

Rudderstack







by Head of Product

Customer Data Platform

Details

Review Date	08/31/2023
Purchase Date	Q2'22
Implementation Time	2 months
Product Still in Use	Yes
Purchase Amount	per event processed
Intent to Renew	90%
Review Source	Elion

Product Rating

Product Overall		4.5
Use Case Fit		5.0
Ease of Use		4.0
API		5.0
Integrations		5.0
Support		4.5
Value		4.5

About the Reviewer

Purchasing Team User
Implementation Team Product Oversight

Reviewer Organization

Virtual-First Provider Chronic Care

Reviewer Tech Stack

Braze Sentry Facebook Ads
Google Ads Google Tag Manager
Mixpanel

Other Products Considered

Amplitude Freshpaint mParticle
Segment Snowplow

Summary

- **Product Usage:** RudderStack is utilized for handling sensitive customer data across several platforms including Mixpanel and Braze, and for facilitating integrations with advertising platforms including Facebook and Google.
- **Strengths:** The platform offers hands-on technical support, has robust integrations with other tools and offers full control over data security; beneficial for healthcare where stringent data protocols are required.
- **Weaknesses:** RudderStack demands a higher degree of technical capability from users and has a steeper learning curve compared to competitors; its documentation may not be always up-to-date given frequent updates.
- **Overall Judgment:** RudderStack is a reliable and secure data management tool that provides users with a strong degree of control over their data, its strong technical support and integration capabilities make it a valuable investment for organizations dealing with sensitive customer data.

Review

So today, we're chatting about RudderStack and how it's used at your company. Before we jump into that, could you give a brief overview of the company and your role there?

We started as a direct-to-consumer clinic that focused on assisting individuals with chronic, hard-to-treat conditions that didn't have a single treatment option. We took a comprehensive approach to care, providing patients with not only a physician but also allied health professionals such as registered dietitians and health coaches. Our aim was to tackle the condition from multiple angles rather than relying on episodic visits to the doctor every six weeks. Our care model was ongoing and subscription-based, with most of our platform communication happening through text, though we also offered occasional video visits. After about a year, we made a pivot in our business strategy. We continued our direct-to-consumer offering, but we also started selling our platform to traditional in-person clinics. This allowed them to incorporate a virtual care component and change their treatment approach for chronic patients who weren't seeing improvements from a few office visits each year.

My role was leading product at the company.

How long have you been using RudderStack?

A little over a year.

What drove you to look for a product like RudderStack?

When we initially started selling directly to consumers, we were mainly focused on analyzing the performance of our sales funnel and user behavior. We knew that we needed advanced analytics, and I was anticipating the need to integrate marketing tools for attribution, engagement, management, and other functions. I had been on teams that had attempted to build these integrations in-house, but it required a significant amount of maintenance. Therefore, we decided that investing in a customer data platform (CDP) solution was the best option, especially considering our small engineering team.

The two CDPs I was most familiar with were Segment and mParticle. So we chose to start with those. However, we soon discovered that most vendors are unwilling to sign a Business Associate Agreement (BAA) with us. Even some analytics vendors, like Amplitude at the time, refused to sign a BAA, meaning we couldn't use their services in a healthcare context without first anonymizing all the data. I had concerns about anonymizing the data, particularly for attribution-based use cases. We anticipated that users would interact with us multiple times before making a purchase, and it would be challenging to match all the anonymized data back to specific users. And given the large portion of our budget that was going to be spent on marketing, we needed to be able to analyze how it was working.

Toward the end of our comparison, we narrowed it down to Freshpaint and RudderStack. We also had a fallback option, which was either using Mixpanel's not-so-great built-in tool or getting our data into a data warehouse and using a reverse ETL to extract and share it across other sources. For the reverse ETL approach, we considered Snowplow and another option, but it would require a more sophisticated setup. We'd have to set up the data warehouse and organize everything, which would involve more initial work. However, in the long term, we thought this approach would be better because we needed to combine information with our EHR (Electronic Health Record), which wouldn't be part of the marketing stack. So having a central warehouse to organize and connect the data, and then send it out to relevant places, seemed like a good idea. But it would require a significant upfront investment of time and effort, which we

weren't ready for yet. That's why RudderStack seemed appealing, as it checked many of the boxes we were looking for.

What were the key requirements you were using to evaluate these vendors?

The biggest factor for us was their willingness to sign a BAA and provide proof of other healthcare customers. Since they would be handling our sensitive data, we wanted assurance that they would keep it secure and had a good track record in doing so. Second, we considered the price and whether it would fit within our budget. Third, we considered their compatibility with other tools. We were making several vendor decisions simultaneously related to our analytics and marketing stack. Each of these potential vendors had different levels of integration with other tools, so we checked which ones they worked with out of the box. Finally, we also looked into how well they worked with popular ad networks like Meta and Google. We wanted to see if they could replicate some of the functionality of tools like Google Tag Manager, or if we would have to do that separately to achieve the ad reporting and performance we needed.

How did the different options stack up against one another?

So, philosophically, Freshpaint and other analytics tools are quite different. Freshpaint uses an implicit tagging model, where their JavaScript widget captures all possible events on a page and allows you to retroactively define what you want those events to be. It's similar to how Heap does analytics. On the other hand, tools like Amplitude or Mixpanel use explicit events, where you have to define events upfront. Google Analytics is interesting because it's a hybrid between the two.

I was initially skeptical of the implicit data approach because our analytics and product organization wasn't very large at the time. It takes a lot of discipline to keep the data clean in this approach. In my career, I've been in situations where marketing looks at one number and I look at a different number, even though we are talking about the same thing. It happens because marketing is looking at an event called "user clicks_button" while I'm looking at "purchase complete" or something similar. With a tool like Freshpaint, anyone can retroactively create their own events, so it becomes crucial to have strong organizational discipline to ensure the numbers are reliable and trustworthy.

My organizational context was also one where the head of engineering didn't believe in using any marketing tools. He had been in healthcare for a long time, and they had always built their own tools. This approach took up a lot of engineering resources that could have been better utilized elsewhere. My biggest concern was that if we didn't accurately report the numbers, such as conversions or funnel drop-off, people, especially the other members of the executive team, would start using unreliable criteria to make decisions. I wanted to be able to trust the platform we used, and I was doubtful that Freshpaint's implicit data capture approach would provide that level of trust.

I was really impressed with the engineering team at RudderStack compared to Freshpaint. The RudderStack team seemed more capable of working with us, which was a major factor for me. We had our own software engineers on-site, so I wasn't too concerned about integrating with RudderStack. However, throughout the entire process, RudderStack went above and beyond. They created a Slack channel for us and helped troubleshoot in the sandbox. In contrast, Freshpaint took a more sales-oriented approach. We were mainly communicating with their account representatives and not as much with their technical team. This raised some doubts about the ongoing support we would receive from them. With RudderStack, on the other hand, we received incredible support throughout.

Another reason we chose RudderStack was because their solution seemed scalable. When we started, our needs were not fully developed, but we anticipated growth. RudderStack demonstrated impressive progressions, both on their website and in their development process. This was reassuring because it showed that, as our needs became more sophisticated or our capabilities grew, RudderStack would be able to grow with us.

That being said, Freshpaint may be a better solution for some organizations. If you don't have a strong engineering team, if you have no prior experience with this kind of work, if you lack a structured approach to defining events, or if it's not a significant aspect of your company culture or something you're actively trying to incorporate into your culture, then Freshpaint would be a better solution.

Do you know how the pricing compared between vendors?

RudderStack had much more favorable pricing compared to Freshpaint. With RudderStack, the pricing was based on a per-event-processed model, which was quite low. In contrast, Freshpaint's pricing was determined by the number of users. This meant that every anonymous user who visited your website would be counted toward the total number of users, potentially leading to higher costs. If a marketing campaign performed exceptionally well, we could end up with a large number of users who were not necessarily engaged or active on the website. It was unnerving to think that costs were based on the total number of users, rather than active or repeat users.

What ultimately led you to go with RudderStack?

There were two main factors that influenced our decision. First, the data was stored in our AWS instance, which meant that we had full control over its security and encryption. This was important to us because we wanted to avoid any issues with sharing data with third parties or experiencing data leaks. RudderStack's approach eliminated their liability and placed it on us instead. If there were any mistakes in how we set up our S3 buckets or if a leak occurred, it would affect everything, not just RudderStack. This gave us a greater sense of control over the data's security.

The second factor was whether RudderStack would be able to accommodate our future needs, as we had planned to start transmitting data from the EMR. We also planned to use a third-party pharmacy, Truepill. We wanted to send that data back to the warehouse, join it, and then use our CRM, Braze, to send customers a text message when their prescription arrived. We believed that RudderStack's setup would allow us to enable these use cases effectively, while I was skeptical whether Freshpaint could support them in the way we desired.

Freshpaint initially seemed to have a lot of integrations listed on their website. However, upon closer inspection, we realized that their connections were not as comprehensive as they claimed. For instance, they claimed to integrate with our error reporting tool, Sentry, but in reality, the integration only provided a basic alert without the ability to extract detailed information from Sentry. We wanted to perform troubleshooting tasks like identifying the user associated with an error message, but Freshpaint lacked the necessary depth of integration. When we examined RudderStack, we found that their connections were much more robust and reliable.

How did the way RudderStack handled analytics differ from Freshpaint?

One major aspect of Freshpaint is that it stores data on everything that happens and then sends the data back to you. A big difference, which I think is a philosophical one, is that in RudderStack you have to define every event you want to capture. Someone has to say they want that event to have a specific name and occur in a certain client and environment. In Freshpaint, you can do that, but the default way is to simply drop their code in. It captures everything it can see, and later you can label the stuff that's important to you.

It's a trade-off. I think the Freshpaint approach is really appealing for new startups who are just starting out and don't have a clear understanding of what's important yet. However, it's difficult to trust that you're actually getting what you think you need or want when you're labeling things retroactively, or that the events are firing as expected, especially when the volume is low. When you're new, you're not going to have a million people clicking on your buttons or whatever it is you're measuring. So with only tens or hundreds of data points, there's a lot of uncertainty. You can easily get a false sense of precision by relying on such limited information.

Another crucial factor for me at that time was recognizing the trade-off and realizing that our organization needed to prioritize the development of competence and discipline in defining events early on. I firmly believed that this should be a fundamental part of our product development cycle and routine – every release should be thoroughly instrumented. This approach aligns with my product philosophy that we must measure the results of everything we build and construct it in a way that allows for easy measurement and hypothesis testing.

To accomplish this effectively, it's essential to know what events we need to track from the beginning. By understanding the events upfront, we can align our development process with the way I believed we should build, especially when it came to validating key elements in the market. By doing the work upfront, we could confidently state that people liked a specific feature because they exhibited specific behavior. We could then instrument our product accordingly to track and analyze the desired behavior.

As an example, we want to see if a certain value proposition resonates with people, based on how they interact with our landing pages. If we want to evaluate this using scroll depth as our measurement, then we need to track scroll depth. Without that discipline, we might just throw up two landing pages and look at the results later in Freshpaint. But we might not have a good way to evaluate it, unless we've put in the work upfront.

RudderStack follows a hypothesis-driven philosophy, which I really like. In other organizations using tools like Freshpaint, they often rely on second- or third-proxy metrics. They try to get at the data in a certain way, like when someone clicks a button or when they scroll to the bottom of the page. But by doing that, they miss out on a lot of valuable data. And if you're comparing two things, you're already looking for a small difference in means. So you won't get a great signal-to-noise ratio, and that's when trust in analytics can break down.

My understanding is that Freshpaint helps you extract the data, and then they pipe it into something like Braze or Salesforce. Was RudderStack doing that as well?

They will do that too. Imagine someone visits your website and signs up for an account. Of course, I want to track that in Mixpanel. How many people visited today? Who signed up for an account? What other actions did they take? But we also need to inform Braze about it, so we can send them a welcome email and start them on the drip campaign. RudderStack handles this. You only need to define the event once, and then it will send the relevant data to all the necessary sources.

However, there are certain things that RudderStack cannot handle in the same way. For example, let's say that in our EMR, one of our doctors diagnoses a person with a disease, and as a result, we want to trigger the welcome series email specifically for that disease. The EMR data wouldn't flow through RudderStack like other types of information would, such as events from our website. So we send that data from the EMR to our data warehouse through RudderStack. Then, in the data warehouse, we would transform the data to match the patient ID with the customer ID and join them in the same table. After that, we would extract the data from the data warehouse and send it back to Braze, which would trigger an event in Braze.

Typically, data flows in one direction, either from user action or the server side to all of your sources. RudderStack enables this flow, but it also offers a mechanism to send the data back to other sources when a relevant event occurs downstream. This ensures that all the sources that need to be aware of this event are updated.

How did you find the sales process overall?

They were pretty engineering-forward, which was great for us. We wanted to test if we could integrate this SDK, how difficult it would be, and how well it would work with other components. I remember we actually did a proof of concept to see how the data flowed and how we could troubleshoot. And it turned out to be quite easy, which was one of the reasons we were convinced to go ahead with it.

How was the onboarding and setup process?

They were hands-on and very active in Slack. There was one strange bug that seemed to involve a combination of RudderStack and Mixpanel, regarding the way it handled joining users. They were incredibly helpful in troubleshooting that issue. It turned out that many others had also experienced this bug, which was just an odd quirk of the way Mixpanel worked. They remained hands-on throughout the entire process and assisted us with correctly configuring the S3 bucket in AWS. They were really involved in all aspects.

We didn't really explore the reverse ETL functionality, which RudderStack also supports. Their default model is focused on the forward side of data processing, where they capture client events, send them to the database, and transfer them out. However, they also have a feature that extracts data from the database and sends it out, similar to what Snowplow does. This is something we envision using in the future. While we haven't reached that use case yet, I have confidence that they will be very helpful when we do.

What specific use cases do you use RudderStack for?

RudderStack connects our client and server side actions to various analytics and CRM platforms, like Braze. It also facilitates integration with the Facebook Conversion API, which was important for us because we wanted to ensure patient privacy by minimizing the data sent to Facebook. To prevent any accidental capture of sensitive information, we did not install the Facebook Client SDK on our website, as it had the potential to capture any data. Although Facebook claimed to handle the data in a HIPAA-compliant manner, we didn't want to take any risks. Instead, we selectively sent events such as account sign-ups and checkouts, connecting them to the originating campaign. RudderStack also connected us with Google Tag Manager and Google Ads.

There were certain functionalities we initially planned to use but didn't due to changes in our business strategy. For example, we had initially anticipated extensively using influencer and affiliate marketing during the early stages of the company. RudderStack provided integration with various tools for this purpose, allowing us to track the source of referrals and provide commissions to influencers if a user signed up through their embedded button on a blog post. Although this functionality existed in RudderStack, we never ended up activating it because our business focus shifted.

Similarly, we had initially anticipated using a multi-touch attribution provider or a mobile measurement provider. The one I had the most familiarity with was Branch, but there were also options like Tune and AppsFlyer. I expected us to go with one of them, as they integrated well with RudderStack. However, things took a different turn when the business started leaning more toward a B2B approach, so we didn't end up using any of those providers.

Do you see any weaknesses with RudderStack?

RudderStack, as a startup, had less community and developer documentation support compared to Segment. Their biggest challenge is competing against the behemoth player, Segment, which offers a wider range of features and has experienced significant growth. However, RudderStack compensates for this by providing excellent hands-on technical support. They focus more on engineering and less on product/sales/marketing. For someone leading product or marketing who doesn't have a background in the technical growth stack or analytics stack, Freshpaint may seem more appealing. It's easier to get started with given that you don't need to define everything upfront. RudderStack likely doesn't target this segment of users aggressively, as it has a slightly steeper technical learning curve.

How is the reliability of the platform?

When it comes to bugs, the main issue is the occasional quirks in data transmission and formatting from different sources. This often requires troubleshooting, but this issue isn't exclusive to RudderStack. Some of the bigger platforms handle it better. Let's take the example of defining a property to identify a user, such as their email address. The way you include that email address in the fired event might be interpreted differently by Mixpanel and Braze. Braze can record it on the user's record for future communication, while Mixpanel might use it for other actions related to the logged-in user. So it's crucial to format everything correctly for it to work as intended. Some advanced platforms, including possibly RudderStack, handle this transformation work for you to ensure alignment. However, there can still be occasional hiccups. I'm not sure if Freshpaint handles this any better, but this is an area where people often get confused when the numbers don't match their expectations.

How did you feel about developer experience and the level of documentation?

The developer documentation is super solid. It proved to be extremely helpful. The only minor issue I noticed initially, which improved over time, was their struggle to keep up with the fast development pace. Whenever they added new partners or made fixes, it took a little while for the documentation to be updated. However, their technical team would always inform us promptly through Slack or provide us with the latest information, so we never felt left out. Considering they were a rapidly expanding startup, it's understandable that their documentation wasn't always completely up to date, but overall, they were pretty solid.

How was the quality of their integrations?

Compared to Freshpaint, RudderStack had much deeper integrations, at least when we were looking at the early integrations. The connections were not just superficial, but actually fully connected. RudderStack seemed to have a vast library of vendors to connect with right from the start. On the other hand, Freshpaint was still building their library and would have limited us to a smaller number of vendors. It's possible that both platforms are more evenly matched now, but RudderStack definitely had a strong advantage at the time.

How was their support?

It was really good. When healthcare funding started to become limited, they had to lay off some employees because they were uncertain about the future. They wanted to extend their current funding as much as possible. In the end, I believe they managed to navigate through the situation successfully. We did lose one of our account representatives, but it didn't significantly affect our ability to use the product.

Are they very focused on the healthcare space as their predominant target market?

Yeah, I think RudderStack, just like Freshpaint, stumbled upon that niche accidentally. So they probably have similar stories. If my memory serves me right, the Freshpaint folks came from Heap. And at RudderStack, they also came from a similar background. They noticed that healthcare orgs were storing data on the CDP vendor's servers, but could be using their own cloud instances instead. So their main goal was just to act as the conduit, or "pipes," for this data. By doing that, they were able to reduce costs by 90%, and they could pass on 80% of those savings to customers while still making significant profits. That mindset was built into their foundation right from the beginning.

Do you feel like you made the right decision with RudderStack?

Yeah, one thousand percent. The only thing I might have done differently is considering Amplitude for our healthcare needs, as they came out with support for it a few months after we went with RudderStack. I really like Amplitude. Even though their CDP tool isn't amazing, it works really well with Amplitude, so it's worth it. If we had started a few months later, I might have gone with Amplitude, but I still think RudderStack was a good choice for us.

If I had known that direct-to-consumer marketing wouldn't be a big part of our future, we probably would have invested less and made some different decisions. But even though it hasn't been a big focus, RudderStack has still been really useful for us.

What areas of growth would you suggest for RudderStack?

I think it would be beneficial if they provided more assistance in structuring the data warehouse and handling the data transformation within it. Unlike their competitors, they seem to leave that responsibility to you, which means you have to figure it out on your own. Healthcare use cases are unique, so a one-size-fits-all approach like what Segment offers might not work well. However, one potential area for expansion is in the reverse ETL side, actively integrating with healthcare-specific vendors such as EMRs. Integrating with EMRs is challenging, so I understand why it's not their main focus at the moment. Maybe extracting data from systems like Epic or Canvas could be their next frontier.

Do you have any general advice for folks who are going through this decision process right now?

I would suggest appointing a highly skilled engineer to have singular responsibility and have them work closely with the RudderStack team, as we found that to be the most successful approach. We assigned one of our staff engineers to collaborate with RudderStack through Slack, and this helped us solve many of the issues we faced.

At first, I made the mistake of assuming that anyone with some JavaScript experience could handle this project. However, if you haven't worked on consumer application development before, it can be quite unfamiliar. Some of the people we hired had healthcare experience, but they lacked experience with consumer products. As a result, they struggled with understanding the object model, which is not unique to RudderStack, but it can be challenging for those new to consumer-facing projects. I'd recommend having someone on board who has used similar tools in the past, which will make the process much easier for them. Additionally, having a single point of contact who can build a strong relationship with the RudderStack sales team is probably the best path to success.