



---

## **Endpoint Location Service**

### **Architecture**

Version 1.2 — 30 June 2009

---

**National E-Health Transition Authority Ltd**

Level 25

56 Pitt Street

Sydney, NSW, 2000

Australia.

[www.nehta.gov.au](http://www.nehta.gov.au)**Disclaimer**

NEHTA makes the information and other material (“Information”) in this document available in good faith but without any representation or warranty as to its accuracy or completeness. NEHTA cannot accept any responsibility for the consequences of any use of the Information. As the Information is of a general nature only, it is up to any person using or relying on the Information to ensure that it is accurate, complete and suitable for the circumstances of its use.

**Document Control**

This document is maintained in electronic form. The current revision of this document is located on the NEHTA Web site and is uncontrolled in printed form. It is the responsibility of the user to verify that this copy is of the latest revision.

**Copyright © 2009, NEHTA.**

This document contains information which is protected by copyright. All Rights Reserved. No part of this work may be reproduced or used in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems—without the permission of NEHTA. All copies of this document must include the copyright and other information contained on this page.

# Table of contents

<b>Table of contents</b> .....	<b>iii</b>
<b>Document information</b> .....	<b>v</b>
Change history .....	v
<b>1 Executive Overview</b> .....	<b>1</b>
1.1 Purpose .....	1
1.2 Implementation .....	1
1.3 Document Exchange Interactions .....	1
1.4 Deployment .....	1
1.5 Usage .....	2
<b>2 Preface</b> .....	<b>3</b>
2.1 Document Purpose .....	3
2.2 Intended Audience .....	3
2.3 Definitions, Acronyms, and Abbreviations .....	3
2.4 References and Related Documents .....	5
<b>3 Introduction</b> .....	<b>6</b>
3.1 Solution Overview .....	6
3.2 Solution Scope .....	6
3.2.1 Solutions Out of Scope .....	6
3.3 Solution Goals and Objective .....	6
3.4 Assumptions and Dependencies .....	7
<b>4 Solution States</b> .....	<b>8</b>
4.1 Today .....	8
4.1.1 Healthcare Provider Directories .....	8
4.2 Tomorrow .....	8
4.2.1 Service Directory Providers .....	8
4.3 Future .....	9
4.3.1 Healthcare Service Providers .....	9
4.3.2 Non-Healthcare Service Providers .....	9
4.3.3 Deployment .....	9
4.3.4 Non-Web Services .....	9
<b>5 Business Perspective</b> .....	<b>11</b>
5.1 Requirements .....	11
5.2 Document Exchange Scenarios .....	11
5.2.1 Retrieve .....	11
5.2.2 Notify and Retrieve .....	12
5.2.3 Deliver .....	13
5.2.4 Deliver and Notify .....	13
5.2.5 Acknowledge .....	14
5.2.6 Caching of Interactions .....	14
5.3 ELS Community .....	17
<b>6 Information Perspective</b> .....	<b>20</b>
6.1 Requirements .....	20
6.2 Services and Provider Directories .....	20
6.3 Interaction Data Structure .....	20
6.3.1 Interaction .....	21
6.3.2 Service category Identifiers .....	22
6.3.3 Target Identifiers .....	22
6.3.4 Service Provider .....	22
6.3.5 Service Interface .....	22

6.3.6	Service Endpoint .....	23
6.4	Qualified Certificate Reference .....	23
6.4.1	Structure .....	23
6.4.2	Cardinality, Organizations, and Usage .....	24
6.5	Interaction Request .....	24
6.5.1	serviceInterface Identifier .....	24
6.6	Summary of Information Types .....	24
<b>7</b>	<b>Technical Perspective .....</b>	<b>26</b>
7.1	Requirements.....	26
7.2	Lookup Package .....	26
7.2.1	Operation listInteractions .....	26
7.2.2	Operation validateInteraction.....	26
7.2.3	Error Code .....	27
7.2.4	Security Considerations .....	27
7.3	Publish Package.....	27
7.3.1	Operation addInteraction .....	28
7.3.2	Operation removeInteraction .....	28
7.3.3	Return Codes .....	28
7.3.4	Error Code .....	29
7.3.5	Security Considerations .....	29
7.3.6	Non-standardized Operations.....	30
<b>8</b>	<b>Enabler Dependencies .....</b>	<b>31</b>
8.1	UHI .....	31
8.1.1	ELS Bootstrap Reference .....	31
8.2	NASH.....	31
8.2.1	Single X.509 Certificate per HPIO .....	31
8.2.2	Multiple X.509 Certificates per HPI.....	31
8.2.3	Trust Based on Certificates .....	31
	<b>Appendix A References.....</b>	<b>33</b>

# Document information

## Change history

---

Version	Date	Comments
1.0 draft	2008-09-01	Draft for review
1.1	2008-12-01	Final
1.2	2009-05-05	Renamed as Endpoint Location Service - Final

---

This page is intentionally left blank

# 1 Executive Overview

## 1.1 Purpose

An Endpoint Location Service (ELS) is a simple directory of technical services. In the short term, these services will be Web services facilitating message/document exchange. However, an ELS implementation could allow clients in the e-health community to locate any electronic service offered by healthcare provider organisations.

ELS can facilitate any kind of electronic service resolution, but is primarily used for determining how to transfer clinical documents. It allows a message producer (source) to transfer its message to an intended recipient (target), even if the producer has no prior knowledge of the method chosen by the recipient to handle such a transfer.

Information required to perform an ELS lookup includes the target healthcare organisation and kind of message.

## 1.2 Implementation

Currently, an ELS must be implemented as a Web service, although other components may be included, for example, a Web application.

A lookup operation returns a set of structures containing technical service endpoints to perform document transfers. In the short to medium term, all endpoints will be for Web services.

Healthcare organisations may deploy their own ELS or engage a third party organisation to host an instance on their behalf.

## 1.3 Document Exchange Interactions

An ELS record is referred to as an *Interaction*. An *Interaction* contains an attribute, *serviceInterface*, which defines the mode of document transfer. Principal modes are *delivery* (upload) and *retrieval* (download).

Services may be hosted by third parties on behalf of healthcare providers. In such cases, a notification may have to be sent to a healthcare provider after a document is transferred. Alternatively, if a retrieval service is used, a notification may be sent to alert a target organisation that a particular document is ready.

A particular *serviceInterface* implies a corresponding technical service type and it is possible that a full interaction scenario may include more than one service invocation, for example, notify and retrieve. The *serviceInterface* attribute formalizes the service type. It would not be possible to send a clinical document to a *notification* service or a notification to a *delivery* service.

## 1.4 Deployment

Every healthcare organisation has one associated ELS. It should be possible for a single ELS instance to be shared by several organisations. NEHTA believes a distributed topology of ELS instances to be optimal for purposes of deployment and administration. However, in theory it is possible for all healthcare organisations to share the same ELS.

## 1.5 Usage

Use of an ELS is required when a document provider does not know how to transfer its clinical document to a specific recipient. For example, a pathology laboratory (document provider) needs to transmit a pathology report (document) to a medical clinic (target) for the first time. However, if the laboratory has previously resolved the clinic's ELS to locate its document exchange services through that clinic's ELS, there is no need for an ELS lookup.

Once an ELS record (interaction) has been downloaded, it may be reused indefinitely. If a failure occurs using a previously obtained interaction, an ELS operation can be used to check its validity.

## 2 Preface

### 2.1 Document Purpose

This document describes the architecture for the Endpoint Location Service (ELS). The architecture conforms to anticipated business scenarios and requirements, referenced in section 2.4. Architecture perspectives conform to the NEHTA Interoperability Framework [NIF2007].

### 2.2 Intended Audience

This document may be read by:

- **Solution Architects and System Analysts**, for the purpose of understanding ELS as a key piece of e-health infrastructure.
- **ELS developers**, for the purpose of understanding the interfaces to be exposed by an ELS instance, and to become familiar with the proposed usage.
- **Service developers**, for the purpose of understanding the relationship of document transfer services with ELS references.
- **NEHTA Package Architects and Developers**. At the time of writing, these projects include Discharge Summaries, Pathology Reports, Medications Management, and Referrals. It is advisable that the designers and implementers of these specifications understand ELS as a supporting infrastructure service.

### 2.3 Definitions, Acronyms, and Abbreviations

CA	Certification Authority - a trusted entity that establishes healthcare provider membership in the e-health community by signing the providers X.509 certificate with their own (CA) private key. The certificate containing the corresponding public key would be stored by Web service clients.
Document	A clinical document or ancillary message, such as a notification or acknowledgement for a clinical document. Documents are usually represented in XML, however elements within such documents may contain non-XML data, e.g. formatted according to HL7.
CRL	Certificate Revocation List
ELS	Endpoint Location Service
Endpoint	A URI including network protocol and address. It provides the binding of an interface to an implementation.
FTPS	File transfer protocol over SSL (Secure sockets layer)
HPII	Healthcare Provider Identifier for an Individual
HPIO	Healthcare Provider Identifier for an Organisation
HTTP	Hypertext Transfer Protocol
IHI	Individual Healthcare Identifier

Interaction	Pattern of communication whereby a target obtains its document – corresponds to the patterns outlined in [CPIS2008].
LDAP	Lightweight Directory Access Protocol – a TCP based directory, especially used to retrieve X.509 certificates.
NASH	National Authentication Service for Health – NASH will endorse X.509 certificate profiles and CA(s) for the e-health community.
NEHTA	National e-Health Transition Authority
OCA	Organisational Certificate Authority – an intermediate CA. An OCA certificate is signed by a root or self-signed CA certificate.
OCSP	Online Certificate Status Protocol
Service	In this document the term service generally refers to a technical service as per [NIF2007]. A technical service is usually a Web service.
Service Category	Service types encompassed by a medical realm. Categories are initially expected to be document types specified by the NEHTA packages. Each service category will be identified by a URI.
Service Interface	URI-based definition of a service offered by a role. Initially these interfaces will describe a Web service.
Service Provider	An organisation that hosts a Web service. This could be the target, source, or a third party.
SFTP	File transfer protocol using secure shell (SSH)
Source	Document suppliers / compilers. For clinical documents a source is a healthcare organisation, e.g. pathology laboratory.
SSH	Secure Shell
SSL	Secure Sockets Layer (SSL 3.0 is largely equivalent to TLS 1.0)
Target	The final destination (intended recipient) of a document. For clinical documents a target is a healthcare provider organisation, e.g. medical clinic.
TLS	Transport Layer Security
UHI	Unique Healthcare Identifier (HPIO, HPID, or IHI)
UHI Service	Proposed National UHI Web service
URI	Uniform Resource Identifier – a unique identifier based on a string generally formatted as: protocol+host+resource.
URL	Uniform Resource Locator – a subset of URI allowing a resource such as a document to be located, especially on a network.
XML	Extensible Markup Language
WSDL	Web Services Definition Language
WSS	Web Services Security

## 2.4 References and Related Documents

Primary related documents are included in Appendix A. In particular, the architecture is influenced by scenarios outlined in 5.2, and conforms to requirements specified in [ELSR2008].

## 3 Introduction

### 3.1 Solution Overview

ELS usage will be a key process in the national e-health environment. An application attempting to establish communications with some service can use an ELS to dynamically discover the service implementation.

### 3.2 Solution Scope

Services resolvable through the ELS are not restricted, however in the short term it will be used to facilitate the exchange of clinical documents. Other scenarios can be realised by extending *serviceInterface* types and their associated semantics.

ELS interfaces need not change simply because new kinds of services are deployed in the future. For example, an ELS could be implemented to resolve the endpoints of other ELS instances or to resolve services that return disclosure statements, quality of service agreements, organisational charters, etc.

#### 3.2.1 Solutions Out of Scope

ELS is not designed to be a general purpose directory. It will not support searching based on extensible properties. An ELS client needs to supply a unique identifier of the target as an input parameter. In addition, the client must supply the kind of service (*serviceCategory*) supported by the services it wants to resolve.

Searches such as *"find all referral services of paediatricians in Sydney who can handle immediate referral of patient ABC"*, are not supported by ELS. Instead, a typical search may be similar to *"find all referral services for healthcare provider identified by HPIO 0000555566667777"*.

### 3.3 Solution Goals and Objective

Broad design goals are listed below. They conform to requirements outlined in [ELSR2008].

1. ELS are implemented as Web services.
2. Services can be resolved for a specific HPIO (or other unique identifier).
  - a. Services can be resolved for specific service types, e.g. pathology, referral, etc.
  - b. Services can be resolved for specific kinds of exchange, e.g. delivery, retrieval.
    - i. The service endpoint can be resolved.
    - ii. The service provider can be identified.
    - iii. X.509 Certificates to be used to secure service data exchange can be resolved through references associated with the service.
3. Service categories can be extended without the need to change ELS interfaces.
4. Service interfaces can be extended without the need to change ELS interfaces.
5. The WSDL for the ELS lookup interface will be published by NEHTA.

6. The WSDL for the ELS update interface will be published by NEHTA.

### 3.4 Assumptions and Dependencies

Assumptions below translate into explicit requirements; see [ELSR2008].

1. At a future time a national UHI service will be implemented providing mappings from HPIO to associated ELS instance.
2. At a future time the NASH will be implemented.
  - a. NASH will act as an e-health certification authority (CA).
  - b. NASH will allow resolution of X.509 certificate references associated with an HPIO.
3. NEHTA packages such as Pathology Reporting and Discharge Summaries can rely on ELS to resolve document exchange services.
4. *Most ELS data is maintained from the perspective of information consumers.*
  - a. Healthcare organisations that need to receive documents will be responsible for keeping ELS data up to date. They may choose to delegate this function to a nominated messaging provider organisation(s).
  - b. Healthcare providers that need to send documents must lookup the recipient (target) ELS prior to sending a document for the first time.
  - c. Interactions with retrieval interfaces will be organised from the perspective of information suppliers, so the target must lookup the endpoint for a document source, i.e. the usual roles are reversed.
5. Services referenced through an ELS may be outsourced by healthcare organisations.
  - a. Implementations may be outsourced.
  - b. Hosting may be outsourced.
6. Service information resolved using an ELS may be reused without resorting to subsequent ELS lookups.
7. Service clients will require a means to check the validity of service endpoint bindings.
  - a. ELS will provide an operation to support validity checking of a record.

## 4 Solution States

### 4.1 Today

ELS will be implemented to facilitate lookup of services for exchanging clinical documents and related artefacts, including notifications and acknowledgments.

Services will be segmented into document categories initially corresponding to the NEHTA packages. Pathology Reporting (see [PRRPES2008]) and Discharge Summaries packages will rely on ELS to resolve services.

#### 4.1.1 Healthcare Provider Directories

In the absence of a standard method of obtaining endpoints, some healthcare messaging providers have devised local directory solutions of their own. These messaging providers maintain location information for document exchange on behalf of their clients, actual healthcare provider organisations. Some messaging providers use secured email to transfer documents.

Aside from endpoint resolution, such directories include information equivalent to service categories and service interfaces. They may also include phone numbers, addresses, and contact names. Finally, they may store certificates whose public key is used to encrypt documents for recipients. Most directories of this nature are maintained by and from the perspective of the document sender.

Principal drawbacks to having a document originator (source) responsible for maintaining directory information are scalability, maintenance, and national consistency. As the number of recipients increases, it becomes more problematic to store the necessary records.

If more than one document source relies on the same document target, their directories must contain duplicate information. Although endpoints are unlikely to change often, whenever there is a change, the risk of data inconsistency between disparate directories increases.

### 4.2 Tomorrow

It is anticipated that some jurisdictions or other agencies will build and operate ELS implementations in the near future. Certificate references may refer alternatively to certificates signed by the Medicare OCA or an equivalently trusted entity.

#### 4.2.1 Service Directory Providers

ELS places the responsibility of maintenance on document recipients rather than document senders. Using such a scheme should improve consistency because only one update is required when a service's details change. For example, when the target healthcare organisation switches the host address of some service, only one ELS requires an update.

ELS does not preclude the use of provider directories which do not conform to the ELS specification. Updates may be triggered from provider directories for some time to come. However, it is crucial that any update to a non-ELS directory is immediately reflected in the standard ELS, i.e. the instance associated with the relevant healthcare organisation.

Location information in non-ELS directories may become unnecessary. Demographic information, which is not required by ELS, can continue in specialized directories, especially directories of medical practitioners. Note that healthcare providers may still use intermediary organisations to

exchange documents, by email and other (non-Web service) mechanisms. Consequently, existing directories may persist well into the future in some form.

### 4.3 Future

In the long term, ELS should become a natural part of client workflow for document exchanges. If service endpoints have not been resolved, the ELS will be consulted. In the infrequent event that a service fails because it has been decommissioned, ELS can be used to determine whether the client reference has become invalid. The reference can be refreshed with another ELS lookup.

A UHI record will provide the starting point to obtain ELS service endpoints for all healthcare providers. Consequently, any member of the e-health community will be able obtain the interactions and endpoints of any healthcare provider.

#### 4.3.1 Healthcare Service Providers

Healthcare providers will be accessible in an ELS directory using an identifier (URI) that includes its HPIO. This document typically refers to HPIO-based access, despite the fact that HPIOs cannot be assigned yet.

#### 4.3.2 Non-Healthcare Service Providers

Many healthcare providers will choose to outsource their services to specialist messaging providers. It is an ELS requirement that these provider organisations are able to be uniquely identified. In the future, the format and maintenance of such identities will become standard.

#### 4.3.3 Deployment

NEHTA intends to allow the market to decide how ELS instances will be deployed.

There are two possibilities:

1. A central ELS will be implemented for all healthcare provider organisations and individuals.
2. Many ELS instances could co-exist simultaneously. Some will be associated with one HPIO, while others will be associated with more than one.

It is more probable that ELS implementations will be distributed. A hospital will perhaps host an ELS associated with its primary and subsidiary HPIOs. Several GPs may share an outsourced ELS instance. Larger clinics may choose to host their own ELS. Pathology laboratories could host their own ELS or have it hosted by an IT service provider. Although the distributed scenario may appear chaotic, because records are only updated in one ELS instance, the situation should be self-regulating.

The distributed model poses a risk to consistency of the UHI service because it must be updated whenever a healthcare provider changes its ELS instance. On the other hand, if a central ELS were to gain acceptance in the marketplace, there would be no need for the UHI service to contain an operation returning ELS references.

#### 4.3.4 Non-Web Services

NEHTA standards are currently predicated on Web services. Web services are not suitable for transferring large binary files, e.g. CT scan images, using a Web service, especially when they must be encrypted. ELS is designed to be

extensible, so other mechanisms and protocols (e.g. FTPS, SFTP) may be returned by an ELS to exchange non-XML document types.

# 5 Business Perspective

## 5.1 Requirements

Business requirements are outlined in [ELSR2008]. ELS is designed as a specialized directory, containing information to allow document transfer between source and target healthcare providers. An ELS record or *Interaction* contains a service endpoint for that purpose.

## 5.2 Document Exchange Scenarios

Patterns outlined in [CPIS2008] appear in the lists below. To exchange a clinical document from a source to a target, these patterns are:

1. *Notify and Retrieve*
2. *Deliver*

To obtain a clinical document from a source it is:

- *Retrieve*

To acknowledge receipt of a document to a source it is:

- *Acknowledge*

Scenarios below assume that the target HPIO or equivalent unique identifier has been determined in advance. Depending on the circumstances, the target organisation would come from a patient, referring GP, hospital, or prior document order/request. An HPIO would be obtained from the UHI service, when it becomes available.

Although not always stated explicitly, any service may be hosted by a service provider acting on behalf of some entity (source or target). When this is the case, the message is transmitted to its final destination using a mechanism agreed between the target and its messaging provider.

### 5.2.1 Retrieve

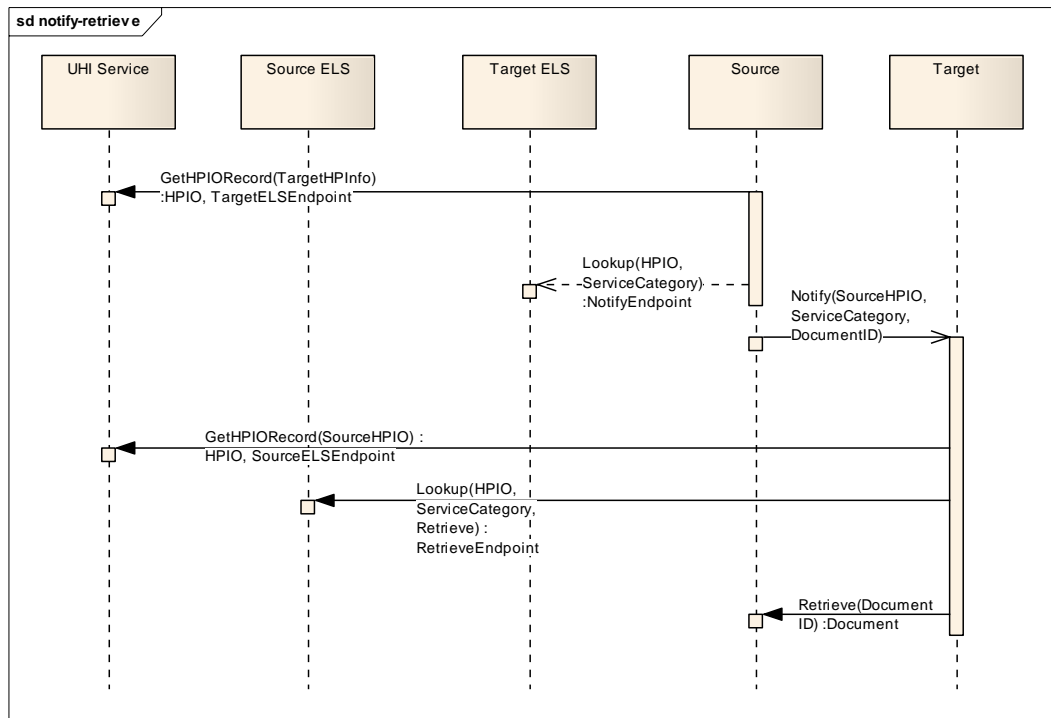
In a *Retrieve* scenario the target acts as a service client. A retrieval service is a download service hosted by or on behalf of the document source.

Standalone *retrieve interactions* are probably impractical for the following reasons.

1. A target may not know a document exists in which it is interested. Suppose *ABC's* patient *PAT* is discharged from hospital *HOS*. The discharge summary is required by *ABC* for informed treatment of *PAT* but there is no automatic means for *ABC* to know this. (However, *ABC* may.)
2. *ABC* cannot ensure that any document source using its ELS will provide the necessary download service. This would require prior agreement between two parties such as *ABC* and *HOS*.
3. Even when a source does host a *retrieve* service and a target is aware of its endpoint, polling would still be necessary. Although [PRRPES2008] allows for polling through the *listSealedReports* operation, it is wasteful in terms of time and resources.

Targets cannot offer *retrieval interaction*, but can offer *notify interactions*. When they do, they should also support an alternate approach such as *deliver*, unless they have come to an arrangement with every source for each service category they support. Sources can always offer *retrieval interactions*.

### 5.2.2 Notify and Retrieve



**Figure 1 Notify and Retrieve**

The source healthcare provider process begins by resolving the endpoint for the target ELS. It invokes the UHI lookup service, supplying information to resolve the target healthcare organisation. It then obtains the interaction applicable to the service category of the document to be transferred.

ELS may return a set of interactions. If the set is empty, the implication is that the target does not support the indicated service category. If the target supports more than one interaction for the service category, it is up to source to decide which one to use.

In this case, the source uses the *Notify* interaction. It sends a notification through the target service (contained in the returned interaction) that a document is ready for collection. The actual transfer will be performed when the target calls a download (*retrieve*) service hosted by or on behalf of the source. If the source does not host a retrieval service, it must not choose this interaction.

Next, the target resolves the ELS endpoint of the source through the UHI service. It uses the source ELS to resolve the endpoint for the retrieve interaction. Since the inputs to the lookup operation are the service category as well as a designated interaction (*retrieve*), only one associated endpoint per interaction is returned. The target then acts as a client of the source service to download the document.

The notification may contain the retrieval endpoint in which case there would be no need to consult the source ELS. Finally, as with most interaction modes, the target may send an acknowledgement. See 5.2.5.

### 5.2.3 Deliver

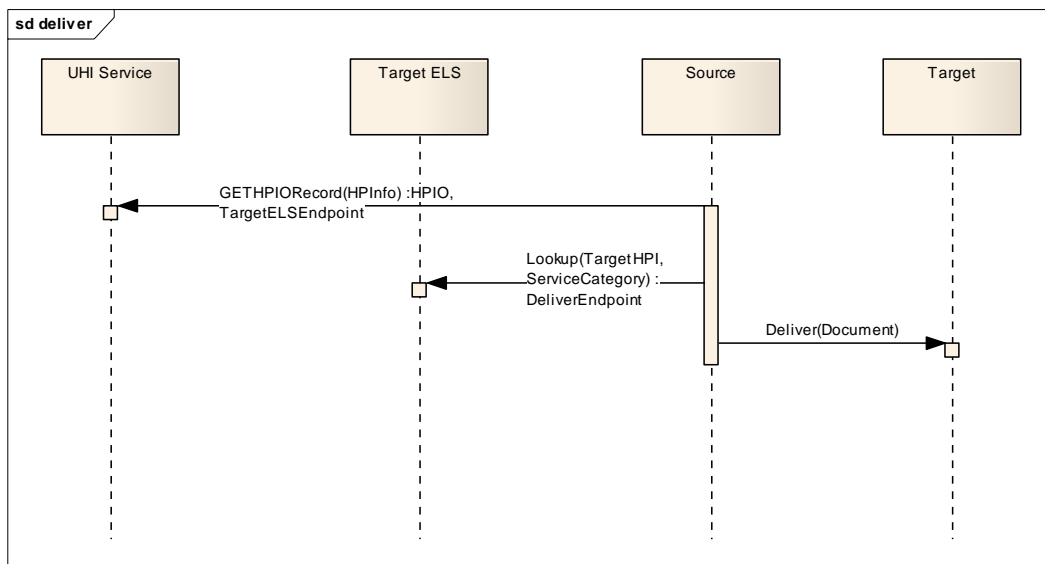


Figure 2 Deliver

*Deliver* is the simplest interaction. Firstly, the UHI service is used to resolve the target ELS endpoint. Secondly, the target ELS is used to lookup an interaction. These steps are the same regardless of which transfer mode is desired.

In this case *Deliver* is chosen by the source. The source process acts as a client of the target Web service to upload the document.

Finally the target should send a document acknowledgement to the source. See 5.2.5.

### 5.2.4 Deliver and Notify

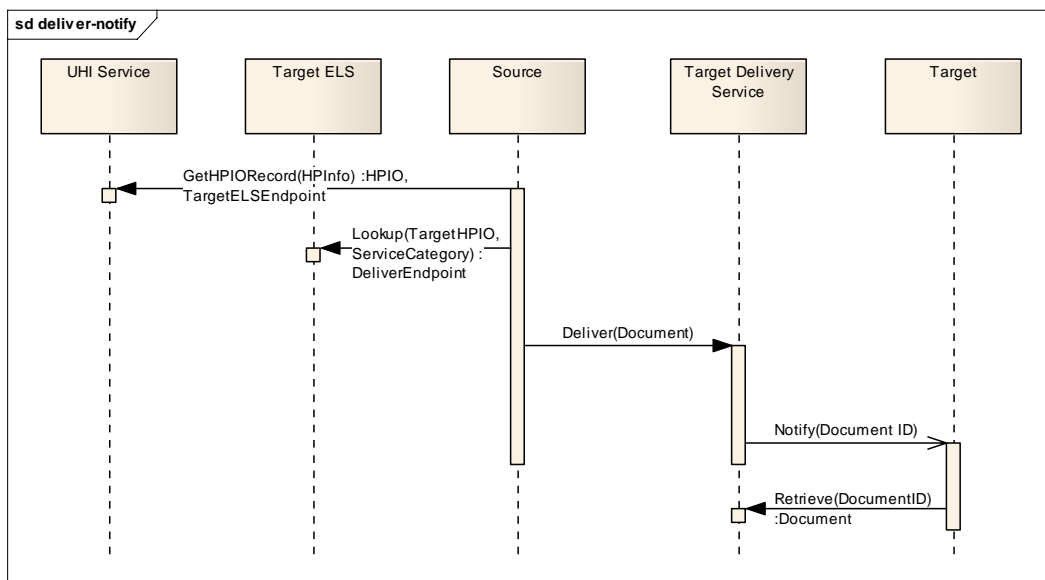


Figure 3 Deliver and Notify

This scenario is the same the deliver scenario from the perspective of ELS. Notification and retrieval shown in Figure 3 is accomplished behind the scenes as far as the source organisation is concerned. Here, a messaging provider hosts a deliver service on behalf of the target. Delivery of the document is achieved by invoking a Web service exactly as in the *deliver* interaction. The exchange thereafter is exclusively between the messaging provider and the target.

Note that the document will be encrypted for the target according to [XSPP2009]. A wrapper document will have been encrypted for the service provider according to [WSP2009].

This scenario may also apply in situations where a messaging organisation defers processing of clinical documents but handles notifications as they arrive. There are other scenarios that can be applied equally well.

### 5.2.5 Acknowledge

Although the preceding figures do not show document acknowledgement, in practice, an acknowledgement will almost always be required. Acknowledgements are intended to alert the source that a target has actually received a document, notwithstanding behind the scenes processes.

Acknowledgements are most important when an agent or third party hosts a *deliver* Web service or acts as a client to a *retrieve* Web service. NEHTA packages endpoint specifications such as [PRRPES2008] will usually allow multiple attempts to download or upload a document. Service invocation with the same input is considered an idempotent operation. After an acknowledgement arrives there is no further need to attempt document transfer.

Acknowledgments are required even when the healthcare provider itself hosts a *deliver* Web service or is a *retrieve* client. It is possible that the document recipient crashes immediately after transfer, leaving the source unaware of whether the whole document was received.

ELS can be used to resolve the source supported endpoint for document acknowledgement. The sequence is the same as for the *deliver* interaction since an acknowledgement can itself be regarded as a document. An acknowledgement *serviceInterface* can be supplied to the ELS lookup so that only the relevant interaction is returned.

This is illustrated by Figure 4.

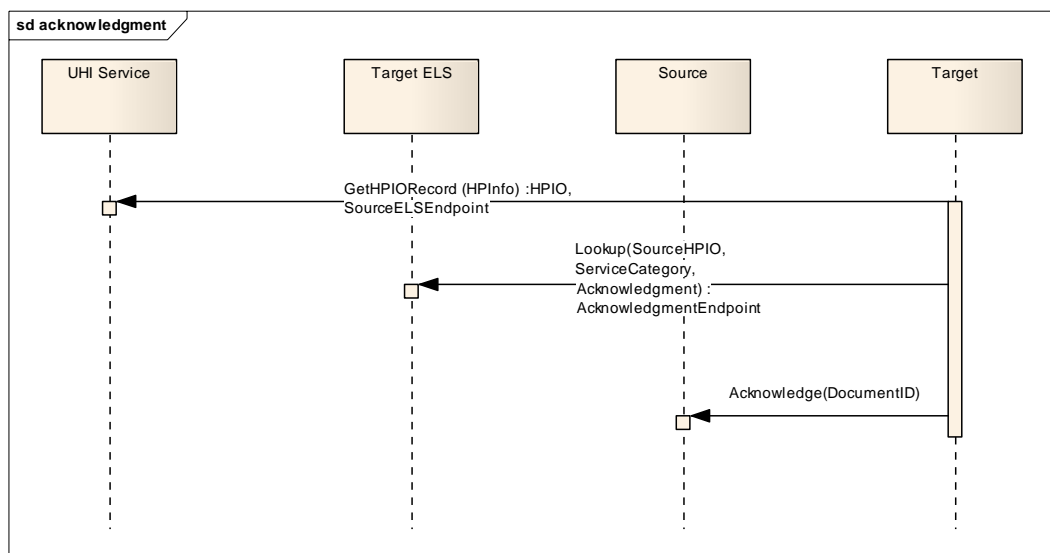


Figure 4 Acknowledge Interaction

### 5.2.6 Caching of Interactions

ELS client processes should cache interaction information. In most cases, the ELS need only be contacted once per HPIO/Service category lookup. Once the interactions are downloaded, subsequent ELS lookups are not required. The scenarios may therefore simplify to those indicated in the diagrams below.

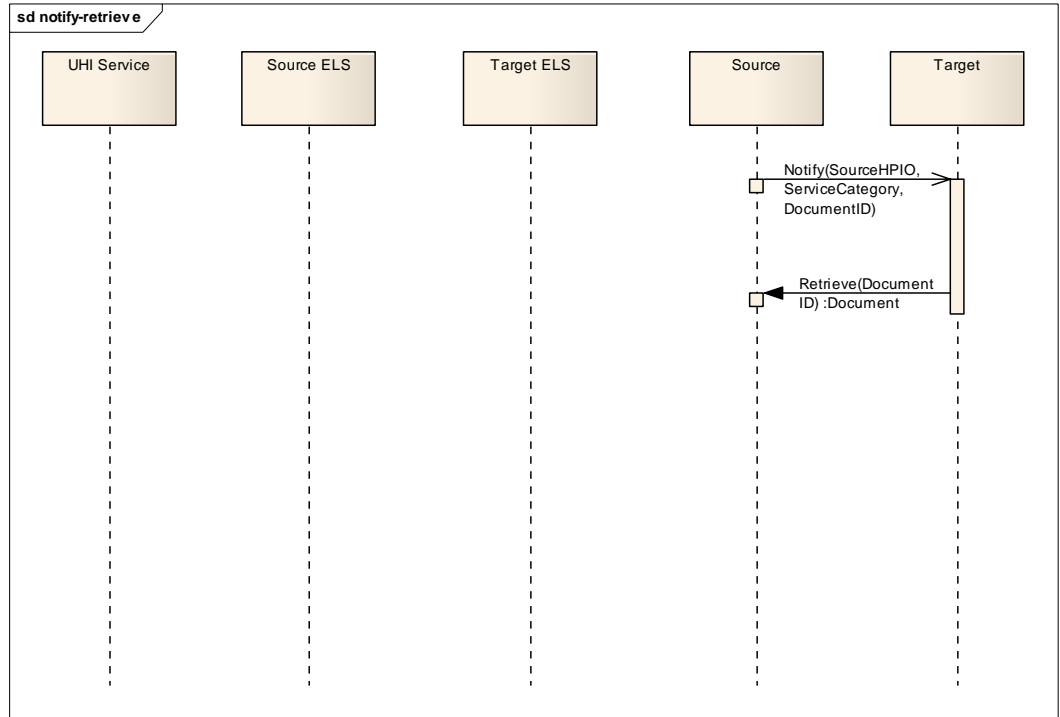


Figure 5 Notify and Retrieve without Lookup

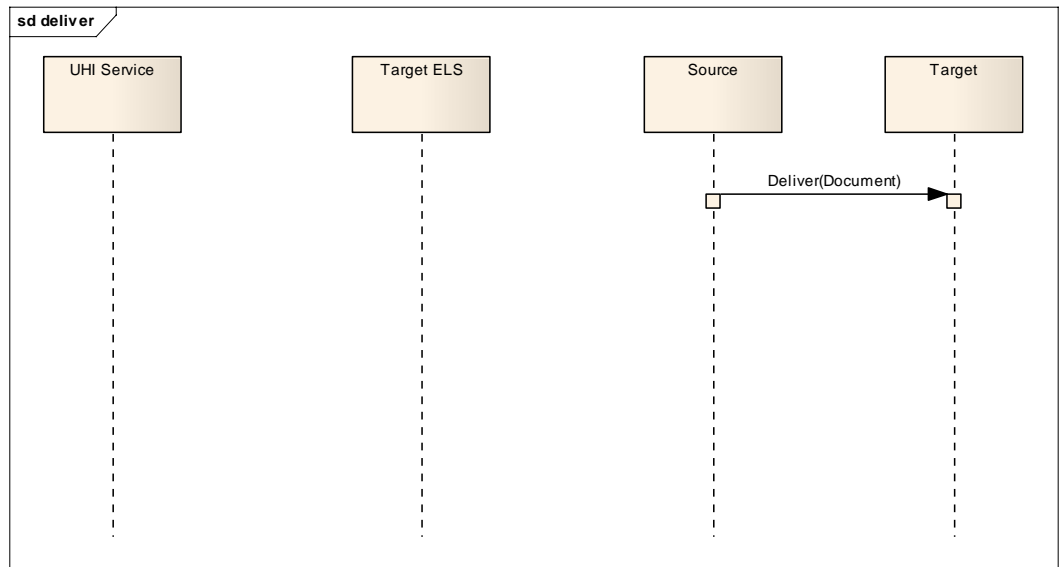
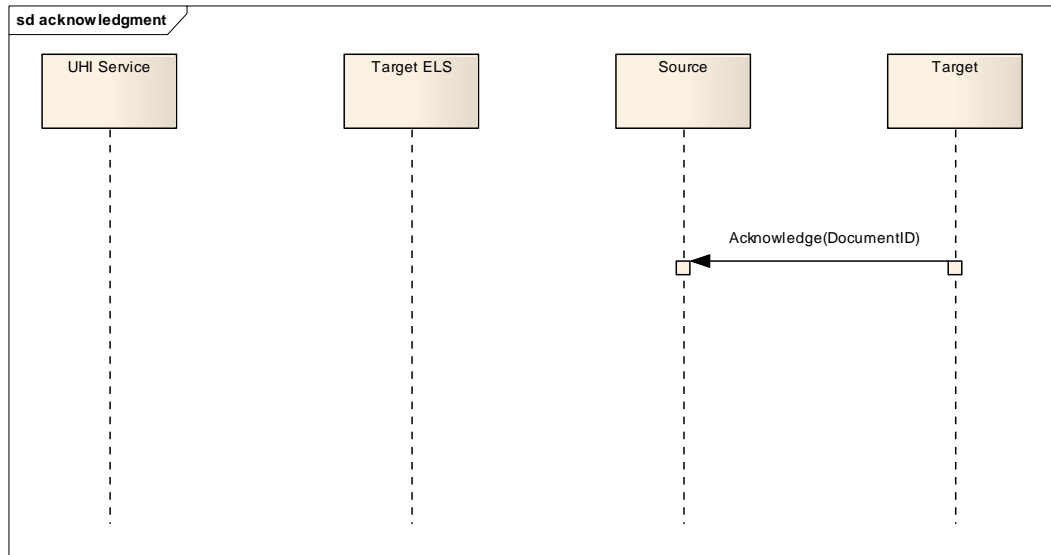


Figure 6 Deliver without Lookup

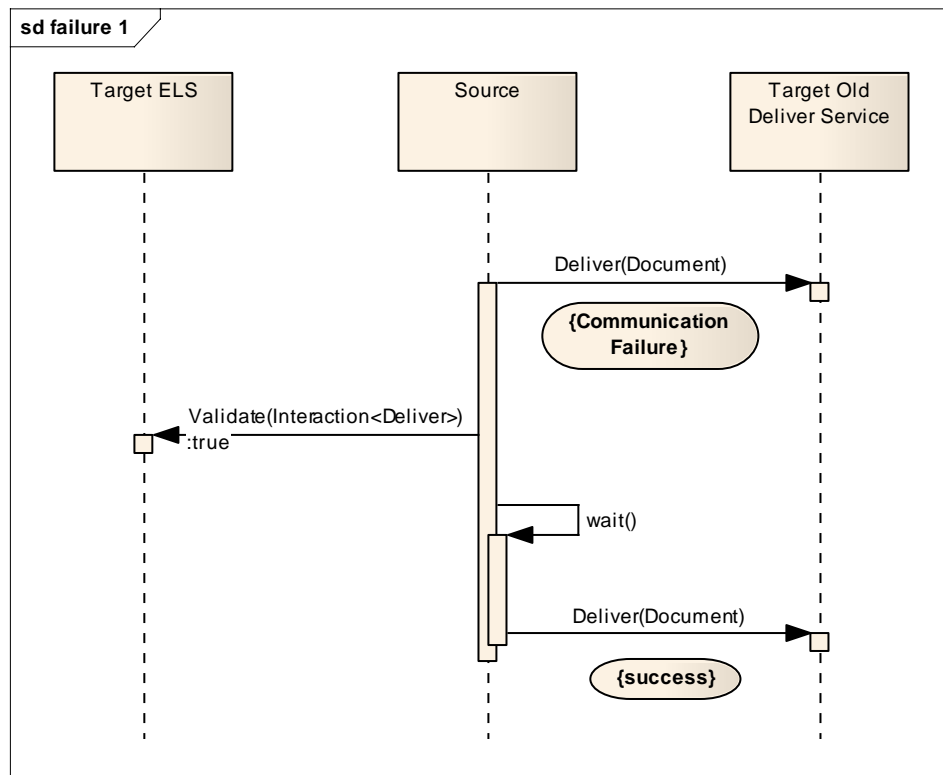


**Figure 7 Acknowledge without Lookup**

5.2.6.1 Failure Situations

If a failure does occur, it may become necessary to check that the cached interaction is still valid. ELS has an operation to accomplish this. The diagrams below illustrate the sequences to be completed in the event of communications failure.

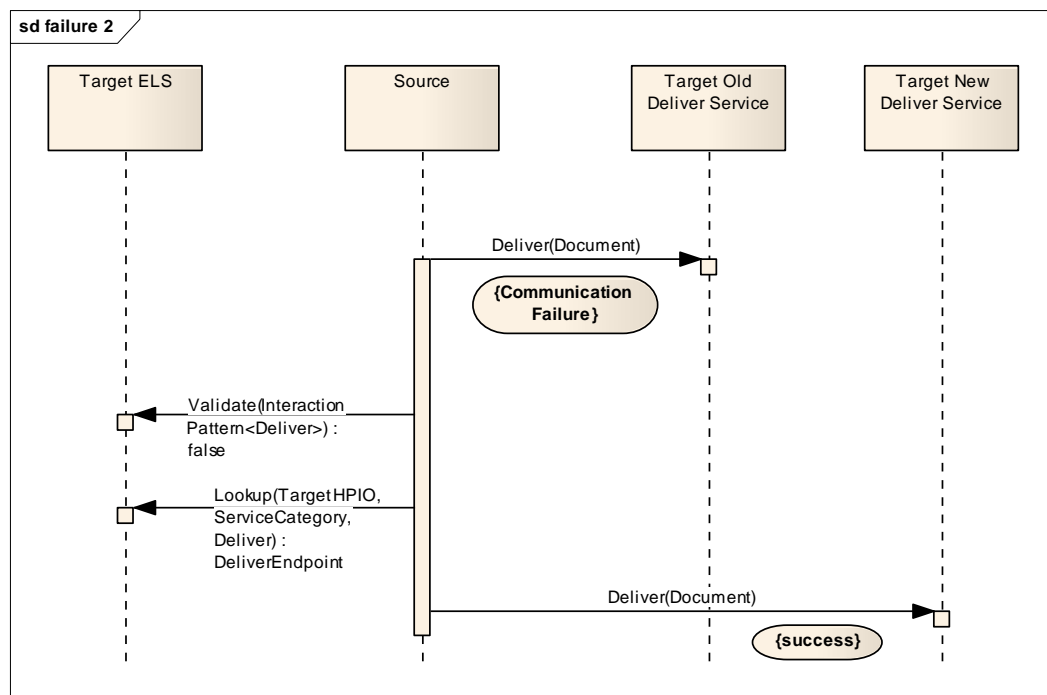
ELS endpoints can also be cached. If a failed service runs on the same host as the ELS, the operation to verify the interaction is also likely to fail. One possible cause may be that the ELS service for the relevant HPIO has moved. Assuming the UHI service has been deployed, a client can check the status by looking up the HPIO record and checking that the ELS endpoint matches the cached value.



**Figure 8 Temporary Failure**

In Figure 8 a communications failure occurred. The client (source) is unaware if the failure is due to the service no longer being available, or is caused by a temporary problem with the host, network, DNS, etc. An operation is invoked

against the target ELS to validate the cached interaction. In this case the validation operation returns *true*, indicating that the interaction and endpoint agree with the remotely cached copy. Presumably, the problem leading to the error will be resolved in the near future, at which time the originally called service can be used.



**Figure 9 Failure Due to Service Update**

Figure 9 shows the sequence required when the cached copy has become stale. An error occurs, just as in the previous case. When the client (source) invokes the validation operation on the target ELS, *false* is returned, indicating that the remote copy is no longer valid. A fresh lookup is necessary and the updated service endpoint is returned.

### 5.3 ELS Community

Healthcare organisations participating in the e-health community should also participate in the ELS community. At present, organisations coordinating messaging between healthcare providers have independently developed their own provider directories. A standard ELS can co-exist with existing solutions as long as organisations adhere to the following rules:

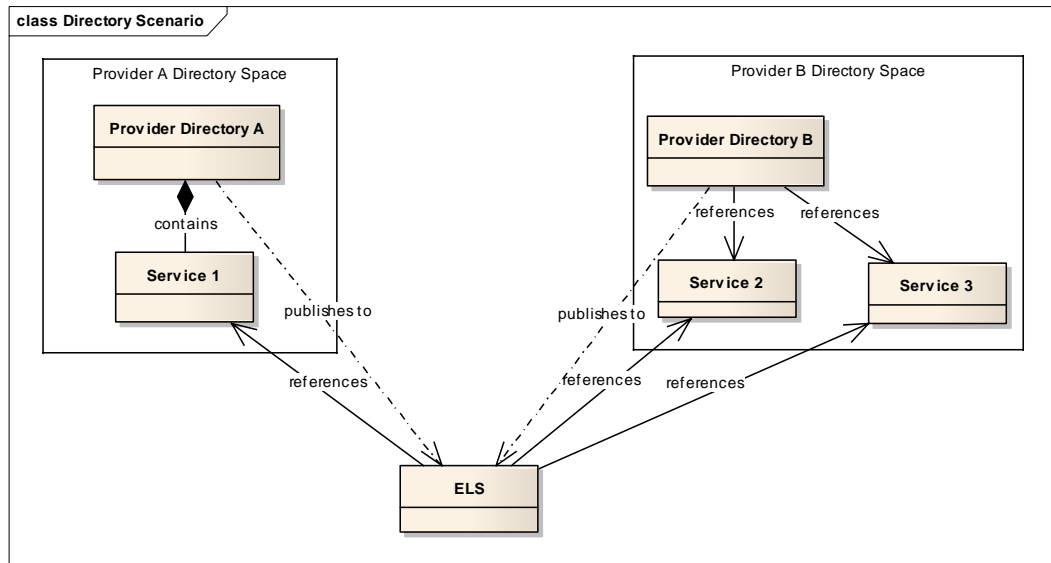
1. Only one ELS implementation is associated with any healthcare organisation.
2. Whenever a service is added or removed from a provider directory, the corresponding ELS must be updated to reflect the current state. (This implies that the provider must have permission to modify the ELS or delegate the update to the pertinent healthcare organisation.)

As long as these rules are followed, there can be any number of local directories supporting different configurations. A sample situation is illustrated by Figure 10.

In this diagram there are two provider directories referencing various services. The containment relationship of *Provider Directory A* to *Service 1* is meant to indicate that both the directory and service are hosted by the same entity. Services and directories could be co-located; it is not important from the perspective of the ELS.

Both directory providers must ensure that any services they refer to are also referenced by the ELS. Directory business processes, automated or otherwise, can utilize the update interface supported by the ELS specification.

The ability to develop common workflows in which provider directories publish their changes to an ELS is a key reason that an update interface has been specified. See 7.3.5.3.



**Figure 10 Provider Directory Example Deployment**

Figure 11 reflects the relationships among healthcare providers, ELS providers and implementations, local provider directories, the UHI service, and ELS data.

For the purpose of data consistency, there is a 1:N relationship between ELS instance and healthcare provider organisations. A provider must not be associated with more than one ELS instance, although that instance may expose more than one endpoint, for example, to use WSS or TLS transports.

In time there will be a global UHI service instance, exposing operation(s) to obtain healthcare provider records. Each record will contain associated ELS endpoint(s).

Provider directories may be associated with many healthcare providers. In such cases they need to update ELS *interactions* to be consistent with local data.

A key reason that provider directories need to coexist with ELS is that they currently provide routing services for several healthcare organisations. Some of these directories are only interested in maintaining listings for one service category. By contrast, an ELS instance is responsible for maintaining interactions for every service category supported by a healthcare provider.

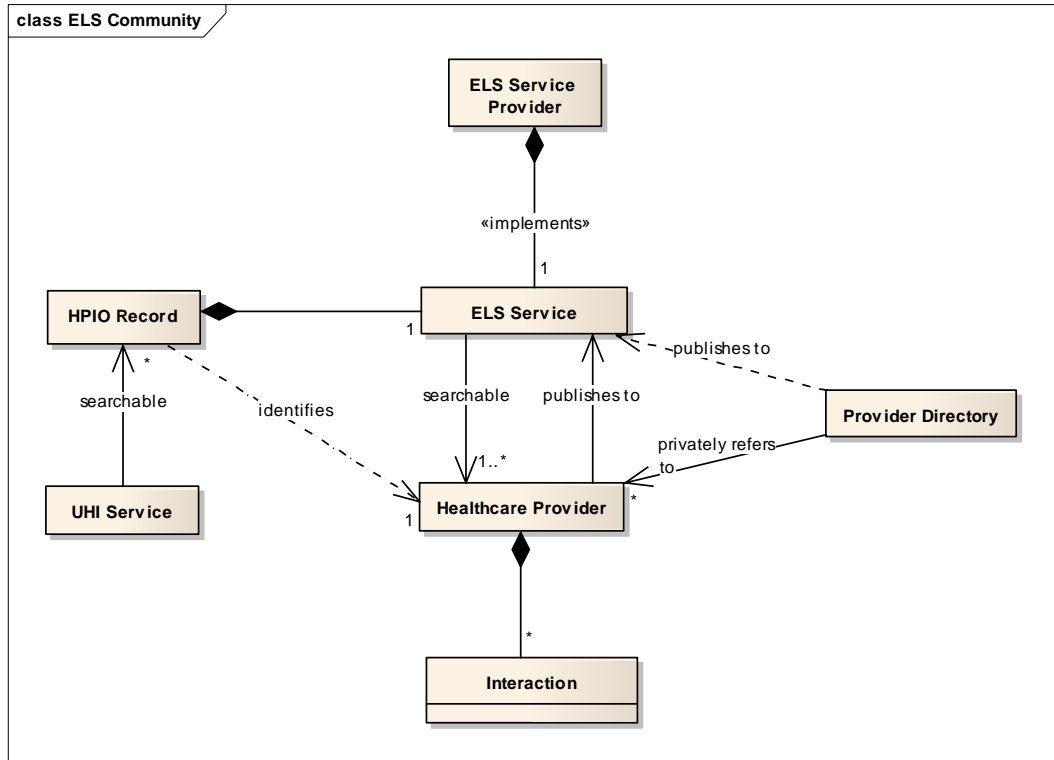


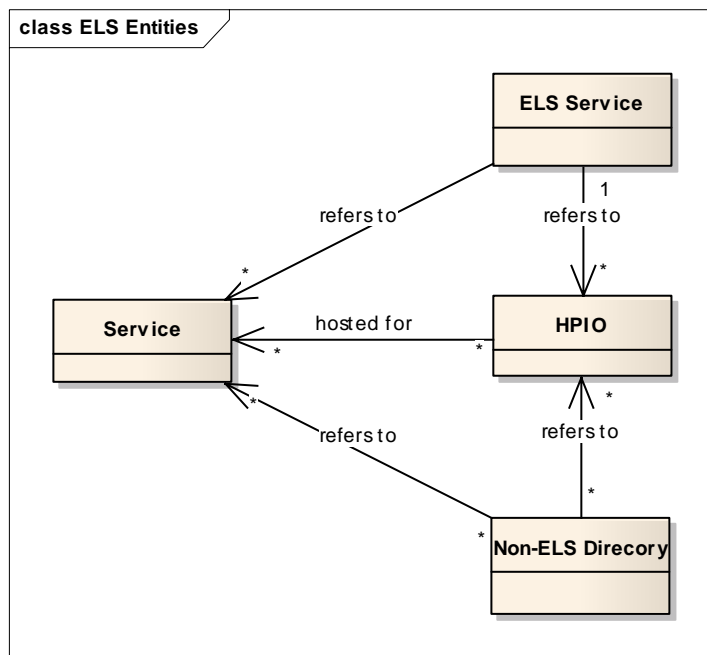
Figure 11 ELS Community Model (Services Omitted)

## 6 Information Perspective

### 6.1 Requirements

Informational requirements are outlined in [ELSR2008]. Primarily, ELS data structures model interactions for document exchange. AN ELS lookup returns zero or more interactions supported for a particular service category.

### 6.2 Services and Provider Directories



**Figure 12 ELS Services and Provider Directories**

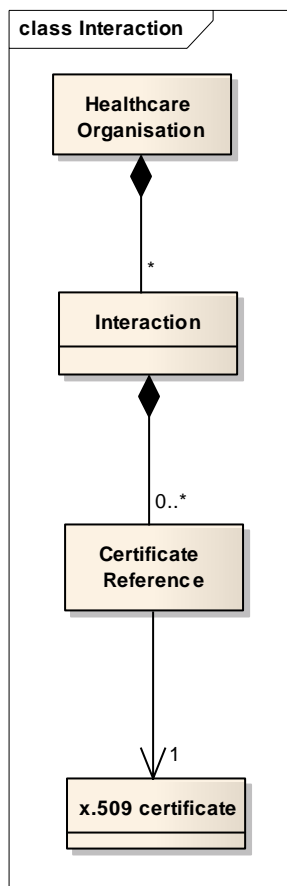
Services are Web services that handle document exchange. Figure 12 illustrates the relationship of HPIO, ELS, provider directories and service information, contained by *Interaction* records.

Provider directories are systems outside the scope of this specification. Consequently, the cardinality of their relationships to healthcare providers and services cannot be restricted. The reality of any particular implementation may be different to that depicted here.

Services are shown as aggregations of a HPIO because [WSP2009] and [PRRPES2008] imply no restriction on multiple healthcare providers using the same service for document exchange.

### 6.3 Interaction Data Structure

Interactions in Figure 11 are shown as containments because an interaction is specific to a HPIO and its existence contingent on that of the HPIO. See Figure 13.



**Figure 13 Interaction Structure**

Although an HPIO is shown as having zero or more interactions, it would be useless for an HPIO with no interactions to be present in a ELS. However, such a situation may temporarily exist.

Each *Interaction* refers to exactly one service. It also refers to zero or more certificate references. In line with [XSPP2009], an X.509 certificate(s) is required to encrypt and/or sign a clinical document prior to transfer. However, for retrieval services, a target X.509 certificate will be used for the purpose.

In time, the certificate can be obtained from the NASH service using a QualifiedCertRef. Multiple certificates could be issued for any given HPIO; the certificate reference specifies which to use for the indicated service (see 6.4).

### 6.3.1 Interaction

Interactions are specified in an XML schema. Its structure is outlined below:

- Interaction: xsd:complexType
  - serviceCategory: xsd:anyURI
  - target: xsd:anyURI
  - serviceProvider: xsd:anyURI
  - serviceInterface: xsd:anyURI
  - serviceEndpoint: xsd:anyURI
  - certRef: ElsCertRef[0..\*]

Attribute *serviceCategory* identifies the business service. NEHTA packages will be responsible for defining this attribute.

*serviceCategory* will usually imply which kind of clinical document is to be transferred. It may be broad enough to encompass the entire range of

possible document formats and clinical terminologies, or limited to a single format/terminology grouping. There may be many *serviceCategories*, each with a single targeted *serviceInterface*.

Attribute *target* is a qualified name uniquely identifying the healthcare provider. In the long term, it is anticipated to qualify a unique HPIO (see 4.3.1).

Element *serviceInterface* identifies the service style, such as delivery, notification, or retrieval. In the short term it will also indicate the concrete binding (HTTP, HTTPS, etc) for a WSDL port.

Element *serviceEndpoint* is the URL of the service location.

Element *serviceProvider* is a URI that uniquely identifies the organisation which hosts the service. Messaging providers which are not healthcare organisations will probably be identified by some form of HPIO in the long term.

*qualifiedCertRef* contains a sequence of X.509 certificate references (see 6.4).

Attributes of type *xsd:anyURI* should be URIs conforming to [QI2008].

### 6.3.2 Service category Identifiers

Service category values will be defined by NEHTA packages. Below are the values from the pathology results reporting package [PRRPES2008].

- <http://ns.nehta.gov.au/Pth/RR/SR/Category/Const/SealedReport/1.0>
- <http://ns.nehta.gov.au/Pth/RR/SRAck/Category/Const/Ack/1.0>

### 6.3.3 Target Identifiers

Target Identifiers are qualified identifiers representing the target entities. In the future, the qualified identifier will be based on HPIOs. An example is:

- <http://ns.nehta.gov.au/Id/Const/UhiHpio/1.0#8036000000000212>

### 6.3.4 Service Provider

Service Providers are qualified identifiers that include a unique identifier for an organisation. A service provider may be an outsourced organisation that does not provide healthcare services. See 4.3.1.

### 6.3.5 Service Interface

Service Interface identifies type of service and transport binding. Currently, these bindings will be based on WSDL interfaces, but they may be expanded to other types in the future. NEHTA currently endorses two kinds of transport binding:

- Web service security (WSS) transport binding; in line with the corresponding profile of [WSP2009].
- TLS style transport binding; in line with the corresponding profile of [WSP2009].

Values from the pathology results reporting packing include:

- <http://ns.nehta.gov.au/Pth/RR/SR/Svc/Consumer/1.0>
- <http://ns.nehta.gov.au/Pth/RR/SR/Svc/Supplier/1.0>
- <http://ns.nehta.gov.au/Pth/RR/SRNotify/Svc/Consumer/1.0>
- <http://ns.nehta.gov.au/Pth/RR/SRAck/Svc/Consumer/1.0>

Values from the clinical document delivery package ([CDDEP2009]) include:

- `http://ns.nehta.gov.au/Cdd/Intf/SealedMessageConsumer/TLS/1.0`
- `http://ns.nehta.gov.au/Cdd/Intf/SealedMessageConsumer/WSS/1.0`
- `http://ns.nehta.gov.au/Cdd/Intf/SealedMessageSupplier/TLS/1.0`
- `http://ns.nehta.gov.au/Cdd/Intf/SealedMessageSupplier/WSS/1.0`
- `http://ns.nehta.gov.au/Cdd/Intf/SealedAcknowledgementConsumer/TLS/1.0`
- `http://ns.nehta.gov.au/Cdd/Intf/SealedAcknowledgementConsumer/WSS/1.0`
- `http://ns.nehta.gov.au/Cdd/Intf/SealedAcknowledgementSupplier/TLS/1.0`
- `http://ns.nehta.gov.au/Cdd/Intf/SealedAcknowledgementSupplier/WSS/1.0`

### 6.3.6 Service Endpoint

Service endpoint is simply the binding address for the interface. Currently supported protocols are HTTP 1.1 or HTTPS. See [WSP2009]. Other protocols may be used in the future. Example values:

- `http://sample.services.com:8080/pathrpt/SealedRptConsumer`
- `https://sample.services.com/pathrpt/SealedRptConsumerTLS`

## 6.4 Qualified Certificate Reference

Qualified certificate reference has been included in the ELS specification to identify certificates associated with particular services. It allows for different methods of referencing and/or locating an X.509 certificate. See [QCR2009].

Certificates are used to encrypt a session key which is used to encrypt the contents of an XML document [XSP2009] or a Web service request [WSP2009].

### 6.4.1 Structure

- `ElsCertRef`: `xsd:complexType`
  - `qcr`: `common:qualifiedCertRef`
  - `useQualifier`: `xsd:anyURI`

Element `qcr` is a qualified certificate reference as defined by [QCR2009].

Element `useQualifier` is a URI whose value defines how the certificate is to be used in calling the associated Web service. Presently, the following values are defined:

- `http://ns.nehta.gov.au/Qcr/Use/Payload/KeyEnc/1.0`
  - The public key of the certificate will be used to encrypt the symmetric key to be used for encrypting the payload for the target organization. See [XSP2009].
- `http://ns.nehta.gov.au/Qcr/Use/Transport/KeyEnc/1.0`
  - The public key of the certificate will be used to encrypt the symmetric key to be used for encrypting the contents of the Web service request for the service provider. See [WSP2009]. This value should be used in *Interaction* records where *serviceProvider* and *target* identify separate organizations. Such a situation usually implies that a target healthcare organization uses a messaging provider to host the service on its behalf.
- `http://ns.nehta.gov.au/Qcr/Use/PayloadTransport/KeyEnc/1.0`
  - The public key of the certificate is used to encrypt the symmetric key(s) used to encrypt the payload as well as the contents of the Web service request. If an *Interaction* record contains the same

value for *target* and *serviceProvider* this usually implies that a healthcare provider hosts and manages its own Web service.

### 6.4.2 Cardinality, Organizations, and Usage

Although an *Interaction* record may contain any number of qualified certificate references, in practice there will be only one or two elements. If there is one reference, then *target* and *serviceProvider* will have the same value and the *useQualifier* will be `http://ns.nehta.gov.au/Qcr/Use/PayloadTransport/KeyEnc/1.0`. Two references implies that *target* and *serviceProvider* are different. The qualified reference with *useQualifier* value equal to `http://ns.nehta.gov.au/Qcr/Use/PayLoad/KeyEnc/1.0` must be used to encrypt the payload according to [XSPP2009]. In this case there must be another reference with *useQualifier* value equal to `http://ns.nehta.gov.au/Qcr/Use/Transport/KeyEnc/1.0`. That certificate must be used to encrypt the Web service request according to [WSP2009].

Since payload encryption is only appropriate for a target healthcare provider, a source organisation cannot supply a certificate for that purpose. One certificate reference will exist for retrieval interactions. Its *useQualifier* value should be equal to `http://ns.nehta.gov.au/Qcr/Use/Transport/KeyEnc/1.0` for the purpose of encrypting the Web service request. The caller must specify which certificate will be used to secure the payload in the Web service invocation.

## 6.5 Interaction Request

Type *InteractionRequest* is used as input to an ELS lookup operation (see 7.2). Its structure is outlined below:

- `xsd:complexType: InteractionRequest`
  - `xsd:anyURI: target`
  - `xsd:anyURI: serviceCategory [1..*]`
  - `xsd:anyURI: serviceInterface [0..*]`

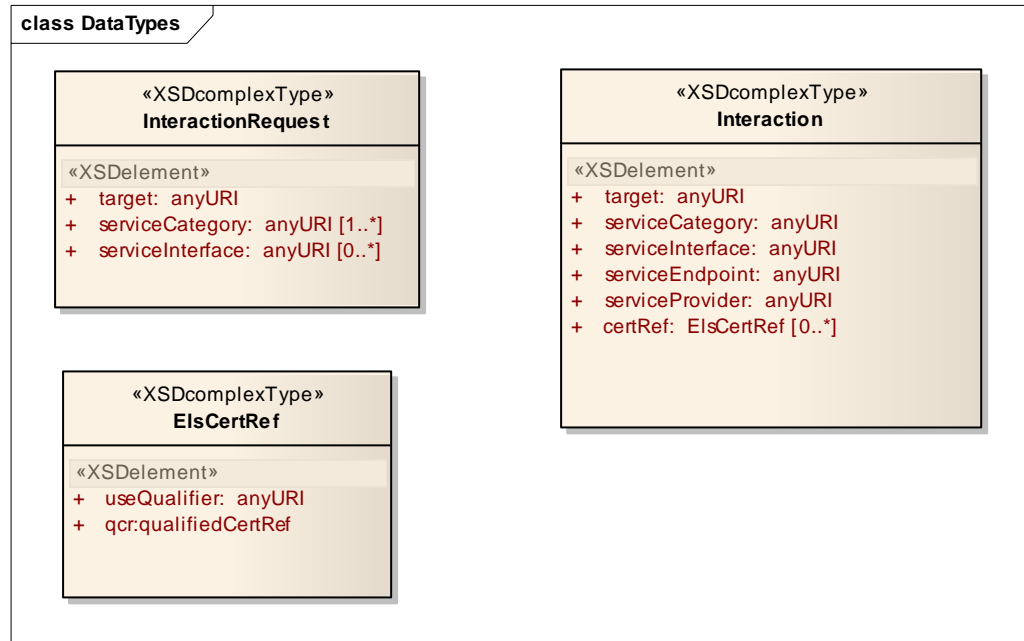
Attributes *target* and *serviceCategory* correspond to attributes of the same name in class *Interaction*. Both attributes are required as input.

### 6.5.1 serviceInterface Identifier

Optionally, element(s) *serviceInterface*, if present, correspond to identifiers listed in 6.3.5. If present, these would have the effect of narrowing the search.

## 6.6 Summary of Information Types

Figure 14 is the class diagram for basic ELS data types. Explanations are given in the sections above.

**Figure 14 ELS Data Types**

# 7 Technical Perspective

## 7.1 Requirements

Technical requirements are outlined in [ELSR2008]. The interface is logically split into two packages, containing lookup and update operations.

## 7.2 Lookup Package

Figure 15 summarises the lookup package. There are two operations, `listInteractions` and `validateInteraction`.

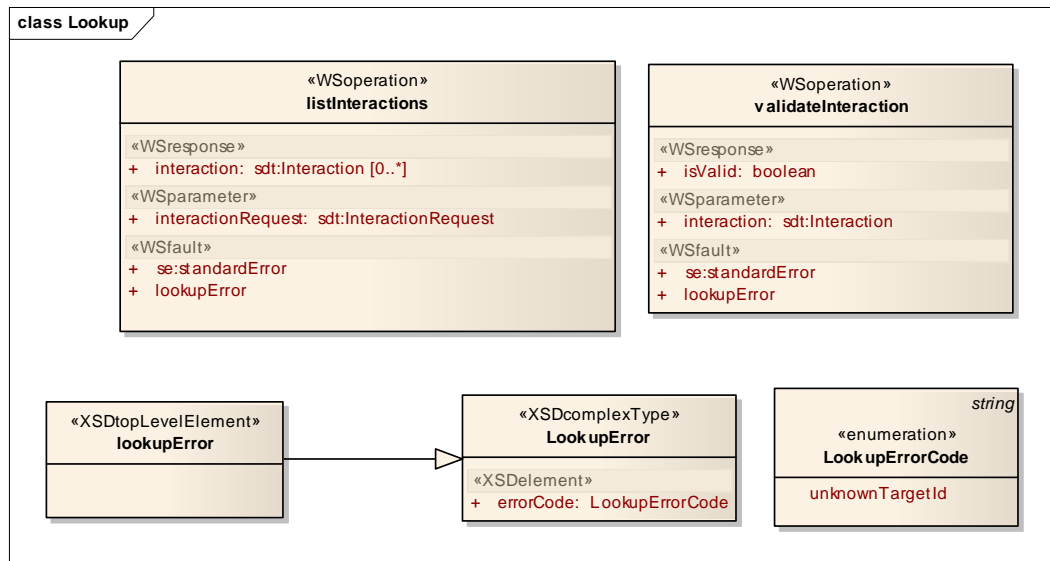


Figure 15 ELS Lookup Package

### 7.2.1 Operation listInteractions

This operation is the core of the specification, realising the primary purpose of ELS. It takes an `InteractionRequest` (see ELS Data Types) as input and returns a set of `Interaction` (see 6.3). The following notes apply.

1. If the optional `interaction` (see 6.5.1) is omitted from the request all interactions will be returned for the input healthcare provider and service category. It will then be up to client to choose which interaction they wish to use.
2. It will be possible for the returned set to contain more than one interaction of the same kind (e.g. *deliver*) for the same service category. The various interactions would refer to different service providers and endpoint addresses. When such situations arise it is up to the client to choose which interaction to use.
3. It is not an error condition if the returned set is empty. It simply means no interactions matched the input.

Returned interactions can be reused. There should be no need to contact the ELS again for the same lookup.

### 7.2.2 Operation validateInteraction

This operation should be used infrequently. Its purpose is to provide confirmation that a remote reference to an interaction has expired (see 5.2.6). A single `Interaction` is required as input. The operation returns `true` if the interaction currently exists; `false` if the interaction has been removed.

If *false* is returned, the client should call `listInteractions` to obtain an up-to-date interaction set for the same `documentCategory`.

### 7.2.3 Error Code

There is only one ELS-specific error code for the lookup package. It contains an enumeration of one value, `unknownTargetId`. This error can be returned from `validateInteraction` and `listInteractions`, indicating that the target attribute of an `Interaction` or an `InteractionRequest` generates no match, i.e. the ELS is not associated with the supplied identifier. This indicates the client is attempting a request using the wrong ELS instance.

Other faults that may be generated are defined as part of the standard error package defined in [WSP2009]. Currently these are identified as follows:

```
servicePermanentUnavailable
serviceTemporaryUnavailable
certificateSkiMissing
certificateKeyUsage
certificateUnidentified
invalidCredentials
notAuthenticated
notAuthorised
badParam
badlyFormedMsg
badTimestamp
badSignature
badEncryption
badSigEncOrder
badCertificateTransmitted
badWsaAction
badWsaMessageId
badWsaTo
badAlgorithmDataEncryption
badAlgorithmKeyEncryption
badAlgorithmC14N
badAlgorithmDigest
badAlgorithmSignature
```

### 7.2.4 Security Considerations

As required by [ELSR2008], all ELS records may be obtained by any healthcare provider in the e-health community. Communications between client and server MUST be secured in accordance with [WSP2009]

The public key used to verify the digital signature will be extracted from a referenced X.509 certificate signed by a trusted CA (see [WSP2009]).

## 7.3 Publish Package

Figure 16 summarizes the publish package. There are two operations, `addInteraction` and `removeInteraction`.

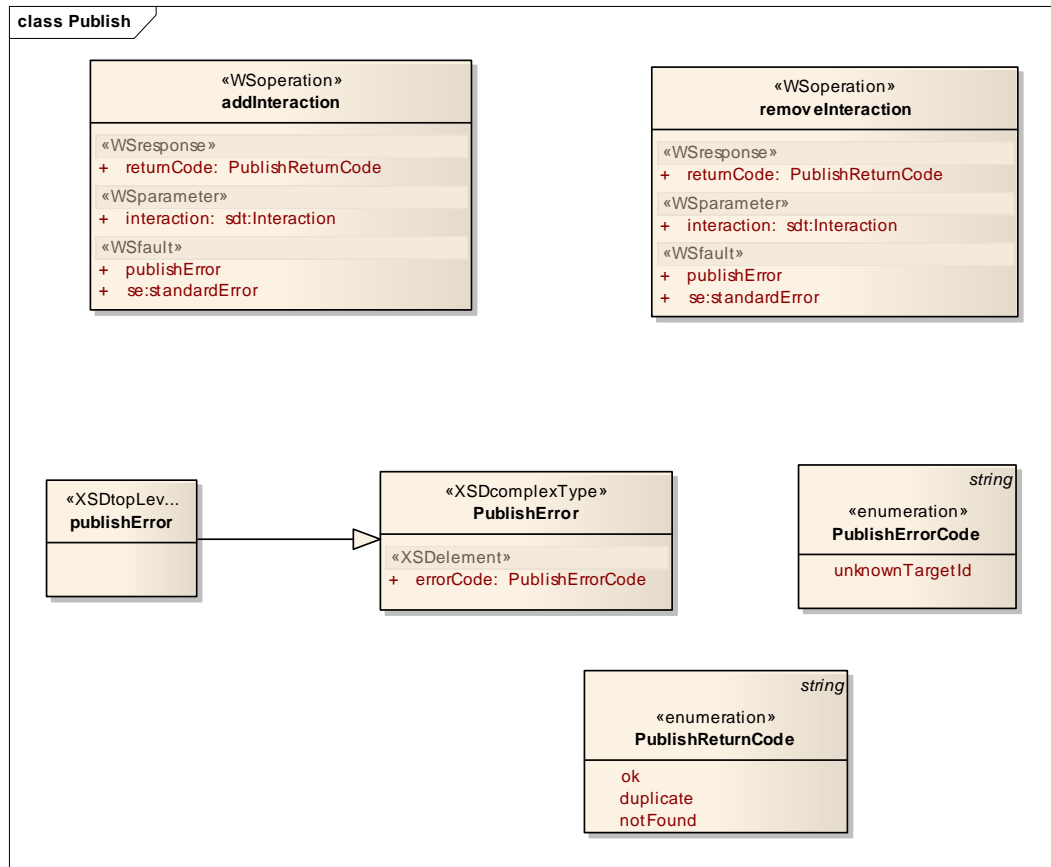


Figure 16 ELS Publish Package

### 7.3.1 Operation addInteraction

This operation is used to create a new record in the ELS. Its input is the Interaction to insert. It returns enumerated type `ELSPublishReturnCode` (see 7.3.3).

### 7.3.2 Operation removeInteraction

This operation is used to remove a record from the ELS. Its input is the Interaction to delete. It returns enumerated type `ELSPublishReturnCode` (see 7.3.3).

### 7.3.3 Return Codes

`ELSPublishReturnCode` is an enumeration. Its values and meanings are:

- ok
  - returned from `addInteraction`:
    - the Interaction was successfully created.
  - returned from `removeInteraction`:
    - the Interaction was successfully deleted
- duplicate
  - returned from `addInteraction`:
    - the Interaction already exists. This is not considered an error, especially since it is possible that the virtual circuit between client and (ELS) server may be disconnected after an insertion but prior to the client receiving a response. If

the client subsequently tries to add the interaction, it will receive this code.

- Will not be returned from `removeInteraction`.
- `notFound`
  - returned from `removeInteraction`:
    - the `Interaction` could not be located in the ELS. This is not considered an error, especially since it is possible that the virtual circuit between client and (ELS) server may be disconnected after a removal but prior to the client receiving a response. If the client subsequently tries to remove the interaction, it will receive this code.
  - Will not be returned from `addInteraction`.

### 7.3.4 Error Code

There is only one ELS-specific error code for the publish package. It contains an enumeration of one value, `unknownTargetId`. This error can be returned from `addInteraction` and `removeInteraction`, indicating that the `target` attribute of an `Interaction` generates no match. This indicates the client is attempting a request using the wrong ELS instance.

As for the lookup interface, NEHTA standard error faults may be generated (see 7.2.3).

### 7.3.5 Security Considerations

Requirements are the same as for the lookup package; see 7.2.4. However there is an additional trust consideration for publishing.

#### 7.3.5.1 Target Healthcare Provider Update

In order for a target healthcare provider to update its associated ELS it must already have an association created. The mechanism to accomplish this will not be defined by this specification, however see 7.3.6.

A valid target MUST be able to invoke operations on the publish package for its own ELS instance. Updates are permitted by the ELS if the X.509 certificate used to sign the request corresponds to a target certificate issued by a trusted CA against its unique identity.

#### 7.3.5.2 OCAs

For read purposes, ELS implementations will have to store the root certificate of any CA it is prepared to trust. Optionally, it may trust intermediate organizational certificates of such a CA. Ideally the ELS should reconstruct certificate chains to up to a trusted root CA issuer.

In the short term, certificates issued by a Medicare Australia organizational CA (OCA) should be trusted. In the longer term, it is anticipated that NASH will publish an approved list of OCAs to be trusted by participants in the e-health community.

When a certificate issued by a Medicare OCA is used, care should be taken that the credential was issued with a policy appropriate for messaging. For example, communications for clinical documents exchanged should not be secured using certificates with a policy targeted for the purpose of Medicare billing.

For update purposes, ELS implementations may rely on identity mappings that are present in certificates signed by a CA it is prepared to trust. Such attributes may form part of the DN (distinguished name) or alternate DN

present in an X.509 certificate. Authority to update an ELS instance may be delegated to a messaging provider by a healthcare target (see 7.3.5.3).

### 7.3.5.3 Delegated Updates

The situation where a provider directory is required to update an ELS instance may be problematic. In general, such updates are necessary to ensure consistency between its own data and the ELS (see 5.3).

It is anticipated that directory providers will also be messaging providers. Assume that such an entity has an appropriate X.509 certificate whereby its identity can be mapped in the same manner as a healthcare provider. This credential can be used to authenticate the entity to an ELS instance. If that instance allows targets to delegate update operations, then the messaging provider may be permitted to add and remove interactions as long as:

- a. The target has delegated appropriate access to that messaging provider;
- b. The *serviceProvider* URI in an input *interaction* matches the identity of that messaging provider. This restriction is to prevent any particular service provider entity from removing interactions of another service provider.

Any ELS implementation may choose the mechanism to implement delegated updates. A role-based approach is recommended.

### 7.3.6 Non-standardized Operations

The update interface is deliberately limited. Standard operations exist so that healthcare providers can consistently maintain the interactions they support, especially on ELS instances provided and/or administered by third parties.

It is probable that the bulk of updates will be done by non-standard means, e.g. by Web form, email, telephone, or face-to-face.

A starting point for the standard operations is that a healthcare provider is registered with an ELS instance. Clearly, there must be a means to accomplish this. A healthcare provider may also want to change its ELS instance or discontinue its practice. So, there must also be a way to deregister a provider.

Note that the ELS specification does not define any register or deregister operations. Instead, individual ELS providers may provide this functionality in their own way.

# 8 Enabler Dependencies

## 8.1 UHI

UHI is not strictly required for prototype implementation. However, it would be preferable if UHI services became available prior to production implementations. There are two dependencies for ELS:

1. Identifiers such as the HPIO need to be allocated against healthcare providers for ELS lookups to make sense (see 6.3.3). Until UHI is realized, interim identifiers will be endorsed by NEHTA.
2. It has been assumed that the UHI record will contain the ELS endpoint (see 8.1.1). There has to be a least one known address to bootstrap the document exchange process. In the most likely long term deployment scenario, there would be multiple distributed ELS instances, and ELS location via the UHI is therefore highly desirable.

### 8.1.1 ELS Bootstrap Reference

The UHI record would have to return the following data triple:

1. An ELS endpoint, of type URL.
2. A combined reference to the ELS transport binding and interface version. In the immediate future, allowable transport bindings are WSS and TLS.
3. An X.509 certificate reference, as defined by [QCR2009]. The client would encrypt ELS requests using this certificate. The exact semantics would depend on which binding (WSS or TLS) were employed.

## 8.2 NASH

### 8.2.1 Single X.509 Certificate per HPIO

References captured by `QualifiedCertRef` (see 6.4) will ultimately depend on the NASH implementation. If NASH endorses more than one certificate directory, service clients must be prepared to resolve certificate references from multiple directories in all associated forms (HTTP, LDAP).

It is also possible that multiple mechanisms may be used to ensure certificates have not been revoked (e.g. CRL, OCSP).

### 8.2.2 Multiple X.509 Certificates per HPI

It is likely that NASH will permit multiple certificates for a healthcare provider, including soft certificates and certificates contained on smartcards or other mobile devices. In this situation a unique organisational identifier could not uniquely identify a certificate.

To allow for such situations, NASH certificate profile(s) will provide appropriate service binding(s) using X.509 attributes.

### 8.2.3 Trust Based on Certificates

Most trust issues identified by this document are very simple, and can be resolved using X.509 certificates. These conditions are especially important:

- Certificate(s) used for Web service invocations are signed by a trusted CA and have been issued with an appropriate policy. See 7.3.5.2.

- There is an appropriate identity mapping contained in every certificate issued to a member of the e-health community.

# Appendix A References

- [CPIS2008] NEHTA, *Concepts and Patterns for Implementing Services v2.0*, 1 December 2008.
- [NIF2007] NEHTA, *Interoperability Framework v2.0*, 17 August 2007.
- [PRRPES2008] NEHTA, *Pathology Result Reporting Package (v1.0 Draft) - Endpoint Specification v2.1*, 4 September 2008.
- [QI2008] NEHTA, *Qualified Identifiers v1.0*, 1 December 2008.
- [ELSR2008] NEHTA, *ELS Requirements v1.1*, 1 December 2008.
- [WSP2009] NEHTA, *Web Services Profile v3.1*, 20 February 2009.
- [XSPP2009] NEHTA, *XML Secured Payload Profile v1.1*, 20 February 2009.
- [QCR2009] NEHTA, *Qualified Certificate Reference, v1.1*, 5 May 2009.
- [CDDEP2009] NEHTA, *Clinical Document Delivery Endpoint Specification, v1.0*, 31 March, 2009.