



Service Instance Locator

Business and Information 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	iv
Change history	iv
1 Introduction	5
1.1 Background	5
1.2 Purpose	5
1.3 Status	5
1.4 Document Map	5
2 Community Model	6
2.1 Community Objective	6
2.2 Summary of Concepts from Information Model	6
2.3 Community Roles	7
2.3.1 Service Owner SIL Role	7
2.3.2 Service Host SIL Role	8
2.3.3 Service Publisher Role	8
2.3.4 Service Lookup Client Role	8
2.4 Privacy Policy	9
2.5 Community Instance Example for Publication	9
3 Business Processes	11
3.1 Publication	11
3.1.1 Publish Technical Service Instance Record	11
3.1.2 Delete Technical Service Instance Record	12
3.1.3 Publish Business Service Instance Record	13
3.1.4 Add Technical Realisation to BSIR	14
3.1.5 Delete Technical Realisation from BSIR	15
3.1.6 Update a Role of a Technical Realisation	16
3.1.7 Delete Business Service Instance Record	17
3.1.8 Lookup All Business Service Instance Records	17
3.1.9 Lookup BSIRs	19
Lookup Cache	19
Lookup SIL	20
3.1.10 Lookup TSIRs	21
4 Information Architecture	23
4.1 Class: <i>QualifiedId</i> (from <i>SecureMessaging</i>)	23
4.2 Class: <i>OrgId</i> (from <i>SecureMessaging</i>)	24
4.3 Class: <i>DirectoryEndpoint</i> (from <i>NASH</i>)	24
4.4 Class: <i>CertificateRef</i> (from <i>NASH</i>)	24
4.5 Enum: <i>Status</i>	25
4.6 Enum: <i>TsirSource</i>	25
4.7 Class: <i>BusinessServiceInstanceRecord</i>	26
4.8 Class: <i>TechnicalRealisation</i>	26
4.8.1 Values for <i>patternName</i>	27
4.9 Class: <i>TechnicalRole</i>	29
4.10 Abstract Class: <i>TechnicalServiceCommon</i>	30
4.11 Class: <i>TechnicalServiceEndpoint</i>	30
4.12 Class: <i>TechnicalServiceRef</i>	31
4.13 Class: <i>TechnicalServiceInstanceRecord</i>	32
Appendix A: References	34

Document information

Change history

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

1 Introduction

1.1 Background

The Service Instance Locator has been identified as a piece of key National E-Health Infrastructure. An application that is attempting to establish communications with a business service will use the Service Instance Locator to locate the correct technical realisation of that business service and find the correct endpoint(s).

1.2 Purpose

This document describes the community model and business processes, business services, and information model that relate to the Service Instance Locator.

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. This document contains two logically separate pieces of work: The Business Architecture, and the Information Architecture. These two are better read together than separately.

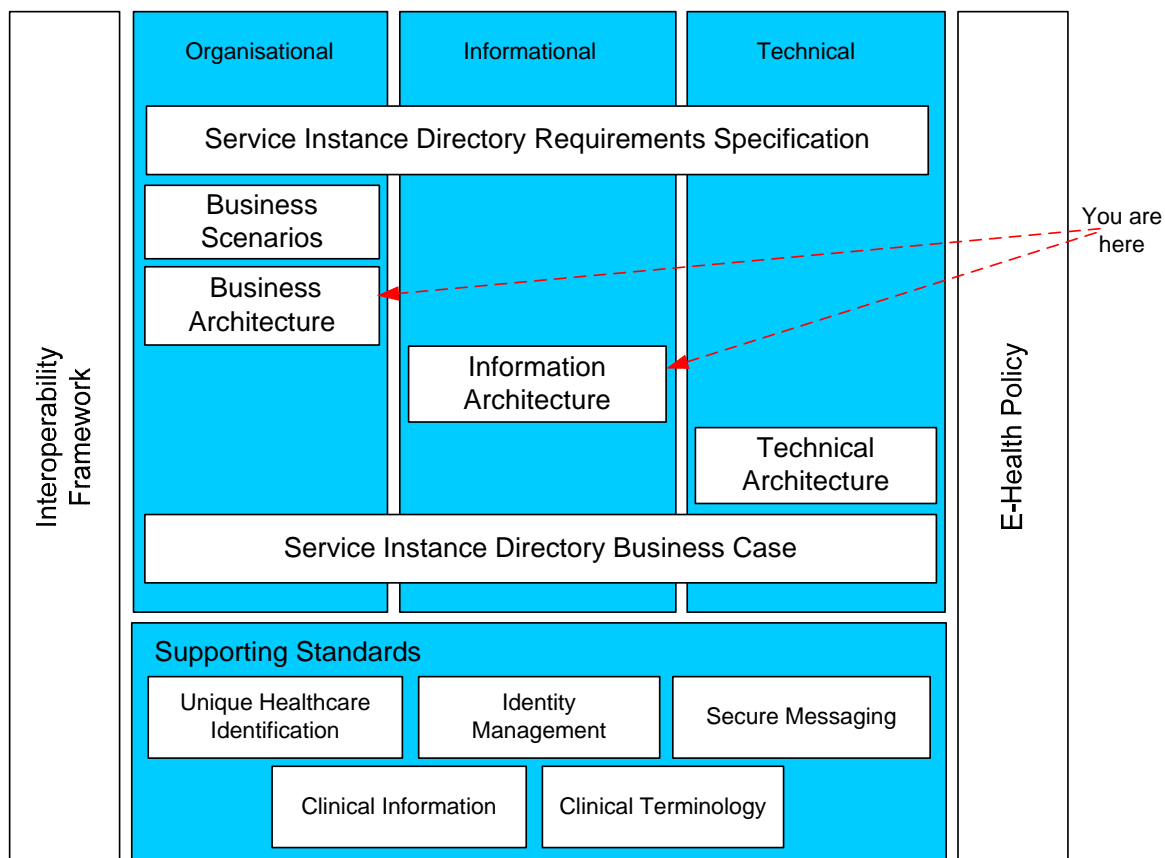


Figure 1: Document Map

2 Community Model

This section explains the community model and the objectives of the Service Instance Locator community, and gives details about the roles that interact in this community.

2.1 Community Objective

The objective of the Service Instance Locator community is to facilitate the publication and discovery of web services endpoints that realise business services offered by members of the e-health community. Publishers will advertise business services that they own, or technical service endpoints that they host to their SIL. Clients wishing to use services will lookup the SIL for an organisation to discover services of particular business and/or technical types and where they are hosted.

2.2 Summary of Concepts from Information Model

In order to describe the community and its business processes, we introduce the key terms from the Information Architecture (Section 4).

Business Service Instance Record (BSIR)

A record which describes a business service offered by an organisation, including references to the technical services that realises the business service.

Technical Service Instance Record (TSIR)

The record kept by the SIL of the type and endpoint information about a web service instance hosted by an organisation. The type is specified as a WSDL namespace, and the endpoint is a web services URI.

TSIRs that are created at the same time as a BSIR by the same organisation, for the BSIR's exclusive reference, and are in the same SIL are called *Dependent TSIRs*. Other TSIRs are called *Independent TSIRs* and their lifecycle are not connected to the lifecycles of any BSIRs that refer to them.

Technical Realisation

A named pattern by which a business service can be accessed using one or more technical roles, each of which has an endpoint. A BSIR may have one or more technical realisations. The technical realisation has named *Technical Roles* most of which contain references to technical services that implement the role.

Technical Role

A description of one of possibly several roles within a *Technical Realisation*. Most *Technical Roles* have a *TSIR Reference* which facilitates the lookup of the TSIR containing the endpoint which plays the role, but some roles will be determined at "message time" as the result of information in a notification. Roles may also refer to the PKI certificates of the party that will play the role, even if the endpoint will be determined later.

TSIR Reference

The information needed to lookup a TSIR in the (possibly remote) SIL where it is stored, including the host, PKI Certificate and endpoint of that SIL, and the WSDL type and status of the TSIR.

2.3 Community Roles

Instances of the Service Instance Locator may play either or both of the roles *Service Owner SIL* and *Service Host SIL*. Instances of the SIL playing the *Service Owner SIL* role will also play the *Service Lookup Client* role, as the business services they hold contain references to the technical services that realise them, and the *Service Owner SIL* will look up the endpoint details of these technical services in a SIL playing the *Service Host SIL* role.

When an organisation hosts its own services, that organisation's SIL instance plays all three roles: *Service Owner SIL*, or *Service Host SIL*, and *Service Lookup Client*. However, when an organisation outsources its technical service hosting, its SIL will play the *Service Owner SIL* and *Service Lookup Client* roles, and the service provider organisation that it outsources to will have another SIL instance that plays the *Service Host SIL* role. This means that each organisation only publishes the records that it is responsible for into its own SIL, and that *Service Owner SILs* have references to *Service Host SILs* so that they can lookup the current technical endpoint information stored and maintained there by the hosts of the technical services.

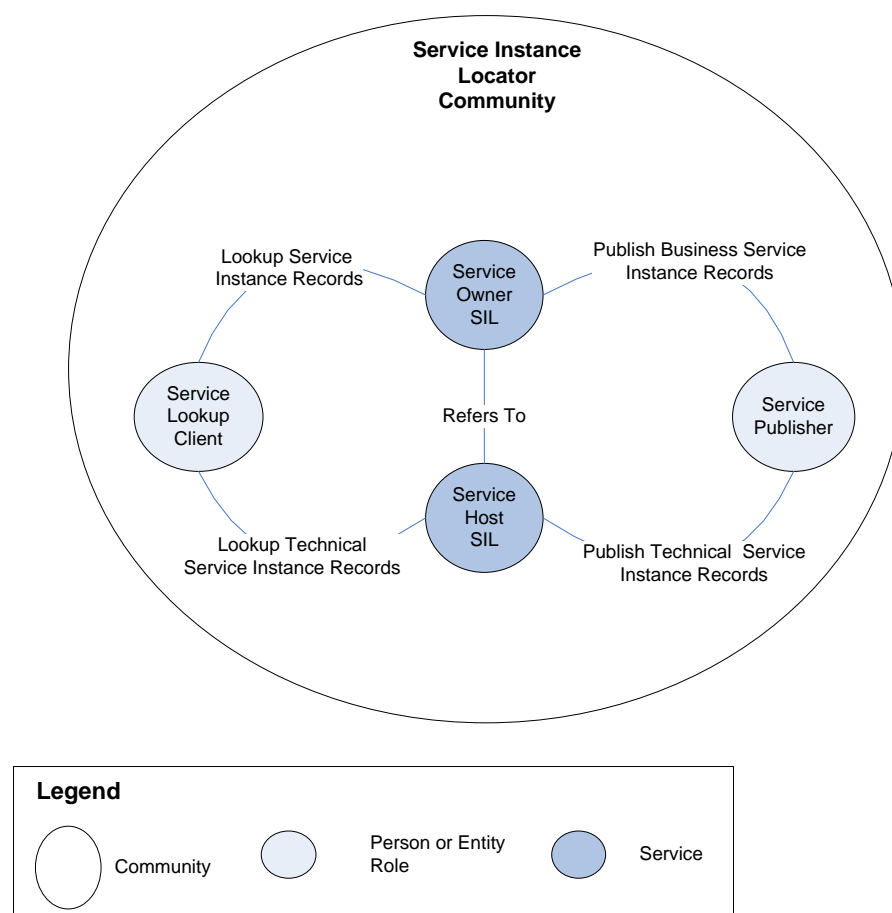


Figure 2: Service Instance Locator Community Model

2.3.1 Service Owner SIL Role

The *Service Owner SIL* role is a service that accepts publications of BSIRs from a *Service Publisher*, which describe business services and their technical realisations according to the patterns documented in [CPIS2008]. Each pattern of service realisation makes reference to one or more roles in the pattern which is filled by a Web services endpoint. A *Service Host SIL* will contain TSIRs which contain all the information needed to bind to, and securely invoke, a Web service endpoint. Each role description of a supported technical realisation of a business service contains a logical reference to a TSIR in the *Service Host SIL* of the service host organisation.

When requested to return details of a BSIR, a SIL will be responsible for making calls (acting in the *Service Lookup Client* role) to the *Service Host SILs* to which the business service refers, so that all details about a business service's realisation(s) can be obtained in a single lookup.

This role is filled by computing services when instantiated, and hence is shown in Figure 2 as services, rather than person or entity roles.

2.3.2 Service Host SIL Role

The *Service Host SIL* role is a service that accepts publications of TSIRs from a service provider organisation acting in the *Service Publisher* role. These records always describe Web service endpoints that are hosted by the organisation operating the SIL.

When requested by a *Service Lookup Client* role (which may be played by a *Service Owner SIL*, or a Web services end-user), the *Service Host SIL* will lookup and return TSIRs using a set of criteria.

2.3.3 Service Publisher Role

The *Service Publisher* is the party who is responsible for "advertising" the business services, and/or their technical realisations that are available from a healthcare provider organisation, as identified by a unique organisational identifier. A *Service Publisher* uses a SIL instance playing the *Service Owner SIL* role to publish details of all the business services offered by an organisation, as well as the references required to technical services which realise the business service. When an organisation hosts its own technical services, these references will be to TSIRs in the same organisation's SIL (i.e. this single SIL instance is playing both *Service Host SIL* and *Service Owner SIL* roles), but they may also be references to records in the SIL of another organisation that hosts a technical service on behalf of the business service owner (i.e. the Service Provider's SIL plays the *Service Host SIL* role).

Some *Service Publishers* will only publish TSIRs containing Web services endpoints that they host on behalf of a range of other organisations. A TSIR will be published into the SIL of the organisation that hosts the Web service that it represents, whether hosted on its own behalf, or on behalf of another organisation. TSIRs published may be referred to by BSIRs in the SIL of the organisation that "owns" the service, i.e. the organisation on behalf of whom it is hosted. Organisations that offer applications or middleware services on behalf of health provider organisations may publish only TSIRs into their SILs.

2.3.4 Service Lookup Client Role

The *Service Lookup Client* initiates the lookup of service endpoints by invoking lookup operations at the *Service Owner SIL* of the organisation with whom they wish to communicate. When attempting communication for the first time, the *Service Lookup Client* may either request all information about a business service, and choose an appropriate technical realisation of the business service with which to communicate. *Service Lookup Clients* may also request only BSIRs that have a particular pattern of technical realisation, or even support a particular WSDL interface. Once they have been given the BSIR of interest, and any TSIR information that it refers to, they may also lookup TSIRs directly from the *Service Host SIL*.

In some communications patterns a set of lookup criteria for a TSIR containing a Web services endpoint for a document *get()* interface may be included in a notification. In these cases, the TSIR representing this endpoint may be looked up by a *Service Lookup Client* at a *Service Host SIL* directly, as its business service context is already known by the client.

The *Service Lookup Client* role may also be played by a *Service Owner SIL* when looking up Technical Service Instance Records that fill roles in patterns of business service realisations at a *Service Host SIL*.

2.4 Privacy Policy

The Privacy Assessment for the SIL has determined that there is no privacy impact of allowing the Service Instance Records from any, or all, SIL Directory Servers to be public information. However, the Web services profile deployed will ensure that all records published to, and looked up from a SIL will be

- From an authenticated Organisational Identity, and
- Safe from tampering;

Thereby assuring that the records looked up contain the same information that is published about the structure and location of business services delivered through one or more technical services.

2.5 Community Instance Example for Publication

In Figure 3 there are three SILs, of which two represent healthcare provider organisations offering business services to the community, and the third represents an infrastructure provider which offers only technical services. The two example BSIRs (yellow boxes) each have two Technical Realisations which are tagged with pattern names corresponding to the Interaction Patterns described in the Concepts and Patterns for Implementing Services [CPIS2008]. For example: "Put", "Get with notifications", "Store with notifications" and "Store". The Interaction Patterns specify the use of one or two web services operations to achieve the business goal of transmitting a document from a sender to a recipient. The operations are available through web services interfaces, each of which has its endpoint advertised through a TSIR which lives in the SIL of the service host.

In the case of a "Get with notifications" interaction, the TSIR for the notification endpoint will be referred to by the BSIR of the intended recipient (e.g. Org Y's BSIR Technical Realisation 1), and the TSIR for the *get* endpoint will be in the SIL of the sender. Whereas in the case of a "Store with notifications" interaction (e.g. Org Y's BSIR Technical Realisation 2), both the notification and store endpoints will be referred to by the BSIR.

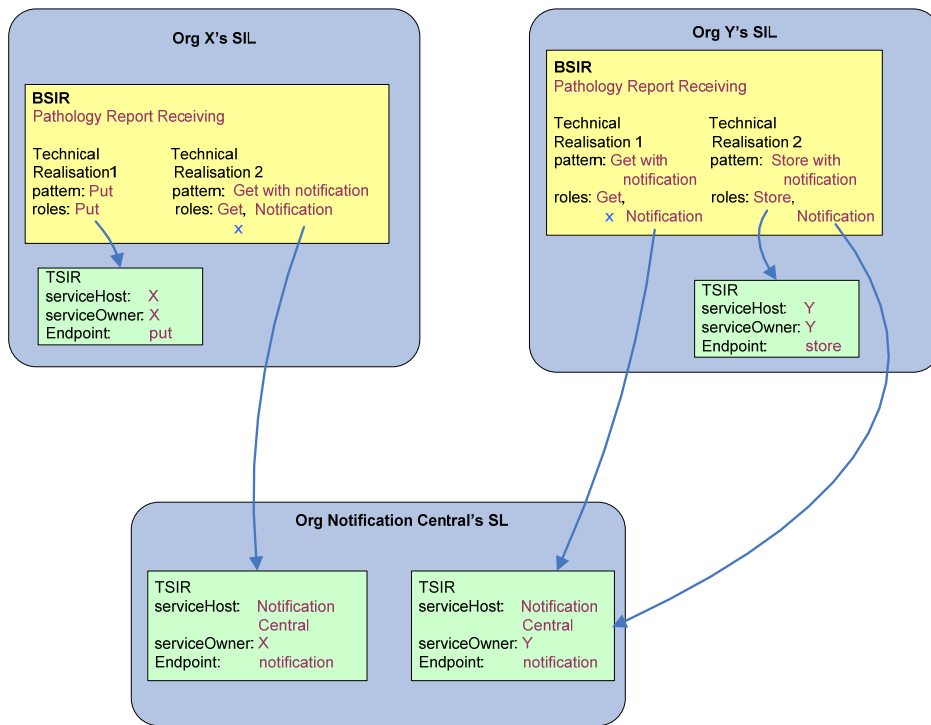


Figure 3: Example SIL Community Instance for Publication

3 Business Processes

3.1 Publication

3.1.1 Publish Technical Service Instance Record

3.1.1.1 Description

This process allows Service Publishers to make a Technical Service Instance Record (TSIR) available through the SIL of the technical service's host.

3.1.1.2 Preconditions

No other TSIR with the same key information as the new TSIR may already be published in this SIL.

3.1.1.3 Notes

The published TSIR will be an independent TSIR, and its lifecycle is not related to that of a BSIR.

3.1.1.4 Steps

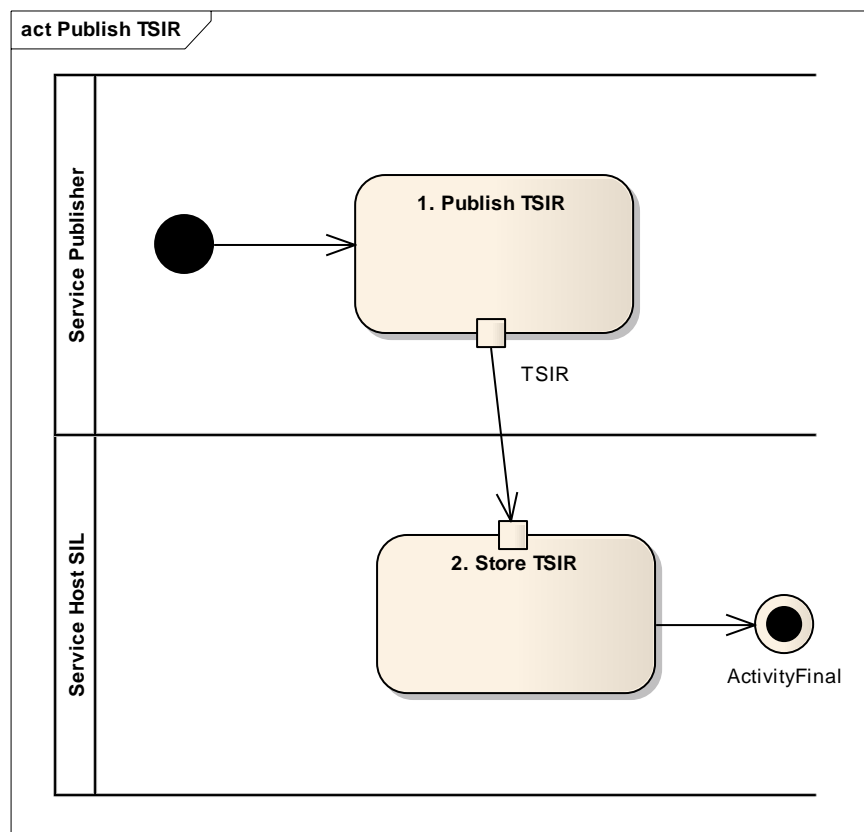


Figure 4: Publish TSIR Business Process

1. The Service Publisher creates a Technical Service Instance Record (TSIR) and publishes it to the Service Host's SIL.
2. The SIL stores the TSIR and makes it available to Service Lookup Clients in the SIL community.

3.1.2 Delete Technical Service Instance Record

3.1.2.1 Description

This process allows for published Technical Service Instance Records (TSIRs) to be deleted from a SIL.

3.1.2.2 Preconditions

The TSIR corresponding to the key parameters that are passed between the Technical Service Publisher and the Service Host's SIL must already exist in the SIL.

The TSIR MUST NOT be a dependent TSIR that is attached to a BSIR in the same SIL.

3.1.2.3 Notes

The Service Instance Locator Information Architecture (see Section 4) specifies what attributes of a Technical Service Instance Record constitute a key.

3.1.2.4 Steps

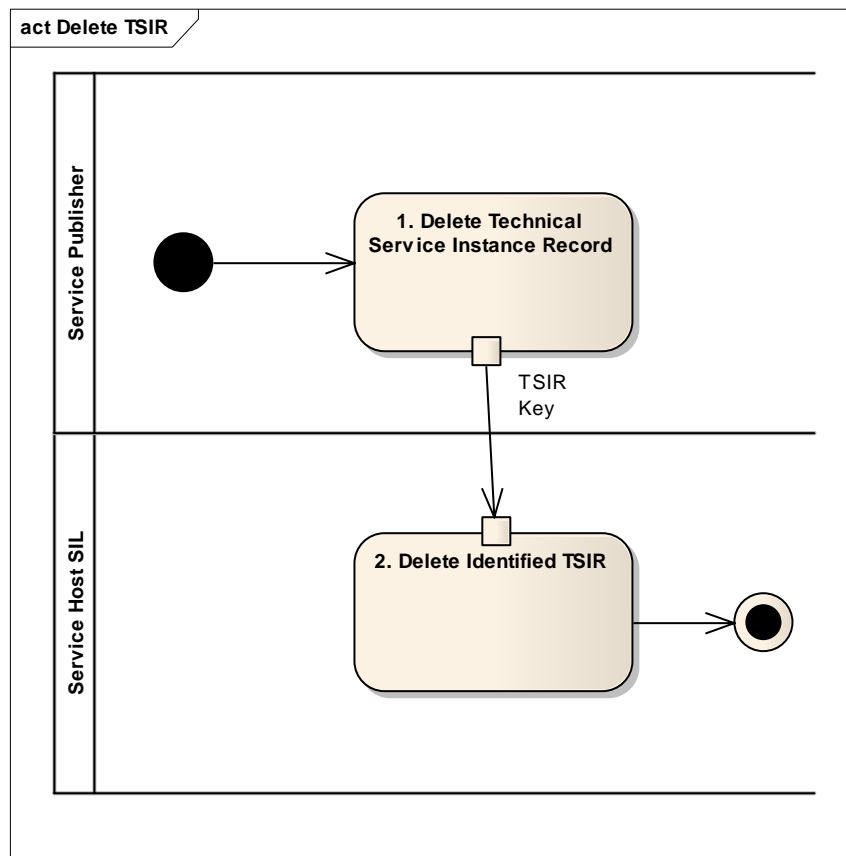


Figure 5: Delete TSIR Business Process

1. The Service Publisher requests deletion of an identified TSIR from the Service Host's SIL.
2. The SIL deletes the TSIR that corresponds to the TSIR Key that the Publisher nominates.

3.1.3 Publish Business Service Instance Record

3.1.3.1 Description

This process facilitates the publication of a BSIR into the SIL of the Service Owner, and also publishes any dependent TSIRs contained within the BSIR in the same SIL.

3.1.3.2 Preconditions

No BSIR with the same key data must already be published into the SIL.

3.1.3.3 Notes

The Service Instance Locator Information Architecture (see Section 4) specifies what attributes of a BSIR constitute a key.

3.1.3.4 Steps

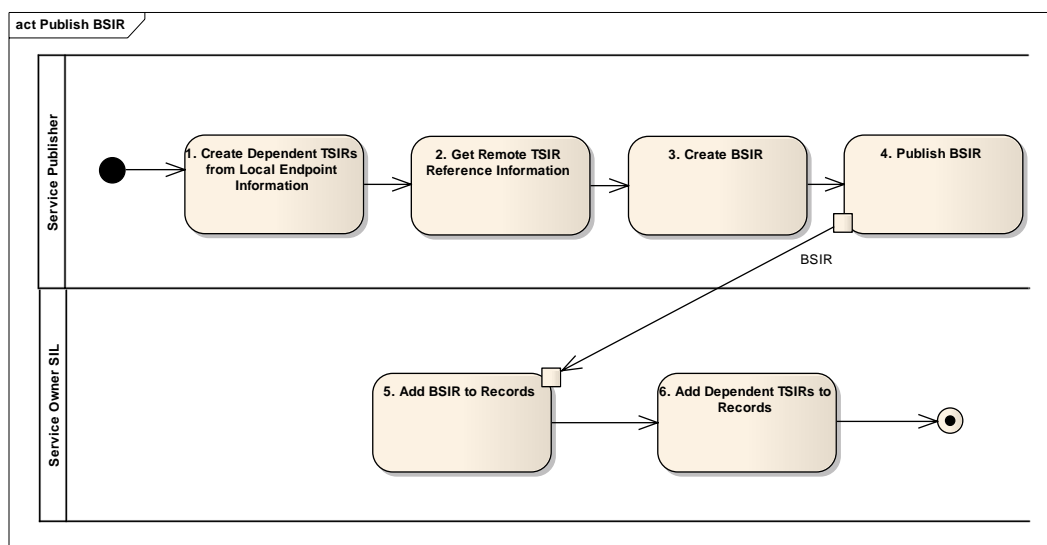


Figure 6: Publish BSIR Process

1. The Service Publisher finds local endpoint information for technical services that are to be hosted locally which will make up endpoints for any technical realisation of the Business Service to be published. This information is populated into a (dependent) TSIR, which will be attached to the BSIR (in step 4).
2. The Service Publisher finds remote TSIR reference information for technical services that are to be hosted remotely (i.e. services whose TSIRs will appear in the SIL for a remote host of the services on behalf of the service owner) and populate one or more technical realisations of the Business Service to be published. Usually TSIRs that will be located using these references will already have been published into the Technical Service Host's SIL.
3. The Service Publisher creates a BSIR using the dependent TSIRs created in step (1), and references to TSIRs gathered in step (2), to create one or more *Technical Realisations* for the business service, and populate these into the BSIR along with metadata about the records themselves.
4. The Business Service Publisher publishes the BSIR (with attached dependent TSIRs) by invoking the correct operation on its organisation's SIL.
5. The SIL must add the details of the BSIR to its records, and make this BSIR available to Service Lookup Clients.

6. The SIL must extract any dependent TSIRs that are part of the BSIR's technical realisations, and publish them locally as a set of TSIRs which must be made available to Service Lookup Clients.

3.1.4 Add Technical Realisation to BSIR

3.1.4.1 Description

This process facilitates the addition of a new technical realisation to a BSIR already published to the SIL of the Service Owner, and also publishes any attached dependent TSIRs in the same SIL.

3.1.4.2 Preconditions

The identified BSIR must already exist in the target SIL.

The identified BSIR must NOT already contain a technical realisation with the same pattern name and version as the new technical realisation to be added.

3.1.4.3 Notes

The Service Instance Locator Information Architecture (see Section 4) specifies what attributes of a BSIR constitute a key.

3.1.4.4 Steps

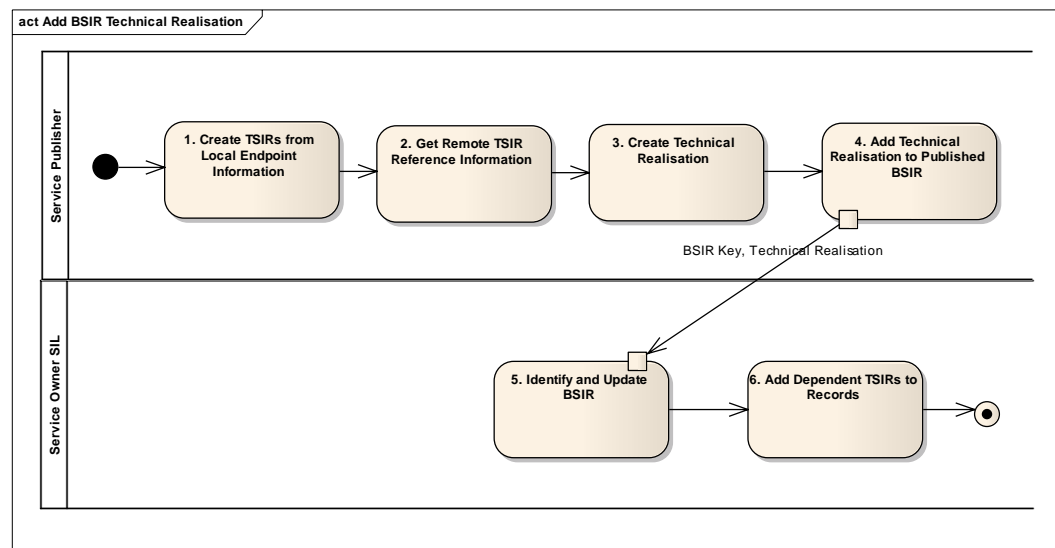


Figure 7: Add Technical Realisation to BSIR Process

1. The Publisher finds local endpoint information for technical services that are to be hosted locally which will make up endpoints for the new technical realisation to be added to a BSIR already published. New (dependent) TSIRs are created for these endpoints.
2. The Publisher finds remote technical service reference information for technical services that are to be hosted remotely that fill a role of the technical realisations to be added to a BSIR already published. Usually TSIRs that will be located using these references will already have been published into the Technical Service Host's SIL.
3. The Publisher creates a Technical Realisation using the dependent TSIRs and references to TSIRs gathered in steps (1) and (2) and identifies the BSIR to which they should be added.
4. The Business Service Publisher requests that the SIL adds the new Technical Realisation to the identified BSIR by invoking the correct operation.

5. The SIL must add the Technical Realisation to the BSIR Identified and make the updated BSIR available to Service Lookup Clients.
6. The SIL must extract any dependent TSIRs and these must be made available to Service Lookup Clients.

3.1.5 Delete Technical Realisation from BSIR

3.1.5.1 Description

This process facilitates the deletion of an existing technical realisation that is part of a BSIR already published to the SIL of the Service Owner.

3.1.5.2 Preconditions

The identified BSIR must already exist in the target SIL.

The identified BSIR must already contain another technical realisation other than the one to be deleted, otherwise this process will fail as the operation on the SIL may not violate the structural constraints of BSIRs by reducing their set of Technical Realisations to empty.

3.1.5.3 Notes

Any dependent TSIRs attached to the nominated Technical Realisation **MUST** be deleted with the Technical Realisation.

The Service Instance Locator Information Architecture (see Section 4) specifies what attributes of a BSIR constitute a key.

3.1.5.4 Steps

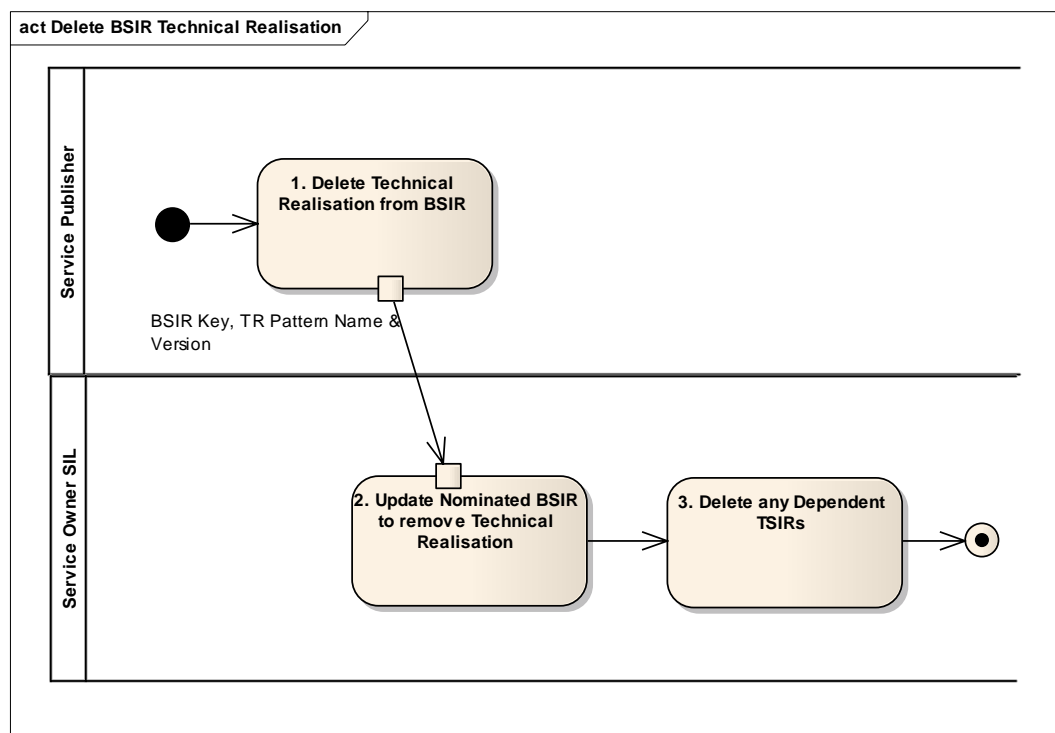


Figure 8: Delete Technical Realisation from BSIR Process

1. The Publisher identifies a BSIR to be changed, and the name of the pattern of a Technical Realisation to be deleted from it, and invokes the appropriate operation.
2. The SIL must remove the named Technical Realisation from the BSIR identified and make the updated BSIR available to Service Lookup Clients.

3. The SIL must remove any dependent TSIRs attached to the Technical Realisation.

3.1.6 Update a Role of a Technical Realisation

3.1.6.1 Description

This process facilitates the update of a Role of Technical Realisation of a BSIR. The new Role may contain a dependent TSIR, or a reference to a TSIR. Any dependent TSIRs are also published. The existing Role, and its possible dependent TSIRs will be deleted.

3.1.6.2 Preconditions

The identified BSIR must already exist in the target SIL.

The named technical realisation pattern must already exist in the nominated BSIR, and it must contain the nominated Role.

3.1.6.3 Notes

The Service Instance Locator Information Architecture (see Section 4) specifies what attributes of a BSIR constitute a key.

3.1.6.4 Steps

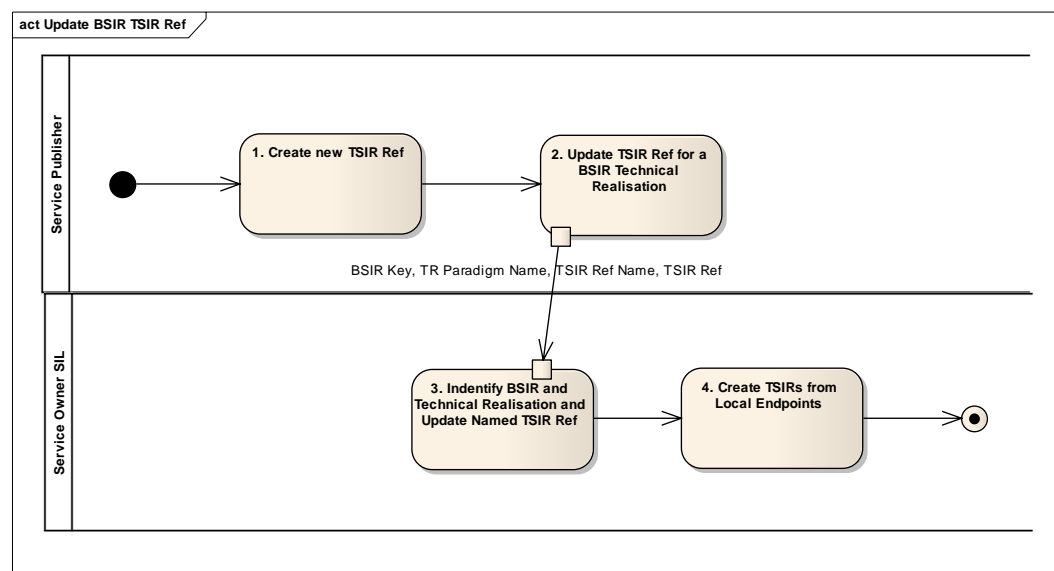


Figure 9: Update BSIR TSIR Reference Process

1. The Publisher creates a new Technical Role data structure which may refer to a remote TSIR, or may attach a dependent TSIR. The Publisher wishes to direct Service Clients to use the new Web service referred to by the referenced or contained TSIR in place of the one in the current Technical Role in a nominated BSIR.
2. The Business Service Publisher requests that the SIL updates a nominated Role of a Technical Realisation by invoking the correct operation.
3. The SIL must delete a dependent TSIR if one is contained by the existing Role.
4. If the new Role contains a dependent TSIR, the SIL must store the TSIR and it must be made available to Service Lookup Clients.

3.1.7 Delete Business Service Instance Record

3.1.7.1 Description

This process allows for published Business Service Instance Records (BSIRs) to be deleted from a SIL.

3.1.7.2 Preconditions

The BSIR corresponding to the key parameters that are passed between the Business Service Publisher and the Service Owner's SIL must already exist in the SIL.

3.1.7.3 Notes

The Service Instance Locator Information Architecture (see Section 4) specifies what attributes of a Business Service Instance Record constitute a key.

3.1.7.4 Steps

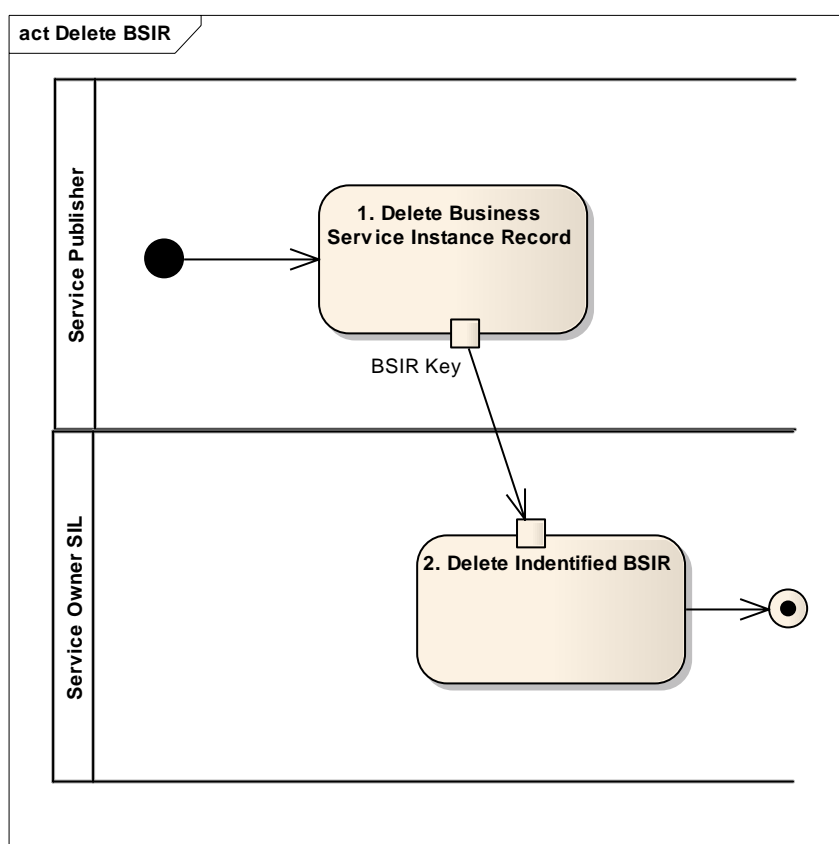


Figure 10: Delete BSIR Business Process

1. The Business Service Publisher requests deletion of an identified BSIR from the Service Owner's SIL.
2. The SIL deletes the BSIR that corresponds to the BSIR Key that the Publisher nominates.

3.1.8 Lookup All Business Service Instance Records

3.1.8.1 Description

This process allows a Service Client which has possession of an Organisational Identifier to find all the Business Service Instance Records (BSIRs) offered by the organisation identified. The information model for BSIRs is given in the Service Instance Locator Information Architecture (see Section 4).

3.1.8.2 Preconditions

The SIL for the organisation required is running and populated with BSIRs.

3.1.8.3 Notes

It is possible for the endpoint for the SIL to resolve to a Server that hosts SIL information for many organisations, and so the Organisational Identifier is always the first parameter of any lookup operation.

The SIL Information Architecture defines the contents of a SIL Service as a set of TSIRs and BSIRs. This implies that there is no specific ordering to the SIRs that will be returned.

3.1.8.4 Steps

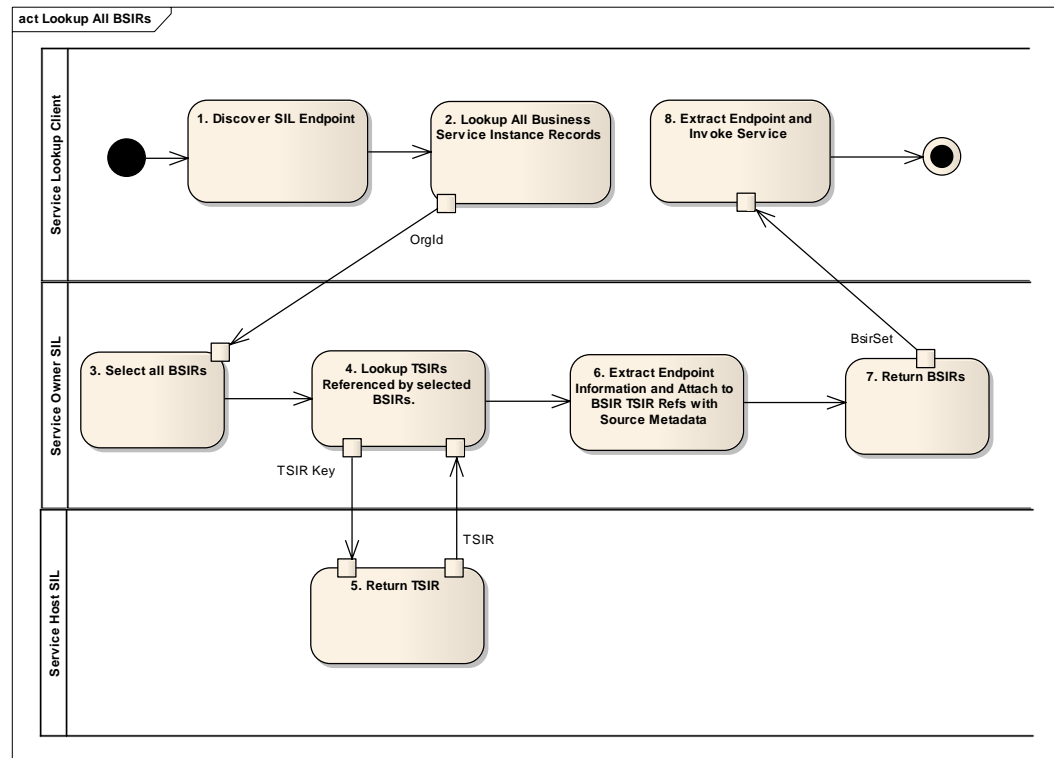


Figure 11: Lookup All BSIRs Business Process

1. The Service Lookup Client first discovers the endpoint for an organisation's SIL. This is done by an out-of-band mechanism, and will eventually be facilitated by the UHI Provider Directory, which will contain a URL for each organisation where its SIL can be located, as well as the OrgId of the organisation hosting the SIL.
2. The Service Lookup Client requests the organisation's SIL to return it all the BSIRs for that OrgId.
3. The SIL selects all the BSIRs for this organisation.
4. For each TSIR Reference in each Technical Realisation of the selected BSIRs:
 - a. check whether the reference is to a TSIR in this SIL, or
 - b. check whether there is a cached TSIR corresponding to the TSIR Ref, or
 - c. make a lookup to retrieve the TSIR from the remote SIL, using the fields of the TSIR Ref. Cache the TSIR. Or
 - d. If the lookup fails, note that the TSIR was unavailable.
5. The SIL for the Service Host returns the requested TSIR.

6. The Service Owner's SIL extracts the endpoint information from the TSIRs found in (4), and attaches this to the TSIR Refs in the selected BSIRs. It also adds metadata to indicate whether the independent TSIRs were cached, retrieved, or unavailable.
7. Return the BSIR.
8. Use the endpoint information in the chosen technical realisation to invoke a web service (or services).

3.1.9 Lookup BSIRs

3.1.9.1 Description

This process allows a Service Client which has possession of an Organisational Identifier to find all the Business Service Instance Records (BSIRs) offered by the organisation identified, restricted to those that match the parameters.

3.1.9.2 Preconditions

The SIL for the organisation required is running and populated with BSIRs.

3.1.9.3 Notes

It is possible for the endpoint for the SIL to resolve to a Server that hosts SIL information for many organisations, and so the Organisational Identifier is always the first parameter of any lookup operation.

The SIL Information Architecture defines the contents of a SIL Service as a set of TSIRs and BSIRs. This implies that there is no specific ordering to the SIRs that will be returned.

3.1.9.4 Steps

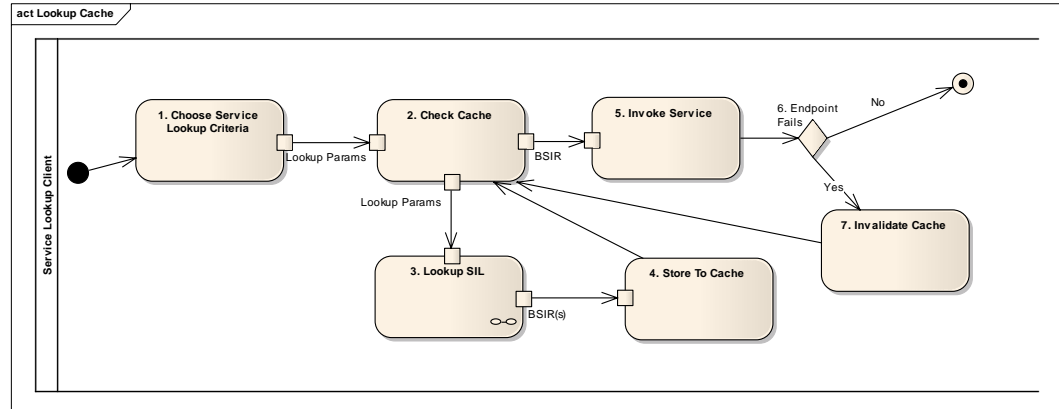


Figure 12: Lookup Cache

Lookup Cache

1. Choose Service Lookup Criteria. The Lookup operations support selection of BSIRs based on Business Service Type, Status, Realisation Pattern, WSDL Interface. In this step the querying application selects which of these criteria it wishes to select for lookup.
2. Check Cache. A BSIR which matches the criteria may be already present in the cache. If so, then go to step 6, if not, then go to step 3.
3. Lookup SIL. This is a composite activity. See next process description.
4. Store to Cache. The BSIR(s) that are returned from the Lookup are stored to the cache, and then we return to step 2, which should result in that activity passing BSIR(s) to step 5.
5. Extract Endpoint & Invoke Service.

6. If the Endpoint Fails to bind to the service expected, then go to 7, otherwise end process.
7. Invalidate Cache. The BSIR which contained the endpoint which failed to bind must be made invalid in the cache. Return to step 2. If there are other possible values in the cache that match the lookup criteria, this may result in them being attempted. Otherwise the cache check will fail, resulting in a fresh SIL lookup.

Lookup SIL

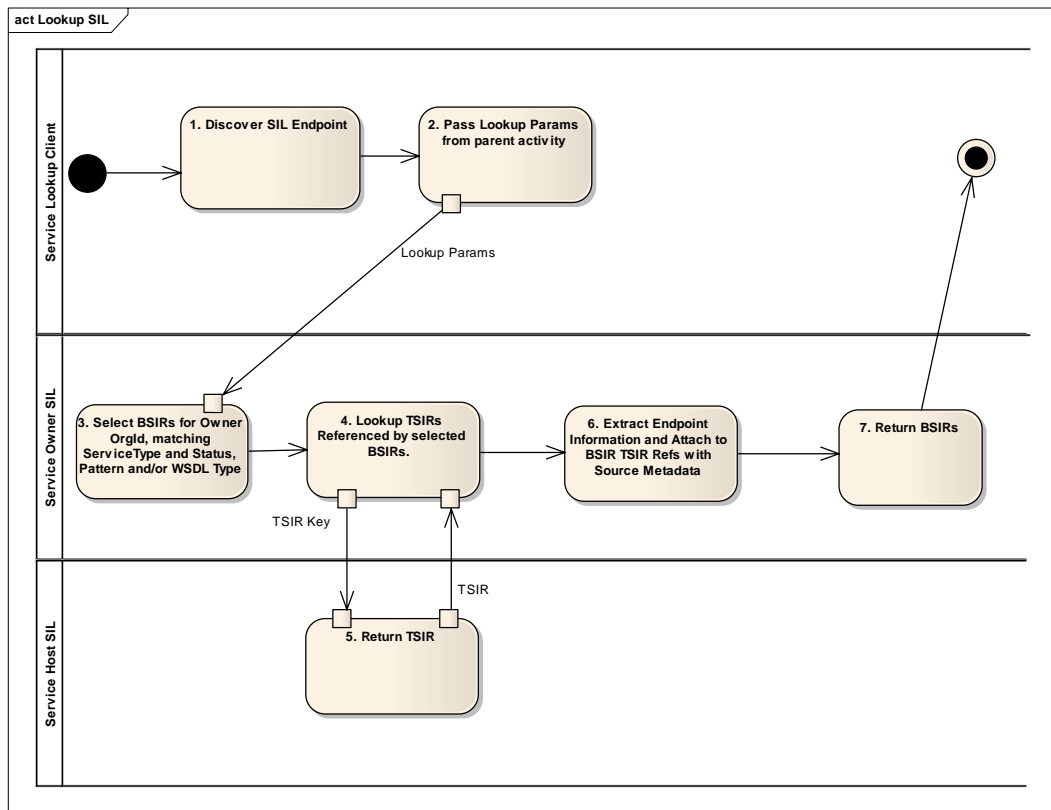


Figure 13: Lookup SIL

1. The Service Lookup Client first discovers the endpoint for an organisation's SIL. This is done by an out-of-band mechanism, and will eventually be facilitated by the UHI Provider Directory, which will contain a URL for each organisation where its SIL can be located, as well as the OrgId of the organisation hosting the SIL.

Invoke the lookup operation on the organisation's SIL to return all the matching BSIRs for the lookup criteria.

2. The SIL selects the BSIRs which match the parameters provided.
3. For each TSIR Reference in each Technical Realisation of the selected BSIRs:
 - a. check whether the reference is to a TSIR in this SIL, or
 - b. check whether there is a cached TSIR corresponding to the TSIR Ref, or
 - c. make a lookup to retrieve the TSIR from the remote SIL, using the fields of the TSIR Ref. Cache the TSIR. Or
 - d. If the lookup fails, note that the TSIR was unavailable.
4. The SIL for the Service Host returns the requested TSIR.
5. The Service Owner's SIL extracts the endpoint information from the TSIRs found in (4), and attaches this to the TSIR Refs in the selected

BSIRs. It also adds metadata to indicate whether the independent TSIRs were cached, retrieved, or unavailable.

6. Return the BSIR.

3.1.10 Lookup TSIRs

3.1.10.1 Description

This process allows a Service Lookup Client which has possession of an Organisational Identifier to find the Technical Service Instance Records (TSIR) for services hosted on behalf of that organisation by the owner of the SIL, which match the parameters.

3.1.10.2 Preconditions

The SIL for the technical service host organisation required is running and populated with TSIRs.

3.1.10.3 Notes

It is possible for the endpoint for the SIL to resolve to a Server that hosts SIL information for many organisations, and so when looking up TSIRs the Organisational Identifier of the service host is always the first parameter of any lookup operation.

3.1.10.4 Steps

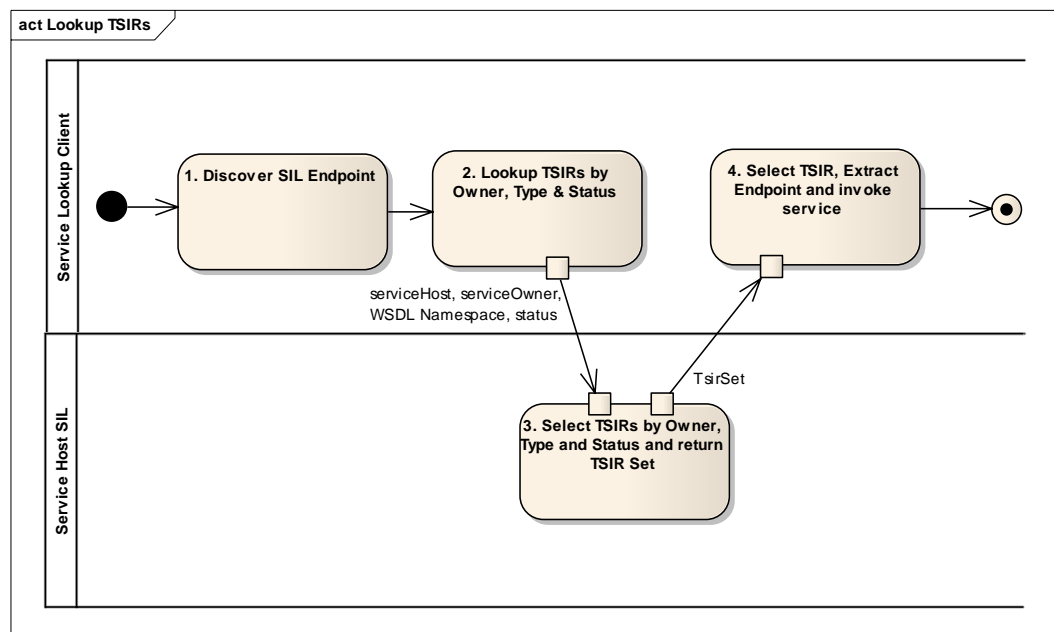


Figure 14: Lookup TSIRs

1. The Service Lookup Client first discovers the endpoint for an organisation's SIL. This may be done
 - without reference to a BSIR by an out-of-band mechanism, and will eventually be facilitated by the UHI Provider Directory, which will contain a URL for each organisation where its SIL can be located, as well as the OrgId of the organisation hosting the SIL
 - by using the information in a TSIR Reference of a BSIR, which contains the SIL endpoint and the OrgId of the SIL host. In cases where a BSIR Lookup returns a TSIR Ref without attached endpoint information (source "Unavailable") the lookup client will need to make its own query at the TSIR's host SIL.

2. The Service Lookup Client chooses the OrgId of a technical service owner (the organisation on behalf of whom the service is being hosted – the ultimate recipient of a business communication). It may then choose to restrict its lookup to a particular service Status or WSDL Type by populating the appropriate parameters. It then requests the SIL to return it the matching TSIRs.
3. The SIL selects the TSIRs with the right service owner (and the right service host), restricts the set to those matching any Status or WSDL Type parameters, and returns a set of TSIRs.
4. The Service Look Client can now inspect the returned TSIRs and extract the endpoint(s) needed to make an invocation.

4 Information Architecture

The SIL has a limited scope, and stores information relating how technical web services configurations realise the business goal of a business service. It allows for multiple technical realisations of a named business service type to be advertised. Once the client of the business service chooses the technical realisation to use, it can locate the technical services it requires and their *endpoint*, *owner*, *host* and *type*, as well as metadata to facilitate caching. The *owner* is the organisation which is the identified target of the business service (for example a GP practice), and is known to the client before it uses the SIL. The *host* is the organisation which runs the technical service (for example a middleware hub), and may be the same as the owner. The *type* is information about technical communications protocols of the service.

The only data stored in an instance of a SIL is a set of Business Service Instance Records (BSIRs) and a set of Technical Service Instance Records (TSIRs). The term "set" is used advisedly, as there is no ordering of the SIRs.

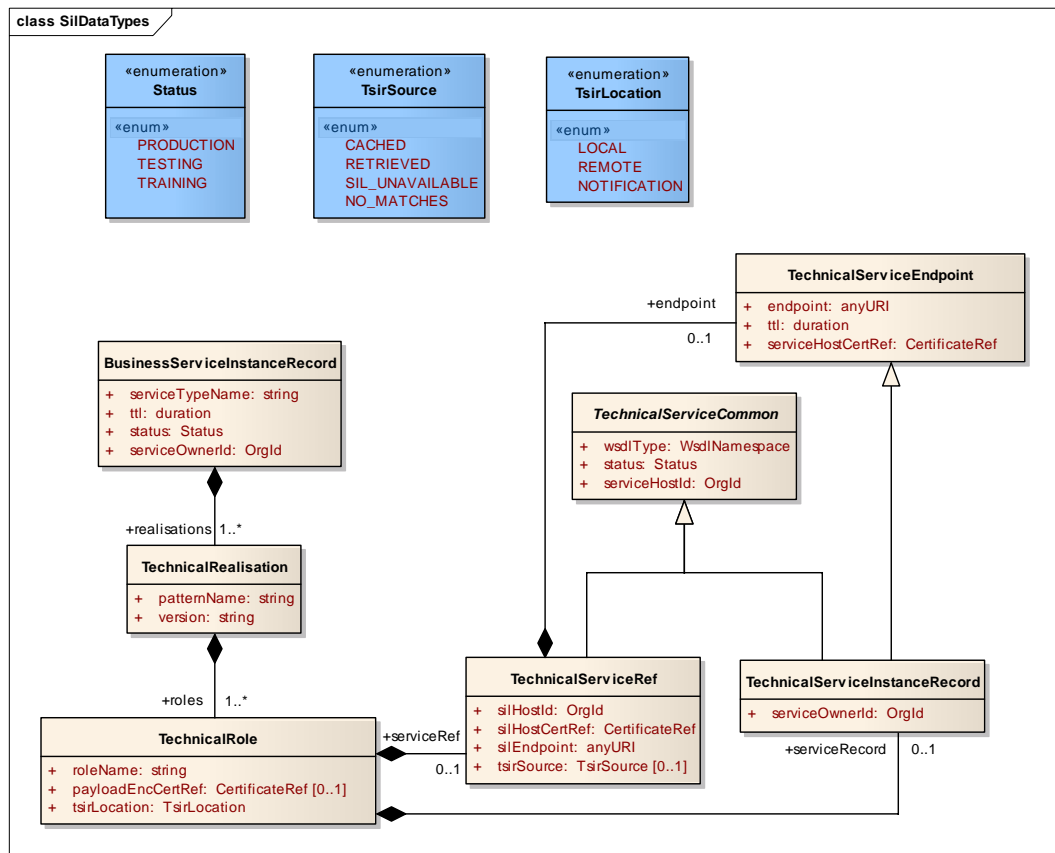


Figure 15: Service Instance Record UML Model

The details of the attributes and references of the classes above is supplied in the following subsections. The "M", "O", "C" tags supplied with each attribute or reference indicate whether it is Mandatory, Optional or Conditional, respectively.

4.1 Class: *QualifiedId* (from *SecureMessaging*)

The *QualifiedId* datatype is used to identify arbitrary entities within a context.

Attribute or Reference Name	Attribute or Reference Type	M O C	Description
qualifier	<<string>> Qualifier	M	The string name of the context in which the <i>id</i> value is relevant.
id	<<string>> Identifier	M	The identifier string of an entity within the context of the <i>qualifier</i> value.

4.2 Class: *OrgId* (from *SecureMessaging*)

The *OrgId* datatype is a subtype of the *QualifiedId* datatype used to identify Organisations. It has no additional attributes, but its *id* attribute is expected to identify an organisation.

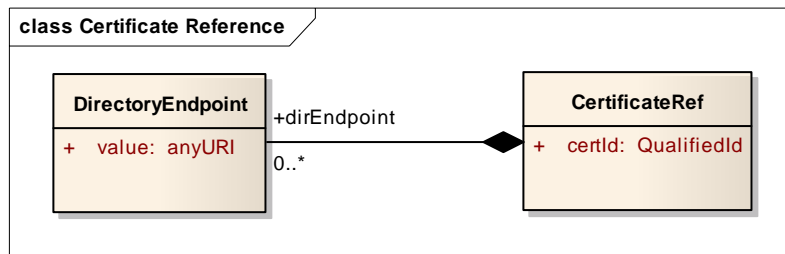


Figure 16: Certificate Reference

4.3 Class: *DirectoryEndpoint* (from *NASH*)

The *DirectoryEndpoint* datatype is used to locate a network endpoint for access to a Certificate Directory.

Attribute or Reference Name	Attribute or Reference Type	M O C	Description
value	anyURI	M	The URI of the endpoint, which includes the protocol specification and internet address.

4.4 Class: *CertificateRef* (from *NASH*)

The *CertificateRef* datatype is used to provide all the information required to lookup a PKI certificate from a Certificate Directory.

Attribute or Reference Name	Attribute or Reference Type	M O C	Description
certId	QualifiedId	M	The qualified string name of the certificate. The qualifier provides information on the attribute in the directory to match when looking up the certificate at the Directory specified by the <i>dirEndpoint</i>

			field.
dirEndpoint	DirectoryEndpoint	0..*	The location(s) at which to look up the <i>certId</i> . If there is no value present, it is presumed that an <i>a priori</i> Directory endpoint will be used.

4.5 Enum: *Status*

This enumeration lists the possible deployment statuses for services.

Tag	Description
PRODUCTION	The service is deployed in production mode.
TESTING	The service is deployed for testing purposes.
TRAINING	The service is deployed for training purposes.

These status labels are aligned with HL7 Version 2.4 standards according to Australian standards [AS4700-2005].

4.6 Enum: *TsirSource*

This enumeration lists the possible sources for endpoint information attached to *TechnicalServiceRef* objects.

Tag	Description
CACHED	The endpoint information is from a TSIR in a different SIL but was returned from a cached value, and not looked up again.
RETRIEVED	The endpoint information has been retrieved from a remote SIL by a lookup that was just made.
SIL_UNAVAILABLE	There is no endpoint information available from the cache and the SIL referred to by a TechnicalServiceRef is unavailable to lookup TSIRs.
NO_MATCHES	There is no endpoint information available from the cache and the SIL referred to by a TechnicalServiceRef returns no results which match the required criteria.

4.7 Class: *BusinessServiceInstanceRecord*

The *BusinessServiceInstanceRecord* is the data structure that captures the information required to identify a service instance. A Business Service **MUST** always be advertised into the SIL for its service owner.

Attribute or Reference Name	Attribute or Reference Type	M O C	Description
serviceTypeName	string	M	The name of the business service, as specified in the NEHTA (or other) specification.
ttl	duration	M	Time To Live. The period for which clients of the SIL may cache a returned BSIR.
status	Status	M	The production status of the business service.
serviceOwner	OrgId	M	The identifier of the organisation that is the logical owner of the service. In many services this corresponds to the ultimate recipient of some communication.
realisations	TechnicalRealisation	M 1..*	A reference to one or more alternative technical realisations of the business service.

Key: serviceTypeName, status, serviceOwner

4.8 Class: *TechnicalRealisation*

The *TechnicalRealisation* class describes a possible configuration of web services that allows the business service to be realised in a certain pattern of endpoint interactions. The patterns are described in the NEHTA *Concepts and Patterns for Implementing Services* [CPIS2008].

Attribute or Reference Name	Attribute or Reference Type	M O C	Description
patternName	string	M	See Section 4.8.1 for details on how this field is used.
version	string	M	The Version of the specification for the service endpoints referred to by this <i>TechnicalRealisation</i> . The string MUST be formatted thus: <code>v{Major}.{Minor}[.{Rev}]</code>

roles	TechnicalRole	M 1..*	References to one or more <i>TechnicalRole</i> instances, which when used according to the pattern, will realise the containing business service.
-------	---------------	-----------	---

Key: <owning BSIR>, patternName, version

4.8.1 Values for *patternName*

The *Interaction Patterns* specified in the Concepts and Pattern for Implementing Services [CPIS2008] only name the roles involved in the pattern, and do not take account of the entities which might fill these roles. Therefore we need a number of variations on the set of patterns so that a BSIR can correctly identify which parties are responsible for filling these roles. For example, the “Store with Notification” pattern does not specify whether the store is chosen by the sender, or by the receiver of the message.

The following strings may be used in the *patternName* field of a *TechnicalRealisation* to indicate the use of particular Interaction Patterns, and indicate in some cases who fills the roles in the interaction.

These interaction patterns assume that a party A is the initiator of communications, and that party B is a recipient/responder to the communications. i.e. the *TechnicalRealisation* is part of a BSIR that is owned by party B.

Future extensions to these patterns may be published in updated versions of *Concepts and Pattern for Implementing Services* [CPIS2008].

<i>patternName</i> string value	Interaction Pattern from CPIS	<i>TechnicalServiceRef</i> roles filled by
“perform”	<i>Perform</i>	<ul style="list-style-type: none"> “perform”, the endpoint for the interactive service offered by B.
“put”	<i>Put</i>	<ul style="list-style-type: none"> “put”, the endpoint for asynchronously accepting
n/a	<i>Get</i>	<ul style="list-style-type: none"> n/a – a TSIR with the appropriate WSDL type will be available at the initiator’s SIL. Not relevant content for a BSIR Technical Realisation.
“get-with-notif”	<i>Get with notifications</i>	<ul style="list-style-type: none"> “notif” – the endpoint for the notification store that party B polls to be informed of new content of this business service type. Optionally a “get” Role may be

		<p>included, but it must have a <i>tsirLocation</i> value of "NOTIFICATION" and no <i>serviceRef</i> or <i>serviceRecord</i> values. The party owning the get endpoint (A) will be identified in the notification received at the "notif" endpoint. The get endpoint will be available via lookup at A's SIL as a TSIR with the appropriate WSDL type.</p>
"store"	<i>Store</i>	<ul style="list-style-type: none"> "store" – the endpoint for an encrypted store which party B polls for all new content of this business service type.
"recipient-store-with-notif"	<i>Store with notifications, in which the store is chosen by the recipient.</i>	<ul style="list-style-type: none"> "store" – the endpoint for an encrypted store used by party B for all content of this business service type. The <i>TechnicalRole</i> MUST contain both a <i>TechnicalServiceRef</i> in the <i>serviceRef</i> field, and a certificate reference in the <i>payloadEncCertRef</i> field. The <i>tsirLocation</i> field MUST be set to "REMOTE". "notif" – the endpoint for the notification store that party B polls to be informed of new content of this business service type.
"initiator-store-with-notif"	<i>Store with notifications, in which the store is chosen by the initiator.</i>	<ul style="list-style-type: none"> "store" – this <i>TechnicalRole</i> MUST have its <i>tsirLocation</i> value set to "NOTIFICATION", and its <i>payloadEncCertRef</i>

		<p>field MUST contain the reference to the certificate which must be used to encrypt the payload. The endpoint for an encrypted store used by party A to actually transmit the payload will be looked up with information that is contained in the notification delivered to the "notif" endpoint.</p> <ul style="list-style-type: none"> • "notif" – the endpoint for the notification store that party B polls to be informed of new content of this business service type.
--	--	--

4.9 Class: *TechnicalRole*

This abstract group of fields is used in other classes which need to identify the WSDL type and status of a web service.

Attribute or Reference Name	Attribute or Reference Type	M O D	Description
roleName	string	M	The name of a role in the technical realisation for which the owner of the business service is responsible.
payloadEncCertRef	CertificateRef	O	The reference of the certificate which the owner of the business service wishes the initiator of the communications to use to encrypt the payload that will be placed in a store.
tsirLocation	TsirLocation	M	<p>If the value is set to "LOCAL" then the <i>serviceRecord</i> field must be populated.</p> <p>If the value set to "REMOTE" then the <i>serviceRef</i> field MUST be populated.</p> <p>If the value is set to "NOTIFICATION" then both <i>serviceRecord</i> and</p>

			<i>serviceRef</i> fields MUST be empty.
serviceRef	TechnicalServiceRef	C	A <i>TechnicalServiceRef</i> which identifies one or more TSIRs via their type, status and containing SIL.
serviceRecord	TechnicalServiceInstanceRecord	O	A TSIR which describes an endpoint which may be used to fill this role.

Key: <owning TechnicalRealisation>, roleName

4.10 Abstract Class: *TechnicalServiceCommon*

This abstract group of fields is used in other classes which need to identify the WSDL type and status of a web service.

Attribute or Reference Name	Attribute or Reference Type	M O D	Description
wsdIType	WsdINamespace	M	The identifier for a WSDL interface type.
status	Status	M	The production status of the technical service. If this field group appears in the context of a BSIR, then the status of the BSIR MUST be the same as the status of the technical service that this field describes.
serviceHostId	OrgId	M	The identifier of the organisation which hosts a technical service (may be the same as the identity of the service owner if an organisation hosts its own services, or may be different if services are outsourced).

4.11 Class: *TechnicalServiceEndpoint*

The fields of this class are always used in the context of a *TechnicalServiceRef* or *TechnicalServiceInstanceRecord*, which give the WSDL type, status and service host identity of the web service.

Attribute or Reference Name	Attribute or Reference Type	M O D	Description
endpoint	anyURI	M	The location at which the service can be invoked.
ttl	duration	M	The period for which this

			endpoint can be cached.
serviceHostCertRef	CertificateRef	M	A reference to a certificate that is issued to the service host organisation, which is being used to secure communication to this endpoint.

4.12 Class: *TechnicalServiceRef*

The *TechnicalServiceRef* contains WSDL type, status information and the Id of the service host for a web service, and a reference to the SIL of the service host, and identity and certificate information for the host of this SIL. This information can be thought of as a stored query, which can be used to look up appropriate *TechnicalServiceInstanceRecords* at the nominated SIL.

In addition when returned as part of BSIR resulting from a lookup operation, its *endpoint* field can contain the endpoint information that results from the SIL retrieving technical endpoint information on behalf of the lookup client. The *tsirSource* field indicates whether the *TechnicalServiceEndpoint* present in the *endpoint* field is "CACHED" by this SIL, or "RETRIEVED" by remote lookup on the SIL nominated by the *silEndpoint* field. If the *tsirSource* indicates "SIL_UINAVAILABLE", then the *endpoint* field MUST be empty. In the "SIL_UINAVAILABLE" case, the lookup client must make a second invocation on the SIL nominated by the *TechnicalServiceRef* to retrieve a TSIR directly.

When publishing a BSIR, the *TechnicalServiceRefs* contained by the BSIR's *TechnicalRealisations* refer to TSIRs that are published by a separate invocation on another SIL, either before or after the BSIR is published.

Attribute or Reference Name	Attribute or Reference Type	M O C	Description
WsdType <i>(from TechnicalServiceCommon)</i>	WsdInNamespace	M	The namespace of the WSDL specification that the technical service referred to implements.
status <i>(from TechnicalServiceCommon)</i>	Status	M	The status of the technical service referred to.
serviceHostId <i>(from TechnicalServiceCommon)</i>	OrgId	M	The organisational Id of the host of the technical service being referred to.
silHostId	OrgId	M	The identifier of the organisation which hosts the SIL referenced by <i>silRef</i> .
silHostCertRef	CertificateRef	M	The reference to the certificate that MUST be used for secure communication to the

			<i>silEndpoint</i> .
silEndpoint	anyURI	M	The location of the SIL in which the Technical Service Instance Record for this role is located.
tsirSource	TsirSource	O	MUST not be present in a BSIR to be published. When returned as the result of a lookup it indicates the source of the <i>endpoint</i> which is attached, or the reason for no endpoint being present.
endpoint	TechnicalServiceEndpoint	O	Endpoint information about the web service instance that fulfils the lookup criteria given in all of the other fields of this <i>TechnicalServiceRef</i> . When publishing a BSIR containing a <i>TechnicalServiceRef</i> this field MUST be empty. When returned as a result of a lookup of a business service, this field contains endpoint information looked up on behalf of the caller by the SIL, or if the referenced SIL is unavailable, then it will be empty.

Key: <owning TechnicalRole>

4.13 Class: *TechnicalServiceInstanceRecord*

The *TechnicalServiceInstanceRecord* class holds all information about a technical service instance relevant to a service client wishing to invoke a web service.

Attribute or Reference Name	Attribute or Reference Type	M O C	Description
wsdlType (from <i>TechnicalServiceCommon</i>)	Wsdlnamespace	M	The namespace of the WSDL specification that the service implements.

status <i>(from TechnicalServiceCommon)</i>	Status	M	The production status of the technical service.
serviceHostId <i>(from TechnicalServiceCommon)</i>	OrgId	M	The organisational Id of the host of the service whose endpoint is stored in the <i>endpoint</i> field.
endpoint <i>(from TechnicalServiceEndpoint)</i>	anyURI	M	The location at which the service can be invoked.
ttl <i>(from TechnicalServiceEndpoint)</i>	duration	M	Time To Live. The period for which clients of the SIL may cache a returned endpoint.
serviceHostCertRef <i>(from TechnicalServiceEndpoint)</i>	CertificateRef	M	The reference to the certificate of the <i>serviceHostId</i> to be used when communicating with the <i>endpoint</i> .
serviceOwnerId	OrgId	M	The organisational Id of the organisation on whose behalf the service is hosted. (In the case of document transmission services, this Id can be considered equivalent to the ultimate recipient.)

Key: wsdlType, status, serviceHostId, serviceOwnerId

Appendix A: References

- [AS4700-2005] Standards Australia, AS 4700.1, *Implementing of Health Level Seven, Version 2.4, Part 1: Patient Administration*, 2005
- [NIF2006] NEHTA, *Interoperability Framework*, Version 1.0, 2006 04-01.
- [CPIS2008] NEHTA, *Concepts and Patterns for Implementing Services*, Version 2.0, 2008.