



Service Instance Locator

Technical Architecture

Version 1.0 draft — 1 September 2008

Draft for comment

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 Introduction	1
1.1 Background	1
1.2 Purpose	1
1.3 Status	1
1.4 Document Map	1
2 Technical Issues	2
2.1 Required Information for Bootstrapping	2
2.2 Requirements on other Infrastructure Services	2
2.2.1 NASH	2
2.2.2 UHI	3
2.3 Web Services Profile	3
2.4 Transaction History	3
3 SIL Faults	4
3.1 Fault: <i>UnknownBusinessService</i>	4
3.2 Fault: <i>DuplicateServiceType</i>	5
3.3 Fault: <i>UnknownPattern</i>	5
3.4 Fault: <i>UnknownVersion</i>	5
3.5 Fault: <i>CannotDeleteOnlyRealisation</i>	5
3.6 Fault: <i>UnknownServiceType</i>	6
3.7 Fault: <i>UnknownTechnicalService</i>	6
3.8 Fault: <i>DuplicateTechnicalService</i>	6
4 Directory Server Publish Interface	7
4.1 Interface: <i>SilPublish</i>	7
4.1.1 Operation: <i>publishBsir</i>	7
4.1.2 Operation: <i>updateBsirTtl</i>	8
4.1.3 Operation: <i>addBsirTechnicalRealisation</i>	9
4.1.4 Operation: <i>deleteBsirTechnicalRealisation</i>	10
4.1.5 Operation: <i>updateBsirRole</i>	11
4.1.6 Operation: <i>deleteBsir</i>	12
4.1.7 Operation: <i>publishTsir</i>	12
4.1.8 Operation: <i>deleteTsir</i>	13
5 Directory Server Lookup Interfaces	14
5.1 Design Rationale	14
5.2 Interface: <i>SILLookup</i>	15
5.2.1 Operation: <i>lookupBsirs</i>	15
5.2.2 Operation: <i>lookupTsirs</i>	16
Appendix A: References	17
Appendix B: XSD	18
B.1 SilDataTypes.xsd	18
B.2 SilFaults.xsd	19
Appendix C: WSDL	21
C.1 SilPublishInterface.wsdl	21
C.2 SilPublish.wsdl	26
C.3 SilLookupInterface.wsdl	29

C.4 SilLookup.wsdl 31

Document information

Change history

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

This page is intentionally left blank.

1 Introduction

1.1 Background

The Service Instance Locator (SIL) has been identified as a piece of key National E-Health Infrastructure. An application that is attempting to locate a business service, and the technical endpoints that realise the goals of that business service, will use the SIL to locate the correct service endpoint.

1.2 Purpose

The Technical Architecture for the Service Instance Locator explains the Web Services specifications for the Service Instance Locator. It also explains how a Web Service implementing the specification can be accessed by a client of an e-health service wishing to locate its endpoint.

1.3 Status

This document is a draft and has been released for comment and feedback purposes.

1.4 Document Map

This diagram represents the relationship between this document and other related specifications.

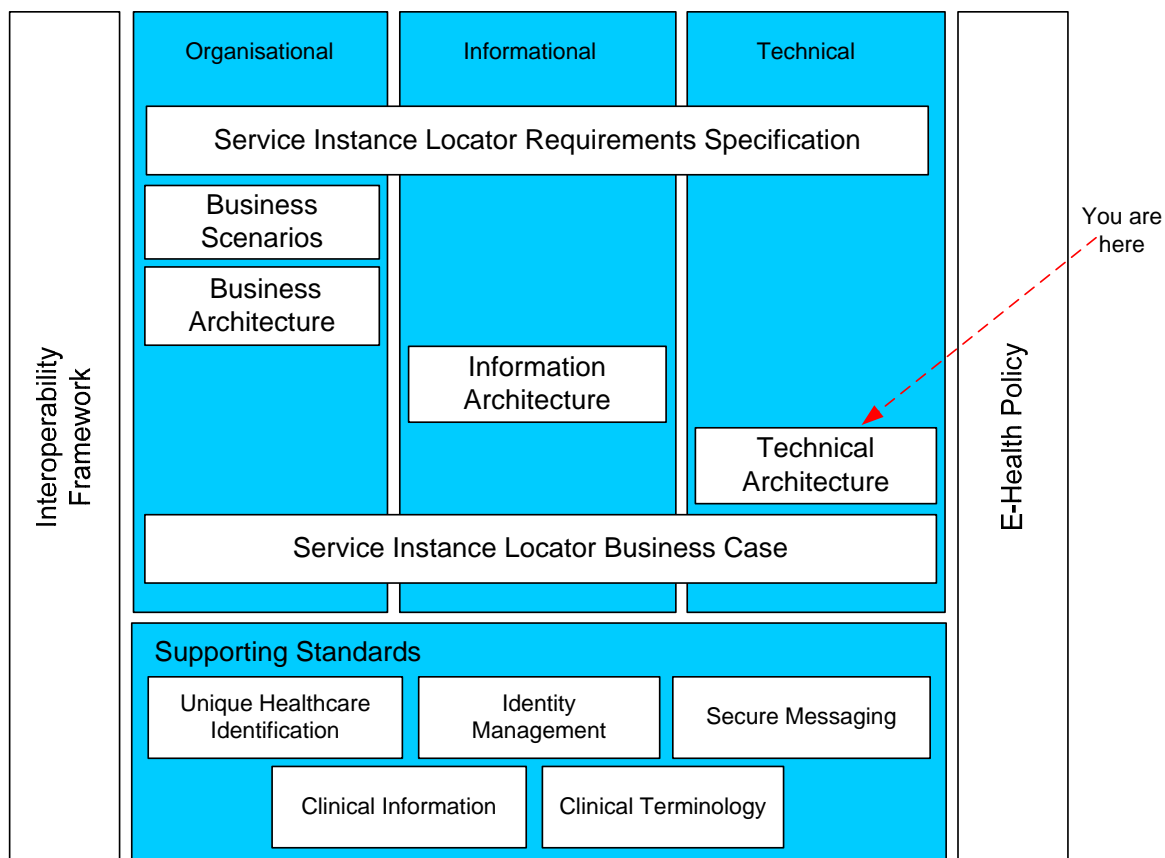


Figure 1: Document Map

2 Technical Issues

2.1 Required Information for Bootstrapping

The Service Instance Locator (SIL) is designed to be implemented as a logical directory of business services and their technical endpoints *supported by a single organisation*. The deployment granularity can be as small as one SIL for each separate organisational identifier, or as large as one SIL for an arbitrarily large collection of organisational identifiers.

The required information to locate the correct SIL for an identified organisation consists of

1. the organisational identifier of the organisation who owns the services which need to be discovered, and
2. a URL where the lookup interface to the SIL for this organisation is located
3. the organisational identifier of the SIL host (if the organisation hosts its own SIL, this is the same as (1) above).
4. a reference to the PKI certificate which the organisation (3) is using to secure communications to the SIL.

Eventually this information will be available from the UHI Service. However, until the UHI service is available, the URL where the lookup interface to the SIL can be located and the organisational identifiers of that organisation and its SIL host will be available only directly from the organisation concerned, using web pages, phone, fax, email etc.

2.2 Requirements on other Infrastructure Services

2.2.1 NASH

In the short term, before the UHI Services are operating, there must be an alignment between the organisational identifiers which are used in the NASH as identifying information for security certificates in a Certificate Directory, and the organisational identifiers used to secure establish web services connections to the SIL. These will also be the identifiers used inside the Service Instance Records. In some cases an organisation identified by a single identifier will have only a single certificate in the NASH, but in other cases there may be several certificates issued for different purposes. Each certificate is uniquely identified by a qualified identifier which can be located at a particular Certificate Directory. The identifier may be a Subject Key Identifier, or some other more long lived identifier, and the qualifier of that qualified id will indicate which attribute of the certificate must be matched. Certificates are also used in the SIL to assist service clients to locate the correct keys for communication with the services published in the SIL.

NOTE: As the approach for the implementation of NASH is in flux at the time of writing, the relationship between SIL and NASH may change as this implementation becomes more concrete.

NOTE: The SIL information model contains a generalised Certificate Reference data structure that allows for any trusted Certificate Directory to be used for the storage and lookup of certificates. This allows the SIL to be implemented before the NASH becomes available.

2.2.2 UHI

In the medium term, after the UHI Services are operating, the organisational identifiers used in both NASH and SIL will migrate to be the identifiers that are allocated by UHI. The use of HPI-Os is limited by policy to be allocated only to organisations that directly provide healthcare services, and may not be issued to organisations which provide technical services to these organisations, such as the hosting of Notification Services, Encrypted Stores, and the SIL itself. These organisations will be issued Service Provider Identifiers, or SPI-Os.

In addition to this dependency on the UHI allocated identifiers, there is requirement that each record pertaining to a HPI-O in the UHI provider directory will have fields that contain:

- the URL where the organisation identified by the HPI-O can be contacted for access to its SIL's Lookup interface,
- the identity of the SIL host (probably an SPI-O),
- a reference to the Certificate that the SIL Host is using to secure communications with the SIL.

2.3 Web Services Profile

The endpoints for SIL interfaces MUST all be for secure Web services, as specified in the "End-to-end Security" profile of the NEHTA Web Services Profile v3.0 specification [NWSP2008].

2.4 Transaction History

The NEHTA Enterprise Architecture generally requires services published by NEHTA to specify how the history of transactions carried out at a service is stored and/or queried. SIL is, however, a service which has the goal of providing the most up to date endpoints for services owned and hosted by particular organisations. As such, the history of the endpoints of services offered is irrelevant to this goal. Furthermore, access to out-of-date endpoint information via a history of Service Instance Records that are published to a SIL may in fact be dangerous, as it is possible (although unlikely) that old endpoints may be left running, and that clinical data may be directed to the wrong service instance. Therefore, there is no attempt to specify what the transaction history of a SIL is, and no mechanism is provided to access this history.

3 SIL Faults

This section documents the faults that can be raised as the result of errors in executing the operations supported by the SIL's Publish and Lookup interfaces.

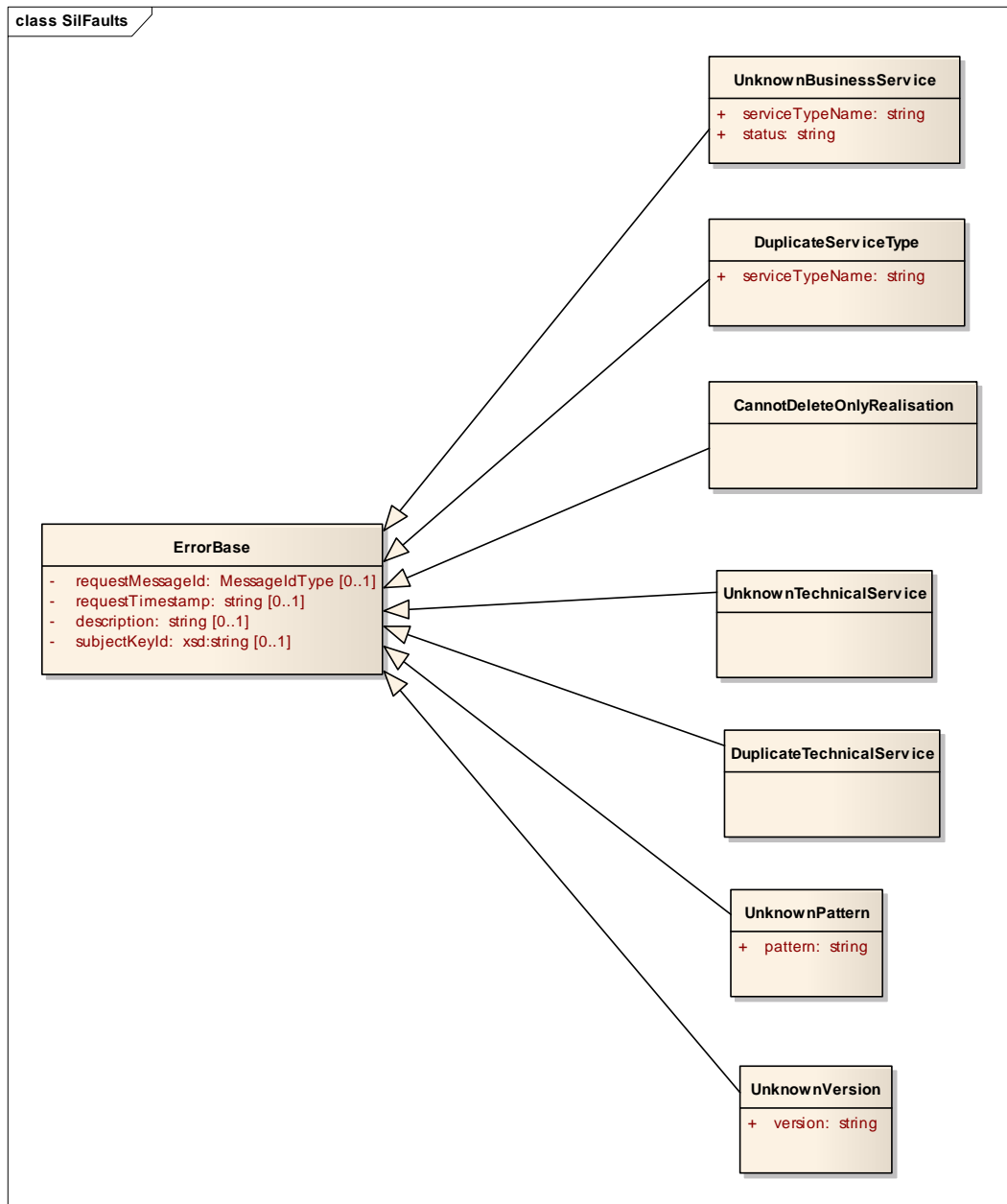


Figure 2: SIL Faults

3.1 Fault: *UnknownBusinessService*

This fault inherits the Secure Messaging Common *ErrorBase* attributes. It also contains the following fields.

Field	Type	M O C	Description
<i>serviceTypeName</i>	<i>string</i>	M	The service type name component of the business service key that was not found in the SIL.

<i>status</i>	<i>Status</i>	M	The status component of the business service key that was not found in the SIL.
---------------	---------------	---	---

3.2 Fault: *DuplicateServiceType*

This fault inherits the Secure Messaging Common *ErrorBase* attributes.

Field	Type	M O C	Description
<i>serviceName</i>	<i>string</i>	M	The service type name component of the business service key that was already found in the SIL.
<i>status</i>	<i>Status</i>	M	The status component of the business service key that was already found in the SID.

3.3 Fault: *UnknownPattern*

This fault inherits the Secure Messaging Common *ErrorBase* attributes. It also contains the following fields.

Field	Type	M O C	Description
<i>patternName</i>	<i>string</i>	M	The technical realisation pattern for a business service which could not be located.

3.4 Fault: *UnknownVersion*

This fault inherits the Secure Messaging Common *ErrorBase* attributes. It also contains the following fields.

Field	Type	M O C	Description
<i>version</i>	<i>string</i>	M	The technical realisation version for a business service which could not be located.

3.5 Fault: *CannotDeleteOnlyRealisation*

This fault inherits the Secure Messaging Common *ErrorBase* attributes. It also contains the following fields: *None*.

Semantics: The BSIR must have at least one Technical Realisation, and the operation that raised this fault is attempting to delete it.

3.6 Fault: *UnknownServiceType*

This fault inherits the Secure Messaging Common *ErrorBase* attributes. It also contains the following fields.

Field	Type	M O C	Description
<i>serviceTypeName</i>	<i>string</i>	M	The service type name that was not found in the SIL.

3.7 Fault: *UnknownTechnicalService*

This fault inherits the Secure Messaging Common *ErrorBase* attributes. It also contains the following fields: *None*

Semantics: It indicates that a technical service instance record could not be found to match the given criteria.

3.8 Fault: *DuplicateTechnicalService*

This fault inherits the Secure Messaging Common *ErrorBase* attributes. It also contains the following fields: *None*.

Semantics: The TSIR cannot be published as there is already a record with the same attributes present in the SIL.

4 Directory Server Publish Interface

The SIL role in the community model of the SIL Business Architecture [SILBUS2008] supports an interface to allow authorised publishers to publish an organisation’s business service and technical service details to the SIL. The Service Publisher role act as a client to this interface of the SIL.

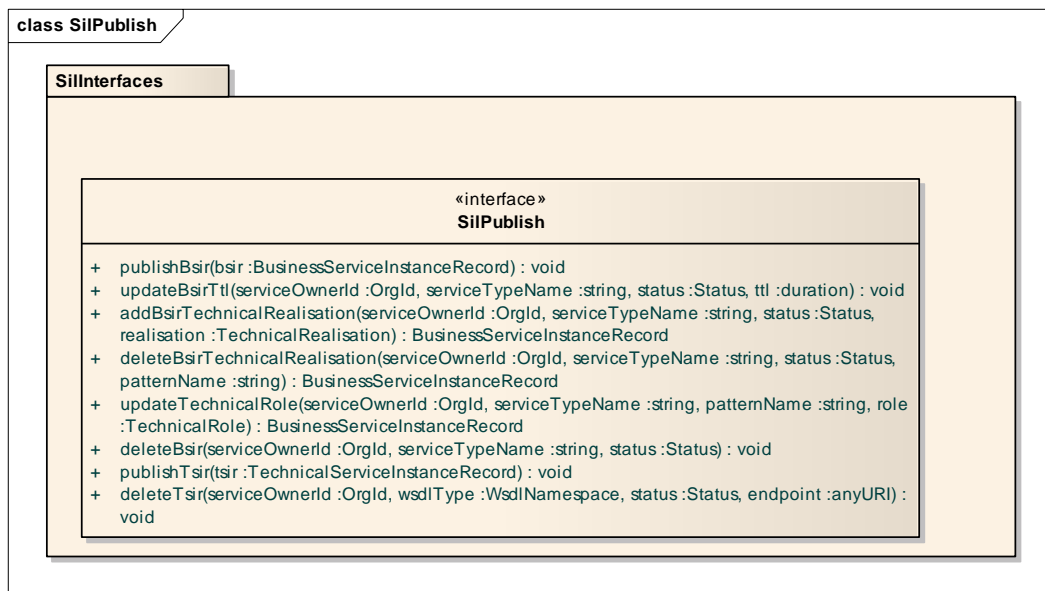


Figure 3: SIL Publication Interface

4.1 Interface: *SilPublish*

This interface supports the publication, updating and deletion of Business Service Instance Records (BSIRs), and Technical Service Instance Records (TSIRs).

As explained in the SIL Information Architecture [SILBIA2008], each BSIR contains one or more Technical Realisations, which have named Technical Roles containing a TSIR Reference for each endpoint required by the Technical Realisation.

A TSIR Reference contains all the information about the SIL in which appropriate TSIRs can be looked up to fulfil a Technical Role in a Technical Realisation of a business service. It also contains all the query parameters needed to do a lookup for a TSIR at that SIL. If the SIL in question is the local SIL to which the BSIR is to be published, then the TSIR (or TSIRs) which is referred to may be published before or after the BSIR, or their endpoint information can be attached to the TSIR Reference, and published at the same time. If the SIL is a remote SIL, then the TSIR(s) must be published separately into that SIL.

4.1.1 Operation: *publishBsr*

4.1.1.1 Parameters

The operation SHALL accept the following parameters.

Parameter	Type	M O C	Description
<i>bsir</i>	<i>BusinessServiceI</i>	M	A new Business Service Instance

	<i>nstanceRecord</i>	Record to be made available to clients of the lookup interfaces by the SIL.
--	----------------------	---

4.1.1.2 Return Value

void

4.1.1.3 Faults

Fault Type	Description
<i>DuplicateServiceType</i>	This fault MUST be raised when the key information in the BSIR argument (<i>serviceName</i> & <i>status</i>) clashes with an already published BSIR.

4.1.1.4 Description

The *publishBsir* operation passes a new BSIR to the SIL for publication. The publisher (invoker of this operation) MUST be authorised to publish on behalf of the organisation given in the *serviceOwner* field of the *bsir* argument.

If the BSIR does not meet any of the error criteria listed with the Faults above, then it will be placed in the SIL and made available for lookup clients.

If the BSIR contains any *TechnicalServiceEndpoints* attached to a *TechnicalServiceRef*, then a TSIR MUST be published to this SIL. All of the fields of the *TechnicalServiceEndpoint* must be copied into the new TSIR. In addition the following fields of the TSIR MUST be populated from the *TechnicalServiceEndpoint's* parent *TechnicalServiceRef* context:

- *wsdlType* – from the same field of the parent *TechnicalServiceRef*
- *status* - from the same field of the parent *TechnicalServiceRef*
- *serviceOwnerId* – from the *serviceOwnerId* field of the *BusinessServiceInstanceRecord* that contains the parent *TechnicalServiceRef*
- *serviceHostId* – MUST be the same as *serviceOwnerId*

4.1.2 Operation: *updateBsirTtl*

4.1.2.1 Parameters

The operation SHALL accept the following parameters.

Parameter	Type	M O C	Description
<i>serviceOwnerId</i>	<i>OrgId</i>	M	Restricts the match of the following two parameters to BSIRs owned by this organisation.
<i>serviceName</i>	<i>string</i>	M	The first part of the key used to match a particular BSIR.
<i>status</i>	<i>string</i>	M	The second part of the key used to match a particular BSIR.
<i>ttl</i>	<i>duration</i>	M	The new ttl value for the identified BSIR.

4.1.2.2 Return Value

Void

4.1.2.3 Faults

Fault Type	Description
<i>UnknownBusinessService</i>	This fault MUST be raised when the key information in the <i>serviceName</i> and <i>status</i> arguments does not match any BSIR contained in the SIL.

4.1.2.4 Description

The *updateBsrTtl* operation passes parameters to the SIL to identify a BSIR that is to be updated. The *ttl* parameter is used to provide a new value for the *ttl* field of the nominated BSIR. If a BSIR is correctly identified then its *ttl* field MUST be updated. Otherwise a fault MUST be raised.

4.1.3 Operation: *addBsrTechnicalRealisation*

4.1.3.1 Parameters

The operation SHALL accept the following parameters.

Parameter	Type	M O C	Description
<i>serviceOwnerId</i>	<i>OrgId</i>	M	Restricts the match of the following two parameters to BSIRs owned by this organisation.
<i>serviceName</i>	<i>string</i>	M	The first part of the key used to match a particular BSIR.
<i>status</i>	<i>Status</i>	M	The second part of the key used to match a particular BSIR.
<i>realisation</i>	<i>Technical Realisation</i>	M	The new technical realisation to be added to the identified BSIR.

4.1.3.2 Return Value

Void

4.1.3.3 Faults

Fault Type	Description
<i>UnknownBusinessService</i>	This fault MUST be raised when the key information in the <i>serviceName</i> and <i>status</i> arguments does not match any BSIR contained in the SIL.

4.1.3.4 Description

The *addBsrTechnicalRealisation* operation passes its first three arguments to the SIL to identify a BSIR that is to be updated. The *realisation* parameter is used to provide an additional value for the multi-valued *realisations* field of the nominated BSIR. If a BSIR is correctly identified then its *realisations* field MUST be updated. Otherwise a fault MUST be raised.

If an existing *TechnicalRealisation* is contained by the BSIR's *realisations* which has the same *patternName* as the one being passed as the *realisation* argument, then it must be deleted and replaced with the new *TechnicalRealisation*.

4.1.4 Operation: *deleteBsrTechnicalRealisation*

4.1.4.1 Parameters

The operation SHALL accept the following parameters.

Parameter	Type	M O C	Description
<i>serviceOwnerId</i>	<i>OrgId</i>	M	Restricts the match of the following two parameters to BSIRs owned by this organisation.
<i>serviceName</i>	<i>string</i>	M	The first part of the key used to match a particular BSIR.
<i>status</i>	<i>string</i>	M	The second part of the key used to match a particular BSIR.
<i>patternName</i>	<i>string</i>	M	The name of the pattern of the technical realisation to be deleted from the identified BSIR.
<i>version</i>	<i>string</i>	M	The version of the technical realisation to be deleted from the identified BSIR.

4.1.4.2 Return Value

Void

4.1.4.3 Faults

Fault Type	Description
<i>UnknownBusinessService</i>	This fault MUST be raised when the key information in the <i>serviceName</i> and <i>status</i> arguments does not match any BSIR contained in the SIL (unless the <i>status</i> is invalid, see below).
<i>UnknownPattern</i>	This fault will be raised if a BSIR is correctly identified by the initial arguments of the operation, but does not contain a TechnicalRealisation that has a <i>pattern</i> field that matches the <i>pattern</i> argument.
<i>CannotDeleteOnlyRealisation</i>	This fault MUST be raised if deleting the nominated Technical Realisation would leave the BSIR without any Realisations, as this violates constraints on the BSIR structure.

4.1.4.4 Description

The *deleteBsrTechnicalRealisation* operation passes parameters to the SIL to identify a BSIR that is to be updated. If they do not identify a BSIR, then the *UnknownBusinessService* fault MUST be raised.

The *patternName* parameter is used to identify the Technical Realisation of the nominated BSIR which must be deleted. If a the *patternName* argument does not identify a Technical Realisation then the *NoSuchPattern* fault MUST be raised. The *CannotDeleteOnlyRealisation* MUST be raised if the *patternName* argument nominates the only Technical Realisation of the BSIR. BSIRs MUST always have at least one Technical Realisation.

4.1.5 Operation: *updateBsirRole*

4.1.5.1 Parameters

The operation SHALL accept the following parameters.

Parameter	Type	M O C	Description
<i>serviceOwnerId</i>	<i>OrgId</i>	M	Restricts the match of the following two parameters to BSIRs owned by this organisation.
<i>serviceName</i>	<i>string</i>	M	The first part of the key used to match a particular BSIR.
<i>status</i>	<i>string</i>	M	The second part of the key used to match a particular BSIR.
<i>patternName</i>	<i>string</i>	M	The name of the pattern of the technical realisation in which the <i>TechnicalRole</i> will be updated.
<i>role</i>	<i>Technical Role</i>	M	The new technical role to update a role of a named technical realisation of the identified BSIR.

4.1.5.2 Return Value

Void

4.1.5.3 Faults

Fault Type	Description
<i>UnknownBusinessService</i>	This fault MUST be raised when the key information in the <i>serviceName</i> and <i>status</i> arguments does not match any BSIR contained in the SIL.
<i>NoSuchPattern</i>	This fault will be raised if a BSIR is correctly identified by the initial arguments of the operation, but does not contain a <i>TechnicalRealisation</i> that has a <i>patternName</i> field that matches the <i>patternName</i> argument.

4.1.5.4 Description

The *updateBsirTsrRef* operation passes parameters to the SIL to identify a BSIR that is to be updated. If a BSIR is not correctly identified then an *UnknownBusinessService* fault MUST be raised.

The *patternName* parameter is used to identify one of the *TechnicalRealisations* in the BSIR's multivalued *realisations* field. If no such *TechnicalRealisation* exists, the *NoSuchPattern* fault must be raised.

The *role* argument MUST be used to replace the *TechnicalRole* in the *roles* field of the identified *TechnicalRealisation* which has the same *roleName* field as the *TechnicalRole* already published. If no such *TechnicalRole* is currently published, then the *role* argument MUST be added to the identified *TechnicalRealisation's* *roles* field.

If the *TechnicalRole* has any TSIRs attached via its *serviceRecord* field, then these must be published into the SIL.

Any TSIRs that were originally published by attaching *TechnicalServiceEndpoints* to the *TechnicalServiceRef* in a published BSIR MUST be deleted from the SIL.

4.1.6 Operation: *deleteBsir*

4.1.6.1 Parameters

The operation SHALL accept the following parameters.

Parameter	Type	M O C	Description
<i>serviceOwnerId</i>	<i>OrgId</i>	M	Restricts the match of the following two parameters to BSIRs owned by this organisation.
<i>serviceName</i>	<i>string</i>	M	The first part of the key used to match a particular BSIR.
<i>status</i>	<i>string</i>	M	The second part of the key used to match a particular BSIR.

4.1.6.2 Return Value

Void

4.1.6.3 Faults

Fault Type	Description
<i>UnknownBusinessService</i>	This fault MUST be raised when the key information in the <i>serviceName</i> and <i>status</i> arguments does not match any BSIR contained in the SIL (unless the <i>status</i> is invalid, see below).

4.1.6.4 Description

The *deleteBsir* operation passes parameters to the SIL to identify a BSIR that is to be deleted. If the arguments do not identify a BSIR then a fault MUST be raised.

Any TSIRs that were originally published by attaching *TechnicalServiceEndpoints* to the *TechnicalServiceRefs* in the BSIR MUST be deleted from the SIL.

4.1.7 Operation: *publishTsir*

4.1.7.1 Parameters

The operation SHALL accept the following parameters.

Parameter	Type	M O C	Description
<i>tsir</i>	<i>TechnicalServiceInstanceRecord</i>	M	A new TSIR to be made available to clients of the lookup interfaces by the SIL.

4.1.7.2 Return Value

Void

4.1.7.3 Faults

None

4.1.7.4 Description

The *publishTsr* operation passes a new TSIR to the SIL for publication. The publisher (invoker of this operation) MUST be authorised to publish on behalf of the organisation given in the *serviceOwner* field of the *tsir* argument.

If the key fields of the TSIR (*serviceOwner*, *wsdlNamespace*, *status*) are the same as an existing TSIR in this SIL, then it MUST be replaced by the new TSIR. This means that updating the *ttl* of a TSIR is achieved by re-publishing the TSIR.

Note: Updating the *serviceHost* of the TSIR is inconsistent with the publication model of the SIL which has a principle that a TSIR resides in the SIL of the organisation which hosts the service. To change the *serviceHost* of a TSIR will require it to be deleted from this SIL, and published into the SIL of the service's new host.

4.1.8 Operation: *deleteTsr*

4.1.8.1 Parameters

The operation SHALL accept the following parameters.

Parameter	Type	M O C	Description
<i>serviceOwner Id</i>	<i>OrgId</i>	M	The owner of the Technical Service Instance Record to be deleted.
<i>wsdlType</i>	<i>Wsdlnamespace</i>	M	The WSDL type of the TSIR.
<i>status</i>	<i>Status</i>	M	The status of the TSIR.
<i>endpoint</i>	<i>anyURI</i>	M	The endpoint of the TSIR.

4.1.8.2 Return Value

Void.

4.1.8.3 Faults

Fault Type	Description
<i>UnknownTechnicalService</i>	This fault MUST be raised if the parameters together do not identify a currently advertised TSIR.

4.1.8.4 Description

The *deleteTsr* operation passes the identifying fields for a Technical Service Instance Record to the SIL to identify a TSIR that is to be deleted. The Directory Server MUST delete or archive the nominated TSIR and make it inaccessible to lookup clients.

5 Directory Server Lookup Interfaces

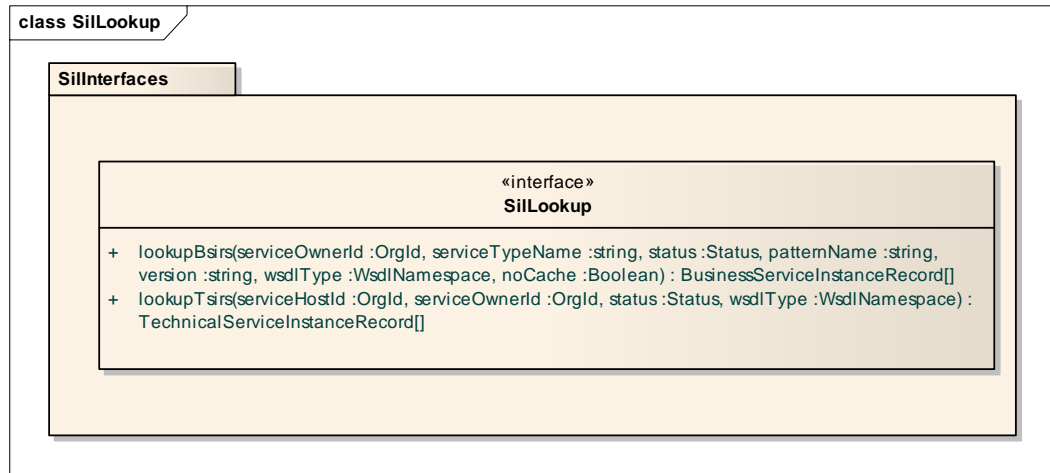


Figure 4: SIL Lookup Interfaces

5.1 Design Rationale

The model for a “logical SIL” is that it stores information about the business services that are *owned* by an organisation and the technical services that are *hosted* by that organisation. In other words the SIL logically represents the services supported by a single organisation. Organisations publish the details of business services that are owned by them, but these services may be hosted by any party. Conversely, they publish details about only the technical services that they host, even if the services are “owned” by another organisation. Business Service Instance Records are published by the organisation that owns the service into its SIL, and they have *references* to Technical Service Instance Records, at the SIL of the organisation that hosts the service. This is so that the endpoints of a service can be published by the organisation that is responsible for the service, i.e. organisations are responsible for their business services and referring to the technical services that implement them, as well as for the technical services that they host on their own behalf, or on behalf of another organisation to help them realise their business services.

The design of the SIL permits a single physical deployment of a SIL service to host the logical repository of records for multiple organisations. We do not require that a separate web services endpoint is made available for each logical SIL that the physical service hosts. Therefore, each operation in the SillLookup interface requires an appropriate organisational identifier as its first parameter, to identify the “logical SIL” that is being queried. In the case of BSIRs, this first parameter is always the service owner’s Id, and in the case of TSIRs it is the service host’s Id. TSIRs may be hosted on behalf of another organisation, and so the organisation Id of the service owner is always the second parameter of lookup operations for TSIRs.

5.2 Interface: *SILLookup*

5.2.1 Operation: *lookupBsirs*

5.2.1.1 Parameters

The operation SHALL accept the following parameters.

Parameter	Type	M O C	Description
<i>serviceOwnerId</i>	<i>OrgId</i>	M	The owner of the BSIRs to be looked up.
<i>serviceName</i>	<i>string</i>	O	The name of the business service offered by the owner, as specified in the NEHTA (or other) specifications.
<i>status</i>	<i>Status</i>	M	The status of the business service advertised in the SIL.
<i>patternName</i>	<i>string</i>	O	The interaction pattern of the Technical Realisation for the Business Service that is required.
<i>version</i>	<i>string</i>	O	The version of the Technical Realisation for the Business Service that is required.
<i>wsdlType</i>	<i>wsdlName space</i>	O	The WSDL Type of a <i>TechnicalServiceRef</i> owned by the BSIR.
<i>noCache</i>	<i>boolean</i>	M	If TRUE, the SIL MUST NOT use cached TSIR Values, and MUST lookup each remote TSIR using the <i>TechnicalServiceRef</i> information contained in the BSIR identified by the other parameters. If FALSE, all valid cached TSIRs MUST be returned without conducting additional lookups.

5.2.1.2 Return Value

- The type of the return value is defined as *Set of BusinessServiceInstanceRecord* (i.e. there is no prescribed order in which BSIRs are returned when multiple of them are matched).

5.2.1.3 Faults

Fault Type	Description
<i>UnknownBusinessService</i>	MUST be raised when the <i>status</i> and <i>serviceName</i> are given, but they do not identify a BSIR in this SIL.

5.2.1.4 Description

The *lookupBsirs* operation will be used by Service Clients to discover a possible business service for a service owner which matches any of their specified criteria, including business service type name, status, technical realisation pattern and WSDL type. A BSIR owned by the *serviceOwner* will be selected if all of the optional arguments *serviceName*, *status*,

patternName, *version* and *wsdlNamespace*, when provided, are exact string matches for the equivalent fields in the BSIR.

If one or more matching BSIR is selected, the SIL MUST return the BSIRs with all endpoint information from the actual TSIRs referred to by any contained *TechnicalServiceRefs*. This information MUST be returned as *TechnicalServiceEndpoints* in the *tses* field of the *TechnicalServiceRefs* either:

- by local lookup, if the TSIR(s) are in the same SIL, or
- by looking up the SIL's cache for previously retrieved TSIRs, if they are not yet expired. The *tSirSource* of the *TechnicalServiceRef* must then be set to "CACHED", or
- by making a *lookupTsirs* invocation on the remote SIL referenced by the *TechnicalServiceRef's* *sidEndpoint* attribute, using the *status* of the containing BSIR as the *status* argument to this call, and the values stored in the *TechnicalServiceRef* for the other arguments. The *tSirSource* of the *TechnicalServiceRef* must then be set to "RETRIEVED", or if this call fails, then
- by returning an empty *tses* value, and setting the *tSirSource* of the endpoint to "SIL_UNAVAILABLE" if the SIL referred to by the *TechnicalServiceRef* was unavailable, or "NO_MATCHES" if the SIL returned no matching TSIRs.

5.2.2 Operation: *lookupTsirs*

5.2.2.1 Parameters

The operation SHALL accept the following parameters.

Parameter	Type	M O C	Description
<i>serviceHostId</i>	<i>OrgId</i>	M	The identifier of the host of the TSIRs to be looked up.
<i>serviceOwnerId</i>	<i>OrgId</i>	M	The identifier of the organisation on behalf of whom the service is operated.
<i>wsdlType</i>	<i>wsdlNamespace</i>	O	The WSDL type which the returned endpoints implement.
<i>status</i>	<i>Status</i>	M	The status which matched TSIRs will have.

5.2.2.2 Return Value

The type of the return value is defined as *a Set of TechnicalServiceInstanceRecord*. That is, there may be multiple returned TSIRs, and the order of their appearance is arbitrary and insignificant.

5.2.2.3 Faults

There are no service-specific faults.

5.2.2.4 Description

The *lookupTsirsByOwnerAndType* operation will be used by Service Clients to discover all the technical services which match their specified service host and service owner. If the arguments *wsdlNamespace* or *status* are given TSIRs will be matched and returned if there is an exact string match for the equivalent field in a TSIR with the correct owner and host.

Appendix A: References

- [NIF2006] NEHTA, Interoperability Framework, 1.0, 2006-04-01.
- [NWSP2008] NEHTA, Web Services Profile, 3.0, 2008-09-01.
- [CPIS2008] NEHTA, Concepts and Patterns for Implementing Services, v2.0, 2008.
- [SILBIA2008] NEHTA, Service Instance Locator Business and Information Architecture, v1.0.

Appendix B: XSD

B.1 SilDataTypes.xsd

```

<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema
  elementFormDefault="qualified"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  targetNamespace="urn:xml-gov-au:nehta:types:SilDataTypes:1.0"
  xmlns:smi="urn:xml-gov-au:nehta:types:SecureMessaging:1.0"
  xmlns:nash="urn:xml-gov-au:nehta:types:NashCertRef:1.0"
  xmlns:tns="urn:xml-gov-au:nehta:types:SilDataTypes:1.0">
  <xsd:import
    namespace="urn:xml-gov-au:nehta:types:SecureMessaging:1.0"
    schemaLocation="../../smi/xsd/SecureMessaging.xsd"/>
  <xsd:import namespace="urn:xml-gov-au:nehta:types:NashCertRef:1.0"
    schemaLocation="../../nash/xsd/NashCertRef.xsd"/>
  <xsd:simpleType name="Status">
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="PRODUCTION"/>
      <xsd:enumeration value="TESTING"/>
      <xsd:enumeration value="TRAINING"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name="TsirSource">
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="CACHED"/>
      <xsd:enumeration value="RETRIEVED"/>
      <xsd:enumeration value="SIL_UNAVAILABLE"/>
      <xsd:enumeration value="NO_MATCHES"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name="TsirLocation">
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="NOTIFICATION"/>
      <xsd:enumeration value="LOCAL"/>
      <xsd:enumeration value="REMOTE"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:complexType name="TechnicalServiceRef">
    <xsd:sequence>
      <!-- The following elements should be a group called
           TechnicalServiceCommon -->
      <xsd:element name="wsdlType" type="smi:WsdlNamespace"/>
      <xsd:element name="status" type="tns:Status"/>
      <xsd:element name="serviceHostId" type="smi:OrgId"/>
      <!-- end group -->
      <xsd:element name="silHostId" type="smi:OrgId"/>
      <xsd:element name="silCertRef" type="nash:CertificateRef"/>
      <xsd:element name="silEndpoint" type="xsd:anyURI"/>
      <xsd:element name="tsirSource" type="tns:TsirSource"
minOccurs="0"/>
      <xsd:element name="endpoint"
type="tns:TechnicalServiceEndpoint"
minOccurs="0"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="TechnicalServiceEndpoint">
    <xsd:sequence>
      <xsd:element name="endpoint" type="xsd:anyURI"/>
      <xsd:element name="ttl" type="xsd:duration"/>
      <xsd:element name="serviceCertRef"
type="nash:CertificateRef"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="TechnicalRealisation">
    <xsd:sequence>
      <xsd:element name="patternName" type="xsd:string"/>
      <xsd:element name="version" type="xsd:string"/>
      <xsd:element name="roles" type="tns:TechnicalRole"
minOccurs="1" maxOccurs="unbounded"/>

```

```

        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="TechnicalRole">
        <xsd:sequence>
            <xsd:element name="roleName" type="xsd:string"/>
            <xsd:element name="payloadEncCertRef"
type="nash:CertificateRef"
                minOccurs="0"/>
            <xsd:element name="tsirLocation" type="tns:TsirLocation"/>
            <xsd:element name="serviceRef" type="tns:TechnicalServiceRef"
                minOccurs="0"/>
            <xsd:element name="serviceRecord"
type="tns:TechnicalServiceInstanceRecord"
                minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="TechnicalServiceInstanceRecord">
        <xsd:complexContent>
            <xsd:extension base="tns:TechnicalServiceEndpoint">
                <xsd:sequence>
                    <!-- The following elements should be a group called
TechnicalServiceCommon -->
                    <xsd:element name="wsdlType"
type="smi:Wsdlnamespace"/>
                    <xsd:element name="status" type="tns:Status"/>
                    <xsd:element name="serviceHostId" type="smi:OrgId"/>
                    <!-- end group -->
                    <xsd:element name="serviceOwnerId" type="smi:OrgId"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="BusinessServiceInstanceRecord">
        <xsd:sequence>
            <xsd:element name="serviceTypeName" type="xsd:string"/>
            <xsd:element name="ttl" type="xsd:duration"/>
            <xsd:element name="status" type="tns:Status"/>
            <xsd:element name="serviceOwnerId" type="smi:OrgId"/>
            <xsd:element name="realisations"
type="tns:TechnicalRealisation"
                minOccurs="1" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:schema>

```

B.2 SilFaults.xsd

```

<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema
    elementFormDefault="qualified"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="urn:xml-gov-au:nehta:types:SilFaults:1.0"
    xmlns:tns="urn:xml-gov-au:nehta:types:SilFaults:1.0"
    xmlns:sdt="urn:xml-gov-au:nehta:types:SilDataTypes:1.0"
    xmlns:smi="urn:xml-gov-au:nehta:types:SecureMessaging:1.0">

    <xsd:import namespace="urn:xml-gov-au:nehta:types:SilDataTypes:1.0"
        schemaLocation="SilDataTypes.xsd"/>

    <xsd:import namespace="urn:xml-gov-
au:nehta:types:SecureMessaging:1.0"
        schemaLocation="../../smi/xsd/SecureMessaging.xsd"/>

    <xsd:complexType name="UnknownBusinessServiceError">
        <xsd:complexContent>
            <xsd:extension base="smi:ErrorBase">
                <xsd:sequence>
                    <xsd:element name="serviceOwnerId" type="smi:OrgId"/>
                    <xsd:element name="serviceTypeName" type="xsd:string"/>
                    <xsd:element name="status" type="sdt:Status"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

```

```

</xsd:complexType>

<xsd:complexType name="DuplicateBusinessServiceError">
  <xsd:complexContent>
    <xsd:extension base="smi:ErrorBase">
      <xsd:sequence>
        <xsd:element name="serviceOwnerId" type="smi:OrgId"/>
        <xsd:element name="serviceTypeName" type="xsd:string"/>
        <xsd:element name="status" type="sdt:Status"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="DuplicateTechnicalServiceError">
  <xsd:complexContent>
    <xsd:extension base="smi:ErrorBase">
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="UnknownVersionError">
  <xsd:complexContent>
    <xsd:extension base="smi:ErrorBase">
      <xsd:sequence>
        <xsd:element name="version" type="xsd:string"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="UnknownPatternError">
  <xsd:complexContent>
    <xsd:extension base="smi:ErrorBase">
      <xsd:sequence>
        <xsd:element name="patternName" type="xsd:string"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="CannotDeleteOnlyRealisationError">
  <xsd:complexContent>
    <xsd:extension base="smi:ErrorBase"/>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="UnknownTechnicalServiceError">
  <xsd:complexContent>
    <xsd:extension base="smi:ErrorBase"/>
  </xsd:complexContent>
</xsd:complexType>

  <xsd:element name="UnknownBusinessServiceFault"
type="tns:UnknownBusinessServiceError"/>
  <xsd:element name="DuplicateBusinessServiceFault"
type="tns:DuplicateBusinessServiceError"/>
  <xsd:element name="DuplicateTechnicalServiceFault"
type="tns:DuplicateTechnicalServiceError"/>
  <xsd:element name="UnknownPatternFault"
type="tns:UnknownPatternError"/>
  <xsd:element name="UnknownVersionFault"
type="tns:UnknownVersionError"/>
  <xsd:element name="CannotDeleteOnlyRealisationFault"
type="tns:CannotDeleteOnlyRealisationError"/>
  <xsd:element name="UnknownTechnicalServiceFault"
type="tns:UnknownTechnicalServiceError"/>

</xsd:schema>

```

Appendix C: WSDL

C.1 SilPublishInterface.wsdl

```

<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions
xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:tns="urn:xml-gov-
au:nehta:service:ServiceInstanceLocator:SilPublish:1.0"
  xmlns:sdt="urn:xml-gov-au:nehta:types:SilDataTypes:1.0"
  xmlns:faults="urn:xml-gov-au:nehta:types:SilFaults:1.0"
  xmlns:smi="urn:xml-gov-
au:nehta:types:SecureMessaging:1.0"
  xmlns:nash="urn:xml-gov-au:nehta:types:NashCertRef:1.0"
  name="SilPublish"
  targetNamespace="urn:xml-gov-
au:nehta:service:ServiceInstanceLocator:SilPublish:1.0">
  <wsdl:types>
    <xsd:schema targetNamespace="urn:xml-gov-
au:nehta:service:ServiceInstanceLocator:SilPublish:1.0"
      elementFormDefault="qualified">
      <xsd:import namespace="urn:xml-gov-
au:nehta:types:SilDataTypes:1.0"
        schemaLocation="../xsd/SilDataTypes.xsd"/>
      <xsd:import namespace="urn:xml-gov-au:nehta:types:SilFaults:1.0"
        schemaLocation="../xsd/SilFaults.xsd"/>
      <xsd:import namespace="urn:xml-gov-
au:nehta:types:SecureMessaging:1.0"
        schemaLocation="../../smi/xsd/SecureMessaging.xsd"/>
      <xsd:import namespace="urn:xml-gov-
au:nehta:types:NashCertRef:1.0"
        schemaLocation="../../nash/xsd/NashCertRef.xsd"/>
      <xsd:element name="publishBsir">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="bsir"
type="sdt:BusinessServiceInstanceRecord"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="publishBsirResponse">
        <xsd:complexType/>
      </xsd:element>
      <xsd:element name="updateBsirTtl">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="serviceOwnerId" type="smi:OrgId"/>
            <xsd:element name="serviceName" type="xsd:string"/>
            <xsd:element name="status" type="sdt:Status"/>
            <xsd:element name="ttl" type="xsd:duration"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="updateBsirTtlResponse">
        <xsd:complexType/>
      </xsd:element>
      <xsd:element name="addBsirTechnicalRealisation">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="serviceOwnerId" type="smi:OrgId"/>
            <xsd:element name="serviceName" type="xsd:string"/>
            <xsd:element name="status" type="sdt:Status"/>
            <xsd:element name="tr"
type="sdt:TechnicalRealisation"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="addBsirTechnicalRealisationResponse">
        <xsd:complexType/>

```

```

</xsd:element>
<xsd:element name="deleteBsirTechnicalRealisation">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="serviceOwnerId" type="smi:OrgId"/>
      <xsd:element name="serviceName" type="xsd:string"/>
      <xsd:element name="status" type="sdt:Status"/>
      <xsd:element name="patternName" type="xsd:string"/>
      <xsd:element name="version" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="deleteBsirTechnicalRealisationResponse">
  <xsd:complexType/>
</xsd:element>
<xsd:element name="updateTechnicalRole">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="serviceOwnerId" type="smi:OrgId"/>
      <xsd:element name="serviceName" type="xsd:string"/>
      <xsd:element name="status" type="sdt:Status"/>
      <xsd:element name="patternName" type="xsd:string"/>
      <xsd:element name="version" type="xsd:string"/>
      <xsd:element name="technicalRole"
type="sdt:TechnicalRole"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="updateTechnicalRoleResponse">
  <xsd:complexType/>
</xsd:element>
<xsd:element name="deleteBsir">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="serviceOwnerId" type="smi:OrgId"/>
      <xsd:element name="serviceName" type="xsd:string"/>
      <xsd:element name="status" type="sdt:Status"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="deleteBsirResponse">
  <xsd:complexType/>
</xsd:element>
<xsd:element name="publishTsir">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="tsir"
type="sdt:TechnicalServiceInstanceRecord"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="publishTsirResponse">
  <xsd:complexType/>
</xsd:element>
<xsd:element name="deleteTsir">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="serviceOwnerId" type="smi:OrgId"/>
      <xsd:element name="wsdlType" type="smi:Wsdlnamespace"/>
      <xsd:element name="status" type="sdt:Status"/>
      <xsd:element name="endpoint" type="xsd:anyURI"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="deleteTsirResponse">
  <xsd:complexType/>
</xsd:element>
</xsd:schema>
</wsdl:types>
<wsdl:message name="publishBsirInMsg">
  <wsdl:part name="parameters" element="tns:publishBsir"/>
</wsdl:message>
<wsdl:message name="publishBsirOutMsg">
  <wsdl:part name="parameters" element="tns:publishBsirResponse"/>

```

```

</wsdl:message>
<wsdl:message name="updateBsirTtlInMsg">
  <wsdl:part name="parameters" element="tns:updateBsirTtl"/>
</wsdl:message>
<wsdl:message name="updateBsirTtlOutMsg">
  <wsdl:part name="parameters" element="tns:updateBsirTtlResponse"/>
</wsdl:message>
<wsdl:message name="addBsirTechnicalRealisationInMsg">
  <wsdl:part name="parameters"
element="tns:addBsirTechnicalRealisation"/>
</wsdl:message>
<wsdl:message name="addBsirTechnicalRealisationOutMsg">
  <wsdl:part name="parameters"
element="tns:addBsirTechnicalRealisationResponse"/>
</wsdl:message>
<wsdl:message name="deleteBsirTechnicalRealisationInMsg">
  <wsdl:part name="parameters"
element="tns:deleteBsirTechnicalRealisation"/>
</wsdl:message>
<wsdl:message name="deleteBsirTechnicalRealisationOutMsg">
  <wsdl:part name="parameters"
element="tns:deleteBsirTechnicalRealisationResponse"/>
</wsdl:message>
<wsdl:message name="updateTechnicalRoleInMsg">
  <wsdl:part name="parameters" element="tns:updateTechnicalRole"/>
</wsdl:message>
<wsdl:message name="updateTechnicalRoleOutMsg">
  <wsdl:part name="parameters"
element="tns:updateTechnicalRoleResponse"/>
</wsdl:message>
<wsdl:message name="deleteBsirInMsg">
  <wsdl:part name="parameters" element="tns:deleteBsir"/>
</wsdl:message>
<wsdl:message name="deleteBsirOutMsg">
  <wsdl:part name="parameters" element="tns:deleteBsirResponse"/>
</wsdl:message>
<wsdl:message name="publishTsirInMsg">
  <wsdl:part name="parameters" element="tns:publishTsir"/>
</wsdl:message>
<wsdl:message name="publishTsirOutMsg">
  <wsdl:part name="parameters" element="tns:publishTsirResponse"/>
</wsdl:message>
<wsdl:message name="deleteTsirInMsg">
  <wsdl:part name="parameters" element="tns:deleteTsir"/>
</wsdl:message>
<wsdl:message name="deleteTsirOutMsg">
  <wsdl:part name="parameters" element="tns:deleteTsirResponse"/>
</wsdl:message>
<wsdl:message name="StandardFaultMsg">
  <wsdl:part name="parameters" element="smi:StandardFault"/>
</wsdl:message>
<wsdl:message name="UnknownBusinessServiceFaultMsg">
  <wsdl:part name="parameters"
element="faults:UnknownBusinessServiceFault"/>
</wsdl:message>
<wsdl:message name="DuplicateBusinessServiceFaultMsg">
  <wsdl:part name="parameters"
element="faults:DuplicateBusinessServiceFault"/>
</wsdl:message>
<wsdl:message name="DuplicateTechnicalServiceFaultMsg">
  <wsdl:part name="parameters"
element="faults:DuplicateTechnicalServiceFault"/>
</wsdl:message>
<wsdl:message name="UnknownPatternFaultMsg">
  <wsdl:part name="parameters" element="faults:UnknownPatternFault"/>
</wsdl:message>
<wsdl:message name="UnknownVersionFaultMsg">
  <wsdl:part name="parameters" element="faults:UnknownVersionFault"/>
</wsdl:message>
<wsdl:message name="CannotDeleteOnlyRealisationFaultMsg">
  <wsdl:part name="parameters"
element="faults:CannotDeleteOnlyRealisationFault"/>
</wsdl:message>
<wsdl:message name="UnknownTechnicalServiceFaultMsg">

```

```

        <wsdl:part name="parameters"
element=" faults:UnknownTechnicalServiceFault" />
    </wsdl:message>
    <wsdl:portType name="SilPublish">
        <wsdl:operation name="publishBsir">
            <wsdl:input message="tns:publishBsirInMsg"
                wsam:Action="urn:xml-gov-
au:nehta:service:ServiceInstanceLocator:SilPublish:1.0:SilPublish:publish
BsirRequest" />
            <wsdl:output message="tns:publishBsirOutMsg"
                wsam:Action="urn:xml-gov-
au:nehta:service:ServiceInstanceLocator:SilPublish:1.0:SilPublish:publish
BsirResponse" />
            <wsdl:fault name="StandardFault" message="tns:StandardFaultMsg"
                wsam:Action="urn:xml-gov-
au:nehta:service:ServiceInstanceLocator:SilPublish:1.0:SilPublish:publish
Bsir:Fault:StandardFault" />
            <wsdl:fault name="DuplicateBusinessServiceFault"
                message="tns:DuplicateBusinessServiceFaultMsg"
                wsam:Action="urn:xml-gov-
au:nehta:service:ServiceInstanceLocator:SilPublish:1.0:SilPublish:publish
Bsir:Fault:DuplicateBusinessServiceFault" />
        </wsdl:operation>
        <wsdl:operation name="updateBsirTtl">
            <wsdl:input message="tns:updateBsirTtlInMsg"
                wsam:Action="urn:xml-gov-
au:nehta:service:ServiceInstanceLocator:SilPublish:1.0:SilPublish:updateB
sirTtlRequest" />
            <wsdl:output message="tns:updateBsirTtlOutMsg"
                wsam:Action="urn:xml-gov-
au:nehta:service:ServiceInstanceLocator:SilPublish:1.0:SilPublish:updateB
sirTtlResponse" />
            <wsdl:fault name="StandardFault" message="tns:StandardFaultMsg"
                wsam:Action="urn:xml-gov-
au:nehta:service:ServiceInstanceLocator:SilPublish:1.0:SilPublish:updateB
sirTtl:Fault:StandardFault" />
            <wsdl:fault name="UnknownBusinessServiceFault"
                message="tns:UnknownBusinessServiceFaultMsg"
                wsam:Action="urn:xml-gov-
au:nehta:service:ServiceInstanceLocator:SilPublish:1.0:SilPublish:updateB
sirTtl:Fault:UnknownBusinessServiceFault" />
        </wsdl:operation>
        <wsdl:operation name="addBsirTechnicalRealisation">
            <wsdl:input message="tns:addBsirTechnicalRealisationInMsg"
                wsam:Action="urn:xml-gov-
au:nehta:service:ServiceInstanceLocator:SilPublish:1.0:SilPublish:addBsir
TechnicalRealisationRequest" />
            <wsdl:output message="tns:addBsirTechnicalRealisationOutMsg"
                wsam:Action="urn:xml-gov-
au:nehta:service:ServiceInstanceLocator:SilPublish:1.0:SilPublish:addBsir
TechnicalRealisationResponse" />
            <wsdl:fault name="StandardFault" message="tns:StandardFaultMsg"
                wsam:Action="urn:xml-gov-
au:nehta:service:ServiceInstanceLocator:SilPublish:1.0:SilPublish:addBsir
TechnicalRealisation:Fault:StandardFault" />
            <wsdl:fault name="UnknownBusinessServiceFault"
                message="tns:UnknownBusinessServiceFaultMsg"
                wsam:Action="urn:xml-gov-
au:nehta:service:ServiceInstanceLocator:SilPublish:1.0:SilPublish:addBsir
TechnicalRealisation:Fault:UnknownBusinessServiceFault" />
        </wsdl:operation>
        <wsdl:operation name="deleteBsirTechnicalRealisation">
            <wsdl:input message="tns:deleteBsirTechnicalRealisationInMsg"
                wsam:Action="urn:xml-gov-
au:nehta:service:ServiceInstanceLocator:SilPublish:1.0:SilPublish:deleteB
sirTechnicalRealisationRequest" />
            <wsdl:output message="tns:deleteBsirTechnicalRealisationOutMsg"
                wsam:Action="urn:xml-gov-
au:nehta:service:ServiceInstanceLocator:SilPublish:1.0:SilPublish:deleteB
sirTechnicalRealisationResponse" />
            <wsdl:fault name="StandardFault" message="tns:StandardFaultMsg"
                wsam:Action="urn:xml-gov-
au:nehta:service:ServiceInstanceLocator:SilPublish:1.0:SilPublish:deleteB
sirTechnicalRealisation:Fault:StandardFault" />

```

```

        <wsdl:fault name="UnknownBusinessServiceFault"
message="tns:UnknownBusinessServiceFaultMsg"
        wsam:Action="urn:xml-gov-
au:nehta:service:ServiceInstanceLocator:SilPublish:1.0:SilPublish:deleteB
sirTechnicalRealisation:Fault:UnknownBusinessServiceFault"/>
        <wsdl:fault name="UnknownPatternFault"
message="tns:UnknownPatternFaultMsg"
        wsam:Action="urn:xml-gov-
au:nehta:service:ServiceInstanceLocator:SilPublish:1.0:SilPublish:deleteB
sirTechnicalRealisation:Fault:UnknownPatternFault"/>
        <wsdl:fault name="UnknownVersionFault"
message="tns:UnknownVersionFaultMsg"
        wsam:Action="urn:xml-gov-
au:nehta:service:ServiceInstanceLocator:SilPublish:1.0:SilPublish:deleteB
sirTechnicalRealisation:Fault:UnknownVersionFault"/>
        <wsdl:fault name="CannotDeleteOnlyRealisationFault"
        message="tns:CannotDeleteOnlyRealisationFaultMsg"
        wsam:Action="urn:xml-gov-
au:nehta:service:ServiceInstanceLocator:SilPublish:1.0:SilPublish:deleteB
sirTechnicalRealisation:Fault:CannotDeleteOnlyRealisationFault"/>
    </wsdl:operation>
    <wsdl:operation name="updateTechnicalRole">
        <wsdl:input message="tns:updateTechnicalRoleInMsg"
        wsam:Action="urn:xml-gov-
au:nehta:service:ServiceInstanceLocator:SilPublish:1.0:SilPublish:updateT
echnicalRoleRequest"/>
        <wsdl:output message="tns:updateTechnicalRoleOutMsg"
        wsam:Action="urn:xml-gov-
au:nehta:service:ServiceInstanceLocator:SilPublish:1.0:SilPublish:updateT
echnicalRoleResponse"/>
        <wsdl:fault name="StandardFault" message="tns:StandardFaultMsg"
        wsam:Action="urn:xml-gov-
au:nehta:service:ServiceInstanceLocator:SilPublish:1.0:SilPublish:updateT
echnicalRole:Fault:StandardFault"/>
        <wsdl:fault name="UnknownBusinessServiceFault"
message="tns:UnknownBusinessServiceFaultMsg"
        wsam:Action="urn:xml-gov-
au:nehta:service:ServiceInstanceLocator:SilPublish:1.0:SilPublish:updateT
echnicalRole:Fault:UnknownBusinessServiceFault"/>
        <wsdl:fault name="UnknownPatternFault"
message="tns:UnknownPatternFaultMsg"
        wsam:Action="urn:xml-gov-
au:nehta:service:ServiceInstanceLocator:SilPublish:1.0:SilPublish:updateT
echnicalRole:Fault:UnknownPatternFault"/>
        <wsdl:fault name="UnknownVersionFault"
message="tns:UnknownVersionFaultMsg"
        wsam:Action="urn:xml-gov-
au:nehta:service:ServiceInstanceLocator:SilPublish:1.0:SilPublish:updateT
echnicalRole:Fault:UnknownVersionFault"/>
    </wsdl:operation>
    <wsdl:operation name="deleteBsir">
        <wsdl:input message="tns:deleteBsirInMsg"
        wsam:Action="urn:xml-gov-
au:nehta:service:ServiceInstanceLocator:SilPublish:1.0:SilPublish:deleteB
sirRequest"/>
        <wsdl:output message="tns:deleteBsirOutMsg"
        wsam:Action="urn:xml-gov-
au:nehta:service:ServiceInstanceLocator:SilPublish:1.0:SilPublish:deleteB
sirResponse"/>
        <wsdl:fault name="StandardFault" message="tns:StandardFaultMsg"
        wsam:Action="urn:xml-gov-
au:nehta:service:ServiceInstanceLocator:SilPublish:1.0:SilPublish:deleteB
sir:Fault:StandardFault"/>
        <wsdl:fault name="UnknownBusinessServiceFault"
message="tns:UnknownBusinessServiceFaultMsg"
        wsam:Action="urn:xml-gov-
au:nehta:service:ServiceInstanceLocator:SilPublish:1.0:SilPublish:deleteB
sir:Fault:UnknownBusinessServiceFault"/>
    </wsdl:operation>
    <wsdl:operation name="publishTsir">
        <wsdl:input message="tns:publishTsirInMsg"
        wsam:Action="urn:xml-gov-
au:nehta:service:ServiceInstanceLocator:SilPublish:1.0:SilPublish:publish
TsirRequest"/>

```

```

        <wsdl:output message="tns:publishTsirOutMsg"
            wsam:Action="urn:xml-gov-
au:nehta:service:ServiceInstanceLocator:SilPublish:1.0:SilPublish:publish
TsirResponse"/>
        <wsdl:fault name="StandardFault" message="tns:StandardFaultMsg"
            wsam:Action="urn:xml-gov-
au:nehta:service:ServiceInstanceLocator:SilPublish:1.0:SilPublish:publish
Tsir:Fault:StandardFault"/>
        <wsdl:fault name="DuplicateTechnicalServiceFault"
            message="tns:DuplicateTechnicalServiceFaultMsg"
            wsam:Action="urn:xml-gov-
au:nehta:service:ServiceInstanceLocator:SilPublish:1.0:SilPublish:publish
Tsir:Fault:DuplicateTechnicalServiceFault"/>
    </wsdl:operation>
    <wsdl:operation name="deleteTsir">
        <wsdl:input message="tns:deleteTsirInMsg"
            wsam:Action="urn:xml-gov-
au:nehta:service:ServiceInstanceLocator:SilPublish:1.0:SilPublish:deleteT
sirRequest"/>
        <wsdl:output message="tns:deleteTsirOutMsg"
            wsam:Action="urn:xml-gov-
au:nehta:service:ServiceInstanceLocator:SilPublish:1.0:SilPublish:deleteT
sirResponse"/>
        <wsdl:fault name="StandardFault" message="tns:StandardFaultMsg"
            wsam:Action="urn:xml-gov-
au:nehta:service:ServiceInstanceLocator:SilPublish:1.0:SilPublish:deleteT
sir:Fault:StandardFault"/>
        <wsdl:fault name="UnknownTechnicalServiceFault"
            message="tns:UnknownTechnicalServiceFaultMsg"
            wsam:Action="urn:xml-gov-
au:nehta:service:ServiceInstanceLocator:SilPublish:1.0:SilPublish:deleteT
sir:Fault:UnknownTechnicalServiceFault"/>
    </wsdl:operation>
</wsdl:portType>
</wsdl:definitions>

```

C.2 SilPublish.wsdl

```

<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap12/"
    xmlns:tns="urn:xml-gov-
au:nehta:service:ServiceInstanceLocator:SilPublish:1.0"
    xmlns:wsp="http://www.w3.org/ns/ws-policy"
    xmlns:wsu="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"

    xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata"
    xmlns:sp="http://docs.oasis-open.org/ws-sx/ws-
securitypolicy/200702"
        name="SilPublish"
        targetNamespace="urn:xml-gov-
au:nehta:service:ServiceInstanceLocator:SilPublish:1.0">
    <wsdl:import namespace="urn:xml-gov-
au:nehta:service:ServiceInstanceLocator:SilPublish:1.0"
        location="SilPublishInterface.wsdl"/>
    <wsdl:binding name="SilPublishBinding" type="tns:SilPublish">
        <wsp:PolicyReference URI="#AddressingPolicy"/>
        <wsp:PolicyReference URI="#SecurityPolicy"/>
        <soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/>
        <wsdl:operation name="publishBsir">
            <soap:operation style="document" soapActionRequired="false"/>
            <wsdl:input>
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal"/>
            </wsdl:output>
            <wsdl:fault name="StandardFault">
                <soap:fault name="StandardFault" use="literal"/>
            </wsdl:fault>

```

```

        <wsdl:fault name="DuplicateBusinessServiceFault">
          <soap:fault name="DuplicateBusinessServiceFault"
use="literal"/>
        </wsdl:fault>
      </wsdl:operation>
      <wsdl:operation name="updateBsirTtl">
        <soap:operation style="document" soapActionRequired="false"/>
        <wsdl:input>
          <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
          <soap:body use="literal"/>
        </wsdl:output>
        <wsdl:fault name="StandardFault">
          <soap:fault name="StandardFault" use="literal"/>
        </wsdl:fault>
        <wsdl:fault name="UnknownBusinessServiceFault">
          <soap:fault name="UnknownBusinessServiceFault"
use="literal"/>
        </wsdl:fault>
      </wsdl:operation>
      <wsdl:operation name="addBsirTechnicalRealisation">
        <soap:operation style="document" soapActionRequired="false"/>
        <wsdl:input>
          <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
          <soap:body use="literal"/>
        </wsdl:output>
        <wsdl:fault name="StandardFault">
          <soap:fault name="StandardFault" use="literal"/>
        </wsdl:fault>
        <wsdl:fault name="UnknownBusinessServiceFault">
          <soap:fault name="UnknownBusinessServiceFault"
use="literal"/>
        </wsdl:fault>
      </wsdl:operation>
      <wsdl:operation name="deleteBsirTechnicalRealisation">
        <soap:operation style="document" soapActionRequired="false"/>
        <wsdl:input>
          <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
          <soap:body use="literal"/>
        </wsdl:output>
        <wsdl:fault name="StandardFault">
          <soap:fault name="StandardFault" use="literal"/>
        </wsdl:fault>
        <wsdl:fault name="UnknownBusinessServiceFault">
          <soap:fault name="UnknownBusinessServiceFault"
use="literal"/>
        </wsdl:fault>
      </wsdl:operation>
      <wsdl:fault name="UnknownPatternFault">
        <soap:fault name="UnknownPatternFault" use="literal"/>
      </wsdl:fault>
      <wsdl:fault name="UnknownVersionFault">
        <soap:fault name="UnknownVersionFault" use="literal"/>
      </wsdl:fault>
      <wsdl:fault name="CannotDeleteOnlyRealisationFault">
        <soap:fault name="CannotDeleteOnlyRealisationFault"
use="literal"/>
      </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="updateTechnicalRole">
      <soap:operation style="document" soapActionRequired="false"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
      <wsdl:fault name="StandardFault">
        <soap:fault name="StandardFault" use="literal"/>
      </wsdl:fault>
    </wsdl:operation>

```

```

        <wsdl:fault name="UnknownBusinessServiceFault">
            <soap:fault name="UnknownBusinessServiceFault"
use="literal"/>
        </wsdl:fault>
        <wsdl:fault name="UnknownPatternFault">
            <soap:fault name="UnknownPatternFault" use="literal"/>
        </wsdl:fault>
        <wsdl:fault name="UnknownVersionFault">
            <soap:fault name="UnknownVersionFault" use="literal"/>
        </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="deleteBsir">
        <soap:operation style="document" soapActionRequired="false"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
        <wsdl:fault name="StandardFault">
            <soap:fault name="StandardFault" use="literal"/>
        </wsdl:fault>
        <wsdl:fault name="UnknownBusinessServiceFault">
            <soap:fault name="UnknownBusinessServiceFault"
use="literal"/>
        </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="publishTsir">
        <soap:operation style="document" soapActionRequired="false"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
        <wsdl:fault name="StandardFault">
            <soap:fault name="StandardFault" use="literal"/>
        </wsdl:fault>
        <wsdl:fault name="DuplicateTechnicalServiceFault">
            <soap:fault name="DuplicateTechnicalServiceFault"
use="literal"/>
        </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="deleteTsir">
        <soap:operation style="document" soapActionRequired="false"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
        <wsdl:fault name="StandardFault">
            <soap:fault name="StandardFault" use="literal"/>
        </wsdl:fault>
        <wsdl:fault name="UnknownTechnicalServiceFault">
            <soap:fault name="UnknownTechnicalServiceFault"
use="literal"/>
        </wsdl:fault>
    </wsdl:operation>
</wsdl:binding>
<wsdl:service name="SilPublishService">
    <wsdl:port name="SilPublish" binding="tns:SilPublishBinding">
        <soap:address location="http://DUMMY_VALUE"/>
    </wsdl:port>
</wsdl:service>
<wsp:Policy wsu:Id="AddressingPolicy">
    <wsam:Addressing>
        <wsp:Policy>
            <wsam:AnonymousResponses/>
        </wsp:Policy>
    </wsam:Addressing>
</wsp:Policy>
<wsp:Policy wsu:Id="SecurityPolicy">
    <sp:Wss11>

```

```

        <wsp:Policy>
          <sp:MustSupportRefKeyIdentifier/>
        </wsp:Policy>
      </sp:Wss11>
    <sp:SignedParts/>
    <sp:EncryptedParts/>
    <sp:AsymmetricBinding>
      <wsp:Policy>
        <sp:InitiatorToken>
          <wsp:Policy>
            <sp:X509Token sp:IncludeToken="http://docs.oasis-
open.org/ws-sx/ws-securitypolicy/200702/IncludeToken/AlwaysToRecipient"/>
          </wsp:Policy>
        </sp:InitiatorToken>
        <sp:RecipientToken>
          <wsp:Policy>
            <sp:X509Token sp:IncludeToken="http://docs.oasis-
open.org/ws-sx/ws-securitypolicy/200702/IncludeToken/Never"/>
          </wsp:Policy>
        </sp:RecipientToken>
        <sp:AlgorithmSuite>
          <wsp:Policy>
            <sp:Basic256/>
          </wsp:Policy>
        </sp:AlgorithmSuite>
        <sp:Layout>
          <wsp:Policy>
            <sp:Lax/>
          </wsp:Policy>
        </sp:Layout>
        <sp:IncludeTimestamp/>
        <sp:EncryptSignature/>
        <sp:OnlySignEntireHeadersAndBody/>
      </wsp:Policy>
    </sp:AsymmetricBinding>
  </wsp:Policy>
</wsdl:definitions>

```

C.3 SilLookupInterface.wsdl

```

<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions
xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:tns="urn:xml-gov-
au:nehta:service:ServiceInstanceLocator:SilLookup:1.0"
xmlns:sdt="urn:xml-gov-au:nehta:types:SilDataTypes:1.0"
xmlns:smi="urn:xml-gov-
au:nehta:types:SecureMessaging:1.0"
name="SilLookup"
targetNamespace="urn:xml-gov-
au:nehta:service:ServiceInstanceLocator:SilLookup:1.0">
  <wsdl:types>
    <xsd:schema targetNamespace="urn:xml-gov-
au:nehta:service:ServiceInstanceLocator:SilLookup:1.0"
elementFormDefault="qualified">
      <xsd:import namespace="urn:xml-gov-
au:nehta:types:SilDataTypes:1.0"
schemaLocation="../xsd/SilDataTypes.xsd"/>
      <xsd:import namespace="urn:xml-gov-
au:nehta:types:SecureMessaging:1.0"
schemaLocation="../../smi/xsd/SecureMessaging.xsd"/>
      <xsd:element name="lookupBsirs">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="serviceOwnerId" type="smi:OrgId"/>
            <xsd:element name="serviceName" type="xsd:string"
minOccurs="0"/>
            <xsd:element name="status" type="sdt:Status"/>
            <xsd:element name="patternName" type="xsd:string"
minOccurs="0"/>

```

```

        <xsd:element name="version" type="xsd:string"
minOccurs="0"/>
        <xsd:element name="wsdlType" type="smi:Wsdlnamespace"
minOccurs="0"/>
        <xsd:element name="noCache" type="xsd:boolean"/>
    </xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="lookupBsirsResponse">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="LookupResult"
type="sdt:BusinessServiceInstanceRecord" minOccurs="0"
maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
<xsd:element name="lookupTsirs">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="serviceHostId" type="smi:OrgId"/>
            <xsd:element name="serviceOwnerId" type="smi:OrgId"/>
            <xsd:element name="status" type="sdt:Status"/>
            <xsd:element name="wsdlType" type="smi:Wsdlnamespace"
minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
</xsd:schema>
</wsdl:types>
<wsdl:message name="lookupBsirsInMsg">
    <wsdl:part name="parameters" element="tns:lookupBsirs"/>
</wsdl:message>
<wsdl:message name="lookupBsirsOutMsg">
    <wsdl:part name="parameters" element="tns:lookupBsirsResponse"/>
</wsdl:message>
<wsdl:message name="lookupTsirsInMsg">
    <wsdl:part name="parameters" element="tns:lookupTsirs"/>
</wsdl:message>
<wsdl:message name="lookupTsirsOutMsg">
    <wsdl:part name="parameters" element="tns:lookupTsirsResponse"/>
</wsdl:message>
<wsdl:message name="StandardFaultMsg">
    <wsdl:part name="parameters" element="smi:StandardFault"/>
</wsdl:message>
<wsdl:portType name="SilLookup">
    <wsdl:operation name="lookupBsirs">
        <wsdl:input message="tns:lookupBsirsInMsg"
wsam:Action="urn:xml-gov-
au:nehta:service:ServiceInstanceLocator:SilLookup:1.0:SilLookup:lookupBsi
rsRequest"/>
        <wsdl:output message="tns:lookupBsirsOutMsg"
wsam:Action="urn:xml-gov-
au:nehta:service:ServiceInstanceLocator:SilLookup:1.0:SilLookup:lookupBsi
rsResponse"/>
        <wsdl:fault name="StandardFault" message="tns:StandardFaultMsg"
wsam:Action="urn:xml-gov-
au:nehta:service:ServiceInstanceLocator:SilLookup:1.0:SilLookup:lookupBsi
rs:Fault:StandardFault"/>
    </wsdl:operation>
    <wsdl:operation name="lookupTsirs">
        <wsdl:input message="tns:lookupTsirsInMsg"
wsam:Action="urn:xml-gov-
au:nehta:service:ServiceInstanceLocator:SilLookup:1.0:SilLookup:lookupTsi
rsRequest"/>

```

```

        <wsdl:output message="tns:lookupTsirsOutMsg"
            wsam:Action="urn:xml-gov-
au:nehta:service:ServiceInstanceLocator:SilLookup:1.0:SilLookup:lookupTsi
rsResponse"/>
        <wsdl:fault name="StandardFault" message="tns:StandardFaultMsg"
            wsam:Action="urn:xml-gov-
au:nehta:service:ServiceInstanceLocator:SilLookup:1.0:SilLookup:lookupTsi
rs:Fault:StandardFault"/>
    </wsdl:operation>
</wsdl:portType>
</wsdl:definitions>

```

C.4 SilLookup.wsdl

```

<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap12/"
    xmlns:tns="urn:xml-gov-
au:nehta:service:ServiceInstanceLocator:SilLookup:1.0"
    xmlns:wsp="http://www.w3.org/ns/ws-policy"
    xmlns:wsu="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"

    xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata"
    xmlns:sp="http://docs.oasis-open.org/ws-sx/ws-
securitypolicy/200702"
        name="SilLookup"
        targetNamespace="urn:xml-gov-
au:nehta:service:ServiceInstanceLocator:SilLookup:1.0">
    <wsdl:import namespace="urn:xml-gov-
au:nehta:service:ServiceInstanceLocator:SilLookup:1.0"
        location="SilLookupInterface.wsdl"/>
    <wsdl:binding name="SilLookupBinding" type="tns:SilLookup">
        <wsp:PolicyReference URI="#AddressingPolicy"/>
        <wsp:PolicyReference URI="#SecurityPolicy"/>
        <soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/>
        <wsdl:operation name="lookupBsirs">
            <soap:operation style="document" soapActionRequired="false"/>
            <wsdl:input>
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal"/>
            </wsdl:output>
            <wsdl:fault name="StandardFault">
                <soap:fault name="StandardFault" use="literal"/>
            </wsdl:fault>
        </wsdl:operation>
        <wsdl:operation name="lookupTsirs">
            <soap:operation style="document" soapActionRequired="false"/>
            <wsdl:input>
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal"/>
            </wsdl:output>
            <wsdl:fault name="StandardFault">
                <soap:fault name="StandardFault" use="literal"/>
            </wsdl:fault>
        </wsdl:operation>
    </wsdl:binding>
    <wsdl:service name="SilLookupService">
        <wsdl:port name="SilLookup" binding="tns:SilLookupBinding">
            <soap:address location="http://DUMMY_VALUE"/>
        </wsdl:port>
    </wsdl:service>
    <wsp:Policy wsu:Id="AddressingPolicy">
        <wsam:Addressing>
            <wsp:Policy>
                <wsam:AnonymousResponses/>
            </wsp:Policy>

```

```
</wsam:Addressing>
</wsp:Policy>
<wsp:Policy wsu:Id="SecurityPolicy">
  <sp:Wss11>
    <wsp:Policy>
      <sp:MustSupportRefKeyIdentifier/>
    </wsp:Policy>
  </sp:Wss11>
  <sp:SignedParts/>
  <sp:EncryptedParts/>
  <sp:AsymmetricBinding>
    <wsp:Policy>
      <sp:InitiatorToken>
        <wsp:Policy>
          <sp:X509Token sp:IncludeToken="http://docs.oasis-
open.org/ws-sx/ws-securitypolicy/200702/IncludeToken/AlwaysToRecipient"/>
        </wsp:Policy>
      </sp:InitiatorToken>
      <sp:RecipientToken>
        <wsp:Policy>
          <sp:X509Token sp:IncludeToken="http://docs.oasis-
open.org/ws-sx/ws-securitypolicy/200702/IncludeToken/Never"/>
        </wsp:Policy>
      </sp:RecipientToken>
      <sp:AlgorithmSuite>
        <wsp:Policy>
          <sp:Basic256/>
        </wsp:Policy>
      </sp:AlgorithmSuite>
      <sp:Layout>
        <wsp:Policy>
          <sp:Lax/>
        </wsp:Policy>
      </sp:Layout>
      <sp:IncludeTimestamp/>
      <sp:EncryptSignature/>
      <sp:OnlySignEntireHeadersAndBody/>
    </wsp:Policy>
  </sp:AsymmetricBinding>
</wsp:Policy>
</wsdl:definitions>
```