nehta

Technical Service Specification

Template Service Interface

Version 1.2 — 26 April 2012

Final

National E-Health Transition Authority

National E-Health Transition Authority Ltd
Level 25
56 Pitt Street
Sydney, NSW, 2000
Australia.
www.nehta.gov.au

Document Information

Approver	PCEHR Design Authority
Owner	Head of PCEHR
Contact officer	PCEHR Design Team
Contributors	PCEHR - Standards, Foundation and Architecture (SFA) Team

Version History

Date	Version	Name	Comments
2011-12-20	1.0	PCEHR Design Authority	Draft for Limited Release
2012-01-13	1.1	PCEHR Design Team	Updated after Tiger Team review
2012-04-26	1.2	PCEHR Template Team	Update to manifest data format. Inclusion of ATS5821 support

This page is intentionally blank

Table of Contents

This page is intentionally blank

Preface

Purpose

The purpose of this document is to provide an implementable technical interface specification for searching and obtaining templates from the Template Service.

Intended Audience

This NEHTA-managed specification is intended for:

- Developers and implementers of PCEHR Conformant Repositories (normative).
- Developers and implementers of software products which seek to interact with the PCEHR System (normative).
- Developers and implementers of software products that exchange clinical documents with other systems using interoperable formats (normative).

This document is technical and assumes a working knowledge of Web Services and the Unified Modelling Language.

Document Map

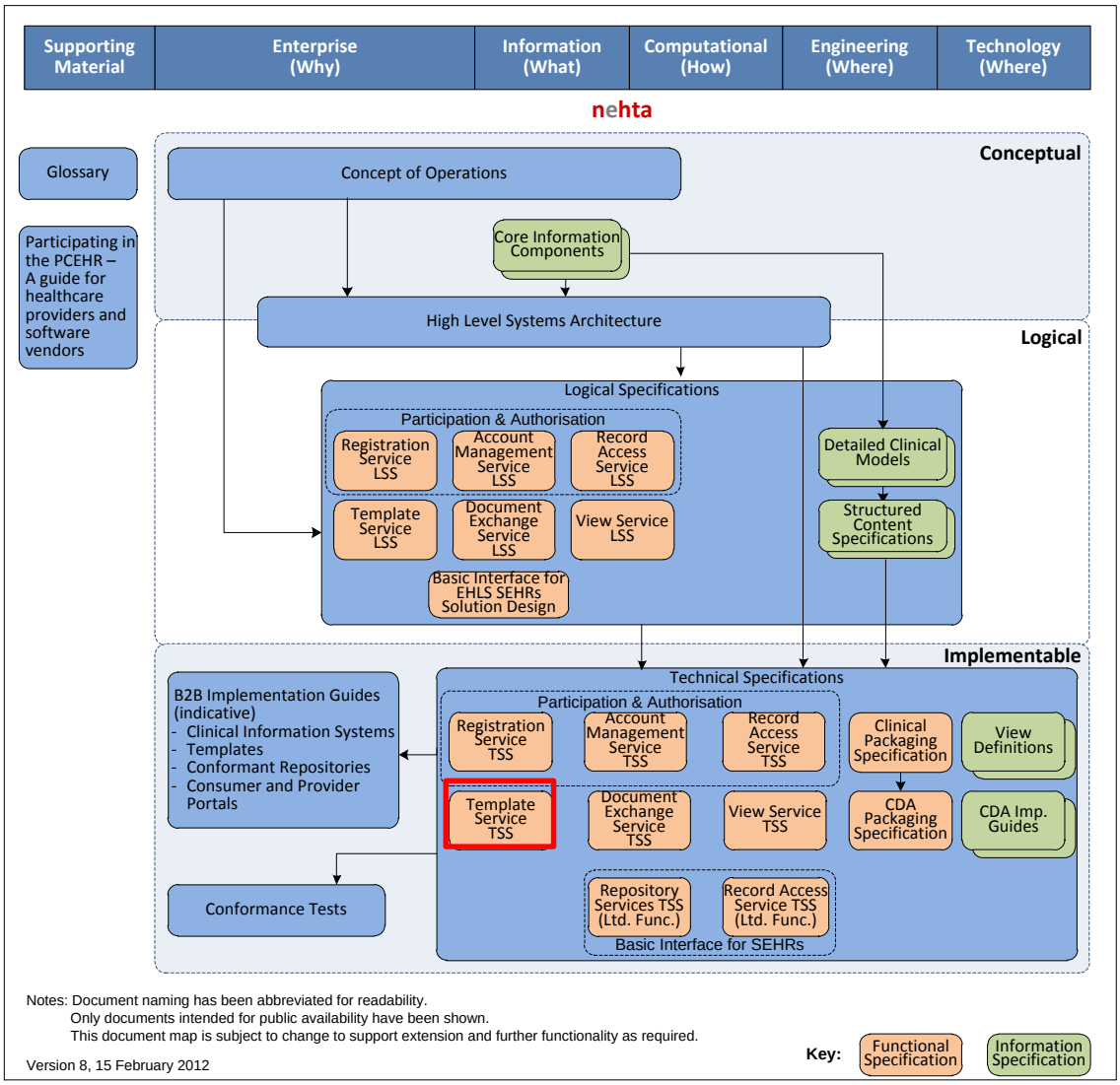

Figure 1 – Document map

Acronyms and Terminology

Please refer to 0 for definitions of the acronyms and terminology used in this document.

The keywords SHALL, SHALL NOT, SHOULD and SHOULD NOT in this document are to be interpreted as described in IETF's RFC 2119.

References

Please refer to Appendix C for details of the references used in this document.

1 Introduction

1.1 Context

A platform independent specification of the Template Service Interface is presented in the *Template Service Interface Logical Service Specification* [LSS2011]. The logical service specification does not mandate particular technologies as the basis for implementation and is therefore not implementable.

1.2 Scope of Document

This technical service specification binds the services, services interfaces and operations defined in the logical service specification [LSS2011] onto a technology platform to a level sufficient to support the implementation of interoperable systems.

1.3 Conformance Points

This specification contains conformance points that identify normative requirements that are to be met by identified members of the Template Service Interface Community (as described in the logical service specification [LSS2011]) in order to comply with this specification when interacting with the Template Service Interface.

Conformance points include requirements on a party (Template User) invoking the service and the party (Template Repository) providing the service.

Any capability required to meet a conformance point SHALL be considered part of the requirements to be met under this specification.

Conformance points are identified in this document by the following notation:

TPLT-T 0	This is an example only. Conformance points SHALL be numbered and contain an identifier of '**TPLT-T**' which identifies them as being applicable to the Template Service Interface technical service specification.

2 Standards and Technology Platform

A standards and technology platform is a collection of standards and technologies which may be used collectively to realise an implementation of one or more service interfaces specified within a logical service specification.

A single service interface within a logical specification must be realised fully by a single technology platform. However each service interface specified within a logical specification may be realised wholly on different standards and technology platforms.

The technology platform for this technical service specification is based on interaction through web service interfaces that conform to identified elements of the E-health Web Services Profile [ATS 5820-2010].

Conformance points

The following conformance points define the application of the E-health Web Services Profile [ATS 5820—2010] to service interactions.

TPLT-T 1	All implementations SHALL conform to the Web Services Base Profile from the Standards Australia E-health Web Services Profile [ATS 5820-2010] for all web service invocations.
TPLT-T 2	All implementations SHALL implement the TLS Security Profile from the Standards Australia E-health Web Services Profile [ATS 5820-2010] for all web service invocations.

3 Computational Viewpoint

The Computational Viewpoint addresses how the service interfaces and service operations defined in the logical service specification [LSS2011] map onto the operation and transport specifications provided by the standards and technology platform.

3.1 Service Interface Realisation

This section shows the service interfaces defined in the logical service specification [LSS2011] and specifies how these are realised on the chosen technology platform.

Conformance points

TPLT-T 3	All implementations SHALL comply with applicable conformance points specified in the *Template Service Interface Logical Service Specification* [LSS2011].

3.1.1 Template-Retrieval Service Interface

The Template Service enables implementers and systems to obtain standardised specifications for clinical documents to be exchanged within the Australian healthcare community. The Template Service is responsible for managing and storing the data representations associated with all of the data formats stored within a PCEHR, but can also store specifications for clinical documents not associated with the PCEHR.

Each template describes the data format, how to validate against this format, and the mechanisms for displaying or accessing all or parts of the data associated with a document matching the template.

The Template-Retrieval Service Interface provides the mechanism for those templates to be accessed and retrieved.

3.1.1.1 Actors and roles

Template User

The *Template User* represents the party responsible for obtaining template information from the Repository. This role will typically be realised by a Conformant Repository or Clinical Information System.

Template Repository

The *Template Repository* role represents the party responsible for persisting and sharing templates so that they may be accessed by authorised users. This role will be fulfilled by the national Template Service.

3.1.1.2 Precondition

Conformance points

TPLT-T 4	A connection SHALL be established between the Template User and Template Repository in conformance with the specification for TLS Security Profile from the Standards Australia E-health Web Services Profile [ATS 5820-2010].
TPLT-T 5	The *Template User* SHALL utilise a NASH certificate for authentication.
TPLT-T 35	The *Template User* and *Template Repository* SHALL include a Transmission Signature (sections 4.1.1.2 and 4.1.2.3) containing a signed attestation of elements contained within the SOAP message on all SOAP Request and Response messages, except where the response contains a SOAP Fault.

TPLT-T 36	The *Template User* and *Template Repository* SHALL perform the signature using a certificate that asserts the same identity as that asserted in the TLS connection.
TPLT-T 6	The *Template User* SHALL create a searchTemplateRequest or a retrieveTemplateRequest.

3.1.1.3 Postconditions

There are no postconditions, since the use the *Template User* makes of any retrieved information is outside the scope of the interface specification.

3.1.1.4 Interaction

Conformance points

TPLT-T 7	This operation SHALL be realised as a synchronous query between the *Template User* and the *Template Repository*. The response SHALL be returned on the same software communication connection.

3.1.1.5 Inputs, Outputs and Faults

This section details the data which is submitted to the service as an input, the response returned and the details of any faults. The data types are realised as XSDs within Appendix A.

The WSDLs and service interfaces for this service are also defined in Appendix A.

Input message

Please refer to section 4.1.1.

Output Message

Please refer to section 4.1.2.

Service Fault

Please refer to section 4.1.3.

4 Information Viewpoint

The Information Viewpoint specifies how the information concepts used to support interaction between service participants may be realised as physical constructs, such as XML Schema Definitions.

Note: The error code tables may be subject to extension as the development of the PCEHR system progresses.

4.1 Information Data Type Realisation

4.1.1 Input Message Data Types

4.1.1.1 Transmission Timestamp

The input messages contain a timestamp header element shared with all external interfaces to the PCEHR. The XSD for this request is defined in Appendix A.

The logical structure is realised as follows in Table 1.

Table 1 - timestamp Header

Element Name	Type	Cardinality	Remarks
timestamp		**1..1**	
created	dateTime	1..1	Time at SOAP message creation. Inclusive of Date, Time and UTC Timezone. E.g. 2011-10-25T03:06:13Z
expires	dateTime	0..1	For future use.
/timestamp	**-**	**-**	**-**

4.1.1.2 Transmission Signature

The input messages contain a signature header element shared with all external interfaces to the PCEHR. The XSD for this request is defined in Appendix A.

The logical structure is realised as follows in Table 2.

Table 2 - Transmission Signature in SOAP Header

Element Name	Type	Cardinality	Remarks
signature		**1..1**	
signature	ds:signature	1..1	A signed attestation of key SOAP message elements using the ATS 5821 specification.
/signature	**-**	**-**	**-**

Conformance points

TPLT-T 37	The elements signed by the Transmission Signature SHALL include the PCEHR Header element (as defined in section4.1.1.3) and the SOAP Body element.
TPLT-T 38	The elements signed by the Transmission Signature SHOULD include the Transmission Timestamp element (as defined in section 4.1.1.1)

TPLT-T 39	The *Template User* and *Template Repository* SHALL calculate the ds:DigestValue as specified in "Section 4. XML Signature Profile" as specified in [ATS 5821-2010] without the application of transient transmission specific transformations such as MTOM/XOP.
TPLT-T 40	The ds:SignedInfo element type SHALL be realised in conformance with "Section 4. XML Signature Profile" as specified in [ATS 5821-2010]
TPLT-T 41	The fragment identifier used to realise the ds:Reference element SHALL refer to the "ID" attribute specified in section 3.3 of [W3C-XML-1.1] of the element referenced.
TPLT-T 42	The ds:signature element type SHALL be realised in conformance with "Section 4. XML Signature Profile" as specified in [ATS 5821-2010]

4.1.1.3 Common Header

The input messages contain a common header shared with all external interfaces to the PCEHR. The input request structure is defined in the logical service specification [LSS2011]. The XSD for this request is defined in Appendix A.

The logical structure is realised as follows in Table 3.

Table 3 - PCEHRHeader

Element Name	Type	Cardinality	Remarks
PCEHRHeader		**1..1**	
User		**1..1**	
IDType	String	1..1	Values ("HPII", "PortalUserIdentifier", "LocalSystemIdentifier")
ID	String	1..1	PCEHR identity, 16 digit HPI-I number or Other User ID
role	String	0..1	Optional User Role
userName	String	1..1	Username
useRoleForAudit	Boolean	1..1	if true PCEHR will use `role` as the user name for audit, else PCEHR will use `userName` as the user name for audit
/User			
ihiNumber	String	0..1	PCEHR individual's 16 digit IHI number
productType		**1..1**	
vendor	String	1..1	client system's vendor name
productName	String	1..1	client system's product name
productVersion	String	1..1	client system's product version

platform	String	1..1	client system's platform
/productType			
clientSystemType	String	1..1	Values ("CCP","CPP", "CIS", "CSP", "CRP", "HI", "Medicare", "Other")
accessingOrganisation		**0..1**	
organisationID	String	1..1	The 16 digit Healthcare Organisation Identifier (HPI-O)
organisationName	String	1..1	Healthcare Organisation Name
alternateOrganisationName	String	0..1	Alternate Healthcare Organisation Name
/accessingOrganisation			
/PCEHRHeader			

Conformance points

TPLT-T 8	The *Template User* SHALL set the `User.ID` to either: • Preferably, if known, the 16 digit of HPI-I of the provider originating the Template Service interface request • Or alternatively, a local identifier of the provider/support operator originating the Template Service interface request. The Template User SHALL set the User.IDType to the relevant value to identify the type of User.ID.
TPLT-T 9	The *Template User* SHALL set the `productType.vendor` to the vendor name of the client system.
TPLT-T 10	The *Template User* SHALL set the `productType.productName` to the product name of the client system.
TPLT-T 11	The *Template User* SHALL set the `productType.productVersion` to the product version of the client system.
TPLT-T 12	The *Template User* SHALL set the `productType.platform` to the client system vendor.
TPLT-T 13	The *Template User* SHALL set the `clientSystemType` according to the type of system being used to originate the Template Service interface request.

4.1.1.4 searchTemplateRequest

The input request structure is defined in the logical service specification [LSS2011] The XSD for this request is defined in Appendix A.

The logical structure is realised as follows in Table 4.

Table 4 – Search Template Request

Element Name	Type	Cardinality	Remarks
PCEHRHeader		**1..1**	See Table 3 - PCEHRHeader
searchTemplate		**1..1**	
templateID	String	0..1	Optional unique identifier of template – enables retrieval of metadata for template without needing to retrieve entire package
templateMetaData	templateMetaDataType	0..1	Optional metadata input for searching
/searchTemplate			

Table 5 – templateMetadataType

Element Name	Type	Cardinality	Remarks
metadata		**1..300**	
name	String	1..1	Metadata element name
value	String	1..1	Metadata element value
/metadata			

Conformance points

TPLT-T 14	The *Template User* SHALL utilise the `WS-Addressing MessageID` in the SOAP Header to realise the logical *Request Id* from the LSS Common Header.

4.1.1.5 retrieveTemplateRequest

The input request structure is defined in the logical service specification [LSS2011] . The XSD for this request is defined in Appendix A.

The logical structure is realised as follows in Table 6.

Table 6 – Retrieve Template Request

Element Name	Type	Cardinality	Remarks
PCEHRHeader		**1..1**	See Table 3 - PCEHRHeader
getTemplate		**1..1**	
templateID	String	1..1	Unique identifier of template
serviceRequestorOption	String	1..1	Parameter controlling elements to be returned within the template. Values "FullPackage", "MachineUsable"
/getTemplate			

Conformance points

TPLT-T 15	The *Template User* SHALL utilise the `WS-Addressing MessageID` in the SOAP Header to realise the logical *Request Id* from the LSS Common Header

4.1.2 Output Message Data Types

4.1.2.1 searchTemplateResponse

The output response structure is defined in the logical service specification [LSS2011]. The XSD for this request is defined in Appendix A.

The logical structure is realised as follows in Table 7.

Table 7 – Search Template Response

Element Name	Type	Cardinality	Remarks
searchTemplateResponse			
responseStatus		1..1	See Table 15 – Response Status Response
template		**1..***	List of all templates matching search
templateMetaData	templateMetadataType	1..1	Metadata for template
usageMetaData	templateUsageMetadataType	0..1	Optional PCEHR usage information metadata
/template			
/searchTemplateResponse			

Table 8 – templateMetadataType

Element Name	Type	Cardinality	Remarks
metadata		**1..300**	
name	String	1..1	Metadata element name
value	String	1..1	Metadata element value
/templateMetadata			

Table 9 – templateUsageMetadataType

Element Name	Type	Cardinality	Remarks
usageMetadata		**1..1**	
PCEHRAcceptedStartDate	dateTime	1..1	Start date for PCEHR accepting documents based on the template
PCEHRAcceptedEndDate	dateTime	0..1	Optional end date when PCEHR ceases accepting documents based on the template
containsPCEHRAtomicData	boolean	0..1	Indicator that PCEHR considers atomic data may be found in documents based on this template
/usageMetadata			

Response Codes

Table 10 – searchTemplateResponse response codes

Status Code	Description	Suggested action
PCEHR_SUCCESS	SUCCESS	The list of metadata for the matching templates can be processed.
PCEHR_ERROR_1802	No Search Item	Where a user interaction is occurring, a message should be displayed indicating that no results were found. The search may be retried with less restrictive search criteria to enable finding results which may be excluded by the submitted search criteria.
PCEHR_ERROR_1803	Invalid Search criteria	Where a user interaction is occurring, a message should be displayed indicating that the search criteria is invalid. This may be due to user input within fields (such as invalid logical requests), or the supply of unknown values for the "name" parameters. The search may be retried with different criteria.
PCEHR_ERROR_1804	Too many results found to be returned	Where a user interaction is occurring, a message should be displayed indicating that too many results matched the search criteria, and they cannot be returned. The search may be retried with more restrictive search criteria to reduce the number of matching results.

4.1.2.2 retrieveTemplateResponse

The output response structure is defined in the logical service specification [LSS2011] . The XSD for this request is defined in Appendix A.

The logical structure is realised as follows in Table 11.

Table 11 – Retrieve Template Response

Element Name	Type	Cardinality	Remarks
getTemplateResponse			
responseStatus		1..1	See Table 15 – Response Status Response
template		**0..1**	Template being retrieved
templateMetaData	templateMetadataType	1..1	Metadata for template
usageMetaData	templateUsageMetadataType	0..1	Optional PCEHR usage information metadata
Package	Base64binary	1..1	The template package
/template			
persistInCacheExpiry	dateTime	0..1	Optional expiry time for any cached copy of the retrieval request
/getTemplateResponse			

Table 12 – templateMetadataType

Element Name	Type	Cardinality	Remarks
metadata		**1..300**	
name	String	1..1	Metadata element name
value	String	1..1	Metadata element value
/metadata			

Table 13 – templateUsageMetadataType

Element Name	Type	Cardinality	Remarks
usageMetadata		**1..300**	
PCEHRAcceptedStartDate	dateTime	1..1	Start date for PCEHR accepting documents based on the template
PCEHRAcceptedEndDate	dateTime	0..1	Optional end date when PCEHR ceases accepting documents based on the template
containsPCEHRAtomicData	boolean	0..1	Indicator that PCEHR considers atomic data may be found in documents based on this template
/usageMetadata			

Response Codes

Table 14 - retrieveTemplateResponse response codes

Status Code	Description	Suggested Action
PCEHR_SUCCESS	SUCCESS	The template package can be processed.
PCEHR_ERROR_1800	No Package	The package for the specified Template ID is not available for download. An error should be logged or displayed. The retrieval should not be retried.
PCEHR_ERROR_1801	Invalid Template ID	The Template Identifier is not present in the Template Service. This may be because the format of the identifier is invalid, or the template package is not present. An error should be logged or displayed. The retrieval should not be retried.

4.1.2.3 Transmission Signature

The ouput messages contain a signature header element shared with all external interfaces to the PCEHR. This is the same element as is present in the input message.

The logical structure is realised as defined in Table 2 in Section4.1.1.2 above.

Conformance points

TPLT-T 43	The elements signed by the Transmission Signature SHALL include the SOAP Body element
TPLT-T 44	The *Template User* and *Template Repository* SHALL calculate the ds:DigestValue as specified in "Section 4. XML Signature Profile" as specified in [ATS 5821-2010] without the application of transient transmission specific transformations such as MTOM/XOP.
TPLT-T 45	The ds:SignedInfo element type SHALL be realised in conformance with "Section 4. XML Signature Profile" as specified in [ATS 5821-2010]

TPLT-T 46	The fragment identifier used to realise the ds:Reference element SHALL refer to the "ID" attribute specified in section 3.3 of [W3C-XML-1.1] of the element referenced.
TPLT-T 47	The ds:signature element type SHALL be realised in conformance with "Section 4. XML Signature Profile" as specified in [ATS 5821-2010]

4.1.3 Service Fault Data Types

4.1.3.1 GenericServiceFault

The output response structure is defined in the logical service specification [LSS2011]. The XSD for this request is defined in Appendix A.

The logical structure is realised as follows in Table 15.

Table 15 – Response Status Response

Element Name	Type	Cardinality	Remarks
responseStatus		**1..1**	
code	String	1..1	Status Code for the result of the transaction
description	String	1..1	Brief status description
details	String	0..1	Additional detail of the response
/responseStatus			

The following conformance points apply to a provider of this service when creating a service fault.

Conformance Points

TPLT-T 16	The *Template Repository* SHALL respond with a SOAP fault if an error occurs in processing the **`Template-Retrieval`** operation.
TPLT-T 17	The *Template Repository* SHOULD respond with a fault containing a **`se:standardError`** element in the circumstances defined for this fault by [ATS 5820—2010].

4.1.3.2 StandardError Fault Codes

The standard errors are shared with all external interfaces to the PCEHR. The following SOAP faults may be returned by any of those interfaces, and are not necessarily relevant to the Template Service interface. The *Template User* needs to be able to handle any SOAP fault encountered, even if it has not been previously documented.

Conformance Points

TPLT-T 48	The *Template User* SHALL be able to accept any SOAP fault generated by the *Template Repository*.

Table 16 – Standard Error fault codes

errorCode	Message
badWsaMessageId	PCEHR_ERROR_0001 - Message ID element is missing
badlyFormedMsg	PCEHR_ERROR_0002 - SOAP header fault
badlyFormedMsg	PCEHR_ERROR_0003 - SOAP body fault
notAuthorised	PCEHR_ERROR_0004 - Authorisation denied
serviceTemporaryUnavailable	PCEHR_ERROR_0005 - Back end system temporary unavailable
badParam	PCEHR_ERROR_0006 - Invalid common parameter
badParam	PCEHR_ERROR_0007 - Invalid IHI
badParam	PCEHR_ERROR_0008 - Invalid HPI-I
badParam	PCEHR_ERROR_0009 - Invalid HPI-O
badlyFormedMsg	PCEHR_ERROR_0010 - The request did not contain the expected message format
serviceTemporaryUnavailable	PCEHR_ERROR_0011 - Unexpected service exception error
serviceTemporaryUnavailable	PCEHR_ERROR_0012 - Unexpected back end exception error
serviceTemporaryUnavailable	PCEHR_ERROR_0013 - Invalid back end response
serviceTemporaryUnavailable	PCEHR_ERROR_0014 - Unknown back end error code
badParam	PCEHR_ERROR_0501 - Accessing organisation element is not required
badParam	PCEHR_ERROR_0502 - Accessing organisation element is missing
badParam	PCEHR_ERROR_0503 - User name for audit element is missing
badParam	PCEHR_ERROR_0504 - User role for audit element is missing
badParam	PCEHR_ERROR_0505 - Invalid HPI-O
badlyFormedMsg	PCEHR_ERROR_0506 - Invalid request
serviceTemporaryUnavailable	PCEHR_ERROR_0507 - Unexpected service exception error
badlyFormedMsg	PCEHR_ERROR_0509 - SOAP header fault
badlyFormedMsg	PCEHR_ERROR_0510 - SOAP body fault
badParam	PCEHR_ERROR_0511 - ClientSystemType is missing
badEncryption	PCEHR_ERROR_0512 - Not a HTTPS request
badWsaAction	PCEHR_ERROR_0513 - Invalid WS-addressing action
badWsaTo	PCEHR_ERROR_0514 - WS-addressing "to" field is missing
serviceTemporaryUnavailable	PCEHR_ERROR_0515 - Audit service temporary unavailable
serviceTemporaryUnavailable	PCEHR_ERROR_0516 - Access service temporary unavailable
serviceTemporaryUnavailable	PCEHR_ERROR_0517 - Service temporary unavailable

errorCode	Message
serviceTemporaryUnavailable	PCEHR_ERROR_0518 - Back end system temporary unavailable
notAuthorised	PCEHR_ERROR_0519 - System authorisation denied
badSignature	PCEHR_ERROR_0520 - The SOAP request has not been signed, or is signed incorrectly
badAlgorithmC14N	PCEHR_ERROR_0521 - The algorithm used for canonicalizing the data is not acceptable
badAlgorithmDigest	PCEHR_ERROR_0522 - The algorithm used for calculating the digest is not acceptable
badAlgorithmSignature	PCEHR_ERROR_0523 - The algorithm used for signing is not acceptable

4.2 SOAP Header

Conformance points

TPLT-T 18	The *Template User* SHALL set the values of the `MessageID`, `TO` and `Action` fields of the `WS-Addressing` header in accordance with ATS 5820-2010 Section 6 – Metadata specification.
TPLT-T 19	The *Template Repository* SHALL set the values of the `MessageID`, `Action` and `RelatedTo` fields of the `WS-Addressing` header in accordance with ATS 5820-2010 Section 6 – Metadata specification.

4.3 Template Package

The Template Service Interface is a mechanism for accessing templates (as template packages) provided by the Template Service.

The detail of the contents and use of a template package is not within the scope of the Template Service Interface, so is not specified in this document. In brief, a Template encapsulates the definition of a clinical document type and includes:

- a unique identifier
- data definitions
- data validation definition
- information about how to render a clinical document
- supporting material (such as implementation guides).

This information is packaged as a number of different files to meet the various needs, as well as metadata information about each file, and about the template as a whole.

The following defines the structure of the template package as a component retrieved by the interface.

4.3.1 Template Package Structure

The template itself consists of a number of files with different uses. They are transported as a ZIP file, which may be considered to have the following folder structure (Figure 2).

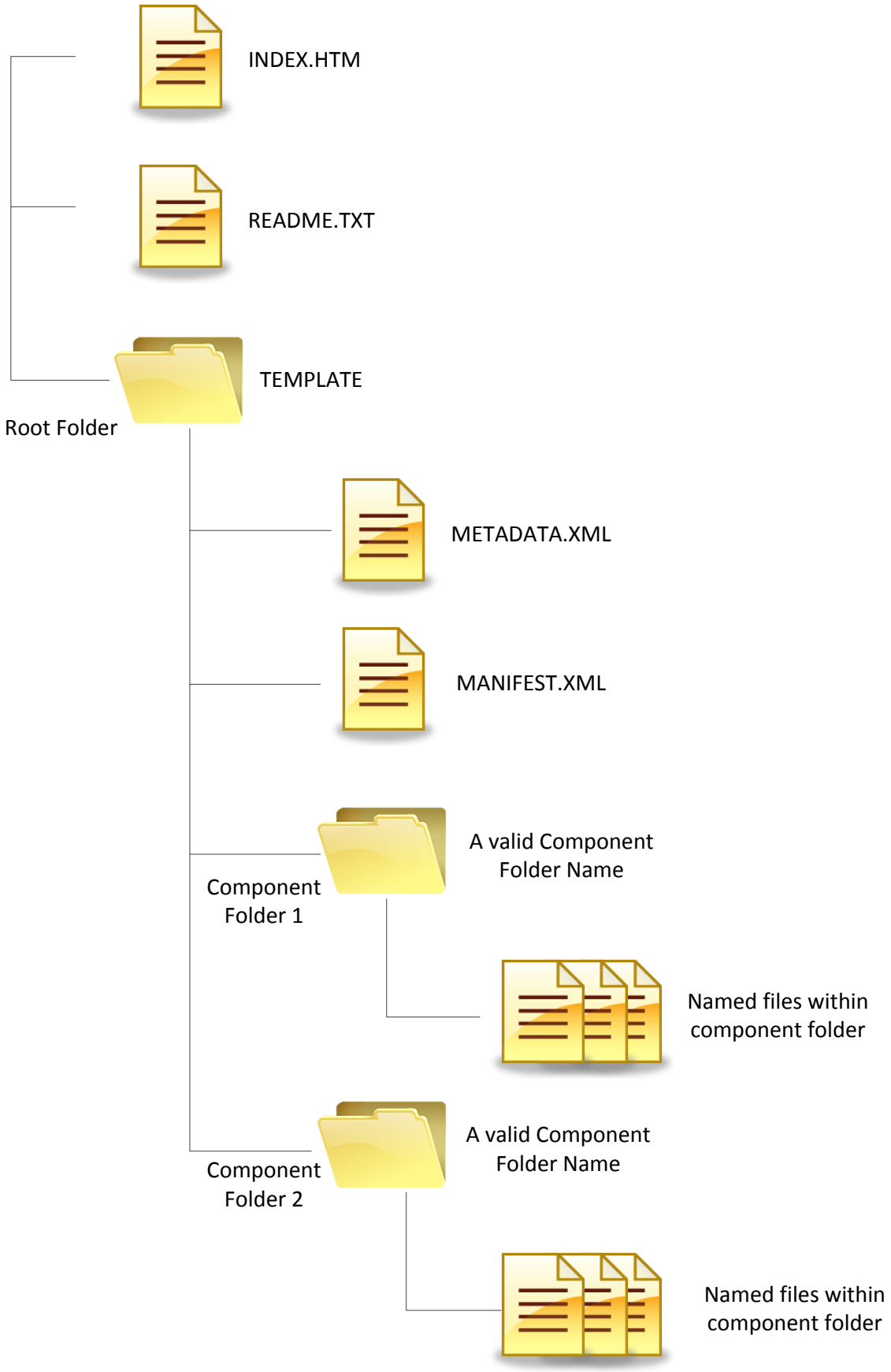

Figure 2 - Template Package Structure

Table 17 describes the files in the template package.

Table 17 – Files in the Template Package

File Name	Required	Description
INDEX.HTM	Optional	This file provides an HTML entry point into the package contents. If present, it can be used to provide a means to display or render selected components of the package.
README.TXT	Optional	This file contains informative data about the Template package. If present, it will contain information about the actual retrieval (such as search parameters, date of access etc.).
Root Folder – TEMPLATE	Mandatory	This folder defines the root of the template package contents. The name of this folder is fixed to TEMPLATE.
METADATA.XML	Mandatory	This file contains template package metadata. See [LSS2011] for logical content.
MANIFEST.XML	Mandatory	This file contains the definition for the template component files. It contains metadata for each file contained within the component type folders, including the file name and location. The [LSS2011] defines the structure.
Component type folders	Mandatory	Template components get bundled into component types to enable segregation of content, and enhance usability when accessed via the Template Portal. There is no naming restriction on the component folder names, but the files contained within them must be explicitly referenced in the MANIFEST.XML file. Refer [LSS2011] for conformance point for component type folder naming information
Components	Mandatory	Within each component folder, there are different format of files are stored. Each file in the component folder is referenced in MANIFEST.XML file.

TPLT-T 20	The Root folder SHALL be given the fixed name of TEMPLATE.
TPLT-T 21	The TEMPLATE folder shall contain only folders and the METADATA.XML and MANIFEST.XML files.
TPLT-T 22	Any automated processing SHALL NOT make use of the INDEX.HTM or README.TXT files for any explicit or implicit part of the process of using the template.
TPLT-T 23	Implementations SHALL ignore all files within component type folders that are not specified in the manifest file, and other content that is outside the TEMPLATE folder.
TPLT-T 24	A template package SHALL be a ZIP archive that conforms to the ZIP archive specified by [PK2007].

Package Parts

While distinct parts of the template package are referred to as folders, within a ZIP they are merely ZIP items with fully qualified names (folders may be created when extracting from the archive).

The logical package parts are:

- Metadata part
- Manifest part
- Package components within component folders.

Package part terms

The following defines the key terms used in the specification of the packaging parts below.

- The *n-prefix* is a value which is prefixed to a ZIP item to produce a fully qualified name. It may be seen as representing the root folder path of the template package.
- The *n-group* is the name of the component group folder.
- The *n-filename* is the name of a component part of the template package.
- An *n-token* represents the definition of valid characters which may be used in the above names.

TPLT-T 25	The *n-prefix* SHALL be the string "TEMPLATE/"
TPLT-T 26	An *n-group* SHALL be an *n-token*.
TPLT-T 27	An *n-filename* SHALL be an *n-token*.
TPLT-T 28	An *n-token* SHALL be a sequence of one or more the following US-ASCII characters: 0-9, A-Z, a-z, "-", "_" and "."

It is best practice to avoid starting or ending *n-tokens* with a full-stop, or starting with a hyphen.

TPLT-T 29	All *ZIP item names* SHALL consist of only printable US-ASCII characters.
TPLT-T 30	The *template metadata part* SHALL be stored in a ZIP item with *a ZIP item name* of the concatenation of *n-prefix* and the literal string "`METADATA.XML`".
TPLT-T 31	The *template metadata part* SHALL be an XML document.
TPLT-T 32	The *template manifest part* SHALL be stored in *a ZIP item* with a *ZIP item name* of the concatenation of *n-prefix* and the literal string "`MANIFEST.XML`".
TPLT-T 33	The *template metadata part* SHALL be an XML document.
TPLT-T 34	Any automated processing SHALL NOT make use of the template component folders as an explicit or implicit part of the process of using the template.

4.3.2 Package Structure Content

The structure of a template package is defined above and within the [LSS2011]. The detailed specification of the content which makes up a Template Package is defined in the Template Package Technical Specification [PKGSPEC].

5 Engineering Viewpoint

The engineering viewpoint includes definitions of mechanisms and functions to support distributed interactions between computational objects as a series of templates (i.e. patterns) for computational interactions. These, in turn, are parameterized to support a range of different policies defined in the enterprise, information or computational specifications.

Examples of such functions are:

- repository (e.g. storage and information organisation function)
- security (e.g. access control, authentication, security audit, integrity and confidentiality functions)
- network services (e.g. naming services, time services and directory)
- type repository functions.

The engineering viewpoint is relevant for those who are providing infrastructure services and functions, such as system architects, network architects, security architects and middleware specialists.

5.1 Discovery Services

The location of the templateRetrieval Service interface will be shared between parties before interaction. Dynamic discovery mechanisms will not be provided.

Appendix A XSD and WSDL

A.1 Data Types

A.1.1 XML schema – Template Retrieval Common Data Types

A.1.1.1 PCEHR_TemplatesTypes.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Version 1.0.1 issued 11 April 2012--><xs:schema
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:ns1="http://ns.electronichealth.net.au/tplt/xsd/common/TemplatesCoreElem
ents/1.0"
targetNamespace="http://ns.electronichealth.net.au/tplt/xsd/common/TemplatesCo
reElements/1.0" elementFormDefault="qualified"
attributeFormDefault="unqualified">
   <xs:complexType name="templateMetadataType">
      <xs:sequence>
         <xs:element name="metadata" minOccurs="1" maxOccurs="300">
            <xs:complexType>
               <xs:sequence>
                  <xs:element name="name" minOccurs="1" maxOccurs="1">
                     <xs:simpleType>
                        <xs:restriction base="xs:string">
                           <xs:minLength value="1"/>
                           <xs:maxLength value="50"/>
                        </xs:restriction>
                     </xs:simpleType>
                  </xs:element>
                  <xs:element name="value" minOccurs="1" maxOccurs="1">
                     <xs:simpleType>
                        <xs:restriction base="xs:string">
                           <xs:minLength value="1"/>
                           <xs:maxLength value="255"/>
                        </xs:restriction>
                     </xs:simpleType>
                  </xs:element>
               </xs:sequence>
            </xs:complexType>
         </xs:element>
      </xs:sequence>
   </xs:complexType>
   <xs:complexType name="templateUsageMetadataType">
      <xs:sequence>
         <xs:element name="PCEHRAcceptedStartDate" maxOccurs="1">
            <xs:simpleType>
               <xs:restriction base="xs:dateTime">
                  <xs:whiteSpace value="collapse"/>
               </xs:restriction>
            </xs:simpleType>
         </xs:element>
         <xs:element name="PCEHRAcceptedEndDate" minOccurs="0" maxOccurs="1">
            <xs:simpleType>
               <xs:restriction base="xs:dateTime">
                  <xs:whiteSpace value="collapse"/>
               </xs:restriction>
```

```
          </xs:simpleType>
       </xs:element>
       <xs:element name="containsPCEHRAtomicData" minOccurs="0"
maxOccurs="1">
          <xs:simpleType>
             <xs:restriction base="xs:boolean">
                <xs:whiteSpace value="collapse"/>
             </xs:restriction>
          </xs:simpleType>
       </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

A.1.2 PCEHR_GetTemplate.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Version 1.0.1 issued 11 April 2012--><xs:schema
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:ns1="http://ns.electronichealth.net.au/pcehr/xsd/common/CommonCoreElemen
ts/1.0"
xmlns:ns2="http://ns.electronichealth.net.au/tplt/xsd/interfaces/GetTemplate/1
.0"
xmlns:ns3="http://ns.electronichealth.net.au/tplt/xsd/common/TemplatesCoreElem
ents/1.0"
targetNamespace="http://ns.electronichealth.net.au/tplt/xsd/interfaces/GetTemp
late/1.0" elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:import
namespace="http://ns.electronichealth.net.au/pcehr/xsd/common/CommonCoreElemen
ts/1.0" schemaLocation="../Common/PCEHR_CommonTypes.xsd"/>
  <xs:import
namespace="http://ns.electronichealth.net.au/tplt/xsd/common/TemplatesCoreElem
ents/1.0" schemaLocation="../Common/PCEHR_TemplatesTypes.xsd"/>
  <xs:element name="getTemplate">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="templateID">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:minLength value="1"/>
              <xs:maxLength value="100"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
        <xs:element name="serviceRequestorOption">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:minLength value="1"/>
              <xs:maxLength value="100"/>
              <xs:whiteSpace value="collapse"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
```

```
   <xs:element name="getTemplateResponse">
      <xs:complexType>
         <xs:sequence>
            <xs:element name="responseStatus" type="ns1:responseStatusType"/>
            <xs:element name="template" minOccurs="0">
               <xs:complexType>
                  <xs:sequence>
                     <xs:element name="templateMetadata"
type="ns3:templateMetadataType"/>
                     <xs:element name="usageMetadata"
type="ns3:templateUsageMetadataType" minOccurs="0"/>
                     <xs:element name="package">
                        <xs:simpleType>
                           <xs:restriction base="xs:base64Binary"/>
                        </xs:simpleType>
                     </xs:element>
                  </xs:sequence>
               </xs:complexType>
            </xs:element>
            <xs:element name="persistInCacheExpiry" minOccurs="0">
               <xs:simpleType>
                  <xs:restriction base="xs:dateTime"/>
               </xs:simpleType>
            </xs:element>
         </xs:sequence>
      </xs:complexType>
   </xs:element>
</xs:schema>
```

PCEHR_GetTemplate
.xsd

A.1.3 PCEHR_SearchTemplate.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Version 1.0.1 issued 11 April 2012--><xs:schema
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:ns1="http://ns.electronichealth.net.au/pcehr/xsd/common/CommonCoreElemen
ts/1.0"
xmlns:ns2="http://ns.electronichealth.net.au/tplt/xsd/interfaces/SearchTemplat
e/1.0"
xmlns:ns3="http://ns.electronichealth.net.au/tplt/xsd/common/TemplatesCoreElem
ents/1.0"
targetNamespace="http://ns.electronichealth.net.au/tplt/xsd/interfaces/SearchT
emplate/1.0" elementFormDefault="qualified"
attributeFormDefault="unqualified">
   <xs:import
namespace="http://ns.electronichealth.net.au/pcehr/xsd/common/CommonCoreElemen
ts/1.0" schemaLocation="../Common/PCEHR_CommonTypes.xsd"/>
   <xs:import
namespace="http://ns.electronichealth.net.au/tplt/xsd/common/TemplatesCoreElem
ents/1.0" schemaLocation="../Common/PCEHR_TemplatesTypes.xsd"/>
   <xs:element name="searchTemplate">
      <xs:complexType>
         <xs:sequence>
            <xs:element name="templateID" minOccurs="0">
               <xs:simpleType>
                  <xs:restriction base="xs:string">
                     <xs:maxLength value="100"/>
                     <xs:minLength value="1"/>
```

```
                </xs:restriction>
             </xs:simpleType>
          </xs:element>
          <xs:element name="templateMetadata" type="ns3:templateMetadataType"
minOccurs="0"/>
        </xs:sequence>
     </xs:complexType>
   </xs:element>
   <xs:element name="searchTemplateResponse">
     <xs:complexType>
       <xs:sequence>
          <xs:element name="responseStatus" type="ns1:responseStatusType"/>
          <xs:element name="template" minOccurs="0" maxOccurs="unbounded">
             <xs:complexType>
                <xs:sequence>
                   <xs:element name="templateMetadata"
type="ns3:templateMetadataType"/>
                   <xs:element name="usageMetadata"
type="ns3:templateUsageMetadataType" minOccurs="0"/>
                </xs:sequence>
             </xs:complexType>
          </xs:element>
       </xs:sequence>
     </xs:complexType>
   </xs:element>
</xs:schema>
```

A.1.4 XML schema – PCEHR Common Header Data Types

The content shown for the PCEHR common header data types is extracted from the schema file containing all PCEHR common data types. The full schema file is available for executable use, but for readability, only common header related items have been included in the figure below.

A.1.4.1 PCEHR_CommonTypes.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Version 1.0.1 issued 11 April 2012--><xs:schema
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:ns1="http://ns.electronichealth.net.au/pcehr/xsd/common/CommonCoreElemen
ts/1.0" xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
targetNamespace="http://ns.electronichealth.net.au/pcehr/xsd/common/CommonCore
Elements/1.0" elementFormDefault="qualified"
attributeFormDefault="unqualified">
<xs:import namespace="http://www.w3.org/2000/09/xmldsig#"
schemaLocation="xmldsig-core-schema.xsd"/>
   <xs:element name="timestamp" type="ns1:timestampType"/>
   <xs:complexType name="timestampType">
     <xs:sequence>
        <xs:element name="created" type="xs:dateTime" minOccurs="1"
maxOccurs="1"/>
        <xs:element name="expires" type="xs:dateTime" minOccurs="0"
maxOccurs="1"/>
     </xs:sequence>
   </xs:complexType>
   <xs:element name="signature" type="ns1:signatureContainerType"/>
   <xs:complexType name="signatureContainerType">
```

```
      <xs:sequence>
         <xs:element ref="ds:Signature" minOccurs="1" maxOccurs="1"/>
      </xs:sequence>
   </xs:complexType>
```

… Elements removed from file …

```
<xs:complexType name="responseStatusType">
   <xs:sequence>
      <xs:element name="code" minOccurs="1" maxOccurs="1">
         <xs:simpleType>
            <xs:restriction base="xs:string">
               <xs:minLength value="1"/>
            </xs:restriction>
         </xs:simpleType>
      </xs:element>
      <xs:element name="description" type="xs:string" minOccurs="1"
maxOccurs="1"/>
      <xs:element name="details" type="xs:string" minOccurs="0" maxOccurs="1"/>
   </xs:sequence>
</xs:complexType>
```

… Elements removed from file …

```
<xs:element name="PCEHRHeader">
   <xs:complexType>
      <xs:sequence>
         <xs:element name="User" minOccurs="1" maxOccurs="1">
            <xs:complexType>
               <xs:sequence>
                  <xs:element name="IDType" minOccurs="1" maxOccurs="1">
                     <xs:simpleType>
                        <xs:restriction base="xs:string">
                           <xs:enumeration value="HPII"/>
                           <xs:enumeration value="PortalUserIdentifier"/>
                           <xs:enumeration value="LocalSystemIdentifier"/>
                        </xs:restriction>
                     </xs:simpleType>
                  </xs:element>
                  <xs:element name="ID" type="xs:string" minOccurs="1"
maxOccurs="1"/>
                  <xs:element name="role" type="xs:string" minOccurs="0"
maxOccurs="1"/>
                  <xs:element name="userName" type="xs:string" minOccurs="1"
maxOccurs="1"/>
                  <xs:element name="useRoleForAudit" type="xs:boolean"
minOccurs="1" maxOccurs="1"/>
               </xs:sequence>
            </xs:complexType>
         </xs:element>
         <xs:element ref="ns1:ihiNumber" minOccurs="0" maxOccurs="1"/>
         <xs:element name="productType" minOccurs="1" maxOccurs="1">
            <xs:complexType>
               <xs:sequence>
                  <xs:element name="vendor" type="xs:string" minOccurs="1"
maxOccurs="1"/>
                  <xs:element name="productName" type="xs:string" minOccurs="1"
maxOccurs="1"/>
                  <xs:element name="productVersion" type="xs:string"
minOccurs="1" maxOccurs="1"/>
                  <xs:element name="platform" type="xs:string" minOccurs="1"
maxOccurs="1"/>
               </xs:sequence>
            </xs:complexType>
         </xs:element>
         <xs:element name="clientSystemType" minOccurs="1" maxOccurs="1">
            <xs:simpleType>
```

```
            <xs:restriction base="xs:string">
               <xs:enumeration value="CIS"/>
               <xs:enumeration value="CSP"/>
               <xs:enumeration value="CRP"/>
               <xs:enumeration value="HI"/>
               <xs:enumeration value="Medicare"/>
               <xs:enumeration value="CPP"/>
               <xs:enumeration value="CCP"/>
               <xs:enumeration value="Other"/>
            </xs:restriction>
         </xs:simpleType>
      </xs:element>
      <xs:element name="accessingOrganisation" minOccurs="0" maxOccurs="1">
         <xs:complexType>
            <xs:sequence>
               <xs:element name="organisationID" type="xs:string"
minOccurs="1" maxOccurs="1"/>
               <xs:element name="organisationName" type="xs:string"
minOccurs="1" maxOccurs="1"/>
               <xs:element name="alternateOrganisationName" type="xs:string"
minOccurs="0" maxOccurs="1"/>
            </xs:sequence>
         </xs:complexType>
      </xs:element>
   </xs:sequence>
</xs:complexType>
</xs:element>
```

… Elements removed from file …

```
<xs:element name="ihiNumber">
   <xs:simpleType>
      <xs:restriction base="xs:string">
         <xs:minLength value="16"/>
         <xs:maxLength value="16"/>
      </xs:restriction>
   </xs:simpleType>
</xs:element>
```

… Elements removed from file …

```
</xs:schema>
```

A.1.5 XML Schema – Standard Error

A.1.5.1 wsp-StandardError-2010.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"

xmlns:tns="http://ns.electronichealth.net.au/wsp/xsd/StandardError/2010"

targetNamespace="http://ns.electronichealth.net.au/wsp/xsd/StandardError/2010"
            elementFormDefault="qualified">
   <xsd:element name="standardError" type="tns:StandardErrorType"/>
   <xsd:simpleType name="StandardErrorCodeType">
      <xsd:restriction base="xsd:string">
         <xsd:enumeration value="servicePermanentUnavailable"/>
         <xsd:enumeration value="serviceTemporaryUnavailable"/>
         <xsd:enumeration value="certificateSkiMissing"/>
         <xsd:enumeration value="certificateKeyUsage"/>
         <xsd:enumeration value="invalidCredentials"/>
```

```
         <xsd:enumeration value="notAuthenticated"/>
         <xsd:enumeration value="notAuthorised"/>
         <xsd:enumeration value="badParam"/>
         <xsd:enumeration value="badlyFormedMsg"/>
         <xsd:enumeration value="badTimestamp"/>
         <xsd:enumeration value="badSignature"/>
         <xsd:enumeration value="badEncryption"/>
         <xsd:enumeration value="badSigEncOrder"/>
         <xsd:enumeration value="badCertificateTransmitted"/>
         <xsd:enumeration value="badWsaAction"/>
         <xsd:enumeration value="badWsaMessageId"/>
         <xsd:enumeration value="badWsaTo"/>
         <xsd:enumeration value="badAlgorithmDataEncryption"/>
         <xsd:enumeration value="badAlgorithmKeyEncryption"/>
         <xsd:enumeration value="badAlgorithmC14N"/>
         <xsd:enumeration value="badAlgorithmDigest"/>
         <xsd:enumeration value="badAlgorithmSignature"/>
      </xsd:restriction>
   </xsd:simpleType>
   <xsd:complexType name="StandardErrorType">
      <xsd:sequence>
         <xsd:element name="errorCode" type="tns:StandardErrorCodeType"
minOccurs="1" maxOccurs="1"/>
         <xsd:element name="message" type="xsd:string" minOccurs="1"
maxOccurs="1"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:schema>
```

wsp-StandardError-2010.xsd

A.2 Template Retrieval Interface

A.2.1 Interface Definition

The following WSDL specifications define the Template Retrieval SOAP interfaces.

A.2.1.1 B2B_GetTemplateInterface.wsdl

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Version 1.1 issued 11 April 2012-->
<wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata"
xmlns:pcehr="http://ns.electronichealth.net.au/tplt/svc/GetTemplate/1.1"
xmlns:
pcehr_b2b="http://ns.electronichealth.net.au/pcehr/xsd/common/CommonCoreElemen
ts/1.0"
xmlns:ns2="http://ns.electronichealth.net.au/tplt/xsd/interfaces/GetTemplate/1
.0" xmlns:tns="http://ns.electronichealth.net.au/wsp/xsd/StandardError/2010"
xmlns:ns=http://www.w3.org/2000/09/xmldsig#
xmlns:ns1="http://ns.electronichealth.net.au/tplt/xsd/common/TemplatesCoreElem
ents/1.0" name="getTemplate"
targetNamespace="http://ns.electronichealth.net.au/tplt/svc/GetTemplate/1.1">
   <wsdl:types>
      <xsd:schema
targetNamespace="http://ns.electronichealth.net.au/tplt/svc/GetTemplate/1.1"
elementFormDefault="qualified">
         <xsd:import
namespace="http://ns.electronichealth.net.au/pcehr/xsd/common/CommonCoreElemen
ts/1.0" schemaLocation="../../schema/Common/PCEHR_CommonTypes.xsd"/>
         <xsd:import
namespace="http://ns.electronichealth.net.au/tplt/xsd/interfaces/GetTemplate/1
.0" schemaLocation="../../schema/External/PCEHR_GetTemplate.xsd"/>
         <xsd:import
namespace="http://ns.electronichealth.net.au/wsp/xsd/StandardError/2010"
schemaLocation="../../schema/Common/wsp-StandardError-2010.xsd"/>
      </xsd:schema>
   </wsdl:types>
<wsdl:message name="getTemplateInMsg">
      <wsdl:part name="timestampHeader" element="pcehr_b2b:timestamp"/>
      <wsdl:part name="signatureHeader" element="pcehr_b2b:signature"/>
      <wsdl:part name="PCEHRHeader" element="pcehr_b2b:PCEHRHeader"/>
      <wsdl:part name="parameters" element="ns2:getTemplate"/>
   </wsdl:message> <wsdl:message name="getTemplateOutMsg">
 <wsdl:part name="signatureHeader" element="pcehr_b2b:signature"/>
      <wsdl:part name="parameters" element="ns2:getTemplateResponse"/>
   </wsdl:message>
   <wsdl:message name="standardErrorMsg">
      <wsdl:part name="parameters" element="tns:standardError"/>
   </wsdl:message>
   <wsdl:portType name="GetTemplatePortType">
      <wsdl:operation name="getTemplate" parameterOrder="parameters">
         <wsdl:input message="pcehr:getTemplateInMsg"
wsam:Action="http://ns.electronichealth.net.au/tplt/svc/GetTemplate/1.1/GetTem
platePortType/getTemplateRequest"/>
         <wsdl:output message="pcehr:getTemplateOutMsg"
wsam:Action="http://ns.electronichealth.net.au/tplt/svc/GetTemplate/1.1/GetTem
platePortType/getTemplateResponse"/>
         <wsdl:fault name="standardError" message="pcehr:standardErrorMsg"
wsam:Action="http://ns.electronichealth.net.au/tplt/svc/GetTemplate/1.1/GetTem
platePortType/Fault/standardError"/>
      </wsdl:operation>
   </wsdl:portType>
```

```
</wsdl:definitions>
```

A.2.1.2 B2B_SearchTemplateInterface.wsdl

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Version 1.1 issued 11 April 2012-->
<wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:pcehr="http://ns.electronichealth.net.au/tplt/svc/SearchTemplate/1.1"
xmlns:pcehr_b2b
="http://ns.electronichealth.net.au/pcehr/xsd/common/CommonCoreElements/1.0"
xmlns:ns2="http://ns.electronichealth.net.au/tplt/xsd/interfaces/SearchTemplat
e/1.0"
xmlns:tns="http://ns.electronichealth.net.au/wsp/xsd/StandardError/2010"
xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata"
xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata"
xmlns:ns="http://www.w3.org/2000/09/xmldsig#"
xmlns:ns1="http://ns.electronichealth.net.au/tplt/xsd/common/TemplatesCoreElem
ents/1.0" name="searchTemplate"
targetNamespace="http://ns.electronichealth.net.au/tplt/svc/SearchTemplate/1.1
">
   <wsdl:types>
      <xsd:schema
targetNamespace="http://ns.electronichealth.net.au/tplt/svc/SearchTemplate/1.1
" elementFormDefault="qualified">
         <xsd:import
namespace="http://ns.electronichealth.net.au/pcehr/xsd/common/CommonCoreElemen
ts/1.0" schemaLocation="../../schema/Common/PCEHR_CommonTypes.xsd"/>
         <xsd:import
namespace="http://ns.electronichealth.net.au/tplt/xsd/interfaces/SearchTemplat
e/1.0" schemaLocation="../../schema/External/PCEHR_SearchTemplate.xsd"/>
         <xsd:import
namespace="http://ns.electronichealth.net.au/wsp/xsd/StandardError/2010"
schemaLocation="../../schema/Common/wsp-StandardError-2010.xsd"/>
      </xsd:schema>
   </wsdl:types>
   <wsdl:message name="searchTemplateInMsg">
   <wsdl:part name="timestampHeader" element="pcehr_b2b:timestamp"/>
      <wsdl:part name="signatureHeader" element="pcehr_b2b:signature"/>
      <wsdl:part name="PCEHRHeader" element="pcehr_b2b:PCEHRHeader"/>
      <wsdl:part name="parameters" element="ns2:searchTemplate"/>
   </wsdl:message>
   <wsdl:message name="searchTemplateOutMsg">
<wsdl:part name="signatureHeader" element="pcehr_b2b:signature"/>
      <wsdl:part name="parameters" element="ns2:searchTemplateResponse"/>
   </wsdl:message>
   <wsdl:message name="standardErrorMsg">
      <wsdl:part name="parameters" element="tns:standardError"/>
   </wsdl:message>
   <wsdl:portType name="SearchTemplatePortType">
      <wsdl:operation name="searchTemplate" parameterOrder="parameters">
         <wsdl:input message="pcehr:searchTemplateInMsg"
wsam:Action="http://ns.electronichealth.net.au/tplt/svc/SearchTemplate/1.1/Sea
rchTemplatePortType/searchTemplateRequest"/>
         <wsdl:output message="pcehr:searchTemplateOutMsg"
wsam:Action="http://ns.electronichealth.net.au/tplt/svc/SearchTemplate/1.1/Sea
rchTemplatePortType/searchTemplateResponse"/>
         <wsdl:fault name="standardError" message="pcehr:standardErrorMsg"
wsam:Action="http://ns.electronichealth.net.au/tplt/svc/SearchTemplate/1.1/Sea
rchTemplatePortType/Fault/standardError"/>
```

```
      </wsdl:operation>
   </wsdl:portType>
</wsdl:definitions>
```

A.2.2 TLS Binding

The following WSDL specifications define the bindings of the Template Retrieval interfaces based on the TLS Security Profile defined in ATS 5820-2010.

A.2.2.1 B2B_GetTemplate.wsdl

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Version 1.1 issued 11 April 2012--><wsdl:definitions
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:pcehr="http://ns.electronichealth.net.au/tplt/svc/GetTemplate/1.1"
xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata"
xmlns:wsp="http://www.w3.org/ns/ws-policy" xmlns:sp="http://docs.oasis-
open.org/ws-sx/ws-securitypolicy/200702" name="getTemplate"
targetNamespace="http://ns.electronichealth.net.au/tplt/svc/GetTemplate/1.1">
   <wsdl:import
namespace="http://ns.electronichealth.net.au/tplt/svc/GetTemplate/1.1"
location="B2B_GetTemplateInterface.wsdl"/>
   <wsdl:binding name="getTemplateServiceSOAP12Binding"
type="pcehr:GetTemplatePortType">
      <soap12:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/>
      <wsp:PolicyReference URI="#AddressingPolicy"/>
      <wsp:PolicyReference URI="#TlsPolicy"/>
      <wsdl:operation name="getTemplate">
         <soap12:operation soapActionRequired="false" style="document"/>
         <wsdl:input>
<soap12:header message="pcehr:getTemplateInMsg" part="timestampHeader"
use="literal"/>
            <soap12:header message="pcehr:getTemplateInMsg"
part="signatureHeader" use="literal"/>
            <soap12:header message="pcehr:getTemplateInMsg" part="PCEHRHeader"
use="literal"/>
            <soap12:body parts="parameters" use="literal"/>
         </wsdl:input>
         <wsdl:output>
            <soap12:header message="pcehr:getTemplateOutMsg"
part="signatureHeader" use="literal"/>
            <soap12:body parts="parameters" use="literal"/>
         </wsdl:output>
         <wsdl:fault name="standardError">
            <soap12:fault name="standardError" use="literal"/>
         </wsdl:fault>
      </wsdl:operation>
   </wsdl:binding>
   <wsdl:service name="getTemplateService">
      <wsdl:port name="getTemplateSOAP12Port"
binding="pcehr:getTemplateServiceSOAP12Binding">
         <soap12:address location="http://localhost:8011/"/>
      </wsdl:port>
   </wsdl:service>
   <wsp:Policy xml:id="AddressingPolicy">
```

```
      <wsam:Addressing/>
   </wsp:Policy>
   <wsp:Policy xml:id="TlsPolicy">
      <sp:TransportBinding>
         <wsp:Policy>
            <sp:TransportToken>
               <wsp:Policy>
                  <sp:HttpsToken>
                     <wsp:Policy>
                        <sp:RequireClientCertificate/>
                     </wsp:Policy>
                  </sp:HttpsToken>
               </wsp:Policy>
            </sp:TransportToken>
         </wsp:Policy>
      </sp:TransportBinding>
   </wsp:Policy>
</wsdl:definitions>
```

B2B_GetTemplate.wsdl

A.2.2.2 B2B_SearchTemplate.wsdl

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Version 1.1 issued 11 April 2012--><wsdl:definitions
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:pcehr="http://ns.electronichealth.net.au/tplt/svc/SearchTemplate/1.1"
xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata"
xmlns:sp="http://docs.oasis-open.org/ws-sx/ws-securitypolicy/200702"
xmlns:wsp="http://www.w3.org/ns/ws-policy" name="searchTemplate"
targetNamespace="http://ns.electronichealth.net.au/tplt/svc/SearchTemplate/1.1
">
   <wsdl:import
namespace="http://ns.electronichealth.net.au/tplt/svc/SearchTemplate/1.1"
location="B2B_SearchTemplateInterface.wsdl"/>
   <wsdl:binding name="searchTemplateServiceSOAP12Binding"
type="pcehr:SearchTemplatePortType">
      <soap12:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/>
      <wsp:PolicyReference URI="#AddressingPolicy"/>
      <wsp:PolicyReference URI="#TlsPolicy"/>
      <wsdl:operation name="searchTemplate">
         <soap12:operation soapActionRequired="false" style="document"/>
         <wsdl:input>
   <soap12:header message="pcehr:searchTemplateInMsg" part="timestampHeader"
use="literal"/>
            <soap12:header message="pcehr:searchTemplateInMsg"
part="signatureHeader" use="literal"/>
            <soap12:header message="pcehr:searchTemplateInMsg"
part="PCEHRHeader" use="literal"/>
            <soap12:body parts="parameters" use="literal"/>
         </wsdl:input>
         <wsdl:output>
<soap12:header message="pcehr:searchTemplateOutMsg" part="signatureHeader"
use="literal"/>
            <soap12:body parts="parameters" use="literal"/>
         </wsdl:output>
         <wsdl:fault name="standardError">
```

```
            <soap12:fault name="standardError" use="literal"/>
         </wsdl:fault>
      </wsdl:operation>
   </wsdl:binding>
   <wsdl:service name="searchTemplateService">
      <wsdl:port name="searchTemplateSOAP12Port"
binding="pcehr:searchTemplateServiceSOAP12Binding">
         <soap12:address location="http://localhost:8011/"/>
      </wsdl:port>
   </wsdl:service>
   <wsp:Policy xml:id="AddressingPolicy">
      <wsam:Addressing/>
   </wsp:Policy>
   <wsp:Policy xml:id="TlsPolicy">
      <sp:TransportBinding>
         <wsp:Policy>
            <sp:TransportToken>
               <wsp:Policy>
                  <sp:HttpsToken>
                     <wsp:Policy>
                        <sp:RequireClientCertificate/>
                     </wsp:Policy>
                  </sp:HttpsToken>
               </wsp:Policy>
            </sp:TransportToken>
         </wsp:Policy>
      </sp:TransportBinding>
   </wsp:Policy>
</wsdl:definitions>
```

B2B_SearchTemplate
.wsdl

Appendix B Acronyms and Terminology

B.1 Acronyms

Acronym	Explanation
PCEHR	Personally Controlled Electronic Health Record
WSDL	Web Service Definition Language
WSP	Web Service Profile – Commonly used to refer to the ATS-5820 Web Service Profile.
XSD	XML Schema Definition

B.2 Specialised Terminology

Term	Explanation
NASH certificate	A NASH certificate is a digital certificate that is compliant with the NASH certificate policies.
Service	A Service encapsulates the collaboration which occurs between two or more parties to achieve a goal. Each participant in the service may offer multiple Service Interfaces.
Service Interface	A Service Interface is a logical grouping of operations which be offered by a participant within the context of a Service.
Service Operation	A Service Operation is a specific function which supports communication between two participants.

Appendix C References

Tag	Name	Version Release Date
[ATS 5820-2010]	ATS 5820-2010 Australian Technical Specification E-health Web Services Profile	2010
[ATS 5821-2010]	ATS 5821-2010 Australian Technical Specification E-health XML secured payload profiles	2010
[LSS2011]	Template Service Interface Logical Service Specification	1.0 December 2011
[PK2007]	APPNOTE.TXT - .ZIP File Format Specification http://www.pkware.com/documents/casestudies/APPNOTE.TXT	Version 6.3.2 Release 28 September 2007
[PKGSPEC]	Template Package Specification	1.0 April 2012
[RFC2119]	IETF, *RFC 2119: Keywords for use in RFCs to Indicate Requirement Levels*, S. Bradner http://ietf.org/rfc/rfc2119.txt	March 1997
[W3C-XML-1.1]	W3C Recommendation Extensible Markup Language (XML) 1.1 (Second Edition) http://www.w3.org/TR/2006/REC-xml11-20060816	1.1 29 Sepetember 2006