HL7® FHIR® DevDays

JUNE | 2023

The FHIR Bulk Data API and what's new in v2!

Dan Gottlieb (@gotdan)

HL7 FHIR DevDays 2023 | Hybrid Edition, Amsterdam | June 6–9, 2023 | @HL7 | @FirelyTeam | #fhirdevdays | www.devdays.com

ORGANIZED BY firely | HL7 International

HL7®, FHIR® and the flame Design mark are the registered trademarks of Health Level Seven International and are used with permission.

slides: https://bit.ly/bulk-june-2023

Agenda

- Why a FHIR Bulk Data API?
- Technical Architecture & what's new in 2.0
- Open Source Tools
- Adoption!
- Next steps and how to get involved

FHIR APIs

Patient

FHIR REST API

Panel

FHIR Bulk Data API

Population

Use Cases

- Internal clinical data warehouse for study cohort identification
- Machine learning with training data from EHR
- Claims in EHR to provide comprehensive view
- Integration population health system with EHR system
- Transferring records from one EHR to another
- Payer database to assess care quality
- Reportable disease submission or other registry

Let's enhance FHIR® to support population level data access

- **FHIR Resources** as a standard data model to simplify data parsing and mapping
- **FHIR Operation API** to initiate the data extracts
- **SMART Backend Services Authorization** as security model

Focused Scope + Complementary Technologies

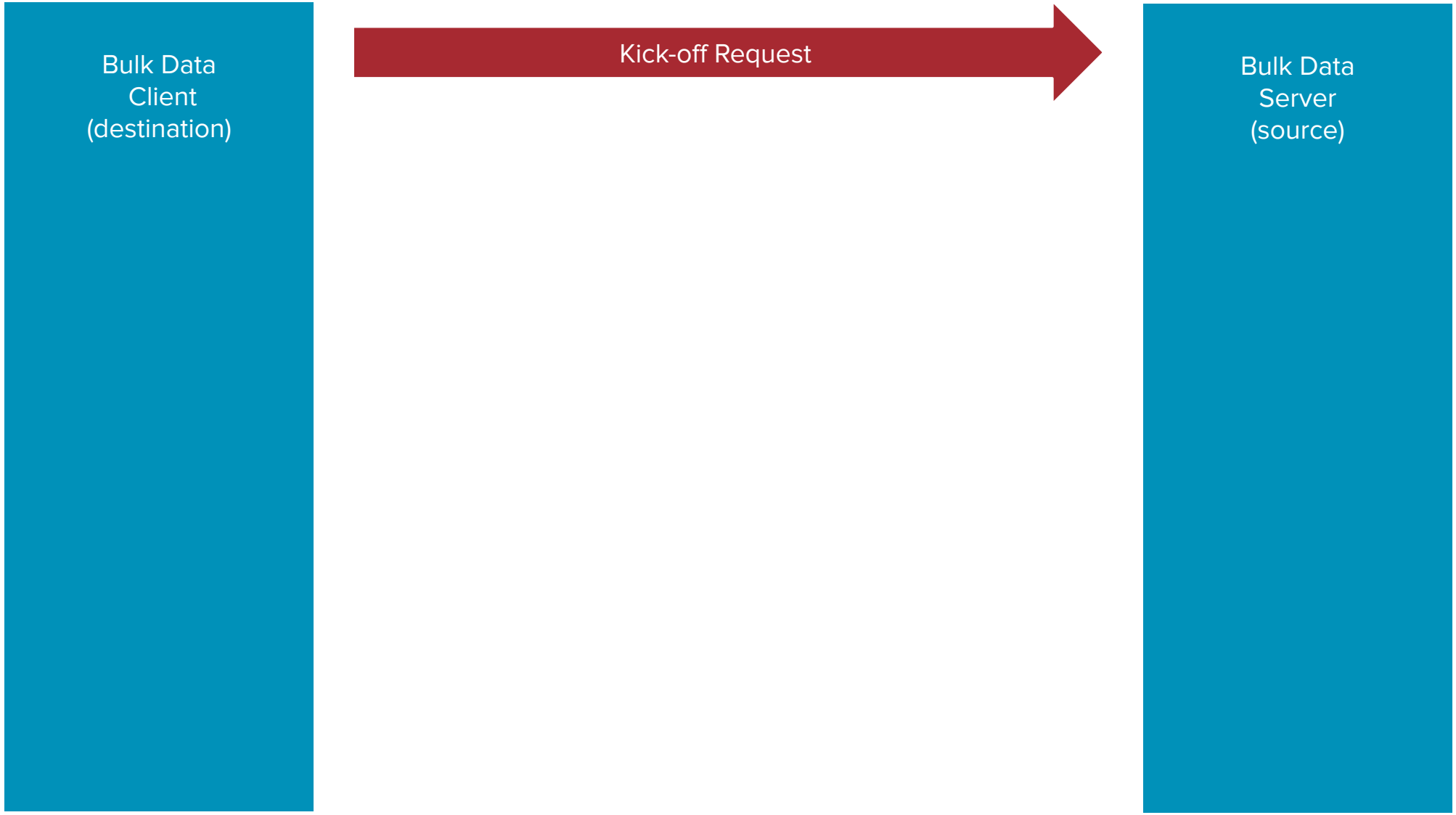
- Legal framework for sharing data between partners needs to be set up out-of-band (BAAs, SLAs, DUAs)
- Real-time data - data loaded through bulk APIs can be supplemented with real time FHIR REST API calls or subscriptions
- Patient matching - it’s possible to include identifiers like subscriber number in Bulk Export FHIR resources
- Data transformation - can serve as a foundation for data pipeline

Technical Architecture
Bulk Data Access Implementation Guide

Bulk Data Access IG Versions

STU1 (v1)	STU2 (v2)
Initial version	Incorporates experience from early implementations
Published August 2019	Published November 2021

Kick-off Request



Kick-off Request

- FHIR Operation for all data on all patients (all data in the patient “compartment”)
  `[FHIR Server Base]/Patient/$export`

- FHIR Operation for all data on a group of patients (eg. research cohort, plan members)
  `[FHIR Server Base]/Group/[group id]/$export`

- FHIR Operation for all data on the server
  `[FHIR Server Base]/$export`

- Asynchronous requests with status polling (HTTP GET only in v1, GET or POST in v2)
  `Prefer: respond-async`

Kick-off Operation Parameters

_outputFormat	The format for the generated bulk data files Currently, only **ndjson** is required
_since	Filter results by FHIR resource modified date FHIR instant timestamp *(required for servers to support)*
_type	Filter results by comma delimited list of FHIR resource types *(optional for servers to support)*
_typeFilter	Filter using FHIR REST queries *(optional and experimental)*

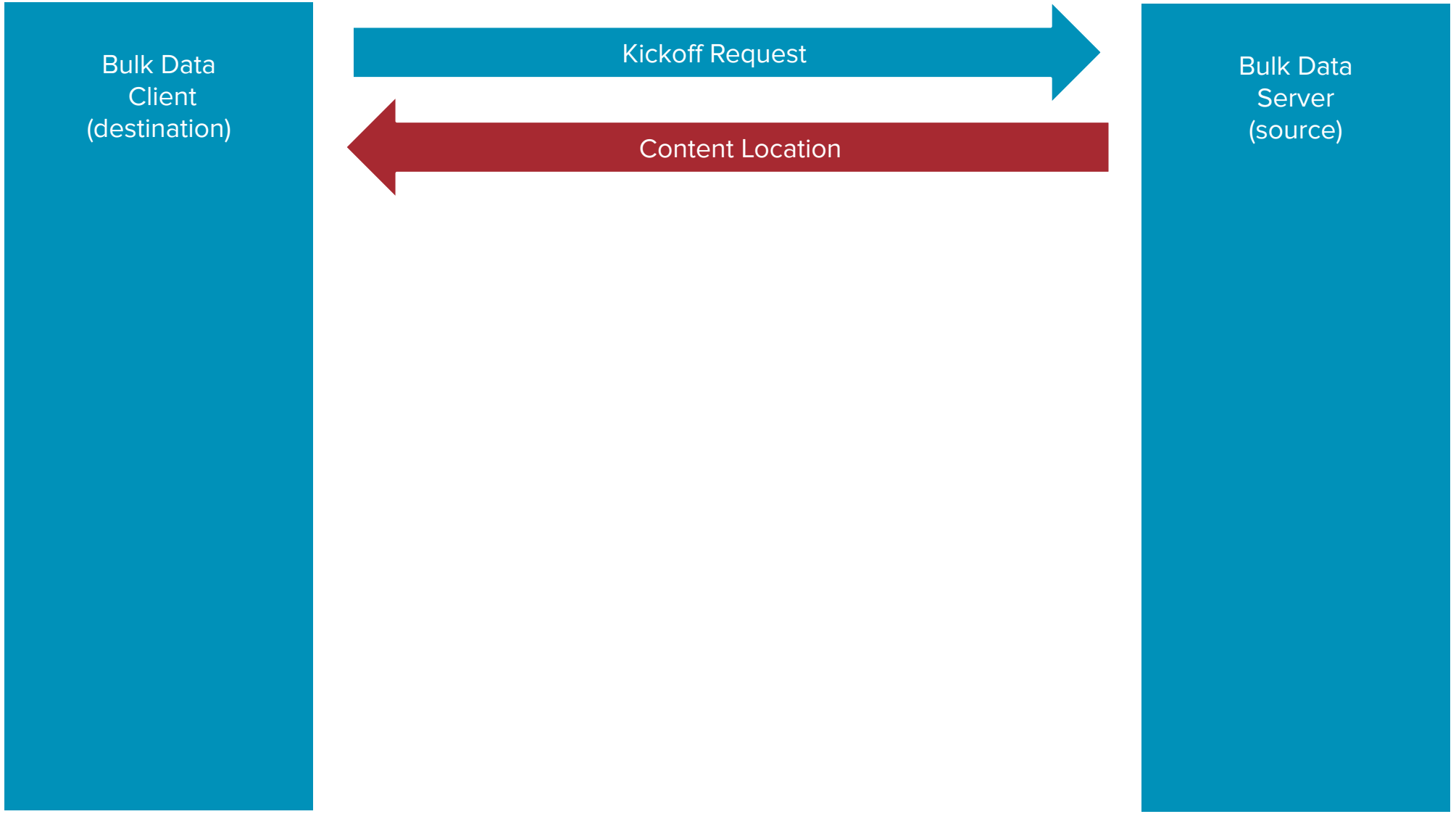
Updated in v2

Kick-off Operation Parameters

New in v2

_elements	FHIR resource elements to return e.g., `Patient.id, Patient.identifier` *(optional and experimental)*
Patient	FHIR Patient References to limit data returned *(optional, not valid for GET requests or system level requests)*
includeAssociatedData	Metadata resources to include with response e.g., `LatestProvenanceResources or RelevantProvenanceResources` *(optional and experimental)*

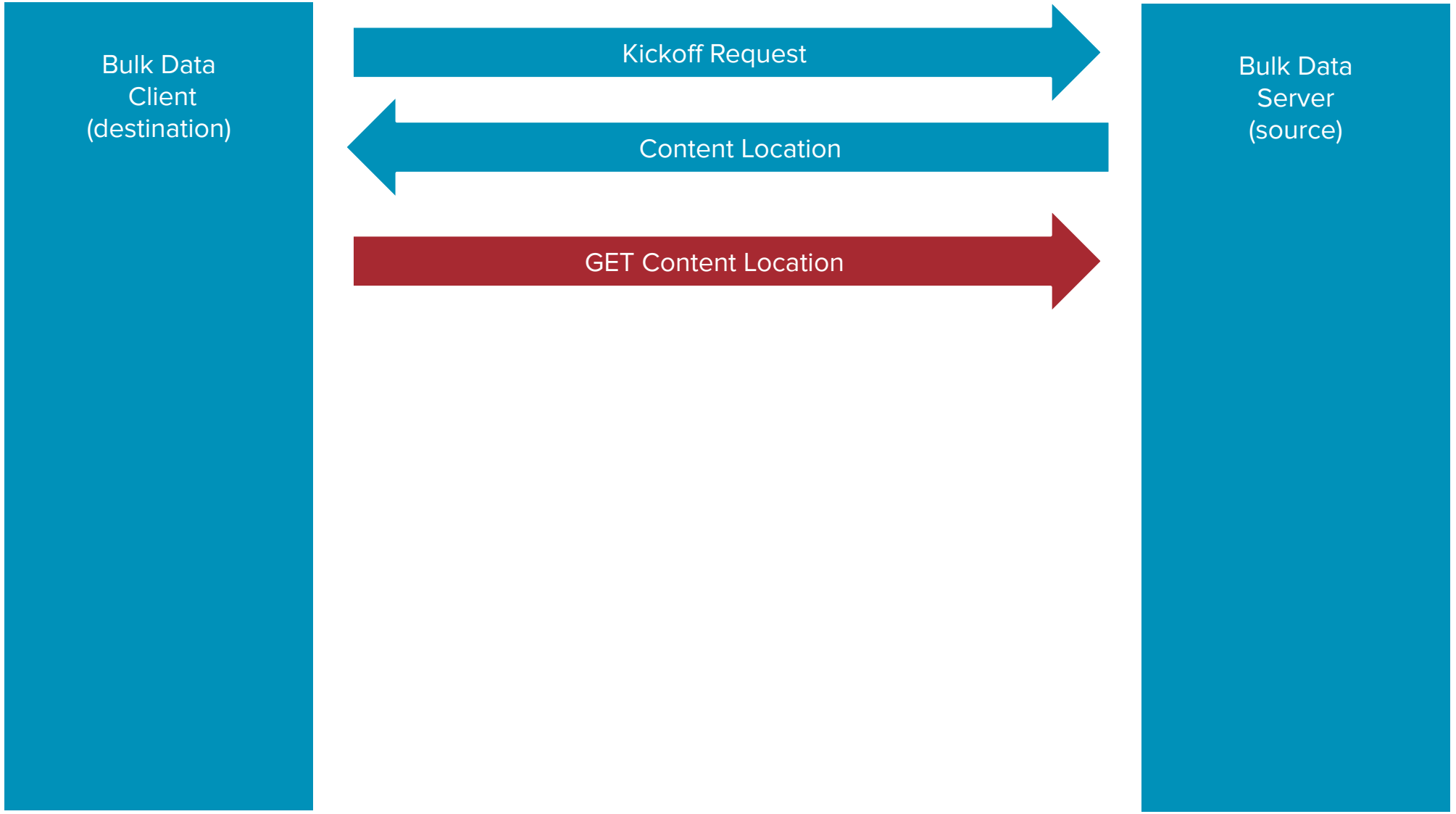
Kick-off Response



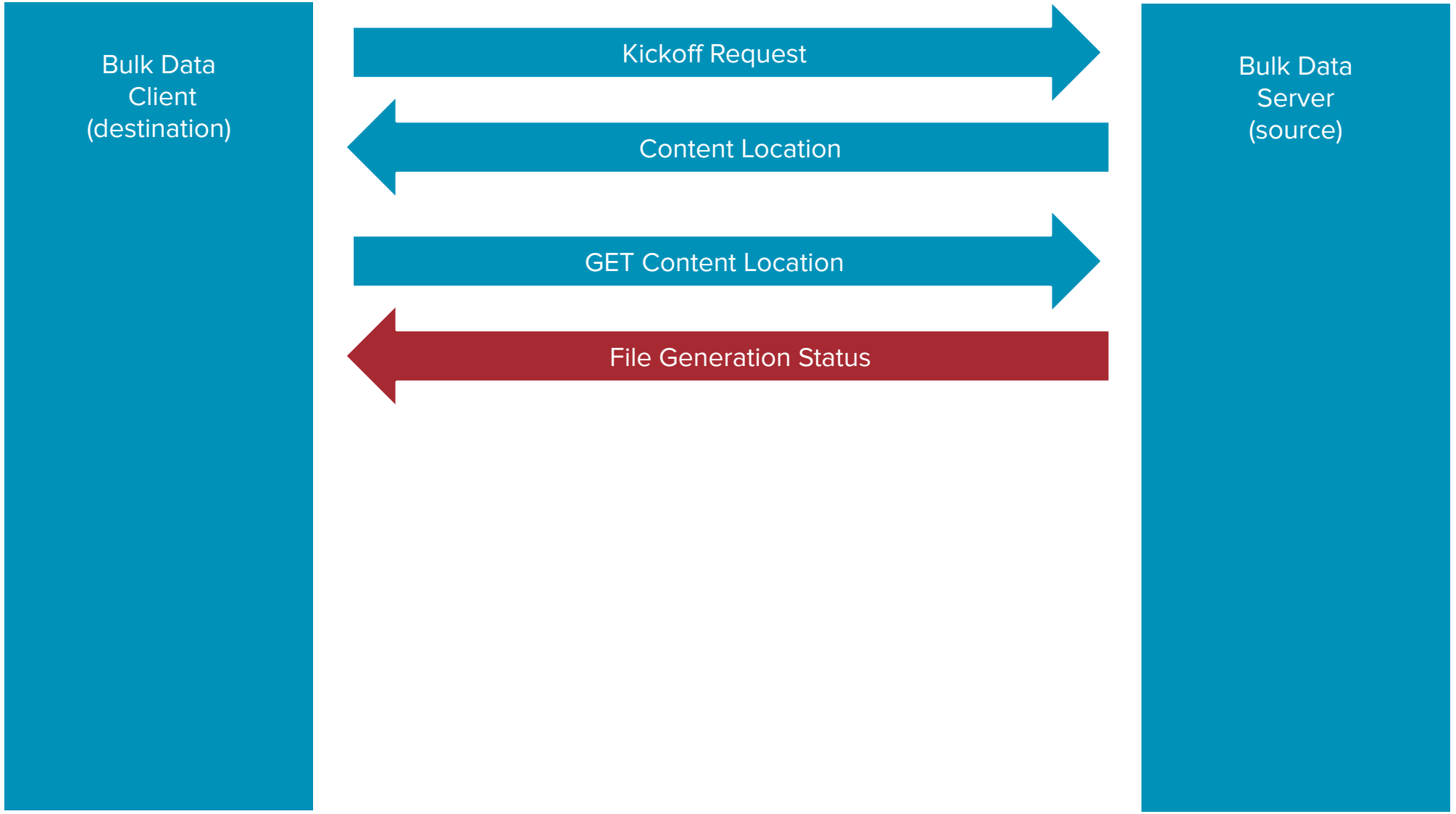
Kick-off Response Header

```
Status: 202 Accepted
Content-Location: [URL for status or deleting request]
```

Status Request #1



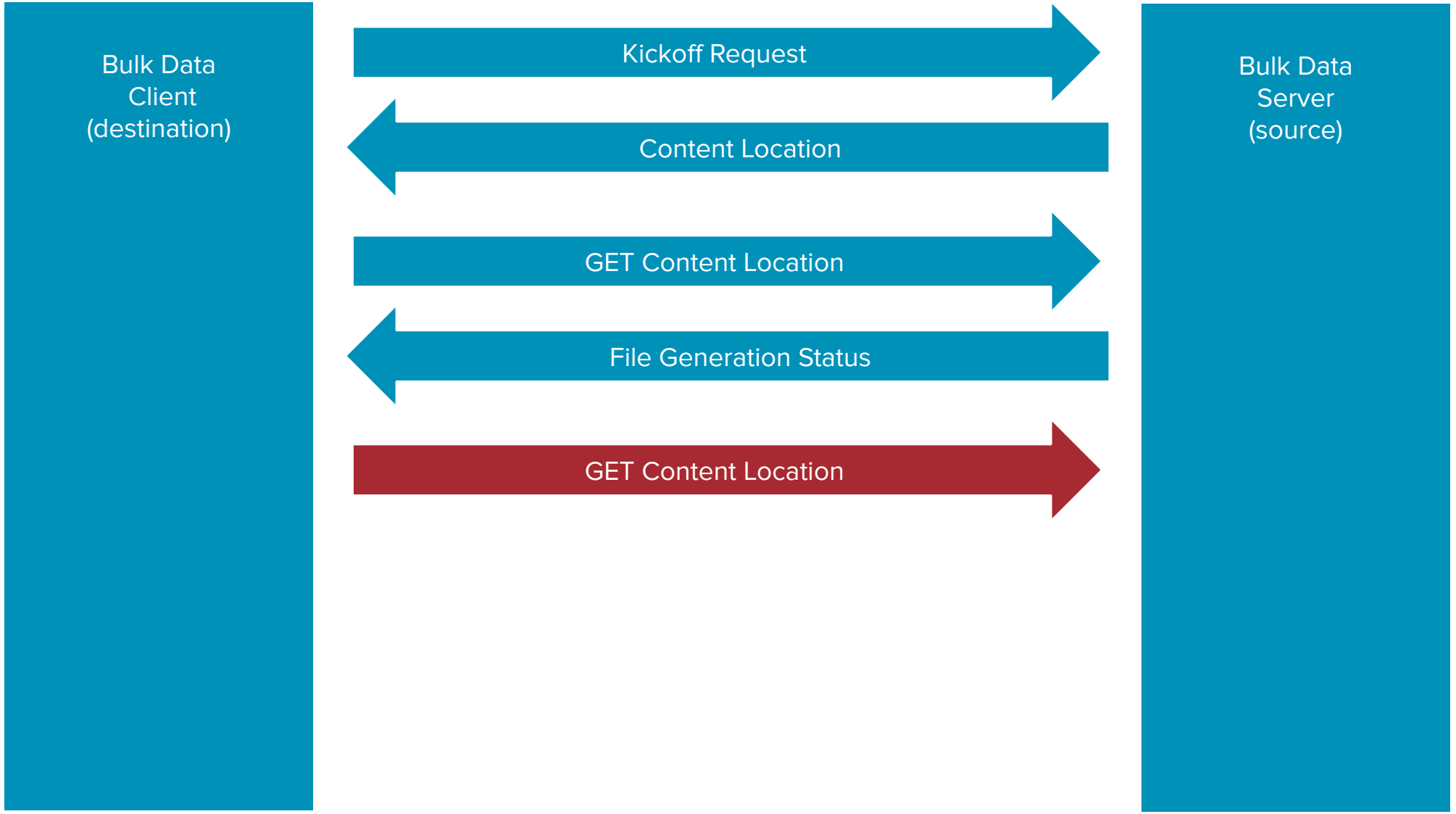
Status Response #1



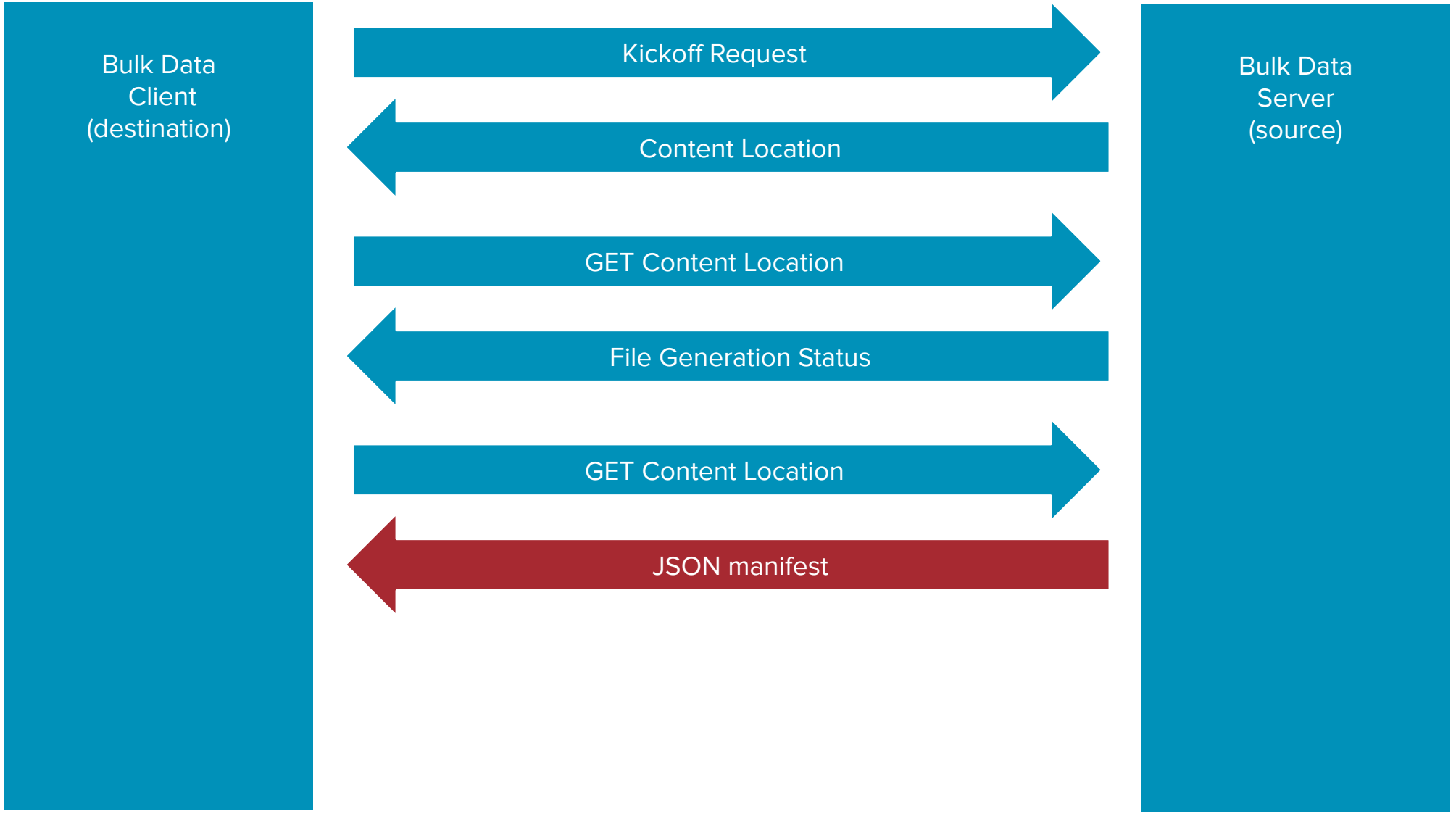
In-Progress Status Response Header

```
Status: 202 Accepted
X-Progress: “50% complete”
Retry-After: 120
```

Status Request #2



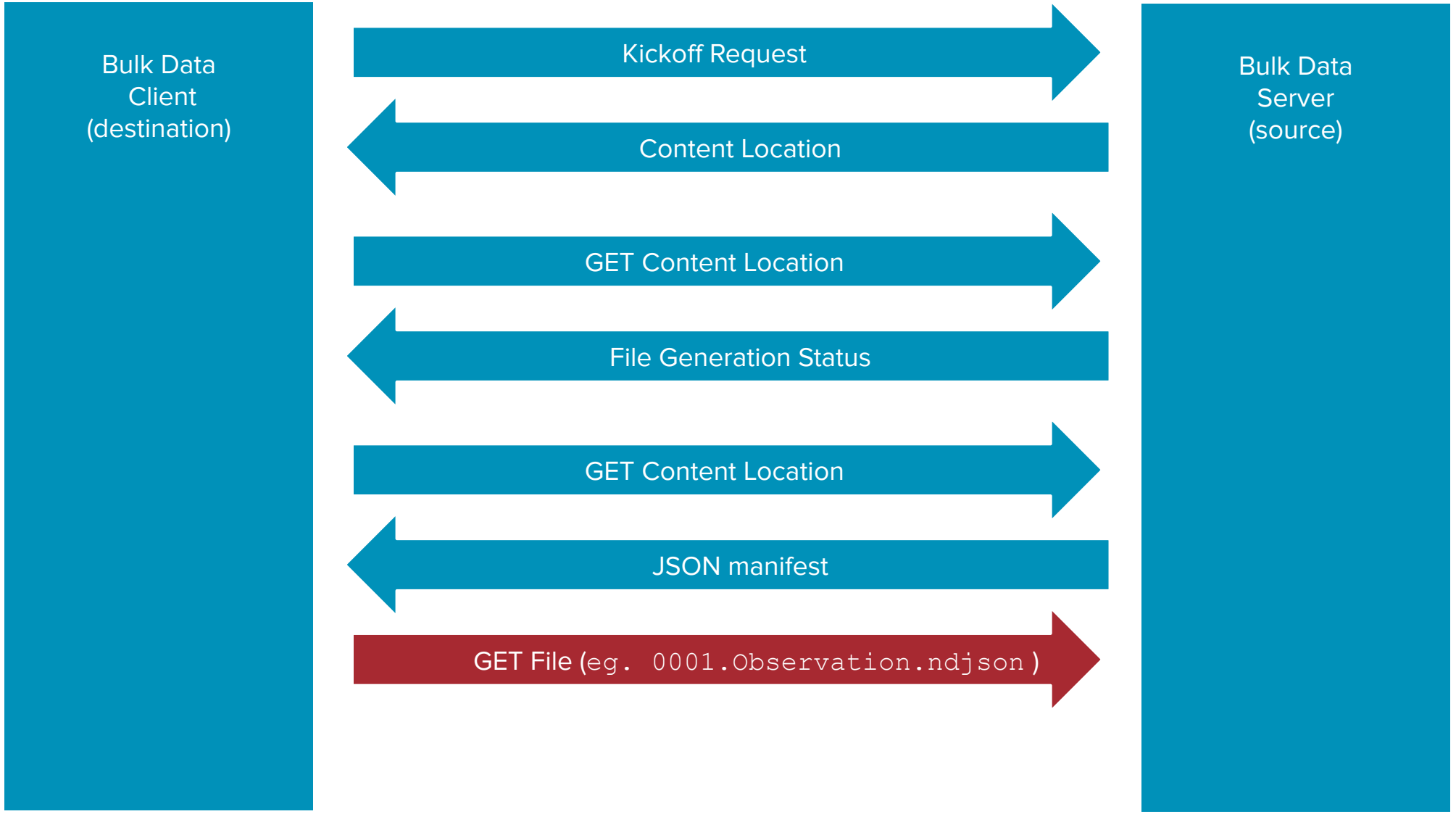
Status Response #2



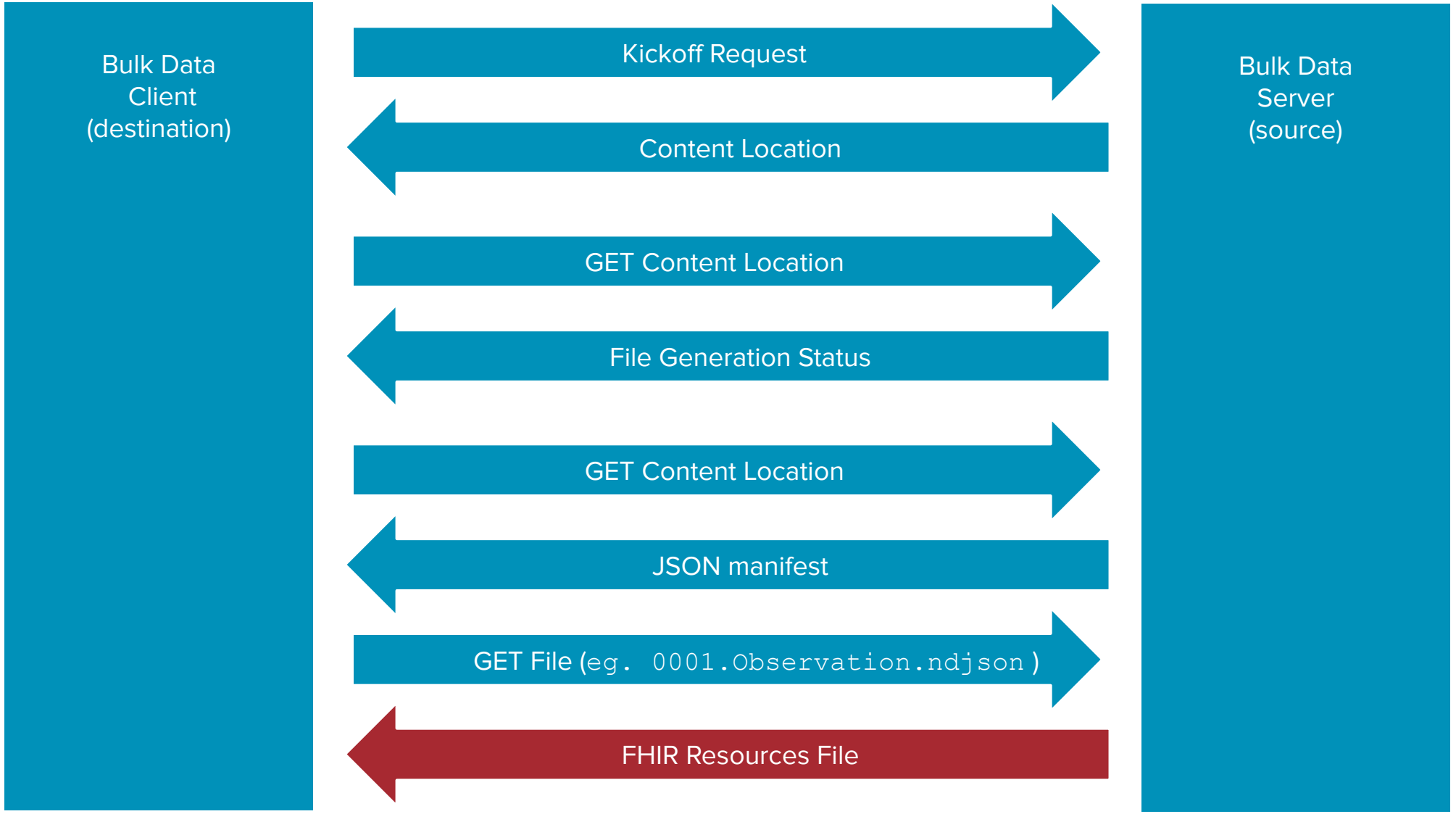
Status Complete Response Body

```
1  "transactionTime" : "2020-07-13T13:28:17.239Z",
2  "request" : "https://example.com/Patient/$export?_type=Patient,Observation",
3  "requiresAccessToken" : true,
4  "output" : [{
       "type" : "Patient",
       "url"  : "https://example.com/files/patient_file_1.ndjson"
   },{
       "type" : "Patient",
       "url" : "https://example.com/files/patient_file_2.ndjson"
   },{
       "type" : "Observation",
       "url" : "https://example.com/filesw/observation_file_1.ndjson"
   }],
5  "deleted" : [{
       "type" : "Bundle",
       "url" : "https://example.com/output/del_file_1.ndjson"
   }],
6  "error" : [{
       "type" : "OperationOutcome",
       "url"  : "https://example.com/files/error_file_1.ndjson"
   }]
```

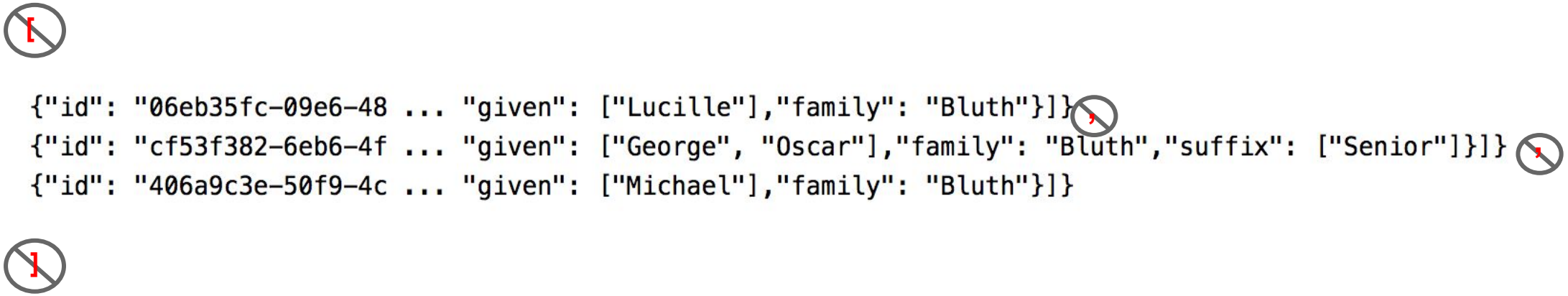
File Request



File Response



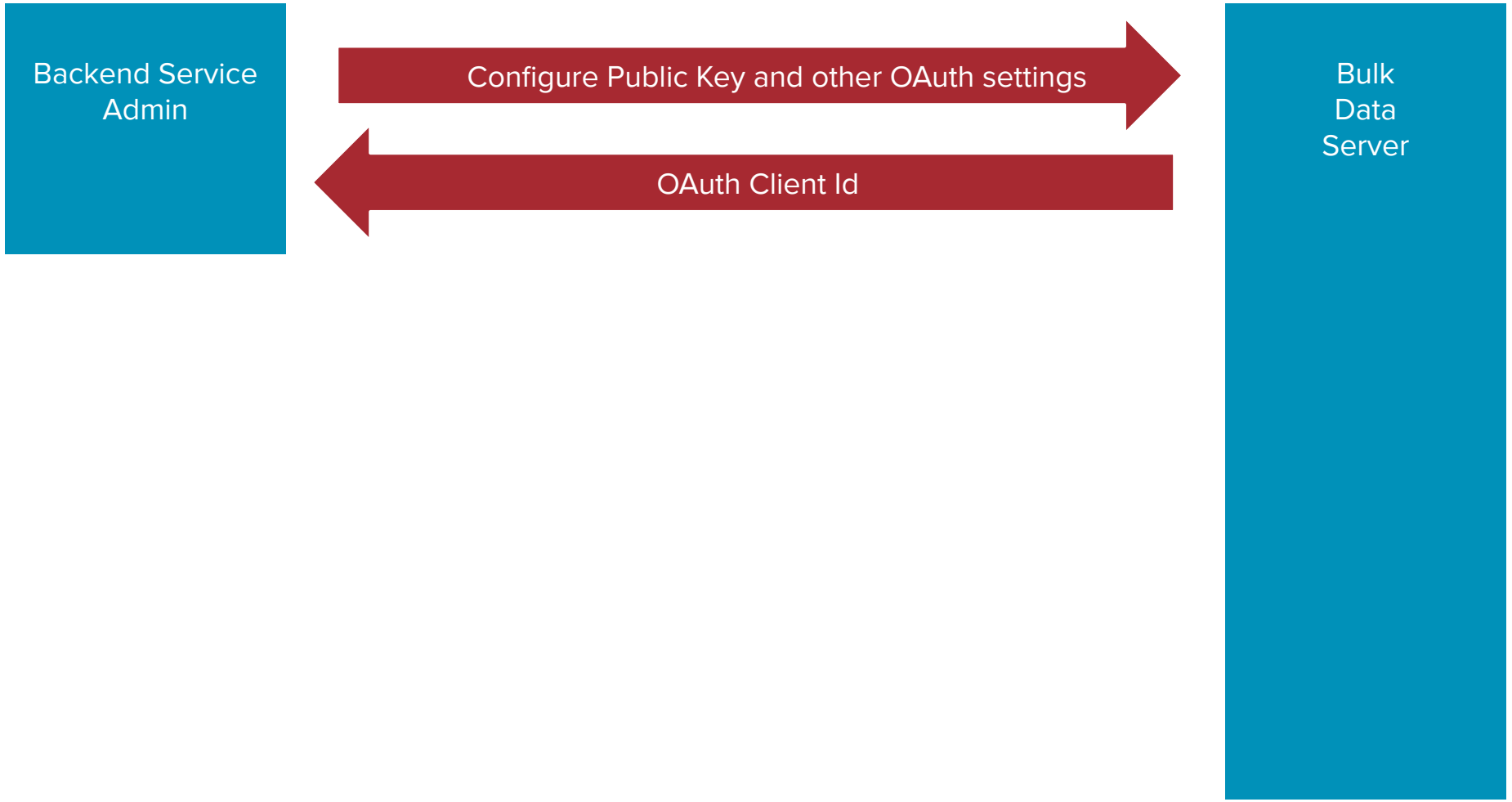
NDJSON


[
{"id": "06eb35fc-09e6-48 ... "given": ["Lucille"],"family": "Bluth"}]}
,
{"id": "cf53f382-6eb6-4f ... "given": ["George", "Oscar"],"family": "Bluth","suffix": ["Senior"]}]}
,
{"id": "406a9c3e-50f9-4c ... "given": ["Michael"],"family": "Bluth"}]}
]

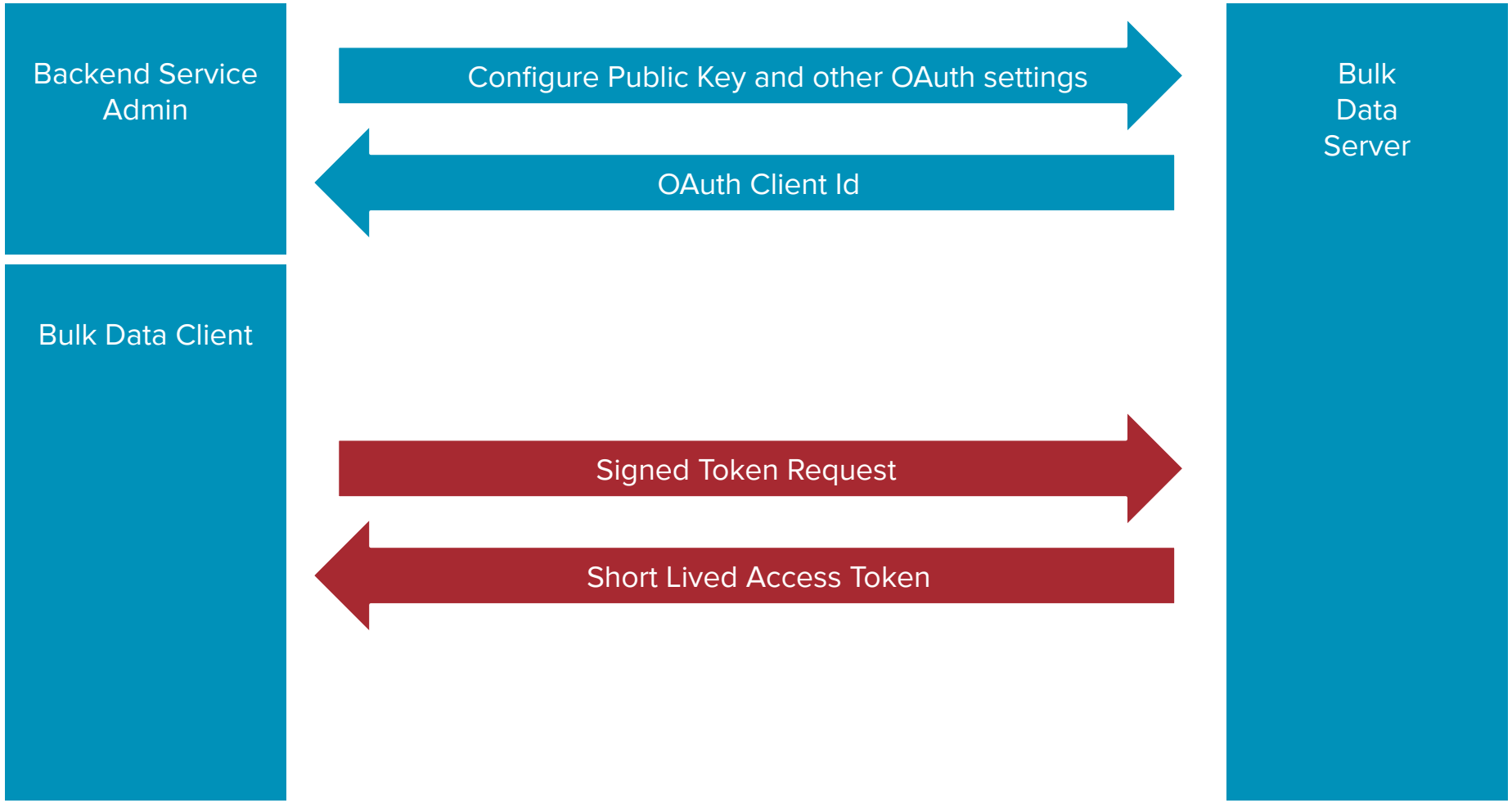
SMART Backend Services Authorization

- Out-of-band app registration (can use Dynamic Client Registration or portal)
- Apps can register public key (JWKS format) or URL for public key
- Token requests signed with private key
- Short-lived access tokens

Registration Flow (once)

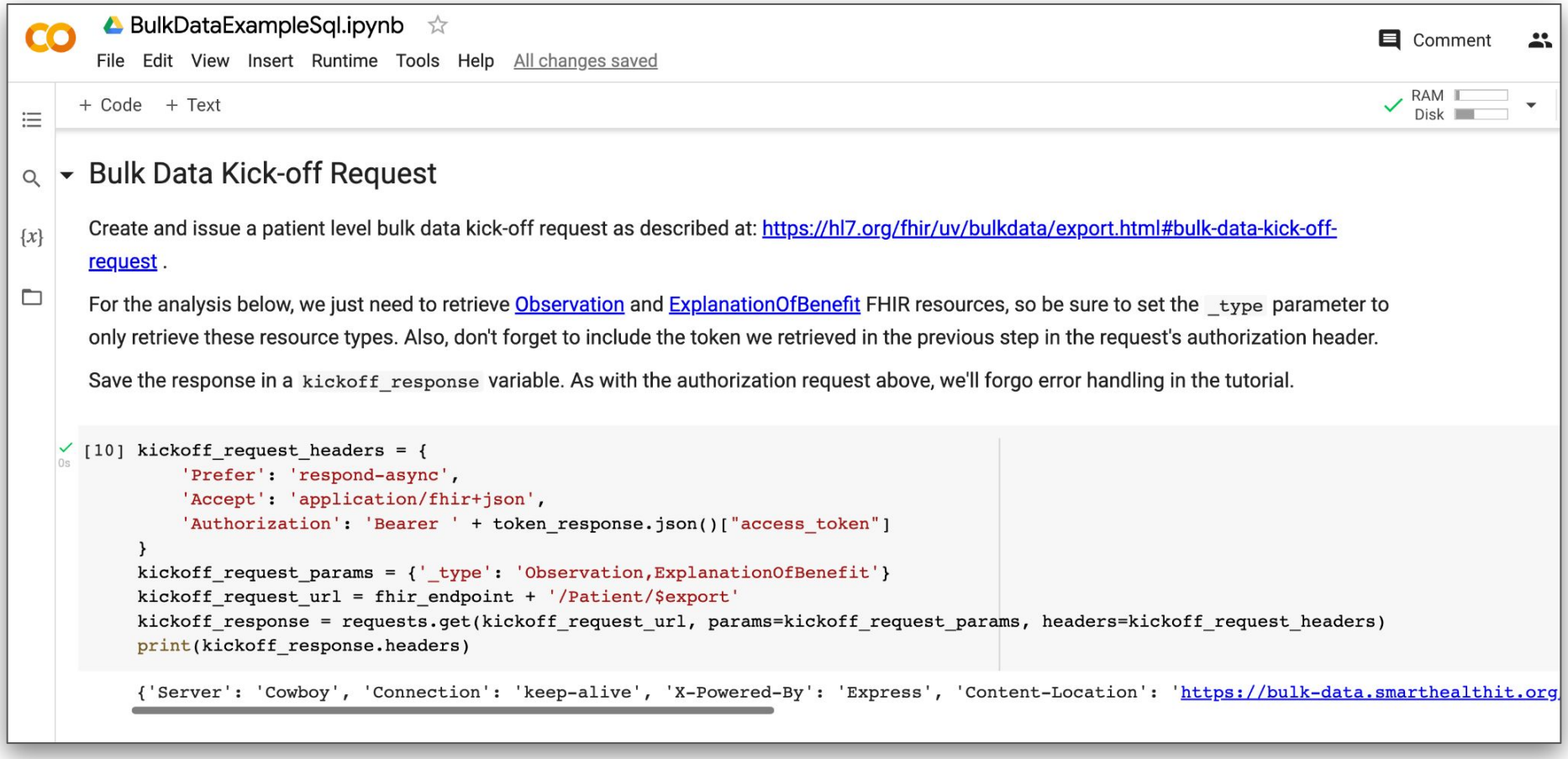


Authorization Flow (min. once per request)



Tutorial (Python): EHR Export to SQL Exploration

https://bit.ly/bulk-tutorial



Major Enhancements in v2

Historical Group Data

- Server side with revised guidance on the "**`_since`**" parameter

  "In the case of a Group level export, servers MAY return additional resources modified prior to the supplied time if the resources belong to the patient compartment of a patient added to the Group after the supplied time (this behavior should be clearly documented by the server)."

- Client side with the "**`_elements`**" and "**`patients`**" parameters
  - Make a request to get just the ids of patients in the group with "_elements"
  - Use the "patients" parameter to get data for patients not previously retrieved
  - Use the "patients" parameter and the "_since" parameter to get new data for remaining patients

Attachments

Resource elements of type Attachment must contain:

data element with a Base64 encoded version of the file

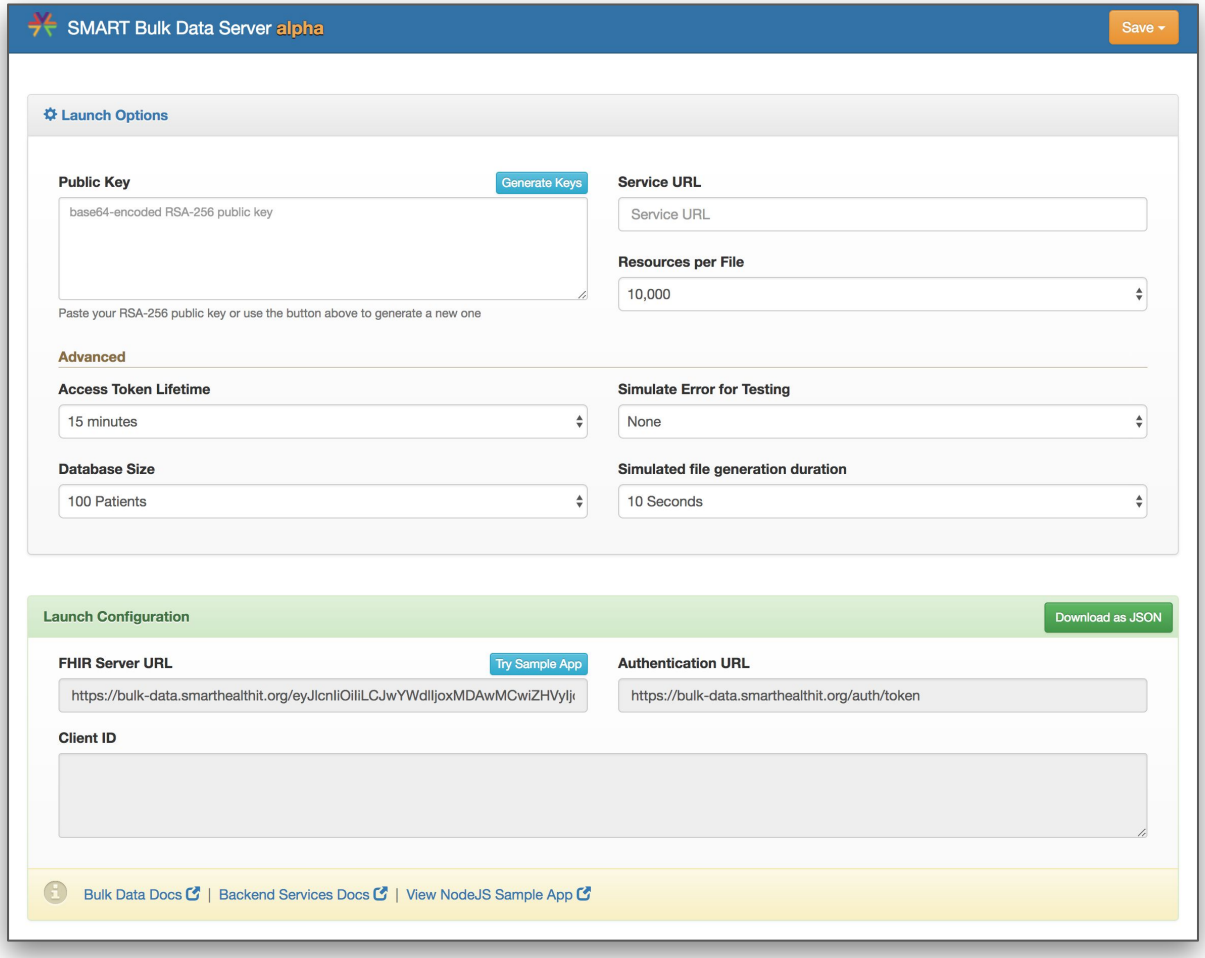
OR

url element with an absolute URL for the file accessible using the same authentication as the ndjson files

Open Source Tools

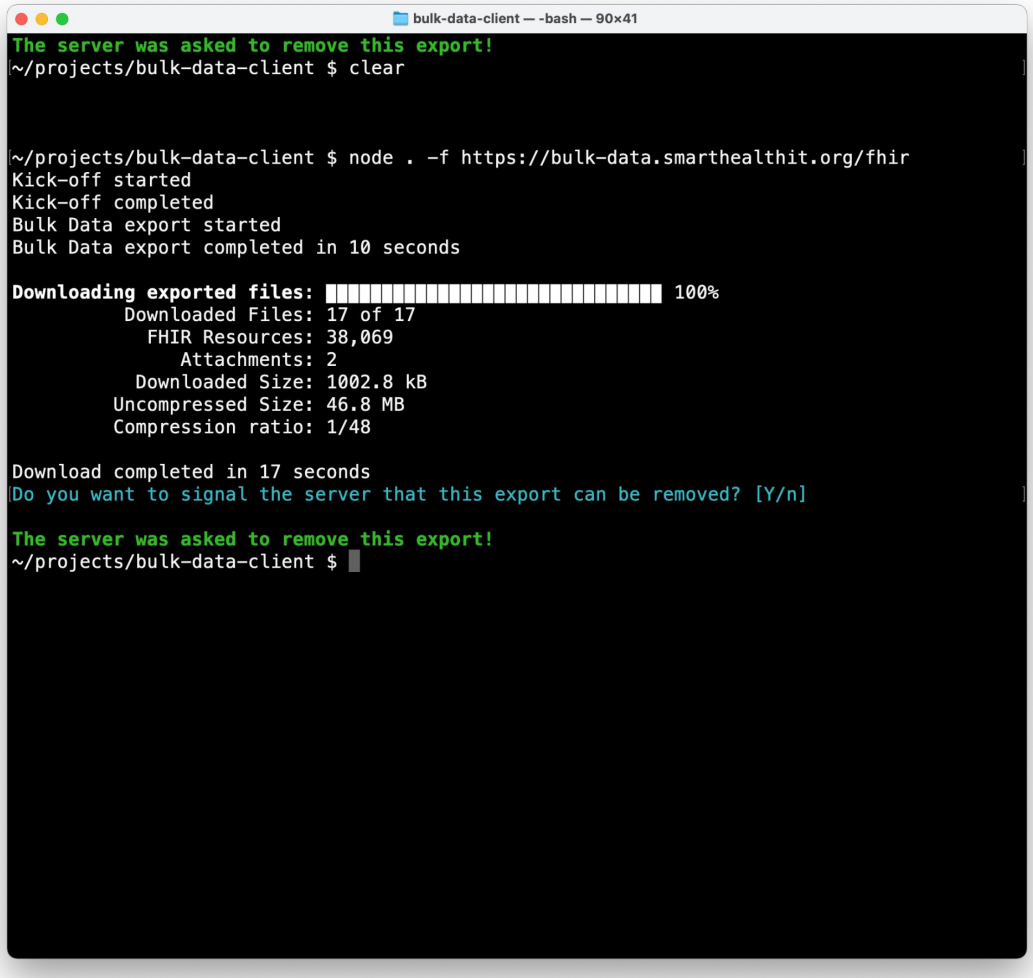
SMART Reference Server Implementation

https://bulk-data.smarthealthit.org



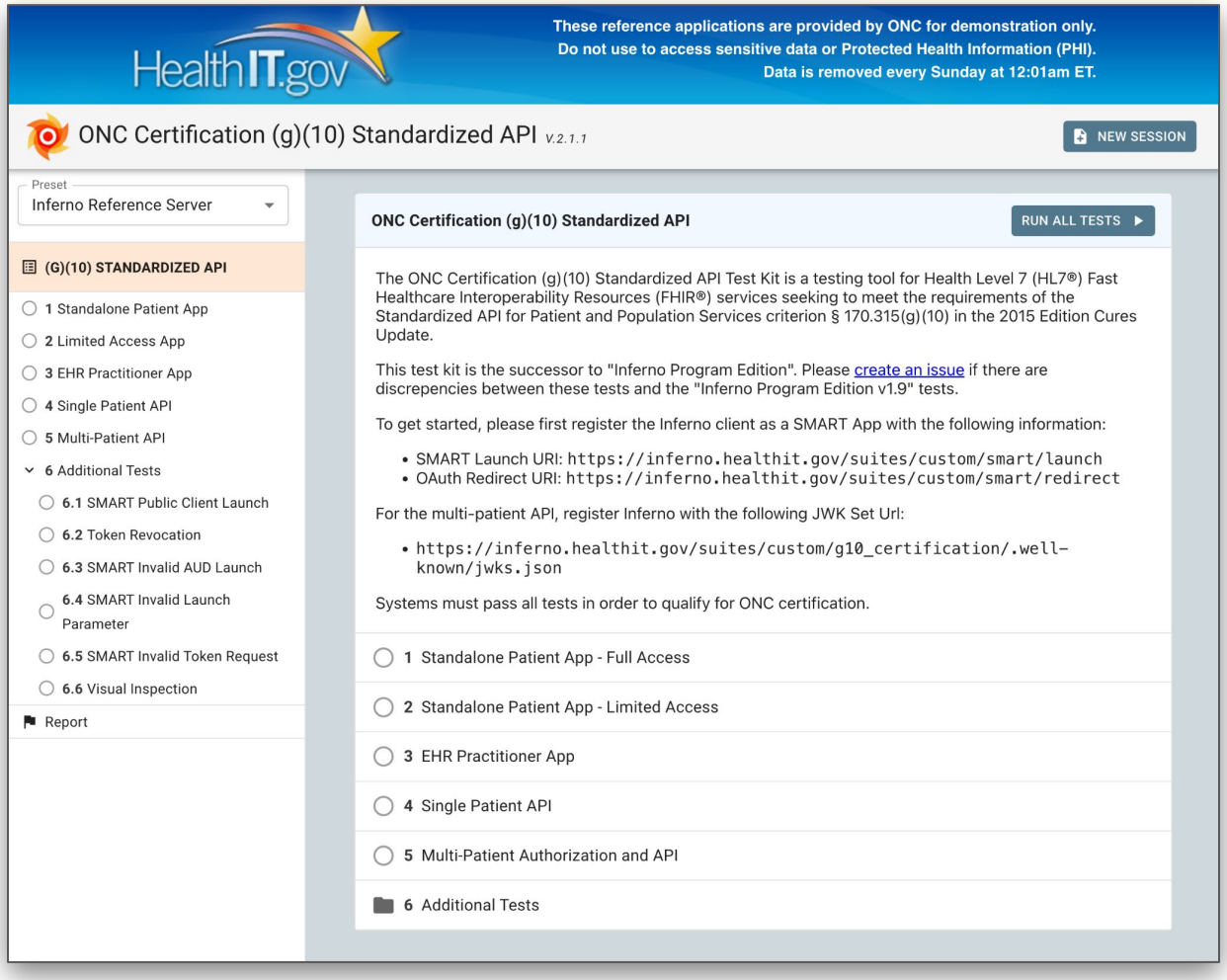
SMART Sample Client

https://github.com/smart-on-fhir/bulk-data-client



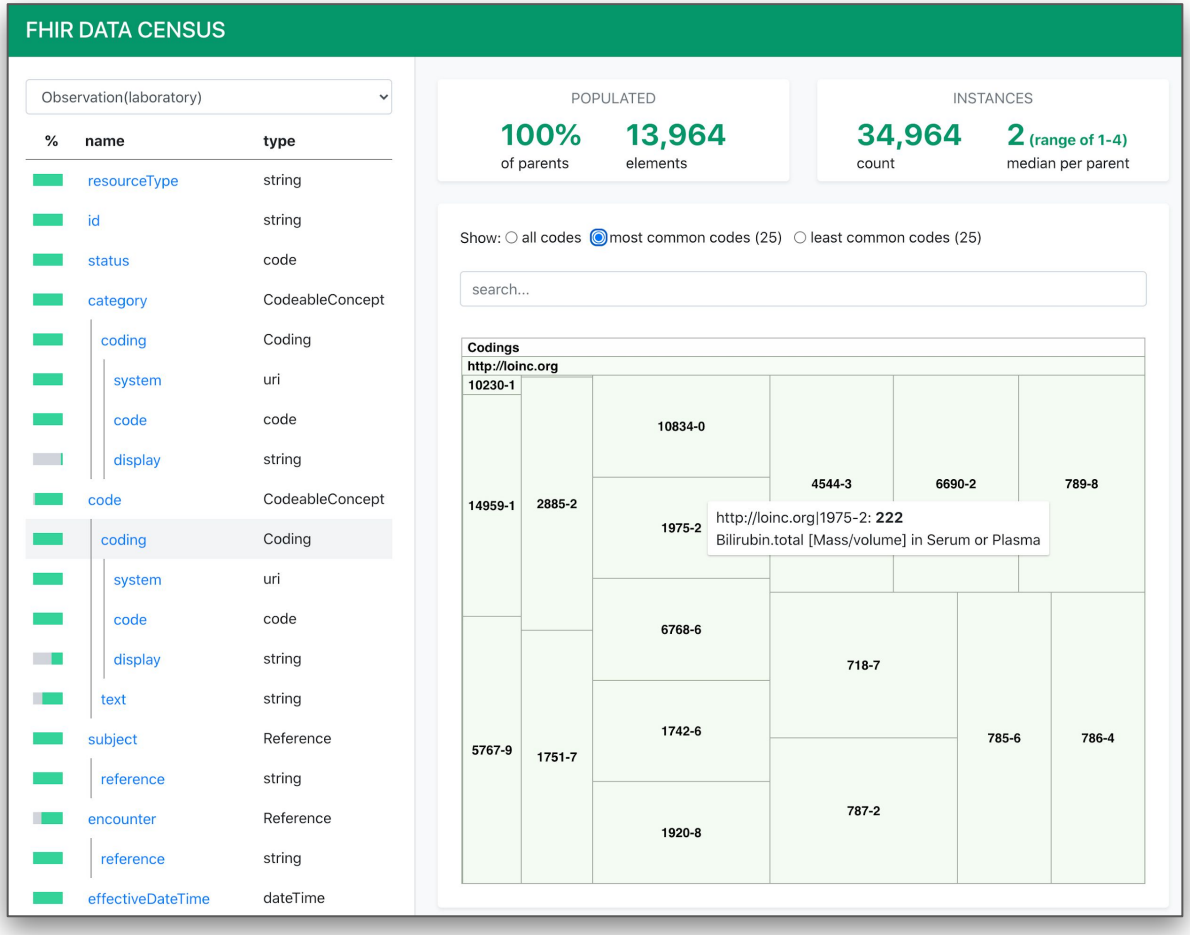
ONC Inferno Testing Tool

https://inferno.healthit.gov



FHIR Data Census Tool

https://github.com/sync-for-science/data-census



Adoption

Growing number of implementations!

Open Source FHIR Servers

- Microsoft
- HAPI
- Merative (IBM)

Commercial FHIR Servers

- Azure API for FHIR
- CareEvolution
- Firely Server (SQL endpoint)
- Minerva
- Google Healthcare API (in preview)

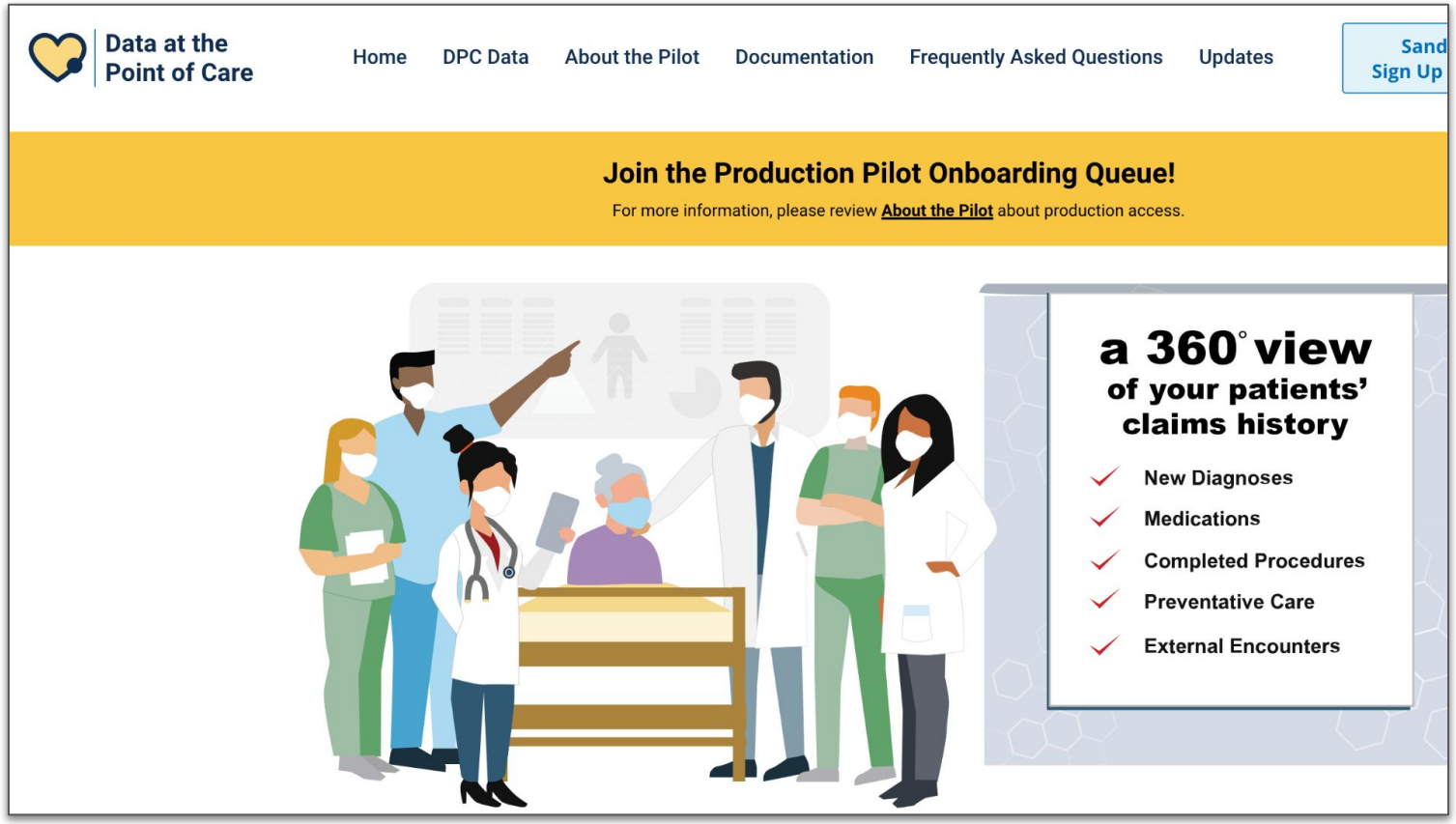
EHRs

- 269+ ONC Certified Implementations
- Including: Epic, Cerner and Meditech

Payor Data Servers

- CMS ACO Beneficiary Claims Data (pilot)
- CMS Data at the Point of Care (pilot)
- CMS Claims to Part D Sponsors (pilot)

dpc.cms.gov


Data at the Point of Care
Home
DPC Data
About the Pilot
Documentation
Frequently Asked Questions
Updates
Sand
Sign Up
Join the Production Pilot Onboarding Queue!
For more information, please review About the Pilot about production access.
a 360° view
of your patients' claims history
New Diagnoses
Medications
Completed Procedures
Preventative Care
External Encounters

Next Steps

Get Involved!

- Use the APIs in real-world use cases and other IGs - collect ideas for v3
- Open source modules (eg. de-identification, data quality, NLP)
- Bulk Import Operation (early draft proposal)
- Bulk patient matching (early draft proposal)
- Standardize analytic approaches (SQL-on-FHIR v2 project)

Resources

- Bulk Data Implementation Guide

  STU2 (v2): https://hl7.org/fhir/uv/bulkdata/

  STU1 (v1): http://hl7.org/fhir/uv/bulkdata/STU1.0.1/

- SMART Server Reference Implementation

  https://bulk-data.smarthealthit.org

- Bulk Data Discussion (Bulk Data Stream on FHIR Zulip Chat)

  https://chat.fhir.org/#narrow/stream/bulk.20data

- Dan@CentralSquareSolutions.com

ORGANIZED BY

Slides

https://bit.ly/bulk-june-2023

Feedback?

devdays.com/feedback