Zelis Provider Nexus
by Head of Product

`Provider Directory Management`   `Provider Data`

Details

Review Date	08/24/2023
Purchase Date	Q2'22
Implementation Time	3 weeks
Product Still in Use	Yes
Purchase Amount	150k
Intent to Renew	85%
Review Source	Elion

Product Rating

Product Overall	▬▬▬▬	3.5
Use Case Fit	▬▬▬▬	3.5
Ease of Use	▬▬▬▬	4.0
API	▬▬▬▬	3.5
Integrations		N/A
Support	▬▬	2.0
Value	▬▬▬▬	3.5

About the Reviewer

`Purchasing Team`   `Implementation Team`   `Product Oversight`

Reviewer Organization

`N/A`

Reviewer Tech Stack

`N/A`

Other Products Considered

`Ribbon`

Summary

- **Product Usage: The reviewer uses Zelis in their health insurance navigation company to identify in-network providers at various health plans and help customers understand their insurance plans.**

- **Strengths: The main strengths of Zelis include direct access to data from insurance carriers, overall API reliability, and established relationships with many insurance carriers.**

- **Weaknesses: Some weaknesses noted include outdated API structure, lack of transparency about data updates and availability, restrictions in API usage, and poor support service.**

- **Overall Judgment: Despite some obstacles, the reviewer feels they made the correct choice due to Zelis' unique data partnership capability and regards it as the best available option for their specific use case.**

So today, we're chatting about Zelis 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?

Yeah, we're a health insurance navigation company that assists individuals in finding, enrolling in, and navigating their health insurance plans. I run our internal and external products that help both our team of insurance navigators and our customers understand their health insurance plans.

How long have you been using Zelis?

A little over a year.

What caused you to look into purchasing Zelis?

It's a well-known problem in healthcare that getting accurate and comprehensive provider network data is challenging. In order to match people with health insurance plans and help them find doctors who are in-network, we have to know which doctors are in-network with their plans. And it's hard to do that, going directly to each insurance carrier.

What problems does Zelis solve for you?

Primarily identifying in-network providers at various health plans.

What set of requirements did you use to evaluate Zelis and other competitors?

We assessed the quality of the data by focusing on its accuracy. We wanted to determine if the data being provided was reliable. Specifically, we checked if the doctors actually exist, if they are practicing at the specified location, and if they are in-network with the insurance plans mentioned in the data source.

We also examined the comprehensiveness of the data. This involved looking into how many different networks are included in the dataset for various insurance carriers.

Additionally, we considered other factors related to API performance and the overall developer experience, how difficult it was to integrate and if it changed over time. The frequency of data updates was also crucial, as it directly impacted the quality and accuracy of the information.

Finally, the pricing model was also a significant consideration in our evaluation.

How did you go about vetting the quality and coverage?

We conducted spot checks, and it was quite challenging. Like many older healthcare organizations, they were not willing to provide pilots or limited production access. Instead, they only offered staging access, which had fake data. Therefore, it wasn't very helpful for detecting quality issues. We had to engage in many lengthy conversations to fully comprehend how the data is received, sourced, and how it is modeled.

What led you to go with Zelis?

I reached out to pretty much everyone, or at least everyone we knew of. There weren't really any other options with insurance carrier networks that were as extensive and comprehensive. That was the main consideration. In terms of quality, it's not fantastic, but it's still better than anything else out there. So compared to other options, it offers a good balance of quality and comprehensiveness in the data.

What are some of the other vendors that you looked at while you were evaluating Zelis?

We looked at Ribbon, and we came across a bunch of old-school healthcare companies that claimed to do what we needed, but when we actually spoke to them, it turned out they didn't. Ultimately, we found that our two real options were Ribbon and Zelis.

Zelis and Ribbon have different approaches. Zelis obtains data directly from insurance carriers, while Ribbon, at least during our conversation, did not have direct access to data from insurance carriers. Because of this, there are notable differences in the frequency, quality, and comprehensiveness of the data they possess. Zelis may not have everything, but they definitely have much more data compared to Ribbon.

So in terms of the coverage across carriers, you found Zelis to be superior. Did you also find that the overall data quality within those carriers was higher with Zelis?

Yes, but this was a year and a half ago, so things may have changed with Ribbon.

How did the different pricing models compare between Zelis and Ribbon?

Ribbon's pricing model was much better—more reasonable and logical. However, ultimately Ribbon didn't meet our needs, so we were left with no real alternative.

How did you find the sales process?

One thing I've noticed, and it's not just specific to Zelis, but it's particularly evident in the healthcare and health insurance industries, is that older API and data companies are very hesitant to conduct pilots, share test data, or let you try before you buy. They also avoid answering basic questions about customer churn and other ways to assess their product's quality. This creates a lot of friction, especially for companies that don't want to commit to a long-term contract for a product they've never tried, especially at high price points. As a result, the conversations and negotiations take much longer than necessary, with the companies trying to prove their value without actually providing any tangible proof. They rely on reference customers, claiming they work with top customers in the industry, but they don't actually connect us with them. It's really difficult to assess the quality of a product without being able to try it firsthand. I understand that these are production systems with real data, which can make it legally challenging to conduct tests, but there are ways to work around this. For example, a contract could include a one-month opt-out termination clause for convenience on both sides, essentially a pilot.

How was the onboarding process?

So, Zelis is basically a set of APIs that we use to integrate the data. The documentation provided is pretty good. However, the way the APIs are structured is a bit outdated. Our team thought it wasn't the most logical approach, considering current trends and how data components are typically modeled. Nevertheless, our engineers were able to handle it without much difficulty. So that part was manageable.

We did encounter some issues with query latency and speed, which were mostly on the provider's end. They had some strange limitations and restrictions on the number of requests we could make and how the API was set up, which posed some challenges. However, it wasn't too difficult to work around those obstacles.

Once we signed the contract and received a production key, we were able to access the data smoothly from an engineering perspective. However, when it came to data comprehensiveness, things weren't as smooth. Since we couldn't test it beforehand, we did a lot of auditing once we gained access. Unfortunately, we found that the data wasn't as comprehensive as we were led to believe. It was still better than anything else available, but there were definitely significant gaps.

And their support team wasn't very knowledgeable about these gaps, so it took us a long time to understand where the gaps were and why they were happening.

Was there a resolution?

On the query speed, mostly. On the data availability, somewhat. In a lot of cases, we had to set up workflows on our end to be aware of where those gaps were and then solve them on our own.

Are the main use cases that you're using Zelis for now still oriented toward knowing which providers are in which network?

We use it in multiple places within our platform because it has a lot of implications, but fundamentally, that's what we use it for.

Do you use any features that Zelis provides or ways in which they make their data available across different channels or mediums?

No, not really. We've had to add a lot of additional functionality to it in order to use it for different types of aggregations. We handle some resolution on our own, but overall, the structure they've provided is pretty straightforward.

How would you describe Zelis's strengths?

Their primary strength, and the main reason we use Zelis, is that they have direct feeds to most carriers. Although they don't do much data cleaning or make many improvements on top of it, we can handle that ourselves. The difficult part is actually obtaining the necessary credentials and permissions to integrate with these carriers. Writing the API integrations is not technically challenging, but gaining permission is. This is where Zelis shines, as they have established long-standing relationships with these carriers. In my opinion, Zelis's primary value lies in their unique data partnership.

Is the platform reliable and relatively bug free?

It's pretty stable. The actual content and data are not always stable, but as an API, it's generally reliable. We haven't experienced much downtime, if any at all, which is a good sign. But as I mentioned before, there are occasional gaps in the data. It's not always clear when an update is occurring or when the last update happened, so keeping track of the data's current state can be a challenge.

How would you characterize the documentation and the overall developer experience?

Based on what our engineers told me, the overall experience was fine. They found the developer experience to be satisfactory, and everything seemed mostly reasonable. Our engineers did have a few questions for their team, particularly regarding the data models and the functionality of certain queries, and they were able to find answers. The documentation was pretty good too. However, our team did have some frustrations with certain limitations they felt were arbitrarily imposed, such as restrictions on radius searching and available aggregations. Our team believed that these features should have been available but were not. Overall, though, the developer experience was still positive.

How do you feel about the account management and support teams?

Not great. They are quick to respond, but their responses are unhelpful, and it takes a long time to get the necessary details. The people who handle the inquiries don't have any knowledge about how the system works or the data involved, so we end up going back and forth with multiple emails just to figure out the basics. Eventually, they bring in someone who might have the information we need, but it often requires involving yet another person. Throughout this whole process, we already know exactly what the problem is, and we just want them to connect us with the engineer who can solve it. It can be quite frustrating.

Looking back, do you feel like you made the correct assessment in going with Zelis?

Yes. I don't think there are any other options for our use case.

Can you talk about some of the areas of growth that you might recommend for Zelis?

Yeah, I think there are three main things that can be improved. First, they should have a sales process that allows people to try the product before they buy it, or at least have a short period to test it before committing long-term. Second, they need to be more transparent about the data they have and don't have and how often it's updated. It was really difficult to get that information. And third, for the data product itself, they should enhance the API setup to enable more robust aggregations in the queries.

Do you have any advice for buyers who are selecting this type of product right now?

I think it's fairly axiomatic, but it's essential to clearly define the use case or cases you're currently purchasing for as well as future needs. During our search, we encountered numerous companies in related fields that cater to adjacent use cases. Unfortunately, we wasted a lot of time talking to these companies because we hadn't precisely articulated our own use case and its unique aspects compared to others in the market. If we had done that up front, it would have saved us time. There are plenty of alternative options and competitors to Zelis for different adjacent use cases. If you don't require extensive data coverage or frequent updates from insurance carriers, or if your focus is more direct-to-consumer oriented, defining the use case early on will ensure that what you're buying aligns with it.