Medplum
by Head of Product

Headless EHR   Electronic Health Record

Details

Review Date	07/10/2023
Purchase Date	Q1'23
Implementation Time	N/A
Product Still in Use	Yes
Purchase Amount	N/A
Intent to Renew	N/A
Review Source	Elion

Product Rating

Product Overall	████████	5.0
Use Case Fit	████████	5.0
Ease of Use	███	2.0
API	██████	4.0
Integrations	█████	3.0
Support	████████	5.0
Value	████████	5.0

About the Reviewer

Purchasing Team   Product Oversight

Reviewer Organization

Specialty Practice   Cardiology

Reviewer Tech Stack

Healthjump   Retool   PointClickCare

Vanta   Auth0   Bonfhir

Other Products Considered

N/A

Summary

- **Product Usage: The user is using Medplum for robust data centralization of heart disease patients, with Medplum hosting their data and supporting security and performance measures.**

- **Strengths: Medplum's open-source technology, alignment with HL7 FHIR standards, comprehensive documentation, scalable platform, HIPAA and SOC2 compliance, flexibility of access control lists, and the ability to import synthetic data for testing have been highlighted as strong attributes.**

- **Weaknesses: The front-end tool has limitations, and there were challenges with discrepancies between different FHIR standards and versions supported by Medplum.**

- **Overall Judgment: Even though some discrepancies and challenges were encountered, the user is satisfied with Medplum's robust data handling, comprehensive documentation, security protocols, and scalable platform.**

We're discussing Medplum and its application within your organization. Could you briefly describe your company and your role there?

I'm the Head of Product at a healthcare technology company that develops robust, value-based care solutions for cardiology practices, primarily independent ones. We help enable a network of cardiology practices with our product and tech-enabled services designed to enhance the performance and quality of care for heart failure patients.

Great. So, it seems like you decided on Medplum some time ago. When did you finalize this decision, and how long have you been utilizing it in production?

To be completely transparent, we haven't launched it in production yet, nor have we technically purchased it. We've been actively working with Medplum since March, so it's been around five months of close work with the platform. We're exclusively operating in a cloud environment hosted in AWS. We decided to build upon Medplum's open-source technology back in March, but we don't anticipate requiring a paid version of Medplum's offering until we launch our product in production later this year.

It's interesting that you've managed to defer the purchase process during development. Was there anything special about your negotiation that allowed this, or is it a standard practice?

It's fairly standard, and I believe it's a core part of their value and principle as a company. They position themselves as an open-source-first offering, allowing innovative and bootstrapped companies to build healthcare solutions rapidly without initial cost, testing the quality and value before making a commitment. We didn't find anything directly comparable during our search.

Understood. I'd like to learn more about the product: how you're utilizing it, building upon it, and the usefulness of the primitives.

Sure, our primary goal is to centralize all information for patients with cardiovascular disease across their care continuum. Our customers are using existing EHR technology in their cardiology practices, but it's not the only relevant part of the patient journey for value-based care. We source data from multiple places - claims data, hospital admission, discharge, and transfer events, and EHR data in both specialty and primary care settings.

We needed a central service for these disparate data sources and wanted to use the best storage solution available. Medplum aligns well with the modern HL7 FHIR standards. They provide a comprehensive, developer-focused support system with detailed documentation, helping us get started quickly. We primarily use Medplum to host data from external sources. We regularly take extracts from an EHR, convert them into a standard FHIR format, and use Medplum as the primary host. The built-in security and performance measures took care of our concerns with handling and storing sensitive data.

You've mentioned security and performance as notable attributes of Medplum. Can we delve into those, starting with performance? How does Medplum improve the performance of searching, accessing, or writing data for you?

Some of this still needs validation. However, we have partners who've used Medplum for enabling similar solutions and found their platform to be scalable and supportive of their company growth, which gave us confidence. We have

generated a significant number of resources in support of our product development process without any issues in storing, accessing, or computing complex operations on this scale of data. Once we move to their cloud solution for production, we'll benefit greatly from their support and will not be reliant on internal dev or cloud ops resources for maintenance and upgrades. But for now, we're largely guided by the positive testimonials.

Yeah, I'd love to discuss the security component. But quickly, you mentioned that you're self-hosting now, but are looking to transition to the cloud or to Medplum hosting. How do you perceive the value of being self-hosted versus Medplum-hosted?

We spent about a week assessing both options. A portfolio partner company of ours quickly decided to use the hosted route for production, primarily due to timing and risk considerations rather than cost-effectiveness. We benefited from knowledge and lessons learned through our shared technology leader. One advantage of using an open-source tool like Medplum is that we don't pay until we go to production. Estimating the right tier from Medplum's pricing model, without knowing how fast we will ultimately scale, was a challenge. We compared the average human capital cost needed to reliably manage this product, factoring in potential downtime and upgrades, and found that it would be much more expensive to handle everything in-house rather than paying Medplum for support. We plan to reevaluate this decision as we grow and have a plan for transitioning back to a hosted solution if needed.

That makes sense. I'd love to understand better how the security system works, especially the access control lists (ACLs), and how you've implemented security controls for your workflows.

We will absolutely benefit from Medplum's HIPAA and SOC2 compliance that they have incorporated into their product. They've also implemented penetration testing and other security controls. From an ACL perspective, we appreciate the flexibility of FHIR and Medplum's full embrace of it. We've kept our access controls relatively simple with just two levels of permissions: one at the provider level and one at the admin level. Admins have the ability to view and manage all patient data while providers are limited to data for patients and encounters affiliated with their organization. We use Auth0 for user authentication and identity, which also enables a shared authorization layer with Medplum to reliably control user permissions throughout our product. Our current setup is not overly complex, but it doesn't need to be given our current use case. Medplum also supports patient-level data access, but we haven't explored that yet, as it's not a requirement for product offering in the near term.

Moving on to the actual build process, it sounds like they take a very developer-focused approach. How are you finding the core APIs, technologies, and SDKs? What's it like to build on top of Medplum?

Getting started with Medplum is quite simple if you have a basic understanding of FHIR. The ease of generating and uploading synthetic data is very useful to begin initial testing and product design. Medplum's primary value and focus is clearly centered around supporting a robust backend solution, and, as such, their frontend application can often be limited and challenging to use. We determined early on in our testing that any complex frontend internal workflows would need to be built within our own tooling.

In the beginning, we used synthetic patient data to test our backend algorithms and create working prototypes of our product capabilities. This process only took about three weeks. We have faced occasional challenges with discrepancies between different FHIR standards, versions, and capabilities supported by Medplum, but we anticipate these hurdles will diminish with future upgrades planned in the coming months. On the self-hosting front, it took about a day and a half for our developers to deploy Medplum in our cloud environment. Maintenance has been minimal so far.

Could you explain what you meant by needing to reimplement many of the front-end components?

I believe I was referring primarily to the basic frontend application and tooling that Medplum has built to help users get started. It's essentially a database management layer using their React components, but it doesn't have much flexibility or capacity to handle complex tasks. We ended up using very little of their React UI component library in our product, relying instead on another open-source technology that works better for our needs.

You mentioned that someone else is building open-source on top of Medplum, or specifically FHIR?

Yes, we have taken advantage of an open-source toolkit developed by a group of health tech incubation companies affiliated with our shared VC firm. It provides libraries and various tools that support common health tech use cases to help accelerate and improve reliability of developing solutions that utilize the Medplum API.

Shifting gears a bit, how do you handle pulling in exogenous EHR data, in relation to Medplum?

We've yet to fully assess Medplum's capabilities in this area. We plan to use their subscription capability for ADT feed ingestion. For our EHR integrations, we decided to maintain a flexible middleware between Medplum and our EHR extraction vendor due to the anticipated challenge handling anomalies in the EHR data ingestion process. We've put a lot of effort into mapping the clinical data into FHIR-standard resources, but our approach will likely remain quite manual in the short term. Eventually, we expect to also use Medplum's bot framework for real-time webhook-like integrations.

How did you find the sales process with Medplum?

We haven't had a sales process yet. We're very aware of the pricing model and we plan to engage with them soon, but we haven't gone through the process fully yet.

Did you consider other products?

We did review and speak with several vendors with similar offerings. Some quotes we received were cost-prohibitive and unclear how they would scale with us. Redox was a contender, but we didn't find their pricing model to be feasible for our phase and maturity as a company. Medplum, on the other hand, was a point solution that was relatively inexpensive and didn't require up front consulting fees to get started.

What do you like most about Medplum?

Their documentation is not only comprehensive but also incredibly user-friendly. They really know this space well. It's been enormously helpful for our team in developing a working knowledge of the best practices in working with FHIR. They also have a Discord server that they offer their developers and partners which is completely free. The responsiveness from their CEO is truly amazing.

What do you dislike most about Medplum?

Their front-end application, certainly. it seems to be more tailored to showcasing basic usage of their UI component library rather than supporting comprehensive data management requirements. There's a sizable gap between the potential of what you can build using Medplum and what you experience if you first begin testing with this sample application that Medplum provides.

Is there anything else worth mentioning?

Yes, we were initially discouraged from relying on Medplum's alternative GraphQL API protocol for engaging with Medplum-hosted FHIR resources. At the time, it seemed to be less reliable and lacked sufficient documentation. However, in recent weeks, our team has started investigating its potential more. It, in principle, should dramatically reduce the number of synchronous calls and simplify the data post-processing required for many common use cases that combine data points across multiple FHIR resources.

Generally, though, Medplum seems to be very much at the forefront of technology trends and I suspect they will be fast followers as better technologies and best practices come out.