**Introduction:**

The Health Record Ingestion Service (HRI) leverages IBM Event streams as one of its core underlying technologies when serving as a message broker between data producers and data consumers. Built on Apache Kafka, IBM Event Streams is a high-throughput, fault-tolerant, event streaming platform that helps teams build intelligent, responsive, event-driven applications.

* For more information on HRI:

<https://pages.github.ibm.com/wffh-hri/docs/>

* For more information on IBM Event Streams: <https://cloud.ibm.com/docs/EventStreams?topic=EventStreams-getting_started>

Individual messages are streamed to HRI by data producers and are then available on Kafka topics for data consumers to consume. However, the IBM Event streams technology enforces a 1 MB message size limit on any single message. In the event the data producer needs to send messages >1 MB, there are a few different implementation options to consider. The sections below provide guidance on potential implementation methods and when one should be selected over another.

*Note: the scope of this guidance is limited to splitting FHIR bundles and individual FHIR resources specifically. It does not include guidance on splitting other types of data formats (HL7 v2, 837i/p, etc.).*

**Message categories:**

There are 2 categories that large data packages typically fall into. A separate implementation approach is recommended for each category.

* Category 1: Messages being sent to HRI are 'full FHIR' (e.g.: at the end of message ingestion, the full view of the data should be representable directly in FHIR).

The recommended approach for this category of large data is to split apart the FHIR resource in such a manner that it will be able to be reconstructed post-HRI. Specific guidance is provided below in the ‘Category 1 messages’ section.

* Category 2: Messages being sent to HRI are large data blobs (text/binary/pdf/etc.).

The recommended approach for this category of large data is to send HRI a FHIR resource that embeds a reference to the large data (such as binary or large text) that has been passed to the platform in another location such as to a COS bucket. Examples of this include the ImagingStudy FHIR resource as well as the DocumentReference FHIR resource. In this scenario the JSON FHIR sent to HRI represents a complete FHIR resource but that resource alone does not provide the complete picture of the data. The ‘pointer’ within the FHIR resource must be leveraged to obtain the actual binary or large text data that resides in some external storage system (i.e. the binary or large text data is not passed within the FHIR resource itself). Specific guidance is provided below in the ‘Category 2 messages’ section.

**Category 1 messages:**

The recommended approach for category 1 messages is to have the data producer split apart the large message and have the data consumer/data pipeline ‘stitch’ the message back together if needed. Prior to splitting messages apart, it is strongly recommended that data producers first compress messages using native Kafka Zstandard (zstd) compression. If messages are still too large post compression, they will need to be split apart by the data producer.

To support handling of split messages, metadata will be required in the Kafka message header as well as, in some cases, the FHIR resources themselves (Kafka payload).

* Two metadata attributes will need to be included with each Kafka message header to support handling of split messages, ‘message-number’ and ‘last-message-indicator’.

When Kafka messages are split into multiple messages, the two Kafka message header metadata elements will be required to know how many split messages were created and when the last one has been produced.

* Two metadata attributes have been added to each FHIR resource to further support very granular split resource use cases: ‘split-resource-number’ and ‘last-split-resource-indicator’.

split-resource-number:

<Resource>.meta.extension.value (set <Resource>.meta.extension.url="<http://ibm.com/fhir/cdm/StructureDefinition/split-resource-number>")

last-split-resource-indicator:

<Resource>.meta.extension.value (set <Resource>.meta.extension.url="<http://ibm.com/fhir/cdm/StructureDefinition/last-split-resource-indicator>")

When an individual FHIR resource must be split apart, the ‘split-resource-number’ contains a sequence number of each split FHIR resource. The ‘last-split-resource-indicator’ identifies when the last message of the split records has been produced. The ‘last-split-resource-indicator’ should be set to ‘true’ on the split message which has the largest/last sequence number. For resources that are not split, these two metadata attributes would not be expected to be populated.

The following guidance is suggested when data producers encounter a FHIR bundle that, once compressed, is > 1 MB or, if sending a single FHIR resource, that single FHIR resource once compressed, is > 1 MB. *Note: the 1 MB size limitation applies to the combined size of the Kakfa header and payload.*

1. In the case of FHIR bundles, attempt to keep resources intact rather than splitting apart an individual resource.

Example 1: Assume your message includes a bundle resource with a patient resource, insurance plan resource and a coverage resource. The initial Kafka message should be split into 4 individual messages:

1. a message containing the bundle resource
2. a message containing the patient resource
3. a message containing the insurance plan resource
4. a message containing the coverage resource.

1a. When splitting a FHIR bundle ‘cleanly’ at a resource level as depicted above, **populate metadata in the Kafka message header** level to track messages that have been split.

Using Example 1 from above, populate each of the Kafka message headers with information such that this bundle could be reconstructed:

Kafka message 1 header (*Kafka header for bundle resource message*)

* Set message-number = 1
* Set last-message-indicator = false

Kafka message 2 header (*Kafka header for patient resource message*)

* Set message-number = 2
* Set last-message-indicator = false

Kafka message 3 header (*Kakfa header for insurance plan resource message*)

* Set message-number = 3
* Set last-message-indicator = false

Kafka message 4 header (*Kafka header for coverage resource message*)

* Set message-number = 4
* Set last-message-indicator = true\*

*\*indicates that this is the last Kafka message of the split messages*

1b. When splitting a FHIR resource bundle into a series of individual FHIR resources (i.e. not splitting within a single resource), the post-split bundle FHIR resource itself will still need to contain a subset of information about each of the resources within the bundle. Specifically, base resource content (id, meta, implicitRules, language) from each resource within the full FHIR bundle resource will need to be included in the post-split FHIR bundle resource.

The code snippet below is taken from a full non-split FHIR bundle. If we were to split the bundle, we would expect the post-split FHIR bundle resource to include the information highlighted in blue below. In other words, the post-split FHIR bundle resource must carry both information for the bundle resource as well as general/meta information about each additional resource within the bundle (i.e. the ‘claim’ resource in the example below). However, the post-split FHIR bundle will no longer carry the full set of data for each resource as it did prior to the split.

{

"resourceType": "Bundle",

"id": "5415b7e8-1fb5-47d9-8b77-8303718ec42b",

"meta": {

"lastUpdated": "2020-06-01T16:43:05.973+00:00"

},

"type": "searchset",

"link": [ {

"relation": "self",

"url": "http://hapi.fhir.org/baseR4/Claim?\_count=3&\_pretty=true"

}, {

"relation": "next",

"url": "http://hapi.fhir.org/baseR4?\_getpages=5415b7e8-1fb5-47d9-8b77-8303718ec42b&\_getpagesoffset=3&\_count=3&\_pretty=true&\_bundletype=searchset"

} ],

"entry": [ {

"fullUrl": "http://hapi.fhir.org/baseR4/Claim/5554b45b-668a-4f5c-b290-d4b5f5ddfc27",

"resource": {

"resourceType": "Claim",

"id": "5554b45b-668a-4f5c-b290-d4b5f5ddfc27",

"meta": {

"versionId": "1",

"lastUpdated": "2020-03-24T19:42:52.014+00:00",

"source": "#anRjJex8JdtnGhIV",

"tag": [ {

"system": "https://smarthealthit.org/tags",

"code": "Covid19 synthetic population from Synthea"

} ]

},

"status": "active",

"type": {

"coding": [ {

"system": "http://terminology.hl7.org/CodeSystem/claim-type",

"code": "institutional"

} ]

},

"use": "claim",

"patient": {

"reference": "Patient/003b89e6-c7df-459a-83db-3a28db042c71",

"display": "Deon400 Lebsack687"

},

"billablePeriod": {

"start": "2014-11-05T05:00:20-06:00",

"end": "2014-11-05T05:15:20-06:00"

},

"created": "2014-11-05T05:15:20-06:00",

"provider": {

"reference": "Organization/7a208ea5-66f6-392a-8472-059d61de2410",

"display": "MIDWEST ANESTHESIA PARTNERS LLC"

},

"priority": {

"coding": [ {

"system": "http://terminology.hl7.org/CodeSystem/processpriority",

"code": "normal"

} ]

},

"facility": {

"reference": "Location/21bd6648-ed47-44ec-98a3-da7b6d47eab7",

"display": "MIDWEST ANESTHESIA PARTNERS LLC"

},

"supportingInfo": [ {

"sequence": 1,

"category": {

"coding": [ {

"system": "http://terminology.hl7.org/CodeSystem/claiminformationcategory",

"code": "info"

} ]

},

"valueReference": {

"reference": "Immunization/e810c578-db45-448a-b30c-852c8597a5b4"

}

} ],

"insurance": [ {

"sequence": 1,

"focal": true,

"coverage": {

"display": "UnitedHealthcare"

}

} ],

"item": [ {

"sequence": 1,

"productOrService": {

"coding": [ {

"system": "http://snomed.info/sct",

"code": "162673000",

"display": "General examination of patient (procedure)"

} ],

"text": "General examination of patient (procedure)"

},

"encounter": [ {

"reference": "Encounter/e5762d2a-03ce-4e2d-908a-1c58bd9776c9"

} ]

}, {

"sequence": 2,

"informationSequence": [ 1 ],

"productOrService": {

"coding": [ {

"system": "http://hl7.org/fhir/sid/cvx",

"code": "140",

"display": "Influenza, seasonal, injectable, preservative free"

} ],

"text": "Influenza, seasonal, injectable, preservative free"

},

"net": {

"value": 123.88,

"currency": "USD"

}

} ],

"total": {

"value": 113.86,

"currency": "USD"

}

},

"search": {

"mode": "match"

}

},

.

.

.

.

.

1c. When splitting a FHIR resource bundle apart into individual FHIR resources (i.e. not splitting within a single resource), the post-split individual resources will contain the base resource content (id, meta, implicitRules, language) as well as the full set of data. The ‘id’ from the base resource content is the element that allows the post-split individual FHIR resources to be connected back to the post-split FHIR resource bundle, i.e. allows the bundle to be reassembled.

The code snippet below is taken from a full FHIR bundle resource. If we were to split the bundle, we would expect the post-split claim FHIR resource to include the information highlighted in blue below. In other words, the post-split claim resource carries the base resource content as well as the data itself. The ‘id’ within the base resource content for post-split claim FHIR resource would match to an ‘id’ of one of the resources within the ‘entry’ array of the post-split apart FHIR bundle resource.

{

"resourceType": "Bundle",

"id": "5415b7e8-1fb5-47d9-8b77-8303718ec42b",

"meta": {

"lastUpdated": "2020-06-01T16:43:05.973+00:00"

},

"type": "searchset",

"link": [ {

"relation": "self",

"url": "http://hapi.fhir.org/baseR4/Claim?\_count=3&\_pretty=true"

}, {

"relation": "next",

"url": "http://hapi.fhir.org/baseR4?\_getpages=5415b7e8-1fb5-47d9-8b77-8303718ec42b&\_getpagesoffset=3&\_count=3&\_pretty=true&\_bundletype=searchset"

} ],

"entry": [ {

"fullUrl": "http://hapi.fhir.org/baseR4/Claim/5554b45b-668a-4f5c-b290-d4b5f5ddfc27",

"resource": {

"resourceType": "Claim",

"id": "5554b45b-668a-4f5c-b290-d4b5f5ddfc27",

"meta": {

"versionId": "1",

"lastUpdated": "2020-03-24T19:42:52.014+00:00",

"source": "#anRjJex8JdtnGhIV",

"tag": [ {

"system": "https://smarthealthit.org/tags",

"code": "Covid19 synthetic population from Synthea"

} ]

},

"status": "active",

"type": {

"coding": [ {

"system": "http://terminology.hl7.org/CodeSystem/claim-type",

"code": "institutional"

} ]

},

"use": "claim",

"patient": {

"reference": "Patient/003b89e6-c7df-459a-83db-3a28db042c71",

"display": "Deon400 Lebsack687"

},

"billablePeriod": {

"start": "2014-11-05T05:00:20-06:00",

"end": "2014-11-05T05:15:20-06:00"

},

"created": "2014-11-05T05:15:20-06:00",

"provider": {

"reference": "Organization/7a208ea5-66f6-392a-8472-059d61de2410",

"display": "MIDWEST ANESTHESIA PARTNERS LLC"

},

"priority": {

"coding": [ {

"system": "http://terminology.hl7.org/CodeSystem/processpriority",

"code": "normal"

} ]

},

"facility": {

"reference": "Location/21bd6648-ed47-44ec-98a3-da7b6d47eab7",

"display": "MIDWEST ANESTHESIA PARTNERS LLC"

},

"supportingInfo": [ {

"sequence": 1,

"category": {

"coding": [ {

"system": "http://terminology.hl7.org/CodeSystem/claiminformationcategory",

"code": "info"

} ]

},

"valueReference": {

"reference": "Immunization/e810c578-db45-448a-b30c-852c8597a5b4"

}

} ],

"insurance": [ {

"sequence": 1,

"focal": true,

"coverage": {

"display": "UnitedHealthcare"

}

} ],

"item": [ {

"sequence": 1,

"productOrService": {

"coding": [ {

"system": "http://snomed.info/sct",

"code": "162673000",

"display": "General examination of patient (procedure)"

} ],

"text": "General examination of patient (procedure)"

},

"encounter": [ {

"reference": "Encounter/e5762d2a-03ce-4e2d-908a-1c58bd9776c9"

} ]

}, {

"sequence": 2,

"informationSequence": [ 1 ],

"productOrService": {

"coding": [ {

"system": "http://hl7.org/fhir/sid/cvx",

"code": "140",

"display": "Influenza, seasonal, injectable, preservative free"

} ],

"text": "Influenza, seasonal, injectable, preservative free"

},

"net": {

"value": 123.88,

"currency": "USD"

}

} ],

"total": {

"value": 113.86,

"currency": "USD"

}

},

"search": {

"mode": "match"

}

},

.

.

.

.

.

.

*Note: once a FHIR bundle has been split apart, it will be unable to be validated against a FHIR bundle profile.*

1. There may be times where you are unable to keep an individual FHIR resource fully intact and will be forced to split within an individual FHIR resource; this may occur for a FHIR resource either within, or independent of, a bundle.

Example 2: Assume your message includes a bundle resource with a patient resource, an insurance plan resource and a coverage resource, and that you’ve already split the single message into 4 individual messages (bundle resource, patient resource, insurance plan resource and coverage resource). Further assume the coverage resource, once split and compressed, is still > 1 MB. In this case the coverage resource will need to be split into multiple resources. Let us assume in this case that the coverage resource is split into 3 messages. The initial Kafka message should be split into 6 individual messages:

1. a message containing the bundle resource
2. a message containing the patient resource
3. a message containing the insurance plan resource
4. a message containing a partial coverage resource (1 of 3).
5. a message containing a partial coverage resource (2 of 3).
6. a message containing a partial coverage resource (3 of 3).

2a. **Set metadata within the Kafka header and the FHIR resources** to track messages that have been split across and within resources.

Using Example 2 from above, metadata is being kept at Kafka message header level, tracking the total number of messages in the bundle AND metadata is also being kept at the FHIR resource level (for the split coverage resource), tracking the total number of messages for that resource.

Kafka message 1 header (bundle resource)

* Set message-number = 1
* Set last-message-indicator = false

Kafka message 2 header (patient resource)

* Set message-number = 2
* Set last-message-indicator = false

Kafka message 3 header (insurance plan resource)

* Set message-number = 3
* Set last-message-indicator = false

Kafka message 4 header (coverage resource [1])

* Set message-number = 4
* Set last-message-indicator = false

Kafka message 4 payload for FHIR coverage resource:

* Set FHIR resource metadata: split-resource-number = 1
* Set FHIR resource metadata: last-split-resource-indicator = false

Kafka message 5 header (coverage resource [2])

* Set message-number = 5
* Set last-message-indicator = false

Kafka message 5 payload for FHIR coverage resource:

* Set FHIR resource metadata: split-resource-number = 2
* Set FHIR resource metadata: last-split-resource-indicator = false

Kafka message 6 header (coverage resource [3])

* Set message-number = 6
* Set last-message-indicator = true\*

*\*indicates that this is the last Kafka message of the split messages*

Kafka message 6 payload for FHIR coverage resource:

* + - * Set FHIR resource metadata: split-resource-number = 3
      * Set FHIR resource metadata: last-split-resource-indicator = true\*\*

*\*\*indicates the this is the last FHIR coverage resource instance*

2b. The code below represents a full FHIR patient resource. If we were to split the resource into 2 FHIR patient resources, we would expect each post-split FHIR patient resource to include the type of information highlighted in blue below (base resource content) as well as any additional data elements specifically required by the resource. Within the post-split resources we would also expect the additional FHIR resource meta elements related to splitting records, split-resource-number and last-split-resource-indicator, to be set as described above (note: these 2 meta elements are not depicted in the example resource below).

Each post-split FHIR patient resource will carry duplicative ‘meta’ information (base resource content). Within the base resource content, the value in the ‘id’ field must match across each of the post-split FHIR patient resources, allowing for resources to be stitched back together. At this point it is expected that no individual post-split FHIR resource will carry the full set of patient data as the data has now been split across multiple resources.

{

"resourceType": "Patient",

"id": "b100037f-b42a-4e5f-90b5-fc0bf5612fa5",

"meta": {

"versionId": "1",

"lastUpdated": "2020-03-24T23:07:53.397+00:00",

"source": "#CnxzKeo9xCsEdgw8",

"profile": [ "http://hl7.org/fhir/us/core/StructureDefinition/us-core-patient" ],

"tag": [ {

"system": "https://smarthealthit.org/tags",

"code": "Covid19 synthetic population from Synthea"

} ]

},

"text": {

"status": "generated",

"div": "<div xmlns=\"http://www.w3.org/1999/xhtml\">Generated by <a href=\"https://github.com/synthetichealth/synthea\">Synthea</a>.Version identifier: 7ff8db58\n . Person seed: -1720285496239491927 Population seed: 12345</div>"

},

"extension": [ {

"url": "http://hl7.org/fhir/us/core/StructureDefinition/us-core-race",

"extension": [ {

"url": "ombCategory",

"valueCoding": {

"system": "urn:oid:2.16.840.1.113883.6.238",

"code": "2028-9",

"display": "Asian"

}

}, {

"url": "text",

"valueString": "Asian"

} ]

}, {

"url": "http://hl7.org/fhir/us/core/StructureDefinition/us-core-ethnicity",

"extension": [ {

"url": "ombCategory",

"valueCoding": {

"system": "urn:oid:2.16.840.1.113883.6.238",

"code": "2186-5",

"display": "Not Hispanic or Latino"

}

}, {

"url": "text",

"valueString": "Not Hispanic or Latino"

} ]

}, {

"url": "http://hl7.org/fhir/StructureDefinition/patient-mothersMaidenName",

"valueString": "Sean831 Durgan499"

}, {

"url": "http://hl7.org/fhir/us/core/StructureDefinition/us-core-birthsex",

"valueCode": "M"

}, {

"url": "http://hl7.org/fhir/StructureDefinition/patient-birthPlace",

"valueAddress": {

"city": "Alhambra",

"state": "California",

"country": "US"

}

}, {

"url": "http://synthetichealth.github.io/synthea/disability-adjusted-life-years",

"valueDecimal": 0

}, {

"url": "http://synthetichealth.github.io/synthea/quality-adjusted-life-years",

"valueDecimal": 32

} ],

"identifier": [ {

"system": "https://github.com/synthetichealth/synthea",

"value": "fd16075a-ef85-4896-836a-5000fb88a682"

}, {

"type": {

"coding": [ {

"system": "http://terminology.hl7.org/CodeSystem/v2-0203",

"code": "MR",

"display": "Medical Record Number"

} ],

"text": "Medical Record Number"

},

"system": "http://hospital.smarthealthit.org",

"value": "fd16075a-ef85-4896-836a-5000fb88a682"

}, {

"type": {

"coding": [ {

"system": "http://terminology.hl7.org/CodeSystem/v2-0203",

"code": "SS",

"display": "Social Security Number"

} ],

"text": "Social Security Number"

},

"system": "http://hl7.org/fhir/sid/us-ssn",

"value": "999-65-8560"

}, {

"type": {

"coding": [ {

"system": "http://terminology.hl7.org/CodeSystem/v2-0203",

"code": "DL",

"display": "Driver's License"

} ],

"text": "Driver's License"

},

"system": "urn:oid:2.16.840.1.113883.4.3.25",

"value": "S99969793"

}, {

"type": {

"coding": [ {

"system": "http://terminology.hl7.org/CodeSystem/v2-0203",

"code": "PPN",

"display": "Passport Number"

} ],

"text": "Passport Number"

},

"system": "http://standardhealthrecord.org/fhir/StructureDefinition/passportNumber",

"value": "X89326739X"

} ],

"name": [ {

"use": "official",

"family": "Bergnaum523",

"given": [ "Maria750" ],

"prefix": [ "Mr." ]

} ],

"telecom": [ {

"system": "phone",

"value": "555-756-2656",

"use": "home"

} ],

"gender": "male",

"birthDate": "1987-06-09",

"address": [ {

"extension": [ {

"url": "http://hl7.org/fhir/StructureDefinition/geolocation",

"extension": [ {

"url": "latitude",

"valueDecimal": 34.1843383660385

}, {

"url": "longitude",

"valueDecimal": -118.55511589180004

} ]

} ],

"line": [ "126 Sanford Walk" ],

"city": "Los Angeles",

"state": "CA",

"country": "US"

} ],

"maritalStatus": {

"coding": [ {

"system": "http://terminology.hl7.org/CodeSystem/v3-MaritalStatus",

"code": "M",

"display": "M"

} ],

"text": "M"

},

"multipleBirthBoolean": false,

"communication": [ {

"language": {

"coding": [ {

"system": "urn:ietf:bcp:47",

"code": "en-US",

"display": "English"

} ],

"text": "English"

}

} ]

}

1. Regardless of whether a bundle is split into multiple individual resources or an individual resource is split into multiple resources (or both), the following guidance applies:
   1. Each resource must be valid FHIR such that they meet the JSON schema specification (aka: required fields must be provided on all instances of the same resource in the split,  properly structured FHIR JSON adhering to the profile definition, etc.).
   2. In addition to supporting metadata, each split record must minimally contain key identifying data elements such that records can be stitched back together if desired.
   3. When providing HRI with a count of records, the count should include the split records (i.e. if a single source record was split into 5 separate messages, the count provided to HRI should be '5').
   4. Each split message must be sent on the same Kafka partition.
   5. It is not required that split messages be sent in sequence nor must they be sent one after another. However, it is required that the last message of split messages must be sent last.
   6. The data producer and data consumer must be aligned on the handling and tracking of split messages.

**Category 2 messages:**

The recommended approach for category 2 messages is to have the data producer store the actual payload in an external data store and only transfer a pointer to that data via HRI. The consumer reading from HRI would need the ability to recognize this type of ‘pointer payload’ and would then need to read the data from the external store.

In this scenario, data producers would need the ability to write to the external data store while the data consumers would need the ability to read from the external data store. Additionally, the data producer would need to account for any messages that failed HRI as this may result in the need for the data producer to remove/rollback data from the external data store. Finally, the data producer and data consumer(s) would need to define and implement an approach for managing/archiving data within the external data store.

Again, in this scenario, the messages sent to HRI should include a pointer to that data via HRI. This can be achieved by populating metadata as part of the message header to track ‘pointer payload’ messages. The data element carrying the pointer will need to be identified for each FHIR reference. For example, within the DocumentReference FHIR resource, the pointer would be stored within the DocumentReference.content.attachment element.

The example below outlines 5 messages (2 of which are ‘pointer payload’ messages) and demonstrates how the need for the consumer to recognize ‘pointer payload’ messages and source the actual payload based on the pointer metadata provided in the message.

