

Integrating the Healthcare Enterprise



5

**IHE IT Infrastructure
Technical Framework Supplement**

10

**Cross-Community Fetch
(XCF)**

15

Rev. 1.6 – Trial Implementation

20 Date: July 12, 2019
Author: ITI Technical Committee
Email: iti@ihe.net

25

Please verify you have the most recent version of this document. See [here](#) for Trial Implementation and Final Text versions and [here](#) for Public Comment versions.

Foreword

30 This is a supplement to the IHE IT Infrastructure Technical Framework V16.0. Each supplement undergoes a process of public comment and trial implementation before being incorporated into the volumes of the Technical Frameworks.

This supplement is published on July 12, 2019 for trial implementation and may be available for testing at subsequent IHE Connectathons. The supplement may be amended based on the results of testing. Following successful testing it will be incorporated into the IT Infrastructure Technical Framework. Comments are invited and may be submitted at
35 http://www.ihe.net/ITI_Public_Comments.

This supplement describes changes to the existing technical framework documents.

“Boxed” instructions like the sample below indicate to the Volume Editor how to integrate the relevant section(s) into the relevant Technical Framework volume.

<i>Amend Section X.X by the following:</i>
--

40 Where the amendment adds text, make the added text **bold underline**. Where the amendment removes text, make the removed text **~~bold strikethrough~~**. When entire new sections are added, introduce with editor’s instructions to “add new text” or similar, which for readability are not bolded or underlined.

45 General information about IHE can be found at <http://www.ihe.net>.

Information about the IHE IT Infrastructure domain can be found at http://www.ihe.net/IHE_Domains.

Information about the structure of IHE Technical Frameworks and Supplements can be found at http://www.ihe.net/IHE_Process and <http://www.ihe.net/Profiles>.

50 The current version of the IHE Technical Framework can be found at http://www.ihe.net/Technical_Frameworks.

CONTENTS

55	Introduction.....	5
	Open Issues and Questions	5
	Closed Issues.....	5
	Volume 1 – Profiles	7
60	1.7 History of Annual Changes.....	7
	29 Cross-Community Fetch (XCF) Profile.....	7
	29.1 Actors/Transactions.....	8
	29.2 XCF Profile Options	8
	29.2.1 Asynchronous Web Services Exchange Option (WS-Addressing based)	9
65	29.3 XCF Actor Groupings and Profile Interactions	9
	29.3.1 XCF Required Groupings.....	9
	29.3.2 XDS/XCA Interactions (Informative).....	10
	29.3.2.1 “responding agent” for XDS (Grouping with Document Consumer)	10
	29.3.2.2 “responding agent” for XCA	11
70	29.3.2.3 “initiating agent” for XDS.....	12
	29.3.2.4 “initiating agent” for XCA	13
	29.3.3 Profile Interactions (Informative).....	14
	29.4 XCF Process Flow	15
	29.4.1 Use Cases	15
75	29.4.1.1 Patient Summary Service with Translation/Transforming Use Case	16
	29.4.1.2 Highly Regulated Data Sharing Scenarios	16
	29.4.2 Process Flow	16
	29.5 XCF Profile Security Considerations.....	18
	29.5.1 XCF Risk Assessment.....	18
80	Appendix B – Transaction Summary Definitions.....	20
	Volume 2b – Transactions.....	21
	3.63 Cross Gateway Fetch.....	21
	3.63.1 Scope	21
	3.63.2 Use Case Roles.....	21
85	3.63.3 Referenced Standards.....	21
	3.63.4 Messages	22
	3.63.4.1 Cross Gateway Fetch Request.....	22
	3.63.4.1.1 Trigger Events	22
	3.63.4.1.2 Message Semantics.....	22
90	3.63.4.1.2.1 Query Parameters for Cross Gateway Fetch Requests	22
	3.63.4.1.2.2 Use of homeCommunityId.....	23
	3.63.4.1.2.3 Inclusion of Document Associations	24
	3.63.4.1.3 Cross Gateway Fetch request message example	24
	3.63.4.1.4 Expected Actions	25
95	3.63.4.2 Cross Gateway Fetch Response.....	25
	3.63.4.2.1 Trigger Events	25

	3.63.4.2.2 Message Semantics.....	25
	3.63.4.2.3 Expected Actions.....	26
	3.63.4.2.4 Document Metadata.....	26
100	3.63.4.2.5 Error Codes.....	27
	3.63.5 Protocol Requirements.....	27
	3.63.6 Security Considerations.....	28
	3.63.6.1 Security Audit Considerations.....	28
	3.63.7 Sample Request Message (Informative).....	28
105	3.63.8 Sample Response Message (Informative).....	29
	Volume 3 – Cross-Transaction Specifications and Content Specifications.....	38

110

115

Introduction

120 The Cross-Community Fetch (XCF) Profile defines a single transaction for accessing medical data between gateways that facilitate multiple dimensions of communication (trust, semantics, encoding, legislation, authority, etc.). The profile is highly inspired by the Cross Gateway Query/Cross Gateway Retrieve transactions and integrates these originally distinct transactions.

125 In specific use cases, for example when a few dynamically created documents need to be accessed from interacting communities with centralized data localization, a single transaction (versus independent query and retrieve) may reduce the coordination and maintenance of the transactional dependencies and transaction states.

130 For such use cases, and in environments where stateless Responding Gateways can be designed, it simplifies the implementation of such Responding Gateways. However, it may increase the implementation complexity of Initiating Gateways serving some types of communities, such as XDS Affinity Domains. XCF offers a different deployment option from the general purpose XCA Profile.

Open Issues and Questions

- 135 **XCF008:** This supplement introduces the concept of automated document transforms at the XCF Responding Gateway while not necessarily making the transformed document persistent. For patient safety and traceability reasons this aspect might need to be discussed further.
- XCF009:** The relationship/difference to on-demand should be provided.

Closed Issues

- 140 **XCF001:** Can the XDS FindDocuments UUID be used as a query id for the Fetch, too? Decision is made (TCon on 2011-04-20) to assign a new UUID as Fetch is an adapted subset of FindDocuments query with a specific semantics.
- 145 **XCF002:** How does the initiating side interact with an XDS affinity domain (e.g., if the Initiating Gateway is part of that affinity domain)? How can an XDS Document Consumer interact with an XCF Initiating Gateway? Are there at all any use cases, where it makes sense that an XDS document consumer performs cross-community document sharing by using XCF instead of XCA? See Section 29.3 for guidance.
- XCF003:** What happens if a message is routed through a network of both XCA and XCF gateways? Are these gateways interoperable? See Section 29.3 for guidance.
- XCF004:** How can document relationships be expressed in a Fetch response? Document relationship can be expressed by ebRS Association Objects which are places within the RegistryObjectList.
- 150 **XCF005:** OASIS ebXML Registry allows for iterations on query results by defining an optional startIndex element within a query request. By using this element one of the major limitations of Fetch – the inability to deal with large result sets – could be resolved. Should the use of startIndex be allowed as an option for the XCF Profile? Decision is: NO. Such functionality may

155 contradict the “simple” approach of the profile by greatly adding to its potential complexity.
XCA is assumed to be used for massive data sets that require special treatment.

XCF006: How should actors and transaction be named? Options discussed for the transaction are “Query and Retrieve”, “Simple Query” and “Cross Gateway Simple Query”. The actor names “Initiating Gateway” and “Responding Gateway” are already used for XCA Gateways. Decision is “Cross-Community Fetch” for the profile and “Cross Gateway Fetch” for the transaction.

160 **XCF007:** Risk analysis must be performed on the question whether a Responding Gateway
should respond with a NoConsent error in case that the patient has not given consent to sharing
his data. As an alternative a “neutral” error or an empty result set could be returned. Using a
NoConsent error only makes sense if the patient could then give the required consent at the point
of care which would allow a physician to re-issue the request. Decision: In a rather general
165 environment issuing such an error might be considered as an unlawful data disclosure by
revealing that there (1) is such a patient that (2) has data available but (3) did not consent into
electronic data processing of his medical information. While the accompanying XUA assertion
of a health care professional may be interpreted as forming a trusted environment that may
actually justify this error message as facilitator, such an assumption is not valid for all
170 environments/contexts. Therefore, such an error may not be issued by default. We agreed to
return no documents.

Volume 1 – Profiles

1.7 History of Annual Changes

175 *Add the following bullet to the end of the bullet list in Section 1.7*

- Added the Cross-Community Fetch Profile for exchanging accessing medical data between stateless gateways that facilitate multiple dimensions of communication (trust, semantics, encoding, legislation, authority, etc.).

180 *Add the following new section*

29 Cross-Community Fetch (XCF) Profile

185 The Cross-Community Fetch (XCF) Profile defines a single transaction for accessing medical data between gateways that facilitate multiple dimensions of communication (trust, semantics, encoding, legislation, authority, etc.). The profile is highly inspired by the Cross Gateway Query/Cross Gateway Retrieve transactions.

In specific use cases, for example when a few dynamically created documents need to be accessed from interacting communities with centralized data localization; a single transaction (versus independent query and retrieve) may reduce the coordination and maintenance of the transactional dependencies and transaction states.

190 For such use cases, and in environments where stateless Responding Gateways can be designed, XCF simplifies the implementation of such Responding Gateways. However, it may increase the implementation complexity of Initiating Gateways serving some types of communities, such as XDS Affinity Domains. XCF offers a different deployment option from the general purpose XCA Profile.

195 The transaction fetches a small number of documents based upon a few retrieval parameters. This transaction is simplified to permit easier implementation and better performance on Responding Gateways.

Transcoding and translation of the documents and other data can be performed on the Responding Gateway as part of the transaction.

200 The XCF Profile stipulates that the following prerequisites are met:

- the document properties to be communicated are known in advance
- the result data sets can be characterized in advance
- the documents are feasible to be returned in a single response
- no further selection and/or manual interaction is needed in the communication process

- 205
- pre-conditions, such as purpose of use, legitimate data, and environment, are agreed upon in advance and are documented in a community or framework agreement
 - the document fetching may not always be repeatable – it may not be assumed in every case that the same query with the same query parameters will return the same document version with the same document id.

210 Ideally, only one document will satisfy the Fetch (e.g., only the most current instance of a patient summary is provided by the Responding Gateway). If the size of the set of documents matching the request is too large to be packed into a single response, an error code is returned by the Responding Gateway. The assumption is that the Cross-Community Access (XCA) Profile is used when requests are expected to return a large number of documents.

215 **29.1 Actors/Transactions**

Figure 29.1-1 shows the actors directly involved in the XCF Profile and the relevant transactions between them.

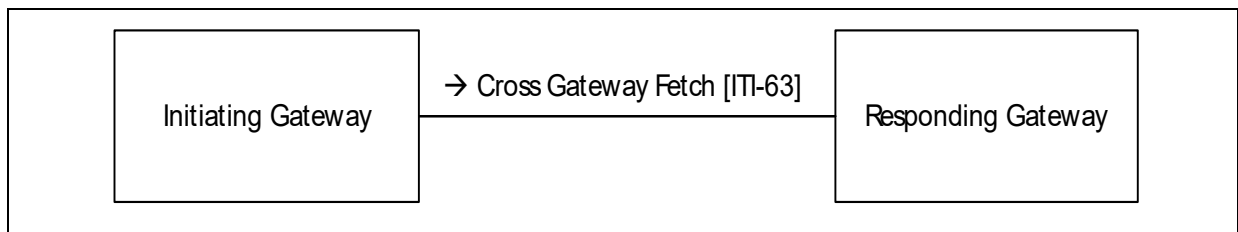


Figure 29.1-1: XCF Actor Diagram

220 Table 29.1-1 lists the transactions for each actor directly involved in the XCF Profile. In order to claim support of this profile, an implementation must perform the required transactions (labeled “R”). Transactions labeled “O” are optional. A complete list of options defined by this profile and that implementations may choose to support is listed in Section 29.2.

Table 29.1-1: XCF Profile - Actors and Transactions

Actors	Transactions	Optionality	Reference
Initiating Gateway	Cross Gateway Fetch [ITI-63]	R	ITI TF-2b: 3.63
Responding Gateway	Cross Gateway Fetch [ITI-63]	R	ITI TF-2b: 3.63

225

29.2 XCF Profile Options

Options that may be selected for this profile are listed in the Table 29.2-1 along with the actors to which they apply. Dependencies between options when applicable are specified in notes.

Table 29.2-1: XCF - Actors and Options

Actor	Options	Reference
Responding Gateway	No options defined	–
Initiating Gateway	Asynchronous Web Services Exchange	Section 29.2.1

230

The Responding Gateways shall support WS-Addressing based Asynchronous Web Services Exchange on the Cross Gateway Fetch (see ITI TF-2x: Appendix V.3). Support for this function is required in order to enable use of Asynchronous Web Services Exchange in any cross-community interaction.

235 **29.2.1 Asynchronous Web Services Exchange Option (WS-Addressing based)**

Asynchronous processing is necessary to support scaling to large numbers of sources and recipients because Asynchronous Web Services Exchange allows for more efficient handling of latency and scale. This WS-Addressing based Asynchronous Web Services Exchange stack relies on the Web Service Addressing Stack (see also ITI TF-2x Appendix: V.3).

240 Initiating Gateways which support Asynchronous Web Services Exchange shall support WS-Addressing based Asynchronous Web Services Exchange on the Cross Gateway Fetch [ITI-63] transaction.

29.3 XCF Actor Groupings and Profile Interactions

29.3.1 XCF Required Groupings

245 An actor from this profile (Column 1) shall implement all of the required transactions and/or content modules in this profile *in addition to* all of the transactions required for the grouped actor (Column 2).

250 If this is a content profile, and actors from this profile are grouped with actors from a workflow or transport profile, the Content Bindings reference column references any specifications for mapping data from the content module into data elements from the workflow or transport transactions.

Table 29.3.1-1: XCF - Required Actor Groupings

XCF Actor	Actor to be grouped with	Reference	Content Bindings Reference
Initiating Gateway	ATNA / Secure Node or Secure Application	ITI TF-1: 9	--
	CT / Time Client	ITI TF-1: 7	--
	XUA / X-Service User	ITI TF-1: 13	--
Responding Gateway	ATNA / Secure Node or Secure Application	ITI TF-1: 9	--
	CT / Time Client	ITI TF-1: 7	--

XCF Actor	Actor to be grouped with	Reference	Content Bindings Reference
	XUA / X-Service Provider	ITI TF-1: 13	--

29.3.2 XDS/XCA Interactions (Informative)

255 Interoperable interaction between communities which have chosen to implement only XCF and those that are based on XDS or XCA may be enabled through transformation agents. IHE does not specify the mechanism used by such transformation agents or any details about their implementation. The following sections give a high level perspective on the challenges of enabling four cases of agents:

- 260 1. “responding agent” for XDS- acts as an XCF Responding Gateway and converts incoming Cross Gateway Fetch transactions into XDS transactions to collect the content needed for the response.
- 265 2. “responding agent” for XCA- acts as an XCF Responding Gateway and converts incoming Cross Gateway Fetch transactions into XCA transactions to collect the content needed for the response.
- 270 3. “initiating agent” for XDS – acts as an XCF Initiating Gateway and converts XDS transactions into Cross Gateway Fetch transactions to collect content from XCF only communities.
- 270 4. “initiating agent” for XCA– acts as an XCF Initiating Gateway and converts XCA transactions into Cross Gateway Fetch transactions to collect content from XCF only communities.

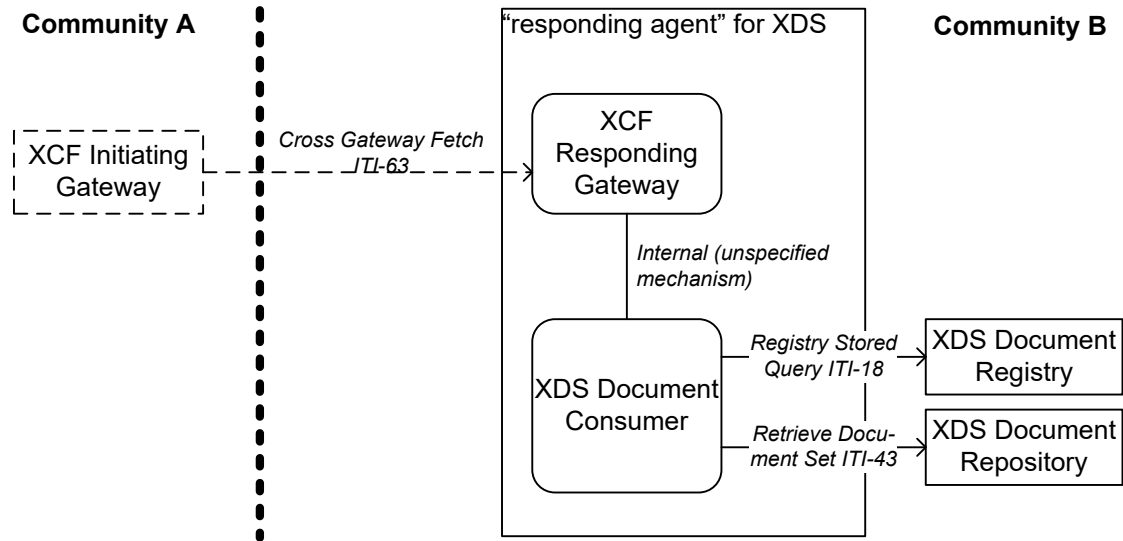
Some agents are relatively easy to implement and others are quite complicated. In environments where integration of with XCA and XDS is important it would be advisable to consider XCA with the On-Demand Documents Option as an alternative to the use of XCF.

275 29.3.2.1 “responding agent” for XDS (Grouping with Document Consumer)

A “responding agent” for XDS converts incoming Cross Gateway Fetch transactions into Registry Stored Query [ITI-18] and Retrieve Document Set [ITI-43] transactions which are directed to a local XDS Registry/Repository. This type of agent has value because it allows access by XCF only communities to content within XDS based communities.

280 A “responding agent” for XDS can be enabled through a relatively simple grouping of XDS Document Consumer and XCF Responding Gateway. The agent must convert the Cross Gateway Fetch query into a collection of Registry Stored Query and Retrieve Document Set transactions. This conversion is relatively straightforward; the query in the Cross Gateway Fetch transaction maps closely to the Find Documents stored query of Registry Stored Query and from this query
285 the agent can generate appropriate Retrieve Document Set transactions to get the document contents. Several additional details need to be managed by the agent, like supplying document

associations and handling situations when the results are too large to be returned in the Cross Gateway Fetch response. Figure 29.3.2.1-1 depicts this environment.



290

Figure 29.3.2.1-1: “responding agent” for XDS

29.3.2.2 “responding agent” for XCA

A “responding agent” for XCA converts incoming Cross Gateway Fetch transactions into Cross Gateway Query and Cross Gateway Retrieve transactions which are directed to a XCA Responding Gateway. This type of agent has value because it allows access by XCF only communities to content within XCA based communities.

A “responding agent” for XCA groups with an XCA Initiating Gateway in order to initiate Cross Gateway Query and Cross Gateway Retrieve transactions to a XCA Responding Gateway. The agent must convert the Cross Gateway Fetch query into an appropriate query supported by Cross Gateway Query and must interpret and collect the results of the Cross Gateway Query and Cross Gateway Retrieve in order to respond to the Cross Gateway Fetch transaction. The query mapping and translation across transactions is equivalent to the work involved in a “responding agent” for XDS. Figure 29.3.2.2-1 depicts this environment.

300

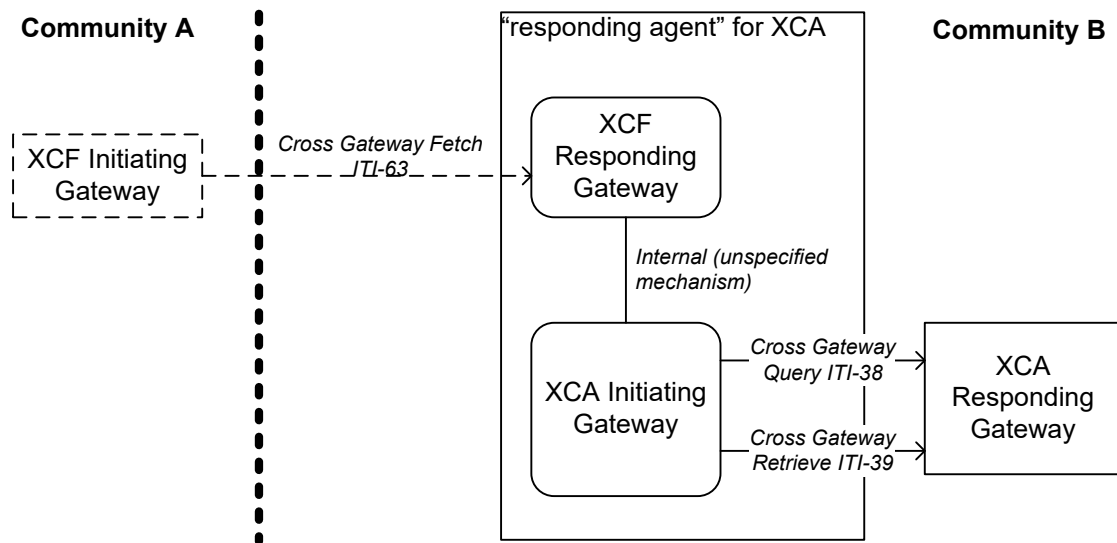


Figure 29.3.2.2-1: “responding agent” for XCA

305 **29.3.2.3 “initiating agent” for XDS**

An “initiating agent” for XDS enables access by the significant number of products supporting the XDS Document Consumer to content within a community that only supports XCF. Without this kind of enablement EMR/EHR systems (and others) will be cut off from the content held by a community that chooses to support only XCF. Enabling this interaction is more difficult than the other direction and this section only skims the surface of the work involved.

310 The “initiating agent” for XDS must be able to convert the contents of Registry Stored Query [ITI-18] transactions into Cross Gateway Fetch transactions. Typically, this will involve the conversion of the Find Documents stored query. Along with copying all the parameters, the “initiating agent” for XDS must also manage a few other aspects of the query request. Handling
 315 the response from the Cross Gateway Fetch transactions involves storing locally the parts not immediately requested by the XDS Document Consumer and returning only the parts that are appropriate. For example, the Cross Gateway Fetch transaction will return the documents associated with the metadata. The “initiating agent” for XDS cannot return these documents in the Registry Stored Query transaction so must save them locally in order to be able to return
 320 them upon receipt of a Retrieve Document Set transaction. The local storage, called “Copy of Community B content” in the diagram, looks a little like an XDS Registry/Repository system managed and used by the “initiating agent” for XDS. This storage will also need to hold the metadata returned in the Cross Gateway Fetch to respond to Registry Stored Query transactions that use stored queries other than Find Documents. The complete design of the “initiating agent”
 325 for XDS is a non-trivial task and not further described by IHE.

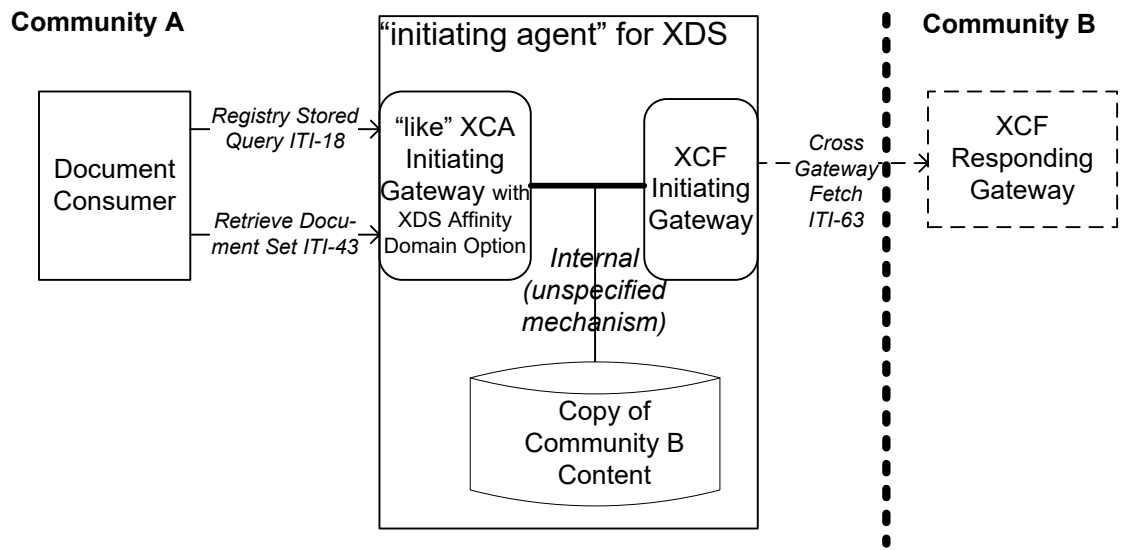


Figure 29.3.2.3-1: “initiating agent” for XDS

29.3.2.4 “initiating agent” for XCA

330 An “initiating agent” for XCA enables access by communities using a XCA Initiating Gateway to access content within a community that only supports XCF. Without this kind of enablement XCA communities will be cut off from the content held by a community that chooses to support only XCF. Enabling this interaction is very similar to the enablement for XDS. Figure 29.3.1.4-1 presents a view of how this enablement might be designed and represents the small differences

335 from the “initiating agent” for XDS described in Section 29.3.2.4.

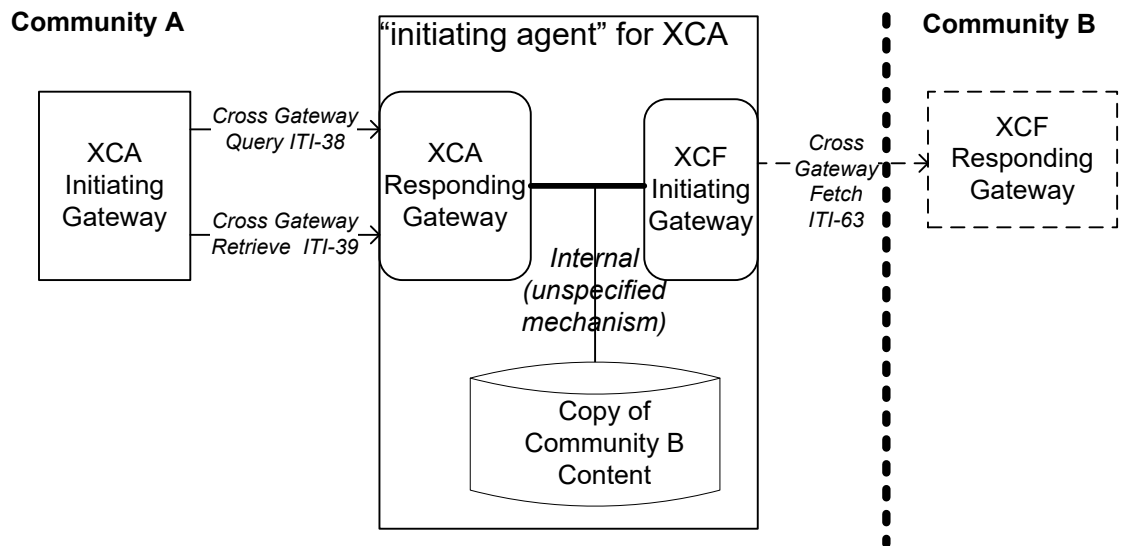


Figure 29.3.2.4-1: “initiating agent” for XCA

29.3.3 Profile Interactions (Informative)

Potential interactions for XCF with other IHE profiles are illustrated in this real world example.

340 It is assumed that a gateway infrastructure is set up for sharing patient’s medical summary data among autonomous regions. Each region collects a different set of data for its patients and makes use of its own document schema and builds upon its specific taxonomies for coding values. Patients may define privacy policies for their data and give general consent for data sharing in their region of affiliation while healthcare professionals are authenticated in the region of care.

345 Gateways perform all the transcoding, trust brokerage and access control enforcement such that all the (technical) complexity of this use case is hidden from the existing regional infrastructures and the acting persons.

As a result of this hidden complexity, from the physician’s perspective this use case is just a single operation: retrieval of an identified patient’s medical summary.

350 Organizationally, the concrete service delivery steps may be assigned to existing relevant IHE profiles for partial task fulfillment.

Table 29.3.3-1: IHE Profile Assignment / Interaction

Service Delivery Step	Support provided by IHE Profiles
Claim about requestor authenticity	Initiating gateway grouped with IHE XUA X-Service-User Responding gateway grouped with IHE XUA X-Service-Provider
Provisioning of requestor identity attributes (local roles, permissions, treatment context, delegation)	IHE XUA attributes
Establishing and Verifying Trust Relationships between the initiating and responding gateways	Deployment of the gateways as IHE ATNA Secure Nodes
Establishing an Audit Trail for traceability between initiating and responding gateway	IHE ATNA Audit Trail
Verification of the patient’s privacy consent	IHE BPPC encoded consents accessed through IHE XDS transactions.
Assurance of health information integrity and originator authenticity	IHE DSG for document digital signatures
Policy Decision and Policy Enforcement at each of the gateways	Policy Decision and Policy Enforcement (IHE White Paper on Access Control)
Health Information Exchange with opaque regional infrastructures	May be implemented by IHE XDS or other Entity Services → intentionally opaque
Canonical encoding of patient summary document for sharing information among autonomous regions	e.g., IHE PCC XDS-MS (Medical Summary Document Content) as content model

355 The translation and transcoding of patient summary data from a regional encoding into the canonical encoding at the Responding Gateway and the reverse transformation at the Initiating Gateway are out of the scope of IHE and subject to individual implementation (even though profiles like IHE SVS can help with the management of value sets). The Figure 29.3.3-1 shows

the respective document transformations, which require – from the consumer’s perspective - at least two intermediary documents.

360

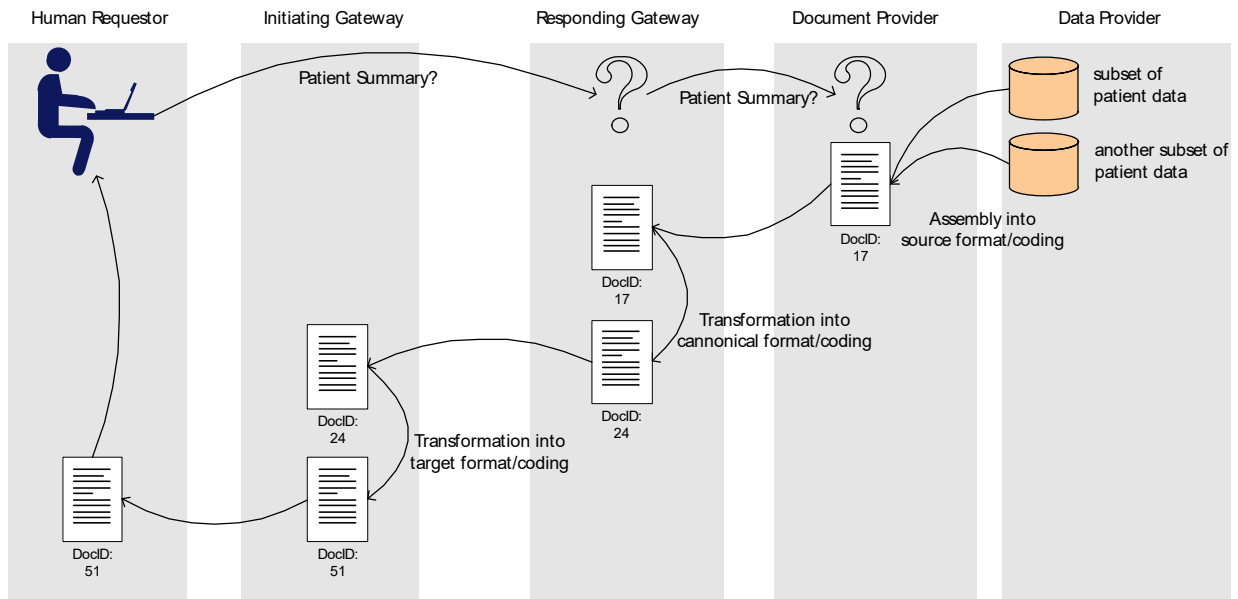


Figure 29.3.3-1: Document ID flow in cross-country use case with translation/transcoding scenario

This use case gives an impression of the problems that may arise if data discovery and retrieval across data processing gateways are split among multiple transactions within a logical session:

365

- Consents and policies have to be enforced again with each request. This requires that each transaction carries enough information to allow the responding gateway to discover and enforce the matching policies

By using the Cross Gateway Fetch transaction, the participating Responding Gateways and actors do not have save the intermediate formats and objects. Another way of avoiding saving intermediate formats is to use the On-Demand Documents Option of XCA. For accessing the requested, transcoded data all security checks must only be performed once (including certificate verification, consent fetching, and policy enforcements) and common information for policy discovery and assessment (particularly patient identifier and document type) is available at the responding gateway before any medical data has been accessed. However, it may increase the implementation complexity of Initiating Gateways serving some types of communities, such as XDS Affinity Domains.

370

375

29.4 XCF Process Flow

29.4.1 Use Cases

380 **29.4.1.1 Patient Summary Service with Translation/Transforming Use Case**

This use case may be supported either by XCA or XCF. A typical use case where data is processed on gateways is health data sharing among autonomous regions (states, countries) with distinct healthcare infrastructures and regulatory frameworks. As modifications on existing services and systems are typically not possible, gateways are used to encapsulate the specifics of the regional infrastructures and regulations. These gateways perform a transformation of data schema and coding from regional format to a canonical format and vice versa and implement means to broker trust among the regions by acting as guarantors for the enforcement of agreed security services (e.g., on originator authenticity and proper authentication).

385
390 Health data sharing among autonomous regions is limited to an agreed set of documents because for example:

- many of the use cases of cross-regional care cover unscheduled care scenarios where a physician does not want to access the full EHR of a patient but is rather interested in aggregated health status information
- reimbursement regulations only cover specific phases of a treatment (e.g., Dutch patients being allowed to go to Germany for certain surgeries) that require access to be restricted to a defined set of documents

395
The XCF Profile provides simple access to documents of limited number and volume within a gateway infrastructure, where the initiating regions have simple environments.

29.4.1.2 Highly Regulated Data Sharing Scenarios

400 Two states are enabling access for their citizen’s emergency data-sets. The contents of the data set are well specified in advance, and only documents that are sanctioned will be accessed. A framework or community agreement needs to exist that governs, which documents, what contents and encoding are to be communicated under which conditions and environments.

405 Both states reserve the right to enforce policies at their respective domain and may not be forced to adapt or change their existing IT systems due to the principle of sovereignty.

Furthermore, only the most recent version of the emergency data-set is to be communicated at any time, potentially existing older version must not be communicated or made available for patient safety reasons: querying for just “any” document is disallowed.

29.4.2 Process Flow

410 Figure 29.4.2-1 shows a typical Cross-Community Fetch data access pattern where the XCF Profile can be used:

1. An initial requestor requests a defined kind of document about an already identified patient. The requestor connects to the Initiating Gateway using a mechanism not constrained by the XCF Profile. Examples of such mechanisms are filling a form at a web portal that is grouped with the Initiating Gateway or using a proprietary document type specific web service that is grouped with the Initiating Gateway (e.g., a dedicated “patient summary service”).

2. The Initiating Gateway initiates a XCF Fetch request message to the Responding Gateway will be able to fulfill the initial request.
- 420 3. The Responding Gateway processes the request message (e.g., enforcing community specific security policies) and obtains the requested information from any data managing actor within its community. The provided data is processed as previously agreed between the communicating communities (encoding, schema, etc.).
- 425 4. The Responding Gateway sends a XCF Fetch response message to the Initiating Gateway.
5. The Initiating Gateway verifies that the data provided confirm to the agreed policies. It processes the data to match its domain’s local policies and sends it to the Initial Requestor.

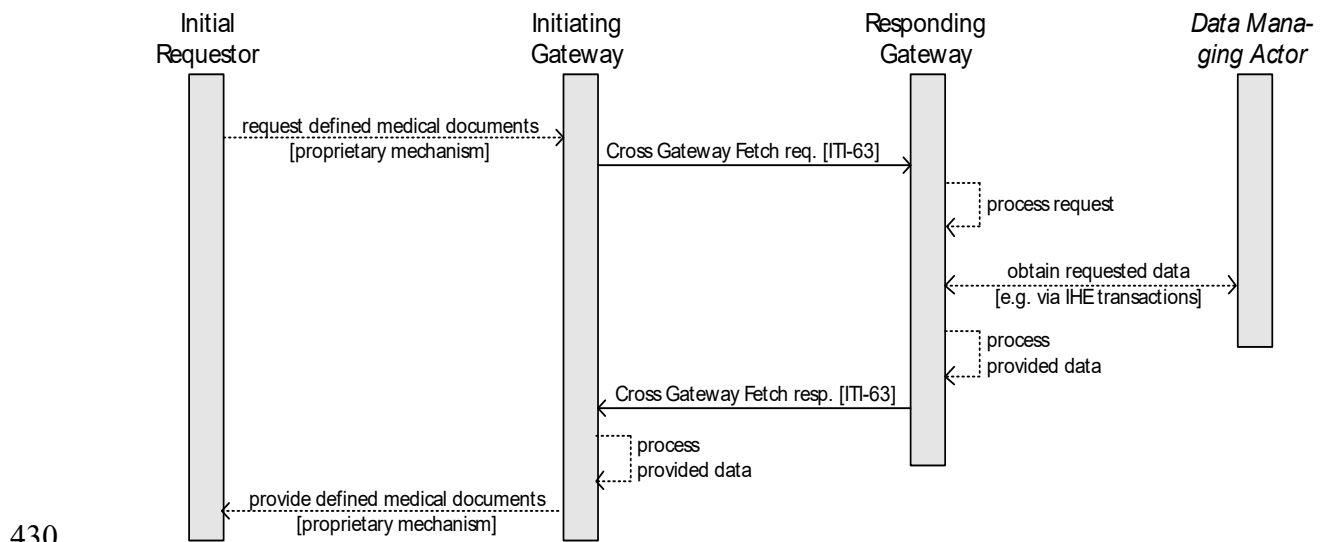


Figure 29.4.2-1: Basic Process Flow in XCF Profile

Figure 29.4.2-2 shows an illustration of this basic process flow described above. In this scenario, the Audit Trail and Node Authentication (ATNA) and IHE Cross-Enterprise User Assertion (XUA) Profiles are used to safeguard the communication and allowing the responding side to enforce fine-grained local security policies. Part of the initial request processing at the Responding Gateway is the verification that the patient has given consent to the use of his data for this purpose. For this purpose, the Responding Gateway may be grouped with a Document Consumer to obtain an IHE Basic Patient Privacy Consent (BPPC) coded consent document via XDS Registry Stored Query [ITI-18] and Retrieve Document Set [ITI-43] transactions.

435

440 If the responding community is organized as a XDS Affinity Domain, XDS can be used to obtain the requested data (see Section 29.3.2.1). In this case the Responding Gateway is grouped with a Document Consumer (which may be the same as used for consent retrieval) that initiates the XDS transactions for obtaining the requested data.

In advance, the communicating communities agreed on a canonical format for sharing documents. It is the responsibility of the Responding Gateway to transform, translate and transcode the data from its local format to the canonical format as agreed between the Initiating and Responding Gateways. The reverse action is taken at the Initiating Gateway: the data received from the Responding Gateway is transformed at the Initiating Gateway, translated and transcoded into the local format.

450

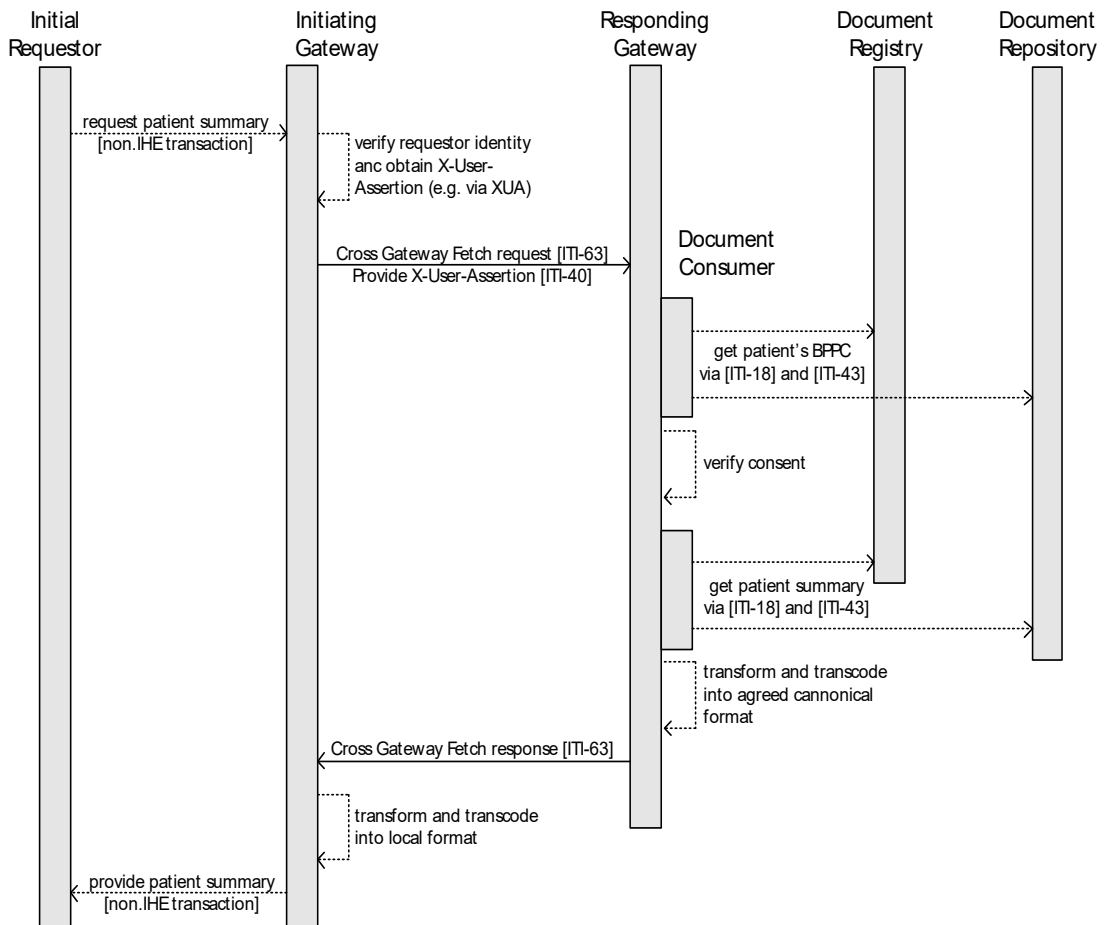


Figure 29.4.2-2: Process Flow with Consent/Policy Enforcement and Document Transcoding

29.5 XCF Profile Security Considerations

455 29.5.1 XCF Risk Assessment

The risk analysis for XCF enumerates assets, threats, and mitigations. The complete risk data may be found at ftp://ftp.ihe.net/IT_Infrastructure/iheit9-2011-2012/Technical_Cmte/Profile_Work/XCA-QueryRetrieve/XCFetch_Risk_assessment_and_mitigation_table_V1.xls.

460 This risk analysis extends the general IHE risks and threats analysis (see ITI TF-1: Appendix L)
for risks and mitigations that are specific to the XCF actors. Vendor and operators of XCF actors
are also advised that many risks cannot be mitigated by the IHE profile and instead the
responsibility for mitigation is transferred to the vendor, and occasionally to the communities
and enterprises that operate XCF gateways. In these instances, IHE fulfills its responsibility to
465 notify affected parties through the following section.

The following general mitigations shall be implemented by all XCF actors. These mitigations
moderate all currently known high impact risks. Implementers are strongly advised to
periodically reassess threats and mitigations to those threats, and to employ robust and secure
design, programming and operational management practices.

- 470 • In case that any or both of the gateways perform transcoding, transformation or
translation of metadata or document data, the following mitigation addresses the risks
associated with wrong transformations. The original (human) requestor should be given
the ability to additionally fetch an original document which is not automatically modified
during the Cross Gateway Fetch transaction. This may either be implemented through a
475 dedicated document class code for original data or by using the document relationship
mechanism as described in ITI TF-2b: 3.63.4.1.2.3.
- All actors in XCF shall be grouped with an ATNA Secure Node and a CT Time Client to,
respectively, ensure confidentiality and consistent logs.
- 480 • Document metadata shall include a hash of the document content to enable low/moderate
assurance document integrity confirmation. Use of document digital signatures (DSG)
may be used, if needed, for more high assurance document integrity and non-repudiation
of origin purposes.
- The Initiating Gateway should issue Cross Gateway Fetch requests that result in a single,
unambiguous, document to be found and returned whenever possible and must supply
485 both a patient identifier and a document class code identifier.
- To reduce the ability of attackers to “phish” for data, a Responding Gateway which
receives a fetch request for unknown patient identifiers or document class codes shall
return a response containing zero documents, with no further information. This applies to
patient identifiers and class code identifiers that are properly formatted or improperly
490 formatted.
- Initiating Gateways shall provide an X-user Assertion (XUA) with the Cross Gateway
Fetch request in order to allow the Responding Gateway to enforce a local security
policy. Responding Gateways should assess a local security policy before responding to
the request or retrieval of any metadata or data. The local security policy should include
495 the enforcement of a Basic Patient Privacy Policy (BPPC).
- Initiating Gateways may verify the patient identifier included with a received document’s
header against the original patient identifier that was used for the request.

General developmental and operational best practices should be observed.

Add the following new item to Appendix B

500

Appendix B – Transaction Summary Definitions

Cross Gateway Fetch - fetches a document or a set of documents from a remote community that matches a given set of metadata attribute values.

Volume 2b – Transactions

505 *Add Section 3.63*

3.63 Cross Gateway Fetch

This section corresponds to transaction [ITI-63] of the IHE ITI Technical Framework. Transaction [ITI-63] is used by the Initiating Gateway and Responding Gateway Actors.

3.63.1 Scope

510 This transaction is used to fetch a document or a set of documents that match a given set of metadata attribute values.

The transaction always returns:

- Metadata, if any, for zero or more registry objects, and
- Zero or more Association objects (linking the targeted DocumentEntries), and
- 515 • Document contents, if any.

3.63.2 Use Case Roles

Actor: Initiating Gateway

Role: Initiates the Cross Gateway Fetch transaction for obtaining a defined set of documents

Actor: Responding Gateway

520 **Role:** Responds to a Cross Gateway Fetch transaction by providing the registry data and document content of a defined set of documents

3.63.3 Referenced Standards

ebRIM	OASIS/ebXML Registry Information Model v3.0 This model defines the types of metadata and content that can be stored in an ebXML Registry, a basis for and subset of Document Sharing metadata.
ebRS	OASIS/ebXML Registry Services Specifications v3.0 This defines the services and protocols for an ebXML Registry, used as the basis for the XDS Document Registry
MTOM	SOAP Message Transmission Optimization Mechanism http://www.w3.org/TR/soap12-mtom/ This is a method for optimizing the transmission and/or wire format of SOAP messages.
XOP	XML-binary Optimized Packaging http://www.w3.org/TR/2005/REC-xop10-20050125/ . This is a means of more efficiently of converting an XML Infoset with certain types of content into a stream of bytes for transmission.
See ITI TF-2x: Appendix V for other referenced standards for SOAP encoding. See ITI TF-2a: 3.18 for the Registry Stored Query [ITI-18] transaction. See ITI TF-3: 4.2 for other referenced standards for metadata element encoding.	

3.63.4 Messages

525

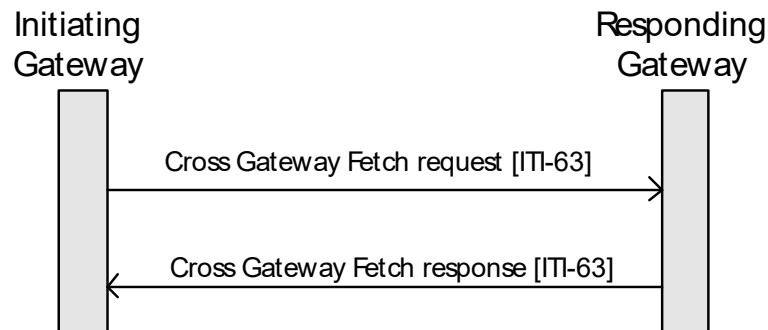


Figure 3.63.4-1: Interaction Diagram

3.63.4.1 Cross Gateway Fetch Request

530 The Cross Gateway Fetch request message is implemented as an ebRS Registry Stored Query with requesting the registry items and their linked repository items as a response. The Cross Gateway Fetch request message is fully compliant with the ebRS 3.0 standard. The request message shall use SOAP 1.2 MTOM with XOP encoded attachments.

3.63.4.1.1 Trigger Events

535 This message is initiated when the Initiating Gateway has determined that it must interact with the Responding Gateway to obtain a document as was requested from an initial requestor.

3.63.4.1.2 Message Semantics

The ebXML Registry stored query facility (Invoke Stored Query transaction) as profiled for the Cross Gateway Fetch request message shall contain the following parameters:

- 540 • **returnType** – shall be “LeafClassWithRepositoryItem” which specifies that the AdhocQueryResponse may contain a collection of ExtrinsicObject XML elements as defined in ebRIM Schema accompanied with their repository items
- **Query ID** – shall be "urn:uuid:f2072993-9478-41df-a603-8f016706efe8" which indicates a Fetch (which is an adaption of the findDocuments Query as defined in ITI TF-2a: 3.18.1)
- 545 • **Query Parameters** – as defined in the Query Parameters section below.

Other IHE stored query types as listed in ITI TF-2a: 3.18.1 are not defined by this transaction and the Responding Gateway may return an error

3.63.4.1.2.1 Query Parameters for Cross Gateway Fetch Requests

550 The following table lists the parameters that may be used for Cross Gateway Fetch requests. Parameters other than the ones listed below shall not be used.

Table 3.63.4.1.2.1-1: Query Parameters for Cross Gateway Fetch

Parameter Name	Description	Opt	Mult
\$XDSDocumentEntryPatientId	See ITI TF-3: 4.2.3.2-1	R	-
\$XDSDocumentEntryClassCode	See ITI TF-3: 4.2.3.2-1	R	M
\$XDSDocumentEntryTypeCode	See ITI TF-3: 4.2.3.2-1	O	M
\$XDSDocumentEntryPracticeSettingCode	See ITI TF-3: 4.2.3.2-1	O	M
\$XDSDocumentEntryCreationTimeFrom	See ITI TF-3: 4.2.3.2-1	O	-
\$XDSDocumentEntryCreationTimeTo	See ITI TF-3: 4.2.3.2-1	O	-
\$XDSDocumentEntryServiceStartTimeFrom	See ITI TF-3: 4.2.3.2-1	O	-
\$XDSDocumentEntryServiceStartTimeTo	See ITI TF-3: 4.2.3.2-1	O	-
\$XDSDocumentEntryServiceStopTimeFrom	See ITI TF-3: 4.2.3.2-1	O	-
\$XDSDocumentEntryServiceStopTimeTo	See ITI TF-3: 4.2.3.2-1	O	-
\$XDSDocumentEntryHealthcareFacilityTypeCode	See ITI TF-3: 4.2.3.2-1	O	M
\$XDSDocumentEntryEventCodeList ^{See Note 1}	See ITI TF-3: 4.2.3.2-1	O	M
\$XDSDocumentEntryConfidentialityCode ^{See Note 1}	See ITI TF-3: 4.2.3.2-1	O	M
\$XDSDocumentEntryAuthorPerson	See ITI TF-3: 4.2.3.2-1	O	M
\$XDSDocumentEntryFormatCode	See ITI TF-3: 4.2.3.2-1	O	M
homeCommunityId	See Section 3.63.4.1.2.2	R	-

Note 1: Supports AND/OR semantics as specified in ITI TF-2a: 3.18.4.1.2.3.5.

555 Coded values shall be coded according to specification in ITI TF-2a: 3.18.4.1.2.3.4 *Coding of Code/Code-Scheme*.

560 The value for the \$XDSDocumentEntryAuthorPerson parameter is a pattern compatible with the SQL keyword LIKE which allows the use of the following wildcard characters: % to match any (or no) characters and _ to match a single character. The match shall be applied to the text contained in the Value elements of the authorPerson Slot on the author Classification (value strings of the authorPerson sub-attribute).

3.63.4.1.2.2 Use of homeCommunityId

565 The Cross Gateway Fetch request shall contain the homeCommunityId, which is a globally unique identifier for a community and is used to obtain the Web Services endpoint of services that provide access to data in that community. homeCommunityId is structured as an OID limited to 64 characters and specified in URI syntax, for example the homeCommunityId of 1.2.3 would be formatted as urn:oid:1.2.3.

The use of homeCommunityId in conjunction with Cross Gateway Fetch is as follows:

- It is a parameter to Fetch requests

- 570
- The homeCommunityId value is specified as the home attribute on the AdhocQuery element of the Fetch request, as in: `<AdhocQuery id="..." home="urn:oid:1.2.3" ... >`
 - Each Fetch request shall only have one homeCommunityId value. Separate individual Fetch requests can be used to fetch data associated with different homeCommunityIds.

575 A Responding Gateway that receives a Cross Gateway Fetch request message shall behave as follows:

- If the homeCommunityId is an identifier for a community represented by the Responding Gateway, the Responding Gateway shall process the request.
 - If the homeCommunityId is an identifier for another community that is supported by the Responding Gateway, the Responding Gateway shall forward the request to the Responding Gateway that is responsible for that community.
 - If the value of homeCommunityId is not known by the Responding Gateway, the Responding Gateway shall return an XDSUnknownCommunity error code.
 - Verify the homeCommunityId is specified on the query and return an XDSMissingHomeCommunityId error code if missing.
- 580

585 3.63.4.1.2.3 Inclusion of Document Associations

In certain situations, such as patient safety reasons and local policy considerations, the Responding Gateway might be required to transport document associations alongside the requested documents. The ability to transport such additional information is explicitly supported by this transaction. Examples (non-exhaustive) of such associations can be an association to the original document that served as the basis for a transcoding/translation, an appended message such as a dispensation notice accompanying an ePrescription, or a special legal disclaimer required to be put in by the document provider.

590

The decision on what associations are to be included is based on the Responding Gateway's local policy and on the community agreement that sanctions the data transfer.

595 In any case, the Responding Gateway only returns associations between documents which are listed in the table at ITI TF-3: 4.2.2.2-1. It never returns other associations, for example hasMember.

3.63.4.1.3 Cross Gateway Fetch request message example

```

600 <query:AdhocQueryRequest>
    <query:ResponseOption returnComposedObjects="true"
        returnType="LeafClassWithRepositoryItem"/>
    <rim:AdhocQuery id="urn:uuid:f2072993-9478-41df-a603-8f016706efe8"
        home="2.16.17.710.780.1000">
        <!--Query slots go in here -->
605 </rim:AdhocQuery>
    </query:AdhocQueryRequest>

```


3.63.4.1.4 Expected Actions

The Responding Gateway shall:

1. Accept a parameterized query in an AdhocQueryRequest message
- 610 2. Verify the required parameters are included in the request
3. Process the query by discovering and fetching DocumentEntries matching the query request parameters.
4. If sanctioned by a community agreement, return appropriate Association objects linking the targeted DocumentEntries (targeted by the query request parameters).
- 615 5. Return an error and zero documents for the following conditions:
 - a) Unknown query ID
 - b) Required parameter missing
 - c) Invalid or unknown patient identifier
 - d) Unknown or missing homeCommunityId

620 See Section 3.63.4.2.5 for further error conditions and error message encoding see ITI TF-3: 4.3.4.2-2.

If a problem occurred while transcoding documents, a TranscodingError is issued which is included in the status return value of partial success (see Section 3.63.4.2.5).

- 625 1. Respond to the Cross Gateway Fetch request with a Cross Gateway Fetch response message (see Section 3.63.4.2)

3.63.4.2 Cross Gateway Fetch Response

The Cross Gateway Fetch response message includes the registry items (metadata) and documents listed in the registry items. The response message shall use SOAP 1.2 MTOM with XOP encoding attachments.

630 3.63.4.2.1 Trigger Events

The Cross Gateway Fetch response message is triggered by a Cross Gateway Fetch request message.

3.63.4.2.2 Message Semantics

635 The Cross Gateway Fetch response message semantics and syntax shall comply with ITI TF-2a: 3.18.4.1.2 with the following additions.

The response message syntax reuses the XML element `<Document/>` with namespace `urn:ihe:iti:xds-b:2007`. This element shall appear as the last element child of an `<ExtrinsicObject/>` element. The Document contents are associated with the `<DocumentEntry/>` (ExtrinsicObject) metadata by the fact that it is nested within the XML message. This format is considered the unoptimized format - the only one that can be represented

640

in pure XML. This is not the wire-format for the message but is what is specified by the schema and the WSDL (the XOP/MTOM optimization is applied afterwards).

The respective part of the <DocumentEntry/> metadata looks like this:

```
645 <rim:ExtrinsicObject>
      <!-- lots of stuff missing here -->
      <xdsect:Document xmlns:xdsect="urn:ihe:iti:xds-b:2007">
            VGhpcyBpcyBteSBkb2N1bWVudC4KCk10IGlzIGdyZWFOIQo=
      </xdsect:Document>
650 </rim:ExtrinsicObject>
```

The MTOM/XOP optimization of this content replaces the contents of the <Document/> element with a XOP reference to a different MIME part which holds the content. It is this moving of the bulky content out of the XML where it is difficult to handle and into the raw MIME multipart frame that is considered the optimization of MTOM/XOP. The resulting <Document/> element is depicted in ITI TF-2b: 3.43.5.1.2.

In addition to document metadata and document content, a Cross Gateway Fetch response may explicitly encode relationships among the documents provided. These are provided as Associations linking the DocumentEntries and placed into the <RegistryObjectList> element (see ITI TF-3: 4.2.2.2 for details).

As a result, the Cross Gateway Fetch response, if configured to do so (see Section 3.63.4.1.2.3), shall return document associations as sanctioned and explicitly approved by the local policy of the document or data provider.

A Cross Gateway Fetch response shall only contain those associations whose target and source objects are contained in the response.

A Cross Gateway Fetch response shall only return relationships between documents, any potential hasMember associations shall never be returned.

A Cross Gateway Fetch response shall not contain other objects than DocumentEntries and associations between DocumentEntries.

A Cross Gateway Fetch response shall contain the document contents for each DocumentEntry in the returned metadata.

3.63.4.2.3 Expected Actions

If the Cross Gateway Fetch Response is received by the Initiating Gateway shall:

- process the message received and make it available to the requestor

3.63.4.2.4 Document Metadata

Each provided document is further classified by metadata attributes (registry items associated with the document). The Responding Gateway shall provide document metadata attributes as

specified in ITI TF-3: Table 4.1.3.2-1. Metadata other than that included in these tables shall not be provided by the Responding Gateway and shall not be processed by the Initiating Gateway.

680 The classification schemes as defined in ITI TF-3: 4.2.5.2 shall be used.

3.63.4.2.5 Error Codes

Error conditions shall be covered according to ITI TF-3: 4.2.4.

685 Failures that originate in the SOAP header (including SAML assertions that are provided within the SOAP header) shall be covered by the respective error messages of the respective protocol or standard: if the Responding Gateway (acting as X-Service Provider) is unable to successfully process a X-User Assertion, it shall return an error code as described in WS-Security core specification section 12 (Error Handling, using the SOAP Fault mechanism), and the ATNA Audit event for authentication failure shall be returned according to ATNA rules (see ITI TF-2b: 3.40.4.1.3).

690 **3.63.5 Protocol Requirements**

The requirements for Web Services transport for Synchronous and WS-Addressing based Asynchronous are described in this section.

695 The Cross Gateway Fetch request and response will be transmitted using Synchronous or WS-Addressing based Asynchronous Web Services Exchange, according to the requirements specified in ITI TF-2x: Appendix V.3. The protocol requirements are identical to the Registry Stored Query except as noted below.

XML namespace prefixes are for informational purposes only and are documented in ITI TF-2x: Appendix V, Table V.2.4-1.

700 **Responding Gateway:** These are the requirements for Synchronous or WS-Addressing based Asynchronous Cross Gateway Fetch, presented in the order in which they would appear in the Responding Gateway WSDL definition:

- The following types shall be imported (xsd:import) in the /definitions/types section:
 - namespace=" urn:oasis:names:tc:ebxml-regrep:xsd:query:3.0",
schemaLocation="query.xsd"
- 705 • The /definitions/message/part/@element attribute of the Cross Gateway Fetch Request message shall be defined as “query:AdhocQueryRequest”
- The /definitions/message/part/@element attribute of the Cross Gateway Fetch Response message shall be defined as “query:AdhocQueryResponse”
- for additional attribute requirements refer to Table 3.63.5-1.

710 **Table 3.63.5-1: Attribute Requirements**

/definitions/portType/operation@name	RespondingGateway_CrossGatewayFetch
/definitions/portType/operation/input/@wsaw:Action	urn:ihe:iti:2011:CrossGatewayFetch

/definitions/portType/operation/output/@wsaw:Action	urn:ihe:iti:2011:CrossGatewayFetch
/definitions/binding/operation/soap12:operation/@soapActionRequired	false

These are the requirements that affect the wire format of the SOAP message. The other WSDL properties are only used within the WSDL definition and do not affect interoperability. For informative WSDL for the Responding Gateway see ITI TF-2x: Appendix W.

715 The Responding Gateway should:

- return either zero documents or XDSUnknownPatientId (if local policy permits) if unknown/invalid patient identifiers are provided in the request.
- return zero documents if no valid consent for the patient was found.
- return zero documents if privacy or security provisions are not met/violated.

720 • be configured for the maximum response size it supports. If the response exceeds this configured size, the Responding Gateway shall return XDSTooManyResults and zero documents.

3.63.6 Security Considerations

3.63.6.1 Security Audit Considerations

725 Both the Initiating Gateway and Responding Gateway shall audit the Cross Gateway Fetch transaction. The audit entries shall be equivalent to the entries required for the Registry Stored Query.

730 The Initiating Gateway shall audit the Cross Gateway Fetch as if it were a Document Consumer except that for EventTypeCode the Initiating Gateway shall specify EV(“ITI-63”, “IHE Transactions”, and “XCF Fetch”). See ITI TF-2a: 3.18.5.1.1.

The Responding Gateway shall audit the Cross Gateway Fetch as if it was a Document Registry except that for EventTypeCode the Responding Gateway shall specify EV(“ITI-63”, “IHE Transactions”, “XCF Fetch”). See ITI TF-2a: 3.18.5.1.2.

735 The Responding Gateway shall audit the creation of any intermediate format/object as if it was a Document Source except that for EventTypeCode, the Responding Gateway shall specify EV(“ITI-63”, “IHE Transactions”, “XCF Fetch Intermediate Document Creation”) and the EventActionCode shall be “C” for Create. See ITI TF-2b: 3.41.7.1.1.

3.63.7 Sample Request Message (Informative)

740

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" ... >
<soapenv:Header> ... </soapenv:Header>
<soapenv:Body>
<query:AdhocQueryRequest>
  <query:ResponseOption returnComposedObjects="true"
```

```

745         returnType="LeafClassWithRepositoryItem"/>
        <rim:AdhocQuery id="urn:uuid:f2072993-9478-41df-a603-8f016706efe8"
home="2.16.17.710.780.1000.990.1">
        <rim:Slot name="$XSDDocumentEntryPatientId">
750         <rim:ValueList>
            <rim:Value>
                'AT12998493069126^^^&amp;2.16.17.710.780.1000.990.1&amp;ISO'
            </rim:Value>
        </rim:ValueList>
        </rim:Slot>
755 <rim:Slot name="$XSDDocumentEntryClassCode">
        <rim:ValueList>
            <rim:Value>('57833-6^^2.16.840.1.113883.6.1')
            </rim:Value>
        </rim:ValueList>
760 </rim:Slot>
        </rim:AdhocQuery>
        </query:AdhocQueryRequest>
    </soapenv:Body>
</soapenv:Envelope>

```

765 3.63.8 Sample Response Message (Informative)

```

<?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
770 <env:Header xmlns:addressing="http://www.w3.org/2005/08/addressing"...</env:Header>
    <env:Body>
        <query:AdhocQueryResponse status="urn:oasis:names:tc:ebxml-
regrep:ResponseStatusType:Success" xmlns:query="urn:oasis:names:tc:ebxml-regrep:xsd:query:3.0">
            <ns1:RegistryObjectList xmlns:ns1="urn:oasis:names:tc:ebxml-regrep:xsd:rim:3.0">
775 <ns1:ExtrinsicObject home="2.16.17.710.780.1000.990.1" id="urn:uuid:283be5bb-2fda-4fc4-
bc06-eeb61efd5c7a" isOpaque="false" lid="urn:uuid:283be5bb-2fda-4fc4-bc06-eeb61efd5c7a"
mimeType="text/xml" objectType="urn:uuid:7edca82f-054d-47f2-a032-9b2a5b5186c1"
status="urn:oasis:names:tc:ebxml-regrep:StatusType:Approved">
                <ns1:Slot name="creationTime">
780 <ns1:ValueList>
                    <ns1:Value>20110311132002</ns1:Value>
                </ns1:ValueList>
            </ns1:Slot>
            <ns1:Slot name="hash">
785 <ns1:ValueList>
                    <ns1:Value>e4f92dadaa0316ee5379e6ed50e18e5f47a2eed8</ns1:Value>
                </ns1:ValueList>
            </ns1:Slot>
            <ns1:Slot name="languageCode">
790 <ns1:ValueList>
                    <ns1:Value>de-AT</ns1:Value>
                </ns1:ValueList>
            </ns1:Slot>
            <ns1:Slot name="legalAuthenticator">
                <ns1:ValueList>

```

```

795         <ns1:Value>admin^Admin^Spirit^^^Spirit Admin
User^^^^&2.16.17.710.780.1000.903.1.1.3.3&ISO</ns1:Value>
        </ns1:ValueList>
        </ns1:Slot>
        <ns1:Slot name="repositoryUniqueId">
800         <ns1:ValueList>
            <ns1:Value>2.16.17.710.780.1000.990.1</ns1:Value>
        </ns1:ValueList>
        </ns1:Slot>
        <ns1:Slot name="serviceStartTime">
805         <ns1:ValueList>
            <ns1:Value>20110311</ns1:Value>
        </ns1:ValueList>
        </ns1:Slot>
        <ns1:Slot name="serviceStopTime">
810         <ns1:ValueList>
            <ns1:Value>20110311</ns1:Value>
        </ns1:ValueList>
        </ns1:Slot>
        <ns1:Slot name="size">
815         <ns1:ValueList>
            <ns1:Value>21479</ns1:Value>
        </ns1:ValueList>
        </ns1:Slot>
        <ns1:Slot name="sourcePatientId">
820         <ns1:ValueList>
            <ns1:Value>AT12998493069126^^^&2.16.17.710.780.1000.990.1&ISO</ns1:Value>
        </ns1:ValueList>
        </ns1:Slot>
        <ns1:Slot name="sourcePatientInfo">
825         <ns1:ValueList>
            <ns1:Value>PID-
3|AT12998493069126^^^&2.16.17.710.780.1000.990.1&ISO</ns1:Value>
            <ns1:Value>PID-5|Barrel^Linda</ns1:Value>
            <ns1:Value>PID-7|19791105</ns1:Value>
830         </ns1:ValueList>
        </ns1:Slot>
        <ns1:Name>
            <ns1:LocalizedString charset="UTF-8" value="Patient Summary"/>
        </ns1:Name>
835         <ns1:Description>
            <ns1:LocalizedString charset="UTF-8" value="Patient Summary"/>
        </ns1:Description>
        <ns1:VersionInfo versionName="1"/>
        <ns1:Classification classificationScheme="urn:uuid:f33fb8ac-18af-42cc-ae0e-
840 ed0b0bdb91e1" classifiedObject="urn:uuid:283be5bb-2fda-4fc4-bc06-eeb61efd5c7a"
id="urn:uuid:3139b6b2-2fc4-442f-85f2-ddfc7a1f1fc1" lid="urn:uuid:3139b6b2-2fc4-442f-85f2-
ddfc7a1f1fc1" nodeRepresentation="not used" objectType="urn:oasis:names:tc:ebxml-
regrep:ObjectType:RegistryObject:Classification">
        <ns1:Slot name="codingScheme">
845         <ns1:ValueList>
            <ns1:Value>Connect-a-thon healthcareFacilityTypeCodes</ns1:Value>

```

```

      </nsl:ValueList>
    </nsl:Slot>
    <nsl:Name>
850     <nsl:LocalizedString charset="UTF-8" value="not used"/>
    </nsl:Name>
    <nsl:Description/>
    <nsl:VersionInfo versionName="1"/>
  </nsl:Classification>
855   <nsl:Classification classificationScheme="urn:uuid:a09d5840-386c-46f2-b5ad-
9c3699a4309d" classifiedObject="urn:uuid:283be5bb-2fda-4fc4-bc06-eeb61efd5c7a"
id="urn:uuid:35ec49c5-4cde-434a-a925-cd7d9473d0d4" lid="urn:uuid:35ec49c5-4cde-434a-a925-
cd7d9473d0d4" nodeRepresentation="urn:epSOS:ps:ps:2010" objectType="urn:oasis:names:tc:ebxml-
860   regrep:ObjectType:RegistryObject:Classification">
    <nsl:Slot name="codingScheme">
      <nsl:ValueList>
        <nsl:Value>epSOS formatCodes</nsl:Value>
      </nsl:ValueList>
    </nsl:Slot>
865   <nsl:Name>
      <nsl:LocalizedString charset="UTF-8" value="epSOS Patient Summary"/>
    </nsl:Name>
    <nsl:Description/>
    <nsl:VersionInfo versionName="1"/>
870   </nsl:Classification>
    <nsl:Classification classificationScheme="urn:uuid:ccc5598-8b07-4b77-a05e-
ae952c785ead" classifiedObject="urn:uuid:283be5bb-2fda-4fc4-bc06-eeb61efd5c7a"
id="urn:uuid:57932d1b-f515-497f-88f5-dbc92c33d6de" lid="urn:uuid:57932d1b-f515-497f-88f5-
875   dbc92c33d6de" nodeRepresentation="not used" objectType="urn:oasis:names:tc:ebxml-
regrep:ObjectType:RegistryObject:Classification">
    <nsl:Slot name="codingScheme">
      <nsl:ValueList>
        <nsl:Value>Connect-a-thon healthcareFacilityTypeCodes</nsl:Value>
      </nsl:ValueList>
880   </nsl:Slot>
    <nsl:Name>
      <nsl:LocalizedString charset="UTF-8" value="not used"/>
    </nsl:Name>
    <nsl:Description/>
885   <nsl:VersionInfo versionName="1"/>
  </nsl:Classification>
    <nsl:Classification classificationScheme="urn:uuid:f4f85eac-e6cb-4883-b524-
f2705394840f" classifiedObject="urn:uuid:283be5bb-2fda-4fc4-bc06-eeb61efd5c7a"
id="urn:uuid:82532349-f7e9-49b7-b0c0-fbeedeac789e" lid="urn:uuid:82532349-f7e9-49b7-b0c0-
890   fbeedeac789e" nodeRepresentation="N" objectType="urn:oasis:names:tc:ebxml-
regrep:ObjectType:RegistryObject:Classification">
    <nsl:Slot name="codingScheme">
      <nsl:ValueList>
        <nsl:Value>Connect-a-thon confidentialityCodes</nsl:Value>
895   </nsl:ValueList>
    </nsl:Slot>
    <nsl:Name>
      <nsl:LocalizedString charset="UTF-8" value="Normal"/>
    </nsl:Name>

```

```

900     <ns1:Description/>
        <ns1:VersionInfo versionName="1"/>
    </ns1:Classification>
    <ns1:Classification classificationScheme="urn:uuid:41a5887f-8865-4c09-adf7-
905 e362475b143a" classifiedObject="urn:uuid:283be5bb-2fda-4fc4-bc06-eeb61efd5c7a"
id="urn:uuid:8de3d828-2973-40e6-9e28-c9b7cecf3543" lid="urn:uuid:8de3d828-2973-40e6-9e28-
c9b7cecf3543" nodeRepresentation="60591-5" objectType="urn:oasis:names:tc:ebxml-
regrep:ObjectType:RegistryObject:Classification">
    <ns1:Slot name="codingScheme">
        <ns1:ValueList>
910     <ns1:Value>2.16.840.1.113883.6.1</ns1:Value>
        </ns1:ValueList>
    </ns1:Slot>
    <ns1:Name>
        <ns1:LocalizedString charset="UTF-8" value="Patient Summary"/>
915 </ns1:Name>
    <ns1:Description/>
    <ns1:VersionInfo versionName="1"/>
    </ns1:Classification>
    <ns1:Classification classificationScheme="urn:uuid:93606bcf-9494-43ec-9b4e-
920 a7748d1a838d" classifiedObject="urn:uuid:283be5bb-2fda-4fc4-bc06-eeb61efd5c7a"
id="urn:uuid:ef5706fe-dac9-41b2-873a-adbbc119a14d" lid="urn:uuid:ef5706fe-dac9-41b2-873a-
adbbc119a14d" nodeRepresentation="" objectType="urn:oasis:names:tc:ebxml-
regrep:ObjectType:RegistryObject:Classification">
    <ns1:Slot name="authorInstitution">
925     <ns1:ValueList>
        <ns1:Value>spirit^^^^1.2.40.0.32.6.1.10&amp;ISO^^^2.16.17.710</ns1:Value>
        </ns1:ValueList>
    </ns1:Slot>
    <ns1:Slot name="authorPerson">
930     <ns1:ValueList>
        <ns1:Value>admin^Admin^Spirit^^^Spirit Admin
User^^^^&amp;2.16.17.710.780.1000.903.1.1.3.&amp;ISO</ns1:Value>
        </ns1:ValueList>
    </ns1:Slot>
935 </ns1:Name/>
    <ns1:Description/>
    <ns1:VersionInfo versionName="1"/>
    </ns1:Classification>
    <ns1:Classification classificationScheme="urn:uuid:f0306f51-975f-434e-a61c-
940 c59651d33983" classifiedObject="urn:uuid:283be5bb-2fda-4fc4-bc06-eeb61efd5c7a"
id="urn:uuid:fc3c09f6-2d49-4fc5-bba1-delf4f34f59a" lid="urn:uuid:fc3c09f6-2d49-4fc5-bba1-
delf4f34f59a" nodeRepresentation="60591-5" objectType="urn:oasis:names:tc:ebxml-
regrep:ObjectType:RegistryObject:Classification">
    <ns1:Slot name="codingScheme">
945     <ns1:ValueList>
        <ns1:Value>2.16.840.1.113883.6.1</ns1:Value>
        </ns1:ValueList>
    </ns1:Slot>
    <ns1:Name>
950     <ns1:LocalizedString charset="UTF-8" value="Patient Summary"/>
    </ns1:Name>
    <ns1:Description/>
    <ns1:VersionInfo versionName="1"/>

```



```

955     </ns1:Classification>
        <ns1:ExternalIdentifier id="urn:uuid:b85f024c-0e7c-4c1d-a409-fb8e5010e235"
identificationScheme="urn:uuid:2e82c1f6-a085-4c72-9da3-8640a32e42ab" lid="urn:uuid:b85f024c-0e7c-
4c1d-a409-fb8e5010e235" objectType="urn:oasis:names:tc:ebxml-
960     regrep:ObjectType:RegistryObject:ExternalIdentifier" registryObject="urn:uuid:283be5bb-2fda-4fc4-
bc06-eeb61efd5c7a" value="2.16.17.710.780.1000.902.1.1.3.2^BE978FEFD3BB789">
        <ns1:Name>
          <ns1:LocalizedString charset="UTF-8" value="XDSDocumentEntry.uniqueId"/>
        </ns1:Name>
        <ns1:Description/>
        <ns1:VersionInfo versionName="1"/>
965     </ns1:ExternalIdentifier>
        <ns1:ExternalIdentifier id="urn:uuid:349a345c-831a-4215-95f9-99bf5a8c21b6"
identificationScheme="urn:uuid:58a6f841-87b3-4a3e-92fd-a8ffeff98427" lid="urn:uuid:349a345c-831a-
4215-95f9-99bf5a8c21b6" objectType="urn:oasis:names:tc:ebxml-
970     regrep:ObjectType:RegistryObject:ExternalIdentifier" registryObject="urn:uuid:283be5bb-2fda-4fc4-
bc06-eeb61efd5c7a" value="AT12998493069126^^^&2.16.17.710.780.1000.990.1&ISO">
        <ns1:Name>
          <ns1:LocalizedString charset="UTF-8" value="XDSDocumentEntry.patientId"/>
        </ns1:Name>
        <ns1:Description/>
975     <ns1:VersionInfo versionName="1"/>
        </ns1:ExternalIdentifier>

        <ns1:Document>
          <Include href="cid: BE978FEFD3BB789 "
980     xmlns="http://www.w3.org/2004/08/xop/include"/>
          UjBsR09EbGhJZ0dTQUxNQUPBUUNBRU1tQ1p0dU1GUXhEUzhi.....
        </ns1:Document>
        </ns1:ExtrinsicObject>
        <ns1:ExtrinsicObject home="2.16.17.710.780.1000.990.1" id="urn:uuid:c9b9912b-d715-49d3-
985     8348-e23d63463d90" isOpaque="false" lid="urn:uuid:c9b9912b-d715-49d3-8348-e23d63463d90"
mimeType="text/xml" objectType="urn:uuid:7edca82f-054d-47f2-a032-9b2a5b5186c1"
status="urn:oasis:names:tc:ebxml-regrep:StatusType:Approved">
        <ns1:Slot name="creationTime">
          <ns1:ValueList>
990     <ns1:Value>20110311132002</ns1:Value>
          </ns1:ValueList>
        </ns1:Slot>
        <ns1:Slot name="hash">
          <ns1:ValueList>
995     <ns1:Value>509658697e16988640f9056c05076dce0757b521</ns1:Value>
          </ns1:ValueList>
        </ns1:Slot>
        <ns1:Slot name="languageCode">
          <ns1:ValueList>
1000     <ns1:Value>de-AT</ns1:Value>
          </ns1:ValueList>
        </ns1:Slot>
        <ns1:Slot name="legalAuthenticator">
          <ns1:ValueList>
1005     <ns1:Value>admin^Admin^Spirit^^^Spirit Admin
User^^^&2.16.17.710.780.1000.903.1.1.3.3&ISO</ns1:Value>
          </ns1:ValueList>

```

```

1010     </ns1:Slot>
        <ns1:Slot name="repositoryUniqueId">
            <ns1:ValueList>
                <ns1:Value>2.16.17.710.780.1000.990.1</ns1:Value>
            </ns1:ValueList>
        </ns1:Slot>
1015     <ns1:Slot name="serviceStartTime">
            <ns1:ValueList>
                <ns1:Value>20110311</ns1:Value>
            </ns1:ValueList>
        </ns1:Slot>
1020     <ns1:Slot name="serviceStopTime">
            <ns1:ValueList>
                <ns1:Value>20110311</ns1:Value>
            </ns1:ValueList>
        </ns1:Slot>
1025     <ns1:Slot name="size">
            <ns1:ValueList>
                <ns1:Value>106426</ns1:Value>
            </ns1:ValueList>
        </ns1:Slot>
1030     <ns1:Slot name="sourcePatientId">
            <ns1:ValueList>
                <ns1:Value>AT12998493069126^^^&amp;2.16.17.710.780.1000.990.1&amp;ISO</ns1:Value>
            </ns1:ValueList>
        </ns1:Slot>
1035     <ns1:Slot name="sourcePatientInfo">
            <ns1:ValueList>
                <ns1:Value>PID-
3|AT12998493069126^^^&amp;2.16.17.710.780.1000.990.1&amp;ISO</ns1:Value>
                <ns1:Value>PID-5|Barrel^Linda</ns1:Value>
                <ns1:Value>PID-7|19791105</ns1:Value>
            </ns1:ValueList>
1040     </ns1:Slot>
        <ns1:Name>
            <ns1:LocalizedString charset="UTF-8" value="Patient Summary"/>
        </ns1:Name>
1045     <ns1:Description>
            <ns1:LocalizedString charset="UTF-8" value="Patient Summary"/>
        </ns1:Description>
        <ns1:VersionInfo versionName="1"/>
        <ns1:Classification classificationScheme="urn:uuid:f0306f51-975f-434e-a61c-
1050     c59651d33983" classifiedObject="urn:uuid:c9b9912b-d715-49d3-8348-e23d63463d90"
            id="urn:uuid:14af14d5-d9ab-483d-890c-ae10e31efed3" lid="urn:uuid:14af14d5-d9ab-483d-890c-
            ae10e31efed3" nodeRepresentation="60591-5" objectType="urn:oasis:names:tc:ebxml-
            regrep:ObjectType:RegistryObject:Classification">
1055     <ns1:Slot name="codingScheme">
            <ns1:ValueList>
                <ns1:Value>2.16.840.1.113883.6.1</ns1:Value>
            </ns1:ValueList>
        </ns1:Slot>
        <ns1:Name>

```

```
1060         <ns1:LocalizedString charset="UTF-8" value="Patient Summary"/>
        </ns1:Name>
        <ns1:Description/>
        <ns1:VersionInfo versionName="1"/>
        </ns1:Classification>
1065     <ns1:Classification classificationScheme="urn:uuid:f4f85eac-e6cb-4883-b524-
        f2705394840f" classifiedObject="urn:uuid:c9b9912b-d715-49d3-8348-e23d63463d90"
        id="urn:uuid:19944c6f-3f70-4f0b-9d5b-f1dfa616dfb4" lid="urn:uuid:19944c6f-3f70-4f0b-9d5b-
        f1dfa616dfb4" nodeRepresentation="N" objectType="urn:oasis:names:tc:ebxml-
1070     regrep:ObjectType:RegistryObject:Classification">
        <ns1:Slot name="codingScheme">
        <ns1:ValueList>
            <ns1:Value>Connect-a-thon confidentialityCodes</ns1:Value>
        </ns1:ValueList>
        </ns1:Slot>
1075     <ns1:Name>
        <ns1:LocalizedString charset="UTF-8" value="Normal"/>
        </ns1:Name>
        <ns1:Description/>
        <ns1:VersionInfo versionName="1"/>
1080     </ns1:Classification>
        <ns1:Classification classificationScheme="urn:uuid:a09d5840-386c-46f2-b5ad-
        9c3699a4309d" classifiedObject="urn:uuid:c9b9912b-d715-49d3-8348-e23d63463d90"
        id="urn:uuid:27536806-1220-488a-8882-80f23c451fb9" lid="urn:uuid:27536806-1220-488a-8882-
1085     80f23c451fb9" nodeRepresentation="urn:ihe:iti:xds-sd:pdf:2008"
        objectType="urn:oasis:names:tc:ebxml-regrep:ObjectType:RegistryObject:Classification">
        <ns1:Slot name="codingScheme">
        <ns1:ValueList>
            <ns1:Value>epSOS formatCodes</ns1:Value>
        </ns1:ValueList>
1090     </ns1:Slot>
        <ns1:Name>
        <ns1:LocalizedString charset="UTF-8" value="Scanned Documents PDF"/>
        </ns1:Name>
        <ns1:Description/>
1095     <ns1:VersionInfo versionName="1"/>
        </ns1:Classification>
        <ns1:Classification classificationScheme="urn:uuid:f33fb8ac-18af-42cc-ae0e-
        ed0b0bdb91e1" classifiedObject="urn:uuid:c9b9912b-d715-49d3-8348-e23d63463d90"
        id="urn:uuid:3d2c298a-f4c0-4ed4-803d-ad2a9698cd95" lid="urn:uuid:3d2c298a-f4c0-4ed4-803d-
1100     ad2a9698cd95" nodeRepresentation="not used" objectType="urn:oasis:names:tc:ebxml-
        regrep:ObjectType:RegistryObject:Classification">
        <ns1:Slot name="codingScheme">
        <ns1:ValueList>
            <ns1:Value>Connect-a-thon healthcareFacilityTypeCodes</ns1:Value>
1105     </ns1:ValueList>
        </ns1:Slot>
        <ns1:Name>
        <ns1:LocalizedString charset="UTF-8" value="not used"/>
        </ns1:Name>
1110     <ns1:Description/>
        <ns1:VersionInfo versionName="1"/>
        </ns1:Classification>
```

```

1115     <ns1:Classification classificationScheme="urn:uuid:ccc5598-8b07-4b77-a05e-
ae952c785ead" classifiedObject="urn:uuid:c9b9912b-d715-49d3-8348-e23d63463d90"
id="urn:uuid:9546ea5b-9a8b-4214-98df-9107d31a13e4" lid="urn:uuid:9546ea5b-9a8b-4214-98df-
9107d31a13e4" nodeRepresentation="not used" objectType="urn:oasis:names:tc:ebxml-
regrep:ObjectType:RegistryObject:Classification">
1120         <ns1:Slot name="codingScheme">
<ns1:ValueList>
<ns1:Value>Connect-a-thon healthcareFacilityTypeCodes</ns1:Value>
</ns1:ValueList>
</ns1:Slot>
<ns1:Name>
1125     <ns1:LocalizedString charset="UTF-8" value="not used"/>
</ns1:Name>
<ns1:Description/>
<ns1:VersionInfo versionName="1"/>
</ns1:Classification>
1130     <ns1:Classification classificationScheme="urn:uuid:93606bcf-9494-43ec-9b4e-
a7748d1a838d" classifiedObject="urn:uuid:c9b9912b-d715-49d3-8348-e23d63463d90"
id="urn:uuid:9837631b-d5b7-43b3-b646-81b71c782695" lid="urn:uuid:9837631b-d5b7-43b3-b646-
81b71c782695" nodeRepresentation="" objectType="urn:oasis:names:tc:ebxml-
regrep:ObjectType:RegistryObject:Classification">
1135         <ns1:Slot name="authorInstitution">
<ns1:ValueList>
<ns1:Value>spirit^^^^1.2.40.0.32.6.1.10&amp;ISO^^^2.16.17.710</ns1:Value>
</ns1:ValueList>
</ns1:Slot>
1140     <ns1:Slot name="authorPerson">
<ns1:ValueList>
<ns1:Value>admin^Admin^Spirit^^^Spirit Admin
User^^^^&amp;2.16.17.710.780.1000.903.1.1.3.3&amp;ISO</ns1:Value>
</ns1:ValueList>
</ns1:Slot>
1145     <ns1:Name/>
<ns1:Description/>
<ns1:VersionInfo versionName="1"/>
</ns1:Classification>
1150     <ns1:Classification classificationScheme="urn:uuid:41a5887f-8865-4c09-adf7-
e362475b143a" classifiedObject="urn:uuid:c9b9912b-d715-49d3-8348-e23d63463d90"
id="urn:uuid:fb406f62-e002-4aa4-948c-90a959548f6d" lid="urn:uuid:fb406f62-e002-4aa4-948c-
90a959548f6d" nodeRepresentation="60591-5" objectType="urn:oasis:names:tc:ebxml-
regrep:ObjectType:RegistryObject:Classification">
1155         <ns1:Slot name="codingScheme">
<ns1:ValueList>
<ns1:Value>2.16.840.1.113883.6.1</ns1:Value>
</ns1:ValueList>
</ns1:Slot>
1160     <ns1:Name>
<ns1:LocalizedString charset="UTF-8" value="Patient Summary"/>
</ns1:Name>
<ns1:Description/>
<ns1:VersionInfo versionName="1"/>
</ns1:Classification>
1165     <ns1:ExternalIdentifier id="urn:uuid:f3e1201b-faca-4f25-b40d-ab85fd3fcb89"
identificationScheme="urn:uuid:2e82c1f6-a085-4c72-9da3-8640a32e42ab" lid="urn:uuid:f3e1201b-faca-

```

```

4f25-b40d-ab85fd3fcb89" objectType="urn:oasis:names:tc:ebxml-
1170   regrep:ObjectType:RegistryObject:ExternalIdentifier" registryObject="urn:uuid:c9b9912b-d715-49d3-
      8348-e23d63463d90" value="2.16.17.710.780.1000.902.1.1.3.2^1F384DBE2BC97E1">
        <ns1:Name>
          <ns1:LocalizedString charset="UTF-8" value="XSDDocumentEntry.uniqueId"/>
        </ns1:Name>
        <ns1:Description/>
        <ns1:VersionInfo versionName="1"/>
1175   </ns1:ExternalIdentifier>
        <ns1:ExternalIdentifier id="urn:uuid:5ce61557-3b49-4950-9435-e40b4c216eb0"
identificationScheme="urn:uuid:58a6f841-87b3-4a3e-92fd-a8ffeff98427" lid="urn:uuid:5ce61557-3b49-
4950-9435-e40b4c216eb0" objectType="urn:oasis:names:tc:ebxml-
1180   regrep:ObjectType:RegistryObject:ExternalIdentifier" registryObject="urn:uuid:c9b9912b-d715-49d3-
      8348-e23d63463d90" value="AT12998493069126^^^&2.16.17.710.780.1000.990.1&ISO">
        <ns1:Name>
          <ns1:LocalizedString charset="UTF-8" value="XSDDocumentEntry.patientId"/>
        </ns1:Name>
        <ns1:Description/>
        <ns1:VersionInfo versionName="1"/>
1185   </ns1:ExternalIdentifier>
        <ns1:Document>
          <Include href="cid:1F384DBE2BC97E1"
1190   xmlns="http://www.w3.org/2004/08/xop/include"/>
        </ns1:Document>
      </ns1:ExtrinsicObject>
        <ns1:Association associationType="urn:ihe:iti:2007:AssociationType:XFRM"
id="urn:uuid:a193b283-fbd6-40ca-8de8-fc804e7659ce" lid="urn:uuid:a193b283-fbd6-40ca-8de8-
fc804e7659ce" objectType="urn:oasis:names:tc:ebxml-regrep:ObjectType:RegistryObject:Association"
1195   sourceObject="urn:uuid:c9b9912b-d715-49d3-8348-e23d63463d90" status="urn:oasis:names:tc:ebxml-
      regrep:StatusType:Approved" targetObject="urn:uuid:283be5bb-2fda-4fc4-bc06-eeb61efd5c7a">
        <ns1:Name/>
        <ns1:Description/>
        <ns1:VersionInfo versionName="1"/>
1200   </ns1:Association>
      </ns1:RegistryObjectList>
    </query:AdhocQueryResponse>
  </env:Body>
</env:Envelope>

```

1205

Volume 3 – Cross-Transaction Specifications and Content Specifications

ITI TF-3: Update Error Reporting section of the ITI TF-3 Table 4.2.4.1-2.

1210

Table 4.2.4.1-2: Error Codes (previously Table 4.1-11)

Error Code	Discussion	Transaction (See Note 1)
...	...	
PartialTransformReplaceNotProcessed	An XDR Document Recipient did not process some part of the content. Specifically the parts not processed are Transform and Replace semantics	P
<u>TranscodingError</u>	<u>The requested document cannot be provided due to a transcoding / translation error.</u>	<u>XGF</u>
UnresolvedReferenceException	The recipient cannot resolve an entryUUID reference in the transaction.	P, R
...		
XDSMissingHomeCommunityId	A value for the homeCommunityId is required and has not been specified	SQ, XGQ, RS, XGR, <u>XGF</u>
...		
XDSStoredQueryMissingParam	A required parameter to a stored query is missing.	SQ, XGQ, <u>XGF</u>
XDSStoredQueryParamNumber	A parameter which only accepts a single value is coded with multiple values	SQ, XGQ, <u>XGF</u>
XDSTooManyResults	Query resulted in too many results. No results are returned.	SQ, XGQ, <u>XGF</u>
...		
XDSUnknownPatientId	Patient Id referenced in metadata is not known by the receiving actor. The codeContext shall include the value of patient Id in question.	P,R, XGQ, <u>XGF</u> Note: this error code is not used in the response to Registry Stored Query
...		
XDSUnknownStoredQuery	The Query Id provided in the request is not recognized.	SQ, XGQ, <u>XGF</u>
...		

Note 1:

P = Provide and Register-b

R = Register-b

1215 SQ = Stored Query
RS = Retrieve Document Set
XGQ = Cross Gateway Query
XGR = Cross Gateway Retrieve
XGF = Cross Gateway Fetch

...

1220

<i>Update ITI TF-3: Table 4.2.4.2-3 table heading to add the Cross Gateway Fetch</i>
--

Table 4.2.4.2-3: [ITI-18] Stored Query, [ITI-63] Cross Gateway Fetch, and [ITI-38] Cross Gateway Query Responses