

FHIR **BULK DATA** API

Extending FHIR to Population Level Datasets

Dan Gottlieb and @JoshCMandel, 2018



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

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)

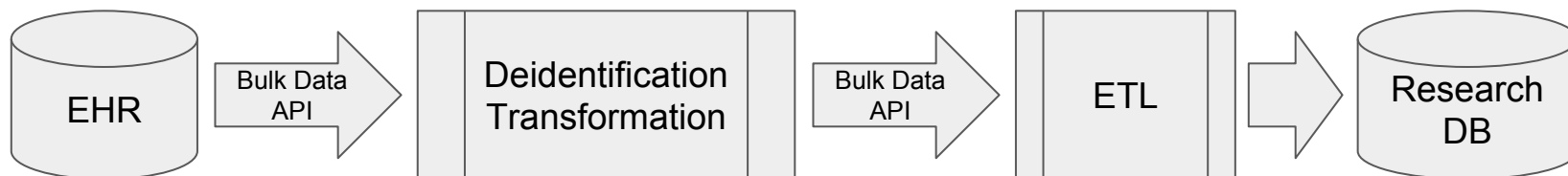
- **Proprietary data model** overburdens IT organizations and data analysts with manual and repetitive work to map the data
- **Proprietary (or manual)** data extraction needs to be built for each system and logistics like firewall support need to be configured each time
- **FHIR API (programming interface)**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

Let's enhance to support population level data access

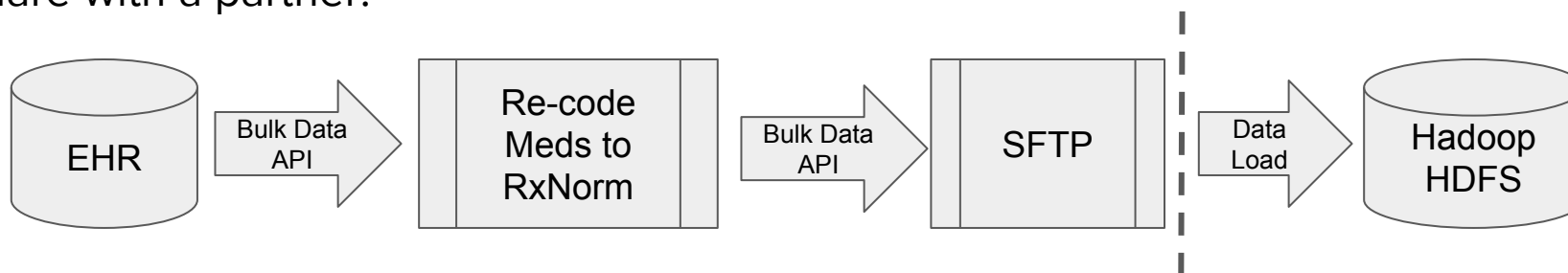
- **FHIR Resources as a standard data model** to simplify data parsing and mapping
- **FHIR Operation API** to initiate the data extracts
- **SMART Backend Services Authentication and Authorization** as security model

Pipelines can support many scenarios

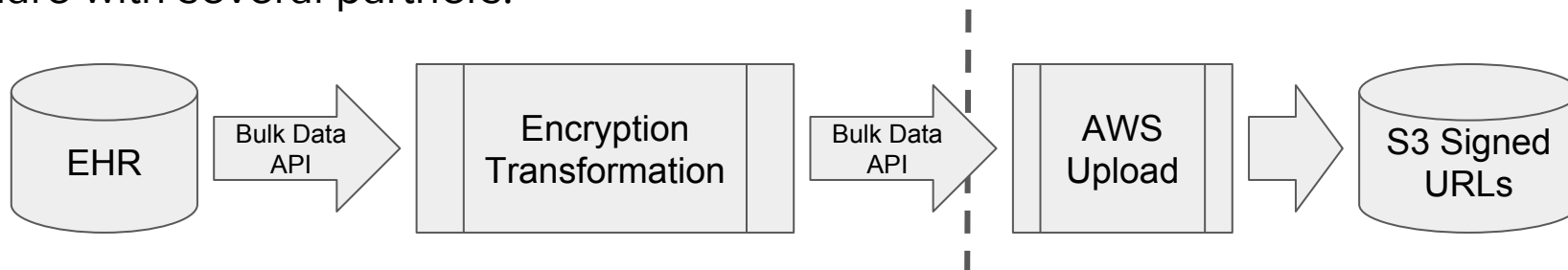
Create a deidentified view for researchers:



Share with a partner:



Share with several partners:



Design Goals

- Focus on enabling automated communication between backend services and EHRs/clinical systems
- Use mature, stable technologies wherever possible
- Small API surface area
 - Limit number of query parameters
 - Limit number of serialization formats
- Reuse as much of existing FHIR semantics as possible
 - Data models
 - API format and data types
 - Implementation guide structure
- Use existing standards based authentication and authorization
 - Base on widely used OAuth (SMART) standard
- Structure for efficiently generating and loading large datasets
 - Asynchronous operation
 - One data type per file
 - Streaming data

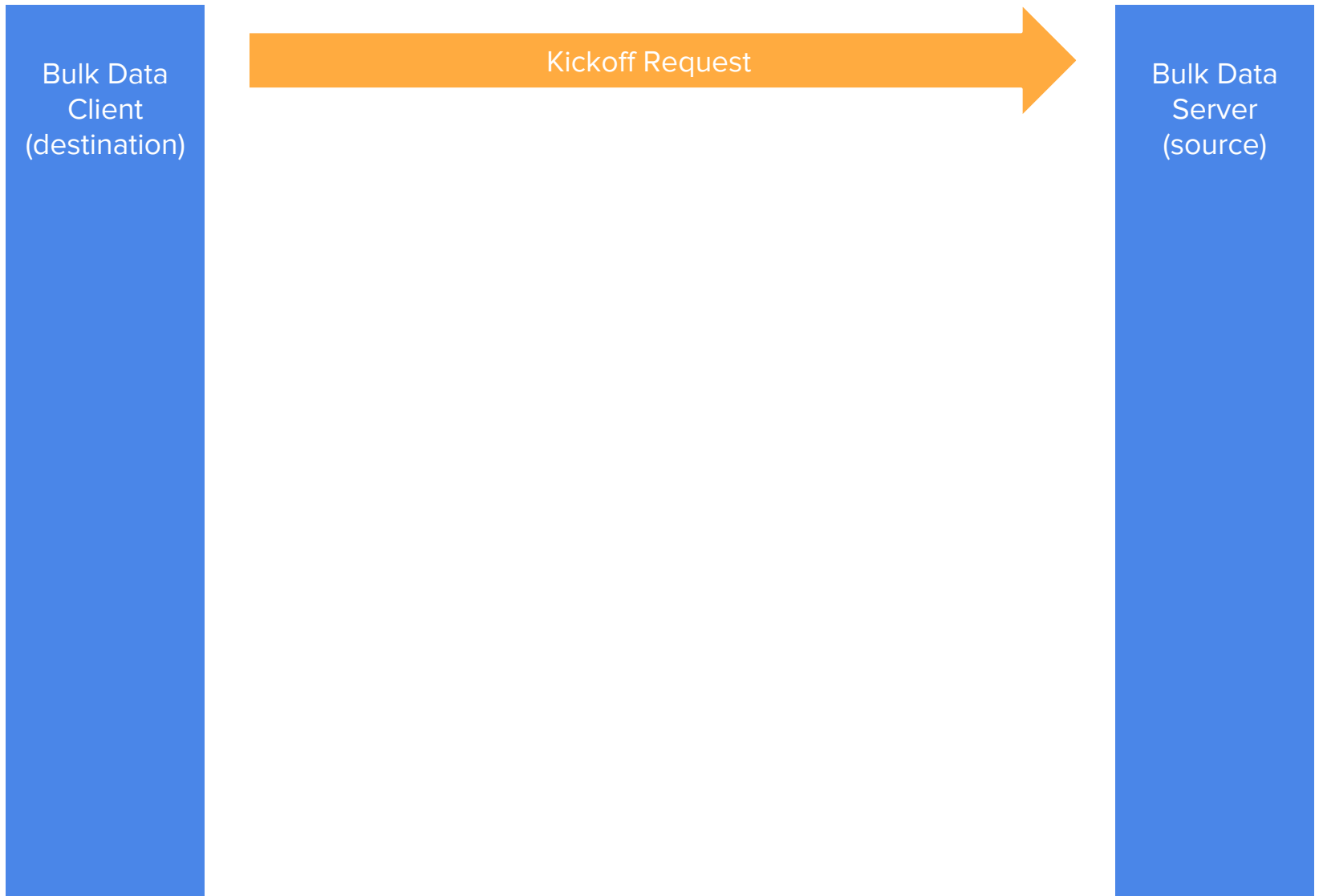
Focused Scope

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)

Architecture

Kickoff Request



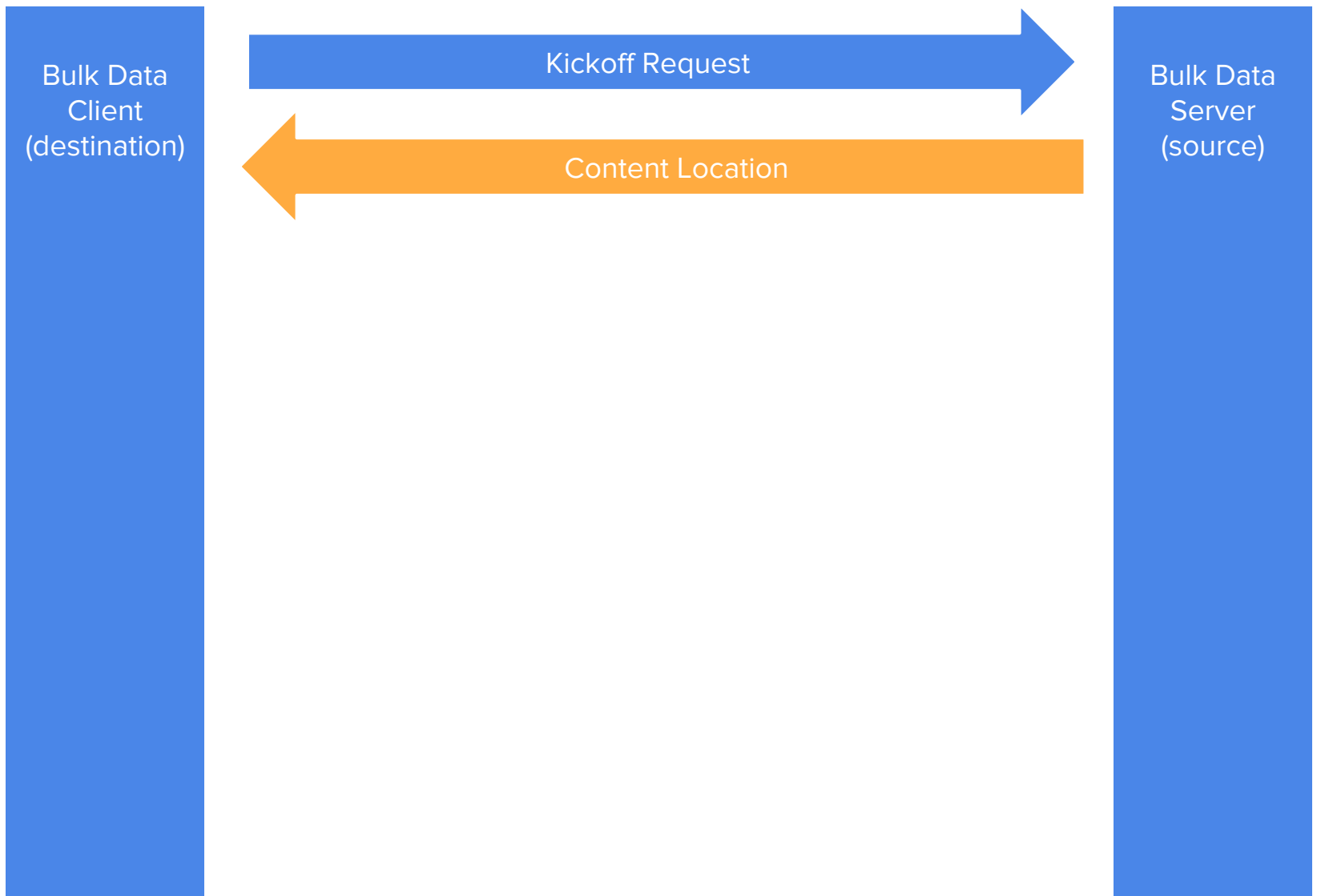
Kick Off Request

- 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 **(new!)**
`[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

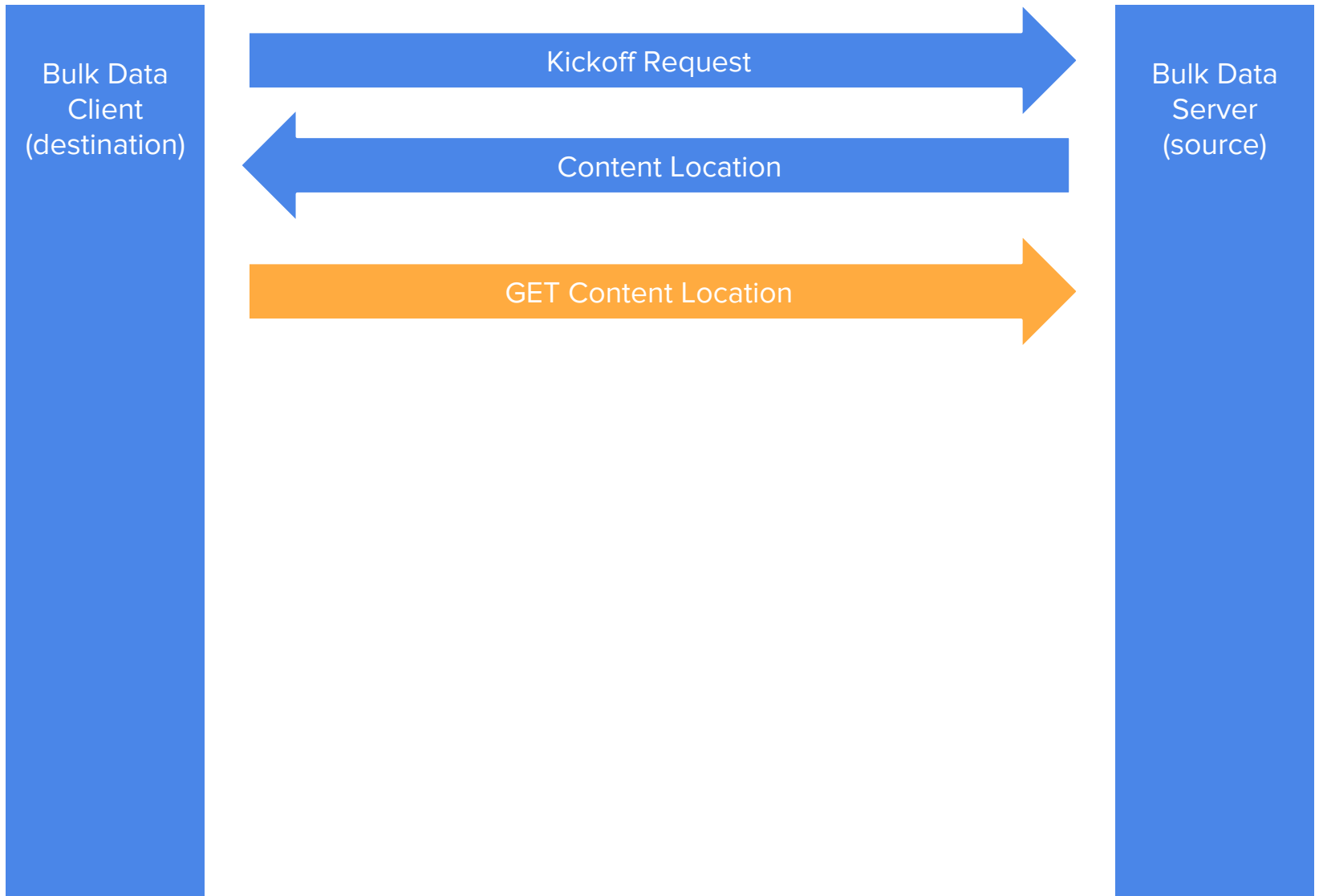


Kick Off Response

Status: 202 Accepted

Content-Location: [URL for status or deleting request]

Status Request #1



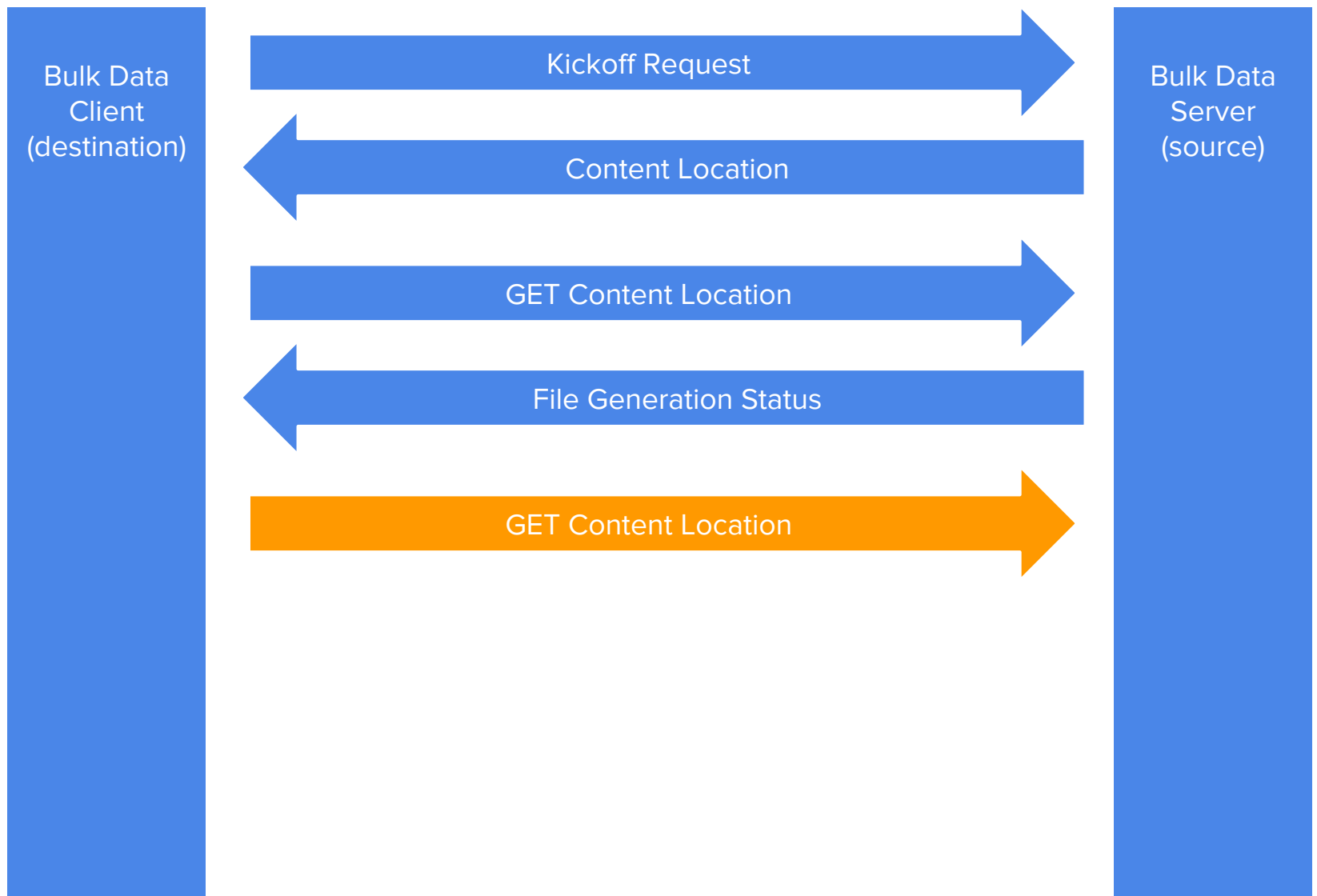
Status Response

Status: 202 Accepted

X-Progress: "50% complete"

Retry-After: 120

Status Request #2



Status Response Header

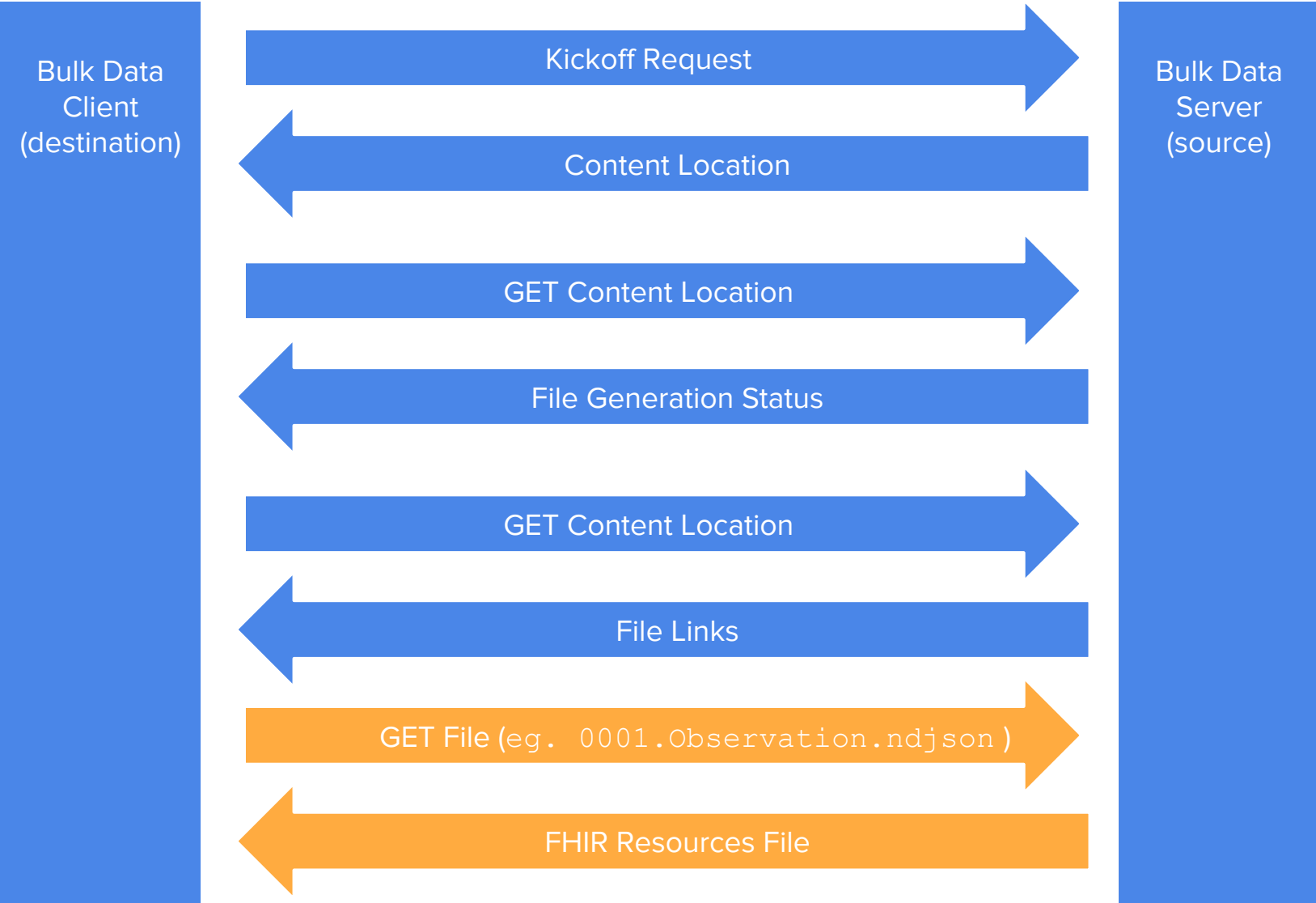
Status: 200 OK

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

[

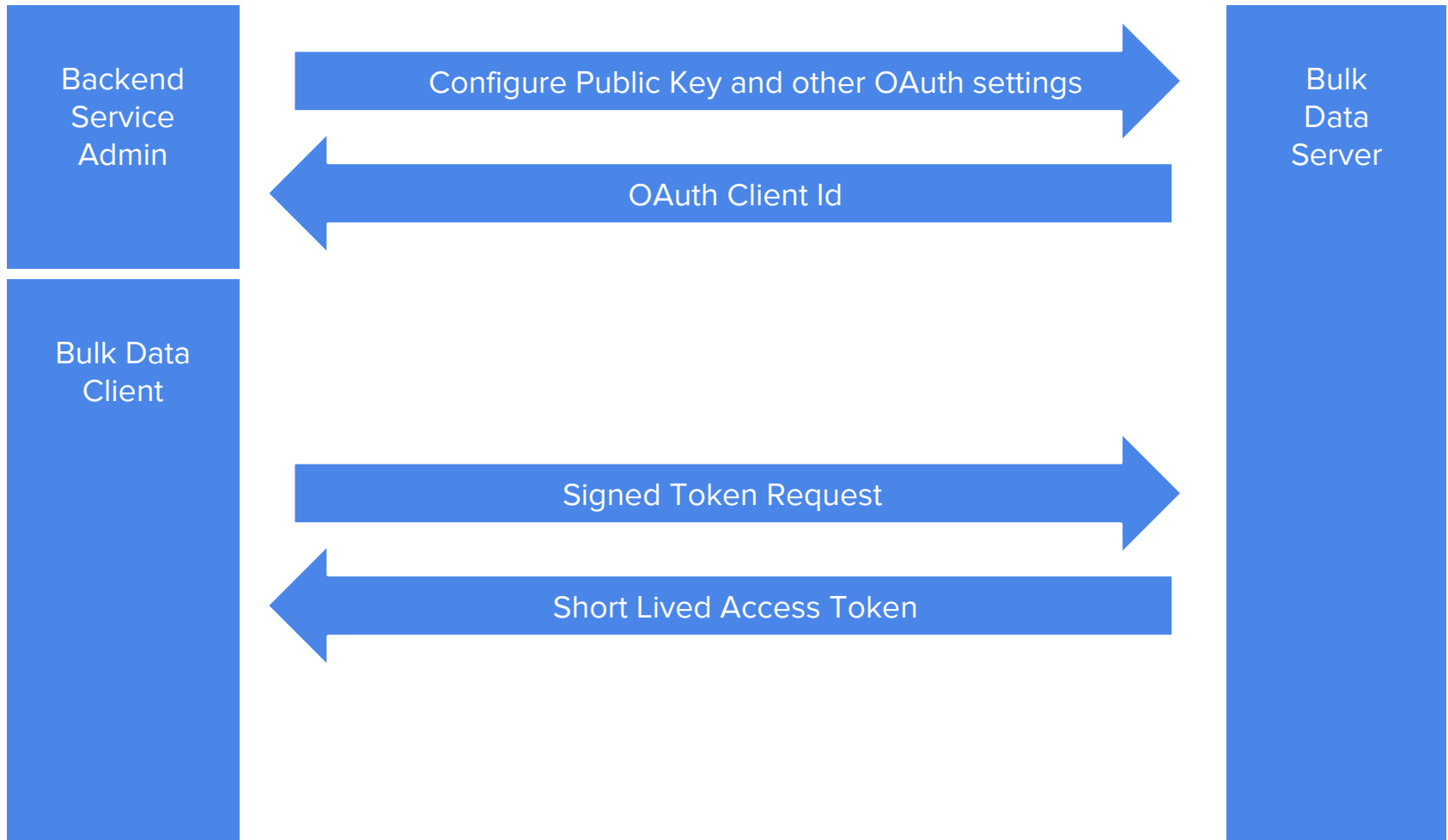
```
{"id": "06eb35fc-09e6-48 ... "given": ["Lucille"],"family": "Bluth"}} ,  
{"id": "cf53f382-6eb6-4f ... "given": ["George", "Oscar"],"family": "Bluth","suffix": ["Senior"]}] }  
{"id": "406a9c3e-50f9-4c ... "given": ["Michael"],"family": "Bluth"}}
```

]

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 **(new!)**
- 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)



Recent Changes (since v0.2)

Export Operation

- System wide \$export to encompass resources like ValueSet
- Renamed “secure” to “requiresAccessToken”
- Added “Error” and “Count” properties to completion response
- Added *experimental* _typeFilter query parameter to limit data being exported


```
$export?  
  _type=  
    MedicationRequest,  
    Condition&  
  _typeFilter=  
    MedicationRequest%3Fstatus%3Dactive,  
    MedicationRequest%3Fstatus%3Dcompleted%26date%3Dgt2018-07-01T00%3A00%3A00Z
```

Authorization

- Updated recommended access token lifetime to five minutes
- Clarify client registration mechanisms including public keys
 - Public key types
 - EHRs validate RS384 and ES384 signatures
 - Public Key registration based on JWKS with "bare key" properties for RSA and ECDSA
 - Preferred via "JWKS URL", so clients can update keys at will
 - Acceptable by directly providing a JWKS (static key, submitted at registration time)
 - Detailed JWT header (alg, kid, typ, jku) and body (iss, sub, aud, exp, jti) claims
 - Re-use client_id for "iss" and "sub" JWT body claims
- Clarify rules for servers verifying authentication JWTs
 - Resolve public keys before validating
 - Ensure JTI is unique for a "sub" within the 5-minute JWT lifetime
- Added error handling expectations (per OAuth 2 base spec)

Tools and Resources

SMART Reference Server Implementation

 SMART Bulk Data Server **alpha** Save ▾

Launch Options

Public Key Generate Keys

Paste your RSA-256 public key or use the button above to generate a new one

Service URL

Resources per File

Advanced

Access Token Lifetime

Simulate Error for Testing

Database Size

Simulated file generation duration


Launch Configuration

Download as JSON


FHIR Server URL Try Sample App

Authentication URL

Client ID

 [Bulk Data Docs](#) | [Backend Services Docs](#) | [View NodeJS Sample App](#)

SMART Sample GUI Client



FHIR Bulk Downloader

sample app

Resources to Download

<input checked="" type="checkbox"/> AllergyIntolerance (30)	<input checked="" type="checkbox"/> CarePlan (239)	<input checked="" type="checkbox"/> Claim (1,706)	<input checked="" type="checkbox"/> Condition (431)
<input checked="" type="checkbox"/> DiagnosticReport (375)	<input checked="" type="checkbox"/> Encounter (1,369)	<input checked="" type="checkbox"/> Goal (200)	<input checked="" type="checkbox"/> Immunization (726)
<input checked="" type="checkbox"/> MedicationRequest (337)	<input checked="" type="checkbox"/> Observation (5,157)	<input checked="" type="checkbox"/> Organization (131)	<input checked="" type="checkbox"/> Patient (100)
<input checked="" type="checkbox"/> Procedure (732)			

Patients Group

No Group (include all the patients) ▾

Instead of exporting all patients, you can select a group and only work with patients within that group.

Filter by Modification Date/Time

No Time Filter (include everything) ▾

You can filter the data and only include resources that have been modified after the specified date.

Download Link

[http://localhost:9443/eyJlcmluOiIiLCJwYWdlIjoxMDAwMwZHVyYjoxMCwidGx0IjoxNSwibSI6MX0/fhir/Patient/\\$everything](http://localhost:9443/eyJlcmluOiIiLCJwYWdlIjoxMDAwMwZHVyYjoxMCwidGx0IjoxNSwibSI6MX0/fhir/Patient/$everything) [Prepare Download!](#)

Files to Download 13

1.AllergyIntolerance.ndjson	1.CarePlan.ndjson	1.Claim.ndjson	1.Condition.ndjson
1.DiagnosticReport.ndjson	1.Encounter.ndjson	1.Goal.ndjson	1.Immunization.ndjson
1.MedicationRequest.ndjson	1.Observation.ndjson	1.Organization.ndjson	1.Patient.ndjson
1.Procedure.ndjson			

Example Analytics

With tools like Apache Drill, Spark SQL, and data warehouses from cloud vendors

Rich query with joins over JSON data allows queries like

Apache Drill example query to find most common Observation codes...

```
select      c.coding.code, count(*)
from        dfs.root.`/data/fhir/Observation*.json`),
           unnest(t.code.coding) c(coding)
group by    c.coding.code
order by    2 desc
limit       10;
```

See github.com/fhir-fuel/fhir-storage-and-analytics-track for details

Next Steps

Project Timeline

2017	Initial Specification SMART/ONC Meeting on Bulk Data
2018	SMART Bulk Data Reference Implementation Server New Orleans Connectathon - Bulk Data API Draft Implementation Guide Argonaut Kick-Off Cologne Connectathon - Authentication and Authorization Baltimore Connectathon - Integration Testing Argonaut Funded Security Review
2019	Final Draft Implementation Guide FHIR Implementation Guide For Bulk Data 1.0

Open Questions & Future Work

- How do the APIs perform for real-world use cases?
- Are other forms of bulk data APIs required?
- \$import Operation?
- Standardizing group management

Get Involved!

- Implementations of bulk data servers and clients
- Open source test suite providing feedback on implementations (including error conditions and any optional bells/whistles)
- Open source reference implementation of components like de-identification and master patient index that can act as nodes in an export pipeline

Links

- Draft Bulk Data Documentation:
<https://github.com/smart-on-fhir/fhir-bulk-data-docs>
- SMART Server Reference Implementation:
<https://bulk-data.smarthealthit.org>
- SMART Client Reference Implementation:
<https://github.com/smart-on-fhir/sample-apps-stu3/tree/master/fhir-downloader>
- Discussion Group (Bulk Data Stream on FHIR Zulip Chat):
<https://chat.fhir.org/#narrow/stream/bulk.20data>