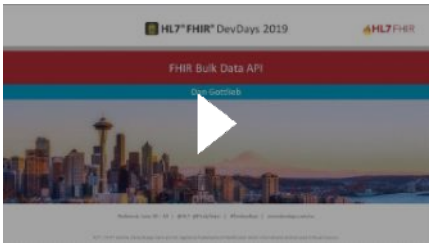
Bulk Data API

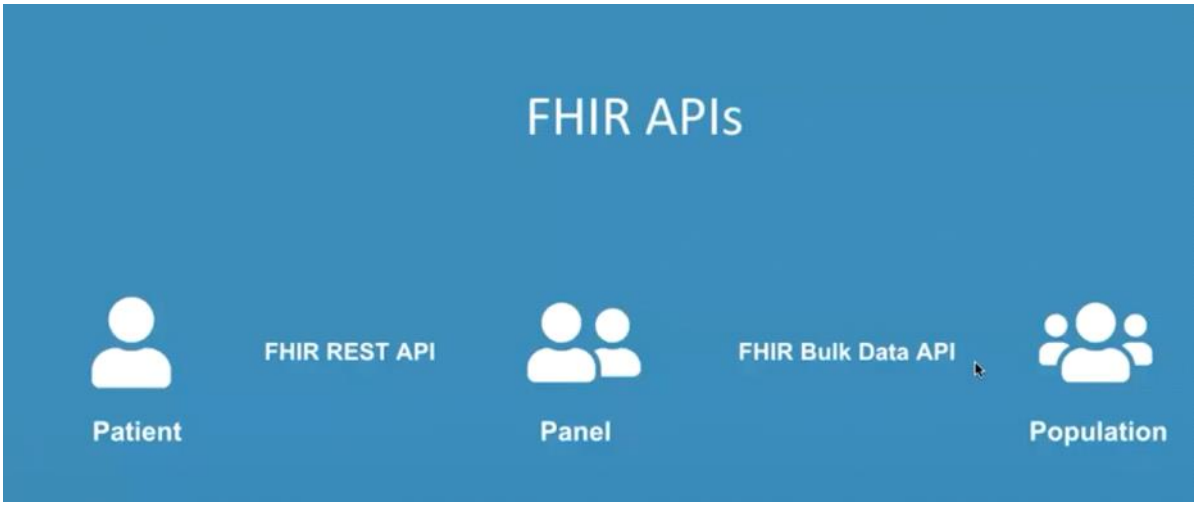
07 December 2022 04:34 PM

Dan Gottlieb - FHIR Bulk Data API | DevDays Redmond 2019

Dan Gottlieb - FHIR Bulk Data API and what's new in v2 | DevDays June 2022

https://hl7.org/fhir/uv/bulkdata/export/index.html





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

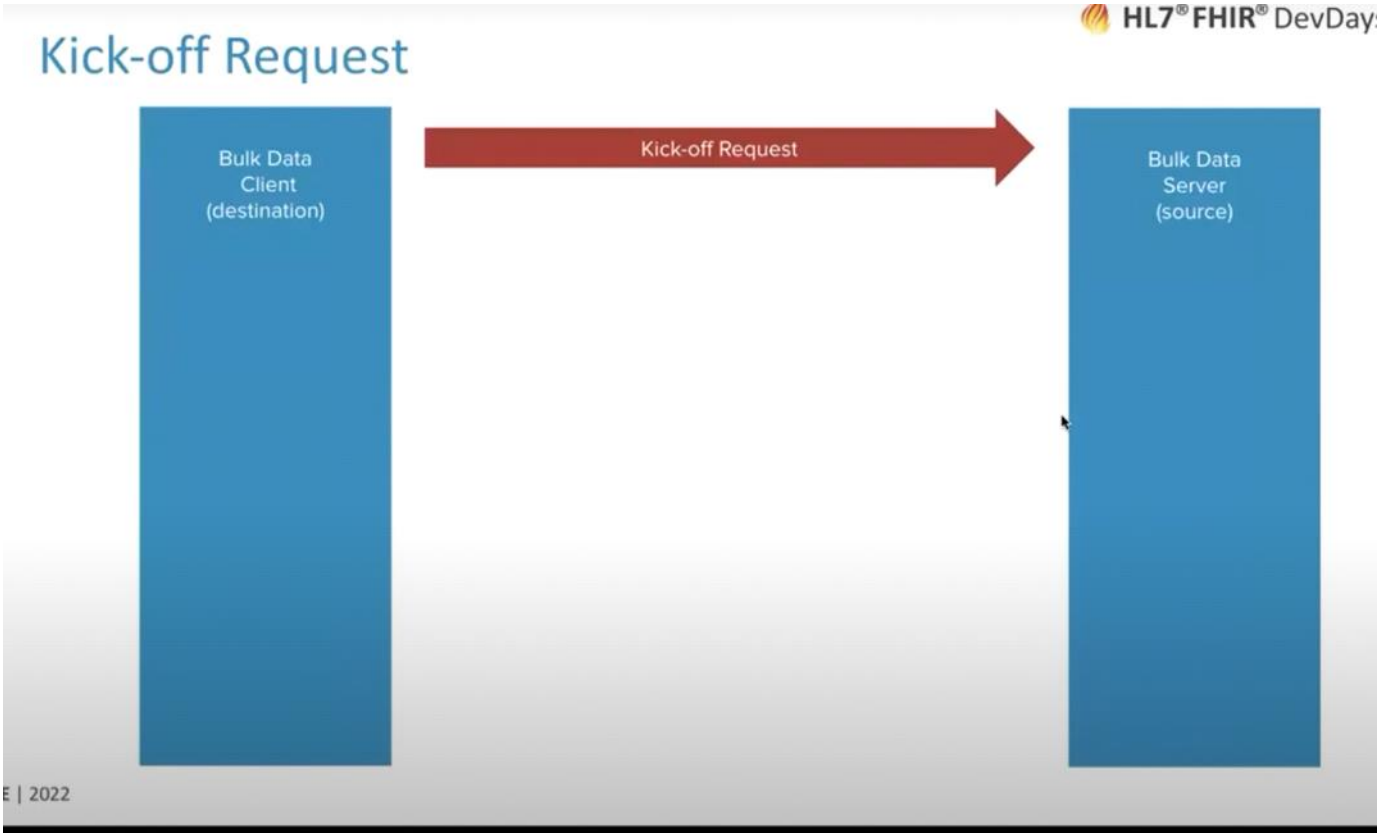
STU1 (v1)	STU2 (v2)
Initial version	Incorporates experience from early implementations
Published August 2019	Published November 2021

Kick-off Request

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

Updated in v2

- 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`



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)*

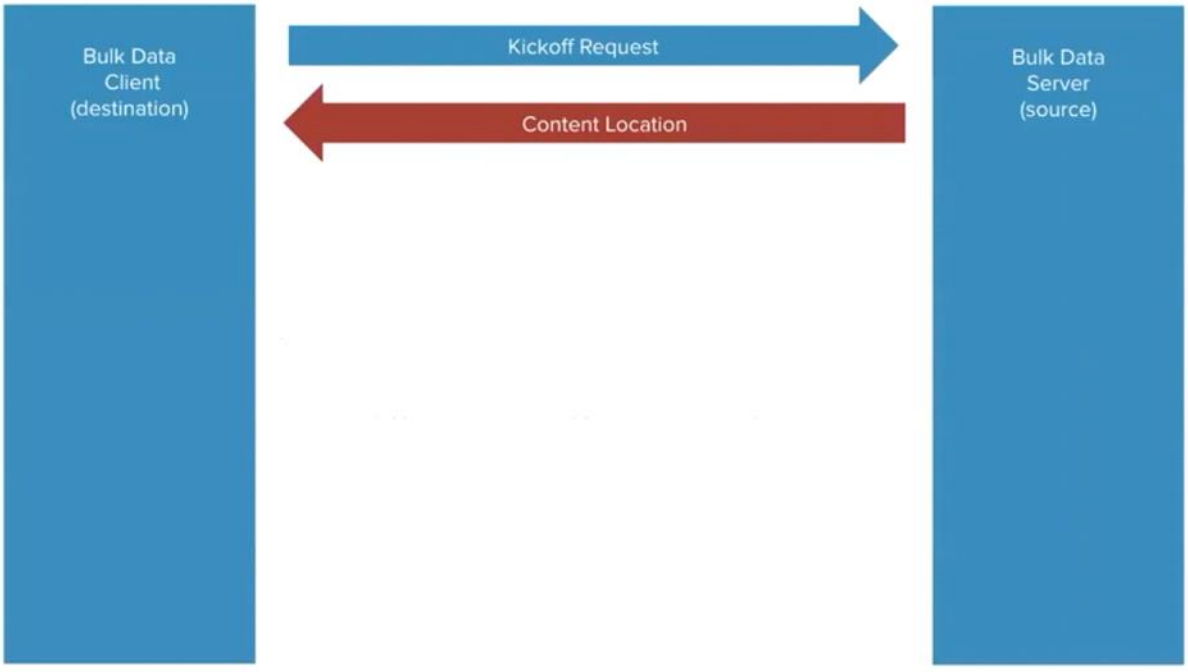
Updated in v2

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

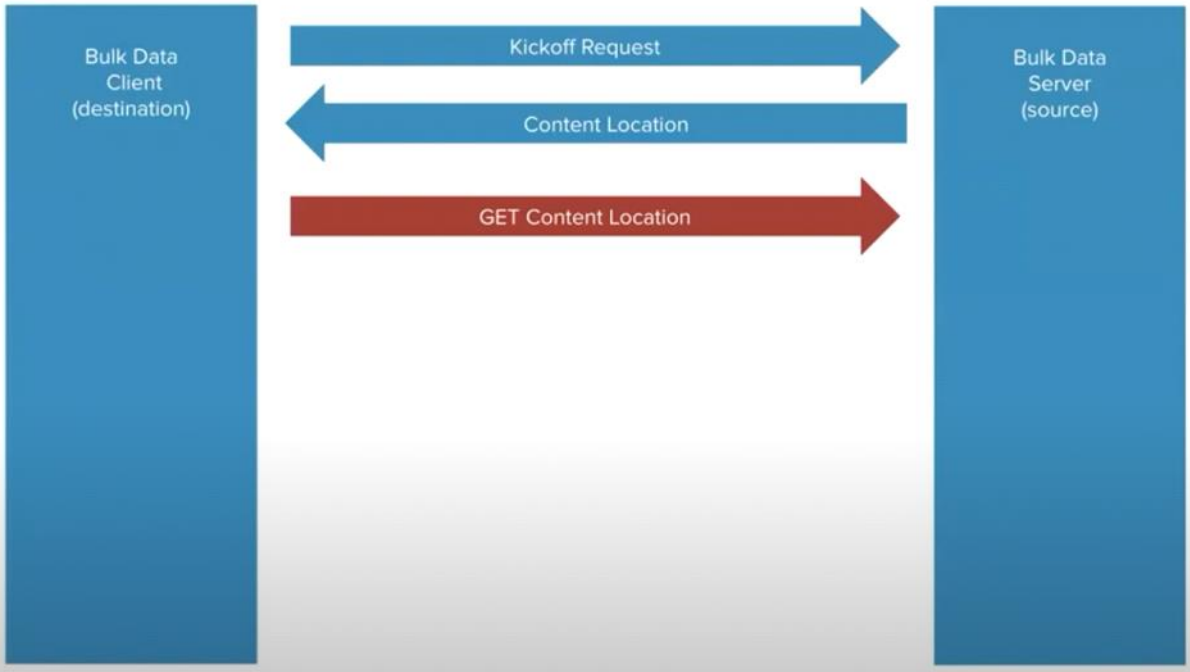


Kick-off Response Header

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

HL7® FHIR® DevDays

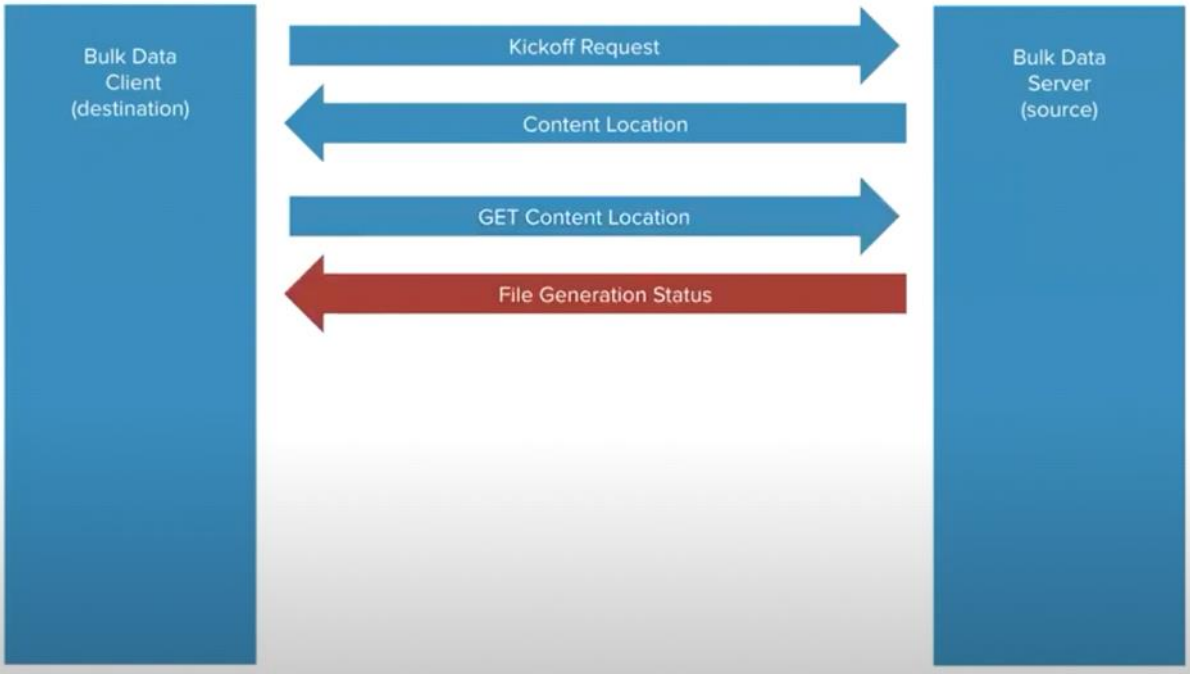
Status Request #1



2022 15

HL7® FHIR® DevDa

Status Response #1



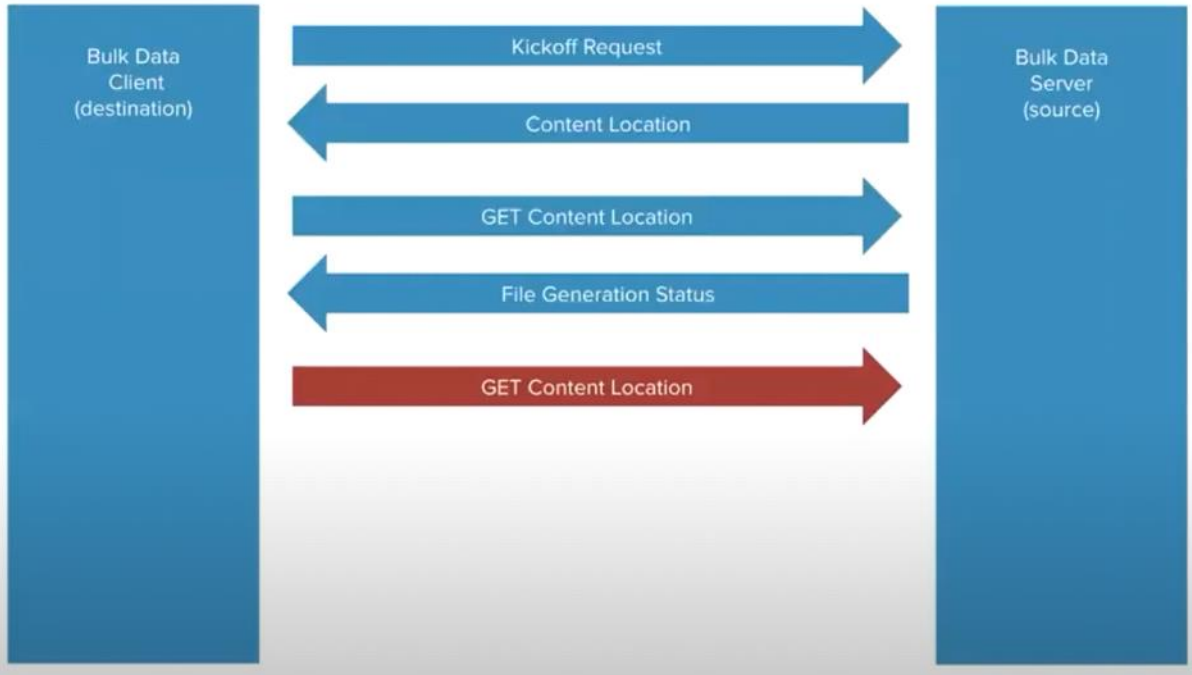
E | 2022

In-Progress Status Response Header

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

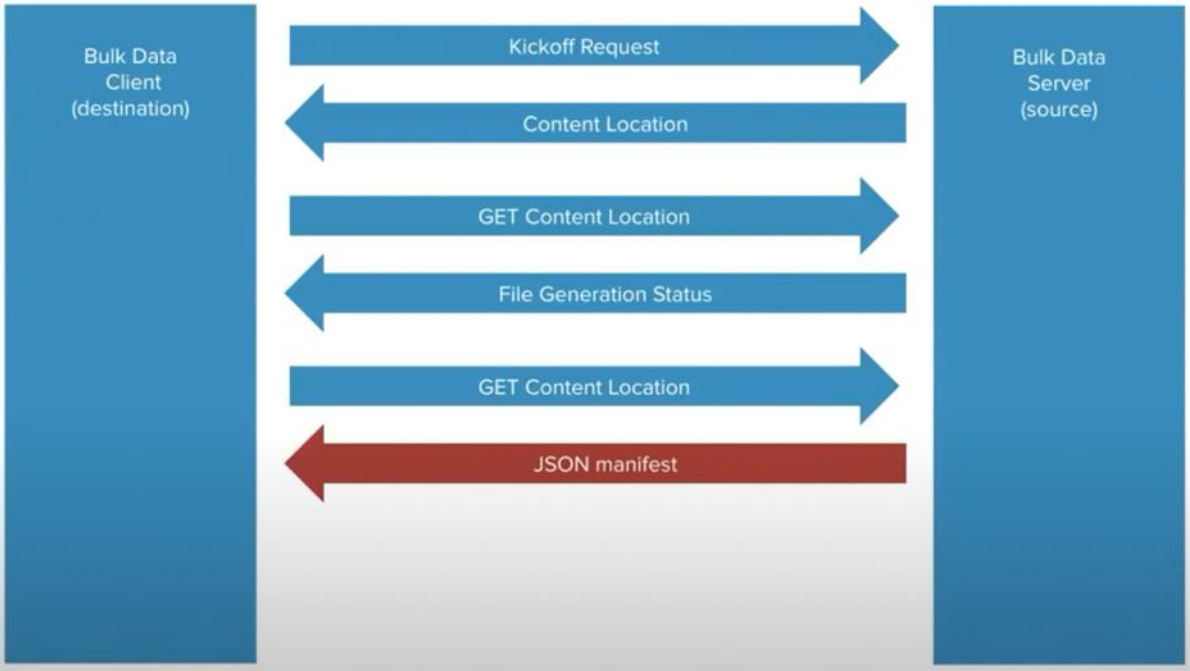
HL7® FHIR® DevDays

Status Request #2



HL7® FHIR® DevDays

Status Response #2



HL7® FHIR® De

Status Complete Response Body

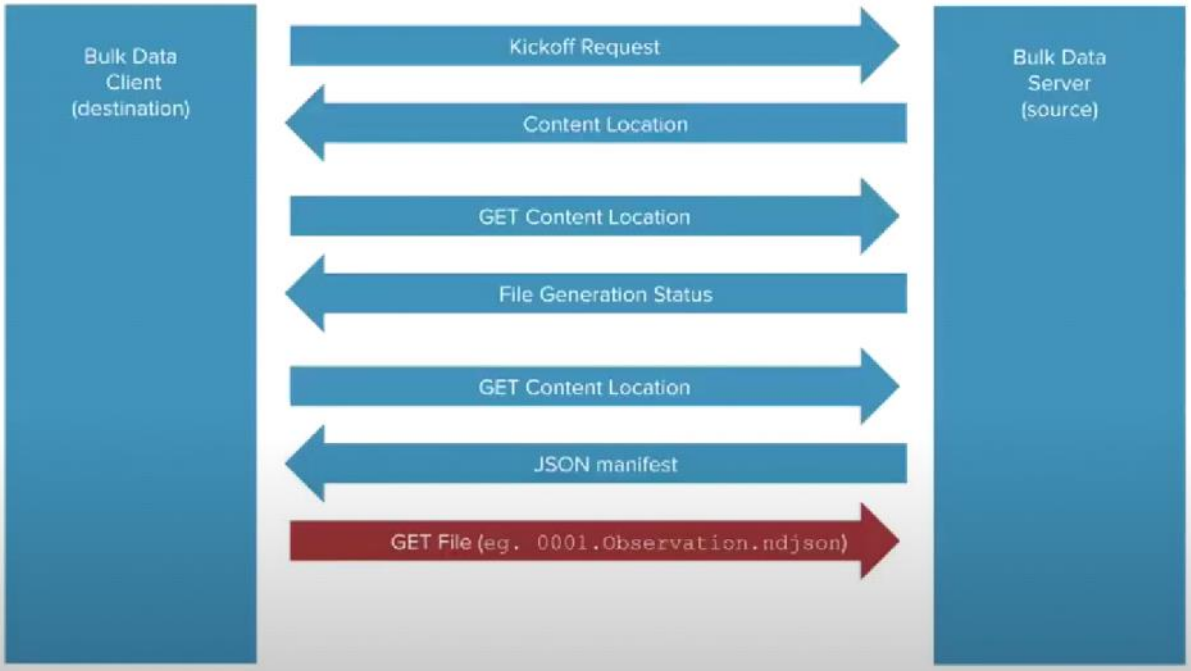
```
1  "transactionTime" : "2020-07-13T13:28:17.239Z",
2  "request" : "https://example.com/Patient/$export?_type=Patient,Observation",
3  "requiresAccessToken" : true,
4  "output" : [{
       "type" : "Patient",
       "url"  : "https://example.com/files/patient_file_1.ndjson"
   },{
       "type" : "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"
   }]
```

New in v2

Updated in v2

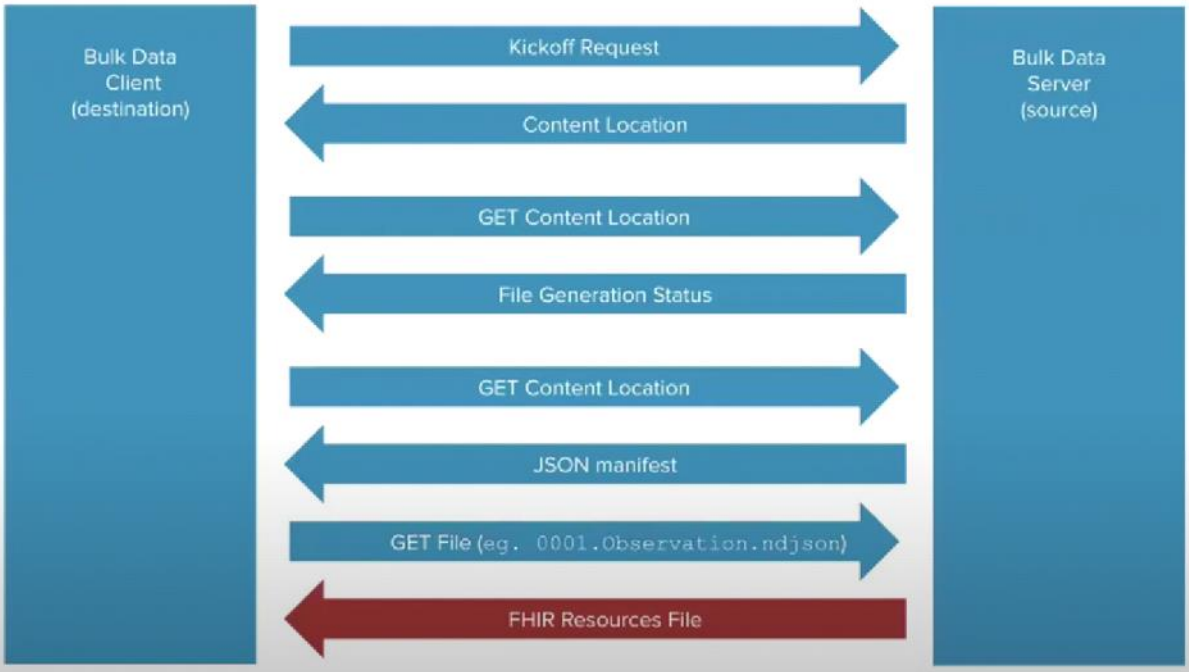
HL7® FHIR® DevDays

File Request

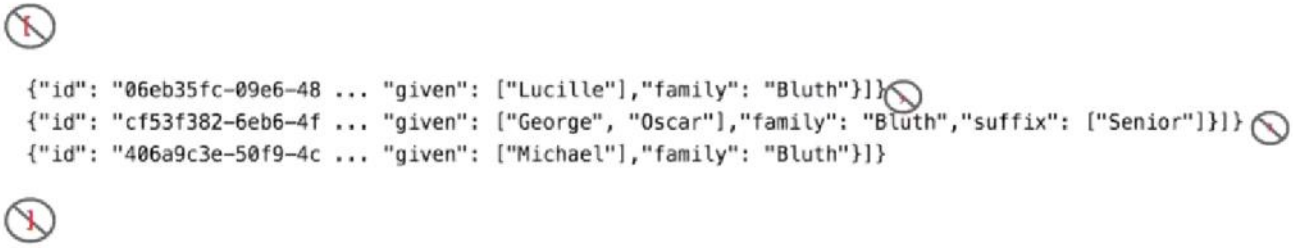


HL7® FHIR® DevDays

File Response



NDJSON

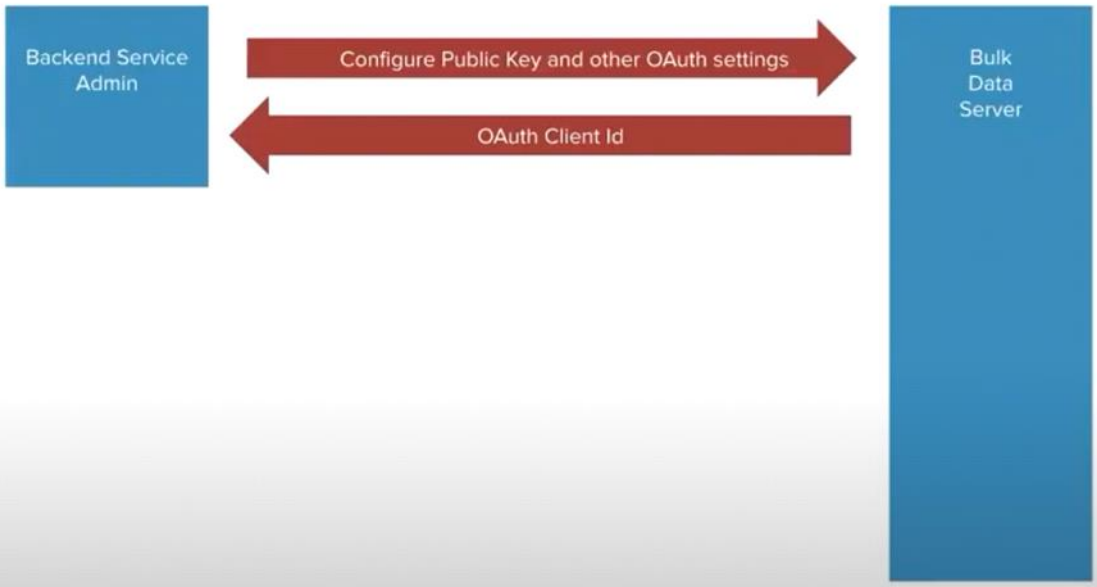


SMART Backend Services Authorization

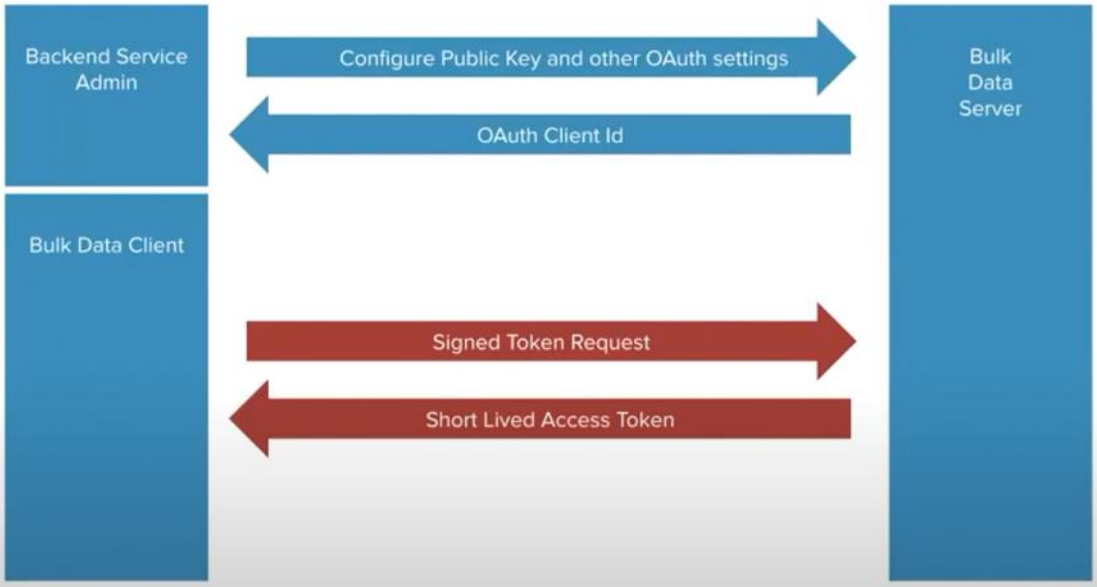
- Out-of-band app registration (can use Dynamic Client Registration or portal)
- Apps can register public key (JWKS format) or URL for public key
- Token requests signed with private key
- System level scope (parallels SMART “user” and “patient” scopes)

  `system/[resourceType].read`
- Short-lived access tokens

Registration Flow (once)



Authorization Flow (min. once per request)



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

https://bit.ly/fhir-bulk-api

FHIR BULK DATA API

Extending FHIR to Population Level Datasets

https://bit.ly/fhir-bulk-api

Today

Three recent examples:

- Large AMC syncing progress notes from a third party clinic into EHR
- Integration population health system with EHR system
- Machine learning startup obtaining training data from cloud EHR

https://bit.ly/fhir-bulk-api

Today

Three recent examples:

- Large AMC syncing progress notes from a third party clinic into EHR
- Integration population health system with EHR system
- Machine learning startup obtaining training data from cloud EHR

Other common use cases:

- Payer database to assess care quality
- Claims in EHR to provide comprehensive view
- Internal clinical data warehouse for study cohort identification
- Reportable disease submission or other registry

Sharing population level data is cumbersome

Healthcare organizations often use CSV EHR and data warehouse extracts to share clinical data (or don't share it at all)

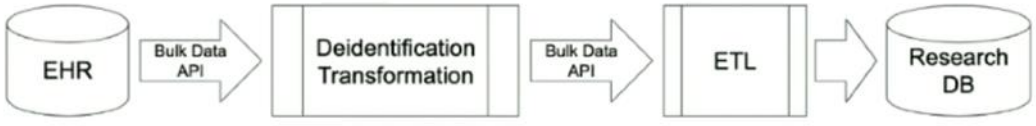
- **Proprietary data model** overburdens IT organizations and data analysts with manual and repetitive work to map the data
- **Proprietary (or manual)** data extraction needs to be built for each system and logistics like firewall support need to be configured each time
- **FHIR API (programming interface)** is great for obtaining data on patients or small panels, but inefficient for this type of large query

Let's enhance FHIR to support population level data access

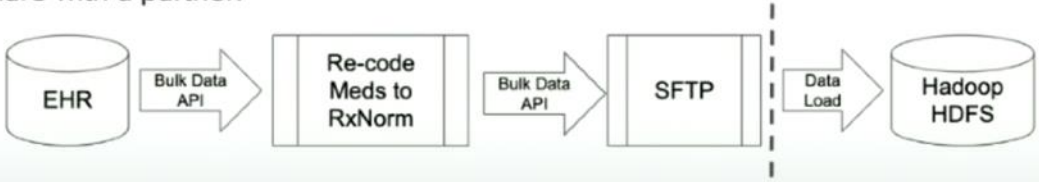
- **FHIR Resources as a standard data model** to simplify data parsing and mapping
- **FHIR Operation API** to initiate the data extracts
- **SMART Backend Services Authentication and Authorization** as security model

Pipelines can support many scenarios

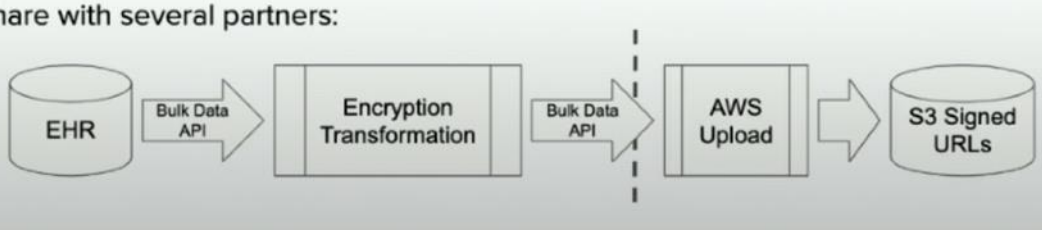
Create a deidentified view for researchers:



Share with a partner:



Share with several partners:

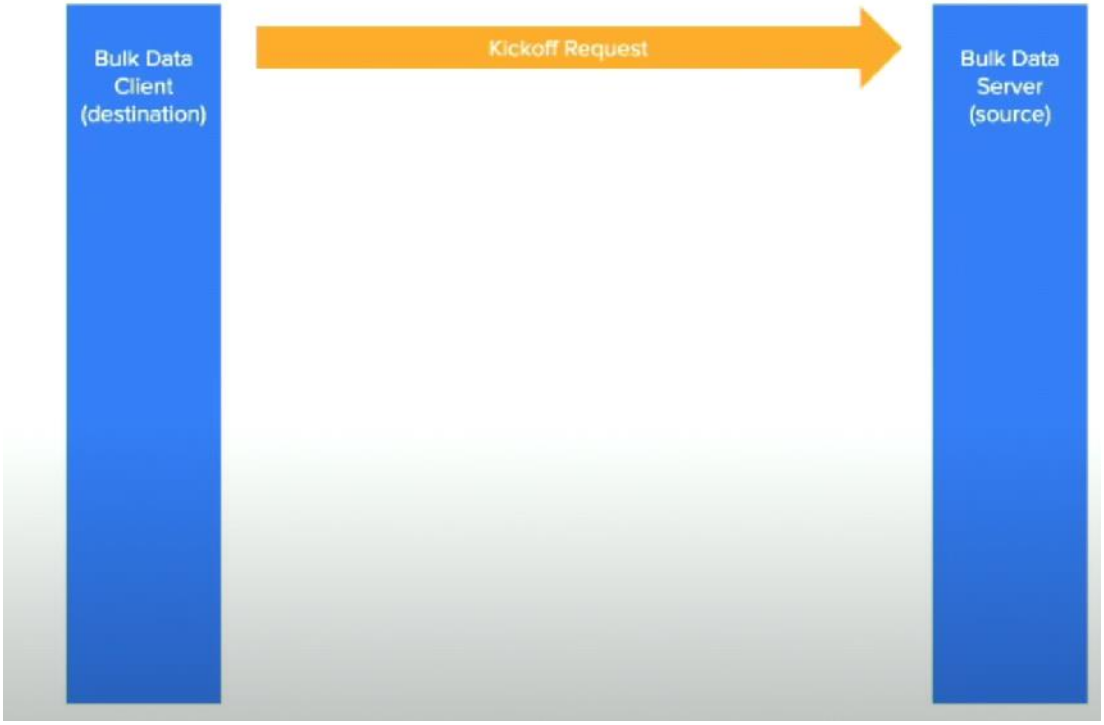


Focused Scope

Out of scope for initial version:

- Legal framework for sharing data between partners - BAAs, SLAs, DUAs continue to be negotiated and completed out-of-band
- Real-time data (although data loaded through bulk data can be supplemented at with synchronous FHIR REST API calls)
- Data transformation - different step of the ETL process
- Patient matching (although, it's possible to include identifiers like subscriber number in FHIR resources)

Kickoff Request



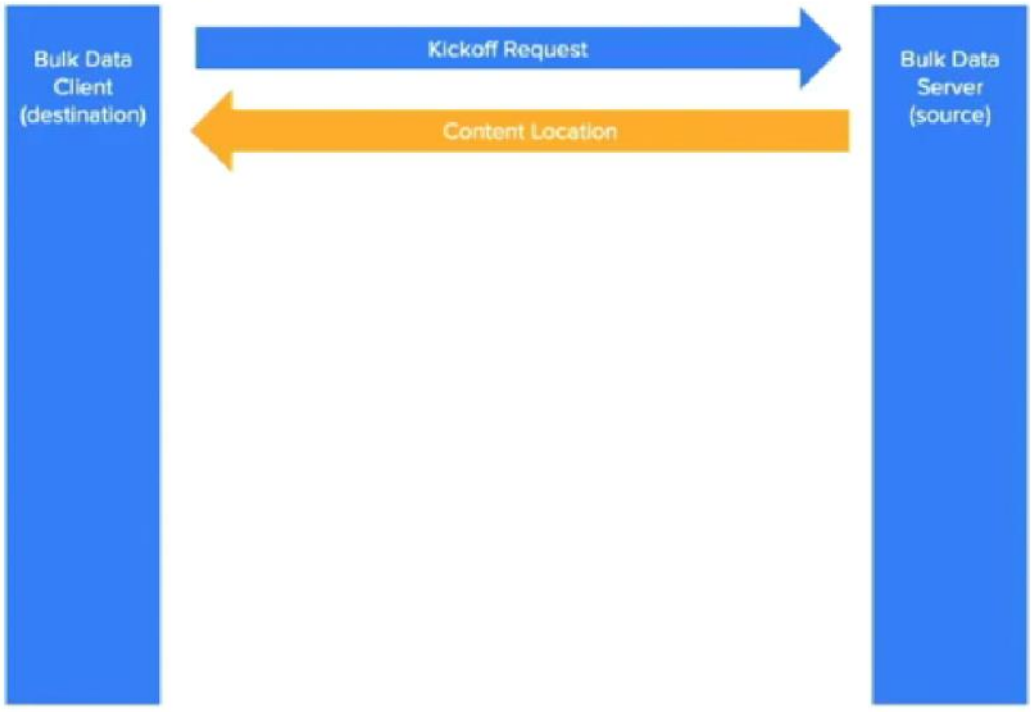
Kick Off Request

- FHIR Operation for all data on all patients
  `[FHIR Server Base]/Patient/$export`
- FHIR Operation for all data on a group of patients
  `[FHIR Server Base]/Group/[group id]/$export`
- FHIR Operation for all data on the server
  `[FHIR Server Base]/$export`
- Asynchronous requests with status polling
  `Prefer: respond-async`

Query Parameters (filters)

_outputFormat	The format for the generated bulk data files (currently, only **ndjson** is supported)
_since	FHIR resource modified date (FHIR instant timestamp)
_type	Comma delimited list of FHIR resource types
[group id]	Predefined set of patients (research cohort, plan members, employer)
_typeFilter	Experimental syntax to limit data returned

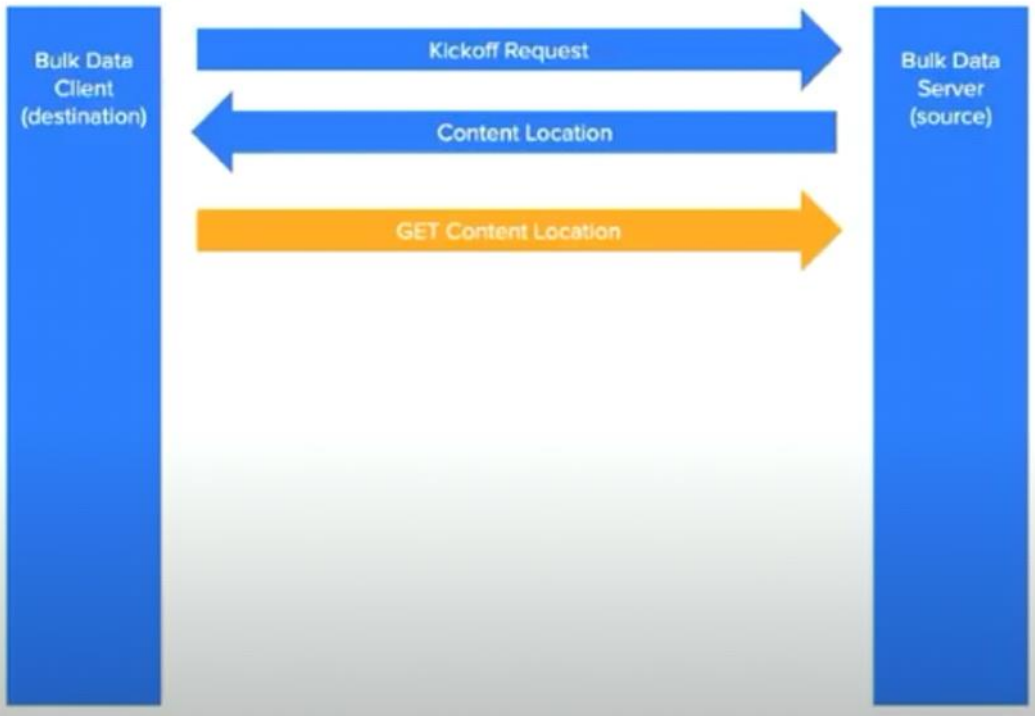
Kickoff Response



Kick Off Response

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

Status Request #1



Status Response

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

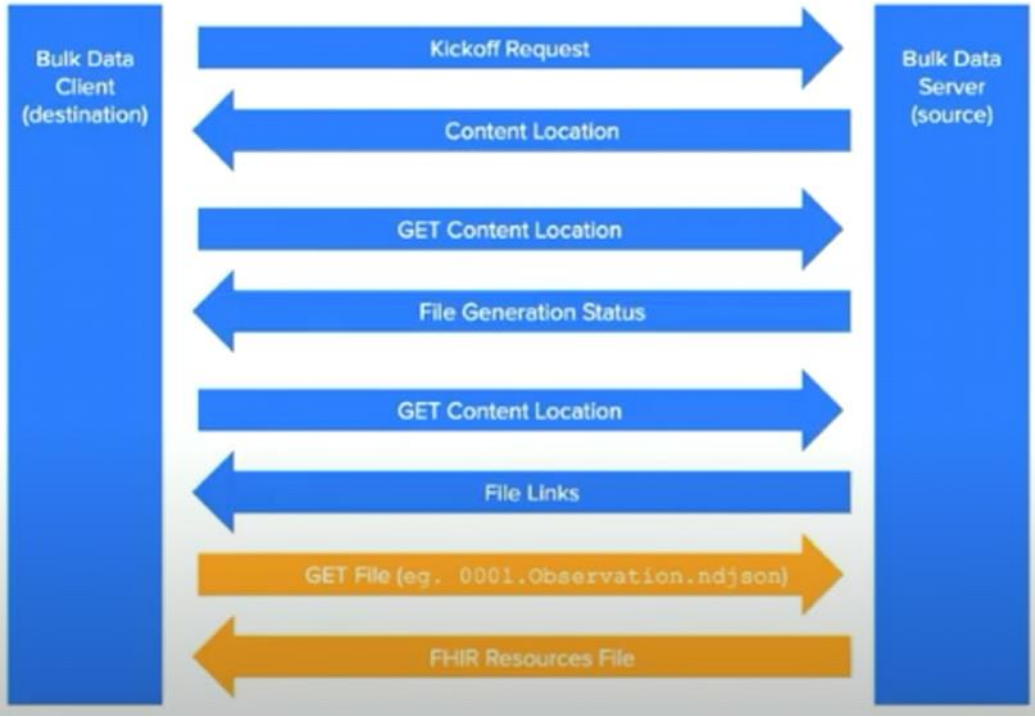
Status Response Header

```
Status: 200 OK
Expires: Mon, 12 Mar 2018 23:59:59 GMT
```

Status Response Body

```
{
1  "transactionTime" : "[instant]",
2  "request" : "[base]/Patient/$export?_type=Patient,Observation",
3  "requiresAccessToken" : true,
4  "output" : [{
       "type" : "Patient",
       "url"  : "http://serverpath2/patient_file_1.ndjson"
   },{
       "type" : "Patient",
       "url" : "http://serverpath2/patient_file_2.ndjson"
   },{
       "type" : "Observation",
       "url" : "http://serverpath2/observation_file_1.ndjson"
   }],
5  "error" : [{
       "type" : "OperationOutcome",
       "url"  : "http://serverpath2/error_file_1.ndjson"
   }]
}
```

File Request



FHIR Resources

Data models representing discrete clinical and administrative units (patient, practitioner, allergy, medication order, etc.)

- Currently around 100 have been defined
- Can reference other resources by their URL
- Don't include the kitchen sink

  "We only include data elements if we are confident that most normal implementations using that resource will make use of the element"
  – Grahame Grieve (FHIR Product Director)
- But, support extensions
- MU3 Common Clinical Dataset defines subset

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"}]}
]
```

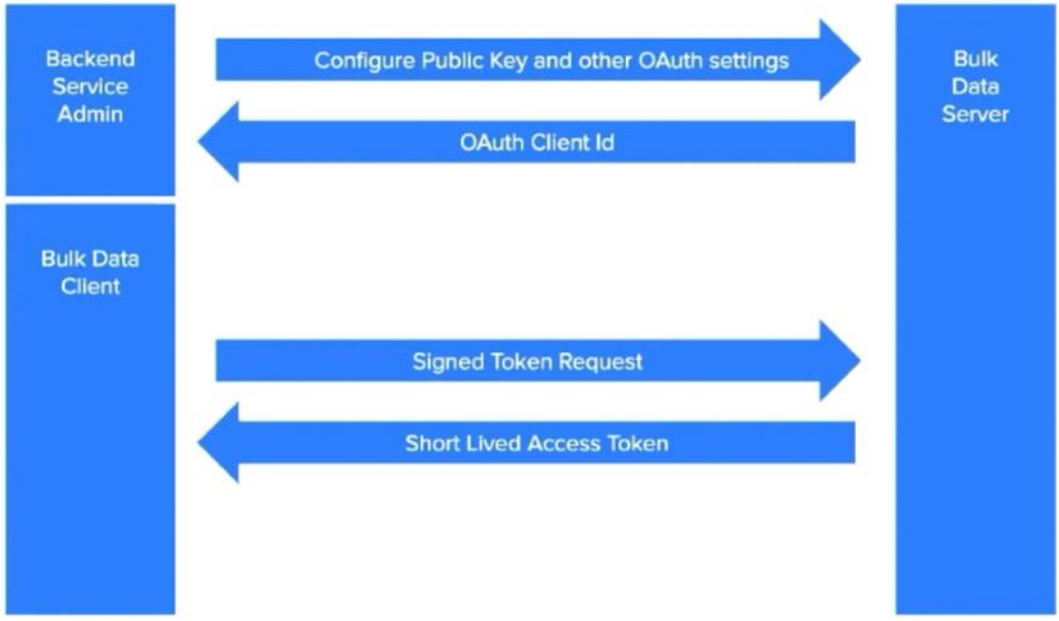
The red content are diff b/w regular json

SMART 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
- System level scope (parallels SMART "user" scope)

  `system/[resourceType].read`
- Short-lived access tokens

Security Flow (SMART Backend Services)



Security Flow (SMART Backend Services)