Cookbook for Regional Interoperability Detailed Design Paper #010

FHIR Aggregator Service

PRELIMINARY DRAFT

Version 1.0 – 7th July 2019

Abstract Interoperability Cookbook Anchor Points

Section	Title
3.1.2	FHIR Service Bus

Table of Contents

Version Control

Version	Release Date	Released By	Reason for Release
1.0	7/07/2019	R Hickingbotham	Preliminary draft

Reviewers

Initials	Name	Role	Organisation

1 Introduction

1.1 Purpose of this Document

This document is one of a series of design papers which underpin the Abstract of a Cookbook for Regional Interoperability (the Abstract Cookbook). These papers, in their totality, describe the technical components and the standards which form the YHCR System of Systems. They are intended as a basis for developing or procuring software and so are expressed at a level of precision which aims to avoid ambiguity but consequentially, they are focussed to technical readers.

Design papers are anchored to topics which are discussed in the Abstract Cookbook. They are elaborations of the concepts which were first introduced by the abstract and new content is further detail rather than variations of previously established core principles.

This document (design paper 010 - “FHIR Aggregator Service”) is a design for one of the key regional components of the system of systems - the FHIR Aggregator. This is a component which simplifies access to data. It provides a single endpoint which data consumers can target to interact with all data providers participating in the YHCR.

1.2 Functions of the FHIR Aggregator

In addition to simplifying the YHCR from the perspective of data consumers, the FHIR Aggregator plays an important role in standardising data and enforcing controls over access to data. Specifically, the functions of the FHIR aggregator are to:

- present data which is accessed from a number of different sources as through it was part of a unified data model;
- target patient-centric searches for data at those data providers who have has contact with the patient;
- enforce consent and other policy-based access rules;
- enforce the scope of data access rights which are appropriate given a data consumer’s user’s role and reason for accessing the YHCR;
- audit requests for access and the resulting release of data by data providers.

1.3 Unified Data Model

The concept of a unified data model was introduced by design paper 001 – “A Unified Data Model for FHIR”. In brief, the requirement for the aggregator is to present resources which have been sourced from a number of different locations as if they were being served from a single database. In particular, resource identifiers used locally are disambiguated so that they are unique across the YHCR, and resources which related to physical concepts, the identity of which all participants to the YHCR should agree, are deduplicated. At the outset the aggregator will deduplicate:

- Patients;
- Practitioners;
- Organisations.

As coding standards improve at data providers then the goals for de-duplication may widen to include:

- People (such as next of kin);
- Locations;
- Medications (the drug catalogue).

Usually searches for deduplicated concepts will return the regional ‘golden-record’ for a concept. Normally this record will have been built from a primary source of data outside of the YHCR but may be supplemented by a regionally managed data. There will be circumstances when a data consumer wishes to access local versions of the concept. For instance, a consumer may wish to display all addresses for a patient as known to individual providers. A search for a patient would normally return the regional record and specific searches need to be constructed to access local records. This paper offers techniques for data consumers to use to deconstruct the unified model.

1.4 Relationship of this Document with Other Standards

This paper is a statement of intent rather than a design and does not rely on any particular standard although many standards will be used in the implementation of the intent:

- FHIR;
- SNOMED-CT;
- DM+D;
- ICD10;
- LOINC;
- Read Codes.

1.5 Intended Users of the This Document

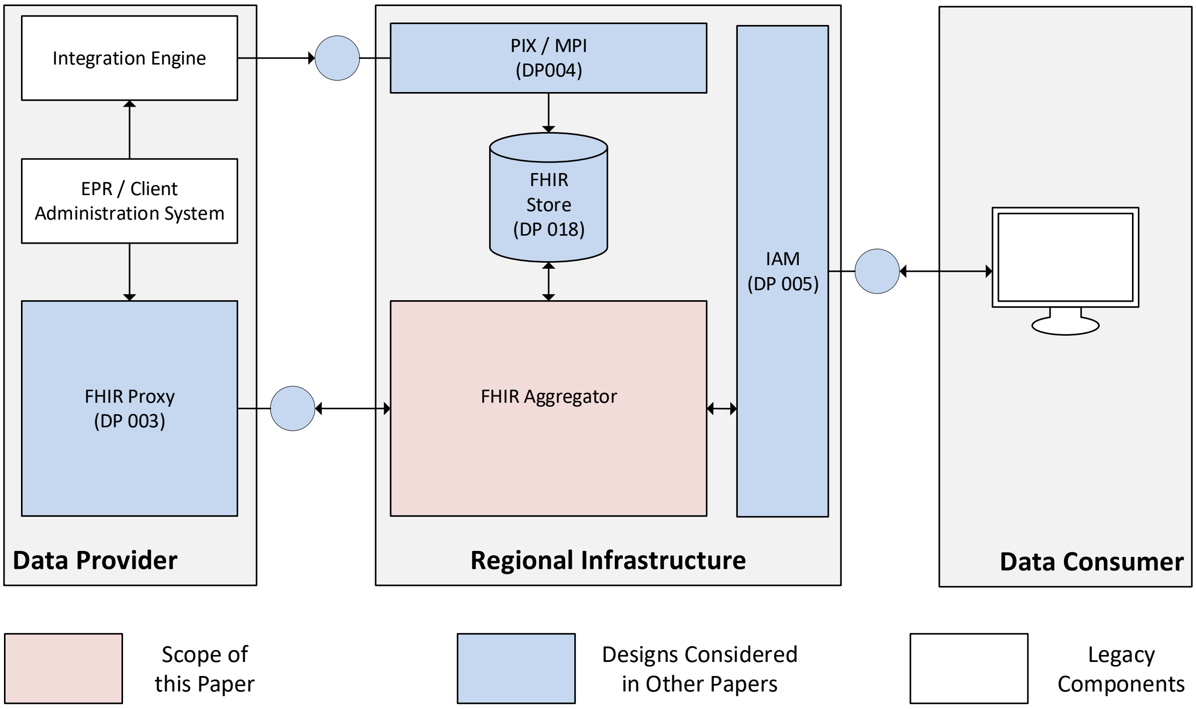
Implementors of the Regional FHIR Aggregator and Data Consumers using it to simplify access to data.

2 Functional Overview

2.1 Relationship with Other Components

The FHIR Aggregator is a regional component that interacts with the regional FHIR Store and local FHIR proxies with the objective of presenting a unified view of regional data to a data consumer.

These relationships are illustrated below.



The regional FHIR Store is used to persist:

- Regional *Patient* and *Linkage* resources which allow a patient centric search to be targeted to locations which have registered contact with a patient through the regional PIX server;
- The regional version of other deduplicated resources including *Practitioners* and *Organisations* with associated *Linkages* to local versions of these concepts;
- *Consent* resources which signify patients' sanction of a data access policy;
- *Audit* resources written by the FHIR aggregator and other regional components;
- Other clinical data.

The FHIR Aggregator is abstracted from direct interactions with data consumers by the Identity and Access Management (IAM) Server. IAM will only pass-on authorised requests to the FHIR Aggregator.

2.2 Local Identifiers, Regional Identifiers, and Resource Provenance

Every resource served from a local proxy or persisted in the regional FHIR store has a local logical identifier which is unique to the data source. The choice of the format of the identifier is left to the implementor with the only provisions that it is unique to the source, consists of case sensitive alphanumeric characters plus '–' or '.', and is 59 characters or less in length.

Resources also have a regional logical identifier which is constructed from the local identifier as follows:

```
<data source>.<local identifier>
```

where <data source> is a 4 character code which is assigned to a data provider as part of the onboarding process (design paper 020: "Onboarding Data Providers"). For instance, a QuestionnaireResponse maintained by Helm may of a local logical identifier of:

```
a07e248c-6171-44e3-9543-f24ea7e6db2e
```

When served from the FHIR aggregator it will have a regional logical identifier of:

```
HELM.a07e248c-6171-44e3-9543-f24ea7e6db2e
```

The FHIR standard advises consumers not to rely on the structure of logical identifiers and whilst the convention will be consistently applied by the FHIR Aggregator a better method of determining the provenance of a FHIR resource is to examine the meta data Source tag property of a resource:

```
"meta": {
    "tag": [
        {
           "system": "https://yhcr.nhs.uk/Source",
           "code": "HELM",
           "display": "Helm"
        }
    ]
```

A Source tag property must be included at source by all data providers.

Tags are searchable and so the search:

```
QuestionnaireResponse?_tag=https://yhcr.nhs.uk/Source|HELM
```

will return only questionnaire responses from the Helm data source. Note that queries involving the Source tag are optimised by the FHIR aggregator and in this case the search would only be presented to Helm.

The regional FHIR store has the source identifier: YHCR

2.3 Behaviour of Standard REST Operations

The FHIR Aggregator acts as a proxy to data providers by routing requests made by data consumers so that they are serviced by one or a number of data sources. The routing mechanisms are noted below:

REST Verb	Function
GET (direct retrieval)	`GET /Patient/ASRC:172364367` - the request is routed to a single data source: ASRC and the local *Patient* resource is returned.

GET
(search of regionally held, de-duplicated resource types)
	`GET /Patient?family=Brown`
	- searches the local FHIR store and returns the regional 'golden-record' *Patient* resources.
	
	The search can be targeted at specific sources by including Source tags in the search string:
	
	`GET /Patient?family=Brown&_tag=https://yhcr.nhs.uk/Source|LTH1`

<table>
<tr><td></td><td>- the search is targeted at LTH1 and local <i>Patient</i> resources are returned.</td></tr>
<tr><td>GET
(patient centric search)</td><td><code>GET /Observation?subject=Patient/YHCR:647dafde-7261-48b2-a2ad-d967a6445942</code>
- searches all data sources where the data provider has registered contact with the patient with PIX. The Patient resource reference is substituted for a local reference.

The search can be targeted to specific sources by including Source tags in the search string:

<code>GET /Observation?patient=Patient/YHCR:647dafde-7261-48b2-a2ad-d967a6445942&_tag=https://yhcr.nhs.uk/Source|LTH1</code>
- searches LTH1 for observations for the patient.

A local resource reference achieves the same effect:
<code>GET /Observation?patient=Patient/LTH1:1237229412</code>

Note that if more than one local resource reference is included in a search term or there are a mix of local resource references and Source tags specified then the intersection of the sources will be searched. So:

<code>GET /Observation?patient=(Patient/LTH1:1237229412, Patient/YAS1:9353636536)&performer=Practitioner/YAS1:y7g5</code>

Will search the YAS1 source and not the LTH1 source.</td></tr>
<tr><td>GET
(non-patient centric search)</td><td><code>GET /HealthcareService?category=http://hl7.org/fhir/service-category|7</code>
- searches all data sources

The search can be targeted to specific sources by including Source tags in the search string:

<code>GET /HealthcareService?category=http://hl7.org/fhir/service-category|7&_tag=https://yhcr.nhs.uk/Source|LTH1</code>
- searches LTH1 for community healthcare services.

Including local resource references in search strings also restricts searches to specified sources.

<code>GET /HealthcareService?category=http://hl7.org/fhir/service-category|7&organisation=LTH1:57239</code>

Note that the same rules for resolving conflicting resource specifications as specified apply.</td></tr>
<tr><td>POST</td><td><code>POST /Appointment</code>
- creates an appointment in that data source that is specified by the Source meta tag. If no source is specified, then the attempt is rejected by the aggregator.

The creation may be conditional as in the example below.

<code>POST /Appointment?patient=nhs|1234567890&date= 2013-01-14T00:00</code></td></tr>
</table>

	If the condition includes local resource references and these must be within the same data source as specified by the Source meta tag.
PUT or PATCH (direct resource reference)	`PUT /Appointment/YHCR:638391222` - updates an appointment at the data source implied by the regional logical identifier. If the resource includes a Source meta tag, then this must be consistent with the source implied by the resource identifier

PUT or PATCH
(conditional)
	The data sources to update are determined in the same manner as searches. I.e.:
	i) updates to regionally held resource types (in the absence of a data source specifier) are made against the regional FHIR Store.
	`PATCH /Organisation?active=true`
	ii) the update maybe targeted to non-regional data sources by specifying a Source meta tag, including a local reference in a search term or specifying the Source meta tag in the resource content.
	`PATCH /Organisation?active=true&_tag=https://yhcr.nhs.uk/Source|LTH1`
	ii) Patient centric conditions cause the update to be targeted at all data providers who have registered contact with the patient.
	`PUT /Appointment?patient=nhs|1234567890&date= 2013-01-14T00:00`
	iii) Patient centric conditions may be targeted at a specific data sources by specifying a Source meta tag, including a local reference in a search term or specifying the Source meta tag in the resource content.
	`PUT /Appointment?subject=nhs|1234567890&date= 2013-01-14T00:00&incomingreferal=ReferalRequest/YAS1:8363922`
	iv) Non-patient centric conditional updates against non-regionally held de-duplicated resource types are issued to all data sources unless they are explicitly targeted by specifying a Source meta tag, including a local reference in a search term or specifying the Source meta tag in the resource content.

Note that the ability to create or modify resources will be severely constrained and will be specified in the CabailityStatement offered by each data source. The aggregator will optimise POST, PUT and PATCH requests by only issuing them to those data sources which support the operation.

2.3.1 Patient Centric Searches

Design paper 005 – "Identity and Access Management", identifies a set of resource types which relate to or may relate to a patient. In the above rules a search is patient centric if it includes a term which specifies the patient. The term must be a direct reference to the patient in one of the following formats:

- a relative Patient reference in the form Patient/<data source>.<local identifier>;
- an absolute reference in the form https://<server>/Patient/<data source>.<local identifier>;

- an NHS number in the form nhs|<number>.

A search that uses a chain term into the properties of the subject patient eg:

```
GET /Observation?subject.family=Brown
```

is NOT patient centric.

Patient centricity is important for targeting queries to data sources and for enforcing the scope of data accessible for various reasons for accessing the YHCR.

2.3.2 Regionally Held De-Duplicated Resource Types

The following are regionally held de-duplicated resource types:

- Patient;
- Organisation;
- Practitioner;
- Linkage.

Unless a data source is specified as detailed in the above rules, then a search against these data types will return results from the regional FHIR store.

2.4 Querying Regional Linkage Resources

Linkage resources are maintained in the regional FHIR store to link regional 'golden-record' data to local resources representing the same concept. Querying the regional resource and 'including' referenced resources is a useful technique for a data consumer to obtain local equivalents of regional resources.

For instance, if a consumer wishes to display all addresses known to all data providers for a Patient then the following query will return all local *Patient* resources.

```
GET /Linkage?source=/Patient/YHCR:35567d84-5653-4bd3-953e-
45b6ad54f5d6&_include=Linkage:item
```

Patient linkages are built when a data provider informs the YHCR about contact with a patient and so can be expected to be complete. Linkages for other resource types are built when the YHCR encounters an instance of a resource at a locality and so are only complete in respect of data of which the YHCR is aware.

2.5 Enforcement of Role and Reason Based Access Rights

The reason for access and the regional role that a user at a data consumer performs is established when the data consumer makes a claim to the regional Identity and Access Management server (design paper 005). From this point on the scope of data which can be accessed is controlled by the FHIR Aggregator.

The rules are detailed in design paper 005 but in summary these distil to:

- if access is in the context of a patient (as is necessary for reasons of direct care) then patient identifiable data can only be accessed for the patient in context;
- if access is for general administrative purposes then only data in the Regional FHIR Store is available

- if the role of the accessor is an auditor then only *AuditEvents* are accessible,

2.6 Enforcement of Consent and Policy-Based Access Rights

Design paper 008 – “Data Access and Consent Management” establishes mechanisms for describing a policy for allowing access to data and for enabling a citizen to opt in or out of the policy.

The FHIR Aggregator is responsible for enforcing data access policies.

Policy rules are expressed as data points which are covered by the policy and the context in which the policy applies. Policies are tied to a context. Context is established by the organisation accessing the YHCR, the role of the use and their reason for access. Context is derivable from the claim made to the Identity an Access Management service and is available to the FHIR Aggregator when determining whether a policy applies.

Data points covered by a policy are expressed as FHIR search terms.

The FHIR Aggregator tests each FHIR resource before it releases it for compliance with applicable policies. It employs a strategy of augmenting FHIR searches so that they include all data needed to test a policy so avoiding unnecessary exchanges with a data provider.

2.7 Pagination

Data consumers can request a paginated result set from the FHIR Aggregator. This is an efficient mechanism for consumers limiting the amount of data which is shown to their end-users and reduces load on data providers.

The FHIR Aggregator mediates pagination requests by maintaining a cache of results. Requests for a page of results might be served from the Aggregator’s cache or might require a request for further results from one or more data providers which are supplying the cache.

2.8 Asynchronous Queries

An asynchronous query is one where a data consumer requests data from the FHIR Aggregator, but the results are assembled over time and are obtained by the consumer via subsequent interactions.

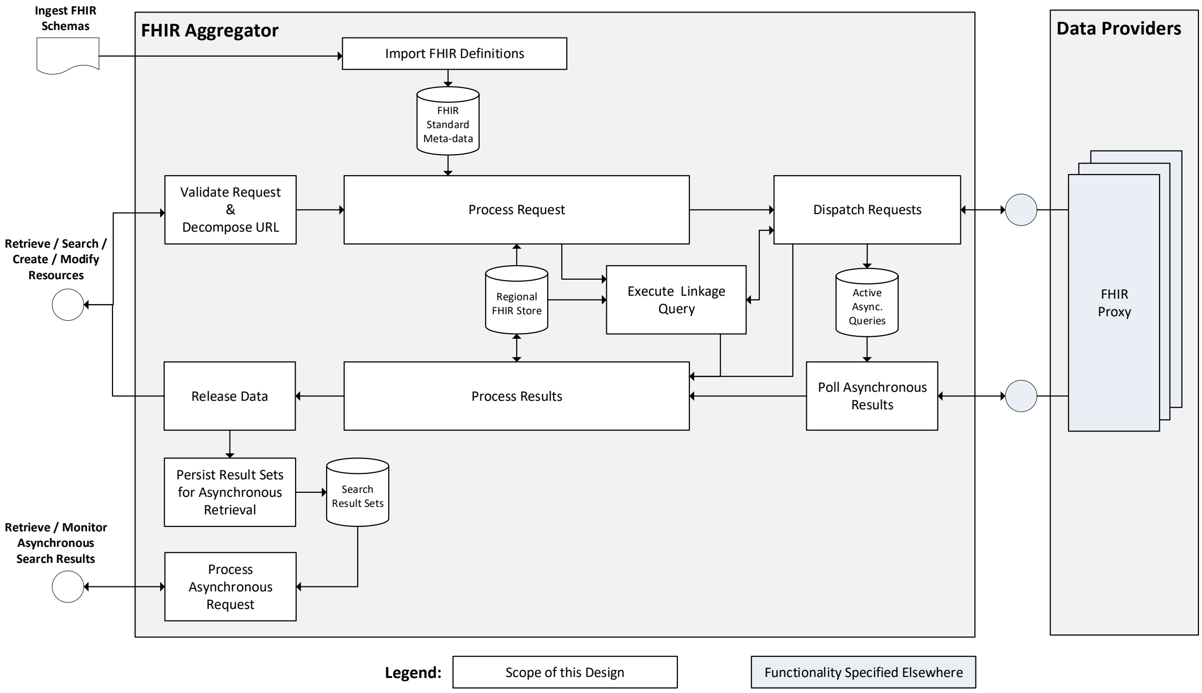
Data providers, through their FHIR proxy implementations (design paper – 003), also support asynchronous queries.

The FHIR Aggregator mediates with data providers by accepting an asynchronous query and offsetting it onto data providers, The FHIR Aggregator collects results, asynchronously, and, when all data providers have supplied results, makes them available to the data consumer.

3 Processing Model

The following sections detail the processing model for the FHIR Aggregator. The model defines message pathways which are a sequence of processing steps which involved in servicing an inbound FHIR request.

The high-level model is as illustrated below:



The FHIR Aggregator offers 3 inbound services:

1. A service for ingesting FHIR Schemas: an objective of the design is for the aggregator to be indifferent to the version of FHIR which it processes. Whilst much of the processing is specific the structure of FHIR resources, this detail can, in the main, be derived from schemas and code tables. This service loads these details from files.

2. A RESTful FHIR service which supports the standard FHIR operations. This service distributes the operation to appropriate data providers and aggregates results.

3. A RESTful service which supports asynchronous FHIR operations. This service allows an invoker to monitor progress of an asynchronous query and to retrieve the search results.

The REST services run over HTTPS and are secured in accordance with the public key infrastructure described by design paper 016 – “Securing the YHCR”.

The FHIR Schema ingestion service can only be accessed by appropriately privileged members of the YHCR support organisation.

The functionality of the FHIR aggregator is implemented using a componentised or micro-service architecture. Components are chained together in a message pathway. Components communicate with each other by sending messages which are queued at the boundary of the component.

The processing of the message pathways is explained below.

3.1 Import FHIR Definitions

This component reads FHIR schemas and other metadata files and populates internal persistent data structures. The metadata, which reflects Y&H FHIR profiles, will be published in JSON format by the Data Architecture Design Authority (DADA). These documents are profiled versions of those published by Care Connect, where available, or otherwise HL7 itself.

Schema files describe the properties of resources, data structures used by resources, value sets, and coding systems.

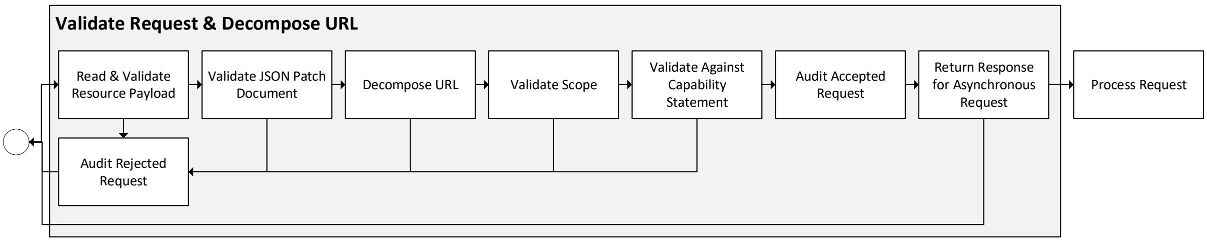
The FHIR Aggregator uses resource schemas to validate the content of resources created or modified by the FHIR Proxy. The aggregator supports multiple schema versions or profiles. Importing schemas for one profile should not impact the definition of another.

Schemas can be imported in part (i.e.: individual resource definitions, coding tables etc.) or in whole. Configuration data is linked to resource definitions. Importing a schema should preserve all configuration settings which are existent for the resources and profiles covered by the schema.

Importing a schema should not require system downtime. The import process may need to suspend dependant processes during key stages of the import, but the message architecture should allow those processes to be suspended without impacting availability of the service as a whole.

3.2 Validate Request & Decompose URL

Performs sanity checks before progressing with further processing. a validation failure will cause the synchronous HTTPS connection to be terminated with an HTTP error.


Validate Request & Decompose URL
Read & Validate Resource Payload
Validate JSON Patch Document
Decompose URL
Validate Scope
Validate Against Capability Statement
Audit Accepted Request
Return Response for Asynchronous Request
Process Request
Audit Rejected Request

3.2.1 Read and Validate Resource Payload

For PUT, POST requests the HTTP body must contain a FHIR resource. Resource meta data identifies the resource type and FHIR profile on which it is based. The request will be rejected if the profile is not known to the FHIR Aggregator.

Optionally, validation will check that only approved coding systems are being used and/or that a code is valid for the coding system. Approved coding systems and the scope of validation performed are configured for individual properties, by resource and resource profile.

Note that resources being created through the aggregator must identify the location of the resource in the meta tag “Source”.

3.2.2 Validate JSON Patch Document

For PATCH requests, the HTTP body must contain a JSON patch document: an array of instructions composed of an operation path and value.

Validation ensures that paths exist in the resource schema definitions, values comply with format constraints and value sets, and that remove operations are not operating on mandatory parameters.

Optionally, validation will check that only approved coding systems are being used and/or that a code is valid for the coding system.

3.2.3 Decompose URL

Extracts the resource path, search string, search modifiers and directives from the URL and populate an internal data structure.

3.2.4 Validate Scope

Ensures that the request is appropriate given the reason for accessing the YHCR. Section 2.5 details three circumstances that can be controlled by the aggregator:

1) Access to the YHCR is in the context of the patient and all patient identifiable data which is exchanged through the aggregator must relate to that patient;
2) Access is for general administrative purposes and any data in the regional FHIR store can be targeted other than *AuditEvents*;
3) Access is being made by an auditor and only *AuditEvents* can be read.

This component also ensures self-consistency in requests which target specific endpoints.

Access in the Context of a Patient

Design Paper 005 – “Identity and Access Management” identifies those resource types which are patient identifiable and those which are potentially patient identifiable. If an operation is for a resource type that is patient identifiable, and the claim made by the user identified the patient in context (which is mandatory if the reason for access is direct care) then this component validates that:

- a search for resources explicitly references the patient in context as the subject of the resources being searched;
- a resource being created has the patient in context as the subject of the resource;
- a resource being updated is conditional (as per the search term) on the subject of the resource being the patient in context.

For each patient identifiable resource type there are one or more search terms or parameters which identify the subject of the resource. These are managed through configuration of the FHIR Aggregator.

General Administrative Access

Resources can only be created in the regional FHIR store. The source meta tag of the resource must be the identifier of the regional FHIR Store.

Only resources in the regional FHIR Store can be updated or patched. The resource identifier must be prefixed with the identifier of the regional FHIR Store.

Only resources in the regional FHIR Store can be retrieved. The resource identifier must be prefixed with the identifier of the regional FHIR Store.

Searches and conditional updates must be targeted at the regional FHIR Store. This is ensured by the processing at 3.3.3 - “Determine Target Endpoints”.

Auditor Access

Only GET operations on *AuditEvents* are permitted.

Care must be taken to ensure that use of the _include directive does not open up access to patient data. Specifically, an attempt to _include an *entity* must be rejected.

Self-Consistency

A conditional POST which uses resource references or the Source meta tag in the condition must be contained within the source specified by the resource's Source meta tag.

The resource reference in a PUT must be at the same source as that specified by the resource's Source meta tag.

A conditional PUT or PATCH which uses resource references or the Source meta tag in the condition must be contained within the same source and the source specified by the resource's Source meta tag if available.

3.2.5 Validate Against Capability Statement

Validates that the request is within the capability of the FHIR Aggregator as defined by its *CapabilityStatement*

The FHIR Aggregators capability is equivalent to the union of all capabilities of data providers registered with the YHCR. It is computed as part of the onboarding process for data providers (design paper 020).

3.2.6 Audit Accepted Request

An *AuditEvent* resource is written to the regional FHIR store.

3.2.7 Return Response for Asynchronous Request

If the request indicated that an asynchronous response is preferred, then:

i. generate a unique identifier for the request.
ii. respond immediately with a 202 HTTP response code and content location constructed from the URL of the asynchronous processing endpoint and the unique request identifier.
iii. continue with the Process Request pathway as for synchronous requests.

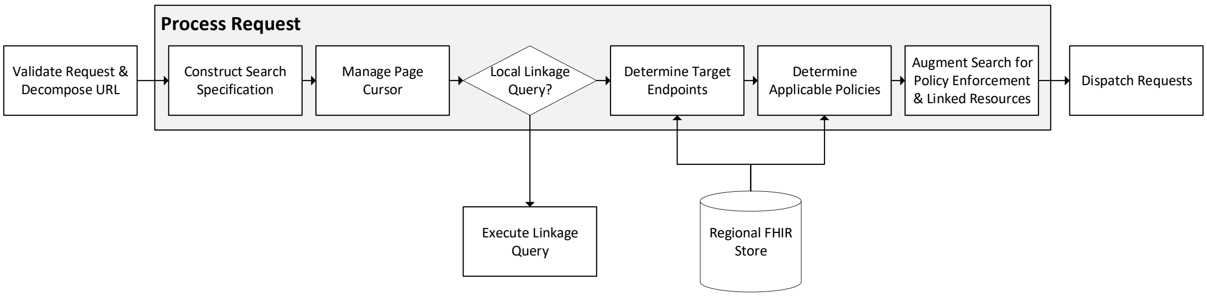
3.2.8 Audit Rejected Request

If any above validation processes cause the request to be rejected then an *AuditEvent* resource is written to the regional FHIR store.

The *outcome* and *outcomeDesc* properties identify the reason for rejection.

3.3 Process Request

Manages a pagination cache, determines the endpoints on which the request is to be placed, and augments the request to ensure that all data needed to process the response is available.


Process Request
Validate Request & Decompose URL
Construct Search Specification
Manage Page Cursor
Local Linkage Query?
Determine Target Endpoints
Determine Applicable Policies
Augment Search for Policy Enforcement & Linked Resources
Dispatch Requests
Execute Linkage Query
Regional FHIR Store

3.3.1 Manage Page Cursor

Searches which include the _count directive receive results as discrete pages. Search results for a page are returned with URL's which enable the consumer to easily move through pages. As explained in 2.7, the pagination strategy for the FHIR Aggregator limits load on data providers by only requesting data from the providers where it essential to be able to serve the current paginated search request. I.e.: the FHIR Aggregator will not 'look ahead' for data in anticipation of the next page being requested.

The internal pagination structure identifies:

- the consumer for whom the cache is being maintained;
- the query search string;
- the data providers which are servicing the query and for each provider:
  - the number of resources returned to the consumer to date;
  - the number of resources cached at the FHIR aggregator which have yet to be returned to the consumer;
  - an indicator as to whether the query has been exhausted at the provider.
- a list of resources contained within pages which have been returned to the data consumer (backward cache);
- a list of resources (the correct sort order) which have yet to be returned to the consumer (forward cache);
- resources obtained from data providers which have collected because of _include or _revinclude directives.

On first sight of a paginated query, the FHIR Aggregator creates the data structure and allocates it a unique identifier. The identifier is used in constructing the URLs of the navigation links in the query response. The query is then passed down the message pathway and issued to all data providers that are a potential source of results.

For subsequent requests for the same query then it may be possible to serve results directly from the page cache without placing further requests with data providers. Results can be constructed here for

- a page which has already been served where results are in the backward cache;
- a new page from an unsorted query where there is more than page size of resources in the forward cache;
- a new page from a sorted query where there is more than a page size of resources in the forward cache for every non-exhausted data provider.

If one of these conditions does not hold then the query is passed down the message pathway and issued to all data providers from whom further results are required.

3.3.2 Regional Linkage Query?

The regional linkage query as specified by 2.4 requires special processing. The query identified as:

Resource Type: Linkage, Include: Linkage:item.

3.3.3 Determine Target Endpoints

Identifies the target data providers to which the request is to be issued.

Resource Type	Qualifier	Data Sources
Non-Administrative, Patient Centric Query		
Patient	Source implied search string [1]	Sources implied in search string
	No sources implied in search string [1]	Regional FHIR Store
*	Source implied search string [1]	Sources implied in search string
	No sources implied in search string [1]	Sources with registered patient contact [2]
Non-Administrative, Non-Patient-Centric Query		
Organisation/ Practitioner	Source implied search string [1]	Sources implied in search string
	No sources implied in search string [1]	Regional FHIR Store
*	Source implied search string [1]	Sources implied in search string
	No sources implied in search string [1]	All data providers offering the resource type
Administrative Query		
AuditEvent [3]	Source implied search string [1]	Sources implied in search string
	No sources implied in search string [1]	Regional FHIR Store
*		Regional FHIR Store
Resource Retrieval		
*		Source determined by resource identifier
Create Resource		
*		Source determined by resource Source meta tag
Update Resource (Direct resource reference)		
*		Source determined by resource identifier
Update Resource (Conditional update)		
*	Source implied search string [1]	Sources implied in search string
	No sources implied in search string [1]	All data providers offering the resource type

(1) A source is implied by a search string if a search term includes a) a Source meta tag, b) a resource reference. Multiple search terms may lead to inconsistencies in the determination of sources. In this case the interaction of all alternative sources implied by search terms is taken.

(2) Patient contact is registered with the PIX server (design paper 004). The FHIR Aggregator determines sources by query the Linkage resources in the regional FHIR Store which are related to the relevant Patient.

(3) On the Auditor role has access to *AuditEvents*.

Note that the sources for paginated queries are initially determined by the method detailed above but for follow-up page requests sources are determined by the need to refresh data by the component detailed in 3.3.1.

3.3.4 Determine Applicable Policies

Identifies those data access policies which might ultimately apply to data released to the consumer. Before releasing data to consumers, it will be tested by 3.7.1 to determine whether it is covered by a data access policy. If so, resources may be withheld or released conditionally.

As detailed design paper 008 – “Data Access and Consent Management”, a data access policy specifies the data points to which it applies. A data point is the combination of a FHIR resource type and a search path which can be tested to determine whether a resource is within scope of the policy.

A search path may be contained within the resource. For an example for an *Observation* a search path:

```
code='pregnancy status'
```

might be used to restrict access to pregnancy status and this information is wholly contained within the resource itself: it is sufficient to have access to an Observation resource in isolation to determine whether the policy applies.

Search paths can extend outside of the boundary of the resource. A chained search path uses properties of resources that are referenced by the resource that is controlled by the policy. For example, for an *Observation* a search path:

```
encounter.type='pregnancy test'
```

might be used to restrict access to any observations taken under an *Encounter* which is designated as a pregnancy test. To determine whether an observation is covered by this policy then access to the related *Encounter* resource is also required.

Pre-processing of the request can determine the policies which apply to the content which is returned. With this knowledge the request can be enhanced to collect all relevant data in one search.

Applicable policies are:

1) Policies that apply to a particular data provider which will be a participant in the request.
2) Policies to which the subject of a patient-centric request has consented and reference the resource types being collected by a search in their scope.
3) Policies with reference the resource types being collected under a non-patient-centric search.

The resource types being collected include the resource type being retrieved or that is the subject of a search and any resource types included or reverse included in a search.

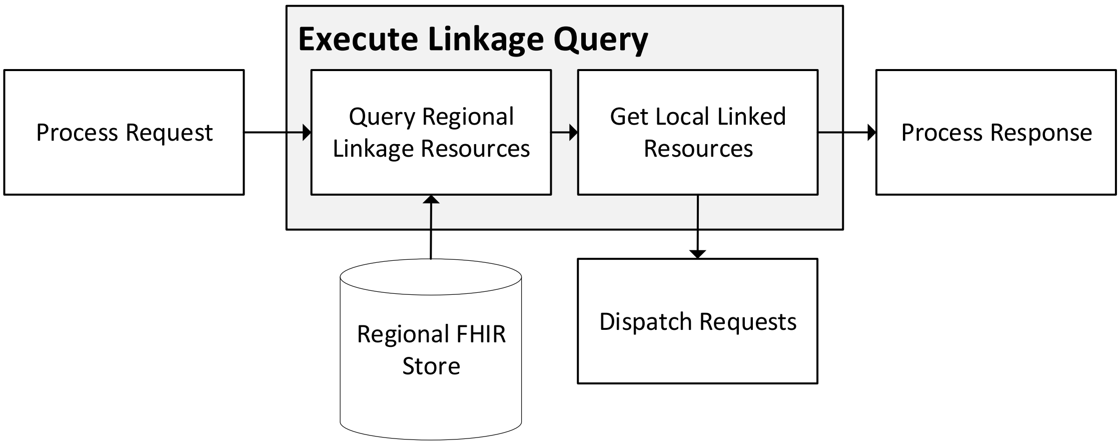
3.3.5 Augment Search for Policy Enforcement Resources

If data access policies apply, then the search (or resource retrieval) may need to be modified to ensure that all properties and resources which are required to test the policy are collected in a single query without needing follow interactions with the data source.

i) For each applicable policy determine the properties and chained resources which are required to enforce the policy;
ii) If the query uses the _summary directive and additional properties are required for policy enforcement which are not summary properties, then replace the _summary directive with an appropriately constructed _elements directive;
iii) Add resources included in the search through the _include directive to include those required for policy enforcement;
iv) If a resource retrieval request requires additional resource for policy enforcement purposes, then modify it to be a search based on the resource identifier.

3.4 Execute Linkage Query

A linkage query retrieves Linkage resources from the regional FHIR store and includes in the result set local resources which are referenced by the *Linkage*.



3.4.1 Query Regional Linkage Resources

Executes the query against the Regional FHIR Store.

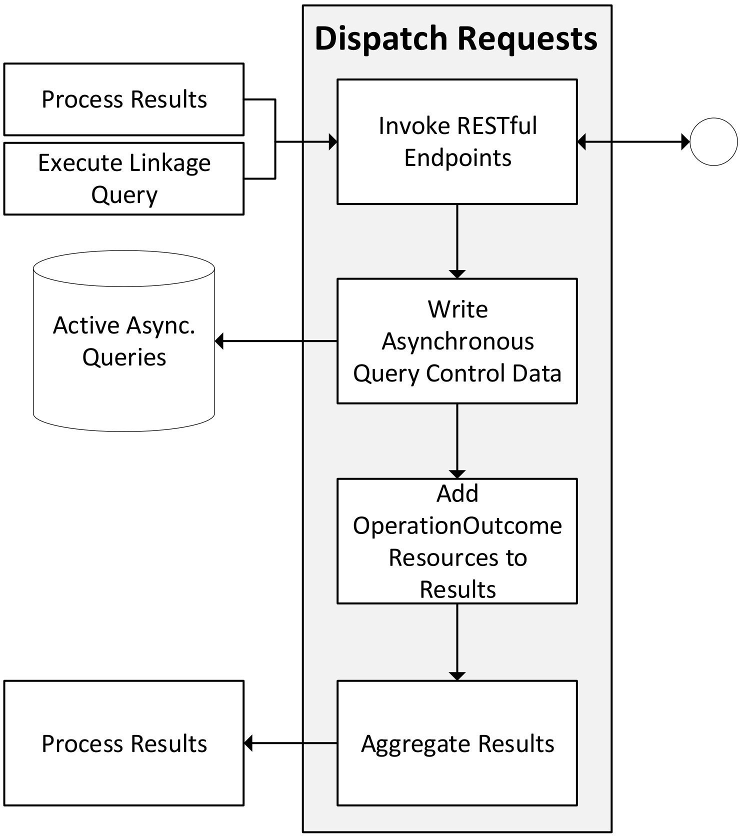
3.4.2 Get Local Linked Resources

For each linked item, constructs a resource retrieval request again the source of resource and issues the request to the data provider through Dispatch Requests.

Resources are retrieved in parallel and are results are added the query result set.

3.5 Dispatch Request

Requests are dispatched in parallel to all data sources.



3.5.1 Invoke RESTful Endpoints

An HTTPS client issues requests against each target endpoint. Requests are issues in parallel and return synchronously. A configurable timeout period can be set for each endpoint.

3.5.2 Write Asynchronous Query Control Data

Results for asynchronous requests are collected by a different message pathway, A simple data structure is persisted to record the event that requests have been issued for an asynchronous query. The data structure is keyed by the identifier generated in 3.2.7 and lists the endpoint to which the request has been issued, any error returned by the endpoint and the address of the endpoint to poll for results. The data structure also records any meta data about the request which has been collected in previous processing such as applicable data access policies.

The synchronous connection from the data consumer was terminated earlier in the message pathway and processing for the pathway ceases once control data has been written for all endpoints.

3.5.3 Add OperationOutcome Resources to Results

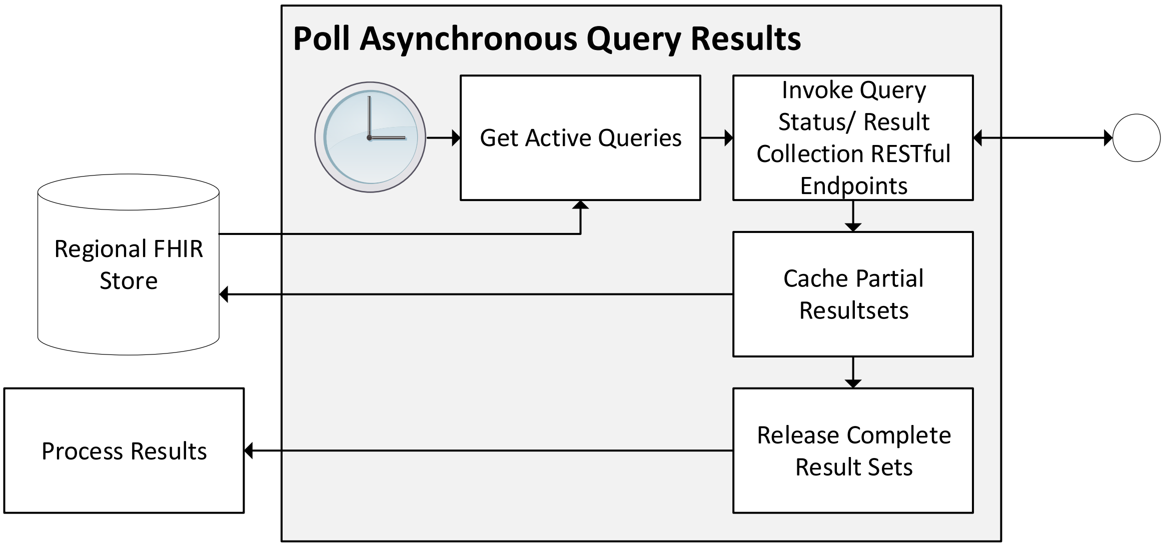
If the RESTful endpoint returns an error or times out, then this is recorded in an *OperationOutcome* resource which is included in the result set returned to the consumer. The YHCR profile for an OperationOutcome is provided by design paper 017 – "Data Quality Reporting".

3.5.4 Aggregate Results

Results returned by the RESTful endpoint are added to consolidated result set. Once the final endpoint returns then the consolidated result set is passed to the Process Results pathway.

3.6 Poll Asynchronous Query Results

Asynchronous results are collected by a process which run periodically and polls the asynchronous collection endpoint at data providers. The period between polling is configurable.


Poll Asynchronous Query Results
Get Active Queries
Invoke Query Status/ Result Collection RESTful Endpoints
Regional FHIR Store
Cache Partial Resultsets
Process Results
Release Complete Result Sets

3.6.1 Get Active Queries

Queries a local database for active control data written by 3.5.2. The control data lists endpoints to which the asynchronous query was issued and determines whether collection is complete or not. Where collection is collection is incomplete then…

3.6.2 Invoke Query Status/Result Collection RESTful Endpoints

The query will determine whether results are to ready to be collected. If results are available, then the follow up invocations are issued until collection is complete. A transient error is ignored and collection resumes at the next poll. A permanent error is recorded as an *OperationOutcome* and added to the result set recorded for the source. Collection attempts stop on receipt of a permanent error.

3.6.3 Cache Partial Results

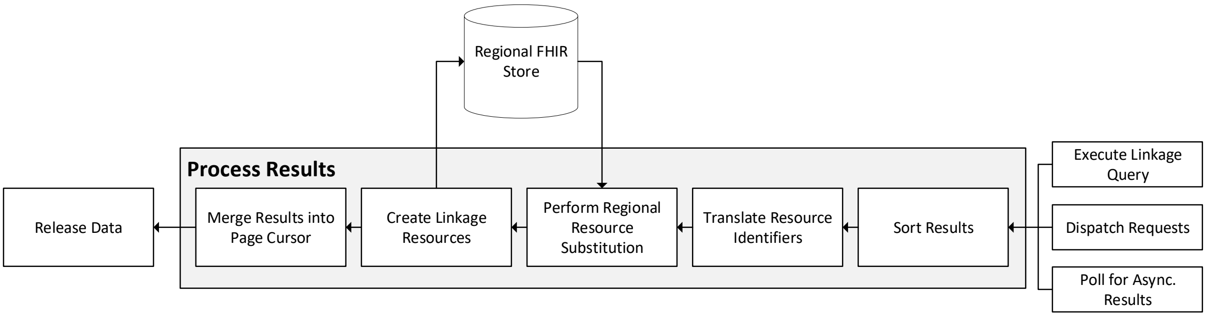
Results returned from an endpoint are recorded as an array against the control data for the query. Once all results are collected from an endpoint (or a permanent error is encountered) then collection for the endpoint is marked as complete.

3.6.4 Release Complete Result Sets

Once collection is complete from all endpoints then a consolidated result set is constructed. *OperationOutcomes* are added for any asynchronous query request which failed at step 3.5.1. Query results are passed to the Process Results pathway.

3.7 Process Results

A pathway which is used by all result aggregation methods.



3.7.1 Sort Results

Resources in the consolidated result set are sorted according to the search terms specified in the request. Note that this requires search parameters to be translated into resource properties using schema definitions imported in 3.1

3.7.2 Translate Resource Identifiers

All properties of type ‘Reference’ in all resources in the result set must be rebased to become a regional resource identifier as specified in 2.2

3.7.3 Create Linkage Resources

If the result includes *Organisation* and *Practitioner* resources, then these are candidates for linkage from a regional resource. Organisation resources business identifiers should include an ODS code and Practitioner resources an SDS code. These are used to identifier the regional resource which represents the equivalent concept.

For each Organisation and Practitioner resource in the result set:

i) Query the Regional FHIR Store based on the appropriate business identifier;
ii) If no regional resource exists, then create one populated from the local resource. Mark the regional resource as requiring a refresh a national system (design papers 013 – "Interfaces with ESR" and 014 – "Interfaces with ODS";
iii) If a regional resource exists but no *Linkage* resource exists to the local equivalent, then create a *Linkage* resource.

3.7.4 Perform Regional Resource Substitution

Replaces references to Patients, Organisations and Practitioners with references to the regional golden record.

For each resource in the result set and for each reference to a Patient, Organisation or Practitioner:

i) Query the Regional FHIR Store Linkage resources for a link to the resource;
ii) Substitute the reference with a reference to an equivalent resource in the Regional FHIR Store.

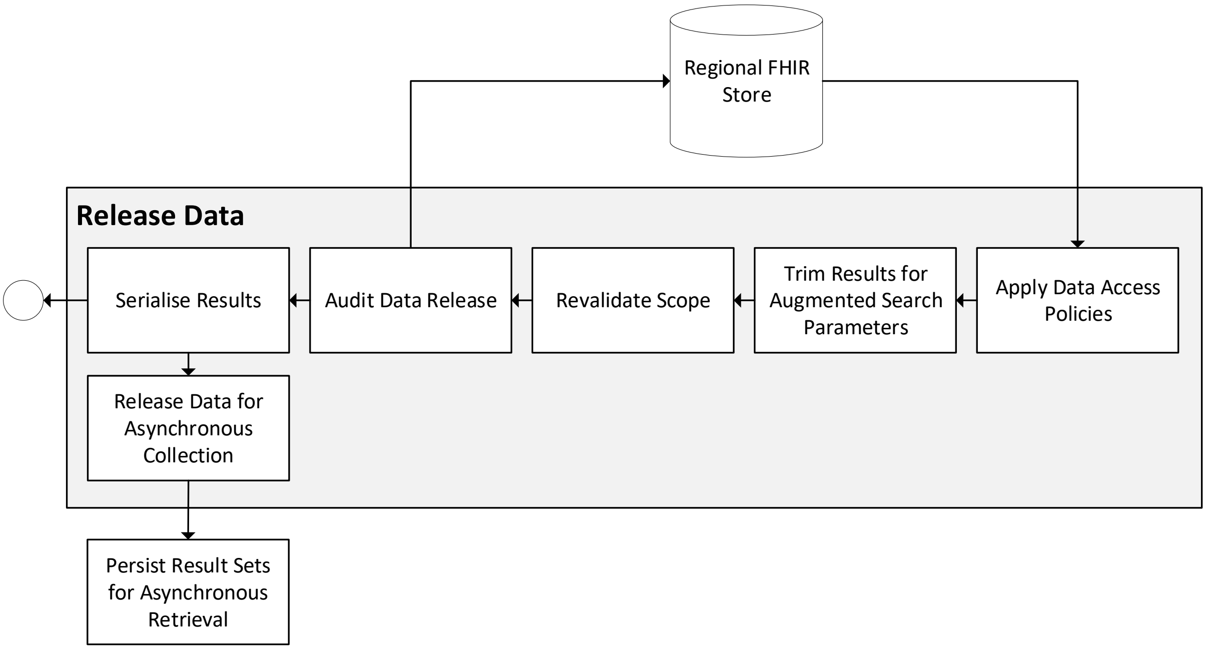
3.7.5 Merge Results into Page Cursor

If the response is the result of a query issued due to insufficient data being held in a paginated query cache, then the results must be added to the cache and the result set reconstructed from cached data.

Details of the cache structure are in 3.3.1.

3.8 Release Data

Applies common controls to all data released by the FHIR Aggregator. Data is either released synchronously over the socket connection which initiated the request or persisted locally for asynchronous collection.


Regional FHIR Store
Release Data
Serialise Results
Audit Data Release
Revalidate Scope
Trim Results for Augmented Search Parameters
Apply Data Access Policies
Release Data for Asynchronous Collection
Persist Result Sets for Asynchronous Retrieval

3.8.1 Apply Data Access Policies

Tests resources for each of the applicable data access policies identified in 3.3.4. Policies are processed in order pf precedence and determine whether a resource:

- is removed from the result set;
- is released to the consumer with qualification;
- is released to the consumer;

Tests are expressed as search terms which, given the augmentation of included resources performed in 3.3.5 should be determinable entirely given the data in the result set.

Resources may be removed from the result set and *OperationOutcomes* inserted depending on the actions required by applicable policies.

3.8.2 Trim Results for Augmented Search Parameters

Resources may have been included in the result set because of the need to augment results for policy enforcement. These results are now removed.

3.8.3 Revalidate Scope

Requests can be made to the FHIR Aggregator where the scope of resources returned cannot be validated without access to the data. An example is a simple resource retrieval by resource identifier.

This component re-applies the rules specified by 3.2.4 to each resource being released. Any single resource failing scope validation results in the whole result set being rejected and an error being returned to the data consumer.

Note that scope validation at this stage resolves ambiguity for certain possibly patient identifiable resource types as detailed by Design Paper 005 – “Identity and Access Management” which may or may not be patient identifiable depending on context.

3.8.4 Audit Data Release

An *AuditEvent* is written to the regional FHIR Store.

3.8.5 Serialize Results

The REST request stipulates the format in which data is to be returned. Supported options are XML or JSON. This module serialises an internal object graph structure into the required representation.

3.8.6 Release Data for Asynchronous Collection

The option to collect asynchronously was established by the invoker of the service and a unique identifier was assigned to the request. Backend processing ensures that a complete set of search results can be released by the aggregator at this stage.

3.9 Persist Result Sets for Asynchronous Retrieval

Persists the paginated result set. Each page is assigned a unique identifier and can be retrieved independently.

3.10 Process Asynchronous Request

Handles the two categories of request which are supported by the asynchronous processing endpoint:

1. Request status update.
2. Retrieve result page.

Status update requests quote a unique request identifier and if validly formed the possible responses are an HTTP 202 response code for requests which are in progress or an HTTP 200 response code for requests where the search is complete, and results may be collected.

If issuing an HTTP 200 response, then the HTTP is a structure which includes an array of URLs from which individual search pages can be retrieved.

Result page retrieval requests are made against one of the advertised URLs. The response is a bundle of resources retrieved from the original search request. Once retrieved the persisted bundle is marked as such and will be purged by a batch process (not shown in the processing model). An attempt to retrieve a purged bundle will result in a 404 HTTP response.

Appendix 1 – Maturity Matrix

Section	Narrative	Consultative	Draft	Normative
1 Introduction 1.1 Purpose of this Document	X			
1.2 Functions of the FHIR Aggregator	X			
1.3 A Unified Data Model	X			
1.4 Relationship of this Document with Other Standards	X			
1.5 Intended Users of this Document	X			
2 Functional Overview 2.1 Relationship with Other Components	X			
2.2 Local Identifiers, Regional Identifiers, and Resource Provenance			X	
2.3 Behaviour of Standard REST Operations 2.3.1 Patient Centric Searches			X	
2.3.2 Regionally Held De-Duplicated Resource Types			X	
2.4 Querying Regional Linkage Resources			X	
2.5 Enforcement of Role and Reason Based Access Rights			X	
2.6 Enforcement of Consent and Policy-Based Access Rights			X	
2.7 Pagination			X	
2.8 Asynchronous Queries			X	
3 Processing Model 3.1 Import FHIR Definitions		X		
3.2 Validate Request & Decompose URL 3.2.1 Read and Validate Resource Payload		X		
3.2.2 Validate JSON Patch Document		X		
3.2.3 Decompose URL		X		
3.2.4 Validate Scope		X		
3.2.5 Validate Against Capability Statement		X		
3.2.6 Audit Accepted Request		X		
3.2.7 Return Response for Asynchronous Request		X		
3.2.8 Audit Rejected Request		X		

3.3 Process Request 3.3.1 Manage Page Cursor		X		
3.3.2 Regional Linkage Query?		X		
3.3.3 Determine Target Endpoints		X		
3.3.4 Determine Applicable Policies		X		
3.3.5 Augment Search for Policy Enforcement Resources		X		
3.4 Execute Linkage Query 3.4.1 Query Regional Linkage Resources		X		
3.4.2 Get Local Linked Resources		X		
3.5 Dispatch Request 3.5.1 Invoke RESTful Endpoints		X		
3.5.2 Write Asynchronous Query Control Data		X		
3.5.3 Add OperationOutcome Resources to Results		X		
3.5.4 Aggregate Results		X		
3.6 Poll Asynchronous Query Results 3.6.1 Get Active Queries		X		
3.6.2 Invoke Query Status/Result Collection RESTful Endpoints		X		
3.6.3 Cache Partial Results		X		
3.6.4 Release Complete Result Sets		X		
3.7 Process Results 3.7.1 Sort Results		X		
3.7.2 Translate Resource Identifiers		X		
3.7.3 Create Linkage Resources		X		
3.7.4 Perform Regional Resource Substitution		X		
3.7.5 Merge Results into Page Cursor		X		
3.8 Release Data 3.8.1 Apply Data Access Policies		X		
3.8.2 Trim Results for Augmented Search Parameters		X		
3.8.3 Revalidate Scope		X		
3.8.4 Audit Data Release		X		
3.8.5 Serialize Results		X		
3.8.6 Release Data for Asynchronous Collection		X		
3.9 Persist Result Sets for Asynchronous Retrieval		X		
3.10 Process Asynchronous Request		X		