Integrating the Healthcare Enterprise

IHE IT Infrastructure Technical Framework Supplement

Mobile access to Health Documents (MHD)

Draft for Public Comment

Date:	June 05, 2012
Author:	IHE ITI Technical Committee
Email:	iti@ihe.net

Foreword

This is a supplement to the IHE ITI Technical Framework V8.0. Each supplement undergoes a process of public comment and trial implementation before being incorporated into the volumes of the Technical Frameworks.

This supplement is published on June 05, 2012 for Public Comment. Comments are invited and may be submitted at http://www.ihe.net/iti/iticomments.cfm. In order to be considered in development of the Trial Implementation version of the supplement comments must be received by July 05, 2012.

This supplement describes changes to the existing technical framework documents and where indicated amends text by addition (**bold underline**) or removal (~~**bold strikethrough**~~), as well as addition of new sections introduced by editor's instructions to "add new text" or similar, which for readability are not bolded or underlined.

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

Replace Section X.X by the following:

CONTENTS

Introduction to this Supplement

The Mobile access to Health Documents (MHD) profile defines a simplified RESTful interface to an XDS like environment. It defines transactions to a) submit a new document from the mobile device to a document receiver, b) get the metadata for an identified document, c) find document entries containing metadata based on query parameters, and d) retrieve a copy of a specific document.

These transactions leverage the document content and format agnostic metadata concepts from XDS, but simplify them for access by mobile devices. The MHD profile does not replace XDS. It can be used to allow mobile devices constrained access to an XDS health information exchange. The following figure shows one possible way to implement MHD with a document sharing environment (that may, but is not necessarily, XDS based). This implementation choice is not mandatory and we recognize other architectures will be implemented.

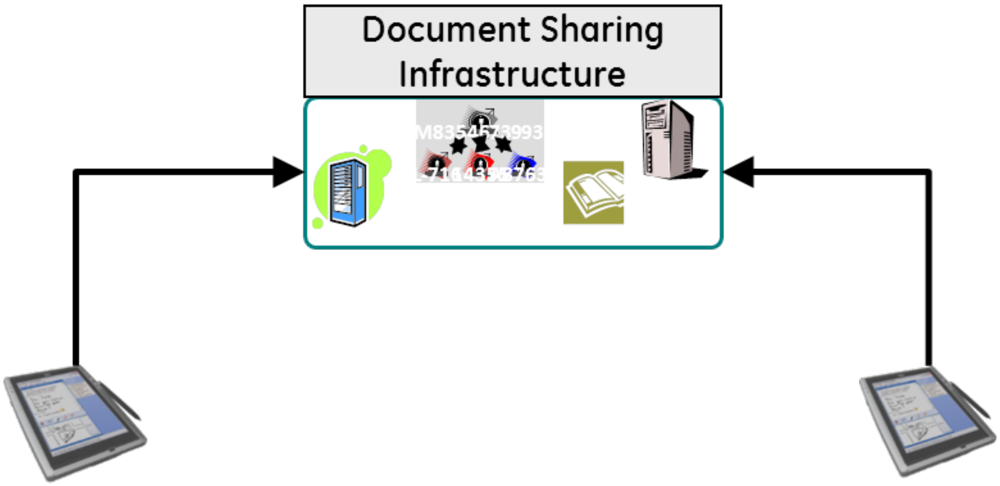


Figure 1: Mobile access to a Document Sharing environment.

The XDS profile is specifically designed to support the needs of Cross-Enterprise security, privacy, interoperability, and includes characteristics to support this level of policy and operational needs. The MHD profile has simplified the interactions in ways that are more consistent with a single policy domain use. The MHD transactions are not specifically tied to XDS, and some of the system implementations envisioned would interface directly to an organizational EHR, or a multi-national PHR.

The following lists a few examples of the environments which might choose to use the MHD profile instead of the XDS profile. The MHD profile supports a broad set of the XDS use cases and functionality while keeping the technology as simple as possible.

- Medical devices such as those targeted by the Patient Care Devices (PCD) domain or Continua organization, submitting data in the form of documents.

- Kiosks used by patients in hospital registration departments, where it is anticipated that a hospital staff member will review, edit, and approve the document before it is allowed into the hospital system.
- PHR publishing into a staging area for subsequent import into an EHR or HIE.
- Patient or provider application that is configured to securely connect to a PHR in order to submit a medical history document.
- Electronic measurement device participating in an XDW workflow and pulling medical history documents from an HIE.
- A General Practitioner physician's office with minimal IT capabilities using a mobile application to connect to an HIE or EHR.

Open Issues and Questions

- MHD_010: For this profile we took the view that XDS is really about the lifecycle of Document Entries (not documents), so this profile defines a RESTful object that represents the XDS Document Entry, and is accessible at http://<location>/<patientID>/DocumentEntry/<entryUUID>/. This approach was used so that there would be an object definition that we can do the RESTful operators on. Thus a GET on this will return the document entry metadata, a PUT would create a new entry (possibly in future developments supporting replace). Etc. The drawback of this is that we really want in our scope that a full patientID be first fully specified, not just the local part. This means that it should include the assigning authority as well as the patient id value. The second drawback is that the size or list of potential patientIDs is endless, so a service would need to have virtual entry points that dynamically handle the patientIDs. Although there are ways to be aware of all current patient ID values in an XDS Affinity Domain (e.g., be a PIX consumer, just like the XDS Registry does), it still is an open-set.
- MHD_011: Should the Document Source response to a GET on http://<location>/<patientID>/DocumentEntry/ be specified? One possibility could be to get a list of ALL applicable and available DocumentEntries for that patient, subject to local configuration, system capability, and policy? The same question for http://<location>/<patientID>/Document/. These are not specified in this version of the Profile. We request comments include specific suggestions. Possible responses include:

  Could return an HTTP directory page as is common for web servers.

  Could return a JSON array of available DocumentEntries

  Could be a way to bring in hData/ATOM
- MHD_012: This supplement forbids the use of HTTP "Conditional GET" because there is concern that the real meaning of metadata values (for example the Document Entry metadata value for creationTime) is slightly different than what HTTP "Conditional GET" would require for consistent processing. The implementation of this may be very tricky for a Proxy implementation to achieve. Please suggest specific rules for handling this or other HTTP headers in light of specific XDS characteristics.

- MHD_013: This supplement supports creation of only one document at a time through the DocumentEntry and, for simplicity sake, does not require the submitter to specify attributes of a Submission Set. This means that a Source can only publish one document at a time, and the service side will need to create the submission set based on the document entry and local configuration. This submission set restriction to one document may be a concern. There is concern that available information may not be sufficient to permit this creation, specifically for the submission set contentTypeCode, intendedRecipient, and SourceID.
- MHD_014: This supplement only supports access to the DocumentEntry. There is no access to the Submission Set, Folders, or Associations. There does not appear to be a use case need, but we invite comments if there are needs for access to each of these.
- MHD_015: This supplement does not include specific hData use. Please provide public comment indicating the interest in including hData, and in what way the profile would change to leverage hData. I expect that re-introducing hData will be a natural and appropriate addition for Trial Implementation. There is a concern that the use of hData will introduce ATOM feeds, which may not be necessary, and may require both XML and JSON encoding.
- MHD_016: The Security Risk Assessment, as defined in the IHE Risk Assessment Cookbook has not been executed on this profile. We encourage specific risks be described in public comment to inform this Risk Assessment.
- MHD_017: In the mobile environment a success/failure is typically all that is desired. Mapping a rich error code that is available in a SOAP environment (XDS error codes) to a HTTP 5xx/4xx environment is not obvious. XDS supports a rich set of error codes, how is this detail returned back to the client side in MHD? Should we try to express the rich list through the http numeric, text, or not attempt to provide the richness of the error codes supported by XDS. Please explain the mobile device specific cases that would benefit from more detailed error codes.
- MHD_018: Why do we not reuse ITI-12 Retrieve Document for Display? This transaction seems to be almost identical to the proposed GET transaction. The only difference is the location of the PatientID, and the use of documentUID rather than EntryUUID. This would introduce asymmetry to the API, but would re-use transactions that we have available. Should we at-least indicate a grouping behavior?
- MHD_019: The JSON encoding uses anonymous objects. This is means that implementations will need to take into account the expected object type of a response from that URL. The alternative would be to add an object name as the outermost level. A named object would open the potential for a URL to respond with more than one kind of object. Will explicitly named objects be needed in the future?

Closed Issues

- MHD_001: Standards selection is to define simple HTTP RESTful transactions using the Document Entry metadata as the object (resource) acted upon. Metadata is described in JSON format.
- MHD_002: Security model is undefined as there are plenty of HTTP based security models that layer in between the low level transport (TCP) and the HTTP encoding. These security

models can be layered in without modifying the characteristics of this profile. The use of TLS will be encouraged, specifically the use of ATNA, but will not be mandated. Security Standards known to the editing team are:

1. Kerberos (as used in IHE RID + EUA)
2. SAML (modified XUA for non SOAP – Likely SAML SSO Profile)
3. OpenID
4. OAuth 2.0 or later
5. Hybrid Auth -- http://hybridauth.sourceforge.net/index.html
6. TLS Client Authentication

- MHD_003: The Document List transaction is constrained from the XDS Queries to just FindDocuments.
- MHD 005_ We know that the length of the URL encoding could become long. We believe that is still within URL length limits and can be supported. The HTTP RFCs do not specify a limit, and current servers are prepared for very long URLs.
- MHD_006 We chose JSON encoding because it is native to many mobile platforms. The choice also makes a clear distinction from the current XML encoding, thus limiting our XML encodings to only the SOAP transports. The expectation is that we will change XML encoded values into JSON encoded values as well, so that the mobile device doesn't need to have both JSON and XML processing. HL7 v2 values will be converted as needed.
- MHD_008 There has been a request for further simplification of encoding of codes and identifiers omit the assigning authority. For example patientID might not need to encode the assigning authority given that the mobile client and server are in the same domain. This seems risky and adds little value. We require full specification keeping with XDS criteria. Is this a good decision? MHD does not forbid local use of this practice, but it simply means that the system is not compliant with this profile.

Volume 1 – Profiles

Add to Section X

X Mobile access to Health Documents (MHD) Profile

Mobile device applications is an emerging platform for healthcare software. These devices are resource and platform constrained, which drives the implementer to use simpler network interface technology. There are a number of sources of documents, for example hosted by a Health Information Exchange (HIE), large health provider electronic health record (EHR), or personal health record (PHR).

The Mobile access to Health Documents (MHD) profile strives to define one standardized interface to health documents for use by mobile devices so that deployment of mobile applications is more consistent and reusable. In this context, mobile devices include tablets and smart-phones, and also include embedded devices like home-health devices. This profile is also applicable in larger systems where the needs are simple, such as to pull the latest summary for display on a secondary monitor. The critical aspects of the 'mobile device' are that it is resource constrained, has a simple programming environment (e.g., JSON, javascript), simple network stack (e.g., HTTP), and simple display functionality (e.g., HTML browser). The goal is to limit the additional libraries that would be necessary to process SOAP, WSSE, MIME-Multipart, MTOM/XOP, ebRIM, and multi-depth XML.

The Mobile access to Health Documents (MHD) profile defines actors and transactions. There is one set of actors and a transaction used to submit a single new document from the mobile device to a receiving system. The other set of actors and transactions are used to get a list of document entries containing metadata, and to retrieve a copy of a specific document.

These transactions leverage the metadata concepts from XDS, but simplify the technology requirements for access by mobile devices. The MHD profile does not replace XDS. It enables simplified access by mobile devices to an XDS (or a similar) document management environment containing health information.

X.1 MHD Actors, Transactions, and Content Modules

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

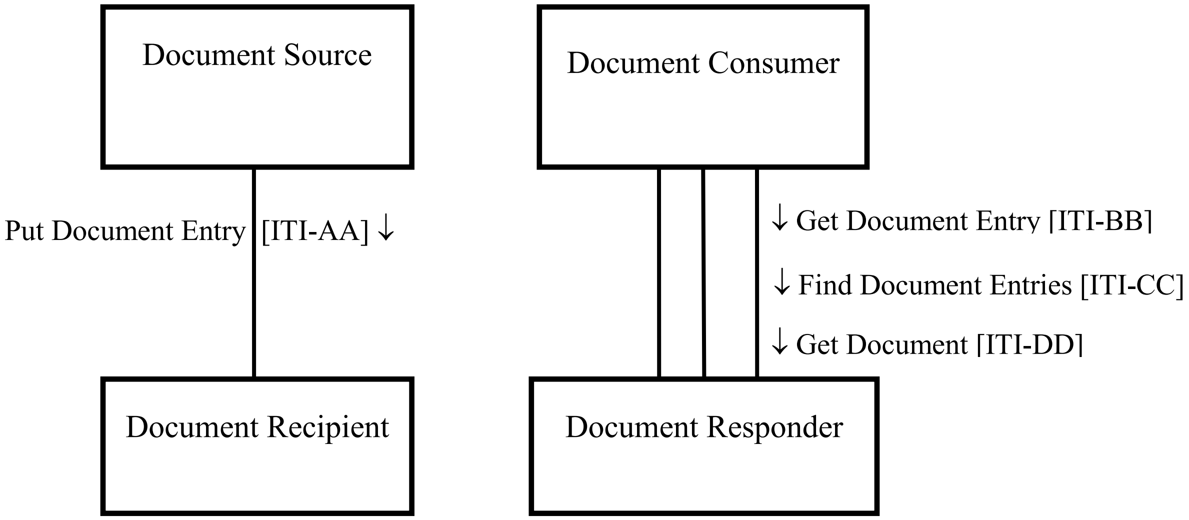


Figure X.1-1: MHD Actor Diagram

255 Table X.1-1 lists the transactions for each actor directly involved in the MHD Profile. In order to claim support of this Profile, an implementation of an actor must perform the required transactions (labeled "R") and may support the optional transactions (labeled "O"). Actor groupings are further described in Section X.3.

260 **Table X.1-1: MHD - Actors and Transactions**

Actors	Transactions	Optionality	Section in Vol. 2
Document Source	Put Document Entry [ITI-AA]	R	ITI TF-2b:3.AA
Document Recipient	Put Document Entry [ITI-AA]	R	ITI TF-2b:3.AA
Document Consumer	Get Document Entry [ITI-BB]	O (Note 1)	ITI TF-2b:3.BB
	Find Document Entries [ITI-CC]	O (Note 1)	ITI TF-2b:3.CC
	Get Document [ITI-DD]	O (Note 1)	ITI TF-2b:3.DD
Document Responder	Get Document Entry [ITI-BB]	R	ITI TF-2b:3.BB
	Find Document Entries [ITI-CC]	R	ITI TF-2b:3.CC
	Get Document [ITI-DDD]	R	ITI TF-2b:3.DD

Note 1: Document Consumer shall implement at least one transaction: Get Document Entry, Find Document Entries, or Get Document.

X.1.1 Actor Descriptions and Actor Profile Requirements

The Document Source and Document Consumer actors are designed so that they can easily be 265 implemented on a mobile device, and yet have sufficient functionality to support a wide range of mobile applications and use cases.

The Document Recipient and Document Responder are expected to be implemented in a service environment and thus do not have the mobile device constrained environment.

The transactions in the MHD Profile correspond to the following equivalent transactions used in XDS.

- MHD Put Document Entry→ XDS Provide and Register
- MHD Get Document Entry → XDS Registry Stored Query – GetDocuments
- MHD Find Document Entries → XDS Registry Stored Query – FindDocuments
- MHD Get Document → XDS Retrieve Document Set

The MHD transactions are not precisely equal to the XDS transactions as the MHD profile provides less functionality. These limitations are:

- the MHD PutDocumentEntry can only publish one new document at a time and it does not support submission set metadata.
- the MHD GetDocumentEntry and GetDocument can only pull one document at a time.
- the MHD FindDocumentEntries supports only the OR operator within parameters.
- there is no access to SubmissionSets, Folders, or Associations.

In XDS, the Document Registry and Document Repository actors are independent to enable the widest possible deployment architectures. In contrast, the MHD profile combines the Registry and Repository functionality in one MHD Document Responder. This is expected to ease configuration needs on the mobile health application and mobile health application deployment, and reduce the overall solution complexity. The MHD Document Recipient and the MHD Document Responder actors are independent because there are use cases where only one is needed, such as supporting a mobile medical measuring device that simply creates and submits new documents. More general purpose systems would likely implement both of these actors to provide a complete service definition for the hosting organization.

Due to these simplifying constraints, the MHD profile can be used as an interface to an XDS environment, but the MHD profile does not support all of the functionality supported by the XDS Document Source and XDS Document Consumer.

X.2 MHD Actor Options

Options that may be selected for this Profile are listed in the table X.2-1 along with the Actors to which they apply. Dependencies between options when applicable are specified in notes.

Table X.2-1: MHD - Actors and Options

Actor	Options	Volume & Section
Document Source	*No options defined*	- -
Document Recipient	*No options defined*	- -
Document Consumer	*No options defined*	- -

Actor	Options	Volume & Section
Document Responder	*No options defined*	- -

X.3 MHD Actor Required Groupings

Actor(s) which are required to be grouped with another Actor(s) are listed in this section. The grouped Actor may be from this profile or a different domain/profile. These mandatory required groupings, plus further descriptions if necessary, are given in the table below.

An Actor from this profile (Column 1) must implement all of the required transactions in this profile in addition to all of the required transactions for the grouped profile/actor listed (Column 2).

Table X.3-1: MHD - Actors Required Groups

MHD Actor	Actor to be grouped with	Technical Framework Reference	Content Binding Reference
Document Source	None		
Document Recipient	None		
Document Consumer	None		
Document Responder	None		

X.4 MHD Overview

This profile assumes that the prime resource that is being operated on via RESTful operations is the XDS Document Entry, which is the metadata that describes a document. Thus the profile's prime focus is on actions upon the Document entry, rather than the document itself. But the profile does also provide access to the document.

The MHD Profile defines a base URL pattern that includes, as a mandatory component, a patient identifier. This is a typical HTTP RESTful pattern and has the advantage of making it clearer that these are patient-centric transactions. Where this mobile device gets the patient ID is out of scope for the MHD profile with expectations that this could come from a previous browser session, some service call, or be configured. The inclusion of the Patient Identity on the URL should make the privacy and security enforcement model more straightforward (See Security Considerations).

X.4.1 Concepts

The MHD profile supports a broad set of the XDS use cases and functionality while keeping the technology as simple as possible. Example Use cases are:

- Medical devices such as those targeted by the Patient Care Devices (PCD) domain or Continua organization, submitting data in the form of documents.
- Kiosks used by patients in hospital registration departments, where it is anticipated that a hospital staff member will review, edit, and approve the document before it is allowed into the hospital system.
- PHR publishing into a staging area for subsequent import into an EHR or HIE.
- Patient or provider application that is configured to securely connect to a PHR in order to submit Recording history document.
- Electronic measurement device participating in an XDW workflow and pulling medical history documents from an HIE.
- A General Practitioner physician's office with minimal IT capabilities using a mobile application to connect to an HIE or EHR.

These use cases can be generalized into two general use cases. The first general use case is one where a new document is published from the mobile device. The second general use case is one where the mobile device needs to discover available documents and retrieve documents of interest. There are clearly complex use cases that combine these two general use cases. These are not specifically diagramed.

X.4.2 Use Case #1: Publication of new documents

X.4.2.1 Publication of new documents Use Case Description

In this use case there is a single new document that is published from the mobile device. An example might be that the mobile device is a medical device that has acquired new health measurements, or the mobile device has a user-interface used to capture user input such as a Patient Consent.

The use cases presume that the mobile device knows the patient identity. The patient identity might be obtained through some transactional method such as PIX/PDQ, might simply be entered via some device interface (RFID, Bar-Code), a user interface, or be specified in a configuration setting (e.g., mobile PHR Application). This use case also presumes that the mobile device knows the location of the URL endpoints, likely through a configuration setting, or a workflow driven by a web interface.

X.4.2.2 Publication of new documents Process Flow

The publication of a new document is done using the PutDocumentEntry transaction, which carries both the document entry metadata and the document.

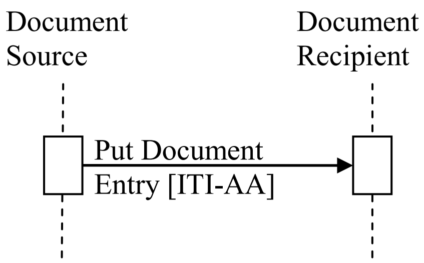


Figure X.4.2.2-1: Basic Process Flow in MHD Profile

X.4.3 Use Case #2: Discovery and Retrieval of existing documents

X.4.3.1 Discovery and Retrieval of existing documents Use Case Description

In this use case the mobile device needs access to existing documents. An example might be that the mobile device is involved in a workflow and needs to determine the current state of the workflow, or where the mobile device needs to discover the most current medical summary.

X.4.3.2 Discovery and Retrieval of existing documents Process Flow

The Find Document Entries transaction is used to provide parameterized queries that result in a set of Document Entries.

The Get Document Entry transaction is used to get the metadata for a specific Document Entry.

The Get Document transaction is used to get the document itself.

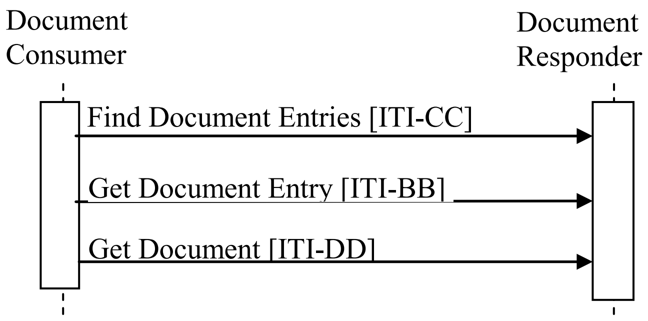


Figure X.4.3.2-1: Basic Process Flow in MHD Profile

X.5 MHD Security Considerations

There are many security and privacy concerns with mobile devices, simply because they are harder to physically control. Many common information technology uses of HTTP, including the RESTful pattern, are accessing far less sensitive information than health documents. These factors present an especially difficult challenge for the security model. It is recommended that application developers utilize a Risk Assessment in the design of the applications, and that the operational environment utilize a Risk Assessment in the design and deployment of the operational environment.

There are many reasonable methods of securing the interoperability transactions. These security models can be layered in without modifying the characteristics of the MHD profile transactions. The use of TLS is encouraged, specifically the use of the ATNA profile. User authentication on mobile devices is typically handled by a more light weight authentication system such as HTTP Authentication, OAuth, or Open-ID. These user authentication methods do have specifications (WS-Trust) for interacting with SAML, such as defined in the XUA profile. IHE does have a good set of profiles for the use of Enterprise User Authentication (EUA) on HTTP based devices, with bridging to Cross-Enterprise User Assertion (XUA) for the backend. In all of these cases the network communication security, and user authentication are layered in at the HTTP transport layer and thus don’t modify the interoperability characteristics defined in the MHD profile.

The Security Audit logging (e.g., ATNA) is not mandated by the MHD profile, although it is recommended. Support for ATNA-based audit logging on the mobile health device may be beyond the ability of this constrained environment. This would mean that the operational environment must choose how to mitigate the risk of relying only on the service side audit logging.

The Resource URL pattern defined in this profile does include the Patient ID. The advantage of this is to place clear distinction of the patient identity on each transaction, thus enabling strong patient-centric privacy and security controls. This URL pattern does present a risk when using typical web server audit logging of URL requests, and browser history. In both of these cases the URL with the patient identity is clearly visible. These risks need to be mitigated in system or operational design.

X.6 MHD Cross Profile Considerations

X.6.1 MHD Actor grouped with XDS infrastructure

When the MHD Document Recipient actor is acting as is a proxy for an XDS environment, it could be grouped with an XDS Document Source or an XDS Integrated Document Source/Repository. In this way the Put Document Entry transaction would be converted by the grouped system into an XDS Provide and Register Document Set-b transaction. It is expected that this system would be configured to support only a designated set of mobile devices authorized by the hosting organization and use the security model defined by that hosting organization. The proxy would be expected to fill in any necessary missing information, convert any user authentication credentials, and implement fully the IHE ATNA Secure Node or Secure Application actors.

When the MHD Document Responder actor is acting as a proxy for an XDS environment, it could be grouped with an XDS Document Consumer. In this way the Get Document Entry, Find Document Entries, and Get Document transactions will be supported in the system through the use of the XDS Registry Stored Query and XDS Retrieve Document Set-b transactions as needed. It is expected that this proxy would be configured to support a designated set of mobile devices and the security model defined by the hosting organization. The proxy would be expected to fill in any necessary missing information, convert any user authentication credentials, and implement fully the IHE-ATNA Secure Node or Secure Application actors.

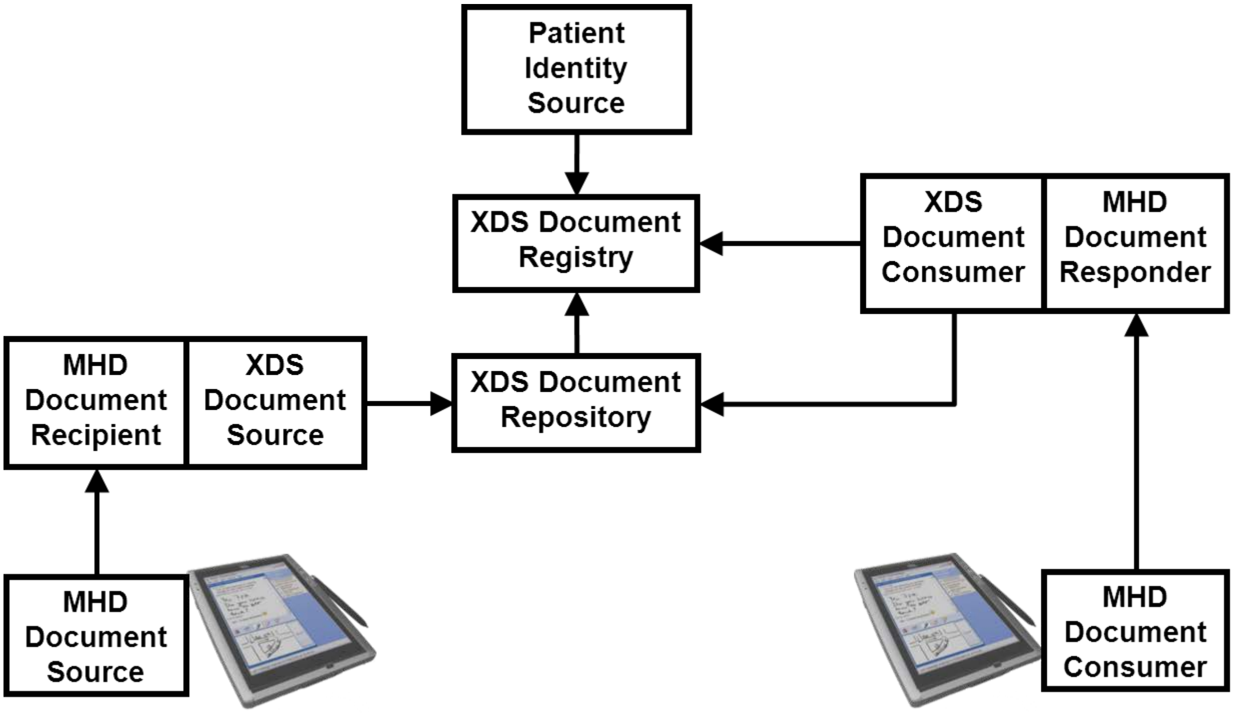
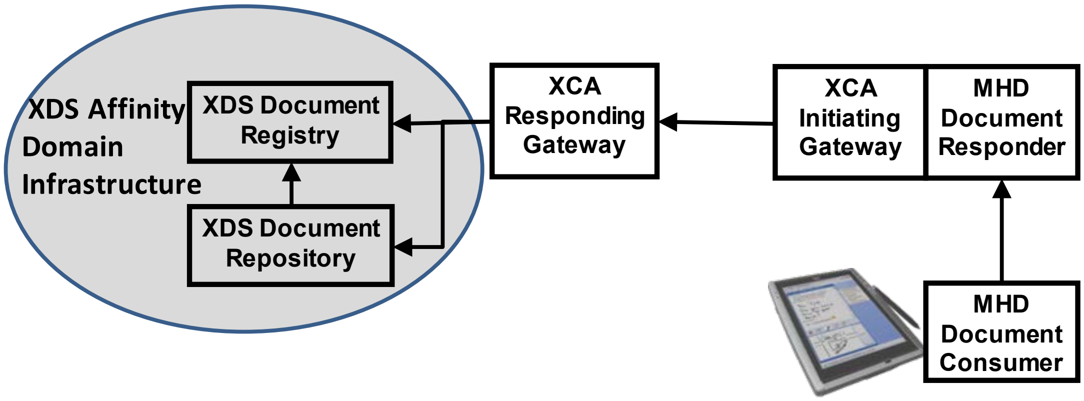


Figure X.6.1-1: MHD Actors grouped with XDS

X.6.2 MHD Actor grouped with XCA infrastructure

When a MHD Document Responder acts as a proxy into an XCA environment, it could be grouped with an XCA Initiating Gateway. This type of MHD Document Responder will support the Find Document Entries and Get Document transactions by utilizing the XCA Cross Gateway Query and XCA Cross Gateway Retrieve transactions as necessary. This type of proxy would be configured to support a designated set of mobile devices and enable a security model as defined by the hosting organization. The proxy would be required to fill in any necessary missing information, convert any user authentication credentials, and implement fully the IHE-ATNA Secure Node or Secure Application.



435 **Figure X.6.2-1: MHD Actors grouped with XCA**

Appendices

Actor Summary Definitions

Update (and add) the following terms to the IHE TF General Introduction Namespace list of Actors:

Document Source - The Document Source Actor is the producer and publisher of documents **and metadata**~~. It is responsible for sending documents to a Document Repository Actor. It also supplies metadata to the Document Repository Actor for subsequent registration of the documents with the Document Registry Actor.~~

Document Consumer - The Document Consumer Actor queries for document metadata meeting certain criteria, and may retrieve selected documents.

Document Recipient: This actor receives ~~a set of~~ documents **and metadata** sent by another actor. ~~Typically this document set will be made available to the intended recipient who will choose to either view it or integrate it into a Health Record.~~

Document Responder - The receiver of and responder to requests for document entries and documents.

Transaction Summary Definitions

Add the following terms to the IHE TF General Introduction Namespace list of Transactions:

Put Document Entry This transaction is used to transfer a document and metadata, equivalent to a Provide and Register Document Set-b transaction.

Get Document Entry – This transaction is used to get the metadata for a particular Document Entry.

Find Document Entries – This transaction is used to provide parameterized queries that result in a list of Document Entries.

Get Document – This transaction is used to get a single document.

Volume 2 – Transactions

Add sections 3.AA, 3.BB, 3.CC and 3.DD

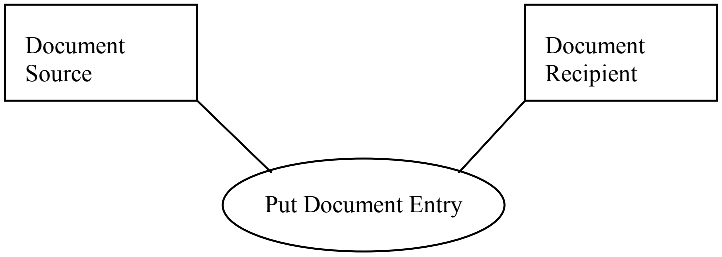
3.AA Put Document Entry ITI-AA

This section corresponds to Transaction ITI-AA of the IHE Technical Framework. Transaction ITI-AA is used by the Document Source and Document Recipient actors.

3.AA.1 Scope

This transaction is used to publish a new document entry and document.

3.AA.2 Use Case Roles



Actor: Document Source

Role: Sends Document Entry and Document to the Receiver for publication.

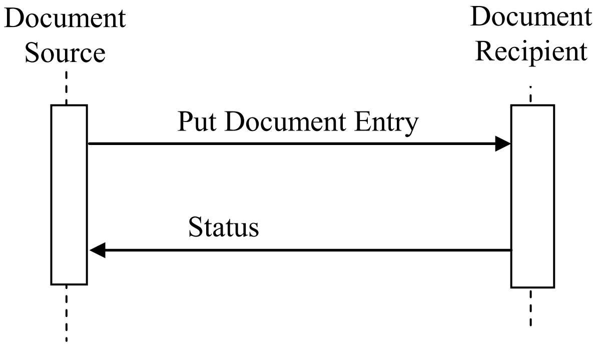
Actor: Document Receiver

Role: Accepts the document and metadata sent from the Source.

3.AA.3 Referenced Standard

RFC2616	IETF Hypertext Transfer Protocol –HTTP/1.1
RFC3986	IETF Uniform Resource Identifier (URI): Generic Syntax
RFC4627	The application/json Media Type for JavaScript Object Notation (JSON)

3.AA.4 Interaction Diagram



3.AA.4.1 Put Document Entry Message

This message uses the HTTP POST method on the target Document Entry to convey the metadata and document.

3.AA.4.1.1 Trigger Events

This method is sent when the Document Source needs to create a Document Entry.

3.AA.4.1.2 Message Semantics

The HTTP POST method is used, with HTTP Accept set to application/json. The message is a MIME multi-part which conforms to the following requirements:

1. The first mime part is the metadata encoded using the JSON encoding of DocumentEntry.
2. The second mime part is the document.

The Source shall generate a unique UUID for the entryUUID using the well-known rules for UUID creation. The Put Document Entry is sent to the URL for this patient and entryUUID, i.e., the intended Document Entry. The format for a Document Entry URL is:

http://<location>/<patientID>/DocumentEntry/<entryUUID>/

Where:

Location – a locally defined root part of arbitrary path.

patientID – the CX encoded patient ID (where XDS is used, this is the XDS Affinity Domain Patient ID). This value may need to be transformed for URL encoding.

entryUUID – the UUID value.

For example:

http://blah.com/blah/144ba3c4aad24e9%5E%5E%5E%261.3.6.1.4.1.21367.2005.3.7%26ISO/DocumentEntry/14a9fdec-0af4-45bb-adf2-d752b49bcc7d/

3.AA.4.1.2.1 JSON encoding of a Document Entry

Where XDS Metadata values are encoded using HL7 v2, these are simply expressed as the HL7 v2 encoded string. See ITI TF-3: Table 4.1-3 Data Types. These values result in strings and therefore are used without modification.

Where Metadata values are encoding in XDS using XML, we transform them into JSON value strings as shown in Table 3.AA.4.1.2.1-1 XDS XML Data Type Encoding for mHealth.

Table 3.AA.4.1.2.1-1: XDS XML Data Type Encoding for mHealth

XDS Data Type	JSON encoding	JSON example

Coded Value	classCode:{
	code:*string*,
	codingScheme:*string*,
	codeName:*string*
	}
		classCode: {
		code:"2345-3",
		codingScheme:"OINK",
		codeName:"Authorte garbage"
		}

Author value	Author: {
	authorInstitution:*XON*,
	authorPerson:*XCN*,
	authorRole:*string*,
	authorSpecialty:*string*
	}
		Author:{
		authorInstitution:”Hospital^^^^^^^^^^1.2.3.4.5.6.7.8.9.1789.45”,
		authorPerson:”Name of Author”,
		authorRole:”name of role”,
		authorSpecialty:”specialty of author”
		}

The XDS Document Entry metadata as defined in Table ITI TF-3:4.1-5 are encoded using JSON according to Table 3.AA.4.1.2.1-2 XDS Document Entry JSON encoding. All other encoding and validation rules found in XDS apply.

Table 3.AA.4.1.2.1-2: XDS Document Entry JSON encoding

XDSDocumentEntry Name	JSON value encoding
Author	`Author value`
availabilityStatus	`String`
classCode	`Coded value`
Comments	`String`
confidentialityCode	`Coded value`
creationTime	`DTM`
entryUUID	`UUID`
eventCodeList	`Coded Value`
formatCode	`Coded value`

XDSDocumentEntry Name	JSON value encoding
hash	`String`
healthcareFacilityTypeCode	`Coded Value`
homeCommunityId	`anyURI`
languageCode	`String`
legalAuthenticator	`XCN`
mimeType	`String`
patientId	`CX`
practiceSettingCode	`Coded Value`
repositoryUniqueId	`String`
serviceStartTime	`DTM`
serviceStopTime	`DTM`
size	`String`
sourcePatientId	`CX`
sourcePatientInfo	`String`
title	`String`
typeCode	`Coded value`
uniqueId	`String`

Note: Not all of these attributes are valid for PutDocumentEntry transaction. See the Provide and Register Document Set-b [ITI-41] transaction for rules for use with XDS.

All values shall be encoded using RFC-1738 rules.

Any attribute with a single value shall be encoded as the attribute name and value.

```
comments: "This is a comment"
```

Any attribute that has multiple values shall be encoded as a JSON array.

```
comments: ["This is a comment", "This is also a comment"]
```

Any attribute made up of multiple attributes shall be encoded as a JSON object

```
classCode: { code:"2345-3",codingScheme:"OINK", codeName:"garbage"}
```

An attribute with multiple values and multiple attributes would look like:

```
classCode: [{code:"2345-3",codingScheme:"OINK", codeName:"garbage"},
{code:"2345-2",codingScheme:"OINK", codeName:"trash"}]
```

A complete document entry would be encoded as an anonymous JSON object (See Open Issue MHD_019) (Note: the example below does not include all XDS required attributes)

```
{patientID: "144ba3c4aad24e9^^^&1.3.6.1.4.1.21367.2005.3.7&ISO" ,
classCode: [{code:"2345-3",codingScheme:"OINK",
codeName:"garbage"},{code:"2345-2",codingScheme:"OINK",
codeName:"trash"}],
title="document title",entryUUID:”urn:uuid:14a9fdec-0af4-45bb-adf2-
d752b49bcc7d “}
```

3.AA.4.1.3 Expected Actions

The Document Recipient shall verify the Document Entry attributes for consistency with the requirements as specified for attributes sent through the Provide and Register Document Set-b [ITI-41] transaction when used with XDS.

When the MHD Document Recipient is grouped with an XDS Document Source the Document Entry shall be transformed into a proper Provide and Register Document Set-b [ITI-41] transaction when used with XDS. The Document Recipient shall create appropriate SubmissionSet metadata based on the Document Entry metadata. Some SubmissionSet metadata is not directly derivable; these values are left to the implementer of the Document Recipient. Table 3.AA.4.1.3-1 is provided as guidance and is not to be considered mandatory mapping.

Table 3.AA.4.1.3-1: XDS Submission Set potential derivation

XDSSubmissionSet Attribute	Potentially Derived from
author	`DocumentEntry.author`
comments	`DocumentEntry.comment`
contentTypeCode	Chosen from a lookup table based on mobile device ID, or other document metadata like `classCode or formatCode`
intendedRecipient	Configured value, derived from specific use case, or left empty
patientId	`DocumentEntry.patientID`
sourceId	Configured value indicating the identity of the MHD Document Recipient
submissionTime	The current date/time
title	`DocumentEntry.title`

3.AA.4.2 Status Message

The Document Recipient returns a HTTP Status code appropriate to the processing.

3.AA.4.2.1 Trigger Events

This message shall be sent once the document is received and completely processed.

3.AA.4.2.2 Message Semantics

If the Document Recipient has fully processed the Put transaction then the Document Recipient shall return the HTTP response code 201 – Created to indicate success.

If the Document Recipient cannot recognize the posted data, then the Document Recipient shall return the HTTP response code 400 – Bad Request.

HTTP GET or PUT with an If-Unmodified-Since header shall result in HTTP response code 501 – Not implemented. Note: Other HTTP response codes may be returned by the Document Recipient Actor, indicating conditions outside of the scope of this transaction, for example, 401 – Authentication Failed might be returned if Document Recipient Actor is grouped with the EUA profile Kerberized Server Actor.

The Document Recipient Actor should complement the returned error code with a human readable description of the error condition.

Document Recipient Actors may return HTTP redirect responses (responses with values of 301, 302, 303 or 307) in response to a request. Document Source Actors must follow redirects, but if a loop is detected, it may report an error.

3.AA.4.2.3 Expected Actions

Process the results according to application defined rules.

If an error condition cannot be automatically recovered, at a minimum, the error should be displayed to the user by the Document Source Actor.

3.AA.5 Security Considerations

See the general Security Considerations in ITI TF-1:X.6

3.AA.5.1 Security Audit Considerations

The Security audit criteria are similar to those for the XDS profile as the transaction does export a document. Grouping with an ATNA Secure Node or Secure Application is recommended, but not mandated. The Document Source may be considered overburdened to fully implement the requirements of Secure Node or Secure Application. The Document Recipient is more full featured and should generate the equivalent to the audit event defined in ITI TF-2b:3.41.7.1.2 Document Repository or Document Recipient audit message.

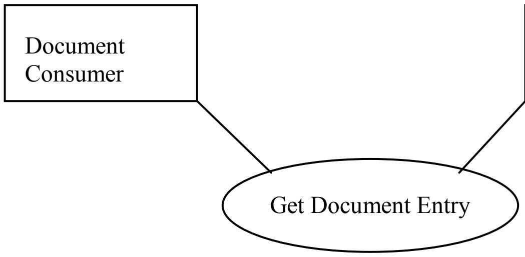
3.BB Get Document Entry ITI-BB

This section corresponds to Transaction ITI-BB of the IHE Technical Framework. Transaction ITI-BB is used by the Document Consumer and Document Responder actors.

3.BB.1 Scope

This transaction is used to get a document entry, the metadata for a document.

3.BB.2 Use Case Roles



Actor: Document Consumer

Role: Requests a Document Entry from a Document Responder

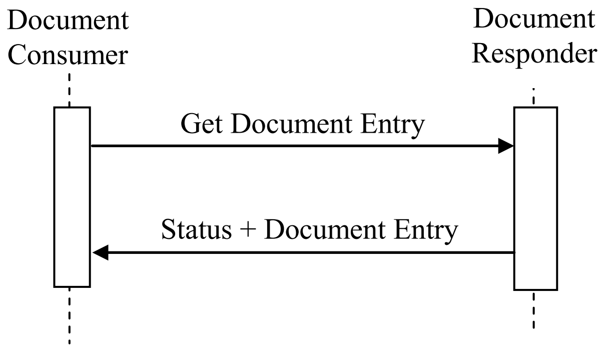
Actor: Document Responder

Role: Provides the Document Entry.

3.BB.3 Referenced Standard

RFC2616	IETF Hypertext Transfer Protocol –HTTP/1.1
RFC3986	IETF Uniform Resource Identifier (URI): Generic Syntax
RFC4627	The application/json Media Type for JavaScript Object Notation (JSON)

3.BB.4 Interaction Diagram



605 ### 3.BB.4.1 Get Document Entry Message

This message uses the HTTP GET method on the target Document Entry to retrieve the document metadata.

3.BB.4.1.1 Trigger Events

This method is sent when the Document Consumer needs to get a specific Document Entry.

610 #### 3.BB.4.1.2 Message Semantics

The Document Entry URL includes the patient ID and the entryUUID. The format for the Document Entry URL is:

http://<location>/<patientID>/DocumentEntry/<entryUUID>/

Where:

615 **location** – a locally defined root part of arbitrary path

patientID – the CX encoded patient ID (where XDS is used, this is the XDS Affinity Domain Patient ID). This value may need to be transformed URL encoding.

entryUUID – the UUID value

HTTP If-Unmodified-Since header shall not be included in the GET request.

620 #### 3.BB.4.1.3 Expected Actions

The Document Responder shall gather the Document Entry represented by the patientIdD and entryUUID.

When the Document Responder is grouped with an XDS Document Consumer the Document Entry can be gathered through the use of the XDS Registry Stored Query [ITI-18] -

GetDocuments given the entryUUID. The Document Responder shall verify that the DocumentEntry to be returned is consistent with the patientID.

3.BB.4.2 Status + Document Entry Message

The Document Responder returns a HTTP Status code appropriate to the processing as well as the contents of the requested Document Entry.

3.BB.4.2.1 Trigger Events

This message shall be sent once the document entry is retrieved.

3.BB.4.2.2 Message Semantics

The HTTP body shall be the Document Entry encoded using JSON as defined in Section 3.AA.4.1.2.1 JSON encoding of a Document Entry.

If the patientID or entryUUID are missing or malformed, the Document Responder Actor shall return HTTP response code 400 - Bad Request.

If the specified patientID or entryUUID is not known to the Document Responder Actor, it shall return HTTP response-code 404 (not found) with the suggested reason-phrase “Document Entry UUID not found”.

If the HTTP request specified is otherwise not a legal value according to this profile, the Document Responder Actor shall return HTTP response-code 403 (forbidden) with the suggested reason-phrase “request type not supported”.

Note: Other HTTP response codes may be returned by the Document Responder Actor, indicating conditions outside of the scope of this profile, for example, 401 – Authentication Failed might be returned if Document Responder Actor is grouped with the EUA profile Kerberized Server Actor.

The Document Responder Actor should complement the returned error code with a human readable description of the error condition.

Document Responder Actors may return HTTP redirect responses (responses with values of 301, 302, 303 or 307) in response to a request. Document Consumer Actors must follow redirects, but if a loop is detected, it may report an error.

3.BB.4.2.3 Expected Actions

Process the results according to application defined rules.

If an error condition cannot be automatically recovered, at a minimum, the error should be displayed to the user by the Document Consumer Actor.

3.BB.5 Security Considerations

See the general Security Considerations in ITI TF-1:X.6

3.BB.5.1 Security Audit Considerations

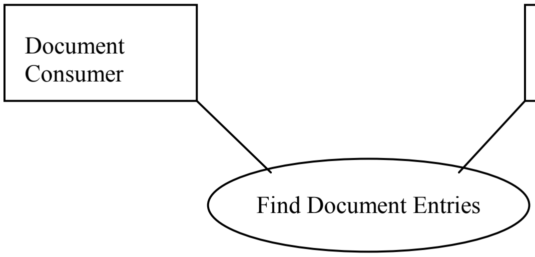
The Security audit criteria are similar to those for the XDS profile as the transaction does import a document entry. Grouping with a Secure Node or Secure Application is recommended, but not mandated. The Document Consumer may be considered overburdened to fully implement the requirements of Secure Node or Secure Application. The Document Responder is more full featured and should generate the equivalent of the audit event defined in ITI TF-2a:3.18.5.1.2 Document Registry audit message.

3.CC Find Document Entries ITI-CC

This section corresponds to Transaction ITI-CC of the IHE Technical Framework. Transaction ITI-CC is used by the Document Consumer and Document Responder actors.

3.CC.1 Scope

The Find Document Entries transaction is used to get the Document Entries that satisfy a number of parameters, equivalent to ITI-18 (Registry Stored Query), FindDocuments stored query from ITI TF-2a:3.18.4.1.2.3.7.1:



3.CC.2 Use Case Roles

Actor: Document Consumer

Role: Requests document entries satisfying a set of key/value pair encoded metadata attributes

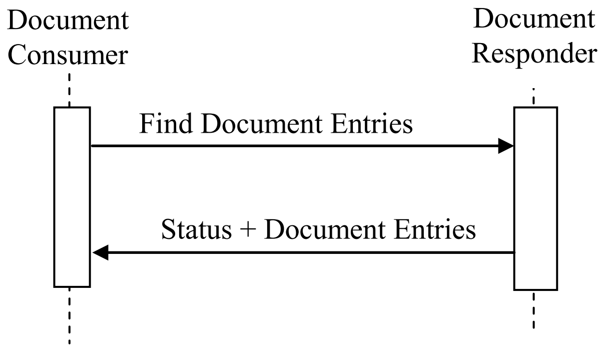
Actor: Document Responder

Role: Services the query and returns one or more Document Entries.

3.CC.3 Referenced Standard

RFC2616	IETF Hypertext Transfer Protocol –HTTP/1.1
RFC3986	IETF Uniform Resource Identifier (URI): Generic Syntax
RFC4627	The application/json Media Type for JavaScript Object Notation (JSON)

3.CC.4 Interaction Diagram



3.CC.4.1 Find Document Entries Message

This is a GET parameterized query from a Document Consumer to a Document Responder

3.CC.4.1.1 Trigger Events

This method is sent when the Document Consumer needs to discover Document Entries meeting various XDS metadata parameters.

3.CC.4.1.2 Message Semantics

The Find Document Entries message is an HTTP GET request that can be sent to the FindDocuments URL

http://<location>/<PatientID>/FindDocumentEntries?<parameters>

Where:

location – a locally defined root part of arbitrary path.

patientID – the CX encoded patient ID (where XDS is used, this is the XDS Affinity Domain Patient ID). This value may need to be transformed for URL encoding.

Parameters – key/value pairs in accordance with RFC2616 for encoding GET queries. The query string (i.e., the string after the '?' and before the '#' in the HTTP action) is created as a list of key/value pairs, using the following table (from ITI TF-2a:3.18.4.1.2.3.7.1 FindDocuments) to identify the key names (which correspond to the control names in HTML) and to determine the multiplicity rules. The values are encoded using the encoding methods defined in Table ITI TF-3:4.1-3: Data Types. CE is the HL7 v2.5 data type for encoding coded values and is described in ITI TF-3a:3.18.4.1.2.3.4 "Coding of Code/Code-Scheme". Matching rules for coded values are as defined in XDS. Note that HTTP convention allows for multiple instances of a parameter to be requested with different values, multiple values of the same parameter name SHALL be treated in an OR relationship. There is no support for XDS query mechanism for query parameters of the same name in an AND relationship. If the “status” parameter is not specified, the value “urn:oasis:names:tc:ebxml-regrep:StatusType:Approved” shall be assumed.

XDS - Parameter Name	mHealth Parameter Name	Encoded
$XDSDocumentEntryClassCode	classCode	CE
$XDSDocumentEntryTypeCode	typeCode	CE
$XDSDocumentEntryPracticeSettingCode	practiceSettingCode	CE
$XDSDocumentEntryCreationTimeFrom	creationTimeFrom	DTM
$XDSDocumentEntryCreationTimeTo	creationTimeTo	DTM
$XDSDocumentEntryServiceStartTimeFrom	serviceStartTimeFrom	DTM
$XDSDocumentEntryServiceStartTimeTo	serviceStartTimeTo	DTM
$XDSDocumentEntryServiceStopTimeFrom	serviceStopTimeFrom	DTM
$XDSDocumentEntryServiceStopTimeTo	serviceStopTimeTo	DTM
$XDSDocumentEntryHealthcareFacilityTypeCode	healthcareFacilityTypeCode	CE
$XDSDocumentEntryEventCodeList	eventCodeList	CE
$XDSDocumentEntryConfidentialityCode	confidentialityCode	CE
$XDSDocumentEntryAuthorPerson	authorPerson	XCN
$XDSDocumentEntryFormatCode	formatCode	CE
$XDSDocumentEntryStatus	status	String

For example:

```
http://blah.com/blah/144ba3c4aad24e9%5E%5E%5E%261.3.6.1.4.1.21367.2005.3.7%26ISO/FindD
ocumentEntries?classCode=”2345-3%5EGarbage%5EOINK”&serviceStartTimeFrom=”200501020304”
```

3.CC.4.1.3 Expected Actions

The query will be processed using the same rules as for XDS Registry Stored Query for FindDocuments. This may be accomplished through grouping with a Document Consumer, and transforming the parameters and returned metadata.

3.CC.4.2 Status + Document Entries Message

The Document Responder returns a HTTP Status code appropriate to the processing as well as a list of the matching document entries.

3.CC.4.2.1 Trigger Events

This message is sent containing the Document Entries found using the query parameters as soon as the result is ready.

3.CC.4.2.2 Message Semantics

When the query is successful, the message contains a HTTP response status code of 200, and a body containing a JSON Array of JSON encoded Document Entries.

Example:

```
[
{patientID: "144ba3c4aad24e9^^^&1.3.6.1.4.1.21367.2005.3.7&ISO" ,
availabilityStatus: "urn:oasis:names:tc:ebxml-
regrep:StatusType:Approved", classCode: [{code:"2345-
3",codingScheme:"OINK",
codeName:"garbage"},{code:"2345-2",codingScheme:"OINK",
codeName:"trash"}],
title="document title number 1",entryUUID:”urn:uuid:14a9fdec-0af4-45bb-
adf2-d752b49bcc7d “},
{patientID: "144ba3c4aad24e9^^^&1.3.6.1.4.1.21367.2005.3.7&ISO" ,
availabilityStatus: "urn:oasis:names:tc:ebxml-
regrep:StatusType:Approved", classCode: [{code:"2345-
3",codingScheme:"OINK",
codeName:"garbage"},{code:"2345-2",codingScheme:"OINK",
codeName:"trash"}],
title="document title number 2",entryUUID:”urn:uuid:24a9fdec-0af4-45bb-
adf2-d752b49bcc7d “}
]
```

If the patientID is missing, the Document Responder Actor shall return HTTP response code 400 - Bad Request.

If the specified parameters do not result in Document Entries the Document Responder Actor shall return HTTP response-code 404 (not found) with the suggested reason-phrase “No Document Entries found”.

If the HTTP request specified is otherwise not a legal value according to this profile, the Document Responder Actor shall return HTTP response-code 403 (forbidden) with the suggested reason-phrase “request not supported”.

Note: Other HTTP response codes may be returned by the Document Responder Actor, indicating conditions outside of the scope of this profile, for example, 401 – Authentication Failed might be returned if Document Responder Actor is grouped with the EUA profile Kerberized Server Actor.

The Document Responder Actor should complement the returned error code with a human readable description of the error condition.

Document Responder Actors may return HTTP redirect responses (responses with values of 301, 302, 303 or 307) in response to a request. Document Consumer Actors must follow redirects, but if a loop is detected, it may report an error.

3.CC.4.2.3 Expected Actions

The return result shall be processed according to application behavior.

If an error condition cannot be automatically recovered, at a minimum, the error should be displayed to the user by the Document Consumer Actor.

3.CC.5 Security Considerations

See the general Security Considerations in ITI TF-1:X.6

3.CC.5.1 Security Audit Considerations

The Security audit criteria are similar to those for the XDS profile as the transaction does import a document entry. Grouping with an ATNA Secure Node or Secure Application is recommended, but not mandated. The Document Consumer may be considered overburdened to fully implement the requirements of Secure Node or Secure Application. The Document Responder is more full featured and should generate an equivalent event to the audit event defined in TF-2a:3.18.5.1.2 Document Registry audit message.

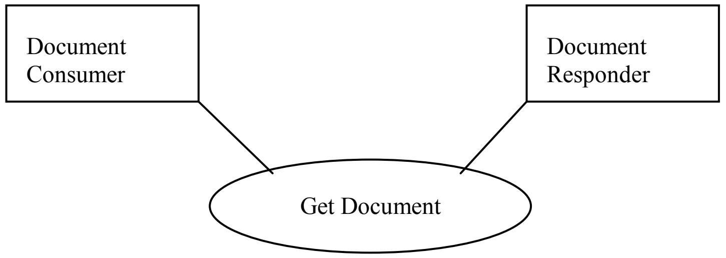
3.DD Get Document ITI-DD

This section corresponds to Transaction ITI-DD of the IHE Technical Framework. Transaction ITI-DD is used by the Document Consumer and Document Responder actors.

3.DD.1 Scope

The Get Document transaction is used by the Document Consumer to retrieve a document from the Document Responder.

3.DD.2 Use Case Roles



Actor: Document Consumer

Role: Requests a document identified by URL from the Document Responder

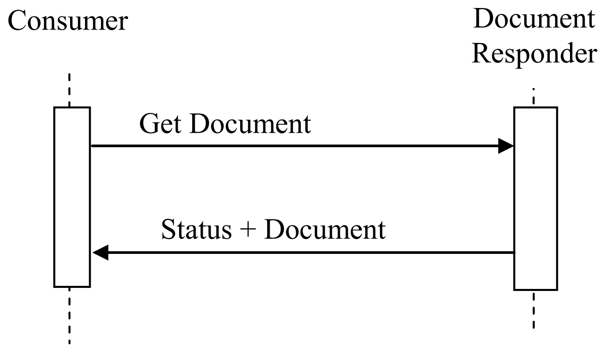
Actor: Document Responder

Role: Serves the document at the provided resource URL to the Document Consumer

3.DD.3 Referenced Standard

RFC2616	IETF Hypertext Transfer Protocol – HTTP/1.1

3.DD.4 Interaction Diagram



3.DD.4.1 GET Document Message

This message is a HTTP GET request to retrieve the document.

3.DD.4.1.1 Trigger Events

The Document Consumer needs a copy of an identified document.

3.DD.4.1.2 Message Semantics

The Document Consumer sends a HTTP GET request to the server. The Document Consumer may use content negotiation by providing a HTTP Accept header, according to the semantics of the HTTP protocols (see RFC 2616, section 14.1). The only MIME type assured to be returned is the MIME type indicated in the Document Entry.

http://<location>/<patientID>/Document/<entryUUID>/

Where:

location – a locally defined root part of arbitrary path.

patientID – the CX encoded patient ID (where XDS is used, this is the XDS Affinity Domain Patient ID). This value may need to be transformed for URL encoding.

entryUUID – the UUID value.

HTTP If-Unmodified-Since header shall not be included in the GET request.

3.DD.4.1.3 Expected Actions

The Document Responder shall provide the document in the requested MIME type, or reply with an HTTP status code indicating the error condition. The Document Responder is not required to transform the document.

3.DD.4.2 Status + Document Message

This is the return message sent by the Document Responder.

3.DD.4.2.1 Trigger Events

The HTTP Response message is sent when completing the GET Document request.

3.DD.4.2.2 Message Semantics

This message complies with the HTTP response message, as required by RFC 2616. The HTTP body contains the Document requested.

If the patientID or entryUUID are missing, the Document Responder Actor shall return HTTP response code 400 - Bad Request.

If the specified entryUUID is not known to the Document Responder Actor or it doesn't correlate to the provided patientID, it shall return HTTP response-code 404 (not found) with the suggested reason-phrase "Document Entry UUID not found".

If the Document Responder Actor is not able to format the document in any content types listed in the 'Accept' field, it shall return HTTP response code 406 – Not Acceptable.

If the HTTP request specified is otherwise not a legal value according to this profile, the Document Responder Actor shall return HTTP response-code 403 (forbidden) with the suggested reason-phrase "request type not supported".

Note: Other HTTP response codes may be returned by the Document Responder Actor, indicating conditions outside of the scope of this profile, for example, 401 – Authentication Failed might be returned if Document Responder Actor is grouped with the EUA profile Kerberized Server Actor.

The Document Responder Actor should complement the returned error code with a human readable description of the error condition.

Document Responder Actors may return HTTP redirect responses (responses with values of 301, 302, 303 or 307) in response to a request. Document Consumer Actors must follow redirects, but if a loop is detected, it may report an error.

3.DD.4.2.3 Expected Actions

The Document Consumer is expected to continue its workflow upon receiving the document.

If an error condition cannot be automatically recovered, at a minimum, the error should be displayed to the user by the Document Consumer Actor.

3.DD.5 Security Considerations

See the general Security Considerations in ITI TF-1:X.6

3.DD.5.1 Security Audit Considerations

The Security audit criteria are similar to those for the XDS profile as the transaction does import a document entry. Grouping with a Secure Node or Secure Application is recommended, but not mandated. The Document Consumer may be considered overburdened to fully implement the requirements of Secure Node or Secure Application. The Document Responder is more full featured and should generate an equivalent event to the audit event defined in ITI TF-2b:3.43.6.1.2 Document Repository audit message.