



**eHealth Integration Sample Code v2.0**  
**Module - Core**

3 February 2015

Approved for external use

**National E-Health Transition Authority Ltd**

Level 25, 56 Pitt Street

Sydney, NSW 2000

Australia

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

**Acknowledgements****Council of Australian Governments**

The National E-Health Transition Authority is jointly funded by the Australian Government and all State and Territory Governments.

**HL7 International**

This document includes excerpts of HL7® International standards and other HL7 International material. HL7 International is the publisher and holder of copyright in the excerpts. The publication, reproduction and use of such excerpts is governed by the HL7 IP Policy (see <http://www.hl7.org/legal/ippolicy.cfm>) and the HL7 International License Agreement.

**Disclaimer**

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

**Document control**

This document is maintained in electronic form and is uncontrolled in printed form. It is the responsibility of the user to verify that this copy is the latest revision.

**Copyright © 2015 National E-Health Transition Authority Ltd**

This document contains information which is protected by copyright. All Rights Reserved. No part of this work may be reproduced or used in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems—without the permission of NEHTA. All copies of this document must include the copyright and other information contained on this page.

# Document information

## Key information

|                              |   |
|------------------------------|---|
| <b>Owner</b>                 | Head of Strategy, Architecture and Clinical Informatics     |
| <b>Contact for enquiries</b> | NEHTA Help Centre   |
|                              | t: 1300 901 001   |
|                              | e: <a href="mailto:help@nehta.gov.au">help@nehta.gov.au</a> |

## Product version history

| Product version | Date          | Release comments   |
|-----------------|---------------|--|
| 1.0             | February 2014 | Initial release (HIPS 4.1.0)   |
| 2.0             | February 2015 | See release note (NEHTA-2040:2015) for details of changes and bug fixes. |

# Table of contents

|           |  |          |
|-----------|--|----------|
| <b>1.</b> | <b>Introduction .....</b>                                  | <b>5</b> |
| 1.1       | Purpose.....   | 5        |
| 1.2       | Summary .....  | 5        |
| <b>2.</b> | <b>Architectural Detail.....</b>                           | <b>6</b> |
| 2.1       | Component Model.....                                       | 6        |
| 2.1.1     | PCEHR Participation and Authorisation Model.....           | 6        |
| 2.2       | Business Logic.....  | 7        |
| 2.2.1     | Database Loader Service .....                              | 7        |
| 2.2.2     | Patient Identifier.....                                    | 7        |
| 2.2.3     | Get Validated IHI .....                                    | 9        |
| 2.2.4     | Check Whether PCEHR is Advertised.....                     | 11       |
| 2.2.5     | Consent and Participation .....                            | 12       |
| 2.2.6     | Upload or Supersede Document to PCEHR.....                 | 14       |
| 2.2.7     | Upload or Supersede Discharge Summary Level 1A.....        | 20       |
| 2.2.8     | Remove Document from PCEHR.....                            | 21       |
| 2.2.9     | Get Operation Status.....                                  | 23       |
| 2.2.10    | Get Queued Operation List .....                            | 23       |
| 2.2.11    | Search for Provider Individual Details - Synchronous.....  | 23       |
| 2.2.12    | Search for Provider Individual Details – Asynchronous..... | 24       |
| 2.2.13    | Gain PCEHR Access .....                                    | 26       |
| 2.2.14    | Document List – Registry Stored Query (ITI-18) .....       | 28       |
| 2.2.15    | Download Document - Retrieve Document Set (ITI-43) .....   | 29       |
| 2.2.16    | Change History View .....                                  | 29       |
| 2.2.17    | Get View.....  | 29       |
| 2.2.18    | List Patients in Hospital.....                             | 30       |
| 2.2.19    | List Patients Episodes in Hospital .....                   | 30       |
| 2.2.20    | Assisted Registration .....                                | 31       |
| 2.2.21    | Contracted Service Provider (CSP) Usage.....               | 33       |
| 2.2.22    | Multi-Tenant for IHI .....                                 | 33       |
| 2.2.23    | Reference Services.....                                    | 34       |
| 2.2.24    | Common Schemas .....                                       | 35       |
| 2.3       | Database Resource Access Layer .....                       | 36       |
| 2.3.1     | HealthProviderOrganisationPatient Table.....               | 37       |
| 2.3.2     | Queuing Data Model.....                                    | 38       |
| 2.3.3     | Clinical Document Data Model .....                         | 40       |

# 1. Introduction

## 1.1 Purpose

eHISC Core provides services for the following functions:

- Database Loader – for processing HL7 messages from the PAS
- Get Validated IHI – for interacting with the Healthcare Identifier (HI) service
- Check whether a PCEHR is Advertised
- Consent and Participation Services
- Upload or Supersede Document to the PCEHR
- Remove Document from the PCEHR
- Search for a HPI-I
- Provide assisted registration to the PCEHR
- View documents on the PCEHR

Throughout the document the term ***“Participating Organisation”*** is used which related to one of the following system implementers of eHISC:

- Health Provider Organisation (HPO),
- Contracted Service Provider (CSP) or
- Jurisdiction

## 1.2 Summary

The eHISC suite consists of the following products:

- **eHISC-Core:** A middleware and communications solution to enable a CIS (Clinical Information System) and a PAS (Patient Administration System) to interact with the National eHealth Record System. The solution can interface with an Enterprise Service Bus (ESB) to receive HL7 records from the PAS systems for patient and episode information and IHI lookups, and CDA documents from the clinical systems for upload to PCEHR. It can also be used as a broker to the PCEHR without the need of an interface to an ESB for upload and retrieval of documents from the PCEHR.
- **eHISC-UI:** An extension of the core services provided by the eHISC-Core product, providing a web-based user interface for fulfilling common interaction requirements with the PCEHR, including features such as PCEHR Viewing (including Prescriptions), Level 1A Document Uploads (eHISC Core can support all levels and types of document uploads), PCEHR Document Removal, PCEHR Consent Withdrawal, Hidden PCEHR Disclosure, Assisted Registration, HPI-I Search and Data Integrity.

## 2. Architectural Detail

### 2.1 Component Model

#### 2.1.1 PCEHR Participation and Authorisation Model

Each hospital facility within Australia, that intends to upload discharge summaries to the PCEHR system, must be associated with a participating Healthcare Provider Organisation (HPO), identified by an HPI-O. Multiple hospitals may be associated with same HPO. For example:

- Some or all hospitals within a participating organisation may be associated to the seed HPO for the participating organisation.
- Some or all hospitals within a Local Health Network (LHN) may be associated to the network HPO for that LHN.
- Some or all hospitals may themselves be a network HPO.

For any patient record to which eHISC has assigned a valid IHI, the creation of a new episode of care in the Patient Administration System (PAS) at the hospital will trigger eHISC to send a message to the PCEHR system, asking if the consumer has a PCEHR. The message will include:

1. The consumer's IHI;
2. The hospital's associated healthcare provider organisation's HPI-O; and
3. The local system identifier of the authorised employee for the hospital or the interactive user.

Once the PCEHR system receives this message, the PCEHR system will proceed to verify the right of the HPO to access the PCEHR system, and determine whether the HPO is authorised to know whether the consumer has a PCEHR. The system then advises the HPO whether the consumer has a PCEHR, with the answer being "no" if the HPO is not authorised to know.

eHISC will store this advice separately for each participating HPO. In the case where the consumer has elected to hide the existence of his/her PCEHR, it is possible that one participating HPO has gained access and therefore "knows" the consumer has a PCEHR, while another HPO has not gained access and therefore does not "know" that the same consumer has a PCEHR. The Participating Organisation should respect this separation of information for privacy reasons.

When the discharge summary for the episode of care is distributed, if the PCEHR system's advice to the discharging hospital's HPO indicated that the consumer has a PCEHR, the default setting is that the discharge summary record will be uploaded, unless the consumer requests that it is not uploaded (via withdrawal of consent), or the clinical information system user requests that it is not uploaded.

Otherwise, if the advice indicated the consumer does not have a PCEHR, the default setting is that the discharge summary record will not be uploaded, unless the consumer disclosed the existence of the PCEHR

## 2.2 Business Logic

### 2.2.1 Database Loader Service

#### 2.2.1.1 Description

This service is designed to accept messages from patient administration systems via a message broker. This message broker should transform the messages to comply with the format expected by eHISC, which is based on the international HL7 standard, version 2.3.1. Refer to the document “eHISC HL7 Interface Specification” for details of this format.

This service stores the patient and episode information into the PCEHR Data Store, and triggers the automatic IHI lookup and check for advertised PCEHR.

A failure to store the message, patient or episode will result in a negative acknowledgement being returned. A failure to obtain an IHI or check the PCEHR status will not result in a negative acknowledgement. The IHI is not returned in the acknowledgement, but stored into the PCEHR Data Store.

An example of the minimal input for registering a patient is as follows:

```
MSH|^~\&|App|Facility|||DateTime|03V1|ADT^A28|MsgID|P|2.3.1|||AL|NE|AU|ASCII|EN  
EVN|A28|20120716011454|||Operator  
PID|||MRN^^^Facility^MR||Surname^First Name^Middle Names^^^L^A||DOB|Sex|||Address Line  
1^Address Line 2^Suburb^State^Postcode^^H
```

### 2.2.2 Patient Identifier

Each eHISC service that is designed to act upon a single patient record will contain a parameter “patientIdentifier”. The Patient Identifier object will be used to identify which hospital and patient to operate upon. The Patient Identifier can be either:

- Medical Record Number (MRN), scoped within a specified hospital,
- State Patient Identifier, scoped within the Participating Organisation,
- Registered Enterprise Patient, scoped within the Participating Organisation,
- Validated Individual Healthcare Identifier (IHI), scoped nationally,
- Patient Master Identifier, internal to the PCEHR Data Store, or
- Demographic, used within Assisted Registration calls when the person has not been admitted to Hospital.

#### 2.2.2.1 Mrn

This class represents a Medical Record Number (MRN) that identifies a patient at a hospital. It is usually allocated by the hospital PAS or PMI. This value is stored in the Mrn column of the HospitalPatient table in the PCEHR Data Store.

### 2.2.2.2 StatePatientId

This class represents a number or code that identifies a patient across an entire Participating Organisation. It can be allocated by a type of EMPI (Enterprise Patient Master Index), but may also be allocated by a PAS. This value is stored in the StatePatientId column of the PatientMaster table in the PCEHR Data Store.

### 2.2.2.3 RegisteredEnterprisePatient

This class is a patient identifier that identifies a patient who is registered at the enterprise level using a "StatePatientId" and can extend the registration to a new facility using the supplied "Mrn".

The biggest difference between a StatePatientId and a RegisteredEnterprisePatient is that the RegisteredEnterprisePatient allows the operation to access patient records in a different facility to the facility where the patient was originally registered.

The following business logic is used when the RegisteredEnterprisePatient class is used to identify the patient:

- Get the patient information using the StatePatientId
  - If the patient cannot be found then set the response to null and return
- Get the hospital patient information using the MRN
  - If the hospital patient record is found and the PatientMasterId matches with the patient information from the StatePatientId then return the patient information
  - If the hospital patient record is found but the PatientMasterId does not match then it means the hospital patient is on the wrong patient master record
    - Log an error message with the relevant information and return a "IncorrectStatePatientId" result
- If the hospital patient record is not found then use the PatientMasterId to retrieve the MRN of the patient
  - If an MRN is returned then the patient is currently registered using a different MRN
    - Log an error message with the relevant information and return a "IncorrectMrn" result
  - If an "InvalidPatient" result was returned then we can create a hospital patient record
    - Use the PatientMasterId obtained from the StatePatientId look up, the hospital passed in and MRN passed in via the patient identifier to create the hospital patient record
    - If this fails log an error message and return a "DatabaseError" result
    - Check the local IHI information of the patient and if not "OK" or "InvalidIhi" then return the result
    - Validate the IHI which if successful will perform a PCEHR check against the current facility and store the result.

### 2.2.2.4 ValidatedIhi

This class represents the set of information that makes up a validated IHI. The information includes everything that is required to determine whether the IHI remains valid, and to revalidate the IHI. These values are stored in the PatientMaster and PatientMasterIhi tables in the PCEHR Data Store.



When a validated IHI is used to identify a patient in the eHISC service call, then eHISC will create or update the patient record as necessary, taking the incoming information as authoritative. This makes it possible to operate eHISC in a distributed system where eHISC does not receive a feed of PAS messages and does not make its own connections to the HI Service.

Note that this class does not contain properties for Medicare Card Number or DVA File Number, because this information is not required for IHI validation, but only when the IHI is first retrieved.

#### **2.2.2.5 PatientMasterId**

This class represents the internal primary key of the PatientMaster table in the PCEHR Data Store. This option is made available for applications that share use of the PCEHR Data Store with eHISC and hence have direct knowledge of the database keys.

#### **2.2.2.6 Demographic**

This class defines a patient's demographics for use within the registration process. This type of patient identification can only be used with Assisted Registration and cannot be used with other eHISC services that access the PCEHR system.

### **2.2.3 Get Validated IHI**

This function retrieves the IHI information that must be inserted in a CDA clinical document to identify the patient and to allow the receiver of the document to re-validate the IHI.

#### **2.2.3.1 Business Rules / Functional Business Logic**

eHISC will first attempt to locate an existing patient record using the given patient identifier. With that patient record:

1. If the stored date of birth does not match that which is specified in the service call, then an *InvalidDateOfBirth* error is returned and no further action is taken. If the *RegisteredDateOfBirthEnabled* flag is true then the date of birth specified in the service call will also be checked against the registered date of birth.
2. If no IHI has been obtained for the patient record, then eHISC will attempt to obtain the IHI using the current demographic information and assign the IHI to the local patient record, possibly creating an exception alert in the process. An example of an exception alert is when the IHI status changes from Active to another status, or the IHI number is already assigned to another patient record from the same hospital.
3. If an IHI is obtained or had already been obtained for the specified patient record, and there is an outstanding exception alert on the IHI, such as a suspected duplicate or replica, then eHISC will not return the IHI to the caller.
4. If the IHI was obtained or last validated outside the time period that has been configured for this purpose, then eHISC will attempt to validate the IHI information with the HI Service, and will only return the IHI to the caller if the validation was successful. However if the HI Service is unavailable, the IHI will be returned with a warning that validation must still occur before the IHI can be trusted. This will occur when the document with the IHI embedded is passed back to eHISC for upload.
5. If the IHI was obtained or last validated within the configured valid time period, then the IHI will be returned immediately without triggering another validation.

### 2.2.3.2 Usage Notes

This method is intended for use after a patient is registered in the PAS, to extract the IHI that was obtained from the HI Service, for use in a clinical document that will be distributed to an external health provider, to a shared repository or the PCEHR system.

One model for usage is where the clinical system is enhanced to handle IHI and CDA directly:

- A clinical system user finalises a discharge summary for a patient who has a PCEHR
- The system makes a call to an ESB to find or validate the IHI for the patient
- The ESB calls this method of eHISC and returns the validated IHI to the clinical system
- The clinical system produces a CDA discharge summary document with the validated IHI
- The system makes a call to the ESB to upload the discharge summary to the PCEHR
- The ESB sends the CDA document to eHISC for upload to the PCEHR system

However, there are other implementations where the clinical system is not aware of IHIs, and thus custom-developed middleware would be needed to handle the conversion of HL7 discharge summaries to CDA format:

- A clinical system user finalises a discharge summary for a patient who has a PCEHR
- The clinical system delivers the discharge summary data to the ESB for upload to PCEHR, in the form of an HL7 message
- The ESB calls this method of eHISC to obtain the IHI for the patient
- The ESB embeds the IHI into the HL7 message and delivers it to a CDA conversion middleware
- The middleware converts the discharge summary from HL7 to CDA format and returns it to ESB
- The ESB sends the CDA document to eHISC for upload to the PCEHR

## 2.2.4 Check Whether PCEHR is Advertised

### 2.2.4.1 Description

A user-facing system may call this service method to obtain information necessary to inform the user about whether a patient has registered for a PCEHR and wants that PCEHR to be visible to the provider organisations participating in his/her healthcare.

This method wraps the PCEHR B2B Gateway service that NEHTA calls “Does PCEHR Exist”. That name is misleading because the result is not strictly whether a PCEHR exists for the consumer, but is affected by whether the consumer has chosen to advertise the existence of his/her PCEHR. The result is also affected by whether the HPI-O is on the provider access list of the consumer’s PCEHR, and if so what read access permission the consumer has given the HPI-O.

The result is also used internally by eHISC to drive the related function to determine whether a patient wants the discharge summary to be uploaded to his/her PCEHR. The latter service is described in the Patient Participation section. The main difference is that Patient Participation is also affected by the disclosure of the existence of a PCEHR. If the patient has disclosed the existence to the HPO then they are considered to be participating regardless of whether the PCEHR is hidden.

The response contains a property `AccessCodeRequired`, which provides some information as to whether an access code is required in order to access the patient’s PCEHR to list or view documents:

- Null – thus meaning that the patient has not registered for PCEHR, or has chosen to hide the existence of his/her PCEHR. The patient may still give advice of PCEHR existence and may or may not provide an access code.
- With Code – thus meaning that the provider may not gain access to the PCEHR unless their patient provides an access code.
- Without Code – thus meaning that access is open and no Record Code is required. The patient may still advise of a Document Code to grant restricted access.
- Access Granted – thus meaning that access has been granted and no code is required unless the patient advises of a change to access level and provides an access code.

### 2.2.4.2 Usage Note

For implementations where the HI Service is called by eHISC and/or HL7 patient and episode messages are sent to eHISC, this method may not be required to be called separately, because:

- eHISC automatically calls `DoesPCEHRExist` immediately after obtaining an IHI from the HI Service
- eHISC automatically calls `DoesPCEHRExist` after creating a new episode for an existing patient

Thus, this method is a mechanism to manually trigger calls to `DoesPCEHRExist`. It is primarily of use where:

- The Participating Organisation does not provide a feed of HL7 messages from the PAS into eHISC, therefore this is the main mechanism to check whether a patient has a PCEHR or not.
- The clinical system requires the latest information about the access status for the PCEHR, such as immediately before or after a call to `Gain Access`.

## 2.2.5 Consent and Participation

### 2.2.5.1 Consent to Upload

Under the PCEHR consent model, consent to upload a clinical document to the PCEHR is assumed, because in the process of registering for a PCEHR a blanket consent is extended to all providers.

Although this is expected to be rare, it is a NEHTA requirement that the system support the ability for the patient to withdraw this assumed consent at any time. eHISC will store a flag against each patient episode that indicates whether the patient has withdrawn consent to upload documents for that episode.

eHISC will provide a web service method “RecordConsent” that can be used in two situations:

1. Withdrawal of Consent: When the patient has advised the provider that he/she does not want the discharge summary to be uploaded.
2. Rescind Withdrawal: when such a withdrawal of consent was recorded in error.

Whenever a clinical system requests to upload or supersede a clinical document, eHISC will first check the flag for the episode to which the clinical document relates, to find whether consent has been withdrawn. If consent has been withdrawn then eHISC will refuse to upload the document.

A consumer cannot withdraw consent after a document has been uploaded to the PCEHR. So long as the first version was uploaded while the consumer was consenting, then any later versions of that document can be uploaded. Therefore the correct process is to avoid recording a withdrawal of consent if there are documents already uploaded.

However, if the consumer withdrew consent but the information was not entered into the system in time and the document was uploaded in error, then the correct process is to first remove the document from the PCEHR and then enter the withdrawal of consent to prevent it being uploaded again.

### 2.2.5.2 Participation Status

In most cases when a patient has registered for a PCEHR, eHISC will find out that the PCEHR exists when it calls the PCEHR B2B Gateway method “doesPCEHRExist”. At a minimum, this call is triggered once at the creation of an episode of care.

However for those rare cases when a patient has chosen to hide the existence of his/her PCEHR, but the patient wishes the provider organisation to upload the document despite this, eHISC will provide a web service method “RecordDisclosure”. This service can be used in two situations:

1. Disclose PCEHR: when the patient has advised the provider organisation of the existence of his/her PCEHR (and by inference, consented to have his/her discharge summary uploaded), even though the PCEHR is not advertised (or may later become not advertised).
2. Rescind Disclosure: when such a disclosure was recorded in error.

eHISC will record the disclosure for each patient at each health provider organisation (HPO). If a Participating Organisation using eHISC consists of multiple HPOs, then the patient’s disclosure to one HPO will not automatically apply to that patient at any other HPO.

When a clinical system requests the participation status for a certain patient, or requests a list of patients who have changed participation status since a certain date, then eHISC will indicate that the patient is participating in PCEHR if the patient has disclosed the existence of a PCEHR to the health provider organisation, otherwise it will indicate whether the existence of the PCEHR is currently advertised or not.

### 2.2.5.3 Refresh Participation Status

eHISC was designed to store the PCEHR advertised and disclosed statuses, for a patient, separately for each healthcare provider organisation, to support the consumer's right to choose which organisations have access to their eHealth Record. Some Participating Organisation require that disclosure is stored once for each patient across the entire Participating Organisation, even when they have multiple facilities with separate HPI-O's.

The RefreshPatientParticipationStatus service operation combines the PCEHR advertised and disclosed lookup for different facilities and refreshes the advertised status from the PCEHR system. The following logic is performed when the method is called:

- Retrieves information about the current facility and the disclosure facility. If no disclosure facility is provided then the current facility will be used as the disclosure facility.
  - If either of the facilities cannot be found then throw an "ItemNotFoundException" with the "ItemType" set to "Hospital"
- Retrieve the patient information from the current facility including the MRN and State Patient ID.
  - If the patient is not found then throw an "ItemNotFoundException" with the "ItemType" set to "Patient".
- Look up the current disclosure status of the patient at the disclosure facility.
- Validate the patients IHI
  - If the response from the HI Service is "HiServiceError" then throw a "HiServiceException"
  - If the response is "PcehrServiceError" then throw a "PcehrServiceException"
  - If the response is **not** OK then throw a "HipsResponseException"
  - If the response is OK then check the local IHI information. If this is not valid then throw a "InvalidIhiException"
- If the "ForceRefresh" parameter is set to "Never" then return the patient participation status without refreshing the PCEHR status.
- If the "ForceRefresh" parameter is set to "WhenNotAdvertised", and either the advertised or disclosed status of the patient is true, then return the patient participation status without refreshing the PCEHR status.
- Otherwise call the "DoesPcehrExist" method to refresh the PCEHR advertised status.
  - If the response is **not** OK then a "PcehrServiceException" will be thrown.

## 2.2.6 Upload or Supersede Document to PCEHR

This service implements a “fire and forget” pattern that adds a document instance to the queue for uploading to the appropriate repository for the document type. The service will return as soon as the item is added to the queue.

### 2.2.6.1 Patient Matching

The patientIdentifier parameter is used to look up a patient record. If the patient identifier is of type Mrn, StatePatientId or PatientMasterId then the patient must already exist in the PCEHR Data Store, otherwise an “InvalidPatient” error will be returned. However, if the patient identifier is of type ValidatedIhi, then the patient need not exist in the PCEHR Data Store; if they do not exist, a minimal stub record will automatically be created.

### 2.2.6.2 IHI Validation

If the IHI has an unresolved data-quality alert, then an “UnresolvedIhiAlert” error will be returned and the document will not be placed on the queue for upload.

If the IHI was last validated more than the configured period of time in the past, the IHI will be revalidated with the HI Service. If validation returns no records found with the IHI and the stored demographic information, an “InvalidIhi” error will be returned and the document will not be placed on the queue for upload. However, if the HI Service is unavailable then the document will be placed on the queue with a stale IHI, which will be revalidated when the item is taken off the queue to be processed.

Each time an upload operation is taken off the queue to be processed, eHISC will check that the IHI was validated within the configured period. If there is an outage that stops the upload for longer than the configured period, then eHISC will automatically revalidate the IHI before attempting the upload again.

This has a critical impact on implementations where eHISC is not connected to the HI Service. Without a connection to the HI Service, eHISC cannot upload a document it remains on the queue beyond the configured period. Therefore, the configured period should be long enough that eHISC will not need to revalidate the IHI. Otherwise, documents will fail to upload and need to be resent with a newly validated IHI.

### 2.2.6.3 Document Validation

eHISC will not run a full CDA validation on documents before uploading them to the PCEHR, because that would duplicate the work that the PCEHR system does itself. If documents fail validation that implies there is a deficiency in the software that generated the CDA document. The document should be resent from the source system after the deficiency has been corrected.

eHISC will extract and validate the following items from the document:

- The Document Type is extracted from the <code> element. The code must match to a code that is configured in the DocumentType table. Each document type is associated with a repository. eHISC will connect to the associated repository (PCEHR) for uploading and removing documents.
- The Document ID is extracted from the <id> element. The root must be an OID or a UUID. The extension is optional. Note that the PCEHR system requires that the root is unique, so when using an extension to show a user-friendly numeric ID, it is necessary to repeat the extension inside the root.
- The Set ID is extracted from the <setId> element. The root must be an OID or a UUID. The extension is optional. Although the Set ID is optional in the CDA implementation guide, it is mandatory for eHISC.

- The IHI is extracted from the <id> element whose assigningAuthorityName is "IHI". The IHI must match the IHI assigned to the patient in the eHISC database, otherwise an "InvalidIhi" error is returned and the document is not uploaded.

If the CDA document is not valid XML, or eHISC is unable to extract any of these items, an "InvalidDocument" error is returned.

#### 2.2.6.4 Document Format Codes (Template Package IDs)

The document format code, also known as a template package ID, is used to specify which validation rules the PCEHR system will apply to the document when it is uploaded. There is a different format code for each conformance level of each document type. Also, as the PCEHR system is upgraded over time, there are new document format codes that can be used. These new format codes allow for changes to be made in the document validation rules over time, without affecting systems that upload documents developed under the older rules.

Each of the document format codes that eHISC will use for uploading documents must be configured in the DocumentFormat table. If the specified format code is not found in the DocumentFormat table, an "InvalidDocument" error is returned.

For Participating Organisations that upload only one conformance level of one document type, there is no need to include the parameter "documentFormatCode". When this parameter is omitted or null, eHISC will use the format code that is configured as "DefaultDocumentFormatCode" in the web.config file. Otherwise, specify the format code in the parameter "documentFormatCode" of each upload request.

The HPI-I Relaxed templates are restricted to Participating Organisations who have been granted permission from the PCEHR system operator, for the document author's identifier to be a local system identifier instead of an HPI-I. This permission may be time-limited, after which Participating Organisations will need to transition to the HPI-I Enforced templates.

As of eHISC 2.0, the format codes for supported document types are:

| Document Type     | Version | HPI-I   | Conformance Level | Document Format Code          |
|-------------------|---------|---------|-------------------|-------------------------------|
| Discharge Summary | R2      | Relaxed | 1A                | 1.2.36.1.2001.1006.1.20000.12 |
|                   |         |         | 1B                | 1.2.36.1.2001.1006.1.20000.9  |
|                   |         |         | 2                 | 1.2.36.1.2001.1006.1.20000.10 |
|                   |         |         | 3A                | 1.2.36.1.2001.1006.1.20000.11 |
|                   | R3      | Relaxed | 1A                | 1.2.36.1.2001.1006.1.20000.13 |
|                   |         |         | 1B                | 1.2.36.1.2001.1006.1.20000.14 |
|                   |         |         | 2                 | 1.2.36.1.2001.1006.1.20000.15 |
|                   |         |         | 3A                | 1.2.36.1.2001.1006.1.20000.16 |
|                   |         |         | 3B                | 1.2.36.1.2001.1006.1.20000.17 |
|                   | R4      | Relaxed | 1A                | 1.2.36.1.2001.1006.1.20000.18 |
|                   |         |         | 1B                | 1.2.36.1.2001.1006.1.20000.19 |
|                   |         |         | 2                 | 1.2.36.1.2001.1006.1.20000.20 |
|                   |         |         | 3A                | 1.2.36.1.2001.1006.1.20000.21 |

| Document Type             | Version | HPI-I    | Conformance Level | Document Format Code          |
|---------------------------|---------|----------|-------------------|-------------------------------|
|                           |         | Enforced | 3B                | 1.2.36.1.2001.1006.1.20000.22 |
|                           |         |          | 1A                | 1.2.36.1.2001.1006.1.20000.23 |
|                           |         |          | 1B                | 1.2.36.1.2001.1006.1.20000.24 |
|                           |         |          | 2                 | 1.2.36.1.2001.1006.1.20000.25 |
|                           |         |          | 3A                | 1.2.36.1.2001.1006.1.20000.26 |
|                           |         |          | 3B                | 1.2.36.1.2001.1006.1.20000.27 |
| PCEHR Prescription Record | R4      | Relaxed  | 3A                | 1.2.36.1.2001.1006.1.170.2    |
|                           |         | Enforced | 3A                | 1.2.36.1.2001.1006.1.170.3    |
| PCEHR Dispense Record     | R4      | Relaxed  | 3A                | 1.2.36.1.2001.1006.1.171.2    |
|                           |         | Enforced | 3A                | 1.2.36.1.2001.1006.1.171.3    |
| Event Summary             | R4      | Relaxed  | 3A                | 1.2.36.1.2001.1006.1.16473.9  |
|                           |         |          | 3B                | 1.2.36.1.2001.1006.1.16473.8  |
|                           |         | Enforced | 3A                | 1.2.36.1.2001.1006.1.16473.10 |
|                           |         |          | 3B                | 1.2.36.1.2001.1006.1.16473.11 |
| Shared Health Summary     | R4      | Relaxed  | 3A                | 1.2.36.1.2001.1006.1.16575.4  |
|                           |         |          | 3B                | 1.2.36.1.2001.1006.1.16575.5  |
|                           |         | Enforced | 3A                | 1.2.36.1.2001.1006.1.16575.6  |
|                           |         |          | 3B                | 1.2.36.1.2001.1006.1.16575.7  |
| Specialist Letter         | R4      | Relaxed  | 1A                | 1.2.36.1.2001.1006.1.16615.13 |
|                           |         |          | 1B                | 1.2.36.1.2001.1006.1.16615.14 |
|                           |         |          | 2                 | 1.2.36.1.2001.1006.1.16615.15 |
|                           |         |          | 3A                | 1.2.36.1.2001.1006.1.16615.16 |
|                           |         |          | 3B                | 1.2.36.1.2001.1006.1.16615.17 |
|                           |         | Enforced | 1A                | 1.2.36.1.2001.1006.1.16615.18 |
|                           |         |          | 1B                | 1.2.36.1.2001.1006.1.16615.19 |
|                           |         |          | 2                 | 1.2.36.1.2001.1006.1.16615.20 |
|                           |         |          | 3A                | 1.2.36.1.2001.1006.1.16615.21 |
|                           |         |          | 3B                | 1.2.36.1.2001.1006.1.16615.22 |

### 2.2.6.5 Episode Matching

Every clinical document uploaded by eHISC must be attached to an episode, which can be an inpatient admission, an emergency visit, an outpatient appointment or another type configured in EpisodeType.

There are two ways to create and manage episodes. One is by sending HL7 messages (such as A01 or A08) to the eHISC PAS Loader, while the other is to use the ValidatedIhi type of patient identifier and allow eHISC to create and manage episode stub records for uploaded documents.



eHISC will look for a previous episode stub for the upload of a document with the same Set ID as this document instance. If one is found then this episode stub is used and updated to the given admission date/time. Otherwise, the admission date and time are used to match an Episode record.

If there is no episode matched, and the patient identifier is of type ValidatedIhi, then an episode stub record will be created. The stub record will have the document Set ID as its source system episode ID, so that future supersedes and removals of this document can proceed even if the admission date/time has changed. For other patient identifier types, if there is no episode matched, or there is more than one episode with the same admission date and time (within one minute), then an “InvalidEpisode” error will be returned.

#### **2.2.6.6 Consent Checking**

The ability to handle a patient’s withdrawal of consent is a core requirement for Clinical Information Systems connecting to the PCEHR System. Participating Organisations using eHISC can make use of the eHISC consent management services or ignore this functionality and handle consent in other systems. In line with the PCEHR consent model, when an episode is created, the default value for UploadConsent is true, that the patient has given consent to upload documents. This can be changed using the RecordConsent service.

Once an episode is matched, eHISC will check the UploadConsent flag. If the flag is set to false, this means the patient withdrew consent to upload this document. eHISC will return a “ConsentWithdrawn” error and prevent the upload of the document.

#### **2.2.6.7 Age Checking**

Some Participating Organisations may have a policy that they will disable uploading documents for children under a certain age. This age can be configured for each hospital in the Hospital table. This feature can also be disabled by setting the value 0.

For consistency, eHISC will calculate the age of the patient at the time of admission, rather than the time of discharge or the time of the upload.

If the value of “UploadDocumentMinimumAge” configured for the hospital is non-zero, and the patient was under the configured age at the time of admission, then eHISC will return a “PatientUnderAge” error and the document will not be added to the queue.

### 2.2.6.8 Attachment Validation

eHISC will ensure that each attachment is under the maximum size for attachments in the PCEHR system (10 megabytes) and that each attachment belongs to one of the supported file types. If either of these checks fails, an “InvalidDocument” error will be returned and the document will not be added to the queue.

The types of attachment files supported by the PCEHR are:

| MIME type       | File Extensions | Description                      |
|-----------------|-----------------|----------------------------------|
| image/gif       | .gif            | Graphics Interchange Format      |
| image/jpeg      | .jpg, .jpeg     | Joint Photographic Experts Group |
| image/tiff      | .tif, .tiff     | Tagged Image File Format         |
| image/png       | .png            | Portable Network Graphics        |
| application/pdf | .pdf            | Portable Document Format         |

### 2.2.6.9 CDA Packaging

An electronic signature is created for the CDA document. The signature asserts the author’s name and either the author’s local system identifier or HPI-I that is extracted from the CDA document. The signature is created using the NASH certificate for the HPI-O that is in use by the hospital. This certificate has the serial number indicated by the PcehrCertSerial value in the HealthProviderOrganisation record in the database.

An “InvalidDocument” error is returned if these elements cannot be extracted from the CDA document, or the logo exceeds the maximum dimensions of 400 x 100 pixels.

An XDM package (ZIP file) is created, consisting of:

- The XML provided in the cdaDocument parameter as CDA\_ROOT.XML. The XML is unchanged except that integrity check attributes are added to the logo reference if eHISC is including an organisational logo from the hospital configuration.
- The created electronic signature as CDA\_SIGN.XML
- If a logo file is configured for the hospital, and not provided as an attachment, the logo file from the hospital configuration is included with the file name specified in the document.
- Any additional attachments provided in the attachments parameter.

After the consent check, signing and packaging, the upload request is added to the upload queue and the service returns to the caller. After the service has returned, the status of the document upload may be determined using the GetOperationStatus method.

#### **2.2.6.10 Cancellation of a Queued Upload Operation**

It may be necessary to manually cancel a queued upload operation, if the PCEHR system returns an error message that is documented to have the meaning of system temporarily unavailable, but is actually due to an error in the structure of the uploaded document or metadata. This can be done by setting the queue status in the PcehrMessageQueue record to 3 (failed).

Each time an upload operation is taken off the queue for processing, eHISC checks the current value of the QueueStatusId in the PcehrMessageQueue record. If the queue status is no longer set to 1 (pending) then eHISC will stop processing this upload operation, and the queue processor will move to the next message in the queue.

#### **2.2.6.11 Ordering of Queued Operations**

If the PCEHR system is temporarily unavailable, the queued upload operation will be placed on a retry queue. The retry queue is standard functionality of the MSMQ binding in the web.config file. When a message has reached the receiveRetryLimit (set to 3) it is then paused to wait for a period of time set by the retryCycleDelay (set to 5 minutes). The maxRetryCycles is set to 6000 so that the PCEHR can be completely offline for a minimum of 20 days (excluding connection timeout delays for each retry) before the MSMQ is placed into a halted state.

While one or more operations are on the retry queue, other upload requests can arrive. eHISC will attempt to process these operations immediately as the PCEHR system may now have sufficient capacity to process these requests.

However it is important to ensure that requests that relate to the same document set are processed in the correct order. Therefore, each time an upload operation is taken off the queue for processing, eHISC checks whether there are any earlier pending operations for the same document set. If there are any, then eHISC will roll back the queue transaction, so that the upload operation will be retried after the earlier request is completed.

#### **2.2.6.12 Determination of Request Type**

When an upload operation is taken off the queue for processing, the Set ID and Document ID are used to locate existing records in the database that related to the document that is being uploaded.

If the Document ID exists in the ClinicalDocumentVersion table, then the queued operation will be marked as failed because the document has already been uploaded. eHISC will not attempt to upload the document again.

If the Document ID does not exist, but the Set ID exists in the ClinicalDocument table, then a request is generated with the type of replacement (supersede) of the most recently uploaded Document ID in the ClinicalDocumentVersion table that is recorded against the matched record in the ClinicalDocument table. This is the mechanism by which eHISC ensures that it only replaces documents that it uploaded itself.

If neither the Document ID nor the Set ID exists in the database, then the request type is an upload of a new document.

#### **2.2.6.13 Auditing of Request and Response**

eHISC will upload or supersede the clinical document to the PCEHR National Repository, and write an audit record into the PcehrAudit table. This audit record contains the full SOAP request and response, and is vital for troubleshooting when documents have failed to upload.

### 2.2.6.14 Response Classification

Using the definitions of the PCEHR error messages in the Technical Service Specification, each response from the PCEHR is classified as one of:

- Success
- Warning (e.g. operation successful but persisted as unstructured document)
- Duplicate Document ID
- System Temporarily Unavailable
- Unrecoverable Error

In the case of a success, warning or duplicate document ID message, the queued operation is deleted and records of the document are stored into the `ClinicalDocument` and `ClinicalDocumentVersion` tables.

In the PCEHR model, superseding a removed document changes its status back to active; all versions, including the one that was removed, are once again visible to both consumers and providers. Accordingly, in the eHISC `ClinicalDocument` table the document status will be reset to "Uploaded" (even if it was previously "Removed"). The removal date and removal reason will be reset to null and -1 (unknown) respectively.

The reason why duplicates are treated the same as success at this point is because the document was missing from the eHISC database even though it was already on the PCEHR system. This can happen in the case when the eHISC database is restored from backups following the disaster recovery process, and documents that were uploaded after the recovery point are resent to ensure they are recorded correctly in the restored eHISC database.

In the case of a message classified as System Temporarily Unavailable, eHISC will roll back the queue transaction, so that the upload operation will be retried. The retry behaviour is controlled by the MSMQ settings in the `web.config` file. See the eHISC Installation Guide for details.

In the case of an unrecoverable error, such as an invalid document structure, the queued operation is marked as failed, and the queue processor will move to the next message in the queue.

### 2.2.7 Upload or Supersede Discharge Summary Level 1A

This service is used specifically to upload a Level 1A Compliant Discharge Summary to the PCEHR. Level 1A documents can also be uploaded using the main Upload or Supersede web service, however this is specific to Level 1A documents as it does not require a CDA document format to be pre-created and passed to the web service. It simply requires some specific metadata (for the CDA header) and the PDF attachment, which the service is then able to create the CDA document and then upload to the PCEHR.

It is important to note that the Facility/Hospital and associated signing certificates must be fully configured within eHISC for this process to occur without errors.

It is important to note that when the returned object from the request (`UploadDischargeSummaryLevel1AResponse`) has a success response this indicates that the message has successfully passed validation and has been placed in the eHISC message queue (MSMQ) waiting to be uploaded to the PCEHR, it does not mean that the document has yet been successfully uploaded to the PCEHR. The eHISC message queue status can be queried using the `GetQueuedOperationList` and `GetIndividualOperationStatus` web services.

#### 2.2.7.1 Business Rules

1. The *Hospital*, *HospitalPatient*, and *PatientMaster* records for the provided *PatientIdentifier* are resolved using a mechanism appropriate to the provided *PatientIdentifier* type.

2. A patient requires a *PatientMasterIhi* with a current and verified IHI.  
  
If an IHI is not present, or is found to be invalid for any reason, the upload is not performed and an error is returned.  
  
If an IHI is present but out of date an attempt to revalidate it via the HI Service is performed. Upload will continue if this revalidation verifies the IHI. However, if the revalidation indicates the IHI is invalid then the upload is not performed.
3. The *Episode* is resolved by finding *Episode* records matching the resolved *HospitalPatient* and provided *AdmissionDate*. *Episodes* will be found if they are the only *Episode* for the *HospitalPatient* within a configured tolerance of minutes of the provided *AdmissionDate*. *Episodes* will also only be found if they do not have a cancelled *EpisodeLifecycle*.
4. The *DocumentId* and *DocumentSetId* are generated internally by eHISC and used for the CDA document. These are sequentially generated numbers from the *CDASetNumber* and *CDADocumentNumber* tables.
5. The *CDASetNumber* table contains an auto-generated, sequentially increasing, *DocumentSetId* number with the related *EpisodeId*, *AdmissionDateTime*, *DischargeDateTime*, *ModeOfSeparation* *DocumentFormat* and *DocumentType* code.  
The *CDADocumentNumber* table contains an auto-generated, sequentially increasing, *DocumentId* number with the related *DocumentCreationDateTime* and *CDASetNumber* record.
6. If the *Episode* does not already have a *ClinicalDocument* record then the *DocumentSetId* is auto generated as the next *DocumentSetId* available. The *DocumentId* is also auto generated as the next *DocumentId* available.
7. If the *Episode* already has a *ClinicalDocument* record for an eDischarge Summary then the *DocumentSetId* will be set to the same value as the previously created *ClinicalDocument*.
  - a. If the *DocumentCreationDateTime*, *AdmissionDateTime*, *DischargeDateTime*, and *ModeOfSeparation* is the same as another record for the same *DocumentSetId* then this would indicate that the same document was attempting to be uploaded again. In this case an error will be returned to the user specifying that the same document is attempting to be uploaded.
  - b. If the *DocumentCreationDateTime*, *AdmissionDateTime*, *DischargeDateTime*, and *ModeOfSeparation* is not the same as another record for the same *DocumentSetId* then the *DocumentId* will be auto generated as the next *DocumentId* available.
8. The *PatientIdentifier* maybe either type of *Mrn*, *PatientMasterId*, *StatePatientId* or *ValidatedIhi*. The *PatientIdentifier Demographic* type will not be resolved and if used will return an error.
9. As it is suggested that the HL7 *DatabaseLoader* is used to gather the patient information within eHISC for a 1A upload, but not essential, then the *PatientIdentifier* is best utilised as the *Mrn*, or *StatePatientId* (if one is used). If the *ValidatedIhi* is used then the *PatientAddress* and the *PatientContactDetails* will become mandatory as they are required information for a Level 1A Compliant Discharge Summary upload to the PCEHR.
10. The *ParticipatingProvider* with the *CdaHeaderMetadata* object has a conditional mandatory rule over the *LocalIdentifier* and *Hpii* attributes as at least one must be populated or else an error will be returned. If both of the *LocalIdentifier* and *Hpii* attributes are added then the *Hpii* will be selected over the *LocalIdentifier*.

## 2.2.8 Remove Document from PCEHR

This service implements a “fire and forget” pattern that adds a request to a queue for removing a document from the associated repository for the document type. The service will return as soon as the item is added to the queue.

### **2.2.8.1 Business Rules / Functional Business Logic**

The remove document operation will have identical patient matching, IHI validation and episode matching logic to the upload or supersede document operation.

### **2.2.8.2 Cancellation of a Queued Remove Operation**

It may be necessary to manually cancel a queued remove operation, if the PCEHR system returns an error message that is documented to have the meaning of system temporarily unavailable, but is actually due to an error in the removal request. This can be done by setting the queue status in the PcehrMessageQueue record to 3 (failed).

Each time a remove operation is taken off the queue for processing, eHISC checks the current value of the QueueStatusId in the PcehrMessageQueue record. If the queue status is no longer set to 1 (pending) then eHISC will stop processing this operation, and the queue processor will move to the next operation in the queue.

### **2.2.8.3 Ordering of Queued Operations**

If the PCEHR system is temporarily unavailable, the queued remove operation will be placed on a retry queue. The retry queue is standard functionality of the MSMQ binding in the web.config file. When a message has reached the receiveRetryLimit (set to 3) it is then paused to wait for a period of time set by the retryCycleDelay (set to 5 minutes). The maxRetryCycles is set to 6000 so that the PCEHR can be completely offline for a minimum of 20 days (excluding connection timeout delays for each retry) before the MSMQ is placed into a halted state.

While one or more operations are on the retry queue, other remove requests can arrive. eHISC will attempt to process these operations immediately as the PCEHR system may now have sufficient capacity to process these requests.

However it is important to ensure that requests that relate to the same document set are processed in the correct order. Therefore, each time a remove operation is taken off the queue for processing, eHISC checks whether there are any earlier pending operations for the same document set. If there are any, then eHISC will roll back the queue transaction, so that the remove operation will be retried after the earlier request is completed.

### **2.2.8.4 Auditing of Document Removal**

The additional audit information provided in the "auditInformation" parameter will be stored in the RemoveAudit table.

### **2.2.8.5 Auditing of Request and Response**

After sending the remove document request to the PCEHR system, eHISC will write an audit record into the PcehrAudit table. This audit record contains the full SOAP request and response.

### **2.2.8.6 Response Classification**

Using the definitions of the PCEHR error messages in the Technical Service Specification, each response from the PCEHR remove document operation will be classified as one of:

- Success

- System Temporarily Unavailable
- Unrecoverable Error

In the case of a success message, the queued operation will be deleted and the ClinicalDocument record will be updated. eHISC will set the DocumentStatus to removed, set the RemovalDate to the current time, and set the RemovalReason to the value provided in the “reason” parameter.

In the case of a message classified as System Temporarily Unavailable, eHISC will roll back the queue transaction, so that the remove operation will be retried. The retry behaviour is controlled by the MSMQ settings in the web.config file. See the eHISC Installation Guide for details.

In the case of an unrecoverable error, the queued operation is marked as failed, and the queue processor will move to the next message in the queue.

## **2.2.9 Get Operation Status**

This service is used to return the list of pending and failed Queued Operations, Uploaded Documents and Document Versions for a particular patient for a specific episode.

### **2.2.9.1 Business Rules / Functional Business Logic**

This service is intended for system analysis of pending or failed operation requests from the eHISC operational message queue.

## **2.2.10 Get Queued Operation List**

This service is used to return a list of active pending and failed Queued Operations.

### **2.2.10.1 Business Rules / Functional Business Logic**

This service is intended as a simplistic list of pending or failed operation requests from the eHISC operational message queue. Results returned can then be used to perform a request on the more detailed Get Operation Status (either with the “GetOperationStatus” or the “GetIndividualOperationStatus”) service for more detailed analysis.

## **2.2.11 Search for Provider Individual Details - Synchronous**

### **2.2.11.1 Description**

This service is used to search for the Provider Details for an individual in a synchronous process. A provider search can be performed as either an identifier search to validate a known HPI-I, or to retrieve the HPI-I using the registration id (for example the AHPRA number) thus verifying the provider or perform a demographic search to find the provider.

### **2.2.11.2 Business Rules / Functional Business Logic**

- 1) Following a HPI-I number/registration Id or Demographic HPI-I record search, if the HI Service finds a single match to an HPI-I record which is a ‘Resolved’ duplicate as a status, the HI Service will return information message (WSE0134) and the primary HPI-I record with search criteria details.

**2) HpiiIdentifierQuery:**

- a. The hpiiNumber and registrationId and number are conditionally mandatory and thus only one of the two items must be provided for a single search.
- b. The hpioNumber is a mandatory requirement and must be provided for auditing requirements.
- c. The familyName is also a mandatory requirement.
- d. The familyName and givenName must not provide invalid characters. Only alpha and numeric characters, apostrophes, full stops and hyphens are acceptable. Spaces are also acceptable but must not appear immediately before or after apostrophes and hyphens.
- e. When searching by registrationId the search will be case sensitive.

**3) hpiiDemographicQuery:**

- a. The hpioNumber is a mandatory requirement and must be provided for auditing requirements.
- b. The familyName is a mandatory requirement.
- c. The familyName and givenName must not provide invalid characters. Only alpha and numeric characters, apostrophes, full stops and hyphens are acceptable. Spaces are also acceptable but must not appear immediately before or after apostrophes and hyphens.
- d. The dateOfBirth is a mandatory requirement.
- e. The sex is a mandatory requirement.
- f. The australianAddress and internationalAddress are conditionally mandatory and thus only one of the two items must be provided for a single search.
  - i. If the australianAddress is provided then the following are mandatory:
    1. streetName
    2. suburb
    3. postcode
    4. state
  - ii. If the internationalAddress is provided then the following is mandatory:
    1. country

## **2.2.12 Search for Provider Individual Details – Asynchronous**

### **2.2.12.1 Description**

This service is used to search for the Provider Details for an individual in an asynchronous process.

### **2.2.12.2 Business Rules / Functional Business Logic**

- 1) All rules for the HpiiIdentifierQuery and HpiiDemographicSearch apply for all validation and search requests.
- 2) HpiiBatchSubmit:
  - a. Asynchronous batch searching allows provider individuals to search for up to 50 search requests per message. A badlyFormedMsg will be returned in the response file when the batch file size is greater than 50 requests.



3) HpiiBatchRetrieve

- a. The HPI-I Service will only permit the requestor of the HPI-I batch search to retrieve the details of the HPI-I batch search. The HI Service will validate the requestor for the services: and a not authorised error will be returned when the requestor is invalid – error WSE9050.
- b. The HPI-I Service will hold the completed search results for a period of exactly 14 calendar days from the date the batch request has reached a COMPLETED status. The results will be deleted after the timeframe has elapsed for searches when the status of the batch request is 'Completed', 'Retrieved' or 'Error'.

## 2.2.13 Gain PCEHR Access

### 2.2.13.1 Description

This function is used when a health provider organisation (HPO) wants to gain access to an individual's PCEHR for subsequent viewing and/or downloading information from the PCEHR.

This can be performed:

- With an Access Code (Open Access),
- Without an Access Code or
- With Emergency Access

A call to the "IsPcehrAdvertised" method described in the Document Production specification will return the following access code required for the individual's PCEHR:

- Null – thus meaning that the individual has not registered for PCEHR, or has chosen to hide the existence of his/her PCEHR. The individual may give advice of PCEHR existence and may or may not provide an access code.
- With Code – thus meaning that a code must be provided to gain access
- Without Code – thus meaning that access is open and no Record Code is required. The patient may advise of a Document Code to grant restricted access.
- Access Granted – thus meaning that access has been granted and no code is required unless the patient advises of a change to access level and provides an access code.

As an individual can change their access at any time and so a call to the "IsPcehrAdvertised" method should be performed before each call to the gain access function.

NOTE: The "AccessCodeRequired" for an individual is stored in the eHISC "HealthProviderOrganisationPatient" table after "IsPcehrAdvertised" has been called. However, since this is a transactional system and other calls may already be updating the "AccessCodeRequired" in another thread, it is essential that a call to the "IsPcehrAdvertised" is performed to ensure complete up-to-date values of the "AccessCodeRequired"

### 2.2.13.2 Business Rules / Functional Business Logic

The following business rules have been derived from the NEHTA diagram entitled “Accessing & Viewing the PCEHR”, and reproduced below:

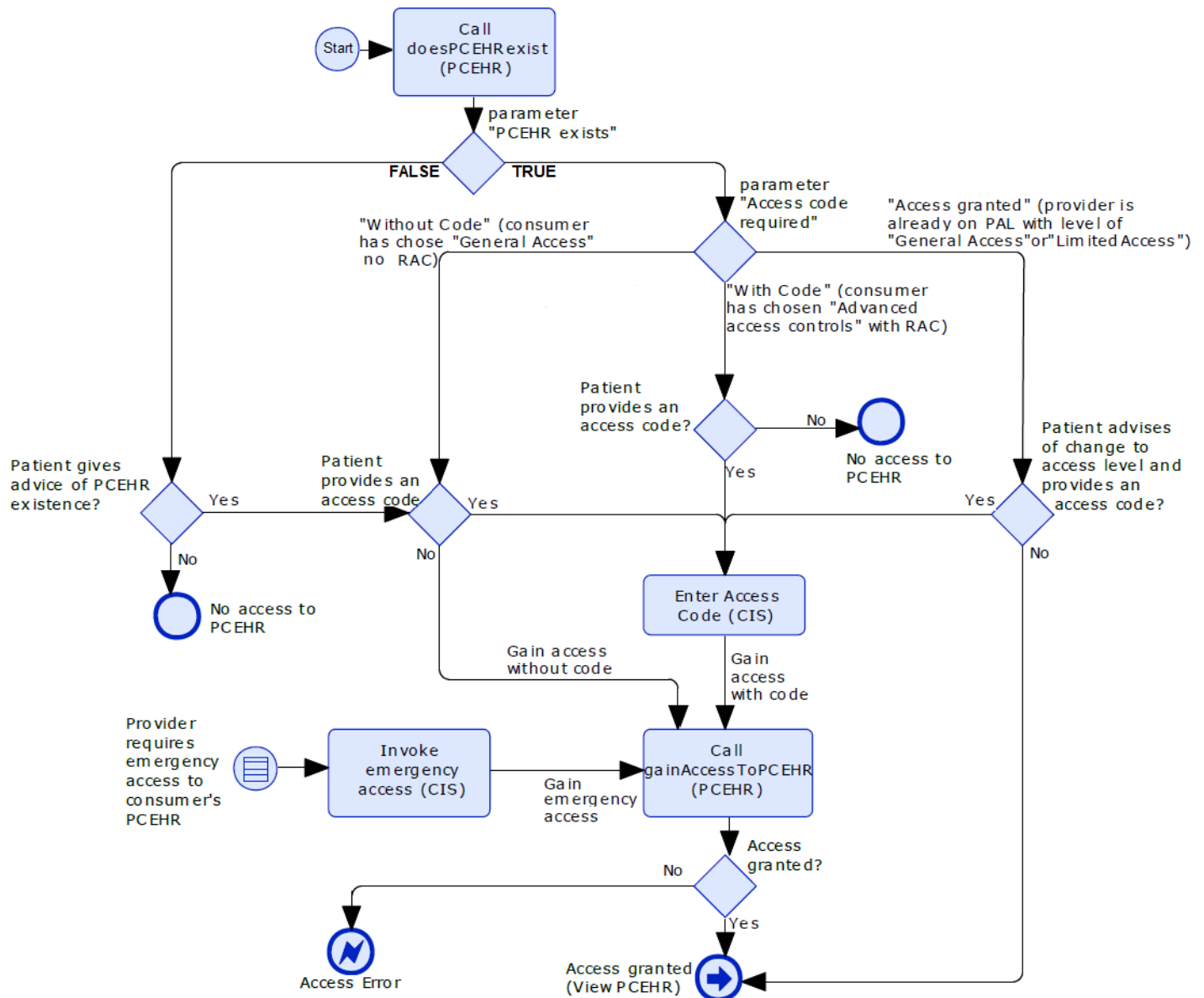


Figure 1: Gain Access Business Logic Diagram from NEHTA

The derived business logic is:

1. The IHI number must be verified and must be added to the PCEHR header. All calls must use the IHI value.
2. With the verified IHI number:
  - a. If Access Type for the individual's PCEHR is "WithCode" then the "AccessType" of the patient must be added into "GainPCEHRAccess">"PCEHRRecord">"AuthorisationDetail" object. This can be either as "AccessCode" or "EmergencyAccess"
    - i. If the "AccessType" is set to "AccessCode" then the individual's PCEHR access code must be added to the "GainPCEHRAccess">"PCEHRRecord">"AuthorisationDetail">"AccessCode" field.

- b. If Access Type for the individual's PCEHR is "WithoutCode" and the individual has not given the provider an access code, and the provider does not choose to assert emergency access, then the "GainPCEHRAccess">"PCEHRRecord">"AuthorisationDetail" object will be set to NULL. If the individual has given the provider an access code or the provider chooses to assert emergency access, then the "AccessType" of the patient must be added into "GainPCEHRAccess">"PCEHRRecord">"AuthorisationDetail" object.
- c. If Access Type for the individual's PCEHR is "AccessGranted" and the individual has not given the provider an access code, and the provider does not choose to assert emergency access, then "GainPCEHRAccess" would not need to be actioned. If the individual has given the provider an access code or the provider chooses to assert emergency access, then "GainPCEHRAccess" must be actioned and the "AccessType" of the patient must be added into "GainPCEHRAccess">"PCEHRRecord">"AuthorisationDetail" object.
- d. If the parameter "PCEHR Exists" is false and therefore the Access Type for the individual's PCEHR is "null":
  - i. If the provider requires emergency access to the PCEHR then the AccessType of EmergencyAccess must be added into the GainPCEHRAccess.
  - ii. Otherwise if the individual has not advised of PCEHR existence then no access is possible and the GainPCEHRAccess must not be actioned.
  - iii. Otherwise if the individual has advised of PCEHR existence but has not provided an access code, then GainPcehrAccess must be actioned and the AuthorisationDetail object will be set to NULL.
  - iv. Otherwise if the individual has provided an access code then GainPcehrAccess must be actioned and the AccessType of AccessCode must be added into the GainPCEHRAccess > PCEHRRecord > AuthorisationDetail object.

## 2.2.14 Document List – Registry Stored Query (ITI-18)

### 2.2.14.1 Description

This function is used to retrieve a list of documents from the PCEHR repository. There are several standard overloaded GetDocumentList methods for simple queries; if a more complex query is required then the GetDocumentList that uses the DocumentQuery object should be used. This is also described as the getDocumentList() function.

Zero or more listed documents can be returned from this function.

### 2.2.14.2 Business Rules / Functional Business Logic

1. The IHI number must be verified and must be added to the PCEHR header. All calls must use the IHI value.
2. If the count of the list of documents from a successful repository request is zero (0) then a GetDocumentListResponse object will still be returned, however the DocumentList attribute will have a null or empty list.
3. For the methods that use the CreationTimeStart, CreationTimeEnd, ServiceTimeStart and ServiceTimeEnd; they are all optional parameters. If null values are passed then the query request will not include those parameters.
4. CreationTimeStart, CreationTimeEnd, ServiceTimeStart and ServiceTimeEnd parameters have a minimum precision value of a second.
5. Entering a value for the parameter for the ServiceTimeStart will enter a query against the ServiceStartTimeFrom to return all documents whose ServiceStartTime is after this value entered.
6. Entering a value for the parameter for the ServiceTimeEnd will enter a query against the ServiceStopTimeTo to return all documents whose ServiceStopTime is before this value entered.

7. Using the DocumentQuery object the DocumentClassCode, DocumentStatus, FormatCode, HealthCareFacilityType and PracticeSettingTypes are all ILists and thus can have 1 or more values passed in.

## **2.2.15 Download Document - Retrieve Document Set (ITI-43)**

### **2.2.15.1 Description**

This function is used to retrieve a single XDS document element. ("DEXS-T 19" & "DEXS-T 21" from "PCEHR Document Exchange Service IHE XDS-b Technical Service Specification v1.3"). This is also described as the `getDocument()` function.

### **2.2.15.2 Business Rules / Functional Business Logic**

1. Only a single document is returned from each call.
2. The returned document is unpacked and unencrypted as a CDA document along with any attachment files included in the package.
3. If the Save Document flag in the Document Request parameter is set, then eHISC will save a copy of the downloaded document in the Downloaded Document table in the local database.
4. If the consumer's IHI, date of birth, sex or family name in the downloaded document does not match the relevant stored information for the patient, a Demographic Mismatch Warning will be returned in the eHISC response. It is recommended that a user-facing application displaying the document makes this warning visible to the user.

## **2.2.16 Change History View**

### **2.2.16.1 Description**

This function is used to retrieve a list of all versions of a document from the PCEHR repository, that are previous or subsequent versions of to a single identified document instance. The documents that are returned are the historical documents from a document tree.

### **2.2.16.2 Business Rules / Functional Business Logic**

1. The IHI number must be verified and must be added to the PCEHR header. All calls must use the IHI value.
2. The unique document ID, which is passed to this method, can be any of the unique document IDs that are within the historical document tree. Thus, no matter which unique document ID is passed all historical documents from the set will be returned.

## **2.2.17 Get View**

### **2.2.17.1 Description**

This function is used to access a PCEHR view service, using the parameters that are defined for the view service, and receive the results as a CDA document.

This version of eHISC supports only the Prescription and Dispense View, however, the web service interface is designed to be extended to support any PCEHR view service that is available via the 'GetView' interface.

### **2.2.17.2 Business Rules / Functional Business Logic**

1. The patient must be identified with a verified IHI number. eHISC will ensure that the IHI number is verified and added to the PCEHR header. All calls must use the IHI value.
2. The “user” object which is passed to this method must contain the name, role and identifier of the person responsible for the action, who is typically the interactive user of the clinical system. If the HPI-I of the person is known, the Role must be “ProviderIndividual” and the HPI-I must be provided.
3. The “parameters” object which is passed to this method must be an instance of “PrescriptionAndDispenseViewRequest” which is the only concrete subclass of the abstract base class ‘ViewParametersBase’ in this version of eHISC.

## **2.2.18 List Patients in Hospital**

### **2.2.18.1 Description**

This function is used to access a list of patients in hospital who have an active verified IHI. The list can be filtered to include patients with or without a PCEHR, with or without with an IHI, and to include patients who were discharged within a specified number of days.

### **2.2.18.2 Business Rules / Functional Business Logic**

1. The “user” object which is passed to this method must contain the name, role and identifier of the person responsible for the action, who is typically the interactive user of the clinical system. Although this function does not access the HI Service or PCEHR system, any errors that occur during the operation will be logged with the user’s identity.
2. The hospital code, the “hospitalCodeSystem” object, which is passed to this method acts to scope the hospital code.
3. While there may be several episodes that fall within the specified number of days since discharge, only one record is returned for each patient in each hospital, with the details from the most recent matching episode.
4. Matching patients are either current inpatients (as created by an ADT-A01 message) or recently discharged inpatients (as created by an ADT-A03 message), and must have an active verified IHI.

## **2.2.19 List Patients Episodes in Hospital**

### **2.2.19.1 Description**

This function is used to access a list of admitted or discharged episodes for a specific patient in a hospital. The list can be filtered to include patients who were discharged within a specified number of days as well as include all the documents for a single document type for the patient.

### **2.2.19.2 Business Rules / Functional Business Logic**

5. The “user” object which is passed to this method must contain the name, role and identifier of the person responsible for the action, who is typically the interactive user of the clinical system. Although this function does not access the HI Service or PCEHR system, any errors that occur during the operation will be logged with the user’s identity.
6. The hospital code, the “hospitalCodeSystem” object, which is passed to this method acts to scope the hospital code.
7. There may be several episodes that fall within the specified number of days since discharge and these will be included as well as currently admitted episodes.

8. Matching patients are either current inpatients (as created by an ADT-A01 message) or recently discharged inpatients (as created by an ADT-A03 message).

## 2.2.20 Assisted Registration

### 2.2.20.1 Description

Assisted Registration was built to allow the fast and easy registration of patients to the PCEHR.

### 2.2.20.2 Business Rules / Functional Business Logic

The Assisted Registration Service in eHISC will implement the following business rules:

- Provide the ability to register adults (over 14 years of age) with the PCEHR. The registration of dependants is not in scope for this project.
- Maintain an audit log of successful registrations. If the verification method is response then the return code must be masked or encrypted in the log.
- Allow the submission of scanned copies of the registration consent form.
- Allow all Identity Verification Methods to be used.
- Provide a list of patients with an IHI but without a PCEHR using a simple data protocol.
- Allow patient registration based on the identifiers from the list and the patient's assertions. When using identifiers other than a verified IHI the associated IHI should have been verified within the previous 24 hours or eHISC should re-verify the IHI.
- Allow patient registration based on a Verified IHI and the patient's assertions.
- Allow patient registration based on a patient's demographics and assertions.
- Allow IHI validation to occur within eHISC prior to sending the registration request to the PCEHR when using patient demographics.
- Allow the registration request to be sent to the PCEHR without IHI validation when using patient demographics.
- Must meet all NOC and CCA mandatory tests.

### 2.2.20.3 Functional Validations

The following errors may be generated due to functional validations of the registration request. In the table below, RP refers to the "RegisterPatient" service that is for assisting an individual to register for their own PCEHR, while RD refers to the "RegisterDependant" service that is for assisting a parent to register for their child's PCEHR.

| eHISC Error Message                             | Applies to |
|---|------------|
| Individual's details must be provided.          | RP & RD    |
| The Indigenous Status must be provided.         | RP & RD    |
| The Evidence Of Identity Type must be provided. | RP & RD    |
| No given name provided. <sup>1</sup>            | RP & RD    |

---

<sup>1</sup> Prefixed with "Representative: " or "Patient: " to indicate which person's demographics were invalid

| eHISC Error Message   | Applies to |
|---|------------|
| No family name provided. <sup>1</sup>   | RP & RD    |
| Cannot have future dated date of birth. <sup>1</sup>                                      | RP & RD    |
| Date of birth cannot be more than 140 years ago. <sup>1</sup>                             | RP & RD    |
| Must include a Medicare or DVA number. <sup>1</sup>                                       | RP & RD    |
| Medicare number is incorrect. <sup>1</sup>  | RP & RD    |
| Medicare IRN must be a number between 1 and 9. <sup>1</sup>                               | RP & RD    |
| Representative details must be provided.  | RD only    |
| Consent form {0} with size {1}B exceeds the 200,000B limit for uploading to PCEHR.        | RP & RD    |
| Filename contains path separator characters.  | RP & RD    |
| Consent form {0} has an invalid filename.   | RP & RD    |
| Consent form {0} is not a supported type for uploading to PCEHR.                          | RP & RD    |
| Must provide a response for ACIR documents consent.                                       | RP & RD    |
| Must provide a response for AODR documents consent.                                       | RP & RD    |
| Must provide a response for MBS documents consent.  | RP & RD    |
| Must provide a response for MBS Past Assimilation documents consent.                      | RP & RD    |
| Must provide a response for PBS documents consent.  | RP & RD    |
| Must provide a response for PBS Past Assimilation documents consent.                      | RP & RD    |
| The latest terms and conditions have not been accepted.                                   | RP & RD    |
| IVC Correspondence Channel has not been specified.  | RP & RD    |
| Mobile phone number is required for IVC SMS correspondence.                               | RP & RD    |
| Invalid mobile phone number.  | RP & RD    |
| Email address is required for IVC email correspondence.                                   | RP & RD    |
| Invalid email address.  | RP & RD    |
| Representative declaration is required for assisted registration.                         | RD only    |
| Dependant cannot be older than 18 years.  | RD only    |
| There cannot be less than a 14 year age gap between the dependant and the representative. | RD only    |
| The representatives Medicare number must be provided.                                     | RD only    |
| The dependant and the representative must be on the same Medicare card.                   | RD only    |
| An individual cannot be less than 14 years old.   | RP only    |



## 2.2.21 Contracted Service Provider (CSP) Usage

### 2.2.21.1 CSP for HI Service

The 'CSP for HI Service' changes how eHISC invokes the HI Service when requested by a health provider organisation to search or validate IHI or HPI-I numbers.

For HPOs that provide their Medicare site certificate to the operator of eHISC, the standard invocation logic will apply. The HPI-O of the accessing organisation is not required and not permitted in the SOAP header because the HI Service determines the accessing organisation's identity based on the certificate that is presented.

For HPOs that do not provide their Medicare site certificate to the operator of eHISC, but instead nominate to Medicare that the operator of eHISC is their Contracted Service Provider (CSP), altered invocation logic applies. Medicare would issue a CSP certificate to the operator of eHISC and this certificate must be stored as the certificate to be used by eHISC when invoking the HI Service for this HPO. In this case, the HPI-O of the accessing organisation is included in the SOAP header on each invocation of the HI Service. This is configured by populating the HiCertSerial with the CSP certificate serial number and setting the HiCsp column to true in the HealthProviderOrganisation table.

### 2.2.21.2 CSP for PCEHR System

The 'CSP for PCEHR System' changes how eHISC invokes the PCEHR B2B Gateway when requested by a health provider organisation to access the PCEHR of a patient.

For HPOs that provide their NASH certificate to the operator of eHISC, the standard invocation logic will apply. The client system type "CIS" is presented in the SOAP header on each invocation.

For HPOs that do not provide their NASH certificate to the operator of eHISC, but instead nominate to DHS that the operator of eHISC is their Contracted Service Provider (CSP), altered invocation logic applies. Medicare would issue a NASH Supporting Organisation certificate to the operator of eHISC and this certificate must be stored as the certificate to be used by eHISC when invoking the PCEHR B2B Gateway for this HPO. In this case, the client system type "CSP" is presented in the SOAP header on each invocation. This is configured by populating the PcehrCertSerial with the CSP certificate serial number and setting the PcehrCsp column to true in the HealthProviderOrganisation table.

## 2.2.22 Multi-Tenant for IHI

The 'Multi-Tenant for IHI' modifies how eHISC manages the assignment of an IHI number to each patient record, in the context of multiple Health Provider Organisations (HPO) using the one instance of eHISC.

eHISC has a concept of networks, where organisations that have a common seed organisation can belong to the same network. These are configured in the *HealthProviderOrganisationNetwork* table within eHISC. eHISC must have at least one network configured.

For HPOs that belong to one network when an IHI is obtained by a network organisation, it is stored against the shared patient record and is available for use by any other network organisation during the configured period. This aligns with existing practice in a network of public hospital facilities.

For HPOs that do not belong to the same network, the IHI will be obtained and stored separately for each seed organisation in eHISC. The IHI that is obtained by one organisation will not be available for use by organisations that are in different networks. When organisations in other networks request the IHI or attempt to access the PCEHR, eHISC will perform a new search of the HI Service to obtain the IHI for the new organisation. This ensures that the HI Service audit log records a disclosure of the IHI to the new organisation.

### 2.2.23 Reference Services

The following items in the eHISC database are considered reference data and are cached in memory for faster access.

| Schema Item         | Tables   | Information Represented  |
|---------------------|--|--|
| Hospital            | Hospital<br>HealthProviderOrganisation<br>HospitalAddress, Address<br>HospitalContact, Contact<br>HospitalCode, CodeSystem | Name, Authorised Employee, Logo Image<br>HPI-O, Certificate Serial Numbers<br>Hospital Addresses<br>Hospital Contact Methods (e.g. Phone, Fax)<br>Hospital Codes   |
| Title               | Title  | Name Titles (Dr, Ms, Mr, etc.)   |
| Suffix              | Suffix   | Name Suffixes (Senior, Junior, etc.)   |
| Sex                 | Sex  | Sex Codes and Descriptions   |
| Country             | Country  | Country Codes and Descriptions   |
| State               | State  | Australian State Codes and Descriptions  |
| EpisodeType         | EpisodeType  | Episode Type Codes and Descriptions  |
| CodeSystem          | CodeSystem   | Coding Systems and Namespaces  |
| HospitalCode        | HospitalCode<br>CodeSystem   | Hospital Codes<br>These codes are used to look up a hospital.  |
| AddressType         | AddressType  | Address Types<br>(Home, Temporary, Business, Mailing, etc.)  |
| EpisodeLifecycle    | EpisodeLifecycle   | Episode Lifecycle Statuses<br>(Pre-admit, Admitted, Discharged, Cancelled, etc.)   |
| DocumentType        | DocumentType   | Document Types<br>(Discharge Summary, etc.)  |
| DocumentFormat      | DocumentFormat<br>CodeSystem   | Document Format Codes / Template IDs<br>For example, the format code “1.2.36.1.2001.1006.1.20000.16” represents the validation rules for “PCEHR Release 3 Discharge Summary Level 3A”, with the relaxation of display name for mode of separation. |
| MedicareExclusion   | MedicareExclusion  | Certain values that are populated by PAS systems in the MedicareNumber field, which indicate that the person is ineligible for Medicare benefits or that the Medicare number is unknown.   |
| ResponseChannel     | RegistrationResponseChannel  | Registration response codes for Assisted Registration.   |
| DocumentConsentType | RegistrationConsentType  | Registration document consent code for Assisted Registration.  |

| Schema Item          | Tables                           | Information Represented                                   |
|----------------------|----------------------------------|---|
| IndigenousStatusType | IndigenousStatusType             | Indigenous status codes for Assisted Registration.        |
| IDEvidenceType       | RegistrationIdentityEvedenceType | Identity evidence method codes for Assisted Registration. |

#### **2.2.23.1 Reload Reference Data**

This method instructs eHISC to reload all reference data from the database. The system administrator can invoke this method after making a change to reference data in the eHISC database, instead of restarting the eHISC application server.

#### **2.2.23.2 Get Hospital Details**

This method returns a set of information about a specified hospital, which is useful for the generation of a CDA document from that hospital.

### **2.2.24 Common Schemas**

#### **2.2.24.1 User Details**

The user is included as a parameter on all calls to eHISC in order to assert the authorisation role under which any calls to the HI Service or PCEHR System should take place.

#### **2.2.24.2 eHISC Response**

This object is used to wrap up a response status indicator, error code, description and details that are returned from eHISC to calling systems.

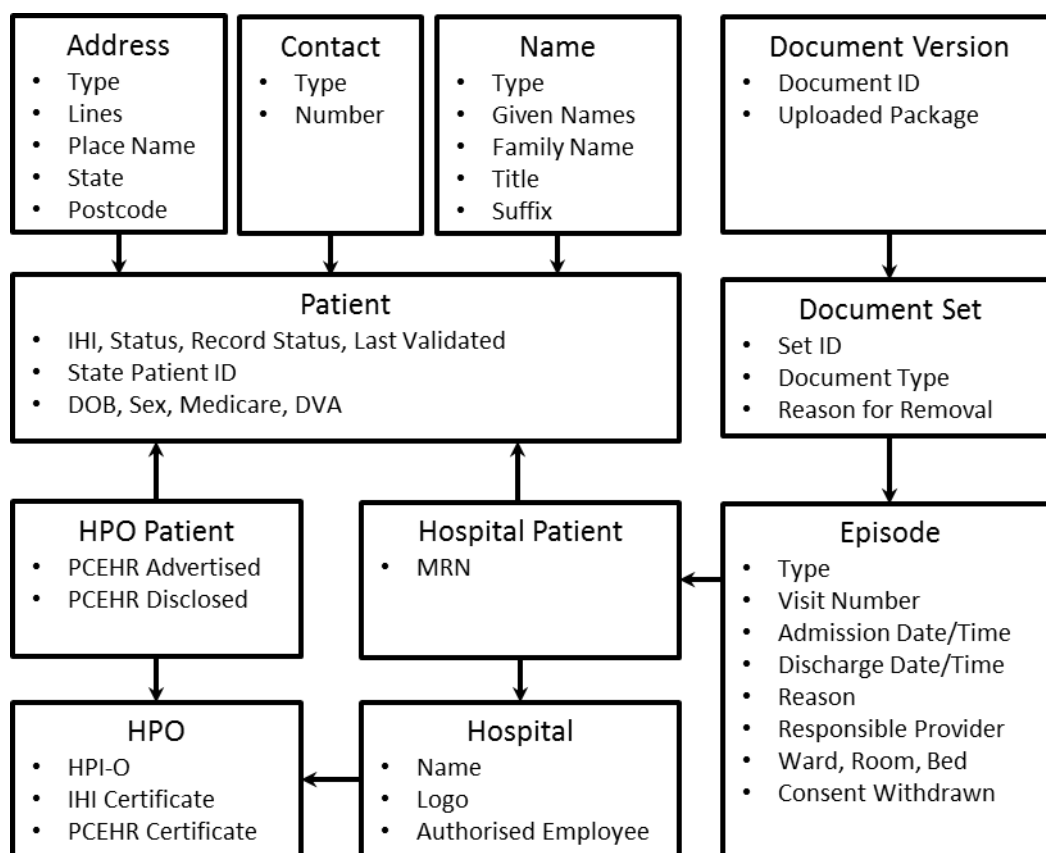
## 2.3 Database Resource Access Layer

The PCEHR Data Store contains certain information about hospital patients that is relevant and required for the PCEHR connectivity in eHISC.

The key requirements for this design were:

- Each hospital belongs to a Health Provider Organisation (HPO) with an HPI-O.
- A disclosure of PCEHR can be stored for each HPO that a patient visits.
- A withdrawal of consent to upload documents can be stored for each episode.

An overview of the data model is given below:



### 2.3.1 HealthProviderOrganisationPatient Table

After eHISC calls the PCEHR's "doesPCEHRExist" service method, the result is stored in the HPO Patient table, "HealthProviderOrganisationPatient".

The HealthProviderOrganisationPatient table will have the following columns:

| Column                       | Type         | Description   |
|------------------------------|--------------|---|
| HealthProviderOrganisationId | int          | The HPO to which this record relates.   |
| PatientMasterId              | int          | The patient to which this record relates.   |
| PcehrAdvertised              | nullable bit | The value that the PCEHR System last indicated to this HPO as to whether the PCEHR for this patient exists.   |
| AccessCodeRequiredId         | int          | The value that the PCEHR System last indicated to this HPO as to whether access to this patient's PCEHR is granted and whether a code is required for this HPO to gain access:<br>-1: Unknown<br>0: With Code<br>1: Without Code<br>2: Access Granted |
| PcehrDisclosed               | bit          | Whether the patient has disclosed the existence of his/her PCEHR to this HPO. The patient is known to be participating in the PCEHR System if the patient has disclosed the existence of a PCEHR.   |
| DateCreated                  | datetime     | The date and time when the record for this HPO and patient was created.   |
| UserCreated                  | varchar(256) | The domain and login of the user identified by the source system as responsible for the action that triggered the creation of this record.  |
| DateModified                 | datetime     | The date and time when the record was last modified.  |
| UserModified                 | varchar(256) | The domain and login of the user identified by the source system as responsible for the action that last modified this record.  |

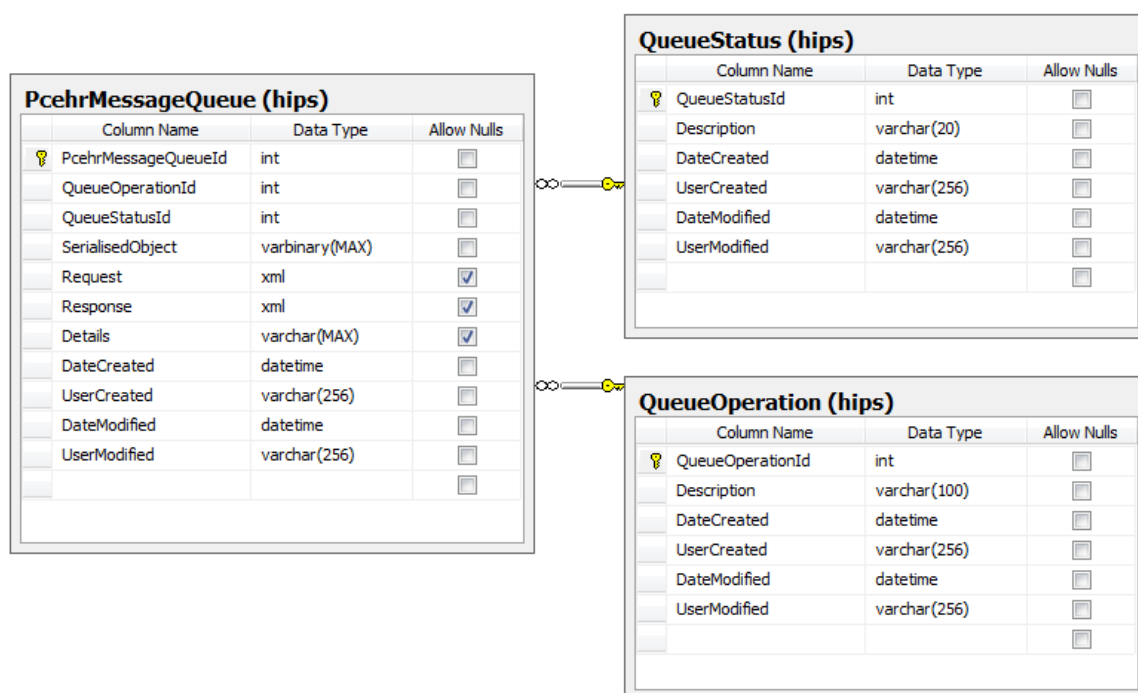
## 2.3.2 Queuing Data Model

After each request to UploadOrSupersedeDocument or Remove has been validated, a record of the queued operation with “Pending” status will be added to the “PcehrMessageQueue” table in the PCEHR Data Store as illustrated below, if and only if the operation is handed off to the Microsoft Message Queue (MSMQ).

Normal processing is to delete these records of the queued operations once the operation has successfully completed, and so the table will contain only pending operations and failures. We have added a configuration item to allow successful operations to be retained for testing purposes.

The configuration item “DeleteQueuedItemOnSuccess” will default to true. If this configuration item is true then the record of the queued item will be deleted after successful processing.

If the queued operation is unsuccessful, or the above configuration item is set to false, then the QueueStatusId column will be updated to reflect the Success or Failure as appropriate, the SOAP request and response will be populated into the Request and Response columns, and any additional error information including a eHISC exception or stack trace will be populated into the Details column.



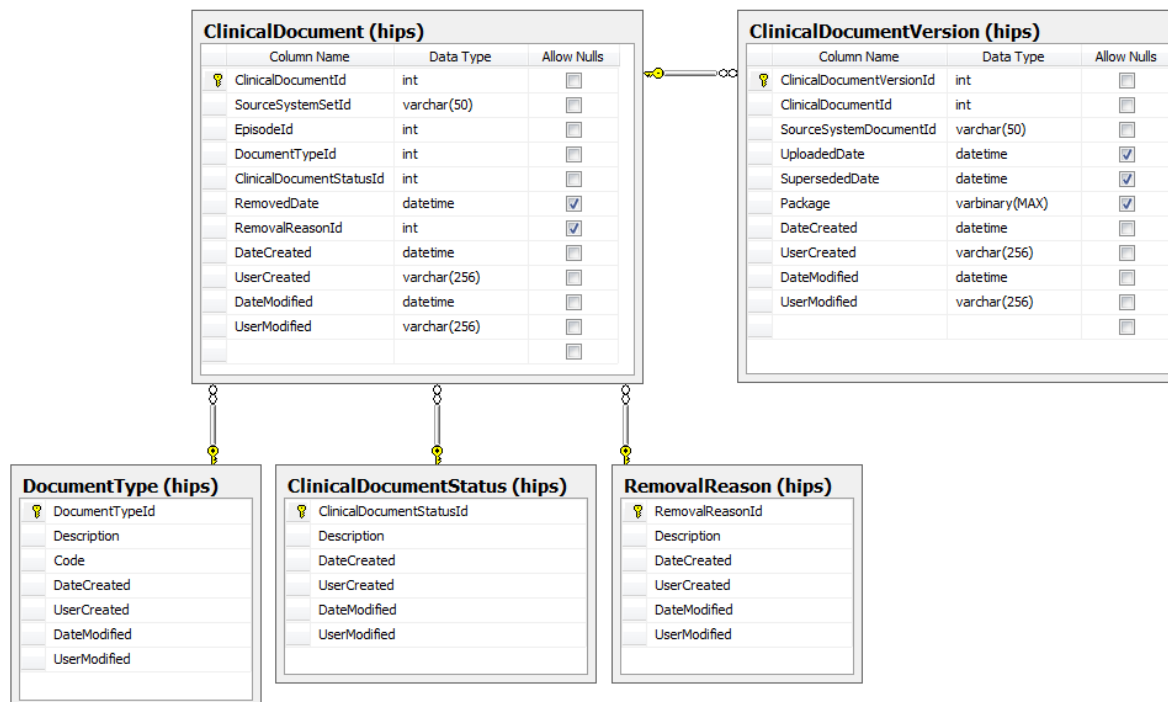
### 2.3.2.1 PcehrMessageQueue Table

The PcehrMessageQueue table will have the following columns:

| Column              | Type           | Description   |
|---------------------|----------------|---|
| PcehrMessageQueueId | int            | Auto-incrementing primary key   |
| QueueOperationId    | int            | The operation type that has been queued:<br>UploadOrSupersede<br>Remove   |
| QueueStatusId       | int            | The status of the queued operation:<br>Pending<br>Success (unused by default)<br>Failure                              |
| SerializedObject    | varbinary(max) | The .NET serialised object that has been added to the MSMQ.   |
| Request             | XML            | The SOAP request sent to the PCEHR B2B Gateway. Only populated after the queued operation has been processed.         |
| Response            | XML            | The SOAP response received from the PCEHR B2B Gateway. Only populated after the queued operation has been processed.  |
| Details             | varchar(max)   | The exception message and stack trace from eHISC showing where a failure in processing the queued operation occurred. |
| DateCreated         | datetime       | The date and time when the queued operation was created.  |
| UserCreated         | varchar(256)   | The domain and login of the user identified by the source system as responsible for the action.                       |
| DateModified        | datetime       | The date and time when the queued operation was last modified.  |
| UserModified        | varchar(256)   | The domain and login of the user identified by the source system as responsible for the action.                       |

### 2.3.3 Clinical Document Data Model

Documents that are *successfully* uploaded, superseded or removed will be stored within the PCEHR Data Store in the data model illustrated below.



All versions of a document that are uploaded to PCEHR by eHISC will be associated with an episode, which is associated with a patient in a certain hospital, which is associated with the patient master record and the hospital record.

The PCEHR Data Store model makes this hierarchy explicit, and assigns identifiers at each level:

- Clinical Document Version (source system document ID)
  - Clinical Document (source system set ID)
    - Episode (visit ID, admission date/time)
      - Hospital Patient (MRN)
        - Patient Master (enterprise patient ID, IHI)
        - Hospital (Hospital Code)
          - Health Provider Organisation (HPI-O)



### 2.3.3.1 ClinicalDocument Table

The “ClinicalDocument” table stores the information that relates to the overall document.

| Column                   | Type                | Description  |
|--------------------------|---------------------|--|
| ClinicalDocumentId       | int                 | Auto-incrementing primary key  |
| SourceSystemSetId        | varchar(50)         | The source system’s unique identifier of the overall document, which must not change between versions of the same document. This is populated from the “root” and “extension” attributes of the “setId” element of the CDA document, separated by ^ (caret). |
| EpisodeId                | int                 | The episode to which this document relates. The hospital and patient are identified via this link to the episode.  |
| DocumentTypeId           | int                 | The type of document, such as discharge summary or event summary. This is populated from the “code” element of the CDA document.   |
| ClinicalDocumentStatusId | int                 | The overall document status:<br>1: Active<br>2: Removed  |
| RemovedDate              | datetime (nullable) | If currently removed, the date and time when the document was last removed. This will be reset to null if a new version of a removed document is uploaded.   |
| RemovalReasonId          | int                 | If currently removed, the reason for removal:<br>-1: Not Removed<br>1: Withdrawn<br>2: Elect to Remove<br>3: Incorrect Identity<br>This will be reset to -1 if a new version of a removed document is uploaded.  |
| DateCreated              | datetime            | The date and time when the clinical document was first uploaded.   |
| UserCreated              | varchar(256)        | The domain and login of the user identified by the source system as responsible for the upload.  |
| DateModified             | datetime            | The date and time when the clinical document was last uploaded or removed.   |
| UserModified             | varchar(256)        | The domain and login of the user identified by the source system as responsible for the action.  |

### 2.3.3.2 ClinicalDocumentVersion Table

The “ClinicalDocumentVersion” table stores the information that relates to an individual version of the document.

| Column                    | Type            | Description  |
|---------------------------|-----------------|--|
| ClinicalDocumentVersionId | int             | Auto-incrementing primary key  |
| ClinicalDocumentId        | int             | The clinical document that this is a version of.   |
| SourceSystemDocumentId    | varchar(50)     | The source system’s unique identifier of the document instance, which will change between versions of the same document. This is populated from the “root” and “extension” attributes of the “id” element of the CDA document, separated by ^ (caret). |
| UploadedDate              | datetime        | The date and time when this version was uploaded.  |
| SupersededDate            | datetime (null) | If this version has been superseded by a later version, the date and time when this version was superseded.  |
| Package                   | varbinary(max)  | The CDA package ZIP file that was uploaded.  |
| DateCreated               | datetime        | The date and time when this clinical document version was uploaded.  |
| UserCreated               | varchar(256)    | The domain and login of the user identified by the source system as responsible for the upload.  |
| DateModified              | datetime        | The date and time when this clinical document version was uploaded or superseded.  |
| UserModified              | varchar(256)    | The domain and login of the user identified by the source system as responsible for the most recent upload or supersede action on this version.  |