

nehta

Service Instance Locator

Architecture

Version 1.1 — 1 December 2008

Release

National E-Health Transition Authority Ltd

Level 25

56 Pitt Street

Sydney, NSW, 2000

Australia.

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 © 2008, 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

Table of contents	iii
Document information	v
Change history	v
1 Executive Overview	1
1.1 Purpose	1
1.2 Implementation	1
1.3 Document Exchange Interactions	1
1.4 Deployment	1
1.5 Usage	1
2 Preface	2
2.1 Document Purpose	2
2.2 Intended Audience	2
2.3 Definitions, Acronyms, and Abbreviations	2
2.4 References and Related Documents	3
3 Introduction	4
3.1 Solution Overview	4
3.2 Solution Scope	4
3.2.1 Solutions Out of Scope	4
3.3 Solution Goals and Objective	4
3.4 Assumptions and Dependencies	5
4 Solution States	6
4.1 Today	6
4.1.1 Healthcare Provider Directories	6
4.2 Tomorrow	6
4.2.1 Service Directory Providers	6
4.3 Future	7
4.3.1 Non-Healthcare Service Providers	7
4.3.2 Deployment	7
4.3.3 Non-Web Services	7
5 Business Perspective	8
5.1 Requirements	8
5.2 Document Exchange Scenarios	8
5.2.1 Retrieve	8
5.2.2 Notify and Retrieve	9
5.2.3 Deliver	10
5.2.4 Deliver and Notify	10
5.2.5 Acknowledge	12
5.2.6 Caching of Interactions	12
5.3 SIL Community	16
6 Information Perspective	18
6.1 Requirements	18
6.2 Services and Provider Directories	18
6.3 Interaction Data Structure	18
6.3.1 Interaction	19
6.3.2 Interaction Identifier URIs	20
6.3.3 Service category Identifiers	20
6.3.4 Target Identifiers	20
6.4 Interaction Role	20
6.4.1 Role Name	21

6.4.2	Service Provider.....	21
6.4.3	Service Interface	21
6.4.4	Service Endpoint.....	21
6.5	Qualified Certificate Reference	21
6.5.1	Structure	22
6.6	Interaction Request	23
6.6.1	Interaction Identifier	23
6.7	Summary of Information Types	23
7	Technical Perspective	24
7.1	Requirements.....	24
7.2	Lookup Package	24
7.2.1	Operation listInteractions	24
7.2.2	Operation validateInteraction.....	25
7.2.3	Error Code	25
7.2.4	Security Considerations	25
7.3	Publish Package.....	26
7.3.1	Operation addInteraction	26
7.3.2	Operation removeInteraction	26
7.3.3	Return Codes	26
7.3.4	Error Code	27
7.3.5	Security Considerations	27
7.3.6	Non-standardized Operations.....	28
8	Enabler Dependencies	29
8.1	UHI	29
8.1.1	SIL Bootstrap Reference	29
8.2	NASH.....	29
8.2.1	Single X.509 Certificate per HPIO	29
8.2.2	Multiple X.509 Certificates per HPI.....	29
8.2.3	Dual Usage	30
8.2.4	Certification Authority.....	30
	Appendix A References.....	31

Document information

Change history

Version	Date	Comments
1.0 draft	2008-09-01	Draft for review
1.1	2008-12-01	Release

This page is intentionally left blank

1 Executive Overview

1.1 Purpose

A Service Instance Locator (SIL) is a simple directory of technical services for message exchange. SIL allows a client in the e-health community to locate various electronic services offered by healthcare provider organisations.

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

Information required to perform a SIL lookup includes the target healthcare organisation and kind of message.

1.2 Implementation

SILs are implemented as Web services. A lookup operation returns a set of structures containing Web service endpoints to perform document transfers.

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

1.3 Document Exchange Interactions

Different modes or *interactions* are defined to transfer documents. Core interactions are *deliver* (upload) and *retrieve* (download).

A delivery service may be hosted by a third party on behalf of a healthcare provider. In such a case, a notification may have to be sent to a healthcare provider service after a document is transferred.

Consequently, the structure returned by SIL lookup contains information on how to use transfer services. An interaction comprises one or more *roles*, where each role has an associated service. It is not possible to send a clinical document to a *notify* service or a notification to a *deliver* service.

1.4 Deployment

Every healthcare organisation may have one associated SIL, but the same SIL instance may be shared by more than one organisation. Therefore, a single SIL could be deployed as a central service; a small set of SILs could be deployed as regional services, or a large set of SILs could comprise a fully distributed deployment model.

1.5 Usage

Use of a SIL service is only necessary if a document provider (e.g. pathology laboratory) has no knowledge of how to transfer a clinical document (e.g. pathology report) to a specific recipient (e.g. medical clinic). If a laboratory has previously resolved a certain GP's document exchange services through that GP's SIL, there is no need to repeat the lookup.

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

2 Preface

2.1 Document Purpose

This document describes the architecture for the Service Instance Locator (SIL). The architecture conforms to anticipated business scenarios and requirements, referenced in section 2.4. Architecture perspectives conform to the NEHTA Interoperability Framework [IF2007].

2.2 Intended Audience

This document may be read by:

- **Solution Architects and System Analysts**, for the purpose of understanding SIL as a key piece of e-health infrastructure.
- **SIL developers**, for the purpose of understanding the interfaces to be exposed by a SIL instance, and to become familiar with the proposed usage.
- **Service developers**, for the purpose of understanding the relationship of document transfer services with SIL references.
- **NEHTA Work Package Collaborators**, for the purpose of understanding SIL 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 e-health 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.
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
IHI	Individual Healthcare Identifier
Interaction	Pattern of communication whereby a target obtains its document – corresponds to the patterns outlined in [CPIS2008].
Interaction Role	Function within an interaction. Roles are played by healthcare providers or intermediaries.
NASH	National Authentication Service for Health – NASH will be a CA for e-health technical service providers.

Service	In this document the term service generally refers to a technical service as per [IF2007]. 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 work 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)
SIL	Service Instance Locator
Source	Document suppliers / compilers. For clinical documents a source is a healthcare organisation, e.g. pathology laboratory.
Target	The final destination (intended recipient) of a document. For clinical documents a target is a healthcare provider organisation, e.g. medical clinic.
UHI	Unique Healthcare Identifier (HPIO, HPID, or IHI)
UHI Service	Proposed National UHI Web service

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 [SILR2008].

3 Introduction

3.1 Solution Overview

Service Instance Locator (SIL) will be a key piece of the National E-Health Infrastructure. An application attempting to establish communications with some service will use the SIL to find the service implementation.

SIL data structures conform to the following hierarchy.

- HPIO
 - Interaction
 - Interaction Role
 - Provider Identifier
 - Interface
 - Binding
 - Certificate

3.2 Solution Scope

Services resolvable through the SIL are intended to facilitate the exchange of clinical documents. However, the solution does not preclude other scenarios that could be realised by extending interactions and associated semantics.

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

3.2.1 Solutions Out of Scope

SIL is not designed to be a general purpose directory. It will not support searching based on extensible properties. An entity using the SIL needs to know the HPIO of the healthcare provider in advance, as well as the kind of service it want 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 SIL design.

3.3 Solution Goals and Objective

Broad design goals are listed below. They conform to requirements outlined in [SILR2008].

1. SILs are implemented as Web services.
2. Services can be resolved for a specific HPIO (or other kind of entity).
 - a. Services can be resolved for specific service types.
 - b. Services can be resolved for specific interactions.
 - i. Services can be resolved for specific Interaction Roles within interactions.
 - ii. The service provider can be resolved.
 - 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 SIL interfaces.
4. Interactions can be extended without the need to change SIL interfaces.
5. A standard SIL lookup interface will be defined using WSDL.
6. A standard SIL update interface will be defined using WSDL.

3.4 Assumptions and Dependencies

Some of the assumptions below translate into explicit requirements, see [SILR2008].

1. At a future time a national UHI service will be implemented providing a mapping from HPIO to associated SIL instance.
 - a. If the future UHI service does not include such a feature it will be necessary to provide another mechanism to bootstrap the service location process.
2. At a future time the NASH will be implemented.
 - a. NASH will act as an e-health certification authority (CA).
 - b. NASH allows resolution of X.509 certificate references associated with a HPIO.
3. NEHTA work packages such as Pathology Reporting and Discharge Summary will rely on SIL to resolve document exchange services.
4. Most SIL data is maintained from the perspective of information consumers.
 - a. Healthcare organisations aiming to receive documents will be responsible for keeping SIL data up to date.
 - b. Healthcare providers wishing to send documents need to lookup the recipient (target) SIL prior to sending a document for the first time.
 - c. Some interactions (presently *retrieve*) may be organised from the perspective of information suppliers.
5. Services referenced through a SIL may be outsourced by healthcare organisations.
 - a. Implementations can be outsourced.
 - b. Hosting can be outsourced.
6. Service information resolved using a SIL may be reused without resorting to subsequent SIL lookups.
7. Service clients will require a means to check the validity of service endpoint bindings.
 - a. SIL will provide an operation to support validity checking.

4 Solution States

4.1 Today

SIL 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 work packages. Pathology Reporting (see [PRRPES2008]) and Discharge Summary packages will rely on a SIL to resolve services. Currently the UHI service is unavailable, so an alternate means will be necessary to bootstrap SIL references.

4.1.1 Healthcare Provider Directories

In the absence of a standard method of obtaining endpoints, some healthcare providers have devised directory solutions. These organisations maintain location information for document exchange to partner healthcare providers. It appears such organisations use email as the primary means to transfer documents.

Aside from endpoint resolution, such directories include the information equivalent to interactions and document categories. 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 and maintenance. As the number of recipients increases, it becomes more difficult 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, potential exists that one of the senders will fail to update their directory.

4.2 Tomorrow

It is anticipated that NEHTA will construct and maintain a SIL service to support initial work package implementations. In the longer term the UHI service will be available. A UHI record should contain endpoint and certificate information for the HPIO's SIL (see 8.1).

Certificate references may refer alternately to certificates signed by the NASH root CA or a possible intermediary CA.

4.2.1 Service Directory Providers

SIL places the burden of maintenance on document recipients rather than document senders. Using such a scheme should improve consistency because only one update would be required when a service's details change. For example, when the target HPIO switches a service to a new host address, only one SIL requires an update.

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

In time, location information in sub-SILs may become unnecessary, while demographic information not required by a SIL continues. Note that healthcare providers may still use intermediary organisations to exchange documents, using email or some other non-Web service means. Therefore, existing directories may persist well into the future.

4.3 Future

In the long term, SIL will be a natural part of client workflow for document exchanges. If service endpoints have not been resolved, the SIL will need to be consulted. In the infrequent event that a service fails because it has been decommissioned, SIL will be used to determine that the client reference has become invalid. It will then be used to refresh the reference with another lookup.

It is envisioned that a UHI record will provide the starting point to obtain SIL 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 Non-Healthcare Service Providers

In addition to providing HPIOs, there is also a need to provide identifiers for non-healthcare organisations that maintain services. Many healthcare providers will choose to outsource their services. It is a SIL requirement that these organisations are identified. At the time of writing, it is unclear what standard approach to this issue will emerge, but it is thought a limiting form of HPIO will be issued by the UHI service.

4.3.2 Deployment

It is difficult to foretell exactly how SIL instances will be deployed in the future. NEHTA intends to allow the market to decide SIL deployment.

There are two primary scenarios.

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

It is more probable that SIL implementations will be distributed. A hospital will perhaps host one SIL, associated with its own and subsidiary HPIOs. Several GPs may share an outsourced SIL instance. Larger clinics may choose to host their own SIL. Pathology laboratories could host their own SIL or have it hosted by an IT service provider. Although the distributed scenario may appear chaotic, because records are only updated in one SIL 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 SIL instance. On the other hand, if a central SIL were to gain acceptance in the marketplace, there would be no need for the UHI service to contain an operation returning SIL references.

4.3.3 Non-Web Services

NEHTA standards are currently predicated on Web-services. It is somewhat problematic to transfer documents containing binary data, e.g. CT scan images, using a Web service, especially when they must be encrypted. In the future, other mechanisms/protocols (e.g. FTPS, SFTP) may be returned by a SIL to exchange non-XML document types.

5 Business Perspective

5.1 Requirements

Business requirements are outlined in [SILR2008]. SIL primarily exists to determine how to transfer documents between source and target healthcare providers, and to resolve service endpoints to accomplish the transfers.

5.2 Document Exchange Scenarios

SIL supports interactions in line with patterns outlined in [CPIS2008]. To exchange a clinical document from a source to a target, these are:

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

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 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. The HPIO would be obtained from the HPIO 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 the final recipient (if required) by out-of-bands mechanisms.

5.2.1 Retrieve

Retrieve is the simplest of interaction patterns from the target's perspective because there is no need to host a Web service. Retrieve is a service hosted by a document source.

A *retrieve* interaction by itself could be impractical. There are three reasons for this.

1. A target cannot always know who provides a required document and when the document is due. Suppose *ABC*'s patient *PAT* is discharged from hospital *HOS*. The discharge summary is required by *ABC* but the interaction does not provide a means of communicating this fact. (However, *ABC* may.)
2. *ABC* cannot ensure that any document source using its SIL 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 a *retrieve* interaction, but can offer *notify and retrieve* interactions. When they do, they should also support an alternate interaction such as *deliver*, unless they have come to an arrangement with every source

for a particular service category, such as pathology results reports. Sources can always offer *retrieve*, which means they must provide a download service.

5.2.2 Notify and Retrieve

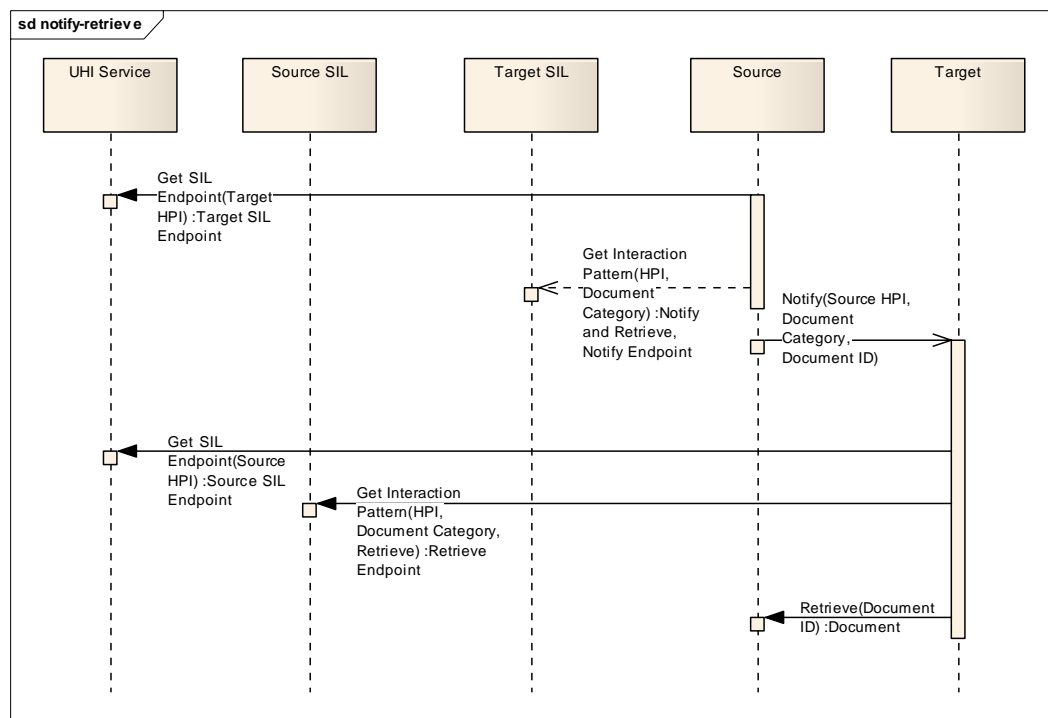


Figure 1 Notify and Retrieve

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

SIL 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 and Retrieve* 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 SIL endpoint of the source through the UHI service. It uses the source SIL 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.

As with any interaction, the target must send an acknowledgement for every document. See 5.2.5.

5.2.3 Deliver

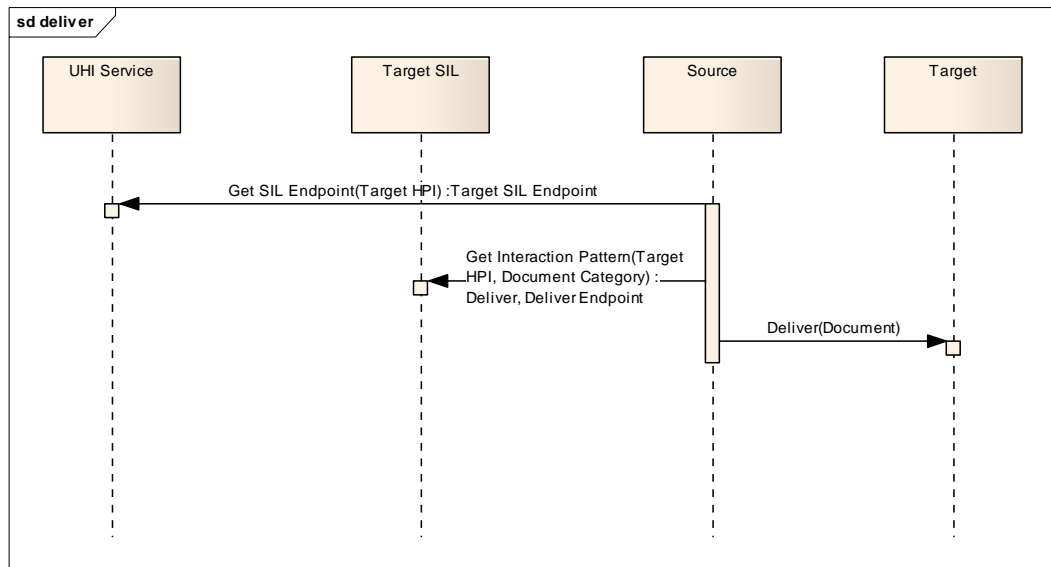


Figure 2 Deliver

Deliver is the simplest interaction. Firstly, the UHI service is used to resolve the target SIL endpoint. Secondly, the target SIL is used to lookup supported interactions. These steps are the same regardless of which interaction is chosen.

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.

As with any interaction, the target must send an acknowledgement for every document. See 5.2.5.

5.2.4 Deliver and Notify

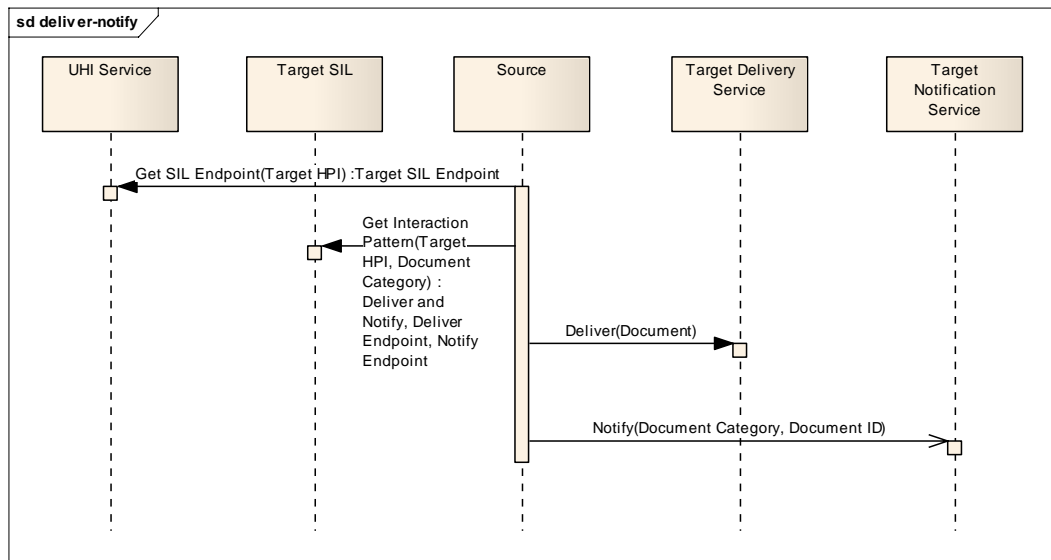


Figure 3 Deliver and Notify

Delivery of the document is achieved by invoking a Web service exactly as in the *deliver* interaction. Further, another Web service is invoked after successful document upload.

This interaction is important in situations where an upload service is maintained by a third party organisation on behalf of a healthcare provider. After the source transfers the document it notifies the actual target that the document transfer is complete. Since the document is now stored by the third

party, the target requires some out of band mechanism to effect the final transfer.

Note however, that the document will be encrypted for the target according to [XSP2008]. A wrapper document will have been encrypted for the service provider as per **Error! Reference source not found.**

Deliver and Notify also applies in situations where a target organisation defers processing of clinical documents but handles notifications as they arrive. There are other scenarios that can be applied equally well.

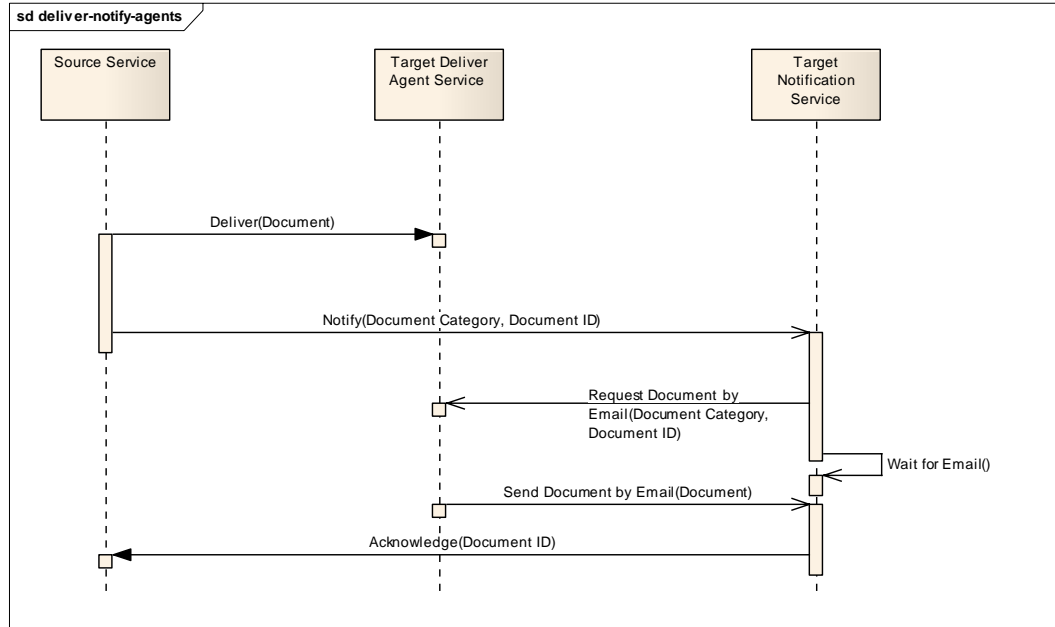


Figure 4 Deliver and Notify with Agent Request/Response

Figure 4 illustrates a sequence where the notification process acts on the notification by soliciting delivery from its agent by email, and waiting. The agent responds, delivering the document via email.

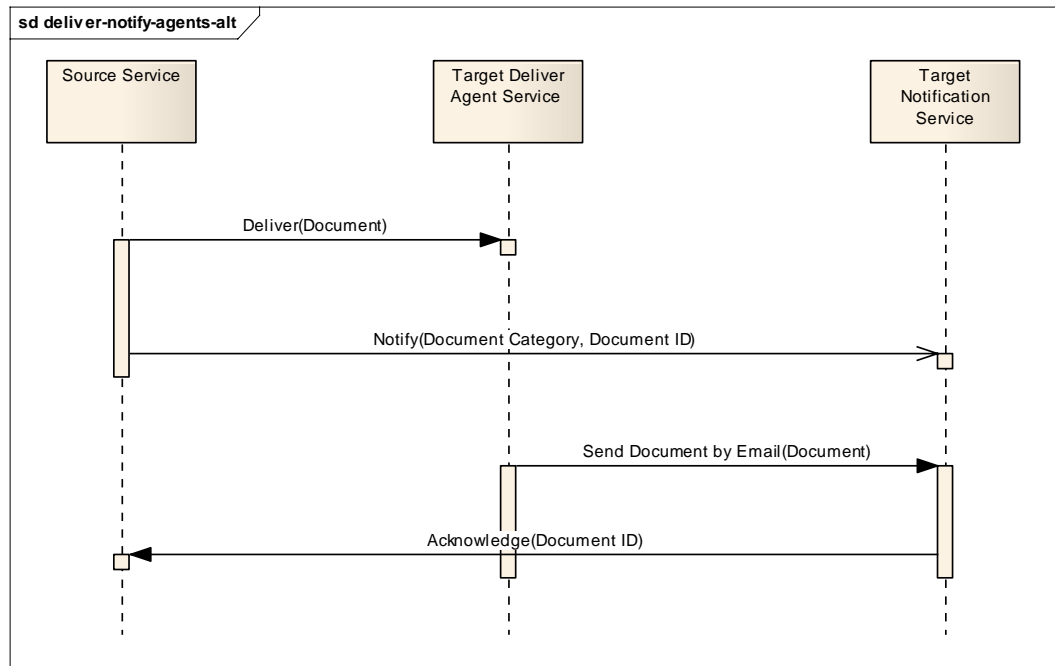


Figure 5 Deliver and Notify with Agent Unsolicited Delivery

Figure 5 illustrates a slightly different scenario, where the agent sends the document asynchronously (without waiting for a request) sometime after it is received.

5.2.5 Acknowledge

Although only Figure 4 and Figure 5 show target acknowledgement, an acknowledgement will be required in all cases. Acknowledgements are intended to alert the source that a target really has 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 work 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 able to be processed.

SIL can be used to resolve the source supported endpoint for document acknowledgement. The sequence is the same as for the *deliver* interaction since the acknowledgement can itself be regarded as a document. The interaction identifier can be supplied to the SIL lookup.

This is illustrated by Figure 6.

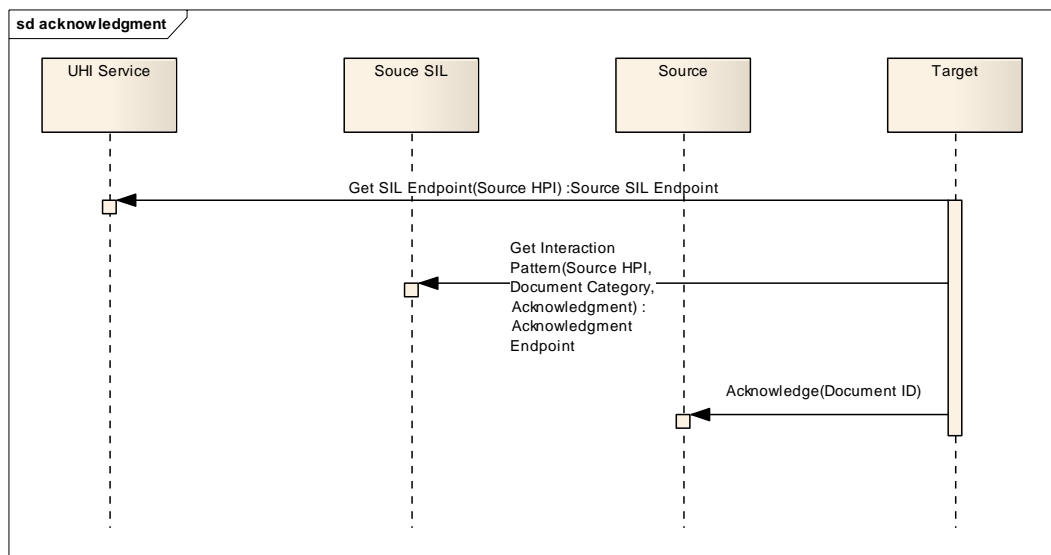


Figure 6 Acknowledge Interaction

5.2.6 Caching of Interactions

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

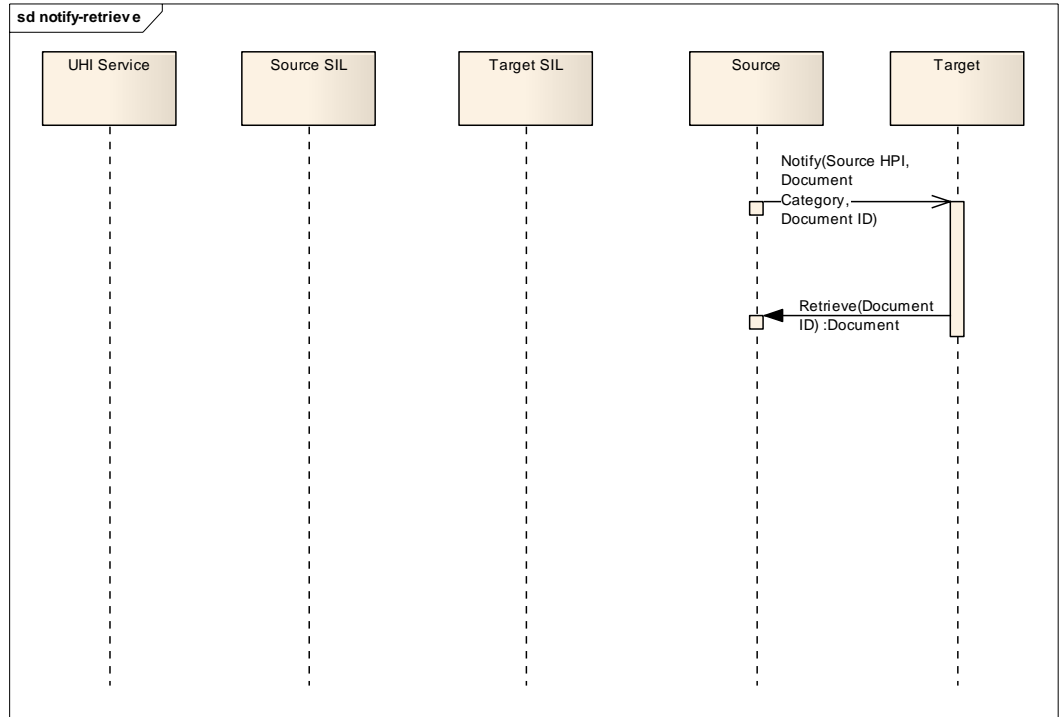


Figure 7 Notify and Retrieve without Lookup

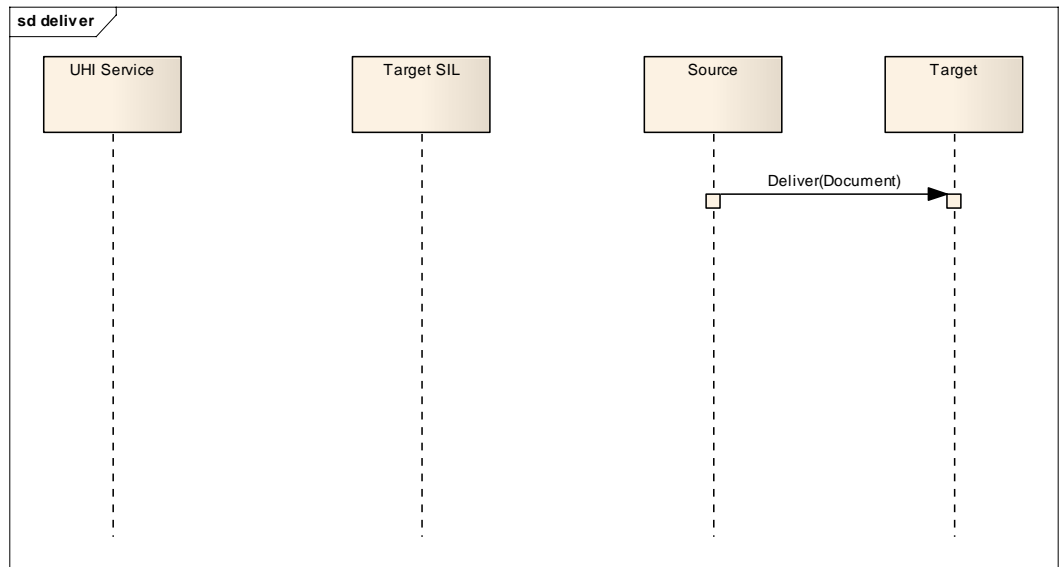


Figure 8 Deliver without Lookup

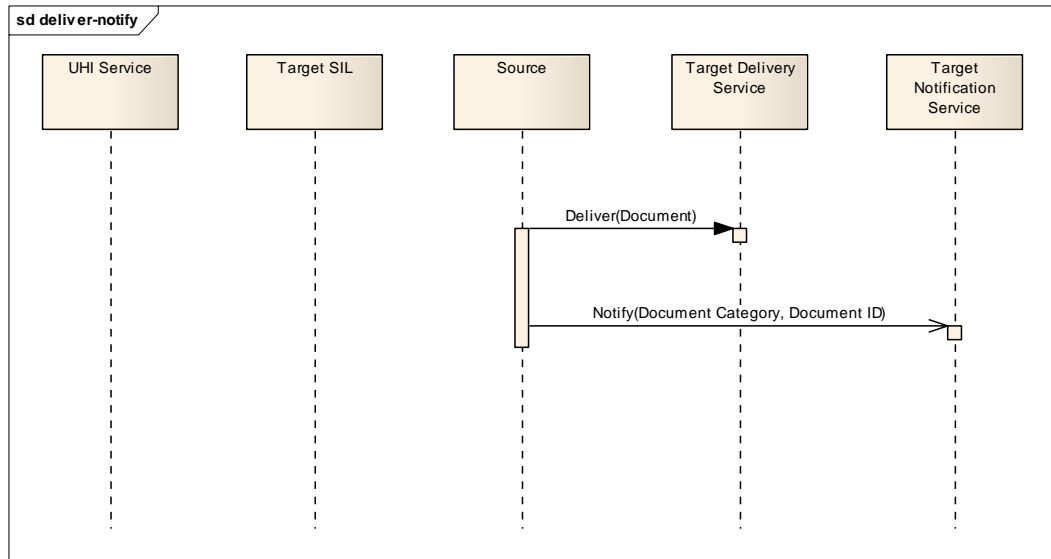


Figure 9 Deliver and Notify without Lookup

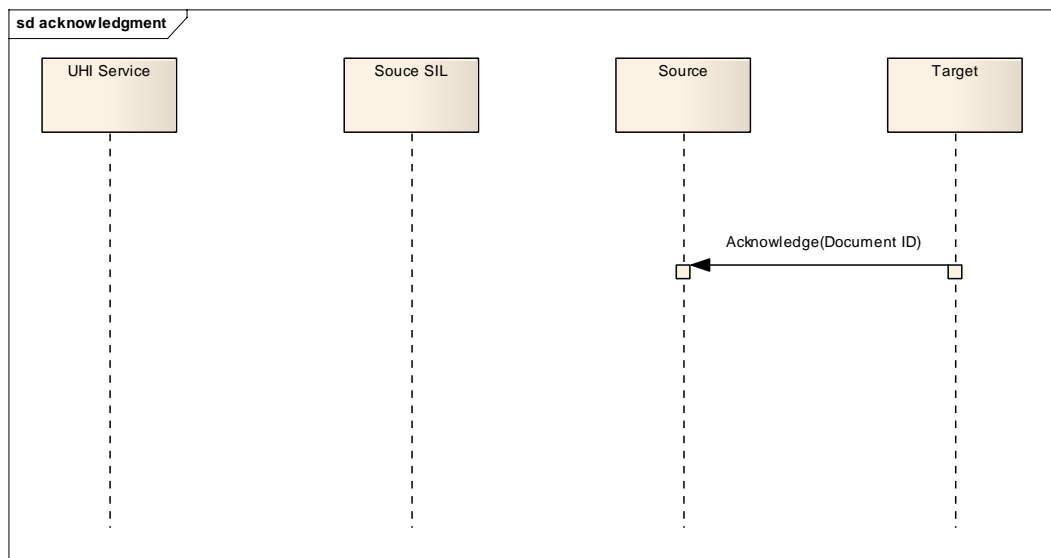


Figure 10 Acknowledge without Lookup

5.2.6.1 Failure Situations

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

SIL endpoints can also be cached. If a failed service runs on the same host as the SIL, the operation to verify the interaction is also likely to fail. One possible cause may be that the SIL 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 SIL endpoint matches the cached value.

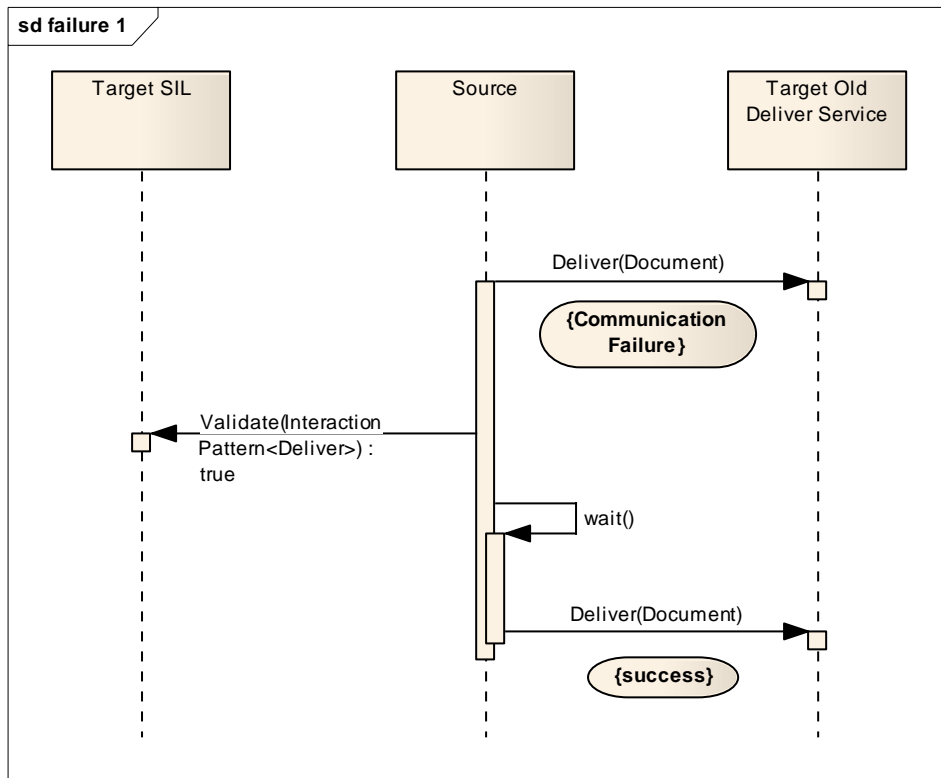


Figure 11 Temporary Failure

In Figure 11 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 SIL 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.

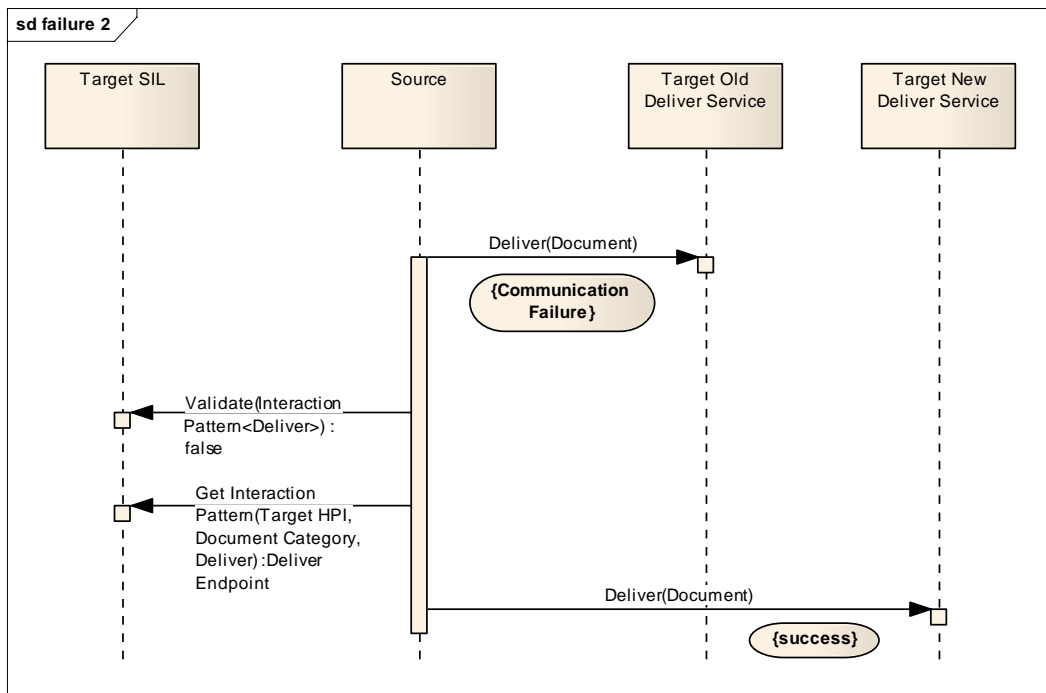


Figure 12 Failure Due to Service Update

Figure 12 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 SIL, *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 SIL Community

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

1. There may only be one standard SIL endpoint associated with every HPIO.
2. Whenever a service is added or removed from a provider directory, the corresponding SIL must be updated to reflect the current state.

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 13.

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 SIL.

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

The ability to develop common workflows in which provider directories publish their changes to a SIL is the key reason an update interface has been specified. See 7.3.5.2.

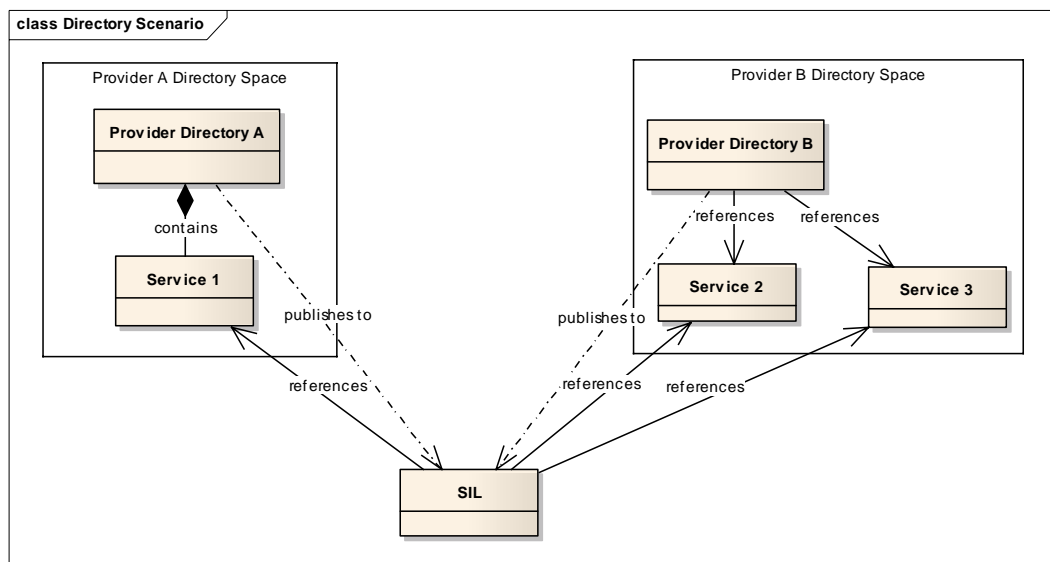


Figure 13 Provider Directory Example Deployment

Figure 14 reflects the relationships among healthcare providers, SIL providers and implementations, local provider directories, the UHI service, and SIL data.

A HPIO can be associated with only one SIL, although several HPIOs can share a single SIL instance.

There is a global UHI service instance. It will have operation(s) to obtain healthcare provider records. Each record contains the associated SIL endpoint.

Provider directories may be associated with many healthcare providers. If they are, they need to update interaction and endpoint information against the relevant SIL for each HPIO they maintain.

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

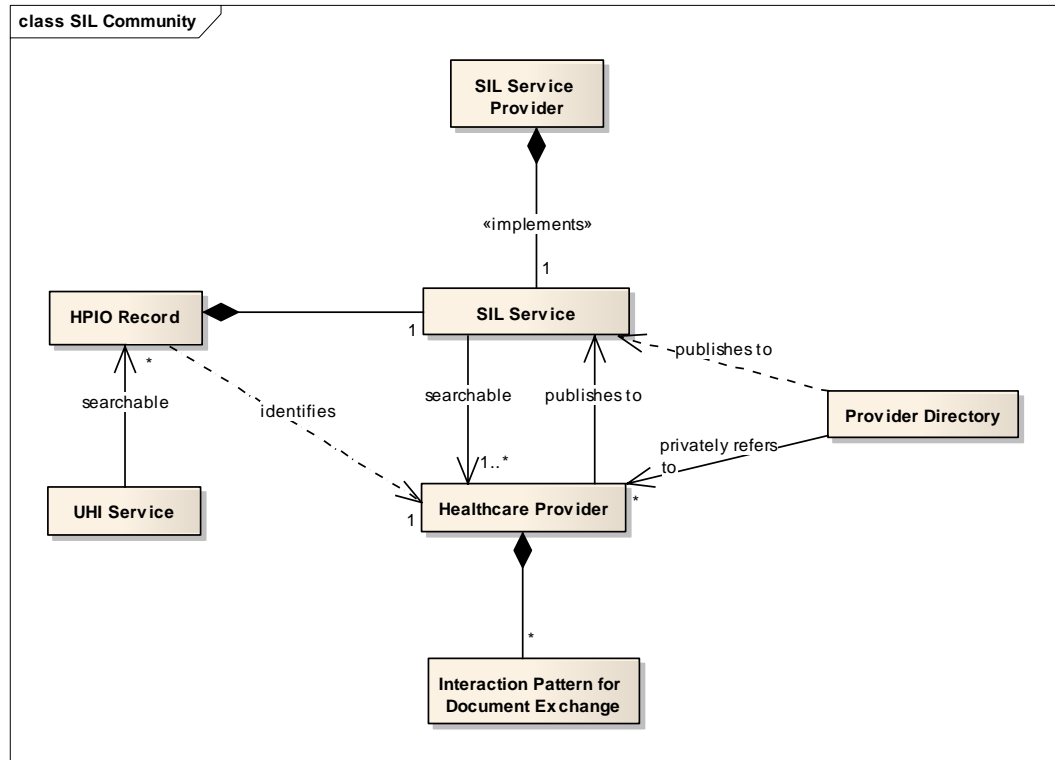


Figure 14 SIL Community Model (Services Omitted)

6 Information Perspective

6.1 Requirements

Informational requirements are outlined in [SILR2008]. Primarily, SIL data structures model interactions for document exchange. A SIL lookup returns zero or more interactions supported for a particular service category.

6.2 Services and Provider Directories

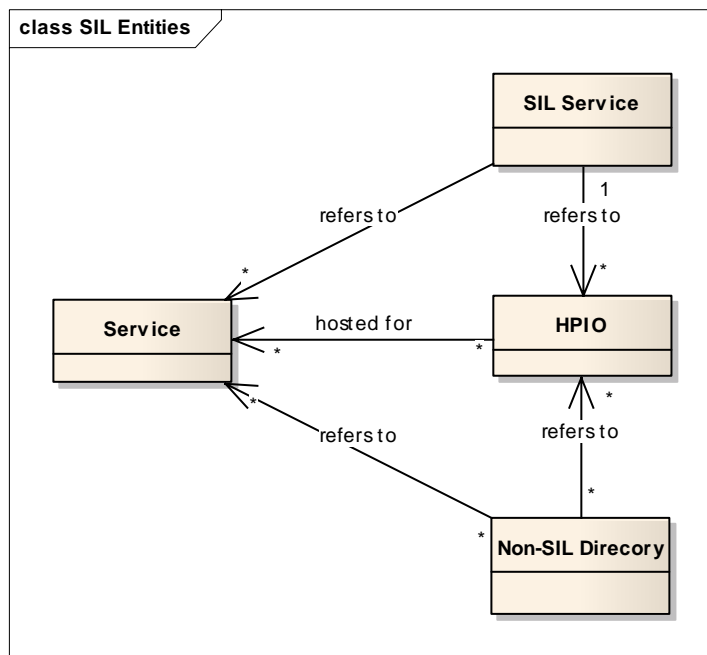


Figure 15 SIL Services and Provider Directories (Interactions omitted)

Services are Web services that handle document exchange. Figure 15 illustrates the relationship of HPIO, SIL, provider directories and services. Interactions are omitted.

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 **Error! Reference source not found.** and [PRRPES2008] imply no restriction on multiple healthcare providers using the same service for document exchange.

6.3 Interaction Data Structure

Interactions in Figure 14 are shown as containments because an interaction is specific to a HPIO and its existence contingent on that of the HPIO. More correctly, from the perspective of SIL information, interactions contain Interaction Roles which aggregate a service. See Figure 16.

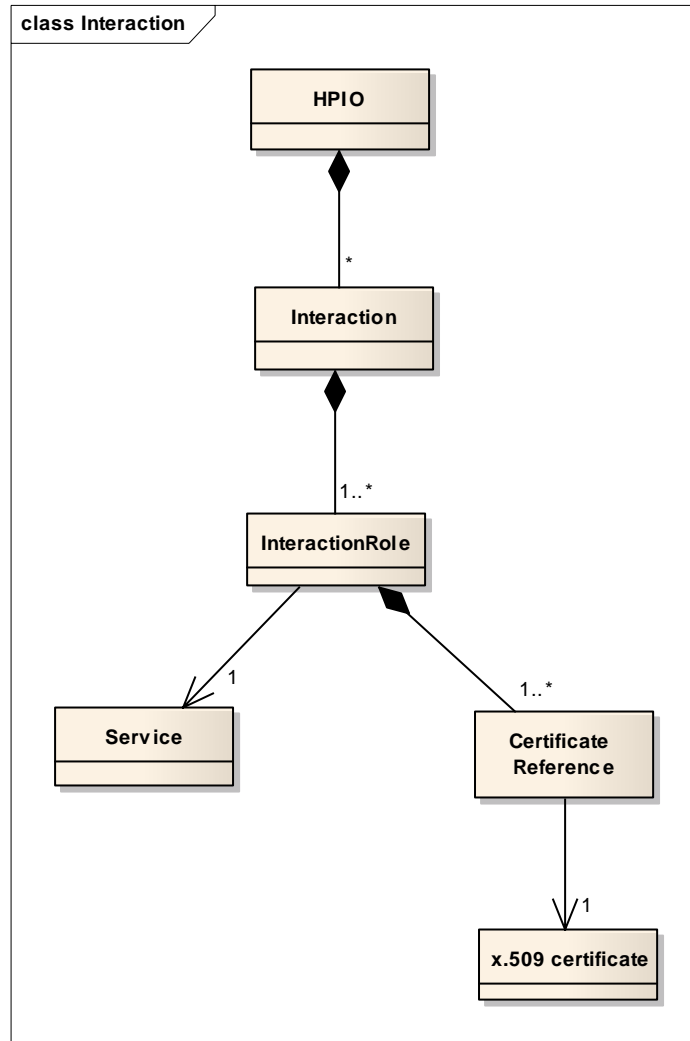


Figure 16 Interaction Structure

Although a HPIO is shown as having zero or more interactions, it would be useless for a HPIO with no interactions to be present in a SIL. Interactions need at least one role. *Retrieve*, *deliver*, *notify* and *retrieve*, and *acknowledge* interactions have one role. *Notify* and *retrieve* has a role corresponding to *notify*, but a target role is not needed for the *retrieve* because the target is a client. *Deliver* and *notify* has two roles.

No interaction with more than two roles is currently defined. The unlimited cardinality allows for future interactions which may contain more than two roles.

Each role refers to exactly one service. It also refers to one or more certificate references. In line with [XSP2008], at least one certificate will be required for use with a service.

In time, the certificate can be obtained from the NASH service using the reference. Multiple certificates could be issued for any given HPIO; the certificate reference specifies which to use for the indicated service (see 6.5).

6.3.1 Interaction

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

- `xsd:complexType: Interaction`
 - `xsd:anyURI: interaction`
 - `xsd:anyURI: serviceCategory`
 - `xsd:anyURI: target`

- Role: role[1..*]

Attribute `interaction` identifies the interaction type.

Attribute `serviceCategory` identifies the kind of document to be exchanged.

Attribute `target` is a qualified name containing the HPIO.

Element `role` is a sequence of Roles (see 6.4).

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

6.3.2 Interaction Identifier URIs

`Interaction::Interaction` can have the following values (for pathology results reporting):

- `http://ns.nehta.gov.au/Pth/RR/SR/Scenario/Const/Deliver/1.0`
- `http://ns.nehta.gov.au/Pth/RR/SR/Scenario/Const/DeliverAndNotify/1.0`
- `http://ns.nehta.gov.au/Pth/RR/SR/Scenario/Const/NotifyAndRetrieve/1.0`
- `http://ns.nehta.gov.au/Pth/RR/SR/Scenario/Const/Retrieve/1.0`
- `http://ns.nehta.gov.au/Pth/RR/SRAck/Scenario/Const/Deliver/1.0`

This list will be extended for future NEHTA work package releases.

6.3.3 Service category Identifiers

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

- `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`

6.3.4 Target Identifiers

Target Identifiers are qualified identifiers that include the HPIO assigned by the UHI. An example is:

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

6.4 Interaction Role

Roles are specified in an XML schema. Its structure is:

- `xsd:complexType: InteractionRole`
 - `xsd:anyURI: roleName`
 - `xsd:anyURI: serviceProvider`
 - `xsd:anyURI: serviceInterface`
 - `xsd:anyURI: serviceEndpoint`
 - `QualifiedCertRef: qualifiedCertRef[1..*]`

Attribute `roleName` identifies the role within the parent interaction.

Element `serviceInterface` identifies the interface for the service (service type). In the short term this will be a WSDL type.

Element `serviceEndpoint` is the service location or instance binding for the interface. It will be comprised of protocol, port, and DNS name/IP address.

Element `serviceProvider` is a qualified identifier containing the HPIO that hosts (makes available) the service. Non- healthcare organisations may also host a service. They will probably be identified by some form of HPIO (see 4.3.1).

qualifiedCertRef contains a sequence of X.509 certificate references (see 6.5).

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

6.4.1 Role Name

Currently `InteractionRole::roleName` may have the following values:

- `http://ns.nehta.gov.au/Pth/RR/SR/Role/Const/Consumer/1.0`
- `http://ns.nehta.gov.au/Pth/RR/SR/Role/Const/Supplier/1.0`
- `http://ns.nehta.gov.au/Pth/RR/SRNotification/Role/Const/Consumer/1.0`
- `http://ns.nehta.gov.au/Pth/RR/SRAck/Role/Const/Consumer/1.0`

These may be extended in the future.

6.4.2 Service Provider

Service Providers are qualified identifiers that include a HPIO. If the service provider is not a healthcare provider, but an outsourced organisation, it is thought that the identifier will be a special form of HPIO (see 4.3.1). An example value would be similar to:

- `http://ns.nehta.gov.au/Id/Const/UhiHpio/1.0#803600000001555`

6.4.3 Service Interface

Service Interface identifies type of service. Currently, it will always be a WSDL type, but may be expanded to other types in the future. 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`

These correspond to the role names of 6.4.1.

6.4.4 Service Endpoint

Service endpoint is simply the binding address for the interface. Currently HTTP 1.1 will be the protocol, as outlined in **Error! Reference source not found.** Protocols other than HTTP may be used in the future. An example value is:

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

6.5 Qualified Certificate Reference

Qualified certificate reference has been included in the SIL specification to identify certificates associated with particular services. It allows for different methods of referencing an X.509 certificate.

It is anticipated that the primary means of referencing a certificate will be a unique subject serial number. At the time of writing, it is expected to be the hardware device identifier (HDI), an X.509 v3 extension field, present in certificates issued by the UHI service in it guise as CA. Qualified certificate references may have various and extensible reference types, and so anticipate future changes in how X.509 certificates will be identified and retrieved.

6.5.1 Structure

- `xsd:complexType: QualifiedCertRef`
 - `xsd:anyURI: typeQualifier`
 - `xsd:anyURI: useQualifier`
 - `xsd:String: value`

The type qualifier contains the kind of reference. It is anticipated that the most common way to resolve a certificate will be the HDI, a v3 extension present in a UHI issued certificate. It's anticipated that the HDI will be a 16 digit number, similar to the HPIO. An example may be:

`typeQualifier:`

```
http://ns.nehta.gov.au/Qcr/Ref/Const/Hdi/1.0
```

`value:`

```
8034000000009999
```

This informs the SIL query process to lookup the certificate by HDI from the NASH service.

Alternatively, the reference may be a UHI assigned serial number. If this were expressed as a hexadecimal value, it may be similar to:

`typeQualifier:`

```
http://ns.nehta.gov.au/Qcr/Ref/Const/SerialNum/1.0
```

`value:`

```
10C97D56A056
```

The use qualifier contains a URI denoting the allowable certificate usage. It pertains to whether the certificate is used for a payload or transport entity. A transport entity is an entity hosting a service, possibly an intermediary in the document delivery chain. According to [PRRPES2008], documents are encrypted with a symmetric session key for the target, as per [XSP2008]. They are then encrypted with another session key for the intermediary, as per **Error! Reference source not found.** The target's public key is used to encrypt the target's session key, and the transport entity's public key is used to encrypt the transport's session key. If there is no intermediary the payload is still encrypted twice, but both session keys will be encrypted with the target's public key.

Here are the currently allowable values:

`useQualifier:`

```
http://ns.nehta.gov.au/Qcr/Use/PayLoad/Const/KeyEnc/1.0
```

```
http://ns.nehta.gov.au/Qcr/Use/PayLoad/Const/SignKeyEnc/1.0
```

```
http://ns.nehta.gov.au/Qcr/Use/Transport/Const/Sign/1.0
```

```
http://ns.nehta.gov.au/Qcr/Use/Transport/Const/KeyEnc/1.0
```

```
http://ns.nehta.gov.au/Qcr/Use/Transport/Const/SignKeyEnc/1.0
```

```
http://ns.nehta.gov.au/Qcr/Use/PayloadTransport/Const/Sign/1.0
```

```
http://ns.nehta.gov.au/Qcr/Use/PayloadTransport/Const/KeyEnc/1.0
```

```
http://ns.nehta.gov.au/Qcr/Use/PayloadTransport/Const/SignKeyEnc/1.0
```

Certificates which have a "Sign" usage are used to verify signed responses. Such are possible only from the peer to whom the connection is established, so there is no need for signing usage against the payload entity (unless the payload and transport entity is the same, i.e. there is no intermediary).

If there is no intermediary, the use of "PayloadTransport" use qualifiers is recommended.

6.6 Interaction Request

Type `InteractionRequest` is used as input to a SIL lookup operation (see 7.2). Its structure is outlined below:

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

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

6.6.1 Interaction Identifier

Optionally, element(s) `interaction`, if present, correspond to interaction identifiers as described in 6.3.2. If present, these would have the effect of narrowing the search.

6.7 Summary of Information Types

Figure 17 is the class diagram for basic SIL data types. Explanations are given in the sections above.

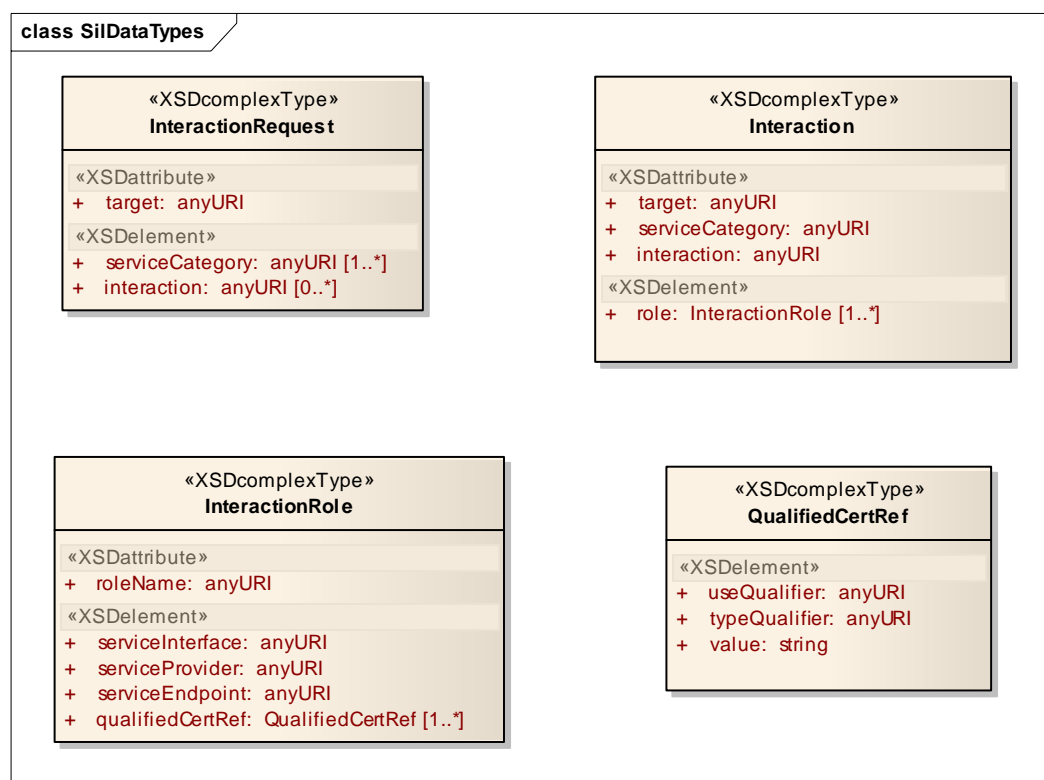


Figure 17 SIL Data Types

7 Technical Perspective

7.1 Requirements

Technical requirements are outlined in [SILR2008]. The interface is logically split into two packages, read and update operations.

7.2 Lookup Package

Figure 18 summarises the lookup or read package. There are two operations, `listInteractions` and `validateInteraction`.

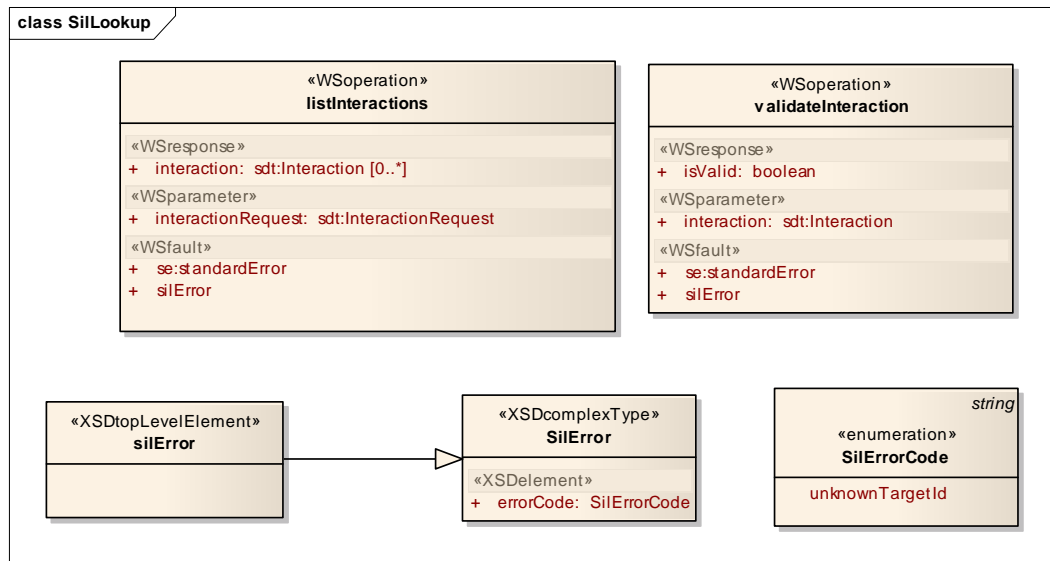


Figure 18 SIL Lookup Package

7.2.1 Operation listInteractions

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

1. If the optional `interaction` (see 6.6.1) is omitted from the request all interactions will be returned for the input HPIO 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. In this case the roles would refer to different service providers and endpoint addresses. For example, two different service providers may host services on behalf of the same healthcare provider for different document sources. If both sources support the *retrieve* interaction, the target would have two different *notify and retrieve* interactions. Alternatively, a target may be moving toward hosting of its own *deliver* service that is presently hosted by an agent. If a transition period was desired to switch to the new service, two *deliver* interactions could temporarily co-exist. 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 SIL 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). It takes a single `Interaction` as input, returning `true` if the interaction is still current, or `false` if the interaction has been removed.

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

7.2.3 Error Code

There is only one SIL-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 SIL is not associated with the supplied HPI. This indicates the client is attempting a request using the wrong SIL instance.

Other faults that may be generated are defined as part of the standard error package defined in **Error! Reference source not found.**. 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 [SILR2008], all SIL records may be obtained by any healthcare provider in the e-health community. Communications between client and server MUST be secured in accordance with **Error! Reference source not found.**

The public key used to verify the digital signature will be extracted from a referenced X.509 certificate signed by a trusted CA (see **Error! Reference source not found.**). It is expected that NASH will issue device certificates for HPIOs.

7.3 Publish Package

Figure 19 summarizes the publish package. There are two operations, `addInteraction` and `removeInteraction`.

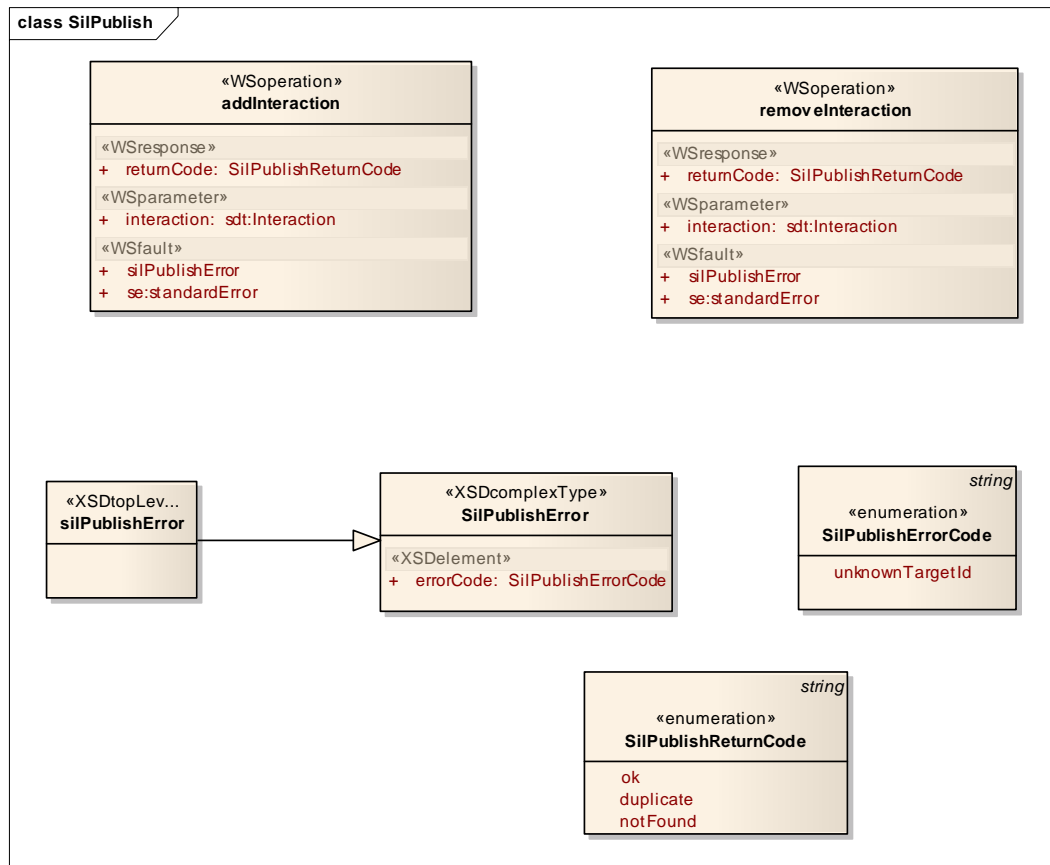


Figure 19 SIL Publish Package

7.3.1 Operation addInteraction

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

7.3.2 Operation removeInteraction

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

7.3.3 Return Codes

`SILPublishReturnCode` 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 (SIL) 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 SIL. This is not considered an error, especially since it is possible that the virtual circuit between client and (SIL) 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 SIL-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 SIL 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 SIL it must already have an association created. The mechanism to accomplish this will not be defined (see 7.3.6).

A valid target MUST be able to update (invoke operations of the publish package) its own SIL instance. SIL updates are permitted by the SIL if the X.509 certificate used to sign the request corresponds to a target certificate issued by a trusted CA against its HPIO.

For read purposes, SIL implementations will have to store the root and/or intermediate certificates of CAs it is prepared to trust. In the short term, test certificates issued by HPIO/NASH should be trusted.

For update purposes, SIL implementations may have to store certificates of the HPIOs it is associated with as well as any additional organisations it is prepared to trust, at least until the NASH service is available. Trust can sometimes be established by out-of-band means, e.g. for external directory providers (see 7.3.5.2).

7.3.5.2 Sub-SIL Provider Update

More problematic is the situation where a provider directory is required to update a SIL to ensure consistency between its own data and the SIL (see 5.3). Even when the NASH comes online, at the time of writing, NEHTA has no authorization model. There is no reference service, either existing or planned, for a SIL instance to determine if an update operation should be permitted.

The only workaround will be for clients (local directory providers) and server (SIL) to arrange trust relationships through an out-of-band mechanism.

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 SIL 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 (even if the mechanism to execute them is not standard) is that a healthcare provider is associated with a SIL instance. There must also be a way to disassociate a provider from a SIL instance. This could occur because a healthcare provider wants to change its SIL instance or because it has ceased practice/operations.

Healthcare to SIL association/disassociation operations are not defined. It is left to individual SIL providers to 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 SIL:

1. HPIOs need to be allocated against healthcare providers for SIL lookups to make sense (see 6.3.4). Until UHI is realized, interim identifiers could be allocated by NEHTA.
2. It has been assumed that the UHI record will contain the SIL endpoint (see 8.1.1). If there is a centralized SIL (see 4.3.2) such an operation would be wasteful, since it would always return the same address. There has to be a least one known address to bootstrap the document exchange process. Presently, the assumption is that there will be two: one for UHI and one for NASH. A centralized SIL could bump the count to three. In the more likely deployment scenario, there would be multiple distributed SIL instances, and SIL location via the UHI is desirable.

8.1.1 SIL Bootstrap Reference

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

1. A SIL endpoint, of type URL. A reference to the interface type is not necessary, since the interface is the same for all SILs.
2. An X.509 certificate reference. The client would expect responses from the SIL to be signed with the public key of this certificate. This could be a structure similar to the `QualifiedCertRef` (see 6.5). It would not require a usage attribute, since the only usage necessary for SIL communications is signing.

A reference to the SIL service provider is likewise not required if the certificate is supplied. On the other hand, these considerations may depend on whether NASH will support issuing more than one certificate per healthcare provider. It seems almost certain that it will.

8.2 NASH

8.2.1 Single X.509 Certificate per HPIO

References captured by `QualifiedCertRef` (see 6.5) will depend on the NASH implementation. If NASH issues only one certificate for a particular healthcare organisation, the certificate must support usages of at least signing and key encipherment.

8.2.2 Multiple X.509 Certificates per HPI

It is extremely likely that the NASH will permit multiple certificates for a healthcare provider, including soft certificates and certificates contained on smartcards or other mobile devices.

In this case it becomes necessary to know what kind of certificate reference the NASH service will resolve. It is assumed that a v3 extension, hardware device identifier, will be employed for device-specific identifiers. Soft certificates may be identified by the serial number. Any attribute that uniquely identifies the certificate, could be used. Note that in this situation the HPIO itself would not uniquely identify a certificate.

8.2.3 Dual Usage

Currently, [XSPP2008] requires certificates with Signing and Key Encipherment usage. If UHI issues certificates supporting these, it should not be necessary to reference more than one from the `InteractionRole` data structure (see 6.4).

8.2.4 Certification Authority

Most trust issues identified by this document are very simple. They can be resolved by ensuring the certificate(s) used for Web service invocations are signed by a trusted CA. The HPIO/NASH service will constitute the trusted CA in the e-health community in the medium term.

Appendix A References

- [CPIS2008] NEHTA, *Concepts and Patterns for Implementing Services v2.0*, 1 December 2008.
- [IF2007] 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.
- [SILR2008] NEHTA, *SIL Requirements v1.1*, 1 December 2008.
- [WSP2008] NEHTA, *Web Services Profile v3.0*, 1 December 2008.
- [XSPP2008] NEHTA, *XML Secured Payload Profile v1.0*, 1 December 2008.