

Medplum

by Principal Engineer








Headless EHR

Electronic Health Record

Details

Review Date	10/04/2023
Purchase Date	Q4'22
Implementation Time	4 months
Product Still in Use	N/A
Purchase Amount	Consulting fees for self-hosted enterprise plan
Intent to Renew	N/A
Review Source	Vendor

Product Rating

Product Overall		4.5
Use Case Fit		5.0
Ease of Use		3.5
API		5.0
Integrations		5.0
Support		5.0
Value		5.0

About the Reviewer

- Purchasing Team
- Implementation Team
- Product Oversight

Reviewer Organization

- Virtual-First Provider
- Men's Health
- Women's Health
- Behavioral Health

Reviewer Tech Stack

N/A

Other Products Considered

N/A

Summary

- **Product Usage:** The reviewer mainly used Medplum as a back-end solution and data model to act as the source of truth for patient information and coordinating with other systems.
- **Strengths:** Medplum is praised for its commitment to FHIR as a standard, its comprehensive implementation, opportunities for integration with the wider ecosystem, and its highly knowledgeable team.
- **Weaknesses:** Medplum’s user experience components are deemed weak, as they did not integrate well with the reviewer’s existing UI framework.
- **Overall Judgment:** Overall, the reviewer thinks choosing Medplum to create a singular, well-planned clinical data repository was the right decision. They were particularly happy with the support and account management.

Review

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

It's a telehealth company that owns multiple brands focusing on various health conditions. People would sign up for our services based on their specific health needs. We would then guide them through the process, starting from initial patient engagement to clinical management of a large number of patients. We also owned automated and high-volume pharmacies that were integrated into our services. We had different brands for different health conditions, some of which were quite complex. Our mental health brand, for example, provided not only medication but also psychiatric help, following a comprehensive care model.

In terms of my role, I was primarily responsible for the clinical care aspect as a principal engineer. I identified appropriate technical solutions for the business problems raised by our product management team. One of the challenges I faced was how to unify patient data across multiple brands, especially as we acquired different companies. We had to integrate legacy systems from two startups that had been operating for over five years. My main focus was finding the best technical solution to establish a unified view and streamline workflows for our patients.

What drove the need for a tool like Medplum?

The business problem at hand was how to merge a substantial system that already had its own patient view and data model with our existing systems. These systems held patient data in different ways and had varying structures. We aimed to find a solution to unify this data so that different views were not required. Our goal was for providers to access patient information without having to navigate multiple systems with inconsistent data. Additionally, this unification would enable us to analyze aggregated information across all patients, allowing us to identify correlations between specific problems and treatments. Ultimately, the objective was to achieve both data unification and operational unification.

What key requirements were you looking for as you evaluated Medplum and its competitors?

My main driver was to find a unified model and language, a ubiquitous language, for organizing patient information. Different stakeholders have different interpretations of terms like "prescription," which creates confusion. Engineers without healthcare expertise had made decisions on how data was structured, labeled, and connected, resulting in different systems using different terms for the same thing. I wanted a technical solution, like Medplum or another EHR, to establish a standard language for our clinical services. By adopting existing standards, we could avoid wasting time arguing over naming and structuring as the team grows.

Another requirement was a headless EHR that our engineering staff could easily deploy and integrate with our infrastructure. We preferred self-hosting over using an EHR as a service, to maintain familiarity with and control over the technical implementation. It needed to seamlessly integrate with the rest of our infrastructure, as it is a major component of our infrastructure.

What drove your decision around self-hosting versus EHR as a service, and what particularly stood out about the Medplum headless EHR?

It's easier for us to go through our technical review if we self-host rather than use an EHR as a service. The latter option would require involving more entities like Legal and Compliance, which would have made it more challenging for us

to push through. Since it's a crucial part of our business, we, as an engineering culture, preferred to build our own solution rather than buying one. If we had to not only buy but also rely on someone else to manage it, it would have been more difficult for us to agree on it. Luckily, we had a highly competent infrastructure team who could handle the self-hosted option without additional burden, so it made sense for us.

What were some of the competitors you looked at?

Our product management groups in both our parent and acquired company thoroughly evaluated various vendors in the EHR space over the course of several years. However, they had not considered Medplum. I was primarily interested in utilizing it as a unified patient clinical data repository rather than a complete EHR, as I believed it was a foundation we could build upon. This was especially important because our unique workflows didn't align well with other EHRs, which at that time didn't adequately support telehealth integration and streamlining workflows for providers.

Medplum hadn't been on their radar, mainly due to its engineering-focused nature. It's designed to be a headless solution that can be customized through engineering development. It integrates with our existing systems, user experience, and workflows. It's not entirely custom made, more like off-the-rack with room for adjustments. After considering several other vendors (which I can't recall at the moment), we determined that Medplum ticked all the boxes, and we decided to move forward with a pilot.

One of the key factors we liked about Medplum was its open-source nature. We also appreciated the expertise of the founding team, as their background and experience aligned well with our needs. When we spoke with them, it was evident that the CTO understood both the technological aspects and the business challenges we were addressing. This was a significant advantage compared to talking to a sales representative who might lack the necessary technical knowledge. Additionally, we conducted a code inspection of their open-source product and were impressed with its high technical quality and how nicely it was done.

How was the sales process and the onboarding and implementation?

I thought the Medplum team delivered on their promises. During the sales process, we essentially did a proof of concept with them. They helped us map one of our core systems into Medplum and demonstrated how to do it. They also worked on creating a user experience that matched our existing one for clinicians. The sales process was very hands-on, involving a lot of show and tell, which convinced our CTO to support Medplum.

During onboarding and setup, they again followed through on what they promised. They collaborated with our infrastructure team and made adjustments to their self-hosted options to align with our infrastructure guidelines. They also provided valuable insight into how we should approach the clinical domain, helping our engineers and product managers understand how to model aspects we were concerned about. Overall, I think they did an outstanding job, and it was a great experience for me.

How does Medplum fit into your clinical workflow?

We had three pilot projects that were a bit disjointed due to internal decisions that were not necessarily logical but made sense in our culture. The main project was unifying two major patient bases. To accomplish this, we created a master patient index (MPI) using Medplum and a FHIR-based model. This allowed us to integrate data from our existing systems and create a comprehensive patient database. The goal was to offer this data, which was now in a standard API, which is the FHIR format and standard, and integrate with external systems such as insurance and patient deduplication services.

Another pilot project we explored was using the tasks and workflow features of FHIR to standardize care plans and workflows. This project was still in progress when I left, so I'm unsure of the outcome. The focus was on lab results, where we integrated our lab provider with our enterprise event bus. When we received lab results, it triggered tasks for physicians to review. This pilot involved integrating with our existing provider workflow and seeing if we could migrate from our legacy system to Medplum or add features and functionalities to our system using Medplum as the driver.

The third pilot involved multiple tied pharmacies and the need for a unified formulary. It was challenging because different stakeholders had different interpretations of what a formulary meant. Essentially, we needed a master list of drugs and treatments that clinicians could prescribe. This required consolidating various drug databases into a unified structure that all our systems could use. The pilot was incomplete when I left. We were exploring whether Medplum could serve as a unified medication repository.

Were you using Medplum primarily as a back-end solution and data model and less so as a front-end application?

Yes, absolutely. The main objective was to establish Medplum as the ultimate authority on patient information. We wanted to ensure that the providers had all the relevant patient data at their fingertips and then take advantage of other Medplum features as needed. Essentially, we aimed to make Medplum the center of the spider web of these complex systems.

What do you see as the key strengths for Medplum?

I think the core strengths lie in their commitment to FHIR as a standard and their dedication to implementing it comprehensively. While the healthcare industry can be complex, this approach allows for easy understanding of specifications. The primary advantage of this FHIR-first implementation is the increased opportunities for integration with the wider ecosystem once your system adopts it. Additionally, the Medplum team is highly knowledgeable and capable, ensuring that any questions or concerns are promptly addressed.

Do you see any areas of weakness?

They have user experience components that support FHIR, but it's not their strongest area. We didn't end up using their UI components because they didn't integrate well with our existing UI framework. Additionally, it was just as easy for us to have our existing components communicate with the back end rather than using theirs. This was a bit disappointing for me. It would have been nice if they had a more user-friendly prototyping tool, but I understand that it's not their main focus. Ultimately, what we really wanted was their well-crafted back end.

How reliable and stable is Medplum?

We had no issues whatsoever at the pilot level.

How was the developer experience?

We built everything using their APIs. We also made use of their bot. Their architecture was great for extending functionality and hooking into events. We really liked their bot framework, as it made it easy to integrate webhooks into our existing system or connect to AWS lambdas running in our environment. It was very effective.

Their APIs were solid and reliable. There is a FHIR standard that defines what the API should be, and Medplum complied with it. For more complex queries, we used GraphQL, which was also defined by FHIR. It worked well for us. The

documentation was pretty good, and we didn't receive any complaints from our developers about the overall developer experience.

How was the integration experience?

They've thought about integrating into testing and staging, and being multi-tenant from the beginning allowed us to deploy and create test environments easily. We could even do it on a feature branch level. This turned out to be one of the key features of Medplum—they understood the needs of larger organizations like us with more complex engineering workflows. In addition to the production version, they also understood the importance of having an integration environment where we could test things on a larger scale and integrate with multiple systems. They seamlessly integrated with our complex architecture, meeting our needs for continuous integration and deployment.

How is support and account management?

Exceptional. Very, very good.

Do you think you made the right decision on going with Medplum?

I think so. While there were many differing opinions on how it should function and be engineered, I believe choosing to create a singular, well-planned clinical data repository was the right decision.

Do you have advice for people making similar decisions at other organizations?

FHIR is really impressive in many ways, but it can be difficult for people to learn and understand. The FHIR spec is quite abstract and can be challenging for engineers. That's why it's important to seek help from those who have previous experience with FHIR implementation.

The benefits of adopting FHIR standards are significant. In the case of using Medplum as the EHR, I would suggest creating adapters around it to simplify the complexity of the user experience. These adapters would only handle the specific data needed for your organization and would communicate with the FHIR server. An example would be a small, lightweight adapter whose sole purpose is to translate the required data into code for Medplum's API. This approach reduces the cognitive load on other developers, since they don't have to become experts in the full FHIR API. Instead, they can integrate with the simplified API they've created.

My advice is to focus on simplifying the integration process through tools and leveraging the expertise of those who have experience with FHIR implementation.