

CDS Hooks Proposals

Overview

This document covers multiple proposals to the CDS Hooks specification. It was jointly developed by Stanson Health and Athenahealth and is based on a real-world implementation of CDS Hooks. It covers the following areas:

1. Decisions / Suggestions: utilization of FHIR Resources to identify the specific object the Service is building the suggestion around
2. CDS Request Context: top-level attribution for Requesting Entity
3. Source: allow the Service to cleanly differentiate between the guidance's primary source, secondary sources/evidence, and SMARTapp links
4. Card/Resource relationship: create a relationship between each card and its related Resource(s)
5. Workflow vs Indicator: create a separate field that allows the Service to specify whether the user is allowed to proceed with their workflow if a suggestion is not taken
6. Suggestions: allow Service to suggest user actions that are not related to creation/deletion of Resources. Ex: dismissal of card.

Decisions / Suggestions Proposal

Currently, CDS Hooks [Response](#) allows CDS Services to communicate a decision taken by the user back to the EMR via **Decision** objects. Furthermore, CDS Service can suggest actions to the EMR via Suggestion objects.

Both Decisions and Suggestions can contain an array of strings to create and delete resources.

Use Cases

CDS Service may need to communicate to the EMR any of the following decisions:

- Create resource (ex. a new order)
- Remove resource (ex. Proposed order)
- Update resource (ex. Update dosage on a medication order)

In this case of removing resources, an id may be sufficient for the EMR to know what to do. However, creating and especially updating resources requires more details.

Proposal

We propose to extend Decision / Suggestion objects as follows:

- **Create:** array of FHIR resources representing objects that the EMR need to create. This could be handled as a FHIR bundle.
- **Update:** array of FHIR resources representing objects that the EMR needs to update. This could be handled as a FHIR bundle.

```
{
  "decisions": {
    "create": {
      "resourceType": "Bundle",
      "resource": {
        "entry": [
          {
            "resource": {
              "resourceType": "Medication"
              //...
            }
          }
        ]
      }
    },
    "delete": {
      "resourceType": "Bundle",
```

```
    "resource": {
      "entry": [
        {
          "resource": {
            "resourceType": "Order"
            //...
          }
        }
      ]
    }
  },
  "update": {
    "resourceType": "Bundle",
    "resource": {
      "entry": [
        {
          "resource": {
            "resourceType": "DiagnosticOrder"
            //...
          }
        }
      ]
    }
  }
}
```

Organization Context Proposal

Currently, CDS Hooks defines the context for a new CDS request using a structure like this one:

```
{
  "hookInstance" : "d1577c69-dfbe-44ad-ba6d-3e05e953b2ea",
  "fhirServer" : "http://hooks.smarthealthit.org:9080",
  "hook" : "patient-view",
  "redirect" : "http://hooks2.smarthealthit.org/service-done.html",
  "user" : "Practitioner/example",
  "context" : [],
  "patient" : "1288992",
  "prefetch" : {... }
}
```

For some CDS Services, it's necessary to know which organization (or client) is requesting the service. For example, it may be that a given client purchased only a subset of CDS Service's capabilities and CDS Service needs this information to properly handle the request.

There are currently two ways of potentially handling this: **'fhirServer'** Url and **'context'** object:

- **FhirServer** Url can include an identifier that indicates which organization (or hospital or practice, etc) is requesting the service. The primary issue with this approach is that the CDS service has to parse out the relevant identifier, thereby hardcoding assumptions about how the Url is structured. If those assumptions are ever violated, the process breaks down.
- **Context** object can reference the relevant FHIR resources, such as [Organization](#) to establish the context. However, the spec is currently rather ambiguous about the logistics of using context. For instance, whose responsibility is it to define what is to be included, the CDS Service or the EMR? What guarantees does a CDS Service have that the necessary information will be available?

We believe that the concept of **"Requesting Entity"** (or "organization") is likely necessary for any commercial CDS Service provider. Therefore, it should be elevated to a top level attribute, like patient.

Source Proposal

Currently, CDS Hooks **source** supports a single link. However, it is unclear what information should be a CDS Service should store in this field. One possibility to use the field to refer to CDS service that served the guidance. This is particularly useful because it allows a service to control how it is presented to end users. Another reasonable choice, is using the source field to store the primary source of the guidance provided on a card (ex. *Choosing Wisely*). Ideally the CDS hooks specification should support the ability for both of these concepts to be conveyed from a service to the EHR.

Additionally, it may be appropriate to provide references to secondary sources of guidance, which may be or may not be in the form of links. Currently, the CDS Hooks specification does not provide a means of conveying this information outside of including them within *card.details*. In our opinion storing secondary sources in it's own field rather than within details is preferred as it gives more control to the EHR over the presentation of secondary sources (ex. including this information in a modal link or a more/less view near the primary source).

Proposal

In order to address the needs described above we propose the following changes the CDS Hooks specification:

1. Update the CDS hooks documentation to specify that the ***cards.source*** field should be used to store the primary source of the guidance provided within the card.
2. Add a `card.secondarySources` field containing markdown of secondary supporting references:

```
secondarySources: "# Supporting Evidence ..."
```

3. Add a ***userFacingDescription*** field to the service attributes that is transmitted as part of service discovery:

```
userFacingDescription: {  
  url: 'http://some-url-to-service',  
  label: 'service label' // limit to 40 characters (or  
    some other small limit)  
}
```

Supporting multiple sources avoids utilization of **links** to provide multiple sources. This allows the EHR to differentiate between SMARTApp links and referential links.

Card/Resource Relationship Proposal

Currently, the CDS-hooks specification does not provide a mechanism associating the CDS card to the order(s), or other resources, that it is in response to. For example, if a CDS information card came back from a CDS service indicating that a prescription dosage was too high given the patient's age, it would be beneficial for the EHR to know the particular order the information card was referring to so that it might apply special formatting on the order (or other actions) when the user interacts with the card.

Proposal

We propose to extend the CDS Card specification as follows

- Include a `relatedResources` field to identify the resource(s) that the card is in response to.

```
{
  relatedResources: [
    {
      resourceType: "Diagnostic Order",
      identifier: "1234" // the order id
    },
    ...
  ],
  ...,
}
```

Each related resource has the following fields:

- **resourceType**: String, the type of resource the card is tied to.
- **identifier**: String, an ID of the resource.

Workflow separate from Indicator Proposal

Currently the CDS-Hooks specification uses the field 'indicator' in a card to describe two different concepts: the urgency/importance of the card, and how the workflow within the EMR should process the card. While these two concepts are somewhat similar and may appear to overlap, they are distinct. The urgency/importance indicates to the EMR how to display the card to the end-user. The workflow processing indicates if the user can dismiss/ignore the card, must explicitly accept a suggestion or provide a reason for not accepting a suggestion, or has no choice but to accept a suggestion.

Proposal

- Modify Card.indicator to remove 'hard-stop' as it explicitly is indicating a workflow processing approach
- Add Card.workflowImpact required field with values of '**soft-stop**', '**medium-stop**', and '**hard-stop**':
 - Soft-stop - The user can continue the workflow without providing any reason or accepting a suggestion.
 - Medium-stop - The user must provide a reason for not accepting a suggestion before proceeding in the workflow.
 - Hard-stop - The user must accept a suggestion to proceed in the workflow.

Suggestions Proposal

Card Action Notifications

An EHR might allow users to take actions on CDS cards outside of clicking suggestion buttons or links. For example, the EHR could allow users to dismiss a card so that it no longer appears, ignoring any suggestions on the card. There is currently no mechanism in the CDS-hook spec that allows for the EHR to notify the CDS service that such an action has taken place.

Proposal

Extend the CDS-hooks specification to define a set of Card Actions so that the CDS service can inform the EHR that it should be notified when an action takes place via the CDS services's analytics endpoint. The list of actions will be communicated to the EHR via service discovery.

List of Card Actions:

- **dismissal**: Occurs when a card is dismissed from the EHR.
- Other actions?

The CDS Card specification would also be extended to include a UUID field uniquely identifying the card:

```
{  
  uuid: "e2577d69-afce-32aa-aa32-3f01a243a1ab"  
}
```

When an action takes place the EHR would notify the CDS service via the analytics endpoint and include the particular action it is report within the body of the POST:

```
"card-action": "dismissal"
```