Integrating the Healthcare Enterprise

IHE IT Infrastructure (ITI)
Technical Committee White Paper
2008-2009

Publish/Subscribe Infrastructure for XDS.b

Version 2.0

August 22, 2008

1 Foreword

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

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

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

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

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

Table of Contents

2 Introduction

Event driven information exchange patterns dominate the data interchange in most healthcare settings. For example, most HL7 version 2.x interfaces send messages based on events within the sender's system. Most current IHE profiles assume either static, out-of-band determination of the senders and receivers of event driven information exchange, or describe query-response patterns. There is a need for a profiled dynamic infrastructure for event-driven information exchange patterns within IHE. This white paper describes such a framework based on the publish/subscribe data exchange model as applied to the XDS.b integration profile. An additional goal is to provide a blueprint for other IHE profiles to add publish/subscribe functionality.

Publish/subscribe patterns of data exchange are conceptually simple, and involve a limited numbers of actors and transactions. The transactions allow for automating the determination of information consumers based on events or content "topics". For example, if an IHE profile describes information content where a diagnosis is present and coded using a particular coding system, a subscriber can request to receive notifications when a diagnosis code from a particular set of codes is present. If the subscription is accepted, the system which keeps track of information recipients will start sending notifications when new data matches the described criteria.

The above example demonstrates two important issues which need to be addressed by profilers and implementers of publish/subscribe interactions. The first one is that the implementation of transactions and actors are dependent on the information exchange environment. Prescribing a specific technology for a general publish/subscribe infrastructure and transaction content is not a practical option. This paper describes the publish/subscribe actors and their transactions in the context of the XDS.b profile, with a secondary goal to serve as an example or blueprint for binding these actors and transactions to other IHE profiles.

The second issue that needs to be addressed is the use of subscription topics. Based on the information exchange environment, topics can be described in various ways. This paper presents a specific approach within the context of XDS.b, including a discussion on how topics can be extended in the future.

The implementation of publish/subscribe in healthcare environments also needs to take into account the need for security and privacy of the exchanged information. The described methodology to add publish/subscribe transactions to an IHE XDS affinity domain leverages existing access controls and auditing controls from the underlying profile, however there has been no risk assessment done for the proposed transactions.

The IHE ITI Technical Committee will appreciate feedback on this white paper, in particular comments about the implementability of the proposed transactions and workflows, and suggestions about the future direction this material might take.

2.1 Open Issues and Questions

1. This white paper uses WS-Notification. Is this choice appropriate?

2. Notification or pushing: the notification can be just a DocID, the Metadata, or the full document. This white paper uses the Document ID as the notification to limit security exposure and leverage access control points.
3. Topics/Filters: the white paper uses a stored query based model for filters. There are other possible models, e.g. XPath expressions.
4. Reflecting the Topic filter in the Notification - it is an option in WS-Notification, however revealing this information (as well as the producer reference) could be introducing additional risks to reveling patient information.
5. No risk assessment has been done on the transactions presented. This is necessary to be completed before a supplement to the ITI Technical Framework is published in the future.

2.2 Closed Issues

1. The PCC APS profile may have a specific use case of interest. Added to use cases
2. Reuse XDS transactions, or use specific web services transactions - use WS-Notification
    1. ebRIM/ebRS 3.0 define services compatible with these use cases. - use Stored Queries for topics
3. In this specification the only modification allowed to existing subscriptions is the end time of a subscription. While the underlying standard allows for other types of modifications, this paper adds this restriction for the sake of simplicity. Other types of modifications can be achieved via initiating a new subscription, and canceling the original – decision is to use subscription modification as described in WS-Notification.

2.3 Future Development

1. Expand the white paper to cover other use cases. This can be done, for example, as a cookbook for adding pub/sub to any IHE profile, or by defining a general publish/subscribe infrastructure, based on WS-Notification, where the topic/filter space and notification content are futrther specified using content profiles.

3 Use Cases

3.1 Unexpected Notification

A patient in the emergency department has all her relevant available documents retrieved via XDS.b transactions. As initial triage of the patient is done, an additional document regarding diagnostic results for this patient is registered in the XDS registry. Currently, there is no way for the Emergency department to learn about the existence of this new information. With a publish/subscribe infrastructure, the initial query to the registry would be accompanied with a subscription request, as a result of which a notification (or the document itself) would be send to the emergency department. The subscription will be terminated once the patient is no longer under the care of the emergency department's institution.

3.2 Long-term subscription

A patient visits their PCP after being discharged from a hospital, which belongs to the same XDS affinity domain as the provider's organization. The provider sends a query to the affinity domain registry, and retrieves the hospital discharge summary. The patient also has follow-up visits with a specialist at the hospital, and these visit summaries (including diagnostic test results) are registered in the affinity domain registry. Currently, the PCP would have to periodically query the registry for documents about the patient in order to retrieve the follow-up visit summaries. With a publish/subscribe infrastructure, the PCP would have a subscription for all his patients, so that notifications (or the documents themselves) would have been received as the summaries were registered in the affinity domain registry.

3.3 Antepartum Record Availability

From the Ante-partum Record profile under development in the PCC Technical Committee:

> During the 40 weeks of a typical pregnancy duration, the patient will have an initial History and Physical Examination, followed by repetitive office visits with multiple laboratory studies, imaging (usually ultrasound) studies, and serial physical examinations with recordings of vital signs, fundal height, and the fetal heart rate. As the patient is seen over a finite period in the office, aggregation of specific relevant data important to the evaluation of the obstetric patient upon presentation to Labor and Delivery is caputured on paper forms. The antepartum record contains the most critical information needed including the ongoing Medical Diagnoses, the Estimated Due Date, outcomes of any prior pregnancies, serial visit data on the appropriate growth of the uterus and assessments of fetal well being, authorizations, laboratory and imaging studies. This data must all be presented and evaluated upon entry to the Labor and Delivery Suite to ensure optimal care for the patient and the fetus.

Once the electronic means of communicating the Antepartum Record are established via this new profile, the ability of the PCC content consumer to establish a subscription for the Antepartum Record updates for a given expectant mother will enhance the ability to automate the delivery of the information in a timely manner.

3.4 Public Health Surveillance

For the purposes of detecting patterns of infectious disease outbreaks, public health organizations may need to receive notifications on topics based on document content, such as specific diagnoses. While this is similar to the long-term subscription use case, the different use of topics in this use case makes an important difference. There needs to be further investigation as to how topics would be represented in order for this use case to be addressed.

4 Actors/Transactions

The following diagram represents the main actors in a publish/subscribe setup (aka brokered notification). Note that the Publisher actor is added for completeness, as this white paper does not prescribe any specific method for relaying the occurrence of an event (e.g. a submission set registration to the XDS.b Registry) to the Notification Broker. It is possible, however, that other IHE profiles may find the Publisher actor useful within the context of their transaction flows.

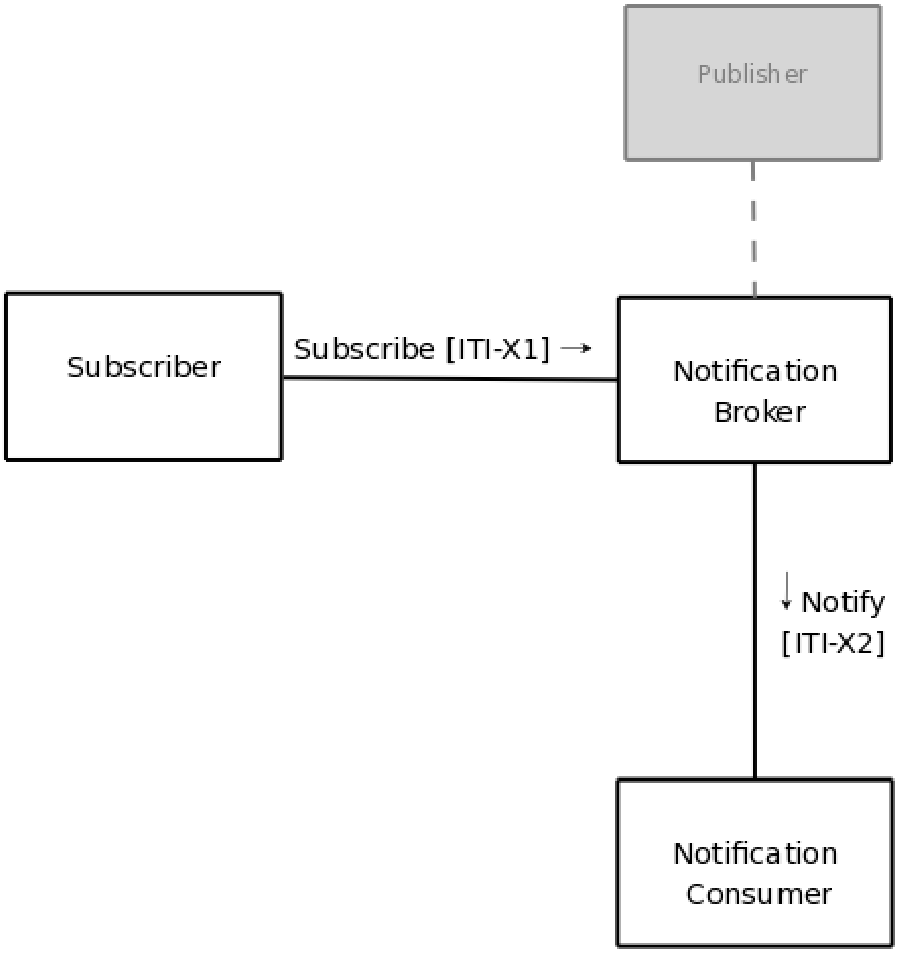


Figure Title 4.0: Main actors in a publish/subscribe setup

4.1 Actor Definitions

4.1.1 Subscriber

The Subscriber actor initiates, modifies, and terminates subscriptions on behalf of a Notification Consumer. Within an XDS affinity domain the subscriber will most likely be combined with a document consumer actor.

4.1.2 Notification Broker

The Notification Broker keeps track of all subscriptions in the XDS affinity domain, it must be aware of events in the XDS registry (e.g. that a new document has been registered), and based on the metadata associated with the document sends notifications to interested subscribers. This actor is the receiver of subscription requests, subscription modifications, and subscription cancellations. It also keeps track of the time limits of subscriptions. This actor may be combined with the XDS.b Document Registry actor.

4.1.3 Notification Consumer

This actor receives the notification about an event, when the subscription filters specified for this Notification Consumer are satisfied. Within XDS.b, this actor will very likely be grouped with a document consumer.

4.2 Transaction Definitions

4.2.1 Subscribe

This transaction is sent by the Subscriber to the Notification Broker in order to start a subscription for a particular set of topics, indicating possible start and end time for the subscription. Subscriptions can be modified. Any subscriber actor can cancel a subscription, as long as they have the subscription id.

4.2.2 Notify

This is a transaction from the Notification Broker to the Notification Consumers, sending a notification about the availability of a document or documents of interest, based on the subscribers' filter on selected topics.

4.3 Process Flow

The following diagram illustrates the process flow within an XDS.b affinity domain reflecting the use case presented in section 3.3:

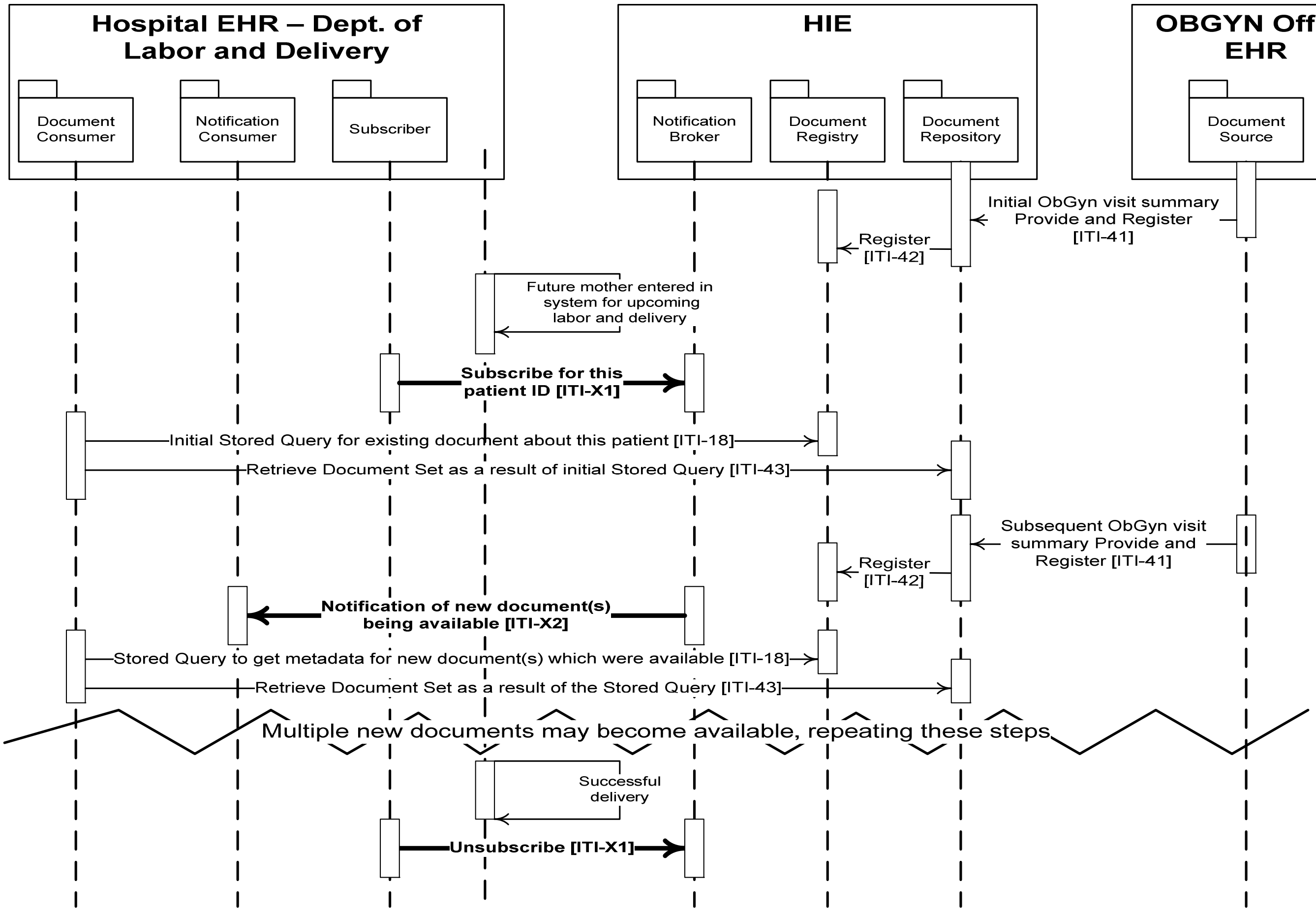

Hospital EHR – Dept. of Labor and Delivery
HIE
OBGYN Office EHR
Document Consumer
Notification Consumer
Subscriber
Notification Broker
Document Registry
Document Repository
Document Source
Initial ObGyn visit summary Provide and Register [ITI-41]
Register [ITI-42]
Future mother entered in system for upcoming labor and delivery
Subscribe for this patient ID [ITI-X1]
Initial Stored Query for existing document about this patient [ITI-18]
Retrieve Document Set as a result of initial Stored Query [ITI-43]
Subsequent ObGyn visit summary Provide and Register [ITI-41]
Register [ITI-42]
Notification of new document(s) being available [ITI-X2]
Stored Query to get metadata for new document(s) which were available [ITI-18]
Retrieve Document Set as a result of the Stored Query [ITI-43]
Multiple new documents may become available, repeating these steps
Successful delivery
Unsubscribe [ITI-X1]

Figure Title 4.2: Interaction diagram

The above interaction diagram is showing a grouping of a Document Consumer, a Notification Consumer and a Subscriber actors on one side, and a grouping of a Registry, a Repository and a Notification Broker actors on the other side. The emphasized transactions are described in this white paper, while the interactions with the grouped XDS.b actors are also shown. Note that the grouping presented here is not required.

4.4 Security Considerations

The generic form of the transactions in this white paper would only be able to have a minimal security considerations, because the topics would not be known, therefore no asset or threats can be assessed.

In this white paper we will assume the XDS.b risk assessment, and incorporate the mandatory controls defined.

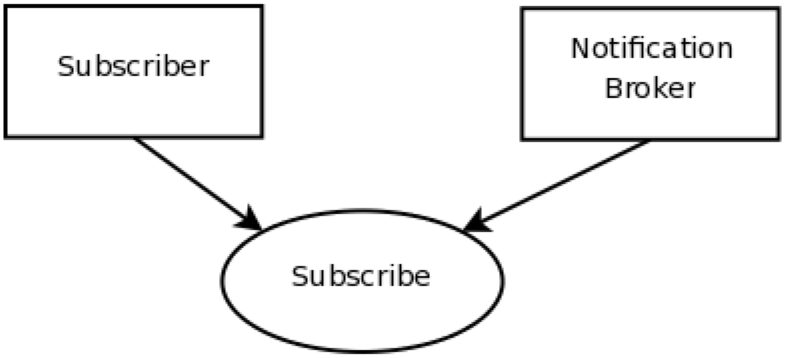
Detailed Transaction Definitions

5 Subscribe

5.1 Scope

This transaction involves a request by the Subscriber actor to the Notification Broker to start a subscription using a particular set of filters, or to modify or cancel an existing subscription.

5.2 Use Case Roles



Actor: Subscriber

Role: Sends, on the behalf of Notification Consumers, subscription requests, subscription modifications, or subscription cancellation messages to the Notification Broker

Actor: Notification Broker

Role: Manages subscriptions of Notification Consumers

5.3 Referenced Standards

OASIS Web Services Notification Family of Standards

WS-BaseNotification 1.3 OASIS Standard

WS-BrokeredNotification 1.3 OASIS Standard

WS-Topics 1.3 OASIS Standard

IHE ITI TF-2:3.18 - Registry Stored Query Transaction

5.4 Interaction Diagram

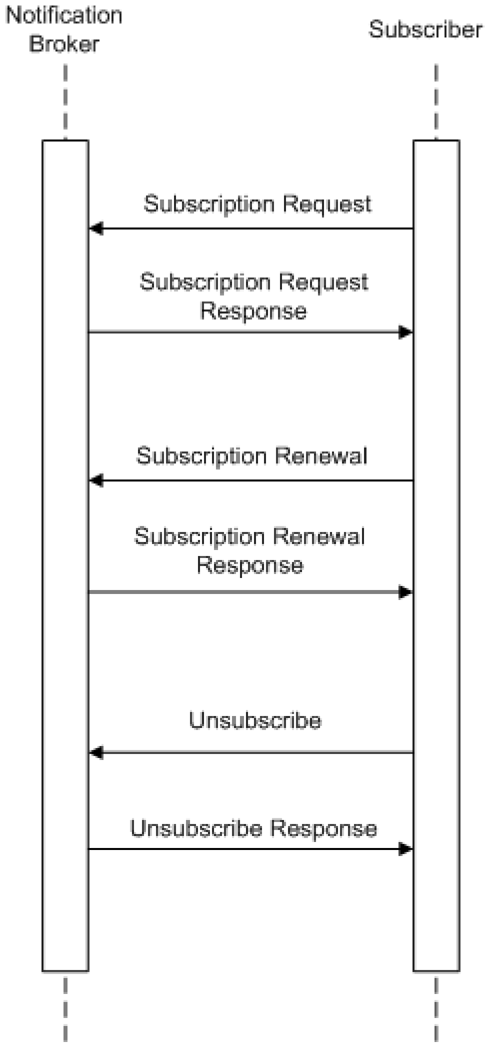


Figure 5-1 Subscribe Sequence

5.4.1 Subscription Request

5.4.1.1 Trigger

A Notification Consumer's need to initiate a subscription will cause the Subscriber to trigger a Subscription Request message.

5.4.1.2 Message Semantics

This white paper doesn't specify how the Notification Consumer and Subscriber communicate the need to initiate a subscription. One possible way to do that is to have a combined Notification Consumer and Subscriber actor.

The Subscription Request message shall comply with the requirements in the WS-BaseNotification standard. The *wsnt:ConsumerReference* element contains the Web Services end point where notifications must be sent. The *wsnt:Filter* element contains the topics and values for these topics for which a notification must be sent. This specification uses a "topic dialect" based on the XDS Stored Query syntax and semantics.

5.4.1.3 Expected Actions

The Notification Broker shall be capable of receiving multiple concurrent Subscription Requests and responding accordingly with Subscription Request Response messages.

The Notification Broker shall keep track of each unique subscription and will assign a unique referencable Web Services end point where subsequent modifications and a cancellation can be sent.

The Subscriber may indicate the duration of the subscription using the *wsnt:InitialTerminationTime* element, where a time stamp (expressed as an XML Schema *dateTime* data type value) or a duration (expressed as an XML Schema *duration* data type value) can be used.

5.4.1.4 Example

```
<?xml version="1.0" encoding="UTF-8"?>
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
    xmlns:a="http://www.w3.org/2005/08/addressing"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:wsnt="http://docs.oasis-open.org/wsn/b-2"
    xmlns:rim="urn:oasis:names:tc:ebxml-regrep:xsd:rim:3.0"
    xsi:schemaLocation="http://www.w3.org/2003/05/soap-envelope http://www.w3.org/2003/05/soap-
envelope http://www.w3.org/2005/08/addressing http://www.w3.org/2005/08/addressing/ws-addr.xsd
http://docs.oasis-open.org/wsn/b-2 http://docs.oasis-open.org/wsn/b-2.xsd
urn:oasis:names:tc:ebxml-regrep:xsd:rim:3.0 ../schema/ebRS/rim.xsd">
    <s:Header>
        <a:Action>http://docs.oasis-open.org/wsn/bw-
2/NotificationProducer/SubscribeRequest</a:Action>
        <a:MessageID>382dcdc7-8e84-9fdc-8443-48fd83bca938</a:MessageID>
    </s:Header>
    <s:Body>
        <wsnt:Subscribe>
            <!-- The Consumer on whose behalf the subscription is requested - the address where
the notification is to be sent -->
            <wsnt:ConsumerReference>
                <a:Address>https://NotificationConsumerServer/xdsBnotification</a:Address>
            </wsnt:ConsumerReference>
            <wsnt:Filter>
                <wsnt:TopicExpression Dialect="urn:ihe:iti:xds-b:pubsub:2008">
                    <rim:AdhocQuery id="urn:uuid:14d4debf-8f97-4251-9a74-a90016b0af0d">
                        <rim:Slot name="$XDSDocumentEntryPatientId">
                            <rim:ValueList>

<rim:Value>'st3498702^^^&amp;1.3.6.1.4.1.21367.2005.3.7&amp;ISO'</rim:Value>
                            </rim:ValueList>
                        </rim:Slot>
                        <rim:Slot name="$XDSDocumentEntryHealthcareFacilityTypeCode">
                            <rim:ValueList>
                                <rim:Value>('Emergency Department')</rim:Value>
                            </rim:ValueList>
                          </rim:Slot>
                    </rim:AdhocQuery>
                </wsnt:TopicExpression>
            </wsnt:Filter>
            <wsnt:InitialTerminationTime>2008-05-31T00:00:00.00000Z</wsnt:InitialTerminationTime>
        </wsnt:Subscribe>
    </s:Body>
</s:Envelope>
```

The URN identifying the Topics dialect in this example is defined in the ***Topics and Filter Expressions*** section of this white paper.

5.4.2 Subscription Response

5.4.2.1 Trigger

This message is an immediate response to a Subscription Request, and it is sent from the Notification Broker to the Subscriber.

5.4.2.2 Message Semantics

The Subscription Request Response message shall comply with the requirements in the WS-BaseNotification standard, including the use of the appropriate SOAP Fault messages.

The subscription identifier is assigned by the Notification Broker as a Web Services end point, communicated in the response in the SOAP body in wsnt:SubscribeResponse/wsnt:SubscriptionReference. In order to modify the subscription, or to unsubscribe, these requests are sent to this SubscriptionReference end point.

5.4.2.3 Expected Actions

If the Subscriber had indicated a requested duration for the subscription, the Subscription Manager shall send the assigned duration for the subscription using the *wsnt:TerminationTime* element.

If the Subscriber had not indicated a requested duration for the subscription, the Subscription Manager may send an assigned duration for the subscription (if any), using the *wsnt:TerminationTime* element.

If the Subscribtion Manager sends an assigned duration for the subscription, the Subscriber shall associate the assigned duration with the accepted subscription request.

The Subscriber shall associate the accepted subscription request with the subscription reference address assigned by the Notification broker in order to be able to send modifications or cancellations for existing subscriptions.

5.4.2.4 Example

```
<?xml version="1.0" encoding="UTF-8"?>
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
    xmlns:a="http://www.w3.org/2005/08/addressing"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:wsnt="http://docs.oasis-open.org/wsn/b-2"
    xsi:schemaLocation="http://www.w3.org/2003/05/soap-envelope http://www.w3.org/2003/05/soap-
envelope http://www.w3.org/2005/08/addressing http://www.w3.org/2005/08/addressing/ws-addr.xsd
http://docs.oasis-open.org/wsn/b-2 http://docs.oasis-open.org/wsn/b-2.xsd">
    <s:Header>
        <a:Action>http://docs.oasis-open.org/wsn/bw-
2/NotificationProducer/SubscribeResponse</a:Action>
```

```
    </s:Header>
    <s:Body>
        <wsnt:SubscribeResponse>
            <!-- A WS endpoint, where modification and cancelation requests for this subscription 
must be sent -->
            <wsnt:SubscriptionReference>
                <a:Address>https://NotificationBrokerServer/Unique Subscription/382dcdc7-8e84-
9fdc-8443-48fd83bca938</a:Address>
            </wsnt:SubscriptionReference>
            <wsnt:TerminationTime>2008-05-31T00:00:00Z</wsnt:TerminationTime>
        </wsnt:SubscribeResponse>
    </s:Body>
</s:Envelope>
```

5.4.3 Subscription Renewal Request

5.4.3.1 Trigger

The need for a Notification Consumer to modify an existing subscription will cause the associated Subscriber to trigger a Subscription Renewal message.

5.4.3.2 Message Semantics

This message conveys a different termination time for an existing subscription. This message shall not be used to modify other subscription parameters of an existing subscription. This message shall not be used for subscriptions which are past their termination time.

5.4.3.3 Expected Actions

The subscriber shall send this message to the unique Web Services end point associated with the existing subscription.

The Notification Broker shall modify the corresponding subscription, and respond with a Subscription Renewal Response message. If the Notification Broker cannot modify the corresponding subscription, it shall return a ResourceUnknownFault or an UnacceptableTerminationTimeFault SOAP Fault message as approprite.

5.4.3.4 Example

```
<?xml version="1.0" encoding="UTF-8"?>
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
    xmlns:a="http://www.w3.org/2005/08/addressing"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:wsnt="http://docs.oasis-open.org/wsn/b-2"
    xsi:schemaLocation="http://www.w3.org/2003/05/soap-envelope http://www.w3.org/2003/05/soap-
envelope http://www.w3.org/2005/08/addressing http://www.w3.org/2005/08/addressing/ws-addr.xsd 
http://docs.oasis-open.org/wsn/b-2 http://docs.oasis-open.org/wsn/b-2.xsd">
    <s:Header>
        <a:Action>http://docs.oasis-open.org/wsn/bw-2/SubscriptionManager/RenewRequest</a:Action>
        <a:MessageID>382dcdc8-8e85-9fdd-8444-48fd83bca939</a:MessageID>
        <a:To>https://NotificationBrokerServer/Unique Subscription/382dcdc7-8e84-9fdc-8443-
48fd83bca938</a:To>
    </s:Header>
    <s:Body>
        <wsnt:Renew>
            <!-- This illustrates the use of the xs:duration XML Schema data type - the 
subscription ends after one day -->
            <wsnt:TerminationTime>P1D</wsnt:TerminationTime>
        </wsnt:Renew>
    </s:Body>
</s:Envelope>
```

The subscription modification example above shows the use of another way to specify subscription duration - using the XML Schema *duration* data type instead of a time stamp. The renewal response below uses the XML Schema *dateTime* datatype to indicate the precise moment in time when the subscription will expire.

5.4.4 Subscription Renewal Response

5.4.4.1 Trigger

This message is an immediate response to a Subscription Renewal message and is sent from the Notification Broker to the Subscriber.

5.4.4.2 Message Semantics

The message contains the new termination time of the subscription.

5.4.4.3 Expected Actions

The Subscriber shall update the termination time of the existing subscription with the new time as sent by the Notification Broker.

5.4.4.4 Example

```
<?xml version="1.0" encoding="UTF-8"?>
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
    xmlns:a="http://www.w3.org/2005/08/addressing"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:wsnt="http://docs.oasis-open.org/wsn/b-2"
    xsi:schemaLocation="http://www.w3.org/2003/05/soap-envelope http://www.w3.org/2003/05/soap-
envelope http://www.w3.org/2005/08/addressing http://www.w3.org/2005/08/addressing/ws-addr.xsd
http://docs.oasis-open.org/wsn/b-2 http://docs.oasis-open.org/wsn/b-2.xsd">
    <s:Header>
        <a:Action>http://docs.oasis-open.org/wsn/bw-
2/SubscriptionManager/RenewResponse</a:Action>
    </s:Header>
    <s:Body>
        <wsnt:RenewResponse>
            <wsnt:TerminationTime>2008-05-22T15:39:14-05:00</wsnt:TerminationTime>
        </wsnt:RenewResponse>
    </s:Body>
</s:Envelope>
```

5.4.5 Unsubscribe Request

5.4.5.1 Trigger

When a subscription is no longer needed, a Subscriber will trigger an Unsubscribe message.

5.4.5.2 Message Semantics

The message conveys the request to cancel an existing subscription.

5.4.5.3 Expected Actions

The subscriber shall send this message to the unique Web Services end point associated with the existing subscription.

The Notification Broker shall cancel the corresponding subscription, and respond with an Unsubscribe Response message. In the case when for whatever reason the subscription cannot be canceled, the Notification Broker shall respond with a ResourceUnknownFault or an UnableToDestroySubscriptionFault SOAP Fault message as appropriate.

5.4.5.4 Example

```
<?xml version="1.0" encoding="UTF-8"?>
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
    xmlns:a="http://www.w3.org/2005/08/addressing"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:wsnt="http://docs.oasis-open.org/wsn/b-2"
    xsi:schemaLocation="http://www.w3.org/2003/05/soap-envelope http://www.w3.org/2003/05/soap-
envelope http://www.w3.org/2005/08/addressing http://www.w3.org/2005/08/addressing/ws-addr.xsd
http://docs.oasis-open.org/wsn/b-2 http://docs.oasis-open.org/wsn/b-2.xsd">
    <s:Header>
        <a:Action>http://docs.oasis-open.org/wsn/bw-
2/SubscriptionManager/UnsubscribeRequest</a:Action>
        <a:MessageID>382dcdc9-8e86-9fde-8445-48fd83bca93a</a:MessageID>
        <a:To>https://NotificationBrokerServer/Unique Subscription/382dcdc7-8e84-9fdc-8443-
48fd83bca938</a:To>
    </s:Header>
    <s:Body>
        <wsnt:Unsubscribe/>
    </s:Body>
</s:Envelope>
```

5.4.6 Unsubscribe Response

5.4.6.1 Trigger

This message is an immediate response to an Unsubscribe message, and it is sent from the Notification Broker to the Subscriber.

5.4.6.2 Message Semantics

This message indicates that an Unsubscribe message was successfully processed.

5.4.6.3 Expected Actions

The Notification Broker shall cancel the corresponding subscription.

The Subscriber shall mark the corresponding subscription as successfully terminated.

5.4.6.4 Example

```
<?xml version="1.0" encoding="UTF-8"?>
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
    xmlns:a="http://www.w3.org/2005/08/addressing"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:wsnt="http://docs.oasis-open.org/wsn/b-2"
    xsi:schemaLocation="http://www.w3.org/2003/05/soap-envelope http://www.w3.org/2003/05/soap-
envelope http://www.w3.org/2005/08/addressing http://www.w3.org/2005/08/addressing/ws-addr.xsd
http://docs.oasis-open.org/wsn/b-2 http://docs.oasis-open.org/wsn/b-2.xsd">
```

```
    <s:Header>
        <a:Action>http://docs.oasis-open.org/wsn/bw-
2/SubscriptionManager/UnsubscribeRequest</a:Action>
    </s:Header>
    <s:Body>
        <wsnt:UnsubscribeResponse/>
    </s:Body>
</s:Envelope>
```

5.5 Security Considerations

Relevant XDS Affinity Domain Security background is discussed in the Register Document transaction (see ITI TF-2: 3.14.5.1).

5.5.1 Audit Record Considerations

The Subscribe Transaction is a Query Information event as defined in table ITI TF-2:3.20.6-1. The Actors involved shall record audit events according to the following:

5.5.1.1 Subscriber audit message:

	Field Name	Opt	Value Constraints
Event **AuditMessage/ EventIdentification**	EventID	M	EV(110112, DCM, “Query”)
	EventActionCode	M	“C” (Create) for Subscription message exchange “U” (Update) for Renewal message exchange “D” (Delete) for Unsubscribe message exchange
	EventDateTime	*M*	*not specialized*
	EventOutcomeIndicator	*M*	*not specialized*
	EventTypeCode	M	EV(“ITI-XX”, “IHE Transactions”, “Subscribe for Registry Stored Query”)
Source (Subscriber) (1)			
Human Requestor (0..n)			
Destination (Notification Broker) (1)			
Audit Source (Subscriber) (1)			
Patient (0..1)			
Query Parameters(1)			

Where:

Source **AuditMessage/ ActiveParticipant**	UserID	C	When WS-Addressing is used: <ReplyTo/>
	AlternativeUserID	M	the process ID as used within the local operating system in the local system logs.
	UserName	*U*	*not specialized*
	UserIsRequestor	M	“true”
	RoleIDCode	M	EV(110153, DCM, “Source”)
	NetworkAccessPointTypeCode	M	“1” for machine (DNS) name, “2” for IP address
	NetworkAccessPointID	M	The machine name or IP address, as specified in RFC 3881.
Human Requestor (if known) **AuditMessage/ ActiveParticipant**	UserID	M	Identity of the human that initiated the transaction.
	AlternativeUserID	*U*	*not specialized*
	UserName	*U*	*not specialized*
	UserIsRequestor	M	“true”
	RoleIDCode	U	Access Control role(s) the user holds that allows this transaction.
	NetworkAccessPointTypeCode	*NA*	
	NetworkAccessPointID	*NA*	

Destination **AuditMessage/ ActiveParticipant**	UserID	M	SOAP endpoint URI.
	AlternativeUserID	*U*	*not specialized*
	UserName	*U*	*not specialized*
	UserIsRequestor	M	“false”
	RoleIDCode	M	EV(110152, DCM, “Destination”)
	NetworkAccessPointTypeCode	M	“1” for machine (DNS) name, “2” for IP address
	NetworkAccessPointID	M	The machine name or IP address, as specified in RFC 3881.

Audit Source **AuditMessage/ AuditSourceIdentification**	*AuditSourceID*	*U*	*Not specialized.*
	AuditEnterpriseSiteID	*U*	*not specialized*
	AuditSourceTypeCode	*U*	*not specialized*

Patient (if known) **(AudittMessage/ ParticipantObjectIdentification)**	ParticipantObjectTypeCode	M	“1” (Person)
	ParticipantObjectTypeCodeRole	M	“1” (Patient)
	ParticipantObjectDataLifeCycle	*U*	*not specialized*
	ParticipantObjectIDTypeCode	M	EV(2, RFC-3881, “Patient Number”)
	ParticipantObjectSensitivity	*U*	*not specialized*
	ParticipantObjectID	M	The patient ID in HL7 CX format.
	ParticipantObjectName	*U*	*not specialized*
	ParticipantObjectQuery	*U*	*not specialized*
	ParticipantObjectDetail	*U*	*not specialized*
	ParticipantObjectTypeCode	M	“2” (system object)
	ParticipantObjectTypeCodeRole	M	“24” (query)
	ParticipantObjectDataLifeCycle	*U*	*not specialized*
	ParticipantObjectIDTypeCode	M	EV(“ITI-XX”, “IHE Transactions”, “Subscribe for Registry Stored Query”)
	ParticipantObjectSensitivity	*U*	*not specialized*
	ParticipantObjectID	*U*	*not specialized*

	ParticipantObjectName	C	If known the value of <ihe:HomeCommunityId/>
	ParticipantObjectQuery	M	the value of <wsnt:Filter> element, base64 encoded.
	ParticipantObjectDetail	*U*	*not specialized*

5.5.1.2 Notification Broker audit message:

	Field Name	**Opt**	**Value Constraints**
Event **AuditMessage/ EventIdentification**	EventID	M	EV(110112, DCM, “Query”)
	EventActionCode	M	“C” (Create) for Subscription message exchange “U” (Update) for Renewal message exchange “D” (Delete) for Unsubscribe message exchange
	EventDateTime	*M*	*not specialized*
	EventOutcomeIndicator	*M*	*not specialized*
	EventTypeCode	M	EV(“ITI-XX”, “IHE Transactions”, “Subscribe for Registry Stored Query”)
Source (Subscriber) (1)			
Destination (Notification Broker) (1)			
Audit Source (Notification Broker) (1)			
Patient (0..1)			
Query Parameters(1)			

Where:

Source **AuditMessage/ ActiveParticipant**	UserID	C	When WS-Addressing is used: <ReplyTo/>
	AlternativeUserID	*U*	*not specialized*
	UserName	*U*	*not specialized*
	UserIsRequestor	M	“true”
	RoleIDCode	M	EV(110153, DCM, “Source”)
	NetworkAccessPointTypeCode	M	“1” for machine (DNS) name, “2” for IP address
	NetworkAccessPointID	M	The machine name or IP address, as specified in RFC 3881.

Destination **AuditMessage/ ActiveParticipant**	UserID	M	SOAP endpoint URI.
	AlternativeUserID	M	the process ID as used within the local operating system in the local system logs.
	UserName	*U*	*not specialized*
	UserIsRequestor	M	“false”
	RoleIDCode	M	EV(110152, DCM, “Destination”)
	NetworkAccessPointTypeCode	M	“1” for machine (DNS) name, “2” for IP address
	NetworkAccessPointID	M	The machine name or IP address, as specified in RFC 3881.

Audit Source **AuditMessage/ AuditSourceIdentification**	*AuditSourceID*	*U*	*Not specialized.*
	AuditEnterpriseSiteID	*U*	*not specialized*
	AuditSourceTypeCode	*U*	*not specialized*

	ParticipantObjectTypeCode	M	“1” (Person)

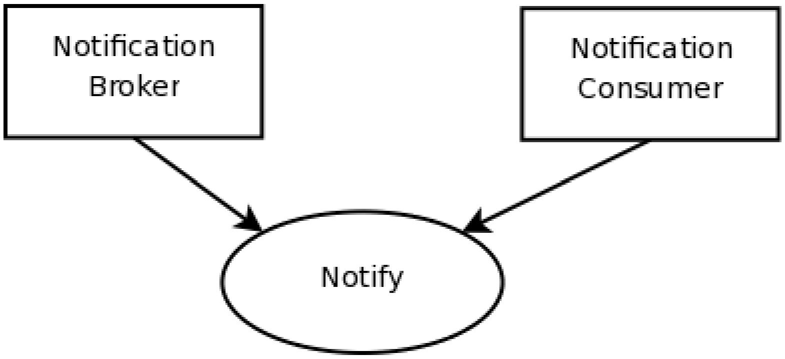
	ParticipantObjectTypeCodeRole	M	“1” (Patient)
	ParticipantObjectDataLifeCycle	*U*	*not specialized*
	ParticipantObjectIDTypeCode	M	EV(2, RFC-3881, “Patient Number”)
	ParticipantObjectSensitivity	*U*	*not specialized*
	ParticipantObjectID	M	The patient ID in HL7 CX format.
	ParticipantObjectName	*U*	*not specialized*
	ParticipantObjectQuery	*U*	*not specialized*
	ParticipantObjectDetail	*U*	*not specialized*
Query Parameters **(AudittMessage/ ParticipantObjectIdentification)**	ParticipantObjectTypeCode	M	“2” (system object)
	ParticipantObjectTypeCodeRole	M	“24” (query)
	ParticipantObjectDataLifeCycle	*U*	*not specialized*
	ParticipantObjectIDTypeCode	M	EV(“ITI-XX”, “IHE Transactions”, “Notification for Registry Stored Query”)
	ParticipantObjectSensitivity	*U*	*not specialized*
	ParticipantObjectID	*U*	*not specialized*
	ParticipantObjectQuery	M	the value of <wsnt:Filter> element, base64 encoded.
	ParticipantObjectDetail	*U*	*not specialized*

6 Notify

6.1 Scope

This transaction delivers a notification from the Notification Broker to the Notification Consumer about an event which matches an existing subscription.

6.2 Use Case Roles



Actor: Notification Broker

Role: Sends notifications to subscribed Notification Consumers based on events occurring in an XDS.b Document Registry

Actor: Notification Consumer

Role: Receives and processes notifications about events matching a set of filter expressions.

6.3 Referenced Standards

OASIS Web Services Notification Family of Standards

WS-BaseNotification 1.3 OASIS Standard

WS-BrokeredNotification 1.3 OASIS Standard

WS-Topics 1.3 OASIS Standard

IHE ITI TF – XDS.b Supplement 2:3.43.4.2.2

6.4 Interaction Diagram

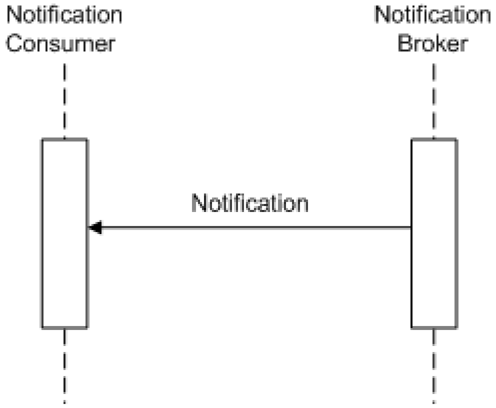


Figure 6-1 Notify Sequence

6.4.1 Notification

6.4.1.1 Trigger

When an event occurs, where the topics of the event match the filter requirements of one or more existing subscriptions, the Notification Broker will trigger a Notification message to the corresponding Notification Consumer.

6.4.1.2 Message Semantics

The events, for which a Notification Consumer within an XDS Affinity Domain subscribes, are registrations at the XDS.b Registry. While this paper does not specify how the Notification Broker becomes aware of the registrations, there are several possible approaches:

- Combined actor: XDS.b Registry and Notification Broker
- Forwarded transaction: The XDS.b Registry can be enhanced to forward all Register transactions to the Notification Broker. ebXML-RS 3.0 supports this registry functionality.
- Proxy: The Notification Broker can sit in front of the registry and intercept all Register transactions sent by Document Repositories, before forwarding them to the Registry

Depending on the XDS submission, which triggered the notification, there may be one or more documents whose metadata matches the filter conditions of each corresponding subscription. The notification message contains the document unique ID of each document relevant to the particular Notification Consumer.

6.4.1.3 Expected Actions

The Notification Consumer needs to convey the notification information to other systems and/or users. An XDS.b Document Consumer can then use the information to obtain the metadata about the document(s) from the XDS.b registry as necessary for the relevant workflows. This white paper does not specify how the Document Unique ID is conveyed from the Notification Consumer to an XDS.b Document Consumer. One possible solution is to have a combined Notification Consumer/Document Consumer actor, or even a Subscriber/Notification Consumer/Document Consumer actor.

The Notification Broker may send the filter conditions of the subscription, and/or the address of the Document Source where the document submission originated. Both of these options increase certain security risks, their use should be determined by local policy to security and confidentiality.

This white paper defines the following structure for conveying a Notification:

- A NotificationContent element, which may repeat, and which contains:
    - A required DocumentUniqueId, representing the XDSDocumentEntry.uniqueId attribute in the XDS metadata
    - An optional HomeCommunityId, representing the home attribute of the Identifiable class in the ebXML-RS RIM

6.4.1.4 Example

```
<?xml version="1.0" encoding="UTF-8"?>
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
    xmlns:a="http://www.w3.org/2005/08/addressing"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:wsnt="http://docs.oasis-open.org/wsn/b-2"
    xmlns:ihe="urn:ihe:iti:pub-sub:2008"
    xmlns:rim="urn:oasis:names:tc:ebxml-regrep:xsd:rim:3.0"
    xsi:schemaLocation="http://www.w3.org/2003/05/soap-envelope http://www.w3.org/2003/05/soap-
envelope http://www.w3.org/2005/08/addressing http://www.w3.org/2005/08/addressing/ws-addr.xsd
http://docs.oasis-open.org/wsn/b-2 http://docs.oasis-open.org/wsn/b-2.xsd urn:ihe:iti:pub-
sub:2008 ../schema/IHE/XDS.b_PublishSubscribe.xsd urn:oasis:names:tc:ebxml-regrep:xsd:rim:3.0
../schema/ebRS/rim.xsd">
    <s:Header>
        <a:Action>http://docs.oasis-open.org/wsn/bw-2/NotificationConsumer/Notify</a:Action>
        <a:MessageID>382dcdca-8e87-9fdf-8446-48fd83bca93b</a:MessageID>
        <a:To>https://NotificationConsumerServer/xdsBnotification</a:To>
    </s:Header>
    <s:Body>
        <wsnt:Notify>
            <wsnt:NotificationMessage>
                <wsnt:SubscriptionReference>
                    <a:Address>https://NotificationBrokerServer/Unique Subscription/382dcdc7-
8e84-9fdc-8443-48fd83bca938</a:Address>
                </wsnt:SubscriptionReference>
                <wsnt:Topic Dialect="urn:ihe:iti:xds-b:pubsub:2008">
                    <rim:AdhocQuery id="urn:uuid:14d4debf-8f97-4251-9a74-a90016b0af0d">
                        <rim:Slot name="$XDSDocumentEntryPatientId">
                            <rim:ValueList>

<rim:Value>'st3498702^^^&amp;1.3.6.1.4.1.21367.2005.3.7&amp;ISO'</rim:Value>
                            </rim:ValueList>
                        </rim:Slot>
                        <rim:Slot name="$XDSDocumentEntryHealthcareFacilityTypeCode">
                            <rim:ValueList>
                                <rim:Value>('Emergency Department')</rim:Value>
                            </rim:ValueList>
                        </rim:Slot>
                    </rim:AdhocQuery>
                </wsnt:Topic>
                <wsnt:ProducerReference>
                    <a:Address>https://DocumentSource</a:Address>
                </wsnt:ProducerReference>
                <wsnt:Message>
                    <ihe:NotificationContent>
                        <ihe:DocumentUniqueId>1.3.2458.2423.456.5.3.5</ihe:DocumentUniqueId>
                        <ihe:HomeCommunityId>urn:oid:1.3.2458.2423.458.3.2</ihe:HomeCommunityId>
                    </ihe:NotificationContent>
                </wsnt:Message>
            </wsnt:NotificationMessage>
        </wsnt:Notify>
    </s:Body>
</s:Envelope>
```

6.5 Security Considerations

Relevant XDS Affinity Domain Security background is discussed in the Register Document transaction (see ITI TF-2: 3.14.5.1).

6.5.1 Audit Record Considerations

The Notification Transaction is an Export event, as defined in table 3.20.6-1. The Actors involved in the transaction shall create audit data in conformance with DICOM (Supp 95) “Data Export”/”Data Import”, with the following exceptions.

6.5.1.1 Notification Consumer audit message:

	Field Name	Opt	Value Constraints
Event **AuditMessage/** **EventIdentification**	EventID	M	EV(110107, DCM, “Import”)
	EventActionCode	M	“C” (Create)
	EventDateTime	*M*	*not specialized*
	EventOutcomeIndicator	*M*	*not specialized*
	EventTypeCode	M	EV(“ITI-XX”, “IHE Transactions”, “Notification for XDS”)
Source (Notification Broker) (1)			
Destination (Notification Consumer) (1)			
Human Requestor (0..n)			
Audit Source (Notification Consumer) (1)			
Patient (0..1)			
Document (1..n)			

Where:

Source **AuditMessage/** **ActiveParticipant**	UserID	M	When WS-Addressing is used: <From/>
	AlternativeUserID	*U*	*not specialized*
	UserName	*U*	*not specialized*
	UserIsRequestor	M	“false”
	RoleIDCode	M	EV(110153, DCM, “Source”)
	NetworkAccessPointTypeCode	M	“1” for machine (DNS) name, “2” for IP address
	NetworkAccessPointID	M	The machine name or IP address, as specified in RFC 3881.

Destination **AuditMessage/ ActiveParticipant**	UserID	C	When WS-Addressing is used: <ReplyTo/>
	AlternativeUserID	M	the process ID as used within the local operating system in the local system logs.
	UserName	*U*	*not specialized*
	UserIsRequestor	M	“true”
	RoleIDCode	M	EV(110152, DCM, “Destination”)
	NetworkAccessPointTypeCode	M	“1” for machine (DNS) name, “2” for IP address
	NetworkAccessPointID	M	The machine name or IP address, as specified in RFC 3881.
Human Requestor (if known) **AuditMessage/ ActiveParticipant**	UserID	M	Identity of the human that initiated the transaction.
	AlternativeUserID	*U*	*not specialized*
	UserName	*U*	*not specialized*
	UserIsRequestor	M	“true”
	RoleIDCode	U	Access Control role(s) the user holds that allows this transaction.
	NetworkAccessPointTypeCode	*NA*	
	NetworkAccessPointID	*NA*	

Audit Source **AuditMessage/ AuditSourceIdentification**	*AuditSourceID*	*U*	*Not specialized.*
	AuditEnterpriseSiteID	*U*	*not specialized*
	AuditSourceTypeCode	*U*	*not specialized*

Patient (if-known) **(AudittMessage/ ParticipantObjectIdentification)**	ParticipantObjectTypeCode	M	“1” (Person)
	ParticipantObjectTypeCodeRole	M	“1” (Patient)
	ParticipantObjectDataLifeCycle	*U*	*not specialized*
	ParticipantObjectIDTypeCode	M	EV(2, RFC-3881, “Patient Number”)
	ParticipantObjectSensitivity	*U*	*not specialized*
	ParticipantObjectID	M	The patient ID in HL7 CX format.
	ParticipantObjectName	*U*	*not specialized*
	ParticipantObjectQuery	*U*	*not specialized*
	ParticipantObjectDetail	*U*	*not specialized*
Document **(AudittMessage/ ParticipantObjectIdentification)**	ParticipantObjectTypeCode	M	“2” (System)
	ParticipantObjectTypeCodeRole	M	“3” (report)
	ParticipantObjectDataLifeCycle	*U*	*not specialized*
	ParticipantObjectIDTypeCode	M	EV(9, RFC-3881, “Report Number”)
	ParticipantObjectSensitivity	*U*	*not specialized*
	ParticipantObjectID	M	The value of <ihe:DocumentUniqueId/>
	ParticipantObjectName	C	If known the value of <ihe:HomeCommunityId/>
	ParticipantObjectQuery	*U*	*not specialized*
	ParticipantObjectDetail	*U*	*not specialized*

3.43.6.1.2 Notification Broker audit message:

	Field Name	Opt	Value Constraints
	EventID	M	EV(110106, DCM, “Export”)

	EventActionCode	M	"R" (Read)
	EventDateTime	*M*	*not specialized*
	EventOutcomeIndicator	*M*	*not specialized*
	EventTypeCode	M	EV("ITI-XX", "IHE Transactions", "Notification for XDS")
Source (Notification Broker) (1)			
Destination (Notification Consumer) (1)			
Audit Source (Notification Broker) (1)			
Document (1..n)			

Where:

Source **AuditMessage/ ActiveParticipant**	UserID	M	When WS-Addressing is used: <From/>
	AlternativeUserID	M	the process ID as used within the local operating system in the local system logs.
	UserName	*U*	*not specialized*
	UserIsRequestor	M	"false"
	RoleIDCode	M	EV(110153, DCM, "Source")
	NetworkAccessPointTypeCode	M	"1" for machine (DNS) name, "2" for IP address
	NetworkAccessPointID	M	The machine name or IP address, as specified in RFC 3881.

Destination **AuditMessage/ ActiveParticipant**	UserID	C	When WS-Addressing is used: <ReplyTo/>
	AlternativeUserID	*U*	*not specialized*
	UserName	*U*	*not specialized*
	UserIsRequestor	M	"true"
	RoleIDCode	M	EV(110152, DCM, "Destination")
	NetworkAccessPointTypeCode	M	"1" for machine (DNS) name, "2" for IP address
	NetworkAccessPointID	M	The machine name or IP address, as specified in RFC 3881.

Audit Source **AuditMessage/ AuditSourceIdentification**	*AuditSourceID*	*U*	*Not specialized.*
	AuditEnterpriseSiteID	*U*	*not specialized*
	AuditSourceTypeCode	*U*	*not specialized*

Document URI **(AudittMessage/ ParticipantObjectIdentification)**	ParticipantObjectTypeCode	M	"2" (System)
	ParticipantObjectTypeCodeRole	M	"3" (report)
	ParticipantObjectDataLifeCycle	*U*	*not specialized*
	ParticipantObjectIDTypeCode	M	EV(9, RFC-3881, "Report Number")
	ParticipantObjectSensitivity	*U*	*not specialized*
	ParticipantObjectID	M	The value of <ihe:DocumentUniqueId/>
	ParticipantObjectName	C	If known the value of <ihe:HomeCommunityId/>
	ParticipantObjectQuery	*U*	*not specialized*
	ParticipantObjectDetail	*U*	*not specialized*

Subscription Topics and Filter Expressions

Subscription topics are the concepts used for describing under what conditions the subscriber would like to receive a notification. Filter expressions bind topics to a particular value (or set of values) and combine them into expressions describing the actual conditions. Topics are context dependent, and as such are closely related to the information for which a subscription is requested. At the same time, the ability of the Subscription Manager to quickly process and publish the notifications may depend on the ease of computability of the topics.

7 Topics for XDS.b Publish/Subscribe

The XDS.b profile specifies the particular metadata which is registered about the documents being shared. The Stored Query transaction (ITI-18) uses a subset of the metadata to build a list of queries available to a XDS Document Consumer to search for documents with specific characteristics. The list of queries is in section 3.18.4.1.2.3.7 currently in the TF Supplement for trial implementation. This paper chooses to restrict the list of topics to the metadata represented by query parameters, and restrict the semantics of filter expressions to the semantics of the corresponding stored query.

7.1 Building Filter Expressions

The filter expressions used in this white paper use the syntax of the Stored Query transaction. In order to identify this syntax, the following URN is hereby defined as the "topics dialect":

```
urn:ihe:iti:xds-b:pubsub:2008
```

A good understanding of the Stored Query transaction and the XDS.b metadata is necessary to understand how the filter expressions work. In general, for each submission set registered, the filter expression will yield a match, if the corresponding query would return the same object(s) from an XDS.b registry, which contained only this submission set. For example, the stored query

```
<rim:AdhocQuery id="urn:uuid:14d4debf-8f97-4251-9a74-a90016b0af0d"
xmlns:rim="urn:oasis:names:tc:ebxml-regrep:xsd:rim:3.0">
    <rim:Slot name="$XDSDocumentEntryPatientId">
        <rim:ValueList>
            <rim:Value>'st3498702^^^&1.3.6.1.4.1.21367.2005.3.7&ISO'</rim:Value>
        </rim:ValueList>
    </rim:Slot>
    <rim:Slot name="$XDSDocumentEntryEventCodeList">
        <rim:ValueList>
            <rim:Value>('44950' '44955' '44960' '44970' '44979')</rim:Value>
        </rim:ValueList>
    </rim:Slot>
</rim:AdhocQuery>
```

will return all document entries for patient with ID st3498702 (assigned by an authority identified by the OID 1.3.6.1.4.1.21367.2005.3.7) where the event code metadata contains at least one of the codes listed (in this case CPT codes for various appendectomies). When used as a filter expression, the same structure will yield a match against a document entry in an XDS.b registry submission, where the document entry is for patient with ID st3498702 and the event code is "44970". The following snippet shows an example of such a submission:

```
<lcm:SubmitObjectsRequest>
```

```
    <rim:RegistryObjectList>
        <rim:ExtrinsicObject id="Document01" mimeType="text/xml" objectType="urn:uuid:7edca82f-
054d-47f2-a032-9b2a5b5186c1">
...
            <rim:Classification classificationScheme="urn:uuid:2c6b8cb7-8b2a-4051-b291-
b1ae6a575ef4" classifiedObject="Document01" nodeRepresentation="44950">
                <rim:Name>
                    <rim:LocalizedString value="Appendectomy"/>
                </rim:Name>
                <rim:Slot name="codingScheme">
                    <rim:ValueList>
                        <rim:Value>CPT codes</rim:Value>
                    </rim:ValueList>
                </rim:Slot>
            </rim:Classification>
...
            <rim:ExternalIdentifier id="ei01" registryObject="Document01"
identificationScheme="urn:uuid:58a6f841-87b3-4a3e-92fd-a8ffeff98427"
value="'st3498702^^^&amp;1.3.6.1.4.1.21367.2005.3.7&amp;ISO'">
                  <rim:Name>
                    <rim:LocalizedString value="XDSDocumentEntry.patientId"/>
                </rim:Name>
            </rim:ExternalIdentifier>
...
        <rim:ExtrinsicObject>
    </rim:RegistryObjectList>
```

When a filter expression is constructed, the whole stored query expression (as shown above) is included directly in the Subscribe Request message:

```
<?xml version="1.0" encoding="UTF-8"?>
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
    xmlns:a="http://www.w3.org/2005/08/addressing"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:wsnt="http://docs.oasis-open.org/wsn/b-2"
    xmlns:rim="urn:oasis:names:tc:ebxml-regrep:xsd:rim:3.0"
    xsi:schemaLocation="http://www.w3.org/2003/05/soap-envelope http://www.w3.org/2003/05/soap-
envelope http://www.w3.org/2005/08/addressing http://www.w3.org/2005/08/addressing/ws-addr.xsd
http://docs.oasis-open.org/wsn/b-2 http://docs.oasis-open.org/wsn/b-2.xsd
urn:oasis:names:tc:ebxml-regrep:xsd:rim:3.0 ../schema/ebRS/rim.xsd">
    <s:Header>
        <a:Action>http://docs.oasis-open.org/wsn/bw-
2/NotificationProducer/SubscribeRequest</a:Action>
        <a:MessageID>382dcdc7-8e84-9fdc-8443-48fd83bca938</a:MessageID>
    </s:Header>
    <s:Body>
        <wsnt:Subscribe>
            <!-- The Consumer on whose behalf the subscription is requested - the address where
the notification is to be sent -->
            <wsnt:ConsumerReference>
                <a:Address>https://NotificationConsumerServer/xdsBnotification</a:Address>
            </wsnt:ConsumerReference>
            <wsnt:Filter>
                <wsnt:TopicExpression Dialect="urn:ihe:iti:xds-b:pubsub:2008">
                    <rim:AdhocQuery id="urn:uuid:14d4debf-8f97-4251-9a74-a90016b0af0d">
                        <rim:Slot name="$XDSDocumentEntryPatientId">
                            <rim:ValueList>

<rim:Value>'st3498702^^^&1.3.6.1.4.1.21367.2005.3.7&ISO'</rim:Value>
                            </rim:ValueList>
                        </rim:Slot>
                        <rim:Slot name="$XDSDocumentEntryEventCodeList">
                            <rim:ValueList>
                                <rim:Value>('44950' '44955' '44960' '44970' '44979')</rim:Value>
                            </rim:ValueList>
                        </rim:Slot>
                    </rim:AdhocQuery>
                </wsnt:TopicExpression>
            </wsnt:Filter>
            <wsnt:InitialTerminationTime>2008-07-31T00:00:00.00000Z</wsnt:InitialTerminationTime>
        </wsnt:Subscribe>
    </s:Body>
</s:Envelope>
```

How the filter expression is evaluated, and how the matching against the existing subscriptions is done, is out of scope of this white paper. It is expected that such implementation detail will allow vendors to differentiate themselves in the marketplace.

It is important to note that not all stored queries, and not all parameters defined for the stored queries, are suitable for filter expressions. The following stored queries and associated parameters can be used in subscription requests:

7.1.1 Subscriptions based on FindDocuments query

Filter expressions based on the FindDocuments stored query will yield a match any time a document entry for a particular patient, and with the appropriate metadata restrictions, is part of a registry submission.

1. **$XDSDocumentEntryPatientId**: this required parameter contains the patient ID for which a document entry is being registered in the XDS.b registry
2. **$XDSDocumentEntryClassCode** and **$XDSDocumentEntryClassCodeScheme**: these parameters match against the **XDSDocumentEntry.classCode** metadata elements in a given registry submission
3. **$XDSDocumentEntryPracticeSettingCode** and **$XDSDocumentEntryPracticeSettingCodeScheme**: these parameters match against the **XDSDocumentEntry.practiceSettingCode** metadata elements in a given registry submission
4. **$XDSDocumentEntryHealthcareFacilityTypeCode** and **$XDSDocumentEntryHealthcareFacilityTypeCodeScheme**: these parameters match against the **XDSDocumentEntry.healthcareFacilityTypeCode** metadata elements in a given registry submission
5. **$XDSDocumentEntryEventCodeList** and **$XDSDocumentEntryEventCodeListScheme**: these parameters match against the **XDSDocumentEntry.eventCodeList** metadata elements in a given registry submission
6. **$XDSDocumentEntryConfidentialityCode** and **$XDSDocumentEntryConfidentialityCodeScheme**: these parameters match against the **XDSDocumentEntry.confidentialityCode** metadata elements in a given registry submission
7. **$XDSDocumentEntryFormatCode**: this parameter matches against the **XDSDocumentEntry.formatCode** metadata elements in a given registry submission

7.1.2 Subscriptions based on FindSubmissionSets query

Filter expressions based on the FindSubmissionSets stored query can be used to subscribe for events, where the source ID, author, or content type of submission sets about a particular patient are the topics of the subscription.

1. **$XDSSubmissionSetPatientId**: this required parameter contains the patient ID for which a submission set is being registered in the XDS.b registry
2. **$XDSSubmissionSetSourceId**: this parameter matches against the **XDSSubmissionSet.sourceId** metadata elements in a given registry submission
3. **$XDSSubmissionSetAuthorPerson**: this parameter matches against the **XDSSubmissionSet.authorPerson** metadata elements in a given registry submission
4. **$XDSSubmissionSetContentType**: this parameter matches against the **XDSSubmissionSet.contentTypeCode** metadata elements in a given registry submission

7.1.3 Subscriptions based on FindFolders query

Filter expressions based on the FindFolders stored query can be used to subscribe for patient-specific events related to Folders with a particular code or set of codes.

1. **$XDSFolderPatientId**: this required parameter contains the patient ID associated with the folder, for which a submission set is being registered in the XDS.b registry
2. **$XDSFolderCodeList** and **$XDSFolderCodeListScheme**: these parameters match against the **XDSFolder.codeList** metadata elements in a given registry submission

Note to Reader: Document-centric queries are not included in this list. Any use cases supporting inclusion of these queries in the list are welcome.

7.2 Combining topics in filter expressions

A filter expression is equivalent to a specific stored query with certain parameters. Topics expressed as query parameters and used in the expressions must satisfy the same requirements as a corresponding Stored Query:

- the values for all specified topics must match (AND all different topics)
- at least one of the values of multi-valued parameters must match (OR the values in a multi-valued query parameter)

Since the current catalog of queries for Stored Query always has either the patient ID or the document ID as a required parameter, subscriptions are only allowed on a per-patient or per-document basis.

Note that there are various other ways to specify topics, and filter expressions. One of the most common is the use of XPath expressions as the filter expression. Future extensions to this white paper and future profiles can and should investigate the feasibility of this and other alternative approaches.