

EPISODE 688

[INTRODUCTION]

[0:00:00.3] JM: Airbnb began in 2008 as a monolithic rails application, serving the simple purpose of listing homes for rental. Over time, the number of listings increased dramatically, as did the number of people who were renting. With that scale, the Airbnb rails application had to be broken into different services and entire teams were built out to focus on challenges, such as pricing, application infrastructure and search.

Surabhi Gupta has worked at Airbnb for the past five years. She joined in 2013 to work on the search team and has worked on different teams at Airbnb overtime. Today, she is a Director of Engineering leading the homes business for Airbnb, which includes growth, search, hosts, pricing and business travel. Surabhi is in charge of a lot of different areas of Airbnb.

Surabhi has helped scale Airbnb through a hyper growth period and joins the show to share those experiences. One distinct area that we spent time on was Airbnb's search engine. Surabhi formerly worked at Google and she described how the engineering problem of a search engine for homes differs from a general-purpose search engine like Google's.

Before we start the show today, I want to mention that we're looking for sponsors for Q4. We have around 50,000 engineers that listen to Software Engineering Daily on a regular basis. If you are interested in reaching that volume of users, we are probably cost-efficient option to advertising compared to certain other outlets, like perhaps conferences. Conferences are quite expensive to sponsor, but Software Engineering Daily you can reach more engineers than a conference and probably going to be cheaper.

We're also hiring. You can go to softwareengineeringdaily.com/jobs and look at our openings for roles such as writers and podcaster. We have very high standards for the podcaster that we are hiring, but we are looking for a podcaster. We're also looking for a few engineering roles, so please check out softwareengineeringdaily.com/jobs.

[SPONSOR MESSAGE]

[0:02:11.8] JM: Accenture is hiring software engineers and architects skilled in modern cloud native tech. If you're looking for a job, check out softwareengineeringdaily.com/accenture. That's softwareengineeringdaily.com/accenture. Working with over 90% of the Fortune 100 companies, Accenture is creating innovative cutting-edge applications for the cloud. They are the number one integrator for Amazon Web Services, Microsoft Azure, Google cloud platform and more.

Accenture innovators come from diverse backgrounds and cultures, and they work together to solve clients' most challenging problems. Accenture is committed to developing talent, empowering you with leading-edge technology and providing exceptional support to shape your future and work with a global collective that's shaping the future of technology.

Accenture's Technology Academy established with MIT is just one example of how they will equip you with the latest tech skills. That's why they've been recognized on Fortune 100's best companies to work for list for 10 consecutive years.

Grow your career while continuously learning, creating and applying new cloud solutions now. Apply for a job today by going to softwareengineeringdaily.com/accenture. That's softwareengineeringdaily.com/accenture.

[INTERVIEW]

[0:03:43.3] JM: Surabhi Gupta, you are a Director of Engineering at Airbnb. Welcome to Software Engineering Daily.

[0:03:47.8] SG: Thank you for having me.

[0:03:50.4] JM: You are a Director of Engineering and you've been at Airbnb for more than five years, how did you get started at Airbnb?

[0:03:59.6] SG: Sure. Maybe I can start with a little bit actually what I was doing before that. I spent six years at Google and I worked on search and I worked with the search ranking team and then on a project that became part of Google now. I was always fascinated by travel and

data. I actually had made a booking, this was earlier in 2013 with my family, we'd gone to Venice and we just had this really great listing and then I got introduced to Airbnb. That's what got me interested in the company. Just as I spoke to more people and I heard about the vision, it was very exciting it was something that I wanted to be a part of.

[0:04:39.4] JM: When you look at the search challenges of Airbnb, what's unique in the set of challenges in doing searching on Airbnb?

[0:04:47.8] SG: Yeah. It's a good question. I remember early on, I used to talk to a lot of candidates that would say, "Okay, you work on search at Google, or one of these other companies," and how can it be that different? Actually, I like to think of it as this traditional search where you express what you're looking for. You expect the most relevant result right at the top, right? If you go to Google, you don't want to click on the second or third result. You just want, expect the top result at the first position. If it's something that you're not happy with, you'll reformulate what you're looking for.

If you think of marketplace searches, it's typically based on what's available, right? What's the available supply and you want to match supply and demand. I would say that Airbnb adds another layer to this, which is that both sides of the marketplace have preferences. What this means is that a guest comes in, let's say I want to travel. I was on that trip to Venice. I had certain expectations of that trip where we wanted to stay, how many people were there, it was my family, so that meant that we wanted to stay near where there was good food available and maybe not too loud.

Then the host also has preferences. The host also might say that, "Hey, actually my place is good for these types of guests, or maybe guests that are traveling." It's actually great for families. For another place, they might have a lot of stairs inside and might not actually be perfect. This is what makes – fundamentally what makes Airbnb search very complex, which is that the preferences that a guest has are very complex, and then matching those with what is right for the host, that's where the magic comes in.

I think a lot of people assume that okay, if you've worked on traditional search where you're optimizing for that first click it translates over to our marketplace. I would say that that is also

another layer of change, where for Airbnb we are optimizing for that booking ,right, and that process of searching, figuring out what's right for you, figuring out the inventory, and then making the final booking. That entire process can take multiple days. Typically, you think of – or people are not used to thinking of search as this long process that you find you want to get the booking conversion.

[0:07:11.3] JM: Naively, somebody who's thinking about search at Airbnb might say, “Well, it's just a matter of star ratings and is it a positive or negative review.” You mix that with a geolocation search and voila, you've got a search engine. Actually, it's much different because in Google search, you are not supply constrained. There's no problem with how many people click on a webpage. Unless, that web page was behind some limited bandwidth connection, or didn't have a CDN, or something like that. With Airbnb, you are actually supply constrained, and so you can't give the same listing to multiple people on the same day.

[0:07:54.6] SG: Yeah, absolutely. I mean, once you've booked, that place is gone, right? We've invested a lot in helping hosts how to manage that calendar. We have a lot of pricing tools available, just helping hosts be more successful. Yeah. I mean, all of that absolutely adds complexity to the search process.

[0:08:13.4] JM: You've got these different sources of data that can contribute to a good search ranking. How often does that index get updated, or built, or in the year or so that you were working on the search team, how did that search indexing infrastructure change?

[0:08:33.1] SG: Yeah. One thing is that for us, it's incredibly important to have real-time search, which means that if somebody makes a booking, you want to make sure that your index immediately gets updated, to make sure that you only have listings that are available, right? You don't want to find this great listing and then when you're trying to book it it's like, “Oh. Oops, sorry. Actually the dates just got booked, right?” That's not a good experience.

I would say that it's super important for it to be real-time. That is just been baked in as one of the key constraints and has always been part of our indexing system. I would say the big change that happened in our architecture was probably two years ago, where we just had to scale our systems. We're fortunate where our product is every transaction is worth a lot, but it's not that

we have hundreds of billions of documents, right? Each listing as you could consider it as a document. We have to go, when we built out our sharded search infrastructure, that was a big change to how we did indexing and how we rearchitected, rearchitected that search.

[0:09:38.1] JM: Were you using Elasticsearch?

[0:09:39.7] SG: No. We won't. We use Lucene. We built on top of Lucene. What I would say the complexity for us is that we have a lot of complex filters that we need to apply. If you just look at pricing and availability, we offer hosts many different rules to do each of those. With pricing, you might say that – or actually, let's talk about availability. You might say that hey, I actually don't want back-to-back bookings. I don't want more than 10 days of the month booked. We give you the ability to set all of these rules, but then that means that the indexing system needs to keep track of all of that. I think as this got more and more complex, we just have to build our own custom filtering logic.

[0:10:22.3] JM: It sounds like there's – if you compare Airbnb to something like Twitter, you want your search index to be strongly consistent. Although, that consistency might be easier to achieve than something in Twitter, because the volume of transactions is lower.

[0:10:39.5] SG: Yeah. Our volume is significantly lower than Twitter. For us, it's about complexity of the data.

[0:10:46.0] JM: What do you mean by that?

[0:10:47.1] SG: Just the example I was giving you with the calendar rules, or the pricing rules, right? You have a listing that is available for a certain number of days in the year and if there's a booking, then it changes the makeup of that calendar, right? Because somebody might have a rule, what I mentioned earlier which was you don't want back-to-back reservations. Then that changes the makeup of your calendar.

It has to be dynamically down. You have to keep re-computing this. I would say that that – the way I think of it is it's fewer documents, or fewer listings, but each listing has many different instances, right? Because it's available for many different days and that can change.

[0:11:28.9] JM: Now how far does a rule-based ranking system take you? Because I think of Airbnb as a fairly new domain, even today after it's been in the market for a while. It seems like you would – there's trade-off between how much machine learning you would want to do, because you want to keep it explainable, you want the search for infrastructure to be understandable so it's easier to study, but maybe there are good ways of introducing machine learning in a controlled way that keep the search index construction model understandable. Did search start to involve machine learning when you were working on it? Was it just a straightforward rule-based ranking system?

[0:12:08.6] SG: I mean, when I first started it was a rule-based system. Then we moved to a machine learning model. I think it was sometime in 2015. Part of it is that because the ranking system actually is trying to do this two-sided marketplace matching and it's also trying to take into account not just bookings, but cancellations, rejections, can you book it instantly, what is the likelihood of a five-star trip? It's a very complex interaction model. With the rule-based system, I think can only take you so far because of that complexity.

What we found is that in 2015 when we move to ranking to a machine learning model, we actually have significant gains. I think our ranking team has been the most predictable team in terms of driving business impact, just continue to do.

[0:12:58.9] JM: How does that marketplace matching work? Because you don't know in advance where I'm going to search, so you can't – I mean, I'm assuming that for me, Airbnb doesn't have, or maybe you do have a global ranking that gets generated, that you know in advance and you have it pre-cached, there's a ranking for all of the listings that Jeff could consider and it's pre-calculated in order of how much I'm going to like it.

I'm assuming there's some set of characteristics that you can pre-calculate for me and then some set of characteristics you can pre-calculate for all of the listings on a regular basis. Then the real-time is when I'm searching, it's more of the matching. Maybe it's you search within a geo and then you take that geo and then do a marketplace matching system within the geo. Is that accurate?

[0:13:52.8] SG: I mean, I think you're asking – there's always a combination of features that you can compute offline and things that you do online, right? Yes and I think that part of what you're also asking is how much personalization do we do. What we found, for example is as somebody is narrowing in on their search, you can see that the number of times somebody has viewed one listing becomes very predictive of the final booking, right? Because if you like a listing, you're going to keep coming back to it. This is what I think of a short-term personalization, which is trying to take into account your – which is what you've liked in that session, or in a few sessions.

[SPONSOR MESSAGE]

[0:14:40.9] 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 in 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 of their stack in one place. 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 softwareengineeringdaily.com/datadog.

[INTERVIEW CONTINUED]

[0:15:46.9] JM: More contextually, the general infrastructure challenges at Airbnb I think were mostly being solved by a team that was building infrastructure on Mesos. You were coming from Google where you had all the infrastructure was really well laid out. Google have had plenty of years to build this infrastructure layer where people could easily provision services and there's all these rich – it's like a platform as a service on steroids at Google as I understand. At Airbnb, they were probably still in the process of rolling that platform as a service experience internally.

How did the infrastructure accessibility and the experience of spinning up infrastructure, or just dealing with infrastructure, how did it compare between Airbnb in those early days versus where you were coming from at Google?

[0:16:36.1] SG: I mean, it was a very big difference. I think what's actually interesting to talk about is the evolution that we have gone through in our infrastructure here. I would say that every year feels like a big change, but definitely over the last two years we've changed our architecture quite a bit. I can share a little bit more about that, because I think that it's an interesting transformation that we've had to go through.

We historically have been in a monolithic Ruby on Rails application. We had a few services that were built outside of this monolith search and pricing just because of the scale. Otherwise, it was largely a monolith. What we found over the years was that there was just so much business logic coupled with all of the front-end logic and talking to the databases. There was just all of this logic inside of the monolith, that code ownership became very unclear, which meant that things like performance, bugs, we just started to have all of those other problems.

What we realized was that we really needed to start evolving our infrastructure. About two and a half years ago, we started this team called core services. I would say that was step one in this journey, which was we wanted to make sure that the core data that everybody wanted to work with, so that is what is a listing? How do you store a listing? How do I add a field to a listing? That was its own service and the service owned its own data. It wasn't a shared database that had information about listings and reviews and reservations and user and all of that. There was a database for all of the listing data.

We had this project where we started pulling out components like this and made sure that all reads and writes went through these services. We did this for listing, we did this for pricing availability, reservation, user reviews. We're still fully transitioning into it, but this was a massive effort that I think was the foundation for some of the other work that we wanted to do, which was last year we said, "Okay, we've done the core services, but we actually want to pull out all of our presentation layers as well."

Essentially, we would make sure that our overall systems weren't – or we weren't really going through our monolithic app for most of the critical flows. We actually made this an engineering-wide mandate. We got every team really – it wasn't just that you could say one team could have done this alone. I mean, every team had responsibilities and everybody had to work together.

There's a lot of companies that go through this transition and I think that we definitely have been on this journey as well. I think it's a really positive change. I'm glad that we started executing on this when we did. Then within my team, I run engineering for homes. I actually have had a central team that helped out all the other product teams and making sure that they were building out these services, the dependent services with the right architecture that the consistent standards were being followed, because I think what you want is you don't want to trade one set of problems with another. You want to make sure that such a big change is coordinated.

[0:19:53.0] JM: What kinds of standardization was being introduced across the company, between those different services?

[0:19:59.6] SG: Yeah. I mean, the standardization is even just how do you do authentication, right? This one person should be building an authentication service. You don't need multiple people doing that. Even just having that as a first step, the high-level architecture, what are the common pieces that everyone needs and let's make sure one team is tasked with this. I would say that's a very important step.

The other is when you're writing a service, what do you use for to define the APIs? What do you use for inter-service communication? How do you structure the code within the service? I mean, there's a lot of these decisions that need to be made. What language do you use, right? You want to make sure that all of this can be managed.

[0:20:41.5] JM: There's somebody out there that is working at a bank, or an insurance company and they're going through this process of how to break up their monolith into some services and some front-end business logic that's going to make things easier to work with going forward. They're in this constant push and pull between how do we communicate with all the different stakeholders and how do we continue to iterate on new features that the business side is

demanding, while we're trying to do this refactoring? What general advice would you give to people who are trying to go through that migration?

[0:21:20.3] SG: I would say that this can't just be a side project. Sometimes I hear people ask questions of okay, how do you make sure that this becomes a company-wide priority? Can we just set aside couple of engineers to do this? I would say the answer is no. You have to make a very conscious decision on whether you want to make this change. If you do, you have to be all-in, because everybody's workflow will change in this new world. Even the way you think of your developer infrastructure, right? If you build it for a monolith, it's actually very different from having many different services.

You want to make sure that there is a clear mandate, because otherwise, the effort just won't be successful. Then people will say, "Oh, we can't do this because it's not going to be successful." The project can absolutely be successful, but you have to make sure that you're giving it the space. Frankly, you also have to get people outside of engineering convinced, right? Because if you're going to be allocating a significant portion of your engineering to make this change, everybody else has to understand why other parts of development are going to slow down.

[0:22:27.5] JM: Was there a prioritization of which services got created first? When you were trying to break this into a microservices, or service oriented thing, was it like, let's start with something that's really non-business critical, or was it the whole organization all of a sudden, or not – maybe not all of a sudden, but was just gradually like, we're going to be moving into services. Let's reorg. Let's focus on this. How did that mandate overall strategy, the high-level strategy manifest?

[0:22:57.8] SG: There were two engineers that came together and did a pilot for one of the core pages, because I think that you want to make sure that the new proposed structure can work for critical pages, right? If you do this for non-business critical pages, maybe it doesn't have the scale. Maybe it just might not be the right test.

[0:23:17.5] JM: Wait. You started with something that was mission critical, interesting. Just stay on that point for one second. I find that really interesting, because that completely contrasts with the early Netflix story of moving to microservices and they're like, "Let's start with a job board,

because it's very simple and it's not mission-critical and it'll be comfortable to do it." Instead, you started with the uncomfortable part.

[0:23:37.5] SG: I think it depends on what you're trying to validate, right? We knew that we could build services. This wasn't the first time we were building a service. That wasn't a question for us. It was more about, can we pull a presentation layer out? That has a lot of problems in it right now with code ownership, with complexity. That's why we wanted to make sure that before we get the whole organization on this journey, we wanted to make sure that this could work for a critical page.

I think it was, we almost had to make the change, because it was becoming difficult for engineers to be able to push code to production. When you have hundreds of engineers working on one monolith, it actually – it just slows people down. I think we knew that we wanted to make this work. I would say that as we embarked on this, I think the team had spent maybe four or five months and it was pretty clear that this was going to head in the direction of being successful.

At that point we said, "Okay, let's set a goal for 2018." This was I think in September of last year, of the prior year. We said, "All right." We didn't say that 100% of all code has to move out of the monolith. We actually just said, "Let's pick these four core pages on the guest flow and two on the host flow," right? We scoped it and we said, let's get all of the dependencies implemented as part of this. We did scope it out in that way. Then right now, we're trying to figure out, okay should we – what should we do about the longtail? Those are the conversations we're having now.

[0:25:12.6] JM: Wow. Okay, so when you said we want to make sure we get all the dependencies covered, you're talking about the databases. We want to make sure we're hitting the host database, we're hitting the guest database, we're hitting the homes database.

[0:25:27.5] SG: The way we think of services is there's the data layer, there's the mid-tier layer and the presentation services. The data services are actually the ones that talk to the database. They own their own data. We had moved those out as part of the core services team effort.

Those largely didn't – I mean, we have a team staffed up on it that continues to be lots of work, but that was largely –

[0:25:51.1] JM: Misdecoupled.

[0:25:52.2] SG: It was decoupled. That had started a year and a half, two years before this other effort. We had the data layer on the way. I think that the hardest part actually was making sure the mid-tier was coordinated, because I think it's very clear that okay, the listing page is going to have a presentation service. Then all of these presentation services would call some common mid-tier services. You want to make sure that how you do listing permissions, or that's one example, but there's a lot of these, I call them dependent services, but mid-tier services that you want to make sure that you can understand that, “Oh, these presentation services have the same use case. Let's not build two different services, or two services that overlap 80% that can happen a lot.”

[0:26:38.2] JM: Right, right. The listing presentation service is aggregating data from the pricing service and the deliver the latest photos service, or the description gathering service. There's four or five different services, and then you have some thread that gets spun off to all of those within the presentation service and then it makes sure is aggregating all of these and keeping a reasonable latency and has some failover and stuff.

[0:27:07.9] SG: Yeah. I mean, that's definite part of what we care about.

[0:27:11.4] JM: You went from the search team to the marketplace conversion team. It sounds like in some ways, the marketplace conversion team had overlap between what you were doing on search, because in search, to some degree what you're trying to optimize is a conversion. You want somebody to search and find something that they are willing to buy and to stay at. I'm assuming that's not the only thing you're optimizing for, because if you just got them to buy something and they had a terrible experience, that would that would not be a good outcome. What does the marketplace conversion team entail?

[0:27:47.1] SG: Yeah. Actually yeah, I realized that I didn't actually talked about my progression within Airbnb. Yeah, I was managing the search team and then I took on the umbrella team,

which was the marketplace conversion team. Then actually fast-forward a little bit and we moved into businesses last year and that's when I started managing engineering for the homes business. I can definitely talk about marketplace conversion.

That area was responsible for search, pricing, the booking experience, and also where the core services team lived. Those were some of the areas that then I was responsible for. It definitely helped a lot having the experience from search, just because you think of many aspects of the marketplace.

[0:28:30.3] JM: Do that marketplace conversion team get spun up? Is that when you joined it, or was it already in existence, like separate from the search team?

[0:28:39.1] SG: It was in existence, and so it was part of it. Then I started leading that team, I want to say, I think in 2016.

[0:28:48.1] JM: What's the interaction between the business side and the engineering side with a team like marketplace conversion?

[0:28:54.9] SG: Yeah. Maybe I can talk about the homes business actually, because what we did then is that we found these businesses that actually run – are responsible for their PNLs. We definitely – it's not just an engineering product team, it's actually all aspects of what it takes to run the homes business. We have somebody from policy, we have somebody from the customer support team, we have legal marketing. There's many aspects to what it takes to run a successful business. We have all of these people that come together to make sure that we are not just thinking of one aspect of what somebody needs to book a home, but we can actually take a more holistic view.

[0:29:36.7] JM: Sure. Give me an overview.

[0:29:38.3] SG: Sure. The homes business, it is the core business and it's what Airbnb started with. Just as we've grown, we have our policy team that works with different cities and makes sure that we can do what's right in each city. We're spending money on marketing and performance marketing, and so we want to make sure that our engineering team can work with

the performance marketers, right? I think what you what you find is that you do the best work when you bring the right skill sets together. It's not just engineering, or product that helps drive that forward.

[0:30:23.4] JM: Homes is one of the verticals. Then some of the other verticals are experiences and then I guess, there's probably some other businesses that are not fully public yet. I think of homes is that's – I think of that as everything. That's everything I use on Airbnb, right?

[0:30:38.6] SG: Airbnb's ambitions are to really think of travel as a whole. In fact, our experiences business launched in 2016. We're seeing really great growth there. Lots of great experiences. That's definitely I would say a big area of investment. We also have our luxe business, which is the really high-end places. Then finally, we have China which is a geographical focus.

[0:31:07.2] JM: Ah, yes. Okay. Now, as you have seen the company from so many different angles and you've also worked at Google. I'm sure you also spent plenty of time talking to other people at other engineering companies. I'm curious what you see as canonical problems across technology companies these days. As one example to kick that off that I'll maybe bring up to you is a challenge that I have talked to a lot of enterprises about recently, so there's obviously that we've got a monolith and we're migrating the monolith and that's challenging.

There's also the data challenge. Plenty of these companies, like legacy enterprises. This is true of legacy enterprises, but it's also true of somewhat newer companies like Airbnb, that the challenge of accessing data, permissioning the data throughout a company correctly, you want to have data scientists who can have free rein over the data, they can explore the data, they can have insights about it, maybe they can build machine learning models against it. This collection of problems where it's you've got ETL, you've got data engineering, you've got data science, you've got business analysts, you've got a VP of Marketing perhaps who is brilliant at marketing and can think in mathematical terms, but has no idea how long it takes to access a data reporter. They don't know how to get yet at the data report.

I think of the data platform as one canonical issue that companies are – large enterprises, and I guess enterprises, just large and small are encountering these days. Now just use that as a

kicking off point. Maybe you could explore that subject, or if you have other canonical things that you're seeing, either through the eyes of Airbnb, or your other conversations with other companies.

[0:32:51.6] SG: Yeah. One thing I'd say is that any company that goes through hyper-growth will have to tackle many problems, both organizationally and technically. Meaning that your systems don't scale up as quickly as the number of people you have. Then you have a lot of people making a lot of changes and that just necessary, it's rearchitecting your system. Even just redoing things like your performance review process and how you recruit and how you bring in people, just every aspect, how you evaluate people internally.

I just want to say that all hyper-growth companies will have challenges, both on the technical front and just on the organizational scaling front. I can definitely go deeper into some of the technical –

[0:33:36.0] JM: Please. Whatever is on your mind these days, or whatever you think I might hear from you that we might not hear from other companies, or from other individuals.

[0:33:45.5] SG: Okay. Maybe I can share a couple of areas that I think – I guess, we went into monolith a lot, so monolith to SOA. That's definitely one big area that you're right, everybody goes through the transition. I've given you more context there.

Another one that I think a lot of companies are trying to figure out is how to have high quality of their data. What that means is you touched on that with ETL and data engineers and all of that. I just think of what is the architecture that's going to make sure that when data lands in the warehouse, it is high quality that you can have tests against it. You can make sure that when the data is in, the job is complete, that you know that it meets all of the requirements you have. Not just in terms of SLAs, but also in terms of the quality. I would say that's an area that we definitely have been thinking about a lot.

What's interesting also over there is that's an area that also changes when you build these services, right? Because earlier, if you just have one big database with all of your information and that's one monolith writing to it, then the way you write to your data warehouse is different

from when you have many different services. The reason for that is when you have many services that own their own data, you want those services to write directly to the data warehouse. That is definitely a change again, that we we've defined and that we are beginning to make.

[0:35:11.6] JM: Did you say you want those services writing directly to the data warehouse?

[0:35:15.9] SG: What you don't want is this translation layer, where you have these services that write to the database and then from the database you get these mutations and then you recreate the business logic, right? You don't want to create duplication. You do want services to be able to eventually write these mutations to the data warehouse. There's a lot that goes in that process, but all I'm saying is you don't want – you want to avoid duplication of business logic.

[0:35:41.8] JM: Can you tell me more about the patterns by which different service teams get their data into the data warehouse, or how it makes its way into the hands of business analysts, or machine learning developers?

[0:35:56.0] SG: Yeah. We have what's in place today and then there's the architecture we want to move to. The way it works today is that we have these services that write to the database and then we look at the logs from the database and have a lot of hive scripts that recreate all of that business logic transforms that data into the schema that what we call is core data. That is roughly the architecture today.

Then from core data, there is downstream scripts that converted to derived tables. From there, then typically people do their analytics and the machine learning models over that. We are definitely revisiting that entire pipeline to make sure that it can support the scale we want and the testing we want and the reels, the code reels that we want.

[SPONSOR MESSAGE]

[0:36:56.8] JM: Backblaze is a company that offers unlimited cloud backup for Macs and PCs. You can use it to backup documents, music, photos, videos, drawings, projects, everything you

can think of. It's \$5 a month and you can use Backblaze to access all of your data anywhere in the world on the web, or on the phone.

It gives you the option to restore your files. You can restore files depending on your needs. You can also restore by mail. If you have a complete system failure, Backblaze has a feature where you can buy a hard drive and they put all your data on it and then overnight it via FedEx. Backblaze has restored over 600 petabytes and 30 billion files when people have lost them.

You want to have some backup system in place, or not. Maybe you are the Snapchat generation and you want to just have a Snapchat-like experience for all your files. In that case, Backblaze would not be for you. If you do not want disappearing files, you can check out backblaze.com/sed. Backblaze gives you a fully featured 15-day free trial at backblaze.com/sed. You can go there and play around with it and protect yourself from potential bad times.

Start it today by going to backblaze.com/sed and support Software Engineering Daily.

[INTERVIEW CONTINUED]

[0:38:42.0] JM: When you look at Airbnb's infrastructure and you look at the amazing managed services that are available to cloud providers, you think about your time at Google and the services you had available at Google. I realized that was five years ago now. Do you get the sense that the cloud services that are available to developers, are they nearing a place where it is on par with what you have available to you as an engineer at Google? To what degree is this Google infrastructure for everyone is this actually coming true?

[0:39:19.7] SG: It's hard for me to say, because the thing is systems keep evolving so much, right? There are certain things that are just a standard now, right? How do inter-service communication? There's thrift that a lot of people use and it was prior to that at Google, we had protocol buffers. Things like that are definitely standardized. I would say infrastructure just as a whole is constantly maturing and the open source technology is very good. We are on AWS and so are a lot of companies. I think all of that is definitely great.

I think that it's also just how long you've been working on a problem, how many engineers you have working on it, right? Even when I joined Google, the engineering team was bigger than where we are at Airbnb now. I think it's also interesting to just think of the scale and how long you've been doing something. You're right, I haven't been in Google now for five years, but when I was there I felt that open source was – it wasn't quite at par. Also in the last five years, there's been a lot of advancement in just what's out there in the industry. I know, I'm actually curious to hear from a current Googler how they think it compares.

[0:40:31.7] JM: Yeah. I would be too. We're going to get an anonymous source. Those are in these days. Management, that is I'm assuming a lot of what your time is spent on these days. Airbnb has been in hyper-growth for a while. If you were to not delegate properly, your work would just get impossibly complex. What have you learned about delegation, or building ways for you to stay sane while your responsibilities have increased, the scope of the company has increased, the population of engineers has increased? Do you have any tips for managers out there?

[0:41:14.1] SG: Yeah. I mean, honestly for me it's felt like a different job every six months. Part of the reason is that what we – I touched on this earlier, but when you have a company that's growing so fast, you need to figure out okay, is recruiting a bottleneck is the way you onboard engineers a bottleneck? There could be bottlenecks at many points. It's almost like every six months you want to rethink how you're doing things.

We've definitely had to go through a lot of that change. I would say for me, I think what's been helpful is actually even when I do my staff meetings with my directs, just aligning on what are the shared problems, what are the biggest problems and then who within our teams we can empower to take on a problem. I think that getting that process going is very important, because I alone, or even my directs and I alone, we cannot solve these problems.

The best thing we can do is find, or empower people that are already excited to work on this, right? Give them a chance to really channel that energy into coming up with a new proposal. That's definitely a model that I have tried a lot on my team, which is, and I had this realization also when I had this eng manager come to me and say, "Hey, I'm looking for some

opportunities. What can I work on?" I realized that actually, there's so much to work on. The person just didn't know what would be the highest impact project.

Even just making it clear that hey, these are the areas we think are worth working more on and here are the opportunities available. If you're interested, do you want to work on them? I think that getting that going is definitely super important. Then I think, also more time management. I think it's an area everyone can do more in myself included. Every few months I try, I'm like, "Okay, I'm going to reset my calendar or try to have these chunks of times here." You'll see as time goes on, you maybe get more lax about, "Okay, I'll take this meeting here and I'll take this meeting there."

I know, you need to keep revisiting your schedule, because making it sustainable is really your responsibility, right? No one's going to come and tell you, "Hey, I think you're doing too much." I mean, they might, but I think it's better for you to recognize that and then figure out what you can offload to others.

[0:43:41.8] JM: Tell me about working with the Airbnb senior management, the executive management. How do you ping them for like what they're looking for strategically and what do you – when you're looking at them, what do you seek to emulate?

[0:43:57.5] SG: Yeah. I mean, so the way we're structured now is that businesses run autonomously. We have a homes president and his name is Greg Greeley. I would say he is the person that we work – that I work with most closely in terms of the overall executive leadership team, and also the head of engineering. I would say that on both sides, I spend a lot of time with them on the engineering side making sure that we have an aligned set of priorities and direction of where we're headed. Then with Greg on the homes business side, right? Making sure we're making the right business decisions.

There are projects that sometimes cut across different teams. That's when, "Oh, there's a problem. We're not focusing on some area." That's when either I have a proposal, or somebody else is a proposal and then we share that with the executive team. I would say that with the move to businesses, it's definitely a little bit more structured, the communication. Which I think is

good, because a business has its clear goals and we should be able to execute on it without needing too much support on a day-to-day basis.

[0:45:08.6] JM: How has your interaction with the hiring process changed since you've become a director of engineering?

[0:45:16.1] SG: Yeah. I mean, I think that if I think of my role right now as engineering lead for homes, a lot of it is actually has – a lot of my time actually, the last few months has been focused on how do we make sure that the business as a whole can bring in engineering candidates at the at the speed that we want? I actually partner very closely with the recruiting team on what are the goals that we want to meet? What are the type of engineers we want to bring in? How many recruiters do we need for that? How can we have all eng managers contribute to bringing in great candidates?

I've been spending a fair bit of time on just reviewing this whole process. If I take a step back and think of the hiring process, I think that's one area where just as you scale, things really have to change. An example is at the beginning, all eng managers used to review the packets that would come in, right? I remember I would read every single interview feedback, but then you realize quickly that that doesn't really scale, so you need to figure out then okay, do you set up on-call rotations for people that need to definitely – to make sure somebody reviews the packet and says yes, this is a go, or a no go. That's one example of just how that process is to scale, and there are many such changes that we've had to make.

[0:46:30.3] JM: One thing I am curious about is tooling, not directly at the engineering level, but these next generation office productivity tools, or things like Slack and Asana and Air Table. Are there any of these that have really changed your way of thinking about communication, or have really led to dramatic step changes, or just complete alterations in productivity, or team management?

[0:47:01.0] SG: I mean, I would say the biggest one that we use right now is Slack. Even just when we want to make a – we want to deploy code to production, that's where the communication happens. It's definitely a big part of our workflow. Ultimately, I think all of these tools, even how you write documents, these things are very important. A lot of times, what's

important is also consistency, making sure that everyone is using the same set of tools, because that's where you really get the efficiency. Absolutely, I mean, I these workplace management tools are super important and I hope that they continue to evolve and add better features.

[0:47:42.1] JM: Just to conclude, I'm wondering if there has been a particular challenge that you've encountered at the engineering management level, either a recurring challenge, or a particularly bad, or acute experience that people in engineering management who are listening could learn from.

[0:48:02.2] SG: I would say that people sometimes think that engineering management is a very glamorous job. This is more so for actually individual contributors who are considering the path and also senior management. I think people assume that okay, you get to make all these decisions and oh, it must be so great. I think the truth is that you also have to deal with a lot of problems. The larger your org is, the higher the likelihood that there is some problem. You just have to make sure that that's what you want and that's what you feel excited doing.

[0:48:35.1] JM: Have there been any books that have been particularly useful to you from the management point of view?

[0:48:40.3] SG: I like reading about business strategy and seeing how people think of – how people run their companies. Actually one that we're doing a bit of a book club within homes is it's called [inaudible 0:48:50.8], so I've been reading that. Then another book that actually I really want to read is [inaudible 0:48:55.7] just wrote this book and I want to get my hands on that.

[0:49:00.1] JM: Yeah, definitely. Okay, well Surabhi, this has been really great talking to you. I appreciate you coming on Software Engineering Daily.

[0:49:05.8] SG: Thank you so much.

[END OF INTERVIEW]

[0:49:10.1] JM: DigitalOcean is a reliable, easy-to-use cloud provider. I've used DigitalOcean for years, whenever I want to get an application off the ground quickly. I've always loved the focus on user experience, the great documentation and the simple user interface. More and more, people are finding out about DigitalOcean and realizing that DigitalOcean is perfect for their application workloads.

This year, DigitalOcean is making that even easier with new node types. A \$15 flexible droplet that can mix and match different configurations of CPU and RAM to get the perfect amount of resources for your application. There are also CPU-optimized droplets perfect for highly active frontend servers, or CICD workloads.

Running on the cloud can get expensive, which is why DigitalOcean makes it easy to choose the right size instance. The prices on standard instances have gone down too. You can check out all their new deals by going to do.co/sedaily. As a bonus to our listeners, you will get a \$100 in credit to use over 60 days. That's a lot of money to experiment with.

You can make a \$100 go pretty far on DigitalOcean. You can use the credit for hosting, or infrastructure and that includes load balancers, object storage, DigitalOcean spaces is a great new product that provides object storage, and of course computation. Get your free \$100 credit at do.co/sedaily.

Thanks to DigitalOcean for being a sponsor. The co-founder of DigitalOcean Moisey Uretsky was one of the first people I interviewed and his interview was really inspirational for me, so I've always thought of DigitalOcean as a pretty inspirational company. Thank you, DigitalOcean.

[END]