nehta

Solution Specification

Electronic Transfer of Prescription 1.1

Version 1.1 - 17 December 2010

Final

National E-Health Transition Authority

National E-Health Transition Authority Ltd

Level 25

56 Pitt Street

Sydney, NSW, 2000

Australia.

www.nehta.gov.au

Disclaimer

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

Security

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

Copyright © 2010, NEHTA.

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

Document Information

Change History

Version	Date	Comments
1.0	6 September 2010	Draft for consultation
1.1	17 December 2010	Final

Document Authorisation

Name	Title	Signature
Stephen Johnston	Head of Product and Solution Development	
Dr. Leonie Katekar	Director, Clinical Unit and Clinical Lead	
Toby Mathieson	Program Manager, e-Medication Management	

Table of contents

Preface

Document Purpose

This document defines specifications for Electronic Transfer of Prescription (ETP) services that are proposed as the basis for new national standards.

Intended Audience

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

- Jurisdictions 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 which makes use of the UML2.3 standard [UML2010]. Familiarity with UML and service oriented architecture concepts and patterns is assumed.

Document Map

The following diagram represents the relationship between this document and others within the Electronic Transfer of Prescription Release 1.1 package.
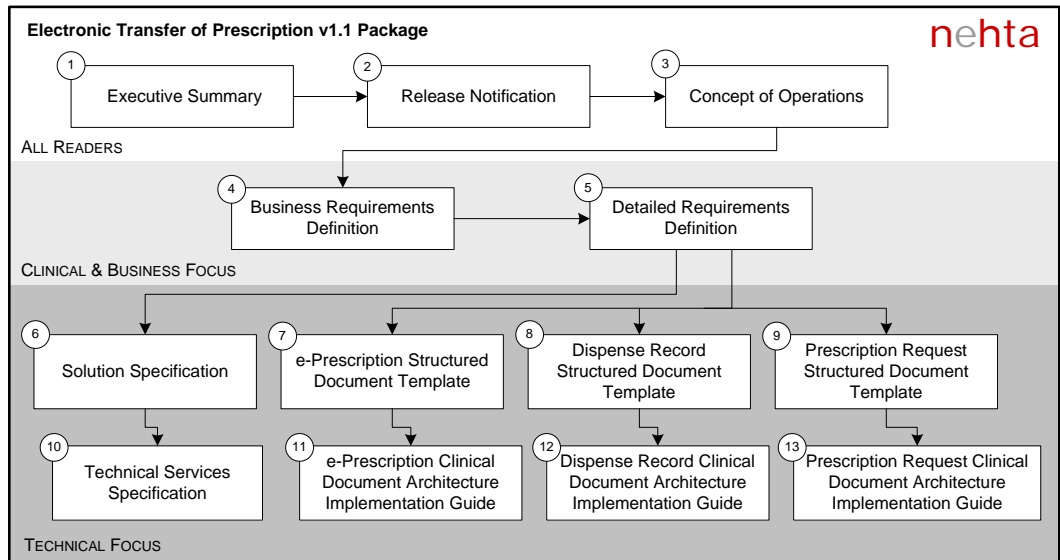
Figure 1   ETP - Document Map

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 lists of all referenced documents, see the References section at the end of the document.

1 Introduction

1.1 Scope of document

This document specifies an Electronic Transfer of Prescriptions (ETP) solution for the generation and transfer of electronic prescriptions and associated dispensing records between participating prescribers and dispensers.
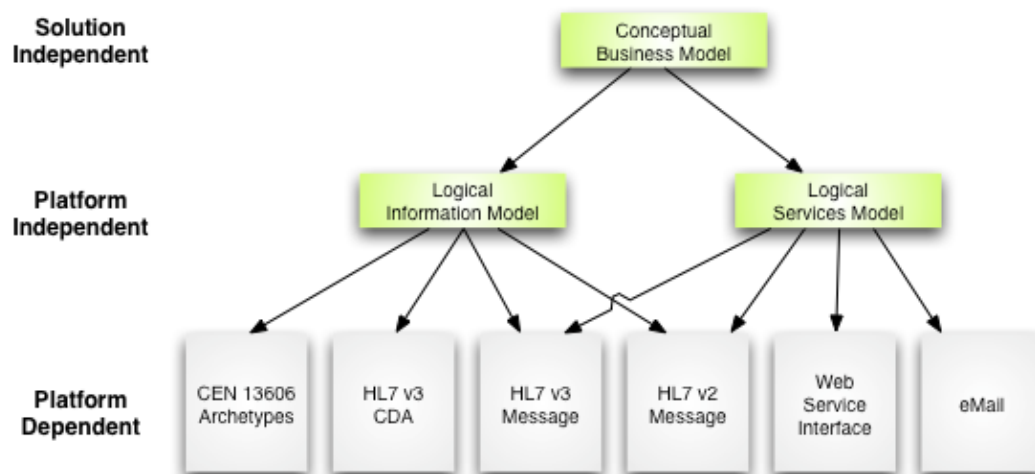
Figure 2: Layering of Specifications

The focus of this document is the specification of a set of ETP services. These services are specified in terms of the required behaviour of the participating systems and the information that is communicated between them.

This document is to be read in conjunction with the Structured Document Templates (SDTs) which are also provided as part of this ETP package (i.e. [ETP-EP_SDT2010], [ETP-ED_SDT2010] and [ETP-PR_SDT2010]). These SDTs specify the clinical information that is communicated between participants using the ETP services.

The combination of this document, plus these SDTs, thereby represents a complete "platform independent" logical service specification for ETP (see Figure 2).

The ETP package also includes an ETP Technical Services Specification [ETP-TSS2010] which provides a "platform dependent" (see Figure 2) specification that binds the logical service specifications to a particular implementation "platform" by specifying particular web services interfaces.

Separating platform independent from platform dependent specifications thereby allows the fundamental ETP service characteristics, on which the key benefits depend, to be defined independently of particular implementation choices.

Section 2 specifies a set of ETP business services. Organisations and individuals who consume and provide these services will be required to conform to policies, legislation, regulations, and practices, however the specification of these is outside the scope of this "logical service specification".

Section 3 is an information view of the ETP business services (as a set of data types) including references to the SDTs in which clinical information is specified (as a logical information model).

Section 4 is the technical view of the ETP business services (i.e. it specifies ETP technical services and provides conformance points for the IT systems that participate in these ETP technical services).

1.2   Key Dependencies

These ETP specifications require the various participants in the ETP services to make appropriate use of several National Infrastructure Services, namely:

- Identifier Services

  This logical service specification requires the use of national Healthcare Identifiers (HI) for healthcare organisations, healthcare provider individuals (i.e. clinicians) and individuals (i.e. subjects of care for whom medications are prescribed). Participants in ETP services are therefore required to make use of the following HI services:

  - HPI-O Services: support identifiers for healthcare provider organisations
  - HPI-I Services: support identifiers for healthcare provider individuals
  - IHI Services: support identifiers for individuals.

- Public Key Infrastructure and Authentication Services

  - In accordance with these specifications, participants in ETP services require the existence of an appropriate Public Key Infrastructure (PKI) which allows for healthcare individuals and organisations to be authenticated
  - When fully defined and available, NASH will serve as the common National Infrastructure Service providing both a PKI, and the identity credentials based upon it.

- National Clinical Terminology and Information Service

  - The use of AMT clinical terminologies is specified in the ETP SDTs.

2 Business View

The requirements addressed by this solution specification are documented in the Business Requirements Definition [ETP-BR2010] and Detailed Requirements Definition [ETP-DR2010] documents that are also part of the ETP package.

This section specifies the ETP business services in terms of the roles played by the various IT systems and parties[1] that participate in these services, the relationships and constraints that apply to each role, and the service functions provided.

The policies, legislation, regulations, and practices that constrain the various parties in their participation in the ETP business services are not specified here[2]. Rather, this business view is intended to provide a non-normative description of the business context and to define key concepts and terms that are then used to express the technical specifications and conformance points that are contained in sections 3 and 4.

2.1 ETP Roles

The roles fulfilled by the various types of parties and IT systems that either consume or provide ETP services are shown in Figure 3.
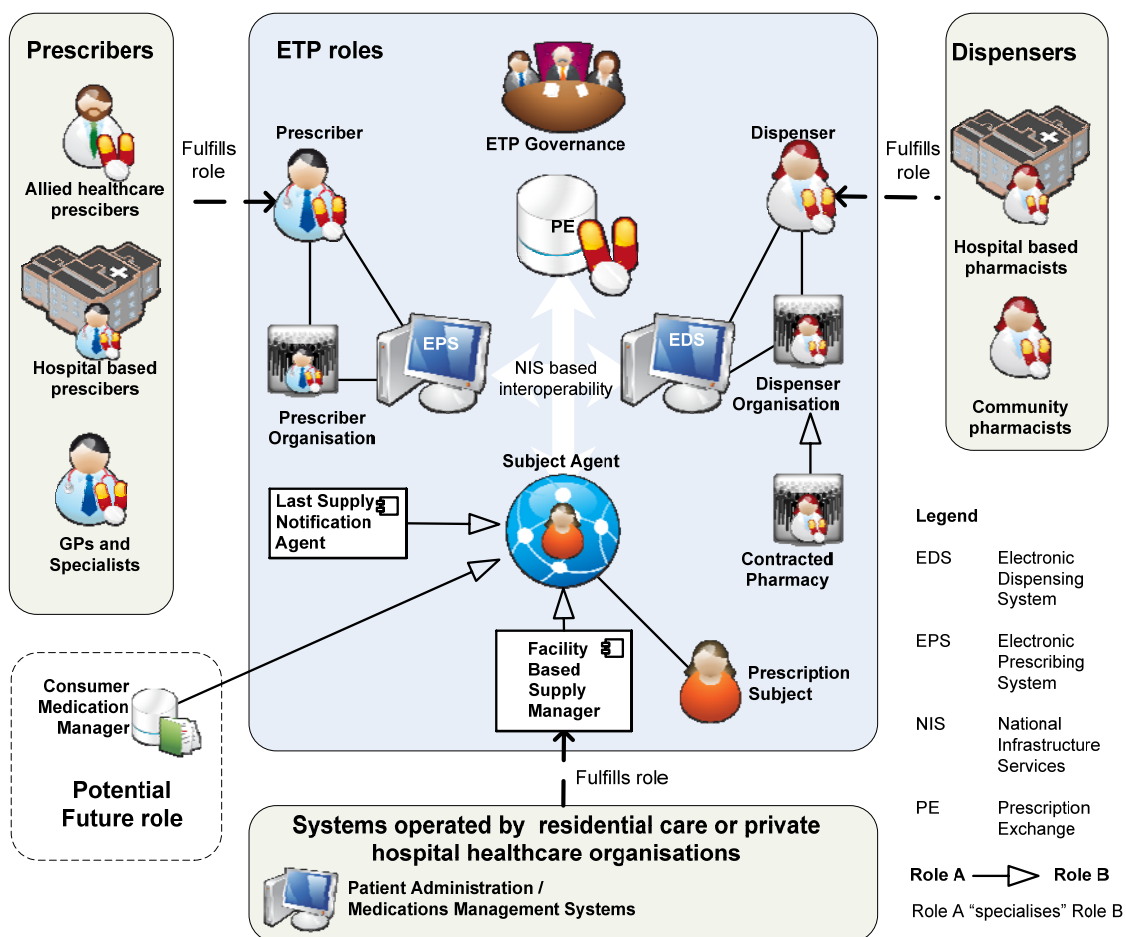
Figure 3: ETP roles and typical participating entities

---

[1]   The concept of "party" models a person or organisation that fulfils one or more roles, according to the constraints that apply to those roles.

[2]   Other documents that address conformance and compliance at the business level will be subsequently developed.

The roles are:

- **ETP Governance**

  Defines the policies that govern the behaviour of all the participants in the ETP community.

- **Prescription Subject**

  The *Prescription Subject* is the individual for whom a prescribed medication is intended.

- **Prescriber**

  A *Prescriber* is an individual who provides healthcare and who creates prescriptions in accordance with all relevant legislative, regulatory and professional requirements.

  A *Prescriber* shall participate in ETP services as the representative of an identified *Prescriber Organisation.*

  The role of *Prescriber* may be fulfilled by parties that are either hospital or community based healthcare provider individuals.

  The *Prescriber* chooses either to create a new prescription as an electronic prescription that is sent to a *Prescription Exchange* or as a paper prescription that is not (and is therefore not the subject of this specification).

- **Prescriber Organisation**

  A *Prescriber Organisation* is a healthcare organisation which is represented by one or more *Prescribers.*

- **Dispenser**

  A *Dispenser* is an individual who provides healthcare and who dispenses a prescription in accordance with all relevant legislative, regulatory and professional requirements.

  A *Dispenser* shall participate in ETP services as the representative of an identified *Dispenser Organisation*

  This role may be fulfilled by pharmacists and other authorised healthcare providers that are either hospital or community based.

- **Dispenser Organisation**

  A *Dispenser Organisation* is a healthcare organisation which is represented by one or more *Dispensers.*

- **Contracted Pharmacy**

  A *Contracted Pharmacy* is a *Dispenser Organisation* that dispenses and supplies medications under a contract to a residential care facility, private hospital or other similar healthcare facility

- **Electronic Prescribing System**

  An *Electronic Prescribing System* is a component of a clinical information system that is used by a *Prescriber* to prescribe medications.

  An *Electronic Prescribing System* is operated by (or on behalf of) an identified *Prescriber Organisation.*

- **Electronic Dispensing System**

  An *Electronic Dispensing System* is a component of a clinical information system that is used by a *Dispenser* to dispense medications.

  An *Electronic Dispensing System* is operated by (or on behalf of) an identified *Dispenser Organisation.*

- **Prescription Exchange**

  A *Prescription Exchange* is a system that manages the electronic documents and records that support the ability to prescribe and dispense medications. A *Prescription Exchange* therefore contains, for each prescription created as an electronic document, the electronic prescription document plus its associated electronic dispense documents. A *Prescription Exchange* also reflects the "dispensing state" of each of these prescriptions.

  A *Prescription Exchange* makes the electronic prescription and dispense documents available to *Dispensers* based on its view of the dispensing state, while preventing access to prescriptions that have been cancelled, are expired, or have been fully dispensed.

- **Subject Agent**

  A *Subject Agent* is a software agent that acts on behalf of a *Prescription Subject*. This role provides for the ETP services architecture to be extended (under policies governed by the *ETP Governance* role) by defining new roles as specialisations of *Subject Agent*. Each specialisation represents a constrained ability to act on behalf of the *Prescription Subject*. Currently the *Facility Based Supply Manager* and *Last Supply Notification Agent* are defined. A potential future role of *Consumer Medication Services Manager* is shown in Figure 3; this role is currently undefined - it is expected to be defined in the context of the Personally Controlled Electronic Healthcare Record (PCEHR).

- **Facility Based Supply Manager**

  A *Facility Based Supply Manager* is a *Subject Agent* that acts on behalf of a *Prescription Subject* who is under the care of a healthcare facility that uses a *Contracted Pharmacy*. A *Facility Based Supply Manager* receives notifications generated by an *Electronic Prescribing System* and, provided that the identified *Prescription Subject* has consented, forwards these on to the *Contracted Pharmacy* for dispensing and supply.

- **Last Supply Notification Agent**

  A *Last Supply Notification Agent* is a *Subject Agent* that acts on behalf of a *Prescription Subject* and receives notifications generated by an *Electronic Dispensing System* when the last supply of a medication for that *Prescription Subject* has been dispensed.

2.2   ETP Business Services

Figure 4 shows the ETP services architecture in terms of the services, roles, systems and system users.
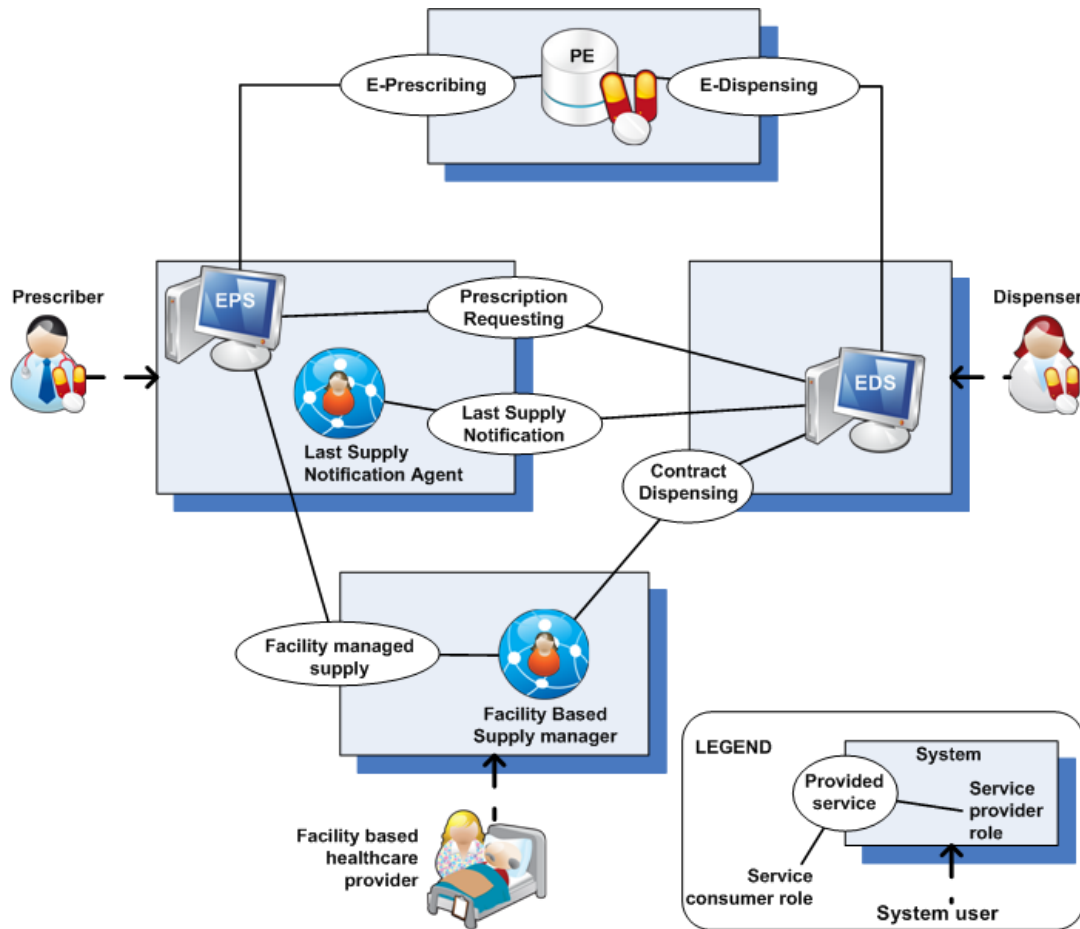
Figure 4: ETP services architecture

ETP defines six business services:

- **E-Prescribing Service:** allows a *Prescriber* to participate in the ETP solution and thereby make their prescriptions available within a *Prescription Exchange* for dispensing.

- **E-Dispensing Service:** allows a *Dispenser* to participate in the ETP solution and thereby download electronic prescriptions (and prior dispense records for a prescription) from a *Prescription Exchange* and subsequently upload resulting electronic dispense records linked to the prescriptions.

- **Prescription Requesting Service:** allows a *Dispenser* to request that an identified *Prescriber* create an electronic prescription that confirms an instruction previously given by that *Prescriber* to that *Dispenser* (in a form other than a prescription). Also provides for the *Prescriber* to respond to these requests by providing the *Dispenser* with an electronic prescription notification.

- **Facility Managed Supply Service:** allows a *Prescriber* to notify a healthcare facility (via their *Facility Based Supply Manager*) when a new prescription has been sent to a *Prescription Exchange* for a *Prescription Subject* who is resident in that facility.

- **Last Supply Notification Service:** allows a *Dispenser* to notify an authorised party (e.g. a Prescriber*),* of the last supply of a medication.

- **Contract Dispensing:** allows a healthcare facility (via their *Facility Based Supply Manager*) to send an electronic prescription notification to a *Contracted Pharmacy*. This will enable a *Dispenser* representing the *Contracted Pharmacy* to subsequently dispense and supply medications.

These ETP services are listed in Table 1, including a high level description of the service functions provided by each service, and the entities that either provide or consume those services.

SERVICE	CONSUMER ROLE	PROVIDER ROLE	SERVICE FUNCTION	
			ID	NAME AND DESCRIPTION
E-Prescribing Service	*Electronic Prescribing System*	*Prescription Exchange*	EP1	**Prescribe**: Publish electronic prescription document.
			EP2	**Cancel Prescription**: Block access by *Electronic Dispensing Systems* to a previously published electronic prescription document.
			EP3	**Retrieve Prescription:** Get a copy of a previously published prescription document.
E-Dispensing Service	*Electronic Dispensing System*	*Prescription Exchange*	ED1	**Initiate Dispense Process:** Initiate a dispensing process and retrieve documents required for dispensing.
			ED2	**Terminate Dispense Process:** Record the conclusion of an initiated dispensing process that was abandoned and did not result in any items being dispensed.
			ED3	**Finalise Dispense process**: Record the conclusion of an initiated dispensing process that resulted in the dispensing of an item, and provide a record of the process in the form of an electronic dispense document.
			ED4	**Reverse Dispense Process**: Reverse the effect of a previously completed dispensing process, withdrawing a previously published electronic dispense document.
			ED5	**Cancel Prescription** Block access by *Electronic Dispensing Systems* to a previously published electronic prescription document.

SERVICE	CONSUMER ROLE	PROVIDER ROLE	SERVICE FUNCTION	
			ID	NAME AND DESCRIPTION
Prescription Requesting Service	*Electronic Dispensing System*	*Electronic Prescribing System*	PR1	**Deliver Prescription Request**: Deliver to an identified *Prescriber* a request to create an electronic prescription corresponding to a previous issued *Prescriber* instruction.
			PR2	**Deliver Prescription Notification**: Deliver to an identified *Dispenser Organisation* an electronic prescription notification; This specification requires that an *Electronic Prescribing System* shall only invoke this service function in response to a prior request received via the Prescription Requesting Service.
Facility Managed Supply Service	*Electronic Prescribing System*	*Facility Based Supply Manager*	FM1	**Deliver Prescription Notification:** Deliver to an identified *Facility Based Supply Manager* an electronic notification that indicates the successful invocation of the Prescribe service function of the E-Prescribing service.
Last Supply Notification Service	*Electronic Dispensing System*	*Last Supply Notification Agent*	LS1	**Deliver Last Supply Notification:** Deliver to an identified *Last Supply Notification Agent* an electronic notification that indicates the successful invocation of the Finalise Dispense Process service function of the E-Dispensing service for the last dispensing of a prescription. Note that this specification does not specify: • The means by which *Last Supply Notification Agents* are registered with *Electronic Dispensing Systems;* or • For which prescriptions these notifications are generated.

SERVICE	CONSUMER ROLE	PROVIDER ROLE	SERVICE FUNCTION	
			ID	NAME AND DESCRIPTION
Contract Dispensing	*Facility Based Supply Manager*	*Electronic Dispensing System*	CD1	**Deliver Prescription Notification:** Deliver, to an identified *Contracted Pharmacy,* a notification that indicates the successful invocation of the Prescribe service function of the E-Prescribing service. Note that the execution of this service function does not constitute an instruction to dispense; it simply informs a *Contracted Pharmacy* that a new prescription is now available in a *Prescription Exchange*. The *Contracted Pharmacy* determines when to dispense and supply medications in accordance with their contract with the healthcare facility that operates the *Facility Based Supply Manager* and with any instructions provided by the *Prescriber*.

Table 1: ETP Services and functions

2.3 Security Overview

2.3.1 Introduction

This section summarises the common approach to addressing security that underpins the ETP business services. This section is descriptive (i.e. non-normative) and is primarily intended to introduce the concept of a Document Access Key (*DAK*) which is a novel feature fundamental to the security of the ETP services.

In the following the term *Secured Clinical Document* is used to refer to an object that comprises a clinical payload (i.e. personal healthcare information about a *Prescription Subject*) and a set of metadata (this is information that is required to support the management of the clinical payload but is not itself personal information). The types of *Secured Clinical Documents* that are relevant to the ETP services are electronic prescriptions and electronic dispense records. These are specified in Section 3.

2.3.2 Service participation management

Every ETP service function is requested by a representative of an identified organisation.

For the *E-Prescribing* and *E-Dispensing* services, the requesting organisation must have previously entered into a participation agreement with the organisation that provides the service. Participation agreements define the legal and commercial context in which the ETP services are provided.

The business and technical services offered by ETP service providers to manage and record the participation of their client organisations is outside the

scope of the NEHTA ETP specifications. This document therefore does not specify participation management services.

2.3.3   Authentication

Service requests and responses are associated with identity credentials which assert the identity of the organisation that is responsible for that request or response (i.e. service consumer organisations and service provider organisations mutually authenticate each other every time they communicate).

Organisations that, under the Healthcare Identifiers Act 2010, are healthcare provider organisations or contracted service providers are required to assert their identity using identifiers issued and managed by the HI services.

2.3.4   Authorisation

Service consumers are required by service providers to possess one or more of the following types of credential (depending on the service):

1. **Identity credentials**

   As described in the previous section, every request or response must be associated[3] with an identity credential that asserts an organisational identity (e.g. a HPI-O).

2. **Retrieval Key**

   For ETP service functions that access *Secured Clinical Documents* held by a *Prescription Exchange,* the service consumer must have possession of a *Retrieval Key* that is specific to the prescription being referenced. *Retrieval Keys* are derived from *DAKs* as described in section 2.3.7.

2.3.5   Confidentiality of data in transit

The mechanisms used to ensure the confidentiality of data in transit between ETP service participants are specific to the implementation "platform". These are therefore specified in the ETP Technical Services Specification [ETP-TSS2010].

2.3.6   Confidentiality of data at rest

The clinical payload component of a *Secured Clinical Document* that is managed by a *Prescription Exchange* is encrypted using a symmetric key that is unique to the prescription.

This symmetric key is derived from the same *DAK* whose possession authorises a service consumer to access that *Secured Clinical Document* as described in the next section.

2.3.7   Document Access Key

As indicated above in sections 2.3.4 and 2.3.6, *DAKs* play two roles in ensuring the security of *Secured Clinical Documents*: authorising access to them via services provided by a *Prescription Exchange,* and encrypting and decrypting their clinical payloads.

A *DAK* is associated with a *Secured Clinical Document* by the system that creates that *Secured Clinical Document*. The same *DAK* may be associated

---

[3]   Authentication may be a feature of the transport layer (e.g. as in mutually authenticated TLS) or it may be performed at the service layer by associating each service request and response message with an authentication credential.

with multiple *Secured Clinical Documents*. In ETP[4] all the *Secured Clinical Documents* for one prescription are associated with the same *DAK* but each prescription is associated with a different *DAK*.

A *DAK* is a mechanism that is used to efficiently communicate a set of cryptographic keys that are be used for different purposes within an overall set of controls that ensure the availability, confidentially and integrity of *Secured Clinical Documents*. Efficiency here means that a complete set of cryptographic keys can be communicated by simply communicating one "parent key" from which all the required keys are then derived using a published algorithm. The mechanism for doing this is described below:

- The *DAK* contains a random value component that has sufficient entropy such that the only practical way for a party to have knowledge of which *DAK* has been associated with a given *Secured Clinical Document* is for such knowledge to be explicitly disclosed to that party[5]

- The derived cryptographic keys are other random values that are derived by using the *DAK* as the input into a one-way function[6] – another input to that function being a well known string that identifies each type of key within the set

DAKs are used by ETP service participants to derive two types of key[7]:

- *Retrieval Key*: a random value that, once qualified by an identifier for a service provider, forms a reference to a *Secured Clinical Document* that is available from the identified service provider.

  A *Retrieval Key* that is qualified by a service provider identifier is termed a *Qualified Retrieval Key*.

  A *Qualified Retrieval Key* – being a reference to a set of *Secured Clinical Documents* (but a reference that must be explicitly disclosed to a party in order for that party to access those *Secured Clinical Documents*) – is therefore revealed as being a capability[8] that acts as a bearer credential. Therefore, knowledge of a *Qualified Retrieval Key* grants the holder the right to access the *Secured Clinical Documents* associated with that *Qualified Retrieval Key*.

  This describes the role of the *DAK* in access authorisation.

- *Cipher Key:* a random value that is used as a symmetric key to encrypt and decrypt the clinical payload component of all the *Secured Clinical Documents* that are associated with a *DAK*.

  This describes the role of the *DAK* in ensuring the confidentially of the data at rest within a *Prescription Exchange*.

Note that it is a feature of the key derivation algorithm that a *Cipher Key* cannot be derived from a *Retrieval Key* (or visa versa). Also, the *DAK* from which they are both derived, cannot be derived from either the *Retrieval Key* or *Cipher Key*. This makes it possible to disclose the *Retrieval Key* to a *Prescription Exchange* without also disclosing the *Cipher Key* or the *DAK.* The result is that the clinical payloads of *Secured Clinical Documents* are not

---

[4] The DAK mechanism is not specific to ETP and can be used as part of the security controls used in other services.

[5] The random value component of a *DAK* is generated on any of the common operating system platforms using a standard system interface to access a Deterministic Random Bit Generator (DBRG)

[6] A cryptographically secure pseudo-random function is used as the one-way function

[7] An *integrity key,* used to calculate a cryptographically secure Message Authentication Code (MAC) that ensures the integrity of the *Clinical Documents* associated with a *DAK* could also be derived but, in ETP, *Secured Clinical Documents* are signed by their authors, so *integrity keys* are not required.

[8] A capability being a communicable, unforgeable token of authority that is at once both a reference to a resource plus an associated set of access rights for that resource.

visible to the operator of a *Prescription Exchange.* This means that the *Prescription Subject* is not required to consent to the operator of a *Prescription Exchange* having access to their private information.

The *Cipher Key* and *Retrieval Key* are transferred indirectly between ETP service participants by transferring the *DAK* from which they are derived. The *DAK* is represented as part of a prescription notification (i.e. on a paper prescription notification the *DAK* is encoded as a linear barcode, in an electronic prescription notification the *DAK* is encoded as a text string).

Figure 5 shows how a *Secured Clinical Document* that represents an electronic prescription is encrypted by an *Electronic Prescribing System* and then subsequently decrypted by an *Electronic Dispensing System*.
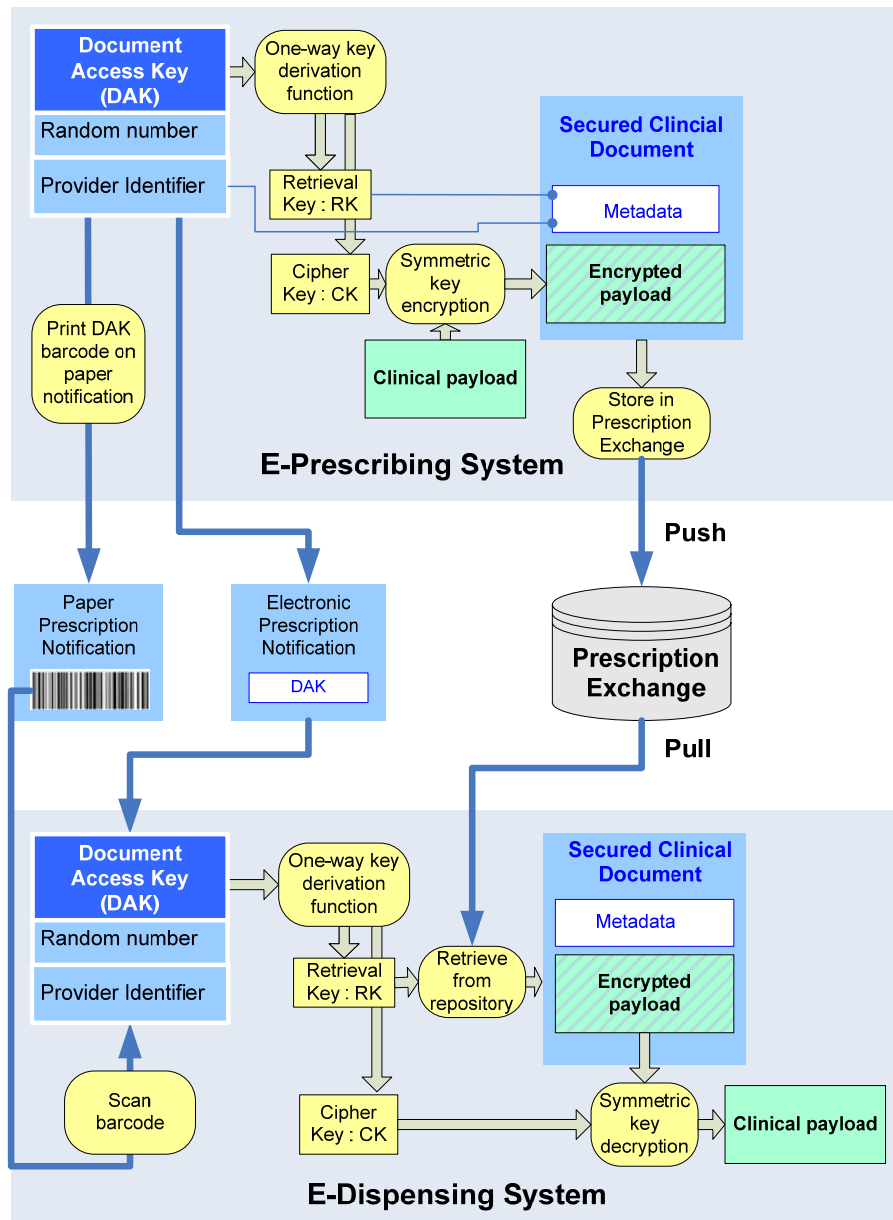
Figure 5: DAK usage

Similarly, after dispensing an item for a given prescription, an *Electronic Dispensing System* generates a *Secured Clinical Document* that represents an electronic dispense record and the *Electronic Dispensing System* sets the *Retrieval Key* of this document to the same as was used for the electronic prescription. The *Electronic Dispensing System* also encrypts the electronic dispense record with the same *Cipher Key* as was used to encrypt the electronic prescription.

In addition to transferring the *Cipher Key* via the *DAK* as described above, the *Cipher Key* may also be transferred between ETP service participants by applying the approach defined in [ATS5821—2010] to encrypt the *Cipher Key* using the public key of an identified *Secured Clinical Document* recipient. In this case the encrypted *Cipher Key* is carried as part of the container which wraps the encrypted clinical payload within the *Secured Clinical Document*. In this case the *Secured Clinical Document* itself carries the *Cipher Key* (encrypted) and any recipient of that *Secured Clinical Document* that was thereby identified by the author of that *Secured Clinical Document* can use their private key to extract the *Cipher Key*, and then to decrypt the clinical payload component.

2.3.8   Audit

Auditing is a key aspect of the security of ETP services – specific conformance points are provided in section 4; the following descriptive (i.e. non-normative) summary is provided:

- All participants are required to store in stable storage an audit record for each request, response or notification that is sent or received by that participant

- At a minimum, for each request received, PES service providers must record the:
    – Identity of the requesting organisation, and the individual who represents that organisation making the request
    – The *Retrieval Key* of the document(s) being accessed
    – Date and time
    – Name of the operation.

- At a minimum, for each response sent, PES service providers must record the:
    – Requesting organisation
    – *Retrieval Key* of the document(s) that was accessed
    – Date and time
    – Name of the operation
    – Result (i.e. success or the specific fault condition).

2.3.9   Document Attestation

Secured Clinical Documents in ETP are digitally signed by their authors – using either an organisational or an individual PKI certificate[9].

Where a *Secured Clinical Document* needs to be treated as the legal equivalent of an existing document that must be signed (e.g. an electronic prescription) then the expectation is that an individual certificate will be required; however, key stakeholders have yet to determine the appropriate form of the digital signature.

2.4   State Management

2.4.1   Architectural Approach

The designs of the ETP business services are based on the principle that clinical processes must continue even when IT systems that support those

---

[9]   The PKI implementation details are left to ETP Technical Services Specification [ETP-TSS2010] to define.

services are not available. Consequently the state of "real world" prescribing and dispensing processes cannot be perfectly aligned with the model of the state of those processes that is reflected by the various systems:

- Events recorded by one system may be communicated to other systems with significant latency or (in highly unusual failure modes) not at all

- Systems can only impose policies that are based on their local knowledge of the state

2.4.2   Dispensing State

A *Prescription Exchange* ceases to make a prescription available to an *Electronic Dispensing System* when that prescription is known to be fully dispensed, cancelled or expired. In order to implement this functionality both the *E-Prescribing* and *E-Dispensing* service providers must access a shared representation of the dispensing state of each prescription.

The *dispensing state* of a prescription is one of:

- **Available to Dispense:** this is the state required for an *E-Dispensing* service provider to allow an *Electronic Dispensing System* to retrieve the electronic prescription in order to dispense it (i.e. to successfully request the "Initiate Dispense Process" service function, such as ED1 in Table 1).

- **Cancelled:** the *Prescriber* has determined that no more dispensing of the prescription should be allowed

- **Fully dispensed:** all items authorised for dispensing have been dispensed

- **Expired:** the prescription is no longer valid because a time limit, set by the *Prescriber*, has expired.

This solution specifies that one role (the *Prescription Exchange*) is the provider of both *E-Prescribing* and *E-Dispensing* services and that all documents for one prescription are managed by one *Prescription Exchange.* The *dispensing state* of a prescription is therefore reflected by one *Prescription Exchange.*

2.4.3   Prescription Exchange Service (PES) Events

A *dispensing state* transition is triggered within a *Prescription Exchange* as a result of the occurrence of a "clinical event" that is managed by an *Electronic Prescribing System* or an *Electronic Dispensing System* – but only after knowledge of that event is received by the *Prescription Exchange.*

The occurrence of an event is communicated to the *Prescription Exchange* by the successful execution of a service function provided by the *Prescription Exchange*.

The term *PES Event* is defined as an event that is communicated to a *Prescription Exchange* and may result in some change in the state of a prescription that is reflected by a *Prescription Exchange*.

The following *PES Events* may trigger a *Prescription Exchange* to reflect a change in its *dispensing state*:

- **Prescription Available**: this *PES Event* is communicated by the successful execution of the *E-Prescribing* "Prescribe" service function (EP1 in Table 1).

- **Prescription Cancelled:** this *PES Event* is communicated by the successful execution of the *E-Prescribing* "Prescription cancellation" service function (EP2 in Table 1) or the *E-Dispensing* "Prescription Cancellation" service function (ED5 in Table 1)

- **Dispense Finalised:** this *PES Event* is communicated by the successful execution of the *E-Dispensing* "Finalise Dispense process" service function (ED3 in Table 1).

- **Dispense Reversed**: this *PES Event* is communicated by the successful execution of the *E-Dispensing* "Reverse dispense process" service function (ED4 in Table 1).
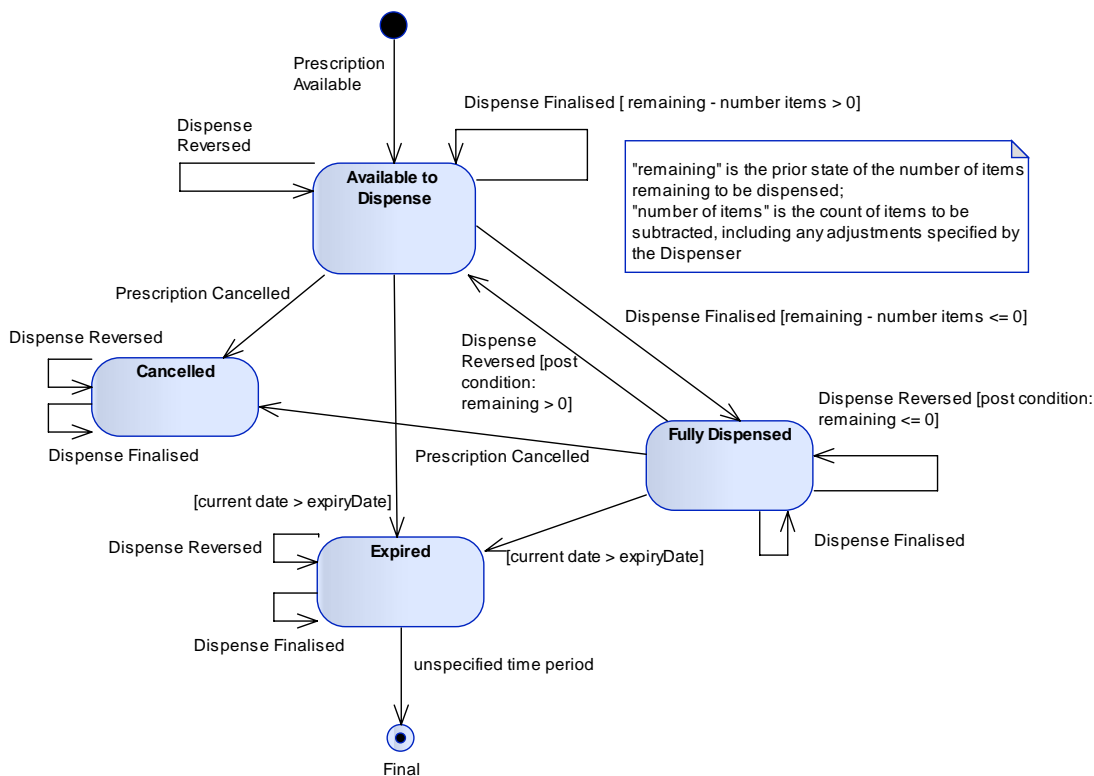
Figure 6: Dispensing States and Transitions

Two additional *PES Events* are defined that, while they do not change the *dispensing state*, nevertheless change another aspect of the state of a prescription that is reflected by a *Prescription Exchange*. These two additional *PES Events* are:

- **Dispense Initiated:** this *PES Event* is communicated by the successful execution of the *E-Dispensing* "Initiate dispense process" service function (ED1 in Table 1).

- **Dispense Terminated:** this *PES Event* is communicated by the successful execution of the *E-Dispensing* "Terminate dispense process" service function (ED2 in Table 1).

These two *PES Events* allow the *Prescription Exchange* to keep track of cases where a dispensing process was initiated but its subsequent outcome has not yet been successfully communicated to the *Prescription Exchange*. The *Prescription Exchange* does not prevent a new dispensing process from being initiated even in these circumstances (i.e. it may simply be due to a communication or system failure and therefore the principle of not preventing clinical processes from proceeding in cases of system failure would apply). It does, however, keep a list of "current interests" in prescriptions by *Dispenser Organisations*. A *Current Interest* is created when a "Dispense Initiated" *PES Event* is signalled and deleted when a corresponding "Dispense Finalised" or "Dispense Terminated" *PES Event* is signalled. *Dispensers* are made aware of any *Current Interests* when they initiate a dispensing process. If any *Current Interests* exist then the *Dispenser* is expected to attempt to determine the state of any prior dispensing process(es) before either choosing to continue

the dispensing process which they initiated or to abandon it. The precise obligations of *Dispensers* in this regard are not defined in this document.

3   Information View

This chapter defines a set of data types that specify the information that is transferred between ETP service participants; The data types specified herein are referenced by the conformance points contained in section 4. This section is therefore normative in the sense that data types referenced in section 4 shall comply with the data type specification given in this section (3).

3.1   Common Data Types

This section defines data types that are used commonly in other data types
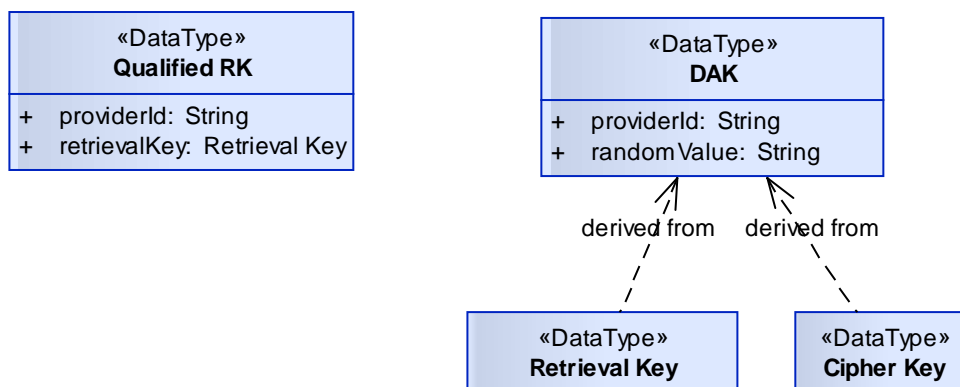
Figure 7: DAK and related data types

3.1.1   Document Access Key

The concept of the *Document Access Key (DAK)* is described in section 2.3.7. *DAKs* are used to efficiently communicate a set of cryptographic keys that are used to secure clinical documents.

A `DAK` data type represents the *DAK* concept; it comprises two elements:

- `providerId`: a unique identifier for the service provider that operates the *Prescription Exchange* that will manage all documents associated with the *DAK*.

- `randomValue`: a random value with sufficient entropy to ensure that it is infeasible for an entity to obtain a *DAK* unless it was explicitly provided to that entity.

3.1.2   Keys derived from a Document Access Key

A `DAK` is used to derive two cryptographic keys:

- `Retrieval Key`*:* a random value that is unique to a set of related clinical documents that are managed by the same service provider.

- `Cipher Key`*:* a symmetric key (a random value) that is used to encrypt and decrypt the clinical content of electronic prescriptions and dispense records.

3.1.3   Qualified Retrieval Key

A `Qualified RK` is a `Retrieval Key` combined with the identifier of the service provider (in this case *PES Operator*) that manages all the clinical

documents that are associated with that `Retrieval Key` (i.e. all the documents for one prescription).

3.1.4   Current Interest

An instance of the `Current Interest` data type represents the registered interest of one *Dispenser Organisation* in one prescription. This data type is used in service response messages that are sent from a *Prescription Exchange* in response to service requests that were made at a time when a dispensing process may still be in progress. See section 4 for the service operations that may generate such a message.
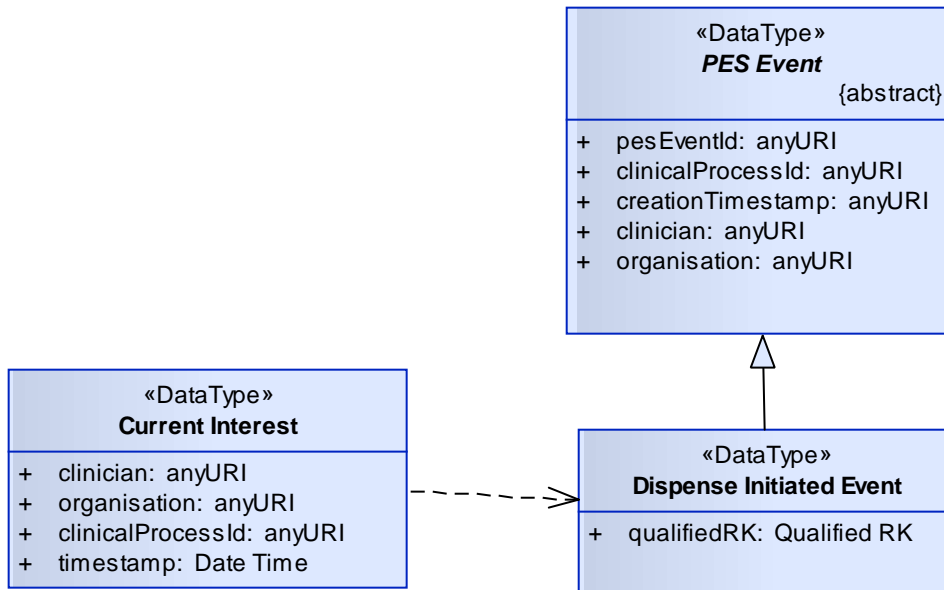
Figure 8: Current Interest

A prescription that may be in the process of being dispensed is one for which a `Dispense Initiated Event` (see section 3.2.3.5) has been received by the *Prescription Exchange* but for which no corresponding `Dispense Terminated Event` (see section 3.2.3.9) or `Dispense Finalised Event` (see section 3.2.3.7) has been received.

The `Current Interest` represents information that was included in the `Dispense Initiated Event` which signalled the commencement of a dispensing process as follows:

- `clinician`: the identity of the *Dispenser*, as specified in the `clinician` element of the `Dispense Initiated Event`

- `organisation`: the identity of the *Dispenser Organisation*, as specified in the `organisation` element of `Dispense Initiated Event`

- `clinicalProcessId`: the identity of the associated dispensing process instance, as specified in the `clinicalProcessId` element of `Dispense Initiated Event`

- `timestamp`: the time at which the `Dispense Initiated Event` occurred, as specified in the `creationTimestamp` element of `Dispense Initiated Event`.

3.1.5   Clinical Documents

There are two types of clinical document transferred between ETP service participants:

- `E-Prescription Document`: represents a prescription created by the Prescriber

- `E-Dispense Document`: represents a record of a successfully completed dispensing process; created by the Dispenser.
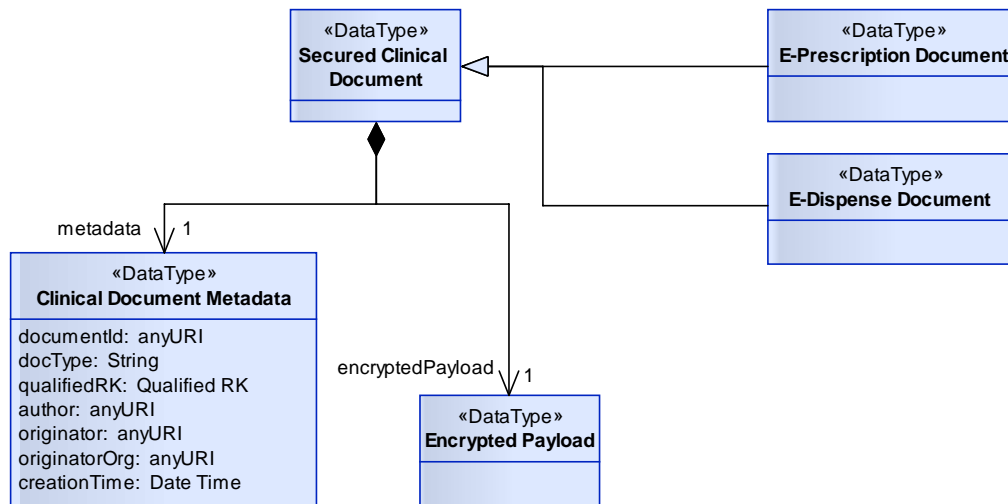
Figure 9: Data types that represent clinical documents

As shown in Figure 9, both these document types are represented as specialisations of a common `Secured Clinical Document` data type.

A `Secured Clinical Document` represents a clinical document which has been encrypted and then bundled together with a set of common document metadata which is provided for the purposes of managing that clinical document within a repository.

A `Secured Clinical Document` comprises*:*

- `Clinical Document Metadata:` A set of generic document attributes used for document management*.* This includes:

  - `documentId:` a unique identifier for the `Secured Clinical Document`

  - `docType:` a string identifying the `Secured Clinical Document` type, such as "`E-Prescription Document`" or "`E-Dispense Document`"

  - `qualifiedRK:` the `Qualified RK` for the document (see section 3.1.3)

  - `author:` identifies the person or organisation that created the `Secured Clinical Document` and attests to its clinical content

  - `originator:` identifier of the individual who is responsible for the issuance (outside the organisational boundary in which it was created) of the `Secured Clinical Document`

  - `originatorOrg:` identifier of the organisation that is responsible for the issuance (outside the organisational boundary in which it was created) of the `Secured Clinical Document` (e.g. medical practice, pharmacy).

- creationTime: time and date (UTC) of Secured Clinical Document creation

- Encrypted Payload: the clinical payload encrypted using a Cipher Key (see section 3.1.2)

  - The Encrypted Payload element of an E-Prescription Document shall comply with the logical information model specified in [ETP-EP_SDT2010]

  - The Encrypted Payload element of an E-Dispense Document shall comply with the logical information model specified in [ETP-ED_SDT2010].

3.2 PES Event data types

The concept of the *PES Event* was defined in section 2.4.3. There are six types of *PES Event* that can trigger changes in the state managed by a *Prescription Exchange*.

This section defines:

- The information that is communicated by a *Electronic Prescribing System* or *Electronic Dispensing System* to a *Prescription Exchange* when signalling each *PES Event*

- The information that is returned as acknowledgement that the *PES Event* was successfully processed (i.e. without generating a fault) by the *Prescription Exchange.*

3.2.1 Data types used to signal PES Events
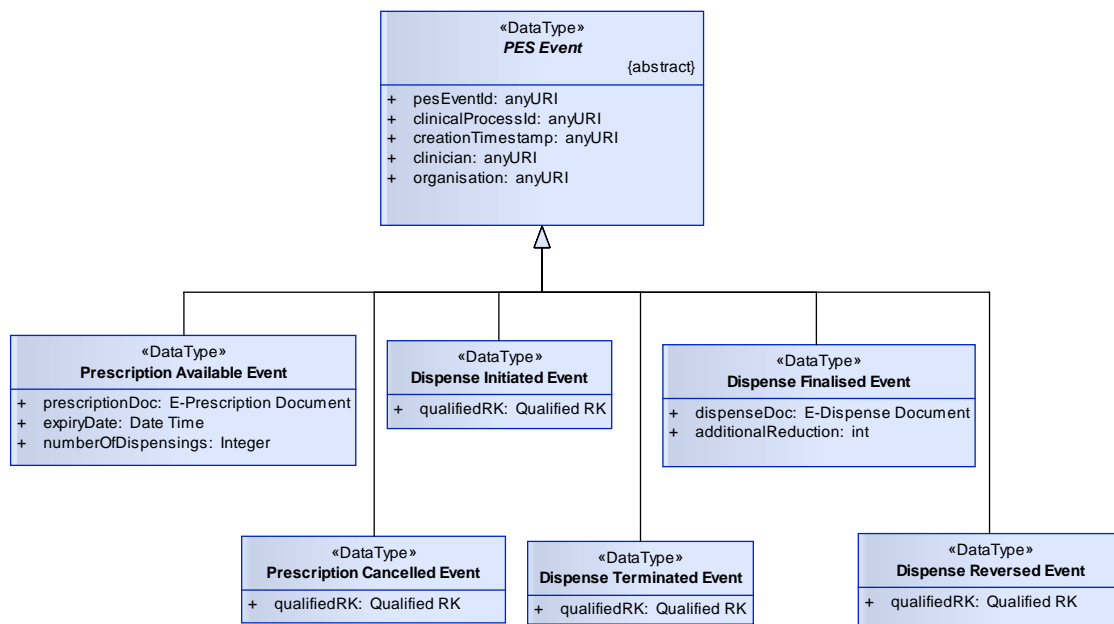
Figure 10: PES Events

An instance of the PES Event data type represents the data that is common to all *PES Events*:

- pesEventId: the globally unique identity of this *PES Event*

- clinicalProcessId: the identity of a clinical process that this *PES Event* is associated with. All *PES Events* for the same clinical process shall have the same clinicalProcessId

- **creationTimestamp**: a date and time (UTC) that this *PES Event* was created by the *Electronic Prescribing System* or *Electronic Dispensing System* which is signalling this *PES Event*

- **clinician**: the *Prescriber* or *Dispenser* taking clinical responsibility for the associated clinical process

- **organisation**: the *Prescriber Organisation* or the *Dispenser Organisation* responsible for the associated clinical process.

3.2.2   Data types used to acknowledge PES Events

For every `PES Event` received by a *Prescription Exchange,* the *Prescription Exchange* either signals back an acknowledgment of that `PES Event` or raises a fault.

A *Prescription Exchange* acknowledges a `PES Event` that it processed without raising a fault by sending an instance of the appropriate specialisation of the `PES Event Acknowledgment` data type.
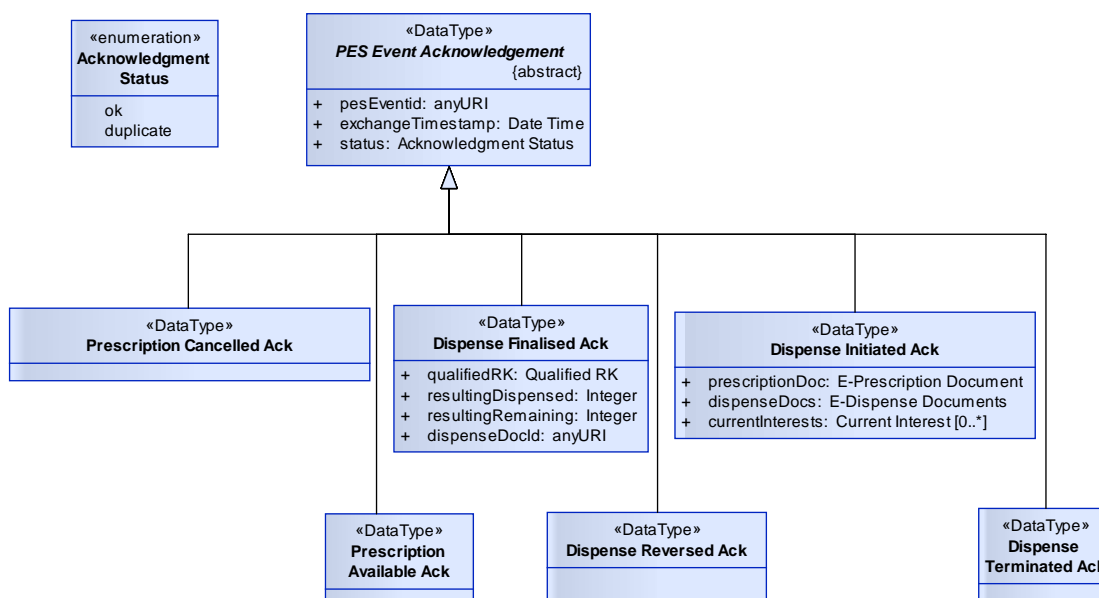
Figure 11: Acknowledgement of *PES Events*

The `PES Event Acknowledgement` data type represents the data that is common to all acknowledgments provided by a *Prescription Exchange.* A `PES Event Acknowledgement` comprises:

- **pesEventId**: the globally unique identity of the `PES Event` being acknowledged

- **exchangeTimestamp**: the date and time (UTC) at which the *Prescription Exchange* processed the `PES Event` being acknowledged; all resulting changes to state managed by the *Prescription Exchange* are assumed to have taken place simultaneously at this time

- **status**: the value `duplicate` indicates that the *Prescription Exchange* has already processed a `PES Event` with the same `pesEventId`. For duplicates the `exchangeTimestamp` is the timestamp associated with the original `PES Event` (i.e. it remains the time at which all changes were simultaneously made to the state that is managed by the *Prescription Exchange*). If the `PES Event` being acknowledged was not a duplicate then the status has the value `OK`.

3.2.3   Specialised PES Events

The occurrence of each of the six types of *PES Event* is signalled using a specialisation of the base `PES Event` data type.

Similarly the acknowledgment of each of the six types of *PES Event* is signalled using a specialisation of the base `PES Event Acknowledgement` data type. The following sub-sections define these specialisations.

3.2.3.1   Prescription Available Event occurrence

The occurrence of a "Prescription Available" *PES Event* is signalled using an instance of the `Prescription Available Event` data type.
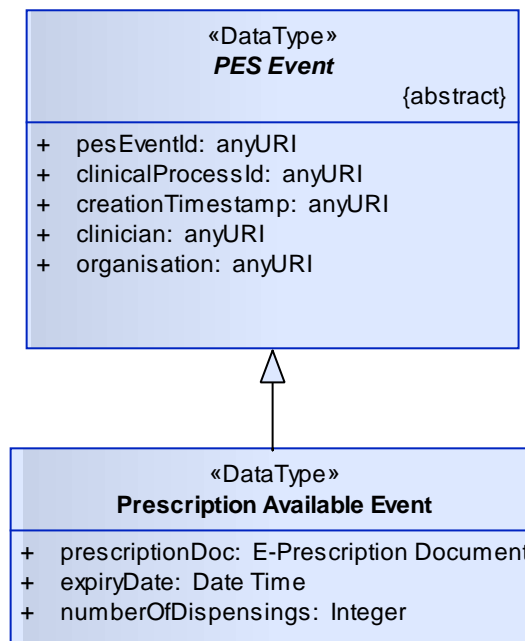
Figure 12: Prescription Available Event

`Prescription Available Event` extends `PES Event` to include:

- `prescriptionDoc`: the `E-Prescription Document` that represents the electronic prescription for which the `Prescription Available Event` signals a "Prescription Available" *PES Event*

- `expiryDate`: the date (UTC) after which the *Prescription Exchange* shall transition the dispensing state to "`expired`" (see section 2.4)

- `numberOfDispensings`: the maximum number of times that the prescription can be dispensed; equal to the number of repeats plus one.

3.2.3.2   Prescription Available event acknowledgement

The acknowledgement of a "Prescription Available" *PES Event* is an instance of the `Prescription Available Ack` data type.

`Prescription Available Ack` does not require extensions to the base `PES Event Acknowledgement` data type*.*

«DataType»
PES Event Acknowledgement
{abstract}

+   pesEventid: anyURI
+   exchangeTimestamp: Date Time
+   status: Acknowledgment Status

«DataType»
Prescription Available Ack
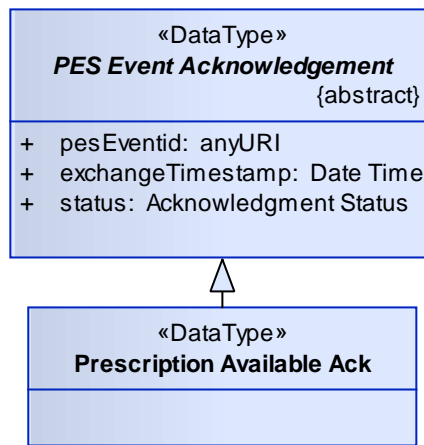
Figure 13: Prescription Available Acknowledgement

3.2.3.3   Prescription Cancelled event occurrence

The occurrence of a "Prescription Cancelled" *PES Event* is signalled using an instance of the `Prescription Cancelled Event` data type.

«DataType»
PES Event
{abstract}

+   pesEventId: anyURI
+   clinicalProcessId: anyURI
+   creationTimestamp: anyURI
+   clinician: anyURI
+   organisation: anyURI

«DataType»
Prescription Cancelled Event

+   qualifiedRK: Qualified RK
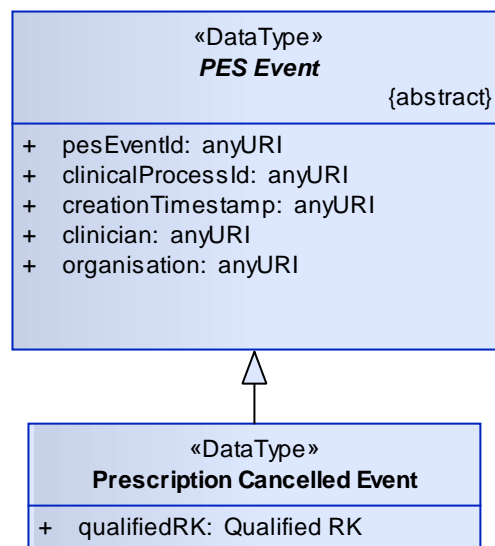
Figure 14: Prescription Cancelled Event

`Prescription Cancelled Event` extends the base `PES Event` data type to include:

*   `qualifiedRK`: the reference for the prescription being cancelled.

3.2.3.4   Prescription Cancelled Event acknowledgement

The acknowledgement a "Prescription Cancelled" P*ES Event* is an instance of the `Prescription Cancelled Ack` data type.
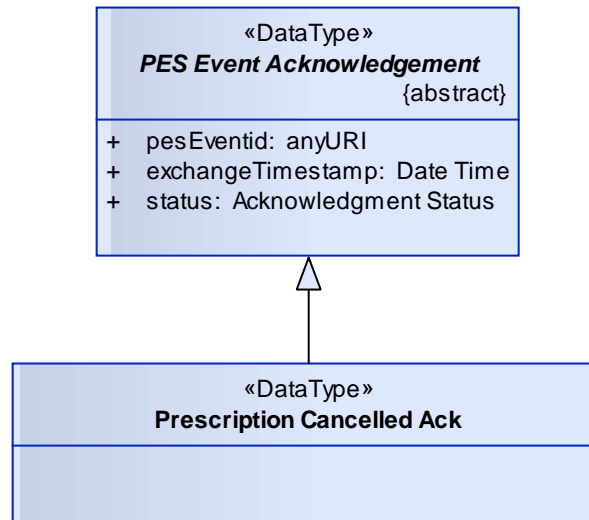
Figure 15: Prescription Cancelled Acknowledgement

`Prescription Cancelled Ack` does not require extensions to the base `PES Event Acknowledgement`.

3.2.3.5   Dispense Initiated event occurrence

The occurrence of a "Dispense Initiated" *PES Event* is signalled using an instance of the `Dispense Initiated Event` data type.
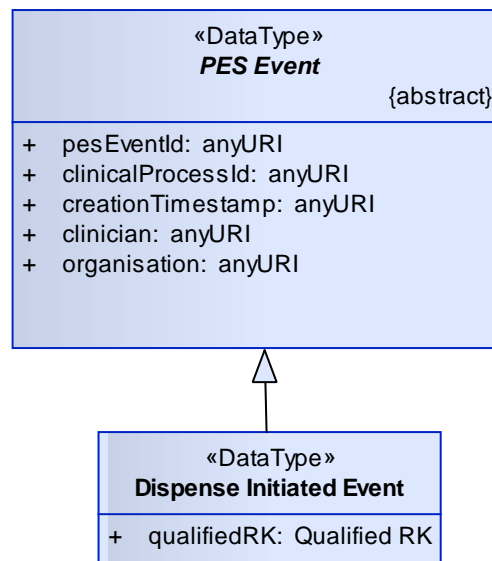
Figure 16: Dispense Initiated Event

`Dispense Initiated Event` extends the base `PES Event` data type to include:

- `qualifiedRK`: the reference for the prescription being dispensed.

The `clinicalProcessId` of the `Dispense Initiated Event` shall identify the associated end-to-end dispensing process (which may span multiple *PES Events*). The *Electronic Dispensing System* must use the same `clinicalProcessId` in those *PES Events* that relate to the same dispensing process (i.e. those *PES Events* that signal the successful completion, termination or reversal of that dispensing process).

The `clinicalProcessId` of the `Dispense Initiated Event` shall be unique for a given prescription.

3.2.3.6 Dispense Initiated event acknowledgement

The acknowledgement of a "Dispense Initiated" *PES Event* is an instance of the `Dispense Initiated Ack` data type*.*

```
«DataType»
PES Event Acknowledgement
                    {abstract}
+   pesEventid: anyURI
+   exchangeTimestamp: Date Time
+   status: Acknowledgment Status
```

```
«DataType»
Dispense Initiated Ack
+   prescriptionDoc: E-Prescription Document
+   dispenseDocs: E-Dispense Document [0..*]
+   currentInterests: Current Interest [0..*]
```

Figure 17: Dispense Initiated Acknowledgement

`Dispense Initiated Ack` extends `PES Event Acknowledgement` to include:

- `prescriptionDoc:` the `E-Prescription Document` that existed for the prescription being dispensed at the time that the `Dispense Initiated Event` was processed by the *Prescription Exchange* that manages the prescription

- `dispenseDocs:` a set of all the `E-Dispensing Documents` that existed for the prescription being dispensed at the time that the `Dispense Initiated Event` was processed by the *Prescription Exchange* that manages the prescription

- `currentInterests:` a set of all the `Current Interests` (see 3.1.4) that existed for the prescription being dispensed at the time that `Dispense Initiated Event` was processed by the *Prescription Exchange* that manages the prescription.

3.2.3.7 Dispense Finalised event occurrence

The occurrence of a "Dispense Finalised" *PES Event* is signalled using an instance of the `Dispense Finalised Event` data type.
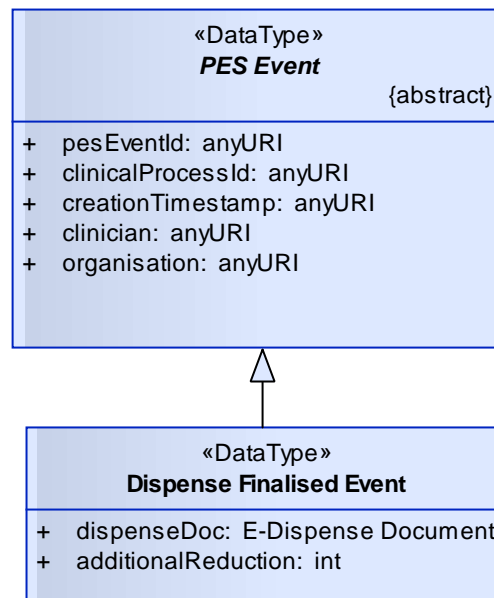
Figure 18: Dispense Finalised Event

`Dispense Finalised Event` extends `PES Event` to include:

- `dispenseDoc`: the `E-Dispense Document` that represents the electronic dispense record created by the *Dispenser*

- `additionalReduction`: the number by which a *Dispenser* chooses to reduce the number of repeats available for a given prescription; it is a number between zero and the total number of repeats that would otherwise have remained after the completion of the dispensing process.

The `clinicalProcessId` shall match the `clinicalProcessId` of the `Dispense Initiated Event` that signalled the commencement of the dispensing process being finalised.

3.2.3.8   Dispense Finalised event acknowledgement

The acknowledgement of a "Dispense Finalised" *PES Event* is an instance of the `Dispense Finalised Ack` data type.
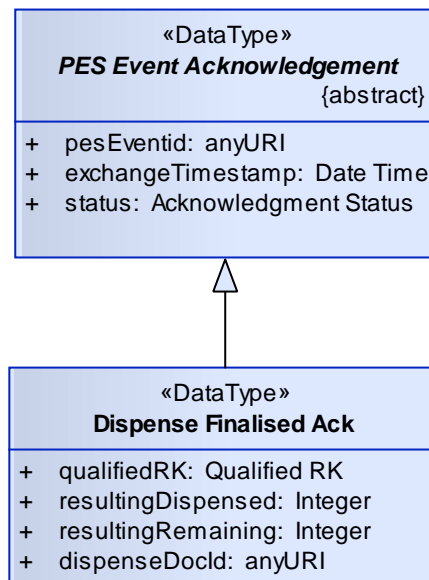
Figure 19: Dispense Finalised Acknowledgement

`Dispense Finalised Ack` extends `PES Event Acknowledgement` to include:

- `qualifiedRK:` the reference for the prescription to which the *PES Event* being acknowledged relates

- `resultingDispensed:` represents the number of times that the prescription has been dispensed

- `resultingRemaining:` represents the number of remaining times that the prescription may be dispensed

- `dispenseDocId:` the globally unique document identity of the `E-Dispense Document` that was included in the `Dispense Finalised Event` which signalled the *PES Event* that is being acknowledged.

3.2.3.9   Dispense Terminated event occurrence

The occurrence of a "Dispense Terminated" *PES Event* is signalled using an instance of the `Dispense Terminated Event` data type.
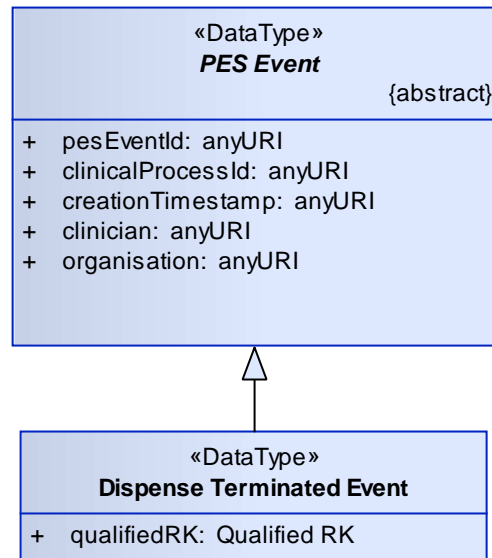
Figure 20: Dispense Terminated Event

`Dispense Terminated Event` extends the base `PES Event` data type to include:

- `qualifiedRK`: the reference for the prescription whose dispensing was terminated.

The `clinicalProcessId` shall match the `clinicalProcessId` of the `Dispense Initiated Event` that signalled the commencement of the dispensing process being terminated.

3.2.3.10   Dispense Terminated event acknowledgement

The acknowledgement of a "Dispense Terminated" *PES Event* is an instance of the *Dispense Terminated Ack* data type.

`Dispense Terminated Ack` does not require extensions to the base `PES Event Acknowledgement` data type.
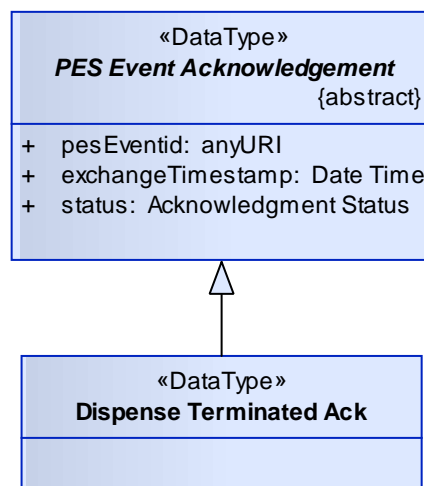
Figure 21: Dispense Terminated Acknowledgement

3.2.3.11   Dispense Reversed event occurrence

The occurrence of a "Dispense Reversed" *PES Event* is signalled using an instance of the `Dispense Reversed Event` data type.
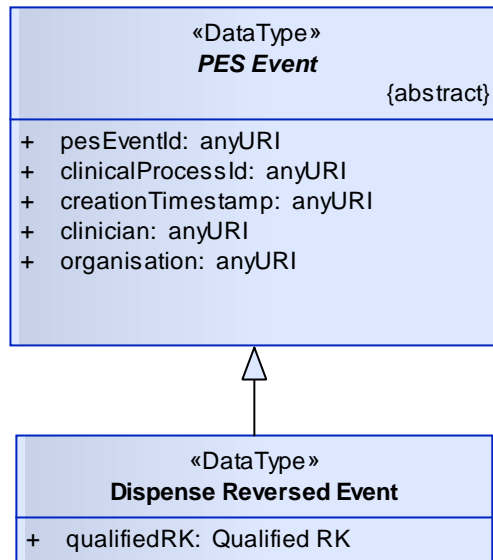
Figure 22: Dispense Reversed Event

`Dispense Reversed Event` extends the base `PES Event` data type to include:

- `qualifiedRK`: the reference for the prescription whose dispensing was reversed.

The `clinicalProcessId` shall match the `clinicalProcessId` of the `Dispense Initiated Event` that signalled the commencement of the dispensing process being reversed.

3.2.3.12 Dispense Reversed event acknowledgement

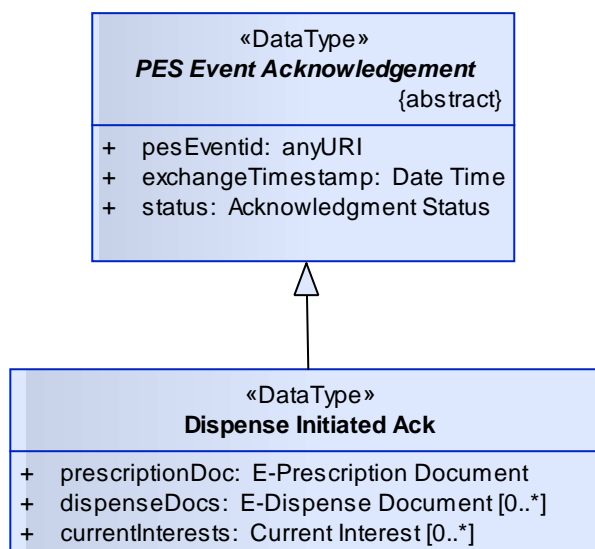The acknowledgement of "Dispense Reversed" *PES Event* is an instance of the `Dispense Reversed Ack` data type.

`Dispense Reversed Ack` does not require extensions to the base `PES Event Acknowledgement` data type*.*
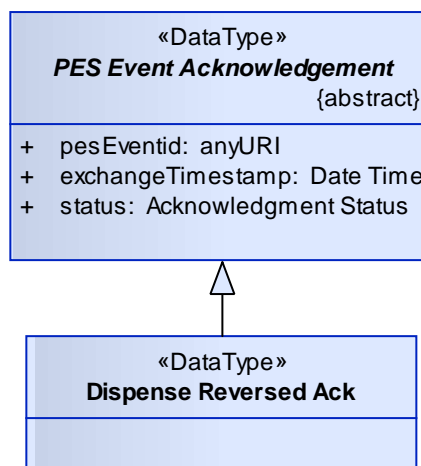
Figure 23: Dispense Reversed Acknowledgement

3.3   Retrieval Request Data Types

A Retrieval Request is defined as a type of service function that queries (but cannot change) the state managed by a *Prescription Exchange* (other than audit records kept by the *Prescription Exchange*).

Only one *Retrieval Request* service function is specified – the *E-Prescribing* "Retrieve Prescription" service function (EP3 in Table 1).

This section defines:

1. The information that is communicated to a *Prescription Exchange* for each *Retrieval Request*

2. The information that is returned as a response to a *Retrieval Request* that was received and accepted as valid by the *Prescription Exchange*.

3.3.1   Retrieval Request


```
            «DataType»
          Retrieval Request
                          {abstract}

+   qualifiedRK: Qualified RK
+   requestId: anyURI
+   requestTimestamp: Date Time
+   clinician: anyURI
+   organisation: anyURI
```
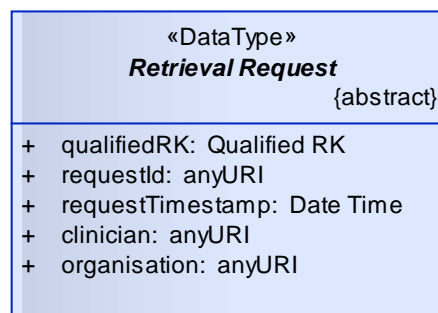
Figure 24: Retrieval Request

An instance of the `Retrieval Request` data type represents the data that is common to all *Retrieval Requests*[10]:

*   `qualifiedRK`: the reference for the prescription being retrieved

*   `requestId`: a globally unique identity of this *Retrieval Request*.

*   `requestTimestamp`: the date and time (UTC) that the request was made according to the system (e.g. *Electronic Prescribing System*) responsible for communicating the *Retrieval Request*

*   `clinician`: the *Prescriber* or *Dispenser* taking clinical responsibility for the *Retrieval Request*

*   `organisation`: the *Prescriber Organisation* responsible for the *Retrieval Request*

3.3.2   Retrieval Response

For every `Retrieval Request` received by a *Prescription Exchange*, the *Prescription Exchange* either returns an instance that is a specialisation of the `Retrieval Response` data type or raises a fault.

---

[10]   Current only one is specified which returns the *E-Prescription Document* but an extensible model is used that allows for other types of retrieval request in the future.
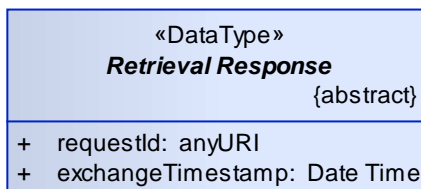
Figure 25: Retrieval Response

An instance of the `Retrieval Response` data type represents the data that is common to all responses provided by a *Prescription Exchange.* It comprises:

- `requestId:` the unique identity of the *Retrieval Request* being responded to

- `exchangeTimestamp:` the date and time (UTC) at which the *Prescription Exchange* processed the *Retrieval Request*.

3.3.3   Specialised Retrieval Requests

There is only one type of *Retrieval Request* – a *Prescription Retrieval Request*

3.3.3.1   Prescription Retrieval Request

A request to retrieve an `E-Prescription Document` is communicated using an instance of the `Prescription Retrieval Request` data type.
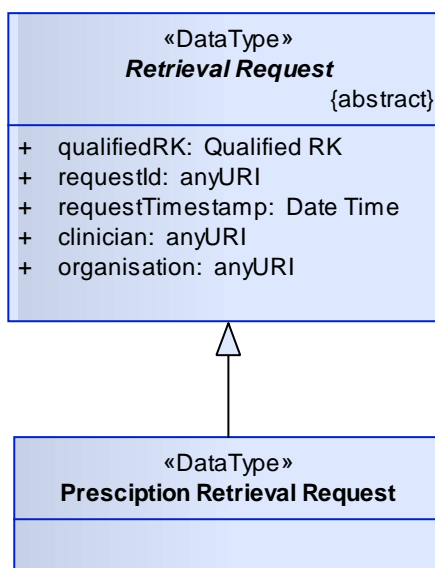
Figure 26: Prescription Retrieval Request

`Prescription Retrieval Request` does not require extensions to the base *Retrieval Request* data type

3.3.3.2   Prescription Retrieval Response

A response to a request to retrieve an *E-Prescription Document* is communication using an instance of the `Prescription Retrieval Response` data type.
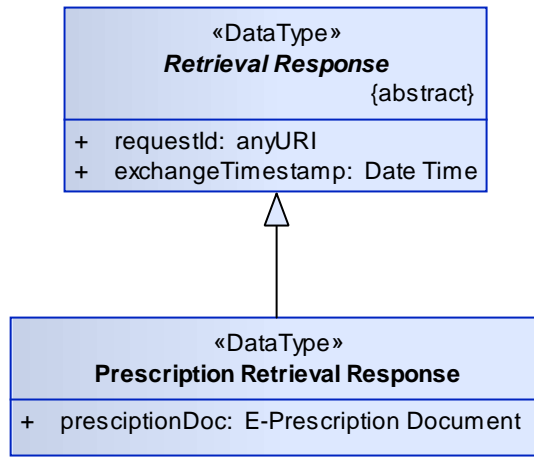
Figure 27: Prescription Retrieval Response

The `Prescription Retrieval Response` extends `Retrieval Response` to include:

- `prescriptionDoc:` this is the `E-Prescription Document` associated with the prescription.

3.4   ETP Notification Data Types

An *ETP Notification* is defined as a type of service function that notifies an identified ETP service participant of the occurrence of an event.

The only obligation of a service provider IT system that receives an *ETP Notification* is to acknowledge its delivery.

3.4.1   ETP Notification
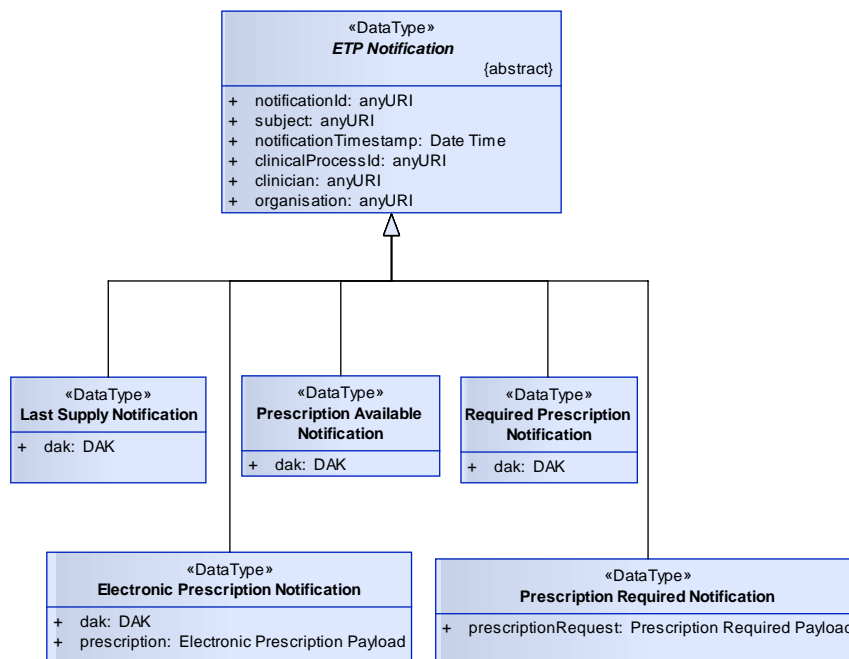
Figure 28: ETP Notifications

An instance of the `ETP Notification` data type represents the data that is common to all types of *ETP Notification*:

- `notificationId`: the globally unique identity of the notification

- `subject`: the IHI of the *Prescription Subject*

- `clincalProcessId`: the identity of a clinical process instance that this `ETP Notification` is associated with – there is no restriction on the value of this attribute – except where it is constrained by a particular `ETP Notification` specialisation.

- `notificationTimestamp`: the date and time (UTC) that the notification was created according to the system (e.g. *Electronic Prescribing System*) responsible for generating the notification

- `clinician`: the *Prescriber* or *Dispenser* taking clinical responsibility for generating the notification

- `organisation`: the *Prescriber Organisation* or *Dispenser Organisation* responsible for generating the notification.

3.4.2   ETP Delivery Notification Acknowledgement

A service provider that receives an `ETP Notification` is required to acknowledge its delivery using an instance of the `ETP Notification Delivery Ack` data type.

«DataType» **ETP Notification Delivery Ack**
+   notificationId: anyURI   +   ackTimestamp: Date Time
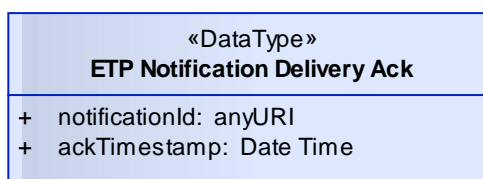
Figure 29: ETP Notification Delivery Acknowledgement

An `ETP Notification Delivery Ack` comprises*:*

- `notificationId`: the identity of the notification whose delivery is being acknowledged

- `ackTimestamp`: the date and time (UTC) when the *ETP Notification* was received*.*

3.4.3   Specialised ETP Notifications

3.4.3.1   Prescription Required Notification

A `Prescription Required Notification` is sent from an *Electronic Dispensing System* to an *Electronic Prescribing System* to inform a *Prescriber* that a *Dispenser* requires the *DAK* for a prescription that confirms an earlier prescriber instruction.
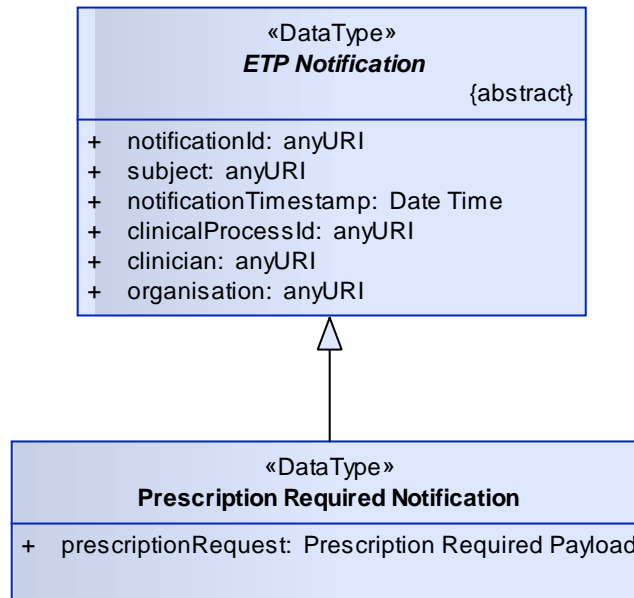
Figure 30: Prescription Required Notification

The `Prescription Required Notification` data type represents the information sent by an *Electronic Dispensing System* to an *Electronic Prescribing System* as a request to provide an electronic prescription that confirms a prior prescription instruction.

The `Prescription Required Notification` data type extends the base `ETP Notification` data type to include:

- `prescriptionRequest:` the document created by the *Dispenser* in response to the prior prescription instruction. The payload must comply with [ETP-PR_SDT2010].

3.4.3.2   Required Prescription Notification

A `Required Available Notification` is sent, in response to a `Prescription Required Notification`, to inform a *Dispenser* that an electronic prescription, that confirms a prior instruction, has been published to a *Prescription Exchange*.
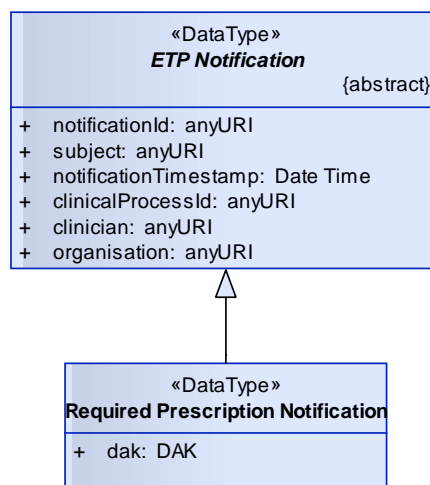
Figure 31: Required Prescription Notification

The `Required Prescription Notification` data type extends the base `ETP Notification` data type to include:

- `dak:` the *DAK* that is associated with the prescription.

Note that the `clinicalprocessId` of a `Required Prescription Notification` matches the `clinicalprocessId` of the `Prescription Required Notification` being responded to.

3.4.3.3 Prescription Available Notification

A `Prescription Available Notification` is sent to inform the receiver that an electronic prescription has been published to a *Prescription Exchange*.
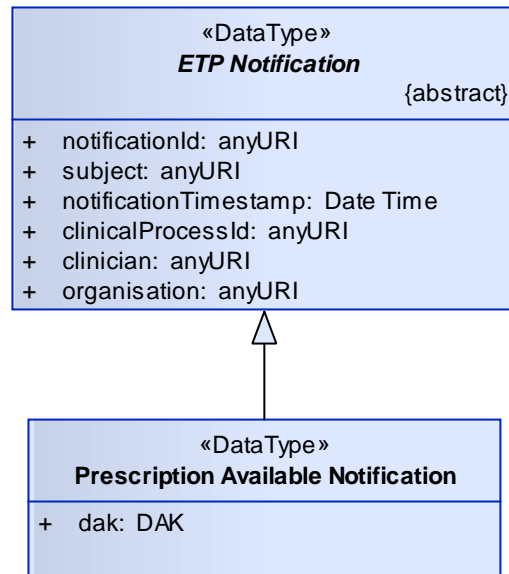
Figure 32: Prescription Available Notification

The `Prescription Available Notification` data type extends the base `ETP Notification` data type to include:

- `dak:` the *DAK* that is associated with the prescription.

3.4.3.4 Electronic Prescription Notification

An `Electronic Prescription Notification` is sent to inform the receiver that an electronic prescription has been published to a *Prescription Exchange* and to provide the notification receiver with an electronic form of the prescription document.
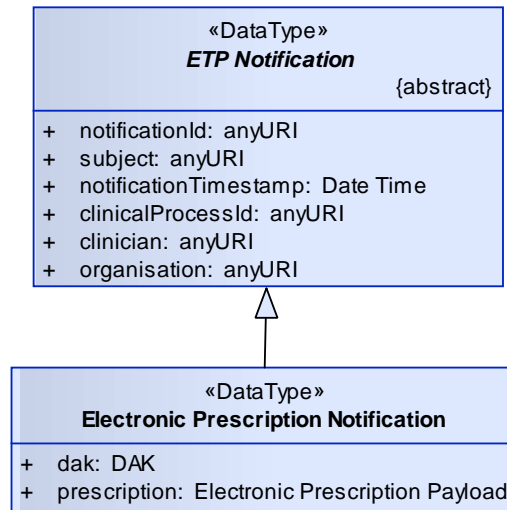
Figure 33: Electronic Prescription Notification

The `Electronic Prescription Notification` data type extends the base `ETP Notification` data type to include:

- `dak`: the *DAK* that is associated with the prescription.

- `prescription`: the electronic prescription; this element must comply with [ETP-EP_SDT2010].

3.4.3.5   Last Supply Notification

A `Last Supply Notification` is sent from an *Electronic Dispensing System* to a *Last Supply Notification Agent* to inform the *Prescription Subject* (or a party acting on their behalf) that a dispensing process has been finalised for the last supply of a prescription.
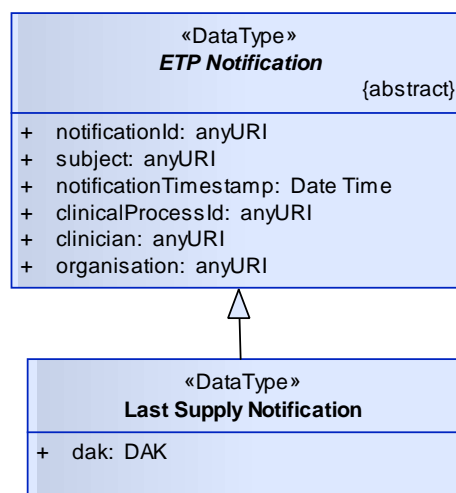
Figure 34: Last Supply Notification

The `Last Supply Notification` data type extends the base `ETP Notification` to include:

- `dak`: the `DAK` for the prescription for which the last dispensing occurred

4   Technical View

4.1   Platform Independence

This chapter defines a technical architecture and "platform independent" conformance points.

4.2   Modelling language

The modelling language used in this technical view is UML2.3 and, in particular, a profile of UML2.3 defined as the SOA Modelling Language [SOAML].

The SOA Modelling Language (SoaML) concepts that are used include participants, service contracts, and interfaces. With respect to this technical view, they are defined as follows:

- **Participant**: a role that is fulfilled by a type of system; participants provide and consume services. A participant is represented as a SoaML `<<participant>>` which is a UML stereotype that extends the UML `Class` metaclass.

- **Service contract:** an agreement between participants for how a service is to be provided and consumed. In this technical view these agreements are expressed in terms of the conformance points that apply to particular participants and are primarily expressed in terms of the interfaces provided and required by each participant role. A service contract is represented as a SoaML `<<serviceContract>>` which is a UML stereotype that extends the UML `Collaboration` metaclass.

- **Interface:** follows the UML semantics (i.e. an interface declares a set of public features and obligations that constitute a coherent service offered by a classifier[11] [UML2.3]).

4.3   Technical Services Architecture Overview

4.3.1   Participants

The roles defined in section 2.1 that represent systems are directly represented in this technical view as SoaML `<<participants>>`, as follows:

- `Electronic Prescribing System`
- `Electronic Dispensing System`
- `Prescription Exchange`
- `Facility Based Supply Manager`
- `Last Supply Notification Agent`

4.3.2   Service Contracts

In this platform independent technical view there is a one-to-one correspondence between the ETP business services defined in section 2.2 and the technical services that support them.

---

[11] In this context a SoaML `Participant`

Figure 35 shows the relationships between the participants and the ETP services (expressed as SoaML <<serviceContracts>>).
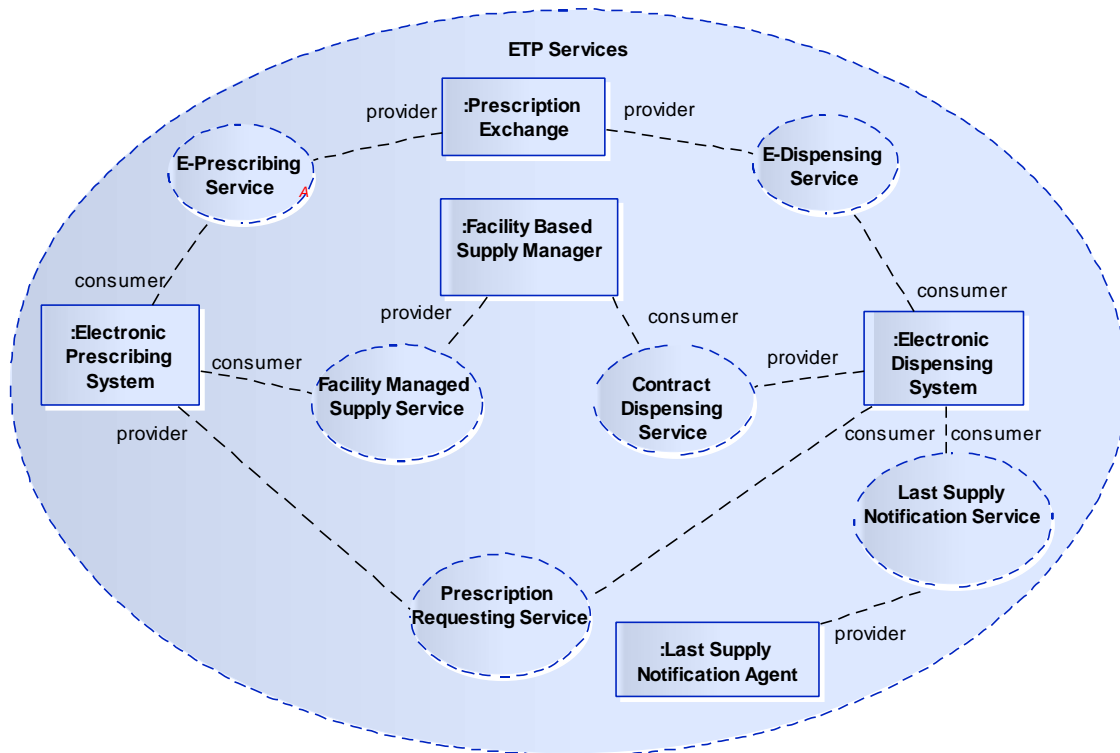
Figure 35: ETP Services Architecture (SoaML)

Note that there is one service contract corresponding to each of the business services defined in section 2.2. The following are the defined <<serviceContracts>>:

- E-Prescribing Service

- E-Dispensing Service

- Prescription Requesting Service

- Facility Managed Supply Service

- Contract Dispensing Service

- Last Supply Notification Service

4.3.3　Service Interfaces

Each of the <<serviceContracts>> defined in the previous section, with the exception of the Prescription Requesting Service, are "simple services" in the sense that these are each defined in terms of one interface; one participant implements a provider of that interface and the other participant implements an invoker of that interface.

The Prescription Requesting Service, however, requires two interfaces:

- The Electronic Prescribing System <<participant>> implements a provider of the Prescription Requesting interface and an invoker of the Prescription Receiving interface

- The Electronic Dispensing System <<participant>> implements an invoker of the Prescription Requesting interface and a provider of the Prescription Receiving interface.

Figure 36 shows the participants and the interfaces that they require and provide.
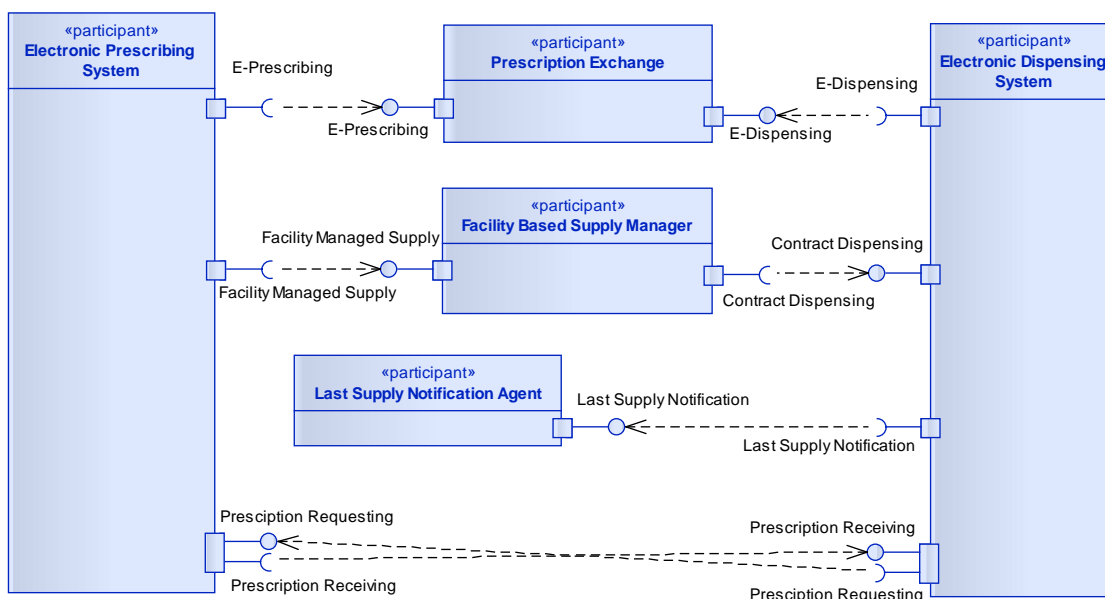
Figure 36: ETP Participants and services interfaces (SoaML)

In this platform independent technical view, interfaces are defined as UML interfaces with operations that have one input request message and return a response message. Request and response messages are defined in terms of the data types specified in section 3.

There are two types of ETP interface:

- Figure 37 shows those interfaces supported by a `Prescription Exchange;` for these interfaces the eventual receiver of a document (e.g. a *Dispenser Organisation*) that is sent via these interfaces is not identified by the sender of that document (e.g. a *Prescriber Organisation*)

- Figure 38 shows those interfaces that are used to send information point-to-point from a sender to known receiver that is identified by the sender.

«interface» **E-Prescribing** {abstract}
+ prescribe(Prescription Available Event) : Prescription Available Ack
+ retrievePrescription(Prescription Retrieval Request) : Prescription Retrieval Response
+ cancelPrescription(Prescription Cancelled Event) : Prescription Cancelled Ack

«interface» **E-Dispensing** {abstract}
+ initiateDispense(Dispense Initiated Event) : Dispense Initiated Ack
+ finaliseDispense(Dispense Finalised Event) : Dispense Finalised Ack
+ terminateDispense(Dispense Terminated Event) : Dispense Terminated Ack
+ reverseDispense(Dispense Reversed Event) : Dispense Reversed Ack
+ cancelPrescription(Prescription Cancelled Event) : Prescription Cancelled Ack

Figure 37: Prescription Exchange Interfaces

«interface»
Prescription Receiving
{abstract}
+ deliverNotification(Required Prescription Notification) : ETP Notification Delivery Ack

«interface»
Prescption Requesting
{abstract}
+ deliverNotification(Prescription Required Notification) : ETP Notification Delivery Ack

«interface»
Facility Managed Supply
{abstract}
+ deliverNotification(Electronic Prescription Notification) : ETP Notification Delivery Ack

«interface»
Contract Dispensing
{abstract}
+ deliverNotification(Prescription Available Notification) : ETP Notification Delivery Ack

«interface»
Last Supply Notification
{abstract}
+ deliverNotification(Last Supply Notification) : ETP Notification Delivery Ack

Figure 38: Point-to-point ETP interfaces

The following table summarises the relationships between service contracts, provided interfaces and participants:

SERVICE CONTRACT	SERVICE PROVIDER (RESPONDER)		SERVICE CONSUMER (INITIATOR)	
	Participant	Provided Interface	Participant	Provided Interface
E-Prescribing Service	Prescription Exchange	E-Prescribing	Electronic Prescribing System	
E-Dispensing Service	Prescription Exchange	E-Dispensing	Electronic Dispensing System	
Prescription Requesting Service	Electronic Prescribing System	Prescription Requesting	Electronic Dispensing System	Prescription Receiving
Facility Managed Supply Service	Facility Based Supply Manager	Facility Managed Supply	Electronic Prescribing System	
Contract Dispensing Service	Electronic Dispensing System	Contract Dispensing	Facility Based Supply Manager	

Last Supply Notification Service	Last Supply Notification Agent	Last Supply Notification	Electronic Dispensing System	

Table 2: Services, participants and interfaces

4.4 Common Specifications

The following specifications apply to multiple ETP service participants and/or to the same participant but in respect of multiple interfaces or operations.

4.4.1 Identity Credentials and PKI

The following conformance points specify authentication controls that are independent of implementation choices. The implementation details of a PKI, and of identity credentials issued in accordance with it, are specified in the ETP Technical Services Specification [ETP-TSS2010].

In the following an *Invoker System* is any ETP service participant that invokes an ETP interface and a *Provider System* is any ETP service participant that provides an ETP interface:

ETP 1　　All *Invoker Systems* shall associate an identity credential with every service request using an appropriate mechanism that allows the *Provider System* to determine the identity of the organisation that is responsible for invoking the service.

ETP 2　　All *Provider Systems* shall associate an identity credential with every service response using an appropriate mechanism that allows the *Invoker Systems* to determine the identity of the organisation that is responsible for responding to the service request.

　　　　Note that appropriate mechanisms may include cases where identity credentials are associated with network connections used to securely transport service responses.

ETP 3　　All *Invoker Systems* and *Provider Systems* shall generate and validate identity credentials within a X.509 compliant PKI that allows them to mutually authenticate each other with a high level of assurance according to [NeAF].

ETP 4　　If a *Provider System* cannot validate an identity credential provided by an *Invoker System* then the *Provider System* shall reject the request.

ETP 5　　If an authenticated *Invoker System* is not a participant in the service then the *Provider System* shall reject the request and inform the *Invoker System* that a fault condition of type `notAuthorised` (as defined in Table 3) was detected.

ETP 6　　If an *Invoker System* cannot validate an identity credential provided by a *Provider System* then the *Invoker System* shall escalate the issue to a responsible representative of the organisation receiving the response (i.e. the organisation operating the *Invoker System*).

ETP 7　　All *Invoker Systems* and *Provider Systems* should comply with RFC 5280[12] for local validation and management of certificates.

ETP 8　　All *Invoker Systems* and *Provider Systems* should cache certificate status information for the period of time recommended by the certificate revocation list (CRL) or on-line certificate status profile (OCSP) `'nextUpdate'` field.

---

[12] See [RFC5280]

ETP 9   All *Invoker Systems* and *Provider Systems* should not contact the certificate issuer each time a certificate is used (i.e. certificate checks should occur out-of-band wherever possible).

4.4.2   Common Fault Types

The following types of "logical fault condition" that are detected by one or more participants in the ETP services.

As shown below, none of these "logical faults" are considered to be recoverable at the system level (i.e. all indicate conditions that must be brought to the attention of relevant system users). Note that ETP Technical Services Specification [ETP-TSS2010] will define additional faults. These may include new fault conditions that are amenable to being handled at the system level; for example, by changing system or transport layer configuration and retrying a failed service invocation.

FAULT TYPE	FAULT CONDITION	RECOVERABLE FAULT
`invalidProviderId`	The `providerId` element of a `Qualified RK` does not identify an operator of a `Prescription Exchange` to which the request message can be directed.	No
`unknownRetrievalKey`	`Qualified RK` does not reference an existing prescription.	No
`duplicateRK`	`Qualified RK` is already associated with an existing prescription.	No
`duplicateDocumentId`	The `documentId` element of an `E-Prescription Document` is not unique.	No
	The `documentId` element of an `E-Dispense Document` is not unique.	No
`unmatchedProcessId`	The `clinicalProcessId` element of a `Dispense Terminated` or `Dispense Finalised` or `Dispense Reversed PES Event` does not match that of a `Dispense Initiated PES Event` previously signalled for the prescription.	No
`invalidProcessId`	The `clinicalProcessId` element of a `Dispense Terminated` or `Dispense Finalised PES Event` matches that of a `Dispense Initiated PES Event` that has been previously matched by a different `Dispense Terminated` or `Dispense Finalised PES Event`. Or The `clinicalProcessId` element of a `Dispense Reversed PES Event` matches that of a `Dispense Initiated PES Event` that has not been previously matched by `Dispense Finalised PES Event`	No
`duplicateProcessId`	The `clinicalProcessId` element of a `Dispense Initiated PES Event` is not unique within the context of the target prescription.	No
`fullyDispensed`	`Qualified RK` references a prescription with a dispensing state of	No

	fullyDispensed.	
cancelled	Qualified RK references a prescription with a dispensing state of cancelled.	No
expired	Qualified RK references a prescription with a dispensing state of expired.	No
notAuthenticated	The organisation responsible for a service request could not be authenticated as a healthcare organisation or a contract service provider.	No
notAuthorised	The service request was not authorised	No
invalidMessage	A service request or response message did not conform to the required schema.	No

Table 3: Common faults

4.4.3   Reliable communications

The invokers of all the ETP interfaces are required to manage communications failures by retrying service invocations.

The providers of ETP interfaces are required to respond without unnecessary delay to every service request received.

ETP 10   If an *Invoker System* does not receive a response within a time limit determined by that *Invoker System* then it shall continue to retry the operation until a response is received, or an authorised user explicitly terminates the retry process, or a time limit is exceeded.

ETP 11   An *Invoker System* shall escalate the management of all faults classified as unrecoverable in Table 3 to the user who was responsible for invoking the service request or to another user authorised by the *Invoker System* to access the local healthcare records of the *Prescription Subject* identified in the service request.

ETP 12   If an *Invoker System* receives a fault response, and the fault condition is classified as a recoverable, then it shall either:

ETP 12-1   Manage the fault as an "unrecoverable fault" in accordance with ETP 11; or

ETP 12-2   Manage the fault by retrying the operation until a response is received, or an authorised user explicitly terminates the retry process[13]

ETP 13   A retried operation shall be an invocation of the same operation, where service request message is identical to the service request message for which no response was received.

ETP 14   An *Invoker System* shall warn the user, who is responsible for the invocation of a service interface, of service requests for which no response has been received within the time period specified in a service level agreement as the maximum response time for the particular operation plus an allowance for communications delays.

ETP 15   An *Invoker System* shall provide a capability that allows an authorised user to terminate the process of retrying service requests:

ETP 15-1   An authorised user is the user who was responsible for invoking the service request or an administrative user acting with their consent.

---

[13]   Typically the system would change some relevant aspect of the system or transport layer configuration before retrying.

ETP 15-2   If this system capability is used then the *Invoker System* shall warn the user that the operation may have previously succeeded.

ETP 15-3   An *Invoker System* shall only allow this capability to be used after an extended period of time has elapsed during which *Invoker System* is operational and is connected to the internet. The minimum period shall comply with a service level agreement expressed in terms of the `MinRetryPeriod` metric.

ETP 16   The *Invoker System* shall ensure that the maximum period during which retries are attempted shall comply with a service level agreement expressed in terms of the `MaxRetryPeriod` metric.

ETP 17   The *Invoker System* shall ensure that the maximum interval between successive attempts to send the same service request instance shall comply with a service level agreement expressed in terms of the `MaxRetryInterval` metric.

ETP 18   The *Invoker System* shall ensure that the minimum interval between successive attempts to send the same service request instance shall comply with a service level agreement expressed in terms of the `MinRetryInterval` metric.

4.4.4   Data Security

Security of data while it is in transit between ETP participants relies on encryption mechanisms that are specified in the ETP Technical Services Specification [ETP-TSS2010].

Security of the data while at rest within a `Prescription Exchange` is provided by encrypting the personal health information using `DAKs` and protecting these `DAKs` in accordance with the following:

ETP 19   An *Invoker System* shall encrypt the clinical payload component of a `Secured Clinical Document` using the `Cipher Key` derived from same `DAK` as was used to derive the `Retrieval Key` which is an element of the `Qualified RK` that is used to reference that `Secured Clinical Document`.

ETP 20   An *Invoker System* or a *Provider System* (i.e. any system that fulfils any of the participant roles that are listed in section 4.3.1):

ETP 20-1   Shall not derive a `Cipher Key` or a `Retrieval Key` from a `DAK` unless the user of that system is authorised to access information for the *Prescription Subject* who is identified in the `E-Prescription Document` that was associated with that `DAK` in accordance with section 4.7.5.1.

ETP 20-2   Should not store a `DAK` or its derived `Cipher Key` on stable storage without first encrypting these values using a strong, secret encryption cipher key of at least 128 bits.

4.4.5   End-point Location

Mechanisms will be defined in the ETP Technical Services Specification [ETP-TSS2010] by which the appropriate service provider participant can be located based on their identity (as a healthcare provider or a contracted service provider).

4.5   Prescription Exchange acting as prescription manager

4.5.1   Introduction

A `Prescription Exchange` is a system that manages and provides access to the ETP electronic documents.

This section (4.5) defines the conformance points that apply when a `Prescription Exchange` acts as a "prescription manager".

A `Prescription Exchange` acts as a prescription manager when it services a request for a prescription that it manages. All the prescriptions managed by a `Prescription Exchange` are referenced using `Qualified RKs` where the `providerId` matches the identity of the *Prescription Exchange Operator* that operates that `Prescription Exchange`.

If it is not acting as a prescription manager then a `Prescription Exchange` acts as an "intermediary" and section 4.6 defines the conformance points that apply in that case.

Each prescription is managed by one `Prescription Exchange`; the `Electronic Prescribing System` which creates an `E-Prescription Document` selects the target `Prescription Exchange` to manage that prescription. The identity of this target `Prescription Exchange` forms part of the `Qualified RK` which is the reference used when referring to that prescription. Subsequently all dispensing records (`E-Dispense Documents`) for the same prescription are also stored by the same target `Prescription Exchange` as stores the `E-Prescription Document` for that prescription.

Linking of `E-Dispense Documents` and an `E-Prescription Document` is achieved by using the same `Qualified RK` as the reference to all the documents for one prescription. Each document also has its own unique identity that is independent of the `Qualified RK`.

4.5.2   State managed by a Prescription Exchange

A `Prescription Exchange` is required to:

- Store `E-Prescription Documents` and `E-Dispense Documents`
- Represent the dispensing state of prescriptions
- Represent any *Current Interests* (see section 3.1.4) that exist for each prescription.

The following is a logical information model of the data stored by a `Prescription Exchange` for each prescription that it manages. Note that this logical information model is not a normalised data model suitable for implementing a `Prescription Exchange` – it is simply defined so as to conveniently specify the semantics of the operations supported by a `Prescription Exchange` (i.e. these operations are specified in terms of the effect that they produce on this logical information).

PE Prescription State
+   qualifiedRK: Qualified RK
+   dispensingState: Dispensing State
+   expiryDate: Date Time
+   dispensed: Integer
+   remaining: Integer
+   currentInterests: Current Interest [0..*]
+   ePrescriptionDoc: E-Prescription Document
+   eDispenseDocs: E-Dispense Document [0..*]

«enumeration» Dispensing State
fullyDispensed
availableToDispense
cancelled
expired
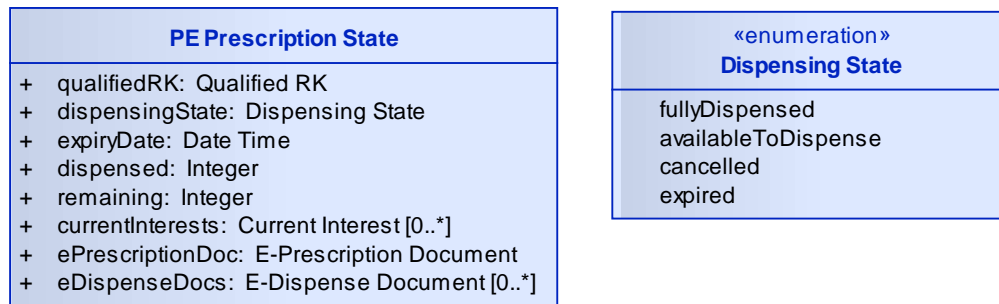
Figure 39: State managed by a Prescription Exchange

The state that is managed by a `Prescription Exchange` for each managed prescription is represented as an instance of the `PE Prescription State` class; this class comprises the following attributes:

- `qualifiedRK:` the reference for this prescription (note this will match the `Qualified RK` for the `ePrescriptionDoc` and all the `eDispenseDocs`)

- `dispensingState:` the dispensing state of the prescription

- `expiryDate:` the date and time after which the prescription may not be dispensed

- `dispensed:` the total number of times that this prescription has been dispensed

- `remaining:` the total number of times remaining for this prescription to be dispensed

- `currentInterests:` one element of this list exists for each instance of a `Dispense Initiated PES Event` that has been received by the `Prescription Exchange` for this prescription but for which no corresponding `Dispense Terminated` or `Dispense Finalised PES Event` has been received

- `ePrescriptionDoc:` the electronic prescription

- `eDispenseDocs :` one element of this list exists for each electronic dispense record for this prescription.

4.5.3   Interfaces

ETP 21   A `Prescription Exchange` shall implement a provider of the E-`Prescribing` interface.

ETP 22   A `Prescription Exchange` shall implement a provider of the E-`Dispensing` interface

4.5.4   Authentication and Authorisation

ETP 23   A `Prescription Exchange` shall authenticate the organisation that is responsible for each service request in accordance with 4.4.1.

ETP 24   A `Prescription Exchange` shall authorise service requests in accordance with policies determined by the *ETP Governance Authority*.

4.5.5 Common Processing of PES Events

A `Prescription Exchange` is informed of the occurrence of `PES Events` as a result of the invocation of the operations listed in the following table.

INTERFACE	OPERATION	REQUEST DATA TYPE	RETURN DATA TYPE	FAULT CONDITIONS
E-Prescribing	prescribe	Prescription Available Event	Prescription Available Ack	invalidProviderId invalidTimestamp duplicateRK duplicateDocumentId duplicateProcessId notAuthenticated notAuthorised invalidMessage
E-Prescribing	cancelPrescription	Prescription Cancelled Event	Prescription Cancelled Ack	invalidProviderId invalidTimestamp unknownRetrievalKey cancelled expired notAuthenticated notAuthorised invalidMessage
E-Dispensing	initiateDispense	Dispense Initiated Event	Dispense Initiated Ack	invalidProviderId invalidTimestamp unknownRetrievalKey duplicateProcessId fullyDispensed cancelled expired notAuthenticated notAuthorised invalidMessage
E-Dispensing	finaliseDispense	Dispense Finalised Event	Dispense Finalised Ack	invalidProviderId invalidTimestamp unknownRetrievalKey duplicateDocumentId unmatchedProcessId invalidProcessId notAuthenticated notAuthorised invalidMessage

INTERFACE	OPERATION	REQUEST DATA TYPE	RETURN DATA TYPE	FAULT CONDITIONS
E-Dispensing	terminateDispense	Dispense Terminated Event	Dispense Terminated Ack	invalidProviderId invalidTimestamp unknownRetrievalKey unmatchedProcessId invalidProcessId notAuthenticated notAuthorised invalidMessage
E-Dispensing	reverseDispense	Dispense Reversed Event	Dispense Reversed Ack	invalidProviderId invalidTimestamp unknownRetrievalKey unmatchedProcessId invalidProcessId notAuthenticated notAuthorised invalidMessage
E-Dispensing	cancelPrescription	Prescription Cancelled Event	Prescription Cancelled Ack	invalidProviderId invalidTimestamp unknownRetrievalKey cancelled expired notAuthenticated notAuthorised invalidMessage

Table 4: PES Events

The following conformance points in this section apply for each operation listed in Table 4.

ETP 25   A Prescription Exchange shall detect the fault conditions listed in Table 4, and defined in Table 3, and, except when doing so would impose an unacceptable processing load on the Prescription Exchange, shall respond to the invoker with a fault of the corresponding type.

ETP 26   If a Prescription Exchange does not detect a fault then it shall return to the invoker an instance of return data type that is listed in Table 4.

ETP 27   A Prescription Exchange shall respond either with a fault or with an acknowledgement without unnecessary delay and in accordance with a service level agreement expressed in terms of the PesEventResponse metric.

ETP 28   A Prescription Exchange shall, for a defined time period, detect service requests which communicate the occurrence of a PES Event that has been previously recorded by the Prescription Exchange:

ETP 28-1   Duplicates are PES Events where the pesEventIds match

ETP 28-2   The time period during which duplicate detection is required starts when the Prescription Exchange receives the original request and continues for a period specified in a service level agreement

and which is not less than the value of the `MaxRetryPeriod` that is defined in ETP 16

ETP 28-3 A `Prescription Exchange` shall ensure that the maximum period during which retries are attempted shall comply with a service level agreement expressed in terms of the `MaxRetryPeriod` metric.

ETP 29 A `Prescription Exchange` shall set the status attribute of the `PES Event Acknowledgement` data type to `duplicate` if acknowledging a duplicate else it shall set it to `ok`.

ETP 30 A `Prescription Exchange` shall create audit log entries in stable storage for every `PES Event` received. These entries shall comprise:

ETP 30-1   The name of the interface over which it was received

ETP 30-2   The attributes of the base `PES Event` data type

ETP 30-3   The specific type of the `PES Event`

ETP 30-4   if no fault is raised, the attributes of the base `PES Event Acknowledgement` data type; or

ETP 30-5   If a fault is raised, the fault condition

4.5.6   State Changes as a result of PES Events

`PES Events` that are processed by a `Prescription Exchange` without raising a fault will change the state that is managed by that `Prescription Exchange` for the prescription that is referenced by that `PES Event`. This state is represented as an instance of the `PE Prescription State` class defined in section 4.5.2. The conformance points in this section define the effect of a `PES Event` on the attributes of this class.

ETP 31 If no fault is raised from the invocation of the `prescibe` operation of the `E-Prescribing` interface then the `Prescription Exchange` shall update its stable storage to reflect the following resulting values for the referenced instance of the `PE Prescription State` class:

PE PRESCRIPTION STATE ATTRIBUTE	RESULTING VALUE
qualifiedRK	Prescription Available Event.qualifiedRK
dispensingState	availableToDispense
expiryDate	Prescription Available Event.expiryDate
dispensed	0
remaining	Prescription Available Event.numberOfDispensings
currentInterests	Empty list
eDispenseDocs	Empty list
ePrescriptionDoc	Prescription Available Event.prescriptionDoc

ETP 32 If no fault is raised from the invocation of the `cancelPrescription` operation of the `E-Prescribing` interface then the `Prescription Exchange` shall update its stable storage to reflect the following resulting values for the referenced instance of the `PE Prescription State` class:

PE PRESCRIPTION STATE ATTRIBUTE	RESULTING VALUE
qualifiedRK	No change
dispensingState	cancelled
expiryDate	No change
dispensed	No change
remaining	No change
currentInterests	No change
No change	No change
ePrescriptionDoc	No change

ETP 33   If no fault is raised from the invocation of the `cancelPrescription` operation of the `E-Dispensing` interface then the `Prescription Exchange` shall update its stable storage to reflect the following resulting values for the referenced instance of the `PE Prescription State` class:

PE PRESCRIPTION STATE ATTRIBUTE	RESULTING VALUE
qualifiedRK	No change
dispensingState	cancelled
expiryDate	No change
dispensed	No change
remaining	No change
currentInterests	No change
No change	No change
ePrescriptionDoc	No change

ETP 34   If no fault is raised from the invocation of the `initiateDispense` operation of the `E-Dispensing` interface then the `Prescription Exchange` shall update its stable storage to reflect the following resulting values for the referenced instance of the `PE Prescription State` class:

PE PRESCRIPTION STATE ATTRIBUTE	RESULTING VALUE
qualifiedRK	No change
dispensingState	No change
expiryDate	No change
dispensed	No change
remaining	No change
currentInterests	Add one element where currentInterests.clincian = Dispense Initiated Event.clinican currentInterests.organisation = Dispense Initiated Event.organisation currentInterests.clincialProcessId = Dispense Initiated Event.clincialProcessId currentInterests.timestamp = Dispense Initiated

	Event.creationTimestamp
eDispenseDocs	No change
ePrescriptionDoc	No change

ETP 35   If no fault is raised from the invocation of the `finaliseDispense` operation of the `E-Dispensing` interface then the `Prescription Exchange` shall update its stable storage to reflect the following resulting values for the referenced instance of the `PE Prescription State` class:

PE PRESCRIPTION STATE ATTRIBUTE	RESULTING VALUE
qualifiedRK	No change

dispensingState	If dispensingState != "availableToDispense" then no change;
	
	else
	
	If (remaining - 1 - Dispense Finalised Event.additionalReduction) <= 0 then
	
	dispensingState = "fullyDispensed"

expiryDate	No change
dispensed	dispensed + 1;
remaining	remaining - 1 - Dispense Finalised Event.additionalReduction
currentInterests	Delete the element where currentInterests.clincialProcessId = Dispense Finalised Event.clincialProcessId
eDispenseDocs	Add Dispense Finalised Event.dispenseDoc to the list
ePrescriptionDoc	No change

Note that (in accordance with ETP 25) the fault `unmatchedProcessId` (defined in Table 3) will be raised by the `Prescription Exchange` if no `currentInterests` element exists where `currentInterests.clincialProcessId` = `Dispense Finalised Event.clincialProcessId`.

ETP 36   If no fault is raised from the invocation of the `terminateDispense` operation of the `E-Dispensing` interface then the `Prescription Exchange` shall update its stable storage to reflect the following resulting values for the referenced instance of the `PE Prescription State` class:

PE PRESCRIPTION STATE ATTRIBUTE	RESULTING VALUE
qualifiedRK	No change
dispensingState	No change
expiryDate	No change
dispensed	No change
remaining	No change
currentInterests	Delete the element where

	currentInterests.clincialProcessId =
	Dispense Terminated Event.clincialProcessId
eDispenseDocs	No change
ePrescriptionDoc	No change

Note that (in accordance with ETP 25) the fault `unmatchedProcessId` (defined in Table 3) will be raised by the `Prescription Exchange` if no `currentInterests` element exists where `currentInterests.clincialProcessId` = `Dispense Terminated Event.clincialProcessId`.

ETP 37   If no fault is raised from the invocation of the `reverseDispense` operation of the `E-Dispensing` interface then the `Prescription Exchange` shall update its stable storage to reflect the following resulting values for the referenced instance of the `PE Prescription State` class:

PE PRESCRIPTION STATE ATTRIBUTE	RESULTING VALUE
qualifiedRK	No change
dispensingState	If (remaining + 1 + additionalReduction) > 0 then dispensingState = "availableToDispense" Where additionalReduction is the value specified in the Dispense Finialised Event that is being reversed
expiryDate	No change
dispensed	dispensed - 1;
remaining	remaining + 1 + additionalReduction Where additionalReduction is the value specified in the Dispense Finalised Event that is being reversed
currentInterests	No change
eDispenseDocs	Remove the element of the list that was added by the Dispense Finalised Event that is being reversed
ePrescriptionDoc	No change

4.5.7   Other Operations provided

4.5.7.1   E-Prescribing::retrievePrescription

The conformance points in this section apply to the processing of invocations of the `retrievePrescription` operation of the `E-Prescribing` interface.

ETP 38   A `Prescription Exchange` shall detect the following fault conditions: `invalidMessage`, `invalidProviderId`, `invalidTimestamp` and `unknownRetrievalKey` (defined in Table 3) and shall respond to the invoker with the corresponding fault type.

ETP 39   If the `Prescription Exchange` does not detect a fault then it shall respond with an instance of the `Prescription Retrieval Response` data type.

ETP 40   A `Prescription Exchange` shall respond either with a fault or with a `Prescription Retrieval Response` without unnecessary delay and within a time period specified as part of a service level agreement expressed in terms of the `RetrievalResponse` metric.

ETP 41   A `Prescription Exchange` shall create audit log entries in stable storage for every `Retrieval Request` received. These entries shall comprise:

ETP 41-1   An indication that the `Retrieval Request` was received via the `E-Prescribing::retrievePrescription` interface

ETP 41-2   The attributes of the base `Retrieval Request` data type

ETP 41-3   The specific type of the `Retrieval Request,` and either

ETP 41-4   if no fault is raised, the attributes of the base `Retrieval Response` data type; or

ETP 41-5   If a fault is raised, the fault condition

4.6   Prescription Exchange acting as intermediary

This ETP specification supports two different technical "interworking models" that both support the policy that any prescription can be dispensed at any pharmacy.

These models provide `Electronic Dispensing Systems` with two ways of interworking with `Prescription Exchanges`:

1. ***Client based interworking***: in this model an `Electronic Dispensing System` interworks with multiple `Prescription Exchanges`. From the business view, the *Dispenser Organisation* may therefore be required to have business relationships with all `Prescription Exchange` operators. See Figure 40.

2. ***Server based interworking***: in this model an `Electronic Dispensing System` only interworks with one `Prescription Exchange`. This requires that each `Prescription Exchange` act as an intermediary for another `Prescription Exchange` whenever an `E-Dispensing` interface operation is invoked for a prescription that it does not manage. From the business view, the *Dispenser* is only required to have a business relationship with one `Prescription Exchange` operator. See Figure 41.
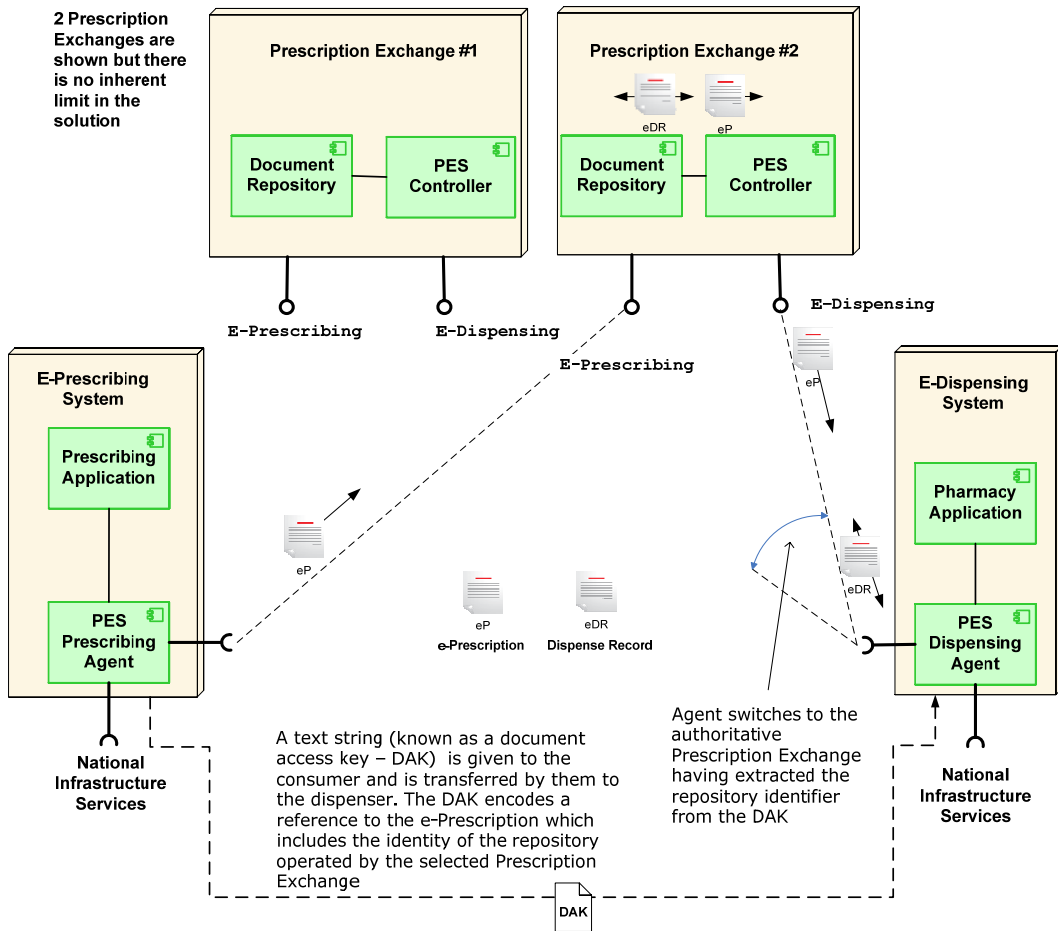
Figure 40: Client Based Interworking
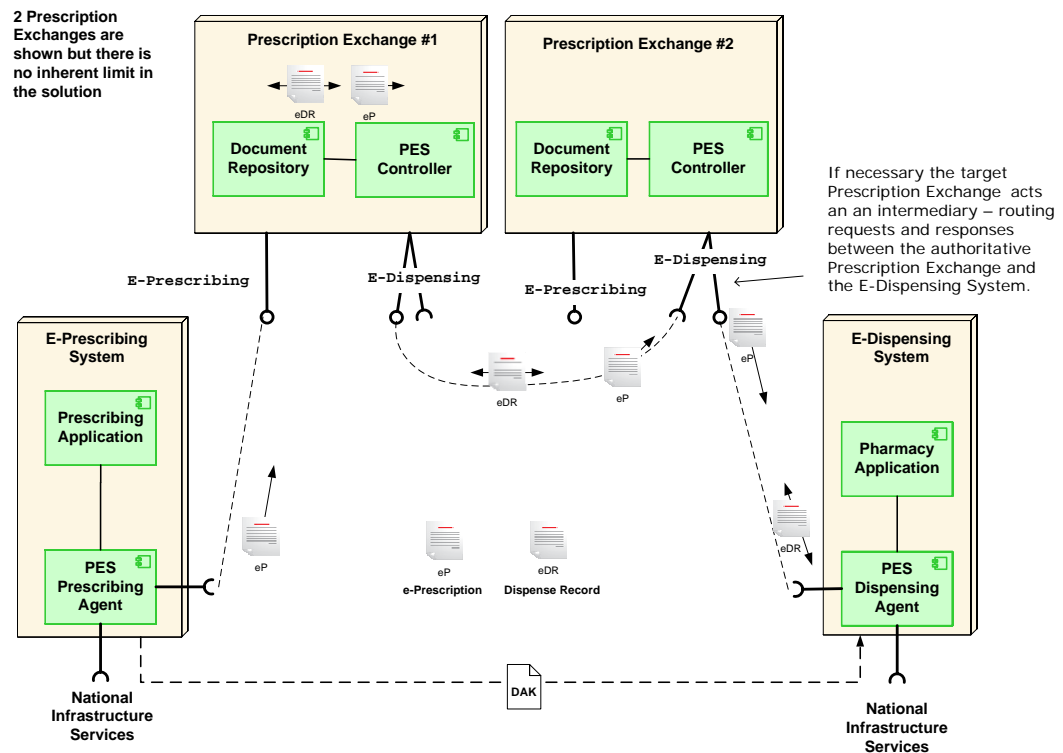
Figure 41: Server based interworking

The conformance points listed in this section are necessary to support this "server based interworking" model. They specify how a `Prescription Exchange` must process `E-Dispensing` interface operations that identify a prescription that is managed by a different `Prescription Exchange`.

ETP 42   A `Prescription Exchange` shall act as an intermediary when it receives an `E-Dispensing` operation request where the `Qualified RK` contains a `providerId` that is associated with a different `Prescription Exchange`.

ETP 43   If the `providerId` is not associated with a known `Prescription Exchange` operator, a `Prescription Exchange` shall reject the request and inform the requesting `Electronic Dispensing System` that a fault condition of type `invalidProviderId` (as defined in Table 3) occurred.

The remaining conformance points listed in this section apply when the `Prescription Exchange` (acting as an intermediary) does not detect faults in accordance with ETP 43.

ETP 44   A `Prescription Exchange` (acting as an intermediary) shall route a service request to the `Prescription Exchange` that is the manager of the prescription referenced by the service request - that is the `Prescription Exchange` associated with the `providerId` identified in the `Qualified RK` that is specified in the service request message.

ETP 45   A `Prescription Exchange` (acting as an intermediary) shall route all service response messages received from the `Prescription Exchange` managing the prescription to the `Electronic Dispensing System` that initiated the original service request message.

ETP 46   A `Prescription Exchange` (acting as an intermediary) shall route all request and response messages without unnecessary delay and in accordance with a service level agreement expressed in terms of the `ForwardingDelay` metric.

ETP 47   A `Prescription Exchange` (acting as an intermediary) shall not retry service invocations (note that the invoking `Electronic Dispensing System` is responsible for retries).

4.7   Electronic Prescribing System

4.7.1   Introduction

An `Electronic Prescribing System` is a component of a clinical information system that is used by a *Prescriber* to prescribe medications.

This document does not provide a complete specification of an `Electronic Prescribing System` - it only specifies the behaviour that is required for an `Electronic Prescribing System` to participate in the ETP services.

4.7.2   Digital Prescriber Signatures

The approved form of a digital signature for an `E-Prescription Document` will be specified in the Electronic Transfer of Prescription Release 1.1 – e-Prescription Clinical Document Architecture Implementation Guide [ETP-EP_CDAIG2010]. This section only provides high level conformance points.

ETP 48   Each `E-Prescription Document` shall be digitally signed by the *Prescriber* who is responsible for the prescription under the relevant legislation.

ETP 49   An `Electronic Prescribing System` shall render the contents of each `E-Prescription Document` for display to the *Prescriber* and

shall provide the ability for the *Prescriber* to review the displayed contents and to choose either to approve or to reject the `E-Prescription Document` while viewing the display.

ETP 50   An `Electronic Prescribing System` shall apply a digital signature to an `E-Prescription Documents` if and only if it has been displayed to the *Prescriber* and approved by the *Prescriber.*

4.7.3   Interfaces

ETP 51   An `Electronic Prescribing System` shall implement a provider of the `Prescription Requesting` interface.

ETP 52   An `Electronic Prescribing System` shall implement an invoker for the `Prescription Receiving` interface.

ETP 53   An `Electronic Prescribing System` shall implement an invoker for the `E-Prescribing` interface.

ETP 54   An `Electronic Prescribing System` may implement an invoker for the `Facility Managed Supply` interface.

4.7.4   Authentication and Authorisation

Authorisation policies are provided in Appendix B however these policies may be varied under the governance of the *ETP Governance Authority.*

ETP 55   An `Electronic Prescribing System` shall authenticate the organisation that is responsible for each service request in accordance with 4.4.1.

ETP 56   An `Electronic Prescribing System` shall authorise service requests in accordance with policies determined by the *ETP Governance Authority*[14].

ETP 57   An `Electronic Prescribing System` shall individually authenticate and authorise it local users in accordance with policies determined by the *Prescribing Organisation* which those users represent*.*

4.7.5   E-Prescribing Service

In accordance with the service contract for the `E-Prescribing Service,` the `Electronic Prescribing System` invokes the `E-Prescribing` interface provided by a `Prescription Exchange.`

4.7.5.1   E-Prescribing::prescribe (invoker)

ETP 58   An `Electronic Prescribing System` shall generate a `DAK` for each prescription to be sent to a `Prescription Exchange.`

ETP 59   An `Electronic Prescribing System` shall generate a `DAK` using a cryptographically secure random number generator to obtain a value for the `randomValue` attribute of the `DAK` and setting the `providerId` component of the `DAK` to identify the operator of the target `Prescription Exchange` that will manage the prescription.

ETP 60   An `Electronic Prescribing System` shall derive a `Retrieval Key` from a `DAK` using a cryptographically secure pseudo random number function as a one-way function that takes the DAK as an input parameter.

---

[14]   The initial set of policies is included in Appendix B.

ETP 61   An `Electronic Prescribing System` shall derive a `Cipher Key` from a `DAK` using a cryptographically secure pseudo random number function as a one-way function that takes the DAK as an input parameter.

ETP 62   An `Electronic Prescribing System` shall invoke the `E-Prescribing::prescribe` operation of a `Prescription Exchange` to make a prescription available electronically to `Electronic Dispensing Systems`; the service request shall be a valid instance of the `Prescription Available Event` data type.

ETP 63   With respect to the instance of the `Prescription Available Event` data type created for the `prescribe` request, an `Electronic Prescribing System` shall construct the `prescriptionDoc.encryptedPayload` by encrypting the payload with the `Cipher Key` derived from the DAK and shall set `prescriptionDoc.metadata.qualifiedRK.retrievalKey` to the `retrievalKey` attribute of the same DAK and shall set `prescriptionDoc.metadata.qualifiedRK.providerId` to the `providerId` attribute of the same DAK.

ETP 64   If an `Electronic Prescribing System` receives a response that indicates a successful invocation then the `Electronic Prescribing System` shall record, in stable storage, the DAK that was associated with the request including its association with the representation of the `Prescription Available Event` that is maintained as part of the local healthcare record.

4.7.5.2   E-Prescribing:: retrievePrescription (invoker)

ETP 65   An `Electronic Prescribing System` shall allow an authorised user to select a previously issued prescription issued for the subject of that local health record and to retrieve the associated `E-Prescription Document` from the `Prescription Exchange` that manages it and display it to the user.

ETP 66   An `Electronic Prescribing System` shall retrieve the `E-Prescription Document` from the `Prescription Exchange` by invoking the `E-Prescribing::retrievePrescription` operation of a `Prescription Exchange`; the service request shall be a valid instance of the `Prescription Retrieval Request` data type.

4.7.5.3   E-Prescribing:: cancelPrescription (invoker)

ETP 67   An `Electronic Prescribing System` shall allow an authorised user to select a prescription previously issued and to cancel that prescription thereby making the `E-Prescription Document` no longer available for retrieval by an `Electronic Dispensing System`.

ETP 68   An `Electronic Prescribing System` shall cancel the prescription by invoking the `E-Prescribing::cancelPrescription` operation of the `Prescription Exchange` that manages the prescription; the service request shall be a valid instance of the `Prescription Cancellation Event` data type

ETP 69   An `Electronic Prescribing System` shall warn the user that a successful cancellation of a prescription does not relieve the user of their responsibility to inform the *Prescription Subject* that the prescription has been cancelled.

ETP 70   If no fault is received, the `Electronic Prescribing System` shall:

ETP 70-1   Record, in local stable storage, an indication that the prescription was successfully cancelled

ETP 70-2   Inform the user that the cancellation was successful.

4.7.6   Prescription Requesting Service

In accordance with the service contract for the `Prescription Requesting Service`, the `Electronic Prescribing System` implements the `Prescription Requesting` interface and invokes the `Prescription Receiving` interface provided by an `Electronic Dispensing System`.

4.7.6.1   Prescription Requesting:: deliverNotification (provider)

ETP 71   An `Electronic Prescribing System` shall respond to all received invocations of the `deliverNotification` operation of the `Prescription Requesting` interface with an instance of the ETP `Notification Ack` data type or with a fault.

ETP 72   An `Electronic Prescribing System` shall, except when doing so would impose an unacceptable processing load, respond with a fault in the following circumstances:

ETP 72-1   If authentication as described in section 4.7.4 fails then the fault `notAuthenticated` shall be raised

ETP 72-2   If authorisation as described in section 4.7.4 fails then the fault `notAuthorised` shall be raised

ETP 72-3   If the message received via the `deliverNotification` operation of the `Prescription Requesting` interface is not a valid instance of the `Prescription Required Notification` data type then the fault `invalidMessage` shall be raised.

ETP 73   An `Electronic Prescribing System` shall inform the *Prescriber* who is identified in a `Prescription Required Notification` when a `Prescription Required Notification` is received via the `Prescription Requesting` interface without fault.

ETP 74   For each `Prescription Required Notification` received without faults via the `Prescription Required` interface the `Electronic Prescribing System` shall:

ETP 74-1   Display to the *Prescriber*, who is identified in the `Prescription Required Notification`, the details of a received `Prescription Required Notification`

ETP 74-2   Allow the *Prescriber* to choose to create an `E-Prescription Document` that is populated using the values of the corresponding elements of the `Prescription Required Notification`. The `Electronic Prescribing System` shall allow any elements populated from the `Prescription Required Notification` to be overwritten by the *Prescriber*.

ETP 74-3   Allow the *Prescriber* to choose to make the `E-Prescription Document` available by invoking the `prescribe` operation of the `E-Prescribing` interface in accordance with section 4.7.5.1.

ETP 74-4   Allow the *Prescriber* to choose to respond to the *Dispenser* responsible for the `Prescription Required Notification` in accordance with section 4.7.6.2.

ETP 75   An `Electronic Prescribing System` shall create audit log entries in stable storage for every `Prescription Required Notification` received. These entries shall comprise:

ETP 75-1   An indication that the `Prescription Required Notification` was received via the `Prescription Required` interface

ETP 75-2   The attributes of the base `ETP Notification` data type

ETP 75-3   The specific type of the `ETP Notification,` and either

ETP 75-4   if no fault is raised, the attributes of the base `ETP Notification Ack` data type; or

ETP 75-5   If a fault is raised, the fault condition.

4.7.6.2   Prescription Receiving:: deliverNotification (invoker)

ETP 76   If instructed by the *Prescriber,* an `Electronic Prescribing System` shall respond to a received `Prescription Required Notification` by sending a `Required Prescription Notification` to the *Dispenser Organisation* that is identified in the `Prescription Required Notification.`

ETP 77   A `Required Prescription Notification` sent in response to a `Prescription Required Notification` shall be an instance of the `Required Prescription Notification` data type and shall:

ETP 77-1   Contain a value for the `clinicalProcessId` attribute that is set to the same value as the `clinicalProcessId` of the `Prescription Required Notification`

ETP 77-2   Contain a value for the `DAK` attribute that is associated with an `E-Prescription Document` that has been previously successfully sent to a *Prescription Exchange* by invoking the `prescribe` operation of the `E-Prescribing` interface (see section 4.7.5.1).

Note that this `DAK` may reference any `E-Prescription Document` created by the *Prescriber* regardless of whether or not it was prepared in accordance with the conformance points listed in section 4.7.6.2.

ETP 78   An `Electronic Prescribing System` shall send a `Required Prescription Notification` by invoking the `deliverNotification` operation of the `Prescription Receiving` interface provided by the *Dispenser Organisation*.

4.7.7   Facility Managed Supply Service (optional)

If an `Electronic Prescribing System` implements an invoker for the optional `Facility Managed Supply` interface then it may participate in the `Facility Managed Supply Service`.

4.7.7.1   Facility Managed Supply::deliverNotification (invoker)

The conformance points in this section only apply if an `Electronic Prescribing System` implements an invoker of the optional `Facility Managed Supply` interface.

ETP 79   An `Electronic Prescribing System` shall record, for each *Prescription Subject* that has consented to the use of the `Facility`

Managed Supply Service for their prescriptions, a record that consent was granted to an identified healthcare organisation that provides the Facility Managed Supply Service.

ETP 80   After a successful invocation of the E-Prescribing::prescribe operation, an Electronic Prescribing System may construct an Electronic Prescription Notification and send it to the Facility Based Supply Manager consented to by the *Prescription Subject*.

ETP 81   An Electronic Prescribing System shall send an Electronic Prescription Notification to a Facility Based Supply Manager by invoking the Facility Managed Supply::deliverNotification interface.

4.8   Electronic Dispensing System

4.8.1   Introduction

An Electronic Dispensing System is a component of a clinical information system that is used by a *Dispenser* to dispense medications.

This Solution Specification does not provide a complete specification of an Electronic Dispensing System - it only specifies the behaviour that is required for an Electronic Dispensing System to participate in the ETP services.

4.8.2   Interfaces

ETP 82   An Electronic Dispensing System shall implement a provider of the Prescription Receiving interface.

ETP 83   An Electronic Dispensing System shall implement an invoker for the Prescription Requesting interface.

ETP 84   An Electronic Dispensing System shall implement an invoker for the E-Dispensing interface.

ETP 85   An Electronic Dispensing System may implement a provider of the Contract Dispensing interface.

ETP 86   An Electronic Dispensing System may implement an invoker for the Last Supply Notification interface.

4.8.3   Authentication and Authorisation

Authorisation policies are provided in Appendix B however these policies may be varied under the governance of the *ETP Governance Authority.*

ETP 87   An Electronic Dispensing System shall authenticate the organisation that is responsible for invoking each service request in accordance with 4.4.1.

ETP 88   An Electronic Dispensing System shall authorise service requests in accordance with policies determined by the *ETP Governance Authority*.

ETP 89   An `Electronic Dispensing System` shall individually authenticate and authorise it local users in accordance with policies determined by the *Dispensing Organisation* which those users represent.

4.8.4   Prescription Requesting Service

In accordance with the service contract for the `Prescription Requesting Service`, the `Electronic Dispensing System` implements the `Prescription Receiving` interface and invokes the `Prescription Requesting` interface provided by an `Electronic Prescribing System`.

The selection of the target `Electronic Prescribing System` instance is outside the scope of this specification – the expectation is that either the `Electronic Dispensing System` will use the Endpoint Location Service directly or it will use a SMD Sender Intermediary.

4.8.4.1   Prescription Receiving::deliverNotification (provider)

ETP 90   An `Electronic Dispensing System` shall respond to all received invocations of the `deliverNotification` operation of the `Prescription Receiving` interface with an instance of the `ETP Notification Ack` data type or with a fault.

ETP 91   An `Electronic Dispensing System` shall, except when doing so would impose an unacceptable processing load, respond with a fault in the following circumstances:

> ETP 91-1   If authentication as described in section 4.8.3 fails then the fault `notAuthenticated` shall be raised

> ETP 91-2   If authorisation as described in section 4.8.3 fails then the fault `notAuthorised` shall be raised

> ETP 91-3   If the message received via the `deliverNotification` operation of the `Prescription Receiving` interface is not a valid instance of the `Required Prescription Notification` data type then the fault `invalidMessage` shall be raised

> ETP 91-4   If the `clinicalProcessId` of the `Required Prescription Notification` does not match the `clinicalProcessId` of a `Prescription Required Notification` that was previously sent by the `Electronic Dispensing System` to the `Electronic Prescribing System` (which invoked the operation) then the `unmatchedProcessId` fault shall be raised.

ETP 92   An `Electronic Dispensing System` shall inform a *Dispenser* when a `Required Prescription Notification` is received without faults via the `Prescription Receiving` interface.

ETP 93   An `Electronic Dispensing System` shall allow the *Dispenser* to reconcile a prescription identified in the `Required Prescription Notification` received without faults via the `Prescription Receiving` interface with the prior local record of the dispensing. Matching shall be based on the `clinicalProcessId` of the `Required Prescription Notification`.

ETP 94   For each `Required Prescription Notification` received without faults via the `Prescription Receiving` interface the `Electronic Dispensing System` shall:

ETP 94-1   Allow the *Dispenser* to download the *PES Documents* by invoking the `E-Dispensing::initiateDispense` operation in accordance with section 4.8.5.1.

ETP 94-2   Allow the *Dispenser* to compare the *PES Documents* and local records and allow the *Dispenser* to record the result of the reconciliation as either successful or not successful

ETP 94-3   Inform the *Dispenser* for cases where reconciliation is not successful that they must take appropriate action to resolve discrepancies.

ETP 94-4   For cases where reconciliation is successful, record the finalisation of the dispensing process by invoking the `E-Dispensing::finaliseDispense` operation in accordance with section 4.8.5.2

ETP 94-5   For cases where reconciliation is not successful, record the termination of the dispensing process by invoking the `E-Dispensing::terminateDispense` operation in accordance with section 4.8.5.3.

ETP 95   An `Electronic Dispensing System` shall create audit log entries in stable storage for every `Required Prescription Notification` received. These entries shall comprise:

ETP 95-1   An indication that the `Required Prescription Notification` was received via the `Prescription Receiving Dispensing` interface

ETP 95-2   The attributes of the base `ETP Notification` data type

ETP 95-3   The specific type of the `ETP Notification,` and either

ETP 95-4   if no fault is raised, the attributes of the base `ETP Notification Ack` data type; or

ETP 95-5   If a fault is raised, the fault condition

ETP 95-6   The result of reconciliation.

4.8.4.2   Prescription Required::deliverNotification (invoker)

ETP 96   Whenever a prescription medication is dispensed without an `E-Prescription Document, an Electronic Dispensing System` shall:

ETP 96-1   Allow the *Dispenser* to choose to construct a `Prescription Required Notification` based on the local record of dispensing, and, if the *Dispenser* so chooses*:*

ETP 96-2   Allow the *Dispenser* to identify the *Prescriber* and *Prescriber Organisation*

ETP 96-3   Allow the *Dispenser* to choose to send the `Prescription Required Notification`

ETP 96-4   If so instructed by the *Dispenser,* the `Electronic Dispensing System` shall send the `Prescription Required Notification` by invoking the `Prescription Requesting::deliverNotification` operation of the target `Electronic Prescribing System.`

4.8.5   E-Dispensing Service

In accordance with the service contract for the `E-Dispensing Service`, the `Electronic Dispensing System` invokes the `E-Dispensing` interface provided by a `Prescription Exchange`. The selection of the `Prescription Exchange` on which to invoke an `E-Dispensing` operation is outside the scope of this specification.

4.8.5.1   E-Dispensing::initiateDispense (invoker)

ETP 97   An `Electronic Dispensing System` shall invoke the `E-Dispensing::initiateDispense` operation of a `Prescription Exchange` for each dispensing process for which a DAK has been provided; the service request shall be a valid instance of the `Dispense Initiated Event` data type.

ETP 98   An `Electronic Dispensing System` shall derive a `Retrieval Key` and a `Cipher Key` from the DAK provided.

ETP 99   For a successful invovation, the `Electronic Dispensing System` shall record in stable storage, a record a record of the fact that the dispense process identified by the `Dispense Initiated Event` `clinicalProcessId` attribute was successfully initiated.

ETP 100   If the `Electronic Dispensing System` receives a response where the `Dispense Initiated Ack` contains a non empty `currentInterests` attribute then the `Electronic Dispensing System` shall display the details of each of the `currentInterests` to the *Dispenser* and shall not proceed to dispense the prescription until the *Dispenser* has explicitly instructed the `Electronic Dispensing System` to proceed to dispense that prescription.

4.8.5.2   E-Dispensing::finaliseDispense (invoker)

ETP 101   An `Electronic Dispensing System` shall invoke the `E-Dispensing::finaliseDispense` operation of a `Prescription Exchange` for each dispensing process that was initiated by that `Electronic Dispensing System` (by successfully invoking the `initiateDispense` operation) and was subsequently successfully completed.

ETP 102   A `Dispensing::finaliseDispense` operation service request shall be a valid instance of the `Dispense Finalised Event` data type where the `clinicalProcessId` element matches that of the `Dispense Initiated Event` previously sent via the successfully invoked `initiateDispense` operation.

ETP 103   With respect to the instance of the `Dispense Finalised Event` data type created for the `finaliseDispense` request, an `Electronic Dispensing Systems` shall construct the `dispenseDoc.encryptedPayload` by encrypting the payload with the `Cipher Key` derived from the DAK provided in accordance ETP 97 and shall set `dispenseDoc.metadata.qualifiedRK.retrievalKey` to the `retrievalKey` attribute of the same DAK and shall set `dispenseDoc.metadata.qualifiedRK.providerId` to the `providerId` attribute of the same DAK.

ETP 104   For a successful invovation, the `Electronic Dispensing System` shall record, in stable storage, a record of the fact that the dispense

process identified by the `Dispense Finalised Event`
`clinicalProcessId` attribute was successfully finalised.

4.8.5.3   E-Dispensing::terminateDispense (invoker)

ETP 105   An `Electronic Dispensing System` shall invoke the `E-Dispensing::terminateDispense` operation of a `Prescription Exchange` for each dispensing process that was initiated by that `Electronic Dispensing System` (by successfully invoking the `initiateDispense` operation) and was subsequently concluded without dispensing any medications.

ETP 106   A `Dispensing::terminateDispense` operation service request shall be a valid instance of the `Dispense Terminated Event` data type where the `clinicalProcessId` element matches that of the `Dispense Initiated Event` previously sent via the successfully invoked `initiateDispense` operation.

ETP 107   For a successful invovation, the `Electronic Prescribing System` shall record, in stable storage, a record of the fact that the dispense process identified by the `Dispense Terminated Event` `clinicalProcessId` attribute was successfully terminated.

4.8.5.4   E-Dispensing::reverseDispense (invoker)

ETP 108   An `Electronic Prescribing System` shall invoke the `E-Dispensing::reverseDispense` operation of a `Prescription Exchange` for each dispensing process that was initiated by that `Electronic Prescribing System` (by successfully invoking the `initiateDispense` operation) and was subsequently successfully finalised (by invoking the `finaliseDispense` operation) and then where the dispensing was subsequently reversed without supplying the medications.

ETP 109   A `Dispensing::reverseDispense` operation service request shall be a valid instance of the `Dispense Reversed Event` data type where the `clinicalProcessId` element matches that of the `Dispense Initiated Event` previously sent via the successfully invoked `initiateDispense` operation.

ETP 110   For a successful invovation, the `Electronic Dispensing System` shall record, in stable storage, a record of the fact that the dispense process identified by `Dispense Reversed Event` `clinicalProcessId` attribute was successfully reversed.

4.8.5.5   E-Dispensing:: cancelPrescription (invoker)

ETP 111   An `Electronic Dispensing System` shall allow its authorised users to cancel a prescription for which they have been provided the DAK.

ETP 112   An `Electronic Dispensing System` shall cancel the prescription by invoking the `E-Dispensing::cancelPrescription` operation of the `Prescription Exchange` that manages the prescription; the service request shall be a valid instance of the `Prescription Cancellation Event` data type

ETP 113   An `Electronic Dispensing System` shall warn the user that a successful cancellation of a prescription does not relieve the user of their responsibility to inform the *Prescription Subject* that the prescription has been cancelled.

ETP 114   For a successful invovation, the `Electronic Dispensing System` shall:

ETP 114-1 Record, in local stable storage, an indication that the prescription was successfully cancelled

ETP 114-2 Inform the user that the cancellation was successful.

4.8.6   Contract Dispensing Service (optional)

If an `Electronic Dispensing System` supports the optional `Contract Dispensing` interface then it may participate in the `Contract Dispensing Service.`

4.8.6.1   Contract Dispensing::deliverNotification (provider)

The conformance points in this section only apply if an `Electronic Dispensing System` implements the optional `Contract Dispensing` interface.

ETP 115   An `Electronic Dispensing System` shall respond to all received invocations of the `deliverNotification` operation of the `Contract Dispensing` interface with an instance of the ETP `Notification Ack` data type or with a fault.

ETP 116   An `Electronic Dispensing System` shall, except when doing so would impose an unacceptable processing load, respond with a fault in the following circumstances:

ETP 116-1 If authentication as described in section 4.8.3 fails then the fault `notAuthenticated` shall be raised

ETP 116-2 If authorisation as described in section 4.8.3 fails then the fault `notAuthorised` shall be raised

ETP 116-3 If the message received via the `deliverNotification` operation of the `Contract Dispensing` interface is not a valid instance of the `Prescription Available Notification` data type then the fault `invalidMessage` shall be raised.

ETP 117   An `Electronic Dispensing System` shall inform a *Dispenser* when a `Prescription Available Notification` is received without faults via the `Contract Dispensing` interface.

ETP 118   For each `Prescription Available Notification` received without faults via the `Contract Dispensing` interface an `Electronic Dispensing System` shall allow the *Dispenser* to manage a prescription identified in a `Prescription Available Notification`, and:

ETP 118-1 Allow the *Dispenser* to download the *PES Documents* by invoking the `E-Dispensing::initiateDispense` operation in accordance with section 4.8.5.1.

ETP 118-2 For a dispense process that was finalised, record the finalisation of the dispensing process by invoking the `E-Dispensing::finaliseDispense` operation in accordance with section 4.8.5.2

ETP 118-3 For a dispense process that was abandoned prior to dispensing, record the termination of the dispensing process by invoking the `E-Dispensing::terminateDispense` operation in accordance with section 4.8.5.3

ETP 118-4 For a dispense process the was successfully finalised but for which supply was not made, record the reversal of the dispensing process by invoking the `E-Dispensing::reverseDispense` operation in accordance with section 4.8.5.4.

ETP 119   An `Electronic Dispensing System` shall create audit log entries in stable storage for every `Prescription Available Notification` received via the `Contract Dispensing`. These entries shall comprise:

ETP 119-1   An indication that the `Prescription Available Notification` was received via the `Contract Dispensing` interface

ETP 119-2 The attributes of the base `ETP Notification` data type, and

ETP 119-3 The specific type of the `ETP Notification,` and either

ETP 119-4 if no fault is raised, the attributes of the base `ETP Notification Ack` data type; or

ETP 119-5 If a fault is raised, the fault condition.

4.8.7   Last Supply Notification Service (optional)

If an `Electronic Dispensing System` implements the optional invoker for `Last Supply Notification` interface then it may participate in the `Last Supply Notification Service`.

4.8.7.1   Last Supply Notification::deliverNotification (invoker)

The conformance points in this section only apply if the `Electronic Dispensing System` implements the optional invoker for the `Last Supply Notification` interface.

ETP 120   An `Electronic Dispensing System` shall record, for each *Prescription Subject* who has consented to the use of the `Last Supply Notification Service` for their prescriptions, a record that consent was granted to an identified healthcare organisation that provides the `Last Supply Notification Service.`

ETP 121   Whenever a prescription is supplied for the last time for a *Prescription Subject* who has consented to the use of the `Last Supply Notification Service` an `Electronic Dispensing System` shall:

ETP 121-1   Allow the *Dispenser* to choose to construct a `Last Supply Notification` based on the local record of dispensing , and, if the *Dispenser* so chooses*:*

ETP 121-2 Allow the *Dispenser* to identity the operator of a `Last Supply Notification Agent`

ETP 121-3 Allow the *Dispenser* to locate the appropriate `Last Supply Notification Agent` instance based on the identity of the organisation

ETP 121-4 Allow the *Dispenser* to choose to send the `Last Supply Notification`

ETP 121-5 If so instructed by the *Dispenser,* the `Electronic Dispensing System` shall send the `Last Supply Notification` by invoking the `Last Supply Notification Requesting::deliverNotification` operation of the target `Last Supply Notification Agent`.

4.9 Facility Based Supply Manager

4.9.1 Introduction

A `Facility Based Supply Manager` may be operated by healthcare facilities that manage the supply of medications for their patients such as residential care facilities and private hospitals.

4.9.2 Interfaces

ETP 122   A `Facility Based Supply Manager` shall implement a provider of the `Facility Managed Supply` interface.

ETP 123   A `Facility Based Supply Manager` shall implement an invoker for the `Contract Dispensing` interface.

4.9.3 Authentication and Authorisation

ETP 124   A `Facility Based Supply Manager` shall authenticate the organisation that is responsible for each service request in accordance with 4.4.1.

ETP 125   A `Facility Based Supply Manager` shall authorise service requests in accordance with the policies determined by the *ETP Governance Authority*.

4.9.4 Facility Managed Supply Service

In accordance with the service contract for the `Facility Managed Service`, a `Facility Based Supply Manager` implements the `Facility Managed Supply` interface.

4.9.4.1 Facility Managed Supply:: deliverNotification (provider)

ETP 126   A `Facility Based Supply Manager` shall respond to all received invocations of the `deliverNotification` operation of the `Facility Managed Supply` interface with an instance of the `ETP Notification Ack` data type or with a fault.

ETP 127   A `Facility Based Supply Manager` shall, except when doing so would impose an unacceptable processing load, respond with a fault in the following circumstances:

ETP 127-1 If authentication as described in section 4.9.3 fails then the fault `notAuthenticated` shall be raised

ETP 127-2 If authorisation as described in section 4.9.3 fails then the fault `notAuthorised` shall be raised

ETP 127-3 If the message received via the `deliverNotification` operation of the `Facility Managed Supply` interface is not a valid

instance of the `Electronic Prescription Notification` data type then the fault `invalidMessage` shall be raised.

ETP 128   A `Facility Based Supply Manager` shall process every `Electronic Prescription Notification` received without faults as follows:

ETP 128-1 If (and only if) the *Prescription Subject,* who is identified in the subject attribute of `Electronic Prescription Notification` has consented to the operator of the `Facility Based Supply Manager` managing their medication supply, then the `Facility Based Supply Manager` may send a `Prescription Available Notification` to a *Contracted Pharmacy.*

ETP 129   An `Electronic Prescribing System` shall create audit log entries in stable storage for every `Electronic Prescription Notification` received. These entries shall comprise:

ETP 129-1 An indication that the `Electronic Prescription Notification` was received via the `Facility Managed Supply` interface

ETP 129-2 The attributes of the base `ETP Notification` data type, and

ETP 129-3 The specific type of the `ETP Notification,` and either

ETP 129-4 if no fault is raised, the attributes of the base `ETP Notification Ack` data type; or

ETP 129-5 If a fault is raised, the fault condition.

4.9.5   Contract Dispensing Service

In accordance with the service contract for the `Contract Dispensing Service,` a `Facility Based Supply Manager` implements an invoker for `Contract Dispensing` interface`.`

4.9.5.1   Contract Dispensing::deliverNotification (invoker)

ETP 130   A `Facility Based Supply Manager` shall (in accordance with ETP 128) send a `Prescription Available Notification` to a *Contracted Pharmacy* by invoking the `deliverNotification` operation of the `Contract Dispensing` interface offered by a *Electronic Dispensing System* operated by (or on behalf of) that *Contracted Pharmacy.*

ETP 131   The `Facility Based Supply Manager` shall construct the `Prescription Available Notification` by populating it from the corresponding elements of the same name of the `Electronic Prescription Notification` received via the `deliverNotification` operation of the `Facility Managed Supply` interface in accordance with section 4.9.4.1

4.10   Last Supply Notification Agent

4.10.1   Introduction

A `Last Supply Notification Agent` acts on behalf of *Prescription Subject* and receives notification of the last supply of a prescription.

4.10.2   Interfaces

ETP 132   A `Last Supply Notification Agent` shall implement a provider of the `Last Supply Notification` interface.

4.10.3   Authentication and Authorisation

ETP 133   A `Last Supply Notification Agent` shall authenticate the organisation that is responsible for each service request in accordance with 4.4.1.

ETP 134   A `Last Supply Notification Agent` shall authorise service requests in accordance with policies determined by the ETP Governance Authority.

4.10.4   Last Supply Notification Service

In accordance with the service contract for the `Last Supply Notification Service`, a `Last Supply Notification Agent` implements the `Last Supply Notification` interface.

4.10.4.1   Last Supply Notification:: deliverNotification (provider)

ETP 135   A `Last Supply Notification Agent` shall respond to all received invocations of the `deliverNotification` operation of the `Last Supply Notification` interface with an instance of the `ETP Notification Ack` data type or with a fault.

ETP 136   A `Last Supply Notification Agent` shall, except when doing so would impose an unacceptable processing load, respond with a fault in the following circumstances:

ETP 136-1 If authentication as described in section 4.10.3 fails then the fault `notAuthenticated` shall be raised

ETP 136-2 If authorisation as described in section 4.10.3 fails then the fault `notAuthorised` shall be raised

ETP 136-3 If the message received via the `deliverNotification` operation of the `Last Supply Notification` interface is not a valid instance of the `Last Supply Notification` data type then the fault `invalidMessage` shall be raised.

ETP 137   For each `Last Supply Notification` received without faults via the `Last Supply Notification` interface the `Last Supply Notification Agent` shall:

ETP 137-1 Determine if the identified *Prescription Subject* has consented to the operator of the `Last Supply Notification Agent` managing their `Last Supply Notification`, and if so then:

ETP 137-2   Allow a user who is authorised by the operator of the `Last Supply Notification Agent` to view the details of the `Last Supply Notification.`

ETP 138   A `Last Supply Notification Agent` shall create audit log entries in stable storage for every `Last Supply Notification` received. These entries shall comprise:

ETP 138-1 An indication that the `Last Supply Notification` was received via the `Last Supply Notification` interface

ETP 138-2 The attributes of the base `ETP Notification` data type

ETP 138-3 The specific type of the `ETP Notification,` and either

ETP 138-4 if no fault is raised, the attributes of the base `ETP Notification Ack` data type; or

ETP 138-5 If a fault is raised, the fault condition.

Definitions

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.
CRL	Certificate Revocation List
EDS	Electronic Dispensing System
EPS	Electronic Prescribing System
ETP	Electronic Transfer of Prescription
HPI-I	Healthcare Provider Identifier – Individual
HPI-O	Healthcare Provider Identifier – Organisation
PCEHR	Personally Controlled Electronic Health Record
IHI	The Individual Healthcare Identifier
NASH	National Authentication Service for Health
NEHTA	National E-Health Transition Authority
OCSP	On-line Certificate Status Profile
PE	Prescription Exchange
PES	Prescription Exchange Service
PKI	Public Key Infrastructure
UML	Unified Modelling 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 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.
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.
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

Term	Description
	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 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.

Term	Description
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 Service - Dispense Record	A dispense record submitted to a Prescription Exchange Service (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.
Reciprocal Healthcare Agreements	The Australian Government has signed Reciprocal Health Care Agreements (RHCA) with the governments of the United Kingdom, Sweden, the Netherlands, Belgium, Finland, Norway, Malta and Italy which entitles visitors to limited subsidised health services for medically necessary treatment while visiting Australia.
Regulation 24	Government legislation that allows the Subject of Care to receive the supply of multiple repeat medication items concurrently. This regulation allows pharmacists to supply a pharmaceutical benefit and all of its repeats at the one time. The prescription must be endorsed by the doctor with the words 'Regulation 24' or 'Reg 24' if it is a medicine supplied under the PBS or 'hardship conditions apply' if it is being supplied under the RPBS.
Repatriation Pharmaceutical Benefits Scheme	The Repatriation Pharmaceutical Benefits Scheme (RPBS) provides a wide range of Pharmaceuticals and dressings at a concession rate for the treatment of eligible veterans, war widows/widowers, and their dependants.
Individual Health Identifier	A unique 16 digit number used to identify individuals who receive care in the Australian Healthcare system.

References

Package Documents

The documents listed below are part of the suite delivered in the Electronic Transfer of Prescription package.

Electronic Transfer of Prescription Package Documents			
[REF]	**Document Name**	**Publisher**	**Link**
[ETP-ES2010]	Electronic Transfer of Prescription Release 1.1 – Executive Summary	NEHTA 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		

References

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

Reference Documents			
[REF]	Document Name	Publisher	Link
[AST5820]	ATS 5820—2010 E-health web services profiles	Standards Australia 2010	http://infostore.saiglobal.com/store/Details.aspx?ProductID=1391033
[AST5821]	ATS 5821—2010 E-health XML secured payload profiles	Standards Australia 2010	http://infostore.saiglobal.com/store/Details.aspx?productID=1391034
[AST5822]	ATS 5822—2010 E-health secure message delivery	Standards Australia 2010	http://infostore.saiglobal.com/store/Details.aspx?ProductID=1391035
[LEVY]	Levy, Henry M., Capability-Based Computer Systems	Digital Equipment Corporation 1984. ISBN 0-932376-22-3	http://www.cs.washington.edu/homes/levy/capabook/
[NeAF]	National e-Authentication Framework	Department of Finance and Deregulation Australian Government Information Management Office	http://www.finance.gov.au/e-government/security-and-authentication/docs/NeAF-framework.pdf
[RFC5280]	RFC 5280 Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile	IETF	http://www.ietf.org/rfc/rfc5280.txt
[SOAML]	Service oriented architecture Modeling Language (SoaML) - Specification for the UML Profile and Metamodel for Services (UPMS) OMG Adopted Specification Finalisation Task Force Beta 2 document (FTF Beta 2)	OMG 2009	http://www.omg.org/spec/SoaML/
[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
[EHSTD]	Standards for E-Health Interoperability, An E-Health Transition Strategy Version 1.0	NEHTA 08/05/2007	http://www.nehta.gov.au/component/docman/doc_download/252-standards-for-e-health-interoperability-v10
[IF2007]	Interoperability Framework v2.0	NEHTA 2008	http://www.nehta.gov.au/ (Home > Publications)

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 Program Manager, Electronic Medication Management, NEHTA	Toby.Mathieson@nehta.gov.au	+61 2 8298 2605

Appendix A: Metrics

This section provides values for metrics that are referenced in this document and are expected to be defined within Service Level Agreements (SLAs). Note these are indicative only; service level agreements will be defined that fully define appropriate metrics and the test conditions in which they apply.

Metric	Target Value Range	Allowable number of test results outside target range
PesEventResponse	<3 seconds	5%
RetrievalResponse	<3 seconds	5%
ForwardingDelay	<0.5 seconds	5%
MinRetryPeriod, MaxRetryPeriod	3 days <= MinRetryPeriod <= MaxRetryPeriod <= 7 days	0%
MaxRetryInterval	<1 hours	5%
MinRetryInterval	>10 seconds	5%

Appendix B: Authorisation policies

Authorisation policies will be determined by the *ETP Governance* authority from time to time – the following initial set of polices shall be supported.

B.1   Prescription Exchange

Note the following policies are applied by the `Prescription Exchange` that manages the prescription.

INTERFACE	OPERATION	POLICY DEFINITION
E-Prescribing	prescribe	The authenticated requesting organisation shall be a *Prescriber Organisation* that is a known party to an agreement with the operator of the `Prescription Exchange`.
E-Prescribing	cancelPrescription	The authenticated requesting organisation shall be a *Prescriber Organisation* that is a known party to an agreement with the operator of the `Prescription Exchange`.
E-Prescribing	retrievePrescription	The authenticated requesting organisation shall be a *Prescriber Organisation* that is a known party to an agreement with the operator of the `Prescription Exchange`.
E-Dispensing	initiateDispense	The authenticated requesting organisation shall be a healthcare organisation.
E-Dispensing	finaliseDispense	The authenticated requesting organisation shall be a healthcare organisation and its HPI-O shall match that of the healthcare organisation responsible for the matching E-`Dispensing::initiateDispense` operation)
E-Dispensing	terminateDispense	The authenticated requesting organisation shall be a healthcare organisation and its HPI-O shall match that of the healthcare organisation responsible for the matching E-`Dispensing::initiateDispense` operation)
E-Dispensing	reverseDispense	The authenticated requesting organisation shall be a healthcare organisation and its HPI-O shall match that of the healthcare organisation responsible for the matching E-`Dispensing::initiateDispense` operation)
E-Dispensing	cancelPrescription	The authenticated requesting organisation shall be a healthcare organisation.

B.2   Electronic Prescribing System

INTERFACE	OPERATION	POLICY DEFINITION
Prescription Requesting	deliverNotification	The authenticated requesting organisation shall be a healthcare

		organisation.
E-Prescribing	prescribe	The local *Electronic Prescribing System* user shall be a user who has been authorised to access and update the local health records of the *Prescription Subject* for whom the prescription is created.
E-Prescribing	cancelPrescription	The local *Electronic Prescribing System* user shall be a user who has been authorised to access and update the local health records of the *Prescription Subject* who's prescription is being cancelled.
E-Prescribing	retrievePrescription	The local *Electronic Prescribing System* user shall be a user who has been authorised to view the local health records of the *Prescription Subject* who's prescription is being retrieved.

B.3   Electronic Dispensing System

INTERFACE	OPERATION	POLICY DEFINITION
Prescription Receiving	deliverNotification	The *Prescriber Organisation* shall be the same organisation to which the target `Electronic Dispensing System` previously sent a `Prescription Required Notification` to which the service invocation being authorised is the reply
Contract Dispensing	deliverNotification	The authenticated requesting organisation shall be a healthcare organisation to which the *Dispenser Organisation* that operates the target `Electronic Dispensing System` is contracted
E-Dispensing	initiateDispense	The local *Electronic Prescribing System* user shall be a user who : • has been authorised to access and update the local dispensing records of the *Prescription Subject* who's prescription is being dispensed. • has knowledge of the `DAK` either because the *Prescription Subject* has provided it or because it has been received electronically via the `Contract Dispensing` or `Prescription Receiving` interfaces.
E-Dispensing	finaliseDispense	The local *Electronic Dispensing System* user shall be a user who has been authorised to access and modify the local dispensing records of the *Prescription Subject* for the prescription being processed.
E-Dispensing	terminateDispense	The local *Electronic Dispensing System*

		user shall be a user who has been authorised to access and modify the local dispensing records of the *Prescription Subject* for the prescription being processed.
E-Dispensing	reverseDispense	The local *Electronic Dispensing System* user shall be a user who has been authorised to access and modify the local dispensing records of the *Prescription Subject* for the prescription being processed.
E-Dispensing	cancelPrescription	The local *Electronic Prescribing System* user shall be a user who : • has been authorised to access and update the local dispensing records of the *Prescription Subject* who's prescription is being cancelled. • has knowledge of the DAK either because the *Prescription Subject* has provided it or because it has been received electronically via the Contract Dispensing or Prescription Receiving interfaces.

B.4   Facility Based Supply Manager

INTERFACE	OPERATION	POLICY DEFINITION
Facility Managed Supply	deliverNotification	The authenticated requesting organisation shall be a *Prescriber Organisation* that is a known party to an agreement with the operator of the Facility Based Supply Manager

B.5   Last Supply Notification Agent

INTERFACE	OPERATION	POLICY DEFINITION
Last Supply Notification Forwarding	deliverNotification	The authenticated requesting organisation shall be a *Dispenser Organisation* that is a known party to an agreement with the operator of the Last Supply Notification Agent.