

nehta

Connectivity

Architecture

Version 1.0 — 1 December 2008

Release

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 © 2008, 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 Executive overview	7
2 Preface	9
2.1 Document purpose	9
2.2 Intended audience	9
2.3 Document map	9
2.4 Definitions, acronyms, abbreviations	10
2.5 Normative references	10
2.6 Overview	10
3 Introduction	11
3.1 Overview	11
3.1.1 Connectivity	11
3.1.2 Connectivity architecture	11
3.2 Requirements	12
3.3 Assumptions and dependencies	12
3.3.1 Provider organisation focus	12
3.3.2 Service orientation	13
3.3.3 Specific business functionality	13
3.3.4 Public Key Infrastructure (PKI)	14
4 Business viewpoint	15
4.1 Interactions	15
4.1.1 Bootstrapping interaction	16
4.1.2 Identification interaction	17
4.1.3 Core connectivity interaction	18
4.2 Services and roles	20
4.2.1 Business services	20
4.2.2 Identification services	21
4.2.3 PKI services	21
4.2.4 Directory services	22
4.3 Access control	23
5 Information viewpoint	24
5.1 Semantic information model	24
5.2 Data types	24
5.2.1 Identifiers	24
5.2.2 Certificates	24
5.3 Terminology	25
5.3.1 Service identifiers	25
5.3.2 Content terminology	25
6 Technology viewpoint	26
6.1 Services technology	26
6.1.1 Web services	26
6.2 Patterns for services	27
6.3 Security	27
6.3.1 XML Secured Payload	27
6.3.2 Checking digital signatures	28
6.4 Reliability	29
6.4.1 Business acknowledgements	29

6.4.2	Operation acknowledgements	29
6.4.3	Network acknowledgements	29
6.5	Service interface specifications	30
6.5.1	IHI services.....	30
6.5.2	HPI services	30
6.5.3	NASH services	30
6.5.4	Service Instance Locator	30
Appendix A: Informative references		31

Document information

Change history

Version	Date	Comments
1.0	2008-12-01	Release

This page is intentionally left blank.

1 Executive overview

Connectivity is establishing and conducting communications in a reliable and secure manner.

This document describes an architecture for achieving connectivity that supports:

- Structured application-to-application communications between computer systems;
- Communications with external provider organisations; and
- Addressing the needs of specific healthcare processes.

This architecture focuses on addressing the communication needs of healthcare provider organisations. It does not directly address the communication needs of individuals: individuals will interact with services via healthcare providers. It also currently does not address providers operating services: they will have to operate services from an organisation.

This architecture describes how structured electronic communications will work in the national e-health environment. The architecture consists of:

- The concept of services in the national e-health environment.

This architecture views the business environment as a set of services. Services are developed to address specific business functionality. The focus is on the specific business services.

In the e-health environment, there will be a distributed set of services. These services are invoked securely over the Internet. These services can be considered as being of two types: infrastructure services and business specific services.

- The actual infrastructure services which will be available.

The infrastructure services are those services that provide common functionality that is needed by many different business processes. They are not specific to any particular healthcare business process. They are essential for providing connectivity, and NEHTA will be implementing and deploying some of them on a national basis. These services are:

- Identification services for providing identifiers for entities in the healthcare environment (e.g. providers and individuals);
- PKI services for issuing and managing Public Key Infrastructure (PKI) certificates; and
- Service Instance Locators (SIL) to allow applications to obtain the information needed to use other services.

The business specific services will be defined to support specific healthcare business needs. For example, there will be services to support communication of pathology, discharge summary, medication and referral data.

- The interactions of those services: how they are used together to achieve connectivity.

Three interactions have been identified. They consist of the bootstrapping process to obtain information about the national infrastructure services, the identification interaction to identify entities, and the core connectivity interaction to invoke a business specific operation.

The core connectivity process is the main interaction that is used. It consists of:

- a. Identifying the entities involved by using the identification services (one of these entities will be the target organisation that communications is being established to);
- b. Determining which service instance to invoke using the Service Instance Locator (SIL) for the target entity;
- c. Obtaining the PKI public certificates needed to secure the communications by using the PKI service; and
- d. Invoking the business specific service instance to perform the desired operation.

In summary, this architecture describes how structure electronic communications will work in the national e-health environment. This architecture defines the services approach, the national infrastructure services, and the interactions using them to achieve reliable and secure communications.

2 Preface

2.1 Document purpose

This document describes the architecture for achieving connectivity in the national e-health environment.

2.2 Intended audience

This is a technical document.

For a non-technical introduction to this connectivity architecture, please see the *Connectivity: Introductory Guide* [CIG2008]. That document contains examples of the connectivity interactions, which can be easier to understand than the more formal interactions described in this document. It is suggested that new readers should read the *Connectivity: Introductory Guide* first.

This document is intended for:

- Enterprise architects who develop strategic plans for solutions that need to establish connectivity with other systems.
- Solution architects who develop solutions that implement or interact with other systems.
- Software developers who implement the solutions designed by the solution architects.

The reader is expected to understand the concepts of Service-Oriented Architecture and Public Key Infrastructure (PKI).

2.3 Document map

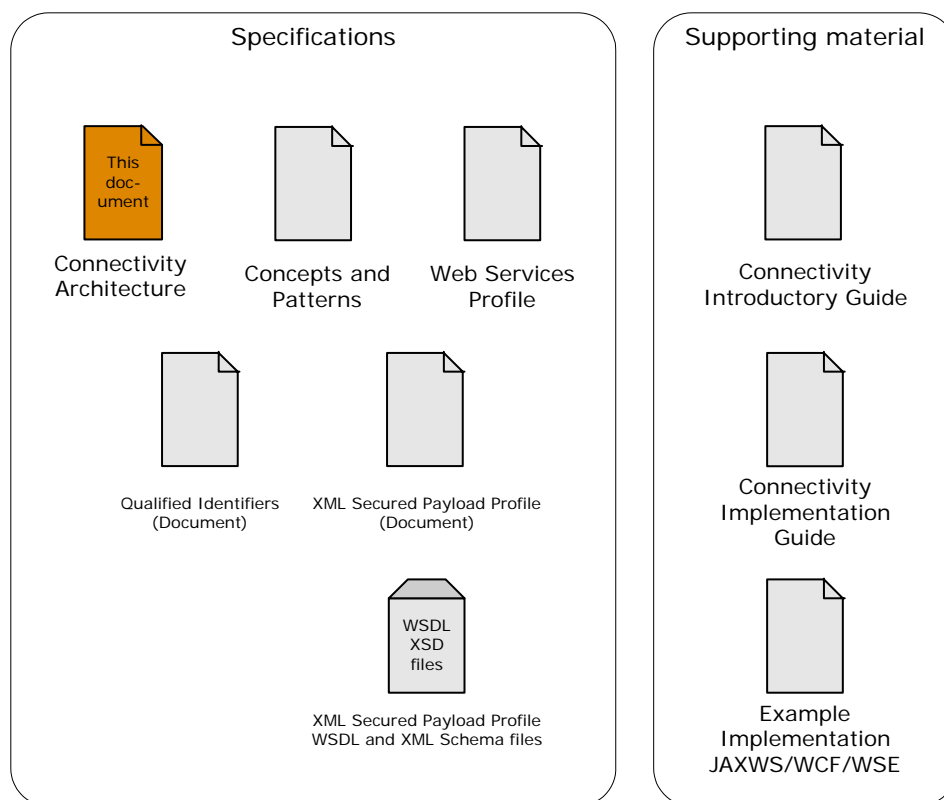


Figure 1: Connectivity document map

2.4 Definitions, acronyms, abbreviations

HPI	Healthcare Provider Identifier
HPI-I	Healthcare Provider Identifier for Individuals
HPI-O	Healthcare Provider Identifier for Organisations
IHI	Individual Healthcare Identifier
NASH	National Authentication Service for Health
NEHTA	National E-Health Transition Authority
PKI	Public Key Infrastructure
SIL	Service Instance Locator

2.5 Normative references

The following NEHTA specifications and other references contain provisions which, through reference in this text, constitute provisions of this Specification. At the time of publication, the editions indicated were valid. All Specification and other references are subject to revision: all users of this Specification are therefore encouraged to investigate the possibility of applying the most recent edition of the Specification and other references listed below.

[CPIS2008]	NEHTA, <i>Concepts and Patterns for Implementing Services v2.0</i> , 1 December 2008.
[CIG2008]	NEHTA, <i>Connectivity: Introductory Guide v1.0</i> , 1 December 2008.
[QI2008]	NEHTA, <i>Qualified Identifiers v1.0</i> , 1 December 2008.
[XSP2008]	NEHTA, <i>XML Secured Payload v1.0</i> , 1 December 2008.

Additional references can be found in Appendix A: "Informative references."

2.6 Overview

Chapter 3 provides an introduction to connectivity.

The rest of this document follows the structure recommended by the *Interoperability Framework* [NIF2007]. The architecture is described from three different perspectives: business, information and technology.

Chapter 4 describes connectivity from the business viewpoint.

Chapter 5 describes connectivity from the information viewpoint.

Chapter 6 describes connectivity from the technology viewpoint.

3 Introduction

This chapter provides an overview of the connectivity architecture and the assumptions and dependencies it is built on.

3.1 Overview

3.1.1 Connectivity

Connectivity is establishing and conducting communications in a reliable and secure manner.

In the e-health environment there is the need for connectivity with different provider organisations. For example, communications can occur with hospitals, pathology laboratories, specialist clinics and general practice.

This document describes the architecture for achieving connectivity for structured application-to-application electronic communication with external provider organisations to support healthcare business processes.

There are three key aspects:

- It is about structured application-to-application electronic communications between computer systems. This involves having shared terminologies and clinical data structures between the participants. The connectivity architecture is concerned with the exchange of terminologies and clinical data, but does not define them. They will be defined by other specifications outside of this architecture.

Unstructured communications is also important in healthcare, but it is not addressed by this architecture. Unstructured communications often occurs between people in an ad hoc manner. For example, it includes telephone and email communications. These are outside the scope of this architecture.

- It is about communications between different entities. The focus is on communications with external provider organisations.

Communication inside an organisation is also important in healthcare, but is outside the scope of this architecture. Although aspects of this architecture can be used to implement systems inside an organisation, that use will not be covered by this architecture.

- It supports a healthcare business process. This means the communications supports a specific health process. It involves data for a particular purpose and specific rules for how that data is processed.

3.1.2 Connectivity architecture

This architecture describes how structured electronic communications will work in the national e-health environment.

It describes:

- The concept of services in the national e-health environment;
- The actual infrastructure services which will be available; and
- The interactions of those services: how they are used together to achieve connectivity.

3.2 Requirements

This architecture has been defined to meet the following requirements:

- Support many types of business domain services.
- Distribute service instances across a national e-health environment.
- Support service instances provided by multiple independent entities.
- Allow for service instances to be added, removed, and changed.
- Make use of the national infrastructure service for identifiers.
- Make use of the national infrastructure service for PKI certificates.
- Provide reliable exchange.

These requirements have been distilled from the expected needs of pathology result reporting, discharge summary, referrals and medications management.

3.3 Assumptions and dependencies

The architecture has been designed with a number of assumptions and dependencies. They are:

- Provider organisation focus;
- Service orientation;
- Specific business functionality; and
- Public Key Infrastructure (PKI).

3.3.1 Provider organisation focus

In the national e-health environment, there are three general categories of entities: individuals, providers, and organisations. This architecture is concerned with communications with organisations: either organisation to organisation, or provider to organisation.

The architecture focuses on achieving communications with entities owned by healthcare providers. These entities are devices or computer programs that are used or operated by healthcare provider organisations.

Individuals have a role to play in their own healthcare, but their connectivity needs are currently not directly addressed by this architecture. Their involvement with the national e-health environment will be conducted through their encounters with healthcare providers. Those providers or provider organisations will interact on behalf of the individual.

Providers might also need to provide services for other providers or organisations to use. Currently, that ability is not supported by this architecture. Providers will have to operate services through an organisation. If an individual provider wishes to operate services, they will have to go through an organisation (even if that organisation is a sole trader organisation).

The term organisation refers to the systems at the organisation's boundary. An organisation may be made up of many systems. This architecture is concerned with those systems that are exposed to external entities to interact with, sometimes referred to as the organisation's interface. This architecture is not concerned with how organisations implement their internal systems, as long as the interfaces at the organisation's boundaries are met. An organisation can use some of the architecture for their internal systems, but that is beyond the scope of this architecture.

3.3.2 Service orientation

This architecture takes a services view of the business environment.

The main building block for achieving interoperability is the concept of a service. Services are units of well-defined functionality offered with a standard interface. Services can be offered by a range of different entities: from the national level down to individual healthcare organisations. Having a standard interface allows external entities to independently use or implement those services.

Services will be accessed using a common approach which encourages consistency and reuse. The common approach will also ensure that communications with the services are secure and reliable.

Business processes are analysed in terms of business services and the interactions between them. These business services clearly identify and separate functions and responsibilities.

The business services are realised as technical services. A technical service may be a physical service or involve manual processes. This architecture focuses on electronic services implemented by computer systems, so that is the only type of technical service referred to by this architecture.

In describing services, the following concepts will be used:

Service provider

The entity that offers a service that can be used by another entity (i.e. by the service invoker).

Service invoker

The entity that uses a service that is provided by another entity (i.e. by the service provider).

Service request

A message that is sent from the service invoker to the service provider to invoke (i.e. use) the service.

Service response

A message that is sent from the service provider to the service invoker. It indicates the result (if any) of using the service.

For additional service concepts, see the *Concepts and Patterns for Implementing Services* [CPIS2008].

3.3.3 Specific business functionality

This architecture includes services that address specific healthcare business functions. The business services reflect the specific healthcare function being performed.

The technical services derived from the business services will therefore also be specific to the healthcare function being provided.

3.3.3.1 Generic vs specific

This approach results in specific technical services which are tied to specific business functions. In some situations, the technical services will offer functions that are peculiar to that business function. In other situations, the technical services will have functions that are very similar to those provided by other technical services—but they will always be specified as a different service.

For example, this approach will result in having a service to deliver pathology result reports and a completely different service to deliver discharge summaries. It will not result in a generic document delivery service that is used for both functions.

Some of the benefits of this approach are:

- Design time binding of services instead of run-time binding.
- Services are tightly coupled to the business logic that they support.
- Services can support all the necessary business requirements.
- Services do not need to support functionality that is not needed by the business requirements.
- Services can evolve independently to match changing business requirements that affect one business process but not the others.
- The function provided by a service is indicated by its interface.
- Services can be automatically orchestrated using information exposed by their interfaces.
- It reduces the risk of accidental information management errors.

There are arguments for generic service interfaces, but they do not provide the same benefits as specific services. It has been argued that a generic approach requires less work to implement, because one generic service implementation can be reused to support multiple business processes. This argument does not take into account that this only affects the interface level, and that the backend implementation will still need to be specific to a business process. There is nothing preventing an implementation from reusing code to simplify the effort required to implement specific interfaces; for example, implementing specific facades onto a generic middle layer. A generic approach compromises the benefits listed above.

3.3.4 Public Key Infrastructure (PKI)

This architecture makes use of Public Key Infrastructure (PKI) to provide security functions.

PKI is a standard mechanism for managing cryptographic keys, and they will be used by the architecture to support digital signing and encryption.

4 Business viewpoint

This chapter describes how connectivity is achieved from the business viewpoint.

The main focus of this chapter is describing the interactions used in the architecture (section 4.1). These interactions bring together the services defined by the architecture (section 4.2).

Access control is also briefly described in section 4.3.

4.1 Interactions

This section describes the interactions used in this architecture to achieve connectivity.

Interactions need to be considered at two levels:

- At the business level, the interactions are derived from the business processes. They will be different depending on which business process is being supported.

The details of the business level interactions are outside the scope of this document. They will be defined by the particular business service documents. For example, for the pathology result reporting business processes see the Pathology Result Reporting Package [PRR2008].

- At the infrastructure level, there will also be common interactions used to establish and support the business processes.

The details of the infrastructure level processes will be covered by this document. There are three infrastructure interactions that have been identified:

- Bootstrapping interaction;
- Identification interaction; and
- Core connectivity interaction.

The three interactions are shown together in Figure 2: Interactions combined. For simplicity, the IHI and HPI-I services have been omitted from this diagram.

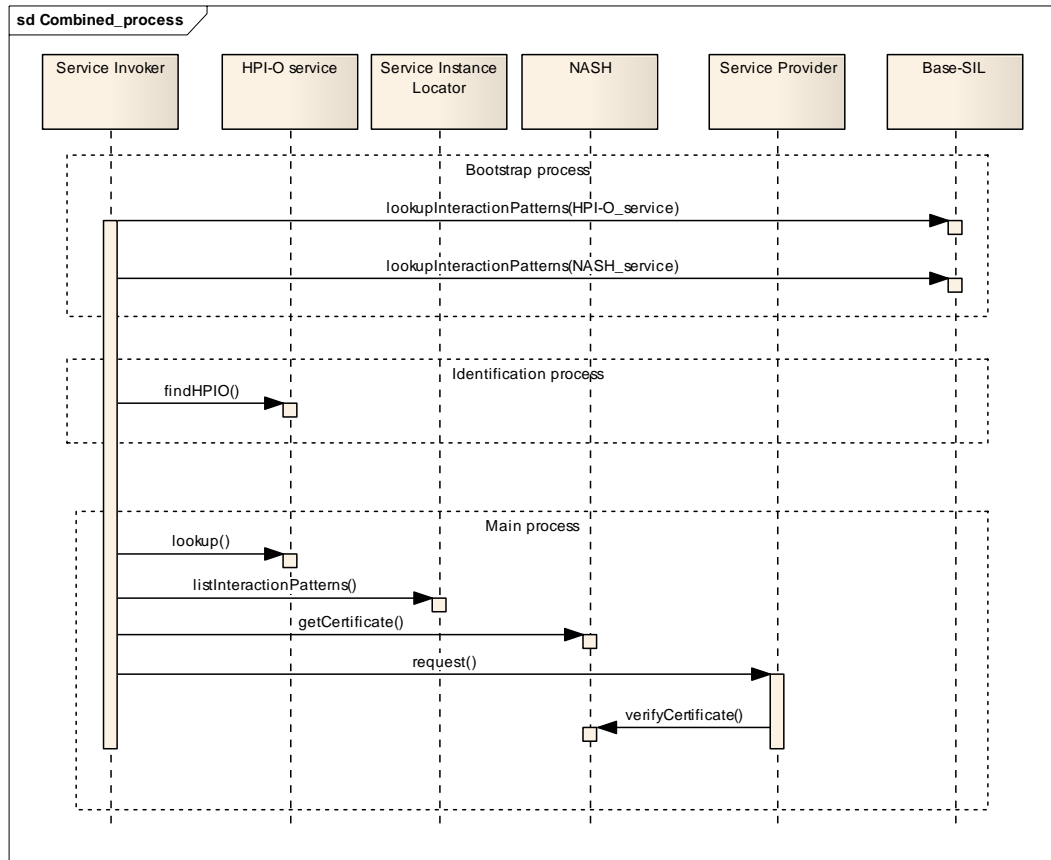


Figure 2: Interactions combined

4.1.1 Bootstrapping interaction

4.1.1.1 Objective

The bootstrapping interaction allows a service invoker to obtain the information needed to invoke the national infrastructure services. It is used to obtain the address and security information for services such as the IHI, HPI-I, HPI-O, and NASH services. Although these services will be fairly stable, this interaction provides a mechanism that can be used to initialize service invokers and reconfigure them if the services change (e.g. when their digital certificates are renewed). It also provides a mechanism for supporting other national infrastructure services when they become available.

This bootstrapping interaction is optional. Service invokers can be pre-populated with the addresses and certificates for the known national infrastructure services they will use, instead of using this bootstrapping interaction to find them. If there is no Base Service Instance Locator, this pre-population approach will have to be used.

This bootstrapping interaction is used before the identification interaction (section 4.1.2). The main interaction relies on using the national infrastructure, so this bootstrapping interaction provides it the information needed to use them.

4.1.1.2 Prerequisites

This interaction assumes the following is known by the service invoker:

- Location of the Base Service Instance Locator (Base-SIL).
- PKI certificates for accessing the Base-SIL.
- Identifier for the national infrastructure entity.

The Base Service Instance Locator (Base-SIL) is an instance of the SIL that contains entries for the national infrastructure services. Normally, SIL

instances contain entries relating to particular provider organisations. In the case of the Base-SIL, the entries are associated with the national body instead of any particular provider organisation.

4.1.1.3 Process

The steps are:

1. Query the Base-SIL for the IHI service.
2. Query the Base-SIL for the HPI-O service.
3. Query the Base-SIL for the HPI-I service.
4. Query the Base-SIL for the NASH service.

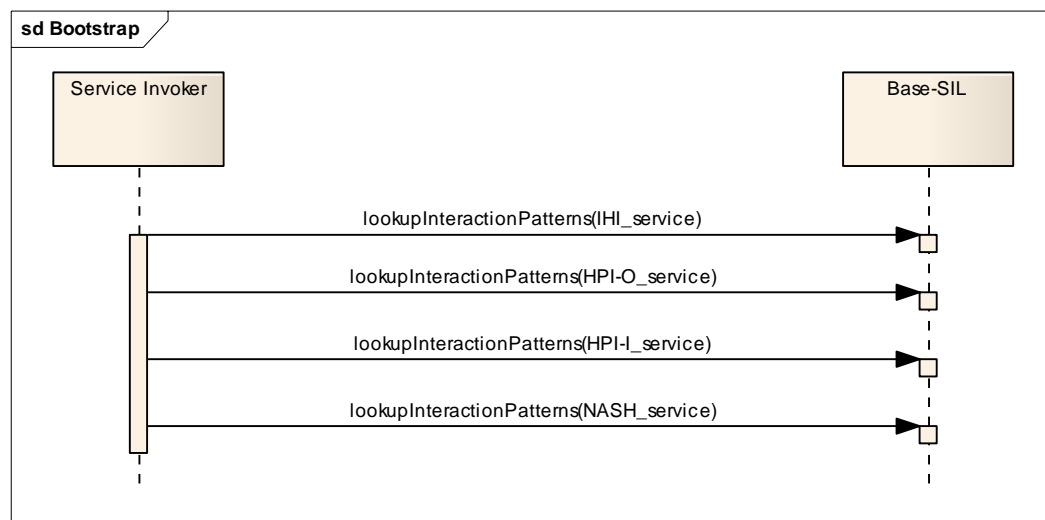


Figure 3: Bootstrapping interaction

4.1.2 Identification interaction

4.1.2.1 Objective

The identification interaction obtains identifiers for the entities involved in the communications. Typically this involves obtaining the IHI number, HPI-I numbers and HPI-O numbers.

This interaction is only needed if some or all of these identifiers are not known beforehand. For example, if they are provided by the patient or have been saved from a previous event, then this interaction is not needed.

4.1.2.2 Prerequisites

This interaction assumes the following are known:

- Traits to identify the entity.
- The location of the identification services (see the bootstrapping interaction in section 4.1.1)

4.1.2.3 Process

The step is:

1. Query the identification service using the traits.

This interaction is the same for every type of identifier. The appropriate identification service is used for the type of identifier being obtained.

Examples of different identifier types are:

- The IHI number for any references to individuals in the communications.
- The HPI-I numbers for the healthcare professionals involved.

- The HPI-O numbers for the organisations involved.

This interaction is illustrated in Figure 4 for the IHI service. Similar sequence diagrams can be shown for the HPI-I and HPI-O services.

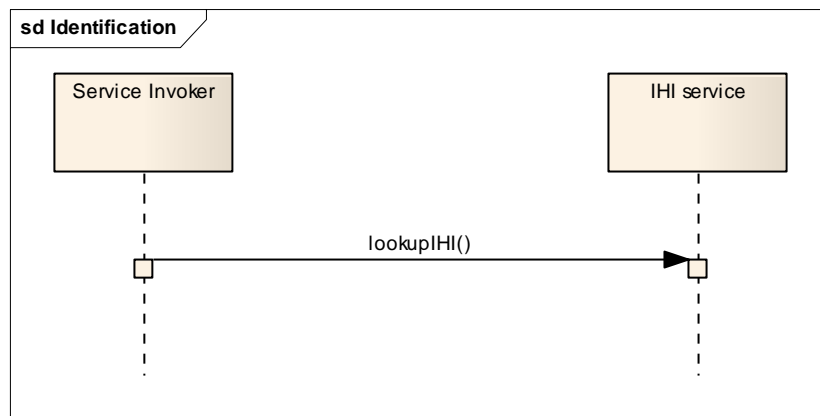


Figure 4: Identification interaction (for IHI)

4.1.3 Core connectivity interaction

4.1.3.1 Objective

The core connectivity interaction enables a service operation to be invoked. This interaction is about obtaining the necessary information to invoke a service (i.e. obtaining identifiers, certificates and location) and then invoking that service.

For the service invoker, it involves:

- Determining where the service instance is (i.e. the address) and what certificates are required;
- Obtaining the certificates required to secure the communications;
- Sending the request; and
- Processing the response.

For the service provider, it involves:

- Validating and verifying the request;
- Performing the operation; and
- Sending back the response.

4.1.3.2 Prerequisites

This interaction assumes the following is known by the service invoker:

- The type of service being sought. This is known as the *service category*.
- Identity of the entity whose service instance is being sought.
- Identities of other entities involved in, or referred to by, the communications (see the identification interaction in section 4.1.2)
- The location of the national HPI-O service (see the bootstrapping interaction in section 4.1.1)
- The location of the NASH service (see the bootstrapping interaction in section 4.1.1)

4.1.3.3 Process

For the service invoker, the steps are:

1. Obtain the location of the target entity's Service Instance Locator from the national HPI-O service, by looking up the record for that HPI-O and then extracting the SIL information from that record.
2. Obtain the interaction patterns supported by the entity for the desired service category from the entity's Service Instance Locator.
3. Choose which interaction pattern to use. In many cases an entity will only support one interaction pattern. If more than one is supported, the service invoker chooses one of them to use.
4. Obtain the necessary digital certificates for the chosen interaction pattern from the NASH. The certificates needed are indicated in the SIL interaction pattern information.
5. Create the service request.
6. Sending a service request to the service provider.
7. Receiving a service response from the service provider.
8. Validating the service response.
9. Verifying the digital certificates used by the service provider.
10. Processing the response.

For the service provider, the steps are:

1. Receiving the service request from the service requestor.
2. Validating the service request.
3. Verifying the digital certificates used by the service invoker.
4. Performing the operation.
5. Sending a service response back to the service requestor.

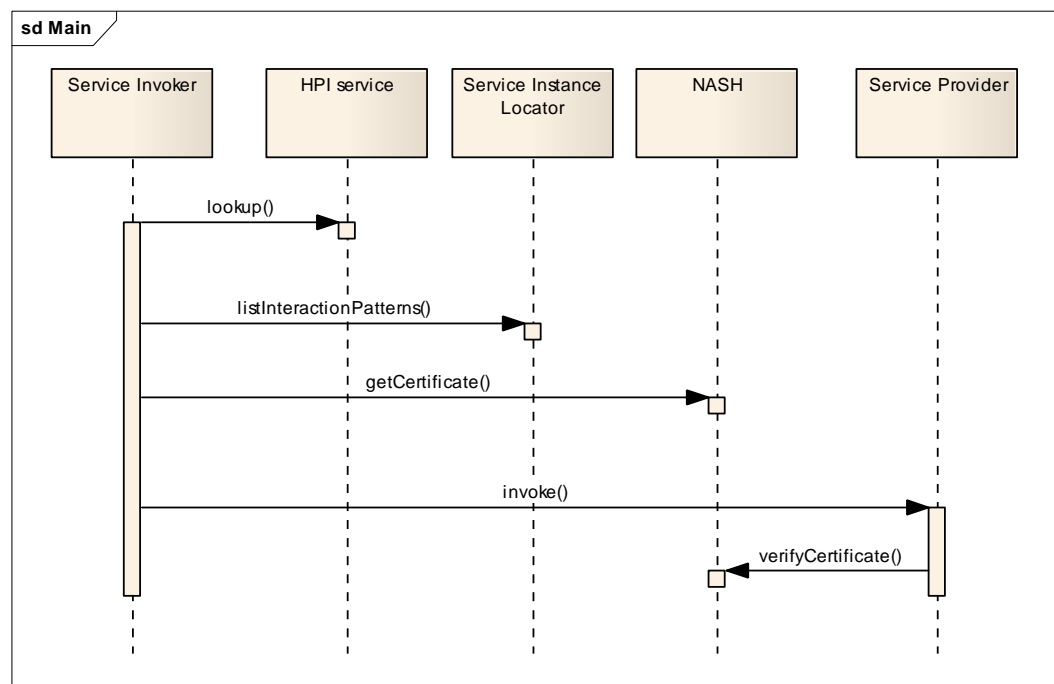


Figure 5: Core connectivity interaction

These interactions define the functionality required to achieve connectivity, but there are many ways to implement a system that has this behaviour. For example, the use of retries and caching is described in the *Connectivity: Implementation Guide* [CIG2008]. An example of one way to implement these functions is described in the *Connectivity Component: High Level Design* [CCHLD2008]

4.2 Services and roles

This section describes the services used by the architecture.

The business functions of these services are described in this section. The technical realisations of these services are described in section 6.5.

These services are:

- Business services
- Identification services
- PKI services; and
- Directory services

These services are illustrated in Figure 6. Although there can be many different types of business services, only four examples are shown in this figure.

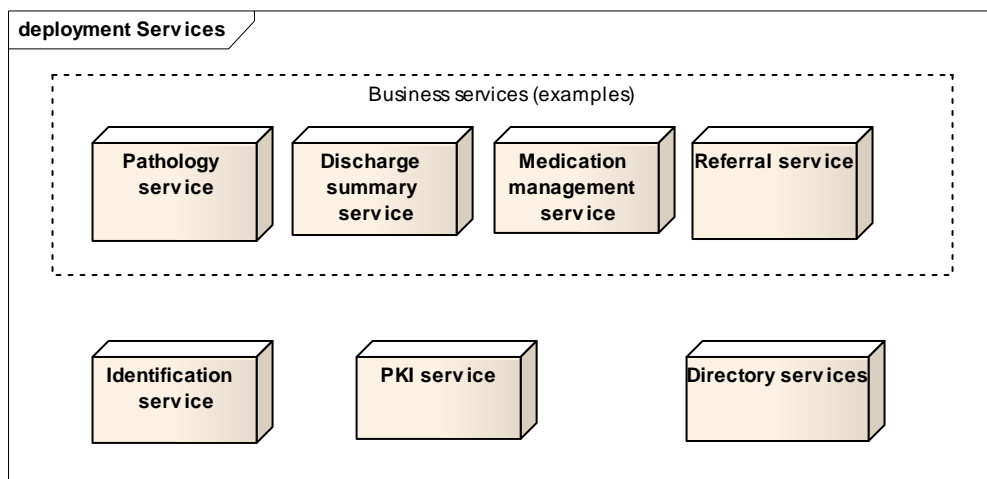


Figure 6: Types of connectivity services

4.2.1 Business services

The approach begins with identifying the business services. These are services that arise from the function of the organisation and its processes. In the case of healthcare, these will be services that provide specific healthcare related functions. For example, pathology, discharge, referrals, and medication management.

Business services are technology independent. They can be implemented using different mechanisms. For example, from hand written paper carried by people through to electronic data carried over computer networks. Although the focus of this document is on the latter, business services need to be defined independently of the technology.

Business services can then be implemented by technical services. This is described in section 6.1.

4.2.2 Identification services

All of the entities involved in communications are usually identified. These entities include those involved in conducting the communications as well as the subject of the communications and related entities. For example, when sending a pathology result, the entities involved in the communication would be the pathology laboratory and the general practice; the subject of the communication is the patient, and the related entities could be the laboratory officer and the doctor.

In some scenarios, the entities do not need to be identified for the business process but they still might need to be identified for access control and auditing purposes. Currently, there are no scenarios that require anonymity (i.e. where one or more entities are not identified), but this requirement might appear in future business scenarios.

Identification services are needed by the architecture. Unique identifiers are assigned to entities, and these services provide methods to support the use of those identifiers. These methods include:

- Searching for an entity's identifier based on identifying traits about the entity.
- Obtain information about an entity from its identifier.

NEHTA will be providing at least two types of identifier services:

- Individual Healthcare Identifier (IHI) for identifying consumers of healthcare services.
- Healthcare Provider Identifier (HPI) for identifying healthcare providing people and healthcare providing organisations.

There are two types of HPIs: Healthcare Provider Identifier for Organisations (HPI-O) which are issued to organisations (e.g. general practice, hospitals, pathology laboratories), and Healthcare Provider Identifier for Individuals (HPI-I) which are issued to people (e.g. general practitioners, specialists, pharmacists).

The IHI and HPI services are being developed by the NEHTA Unique Healthcare Identifier (UHI) project.

4.2.3 PKI services

The PKI services support the use of PKI in the national e-health environment.

This architecture uses PKI to authenticate entities, and to digitally sign and encrypt data.

These will be based on Public Key Infrastructure (PKI) standards around X.509v3 certificates. They will provide a number of methods to support the use of X.509v3 certificates. For establishing connectivity, these methods are used:

- Retrieval of the public certificate(s) issued to an entity based on the entity's identifier.
- Retrieval of a particular public certificate based on a value that identifies that particular certificate.
- Checking if a given certificate key pair has been revoked or not.

NEHTA will be providing these services through the:

- National Authentication Service for Health (NASH).

4.2.4 Directory services

The directory services provide a mechanism to discover available service instances and to obtain the information necessary to invoke the service instances¹.

This service is very important in the national e-health environment. The environment is distributed, where there are many service instances running and controlled by different organisations. Over time, service instances can be added, removed, or changed. The service instance services provide a place to obtain information about these dynamically changing service instances.

This is only a service for providing technical information about known service instances. This service should not be confused with a “yellow pages” business service directory, which is used to find out who offers what type of healthcare services. The directory services described here is only used after a particular provider has been already identified.

The following functionality is provided:

- Retrieval of the address where the service instance is available.
- Retrieval of the certificate required to invoke the service instance.

NEHTA will be defining specifications for:

- Service Instance Locators (SIL).

For more information, see [SIL2008].

There are two types of SIL operating: the provider organisation’s SIL and the base SIL.

4.2.4.1 Provider Organisation’s SIL

It is expected that every provider organisation that makes services available will be publishing those services in their SIL.

There will be at most one Service Instance Locator associated with each provider organisation. If the organisation offers technical services, they would publish the information needed to invoke those service instances to their SIL.

A single instance of the SIL can service one or more provider organisations. For example, one company could host the SIL for several provider organisations.

The design of the Service Instance Locator allows for different deployment options. At one extreme, every provider organisation can provide their own SIL; that SIL will only contain entries for that provider organisation and for no other provider organisation. An instance of a SIL could be shared, where it contains entries for several provider organisations. At the other extreme, a single SIL could be deployed to contain entries for every provider organisation in the country.

4.2.4.2 Base SIL

There are a number of national infrastructure services in the national e-health environment and these services need to be located to be used. This is the purpose of the Base Service Instance Locator (Base-SIL).

The Base-SIL is just another instance of a SIL. Instead of containing entries of services offered by a provider organisation, it contains entries corresponding to the national infrastructure services.

¹ The term “service instance” is defined in the *Concepts and Patterns for Implementing Services* [CPIS2008]. It refers to a running deployment of a service. It is different from a service implementation: a system (i.e. computer programs) that could be deployed in many places. It is also different from a service interface specification: a standard that could be implemented by many different systems.

For example, it will contain entries for the IHI service, HPI-I service, HPI-O service and NASH. Other national services will be added when they become available.

The Base-SIL can be used as the first point of call to obtain the information about the national services. It can be used to bootstrap the process of using business services.

The Base-SIL will operate at a well known address and with well known digital certificates.

4.3 Access control

Access control (also known as authorization) is an important feature that must be present. It controls who can access data and can invoke operations.

This version of the architecture does not prescribe any particular access control mechanisms. Standard access control mechanism(s) may be defined in the future if there is sufficient requirements and demands for them.

Access control is still needed by services. Therefore, individual service specifications, service implementations and service instances must determine the appropriate access control mechanism they use. The mechanism needs to take into account the nature and purpose of the service to determine what is appropriate.

For example, some services might allow all entities authenticated by the NASH to have access. Other services might establish one-on-one relationships with entities through external mechanisms, and then only allow those entities access.

5 Information viewpoint

This chapter describes elements of the connectivity architecture from the information viewpoint.

Most of the details about information will be the subject of the particular services being specified. Therefore, this chapter will only describe the common data types used to establish connectivity (section 5.2) and the different types of terminology required (section 5.3).

5.1 Semantic information model

The information models are specific to the particular service which uses it.

There is no single information model that applies to every service. When Web services is the implementation technology (see section 6.1), any information model must be represented using XML.

Information models for each service are defined in the specification documents for that particular service. For example,

- Information model for the Service Instance Locator (SIL) can be found in [SIL2008].

5.2 Data types

There are several common data types used by this architecture.

5.2.1 Identifiers

Entities need to be identified in a unique manner that the communicating entities agree upon.

Different types of identifiers can be used to identify an entity.

Although NEHTA is developing IHI and HPI national identifiers, other types of identifiers could be used. For example, during the transition period other types of identifiers could be used, or in some situations different types of locally issued identifiers might be more suitable.

To support the use of different types of identifiers, the concept of a qualified identifier is used in this architecture. This is simply the identifier value and a qualifier that uniquely distinguishes what type of identifier it is.

Qualified identifiers are defined in [QI2008].

5.2.2 Certificates

Public Key Infrastructure (PKI) uses X.509v3 certificates to represent the public key issued to an entity.

The architecture needs to reference these certificates. The primary point where this is used is in SIL entries to indicate which certificate is used for encrypting the service requests.

Certificates can be referenced using a number of different mechanisms, depending on how the certificate authority has created the certificates. A digest of the certificate can be used. If the Common Name is unique, it can also be used.

NEHTA is developing the National Authentication Service for Health (NASH) which will be issuing certificates, and it will have a preferred identification mechanism for certificates.

To allow for other types of certificates and mechanisms to identify certificates, the concept of a qualified certificate reference is used in this architecture. This is simply the reference value and a qualifier. The qualifier indicates what type of certificate it is (i.e. implicitly indicating who issued it) and what type of reference value is being used.

Qualified certificate references are defined in [SIL2008]. The SIL adds to the qualified certificate reference a usage field, which indicates the purpose for the certificate. For example, its usage identifies it as the certificate to be used for encrypting the payload.

5.3 Terminology

5.3.1 Service identifiers

This architecture involves different types of services and different processes involving those services. These will be given unique identifiers to distinguish them and to refer to them.

These identifier values will be defined in the specifications for the particular service.

Uniform Resource Identifiers (URI) will be used for these identifiers. It is the responsibility of the specifications to assign a globally unique URL value. Since it is a URI, it is relatively easy to ensure they are assigned uniquely.

5.3.2 Content terminology

Standard data structures and terminology are required for the data being communicated.

This is specific to the business domain being considered. Therefore, they will need to be described in the domain service documentation. For example, for the pathology result reporting terminology see the Pathology Result Reporting Package [PRR2008].

6 Technology viewpoint

This chapter describes the architecture from the technology viewpoint.

The architecture implements some of its business services using a services implementation technology such as Web services (section 6.1). These services will be defined by following a set of common design patterns (section 6.2). For some services, the data being transmitted needs extra security over what is provided by transport security therefore a standard payload security mechanism is defined that can be used by different service specifications (section 6.3).

The definition of a service comes together in its service interface specification. Some example services are described in section 6.4.

6.1 Services technology

Business services will be implemented using the most appropriate technology for that service.

The first protocol recommended by NEHTA is Web services. Web services is suitable for operation based machine to machine communications. However, Web services is not suitable for other types of communications. For example, it is not designed to handle the transmission of large datasets and streaming information. Different protocols will be needed for tasks where Web services cannot be used.

6.1.1 Web services

When Web services is used, the service interface specifications will be specified (in part) using the Web Services Description Language (WSDL) and service invokers and service providers will communicate using SOAP messages.

Details about the particular Web services standards and the profiles used by the architecture are described in Web Services Profile v3.0 [WSP2008].

6.1.1.1 Distributed over the Internet

The services are distributed across the nation and connected together using a common computer network.

The network is the Internet. Service invokers will securely connect to service providers over the Internet.

6.1.1.2 No centralised connectivity service

There is no centralised service to handle communication issues for the distributed entities.

Inside a single organisation, sometimes an Enterprise Service Bus (ESB) component is used to handle communications, but there is no ESB for the nation as a whole. Therefore, each service must manage its own communications.

6.1.1.3 Reusing infrastructure services

Each service must manage its own interface to the network. Since there is no centralised service to handle communications, each service provider and service invoker must handle their own connectivity needs.

This task of managing communications is made easier by the infrastructure services. These provide common services that every entity can use. Instead of

each service having to implement their own infrastructure, they can make use of the national infrastructure services.

Currently, there are national infrastructure services to manage identifiers and digital certificates. There is also the Service Instance Locators (SIL) to manage service instances. These are described in section 6.4.

6.2 Patterns for services

Different types of Web services can be defined, deployed and invoked in different ways. Although there can be an infinite variety of services, a few common patterns for services can be identified. These patterns help improve consistency between services, which can lead to reduced costs in understanding, design, development and deployment.

The patterns identify how services are used to implement a particular process.

The patterns identified for the architecture are documented in *Concepts and Patterns for Implementing Services* [CPIS2008]. The patterns are applied to particular business services, and they are used in the specific specifications for a particular business domain (e.g. pathology).

Different patterns can be used by different service specifications. The particular pattern that will be used will depend on the particular business service being implemented and the capabilities of the entities involved.

These patterns are defined abstractly. It is the responsibility of the service interface specification to define how the patterns are concretely implemented.

6.3 Security

The security in this architecture is built upon the transport layer security of the protocols being used. For Web services, security is provided by WS-Security.

For situations when application level security is required, an XML Secured Payload has been defined. This is described in section 6.3.1.

The checking of digital signatures is an important part of security. This occurs in many places, such as when using WS-Security or the XML Secured Payload. To ensure a common understanding of the signature checking process, this architecture describes the process in section 6.3.2.

6.3.1 XML Secured Payload

There are two security features that are often required by an application: to authenticate the source of some data and that the data has not been modified, and to ensure the confidentiality of data. These two features can be provided by using digital signing and encryption.

The connectivity architecture has defined a mechanism for representing signed data and encrypted data. The mechanism has been defined for XML formatted data, and is described in the *XML Secured Payload Profile* [XSP2008].

The XML Secured Payload was designed to represent signed and encrypted data that is being transmitted over Web services. It is used to provide additional security over what the transport layer security mechanisms can provide. It is used to implement application level security requirements, such as ensuring end-to-end security from the originating party to the destination party.

The XML Secured Payload can also be used as a mechanism to store secured data. Such as storing signatures for later checking, or storing confidential data in a database.

6.3.2 Checking digital signatures

When digitally signed data is received, that signature should be checked before relying on the signature.

This section describes the types of checks that should be performed on a digital signature. The tests described in this section, can be applied to different types of signatures: such as for the XML Secured Payload (section 6.3.1) and WS-Security.

The checks can be divided into three categories:

- Confirm identities;
- Signature validation; and
- Certificate path validation.

The three categories of checking are important for ensuring that signatures can be trusted. The checks ensure that the signer is the expected entity, the signature was created by the certificate owner, and that certificate is still trusted.

6.3.2.1 Confirm identities

Check that the identity of the signer according to the policy for the business process being executed.

For example, the policy might state that the signer is identified in the data. In this situation, the identifier in the data must match the signature creator, who's identity can be found using the signing certificate.

This check prevents false claims being made in the data about who signed the data. It will detect cases when the data claims it is signed by one entity but is actually signed by a different entity.

6.3.2.2 Signature validation

Check the integrity of the signature. Perform the appropriate cryptographic operation to verify the signature received was applied to the data received by the private key corresponding to the signer's certificate.

This check prevents a disconnection between the signature and the data. For example, if the data has been tampered with, the wrong key used, or the wrong data signed.

6.3.2.3 Certificate path validation

Check that the signer is a trusted entity.

This check involves checking the following:

- The integrity of the signing certificate. This checks if the issuer has correctly signed the certificate. It will detect forged certificates which were not issued by the claimed issuing authority.
- The certificate has been issued by a trusted authority. This checks if the issuer is one that is accepted by the application. It will detect certificates issued by parties that the application does not trust (even though they are valid certificates). This is usually a check to see if the certificate was issued by a known Certificate Authority.
- The certificate has not been revoked. This checks if the issuer still recognises the certificate. This check can either be performed using Certificate Revocation Lists (CRL) or a real-time service such as OCSP [RFC5280] [RFC2560].

If there are multiple certificate authorities, each of the above checks may need to be performed on each component of the certificate chain—all the way

up to the root certificate. If an intermediate certificate authority is trusted, then the checking can stop there.

Further details on certificate path validation can be found in [RFC5280].

6.4 Reliability

Nearly all communications in health need to be reliable. Mechanisms must be in place to ensure that messages are delivered. In a distributed environment, each service invoker or service provider must play a role in providing reliability.

These mechanisms need to have a defined Service Level Agreement (SLA) when they are deployed. This SLA may be defined as a part of the service interface specification, the conformance criteria, or by mutual agreement between communicating parties.

In this architecture, three mechanisms have been identified that can be used to provide reliability:

- Business acknowledgements;
- Operation acknowledgements; and
- Network acknowledgements.

6.4.1 Business acknowledgements

The business process being performed may have business level acknowledgements. These are sent by the destination entity back to the source entity to indicate when the data has been received and/or acted upon. The source entity knows that the data was reliably received when it has received the acknowledgement.

Business acknowledgements can be sent via a separate operation, or in the response message of invoking a service operation. When sent in a separate operation the main process is followed with the two parties reversed.

If a failure occurs at this level, the entity needs to respond according to the rules of the business process or escalate the problem. For example, the business process could require alternative services to be invoked. The problem could also be escalated to a person to deal with.

6.4.2 Operation acknowledgements

Every operation invocation results in a service response. This is the request-response pattern described in the *Concepts and Patterns for Implementing Services* [CPIS2008].

The service invoker knows that the data was reliably received by the service provider when it has received the response.

If a failure occurs at this level, the entity needs to respond according to the service interface specification or escalate the problem. For example, the operation may allow retries, in which case the service invoker could attempt to retry the operation at a later time. The problem could also be escalated to a person to deal with.

6.4.3 Network acknowledgements

The service invoker is unable to invoke a service if a network connection cannot be established to it. In this situation, the service invoker clearly knows that the data was not delivered.

If a failure occurs at this level, the entity can either retry the operation or escalate the problem. For example, the service invoker may wait an hour and

retry the operation. The problem could also be escalated to a person to deal with.

6.5 Service interface specifications

This section describes some of the national infrastructure services that NEHTA is currently developing.

6.5.1 IHI services

Individual Healthcare Identifiers (IHI) are unique identifiers for individuals in the healthcare system. Sometimes individuals are referred to as consumers or patients.

This is one type of identification service as identified in section 4.2.2.

6.5.2 HPI services

Healthcare Provider Identifiers (HPI) are unique identifiers for healthcare providers for both individuals and organisations.

This is another type of identification service as identified in section 4.2.2.

6.5.3 NASH services

The National Authentication Services for Health (NASH) provides services for issuing PKI keys and certificates, and to support the use of those PKI certificates.

This is an authentication service as identified in section 4.2.3.

6.5.4 Service Instance Locator

The Service Instance Locator (SIL) are services for obtaining information for invoking a service instance. This information includes the network address of where the service is available and the certificates required to use it.

This is a directory service as identified in section 4.2.4.

For further details about the SIL, see [SIL2008].

Appendix A: Informative references

- [CIG2008] NEHTA, *Connectivity: Implementation Guide v1.0*, 1 December 2008.
- [CCHLD2008] NEHTA, *Connectivity Component: High Level Design*, 1 December 2008.
- [NIF2007] NEHTA, *Interoperability Framework v2.0*, 2007.
- [PRR2008] NEHTA, *Pathology Result Reporting Package v1.0: Readers' Guide*, 1 December 2008.
- [RFC2560] IETF, *X.509 Internet Public Key Infrastructure Online Certificate Status Protocol – OCSP*, M. Myers, R. Ankney, A. Malpani, S. Galperin, C. Adams, June 1999.
- [RFC5280] IETF, *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*, D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, W. Polk, May 2008.
- [SIL2008] NEHTA, *Service Instance Locator: Architecture v1.0*, 1 December 2008.