**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 11

Volume 1 – Profiles 14

10.2 XDS.b Actor Options 14

10.2.X Deferred Response Option 14

18.2 XCA Integration Profile Options 15

18.2.X Deferred Response Option 15

Volume 2a – Transactions 16

3.18.X TBD 16

Volume 2b – Transactions 17

3.38.X TBD 17

3.39.X TBD 17

3.43.X TBD 17

# 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. The Deferred Response option reflects the more detailed understanding and feedback from implementers regarding processing that may result in significant delay. The existing WS-Addressing and AS4 Asynchronous Web Services Exchange Options can support some scenarios with delayed response but not environments where the delay in responding may be as much as days or weeks. These cases require a mechanism that is managed by the application and which supports recovery through system restart. Deferred Response mode provides applications with such capability. In doing so it also adds responsibilities to the application, in particular for managing message correlation, creating application level acknowledgements and determining where to send a Deferred Response message. The new flexibility allowed by the Deferred Response option is deemed worthy of these additional requirements on the application. For more information about Deferred Response and Asynchronous messaging in general see <http://wiki.ihe.net/index.php?title=Asynchronous_Messaging>.

## Open Issues and Questions

* What format should the acknowledgement messages be? Prior IHE work on this suggested ebXML RegistryResponse for both Query and Retrieve.
  + If ebXML RegistryResponse is used, what status values should be used? ebXML defines the value Unavailable for this case, while IHE tentatively defined the values RequestAccepted and ResponseAccepted.
* Will we need to explicitly indicate deferred processing in the body of the request, or will the Action be sufficient?
  + Deferred XCPD includes this explicitly in the responsePriorityCode field.
  + For the XCA Query, the ebXML AdhocQueryRequest has no analog, but it is extensible with Slots.
  + The XCA Retrieve message is defined by IHE, so it could be extended.
  + **(Tentative choice)** This is made explicit by the presence of the response endpoint.
* How will the Responding Gateway know what endpoint to send the response to?
  + Deferred XCPD includes this explicitly in the respondTo field.
  + **(Tentative choice)** For the XCA Query, the ebXML AdhocQueryRequest has no analog, but it is extensible with Slots.
  + The XCA Retrieve message is defined by IHE, so it could be extended.
  + As an alternate method, we could define a SOAP header block for the response endpoint similar to WS-Addressing ReplyTo. NOTE: This would be a problem if combining deferred and AS4 mechanisms.
* 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.
  + **(Tentative choice)** Endpoint only
* What error cases will exist, and how will we handle them?
  + Sending a deferred request to a Responding Gateway that doesn’t support it
  + Existing error cases will be unchanged
* Alternate solutions that are feasible and are being considered:
  + 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 direct notification pushes or stored notification pulls could be used.
    - 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
      * 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: Broker cancels subscription when complete, returns notification – there is a CP adding this capability. Need to look into.
        + Option: Add a workflow that contains the explicit notification (see next option).
      * Cross-community versions of these transactions would have to be defined.
      * All participants would have to know responder requires DSUB.
  + 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.
      * Technical Committee feels this option may be too complex.
* For the stated use cases, may a Responding Gateway always implicitly force use of Deferred by returning an error code when a synchronous transaction is used?
  + Legacy paper: **Tentatively yes, but this should be discouraged**. There are two cases where the synchronous transaction should succeed:
    - If the requested documents have already been registered by a previous Deferred operation, and no new paper documents for this query exist. Consider making this a hard requirement.
    - If the requested documents can be registered before a synchronous timeout occurs.
  + Timeouts on Retrieve: **No**. In this case, the Responding Gateway should always attempt the synchronous operation first.
* For the stated use cases, how will an Initiating Gateway know whether to use the Deferred Response option vs. a synchronous call for a given Responding Gateway?
  + Option: Explicit error code suggesting deferred
  + Legacy paper: In this case, the Initiating Gateway is configured to know which Responding Gateways require use of the Deferred option, because they have this mode of operation.
    - Even in this case, is it possible to sometimes use synchronous calls? If it can be known that a source of clinical information is purely legacy, with no new paper documents being added for this patient, then once a given deferred query has completed, any subsequent queries may be made synchronously. However, this is an edge case, and would require significant logic in the Initating Gateway as to be unlikely to implement.
  + Timeouts on Retrieve: In this case, the Initiating Gateway could trigger use of the Deferred option, if available, in response to getting a network timeout from a synchronous transaction.
* Will this option break existing implementers that do not support it?
  + It should not. As currently designed,
    - Initiating Gateways should only send deferred requests to Responding Gateways that support this option.
    - If a Responding Gateway that doesn't support this option receives the deferred WS-Addressing Action, it should reject the message, as it doesn't understand it.
    - If a Responding Gateway that doesn't support this option receives the deferred response endpoint in a Query request as an extra Slot, it should ignore it, as this is the defined ebXML behavior.
    - If a Responding Gateway that doesn't support this option receives the deferred response endpoint in a Retrieve request as an additional element, what should happen? **This point needs research**.
      * The additional element would violate the existing schema. We will need to update the XDS.b\_DocumentRepository.xsd schema, as was done for the On-Demand Supplement, but will existing gateways that do not intend to support the option need to adopt the new schema? How did this work with On-Demand?
      * If existing gateways that do not intend to support the option do not adopt the new schema, then behavior is undefined. They could ignore it, they could reject the message with a SOAP fault, or something else. Since this is an error case anyway, this is probably an acceptable outcome.
  + Note that if we were to pass the response endpoint as a SOAP header block instead, there would be no danger of breaking existing implementers.
* 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?
  + **Tentatively yes, but needs more research**.
  + The same WS-Addressing action as synchronous would be used.
  + To allow for the possibility of a deferred response, the requester would need to provide an endpoint to receive that response. Presence of this endpoint would trigger the potential for deferred.
  + For the synchronous response, the requester would either receive a normal XCA response or a deferred acknowledgement, indicating the deferred mechanism was used.
    - Would there be a problem with this indeterminate response?
  + The Initiating Gateway should still only use this capability with Responding Gateways that support it (so it should still be a named option). See above question on breaking existing implementers.
  + Note that this is not how the XCPD Deferred mechanism works. That mechanism is explicit, as is the tentative design of XCA Deferred.
* Will a Deferred Response option also need to be added to the XDS.b profile, due to potential groupings? If not, it seems like some grouped transactions will not work.
  + Analysis: Responding Gateway, grouped with a Document Consumer, using Deferred.
    - 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 tentatively decided no**. Alternative mechanisms such as WS-Addressing style or AS4 style Asynchronous options, polling, or other unspecified mechanisms may be used.
  + Analysis: Initiating Gateway, grouped with a Document Consumer, using Deferred.
    - 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?
    - **Tentatively no**, using the same reasoning as the previous case.
  + Analysis: Initiating Gateway, which supports the XDS Affinity Domain Option, using Deferred.
    - 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).
    - **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.
    - **Tentative choice** Option: Add Deferred Response options for ITI-18 and ITI-43.
    - 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.
* Can the Deferred Response option be used with the AS4-style Asynchronous option?
  + **Needs research. Was this done for Deferred XCPD**?
* Should this supplement include its own use cases? The XCPD Deferred Response Option does not.

## Closed Issues

* Alternate solutions that have already been considered and rejected. Driving reasons are in bold.
  + 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: AS4-style Asynchronous messaging
    - In this alternative, the interested party would use the AS4 asynchronous mechanism to send an XCA Cross Gateway Query.
    - Pros
      * Existing mechanism; no new specifications development.
      * Some degree of longer latency supported.
    - Cons
      * **It cannot handle long latencies (e.g. days). See Closed Issue AS4-5.**
      * 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.
  + 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
      * 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 feels this option may be too complex.
  + 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.
* Could this method be used for the Responding Gateway to find the response endpoint? If the Initiating Gateway is grouped with an XUA X-Service User, its HomeCommunityID is included in the SOAP header, and the Responding Gateway could use that to look up a deferred response endpoint. This technique was used by the US eHealth Exchange for deferred XCPD, as use of the respondTo element was not yet defined by IHE.
  + Tech Committee did not think this was appropriate.
* Will we include in scope the possibility of multiple responses to a single request, each response coming asynchronously?
  + Currently this is considered out of scope as a driving requirement. However, note that some alternate solutions such as DSUB provide this.
* What potential behavior 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.
  + 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

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.

The Deferred Response Option reflects the more detailed understanding and feedback from implementers regarding processing that may result in significant delay. The existing Asynchronous Web Services Exchange Options can support some scenarios with delayed response but not environments where the delay in responding may be as much as days or weeks. These cases require a mechanism that is managed by the application and which supports recovery through system restart. Deferred Response mode provides applications with such capability. In doing so it also adds responsibilities to the application, in particular for managing message correlation, creating application level acknowledgements and determining where to send a Deferred Response message. The new flexibility allowed by the Deferred Response Option is deemed worthy of these additional requirements on the application. For more information about Deferred Response and Asynchronous messaging in general see <http://wiki.ihe.net/index.php?title=Asynchronous_Messaging>.

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

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.

The Deferred Response Option reflects the more detailed understanding and feedback from implementers regarding processing that may result in significant delay. The existing Asynchronous Web Services Exchange Options can support some scenarios with delayed response but not environments where the delay in responding may be as much as days or weeks. These cases require a mechanism that is managed by the application and which supports recovery through system restart. Deferred Response mode provides applications with such capability. In doing so it also adds responsibilities to the application, in particular for managing message correlation, creating application level acknowledgements and determining where to send a Deferred Response message. The new flexibility allowed by the Deferred Response Option is deemed worthy of these additional requirements on the application. For more information about Deferred Response and Asynchronous messaging in general see http://wiki.ihe.net/index.php?title=Asynchronous\_Messaging.

Volume 2a – Transactions

Update Section 3.18.X as shown

## 3.18.X TBD

TBD

Volume 2b – Transactions

Update Section 3.38.X as shown

## 3.38.X TBD

TBD

Update Section 3.39.X as shown

## 3.39.X TBD

TBD

Update Section 3.43.X as shown

## 3.43.X TBD

TBD