DevDays 10th Edition

HL7® FHIR® DevDays

NOV | 2020

Exchanging FHIR Data – Choosing the right approach

Lloyd McKenzie – Gevity Consulting

HL7 FHIR DevDays 2020, Virtual Edition, November 17–20, 2020 | @FirelyTeam | #fhirdevdays | www.devdays.com/november-2020

ORGANIZED BY firely

Who am I?

- Name: Lloyd McKenzie
- Company: Gevity
- Background:
  - One of FHIR's 3 initial editors
  - Co-chair FMG & FHIR Infrastructure
  - Heavily involved in HL7 and healthcare exchange for last 20 years
    - v2, v3, CDA, etc.
  - Technical lead for Da Vinci Health Record Exchange (HRex)
  - lmckenzie@gevityinc.com

Learning Objectives

- What are the FHIR exchange options?
- What are the considerations in choosing between the options?

Creative Commons - Attribution

- Download, use as you wish, just thank HL7 (and maybe me)

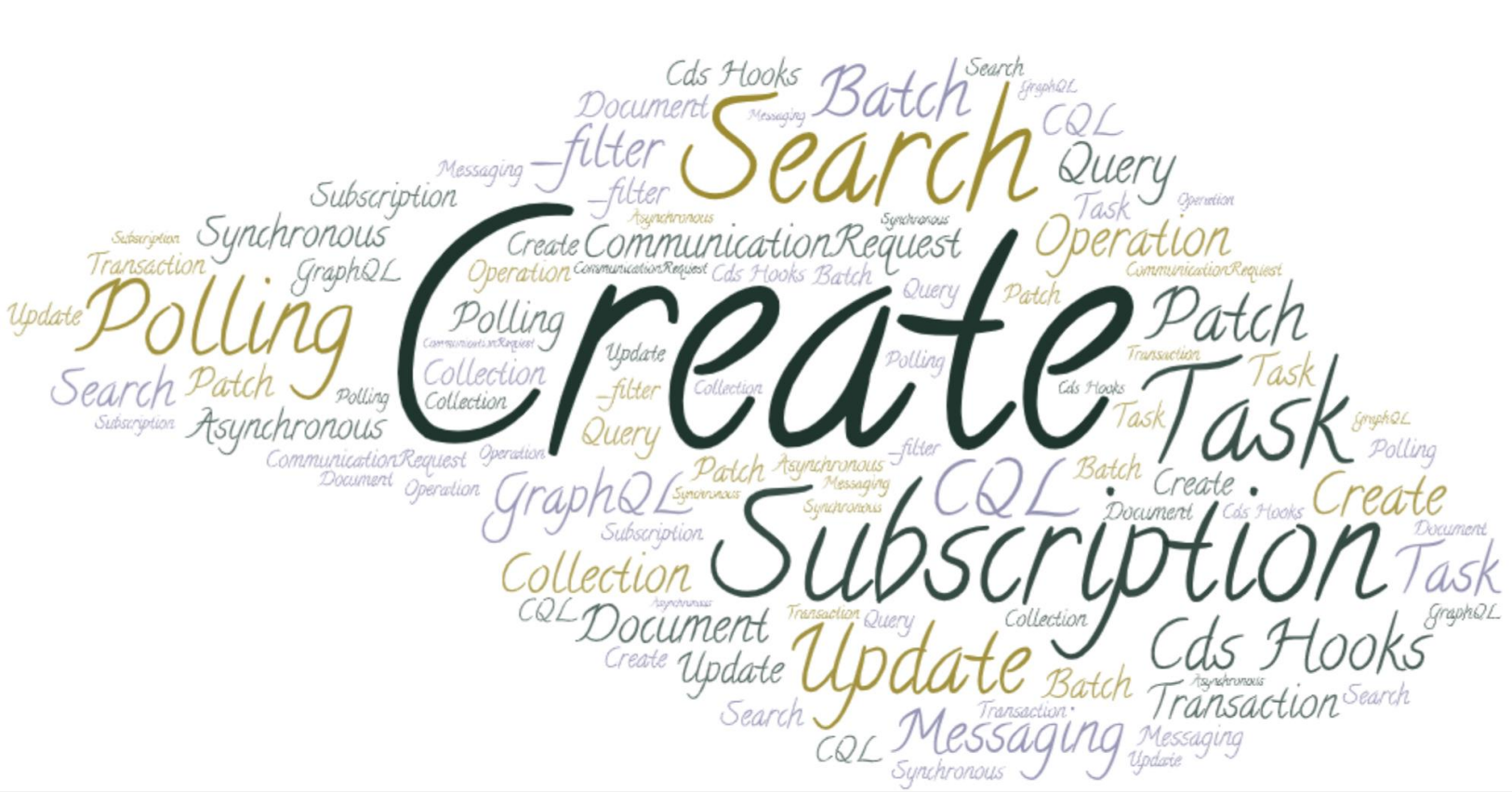
- You might also enjoy “Flicker or Bonfire” – my architecture presentation from last year

Context for this presentation

- 20+ Implementation Guides
- Different approaches
- Questions about why approach chosen
  - Needed a better answer than "because that's what we decided"
- Undertook a deep analysis of what the options were and what should be used when

FHIR defines a **LOT** of ways to share data

- 21 in total
  - Or more…



So – how to choose?

- How re-useable / multi-purpose?
- How widely adopted?

Approach	Re-use	Adoption	Description
RESTful search	High	High	The *data consumer* uses the FHIR search to describe the desired data and, using _include and _revinclude, the desired resources and the *data source* returns the requested information if available.
batch search	High	High	The *data consumer* sends a FHIR batch request to the *data source* containing multiple search, _filter, GraphQL, query and/or operation requests. All of the requests are executed and the responses are returned in a batch response.
Polling	High	High	The *data consumer* queries the *data source* at regular intervals checking to see if there is new data that matches a particular criteria.
Transaction Bundle	High	High	The *data source* creates a 'transaction' Bundle requesting the creation and/or updating of various resources and posts it to the RESTful endpoint of the *data consumer*.
•••			
FHIR Operation	Low	Moderate	The *data consumer* invokes a custom operation on the *data source* requesting information by parameters on the URL and/or in the body and the response to the operation (synchronous or asynchronous) contains the requested data.
FHIR Messaging	Low	Moderate	The *data consumer* sends a FHIR message to the *data source* requesting information and the data source responds (synchronously or asynchronously) with a message containing the requested data.
Query search	Low	Low	The *data consumer* invokes a custom query operation on the *data source* and receives back a search Bundle

Re-useable/multi-purpose

- REST Encounter update
  - Revise info as part of pre-admit message
  - Capture “patient arrival”
  - Transfer patient
  - Schedule for discharge
  - Correct “reason for admission”
  - Capture end time of patient phone-call
  - Etc.
- V2 ADT A01 message
  - Patient has been admitted

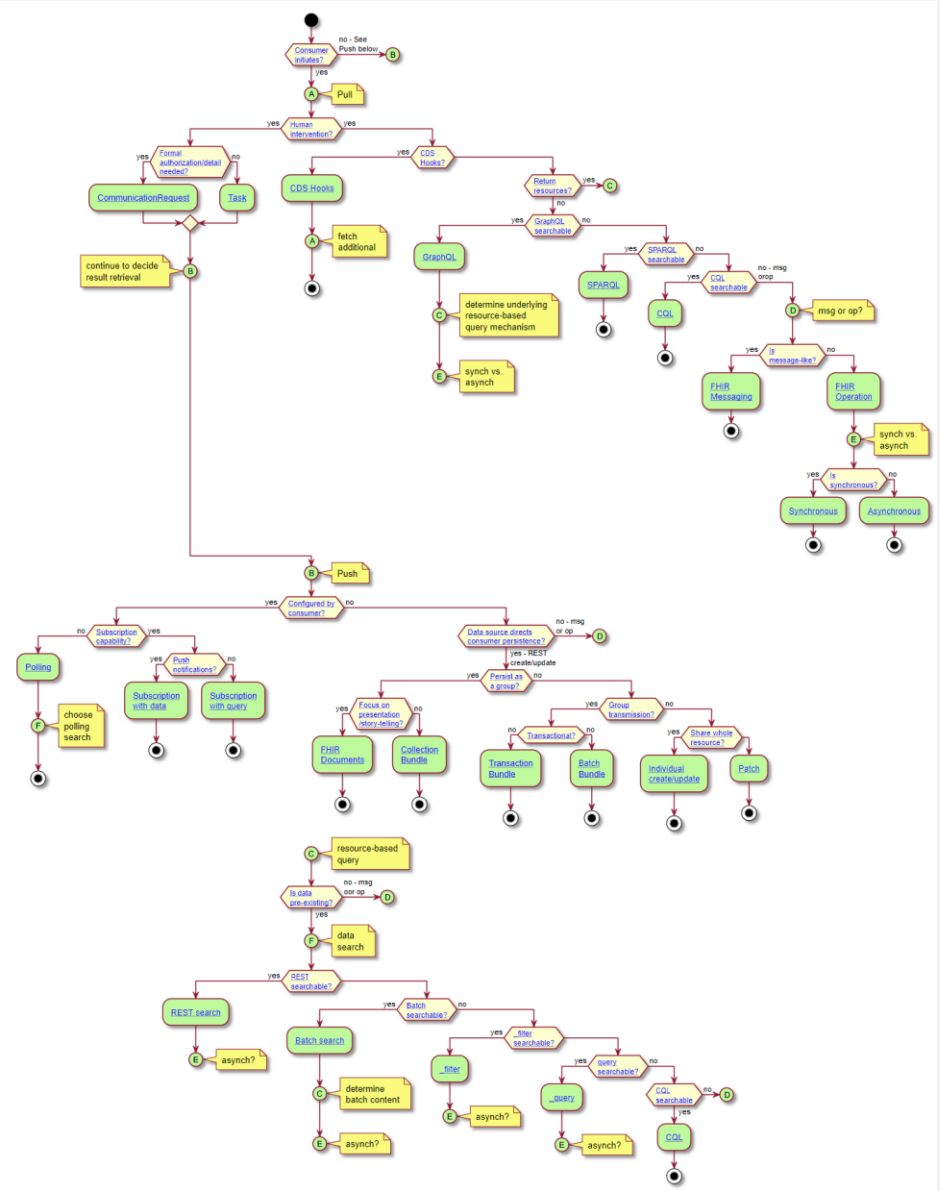
How widely adopted?

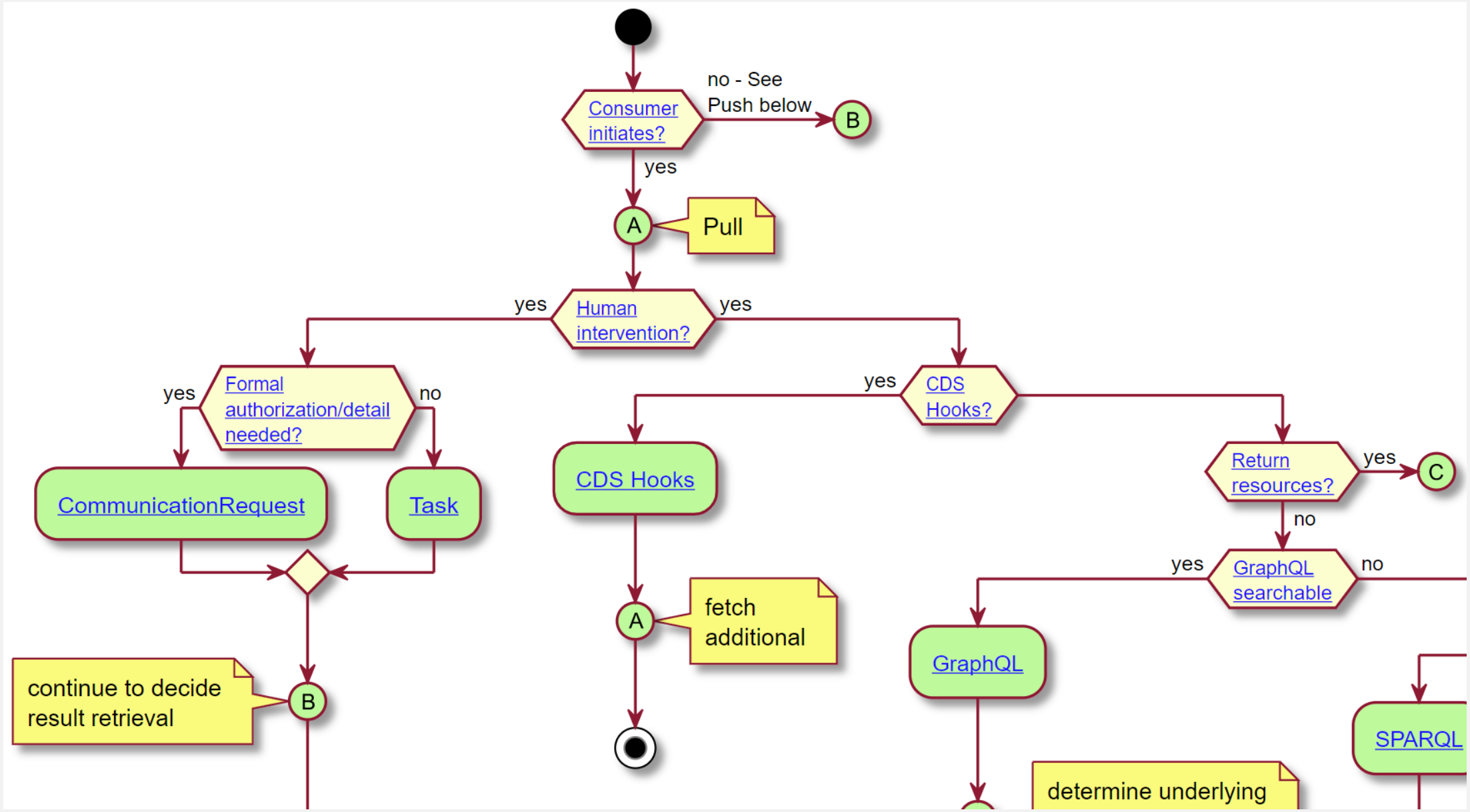
- Often equivalent to “how easy to adopt?”
  - Do it all or do part?
  - Ease of layering over legacy
  - E.g. REST search vs. _filter

How to choose

- What's your use-case?

- https://build.fhir.org/ig/HL7/davinci-ehrx/exchanging.html




Consumer initiates?
no - See Push below
B
yes
A
Pull
yes
Human intervention?
yes
yes
Formal authorization/detail needed?
no
CommunicationRequest
Task
continue to decide result retrieval
B
yes
CDS Hooks?
CDS Hooks
A
fetch additional
Return resources?
yes
C
no
yes
GraphQL searchable
no
GraphQL
SPARQL
determine underlying

Some caveats

- Informative, not normative
- Other factors will matter
  - Legacy capabilities
  - Existing infrastructure
  - Internal architecture conventions
  - Personal preference
  - What do your partners support?

What are the options?

- Pull resources
  - Read, search, batch search, vread, history, _filter, _query, CQL, Task, CommunicationRequest, operations, messaging
- Pull data elements
  - CDS Hooks, GraphQL, SPARQL, CQL, operations, messaging
- Consumer-directed push
  - Polling, subscription – notification, subscription w/ data
- Source-directed push
  - REST create/update, batch, transaction, patch, document, collection, Operation, Message

Key decision points

- Pull vs. push
- Are humans involved?
- Are you retrieving resources or data elements?
- Is push ‘consumer-configured’?
- Does source direct persistence?
- Does data already exist?
- Store as a collection or individually?
- Synchronous or asynchronous?

Where is this work going?

- Reviewed and discussed by various work groups
- Found in Sept 2020 ballot release of Davinci HRex IG
  - Publication target Q1 2021
  - All Da Vinci IGs will reference
  - Will also explain tree navigation rationale
  - Encourage other IGs to do the same
- Will migrate into FHIR R5 when it comes out
- Work in progress will be here:
  - https://build.fhir.org/ig/HL7/davinci-ehrx/exchanging.html

Summary

- Many options to share data
  - Some widely adopted (easier)
  - Some provide greater re-useability
  - Standard guidelines to help choose
  - Greater consistency in choices = more interoperability

Contact

- During DevDays, you can find / reach me here:
  - Via Whova App – Speaker's Gallery
  - Email: lmckenzie@gevityinc.com

- Or, better yet, include the community and ask/discuss on http://chat.fhir.org