

[Home](#)

Exported Sep 09, 2020 at 12:58 PM

HL7 FHIR eReferral Workstream

Messages

Campfire

To-dos

Docs & Files

Check-ins

Schedule

Forwards

Added by Tim Berezny • Updated Aug 05, 2018 at 3:20 PM

/MessageHeader

60% - Pilot Ready

Links

- [/MessageHeader - Canadian eReferral Profile](#)
- [/MessageHeader - FHIR 3.2 Standard](#)

Relevant chat.fhir.org threads

- none

ValueSets

- **.event:** NewServiceRequest | UpdateServiceRequest | NewAppointment | UpdateAppointment | UpdateStatus|RequestForInformation | ResponseToRequestInformation|SendMessage Need to finalize

Notes

- Use of the MessageHeader serves two primary purposes:
 - 1) Inform intermediary relay points (like the Ontario HIAL) of the final destination for the payload (with **destination.endpoint**)
 - 2) Inform the destination system of the purpose of the payload, which can inform how the payload is processed (with **event.eventCoding**)

Questions

- Q: Valueset for .event requires discussion. The initial pilot use case will be NewServiceRequest.
- Q: Could MessageHeader.reason be used to describe referral, selfreferral, transfer of care, etc... (which was previously proposed for ServiceRequest.code, but it doesn't really fit there)

Important Decisions & Rationale

- none

Comments

Use the comments on this page to discuss the development of this resource profile. Conclusions made from the discussion will be added into this document, and the profile where applicable.

Comments & Events

**Alfred Wong**, VP of Engineering, Think Research

After talking to Tim, we thought it might be a good idea to add an identifiers attribute to the MessageHeader resource. That is, MessageHeader.identifiers with a 0..* relationship.

We feel this will be generic enough to help in the following ways:

- For relay type of services like Ontario's provincial HIAL or ThinkResearch's HUBLY service will provide a more consistent method in extracting metadata around delivering the FHIR messages
- In a non-relay type of relationship, it will provide transportation metadata between two different FHIR services that each would like to understand.
- It is generic enough for everyone else to use if they feel they need to use such a mechanism.

Anyone's feedback will be welcome.

Aug 21, 2018 at 9:48 AM · Notified 1 person



Tim Berezny, CTO Caredove, Chair FHIR eReferral Specification Working Group

After pondering it a bit, I think we can accomplish the same thing by putting the HUBLY id into the `subscription.channel.endpoint`.

e.g.,

`Subscription.channel.endpoint = www.hubly.com/api/referral_id=12345`

This approach has a few of benefits:

- Don't have to extend anything
- HUBLY doesn't have to search in the body of the message

Aug 21, 2018 at 10:08 AM · Notified 1 person



Alfred Wong, VP of Engineering, Think Research

Hey Tim,

I don't think the Subscription resource concept totally works for multiple relay servers but we can definitely talk about that more later.

Here are some more of my thoughts around MessageHeader and a Pure FHIR based relay server.

[Use Case 1 - Multi-Relay System \(2 or more\)](#)

Patient A sees their healthcare provider and Provider A provides a referral to see Specialist B. Provider A's system belongs to Relay System A, while Specialist B's system belongs Relay System B. Provider System A sends an electronic referral to Relay System A which then forwards it on to Relay System B where it finally reaches the destination of Provider System B. Once Specialist B accepts the referral the acknowledgement is then transmitted to from Provide System B to Relay System B which then forwards it back to Relay System A which then delivers the acknowledgment to Provider System A.



Use Case 2 - One Relay With Multiple Provider Systems

Patient A sees their healthcare provider and Provider A provides a referral to see Specialist B. Provider A's system belongs to Relay System A and Specialist B's system belongs to Relay System A. At the same time, Provider A decides that Patient A also needs to join an Aqua Fit class at their local community centre and decides to send an electronic referral there (Provider System C). Both Provider System B and Provider System C receive the electronic referral and send an acknowledgement back to Provider A's system through the single relay system.



Use Case 3 - Point-to-Point or One Relay with only Two Systems

Patient A sees their healthcare provider and Provider A provides a referral to see Specialist B. Provider A's system belongs to Relay System A and Specialist B's system belongs Relay System A. Provider System A send an electronic referral to Relay System A which then forwards it to Provider System B. Once Specialist B accepts the referral the acknowledgement if then transmitted from Provider System B to Relay System A which then forwards it Provider System A of the acknowledgement.



Relay Concepts

When designing a pure FHIR relay server, the Message Flow defined by the HL7 organization provides the framework to support an efficient relay FHIR system. The main goals we should be thinking about when creating a pure FHIR relay server are:

- High efficiency throughput of the messages through the relay servers
- Reduce complexity required to route a message
- Provide a secure and private mechanism to relay the messages but also a useful audit log on the flow of the messages.

The goal of a relay system is to provide the ability from Provider System A to send a message to Provider System B and vice versa without each provider system knowing that there are different

destinations or multiple relay servers (hops) inbetween the different provider systems. Some of the problems a relay system needs to concern itself with are:

1. Given a referral, how do you route it to the right destination?
2. Given a referral, how do you route the acknowledgment back to the original source without looking at the ServiceRequest resource?
3. Given a referral, how do you provide state like functionality using a stateless medium like HTTP and Message flow to provide accountability and reliability?

Non-Relay or 1-Relay with Only Two Systems

For the non-relay (Point-to-Point) or just one relay system between two systems, the relay system can use all of the attributes and flows related to a simple point to point integration as it the system knows that:

- It only needs to route it to one destination
- It knows the acknowledgement goes back to the other system
- It can create simple audit trails and security measures only for the two systems

Attributes

- **identifiers (0..*) [new]** - Following the same guidelines about identifiers (i.e. if it exists do not remove it) it can provide the ability for relay systems to attach stateful context to messages that are being passed between each system inside the relay system's network of end points. It also solves the following other problems:
 - Reduced complexity to route a message - Can use the stored identifier to route messages without looking at the MessageHeader.source.endpoint or MessageHeader.destination.endpoint if it cached that information in the router. If the endpoints are not available it removes the need for the relay to search through the Bundle.entries to determine the endpoint to hit ($O(n)$ operation vs $O(1)$ - Bundler.entries[0].identifiers).
 - Scales with N-relay systems - Using the previous point, having to store stateful information with the bundle will allow N-relay servers to work with each other well
 - If it is a point-to-point or 1 relays system this is not needed, hence the 0 cardinality.

If anybody has any thoughts or issues with these ideas, I'm happy to discuss.

Sep 17, 2018 at 1:37 AM · Notified 92 people



Tim Berezny, CTO Caredove, Chair FHIR eReferral Specification Working Group

The main alternative to "How does a sending system insert its own identifiers into API submitted eReferral, and have that returned with subsequent messages" thus far as been submitting the identifier in the initial ServiceRequest submission, and then returning that identifier in all subsequent MessageHeaders (which would require an extension).

Another option has just been [brought to my attention](#), submit the identifier in the MessageHeader.focus.identifier element, and then persist THAT with all subsequent messages into the same field.

After consideration, i believe this to be the preferred option.

Dec 11, 2018 at 4:48 PM · Notified 92 people