



---

## **Endpoint Location Service**

### **Detailed Design**

Version 1.0 – 30/06/2009

---

**National E-Health Transition Authority Ltd**

Level 25

56 Pitt Street

Sydney, NSW, 2000

Australia.

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

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

**Security**

The content of this document is confidential. The information contained herein must only be used for the purpose for which it is supplied and must not be disclosed other than explicitly agreed in writing with NEHTA.

**Copyright © 2009, NEHTA.**

This document contains information which is protected by copyright. All Rights Reserved. No part of this work may be reproduced or used in any form or by any means without the permission of NEHTA. All copies of this document must include the copyright and other information contained on this page.

**Table of contents**

<b>1</b>	<b>Introduction .....</b>	<b>1</b>
1.1	Purpose .....	1
1.2	Acronyms .....	1
1.3	Architectural Overview.....	1
<b>2</b>	<b>Client Tier .....</b>	<b>4</b>
2.1	Client/Server Interface .....	4
2.2	Authentication .....	4
2.3	Access Control.....	5
<b>3</b>	<b>Web Tier .....</b>	<b>6</b>
3.1	Authentication .....	6
3.2	Access Control.....	6
3.2.1	Privileges .....	6
3.3	Web Services .....	7
3.3.1	Web Service Addressing .....	7
3.3.2	Logging and Auditing .....	7
3.3.3	Web Service Methods.....	7
3.4	Flex Web Facade.....	9
3.4.1	Remote Methods .....	10
<b>4</b>	<b>Business Tier .....</b>	<b>14</b>
4.1	Business Facade .....	14
4.1.1	Business Methods.....	14
4.2	Transaction Management .....	19
<b>5</b>	<b>Data Tier.....</b>	<b>20</b>
5.1	Database Access.....	20
5.2	Transaction Management .....	20
5.3	Auditing .....	20
5.4	Data Facade.....	20
5.4.1	Data Methods .....	20
5.5	Data Transfer Objects (DTOs).....	25
5.6	Search Criteria Objects .....	26
5.7	Data Access Objects (DAOs).....	26
<b>6</b>	<b>Database .....</b>	<b>28</b>
6.1	Security Tables.....	28
6.1.1	Organisation.....	28
6.1.2	Privilege.....	28
6.1.3	OrganisationPrivilege .....	28
6.2	ELS Reference Tables .....	29
6.2.1	RefServiceCategory .....	29
6.2.2	RefInteraction.....	30
6.2.3	RefOrganisationId .....	30
6.2.4	RefCertificateUseQualifier .....	31
6.2.5	RefCertificateTypeQualifier .....	31
6.3	ELS Data Tables .....	31
6.3.1	Interaction .....	31
6.3.2	CertificateRef.....	32
6.4	ELS Audit Tables.....	33
6.4.1	InteractionAudit .....	33
6.4.2	CertificateRefAudit.....	33
6.5	Metadata Tables .....	34
6.5.1	Sequence.....	34
<b>7</b>	<b>References.....</b>	<b>35</b>

# Document Information

## Change history

Version	Date	Comments
1.0	11/02/2009	Renamed from SIL to ELS. Changes to ELS schema.
1.0	30/06/2009	Minor revisions

This page has been left blank intentionally.



# 1 Introduction

## 1.1 Purpose

The purpose of this document is to define the detailed design for the reference implementation of the ELS server.

The reference implementation of the ELS server is intended to verify that standards published by NEHTA, particularly the *Web Services Profile* [WSP2008], can be implemented using existing development tools and platforms.

It is also intended that eventually external vendors will interface with the reference platform as an integration exercise.

It is anticipated that the reference implementation of the ELS server will be initially developed in Java and be capable of running on diverse operating systems.

## 1.2 Acronyms

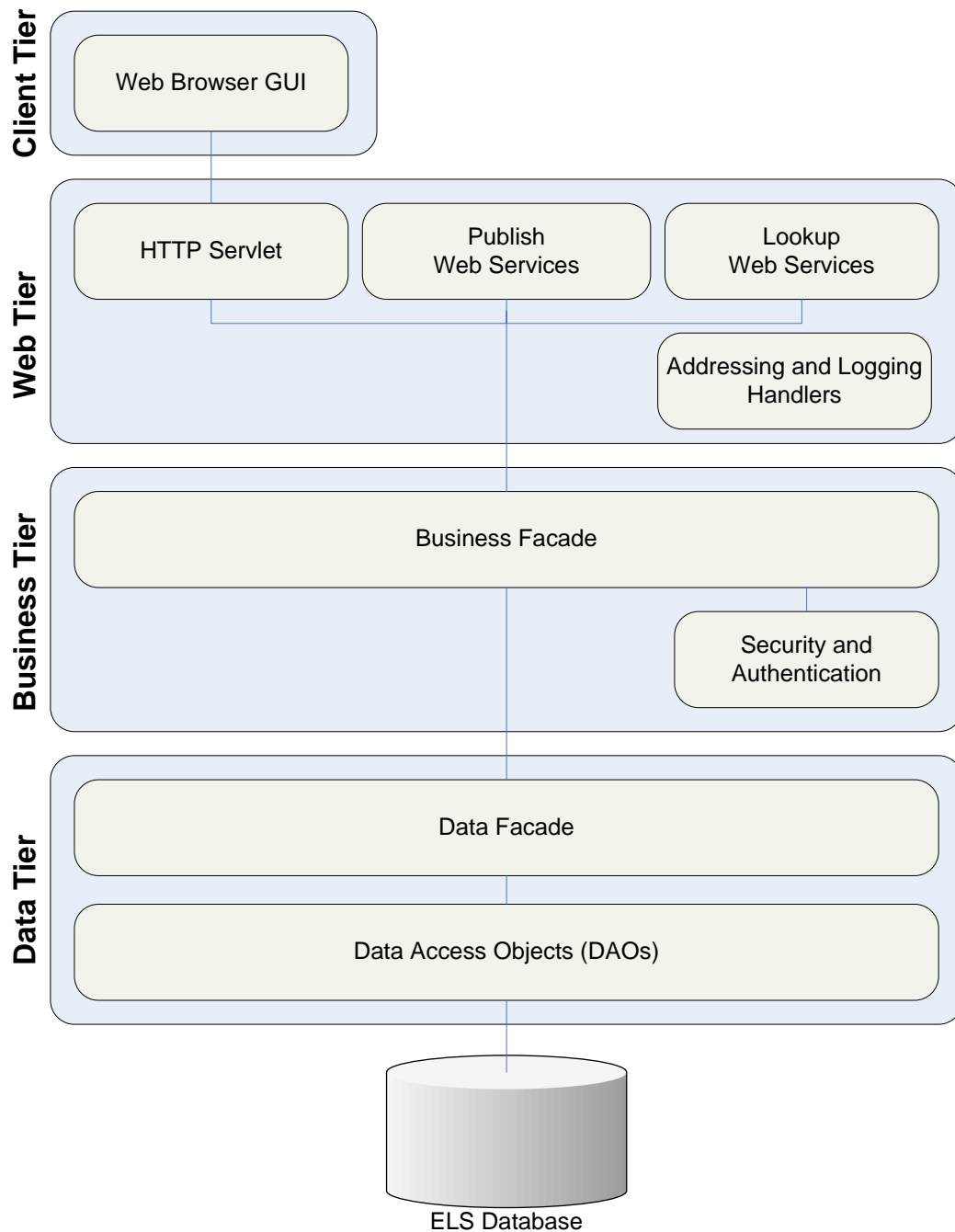
ELS	Endpoint Location Service
WSDL	Web Service Definition Language
XSD	XML Schema Definition

## 1.3 Architectural Overview

The functionality of the ELS server is exposed via Web services. The Web services MUST comply with the NEHTA standard: *Web Services Profile* [WSP2008].

A Web browser client for administering the ELS server is also provided, using an Adobe Flex client application which communicates with the ELS server using mutually-authenticated TLS over HTTP.

The requirements for the reference implementation of the ELS server are defined in the NEHTA document: *Endpoint Location Service – Architecture* [ELSA2009].



**Figure 1: Tiered Design**

The ELS Server consists of four tiers:

#### **Client Tier**

The Client Tier provides a graphical user interface (GUI), hosted within a Web-browser and available over the internet, for administering the contents of the ELS server.

#### **Web Tier**

The Web Tier is the external facing layer of the ELS server which exposes the functionality of the ELS server to other applications. The Web Tier provides two main interfaces:

- Web Services (lookup and publish);
- HTTP Servlet (for interface with the Client Tier).

**Business Tier**

The Business Tier provides a set of business-level methods that are used by all Web Tier components to access the system functionality. It defines a set of business level units of work and applies policies for transactions and security.

**Data Tier**

The Data Tier manages the persistence of ELS data to a datastore (database).

## 2 Client Tier

The Client Tier provides a Web-browser hosted graphical user interface (GUI) for administering users and interactions maintained by the ELS server.

The GUI is built using the Adobe Flex platform and deployed in Shockwave Flash (SWF) format. The GUI requires the Adobe Flash (version 9 or later) plugin to be installed in the client's Web browser.

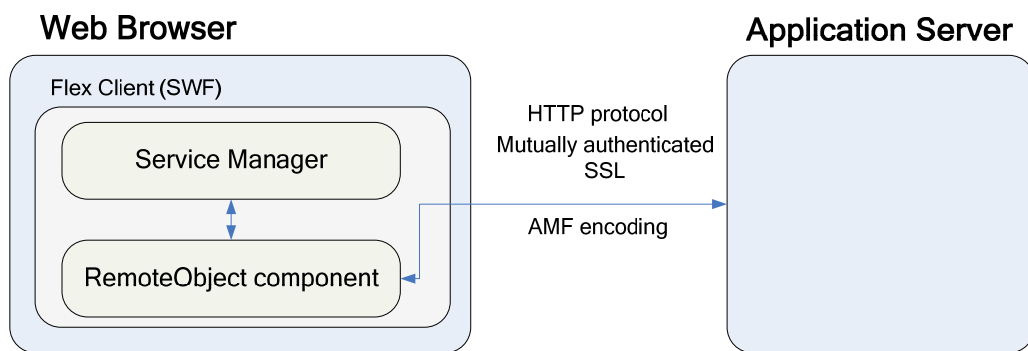
### 2.1 Client/Server Interface

The communication between the GUI client and the ELS server is performed using mutually authenticated SSL over HTTP.

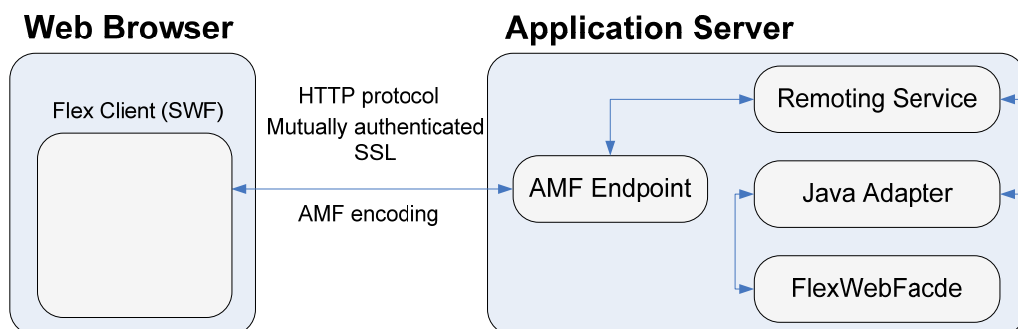
The Flex remoting server – part of the Adobe BlazeDS framework – is used to provide the client/server interface. Remoting allows a Flex application to directly invoke methods of Java objects deployed in the application server.

BlazeDS uses Action Message Format (AMF) encoding over HTTP for client/server communication. This is a high-performance binary format (more compact and efficient than SOAP).

On the client, the `ServiceManager` class is used to invoke remote methods on the server.



On the server, the `FlexWebFacade` class (see section 3.4) is used to provide the methods which may be invoked from the client.



XML configuration files – `blazeds/flex/services-config.xml` and `blazeds/flex/remoting-config.xml` – are used to configure the AMF channel and the remote services respectively.

### 2.2 Authentication

The identity of the user is determined from the organisation that owns the client certificate used in the mutually authenticated SSL connection with the server.

The only credential required is the client certificate, meaning that anybody with the appropriate certificate installed in their Web browser is identified as a valid user for that organisation. This authentication model is identical to that used for the Web service interface.

## **2.3 Access Control**

Once authenticated, the client user (source organisation) may only be able to access a subset of the data, or only perform specific operations. This access control is provided by using privileges.

Refer to section 3.2 for a detailed description of the use of privileges for access control.

## 3 Web Tier

The Web tier is the external facing tier of the ELS server which exposes the functionality of the ELS server to other applications.

The Web tier is a thin layer containing no business logic. The responsibilities of the Web tier include:

- Authenticating the source of the Web service operation;
- Determining the security context of the source, including any roles, privileges and filters which may affect which operations the source may perform and to which data the source may access;
- Calling the appropriate function in the business layer to perform the operation, passing the security context for access control;
- Returning any results, or error conditions, to the source.

### 3.1 Authentication

All Web requests are signed using the private key of the source organisation and encrypted using the public key of the target organisation. The public certificate corresponding to the signing key will be present in every Web request. The signing certificate will be used to identify the source organisation of the Web service request.

The signing certificate will be authenticated with the appropriate certificate authority.

No other credentials (e.g. username/password or individual certificate) are used for authentication.

### 3.2 Access Control

Once authenticated, the source organisation may only be able to access a subset of the data, or only perform specific operations. This access control is provided by using privileges.

#### 3.2.1 Privileges

The following privileges are predefined:

Privilege	Description
SERVICE_TARGET	A privilege assigned to organisations to indicate that they may be used as the 'target' of an interaction.
SERVICE_PROVIDER	A privilege assigned to organisations to indicate that they may be used as the provider of services for other organisations. Note that a target is implicitly able to use itself as the service provider.
SYSTEM_ADMINISTRATOR	A privilege assigned to organisations to indicate that they may perform administrative functions. The administrative functions (e.g. registering organisations) are only available via the Web browser client interface.

Every authenticated organisation will implicitly have the privilege to lookup ELS records.

The registration of Healthcare Provider Organisations is stored in the Organisation table. The assignment of privileges to Healthcare Provider Organisations is stored in the OrganisationPrivilege table.

As well as privileges directly assigned to organisations, a mechanism is also required to support delegating of service provision to other organisations. The OrganisationPrivilege table is used to store information on the delegation of privileges.

*Example 1:* Organisation GP1 provides their own services.

The following privileges are required:

- GP1 has SERVICE\_TARGET privilege

Every organisation with the SERVICE\_TARGET privilege implicitly has the SERVICE\_PROVIDER privilege for their own services.

*Example 2:* Organisation GP2 uses SP1 to provider their services.

The following privileges are required:

- GP2 has SERVICE\_TARGET privilege
- SP1 has SERVICE\_PROVIDER privilege

The following delegated privileges are required:

- GP2 delegates SERVICE\_PROVIDER privilege to SP1

### 3.3 Web Services

The external interface 'contract' is defined by WSDL and XSD documents provided in the *Endpoint Location Service - Architecture* [ELSA2009].

The WSDL documents comply with the NEHTA requirements for secure messaging, including the technical guidelines for Web service definition, security and addressing.

The interfaces for the Web tier will be generated using an appropriate tool from the WSDL and XSD documents provided.

#### 3.3.1 Web Service Addressing

Verification of the addressing header elements of SOAP messages is handled using the `AddressingHandler` SOAP handler. SOAP handlers are defined in JAX-WS bindings specified when the Java classes are generated from the WSDL (via `wsimport`).

The `AddressingHandler` is responsible for implementing the addressing policy [WSP2008].

#### 3.3.2 Logging and Auditing

Logging and Auditing is performed using the `LoggingHandler` SOAP handler. SOAP handlers are defined in JAX-WS bindings specified when the Java classes are generated from the WSDL (via `wsimport`).

The server uses the standard Java Logger and will log the details of inbound/outbound/fault SOAP messages when the log level of the server is set to INFO or finer.

#### 3.3.3 Web Service Methods

3.3.3.1	listInteractions
Parameters	<p>String target Required. The qualified identifier of the service target.</p> <p>List&lt;String&gt; serviceCategories A list of one or more service categories.</p> <p>List&lt;String&gt; interactions Optional. A list of zero or more interaction names.</p>
Returns	<p>List&lt;Interaction&gt;</p> <p>A list of zero or more Interaction records that match the search criteria.</p>
Throws	<p>LookupErrorMsg If the <code>target</code> is not a registered organisation for this ELS.</p> <p>StandardErrorMsg On any other error.</p>
Description	<p>This operation is the core of the specification, realising the primary purpose of ELS.</p> <p>The following notes apply:</p> <ol style="list-style-type: none"> <li>1. If no <code>interactions</code> are provided as parameters, then all interactions will be returned for the input HPIO and service categories. It will then be up to client to choose which interaction they wish to use.</li> <li>2. It will be possible for the returned set to contain more than one interaction of the same kind (e.g. <i>deliver</i>) for the same service category. In this case the roles would refer to different service providers and endpoint addresses. For example, two different service providers may host services on behalf of the same healthcare provider for different document sources. If both sources support the <i>retrieve</i> interaction, the target would have two different <i>notify and retrieve</i> interactions. Alternatively, a target may be moving toward hosting of its own <i>deliver</i> service that is presently hosted by an agent. If a transition period was desired to switch to the new service, two <i>deliver</i> interactions could temporarily co-exist. When such situations arise it is up to the client to choose which interaction to use.</li> <li>3. It is not an error condition if the returned set is empty. It simply means no interactions matched the input.</li> </ol> <p>Returned interactions can be reused. There should be no need to contact the ELS again for the same lookup.</p>

3.3.3.2	validateInteraction
Parameters	<p>Interaction interaction Required. The interaction record to be validated.</p>
Returns	<p>boolean true if the interaction is valid, false if invalid.</p>
Throws	<p>LookupErrorMsg</p>

	<p>If the <code>target</code> within the interaction is not a registered organisation for this ELS.</p> <p><code>StandardErrorMsg</code></p> <p>On any other error.</p>
Description	<p>This operation should be used infrequently. Its purpose is to provide confirmation that a remote reference to an interaction has expired. It take a single <code>Interaction</code> as input, returning <i>true</i> if the interaction is still current, or <i>false</i> if the interaction has been removed.</p> <p>If <i>false</i> is returned, the client should call <code>listInteractions</code> to obtain an up-to-date interaction.</p>

### 3.3.3.3 addInteraction

Parameters	<p><code>Interaction interaction</code></p> <p>Required. The interaction record to be added.</p>
Returns	<p><code>PublishReturnCode</code></p> <p>Returns <i>ok</i> if the interaction was added, else <i>duplicate</i> if an identical interaction was already present.</p>
Throws	<p><code>PublishErrorMsg</code></p> <p>If the <code>target</code> within the interaction is not a registered organisation for this ELS.</p> <p><code>StandardErrorMsg</code></p> <p>On any other error.</p>
Description	Add an interaction to the ELS.

### 3.3.3.4 removeInteraction

Parameters	<p><code>Interaction interaction</code></p> <p>Required. The interaction record to be removed.</p>
Returns	<p><code>PublishReturnCode</code></p> <p>Returns <i>ok</i> if the interaction was removed, else <i>notFound</i> if the interaction was not present in the ELS.</p>
Throws	<p><code>PublishErrorMsg</code></p> <p>If the <code>target</code> within the interaction is not a registered organisation for this ELS.</p> <p><code>StandardErrorMsg</code></p> <p>On any other error.</p>
Description	Remove an interaction from the ELS.

## 3.4 Flex Web Facade

The interface to the Web Tier from the Flex client is via the `FlexWebFacade` class. An instance of the `FlexWebFacade` is created for each HTTP session.

### 3.4.1 Remote Methods

The `FlexWebFacade` provides the following methods which may be remotely called by the Flex client:

#### 3.4.1.1 `getUserContext`

Parameters	(none)
Returns	<p><code>UserContextDTO</code></p> <p>The context details of the user (organisation) that has established a mutually authenticated SSL connection with the server.</p>
Throws	<p><code>MessageException</code></p> <p>On any error.</p>
Description	<p>Return the context details of the user (organisation) that has established a mutually authenticated SSL connection with the server. These details will include:</p> <ul style="list-style-type: none"> <li>• Organisation details;</li> <li>• Privileges of the organisation;</li> <li>• Client certificate details.</li> </ul>

#### 3.4.1.2 `addInteraction`

Parameters	<p><code>InteractionDTO interaction</code></p> <p>The interaction record to be inserted.</p>
Returns	(none)
Throws	<p><code>MessageException</code></p> <p>On any error.</p>
Description	<p>Add a new interaction.</p> <p>Refer to the <code>BusinessFacade</code> <a href="#">addInteraction</a> method for a complete description.</p>

#### 3.4.1.3 `updateInteraction`

Parameters	<p><code>InteractionDTO interaction</code></p> <p>The interaction record to be updated.</p>
Returns	(none)
Throws	<p><code>MessageException</code></p> <p>On any error.</p>
Description	<p>Update the details of an existing interaction record.</p> <p>Refer to the <code>BusinessFacade</code> <a href="#">updateInteraction</a> method for a complete description.</p>

#### 3.4.1.4 `removeInteraction`

Parameters	<code>Integer interactionId</code>
------------	------------------------------------

	The internal identifier of the interaction to be deleted.
Returns	(none)
Throws	MessageException On any error.
Description	Delete an interaction record from the ELS database. Refer to the <code>BusinessFacade</code> <code>removeInteraction</code> method for a complete description.

#### 3.4.1.5 findInteractions

Parameters	InteractionSearchCriteria searchCriteria  The criteria used to constrain the matching interactions. The search criteria must include <code>serviceTarget</code> to match interactions belonging to a particular organisation.
Returns	List<InteractionDTO>  A list of zero or more matching interaction records.
Throws	MessageException On any error.
Description	Search in the ELS database for interaction records that match the search criteria.  Returns a list of zero or more matching interaction records. Refer to the <code>BusinessFacade</code> <code>findInteractions</code> method for a complete description.

#### 3.4.1.6 addOrganisation

Parameters	OrganisationDTO organisation  The organisation record to be inserted.
Returns	(none)
Throws	MessageException On any error.
Description	Register a new organisation. Refer to the <code>BusinessFacade</code> <code>addOrganisation</code> method for a complete description.

#### 3.4.1.7 updateOrganisation

Parameters	OrganisationDTO organisation  The organisation record to be updated.
Returns	(none)
Throws	MessageException On any error.

Description	Update the details of a registered organisation. Refer to the <code>BusinessFacade</code> <code>updateOrganisation</code> method for a complete description.
-------------	--

#### 3.4.1.8 `removeOrganisation`

Parameters	<code>QualifiedIdDTO organisationId</code> The qualified identifier of the organisation to be deleted.
Returns	(none)
Throws	<code>MessageException</code> On any error.
Description	Delete a registered organisation. Refer to the <code>BusinessFacade</code> <code>removeOrganisation</code> method for a complete description.

#### 3.4.1.9 `findOrganisations`

Parameters	<code>OrganisationSearchCriteria searchCriteria</code> The criteria used to constrain the matching organisations.
Returns	<code>List&lt;OrganisationDTO&gt;</code> A list of zero or more matching organisation records.  Only the BRIEF view of matching organisation is returned.
Throws	<code>MessageException</code> On any error.
Description	Search for organisation records that match the search criteria. Returns a list of zero or more matching organisation records.

#### 3.4.1.10 `getOrganisationQualifiers`

Parameters	(none)
Returns	<code>List&lt;ReferenceListItemDTO&gt;</code> The list of allowed organisation qualifiers.
Throws	<code>MessageException</code> On any error.
Description	Return the list of allowed organisation qualifiers. This information can be used as the data provider for GUI controls (such as populating the items in a combo box).

#### 3.4.1.11 `getServiceCategories`

Parameters	(none)
------------	--------

Returns	List<ServiceCategoryDTO> The list of allowed service categories, which in turn includes the interaction and role names for each category.
Throws	MessageException On any error.
Description	Return the list of allowed service categories. This information can be used as the data provider for GUI controls (such as populating the items in a combo box).

#### 3.4.1.12 getCertificateUseQualifiers

Parameters	(none)
Returns	List<ReferenceListItemDTO> The list of allowed certificate use qualifiers.
Throws	MessageException On any error.
Description	Return the list of allowed certificate use qualifiers. This information can be used as the data provider for GUI controls (such as populating the items in a combo box).

#### 3.4.1.13 getCertificateTypeQualifiers

Parameters	(none)
Returns	List<ReferenceListItemDTO> The list of allowed certificate type qualifiers.
Throws	MessageException On any error.
Description	Return the list of allowed certificate type qualifiers. This information can be used as the data provider for GUI controls (such as populating the items in a combo box).

## 4 Business Tier

The Business Tier provides a set of business-level methods that are used by all Web Tier components to access the system functionality. It defines a set of business level units of work and applies policies for transactions and security, thus codifying the rules for maintaining the integrity and consistency of the system. The granularity of methods is such that presentation applications will usually call exactly one method on this interface for each action taken by a user.

To ensure the consistency and integrity of the system is maintained, Web Tier components are not permitted to directly access the lower-level application components at any time.

The Business Tier should at all times reflect the set of business-level actions of the system. Since businesses change over time, it is expected that the set of methods in this component will be modified as required to reflect those business changes. With this in mind, the Business Tier methods are intended to be lightweight, and it is relatively easy to add new methods to the Business Tier.

### 4.1 Business Facade

The interface to the Business Tier is via the `BusinessFacade` class. The Web Tier must instantiate an instance of the `BusinessFacade`, passing a `SecurityContext` as a construction argument. The `SecurityContext` identifies the user – e.g. the HPI-O – and can be used to apply access control policies.

#### 4.1.1 Business Methods

The `BusinessFacade` has the following methods:

##### 4.1.1.1 addInteraction

Parameters	InteractionDTO interaction The interaction record to be inserted.
Returns	(none)
Throws	<p><code>ElsDataException</code> If the argument was invalid.</p> <p><code>ElsSecurityException</code> If the calling does not have the required privileges to perform the operation.</p> <p><code>ElsUnknownOrganisationException</code> If either the <i>serviceTarget</i> of the interaction or the <i>serviceProvider</i> of any of the interaction roles are not registered organisations.</p> <p><code>ElsDuplicateInteractionException</code> If the interaction was an exact duplicate of an existing interaction.</p> <p><code>ElsSystemException</code> On any other error.</p>

Description	<p>Add a new interaction.</p> <p>The details of the interaction record are validated to ensure they specify a valid interaction. If not, a <code>ElsDataException</code> is thrown.</p> <p>The <i>serviceTarget</i> organisation of the new interaction must be registered in the ELS and have <code>SERVICE_TARGET</code> privilege.</p> <p>If the <i>serviceProvider</i> organisation of any of the roles of the interaction is different to the <i>serviceTarget</i>, then the <i>serviceProvider(s)</i> must be registered in the ELS and have <code>SERVICE_PROVIDER</code> privilege.</p> <p>The caller (organisation) must comply with ONE of the following:</p> <ol style="list-style-type: none"> <li>1. Have <code>SYSTEM_ADMINISTRATOR</code> privilege – system administrators may view/edit any data;</li> <li>2. Be the <i>serviceTarget</i> organisation – an organisation can always view/edit their specific data;</li> <li>3. Be the <i>serviceProvider</i> organisation for all of the interaction roles, and be a delegated service provider of the <i>serviceTarget</i>.</li> </ol> <p>If any of the access control restrictions are not met, a <code>ElsSecurityException</code> is thrown.</p> <p>A check is made whether the new interaction is an exact copy of an existing interaction. In this case a <code>ElsDuplicateInteractionException</code> is thrown.</p> <p>Finally, the interaction is inserted into the ELS database.</p>
-------------	--

#### 4.1.1.2 updateInteraction

Parameters	<p><code>InteractionDTO interaction</code></p> <p>The interaction record to be updated.</p>
Returns	(none)
Throws	<p><code>ElsDataException</code></p> <p>If the argument was invalid.</p> <p><code>ElsSecurityException</code></p> <p>If the calling does not have the required privileges to perform the operation.</p> <p><code>ElsUnknownOrganisationException</code></p> <p>If either the <i>serviceTarget</i> of the interaction or the <i>serviceProvider</i> of any of the interaction roles are not registered organisations.</p> <p><code>ElsUnknownInteractionException</code></p> <p>If the interaction record to be replaced was not found.</p> <p><code>ElsDuplicateInteractionException</code></p> <p>If the interaction was an exact duplicate of an existing interaction.</p> <p><code>ElsSystemException</code></p> <p>On any other error.</p>
Description	<p>Update the details of an existing interaction record.</p> <p>The details of the interaction record are validated to ensure they</p>

specify a valid interaction. If not, a `ElsDataException` is thrown.

The *serviceTarget* organisation of the interaction must be registered in the ELS and have `SERVICE_TARGET` privilege.

If the *serviceProvider* organisation of any of the roles of the interaction is different to the *serviceTarget*, then the *serviceProvider(s)* must be registered in the ELS and have `SERVICE_PROVIDER` privilege.

The caller (organisation) must comply with ONE of the following:

1. Have `SYSTEM_ADMINISTRATOR` privilege – system administrators may view/edit any data;
2. Be the *serviceTarget* organisation – an organisation can always view/edit their specific data;
3. Be the *serviceProvider* organisation for all of the interaction roles, and be a delegated service provider of the *serviceTarget*.

If any of the access control restrictions are not met, a `ElsSecurityException` is thrown.

A check is made to find the interaction record to be replaced. If not found, a `ElsUnknownInteractionException` is thrown.

A check is made whether the interaction is an exact copy of an existing interaction (other than the one being replaced). In this case a `ElsDuplicateInteractionException` is thrown.

Finally, the interaction is updated in the ELS database.

#### 4.1.1.3 removeInteraction

Parameters	<p>Integer interactionId</p> <p>The internal identifier of the interaction to be deleted.</p>
Returns	(none)
Throws	<p><code>ElsDataException</code></p> <p>If the argument was invalid.</p> <p><code>ElsSecurityException</code></p> <p>If the calling does not have the required privileges to perform the operation.</p> <p><code>ElsUnknownInteractionException</code></p> <p>If the interaction was not found.</p> <p><code>ElsSystemException</code></p> <p>On any other error.</p>
Description	<p>Delete an interaction record from the ELS database.</p> <p>The interaction record to be deleted is fetched from the ELS database. If not found, a <code>ElsUnknownInteractionException</code> is thrown.</p> <p>The caller (organisation) must comply with ONE of the following:</p> <ol style="list-style-type: none"> <li>1. Have <code>SYSTEM_ADMINISTRATOR</code> privilege – system administrators may view/edit any data;</li> <li>2. Be the <i>serviceTarget</i> organisation – an organisation can always view/edit their specific data;</li> </ol>

	<p>3. Be the <i>serviceProvider</i> organisation for all of the interaction roles.</p> <p>If any of the access control restrictions are not met, a <code>ElsSecurityException</code> is thrown.</p>
--	---

#### 4.1.1.4 findInteractions

Parameters	<p>InteractionSearchCriteria searchCriteria</p> <p>The criteria used to constrain the matching interactions. The search criteria must include either <i>interactionId</i> to match a single interaction or <i>serviceTarget</i> to match interactions belonging to an organisation.</p> <p>ViewType viewType</p> <p>FULL or BRIEF. Specifies whether the related details of the interaction (roles and certificates) are returned (FULL) or not (BRIEF). Requesting a BRIEF view may improve performance if the related details are not required.</p>
Returns	<p>List&lt;InteractionDTO&gt;</p> <p>A list of zero or more matching interaction records.</p>
Throws	<p>ElsDataException</p> <p>If the argument was invalid.</p> <p>ElsSystemException</p> <p>On any other error.</p>
Description	<p>Search in the ELS database for interaction records that match the search criteria.</p> <p>There are no access controls placed on viewing interaction records.</p> <p>Returns a list of zero or more matching interaction records.</p>

#### 4.1.1.5 addOrganisation

Parameters	<p>OrganisationDTO organisation</p> <p>The details of the new organisation.</p>
Returns	(none)
Throws	<p>ElsDataException</p> <p>If any mandatory details of the record are missing or otherwise invalid.</p> <p>ElsSecurityException</p> <p>If the caller does not have the required privileges to perform the operation.</p> <p>ElsDuplicateOrganisationException</p> <p>If an existing record already exists with identical details.</p> <p>ElsSystemException</p> <p>On any other error.</p>
Description	Insert a new organisation record into the ELS database. Checks

	<p>for exact duplicate prior to insert.</p> <p>The caller (organisation) must have the SYSTEM_ADMINISTRATOR privilege.</p>
4.1.1.6	updateOrganisation
Parameters	<p>OrganisationDTO organisation</p> <p>The organisation record to be updated.</p>
Returns	(none)
Throws	<p>ElsDataException</p> <p>If any mandatory details of the record are missing or otherwise invalid.</p> <p>ElsSecurityException</p> <p>If the caller does not have the required privileges to perform the operation.</p> <p>ElsUnknownOrganisationException</p> <p>If the organisation was not found.</p> <p>ElsSystemException</p> <p>On any other error.</p>
Description	<p>Update the details of an existing organisation record in the ELS database.</p> <p>The caller (organisation) must have the SYSTEM_ADMINISTRATOR privilege.</p>
4.1.1.7	removeOrganisation
Parameters	<p>QualifiedIdDTO organisationId</p> <p>The qualified identifier of the organisation to be deleted.</p>
Returns	(none)
Throws	<p>ElsDataException</p> <p>If the argument was invalid.</p> <p>ElsSecurityException</p> <p>If the caller does not have the required privileges to perform the operation.</p> <p>ElsUnknownOrganisationException</p> <p>If the organisation was not found.</p> <p>ElsSystemException</p> <p>On any other error.</p>
Description	<p>Delete an organisation record from the ELS database.</p> <p>Any interactions of which the organisation is a <i>serviceTarget</i> are also deleted.</p> <p>The organisation cannot be deleted if it is the <i>serviceProvider</i> for other organisations.</p> <p>The caller (organisation) must have the SYSTEM_ADMINISTRATOR privilege.</p>

## 4.1.1.8 findOrganisations

Parameters	<p>OrganisationSearchCriteria searchCriteria</p> <p>The criteria used to constrain the matching organisations.</p> <p>ViewType viewType</p> <p>FULL or BRIEF. Specifies whether the related details of the organisation (such as privileges) are returned (FULL) or not (BRIEF). Requesting a BRIEF view may improve performance if the related details are not required.</p>
Returns	<p>List&lt;OrganisationDTO&gt;</p> <p>A list of zero or more matching organisation records.</p>
Throws	<p>ElsDataException</p> <p>If the arguments were invalid.</p> <p>ElsSystemException</p> <p>On any other error.</p>
Description	<p>Search in the ELS database for organisation records that match the search criteria.</p> <p>No access controls are placed on the viewing of organisation records.</p> <p>A limit may be placed on the number of returned records.</p> <p>Returns a list of zero or more matching organisation records.</p>

## 4.1.1.9 getOrganisation

Parameters	<p>QualifiedIdDTO organisationId</p> <p>The qualified identifier of a particular organisation.</p>
Returns	<p>OrganisationDTO</p> <p>The details of the matching organisation.</p>
Throws	<p>ElsDataException</p> <p>If the arguments were invalid.</p> <p>ElsUnkownOrganisationException</p> <p>If the organisation was not found.</p> <p>ElsSystemException</p> <p>On any other error.</p>
Description	<p>Search in the ELS database for organisation records that match the search criteria.</p> <p>Returns a list of zero or more matching organisation records.</p>

## 4.2 Transaction Management

Each method of the `BusinessFacade` is considered to be a discrete unit of work and is performed within a single transaction. If a call to a business method is successful, any database or state changes are committed before the business method returns. If a call to a business method fails, any business or state changes are rolled back before the business method returns.

## 5 Data Tier

The Data Tier manages the persistence of ELS data to the ELS database as well as in-memory caching of reference data for performance improvement.

### 5.1 Database Access

Java Database Connectivity (JDBC) will be used to access (read/write) the ELS database.

### 5.2 Transaction Management

Database (JDBC) connections will be obtained from the JDBC connection pool managed by the Sun Java Application Server. It is the responsibility of the Business Tier to obtain connections, commit/rollback transactions and finally return connections to the JDBC connection pool.

### 5.3 Auditing

All changes made to the ELS Data tables (Interaction and CertificateRef) are audited, with the audit data stored in the ELS Audit Tables. In addition to the data that was modified, the following information is stored with each audit record:

- `operationType`: The type of data operation that was performed: Insert, Update or Delete;
- `operationDate`: The date/time of the operation;
- `operationUserId`: The qualified identifier of the user (organisation) that performed the operation.

Auditing is performed at the database level using triggers on the ELS Data Tables.

### 5.4 Data Facade

The interface to the Data Tier is via the `DataFacade` class. The Business Tier must instantiate an instance of the `DataFacade`, passing a `database Connection` as construction argument. The `database Connection` is required so that multiple data access operations can be performed in a single transaction. It is assumed that the Business Tier has already applied authentication and access control policy.

#### 5.4.1 Data Methods

The `DataFacade` has the following methods:

##### 5.4.1.1 `addInteraction`

Parameters	InteractionDTO interaction The interaction record to be inserted.
Returns	
Throws	ElsDataException If any mandatory details of the interaction record are missing or otherwise invalid. ElsDuplicateInteractionException

	<p>If an existing interaction record already exists with identical details.</p> <p>ElsSystemException</p> <p>On any other error.</p>
Description	<p>Insert a new interaction record into the ELS database.</p> <p>Performs the following validations prior to insert:</p> <ul style="list-style-type: none"> <li>• All required values are defined;</li> <li>• All string fields do not exceed maximum lengths;</li> <li>• The structure of the interaction conforms to static business rules and also to the configured interactions in the reference data;</li> <li>• All foreign references are valid.</li> </ul> <p>If any of the validations fail, a ElsDataException is thrown.</p> <p>Also, a check is made to see if a duplicate interaction with the same details does not already exist. If so, a ElsDuplicateInteractionException is thrown.</p>

5.4.1.2 removeInteraction

Parameters	<p>Integer interactionId</p> <p>The internal identifier of the interaction to be deleted.</p>
Returns	(none)
Throws	<p>ElsDataException</p> <p>If the argument was invalid.</p> <p>ElsUnknownInteractionException</p> <p>If the interaction was not found.</p> <p>ElsSystemException</p> <p>On any other error.</p>
Description	Delete an interaction record from the ELS database.

5.4.1.3 findInteractions

Parameters	<p>InteractionSearchCriteria searchCriteria</p> <p>The criteria used to constrain the matching interactions. The search criteria must include either interactionId to match a single interaction or serviceTarget to match interactions belonging to an organisation.</p> <p>Boolean lockForUpdate</p> <p>Should update locks be placed on the matching interaction rows in the database? This is useful if the matching interaction records are to be updated or deleted in the same database transaction – it guarantees that another user does not change the interaction records in the interim.</p> <p>ViewType viewType</p> <p>FULL or BRIEF. Specifies whether the related details of the interaction (roles and certificates) are returned</p>
------------	--

	(FULL) or not (BRIEF). Requesting a BRIEF view may improve performance if the related details are not required.
Returns	List<InteractionDTO> A list of zero or more matching interaction records.
Throws	ElsDataException If the argument was invalid. ElsSystemException On any other error.
Description	Search in the ELS database for interaction records that match the search criteria. Returns a list of zero or more matching interaction records.

#### 5.4.1.4 addOrganisation

Parameters	OrganisationDTO organisation The organisation record to be inserted.
Returns	(none)
Throws	ElsDataException If any mandatory details of the record are missing or otherwise invalid. ElsDuplicateOrganisationException If an existing record already exists with identical details. ElsSystemException On any other error.
Description	Insert a new organisation record into the ELS database. Checks for exact duplicate prior to insert.

#### 5.4.1.5 updateOrganisation

Parameters	OrganisationDTO organisation The organisation record to be updated.
Returns	(none)
Throws	ElsDataException If any mandatory details of the record are missing or otherwise invalid. ElsUnknownOrganisationException If the organisation was not found. ElsSystemException On any other error.
Description	Update the details of an existing organisation record in the ELS database.

## 5.4.1.6 removeOrganisation

Parameters	QualifiedIdDTO organisationId The qualified identifier of the organisation to be deleted.
Returns	(none)
Throws	ElsDataException If the argument was invalid. ElsUnknownOrganisationException If the organisation was not found. ElsSystemException On any other error.
Description	Delete an organisation record from the ELS database.

## 5.4.1.7 findOrganisations

Parameters	OrganisationSearchCriteria searchCriteria The criteria used to constrain the matching organisations. The search criteria must include either interactionId to match a single interaction or serviceTarget to match interactions belonging to an organisation. ViewType viewType FULL or BRIEF. Specifies whether the related details of the organisation (such as privileges) are returned (FULL) or not (BRIEF). Requesting a BRIEF view may improve performance if the related details are not required.
Returns	List<OrganisationDTO> A list of zero or more matching organisation records.
Throws	ElsDataException If the arguments were invalid. ElsSystemException On any other error.
Description	Search in the ELS database for organisation records that match the search criteria. Returns a list of zero or more matching organisation records.

## 5.4.1.8 getOrganisation

Parameters	QualifiedIdDTO organisationId The qualified identifier of a particular organisation. boolean required If the organisation is not found, and requires is <i>true</i> , a <i>ElsUnknownOrganisationException</i> is thrown, else null is returned.
Returns	OrganisationDTO The details of the matching organisation.

Throws	<p>ElsDataException If the arguments were invalid.</p> <p>ElsUnkownOrganisationException If the organisation was not found (and required is <i>true</i>).</p> <p>ElsSystemException On any other error.</p>
Description	<p>Return the full details of the requested organisation.</p> <p>May return null if not found and required is <i>false</i>.</p>

#### 5.4.1.9 getServiceCategories

Parameters	(none)
Returns	<p>List&lt;ServiceCategoryDTO&gt;</p> <p>The details of all configured service categories.</p>
Throws	<p>ElsSystemException On any other error.</p>
Description	<p>Return the details of all configured service categories from the ELS database (or an in-memory cache).</p>

#### 5.4.1.10 getServiceCategory

Parameters	<p>String serviceCategory</p> <p>The URL of a service category.</p>
Returns	<p>ServiceCategoryDTO</p> <p>The details of the matching service category.</p>
Throws	<p>ElsDataException If the arguments were invalid or the service category was not found.</p> <p>ElsSystemException On any other error.</p>
Description	<p>Search in the ELS database (or an in-memory cache) for the details of the given service category.</p>

#### 5.4.1.11 getCertificateUseQualifiers

Parameters	(none)
Returns	<p>List&lt;CertificateUseQualifierDTO&gt;</p> <p>The details of all configured certificate use qualifiers.</p>
Throws	<p>ElsSystemException On any other error.</p>
Description	<p>Return the details of all configured certificate use qualifiers from the ELS database (or an in-memory cache).</p>

5.4.1.12 `getCertificateUseQualifier`

Parameters	String <code>useQualifier</code>
Returns	<code>CertificateUseQualifierDTO</code> The details of the requested certificate use qualifier.
Throws	<code>ElsDataException</code> If the arguments were invalid or the record was not found. <code>ElsSystemException</code> On any other error.
Description	Return the details of the requested certificate use qualifier from the ELS database (or an in-memory cache).

5.4.1.13 `getCertificateTypeQualifiers`

Parameters	(none)
Returns	<code>List&lt;CertificateTypeQualifierDTO&gt;</code> The details of all configured certificate type qualifiers.
Throws	<code>ElsSystemException</code> On any other error.
Description	Return the details of all configured certificate type qualifiers from the ELS database (or an in-memory cache).

5.4.1.14 `getCertificateTypeQualifier`

Parameters	String <code>typeQualifier</code>
Returns	<code>CertificateTypeQualifierDTO</code> The details of the requested certificate type qualifier.
Throws	<code>ElsDataException</code> If the arguments were invalid or the record was not found. <code>ElsSystemException</code> On any other error.
Description	Return the details of the requested certificate type qualifier from the ELS database (or an in-memory cache).

## 5.5 Data Transfer Objects (DTOs)

Data Transfer Objects (DTOs) are Java objects used to transfer data between all tiers of the application. The structure of the DTOs very closely resembles the database table structure.

The concept of a DTO is loosely based on the *Java Design Pattern – Transfer Objects [DTO]*.

The Data Transfer Objects do not contain any JDBC code.

The following DTOs are used for ELS reference data:

**RefServiceCategoryDTO**

Data Transfer Object for the [RefServiceCategory](#) table.

**RefInteractionNameDTO**

Data Transfer Object for the [RefInteraction](#) table.

**RefOrganisationIdDTO**

Data Transfer Object for the [RefOrganisationId](#) table.

**RefCertificateUseQualifierDTO**

Data Transfer Object for the [RefCertificateUseQualifier](#) table.

**RefCertificateTypeQualifierDTO**

Data Transfer Object for the [RefCertificateTypeQualifier](#) table.

The following DTOs are used for ELS instance data:

**InteractionDTO**

Data Transfer Object for the [Interaction](#) and related tables.

**CertificateRefDTO**

Data Transfer Object for the [CertificateRef](#) table.

**QualifiedIdDTO**

Data Transfer Object for qualified identifiers. Qualified identifiers are stored in the database in serialized form: '<qualifier>#<id>'.

**OrganisationDTO**

Data Transfer Object for the [Organisation](#) and [OrganisationPrivilege](#) tables.

## 5.6 Search Criteria Objects

Search Criteria Objects are Java objects used to define search criteria which can be used for searching for data.

The Search Criteria objects are mutable. They are constructed using a default constructor and rely on setter methods for initialisation. Search Criteria Objects may be validated before use to ensure that searches are made that will properly utilise indexes and match a limited amount of data.

The Search Criteria objects do not contain any JDBC code.

The following Search Criteria objects are available:

**InteractionSearchCriteria**

Provides search criteria for the [Interaction](#) table.

**OrganisationSearchCriteria**

Provides search criteria for the [Organisation](#) and [OrganisationPrivilege](#) tables.

## 5.7 Data Access Objects (DAOs)

Data Access Objects (DAOs) are Java objects used to access the database. The concept of a DAO is based on the *Core J2EE Design Pattern – Data Access Object [DAO]*.

The Data Access Objects are provided a database connection during construction.

The Data Access Objects interact with DTOs and Search Criteria objects as parameters and results.

The Data Access Objects are used by the `DataFacade` within the Data Tier. They should not be directly used outside the Data Tier.

The following DAOs are available for the ELS reference tables:

**RefServiceCategoryDAO**

Data Access Object for the RefServiceCategory table.

**RefInteractionDAO**

Data Access Object for the RefInteraction table.

**RefOrganisationIdDAO**

Data Access Object for the RefOrganisationId table.

**RefCertificateUseQualifierDAO**

Data Access Object for the RefCertificateUseQualifier table.

**RefCertificateTypeQualifierDAO**

Data Access Object for the RefCertificateTypeQualifier table.

The following DAOs are available for the ELS data tables:

**InteractionDAO**

Data Access Object for the Interaction table.

**CertificateRefDAO**

Data Access Object for the CertificateRef table.

**OrganisationDAO**

Data Access Object for the Organisation and OrganisationPrivilege tables.

## 6 Database

The database design matches the Web service interface as closely as possible to minimise the transformations required by the ELS server.

### 6.1 Security Tables

Security tables are used to store information for the management of access control.

#### 6.1.1 Organisation

The *Organisation* table stores details of Healthcare Provider Organisations that have particular privileges within the ELS.

Column	Type	Description
organisationId	Varchar(100) not null	Primary key. Qualified identifier of a Healthcare Provider Organisation.
organisationName	Varchar(100) not null	Unique name of the Healthcare Provider Organisation.

Constraints

1. Primary Key (organisationId)
2. Unique (organisationName)

#### 6.1.2 Privilege

The *Privilege* table stores the privileges which may be assigned to Healthcare Provider Organisations.

Column	Type	Description
privilegeId	Varchar(50) not null	Primary key. Identifier of the privilege.
privilegeName	Varchar(100) not null	Unique name of the Privilege.

Constraints

1. Primary Key (privilegeId)
2. Unique (privilegeName)

The contents of the *Privilege* table will be configured as part of the installation process using bootstrap data. See section 3.2 for descriptions of the predefined privileges.

#### 6.1.3 OrganisationPrivilege

The *OrganisationPrivilege* table stores the privileges assigned to Healthcare Provider Organisations.

In order for an organisation to delegate another organisation (or organisations) as its service provider, the *OrganisationPrivilege* table has a *delegatedOrganisationId* column which can be used to delegate a privilege to another organisation.

At present, only the SERVICE\_PROVIDER privilege may be delegated by an organisation that already has the SERVICE\_TARGET privilege. I.e. A service target can nominate another organisation to provide a service on its behalf.

Column	Type	Description
organisationId	Varchar(100) not null	Identifier of a Healthcare Provider Organisation. References the <i>Organisation</i> table.
privilegeId	Varchar(50) not null	Identifier of a Privilege. References the <i>Privilege</i> table.
delegatedOrganisationId	Varchar(100) not null	Identifier of a Healthcare Provider Organisation that has been delegated the privilege.  If the privilege is NOT delegated, this will equal the organisationId.

#### Constraints

1. Primary Key (organisationId, privilegeId, delegatedOrganisationId)
2. Foreign Key (organisationId) references Organisation
3. Foreign Key (privilegeId) references Privilege
4. Foreign Key (delegatedOrganisationId) references Organisation

## 6.2 ELS Reference Tables

The ELS reference tables provide reference data that is configured during the installation procedure. The reference data can be used to enforce database integrity (via constraints) and to provide valid options for data entry and maintenance.

### 6.2.1 RefServiceCategory

The *RefServiceCategory* table provides a reference of the allowed service categories.

The contents of the *RefServiceCategory* table will be populated during installation from bootstrap data.

Column	Type	Description
serviceCategory	Varchar(100) not null	Name of the service category. E.g. "http://ns.nehta.gov.au/Pth/RR/SR/Category/Const/SealedReport/1.0".
label	Varchar(30) not null	A short label for the service category. E.g. "Sealed Report".
description	Varchar(*)	A longer description of the purpose of the service category.

#### Constraints

1. Primary Key (serviceCategory)

## 6.2.2 RefInteraction

The *RefInteraction* table provides a reference of the allowed interaction names. An interaction must belong to a single service category, hence the inline reference rather than a separate relationship table.

The contents of the *RefInteraction* table will be populated during installation from bootstrap data.

Column	Type	Description
interactionName	Varchar(100) not null	Name of the service interaction. E.g. "http://ns.nehta.gov.au/Pth/RR/SR/Scenario/Const/Deliver/1.0".
serviceCategory	Varchar(100) not null	References the <i>ServiceCategory</i> table.
serviceInterface	Varchar(100) not null	Name of the service interface. E.g. "http://ns.nehta.gov.au/Pth/RR/SR/Svc/Const/Consumer/1.0".
label	Varchar(30) not null	A short label for the interaction name. E.g. "Deliver".
description	Varchar(*)	A longer description of the purpose of the interaction name.

Constraints

1. Primary Key (interactionName)
2. Foreign key (serviceCategory) references RefServiceCategory

## 6.2.3 RefOrganisationId

The *RefOrganisationId* table provides a reference for the qualifiers and identifiers to be used for Organisations.

The contents of the *RefOrganisationId* table will be populated during installation from bootstrap data.

Column	Type	Description
qualifier	Varchar(100) not null	Name of the organisation qualifier. E.g. <a href="http://ns.nehta.gov.au/Id/Const/UhiHpio/1.0">http://ns.nehta.gov.au/Id/Const/UhiHpio/1.0</a>
idPattern	Varchar(100)	An optional regular expression which can be used to validate an identifier. E.g. 800363[0-9]{9}[0-9A-Z]{1}
label	Varchar(30) not null	A short label for the use qualifier. E.g. "HPI-O".
description	Varchar(*)	A longer description of the qualifier.

Constraints

1. Primary Key (qualifier)

### 6.2.4 RefCertificateUseQualifier

The *RefCertificateUseQualifier* table provides a reference of the allowed use qualifiers for certificate references.

The contents of the *RefCertificateUseQualifier* table will be populated during installation from bootstrap data.

Column	Type	Description
useQualifier	Varchar(100) not null	Name of the use qualifier. E.g. <a href="http://ns.nehta.gov.au/Qcr/Use/Const/Sign/1.0">http://ns.nehta.gov.au/Qcr/Use/Const/Sign/1.0</a>
label	Varchar(30) not null	A short label for the use qualifier. E.g. Signing
description	Varchar(*)	A longer description of the purpose of the use qualifier.

Constraints

1. Primary Key (useQualifier)

### 6.2.5 RefCertificateTypeQualifier

The *RefCertificateTypeQualifier* table provides a reference of the allowed type qualifiers for certificate references.

The contents of the *RefCertificateTypeQualifier* table will be populated during installation from bootstrap data.

Column	Type	Description
typeQualifier	Varchar(100) not null	Name of the type qualifier. E.g. <a href="http://ns.nehta.gov.au/Qcr/Ref/Const/SerialNum/1.0">http://ns.nehta.gov.au/Qcr/Ref/Const/SerialNum/1.0</a>
label	Varchar(30) not null	A short label for the type qualifier. E.g. Serial Number
description	Varchar(*)	A longer description of the purpose of the type qualifier.

Constraints

1. Primary Key (typeQualifier)

## 6.3 ELS Data Tables

### 6.3.1 Interaction

The *Interaction* table is the core data table in the ELS, storing the service interaction details.

Column	Type	Description
interactionId	Integer not null	Primary key

serviceTargetId	Varchar(100) not null	Qualified identifier of the owner (target) of the service.
serviceCategory	Varchar(100) not null	Category (domain) of service. E.g. Discharge Summary
interactionName	Varchar(100) not null	Name of the service interaction. E.g. put
serviceInterface	Varchar(100) not null	Namespace of the WSDL.
serviceProviderId	Varchar(100) not null	Qualified id of the service provider. This will equal the serviceTargetId of the parent interaction if the owner hosts their own services.
serviceEndpoint	Varchar(100) not null	URL of the Web service endpoint.

#### Constraints

1. Primary Key (interactionId)
2. Foreign Key (serviceTargetId) references Organisation
3. Foreign Key (serviceCategory) references RefServiceCategory
4. Foreign Key (interactionName) references RefInteraction

### 6.3.2 CertificateRef

Each *Interaction* can have zero or more *CertificateRefs*.

A *CertificateRef* provides details of an X.509 certificate required for secured communication. It is anticipated that the type of certificate data stored will correspond to a unique certificate identifier, such as Subject Key Identifier, or another uniquely identifying X.509 v3 extension. As an interim measure while the NASH does not exist, the entire X.509 certificate may be stored in PEM format.

Column	Type	Description
interactionId	Integer not null	References the Interaction table.
useQualifier	Varchar(100) not null	The allowable certificate usage. Such as Signing, KeyEncipherment or both.
typeQualifier	Varchar(100) not null	The type of certificate reference. This could be any unique attribute of the certificate, such as HDI or SKI, or even PEM if the whole certificate is referenced.
value	Varchar(*) not null	The value of the certificate reference. The contents depends on the typeQualifier. i.e. If the SKI is stored, this may be a base64 encoded byte array.

## Constraints

1. Primary Key (interactionId, useQualifier)
2. Foreign Key (interactionId) references Interaction
3. Foreign Key (useQualifier) references RefCertificateUseQualifier
4. Foreign Key (typeQualifier) references RefCertificateTypeQualifier

## 6.4 ELS Audit Tables

### 6.4.1 InteractionAudit

The *InteractionAudit* table is used to audit changes made to the *Interaction* table.

Column	Type	Description
interactionId	Integer not null	Identifier of an interaction.
serviceTargetId	Varchar(100) not null	Qualified identifier of the owner (target) of the service.
serviceCategory	Varchar(100) not null	Category (domain) of service. E.g. Discharge Summary
interactionName	Varchar(100) not null	Name of the service interaction. E.g. put
serviceInterface	Varchar(100) not null	Namespace of the WSDL.
serviceProviderId	Varchar(100) not null	Qualified id of the service provider. This will equal the serviceTargetId of the parent interaction if the owner hosts their own services.
serviceEndpoint	Varchar(100) not null	URL of the Web service endpoint.
operationType	Char(1) not null	Type of operation: I: Insert, U: Update, D: Delete.
operationDate	DateTime	Date/time of the operation.
operationUserId	Varchar(100) not null	Qualifier identifier of the user (organisation) that performed the operation.

### 6.4.2 CertificateRefAudit

The *CertificateRefAudit* table is used to audit changes to the *CertificateRef* table.

Column	Type	Description
interactionId	Integer not null	References the Interaction table.
useQualifier	Varchar(100) not null	The allowable certificate usage. Such as Signing, KeyEncipherment or both.
typeQualifier	Varchar(100) not null	The type of certificate reference. This could be

		any unique attribute of the certificate, such as HDI or SKI, or even PEM if the whole certificate is referenced.
operationType	Char(1) not null	Type of operation: I: Insert, U: Update, D: Delete.
operationDate	DateTime	Date/time of the operation.
operationUserId	Varchar(100) not null	Qualifier identifier of the user (organisation) that performed the operation.

## 6.5 Metadata Tables

Metadata tables are used to store configuration that can be used to customise the ELS server. Customisation may be required to implement changes in access control policy that are made after development is complete and allow changes to some business logic without the need for code changes.

### 6.5.1 Sequence

The Sequence table is used to allocate unique internal identifiers. As different database vendors provide different mechanisms for the creation of identifiers (such as sequences or identity columns), this table helps to improve the use of the ELS on different databases.

Column	Type	Description
sequenceName	Varchar(50) not null	Primary key
nextVal	Integer not null	The next available value of this sequence.
batchSize	Integer not null	For performance, sequences are allocated in batches.

#### Constraints

1. Primary Key (sequenceName)

## 7 References

- [DAO] Core J2EE Patterns – Data Access Object  
<http://java.sun.com/blueprints/corej2eepatterns/Patterns/DataAccessObject.html>
- [DTO] Java Design Pattern – Transfer Object  
<http://java.sun.com/blueprints/patterns/TransferObject.html>
- [WSP2008] NEHTA, Web Services Profile, V3.0, 17 March 2008
- [ELSA2009] NEHTA, Endpoint Location Service – Architecture, V1.2, 30 June 2009
- [ELSIG2008] NEHTA, Endpoint Location Service – Implementation Guide, V1.0, 6 October 2008