EPISODE 1071

[INTRODUCTION]

[00:00:00] JM: There are many bad recipe websites. Every time I navigate to a recipe website, it feels like my browser is filling up with spyware. The page loads slowly. Everything seems broken. I can feel the 25 different JavaScript adtech tags interrupting each other. Whether I'm searching for a banana bread or a spaghetti sauce recipe, recipe sites usually make me lose my appetite. Anycart is a recipe platform that allows users to buy all the ingredients for recipe and have those ingredients delivered. It is a vertically integrated content site and delivery system. It's also beautifully designed and extremely performant. I learned about it from Zach Bloom, who works at Cloudflare, as he mentioned it as a case study in performance.

Rafael Sanches is a founder of Anycart, and he joins the show today to talk about building a recipe delivery service and the innovations in performance that were necessary to build it. If you have an idea for an episode, whether it's about a company or a project you're working on, you can go to softwaredaily.com and submit a topic. We are always looking for good ideas, and you can also support the show by becoming a subscriber and you will get ad-free episodes. You can do all that at softwaredaily.com, and thanks for listening.

[SPONSOR MESSAGE]

[00:01:26] JM: You have a devops story to tell, and GitLab would like to hear it virtually. The 2020 GitLab Commit User Conference will be held as a virtual event on Wednesday, August 26th, and the call for proposals or CFP window is open through June 1st. Whatever your devops story is, you can submit it to this CFP. Maybe you've managed a culture change or discovered the ultimate devops hack or cut your release time in half. Whatever your story is, you can share it at Commit.

Go to softwareengineeringdaily.com/commitcfp, or you can register for the event at softwareengineeringdaily.com/gitlabcommit. If you've got a great devops story, go to softwareengineeringdaily.com/commitcfp. If you are interested in attending the virtual GitLab Commit User Conference, go to softwareengineeringdaily.com/gitlabcommit.

Thanks to GitLab for being a sponsor of Software Engineering Daily.

[INTERVIEW]

[00:02:38] JM: Rafael Sanches, welcome to the show.

[00:02:41] RS: Thank you. Thank you for having me.

[00:02:43] JM: One thing I'm pleased about with your website, Anycart, is that it is a recipe website that is not terrible. There are so many bad recipe websites and I'm definitely experiencing this in the pandemic when I'm looking up a recipe for banana bread and I'm trying to open it and then it's like sending me to like a different website and then 50 popups and all kinds of problems. Why are there so many bad recipe websites?

[00:03:15] RS: I think the problem is the monetization, right? People need to make money and ads is the only way for publishers today. Actually that's the biggest reason why we started Anycart, was to allow our customers or our users to buy the recipes that they're looking for on the site. Imagine like it's part of the user experience to – As a user, you want to cook the recipe. Why not allow you to buy them in the most seamless way as possible? That's how we started the company.

[00:03:48] JM: You've done multiple recipe-based businesses. Why do you like cooking software so much?

[00:03:55] RS: I think as a software engineer, I just really want to be part of a daily life of millions of people, and I think cooking is one of those use cases. One of the top queries on Google is definitely cooking and recipes. I was just like, "Well, I could open a company that does anything in 2010," like when apps were a big thing and everybody could do an app and become a real company. Then I just listed everything that I could do, and I think the recipes and cooking was one of the biggest use cases that I could have a big impact on. I just focused on my effort and then I recruited my wife and my brother and then we started a company. That's the story.

[00:04:44] JM: Anycart is a platform where users can order all of the ingredients in a recipe. Explain how it works.

[00:04:55] RS: Yeah. As an engineer, you can already see that's not the easiest of the problems to solve, because we need to know everything about the recipe. We need to know like users – Because we also price any recipe site, we also have a Chrome extension and a marketplace for publishers. If you have a recipe site, we allow you to make your content shoppable with a marketplace. Imagine like you could write anything. You write like beans in a way that's different than the other person. We need to match that what you typed in the ingredient section with exactly all the products that are available right now in real-time on Whole Foods close to your house. But not only Whole Foods that carry 30,000 items, but also Safeway, Walmart and all any grocery retailer, right? All recipes need to be created not only for a grocery store, but any grocery store. I think that was the challenge there in the engineering part, yeah.

[00:05:59] JM: The recipes, who are they contributed by? Is it random people or is it professionals?

[00:06:05] RS: Yeah. When we started the company we're like, "Oh! We should price the entire internet." I think the entire internet should be shoppable. That was our mentality, where we wanted to do something big. We wanted to help users to just buy whatever they see on Google. Then we created a Chrome extension that would just price entire Google. We overwrote the Google UI with our extension and then we just priced every result to Google, and you could just one-click shop Google directly, right?

When you click in these sites on the internet, you would – Or extension would popup and price the recipe in any site on the web. Then we've done this, like it's insane. The technical challenge is very difficult. But then we realized the user experience was not there because a lasagna from the internet would cause like $80, right? Which is ridiculous, like it'd be way to cheaper to just buy a lasagna from a restaurant.

The reason for this that content on the internet is built for ad monetization, so it needs to look good in a photo, but the recipe itself is not made for shopping, right? Let's say if it's beautiful on the photo to have capers on top of the lasagna, then people put capers on the ingredient list.

But just like a caper, if you have to buy on Whole Foods, it would cost you $15 or $8, right? Quickly, we discovered that content on the internet is made for ad monetization, not for cooking, like for home cooking, right? Then we decided to have complete control over everything. We have end-to-end user experience where you arrive to our site, you find something to cook. We guarantee inventory delivery on your zip code and then we provide you step-by-step cooking instructions with videos. Everything we have complete control. That's our final solution, because we said, "Look, we're actually going to build content from bottom-up for making the content cheap." If the content is built using groceries from the beginning, then you know we can't put capers just because we need like to make a beautiful photo. Then we make recipes that's very reliable and use the full-size products from the grocery story. I think that's the story there. That was a massive like change in our business, because we had an app that would allow users to create videos step-by-step, user-generated, and we got very lucky. Because of that app that it was very fast for us to create these videos in-house, so that's our final solution there.

[00:08:44] JM: The experience for those who don't know, is you go to the Anycart website and all of these videos load and it's videos of people cooking stuff and you can click into one. You can see the full recipe. You can get all the ingredients for it. You can easily have those ingredients to a shopping cart and then you can order them. It's kind of amazing because it works on a bunch of different ZIP codes. How are the groceries fulfilled? When I go on Anycart and I go to fettuccine Alfredo. I add all the fettuccine Alfredo ingredients to my cart. How is that going to get fulfilled?

[00:09:20] RS: Yeah, the way we work right now is that we have a relationship of the partners that we have listed there. Now we have 9 retailers. In 2020, many, many retailers, the bigger ones like Walmart, Amazon, Safeway, Albertsons, they have their own fulfillment and delivery systems. When we started the company, the biggest problem that they had was it was not like – This is not a problem today, but when we started the company August last year, their biggest problem was that only 2% of America was shopping online. They had a demand issue. Then we just said, "You know what? We're going to solve this demand issue by connecting every recipe site in the world generating an app user experience that's like Uber Eats for cooking."

Anycart.com today, if you land there, we model the entire user experience like on Uber Eats. Because why can you go to Uber Eats and with a few clicks you have your meal in your house in less than one hour, and you can't do that when you want to cook carbonara or a lasagna or anything else, right? I think what's quite obvious for a team that the world needed a user experience like that for groceries. Then we carved out a place on that category for us.

But like fast-forward today, demand is not their biggest problems. They have too much demand because of COVID. We are trying a lot of other things to help. For example, one thing that we do is on Amazon or Safeway or any other site, today like you can't get a timeslot. We're helping our customers place order whenever a timeslot comes available. We place it for you. Things like, they were implemented like in a day. That's helps a customer to have a better user experience. Also, another thing that we do for COVID is just all these customers that they never bought anything online, like grocery specially, right? They have a lot of questions. They really need a really robust customer service and a streamlined like user experience. We're working over the clock to just like streamline our product as much as possible. Yeah, it's pretty challenging times to launch a product, but it's interesting.

[00:11:38] JM: You're saying the fulfillment system, first of all, it's completely integrated with the grocery suppliers. I mean, that's a pretty interesting timing that you can integrate directly with Safeway and H-E-B and these different places that have pretty good technology at this point. I mean, it's probably what? Four, five, six years ago when they first started really investing in the digital transformation for the grocery store, and now that has reached fruition, and you can actually build a platform on top of a grocery store, which is pretty cool. But it sounds like you're hitting fulfillment bottlenecks just because there's so much demand from grocery stores now.

[00:12:21] RS: Yes. I think that's kind of where we were, and then right now I think we don't have a demand problem. I think our product is like users love it, like our KPIs are through the roof. But at the same time I know it's not the best user experience because you like order – I don't know, bananas and you receive a cake or something else. It's just like substitution software that they run is not the best. It's probably like very outdated. We're thinking about developing all these tools for retailers to just help them out to say like, "Look, this is for free. Just take it." Because if you are doing a better job at substitutions, out of stocks, all these things, then our product work otherwise. If you go to anycart.com today and you order chicken or the chicken doesn't arrive because of out of stock, you can't make a full meal, right? The user experience would be pretty broken. We really like train software engineers just trying to solve all these problems one-by-one. I think definitely the fulfillment is the big problem, and actually we can have a massive impact. If we just dedicate a few weeks or months to solve these problems for the retailers, we can actually help them out quite a lot, because we iterate so fast right now. It's pretty interesting.

[SPONSOR MESSAGE]

[00:13:49] JM: If you listen to this show, you are probably a software engineer or a data scientist. If you want to develop skills to build machine learning models, check out Springboard. Springboard is an online education program that gives you hands-on experience with creating and deploying machine learning models into production, and every student who goes through Springboard is paired with a mentor, a machine learning expert who gives that student one-on-one mentorship support over video.

The Springboard program offers a job guarantee in its career tracks, meaning that you do not have to pay until you secure a job in machine learning. If you're curious about transitioning into machine learning, go to softwareengineeringdaily.com/springboard. Listeners can get $500 in scholarship if they use the code AI Springboard. This scholarship is for 20 students who enroll by going to softwareengineeringdaily.com/springboard and enter the code AI springboard. It takes about 10 minutes to apply. It's free and it's awarded on a first-come first-served basis. If you're interested in transitioning into machine learning, go to softwareengineeringdaily.com/springboard.

Anyone who is interested and likes the idea of building and deploying machine learning models, deep learning models, you might like Springboard. Go to softwareengineeringdaily.com/springboard, and thank you to Springboard for being a sponsor.

[INTERVIEW CONTINUED]

[00:15:26] JM: Now that we've given an overview of what Anycart does, we can talk a little bit through some of the architectural issues and the software. Again, when I land on Anycart, there a bunch of videos that load quickly and there are these quick little snippets of recipe videos. I scroll the page. It's a really fast loading set of content. Everything feels really snappy. Can you give me an overview of what happens when I land on the Anycart page?

[00:15:56] RS: I think I can start from the beginning like when we started the company. Okay. When you land on Anycart, what happens is that we redirect you to the closest service as possible to your house. We have 24 locations in the world today like as a global service. We built global-first. We don't believe like today like when we started the company, we didn't believe in like building anything for just the US, proving in the US and then going local. I don't think that should be the norm. If you build any app, it should be global first. That was our belief. We defend it and I think it will payoff for us quite a lot. Anyway, because it's global-first, it's also US. We have multiple locations in the US. These experiences are very snappy.

Right now I think you're in San Francisco, I would imagine? You're in San Francisco, you hit Oregon servers, I think. Then we have a web server there, and I think in Oregon we have also database server very close by. Not only have web servers that are in 24 locations, but we also have database servers there in at least five locations across the world. We built these web apps just like you would build an Android app or like an iPhone app. It's just isomorphic architecture using NodeJS. I think the mentality there was like, "Well, what if 100% of our web code was written in JavaScript? Then we could run that on the server with node and also on the client. We share the code between both.

Then to Google, we spit out ATLM. So it's like render really quickly on the browser as well. The user, like the first render on our pages is like in 300 milliseconds, you already get meaningful paint. We were able to do these kind of things like thinking through the user experience, right? I think the biggest – When we started the company, actually we've done like – It's like a user-generated publisher app. The idea was like, "Well, if we grow, we're going to be on Google," and Google is going to be throwing us a lot of traffic.

Well, how do we make this company work without having to think about Google? How do we make the company work in a way that it's Google's job to send us the traffic? If they don't send us the traffic, they have a problem on their algorithm because the user experience is so superior than the other competitors that like if they don't send us the traffic, they're doing a bad job at search. That's kind of our mentality. We really fought through like, "Okay. If you're in Indonesia or if you're in Italy or if you're anywhere in the world, in Brazil." It foresights so much faster than our competitors, if Google doesn't – An the content is so much better, right? Step-by-step cooking videos, it's so much better than content that's just like a photo a bunch of texts. We've done not only the step-by-step videos, but also, "Well, when you click from Google to our site, it takes like 300 milliseconds to start seeing it." If you click in any other site from the list, it's like 10 seconds, 5 seconds or something more. Our idea was like latency was really important. We really fought through like, "Okay. If we put all these node servers everywhere but the database is not close by, then you still have 300 milliseconds of speed of light for the node server to hit to our database."

Every API will still be like 300 milliseconds. We're like, "Okay. Let's put not only node servers, but also having like the database close to the users." So then we created Java localized database as well. We also replicate the database in all the Java locations. Because we've done that, it's pretty ridiculous, because think about it. An Android app, if you have a cluster database like this that's fighting against speed of light, all the inserts in the database are super slow. You have to batch everything. Because you have to batch everything, like the response to the user is never fast, like for writing in the database.

Then the same kind of code that we had to do on the server side we had to do also in the client side. When you write a comment on Anycart or on an app, we just fake it for you and then alter we write into the database. It looks immediate for the user, but actually we're batching everything also on the client. This is kind of the tradeoffs that we discovered, like when writing apps that work in multiple locations. If the database, from day one, if you develop your software like clustering the database, then everything else would follow, I think. I think that's kind of what we learned using this architecture. I think it matches really, really well with apps, modern apps. Let's say our app that users can like record. All those videos that you see on our page, 100% of them were created using our native app. We have an iPhone and Android app that allows you to record those videos exactly like those videos. So all those were recorded using our app.

Yeah, imagine uploading all these videos through the internet where it's really slow. If it wasn't batched, if it wasn't like in a sync writing of the database, you would lose data or there'll be a transaction failure. So we had to do like a bunch of retries. We had to do queues. We have to do all these things in the clients as well. It's kind of the same code, and I think slow writes and like

Java localized databases really match the new architectures for modern native apps as well. Even on the app side, you really want to like batch things, the writes. Zero latency for the user means like the data is already there, or at write time, you have to fake it a little bit I think or having a queue, or the user might not have connection. Even if these are connections fast, like he's trying to write something, he might not have connection at that point. It's the same problem, right? I think we got very lucky there in that sense, in the sense that –Let me see how to rephrase it. In the sense that because our database is so slow to write, we had to redesign even the apps from day one, right?

If you write your code from one is written in a way to accommodate slow writes, then everything else falls in place. Even in the native app side or on the – Does it make sense at all?

[00:22:32] JM: It does. I'd like to know a little bit more about the caching infrastructure or the CDN infrastructure, because you're talking about geos, and I'm wondering if you're talking about geos where you have all of your server infrastructure, if you're talking about CDN geos.

[00:22:48] RS: Yeah, exactly. We have CDNs and Cloudflare and all these things, but we actually have a master-master [inaudible 00:22:54] architecture. Our servers, they have the same configuration and all of them have the database and other things in different countries and in different locations. Actually, like to write in our database, either it writes in our servers or the write fails and then you have to retry. Does that make sense?

[00:23:16] JM: It does. The advantage of that performance-wise is what exactly?

[00:23:21] RS: Reads are really – If you're in Brazil, our time to respond to your request is going to be 30 milliseconds, right? But if you're in the US, 30 milliseconds as well for reads. Anywhere in the world is going to be 30 milliseconds maximum. But writes are going to be like 500 milliseconds or something like that, because we added the [inaudible 00:23:43] as a cluster place. Does it make sense?

Writes are really slow and they lock-in as a disaster. It's really difficult to manage writes in a geo manner. But then like reads are super-fast. Basically, we don't need caching because we know our reads are going to be local. I think that's the tradeoff there.

[00:24:04] JM: What I don't understand is why any of these matters so much. I think about Anycart, like I log in Anycart. I'm hitting some geo that's closest to me and I'm only going to be writing to the database at that geo. That's all that really matters to me. I'm just ordering groceries from a place nearby. Why do you need this instantly replicated multi-geo thing?

[00:24:25] RS: Let's say we become a huge company that we are like also in Brazil or like anywhere else in Asia and then like people start – Imagine the user experience in the world is like you search for a Lasagna on Google and that lasagna is shoppable. You can click and can buy with your like local Safeway store. It arrives to your house in less than one hour. In that scenario, you really don't want the latency. Even in the US, if you have to connect in Oregon opposed to New York and you have the 70 milliseconds extra latency, to me is pretty annoying. For example, most people in the people, they host their servers in like East Coast. The user experience in West Coast is like exponentially slower for a user at least. Yeah, I see what you're saying. Maybe it doesn't matter that much, but I think users notice it [inaudible 00:25:18] at least at read times. I think it really, really matters. Because in my mind, every 10 milliseconds, the user experience is double. Let's say if you have a server that's replying 50 milliseconds compared to a server that replies in 10 milliseconds, it's way faster like 10 milliseconds than 50, in my opinion, as like a compounding thing I think. I might be crazy.

[00:25:44] JM: No. I'm still so confused. If you're talking about writes – So maybe you could just clarify what writes are happening in the Anycart database, because when I think about the writes that are happening to an Anycart database, it's like people are creating new recipes. You're like adding new ingredients to the database. Then the writes that actually matter are just people ordering groceries from a geo close to them. What –

[00:26:1] RS: Actually if you think about like what's happening when you're browsing the UI, as the user browsers, the inventory at the start which has 30,000 different items are constantly changing as the browsers. Let's say you search for bananas and then we're not going to be checking like Amazon inventory every like 10 seconds. Rather, we check the inventory when users are browsing. Then as you browse, like say, "Oh! This banana just became out of stock," and then we updated it. There're a million updates like happening, but we batch them like crazy.

Also, we track like every page view, every click that you do, it's saved in our database. We cluster them. We normalize them. We were doing like a million writes. I think the point is like we could write them synchronously if we had a database that's really fast and it's like saved in my local data center. But because the database is so slow, we have to batch everything that we're doing, and like every five minutes do this write. I think this kind of architecture is very different, right? Because if you have a fast database to write, I think you're so busy trying to push new features through that you're just going write them super-fast. You're just going to like start like creating a lot of writes on the database rather than batching them, for example, because that's kind of everybody do.

I think the point that was trying to make is that because this was a constraint that we had like from day one, I think we got very lucky. Yes, the site looks snappy. There's a lot of frontend work to do that to make that work, but there's also a lot of backend work that needed to be done by the engineers to make this happen. I think that's kind of the idea. I think [inaudible 00:28:01] performance. Number one, I hate like cloud services like AWS or like Google Cloud. I think they're super overpriced and I just hate them, but I'm very opinionated.

[00:28:18] JM: I think this is why Zach Bloom introduced us. Zach Bloom from Cloudflare, who's been on the show a couple of times, he was like, "Hey, have you talked to anybody recently who I should talk to?" He's like, "Oh, you should definitely talk to this guy, Anycart, Rafael. He's running a bunch of bare-metal servers." I don't know, but Cloudflare is a cloud service. I mean, tell me what are you talking – Are you like renting colo space or are you like racking and stacking from your apartment building? What's going on?

[00:28:48] RS: Yeah, so the box there. I mean, for some things like – Go to eBay. This is ridiculous. Go to eBay, search for a GPU tesla. You can get a tesla GPU for 500 bucks on eBay today used, because the mining industry just blew up. Everybody is like – It's ridiculous. Just think about this. I can't mount a server, a GPU server with Tesla K80 for thousand dollars. I can buy one on eBay today and mount the server, because the GPU is 500 and I get like a motherboard of CPU for 390 on eBay. I can have a rack, like we have 10 Tesla GPUs for like $10,000. Why would you pay Amazon like $2,000 for that machine a month, right? For one machine, right? There is no sense. There is no way I'm going to do that.

I mean, we have funding. We could spend the money, but like there's no way I'm going to pay the money for – I think if we're a software company that our whole business runs like knowing that it's a low-margin business. It's going to be all about the pennies. Even if it was an app-based business or if it was – The grocery business is definitely a pennies business, like it's all about the margins.

I think I really love performance as well. So bare-metals, like I have the entire machine for me. The entire thing is like just for me. We have like NVME disks that are running like at 3 terabytes per second. It's so much faster. Nobody is going to convince me that the cloud is better, and the reason for that that we run a flat architecture. We just like boot up a new machine and then it download the configuration, the database, the files that it needs to download. Then as soon as it finished downloading, it puts itself in the cluster and then it runs, right?

I think the point is like to create this architecture, even if I used AWS or Google Cloud or Microsoft Azure, like I would have to do all that work by myself like anyway. They're not helping me distributing my workload anyway. I think the question is like why wouldn't you pay the cheapest provider per raw performance if that's the case, I think. That's the question.

[00:31:13] JM: You're not mentioning the operational expense.

[00:31:16] RS: I mean, it's just me, like it's one guy. Literally like – I've done this all by myself. There's nobody. Actually, nobody for this are behind the servers and setting up the machines. I'm the only one – I'm the CEO of the company today. Like I'm the only one doing this, like setting up the machines and like creating a new machine. I Mean, the machines are so powerful that like we rarely have to add a new machine anyway. One example, we pay $180 to have a machine in Hillsborough in Oregon. This machine has 256 gigs of RAM. It has two terabytes of NVME disk and it has 24 CPUs that are multithread. We have 48 virtual cores, whatever. $185 for that machine, right?

I don't know of any service that would need like more than one of these machines. If you cash everything, like if it's a Java server or like a NodeJS server, you really need to have a lot of traffic to need more than one of these. Anyway, we have like five of these or ten of these. I think now we have a lot of these, but like we don't have a lot of servers because of traffic. We have it for like replication and like stability of the servers and things like that, because just adding one more of these machine is so much more. It's so much processing that you really don't have any maintenance, I think. That's kind of my take on this one.

[SPONSOR MESSAGE]

[00:32:56] JM: When I'm building a new product, G2i is the company that I call on to help me find a developer who can build the first version of my product. G2i is a hiring platform run by engineers that matches you with React, React Native, GraphQL and mobile engineers who you can trust. Whether you are a new company building your first product, like me, or an established company that wants additional engineering help, G2i has the talent that you need to accomplish your goals.

Go to softwareengineeringdaily.com/g2i to learn more about what G2i has to offer. We've also done several shows with the people who run G2i, Gabe Greenberg, and the rest of his team. These are engineers who know about the React ecosystem, about the mobile ecosystem, about GraphQL, React Native. They know their stuff and they run a great organization.

In my personal experience, G2i has linked me up with experienced engineers that can fit my budget, and the G2i staff are friendly and easy to work with. They know how product development works. They can help you find the perfect engineer for your stack, and you can go to softwareengineeringdaily.com/g2i to learn more about G2i.

Thank you to G2i for being a great supporter of Software Engineering Daily both as listeners and also as people who have contributed code that have helped me out in my projects. So if you want to get some additional help for your engineering projects, go to softwareengineeringdaily.com/g2i.

[INTERVIEW CONTINUED]

[00:34:45] JM: Not to put a knife on your back, but I did go to the website yesterday and it was down.

[00:34:53] RS: Yeah, that's the release. We're releasing like crazy. We haven't implemented anything for like whenever –

[00:35:01] JM: Planned maintenance.

[00:35:02] RS: Yeah. We just deployed like crazy. We have like a Slack script that allows anyone in the company to deploy it anytime. We just deploying like crazy.

[00:35:11] JM: That's very funny.

[00:35:11] RS: Yeah. Whenever it deploys, it stays like 5 minutes down or something, huge.

[00:35:17] JM: Okay. Does that mean you're managing your own databases?

[00:35:21] RS: Yeah.

[00:35:22] JM: What database are you using?

[00:35:24] RS: We're using MariaDB. What's the name of the – Galera Cluster? For clustering in multiple locations.

[00:35:32] JM: Okay.

[00:35:33] RS: I mean, it's not perfect – We've done this three years ago. Three years ago, it was the best thing that we could find. I think today there are many other better solutions probably. I wouldn't recommend this for the technology or anything else. I think everything that we've done was not like, "Oh, we need to run like a geo-localized database because it's so much fun to learn how to do that." Of course I don't want to be managing databases and things like that, but everything that we decided was like, "Oh! It has to be close to the user because the user experience are going to be so much better." We're going to build a global product. People in Brazil, like if it's like 180 milliseconds faster for them, it's a lot faster. It's like so much faster. Then we're going to have a lot of users anywhere in the world. So like we're just like reverse engineer, "Okay, what's the best architecture to fit in that use case?" We're like, "We want to build a global project that's fast for anybody in any location." That's how we fought like, "Okay. So putting database servers everywhere is very expensive because you need like better machines." Ideally, you don't want to like your MySQL MariaDB server to be running the same machines that you have like web traffic. You need at least like one database machine plus one Java machine and then you like a really tiny like NodeJS machine separate for every location.

Okay. So let's have like five of these databases everywhere plus like 20 or 30 NodeJS, like tiny machines running a lot of traffic. It's hilarious, we have this – In one of the countries, we have 9 million users every month that go there and there's one machine, like one NodeJS machine that has one CPU and two gigabytes of RAM like running all these traffic. It's crazy.

[00:37:23] JM: By the way, and how are you setting up the servers in the other geos?

[00:37:28] RS: Right now we use the cheapest available service. Sometimes it's AWS, like Amazon. Sometimes it's Google. For example, in Brazil, we prefer Google. In Japan, we use AWS. Depending what's the cheapest in the location, we take it. It doesn't really matter for us who's the machine from. Usually like when it's a small instance, it's cheaper to use a cloud. If it's like a tiny machine, like two gigabytes of RAM and plus one CPU, then we use cloud for that. But it's like $10 per month machine, right? Then it's like super funny that I get all these traffic in this tiny machine, but the machine is only doing IO. It's not doing anything like on the disk. It's not doing anything. It's just like internet traffic passing through. That one is okay like to use cloud. We just use the cheapest available, right? We have like five different cloud that we use. Also, for the bare-metals, like we have different – like in Europe, we really like the Hetzner for example in Europe. It's really good.

Then in the US, there's OEH, is doing a really good job as well. We pick the one that has best pricing per performance. I think in the future, if we continued this, we just need to hire somebody to do exactly what we are already doing for our self. Basically, arbitraging. Let's say Amazon gives you a better cost per server than Google in Los Angeles, then automatically switching the server to Google if Google is providing better price.

I think that's kind of the future where there's going to be somebody in the middle like reverse bidding and then sending the traffic to whoever is cheaper. Basically, I'll build that company, right? I'll totally do that in the future, because that's what we are doing like kind of dynamically already. If we just like allowed, provided that as a service to people, that would be pretty interesting.

[00:39:23] JM: There's your AWS.

[00:39:25] RS: Yeah, but that's the problem, right? They cannot do that themselves because it's going to be a race to the bottom for them, right? Whoever –

[00:39:30] JM: No. I mean there's your AWS. You already got the retail side. You just need your AWS business and now you completed the story.

[00:39:38] RS: Yeah, we're proxy for the retailers. We're like proxy for retailers. Then from –

[00:39:43] JM: Yeah.

[00:39:44] RS: For the service, Anycart.

[00:39:48] JM: It must be crazy scaling up in this time when online shopping is just going bananas. What's that been like? Are you sleeping?

[00:39:58] RS: Yeah. For me, it's not about – Again, like having those bare-metals really give me a peace of mind because I know there's no traffic that we cannot like handle right now, because like if we have like a bunch of like super powerful machines, it's really impossible to get all the traffic. But I do – We're losing like sleep right now, like customer service with just like all these retailers. They're having like scaling problems. Maybe they use the cloud. I don't know.

Yeah, the scaling problem is definitely not – Line engineering-wise, it's not a problem. It's definitely a problem of like customer service. How do you handle like all these users who have like bad user experience? They got a bad substitution or they order their groceries and they order like 50 items and only a banana show up in your door. Literally, there was a person who carried the bag of only a banana and delivered to your house. Imagine the frustration of the users is like what keeps me at night these days and we're like trying to so hard to solve it. It's very difficult. Definitely, that's our biggest problem right now scaling that up.

I mean, we're just tiny startups. We're just a tiny startup learning all these problem that's like such a huge problem, right? It's a national problem. We're learning very fast. Applying a lot of technology to solve the problem, and I think we're going to come up with something very unique very soon. I mean, we launched two weeks ago.

[00:41:31] JM: The shoppers, those are provided by Safeway or H-E-B or whatever, or do you have to get your own shoppers?

[00:41:38] RS: Yes, they have their own shoppers. All the major retailers have their own pick and packing and delivery systems. I think that's the idea. We focus – The entirety of our business was like let's figure out a way to solve the demand. We're like, "Okay, yeah, let's do the best user experience for recipes." So people are more interested in buying online, because the supply side was already there, right? We don't need to hire 300,000 people to drive to the store, pick something from the shelf that was just put in the shelf. That makes no sense.

What's happening right now in the industry, so we are a little frustrated by that, I mean it is what it is. I think people are trying their best. But definitely what you're saying is correct. Because Amazon has their own fulfillment and delivery, Safeway has their own, Walmart has their own. I think more and more retailers, at least the big ones, will continue having that.

[00:42:34] JM: Because Zach from Cloudflare introduced us, I was just wondering, are you doing anything interesting at the edge, or do you have any edge workers or edge storage stuff that would might be interesting?

[00:42:46] RS: Yeah. All these thing that I told you about, like we have like 24 or 30 different – I think we have like 30 different locations across the world that do like our web app. Our web app is just that could run on my computer like a little NodeJS thing and my computer will start processing prediction traffic if we just open our an Argo Tunnel, right?

Let's say you open Argo Tunnel. The machine puts itself in the cluster and already starts processing traffic from San Francisco. I think that's a very – We implemented this like three years ago. There was no like edge workers on Cloudflare for example, but it's very similar. The difference is like we have complete control. We put a varnish instance. We have like reverse proxying those instances. We have NodeJS. We have all these things running on the machine. We have like full control.

But now that Cloudflare has the edge workers, we might – Because we have an isomorphic web app. We might just get the bundle that we send to the browser then we're going to try to adapt that bundle to the workers. Then if we could run our code on a Cloudflare workers, it would be even faster, right? We're going to get some 10 milliseconds out of the equation. That'd be so cool.

[00:44:10] JM: Yeah, you really are a performance nut.

[00:44:12] RS: Yeah. I try to say the user experience, but it's difficult to – It's kind of like an obsession. There is nothing – After some point, there's nothing else you can do for your users, not just like destroying the latency, right? At some point, like the user experience is so streamlined that the only thing you can do to differentiate your business is latency I think.

I think using latency as a part of the user experience, as a feature of the service is being in the culture of the company since we started. I think that's a big differentiation, because once you start getting the organic traffic like from Google or anybody else, it's basically very hard for your competitors to copy you right? Because like I said, who is going to like migrate their database to a geo-localized database once their service is up for 10 years? There's no way. There's no way.

If you don't do it from day one, it's going to be really, really difficult. I think you can put a million CDN in front or service is always going to be faster. Let's say you started a recipe site tomorrow or like Anycart's competitor or something, like if you don't do the geo-localize, then our service is always going to be faster, the faster, snappier. Because then like even if it's like you put your servers on Amazon, like East Coast or something like that, it's going to be faster for East Coast, but we're going to be so much faster in West Coast, because we can put it very close to the users. I think that's kind of the mentality there, performance net.

[00:45:51] JM: Okay, last question. What's the hardest engineering problem you're working on right now? What's really giving you a lot of trouble?

[00:45:58] RS: I showed you the GPU server. We want to be best search engine in the world for products. Let's say you search – If you go right now on the site, try to search pizza. If you search pizza, we give you frozen pizza, then we'll give all the meals that you can do with pizza. Then we give you flour, we give you tomato sauce, we give you cheese and then we give other stuff. Imagine like that user experience. Like you have an intent, which is pizza, but we give you like the best search engine possible. We just started working on this. We were so busy to just like launch the service.

That then now we launched, to me it's just like when you have the most competitive sports in the world, it might be like Formula-1, or like you need a car. To just start the competition, you need at least a car. But then after you start competing, you start like tuning your car. I think search engine is like where we're going to be best in the world. They only data that we have for every product is just the title of the product on Amazon or on Safeway or all the retailers. We don't have any other data. What we started doing now, we get all these products, we process their photos. We extract the labels of the text in the photos. We're able to do like some GPU processing to extract the texts. Let's say you search now on any card, you're going to search like frozen – I don't know, something, frozen tacos. We detect that frozen wasn't the label of the product. Then we started indexing our search and just making a really good search. I think that's the most fun. I didn't say challenging. It's the most fun. I'm having so much fun with these kind of things. It's so much fun. Just like installing the GPU servers so it can do this at scale and this thing is really fun for me.

But I do believe that as a company mission, if we're the best place for search of product, if you have an intent to come to us, if you need to buy groceries or any kind of, I think that's a very good place to be and being very innovative and like, "Okay, we can't extract more data than we have already in the title. Let's pick from the photo. Let's do – We do have a lot of work with the photos, because buying groceries online is pretty frustrating user experience, because when you go the store, it's so inspiring visually. You're like in this place. You see an avocado, it's beautiful, you could touch it. It's so good. But when you go online, there's like this white background photo and it's ridiculous. There's no sense.

We've done a lot of code to like – We remove all the background. We're trying to put on a black background or like changing the background, do all these things. Yeah, it's such a simple thing, right? If you compound on the user experience, like not only – There's all these things that we're doing, to process the images to make like better. I think the culture of the company is always that, right? Okay, there is a problem, it's the images. Let's go very deep. Let's extract all the problems. What can we do? Do I have to put all these GPU service in my garage? Yes, sure. Let's do it. We don't care. It's one more thing we need to do. I think going very deep technically is like I think where we can really improve the user experience on groceries online. I think that's our goal.

[00:49:21] JM: Okay. Rafael, thanks for coming on the show. It's been great talking.

[00:49:24] RS: Thank you so much. Have a good day.

[END OF INTERVIEW]

[00:49:34] JM: Today's show is sponsored by Datadog, a monitoring and analytics platform that integrates with more than 250 technologies, including AWS, Kubernetes and Lambda. Datadog unites metrics, traces and logs in one platform so that you can get full visibility into your infrastructure and your applications. Check out new features like trace search and analytics for rapid insights into high-cardinality data, and Watchdog, an auto detection engine that alerts you to performance anomalies across your applications.

Datadog makes it easy for teams to monitor every layer or their stack in one place, but don't take our word for it, you can start a free trial today and Datadog will send you a T-shirt for free at softwareengineeringdaily.com/datadog. To get that T-shirt and your free Datadog trial, go to software engineeringdaily.com/datadog.

[END]