# Statewide Send and Receive Patient Record Exchange

## Technical Specification

Version 1.1 ● January 10, 2013

# Technical Specification

# Statewide Send and Receive Patient Record Exchange

Version: [1.1] Revision Date [01/10/2013]

| | | |
|---|---|---|
| **Executive Sponsor** | | |
| [Name] | [Email] | [Telephone] |
| Signature | | Date |

| | | |
|---|---|---|
| **Technology Sponsor** | | |
| [Name] | [Email] | [Telephone] |
| Signature | | Date |

| | | |
|---|---|---|
| **Project Manager** | | |
| [Name] | [Email] | [Telephone] |
| Signature | | Date |

| | | |
|---|---|---|
| **Security Officer** | | |
| [Name] | [Email] | [Telephone] |
| Signature | | Date |

# Table of Contents

# 1 Document History

Paper copies are valid only on the day they are printed. Contact the author if you are in any doubt about the accuracy of this document.

## Revision History

| Revision Number | Revision Date | Summary of Changes | Author |
|---|---|---|---|
| V0.1 | 06/27/2011 | Draft 0.1 | Vince Lewis |
| V0.2 | 07/19/2011 | Draft 0.2 | Vince Lewis |
| V0.3 | 08/01/2011 | Draft 0.3 | Vince Lewis |
| V0.4 | 08/21/2011 | CA comments, Federation | Vince Lewis |
| V0.5 | 08/31/2011 | Cleanup, additional mappings for all S&I fields | Vince Lewis, Charlene Lichtner, Sean Kelly |
| V0.6 | 09/13/2011 | Final comments | Vince Lewis |
| V0.7 | 09/27/2011 | Proposed for Final Review and Subgroup Ratification. | Vince Lewis,<br>Sean Kelly |
| V1.0 | 11/2/2011 | Minor edits based on feedback from ONC | Vince Lewis, Nick VanDuyne |
| V1.1 | 01/10/2013 | Updates after environment review of latest specifications | Lin Wan, Salim Kizaraly |

## Reference Documents

Please see the following documents for more information:

| Document Name | Version | Author |
|---|---|---|
| | | |
| | | |
| | | |
| | | |

# Distribution List

This document has been distributed to the following participants:

| Name | Position | Company | Action |
|---|---|---|---|
| | | | |
| | | | |

# 2 Introduction

The Office of the National Coordinator for Health Information Technology (ONC) is at the forefront of the administration's health IT efforts and is a resource to the entire health system to support the adoption of health information technology and the promotion of nationwide health information exchange to improve health care.

The Nationwide Health Information Network (NwHIN) is a set of standards, services and policies that enable secure health information exchange over the Internet. The network will provide a foundation for the exchange of health information across diverse entities, within communities and across the country, helping to achieve the goals of the HITECH Act. This critical part of the national health IT agenda will enable health information to follow the consumer, be available for clinical decision making, and support appropriate use of healthcare information beyond direct patient care so as to improve population health.

## 2.1 Purpose of this document

The purpose of this document is to outline the technical design for a Statewide Send and Receive Patient Record Exchange service. This is also referred to as a "Push" Model within state Health Information Exchange utilizing intermediaries such as: Health Information Organizations or Health Information Service Providers for message routing. It is based on the Functional specification for this exchange.

## 2.2 Definitions

The following definitions are important to keep in mind throughout the Statewide Send and Receive Patient Record Exchange System Requirements specification:

- **Affinity Domain** is a group of healthcare enterprises that have agreed to work together using a common set of policies and share a common infrastructure. With Direct, there is an implication of shared trust anchors.
- **Certificate Authority** (CA) is defined as an organization that issues digital certificates in a public key infrastructure environment
- **Healthcare Provider Directory (HPD)** is an IHE profile which supports management (persistence and access) to healthcare provider's information in a directory structure. Two categories of healthcare providers are included in the directory.
    - Individual Provider – A person who provides healthcare services, such as a physician, nurse, or pharmacist.
    - Organizational Providers – Organizations that provide or support healthcare services, such as hospitals, Healthcare Information Exchanges (HIEs), Integrated Delivery Networks (IDNs), and Associations.

- **Health Information Exchange** (HIE) is defined as the transfer of healthcare information electronically and securely across Health Information Organizations (HIO) within a region such as a state, community, hospital system or a physician network. HIE may also denote an HIO that provides HIE services.
- **Health Information Organization** is an organization that oversees and governs the exchange of health-related information among organizations according to nationally recognized standards.
- **Health Information Service Provider** (HISP) is defined as an entity that is responsible for delivering health information as messages between senders and receivers over the Internet
- **IHE Cross-Enterprise Document Media Interchange** (XDM) **profile** is a specification or the exchange of electronic health record documents on portable media. XDM provides an option for zipped file transfer over e-mail, which is very relevant to the Direct Project specifications.
- **IHE Cross-Enterprise Document Reliable** (XDR) **Interchange profile** is a specification for the interchange of electronic health record documents through reliable point-to-point network communication, based on pushing information.
- **Integrating the Healthcare Enterprise** (IHE) is a group of healthcare industry stakeholders that promotes and defines coordination of established standards to provide meaningful and effective information exchange.
- **Lightweight Directory Access Protocol (LDAP)** is an application protocol for querying and modifying data of directory services implemented in Internet Protocol (IP) networks.
- **Nodes** are systems with IP or URL addresses owned or used by Entities to send and receive messages
- **Patient Look-up Service** (PLS) is defined as a system that allows healthcare clinicians to retrieve patient records across disparate health system through the use of a Record Locator Service.
- **Payload Conversion** is defined as the translation service between various content types. In the context of this document, payload conversion refers to content conversions between HL7 V2 messages, HL7 V3 messages, and clinical documents (e.g. CCD).
- **Private Key and Public Key** In public key cryptography, a public and private key are created simultaneously using the same algorithm (a popular one is known as RSA) by a certificate authority (CA). The private key is given only to the requesting party and the public key is made publicly available (as part of a digital certificate) in a directory that all parties can access. The private key is never shared with anyone or sent across the Internet. You use the private key to decrypt text that has been encrypted with your public key by someone else (who can find out what your public key is from a public directory).
- **Protocol Conversion** is defined as the translation service between various messaging protocols. In the context of this document, protocol conversion refers to the step up/step down conversion needed for Direct Protocols and IHE specifications. Specifically it includes conversion to and from S/MIME/SMTP and IHE based SOAP XML.

- **Provider Directory** (PD) refers to a persistence store with entries that pertain to end users acting as individual providers or other healthcare clinicians. Also stored are entities such as organizations or departments and the relationships between providers and entities.
- **Regional Health Information Organization** (RHIO) is a multi-stakeholder organization responsible for information exchange among stakeholders. RHIO's are also organizations that oversee and govern the exchange of health-related information among organizations according to nationally recognized standards.
- **Secure/Multipurpose Internet Mail Extensions** (**S/MIME)** is a standard for public key encryption and signing of MIME (extended email) data.
- **Security Domain** is defined as the domain, specified by Domain Name(s) taken by a HISP for the control of Direct Addresses, Certificate Common Names, Nodes, and Service Endpoints.
- **Service Endpoints** is defined as a destination address for the receipt of Web Service request messages, usually defined by a Uniform Resource Identifier (URI). In the Direct project a Service Endpoint can also be an email address for the receipt of S/MIME/SMTP based messages (Direct Address).
- **Service Registry** (SR) is a registry which contains the complete definition (messaging framework, protocols, payload, and clinical vocabulary) of services supported by a Node to either send or receive messages.
- **Simple Mail Transfer Protocol (SMTP)** is an Internet standard for electronic mail (e-mail) transmission across Internet Protocol (IP) networks.
- **Simple Object Access Protocol (SOAP)** is a protocol specification for exchanging structured information in the implementation of Web Services in computer networks. It relies on Extensible Markup Language (XML) for its message format, and usually relies on other Application Layer protocols, most notably Hypertext Transfer Protocol (HTTP), for message negotiation and transmission.
- **Trust Anchor** is a public key and associated information that is directly trusted by an application or system to validate digital signatures, including signatures covering other public keys that are signed by the trust anchor.
- **Uniform Resource Identifier (URI)** is a string of characters used to identify a name or a resource on the internet. The URI syntax consists of a URI scheme name (such as "http", "ftp", "mailto" or "file") followed by a colon character, and then by a scheme-specific part.

# 3 System Functions

## 3.1 System Functions

The Statewide Send and Receive patient record includes the following system functions defined in Table

1 below:

Priority Definition:

1 = Mandatory
2 = Optional

**Table 1 Summary of Functional Requirements**

| Function | Priority | Description |
|---|---|---|
| **FR-1 Find Entity Electronic Service Information** | **1** | **A sender queries a Provider Directory Service to determine Organizational Information gathered in the directory. This information includes Direct Address, service endpoints, protocols and payload types accepted by each entity. Information on individual providers associated with the organization may also be obtained.** |
| **FR-2 Find Provider Electronic Service Information** | **1** | **A sender queries a Provider Directory Service to find the intended recipient's Electronic service Information including the Direct Address and to determine the recipient's demographic and professional information. Certificate, service and Organization information may also be obtained.** |
| **FR-2A Retrieve Certificates** | **1** | **A recipient's Direct Certificates are obtained from a Certificate Directory based on their Direct Address.** |
| **FR-3 Send Patient Record** | **1** | **A sender selects the records he wishes to send to the recipient that is selected from a Provider Directory. The records are formatted into an appropriate protocol message and sent via the HISP Direct Services.** |
| **FR-4 Protocol Conversion** | **2** | **Protocol conversion from the sender's framework/protocol to the recipient framework/protocol based on their system protocol capabilities. Framework/Protocols are S/MIME/SMTP and XDR/SOAP.** |
| **FR-5 Deliver Patient Records** | **1** | **Patient's records are delivered to the recipient's service end point, which may not be the recipient's clinical application.** |
| **FR-6 Validate Sender's certificate and Identity** | **1** | **The recipient's system retrieves the message, obtains the providers' certificates, decrypts the message, and verifies the digital signature of the sender.** |
| **FR-7 View Patient Record** | **1** | **The recipient extracts the records from the received message and views the document within the local clinical application.** |

| Function | Priority | Description |
|---|---|---|
| **FR-8 Extract Patient record**<br>**(Optional)** | **2** | **The recipient extracts the records from the received message and views the records within an EHR application. Additionally, the provider can import an XML document directly into the EHR application as discrete data.** |

# 4 Integration Profiles

A Statewide Send and Receive Patient Record Exchange will enable healthcare providers to exchange patient records with other providers. This document describes a detailed technical implementation of the NwHIN Direct framework which can be used to provide this functionality. This framework is augmented with a Provider Directory and a Certificate Directory for use in identifying intended recipients and securing the outgoing message. The Provider Directory allows for demographic based lookup of an intended recipient's Direct Provider Address, a unique identifying value which relates to the provider and to specific organization that the provider is associated with.

The Certificate Directory is needed when the SMTP protocol using S/MIME is used to send a message within Direct. The Certificate directory allows certificate discovery based on the Provider Direct Address, and certificate retrieved from a Certificate Directory is used to perform security operations required by Direct transport and security specifications. These concepts are demonstrated in the Direct Project's reference implementation.

Extensive work has been done in the Direct project focusing mainly on functionalities provided by HISPs. In contrast, the Statewide Send and Receive Patient Records Functional Specification aims to cover end-to-end communication route between edge systems within a HISP domain, as well across HISP domains, enabled by HISP actors representing each domain. This technical specification describes the different actors involved in such end-to-end communications and the functionalities they provide.

## 4.1 Actors

Table 2 below table outlines a list of actors who will interact with the Statewide Send and Receive Patient Record Exchange:

**Table 2 — Actors**

| **Actor Name** | **Transactions** |
|---|---|
| **Provider Directory** | **Provider Information Query Request** |
| **Provider Information Consumer** | **Provider Information Query Request** |
| **Certificate Directory** | **Get Current Certificates** |
| **Certificate Service Consumer** | **Get Current Certificates** |
| **Sending System** | **Provider Information Query Request, Mutual Authentication, ProvideAndRegister or S/MIME/SMTP** |
| **Receiving System** | **Mutual Authentication, ProvideAndRegister or S/MIME/SMTP** |
| **Direct XD* Service** | **Mutual Authentication, ProvideAndRegister, GetLocalEndpoint** |
| **Direct SMTP Service** | **MIME Over SMTP,S/MIME Over SMTP, GetLocalEndpoint, GetPrivateKey** |
| **Direct Configuration Service** | **GetLocalEndpoint, GetPrivateKey** |

## 4.1.1 Actor Details

### 4.1.1.1 Provider Directory

The principal function of the Provider Directory is to provide a mechanism for retrieving the Individual Level Provider or Entity Level Provider Electronic Service Information, including the Direct Address. This address may then be used in routing messages within Health Information Service Provider (HISP) Security Domain. The Direct address is always used in routing a message external to the HISP. The Direct Address is also used to retrieve public certificate for encryption and authentication (digital signature validation). The HISP can also use the address to lookup sender private keys to sign outgoing message for users it represents.

The Provider Directory will also contain other important information related to the providers, their organizations and the services offered by the provider-organizations.

### 4.1.1.2 Provider Information Consumer

A Provider Information Consumer is any system which needs to use the Provider Electronic Service information including the Direct Address or needs to obtain other provider information from the Provider Directory.

### 4.1.1.3 Certificate Directory

A Certificate Directory is a system used to provide the Public Certificate (or Public Key) for organizations and individual providers based on their Provider Direct Address.

### 4.1.1.4 Certificate Service Consumer

A Certificate Service Consumer is any system which needs to use a Public Certificate..

### 4.1.1.5 Sending System

The Sending system is a system which is capable of providing a patient record (e.g. CCD, C32, PDF, etc.) and supplying it to another system. Typically a Sending System is the primary consumer of the Provider Electronic Service information including the Direct Address from the Provider Directory.

### 4.1.1.6 Receiving System

The Receiving systems is a system capable of receiving and consuming a patient record (e.g. CCD, C32, PDF, etc.) from an external system

### 4.1.1.7 Direct XD* Services

Direct XD* Services are services capable of satisfying the Direct XD* specification. This means they are capable of handling an IHE ProvideAndRegister transaction with the minimal metadata compliment as described in the Direct specification. When messaging takes place between SMTP and XDR based clients, Protocol Conversion is performed by the Direct XD* Services.

Please See: XDR and XDM for Direct Messaging Specification.

### 4.1.1.8 Direct SMTP Service

The Direct SMTP Service is capable of receiving and relaying email messages based on the SMTP and the Provider Direct Address. It must be capable of receiving messages in S/MIME and validating and decrypting them using the Certificate Directory and the Direct Configuration Services. It may be capable of routing such messages to an EHR/XD application using the Direct XD* services. Other protocols may also be used within the HISP Security Domain. Direct SMTP Service must be capable of receiving a patient record in MIME over standard secure email protocols. The Direct SMTP Service must be capable of allowing secure email retrieval by recipients.. The service must be capable of encrypting a message and signing it (S/MIME) using the recipient Provider Direct Address, the Certificate Directory and the Direct Configuration Services. If the destination address is external to its domain, the service must be capable of forwarding this S/MIME message over SMTP to another Direct SMTP Service, discovered using DNS.

The Direct SMTP service must conform to transport and security requirements for Direct as specified in detail in the Applicability Statement of Secure Health Transport, and the S&I Framework modular specification Direct Transport and Security Specification SMTP and SMIME. It must be able to satisfy certificate discovery requirements as specified in the Certificate Discovery for Direct Project Implementation Guide and Direct Certificate Discovery Implementation Guide. (Modular specification).

### 4.1.1.9 Direct Configuration Service

The Direct Configuration Service in the Direct reference implementation maintains and provides Private Keys to the Direct SMTP Service, based on Provider Direct Address. All interfacing is internal to the HISP Security Domain. It is capable of providing internal service endpoints for providers that wish to receive non-SMTP Direct messages at an EHR system within the security domain. Endpoints are indexed by Direct Address. The Direct reference implementation supports XDR ProvideAndRegister requests.

The Configuration Service is also capable of serving as a local Certificate Directory, holding provider and entity level public certificates and trust anchors that are available within the HISP Security Domain. This service is entirely internal to the HISP and is not intended to be the same as the Certificate Directory Service.

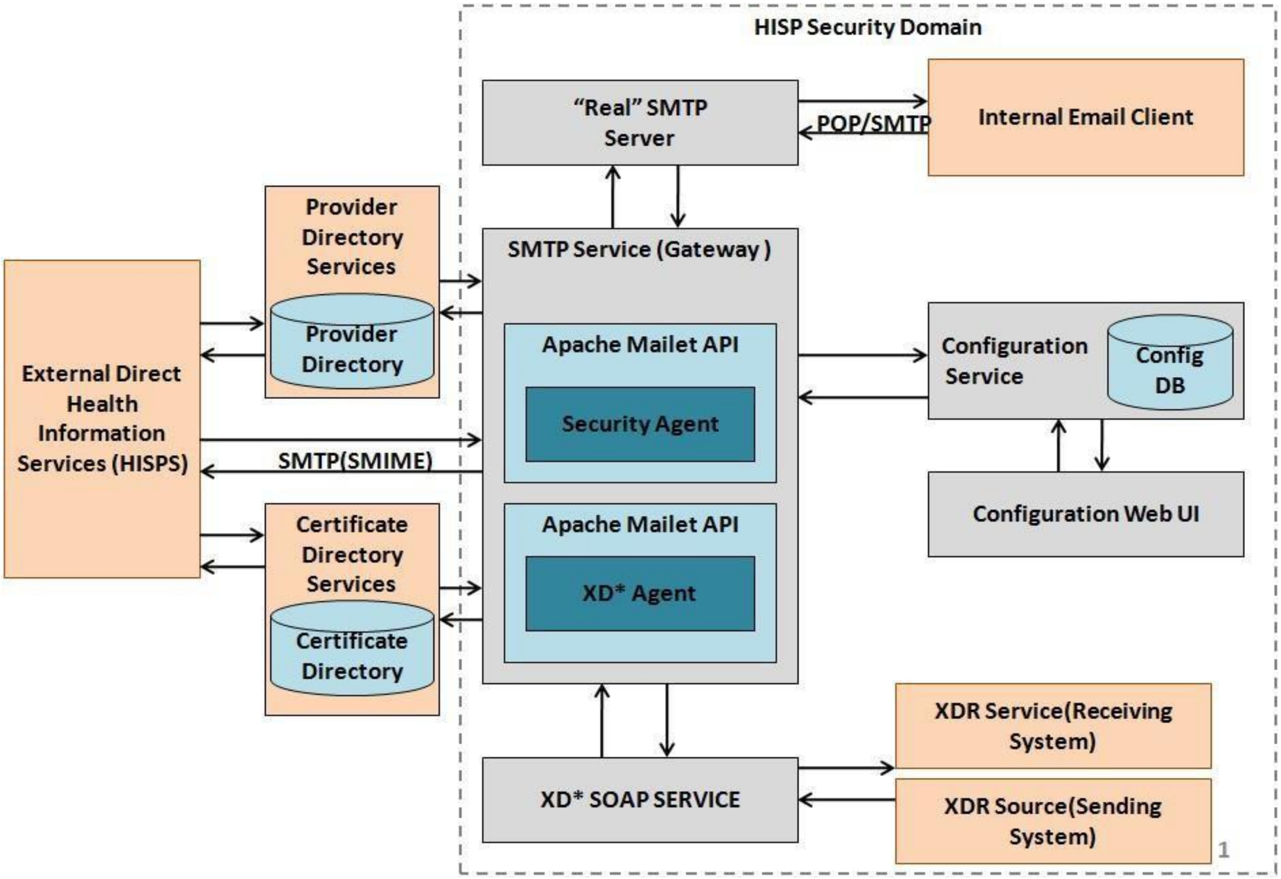
## 4.2 High Level Architecture

Figure 1 below is a high-level architecture diagram of a Direct HISP, as in the Direct reference implementation, including the Provider Directory Services and the Certificate Directory Services that are not part of the Direct Reference Implementation. XDR/SOAP based services may be used as part of Direct by EHR Systems within the HISP Security Domain, as in the Reference Implementation. SMTP can be used within the security domain by SMTP based clients (e.g. email clients or SMTP based EHRs). Other protocols may be used within the HISP Security Domain. Transformation between XDR/SOAP and S/MIME/SMTP is specified by the Direct Project.

When messaging is external to the HISP Security Domain, Direct requires the use of the “S/MIME over

SMTP"(S/MIME/SMTP). For messages originating from and XDR sending system, this involves transforming XDR to XDM messages and wrapping the XDM in S/MIME. This allows for the use of a single protocol across HISPs and for the use of Provider Direct Address based public certificate discovery as the single security mechanism across HISPs. "S/MIME/SMTP" messages coming in from an external HISP can either be forwarded to SMTP clients or transformed to XDR for EHR/XD system consumption. If other protocols are used, transformation to and from SMTP must be supported.

**Figure 1 High-Level Architecture Diagram of a Direct HISP**



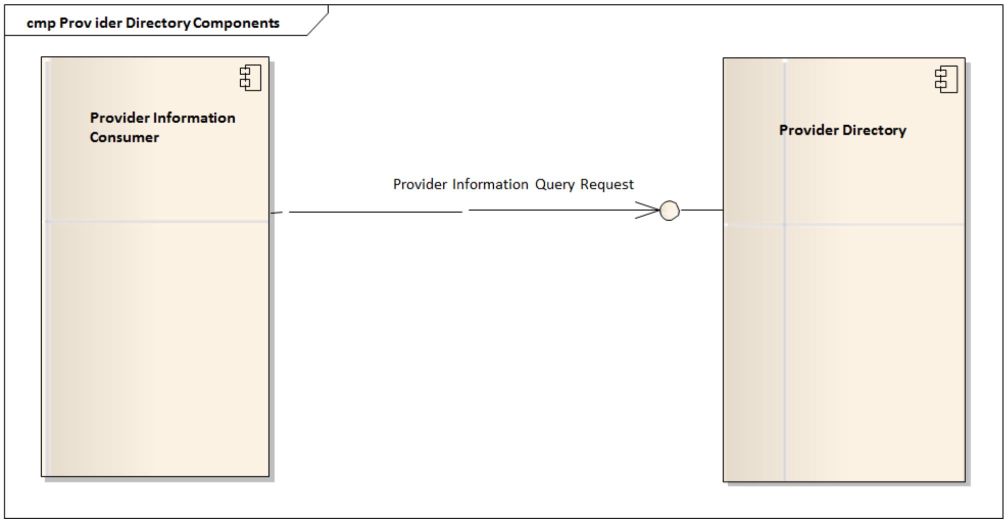
## 4.3 Activity Diagrams with Transactions

The following diagrams break down the high level architecture into a series of activities between actors based on a standard set of transactions. These Activities are based on the System Functions as detailed in the "Statewide Send and Receive Patient Records Functional Specification".

### 4.3.1 Find Electronic Address

This is the standard Provider Directory activity. Details of the Request and response messages are provided later in this document. Details on this transaction can be found in the Detailed Interaction Find Electronic Address section of this document. Figure 2 below depicts the Provider Directory Components.

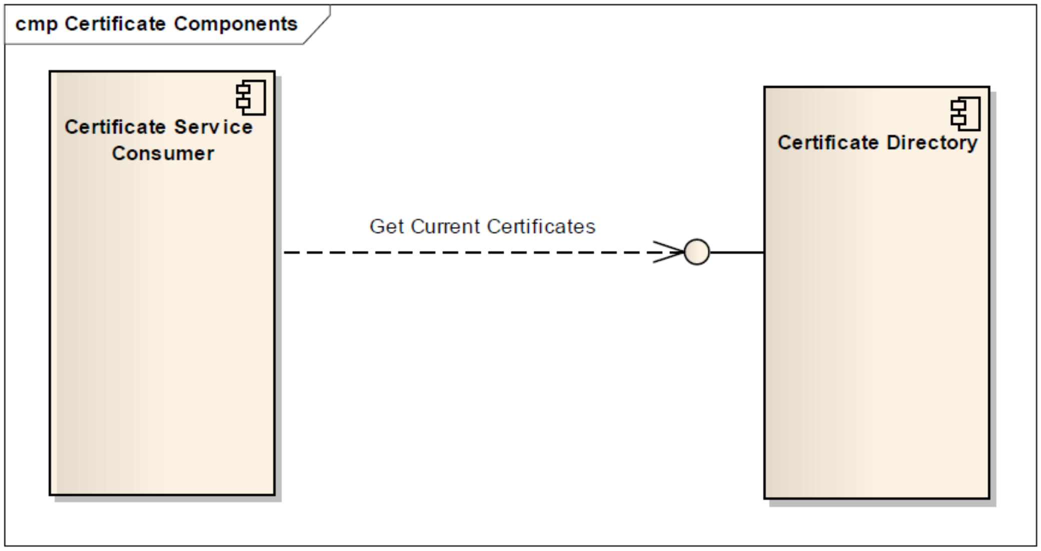
**Figure 2 Provider Directory Components**


cmp Provider Directory Components
Provider Information Consumer
Provider Directory
Provider Information Query Request

## 4.3.2 Retrieve Certificates

This is the standard certificate activity for the discovery of current certificates by a generic consumer, based on Provider Direct Address. Details on this transaction can be found in the Detailed Interaction Retrieve Certificates section of this document. Figure 3 below depicts the Certificates Components.

**Figure 3 Certificate Components**



## 4.3.3 Send Patient Records within the HISP Domain

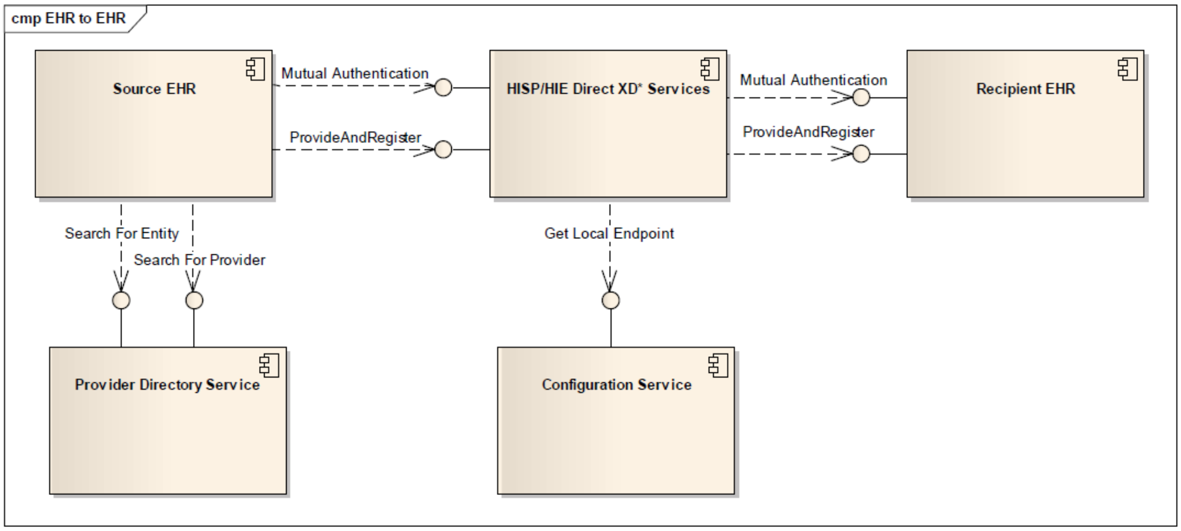
Patient Record messages sent between SMTP based clients within the HISP security domain are typically managed by the Direct SMTP Service as standard email. Communication between the non-S/MIME SMTP based client and the Direct SMTP Service must use TLS. EHRs may implement SMTP based client functionality or XDR functionality if allowed by the HISP. When messaging takes place between

SMTP and XDR based clients, Protocol Conversion is performed by the HISP/HIE Direct XD* Service. Protocols other than XDR/SOAP may also be used within the HISP if conversion is provided. Alternatively, EHRs may incorporate the capability to send and receive S/MIME/SMTP messages directly without the use of the HISP's Direct SMTP Service. In order to participate in this manner an EHR must comply with the Direct Applicability Statement.

The activity in Figure 4 below describes the XDR to XDR SOAP based sending of a patient record within a HISP Security Domain, as in the Reference Implementation. The standard TLS Bidirectional authentication mechanism is used. By allowing the HISP services to act as an intermediary between the EHRs, several simplifications for the EHRs are gained. First they do not need to understand anything about the recipient and author other than their Provider Direct Address. No knowledge of destination protocol specifics or endpoint is needed; the HISP is responsible for the routing information.

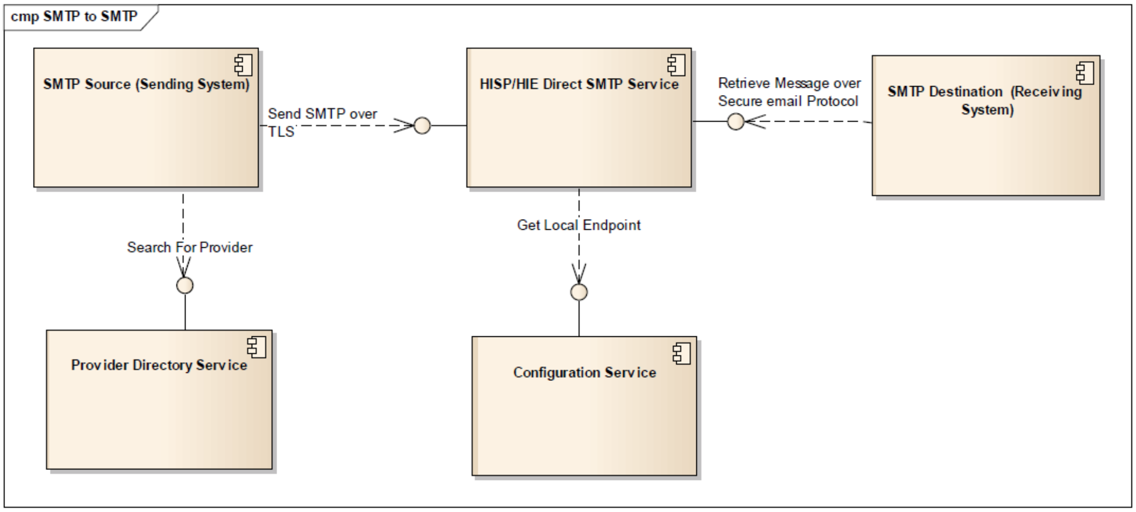
Second, the security model for the EHRs is simplified. They only need to share certificates with the HISP, and they do not need to share a CA trust model or implement out-of-band certificate exchange with any of the other EHRs within the domain.

**Figure 4 XDR to XDR SOAP Sending within a HISP Security Domain**



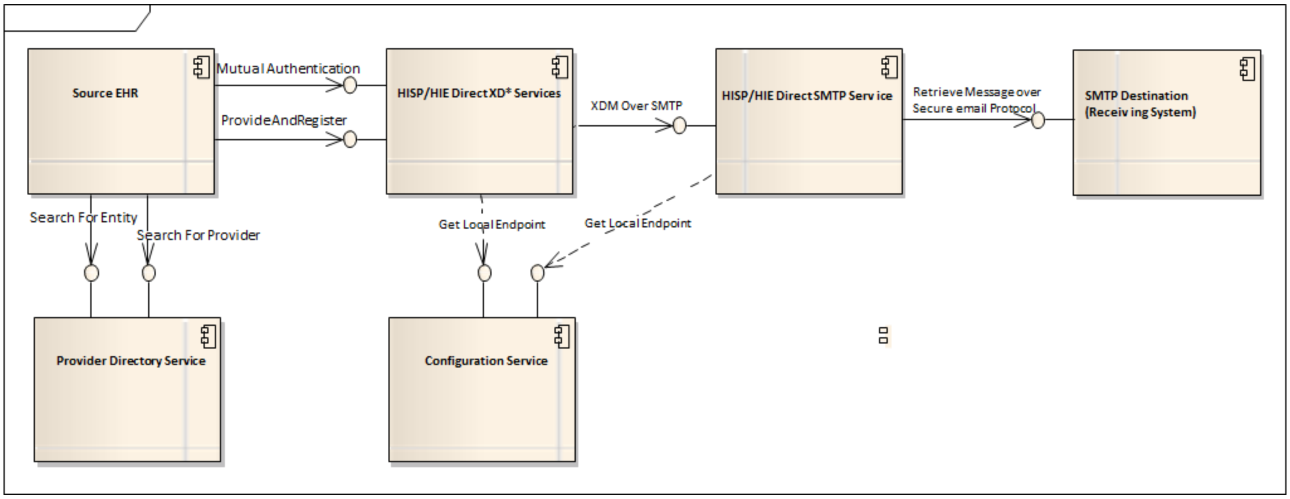
The following activity in Figure 5 below describes SMTP to SMTP based transfer of a patient data message within a HISP. The SMTP source and destination can be an EHR or a simple email client which handles secure protocols. The Provider Directory must be queried to obtain the Provider Direct Address if not already known by the SMTP Source. The Configuration Service query is made to ensure that the destination is SMTP and does not require protocol conversion.

**Figure 5 SMTP to SMTP Sending within a HISP Security Domain**


cmp SMTP to SMTP
SMTP Source (Sending System)
Send SMTP over TLS
HISP/HIE Direct SMTP Service
Retrieve Message over Secure email Protocol
SMTP Destination (Receiving System)
Search For Provider
Get Local Endpoint
Provider Directory Service
Configuration Service

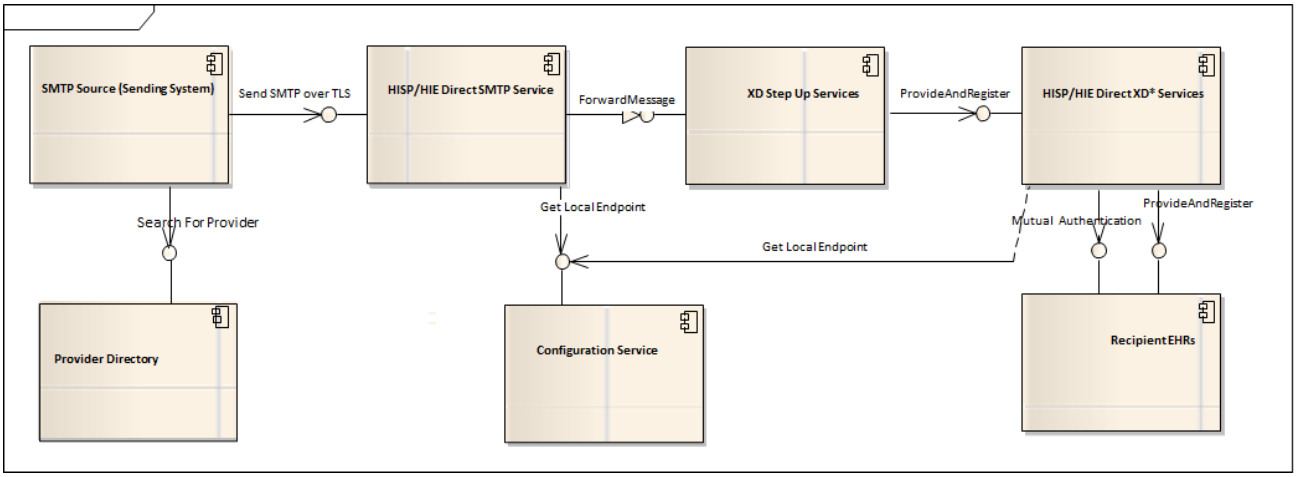
The following activity in Figure 6 below describes XDR to SMTP based transfer of a patient data message within a HISP. The Direct XD* Services recognize the recipient Provider Direct Address is a local SMTP endpoint and wraps the ProvideAndRegister request in an XDM package. It is forwarded to the Direct SMTP Service as an SMTP call using MIME. The Direct SMTP Service then routes the message locally to the appropriate recipient. The XDR to SMTP conversion performed in this transaction is described in detail in the Protocol Conversion section below.

**Figure 6 XDR to SMTP Sending within a HISP Security Domain**


Source EHR
Mutual Authentication
ProvideAndRegister
HISP/HIE Direct XD* Services
XDM Over SMTP
HISP/HIE Direct SMTP Service
Retrieve Message over Secure email Protocol
SMTP Destination (Receiving System)
Search For Entity
Search For Provider
Get Local Endpoint
Get Local Endpoint
Provider Directory Service
Configuration Service

The following activity in Figure 7 below describes SMTP to XDR based transfer of a patient data message within a HISP. The Direct SMTP Services recognize the recipient Provider Direct Address is XDR/SOAP based, and the message is converted by the XD* Step Up Service (internal to the mail service) into a ProvideAndRegister message. This ProvideAndRegister message is sent to the HISP Direct XD* services, which determines the XDR/SOAP final endpoint based on the Direct Configuration Service. The message is sent using TLS with Bidirectional Authentication to the final endpoint of the recipient EHR. The SMTP to XDR conversion performed in this transaction is described in detail in the Protocol Conversion section below.

**Figure 7 SMTP to XDR Sending within a HISP Security Domain**



## 4.3.4 Sending Patient Records Outside of the HISP Domain

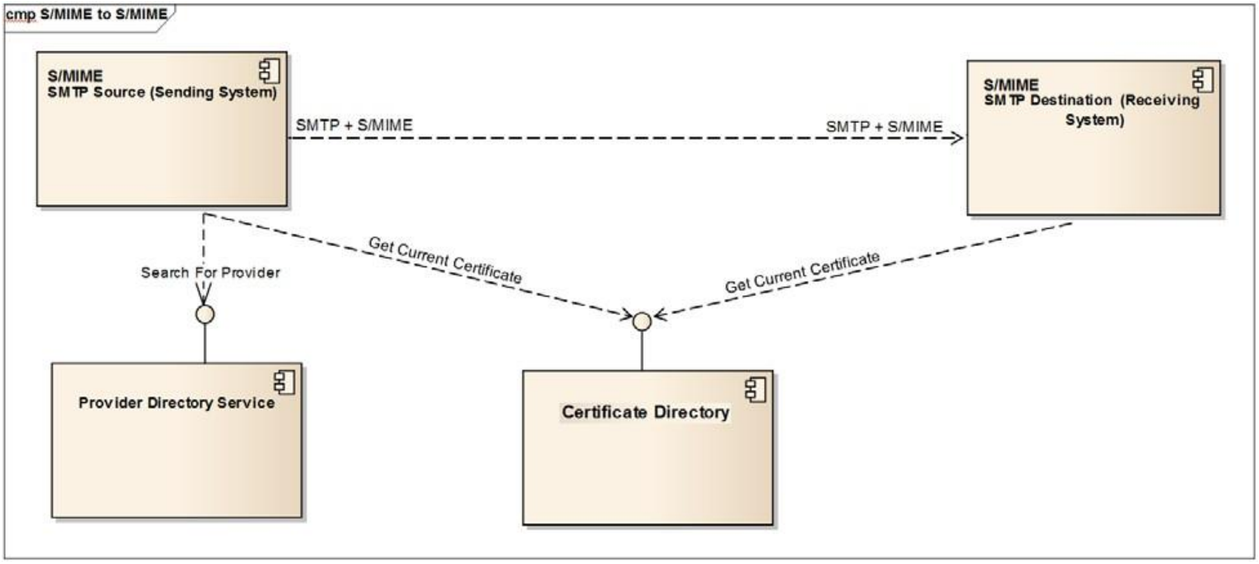
These activities describe the sending of a patient record external to a HISP Security Domain but within a HISP Affinity Domain. In the first case, S/MIME/SMTP is the only protocol used across the HISPs. In the second case, one or both the sending and receiving systems involved are XD based. No matter what, when traversing from HISP to HISP, the S/MIME/SMTP Direct Backbone is used. From an XD based EHR point of view, nothing actually changes, the Direct address is obtained and HISP Direct XD* Service is contacted in the same way as within the HISP. However behind the scenes, a process called Protocol Conversion takes place.

### 4.3.4.1 SMTP Only

The activity in Figure 8 below describes S/MIME-based SMTP to SMTP transfer of a patient data message across HISPs within a HISP affinity domain. The SMTP S/MIME source and destination can be an EHR or an email server which handles S/MIME and SMTP protocols and complies with the Direct Applicability Statement. The Provider Directory must be queried to obtain the Provider Direct Address if not already known by the SMTP Source. An EHR capable of S/MIME-based SMTP transfer must be able to handle plain text documents, CCD attachments, PDF attachments and XDM. The sending system should ensure that the message is in a format that can be processed by the receiving system, both in terms of the type of document as well as the SMTP+S/MIME protocol. The receiving system MUST provide a failure disposition response when an unacceptable document is received. This destination-capability information may be obtained from the query to the Provider Directory Service or may require a separate query to a Configuration Service.The SMTP to SMTP transaction across HISPs must conform to requirements as defined in the following specifications:

- Applicability Statement of Secure Health Transport;
- Direct Transport and Security Specification SMTP and SMIME

**Figure 8 S/MIME-based SMTP to SMTP Sending within a HISP Affinity Domain**

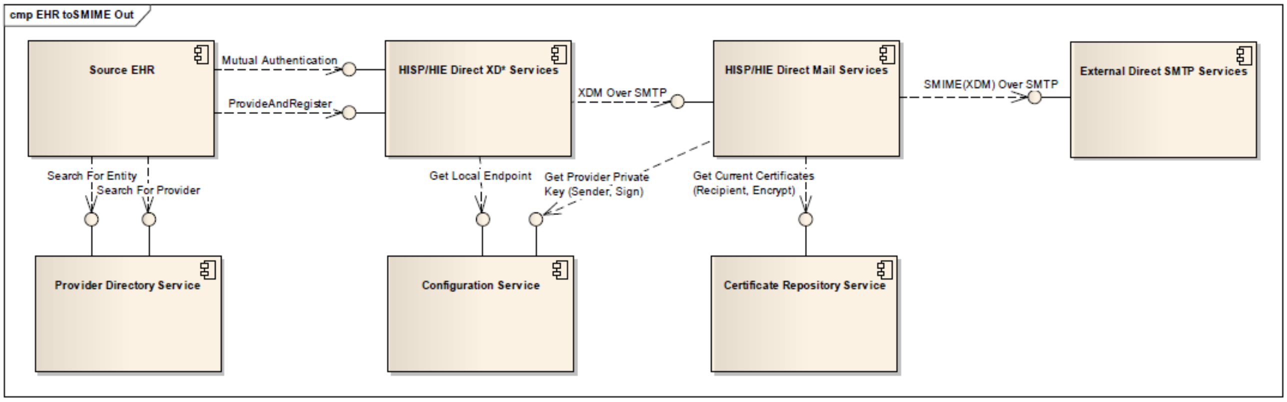


## 4.3.4.2 Protocol Conversion

Protocol Conversion is a complexity that is hidden from the EHR/XD systems and accomplished by the HISPs. Upon discovering that a recipient Provider Direct Address belongs to an external HISP, or is SMTP based, the protocol conversions between XDR/SOAP and S/MIME/SMTP are leveraged. The conversion is divided into two separate activities in order to reduce the complexity on either side of the transaction. These same activities are used when a SOAP system transacts with an SMTP based client within the HISP security domain.

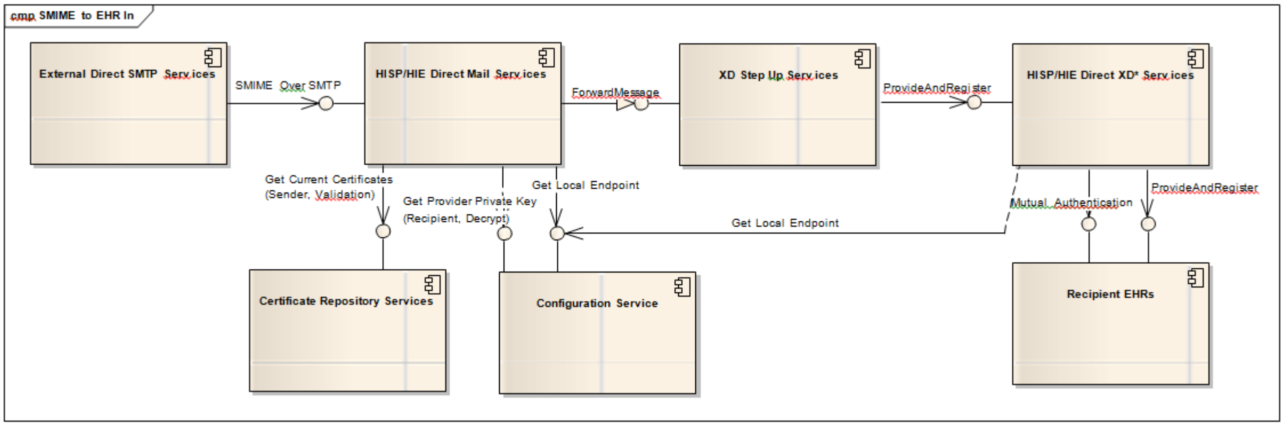
As depicted below in Figure 9, the first activity describes the XDR sending side of the protocol conversion. When the Direct XD* Services recognize a Provider Direct Address is not within its security domain or is a local SMTP endpoint, the ProvideAndRegister request message is wrapped in an XDM format. It is forwarded to the Direct SMTP Service as an SMTP call using MIME. The Direct SMTP Service uses the author’s Provider Direct Address to obtain the Private Key for signature from the Direct Configuration Service, and gets the recipient’s public certificate from the Certificate Repository for encryption. The Direct SMTP Service verifies trust and validity of the recipient’s certificate and uses S/MIME to sign and encrypt the MIME (XDM) message per the Direct Applicability Statement before forwarding it to the recipients Direct SMTP Service at another HISP. If the destination is local, the MIME message is routed internally.

**Figure 9 Send of XDR to SMTP Protocol Conversion**


cmp EHR toSMIME Out
Source EHR
Mutual Authentication
ProvideAndRegister
HISP/HIE Direct XD* Services
XDM Over SMTP
HISP/HIE Direct Mail Services
SMIME(XDM) Over SMTP
External Direct SMTP Services
Search For Entity
Search For Provider
Get Local Endpoint
Get Provider Private Key (Sender, Sign)
Get Current Certificates (Recipient, Encrypt)
Provider Directory Service
Configuration Service
Certificate Repository Service

The second activity as depicted in Figure 10 below, is receiving the S/MIME based message by the Direct SMTP Service. Upon reception, the recipients Private Key is obtained from the Direct Configuration Service for decryption of the message. The sender's public certificate can be obtained from the Certificate Repository Service for Validation of the sender's signature and trust  If the destination endpoint is XDR/SOAP based, the message is converted by the XD* Step Up Service (internal to the mail service) into a ProvideAndRegister message.  This ProvideAndRegister message is sent to the HISP Direct XD* services, which determines the XDR/SOAP final endpoint based on the Direct Configuration Service. The message is sent using TLS with Bidirectional Authentication to the final endpoint of the recipient EHR.

**Figure 10 Receipt of S/MIME based Message by Direct SMTP Service**


cmp SMIME to EHR In
External Direct SMTP Services
SMIME Over SMTP
HISP/HIE Direct Mail Services
ForwardMessage
XD Step Up Services
ProvideAndRegister
HISP/HIE Direct XD* Services
Get Current Certificates (Sender, Validation)
Get Provider Private Key (Recipient, Decrypt)
Get Local Endpoint
Get Local Endpoint
Mutual Authentication
ProvideAndRegister
Certificate Repository Services
Configuration Service
Recipient EHRs

The protocol conversion describe above must conform to the XDR and XDM for Direct Messaging Specification, and the Direct Message Conversion Specification XDR and XDM from S&I Framework Modular Specification for Direct Exchange project.

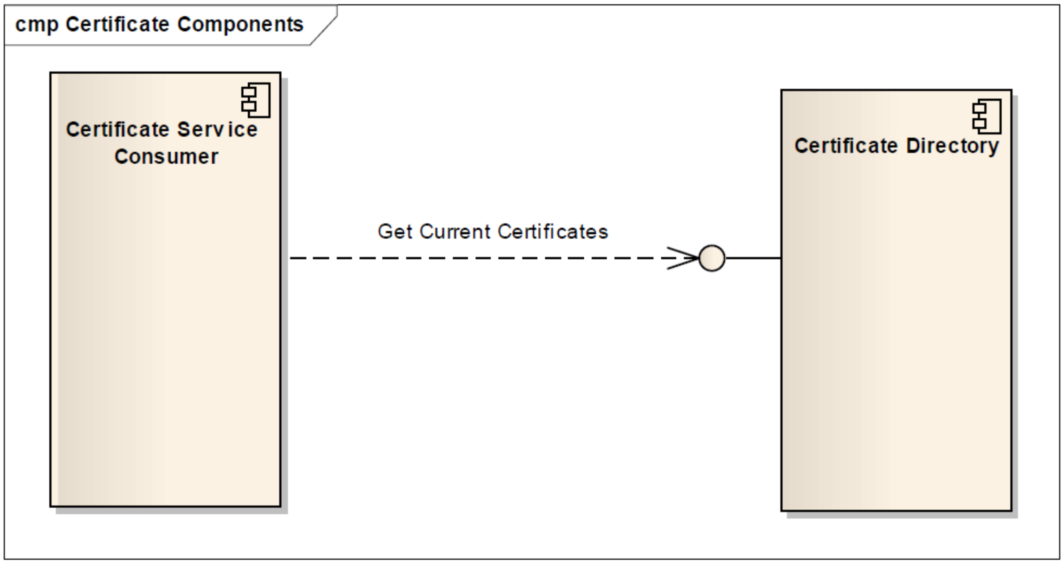
## 4.3.5 Validate Sender/Receiver Certificate and Trust

The activity for validating trust in a sender or recipient , which uses the Get Current Certificate operation to retrieve the public certificate of the sender or recipient from the Certificate Directory, in a HISP to

HISP SMTP transaction is depicted in Figure 11 below. HISPs sending and receiving Direct messages must verify trust and validity of sender/recipient certificate as required by the Direct Applicability Statement. If trust cannot be verified, a sending system must not send the message and a receiving system must reject the message. HISPs receiving Direct messages must verify the validity of the message by checking the S/MIME signature, which originated with the sender's private key. If the two do not agree, the message is rejected.

In a HISP internal XDR transaction, as in the Reference Implementation, the sender or receiver system is verified using TLS mutual authentication (per IHE). This verifies the sender's (client) certificate is in a recipient's application trust store and that the corresponding private key was used to sign the message. Other internal HISP protocols must guarantee the same level of security.

**Figure 11 Validating Sender/Receiver Certificate and Identity**



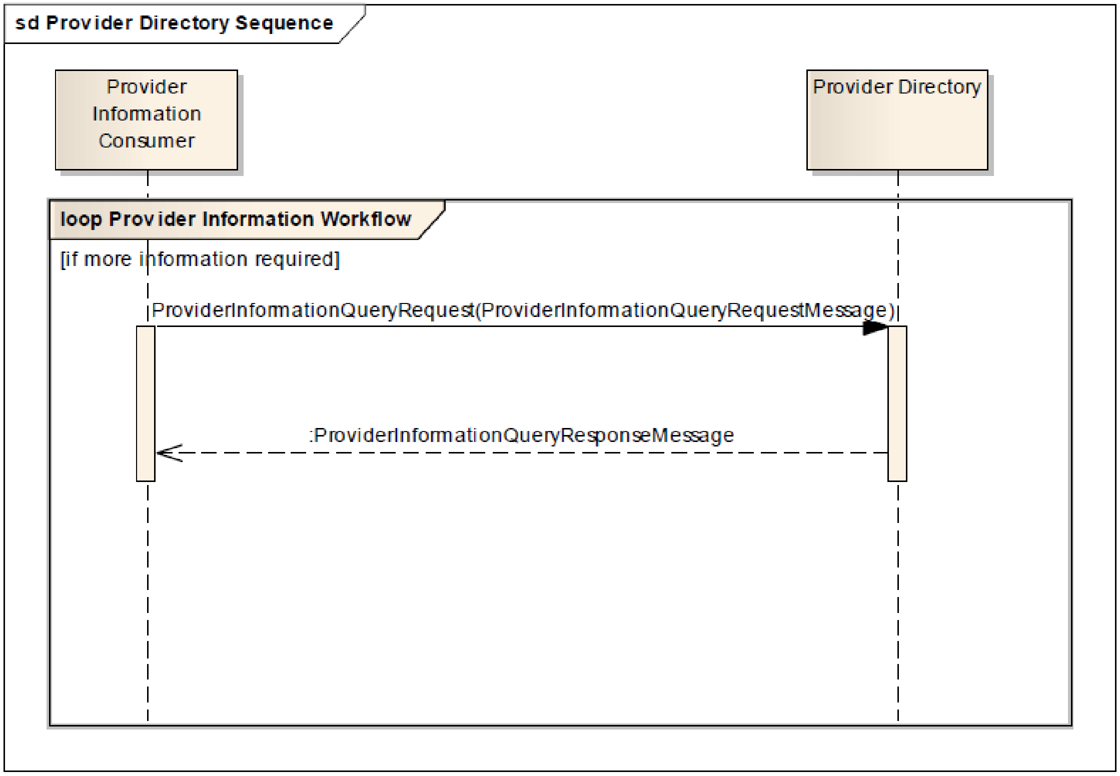
# 5 Detailed Interaction

## 5.1 Detailed Interactions – Transactions

### 5.1.1 Find Electronic Address

The Sequence depicted in Figure 12 below defines the Query/Response for interacting with a Provider Directory.

**Figure 12 Provider Directory Sequence to Find an Electronic Address**


sd Provider Directory Sequence
Provider Information Consumer
Provider Directory
loop Provider Information Workflow
[if more information required]
ProviderInformationQueryRequest(ProviderInformationQueryRequestMessage)
:ProviderInformationQueryResponseMessage

#### 5.1.1.1 Triggers

System requests Provider Information.

#### 5.1.1.2 Transaction Message Details

The Provider Information Workflow loop implies that the ProviderInformationQueryRequest may be made multiple times with various query and response parameters in order to accommodate consumer workflow.

The EHR|HIE Interoperability Workgroup has come to the determination that two different implementations of the ProviderInformationQueryRequest shall be allowed. The first is the IHE based Healthcare Provider Directory (HPD) Framework. The specific HPD functionality is described in detail in the "IHE ITI Technical Framework Supplement Healthcare Provider Directory". The second is an augmented version of this framework that this group calls **"HPD Plus**".

The major difference between HPD and HPD Plus is the requirement for a specific persistence mechanism. The HPD framework specifically calls out an LDAP schema and implies an LDAP server for persistence. The idea behind HPD Plus is to decouple the HPD interface (WSDL and XML schema) from a specific persistence mechanism. This allows, for example, an implementer to use a Relational Database model without an LDAP server.

In order for HPD to implement the full S&I framework provider data requirements, the IHE HPD LDAP schema must be enhanced to cover the missing fields, both query and response types. HPD Plus defines how S&I fields not covered by HPD are mapped to an extended LDAP schema to allow for query and response transactions conforming to the HPD interface. An HPD Plus implementation should implement all of the S&I provider directory fields (both query and response types) using its persistence mechanism of choice.

Note that IHE ITI Change Proposal 601 "Extend HPD memberOf to support attributes on relationship" has adopted most of the LDAP schema extension defined in HPD Plus. The change proposal was recently affirmed. There are still some attributes not covered by HPD that' in the S&I Framework data model. Those should be resolved in future versions of HPD.

The ProvideInformationQueryRequest operation is found in the ProviderDirectory WSDL section of this document. The WSDL's ProviderInformationQueryRequest message and the ProviderInformationQueryResponse message are composed of the BatchRequest and BatchResponse elements. These Elements and their detailed breakdown are found in the DSMLv2 XML schema section of this document. Binding to the DSMLv2 schema would be considered standard HPD or HPD Plus, again the only difference in the two is in the Provider Directory implementation.

There is a second operation for data maintenance found in the Provider Directory WSDL. The implementation of the ProviderInformationFeedRequest operation is considered out of scope for this document at this time.

### 5.1.1.3 Data Mapping

The Provider Directory Data Mapping section of this document provides a mapping between current HPD fields and the fields as required by the ONC S&I Framework "Query for Electronic Service Information including Electronic Address Use Case". These S&I Framework fields make up the HPD Plus attributes. The S&I Framework attributes are labeled as to whether or not the fields may be part of the ProviderInformationQueryRequest message, the ProviderInformationQueryResponse message, or both

This document's section on HPD Plus Functionality describes the functional HPD schema elements and how an HPD Plus Provider Directory System should respond to filter values and other fields.

### 5.1.1.4 Sample Messages

The following sample messages are based on a Provider Directory that has the following information for Mark Smith.

- He has a Primary Name – Marcus Smith, and two other names, Mark Smith, and Mark Smithson.
- He works for two Hospitals, St Mary's (which is also known as Saint Mary's) and Sinai Hospital.

#### 5.1.1.4.1 HPD with DSMLv2 Sample Transaction

##### 5.1.1.4.1.1 Request

```
<soap-env:Envelope xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/">
    <soap-env:Body>
      <dsml:batchRequest xmlns:dsml="urn:oasis:names:tc:DSML:2:0:core"
        xmlns:xsd="http://www.w3.org/2001/XMLSchema"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" >
        <dsml:searchRequest  dn="O=HPDTEST,DC=HPD,C=US" scope="wholeSubtree" derefAliases="neverDerefAliases"
        sizeLimit="100">
          <dsml:filter>
            <dsml:approxMatch name="displayName">
              <dsml:value>
                Mark Smith
              </dsml:value>
            </dsml:approxMatch>
          </dsml:filter>
          <dsml:attributes>
            <dsml:attribute name="HcPracticeLocation"/>
            <dsml: attribute name="HcRegisteredAddr"/>
            <dsml: attribute name="hpdProviderPracticeAddress"/>
            <dsml: attribute name="HcIdentifier"/>
            <dsml: attribute name="HcSpecialisation"/>
            <dsml: attribute name="givenName"/>
            <dsml: attribute name="cn"/>
            <dsml: attribute name="memberOf"/>
          </dsml:attributes>
          </dsml:searchRequest>
        </dsml:batchRequest>
      </soap-env:Body>
    </soap-env:Envelope>
```

##### 5.1.1.4.1.2 Response

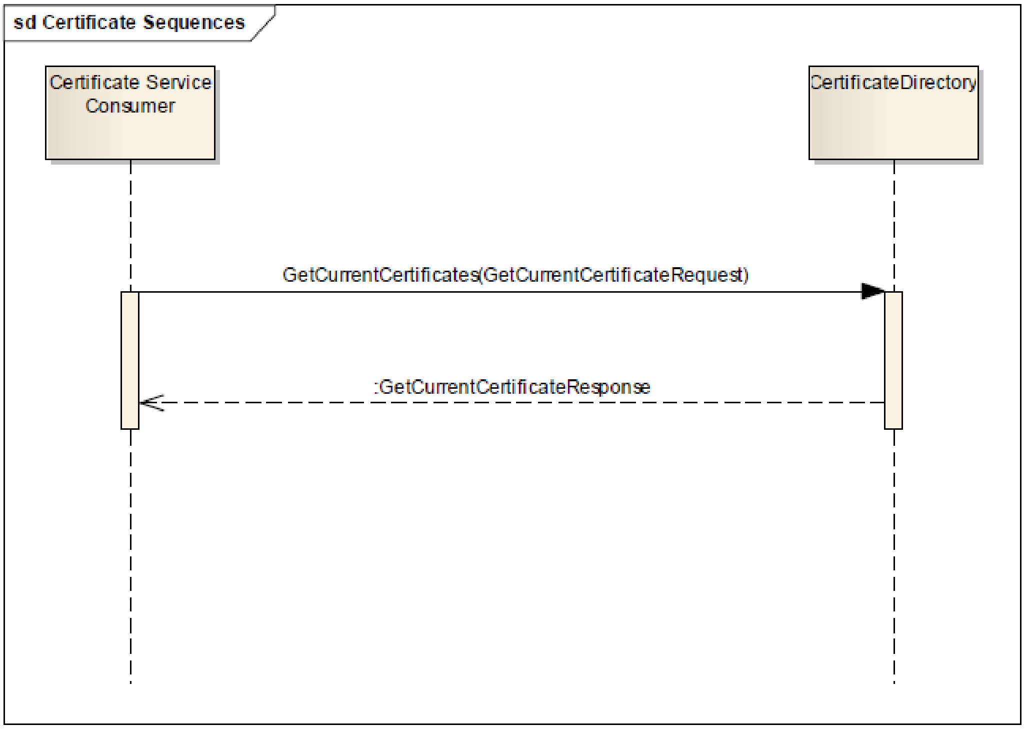
```
<SOAP-ENV:Body>
  <batchResponse  xmlns="urn:oasis:names:tc:DSML:2:0:core">
    <searchResponse>
      <searchResultEntry dn="uid=hpdtest:5,ou=HcProfesssional,O=HPDTEST,DC=HPD,C=US">
        <attr name="cn">
          <value>
            Marcus Smith
          </value>
          <value>
            Mark Smithson
          </value>
          <value>
            Mark Smith
          </value>
        </attr>
        <attr name="HcIdentifier">
          <value>
            1.89.11.00.123:HospId:786868:Active
            </value>
          </attr>
        <attr name="HcSpecialisation">
          <value>
         2.16.840.1.113883.4.340::11:Internal Medicine

            </value>
          <value>
            2.16.840.1.113883.4.340::46:Endocrinology
            </value>
          </attr>
        <attr name="memberOf">
          <value>
            cn=Baltimore General Hospital,ou=Relationship,O=HPDTEST,DC=HPD,C=US
            </value>
          <value>
            cn=hpdTestGroup1,ou=Relationship,O=HPDTEST,DC=HPD,C=US
            </value>
          </attr>
        <attr name="givenName">
          <value>
            Mark Smith
            </value>
          </attr>
        </searchResultEntry>
       <searchResultDone>
         <resultCode/>
           code="0"
        </searchResultDone>
       </searchResponse>
   </batchResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

### 5.1.2 Retrieve Certificates

Figure 13 below depicts the sequence to request and receive a Digital Certificate from the Certificate Directory.

**Figure 13 Sequence to Retrieve Digital Certificates From Certificate Directory**


sd Certificate Sequences
Certificate Service Consumer
CertificateDirectory
GetCurrentCertificates(GetCurrentCertificateRequest)
:GetCurrentCertificateResponse

### 5.1.2.1 Triggers

System requests Digital certificate using Direct Address.

### 5.1.2.2 Transaction Message Details

The Certificate Retrieve mechanism has been specified by the ONC and S&I Framework's Certificate Discovery for Direct project.

- Detailed specification on Certificate Discover for Direct can be found in the following specifications: Applicability Statement of Secure Health Transport;
- Certificate Discovery for Direct Project Implementation Guide;
- Direct Certificate Discovery Implementation Guide. (Modular specification);
- Conforming systems MUST be able to discover certificates using both the DNS as specified in Section 5 of the applicability statement and LDAP as described by the S&I Framework Certificate Discovery for Direct Project Implementation Guide. The main steps of Direct Certificate Discovery are:Using the known Direct address, the consumer submits a DNS query for a CERT resource record;
- If a valid digital certificate for the individual is returned then that certificate is used;
- Else the consumer submits a DNS query for a CERT resource record using the organization Direct domain;
- If a valid digital certificate for the organization Direct domain is returned then that certificate is used;
- Else the consumer submits a DNS query for a SRV resource record for the LDAP service;
- If a valid LDAP resource it returned then the consumer submits the Direct address to that service with anonymous binding;
- If a valid digital certificate is returned then that certificate is used;
- Else if no usable LDAP SRV resource record is returned or no valid digital certificate is returned then the consumer must use some other approach to obtain the certificate of interest.

Other certificate discovery methods in addition to DNS and LDAP may be supported as well.

### 5.1.3 Send Patient Record within HISP Domain

Figure 14 below depicts the sequence for XDR-to-XDR sending of patient records within the HISP's domain, Figure 15 depicts the sequence for SMTP-to SMTP sending of patient records within the HISP's domain.

**Figure 14 XDR-to-XDR Send Patient Record within HISP Domain**

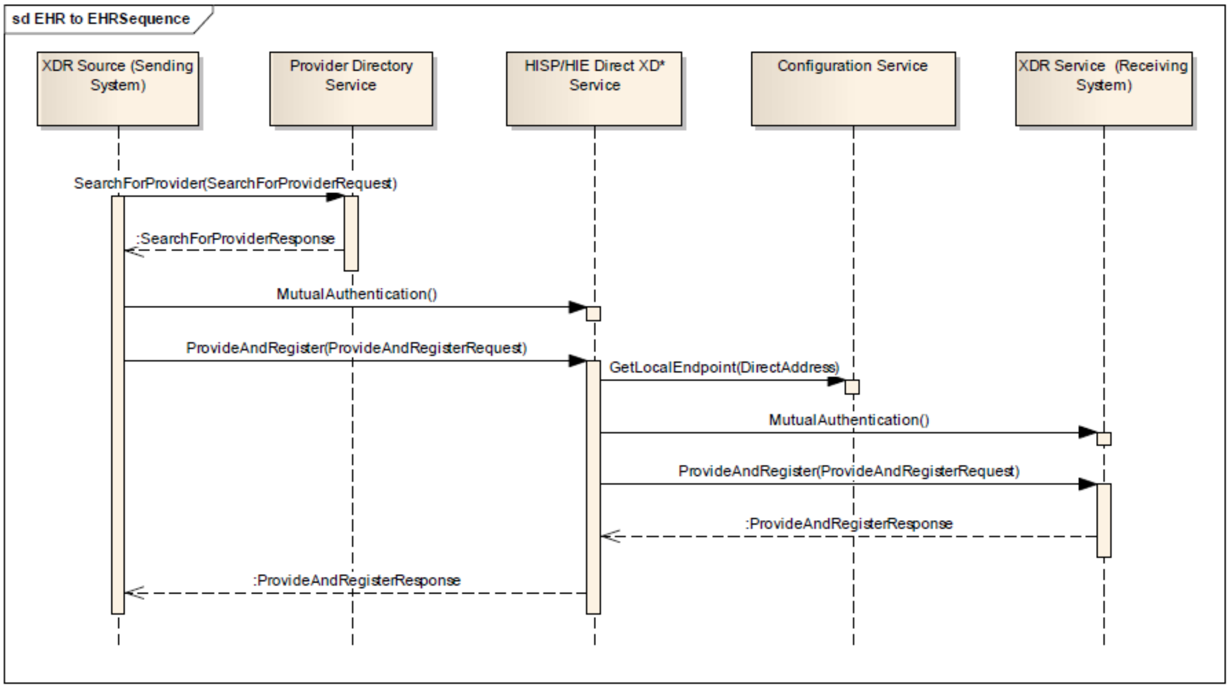
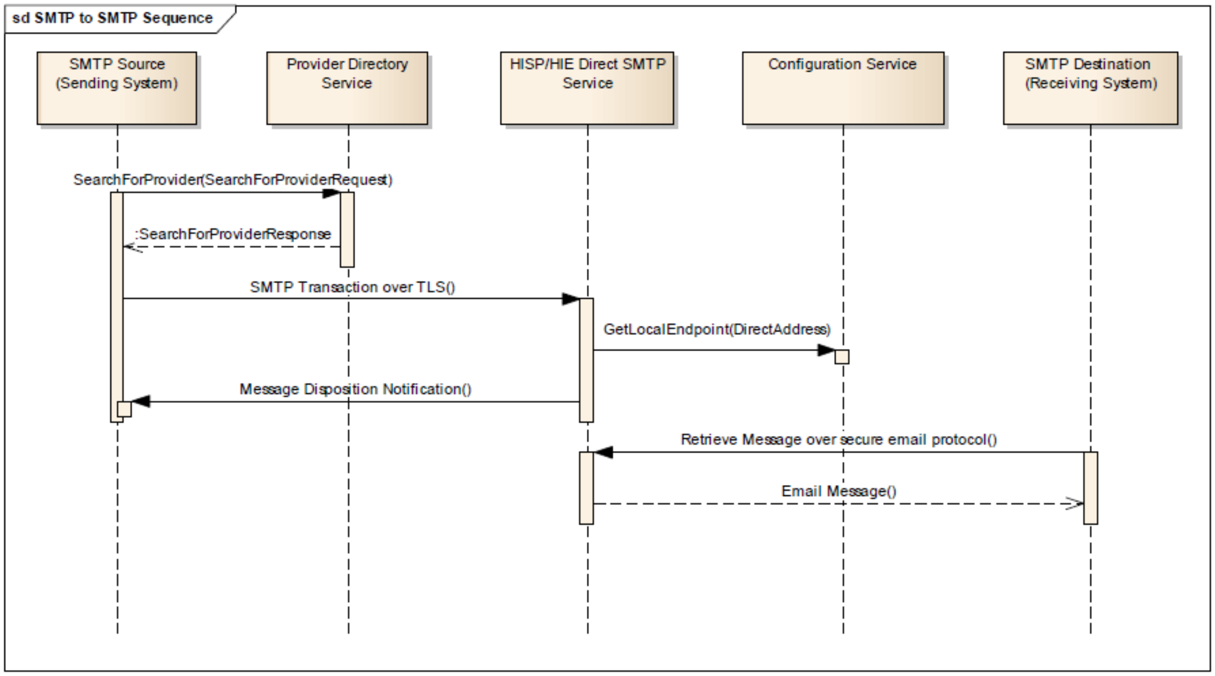

sd EHR to EHRSequence
XDR Source (Sending System)
Provider Directory Service
HISP/HIE Direct XD* Service
Configuration Service
XDR Service (Receiving System)
SearchForProvider(SearchForProviderRequest)
:SearchForProviderResponse
MutualAuthentication()
ProvideAndRegister(ProvideAndRegisterRequest)
GetLocalEndpoint(DirectAddress)
MutualAuthentication()
ProvideAndRegister(ProvideAndRegisterRequest)
:ProvideAndRegisterResponse
:ProvideAndRegisterResponse

**Figure 15 SMTP-to-SMTP Send Patient Record within HISP Domain**


sd SMTP to SMTP Sequence
SMTP Source (Sending System)
Provider Directory Service
HISP/HIE Direct SMTP Service
Configuration Service
SMTP Destination (Receiving System)
SearchForProvider(SearchForProviderRequest)
:SearchForProviderResponse
SMTP Transaction over TLS()
GetLocalEndpoint(DirectAddress)
Message Disposition Notification()
Retrieve Message over secure email protocol()
Email Message()

The transaction sequences of XDR-to-SMTP and SMTP-to-XDR transactions within a HISP domain are similar to those depicted in the Protocol Conversion section below and are therefore not repeated here.

### 5.1.3.1 Triggers

Sending System sends a patient record to a receiving System that is internal to the HISP domain.

### 5.1.3.2 Transaction Message Details

The messages in an XDR transaction are the IHE XDR ProvideAndRegister request and response.
For SMTP based transactions the message is an Internet Message Format document conforming to RFC 5322 message with a valid MIME body.

## 5.1.4 Protocol Conversion

Figure 16 below depicts the sequence of activity for S/MIME-to-XDR Protocol Conversion. Figure 17 depicts the sequence of activity for XDR-to-S/MIME Protocol Conversion.

**Figure 16 S/MIME-to-XDR Protocol Conversion**

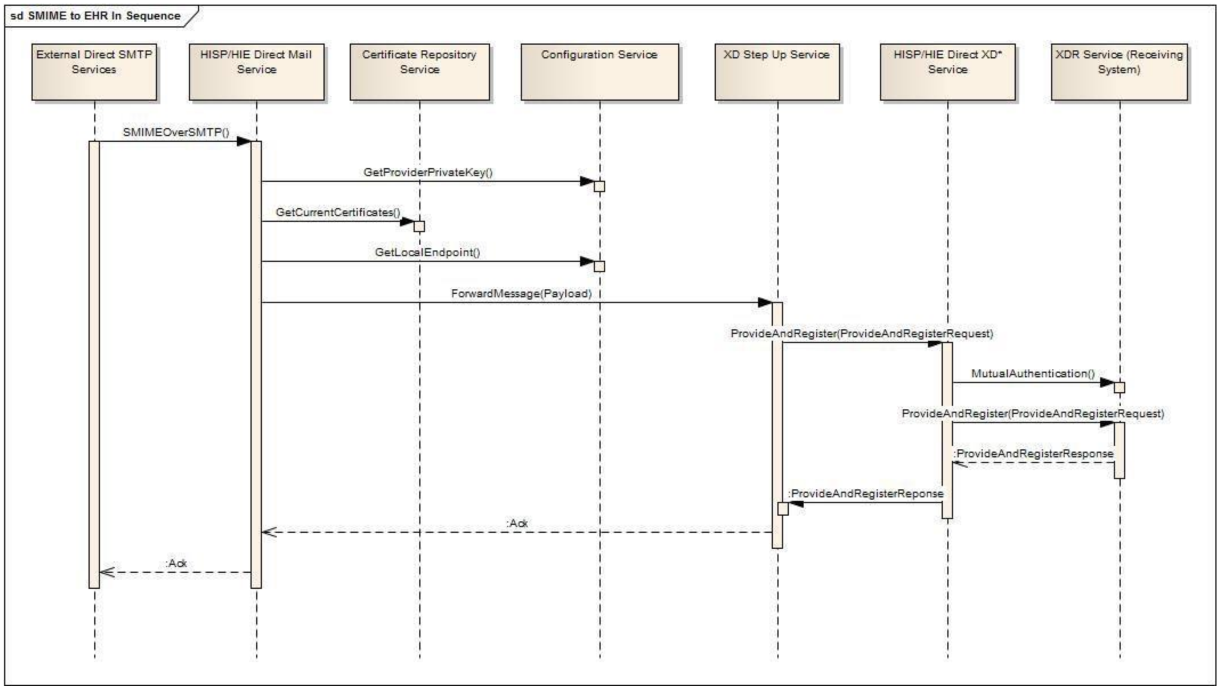
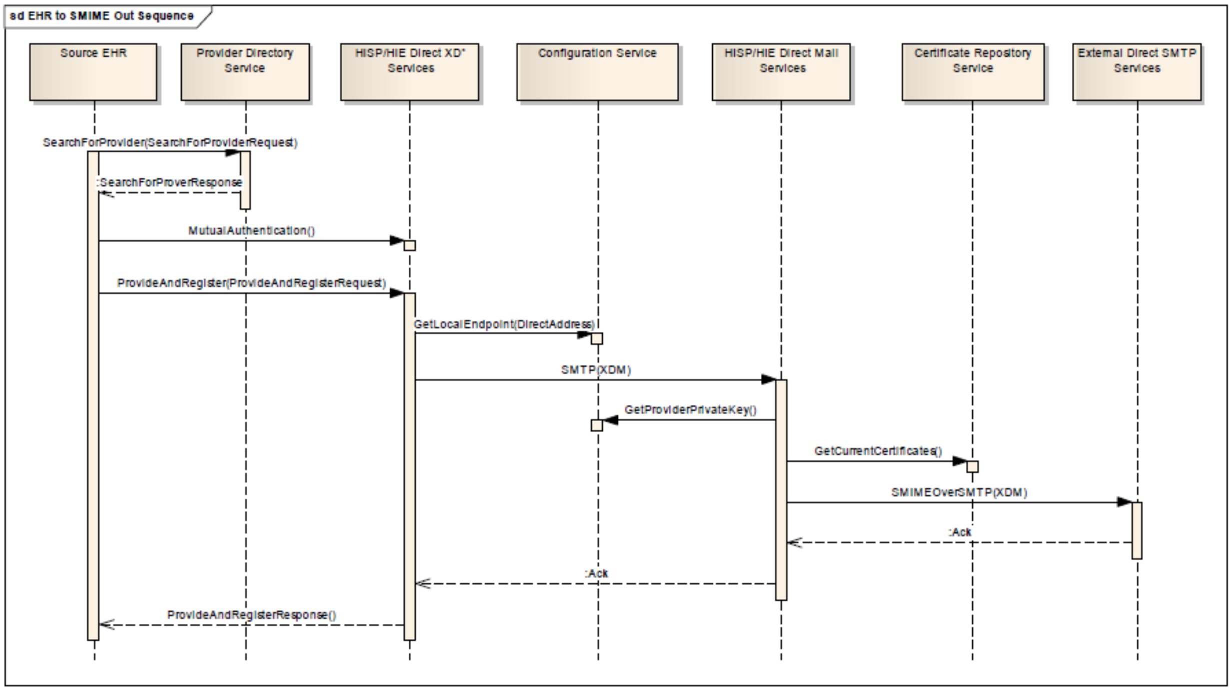

sd SMIME to EHR In Sequence
External Direct SMTP Services
HISP/HIE Direct Mail Service
Certificate Repository Service
Configuration Service
XD Step Up Service
HISP/HIE Direct XD* Service
XDR Service (Receiving System)
SMIMEOverSMTP()
GetProviderPrivateKey()
GetCurrentCertificates()
GetLocalEndpoint()
ForwardMessage(Payload)
ProvideAndRegister(ProvideAndRegisterRequest)
MutualAuthentication()
ProvideAndRegister(ProvideAndRegisterRequest)
:ProvideAndRegisterResponse
:ProvideAndRegisterReponse
:Ack
:Ack

**Figure 17 XDR-to-S/MIME Protocol Conversion**


sd EHR to SMIME Out Sequence
Source EHR
Provider Directory Service
HISP/HIE Direct XD* Services
Configuration Service
HISP/HIE Direct Mail Services
Certificate Repository Service
External Direct SMTP Services
SearchForProvider(SearchForProviderRequest)
:SearchForProverResponse
MutualAuthentication()
ProvideAndRegister(ProvideAndRegisterRequest)
GetLocalEndpoint(DirectAddress)
SMTP(XDM)
GetProviderPrivateKey()
GetCurrentCertificates()
SMIMEOverSMTP(XDM)
:Ack
:Ack
ProvideAndRegisterResponse()

### 5.1.4.1 Triggers

Sending System sends a patient record to a receiving System that is external to the HISPdomain.

### 5.1.4.2 Transaction Message Details

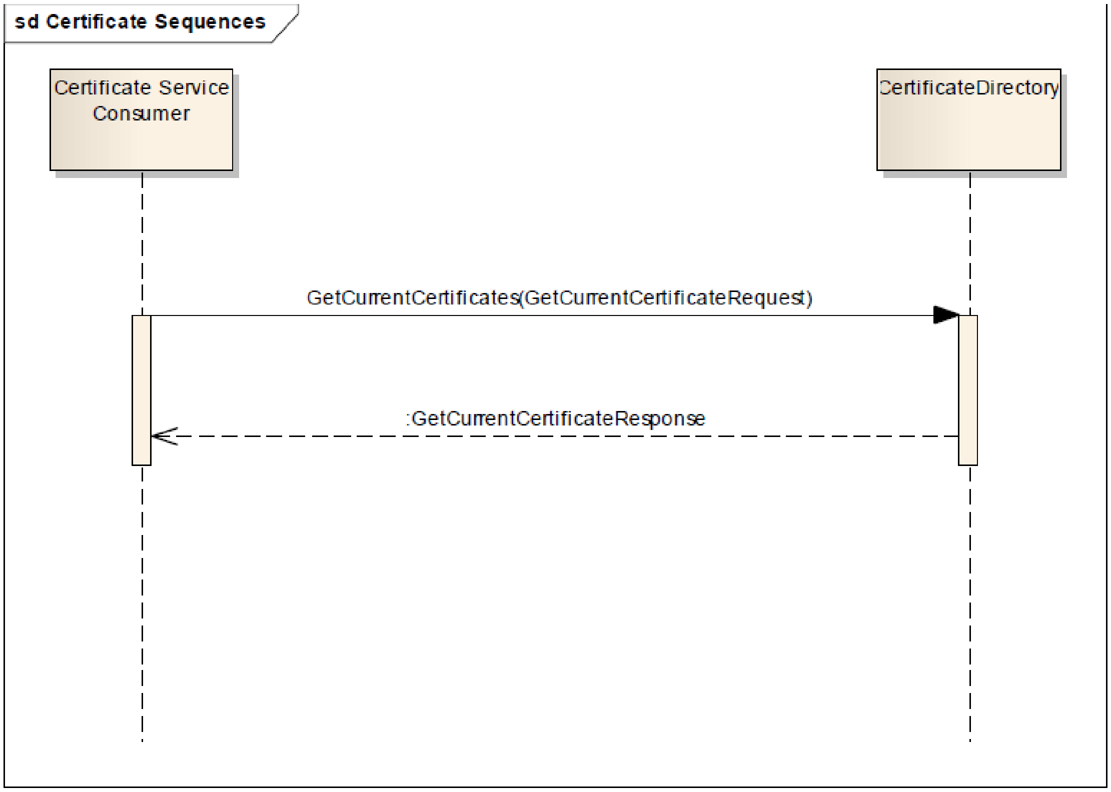
For XD bases transactions the messages are the ProvideAndRegister Request and Response. For SMTP based transactions the message is an SMTP message which may contain an IHE XDM attachment or a raw Document attachment (such as a CCD or a PDF). The XDR and XDM for Direct Messaging Specification, and the Direct Message Conversion Specification XDR and XDM from S&I Framework Modular Specification for Direct Exchange project provides details on conversion details involved in such protocol conversion transactions, including:

- Transport Conversion, which details how to map between SMTP and RFC 5322 header constructs to semantically identical constructs in SOAP, and vice versa;
- Packaging Conversion, which details how to transform an XDM package to the equivalent SOAP XDR package and vice versa;
- Metadata Conversion, which details how to create XD* Metadata from an RFC 5322 document.

## 5.1.5 Validate Sender/Receiver Certificate and Identity

Figure 18 below depicts the sequence to validate a sender’s certificate and identity.

**Figure 18 Validate Sender’s Certificate and Identity**


sd Certificate Sequences
Certificate Service Consumer
CertificateDirectory
GetCurrentCertificates(GetCurrentCertificateRequest)
:GetCurrentCertificateResponse

### 5.1.5.1 Triggers

System requests digital certificate using a Direct Address, and verifies its validity and trust.

### 5.1.5.2 Transaction Message Details

- The certificate retrieval mechanism is a combination of DNS and LDAP certificate look up, as detailed in the Retrieve Certificates section above. The verification process has been detailed in the following specifications: Applicability Statement of Secure Health Transport;
- Direct Transport and Security Specification SMTP and SMIME

A system must check the following conditions for certificate validity:

- Has not expired
- Has a valid signature
- Has not been revoked
- Binding to the expected entity
- Has a trusted certificate path.

A receiving system must also check the validity of digital signature on the S/MIME message.

Details on these verification steps can be found in the specifications cited above.

# 6 HPD Plus Functionality Description

## 6.1 Technical Specification

If an LDAP server is used to provide the application implementation behind the Provider Directory WSDL, as in standard HPD, it is assumed that it implements the LDAP Technical Specification Road Map, RFC 4510 . This RFC provides a list of several other RFCs that make up the LDAP Technical Specification. Specifically these are:

LDAP: The Protocol [RFC4511]
LDAP: Directory Information Models [RFC4512]
LDAP: Authentication Methods and Security Mechanisms [RFC4513]
LDAP: String Representation of Distinguished Names [RFC4514]
LDAP: String Representation of Search Filters [RFC4515]
LDAP: Uniform Resource Locator [RFC4516] LDAP:
Syntaxes and Matching Rules [RFC4517] LDAP:
Internationalized String Preparation [RFC4518] LDAP:
Schema for User Applications [RFC4519]

Any implementation of HPD Plus which is not LDAP specific, but does use the HPD schema, should implement the LDAP Technical Specification. The following constraints on the LDAP specification are imposed by this document for an HPD Plus implementation:

1) Security requirements are provided by the Security Standards section of this document and the LDAP: Authentication Methods and Security Mechanisms [RFC4513] is not required.
2) The information model can be in any form, it does not have to conform to LDAP: Directory Information Models [RFC4512].

Other more detailed constraints will be handled in the “HOD Plus RDB Implementation Guide” appendix for this document.
HPD Plus implementations shall support query and response transactions as defined in the ESI Query and Response specification.

# 7 Provider Directory Data Mapping

Attributes which are included in the S&I Attribute mapping column must be implemented for an S&I “Query for Electronic Service Information including Electronic Address Use Case” compliant model and are mandatory for HPD Plus. Attributes that have an LDAP Object Class mapping are found in the current HPD LDAP model. S&I Attributes descriptions contain an (R) or an (R,Q) to denote Response or Response and Query attributes respectively.

The S&I Framework calls out two attributes that describe the data source Provider Directory (Unique Reference for Provider Directory and Provider Directory Name). It has been suggested these must be mandatory response values for any query. How this should be implemented will be detailed in a subsequent Implementation Guide.

Many of the attributes defined by S&I Framework and included in the original HPD Plus specification have been adopted in IHE ITI Change Proposal 601 “Extend HPD memberOf to support attributes on relationship”, which was recently affirmed. There are still some attributes not covered by HPD that are in the S&I Framework data model. Those should be resolved in future versions of HPD.

## 7.1 Individual Data Map

The table below defines the individual attributes of the Provider Directory, its corresponding S&I Attribute and LDAP Syntax and Object Class.

**Table 1 Individual Data Map**

| Attribute | LDAP Syntax | LDAP Object Class | S & I Attribute | Single/ Multi Valued | Comments |
|---|---|---|---|---|---|
| **uid** | String | inetOrgPerson | Individual ID (R,Q) | S | RDN Format as defined by ISO 2191 section 9.2 (Issuing Authority Name:ID) |
| **hcIdentifier** | String | HCProfessional | Alternate Individual Identifier.<br>Individual NPI (R,Q) | M | Format as defined by ISO 2191 (Issuing Authority:Type:ID:Status)<br>Type values will be defined by national or regional organizations.<br>Status is defined in section 3.Y1.4.1.2.3 |
| **hcProfession** | String | HCProfessional | Individual Provider Type(R,Q) | M | The values will be defined by national or regional organizations. Allows for typing or textual description. An example of possible types is the list of Individuals or Groups Values from the Healthcare Provider Taxonomy Published by the American Medical Association twice a year. An example of this document can be found at the following reference URL: http://www.adldata.com/Downloads/Glossaries/taxonomy_80.pdf. |

| Attribute | LDAP Syntax | LDAP Object Class | S & I Attribute | Single/ Multi Valued | Comments |
|---|---|---|---|---|---|
| **Description** | String | inetOrgPerson | | S | Provider Type Description defined by HPD.<br>The definitions will be defined by national or regional organizations. See Provider Type for more information. |
| **hpdProviderStatus** | String | HPDProvider | Individual Record Status ( R ) | S | Active/Inactive/Deceased/Retired |
| **displayName** | String | inetOrgPerson | Legal Individual Name (R,Q) | S | Use of language tag and HL7 Name Data Type (XCN) as per PWP Volume 2A Section 3.24; 3.24.5.2.3.1 |
| **Title** | String | inetOrgPerson | Individual Prefix (R,Q) | S | Use of language tag and HL7 Name Data Type (XCN) as per PWP Volume 2A Section 3.24; 3.24.5.2.3.1<br>Multiple titles are separated by a space |
| **givenName** | String | inetOrgPerson | Individual First Name (R,Q) | M | Use of language tag and HL7 Name Data Type (XCN) as per PWP Volume 2A Section 3.24; 3.24.5.2.3.1 |
| **initials** | String | inetOrgPerson | Individual Middle Name (R,Q) | M | Use of language tag and HL7 Name Data Type (XCN) as per PWP Volume 2A Section 3.24; 3.24.5.2.3.1 |
| **Sn** | String | inetOrgPerson | Individual Last Name (R,Q) | M | Provider Last Name Use of language tag and HL7 Name Data Type (XCN) as per PWP Volume 2A Section 3.24; 3.24.5.2.3.1 |
| **Cn** | String | inetOrgPerson | Alternate/Other Known Individual Names (R,Q) | M | Provider Known names ,Use of language tag and HL7 Name Data Type (XCN) as per PWP Volume 2A Section 3.24; 3.24.5.2.3.1 |
| **hpdProviderLanguageSupported** | String | HPDProvider | | M | Provider language supported defined by HPD. |
| **Gender** | String | Natural Person | Individual Gender (R,Q) | S | Using Natural Person auxillary class as defined in RFC 2985 |

| Attribute | LDAP Syntax | LDAP Object Class | S & I Attribute | Single/ Multi Valued | Comments |
|---|---|---|---|---|---|
| **Mail** | String | inetOrgPerson | Individual Email (R) | M | Provider e-mail address |
| **HcSigningCertificate** | Binary | HCProfessional | Individual Public Digital Certificate (Certificate, Usage)<br>(R) | M | Base 64 encoded certificate Public key and certificate<br>for the user's nonrepudiation signing certificate used for health transactions |
| **userSMIMECertificate** | Binary | inetOrgPerson | Individual Public Digital Certificate (Certificate, Usage)<br>(R) | | HPD defined s-Mime Certificate.<br>RFC2798: PKCS#7 SignedData used to support S/MIME; typically used for encrypting mime messages over an email. Other purpose constraint can be found by looking inside the certificates. |
| **userCertificate** | Binary | inetOrgPerson | Individual Public Digital Certificate (Certificate, Usage)<br>(R) | | HPD defined User Certificate.<br>RFC2256: X.509 user certificate for general purpose use; purpose constraint can be found by looking inside the certificates |
| **labeledURI** | String | groupofURLs | | S | HPD Electronic Service URI concept.<br>Points to businessEntity through businessKey value of the IHE Services Directory for Document Sharing (SDDS).[1] |
| **creation Date** | Date | N/A | | S | Defined by HPD. This is an operation attribute that directory servers already maintains. |
| **last Update Date** | Date | N/A | | S | Defined by HPD. This is an operation attribute that directory servers already maintains. |
| **physicalDeliveryOfficeName** | String | inetOrgPerson | | M | HPD defined Provider Facility Name. |
| **hpdProviderLegalAddress** | Postal Addrss | HPDProvider | Individual Address Object (R) | S | Provider legal address. HPD extension per IHE CP 601. |
| **hpdProviderMailingAddress** | Postal Address | HPDProvider | Individual Address Object (R) | M | Mailing address |
| **hpdProviderBillingAddress** | Postal Address | HPDProvider | Individual Address Object (R) | M | Business billing address; |

[1] **Note** that HPD Plus will use the new HPDElectronicService data object for service information instead of the labeledURI attribute. The IWG will work with IHE to resolve the relationship between the new HPDEletronicService object and the labeledURI attribute.

| Attribute | LDAP Syntax | LDAP Object Class | S & I Attribute | Single/ Multi Valued | Comments |
|---|---|---|---|---|---|
| **hpdProviderPracticeAddress** | Postal Address | HPDProvider | Individual Address Object (R) | M | Practice address; |
| **HcPracticeLocation** | DN | HCProfessional | | M | HPD defined Provider Practice Organization. DN of organization the provider practices |
| **telephoneNumber** | Telephone Number | inetOrgPerson | Individual Telephone Object (Number, Usage) (R,Q) | M | As per PWP volume 2A; 3.24 |
| **Mobile** | Telephone Number | inetOrgPerson | Individual Telephone Object (Number, Usage) (R,Q) | M | As per PWP volume 2A; 3.24 Business Mobile |
| **Pager** | Telephone Number | inetOrgPerson | Individual Telephone Object (Number, Usage) (R,Q) | M | As per PWP volume 2A; 3.24 |
| **facsimileTelephoneNumber** | Facsimile Telephone Number | inetOrgPerson | Individual Telephone Object (Number, Usage) (R,Q) | M | As per PWP volume 2A; 3.24 |
| **hpdCredential** | DN | HPDProvider | Individual Professional Degree Object, Individual Certification Object, Individual State License Object (pointer) (R,Q) | M | |
| **hcSpecialisation** | String | HCProfessional | Individual Specialty, | M | A major Grouping i.e. Dermatology, Oncology, Dental, Internal Med Populate with ISO 21298 defined medical specialties. May also be populated with other specialties specified by jurisdiction or organization |
| **memberOf** | DN | HPDProvider | | M | HPD defined Provider Relationship. Groups to which this provider belongs; In search scenarios, it is desirable for a Provider Information Consumer to be able to determine which organizations this individual |

| Attribute | LDAP Syntax | LDAP Object Class | S & I Attribute | Single/ Multi Valued | Comments |
|---|---|---|---|---|---|
| | | | | | provider is a member of. |
| **suffix** | | *HPDPRovider* | Individual Suffix (R,Q) | S | HPD Plus only, Not yet supported in HPD |
| **hpdHasAService** | DN | HPDProvider | Individual Electronic Service Information | | HPD Plus , HPD extension per IHE CP 601. |
| ***dateOfBirth*** | | | Date of Birth (Q) | | HPD Plus only. Not yet supported by HPD |
| ***digitalCertificateID*** | | | Individual Digital Certificate ID | | HPD Plus only. Not yet supported in HPD |
| ***digitalCertificateDN*** | | | Individual Digital Certificate Distinguished Name | | HPD Plus only. Not yet supported in HPD |

## 7.2 Organization Data Map

The table below defines the organization attributes of the Provider Directory, its corresponding S&I Attribute and LDAP Syntax and Object Class.

**Table 4 Organization Data Map**

| Attribute | LDAP Syntax | LDAP Object Class | S&I Attribute | Single/ Multi Valued | Comments |
|---|---|---|---|---|---|
| **uid** | String | Organization | Organization ID (R,Q) | S | RDN Format as defined by ISO 2191 section 9.2 (Issuing Authority Name:ID) |
| **hcIdentifier** | String | HCRegulatedOrganization | Organization NPI, Alternate Organizational Identifier (R,Q) | M | Organization Identifiers |
| **businessCategory** | String | Organization | Organization Type (R,Q) | S | The values will be defined by national or regional organizations. An example is the list of Non Individual Values from the Healthcare Provider Taxonomy Published by the American Medical Association twice a year. An example of this document can be found at the following reference URL: |

| Attribute | LDAP Syntax | LDAP Object Class | S&I Attribute | Single/ Multi Valued | Comments |
|---|---|---|---|---|---|
| | | | | | http://www.adldata.com/Downloads/Glossaries/taxonomy_80.pdf. |
| **Description** | String | Organization | | M | The definitions will be defined by national or regional organizations. See Org Type for more information. |
| **hpdProviderStatus** | String | HPDProvider | Organization Record Status (R) | S | Active/Inactive |
| **HcRegisteredName** | String | HCRegulatedOrganization | Organization Legal Name (R,Q) | M | Use of language tag and HL7 Name Data Type (XCN) as per PWP Volume 2A Section 3.24; 3.24.5.2.3.1 |
| **o** | String | Organization | Organization Alias/Descriptive Name(s) (R) | M | Use of language tag and HL7 Name Data Type (XCN) as per PWP Volume 2A Section 3.24; 3.24.5.2.3.1 |
| **ClinicalInformationContact** | DN | HCRegulatedOrganization | | M | HPD defined Clinical contacts; DN to HCProfessional entry |
| **hpdProviderPracticeAddress** | Postal Address | HPDProvider | Organization Address Object (R) | M | Organization Practice Address |
| **hpdProviderMailingAddress** | Postal Address | HPDProvider | Organization Address Object (R) | M | Organization Mailing Address |
| **hpdProviderBillingAddress** | Postal Address | HPDProvider | Organization Address Object (R) | M | Organization Billing Address |
| **hpdProviderLegalAddress** | Postal Addrss | HPDProvider | Organization Address Object (R) | S | Organization legal address. HPD extension per IHE CP 601. |
| **hpdCredential** | DN | HPDProvider | Organization Certification Object (pointer) | M | Degree is not a valid type for Organizational Provider |
| **hpdProviderLanguageSupported** | String | HPDProvider | | M | Language that the organization supports |
| **HcSpecialisation** | String | HCRegulatedOrganization | Organization Specialty (R,Q) | M | |

| Attribute | LDAP Syntax | LDAP Object Class | S&I Attribute | Single/ Multi Valued | Comments |
|---|---|---|---|---|---|
| **labeledURI** | String | groupofURLs | | M | Electronic address information of an Organization. Points to businessEntity through businessKey value of the IHE Services Directory for Document Sharing (SDDS) |
| **HcSigningCertificate** | Binary | HCRegulatedOrganization | Organization Public Digital Certificate (R ) | M | Signing Certificate |
| **HcOrganizationCertificate** | Binary | HCRegulatedOrganization | Organization Public Digital Certificate (R ) | M | Used for storing health care Organization certificates; Certificate purpose constraint can be found by looking inside the certificates. |
| **telephone** | Telephone Number | Organization | Organization Telephone Object (Number, Usage) (R,Q) | M | Organization Business Phone |
| **facsimileTelephoneNumber** | Facsimile Telephone Number | Organization | Organization Telephone Object (Number, Usage) (R,Q) | M | Organization fax number |
| **Memberof** | DN | HPDProvider | Organization Relationship (Unique Reference for related organization) | M | Organization to which this organization belongs; In search scenarios, it is desirable for a Provider Information Consumer to be able to determine which organizations this organization provider is a member of. |
| **creation Date** | | | | S | This is an operation attribute that directory servers already maintains. |
| **last Update Date** | | | | S | This is an operation attribute that directory servers already maintains. |
| **hpdHasAService** | DN | HPDProvider | Organization Electronic Service Information Object | M | HPD Plus. HPD extension per IHE CP 601. |
| **email** | | | Organization Email Object | M | HPD Plus only, Not yet supported in HPD |
| **policyInformation** | | | Organization Policy Information (R) Object | M | HPD Plus only, Not yet supported in HPD |

| Attribute | LDAP Syntax | LDAP Object Class | S&I Attribute | Single/ Multi Valued | Comments |
|---|---|---|---|---|---|
| ***digitalCertificateID*** | | | Organization Digital Certificate ID | | HPD Plus only. Not yet supported in HPD |
| ***digitalCertificateDN*** | | | Organization Digital Certificate Distinguished Name | | HPD Plus only. Not yet supported in HPD |

## 7.3 Individual-Organization Data Map

While the original HPD Framework does contain a section on Relationships, it only supports member of relationship and does not support other attributes. The table below defines the individual-Organization attributes of the Provider Directory, its corresponding S&I Attribute and LDAP Syntax and Object Class. Many of the changes have been adopted in HPD extension as defined in IHE Change Proposal 601.

**Table 5 Individual-Organization Data Map**

| Attribute | LDAP Syntax | LDAP Object Class | S&I Attribute | Single/ Multi Valued | Comments |
|---|---|---|---|---|---|
| **hpdMemberId** | String | HPDProvider Membership | | S | HPD Plus, HPD extension per IHE CP 601. |
| **hpdHasAProvider** | DN | HPDProvider Membership | Unique Reference for Individual<br>(R) | S | HPD Plus , HPD extension per IHE CP 601. |
| **hpdHasAnOrg** | DN | HPDProvider Membership | Unique Reference for Organization<br>(R) | S | HPD Plus , , HPD extension per IHE CP 601 |
| **hpdHasAService** | DN | HPDProvider Membership | Electronic Service Information Object Pointer<br>(R) | M | HPD Plus, HPD extension per IHE CP 601 |

| Attribute | LDAP Syntax | LDAP Object Class | S&I Attribute | Single/ Multi Valued | Comments |
|---|---|---|---|---|---|
| **telephoneNumber** | TelephoneNumber | HPDProvider Membership | Telephone Object (Number, Usage) (R,Q) | M | HPD Plus , HPD extension per IHE CP 601 |
| **facsimileTelephoneNumber** | TelephoneNumber | HPDProvider Membership | Telephone Object (Number, Usage) (R,Q) | M | HPD extension per IHE CP 601 |
| **Mobile** | TelephoneNumber | HPDProvider Membership | Telephone Object (Number, Usage) (R,Q) | M | HPD extension per IHE CP 601 |
| **Pager** | TelephoneNumber | HPDProvider Membership | Telephone Object (Number, Usage) (R,Q) | M | HPD extension per IHE CP 601 |
| **mail** | String | HPDProvider Membership | Email Object (Email, Usage) (R,Q) | M | HPD Plus , HPD extension per IHE CP 601 |
| ***digitalCertificate ?*** | Binary | | Organization Public Digital Certificate | | HPD Plus only. Not yet supported in HPD |
| ***digitalCertificateID ?*** | | | Organization Digital Certificate ID | | HPD Plus only. Not yet supported in HPD |
| ***digitalCertificateDN ?*** | | | Organization Digital Certificate Distinguished Name | | HPD Plus only. Not yet supported in HPD |

## 7.4 Credential Data Map

The Credential Object provides a means of referencing data on the S&I Individual Certification Object, Individual State License Object and the Individual Digital Certificate Object attributes. The table below defines the Credential attributes of the Provider Directory, its corresponding S&I Attribute and LDAP Syntax and Object Class.

**Table 2 Credential Data Map**

| Attribute | LDAP Syntax | LDAP Object Class | S&I Attribute | Single/ Multi Valued | Comments |
|---|---|---|---|---|---|

| Attribute | LDAP Syntax | LDAP Object Class | S&I Attribute | Single/ Multi Valued | Comments |
|---|---|---|---|---|---|
| **credentialType** | Directory String | HPDProvider Credential | License,Degree, or Certification types | S | Type of Credential<degree, certificate, credential> |
| **credentialName** | Directory String | HPDProvider Credential | License Type, Certificate Type, Granting Institution | S | Name of Credential, degree, or certification that belongs to provider.<br>Follows the ISO21091 naming format as that of the HCStandardRole: |
| **credentialNumber** | Directory String | HPDProvider Credential | Degree, Certificate Number or License Number, | S | Credential Identifier Follows the ISO 21091 UID format: |
| **credentialDescription** | Directory String | HPDProvider Credential | State Issuing, Issuing Authority | S | Additional information on the credential |
| **credentialIssueDate** | Date | HPDProvider Credential | Year Granted or Issued | S | Date when credential was issued to the provider |
| **credentialRenewalDate** | Date | HPDProvider Credential | | S | Date when credential is due renewal |
| **credentialStatus** | Directory String | HPDProvider Credential | status | S | The current status of credential: Active, Inactive, Suspended etc |

## 7.5 Electronic Services Data Map

This Object is defined in HPD Plus first. The table below defines the electronic services attributes of the Provider Directory, its corresponding S&I Attribute and LDAP Syntax and Object Class. Many of the changes have been adopted in HPD extension as defined in IHE Change Proposal 601.

**Table 3 Electronic Services Data Map**

| **Attribute** | **LDAP Syntax** | **LDAP Object Class** | **S&I Attribute** | **Single/ Multi Valued** | **Comments** |
|---|---|---|---|---|---|
| **hpdServiceId** | Directory String | HPDElectronicService | Unique ID for Electronic Service Information<br>(Q, R) | S | HPD Plus, HPD extension per IHE CP601 |
| **hpdServiceAddress** | Directory String | HPDElectronicSerice | Electronic Service Address<br>(Q, R) | S | HPD Plus, HPD extension per IHE CP601 |
| **hpdContentProfile** | Directory String | HPDElectronicSerice | Content Profile Object<br>(Prfile, version, constraints) | M | HPD Plus, HPD extension per IHE CP601 |
| **hpdIntegrationProfile** | Directory String | HPDElectronicSerice | Integration Profile | M | HPD Plus, HPD extension per IHE CP601 |
| **hpdCertificate** | Binary | HPDElectronicSerice | | M | Public Digitial Certificate for this sevice, per HPD extension per IHE CP601. |
| ***hpdSecurityProfile ?*** | | | Security Profile object (profile, version, constraints) | M | An object that identifies, if appropriate, the Security Profile supported by the service.<br>HPD Plus only. Not yet supported in HPD. |
| ***hpdIntegrationProfileVersion ?*** | | | Integration Profile version | M | HPD Plus only. Not yet supported in HPD. |

| Attribute | LDAP Syntax | LDAP Object Class | S&I Attribute | Single/ Multi Valued | Comments |
|---|---|---|---|---|---|
| ***hpdIntegrationProfileOption ?*** | | | Integration Profile Options | M | HPD Plus only. Not yet supported in HPD. |
| ***hpdIntegrationProfilePreferredMethod ?*** | | | Preferred method | S | HPD Plus only. Not yet supported in HPD. |

## 7.6 LDAP Schema Structure

The HPD schema defines LDAP Organizational Unit (OU) containers to organize the information on Providers. Object classes within OU represent objects such as individual providers, organizational providers, credentials. IHE HPD had defined standard OU for individual providers, organizational providers, credentials and relationships. Standard OU has not been defined by IHE for the newly added class of HPDProviderMembership and HPDElectronicServices yet. Below is a table of what HPD Plus uses.

| HPD Object | LDAP Organization Unit | Comments |
|---|---|---|
| Individual Provider | **HCProfessional** | HPD Plus v1.0 defined as Providers. Updated to be consistent with IHE HPD. |
| Organizational Provider | **HCRegulatedOrganization** | HPD Plus v1.0 defined as Organizations. Updated to be consistent with IHE HPD. |
| Individual-Organization | HPDProviderMembership | HPD Plus v1.0 defined as Membership.<br>IHE HPD extension per CP 601 defined the object class but not the standardized OU. The standard OU used here is consistent with HIE HPD naming convention.. |
| HPDProviderCredentials | HPDCredential | HPD Plus v1.0 defined as Credentials. Updated to be consistent with IHE HPD. |
| Electronic Services | HPDElectronicService | HPD Plus v1.0 defined as Services. IHE HPD extension per CP 601 defined the object class but not the standardized OU. The standard OU used here is consistent with HIE HPD naming |

# 8 Technical Aspects

## 8.1 Security Standards

Security for all XDR transactions and Provider Directory transactions will use TLS with mutual authentication, per the IHE standards. This includes HPD and HPD Plus. Also a SAML assertion as part of a SOAP WSSE security header must be included for user non-repudiation. The "NHIN Authorization Framework Specification v 2.0.X" shall provide the requirements for the security mechanism for all SOAP transactions in this document.

## 8.2 Logging

XDR transaction logging is covered in the IHE XDR and XDS specifications. Logging of the user data that is described in the SAML assertion shall be in compliance with the IHE Cross Enterprise User Assertion (XUA) standard.

## 8.3 Federation of Provider Directory

While federation of the Provider Directory is out of scope for this document, the following considerations and options for federation should be considered by the readers and implementers of this specification.

1. The Provider Directory query data element are designed to support federation by including:
    a. Unique Reference for the Provider Directory – allows query to a specific provider directory by a unique reference ID
    b. Provider Directory Name – allows query to a specific Provider Directory by a unique name or multiple Provider Directories a partial name depending on the specific implementation
2. The Provider Directory response data elements are designed to support federation by including:
    a. Unique Reference for the Provider Directory – returns the unique reference ID to a specific Provider Directory associated with each record returned by the query
    b. Provider Directory Name – returns the provider directory name for the specific Provider Directory associated with each record returned by the query

Provider Directory federation may be implemented by any of the following methods:

1. Each HIE/EHR maintains a list of all relevant Provider Directories and uses that list to determine which Provider Directory to query for a specific endpoint.
2. A Provider Directory Gateway is defined and implemented such that it permits a query to a single address and then determines which individual directory or directories should receive the query. The results are collated and returned to the requester, via the gateway, along with the Unique Reference for the Provider Directory and the Provider Directory Name for each record returned.
3. Provider Directories maintain a list of all other Provider Directories and with each query they determine if they are able to answer the query and/or which other Provider Directories should participate in the query response. Each Provider Directory is capable of forwarding the query to

the other relevant Provider Directories and receiving and collating the response(s) to the querying system. The results are returned to the requester along with the Unique Reference for the Provider Directory and the Provider Directory Name for each record returned.

4. The Provider Directories have an internal replication and federation mechanism (like DNS or LDAP) and use this to ensure that each directory can answer the query for all participating Provider Directories).

Each of the above federation models has advantages and limitations base on specific technology deployed, security issues, Provider Directory management, and robustness of the final solution to various operational and security failures.

# 9 Appendix A, Provider Directory WSDL

## 9.1 HPD WSDL

```
<?xml version="1.0" encoding="utf-8"?>
<definitions name="HPDProviderInformationDirectory"
  targetNamespace="urn:ihe:iti:hpd:2010"
  xmlns:tns="urn:ihe:iti:hpd:2010"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:dsml="urn:oasis:names:tc:DSML:2:0:core"
  xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap12/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <documentation>IHE HPD Provider Information Directory</documentation>
  <types>
    <xsd:schema
      targetNamespace="urn:oasis:names:tc:DSML:2:0:core"
      xmlns:dsml="urn:oasis:names:tc:DSML:2:0:core">
      <xsd:include schemaLocation="../schema/DSML/DSMLv2.xsd"/>
    </xsd:schema>
  </types>
  <message name="ProviderInformationRequestMessage">
    <documentation>Provider Information Query/Feed Request Message</documentation>
    <part name="body" element="dsml:batchRequest" />
  </message>
  <message name="ProviderInformationResponseMessage">
    <documentation>Provider Information Query/Feed ResponseMessage</documentation>
    <part name="body" element="dsml:batchResponse"/>
  </message>
  <portType name="ProviderInformationDirectory_PortType">
    <operation name="ProviderInformationQueryRequest">
      <input message="tns:ProviderInformationRequestMessage"
        wsaw:Action="urn:ihe:iti:2010:ProviderInformationQuery"/>
      <output message="tns:ProviderInformationResponseMessage"
        wsaw:Action="urn:ihe:iti:2010:ProviderInformationQueryResponse"/>
    </operation>
    <operation name="ProviderInformationFeedRequest">
      <input message="tns:ProviderInformationRequestMessage"
        wsaw:Action="urn:ihe:iti:2010:ProviderInformationFeed"/>
      <output message="tns:ProviderInformationResponseMessage"
        wsaw:Action="urn:ihe:iti:2010:ProviderInformationFeedResponse"/>
    </operation>
  </portType>
  <binding name="ProviderInformationDirectory_Binding"
    type="tns:ProviderInformationDirectory_PortType">
    <soap:binding style="document"
      transport="http://schemas.xmlsoap.org/soap/http" />
    <operation name="ProviderInformationQueryRequest">
      <soap:operation
        soapAction="urn:ihe:iti:hpd:2010:ProviderInformationQueryRequest" />
      <input>
        <soap:body use="literal" />
      </input>
      <output>
        <soap:body use="literal" />
      </output>
    </operation>
    <operation name="ProviderInformationFeedRequest">
      <soap:operation
        soapAction="urn:ihe:iti:hpd:2010:ProviderInformationFeedRequest" />
      <input>
        <soap:body use="literal" />
      </input>
      <output>
        <soap:body use="literal" />
      </output>
    </operation>
  </binding>
  <service name="ProviderInformationDirectory_Service">
    <port name="ProviderInformationDirectory_Port_Soap" binding="tns:ProviderInformationDirectory_Binding">
      <soap:address location="https://localhost:${HttpsDefaultPort}/ProvoderInformationDirectoryService"/>
    </port>
  </service>
</definitions>
```

# 10 Appendix B, DSMLv2 Schema Definition

```
<xsd:schema targetNamespace="urn:oasis:names:tc:DSML:2:0:core" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="urn:oasis:names:tc:DSML:2:0:core" elementFormDefault="qualified">
    <!-- DSML Requests -->
    <xsd:group name="DSMLRequests">
        <xsd:choice>
            <xsd:element name="authRequest" type="AuthRequest"/>
            <xsd:group ref="BatchRequests"/>
        </xsd:choice>
    </xsd:group>
    <xsd:group name="BatchRequests">
        <xsd:choice>
            <xsd:element name="searchRequest" type="SearchRequest"/>
            <xsd:element name="modifyRequest" type="ModifyRequest"/>
            <xsd:element name="addRequest" type="AddRequest"/>
            <xsd:element name="delRequest" type="DelRequest"/>
            <xsd:element name="modDNRequest" type="ModifyDNRequest"/>
            <xsd:element name="compareRequest" type="CompareRequest"/>
            <xsd:element name="abandonRequest" type="AbandonRequest"/>
            <xsd:element name="extendedRequest" type="ExtendedRequest"/>
        </xsd:choice>
    </xsd:group>
    <!-- DSML Responses -->
    <xsd:group name="DSMLResponses">
        <xsd:choice>
            <xsd:element name="authResponse" type="LDAPResult"/>
            <xsd:element name="searchResultEntry" type="SearchResultEntry"/>
            <xsd:element name="searchResultReference" type="SearchResultReference"/>
            <xsd:element name="searchResultDone" type="LDAPResult"/>
            <xsd:element name="modifyResponse" type="LDAPResult"/>
            <xsd:element name="addResponse" type="LDAPResult"/>
            <xsd:element name="delResponse" type="LDAPResult"/>
            <xsd:element name="modDNResponse" type="LDAPResult"/>
            <xsd:element name="compareResponse" type="LDAPResult"/>
            <xsd:element name="extendedResponse" type="ExtendedResponse"/>
            <xsd:element name="errorResponse" type="ErrorResponse"/>
        </xsd:choice>
    </xsd:group>
    <!--  *************** Batch Envelopes ********************* -->
    <xsd:element name="batchRequest" type="BatchRequest"/>
    <xsd:element name="batchResponse" type="BatchResponse"/>
    <!-- **** Batch Request Envelope **** -->
    <xsd:complexType name="BatchRequest">
        <xsd:sequence>
            <xsd:element name="authRequest" type="AuthRequest" minOccurs="0"/>
            <xsd:group ref="BatchRequests" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="requestID" type="RequestID" use="optional"/>
        <xsd:attribute name="processing" use="optional" default="sequential">
            <xsd:simpleType>
                <xsd:restriction base="xsd:string">
                    <xsd:enumeration value="sequential"/>
                    <xsd:enumeration value="parallel"/>
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:attribute>
        <xsd:attribute name="responseOrder" use="optional" default="sequential">
            <xsd:simpleType>
                <xsd:restriction base="xsd:string">
                    <xsd:enumeration value="sequential"/>
                    <xsd:enumeration value="unordered"/>
                </xsd:restriction>
                </xsd:simpleType>
            </xsd:attribute>
            <xsd:attribute name="onError" use="optional" default="exit">
                <xsd:simpleType>
                    <xsd:restriction base="xsd:string">
                        <xsd:enumeration value="resume"/>
                        <xsd:enumeration value="exit"/>
                    </xsd:restriction>
                </xsd:simpleType>
            </xsd:attribute>
        </xsd:complexType>
        <!-- **** Batch Response Envelope **** -->
        <xsd:complexType name="BatchResponse">
            <xsd:sequence>
                <xsd:group ref="BatchResponses" minOccurs="0" maxOccurs="unbounded"/>
            </xsd:sequence>
            <xsd:attribute name="requestID" type="RequestID" use="optional"/>
        </xsd:complexType>
        <!-- **** Batch Responses **** -->
        <xsd:group name="BatchResponses">
            <xsd:choice>
                <xsd:element name="searchResponse" type="SearchResponse"/>
                <xsd:element name="authResponse" type="LDAPResult"/>
                <xsd:element name="modifyResponse" type="LDAPResult"/>
                <xsd:element name="addResponse" type="LDAPResult"/>
                <xsd:element name="delResponse" type="LDAPResult"/>
                <xsd:element name="modDNResponse" type="LDAPResult"/>
                <xsd:element name="compareResponse" type="LDAPResult"/>
                <xsd:element name="extendedResponse" type="ExtendedResponse"/>
                <xsd:element name="errorResponse" type="ErrorResponse"/>
            </xsd:choice>
        </xsd:group>
        <!-- **** Search Response **** -->
        <xsd:complexType name="SearchResponse">
            <xsd:sequence>
                <xsd:element name="searchResultEntry" type="SearchResultEntry" minOccurs="0" maxOccurs="unbounded"/>
                <xsd:element name="searchResultReference" type="SearchResultReference" minOccurs="0"
maxOccurs="unbounded"/>
                <xsd:element name="searchResultDone" type="LDAPResult"/>
            </xsd:sequence>
            <xsd:attribute name="requestID" type="RequestID" use="optional"/>
        </xsd:complexType>
        <!-- ***** DsmlDN ***** -->
        <xsd:simpleType name="DsmlDN">
            <xsd:restriction base="xsd:string"/>
        </xsd:simpleType>
        <!-- ***** DsmlRDN ***** -->
        <xsd:simpleType name="DsmlRDN">
            <xsd:restriction base="xsd:string"/>
        </xsd:simpleType>
        <!-- ***** Request ID ***** -->
        <xsd:simpleType name="RequestID">
            <xsd:restriction base="xsd:string"/>
        </xsd:simpleType>
        <!-- ***** AttributeDescriptionValue ***** -->
        <xsd:simpleType name="AttributeDescriptionValue">
            <xsd:restriction base="xsd:string">
                <xsd:pattern value="((([0-2](\.[0-9]+)+)|([a-zA-Z]+([a-zA-Z0-9]|[-])*))(;([a-zA-Z0-9]|[-])+)*)"/>
            </xsd:restriction>
        </xsd:simpleType>
        <xsd:simpleType name="NumericOID">
            <xsd:restriction base="xsd:string">
                <xsd:pattern value="[0-2]\.[0-9]+(\.[0-9]+)*"/>
            </xsd:restriction>
        </xsd:simpleType>
        <!-- ***** MAX Integer ***** -->
```

```
<xsd:simpleType name="MAXINT">
    <xsd:restriction base="xsd:unsignedInt">
        <xsd:maxInclusive value="2147483647"/>
    </xsd:restriction>
</xsd:simpleType>
<!-- **** DSML Value **** -->
<xsd:simpleType name="DsmlValue">
    <xsd:union memberTypes="xsd:string xsd:base64Binary xsd:anyURI"/>
</xsd:simpleType>
<!-- **** DSML Control **** -->
<xsd:complexType name="Control">
    <xsd:sequence>
        <xsd:element name="controlValue" type="xsd:anyType" minOccurs="0"/>
    </xsd:sequence>
    <xsd:attribute name="type" type="NumericOID" use="required"/>
    <xsd:attribute name="criticality" type="xsd:boolean" use="optional" default="false"/>
</xsd:complexType>
<!-- **** DSML Filter **** -->
<xsd:complexType name="Filter">
    <xsd:group ref="FilterGroup"/>
</xsd:complexType>
<xsd:group name="FilterGroup">
    <xsd:sequence>
        <xsd:choice>
            <xsd:element name="and" type="FilterSet"/>
            <xsd:element name="or" type="FilterSet"/>
            <xsd:element name="not" type="Filter"/>
            <xsd:element name="equalityMatch" type="AttributeValueAssertion"/>
            <xsd:element name="substrings" type="SubstringFilter"/>
            <xsd:element name="greaterOrEqual" type="AttributeValueAssertion"/>
            <xsd:element name="lessOrEqual" type="AttributeValueAssertion"/>
            <xsd:element name="present" type="AttributeDescription"/>
            <xsd:element name="approxMatch" type="AttributeValueAssertion"/>
            <xsd:element name="extensibleMatch" type="MatchingRuleAssertion"/>
        </xsd:choice>
    </xsd:sequence>
</xsd:group>
<xsd:complexType name="FilterSet">
    <xsd:sequence>
        <xsd:group ref="FilterGroup" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="AttributeValueAssertion">
    <xsd:sequence>
        <xsd:element name="value" type="DsmlValue"/>
    </xsd:sequence>
    <xsd:attribute name="name" type="AttributeDescriptionValue" use="required"/>
</xsd:complexType>
<xsd:complexType name="AttributeDescription">
    <xsd:attribute name="name" type="AttributeDescriptionValue" use="required"/>
</xsd:complexType>
<xsd:complexType name="SubstringFilter">
    <xsd:sequence>
        <xsd:element name="initial" type="DsmlValue" minOccurs="0"/>
        <xsd:element name="any" type="DsmlValue" minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element name="final" type="DsmlValue" minOccurs="0"/>
    </xsd:sequence>
    <xsd:attribute name="name" type="AttributeDescriptionValue" use="required"/>
</xsd:complexType>
<xsd:complexType name="MatchingRuleAssertion">
    <xsd:sequence>
        <xsd:element name="value" type="DsmlValue"/>
    </xsd:sequence>
    <xsd:attribute name="dnAttributes" type="xsd:boolean" use="optional" default="false"/>
    <xsd:attribute name="matchingRule" type="xsd:string" use="optional"/>
    <xsd:attribute name="name" type="AttributeDescriptionValue" use="optional"/>
```

```
</xsd:complexType>
<!--  *************** DSML MESSAGE ******************** -->
<xsd:complexType name="DsmlMessage">
    <xsd:sequence>
        <xsd:element name="control" type="Control" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="requestID" type="RequestID" use="optional"/>
</xsd:complexType>
<!--  *************** LDAP RESULT ********************* -->
<xsd:simpleType name="LDAPResultCode">
    <xsd:restriction base="xsd:string">
        <xsd:enumeration value="success"/>
        <xsd:enumeration value="operationsError"/>
        <xsd:enumeration value="protocolError"/>
        <xsd:enumeration value="timeLimitExceeded"/>
        <xsd:enumeration value="sizeLimitExceeded"/>
        <xsd:enumeration value="compareFalse"/>
        <xsd:enumeration value="compareTrue"/>
        <xsd:enumeration value="authMethodNotSupported"/>
        <xsd:enumeration value="strongAuthRequired"/>
        <xsd:enumeration value="referral"/>
        <xsd:enumeration value="adminLimitExceeded"/>
        <xsd:enumeration value="unavailableCriticalExtension"/>
        <xsd:enumeration value="confidentialityRequired"/>
        <xsd:enumeration value="saslBindInProgress"/>
        <xsd:enumeration value="noSuchAttribute"/>
        <xsd:enumeration value="undefinedAttributeType"/>
        <xsd:enumeration value="inappropriateMatching"/>
        <xsd:enumeration value="constraintViolation"/>
        <xsd:enumeration value="attributeOrValueExists"/>
        <xsd:enumeration value="invalidAttributeSyntax"/>
        <xsd:enumeration value="noSuchObject"/>
        <xsd:enumeration value="aliasProblem"/>
        <xsd:enumeration value="invalidDNSyntax"/>
        <xsd:enumeration value="aliasDerefencingProblem"/>
        <xsd:enumeration value="inappropriateAuthentication"/>
        <xsd:enumeration value="invalidCredentials"/>
        <xsd:enumeration value="insufficientAccessRights"/>
        <xsd:enumeration value="busy"/>
        <xsd:enumeration value="unavailable"/>
        <xsd:enumeration value="unwillingToPerform"/>
        <xsd:enumeration value="loopDetect"/>
        <xsd:enumeration value="namingViolation"/>
        <xsd:enumeration value="objectClassViolation"/>
        <xsd:enumeration value="notAllowedOnNonLeaf"/>
        <xsd:enumeration value="notAllowedOnRDN"/>
        <xsd:enumeration value="entryAlreadyExists"/>
        <xsd:enumeration value="objectClassModsProhibited"/>
        <xsd:enumeration value="affectMultipleDSAs"/>
        <xsd:enumeration value="other"/>
    </xsd:restriction>
</xsd:simpleType>
<xsd:complexType name="ResultCode">
    <xsd:attribute name="code" type="xsd:int" use="required"/>
    <xsd:attribute name="descr" type="LDAPResultCode" use="optional"/>
</xsd:complexType>
<xsd:complexType name="LDAPResult">
    <xsd:complexContent>
        <xsd:extension base="DsmlMessage">
            <xsd:sequence>
                <xsd:element name="resultCode" type="ResultCode"/>
                <xsd:element name="errorMessage" type="xsd:string" minOccurs="0"/>
                <xsd:element name="referral" type="xsd:anyURI" minOccurs="0" maxOccurs="unbounded"/>
            </xsd:sequence>
            <xsd:attribute name="matchedDN" type="DsmlDN" use="optional"/>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="ErrorResponse">
    <xsd:sequence>
        <xsd:element name="message" type="xsd:string" minOccurs="0"/>
        <xsd:element name="detail" minOccurs="0">
            <xsd:complexType>
                <xsd:sequence>
                    <xsd:any/>
                </xsd:sequence>
            </xsd:complexType>
        </xsd:element>
    </xsd:sequence>
    <xsd:attribute name="requestID" type="RequestID" use="optional"/>
    <xsd:attribute name="type">
        <xsd:simpleType>
            <xsd:restriction base="xsd:string">
                <xsd:enumeration value="notAttempted"/>
                <xsd:enumeration value="couldNotConnect"/>
                <xsd:enumeration value="connectionClosed"/>
                <xsd:enumeration value="malformedRequest"/>
                <xsd:enumeration value="gatewayInternalError"/>
                <xsd:enumeration value="authenticationFailed"/>
                <xsd:enumeration value="unresolvableURI"/>
                <xsd:enumeration value="other"/>
            </xsd:restriction>
        </xsd:simpleType>
    </xsd:attribute>
</xsd:complexType>
<!-- *************** Auth ********************* -->
<xsd:complexType name="AuthRequest">
    <xsd:complexContent>
        <xsd:extension base="DsmlMessage">
            <xsd:attribute name="principal" type="xsd:string" use="required"/>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
<!-- *************** Search ********************* -->
<xsd:complexType name="AttributeDescriptions">
    <xsd:sequence minOccurs="0" maxOccurs="unbounded">
        <xsd:element name="attribute" type="AttributeDescription"/>
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="SearchRequest">
    <xsd:complexContent>
        <xsd:extension base="DsmlMessage">
            <xsd:sequence>
                <xsd:element name="filter" type="Filter"/>
                <xsd:element name="attributes" type="AttributeDescriptions" minOccurs="0"/>
            </xsd:sequence>
            <xsd:attribute name="dn" type="DsmlDN" use="required"/>
            <xsd:attribute name="scope" use="required">
                <xsd:simpleType>
                    <xsd:restriction base="xsd:string">
                        <xsd:enumeration value="baseObject"/>
                        <xsd:enumeration value="singleLevel"/>
                        <xsd:enumeration value="wholeSubtree"/>
                    </xsd:restriction>
                </xsd:simpleType>
            </xsd:attribute>
            <xsd:attribute name="derefAliases" use="required">
                <xsd:simpleType>
                    <xsd:restriction base="xsd:string">
                        <xsd:enumeration value="neverDerefAliases"/>
                        <xsd:enumeration value="derefInSearching"/>
                        <xsd:enumeration value="derefFindingBaseObj"/>
                        <xsd:enumeration value="derefAlways"/>
                    </xsd:restriction>
                </xsd:simpleType>
            </xsd:attribute>
            <xsd:attribute name="sizeLimit" type="MAXINT" use="optional" default="0"/>
            <xsd:attribute name="timeLimit" type="MAXINT" use="optional" default="0"/>
            <xsd:attribute name="typesOnly" type="xsd:boolean" use="optional" default="false"/>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
<!-- ***** Search Result Entry ***** -->
<xsd:complexType name="SearchResultEntry">
    <xsd:complexContent>
        <xsd:extension base="DsmlMessage">
            <xsd:sequence>
                <xsd:element name="attr" type="DsmlAttr" minOccurs="0" maxOccurs="unbounded"/>
            </xsd:sequence>
            <xsd:attribute name="dn" type="DsmlDN" use="required"/>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="DsmlAttr">
    <xsd:sequence>
        <xsd:element name="value" type="DsmlValue" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="name" type="AttributeDescriptionValue" use="required"/>
</xsd:complexType>
<xsd:complexType name="DsmlModification">
    <xsd:sequence>
        <xsd:element name="value" type="DsmlValue" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="name" type="AttributeDescriptionValue" use="required"/>
    <xsd:attribute name="operation" use="required">
        <xsd:simpleType>
            <xsd:restriction base="xsd:string">
                <xsd:enumeration value="add"/>
                <xsd:enumeration value="delete"/>
                <xsd:enumeration value="replace"/>
            </xsd:restriction>
        </xsd:simpleType>
    </xsd:attribute>
</xsd:complexType>
<!-- ***** Search Result Reference ***** -->
<xsd:complexType name="SearchResultReference">
    <xsd:complexContent>
        <xsd:extension base="DsmlMessage">
            <xsd:sequence>
                <xsd:element name="ref" type="xsd:anyURI" maxOccurs="unbounded"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
<!-- ************* MODIFY ******************** -->
<xsd:complexType name="ModifyRequest">
    <xsd:complexContent>
        <xsd:extension base="DsmlMessage">
            <xsd:sequence>
                <xsd:element name="modification" type="DsmlModification" minOccurs="0" maxOccurs="unbounded"/>
            </xsd:sequence>
            <xsd:attribute name="dn" type="DsmlDN" use="required"/>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
<!--  *************** ADD ********************* -->
<xsd:complexType name="AddRequest">
    <xsd:complexContent>
```

```
            <xsd:extension base="DsmlMessage">
                <xsd:sequence>
                    <xsd:element name="attr" type="DsmlAttr" minOccurs="0" maxOccurs="unbounded"/>
                </xsd:sequence>
                <xsd:attribute name="dn" type="DsmlDN" use="required"/>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <!-- *************** DELETE ********************* -->
    <xsd:complexType name="DelRequest">
        <xsd:complexContent>
            <xsd:extension base="DsmlMessage">
                <xsd:attribute name="dn" type="DsmlDN" use="required"/>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <!-- *************** MODIFY DN ********************* -->
    <xsd:complexType name="ModifyDNRequest">
        <xsd:complexContent>
            <xsd:extension base="DsmlMessage">
                <xsd:attribute name="dn" type="DsmlDN" use="required"/>
                <xsd:attribute name="newrdn" type="DsmlRDN" use="required"/>
                <xsd:attribute name="deleteoldrdn" type="xsd:boolean" use="optional" default="true"/>
                <xsd:attribute name="newSuperior" type="DsmlDN" use="optional"/>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <!-- ************* COMPARE ******************** -->
    <xsd:complexType name="CompareRequest">
        <xsd:complexContent>
            <xsd:extension base="DsmlMessage">
                <xsd:sequence>
                    <xsd:element name="assertion" type="AttributeValueAssertion"/>
                </xsd:sequence>
                <xsd:attribute name="dn" type="DsmlDN" use="required"/>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <!-- ***** ABANDON ***** -->
    <xsd:complexType name="AbandonRequest">
        <xsd:complexContent>
            <xsd:extension base="DsmlMessage">
                <xsd:attribute name="abandonID" type="RequestID" use="required"/>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <!-- ************* EXTENDED OPERATION ******************** -->
    <xsd:complexType name="ExtendedRequest">
        <xsd:complexContent>
            <xsd:extension base="DsmlMessage">
                <xsd:sequence>
                    <xsd:element name="requestName" type="NumericOID"/>
                    <xsd:element name="requestValue" type="xsd:anyType" minOccurs="0"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="ExtendedResponse">
        <xsd:complexContent>
            <xsd:extension base="LDAPResult">
                <xsd:sequence>
                    <xsd:element name="responseName" type="NumericOID" minOccurs="0"/>
                    <xsd:element name="response" type="xsd:anyType" minOccurs="0"/>
                </xsd:sequence>
            </xsd:extension>
```