

EPISODE 1160

[INTRODUCTION]

[00:00:00] JM: A customer data platform such as Segment allows developers to build analytics and workflows around customer data such as purchases, clicks and other interactions. These customer data platforms, otherwise known as CDPs, are often tightly coupled to an underlying data warehouse technology. High Touch is a platform that provides an unbundled CDP, a platform that sits on top of your own data warehouse. The High Touch team joins the show to talk about what they're building and the CDP ecosystem as a whole.

[INTERVIEW]

[00:00:37] JM: Guys, welcome to Software Engineering Daily.

[00:00:40] S1: Thanks.

[00:00:40] S2: Thanks.

[00:00:42] JM: You all came from Segment. Explain what you learned from Segment and how those insights led to you starting High Touch.

[00:00:49] S2: Yeah. So basically, Josh and I used to work at Segment. And Segment is essentially a platform that helps people connect an event stream. Connect and capture an event stream to a bunch of different destinations. For example, data warehouses and marketing tools and analytics platforms all through one provider. What we learned at Segment was that the data warehousing space was very much rapidly growing. From the time we joined Segment to the time we left Segment, it was it was magnitudes bigger. Many new entrants became mainstream providers, like BigQuery and Snowflake. And now what we basically learned is that every vendor in the analytics and marketing space is writing data into the warehouse, and it's becoming the central place where companies are collecting all their customer information. So that's how we transitioned into High Touch, which is basically a platform that helps you activate all the customer data in your warehouse to do marketing and different business operations without knowing SQL.

[00:01:43] JM: What does that mean? What is an activation?

[00:01:45] S2: Yeah. So activation and marketing is basically the idea of taking customer data and using it for different marketing campaigns or marketing purposes. So this might be taking your whole customer base, finding a list of customers that's a subset of it relevant to a certain marketing campaign and then running a targeted campaign on an email platform or an ad platform. And basically High Touch enables companies to do this on top of all the data in their data warehouse no matter where it's from, whether it's Segment, or Google Analytics, or their own data engineering, or Fivetran, or anything like that.

[00:02:15] JM: So why isn't this problem solved by Segment itself?

[00:02:19] S2: Yep. So Segment itself basically focuses – Primarily the bread and butter of their platform is on the data collection piece, which basically allows your company to create an event stream and pipe it to a bunch of different services. They also have some products that are higher level offerings on top of Segment that allow marketers to do some of this activation piece as well. Basically what we found is that all companies are centralizing their data in a data warehouse. So we want to operate at that layer, because we believe that the problem of getting data into your warehouse is pretty much solved at this point by Segment and a variety of vendors. The top Segment customers are loading data into the warehouse via their own data engineering, services like Fivetran and Stitch, as well as Segment. So we believe that's the ideal layers to that to help companies actually activate that data.

[00:03:05] JM: So the ideal world from your point of view, the customer already has all of their data in the data warehouse and you do stuff on top of that.

[00:03:12] S2: Exactly. Yeah. So we don't focus on data collection at all. We're basically trying to serve companies who are already loading all their data into the data warehouse, which is the new trend both by companies in-house as well as like using vendors.

[00:03:25] JM: So what does that mean? What kinds of things can you build on top of a direct data warehouse integration?

[00:03:30] S2: Yeah. So with High Touch, there're basically two core components. So at a high-level what we do is we take the data warehouse and we let you push it, push data from there to tools that your business teams actually operate in. So the first idea is High Touch Connect, which basically allows you to enter a SQL query into our UI and connect and pull some data out of your data warehouse and then send it to a bunch of different business tools. So these can be CRMs, like Salesforce or HubSpot, as well as marketing platforms like Marketo or Iterable, as well as even support platforms like Zendesk. So if you want some custom data about your users in those platforms so that your business users can actually see that context about the customers, like when's the last time they logged in? Have they finished this sort of onboarding step? How much money are they paying you? Then we allow you to easily do that with any data in your warehouse without having to do any in-house engineering.

And the second offering by High Touch is basically High Touch Activate, which allows marketers to sit on top of the data warehouse and segment the customers into a smaller dynamic list that they want to target for advertisements or emails and pipe these lists out to other platforms like Facebook Ads, or Google Ads, or Salesforce Marketing Cloud to actually run the campaigns. So that allows the other platforms to not really be like data platforms and just be the platforms that send out the campaigns while all the data management is done by High Touch on top of the warehouse.

[00:04:49] JM: What are the engineering problems in building a platform like that?

[00:04:52] S3: Yeah, that's a great question. So the way that High Touch is architected is that it connects directly to a data warehouse. So one of the challenges here is leveraging an existing data system rather than being our own data system. And so what we do is pull data out of a warehouse temporarily and then we use that to sync to destinations. What we don't want to do is actually be our own data store. And so the biggest challenge is kind of like being like a light layer around the warehouse that lets you activate it while not becoming like a database ourselves.

[00:05:20] S1: Yeah. But we actually see that as a huge positive, because it lets us avoid all the biases that other tools have. So most tools want you to pipe data into their own systems,

transform them into models that they understand. And then you end up be constrained by what those tools can do. And our tool lets you kind of use whatever models you have and build it in a way that's extremely flexible and avoiding all those biases.

[00:05:44] S2: Yeah. So instead of asking companies to morph their data into our format, like most tools do, we basically allow you to have a marketing platform that wraps around your data format and can understand your data and allow you to do very particular purposes, very particular tasks on top of data like segmenting customers or running marketing campaigns no matter what the underlying data format is in the warehouse.

[00:06:06] JM: What are the ways in which you see the current infrastructure around customer data as being broken?

[00:06:13] S3: Yeah. So one thing, I think the most important thing is it's kind of about like ownership of data. So when you pipe data into external tools, there's always some kind of like ETL or engineering process involved in getting data into those tools, which is already a pain point. Then the problem is when you want to change data or model it or have like custom views, it becomes quite challenging when using external systems. And so the data model for different companies looks quite different. The way that most marketing tools and CPs work is you conform to their data model rather than them conforming to your data model. And so what you have to do is kind of like collapse down your complex hierarchy of data from your warehouse into a simple user's view, and that means typically they only support like user level properties rather than having like related objects, like the full hierarchy of data. And so the complexity is that in order to get data into these tools, you kind of have to know what data you want ahead of time and then adding additional data is like kind of a pain point as well. These systems make it very challenging to change the data within their systems and like model and add new data.

[00:07:10] S2: So just as an example of that, you won't believe how many companies we've talked to that can't run a simple abandon cart campaign that factors in whether the item in that inventory is still in stock or not. So like a company, like a large retailer has trouble basically running these campaigns on people who have added something to their cart, but abandon it, while factoring in the fact that that item is now out of stock. And that's the reason they invented it.

Another common issue with b2b companies is a lot of them have setups where a user can belong to multiple workspaces or organizations and just tons of marketing tools in the market and customer data platforms in the market just don't support this sort of hierarchy or relationship. We even talk to companies that can't use their own tool in-house. They're a vendor that can't dog food their own tool due to this problem. So essentially what we've found is that, with High Touch, we've made it in a way that it can sit on top of your warehouse data and understand any hierarchy of data just like a database can and build a marketing UI on top of that rather than trying to do it the other way around where the customer has the burden of fitting data into your tool.

[00:08:11] JM: Who do you see as the prototypical operator of this tool? Is it a technical user or is it a less technical user?

[00:08:18] S3: Yeah. There're basically two different use cases for our application. The first is a more technical user, something like a data engineer a data analyst who's tasked with getting data into various business systems. So these could be marketing tools. It could be CRMs. It could be helpdesk systems, things like that. And so the way that it would help a data analyst is it would enable them to put data into these tools without having to write custom pipelines. And so they'd be able to go in, write SQL and push data into these tools and save a bunch of engineering time. The other user with our more visual interface is either like marketing or marketing apps, and these are less technical user who would like a point and click interface. And so what this allows them to do is go into the High Touch UI. It allows them to explore data and then pull data into their marketing tool of choice without having to pull in data engineers. So that saves them from having to ask for favors from engineering.

[00:09:03] S2: And the second use case is super useful for, for example, b2c companies that are running tons of ad hoc campaigns and seasonal campaigns throughout the year and have large marketing teams with large marketing spend. It allows the marketers to actually iterate on all that data faster.

[00:09:17] JM: So do you see High Touch as being useful for setting up these kinds of ad hoc campaigns? Or do you see anything that's useful where you have incoming streams of data and you're using High Touch to react to events as they're coming in?

[00:09:32] S3: Yeah. The reaction kind of happens for us through a data warehouse. And so there is like a middle layer, which is the warehouse that events come in through various sources. And then once you're in the data warehouse, High Touch works off of that. We don't currently operate off of events teams directly, but that's something we're kind of exploring of like other use cases. But right now we kind of act just on top of the data warehouse.

[00:09:50] JM: Does it matter what kind of data warehousing tool we're talking about whether it's Snowflake, or BigQuery, or Spark?

[00:09:56] S2: No. So we support like a large variety of data warehousing tools. SQL is kind of what ties them all together and it's like the universal interface despite there being some quirks for different providers. So we support probably like over 10 data warehousing platforms and data lake platforms that allow you to run SQL, whether it's a BigQuery, or Postgres, or MySQL, or Microsoft Sql, or Spark, or Snowflake. That's kind of the commonality that's there between all companies. They're all trying to get all the data into the warehouse.

[00:10:22] S3: Yeah. That being said, there is a bit of an advantage to using like a more modern data warehouse of using platforms like Snowflake and BigQuery where they have a separation of compute storage. These types of systems are able to spin up compute in like a serverless fashion, which allows the queries to run much faster. And so it allows you to like bring down the latency of how you can activate these different campaigns. so it does work a little better with like the next-gen data warehouses, but we work with data lakes and like existing data warehouses just as well.

[00:10:46] S2: And essentially the fast and rapidly growing adoption of these fast data warehouses is where we see the opportunity is to kind of shape how software that involves a lot of data is actually running in SaaS. So essentially, tons and tons of companies are adopting Snowflake and BigQuery, and that's where we see a big opportunity and allowing them to activate the data on top of that. Now that the technology that they have for their core data

warehouse is actually fast enough to run things like marketing applications, whereas previously this sort of technology was not adopted mainstream. It was only used by vendors individually, like marketing platforms themselves. But now that everyone has it, we think there's an opportunity to have SaaS that live directly on top of your data warehouse.

[00:11:27] JM: Why is it important that there's been this revolution in new data warehousing technologies like Snowflake? Like why is that relevant to your business? Why wouldn't this work you know five years ago?

[00:11:38] S3: Yeah. I think there're two different things. The first one really is just the speed. It turns the warehouse into something that is actually suitable for activation. Previously warehouses like we're really only useful for like analytics purposes. You throw a bunch of data in there and then data analytics would make sense of this data and then run reports, and it's used mostly for analytics purposes. Now that the latency around running queries has gone down dramatically, it's actually possible to build user interfaces directly off of a warehouse and then able to run things like campaigns in near real-time, which is previously not possible.

The second thing is that it allows more data modeling to be done within the data warehouse rather than having to be transformed in an ETL process and back into the warehouse. So this allows – It makes it much easier to do things like transformations inside the warehouse and clean the data. So previously the data in data lakes was like much messier because it's difficult to transform the data. Now that it's fast because of data warehouses like Snowflake, it's easier to model the data and make it cleaner. And so like the quality of data that's inside data warehouses improved. So it's easier to build UIs off it in that sense too.

[00:12:37] S3: Also, the barrier to entry to having good data infrastructure has gone down dramatically, right? Because before you'd have to spin up your own Hadoop cluster and like do a lot of engineering in order to make these types of data warehouses work, now you just spin up a Snowflake or BigQuery on the cloud and it works for you. So smaller companies or companies that have lower budgets can also afford data warehouses.

[00:13:00] S2: And I thought I'll add that the core technology change that really has allowed all these improvements is the separation of compute and storage as well as the cloud. So

separation of compute and storage, basically the way all databases and data warehouses were architected was that compute and storage were tied. As in all your processing was running on an instance and also stored the data. And if you wanted to scale that instance up and you wanted more storage, it was tied to the actual compute. Now the way these data warehouses are architected is that the storage is actually in systems like S3, or Google Cloud storage, and it's these distributed systems that are accessible by a fleet of compute resources on the cloud. And these compute resources are actually shared by cloud providers like Snowflake and Google to power things like BigQuery and Snowflake.

And as a result, if I want to run a query, it can access all the compute resources across of Snowflake or Google's BigQuery clusters even if I personally only have a small amount of data. And that allows basically bursty capacity, since you're sharing resources amongst everyone in the cloud and the compute is not actually tied with the storage of the data. So essentially this core change is what allows companies to, A, dump as much data as they want into the warehouse and then transform it inside of there, because they don't have to worry about the increasing cost of scaling up your individual instances. And B, run really fast queries without provisioning resources ahead of time, since everyone's leveraging the wider resources of everyone using BigQuery or everyone using Snowflake.

[00:14:29] JM: Sorry. Could you go even deeper on why that separation of storage and compute is so important?

[00:14:36] S3: Yeah. It really comes down to scalability. And so when compute and storage are tied, it's difficult to scale up instances like in your own Hadoop cluster. So imagine like if you're somebody who's running your own Hadoop cluster and your usage is going up, but the amount of resources that you're – Like the underlying resources are staying the same. You'll actually have to like ask somebody to provision more resources. And so you can't just scale up and down very quickly, whereas in like a serverless model like what Snowflake provides, you could run like a very complex query, and then behind-the-scenes they just scale it up for you. And that's what the separation of compute and storage really allows, is like that serverless model to exist in the cloud.

[00:15:13] JM: There's been a movement from ETL towards ELT. How does the movement towards ELT change your perspective on data warehousing?

[00:15:24] S3: Yeah, that's a great question. So what ELT really allows is a new user to transform data. So previously ELT was something that typically was handled by data engineering, people who typically wrote pipelines and were like day-to-day software engineers. What ELT allows is people who know the business data very well. They understand SQL very well. It allows people with those roles to go and transform the data, which is good, because it means that there's actually like less code being shipped. All that's really being done is like there's new models being built at SQL. So the engineering overhead is lower and people with like the business domain knowledge are the ones that are actually transforming the data. So that's like a very good movement for the data warehouse space.

[00:16:03] S2: And part of this is only possible because it's so cheap to dump as much data as you want into the warehouse now due to the separation of compute and storage that we mentioned earlier. Now that you can dump all your information to warehouse and don't have to worry about that, you can actually transform it once in the warehouse, whereas previously if you were managing your own warehouse's resource and your own database resources, then you have to worry about how much information can I put in this, and whatnot? Now with the cloud providers and the serverless model, you don't have to worry about that.

[00:16:29] JM: This is a bit out of the purview of what you guys are focused on, but Snowflake is generally used as an OLAP data store and analytical processing data store. But is there an opportunity for Snowflake to be an OLTP database, as well a transactional database?

[00:16:46] S3: Yeah. That's actually a great question, and it is actually pretty relevant to what we're doing. So I think kind of the ambition of snowflake is to offer like different kinds of write patterns and be faster at things like transactional processing. So as more vendors start to connect directly to warehouses, start pulling data out of them, there's going to be like increasing demands to also write data back into the warehouses and then offer things like a database. So I think there's going to be this movement where more and more staff providers are going to build directly on top of the data warehouse and having access to a database that's joined with like the rest of your Snowflake data is very handy. And so for example, like in our application, it would

be pretty nice that we were able to like store some state or other things inside like a Snowflake instance and store like less on our end. And so I think there's definitely like increase in demand for other things like that. And you could even imagine things like CRMs being built directly on top of Snowflake in the future if these underlying primitives are possible.

[00:17:37] JM: Do you have any understanding about what the difference in architecture is between Snowflake and the other popular data warehouses like Red Shift and BigQuery?

[00:17:47] S2: Yeah. So Red Shift actually doesn't have the model where compute is separated from storage. So fundamentally, it's quite different than BigQuery and Snowflake, which are actually pretty similar in terms of the overall architecture. So Red Shift actually has the model where you do have the ability to scale out to multiple nodes, and the difference between Red Shift and Postgres is that. They can distribute storage across multiple nodes and partition it. And B, that it's a columnar database. So a columnar database means if you're only selecting certain columns, you're going to spend a lot more time, or a lot less time running that query because it's actually oriented on columns instead of rows. But if you want to find a particular row very fast, then it's going to take more time. So it's really great for doing aggregations and big data processing on certain columns, but it's bad for serving an application, like logging into a web app, because it's very hard to find a specific row, like a row which contains this user and all the columns for it.

So Red Shift, BigQuery and Snowflake all have this columnar database architecture, which is what ties them together. But the difference between BigQuery and Snowflake and Redshift is that Red Shift actually has compute type of storage, and that's one of the large maintenance hassles of running it. It doesn't benefit from this serverless model. Or it doesn't benefit as much from this cloud scale model. If you do run out of resources or you want to run a query faster, you have to actually go and scale your instance up, which is a very painful process. At Segment internally, we used to use Red Shift and eventually moved to Snowflake. And the difference was day and night. And we've talked to so many vendors where this is also the case. So basically, BigQuery and Snowflake have this compute and storage separation, which is what changes everything.

Actually, I believe that's part of the reason that Snowflake's really taken off. I mean, Google Cloud does not have as much proliferation as a cloud provider than compared to AWS. And their sales processes are not up to par and things like that in terms of the enterprise. And AWS just really hasn't delivered anything that's at par with Snowflake in terms of separating compute and storage. They do have some offerings like AWS Athena, but just the UX and the robustness of these offerings is just not there compared to like Snowflake or BigQuery. They don't even really operate like a database. They can just run SQL on top of your S3 data. They don't have inserts or any of these sort of like database semantics that Snowflake and BigQuery have really nailed to allow people to adopt it in their applications easily.

[00:20:10] JM: Can we go deeper on the idea of the data warehouse as a platform for building other applications?

[00:20:18] S3: Yeah. It's a pretty broad topic, and I think there's going to be different stages that this happens in. For example, like right now, in the current state of data warehouses, it works for some applications we built off of it. I think the first application we'll see where applications are building directly off of it is for a lot of marketing purposes. These are the teams that usually need direct access to data. They're the ones that are the most data-driven. And the data is going to be used in a way where it's pulled from these systems and then used in other systems, whereas with tools like CRMs, it's more of a manual workflow where somebody would go in and like update records, which is like more of like a write pattern. And so it's going to come out in stages, I think. And the first stage we're going to see is warehouses starting to be used more as like customer data platforms. Things that enable marketer use cases, data-driven marketer use cases.

I think the next stage is it's going to kind of depend on the priorities of the data warehouse systems. So the more features that they add to enable these different use cases, the more people can build off of them. Like right now, if you had the goal of building a CRM on top of Snowflake, it's technically not very feasible to do that. Whereas, if snowflake were to shift features that allow faster transactional processing, then it kind of unlocks those different use cases. I think the underlying drive to do it in this way like rather than having an external system is just to have all of your data in one place.

So today, like for a lot of companies, Salesforce is almost like a shadow data warehouse where you have to push things like custom objects and custom fields into it in order to get the full customer review. Whereas if CRMs in the future were built directly off of Snowflake, what do you be able to do is join CRM data, things like leads and context with other data that that you have across your company. And that just really simplifies like the whole process of giving you like a single customer view.

[00:21:59] S1: I think one thing you'll find interesting is that Snowflake has talked about building a kind of an app marketplace where people can ship apps that run in Snowflake Cloud. And then the Snowflake itself is the data store for those apps. So in theory, like companies could ship UIs or like kind of business level apps that run on snowflake as both the compute and the data store, and that could really change how people think of their CRM as their data store or their service desk as a data store.

[00:22:30] S2: Yeah. And we've seen a lot of use cases come up outside of the marketing space as well. Why we think marketing is especially an interesting space for this warehouse movement is, A, because most of the new marketing platforms are actually using these data warehouse technologies behind-the-scenes. So it kind of doesn't make sense for them to have a Snowflake while you also have a Snowflake at your company. And then B, marketers just need access to a really large breadth of data and frequently, because they're always running campaigns and always diving into the customer base. But we've actually seen the applications of this outside of marketing too. We've come across financial institutions that are looking at Snowflake as a way for one financial institution to clean up their data and share it with another very easily. And snowflake is really investing in platform and trying to make this possible, because I believe – I don't have any information, but I believe that they see that as their differentiator between the other providers, kind of the user experience that they're able to build around platform.

And the reason I believe this is Google and AWS, they own the data centers. They own the resources at the end that Snowflake is operating on top of, and they're going to catch up on kind of the user experience of the core OLAP database and data warehousing technology that Snowflakes produce. But eventually what Snowflake can actually differentiate on is these higher

level primitives and the platform that they're building to make the user experience for all these apps really great.

So for example, they've built OAuth type APIs in this thing called Snowflake Data Marketplace where one vendor can just start writing data into your Snowflake or start reading data into your Snowflake really seamlessly as if you're just logging in with Facebook or logging in with Salesforce. I think the UX that they create around this and the innovation that they do here this is just the beginning is what will really be able to allow them to differentiate in the market compared to Google and Amazon, which don't traditionally think of these experiences past the infrastructure level.

[00:24:22] JM: Do you guys have any perspective on the rising importance of DBT? The data build tool?

[00:24:29] S3: Yeah. This kind of goes back to your question of like ETL versus ELT. As you start to do more and more transformations in your warehouse, the complexity starts to rise of like managing the schedule around these, orchestrating the schedules and just having like a framework around organizing these different views in a warehouse. So DBT is a great tool. It's open source. We've seen very, very vast adoption of ETL already, and I think that's only going to continue to grow. But yeah, I think it's something that any organizations like looking at doing more ELT should consider, because it just helps manage that process.

[00:25:00] S2: Yeah. And just to clarify for the audience, ELT is extract load transform; versus ETL, is extract transform load. So ETL was the way of doing things for a long time where people would write scripts that would load data from – That would pull data from a source. So that's the extract step. Transform data into the format that makes sense for the business model and then load it into the warehouse. And all this was done in like one thing written by a data engineer or an engineer. Now people are moving to the model which is ELT, which is get the raw data, put it in the warehouse immediately, sometimes even in like a JSON format that BigQuery and Snowflake actually know how to operate on. And then the analyst can actually transform the data into the right format for each business process within the warehouse using SQL. So this is kind of the way that companies are moving towards today, because instead of asking engineering to change the way a certain field is represented in the warehouse to fit some sort of

business need the analyst who's used to digging through data in SQL can do that directly, which brings more power to the analyst.

The problem here is actually running and orchestrating these transformations is challenging for an analyst to do directly in the warehouse, because they have to run a bunch of create views, replace these views. There could be dependencies between them. You might want to run them in a schedule. And what DBT has taken is an approach that says, "Okay. We'll be a platform that lets you manage all of these transformations and views and table creation on top of the warehouse as an analyst by just writing SQL and not doing any sort of data engineering." So that people don't have to write code to orchestrate this between companies and puts the power back in someone closer to the business user, which is the data analyst that can write SQL.

[00:26:46] JM: So just to revisit exactly what you are building, you see yourself as a layer that sits on top of the data warehouse and can query the data warehouse in order to build customer data analytics, customer data applications on top of those data that sit in your warehouse. Could you give an example of an application that a user might build?

[00:27:07] JM: B. So I'll give our two most common examples. The first most common example is that I have a CRM like Salesforce, or Zendesk, or HubSpot in place and I'm running tons of different customer operations out of this CRM. So this is my sales process, my account management process, my support process. But my company doesn't have good data in that CRM. That said, it does have good data about the customers. Like when's the last time they logged in? Have they completed this step in the onboarding? How much are they paying us in their warehouse? Where business analysts are able to create reports in SQL to service as data.

So what High Touch Connect actually allows you to do is take any data you have in the BI tool, and without writing any code get it to actually display in your CRM and be constantly updating in your CRM so that business teams can actually leverage that data, but without writing any engineering, without spending in-house efforts, without doing any maintenance on all this. So High Touch Connect allows you to get any data you're able to pull out of your warehouse in SQL and make it start showing up automatically in all of these end tools, most commonly CRMs.

In terms of the use case for High Touch Activate, what we see there is that a marketer might currently have a workflow where they ask an analyst every time they're thinking of running a targeted campaign against users who have purchased this item or that item and haven't purchased another item in the last six months, basically all these sort of segments on their customer base. Every time they want to think of a campaign on them, they ask an analyst how many users are in this cohort or how many users are in that cohort? And then they finally decide, which campaign to run and ask an analyst to pull them a CSV of those users. And then they upload into an ad tool like Facebook or they upload it into an email tool like Salesforce Marketing Cloud or Mailchimp to actually run the campaign.

High Touch basically replaces this back and forth and manual process where the marketer is bottlenecked by the analyst. The analyst is spending time just answering specific data requests with the UI that sits on top of the same data source, which is the warehouse that the analyst is writing SQL against and allows the marketer to just use a super friendly UI to do this whole process end-to-end, which is exploring the customer base and syncing that data to destinations where they can actually run the campaign.

Kind of the unique advantage of High Touch data platform as a whole is that it doesn't require any engineering effort to implement. We basically see there being a rise in the data team instead of the developer team. So a lot of companies out there like Segment are really catering to the developer team. At High Touch, we're catering to the data team, which is a team that can write SQL and is an ever-growing customer base. It's a team that's using things like DBT. And their layer that they operate at is SQL. So that's the interface that High Touch uses for all its operations. It sits on top of the warehouse rather than actually requiring them to instrument any code. So to use High Touch, you don't have to put anything in your app. You just connect it to your existing data.

[00:29:52] JM: And so is the point that High Touch is a thin layer over your database or over your data warehouse and you can use it to plug into the integrations with all the other platforms that you need to plug into to build CDP applications?

[00:30:06] S2: Exactly. Exactly. So High Touch is the connection between the data warehouse and all your business applications, whether it's marketing your sales or anything like that.

[00:30:16] JM: And so as an example, like a bunch of Stripe transactions might trigger some CDP application that I've built in High Touch.

[00:30:24] S2: Exactly. Exactly. So you might use High Touch to push your Stripe transactions to show directly on the account in Salesforce, or you might use High Touch to query users inside of your warehouse based on their Stripe transactions and run a campaign to them in Marketo or Iterable. Basically, anything you need to do with data in your warehouse in an end business tool, High Touch is the connection layer for that. It allows you to create a central customer record and operate on that over the warehouse.

[00:30:51] JM: And again, the difference between that and Segment is that Segment actually holds your data, whereas in High Touch you're accessing data in your data warehouse.

[00:31:00] S2: Exactly. In High Touch, the layer we operate at is the warehouse. So we can operate on any of the data in the warehouse. You don't have to put it in any certain format like a Segment Spec. If you use Segment, that's great. We can operate on that data that they write into the warehouse as well, and we don't hold any of your data as well. So basically we're addressing the market of people who are using new data warehouses as a whole rather than people trying to revamp their ETL or improve their event collection, which Segment helps out a lot with, and it's a great product around that.

[00:31:29] JM: So the comparison to go deeper in Segment, there's domain specific code you have to write. And in High Touch, you're just writing SQL.

[00:31:38] S2: Exactly. Exactly. So if you're not capturing certain events from your website, like someone clicked this button or anything like that, Segment helps out with that wonderfully. And if you already have the data you need in a warehouse, which you find is commonly the case, then High Touch sits at the level and lets you unlock the data without writing any code.

[00:31:56] JM: Are the advantages mostly in like just flexibility? The fact that it's not a domain specific language? Is it cost advantages? what exactly is the advantage of if I'm comparing High Touch to a customer data platform like Segment?

[00:32:13] S3: Right. So there's a couple different points. The first one is that it does have all of the data. And so relational data is something that doesn't typically fit very well with like any CDP. CDPs typically understand users, potentially accounts, and that's about all, and they're able to activate data that's structured into that format. What High Touch allows you to do is have a true user model where you have users and you can have Stripe data join with that and you could have product data. Every business has their own unique data and their own unique hierarchy of data. And what High Touch allows you to do is make use of that whole hierarchy. And so the problem with CEP is that a lot of times in order to do things like an abandoned cart campaign or running a campaign against users who have added an item to their cart that's running low on stock, that's something that would end up getting pushed back to data analysts. So data analysts have to write a SQL statement to query these lists of users. And then it would get pushed into a CDP as like a boolean field of users who have one of these such items in their cart. That's not really the ideal workflow. The ideal workflow is that marketing can own this whole campaign end-to-end. And so having a visual builder on top of the warehouse allows you to do all sorts of things, building on top of the relational database, relational model that exists in your warehouse.

And the other thing too is just it's nice to have control over your own data. So you're able to flexibly change this data model over time. You're able to iterate on it. You're able to pull in new data sources easily without having to write new ETIs into external systems. Because you own the data warehouse, because all the data is in one place, what you can do is ELT inside that data warehouse and then activate the data rather than having to pull in engineering resources to push new data into external systems.

[00:33:43] S2: Yeah. So just as an example, if you incorrectly write a bunch of data into your warehouse, you can just use DBT to transform it and fix it. If you incorrectly write a bunch of data into a CDP, like a Segment, or an mParticle, or an ActionIQ, then you actually have to use their tools to fix this data or contact support or something like this. Whereas inside of our tool, High touch, since we sit on top of the warehouse, you can use any tools that are available on top of warehouses or SQL itself to fix these things.

There are so many times where we've asked ourselves while using some SaaS platform, like Salesforce or Zendesk, like how do I fix this data in bulk? Or how do I rename everything that has this into something else in a field on a customer? And it's so hard to do that. There's like 100 ways to do this in Salesforce if you Google it. But if you have access to the underlying data, so you own the data in your warehouse, then you can easily do this in SQL.

[00:34:33] S3: And most CDPs do use data warehouses behind-the-scenes, but you don't get the same level of access to it as you would by using your own data warehouse. So it's the same data store under the hood in both cases. But what's different is when you use a CDP, you don't typically get direct access to that data warehouse. And so you're not able to do complex modeling or have raw SQL access to. Instead what you get is a higher level interface or changing data becomes really difficult.

[00:34:54] S2: And the reason for this is a lot of CDPs were founded before companies all had these fast data warehouses. So CDPs were some of the first adopters of things like Snowflake and BigQuery behind-the-scenes. And now that everyone has these fast data warehouses internally and needs to build them not only for marketing purposes, but just for business intelligence and analytics, our whole philosophy is that we should help you activate the data that you already have and you already have in the best format for your company where it is rather than asking you to bring it somewhere else and move it into our format.

So basically we feel that the shift in the data warehousing space is allowing certain applications to be rewritten with the data warehouse in mind. And Snowflake is growing like crazy. I mean, if you look at some of the stats online, I think I saw that 150 out of the fortune 500 have already signed a contract with Snowflake, which is just really remarkable in terms of their performance as well as what it means for applications that want to be built around the warehouse.

[00:35:54] JM: We did a show a while ago with an open source customer data platform called RudderStack, and RudderStack essentially separates the CDP from the data warehouse. How would you compare to a RudderStack?

[00:36:09] S2: Yeah. So RudderStack is actually directly comparable to Segment. How I would think about it is an open source version of Segment. So RudderStack handles – It's an all-in-one

solution. It handles you know event collection. Piping this to various different services, including your warehouse, and it's kind of built with developers in mind. So our tool is built with the data team and the business user in mind, and it's about activating the data you do have in your warehouse. But RudderStack is completely like one-to-one comparable with Segment. It's really just an open source version of Segment. And they're not really focused on kind of building higher level UIs to activate the data in your warehouse or anything like that. I would say they're more like customer data infrastructure rather than a customer data platform for business users and analysts and marketers to go into and actually activate the data.

[00:36:56] S3: High Touch would actually work quite well alongside RudderStack. So RudderStack could be used to collect data, transform it, clean it and then push into a warehouse, and then High Touch can sit on top of that data and activate it for various purposes.

[00:37:06] S2: Yeah. So we're pretty much vendor agnostic in terms of how you get data into the warehouse. In 2015, 2016 when I joined Segment, there weren't that many vendors writing data into the warehouse. Now Mixpanel, Amplitude, Google Analytics, Segment, RudderStack all know how to write data into the warehouse. Basically every vendor remotely related to marketing or analytics in the space is doing this. And therefore High Touch just sits on top of that and helps you activate the data no matter how you're collecting the data. You can understand anything from these models. And that's where we see the real opportunity. We're not focusing on data collection at all. We think there's a lot of players in this space that are all doing different unique takes on it and a good job at it.

[00:37:46] JM: Do you guys have any predictions for how the data warehousing space looks different in five years?

[00:37:51] S3: Yeah. I think some of the earlier platform stuff that we discussed, I think that's kind of the direction, and there's various different ways that that's going to manifest. One of the ways I think is we'll see faster transactional processing in systems like Snowflake. One of the things we haven't really touched on too much yet, but I think is another important direction that warehouses is going is streaming warehouses. So right now the use cases around warehouses are pretty batch. You would run a query and then you'd like to schedule out like every hour or such on. But the trend is towards data flowing into a warehouse. All the models that you have

built on top of the warehouse, those being computed in real-time and all the models that you have built on those also being computed in real-time. And so as data flows in, everything's going to update in real-time. And so many things in the future like things like BI tools will start to like update real-time. You'll see like live dashboards. The effect for like tools like High Touch is that the latency on campaigns or real-time use cases become even more real-time. So that's something like another very, very large trend in this space.

[00:38:44] S2: Yeah. So basically what we think is the interesting opportunity around data warehouses, and Snowflake and BigQuery in particular I think have the best shot at this, is that SQL is really just an interface. So SQL is a declarative language to do data processing. And today you're often using it with transactional databases, or analytics databases. There's this kind of divide. But we're also seeing SQL propagate into new territories like stream processing. So there's Spark SQL. There's an open source project called Materialize, which uses the Postgres protocol and supports Postgres SQL, but does incremental and real-time computations on top of the data that you're inserting instead of running a SQL query over all the data that you have each time.

We think the opportunity for the players in this space is to make a unified interface where all you're thinking about is SQL and you can selectively enable different types of access patterns on different tables. So you can say, "I want this table to be able to have fast reads of a single row. You should store it as an OLTP format. Or I want this table to be accessible in batch like Snowflake is primarily today. And then I actually want this SQL view to be able to be computed in a streaming way. We see the huge opportunity as to kind of create a super database around this SQL as the interface and allow the user to just express what kind of guarantees or semantics they want around reading and writing this data, and the warehouse to figure out how that actually operates. And it's not just a dream. I do think this is actually becoming possible. And I see each cloud provider working into this direction. For example, Snowflake, Red Shift and BigQuery have all released some form or another incrementally computed materialized views. So this is not like this is in beta. There're caveats and things like that. But basically it means when you re-run a SQL query to produce a new table, which is called a materialized view, for doing this sort of ELT in your warehouse and transformations in your warehouse.

Usually, before, Snowflake, Red Shift and BigQuery and all the datawarehouses would basically rerun that SQL query on top of all your data and re-compute a whole new view and then save it as a table. Today the warehouses are moving into the state where they can incrementally compute that. So if you're averaging something, they can be computing a rolling average instead of computing the average from scratch over all your data when you're refreshing this view nightly. And this is the first step we see towards a truly real-time system. One that some of these open source projects like Materialize and PipelineDB are kind of working towards, and we think that Snowflake and BigQuery are both really making a big strides here. We're excited to see what comes in the warehouse space there.

[00:41:22] JM: Well, that sounds great. Do you guys have any additional points you'd like to add about data warehousing or the data space in general?

[00:41:30] S1: Well, I think one thing to mention is that the same way we saw access to like transformations and like how data is used move from the engineering team to the data team. We truly believe that eventually the business teams will have the same access to data. So right now the business team has to go through the analyst team in order to get queries written for them or to give data pipe to their tools for them. The purpose of High Touch and hopefully other platforms that come after High Touch would be to give the business user that same power so that the business user isn't bottlenecked by waiting for other teams to write queries. And that business user actually is presented data in a way that they can understand, because like business users do use Excel, right? They do have the capability to understand data. But it's just that data in a warehouse is often pretty complicated. So we're building UIs to kind of simplify that where the analyst team can present data to the business team and enable certain tables, enable certain columns. And then the business team can do their own analysis and choose which data they want to send to which tools and how they want to use data.

[00:42:30] S2: Yeah. And we think there're others who are working on this problem as well. Looker, which was acquired by Google is one example. But the way they're trying to approach this problem is very general. They're trying to allow business users to slice and dice data by different dimensions in a lot of complicated ways. We think the right approach to this is vertical BI. So we think the right approach is enabling very particular use cases like marketing segmentation or syncing things to a CRM on top of the warehouse. And we think that's where

there's a lot of opportunity and are excited to see more players pop up in this space, because there's no real reason that you shouldn't be able to produce the same user experience on top of a data warehouse that you can produce inside of an end tools data segmentation layer like Mailchimp.

The other thing I would mention that's unrelated to this user experience point is there's just an increase in consumer privacy, focus amongst companies, data regulation. And owning your own data in the warehouse produces massive advantages here. We think if Snowflake does platform write and if enough startups work on this, then it just simply won't make sense to replicate all your data into tons and tons of different SaaS tools. Instead, it'll make sense for those SaaS tools to live directly on top of your data warehouse so that you can own your data and know exactly where it's flowing. A lot of customer data platforms kind of manage the flow of data and deletion of data from different providers. We just believe this whole problem can go away if data access is happening directly into your own infrastructure. And this is in line with what we're seeing in other industries where people are running more software in their own cloud, which is kind of pseudo-on-prem.

[00:44:04] S3: It'd be so cool if all the SaaS's were like, "All right, you can BYO database. Just bring your own data store." And then like our system will work on top of that so that you get to have control and you can change your records however you want, and then they'll just automatically pop up in our records the exact same way. That's definitely possible if Snowflake does this right.

[00:44:26] JM: Well, that sounds like a good place to close off. Guys, I want to thank you for coming on the show and talking.

[00:44:31] S2: Awesome. Thank you so much, Jeff.

[00:44:31] S3: This is great. Thank you.

[00:44:32] S1: Thanks, Jeff.