Services Directory (ServD)

Version 1.0 – FTF Beta 2~~1~~

_____

OMG Document Number: dtc/~~2012-xx-xx~~2013-06~~5-11~~03

Standard document URL: http://www.omg.org/spec/ServD/1.0

Associated File(s)*: http://www.omg.org/spec/ServD/130601~~120812~~/ServD_model.xmi
http://www.omg.org/spec/ServD/130601~~120812~~/ServD_model.eap
http://www.omg.org/spec/ServD/130601~~120813~~/MaintainableServDCore.wsdl

http://www.omg.org/spec/ServD/130601~~120813~~/ServD_Locator.wsdl
http://www.omg.org/spec/ServD/130601~~120813~~/ServD_Verification.wsdl
~~http://www.omg.org/spec/ServD/120812/ServD_Interfaces.zip~~

~~*Source documents health/12-08-02, health/12-08-08, health/12-09-01, health/12-09-02, health/12-09-03~~

_____

This OMG document replaces the submission document (health/12-08-01, Alpha). It is an OMG Adopted Beta Specification and is currently in the finalization phase. Comments on the content of this document are welcome, and should be directed to *issues@omg.org* by March 25, 2013.

You may view the pending issues for this specification from the OMG revision issues web page *http://www.omg.org/issues/*.

The FTF Recommendation and Report for this specification will be published on June 28 2013. If you are reading this after that date, please download the available specification from http://www.omg.org/spec/ServD/1.0.

INDIRECT, INCIDENTAL, SPECIAL, CONSEQUENTIAL, RELIANCE OR COVER DAMAGES, INCLUDING LOSS OF PROFITS, REVENUE, DATA OR USE, INCURRED BY ANY USER OR ANY THIRD PARTY IN CONNECTION WITH THE FURNISHING, PERFORMANCE, OR USE OF THIS MATERIAL, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

The entire risk as to the quality and performance of software developed using this specification is borne by you. This disclaimer of warranty constitutes an essential part of the license granted to you to use this specification.

RESTRICTED RIGHTS LEGEND

Use, duplication or disclosure by the U.S. Government is subject to the restrictions set forth in subparagraph (c) (1) (ii) of The Rights in Technical Data and Computer Software Clause at DFARS 252.227-7013 or in subparagraph (c)(1) and (2) of the Commercial Computer Software - Restricted Rights clauses at 48 C.F.R. 52.227-19 or as specified in 48 C.F.R. 227-7202-2 of the DoD F.A.R. Supplement and its successors, or as specified in 48 C.F.R. 12.212 of the Federal Acquisition Regulations and its successors, as applicable. The specification copyright owners are as indicated above and may be contacted through the Object Management Group, 109 Highland Avenue, Needham, MA 02494, U.S.A.

TRADEMARKS

MDA®, Model Driven Architecture®, UML®, UML Cube logo®, OMG Logo®, CORBA® and XMI® are registered trademarks of the Object Management Group, Inc., and Object Management Group™, OMG™ , Unified Modeling Language™, Model Driven Architecture Logo™, Model Driven Architecture Diagram™, CORBA logos™, XMI Logo™, CWM™, CWM Logo™, IIOP™ , MOF™ , OMG Interface Definition Language (IDL)™ , and OMG SysML™ are trademarks of the Object Management Group. All other products or company names mentioned are used for identification purposes only, and may be trademarks of their respective owners.

COMPLIANCE

The copyright holders listed above acknowledge that the Object Management Group (acting itself or through its designees) is and shall at all times be the sole entity that may authorize developers, suppliers and sellers of computer software to use certification marks, trademarks or other special designations to indicate compliance with these materials.

Software developed under the terms of this license may claim compliance or conformance with this specification if and only if the software compliance is of a nature fully matching the applicable compliance points as stated in the specification. Software developed only partially matching the applicable compliance points may claim only that the software was based on this specification, but may not claim compliance or conformance with this specification. In the event that testing suites are implemented or approved by Object Management Group, Inc., software developed using this specification may claim compliance or conformance with the specification only if the software satisfactorily completes the testing suites.

OMG's Issue Reporting Procedure

All OMG specifications are subject to continuous review and improvement. As part of this process we encourage readers to report any ambiguities, inconsistencies, or inaccuracies they may find by completing the form listed here: *http://www.omg.org/report_issue.htm*.

Table of Contents

Table of Contents

Preface

About the Object Management Group

OMG

Founded in 1989, the {xe "Object Management Group, Inc. (OMG)"}Object Management Group, Inc. (OMG) is an open membership, not-for-profit computer industry standards consortium that produces and maintains computer industry specifications for interoperable, portable and reusable enterprise applications in distributed, heterogeneous environments. Membership includes Information Technology vendors, end users, government agencies and academia.

OMG member companies write, adopt, and maintain its specifications following a mature, open process. OMG's specifications implement the Model Driven Architecture® (MDA®), maximizing ROI through a full-lifecycle approach to enterprise integration that covers multiple operating systems, programming languages, middleware and networking infrastructures, and software development environments. OMG's specifications include: UML® (Unified Modeling Language™); CORBA® (Common Object Request Broker Architecture); CWM™ (Common Warehouse Metamodel); and industry-specific standards for dozens of vertical markets.

More information on the OMG is available at *http://www.omg.org/*.

OMG Specifications

As noted, {xe "OMG specifications"}OMG specifications address middleware, modeling and vertical domain frameworks. All OMG Specifications are available from this URL:

http://www.omg.org/spec

Specifications are organized by the following categories:

Business Modeling Specifications

Middleware Specifications

- **CORBA/IIOP**
- **Data Distribution Services**
- **Specialized CORBA**

IDL/Language Mapping Specifications

Modeling and Metadata Specifications

- **UML, MOF, CWM, XMI**
- **UML Profile**

Modernization Specifications

Platform Independent Model (PIM), Platform Specific Model (PSM), Interface Specifications

- **CORBAServices**

- **CORBAFacilities**

OMG Domain Specifications

CORBA Embedded Intelligence Specifications

CORBA Security Specifications

All of OMG's formal specifications may be downloaded without charge from our website. (Products implementing OMG specifications are available from individual suppliers.) Copies of specifications, available in PostScript and PDF format, may be obtained from the Specifications Catalog cited above or by contacting the Object Management Group, Inc. at:

> OMG Headquarters
> 109 Highland Avenue
> Needham, MA 02494
> USA
> Tel: +1-781-444-0404
> Fax: +1-781-444-0320
> Email: *pubs@omg.org*

Certain OMG specifications are also available as ISO standards. Please consult *http://www.iso.org*

Typographical Conventions

The type styles shown below are used in this document to distinguish programming statements from ordinary English. However, these {xe "typographical conventions"}conventions are not used in tables or section headings where no distinction is necessary.

Times/Times New Roman - 10 pt.: Standard body text

Helvetica/Arial - 10 pt. Bold: OMG Interface Definition Language (OMG IDL) and syntax elements.

`Courier – 10 pt. Bold:` Programming language elements.

Helvetica/Arial - 10 pt: Exceptions

Note – Terms that appear in *italics* are defined in the glossary. Italic text also represents the name of a document, specification, or other publication.

Issues

The reader is encouraged to report any technical or editing {xe "issues/problems"}issues/problems with this specification to *http://www.omg.org/report_issue.htm*.

1.   Scope

The Services Directory (ServD) specification provides an SOA model to support the discovery of, and access to, service provider individual, and organizational information including; locations, associations, contact details, services, identifiers and many other relevant characteristics/attributes. This information can be relevant for use by both people and/or computers. ~~The Services Directory (ServD) specification provides an SOA model to support the discovery of, and access to, service provider individual, and organization location information.~~ Actors are able to use this information to support secure transfer of personal and confidential information to network end point applications or shared repositories.

This specification incorporates the HL7 document *Service Functional Model Specification: Healthcare, Community Services and Provider Directory Version 2.0, Dec 2009,* and complements that work through the documentation of interfaces required to support service discovery and information sharing between systems.

This specification has been developed to meets the needs of interoperability in healthcare and community services, although at its highest level of abstraction ServD is non-domain specific and could be used in a variety of settings where there is a need for directory-based search facility and secure transmission of sensitive documents amongst a large and dispersed population of service providers.

Both a Platform Independent Model (PIM) and a Platform Specific Model (PSM) based on web services have been included.

The mandatory conformance points for the specification have been included in the Normative Section and otherwise as listed. Individual data items in the data schema have been marked as mandatory or optional as appropriate.

1.1   Example Use Cases

Sample scenarios for which the ServD specification covers are described below. Use cases that elaborate these scenarios are described in Appendix A, in the informative section of this document.

- Mary Brown is a single mother with a teenage daughter who has been refusing to go to school and appearing depressed. She wants to find a counseling service for adolescents. She does not have a car and would prefer a location of the service close to public transport.

- Sally Jones attends hospital for an appendectomy. On admission, Sally informs the intake worker of the name and clinic location of her GP. Upon discharge, the hospital registrar wishes to send a discharge summary to her GP with results of her operation.

- John Simpson has a lump in the neck. His specialist would like to refer him to a radiology service for x-rays, however John is due to be interstate next week for work. The doctor would like to find a radiology service close to where John will be staying whilst away.

- Harry Hubert is admitted to hospital. On admission, the intake worker looks up the services directory and discovers that the address of the referring clinic provided by Harry is different from the hospital provider directory. Upon calling the clinic, the intake worker verifies the change of address. The intake worker would like to change the clinic address on their hospital application and ensure that the change is also reflected in the services directory.

1.2 Overview of Interfaces

The scope of this specification covers the full implementation details of the Interfaces (shown in Green below) and also how these are to be exposed and consumed by application components.
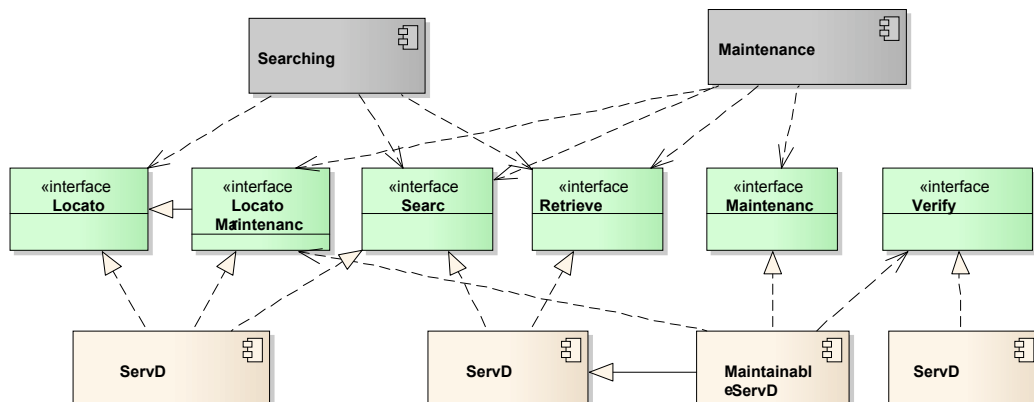
The details of each application component are provided in the Platform Independent Model Specifications Component Model.

The design of the ServD solution is to not provide a single centralized directory, but to support a federation of interconnected ServD directories. The central point of this structure is the ServD Locator that behaves as a kind of Domain Name Services (DNS) for ServD Searching within the Federation (for details of the Federation refer to section 8.3.2).

1.2.1 Locator

The **Locator** interface provides a way for **Searching Applications** to locate relevant **ServD Core** instances in the ServD Federation that expose the **Search** interface.

In simple terms it is an indexer of (or DNS for) **ServD Core** instances.

1.2.2 Locator Maintenance

The **Locator Maintenance** interface provides a way to maintain the federated index of **ServD Core** instances managed by a **ServD Locator**.

1.2.3 Search

The **Search** interface is the core interface in the **ServD** specification. It provides operations to search for lists of summary information about **Organizations**, **Sites**, **Services** and **Providers**.

The address (URI) of the **ServD Core** instance's **Search** interface is provided by the **ServD Locator**.

1.2.4 Retrieve Details

The **Retrieve Details** interface provides a way to retrieve detailed information about Organizations, Sites, Service Sites and Providers.

The address (URI) of the **ServD Core** instance's **Retrieve Details** interface is provided in the results for each item returned by the **Search** interface.

1.2.5 Maintenance

The **Maintenance** Interface provides a way for an authorized **Maintenance Application** to maintain directory data.

1.2.6 Verify Details

The **Verify Details** Interface provides a way for **Maintainable ServD Core** instances to verify updates made by a **Maintenance Application** to attributes which are configured to be validated by calling an external application.

This is the mechanism for certification/registration organizations to be able to electronically validate attributes of Organizations, Sites and Providers.

INFORMATIVE NOTE: EXTERNAL APPLICATIONS MAY BE HOSTED BY HEALTH PRACTITIONER REGISTRATION BOARDS, INSURANCE COMPANIES, REGISTRATION AUTHORITIES ETC.

1.3   Component Capability Discovery

Once an actor has found the Service, Provider, Site or Organization that they are seeking they then need to know what communicate methods are available. These can be either the simple details that they found during the search such as a phone, fax number, or website address. Where the actor is using an application that supports electronic communications the capabilities of the remote system's connectivity options needs to be available, along with any additional information required to make the connection, such as public keys for PKI connections.

The ServD specification facilitates the use of an Endpoint Locator Service (ELS) to determine the capabilities of components in order to enable B2B applications to communicate electronically.

An Endpoint Location Service (ELS) is a simple directory of technical services enabling a producer of an electronic message (source) to transfer a message to an intended recipient (target), even if the producer has no prior knowledge of the method chosen by the recipient to handle the transfer.

The ELS is similar in function to a UDDI service, except that the ELS is not restricted to Web Services, and can be used for secure email addressing, video conferencing addressing or many others.

The ELS is described further in the informative section A.2.4 Endpoint Location Service.

2. Conformance Profile

There are six types of conformance based on the interfaces exposed and/or consumed. They are:

1. ***ServD Locator****.* A component that provides indexing for the network locations of **ServD Core** instances (specified by Conformance Levels 2 and 3) and their Coverage Areas. It exposes the **Locator** and **Locator Maintenance** Interfaces.

2. ***ServD Core****.* A component that exposes the **Search** interface to permit the unsecured searching of health related ***Service*** information. The ***Service*** information can be maintained in anyway, such as via internal systems or a website.

3. ***Maintainable ServD Core****.* A component that exposes the **Maintenance** interface to permit the programmatic maintenance of the ***Service*** information.

4. ***ServD Verifier.*** A component that exposes the **Verify Details** interface to permit the secured verification of claimed attributes by a Provider, Organization or Service.

5. ***Searching Application****.* An application that calls the ServD Locator and ServD Core components to search for health related ***Service*** information.

6. ***Maintenance Application****.* A component that calls the **Maintenance** interface to update ***Service*** information.

These levels of conformance and the conformance points to be tested are specified in the Conformance Tests section 8.11 of this document.

3.   References

3.1   Normative References

The following normative documents contain provisions which, through reference in this text, constitute provisions of this specification. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply.

- HL7 Service Functional Model Specification: Healthcare, Community Services and Provider Directory Version 2.0, Dec 2010, retrieved from *http://hssp-provider-services-directory.wikispaces.com/Active+Work*

- Common Terminology Services 2, 1.0 Beta 2, June 2012, retrieved from *http://www.omg.org/spec/CTS2/1.0/Beta2/*

- W3C, XML Signature Syntax and Processing, W3C Recommendation, 12 February 2002 *http://www.w3.org/TR/2002/REC-xmldsig-core-20020212/Overview.html*

3.2   Informative Reference

For information on the Endpoint Location Service (ELS):

- Technical Report, Endpoint Location Service, TR 5823-2010, v1.3 *http://infostore.saiglobal.com/store/Details.aspx?productid=1391037*

4.   Terms and Definitions

For the purposes of this specification, the following terms and definitions apply.

term	meaning
term	*meaning*
Agency	An agency is an organization that provides health care services, or is connected with the provision of health care services. In this specification, the term *agency* is deprecated in favor of the term "Organization" (abbreviated to "org" in some contexts). See *Organization*.
Anonymous Searcher	An Anonymous Searcher is an actor that does not require authentication to access the system. They will only be able to access information in the public domain, i.e. that is not subject to privacy controls.
Approver	The Approver is an actor that is able to update information in the system for which they are authorized. Any changes they make are not subject to moderation.
Attribute	This term is used in this document to describe properties that can be attached to an Organization, Site or Provider that are used to assist in discovery of these entities through a search.
Authenticated Searcher	An Authenticated Searcher is an actor that requires authentication to access the system. They are able to access additional information not available to the Anonymous Searcher i.e. information that may be subject to privacy controls. They are not able to update any information in the system.
Author	An Author is an actor that is able to update information in the system for which they are authorized. Depending on workflow and authorization, changes to data may be subject to moderation by an Approver. Informative Note: The general process is that an Approver will moderate changes made by an Author.
Component	A Component represents a modular part of a system, that encapsulates its content and whose manifestation is replaceable within its environment. A component defines its behavior in terms of *provided* and *required* interfaces
Coverage Area	A Coverage Area is a geographic region within which Services are provided.
ELS	abbr. *Endpoint Locator Service*. A set of web services that interacts with a repository of information to allow an application to specify (write) or determine (read) the capabilities and location of an Endpoint. These capabilities are not limited to SOAP based web services; the capability can also describe email, REST or other interface types. Note: The ServD interfaces are primarily used to locate an entity that you want to receive services with and basic human based connectivity information (such as phone/fax/website) for computer based interactions where appropriate addressing, security requirements and compatible standards are needed the ELS provides this extended information.
Endpoint	An Endpoint is a network location for a Component with a defined SOA Interface.
Federation	A networked set of otherwise independent ServD instances that are linked by a common Profile and Policy framework.
Interaction (ELS)	An Interaction is an entry in the ELS describing the location and capabilities of an Endpoint.

Jurisdiction	A Jurisdiction is an organizational entity responsible for administration of a ServD instance.
Moderation	The process by which information updated by an Author is verified by an Approver. This process may require a verification of changes at the source Organization.
Organization	A corporate entity that provides, directly or indirectly, health care services.
Policy	A set of business rules or procedures that describe how information with a system is managed in order to protect the security, quality and accuracy of its content.
Profile	A generic extension mechanism for customizing UML models for particular domains and platforms. Extension mechanisms allow refining standard semantics in strictly additive manner, so that they can't contradict standard semantics.
Private Data	Private Data is an instance of a data item that the owner of the data has made available only to authenticated access, and on the basis of a Policy decision. e.g., the address of a Women's Shelter will be withheld from public view.
Provider	A person who provides a Service, such as a doctor (general practice) or a qualified practitioner.
Secure Data	Secured Data is information that can only be provided to authenticated users. e.g., Identifiers
Service	See section 5 Symbols
Service Type	A category of healthcare *Service* that relates to the discipline associated with the *Service* being provided (such as general practice, dentistry, physiotherapy).
Site	A location that provides one or more types of professional, consumer based services.

5. Symbols

API	Application Programming Interface
B2B	Business to Business
CTS2	Common Terminology Services 2
DSMLv2	Directory Services Markup Language Version 2
ELS	Endpoint Locator Service
HCSPDIR	Health & Community Services Provider Directory
HL7	Health Level 7
HSD	Human Service Directory
IHE	Integrating the Healthcare Enterprise
LDAP	Lightweight Directory Access Protocol
NEHTA	National e-Health Transition Authority (Australia)
Service	Where used in **bold** and *italic*, "***Service***", refers to a healthcare or related activity delivered by a provider, such as a doctor or counselor.
UML	Unified Modeling Language

6.   Changes to HL7 Specifications

Currently there are no requirements identified which modify any HL7 specifications.

7. Acknowledgements

The following companies submitted this specification:

- Database Consultants Australia

The following companies supported this specification:

HP

Sparx Systems

MITRE

NEHTA

Mayo Clinic

Vissumpoint

8. Platform Independent Model Specification

The Platform Independent Model provides an implementation independent definition of the interfaces, services, and conceptual data model proposed for the **ServD** standard.

8.1 Component Model

The component model shows the SOA Interfaces that are required to be implemented by each relevant service.
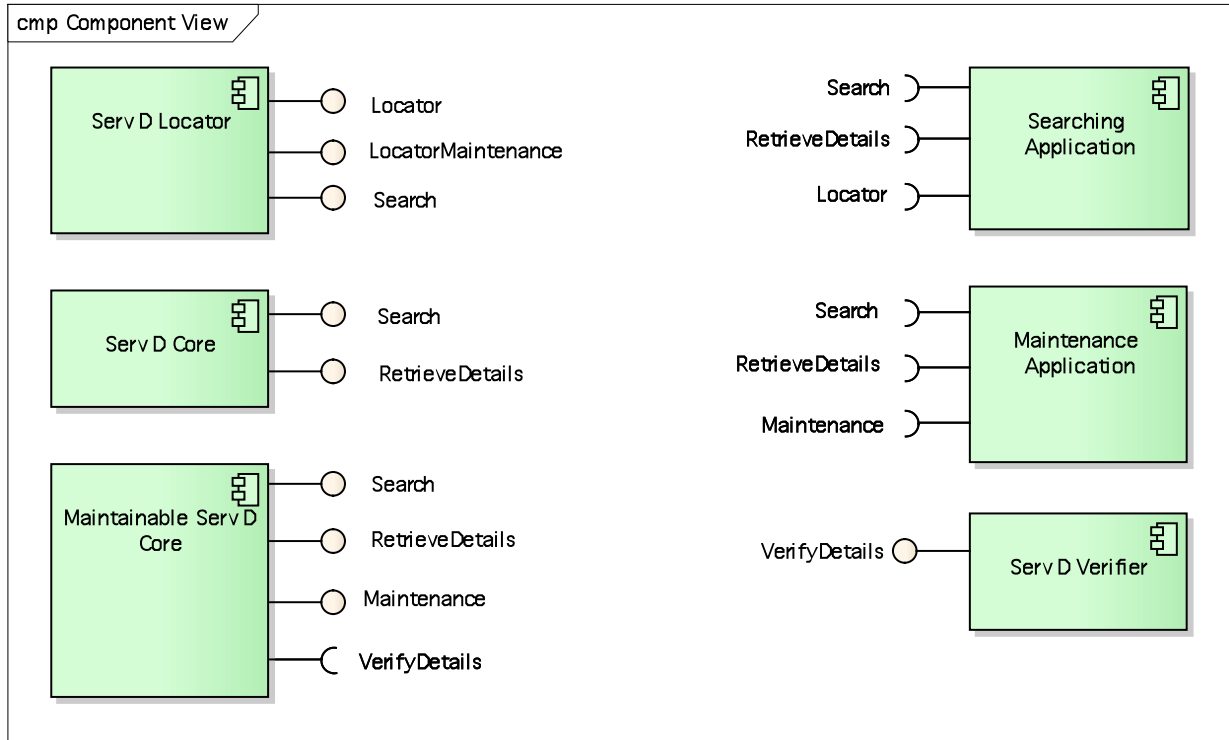
Note: As described in Section 8.11 of this document, each of the Components as shown forms a level of conformance. For details of the actual conformance tests, refer section<u>Conformance Profile Tests</u> ~~Conformance Profile Tests~~ later in this document.

8.1.1 ServD Locator

A **ServD Locator** provides an indexing service for the network locations of **ServD Cores**, and their Coverage Areas or Jurisdiction. It does not store *Service* related information.

A **Searching Application** uses the **ServD Locator** to derive the network location of **ServD Core** instance(s) that are associated with the specified Coverage Area(s).

8.1.2 ServD Core

The **ServD Core** is the minimum required implementation to be certified as **ServD** compliant.

All instances of a **ServD Core** must expose both the **Search** and **Retrieve Details** interfaces to permit the searching for *Service* data.

An instance of a ServD Core component can have its **Search** and **Retrieve Details** interfaces hosted on one or more network resources, provided that the results from the **Search** service use the corresponding **Retrieve Details** component (enabling the use of web server farms).

8.1.3 Maintainable ServD Core

The **Maintainable ServD Core** extends the **ServD Core** by exposing the **Maintenance** interface to enable **Maintenance Applications** to securely update an Organization's *Service* data.

The **Maintenance Application** may be a part of existing software such as a clinical application, or may be a web application directly calling the API, or may be a utility that processes an extract from another directory.

Informative Note: Often the process of adding an Organization requires verification of credentials and allocation of user access which are a part of application administration.

Informative Note: The Maintainable ServD Core ensures that anyone accessing the Maintenance interface is only able to update data for which they are authorized. In some cases an entity may be appointed as proxy to administer updates on behalf of multiple organizations.

8.1.4 Searching Application

A **Searching Application** is any application (or web-site) that searches for Services, Sites, Providers or Organizations based on a set of search criteria.

The Searching application communicates with one or more **ServD Cores**. The network location of the **Retrieve Details** interface may not be the same location as the **Search** interface.

8.1.5 Maintenance Application

A **Maintenance Application** is an application that has authorized access to a **Maintainable ServD Core** to update the Service Directory using the **Maintenance** interface.

8.1.6 ServD Verifier

A **ServD Verifier** verifies Attribute updates provided by a Maintenance Application, by invoking a call to an external application (such as a credentialing authority or certification authority). . When a ServD Verifier confirms that an Attribute is valid, that Attribute in the ServD repository is marked with the **VerificationStatus** property **Verified** and the expiry date is set, if provided.

8.2 Interaction Diagrams

8.2.1 Provider Search

The ServD Locator Search returns the network location(s) of ServD Service instance(s) that are associated with the requested Coverage Area. The Coverage Area may be derived from a physical geographic location provided with the search request, such as a postcode, suburb, or geocode.
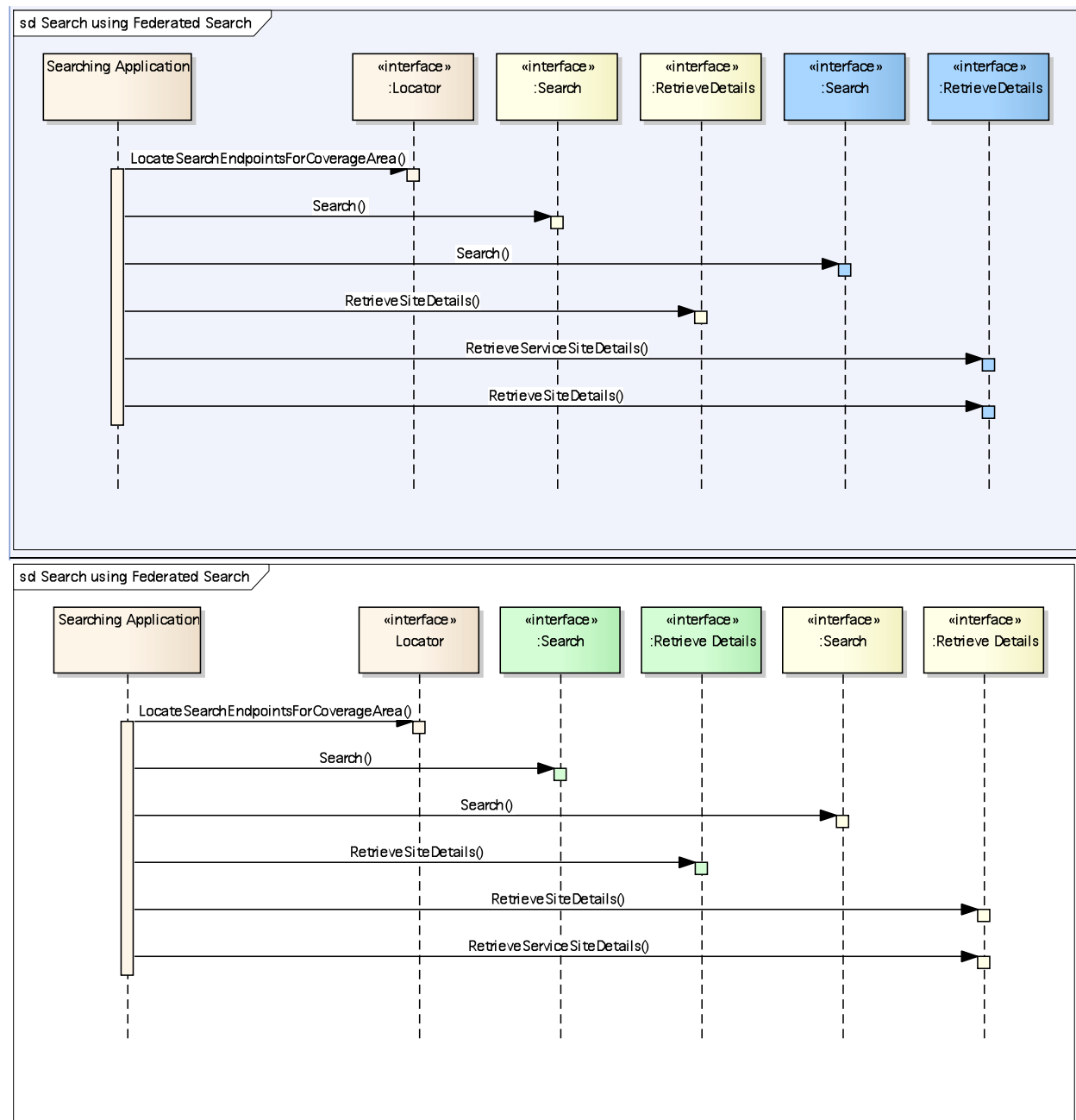
Note in this example the Search was sent to the green and the yellow ServD instances. The retrieval of the *Site* details and the *Service Site* details was from the instance(s) identified by the Locator. The user of the Searching application decided they did not need to see the Service Site details from the Green instance, and requested the detail from the yellow instance.

8.2.2 Provider Search (Federated)

The **ServD Locator** will identify the appropriate Core ServD instances to call, aggregates the results of the call(s) and passes the aggregated data back to the caller. If there are exceptional conditions, such as results from a ServD instance not returned in an adequate period of time, then the service will return the results retrieved to that point, along with exception information.

sd Federated Search - Search Interface

| Searching Application | «interface» ServD Locator Service :Search | «interface» :Search | «interface» :Search | «interface» :RetrieveDetails | «interface» :RetrieveDetails |

Search()

Search()

Search()

RetrieveSiteDetails()

RetrieveServiceSiteDetails()
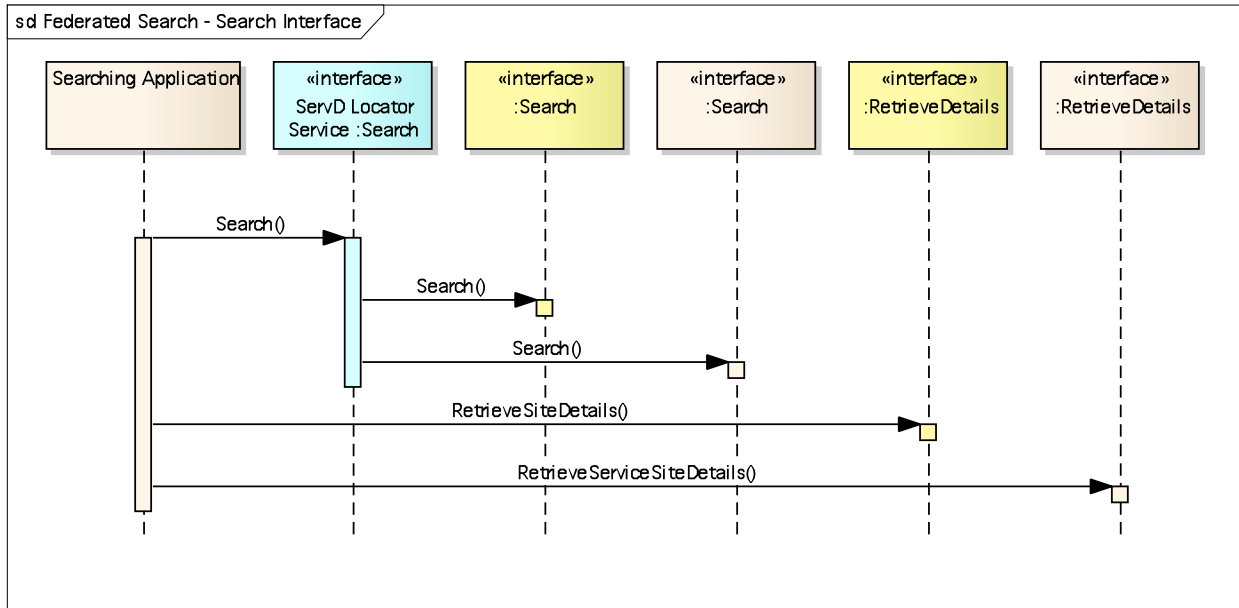
8.2.3 Maintain Provider Record

When a **Maintenance Application** updates a Provider record, the process is represented in the sequence diagram below. The call to the **Search** interface is optional, as the caller will normally have the Provider Id record that they require in order to perform the update to the ServD instance.
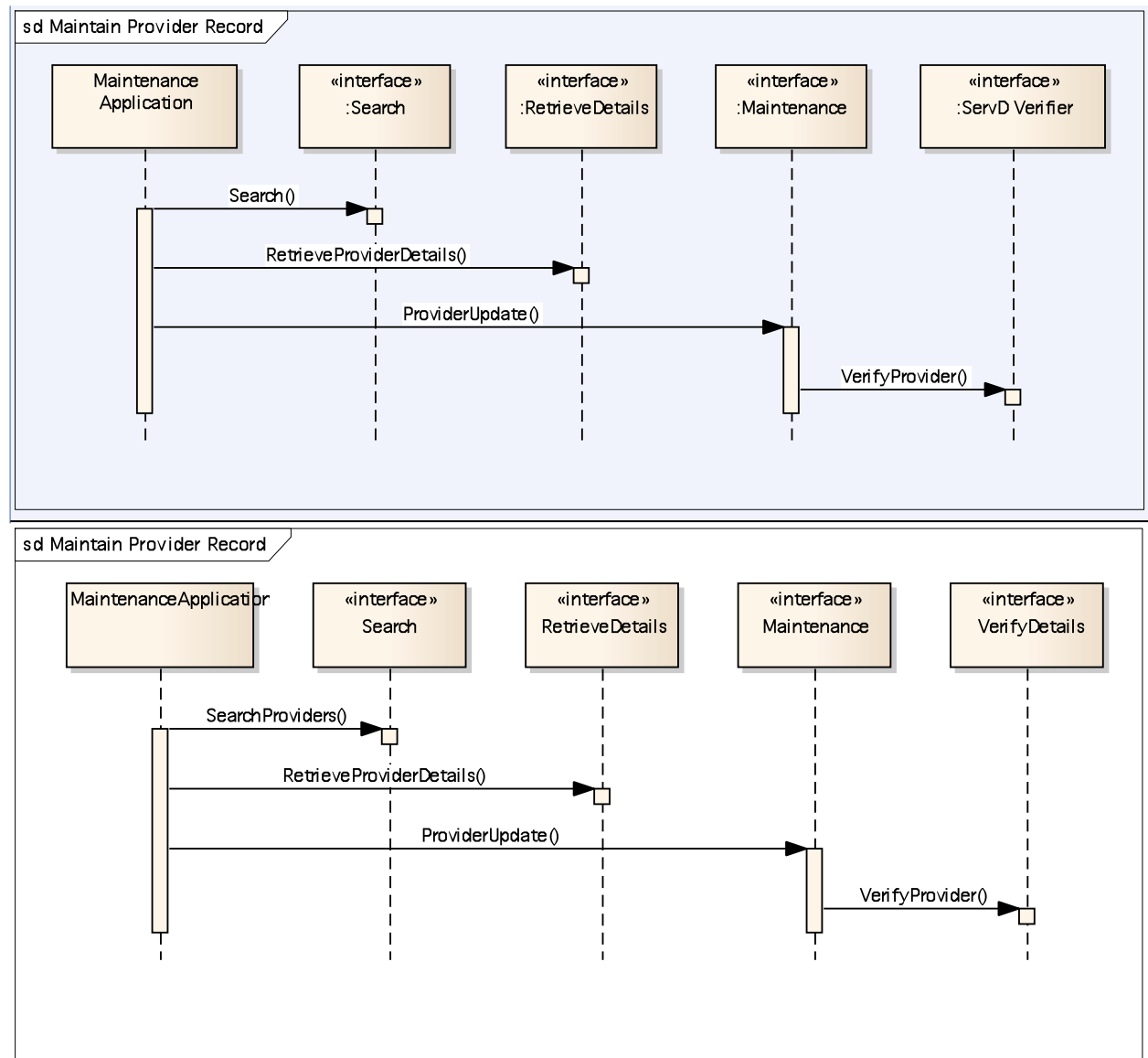
8.2.4 Content Management – Maintain Site Record

In this use case an Author updates a Site record with an Attribute that requires external verification and approval from an Author for the change.

sd Update Site with Verify Details and Approver

Author   Approver   Maintenance Application   «interface» RetrieveDetails   «interface» Maintenance

()

RetrieveSiteDetails()

SiteUpdate()

()

GetRecordsRequiringApproval()

RetrieveSiteDetails()
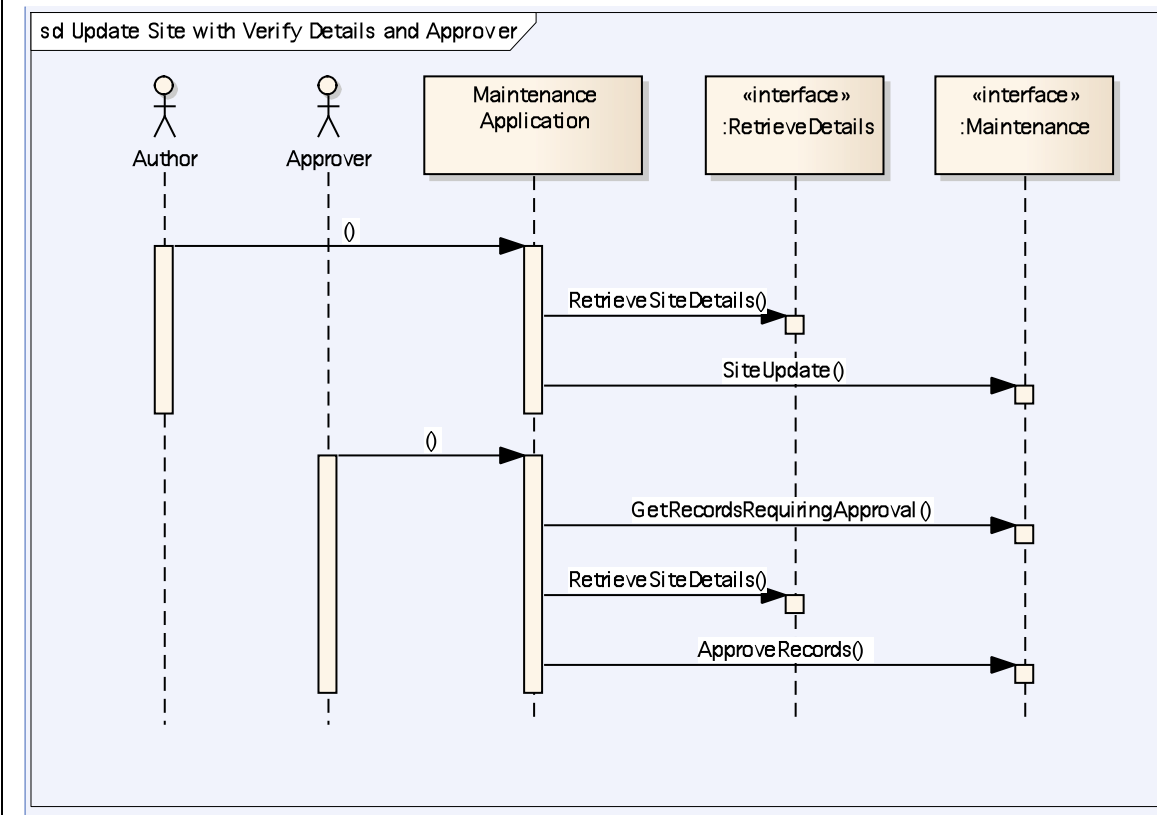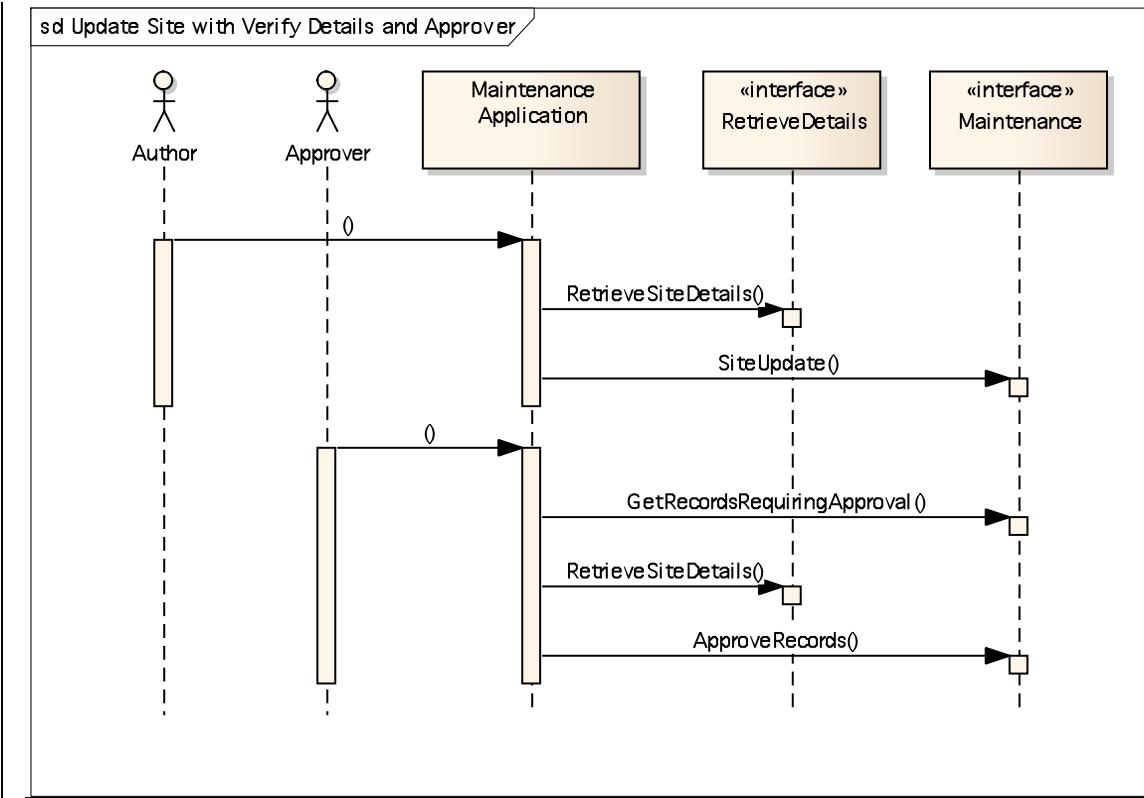
ApproveRecords()

The Author retrieves then updates the Site Details. The Approver then gets the records to Approve, retrieves the specifics of the change to verify the complete context of the update, then approves the changes.

8.2.5 Verify Organization Attribute

When an Organizations attribute is set that is configured to be externally verified the Maintainable ServD Core must call the configured ServD Verifier to check the validity of the attribute to the Organization.
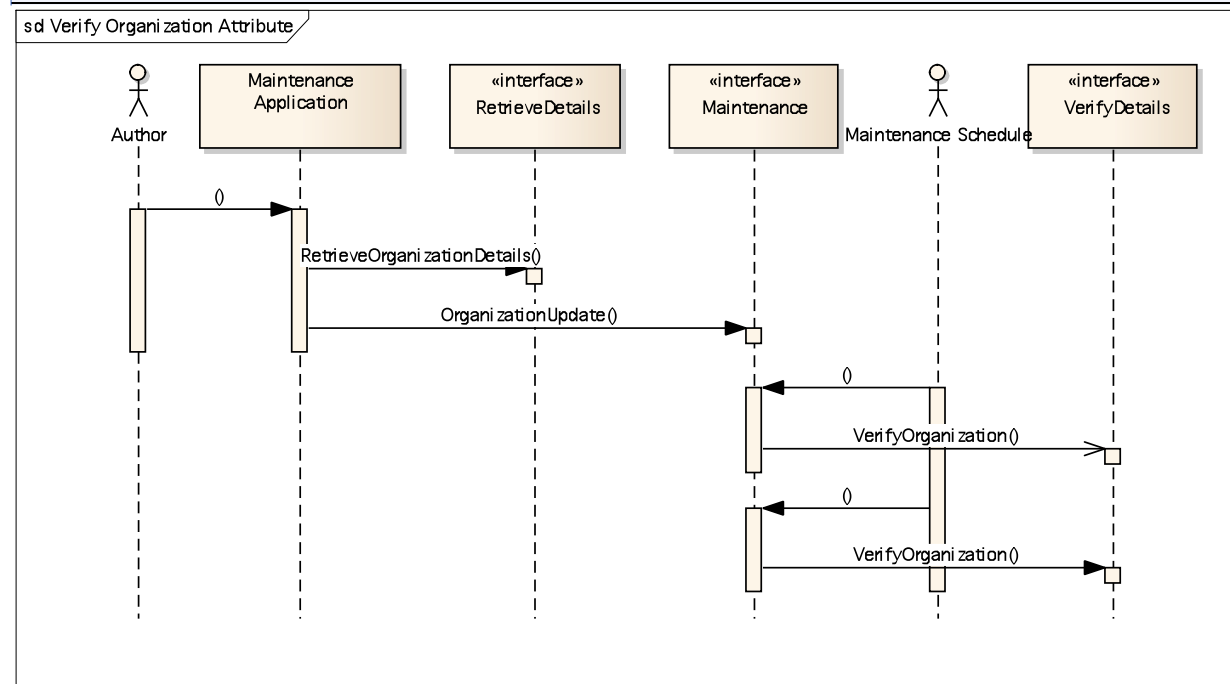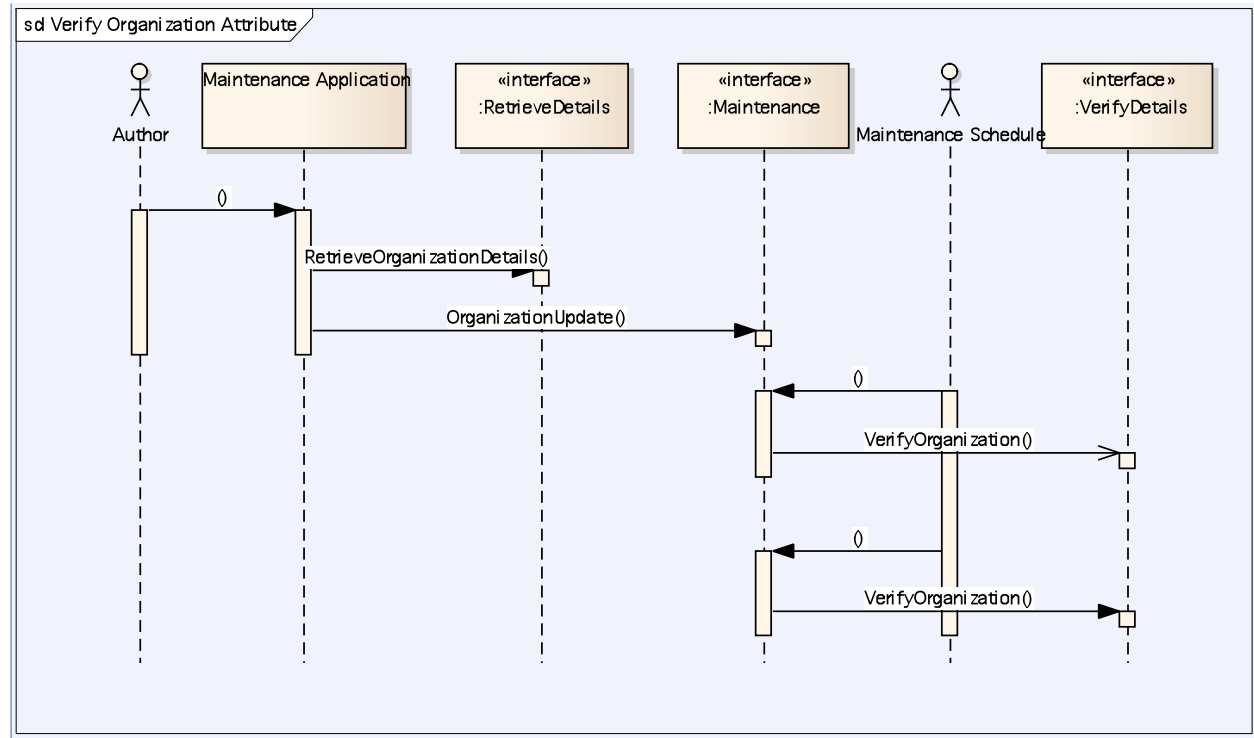
The call to the Verification Details interface may be done after the commit of the Organization Update.

This verification may also be performed on a regular Maintenance schedule to ensure that the latest information is kept up to date. In this example the verify Organization has been called twice (this may be a month apart).

8.2.6 Reject Change Workflow

To facilitate content management workflow the capability for modifications to be rejected by an author is required.
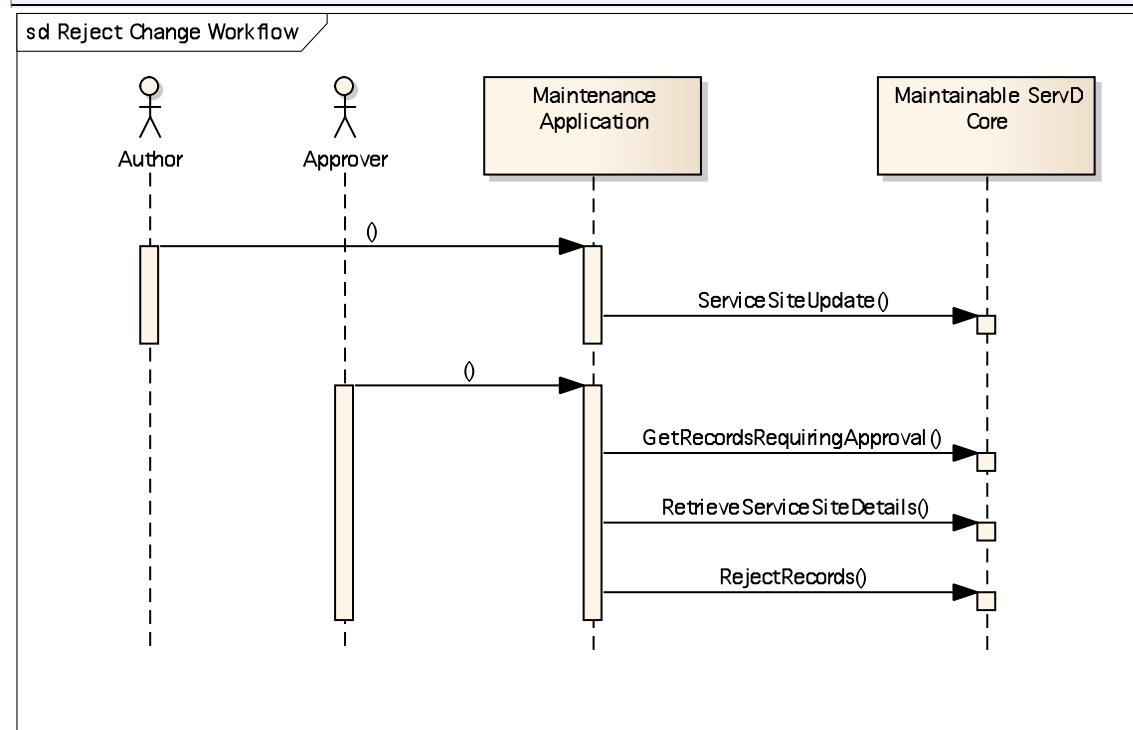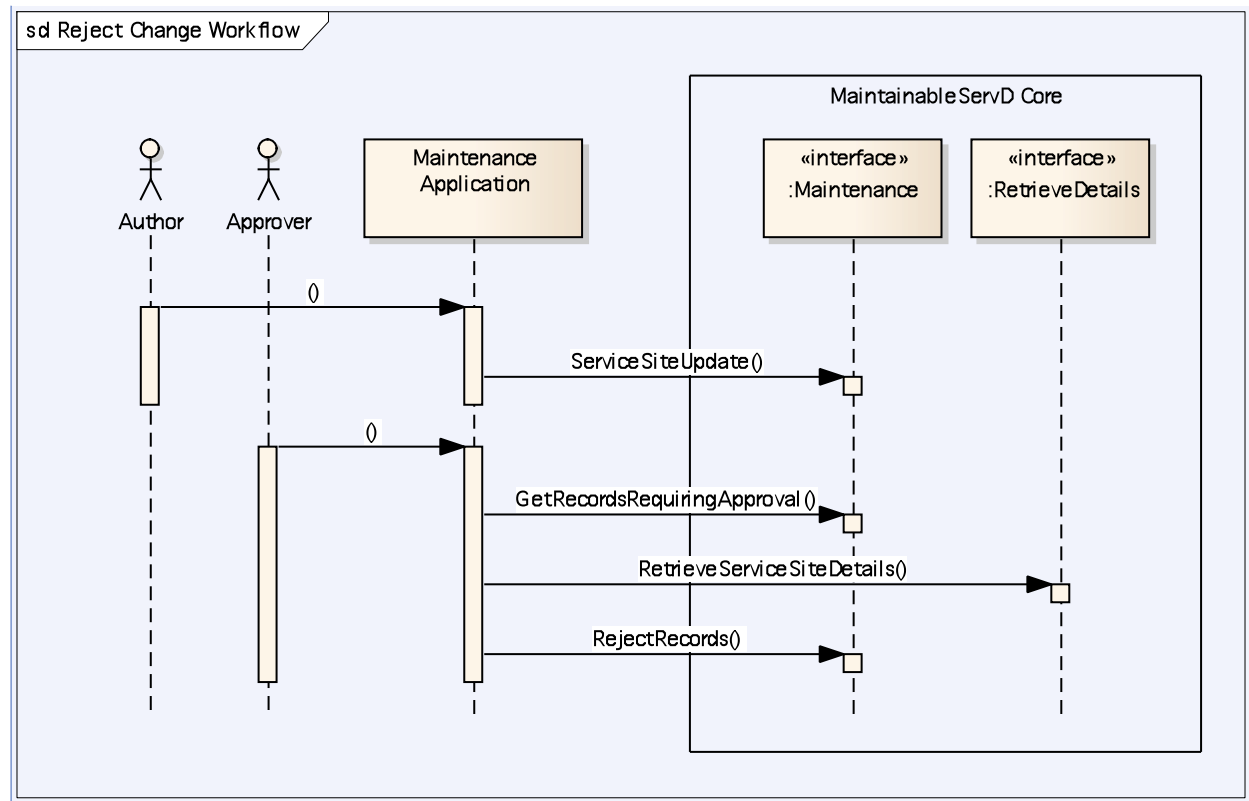
The Author uses the Maintenance Application to update some site details. This will mark the updated Site Records with a SuccessRequiresModeration Status. The current data remains intact and returned by search results.

The Approver then discovers that there is a Record requiring Approval via the GetRecordsRequiringApproval

method. In this example the Approver inspects the Site Details more thoroughly using the RetrieveSiteDetails and then decides that the change is not correct and Rejects the Records that were updated.

The RecordRequiringApproval objects that are returned by the call to GetRecordsRequiringApproval contain the RetrieveDetailsURL that must be used when retrieving the Service Sites complete details (ensuring the GetUnmoderatedDataWhereAvailable parameter is set to true).

8.2.7 Read Reference List (CTS2)

All Reference Lists used by the ServD specification are accessed through the ServD Locator which indicates the location of a CTS2 value set compliant component.
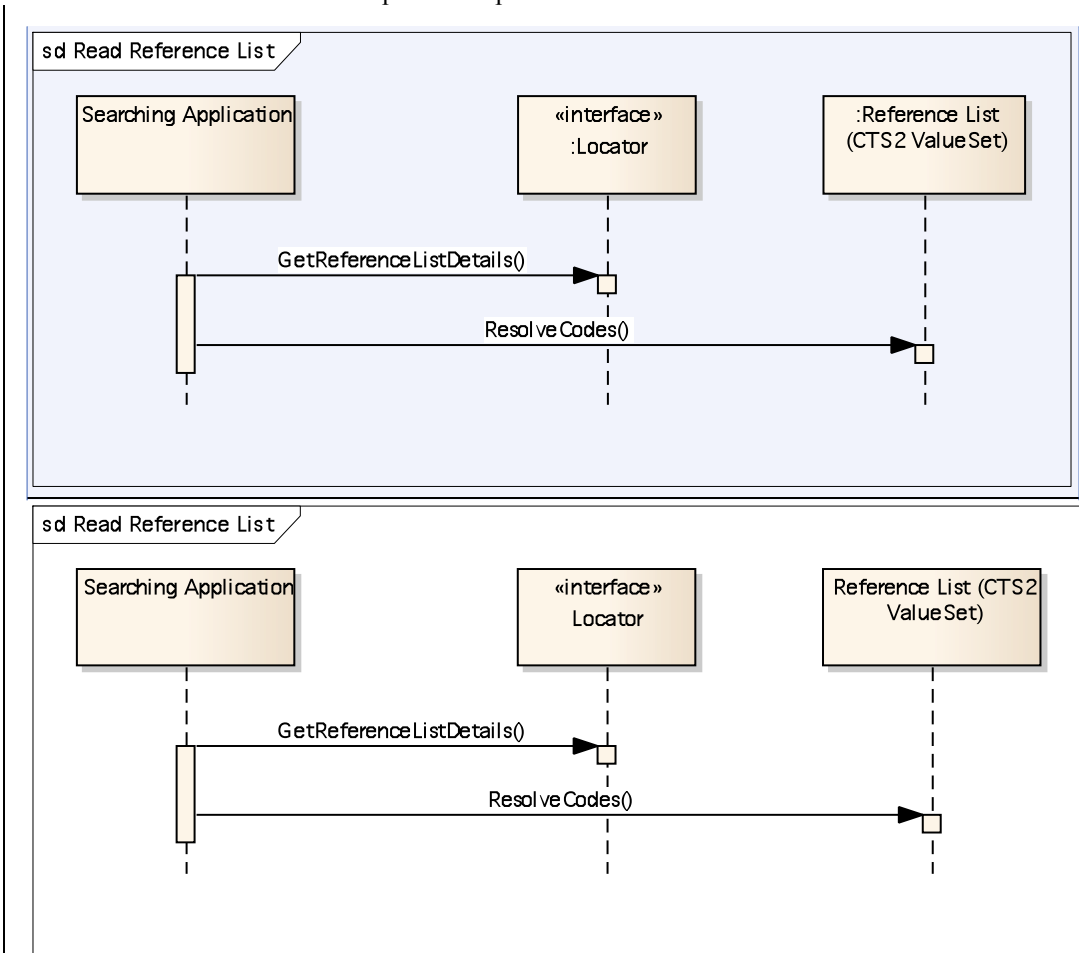
A Searching Application calls the Locator to retrieve the network location (address) of the Reference List so that it can interact with it. The network location at the provided address must expose a REST CTS2 value set interface to be able to interact with the codes.

This address may be cached within the Searching Application, but must be updated at regular intervals to ensure it is up to date. The values from the Reference List may be cached based on the information in the value set results.

Example Results from the GetReferenceListDetails:

Code Type	URL	Description
ServD.AP	http://informatics.mayo.edu/cts2/services/phinvads	Address Purpose
ServD.ETH	http://informatics.mayo.edu/cts2/services/phinvads	Ethnicity
ServD.GEN	http://informatics.mayo.edu/cts2/services/phinvads	Gender

The actual value set that is returned will be determined by the ServD Federation's Profile. For example the ServD.GEN reference list could be configured to be served by the PHVS_AdministrativeGender_HL7_V3 value set.

8.2.8 Content Manager Action

During routine content review/management activities a Content Manager may need to approve updates. This is very similar to the content management performed by an Approver; however a Content Manager typically has a wider range of permissions to approve updates.

In this example an Author updated a Service Site to have an incorrect Service Category Associated, and the Content Manager corrects this.
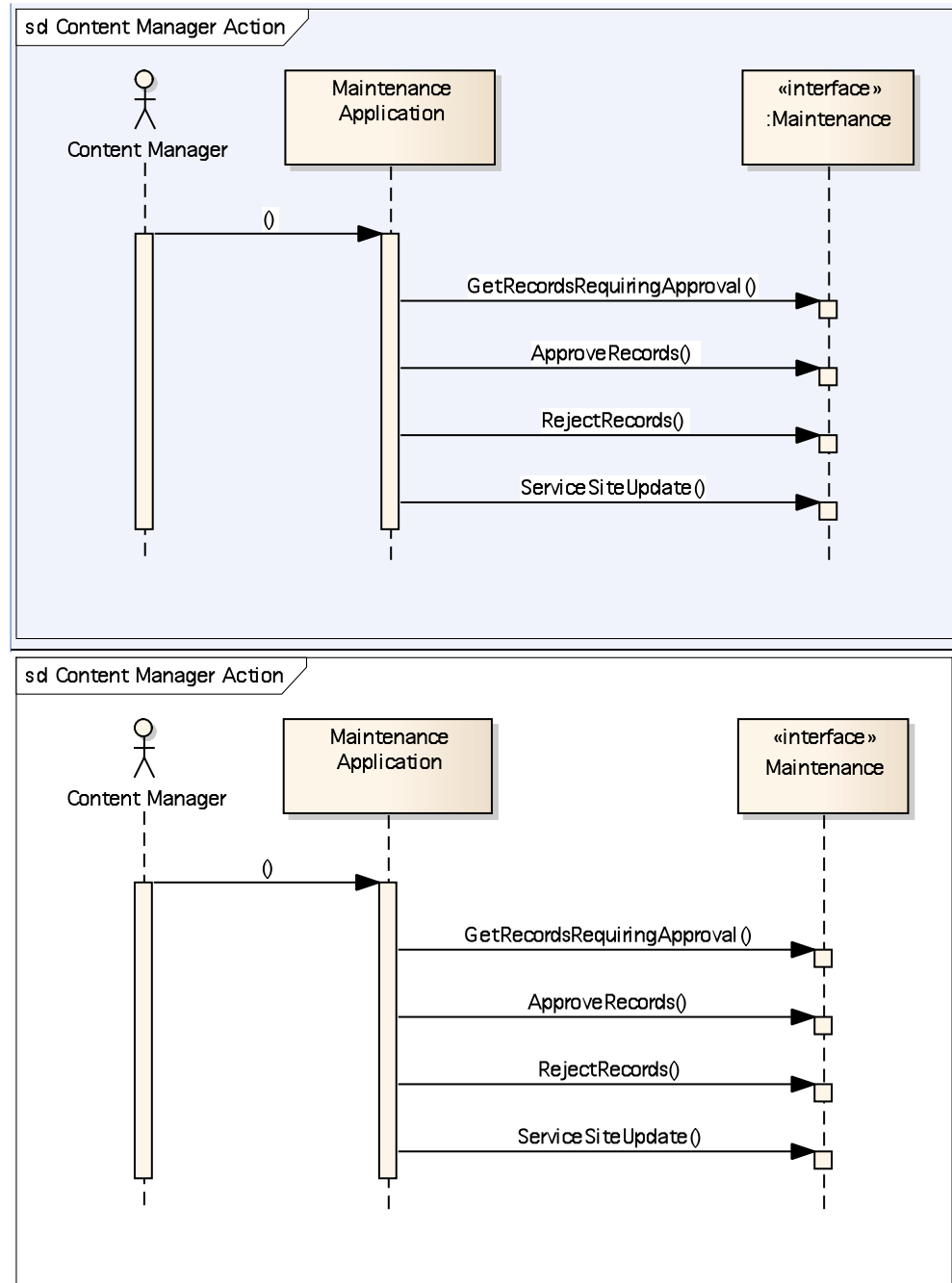
The Content Manager discovers the update through the GetRecordsRequiringApproval method; he approves changes to the Service Site Setting, rejects a change to Coverage Area, and then updates the Service Site's Service Types collection.

8.2.9 Create New Organization Workflow

Creating a new Organization requires the approval of a Content Manager.
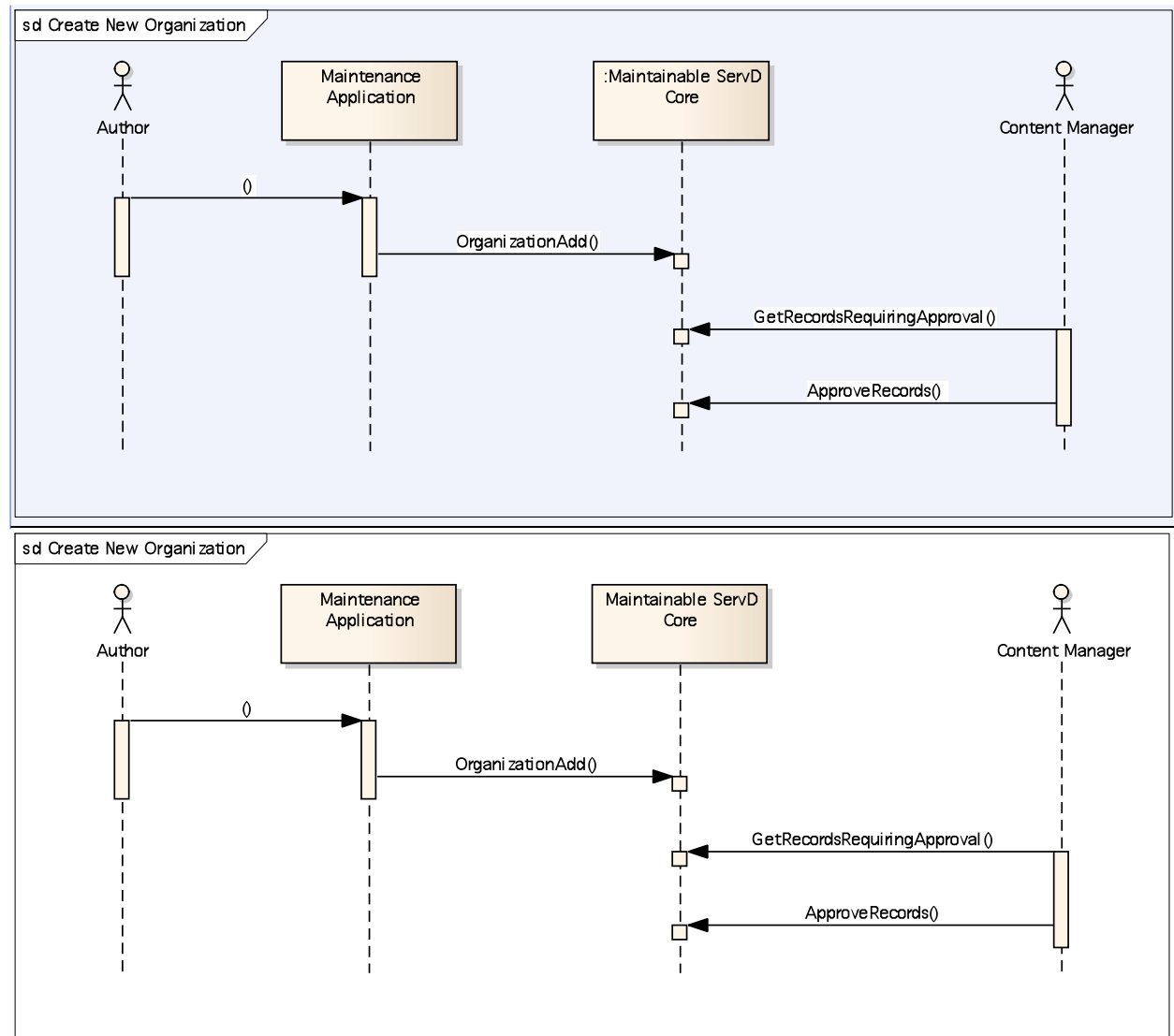
The Creation of new Providers also requires the approval of a Content Manager.

8.2.10 Associating a Provider with an Organization

Once a Provider is created it needs to be associated with one or more Service Sites.
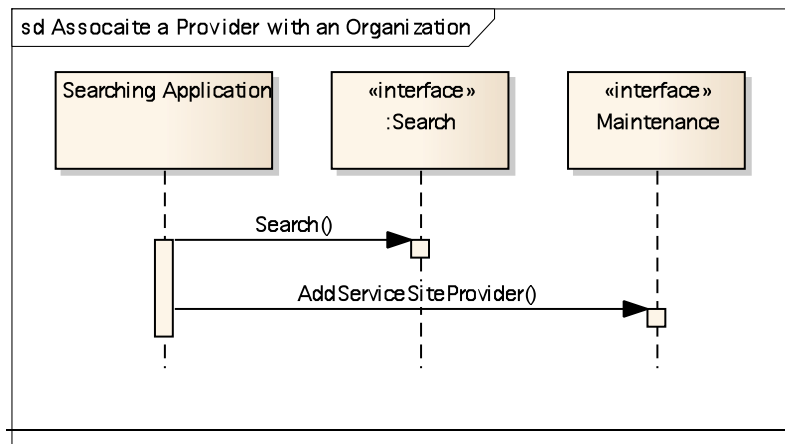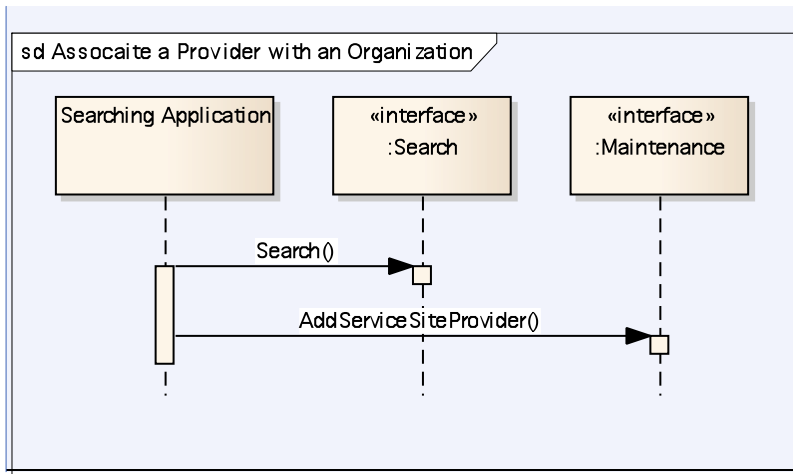
This is performed by searching for the Provider and Also the Service Site so that the Record Ids required to make the association are available. Then the association is created by calling the AddServiceSiteProvider method.

This operation is also able to include any Identifiers that are specific to this association.

I.e. Australian Providers can have different Provider Numbers when providing services at different Sites.

8.2.11 Search for Services with Data Paging

When a Searching Applications makes a search for information sometimes there are a large number of potential matches to the search criteria requested. In these situations the retrieving of all this information is not desirable for many reasons, and a mechanism to load this data in a controlled manner is required.

sd Search with Data Paging

Searching Application | Local Cache | :ServD Core

Search()
Cache Organization details()
MoreServiceSites()
Cache Site details()
Search()

sd Search with Data Paging

Searching Application | Local Cache | ServD Core

Search()
Cache Organization details()
MoreServiceSites()
Cache Site details()
Search()
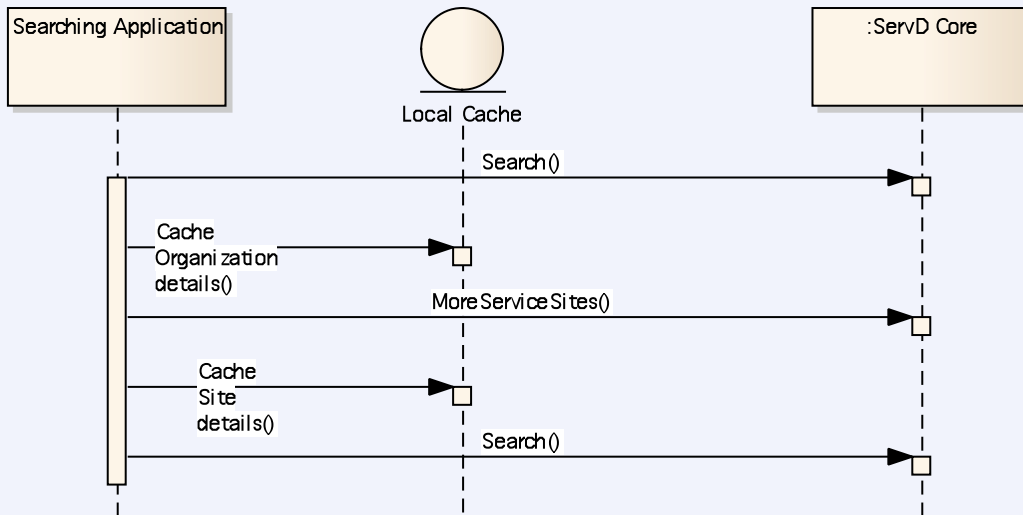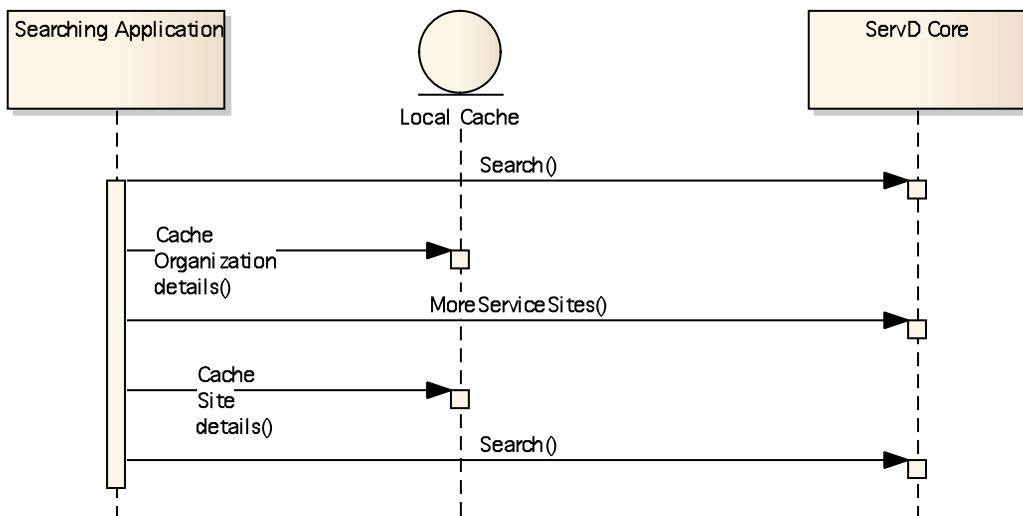
The data that is returned by the Search is returned in pages of data. To continue to the next page of data the Searching Application calls the appropriate "More" method. In this example: MoreServiceSites.

Note: The call to the MoreServiceSites must be made with the same search parameters that the initiating Search was made. This permits the Component that is providing the Search interface an option to retain state should it choose to and identify the same search and continue the results from where it was up to. This is likely to be relevant for implementers of the Federated Search in the ServD Locator.

The Searching Application can cache the details of the searches, particularly for the Organization and Site Names/basic details so that when displaying information to the user about Services the application does not have to search again for the summary details of these entities.

8.2.12 API Key Usage (Optional)

All of the methods in the ServD specification include the API key as an optional parameter to permit ServD instances to be able to restrict calling applications.
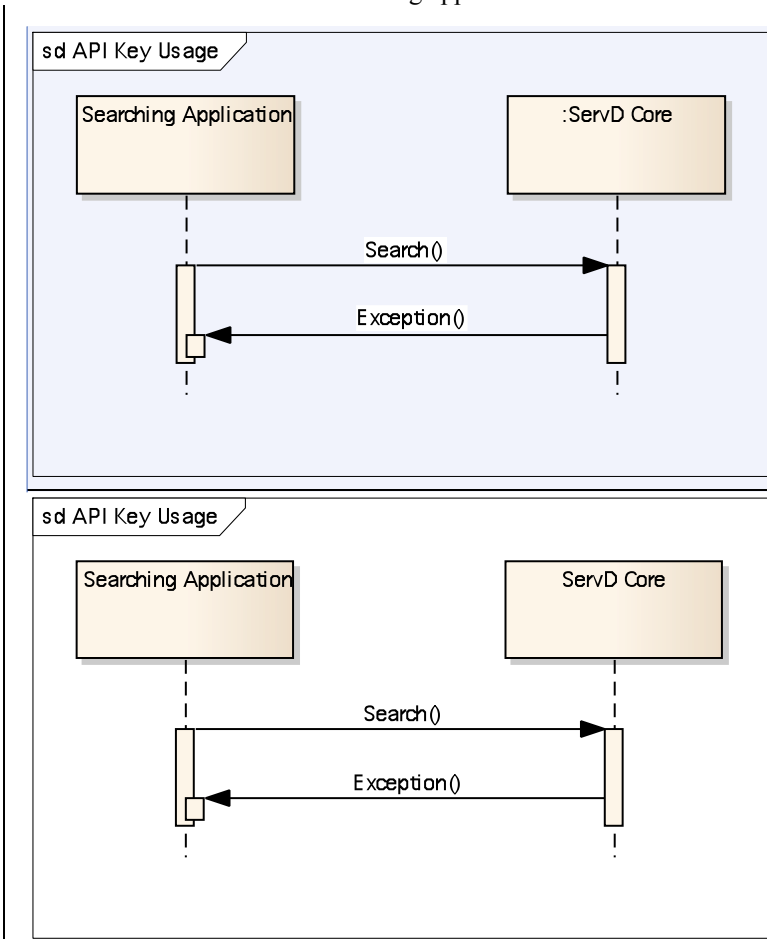
A Searching application makes a call to a ServD method (Search) using an API key that is not registered with the ServD Core.

The ServD core determines that this API key is not registered and throws a SOAP exception.

Informative Note: The use of these keys is often related to billing or constraining of applications permitted to connect to the ServD component.

8.3   Profile Binding

In order to deploy an interconnected set of ServD instances a shared profile must be agreed between all participants.
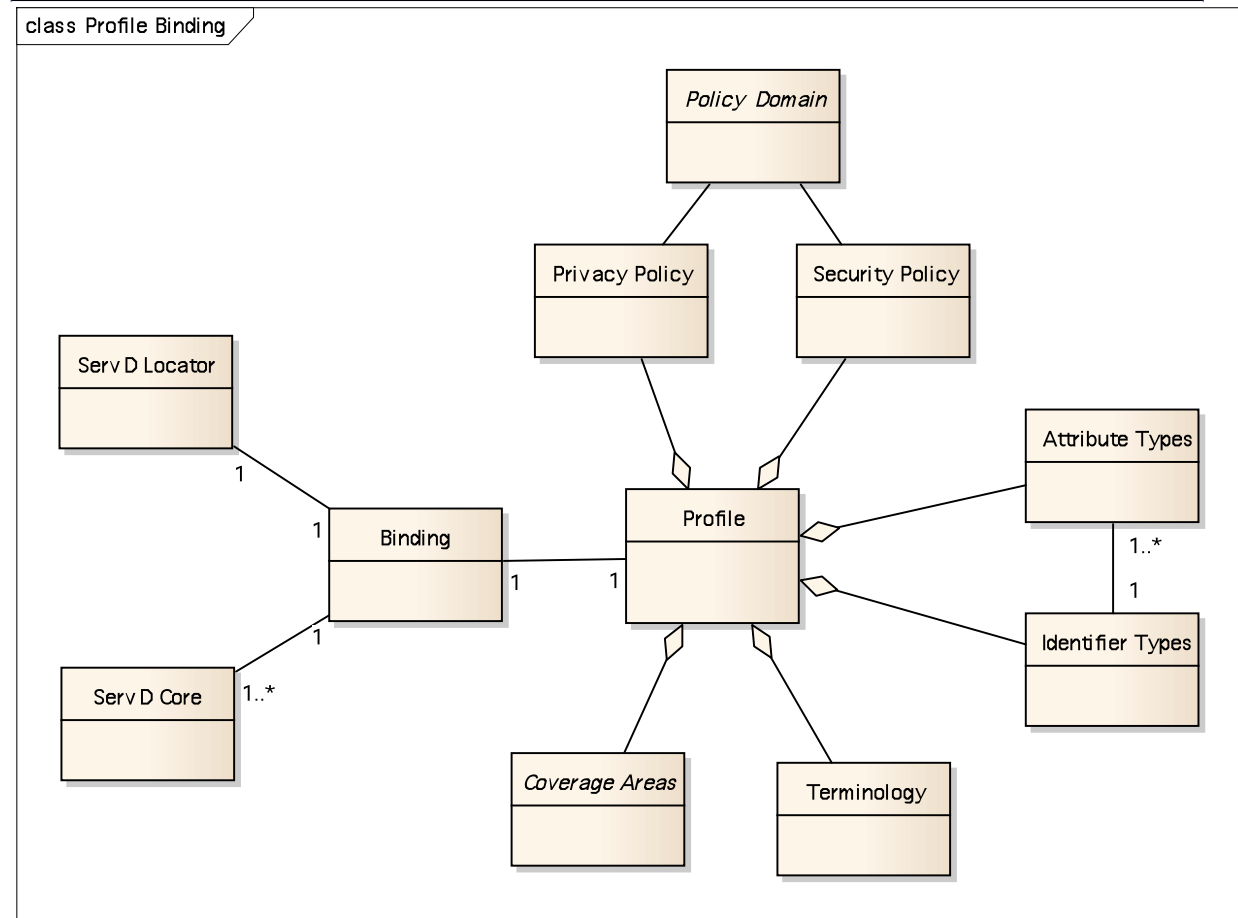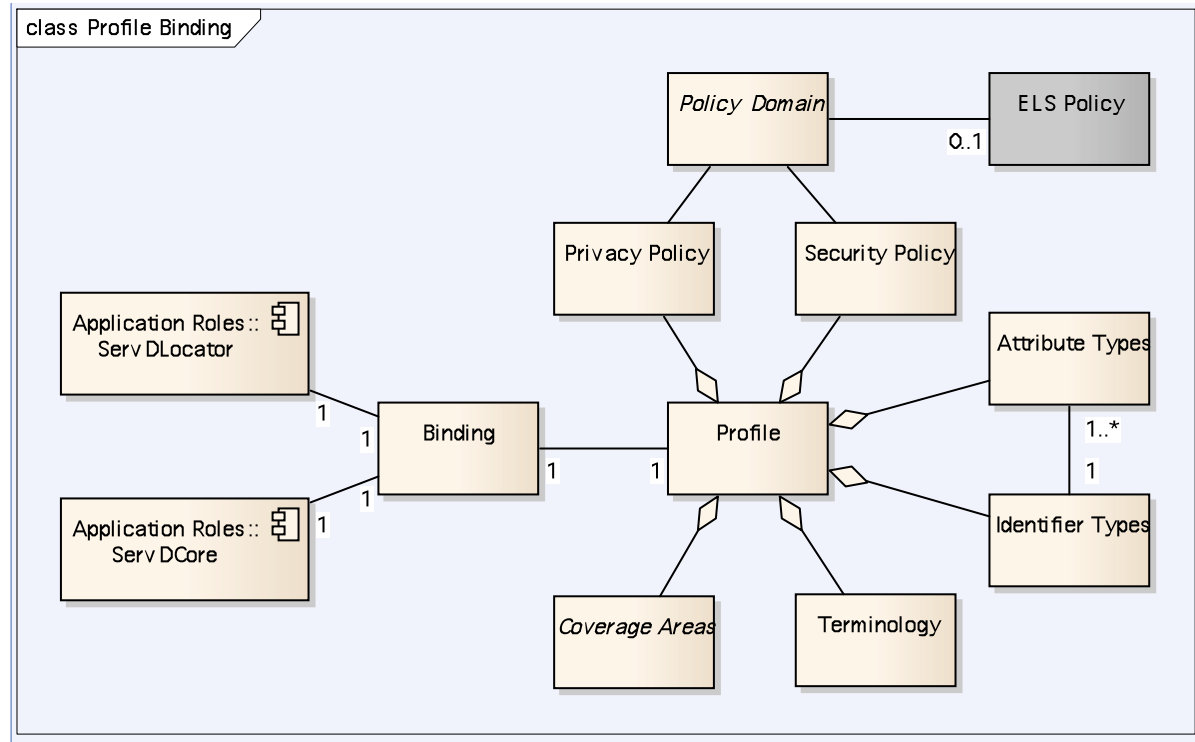
Note: The Profile binding does not need to incorporate an ELS, even if one or more are in use. An ELS is loosely coupled from the ServD Federation and can be managed within its own Policy framework.

8.3.1 Terminology

A base set of reference value sets must be defined by the ServD Locator which is exposed as a set of CTS2 compliant interfaces. If ServD Core instances use different value-sets then reference data must be mapped to/from the profile defined values when exposing data via the ServD interfaces.

This ensures that implementers of Searching Applications are not impacted by differing internal value sets.

Any Maintenance Application must publish its data using the reference value sets of the Profile, not internal data values.

The use of the CTS2 standard for these value sets provides for standard mapping services and also standard ways to keep reference sets up to date.

The Contact Point Type (ServD.CPT) that is expected to be used for Web Contact References should be defined in the ServD Federation Profile.

8.3.2 Federation and Coverage Areas

Multiple ServD Instances may be linked in a Federation. Each Instance may be separately administered on the basis that they are linked with a common Policy framework. Each ServD instance hosts a cohort of Organizations, Sites or Providers and the Coverage Areas they are providing services to. The Coverage Areas of each ServD instance may overlap with others, particularly where boundaries overlap. The Locator maintains an index of ServD Instances associated with each Coverage Area.

The properties used to define a Coverage Area are known as Discriminators, which are defined by Policy. Discriminators may relate to:

- Locations; for example a state or regional boundary

- Specialties; for example, a national ServD instance for physiotherapists.

- Languages; for example a Spanish language directory

Note: It is expected that Coverage Areas may often overlap. In this event a single Search request may generate calls to multiple ServD Instances.
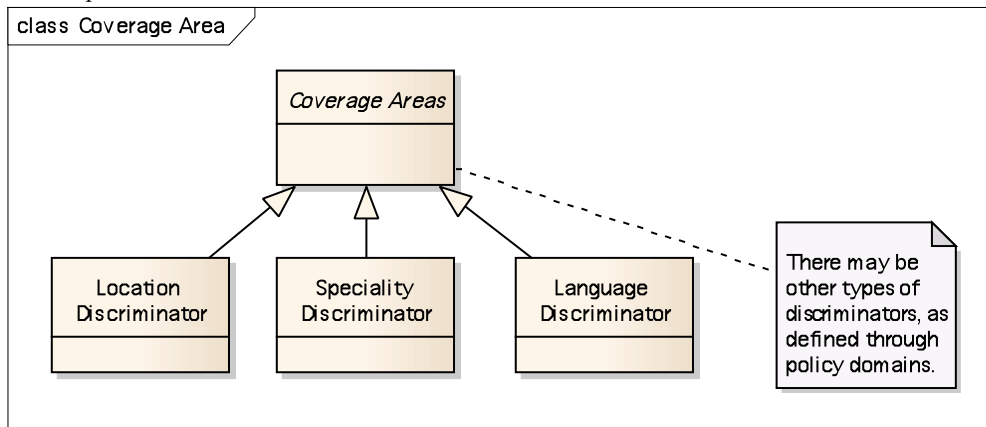
8.3.3 Privacy and Security Policy

As the requirements for Privacy and Security Policy are expected to be different for each ServD Federation, this specification does not constrain how this is to be implemented.

However, every MaintainableRecord in the Information Model may be tagged with Policy; as is consistent with the

Policy of the ServD Federation.

The Policy Domain will determine access to records based on some determination of:

- Record type

- Record Privacy Policy Tag

- Record Data (e.g., which organization)

- Security Role

- Method

Note: The API key can also be used alongside the security policy; however it is not classified as a form of user access security.

8.3.3.1 Security Roles

The Security Roles that are defined by the ServD specification are:

Role	Description
Author	A user that is authorized to modify data. This information is required to be approved before it is "committed" to the general searching methods.
Approver	A user that is authorized to modify and approve/reject modifications to data.
Content Moderator	A user that is authorized to modify and approve/reject modifications to data. Users with this role normally have a greater amount of control on the data and are often the operator of the ServD instance, but could also be an approved 3rd party moderator.

8.3.4 Authentication

Where secure access is required to perform secure searching or maintenance access, the mechanism to perform this can be defined by the implementation of the Maintainable ServD Core or the ServD Locator. This may be

- via digital certificates and https,

- based on IP address ranges,

- or any other mechanism that is deemed acceptable to the Profile.

8.3.5 Authorization

Access rules and roles to records will be defined by the Profile.

When unauthorized access is performed, including an attempt to update information, SOAP error messages must be raised to notify the calling application.

8.3.6 Attributes and Identifiers

As the information model supports the use of extensible Attributes/Identifiers on the key objects, a base list of Attribute Types and Identifier Types may be defined in the Profile for the ServD Federation.

These may also include the Attribute Types which require external verification, and the appropriate Identifier Type used for the verification.

A Profile may specify that if an Identifier Record with a specific Identifier Type is created, then an associated Attribute Record may also automatically be created.

Example: Australian Healthcare Providers have an Identifier (HPI-I) which is stored within a government register.. The Profile could dictate that if a Provider record was allocated an external Identifier (HPI-I), then it will automatically associate an Attribute type "Has HPI-I"; which can be verified with the government register.

This would then permit searches for verified Providers, without knowing the ID.

Example: Health Insurance companies may have different types of cover that they provide to different services. A

single Service Site can have a single Identifier for use with the Health Insurance company, but can have 3 different Attributes for the types of cover that they support.

8.4   Configuration

Although specific implementations of the individual components is likely to require more than this list provided, it is expected that these are the minimum required configuration parameters required to deploy each component type.

8.4.1 ServD Locator

When deploying a ServD Locator the following must be configured:

Item	Description
CTS2 Reference Lists	All of the required CTS2 reference lists must be configured. This is the URL for the CTS2 valueset interface. Note: The use of this standard provides facilities for mapping values if required, but is not mandated by this specification. ServD requires this interface (valueset) to be able to retrieve descriptions for the code values which are stored in the ServD data-store. The available Interface Formats that are supported for the reference lists must be defined. Example: CTS2-REST, CTS2-WSDL
Security	The required security details based on the Profile used by the ServD Federation

8.4.2 ServD Core

When deploying a ServD Core the following must be configured:

Item	Description
ServD Locator URL	The location of the ServD Locator to register with
ServD Verifier Mappings	A List of the Attribute Types that are to be externally verified by a ServD Verifier and the address of the ServD Verifier component, and also the Identifier Type that must be provided for verification purposes. *(for Maintainable ServD Core instances)* **Note**: The security requirements for the Verifier may be different to those described by the ServD Federation's Profile.
CTS2 Reference List Mappings	If not using the standard list of items from the ServD Locator, then a mapping to/from these must be included.
Security	The required security details based on the Profile used by the ServD Federation
Authorization / Privacy Policy	The privacy and authorization policy as defined by the Profile used by the ServD Federation must be implemented.

Item	Description
Default Search Purpose Types	For Names, Contact Points and Addresses where the Purpose Type is required to create summarized items while searching. If no type is specified, or the specified type is not available, these are the types that will be selected.

8.4.3 ServD Verifier

When deploying a ServD Verifier the following must be configured:

Item	Description
ServD Verifier Mappings	A List of the Attribute Types that can be verified. *(Optionally the lists of valid values that are verifiable)*
Security	Any security requirements that may be required by the ServD Verifier. Note: This may be different to the ServD Federation's security policy requirements as these verification facilities normally run independently in a closed environment.

8.4.4 Maintenance Application

When deploying a Maintenance Application the following must be configured:

Item	Description
Maintainable ServD Core URL	The location of the **Maintainable ServD Core**
CTS2 Reference List Mappings	If not using the standard list of items from the ServD Locator, then a mapping to/from these must be included.
Security	The required security details based on the Profile used by the ServD Federation
Authorization / Privacy Policy	The privacy and authorization policy as defined by the Profile used by the ServD Federation must be implemented.

8.4.5 Searching Application

When deploying a Searching Application the following must be configured:

Item	Description
ServD Locator URL	The location of the **ServD Locator** to request Federated Searches and/or Locate **ServD Core** instances for detailed searching. *(This provides the access to the addresses of the CTS2 reference lists)*
CTS2 Reference List Mappings	If not using the standard list of items from the ServD Locator, then a mapping to/from these must be included.

Item	Description
Security	The required security details based on the Profile used by the ServD Federation

8.5   SOA Interfaces

8.5.1   Interface Definition Conventions

Throughout this document the usage of the term "datatype (collection)" can be interpreted as an array of datatype which has a cardinality of [0 ... *]

Example: If an operation returns a String (collection), then the operation will return an array of string with 0 or more elements.

Example: If an entity is referenced by OrganizationSearchParameters (collection), then the entity has a 0 ... * association with the OrganizationSearchParameters.

```
class Interface

        «interface»
          Locator
+   GetReferenceListDetails(ReferenceListTypes, APIkey) :ReferenceListDetail[]
+   LocateSearchEndpointsForCoverageArea(CoverageAreaCode, APIkey) :String[]


        «interface»
      LocatorMaintenance
+   RegisterSearchEndpointForCoverageArea(Address, CoverageAreaCodes, APIkey) :LocatorUpdateResult
+   UnregisterSearchEndpointForAllCoverageAreas(Address, APIkey) :LocatorUpdateResult
+   UnregisterSearchEndpointForCoverageArea(Address, CoverageAreaCodes, APIkey) :LocatorUpdateResult


        «interface»
          Search
+   MoreOrganizations(SearchParameters, ControlParameters, RequestPage, APIkey) :OrganizationList
+   MoreProviders(SearchParameters, ControlParameters, RequestPage, APIkey) :ProviderList
+   MoreServiceSites(SearchParameters, ControlParameters, RequestPage, APIkey) :ServiceSiteList
+   MoreSites(SearchParameters, ControlParameters, RequestPage, APIkey) :SiteList
+   Search(SearchParameters, ControlParameters, APIkey) :SearchResults


        «interface»
       RetrieveDetails
+   RetrieveOrganizationDetails(OrganizationId, IncludeSiteData, IncludeServiceSiteData, IncludeProviderData, GetUnmoderatedDataWhereAvailable, APIkey) :Organization[]
+   RetrieveProviderDetails(ProviderId, GetUnmoderatedDataWhereAvailable, APIkey) :Provider[]
+   RetrievePublicKeys(SiteId, ServiceSiteId, ProviderId, APIkey) :ServD.Results.PublicKey[]
+   RetrieveServiceSiteDetails(ServiceSiteId, GetUnmoderatedDataWhereAvailable, APIkey) :ServiceSite[]
+   RetrieveSiteDetails(SiteId, IncludeServiceSiteData, IncludeProviderData, GetUnmoderatedDataWhereAvailable, APIkey) :Site[]


        «interface»
        Maintenance
+   AddServiceSiteProvider(ServiceSiteId, ProviderId, details, APIkey) :UpdateRecordResult
+   ApproveRecords(Identifiers, APIkey) :UpdateRecordResult[]
+   DeleteRecords(Identifiers, APIkey) :DeleteRecordResult[]
+   GetRecordsRequiringApproval(Page, PageSize, APIkey) :RecordRequiringApproval[]
+   OrganizationAdd(details, APIkey) :UpdateRecordResult[]
+   OrganizationUpdate(details, APIkey) :UpdateRecordResult[]
+   ProviderAdd(details, APIkey) :UpdateRecordResult[]
+   ProviderUpdate(details, APIkey) :UpdateRecordResult[]
+   RejectRecords(Identifiers, APIkey) :UpdateRecordResult[]
+   ServiceSiteAdd(SiteId, details, APIkey) :UpdateRecordResult[]
+   ServiceSiteUpdate(details, APIkey) :UpdateRecordResult[]
+   SiteAdd(OrganizationId, details, APIkey) :UpdateRecordResult[]
+   SiteUpdate(details, APIkey) :UpdateRecordResult[]
+   UpdateServiceSiteProvider(details, APIkey) :UpdateRecordResult
+   UploadServiceSiteCertificate(ServiceSiteId, PublicKey, APIkey) :UpdateRecordResult
+   UploadSiteCertificate(SiteId, PublicKey, APIkey) :UpdateRecordResult


        «interface»
        VerifyDetails
+   VerifyOrganization(AttributeType, AttributeValue, OrganizationIdentifier, Name, addresses, contacts, APIkey) :ServD.Results.AttributeVerificationResult
+   VerifyProvider(AttributeType, AttributeValue, ProviderIdentifier, FamilyName, GivenNames, addresses, contacts, APIkey) :ServD.Results.AttributeVerificationResult
+   VerifySite(AttributeType, AttributeValue, SiteIdentifier, Name, addresses, contacts, APIkey) :ServD.Results.AttributeVerificationResult
```

8.5.2   Locator Interface

The **Locator** interface provides a way to locate ServD Core Instances that expose the **Search** interface for a requested Coverage Area.

In simplest terms it is an indexer of **ServD Core** instances based on Coverage Areas. Each Coverage Area may have

one or more relationships with any other Coverage Area, as defined by the CTS2 reference list 'ServD.CA', which can be found at the URL returned by a call to **GetReferenceListDetails('ServD.CA')**

8.5.2.1 Method: LocateSearchEndpointsForCoverageArea

This method locates web service URL(s) for ServD Core instances that implement the **Search** Interface
Finds the URL of the Web Server that is providing the search interface details for the specified information.

Return Value: String (collection)

A list of **ServD Core** URLs that implement the **ServD Search** interface.

Parameter Name	Data Type	Description
CoverageAreaCode	String	The Code Value of the Coverage Area to find **ServD Core** instances that have relevant data. **cts2:**ServD.CA
APIkey	String	The API key may be included by ServD implementations that wish to restrict access to the interface to those applications that have registered with a key. This is a similar mechanism to how Google Maps, Bing Maps, Facebook apps etc. control application access.

8.5.2.2 Method: GetReferenceListDetails

This method is used to locate the URL(s) of the CTS2 reference list(s) of the defined type(s). These can be used to populate drop down lists, create look-a-head-while-typing user interactions, or validate user input prior to issuing a call to the Search methods.

Return Value: ReferenceListDetail (collection)

A collection of the details of the CTS2 reference list(s), including their web address URL(s) so that each can be accessed by applications to retrieve the reference codes/descriptions.

Parameter Name	Data Type	Description
ReferenceListTypes	String (collection)	The collection of CTS2 domain(s) of lookup list details to be retrieved.
InterfaceFormat	String	The Interface format of the interfaces supported by the URLs returned. The valid values for this field must be defined by the Federations Profile Binding (as described in section 8.4.1). Example: CTS2-REST

Parameter Name	Data Type	Description
APIkey	String	The API key may be included by ServD implementations that wish to restrict access to the interface to those applications that have registered with a key. This is a similar mechanism to how Google Maps, Bing Maps, Facebook apps etc. control application access.

8.5.3 LocatorMaintenance Interface

The **Locator Maintenance** interface exposed by the **ServD Locator** provides a **ServD Core** the capability to update its Coverage Area information.

8.5.3.1 Method: RegisterSearchEndpointForCoverageArea

Register a **ServD Core** instance with the **ServD Locator**

Return Value: LocatorUpdateResult

A simple summary of the number of records that were affected.

Parameter Name	Data Type	Description
Address	String	The URL of a **ServD Core** that implements the **ServD Search** interface
CoverageAreaCodes	String (collection)	A Collection of Coverage Areas to associate with the specified **ServD Core** **cts2:**ServD.CA
<u>**RemoveExistingEntries**</u>	<u>Boolean</u>	<u>If this parameter is true, then any entries not in the list provided will be removed from the configuration.</u>
APIkey	String	The API key may be included by ServD implementations that wish to restrict access to the interface to those applications that have registered with a key. This is a similar mechanism to how Google Maps, Bing Maps, Facebook apps etc. control application access.

8.5.3.2 Method: UnregisterSearchEndpointForAllCoverageAreas

Remove all coverage areas allocated to this **ServD Core** endpoint.

Return Value: LocatorUpdateResult

A simple summary of the number of records that were affected.

Parameter Name	Data Type	Description

Parameter Name	Data Type	Description
Address	String	The URL of a **ServD Core** that implements the **ServD Search** interface that wishes to un-register all data from the Locator's index.
APIkey	String	The API key may be included by ServD implementations that wish to restrict access to the interface to those applications that have registered with a key. This is a similar mechanism to how Google Maps, Bing Maps, Facebook apps etc. control application access.

8.5.3.3 Method: UnregisterSearchEndpointForCoverageArea

Remove a set of CoverageAreaCode(s) from the index for a specific **ServD Core** endpoint.

Return Value: LocatorUpdateResult

A simple summary of the number of records that were affected.

Parameter Name	Data Type	Description
Address	String	The URL of a **ServD Core** that implements the **ServD Search** interface
CoverageAreaCodes	String (collection)	The Code(s) of the Coverage Area(s) to associate with **ServD Core** instance **cts2:**ServD.CA
APIkey		The API key may be included by ServD implementations that wish to restrict access to the interface to those applications that have registered with a key. This is a similar mechanism to how Google Maps, Bing Maps, Facebook apps etc. control application access.

8.5.4 Search Interface

The **Search** interface is the core interface in the **ServD** specification. It provides operations to search for lists of summary information about **Organizations**, **Sites**, **Services** and **Providers**.

Each record in the object model has its own moderation and security policy information associated, and the implementation of this Search interface must understand the structure of the Security Policy and apply this to any data that is returned by the results of the search.

The Location (URI) of the Service providing the **Retrieve Details** interface is provided in the results for each item returned by this **Search** interface.

8.5.4.1 Method: Search

Perform a search for information in the ServD Core.

When a Search is performed with a SearchControlParameters GetUnModeratedData value of False the ServD Core must return data that meets the following criteria:

1. The caller has access to the data

2. The data has a RecordStatus of Complete

   Except when the SearchInputParameter.ModifiedAfterDate field is provided, in which case:

   a. The data has a RecordStatus of Complete and the LastModifiedDate is after the ModifiedAfterDate

   b. Or the data has a RecordStatus of Deleted and the LastModifiedDate is after the ModifiedAfterDate

3. The data satisfies the search parameters provided

When a Search is performed with a SearchControlParameters GetUnModeratedData value of True the ServD Core must return data that meets the following criteria:

1. The caller has access to the data

2. The data has a Record Status of:

   a. Complete (and no pending change that the caller is an author or approver of)

   b. Any other state where the caller is an author or approver of the record.

2. The data satisfies the search parameters provided

Return Value: SearchResults

The returned list of items is not in a hierarchy of Organizations, Sites, Service Sites and Providers. They are returned as a simple list of objects with Ids to reference the other objects in the model.

This is done so that if a **Searching Application** is performing searches for Sites repeatedly they can do a single call to retrieve the Sites, then a secondary call to retrieve the Organizations based on a list of ids, better permitting control over what is requested and cache information in the Searching application, only requesting for new information if it doesn't already have it loaded.

Parameter Name	Data Type	Description
SearchParameters	SearchInputParameters	The set or filter criteria upon which the Search is to be based. In a Federated implementation it is recommended that Search Parameters include Discriminators used to locate appropriate Instances.
ControlParameters	SearchControlParameters	Search control options such as if Moderated data should be returned, and the page size to be used

Parameter Name	Data Type	Description
APIkey	String	The API key may be included by ServD implementations that wish to restrict access to the interface to those applications that have registered with a key. This is a similar mechanism to how Google Maps, Bing Maps, Facebook apps etc. control application access.

8.5.4.2 Method: MoreOrganizations

Retrieve the requested page of Organization summary items based on a previous search.

Return Value: OrganizationList

A page of Organization records to continue loading the results of a previous search.

Parameter Name	Data Type	Description
SearchParameters	SearchInputParameters	These search parameters ,must be the same as those passed into the initial search
ControlParameters	SearchControlParameters	These control parameters must be the same as those passed into the initial search
RequestPage	Integer	The number of the page to request to load (zero based index)
APIkey	String	The API key may be included by ServD implementations that wish to restrict access to the interface to those applications that have registered with a key. This is a similar mechanism to how Google Maps, Bing Maps, Facebook apps etc. control application access.

8.5.4.3 Method: MoreSites

Retrieve the requested page of Site summary items based on a previous search.

Return Value: SiteList

A page of Site records to continue loading the results of a previous search.

Parameter Name	Data Type	Description
SearchParameters	SearchInputParameters	These search parameters must be the same as those passed into the initial search
ControlParameters	SearchControlParameters	These control parameters must be the same as those passed into the initial search
RequestPage	Integer	The number of the page to request to load (zero based index)
APIkey	String	The API key may be included by ServD implementations that wish to restrict access to the interface to those applications that have registered with a key. This is a similar mechanism to how Google Maps, Bing Maps, Facebook apps etc. control application access.

8.5.4.4 Method: MoreServiceSites

Retrieve the requested page of Service Site summary items based on a previous search.

Return Value: ServiceSiteList

A page of Service Site records to continue loading the results of a previous search.

Parameter Name	Data Type	Description
SearchParameters	SearchInputParameters	These search parameters must be the same as those passed into the initial search
ControlParameters	SearchControlParameters	These control parameters must be the same as those passed into the initial search
RequestPage	Integer	The number of the page to request to load (zero based index)
APIkey	String	The API key may be included by ServD implementations that wish to restrict access to the interface to those applications that have registered with a key. This is a similar mechanism to how Google Maps, Bing Maps, Facebook apps etc. control application access.

8.5.4.5 Method: MoreProviders

Retrieve the requested page of Provider summary items based on a previous search.

Return Value: ProviderList

A page of Provider records to continue loading the results of a previous search.

Parameter Name	Data Type	Description
SearchParameters	SearchInputParameters	These search parameters must be the same as those passed into the initial search.
ControlParameters	SearchControlParameters	These control parameters must be the same as those passed into the initial search.
RequestPage	Integer	The number of the page to request to load (zero based index).
APIkey	String	The API key may be included by ServD implementations that wish to restrict access to the interface to those applications that have registered with a key. This is a similar mechanism to how Google Maps, Bing Maps, Facebook apps etc. control application access.

8.5.5 RetrieveDetails Interface

The **Retrieve Details** interface provides methods to retrieve the detailed information about Organizations, Sites, Service Sites and Providers. The Location (URI) of this interface is returned by the **Search** interface.

The data returned by these operations may be either the public information, or information that is still subject to approval as requested by the use of the **GetUnmoderatedDataWhereAvailable** flag being set to true. Records must be omitted from results as indicated by the policy defined by the **RecordPrivacyPolicy** value on the Record.

If a parent record is not able to be accessed, then any child object information will also not be accessible.

8.5.5.1 Method: RetrieveOrganizationDetails

Retrieve the complete set of details about the Organization(s) based on the Identifier(s) returned in a search.

Return Value: Organization (collection)

A collection of Organization records based on the requested identifiers, with the requested level of child objects fully populated.

Parameter Name	Data Type	Description
OrganizationId	String (collection)	The Identifier(s) of the Organization(s) to load.
IncludeSiteData	Boolean	If set to true the Sites collection on the Organization is populated.
IncludeServiceSiteData	Boolean	If set to true the ServiceSites collection on each Site is populated.
GetUnmoderatedDataWhereAvailable	Boolean	For Author/Approvers of the data return the Un-moderated data if any exists.
APIkey	String	The API key may be included by ServD implementations that wish to restrict access to the interface to those applications that have registered with a key. This is a similar mechanism to how Google Maps, Bing Maps, Facebook apps etc. control application access.

8.5.5.2 Method: RetrieveSiteDetails

Retrieve the Complete set of details about the Site(s) based on the Identifier(s) returned in a search.

Return Value: Site (collection)

A collection of Site records based on the requested identifiers, with the requested level of child objects fully populated.

Parameter Name	Data Type	Description
SiteId	String (collection)	The SiteId returned by the Search method.
IncludeServiceSiteData	Boolean	If set to true the ServiceSites collection on each Site is populated.
GetUnmoderatedDataWhereAvailable	Boolean	For Author/Approvers of the data, return the Un-moderated data if any exists.
APIkey	String	The API key may be included by ServD implementations that wish to restrict access to the interface to those applications that have registered with a key. This is a similar mechanism to how Google Maps, Bing Maps, Facebook apps etc. control application access.

8.5.5.3 Method: RetrieveServiceSiteDetails

Retrieve the Complete set of details about the Service Site(s) based on the Identifier(s) returned in a search.

Return Value: ServiceSite (collection)

A collection of Service Site records based on the requested identifiers, with the requested level of child objects fully populated.

Parameter Name	Data Type	Description
ServiceSiteId	String (collection)	The Identifier(s) of the Service Site(s) to load.
GetUnmoderatedDataWhereAvailable	Boolean	For Author/Approvers of the data, return the Un-moderated data if any exists.
APIkey	String	The API key may be included by ServD implementations that wish to restrict access to the interface to those applications that have registered with a key. This is a similar mechanism to how Google Maps, Bing Maps, Facebook apps etc. control application access.

8.5.5.4 Method: RetrieveProviderDetails

Retrieve the Complete set of details about the Provider(s) based on the Identifier(s) returned in a Search.

Return Value: Provider (collection)

A collection of Provider records based on the requested Identifiers, with the requested level of child objects fully populated.

Parameter Name	Data Type	Description
ProviderId	String (collection)	The Identifier(s) of the Provider(s) to load
GetUnmoderatedDataWhere Available	Boolean	For Author/Approvers of the data, return the Un-moderated data, if any exists
APIkey	String	The API key may be included by ServD implementations that wish to restrict access to the interface to those applications that have registered with a key. This is a similar mechanism to how Google Maps, Bing Maps, Facebook apps etc. control application access.

8.5.5.5 Method: RetrieveServiceSiteProviderDetails

Retrieve the Complete set of details about the Service Site Provider (s) based on the Identifier(s) returned in a search.

Return Value: ServiceSiteProvider (collection)

A collection of Service Site Provider records based on the requested identifiers, with the requested level of child objects fully populated.

Parameter Name	Data Type	Description
ServiceSiteProviderId	String (collection)	The Identifier(s) of the Service Site Providers(s) to load.
GetUnmoderatedDataWhereAvailable	Boolean	For Author/Approvers of the data, return the Un-moderated data if any exists.
APIkey	String	The API key may be included by ServD implementations that wish to restrict access to the interface to those applications that have registered with a key. This is a similar mechanism to how Google Maps, Bing Maps, Facebook apps etc. control application access.

8.5.5.6 Method: RetrievePublicKeys

Retrieve the Public Key(s) for the requested records based on the Identifier(s) returned by the Search.

At least one Identifier should be provided; otherwise the function will return an empty collection.

Return Value: PublicKey (collection)

The collection of Public keys that have been requested

Parameter Name	Data Type	Description
SiteId	String (collection)	The list of Identifiers of the Sites to retrieve the Public Key.
ServiceSiteId	String (collection)	The list of Identifiers of the Service Sites to retrieve the Public Key.
ProviderId	String (collection)	The list of Identifiers of the Providers to retrieve the Public Key.
APIkey	String	The API key may be included by ServD implementations that wish to restrict access to the interface to those applications that have registered with a key. This is a similar mechanism to how Google Maps, Bing Maps, Facebook apps etc. control application access.

8.5.6 Maintenance Interface

The **Maintenance** Interface actions requests to insert and update data requested by an authorized **Maintenance Application**.

Each of the Add/Update routines returns an `UpdateRecordResult` collection that includes the identifiers of every new/updated record that was committed to the data store. This includes indication that the resulting object requires Approval, if applicable.

If an Author is not an Approver and attempts to set the RecordStatus to Complete, the **Maintainable ServD Core** will reset the value to RequiresModeration.

A record is updated in its entirety; attempts to update a partial record will be rejected with an error result of type PartialRecordNotUpdatable. Refer to the UpdateRecordResult details for further information.

8.5.6.1 Record Ids and External Ids

Each object has a record Id that is managed by the **Maintainable ServD Core** and all operations on the Interfaces use this Id to identify the record.

When adding a new record its Id must be null and the External Id must have a value (not null). The return value from the update or add method includes the allocated record Id for the corresponding External Id, and the status of the operation (Success, SuccessRequiresModeration, etc.).

This approach is used to permit an implementation to use integers, GUIDs or other types for their internal identifiers, even though the data type of this field is a string (50).

8.5.6.2 Add

When adding records an entire object hierarchy can be added in one call. This permits a single call to add in an Organization, all its Sites and the Service Sites and all Contact Points and associated other records.

Where a parent object fails to be added, any related child records may be skipped and not attempted to be added to the ServD data store. In these cases the method may not include these skipped records in the result list returned by the method.

For example: If the Organization fails to be added, then the list of Organization Names included in the request may be excluded from the results list returned by the AddOrganization method.

8.5.6.3 Update

Updating a record requires its retrieval using the appropriate RetrieveDetails method, changing the appropriate values, and then calling the corresponding Update method.

For child collection records (such as Organization name) the collection does not have to be fully loaded, only the items in the list are updated. New records can be added to the child collections using the Update methods, ensuing that the External Id is populated and the new records Id is null.

To update a child object (e.g. Organization Name) without modifying the parent object (e.g. Organization), set the record status of the parent object to **DoNotUpdate**.

Example: When Updating an Organization's Preferred Name, set the RecordStatus on the Organization object to DoNotUpdate and the RecordStatus on the OrganizationName object to Complete. (If the user is an Author, and not an Approver, the update will change this value to CompleteAndRequiresModeration, and the Update result will be **SuccessRequiresModeration**).

8.5.6.4 Delete

Deleting records in the system is performed explicitly by calling the DeleteRecord method. Each record in the system is identified by its record type and record identifier, which must be provided to delete the record.

As the deletes are logical operations, and also subject to moderation, a deletion is actually the setting of the Record Status to either Deleted or DeletedAndRequiresModeration.

Deleting records from child collections (such as Organization Name) is the same. Removing an OrganizationName from a collection and then calling UpdateOrganization will not delete the record. This permits the call to update to only have the data which is actually changing in the call, and not the entire tree of records.

Deleting a parent record will cascade delete all child records. If a Provider is deleted, only the Service Site Provider records related to that Provider will be deleted, the Service Site record is not deleted, it is not considered a child of the Service Site Provider.

8.5.6.5 Method: OrganizationAdd

Add a new Organization. This will always create an un-approved Organization record. Organization records must be approved by a Content Manager before they will be available via the search interfaces.

Return Value: UpdateRecordResult (collection)

A collection of Identifiers and their Update Statuses

Parameter Name	Data Type	Description
details	Organization	The details of the Organization to add.
APIkey	String	The API key may be included by ServD implementations that wish to restrict access to the interface to those applications that have registered with a key. This is a similar mechanism to how Google Maps, Bing Maps, Facebook apps etc. control application access.

8.5.6.6 Method: OrganizationUpdate

Provide the details of an Organization that needs to be updated (including any child object information)

Return Value: UpdateRecordResult (collection)

A collection of Identifiers and their Update Statuses

Parameter Name	Data Type	Description
details	Organization	The details of the Organization to update.
APIkey	String	The API key may be included by ServD implementations that wish to restrict access to the interface to those applications that have registered with a key. This is a similar mechanism to how Google Maps, Bing Maps, Facebook apps etc. control application access.

8.5.6.7 Method: SiteAdd

Add a new Site to an Organization

Return Value: UpdateRecordResult (collection)

A collection of Identifiers and their Update Statuses

Parameter Name	Data Type	Description
~~**OrganizationId**~~	~~String~~	~~The ID of the Organization to add the new Site~~
details	Site	The details of the Site to be added. All object Identifiers (not the Identifiers Collection Objects) must be null, and External Identifiers populated with values (If you do not require these, just populate with a New GUID value). If they are not that item (and its children) will not be saved.
APIkey	String	The API key may be included by ServD implementations that wish to restrict access to the interface to those applications that have registered with a key. This is a similar mechanism to how Google Maps, Bing Maps, Facebook apps etc. control application access.

8.5.6.8 Method: SiteUpdate

Update the details of a Site

Return Value: UpdateRecordResult (collection)

A collection of Identifiers and their Update Statuses

Parameter Name	Data Type	Description
details	Site	The details of the Site to be updated. All object Identifiers must be the values returned from a Search as is the Object Identifier is the value used to retrieve the records to update. ExternalId values are ignored when locating records to update. Not all child objects need to be included, only those with changes required.
APIkey	String	The API key may be included by ServD implementations that wish to restrict access to the interface to those applications that have registered with a key. This is a similar mechanism to how Google Maps, Bing Maps, Facebook apps etc. control application access.

8.5.6.9 Method: ServiceSiteAdd

Add a Service Site to a Site (by Site ID)

Return Value: UpdateRecordResult (collection)

A collection of Identifiers and their Update Statuses

Parameter Name	Data Type	Description
~~**SiteId**~~	~~String~~	~~The Identifier of the Site to associate this Service Site.~~
details	ServiceSite	The details of the Service Site to be added. All object Identifiers (not the Identifiers Collection Objects) must be null, and External Identifiers populated with values (If you do not require these, just populate with a New GUID value). If they are not that item (and its children) will not be saved.
APIkey	String	The API key may be included by ServD implementations that wish to restrict access to the interface to those applications that have registered with a key. This is a similar mechanism to how Google Maps, Bing Maps, Facebook apps etc. control application access.

8.5.6.10   Method: ServiceSiteUpdate

Update the details of a Service Site

Return Value: UpdateRecordResult (collection)

A collection of Identifiers and their Update Statuses

Parameter Name	Data Type	Description
details	ServiceSite	The details of the Service Site to be updated. All object Identifiers must be the values returned from a Search as this is the value used to retrieve the records to update. The ExternalId values are ignored when locating records to update. Not all child objects need to be included, only those with changes required.
APIkey	String	The API key may be included by ServD implementations that wish to restrict access to the interface to those applications that have registered with a key. This is a similar mechanism to how Google Maps, Bing Maps, Facebook apps etc. control application access.

8.5.6.11   Method: ProviderAdd

Add a new Provider.

Note: In order to satisfy security requirements, a Provider record has a status of **CompleteAndRequiresModeration** until it is associated with a Service Site using the AddServiceSiteProvider operation.

A Content Manager is required to approve the addition of new Providers.

Return Value: UpdateRecordResult (collection)

A collection of Identifiers and their Update Statuses

Parameter Name	Data Type	Description
details	Provider	The details of the Provider to be added. All object Identifiers (not the Identifiers Collection Objects) must be null, and External Identifiers populated with values (If you do not require these, just populate with a New GUID value). If they are not that item (and its children) will not be saved.
APIkey	String	The API key may be included by ServD implementations that wish to restrict access to the interface to those applications that have registered with a key. This is a similar mechanism to how Google Maps, Bing Maps, Facebook apps etc. control application access.

8.5.6.12 Method: ProviderUpdate

Update a Providers Details

Return Value: UpdateRecordResult (collection)

A collection of Identifiers and their Update Statuses

Parameter Name	Data Type	Description
details	Provider	The details of the Provider to be updated. All object Identifiers must be the values returned from a Search as this is the value used to retrieve the records to update. The ExternalId values are ignored when locating records to update. Not all child objects need to be included, only those with changes required.
APIkey	String	The API key may be included by ServD implementations that wish to restrict access to the interface to those applications that have registered with a key. This is a similar mechanism to how Google Maps, Bing Maps, Facebook apps etc. control application access.

8.5.6.13 Method: AddServiceSiteProviderAdd

Associate an existing Service Site with an existing Provider.

Return Value: UpdateRecordResult (collection)

A collection of Identifiers and their Update Statuses

Parameter Name	Data Type	Description
details	ServiceSiteProvider	The details of the Service Site and Provider to associate
APIkey	String	The API key may be included by ServD implementations that wish to restrict access to the interface to those applications that have registered with a key. This is a similar mechanism to how Google Maps, Bing Maps, Facebook apps etc. control application access.

8.5.6.14 Method: ~~UpdateServiceSiteProvider~~Update

Update a specific Service Site Provider record.

Return Value: UpdateRecordResult (collection)

A collection of Identifiers and their Update Statuses.

Parameter Name	Data Type	Description
details	ServiceSiteProvider	The details of the Service Site Provider association to update
APIkey	String	The API key may be included by ServD implementations that wish to restrict access to the interface to those applications that have registered with a key. This is a similar mechanism to how Google Maps, Bing Maps, Facebook apps etc. control application access.

8.5.6.15 Method: UploadSiteCertificate

Upload a Public Key Certificate for a Site.

Return Value: UpdateRecordResult

A collection of Identifiers and their Update Statuses

Parameter Name	Data Type	Description
SiteId	String	The Site to update the Public Key
PublicKey	String	Public Key to be Updated on the Site Record
APIkey	String	The API key may be included by ServD implementations that wish to restrict access to the interface to those applications that have registered with a key. This is a similar mechanism to how Google Maps, Bing Maps, Facebook apps etc. control application access.

8.5.6.16　　　Method: UploadServiceSiteCertificate

Upload a Public Key Certificate for a Service Site.

Return Value: UpdateRecordResult

A collection of Identifiers and their Update Statuses

Parameter Name	Data Type	Description
ServiceSiteId	String	The Site to update the Public Key
PublicKey	String	Public Key to be Updated on the Service Site Record
APIkey	String	The API key may be included by ServD implementations that wish to restrict access to the interface to those applications that have registered with a key. This is a similar mechanism to how Google Maps, Bing Maps, Facebook apps etc. control application access.

8.5.6.17　　　Method: GetRecordsRequiringApproval

This method retrieves a list of added or updated records that are awaiting approval.

Note: Authors who are not Approvers can use this to see what changes have not been approved yet.

Return Value: RecordRequiringApproval (collection)

A collection of summary details describing the changes that require approval.

Parameter Name	Data Type	Description
Page	Integer	The page number of data to return. (0 for first page of data)
PageSize	Integer	The maximum number of items to retrieve as one page of data. This is used to restrict the amount of data that is returned in the results.
APIkey	String	The API key may be included by ServD implementations that wish to restrict access to the interface to those applications that have registered with a key. This is a similar mechanism to how Google Maps, Bing Maps, Facebook apps etc. control application access.

8.5.6.18 Method: ApproveRecords

Approve the data as specified in the list of Identifiers.
This method must be performed by Approvers or Content Managers.

New Organizations and Providers must be approved a Content Manager.

Return Value: UpdateRecordResult (collection)

A collection of the status of all of the record approvals.

Parameter Name	Data Type	Description
Identifiers	RecordIndex (collection)	The list of record types and identifiers to be approved
APIkey	String	The API key may be included by ServD implementations that wish to restrict access to the interface to those applications that have registered with a key. This is a similar mechanism to how Google Maps, Bing Maps, Facebook apps etc. control application access.

8.5.6.19 Method: RejectRecords

Reject the data as specified in the list of Identifiers.

This method must only be performed by Approvers.

Note: If the specified record is not currently marked for approval, then an **ErrorOther** result status will be returned.

Return Value: UpdateRecordResult (collection)

A collection of the status of all of the record rejections.

Parameter Name	Data Type	Description
Identifiers	RecordIndex (collection)	The list of record types and identifiers to be rejected.
Comment	String	A comment describing the reason that the record update has been rejected.
APIkey	String	The API key may be included by ServD implementations that wish to restrict access to the interface to those applications that have registered with a key. This is a similar mechanism to how Google Maps, Bing Maps, Facebook apps etc. control application access.

8.5.6.20   Method: DeleteRecords

Delete one or more records of a specified record type.

This method must only be performed by Approvers or Content Managers.

NOTE: WHERE THERE ARE ASSOCIATED CHILD LINKS, THESE WILL ALSO BE DELETED. DELETIONS ARE NOT AUTOMATICALLY APPROVED. A SEPARATE CALL MUST BE MADE TO APPROVE THESE DELETIONS.
IN MOST IMPLEMENTATIONS THE DELETION IS LIKELY TO BE A LOGICAL DELETION, AND THEREFORE THE DATA WILL NOT BE RETURNED VIA A SEARCH.
TO "UNDELETE" A DELETED RECORD AN AUTHOR MUST CALL THE RELEVANT UPDATE METHOD WITH AN UPDATED RECORDSTATUS PROPERTY.

Example: When deleting an Organization Site, all Service Sites, and Site Geographical Coordinate records will also be deleted.

Return Value: DeleteRecordResult (collection)

A collection of the status of all of the record deletions

Parameter Name	Data Type	Description
Identifiers	RecordIndex (collection)	The list of records to be deleted, identified by their record type and identifier
APIkey	String	The API key may be included by ServD implementations that wish to restrict access to the interface to those applications that have registered with a key. This is a similar mechanism to how Google Maps, Bing Maps, Facebook apps etc. control application access.

8.5.7 VerifyDetails Interface

The Verify Details Interface is exposed by a **ServD Verifier** (which is external to the ServD Federation) to permit a **Maintainable ServD Core** to verify updates made by a **Maintenance Application** that a record has a specific Attribute which is configured to be externally validated.

This is the mechanism for certification/registration Organizations to be able to electronically validate attributes of Organizations, Sites and Provider. For example to validate that a Provider has particular qualifications.

Informative Note: External applications may be hosted by health practitioner registration boards, insurance companies, registration authorities etc.

8.5.7.1 Method: VerifyOrganization

Verify an Organization Attribute.

Return Value: AttributeVerificationResult

The results of the verification of the Attribute, including signed data for the purpose of non-repudiation.

Parameter Name	Data Type	Description
AttributeType	String *(mandatory)*	The type of Attribute associated with the Organization
AttributeValue	String *(mandatory)*	The Attribute value
OrganizationIdentifier	String *(mandatory)*	This is a specific Identifier that is used to identify the Organization to the external application. It is not the OrganizationId from the ServD Core.
Name	String	The name of the Organization (Optional)
addresses	Address (collection)	The current addresses of the Organization (Optional)
contacts	ContactPoint (collection)	The current contacts of the Organization (Optional)
APIkey	String	The API key may be included by ServD implementations that wish to restrict access to the interface to those applications that have registered with a key. This is a similar mechanism to how Google Maps, Bing Maps, Facebook apps etc. control application access.

8.5.7.2 Method: VerifyProvider

Verify a Provider Attribute.

Return Value: AttributeVerificationResult

The results of the verification of the Attribute, including signed data for the purpose of non-repudiation.

Parameter Name	Data Type	Description
AttributeType	String *(mandatory)*	The type of Attribute associated with the Provider
AttributeValue	String *(mandatory)*	The Attribute value
ProviderIdentifier	String *(mandatory)*	This is a specific Identifier that is used to identify the Provider to the external application. It is not the ProviderId from the ServD Core.
FamilyName	String	The Family Name of the Provider (Optional)
GivenNames	String	The Given Name(s) of the Provider (Optional)
addresses	Address (collection)	The current addresses of the Provider (Optional)
contacts	ContactPoint (collection)	The current contacts of the Provider (Optional)
APIkey	String	The API key may be included by ServD implementations that wish to restrict access to the interface to those applications that have registered with a key. This is a similar mechanism to how Google Maps, Bing Maps, Facebook apps etc. control application access.

8.5.7.3 Method: VerifySite

Verify a Site Attribute.

Return Value: AttributeVerificationResult

The results of the verification of the Attribute, including signed data for the purpose of non-repudiation.

Parameter Name	Data Type	Description
AttributeType	String *(mandatory)*	The type of Attribute associated with the Site.
AttributeValue	String *(mandatory)*	The Attribute value

Parameter Name	Data Type	Description
SiteIdentifier	String *(mandatory)*	This is a specific Identifier that is used to identify the Site to the external application. It is not the SiteId from the ServD Core.
Name	String	The name of the Site claimed to have the attribute specified (Optional).
addresses	Address (collection)	The current addresses of the Site (Optional).
contacts	ContactPoint (collection)	The current contacts of the Site (Optional).
APIkey	String	The API key may be included by ServD implementations that wish to restrict access to the interface to those applications that have registered with a key. This is a similar mechanism to how Google Maps, Bing Maps, Facebook apps etc. control application access.

8.6   Reference Lists (CTS2)

The following reference lists are used by the ServD specification. These are to be exposed as value CTS2 value sets as shown in the interaction diagram in section 8.2.7.

The ServD Locator provides the network address of the CTS2 value set that exposes each of these types.

Code	Used By
ServD.ADRT	DataModel.Address.AddressType
ServD.ADRT	Parameters.AddressSearchParameters.AddressType
ServD.ADRT	Parameters.OrganizationSearchParameters.AddressType
ServD.AP	DataModel.Address.AddressPurpose
ServD.AP	Parameters.AddressSearchParameters.AddressPurpose
ServD.AP	Parameters.OrganizationSearchParameters.AddressPupose
ServD.ATTT.Org	DataModel.Organization.Attributes (collection)
ServD.ATTT.Prov	DataModel.Provider.Attributes (collection)
ServD.ATTT.Prov.Cred	DataModel.Provider.Credentials (collection)
ServD.ATTT.Prov.Qual	DataModel.Provider.Qualifications (collection)
ServD.ATTT.Prov.Reg	DataModel.Provider.LicenseOrRegistration (collection)
ServD.ATTT.Site	DataModel.Site.Attributes (collection)
ServD.ATTT.SS	DataModel.ServiceSite.Characteristics (collection)
ServD.CA	DataModel.CoverageArea.Code
ServD.CAC	DataModel.CatchmentArea.Code
ServD.CAC	Parameters.AddressSearchParameters.CatchmentAreaCode
ServD.CPEQ	DataModel.ContactPoint.EquipmentCode
ServD.CPP	DataModel.ContactPoint.ContactPointPurpose
ServD.CPT	DataModel.ContactPoint.ContactPointType
ServD.CPT	Parameters.OrganizationSearchParameters.WebAddressContactPointPupose
ServD.ETH	DataModel.Provider.CulturalEthnicityCode
ServD.ETH	Parameters.ProviderSearchParameters.CulturalEthnicityCode
ServD.ETH	Results.ProviderSearchItem.CulturalEthnicityCode

Code	Used By
ServD.GBSHF	DataModel.SiteGeographicalCoordinate.ShapeFormat
ServD.GEN	DataModel.Provider.GenderCode
ServD.GEN	Parameters.ProviderSearchParameters.GenderCode
ServD.GEN	Results.ProviderSearchItem.GenderCode
ServD.LAN	DataModel.ProviderLanguage.LanguageSpokenCode
ServD.LAN	Parameters.ProviderSearchParameters.LanguageSpokenCode
ServD.OAFT	DataModel.OrganizationAffiliation.AffiliationType
ServD.OAFT	Parameters.OrganizationSearchParameters.AffiliationType
ServD.ONP	DataModel.OrganizationName.NamePurpose
ServD.ONP	Parameters.OrganizationSearchParameters.NamePupose
ServD.ONT	DataModel.OrganizationName.NameType
ServD.ONT	Parameters.OrganizationSearchParameters.NameType
ServD.ORGST	DataModel.Organization.Status
ServD.PNP	DataModel.ProviderName.NamePurpose
ServD.PNT	DataModel.ProviderName.NameType
ServD.PROVST	DataModel.Provider.Status
ServD.RMC	DataModel.ReferralMethod.Code
ServD.SC	DataModel.ServiceSite.ServiceCategory
ServD.SC	Parameters.ServiceSiteSearchParameters.ServiceCategory
ServD.SPCLTY.Org	DataModel.Organization.Specialties (collection)
ServD.SPCLTY.Org	Parameters.OrganizationSearchParameters.SpecialtyCode
ServD.SPCLTY.SS	DataModel.ServiceSite.Specialties (collection)
ServD.SPCLTY.SS	Parameters.ServiceSiteSearchParameters.SpecialtyCode
ServD.SSFPC	DataModel.ServiceSite.FreeProvisionCode
ServD.SSFPC	Parameters.ServiceSiteSearchParameters.FreeProvisionCode
ServD.SSS	DataModel.ServiceSiteSetting.Code
ServD.SSTG	DataModel.ServiceSiteTargetGroup.Code

Code	Used By
ServD.SSTG	Parameters.ServiceSiteSearchParameters.TargetGroup
ServD.ST	DataModel.ServiceType.Value
ServD.ST	Parameters.ServiceSiteSearchParameters.ServiceType
ServD.TITLE	DataModel.ProviderName.TitleCode
ServD.TITLE	Results.ProviderSearchItem.TitleCode

8.7 Common

8.7.1 Overview

class Common

«enumeration»
RecordUpdateResultStatus Enum

Success = 0
SuccessRequiresModeration = 1
ErrorPermission = -1
PartialRecordNotUpdatable = -2
ErrorDuplicateRecord = -3
ErrorValidation = -4
ErrorOther = -5

«enumeration»
SearchResults Status Enum

Success = 0
Error = 1
Partial = 2
ErrorInadequateSearchParameters = 3

«enumeration»
AttributeVerificationStatus Enum

NotApplicable = 0
NotVerifiable = 1
Pending = 2
Verified = 3
Expired = 4
NotValid = 5

+Status 1

DaysOfWeek

+ Friday :bool
+ Monday :bool
+ Saturday :bool
+ Sunday :bool
+ Thursday :bool
+ Tuesday :bool
+ Wednesday :bool

PagedSearchResults

+ CurrentPage :int
+ EstimatedTotalItems :int
+ EstimatedTotalPages :int
+ PageSize :Integer

«enumeration»
RecordTypeEnum

Organization = 1
OrganizationAffiliation = 2
OrganizationName = 3
Provider = 10
ProviderLanguage = 11
ProviderName = 12
ServiceSite = 20
ServiceSiteProvider = 21
ServiceSiteSetting = 22
ServiceSiteTargetGroup = 23
Site = 30
SiteGeographicalCoordinate = 31
Address = 40
Attribute = 41
Available = 42
ContactPoint = 43
NotAvailable = 44
Keyword = 45
Identifier = 46
CatchmentArea = 47
Specialty = 48
ReferralMethod = 49
CoverageArea = 50

+Errors 0..*

RecordResult

+ ExternalId :string
+ Identifier :String
+ RetrieveDetailsURI :String

Result

+ AdvancedErrorDescription :string
+ ErrorCode :string
+ ErrorDescription :string
+ RetrieveDetailsURI :String

+RecordType 1

«enumeration»
Conditional Indicator Enum

NotDefined = 0
Yes = 'Y'
No = 'N'
Unknown = 'U'
Conditional = 'C'

SearchItem

+ Heading :String
+ ImageURI :String
+ LastModifiedDate :DateTime
+ RecordStatus :RecordStatusEnum
+ RetrieveDetailsURI :String
+ SubHeading :String
+ Summary :String

«enumeration»
StringMatchTypeEnum

Exact = 0
Contains = 1
StartsWith = 2
Fuzzy = 3

ModeratedRecord

+ EndDate :DateTime [0..1]
+ ExternalId :string
+ LastModificationDate :DateTime [0..1]
+ RecordPrivacyPolicy :String
+ StartDate :DateTime [0..1]

+RecordStatus 1

«enumeration»
RecordStatus Enum

Complete = 0
Partial = 1
CompleteAndRequiresModeration = 2
CompleteAndModerationRejected = 3
Deleted = -1
DeletedAndRequiresModeration = -2
DeletedAndModerationRejected = -3
DoNotUpdate = -4

class Common

«enumeration»
RecordUpdateResultStatus Enum

Success = 0
SuccessRequiresModeration = 1
ErrorPermission = -1
PartialRecordNotUpdatable = -2
ErrorDuplicateRecord = -3
ErrorValidation = -4
ErrorOther = -5

«enumeration»
SearchResults Status Enum

Success = 0
Error = 1
Partial = 2
ErrorInadequateSearchParameters = 3

«enumeration»
AttributeVerificationStatus Enum

NotApplicable = 0
NotVerifiable = 1
Pending = 2
Verified = 3
Expired = 4
NotValid = 5

DaysOfWeek

+ Friday :bool
+ Monday :bool
+ Saturday :bool
+ Sunday :bool
+ Thursday :bool
+ Tuesday :bool
+ Wednesday :bool

PagedSearchResults

+ CurrentPage :int
+ Errors :Result ([])
+ EstimatedTotalItems :int
+ EstimatedTotalPages :int
+ Status :SearchResultsStatusEnum

+Status

«enumeration»
RecordTypeEnum

Organization = 1
OrganizationAffiliation = 2
OrganizationName = 3
Provider = 10
ProviderLanguage = 11
ProviderName = 12
ServiceSite = 20
ServiceSiteProvider = 21
ServiceSiteSetting = 22
ServiceSiteTargetGroup = 23
Site = 30
SiteGeographicalCoordinate = 31
Address = 40
Attribute = 41
Available = 42
ContactPoint = 43
NotAvailable = 44
Keyword = 45
Identifier = 46
CatchmentArea = 47
Specialty = 48
ReferralMethod = 49
CoverageArea = 50

RecordResult

+ ExternalId :string
+ Identifier :String

Result

+ AdvancedErrorDescription :string
+ ErrorCode :string
+ ErrorDescription :string
+ RecordType :RecordTypeEnum

+Errors

+RecordType

«enumeration»
ConditionalIndicatorEnum

NotDefined = 0
Yes = 'Y'
No = 'N'
Unknown = 'U'
Conditional = 'C'

SearchItem

+ Heading :String
+ ImageURI :String
+ RetrieveDetailsURI :String
+ SubHeading :String
+ Summary :String

«enumeration»
StringMatchTypeEnum

Exact = 0
Contains = 1
StartsWith = 2
Fuzzy = 3

ModeratedRecord

+ EndDate :DateTime?
+ ExternalId :string
+ LastModificationDate :DateTime?
+ RecordPrivacyPolicy :String
+ RecordStatus :RecordStatusEnum
+ StartDate :DateTime?

+RecordStatus

«enumeration»
RecordStatus Enum

Complete = 0
Partial = 1
CompleteAndRequiresModeration = 2
CompleteAndModerationRejected = 3
Deleted = -1
DeletedAndRequiresModeration = -2
DeletedAndModerationRejected = -3
DoNotUpdate = -4

8.7.2 Date-Time Handling – Time-zones

All Date-Time fields in the information model must include time-zone information so that they can be understood by all systems correctly.

The ServD Services should utilize UTC (Co-ordinated Universal Time) to reduce potential issues handling information across time-zones.

8.7.3 AttributeVerificationStatusEnum

The AttributeVerificationStatus provides a Status for the attribute.

String	Description
NotApplicable	There is no **ServD Verifier** configured by the **ServD Core** for the specified Attribute Type.
NotVerifiable	The **ServD Verifier** allocated to this type by the **ServD Core** does not support the specified Attribute Type.
Pending	This attribute has not yet been verified. (not returned by the **ServD Verifier**, but used in Search results for rows that have not been verified yet)
Verified	This attribute has been verified for the specified Organization, Site or Provider.
Expired	This attribute was Verified but has expired (refer to the expiry date for information as to when)
NotValid	This attribute has been checked with the **ServD Verifier** and the specified Organization, Site or Provider has not passed verification.

Referenced By:

- DataModel.Attribute.VerificationStatus
- Results.AttributeVerificationResult.VerificationStatus

8.7.4 ConditionalIndicatorEnum

The Conditional Indicator is used by the Service Site to indicate the Service Sites Eligibility, or if an Appointment is required.

Note: When storing this into a database, it may be stored as a string/char data type so that the textual representation is easier to identify.

String	Description
NotDefined	The condition state is not defined
Yes	~~'Y'~~ The condition is required
No	~~'N'~~ The condition is not required
Unknown	~~'U'~~ The condition state is unknown
Conditional	~~'C'~~ The condition depends on other factors. Refer to other associated fields

Referenced By:

- DataModel.ServiceSite.Eligibility

- DataModel.ServiceSite.AppointmentRequired
- Parameters.ServiceSiteSearchParameters.NoAppointmentReqiured

8.7.5 DaysOfWeek

The Day Of Week Enumeration is used in conjunction with the available time information to mark which days of the week are available.

Property	Data Type	Description
Monday	Boolean	Monday
Tuesday	Boolean	Tuesday
Wednesday	Boolean	Wednesday
Thursday	Boolean	Thursday
Friday	Boolean	Friday
Saturday	Boolean	Saturday
Sunday	Boolean	Sunday

Referenced By:

- DataModel.Available.DaysOfWeek
- Parameters.Availability.DaysOfWeek

8.7.6 ModeratedRecord

The Moderated Record is the abstract base class that includes the properties common to all entities that require moderation and part of the persistent ServD Object Model.

Property	Data Type	Description
ExternalId	String (50) *(mandatory)*	This External Identifier is provided by applications using the Maintenance Interface to identify their own records when adding new records. It may be stored by the **Maintainable ServD Core** but it must be returned in the Add operations on the Maintenance interface.
RecordStatus	RecordStatusEnum	The moderation Status of this record (Complete, Partial, RequiresModeration, Deleted)
RecordPrivacyPolicy	String (250)	The sensitivity of this data and accessibility based on policy. This can be a multi-part policy that could include roles and levels of privacy. **Example:** In the Victorian (Australia) implementation o there is a policy of a single component with 3 possible values. Public, Secure or OwnerOnly *Note: The privacy Policy will be based on an agreed structure that is to be used by the integrated ServD Applications.*
LastModificationDate	DateTime (nullable)	The date and time this record was last modified. The server will set this to the local server time at the point of an add or update, any values entered by a user will be ignored. Note that the approval of a record is not considered to be a modification and should not update this field.
StartDate	DateTime (nullable)	The effective start date for this record
EndDate	DateTime (nullable)	The effective end date of this record

8.7.7 PagedSearchResults

The Paged Search Results provides a common paging foundation under all search list results, to permit the loading of longer lists of data.

Property	Data Type	Description
Status	SearchResultsStatusEnum	The result status of the Search **Example:** Success, Error, Partial
Errors	Result (collection)	Collection of errors
PageSize	Integer	The Number of items included in a page of results
CurrentPage	Integer	The Current page of data that is in this results collection
EstimatedTotalPages	Integer	The Estimated Total number of Pages of data available in the collection
EstimatedTotalItems	Integer	The Estimated Total number of items in the collection

8.7.8 RecordResult

Derived from: **Common.Result**

This base class is used to provide a common set of properties for results for record modifications. Specific Error status codes can be defined in derived classes.

Property	Data Type	Description
RetrieveDetailsURI	String	The network location of the Retrieve Details Interface to request for the detailed information.
Identifier	String (50)	The Primary Identifier of the Record that was modified
ExternalId	String (50)	The External Identifier of the record that modified (if applicable).

8.7.9 RecordStatusEnum

The Record Status Enumeration indicates the state/completeness of the associated record.
When an Author updates a record the **Maintainable ServD Core** will automatically set this value (provided the Author is not also an Approver). When an Approver updates a record he/she is able to set this value to Complete or CompleteAndRequiresModeration. If a record has the Partial record status, then the record must not be updated.

String	Description
Complete	The record is complete and is the current approved content.
Partial	Only some data was able to be loaded (security or privacy).
CompleteAndRequiresModeration	This record is complete, however the data requires moderation and is not available to users apart from the Author/Approver of this specific record.
CompleteAndModerationRejected	This record is complete, however the data was rejected by the moderator. *NOTE: RECORDS WITH THIS STATUS ARE ONLY RETURNED TO THE AUTHOR/APPROVER OF THIS SPECIFIC RECORD*
Deleted	The Record Has been deleted. *NOTE: RECORDS WITH THIS ATTRIBUTE WILL ONLY BE RETURNED TO OWNERS OF THE DATA (AUTHOR/APPROVER) WHERE THE IMPLEMENTATION PERFORMS A LOGICAL DELETION. DATA WITH THIS PROPERTY VALUE MUST NOT BE RETURNED VIA THE SEARCH INTERFACES TO UN-AUTHENTICATED USERS.*
DeletedAndRequiresModeration	The Record Has been deleted, however the delete requires moderation. *NOTE: RECORDS WITH THIS ATTRIBUTE WILL ONLY BE RETURNED TO OWNERS OF THE DATA (AUTHOR/APPROVER) WHERE THE IMPLEMENTATION PERFORMS A LOGICAL DELETION. DATA WITH THIS PROPERTY VALUE MUST NOT BE RETURNED VIA THE SEARCH INTERFACES TO UN-AUTHENTICATED USERS.*
DeletedAndModerationRejected	The Record Has been deleted, however the delete requires moderation. *NOTE: RECORDS WITH THIS ATTRIBUTE WILL ONLY BE RETURNED TO OWNERS OF THE DATA (AUTHOR/APPROVER) WHERE THE IMPLEMENTATION PERFORMS A LOGICAL DELETION. DATA WITH THIS PROPERTY VALUE MUST NOT BE RETURNED VIA THE SEARCH INTERFACES TO UN-AUTHENTICATED USERS.*
DoNotUpdate	This is a special status that is used to indicate to the Maintenance's Update Operations that it should not make any changes to this object. It is used so that during an update changes can be made to child objects without updating parent records. This will not be returned by any operations, only used as input values.

Referenced By:

- Common.ModeratedRecord.RecordStatus

8.7.10 RecordTypeEnum

The Record Type Enumeration is used by the Maintenance Interface for handling batches of information for approval, deletions and update.

This enumeration does not start at 0 to assist in enforcing that the normal default is not used and forces its explicit value to be set.

String	Description
Organization	Organization Record Type
OrganizationAffiliation	Organization Affiliation Record Type
OrganizationName	Organization Name Record Type
Provider	Provider Record Type
ProviderLanguage	Provider Language Record Type
ProviderName	Provider Name Record Type
ServiceSite	Service Site Record Type
ServiceSiteProvider	Service Site Provider Record Type
ServiceSiteSetting	Service Site Setting Record Type
ServiceSiteTargetGroup	Service Site Target Group Record Type
ServiceType	Service Type Record Type
Site	Site Record Type
SiteGeographical Coordinate	Site Geographical Coordinate Record Type
Address	Address Record Type
Attribute	Attribute Record Type
Available	Available Record Type
ContactPoint	Contact Point Record Type
NotAvailable	Not Available Record Type
Keyword	Keyword Record Type
Identifier	Identifier Record Type
CatchmentArea	Catchment Area Record Type
Specialty	Specialty Record Type

String	Description
ReferralMethod	Referral Method Record Type
CoverageArea	Coverage Area Record Type

Referenced By:

- Parameters.RecordIndex.RecordType
- Results.PublicKey.RecordType
- Common.Result.RecordType
- Results.RecordRequiringApproval.RecordType

8.7.11 RecordUpdateResultStatusEnum

When data is modified by the Operations on the Maintenance Interface this enumeration is used to report the status of the update.

String	Description
Success	The Update was successful and does not require moderation.
SuccessRequiresModeration	The Update was successful, but will require moderation before it will be shown in searches from other Organizations.
	NOTE: IF A RECORD WAS REJECTED FROM MODERATION, THEN THIS STATUS WILL BE RETURNED.
ErrorPermission	The update failed as the user does not have permission to update this record.
PartialRecordNotUpdatable	The record that was submitted for update was not in an updatable state (RecordStatus = Partial).
ErrorDuplicateRecord	The Record could not be updated or added as it would cause a duplicate record.
ErrorValidation	A Validation error has occurred saving this record.
ErrorOther	Other type of error has occurred, refer to the error code and description.

Referenced By:

- Results.UpdateRecordResult.Result
- Results.DeleteRecordResult.Result

8.7.12 Result

This base class is used to provide a common set of properties for results. Specific Error status codes can be defined in derived classes.

Property	Data Type	Description

Property	Data Type	Description
ErrorCode	String	A System provided error code. (Optional)
ErrorDescription	String	A textual representation of the error that has occurred, intended to be displayed to a user as to what has occurred. *NOTE: IF RELEVANT THE NAME AND VALUE OF ANY FIELDS IN ERROR SHOULD BE INCLUDED.*
AdvancedErrorDescription	String	This Advanced Error Message description is intended for logging and diagnosis of issues, and not expected to be displayed to users. (Optional)
RecordType	RecordTypeEnum	Which record type was modified.

Referenced By:

- Common.PagedSearchResults.Errors (collection)

8.7.13 SearchItem

The abstract Search Item class provides the common properties that are required for all Search Items.

The content of these values can be determined by the ServD Core component, or its configuration. These values can be used in the same way that search results from internet search engines are presented; in a simple list with headings and thumbnail images.

Property	Data Type	Description
RetrieveDetailsURI	String	The network location of the Retrieve Details Interface to request for the detailed information.
Heading	String	A generalized heading that can present any information for this search item. This is unformatted text.
SubHeading	String	A generalized sub-heading that can present any information for this search item. This provides flexibility for a **ServD Core** implementation to adapt what information is included. This is unformatted text.
Summary	String	A generalized summary description of the search Item. This provides flexibility for a **ServD Core** implementation to adapt what information is included. This is unformatted text.
ImageURI	String	If there is an image associated with this Search Result Item, its URI can be included here.

Property	Data Type	Description
LastModificationDate	DateTime	The date and time this record (or one of its child records) was last modified.
RecordStatus	RecordStatusEnum	The moderation Status of this record (Complete, Partial, RequiresModeration, Deleted) Refer to section 8.5.4.1 for further details of the expected values of this field based on values in the search parameters.

8.7.14 SearchResultsStatusEnum

The Search Results Status is used to describe the status Search Operations of the ServD Search Interface. This is either from a **ServD Core** or a **ServD Locator** that supports the Search Interface, federating the actual search calls.

String	Description
Success	The Search was successful.
Error	An error has occurred during the search. Refer to the Errors collection on the associated search result for details of the errors.
Partial	Only some data was able to be loaded. **Note**: When searching using a federated search by a **ServD Locator** this can be the status associated if some of the searches succeeded, but others failed. In these cases refer to the Errors collection on the associated search result for details of the errors.
ErrorInadequateSearchParameters	The search has returned no data, as the search filter was not sufficiently restricted.

Referenced By:

- Common.PagedSearchResults.Status

8.7.15 StringMatchTypeEnum

String	Description
Exact	Any string match should be an exact match.
Contains	And string match should be considered a wildcard match that contains the entered text.
StartsWith	The provided text should be checked to see if it is the start of any entries. "And" would match to "Anderson", "And" but not "Grand"

String	Description
Fuzzy	Fuzzy Search. The mechanism used to resolve a Fuzzy search is outside the scope of this specification.

Referenced By:

- Parameters.SearchControlParameters.StringMatchType

8.8 Parameters

8.8.1 Overview

class Parameters

Availability
+ AvailableAt [0..1]
+ CheckDate
+ CheckTime
+ IncludeUnknown

SearchInputParameters
+ CoverageAreaCode
+ ModifiedAfterDate [0..1]
+ SearchFor
+ SearchOrganizations
+ SearchProviders
+ SearchServiceSites
+ SearchSites
+ StructuredQuery

+Availability 0..*

+OrganizationSearchParameters 0..*

OrganizationSearchParameters
+ AddressPupose
+ AddressType
+ AffiliationType
+ IdentifierType
+ IdentifierValue
+ Name
+ NamePupose
+ NameType
+ OrganizationId
+ SpecialtyCode
+ WebAddressContactPointPupose

+ProximitySearchParameters 0..1

ProximitySearchParameters
+ ProximityOfLatitude
+ ProximityOfLongitude
+ ProximityRadiusToSearch
+ SpatialReferenceID

+ProviderSearchParameter 0..*

ProviderSearchParameters
+ CulturalEthnicityCode
+ GenderCode
+ GivenNames
+ LanguageSpokenCode
+ ProviderId
+ SpecialInterests
+ Surname

+SiteSearchParameters 0..*

SiteSearchParameters
+ Name
+ SiteId

+AddressSearchParameters 0..1

Address SearchParameters
+ AddressPurpose
+ AddressType
+ CatchmentAreaCode
+ Country
+ IncludeSurrounding
+ Jurisdiction
+ PostalIdentificationCode
+ Region

+Attributes 0..* +Attributes 0..* +Attributes 0..*

HasAttribute
+ AttributeType
+ IncludeExpired
+ IncludeNotValid
+ IncludePending
+ Value

+ServiceSiteSearchParameters 0..*

ServiceSiteSearchParameters
+ FreeProvisionCode
+ IdentifierType
+ IdentifierValue
+ ServiceCategory
+ ServiceSiteId
+ ServiceType
+ SpecialtyCode
+ TargetGroup

+Attributes 0..*

«enumeration»
Common::RecordTypeEnum

Organization = 1
OrganizationAffiliation = 2
OrganizationName = 3
Provider = 10
ProviderLanguage = 11
ProviderName = 12
ServiceSite = 20
ServiceSiteProvider = 21
ServiceSiteSetting = 22
ServiceSiteTargetGroup = 23
Site = 30
SiteGeographicalCoordinate = 31
Address = 40
Attribute = 41
Available = 42
ContactPoint = 43
NotAvailable = 44
Keyword = 45
Identifier = 46
CatchmentArea = 47
Specialty = 48
ReferralMethod = 49
CoverageArea = 50

SearchControlParameters
+ GetUnmoderatedDataWhereAvailable
+ PageSize

+StringMatchType 1

«enumeration»
Common::StringMatchTypeEnum

Exact = 0
Contains = 1
StartsWith = 2
Fuzzy = 3

+RecordType 1

RecordIndex
+ Identifier

class Parameters

Availability
+ AvailabileAt
+ CheckDate
+ CheckTime
+ CurrentLocalSystemTime
+ DaysOfWeek
+ IncludeUnknown

SearchInputParameters
+ AddressSearchParameters
+ Availability ([])
+ OrganizationSearchParameters ([])
+ ProviderSearchParameters ([])
+ ProximitySearchParameters
+ SearchFor
+ SearchOrganizations
+ SearchProviders
+ SearchServiceSites
+ SearchSites
+ ServiceSiteSearchParameters ([])
+ SiteSearchParameters ([])
+ StructuredQuery

+Availability

OrganizationSearchParameters
+ AddressPupose
+ AddressType
+ AffiliationType
+ Attributes ([])
+ IdentifierType
+ IdentifierValue
+ Name
+ NamePupose
+ NameType
+ OrganizationId
+ SpecialtyCode
+ WebAddressContactPointPupose

+OrganizationSearchParameters

+ProximitySearchParameters

ProximitySearchParameters
+ ProximityOfLatitude
+ ProximityOfLongitude
+ ProximityRadiusToSearch
+ SpatialReferenceID

+ProviderSearchParameters

ProviderSearchParameters
+ Attributes ([])
+ CulturalEthnicityCode
+ GenderCode
+ GivenNames
+ LanguageSpokenCode
+ SpecialInterests
+ Surname

+SiteSearchParameters

SiteSearchParameters
+ Attributes ([])
+ Name
+ SiteId

+AddressSearchParameters

AddressSearchParameters
+ AddressPurpose
+ AddressType
+ CatchmentAreaCode
+ Country
+ IncludeSurrounding
+ Jurisdiction
+ PostalIdentificationCode
+ Region

+Attributes   +Attributes   +Attributes

HasAttribute
+ AttributeType
+ IncludeExpired
+ IncludeNotValid
+ IncludePending
+ Value

+ServiceSiteSearchParameters

+Attributes

ServiceSiteSearchParameters
+ Attributes ([])
+ FreeProvisionCode
+ IdentifierType
+ IdentifierValue
+ NoAppointmentReqiured
+ ServiceCategory
+ ServiceType
+ SpecialtyCode
+ TargetGroup

**«enumeration»
Common::RecordTypeEnum**
Organization = 1
OrganizationAffiliation = 2
OrganizationName = 3
Provider = 10
ProviderLanguage = 11
ProviderName = 12
ServiceSite = 20
ServiceSiteProvider = 21
ServiceSiteSetting = 22
ServiceSiteTargetGroup = 23
Site = 30
SiteGeographicalCoordinate = 31
Address = 40
Attribute = 41
Available = 42
ContactPoint = 43
NotAvailable = 44
Keyword = 45
Identifier = 46
CatchmentArea = 47
Specialty = 48
ReferralMethod = 49
CoverageArea = 50

SearchControlParameters
+ GetUnmoderatedDataWhereAvailable
+ PageSize
+ StringMatchType

+StringMatchType

**«enumeration»
Common::
StringMatchTypeEnum**
Exact = 0
Contains = 1
StartsWith = 2
Fuzzy = 3

+RecordType

RecordIndex
+ Identifier
+ RecordType

8.8.2 AddressSearchParameters

Filter options for the searching for records that have the specified address/catchment area values.

Property	Data Type	Description

Property	Data Type	Description
AddressType	String (64)	Defines the primary type of activity at a location (Optional) **Example:** Residential, Business, Temporary **cts2:**ServD.ADRT
AddressPurpose	String (64)	Defines the usage of this address in relation to a specific action (Optional) **Example:** Mailing, Delivery, Shipping, Search Results **cts2:**ServD.AP
Jurisdiction	String (256)	(City, Town, Village, Area) points within a hierarchy that travels from Country down to the lowest level of jurisdiction within the specified country. **Example:** West Melbourne
IncludeSurrounding	Boolean	Include the surrounding Suburbs/Jurisdictions
Region	String (256)	The region or state of this address. **Example:** Victoria
PostalIdentificationCode	String (50)	The postal Identification Code contains the Zip Code or Postcode. Usually this is defined by the national postal service. **Example:** 3003
Country	String (256)	The country **Example:** Australia
CatchmentAreaCode	String (64)	The Catchment Area Code *Note: This often used as a Local Government Area* **cts2:**ServD.CAC

Referenced By:

- Parameters.SearchInputParameters.AddressSearchParameters

8.8.3 Availability

The Availability parameter is used to specify the date/time required for the service to be available. Alternately it may be used to specify the day(s) of the week with optional time.

Note: Availability is used to designate when a ServiceSite or Site is open and/or available rather than to manage individual appointments.

Property	Data Type	Description

Property	Data Type	Description
~~CurrentLocalSystemTime~~	~~DateTime~~ *~~(mandatory)~~*	~~The Current Local System Time is used to provide the requesting systems time-zone information and also a relative point in time to verify that the systems current time is accurate within that zone.~~
AvailableAt	DateTime (nullable)	The date and time that the service should be available at. If the time Available At date is not provided, then the request is assuming that the request does not have a specific date in mind *~~NOTE: BE PARTICULARLY CAREFUL OF TIME-ZONE INFORMATION HERE.~~* *~~THE TIME-ZONE THAT IS PROVIDED TO THE CURRENT LOCAL SYSTEM TIME IS USED AS A CROSS REFERENCE TO THIS FIELD ALSO. IT CAN ENSURE THAT WHEN REQUESTING THE AVAILABLE AT THE TIME-ZONE REQUEST IS CONSISTENTLY APPLIED.~~* *NOTE: THE TIME-ZONE INFORMATION MUST BE INCLUDED IN THE DATE-TIME VALUE, OTHERWISE UTC WILL BE ASSUMED.*
CheckDate	Boolean	If this option is selected, the date component of the AvailableAt parameter is included in the filter. *NOTE: SETTING THIS VALUE TO TRUE WILL THEN IGNORE THE DAYSOFWEEK OPTION.*
CheckTime	Boolean	If this option is selected, the time component of the AvailableAt parameter is included in the filter.
DaysOfWeek	DaysOfWeek	Which day(s) of the week should be checked. *NOTE: SELECTING MULTIPLE DAYS WITH THIS FIELD WILL LOOK FOR RECORDS THAT HAVE AT LEAST ONE OF THE SPECIFIED DAYS AVAILABLE.* **Example:** Mon or Tues *Note: This property will be ignored if the CheckDate flag (above) is set*
IncludeUnknown	Boolean	Include Services where the Availability information is unknown. The typical use of this would be to filter only for those where the availability is actually known. Not including this availability restriction type means that the search will not consider availability.

Referenced By:

- Parameters.SearchInputParameters.Availability (collection)

8.8.4 HasAttribute

The Has Attribute Parameter is used to specify that Provider, Site, or Service Site has the defined value.

NOTE: THIS CODE TYPE THAT IS APPLICABLE TO THE VALUE IN THIS OBJECT CAN EITHER BE ASSOCIATED WITH ONE OF THE FOLLOWING TYPES DEPENDING ON THE PARENT OF THIS RECORD; SERVD.ATTT.ORG, SERVD.ATTT.SITE,

Property	Data Type	Description
AttributeType	String (64) *(mandatory)*	The Attribute Type Associated
Value	String (250)	The Value to search for given the specified Attribute Type
IncludeExpired	Boolean	Include attributes that have an Expired Verification Status
IncludePending	Boolean	Include attributes that have a Pending Verification Status
IncludeNotValid	Boolean	Include attributes that have a NotValid Verification Status *NOTE: THOSE THAT FAILED VERIFICATION WITH A SERVD VERIFIER*

Referenced By:

- Parameters.OrganizationSearchParameters.Attributes (collection)
- Parameters.ProviderSearchParameters.Attributes (collection)
- Parameters.ServiceSiteSearchParameters.Attributes (collection)
- Parameters.SiteSearchParameters.Attributes (collection)

8.8.5 OrganizationSearchParameters

Search for Organizations that have the following properties.

Property	Data Type	Description
OrganizationId	String (50)	A specific Organization ID to locate
Name	String	The Name of the Organization to Search for (when filtering for this field, ensure that the SearchControlParameter StringMatchType is utilized)
Attributes	HasAttribute (collection)	A list of Attributes to filter the Organization by *NOTE: IF THE VALUE SPECIFIED IN THE ATTRIBUTE IS EMPTY, THE REQUEST IS INTENDED TO LOCATE ORGANIZATIONS THAT HAVE THIS ATTRIBUTE TYPE, AND ITS VALUE IS NOT IMPORTANT.*
AffiliationType	String (64)	Specify that we are searching for an Organization that has an Affiliation with a specific Affiliation. **cts2:**ServD.OAFT

Property	Data Type	Description
IdentifierType	String (64)	Indicates that we are looking for an Organization that has a specific identifier included (not its actual value). *Note: Specifying this does not make the IdentiferValue field mandatory.* **Example:** HPI-O
IdentifierValue	String (64)	The value of the Identifier that an Organization must have. If specified, the IdentifierType field must also be populated.
SpecialtyCode	String (64)	The code of a specialty to filter Organizations by **cts2:**ServD.SPCLTY.Org
NameType	String (64)	Specify the Name type to be returned on the Search Items in the returned collection. If the specific type is not available, the ServD Core can choose which Record to include. **Example:** Trading Name **cts2:**ServD.ONT
NamePupose	String (64)	Specify the Name Purpose to be returned on the Search Items in the returned collection. If the specific purpose is not available, the ServD Core can choose which Record to include. **Example:** Search Results **cts2:**ServD.ONP
AddressType	String (64)	Specify the Address type to be returned on the Search Items in the returned collection. If the specific type is not available, the ServD Core can choose which Record to include. **Example:** Administration **cts2:**ServD.ADRT
AddressPupose	String (64)	Specify the Address Purpose to be returned on the Search Items in the returned collection. If the specific purpose is not available, the ServD Core can choose which Record to include. **Example:** Search Results **cts2:**ServD.AP

Property	Data Type	Description
WebAddressContactPoint Purpose	String (64)	Specify the Contact Purpose to be returned on the Search Items in the returned collection. If the specific purpose is not available, the ServD Core can choose which Record to include. *NOTE: THE CONTACT POINT TYPE THAT WILL BE USED IS DEFINED IN THE SERVD FEDERATION'S PROFILE* **cts2:**ServD.CPT

Referenced By:

- Parameters.SearchInputParameters.OrganizationSearchParameters (collection)

8.8.6 ProviderSearchParameters

Search for Providers that have the following properties.Filter options for the searching for records that are within a specified location (by co-ordinates)
This will check that the latitude and longitude of a Sites SiteGeographicalCoordinate falls within the circle created by these parameters.

Property	Data Type	Description
ProviderId	String(50)	The ID of the Provider to search for.
Surname	String	The Surname of the Organization to Search for (when filtering for this field, ensure that the SearchControlParameter StringMatchType is utilized).
GivenNames	String	The Given Name(s) of the Provider to Search for (when filtering for this field, ensure that the SearchControlParameter StringMatchType is utilized).
CulturalEthnicityCode	String (64)	The ethnic background of the Provider. **cts2:**ServD.ETH
LanguageSpokenCode	String (64)	The language spoken by the Provider. **cts2:**ServD.LAN
GenderCode	String (64)	The gender of the Provider to filter to. **cts2:**ServD.GEN
SpecialInterests	String	The special interests of the Provider.
Attributes	HasAttribute (collection)	Filter Providers that have the specified Attributes. *NOTE: IF THE VALUE SPECIFIED IN THE ATTRIBUTE IS EMPTY, THE REQUEST IS INTENDED TO LOCATE PROVIDERS THAT HAVE THIS ATTRIBUTE TYPE, AND ITS VALUE IS NOT IMPORTANT.*

Referenced By:

- Parameters.SearchInputParameters.ProviderSearchParameters (collection)

8.8.7 ProximitySearchParameters

Filter options for the searching for records that are within a specified location (by co-ordinates)
This will check that the latitude and longitude of a Sites SiteGeographicalCoordinate falls within the circle created by these parameters.

Property	Data Type	Description
SpatialReferenceID	String (64) *(mandatory)*	The Co-ordinate system being used (Spatial Reference ID - SRID). *NOTE: THE SPATIAL REFERENCE IDENTIFICATION SYSTEM IS DEFINED BY THE EUROPEAN PETROLEUM SURVEY GROUP (EPSG) STANDARD, WHICH IS A SET OF STANDARDS DEVELOPED FOR CARTOGRAPHY, SURVEYING, AND GEODETIC DATA STORAGE. THIS STANDARD IS OWNED BY THE OIL AND GAS PRODUCERS (OGP) SURVEYING AND POSITIONING COMMITTEE.* **Example:** WGS84 Other relevant information can be found at http://sharpgis.net/post/2007/05/Spatial-references2c-coordinate-systems2c-projections2c-datums2c-ellipsoids-e28093-confusing.aspx
ProximityOfLatitude	Double *(mandatory)*	The Latitude of the reference co-ordinate.
ProximityOfLongitude	Double *(mandatory)*	The Longitude of the reference co-ordinate.
ProximityRadiusTo Search	Double *(mandatory)*	Radius to search (in meters).

Referenced By:

- Parameters.SearchInputParameters.ProximitySearchParameters

8.8.8 RecordIndex

An index to a Record by its primary identifier.

Property	Data Type	Description
RecordType	RecordTypeEnum	Which record type this index relates to.
Identifier	String	The Primary Identifier of the Record this index relates to.

8.8.9 SearchControlParameters

The Search Control Parameters are used to specify what level of data is to be retrieved.

Property	Data Type	Description

Property	Data Type	Description
GetUnmoderatedDataWhere Available	Boolean	Where there is data that is being moderated and you are either the author or approver, explicitly request to see the un-moderated data elements. When this value is false, all searches will return the same fully moderated data, regardless of the user type (Author, Approver or Searcher).
PageSize	Integer	The Number of items to be included in a page request.
StringMatchType	StringMatchTypeEnum	The type of search to perform.

8.8.10 SearchInputParameters

The Search Input parameters are used by the Search interface operations to filter data to find the data that they are looking for. This can be a generic free text style search using the **SearchFor** property, or a more specific filter that explicitly sets filter properties. The location or proximity properties must be specified to give a context of where you are looking to receive the service.

Property	Data Type	Description
AddressSearch Parameters	AddressSearch Parameters *(mandatory)*	Simple Address based searching. This does not have any direct relevance to the coverage areas used by the **Locator** Interface.
ProximitySearch Parameters	ProximitySearch Parameters	Geographic point based filtering. This does not have any direct relevance to the coverage areas used by the **Locator** Interface.
SearchFor	String	The generic "Google" style search string. This can be interpreted by the Search Service anyway that it chooses to locate services. You must select another of the Search flags to indicate which types should be searched. If none are selected, the search will assume it is searching for services. This can also include be interpreted to search for Keywords. Reference: Note: This could be used in OpenSearch style interfaces. (http://www.opensearch.org/Specifications/OpenSearch/1.1) Note: This could also be customized to produce a Search Provider. http://blogs.msdn.com/b/jonbox/archive/2009/09/09/create-a-simple-ie8-search-provider.aspx
CoverageAreaCode	String	The Coverage Area to search within. (This can be used by the ServD Locator to determine which ServD Core endpoints to search) **cts2:**ServD.CA

Property	Data Type	Description
StructuredQuery	String	An Additional Property that can be used as a structured query using techniques such as query by example. The actual format of this syntax is not defined by this standard, and can be defined by the implementer of the Search interface. (Optional)
SearchOrganizations	Boolean	This field is used when the **SearchFor** parameter is specified.
SearchProviders	Boolean	This field is used when the **SearchFor** parameter is specified.
SearchServiceSites	Boolean	This field is used when the **SearchFor** parameter is specified.
SearchSites	Boolean	This field is used when the **SearchFor** parameter is specified.
ModifiedAfterDate	DateTime (nullable)	When specified only include rows where there has been a change after this date Note: modification could also be from a child object of this record
Availability	Availability (collection)	Specifically look for a service that is available at/between the provided times.
OrganizationSearch Parameters	OrganizationSearch Parameters (collection)	Search for details explicitly related to Organization properties.
SiteSearchParameters	SiteSearchParameters (collection)	Search for details explicitly related to Site properties.
ServiceSiteSearch Parameters	ServiceSiteSearch Parameters (collection)	Search for details explicitly related to Service Site properties.
ProviderSearch Parameters	ProviderSearch Parameters (collection)	Search for details explicitly related to Provider properties.

8.8.11 ServiceSiteSearchParameters

Search for Service Sites that have the following properties.

Property	Data Type	Description
ServiceSiteId	String(50)	The ID of the ServiceSite to search for.

Property	Data Type	Description
NoAppointment Reqired	ConditionalIndicator Enum	Indicates whether or not a prospective consumer will require an appointment for a particular service at a Site. To be provided by the Organization. Indicates if an appointment is required for access to this service. If this flag is 'NotDefined,'then this flag is be overridden by the Site's availability flag.
FreeProvisionCode	String (64)	The free provision code provides a link to the Free Provision reference entity to enable the selection of one free provision type. **cts2:**ServD.SSFPC
ServiceCategory	String (64)	Identifies the broad category of service being performed or delivered. Selecting a Service Category then determines the list of relevant service types that can be selected in the Primary Service Type. **cts2:**ServD.SC
ServiceType	String (100)	The Code value of the Service Type. **cts2:**ServD.ST
SpecialtyCode	String (64)	The code of a specialty to filter Organizations by. **cts2:**ServD.SPCLTY.SS
TargetGroup	String (64)	The value from the reference list of the Target Group for the actual Service. **cts2:**ServD.SSTG
Attributes	HasAttribute (collection)	The Service Site Attribute can define things such as Bulk Billing, HMO attributes etc. These will be defined by specific CTS2 mappings where the ontology has not been defined. *NOTE: IF THE VALUE SPECIFIED IN THE ATTRIBUTE IS EMPTY, THE REQUEST IS INTENDED TO LOCATE SERVICE SITE(S) THAT HAVE THIS ATTRIBUTE TYPE, AND ITS VALUE IS NOT IMPORTANT.*
IdentifierType	String (64)	Indicates that we are looking for a Service Site that has a specific identifier included (not its actual value). *NOTE: SPECIFYING THIS DOES NOT MAKE THE IDENTIFERVALUE FIELD MANDATORY.*
IdentifierValue	String (64)	The value of the Identifier that a Service Site must have. If specified, the IdentifierType field must also be populated.

Referenced By:

- Parameters.SearchInputParameters.ServiceSiteSearchParameters (collection)

8.8.12 SiteSearchParameters

Search for Sites that have the following properties.

Property	Data Type	Description
SiteId	String (50)	A specific Site ID to locate.
Name	String	The Name of the Site to Search for (when filtering for this field, ensure that the SearchControlParameter StringMatchType is utilized).
Attributes	HasAttribute (collection)	A list of Attributes to filter the Site by. *NOTE: IF THE VALUE SPECIFIED IN THE ATTRIBUTE IS EMPTY, THE REQUEST IS INTENDED TO LOCATE SITES THAT HAVE THIS ATTRIBUTE TYPE, AND ITS VALUE IS NOT IMPORTANT.*

Referenced By:

- Parameters.SearchInputParameters.SiteSearchParameters (collection)

8.9 Results

8.9.1 Overview

class Results

«enumeration»
Common::RecordUpdateResultStatusEnum

Success = 0
SuccessRequiresModeration = 1
ErrorPermission = -1
PartialRecordNotUpdatable = -2
ErrorDuplicateRecord = -3
ErrorValidation = -4
ErrorOther = -5

DeleteRecordResult

UpdateRecordResult

+Result
1

+Result
1

+RecordType
1

+RecordType
1

«enumeration»
Common::RecordTypeEnum

Organization = 1
OrganizationAffiliation = 2
OrganizationName = 3
Provider = 10
ProviderLanguage = 11
ProviderName = 12
ServiceSite = 20
ServiceSiteProvider = 21
ServiceSiteSetting = 22
ServiceSiteTargetGroup = 23
Site = 30
SiteGeographicalCoordinate = 31
Address = 40
Attribute = 41
Available = 42
ContactPoint = 43
NotAvailable = 44
Keyword = 45
Identifier = 46
CatchmentArea = 47
Specialty = 48
ReferralMethod = 49
CoverageArea = 50

Result
Common::RecordResult

+ ExternalId :string
+ Identifier :String
+ RetrieveDetailsURI :String

PublicKey

+ Identifier :String
+ Value :string

+RecordType
1

RecordRequiringApproval

+ Description :String
+ Identifier :String
+ LastChangeByUser :String
+ LastChangeDate :DateTime
+ Parent :RecordIndex
+ RecordStatus :RecordStatusEnum
+ RetrieveDetailsURI :String
+ Summary :String

+RecordType
1

AttributeVerificationResult

+ AdditionalNotes :String
+ AttributeType :string
+ AttributeValue :string
+ DigitalSignature :Byte [0..*]
+ ExpiryDate :DateTime [0..1]
+ FamilyName :string
+ GivenNames :string
+ Identifier :string
+ Name :string
+ PublicKey :string
+ VerificationString :string

+VerificationStatus
1

+Addresses
0..*

ModeratedRecord
DataModel::Address

+ContactPoints
0..*

ModeratedRecord
DataModel::ContactPoint

«enumeration»
Common::AttributeVerificationStatusEnum

NotApplicable = 0
NotVerifiable = 1
Pending = 2
Verified = 3
Expired = 4
NotValid = 5

ReferenceListDetail

+ CodeType :string
+ Description :string
+ URL :string

LocatorUpdateResult

+ AdvancedErrorDescription :string
+ ErrorCode :string
+ ErrorDescription :string
+ NumberOfAreasAdded :int
+ NumberOfAreasAlreadyExisting :int
+ NumberOfAreasRemoved :int

class Results

DeleteRecordResult

+	Result :RecordUpdateResultStatusEnum

+RecordType →

Result
Common::
RecordResult

+	ExternalId :string
+	Identifier :String

UpdateRecordResult

+	Result :RecordUpdateResultStatusEnum

+RecordType →

«enumeration»
Common::RecordTypeEnum

Organization = 1
OrganizationAffiliation = 2
OrganizationName = 3
Provider = 10
ProviderLanguage = 11
ProviderName = 12
ServiceSite = 20
ServiceSiteProvider = 21
ServiceSiteSetting = 22
ServiceSiteTargetGroup = 23
Site = 30
SiteGeographicalCoordinate = 31
Address = 40
Attribute = 41
Available = 42
ContactPoint = 43
NotAvailable = 44
Keyword = 45
Identifier = 46
CatchmentArea = 47
Specialty = 48
ReferralMethod = 49
CoverageArea = 50

PublicKey

+	Identifier :String
+	RecordType :RecordTypeEnum
+	Value :string

+RecordType →

RecordRequiringApproval

+	Description :String
+	Identifier :String
+	LastChangeByUser :String
+	LastChangeDate :DateTime
+	RecordType :RecordTypeEnum
+	Summary :String

+RecordType →

AttributeVerificationResult

+	AdditionalNotes :String
+	Addresses :ServD.DataModel.Address ([])
+	AttributeType :string
+	AttributeValue :string
+	ContactPoints :ServD.DataModel.ContactPoint ([])
+	DigitalSignature :byte ([])
+	ExpiryDate :DateTime?
+	FamilyName :string
+	GivenNames :string
+	Identifier :string
+	Name :string
+	PublicKey :string
+	VerificationStatus :AttributeVerificationStatusEnum
+	VerificationString :string

+VerificationStatus →

«enumeration»
Common::
AttributeVerificationStatusEnum

NotApplicable = 0
NotVerifiable = 1
Pending = 2
Verified = 3
Expired = 4
NotValid = 5

LocatorUpdateResult

+	AdvancedErrorDescription :string
+	ErrorCode :string
+	ErrorDescription :string
+	NumberOfAreasAdded :int
+	NumberOfAreasAlreadyExisting :int
+	NumberOfAreasRemoved :int

ReferenceListDetail

+	CodeType :string
+	Description :string
+	URL :string

Common::SearchItem

+ Heading :String
+ ImageURI :String
+ LastModifiedDate :DateTime
+ RecordStatus :RecordStatusEnum
+ RetrieveDetailsURI :String
+ SubHeading :String
+ Summary :String

«enumeration»
Common::SearchResults Status Enum

Success = 0
Error = 1
Partial = 2
ErrorInadequateSearchParameters = 3

Common::PagedSearchResults

+ CurrentPage :int
+ EstimatedTotalItems :int
+ EstimatedTotalPages :int
+ PageSize :Integer

+Status
1

OrganizationSearchItem

+ Comment :string
+ Country :string
+ Jurisdiction :string
+ Name :string
+ OrganizationId :String
+ PostalIdentificationCode :string
+ Region :string
+ StreetAddress :string
+ WebAddress :string

+Items
0..*

OrganizationList

+Organizations   0..1

SiteSearchItem

+ Comment :string
+ Country :string
+ Description :string
+ GISGranularity :int
+ GISLatitude :decimal
+ GISLongitude :decimal
+ Jurisdiction :string
+ LocationDescription :string
+ MapReference :string
+ OrganizationId :String
+ PostalIdentificationCode :string
+ Region :string
+ SiteAvailabilityExceptions :string
+ SiteAvailabilityTimeBlock :string
+ SiteId :String
+ StreetAddress :string
+ WebSite :string

+Items
0..*

SiteList

+Sites   0..1

+ContactPoints
0..*

+ContactPoints
0..*

Moderated Record
DataModel::
ContactPoint

+ContactPoints
0..*

SearchResults

+ContactPoints
0..*

+ServiceSites   0..1

ServiceSiteSearchItem

+ OrganizationId :String
+ ServiceName :String
+ ServiceSiteAvailabilityExceptions :string
+ ServiceSiteAvailabilityTimeBlock :string
+ ServiceSiteId :String
+ ServiceType :String
+ SiteId :String

+Items
0..*

ServiceSiteList

+Providers   0..1

ProviderSearchItem

+ Comment :string
+ CulturalEthnicityCode :string
+ FamilyName :string
+ GenderCode :string
+ GivenNames :string
+ OrganizationIds :String [0..*]
+ PrimarySiteId :String
+ ProviderId :String
+ ServiceSiteIds :String [0..*]
+ SiteIds :String [0..*]
+ SpecialInterest :string
+ TitleCode :string

+Items
0..*

ProviderList

+Languages
0..*

Moderated Record
DataModel::
ProviderLanguage

+Errors   0..*

Common::Result

+ AdvancedErrorDescription :string
+ ErrorCode :string
+ ErrorDescription :string

class Results

Common::SearchItem
+ Heading :String
+ ImageURI :String
+ RetrieveDetailsURI :String
+ SubHeading :String
+ Summary :String

«enumeration»
Common::SearchResults Status Enum
Success = 0
Error = 1
Partial = 2
ErrorInadequateSearchParameters = 3

Common::PagedSearchResults
+ CurrentPage :int
+ Errors :Result ([])
+ EstimatedTotalItems :int
+ EstimatedTotalPages :int
+ Status :SearchResultsStatusEnum

+Status

OrganizationSearchItem
+ Comment :string
+ ContactPoints :ContactPoint ([])
+ Country :string
+ Jurisdiction :string
+ Name :string
+ OrganizationId :String
+ PostalIdentificationCode :string
+ Region :string
+ StreetAddress :string
+ WebAddress :string

+Items

OrganizationList
+ Items :OrganizationSearchItem ([])

+Organizations

SiteSearchItem
+ Comment :string
+ ContactPoints :ContactPoint ([])
+ Country :string
+ Description :string
+ GISGranularity :int
+ GISLatitude :decimal
+ GISLongitude :decimal
+ Jurisdiction :string
+ LocationDescription :string
+ MapReference :string
+ OrganizationId :String
+ PostalIdentificationCode :string
+ Region :string
+ SiteAvailabilityExceptions :string
+ SiteAvailabilityTimeBlock :string
+ SiteId :String
+ StreetAddress :string
+ WebSite :string

+Items

SiteList
+ Items :SiteSearchItem ([])

+Sites

SearchResults
+ Organizations :OrganizationList
+ Providers :ProviderList
+ ServiceSites :ServiceSiteList
+ Sites :SiteList

ServiceSiteSearchItem
+ ContactPoints :ContactPoint ([])
+ OrganizationId :String
+ ServiceName :String
+ ServiceSiteAvailabilityExceptions :string
+ ServiceSiteAvailabilityTimeBlock :string
+ ServiceSiteId :String
+ ServiceType :String
+ SiteId :String

+Items

ServiceSiteList
+ Items :ServiceSiteSearchItem ([])

+ServiceSites

ProviderSearchItem
+ Comment :string
+ ContactPoints :ContactPoint ([])
+ CulturalEthnicityCode :string
+ FamilyName :string
+ GenderCode :string
+ GivenNames :string
+ Languages :ProviderLanguage ([])
+ OrganizationIds :String ([])
+ PrimarySiteId :String
+ ProviderId :String
+ ServiceSiteIds :String ([])
+ SiteIds :String ([])
+ SpecialInterest :string
+ TitleCode :string

+Items

ProviderList
+ Items :ProviderSearchItem ([])

+Providers

8.9.2 Overview (part 2)

The Search methods return a Search Item that contains a subset of the data relevant to the entity that is found. This simple class has generic properties that may be populated by the ServD Core Component with content that they configure. Each of the Organization, Site, Service Site and Provider records further extend this generic information with some atomized data that could be used to display the results in a grid.

The data itself is not sorted intentionally, and the searching user is expected to refine their search till they find what they are searching for. This is consistent with other internet search experiences, such as Google search.

8.9.3 AttributeVerificationResult

The Attribute Verification Result is returned by the operations on the **Verify Details** interface to indicate the validity of a claim of a record to have a specific attribute.

Property	Data Type	Description
AttributeType	String (64)	The Attribute type that has been Verified **Example:** business.gov.au **Example:** Health Insurance Agency A **Example:** AHPRA
AttributeValue	String (64)	The Attribute Value that has been verified **Example:** ABN **Example:** Level A **Example:** Dentist
VerificationStatus	AttributeVerification StatusEnum	The result of an Attribute Verification result
AdditionalNotes	String (2048)	Additional Notes provided about this attribute by a **ServD Verifier**. **Example:** This Provider was delisted on 12/1/2012 for breaches of privacy
Identifier	String (64)	The identifier that was requested to be verified **Example:** 28 548 551 396 **Example:** HIA-000321 **Example:** DEN000123a
Name	String (250)	The name that is stored with this identifier's registration **Example:** DATABASE CONSULTANTS AUSTRALIA
FamilyName	String	For Providers, this is their Surname **Example:** Postlethwaite

Property	Data Type	Description
GivenNames	String	For Providers, this is their given names **Example:** Brian
ExpiryDate	DateTime (nullable)	The date that this record will expire
VerificationString	String (458)	The Verification String is the concatenation of the AttributeType, AttributeValue, Status, Identifier, Name and ExpiryDate Fields (in that order) as received from the sender. This can be reconstructed from the base data, but is included in the result to remove any ambiguity as to what was used to sign and create the DigitalSignature. Each field is to be separated by a ":" and the date is to be formatted as YYYY-MM-DD. **Example:** "ABN:123456789A:Verified:DEN12314:General Healthcare:2012-03-12" If there is no expiry date, then the value included for the date will be an empty string. **Example:** "ABN:123456789A:Verified:DEN12314:General Healthcare:" For a Provider, the Name field is replaced with the FamilyName and Given names separated by commas **Example:** "ABN:123456789A:Verified:DEN12314:Smith,Brian:2012-03-12" *Note: String Length: 64+1+64+1+1+1+64+250+1+10 (Calculated from the associated fields)*
PublicKey	String	This public key that can be used to verify the Digital Signature The length of this string is undefined and should be of an un-restricted type. *NOTE: THIS IS A BASE64 ENCODED DIGITAL CERTIFICATE. THE SERVD FEDERATION'S PROFILE WILL SPECIFY THE TYPE OF CERTIFICATE TO BE USED AND ITS PURPOSE.*
DigitalSignature	Byte (collection)	This component is the signed VerificationString as defined above using the PublicKey provided above.
ContactPoints	ContactPoint (collection)	Collection of Contact points for this Site
Addresses	Address (collection)	Collection of Addresses that are currently registered for this Organization/Site/Provider

RETURNED BY:

- Interface.VerifyDetails.VerifyOrganization
- Interface.VerifyDetails.VerifyProvider
- Interface.VerifyDetails.VerifySite

8.9.4 DeleteRecordResult

Derived from: Common.RecordResult

The Delete Record Result indicates that success or failure of a single delete record request. It is returned as a collection by the DeleteRecords method on the Maintenance interface.

Property	Data Type	Description
Result	RecordUpdateResultStatus Enum	Actual result of the delete method

Returned By:

- Interface.Maintenance.DeleteRecords (collection)

8.9.5 LocatorUpdateResult

The result just provides a simple summary of the changes that were made (if any)

Property	Data Type	Description
NumberOfAreasAdded	Integer	The number of new Coverage Areas that have been added to the database
NumberOfAreasAlreadyExisting	Integer	The number of areas that were already registered in the **ServD Locator**
NumberOfAreasRemoved	Integer	Number of Coverage Areas that were removed from the **ServD Locator**
ErrorCode	String	A System provided error code. (Optional)
ErrorDescription	String	A textual representation of the error that has occurred, intended to be displayed to a user as to what has occurred. *NOTE: IF RELEVANT THE NAME AND VALUE OF ANY FIELDS IN ERROR SHOULD BE INCLUDED*
AdvancedErrorDescription	String	This Advanced Error Message description is intended for logging and diagnosis of issues, and not expected to be displayed to users. (Optional)

Returned By:

- Interface.LocatorMaintenance.RegisterSearchEndpointForCoverageArea
- Interface.LocatorMaintenance.UnregisterSearchEndpointForAllCoverageAreas
- Interface.LocatorMaintenance.UnregisterSearchEndpointForCoverageArea

8.9.6 OrganizationList

Derived from: Common.PagedSearchResults

A Paged List of Organizations returned from the Search method.

Property	Data Type	Description
Items	OrganizationSearchItem (collection)	The actual list of Organization items for this page of data.

Referenced By:

- Results.SearchResults.Organizations

Returned By:

- Interface.Search.MoreOrganizations

8.9.7 OrganizationSearchItem

Derived from: Common.SearchItem

The Organization Search Item contains a subset of the Organization information that has been restricted down into an atomized form that can be displayed. Generic information can be populated into the base class SearchItem's Summary properties as desired by a ServD Core instance or component.

Property	Data Type	Description
OrganizationId	String (50)	ID of the Organization
Name	String	The name referenced. From the Organization Name record where Purpose = "Search result"
WebAddress	String	Details the Organization's web address. The exact web address (URL) required to find the website on any internet browser. From the Contact Point where Contact Point Type is the ServD Federation's Profile defined type for Web Contract References, and has the Purpose provided in the Organization Search Parameter.
Comment	String	Brief description of what is offered by an Organization, as a link to further information
StreetAddress	String	Civic address part. It may contain multiple lines. **Example:** 355 Spencer St *Note: Some systems may store this data in separated lines such as AddressLine1, AddressLine2*
Jurisdiction	String	(City, Town, Village, Area) points within a hierarchy that travels from Country down to the lowest level of jurisdiction within the specified country.
Region	String	The region or state of this address. **Example:** California

Property	Data Type	Description
PostalIdentification Code	String	The postal Identification Code contains the Zip Code or Postcode Usually this is defined by the national postal service. **Example:** 3003
Country	String	The country **Example:** Australia
ContactPoints	ContactPoint (collection)	Collection of Contact points for this Site.

Referenced By:

- Results.OrganizationList.Items (collection)

8.9.8 ProviderList

Derived from: Common.PagedSearchResults

A Paged List of Providers returned from the Search method

Property	Data Type	Description
Items	ProviderSearchItem (collection)	The actual list of Provider items for this page of data.

Referenced By:

- Results.SearchResults.Providers

Returned By:

- Interface.Search.MoreProviders

8.9.9 ProviderSearchItem

Derived from: Common.SearchItem

The Provider Search Item contains a subset of the Provider information that has been restricted down into an atomized form that can be displayed. Generic information can be populated into the base class SearchItem's Summary properties as desired by a ServD Core instance or component.

Property	Data Type	Description
ProviderId	String (50)	ID of the Provider
PrimarySiteId	String (50)	Primary Site Id
SiteIds	String (collection)	Collection of all Site Ids that this Provider is associated with.
OrganizationIds	String (collection)	Collection of all Organization Ids (that also associate with the Site Ids above) that this Provider is associated with.

Property	Data Type	Description
ServiceSiteIds	String (collection)	Collection of all ServiceSite Ids (that also associate with the Site Ids above) that this Provider is associated with.
TitleCode	String (64)	The Title of the Provider. **Example:** Mr, Mrs, Dr **cts2:**ServD.TITLE
GenderCode	String (64)	The Gender (sex) of the Provider. **cts2:**ServD.GEN
FamilyName	String	Details the Provider's family name.
GivenNames	String	Details the Provider's given name(s).
SpecialInterest	String	Brief description of any special professional interests a Provider may wish to list.
Comment	String	Brief description of what is offered by a Provider, as a link to further information.
CulturalEthnicityCode	String (64)	The Cultural Ethnicity code provides a link to the Cultural Ethnicity reference entity to enable the selection of one Cultural Ethnicity type. It is the self-identified cultural allegiance or background or ethnicity of the Provider. **cts2:**ServD.ETH
ContactPoints	ContactPoint (collection)	Collection of Contact points for this Site.
Languages	ProviderLanguage (collection)	The language(s) spoken by the Provider.

Referenced By:

- Results.ProviderList.Items (collection)

8.9.10 PublicKey

The PublicKey object is used as a return value for the GetPublicKeys method.

Property	Data Type	Description
RecordType	RecordTypeEnum	Which record type was modified.
Identifier	String (50)	The Primary Identifier of the Record that was modified.

Property	Data Type	Description
Value	String	The public part of the 'keys' allocated to an Organization by an accredited body to support secure exchange of data over the internet. To be provided by the Organization, where available. The length of this string is undefined and should be of an un-restricted type. *Note: This is a base64 encoded digital certificate. The ServD Federation's Profile will specify the type of certificate to be used and its purpose.*

Returned By:

- Interface.RetrieveDetails.RetrievePublicKeys (collection)

8.9.11 RecordRequiringApproval

The Record Requiring Approval is used to summarize a change to a record in the **Maintainable ServD Core** that requires approval by an Approver. It should be able to be used to present to a user a simple representation of the item to be approved so that it could be performed without having to actually load the complete record.

Property	Data Type	Description
RecordType	RecordTypeEnum	The record type that requires approval.
RecordStatus	RecordStatusEnum	The moderation Status of this record (Complete, Partial, RequiresModeration, Deleted)
Identifier	String	The Primary Identifier of the Record that requires approval.
LastChangeByUser	String	The user that has requested the change to be made.
LastChangeDate	DateTime	The date that the change was requested.
Summary	String	The Summary includes the basic description of the entity that is being changed. It is not to contain any mark-up of data. **Example:** For an Organization Address this could be the Organization's Name
Description	String	(Optional) The Description of the Record or it's requested changes. (Recommended) Where this record is externally verified using the VerifyDetails attributes a note as to the current status would be could to be displayed here also. This would permit an approver to wait until the external service has verified the data before approving the update to be visible externally. **Example:** The Address has Changed to 355 Spencer St

Property	Data Type	Description
RetrieveDetailsURI	String	The network location of the Retrieve Details Interface to request for the detailed information.
Parent	RecordIndex	The Record Index of the record that can be loaded from the RetrieveDetails interface. **Example:** For an Organization Address this would contain the Organization's Index

Returned By:

- Interface.Maintenance.GetRecordsRequiringApproval (collection)

8.9.12 ReferenceListDetail

Retrieve the details of a specific CTS2 Reference list so that connecting to the list is possible.

Property	Data Type	Description
CodeType	String (64)	The CTS2 code type for the reference list provided by this address.
URL	String	The Web Address of the Reference List that implements the CTS2 specification.
Description	String	This description can be used on the label to be shown to the user relating to the user interface component for selecting the value.

Returned By:

- Interface.Locator.GetReferenceListDetails (collection)

8.9.13 SearchResults

The Search Results object exposes the results of the Search Operations.
These collections are provided to permit the searcher to.

Property	Data Type	Description
Organizations	OrganizationList	The collection of Organization Summary Items .
Sites	SiteList	The collection of Site Summary Items.
ServiceSites	ServiceSiteList	The collection of Service Site Summary Items.
Providers	ProviderList	The collection of Provider Summary Items.

Returned By:

- Interface.Search.Search

8.9.14 ServiceSiteList

Derived from: Common.PagedSearchResults

A Paged List of Service Sites returned from the Search method.

Property	Data Type	Description
Items	ServiceSiteSearchItem (collection)	The actual list of Service Site items for this page of data.

Referenced By:

- Results.SearchResults.ServiceSites

Returned By:

- Interface.Search.MoreServiceSites

8.9.15 ServiceSiteSearchItem

Derived from: Common.SearchItem

The Service Site Search Item contains a subset of the Service Site information that has been restricted down into an atomized form that can be displayed. Generic information can be populated into the base class SearchItem's Summary properties as desired by a ServD Core instance or component.

Property	Data Type	Description
ServiceSiteId	String (50)	ID of the Service Site
SiteId	String (50)	ID of the Site
OrganizationId	String (50)	ID of the Organization
ServiceSiteAvailabilityTimeBlock	String	A Description of the Availability for this Service Site. It is constructed from the Service Site's AvailableTimes collection. *NOTE: IF THIS INFORMATION IS NOT PROVIDED BY THE SERVICE SITE, IT SHOULD RETURN THE VALUES ENTERED FOR THE SITE.*
ServiceSiteAvailabilityExceptions	String	A description of occasions that this Service Site is not available. It is constructed from the Service Site's NotAvailableTimes collection, and the AvailabilityExceptions field. (May contain HTML formatted text). *NOTE: IF THIS INFORMATION IS NOT PROVIDED BY THE SERVICE SITE, IT SHOULD RETURN THE VALUES ENTERED FOR THE SITE.* **Example:** e.g., public holidays, seasonal closures.
ServiceType	String (64)	Service type from the Primary Service Type on the Service Site record.
ServiceName	String (500)	Further description of the service.

Property	Data Type	Description
ContactPoints	ContactPoint (collection)	Collection of Contact points for this Site.

Referenced By:

- Results.ServiceSiteList.Items (collection)

8.9.16 SiteList

Derived from: Common.PagedSearchResults

A Paged List of Sites returned from the Search method.

Property	Data Type	Description
Items	SiteSearchItem (collection)	The actual list of Site items for this page of data.

Referenced By:

- Results.SearchResults.Sites

Returned By:

- Interface.Search.MoreSites

8.9.17 SiteSearchItem

Derived from: Common.SearchItem

The Site Search Item contains a subset of the Site's information that has been restricted down into an atomized form that can be displayed. Generic information can be populated into the base class SearchItem's Summary properties as desired by a ServD Core instance or component.

Property	Data Type	Description
SiteId	String (50)	ID of the Site
OrganizationId	String (50)	ID of the Organization
Description	String	A textual description of the Service Site.
SiteAvailabilityTimeBlock	String	A Description of the Availability for this Site. It is constructed from the Site's AvailableTimes collection.

Property	Data Type	Description
SiteAvailabilityExceptions	String	A description of occasions that this Site is not available. It is constructed from the Site's NotAvailableTimes collection, and the AvailabilityExceptions field. (May contain HTML formatted text). **Example:** e.g., public holidays, seasonal closures.
MapReference	String	Indicates the page and/or grid reference for the Site.
LocationDescription	String	A text location description for the Site. A succinct description of the location of the Site. The description should include any additional information to help visitors access the Site that is not immediately obvious from the address information. (May contain HTML formatted text.) **Example:** Access to 120 Spencer St. is from Frances St. **Example:** Adjacent to tram stop 45 on line 70.
Comment	String	Brief description of what is offered/included at a Site, as a link to further information (May contain HTML formatted text).
WebSite	String	Website URL for the Organization
GISLatitude	Decimal	The Latitude of the Site
GISLongitude	Decimal	The Longitude of the Site
GISGranularity	Integer	The Granularity of the Site
StreetAddress	String	Civic address part
Jurisdiction	String	(City, Town, Village, Area) points within a hierarchy that travels from Country down to the lowest level of jurisdiction within the specified country.
Region	String	The region or state of this address. **Example:** California
PostalIdentificationCode	String	The postal Identification Code contains the Zip Code or Postcode. Usually this is defined by the national postal service. **Example:** 3003
Country	String	The country **Example:** Australia
ContactPoints	ContactPoint (collection)	Collection of Contact points for this Site.

Referenced By:

- Results.SiteList.Items (collection)

8.9.18 UpdateRecordResult

Derived from: Common.RecordResult

The Update Record result is returned by the Maintenance operations while updating and adding data. It provides a simple structure by which the Maintenance Application can match its internal records that it is publishing into the **Maintainable ServD Core** to the new records. Typically this is returned as items in an array to refer to each record that was created updated and the status of the method.

Property	Data Type	Description
Result	RecordUpdateResultStatusEnum	Actual result of the update method.

Returned By:

- Interface.Maintenance.OrganizationAdd (collection)
- Interface.Maintenance.OrganizationUpdate (collection)
- Interface.Maintenance.SiteAdd (collection)
- Interface.Maintenance.SiteUpdate (collection)
- Interface.Maintenance.ServiceSiteAdd (collection)
- Interface.Maintenance.ServiceSiteUpdate (collection)
- Interface.Maintenance.ProviderAdd (collection)
- Interface.Maintenance.ProviderUpdate (collection)
- Interface.Maintenance.AddServiceSiteProvider
- Interface.Maintenance.UpdateServiceSiteProvider
- Interface.Maintenance.UploadSiteCertificate
- Interface.Maintenance.UploadServiceSiteCertificate
- Interface.Maintenance.ApproveRecords (collection)
- Interface.Maintenance.RejectRecords (collection)

8.10 Information Model

8.10.1 Overview
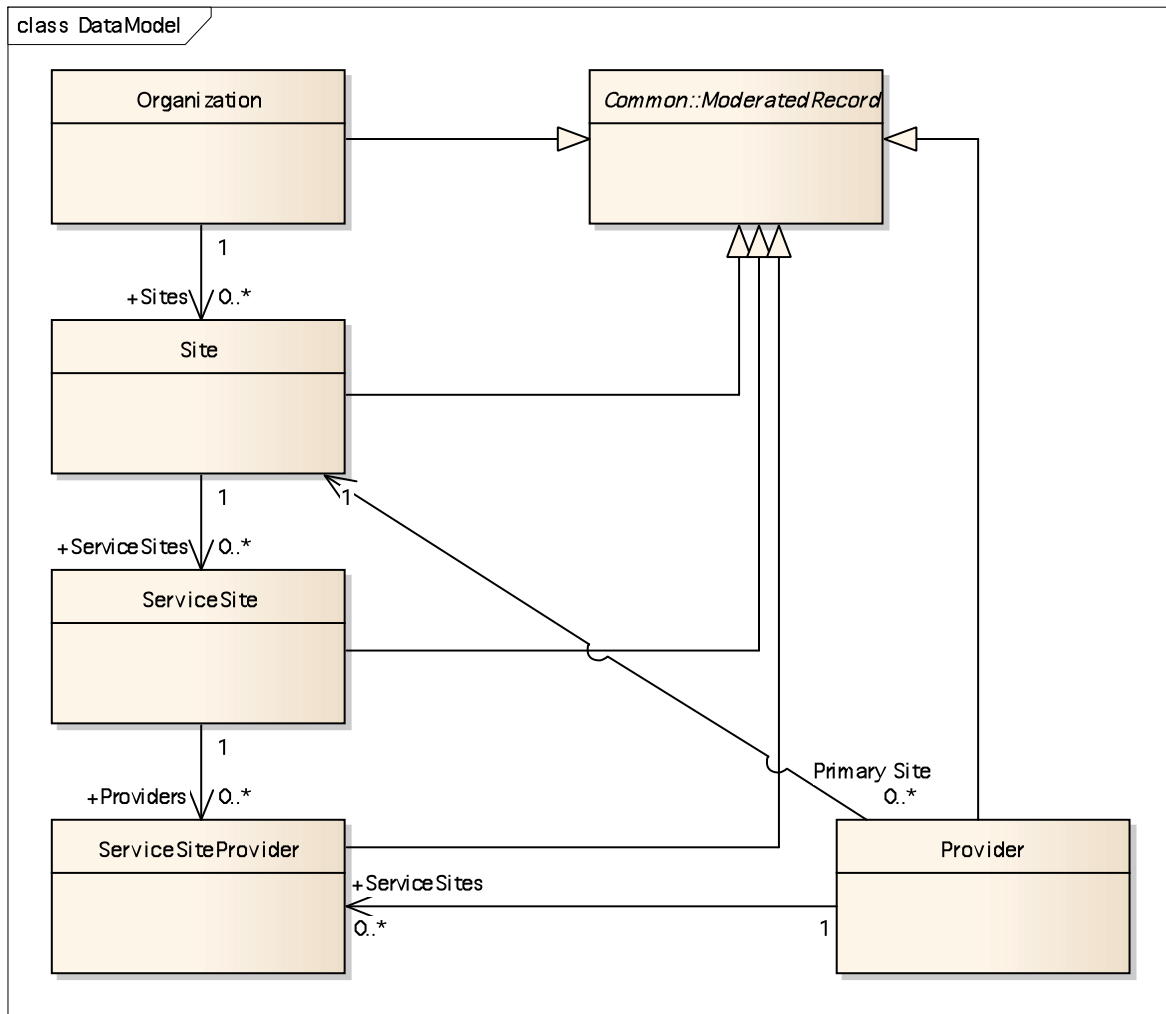
All objects in the ServD Information Model are specializations of a Moderated Record Type. This base type under all records provides the common properties required to support the content management workflow between Authors and Approvers, and also a history via the start and end date properties.

8.10.1.1   Organization

class DataModel - Organization

8.10.1.2   Site

class DataModel - Site
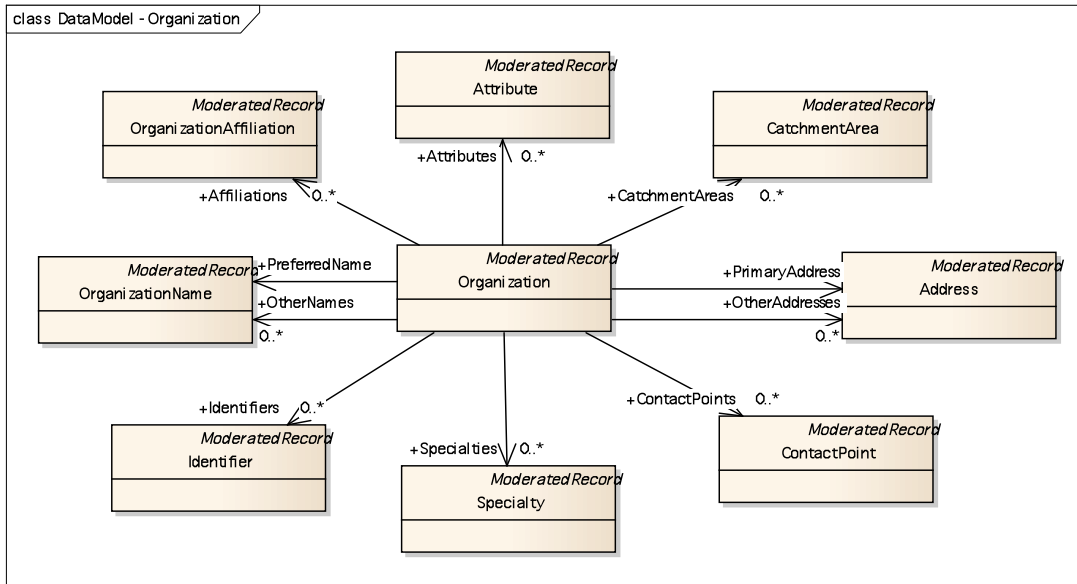
8.10.1.3 ServiceSite
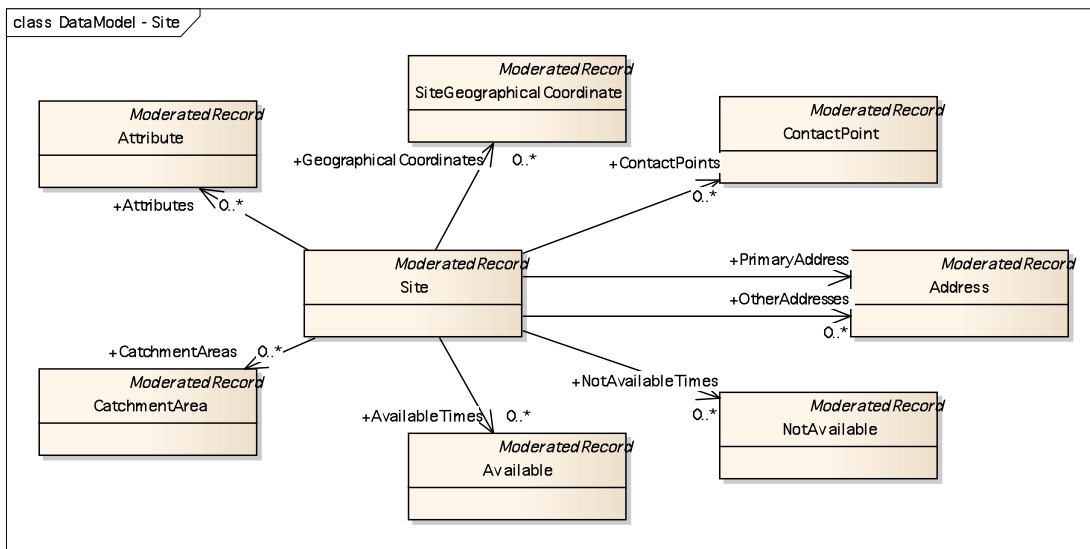
class DataModel - Service Site

8.10.1.4   Provider
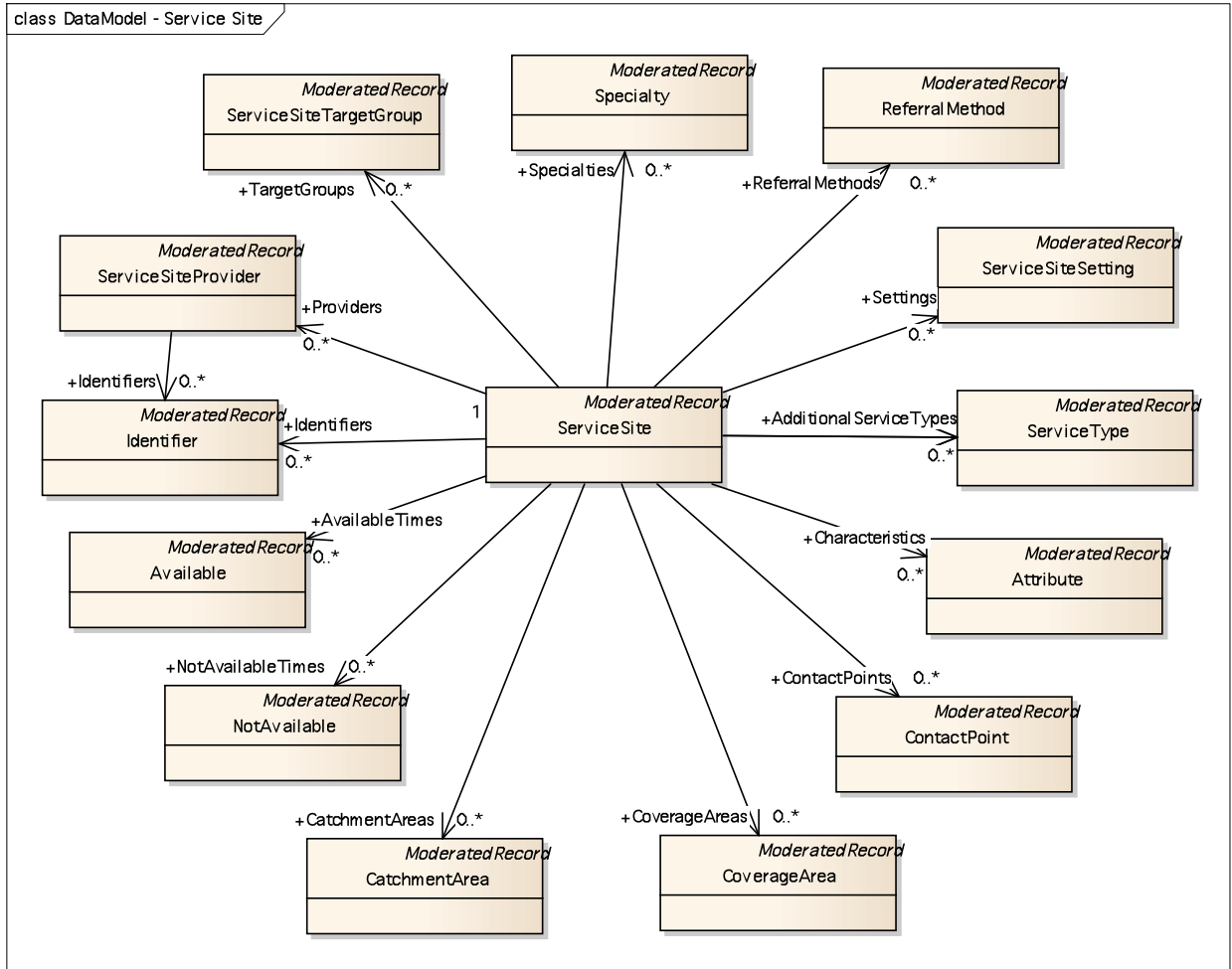
class DataModel - Provider
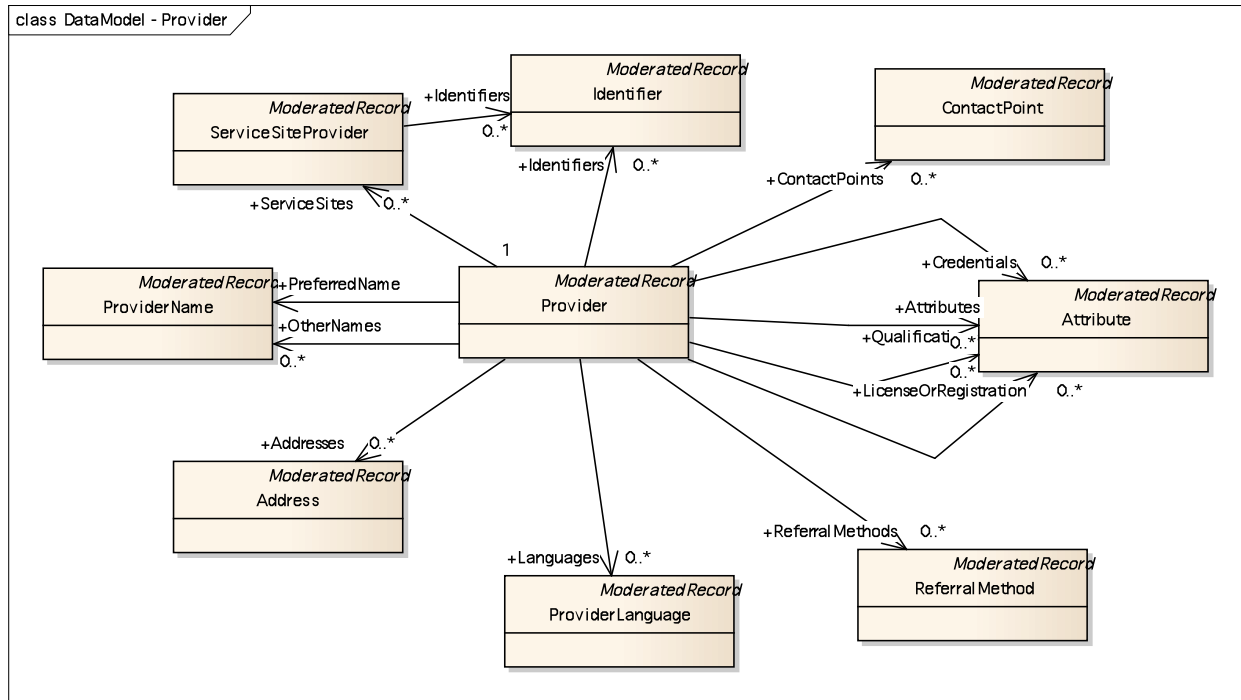
8.10.2 Address

Derived from: Common.ModeratedRecord

An Organization at a particular Site may specify address information that is in addition to, or over-rides, the Organization address identified above. Each address is specified by a type and a purpose. (When seeking an address of a particular type, begin by searching at the lowest level (Site) and if not found, progress up the Organization tree until such an address is found.)

Note: Refer to http://schema.org/PostalAddress for a possible schema for this

Property	Data Type	Description
AddressId	String (50)	A unique Identifier allocated by the saving system When creating a new record, this is blank and will be allocated by the system.
AddressType	String (64) *(mandatory)*	Defines the primary type of activity at a location. **Example:** Residential, Business, Temporary **cts2:**ServD.ADRT
AddressPurpose	String (64) *(mandatory)*	Defines the usage of this address in relation to a specific action. **Example:** Mailing, Delivery, Shipping, Search Results **cts2:**ServD.AP

Property	Data Type	Description
StreetAddress	String (1024)	Civic address part. It may contain multiple lines. **Example:** Level 5, 355 Spencer St *Note: Some systems may store this data in separated lines such as AddressLine1, AddressLine2*
Jurisdiction	String (256)	(City, Town, Village, Area) points within a hierarchy that travels from Country down to the lowest level of jurisdiction within the specified country. **Example:** West Melbourne
Region	String (256)	The region or state of this address. **Example:** Victoria
PostalIdentification Code	String (50)	The postal Identification Code contains the Zip Code or Postcode. Usually this is defined by the national postal service. **Example:** 3003
Country	String (256)	The country **Example:** Australia

Referenced By:

- DataModel.Organization.PrimaryAddress
- DataModel.Organization.OtherAddresses (collection)
- DataModel.Provider.Addresses (collection)
- DataModel.Site.PrimaryAddress
- DataModel.Site.OtherAddresses (collection)
- Results.AttributeVerificationResult.Addresses (collection)

8.10.3 Attribute

Derived from: Common.ModeratedRecord

An attribute is a configurable record that can be attached to several other records to provide an extensible and verifiable name value pair. These are restricted to a known set of names as defined by the ServD Federation's Profile.

Note: This code type that is applicable to the code Value in this object can either be associated with one of the following types depending on the parent of this record; ServD.ATTT.Org, ServD.ATTT.Site, ServD.ATTT.SS, ServD.ATTT.Prov, ServD.ATTT.Prov.Cred, ServD.ATTT.Prov.Qual or ServD.ATTT.Prov.Reg.

Property	Data Type	Description
AttributeId	String (50)	The unique reference identifying an attribute. This is the primary key for the attribute entity.

Property	Data Type	Description
AttributeType	String (64) *(mandatory)*	The type of attribute *Note: This has been restricted to 64 chars to support the use of OIDs Refer to http://www.hl7.org/oid/index.cfm for other details on the literal form of OIDs and why 64 chars has been selected.*
AttributeValue	String (250) *(mandatory)*	The actual value associated with this attribute.
VerificationStatus	AttributeVerificationStatusEnum	The verification status as provided by the external **ServD Verifier**. *Note: If the current date is after the Expiry date and this attribute was verified by a **ServD Verifier** to be **Verified**, then this value should be **Expired**.*
AdditionalVerifier Notes	String (2048)	Additional Notes provided about this attribute by a **ServD Verifier**. **Example:** This Provider was delisted on 12/1/2012 for breaches of privacy.
LastVerification Date	DateTime (nullable)	The date that this record was last verified by the **ServD Verifier**. Must be populated for attribute types that are configured for a **Maintainable ServD Core**. The date will be populated with the current date when saving a record until the **ServD Verifier** has been checked, at which time it will be updated. **Note**: Some configurations may decie to periodically re-check these credentials to ensure this information remains up to date.
ExpiryDate	DateTime (nullable)	The date that this attribute will expire. If this attribute was verified by a **ServD Verifier**, then this date will be provided by the **ServD Verifier**.

Property	Data Type	Description
VerificationString	String (458)	The Verification String is the concatenation of the AttributeType, AttributeValue, Status, Identifier, Name and ExpiryDate Fields (in that order) as received from the sender. This can be reconstructed from the base data, but is included in the result to remove any ambiguity as to what was used to sign and create the DigitalSignature. Each field is to be separated by a ":" and the date is to be formatted as YYYY-MM-DD. **Example:** "ABN:123456789A:Verified:DEN12314:General Healthcare:2012-03-12" If there is no expiry date, then the value included for the date will be an empty string. **Example:** "ABN:123456789A:Verified:DEN12314:General Healthcare:" For a Provider, the Name field is replaced with the FamilyName and Given names separated by commas. **Example:** "ABN:123456789A:Verified:DEN12314:Smith,Brian:2012-03-12" *Note: String Length: 64+1+64+1+1+1+64+250+1+10 (Calculated from the associated fields)*
DigitalSignature	String	The digital signature of the VerificationString as defined using the W3C XML-Signature Syntax and processing. The structure of this string is an XML fragment that complies with the XML Signature Schema Instance (XSD can be found at: http://www.w3.org/TR/2002/REC-xmldsig-core-20020212/xmldsig-core-schema.xsd). The length of this string is undefined and should be of an un-restricted type. *Note: Please refer to http://www.w3.org/TR/2002/REC-xmldsig-core-20020212/Overview.html for details on creating the digital signature and the process for verifying it.* *As the content of this string is not XML, it must be wrapped in a simple XML fragment. <verified><[CDATA[.......]]></verified> where the .... is replaced with the VerificationString* **Example:** <verified><[CDATA[ABN:123456789A:Verified:DEN12314:Smith,Brian:2012-03-12]]></verified>

Referenced By:

- DataModel.Organization.Attributes (collection)
- DataModel.Provider.Attributes (collection)
- DataModel.Provider.Credentials (collection)
- DataModel.Provider.Qualifications (collection)
- DataModel.Provider.LicenseOrRegistration (collection)
- DataModel.ServiceSite.Characteristics (collection)
- DataModel.Site.Attributes (collection)

8.10.4 Available

Derived from: Common.ModeratedRecord

Defines a block of time that is available.

Property	Data Type	Description
AvailableId	String (50)	A unique Identifier allocated by the saving system. When creating a new record, this is blank and will be allocated by the system.
DaysOfWeek	DaysOfWeek	Indicates which Days of the week are available between the Start and End Times.
AllDay	Boolean	Is this always available? (hence times are irrelevant)
AvailableStartTime	DateTime (nullable)	The opening time of day (the date is not included). *Note: If the AllDay flag is set, then this time is ignored.* *Note: Ensure that the time zone information is included*
AvailableEndTime	DateTime (nullable)	The closing time of day (the date is not included). *Note: If the AllDay flag is set, then this time is ignored.* *Note: Ensure that the time zone information is included*

Referenced By:

- DataModel.ServiceSite.AvailableTimes (collection)
- DataModel.Site.AvailableTimes (collection)

8.10.5 CatchmentArea

Derived from: Common.ModeratedRecord

This entity provides for grouping service sites into geographical areas appropriate for specific agencies, each with their own catchment area. This is to assist in the selection of sites for referral management.

Note: This often used as a Local Government Area

Property	Data Type	Description
CatchmentAreaId	String (50)	Directory assigned unique identifier
Code	String (64) *(mandatory)*	From the Catchment Area reference table **cts2:**ServD.CAC

Referenced By:

- DataModel.Organization.CatchmentAreas (collection)
- DataModel.ServiceSite.CatchmentAreas (collection)
- DataModel.Site.CatchmentAreas (collection)

8.10.6 ContactPoint

Derived from: Common.ModeratedRecord

An Organization at a particular Site will have different contact information (perhaps) than the information held at the Organization level independent of Site. This entity provides for any number of contact points that relate specifically to the Organization at that Site. Each contact point is identified through a type and a purpose.

NOTE: THE CONTACT_POINT_FORMAT_STRING FIELD HAS BEEN REMOVED FROM THIS TYPE.
IT WAS INTENDED TO BE USED TO ASSIST IN FORMATTING OF THE STRING, HOWEVER IT APPEARS THAT IT WAS POPULATED WITH A REGULAR EXPRESSION AND THUS MORE RELEVANT FOR VALIDATION, WHICH IS NOT SOMETHING THAT SHOULD BE ON THE OBJECT

Property	Data Type	Description
ContactPointId	String (50)	A unique Identifier that is allocated by the saving system. When adding a new contact point, this is blank and will be allocated.
ContactPointType	String (64) *(mandatory)*	The Type of the contact point. **Example:** Phone, Fax, Pager, email, Web **cts2:**ServD.CPT
ContactPointPurpose	String (64) *(mandatory)*	Description of the Purpose or usage of this contact point. **Example:** Reception, Billing, Referrals, Administration **cts2:**ServD.CPP
ContactOrderPreference	Integer	The preferred Order of usage. *NOTE: THIS WAS ADDED TO PERMIT THE ORDERING OF CONTACT LISTS IN TERMS OF PREFERENCE.*
EquipmentCode	String (64)	Type of physical apparatus that is attached to a phone, where applicable. **Example:** Cell, Fixed Line, TTY, NA **cts2:**ServD.CPEQ
Value	String (2048) *(mandatory)*	The Textual value of the Contact Point. **Example:** For Phone number: (555) 123 4565 **Example:** For email: person@theorganization.com **Example:** For web: www.omg.org

Referenced By:

- DataModel.Organization.ContactPoints (collection)
- DataModel.Provider.ContactPoints (collection)
- DataModel.ServiceSite.ContactPoints (collection)

- DataModel.Site.ContactPoints (collection)
- Results.AttributeVerificationResult.ContactPoints (collection)
- Results.OrganizationSearchItem.ContactPoints (collection)
- Results.ProviderSearchItem.ContactPoints (collection)
- Results.ServiceSiteSearchItem.ContactPoints (collection)
- Results.SiteSearchItem.ContactPoints (collection)

8.10.7 CoverageArea

Derived from: Common.ModeratedRecord

This entity provides the scope within the **ServD Locator** for the purpose of selecting which ServD Directories are to be searched for a specific area.

Property	Data Type	Description
CoverageAreaId	String (50)	Directory assigned unique identifier
Code	String (64) *(mandatory)*	The Coverage Area that facilitates the federation by the **ServD Locator**. **cts2:**ServD.CA

Referenced By:
- DataModel.ServiceSite.CoverageAreas (collection)

8.10.8 Identifier

Derived from: Common.ModeratedRecord

An Organization may be known by any number of external identifiers. Each identifier is specified by a type, an issuing authority, and a controlling authority.

Property	Data Type	Description
IdentifierId	String (50)	An internal unique Identifier that is allocated by the saving system. When adding a new Site, this is blank and will be allocated
IdentifierType	String (64) *(mandatory)*	Medical Record Number, Driver's License, Provincial Health Number, National Health Number, etc. **Example:** HPI-I or HPI-O
IdentifierIssuingAuthority	String (256)	Provincial Department of Health, St. Michael's Hospital (for example)
IdentifierControllingAuthority	String (256)	Public Works and Government Services, Health Region (for example)
Value	String (64) *(mandatory)*	The value of the Identifier.

REFERENCED BY:

- DataModel.Organization.Identifiers (collection)
- DataModel.Provider.Identifiers (collection)
- DataModel.ServiceSite.Identifiers (collection)
- DataModel.ServiceSiteProvider.Identifiers (collection)
- DataModel.Site.Identifiers (collection)

8.10.9 NotAvailable

Derived from: Common.ModeratedRecord

The NotAvailable object defines a range of dates that are not available.
Can be used for cases where the Service or Site is closed, or equipment for some services is under repair and will be unavailable for a period of time.

Property	Data Type	Description
NotAvailableId	String (50)	A unique Identifier allocated by the saving system. When creating a new record, this is blank and will be allocated by the system.
Description	String	The reason that can be presented to the user as to why this time is not available.
NotAvailableBetweenStartDate	DateTime (nullable)	Service is not available (seasonally or for a public holiday) from this date.
NotAvailableBetweenEndDate	DateTime (nullable)	Service is not available (seasonally or for a public holiday) until this date.

Referenced By:
- DataModel.ServiceSite.NotAvailableTimes (collection)
- DataModel.Site.NotAvailableTimes (collection)

8.10.10   Organization

Derived from: Common.ModeratedRecord

An Organization is any Organization, Agency, or collection of Organizations that participates in the delivery of health care services. An agency is not an individual service Provider, though an agency may be the legal representation of an individual service provider.

Property	Data Type	Description
OrganizationId	String (50)	The unique reference identifying an Organization.
Comment	String (2048)	Brief description of what is offered by an Organization, as a link to further information.
IncorporationDate	DateTime (nullable)	Date the Organization was made legally alive.

Property	Data Type	Description
Status	String (64)	The status of the Organization. **Example:** Unapproved, Active, Suspended, Delisted **cts2:**ServD.ORGST
ImageURI	String	If there is an image associated with this Organization, its URI can be included here. **Example:** If this Organization record was for the Mayo clinic, this value would be: http://www.mayoclinic.com/mymayo/img/logo.png
PreferredName	OrganizationName *(mandatory)*	The Preferred Name of the Organization.
OtherNames	OrganizationName (collection)	Collection of Other names the Organization is known as.
Identifiers	Identifier (collection)	Collection of Identifiers associated with this Organization.
PrimaryAddress	Address	Collection of addresses of this Organization (not usually the addresses of the sites where service are provided, often the head office which is not a service providing location).
OtherAddresses	Address (collection)	Collection of addresses of this Organization (not usually the addresses of the sites where service are provided, often the head office which is not a service providing location).
ContactPoints	ContactPoint (collection)	Collection of Contact points for this Site.
Attributes	Attribute (collection)	Collection of attributes for the Organization. **cts2:**ServD.ATTT.Org
Affiliations	OrganizationAffiliation (collection)	Collections of Affiliations associated with this Organization.
Specialties	Specialty (collection)	Collection of Specialties handled by the Organization. **cts2:**ServD.SPCLTY.Org
CatchmentAreas	CatchmentArea (collection)	Collection of catchment areas for this Organization.
Sites	Site (collection)	The collection of sites for this Organization.

Returned By:

- Interface.RetrieveDetails.RetrieveOrganizationDetails (collection)

8.10.11   OrganizationAffiliation

Derived from: Common.ModeratedRecord

An Organization might be associated with other organizations due to a number of different business needs or relevance, such as participation in integrated programs, geographic relevance to an area, or to identify insurance agreements.

Property	Data Type	Description
OrganizationAffiliationId	String (50)	Unique identifier for row
AffiliationType	String (64) *(mandatory)*	A code value representing another entity that the Organization is affiliated with. **cts2:**ServD.OAFT
Note	String (2048)	A description for the user to see that describes the details of the Organizations Affiliation.

Referenced By:

- DataModel.Organization.Affiliations (collection)

8.10.12   OrganizationName

Derived from: Common.ModeratedRecord

An agency may be known by different names distinguished by name type and name purpose.

Example: the legal name might be HollCroft Health Trust, but the trade name might be The HollCroft Center.

Property	Data Type	Description
OrganizationNameId	String (50)	Unique identifier for row
NameType	String (64) *(mandatory)*	The type of name that the Organization is known as. **Example:** Trading Name, Legal Name **cts2:**ServD.ONT
NamePurpose	String (64)	The purpose or usage of this Organization name. **Example:** Mailing Labels, Cheques, Correspondence, Search Results **cts2:**ServD.ONP
Name	String (250) *(mandatory)*	The name referenced.
IsLegalIndicator	Boolean	Set to true for the legal name, false for a non-legal name.

Referenced By:

- DataModel.Organization.PreferredName
- DataModel.Organization.OtherNames (collection)

8.10.13   Provider

Derived from: Common.ModeratedRecord

A Provider is any person that delivers clinical value in the provision of health services. Anyone who becomes a matter of record on the Electronic Health Record, or is part of a scheduling requirement (operator of an expensive piece of testing equipment perhaps) has to be identified as a Provider within this entity.

NOTE: THE SURNAME AND GIVEN NAME(S) HAVE BEEN MOVED TO THE PROVIDER NAME OBJECT

Property	Data Type	Description
ProviderId	String (50)	The unique reference identifying a Provider. This is the primary key for the Provider entity. Generated by the application.
GenderCode	String (64)	The Gender (sex) of the Provider. *NOTE: THE CONTENTS OF THE ATTRIBUTE ARE DETERMINED FROM THE SEX REFERENCE TABLE RENAMED FROM sex_code TO COMPLY WITH HCSP* **cts2:**ServD.GEN
CulturalEthnicityCode	String (64)	The Cultural Ethnicity code provides a link to the Cultural Ethnicity reference entity to enable the selection of one Cultural Ethnicity type. It is the self-identified cultural allegiance or background or ethnicity of the Provider. **cts2:**ServD.ETH
ImageURI	String	If there is an image associated with this Provider, its URI can be included here. **Example:** A URL for the providers photo could be inserted here.
PublicKey	String	The public part of the 'keys' allocated to an Organization by an accredited body to support secure exchange of data over the internet. To be provided by the agency, where available. The length of this string is undefined and should be of an un-restricted type. *NOTE: THIS IS A BASE64 ENCODED DIGITAL CERTIFICATE. THE SERVD FEDERATION'S PROFILE WILL SPECIFY THE TYPE OF CERTIFICATE TO BE USED AND ITS PURPOSE.*
SpecialInterest	String (2048)	Brief description of any special professional interests a Provider may wish to list.
Comment	String (2048)	Brief description of what is offered by a Provider, as a link to further information.
PrimarySiteId	String (50)	The Id of the Primary Site that this Provider works at.

Property	Data Type	Description
Status	String (64)	The status of this Provider. **Example:** Unapproved, Active, Suspended, Deregistered **cts2:**ServD.PROVST
Identifiers	Identifier (collection)	Collection of Identifiers associated with this Provider.
PreferredName	ProviderName *(mandatory)*	The Preferred Name of the Provider.
OtherNames	ProviderName (collection)	Collection of Other names the Provider is known as.
Addresses	Address (collection)	Collection of addresses of this Provider.
Attributes	Attribute (collection)	Collection of attributes for the Provider. **cts2:**ServD.ATTT.Prov
Credentials	Attribute (collection)	Collection of credentials (attributes) for the Provider. **cts2:**ServD.ATTT.Prov.Cred
Qualifications	Attribute (collection)	Collection of Qualifications (attributes) for the Provider. **cts2:**ServD.ATTT.Prov.Qual
LicenseOrRegistration	Attribute (collection)	Collection of Qualifications (attributes) for the Provider A Provider may be regulated by one or more external bodies, and only able to practice at the discretion of those bodies. This entity holds the particular status of the Provider within the various bodies under which they are controlled. **cts2:**ServD.ATTT.Prov.Reg
ContactPoints	ContactPoint (collection)	Collection of Contact points for this Provider.
ServiceSites	ServiceSiteProvider (collection)	Collection of Service Site Ids this Provider works at.
ReferralMethods	ReferralMethod (collection)	Collection of Referral Methods for the Provider.
Languages	ProviderLanguage (collection)	Collection of Languages spoken/understood by the Provider .

Returned By:

- Interface.RetrieveDetails.RetrieveProviderDetails (collection)

8.10.14　ProviderLanguage

Derived from: Common.ModeratedRecord

A Provider is able to deliver services in one or more languages. This entity records the capabilities of the Provider within the identified languages, and is useful for the referral processes.

Property	Data Type	Description
ProviderLanguageId	String (50)	The unique Identifier assigned by the ServD Directory.
LanguageSpokenCode	String (64) *(mandatory)*	A code from a predefined language list. **cts2:**ServD.LAN

Referenced By:

- DataModel.Provider.Languages (collection)
- Results.ProviderSearchItem.Languages (collection)

8.10.15　ProviderName

Derived from: Common.ModeratedRecord

A Provider may be known by one or more names distinguished by type and purpose. For example: married name, professional name, birth name, and so on. Also, any one of those names may be identified as the preferred name or the legal name.

NOTE: THE PREFERRED NAME IS AN INSTANCE OF THIS CLASS ON THE PROVIDER OBJECT, NOT A FLAG ON THIS ITEM

Property	Data Type	Description
ProviderNameId	String (50)	The unique reference identifying an attribute. This is the primary key for the attribute entity.
NameType	String (64)	The Name type covers ways in which the person can be known. **Example:** Professional, Married, Birth, Name by Repute, Alias **cts2:**ServD.PNT
NamePurpose	String (64)	The way in which this name record is intended to be used. **Example:** Mailing, Checks, Correspondence **cts2:**ServD.PNP
FamilyName	String (50) *(mandatory)*	Details the Providers's family name.
GivenNames	String (250)	Details the Provider's given name(s).
TitleCode	String (64)	The Title of the Provider. **Example:** Mr, Mrs, Dr **cts2:**ServD.TITLE

Property	Data Type	Description
IsLegalIndicator	Boolean	Set to true for the legal name, false for a non-legal name.

Referenced By:
- DataModel.Provider.PreferredName
- DataModel.Provider.OtherNames (collection)

8.10.16 ReferralMethod

Derived from: Common.ModeratedRecord

A Service Site may identify the referral methods that are appropriate to its circumstances.

Property	Data Type	Description
ReferralMethodId	String (50)	The unique Identifier assigned by the ServD Directory.
Code	String (64) *(mandatory)*	A code that identifies a referral method that is acceptable to this Site. **cts2:**ServD.RMC

Referenced By:
- DataModel.Provider.ReferralMethods (collection)
- DataModel.ServiceSite.ReferralMethods (collection)

8.10.17 ServiceSite

Derived from: Common.ModeratedRecord

The Service as operated at a Site.

Property	Data Type	Description
ServiceSiteId	String (50)	A unique Identifier that is allocated by the saving system. When adding a new Service Site, this is blank and will be allocated.
SiteId	String (50)	The Id of the Site that this Service Site is associated with.
ServiceCategory	String (64)	Identifies the broad category of service being performed or delivered. Selecting a Service Category then determines the list of relevant service types that can be selected in the Primary Service Type. **cts2:**ServD.SC
PrimaryServiceType	ServiceType	The specific type of service being delivered or performed.
ServiceName	String (500)	Further description of the service.

Property	Data Type	Description
Comment	String (2048)	Brief description of a Service Site or any specific issues not covered by the other attributes.
ExtraDetails	String (2048)	Extra details about the service that can't be placed in the other fields.
FreeProvisionCode	String (64)	The free provision code provides a link to the Free Provision reference entity to enable the selection of one free provision type. **cts2:**ServD.SSFPC
Eligibility	ConditionalIndicator Enum	Describes the eligibility conditions for the Service Site.
EligibilityNote	String (2048)	Describes the eligibility conditions for the service. The description of service eligibility should, in general, not exceed one or two paragraphs. It should be sufficient for a prospective consumer to determine if they are likely to be eligible or not. Where eligibility requirements and conditions are complex, it may simply be noted that an eligibility assessment is required. Where eligibility is determined by an outside source, such as an Act of Parliament, this should be noted, preferably with a reference to a commonly available copy of the source document such as a web page.
AppointmentRequired	ConditionalIndicator Enum	Indicates whether or not a prospective consumer will require an appointment for a particular service at a Site to be provided by the Organization. Indicates if an appointment is required for access to this service. If this flag is 'NotDefined,' then this flag is overridden by the Site's availability flag.
ImageURI	String	If there is an image associated with this Service Site, its URI can be included here.
AdditionalServiceTypes	ServiceType (collection)	Secondary Service types for this particular service.
AvailableTimes	Available (collection)	A Collection of times that the Service Site is available (overriding the Site values).
NotAvailableTimes	NotAvailable (collection)	A Collection of times that the Service Site is not available (overriding the Site values).
AvailabilityExceptions	String (1000)	A description of Site availability exceptions, e.g., public holiday availability. Succinctly describing all possible exceptions to normal Site availability as details in the Available Times and Not Available Times. (May contain HTML formatted text) *NOTE: THIS IS COPIED FROM THE SITE FOR CONSISTENCY*

Property	Data Type	Description
PublicKey	String	The public part of the 'keys' allocated to an Organization by an accredited body to support secure exchange of data over the internet. To be provided by the Organization, where available. The length of this string is undefined and should be of an un-restricted type. *NOTE: THIS IS A BASE64 ENCODED DIGITAL CERTIFICATE. THE SERVD FEDERATION'S PROFILE WILL SPECIFY THE TYPE OF CERTIFICATE TO BE USED AND ITS PURPOSE.*
ELSURL	String (2048)	Web service address of the ELS server's Lookup interface. **Example:** https://els.yourcompany.com/elslookup.svc
ProgramNames	String (256)	Program Names that can be used to categorize the service.
Identifiers	Identifier (collection)	Collection of Identifiers associated with this Service Site.
ContactPoints	ContactPoint (collection)	Collection of Contact points for this Site.
Characteristics	Attribute (collection)	Collection of Characteristics (attributes) for the Service Site. **cts2:**ServD.ATTT.SS
ReferralMethods	ReferralMethod (collection)	Collection of Referral Methods for the Service Site
Settings	ServiceSiteSetting (collection)	Collection of Settings for the Service Site.
TargetGroups	ServiceSiteTargetGroup (collection)	Collection of Target Groups for the Service Site (The target audience that this service is for).
CoverageAreas	CoverageArea (collection)	Collection of coverage areas for this Service Site.
CatchmentAreas	CatchmentArea (collection)	Collection of catchment areas for this Service Site.
Providers	ServiceSiteProvider (collection)	Collection of Provider Ids that work at this Service Site.
Specialties	Specialty (collection)	Collection of Specialties handled by the Service Site. **cts2:**ServD.SPCLTY.SS

Referenced By:

- DataModel.Site.ServiceSites (collection)

Returned By:

- Interface.RetrieveDetails.RetrieveServiceSiteDetails (collection)

8.10.18　　ServiceSiteProvider

Derived from: Common.ModeratedRecord

A Provider may work from zero, one or more service sites.

Property	Data Type	Description
ServiceSiteProviderId	String (50)	The unique Identifier assigned by the ServD Directory.
ServiceSiteId	String (50)	The Id of the Service Site that the provider works at.
ProviderId	String (50) *(mandatory)*	The Id of Provider that is working at this Service Site.
Identifiers	Identifier (collection)	Collection of Identifiers associated with this Service Site Provider. Provider Number (where a Provider has a different number for each place he works).

Referenced By:

- DataModel.Provider.ServiceSites (collection)
- DataModel.ServiceSite.Providers (collection)

8.10.19　　ServiceSiteSetting

Derived from: Common.ModeratedRecord

This entity describes the Service Site with respect to facilities arrangement. Is this a walk-in clinic, a corporate office environment, a surgical suite, and so on?

Property	Data Type	Description
ServiceSiteSettingId	String (50)	A unique Identifier that is allocated by the saving system. When adding a new record, this is blank and will be allocated
Code	String (64) *(mandatory)*	The value from the reference list of the Setting for the actual Service **cts2:**ServD.SSS

Referenced By:

- DataModel.ServiceSite.Settings (collection)

8.10.20 ServiceSiteTargetGroup

Derived from: Common.ModeratedRecord

Some service sites are directed to specific target groups: the Site might be on a reserve and available to First Nations peoples. A Site might be in existence to serve the HIV community, and so on. This entity allows a Site to be classified if required.

Property	Data Type	Description
ServiceSiteTargetGroupId	String (50)	A unique Identifier that is allocated by the saving system. When adding a new record, this is blank and will be allocated.
Code	String (64) *(mandatory)*	The value from the reference list of the Target Group for the actual Service. **cts2:**ServD.SSTG

Referenced By:

- DataModel.ServiceSite.TargetGroups (collection)

8.10.21 ServiceType

Derived from: Common.ModeratedRecord

A Keyword that is to be used by a specific Service Site.
It can be either a primary, or secondary keyword.

Property	Data Type	Description
ServiceTypeId	String (50)	A unique Identifier allocated by the saving system. When creating a new record, this is blank and will be allocated by the system.
Value	String (64) *(mandatory)*	The Code value of the Service Type. **cts2:**ServD.ST

Referenced By:

- DataModel.ServiceSite.PrimaryServiceType
- DataModel.ServiceSite.AdditionalServiceTypes (collection)

8.10.22 Site

Derived from: Common.ModeratedRecord

A Site is some place where service is provided. This may be a bricks-and-mortar location, a mobile laboratory, a temporary location, or any other relevant Site. In the HCP model this is often referred to as a Service Site. The term "service" has been dropped here to avoid confusion with the entity that contains service information.

Property	Data Type	Description
SiteId	String (50)	A unique Identifier that is allocated by the saving system. When adding a new Site, this is blank and will be allocated.
OrganizationId	String (50)	The Id of the Organization that this Site is associated with.
Description	String (250) *(mandatory)*	A textual description of the Service Site.
Comment	String (2048)	Brief description of what is offered/included at a Site, as a link to further information. (May contain HTML formatted text.)
ImageURI	String	If there is an image associated with this Site, its URI can be included here. This could be different to the Organizations Image, for example if a Vision Research wing had a specific custom logo, it would be included here.
AvailableTimes	Available (collection)	A Collection of times that the Service Site is available (overriding the Site values).
NotAvailableTimes	NotAvailable (collection)	A Collection of times that the Service Site is not available (overriding the Site values).
AvailabilityExceptions	String (1000)	A description of Site availability exceptions, e.g., public holiday availability. Succinctly describing all possible exceptions to normal Site availability as details in the Available Times and Not Available Times. (May contain HTML formatted text.)
LocationDescription	String (1000)	A text location description for the Site. A succinct description of the location of the Site. The description should include any additional information to help visitors access the Site that is not immediately obvious from the address information. (May contain HTML formatted text.) **Example:** Access to 120 Spencer St. is from Frances St. **Example:** Adjacent to tram stop 45 on line 70
MapReference	String (100)	Indicates the page and/or grid reference for the Site.

Property	Data Type	Description
MapReferenceSource	String (100)	A Description of the Source of the Map Reference information. **Example:** Melways *Note: This was not in the reference model, but is useful context to present with the Map Reference Field*
PublicKey	String	The public part of the 'keys' allocated to an Organization by an accredited body to support secure exchange of data over the internet. To be provided by the Organization, where available. The length of this string is undefined and should be of an un-restricted type. *Note: This is a base64 encoded digital certificate. The ServD Federation's Profile will specify the type of certificate to be used and its purpose.*
ELSURL	String (2048)	Web service address of the ELS server's Lookup interface. **Example:** https://els.yourcompany.com/elslookup.svc
PrimaryAddress	Address	Collection of addresses of this Site.
OtherAddresses	Address (collection)	Collection of addresses of this Site.
Identifiers	Identifier (collection)	Collection of Identifiers associated with this Site.
Attributes	Attribute (collection)	Collection of attributes for the Site. **cts2:**ServD.ATTT.Site
ContactPoints	ContactPoint (collection)	Collection of Contact points for this Site.
CatchmentAreas	CatchmentArea (collection)	Collection of catchment areas for this Organization.
Geographical Coordinates	SiteGeographicalCoordinate (collection)	Collection of locations that this Site can be found.
ServiceSites	ServiceSite (collection)	Collection of Services provided at this Site.

Referenced By:

- DataModel.Organization.Sites (collection)

Returned By:

- Interface.RetrieveDetails.RetrieveSiteDetails (collection)

8.10.23   SiteGeographicalCoordinate

Derived from: Common.ModeratedRecord

A Site may be described in many different ways in order to place the Site into a geographical context. For example: one method is to assign to each Site a longitude, latitude and altitude. Another method is to assign each Site within some larger context such as a bounded area which could be defined by postal codes, electoral districts, or some other means of dividing areas. Each method of description is represented by a different instance within this entity, allowing a Service Site to have zero, one or more geographical definitions.

Property	Data Type	Description
SiteGeographical CoordinateId	String (50)	A unique Identifier that is allocated by the saving system When adding a new Site geographical co-ordinate, this is blank and will be allocated.
SpatialReferenceID	String (64) *(mandatory)*	The Co-ordinate system being used (Spatial Reference ID - SRID). *NOTE: THE SPATIAL REFERENCE IDENTIFICATION SYSTEM IS DEFINED BY THE EUROPEAN PETROLEUM SURVEY GROUP (EPSG) STANDARD, WHICH IS A SET OF STANDARDS DEVELOPED FOR CARTOGRAPHY, SURVEYING, AND GEODETIC DATA STORAGE. THIS STANDARD IS OWNED BY THE OIL AND GAS PRODUCERS (OGP) SURVEYING AND POSITIONING COMMITTEE.* **Example:** WGS84 Other relevant information can be found at http://sharpgis.net/post/2007/05/Spatial-references2c-coordinate-systems2c-projections2c-datums2c-ellipsoids-e28093-confusing.aspx
Latitude	Decimal	The Latitude of the Site (Optional)
Longitude	Decimal	The Longitude of the Site (Optional)
Altitude	Decimal	The Altitude of the Site (Optional)
Granularity	Integer	The accuracy of the Latitude, Longitude and Altitude fields. Measures in meters.
ShapeFormat	String (64)	This format of the Boundary Shape field. (see http://en.wikipedia.org/wiki/GIS_file_formats) **Example:** WKT **Example:** WKB **cts2:**ServD.GBSHF
SiteShape	String	This is the shape definition of the boundary for this Site.
CatchmentShape	String	This is the shape definition of the "coverage area" for the Site.
IsSearchable Coordinate	Boolean	If true, this coordinate may be used in searches to find a Site. If false, then the value cannot be used in searches. *Note: This has been added to strengthen the privacy implications of location.*

Referenced By:

- DataModel.Site.GeographicalCoordinates (collection)

8.10.24   Specialty

Derived from: Common.ModeratedRecord

An Organization may provide services under zero, one or more specialties. An Organization that is a holding company may have no specialties, while those sites or services held by the holding company may have various. Note that it is not intended that Speciality is used as the basis for a service search. The specialties pertaining to each service offered are more likely to be the ones relevant to the consumer.

Note: This code type that is applicable to the code Value in this object can either be associated with the ServD.SPCLTY.Org or ServD.SPCLTY.SS types, depending on the parent of this record.

Property	Data Type	Description
SpecialtyId	String (50)	The repository unique reference number identifying this item. (This is assigned by the Directory when the item is added.)
Code	String (64) *(mandatory)*	The code of the specialty. *Note: This value set can either be associated with the ServD.ORGSPCLTY or ServD.STSPCLTY types.*

Referenced By:

  * DataModel.Organization.Specialties (collection)
  * DataModel.ServiceSite.Specialties (collection)

8.11 Conformance Profile Tests

8.11.1 ServD Core

	Conformance Points
	Conformance Points
SC1.1	A **ServD Core** must contain a component that implements the **Search** interface.
SC1.2	A **ServD Core** must contain a component that implements the **Retrieve Details** interface.
SC1.3	A **ServD Core** may implement the **Search** interface over an unsecured, unauthenticated connection. Secure or Private data should not be returned over the unsecured connection.
SC1.4	A **ServD Core** may implement the **Retrieve Details** interface over an unsecured, unauthenticated connection. Secure or Private data should not be returned over the unsecured connection.
SC1.5	A **ServD Core** may implement the **Search** interface over a secured, authenticated connection.
SC1.6	A **ServD Core** may implement the **Retrieve Details** interface over a secured, authenticated connection.
SC1.7	The **Search** and **Retrieve Details** interfaces may be distributed, but must be consistent.
SC1.8	Any changes to data made by an Author that are subject to moderation must not be returned in the results of the **Search** or **Retrieve Details** invocation until they have been approved or when the GetUnmoderatedDataWhereAvailable parameter is false
SC1.9	Any changes to data made by an Author that are subject to moderation must be returned in the results of the **Search** or **Retrieve Details** invocation for Author/Approvers of the data when the GetUnmoderatedDataWhereAvailable parameter is true. If the user is not an Author/Owner of the data, then un-moderated data must not be returned, and the last approved content must be returned.

	Conformance Points
SC1.10	All access to the Service using an authenticated connection must be audited.
SC1.11	Secure or Private data must NOT be returned to unauthenticated users.

8.11.2 Maintainable ServD Core

	Conformance Points
SC2.1	A **Maintainable ServD Core** must satisfy all of the requirements of a **ServD Core**.
SC2.2	A **Maintainable ServD Core** must contain a component that implements the **Maintenance** interface over a secured connection.
SC2.3	The authentication applied shall be for the Author or Approver Security Role, ensuring that only access to their authorized Organizations data is permitted.
SC2.4	All accesses to a **Maintainable ServD Core** must be audited.
SC2.5	Any changes to data made by an Author must be moderated.
SC2.6	Any changes to data made by an Approver shall NOT be moderated.
SC2.7	A **Maintainable ServD Core** may verify attributes against an external **ServD Verifier**.
SC2.8	All Attributes (see Information Model - Attribute) must have the **VerificationStatus** value of **NotVerifiable** if there is no external verification service configured for the attribute type. If there is a configured verification service, then the value will be one of; **Pending**, **Verified, NotValid** or Expired as returned by the external **ServD Verifier**.
SC2.9	A data item that is partially retrieved must NOT be updated into the ServD repository (identified by the RecordStatus property).

8.11.3 ServD Locator

	Conformance Points
SL1.1	A **ServD Locator** must implement the **Locator** interface.
SL1.2	A **ServD Locator** must implement the **Locator Maintenance** interface over a secure connection.
SL1.3	A **ServD Locator** may implement the Search interface
SL1.4	A **ServD Locator** must indicate in the status value a result of "Partial" if not all search interfaces were processed during a federated search.

8.11.4 ServD Verifier

	Conformance Points
SV1.1	A **ServD Verifier** must implement the **VerifyDetails** interface.
SV1.2	A **ServD Verifier** may return additional details of the registration.

8.11.5 Searching Application

	Conformance Points
SA1.1	A **Searching Application** must invoke the **Search** Interface to look up a list of entities.
SA1.2	A **Searching Application** must invoke the **Retrieve Details** interface at the network location specified in the results returned by the invocation of the **Search** interface, in order to retrieve more detailed information.
SA1.3	A **Searching Application** must use a **ServD Locator's Locator** interface to locate ServD Services exposing the **Search** Interface, or use the **Search** interface exposed by the ServD Locator if it implements it.
SA1.4	A **Searching Application** may connect to the **Search** interface over a secured connection.
SA1.5	A **Searching Application** may connect to the **Retrieve Details** interface over a secured connection. The **Searching Application** must connect to the network location identified in the results of the **Search** interface call.

8.11.6 Maintenance Application

	Conformance Points
SA2.1	A **Maintenance Application** must update information using the **Maintenance** interface.
SA2.2	A **Maintenance Application** may retrieve information using the **Secure Retrieve Details** interface.
SA2.3	A **Maintenance Application** must connect using either the **Author** or **Approver** Security Role.

9.   Platform Specific Model

The Platform Specific Model will cover how to implement the ServD specification as SOAP Web Services.

The Model file included with this submission includes a PSM for C#.

9.1   SOAP Web Services

The WSDL files are provided as associated files to this specification.

Informative Note: A REST architectural style could also utilize the same XSD data model that is included in the WSDL files for how the result information is returned, with the exception that the data is returned in a JSON format.

File	Description
ServD_Locator.wsdl	The WSDL definition for the ServD Locator.
ServD_Core.wsdl	The WSDL definition for the ServD Core, including the Maintainable ServD Core.
ServD_Verifier.wsdl	The WSDL definition for the ServD Verifier.

Annex A: Informative Content

A.1 Problems Addressed by the Specification

The continuum of care, by its nature, relies on the accurate and secure communication of health related information between health and human services practitioners. The human being is prone to complex conditions that require services by a variety of disciplines and organizational structure.

The use of a directory to assist in the discovery and co-ordination of service providers in traditional IT systems has been by way of a provider table or "address book" within the IT application. As health care organizations have multiple IT applications, there has been a proliferation of address books that each need to be separately updated and maintained.

There are three fundamental problem areas met by this specification:

- The proliferation of service provider address books across multiple applications within the health and community services setting.

- Limited abilities of existing applications to search by service discipline and location.

- The lack of a *source of truth* about practitioners, services and locations that can be accessed universally to enable trusted and secure interoperability between applications.

These aspects are discussed further below.

A.1.1 Source of Truth

Most applications supporting the delivery of care in health and community services setting require an "address book" or a table of providers that can be used for referrals and other forms of notification. As most health care providers have multiple applications, this leads to multiple address books, each of which needs to be maintained manually. The cost of multiple address books is not only the manual effort of replicating updates across multiple systems. There are also significant costs in information sent to the wrong practitioner, or the wrong address. For example, a lab test result not reaching the doctor may have dire consequences for the patient. The industry needs a single source of information, accessible by a standard set of interfaces that can be used for online enquiry or to refresh local address book entries. Due to the need to co-ordinate services across organizational and location boundaries, the scope of such a directory could be at a state of national level. From the provider perspective, they need only to update a single location with changes their information, for this information to then be reflected for all queries to that organization.

A.1.2 Service-Centric Model

Another related issue is that most address books are indexed by provider name or organization name. In many cases, a referring provider may need to perform a search based on service and or location, for example "retrieve a list physiotherapy services within 10kms of the workplace location of the patient". Recent years have seen the establishment of web sites that enable service related queries (for example, www.doctordirectory.com), however there is no standard for the representation of information, nor for APIs that enable information in these directories to be used by applications to refresh their address books, or to support exchange of information between applications.

In a health and community services setting, the discovery, referral, intake, transfer and coordination of care is governed by two key elements:

- A type of service to be provided (Service)

- A physical facility that provides the service (Site), that is accessible within the locality the patient/client.

A single *Organization* may provide multiple *Services* across multiple *Sites*.

The intersection of the *Service* to be provided at a *Site* is called the *Service Site*. In many cases, the object of a search, referral, or notification may be an individual *Provider* (such as a medical specialist). However, in all cases,

the individual *Provider* must have an association with at least one Service Site, even if it they are a single Provider in a private practice. This ensures that search by Service Site will include the discovery of a sole Provider practice in a search. The concepts of Organization, Site, Service Site, and Provider are illustrated below.
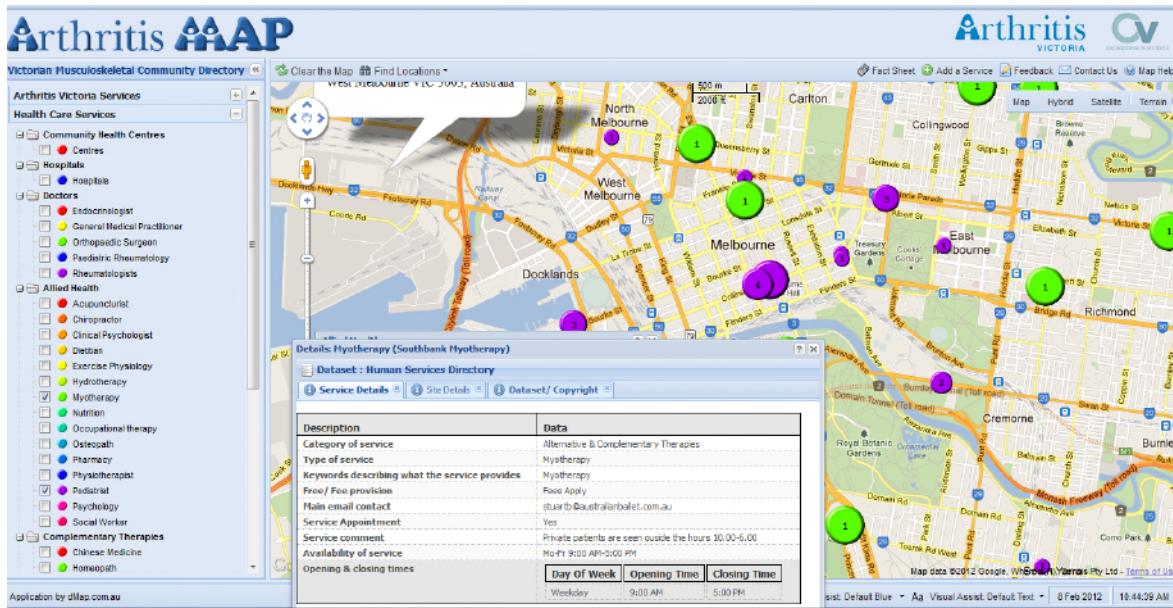
The realization of the concept of the service site is illustrated in work performed by Arthritis Australia, which has uses service site data from the Victorian Human Services Directory (HSD) to provide a publicly available discovery service imposed on a Google map layer. (see www.arthritismap.com.au).

A.1.3 Supporting Interoperability

Interoperability between health and community services applications across organizational boundaries requires information passing between firewalls in all cases to be encrypted, and in many cases, digitally signed. There is a challenge in maintaining access and concurrency to the public certificates for the thousands of health care providers and service locations that may exist within and across jurisdictional boundaries. The ability to have ready access to public certificates is a fundamental need of health care service interoperability.

Futhermore, a sending application needs information about the capabilities of the application it wishes to communicate with. For example, if a sending application creates a referral in HL7CDA format, it needs to know that the receiving application has the capability to consume and interpret an HL7CDA referral. This level of communication thus relies on a published set of standards that inform the capabilities of a vendor clinical application at site X to receive and consume information originating from another vendor's clinical application at site Y.

The ability to securely manage and disclose end point URL addresses, public certificates and transactional capabilities has been addressed by the ServD specification.

A.2 References to Prior Work (non OMG)

A.2.1 HL7 HCSPDIR

Most of functional capabilities of ServD have been previously specified in Service Functional Model Specification: Healthcare, Community Services and Provider Directory Version 2.0, Dec 2010. (HCSPDIR SFM).

The PIM describes minor changes to the terminology and identifies elements to the data model that have been added to the HCSPDIR data model.

The following HCSPD SFM functions have not been elaborated in the current version of ServD. This is because the requirement to manage resources and bookings is outside the scope of ServD release 1. There are no equivalent functions for these in ServD:

- Add a Resource

- Update a Resource

- Relate a Resource to a Location

- Disassociate a Resource from an Organization

- Add a Resource Allocation

- Remove a Resource Allocation

- Add a Resource Calendar

- Update a Resource Calendar

- Add a Provider Allocation

- Update a Provider Allocation

- Find an Available Resource within an Area

- Find all Locations that have a specified Resource

- Find all Resources at a Location

- Find all Resources associated with a Provider Location

Functions have been included in ServD that extend the HCSPDIR SFM to incorporate:

- Endpoint Locator Service (ELS)

- Federated search/indexed directory search.

A.2.2 Human Services Directory (HSD) and Connectingcare

The Human Services Directory (HSD) is an example implementation of a services directory that uses a similar structure to the information model of HCSPDIR. It was developed by the authors under contract to the Department of Health in Victoria, Australia in a project commencing in 2003.

It contained over 14,000 organizations covering 15,000 sites and 23,000 practitioners. Its initial implementation as an internet based public directory was expanded to include web based API access, so that it now supports B2B data access from telephone based nursing services, hospital systems and community health.

The use of HSD has since expanded to incorporate the jurisdictions of the Australian Capital Territory and the Northern Territory, and an ELS function was deployed during 2011. HSD and its ELS support inter agency messaging for referral and discharge summary.

For information on HSD, see www.humanservicesdirectory.vic.org.au.

Service directory instances based on the published HSD data model have also been deployed in other projects in Australia. Connectingcare (www.connectingcare.com) was created as a collaboration between community service providers and the Victorian government as a means of initiating secure electronic messaging between community and hospital. Connectingcare uses the HSD data model and provides a web portal to search for providers and sites, compose electronic documents and dispatch the secured documents via web services or secure email. The service directory content of Connectingcare has been migrated into the HSD so that users will be able to lookup HSD entries from within the Connectingcare portal using the web services API. Connectingcare will be retained as a secure messaging service using the HSD service directory functions and the ELS.

A.2.3 IHE Healthcare Provider Directory

The Healthcare Provider Directory (HPD) is documented as a Technical Framework Supplement for Integrating the Healthcare Organization (IHE). Revision 1.2 was published as a Trial Implementation on August 19 2011.

The HPD describes a publicly shared directory structure that supports queries against, and management of healthcare provider information. The HPD is designed to store information about Organizational Providers and Individual Providers.

There are many similarities between HPD and ServD. In the most part, the data definitions and functions of HPD are incorporated by equivalent concepts within ServD. There are however, some essential differences between ServD and HPD which are described below.

Provider directory vs. services directory

HPD has a central concept of an organizational hierarchy to which organizational units and individual providers may be associated via a relationship "member of". For example, a health service may consist of multiple departments and a provider may be associated with one or more clinic locations. A separate organization record would be allocated to each department.

In comparison, in the ServD model the organization represents the legal entity containing information relevant to its corporate authority such as legal name, address, and identifiers, and staff not associated with service delivery. The organization may operate across multiple sites; each site would correspond to a physical location or address. Multiple services may be provided within a site; thus a single health service location pertaining to an organization may conduct a variety of health related services. This is illustrated in Section A.1 above.

In providing a service directory, the practical deficiency with the HPD is that the corporate relationships does not necessarily map to service delivery. As a workaround, it is potentially possible that the HPD could mimic the concept of a Service Site by assigning an organizational record to each service site. However this creates a secondary problem in that a service site does not necessarily map to the organization's structure.

For example, in the example in Section A.1, the individual pathology laboratories may report to a Department of Pathology that resides at Corporate Head Office. Therefore, is the organization instance of Eastview Pathology a member of the Eastview site, the Head Office Pathology Department, or both? How is a consistent application of mapping then standardized across thousands of organizations within a jurisdiction, to ensure a consistent discovery process?

Although the alignment of a service site with a valid organization is mandatory, ServD is less concerned with downstream department reporting structures, and more concerned with the points of intake and locations of service delivery. The class structure is thus designed to emulate healthcare service co-ordination in the real world view.

Service Locations

The limitations of a provider/organization association were described in Issue HPD007 in the HPD supplement, page 7. In order to manage the situation where a provider works in a subset of locations in an organization,

> "*a provider address must be associated with the provider directly and cannot be derived from the address of the organization provider that this individual is a member of*".

Thus if multiple Providers work at the same location, HPD does not describe an easy way to list the Providers at that location, as there is no class to describe the location. Furthermore, the HPD provider class contains the attributes of multiple Specialties and multiple Addresses; however it does not associate each Specialty with each Address for a Provider. Thus if Dr. Smith is a gynecologist at Address A and an obstetrician at Address B, HPD will describe Dr Smith as a gynecologist and obstetrician, working at Address A and Address B, but we do not know which service he provides at each address. ServD overcomes these limitations through the use of Site to describe the location address and Service Site to describe the available services at the location. A Provider can be associated with one or more Service Sites, and therefore a user can easily find all Providers associated with a Site, and the Services each Provider delivers at each Site, and even the hours and days of the week that they attend the Site.

Additional Information

The Service Site class contains other properties essential for healthcare service co-ordination, such as:

- • Hours of operation
- • Contact details
- • Preferred contact method
- • (Geocodes) latitude and longitude
- • Reference to the ELS for the Service Site

Generally speaking, Service Site is a point of entry into an Organization. Within a single Site there may be multiple points of entry and hence multiple Service Sites.

The elaboration of the discovery process as a health service oriented search fundamentally differentiates the concept provider directory (search for an individual provider name or organization name) from service directory (search for a service).

HPD described the concept of a "yellow pages" search, which was illustrated in a use–case whereby a referring physician searches for an endocrinology specialist according to parameters of locality, language spoken and gender. The search is conducted based on properties attached to the individual provider, and would return a list of providers. ServD would also provide the option of returning a list of endocrinology providers, but alternatively a list of endocrinology service locations within the locality. In reality, the referral may be made to the service site, which in turn would allocate an individual provider to respond to the referral.

HPD assumes LDAP as the underlying protocol for interaction with the directory, which is expressed in DSMLv2. The use of LDAP is consistent with an organization/provider model; however the authors of ServD have concerns that LDAP would overly constrain the expression of ServD as a services oriented directory. At its highest level of abstraction ServD is platform independent. The platform dependent model proposed in this specification is based on web services and SOAP, however this does not preclude implementation of ServD using other protocols such as REST on the basis that the platform dependent implementation conformed to the normative requirements of the platform independent model.

Supporting public and restricted access

The natural progression of the service directory paradigm is the need for a public facing view of the directory. As an example, a health care consumer may wish to search for services within their locality e.g., "Find a drug and alcohol rehabilitation service close to where we live." Other information such as certificates and identifiers may only be available via restricted access. ServD thus incorporates the concept of a public search as well as a secure search.

There is also a requirement to ensure that a user who updates *any* record must be authorized to update that specific record. Page 28 of the HPD document states:

> *If there is any information, in part, or whole, as defined by the HPD profile schema or any extensions to the given schema, that is considered private or sensitive, then appropriate security and privacy procedures shall be considered to protect that information."*

In reality, a user authorized to update a record may be either the registered owner of the record (such as a site administrator), or a proxy that has been authorized to update the record on behalf of the owner. An authorized user may or may not be a member of the organization record being updated.

HPD specifies that updates are based on a receiving a feed from one or more provider information sources but does not propose a mechanism to restrict an information source from updating records owned by another information source within a shared database. This requirement is addressed by ServD.

Federation

Health care services co-ordination and delivery often transcends jurisdictional boundaries. Consumers living close to state borders need to be aware of services provided nearby. The maintenance of database records may be subject to different governance models for each jurisdiction. HPD recognizes the requirement of multiple directory instances but does not propose a mechanism. ServD incorporates the concept of a federated search, within the object model.

Endpoint Locator Service (ELS)

ServD describes the functions of the ELS which are not within the scope of HPD. More information on ELS is provided below.

A.2.4 Endpoint Location Service

The specification for an Endpoint Location Service (ELS) was published as technical report TR 5823—2010 by Standards Australia and can be downloaded at https://infostore.saiglobal.com/store/Details.aspx?productid=1391037. It is currently being assessed as an Australian standard under IT-014 e-health standards.

An Endpoint Location Service (ELS) is a simple directory of technical services enabling a producer of an electronic message (source) to transfer a message to an intended recipient (target), even if the producer has no prior knowledge of the method chosen by the recipient to handle the transfer.

The preconditions for interacting with the ELS is that the source is aware of a unique identifier associated with the target healthcare organization and the kind of message it wishes to transmit. The ELS responds to an information request by providing:

- The network address hosted by the target application (e.g., a URL)

- The public certificate of the target end point

- The capabilities of the target end point; i.e., the kinds of clinical documents that can be consumed by the target application.

- The protocol(s) used by a sending application to connect with a receiving application over a public network.

The ELS design supports multiple ELS instances which are accessible via secure web services. In the context of ServD, the URL of the ELS that relates to an end point is a property of the ServiceSite.

Much of the terminology adopted in the ELS specification TR 5823—2010 has references to other Australian standards work which was created for the local environment. The ServD specification has modified the ELS terminology to remove local references which are not relevant or ambiguous in an international standard. Importantly, the term Service Category was used in TR 5823-2010 to describe the capabilities of the target end point; this can easily be confused with the use of the term Service used in ServD to describe a healthcare service delivered by a provider. We have substituted the term Endpoint Capability for Service Category in ServD.

We have also removed reference to Australian specific terms such as HPIO (which is a form of site identifier), NASH (which is a certification authority), NEHTA (which is an organization established to foster adoption of e-Health in Australia) in preparing the ServD specification.

Notwithstanding the required terminology changes to support internationalization, the functional capabilities of an ELS have been incorporated within this specification, and the contribution of those towards the creators of TR 5823-2010 is respectfully acknowledged.

Use of an ELS operating in conjunction with ServD to facilitate application-to-application communications is described in section A.7 B2B Components / Interfaces.

A.3 Deployment Considerations

Some of the considerations relating to successful implementation of ServD are discussed below. These are not directly related to the information model, but have influenced some of the design considerations behind ServD.

A.3.1 Content management strategy

ServD implementation is a critical resource in sharing information about health and community service organization providers. As a common platform shared by potentially thousands of stakeholders, all contributing to the accuracy and currency of data, an implementation should take into consideration the governance arrangements and processes that ensure that information is accurate, up to date, and that security processes are in place. These should be directed at:

- Encouraging each site to update their own data, or to appoint a proxy service to update records on their behalf

- Implementing a security framework that ensures that unauthorized updates do not occur

- Creating an administrative function to over-see security and content management

The authors' experience is that there is often a proliferation of directories within a jurisdiction, and that stakeholders become annoyed at having to keep multiple directories up to date. The ServD objective is to provide a *single source of truth* for the jurisdiction, which is the first point of contact for changes, so that any changes can accessed to users of the service.

There may be an inclination to support multiple local directories updating the central directory, however this process may lead to lack of update control, unless there was a clearly defined responsibility for each local directory to maintain data for its own constituents. Even then, it may be desirable to have a central administration body vetting changes and implementing a feedback process to the originating site/provider to validate the changes.

In the situation where a B2B application is providing updates on behalf of multiple organizations it may be desirable to accept the data subject to moderation, but flag this data as provisional, pending verification at the source by the central administration body.

For these reasons, the ideal mode of update would be that each site and provider be responsible for self-authorship. The motivation for them to take control is the understanding that the changes made into ServD by them would be available to others via public search, or via applications using the data within the ServD repository.

A.3.2 Relationship between ServD and credentialing authorities

The scope of ServD is to provide information about organizations, providers and the services they provide in a common repository.

ServD is intended to hold unique identifiers for providers and the currency of their credentials, but it is not intended to be the primary system used by the organizations responsible for maintaining the registration records and credentials of providers. It is envisaged that medical providers will apply for registration with an authority, and that authority will use its own specialist database to maintain those records. An example of an authority would be an organization associated with the Federation of State Medical Boards in the US, which have the responsibility of verifying the credentials of practitioners in medical and related disciplines. It is envisaged that the eligibility of a provider to practice will be updated in ServD based on input from the credentialing authorities, expressed as an identifier with effective dates. On the other hand it is unlikely that a credentialing authority would be updating their own system with the associations between providers and the service sites; rather this information could be natively managed within ServD.

Similarly, ServD and the ELS can be used as a repository for public certificates for providers, organizations and sites, but the responsibility for authorizing certificates may be vested in a registration authority (RA) specifically set up for this purpose. A provider or representative of a site would apply to the RA for a certificate, and once the certificate had been issued by the certification authority (CA), would load the certificate into ServD and/or the ELS. The CA would be responsible for notifying ServD when a certificate had been revoked.

A.4 Concepts of Federation

A.4.1 Need for Federation

The Service Directory should be viewed as a national resource. The co-ordination of service delivery to healthcare consumers needs to take into consideration the nature of people to travel, the branch networks of organizations that transit across state boundaries, and the needs of consumers living near state, province and/or territory borders to access cross-border services.

Notwithstanding the needs for a national Service Directory, the governance of data may be subject to control within a more granular, jurisdictional basis. A Jurisdiction may be responsible for Service Sites within agreed geographical boundaries, specialities or other agreed perameters (Coverage Areas).

An expedient alternative to a monolithic national database is to have a network of jurisdictional ServD instances that are autonomously controlled and linked by a common index. The ServD specification attempts to address this objective.

A.4.2 RLUS

The authors reviewed the applicability of the OMG standard *Retrieve Locate and Update Service* (RLUS) as a basis for implementing federation. RLUS was developed to provide a service layer manage patient/client data residing across multiple databases, constituting millions of individual records.

Without making any comment on the suitability of RLUS for its intended purpose, the use of RLUS to maintain a relatively simple list of Coverage Areas and ServD instances was deemed to be inappropriate for ServD purpose.

Refer to http://wiki.hl7.org/index.php?title=Product_RLUS_SFM

A.5 Service Ontology

One of the challenges of implementing a service directory is the definition and management of lookup data, and of reconciling any implementation against a variety of conflicting standards.

There is no agreed international standard, for example, for the classification of *Service* categories. Rather than attempt to define the use of a particular standard as a mandatory compliance point for ServD, it is proposed that implementations support the ability to relate multiple value sets to each other. This becomes even more relevant where multiple ServD instances are interoperating, perhaps across jurisdictions.

The user needs to have the confidence that, if they enter a single set of search criteria, that the search criteria will be meaningful and interpreted in a common way across a set of federated or somehow linked set of directories.

Another situation arises where a ServD instance might be formed by the consolidation of data across multiple directories. Rather than go through a process of normalizing the data, the administrator could create a mapping between the different value sets using CTS2 so that both sets of values can remain valid.

A.6 Use Cases

A.6.1 Search and Retrieve Details

Scenario

Mary Brown is a single mother with a teenage daughter who has been refusing to go to school and appearing depressed. She wants to find a counseling service for adolescents. She does not have a car and would prefer a location of the service close to public transport.

Flow of events

1. Mary enters a free text search in a search portal, requesting adolescent counseling services.

2. The search engine includes summary of Service data in the results.

3. Mary looks at each item in the returned list and requests the portal to display the search results on a map. Upon selecting each entry, the portal retrieves details about the organization location and the Services provided from the Service Directory. Mary selects an organization that meets her Service criteria and is close to the train station.

4. Mary telephones the counseling service to arrange an appointment.

A.6.2 Secure Search & Retrieve Details

Scenario

Sally Jones attends hospital for an appendectomy. On admission, Sally informs the intake worker of the name and clinic location of her GP. Upon discharge, the hospital registrar wishes to send a discharge summary to her GP with results of her operation.

Precondition

Prior to Sally's attendance, the hospital application has retrieved details of all GPs within the hospital's area from the Services Directory, and loaded this information into the application's provider directory.

Flow of events

1. On admission, the intake worker looks up the application's provider directory and attaches the GP and clinic reference to Sally's admission record.

2. On discharge, the registrar creates the discharge summary in the hospital application, which includes the identifiers of the GP and clinic.

3. The hospital application retrieves the certificates, URL address and capabilities of the clinic's application from the Services Directory.

4. The hospital application creates the discharge summary in a format that can be consumed by the clinic's application.

5. The hospital application encrypts the discharge summary and sends it electronically to the clinic.

A.6.3 Federated Search and Retrieve Details

Scenario

John Simpson has a lump in the neck. His specialist would like to refer him to a radiology service for x-rays, however John is due to be interstate next week for work. The doctor would like to find a radiology service close to where John will be staying whilst away.

Flow of events

1. The doctor performs a search in his clinical application.

2. The clinical application looks up the Service Directory, requesting a Service within a 5km radius of John's hotel.

3. The Services Directory recognizes that the location specified is managed by another physical instance of the Services Directory that holds entries for the other state.

4. The search request is performed on the remote Services Directory.

5. The remote Services Directory returns the results of the search.

6. The doctor requests address and contact details about the interstate service.

7. The clinical application retrieves the contact details and electronic information required to construct an electronic referral, from the remote Services Directory.

8. The doctor requests his clinical application to create an electronic referral and send it to the radiology service.

9. The doctor provides a printout of the referral to John.

A.6.4 Maintenance – Author & Approver

Scenario

Harry Hubert is admitted to hospital. On admission, the intake worker looks up the directory and discovers that the address of the referring clinic provided by Harry is different from the hospital provider directory. Upon calling the clinic, the intake worker verifies the change of address. The intake worker would like to change the clinic address on their hospital application and ensure that the change is also reflected in the Services Directory.

Flow of Events

1. The intake worker requests a change of address to the hospital application.

2. The hospital application calls the Services Directory and authenticates itself as an Author.

3. The hospital application requests the Services Directory to update the clinic address details.

4. The Services Directory provisionally updates the clinic address, and returns an acknowledgement to the hospital application that the address has been provisionally updated.

5. An employee of the administrator of the Services Directory (the Approver) is advised that the new clinic address requires moderation.

6. The Approver telephones the clinic to verify the change of address.

7. The Approver updates the clinic address change in the Services Directory from provisional to confirmed

8. The Services Directory sends an email to the clinic advising that their record has been changed.

9. When the hospital application requests a directory update from the Services Directory, the clinic address has been changed from provisional to confirmed.

A.7 B2B Components / Interfaces

The **B2B** Components and Interfaces defined in this standard are not a part of the **ServD** standard. They are included to show how they interact with the **ServD** specification.

A.7.1 B2B Component Model

The B2B Component Model describes how B2B Applications and Services utilize the ServD Services to publish and discover the location of B2B Services, and the capabilities that they expose.
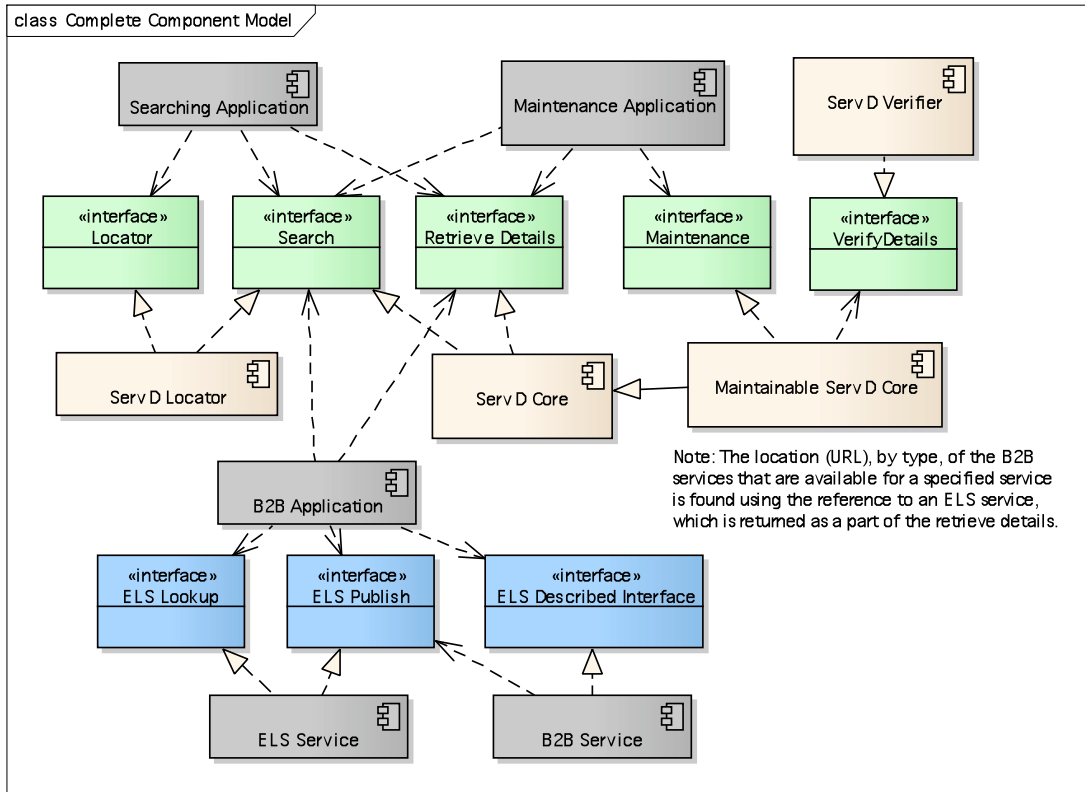
In order for a **B2B Application** to directly communicate with a **B2B Service**, it uses the **Search** interface to locate a *Service*, it then uses the **Secure Retrieve Details** interface to retrieve the network location of the relevant ELS instance. It then calls the **ELS Lookup** Interface to discover the network location of a compatible **B2B Service.**

A B2B Service registers its capabilities through the ELS Publish interface; this includes the mechanism it uses to communicate with the other B2B applications (ELS Described Interface), and also the message payload formats it is able to consume.

ELS Lookup

The **ELS Lookup** interface provides a capacity for **B2B Services** and **B2B Applications** to find other **B2B Services** capabilities and locations (URI).

Operations

Name	Columns	Notes
listInteractions	target, interface name, capability list	The **listInteractions** operation returns the set of capabilities that the requested service is able to support, and their network locations. These are interactionRecords.
validateInteraction	interactionRecord	The **validateInteraction** operation checks if an interaction has expired. If an interaction is determined to no longer be valid, then the caller needs to call the **listInteractions** operation to get the appropriate valid interactions.

ELS Publish

The **ELS Publish** interface permits the adding (and removing) of interaction records that describe the capabilities and locations of services provided by **B2B Services**.
Operations

Name	Columns	Notes
addInteraction	interactionRecord	The **addInteraction** operation is used by a **B2B Service** or **B2B Application** to register its capability, location and **ELS Described Interface** that the B2B Service provides..
removeInteractioin	interactionRecord	The **removeInteraction** operation removes an interaction previously registered from the **ELS Service**.

ELS Described Interface

An ELS Described interface enables B2B applications and services to communicate with each other directly.

The definition of the ELS Described Interface is not specified by the ServD, and could be any interface, provided that all using the interface (B2B Services and B2B Applications) agree to its definition and Name. In the Standards Australia ELS specifications, this is referred to as the ServiceInstance.

The name of this interface, and the network locations (URIs) are contained by interaction records stored in the ELS Service.

An example of an ELS Described interface is the Secure Message Delivery (SMD) Specification ATS 5822-2010 http://infostore.saiglobal.com/store/Details.aspx?ProductID=1391035

The ELS Interaction record also contains a capability value on the interaction record that is used in conjunction with the name for cases where the interface name is not sufficient to indicate that the caller can use it. E.g. if the interface supports the sending of multiple message types using the same interface, then the capability value is used to indicate what message types it supports.

Informative Note: When an ELS Service is configured (preferably when it starts) the service should call the ELS Publish interface to register its network location and all its capabilities (interactions).

ELS Service

An **ELS Service** is a service that provides registration of capability and location services for **B2B Services**.

From the perspective of this ServD specification, it is proposed to use the Standards Australia Technical Report TR 5823-2010 Endpoint Location Service.

Conformance Points
An **ELS Service** must implement the **ELS Lookup** interface.
An **ELS Service** must implement the **ELS Publish** interface.

B2B Service

A **B2B Service** is a service that is registered with the **ELS Service** (and subsequently in the **ServD Core**) which exposes a defined interface (**ELS Described Interface**) for **B2B Applications** to consume.

This registration is performed by the B2B Service adding interaction records.

Registration within the **B2B Service** ensures that the capabilities of the endpoint are kept up to date centrally when the configuration, capabilities (or version) of the service changes.

Conformance Points
A **B2B Service** must implement the **ELS Described Interface** that it registers with the **ELS Service**.

Conformance Points
A **B2B Service** may call the **ELS Publish** interface to register its endpoint configuration/capability details.

B2BApplication

A **B2B Application** is an application that is used to electronically send information from one Business (or Application) to another.

Conformance Points
A **B2B Application** may call the **ELS Lookup** interface to locate a **B2B Service** to interface to.
A **B2B Application** may call the **ELS Publish** interface to register **B2B Services** endpoint configuration/capability details.
A B2B Application may call the ELS Described Interface exposed by a B2B Service.
A **B2B Application** may call the **Search** interface to connect to search for services.
A **B2B Application** may call the **Retrieve Details** interface to retrieve more detailed information for information returned by a call to the **Search** Interface.
The **B2B Application** must connect to the **ServD Core** identified in the results of the **Search** interface call, and not just the same service that was providing the **Search** interface.
A **B2B Application** must connect to the **Secure Retrieve Details** interface over a secured connection.
The **B2B Application** must connect to the Secure **ServD Core** identified in the results of the **Search** or **Secure Search** interface call, and not just the same service that was providing the **Search** or **Secure Search** interface.
The **B2B Application** may read the ELS information from the **Secure Retrieve Details** to determine the capabilities and location (URI) of the service that it desires to connect to, and that implements the **ELS Described Interface** that it requests.
A **B2B Application** must connect to the **Secure Search** interface over a secured connection.

A.7.2 Use Cases

Electronic Messaging Scenario

ABC Home Care is a leading organization delivering services to the aged and disabled in their homes. In order to improve service delivery they have developed a dependency assessment that forms the basis of their care planning. ABC participate in a number of shared care arrangements and would like to ensure that their dependency assessment can be shared with the client's GP and other organizations that deliver care into the client's home.

ABC approach their software vendor SoftwareXL to develop the Dependency Assessment. Software XL suggest that ABC create a collaboration of service providers to jointly design the assessment.

ABC perform a non secure lookup on the services directory to find a list of organizations that deliver aged and disability care in the home with the purpose of setting up a collaboration. ABC also contact their government funding body and industry association to see if they would participate in the development.

The assessment is designed by the collaboration and specified by SoftwareXL.

The specification submitted to the national standards authority for review. Following standards ratification, the assessment specification is published for other vendors to incorporate in their software.

The specification includes a template, data definitions for the fields, and the business rules for processing the data items. It also includes the information required to populate the ELS that describe the dependency assessment.

Software XL develops the dependency assessment and incorporates it into Release 2 of the software

When Release 2 is installed at ABC Home Care's site, the software issues an electronic notification to the ELS to register the dependency assessment as a capability of ABC Home Care and adds a copy of ABC Home Care's public certificate to the ELS entry.

Another organization, Home Helpers has also implemented the dependency assessment. They wish to refer to ABC Home Care and attach the client's dependency assessment to the referral.

The Home Helpers co-ordinator creates the referral and attaches the dependency assessment. The Home Helpers application locates the ELS entry for ABC Home Care, and retrieves the capabilities, public certificate and electronic address for ABC Home Care. The Home Helpers application confirms that ABC Home Care is capable of processing the Dependency Assessment, and thus encrypts the assessment with ABC's public key, and sends the referral with the assessment attached.

B2B Application Calling a B2B Service Operation
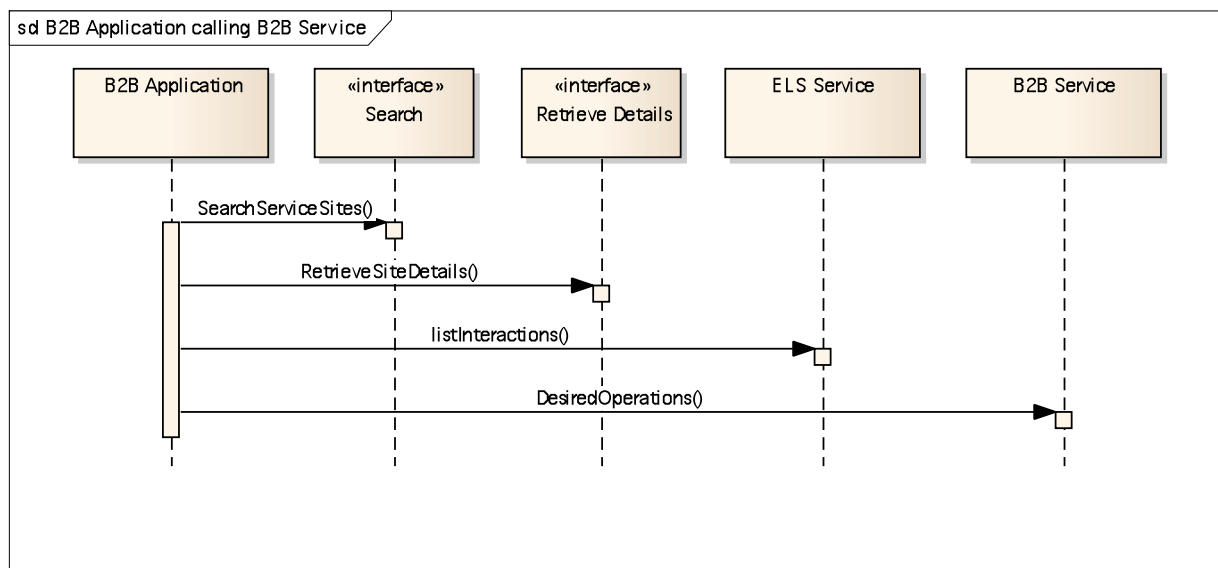
B2B Service Registering services/Operations of a B2B Service

sd B2B Service Registering services of a B2B Service

| B2B Service | «interface» Maintenance | ELS Service |

ServiceSiteAdd()

addInteraction()