



Connectivity Coordinator

Detailed Design

Version 1.0 draft — 30 June 2009

National E-Health Transition Authority Ltd

Level 25

56 Pitt Street

Sydney, NSW, 2000

Australia.

www.nehta.gov.au**Disclaimer**

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

Document Control

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

Copyright © 2009, NEHTA.

This document contains information which is protected by copyright. All Rights Reserved. No part of this work may be reproduced or used in any form or by any means—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.

Table of contents

| | |
|--|------------|
| Table of contents | iii |
| Document information | v |
| Change history | v |
| 1 Introduction | 7 |
| 1.1 Purpose | 7 |
| 1.2 Scope | 7 |
| 1.3 Intended Audience | 7 |
| 1.4 Definitions, acronyms, abbreviations | 7 |
| 1.5 Normative references | 7 |
| 1.6 Overview | 7 |
| 2 Overview | 8 |
| 2.1 Components | 8 |
| 3 Coordinator Facade | 9 |
| 3.1 Summary | 9 |
| 3.1.1 sendDocument | 10 |
| 3.1.2 queueDocument | 11 |
| 3.1.3 sendUndeliveredDocuments | 11 |
| 3.1.4 listOutboundDocuments | 12 |
| 3.1.5 getOutboundDocument | 12 |
| 3.1.6 getOutboundDocumentEvents | 12 |
| 3.1.7 removeOutboundDocument | 13 |
| 3.1.8 retrieveDocuments | 13 |
| 3.1.9 retrieveDocuments(QualifiedId) | 14 |
| 3.1.10 listInboundDocuments | 14 |
| 3.1.11 getInboundDocument | 14 |
| 3.1.12 getInboundDocumentEvents | 15 |
| 3.1.13 removeInboundDocument | 15 |
| 3.1.14 sendAck | 16 |
| 3.1.15 sendUndeliveredAcks | 16 |
| 3.1.16 listOutboundAcks | 16 |
| 3.1.17 listOutboundAckEvents | 17 |
| 3.1.18 removeOutboundAck | 17 |
| 3.1.19 retrieveAcks | 18 |
| 3.1.20 retrieveAcks(QualifiedId) | 18 |
| 3.1.21 listInboundAcks | 19 |
| 3.1.22 getInboundAck | 19 |
| 3.1.23 listInboundAckEvents | 20 |
| 3.1.24 removeInboundAck | 20 |
| 4 Datastore Facade | 21 |
| 4.1 Authorisation and Access Control | 21 |
| 4.2 Interface Summary | 21 |
| 4.3 Interface Methods | 22 |
| 4.3.1 addAck | 22 |
| 4.3.2 addAckEvent | 22 |
| 4.3.3 addDocument | 23 |
| 4.3.4 addDocumentEvent | 23 |
| 4.3.5 close | 23 |
| 4.3.6 commit | 24 |
| 4.3.7 getAck | 24 |
| 4.3.8 getDocument | 25 |
| 4.3.9 listAcks | 25 |
| 4.3.10 listAckEvents | 26 |

| | | |
|----------|--|-----------|
| 4.3.11 | listDocuments | 26 |
| 4.3.12 | listDocumentEvents..... | 27 |
| 4.3.13 | removeAck..... | 27 |
| 4.3.14 | removeDocument..... | 27 |
| 4.3.15 | rollback | 28 |
| 4.3.16 | updateAckStatus..... | 28 |
| 4.3.17 | updateDocumentStatus..... | 28 |
| 4.4 | SQL Schema | 29 |
| 4.4.1 | Document Tables | 29 |
| 4.4.2 | Acknowledgement Tables | 30 |
| 5 | Core Services Facade | 31 |
| 5.1 | Summary | 31 |
| 5.2 | Functions | 31 |
| 5.2.1 | retrieveInteractions..... | 31 |
| 5.2.2 | validateInteraction | 32 |
| 5.2.3 | getOrganisation | 32 |
| 5.2.4 | listOrganisations | 32 |
| 5.2.5 | retrieveCertificate | 32 |
| 5.2.6 | validateCertificate | 33 |
| 5.3 | Caches..... | 33 |
| 5.3.1 | Summary..... | 33 |
| 5.3.2 | Interactions cache..... | 33 |
| 5.3.3 | Certificate cache | 34 |
| 5.3.4 | Cache Table Definition | 34 |
| | Appendix A: Informative references..... | 35 |
| | Appendix B: Change log..... | 36 |

Document information

Change history

| Version | Date | Comments |
|-----------|------------|----------|
| 1.0 draft | 2009-06-30 | Draft |

This page is intentionally left blank.

1 Introduction

1.1 Purpose

The purpose of this document is to provide a detailed design for one possible approach of an implementation focused on being conformant with the NEHTA connectivity related specifications.

1.2 Scope

This document can be viewed as a complete detailed design for an implementation aiming to be compliant with the NEHTA Connectivity related specifications but **MUST NOT** be considered the only way to implement the specifications or as the suggested way to implement these specifications.

1.3 Intended Audience

The intended audience of this document include Architects and Software Developers. This document should be used to form a working understanding of the processes required to support interactions within the e-health environment, aspects of caching used to improve performance and key error handling scenarios required to be handled by an implementation.

1.4 Definitions, acronyms, abbreviations

| | |
|-------|--|
| NEHTA | National E-Health Transition Authority |
| ELS | Endpoint Locator Service |
| NASH | National Authentication Service for Health |
| UHI | Universal Health Identifier |

1.5 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For updated references, the latest edition of the referenced document (including any amendments) applies.

[RFC2119] IETF, RFC 2119: *Keywords for use in RFCs to Indicate Requirement Levels*, S. Bradner, March 1997, <http://ietf.org/rfc/rfc2119.txt>

1.6 Overview

Chapter 2 has an overview of the components

Chapter 3 covers the Coordinator Facade

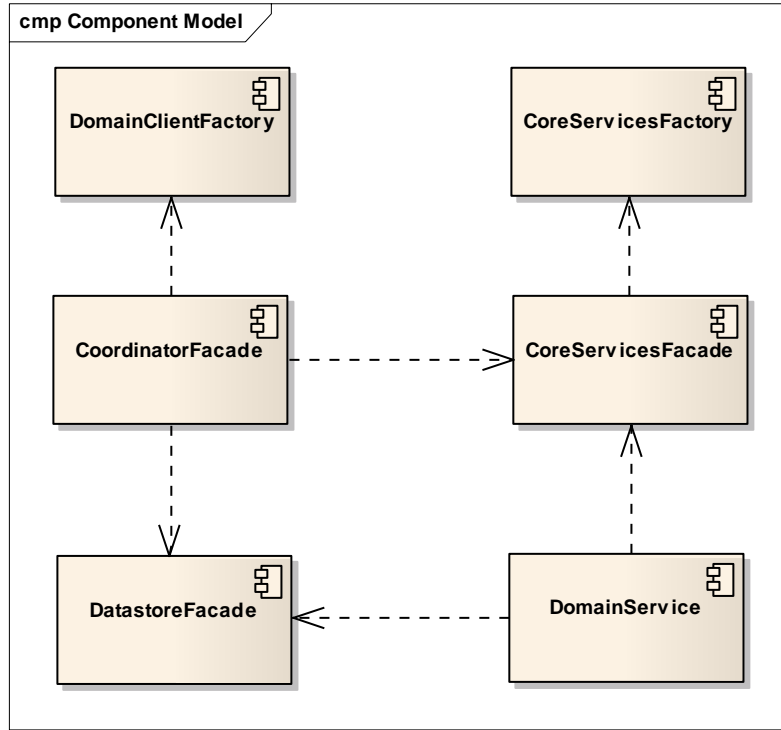
Chapter 4 covers the Datastore Facade

Chapter 5 covers the Core Services Facade

2 Overview

2.1 Components

The following diagram shows an overview of the main components within the Connectivity Coordinator and their dependencies.



The Connectivity Coordinator consists of the following main components:

| Component | Description |
|---------------------|---|
| CoordinatorFacade | Facade for sending and retrieving documents and acknowledgements. |
| DomainClientFactory | Factory for creating domain clients. Domain clients are used to access domain services. |
| DatastoreFacade | Facade for persisting and retrieving documents and acknowledgements. |
| CoreServicesFacade | Facade for accessing the core services such as UHI, ELS and NASH. |
| CoreServicesFactory | Factory for creating core service clients. Core service clients are used to access the core services. |
| DomainService | Service for a particular domain. A domain service is what a domain client invokes. |

3 Coordinator Facade

The *Coordinator Facade* provides a set of functions for sending and retrieving documents and acknowledgements. It uses the *Datastore Facade* to persist and retrieve documents and acknowledgements, and uses the *Core Services Facade* to access the core services. The *Coordinator Facade* is used as a client to access the domain services.

The functions within the *Coordinator Facade* can be grouped into roles depending on their purpose. There are four roles which relate to the patterns defined within *Concepts and Patterns for Implementing Services* [CPIS2008]:

| Role | Description |
|---|--|
| Document consumer client (Deliver) | A client that wishes to send documents to a document consumer service. Documents that are sent are persisted using the <i>Datastore Facade</i> in the document outbound folder. |
| Document supplier client (Retrieve) | A client that wishes to retrieve documents from a document supplier service. Documents that are retrieved are persisted using the <i>Datastore Facade</i> in the document inbound folder. |
| Acknowledgement consumer client (Deliver) | A client that wishes to send acknowledgements to an acknowledgement consumer service. Acknowledgements that are sent are persisted using the <i>Datastore Facade</i> in the acknowledgement outbound folder. |
| Acknowledgement supplier client (Retrieve) | A client that wishes to retrieve acknowledgements from an acknowledgement supplier service. Acknowledgements that are retrieved are persisted using the <i>Datastore Facade</i> in the acknowledgement inbound folder. |

When the *Coordinator Facade* is created a user ID must be specified. This is used to access any certificates required for invocations and to verify the metadata of documents and acknowledgements when they're sent and received.

3.1 Summary

| Function | Description |
|--------------------------|---|
| sendDocument | Sends a document to a receiver organisation. |
| queueDocument | Queues a document for sending. |
| sendUndeliveredDocuments | Resends all documents that have not been sent, including queued and failed sends. |
| listOutboundDocuments | Lists any outbound documents. |
| getOutboundDocument | Gets an outbound document. |

| | |
|--------------------------------|--|
| getOutboundDocumentEvents | Gets the events for an outbound document. |
| removeOutboundDocument | Removes an outbound document. |
| retrieveDocuments | Retrieves all documents from document suppliers. |
| retrieveDocuments(QualifiedId) | Retrieves all documents from a document supplier. |
| listInboundDocuments | Lists any inbound documents. |
| getInboundDocument | Gets an inbound document. |
| getInboundDocumentEvents | Gets the events for an inbound document. |
| removeInboundDocument | Removes an inbound document. |
| sendAck | Sends an acknowledgement to a receiver. |
| sendUndeliveredAcks | Resends all undelivered acknowledgements. |
| listOutboundAcks | Lists any outbound acknowledgements. |
| listOutboundAckEvents | Gets the events for an outbound acknowledgement. |
| removeOutboundAck | Removes an outbound acknowledgement. |
| retrieveAcks | Retrieves all acknowledgements from acknowledgement suppliers. |
| retrieveAcks(QualifiedId) | Retrieves acknowledgements from the specified provider. |
| listInboundAcks | Lists any inbound acknowledgements. |
| getInboundAck | Gets an inbound acknowledgement. |
| listInboundAckEvents | Gets the events for any inbound acknowledgements. |
| removeInboundAck | Removes an inbound acknowledgement. |

3.1.1 sendDocument

| | |
|--------|--|
| Method | DocumentEventDTO sendDocument(DocumentMetadata |
|--------|--|

| | |
|-------------|--|
| | metadata, UnsecuredPayload payload) |
| Parameters | <p>metadata</p> <p>Metadata of the document to be delivered.</p> <p>payload</p> <p>Payload of the document to be delivered.</p> |
| Returns | Event details which include the result of the delivery. |
| Exceptions | <p>IllegalArgumentException</p> <p>If the parameters are not correctly specified.</p> <p>CcException</p> <p>On any other error.</p> |
| Description | Sends a document to a receiver organisation. The document and metadata are persisted to the outbound folder. If the document was successfully delivered, its status will be "delivered", if not, the status will be "undelivered". |
| Role | Document consumer client |

3.1.2 queueDocument

| | |
|-------------|---|
| Method | void queueDocument(DocumentMetadata metadata, UnsecuredPayload payload) |
| Parameters | <p>metadata</p> <p>Metadata of the document to be delivered.</p> <p>payload</p> <p>Payload of the document to be delivered.</p> |
| Returns | none |
| Exceptions | <p>IllegalArgumentException</p> <p>If the parameters are not correctly specified.</p> <p>CcException</p> <p>On any other error.</p> |
| Description | Queues a document for sending. This places it within the outbound folder with a status of "undelivered". |
| Role | Document consumer client |

3.1.3 sendUndeliveredDocuments

| | |
|------------|---|
| Method | List<DocumentEventDTO> sendUndeliveredDocuments() |
| Parameters | None |
| Returns | List of document event details for each send that was attempted. |
| Exceptions | <p>IllegalArgumentException</p> <p>If the parameters are not correctly specified.</p> |

| | |
|-------------|--|
| | CcException On any error. |
| Description | Attempts to send any documents in the outbound folder which have a status of "undelivered". This will include documents that have been queued or documents that have failed to send initially. |
| Role | Document consumer client |

3.1.4 listOutboundDocuments

| | |
|-------------|--|
| Method | List<DocumentMetadata> listOutboundDocuments(DocumentSearchCriteria searchCriteria) |
| Parameters | searchCriteria Document search criteria. |
| Returns | List of document metadata for the documents that match the search criteria. |
| Exceptions | IllegalArgumentException If the parameters are not correctly specified. CcException On any error. |
| Description | Lists all the documents in the document outbound folder with the matching search criteria. This only returns the metadata for the documents. |
| Role | Document consumer client |

3.1.5 getOutboundDocument

| | |
|-------------|--|
| Method | DocumentDTO getOutboundDocument(DocumentId documentId) |
| Parameters | documented ID of the document to be retrieved. |
| Returns | Document with the matching document ID. |
| Exceptions | IllegalArgumentException If the parameters are not correctly specified. CcException On any error. |
| Description | Gets the full document from the outbound folder with the matching document ID. |
| Role | Document consumer client |

3.1.6 getOutboundDocumentEvents

| | |
|--------|------------------------|
| Method | List<DocumentEventDTO> |
|--------|------------------------|

| | |
|-------------|--|
| | <code>getOutboundDocumentEvents(DocumentId documentId)</code> |
| Parameters | documentId ID of the document event to be retrieved. |
| Returns | List of document event details for the matching document ID. |
| Exceptions | IllegalArgumentException If the parameters are not correctly specified. CcException On any error. |
| Description | Gets the events for an outbound document with the matching document ID. |
| Role | Document consumer client |

3.1.7 removeOutboundDocument

| | |
|-------------|--|
| Method | RemoveResultCode <code>removeOutboundDocument(DocumentId documentId)</code> |
| Parameters | documentId ID of the document to be removed. |
| Returns | Result code from the removal. |
| Exceptions | IllegalArgumentException If the parameters are not correctly specified. CcException On any error. |
| Description | Removes a document from the outbound folder with the matching document ID. |
| Role | Document consumer client |

3.1.8 retrieveDocuments

| | |
|-------------|--|
| Method | <code>List<DocumentMetadata> retrieveDocuments()</code> |
| Parameters | None |
| Returns | List of document metadata from the retrieved documents. |
| Exceptions | IllegalArgumentException If the parameters are not correctly specified. CcException On any error. |
| Description | Retrieves a list of documents from a supplier and stores them within the input folder with a status of "retrieved". This done by doing of reverse lookup of the ELS for interactions that support document delivery. The provider ID within the interactions is used to do another ELS |

| | |
|------|---------------------------------------|
| | lookup to find any supplier services. |
| Role | Document supplier client |

3.1.9 retrieveDocuments(QualifiedId)

| | |
|-------------|--|
| Method | List<DocumentMetadata> retrieveDocuments(QualifiedId serviceTarget) |
| Parameters | serviceTarget Service target to retrieve documents from. |
| Returns | List of document metadata. |
| Exceptions | IllegalArgumentException If the parameters are not correctly specified. CcException On any error. |
| Description | Retrieves a list of documents from a supplier with the specified ID into the inbound folder. The service target parameter is used to lookup the ELS for a document supplier service. |
| Role | Document supplier client |

3.1.10 listInboundDocuments

| | |
|-------------|---|
| Method | List<DocumentMetadata> listInboundDocuments(DocumentSearchCriteria searchCriteria) |
| Parameters | searchCriteria Criteria to use when searching for an inbound document. |
| Returns | List of document metadata for the matching search criteria. |
| Exceptions | IllegalArgumentException If the parameters are not correctly specified. CcException On any error. |
| Description | Retrieves a list of inbound documents that match the specified search criteria. This only returns the metadata and not the full document. |
| Role | Document supplier client |

3.1.11 getInboundDocument

| | |
|------------|---|
| Method | DocumentDTO getInboundDocument(DocumentId documentId) |
| Parameters | documentId |

| | |
|-------------|---|
| | Document ID of the inbound document to get. |
| Returns | Document with the matching ID. |
| Exceptions | <p>IllegalArgumentException</p> <p>If the parameters are not correctly specified.</p> <p>CcException</p> <p>On any error.</p> |
| Description | Gets a document from the inbound folder with the matching ID. |
| Role | Document supplier client |

3.1.12 getInboundDocumentEvents

| | |
|-------------|---|
| Method | <p>DocumentEventDTO</p> <p>getInboundDocumentEvents(DocumentId documentId)</p> |
| Parameters | <p>documentId</p> <p>Document ID of the events to get.</p> |
| Returns | List of document events for the matching document ID. |
| Exceptions | <p>IllegalArgumentException</p> <p>If the parameters are not correctly specified.</p> <p>CcException</p> <p>On any error.</p> |
| Description | Gets the events for a document in the inbound folder with the matching document ID. |
| Role | Document supplier client |

3.1.13 removeInboundDocument

| | |
|-------------|---|
| Method | <p>RemoveResultCode</p> <p>removeInboundDocument(DocumentId documentId)</p> |
| Parameters | <p>documentId</p> <p>Document ID of the document to remove.</p> |
| Returns | Status of the removal. |
| Exceptions | <p>IllegalArgumentException</p> <p>If the parameters are not correctly specified.</p> <p>CcException</p> <p>On any error.</p> |
| Description | Removes a document from the inbound folder with the matching document ID. |
| Role | Document supplier client |

3.1.14 sendAck

| | |
|-------------|--|
| Method | AckEventDTO sendAck(AckMetadata metadata, UnsecuredPayload payload) |
| Parameters | metadata Metadata of the acknowledgement payload Payload of the acknowledgement |
| Returns | Acknowledgement event details. |
| Exceptions | IllegalArgumentException If the parameters are not correctly specified. CcException On any error. |
| Description | Sends an acknowledgement to a receiver organisation. The acknowledgement and metadata are persisted to the outbound folder. If the acknowledgement was successfully delivered, its status will be "delivered", if not, the status will be "undelivered". |
| Role | Acknowledgement consumer client |

3.1.15 sendUndeliveredAcks

| | |
|-------------|---|
| Method | List<AckEventDTO> sendUndeliveredAcks() |
| Parameters | none |
| Returns | List of acknowledgement event details. |
| Exceptions | IllegalArgumentException If the parameters are not correctly specified. CcException On any error. |
| Description | Attempts to send any acknowledgements in the outbound folder which have a status of "undelivered". This will include acknowledgements that have failed to send initially. |
| Role | Acknowledgement consumer client |

3.1.16 listOutboundAcks

| | |
|--------|--|
| Method | List<AckMetadata> listOutboundAcks(AckSearchCriteria |
|--------|--|

| | |
|-------------|---|
| | searchCriteria) |
| Parameters | searchCriteria Outbound acknowledgement search criteria |
| Returns | List of acknowledgement metadata for the search criteria. |
| Exceptions | IllegalArgumentException If the parameters are not correctly specified. CcException On any error. |
| Description | Returns a list of outbound acknowledgements that match the search criteria. This returns only the metadata for the acknowledgement. |
| Role | Acknowledgement consumer client |

3.1.17 listOutboundAckEvents

| | |
|-------------|--|
| Method | List<AckEventDTO> listOutboundAckEvents(DocumentId documentId) |
| Parameters | documentId Document ID of the acknowledgement events to retrieve. |
| Returns | List of acknowledgement events. |
| Exceptions | IllegalArgumentException If the parameters are not correctly specified. CcException On any error. |
| Description | Returns a list of acknowledgement events from the outbound folder that match the document ID. |
| Role | Acknowledgement consumer client |

3.1.18 removeOutboundAck

| | |
|------------|--|
| Method | RemoveResultCode removeOutboundAck(DocumentId documentId) |
| Parameters | documentId Document ID of the outbound acknowledgement to remove. |

| | |
|-------------|---|
| Returns | Removal result code. |
| Exceptions | <p>IllegalArgumentException If the parameters are not correctly specified.</p> <p>CcException On any error.</p> |
| Description | Removes an acknowledgement from the outbound folder that matches the document ID. |
| Role | Acknowledgement consumer client |

3.1.19 retrieveAcks

| | |
|-------------|--|
| Method | List<AckMetadata> retrieveAcks() |
| Parameters | None |
| Returns | List of acknowledgement metadata. |
| Exceptions | <p>IllegalArgumentException If the parameters are not correctly specified.</p> <p>CcException On any error.</p> |
| Description | Retrieves a list of acknowledgements from a supplier and stores them within the input folder with a status of "retrieved". This done by doing of reverse lookup of the ELS for interactions that support acknowledgements delivery. The provider ID within the interactions is used to do another ELS lookup to find any supplier services. The method returns a list of metadata of retrieved acknowledgements. |
| Role | Acknowledgement supplier client |

3.1.20 retrieveAcks(QualifiedId)

| | |
|------------|---|
| Method | List<AckMetadata> retrieveAcks(QualifiedId supplierOrganisation) |
| Parameters | <p>supplierOrganisation Organisation that has the acknowledgement supplier service.</p> |
| Returns | List of acknowledgement metadata. |
| Exceptions | <p>IllegalArgumentException If the parameters are not correctly specified.</p> |

| | |
|-------------|---|
| | CcException On any error. |
| Description | Retrieves a list of acknowledgements from a supplier with the specified ID into the inbound folder. The service target parameter is used to lookup the ELS for an acknowledgement supplier service. |
| Role | Acknowledgement supplier client |

3.1.21 listInboundAcks

| | |
|-------------|--|
| Method | List<AckMetadata> listInboundAcks(AckSearchCriteria searchCriteria) |
| Parameters | searchCriteria Inbound acknowledgement search criteria. |
| Returns | List of acknowledgements from the inbound folder that match the search criteria. |
| Exceptions | IllegalArgumentException If the parameters are not correctly specified. CcException On any error. |
| Description | Retrieves a list of acknowledgements from the inbound folder that match the search criteria. |
| Role | Acknowledgement supplier client |

3.1.22 getInboundAck

| | |
|-------------|--|
| Method | AckDTO getInboundAck(DocumentId documentId) |
| Parameters | documentId Document ID of the inbound acknowledgement to retrieve. |
| Returns | Acknowledgement from the inbound folder with the matching document ID. |
| Exceptions | IllegalArgumentException If the parameters are not correctly specified. CcException On any error. |
| Description | Retrieves an acknowledgement from the inbound folder with the matching document ID. |

| | |
|------|---------------------------------|
| Role | Acknowledgement supplier client |
|------|---------------------------------|

3.1.23 listInboundAckEvents

| | |
|-------------|--|
| Method | List<AckDTO> listInboundAckEvents(DocumentId documentId) |
| Parameters | documentId Document ID of the inbound acknowledgement events to retrieve. |
| Returns | List of inbound acknowledgement events. |
| Exceptions | IllegalArgumentException If the parameters are not correctly specified. CcException On any error. |
| Description | Retrieves the acknowledgement inbound events with the matching document ID. |
| Role | Acknowledgement supplier client |

3.1.24 removeInboundAck

| | |
|-------------|--|
| Method | RemoveResultCode removeInboundAck(DocumentId documentId) |
| Parameters | documentId Document ID of the inbound acknowledgement to remove. |
| Returns | Removal status. |
| Exceptions | IllegalArgumentException If the parameters are not correctly specified. CcException On any error. |
| Description | Removes an acknowledgement from the inbound folder with the matching document ID. |
| Role | Acknowledgement supplier client |

4 Datastore Facade

The *DataStoreFacadeInterface* defines an interface for a local datastore whose purpose is to persist documents, acknowledgements and events associated with the delivery/receipt of documents and acknowledgements.

It is anticipated that the implementation of the persistent datastore will vary between vendors. The implementation included with this reference project involves the use of a SQL database for a persistent datastore. Other implementations could use the file system (for example to interface with legacy GP desktop software).

4.1 Authorisation and Access Control

It is assumed that any access control policy has already been applied and that the user has the requisite permissions to access/modify the data specified by the parameters to the operations.

4.2 Interface Summary

| Function | Description |
|------------------|--|
| addAck | Add an acknowledgement to the persistent datastore. |
| addAckEvent | Add an acknowledgement event to the persistent datastore. |
| addDocument | Add a document to the persistent datastore. |
| addDocumentEvent | Add a document event to the persistent datastore. |
| close | Close the persistent datastore. Once closed, an instance of the <i>DataStoreFacadeInterface</i> cannot be re-used. |
| commit | Commit any changes made to the persistent datastore since the last commit/rollback. This will also release any locks placed on entities. |
| getAck | Fetch the details of the acknowledgement record from the persistent datastore. |
| getDocument | Fetch the details of the document record from the persistent datastore. |
| listAcks | Returns a list of acknowledgement metadata that matches the search criteria. |
| listAckEvents | Returns a list of events that have been performed on the given acknowledgement. |
| listDocuments | Returns a list of document metadata that matches the search criteria. |

| | |
|----------------------|--|
| listDocumentEvents | Returns a list of events that have been performed on the given document. |
| removeAck | Remove (delete) the acknowledgement from the persistent data store. |
| removeDocument | Remove (delete) the document from the persistent data store. |
| rollback | Rollback any changes made to the persistent datastore since the last commit/rollback. This will also release any locks placed on entities. |
| updateAckStatus | Update the status of an acknowledgement. |
| updateDocumentStatus | Update the status of a document. |

4.3 Interface Methods

4.3.1 addAck

| | |
|-------------|--|
| Method | int addAck(AckDTO ack) |
| Parameters | ack The acknowledgement record to be persisted. |
| Returns | The primary key allocated to the persisted record. |
| Throws | IllegalArgumentException If the parameters are not correctly specified. This may include mandatory fields and violations of the length of fields. CcException On any other error. |
| Description | Add the ack to the persistent datastore. |

4.3.2 addAckEvent

| | |
|------------|--|
| Method | void addAckEvent(AckEventDTO event) |
| Parameters | event The acknowledgement event record to be persisted. |
| Returns | |
| Throws | IllegalArgumentException If the parameters are not correctly specified. This may include mandatory fields and violations of the length of |

| | |
|-------------|---|
| | fields. CcException On any other error. |
| Description | Add the event to the persistent datastore. |

4.3.3 addDocument

| | |
|-------------|--|
| Method | int addDocument(DocumentDTO document) |
| Parameters | document The document record to be persisted. |
| Returns | The primary key allocated to the persisted record. |
| Throws | IllegalArgumentException If the parameters are not correctly specified. This may include mandatory fields and violations of the length of fields. CcException On any other error. |
| Description | Add the document to the persistent datastore. |

4.3.4 addDocumentEvent

| | |
|-------------|--|
| Method | void addDocumentEvent(DocumentEventDTO event) |
| Parameters | event The document event record to be persisted. |
| Returns | |
| Throws | IllegalArgumentException If the parameters are not correctly specified. This may include mandatory fields and violations of the length of fields. CcException On any other error. |
| Description | Add the event to the persistent datastore. |

4.3.5 close

| | |
|--------|--------------|
| Method | void close() |
|--------|--------------|

| | |
|-------------|--|
| Parameters | |
| Returns | |
| Throws | |
| Description | Close the persistent datastore. Once closed, an instance of the <code>DataStoreFacadeInterface</code> cannot be re-used. |

4.3.6 **commit**

| | |
|-------------|--|
| Method | <code>void commit()</code> |
| Parameters | |
| Returns | |
| Throws | <code>CcException</code> On any error. |
| Description | Commit any changes made to the persistent datastore since the last commit/rollback. This will also release any locks placed on entities. |

4.3.7 **getAck**

| | |
|------------|---|
| Method | <code>AckDTO getAck(FolderTypeCode folder, DocumentId documentId, Lock lockType)</code> |
| Parameters | <p><code>folder</code> Specifies the folder - either <code>INPUT</code> or <code>OUTPUT</code>.</p> <p><code>documentId</code> The unique (external) identifier of the acknowledgement.</p> <p><code>lockType</code> Specifies whether a lock should be placed on the retrieved record in the persistent datastore. Any lock will remain until either <i>commit</i> or <i>rollback</i> is called.</p> |
| Returns | If found, the details of the fetched acknowledgment. If not found, returns <i>null</i> . |
| Throws | <p><code>IllegalArgumentException</code> If the parameters are not correctly specified. This may include mandatory fields and violations of the length of fields.</p> <p><code>CcException</code> On any other error.</p> |

| | |
|-------------|--|
| Description | Fetch the details of the acknowledgement record from the persistent datastore. |
|-------------|--|

4.3.8 getDocument

| | |
|-------------|--|
| Method | DocumentDTO getDocument(FolderTypeCode folder, DocumentId documentId, Lock lockType) |
| Parameters | <p>folder</p> <p>Specifies the folder - either INPUT or OUTPUT.</p> <p>documentId</p> <p>The unique (external) identifier of the document.</p> <p>lockType</p> <p>Specifies whether a lock should be placed on the retrieved record in the persistent datastore. Any lock will remain until either <i>commit</i> or <i>rollback</i> is called.</p> |
| Returns | If found, the details of the fetched document. If not found, returns <i>null</i> . |
| Throws | <p>IllegalArgumentException</p> <p>If the parameters are not correctly specified. This may include mandatory fields and violations of the length of fields.</p> <p>CcException</p> <p>On any other error.</p> |
| Description | Fetch the details of the document record from the persistent datastore. |

4.3.9 listAcks

| | |
|------------|---|
| Method | List<AckMetadata> listAcks(FolderTypeCode folder, AckSearchCriteria searchCriteria) |
| Parameters | <p>folder</p> <p>Specifies the folder - either INPUT or OUTPUT.</p> <p>searchCriteria</p> <p>Criteria used to filter the search.</p> |
| Returns | A list, which may be empty but is never null, of the metadata of matching acknowledgement records. |
| Throws | <p>IllegalArgumentException</p> <p>If the parameters are not correctly specified. This may include mandatory fields and violations of the length of fields.</p> |

| | |
|-------------|--|
| | CcException On any other error. |
| Description | Returns a list of acknowledgement metadata that matches the search criteria. |

4.3.10 listAckEvents

| | |
|-------------|--|
| Method | List<AckEventDTO> listAckEvents(int ackPk) |
| Parameters | ackPk The primary key of a particular acknowledgement. |
| Returns | A list, which may be empty but is never null, of the events that have been performed on the given acknowledgement. |
| Throws | CcException On any error. |
| Description | Returns a list of events that have been performed on the given acknowledgement. |

4.3.11 listDocuments

| | |
|-------------|--|
| Method | List<DocumentMetadata> listDocuments(FolderTypeCode folder, DocumentSearchCriteria searchCriteria) |
| Parameters | folder Specifies the folder - either INPUT or OUTPUT. searchCriteria Criteria used to filter the search. |
| Returns | A list, which may be empty but is never null, of the metadata of matching document records. |
| Throws | IllegalArgumentException If the parameters are not correctly specified. This may include mandatory fields and violations of the length of fields. CcException On any other error. |
| Description | Returns a list of document metadata that matches the search criteria. |

4.3.12 listDocumentEvents

| | |
|-------------|---|
| Method | List<DocumentEventDTO> listDocumentEvents(int documentPk) |
| Parameters | documentPk The primary key of a particular document. |
| Returns | A list, which may be empty but is never null, of the events that have been performed on the given document. |
| Throws | CcException On any error. |
| Description | Returns a list of events that have been performed on the given document. |

4.3.13 removeAck

| | |
|-------------|---|
| Method | void removeAck(int ackPk) |
| Parameters | ackPk The primary key of a particular acknowledgement. |
| Returns | |
| Throws | CcException On any error. |
| Description | Remove (delete) the acknowledgement from the persistent data store. |

4.3.14 removeDocument

| | |
|-------------|--|
| Method | void removeDocument(int documentPk) |
| Parameters | documentPk The primary key of a particular document. |
| Returns | |
| Throws | CcException On any error. |
| Description | Remove (delete) the document from the persistent data store. |

4.3.15 rollback

| | |
|-------------|--|
| Method | void rollback() |
| Parameters | |
| Returns | |
| Throws | |
| Description | Rollback any changes made to the persistent datastore since the last commit/rollback. This will also release any locks placed on entities. |

4.3.16 updateAckStatus

| | |
|-------------|---|
| Method | void updateAckStatus(int ackPk, AckStatusCode status) |
| Parameters | ackPk The primary key of a particular acknowledgement. status The new status of the acknowledgement. |
| Returns | |
| Throws | CcException On any error. |
| Description | Update the status of the acknowledgement. |

4.3.17 updateDocumentStatus

| | |
|-------------|--|
| Method | void updateDocumentStatus(int documentPk, DocumentStatusCode status) |
| Parameters | documentPk The primary key of a particular document. status The new status of the document. |
| Returns | |
| Throws | CcException On any error. |
| Description | Update the status of the document. |

4.4 SQL Schema

The following SQL schema was used in the reference project for the implementation of a SQL datastore for persisting documents, acknowledgements and event:

4.4.1 Document Tables

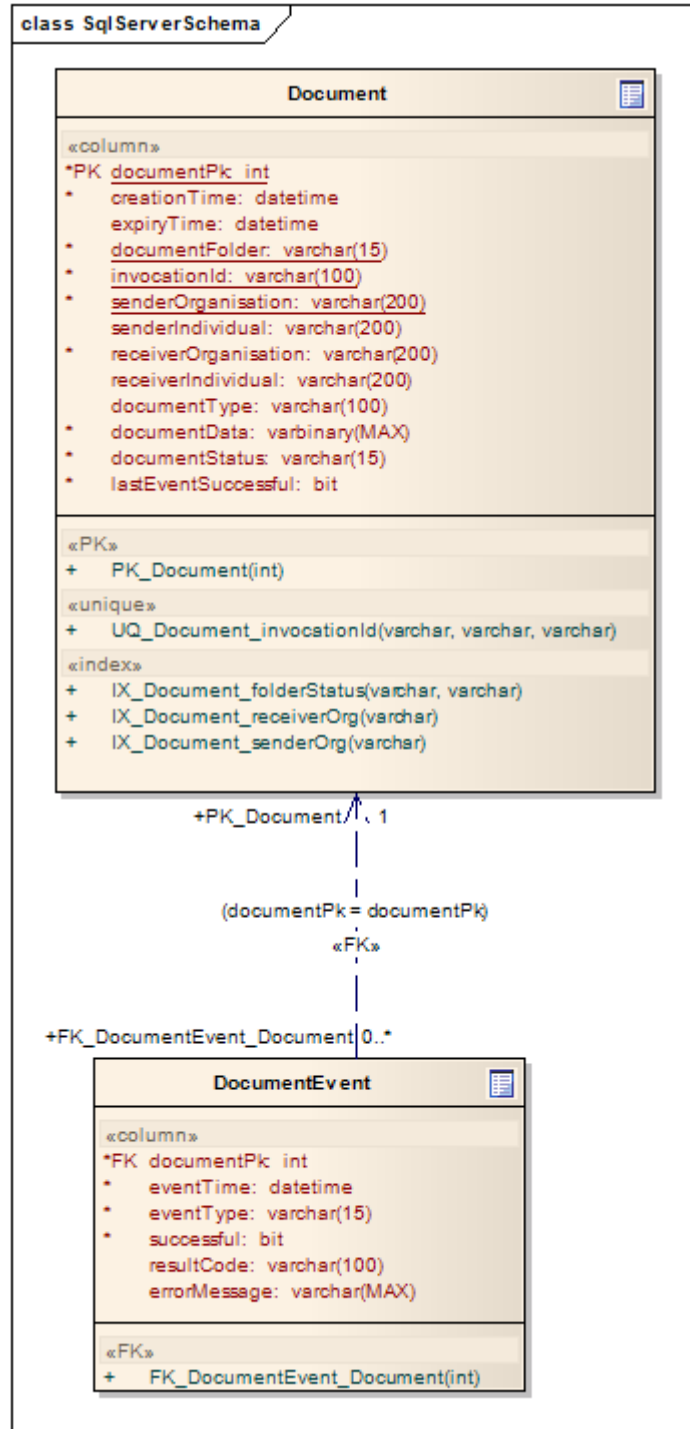


Figure 1: Document and DocumentEvent tables

4.4.2 Acknowledgement Tables

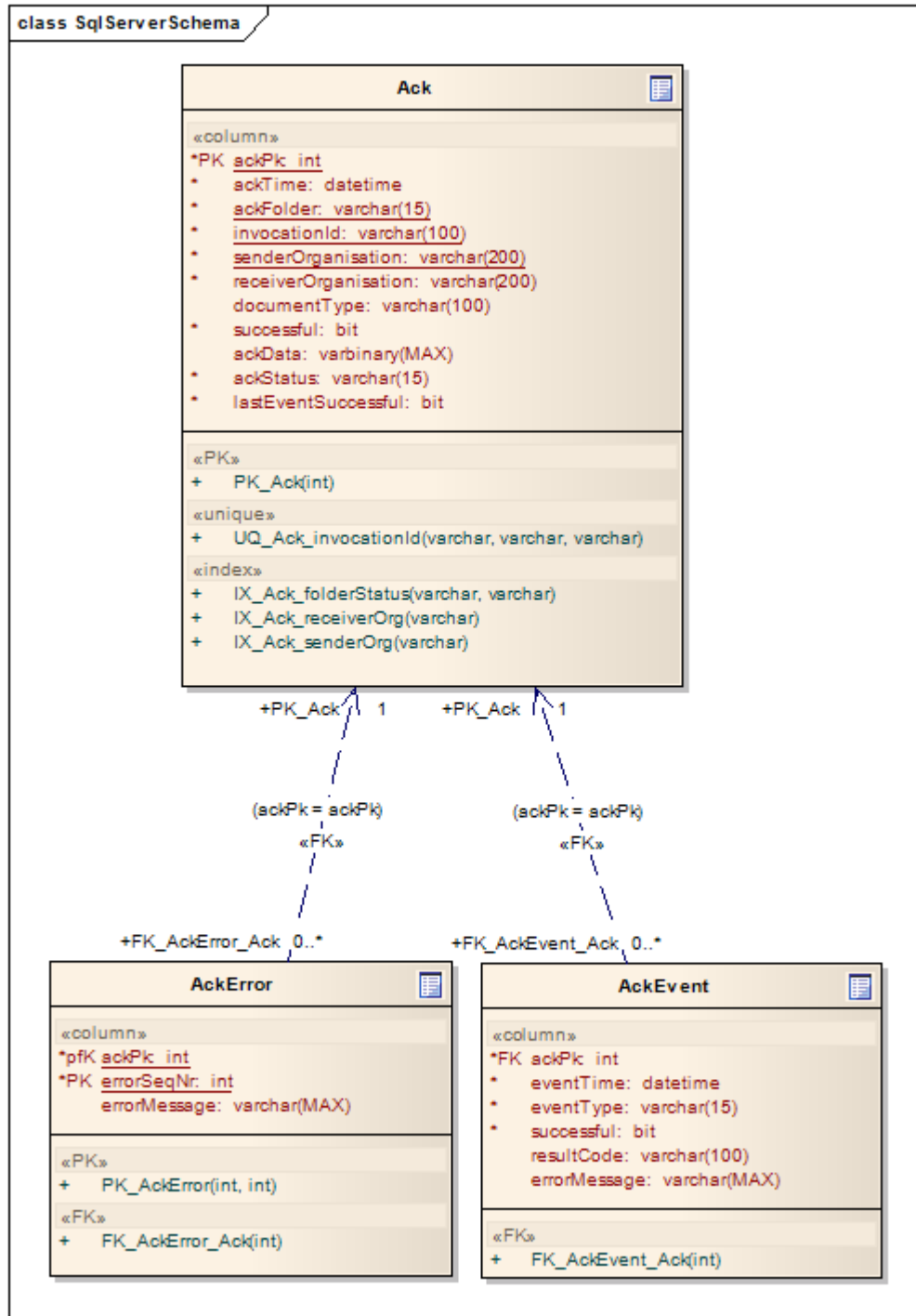


Figure 2: Ack, AckError and AckEvent tables

5 Core Services Facade

The `CoreServicesFacade` class defines a façade for accessing the core services. The core services consist of the UHI, ELS and NASH services. Interactions and certificates are stored in a cache to prevent repeated access to the ELS and NASH services.

5.1 Summary

| Function | Description |
|-----------------------------------|--|
| <code>retrieveInteractions</code> | Retrieves a set of interactions for a provider using the core services. This involves using UHI to find the ELS address and using the ELS to get the interactions. |
| <code>validateInteraction</code> | Validates an interaction to ensure it's still valid. This involves using UHI to find the ELS address and using ELS to check the interaction exists. |
| <code>getOrganisation</code> | Retrieves an organisation from the UHI service. |
| <code>listOrganisations</code> | Retrieves a list of organisations from the UHI service. |
| <code>retrieveCertificate</code> | Retrieves a certificate from the NASH service. |
| <code>validateCertificate</code> | Validates a certificate. This includes checking if it's revoked. |

5.2 Functions

5.2.1 `retrieveInteractions`

| | |
|------------|--|
| Method | <code>List<InteractionDTO> retrieveInteractions(QualifiedId serviceTarget, String serviceCategory, List<String> interfaces)</code> |
| Parameters | <p><code>serviceTarget</code> ID of the provider who has the interaction</p> <p><code>serviceCategory</code> category of the interaction being requested</p> <p><code>interfaces</code> List of interfaces of services being requested</p> |
| Returns | List of interactions for the service target. |
| Exceptions | <p><code>CcException</code> thrown when a system level error occurs</p> |

5.2.2 validateInteraction

| | |
|------------|--|
| Method | boolean validateInteraction(InteractionDTO interaction) |
| Parameters | interaction Interaction to validate |
| Returns | true when the interaction is valid otherwise returns false |
| Exceptions | CcException thrown when a system level error occurs |

5.2.3 getOrganisation

| | |
|------------|---|
| Method | OrganisationDTO getOrganisation(QualifiedId organisationId) |
| Parameters | organisationId ID of the organisation |
| Returns | Organisation details |
| Exceptions | CcException thrown when a system level error occurs |

5.2.4 listOrganisations

| | |
|------------|---|
| Method | List<OrganisationDTO> listOrganisations(OrganisationSearchCriteria searchCriteria) |
| Parameters | searchCriteria search criteria for the organisation |
| Returns | List of organisations that match the criteria |
| Exceptions | CcException thrown when a system level error occurs |

5.2.5 retrieveCertificate

| | |
|--------|---|
| Method | X509Certificate retrieveCertificate(QualifiedCertificateRef qualifiedCertRef) |
|--------|---|

| | |
|------------|--|
| Parameters | qualifiedCertRef qualified certificate reference |
| Returns | X509 certificate matching the certificate reference |
| Exceptions | CcException thrown when a system level error occurs |

5.2.6 validateCertificate

| | |
|------------|---|
| Method | boolean validateCertificate(X509Certificate certificate) |
| Parameters | certificate certificate to validate |
| Returns | true if the certificate is valid otherwise it returns false |
| Exceptions | CcException thrown when a system level error occurs |

5.3 Caches

Caches are used to prevent access to core services whenever possible since the overhead of invoking a Web service can be high.

Cache entries are persisted within a database.

5.3.1 Summary

| Name | Description |
|--------------------|--|
| Interactions cache | Caches interactions that are retrieved from an ELS service. When interactions are retrieved from an ELS service, they are put into the cache. The cache is checked first before an ELS service is invoked. |
| Certificate cache | Caches certificates retrieved from the NASH service. When a certificate is retrieved from the NASH service it is put into the cache. The cache is checked before the NASH service is invoked. |

5.3.2 Interactions cache

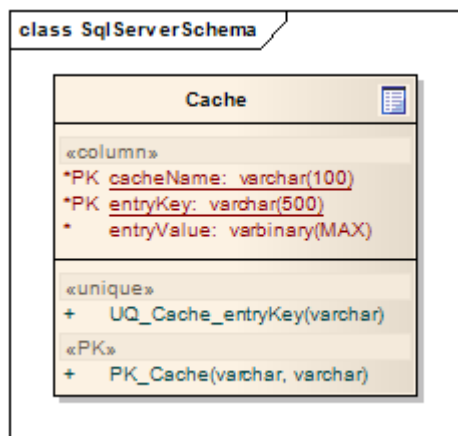
| Function | Description |
|----------|---|
| Adding | Interactions are added to the cache after they have successfully been retrieved from an ELS services. This is done when the retrieveInteractions method is invoked. |

| | |
|----------|---|
| Eviction | Interactions can be evicted from the cache when the <code>validateInteraction</code> method is invoked. When an interaction is invalid, it is removed from the cache. |
|----------|---|

5.3.3 Certificate cache

| Function | Description |
|----------|---|
| Adding | Certificates are added to the cache after they have successfully been retrieved from the NASH service. This is done when the <code>getCertificate</code> method is invoked. |
| Eviction | Certificates can be evicted from the cache when the <code>validateCertificate</code> method is invoked. When a certificate is invalid, it is removed from the cache. A certificate is invalid when its trust chain is invalid, it's expired or it has been revoked. |

5.3.4 Cache Table Definition



Appendix A: Informative references

[CPIS2008] NEHTA, *Concepts and Patterns for Implementing Services 2.0*, 1 December 2008.

Appendix B: Change log

Version 1.0

- Initial release