nehta

# Technical Service Specification

## Electronic Transfer of Prescription 1.1

Version 1.1 — 17 December 2010

Draft for Trial Implementation

National E-Health Transition Authority

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

**Disclaimer**

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

**Document Control**

This document is maintained in electronic form. The current revision of this document is located on the NEHTA Web site and is uncontrolled in printed form. It is the responsibility of the user to verify that this copy is of the latest revision.

# Document information

## Change history

| Version | Date | Comments |
|---|---|---|
| 1.1 | 17 December 2010 | Draft for Trial Implementation |

## Document authorisation

| Name | Title | Signature |
|---|---|---|
| Stephen Johnston | Head of Product and Solutions Development |  |
| Dr Leonie Katekar | Director, Clinical Unit and Clinical Lead |  |
| Toby Mathieson | Program Manager,<br>e-Medication Management |  |

# Table of contents

This page is intentionally left blank

# Preface

## Document purpose

This document defines an implementable, platform-dependent technical services specification for Electronic Transfer of Prescriptions package.

## Intended audience

This document is intended for a technical audience, including (but not limited to) representatives of:

- Commonwealth, state, and territory health departments, and Medicare Australia
- The Australian healthcare standards development community
- Organisations that produce software products used to generate prescriptions
- Organisations that produce software products used by community pharmacies to dispense medications
- Organisations that operate (or plan to operate) services that transfer electronic prescriptions

This is a technical document defining web services and associated XML schemas based on UML specifications. Familiarity with web services definition language (WSDL), XML and UML 2.3 standards is assumed [UML2010].

## Document map

The following diagram represents the relationship between this document and others within the Electronic Transfer of Prescriptions Release 1.1 Package.

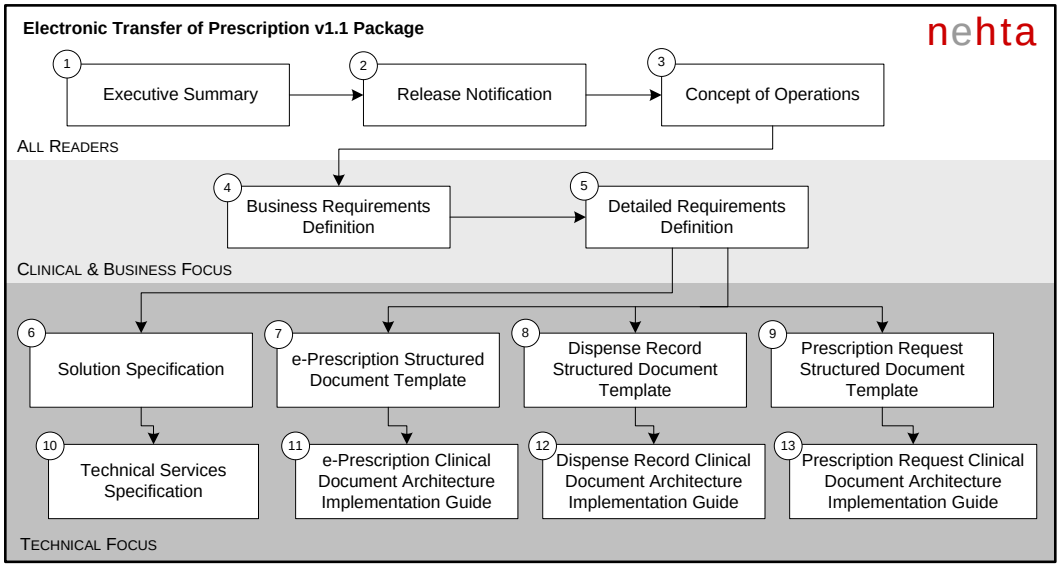


**Figure 1 ETP Draft Release 1.1 package**

## Definitions, acronyms and abbreviations

For lists of definitions, acronyms and abbreviations, see the Definitions section at the end of the document.

## References and related documents

For a list of all referenced documents, see the References section at the end of this document.

# 1 Introduction

## 1.1 Introduction

A platform independent specification of the technical services for the Electronic Transfer of Prescriptions is presented in the ETP Release 1.1 Solution Specification [ETP-SS]. The ETP solution specification does not mandate particular technologies as the basis for implementation and as such is not implementable.

This technical service specification builds upon the solution specification, identifying specific technologies for elements of the solution specification sufficient for implementation of interoperable E-Prescribing, E-Dispensing, Prescription Exchange, Facility Based Supply Manager and Last Supply Notification Agent systems.

## 1.2 Technology Platform

The technology platform for this technical service specification is comprised of:

- Interaction through web service interfaces that conform to identified elements of the E-Health Web Services Profile [ATS 5820—2010]
- Digital signing and encryption of clinical documents using X.509 certificates according to the E-Health XML Secured Payload Profile [ATS 5821—2010]
- Concrete representation of the Prescription [ETP-EP_SDT2010], Dispense Record [ETP-DR_SDT2010] and Prescription Request [ETP-PR_SDT2010] documents using HL7 v3 CDA according to [ETP-EP_CDAIG2010], [ETP-DR_CDAIG2010] and [ETP-PR_CDAIG2010] respectively
- Representation of Document Access Keys (DAKs) as an 18-character string comprising a 3-character prescription exchange (PES) identifier and a 15-character random string
- Re-use of E-Health Secure Message Delivery interfaces [ATS 5822—2010] where appropriate[1].

As previously defined in the solution specification, this specification depends upon the following infrastructure services:

- Healthcare Identification (HI) services for identification of healthcare provider organisations (HPI-O), healthcare provider individuals (HPI-I) and the subject of care (an individual identified by an IHI)
- The National Authentication Service for Health (NASH) for the provision of X.509 certificates used for signing and encryption
- The National Clinical Terminology and Information Service (NCTIS) for definition of codes used in prescription documents.

[1] For efficiency and simplicity, custom interfaces are defined for E-Prescribing and E-Dispensing invocations on the prescription exchange. An alternate implementation of these operations using the Sealed Immediate Message Delivery interface of [ATS 5820—2010] will be defined separately.

## 1.3 Overview

A set of common behaviours is defined in Section 2.

Sections 3 through 6 define conformance points for each role, including the implementation and use of the various service interfaces.

Appendix A defines XML schemas for each of the message types.

Appendix B defines WSDL for interfaces implemented by the roles, except for those already defined in [ATS 5820—2010].

# 2 Common Behaviours

## 2.1 Web Service Profile

The following conformance points define the application of the e-Health web services profile [ATS 5820—2010] to service interactions:

*ETPTSS 1* All implementations shall conform to the Web Services Base Profile from the Standards Australia e-Health Web Services Profiles [ATS 5820—2010] for all web service invocations.

*ETPTSS 2* All implementations shall implement the TLS Security Profile from the Standards Australia e-Health Web Services Profiles [ATS 5820—2010] for all web service invocations.

## 2.2 Document Access Keys

A set of abstract conformance points relating to the creation and use of document access keys (DAKs) is provided in chapter 4 of the ETP Solution Specification [ETP-SS2010]. The following conformance points apply to the implementation of those conformance points:

### 2.2.1 Platform Independent Compliance

*ETPTSS 3* All implementations shall conform to the applicable platform independent DAK conformance points specified in clause 4.4.1 of the ETP Solution Specification.

### 2.2.2 Creation

The creator of a DAK is subject to the following conformance points. At present, only the E-Prescribing System will create DAKs.

*ETPTSS 4* The DAK creator shall create an 18-character DAK string.

*ETPTSS 5* The DAK creator shall only use characters from the following set in the DAK string:

```
23456789ABCDEFGHJKLMNPQRSTUVWXYZabcdefghjkmnpqrstuvwxyz=@#/
```

*ETPTSS 6* The DAK creator shall identify the prescription exchange to which the document will be submitted in the first three characters of the DAK string. Note the implication that if a document is submitted to more than one prescription exchange, it will necessarily have a different DAK in each prescription exchange.

*ETPTSS 7* The DAK creator shall set the remaining 15 characters by choosing randomly from the character set identified in ETPTSS 5. The DAK creator shall ensure that there is no bias in the selection of characters from the set. Note that selecting characters using modulo 59 of a random 8-bit number introduces a bias, since 59 is not a whole factor of 256.

### 2.2.3 DAK Usage

Both creators and recipients of a DAK (referred to as DAK users) are subject to the following conformance points in their usage of DAKs.

*ETPTSS 8* A DAK user shall derive the retrieval key (RK) and cipher key (CK) using the AES-CMAC-PRF-128 algorithm as described in [RFC4615] as the pseudo-random function (*prf*).

*ETPTSS 9* A DAK user shall apply the key generation sequence defined in Section 2.13 of [RFC4306] with the retrieval key corresponding to T1, the cipher key corresponding to T2 concatenated with T3 and the string 'S' set to the literal value "NEHTA-DAK-V01". Thus:

```
RK = prf (DAK, "NEHTA-DAK-V01" | 0x1)
CK1 = prf (DAK, RK | "NEHTA-DAK-V01" | 0x2)
CK2 = prf (DAK, CK1 | "NEHTA-DAK-V01" | 0x3)
CK = CK1 | CK2
```

Where "|" indicates concatenation.

*ETPTSS 10* A DAK user shall use the first three characters of the DAK as the prescription exchange identifier in any Qualified Retrieval Key structure used to identify prescriptions and dispense documents associated with the DAK.

### 2.2.4 Encryption and Decryption

The following conformance points relate to the encryption and decryption of documents using a cipher key (*CK*) derived from a DAK.

*ETPTSS 11* The creator of an E-Dispense Document, E-Prescription Document or Prescription Request Document shall create an Encrypted Payload Container according to Section 3.3.1 of [ATS 5821—2010], with the session key described in Section 5.2.2.2 of [ATS 5821—2010] set to the 256 bit Cipher Key derived from the DAK as defined in Section 2.2.3.

*ETPTSS 12* The creator of an E-Dispense Document, E-Prescription Document or Prescription Request Document shall include an `EncryptedKey` element populated as defined Section 3.3.1 of [ATS 5821—2010] and encrypted using the public key associated with their own organisational X.509 certificate.

*ETPTSS 13* The creator of an E-Dispense Document, E-Prescription Document or Prescription Request Document may include one or more additional `EncryptedKey` elements, each encrypted with the public key of a party authorised to view the enclosed clinical document.

*ETPTSS 14* A party with a DAK for an E-Dispense Document, E-Prescription Document or Prescription Request Document may decrypt the encrypted payload using the algorithms identified in [ATS 5821—2010] and using the 256 bit Cipher Key derived from the DAK as defined in Section 2.2.3.

## 2.3 Public Key Infrastructure

The following conformance points relate to the public key infrastructure (PKI) used for authentication and encryptions of ETP interactions.

*ETPTSS 15* All roles must comply with the identity and PKI conformance points specified in clause 4.4.1 of the ETP Solution Specification.

*ETPTSS 16* All roles shall use X.509 certificates containing a valid HPI-O identifier provided by NASH for:

- Securing TLS connections
- Signing and encrypting payloads according to [ATS 5821—2010]
- Securing message-based communications using the mechanisms defined in [ATS 5822—2010].

## 2.4 Identification

Identification of various elements in the ETP specifications requires unique identifiers across the community. In the following conformance points, the term “qualified identifier” refers to an identifier that satisfies the conformance points of clause 2.6.1 of [ATS 5822—2010].

### 2.4.1 Provider and Individual Identifiers

The following conformance points apply to the representation of identifiers for providers and subjects of care.

*ETPTSS 17* Except for the three-character identifiers contained in DAKs, the creator of an event, request, response, notification or fault shall ensure that provider organisations, provider individuals, and subjects of care are identified using a qualified identifier for their HPI-O, HPI-I or IHI respectively. The scheme for these qualified identifiers is defined in the HI Services specification [HI_SERV_SPEC].

### 2.4.2 Event, Request, Notification, Document and Clinical Process Identifiers

All events, requests, notifications and documents created for use with the ETP technical service specification have explicit identifiers. Clinical process instances associated with these interactions also have explicit identifiers. The following conformance points apply to a system creating these identifiers (the *originator)*.

*ETPTSS 18* The originator shall ensure that event, request, notification and document identifiers are unique across the community of e-Health participants.

*ETPTSS 19* The originator should create event, request, notification and document identifiers as either:

a) A UUID encoded as a URN according to [RFC4122]; or

b) A URL based on a domain name registered by the creating organisation, and an identifier that is unique at the creating organisation.

### 2.4.3 Document Type Identifiers

Each clinical document transmitted in ETP interactions has a document type identifier in the associated metadata which indicates the type of the document (i.e. either an electronic prescription or an electronic dispense record).

*ETPTSS 20* The creator of a clinical document shall encode the document type identifier using a qualified identifier using the base URL: http://ns.electronichealth.net.au/etp/doctype/1.1/

And concatenating an identifier of “prescription” for electronic prescriptions or “dispense” for electronic dispense records.

### 2.4.4 Fault Codes

Fault codes included in faults need to be distinguishable across many clinical applications. The following conformance point applies to fault codes used in any ETP fault indication.

*ETPTSS 21* The originator of a fault shall encode an ETP fault code in any error messages or web service faults using a qualified identifier using the base URL:

`http://ns.electronichealth.net.au/etp/codes/`

And concatenating a fault type name from Section 4.4.2 of [ETP_SS2010].

## 2.5 Date and Time Values

*ETPTSS 22* All xsd:dateTime elements defined in the schemas of this specification shall be transmitted as timezoned values in the UTC timezone.

## 2.6 Transporting Notifications Using SMD

This specification uses the transport mechanisms defined in the Standards Australia E-Health Secure Message Delivery specification [ATS 5822—2010] (SMD) to send ETP Notifications and receive corresponding ETP acknowledgements. This section defines the sending and receipt of ETP Notifications using SMD.

### 2.6.1 Sending Notifications

The following conformance points apply to the ETP Notification Sender.

*ETPTSS 23* The Sender shall use the invocationId field of the outgoing sealed message metadata (clause 7.3.2 of ATS 5822—2010) for the ETP NotificationId (Section 3.4.1 of the ETP-SS2010).

*ETPTSS 24* The Sender shall use the creationTime field of the outgoing sealed message metadata for the ETP NotificationTimestamp (Section 3.4.1 of the ETP-SS2010).

*ETPTSS 25* The Sender shall use the senderOrganisation field of the outgoing sealed message metadata for the ETP Organisation (Section 3.4.1 of the ETP-SS2010).

#### 2.6.1.1 Deferred Mode Sending

The following conformance points apply to the delivery of ETP Notifications using the deferred mode mechanisms defined in [ATS 5822—2010] (SMD).

*ETPTSS 26* The Sender shall deliver the Notification to an identified Receiver as the clinical payload of a deliver operation of the Sealed Message Delivery interface defined in SMD. As per the SMD specification this delivery may pass through one or more Intermediaries before reaching the identified Receiver. Note the implication that the notification is encrypted and addressed using the mechanisms defined in [ATS 5822—2010].

*ETPTSS 27* The Sender shall accept a final Transport Response as defined in [ATS 5822—2010] in response to the Notification.

*ETPTSS 28* The Sender shall interpret a successful final Transport Response as an ETP Notification Delivery Ack.

*ETPTSS 29* The Sender shall interpret the transportResponseTime field of the Transport Response metadata (clause 7.3.3 of ATS 5822—2010) as the ETP Notification Delivery Ack ackTimestamp (Section 3.4.2 of the ETP-SS20102010).

*ETPTSS 30* If the response code of an error transport response is an ETP fault code as defined in Section 3.6 of the ETP Solution Specification [ETP-SS2010], the Sender shall behave as defined for that fault in the ETP Solution Specification.

*ETPTSS 31* If the response code of an error transport response is not an ETP fault code, the Sender shall treat the error as a non-retryable communication error and escalate to a responsible person.

#### 2.6.1.2 Immediate Mode Sending

The following conformance points apply to the delivery of ETP Notifications using the Sealed Immediate Message Delivery interface defined in [ATS 5822—2010], if implemented.

*ETPTSS 32* The Sender may deliver the Notification to an identified Receiver as the clinical payload of a deliver operation of the Sealed Immediate Message Delivery interface defined in SMD.

*ETPTSS 33* The Sender shall interpret a successful web service response containing a Sealed Message as an ETP Notification Delivery Ack.

*ETPTSS 34* The Sender shall interpret the creationTime field of the response message metadata (clause 7.3.2 of ATS 5822—2010) as the ETP Notification Delivery Ack ackTimestamp (Section 3.4.2 of the ETP-SS20102010).

*ETPTSS 35* The Sender shall accept web service faults containing an error code as defined for the Sealed Immediate Message Delivery interface in [ATS 5822—2010].

*ETPTSS 36* If the response code contained in a fault is an ETP fault code, the Sender system shall behave as defined for that fault in the ETP Solution Specification.

*ETPTSS 37* If the response code contained in a fault is not an ETP fault code, the Sender should treat the error as a non-retryable communications error and escalate to a responsible person.

### 2.6.2 Receiving Notifications

The following conformance points apply to the ETP Notification Recipient.

*ETPTSS 38* The Receiver shall interpret the invocationId field of the incoming sealed message metadata as the ETP NotificationId (Section 3.4.1 of the ETP-SS2010).

*ETPTSS 39* The Receiver shall interpret the creationTime field of the incoming sealed message metadata as the ETP NotificationTimestamp (Section 3.4.1 of the ETP-SS2010).

*ETPTSS 40* The Receiver shall interpret the senderOrganisation field of the incoming sealed message metadata as the ETP Organisation (Section 3.4.1 of the ETP-SS2010).

#### 2.6.2.1 Deferred Mode Receipt

The following conformance points apply to the receipt of ETP Notifications using the deferred mode mechanisms defined in [ATS 5822—2010] (SMD).

*ETPTSS 41* The Receiver shall receive Notifications as the clinical payload of a Sealed Message either directly through an implementation of the Sealed Message Delivery interface or through retrieval of sealed messages from a Receiver Intermediary as per [ATS 5822—2010].

*ETPTSS 42* The Receiver shall acknowledge each Notification with a final Transport Response indicating success or error.

*ETPTSS 43* The Receiver shall use a success Transport Response to signal an ETP Notification Acknowledgement.

*ETPTSS 44* The Receiver shall use the transportResponseTime field of the Transport Response metadata as the ETP Notification Delivery Ack ackTimestamp (Section 3.4.2 of the ETP-SS20102010).

*ETPTSS 45* If an error condition defined in the fault codes of Section 4.8 of the ETP Solution Specification has occurred, the Receiver shall return a final error Transport Response with the response code set to the associated fault code to the Sender.

*ETPTSS 46* If an error condition that is not defined in the fault codes of Section 4.4.2 of the ETP Solution Specification has occurred, the Sender shall return an final error Transport Response containing a response code as defined in [ATS 5822—2010] to the Sender.

#### 2.6.2.2 Immediate Mode Receipt

The following conformance points apply to the receipt of ETP Notifications using the Sealed Immediate Message Delivery interface defined in [ATS 5822—2010], if implemented.

*ETPTSS 47* The Receiver may receive Notifications as the clinical payload of the deliver operation of its Sealed Immediate Message Delivery interface.

*ETPTSS 48* If the Notification is successfully processed, the Receiver shall return a sealed message with an empty encrypted payload to signal an ETP Notification Acknowledgement.

*ETPTSS 49* When returning successfully, the Receiver shall use the creationTime field of the returned sealed message metadata as the as the ETP Notification Delivery Ack ackTimestamp (Section 3.4.2 of the ETP-SS20102010).

*ETPTSS 50* If an error condition defined in the fault codes of Section 4.2 of the ETP Solution Specification has occurred, the Receiver shall raise a web service fault with the response code field containing the associated ETP fault code.

*ETPTSS 51* If an error condition that is not defined in the fault codes of Section 4.4.2 of the ETP Solution Specification has occurred, the Receiver shall raise a web service fault containing a response code as defined in [ATS 5822—2010].

# 3 E-Prescribing System

The following subsections specify the required behaviour of an E-Prescribing System implementation.

## 3.1 Platform Independent Conformance

*ETPTSS 52* An E-Prescribing System shall conform to the conformance points of an E-Prescribing System defined in Section 4.7 of the ETP Solution Specification [ETP-SS20102010].

## 3.2 Interface Implementation

The following conformance points define the interface providers and invokers that an E-Prescribing system implements.

*ETPTSS 53* An E-Prescribing System shall implement an invoker for the E-Prescribing web service interface defined by the WSDL in Appendix A.3 for invoking the prescribing operations of a Prescription Exchange. This implementation shall conform to the conformance points of an E-Prescribing Interface invoker defined in Section 4.7.5 of the ETP Solution Specification [ETP-SS2010].

*ETPTSS 54* An E-Prescribing System shall implement the Receiver role of the E-Health Secure Message Delivery specification [ATS 5822—2010] to receive Prescription Required Notifications from E-Dispensing Systems. This implementation shall conform to the conformance points of a Prescription Requesting Interface provider defined in Section 4.7.6.1 of the ETP Solution Specification [ETP-SS2010].

*ETPTSS 55* An E-Prescribing System shall implement the Sender role of the E-Health Secure Message Delivery specification [ATS 5822—2010] to send Prescription Available Notifications to Electronic Dispensing Systems. This implementation shall conform to the conformance points of a Prescription Receiving Interface invoker defined in defined in Section 4.7.6.2 of the ETP Solution Specification [ETP-SS2010].

*ETPTSS 56* The E-Prescribing System shall implement the Sender role of the E-Health Secure Message Delivery specification [ATS 5822—2010] to send Electronic Prescription Notifications to Facility Based Supply Managers. This implementation shall conform to the conformance points of a Facility Managed Supply Service invoker defined in Section 4.7.7 of the ETP Solution Specification [ETP-SS2010].

## 3.3 Service Categories

The following conformance points define service categories used by the E-Prescribing system in its implementation.

*ETPTSS 57* When searching for E-Prescribing services in a service directory, the E-Prescribing System should use the service category:

http://ns.electronichealth.net.au/etp/sc/eprescribing/1.1

*ETPTSS 58* When publishing an [ATS 5822—2010] Sender endpoint in a service directory for receipt of Prescription Required Notifications, the E-Prescribing shall use the service category:

http://ns.electronichealth.net.au/etp/sc/prescriptionrequesting/1.1

*ETPTSS 59* When searching for Prescription Receiving interfaces that can receive Prescription Available Notifications in a service directory, the E-Prescribing System should use the service category:

http://ns.electronichealth.net.au/etp/sc/prescriptionreceiving/1.1

*ETPTSS 60* When searching for Facility Managed Supply interfaces that can receive Electronic Prescription Notifications in a service directory, the E-Prescribing System should use the service category:

http://ns.electronichealth.net.au/etp/sc/facilitymanagedsupply/1.1

## 3.4 Invocation

### 3.4.1 E-Prescribing `prescribe` operation

The following conformance points relate to the invocation of the prescribe operation of the E-Prescribing interface.

*ETPTSS 61* The behaviour of the E-Prescribing system in handling requests, responses and faults relating to the prescribe operation shall comply with the prescribe operation conformance points of Sections 4.7.5.1 of the ETP Solution Specification [ETP-SS20102010].

*ETPTSS 62* The E-Prescribing System shall construct a prescription document in HL7 v3 CDA form as defined in [ETP-EP_CDAIG2010].

*ETPTSS 63* The E-Prescribing System shall generate a DAK and associated retrieval and cipher keys as defined in Section 2.2.

*ETPTSS 64* The E-Prescribing System shall construct a Prescription Available Event using the PrescriptionAvailable XML schema defined in Appendix A.2. This Prescription Available Event shall comply with the encryption conformance points of Section 2.2.4.

*ETPTSS 65* The E-Prescribing System shall deliver the Prescription Available Event to the Prescription Exchange using the prescribe operation of the E-Prescribing web service interface as defined in the WSDL of Appendix A.3.

*ETPTSS 66* The E-Prescribing System shall accept a Prescription Available Acknowledgement encoded using the PrescriptionAvailableAck XML schema defined in Appendix A.2 in the response to a successful prescribe operation.

*ETPTSS 67* The E-Prescribing System shall accept a web service fault encoded as defined for the prescribe operation in the web service interface WSDL of Appendix A.3 in response to a failed prescribe operation.

### 3.4.2 E-Prescribing `retrievePrescription` operation

The following conformance points relate to the invocation of the retrievePrescription operation of the E-Prescribing interface.

*ETPTSS 68* The behaviour of the E-Prescribing system in handling requests, responses and faults relating to the retrievePrescription operation shall conform to the retrievePrescription operation conformance points of Sections 4.7.5.2 of the ETP Solution Specification [ETP-SS2010].

*ETPTSS 69* The E-Prescribing System shall construct a Retrieve Prescription Request using the RetrievePrescriptionRequest XML schema defined in Appendix A.2.

*ETPTSS 70* The E-Prescribing System shall deliver the Retrieve Prescription Request to the Prescription Exchange using the retrievePrescription operation of the E-Prescribing web service interface as defined in the WSDL of Appendix A.3.

*ETPTSS 71* The E-Prescribing System shall accept a Retrieve Prescription Response encoded using the RetrievePrescriptionResponse XML schema defined in Appendix A.2 in response to a successful retrievePrescription operation.

*ETPTSS 72* The E-Prescribing System shall accept a web service fault encoded as defined in the web service interface WSDL of Appendix A.3 in response to a failed retrievePrescription operation.

### 3.4.3 E-Prescribing `cancelPrescription` operation

The following conformance points relate to the invocation of the cancelPrescription operation of the E-Prescribing interface.

*ETPTSS 73* The behaviour of the E-Prescribing system in handling requests, responses and faults relating to the cancelPrescription operation shall conform to the invoker conformance points of Sections 4.7.5.3 of the ETP Solution Specification [ETP-SS2010].

*ETPTSS 74* The E-Prescribing System shall construct a Prescription Cancelled Event using the PrescriptionCancelled XML schema defined in Appendix A.2.

*ETPTSS 75* The E-Prescribing System shall deliver the Prescription Cancelled Event to the Prescription Exchange using the cancelPrescription operation of the E-Prescribing web service interface as defined in the WSDL of Appendix A.3.

*ETPTSS 76* The E-Prescribing System shall accept a Prescription Cancelled Ack encoded using the PrescriptionCancelledAck XML schema defined in Appendix A.2 in response to a successful cancelPrescription operation.

*ETPTSS 77* The E-Prescribing System shall accept a web service fault encoded as defined in the web service interface WSDL of Appendix A.3 in response to a failed cancelPrescription operation.

### 3.4.4 Prescription Required Interface Provider

The following conformance points relate to the receipt of a Prescription Required Notification from an E-Dispensing system.

*ETPTSS 78* The behaviour of the E-Prescribing system in the receipt and handling of a Prescription Required Notification shall conform to the conformance points of Section 4.7.6.1 of the ETP Solution Specification.

*ETPTSS 79* The E-Prescribing System shall receive Prescription Required Notifications encoded according to the PrescriptionOwingNotification XML schema defined in Appendix A.1.

*ETPTSS 80* The E-Prescribing System shall receive Prescription Required Notifications through its implementation of the Receiver role of [ATS 5822—2010] as defined in Section 2.6.2.

### 3.4.5 Prescription Receiving Interface Invoker

The following conformance points relate to the delivery of a Prescription Available Notification to the Prescription Receiving interface of an E-Dispensing system by an E-Prescribing system.

*ETPTSS 81* The behaviour of the E-Prescribing system in the delivery of the prescriptionAvailable notification to an E-Dispensing System shall conform to the conformance points of Section 4.7.6.2 of the ETP Solution Specification [ETP-SS2010].

*ETPTSS 82* The E-Prescribing System shall construct a Prescription Available Notification using the PrescriptionAvailableNotification XML schema defined in Appendix A.1.

*ETPTSS 83* The E-Prescribing System shall send Prescription Available Notifications to a Prescription Receiving interface through its implementation of the Sender role of [ATS 5822—2010] as defined in Section 2.6.2.

### 3.4.6 Facility Managed Supply Interface Invoker

The following conformance points relate to the delivery of an Electronic Prescription Notification to the Facility Managed Supply interface of a Facility Based Supply Manager by an E-Prescribing system.

*ETPTSS 84* The behaviour of the E-Prescribing system in the delivery of the Electronic Prescription Notification to a Facility Based Supply Manager shall conform to the conformance points of Section 4.7.7 of the ETP Solution Specification [ETP-SS2010].

*ETPTSS 85* The E-Prescribing System shall construct an Electronic Prescription Notification using the ElectronicPrescriptionNotification XML schema defined in Appendix A.1.

*ETPTSS 86* The E-Prescribing System shall send each Electronic Prescription Notification to a Facility Managed Supply interface through its implementation of the Sender role of [ATS 5822—2010] as defined in Section 2.6.2.

# 4 E-Dispensing System

The following subsections specify the required behaviour of an E-Dispensing System implementation.

## 4.1 Platform Independent Conformance

*ETPTSS 87* An E-Dispensing System shall conform to the conformance points of an E-Dispensing System defined in Section 4.8 of the ETP Solution Specification [ETP-SS20102010].

## 4.2 Interface Implementation

The following conformance points define the interface providers and invokers that an E-Dispensing system implements.

*ETPTSS 88* An E-Dispensing System shall implement an invoker for the E-Dispensing web service interface defined by the WSDL in Appendix A.4 for invoking the dispensing operations of a Prescription Exchange. This implementation shall conform to the conformance points of an E-Dispensing Interface invoker defined in Section 4.8.5 of the ETP Solution Specification [ETP-SS2010].

*ETPTSS 89* An E-Dispensing System shall implement the Receiver role of the E-Health Secure Message Delivery specification [ATS 5822—2010] for the receipt of Prescription Available Notifications from E-Prescribing Systems. This implementation shall conform to the conformance points of a Prescription Receiving Interface provider defined in Section 4.8.4.1 of the ETP Solution Specification [ETP-SS2010].

*ETPTSS 90* The E-Dispensing System shall implement the Sender role of the E-Health Secure Message Delivery specification [ATS 5822—2010] for sending Prescription Required Notifications to an E-Prescribing System. This implementation shall conform to the conformance points of a Prescription Requesting Interface invoker defined in Section 4.8.4.2 of the ETP Solution Specification [ETP-SS2010].

*ETPTSS 91* The E-Dispensing System shall implement the Sender role of the E-Health Secure Message Delivery specification [ATS 5822—2010] for sending Last Supply Notifications to Last Supply Notification Agents. This implementation shall conform to the conformance points of a Last Supply Notification interface invoker defined in Section 4.8.7 of the ETP Solution Specification [ETP-SS2010].

*ETPTSS 92* An E-Dispensing System may implement the Receiver role of the E-Health Secure Message Delivery specification [ATS 5822—2010] for the receipt of Prescription Available Notifications from Facility Based Supply Managers. This implementation shall conform to the conformance points of a Contract Dispensing interface provider defined in Section 4.8.6 of the ETP Solution Specification [ETP-SS2010].

## 4.3 Service Categories

The following conformance points define service categories used by the E-Prescribing system in its implementation.

*ETPTSS 93* When searching for compatible Prescription Exchange endpoints in a service directory, the E-Dispensing System should use the service category:

http://ns.electronichealth.net.au/etp/sc/edispensing/1.1

*ETPTSS 94* When searching for E-Prescribing System endpoints that can receive compatible Prescription Required Notifications, the E-Dispensing System should use the service category:

http://ns.electronichealth.net.au/etp/sc/prescriptionrequesting/1.1

*ETPTSS 95* When publishing a Prescription Receiving endpoint in a service directory for receipt of Prescription Available Notifications from an E-Dispensing System, the E-Dispensing System shall use the service category:

http://ns.electronichealth.net.au/etp/sc/prescriptionreceiving/1.1

*ETPTSS 96* When searching for Last Supply Notification Agent endpoints that can receive compatible Dispense Completion Notifications in a service directory, the E-Dispensing System should use the service category:

http://ns.electronichealth.net.au/etp/sc/lastsupplynotification/1.1

*ETPTSS 97* When publishing a Contract Dispensing endpoint in a service directory for receipt of Prescription Available Notifications from a Facility Based Supply Manager, the E-Dispensing System shall use the service category:

http://ns.electronichealth.net.au/etp/sc/contractdispensing/1.1

## 4.4 Invocation

### 4.4.1 E-Dispensing `retrievePrescription` operation

The following conformance points relate to the invocation of the retrievePrescription operation of the E-Dispensing interface.

*ETPTSS 98* The behaviour of the E-Dispensing system in handling requests, responses and faults relating to the retrievePrescription operation shall comply with the invoker conformance points of Section 4.8.5 of the ETP Solution Specification [ETP-SS2010].

*ETPTSS 99* The E-Dispensing System shall construct a Retrieve Prescription Request using the RetrievePrescriptionRequest XML schema defined in Appendix A.2.

*ETPTSS 100* The E-Dispensing System shall deliver the Retrieve Prescription Request to the Prescription Exchange using the retrievePrescription operation of the E-Dispensing web service interface defined in Appendix A.4.

*ETPTSS 101* The E-Dispensing System shall accept a Retrieve Prescription Response encoded using the RetrievePrescriptionResponse XML schema defined in Appendix A.2 in response to a successful retrievePrescription operation.

*ETPTSS 102* The E-Dispensing System shall accept a web service fault encoded as defined in the web service interface WSDL of Appendix A.4 in response to a failed retrievePrescription operation.

### 4.4.2 E-Dispensing `initiateDispense` operation

The following conformance points relate to the invocation of the initiateDispense operation of the E-Dispensing interface.

*ETPTSS 103* The behaviour of the E-Dispensing system in handling requests, responses and faults relating to the initiateDispense operation shall comply with the invoker conformance points of Section 4.8.5 of the ETP Solution Specification [ETP-SS2010].

*ETPTSS 104* The E-Dispensing System shall construct a Dispense Initiated Event using the DispenseInitiated XML schema defined in Appendix A.2.

*ETPTSS 105* The E-Dispensing System shall deliver the Dispense Initiated Event to the Prescription Exchange using the initiateDispense operation of the E-Dispensing web service interface as defined in the WSDL of Appendix A.4.

*ETPTSS 106* The E-Dispensing System shall accept a Dispense Initiated Acknowledgement encoded using the DispenseInitiatedAck XML schema defined in Appendix A.2 in the response to a successful initiateDispense operation.

*ETPTSS 107* The E-Dispensing System shall accept a web service fault encoded as defined for the initiateDispense operation in the web service interface WSDL of Appendix A.4 in response to a failed initiateDispense operation.

### 4.4.3 E-Dispensing `dispenseCompleted` operation

The following conformance points relate to the invocation of the dispenseCompleted operation of the E-Dispensing interface.

*ETPTSS 108* The behaviour of the E-Dispensing system in handling requests, responses and faults relating to the dispenseCompleted operation shall comply with the invoker conformance points of Sections 4.8.5 of the ETP Solution Specification [ETP-SS2010].

*ETPTSS 109* The E-Dispensing System shall construct a Dispense Record document in HL7 v3 CDA form as defined in [ETP-DR_CDAIG3010].

*ETPTSS 110* The E-Prescribing System shall use retrieval and cipher keys derived from as defined in Section 2.2 using the DAK associated with the original prescription.

*ETPTSS 111* The E-Dispensing System shall construct a Dispense Completed Event using the DispenseCompleted XML schema defined in Appendix A.2. This Dispense Completed Event shall comply with the encryption conformance points of Section 2.2.4.

*ETPTSS 112* The E-Dispensing System shall deliver the Dispense Completed Event to the Prescription Exchange using the dispenseCompleted operation of the E-Dispensing web service interface as defined in the WSDL of Appendix A.4.

*ETPTSS 113* The E-Dispensing System shall accept a Dispense Completed Acknowledgement encoded using the DispenseCompletedAck XML schema defined in Appendix A.2 in the response to a successful dispenseCompleted operation.

*ETPTSS 114* The E-Dispensing System shall accept a web service fault encoded as defined for the dispenseCompleted operation in the web service interface WSDL of Appendix A.4 in response to a failed dispenseCompleted operation.

### 4.4.4 E-Dispensing `dispenseTerminated` operation

The following conformance points relate to the invocation of the dispenseTerminated operation of the E-Dispensing interface.

*ETPTSS 115* The behaviour of the E-Dispensing system in handling requests, responses and faults relating to the dispenseTerminated operation shall comply with the invoker conformance points of Section 4.8.5 of the ETP Solution Specification [ETP-SS2010].

*ETPTSS 116* The E-Dispensing System shall construct a Dispense Terminated Event using the DispenseTerminated XML schema defined in Appendix A.2.

*ETPTSS 117* The E-Dispensing System shall deliver the Dispense Terminated Event to the Prescription Exchange using the dispenseTerminated operation of the

E-Dispensing web service interface as defined in the WSDL of Appendix A.4.

*ETPTSS 118* The E-Dispensing System shall accept a Dispense Terminated Acknowledgement encoded using the DispenseTerminatedAck XML schema defined in Appendix A.2 in the response to a successful dispenseTerminated operation.

*ETPTSS 119* The E-Dispensing System shall accept a web service fault encoded as defined for the dispenseTerminated operation in the web service interface WSDL of Appendix A.4 in response to a failed dispenseTerminated operation.

### 4.4.5 E-Dispensing `dispenseReversed` operation

The following conformance points relate to the invocation of the dispenseReversed operation of the E-Dispensing interface.

*ETPTSS 120* The behaviour of the E-Dispensing system in handling requests, responses and faults relating to the dispenseReversed operation shall comply with the invoker conformance points of Section 4.8.5 of the ETP Solution Specification [ETP-SS2010].

*ETPTSS 121* The E-Dispensing System shall construct a Dispense Reversed Event using the DispenseReversed XML schema defined in Appendix A.2.

*ETPTSS 122* The E-Dispensing System shall deliver the Dispense Reversed Event to the Prescription Exchange using the dispenseReversed operation of the E-Dispensing web service interface as defined in the WSDL of Appendix A.4.

*ETPTSS 123* The E-Dispensing System shall accept a Dispense Reversed Acknowledgement encoded using the DispenseReversedAck XML schema defined in Appendix A.2 in the response to a successful dispenseReversed operation.

*ETPTSS 124* The E-Dispensing System shall accept a web service fault encoded as defined for the dispenseReversed operation in the web service interface WSDL of Appendix A.4 in response to a failed dispenseReversed operation.

### 4.4.6 E-Dispensing `cancelPrescription` operation

The following conformance points relate to the invocation of the cancelPrescription operation of the E-Dispensing interface.

*ETPTSS 125* The behaviour of the E-Dispensing system in handling requests, responses and faults relating to the cancelPrescription operation shall conform to the invoker conformance points of Sections 4.8.5.5 of the ETP Solution Specification [ETP-SS2010].

*ETPTSS 126* The E-Dispensing System shall construct a Prescription Cancelled Event using the PrescriptionCancelled XML schema defined in Appendix A.2.

*ETPTSS 127* The E-Dispensing System shall deliver the Prescription Cancelled Event to the Prescription Exchange using the cancelPrescription operation of the E-Dispensing web service interface as defined in the WSDL of Appendix A.4.

*ETPTSS 128* The E-Dispensing System shall accept a Prescription Cancelled Ack encoded using the PrescriptionCancelledAck XML schema defined in Appendix A.2 in response to a successful cancelPrescription operation.

*ETPTSS 129* The E-Dispensing System shall accept a web service fault encoded as defined in the web service interface WSDL of Appendix A.4 in response to a failed cancelPrescription operation.

### 4.4.7 Prescription Required Interface Invoker

The following conformance points relate to the delivery of a Prescription Required Notification to E-Prescribing Systems by an E-Dispensing system.

*ETPTSS 130* The behaviour of the E-Dispensing system in the delivery of the Prescription Required Notification shall comply with the Invoker System conformance points of Section 4.8.4.2 of the ETP Solution Specification [ETP-SS2010].

*ETPTSS 131* The E-Dispensing System shall construct a Prescription Request document in HL7 v3 CDA form as defined in [ETP-PR_CDAIG2010] and encode it using Base64 encoding for inclusion in a notification.

*ETPTSS 132* The E-Dispensing System shall construct a Prescription Required Notification using the PrescriptionRequiredNotification XML schema defined in Appendix , with the prescriptionRequest field containing the encoded CDA document constructed as defined above in ETPTSS 131. Note that since the Prescription Required Notification is directly addressed to a provider, it is not necessary to create a DAK for this document and the usual encryption mechanisms in [ATS 5822—2010] are used to secure the clinical payload.

*ETPTSS 133* The E-Dispensing System shall send Prescription Owing Notifications as defined in Section 2.6.2 through its implementation of the Sender role of [ATS 5822—2010].

### 4.4.8 Prescription Receiving Interface Provider

The following conformance points relate to the receipt of a Prescription Available Notification from an E-Prescribing System through the implementation of an [ATS 5822—2010] Receiver.

*ETPTSS 134* The behaviour of the E-Dispensing System in the receipt and handling of a Prescription Available Notification shall comply with the Prescription Receiving Interface Provider conformance points of Section 4.8.4.1 of the ETP Solution Specification.

*ETPTSS 135* The E-Dispensing System shall receive and process Prescription Available Notifications encoded according to the PrescriptionAvailableNotification XML schema defined in Appendix A.1.

*ETPTSS 136* The Facility Based Supply Manager System shall receive Prescription Available Notifications through its implementation of the Receiver role of [ATS 5822—2010] as defined in Section 2.6.2.

### 4.4.9 Contracted Dispensing Interface Provider

The following conformance points relate to the receipt of a Prescription Available Notification from a Facility Based Supply Manager through the implementation of an [ATS 5822—2010] Receiver.

*ETPTSS 137* The behaviour of the E-Dispensing System in the receipt and handling of a Prescription Available Notification from a Facility Based Supply Manager shall comply with the Contract Dispensing Interface Provider conformance points of Section 4.8.4.1 of the ETP Solution Specification.

*ETPTSS 138* The E-Dispensing System shall receive and process Prescription Available Notifications encoded according to the PrescriptionAvailableNotification XML schema defined in Appendix A.1.

*ETPTSS 139* The Facility Based Supply Manager System shall receive Prescription Available Notifications through its implementation of the Receiver role of [ATS 5822—2010] as defined in Section 2.6.2.

### 4.4.10 Last Supply Notification Interface Invoker

The following conformance points relate to the delivery of a Last Supply Notification to a Last Supply Notification Agent through the implementation of an [ATS 5822—2010] Sender.

*ETPTSS 140* The behaviour of the E-Dispensing system in the delivery of the Last Supply Notification and handling of shall comply with the Last Supply Notification Interface Invoker conformance points of Section 4.8.7 of the ETP Solution Specification [ETP-SS2010].

*ETPTSS 141* The E-Dispensing System shall construct a Last Supply Notification using the LastSupplyNotification XML schema defined in Appendix A.1.

*ETPTSS 142* The E-Dispensing System shall send Last Supply Notifications through its implementation of the Sender role of [ATS 5822—2010] as defined in Section 2.6.2.

# 5 Facility Based Supply Manager

The following subsections specify the required behaviour of a Facility Based Supply Manager implementation.

## 5.1 Platform Independent Conformance

*ETPTSS 143* A Facility Based Supply Manager shall conform to the conformance points of a Facility Based Supply Manager defined in Section 4.9 of the ETP Solution Specification [ETP-SS20102010].

## 5.2 Interface Implementation

The following conformance points define the interface providers and invokers that a Facility Based Supply Manager system implements.

*ETPTSS 144* A Facility Based Supply Manager shall implement the Receiver role of the E-Health Secure Message Delivery specification [ATS 5822—2010] for the receipt of Electronic Prescription Notifications from E-Prescribing Systems. This implementation shall conform to the conformance points of a Facility Managed Supply interface provider defined in Section 4.9.4 of the ETP Solution Specification [ETP-SS2010].

*ETPTSS 145* A Facility Based Supply Manager shall implement the Sender role of the E-Health Secure Message Delivery specification [ATS 5822—2010] for sending Prescription Available Notifications to an E-Dispensing system that offers a Contract Dispensing Interface. This implementation shall conform to the conformance points of a Contract Dispensing Interface invoker defined in Section 4.9.5 of the ETP Solution Specification [ETP-SS2010].

## 5.3 Service Categories

The following conformance points define service categories used by the E-Prescribing system in its implementation.

*ETPTSS 146* When publishing an [ATS 5822—2010] Receiver endpoint in a service directory for receipt of Electronic Prescription Notifications, the Facility Based Supply Manager shall use the service category:

http://ns.electronichealth.net.au/etp/sc/facilitymanagedsupply/1.1

*ETPTSS 147* When searching for E-Dispensing Systems that can receive Prescription Available Notifications in a service directory, the Facility Based Supply Manager should use the service category:

http://ns.electronichealth.net.au/etp/sc/contractdispensing/1.1

## 5.4 Invocation

### 5.4.1 Facility Managed Supply Interface Provider

The following conformance points relate to the receipt of an Electronic Prescription Notification from an E-Prescribing System.

*ETPTSS 148* The behaviour of the Facility Based Supply Manager in the receipt and handling of an Electronic Prescription Notification shall comply with the Facility Managed Supply Interface Provider conformance points of Section 4.9.4.1 of the ETP Solution Specification.

*ETPTSS 149* The Facility Based Supply Manager shall receive and process Electronic Prescription Notifications encoded according to the ElectronicPrescriptionNotification XML schema defined in Appendix A.1.

*ETPTSS 150* The Facility Based Supply Manager System shall receive Electronic Prescription Notifications through its implementation of the Receiver role of [ATS 5822—2010] as defined in Section 2.6.2.

### 5.4.2 Contract Dispensing Interface Invoker

The following conformance points relate to the sending of Prescription Available Notifications to a Contract Dispensing interface offered by an E-Dispensing System.

*ETPTSS 151* A Facility Based Supply Manager may forward the DAK from a successfully received Electronic Prescription Notification to an authorised E-Dispensing service that offers a Contract Dispensing Interface.

*ETPTSS 152* If forwarding a DAK, the Facility Based Supply Manager shall create a new Sealed Message containing an XML document constructed according to the PrescriptionAvailableNotification XML schema defined in Appendix A.1 as the clinical payload, with the DAK field of that payload containing the DAK from the received Electronic Prescription Notification.

*ETPTSS 153* If forwarding a DAK, the Facility Based Supply Manager shall send the Sealed Message to the Contract Dispensing Interface of an E-Dispensing System through its implementation of the Sender role of [ATS 5822—2010] as defined in Section 2.6.1.

# 6 Last Supply Notification Agent

The following subsections specify the required behaviour of a Last Supply Notification Agent implementation.

## 6.1 Platform Independent Conformance

*ETPTSS 154* A Last Supply Notification Agent shall conform to the conformance points of a Last Supply Notification Agent defined in Section 4.10 of the ETP Solution Specification [ETP-SS20102010].

## 6.2 Interface Implementation

The following conformance points define the interface providers and invokers that a Last Supply Notification Agent system implements.

*ETPTSS 155* A Last Supply Notification Agent shall implement the Receiver role of the E-Health Secure Message Delivery specification [ATS 5822—2010] for the receipt of Last Supply Notifications from E-Prescribing Systems. This implementation shall conform to the conformance points of a Last Supply Notification interface provider defined in Section 4.10.4 of the ETP Solution Specification [ETP-SS2010].

## 6.3 Service Categories

The following conformance points define service categories used by the E-Prescribing system in its implementation.

*ETPTSS 156* When publishing an [ATS 5822—2010] Receiver endpoint in a service directory for receipt of Last Supply Notifications, the Last Supply Notification Agent shall use the service category:

http://ns.electronichealth.net.au/etp/sc/lastsupplynotification/1.1

## 6.4 Invocation

### 6.4.1 Last Supply Notification Provider

The following conformance points relate to the receipt of a Last Supply Notification from an E-Dispensing System.

*ETPTSS 157* The behaviour of the Last Supply Notification Agent in the receipt and handling of a Last Supply Notification shall comply with the Last Supply Notification Provider conformance points of Section 4.10.4 of the ETP Solution Specification.

*ETPTSS 158* The Last Supply Notification Agent shall receive and process Last Supply Notifications encoded according to the DispenseCompletionNotification XML schema defined in Appendix A.1.

*ETPTSS 159* The Last Supply Notification Agent System shall receive Last Supply Notifications through its implementation of the Receiver role of [ATS 5822—2010] as defined in Section 2.6.2.

# 7 Prescription Exchange

The following subsections specify the required behaviour of a Prescription Exchange implementation.

## 7.1 Platform Independent Conformance

*ETPTSS 160* A Prescription Exchange shall conform to the conformance points of a Prescription Exchange defined in Sections 4.5 and 4.6 of the ETP Solution Specification [ETP-SS20102010].

## 7.2 Interface Implementation

The following conformance points define the interface providers and invokers that a Prescription Exchange system implements.

*ETPTSS 161* A Prescription Exchange shall implement a provider for the E-Prescribing web service interface defined in Appendix A.3. This implementation shall conform to the conformance points of an E-Prescribing Interface provider defined in Section 4.5 of the ETP Solution Specification [ETP-SS2010].

*ETPTSS 162* A Prescription Exchange shall implement a provider for the E-Dispensing web service interface defined in Appendix A.4. This implementation shall conform to the conformance points of an E-Dispensing Interface provider defined in Sections 4.5 and 4.6 of the ETP Solution Specification [ETP-SS2010].

*ETPTSS 163* A Prescription Exchange shall implement an invoker for the E-Dispensing web service interface defined in Appendix A.4. This implementation shall conform to the conformance points associated with a prescription exchange acting as an intermediary in Section 4.6 of the ETP Solution Specification [ETP-SS2010].

## 7.3 Service Categories

The following conformance points define service categories used by the Prescription Exchange in its implementation.

*ETPTSS 164* When publishing an E-Prescribing service endpoint in a service directory, the Prescription Exchange shall use the service category:

http://ns.electronichealth.net.au/etp/sc/eprescribing/1.1

*ETPTSS 165* When publishing an E-Dispensing service endpoint in a service directory, the Prescription Exchange shall use the service category:

http://ns.electronichealth.net.au/etp/sc/edispensing/1.1

*ETPTSS 166* When searching for a E-Dispensing services in a service directory, the Prescription Exchange should use the service category:

http://ns.electronichealth.net.au/etp/sc/edispensing/1.1

## 7.4 Invocation

### 7.4.1 E-Prescribing `prescribe` operation

The following conformance points relate to the provision of the prescribe operation of the E-Prescribing interface.

*ETPTSS 167* The behaviour of the Prescription Exchange in handling requests, responses and faults relating to the prescribe operation shall comply with the provider conformance points of Section 4.5.6 of the ETP Solution Specification [ETP-SS2010].

*ETPTSS 168* The Prescription Exchange shall receive and process Prescription Available Events encoded using the PrescriptionAvailable XML schema defined in Appendix A.2 via the prescribe operation of its E-Prescribing web service interface.

*ETPTSS 169* The Prescription Exchange shall construct and send a Prescription Available Acknowledgement encoded using the PrescriptionAvailableAck XML schema defined in Appendix A.2 in the response to a successful prescribe operation.

*ETPTSS 170* The Prescription Exchange shall construct and raise a web service fault encoded as defined for the prescribe operation in the web service interface WSDL of Appendix A.3 in response to a failed prescribe operation.

### 7.4.2 E-Prescribing `retrievePrescription` operation

The following conformance points relate to the provision of the retrievePrescription operation of the E-Prescribing interface.

*ETPTSS 171* The behaviour of the Prescription Exchange in handling requests, responses and faults relating to the retrievePrescription operation shall comply with the provider conformance points of Sections 4.5.7 of the ETP Solution Specification [ETP-SS2010].

*ETPTSS 172* The Prescription Exchange shall receive and process Retrieve Prescription Requests encoded using the RetrievePrescriptionRequest XML schema defined in Appendix A.2 via the retrievePrescription operation of its E-Prescribing web service interface.

*ETPTSS 173* The Prescription Exchange shall construct and send a Retrieve Prescription Response encoded using the RetrievePrescriptionResponse XML schema defined in Appendix A.2 in response to a successful retrievePrescription operation.

*ETPTSS 174* The Prescription Exchange shall construct and raise a web service fault encoded as defined in the web service interface WSDL of Appendix A.3 in response to a failed retrievePrescription operation.

### 7.4.3 E-Prescribing `cancelPrescription` operation

The following conformance points relate to the provision of the cancelPrescription operation of the E-Prescribing interface.

*ETPTSS 175* The behaviour of the Prescription Exchange in handling requests, responses and faults relating to the E-Prescribing cancelPrescription operation shall comply with the provider conformance points of Section 4.5.6 of the ETP Solution Specification [ETP-SS2010].

*ETPTSS 176* The Prescription Exchange shall receive and process Prescription Cancelled Events from an E-Prescribing System using the PrescriptionCancelled XML schema defined in Appendix A.2 via the cancelPrescription operation of its E-Prescribing web service interface.

*ETPTSS 177* The Prescription Exchange shall construct and send a Prescription Cancelled Ack encoded using the PrescriptionCancelledAck XML schema defined in Appendix A.2 in response to a successful cancelPrescription operation.

*ETPTSS 178* The Prescription Exchange shall construct and raise a web service fault encoded as defined in the E-Prescribing web service interface WSDL of Appendix A.3 in response to a failed cancelPrescription operation.

### 7.4.4 E-Dispensing `initiateDispense` operation

The following conformance points relate to the provision of the initiateDispense operation of the E-Dispensing interface.

*ETPTSS 179* The Prescription Exchange shall receive and process Dispense Initiated Events encoded using the DispenseInitiated XML schema defined in Appendix A.2 via the initiateDispense operation of its E-Dispensing web service interface.

#### 7.4.4.1 Direct Provision of Service

The following conformance points apply to events where the provider identifier in the retrieval key matches the identifier associated with this provider.

*ETPTSS 180* The behaviour of the Prescription Exchange in handling requests, responses and faults relating to the initiateDispense operation shall comply with the provider conformance points of Sections 4.5.6 of the ETP Solution Specification [ETP-SS2010].

*ETPTSS 181* The Prescription Exchange shall construct and send a Dispense Initiated Acknowledgement encoded using the DispenseInitiatedAck XML schema defined in Appendix A.2 in the response to a successful initiateDispense operation.

*ETPTSS 182* The Prescription Exchange shall construct and raise a web service fault encoded as defined for the initiateDispense operation in the web service interface WSDL of Appendix A.4 in response to a failed initiateDispense operation.

#### 7.4.4.2 Indirect Provision of Service

See Section 7.4.7.2 for conformance points for the Prescription Exchange where the provider identifier does not match the identifier associated with this provider.

### 7.4.5 E-Dispensing `dispenseCompleted` operation

The following conformance points relate to the provision of the dispenseCompleted operation of the E-Dispensing interface.

*ETPTSS 183* The Prescription Exchange shall receive and process a Dispense Completed Event using the DispenseCompleted XML schema defined in Appendix A.2 via the dispenseCompleted operation of its E-Dispensing web service interface.

#### 7.4.5.1 Direct Provision of Service

The following conformance points apply to events where the provider identifier in the retrieval key matches the identifier associated with this provider.

*ETPTSS 184* The behaviour of the Prescription Exchange in handling requests, responses and faults relating to the dispenseCompleted operation shall comply with the provider conformance points of Sections 4.5.6 of the ETP Solution Specification [ETP-SS2010].

*ETPTSS 185* The Prescription Exchange shall construct and send a Dispense Completed Acknowledgement encoded using the DispenseCompletedAck XML schema defined in Appendix A.2 in the response to a successful dispenseCompleted operation.

*ETPTSS 186* The E-Dispensing System shall construct and raise a web service fault encoded as defined for the dispenseCompleted operation in the web

service interface WSDL of Appendix A.4 in response to a failed dispenseCompleted operation.

#### 7.4.5.2 Indirect Provision of Service

See Section 7.4.7.2 for conformance points for the Prescription Exchange where the provider identifier does not match the identifier associated with this provider.

### 7.4.6 E-Dispensing `dispenseTerminated` operation

The following conformance points relate to the provision of the dispenseTerminated operation of the E-Dispensing interface.

*ETPTSS 187* The Prescription Exchange shall receive and process Dispense Terminated Events using the DispenseTerminated XML schema defined in Appendix A.2 via the dispenseTerminated operation of its E-Dispensing web service interface.

#### 7.4.6.1 Direct Provision of Service

The following conformance points apply to events where the provider identifier in the retrieval key matches the identifier associated with this provider.

*ETPTSS 188* The behaviour of the Prescription Exchange in handling requests, responses and faults relating to the dispenseTerminated operation shall comply with the provider conformance points of Sections 4.5.6 of the ETP Solution Specification [ETP-SS2010].

*ETPTSS 189* The Prescription Exchange shall construct and send a Dispense Terminated Acknowledgement encoded using the DispenseTerminatedAck XML schema defined in Appendix A.2 in the response to a successful dispenseTerminated operation.

*ETPTSS 190* The Prescription Exchange shall construct and raise a web service fault encoded as defined for the dispenseTerminated operation in the web service interface WSDL of Appendix A.4 in response to a failed dispenseTerminated operation.

#### 7.4.6.2 Indirect Provision of Service

See Section 7.4.7.2 for conformance points for the Prescription Exchange where the provider identifier does not match the identifier associated with this provider.

### 7.4.7 E-Dispensing `dispenseReversed` operation

The following conformance points relate to the provision of the dispenseReversed operation of the E-Dispensing interface.

*ETPTSS 191* The Prescription Exchange shall receive and process Dispense Reversed Event using the DispenseReversed XML schema defined in Appendix A.2 via the dispenseReversed operation of its E-Dispensing web service interface.

#### 7.4.7.1 Direct Provision of Service

The following conformance points apply to events where the provider identifier in the retrieval key matches the identifier associated with this provider.

*ETPTSS 192* The behaviour of the Prescription Exchange in handling requests, responses and faults relating to the dispenseReversed operation shall comply with the provider conformance points of Section 4.5.6 of the ETP Solution Specification [ETP-SS2010].

*ETPTSS 193* The Prescription Exchange shall construct and send a Dispense Reversed Acknowledgement encoded using the DispenseReversedAck XML schema defined in Appendix A.2 in the response to a successful dispenseReversed operation.

*ETPTSS 194* The Prescription Exchange shall construct and raise a web service fault encoded as defined for the dispenseReversed operation in the web service interface WSDL of Appendix A.4 in response to a failed dispenseReversed operation.

#### 7.4.7.2 Indirect Provision of Service

See Section 7.4.8 for conformance points for the Prescription Exchange where the provider identifier does not match the identifier associated with this provider.

### 7.4.8 E-Dispensing `cancelPrescription` operation

The following conformance points relate to the provision of the cancelPrescription operation of the E-Dispensing interface.

*ETPTSS 195* The behaviour of the Prescription Exchange in handling requests, responses and faults relating to the E-Dispensing cancelPrescription operation shall comply with the provider conformance points of Section 4.5.6 of the ETP Solution Specification [ETP-SS2010].

*ETPTSS 196* The Prescription Exchange shall receive and process Prescription Cancelled Events from an E-Dispensing System using the PrescriptionCancelled XML schema defined in Appendix A.2 via the cancelPrescription operation of its E-Prescribing web service interface.

*ETPTSS 197* The Prescription Exchange shall construct and send a Prescription Cancelled Ack encoded using the PrescriptionCancelledAck XML schema defined in Appendix A.2 in response to a successful cancelPrescription operation.

*ETPTSS 198* The Prescription Exchange shall construct and raise a web service fault encoded as defined in the E-Dispensing web service interface WSDL of Appendix A.4 in response to a failed cancelPrescription operation.

#### 7.4.8.1 Indirect Provision of Service

See Section 7.4.8 for conformance points for the Prescription Exchange where the provider identifier does not match the identifier associated with this provider

### 7.4.9 E-Dispensing Indirect Provision of Service

When the provider identifier in the retrieval key associated with an E-Dispensing invocation does not match the identifier of this provider, the Prescription Exchange forwards the requests to the identified Prescription Exchange if possible. This behaviour is fully defined in Section 4.6 of the ETP Solution Specification [ETP-SS2010] except for the following.

*ETPTSS 199* If the provider identifier in the retrieval key of an event does not match any known provider or is not accessible, the Prescription Exchange shall construct and raise a fault as defined for the invoked operation in the web service interface WSDL of Appendix A.4 in response to the operation.

# Definitions

This section explains the terminology used in this document.

## Shortened terms

This table lists abbreviations and acronyms in alphabetical order.

| Term | Description |
|---|---|
| AMT | Australian Medicines Terminology – NEHTA specifications that standardise the identification, naming, and describing of medicine information. |
| CDA | Clinical Document Architecture |
| CIS | Clinical Information System |
| EDS | Electronic Dispensing System |
| EPS | Electronic Prescribing System |
| ETP | Electronic Transfer of Prescription |
| HL7 | Health Level 7 |
| HPI-I | Healthcare Provider Identifier – Individual |
| HPI-O | Healthcare Provider Identifier – Organisation |
| ICT | Information and Communication Technology |
| IETF | Internet Engineering Task Force |
| PCEHR | Personally Controlled Electronic Health Record |
| IHI | The Individual Health Identifier |
| NASH | National Authentication Service for Health |
| NEHTA | National E-Health Transition Authority |
| PE | Prescription Exchange |
| PES | Prescription Exchange Service |
| PKI | Public Key Infrastructure |
| RFC | Request for Comments |
| SMD | Secure Message Delivery |
| XML | Extensible Markup Language |

# Glossary

This table lists specialised terminology in alphabetical order.

| Term | Description |
|---|---|
| Australian Medicines Terminology | NEHTA specifications that standardise the identification, naming, and describing of medicine information. The Australian Medicines Terminology (AMT) delivers standardised identification of brand (trade) products and equivalent generic medicines along with associated components that are supported through standard naming conventions that accurately describe medications. |
| Clinical Document Architecture | The Clinical Document Architecture is a HL7 standard for the representation and machine processing of clinical documents in a way which makes the documents both human readable and machine processable and guarantees preservation of the content by using the eXtensible Markup Language (XML) standard. |
| Clinical Information System | Information computer technology used to store, manage and communicate healthcare information for healthcare providers and individuals, including the transfer of that information between information systems. |
| Department of Veteran's Affairs | Responsible for carrying out Government policy and implementing programs to fulfil Australia's obligations to war veterans and their dependants. |
| Dispense Record | Electronic record generated by a pharmacy information system that records medications dispensed or deferred. |
| Document Access Key | An alpha-numeric string that is used to identify and protect the set of clinical documents (Prescription and Dispense Records) for one prescription. |
| Dose Administration Aid | A device developed to assist patients in better managing their medicines by arranging their medicines into individual doses according to the prescribed dose schedule throughout the day, typically a compartmentalised box into which medicine doses can be placed; intended to help a subject of care to take the correct medicines at the correct times. They can also be designed to be tamper-evident, and can be either a unit-dose pack (i.e. one single type of medicine per compartment) or a multi-dose pack (i.e. different types of medicines per compartment). |
| e-Health | Use of information and communication technology to enable better healthcare outcomes. |
| Electronic Dispense Document | Electronic record generated by a pharmacy information system that records medications dispensed or deferred. |
| Electronic Dispensing System | A component of a clinical information system used to dispense medications. |
| Electronic Medications Management | Use of electronic systems to facilitate and enhance the communication of a prescription or medication order, aiding the choice, administration and supply of a medication through knowledge and decision support, and providing a robust audit trail for the entire medications use process. |
| Electronic Prescribing System | A component of a clinical information system used to prescribe medications. |
| e-Prescribing | Electronic prescribing is the process by which a prescription is electronically generated by a prescriber, authenticated with an electronic signature, securely transmitted to a Prescription Exchange Service for dispensing and supply, downloaded by a supplier, seamlessly integrated into the dispensing software and, in the case of Australian government subsidised prescriptions, is available to be electronically sent to Medicare Australia for claiming purposes. This definition does not preclude the use of paper-based processes to support electronic prescribing activity. Repeat and deferred supply authorisations which are uploaded to a Prescription Exchange Service by a supplier are not electronic authorisations, unless the original |

| Term | Description |
|---|---|
| | prescription was generated by a prescriber as an electronic prescription. |
| Electronic Prescription (or e-Prescription) | A prescription generated in an electronic manner and in a form approved by the relevant State, Territory or Commonwealth Departments of Health to convey a prescriber's authority to supply a medication. |
| Electronic Transfer of Prescription | A component of the EMM program of work that is concerned with the transfer of prescriptions, and its related information, in an electronic manner between participants. |
| Health Level Seven | A non-profit organisation involved in the construction and setting of Healthcare standards. "HL7" is also used to refer to some of the specific standards created by the organization (e.g., HL7 v2.x, v3.0, HL7 RIM). |
| Healthcare Provider Identifier - Individual | A 16 digit unique number used to identify individual providers who deliver healthcare in the Australian healthcare setting. |
| Healthcare Provider Identifier - Organisation | A 16 digit unique number used to identify organisations which deliver care in the Australian healthcare setting. |
| Individual | Persons who are, or could be, the subjects of care in the context of a healthcare event. |
| Individual Electronic Health Record | A secure, private electronic record of an individual's key health history and care information. |
| Individual Healthcare Identifier | A 16 digit unique number used to identify individuals who receive care in the Australian Healthcare system. |
| National Authentication Service for Health | A system for verifying the authenticity of patients and professionals for the purpose of ensuring the privacy of a person's electronic health data, while enabling secure access to the data by the person's authorised health providers. |
| National Clinical Terminology and Information Service | The National Clinical Terminology and Information Service (NCTIS), established by NEHTA, is developing the terminology and information products to support the requirements of e-health for the Australian healthcare community. |
| National E-Health Transition Authority | NEHTA Limited is a not-for-profit company established by the Australian, State and Territory governments to develop better ways of electronically collecting and securely exchanging health information. |
| Personal Health Record | A type of PCEHR that is initiated and maintained by the individual. |
| Personally Controlled Electronic Health Record | A secure, private electronic record of an individual's key health history and care information. |
| Pharmaceutical Benefits Scheme | A scheme set up under the National Health Act. Within Medicare Australia it is a system administered according to the Business Partnership Agreement with the Department of Health and Ageing (DoHA). Through the Pharmaceutical Benefits Scheme the Australian Government makes a range of necessary prescription medicines available at affordable prices to all Australian residents and those overseas visitors eligible under reciprocal Healthcare Agreements by paying part of the cost of the medicine to pharmacies. |
| Prescription | A request from a prescriber to dispense a therapeutic product. Describes the medication that the prescriber (a doctor in most cases) wants to be taken by the patient. It is input to the dispense process. Prescriptions are also used as input for the patient or the nurse on how to use the medication. |
| Prescription Exchange | An intermediary that stores Prescription and Dispense Records to allow them to be accessed by any authorised dispenser. |
| Prescription Exchange Service | An e-Health Service that provides Prescription Exchange functions. |
| Prescription Exchange | A dispense record submitted to a Prescription Exchange Service |

| Term | Description |
|---|---|
| Service - Dispense Record | (see Dispense Record). |
| Public Key Infrastructure | A set of hardware, software, people, policies, and procedures needed to create, manage, store, distribute, and revoke digital certificates. |
| Quality Use of Medicines | A central objective of the National Medicines Policy, applying to decisions about medicine use by individuals and decisions that affect the health of the population. Quality Use of Medicines (QUM) is one of the central objectives of Australia's National Medicines Policy. It means selecting management options wisely, choosing suitable medicines if a medicine is considered necessary, and using medicines safely and effectively. The definition of QUM applies equally to decisions about medicine use by individuals and decisions that affect the health of the population. Australia's National Medicines Policy is a cooperative endeavour to bring about better health outcomes for all Australians, focusing especially on people's access to, and wise use of, medicines. The term "medicine" includes prescription and non-prescription medicines, including complementary healthcare products. |
| Individual Health Identifier | A unique 16 digit number used to identify individuals who receive care in the Australian Healthcare system. |

# References

The following referenced documents are indispensable for the application of this document. Only the version cited applies.

## Package documents

The documents listed below are part of the suite delivered in the e-Discharge Summary Package.

| Electronic Transfer of **Prescription** Package Documents | | | |
|---|---|---|---|
| **[REF]** | **Document Name** | **Publisher** | **Link** |
| [ETP-ES2010] | Electronic Transfer of Prescription Release 1.1 – Executive Summary | NEHTA<br>2010 | http://www.nehta.gov.au/e-communications-in-practice/emedication-management |
| [ETP-RN2010] | Electronic Transfer of Prescription Release 1.1 – Release Note | | |
| [ETP-CO2010] | Electronic Transfer of Prescription Release 1.1 – Concept of Operations | | |
| [ETP-BR2010] | Electronic Transfer of Prescription Release 1.1 – Business Requirements Definition | | |
| [ETP-DR2010] | Electronic Transfer of Prescription Release 1.1 – Detailed Requirements Definition | | |
| [ETP-SS2010] | Electronic Transfer of Prescription Release 1.1 – Solution Specification | | |
| [ETP-EP_SDT2010] | Electronic Transfer of Prescription Release 1.1 – e-Prescription Structured Document Template (SDT) | | |
| [ETP-ED_SDT2010] | Electronic Transfer of Prescription Release 1.1 – Dispense Record Structured Document Template (SDT) | | |
| [ETP-PR_SDT2010] | Electronic Transfer of Prescription Release 1.1 – Prescription Request Structured Document Template (SDT) | | |
| [ETP-TSS2010] | Electronic Transfer of Prescription Release 1.1 – Technical Services Specification | | |
| [ETP-EP_CDAIG2010] | Electronic Transfer of Prescription Release 1.1 – e-Prescription Clinical Document Architecture Implementation Guide | | |
| [ETP-DR_CDAIG2010] | Electronic Transfer of Prescription Release 1.1 – Dispense Record Clinical Document Architecture Implementation Guide | | |
| [ETP-PR_CDAIG2010] | Electronic Transfer of Prescription Release 1.1 – Prescription Request Clinical Document Architecture Implementation Guide | | |

# External References

The documents listed below are non-package documents that have been cited in this document.

| Reference Documents | | | |
|---|---|---|---|
| [REF] | Document Name | Publisher | Link |
| [ATS 5820—2010] | Australian Technical Specification - E-Health Web Services Profiles | Standards Australia 2010 | http://www.e-health.standards.org.au/Home/Publications.aspx |
| [ATS 5821—2010] | Australian Technical Specification - E-Health XML Secured Payload Profiles | Standards Australia 2010 | http://www.e-health.standards.org.au/Home/Publications.aspx |
| [ATS 5822—2010] | Australian Technical Specification - E-Health Secure Message Delivery | Standards Australia 2010 | http://www.e-health.standards.org.au/Home/Publications.aspx |
| [RFC2119] | RFC 2119: Keywords for use in RFCs to Indicate Requirement Levels | IETF 1997 | http://ietf.org/rfc/rfc2119.txt |
| [RFC4122] | A Universally Unique IDentifier (UUID) URN Namespace | IETF 2005 | http://ietf.org/rfc/rfc4122.txt |
| [RFC4306] | Internet Key Exchange (IKEv2) Protocol | IETF 2005 | http://ietf.org/rfc/rfc4306.txt |
| [RFC4615] | The Advanced Encryption Standard-Cipher-based Message Authentication Code-Pseudo-Random Function-128 (AES-CMAC-PRF-128) Algorithm for the Internet Key Exchange Protocol (IKE) | IETF 2006 | http://ietf.org/rfc/rfc4615.txt |
| [UML2010] | OMG Unified Modeling Language (OMG UML), Superstructure, Version 2.3 | OMG 2010 | http://www.omg.org/spec/UML/2.3/ |

# Related reading

The documents listed below may provide further information about the issues discussed in this document.

| Related Documents | | | |
|---|---|---|---|
| [REF] | Document Name | Publisher | Link |
| [IF2007] | Interoperability Framework v2.0 | NEHTA 2007 | http://www.nehta.gov.au/ Search “interoperability 2.0”. |
| [TR 5823—2010] | Technical Report – Endpoint Location Service | Standards Australia 2010 | http://www.e-health.standards.org.au/Home/Publications.aspx |

# Key Contacts

Contacts listed below will be able to clarify provide further information about the issues discussed in this document.

| Contacts | | |
|---|---|---|
| **Contact name** | **Email Address** | **Phone Number** |
| Toby Mathieson<br>Program Manager, Electronic Medication Management, NEHTA | Toby.Mathieson@nehta.gov.au | +61 2 8298 2605 |

# Appendix A: XSD and WSDL

The following subsections contain the WSDL and concrete XML schemas (XSD) for data types referenced in this specification and also the Solution Specification [ETP-SS2010].

## A.1 Notification Data Types

The following XML schema defines the XSD for data associated with the ETP notifications.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            xmlns:tns="http://ns.electronichealth.net.au/etp/xsd/EtpNotification/1.1"
            xmlns:pes="http://ns.electronichealth.net.au/etp/xsd/PesDataTypes/1.1"
            targetNamespace="http://ns.electronichealth.net.au/etp/xsd/EtpNotification/1.1"
            elementFormDefault="qualified">
   <xsd:import namespace="http://ns.electronichealth.net.au/etp/xsd/PesDataTypes/1.1"
               schemaLocation="etp-PesDataTypes-1.1.xsd"/>
   <xsd:element name="electronicPrescriptionNotification"
                type="tns:ElectronicPrescriptionNotificationType"/>
   <xsd:complexType name="ElectronicPrescriptionNotificationType">
      <xsd:complexContent>
         <xsd:extension base="tns:EtpNotificationType">
            <xsd:sequence>
               <xsd:element name="dak" type="pes:DakType" minOccurs="1" maxOccurs="1"/>
               <xsd:element name="prescription" type="xsd:base64Binary" minOccurs="1"
maxOccurs="1"/>
            </xsd:sequence>
         </xsd:extension>
      </xsd:complexContent>
   </xsd:complexType>
   <xsd:complexType name="EtpNotificationType" abstract="true">
      <xsd:sequence>
         <xsd:element name="notificationId" type="xsd:anyURI" minOccurs="1" maxOccurs="1"/>
         <xsd:element name="subject" type="xsd:anyURI" minOccurs="1" maxOccurs="1"/>
         <xsd:element name="notificationTimestamp" type="xsd:dateTime" minOccurs="1"
maxOccurs="1"/>
         <xsd:element name="clinicalProcessId" type="xsd:anyURI" minOccurs="1" maxOccurs="1"/>
         <xsd:element name="clinician" type="xsd:anyURI" minOccurs="1" maxOccurs="1"/>
         <xsd:element name="organisation" type="xsd:anyURI" minOccurs="1" maxOccurs="1"/>
      </xsd:sequence>
   </xsd:complexType>
   <xsd:element name="lastSupplyNotification" type="tns:LastSupplyNotificationType"/>
   <xsd:complexType name="LastSupplyNotificationType">
      <xsd:complexContent>
         <xsd:extension base="tns:EtpNotificationType">
            <xsd:sequence>
               <xsd:element name="dak" type="pes:DakType" minOccurs="1" maxOccurs="1"/>
            </xsd:sequence>
         </xsd:extension>
      </xsd:complexContent>
   </xsd:complexType>
   <xsd:element name="prescriptionAvailableNotification"
                type="tns:PrescriptionAvailableNotificationType"/>
   <xsd:complexType name="PrescriptionAvailableNotificationType">
      <xsd:complexContent>
         <xsd:extension base="tns:EtpNotificationType">
            <xsd:sequence>
               <xsd:element name="dak" type="pes:DakType" minOccurs="1" maxOccurs="1"/>
            </xsd:sequence>
         </xsd:extension>
      </xsd:complexContent>
   </xsd:complexType>
   <xsd:element name="prescriptionRequiredNotification"
                type="tns:PrescriptionRequiredNotificationType"/>
   <xsd:complexType name="PrescriptionRequiredNotificationType">
      <xsd:complexContent>
         <xsd:extension base="tns:EtpNotificationType">
            <xsd:sequence>
               <xsd:element name="prescriptionRequest" type="xsd:base64Binary" minOccurs="1"
maxOccurs="1"/>
            </xsd:sequence>
         </xsd:extension>
      </xsd:complexContent>
   </xsd:complexType>
   <xsd:element name="requiredPrescriptionNotification"
                type="tns:RequiredPrescriptionNotificationType"/>
```

```
   <xsd:complexType name="RequiredPrescriptionNotificationType">
      <xsd:complexContent>
         <xsd:extension base="tns:EtpNotificationType">
            <xsd:sequence>
               <xsd:element name="dak" type="pes:DakType" minOccurs="1" maxOccurs="1"/>
            </xsd:sequence>
         </xsd:extension>
      </xsd:complexContent>
   </xsd:complexType>
</xsd:schema>
```

## A.2 PES Event and Event ACK Types

The following XML schema defines the XSD for data associated with the ETP PES Events and Event Acknowledgement types.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            xmlns:ep="http://ns.electronichealth.net.au/xsp/xsd/EncryptedPayload/2010"
            xmlns:tns="http://ns.electronichealth.net.au/etp/xsd/PesDataTypes/1.1"
            targetNamespace="http://ns.electronichealth.net.au/etp/xsd/PesDataTypes/1.1"
            elementFormDefault="qualified">
   <xsd:import namespace="http://ns.electronichealth.net.au/xsp/xsd/EncryptedPayload/2010"
               schemaLocation="xsp-EncryptedPayload-2010.xsd"/>
   <xsd:simpleType name="AcknowledgementStatusType">
      <xsd:restriction base="xsd:string">
         <xsd:enumeration value="ok"/>
         <xsd:enumeration value="duplicate"/>
      </xsd:restriction>
   </xsd:simpleType>
   <xsd:complexType name="ClinicalDocumentMetadataType">
      <xsd:sequence>
         <xsd:element name="documentId" type="xsd:anyURI" minOccurs="1" maxOccurs="1"/>
         <xsd:element name="docType" type="xsd:anyURI" minOccurs="1" maxOccurs="1"/>
         <xsd:element name="qualifiedRK" type="tns:QualifiedRkType" minOccurs="1"
maxOccurs="1"/>
         <xsd:element name="author" type="xsd:anyURI" minOccurs="1" maxOccurs="1"/>
         <xsd:element name="originator" type="xsd:anyURI" minOccurs="1" maxOccurs="1"/>
         <xsd:element name="originatorOrganisation" type="xsd:anyURI" minOccurs="1"
maxOccurs="1"/>
         <xsd:element name="creationTime" type="xsd:dateTime" minOccurs="1" maxOccurs="1"/>
      </xsd:sequence>
   </xsd:complexType>
   <xsd:complexType name="CurrentInterestType">
      <xsd:sequence>
         <xsd:element name="clinician" type="xsd:anyURI" minOccurs="1" maxOccurs="1"/>
         <xsd:element name="organisation" type="xsd:anyURI" minOccurs="1" maxOccurs="1"/>
         <xsd:element name="clinicalProcessId" type="xsd:anyURI" minOccurs="1" maxOccurs="1"/>
         <xsd:element name="timestamp" type="xsd:dateTime" minOccurs="1" maxOccurs="1"/>
      </xsd:sequence>
   </xsd:complexType>
   <xsd:simpleType name="DakType">
      <xsd:restriction base="xsd:string"/>
   </xsd:simpleType>
   <xsd:complexType name="DispenseFinalisedAckType">
      <xsd:complexContent>
         <xsd:extension base="tns:PesEventAckType">
            <xsd:sequence>
               <xsd:element name="qualifiedRk" type="tns:QualifiedRkType" minOccurs="1"
maxOccurs="1"/>
               <xsd:element name="resultingDispensed" type="xsd:int" minOccurs="1"
maxOccurs="1"/>
               <xsd:element name="resultingRemaining" type="xsd:int" minOccurs="1"
maxOccurs="1"/>
               <xsd:element name="dispenseDocId" type="xsd:anyURI" minOccurs="1"
maxOccurs="1"/>
            </xsd:sequence>
         </xsd:extension>
      </xsd:complexContent>
   </xsd:complexType>
   <xsd:complexType name="DispenseFinalisedEventType">
      <xsd:complexContent>
         <xsd:extension base="tns:PesEventType">
            <xsd:sequence>
               <xsd:element name="dispenseDoc" type="tns:EDispenseDocumentType" minOccurs="1"
maxOccurs="1"/>
               <xsd:element name="additionalReduction" type="xsd:int" minOccurs="1"
maxOccurs="1"/>
            </xsd:sequence>
         </xsd:extension>
```

```
      </xsd:complexContent>
   </xsd:complexType>
   <xsd:complexType name="DispenseInitiatedAckType">
      <xsd:complexContent>
         <xsd:extension base="tns:PesEventAckType">
            <xsd:sequence>
               <xsd:element name="prescriptionDoc" type="tns:EPrescriptionDocumentType"
minOccurs="1"
                            maxOccurs="1"/>
               <xsd:element name="dispenseDoc" type="tns:EDispenseDocumentType" minOccurs="0"
                            maxOccurs="unbounded"/>
               <xsd:element name="currentInterest" type="tns:CurrentInterestType"
minOccurs="0"
                            maxOccurs="unbounded"/>
            </xsd:sequence>
         </xsd:extension>
      </xsd:complexContent>
   </xsd:complexType>
   <xsd:complexType name="DispenseInitiatedEventType">
      <xsd:complexContent>
         <xsd:extension base="tns:PesEventType">
            <xsd:sequence>
               <xsd:element name="qualifiedRk" type="tns:QualifiedRkType" minOccurs="1"
maxOccurs="1"/>
            </xsd:sequence>
         </xsd:extension>
      </xsd:complexContent>
   </xsd:complexType>
   <xsd:complexType name="DispenseReversedAckType">
      <xsd:complexContent>
         <xsd:extension base="tns:PesEventAckType"/>
      </xsd:complexContent>
   </xsd:complexType>
   <xsd:complexType name="DispenseReversedEventType">
      <xsd:complexContent>
         <xsd:extension base="tns:PesEventType">
            <xsd:sequence>
               <xsd:element name="qualifiedRk" type="tns:QualifiedRkType" minOccurs="1"
maxOccurs="1"/>
            </xsd:sequence>
         </xsd:extension>
      </xsd:complexContent>
   </xsd:complexType>
   <xsd:complexType name="DispenseTerminatedAckType">
      <xsd:complexContent>
         <xsd:extension base="tns:PesEventAckType"/>
      </xsd:complexContent>
   </xsd:complexType>
   <xsd:complexType name="DispenseTerminatedEventType">
      <xsd:complexContent>
         <xsd:extension base="tns:PesEventType">
            <xsd:sequence>
               <xsd:element name="qualifiedRk" type="tns:QualifiedRkType" minOccurs="1"
maxOccurs="1"/>
            </xsd:sequence>
         </xsd:extension>
      </xsd:complexContent>
   </xsd:complexType>
   <xsd:simpleType name="DispensingStateType">
      <xsd:restriction base="xsd:string">
         <xsd:enumeration value="availableToDispense"/>
         <xsd:enumeration value="cancelled"/>
         <xsd:enumeration value="expired"/>
         <xsd:enumeration value="fullyDispensed"/>
      </xsd:restriction>
   </xsd:simpleType>
   <xsd:complexType name="EDispenseDocumentType">
      <xsd:complexContent>
         <xsd:extension base="tns:SecuredClinicalDocumentType"/>
      </xsd:complexContent>
   </xsd:complexType>
   <xsd:complexType name="EPrescriptionDocumentType">
      <xsd:complexContent>
         <xsd:extension base="tns:SecuredClinicalDocumentType"/>
      </xsd:complexContent>
   </xsd:complexType>
   <xsd:complexType name="EPrescriptionType">
      <xsd:sequence>
         <xsd:element name="qualifiedRk" type="tns:QualifiedRkType" minOccurs="1"
maxOccurs="1"/>
         <xsd:element name="dispensingState" type="tns:DispensingStateType" minOccurs="1"
                      maxOccurs="1"/>
         <xsd:element name="expiryDate" type="xsd:dateTime" minOccurs="1" maxOccurs="1"/>
         <xsd:element name="dispensed" type="xsd:int" minOccurs="1" maxOccurs="1"/>
         <xsd:element name="remaining" type="xsd:int" minOccurs="1" maxOccurs="1"/>
```

```
         <xsd:element name="currentInterest" type="tns:CurrentInterestType" minOccurs="0"
                     maxOccurs="unbounded"/>
         <xsd:element name="ePrescriptionDoc" type="tns:EPrescriptionDocumentType"
minOccurs="1"
                     maxOccurs="1"/>
         <xsd:element name="eDispenseDoc" type="tns:EDispenseDocumentType" minOccurs="0"
                     maxOccurs="unbounded"/>
      </xsd:sequence>
   </xsd:complexType>
   <xsd:element name="generalEtpError" type="tns:GeneralEtpErrorType"/>
   <xsd:complexType name="GeneralEtpErrorType">
      <xsd:sequence>
         <xsd:element name="errorCode" type="xsd:anyURI" minOccurs="1" maxOccurs="1"/>
         <xsd:element name="message" type="xsd:string" minOccurs="1" maxOccurs="1"/>
      </xsd:sequence>
   </xsd:complexType>
   <xsd:complexType name="PesEventAckType" abstract="true">
      <xsd:sequence>
         <xsd:element name="pesEventId" type="xsd:anyURI" minOccurs="1" maxOccurs="1"/>
         <xsd:element name="exchangeTimestamp" type="xsd:dateTime" minOccurs="1"
maxOccurs="1"/>
         <xsd:element name="status" type="tns:AcknowledgementStatusType" minOccurs="1"
maxOccurs="1"/>
      </xsd:sequence>
   </xsd:complexType>
   <xsd:complexType name="PesEventType" abstract="true">
      <xsd:sequence>
         <xsd:element name="pesEventId" type="xsd:anyURI" minOccurs="1" maxOccurs="1"/>
         <xsd:element name="clinicalProcessId" type="xsd:anyURI" minOccurs="1" maxOccurs="1"/>
         <xsd:element name="creationTimestamp" type="xsd:dateTime" minOccurs="1"
maxOccurs="1"/>
         <xsd:element name="clinician" type="xsd:anyURI" minOccurs="1" maxOccurs="1"/>
         <xsd:element name="organisation" type="xsd:anyURI" minOccurs="1" maxOccurs="1"/>
      </xsd:sequence>
   </xsd:complexType>
   <xsd:complexType name="PrescriptionAvailableAckType">
      <xsd:complexContent>
         <xsd:extension base="tns:PesEventAckType"/>
      </xsd:complexContent>
   </xsd:complexType>
   <xsd:complexType name="PrescriptionAvailableEventType">
      <xsd:complexContent>
         <xsd:extension base="tns:PesEventType">
            <xsd:sequence>
               <xsd:element name="prescriptionDoc" type="tns:EPrescriptionDocumentType"
minOccurs="1"
                            maxOccurs="1"/>
               <xsd:element name="expiryDate" type="xsd:dateTime" minOccurs="1"
maxOccurs="1"/>
               <xsd:element name="numberOfDispensings" type="xsd:int" minOccurs="1"
maxOccurs="1"/>
            </xsd:sequence>
         </xsd:extension>
      </xsd:complexContent>
   </xsd:complexType>
   <xsd:complexType name="PrescriptionCancelledAckType">
      <xsd:complexContent>
         <xsd:extension base="tns:PesEventAckType"/>
      </xsd:complexContent>
   </xsd:complexType>
   <xsd:complexType name="PrescriptionCancelledEventType">
      <xsd:complexContent>
         <xsd:extension base="tns:PesEventType">
            <xsd:sequence>
               <xsd:element name="qualifiedRk" type="tns:QualifiedRkType" minOccurs="1"
maxOccurs="1"/>
            </xsd:sequence>
         </xsd:extension>
      </xsd:complexContent>
   </xsd:complexType>
   <xsd:complexType name="PrescriptionRetrievalRequestType">
      <xsd:complexContent>
         <xsd:extension base="tns:RetrievalRequestType"/>
      </xsd:complexContent>
   </xsd:complexType>
   <xsd:complexType name="PrescriptionRetrievalResponseType">
      <xsd:complexContent>
         <xsd:extension base="tns:RetrievalResponseType">
            <xsd:sequence>
               <xsd:element name="prescriptionDoc" type="tns:EPrescriptionDocumentType"
minOccurs="1"
                            maxOccurs="1"/>
            </xsd:sequence>
         </xsd:extension>
      </xsd:complexContent>
```

```
   </xsd:complexType>
   <xsd:complexType name="QualifiedRkType">
      <xsd:sequence>
         <xsd:element name="providerId" type="xsd:string" minOccurs="1" maxOccurs="1"/>
         <xsd:element name="retrievalKey" type="xsd:base64Binary" minOccurs="1"
maxOccurs="1"/>
      </xsd:sequence>
   </xsd:complexType>
   <xsd:complexType name="RetrievalRequestType" abstract="true">
      <xsd:sequence>
         <xsd:element name="qualifiedRk" type="tns:QualifiedRkType" minOccurs="1"
maxOccurs="1"/>
         <xsd:element name="requestId" type="xsd:anyURI" minOccurs="1" maxOccurs="1"/>
         <xsd:element name="requestTimestamp" type="xsd:dateTime" minOccurs="1"
maxOccurs="1"/>
         <xsd:element name="clinician" type="xsd:anyURI" minOccurs="1" maxOccurs="1"/>
         <xsd:element name="organisation" type="xsd:anyURI" minOccurs="1" maxOccurs="1"/>
      </xsd:sequence>
   </xsd:complexType>
   <xsd:complexType name="RetrievalResponseType" abstract="true">
      <xsd:sequence>
         <xsd:element name="requestId" type="xsd:anyURI" minOccurs="1" maxOccurs="1"/>
         <xsd:element name="exchangeTimestamp" type="xsd:dateTime" minOccurs="1"
maxOccurs="1"/>
      </xsd:sequence>
   </xsd:complexType>
   <xsd:complexType name="SecuredClinicalDocumentType" abstract="true">
      <xsd:sequence>
         <xsd:element name="metadata" type="tns:ClinicalDocumentMetadataType" minOccurs="1"
                      maxOccurs="1"/>
         <xsd:element ref="ep:encryptedPayload" minOccurs="1" maxOccurs="1"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:schema>
```

# A.3 E-Prescribing Interface

## A.3.1 Interface Definition

The following WSDL specification defines the E-Prescribing SOAP interface.

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata"
                  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
                  xmlns:tns="http://ns.electronichealth.net.au/etp/svc/ePrescribing/1.1"
                  xmlns:pes="http://ns.electronichealth.net.au/etp/xsd/PesDataTypes/1.1"
                  xmlns:se="http://ns.electronichealth.net.au/wsp/xsd/StandardError/2010"
                  name="ePrescribing"

targetNamespace="http://ns.electronichealth.net.au/etp/svc/ePrescribing/1.1">
   <wsdl:types>
      <xsd:schema targetNamespace="http://ns.electronichealth.net.au/etp/svc/ePrescribing/1.1"
                  elementFormDefault="qualified">
         <xsd:import namespace="http://ns.electronichealth.net.au/etp/xsd/PesDataTypes/1.1"
                     schemaLocation="../xsd/etp-PesDataTypes-1.1.xsd"/>
         <xsd:import namespace="http://ns.electronichealth.net.au/wsp/xsd/StandardError/2010"
                     schemaLocation="../xsd/wsp-StandardError-2010.xsd"/>
         <xsd:element name="cancelPrescription">
            <xsd:complexType>
               <xsd:sequence>
                  <xsd:element name="request" type="pes:PrescriptionCancelledEventType"
minOccurs="1"
                               maxOccurs="1"/>
               </xsd:sequence>
            </xsd:complexType>
         </xsd:element>
         <xsd:element name="cancelPrescriptionResponse">
            <xsd:complexType>
               <xsd:sequence>
                  <xsd:element name="response" type="pes:PrescriptionCancelledAckType"
minOccurs="1"
                               maxOccurs="1"/>
               </xsd:sequence>
            </xsd:complexType>
         </xsd:element>
         <xsd:element name="prescribe">
            <xsd:complexType>
               <xsd:sequence>
```

```
                  <xsd:element name="request" type="pes:PrescriptionAvailableEventType"
minOccurs="1"
                               maxOccurs="1"/>
               </xsd:sequence>
            </xsd:complexType>
         </xsd:element>
         <xsd:element name="prescribeResponse">
            <xsd:complexType>
               <xsd:sequence>
                  <xsd:element name="response" type="pes:PrescriptionAvailableAckType"
minOccurs="1"
                               maxOccurs="1"/>
               </xsd:sequence>
            </xsd:complexType>
         </xsd:element>
         <xsd:element name="retrievePrescription">
            <xsd:complexType>
               <xsd:sequence>
                  <xsd:element name="request" type="pes:PrescriptionRetrievalRequestType"
minOccurs="1"
                               maxOccurs="1"/>
               </xsd:sequence>
            </xsd:complexType>
         </xsd:element>
         <xsd:element name="retrievePrescriptionResponse">
            <xsd:complexType>
               <xsd:sequence>
                  <xsd:element name="response" type="pes:PrescriptionRetrievalResponseType"
minOccurs="1"
                               maxOccurs="1"/>
               </xsd:sequence>
            </xsd:complexType>
         </xsd:element>
         <xsd:element name="cancelPrescriptionError" type="tns:CancelPrescriptionErrorType"/>
         <xsd:element name="prescribeError" type="tns:PrescribeErrorType"/>
         <xsd:simpleType name="CancelPrescriptionErrorCodeType">
            <xsd:restriction base="xsd:string">
               <xsd:enumeration value="invalidRepositoryId"/>
               <xsd:enumeration value="invalidTimestamp"/>
               <xsd:enumeration value="unknownRetrievalKey"/>
               <xsd:enumeration value="fullyDispensed"/>
               <xsd:enumeration value="cancelled"/>
               <xsd:enumeration value="expired"/>
               <xsd:enumeration value="notAuthorised"/>
            </xsd:restriction>
         </xsd:simpleType>
         <xsd:simpleType name="PrescribeErrorCodeType">
            <xsd:restriction base="xsd:string">
               <xsd:enumeration value="invalidRepositoryId"/>
               <xsd:enumeration value="invalidTimestamp"/>
               <xsd:enumeration value="duplicateRk"/>
               <xsd:enumeration value="duplicateDocumentId"/>
               <xsd:enumeration value="invalidDocumentType"/>
               <xsd:enumeration value="notAuthorised"/>
            </xsd:restriction>
         </xsd:simpleType>
         <xsd:complexType name="CancelPrescriptionErrorType">
            <xsd:sequence>
               <xsd:element name="errorCode" type="tns:CancelPrescriptionErrorCodeType"
minOccurs="1"
                            maxOccurs="1"/>
               <xsd:element name="message" type="xsd:string" minOccurs="1" maxOccurs="1"/>
            </xsd:sequence>
         </xsd:complexType>
         <xsd:complexType name="PrescribeErrorType">
            <xsd:sequence>
               <xsd:element name="errorCode" type="tns:PrescribeErrorCodeType" minOccurs="1"
maxOccurs="1"/>
               <xsd:element name="message" type="xsd:string" minOccurs="1" maxOccurs="1"/>
            </xsd:sequence>
         </xsd:complexType>
      </xsd:schema>
   </wsdl:types>
   <wsdl:message name="cancelPrescriptionInMsg">
      <wsdl:part name="body" element="tns:cancelPrescription"/>
   </wsdl:message>
   <wsdl:message name="cancelPrescriptionOutMsg">
      <wsdl:part name="body" element="tns:cancelPrescriptionResponse"/>
   </wsdl:message>
   <wsdl:message name="prescribeInMsg">
      <wsdl:part name="body" element="tns:prescribe"/>
   </wsdl:message>
   <wsdl:message name="prescribeOutMsg">
      <wsdl:part name="body" element="tns:prescribeResponse"/>
   </wsdl:message>
```

```
   <wsdl:message name="retrievePrescriptionInMsg">
      <wsdl:part name="body" element="tns:retrievePrescription"/>
   </wsdl:message>
   <wsdl:message name="retrievePrescriptionOutMsg">
      <wsdl:part name="body" element="tns:retrievePrescriptionResponse"/>
   </wsdl:message>
   <wsdl:message name="standardErrorMsg">
      <wsdl:part name="fault" element="se:standardError"/>
   </wsdl:message>
   <wsdl:message name="cancelPrescriptionErrorMsg">
      <wsdl:part name="fault" element="tns:cancelPrescriptionError"/>
   </wsdl:message>
   <wsdl:message name="generalEtpErrorMsg">
      <wsdl:part name="fault" element="pes:generalEtpError"/>
   </wsdl:message>
   <wsdl:message name="prescribeErrorMsg">
      <wsdl:part name="fault" element="tns:prescribeError"/>
   </wsdl:message>
   <wsdl:portType name="EPrescribing">
      <wsdl:operation name="cancelPrescription">
         <wsdl:input message="tns:cancelPrescriptionInMsg"

wsam:Action="http://ns.electronichealth.net.au/etp/svc/ePrescribing/1.1/EPrescribing/cancelPre
scriptionRequest"/>
         <wsdl:output message="tns:cancelPrescriptionOutMsg"

wsam:Action="http://ns.electronichealth.net.au/etp/svc/ePrescribing/1.1/EPrescribing/cancelPre
scriptionResponse"/>
         <wsdl:fault name="standardError" message="tns:standardErrorMsg"

wsam:Action="http://ns.electronichealth.net.au/etp/svc/ePrescribing/1.1/EPrescribing/cancelPre
scription/Fault/standardError"/>
         <wsdl:fault name="cancelPrescriptionError" message="tns:cancelPrescriptionErrorMsg"

wsam:Action="http://ns.electronichealth.net.au/etp/svc/ePrescribing/1.1/EPrescribing/cancelPre
scription/Fault/cancelPrescriptionError"/>
         <wsdl:fault name="generalEtpError" message="tns:generalEtpErrorMsg"

wsam:Action="http://ns.electronichealth.net.au/etp/svc/ePrescribing/1.1/EPrescribing/cancelPre
scription/Fault/generalEtpError"/>
      </wsdl:operation>
      <wsdl:operation name="prescribe">
         <wsdl:input message="tns:prescribeInMsg"

wsam:Action="http://ns.electronichealth.net.au/etp/svc/ePrescribing/1.1/EPrescribing/prescribe
Request"/>
         <wsdl:output message="tns:prescribeOutMsg"

wsam:Action="http://ns.electronichealth.net.au/etp/svc/ePrescribing/1.1/EPrescribing/prescribe
Response"/>
         <wsdl:fault name="standardError" message="tns:standardErrorMsg"

wsam:Action="http://ns.electronichealth.net.au/etp/svc/ePrescribing/1.1/EPrescribing/prescribe
/Fault/standardError"/>
         <wsdl:fault name="prescribeError" message="tns:prescribeErrorMsg"

wsam:Action="http://ns.electronichealth.net.au/etp/svc/ePrescribing/1.1/EPrescribing/prescribe
/Fault/prescribeError"/>
         <wsdl:fault name="generalEtpError" message="tns:generalEtpErrorMsg"

wsam:Action="http://ns.electronichealth.net.au/etp/svc/ePrescribing/1.1/EPrescribing/prescribe
/Fault/generalEtpError"/>
      </wsdl:operation>
      <wsdl:operation name="retrievePrescription">
         <wsdl:input message="tns:retrievePrescriptionInMsg"

wsam:Action="http://ns.electronichealth.net.au/etp/svc/ePrescribing/1.1/EPrescribing/retrieveP
rescriptionRequest"/>
         <wsdl:output message="tns:retrievePrescriptionOutMsg"

wsam:Action="http://ns.electronichealth.net.au/etp/svc/ePrescribing/1.1/EPrescribing/retrieveP
rescriptionResponse"/>
         <wsdl:fault name="standardError" message="tns:standardErrorMsg"

wsam:Action="http://ns.electronichealth.net.au/etp/svc/ePrescribing/1.1/EPrescribing/retrieveP
rescription/Fault/standardError"/>
         <wsdl:fault name="generalEtpError" message="tns:generalEtpErrorMsg"

wsam:Action="http://ns.electronichealth.net.au/etp/svc/ePrescribing/1.1/EPrescribing/retrieveP
rescription/Fault/generalEtpError"/>
      </wsdl:operation>
   </wsdl:portType>
</wsdl:definitions>
```

### A.3.2 TLS Binding

The following WSDL defines the binding of the E-Prescribing interface using a policy for TLS authentication and session privacy.

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap12/"
                  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                  xmlns:wsp="http://www.w3.org/ns/ws-policy"
                  xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata"
                  xmlns:sp="http://docs.oasis-open.org/ws-sx/ws-securitypolicy/200702"
                  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
                  xmlns:tns="http://ns.electronichealth.net.au/etp/svc/ePrescribing/1.1"
                  xmlns:pes="http://ns.electronichealth.net.au/etp/xsd/PesDataTypes/1.1"
                  xmlns:se="http://ns.electronichealth.net.au/wsp/xsd/StandardError/2010"
                  name="ePrescribing"

targetNamespace="http://ns.electronichealth.net.au/etp/svc/ePrescribing/1.1">
   <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>
            <!-- Use TLS protocol negotiation instead of <sp:AlgorithmSuite/> -->
         </wsp:Policy>
      </sp:TransportBinding>
   </wsp:Policy>
   <wsdl:import namespace="http://ns.electronichealth.net.au/etp/svc/ePrescribing/1.1"
                location="etp-ePrescribing-Interface-1.1.wsdl"/>
   <wsdl:binding name="EPrescribingBinding" type="tns:EPrescribing">
      <wsp:PolicyReference URI="#AddressingPolicy"/>
      <wsp:PolicyReference URI="#TlsPolicy"/>
      <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
      <wsdl:operation name="cancelPrescription">
         <soap:operation style="document" soapActionRequired="false"/>
         <wsdl:input>
            <soap:body use="literal"/>
         </wsdl:input>
         <wsdl:output>
            <soap:body use="literal"/>
         </wsdl:output>
         <wsdl:fault name="standardError">
            <soap:fault name="standardError" use="literal"/>
         </wsdl:fault>
         <wsdl:fault name="cancelPrescriptionError">
            <soap:fault name="cancelPrescriptionError" use="literal"/>
         </wsdl:fault>
         <wsdl:fault name="generalEtpError">
            <soap:fault name="generalEtpError" use="literal"/>
         </wsdl:fault>
      </wsdl:operation>
      <wsdl:operation name="prescribe">
         <soap:operation style="document" soapActionRequired="false"/>
         <wsdl:input>
            <soap:body use="literal"/>
         </wsdl:input>
         <wsdl:output>
            <soap:body use="literal"/>
         </wsdl:output>
         <wsdl:fault name="standardError">
            <soap:fault name="standardError" use="literal"/>
         </wsdl:fault>
         <wsdl:fault name="prescribeError">
            <soap:fault name="prescribeError" use="literal"/>
         </wsdl:fault>
         <wsdl:fault name="generalEtpError">
            <soap:fault name="generalEtpError" use="literal"/>
         </wsdl:fault>
      </wsdl:operation>
      <wsdl:operation name="retrievePrescription">
         <soap:operation style="document" soapActionRequired="false"/>
         <wsdl:input>
            <soap:body use="literal"/>
```

```
         </wsdl:input>
         <wsdl:output>
            <soap:body use="literal"/>
         </wsdl:output>
         <wsdl:fault name="standardError">
            <soap:fault name="standardError" use="literal"/>
         </wsdl:fault>
         <wsdl:fault name="generalEtpError">
            <soap:fault name="generalEtpError" use="literal"/>
         </wsdl:fault>
      </wsdl:operation>
   </wsdl:binding>
   <wsdl:service name="EPrescribingService">
      <wsdl:port name="EPrescribing" binding="tns:EPrescribingBinding">
         <soap:address location="http://dummy.example.com"/>
      </wsdl:port>
   </wsdl:service>
</wsdl:definitions>
```

# A.4 E-Dispensing Interface

## A.4.1 Interface Definition

The following WSDL specification defines the E-Dispensing SOAP interface.

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata"
                  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
                  xmlns:tns="http://ns.electronichealth.net.au/etp/svc/eDispensing/1.1"
                  xmlns:pes="http://ns.electronichealth.net.au/etp/xsd/PesDataTypes/1.1"
                  xmlns:se="http://ns.electronichealth.net.au/wsp/xsd/StandardError/2010"
                  name="eDispensing"
                  targetNamespace="http://ns.electronichealth.net.au/etp/svc/eDispensing/1.1">
   <wsdl:types>
      <xsd:schema targetNamespace="http://ns.electronichealth.net.au/etp/svc/eDispensing/1.1"
                  elementFormDefault="qualified">
         <xsd:import namespace="http://ns.electronichealth.net.au/etp/xsd/PesDataTypes/1.1"
                     schemaLocation="../xsd/etp-PesDataTypes-1.1.xsd"/>
         <xsd:import namespace="http://ns.electronichealth.net.au/wsp/xsd/StandardError/2010"
                     schemaLocation="../xsd/wsp-StandardError-2010.xsd"/>
         <xsd:element name="cancelPrescription">
            <xsd:complexType>
               <xsd:sequence>
                  <xsd:element name="request" type="pes:PrescriptionCancelledEventType"
minOccurs="1"
                               maxOccurs="1"/>
               </xsd:sequence>
            </xsd:complexType>
         </xsd:element>
         <xsd:element name="cancelPrescriptionResponse">
            <xsd:complexType>
               <xsd:sequence>
                  <xsd:element name="response" type="pes:PrescriptionCancelledAckType"
minOccurs="1"
                               maxOccurs="1"/>
               </xsd:sequence>
            </xsd:complexType>
         </xsd:element>
         <xsd:element name="finaliseDispense">
            <xsd:complexType>
               <xsd:sequence>
                  <xsd:element name="request" type="pes:DispenseFinalisedEventType"
minOccurs="1"
                               maxOccurs="1"/>
               </xsd:sequence>
            </xsd:complexType>
         </xsd:element>
         <xsd:element name="finaliseDispenseResponse">
            <xsd:complexType>
               <xsd:sequence>
                  <xsd:element name="response" type="pes:DispenseFinalisedAckType"
minOccurs="1" maxOccurs="1"/>
               </xsd:sequence>
            </xsd:complexType>
         </xsd:element>
         <xsd:element name="initiateDispense">
            <xsd:complexType>
               <xsd:sequence>
                  <xsd:element name="request" type="pes:DispenseInitiatedEventType"
```

```
minOccurs="1"
                                maxOccurs="1"/>
                  </xsd:sequence>
               </xsd:complexType>
            </xsd:element>
            <xsd:element name="initiateDispenseResponse">
               <xsd:complexType>
                  <xsd:sequence>
                     <xsd:element name="response" type="pes:DispenseInitiatedAckType"
minOccurs="1" maxOccurs="1"/>
                  </xsd:sequence>
               </xsd:complexType>
            </xsd:element>
            <xsd:element name="reverseDispense">
               <xsd:complexType>
                  <xsd:sequence>
                     <xsd:element name="request" type="pes:DispenseReversedEventType"
minOccurs="1" maxOccurs="1"/>
                  </xsd:sequence>
               </xsd:complexType>
            </xsd:element>
            <xsd:element name="reverseDispenseResponse">
               <xsd:complexType>
                  <xsd:sequence>
                     <xsd:element name="response" type="pes:DispenseReversedAckType"
minOccurs="1" maxOccurs="1"/>
                  </xsd:sequence>
               </xsd:complexType>
            </xsd:element>
            <xsd:element name="terminateDispense">
               <xsd:complexType>
                  <xsd:sequence>
                     <xsd:element name="request" type="pes:DispenseTerminatedEventType"
minOccurs="1"
                                maxOccurs="1"/>
                  </xsd:sequence>
               </xsd:complexType>
            </xsd:element>
            <xsd:element name="terminateDispenseResponse">
               <xsd:complexType>
                  <xsd:sequence>
                     <xsd:element name="response" type="pes:DispenseTerminatedAckType"
minOccurs="1"
                                maxOccurs="1"/>
                  </xsd:sequence>
               </xsd:complexType>
            </xsd:element>
            <xsd:element name="cancelPrescriptionError" type="tns:CancelPrescriptionErrorType"/>
            <xsd:element name="generalDispenseError" type="tns:GeneralDispenseErrorType"/>
            <xsd:element name="initiateDispenseError" type="tns:InitiateDispenseErrorType"/>
            <xsd:simpleType name="CancelPrescriptionErrorCodeType">
               <xsd:restriction base="xsd:string">
                  <xsd:enumeration value="invalidRepositoryId"/>
                  <xsd:enumeration value="invalidTimestamp"/>
                  <xsd:enumeration value="unknownRetrievalKey"/>
                  <xsd:enumeration value="fullyDispensed"/>
                  <xsd:enumeration value="cancelled"/>
                  <xsd:enumeration value="expired"/>
                  <xsd:enumeration value="notAuthorised"/>
               </xsd:restriction>
            </xsd:simpleType>
            <xsd:simpleType name="GeneralDispenseErrorCodeType">
               <xsd:restriction base="xsd:string">
                  <xsd:enumeration value="invalidRepositoryId"/>
                  <xsd:enumeration value="invalidTimestamp"/>
                  <xsd:enumeration value="unknownRetrievalKey"/>
                  <xsd:enumeration value="duplicateDocumentId"/>
                  <xsd:enumeration value="invalidDocumentType"/>
                  <xsd:enumeration value="unmatchedProcessId"/>
                  <xsd:enumeration value="notAuthorised"/>
               </xsd:restriction>
            </xsd:simpleType>
            <xsd:simpleType name="InitiateDispenseErrorCodeType">
               <xsd:restriction base="xsd:string">
                  <xsd:enumeration value="invalidRepositoryId"/>
                  <xsd:enumeration value="invalidTimestamp"/>
                  <xsd:enumeration value="unknownRetrievalKey"/>
                  <xsd:enumeration value="duplicateProcessId"/>
                  <xsd:enumeration value="fullyDispensed"/>
                  <xsd:enumeration value="cancelled"/>
                  <xsd:enumeration value="expired"/>
                  <xsd:enumeration value="notAuthorised"/>
               </xsd:restriction>
            </xsd:simpleType>
            <xsd:complexType name="CancelPrescriptionErrorType">
```

```
            <xsd:sequence>
               <xsd:element name="errorCode" type="tns:CancelPrescriptionErrorCodeType"
minOccurs="1"
                            maxOccurs="1"/>
               <xsd:element name="message" type="xsd:string" minOccurs="1" maxOccurs="1"/>
            </xsd:sequence>
         </xsd:complexType>
         <xsd:complexType name="GeneralDispenseErrorType">
            <xsd:sequence>
               <xsd:element name="errorCode" type="tns:GeneralDispenseErrorCodeType"
minOccurs="1"
                            maxOccurs="1"/>
               <xsd:element name="message" type="xsd:string" minOccurs="1" maxOccurs="1"/>
            </xsd:sequence>
         </xsd:complexType>
         <xsd:complexType name="InitiateDispenseErrorType">
            <xsd:sequence>
               <xsd:element name="errorCode" type="tns:InitiateDispenseErrorCodeType"
minOccurs="1"
                            maxOccurs="1"/>
               <xsd:element name="message" type="xsd:string" minOccurs="1" maxOccurs="1"/>
            </xsd:sequence>
         </xsd:complexType>
      </xsd:schema>
   </wsdl:types>
   <wsdl:message name="cancelPrescriptionInMsg">
      <wsdl:part name="body" element="tns:cancelPrescription"/>
   </wsdl:message>
   <wsdl:message name="cancelPrescriptionOutMsg">
      <wsdl:part name="body" element="tns:cancelPrescriptionResponse"/>
   </wsdl:message>
   <wsdl:message name="finaliseDispenseInMsg">
      <wsdl:part name="body" element="tns:finaliseDispense"/>
   </wsdl:message>
   <wsdl:message name="finaliseDispenseOutMsg">
      <wsdl:part name="body" element="tns:finaliseDispenseResponse"/>
   </wsdl:message>
   <wsdl:message name="initiateDispenseInMsg">
      <wsdl:part name="body" element="tns:initiateDispense"/>
   </wsdl:message>
   <wsdl:message name="initiateDispenseOutMsg">
      <wsdl:part name="body" element="tns:initiateDispenseResponse"/>
   </wsdl:message>
   <wsdl:message name="reverseDispenseInMsg">
      <wsdl:part name="body" element="tns:reverseDispense"/>
   </wsdl:message>
   <wsdl:message name="reverseDispenseOutMsg">
      <wsdl:part name="body" element="tns:reverseDispenseResponse"/>
   </wsdl:message>
   <wsdl:message name="terminateDispenseInMsg">
      <wsdl:part name="body" element="tns:terminateDispense"/>
   </wsdl:message>
   <wsdl:message name="terminateDispenseOutMsg">
      <wsdl:part name="body" element="tns:terminateDispenseResponse"/>
   </wsdl:message>
   <wsdl:message name="standardErrorMsg">
      <wsdl:part name="fault" element="se:standardError"/>
   </wsdl:message>
   <wsdl:message name="cancelPrescriptionErrorMsg">
      <wsdl:part name="fault" element="tns:cancelPrescriptionError"/>
   </wsdl:message>
   <wsdl:message name="generalEtpErrorMsg">
      <wsdl:part name="fault" element="pes:generalEtpError"/>
   </wsdl:message>
   <wsdl:message name="generalDispenseErrorMsg">
      <wsdl:part name="fault" element="tns:generalDispenseError"/>
   </wsdl:message>
   <wsdl:message name="initiateDispenseErrorMsg">
      <wsdl:part name="fault" element="tns:initiateDispenseError"/>
   </wsdl:message>
   <wsdl:portType name="EDispensing">
      <wsdl:operation name="cancelPrescription">
         <wsdl:input message="tns:cancelPrescriptionInMsg"

wsam:Action="http://ns.electronichealth.net.au/etp/svc/eDispensing/1.1/EDispensing/cancelPresc
riptionRequest"/>
         <wsdl:output message="tns:cancelPrescriptionOutMsg"

wsam:Action="http://ns.electronichealth.net.au/etp/svc/eDispensing/1.1/EDispensing/cancelPresc
riptionResponse"/>
         <wsdl:fault name="standardError" message="tns:standardErrorMsg"

wsam:Action="http://ns.electronichealth.net.au/etp/svc/eDispensing/1.1/EDispensing/cancelPresc
ription/Fault/standardError"/>
         <wsdl:fault name="cancelPrescriptionError" message="tns:cancelPrescriptionErrorMsg"
```

```
wsam:Action="http://ns.electronichealth.net.au/etp/svc/eDispensing/1.1/EDispensing/cancelPresc
ription/Fault/cancelPrescriptionError"/>
         <wsdl:fault name="generalEtpError" message="tns:generalEtpErrorMsg"

wsam:Action="http://ns.electronichealth.net.au/etp/svc/eDispensing/1.1/EDispensing/cancelPresc
ription/Fault/generalEtpError"/>
      </wsdl:operation>
      <wsdl:operation name="finaliseDispense">
         <wsdl:input message="tns:finaliseDispenseInMsg"

wsam:Action="http://ns.electronichealth.net.au/etp/svc/eDispensing/1.1/EDispensing/finaliseDis
penseRequest"/>
         <wsdl:output message="tns:finaliseDispenseOutMsg"

wsam:Action="http://ns.electronichealth.net.au/etp/svc/eDispensing/1.1/EDispensing/finaliseDis
penseResponse"/>
         <wsdl:fault name="standardError" message="tns:standardErrorMsg"

wsam:Action="http://ns.electronichealth.net.au/etp/svc/eDispensing/1.1/EDispensing/finaliseDis
pense/Fault/standardError"/>
         <wsdl:fault name="generalDispenseError" message="tns:generalDispenseErrorMsg"

wsam:Action="http://ns.electronichealth.net.au/etp/svc/eDispensing/1.1/EDispensing/finaliseDis
pense/Fault/generalDispenseError"/>
         <wsdl:fault name="generalEtpError" message="tns:generalEtpErrorMsg"

wsam:Action="http://ns.electronichealth.net.au/etp/svc/eDispensing/1.1/EDispensing/finaliseDis
pense/Fault/generalEtpError"/>
      </wsdl:operation>
      <wsdl:operation name="initiateDispense">
         <wsdl:input message="tns:initiateDispenseInMsg"

wsam:Action="http://ns.electronichealth.net.au/etp/svc/eDispensing/1.1/EDispensing/initiateDis
penseRequest"/>
         <wsdl:output message="tns:initiateDispenseOutMsg"

wsam:Action="http://ns.electronichealth.net.au/etp/svc/eDispensing/1.1/EDispensing/initiateDis
penseResponse"/>
         <wsdl:fault name="standardError" message="tns:standardErrorMsg"

wsam:Action="http://ns.electronichealth.net.au/etp/svc/eDispensing/1.1/EDispensing/initiateDis
pense/Fault/standardError"/>
         <wsdl:fault name="initiateDispenseError" message="tns:initiateDispenseErrorMsg"

wsam:Action="http://ns.electronichealth.net.au/etp/svc/eDispensing/1.1/EDispensing/initiateDis
pense/Fault/initiateDispenseError"/>
         <wsdl:fault name="generalEtpError" message="tns:generalEtpErrorMsg"

wsam:Action="http://ns.electronichealth.net.au/etp/svc/eDispensing/1.1/EDispensing/initiateDis
pense/Fault/generalEtpError"/>
      </wsdl:operation>
      <wsdl:operation name="reverseDispense">
         <wsdl:input message="tns:reverseDispenseInMsg"

wsam:Action="http://ns.electronichealth.net.au/etp/svc/eDispensing/1.1/EDispensing/reverseDisp
enseRequest"/>
         <wsdl:output message="tns:reverseDispenseOutMsg"

wsam:Action="http://ns.electronichealth.net.au/etp/svc/eDispensing/1.1/EDispensing/reverseDisp
enseResponse"/>
         <wsdl:fault name="standardError" message="tns:standardErrorMsg"

wsam:Action="http://ns.electronichealth.net.au/etp/svc/eDispensing/1.1/EDispensing/reverseDisp
ense/Fault/standardError"/>
         <wsdl:fault name="generalDispenseError" message="tns:generalDispenseErrorMsg"

wsam:Action="http://ns.electronichealth.net.au/etp/svc/eDispensing/1.1/EDispensing/reverseDisp
ense/Fault/generalDispenseError"/>
         <wsdl:fault name="generalEtpError" message="tns:generalEtpErrorMsg"

wsam:Action="http://ns.electronichealth.net.au/etp/svc/eDispensing/1.1/EDispensing/reverseDisp
ense/Fault/generalEtpError"/>
      </wsdl:operation>
      <wsdl:operation name="terminateDispense">
         <wsdl:input message="tns:terminateDispenseInMsg"

wsam:Action="http://ns.electronichealth.net.au/etp/svc/eDispensing/1.1/EDispensing/terminateDi
spenseRequest"/>
         <wsdl:output message="tns:terminateDispenseOutMsg"

wsam:Action="http://ns.electronichealth.net.au/etp/svc/eDispensing/1.1/EDispensing/terminateDi
spenseResponse"/>
         <wsdl:fault name="standardError" message="tns:standardErrorMsg"

wsam:Action="http://ns.electronichealth.net.au/etp/svc/eDispensing/1.1/EDispensing/terminateDi
spense/Fault/standardError"/>
         <wsdl:fault name="generalDispenseError" message="tns:generalDispenseErrorMsg"
```

```
wsam:Action="http://ns.electronichealth.net.au/etp/svc/eDispensing/1.1/EDispensing/terminateDi
spense/Fault/generalDispenseError"/>
         <wsdl:fault name="generalEtpError" message="tns:generalEtpErrorMsg"

wsam:Action="http://ns.electronichealth.net.au/etp/svc/eDispensing/1.1/EDispensing/terminateDi
spense/Fault/generalEtpError"/>
      </wsdl:operation>
   </wsdl:portType>
</wsdl:definitions>
```

### A.4.2 TLS Binding

The following WSDL defines the binding of the E-Dispensing interface using a policy for TLS authentication and session privacy.

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap12/"
                  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                  xmlns:wsp="http://www.w3.org/ns/ws-policy"
                  xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata"
                  xmlns:sp="http://docs.oasis-open.org/ws-sx/ws-securitypolicy/200702"
                  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
                  xmlns:tns="http://ns.electronichealth.net.au/etp/svc/eDispensing/1.1"
                  xmlns:pes="http://ns.electronichealth.net.au/etp/xsd/PesDataTypes/1.1"
                  xmlns:se="http://ns.electronichealth.net.au/wsp/xsd/StandardError/2010"
                  name="eDispensing"
                  targetNamespace="http://ns.electronichealth.net.au/etp/svc/eDispensing/1.1">
   <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>
            <!-- Use TLS protocol negotiation instead of <sp:AlgorithmSuite/> -->
         </wsp:Policy>
      </sp:TransportBinding>
   </wsp:Policy>
   <wsdl:import namespace="http://ns.electronichealth.net.au/etp/svc/eDispensing/1.1"
                location="etp-eDispensing-Interface-1.1.wsdl"/>
   <wsdl:binding name="EDispensingBinding" type="tns:EDispensing">
      <wsp:PolicyReference URI="#AddressingPolicy"/>
      <wsp:PolicyReference URI="#TlsPolicy"/>
      <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
      <wsdl:operation name="cancelPrescription">
         <soap:operation style="document" soapActionRequired="false"/>
         <wsdl:input>
            <soap:body use="literal"/>
         </wsdl:input>
         <wsdl:output>
            <soap:body use="literal"/>
         </wsdl:output>
         <wsdl:fault name="standardError">
            <soap:fault name="standardError" use="literal"/>
         </wsdl:fault>
         <wsdl:fault name="cancelPrescriptionError">
            <soap:fault name="cancelPrescriptionError" use="literal"/>
         </wsdl:fault>
         <wsdl:fault name="generalEtpError">
            <soap:fault name="generalEtpError" use="literal"/>
         </wsdl:fault>
      </wsdl:operation>
      <wsdl:operation name="finaliseDispense">
         <soap:operation style="document" soapActionRequired="false"/>
         <wsdl:input>
            <soap:body use="literal"/>
         </wsdl:input>
         <wsdl:output>
            <soap:body use="literal"/>
         </wsdl:output>
         <wsdl:fault name="standardError">
            <soap:fault name="standardError" use="literal"/>
```

```
            </wsdl:fault>
            <wsdl:fault name="generalDispenseError">
               <soap:fault name="generalDispenseError" use="literal"/>
            </wsdl:fault>
            <wsdl:fault name="generalEtpError">
               <soap:fault name="generalEtpError" use="literal"/>
            </wsdl:fault>
         </wsdl:operation>
         <wsdl:operation name="initiateDispense">
            <soap:operation style="document" soapActionRequired="false"/>
            <wsdl:input>
               <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output>
               <soap:body use="literal"/>
            </wsdl:output>
            <wsdl:fault name="standardError">
               <soap:fault name="standardError" use="literal"/>
            </wsdl:fault>
            <wsdl:fault name="initiateDispenseError">
               <soap:fault name="initiateDispenseError" use="literal"/>
            </wsdl:fault>
            <wsdl:fault name="generalEtpError">
               <soap:fault name="generalEtpError" use="literal"/>
            </wsdl:fault>
         </wsdl:operation>
         <wsdl:operation name="reverseDispense">
            <soap:operation style="document" soapActionRequired="false"/>
            <wsdl:input>
               <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output>
               <soap:body use="literal"/>
            </wsdl:output>
            <wsdl:fault name="standardError">
               <soap:fault name="standardError" use="literal"/>
            </wsdl:fault>
            <wsdl:fault name="generalDispenseError">
               <soap:fault name="generalDispenseError" use="literal"/>
            </wsdl:fault>
            <wsdl:fault name="generalEtpError">
               <soap:fault name="generalEtpError" use="literal"/>
            </wsdl:fault>
         </wsdl:operation>
         <wsdl:operation name="terminateDispense">
            <soap:operation style="document" soapActionRequired="false"/>
            <wsdl:input>
               <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output>
               <soap:body use="literal"/>
            </wsdl:output>
            <wsdl:fault name="standardError">
               <soap:fault name="standardError" use="literal"/>
            </wsdl:fault>
            <wsdl:fault name="generalDispenseError">
               <soap:fault name="generalDispenseError" use="literal"/>
            </wsdl:fault>
            <wsdl:fault name="generalEtpError">
               <soap:fault name="generalEtpError" use="literal"/>
            </wsdl:fault>
         </wsdl:operation>
      </wsdl:binding>
      <wsdl:service name="EDispensingService">
         <wsdl:port name="EDispensing" binding="tns:EDispensingBinding">
            <soap:address location="http://dummy.example.com"/>
         </wsdl:port>
      </wsdl:service>
   </wsdl:definitions>
```