EPISODE 929

[INTRODUCTION]

[00:00:00] JM: Diffbot is a Knowledge Graph that allows developers to interface with the unstructured web as if it was a structured database. In today's show, Diffbot's CEO Mike Tung returns for a second discussion about how he's built Diffbot and how Diffbot is used. The web has many different entities; webpages, topics, people, stories, articles, companies, so much more.

Humans use a search engines to find answers to their questions within webpages. Machines need to find answers to these kinds of questions as well, but a machine is not sophisticated enough to figure out answers from an unstructured webpage. That's where Diffbot comes in. Diffbot brings structure to these webpages and gives them an API interface for developers to build on top of. In order to create this system in a cost-efficient manner, Diffbot runs its own data centers where web scraping, machine learning and API infrastructure are all used to build the Diffbot application.

Mike joins me for an interview about creating Diffbot as well as his strategy for running the business. If you're building a software project, post it on FindCollabs. FindCollbas is the company I'm working on. It's a place to find collaborators for your software projects. We integrate with GitHub and make it easy for you to collaborate with others on your open source projects and find people to work with who have shared interests so that you can actually build software with other people rather than building your software by yourself.

FindCollabs is not only for open source software. It's also a great place to collaborate with other people on low code or no code projects, or find a side project if you're a product manager or somebody who doesn't like to write code. Check it out at findcollabs.com.

[SPONSOR MESSAGE]

[00:02:05] JM: This podcast is brought to you by PagerDuty. You've probably heard of PagerDuty. Teams trust PagerDuty to help them deliver high-quality digital experiences to their

customers. With PagerDuty, teams spend less time reacting to incidents and more time building software. Over 12,000 businesses rely on PagerDuty to identify issues and opportunities in real-time and bring together the right people to fix problems faster and prevent those problems from happening again.

PagerDuty helps your company's digital operations are run more smoothly. PagerDuty helps you intelligently pinpoint issues like outages as well as capitalize on opportunities empowering teams to take the right real-time action. To see how companies like GE, Vodafone, Box and American Eagle rely on PagerDuty to continuously improve their digital operations, visit pagerduty.com.

I'm really happy to have Pager Duty as a sponsor. I first heard about them on a podcast probably more than five years ago. So it's quite satisfying to have them on Software Engineering Daily as a sponsor. I've been hearing about their product for many years, and I hope you check it out pagerduty.com.

[INTERVIEW]

[00:03:32] JM: I became obsessed with the circle of fifths recently.

[00:03:34] MT: Oh, okay. Yeah. That's very elegant.

[00:03:36] JM: Isn't it?

[00:03:37] MT: Yeah, it's genius really. Yeah. All of the skills, it's all math.

[00:03:40] JM: Its' totally genius. The circle of fifths, it's like the periodic table for music. I didn't know how it worked for a very long time, and then I read a little bit about it and I was like, "This is just amazing."

[00:03:55] MT: You should interview a composer sometime. It will blow your mind.

[00:03:59] JM: I'm sure it would. You're a musician?

[00:04:02] MT: I'm not, but I mean I used to play violin for like 18 years or so.

[00:04:06] JM: 18 years. Why did you stop?

[00:04:09] MT: I went to college.

[00:04:12] JM: You're not in college anymore.

[00:04:15] MT: I mean, violin is special to have like upkeep on your skills as a violinist. You need to spend three or four hours a day. You can't do anything else, really. If you don't practice that much a day, you just actually worse. It's not like piano where you can just like sit down on a keyboard and you can jam. You don't need to upkeep it. With a violin, you just get worse if you don't.

[00:04:35] JM: Right. Yeah, piano is cool because you just press buttons.

[00:04:38] MT: Yeah, and it will sound the same if you press it the same way, but now with a fiddle.

[00:04:43] JM: I started the recording. I hope you don't mind. I'm going to include that, because that's very cool, that you played violin for 18 years. I think I'll just also lead off with the fact that you're a patent attorney randomly?

[00:04:54] MT: Yeah. I mean, it was an interest of mine.

[00:04:57] JM: Maybe we can visit that a little bit later, but most people are probably tuning in to hear about Diffbot. So we should talk about that. The previous episode about Diffbot was really popular. People really liked it. In that last episode, we gave an overview for what Diffbot is. We talked about it. We talked a little bit about the engineering. But in case people don't remember or people didn't hear it, can you start off by just giving an overview of what Diffbot is? Give people a reminder?

[00:05:19] MT: Yeah. Hopefully this matches what I said last time. Basically we're an AI research startup. We're based now on Menlo Park. WE just moved to a new office about – What's it? Like three months ago from our place in Mountain View? But it spun out of the work I was doing at Stanford as a grad student. The mission of our company is to try to build the first complete map of human knowledge.

The way that we're trying to do that is by building an automatic system that's able to read and understand every page on the web, classify and extract into like a structure format. We basically offer this information that we extract as a service to customers. So it's an information service. We call it knowledge as a service, and we have about 400 customers that basically pay us to access this Knowledge Graph.

[00:06:06] JM: It's been almost exactly a year since we last spoke. How has the business changed?

[00:06:10] MT: Time flies.

[00:06:12] JM: Time does flies. I was like I don't know if it's been one year or three years.

[00:06:15] MT: Yeah. That's when we just launched the Knowledge Graph, and it just came out. We started talking about it, and I think the market has matured a lot over that whole year. I was just kind of talking about it earlier, but back then I would say – We're just explaining to people what a Knowledge Graph is and people are kind of looking at us weird. Now there're conferences about Knowledge Graph.

I spoke at one, Columbia University, a couple of months back, and there are other companies, they are talking about Knowledge Graphs, and there were people from – It was New York, so they were like people from the banks, from finance. Then – What is it? Gartner recently added Knowledge Graphs as a category now into their hype cycle. There're analysts now tracking the field where there wasn't much attention on this area before. So it's been amazing to me how it's kind of picked up even though this is like a really fundamental –

[00:07:07] JM: Yeah. You want to talk like kind of directly.

[00:07:10] MT: It's surprising to me how people are starting to notice it in the business world, but to us this is what we've been working on for many years. This is like a very fundamental problem of computer science that we're trying to solve.

[00:07:21] JM: Your goal was always the Knowledge Graph?

[00:07:22] MT: Yeah. I mean, back when I was working on AI at Stanford, the main bottleneck that I saw with these AI algorithms is lack of structured information. So structured data and knowledge is really what makes intelligent systems smart. While there are so many great companies out there that are focused on faster deep learning hardware, whether it's GPUs, or CPUs, or specialized asycs.

There are companies out there that are focused on making the software easier to use, like TensorFlow, like PyTorch and making the algorithms themselves better. People are competing on those models. There are not too many companies that are actually providing access to the raw information that AI needs, the actual knowledge. There is no like Amazon of data, like knowledge store. You just have to basically roll up your own sleeves and procure and acquire your own knowledge and then run those algorithms and technologies on your data.

[00:08:24] JM: It's funny, I didn't realized this analogy yet until just now. But did you ever see the company SafeGraph? Have you seen SafeGraph?

[00:08:31] MT: Yes. I think the CEO is like Auren or something? I don't know too much about it though. Yeah.

[00:08:35] JM: Auren, yeah. I like Auren a lot. I've interviewed him several different times. That's like his drumbeat the whole – He's like, "Why aren't there any data companies? We can't do anything without data. We can have all the algorithms in the world. We don't have any data."

[00:08:51] MT: Right. Yeah. I mean, there are some massive tech companies that have data, but they're not in the business of letting you use it, of course, in the camp, and nor should they be, because it's like your personal information. There's kind of like these big tech companies

that are able to do interesting things with data, but it's in the service of selling ads and trying to personalize things for you, right? You can actually as an entrepreneur, someone that's like wanting to do something interesting, leverage the kinds of information and technology that's out there to do that.

[00:09:23] JM: Of course, the data that they have is much different than the data that you have from your Knowledge Graph.

[00:09:29] MT: Yeah, that's another big difference, right? They couldn't even offer the same kind of service that we have, because their data is actually your private personal information. Like I said, they shouldn't be offering that as a service, whereas the Diffbot Knowledge Graph, remember how I said? Our goal is to analyze all the public pages on the web. All the information that's in our Knowledge Graph is public information. It's common information about people, organizations, products, places, all these just basic public information. It's not structured in a way that software can use it, right?

If you think about the internet, it's meant for people to consume. It's not actually structured in a way that you could have it, a piece of software or an intelligent system be able to actually use a webpage, just a document.

[00:10:16] JM: Is there enough volume of data for people to do interesting machine learning operations, like machine learning algorithms built on top of Diffbot data?

[00:10:26] MT: I mean, I think the internet is the world's largest manmade sensor basically. If you think about the sum total of all the information that's on the internet, if you were to print out all the webpages that are on the internet, I mean, it would cover over the earth's surface multiple times over.

Largely, once you encode information into that format of a page, or a blog post, or a news article, or a profile online, it's kind of locked in to that medium. It's not repurposive or usable by an intelligent system later on unless it's been structured in a way kind of like a database format, right?

The amount of information on the web, I think that surpassed any other manmade repository of information that's been created, but any kind of private or public entity, because it's the entire community of all the people around the world building the pages on the web. That's why that was our natural starting point to achieve our mission. The best place, if you want to build a comprehensive map of human knowledge is to start out with that kind of public resource that we have, which is the internet.

[00:11:34] JM: Do you feel like your business is accelerating, or do you feel like it's just been like a slow but steady linear growth curve?

[00:11:42] MT: I think it's still early innings, right? If I would have said earlier that people are actually starting to recognize that word. I mean, the vast majority of business people out there, organizations out there, they're not using that yet. It's still kind of in the early majority I would say.

Our business itself, I think it's pretty unique and that we have a profitable AI company, which is something that not a lot of AI startups can say and it's because really the only operating cost of an AI company like ours, which is like bandwidth and electricity. They're paying for the outputs this AI system is producing. So it can be an incredible business at scale.

[00:12:20] JM: It's already profitable, right?

[00:12:22] MT: Yeah.

[00:12:23] JM: We can talk a little bit more about the business later, but the web, it's always changing, and a lot of data that's getting generated today is on social networks. Social networks are this darker part of the web. I don't mean darker like negatively. It's harder to access.

[00:12:41] MT: Harder for search engines to crawl basically. Yeah.

[00:12:43] JM: Right. Can you index that stuff?

[00:12:45] MT: We don't crawl any parts of the private web, things behind the login, like social media, stuff happening in Messenger. We're focusing more just no like public information, public common information. Just to stay away from privacy issues and things like that. There's already plenty there just in terms of the public information that hasn't been structured. I don't feel like we need to go into that space yet of trying to derive structure from like people's private messages and stuff like that. People are welcome to apply off-the-shelf AI techniques and tools to those if they would like to for their business, but I don't think it needs to be part of our business right now.

[00:13:21] JM: What's the most surprising thing you've seen somebody build on Diffbot?

[00:13:25] MT: We get surprised all the time about interesting ways people are using our service, because a lot of customers come in inbound. They come to our site. They sign up and they start using our service. In terms of the categories of customers that we see, I mentioned, we have like about 400 or so companies that use it. There are people building consumer experiences using our data and there are people building enterprise experiences.

On the consumer side, there're basically major search engines that use Diffbot and app companies, consumer app companies. In terms of major search engines, we have companies like DuckDuckGo, like Bing, like Yandex that are basically using Diffbot to build these knowledge panels. Have you seen those things on kind of the right side of a search query?

[00:14:06] JM: Oh, yeah. Just like Google has.

[00:14:06] MT: Yeah, or you'll see like particle tiles. Yeah. They're essentially doing something like that, but the other search engines are doing that using Diffbot. So you see product tiles. You see like article tiles. Basically, nicer ways of displaying the search result, because you have structured data instead of just like a title and snippet. Kind of those plane search results.

So there's that kind of customer. DuckDuckGo has a similar thing use case where they're using Diffbot. Then there's consumer app companies, companies like Snapchat, companies like Instapaper, companies like Zola, and they are using our articles. We power the article feature

inside Snapchat. So inside the backend for Instapaper, when you Instapaper an article, it's basically sending that URL to our algorithm.

We're analyzing that article page. We're parsing it out as a clean article, and that's what gets pushed to your Kindle. There's like a wedding planning app, which is totally unexpected where if you're building a wedding registry and you just put in some product links, it basically built – It pulls in from Diffbot like the price and the picture of the product and the category of the product and all of a sudden you have a registry, like kind of a shopping cart built out from random webpages on the web.

Then enterprise experiences, I'm talking about just really common business things, like sales. I want to find more customers. I can query the Knowledge Graph, so help me identify all companies that have between 100 and 200 employees that are in, let's say, my selling geography. Help me identify who the CMO is. Help me identify if it's in a certain sector, like tech or agriculture. Essentially, you can query the Knowledge Graph and you get back all of the entities in the world that match that.

For recruiting, I want to find people with certain skills or I want to do diversity-based recruiting. I can essentially target and really specialize and personalize my outreach based on querying the Knowledge Graph just for those people with those specialized skills. We've actually hired a lot of the people at Diffbot that we've discovered through our Knowledge Graph, because we have certain skills that we have that there's only probably a dozen people in the world that have those skills.

[00:16:06] JM: Like what?

[00:16:07] MT: In the area of natural language processing, there are certain special sub-problems of natural language processing like relation extraction, entity linking, core reference resolution, these are basically people that they've done their Ph.D. thesis in that, and we've only found about it because we were looking for that particular thing in the Knowledge Graph.

[00:16:26] JM: Wow! I bet they're pretty happy to come work for you then.

[00:16:29] MT: Well, yeah. I mean, Diffbot is the best place in the world if you are working on that topic when you think about it, because you can essentially apply the fruits of your Ph.D. research to the large-scale NLP problem, apply it to the entire surface of the web, like all the documents on the web, right?

[00:16:44] JM: And help other people develop NLP applications.

[00:16:46] MT: Help us get our Knowledge Graph to be very accurate and help us make money. So it's like directly contributing to the business. So it's super impactful.

[00:16:56] JM: When a company like DuckDuckGo uses you to make those little sidebar things, this is when search for George Clooney on Google and you see like the sidebar and it's like picture of George Clooney, movies that George Clooney was in, things like that. Does DuckDuckGo eagerly fetch everything about the Internet and cache that stuff, or do they just do it on-the-fly when you make a query?

[00:17:24] MT: Yeah. There're only three companies in the United States that actually crawl the entire web one. One of them is Google. One of them is Bing, and the third one is Diffbot.

[00:17:35] JM: Wait. But the way, how do you know that?

[00:17:37] MT: It's to the best of my knowledge, just because I've been working in the search industry for quite a while. If you know the history of it, you know that Yahoo, for example, I think in like around 2010 or so, basically outsourced their crawl of the web to Bing. They're not crawling the web themselves. They're using the Bing API and then they're likely ranking and then resorting the results. Similar thing for DuckDuckGo. So I think they use the Bing API and they use probably other APIs, and then basically query those in the backend and then they blend them together and then they resurface it. They're not themselves crawling the web.

[00:18:11] JM: But Facebook, you'd assume Facebook also crawls the web.

[00:18:16] MT: Facebook, I do have some knowledge of Facebook, and I don't think they are proactively crawling the entire web in the same way that Google is. They're fetching pages very

targeted. So if you share a link in Facebook and then you see it builds like a little preview of that link, then they're fetching that particular link, but it's not a crawl. It's not like they're spidering and then going from that link to all the pages in the web.

[00:18:36] JM: You don't think so?

[00:18:38] MT: No, I don't think so.

[00:18:39] JM: You got to imagine. I mean, all the ad infrastructure they have. There's got to be some upside to them doing that. Maybe not. I guess hard to know. Anyway, sorry to distract you. Okay. Continue with – Okay. Not anybody else, or most people don't index the web. We know that. But like when people are consuming that API, like for grabbing data, do people like grab your entire Knowledge Graph or do they use it on-the-fly. That would be too expensive. Okay.

[00:19:06] MT: They would take them a couple of years to download our Knowledge Graph.

[00:19:09] JM: Okay. Right.

[00:19:11] MT: Then it'd be out of date by the time they finished downloading it, because it's refreshing all the time. It'd just be like saying, "Let's download Google." It wouldn't be that useful, because actually the utility of it is that it's continually refreshing.

What people do is there's a couple of ways people use our products. So they can pass those individual URLs. They can say, "Here's an article page. Here's a product page. Here is a person page. I want to convert this from a URL into a JSON object that is structured." Instead of just HTML, here's the article title, and author, and clean text, and the entities, and people, and organizations that are mentioned in the article, the image of the article, the caption, the comments all parsed out cleanly.

If it's a product page and here's a products price, and here's the SKU and MPN, here's the weight and the shipping cost and the megapixels, like all those various some structured information about that product, the picture. So that's one way of using it. You pass in the page directly and then you run that algorithm on that page.

The second way is you can crawl entire domains. So you can say, “I want all of the products from Home Depot, J.Crew, Macy's, Banana Republic, GAP, and then you essentially get back entire product catalog from all those sites. The third way is by querying our Knowledge Graph. So that's where it's not specific to a site or a URL, but you're just expressing what you want in the Diffbot query language. So you're saying, “I want identify all software engineers that live in San Francisco that live 5 miles from the Twitter headquarters.” So you can essentially form that query inside our DQL, the Diffbot query language, and it will get you back the set of all people that match query.

[SPONSOR MESSAGE]

[00:20:59] JM: Cruise is a San Francisco based company building a fully electric, self-driving car service. Building self-driving cars is complex involving problems up and down the stack. From hardware to software, from navigation to computer vision. We are at the beginning of the self-driving car industry and Cruise is a leading company in the space.

Join the team at Cruise by going to getcruise.com/careers. That’s G-E-T-C-R-U-I-S-E.com/careers. Cruise is a place where you can build on your existing skills while developing new skills and experiences that are pioneering the future of industry. There are opportunities for backend engineers, frontend developers, machine learning programmers and many more positions.

At Cruise you will be surrounded by talented, driven engineers all while helping make cities safer and cleaner. Apply to work at Cruise by going to getcruise.com/careers. That’s getcruise.com/careers.

[INTERVIEW CONTINUED]

[00:22:19] JM: I suppose it's worth pointing out here that you manage your own data center infrastructure as well as using some AWS infrastructure.

[00:22:26] MT: Exactly. Yeah, we have a hybrid cloud.

[00:22:28] JM: Hybrid cloud. Tell more about how you use on-prem and cloud resources.

[00:22:33] MT: Yeah. I mean, a big reason is just because of the kind of workload we have, machine learning workload and costs, really, right? One of the reasons we build our own medal is because like our standard node that builds our Knowledge Graph, it has maybe about 40 cores, like a terabyte of RAM. It has 32 4 terabyte SSD's rated together. Those are big instances that they don't offer readily like on AWS, and if they did, it would be very expensive. So we've done like that kind of cost benefit analysis, and the payback period is actually quite good.

The other reason is that we crawl the web. So when you're crawling the web, you're using a lot of inbound bandwidth and it's actually quite expensive inbound bandwidth is on AWS GCE, any of these things. Where if you just lease like your own dedicated line of data center, it's an unmetered, untapped line. So I don't think you want to crawl the web from one of these pay-as-you-go services.

However, we do use AWS for auto scaling. So that's where I mentioned one of the services that we offer were we'll actually analyze the URL or page that you send us on demand. That service, we could get like a lot of demand or a lot of requests for URLs. It could be early spiky at different periods of the day, right? People could send a huge batch of URLs and then the amount of capacity we need to handle processing those would rapidly increase at a point in time.

So we've built our systems. So we have detectors that will detect high-CPU load during those periods of time, and then we'll automatically spin up instances on the cloud services to catch that load. Then spin them back down when that spike subsides. Auto scaling I think provides this good way to handle those kind of on-demand APIs.

[00:24:30] JM: How aggressively do you replicate the database?

[00:24:34] MT: So we don't have a lot of stores to replicate the Knowledge Graph too many times, but the good thing about analyzing parts of the web is you can always re-create it basically from that page again. So there's actually – As you might imagine, there're lots of

different systems at Diffbot. There are systems for the customer crawls. There are systems for our crawls.

So they had different amounts of replication and redundancy factors. All of the things that are kind of in the critical path of the extraction and stuff all have like multiple load balancers on their frontend. So can survive any single points of failure and they're approximate amount multiple backends in a farm, right? Each of the machines in the farm is able to handle like certain kinds requests.

[00:25:16] JM: Do you have to keep this Knowledge Graph in memory, or can you keep it on disk?

[00:25:20] MT: Right now we're keeping it on disk. It's too big to fully keep it in-memory right now, but that would be great. I've been tracking like certain technologies. I don't know if you've been following like Intel, XPoint, like their crosspoint memory, right? It's basic like kind of a mix between an SSD and RAM. Basically, their incarnation, you can just put it inside a PCI-e slot. But if the price-performance is not quite there yet, but we really are hoping one day we can fit something like that into memory. So that would enable like a new class of kinds of queries that could run on it. We had to make certain tradeoffs basically in the kinds of queries that you can run on the Knowledge Graph due to the fact that it's not on memory.

[00:26:06] JM: Do you have a sense for how much? I mean, I don't know if you did the math, but you know how much more expensive it would be to keep it in memory, to access it in memory?

[00:26:13] MT: and I don't know it offhand. But, yeah, I should do that calculation again. It's been a while since I did that calculation, but it would be – Probably a couple of factors, more expensive, and then you'd have to re-architect some things to keep it loaded in memory. [inaudible 00:26:28]

[00:26:28] JM: If I query DuckDuckGo, it's going to hit the Knowledge Graph fast enough to respond. I mean, it can't be that slow, right?

[00:26:38] MT: Oh, DuckDuckGo call us in real-time. So they send us basically big batches of URLs every night and then they indexed a Diffbot response inside their index.

[00:26:49] JM: Oh! Okay, I got it. Interesting. So if I make a query for George Clooney and George Clooney is not in the cache, I guess they could just not return. They could just like not –

[00:26:59] MT: If they haven't indexed George Clooney, then they're just not going to have a result for that yet.

[00:27:04] JM: That's not the end of the world.

[00:27:05] MT: Yeah. It's basically coupled to their indexing. If they have a page and then they can have the Diffbot indexing on top of it, that indexing is happening in an offline operation that's separate from the query path.

[00:27:18] JM: Right. So I can imagine, have you thought about what kinds of applications you can and can't build with – What applications would people be enabled to build if you could offer the latency of memory rather than disk?

[00:27:33] MT: Yeah, that's really interesting. One thing that would be possible with the latency of memory is the latency of the updates would be much faster. So you could almost like update an entity in real-time much faster. Imagine a news event where – Like a couple months back. It seems that almost every day now there's some like mass shooting, right? Then you have basically this unknown – Basically a person that was relatively unknown prior to that event. Has no web presence. Suddenly their name is being mentioned everywhere. There's something associated with them, like this is a high school students, these certain attributes. It becomes like an entity from out of nowhere.

So that's the kind of thing where you can imagine an AI system like ours that's constantly reading all of the articles on the Internet could form a very complete profile of that shooter minutes from when the event actually occurred if it could be kept in memory, which could be very helpful to law enforcement and the folks that study the kind of thing.

So that kind of real-time application for Wall Street for trading and stuff, there are certain applications that are enabled if you think about like high-frequency trading and things like that. Also the kinds of queries, the complexity of the query. I mentioned earlier, we have to make these tradeoffs. So one of the tradeoffs we made is we basically – It's a graph structure, but we don't actually use one of the shelf graph databases. We end up de-normalizing the data, basically, like two edges out. So you can only execute queries that basically do joins that are like two edges out, right? So I can say, "I want people that are software engineers that work at a company that is based in San Francisco that is an AI company."

So that kind of thing is joining maybe the people and the organization tables, but it's not going like maybe four or five join zone, right? So we wouldn't be able to do that yet in our thing. But if you could keep it on RAM, then you could execute queries that are much deeper that actually are almost like traversing the graph. So you can say, "What's the shortest path between these two cities?" The query could actually try to calculate shortest path in the graph and things like that, or things computing page rank where it's really a graph algorithm.

[00:29:48] JM: On-the-fly.

[00:29:48] MT: Yeah, on-the-fly, can be enabled computing some kind of social rank or or some kind of the trusts, or relevance of certain things could be done on-the-fly.

[00:30:01] JM: In terms of the infrastructure the you actually do have, are you using stuff like Kafka and streaming framework, Spark?

[00:30:11] MT: We're keeping a pretty bare-bones right now.

[00:30:14] JM: It's Python scripts.

[00:30:16] MT: Yeah, that's really – It's Python [inaudible 00:30:19]. Yes, any dev ops person. Now, the way it's organized is we have a variety of different farms, like you mentioned. So there're farms, which is basically a set of machines that perform the same role in front of a load balancer, and there are set of farms that do like that webpage extraction. There's a set of farms that do rendering. So we actually execute each page inside of full rendering engine, which is

different than some crawlers. It's actually like kind of interacting with the page like a videogame and dumping out all the visual information out of the page, all the pixels, the RGB alpha values and stuff like that.

So we have a bank of machines that's just doing rendering and dumping out all those pixels for the machine learning algorithm. They have a bank of machines that's just doing natural language processing, that's taking in all the texts across all the different languages on the web and understanding the meaning of English, and Spanish, and Japanese, and Russian, and German and all those different languages.

We have a batch of machines that's just doing image processing. It's looking at pictures of products. Looking at pictures of cats. Looking at pictures of people and trying to figure out what's inside these images. Then we like to use nginx in front of these banks of machines, and some of them have auto scaling. Some of them are just regular sort of standard deployed Java programs. Other ones are Python programs. The crawling is C++. Then we're using things like containers and like Docker and stuff for certain of these farms to manage the images and things.

[00:31:44] JM: Did you start using Kubernetes? Any infrastructure management –

[00:31:48] MT: No. We haven't we haven't implemented Kubernetes yet, but we're always looking for ways to simplify.

[00:31:55] JM: Would it actually simplify things for you? Would it improve – Because I know there're probably many organizations who adapt Kubernetes and maybe it's like not really additive that much.

[00:32:05] MT: Yeah. It seems like there's been mixed results depending on the organization.

[00:32:08] JM: Right. That's the reality.

[00:32:10] MT: Yeah. I mean, we've looked into it from time to time. I mean, our culture is we make decisions based on the engineering and the physics. Does it actually solve a problem?

Does it simplify things? Yeah, it's all about measuring. Yeah, to see if it can actually lead to some benefit.

[00:32:26] JM: I love it. I think that's a common. It's an epidemic of just like, "Oh! This is a shiny new thing that's trendy and like jump on that." I'm a subject to that kind of trend following. Trend following is anybody, but it's like such a useful engineering reminder always to take a step back and ask, "Does this thing actually solving a problem?"

[00:32:47] MT: I mean, I think I have a bias for using tried-and-true technologies, and we want to innovate on the stuff that we're innovating on and not also kind of take on the additive risks of like other people's innovations as well.

The crawling of the web that we have, for example, that was most largely written by our VP of search, Matt Wells. He's the founder of the Google Blast search engine, and that's all in basically raw C++ code. There's no external dependencies. You type make, and it builds a single binary and all the binaries communicate to each other and it uses UDP to broadcast, to talk to each other. It doesn't use many external dependencies or fancy frameworks that all. It even has its own implementation of malloc for memory allocation. It has its own implementation of DNS.

[00:33:32] JM: I'm not advocate going that far.

[00:33:34] MT: Yeah. So you don't have to worry about other stuff breaking your stuff.

[00:33:39] JM: Is that common in domain-specific applications in implementing your own malloc?

[00:33:44] MT: It's not very common as far as I know in other groups.

[00:33:48] JM: But it served an actual purpose. It wasn't just like to show off.

[00:33:51] MT: It was so that he could actually, at a very fine-grain, determine like exactly how each bit of memory is used inside that program so the program could inspect itself. I mean, that's what you need when you're building something as complicated as like a web crawler.

[00:34:08] JM: That's so different than most of the companies I interview are built on the cloud, and the infrastructure, a lot of it is it's this composition, like composing, choosing from this buffet of AWS services and APIs that are out there. That's an awesome way of building, but it's so different than what you're describing.

[00:34:27] MT: I think the right choice depends on the kind of company you are. If you're building something that's like a consumer application where the primary risk to you as an entrepreneur is kind of the market risk of your product or service, the easiest way to get started is to use one of these cloud services and get a stack running.

If your primary value or competency that your offering is actually the infrastructure and then engineering and the accuracy of the data and things like that. Think about Google. They're not using AWS or something like that. They have their own data center, because that's their primary competency. At least that's the way I think about it.

[00:35:07] JM: In the limit, if Knowledge Graph is on Gartner, that probably means you're starting to get competitors. Are there competitors?

[00:35:16] MT: The word Knowledge Graph is becoming, yeah, like a more popularized word now. There're a set of companies that offer graph databases. That's not exactly what we are. There're a set of companies where they're like of consultants. They'll go in to your enterprise and help you get your data into a Knowledge Graph format to build like your own internal Knowledge Graph so that you can kind of reap the benefits of breaking apart those like internal data silos where you have a better coherent schema for your organization's data. So we're not in not business either. We're in the business of basically providing a Knowledge Graph, like providing the actual graph, the data itself, right?

There's not, I would say, many people that are in that business that we're in. If you think about Knowledge Graph, what most people think about is like the Google Knowledge Graph, the Bing

Knowledge Graph. There are open source Knowledge Graphs like Wikipedia, and Wikidata, and DBpedia. So you can use like Wikidata, which is basically under the umbrella of the Wikimedia organization. But Google and Bing, they don't offer their Knowledge Graphs like as a service that you can benefit from. They keep them for themselves.

[00:36:23] JM: It's kind of surprising. Are you surprised Amazon hasn't gotten to this business yet?

[00:36:26] MT: It's hard. Yeah, I mean –

[00:36:28] JM: At least you could say that about any business that Amazon hasn't gotten into yet.

[00:36:31] MT: Yeah, that's true.

[00:36:34] JM: I'm sure it's coming. It's coming.

[00:36:34] MT: It's kind of surprising. Yeah.

[00:36:36] JM: It's coming. But, yeah. I mean, your approach to pricing I bet it's going to be a good – The fact that you've kept your costs down. You're probably going to be well-positioned even if Amazon stood up something like this. You'd probably be well-position to defend yourself.

[00:36:49] MT: We have Amazon as a customer. I mean, we're happy being a vendor to them right now. There's actually multiple groups in Amazon that use Diffbot.

[00:36:56] JM: Seriously? Any case studies out there, or you can't say –

[00:37:00] MT: I mean, there's not a lot that I can talk about. I could probably speak in just kind of generalities. Amazon is kind of like a conglomerate. If you imagine, there's a team that's working on the actual marketplace where they're trying to get more products on to the marketplace, tying to onboard more merchants to the marketplace.

There is AWS, which is trying to obviously build infrastructure and get more customers to their cloud service, and they're trying to find – Their sales team is trying to find more companies that could use AWS. Then there's Alexa. They're building intelligent systems. If you think about intelligent systems, they're not really intelligent. The reason I think is because they don't have a lot of knowledge.

So they have little bits of knowledge that are hardcoded in it and then you have to speak like a robot basically to interface with it instead of being adapting to you and the way you speak and actually knowing how to answer most questions. That's a category that we're seeing a lot of uptick in as well. A lot of people are trying to build intelligent systems, and all of those systems, you can think about they don't want to build up their own Knowledge Graph. I mean, they don't want to crawl the whole web just so it can build a chat bot or an intelligence system. So we're a natural partner to like a lot of those companies.

[00:38:16] JM: That's epic. What if you offered that as like a SaaS product, like, "Hey, Diffbot." That'd be cool. What if I could just have a device in my apartment that's like I can go, "Hey, Diffbot," and like at the end of every month I get charged just a little bit based on how many queries. That'd be cool.

[00:38:34] MT: Yeah, I think that would be awesome. We're not quite there yet. I mean, we're enabling our customers to build a lot of things like that and we're getting really good at that. Primarily, we're wanting to get our dataset really good right now. We want to have every piece of public information that exists represented in our Knowledge Graph with high-accuracy and reliability.

Then once we're there, we can think about other things, like you're talking about. Then we want to make it today where it's more of like we want to serve the market of the knowledge worker where they don't need to be able to say, "Hey, Diffbot." They're maybe not a software engineer, but they're able to write SQL queries. They're able to use and spin an Excel spreadsheet, and they're able to use a Bloomberg terminal.

So they use knowledge in their day-to-day work and they're willing to learn some kind of query language in order to get that information. That's what we're building out right now, kind of the

Diffbot query language, and we want to integrate that into all of the different software platforms that people currently use.

If you are an Excel junkie, we want to connect the Knowledge Graph to Excel. If you are a salesperson, we want to connect to the Knowledge Graph to Salesforce or to your CRM. If you are someone who is like a financial analyst, we want to connect the Knowledge Graph to Tableau, to those tools that you actually use to do your data analysis.

[00:39:56] JM: We didn't talk much about this last time, but Google tried to do this, or like – I don't know if they're still trying, or maybe it's only in their backend or something. Why didn't Google build this?

[00:40:08] MT: I don't think they've exactly tried to build this business model of offering kind of knowledge as a service. I think there are certain strategic reasons why they wouldn't want to do that, because they don't want competition, basically.

[00:40:21] JM: Oh, yeah. That part.

[00:40:23] MT: Their main business is basically to monetize your attention and to serve the advertisers. So they don't kind of want to provide you directly the answer. They want to divert your traffic to the advertisers so they can get a cut of that. I think there are certain strategic reasons why you're not pursuing that business, but technology-wise, absolutely. They're tried to build automatic web extraction before in the past. They used to have a product called Frugal.

I don't know if you remember that, where it was kind like Google shopping in the previous incarnation, but it was all based on automatically generated results. The quality of that was not very good. It was much worse than products that we have in the Diffbot Knowledge Graph. It's just technically very tough, because if the quality wasn't good, they kind of scrap that product, and then now they have the current incarnation of Google shopping, it's basically an advertising product where the merchants themselves provide the structured data to Google. If you want to get your products and self-list it on Google Shopping, you provide like a feed basically. Your catalog and you provide the prices and stuff.

So when you search on Google Shopping, you see there's only like a very hand-picked list of like retailers on there that you can buy from. So it's a human curated product basically. Of course, the other thing that Google is famous for is acquiring the company Metaweb, which was basically they're the creators of the freebase Knowledge Graph.

[00:41:50] JM: Oh yeah!

[00:41:50] MT: Yeah, and that's what eventually became these knowledge panels that Google uses. The way that Metaweb is built, the freebase product, is basically not based on – It's primarily based on crowd sourced humans. So the idea was let's make something similar to Wikipedia, but where people are editing like the actual records. That's kind of what freebase is. So when they acquired that company, they basically folded that in to their knowledge panels.

So you'll see the vast majority of knowledge panels on Google are actually just Wikipedia pages. They're these head entities, celebrities and stuff like that. Most of the day-to-day entities you interact with your business partners, your vendors, your friends and family don't have knowledge panels. So I think it's very limited in business application. It's good for computer consumer search when you want to target the Taylor Swift query or the Beyoncé query, because you have those celebrities. They had Wikipedia pages. You can show a nice panel for them. They're good for sports events and things like that, but it's not so good for actual business and getting things done and building like useful applications.

After they acquired Metaweb, they essentially shuttered freebase and removed it. As is their kind of MO, they just over time are just removing functionality from their APIs. There isn't even really a full-fledged Google search API anymore that you can use as a developer.

[00:43:11] JM: Really? Maybe once Google Cloud takes off, they'll get more into the API business. The business side of things, we also discussed this a little bit last time, but you've only raised a series A, which I mean only. That's still a lot of money. Plenty businesses don't raise any money. But you raised in 2016. I'm sure you could've raised more by now. It seems like you're just not interested in moving really fast at the expense of perhaps impairing the capital structure that you've worked so hard to build. Do feel any temptation to just like raise a little bit more, run a little bit faster?

[00:43:49] MT: Yeah. We're in a climate right where it's easier to raise capital. So there's obviously that temptation there that you can always raise more money. But when I think to like the kind of companies that I see as like the really iconic companies, companies like Apple, like Microsoft, Google, they only raised a series A. They didn't a raise a series B, C, D, because they really found their business after that first trench of financing.

Then they were able to build lasting iconic companies out of it. I think a company that's building a Knowledge Graph has to build a lasting institution. It can't be – Recently we've seen a lot of companies that have raised lots of rounds of financing that ended up they're not really being a solid business there. The financing was fueling the growth.

[00:44:37] JM: Or we don't know.

[00:44:38] MT: Or we don't know, right? But at least the unit of economics of those businesses are questionable of those companies that have raised like lots and lots of money and have very high valuations, right? At least, to me, the way I want to build a company is I want to build a company that is really fundamentally has really solid technology and good unit economics that I mentioned earlier.

Our margins are really good and there's only basically electricity and bandwidth at scale that we have to worry about in terms of costs. All the people that work at Diffbot are basically trying to just make the AI better. They're not actually serving like API requests or anything like that. They're not actually what the customers are paying for to their services.

It is I think can be a really big business at scale and we want to move as fast as possible. The main bottleneck is in capital, at least not in my mind. It's actually the kind of expertise we need in order to make these AI algorithms really accurate and to scale it. We want to target a trillion entities in the future and will have to invent fundamentally new ways of storing data in order to get there. I want to get there as quickly as possible, but it's more talent-constrained rather than capital-constrained.

[00:45:52] JM: Specifically, the talent-constraint in these specific domains, these like narrow AI domains that I bet these people have offers from Google and Facebook, whatever.

[00:46:02] MT: People at Diffbot, yeah, they could work anywhere they want in the world. There's just not that many people in the world that have built like a web scale crawler, like our VP of search, right? Even the people at Google, there're thousands of people working on the crawling team. No individual person knows how to build the entire Google crawler. So we have those caliber of people that work at Diffbot. These certain areas of information extraction, natural language processing, these are research fields where we're really at the bleeding edge.

So we actually partner with about a dozen or so different academic AI research labs and give them free access to our Knowledge Graph to spur on more research to enable that Ph.D. student that's in academia trying to study knowledge fusion or entity linking to successfully stay there instead of having to join a big company and use our Knowledge Graph and create faster advancement in an area. So that's kind of where I see the constraint of how quickly we can improve this is that.

[SPONSOR MESSAGE]

[00:47:07] JM: Looking for a job is painful, and if you are in software and you have the skillset needed to get a job in technology, it can sometimes seem very strange that it takes so long to find a job that's a good fit for you.

Vettery is an online hiring marketplace to connect highly-qualified workers with top companies. Vettery keeps the quality of workers and companies on the platform high, because Vettery vets both workers and companies access is exclusive and you can apply to find a job through Vetter by going to vetter.com/sedaily. That's V-E-T-T-E-R-Y.com/sedaily.

Once you're accepted to Vettery, you have access to a modern hiring process. You can set preferences for location, experience level, salary requirements and other parameters so that you only get job opportunities that appeal to you.

No more of those recruiters sending you blind messages that say they are looking for a Java rockstar with 35 years of experience who's willing to relocate to Antarctica. We all know that there is a better way to find a job. So check out vettery.com/sedaily and get a $300 sign-up bonus if you accept a job through Vettery.

Vettery is changing the way people get hired and the way that people hire. So check outvettery.com/sedaily and get a $300 at bonus if you accept a job through Vettery. That's V-E-T-T-E-R-Y.com/sedaily.

Thank you to Vettery for being a sponsor of Software Engineering Daily.

[INTERVIEW CONTINUED]

[00:48:57] JM: It confuses me how few people know about Diffbot. I talk to developers all the time and I just don't understand why more people don't know about Diffbot.

[00:49:08] MT: We haven't invested much in marketing or things like that yet. So it amazes me that people are even able to find our service right now, our customers.

[00:49:20] JM: Wow! You're really selling it, Mike.

[00:49:23] MT: I mean it's like, "How did you get through like our onboarding process?"

[00:49:27] JM: Did you ever hear – There's apparently a Larry Page quote, like somebody asked him in the really early days of Google, "Why don't you advertise Google?" He said, "The longer it takes somebody to find out about Google, the better it'll be when people finally find it. So we actually don't care if people take a long time to find us."

[00:49:47] MT: Yeah, we're not doing too many things to artificially kind of create a lot of visibility yet with marketing dollars, but we have already plenty of great customers and partners that we work with right now that give lots of feedback. I mean, we're trying to innovate as quickly as we can. I see it really as we're supply-constrained, not demand-constrained.

[00:50:07] JM: You could hire like a really good CMO maybe. Have you thought about that?

[00:50:11] MT: Yeah. I mean, we eventually want – It's just about focus as an entrepreneur, right?

[00:50:15] JM: Totally.

[00:50:17] MT: But, no. I don't think those two are like mutually exclusive at all. So we recently hired a VP of sales at our company. Before, it was just me kind of managing our three sellers and I was able to do that maybe with about 10% or 20% of my attention. Also, I was working on some of our larger accounts as well as a seller. Now that he's joined maybe about a month or so ago, things are running a lot more smoothly there. Now, basically I'm working for him, because I'm working on certain large accounts.

[00:50:49] JM: Do you think it's advantageous to take a little bit to delegate people? Just wait and like kind of get yourself over – As the CEO, you strain yourself a little bit to understand every domain of the business. Then when you really feel exhausted, that's when you hire.

[00:51:04] MT: I feel like if you don't know anything about that certain job function, you won't know how to hire the right person yet. You won't know what to look for. You don't need to be, I don't think, world-class at that thing, but I think you should always have some experience trying to do it yourself first at least a little so that you know what you want and don't want when you're looking to hire somebody.

[00:51:24] JM: Yeah, I agree with that. I think in software, it can be really particular. Selling accounting software is not the same as selling Diffbot. So even saying like I want to hire VP of sales, you just don't know what you're looking for until you try it yourself.

[00:51:40] MT: Yeah, definitely. I mean, you're right. It's like finding a VP of sales that can understand Diffbot. They are not going to be able to explain to prospects and customers if they themselves can't understand like a pretty technical and complex product and be able to translate that to a business person. Our new VP of sales has worked before at a public

developer tools company before. So he has had experience selling like a pretty horizontal product that can be used for a lot of different purposes and like a pretty technical sell.

[00:52:13] JM: In order to get and retain the super talented AI people, you must have a like a pretty distinct culture.

[00:52:23] MT: I mean, I would say so. I mean, I think our company culture is really one where if you are that super talented AI person, your impact will be felt very directly inside the company. So if you consider like some of these other companies that have AI departments, but they're like a consumer product, the AI part of the only has some effect on the overall business. Is not a direct effect. If it's a consumer product, there's other things just like the winds of popularity and trends and the actual user interface and things like that that affect how much impact you can really have on that ultimate experience.

But if you are working at Diffbot and you'd invent a new way of representing language, or a new way of accurately taking a sentence and parsing out the subject, object, the predicate and things like that, that directly results and the Knowledge Graph tangibly becoming more accurate allowing us to gather more facts, like the actual KPIs of the Knowledge Graph, the accuracy, the comprehensiveness, the depth, the freshness. Those are the things that our customers are paying for.

So developing something new there that, not only is it a large application and you got a lot of impact, because that code you write is executed billions of times per second. You're actually resulting in us being able to collect more money, basically, because the service is now applicable to more people. So that kind of, I guess, gratification is kind of pretty unique.

I think also just like you're working for a CEO that himself is a researcher. I really am working alongside them. I'm giving them advice on how to tune their algorithms on myself developing my own machine learning algorithms. I think that's like a pretty unique culture at our all hands meeting report on the actual machine learning metrics. We care a lot about trying to increase the accuracy of each. We have about 50 or 60 different machine learning problems at Diffbot. We have a dashboard that tracks how accurate each one of those is every morning. Then if you

make some breakthrough in how language is represented, you'll probably cause 40 of those metrics go up like the next morning. So it's hugely impactful to see that.

We have a culture where also because of the caliber of people we hire, we give people a ton of autonomy. We don't actually tell people what to do when they join our company. We just give them a problem – We just identify, "These are the problems we're trying to solve," and then they tell us what to do basically, because they are the expert in that area. There they're free basically to work on any problem in the area of information extraction, because those will ultimately benefit Diffbot.

[00:55:01] JM: I can tell, you really love what you do, and I feel the same way about what I do. I can sometimes become a little too obsessed with my business. I love it. I really love spending time on it. The last year or so has really been an exercise for me in kind of like finding the balance between taking advantage of that obsession and restraining it. Have you had like make any like internal psychological tweaks to restrain yourself at all or are you trying to do that? Is it not a problem for you?

[00:55:41] MT: I mean, I think it's hard as an entrepreneur not to be thinking about you startup your business all the time basically. Even if you're not at work in the office, it's on your mind. You're thinking about it. I mean, I think you're right. You do need to strike a balance, otherwise it's really a marathon. It's not a sprint.

Especially if you have an ambitious and long-reaching mission like ours, you got to take care of your health. You got to find – You got to go out for a run. Go work out. So I think one of the main reasons startups fail is because essentially founders aren't unable to manage like their mental state, basically.

[00:56:19] JM: I think so too.

[00:56:20] MT: Mental health is really important. So you need to find ways to decompress and to keep a clear mind so that –

[00:56:25] JM: Socialize.

[00:56:26] MT: Socialize, yeah. Yeah, it's super important.

[00:56:29] JM: I mean, like the first year of Software Engineering Daily, it was like in my apartment basically the whole time Literally, just like reading. Reading like, "Oh my God! I have an interview about Cassandra tomorrow. I don't know how Cassandra works." I'm just like reading documentation and just like going slightly insane. But not good. I mean, just not good.

[00:56:48] MT: So what have you done to change that?

[00:56:50] JM: Well, I don't have to read Cassandra docs anymore. I read it once. I think it's kind of a cold start thing. I mean, I don't know. Some people, they got the skills and they can like be balanced from day one. You might need to be a little unbalanced in the early days, but I don't recommend it. I don't know.

[00:57:11] MT: I mean, I feel super lucky to have found something that I really enjoy and I wanted to write. You think about so many people don't do that. I can't complain about it.

[00:57:19] JM: Totally. No. No. It's not a complaint at all. It is just like you , aid like finding the keys for longevity. A little bit about your background, then I guess we can close off. So the patent attorney thing? Why did you learn to be a patent attorney?

[00:57:33] MT: It sort of just happened randomly. So I was in grad school. I'm trying to remember how it started. A friend of mine, basically he was a patent attorney. So he was just starting his own practice. He was moving from a big patent law firm to just his own indi practice. He just needed some help initially to write body of a patent basically, which is very technical. It's basically like writing a paper. The last part of the patent is the claims. That's where you legal training to write those claims, because claims is basically kind of like a legal programming language, very specialized way of writing it.

But the actual body of the patent is really just describing the invention. So it actually need somebody that has a technical degree try to understand it and to describe it accurately.

So I started out just writing that part. So he had some clients, and then he would pay me basically a cut of it to write out that part. Then had clients like Panasonic. So I was writing, doing their circuits. They were claiming certain patents around radio, RF patents, radio modulation algorithms and things like that. So I had to be able to understand what the engineer invented, basically transcribe it into a patent.

Then part of being a patent attorney is you have to come up with what's called the alternative embodiments of the invention. So it's not just like what they design you invention A, but then maybe invention B and C that are other ways of implementing that same idea so that a competitor kind of just work around it by doing another way.

So one of those alternative embodiments I remember that I came up with for the original circuit was actually what was used in production. It like better than like the original design.

[00:59:05] JM: Oh, no way!

[00:59:07] MT: Yeah. I really enjoyed that kind of work, because it was technology and technical and EE, and that was kind of what I studied in my bachelors. But was also writing, which I actually like. It was also writing patent claims is a lot like programming. It's like very specific. It's like you can syntax check like patent claims, because they have to – It's almost like everything that's mentioned has to have an antecedent in it. It has to follow like a very formulaic structure. I like that aspect of it.

Over time, from doing this kind of work for my friend I was like, "Well, I know enough about what's needed here and writing the patent. Why not just go take the patent bar?" So I just registered for it and then passed the bar, and so I got a registration number. So the patent law is a little bit different in that it's national. It's not state. It's not like a state bar. So it's all regulated by like Washington, D.C. instead of California. Yeah, that's how I became –

[01:00:00] JM: Is that easier than like the State Bar kind of stuff that lawyers have to take?

[01:00:04] MT: I think, historically, the pass rates are lower for the patent bar.

[01:00:07] JM: I thought lawyers had to like barricade themselves in a library just to pass a bar?

[01:00:14] MT: I mean, basically, the way you study for the patent bar is basically you read this thousand-page, several thousand-page tome called likely the MPEP, the manual of patent examining procedure, and that is basically the patent law. So I just read all the patent law.

[01:00:29] JM: Just end-to-end.

[01:00:29] MT: End-to-end. Probably took me like a full week to read it all.

[01:00:32] JM: Just once through.

[01:00:34] MT: Yeah. It took me a couple of practice tests from before. Then I went into the testing center and took it.

[01:00:40] JM: This was after Stanford? During Stanford?

[01:00:42] MT: This is during.

[01:00:43] JM: During Stanford. Okay. It's a nice side hustle. During your research?

[01:00:48] MT: Yeah, it was during. I had already taken all the coursework.

[01:00:52] JM: You were looking for a way to make money basically, right?

[01:00:55] MT: Basically, just a way to pay the bills and to fund my passion pursuing this Diffbot project.

[01:01:01] JM: I mean, that kind of sounds more fun than like taken a side job at Lockheed Martin or whatever.

[01:01:07] MT: Yeah. The nice thing is the flexibility. So if you're a patent attorney and you have to basically write a patent, I got to a point where I got pretty good at it. I even had like certain

programs and macros I wrote that could generate certain parts of patent. So I could basically pull all-nighters and crank out like an international patent over the course of like a weekend and you could make 15K, 20K from doing that. That basically covered my rent for a couple of months in the Bay Area.

[01:01:37] JM: As we close off, I think there're a lot of opportunity to build technology companies for developers, just developer infrastructure, API companies, databases, whatever. You've built one. You've built a company basically targeted at developers that is really well-positioned. Do you have any general advice for people looking to build a developer tooling company?

[01:02:02] MT: Yeah. Where to begin? I mean, first of all, I think developer tooling companies are great companies to build, because you as an entrepreneur are so much in control of your destiny, because I think that engineers and developers are pretty rational buyers. If you're technical or you understand that audience, first of all, and you understand what the bar is needed to to build a compelling technology or tool for your audience.

So that's totally under your control. So you can start out very cheaply these days building out a prototype. We did just launch on Hacker News. Get other developers to test it out. The nice thing about developer companies is you don't need salespeople at the very beginning. So developers don't generally like to talk to salespeople, and that's not how they prefer to buy. As long as you have a website and you have a specific tool that solves a need and there's documentation and it's well described and it's really compelling, then people find it. So you'll be able to build a business out of it.

[01:03:05] JM: Yeah, and the market is growing.

Okay, Mike. Okay, last question. Any updates to the long-term vision of the company or you think it's pretty much the same as it was last year?

[01:03:15] MT: The long-term vision of the company is to build the first complete map of human knowledge. I mean, last year when we launched it, we basically have the largest Knowledge Graph now that's fully generated by an AI system that's actually available for people to use. So

our focus this year is not on just continuing to grow to size, but we really want to become the most accurate, reliable and trustworthy Knowledge Graph. We want to start getting our name out there a little bit. Start actually integrating our Knowledge Graph into the actual workflows and tools that people use out there in the business and enterprise worlds.

So kind of this year and next year is all about improving the accuracy and reliability and the integration strategy, like integrating it into all these different pieces of software. Then what you can expect from us is work continually going to be adding more and more entity types to our Knowledge Graph, interconnecting those with the other entities in our graph.

So we're going to be rolling out some new types soon. I won't say which ones yet, but basically the knowledge that we have will become more and more completes. All these kinds of – At some point in the near future, any public information that's about a known entity, some fact about it, will be represented in our Knowledge Graph and programmatically accessible by an application.

[01:04:35] JM: Beautiful. Okay, Mike. Thanks for coming back on the show.

[01:04:37] MT: Yeah. It's been a pleasure to be here.

[END OF INTERVIEW]

[01:04:42] JM: As a programmer, you think an object. With MongoDB, so does your database. MongoDB is the most popular document-based database built for modern application developers and the cloud area. Millions of developers use MongoDB to power the world's most innovative products and services, from crypto currency, to online gaming, IoT and more. Try Mongo DB today with Atlas, the global cloud database service that runs on AWS, Azure and Google Cloud. Configure, deploy and connect to your database in just a few minutes. Check it out at mongodb.com/atlas. That's mongodb.com/atlas.

Thank you to MongoDB for being a sponsor of Software Engineering Daily.

[END]