**Integrating the Healthcare Enterprise**



**IHE ITI**

**Technical Framework Supplement**

**XCA Deferred Response**

**XCA Support For Two 2-way Messaging**

**Revision x.x – Draft in Preparation for Public Comment**

<The IHE Documentation Specialist will change the title to just “Draft for Public Comment” or “Trial Implementation” upon publication. Leave “as is” until then.>

Date: <Month xx, 20xx>

Author: Joseph Lamy, ITI Technical Committee

Email: <domain\_name@ihe.net>

**Please verify you have the most recent version of this document.** See [here](http://ihe.net/Technical_Frameworks/) for Trial Implementation and Final Text versions and [here](http://ihe.net/Public_Comment/) for Public Comment versions.

**Foreword**

This is a supplement to the IHE IT Infrastructure Technical Framework <VX.X>. Each supplement undergoes a process of public comment and trial implementation before being incorporated into the volumes of the Technical Frameworks.

*<For Public Comment:>* This supplement is published on <Month XX, 201x> for Public Comment. Comments are invited and can be submitted at <http://www.ihe.net/Public_Comment/#domainname>. In order to be considered in development of the Trial Implementation version of the supplement, comments must be received by <Month XX, 201X>.

*<For Trial Implementation:>* This supplement is published on <Month XX, 201X> for Trial Implementation and may be available for testing at subsequent IHE Connectathons. The supplement may be amended based on the results of testing. Following successful testing it will be incorporated into the <Domain Name> Technical Framework. Comments are invited and can be submitted at <http://www.ihe.net/Public_Comment/#domainname>.

This supplement describes changes to the existing technical framework documents.

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

Amend section X.X by the following:

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

General information about IHE can be found at [www.ihe.net](http://www.ihe.net/).

Information about the IHE <Domain Name> domain can be found at [ihe.net/IHE\_Domains](http://ihe.net/IHE_Domains/).

Information about the organization of IHE Technical Frameworks and Supplements and the process used to create them can be found at [http://ihe.net/IHE\_Process](http://ihe.net/IHE_Process/) and [http://ihe.net/Profiles](http://ihe.net/Profiles/).

The current version of the IHE <Domain name>Technical Framework can be found at [http://ihe.net/Technical\_Frameworks](http://ihe.net/Technical_Frameworks/).

*<Comments may be submitted on IHE Technical Framework templates any time at* [*http://ihe.net/Templates\_Public\_Comments*](http://ihe.net/Templates_Public_Comments/)*. Please enter comments/issues as soon as they are found. Do not wait until a future review cycle is announced.>*

CONTENTS

Introduction to this Supplement 5

Open Issues and Questions 5

Closed Issues 7

Volume 1 – Profiles 15

10.2 XDS.b Actor Options 15

10.2.X Deferred Response Option 15

18.2 XCA Integration Profile Options 16

18.2.X Deferred Response Option 16

18.3.3.X Use of Deferred Responses 17

Volume 2a – Transactions 19

3.18.4 Interaction Diagram 19

3.18.4.1.2.5 Compatibility of Options 19

3.18.4.1.3 Expected Actions 20

3.18.4.1.3.7 AdhocQueryResponse Indicating Deferred Processing 21

3.18.4.2 Registry Stored Query Deferred Results 22

3.18.4.2.1 Trigger Events 23

3.18.4.2.2 Message Semantics 23

3.18.4.2.2.1 Web Services Transport 23

3.18.4.2.2.1.1 Sample SOAP Messages 26

3.18.4.2.3 Expected Actions 29

Volume 2b – Transactions 31

3.38.4 Interaction Diagram 31

3.38.4.1.2 Message Semantics 32

3.38.4.2 Cross Gateway Query Deferred Results 34

3.38.4.2.1 Trigger Events 34

3.38.4.2.2 Message Semantics 34

3.38.4.2.3 Expected Actions 34

# Introduction to this Supplement

This supplement specifies the Deferred Response option to the Cross Gateway Query and Cross Gateway Retrieve transactions defined in the XCA profile, as well as the Registry Stored Query and Retrieve Document Set transactions defined in the XDS.b profile.

The Deferred Response Option is needed when the responding system requires extensive time for processing, as much as days or weeks. This need could arise through delegation of processing to external/remote systems or human interaction. The mechanism enables this by splitting the request and response into two separate two-way exchanges. This requires applications to manage the message correlation, create application level acknowledgements and determine where to send the Deferred Response message. For more information about Deferred Response and Asynchronous messaging in general see <http://wiki.ihe.net/index.php?title=Asynchronous_Messaging>.

Given that there are multiple mechanisms available for longer latency query and retrieve, Deferred Response should be used in the following cases:

* When the delay in responding may be as much as days or weeks.
* When applications must support recovery of the long-running request and response through system restart.

If neither of these cases applies and longer latency query and retrieve is needed, consider either the WS-Addressing-Based Web Services Exchange Option or the AS4 Asynchronous Web Services Exchange Option.

## Open Issues and Questions

* Should canceling an ongoing deferred request be supported?
  + Think of the following examples:
    - A request is accidentally sent that is too broad.
    - An initial response indicates an estimated duration that is considered too long, and the requester may wish to use other means or contact the responder to make sure there isn’t a problem and unnecessary work isn’t done.
    - A responder performs value-added services for an additional charge using the deferred mechanism and returns the charge amount in the explanation of the initial response, and the requester does not wish to incur this charge.
  + Is canceling a response supported in AS4?
* Should the requester be able to pass a value to indicate how long they would wait for responses?
* The Responding Gateway has a lot of flexibility to return results, e.g. as soon as each result is ready, in groups on some schedule, or waiting until all are ready. Should the Initiating Gateway have similar flexibility in returning results to Document Consumers from multiple Responding Gateways?
  + **(Tentative answer) No.** This flexibility was given to the Responding Gateway due to the fact that the internal mechanisms by which it receives long-latency results and assembles them into results messages are not specified. There could be arbitrarily complex workflows involved, as well as preferences negotiated partner-to-partner similar to subscribing to a mailing list. The Initiating Gateway, by contrast, does not appear to have a similar need for this flexibility – it can simply pass along Deferred responses as they come in.
* Currently, the Initiating Gateway consolidates all estimates of Deferred processing from Responding Gateways into a single value, which it returns to a Document Consumer. Should there be a more detailed way to report individual estimates for each Responding Gateway?
* Should an alternate approach to XCA Deferred be used to handle long latency query and retrieve?
  + (See Closed issues for alternates that were already were considered and rejected)
  + Option: AS4-style Asynchronous messaging
    - In this alternative, the interested party would use the AS4 asynchronous mechanism to send XCA transactions.
    - We would likely keep some additional profiling for these use cases.
      * In the requests, we may not need to pass anything extra.
      * In the responses, we would add the slots to indicate deferred processing.
      * We would allow multiple responses to the original request.
    - Yet to be determined is whether we would replace the implementation of the Deferred Results message with the AS4 response, or allow two implementations: one using the the Deferred Results message and one using the AS4 response.
      * The decision could be made on web stack support for AS4.
    - Pros
      * Existing mechanism; no new specifications development.
      * Some degree of longer latency supported.
    - Cons
      * With its significantly distinct usage of the SOAP header, it is better suited for an environment in which all web services operate in this way.
    - Questions
      * Is AS4 suitable for handle long latencies (e.g. days/weeks)? Closed Issue AS4-5 says no, but others say yes.
      * Can AS4 support multiple responses to a single request?

## Closed Issues

* Should an alternate approach to XCA Deferred be used to handle long latency query and retrieve?
  + No. The following alternates were considered and rejected:
  + Option: DSUB
    - In this alternative, the interested party would subscribe to notifications for the patient using the same filters they would use for a document query.
    - Either notification pushes or stored notification pulls could be used.
    - Pros
      * Existing mechanism.
      * Piecemeal responses: the interested party could receive documents as soon as they are available, rather than waiting for all to be completed.
    - Cons
      * **DSUB doesn't support all the parameters XCA query does**. Specifically, it doesn't support date-related queries. The interested parties that initially motivated this work item require date range queries.
      * **DSUB doesn't support long-latency retrieve**, which was the second use case agreed to in scope.
      * No way to explicitly know when the “query” is “done”: For the use cases (e.g. legacy paper) where the requester wants clinical information that exists as of a point in time, but has not yet been assembled into XCA/XDS formats, there is the notion of “done”, representing when all such clinical information is available to be returned. There is no way to express “done” explicitly.
        + Option: Use Service-Level Agreements (SLAs) to implicitly enforce completion of all documents or high-priority documents within a certain time.
        + Option: Add some way for the Subscriber to indicate they want to treat this subscription as an atomic transaction, to “register all documents that are available at this time”. Add a way for the Broker to notify when this is complete, or a way for the Broker to cancel the subscription when complete and notify of that.
        + Option: Add a workflow that contains the explicit notification (see next option).
      * Cross-community versions of these transactions may have to be defined: at least Subscribe and Notify.
      * All participants would have to know responder requires DSUB. Note that this would also be the case for XCA Deferred.
  + Option: DSUB+XDW
    - In this alternative, the interested party would subscribe to notifications for the patient using the same filters they would use for a document query, but would also trigger a specific workflow task representing a single long-latency query.
    - The first document to be returned would be an instance of this XDW workflow document, containing some identifying information for the query (maybe the WS-Addressing MessageID), and a status of “working”.
    - Once all clinical documents for this query have been returned, the responder would return the workflow document with a status of “complete”.
    - Option: Trigger the workflow by requesting a specific format code.
      * There is already a format code for workflow documents: urn:ihe:iti:xdw:2011:workflowDoc. Because Responding Gateways may have other uses for workflow documents, we would need to define a more specific format code.
    - Option: Trigger the workflow by requesting a predefined document ID. The problem with this is that other requesters may also be triggering this same mechanism, so the ID would have to be treated as a pseudo-ID, and a unique document ID would need to be returned for the workflow document.
    - Option: Trigger the workflow by extending the subscription format using WS-Notification extensibility mechanisms, perhaps by passing a flag requesting a workflow document for status of the query, for example:

<wsnt:Subscribe>

<wsnt:ConsumerReference>

...

<TriggerSingleQueryWorkflow>

* + - Pros
      * Existing mechanism: minimal additional specification development needed.
      * Piecemeal responses: the interested party could receive documents as soon as they are available, rather than waiting for all to be completed.
    - Cons
      * Would need to define new workflow document format.
      * Would need to define triggering mechanism for the workflow.
    - **As this depends on DSUB, it shares those shortcomings. In addition, the Technical Committee feels this option may be too complex.**
  + Option: WS-Addressing-style Asynchronous messaging
    - In this alternative, the interested party would use the WS-Addressing asynchronous mechanism to send an XCA Cross Gateway Query.
    - Pros
      * Existing mechanism; no new specifications development.
      * Some degree of longer latency supported.
    - Cons – WS-Addressing-style async is handled within the web stack and has known limitations as typically implemented:
      * It is non-blocking for the requester but not the responder.
      * **It cannot handle long latencies (e.g. days)** or maintain requests across system restarts.
      * It does not allow management of acknowledgements at the application layer.
      * See the US CONNECT team [analysis](https://connectopensource.atlassian.net/wiki/spaces/CONNECTWIKI/pages/8585329/Asynchronous+Messaging+Engineering+Analysis), as well as the original IHE [white paper](ftp://ftp.ihe.net/IT_Infrastructure/iheitiyr8-2010-2011/Technical_Cmte/Profile_Work/DeferredMsging/IHE_ITI_WhitePaper_Async.0810.doc) that justified the deferred mechanism.
  + Option: XDR+XDW
    - In this alternative, the interested party would push two documents using XDR: a workflow document representing a single long-latency query, and a custom document containing clinical document filters, similar to a query or subscription.
    - The workflow document would be similar to that used in DSUB+XDW.
    - Pros
      * Compared to DSUB, may be more straightforward to implement for systems that already support XDR but not DSUB.
    - Cons
      * Would need to define new mechanism for query-subscription filter, when existing ones exist for pull (XDS) and push (DSUB).
      * Would need to define new workflow document format.
      * **Technical Committee considered this option too novel and complex, as it did not take sufficient advantage of existing standards.**
  + Option: HL7 FHIR subscriptions and DocumentReference
    - In this alternative, the interested party would use FHIR subscription mechanisms to subscribe to documents for a patient.
    - Pros
      * Existing mechanism; no new specifications development.
    - Cons
      * **Technical Committee felt that this work item was intended to target communities that use IHE web services profiles rather than FHIR.**
* Will a Deferred Response option also need to be added to the XDS.b profile, due to groupings with XCA?
  + **Yes, but not for all groupings at this time.** The following groupings were considered:
  + Responding Gateway, grouped with a Document Consumer.
    - In this case, the Responding Gateway receives a deferred request, and needs to initiate ITI-18 or ITI-43 request(s) from Document Consumer(s). Do they also have to be deferred?
    - **The committee decided no**. Not all federated requests would even need long latencies. For those that did, alternative mechanisms such as WS-Addressing style or AS4 style Asynchronous options, polling, or other unspecified mechanisms may be used.
  + Initiating Gateway, grouped with a Document Consumer.
    - In this case, the Initating Gateway receives a deferred request, and needs to initiate ITI-18 or ITI-43 request(s) from local Document Consumer(s). Do they also have to be deferred?
    - **The committee decided no**, using the same reasoning as the previous case.
  + **Initiating Gateway, which supports the XDS Affinity Domain Option.**
    - In this case, the Initating Gateway receives ITI-18 or ITI-43 request(s) from Document Consumer(s), and needs to use the Deferred option on ITI-38 or ITI-39 to Responding Gateway(s).
    - **The committee decided yes. This case cannot easily be supported without a Deferred option on ITI-18 and ITI-43**. The long latencies possible in the XCA transactions will prevent even use of asynchronous mechanisms.
    - **Chosen option: Add Deferred Response options for ITI-18 and ITI-43, only in this XCA context.**
    - Option: Explicitly limit the Deferred option to triggering by Initiating Gateways through internal mechanisms, not triggering by ITI-18 or ITI-43.
    - Option: Allow the Initiating Gateway to deal with this case internally.
      * Initiating Gateway could simply not include Responding Gateways that require Deferred in outgoing requests triggered by ITI-18 or ITI-43.
      * Initiating Gateway could immediately return an error code like XDSRegistryNotAvailable, XDSRegistryBusy, or XDSRegistryError, while triggering a deferred request.
      * Note: See above notes on how Initiating Gateways know whether to use Deferred for Responding Gateways.
* Should this Vol 1 content be written towards the 2018 published content, or towards the AS4 supplement? AS4 changed how XDS.b Document Consumer groups with XCA Initiating Gateway.
  + **Currently written towards the 2018 framework.**
* Should this supplement include its own explicit use cases in Vol 1? The XCPD Deferred Response Option does not.
  + **Yes, a single diagram and process flow to incorporate Cases 1 and 3 below to reflect a high latency Responding Gateway.** All use cases in scope are shown:
  + Case 1: A document source has ownership / access to a very large number of paper-based clinical documents, and wishes to provide these documents electronically using IHE Document Sharing profiles. However, it does not know a priori which of these documents will be requested, and it is not cost effective to proactively scan, parse and register all documents. As each step in the workflow potentially involves this manual process, which could take hours to days, the system may need to make use of deferred responses for XCPD, XCA Query, and XCA Retrieve:
    - Initiating Gateway sends deferred XCPD request.
    - User at responding system searches for patient records, determines matches, and triggers deferred XCPD response.
    - Initiating Gateway sends deferred XCA Query request.
    - User at responding system searches for appropriate clinical documents, scans, parses or otherwise generates document metadata, registers document entries, potentially stores documents, and triggers deferred XCA response.
    - Initiating Gateway sends deferred XCA Retrieve request.
    - User at responding system performs any remaining tasks to make documents available, and triggers deferred XCA response.
  + Case 2: A Responding Gateway uses an on-demand document entry to generate a comprehensive longitudinal record (e.g. a CCD) for a patient from a number of sources. In some cases, the time needed to generate this document runs into the minutes, causing Initiating Gateways to time out. This system would make use of deferred responses for XCA Retrieve:
    - Initiating Gateway sends deferred XCA Retrieve request for on-demand entry.
    - Responding Gateway compiles the on-demand document from various sources and triggers deferred XCA response.
  + Case 3: The access decision for a given set of documents and a given requester takes significant time, due to a human in the loop.
    - Initiating Gateway sends deferred XCA Query request.
    - User at responding system evaluates access decision, confirms this requester may access, and triggers deferred XCA response. Note that this may result in a PartialSuccess, where the requester may only see a subset of all documents matching the query.
    - Initiating Gateway sends synchronous XCA Retrieve request and receives documents.
* Should this work item include combining with the AS4-style Asynchronous option? The AS4 work item did not create a wrapper for Deferred XCPD.
  + **No**, but we have designed it in such a way that it could be wrapped in the future.
  + Note that the AS4 option does not blindly wrap transactions; there are specific mappings that must be defined for each transaction to use over AS4.
  + There are the following items that would need to be compatible:
    - Use of additional SOAP header blocks
      * **AS4 allows for this;** it just needs to be wrapped as a message property**.** See XCPD AS4 wrapper(CorrelationTimeToLive).
* How will the Initiating Gateway tell the Responding Gateway where to send the response?
  + Option: Pass information in the request body.
    - Deferred XCPD passes response endpoint in the respondTo field in the body.
    - For XCA Query, the ebXML AdhocQueryRequest has no analog, but it is extensible with Slots.
    - For XCA Retrieve, there are no extensibility mechanisms – we would have to modify the schema.
  + **Chosen Option: Pass information in a SOAP header block**, with mustUnderstand=false (so it can be not understood safely). This allows the same design to be used for both query and retrieve, and avoids impacting current implementers by changing the schema.
* Should the Initiating Gateway specify a response endpoint directly, or a HCID that the Responding Gateway resolves via a directory? Both?
  + A HCID to resolve might help with some dynamic load balancing cases.
  + It might also be slightly more secure, as there is a layer of getting the endpoint through a trusted directory.
  + Finally, if the Initiating Gateway is grouped with an XUA X-Service User, its HomeCommunityID is included in the SOAP header already, so no additional HCID field would be needed.
  + **Endpoint only**. No compelling reason to do otherwise.
* Could/should this mechanism be designed in such a way that it supports an Initiating Gateway making a request and not knowing a priori whether it will be fulfilled synchronously or deferred?
  + **Yes, and we have done so.** We opted not to follow the pattern established by XCPD Deferred, which defines an explicitly different interaction, with different WSDL, for the initial request. Instead, we simply added new optional fields on the request and response for deferred behavior. This allowed us to handle the case where some results were immediately available, by returning them as normal. It also removed the possibility of mistakenly sending a deferred request to a non-deferred endpoint. It also allowed the sender to avoid having to know which responders support deferred a priori.
* Will we include in scope the possibility of multiple responses to a single request, each response coming asynchronously?
  + **Yes**. It was not included in the original scope, but once we designed a solution, it supported this easily.
* Should we limit which errors should be returned in the initial response vs. deferred responses?
  + Example: a malformed request could be caught right away
  + **No. Any error may be sent in either response.**
* Will this option be backward-compatible with existing implementers that do not support it? In other words, will implementers not need to be configured to know whether other systems support this option?
  + **Yes, it will be backward-compatible, so prior configuration is not required, but could be done.** As currently designed, the changes to existing transactions consist of:
    - New fields using extensibility mechanisms: SOAP header blocks and ebXML Slots
    - Making use of existing optional fields: AdhocQueryRequest/@id and AdhocQueryResponse/@requestId
  + In addition, the modified behavior does not break any assumptions.
* Is there any additional behavior that can happen between deferred requests and responses that this option needs to take into account? Consider link/unlink merge/unmerge of the corresponding patient identifier.
  + **No**. The potential for changes to the patient id already exists when synchronous mechanisms are used. An XCPD transaction and a subsequent XCA transaction that uses the obtained patient id are not a single atomic operation. Changes can happen between the two. So no additional complexities are added by using Deferred XCA. Existing error cases/codes (e.g. XDSUnknownPatientId) cover this.

Volume 1 – Profiles

Update Section 10.2 as shown

## 10.2 XDS.b Actor Options

Table 10.2-1b: XDS.b – Actors and Options

| Actor | Options | Vol. & Section |
| --- | --- | --- |
| Document Consumer | Basic Patient Privacy Enforcement | ITI TF-1: 10.2.9 |
| Basic Patient Privacy Proof | ITI TF-2a: 3.18.4.1.3.6 |
| Asynchronous Web Services Exchange | ITI TF-1: 10.2.5 |
| On-Demand Documents | ITI TF-1: 10.2.7 |
| Delayed Document Assembly | ITI TF-1: 10.2.10 |
| **Deferred Response** | **ITI TF-1: 10.2.X** |

Add the following new Section 10.2.X as shown

### 10.2.X Deferred Response Option

The Deferred Response Option is needed when the responding system requires extensive time for processing, as much as days or weeks. This need could arise through delegation of processing to external/remote systems or human interaction. The mechanism enables this by splitting the request and response into two separate two-way exchanges. This requires applications to manage the message correlation, create application level acknowledgements and determine where to send the Deferred Response message. For more information about Deferred Response and Asynchronous messaging in general see http://wiki.ihe.net/index.php?title=Asynchronous\_Messaging.

Given that there are multiple mechanisms available for longer latency query and retrieve, Deferred Response should be used in the following cases:

* When the delay in responding may be as much as days or weeks.
* When applications must support recovery of the long-running request and response through system restart.

If neither of these cases applies and longer latency query and retrieve is needed, consider either the WS-Addressing-Based Web Services Exchange Option or the AS4 Asynchronous Web Services Exchange Option.

Document Consumers which support the Deferred Response Option shall support Deferred Response on the Registry Stored Query [ITI-18] transaction as described in ITI TF-2a: 3.18.X, and on the Retrieve Document Set [ITI-43] transaction as described in ITI TF-2b: 3.43.X.

Update Section 18.2 as shown

## 18.2 XCA Integration Profile Options

Table 18.2-1: XCA Integration Profile - Actors and Options

| Actor | Options | Vol. & Section |
| --- | --- | --- |
| Initiating Gateway | XDS Affinity Domain Option | ITI TF-1: 18.2.1 |
| Asynchronous Web Services Exchange | ITI TF-1: 18.2.2 |
| On-Demand Documents | ITI TF-1: 18.2.4 |
| **Deferred Response** | **ITI TF-1: 18.2.X** |
| Responding Gateway | On-Demand Documents | ITI TF-1: 18.2.4 |
| Persistence of Retrieved Documents | ITI TF-1: 18.2.5 |
| **Deferred Response** | **ITI TF-1: 18.2.X** |

Add the following new Section 18.2.X as shown

### 18.2.X Deferred Response Option

The Deferred Response Option is needed when the XCA Responding Gateway requires extensive time for processing, as much as days or weeks. This need could arise through delegation of processing to external/remote systems or human interaction. The mechanism enables this by splitting the XCA request and response into two separate two-way exchanges. This requires applications to manage the message correlation, create application level acknowledgements and determine where to send the Deferred Response message. For more information about Deferred Response and Asynchronous messaging in general see http://wiki.ihe.net/index.php?title=Asynchronous\_Messaging.

Given that there are multiple mechanisms available for longer latency XCA, Deferred Response should be used in the following cases:

* When the delay in responding may be as much as days or weeks.
* When applications must support recovery of the long-running request and response through system restart.

If neither of these cases applies and longer latency XCA is needed, consider either the WS-Addressing-Based Web Services Exchange Option or the AS4 Asynchronous Web Services Exchange Option.

Initiating Gateways which support the Deferred Response Option shall support Deferred Response on the Cross Gateway Query [ITI-38] transaction as described in ITI TF-2b: 3.38.X, and Cross Gateway Retrieve [ITI-39] transaction as described in ITI TF-2b: 3.39.X. If the Initiating Gateway supports both the XDS Affinity Domain Option and the Deferred Response Option it shall support Deferred Response on the Registry Stored Query [ITI-18] and Retrieve Document Set [ITI-43] transactions.

Responding Gateways which support the Deferred Response Option shall support Deferred Response on the Cross Gateway Query [ITI-38] transaction as described in ITI TF-2b: 3.38.X, and Cross Gateway Retrieve [ITI-39] transaction as described in ITI TF-2b: 3.39.X.

Add the following new Section 18.3.3.X as shown

#### 18.3.3.X Use of Deferred Responses

Figure 18.3.3-X shows an example of Deferred responses in the case of a community that holds paper clinical documents and wishes to defer converting and registering those documents until they are queried for in electronic form.



Figure 18.3.3-X: Deferred response from paper-based community

* In this example, a **Document Consumer** triggers the process flow by sending a Deferred-Capable Registry Stored Query request to its Initiating Gateway. The request includes an endpoint for the Deferred response.
* The **Initiating Gateway** calls out to some number of Responding Gateways with a Cross Gateway Query, also Deferred-Capable. Each request includes an endpoint for the Deferred response, but in this case, the Deferred endpoint for the Initiating Gateway instead.
* The **Responding Gateway** evaluates the query request for errors, stores the request information, checks for any results/errors that may be available immediately, returns the initial response containing those initial results and an indication that more results will be returned later to the Initiating Gateway, and finally notifies staff of the request for additional results, using unspecified mechanisms.
* The **Initiating Gateway** passes the initial response on to the Document Consumer, aggregating responses from multiple Responding Gateways if necessary.
* At this point, the first exchange is complete and all system actors are free to do other work.
* **Staff** members pull the request information and perform whatever work is needed to convert the paper documents to electronic form and persist them: scanning, optical character recognition (OCR), parsing, manual evaluation, storing in an XDS registry/repository or other mechanism, etc.
  + Note that aside from this paper-based example, any time-intensive action may utilize this deferred mechanism. Another example would be human evaluation of the access decisions for this requester and these documents.
* When **Staff** has completed all work, it notifies the Responding Gateway, using unspecified mechanisms.
* The **Responding Gateway** correlates this response to the original request and sends the Cross Gateway Query Deferred Results message to the Initating Gateway endpoint previously sent.
* The **Initating Gateway** receives the response and forwards it as a Registry Stored Query Deferred Results message to the Document Consumer endpoint previously sent.
* The **Document Consumer** returns an acknowledgement to the Initating Gateway.
* The **Initiating Gateway** returns an acknowledgement to the Responding Gateway.

Volume 2a – Transactions

Update Section 3.18.4 as shown. Add the second interaction diagram.

### 3.18.4 Interaction Diagram

**The standard interaction is shown below.**



**If the Deferred Response Option is used, there are additional messages in the interaction.**



Add the following to the end of Section 3.18.4.1.2.5 as shown

###### 3.18.4.1.2.5 Compatibility of Options

…

If the Document Consumer supports the Deferred Response Option, it may trigger this pattern by sending a Deferred-Capable Registry Stored Query Request formatted as follows:

* A DeferredResponseEndpoint element shall be present in the SOAP Header, containing a URL with the Web Services Endpoint where a deferred response may be sent. The SOAP mustUnderstand value shall be “false” or “0”. The schema type is xsd:anyURI. An example follows:

<ihe:DeferredResponseEndpoint xmlns:S=http://www.w3.org/2003/05/soap-envelope

S:mustUnderstand="false">service entry point url</ihe:DeferredResponseEndpoint>

* The id attribute of the AdhocQueryRequest element shall be populated with a unique URI representing this initial request.

Both fields must be present for the request to be considered Deferred-Capable.

The Deferred-Capable Registry Stored Query Request is designed to be fully compatible with Document Registries and Initiating Gateways that do not support the Deferred Response Option, as they will ignore SOAP headers they do not understand.

Update Section 3.18.4.1.3 as shown.

##### 3.18.4.1.3 Expected Actions

The Document Registry 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: …
4. Process the query as appropriate:

* **For Document Registry Actors:** **…**
* **For Initiating Gateway Actors:**
* Initiating Gateway receives a Registry Stored Query by patient id: …
* Initiating Gateway receives a Registry Stored Query by entryUUID or uniqueID: …
* **Initiating Gateway that supports the Deferred Response Option receives a Deferred-Capable Registry Stored Query Request: For those Responding Gateways it chooses to contact, it may send Deferred-Capable Cross Gateway Query Requests as follows: a) It shall include its own Deferred Response endpoint, not the endpoint passed by the Document Consumer. b) It shall include its own AdhocQueryRequest id, not the id passed by the Document Consumer. This id shall be unique for each request to a given Responding Gateway. c) It shall retain the information needed to process future deferred results: the Deferred endpoint and original request ID from the Document Consumer, and the request ID for each Responding Gateway. d) It may choose to limit including the additional fields that identify a request as Deferred-Capable to only those Responding Gateways that it knows support the Deferred Response Option. However, Responding Gateways that do not support this option will ignore the Deferred aspects.**
* **Initiating Gateway that does not support the Deferred Response Option receives a Deferred-Capable Registry Stored Query Request: it shall ignore the extra SOAP header that is not understood and process the message like an ordinary Registry Stored Query. Note: This is existing expected behavior in SOAP.**

1. Return XML formatted metadata in an AdhocQueryResponse message…
2. When the Document Consumer receives the query response from the Initiating Gateway …

…

If the Document Consumer supports the Delayed Document Assembly Option it shall accept the following values of hash and size to indicate that the assembly of the document content has been delayed until the document is retrieved.

* size = 0 (zero)
* hash = da39a3ee5e6b4b0d3255bfef95601890afd80709 (SHA1 hash of a zero length file)

**If the Document Consumer supports the Deferred Response Option it shall accept the AdhocQueryResponse Indicating Deferred Processing, as described in section 3.18.4.1.3.7, indicating that some results of the request will be deferred until a later time.**

**If a Document Consumer that supports the Deferred Response Option receives an AdhocQueryResponse Indicating Deferred Processing, it shall retain the id of the original AdhocQueryRequest in order to correlate future results.**

**If a Document Consumer that does not support the Deferred Response Option receives an AdhocQueryResponse Indicating Deferred Processing, it shall ignore the extra slot(s) and process the message like an ordinary Registry Stored Query response. Note: This is existing expected behavior in ebXML.**

Add the following new Section 3.18.4.1.3.7 as shown

##### 3.18.4.1.3.7 AdhocQueryResponse Indicating Deferred Processing

For use with the Deferred Response Option: to indicate Deferred processing, the following slots are defined for the AdhocQueryResponse element:

* A “DeferredProcessingRequired” slot shall be included.
  + The Value, which is a 256-character maximum string, may be included to explain the nature of the delay. If present, the Value shall be appropriate for display to a user.
* A “DeferredProcessingEstimatedCompletion” slot may be included.
  + The Value is required, and shall be formatted as either a duration, using xsd:duration, or an absolute time, using HL7 V2.5 DTM format as defined in ITI TF-3, Table 4.2.3.1.7-2. The two may be distinguished by the first character, which is ‘P’ for duration and a digit for DTM.

An example of these slots is shown below:

<query:AdhocQueryResponse

xmlns:query="urn:oasis:names:tc:ebxml-regrep:xsd:query:3.0"

xmlns:rs="urn:oasis:names:tc:ebxml-regrep:xsd:rs:3.0"

xmlns:rim="urn:oasis:names:tc:ebxml-regrep:xsd:rim:3.0"

status="urn:ihe:iti:2007:ResponseStatusType:PartialSuccess">

<!-- Results not available now -->

<rs:ResponseSlotList>

<rim:Slot name="DeferredProcessingRequired">

<rim:ValueList>

<rim:Value>This query requires more time. The full results will be returned on the Deferred Response endpoint.</rim:Value>

</rim:ValueList>

</rim:Slot>

<rim:Slot name="DeferredProcessingEstimatedCompletion">

<rim:ValueList>

<rim:Value>PT20M</rim:Value>

</rim:ValueList>

</rim:Slot>

</rs:ResponseSlotList>

<!-- Errors available now -->

<rim:RegistryErrorList>

...

</rim:RegistryErrorList>

<!-- Registry objects available now -->

<rim:RegistryObjectList>

...

</rim:RegistryObjectList>

</query:AdhocQueryResponse>

Add the following new Section 3.18.4.2 as shown

#### 3.18.4.2 Registry Stored Query Deferred Results

This message is used only when a Document Consumer and an Initiating Gateway support the Deferred Response Option.

The message is sent from an Initiating Gateway to a Document Consumer to return Deferred results – registry objects and errors – from a prior Deferred-Capable query request.

##### 3.18.4.2.1 Trigger Events

This message is initiated when the Initiating Gateway determines it has results from a prior Deferred-Capable query request from a Document Consumer.

##### 3.18.4.2.2 Message Semantics

The Registry Stored Query Deferred Results message semantics are based on the Registry Stored Query response, but are packaged differently. For the contents of the AdhocQueryResponse element, see section 3.18.4.1.2 and ITI TF-3:4. For the web services details and sample messages, refer to section 3.18.4.2.2.1 rather than section 3.18.4.1.2.7.

In addition, the AdhocQueryResponse element shall contain the requestId attribute, containing the ID of the original AdhocQueryRequest this Results message is a response to.

The Registry Stored Query Deferred Results message may indicate in the AdhocQueryResponse element that additional Deferred processing is needed, as defined in section 3.18.4.1.3.7.

The Registry Stored Query Deferred Results Acknowledgement message semantics are based on the ebRS 3.0 RegistryResponse message.

###### 3.18.4.2.2.1 Web Services Transport

The Results and Results Acknowledgement messages will be transmitted using Synchronous or Asynchronous Web Services Exchange, according to the requirements specified in ITI TF-2x: Appendix V.

Table 3.18.4.2.2.1-1: 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 |

The specific values for the WSDL describing the Registry Stored Query Deferred Results Service are described in this section.

The Document Consumer shall accept a Registry Stored Query Deferred Results message formatted as a SIMPLE SOAP message and respond with a Registry Stored Query Deferred Results Acknowledgement message formatted as a SIMPLE SOAP message. The Initiating Gateway shall generate the Registry Stored Query Deferred Results message formatted as a SIMPLE SOAP message and accept a Registry Stored Query Deferred Results Acknowledgement message formatted as a SIMPLE SOAP message.

**IHE-WSP201) The attribute /wsdl:definitions/@name shall be “DocumentConsumer”.**

The following WSDL naming conventions shall apply:

wsdl:definitions/@name=**"DocumentConsumer**":

input message -> "Registry**StoredQueryDeferredResults**\_Message"

output message -> "Registry**StoredQueryDeferredResultsAcknowledgement**\_Message"

portType -> "**DocumentConsumer**\_PortType"

operation -> "Registry**StoredQueryDeferredResults**"

SOAP 1.2 binding -> **"DocumentConsumer**\_Binding\_Soap12"

SOAP 1.2 port -> **"DocumentConsumer**\_Port\_Soap12"

**IHE-WSP202) The targetNamespace of the WSDL shall be “urn:ihe:iti:xds-b:2007”**

Document Consumer: These are the requirements for the Registry Stored Query Deferred Results transaction presented in the order in which they would appear in the Document Consumer 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"
* namespace ="urn:oasis:names:tc:ebxml-regrep:xsd:rs:3.0", schemaLocation="rs.xsd"
* The /definitions/message/part/@element attribute of the Registry Stored Query Deferred Results message shall be defined as “query:AdhocQueryResponse”
* The /definitions/message/part/@element attribute of the Registry Stored Query Deferred Results Acknowledgement message shall be defined as “rs:RegistryResponse”
* Refer to Table 3.18.4.2.2.1-2 below for additional attribute requirements
* To support the Asynchronous Web Services Exchange Option on the Initiating Gateway, the Document Consumer shall support the use of a non-anonymous response EPR in the WS-Addressing replyTo header.

Table 3.18.4.2.2.1-2: Additional Attribute Requirements

|  |  |
| --- | --- |
| Attribute | Value |
| /definitions/portType/operation@name | DocumentConsumer\_ RegistryStoredQueryDeferredResults |
| /definitions/portType/operation/input/@wsaw:Action | urn:ihe:iti:2007:RegistryStoredQueryDeferredResults |
| /definitions/portType/operation/output/@wsaw:Action | urn:ihe:iti:2007:RegistryStoredQueryDeferredResultsAcknowledgement |
| /definitions/binding/operation/wsoap12:operation/@soapActionRequired | false |

The following WSDL fragment shows an example of Registry Stored Query Deferred Results transaction definition:

<?xml version="1.0" encoding="utf-8"?>

<definitions ...>

...

<types>

<xsd:schema elementFormDefault="qualified" targetNamespace="urn:ihe:iti:xds-b:2007">

<xsd:import

namespace="urn:oasis:names:tc:ebxml-regrep:xsd:query:3.0"

schemaLocation="schema\query.xsd"/>

<xsd:import

namespace="urn:oasis:names:tc:ebxml-regrep:xsd:rs:3.0"

schemaLocation="schema\rs.xsd"/>

...

</xsd:schema>

</types>

<message name="RegistryStoredQueryDeferredResults\_Message">

<documentation>Registry Stored Query Deferred Results</documentation>

<part name="body" element="query:AdhocQueryResponse"/>

</message>

<message name="RegistryStoredQueryDeferredResultsAcknowledgement\_Message">

<documentation>Registry Stored Query Deferred Results Acknowledgement</documentation>

<part name="body" element="rs:RegistryResponse"/>

</message>

...

<portType name="DocumentConsumer\_PortType">

<operation name="DocumentConsumer\_RegistryStoredQueryDeferredResults">

<input message="ihe:RegistryStoredQueryDeferredResults\_Message"

wsaw:Action="urn:ihe:iti:2007:RegistryStoredQueryDeferredResults"/>

<output message="ihe:RegistryStoredQueryDeferredResultsAcknowledgement\_Message"

wsaw:Action="urn:ihe:iti:2007:RegistryStoredQueryDeferredResultsAcknowledgement"/>

</operation>

...

</portType>

...

</definitions>

A full WSDL for the Document Consumer is found in ITI TF-2x: Appendix W.

3.18.4.2.2.1.1 Sample SOAP Messages

The samples in the following two sections show a SOAP request and its relative SOAP response when the Deferred results are a mix of successes and errors, and when further results are to be expected in a later message.

The sample messages also show the WS-Addressing headers <a:Action/>, <a:MessageID/>, <a:ReplyTo/>…; these WS-Addressing headers are populated according to ITI TF-2x: Appendix V: Web Services for IHE transactions. The specific errors and registry objects are omitted for brevity; in a real scenario these will be populated with the appropriate results.

Samples presented in this section are also available online on the IHE FTP site, see ITI TF-2x: Appendix W.

3.18.4.2.2.1.1.1 Sample Registry Stored Query Deferred Results SOAP Request

3.18.4.2.2.1.1.2.1 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:RegistryStoredQueryDeferredResults</a:Action>

<a:MessageID>urn:uuid:1795bb7a-8dc2-403a-9914-fbeab9e2a77e</wsa:MessageID>

</s:Header>

<s:Body>

<query:AdhocQueryResponse

xmlns:query="urn:oasis:names:tc:ebxml-regrep:xsd:query:3.0"

xmlns:rs="urn:oasis:names:tc:ebxml-regrep:xsd:rs:3.0"

xmlns:rim="urn:oasis:names:tc:ebxml-regrep:xsd:rim:3.0"

requestId="urn:uuid:df9b89ed-395e-40a7-8510-0b4a390434c4"

status="urn:ihe:iti:2007:ResponseStatusType:PartialSuccess">

<!-- Results not available now -->

<rs:ResponseSlotList>

<rim:Slot name="DeferredProcessingRequired">

<rim:ValueList>

<rim:Value>This query requires more time. The full response will be returned on the Deferred Response endpoint.</rim:Value>

</rim:ValueList>

</rim:Slot>

<rim:Slot name="DeferredProcessingEstimatedCompletion">

<rim:ValueList>

<rim:Value>PT20M</rim:Value>

</rim:ValueList>

</rim:Slot>

</rs:ResponseSlotList>

<!-- Errors available now -->

<rim:RegistryErrorList>

...

</rim:RegistryErrorList>

<!-- Registry objects available now -->

<rim:RegistryObjectList>

...

</rim:RegistryObjectList>

</query:AdhocQueryResponse>

</s:Body>

</s:Envelope>

3.18.4.2.2.1.1.2.2 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:RegistryStoredQueryDeferredResults</a:Action>

<a:MessageID>urn:uuid:1795bb7a-8dc2-403a-9914-fbeab9e2a77e</wsa:MessageID>

<a:ReplyTo>

<a:Address>http://192.168.2.4:9080/XDS/InitiatingGatewayReceiver.svc</a:Address>

</a:ReplyTo>

</s:Header>

<s:Body>

<query:AdhocQueryResponse

xmlns:query="urn:oasis:names:tc:ebxml-regrep:xsd:query:3.0"

xmlns:rs="urn:oasis:names:tc:ebxml-regrep:xsd:rs:3.0"

xmlns:rim="urn:oasis:names:tc:ebxml-regrep:xsd:rim:3.0"

requestId="urn:uuid:df9b89ed-395e-40a7-8510-0b4a390434c4"

status="urn:ihe:iti:2007:ResponseStatusType:PartialSuccess">

<!-- Rest of AdhocQueryResponse message goes here -->

</query:AdhocQueryResponse>

</s:Body>

</s:Envelope>

3.18.4.2.2.1.1.2 Sample Registry Stored Query Deferred Results Acknowledgement SOAP Response

3.18.4.2.2.1.1.2.1 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:RegistryStoredQueryDeferredResultsAcknowledgement</a:Action>

<a:MessageID>urn:uuid:D6C21225-8E7B-454E-9750-821622C099DB</wsa:MessageID>

<a:RelatesTo>urn:uuid:1795bb7a-8dc2-403a-9914-fbeab9e2a77e</wsa:MessageID>

</s:Header>

<s:Body>

<rs:RegistryResponse

xmlns:rs="urn:oasis:names:tc:ebxml-regrep:xsd:rs:3.0"

status="urn:oasis:names:tc:ebxml-regrep:ResponseStatusType:Success"

</rs:RegistryResponse>

</s:Body>

</s:Envelope>

3.18.4.2.2.1.1.2.2 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:RegistryStoredQueryDeferredResultsAcknowledgement</a:Action>

<a:MessageID>urn:uuid:D6C21225-8E7B-454E-9750-821622C099DB</wsa:MessageID>

<a:RelatesTo>urn:uuid:1795bb7a-8dc2-403a-9914-fbeab9e2a77e</wsa:MessageID>

<a:To s:mustUnderstand="1">http://192.168.2.4:9080/XDS/InitiatingGatewayReceiver.svc</a:To>

</s:Header>

<s:Body>

<rs:RegistryResponse

xmlns:rs="urn:oasis:names:tc:ebxml-regrep:xsd:rs:3.0"

status="urn:oasis:names:tc:ebxml-regrep:ResponseStatusType:Success"

</rs:RegistryResponse>

</s:Body>

</s:Envelope>

##### 3.18.4.2.3 Expected Actions

This message contains results from a prior Registry Stored Query. As such, the Document Consumer shall fulfill the same requirements as it does for the Registry Stored Query response, detailed in section 3.18.4.1.3.

In addition:

1. The Document Consumer shall attempt to correlate this message to the appropriate original request, by matching the requestId attribute of the AdhocQueryResponse element in the results to the id attribute of the AdhocQueryRequest element in the original request.
2. The Document Consumer may verify the format and validity of the results message.
3. If the requestId attribute is not present, no matching request can be found, or the results message is not valid, the Document Consumer shall return a Registry Stored Query Deferred Results Acknowledgement message with RegistryResponse status = “urn:oasis:names:tc:ebxml-regrep:ResponseStatusType:Failure”, and containing a RegistryError element with errorCode “XDSRegistryError” and codeContext explaining the nature of the error.
4. If the matching request can be found and the results message is valid,
   1. If this is the final set of results for this request, i.e. the AdhocQueryResponse does not indicate further Deferred Processing as described in section 3.18.4.1.3.7, the Document Consumer shall delete or otherwise mark the request id as completed, such that no subsequent results messages will match the request.
   2. The Document Consumer shall return a Registry Stored Query Deferred Results Acknowledgement message with RegistryResponse status = “urn:oasis:names:tc:ebxml-regrep:ResponseStatusType:Success”.

Volume 2b – Transactions

Update Section 3.38.4 as shown. Add the second interaction diagram.

### 3.38.4 Interaction Diagram

**The standard interaction is shown below.**



**If the Deferred Response Option is used, there are additional messages in the interaction.**



Add the following to the end of Section 3.38.4.1.2 as shown.

##### 3.38.4.1.2 Message Semantics

…

Initiating Gateways which support the Deferred Response Option shall be capable of sending a Deferred-Capable Cross Gateway Query Request (see ITI TF-2a: 3.18.4.1.2.5) either through internal mechanisms or, when the XDS Affinity Domain Option is also declared, through interaction with an XDS Document Consumer which supports the Deferred Response Option.

Add the following new Section 3.38.4.1.3.3 as shown

**3.38.4.1.3.3 Deferred Response Option**

If a Responding Gateway that supports the Deferred Response Option receives a Deferred-Capable Cross Gateway query request (see ITI TF-2a: 3.18.4.1.2.5):

* The Responding Gateway has the option to defer returning some or all of the results of the query request – registry objects and errors – to one or more subsequent Deferred Results messages, in order to allow for additional processing time. See section 3.38.4.2 for details.
  + How the Responding Gateway determines which results to return immediately and which to defer is not specified.
  + Whether the Responding Gateway returns individual results as soon as each one is available, returns groups of results, or waits until all results are available and returns them in a single Deferred Results message is not specified.
  + How and where processing delays are introduced behind the Responding Gateway is not specified. They may occur in non-XDS mechanisms, or in the unspecified interactions between the Responding Gateway and a grouped Document Consumer. Note that the Deferred Response Option is not supported on Document Registries, so if the Responding Gateway is grouped with a Document Consumer, the ITI-18 transactions it triggers will not be Deferred.
* The Responding Gateway shall return any results immediately available in the initial Cross Gateway Query Response. If none are available, the Responding Gateway shall return an empty response (i.e. no registry objects or errors).
  + The response status shall reflect only those results that are being returned in this message. For an empty response, the response status shall be urn:oasis:names:tc:ebxml-regrep:ResponseStatusType:Success.
  + If any remaining results require additional processing time, the message shall not include those results and shall indicate Deferred processing in the AdhocQueryResponse as described in ITI TF-2a: 3.18.4.1.3.7.
  + If the DeferredProcessingEstimatedCompletion is provided, it shall reflect an estimate of all deferred processing remaining for the request.
* When Deferred processing is complete for a set of results and the Responding Gateway wishes to return them, it shall send a Cross Gateway Query Deferred Results message, as described in section 3.38.4.2.
  + The message shall include all results available at this time, including any already returned.
  + The response status shall reflect only those results that are being returned in this message.
  + If any remaining results require additional processing time, the message shall not include those results and shall indicate Deferred processing in the AdhocQueryResponse as described in ITI TF-2a: 3.18.4.1.3.7.
  + If the DeferredProcessingEstimatedCompletion is provided, it shall reflect an estimate of all deferred processing remaining for the request.
* The Responding Gateway must attempt to return all results of the request, through the combination of initial and Deferred responses.
  + The Responding Gateway should attempt retries if an Initiating Gateway is unavailable. The details and number of attempts are not specified.
  + The Responding Gateway must not send the final Deferred Results message for a request, defined as a Deferred Results message that does not indicate additional Deferred processing, until the initial Cross Gateway Query Response and any other Deferred Results messages have successfully been received/acknowledged.

If a Responding Gateway that supports the Deferred Response Option receives a request that includes a DeferredResponseEndpoint element (see ITI TF-2a: 3.18.4.1.2.5) but not the id attribute of the AdhocQueryRequest element, it shall return the error code XDSRegistryError, and details of the error in the attribute codeContext.

If a Responding Gateway that does not support the Deferred Response Option receives a request that includes a DeferredResponseEndpoint element (see ITI TF-2a: 3.18.4.1.2.5), it shall ignore the extra SOAP header and process the message like an ordinary Cross Gateway Query. Note: This is existing expected behavior in SOAP.

If an Initiating Gateway supports the Deferred Response Option, it shall fulfill the requirements on a Document Consumer that supports the Deferred Response Option, as detailed in ITI TF-2a: 3.18.4.1.3 Expected Actions. In addition:

* If the XDS Affinity Domain Option is supported, when consolidating results from multiple Responding Gateways, the Initiating Gateway shall consolidate the reporting of any Deferred processing for this request as follows:
  + If any responses indicate Deferred processing, the Initiating Gateway shall indicate Deferred processing in the Registry Stored Query response.
  + If a consolidated estimate of all processing time remaining for all Responding Gateways for this request can be determined, the Initiating Gateway may include it in the Registry Stored Query response.
* In the response to the initiator of the transaction – either the Document Consumer or the internal actor – in addition to the query results that were returned by Responding Gateways in Cross Gateway Query responses, the Initiating Gateway may also include any Deferred results that were received before it had returned this response.

If an Initiating Gateway that does not support the Deferred Response Option receives a request that includes a DeferredResponseEndpoint element (see ITI TF-2a: 3.18.4.1.2.5), it shall ignore the extra SOAP header and process the message like an ordinary Cross Gateway Query. Note: This is existing expected behavior in SOAP.

If an Initiating Gateway that does not support the Deferred Response Option receives a response that indicates Deferred processing (see ITI TF-2a: 3.18.4.1.3.7), it shall ignore the extra slot(s) and process the message like an ordinary Cross Gateway Query response. Note: This is existing expected behavior in ebXML.

Add the following new Section 3.38.4.2 as shown

#### 3.38.4.2 Cross Gateway Query Deferred Results

The message is sent from a Responding Gateway to an Initiating Gateway to return Deferred results – registry objects and errors – from a prior Deferred-Capable query request.

##### 3.38.4.2.1 Trigger Events

This message is initiated when Deferred processing is complete for a set of results and the Responding Gateway wishes to return them to the Initiating Gateway.

##### 3.38.4.2.2 Message Semantics

The message semantics are based on the Registry Stored Query Deferred Results. See ITI TF-2a: 3.18.4.2.2.

TBD refer to web services details and sample messages when done

##### 3.38.4.2.3 Expected Actions

The Initiating Gateway shall support the Expected Actions described in ITI TF-2a: 3.18.4.2.3 for the Document Consumer.

In addition, the Initiating Gateway shall support the Expected Actions described in ITI TF-2b: 3.38.4.1.3 for the Initiating Gateway actor.

In addition, if the Initiating Gateway supports the XDS Affinity Domain Option:

* It shall determine if this results message correlates to a prior request from a Document Consumer, by comparing the AdhocQueryResponse requestId to the set of request IDs retained from prior Document Consumer triggered requests. If it does, the Initiating Gateway shall initiate a Registry Stored Query Deferred Results transaction (See ITI TF-2a: 3.18.4.2) to the appropriate Document Consumer, as follows:
  + Direct the message to the Deferred response endpoint of the Document Consumer that was previously retained.
  + Include the results (registry objects and errors) that were passed in the Cross Gateway Query Deferred Results message.
  + Set the requestId of the AdhocQueryResponse to the id of the original ITI-18 AdhocQueryRequest from the Document Consumer that was previously retained.
  + Consolidate the reporting of any remaining Deferred processing for this request:
    - If the Cross Gateway Query Deferred Results message indicates additional Deferred processing, or if any of the requests to other Responding Gateways are not complete (i.e. the most recent response or Deferred Results message indicates additional Deferred processing), then the Initiating Gateway shall indicate additional Deferred processing.
    - If a consolidated estimate of all processing time remaining for all Responding Gateways for this request can be determined, the Initiating Gateway may include it.

TBD ITI-39, ITI-43