a camera in a box and they can sit unattended for up to a month. It’s an unattended ground sensor. So we use these in combination with the rest of our systems to protect the a perimeter. So you could imagine that tower can see behind a tree, for example. We can put these little unattended ground sensors nearby and let it kind of fill in the picture of the world. Again, this all works because they’re communicating with each other.

[00:43:03] JM: I think about the – If you've ever played StarCraft or War Craft, you have a view of the map. You have this little map that you can see, and some areas of the map are black because you do not have insight into what is going on in that area of the world. It seems like your system is built to kind of remove the blackness of the world and to survey the area around the Sentry Towers together with the ghost drones. Can you just tell me more about like the stack of the different systems? Like what the Sentry Tower does and synergy with the ghost drone, and what the overall vision is for these different systems interacting with each other.

[00:43:45] GS: Yeah, absolutely. I'll relay a very quick anecdote that I think describes perfectly with what you're saying. And I completely agree with you, which is If you think about that map at StarCraft. The map initially starts off completely pitch black, except for maybe your own position. And as you reveal more and more your map, you actually understand what's happening in in the world and you can make more thoughtful decisions because of it.

We have had multiple customers come back to us after deploying our stack, whether it's our towers, or our drones. And we've asked them, "How is it been? Is it useful? Have you guys able to do less work now because now you've got the tower defending your perimeter or the drone flying around for you?" And they say, "No. Actually, the opposite way. We've been working harder than ever before because we had no idea all these things were around our base. Or we had no idea about half these things happening in the world. And guys have given us insight into it. And so now we've be able to take proactive action or have been able to realize that something exists when we were flying completely blind prior to this." And I think that's exactly what you're describing with that analogy.

So, for specific examples that I can describe, I think both the Sentry Tower plus Anvil sample is very compelling, which is the tower is detecting threats and then allowing the Anvil drone to fly out and react to them. So the tower might detect an enemy drone 5 km away, for example, and get a positive confirmation of it, "Okay, this might be a threat." And then we can have the tower and the drone work together sharing data to get the drone in position to lock-in. So the drone flies out based on the detections of the tower is getting.

Similarly, we can do the same thing with our Ghost UAS and our tower. If the tower detects something on the ground, maybe a vehicle, and we can send the ghost drone out to go follow that vehicle, for example. Our drones can also work with each other. We can things – Like we can have a drone with a radar on it. Detect something and then send over another drone with camera on it to get optical confirmation on the thing.

I think the best way to summarizes this is that everything we create is about giving operators a better understanding of what's happening so they can make better decisions. So it's kind of closing that loop with our customers.

[00:46:07] JM: As we begin to wind down, could you tell me more about the bigger future for Anduril? What are the other kinds of tools and systems that you expect your yourself needing to build?

[00:46:20] GS: Yeah, absolutely. I think we expect that the Lattice system is going to get used not just on our own products, but across the DoD and national security infrastructure. And that's what we want. We didn’t just build Lattice so that we could fly our own drones around. Although, certainly, we love to do that. But so that we can take all these amazing systems that the nationalist industry has developed and start to connect them all together and make better decisions based on them.

So what we’re starting to think about is how does Lattice work if we want to use it in space to control the satellite or move data around in space? How does Lattice work if we wanted [inaudible 00:46:59] on our 40-pound drone, but a 300-pounds drone on about an airplane flying in the sky? What about all sorts of other use cases for our system? And really the thing here to think about that we’re thinking about is how does this work at scale? What does it mean when hundreds of systems are communicating with each other and working together and working with operators? Not just operators that are right beside the drone, but people that could be hundreds of miles away.

We really believe that at the end of the day, if we create systems for a better understanding of the world, our users can make better decisions. Ore military can make better decisions. And so that the future that that we are trying to build.

[00:47:46] JM: There are lots of different hardware and software systems that you've built yourself. And there's also lots of things that you're taking off-the-shelf. Are there any challenges to integrating third-party hardware, third-party software with the stuff that you've built in-house?

[00:48:06] GS: Yeah, absolutely. We worked very closely with the companies that we partner with to make sure that we’re getting the best and the most out of their systems and sensors.

Almost always when we start a partnership with another company, we're not using their sensors to their best effect.

I can give you one example where when we started working with a vendor for a radar, we notice that it wasn't seeing everything it can see. And so we had to go back with them a number of times to open up that radar system. Work with them to get the most out of it, because we can do things with our Lattice platform that they couldn't do as a sensor provider by themselves. We could run their radar in a more open configuration knowing that our Lattice platform could come in behind it and run machine learning over the top to reduce the number of false alarms it was generating, or use one of our cameras to work together.

And so the biggest thing for us has been – And our vendors and our partners are really excited about this, which is they've built these amazing products and we can take them to the next level by bringing them together and starting to combine them and use them in ways that hadn't been thought before by putting them all under the same platform.

[00:49:15] JM: Just to wind down, can you tell me about your own personal career, your journey and what led you to eventually joining Anduril?

[00:49:24] GS: Absolutely. So for me, I had always wanted to work in the national security space coming out of undergraduate and going forward. I had always felt like these are the most challenging problems and these are the most important problem or our country. They're challenging in the sense that you get access to things that no company could do on its own. No company can put up hundreds of satellites in the sky, put up hundreds of planes in the sky, can have the footprint that our military has. And they're also important, because I believe that our country has an important role to play in maintaining kind of the world that we live in and the productivity and growth in our country and around the world. So I've always wanted to work in this space.

I would say that there were two very interesting experiences in my career that led me to Anduril. The first one was I worked at a company called Palantir, which is very much the predecessor to Anduril. And Palantir is really a software 100% company. It was really thinking about how do you integrate data. How do you do data integration? How do you connect all these pieces of data with each other and how do you present them to a user? And then after that, I went and worked at a more focused research company that really looked at the most challenging problems in the DoD space, mostly once coming from places like DARPA that are thinking 5 to 10 years ahead. And how do we solve these extremely hard problems?

The things I learned from both spaces is that there are people in the research world who don't understand the software world. They can solve the hardest math problems. They can solve the hardest sensor problems, but they're not a Silicon Valley software company. And the Silicon Valley software company that I worked at didn't understand how to do these very challenging sensing autonomy perception problem that there's expertise in the DoD space.

The thing about Anduril is it brings those two worlds together. And that's what really excites me about it. We're a first-class software company, but we're also working with real sensors. We're building real drones. We're putting things in the sky or we're working with partner companies that can do that. That's really, really special. You can find that at other companies in the bay. Nor can you find that in your traditional DoD contractor primes. Putting those two talent bases together with the senior folks who've worked in either industry and just fresh minds coming out of undergraduate or graduate school, Anduril is a really, really special place, because it brings all those people together.

[00:51:52] JM: Okay. Well, Gokul, thank you so much for coming on the show. It's been real pleasure talking to you.

[00:51:57] GS: Absolutely. Thanks for having me, Jeff.

[END OF INTERVIEW]

[00:52:08] JM: Open source tooling is generally preferable to closed source tooling, because with an open source tool, you're going to know what code is running. You're going to know what the community is saying about that code. And you're going to have flexibility. But scaling open source tools is not that easy. You're going to have to spend a lot of time managing and maintaining that open source software. And the alternative is to use closed-source software, which will be scalable, but you won't know exactly what code is running. And you won't have an easy migration path.

Logz.io is a scalable and fully-managed observability platform for monitoring, troubleshooting and security, and it's all based on open source tools like the ELK Stack and Grafana. Logz.io is open source software at the scale that you probably will need if you are a growing company. Sign up for logz.io today by going to logs.io/sedaily and you can get a free t-shirt. You can go to L-O-G-Z.io/sedaily, create an account for logz.io and make a dashboard. Once you make that dashboard, you will get a free logz.io T-shirt.

Thanks for listening to Software Engineering Daily, and I hope you check out logz.io. That's L-O-G-Z.io/sedaily.

[END]