SACRA

EXPERT INTERVIEW

UPDATED
12/06/2021

Sean Lynch, co-founder of Census, on reverse ETL's role in the modern data stack

TEAM

Jan-Erik Asplund
Co-Founder
jan@sacra.com

DISCLAIMERS

This report is for information purposes only and is not to be used or considered as an offer or the solicitation of an offer to sell or to buy or subscribe for securities or other financial instruments. Nothing in this report constitutes investment, legal, accounting or tax advice or a representation that any investment or strategy is suitable or appropriate to your individual circumstances or otherwise constitutes a personal trade recommendation to you.

This research report has been prepared solely by Sacra and should not be considered a product of any person or entity that makes such report available, if any.

Information and opinions presented in the sections of the report were obtained or derived from sources Sacra believes are reliable, but Sacra makes no representation as to their accuracy or completeness. Past performance should not be taken as an indication or guarantee of future performance, and no representation or warranty, express or implied, is made regarding future performance. Information, opinions and estimates contained in this report reflect a determination at its original date of publication by Sacra and are subject to change without notice.

Sacra accepts no liability for loss arising from the use of the material presented in this report, except that this exclusion of liability does not apply to the extent that liability arises under specific statutes or regulations applicable to Sacra. Sacra may have issued, and may in the future issue, other reports that are inconsistent with, and reach different conclusions from, the information presented in this report. Those reports reflect different assumptions, views and analytical methods of the analysts who prepared them and Sacra is under no obligation to ensure that such other reports are brought to the attention of any recipient of this report.

Published on **Dec 06th, 2021**

Sean Lynch, co-founder of Census, on reverse ETL's role in the modern data stack

By **Jan-Erik Asplund**

Background

Sean Lynch is the co-founder of Census. We talked with Sean about the operational analytics necessary to build a product-led company, where reverse ETL fits into modern data stack, and why the data warehouse is becoming the new center of gravity in the organization—and why that threatens the moats of companies like Salesforce.

Interview

Let's start with some context. How did the modern data stack come to be?

The high-level story of Census is that we are a data integration platform. We talk about it in terms of operational analytics; other people in the industry call it "reverse ETL" -- that seems to be the accepted standard term. Both are fancy ways of saying "doing things with data," as opposed to just reading and understanding data. It's taking action on data. Our twist relative

to past generations of integration platforms is that we sit as a layer on top of a data warehouse. So to understand where Census fits, it does make sense to understand the modern data stack as a concept, what that is, and then reverse ETL, or data operationalization, on top of that.
A lot of the modern data stack started with the creation of cloud data warehouses. Amazon Redshift was the big early leader. They launched in 2013; I think they did a private preview in 2012. Snowflake originally launched in 2014, so they're in a sense an overnight success ten years in the making, or a little less than that. Databricks was founded in 2013. Google claims Google BigQuery got started earlier than all of those.

Ultimately this trend around modern data stack started with the shift towards cloud data warehouses. Then you started seeing some of the other pieces that fit into that. Fivetran, for example, had been going for a couple years but pivoted into data connectors in 2014. Fishtown -- which is now Dbt Labs, the creators of DBT -- launched their first version in 2016.

Those are the component pieces when we talk about the modern data stack. They are all, frankly, relatively new. We're talking about an eight year-ish timeframe at this point. And it is an ecosystem all building around the warehouse. The warehouse is the focal point.

Over those eight years, the warehouse has gotten a lot faster. They've had competition with Databricks and Snowflake competing against Amazon and Google. They've become a lot cheaper. They've been designed to support more and more of the data use cases that, when they started, were out of their scope. When they started, they were really, "Hey, this is a cloud version of the data warehouse infrastructure that you might be installing in your data center." That's the Teradata and Netezza type of product. This was the cloud version of it, and it happened to be scalable and cheaper, but it was not necessarily changing the story in terms of use cases.

What changed is that, in addition to becoming super scalable and cheaper, they also started to shift the story of the types of data you could throw at it, what speed, over what timeframe. I like to think about the original term “Data Warehouse” as the Indiana Jones dusty old warehouse where artifacts go to die. Today, the warehouse has evolved into the Tesla manufacturing plant version, where it's shiny and new, things

are moving through it at incredible speed. That's the way that the warehouse has evolved. We -- and Fivetran, dbt, and now a big ecosystem of other products around quality and discoverability, etc -- are all building on top of this very modern manufacturing pipeline, as opposed to the old storage way of thinking about warehouses. That's the core of the modern data stack.
At Census, we think about what we're building as creating a feedback loop using that stack. As a business, you have your applications. You have something like a Fivetran that's ingesting all that data into your data warehouse -- what previously would've been called ETL and is now ELT. You have dbt doing the "T," the transform in your data warehouse. And Census comes along in the last step to provide that return path back to all those apps, to help companies actually do things with those data.

How would you contextualize CDPs like Segment? How does the decision get made -- if there is a decision -- between using something like Census to sync up data and sending it via Segment?

Lets talk about CDPs separately from Segment specifically, because Segment does a bunch of things. They do market themselves as a CDP, but I think they do more than just that.

CDP stands for "customer data platform." At the 5,000-foot level, if you look at a CDP's architecture diagram and what I just described with the modern data stack that Census, Fivetran and dbt -- these sorts of companies -- are building, they look very similar. You have big sets of users/customers sitting in apps, and I'm pulling that data into some sort of central place. I'm usually doing some amount of merging, cleaning or associating all of that data together. It's kind of the "transform" piece. ThenI publish those data points off to destinations.That's a CDP roughly.

CDP tends to serve more of a marketing and advertising market for the most part, or at least that's where most of the traction has been. The CDP pulls in all my various data sets about customers and generates audiences for it that I can use and target.

CDPs historically have also usually been an all-in-one solution. You buy the product; it does all the data ingestion; in a lot of cases, it does the cleaning, the merging and the association based on its model, or it does it automatically for you; and then it does the deployment off to the various different destinations that it supports. There are benefits to having that all-in-one tool.

The problem comes in when you're worried about extensibility and flexibility. Most of the CDPs try to standardize on their view of the model. I'm personally most familiar with Segment, so I'll pick on Segment just a little bit. Segment's model is really around the .identify() and the .group() calls, and you're standardizing to those data structures. There are other CDPs on the market that offer other models. That works great if your business fits into that CDP's model.

But, for example, if you need a data connector that they don't necessarily support, or if you need to structure your business entities differently from the default one in those objects, then you start running into problems. You would expect these things to follow typical power laws. Everybody needs a Google Ads connector on the marketing side; everybody's going to sync data off to Google Ads. But maybe not everybody is syncing data to TikTok or something like that. Does your CDP support TikTok? It may or may not. Does your CDP support a consumer e-commerce use case? You're probably fine. Does it support a three-sided marketplace where you want to target the different entities differently? That might be a bit more of a stretch. That's where you start to run into some of the problems with a classic CDP. It's great as long as you fit into that ecosystem. But if anything you want to do works outside of that ecosystem, then you start to get in trouble.

At Census, we talk a lot about: "Hey, you don't need a CDP. Your warehouse is your CDP. It's whatever DP you want, it's your data platform." Through the modern data stack, you can assemble your CDP that works for your business in whatever way you need it to. You can pull in data from any source with Fivetran. If, for whatever reason, Fivetran doesn't have it, you can use Segment instead. You can use Stitch, or you can build your own. You can do the modeling in whatever form you want -- with dbt probably, but there are other options out there. And then you can broadcast that data out to the destinations that matter through Census, or by yourself. Before they were using Census, a lot of our customers were building their own integrations into these destination tools, pulling it out of data warehouse.

Ultimately, all of these things sitting on the data warehouse are just speaking SQL. The interface that they talk through is SQL queries and common table expressions. So Fivetran can work with dbt, Census, Segment, DataRobot, or whatever you want to get into the AI and ML side. You can stitch together all the different things that are most appropriate for your business.

One of the places where you run into unique needs for connecting data beyond Google Ads is in the product-led growth world. I'd love to dive into the use case of PLG companies selling into the enterprise better by bringing data into their tools. What have you seen for a typical sales motion? What kinds of data seem most useful and why?

PLG companies are where we started. Figma was one of our very first customers, so we worked very closely with them in the early days. Actually, some of the ideas behind Census were informed by things that we had to build custom for Dropbox in a past life, as well. We saw these similar problems.

The interesting twist with PLG is that you're giving anybody in the business decision making power over what software to use. Another label for product-led growth is bottoms-up adoption or bottoms-up growth, where the decision makers are not necessarily your CIO or the IT department anymore. It's maybe an individual, maybe a manager of a team, maybe a division. It's the people who are actually going to be using the software. That means that the old sales motion of enterprise software, where you try to find the most senior person in an organization and start sending them lots of email and try to get them interested in your particular piece of software -- even if, let's say, you're friends with them -- they're probably going to tell you, "Yeah, I think somebody in the company probably cares about that. Maybe I can go find them for you." Despite being in the C-suite, they're not necessarily going to be involved in that decision at all.

The flip that happens in these PLG companies is that I don't necessarily know whom to go prospect or reach out to. I can start doing more marketing, but that's obviously less of what the sales team is interested in doing. This is the product-led part of it: people signing up because they saw this app on Product Hunt or a friend at another company in a similar role is recommending it. They're going to try it for themselves. They might start using it themselves; maybe they're using it with a small team; they share it with other people in the company. It grows from there. You see this organic spread of adoption inside companies. But oftimes that adoption is piecemeal. You have a couple people here and there, you have small groups of teams. You might have people who signed up with their personal email addresses, not their work email addresses.

We see a lot of companies using something like Census to take a look at all of the adoption that's happening in their product today. What a lot of the successful PLG companies are doing is trying to take this very amorphous, loosely defined usage data inside a product and collapse it together into signals that say, "Oh, you should go take a look at Disney, because it turns out that there's a bunch of new use inside Disney. There are these teams that are particularly active." Maybe you want to go reach out to somebody that's got manager in their job title but is not necessarily "head of" or CIO.

More often than not, though, the signals that are really interesting are, "Who's the person that's inviting everybody? Who's the super spreader of that application?" Chances are, that's the person who's your biggest champion inside the company. Looking for those sorts of product signals -- the people who are really engaged, the people who are sharing the product with other people in the team -- and pulling those together, especially across different teams and different organizations, a salesperson then can go in and use that information to say, "I should reach out to this person, that person." This could also complement the old school way of doing things. If you're still doing the standard enterprise sales practice, you could reach out to the CIO and say, "Hey, did you know there's this huge new growth of usage organically inside the product? You probably want to have an official sponsored version, maybe you want to turn on your single sign-on solution and everybody who was using this organically has to use it."

The types of signals that we typically see sales teams using are usage metrics, like who is adopting, who is there on a regular basis, and the team-oriented usage metrics -- are they inviting, are they sharing, are they taking those critical actions for the product. Then there's the temporal aspect: have they been doing that in the last couple of weeks? If they used your product a year ago and engagement is flat or hasn't happened

since, that's going to be a lot less interesting for an upsell. But if you are talking to a team where, "Oh, it looks like they're suddenly using my virtual whiteboard solution at an event," this might be the moment to reach out and talk with them.
Part of the real selling point for us with this operational analytics push is this data is now available near time, almost instantly. You have sales teams that wake up in the morning and are responsive to, "Turns out that Disney is using a lot more of our product now than they were a couple of days ago or last week. Now is an interesting time to strike."

On the flip side, you talked about how people can use their tools differently. Have you seen these various kinds of destination apps evolve to better use this data and add value on top of it? For example, surfacing that this person has used your app or used a certain feature more often recently.

We're seeing a little bit of that now. Some of the destinations that we're working with are more responsive to our worldview than others. We're helping them improve APIs, like Braze and Front. One of the companies that we do a lot of back and forth with in terms of API design is Mixpanel. You wouldn't historically think of Mixpanel as one of these data silos that would be separate from the rest of the data infrastructure. They're really leaning into, "How do we have Mixpanel work really well with companies that are building up these modern data stacks?" We're working with them on that.

One of the biggest changes that we're making there is handling larger data volumes and more frequent data. One of the differences of a reverse ETL tool typically is that we batch sync out large sets of data -- hundreds of millions of records multiple times a day -- because we have a lot more data about what is changing from all these different sources and we're sending it off to these destinations. A big API shift that we're seeing is just handling much more data volume from services like ours.

On the other side, it's interesting. Our very first destination here was Salesforce. Salesforce obviously is giant, so we have very little influence on what their API is doing, but if you look at it from a product perspective, they definitely know which way the winds are heading. You can see it with products like Einstein. Einstein is very interesting. It's like their AI layer

inside Salesforce, but it needs to be driven by good, up-to-date, frequently updated data. The best way to provide that isn't to have your sales reps manually entering those things when they get off the phone with a customer once a week. It's to have that coming in live from whatever sources you have in the product, marketing pages, whatever.
That unlocks the real value of a lot of those AI types of layers and features. I see that also where some of these products have been built by product managers trying to launch AI buzzword things, and they get to the rubber meets the road aspect of, "Oh, turns out in order for us to have anything valuable here we need to have data. How can we get more data here quite quickly?"

It seems that the center of the organization is shifting from the CRM or ERP to the data warehouse, with more and more data going directly into the data warehouse instead of a Salesforce, a HubSpot or an ERP. I'm curious for your thoughts on that. Do you think there is a risk to the moat that those kinds of companies have built?

This is a whole discussion on its own. From my perspective, Salesforce has the most to lose from this shift for the reasons that you mentioned. This idea of having a source of truth for customers is something that most companies want and have been trying to figure out, and if anybody has gotten close to it, it's Salesforce, largely by virtue of their ecosystem. They've been large enough, they've been core enough, and then they built up this platform that the Marketos and the Outreaches of the world -- all of that sales and marketing tool infrastructure -- all have Salesforce integrations. A lot of times, they don't have integrations with anything else. It's really Salesforce or bust. We've all heard, "Which CRM are you going to use? Because if it's not Salesforce, you're eventually going to switch to it, so you might as well just use Salesforce." I won't debate that. As far as CRM primacy, Salesforce is doing great.

But as far as that source of truth for the state of the customer, I do think that they have the most to lose in terms of their status. I think that they also are the most aware of it. They bought Tableau, they're investors in Snowflake. They're very aware and plugged into this ecosystem. I don't think they're naïve or blind to it, but in my mind, they were the closest thing to this. A modern data stack -- this sort of hub around the warehouse -- really has the most potential to affect them.

We're already starting to see that. Over the last couple of months, we've seen an explosion of PLG-focused CRMs, and companies like Calixa and Endgame have been getting a ton of excitement, I think partially for the same reasons. They've been able to get access to a ton of data about customers that has historically been stuck in the Salesforce ecosystem. Now that's more generally available, they can make a play at those very specific workflows and start to offer, "If you're doing sales in a PLG world, do you really need Salesforce? You could really just use us." You're starting to see bits and pieces of that now.

Zapier is at the other end of the spectrum from Segment: an ultra-simple, easy to use, no-code tool for moving data around, but without needing engineering resources. From your perspective, what kinds of teams and culture are a good fit for Census over something like Zapier?

I am a user of Zapier. There is a very nice intellectual model for what you can call the "if this, then that" type of "an event happened here, so that event has this side effect and it goes there." In a world where you have a relatively small number of "if this, then that"s to manage, that's pretty straightforward. You can rationalize that relatively reasonably.

The place where this starts to fall apart is when you scale that, when you take that model and you go from one connector to dozens -- hundreds in some of the deployments -- and you're trying to rationalize "if this event fires in Intercom and sends this to Segment, which then is going to put something in an Airtable and so on and so forth." It becomes really hard to reason about what is being sent to where and transformed and that sort of thing.

One of the benefits of Census and the modern data stack approach is you can start to standardize those definitions in one spot. The definition of whether a customer is a VIP or how you calculate MRR -- you can define those once and you can reuse them, not only on the taking action sales motion side of things but also on the analytics side. So the report that's going to the board, the CEO and the CFO can use the same MRR definition that the marketing team is using to classify or target and that the product team is using to rank product features. You can share that definition in one spot, and you can change it in one spot. The way that Census models work, we keep all

the destinations that are using it in sync. All the numbers change everywhere by defining it in one spot.
It's a model that simplifies complexity. If you're running your own coffee shop and you just need to make sure that a welcome email gets sent when somebody signs up on your Square, I think Zapier is a great solution. I don't mean to trivialize that. If you're a small-scale software company, Zapier can get you going really quickly. It's just that, when it comes to scaling, we've seen over and over again customers that said, "I can't reason about this. I can't debug this. Events are getting dropped. I don't know who owns this. What's happening here?" Centralizing that into one spot is a way to bring sanity to the chaos.

What does the typical Census configuration look like?

A typical Census deployment is usually into a modern data stack. We are sitting on top of a data warehouse. It's usually one of the big three: Redshift, Snowflake, BigQuery. There's usually something already in place where data is loaded into the data warehouse, and there's some amount of transform, cleaning or manipulation of that data going on.

There's usually also a data team in play, though I would say that's starting to shift a little bit. We're seeing the data warehouse increasingly being set up by marketing teams on their own. More often than not, if it's a marketing team, they're doing it with BigQuery -- I suspect it's like, "Oh, you're on Google Ads, are you using Google Analytics?" There's probably a "deploy BigQuery" button as part of that. It's also sometimes sales operations deciding, "Hey, these are the types of reporting I want to do, I can't do it entirely in Salesforce. I'm going to do it in something like a Mode, so let me put this into a data warehouse." But typically there's a data team and Census is sitting on top of that.

You can break down the types of data that we typically see companies working with in two axes. One is in terms of use case, and the most common use cases for us are along a customer journey. That's getting back to the CDP aspect of things, so advertising, marketing, SMS marketing, etc. that you hook up with Google, Facebook, and so on. On the marketing side, it's the new school ones like Braze and Klaviyo, as well as some of the old school ones like Marketo, Pardot, that sort of thing. For sales, Salesforce is our biggest destination.

Support, customer success, even through to the finance and billing side of things with a Stripe or NetSuite. So one axis of the types of use cases are around the different teams, destination tools or customer facing functions. Usually they're custom more than anything else.

The other interesting axis to look at is types of data. When we started, we were really focused on syncing what we call "profile data." This is the profile of a user or the profile of a company. You may have other profiles or objects and nouns in your business model that matter. A common one in PLG is the concept of a team or a workspace, which is different from a company. For example, Disney may have many workspaces, and many of those may exist independently from each other because different people set up workspaces over time. Understanding that Disney as a company, from a sales perspective, is different from the dozen different workspaces that exist separately is a really valuable insight.

Getting back to your earlier question about one of the things that are changing a lot of these tools, we're increasingly seeing tools support the concept of custom objects. So not just their core "whatever their equivalent of a company is" -- an account or an organization or a business -- and not just the person or lead contact, but also custom objects. Salesforce has done this forever, but now HubSpot, Customer.io and Marketo do this too. We're seeing services that are offering this popping up. The use case for us is representing those related auxiliary objects: workspace is one we see over and over again, subscriptions, that sort of thing -- objects that somebody might care about.

When we're talking about profile data, I would say that the use cases there tend to be a little bit more of "a human is looking at a tool that is seeing that profile." In a CRM, I'm going to go look at a company and at the context. That's true in the customer success case: "Hey, is this company a turn risk? I'm going to look at their usage profile." Support is the same thing; finance is as well to a certain extent.

The marketing and advertising side tends to be interested less in the specific instances of a profile and more in data in aggregate, in what you might call segments, lists or audiences. That's another type of data that we do a lot of work with, where you're going to define, for example, your VIP segment. In the past, the way you would do that is just by exporting a CSV of

those, or if you had the properties in those different tools, you could set up the segment definitions in them individually.
One thing that we see our customers doing more and more now is defining segments in one place in SQL. So I'll define what VIP looks like as a segment and get the list of all the user profiles that qualify for that segment, and then I'll sync that list definition into the various destination tools from Census.

The third category of data we see a lot of is event data. When we got started at Census, it didn't feel like a particularly good fit for the data warehouse, but we've seen this shift so much, especially over the last year. When we started out, we thought you're going to want something like a segment to pick up the event and pass it into the destination system as quickly as possible. More and more, though, we are seeing companies say, "I'm going to put my events through the data warehouse as well." This is a bit unintuitive, especially if you think of a warehouse as being slow and creaky. Why would you send events if it's going to take 24 hours for you to load them in there to do anything with them? Part of the reason is that you're going to put the events in the data warehouse anyway because you're going to use them for analysis to generate profiles and that sort of thing.

We're seeing a lot of customers generating different types of events they couldn't before. We call them "synthetic events." These might be events like, "Hey, a user made a purchase offline in a store that didn't generate an event on the iPhone app, but we do still want that to flow through the system as an event."

Another really interesting thing is enriched events, which means taking an event that's happening through the system, but adding more data in context that you have in the data warehouse at the time. One of the early versions of this was: "Hey, we're going to send a welcome email, but we want to send it from the sales rep that owns the account. While we have the user signing up event, we want to add the email address of our sales rep so that we can send a targeted email. And we want to do that joined data before that event propagates through." We're seeing these really interesting events where the data is still the event-based user token action -- this thing happened, a subscription expired, whatever, it's going to be consumed by a service that consumes event-based data -- but it's actually generated in the data warehouse.

The reason this was unintuitive at the start was because you didn't think warehouses were particularly fast. But one of the interesting things that's changed with these cloud warehouses over the last couple of years is they've gotten faster and faster in terms of indexing the data that's being inserted into them. There are still use cases where you really do want the "if this, then that" to say, "This thing happened, I fired a web book, another thing happened a sub-millisecond afterwards." If you're sending a password reset email, you don't want to wait that long. But we now are regularly seeing customers doing the feedback loop through the modern data stack in about 15 minutes, with data flowing into the data warehouse, being ingested and transformed with something like dbt and being sent back out.

How is the data warehouse evolving? Where do you think it's going?

The data warehouse again is really the focal point around a lot of these things. But the place that I am excited about and that we're watching is data warehouses getting even faster. We're now talking about data flowing through the system on the order of minutes. Snowflake has Snowplow, which is their streaming data ingestion. dbt is working with Materialize to do more dbt transforms on a real-time basis. I think this is the really interesting trend over the next one to two years -- or maybe two to three -- where the benefit for end users is they don't have to change anything again. This isn't suddenly, "I have to set up a separate Kafka-type pipeline to get all of the benefits of this." It can actually start to flow through one system. My analytics can get close to real time. My operational analytics from Census or my reverse ETL can get closer to real time.

That's the thing that I'm most excited about from data warehouses in the next two years. It feels like "warehouse" is the wrong word for that at this point: "warehouse" is volume-based and kind of sleepy, and now it's just becoming a lot faster paced.

So the idea is that, with live or much faster streaming, you can bring to many more companies the ability to do something that right now only the biggest tech companies with the most resources are capable of?

Yes, absolutely. The biggest companies can do this. You can do this. This is a benefit of that Zapier-based model -- Zapier, Segment and those sorts of tools are also event-based. Like, this action happens and you can mostly propagate into those tools in relatively real time. But a lot of what they're doing underneath the cover is setting up these stream-based systems. If you had asked me when Census got started a couple of years ago, I would've thought of those as being divergent systems, divergent use cases. Now when I look at the engineering roadmaps for the data warehouse world, I think, more and more, we're going to be looking at it as maybe two types of data, but all flowing through the same infrastructure. You won't -- either as a marketer or on the engineering team, or anywhere on that spectrum -- have to think about two different classes of infrastructure and the data that sits on it anymore. You can see it in one spot.

You were talking about cloud warehouses getting faster and faster, driving a lot of this adoption. If COVID was a tailwind in terms of people wanting to get this kind of data and engage with people based on it, are you seeing any other trends driving the increase in adoption recently?

Data warehouses all got started around 2012, 2013, 2014. The phrase "big data" peaked as a concept in 2014 and petered out from there. Partially because I think the generation of tools that were powering that -- like the Hadoops of the world -- everybody realized, "Oh, this is a huge pain in the butt to manage." Cloud data warehouses were giving you a lot of that scale but with a SQL interface, as opposed to having to learn a whole new processing language or that sort of thing.

I think that is a really interesting trend here. Redshift is ultimately a Postgres version or was started as a special extension to Postgres. Snowflake speaks a very well designed dialect of SQL. One of the things that the cloud data warehouses do, but don't necessarily get credit for, is go back to plain old SQL that everybody understood. So one of the interesting trends of describing adoption here is abandoning the sense that "I need to have big data in order to justify building this infrastructure."

We're seeing the modern data stack being adopted at much earlier startups now, because it's easier to turn these things on. For the most part, it's credit card swipes. You can plug them in, they all talk SQL to each other. You can mix and match them a little bit. It's a lot easier to walk up to a modern data stack than it was four, five or six years ago where it still probably involved a lot of, "Okay, I'm going to get it running, but then I've probably got the phrase EC2 in here somewhere, and I need to start managing some infrastructure in order to get this to do what I actually want."

A lot of this is becoming very turnkey now. Census is a part of that. Like I mentioned before, our biggest competitor is the engineering teams that were building custom integrations. The nice thing about that is they really hate maintaining those custom integrations. They're not big fans of the Salesforce API -- they have some nice things to say about it, but this is not their day job, and it's not what they want to spend their time doing.

We're taking some of the patterns that five years ago would have had to have been cobbled together through open-source projects and custom code, and now making them something that you don't need an engineer to get. You don't even need the engineer to do the work in Census. You can have your marketing team use it. You can have anybody use it. I think that's largely true for the modern data stack overall. It's not this "only big company" thing by virtue of the size of the data or the size of your data team. This is something that smaller companies or roles that are not necessarily data can get started using.

I'm curious to hear how your customer success operation has changed over time, and what it looks like now.

Our customer success operation is nascent -- we're still an early startup -- so there are probably some experts that might read this and would have a lot of very interesting feedback for us. I would say the biggest shift is that, when we got started, most of the people that we talked to thought we were crazy. Part of the reason was, there was a very linear idea of your data pipeline: the data gets loaded in and ends up in your data warehouse, you run some queries, and you build a chart on it. The idea of pulling data back out was a bit nonsensical. Like, we would have people say, "I'm not sure I want that. I don't really know, do I want to do it? Do I trust this as a data source?" A lot of our very early customer success efforts were explaining to people partially the modern data stack itself, but also why it is valuable, why they should think about this as a

feedback loop, as opposed to a very linear start-to-finish, left-to-right pipeline.
That has changed, to a certain extent. I think we're still relatively early in the modern data stack adoption curve. Most of the people that are developing this space maybe spent some time in the DDD Slack, and they're definitely more on the early adopter end of the curve here. But even just over the last year, we've seen more enterprises talking about this. We see a lot more mid-markets. We see interesting companies that are not the leading or the hottest startups on TechCrunch adopting modern data stacks. Their version might be a little bit different. We start to see a bit more Azure as part of these conversations, MySQL, that sort of thing. Maybe we'll get into conversations about on-prem. But they're shifting. They're starting to think in this way. They may be switching to a Snowflake as part of this, and they're adopting more of a cloud mentality.

In some cases, companies that have been around for decades are realizing that they actually may have a leg up on the disruptive startup that's trying to come into their space, because they have access to huge troves of data. If they can go add a couple of data resources to their team, they can suddenly compete in a way that those startups cannot.

I'm spouting words that are just like, "Yeah, we're moving up the adoption curve." But I think that's a lot of what we see day to day in our customer success. We do a little less explaining of what the modern data stack is and why it's not crazy to do this, and a little more of, "Here are all the benefits that you can get." And we are hearing a lot more, "Aha," or even, "Oh, we know. Just help us get it done."

How does Census monetize?

It's a straightforward SaaS-based subscription. We've done a couple of different iterations of it in the past. It's largely based on the scope of customers' use cases. We use a couple different metrics as a proxy for that, but really what we're trying to do is understand how many different teams we impact as part of the integration, so it loosely scales with your usage of additional destinations. It's not explicitly, "Hey, if you're opting into Salesforce, then it's plus one destination"; it's based on the number of fields that we're updating. But you can think of it as: the more destinations, the larger the package. Then there

are more professional services, like enterprise logging, SSO auditing, etc., that you get in higher tiers. It's a fairly standard SaaS-space subscription model.
How is increased customer success reflected in Census's revenue and success?

It's largely a land-and-expand type model. We typically deploy with a particular kind of use case or team in mind. We start working with, say, the product analytics team, sales, customer success or marketing, and we can prove out our value for one of those connections. Then once we're in, we can expand to different functions of the organization. So we'll start with sales and then move on to marketing and customer success and expand from there. As the success of the customer increases, we see more usage for a particular team, as well as expansion into additional teams.

Bigger picture, I'm curious to know more about the shared vision between you at Census and your investors -- a16z, Sequoia -- about the future of the data stack and Census's role in it.

We started working with a16z fairly early. We knew some of the folks over there, so were having conversations with them back in the early days. From their perspective -- and this was echoed by Sequoia as well -- this was 2018 or 2019, and they were seeing the emergence of the modern data stack as a trend. It's interesting because, for me, this is the first time that I've watched a category creation from the front lines. We were starting to write blog posts about the modern data stack in 2019, but in a lot of ways it was a term that was still relatively new. And the reason we were writing blog posts was to try to codify that a little bit as a term that the industry could group around.

It's an interesting flywheel when I think about category creation now. What we saw happen around the phrase "modern data stack," you could kind of see happen around -- the most recent example in my mind is "serverless." And it's happening now with the term "web3." You have this interesting confluence of practitioners, investors, companies in this space that are all talking about this term, trying to define exactly what it is, writing these thought pieces. To a lesser extent, that also happened with "reverse ETL" as a category over the last year, where VCs have come out with their explanation of, "This is

where it is. This is where it fits in the map." We've obviously been telling that story as well.
When we were getting started, there wasn't really any category for reverse ETL. We've already discussed this -- when we were talking with customers, they were looking at us like we were crazy. But we were affixing ourselves to the modern data stack, and Andreessen Horowitz and Sequoia were very interested in this emerging modern data stack. Obviously, they had lots of investments in Databricks and Snowflake, and those were growing like gangbusters. To a certain extent, Fivetran was already there, and in 2018 dbt was just starting to get onto people's radar as well. So they were seeing this as an emerging ecosystem with potentially a lot of different products in the space. I don't think they were necessarily looking at it as a Salesforce disruptor in the way that we talked about it previously. But that was our pitch, when we started to say, "Hey, look, you can use the modern data stack for this."

From their perspective, it was an interesting blend where we were both talking about the modern data stack. They're seeing it as a developing trend and growing. They're seeing the adoption of the base component pieces, and now they're writing the speculative: "Where does it go next? What does this enable in terms of use cases?" And we were coming at them from a use case. We were saying, "Hey, this is how one can leverage a modern data stack." Similarly, I think all of the AI companies were giving a very similar pitch: "You need all of the data. We're going to layer on top of the data. We're going to generate insights and that sort of thing." So our shared vision was around the modern data stack. I think they were looking at it from a perspective of: "This trend is happening. We've invested in the base levels of this infrastructure stack. Where is it going to go from here?"

Reverse ETL ended up being one of those categories that they were interested in, but at the time we were pitching them on, "This should be a component on top of the modern data stack," and that wasn't necessarily part of their vision. Now it very clearly is. We unified through that growth of the modern data stack, and we were able to speak the same language there.

The whole concept of category creation is very fascinating. To watch it play out has been an incredible experience. I'm fascinated in that as a trend, how long that lasts, how long you can continue to ride that wave. How do you propagate it?

I'm interested to hear what you think about reverse ETL and how it's taken off. There's still some confusion around, "Isn't it just ETL backwards?"

We like to talk about it in terms of operational analytics. One of our original pitches was, "This is the second use case of analytics. The first one was looking at the reports, and this is doing things with them. It's closing the loop, it's action-oriented." I think RedPoint may have coined the term "reverse ETL" originally, or at least that may have been where it really started to take off. In the early days, when we were just getting Census off the ground, to help explain it to a data person, we would describe it at the end of the call like a Hail Mary description: "It's Fivetran in reverse." And they'd be like, "Oh, I get it now. I understand."

One of the interesting things that plays out as part of this is: is it just ETL? Fivetran is talking to Salesforce and Snowflake; Census is talking to Snowflake and Salesforce. How fundamentally different is it? If you dive into the details of how the connectors work, there are more fundamental differences, but you have to get to the level of almost engineering the thing to understand that. So I think part of the confusion around reverse ETL is that, in a lot of ways, we're still talking about integrations as a category. As this space matures, I think we will do a better job of working with Fivetran and explaining how these two are complementary and how they work together. But the world is such a Wild West right now. There's so much to build here that we haven't quite gotten to that point yet.

I am very interested in how this evolves in complement to all the other pieces of the modern data stack: data observability, data quality, all of that also has interesting overlaps with this as well. I wish I had answers and a clearer picture of where this is going, but there's a lot happening all at once.

When we first got started, it was just us trying to convince people that we weren't crazy. And now there are actually people in this space. We have competitors that also market themselves as reverse ETL. It's a really interesting experience because, without them, there wouldn't be a reverse ETL category. If it was just us in this space, it would just be Census. You wouldn't talk about it as a space. It seems like you almost need competitors to have category creation. You need people to be building different takes on the same thing.

Because the reverse ETL category is functionally less than a year old, the definition of category and the competitors in the space are still taking different approaches. We're still trying to figure out what is in bounds and what isn't. I don't know when we formalize: "This is the definition of reverse ETL, and these are the things that are not reverse ETL but are like iPaaS or no-code." We're still shaking those things up.

How big do you think the market is, and how are you thinking about Census's TAM and future growth?

When we started, it was very much a focus on product-led growth. Obviously, that has seen its own wave that we've ridden. We've had a lot of those customers grow over the pandemic and we've been able to grow with them, which has been fantastic. It's worked out really well for us. But that's not our TAM.

For us, in terms of target customers, we really think any company that is generating data is a potential target customer. That includes a lot of companies that are not necessarily on the modern data stack at this point. In terms of staging, we started with these early-stage product-led growth companies that were already early adopters of the modern data stack, so there was high alignment there. Now we're expanding to a larger set of people who are still building on loosely a data stack type of model. That's e-commerce companies, marketplaces, and certainly more interesting and less traditional companies as well -- a little bit more fintech or, just generally speaking, finance, where they still have large data sets. It's maybe less the "modern data stack" and instead just "data stack."

Beyond that, the longer-term category is people who are not necessarily on a data stack yet. We think the hub-and-spoke model that we described earlier is still the right view for them. When we talk about TAM there, the space of all integration services -- and this was maybe two years ago -- was about $30 billion. I think that is probably underselling it, just in terms of the proliferation of SaaS now and especially over the last couple of years, and watching the number of early-stage companies that are building really sophisticated business and operations on top of no-code types of tools.

I think that is the future. I continue to be humbled by how much SaaS exists in the world and how much demand there is for SaaS for various workflows. It doesn't seem like we are at all moving towards a consolidation at this point. It seems like the path forward is for companies to have their various point solutions work better and better together. From an integration standpoint, even just building this company, we've been like, "Wow, there's a half dozen other spinoffs that we could potentially build that are purely around how to make software integrate better with each other." There's still so much to do in that space. So that $30 billion number is the type of Forrester number that we can throw around as part of an investor deck, but I think that the space is still very, very early days.

One interesting space you mentioned was helping more established companies -- old incumbents, traditional players in industries -- compete with potentially disruptive startups. Is that a big one that you think about?

Absolutely. All those companies are now talking the language of digital transformation. Implementing it -- and how it's actually implemented -- seems to be in fits and starts. It can still be some VP or CIO that's trying to bake their career on, "We're going to do this with a large organization or subdivision." In theory, a large percentage of the software being built now is much more focused on letting a team or a division or an organization do that, as opposed to the overall company. That should be very possible. The whole product-led growth angle should still be complementary to individual teams adopting it.

The only cases where it's not are when there are extremely hard restrictions that place headwinds on anybody adopting it. But more and more, there seems to be some recognition that individual teams can operate slightly differently on different tools. Especially with the integration platforms, that sort of thing. There's an easier story to say, like, "Hey, we're going to adopt our particular version of our CRM here. It's going to be different for the rest of the organization, but they can still talk to each other through a modern data stack type of approach." I can literally turn on my own BigQuery database, use it for my purposes, start using it for our marketing team, and it can slowly expand and get adopted over time.

I think that you will see more teams internally starting to adopt piecemeal software, which then means that this integration story needs to happen over and over again -- not even between the different tools that one company is using, but the different tools that subdivisions are seeing.

Disclaimers

This transcript is for information purposes only and does not constitute advice of any type or trade recommendation and should not form the basis of any investment decision. Sacra accepts no liability for the transcript or for any errors, omissions or inaccuracies in respect of it. The views of the experts expressed in the transcript are those of the experts and they are not endorsed by, nor do they represent the opinion of Sacra. Sacra reserves all copyright, intellectual property rights in the transcript. Any modification, copying, displaying, distributing, transmitting, publishing, licensing, creating derivative works from, or selling any transcript is strictly prohibited.