Integrating the Healthcare Enterprise

IHE IT Infrastructure (ITI) Technical Framework Supplement 2008-2009

Cross-Community Access XCA

Draft for Trial Implementation

October 10, 2008

Contents

Foreword

Integrating the Healthcare Enterprise (IHE) is an initiative designed to stimulate the integration of the information systems that support modern healthcare institutions. Its fundamental objective is to ensure that in the care of patients all required information for medical decisions is both correct and available to healthcare professionals. The IHE initiative is both a process and a forum for encouraging integration efforts. It defines a technical framework for the implementation of established messaging standards to achieve specific clinical goals. It includes a rigorous testing process for the implementation of this framework. And it organizes educational sessions and exhibits at major meetings of medical professionals to demonstrate the benefits of this framework and encourage its adoption by industry and users.

The approach employed in the IHE initiative is not to define new integration standards, but rather to support the use of existing standards, HL7, DICOM, IETF, and others, as appropriate in their respective domains in an integrated manner, defining configuration choices when necessary. IHE maintain formal relationships with several standards bodies including HL7, DICOM and refers recommendations to them when clarifications or extensions to existing standards are necessary.

This initiative has numerous sponsors and supporting organizations in different medical specialty domains and geographical regions. In North America the primary sponsors are the Healthcare Information and Management Systems Society (HIMSS) and the Radiological Society of North America (RSNA). IHE Canada has also been formed. IHE Europe (IHE-EUR) is supported by a large coalition of organizations including the European Association of Radiology (EAR) and European Congress of Radiologists (ECR), the Coordination Committee of the Radiological and Electromedical Industries (COCIR), Deutsche Röntgengesellschaft (DRG), the EuroPACS Association, Groupement pour la Modernisation du Système d'Information Hospitalier (GMSIH), Société Francaise de Radiologie (SFR), Società Italiana di Radiologia Medica (SIRM), the European Institute for health Records (EuroRec), and the European Society of Cardiology (ESC). In Japan IHE-J is sponsored by the Ministry of Economy, Trade, and Industry (METI); the Ministry of Health, Labor, and Welfare; and MEDIS-DC; cooperating organizations include the Japan Industries Association of Radiological Systems (JIRA), the Japan Association of Healthcare Information Systems Industry (JAHIS), Japan Radiological Society (JRS), Japan Society of Radiological Technology (JSRT), and the Japan Association of Medical Informatics (JAMI). Other organizations representing healthcare professionals are invited to join in the expansion of the IHE process across disciplinary and geographic boundaries.

The IHE Technical Frameworks for the various domains (IT Infrastructure, Cardiology, Laboratory, Radiology, etc.) defines specific implementations of established standards to achieve integration goals that promote appropriate sharing of medical information to support optimal patient care. It is expanded annually, after a period of public review, and maintained regularly through the identification and correction of errata. The current version for these Technical Frameworks may be found at www.ihe.net/Technical_Framework.

The IHE Technical Framework identifies a subset of the functional components of the healthcare enterprise, called IHE Actors, and specifies their interactions in terms of a set of coordinated, standards-based transactions. It describes this body of transactions in progressively greater depth. The volume I provides a high-level view of IHE functionality, showing the transactions organized into functional units called Integration Profiles that highlight their capacity to address specific clinical needs. The subsequent volumes provide detailed technical descriptions of each IHE transaction.

This IHE IT Infrastructure Technical Framework Supplement is issued for Trial Implementation through April 2009.

Comments and change proposals arising from Trial Implementation may be submitted to http://forums.rsna.org under the forum:

"Integrating the Healthcare Enterprise"

Select the sub-forum:

"IHE IT Infrastructure 2008 Supplement for Trial Implementation"

Comment [LF1]: Note that the forum for this year's trial implementation comments has not yet been set up (is this still the process we use anyway? I noticed no comments were submitted in forums for last year's supplements)

The IHE IT Infrastructure Technical Committee will address these comments resulting from implementation, connectathon testing, and demonstrations such as HIMSS 2009. Final text is expected to be published in August 2009**.**

Introduction

The Cross-Community Access profile supports the means to query and retrieve patient relevant medical data held by other communities. A community is defined as a coupling of facilities/enterprises that have agreed to work together using a common set of policies for the purpose of sharing clinical information via an established mechanism. Facilities/enterprises may host any type of healthcare application such as EHR, PHR, etc. A community is identifiable by a globally unique id called the homeCommunityId. Membership of a facility/enterprise in one community does not preclude it from being a member in another community. Such communities may be XDS Affinity Domains which define document sharing using the XDS profile or any other communities, no matter what their internal sharing structure.

This profile is compatible with the XDS.b supplement released in the same year. It is not compatible and will not be retrofitted to the XDS.a transactions where they are different from XDS.b. The significant effect of this is that this profile does not support the XDS.a style retrieve (HTTP GET) but only the XDS.b retrieve which relies on MTOM. Please refer to the XDS.b supplement for more information.

In 2006, a White Paper was developed by the ITI Technical Committee proposing a set of profiles that should be developed to allow cross-community patient data access. This profile is the Cross-Community Access profile described in the White Paper. For more details please refer to the Cross-Community White Paper which can be accessed at http://www.ihe.net/Technical_Framework/index.cfm#IT.

1.1 Open Issues and Questions

XCA002: Support significant delay in response - The current XDS Stored Query does not allow the requester to specify a timeout value for a time frame where a subset of data is returned. The requestor has the choice of aborting an existing request, thus getting nothing, or waits for as long as it takes to get everything. For example, a requestor may want to get as many results as possible in one minute. Current support allows for canceling the request at one minute - hence getting nothing - or waiting for as long as it takes to get everything. In a multiple Affinity Domain environment it is perceived to be more important that the requestor can specify a time frame to receive partial results. It has been suggested that this is a problem that does not need to be solved. The VA has collected statistics regarding its multi-domain healthcare system which states that the average delay is 6-7 seconds as measured at the initiating gateway.

It has been suggested that performance improvements result if the gateways involved in the transport are forwarding data as it is received, rather than collecting it locally and forwarding once all is received. The underlying transport (i.e. HTTP or equivalent) should support the ability for the gateway to send blocks of data as it is received, even though the higher level transport (SOAP) sees this as a single synchronous response.

Supporting an asynchronous mode for the transactions is another solution to consider. In this case the requestor would not have to wait for the response, but would be notified later once the complete response was collected. This has some performance benefits but does not solve the user's problem of having to wait for all the data to arrive. The delay is still experienced by the end user (or application).

XCA012: Asynchronous transactions: Should the query and retrieve transactions have an asynchronous mode? The following use cases related to this have been considered:

1. Support large volume data – see closed issue XCA001
2. Support significant delay in response – see open issue XCA002
3. Support messaging systems which prefer to operate only in an asynchronous mode

Addressing the last use case (others addressed under other issue items) it suggests that if asynchronous messaging was supported this would allow a class of systems to make use of the transactions defined in this profile. Support for a single request with a single response pattern is well described in several places. What requires greater specification would be to support a single request followed by a series of responses. A detailed description of the technical steps of a standards based approach for doing this has not agreed to nor identified in detail. To support requirements #1 and #2 support for multiple responses is required. If the only required use case is #3 then support for a request followed by a single response asynchronous message pattern would satisfy the requirements. Further analysis of the requirements is needed to ensure that the solution satisfies. Given the problems to be addressed and in order to gain more understanding of the use of this profile it is proposed that defining an asynchronous transport for the transactions be deferred for future year work.

XCA015: Specifying homeCommunityId on query: The Registry Stored Query syntax is dependent on ebXRS query support, which defines a schema where parameters are specified in slots. Our desire is to define a way for a set of pairs, where each uniqueID/entryUUID is grouped with a homeCommunityId. Several techniques for this were considered but none stood out as a clear direction. There is also Change Proposal work expected next year that may show a direction for this type of problem. For all these reasons, we have decided that for this season there can only be one homeCommunityId per query. Thus Document Consumers must break up queries to multiple homeCommunityId values into multiple query instances. We assess this restriction as a minor one and expect that as the query syntax is modified to support more complex queries this problem will also be solved.

1.2 Closed Issues

XCA001: Support large volume data - The current XDS Stored Query assumes all data is returned in one synchronous response. It is expected that queries spanning multiple Affinity Domains will result in larger sets of data and it is a requirement that the requestor be able to request that results be chunked in the response. The ebRS Stored Query "Query Iteration" function was considered to accommodate this requirement. While this functionality solves part of the requirement there were many issues that the standard did not resolve. After extensive discussion it was agreed that chunking was already supported by the Stored Query Transaction in two ways:

- A find documents query can request only object references and then retrieve objects separately. This allows for a paging ability where the first request gets all references and subsequent requests get a "page" at a time.
- A find documents query can be refined to select a smaller group of documents, either a smaller time range or restriction on document type or other class of documents.

If a request is received which is too big the gateway processing the request will return an error value which will indicate that the request returned too many records. This will be returned to the requestor who can then refine the query so fewer records are returned. Because chunking algorithms are complicated it is agreed that it does not make sense to design a solution until a clear, demonstrated use case experienced in a real implementation is defined.

XCA003: Ordering of query response data - The XDS Stored Query does not specify any ordering to the query results. Ordering of results is seen as very important but technically difficult in a cross XDS Affinity Domain environment. To support a sorted query result the gateway must get all results from all XDS Affinity Domains and perform the sort against all results before anything is returned. This will result in a potentially long delay in receiving results. Agreed that given that chunking will not be supported it is reasonable to require the consumer to do any sorting that is required.

XCA004: Error handling - Neither the ebXML specification or XDS deals specifically with a situation when results plus errors are both returned. In a cross XDS Affinity Domain environment this situation is much more prevalent, coming up whenever the gateway is able to reach some of the other XDS Affinity Domains and not others. In this case the gateway should return both the results received as well as a message explaining that some XDS Affinity Domains could not be reached. Agreed to include details regarding error messages. Will follow the description in CP 28 which includes the new status return value of partial success.

XCA005: Unique Identification of Patient - Ensuring unique patient identification involves a set of issues to be addressed next year. To integrate disparate regional domains, there's a need to minimize redundancy and provide acceptable performance accessibility. The process of determining which communities to query must be fully addressed as well. For this year we can require that the gateway must ensure that the patient has been uniquely identified prior to the gateway-to-gateway query and leave out of scope how exactly the gateway does that.

XCA006: Interaction with XDS – The interaction between gateways and Registry/Repository has been detailed in a process flow, effectively the gateway acts as a Document Consumer when interacting with the Registry and Repository. See XCA014 for interactions with the XDS Document Consumer.

XCA010: How to handle vocabulary differences across communities. Agreed that scoping for this year will assume all sharing communities are using the same coding vocabulary for metadata elements.

XCA007: Managing metadata – at issue is the technique used by the gateway on resolving references provided in document entry metadata to the originating gateway. This issue comes up in two situations:

- **EntryUUID&UniqueID** - EntryUUID and UniqueID are values in the ExtrinsicObject which can be used in subsequent queries. When the gateway receives a query which only references one of these values it needs to know which gateway to route the request to. The resolution is that the gateway adds to each ExtrinsicObject and ObjectRef the "home" attribute supported by ebXML which references the gateway the object was received from. This home attribute would then be returned as part of the query parameters for the gateway to resolve.
- **Document reference** - The XCA profile will only support the new Web Services based Retrieve, called Retrieve Document Set, which specifies a document uniqueId and repository id. The gateway will need to translate these elements into the gateway which holds the document being retrieved. This profile requires the consumer to specify the "home" element (described above) in the retrieve transaction.

Issue resolved by requiring "home" to be specified in the return from the query in all appropriate elements and requiring subsequent queries by ID to include "home". Also require retrieve to include home in the request.

XCA008: Need Risk Analysis Section to be completed. DONE

XCA009: Need description of audit messages. Audit information added which refers to base transaction audit requirements and does not extend.

XCA011: How to define the mapping from the home attribute to the gateway to be contacted about that home? Resolved the following:

- Static association cached in the gateway: the gateway is required to route subsequent requests including the home attribute to the same "place" that it was received from. This requires the gateway to remember what that place was, and know how to route both subsequent queries and retrieves to that place. The gateway would need to cache this association, probably forever.

XCA013: Should the transactions defined (Gateway Query, Cross Gateway Query, Gateway Retrieve, Cross Gateway Retrieve) be new transactions, or re-use of existing transactions with extra requirements? Agreed that the Stored Query and Retrieve Document Set transactions will be re-used for consumer/gateway interaction, but the transactions between gateways will remain as new transactions.

XCA014: Request that the Document Consumer defined in the XCA profile be identical (or nearly identical) to the Document Consumer defined in the XDS.b profile. To accomplish the Registry Stored Query and the Retrieve Document Set transactions have been updated by this supplement to require the Document Consumer to propagate the homeCommunityId attribute, if present, from a Registry Stored Query response into a subsequent query or retrieve. This is a requirement of the Document Consumer whether it is implementing the XCA or XDS.b profiles. The implication of this change is that compliance with the XDS.b Document Consumer implies compliance with the XCA Document Consumer. The effects of this on the Retrieve Document Set transaction are described in the XDS.b supplement.

2 Profile Abstract

Add the following to section 3 Profile Abstract:

The Cross-Community Access profile supports the means to query and retrieve patient relevant medical data held by other communities. A community is defined as a coupling of facilities/enterprises that have agreed to work together using a common set of policies for the purpose of sharing clinical information via an established mechanism. Facilities/enterprises may host any type of healthcare application such as EHR, PHR, etc. A community is identifiable by a globally unique id called the homeCommunityId. Membership of a facility/enterprise in one community does not preclude it from being a member in another community. Such communities may be XDS Affinity Domains which define document sharing using the XDS profile or any other communities, no matter what their internal sharing structure.

3 Glossary

Add the following to section 4 Glossary:

Community: A community is defined as a coupling of facilities/enterprises that have agreed to work together using a common set of policies for the purpose of sharing clinical information via an established mechanism. Facilities/enterprises may host any type of healthcare application such as EHR, PHR, etc. A community is identifiable by a globally unique id called the homeCommunityId. Membership of a facility/enterprise in one community does not preclude it from being a member in another community. Such communities may be XDS Affinity Domains which define document sharing using the XDS profile or any other communities, no matter what their internal sharing structure.

homeCommunityId: A globally unique identifier for a community. It is used in XCA to obtain the Web Services endpoint of services that provide access to data in that community.

Volume I – Integration Profiles

This section describes the changes required in Volume I of the Technical Framework that result form including this Integration Profile.

History of Annual Changes

<Brief description of what to add to Volume I, section 1.7 which gives a brief overview of "what's new" in the given year of the Technical Framework.>

Add the following bullet to the end of the bullet list in section 1.7

- Added the XCA Profile which defines transactions for query and retrieve of documents outside of a local community.

Add the following section to Table 2-1 Integration Profiles Dependencies in section 2.1

XCA	Audit Trail and Node Authentication	Each XCA Actor shall be grouped with Secure Node Actor	- Required to manage audit trail of exported PHI, node authentication and transport encryption.
XCA	Consistent Time	Each XCA Actor shall be grouped with the Time Client Actor.	To ensure consistency among document and submission set dates.

Add the following section to section 2.2

2.2.18 XCA Integration Profile

The Cross-Community Access profile supports the means to query and retrieve patient relevant medical data held by other communities. A community is defined as a coupling of facilities/enterprises that have agreed to work together using a common set of policies for the purpose of sharing clinical information via an established mechanism. Facilities/enterprises may host any type of healthcare application such as EHR, PHR, etc. A community is identifiable by a globally unique id called the homeCommunityId. Membership of a facility/enterprise in one community does not preclude it from being a member in another community. Such communities may be XDS Affinity Domains which define document sharing using the XDS profile or any other communities, no matter what their internal sharing structure.

The section shall be added to Vol 1

18 XCA Integration Profile

The Cross-Community Access profile supports the means to query and retrieve patient relevant medical data held by other communities. A community is defined as a coupling of facilities/enterprises that have agreed to work together using a common set of policies for the purpose of sharing clinical information via an established mechanism. Facilities/enterprises may host any type of healthcare application such as EHR, PHR, etc. A community is identifiable by a globally unique id called the homeCommunityId. Membership of a facility/enterprise in one community does not preclude it from being a member in another community. Such communities may be XDS Affinity Domains which define document sharing using the XDS profile or any other communities, no matter what their internal sharing structure.

18.1 Actors/ Transactions

Figure 18.1-1 shows the actors directly involved in the XCA Integration Profile and the relevant transactions between them.

Note: The Document Consumer Actor is shown in Figure 18.1-1 to clarify the responsibility of the XDS Affinity Domain Option discussed in section 18.2.

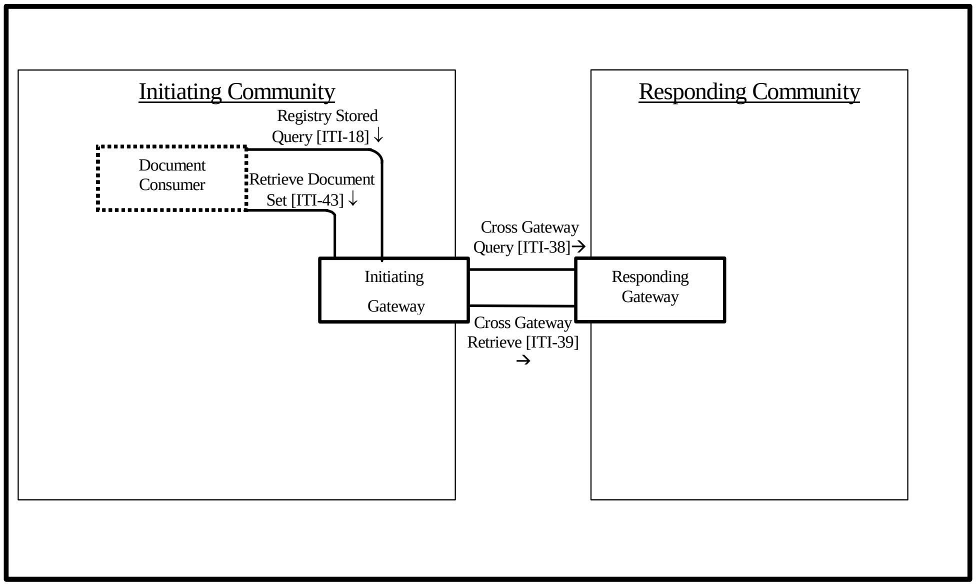

Initiating Community
Registry Stored Query [ITI-18] ↓
Document Consumer
Retrieve Document Set [ITI-43] ↓
Initiating Gateway
Cross Gateway Query [ITI-38]→
Cross Gateway Retrieve [ITI-39] →
Responding Community
Responding Gateway

Figure 18.1-1. XCA Actor Diagram

Table 18.1-1 lists the transactions for each actor directly involved in the XCA Profile. In order to claim support of this Integration Profile, an implementation must perform the required transactions (labeled "R"). Transactions labeled "O" are optional. A complete list of options defined by this Integration Profile and that implementations may choose to support is listed in Volume I, Section 18.2.

Table 18.1-1. XCA Integration Profile - Actors and Transactions

Actors	Transactions	Optionality	Section in Vol. 2
Initiating Gateway	Cross Gateway Query	R	3.38
	Cross Gateway Retrieve	R	3.39
	Registry Stored Query	O	3.18
	Retrieve Document Set	O	3.43
Responding Gateway	Cross Gateway Query	R	3.38
	Cross Gateway Retrieve	R	3.39

Note: When an Initiating or Responding Gateway is grouped with a Document Consumer, there are additional requirements. See 18.2.3 for a description of grouping.

18.2 XCA Integration Profile Options

Options that may be selected for this Integration Profile are listed in the table 18.2-1 along with the Actors to which they apply. Dependencies between options when applicable are specified in notes.

Table 18.2-1 XCA Integration Profile - Actors and Options

Actor	Options	Vol & Section
Initiating Gateway	*XDS Affinity Domain Option* *Asynchronous Web Services Exchange*	ITI TF-1:18.2.1 ITI TF-1:18.2.2
Responding Gateway	*No options defined*	- -

18.2.1 XDS Affinity Domain Option

Initiating Gateways which support the XDS Affinity Domain Option interact with Document Consumers within the XDS Affinity Domain served by the Initiating Gateway.

Initiating Gateway actors which support this option:

- shall **receive** Registry Stored Query [ITI-18] transactions from a local Document Consumer actor and act on those requests on behalf of the Document Consumer. When receiving a Registry Stored Query from a local Document Consumer, shall require the homeCommunityId as an input parameter on relevant queries, and shall specify the homeCommunityId attribute within its Registry Stored Query responses. See Section 18.3.2 for description of homeCommunityId.
- shall **receive** Retrieve Document Set [ITI-43] transactions from a local Document Consumer actor and act on those requests on behalf of the Document Consumer. When

receiving a Retrieve Document Set from a local Document Consumer, shall require the homeCommunityId as an input parameter.

When an Initiating Gateway does not support the XDS Affinity Domain option it is expected to be using non-IHE specified interactions to communicate remote community data to systems within its local community. These proprietary interactions are not further described within any IHE profile.

See the relevant transactions for further details regarding the homeCommunityId attribute.

18.2.2 Asynchronous Web Services Exchange Option

Initiating Gateways which support Asynchronous Web Services Exchange shall support Asynchronous Web Services Exchange on the Cross Gateway Query [ITI-38] and Cross Gateway Retrieve [ITI-39] transactions. If the Initiating Gateway supports both the XDS Affinity Domain Option and the Asynchronous Web Services Option it shall support Asynchronous Web Services Exchange on the Registry Stored Query [ITI-18] and Retrieve Document Set [ITI-43] transactions.

18.2.3 Grouping Rules

Grouping with a Document Consumer Actor is used in situations where an Initiating Gateway and/or Responding Gateway are supporting an XDS Affinity Domain

When an Initiating Gateway is supporting an XDS Affinity Domain, it can choose to query and retrieve from local actors in addition to remote communities. This is accomplished by grouping the Initiating Gateway Actor with a Document Consumer Actor. This grouping allows Document Consumers such as EHR/PHR/etc systems to query the Initiating Gateway to retrieve document information and content from both the local XDS Affinity Domain as well as remote communities. For details see 18.2.3.1. An Initiating Gateway Actor that is not grouped with a Document Consumer Actor is only able to return results from remote communities, so local EHR/PHR/etc systems (Document Consumer Actors) must direct separate query and document retrieve transactions internally and externally.

When a Responding Gateway is supporting an XDS Affinity Domain, it may resolve Cross Gateway Query and Cross Gateway Retrieve Transactions by grouping with a Document Consumer Actor and using the Registry Stored Query and Retrieve Document Set transactions. For details see 18.2.3.2

18.2.3.1 Initiating Gateway grouped with Document Consumer

Initiating Gateways that are grouped with a Document Consumer:

- shall support the XDS Affinity Domain option
- shall **initiate** Registry Stored Query [ITI-18] transactions to a local Document Registry to query local information in response to a received Registry Stored Query [ITI-18] from a local Document Consumer.

- shall **initiate** Retrieve Document Set [ITI-43] transactions to a local Document Repository in response to a received Retrieve Document Set from a local Document Consumer which contains a homeCommunityID indicating the local community.

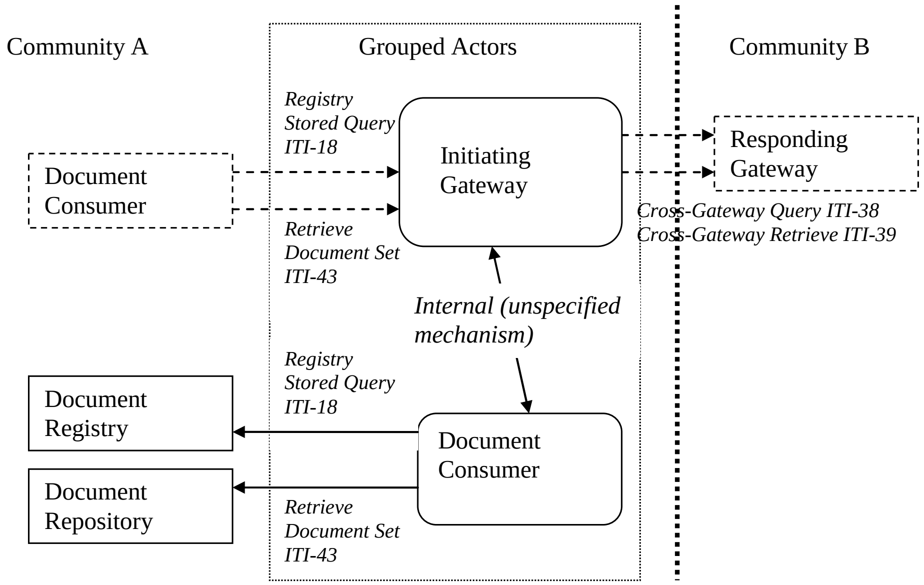

Community A
Grouped Actors
Community B
Registry Stored Query ITI-18
Document Consumer
Initiating Gateway
Responding Gateway
Retrieve Document Set ITI-43
Cross-Gateway Query ITI-38
Cross-Gateway Retrieve ITI-39
Internal (unspecified mechanism)
Registry Stored Query ITI-18
Document Registry
Document Consumer
Document Repository
Retrieve Document Set ITI-43

Figure 18.2-1. Initiating Gateway grouped with Document Consumer

18.2.3.1 Responding Gateway grouped with Document Consumer

Responding Gateways that are grouped with a Document Consumer:

- shall **initiate** a Registry Stored Query [ITI-18] transaction to a local Document Registry to query local information in response to a received Cross Gateway Query [ITI-38]. The Document Registry response must be augmented with the homeCommunityId of the Responding Gateway's community prior to returning in the response to the Cross Gateway Query.
- shall **initiate** a Retrieve Document Set [ITI-43] transaction to a local Document Repository to retrieve local information in response to a Cross Gateway Retrieve [ITI-39].

When a Responding Gateway is not grouped with a Document Consumer actor it is expected to be using non-IHE specified interactions to collect local information in response to a Cross Gateway Query or Cross Gateway Retrieve. These proprietary interactions are not further described within any IHE profile.

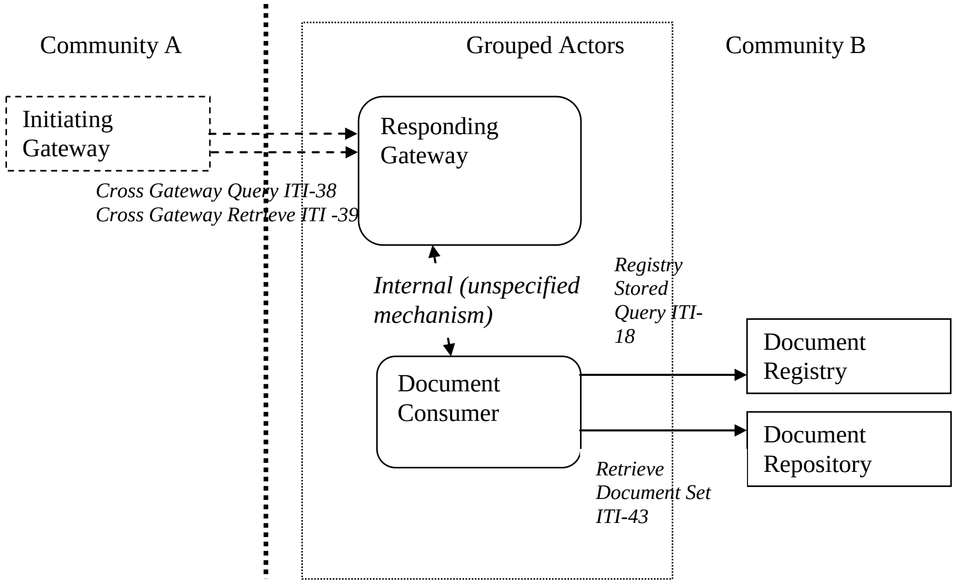

Community A
Grouped Actors
Community B
Initiating Gateway
Responding Gateway
Cross Gateway Query ITI-38
Cross Gateway Retrieve ITI -39
Internal (unspecified mechanism)
Registry Stored Query ITI-18
Document Consumer
Document Registry
Document Repository
Retrieve Document Set ITI-43

Figure 18.2-2. Responding Gateway grouped with Document Consumer

18.3 XCA Process Flow

18.3.1 Use Cases

Assume within a given domain, such as the State of California, we have several healthcare communities (or XDS Affinity Domains or RHIOs). One in Los Angeles is based on IHE-XDS. One in Sacramento is based on another form of healthcare sharing infrastructure. One in San Francisco is also based on IHE-XDS. A patient X, who travels frequently, has received healthcare in each of these communities. Patient X is admitted to a hospital in LA. The attending physician uses his hospital information system to query across multiple domains for healthcare information about this patient. Once found, references to patient data outside the local domain are cached locally for easy future reference.

18.3.2 homeCommunityId defined

This profile makes use of a homeCommunityId value 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. Specifically:

- It is returned within the response to Cross Gateway Query and Registry Stored Query transactions to indicate the association of a response element with a community. Document Consumers process the value in the response as an opaque unique identifier.

- It is an optional parameter to Registry Stored Query requests, not requiring a patient id parameter, and Retrieve Document Set requests to indicate which community to direct the request.
- It is used by Initiating Gateways to direct requests to the community where the initial data originated.

18.3.3 Detailed Interactions

The following diagram presents a high level view of the interactions between actors when both initiating and responding communities are XDS Affinity Domains i.e. use of the XDS Affinity Domain option and the Initiating Gateway and Responding Gateway are each grouped with a Document Consumer. Details on each interaction follow the diagram.

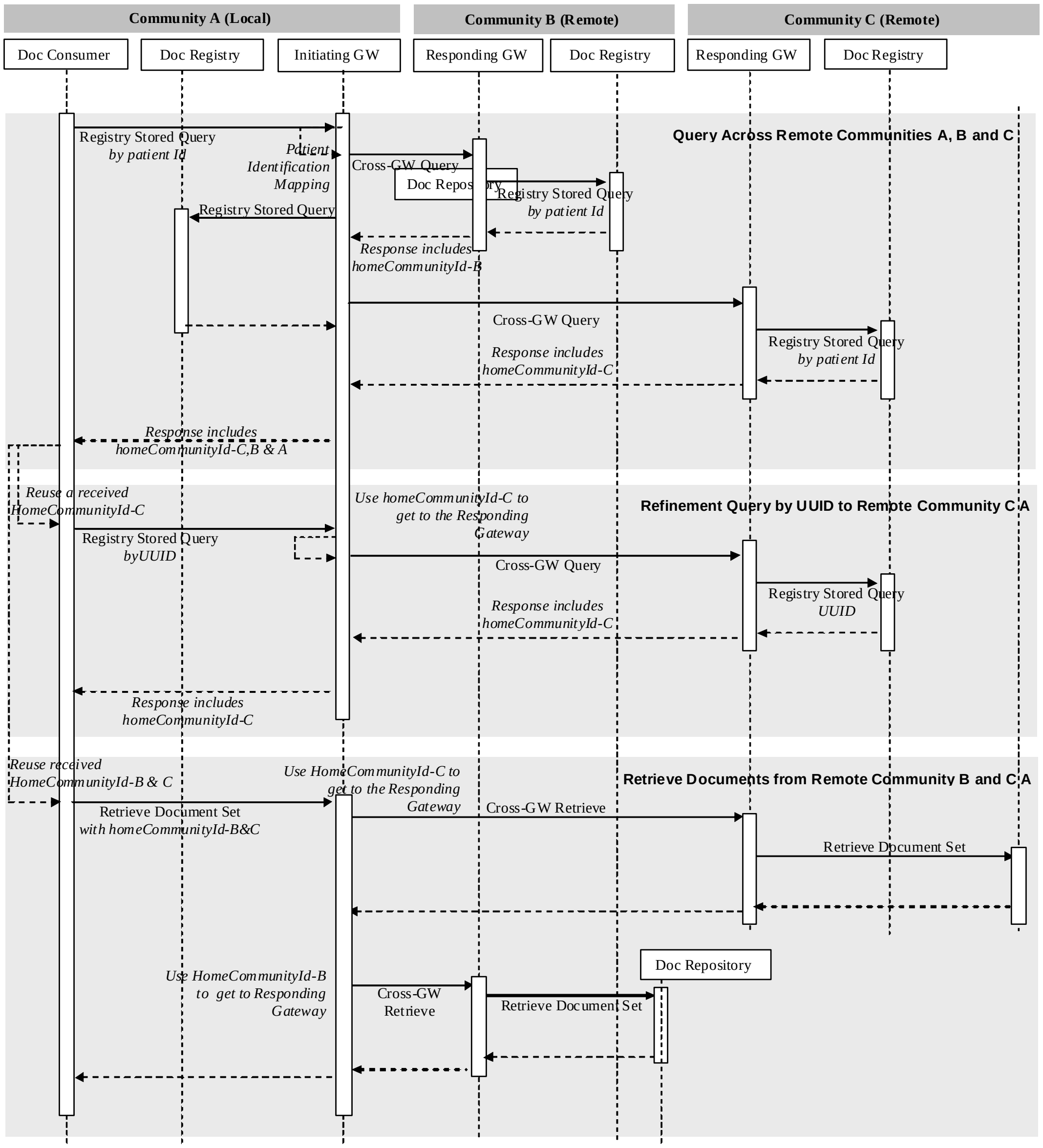

Community A (Local)
Community B (Remote)
Community C (Remote)
Doc Consumer
Doc Registry
Initiating GW
Responding GW
Doc Registry
Responding GW
Doc Registry
Query Across Remote Communities A, B and C
Registry Stored Query by patient Id
Patient Identification Mapping
Cross-GW Query
Doc Repository
Registry Stored Query by patient Id
Registry Stored Query
Response includes homeCommunityId-B
Cross-GW Query
Registry Stored Query by patient Id
Response includes homeCommunityId-C
Response includes homeCommunityId-C,B & A
Refinement Query by UUID to Remote Community C A
Reuse a received HomeCommunityId-C
Use homeCommunityId-C to get to the Responding Gateway
Registry Stored Query byUUID
Cross-GW Query
Registry Stored Query UUID
Response includes homeCommunityId-C
Response includes homeCommunityId-C
Retrieve Documents from Remote Community B and C A
Reuse received HomeCommunityId-B & C
Use HomeCommunityId-C to get to the Responding Gateway
Retrieve Document Set with homeCommunityId-B&C
Cross-GW Retrieve
Retrieve Document Set
Doc Repository
Use HomeCommunityId-B to get to Responding Gateway
Cross-GW Retrieve
Retrieve Document Set

Figure 18.3.3: XCA Detailed Interactions

- **Document Consumer** *initiates a Registry Stored Query request by patient id* – the Document Consumer initiates the initial transaction by formatting a Registry Stored Query request by patient identifier. The consumer uses PDQ, PIX or some other means 440 to identify the XDS Affinity Domain patient id, formats that information plus any other query parameters into a Registry Stored Query request and sends this request to an Initiating Gateway.

- **Initiating Gateway** *processes Registry Stored Query by patient id request* – The Initiating Gateway receives a Registry Stored Query by patient id and must determine a) which Responding Gateways this request should be sent to b) what patient id to use in the Cross Gateway Queries. Detailed specification of these steps is not in the intended scope of this profile. Combination of this profile with other existing profiles (e.g. PIX/PDQ), future profiles or configuration mechanisms is possible. Please refer to *E.7 XCA and Patient Identification Management* for possible use of existing profiles PIX and PDQ. For each Responding Gateway identified, the Initiating Gateway shall update the query with the correct patient identifier corresponding to the Responding Gateway's community and initiates a Cross Gateway Query transaction to the Responding Gateway. If the Initiating Gateway is grouped with a Document Consumer it shall also initiate a Registry Stored Query to the local Document Registry.
- **Responding Gateway** *processes Cross Gateway Query by patient id* – The Responding Gateway within an XDS Affinity Domain processes the Cross Gateway Query by using grouping as a Document Consumer and initiates a Registry Stored Query to the local Document Registry. The Responding Gateway shall update the response from the Document Registry to ensure that the homeCommunityId is specified on every applicable element. This updated response is sent as the response to the Cross Gateway Query.
- **Initiating Gateway** *processes Cross Gateway Query by patient id responses* – The Initiating Gateway collects the responses from all Responding Gateways it contacted. For each response it shall verify homeCommunityId is present in each appropriate element. If the Initiating Gateway initiated a Registry Stored Query to the local Document Registry it shall update the response to that transaction to contain the homeCommunityId value associated with the local community. Once all responses are received the Initiating Gateway consolidates all updated response data into one response to the Document Consumer. The Initiating Gateway shall return to the Document Consumer the same homeCommunityId attribute values that it received from Responding Gateways.
- **Document Consumer** *receives Registry Stored Query by patient id response* – The Document Consumer receives the results of the query from the Initiating Gateway and must account for two unique aspects of the response; namely that a) the homeCommunityId attribute will be specified and b) the Document Consumer may not be able to map the repository id value directly to the Document Repository. There shall be a common coding/vocabulary scheme used across all communities. For example, all communities shall have common privacy consent vocabularies. The Document Consumer shall retain the values of the homeCommunityId attribute for future interaction with the Initiating Gateway.
- **Document Consumer** *initiates a Registry Stored Query by UUID* – Many Registry Stored Queries do not include patient id as a parameter, but instead require one of the entryUUID or uniqueID parameters, generically referred to as UUID. Both of these values are returned as part of the metadata from a query by patient id. The Document

Consumer may do a patient id query to the Initiating Gateway prior to a query by UUID or shall have access to the correct homeCommunityId through some other means. In either case the consumer has the homeCommunityId attribute and shall specify it as a parameter of the query. The Document Consumer puts the homeCommunityId and UUID values plus any other query parameters into a Registry Stored Query request and sends this request to an Initiating Gateway.

- **Initiating Gateway** *processes Registry Stored Query by UUID request* – The Initiating Gateway receives a Registry Stored Query by UUID and determines which Responding Gateway to contact by using the homeCommunityId to obtain the Web Services endpoint of the Responding Gateway. The process of obtaining the Web Services endpoint is not further specified in this profile. If the homeCommunityId represents the local community the Initiating Gateway will initiate a Registry Stored Query to the local Document Registry. The Initiating Gateway shall specify the homeCommunityId in the Cross Gateway Query by UUID which is associated with the Responding Gateway.
- **Responding Gateway** *processes Cross Gateway Query by UUID* – The Responding Gateway within an XDS Affinity Domain processes the Cross Gateway Query by grouping as a Document Consumer and initiating a Registry Stored Query to the local Document Registry. The response to the Cross Gateway query shall contain the homeCommunityId of the responding community. This processing is identical to processing of the Cross Gateway Query by patient id.
- **Initiating Gateway** *receives Cross Gateway Query by UUID response* – The processing of a Cross Gateway Query by UUID response is identical to the processing of a Cross Gateway Query by patient id response, except there is only one response, so consolidation of responses is not needed.
- **Document Consumer** *receives Registry Stored Query by UUID response* – The processing of a Registry Stored Query by UUID response is identical to the processing of a Registry Stored Query by patient id response.
- **Document Consumer** *initiates a Retrieve Document Set* – Prior to issuing a Retrieve Document Set the Document Consumer may issue a Registry Stored Query by patient id to the Initiating Gateway. The response to the Registry Stored Query by patient id or subsequent Registry Stored Query by UUID includes a) the document unique ID b) the repository unique ID c) the homeCommunityId attribute. If the Document Consumer did not issue a Registry Stored Query which returned this information then it shall have acquired the information through some other means. The Document Consumer shall specify these three parameters in its Retrieve Document Set transaction to the Initiating Gateway.
- **Initiating Gateway** *processes Retrieve Document Set* – The Initiating Gateway determines which Responding Gateways to contact by using the homeCommunityId to obtain the Web Services endpoint of the Responding Gateway. If the homeCommunityId represents the local community the Initiating Gateway will initiate a Retrieve Document Set to a local Document Repository. The Retrieve Document Set may contain more than one unique homeCommunityId so the Initiating Gateway shall be capable of initiating

requests to more than one Responding Gateway and consolidating the results. The Initiating Gateway shall specify the homeCommunityId in the Cross Gateway Retrieve which identifies the community associated with the Responding Gateway.

- **Responding Gateway** *processes Cross Gateway Retrieve* – The Responding Gateway within an XDS Affinity Domain processes the Cross Gateway Retrieve by grouping as a Document Consumer and initiating a Retrieve Document Set transaction to the Document Repository identified by the repository unique ID within the request. If the Cross Gateway Retrieve requests multiple documents with different repository unique IDs, the Responding Gateway shall contact multiple Document Repositories and consolidate the responses.

18.4 XCA Security Considerations

18.4.1 XCA Risk Assessment

The risk analysis for XCA enumerates assets, threats, and mitigations. The complete risk data is stored and maintained in a central location. The complete risk data is stored and available from IHE[1].

The purpose of this risk assessment is to notify vendors of some of the risks that they are advised to consider in implementing XCA actors. For general IHE risks and threats please see ITI TF-1: Appendix L. The vendor is also advised that many risks can not be mitigated by the IHE profile and instead the responsibility for mitigation is transferred to the vendor, and occasionally to the XDS Affinity Domain and enterprises. In these instances, IHE fulfills its responsibility to notify affected parties through the following section.

18.4.2 Requirements/Recommendations

The following mitigations shall be implemented by all XCA actors. These mitigations moderate all high impact risks.

- M1: All actors in XCA shall be grouped with an ATNA Secure Node actor and a CT Time Client actor.
- M2: Document metadata shall include a SHA1 hash of the document content. Applications shall have the ability to verify the SHA1 hash of the document with the SHA1 hash in the metadata, if corruption detection is requested.
- M3: Document Consumer implementations shall handle overloading through excessive volume of response data by discontinuing the read on the socket and closing it. The Initiating and Responding Gateways shall respond to disconnection by discontinuing processing of responses.

[1] The risk analysis data may be found at: ftp://ftp.ihe.net/IT_Infrastructure/iheitiyr5-2007-2008/Technical_Cmte/Profile_Work/XC/XCARiskAnalysis.xls

- M4: Document Consumer implementations shall not issue a Registry Stored Query that is not patient specific, i.e. it shall either supply a patient identifier or a unique document entry identifier.
- **M6**: Queries of unknown patient identifiers shall return zero documents with no further information. This applies to patient identifiers that are properly formatted or improperly formatted. By not defining an error code indicating that the identifier is ill-formatted, you are reducing the ability of applications to fish for data. This applies to Document Registries and Responding Gateways, if appropriate.

The following mitigations address the risk of a document being maliciously changed. This mitigation is optional.

- **M5**: Documents may be digitally signed using the DSG profile

The following mitigations are transferred to the vendors, XDS Affinity Domains, and enterprises.

- **T1**: Backup systems for registry metadata, repository documents, and gateway configuration are recommended.
- **T2**: All implementations are recommended to ensure that all received data is propagated appropriately (i.e. without corruption and complete results) or an error is presented.
- **T3**: Network protection services are recommended to be sufficient to guard against denial of service attacks on all service interfaces.
- **T4**: A process that reviews audit records and acts on inappropriate actions is recommended.
- **T5**: It is recommended that service interfaces be implemented with a good design to guard against corruption and denial of service attacks

18.4.3 Policy Choices

Policy choices will not be addressed by this profile. Each community may have different policies. The profile has been designed with this fact in mind and an understanding of enough variety of policies so that any reasonable policy can be implemented without violating the profile.

<Appendix A> Actor Summary Definitions

Update the definition of Document Consumer as shown:

Document Consumer - The Document Consumer Actor **is able to query document metadata and retrieve documents.** queries a Document Registry Actor for documents meeting certain criteria, and retrieves selected documents from one or more Document Repository actors.

Initiating Gateway - supports all outgoing inter-community communications.

Responding Gateway – supports all incoming inter-community communications.

<Appendix B> Transaction Summary Definitions

Cross Gateway Query – send a query from one community to another in order to identify where healthcare information satisfying specific restraints is located.

Cross Gateway Retrieve – request the retrieval of a specific set of healthcare information (a document or documents) from a remote location.

<Appendix E> Cross Profile Considerations

E.6 XCA Integration with XDS and non-XDS communities

This section is informative and suggests some potential configurations that may be used by a community. The following types of community are described:

- An XDS Affinity Domain
- A non-XDS Affinity Domain
- A collection of XDS Affinity Domains
- A collection of non-XDS Affinity Domains
- An XDS Affinity Domain with a “transparent” Gateway

E.6.1 An XDS Affinity Domain

In the example below, the responding community is an XDS Affinity Domain which is served by a Responding Gateway.

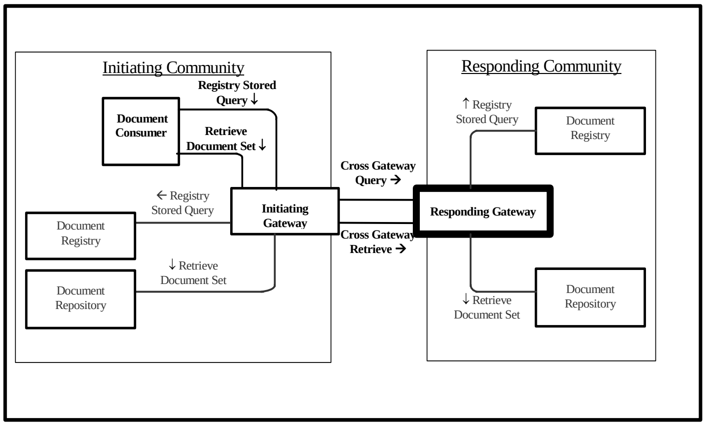


Figure E.6.1: XDS Affinity Domain

E.6.2 A Non-XDS Affinity Domain

In the example below, the responding community is served by a Responding Gateway. However, within this community, there is no XDS Document Registry or Repositories. A proprietary mechanism is used by the Responding Gateway to gather data for the response to the Cross Gateway Query and Cross Gateway Retrieve transactions.

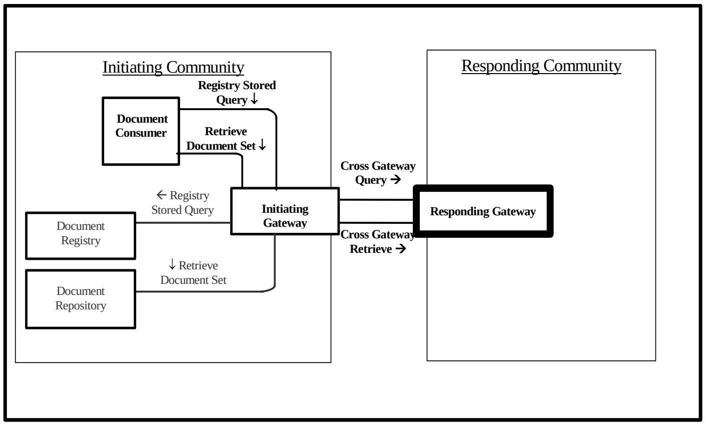


Figure E.6.2: Non-XDS Affinity Domain

E.6.3 A Collection of XDS Affinity Domains (informative)

In the example below, one Responding Gateway is serving two communities. Each one of these communities is an XDS Affinity Domain served by its own Responding Gateway; these two Responding Gateways are hidden from the initiating community.

This example is informative only. This profile does not specifically support this configuration and does not address all the considerations of such a configuration.

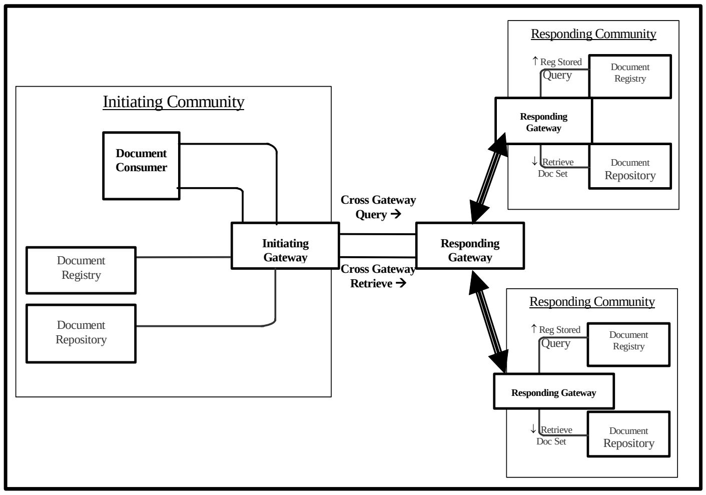


Figure E.6.3: Collection of XDS Affinity Domains

E.6.4 A Collection of Non-XDS Affinity Domains (informative)

In the example below, one Responding Gateway is serving two communities. Each one of these communities is a non-XDS Affinity Domain served by its own Responding Gateway; these two Responding Gateways are hidden from the initiating Community.

This example is informative only. This profile does not specifically support this configuration and does not address all the considerations of such a configuration.

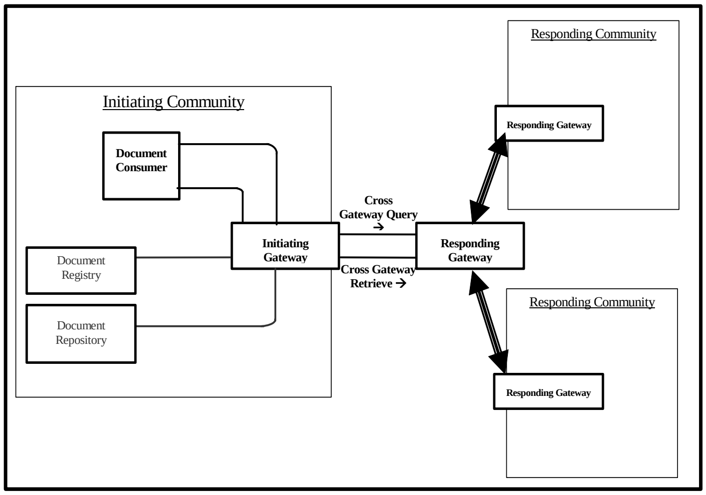


Figure E.6.4: Collection of Non-XDS Affinity Domains

E.6.5 An XDS Affinity Domain with a "Transparent" XCA Gateway (informative)

In the example below, the initiating community is an XDS Affinity Domain where the Initiating Gateway is grouped with the XDS Affinity Domain Document Registry. Thus the Document Consumer interacts with one system to retrieve both local and non-local data. This is called "transparent Gateway" as the Document Consumers do not see the cross-domain communication explicitly, but it is hidden by the Domain Registry and a Proxy Repository. Configuration would be needed to instruct the Document Consumer to interact with the Initiating Gateway when a non-local repository identifier was found in the metadata. In this way the Document Consumer interacts with the Initiating Gateway as a Proxy repository.

This diagram also shows a Responding Gateway grouped with a Document Registry.

This example is informative only. This profile does not specifically support this configuration and does not address all the considerations of such a configuration

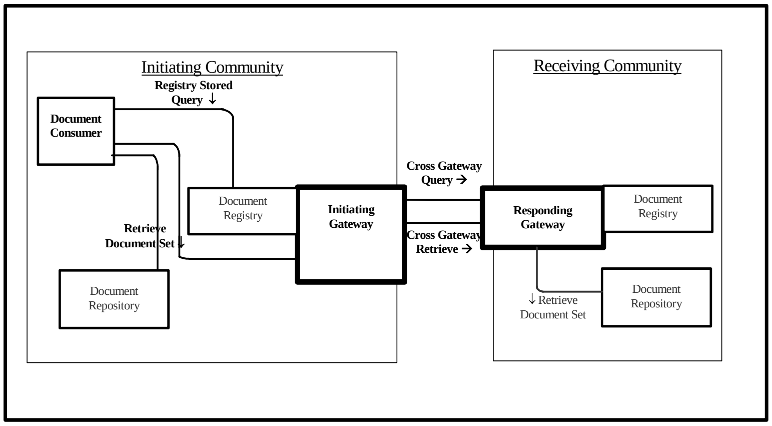


Figure E.6.5: An XDS Affinity Domain with a "Transparent" XCA Gateway

E.7 XCA and Patient Identification Management

This section describes two models for resolving the patient identity in a cross-community exchange environment. As the XCA profile is not intended to address patient identification management, it is therefore necessary to combine XCA with appropriate identification management Integration Profiles. This section is informative and describes only two possible ways to resolve patient identification relying on the existing two IHE Integration Profiles in this domain, Patient Identifier Cross-Referencing (PIX) and Patient Demographics Query (PDQ). The description in this section is only at a high level and more details (not covered here) are necessary for implementation of these models. Other models for patient identification exist and will not be described in this section. Future work by the IHE IT Infrastructure committee may support more sophisticated approaches.

E.7.1 Patient Identification using PIX

The following diagram describes a mechanism for managing patient identities where there is topmost PIX which cross references between communities A, B and C. This diagram assumes that a Responding or Initiating Gateway for each community interacts in order to drive a patient identity feed to the topmost PIX. The diagram does not include processing on the remote communities (B and C) to respond to the query request. The topmost PIX is not defined in this example, but can be assumed to be a PIX Manager, or equivalent, which is accessible to all communities.

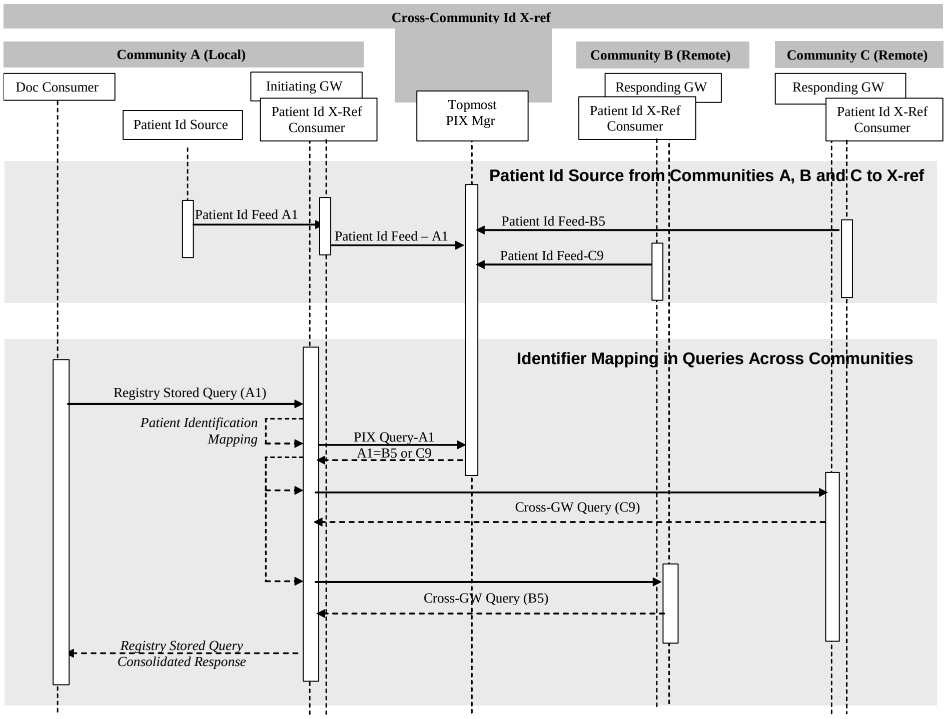


Figure E.7.1: Patient Identification using PIX

E.7.2 Patient Identification using PDQ

The following diagram describes one approach to patient identification in a cross-community exchange where there is no entity which can cross reference between local and remote identifiers. Note that interactions among entities in remote communities (B & C) are not detailed in this diagram.

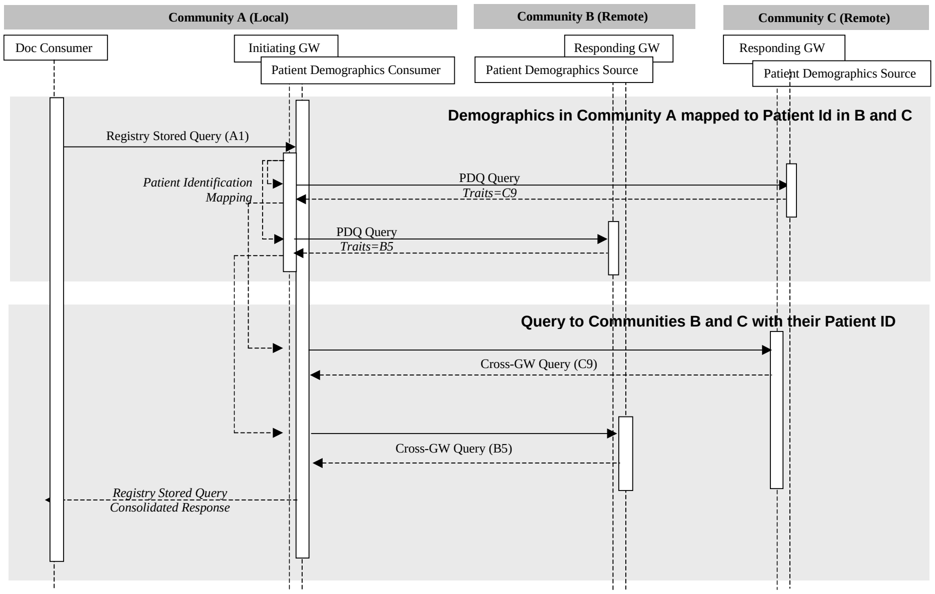

Community A (Local)
Community B (Remote)
Community C (Remote)
Doc Consumer
Initiating GW
Patient Demographics Consumer
Responding GW
Patient Demographics Source
Responding GW
Patient Demographics Source
Demographics in Community A mapped to Patient Id in B and C
Registry Stored Query (A1)
Patient Identification Mapping
PDQ Query
Traits=C9
PDQ Query
Traits=B5
Query to Communities B and C with their Patient ID
Cross-GW Query (C9)
Cross-GW Query (B5)
Registry Stored Query
Consolidated Response

Figure E.7.2: Patient Identification using PDQ

This diagram present a basic approach relying on the existing IHE Patient Demographics Query (PDQ) Integration Profile by the Initiating and Responding Gateways, where the Responding Gateway respond to queries with patient demographics traits for potential patients in the community it serves, thus allowing Initiating Gateways to obtain the patient Id to use in the Cross Gateway Query. The result of this transaction would be a) zero, indicating the patient does not have records at that community b) one, indicating the gateway was able to uniquely identify the patient c) multiple, indicating the Responding Gateway was not able to uniquely identify the patient. In the case of a) or b) the transaction is complete and does not require human intervention. If multiple results are returned this requires human intervention to resolve.

This approach requires a significant number of policy decisions to be in place, coordinated with privacy consent in cross-community environment that are well beyond the scope of the combined use of PDQ and XCA presented in Figure E.7.2. In addition, the integration of a large number of communities with a large number of non overlapping patient populations is likely to require addressing significant scaling issues in allowing Responding Gateways to process the requests for identity resolution.

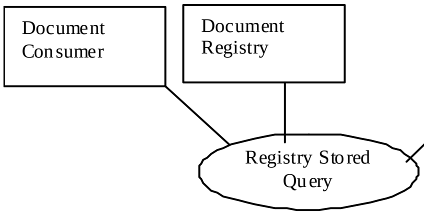
Future IHE work in this area may offer more sophisticated integration profiles that could be combined with XCA.

Volume 2 - Transactions

3.18 Registry Stored Query

Update TF Volume 2 section 3.18.2 to add Initiating Gateway as an actor on the transaction.

3.18.2 Use Case Roles



Actor: Initiating Gateway

Role: Services the stored query by initiating transactions with a selected set of Responding Gateways, Document Registries or other appropriate systems.

Update TF Volume 2 section 3.18.4.1.2.3.8 as follows

3.18.4.1.2.3.8 Use of homeCommunityId

The Registry Stored Query makes use of 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`.

Its use is as follows:

- It is returned within the response to Registry Stored Query **and Cross Gateway Query** transactions to indicate the association of a response element with a community. It is specified as the ebRIM 'home' attribute within the relevant response elements. Document Consumers process the value as an opaque unique identifier.
- It is an optional parameter to Registry Stored Query requests, not requiring a patient id parameter, and Retrieve Document Set requests to indicate which community to direct the request.

For stored queries which do not require the patient id as a parameter, meaning query by EntryUUID or UniqueID:

- **If the Registry Stored Query is being addressed to an Initiating Gateway then the Document Consumer may have previously sent a Registry Stored Query to the Initiating Gateway which included a patient id and saved the homeCommunityId which was returned on the element containing the EntryUUID or uniqueID. If this is not the case the Document Consumer shall have access to the correct homeCommunityId through some other means.**
- If the Document Consumer received the EntryUUID or uniqueID in a previous Registry Stored Query response which contained a homeCommunityId, then the Document Consumer shall specify the homeCommunityId parameter.
- The homeCommunityId value is specified as the home attribute on the AdhocQuery element of the query request, as in:
  `<AdhocQuery id=”…” home=”urn:oid:1.2.3” … >`
- Each query request can have at most one homeCommunityId value. If the Document Consumer specifies multiple entryUUID or uniqueID values they must all be associated with the same homeCommunityId value. Multiple individual query requests can be used to retrieve data associated with different homeCommunityIds. This restriction is expected to be temporary as query syntax is extended to handle multiple homeCommunityIds in a single query request.

Update TF Volume 2 section 3.18.4.1.3 Expected Actions to describe the processing of the homeCommunityId attribute.

3.18.4.1.3 Expected Actions

The responder to the Registry Stored Query shall

1. Accept a parameterized query in an AdhocQueryRequest message
2. Verify the required parameters are included in the request. Additionally, special rules documented in the above section ‘Parameters for Required Queries’ shall be verified.
3. Errors shall be returned for the following conditions:
   - Unknown query ID
   - Required parameter missing
4. **Process the query as appropriate:**
   - **For Document Registry Actors:** Retrieve the internal implementation template of the query based on the Query ID supplied in the query request, substitute appropriate parameters as indicated in section 3.18.4.1.2.3.7 Parameters for Required Queries and execute the query. The Document Registry shall accept the homeCommunityId value if it is specified in a Registry Stored Query request.

- **For Initiating Gateway Actors:**
    - **Initiating Gateway receives a Registry Stored Query by patient id: It shall determine a) which Responding Gateways this request should be sent to and b) what patient id to use in the Cross Gateway Query. Detailed specification of these steps is not in the intended scope of this profile. Combination of this profile with other existing profiles (e.g. PIX/PDQ), future profiles or configuration mechanisms is possible. Please refer to Vol. 1 E.7 XCA and Patient Identification Management for possible use of existing profiles PIX and PDQ. For each Responding Gateway identified, the Initiating Gateway shall update the query with the correct patient identifier corresponding to the Responding Gateway's community and initiates a Cross Gateway Query transaction to the Responding Gateway. If the Initiating Gateway is grouped with a Document Consumer it will also initiate a Registry Stored Query to the local Document Registry.**
    - **Initiating Gateway receives a Registry Stored Query by entryUUID or uniqueID: Verify homeCommunityId has been specified. If missing return Failure status with XDSMissingHomeCommunityId error code. If homeCommunityId not recognized return a Failure or PartialSuccess status with XDSUnknownCommunity error code. Determine which Responding Gateway to contact by using the homeCommunityId to obtain the Web Services endpoint of the Responding Gateway. The process of obtaining the Web Services endpoint is not further specified in this profile. If the homeCommunityId represents the local community the Initiating Gateway shall initiate a Registry Stored Query to the local Document Registry. The Initiating Gateway shall specify the homeCommunityId in the Cross Gateway Query by entryUUID or uniqueID which identifies the community associated with the Responding Gateway. For details regarding the homeCommunityId see Section 3.18.4.1.2.3.8 and Section 3.38.4.1.2.1.**

5. Return XML formatted metadata in an AdhocQueryResponse message. The syntax of the metadata returned by this query shall be in ebRIM and ebRS version 3.0.
    - The Document Registry may specify the homeCommunityID attribute on any appropriate elements.
    - **The Initiating Gateway shall specify the homeCommunityID attribute on all appropriate elements. If the Initiating Gateway contacted a Document Registry, the Document Registry response might not contain the homeCommunityId. In this case the Initiating Gateway shall add the homeCommunityId of its local community to the Document Registry response prior to including it in the consolidated response to the Document Consumer. The homeCommunityId attribute corresponds to the 'home' attribute specified in the ebRIM standard. For more information on**

homeCommunityId see Section 3.18.4.1.2.3.8 and Section 3.38.4.1.2.1. The elements that shall include the home attribute are:

- **If returntype=”LeafClass” the ExtrinsicObject and RegistryPackage elements shall contain the home attribute.**
- **If returnType=”ObjectRef” the ObjectRef element shall contain the home attribute**
- **If the Initiating Gateway is unable to get an appropriate response from a selected Responding Gateway it shall include in its response to the Document Consumer an XDSUnavailableCommunity error code where the context identifies the unavailable Responding Gateway. In this case, and any other error from a Responding Gateway, the Initiating Gateway shall return to the Document Consumer either a Failure status (if no part was successful) or a PartialSuccess status.**

6. **When the Document Consumer receives the query response from the Initiating Gateway it must account for two aspects of the response; namely that a) the homeCommunityId attribute will be specified b) the Document Consumer may not be able to map the repository id value directly to the Document Repository. XCA assumes a common coding/vocabulary scheme is used across all communities. For example, all communities shall have common privacy consent vocabularies. The Document Consumer shall retain the values of the homeCommunityId attribute for future interaction with the Initiating Gateway.**

Update the Retrieve Document Set transaction, Vol. 2 Section 3.43 to add Initiating Gateway as an actor on the transaction.

3.43 Retrieve Document Set

This section corresponds to Transaction ITI-43 of the IHE Technical Framework. The Document Consumer, Document Repository **and Initiating Gateway** actors use transaction ITI-43.

Integration Profiles using this Transaction
Cross-Enterprise Document Sharing-b (XDS.b)
Cross-Community Access (XCA)

3.43.1 Scope

This transaction is used by the Document Consumer to retrieve a set of documents from the Document Repository **or Initiating Gateway**. The Document Consumer has already obtained the XDSDocumentEntry uniqueId and the Document Repository repositoryUniqueId from the Document Registry**/Initiating Gateway** by means of the Registry Stored Query transaction.

The following diagram is updated to add the reference to Initiating Gateway.

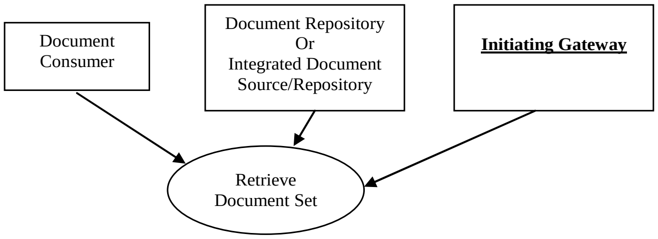

Document Consumer
Document Repository Or Integrated Document Source/Repository
Initiating Gateway
Retrieve Document Set

Figure 3.43.2: Use Case Roles

XDS Actors:

Actor: Document Consumer

Role: Obtains document.

Actor: Document Repository or Integrated Document Source/Repository

Role: Provides documents.

XCA Actors:

Actor: Initiating Gateway

Role: An Initiating Gateway which implements the XDS Affinity Domain option retrieves a set of documents by using the Cross Gateway Retrieve transaction and/or a Retrieve Document Set transaction.

Note: Within this transaction, the Document Repository and Integrated Document Source/Repository actors can be used interchangeably.

Update the Retrieve Document Set transaction, Vol. 2 Section 3.43.4.1.3 as follows.

3.43.4.1.3 Expected Actions

When receiving a Retrieve Document Set Request, a Document Repository **or an Initiating Gateway** shall generate a Retrieve Document Set Response containing the requested documents or error codes if the documents could not be retrieved.

An XCA Initiating Gateway receiving the Retrieve Document Set Request shall use the homeCommunityId to obtain the Web Services endpoint of the Responding Gateways or, in the case where homeCommunityId identifies the local community, use the repositoryUniqueId to obtain the Web Services endpoint of the Document Repositories. The process of obtaining the Web Services endpoint is not further specified in this profile. The Initiating Gateway shall send Cross Gateway Retrieves/Retrieve Document Set

transactions to each appropriate Responding Gateway/Document Repository, consolidate the results, and return them to the Document Consumer.

Add to TF Volume 2 sections 3.38, 3.39

3.38 Cross Gateway Query

This section corresponds to Transaction 38 of the IHE Technical Framework. Transaction 38 is used by cooperating Initiating Gateway and Responding Gateway actors.

3 Scope

The scope of the Cross Gateway Query transaction is based on the Registry Stored Query transaction [ITI-18]. The same set of stored queries is required to be supported and the options controlling what kind of data is returned are the same. Differences from the Registry Stored Query transactions are:

- The Cross Gateway Query is between an Initiating Gateway and Responding Gateway.
- Initiating Gateway shall specify the homeCommunityId attribute in all Cross-Community Queries which do not contain a patient identifier.
- The homeCommunityID attribute shall be returned within all appropriate elements.
- Responding Gateways shall support the Asynchronous Web Services Exchange Option on the Cross Gateway Query. Support for this function is required in order to enable use of Asynchronous Web Services Exchange in any cross-community interaction. Without this support an Initiating Gateway would require unique configuration, per Responding Gateway, to know if Asynchronous Web Services Exchange was supported. It is expected that Asynchronous Web Services Exchange will be desired by the majority of communities.
- Asynchronous Web Services Exchange is an option on the Initiating Gateway, see ITI TF-1:18.2.2.

There shall be an agreed upon common coding/vocabulary scheme used for the Cross Gateway Query. For example, a common set of privacy consent vocabularies shall be used.

3.38.2 Use Case Roles

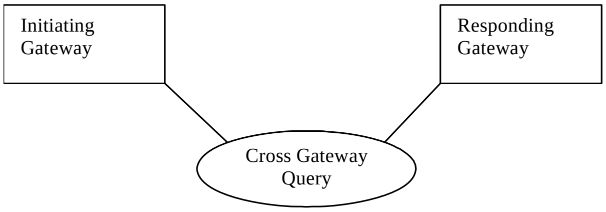


Figure 3.38.2: Use Case Roles

Actor: Initiating Gateway

Role: To formulate a Cross Gateway Query on behalf of a user.

Actor: Responding Gateway

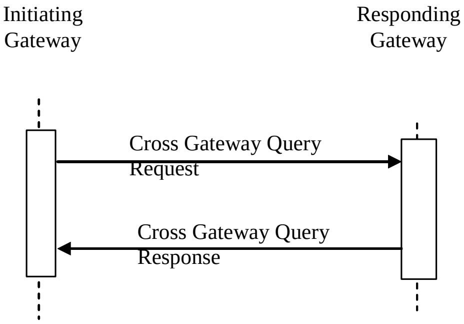
Role: To respond to a Cross Gateway Query based on the internal configuration of the community.

3.38.3 Referenced Standard

Implementers of this transaction shall comply with all requirements described in ITI TF-2: Appendix V Web Services for IHE Transactions.

ebRIM	OASIS/ebXML Registry Information Model v3.0
ebRS	OASIS/ebXML Registry Services Specifications v3.0
Appendix V	ITI TF-2: Appendix V: Web Services for IHE Transactions Contains references to all Web Services standards and requirements of use

3.38.4 Interaction Diagram



3.38.4.1 Cross Gateway Query

This is a query request between an Initiating Gateway and a Responding Gateway. The query request contains:

- A reference to a pre-defined query defined by the Registry Stored Query transaction [ITI-18].
- Parameters to the query. The query parameters are defined by the Registry Stored Query transaction. The homeCommunityId attribute is required for every Registry Stored Query which does not specify a patient identity.

3.38.4.1.1 Trigger Events

This message is initiated when the Initiating Gateway has determined that it must interact with the Responding Gateway to satisfy a Registry Stored Query request received from an XDS.b Document Consumer or a query request from other internal non-IHE actor.

3.38.4.1.2 Message Semantics

The message semantics are based on the Registry Stored Query. See ITI TF-2:3.18.4.1.2. Of special note are the use of homeCommunityId and specifying the patient identifier. These are explained below.

3.38.4.1.2.1 homeCommunityId

The homeCommunityId attribute is required on the Cross Gateway Query. The homeCommunityId 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`.

Its use is as follows:

- It is returned within the response to Cross Gateway Query and Registry Stored Query transactions to indicate the association of a response element with a community. It is specified as the ebRIM 'home' attribute within the relevant response elements. Document Consumers process the value as an opaque unique identifier.
- It is an optional parameter to Registry Stored Query requests, not requiring a patient id parameter, and Retrieve Document Set requests to indicate which community to direct the request.

The Initiating Gateway shall specify the homeCommunityId parameter within all queries which do not include a patient identifier parameter. These would be queries which specify an entryUUID or uniqueID. The homeCommunityId shall contain the value that identifies the communited associated with the Responding Gateway. The homeCommunityId value is specified as the home attribute on the AdhocQuery element of the query request, as in:

```
<AdhocQuery id=”…” home=”urn:oid:1.2.3” … >
```

Each Cross Gateway Query request can have at most one homeCommunityId value. If multiple entryUUID or uniqueID values are specified they must all be associated with the same homeCommunityId value. Multiple individual query requests can be used to retrieve data associated with different homeCommunityIds. This restriction is expected to be temporary as query syntax is extended to handle multiple homeCommunityIds in a single query request.

3.38.4.1.2.2 Specifying patient identifier

The Initiating Gateway shall specify in relevant queries a patient identifier known to the Responding Gateway. The mechanism used by the Initiating Gateway to determine the correct patient identifier to use is outside the intended scope of this profile. The Responding Gateway can expect to be able to resolve the patient identifier. If the patient identifier is unknown by the Responding Gateway's community, the Responding Gateway shall return a successful response with no elements.

3.38.4.1.3 Expected Actions

Actors supporting this transaction shall support the Expected Actions described in ITI TF-2:3.18.4.1.3. In addition:

The Responding Gateway actor shall:

- Verify the homeCommunityId is specified on relevant queries and return an XDSMissingHomeCommunityId error code if missing. Return an XDSUnknownCommunity error code if the value is not known by the Responding Gateway.
- Route the query to the local XDS Document Registry or perform equivalent action to form a query response.
- Ensure that the response contains the value identifying the Responding Gateway's community for the homeCommunityId attribute in every appropriate element. The elements that shall include the ebRIM home attribute are:
    - If returntype="LeafClass" the ExtrinsicObject and RegistryPackage elements shall contain the home attribute.
  - If returnType="ObjectRef" the ObjectRef element shall contain the home attribute.

The Initiating Gateway actor shall:

- On receiving the response from the Responding Gateway, verify the homeCommunityId is present where appropriate. If homeCommunityId is not present in any of the appropriate elements the Initiating Gateway shall reflect an XDSMissingHomeCommunityId to the initiator of the transaction – either the Document Consumer or the internal actor. If homeCommunityId is missing from any element in the response from the Responding Gateway the entire response shall be discarded and the error reflected. The detailed error message associated with the

XDSMissingHomeCommunityId shall include identification of the RespondingGateway that returned the invalid response and the element or elements that were in error.

- If the XDS Affinity Domain Option is supported and if needed, consolidate results from multiple Responding Gateways. This includes reflecting in the consolidated results returned in response to the originating Registry Stored Query [ITI-18] all successes and failures received from Responding Gateways. If both successes and failures are received from Responding Gateways, the Initiating Gateway shall return both RegistryObjectList and RegistryErrorList in one response and specify PartialSuccess status.

3.38.4.1.4 Security Considerations

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

The Initiating Gateway:

- If receiving a Registry Stored Query transaction from a Document Consumer, shall audit as if it were a Document Registry. See Section 3.18.4.2.5.
- In addition, shall audit the Cross Gateway Query as if it were a Document Consumer except that for EventTypeCode the Initiating Gateway shall specify EV(“ITI-38”, “IHE Transactions”, and “Cross Gateway Query”). See Section 3.18.4.2.4.
- In addition, if interacting with a local Document Registry, shall audit as if it were a Document Consumer. See Section 3.18.4.2.5.

The Responding Gateway:

- Shall audit the Cross Gateway Query as if it were a Document Registry except that for EventTypeCode the Responding Gateway shall specify EV(“ITI-38”, “IHE Transactions”, “Cross Gateway Query”). See Section 3.18.4.2.4.
- In addition, if interacting with a local Document Registry, shall audit as if it were a Document Consumer. See Section 3.18.4.2.5.

3.38.5 Protocol Requirements

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

3.38.5-1 WSDL Namespace Definitions

soap	http://schemas.xmlsoap.org/wsdl/soap/
soap12	http://schemas.xmlsoap.org/wsdl/soap12/
wsaw	http://www.w3.org/2006/05/addressing/wsdl/
xsd	http://www.w3.org/2001/XMLSchema
ihe	urn:ihe:iti:xds-b:2007

rs	urn:oasis:names:tc:ebxml-regrep:xsd:rs:3.0
lcm	urn:oasis:names:tc:ebxml-regrep:xsd:lcm:3.0
query	urn:oasis:names:tc:ebxml-regrep:xsd:query:3.0

Initiating Gateway: These are the requirements for the Cross Gateway Query transaction presented in the order in which they would appear in the Initiating 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"
- The /definitions/message/part/@element attribute of the Cross Gateway Query Response message shall be defined as “query:AdhocQueryResponse”
- Refer to table 3.38.5-2 below for additional attribute requirements

3.38.5-2 Additional Attribute Requirements

Attribute	Synchronous Web Services Exchange	Asynchronous Web Services Exchange
/definitions/portType/operation@name	Not applicable	urn:ihe:iti:2007:RespondingGateway_CrossGatewayRetrieveAsyncResponse
/definitions/portType/operation/input/@wsaw:Action	Not applicable	urn:ihe:iti:2007:CrossGatewayRetrieveAsyncResponse
/definitions/binding/operation/soap12:operation/@soapAction	Not applicable	urn:ihe:iti:2007:CrossGatewayRetrieveAsyncResponse

For informative WSDL for the Initiating Gateway actor see Appendix W.

Responding Gateway: These are the requirements for the Cross Gateway Query transaction 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"
- The /definitions/message/part/@element attribute of the Cross Gateway Query Request message shall be defined as “query:AdhocQueryRequest”
- The /definitions/message/part/@element attribute of the Cross Gateway Query Response message shall be defined as “query:AdhocQueryResponse”
- Refer to table 3.38.5-3 below for additional attribute requirements

3.38.5-3 Additional Attribute Requirements

Attribute	Synchronous Web Services Exchange	Asynchronous Web Services Exchange
/definitions/portType/operation@name	urn:ihe:iti:2007:RespondingGateway_CrossGatewayQuery	urn:ihe:iti:2007:RespondingGateway_CrossGatewayQueryAsync
/definitions/portType/operation/input/@wsaw:Action	urn:ihe:iti:2007:CrossGatewayQuery	urn:ihe:iti:2007:CrossGatewayQueryAsync
/definitions/portType/operation/output/@wsaw:Action	urn:ihe:iti:2007:CrossGatewayQueryResponse	Not applicable
/definitions/binding/operation/soap12:operation/@soapAction	urn:ihe:iti:2007:CrossGatewayQuery	urn:ihe:iti:2007:CrossGatewayQueryAsync

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. Full sample request and response messages are in section 3.38.5.1 Sample SOAP Messages.

For informative WSDL for the Responding Gateway actor see Appendix W.

3.38.5.1 Sample SOAP Messages

The samples in the following two sections show a typical SOAP request and its relative SOAP response. The sample messages also show the WS-Addressing headers <Action/>, <MessageID/>, <ReplyTo/>…; these WS-Addressing headers are populated according to the W3C WS-Addressing standard. The body of the SOAP message is omitted for brevity; in a real scenario the empty element will be populated with the appropriate metadata.

Samples presented in this section are also available online on the IHE FTP site, see Appendix W.

3.38.5.1.1 Sample Cross Gateway Query SOAP Request

3.38.5.1.1.1 Synchronous Web Services Exchange

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:a="http://www.w3.org/2005/08/addressing">
  <s:Header>
    <a:Action s:mustUnderstand="1">urn:ihe:iti:2007:CrossGatewayQuery</a:Action>
    <a:MessageID>urn:uuid:def119ad-dc13-49c1-a3c7-e3742531f9b3</a:MessageID>
    <a:ReplyTo>
      <a:Address>http://www.w3.org/2005/08/addressing/anonymous</a:Address>
    </a:ReplyTo>
    <a:To s:mustUnderstand="1">http://localhost/service/IHEXCARespondingGateway.svc</a:To>
  </s:Header>
  <s:Body>
    <query:AdhocQueryRequest xmlns:query="urn:oasis:names:tc:ebxml-regrep:xsd:query:3.0"/>
  </s:Body>
</s:Envelope>
```

3.38.5.1.1.2 Asynchronous Web Services Exchange

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:a="http://www.w3.org/2005/08/addressing">
  <s:Header>
    <a:Action s:mustUnderstand="1">urn:ihe:iti:2007:CrossGatewayQueryAsync</a:Action>
    <a:MessageID>urn:uuid:def119ad-dc13-49c1-a3c7-e3742531f9b3</a:MessageID>
    <a:ReplyTo>
      <a:Address>http://192.168.2.4:9080/XcaService/InitiatingGatewayReceiver.svc</a:Address>
    </a:ReplyTo>
    <a:To s:mustUnderstand="1">http://localhost/XcaService/RespondingGatewayReceiver.svc</a:To>
  </s:Header>
  <s:Body>
    <query:AdhocQueryRequest xmlns:query="urn:oasis:names:tc:ebxml-regrep:xsd:query:3.0"/>
  </s:Body>
</s:Envelope>
```

3.38.5.1.2 Sample Cross Gateway Query SOAP Response

3.38.5.1.2.1 Synchronous Web Services Exchange

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:a="http://www.w3.org/2005/08/addressing">
  <s:Header>
    <a:Action s:mustUnderstand="1">urn:ihe:iti:2007:CrossGatewayQueryResponse</a:Action>
    <a:RelatesTo>urn:uuid:def119ad-dc13-49c1-a3c7-e3742531f9b3</a:RelatesTo>
  </s:Header>
  <s:Body>
    <query:AdhocQueryResponse xmlns:query="urn:oasis:names:tc:ebxml-regrep:xsd:query:3.0"/>
  </s:Body>
</s:Envelope>
```

3.38.5.1.2.2 Asynchronous Web Services Exchange

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:a="http://www.w3.org/2005/08/addressing">
  <s:Header>
    <a:Action s:mustUnderstand="1">urn:ihe:iti:2007:CrossGatewayQueryAsyncResponse</a:Action>
     <a:MessageID>urn:uuid:D6C21225-8E7B-454E-9750-821622C099DB</a:MessageID>
    <a:RelatesTo>urn:uuid:def119ad-dc13-49c1-a3c7-e3742531f9b3</a:RelatesTo>
     <a:To s:mustUnderstand="1">http://localhost:2647/XcaService/InitiatingGatewayReceiver.svc
     </a:To>
</s:Header>
  <s:Body>
    <query:AdhocQueryResponse xmlns:query="urn:oasis:names:tc:ebxml-regrep:xsd:query:3.0"/>
  </s:Body>
</s:Envelope>
```

3.39 Cross Gateway Retrieve

This section corresponds to Transaction 39 of the IHE Technical Framework. Transaction 39 is used by the Initiating Gateway and Responding Gateway actors.

3.39.1 Scope

The scope of the Cross Gateway Retrieve transaction is semantically the same as the Retrieve Document Set transaction [ITI-43]. Differences from the Retrieve Document Set transactions are:

- The Cross Gateway Retrieve is between an Initiating Gateway and a Responding Gateway.
- The ‘homeCommunityId’ parameter is required. This means that the homeCommunityId parameter which is optional on the Retrieve Document Set transaction is required by this transaction.
- Responding Gateways shall support the Asynchronous Web Services Exchange Option on the Cross Gateway Retrieve. Support for this function is required in order to enable use of Asynchronous Web Services Exchange in any cross-community interaction. Without this support an Initiating Gateway would require unique configuration, per Responding Gateway, to know if Asynchronous Web Services Exchange was supported. It is expected that Asynchronous Web Services Exchange will be desired by the majority of communities.
- Asynchronous Web Services Exchange is an option on the Initiating Gateway, see ITI TF-1:1:18.2.2.

3.39.2 Use Case Roles

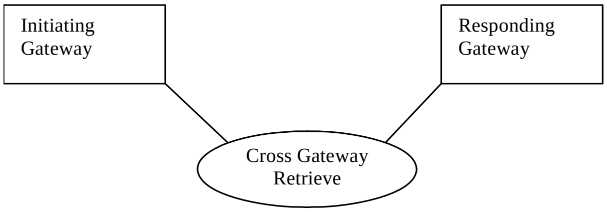


Figure 3.39.2: Use Case Roles

Actor: Initiating Gateway

Role: To formulate a Cross Gateway Retrieve in response to Retrieve Document Set transactions or other internal interaction.

Actor: Responding Gateway

Role: To return the documents requested.

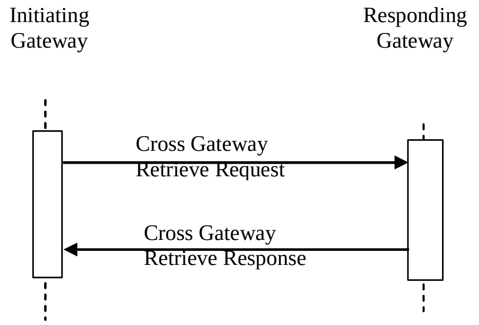
3.39.3 Referenced Standard

Implementors of this transaction shall comply with all requirements described in ITI TF-2:Appendix V Web Services for IHE Transactions.

ebRIM	OASIS/ebXML Registry Information Model v3.0
ebRS	OASIS/ebXML Registry Services Specifications v3.0
Appendix V	ITI TF-2:Appendix V: Web Services for IHE Transactions Contains references to all Web Services standards and requirements of use
MTOM	SOAP Message Transmission Optimization Mechanism

http://www.w3.org/TR/soap12-mtom/

3.39.4 Interaction Diagram


Initiating Gateway
Responding Gateway
Cross Gateway Retrieve Request
Cross Gateway Retrieve Response

3.39.4.1 Cross Gateway Retrieve

The Cross Gateway Retrieve uses the same syntax and standards as the Retrieve Document Set transaction specified in XDS.b. See Section 3.43.

3.39.4.1.1 Trigger Events

This message is initiated by the Initiating Gateway to retrieve a set of documents from another community represented by a Responding Gateway. The Initiating Gateway may be responding to a Retrieve Document Set transaction or may use a proprietary mechanism for triggering the Cross Gateway Retrieve.

3.39.4.1.2 Message Semantics

The message semantics for Cross Gateway Retrieve are the same as Retrieve Document Set. See Section 3.43 4.1.2. The Initiating Gateway shall specify the homeCommunityId parameter within the Retrieve Document Set. The homeCommunityId shall contain the value that identifies the community associated with the Responding Gateway.

3.39.4.1.3 Expected Actions

Actors supporting this transaction shall support the Expected Actions described in the Section 3.43.4.1.3.

The Responding Gateway shall determine the local system or systems which hold the documents requested and interact with those systems. The Responding Gateway may use a Retrieve Document Set transaction or other internally defined interaction, to retrieve the document or documents. If more than one system is contacted the Responding Gateway shall consolidate the results from the multiple systems into one response to the Initiating Gateway. If both successes and failures are received the Responding Gateway may choose to use PartialSuccess status to reflect both failure and success. The Responding Gateway may alternatively choose to suppress the failures and report only successes.

If the XDS Affinity Domain Option is supported, the Initiating Gateway actor shall, if needed, consolidate results from multiple Responding Gateways. This includes reflecting in the consolidated results returned to the originating Retrieve Document Set [ITI-43] all successes and failures received from Responding Gateways. If both successes and failures are received from Responding Gateways, the Initiating Gateway shall return both DocumentResponse and RegistryErrorList elements in one response and specify PartialSuccess status.

3.39.4.1.4 Security Considerations

Both the Initiating Gateway and Responding Gateway shall audit the Cross Gateway Retrieve. The audit entries shall be equivalent to the entries required for the Retrieve Document Set.

The Initiating Gateway:

- If receiving a Retrieve Document Set transaction from a Document Consumer, shall audit as if it were a Document Repository. See Section 3.43.6.
- In addition, shall audit the Cross Gateway Retrieve as if it were a Document Consumer except that for EventTypeCode the Initiating Gateway shall specify EV(“ITI-39”, “IHE Transactions”, and “Cross Gateway Retrieve”). See Section 3.43.6.
- In addition, if interacting with a local Document Repository, shall audit as if it were a Document Consumer. See Section 3.43.6. One audit record shall be created for each Document Repository contacted.

The Responding Gateway:

- Shall audit the Cross Gateway Retrieve as if it were a Document Repository except that for EventTypeCode the Responding Gateway shall specify EV(“ITI-39”, “IHE Transactions”, “Cross Gateway Retrieve”). See Section 3.43.6.
- In addition, if interacting with a local Document Repository, shall audit as if it were a Document Consumer. See Section 3.43.6. One audit record shall be created for each Document Repository contacted.

3.39.5 Protocol Requirements

The Cross Gateway Retrieve request and response will be transmitted using Synchronous or Asynchronous Web Services Exchange, according to the requirements specified in ITI TF-2:Appendix V. The protocol requirements are identical to the Retrieve Document Set except as noted below.

3.39.5-1 WSDL Namespace Definitions

soap	http://schemas.xmlsoap.org/wsdl/soap/
soap12	http://schemas.xmlsoap.org/wsdl/soap12/
wsaw	http://www.w3.org/2006/05/addressing/wsdl/

xsd	http://www.w3.org/2001/XMLSchema
ihe	urn:ihe:iti:xds-b:2007
rs	urn:oasis:names:tc:ebxml-regrep:xsd:rs:3.0
lcm	urn:oasis:names:tc:ebxml-regrep:xsd:lcm:3.0
query	urn:oasis:names:tc:ebxml-regrep:xsd:query:3.0

Initiating Gateway: These are the requirements for the Cross Gateway Retrieve transaction presented in the order in which they would appear in the Initiating Gateway WSDL definition:

- The following types shall be imported (xsd:import) in the /definitions/types section:
    - namespace="urn:ihe:iti:xds-b:2007", schema="IHEXDS.xsd"
- The /definitions/message/part/@element attribute of the Cross Gateway Retrieve Request message shall be defined as “ihe:RetrieveDocumentSetRequest”
- The /definitions/message/part/@element attribute of the Cross Gateway Retrieve Response message shall be defined as “ihe:RetrieveDocumentSetResponse”
- Refer to table 3.39.5-2 below for additional attribute requirements

3.39.5-2 Requirements for portType and Binding attributes

Attribute	Synchronous Web Services Exchange	Asynchronous Web Services Exchange
/definitions/portType/operation@name	Not applicable	urn:ihe:iti:2007:RespondingGateway_CrossGatewayRetrieveAsyncResponse
definitions/portType/operation/input/@wsaw:Action	Not applicable	urn:ihe:iti:2007:CrossGatewayRetrieveAsyncResponse
/definitions/binding/operation/soap12:operation/@soapAction	Not applicable	urn:ihe:iti:2007:CrossGatewayRetrieveAsyncResponse

For informative WSDL for the Initiating Gateway actor see Appendix W.

Responding Gateway: These are the requirements for the Cross Gateway Retrieve transaction 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:ihe:iti:xds-b:2007", schema="IHEXDS.xsd"
- The /definitions/message/part/@element attribute of the Cross Gateway Retrieve Request message shall be defined as “ihe:RetrieveDocumentSetRequest”
- The /definitions/message/part/@element attribute of the Cross Gateway Retrieve Response message shall be defined as “ihe:RetrieveDocumentSetResponse”
- Refer to table 3.39.5-3 below for additional attribute requirements

3.39.5-3 Requirements for portType and Binding attributes

Attribute	Synchronous Web Services Exchange	Asynchronous Web Services Exchange
/definitions/portType/operation@name	urn:ihe:iti:2007:RespondingGateway_CrossGatewayRetrieve	urn:ihe:iti:2007:RespondingGateway_CrossGatewayRetrieveAsync
/definitions/portType/operation/input/@wsaw:Action	urn:ihe:iti:2007:CrossGatewayRetrieve	urn:ihe:iti:2007:CrossGatewayRetrieveAsync
/definitions/portType/operation/output/@wsaw:Action	urn:ihe:iti:2007:CrossGatewayRetrieveResponse	Not applicable
/definitions/binding/operation/soap12:operation/@soapAction	urn:ihe:iti:2007:CrossGatewayRetrieve	urn:ihe:iti:2007:CrossGatewayRetrieveAsync

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. Full sample request and response messages are in section 3.39.5.1 Sample SOAP Messages.

For informative WSDL for the Responding Gateway actor see Appendix W.

The <ihe:RetrieveDocumentSetRequest/> element is defined in ITI TF-2:3.43.5. When used within the Cross Gateway Retrieve the <ihe:HomeCommunityId/> element is required.

The <ihe:RetrieveDocumentSetResponse/> element is defined in ITI TF-2:3.43.5.

3.39.5.1 Sample SOAP Messages

The samples in the following two sections show a typical SOAP request and its relative SOAP response. The sample messages also show the WS-Addressing headers <Action/>, <MessageID/>, <ReplyTo/>…; these WS-Addressing headers are populated according to the W3C WS-Addressing standard. The body of the SOAP message is omitted for brevity; in a real scenario the empty element will be populated with the appropriate metadata.

Samples presented in this section are also available online on the IHE FTP site, see Appendix W.

3.39.5.1.1 Sample Cross Gateway Retrieve SOAP Request

3.39.5.1.1.1 Synchronous Web Services Exchange

```
<s:Envelope
                xmlns:s="http://www.w3.org/2003/05/soap-envelope"
                xmlns:a="http://www.w3.org/2005/08/addressing">
        <s:Header>
                <a:Action s:mustUnderstand="1">urn:ihe:iti:2007:CrossGatewayRetrieve</a:Action>
                <a:MessageID>urn:uuid:0fbfdced-6c01-4d09-a110-2201afedaa02</a:MessageID>
                <a:ReplyTo>
                        <a:Address>http://www.w3.org/2005/08/addressing/anonymous</a:Address>
                </a:ReplyTo>
                <a:To
s:mustUnderstand="1">http://localhost:2647/XcaService/IHEXCAGateway.svc</a:To>
        </s:Header>
        <s:Body>
                <RetrieveDocumentSetRequest xmlns="urn:ihe:iti:xds-b:2007">
                        <DocumentRequest>
                                <homeCommunityId>urn:oid:1.2.3.4</homeCommunityId>
                                <repositoryUniqueId>1.3.6.1.4...1000</repositoryUniqueId>
                                <documentUniqueId>1.3.6.1.4...2300</documentUniqueId>
                        </DocumentRequest>
                        <DocumentRequest>
                                <homeCommunityId>urn:oid:1.2.3.5</homeCommunityId>
                                <repositoryUniqueId>1.3.6.1.4...2000</repositoryUniqueId>
                                <documentUniqueId>1.3.6.1.4...2301</documentUniqueId>
                        </DocumentRequest>
                </RetrieveDocumentSetRequest>
        </s:Body>
</s:Envelope>
```

3.39.5.1.1.2 Asynchronous Web Services Exchange

```
<s:Envelope
                xmlns:s="http://www.w3.org/2003/05/soap-envelope"
                xmlns:a="http://www.w3.org/2005/08/addressing">
        <s:Header>
                <a:Action
s:mustUnderstand="1">urn:ihe:iti:2007:CrossGatewayRetrieveAsync</a:Action>
                <a:MessageID>urn:uuid:0fbfdced-6c01-4d09-a110-2201afedaa02</a:MessageID>
                <a:ReplyTo>

        <a:Address>http://192.168.2.4:9080/XcaService/InitiatingGatewayReceiver.svc
                        </a:Address>
                </a:ReplyTo>
                <a:To
s:mustUnderstand="1">http://localhost:2647/XcaService/RespondingGatewayReceiver.svc</a:To>
        </s:Header>
        <s:Body>
                <RetrieveDocumentSetRequest xmlns="urn:ihe:iti:xds-b:2007">
                        <DocumentRequest>
                                <homeCommunityId>urn:oid:1.2.3.4</homeCommunityId>
                                <repositoryUniqueId>1.3.6.1.4...1000</repositoryUniqueId>
                                <documentUniqueId>1.3.6.1.4...2300</documentUniqueId>
                        </DocumentRequest>
                        <DocumentRequest>
                                <homeCommunityId>urn:oid:1.2.3.5</homeCommunityId>
                                <repositoryUniqueId>1.3.6.1.4...2000</repositoryUniqueId>
                                <documentUniqueId>1.3.6.1.4...2301</documentUniqueId>
                        </DocumentRequest>
                </RetrieveDocumentSetRequest>
        </s:Body>
</s:Envelope>
```

3.39.5.1.2 Sample Cross Gateway Retrieve SOAP Response

3.39.5.1.2.1 Synchronous Web Services Exchange

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:a="http://www.w3.org/2005/08/addressing">
        <s:Header>
                <a:Action
s:mustUnderstand="1">urn:ihe:iti:2007:CrossGatewayRetrieveResponse</a:Action>
                <a:RelatesTo>urn:uuid:0fbfdced-6c01-4d09-a110-2201afedaa02</a:RelatesTo>
        </s:Header>
        <s:Body>
                <RetrieveDocumentSetResponse
                                xmlns="urn:ihe:iti:xds-b:2007"
                                xmlns:lcm="urn:oasis:names:tc:ebxml-regrep:xsd:lcm:3.0"
                                xmlns:query="urn:oasis:names:tc:ebxml-regrep:xsd:query:3.0"
                                xmlns:rim="urn:oasis:names:tc:ebxml-regrep:xsd:rim:3.0"
                                xmlns:rs="urn:oasis:names:tc:ebxml-regrep:xsd:rs:3.0">
                        <rs:RegistryResponse status="urn:oasis:names:tc:ebxml-
regrep:ResponseStatusType:Success"/>
                        <DocumentResponse>
                                <homeCommunityId>urn:oid:1.2.3.4</homeCommunityId>
                                <repositoryUniqueId>1.3.6.1.4...1000</repositoryUniqueId>
                                <documentUniqueId>1.3.6.1.4...2300</documentUniqueId>
                                <mimeType>text/xml</mimeType>

        <Document>UjBsR09EbGhjZ0dTQUxNQUFBUUNBRU1tQ1p0dU1GUXhEUzhi</Document>
                        </DocumentResponse>
                        <DocumentResponse>
                                <homeCommunityId>urn:oid:1.2.3.5</homeCommunityId>
                                <repositoryUniqueId>1.3.6.1.4...2000</repositoryUniqueId>
                                <documentUniqueId>1.3.6.1.4...2301</documentUniqueId>
                                <mimeType>text/xml</mimeType>

        <Document>UjBsR09EbGhjZ0dTQUxNQUFBUUNBRU1tQ1p0dU1GUXhEUzhi</Document>
                        </DocumentResponse>
                </RetrieveDocumentSetResponse>
        </s:Body>
</s:Envelope>
```

3.39.5.1.2.2 Asynchronous Web Services Exchange

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:a="http://www.w3.org/2005/08/addressing">
        <s:Header>
                <a:Action
s:mustUnderstand="1">urn:ihe:iti:2007:CrossGatewayRetrieveAsyncResponse</a:Action>
                <a:MessageID>urn:uuid:D6C21225-8E7B-454E-9750-821622C099DB</a:MessageID>
                <a:RelatesTo>urn:uuid:0fbfdced-6c01-4d09-a110-2201afedaa02</a:RelatesTo>
                <a:To
s:mustUnderstand="1">http://localhost:2647/XcaService/InitiatingGatewayReceiver.svc</a:To>
        </s:Header>
        <s:Body>
                <RetrieveDocumentSetResponse
                                xmlns="urn:ihe:iti:xds-b:2007"
                                xmlns:lcm="urn:oasis:names:tc:ebxml-regrep:xsd:lcm:3.0"
                                xmlns:query="urn:oasis:names:tc:ebxml-regrep:xsd:query:3.0"
                                xmlns:rim="urn:oasis:names:tc:ebxml-regrep:xsd:rim:3.0"
                                xmlns:rs="urn:oasis:names:tc:ebxml-regrep:xsd:rs:3.0">
                        <rs:RegistryResponse status="urn:oasis:names:tc:ebxml-
regrep:ResponseStatusType:Success"/>
                        <DocumentResponse>
                                <homeCommunityId>urn:oid:1.2.3.4</homeCommunityId>
                                <repositoryUniqueId>1.3.6.1.4...1000</repositoryUniqueId>
                                <documentUniqueId>1.3.6.1.4...2300</documentUniqueId>
                                <mimeType>text/xml</mimeType>

        <Document>UjBsR09EbGhjZ0dTQUxNQUFBUUNBRU1tQ1p0dU1GUXhEUzhi</Document>
                        </DocumentResponse>
                        <DocumentResponse>
                                <homeCommunityId>urn:oid:1.2.3.5</homeCommunityId>
                                <repositoryUniqueId>1.3.6.1.4...2000</repositoryUniqueId>
                                <documentUniqueId>1.3.6.1.4...2301</documentUniqueId>
                                <mimeType>text/xml</mimeType>

        <Document>UjBsR09EbGhjZ0dTQUxNQUFBUUNBRU1tQ1p0dU1GUXhEUzhi</Document>
                        </DocumentResponse>
                </RetrieveDocumentSetResponse>
        </s:Body>
</s:Envelope>
```

Update Error Reporting section of the ITI TF (4.1.2.13) to add two new error codes to Table 4.1-11. NOTE: this section assumes the updates to this table specified in the XDS.b Supplement are applied first. Because the list of transactions has grown lengthy they are moved outside the table heading for this update.

Table 4.1-11 – Error Codes

Error Code	Discussion	Transaction (See Note 1)
XDSUnknownCommunity	**A value for the homeCommunityId is not recognized**	**SQ, XGQ, RS, XGR**
XDSMissingHomeCommunityId	**A value for the homeCommunityId is required and has not been specified**	**SQ, XGQ, RS, XGR**
XDSUnavailableCommunity	**A community which would have been contacted was not available . See Note 2.**	**SQ, RS**

Note 1:

P = Provide and Register, Provide and Register-b

R = Register, Register-b

Q = Query

SQ = Stored Query

RS = Retrieve Document Set

XGQ = Cross Gateway Query

XGR = Cross Gateway Retrieve

Note 2:

Two examples of the use of error code XDSUnavailableCommunity are:

1. A Cross Gateway Query or Cross Gateway Retrieve fail because the community identified by a homeCommunityId could not be contacted.
2. A Cross Gateway Query based on Patient ID could not contact some known communities to relay the query.

The error would be generated by the Initiating Gateway and returned in the Registry Stored Query or Retrieve Document Set. This would only apply when XDS Affinity Domain Option was used.