EPISODE 1329

[INTRODUCTION]

[00:00:00] ANNOUNCER: ETL stands for extract, transform load, and refers to the process of integrating data from many different sources into one location, usually a data warehouse. This process has become especially important for companies as they use many different services to collect and manage data. The company, Grouparoo, provides an open source framework that helps you move data between your data warehouse and all of your cloud-based tools. This process of moving data back from the data warehouse to the applications is called reverse ETL. And it's important for things like marketing campaigns and customer service. Grouparoo can easily integrate with your developers' tools and is free and easy to install. In this episode, we talk with Brian Leonard, CEO at Grouparo

[INTERVIEW]

[00:00:49] JM: Brian, welcome to the show.

[00:00:50] BL: Hey, glad to be here.

[00:00:52] JM: Good to have you. We're talking today about ETL, or modern ETL. I feel like ETL is no longer the right way to talk about it. You have ETL. You have reverse ETL. Really what we're talking about is data transfer, right? Just high velocity data transfer.

[00:01:12] BL: Yeah, and there's a pretty strong case that the best way to do ETL is actually ELT, by the way. So just to further confuse the issue. And so overall, I think what we're talking about here is data pipelines and the best way to collect and then make use of your data in your business.

[00:01:32] JM: The companies I've talked to in the past that are in this area; Airbyte, that is open source forward and reverse ETL, Hightouch, that is closed source reverse ETL. You have Fivetran, which is forward ETL. And you have – What's the other one? Census, which is also closed source reverse ETL. All of these companies seem to be doing super well. What else is there?

[00:02:02] BL: What else is there? I've talked with Airbyte. I'm not sure they're focused – I think they're mostly the Fivetran forward ETL. I like that we've retro actively added a new thing to theirs, which is great too with the reverse and the forward. Super good.

[00:02:17] JM: I thought they were forward and reverse. I'm searching. I'm searching. Open source data integration pipelines for modern teams, modern data teams. Get your data pipelines running in minutes with pre-built or custom connectors from the Airbyte UI, API or CLI. Sounds like forward and reverse to me. Extract from sources, load to destinations and transform.

[00:02:39] BL: There you go. So they're going their destinations, among the other issues that we have in this lexicon, is what a source and a destination means. For Airbyte, a source is MailChimp, let's say, Salesforce and like that. And a destination is Snowflake, something like that. For us, it's, well, the reverse. Anyway, back to your question around the other things that are missing from that equation. I think the transform layer. So if Airbyte is doing the extracting and loading into BigQuery, Snowflake, what's doing the transformations? In the open source world, some proprietary stuff, but mostly open sources, is the DBT. That's a big part of that equation. How can we run a sequence of SQL statements to transform that data? Combining the various different tables very often into like a customer roll-up table, for example, or the kinds of things that you would then use in in analytics, BI. So there's the BI picture to that modern data stack too. What are we using to introspect the warehouse? And tools like Tableau, and Looker, and Preset and Metabase are in that category.

And then in the reverse ETL space with us, and Census, and Hightouch, it's about how are we going to make use of that data? How are we going to teach it about our data model and put that combined data or any given insights that we've made back into the tools that we're using? So how can we update Snowflake with what people are doing in our product so that we can better help our sales team reach out to the right people, for example?

[00:04:30] JM: Wait. You were CTO of TaskRabbit for 10 years. That's a little bit of context for your technical savvy. Tell me about the ETL problems or ETL – Or maybe I should say more broadly, data engineering problems that you experienced at TaskRabbit. By the way, TaskRabbit, phenomenal product. I actually think I'm going to use it today.

[00:04:51] BL: Oh, there you go. Well, I'm no longer there, but wholeheartedly believe it's still just at the beginning of its journey.

[00:05:01] JM: Still a great product. Pretty good post-acquisition still. Arguably still a good competitor to Thumbtack or whatever the market leader is. Or maybe they're the market leader. I don't know, it's kind of like a pretty wide open design space. Who's the market leader in that space? Is it still TaskRabbit?

[00:05:19] BL: I mean, any space can be sub divided up. The leader in getting things done in your house, or just people doing it themselves by far.

[00:05:27] JM: I don't know. Whatever it is – Like I need a house cleaner today. I'll tell you, like no offense to Handy. The market leader is not Handy. Sorry.

[00:05:35] BL: Fair enough. So TaskRabbit on getting it done today to my knowledge is probably still the market leader. The putting it out for bid and roofers and things like that, probably more in the Thumbtack. And I've noticed Angie making some moves in this. They rebranded and are kind of going in that direction. So be curious where they are right now.

Anyway, back to the data at TaskRabbit. I think it goes through a – We went through a pretty common journey. I built that app on MySQL. I'd probably picked Postgres today. Built that up on MySQL. We had the data in MySQL. It had people's user accounts, and tasks, and what they've been up to. At some point, we needed to combine data into that. And we started, we created an ETL process. So we were pulling in things from Zendesk, for example. We were updating our – We copied over our product tables into Redshift. Then we transform those a little bit. Maybe there's a transform that does some sort of roll-up or even some prediction of when they're going to book another task or give them a score or something like that, those kinds of transformations. And then we put Looker on top of that. We were one of the early Looker users held out with that pricing for quite a while, much to their chagrin possibly, either way. But some hard negotiations there.

[00:07:08] JM: Whatever. They can afford it.

[00:07:09] BL: Yeah, they're doing fine. They're doing fine. And so we spun everybody up on Looker. In general, try to go more towards the self-serve to add more value to the data team. I was in charge of the data team through most of this story. At some point, we weren't getting our data needs fully met by Redshift. We switched. At that point, Snowflake had come up. Snowflake and BigQuery are the leaders I see now mostly with their separation of the compute power and the storage. And so you can have more in there. It's really closer to how you're really using it and what you're paying for, which is really great.

And then when the reverse ETL comes into play, it starts to be like, “Okay, great, these great reports are in Looker. We've got these cohort analysis. This user 32 is like about to churn or whatever it is, or doing really great.” How can we take advantage of that in our marketing tools? Push an email, for example? We were using Urban Airship and **[inaudible 00:08:17]**. And that's when the reverse ETL thing start to come into question. And so at the time, we built something to synchronize that tool, synchronize from that warehouse into those tools. And it's not the thing the engineers like to work on that much. Like we put together a minimum viable product. We got it going. And yeah, we called it the end of the day, like, “Hey, marketing, you're welcome.” But the thing is great marketing, especially great personalization and retention marketing, they have a new idea every week. And they should always be testing and then seeing how those go. And in general, we couldn't keep up and didn't prioritize that. And so I do crazy things like approve a million dollars for our retention campaign on a Monday and then a Thursday say, “No, I can't get the last time somebody did cleaning into your thing.” And she's like, “Well, what about the goal?” And I said, “Well, good luck. I believe in you.” But they need a day to hit the goal. And then we'd be confused why we didn't hit the goal. And it's because they didn't have the agility of testing that they needed. So I switched my tune. And I'm now here to empower all those teams to be successful.

[00:09:32] JM: So when Srini from Preset connected us, it didn't surprise me at all, but there is another open source forward reverse ETL, or data engineering, or whatever you want to call this space, company. Because this problem is so big, it's such a diverse pain point, and the adjacencies are enormous. And I kind of wonder why that – I mean, okay, here's my thesis on how we got to where we are, and I'd love to hear yours. So we accidentally ended up in a world where the data warehouse is the center of the universe. Everyone, for some period of time, was thinking it was going to be some sort of streaming system or like a Lambda architecture kind of thing. Instead, we just said, “You know what? The simplest way to think about this is we throw everything in the data warehouse, and we go from there.” And then because of that, we had a ton of downstream infrastructure developments. Do you think that thesis is reasonable?

[00:10:28] BL: I think that's super reasonable. I think it's like the warehouse probably driven by execs one and good reports, if I had to guess, in short time period, became the sun in this universe. And now things are revolving around it, and things are going in and things are going out. I think the reverse ETL problem existed way before that happened, which I think was probably in the last, I don't know, five years. I think people have always wanted to make better use of their data and needed ways to do that. It was just always – Because it's transforming their custom data model into something structured. Your crazy Postgres tables, or whatever, into, I don't know, Marketo, or Zendesk. It always felt like it had to be some custom process to normalize your data into that. But as we've gotten better at putting all our data in one place and the normalization be done through those queries, and/or it's just time to solve this problem, us and others have decided that it's finally time to put that data to use in a more, I don't know what the right word is, a platform to make that successful.

[00:11:55] JM: And your stuff is open source, right? You're doing open source. Okay. So I'm very suspect of the closed source providers, and that includes Hightouch, which I invested in. I don't know why you would make this stuff closed source. It doesn't make any sense to me.

[00:12:14] BL: If I had to guess, I don't think it's – I mean, clearly, I'm a proponent of open source. And I think it brings tons of value in, A, the ability to host to yourself. B, the clarity of what's going on. C, this game is about having approximately infinite destinations to use your data. And I think it's an amazing like long tail thing. We're talking to people in Brazil and Vietnam, and integrating with the Brazilian and Vietnamese MailChimps. Like a closed source SaaS US-based service isn't going to get to that in years. Like Ssegment still hasn't gotten to those providers, right? And so like there's a ton of value and all that.

I think if you asked Hightouch, I don't think they'd say like close source is ours. Don't touch it. I think it's more about that they believe in the hosted model, which we're also experimenting with, because not everybody can spin up their own instances to their DevOps things. And they want to outsource that to a SaaS provider. So I don't think they would necessarily go to the map saying you can never look at their source code, but like I think they're probably just traditional SaaS vibes.

[00:13:34] JM: Yeah, but it's like a bunch of kids. It's like kids. I'm telling them, "You guys. What are you doing? Just make it open source. You'd lose nothing. You lose absolutely nothing. Nobody wants to stand up this thing and run it themselves. If they do, they're not your right customer anyway. So just open source it."

[00:13:51] BL: Right. Yeah, I mean, now we're in philosophy land. I clearly believe that open source is good. But maybe it slows down our velocity to do things out in the open. Maybe it doesn't. But there's probably a typical set of concerns. Like why didn't we open source TaskRabbit, I guess, might be a random question?

[00:14:12] JM: That's a valid question. Dude, email, like whatever, ceo@ikea.com and ask him or her. Do they have a CEO? They're like Icelandic or something?

[00:14:23] BL: Swedish man. You don't know that IKEA is Swedish? It's very famously Swedish.

[00:14:27] JM: I'm sorry. The meatballs, I've read about the meatballs.

[00:14:30] BL: Very famously Swedish, my friend. Yeah. So I mean, why is it everything open source? And I mean, I'm sure there are –

[00:14:38] JM: Which is a very, very valid question. It is a truly valid question.

[00:14:43] BL: Auditability, and all kinds of other things. I think that people believe probably rightly that there're some secret sauces in there and maybe there's some dark tactics that they don't want fully exposed. Like that's super normal probably too.

[00:14:58] JM: Maybe.

[00:15:00] BL: Maybe.

[00:15:01] JM: Listen, man, I just wrote a book about Facebook. So I think we could talk about that for a long time.

[00:15:06] BL: Right. Yeah, exactly. Anything could be open source for sure. And I think particularly when it comes for these network things like we're doing, when it comes with your data, when it comes with the ability to self-host, which is super relevant to a lot of financial medical companies, for example, I think there's a bunch of benefits there. We're doing a conference. Get a quick plug-in. September 28th.

[00:15:30] JM: Like a real life conference, or like a virtual one?

[00:15:35] BL: I mean, there will be real humans involved. We will be on videos.

[00:15:39] JM: Will have to wear like a hazmat suit, or you can you actually be in a physical presence where there is air movement between people?

[00:15:48] BL: Yeah, as long as it's your own thing, and you're watching the video stream through our thing. It's virtual conference. Open Source Datastack.

[00:15:54] JM: Come on. Can we like have a restaurant event or something at joining it?

[00:15:58] BL: We're getting there. We're getting there, friend. But opensourcedatastack.com. Us and several other open source companies are presenting on like what this full pipeline looks like. And we think there's a ton of value, the kinds of things we've been saying on open source.

[00:16:13] JM: And like what are some of the talks at the conference? Like what do you see as the most prominent problems and subjects in this ecosystem? I mean, you mentioned a little bit earlier with all those companies, but more accurately the domains, the emergent domains, the emergent problems that true enterprises and startups are dealing with?

[00:16:32] BL: Well, that's not what the talks are. The talks are sort of pulling out like an example company and how you walk through this pipeline, building the modern data stack with the open source tools. And so we have Meltano and Srini from Preset, and Dagster, and Snowplow, and us, and – Oh, gosh. Who am I forgetting? DBT involved in that. But I'm happy to talk about the problems too. I don't think it's a surprise to anyone that like data quality and use is a differentiator in your business. Like we need to make good decisions. We need to hit people with the right messaging at the right time in the product, outside the product, etc. And so like, all of these, anything we're doing with the warehouse thesis is to make better decisions and to create better experiences. And that's different for every single business. And we've seen a lot of people focusing on. And still, there's like a Maslow's hierarchy of needs or something here on like, "Okay, first, we need to be able to read the data we currently have, and then maybe need to put it together in some place. And then we need to create business intelligence on top of that," right? And so people are walking up that chain. I've seen struggles for people creating their first warehouse, and how am I going to get my Postgres data into that? To, "Okay, now all these stuff is locked in." Third-party tools, events and profile data and all kinds of things. How do we combine all of those? And the current best approaches the dump it all in the warehouse sort of thing. If that fits your scale needs? Then it's like, "Okay, great. How are we going to ramp our teams up to make use of this data? How can we transform it into something that is useful? And then how can we teach them to use that so they can self-serve at TaskRabbit? Great day when our operations team like answer their own questions about who the best taskers were in San Francisco or whatever, instead of asking my data analyst, right? And then sort of at the pinnacle of all of that is how can we operationalize this intelligence that we've created to create those better experiences?

[00:19:01] JM: DBT. Why is DBT an N of one company? There are a multitude of ETL and reverse ETL providers. There's only one DBT. What is DVT? Why is it so important in this ecosystem? I asked this question in pretty much every data engineering conversation I have. I need to understand why this company is so momentous.

[00:19:23] BL: I like that. Momentous is like a big deal and creating momentum or something. Anyway –

[00:19:31] JM: I think it’s more like it is of the moment.

[00:19:33] BL: Of the moment. Exactly.

[00:19:35] JM: Or definitive of the moment.

[00:19:37] BL: Exactly. I think that, like I said, we reached this time when like the warehouse is more accessible from a price and sort of zeitgeist perspective. We've got lots of different ways to write into it, which is maybe why there's lots of different like Fivetran-ish things. Like there're many different things you need to write into it. There're many different things you need to write out of it. That's the reverse ETL case. But just in that transforming layer, I think they have a win of extreme focus there, which is how a lot of people win at things. So I respect that a lot. And this new level of accessibility to a broader set of engineers and even people that wouldn't have considered themselves engineers, previously, data analysts.

And so if you can write SQL, you can now do this data engineering task that previously was only accessible to a “software engineer, data engineer” writing Python code, and it all seems super fragile. They've hardened and made accessible a pattern that made this notion of the analytics engineer, or essentially dating analysts, able to do the skills that they could – Accomplish a goal they couldn't before. You buy that? Is that what the other people said? What did they say?

[00:21:14] JM: You know what I think? To the best of my knowledge, it's kind of like you can't have Kubernetes without YAML, right? And you can't have a data engineering ecosystem without DBT.

[00:21:27] BL: That's certainly what they want you to think. And like I think it's the current best way to do it, for sure.

[00:21:33] JM: Hmm. Can you make a better one?

[00:21:37] BL: I mean, theoretically. But they’ve been doing this for a long time. All overnight successes are five years in the making.

[00:21:46] JM: I mean, the DBT story is so phenomenal. I love that story. It's actually kind of similar to the great expectation story. I don't know how much of a believer you are in the great expectations data testing kind of thing. But it's sort of the same story, right? We started as kind of like a consulting firm, and then we have these emergent breakages of data engineering, and we accidentally start a company. It's kind of a beautiful product market fit discovery.

[00:22:12] BL: Exactly. So like it's unclear if I can make a better one, and it's probably not worth the time. Because focus –

[00:22:20] JM: Well, you'll probably find something else. As an emergent property of the brokenness of the space, I'm sure you'll find something else.

[00:22:28] BL: Exactly, exactly. And so I think these whole ecosystems go through like this sine curve or something of like best of breed solutions, like pieced together into the perfect thing for you, and then some really big platform. And so I think if something is going to displace the current situation, it's all moving so fast, which is best of breed solutions sort of around that data warehouse. It's going to be some platform that's bringing all of that together. Now, obviously, we believe we're a best of breed solution on that, and we're not seeking to be that full platform. But a few years down the road, the curve might go the other way and some combo warehouse transform in and out kind of thing might win because it's the integration, essentially.

[00:23:28] JM: So I was looking at your Crunchbase, and Hack VC is one of our investors, right? Ed Roman? I have known Ed Roman since I was like 13 years old.

[00:23:41] BL: Maybe you're old. And that's been a while, because you called some people that are not terribly young kids earlier.

[00:23:49] JM: I'm 33. I'm 33. But I was definitely playing magic 20 years ago. So I met Ed Roman playing magic.

[00:23:55] BL: Very good.

[00:23:56] JM: He was part of one the factions of the Austin Texas Magic player diaspora you might say.

[00:24:05] BL: Diaspora. I like that concept.

[00:24:08] JM: Are you a Magic player? Have you ever played Magic the Gathering?

[00:24:10] BL: No, not Magic. I have a son. He's into Dungeons and Dragon and a little bit of Pokemon. Generally familiar with these things. I miss that level. Like there's all kinds of dorks in the world. I wasn't in that group.

[00:24:25] JM: Hey, man, Magic is a sport. Magic is a sport. It has nothing to do with dorkery.

[00:24:29] BL: I mean, geeks. Geeks are people that are excited about –

[00:24:32] JM: You know what's funny? You know what’s funny? So all the guys, all the guys, all the jocks that used to like beat me up and make fun of me for Magic. Once everything shifted to poker in high school and I started smashing them and taking their money and making more money than all of them make even today in high school, they still didn't respect me. It was kind of funny.

[00:24:51] BL: Well, you know, you pick your vibe of what you respect. And I guess it takes a lot of data and maybe never. It's hard to change people, my friend. It's hard to change people.

[00:25:02] JM: I agree. But even if you don't change them, you can still take their money at the poker table. That's what I like about it.

[00:25:07] BL: I think that's the ideal situation if you really want to get into it.

[00:25:10] JM: Pretty much.

[00:25:11] BL: If they're not learning, like, yeah, that's what you want as on the other side of the table.

[00:25:17] JM: So let me ask you. As somebody who has pitched several businesses to Ed Roman unsuccessfully, how do you convince that guy to invest in your company? By the way, Ed Roman, one of the best investors in the Valley when it comes to infrastructure is my belief, and fairly under the radar for how good he is.

[00:25:40] BL: Yeah, definitely. I mean, as seen by this Magic diaspora situation, super connected and super insightful in helping us out just recently, in fact. I guess I don't know the secret per se, because didn't like over-target him specifically in my pitch. Like maybe you just said, he's one of the best infrastructure investors. So like it's good to have the infrastructure. I don't know if that's what you pitched or not.

[00:26:11] JM: I was actually pitching him on a –

[00:26:13] BL: Magic the Gathering?

[00:26:14] JM: Well, a gaming company. Yeah. So I'm building a gaming company that's like an improvement on Magic the Gathering. And that's what I pitched him on. And it was a not right now kind of response.

[00:26:25] BL: Yeah, he wishes you the best of luck. He's rooting for you for the sidelines.

[00:26:28] JM: He’s rooting for me for the sidelines. He looks forward to seeing lines, not dots.

[00:26:34] BL: That's good. I like that.

[00:26:36] JM: Have you heard that one before?

[00:26:37] BL: I haven't heard that one.

[00:26:38] JM: The lines, not dots?

[00:26:40] BL: I haven't heard lines.

[00:26:41] JM: We invest in lines, not dots. Okay.

[00:26:44] BL: Well, it only takes two dots to make a line, my friends.

[00:26:47] JM: That’s true. Technically, we have dots. We have a dot at the beginning of our conversation. We have a dot halfway through the conversation where you're telling me no. Don't we have a line?

[00:26:58] BL: Yeah, right to the hard no. But VCs are in the business of optionality. So it's a diagonal line right down to the hard no. But it makes one of those little like – Oh, my gosh. I've lost my math terms. It never quite hits the hard no.

[00:27:15] JM: Slope?

[00:27:16] BL: No. The tangent – No.

[00:27:19] JM: The Y intercept?

[00:27:21] BL: Well, no, it doesn't y intercept. It curves. Like infinitely approaching. Hyperbole? No. Hypotenuse? No. Dammit. This is embarrassing.

[00:27:30] JM: Anyway, my theory on this whole –

[00:27:31] BL: Anyway, it wasn't an infrastructure company. So they didn't say yes. But he likes open source. And he likes infrastructure. We had a strong team that has worked together before. He likes data. I don't know. I think that's my advice on it, I guess.

[00:27:46] JM: Yeah. I mean, speaking honestly, if you can invest in the amount of deals that Ed Roman invest in, you probably pass on that crazy podcaster guy pitching you a gaming company. But at the same time, this whole optionality strategy that they run. So Sequoia tried to run this on me, Andreessen tried to run this on me, and I told them, “You will never be able to invest in this company ever again," because you have to deprive them of the optionality. Otherwise, they're just going to exploit you constantly. It's very frustrating.

I did like three pitch meetings with Andreessen. And then at the end they told me, "Sorry, we're going to wait to see more traction." I said, "I'm sorry, you will not have another chance to invest in this if you don't invest now." And that's what you have to do. How'd it go? I didn't get a response. I'm like, "Come on, I give you like three hours of my time, several long winded emails, and you can even like have a follow-up acknowledgement. Like how savage do you need to be?"

[00:28:51] BL: Asymptote. I did some Googling. That's what I was looking for. Yeah. So we brought on Fuel and ENIAC and Ed, and a few others.

[00:29:02] JM: Who's Eniac? Who does Eniac? It's a brand name for a venture firm.

[00:29:07] BL: Yeah, I was told once that – someone corrected me that I was supposed to say Eniac too, but I don't know. Either way, they're out in New York. Four partners, all of which went to Penn with where the ENIAC system was. And so they took that as inspiration working with a guy named Hadley they're. Super well-connected, super bright bunch, super founder, supportive.

[00:29:33] JM: What's your favorite like giant vacuum tube era computer system? Is it ENIAC, UNIVAC? I don't even know anything about them. I just like them because they have names that are entirely in capital letters.

[00:29:47] BL: Yes, exactly. Acronyms. I guess I don't think I've thought of this before. But I think this notion of like all of the computing power, like to put someone on the moon is super interesting me. Like the rooms that they had and all of that to calculate those curves and whatnot. I think that's super interesting. I'm from Houston originally. So I got to –

[00:30:13] JM: Oh, you're from Houston? I'm from Austin. Awesome.

[00:30:15] BL: I lived in Austin. I worked for IBM on Burnet over there.

[00:30:22] JM: I guess our locality didn't overlap in the sense. But I worked in eBay. eBay had an office like right next to IBM. Wait, you're talking my like up north, right?

[00:30:35] BL: Yeah, sort of, I don't know, between the Mopac and 35 metric approximately.

[00:30:43] JM: Up north. It's North Austin. Like far north enough that you kind of dodge some of the worst traffic?

[00:30:49] BL: That's right.

[00:30:50] JM: Yeah.

[00:30:51] BL: IBM made that site a million years ago. And then like eBay and others probably kind of like –

[00:31:01] JM: Adjoinder.

[00:31:02] BL: Was it like Intel opened up right next to Fairchild or whatever in the Valley, that kind of idea.

[00:31:08] JM: Yeah. That kind of vibe. That's exactly what we're talking about. Yeah, that's exactly the kind – Yeah, IBM opens a random office in North Austin. eBay opens a random office right next to it. That's exactly the same vibe.

[00:31:21] BL: Exactly.

[00:31:22] JM: We'll get back to ETL in a sec, or data engineering. But what were you doing at IBM at the time? What year was that? Was like right after college?

[00:31:29] BL: Yeah, I did several internships in college. Any college age people listening, I highly recommend doing those internships. I did three internships while in college. Even co-oping is what we call it. I went to Clemson. Some of those were during spring or fall semesters.

Sort of on and off. That was my last one. And I was in this program IBM had, I think still had, which is called the Extreme Blue Program. So I was working with Tivoli, which IBM had bought, which was kind of like a – I don't know what you want to call it. Like a deployment management, many systems kind of think. Maybe they'd be using Kubernetes or something to get all of these machines these days. Yeah, so I was there for six months. And then Tivoli is really a recruiting program. They don't really care what I did for Tivoli. They then recruited me into working for IBM, which I did in Boston for several years.

[00:32:28] JM: Gotcha. Anyway, so let's go back to – I just don't even want to think about my time working in North Austin ever again. Dude, I drove every day, 30 minutes up, 30 minutes down, just listening to podcasts, searching for an answer that would allow me to escape that grind.

[00:32:48] BL: There it is. Did you find it?

[00:32:51] JM: I think so.

[00:32:51] BL: There you go.

[00:32:54] JM: I was listening to Software Engineering radio at the time. That's the podcast – That was my training ground basically. I would listen to that podcast. And then eventually I heard they were looking for volunteer podcasters. And I said, "Oh! Well, okay, I'll try that." And 12 years later, whatever, here we are. But I digress. We have like 1600 episodes at this point.

Okay, all these integrations, Salesforce integration, MySQL integration, Postgres integration, Redshift integration, Intercom integration, MailChimp integration, Salesforce, I assume. You have to have this N by N data transfer issue, schema transformations. The first time I encountered the depth of this problem was when I talked to George at Fivetran maybe two and a half years ago. Ever since then, I've just heard more and more about it. How do you solve this as an engineering problem where you have these divergent schemas, you have to unify the schemas, and you have to build good data transfer between them? This is like a massive problem. How do you approach it?

[00:34:00] BL: It's interesting, because I think Fivetran has it easier than we do. I think that's every company ever. Grass is always greener kind of thing. But they're writing from a known situation generally. They're using API's from Salesforce, and writing into a fairly small set of destinations, Snowflake, etcetera.

I think the interesting thing about reverse ETL as opposed to the ELT approach, writing into the warehouse, the interesting thing about reverse ETL is that there's also this notion of the customer's data model, right? That is somewhat separate from all those other things. I happen to think that's a good reason to be like – The more it's about your specific data model, the more that open source tooling makes sense. I think that's why DBT is open source, for example, to express that data model in that open source framework. And we're doing the same thing.

So our approach is sort of a declarative approach in that way, which is that you express the data that's important to you. We then have – If it was really N by N, you'd be screwed, obviously, right? And so you query those sources, MySQL, Postgres, Snowflake, BigQuery, etcetera. Then you have your data in a fairly normalized form. Here's the customer, and they're a bunch of integers, and strings, and floats, and whatnot. And then you make an abstraction API to all those destinations and say, "Here, please make this so in MailChimp." And then it's all about unit tests, and API rate limiting, learnings, and all these other things you have to deal with especially on the on the way out.

[00:35:52] JM: Okay, what is the hardest engineering problem you've had to solve so far?

[00:35:57] BL: The hardest engineering problem, single problem, I think is like – I don't know. In general, synchronizing with your source and just general efficiency throughout the system, and is something we were always iterating on and trying to get right. You can't miss anything. Like that's like SLA number one. But at the same time, want to be as efficient as possible both with data, with querying, and especially on the way out with various rate limiting and things like that. So just figuring out the right way to be efficient there.

[00:36:39] JM: And go to market? What about go to market? Is it easy to convince people to use Grouparoo? Or do you have to sort of win them over from other market participants?

[00:36:50] BL: I think that an interesting thing we're seeing is where people are on that, maybe on that pyramid that I talked about earlier. It depends where you are in your data journey. If you're still struggling to get everything in your warehouse and/or querying or something like that, then I get some glazed eyes like, “Wait. You're talking about making use of this data. We can't even reliably pull it together,” that kind of thing.

But if you're in the spot where, essentially, the engineering team or the data team, the data engineering team, has it on their plates to integrate with these things and make use of this data, we take a problem that's six weeks and make it 15 minutes. And we set you up with an architecture for the future in which every engineer I've ever met likes doing things now and they like predicting that they're going to thank their past selves, right? Like, “Oh, smart move.” This migration from MailChimp to Iterable is super easy. It’s one thing we make super easy. For example, “Thank you past self.” Or, “Oh, you want to integrate one more thing? Super easy.” Like, “Oh, thanks.” It's like a Ponzi scheme of software engineering, architectural decisions. Until it all falls apart, which it does sometimes. But when you make the right choice, you feel really good about it. So you like to set yourself up for that. So in those cases, when they're ready to go, we see that it’s pretty easy.

[00:38:25] JM: It seems to me that the economics of this business are so insanely good. You basically have the economics of an API company. It's slightly worse than a raw API company, because you have a lot of data to handle. But maybe you can price that appropriately. How are the economics?

[00:38:44] BL: Well, the economics of selling license for self-hosting are insanely good, first of all. That's a very interesting angle.

[00:38:53] JM: Wait. That's what you do? I think people can just like stand up open source stuff themselves? They need a license.

[00:38:59] BL: There's an enterprise edition that you can self-host. So like in terms of economics, like that's the most amazing one, for sure. Seeing there's no hosting to do and things like that. For the SaaS hosting that we're doing, I think it's yet to be seen what the right data model is, frankly. Right now, out there, we see people charging based on the number of connections. I've seen people charging based on, basically, the number of data movements, runs of the thing based on the number of rows that are being transferred. I'm very interested to find the right way that doesn't, I think, often, disincentivize the right behaviors and customer success with your pricing if you don't do it just right.

And so for example, vaguely the same picture as us but accomplished in a very different way is Segment, which is like events, and then they chain those, and they charge per event, which makes a lot of sense, because that's where their variable costs are. But then you have all these like not so great conversations every six weeks internally when it gets too expensive, “Are we really using this data? Should this really be something that we're sending?” and things like that.

And so charging on the number of connections, we want it to be able to do a number of connections. And charging on the data that you're sending through it, again, goes with our fixed costs, but we don't want to limit our success. And so like, to some degree, I think the equation on those economics are still out, and we have to iterate on what the right model is. Right now we have a flat fee, 150 bucks, and whatever it is, while we're in learning and seed funded mode.

[00:40:54] JM: For self-hosted, are you using Replicated?

[00:40:57] BL: We are not, although it's very interesting. We have Docker things that you can do where you can just spin up an instance like you would any other node project, or node-based.

[00:41:09] JM: Did you look at Replicate? Do you think about that at all?

[00:41:12] BL: I’ve about it a few times, and I went to their site. We're not seeing that. I don’t know. What do you think the benefit would be for us, I guess?

[00:41:20] JM: Replicate is interesting. Full disclosure, they've been a sponsor of us in the past. But I'll give you my honest perspective, or the extent to which I can be honest about this. To me, the problem that they're solving seems almost impossible. Basically, like we're going to be your package manager for a distributed system. Maybe. Can you really do that? Maybe.

They do have some amazing logos, which is very impressive, but I don't know how the logos are actually using their product. So if I knew more about that, then maybe – I mean, for example, ReadMe, I know, uses them. And ReadMe is a very happy participant in their ecosystem. And I like ReadMe a lot. ReadMe is like a perfect example, because ReadMe is fairly simple. We're hosted ReadMe software. That's what we do. It needs to be a distributed systems. So you do need a distributed systems package management system. But it's ultimately not very complicated. ReadMe is like a very, very nice blogging system basically, or crud system. It's a beautiful, beautiful application. But it's not super complicated. There're not as many opportunities for timeouts and external dependencies and stuff as there is in what you're building. So when I think about trusting some third party vendor to be the arbiter of whether or not my software works, I don't know if I'm trusting anybody else other than me and my Docker containers for venting my software. So I probably would do what you're doing. I wouldn't enjoy it, but I would do what you're doing.

[00:42:56] BL: I'm not saying it’s great.

[00:42:57] JM: Well, actually, you can basically use Kubernetes operators, right? That's kind of does what you need to do, right? Or do you even need that?

[00:43:05] BL: Yeah, we've seen like – What are some of these words? Helm. We’ve seen some Helm stuff. We've seen some Ansible stuff. Just various ways to – Cats.

[00:43:19] JM: My cats are fighting. I didn't want to – I need to put one of them in the room. But I can wait. Threw a box – I threw a box at one of them. I threw a box between them. I don't throw boxes at my animals. Just between them.

[00:43:32] BL: That's good. That's important.

[00:43:33] JM: You probably do the same with your kids.

[00:43:34] BL: It's a single tenant architecture between the cats. Something like that.

[00:43:41] JM: Maybe this is not a question you can answer diplomatically, but what is the hardest thing to integrate with? Is it Salesforce just by virtue of the complexity of the schema?

[00:43:51] BL: Well, the schema is kind of complicated in Salesforce, because things like transition between these objects, that's kind of – Like people generally have like a lead, and then they turn it into a contact, which somehow makes an opportunity. Like there're all kinds of things going on there. And so we can do an integration that's like helps you transition those, which is like much more advanced than like – It's actually easy to integrate with Salesforce because it's basically a database. So there're two answers to Salesforce specifically. Like one, it's one of the easier ones, because they don't have any major – You can ask it, it's types. And you can do all kinds of things and write into it. But then when we zoom out and try to solve the full issue and we talk about kind of a custom process of how they want to migrate between these objects, and then it gets a little weird. It's almost like a DBT transformation kind of ish problem, but in Salesforce. That's kind of weird.

The I did a blog post the other day of like who has the best rate limits? I kind of compared some that was kind of interesting, and it's very interesting. Basically, the more enterprise you are, the worst rate limits you have. Basically, highly encouraging people to do batching. And then the more consumer-ish or startup-ish you are, the more higher rate limits you have. And you don't even have these batch API's. So it's kind of pros and cons there. The integration with Intercom was probably the hardest, sort of the most exhausting to build that we've built so far, mainly because we have a full test suite and all this other stuff to make sure that we're doing all these things right. They've got a lag between when you put it in there and then when you can read it back out. So like maybe they're indexing it in Elasticsearch or something. I don't know. Like some sort of back – You write it. Then you read, it's not there. And then like some indeterminate amount of time, after that it is there.

And so our test suite, like to make sure that we get it right, like waits five minutes after every read or something. And it takes like all night to run, which, in general, was very frustrating. Now we record that. Basically, anytime we need to make changes to an Intercom, it's like an overnight process. Just kind of sad.

[00:46:13] JM: Alright, well, I'm running up on time, but we could definitely do another show at some point in the future. This is really interesting. Last closing – Actually, I was going to ask you first, what's the deal with the paintings in the background? Pretty cool paintings.

[00:46:27] BL: Yeah. I'm in my literal Silicon Valley garage right now. And there's like these paneling from whenever this house was built, 1930s. And I got feedback that it looked like I was in a shipping container. And so I dug these paintings. My mom moved, gave me this one. Like this one we had up in our living room a long time ago. And I just kind of kind of put some trees up. I thought it would help.

[00:46:51] JM: Okay, well, good answer. So final real question. If you were not building Grouparoo, what would you be building right now?

[00:46:59] BL: I had a long list of things as I was thinking about what was next for me, and all based on the pain I experienced and things like that were still unsolved. At the top of that list, under Grouparoo, was something that turned out to look a lot like Retool. I think what they're working on is a super interesting idea. Maybe it's a trend you're seeing, but like ways to do things that engineering aren't that excited about but add a lot of value to the business. Admin tooling was something I was super passionate about, making more efficient, and data movement as well.

[00:47:39] JM: So that's super interesting. So the whole low-code space, if I had to name the two most interesting areas in software engineering, I might say, depending on the day, data engineering and low-code. And actually, I'm going to ask you the real final question. What do you think of these low-code data engineering platforms?

[00:47:58] BL: I think I am a low-code did engineering platform.

[00:48:00] JM: Yeah, I guess you are. Kind of, but not like – So interviewed one called prophecy.io. There's one that Matrix invested in I interviewed. There're a number of these ones. There're a number of these ones. So I don't see your platform like drawing boxes and arrows between a bunch of stuff, right?

[00:48:21] BL: Sure. It's not a – Well, I mean, to some degree, I was going to say it's not like a generic wiring things up. But, I guess, to some degree it is. Like you're defining something in a box. And then it's got one arrow, big arrow, mapping that to this other thing. In general, I'm bullish on low-code in particular. It's something we've invested a lot in is the software engineering workflow being brought to the data into this data space. And so when you – A way to do Grouparoo is to define all of these things in code, in Git, in JSON, basically. And then that allows you to review them, and test them, and have staging environments and all that other stuff, which is a progression in general the data space is doing. And so that's what I call low-code. That still enables that workflow, which I think is super, super valuable.

[00:49:23] JM: Okay, real, real final, final question. Could you do open source Webflow, or open source low-code whatever? Open source Bubble, open source Webflow, open source Retool? And why wouldn't you? Or why wouldn't Retool be open source?

[00:49:39] BL: I've seen an open source Retool recently actually on Hacker News. Didn't stick with me, but I definitely saw one or two maybe even. My mind is a little attuned to that space as I was thinking a lot about it before.

[00:49:51] JM: I mean, again, this is the same thing. Why on earth – if you're an integrations company, why on earth would you be closed source? It's like are you digging your own grave? What are you doing?

[00:50:00] BL: I think it's an interesting that Retool put out like some sort of GitHub self-hosted something or another recently. I find that very interesting. They're trying to solve that go to market privacy thing, which is pretty interesting. But I think in the fullness of time, like there won't even open source databases, whatever, 20, 30 years ago, right? And now they are.

[00:50:21] JM: Yeah. But now, it's 2021. Can we like fast forward to the end of the movie yet?

[00:50:26] BL: You got to go through that hero's journey, my friend. I think in the fullness of time, we'll see all these things. We'll see it all.

[00:50:34] JM: Well, on that note, pleasure talking to you. You want to plug the conference one more time? What's the conference?

[00:50:38] BL: Yep, Open Source Datastack Conference, September 28th to 30th, opensourcedatastack.com. The website is open source. You'll like that. There you go.

[00:50:49] JM: Groundbreaking. Well, Brian, thanks for coming to the show. It's been a real pleasure.

[00:50:53] BL: Yeah, nice to talk to you.

[END]