

# Elion

## Buyer Review Transcript

Product Reviewed	Medplum
Integrations Reviewed	Acuity, Candid Health, Zoom
Other Products Mentioned	Healthie, Athena
Date	06/27/2023
Expert Role	Principal Consultant
Organization	AlleyCorp Nord
Payment Model	Value-Based Care

**We're talking about Medplum and how it's used by your company, specifically, in helping other companies implement Medplum. Before we jump into that, could you give just a brief overview of your company and what your role is there?**

I'm a Principal Consultant for AlleyCorp Nord. We specialize in helping very early stage startups get to their initial MVP and we try to help those startups with early concepts, product implementation, and strategy all the way to series A or B. We really focus on bringing engineering resources and helping companies in developing and delivering a product in the early phases. And in the event that they succeed, we help them transition to other partners and companies. We want to stay focused on the early stage. And we provide fractional CTOs, staff, senior, and intermediate engineers, and product designers as well. We've chosen to specialize in three different areas: one is healthcare, which is where we have our largest portfolio so far. And we also specialize in robotics and artificial intelligence.

**That makes sense. And so it sounds like you have worked with a couple of different companies for whom you've implemented Medplum. I'd be curious to see when those decisions or purchases happen? And how long have you been working with Medplum with these two companies?**

So from the first time we encountered the product until now, we have about two years of experience with Medplum across two different clients. Both times, we were involved in selecting the product and recommending Medplum. Since we work with early stage startups, the decision was made very early in the process, because it's really to support our initial implementation. Given the fact that we are very much at the beginning of the life of those companies, they sometimes do not even have a strong vision of what the product or business model is. Most of

the time they haven't found it when we start building the product. We know that what we do is temporary; it has to be a good balance of efficiency versus scalability. All the while, knowing that if they succeed, we're going to need to revisit every single decision because we do not prioritize scale, we prioritize "let's get something out of the door" because that is enough to start operating and convince investors to go forward.

**With that in mind what made you choose Medplum? Maybe it's worth digging into the two different use cases. What are the conditions that made Medplum the right choice for your companies?**

Yeah, I think there are two levels for that decision. The primary decision we made in most cases was to use a headless EHR, based on FHIR and then there's Medplum the product in that category. When we started to work with health tech startups, we did a couple of implementations using other EHRs. So you know we did a couple of implementations using Athena, we did a couple with Healthie, and other EHRs. And we managed to do a successful implementation using those tools, but we encountered a ton of friction using these products, because what we needed them to do was not their primary use case. So that's the main thing we uncovered is that most of the non-headless EHRs tend to have a very rigid workflow that is geared towards their intended target and it works for them. So it's great if you're in a provider group setting, right? With the majority of EHRs, each of them has their own niche and specialty they target, so they implement a workflow that works for those specialists and if that's what you need, I would go ahead and pick them. They bring a ton of value - it's very easy to configure, there's a ton of training material, etc.

Most of the time the companies we implement for are value-based care providers, which is important because now the workflow is very different. Most EHRs have a very well-defined workflow, where you have an appointment with the patient, you get them checked in, you have an encounter, billing, and acquisition of patients is either direct contact, or through a referral workflow. In both cases where we implemented Medplum, the workflow was very different. For example, we acquired patients through data ingestion, so we process claims data on Medicare, Medicaid, data from payers, and ADT feeds from different hospitals. We gathered data, reproduced the patient chart, and we started engaging with the patients and also the regular provider for those patients.

The other place where there's a ton of friction with a traditional EHR is in the patient-provider modeling. With a traditional EHR, you have your patients, and you have your internal clinical team, which can have different roles in the EHR, but it's otherwise fairly rigid. For instance, when you have the messaging platform integrated, it's always a message about the patient to the internal clinical team. In both our cases, we had a third role - a provider who isn't the specialist or the PCP of the patients. And the product we built was used by the patient and their provider as well. Trying to implement a solution like that with a traditional EHR was really painful because we were fighting the workflow of an EHR that did not fit our intended use cases.

**Do you feel that the things that required the most work were on the value-based care component, so bringing in external data and creating patient profiles out of that and then having your panel that you were working off of, or was it more on modeling different provider-patient relationships, or is either one of these challenges sufficient?**

Yeah, it's both of them, but they happen at different times during implementation. Like, in data acquisition. you're sitting there and then you realize that whatever workflow you need at that point is different from what the EHR has in mind. And so you, you know, in certain cases, you have to create phantom resources in the EHR to make them happy in terms of being able to sign off an encounter, so it's a bit of a hack. I've done tons of product implementations in my career, and every time you find yourself fighting your product too much, it never leads to a good outcome. Because then you start to build things outside of your EHR, because they're not really meant for extension. Like they often provide, you know, a couple of custom fields here and there, a couple of parameters on there, but what you end up building is another assistant that then communicates with the EHR and so now you have to look at multiple applications and it starts to become a bit of a mess. And what you're doing is pushing your technical limitations to your employees and operations, and sometimes it makes sense to do that. But sometimes, the trade-off isn't worth it. You have to know where it makes sense to invest a bit more in engineering to provide a better experience for the users.

**So from your perspective, if the workflow deviates in a substantial way from the traditional provider-patient, cash pay, fee-for-service, encounter-based workflows that EHRs tend to be set up for, then a headless EHR might be the way to go. But how did you make the decision to go with Medplum in particular?**

Yeah, so in the past when we did a bunch of implementations with EHRs, we were not very satisfied with the implementation we were providing - not great, a ton of friction. So we started to investigate what the alternatives to that model are and this is when we landed on FHIR as a specification, as an API, as a norm.

And we saw that FHIR brings a ton of value out-of-the-box. The data model is much better than that of any EHR we've seen so far. There's a lot more care dedicated to the way data is modeled and there's no workflow prescription out-of-the-box within FHIR. FHIR does not provide an order of things, they provide resources and definition, but then it's up to you in the implementation to orchestrate those resources in the way that you see fit. And there's a ton of room in the FHIR spec for extensions and external resources. So we said okay, let's investigate that thing. The challenge with FHIR is the complexity, the spec is quite big. If you want to do a good server implementation, it's a lot of work.

So we set out to find a partner, a company that was doing a lot of this for us. And we had a look at different products within that space both on the bigger side (at the moment, the three big cloud providers have a FHIR server offering) and then smaller companies like Medplum and a couple of those. I'll have to remember the ones we investigated, but we quickly landed on Medplum for a couple of reasons.

The first one is it felt like they brought a more modern open source way of doing things, like something that feels more like dev tools than other products in the space. And I think to me it's a big difference. Their product is open source, you can have the code, you can make contributions, you can self-host it, they have no problem with that. There's no limitation in the product if you do that. They can self-host as well. That's their business model and that's great. We've used this and I think it's good.

Their pricing is fairly transparent and public on their websites. All of those elements are quite different from what the traditional EHR model is, where you have to engage with the sales rep and then you go to an entry cycle and there's no transparency in what they're doing, neither in the product nor in the sales cycle. And Medplum is the opposite. So it made us feel quite good about choosing them. It's an approach that we've seen in other industries but I believe they're trying to bring into the healthcare space.

And there are technical elements that made us choose Medplum as well. It's an open source project that we can have a look at, and their tech stack is one we're very familiar with. It's built in Typescript, PostgreSQL, nothing fancy. Everybody can read the code and understand what they do. And it's exactly the same stack we've had in other projects as well. So when we have a question about the product or there's a behavior we don't really understand, I can have anybody on the team have a look at the code and then understand what's there, which is very comforting. It's very transparent in the way it processes things.

And I think maybe some other competitors are also open-source, but Medplum has built an authentication and authorization model on top of the FHIR specification. So, FHIR by itself does not have anything around authorization and or authentication. SMART on FHIR allows you to navigate from one EHR to the other, but it's very limited. Right now, to the best of my understanding, it's up to FHIR implementations to apply an authentication and authorization model.

And what Medplum has built within that space was a very important element of choice for us because it allowed us to not have to implement the backend on our product. And that was a crucial piece because suddenly there is this piece, I don't need a gateway in front of Medplum or even the patient app in order to use Medplum. Which is not the case for, if you go to the healthcare API in Google, you have to build a layer on top of it to authorize because what the product has is not enough.

In Medplum it's not like there are limitations and we can talk about this, but it was good enough that we did not need to worry about implementing that piece. And so for us it's a big win. We don't need to build a backend. We don't need to put ourselves in front of the EHR, we can just expose the EHR so that all the client applications are connected to it.

### **What do you think makes the authentication layer at Medplum unique?**

Yeah. So one is they built a very sensible architecture around this. So they augment all the existing FHIR resources with their own Medplum resources. But the good thing they've done is

they keep using the same FHIR API. So they just add a couple of entities to the FHIR model. So again, it's not a new API, it's the same API that just has new resources to manage that thing on top of FHIR.

And two, out-of-the-box Medplum acts as an OpenID connect provider. So Medplum acts as an identity server that you can use. And three, that open connect provider supports any kind of external OpenID connect provider as well. So you can map any other identity server you want and you can map it to Medplum, and Medplum will handle anything that implements the OpenID connect token exchange protocol. So you can use anything as your authentication provider and then map it to a user in Medplum, which in turn is going to map it to a FHIR profile.

So it's very easy and both our implementation, for example our external authorization was done to Auth0, where we had passwordless and biometrics support done by Auth0. And then we mapped those people back to Medplum, and there's very few things that you need to do to have that working. Internal people log in using Google SSO with their workspace accounts. And all of that flows in, normalizing as employee users and a FHIR profile internally. So that's for the authentication piece.

On the authorization side, they provide quite a flexible model because there's always two things. There's a very simple thing, which is, do you have access to a resource or a specific field in particular? This one is easy - is the user allowed to read patients, practitioners, medications, etc.

But the more complex aspect of authorization is always authorization based on data. And in both instances, since we're doing value-based care and we have a multi-tenant product, we serve different organizations and we need to compartmentalize the data with the organization. Everything revolves around the patients and the records that we attach to the patients.

So we needed to model quite a complex relation - an external practitioner maps to one of several organizations. A patient is managed by one of those organizations. And so when that external practitioner needs to log in, they need to access the patient but also the risk assessment of those patients and then the medication and the chart and everything around it. So it's quite a complex model and we managed to implement it using Medplum.

The way they do it is they primarily use FHIR compartments to model their authorization, which is very smart. I believe it's a very good choice because they naturally exist within the FHIR spec and they are naturally created by the spec, and they extended that model to add an organization compartment, which is not natively in the FHIR spec unfortunately. But they have added a way for us to define for all of these resources, a compartment for organization. And so we can map that complex third party organization healthcare model to those businesses.

It's not magical. There's a bit of complexity there. We've had to have a couple of webhooks. They support the FHIR subscription model completely so we can have webhooks for everything, which is great. And so we've had to implement a couple of webhooks where we needed to update a couple of compartments for related resources here and there. The bottom line is we

didn't need to put something in front to manage our authorization. We can just use Medplum, even for those more complex cases.

**When you're thinking about doing the EHR implementation and you're like, "Okay well one of the traditional EHRs, the workflows just don't make sense, I think we'll need to build something from scratch." And then you learn about FHIR servers and open source platforms and things like Medplum. How do you think about the delta and lift and effort and work that you get from something like Medplum versus trying to build it on your own? What is the difference in implementation time?**

Okay. I had that conversation with the client very recently. Actually, just as an aside, they are somebody that built a custom EHR and now they're facing certification problems and they are looking at Medplum to try to get back to something a bit more standard. That's interesting.

To me the decision at that point was very quick to make. It does a lot - there's still a ton you need to add on - but whatever they provide is quite a lot. I mean for me the biggest lift is FHIR itself. You have a data model that is mature, very well-thought-out and until now we haven't met a use case that we could not solve within FHIR. Where we saw those FHIR resources and we're like, "There's no way this is going to..." We always find a way within FHIR to solve our problems.

And what it gives you back is so much maturity. I personally believe that if you were to go the custom EHR route, you're going to start with a data model that is much simpler than FHIR and then you either, if you're good, you're going to end up with what FHIR has done but you're probably not that good so you're going to end up with a lesser version of it.

And then once you have that data model, then it gives you the full API, transactions, storage, authorization, authentication, all of that is done. So now you can focus on your app and your workflow. So the lift is huge. The downside though was compared to a more traditional EHR, there's still a ton of stuff to do. You need to build your app.

What they call the Medplum app, i.e., the admin panel they have on top of Medplum, I think it's one of the weak points of the product because you can't really have either internal operations team or clinicians use it directly. It's way too technical, way too limited in what it does. So you need to build other tools on top of it.

So that's really the challenge then because even though it does a lot, you still have a ton to build on top of it, and the FHIR data model is complex. There's no way around this documentation. We've met a lot of people that were really intimidated by FHIR, "How do I start? Why do I need that? Do I need this much complexity?" It's really a challenge.

It's funny, we have that even in our implementation team. I remember onboarding new developers and their initial response was, "This is very complex, do we really need to do this? I don't understand." What's next though is you take those same people six months later and you do a poll, and what you get back is, "FHIR is really good and we're very happy to work with FHIR." So to me it's a bit like jazz, right? It's really good but you have to get educated a bit and to make an initial effort because it's not easy and it's not accessible directly. That's a big

challenge. And I think that's a big challenge that a lot of those like Medplum and others are facing in the market right now. It's just the setup.

We're trying as well at ACN, we're working on a couple of projects as well to try to both from an implementation perspective and also from a training material perspective to try to help with that gap.

**Yeah, a lot of times with great dev tools, great programming languages, great frameworks, it takes some time to get used to perhaps some of the magic, some of the complexity that comes through. So I think that makes a ton of sense.**

At the moment, for one implementation, we're also experimenting with a no code tool, specifically Retool, as the way to build internal tooling on top of Medplum. Trying to lower the barrier a bit to build apps on top of it.

**Yeah, I'd actually love to talk about integrations. So the thing that everyone thinks about when they think about FHIR platforms is, "Okay, I'm going to store all my clinical data," but there's also a lot of non-clinical data. There are also a lot of integrations, things that need to be tied in. How do you think about Medplum and its ability to integrate with different resources? Obviously you probably have to write code. I'd be curious if you can talk through maybe some of the integrations that have been interesting. What's been intuitive, counterintuitive?**

Yeah, so I see two different kinds of integrations that we've built. One is really on the pure clinical data side, claims, ADT, things like that, mapping them in. And then we built a couple of integrations on the workflow side. So we've integrated for example, Acuity as a scheduling solution to book appointments, because it was easier to manage within our implementation. We integrated Candid Health for example, for being able to send coded encounters for billing.

What else have we done? Zoom, some other stuff. I would say overall, the tool doesn't provide a lot, but the foundation is good because it gives you on one side the FHIR API, on the other webhooks FHIR support. And so it's quite easy to handle it at this point.

Also, specifically because the FHIR data model is "Fast Healthcare Interoperability Resource", the interoperability aspect is very well-developed and has support for external identifiers, extensions, all of that. And also because, to your point, FHIR is not just the clinical data model, it goes way beyond. It supports modeling financial resources, so there's support for clinical research measures, qualitative and quantitative analysis.

It doesn't cover everything of course. But it's still big. As of the latest revision, it has 157 resources defined. So it's quite extensive - not perfect, but it is good. I think the most challenging part of FHIR and specifically in the context of data integration was the terminology. So you get data that has some ICD-10 codes for example, and data quality is what it is. So you have sometimes codes from five revisions ago that have been duplicated or whatever, and stuff like that. And then you need to map it internally.

And so in all cases we wanted to normalize as much as we can because the idea was let's take data in a form that we can and then normalize it internally so that we can work on the problem. And so you can choose, for example, to normalize on SNOMED, which is great but the mapping between ICD-10 and SNOMED is not trivial, and SNOMED by itself is not a simple system.

So I think the biggest challenge I see so far is really terminology support. We haven't found a great terminology implementation that we feel good about. I think we're still lacking a lot of knowledge around terminology as well because it's getting very technically advanced to understand precisely what the terminology is. So that is the main challenge we've had I think.

**I know you were involved in the procurement decision, were you also involved in the sales process? How did you find that process overall? Was it relatively easy? It sounds like pricing was public - was it fairly straightforward or were there hoops to jump through?**

I think that was also part of making us feel confident about our decision. Since Medplum is a small company, the people you meet are great. I don't know how they're going to evolve, but at the moment, you meet people that know what they're doing, that have great experience. So you do not have the typical salespeople that don't know about implementation problems that you have to go through. We've only had interactions with great people - with Rahul, Cody, and Reshma - those are the people you want to work with.

So it was great. I loved their approach, the sale was done through Reshma, and she was very straightforward and she was like, "Okay, this is what we do, this is what you can do." And there are no commitments. We signed month to month. And what they told us was, "You know what? We want to earn your business so we don't sign you for three years or anything. If you're not happy you can just move. That's fine."

Which is good. I think they all do that because it's very hard to move anyway. So they don't need to make you sign for however many years. And they were open to negotiation as well. So we had some specific use cases where, since we are acquiring a lot of data, we need to create a ton of resources within FHIR and their pricing is based on the number of resources. But we were able to negotiate on that.

And then very quickly we got an implementation specialist, Rahul, to help us. We had them in our Slack and we had the Medplum team on Slack where we could ask a question and support very quickly. So in terms of working with the team, I mean, for me it was a big plus. The attitude they have is great.

We've met a couple of bugs or things we couldn't do and every time we just submit the bug directly on GitHub and they take it, they fix it, they release it, and then we're good. And since everything is on GitHub, it's very easy for us to follow as well. So we know they're working on the bug, we know they fixed it, and we know on which release we are going to get the fix.

**What do you like most about the product if you had to pick one or two things?**



I think their overall architecture technology approach works for us. So the technology they choose, open source, the kind of architecture in terms of extending FHIR makes sense. We understand what they do and so it makes us use the product. It's easy for us to use the product because you take a step back and say, "Ah, if I had to implement it, that choice makes sense." So that's great.

I would also say the experience we had with the team. Transparency in the sales process, easy access to knowledgeable people. There's no tier one, tier two, tier three support that we have to go through. That was really good in comparison with the other EHRs we worked with in the past, where getting to the people that know what they're doing is hard and takes several weeks or months sometimes. This was refreshing.

**Is there anything that you dislike or anything that you would want to change?**

Yeah, I think with implementing Medplum, you have to realize it's not a mature product. We've met bugs, unwanted behaviors. It's not a battle-hardened, 10-year-old product. But, overall it works. I can't complain. And it's fast, so that's good as well. Because being fast has solved a couple of problems as well.

However, sometimes you send the wrong thing and you get a 500 internal error and not appropriate handling. The FHIR spec is not implemented entirely, specifically the search model. I mean the search in FHIR is very, very complex to implement in its entirety. It's not completely implemented in Medplum. There are still corner cases that they don't cover and it's sometimes not easy to know what is covered and what is not. So sometimes you're trying something and then you have to assess whether or not that is not supported or just a bug.

**How long did it take you to get to an MVP across the two implementations that your team did?**

So the first one we did took us about six months. We had a team of four engineers, six months. The second one we're doing is much, much faster. It's just two engineers and I think in four to five months we'll do it. Because we use our lessons learned as well, like the big lift from understanding FHIR data modeling in the first place, as well as being able to reuse parts of what we've done before by separating what is specifically implementation versus what is just plumbing on top of FHIR.

**Got it. That makes a ton of sense. So the main difference between those two projects was not the scope but rather the experience?**

Yep.

**What's the likelihood of you continuing to use and recommend the product in 18 to 24 months?**

If we have other implementations, it might make sense. I'm not going to push it if I don't think it makes sense, or if a traditional EHR is going to better serve the clients for sure. But if it fits the use case, absolutely. We've had a very positive experience overall.

**Is there anything that we haven't covered today that maybe would be worth mentioning? Anything you wish you had known while making the decision or doing the implementation?**

I think for me the overall message and what I want to point out to prospects from Medplum is that it's a good experience, but it's a bit of a change in that they bring a development tooling ethos to healthcare tech.

So if you are expecting a traditional healthcare tech experience, that's not going to work for you. Or you need to be open at least to a change - that's for sure. And also it's not a mature product. You have to accept that this, it's not out-of-the-box. There's missing stuff here and there, so you have to go along with that as well.

I personally believe that the trade-off is worth it. Because I'm on the side of let's move fast. For me, implementation velocity is important, so we'd rather have bugs here and there and have them fixed fast. In comparison with having to wait a month and a half to get access to an API that is under-documented and doesn't work the way I was expecting it to work. For me, that's a catastrophe. I'm losing a ton of time.

But you have to go along with it, it's a risk. It's a small company, so are they going to be around in five years? We don't know. The way I tend to mitigate that risk though is FHIR. So we tend to stick as much as we can with the standard FHIR API. We hardly use their components or stuff, we stick as much as possible to FHIR, as if we're talking to a FHIR server. So the strategy is that if everything goes badly, we should be able to migrate in a reasonable timeframe.