**caBIG®**

**Specimen Identifier Management Service**

**0.9**

**12/14/2010**

|  |  |
| --- | --- |
| Authors | *Carlos Perez* |
| Editor |  |
| Reviewers |  |
| Architecture Inception Team |  |

**DOCUMENT CHANGE HISTORY**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version Number** | **Date** | **Contributor** | **Description** |
| **0.1** | **11/12/2010** | **Carlos Perez** | **Initial Document** |
| **0.2** | **12/08/2010** | **Carlos Perez** | **Updated with WSDL and XMLSchema** |
| **0.3** | **12/14/2010** | **Carlos Perez** | **Cleanup** |
| **0.9** | **12/14/2010** | **Carlos** | **Sent out for internal review** |

**Table of Contents**

[1 Executive Summary: 3](#_Toc280104873)

[1.1 Service Description and Purpose 3](#_Toc280104874)

[1.2 Scope 3](#_Toc280104875)

[1.3 Referenced Standards 3](#_Toc280104876)

[2 Conformance to Platform Independent Model 3](#_Toc280104877)

[2.1 Conformance Profile 3](#_Toc280104878)

[2.2 Dynamic Interactions 3](#_Toc280104879)

[3 Platform Specific Model 3](#_Toc280104880)

[3.1 Overview 3](#_Toc280104881)

[3.2 Domain Model 3](#_Toc280104882)

[3.3 Technology Stack 3](#_Toc280104883)

[3.4 Assumptions and Dependencies 3](#_Toc280104884)

[3.5 Service Interface 3](#_Toc280104885)

[3.5.1 Interface Model 3](#_Toc280104886)

[3.5.1 WSDL Messages and Port Types 3](#_Toc280104887)

[3.5.2 Operations Details for NamingAuthority Interface 3](#_Toc280104888)

[3.5.3 Operations Details for SpecimenIdentifierAuthority Interface 3](#_Toc280104889)

[3.6 Message Information Model 3](#_Toc280104890)

[3.6.1 Information Model 3](#_Toc280104891)

[3.7 Service Interactions 3](#_Toc280104892)

[3.7.1 Interaction Details 3](#_Toc280104893)

[3.8 Implementation Considerations 3](#_Toc280104894)

[3.8.1 Security 3](#_Toc280104895)

[3.8.2 Auditing 3](#_Toc280104896)

[3.8.3 Privacy 3](#_Toc280104897)

[3.8.4 Error Handling 3](#_Toc280104898)

[3.1 Deployment Considerations 3](#_Toc280104899)

[3.1.1 Deployment Modes 3](#_Toc280104900)

[3.1.2 Performance 3](#_Toc280104901)

[3.1.3 Scalability 3](#_Toc280104902)

[3.1.4 Discovery 3](#_Toc280104903)

[3.1.5 Uptime 3](#_Toc280104904)

[3.1.6 Failover 3](#_Toc280104905)

[3.2 Constraints 3](#_Toc280104906)

[3.3 Known Issues 3](#_Toc280104907)

[4 Recommendations for Conformance and Compliance 3](#_Toc280104908)

[4.1 Conformance Assertions 3](#_Toc280104909)

[5 Appendix B – Relevant Standards 3](#_Toc280104910)

[6 Appendix C – Glossary 3](#_Toc280104911)

[7 Appendix B - Glossary Error! Bookmark not defined.](#_Toc280104912)

# 

# Executive Summary:

## Service Description and Purpose

The functionality provided by Specimen Identifier Management service involves the management of globally unique identifiers for biospecimens. The biospecimen identifier is essentially a globally unique name that never expires. Its purpose is to support the exchange of biospecimen information across multiple biospecimen management systems.

## Scope

The scope of this document is limited to the specification of the Specimen Identifier Management service as a web service using the WSDL 1.1 and XML Schema 1.0 standards. The service will focus on the Specimen Identifier Management functional profiles providing operations to create and update a specimen identifier.

## Referenced Standards

The Specimen Identifier Management service conforms with existing web standards and protocols, providing a natural approach to identifier adoption and resolution.

|  |  |
| --- | --- |
| **Standards** | **Description** |
| Open Provenance Model | The Open Provenance Model is a model of provenance that is designed to allow provenance information to be exchanged between systems. |
| PURL | PURLs (**P**ersistent **U**niform **R**esource **L**ocators) are Web addresses that act as permanent identifiers in the face of a dynamic and changing Web infrastructure. |
| UUID | Universal Unique Identifier (UUID) DCE Version 3 and 5. |
| WSDL 1.1 | Web Services Description Language (WSDL), version 1.1  <http://www.w3.org/TR/wsdl> |
| WSRF-RP 1.2 | Web Service Resource Properties, version 1.2 draft 04  <http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ResourceProperties-1.2-draft-04.pdf> |
| WS-SG 1.2 | Web Services Service Groups, version 1.2 draft 02  <http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ServiceGroup-1.2-draft-02.pdf> |
| WSRF-BF 1.2 | Web Service Base Faults, version 1.2 draft 02  http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-BaseFaults-1.2-draft-02.pdf |
| WS-Addressing 03/2004 | Web Services Address, version draft 03/2004  http://www.w3.org/Submission/ws-addressing/ |
| XML 1.0 | eXtensible Markup Language, version 1.0  http://www.w3.org/TR/REC-xml/ |

# Conformance to Platform Independent Model

|  |  |  |
| --- | --- | --- |
| **Conceptual Functional Service Specification Name** | **Conceptual Functional Service Specification Version** | **Description & Link to the Conceptual Functional Service Specification** |
| CS13: Specimen Indentifier Management Service | 1.0.0 |  |

## Conformance Profile

|  |  |
| --- | --- |
| **Conformance Profile No.** | SIDM-CP1 |
| **Conformance Profile Name** | Specimen Identifier Resolution Conformance Profile |
| **Functional Profiles** | |  |  | | --- | --- | | **Functional Profile No.** | **Functional Profile Name** | | SIDM-FP1 | GSID Resolution | |
| **Semantic Profiles** | |  |  | | --- | --- | | **Semantic Profile No.** | **Semantic Profile Name** | | SIDM-SP1 | GSID v1.0 Resolution | |
| **Conformance Profile No.** | SIDM-CP2 |
| **Conformance Profile Name** | Specimen Identifier Edit Profile |
| **Functional Profiles** | |  |  | | --- | --- | | **Functional Profile No.** | **Functional Profile Name** | | SIDM-FP2 | GSID Edit | |
| **Semantic Profiles** | |  |  | | --- | --- | | **Semantic Profile No.** | **Semantic Profile Name** | | SIDM-SP2 | GSID v1.0 Edit | |

## Dynamic Interactions

All interactions between client applications and the service are standard SOAP/HTTPS.

# Platform Specific Model

## Overview

This section describes the service payload model and the service interface contract in detail. Schemas for the service message payloads are included. The service interface is described using WSDL snippets and the complete WSDL is in the appendix. This section also provides guidelines for implementing this service.

## Domain Model



The diagram above depicts the domain model maintained by the GSID service. The domain model involves two classes. The first is the GSID class that has as an attribute a unique identifier and association with other GSID objects in the parent/child relationship. The second class is the Site class that has a many to one relationship to a GSID class. The Site contains information that is used to discover further information about a biospecimen that the GSID refers to.

## Technology Stack

|  |  |
| --- | --- |
| **Technology** | **Affects** |
| Globus Toolkit 4.0.3 | The Globus Toolkit is used to provide client libraries to access this WSRF complaint grid service |
| caCORE SDK 4.3 | The service will be generated from its information model by the caCORE SDK. |
| Globus Toolkit 4.0.3 | GTK is used to develop this WSRF complaint grid service |
| JBoss 5.5.27 | JBoss application server is used to deploy this service |
| Java JDK 1.6 | Java Development Kit 6.0 is required |

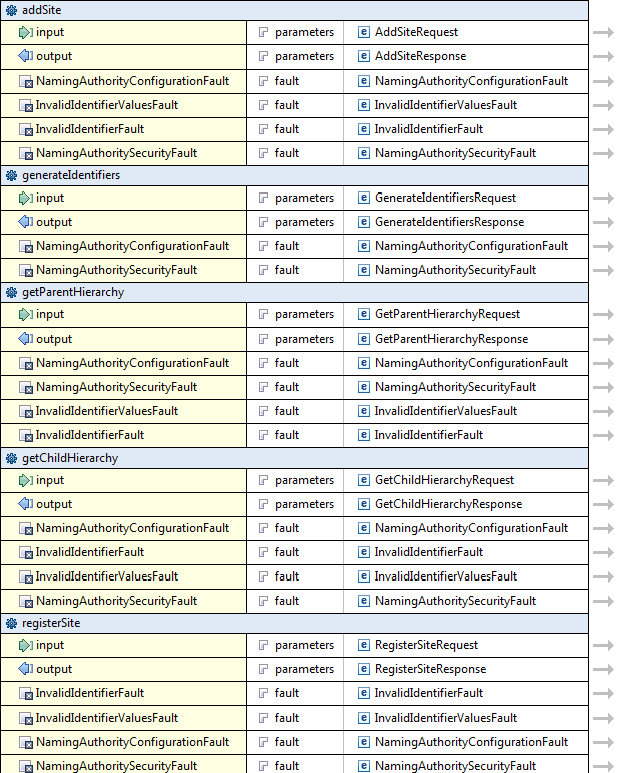
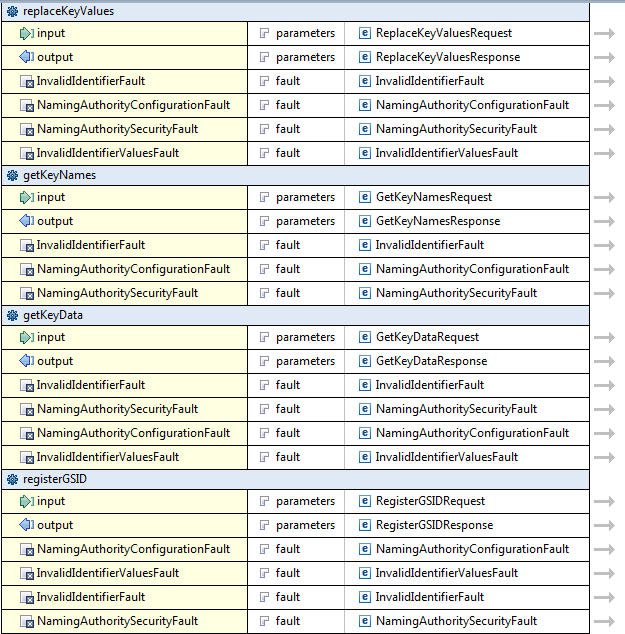
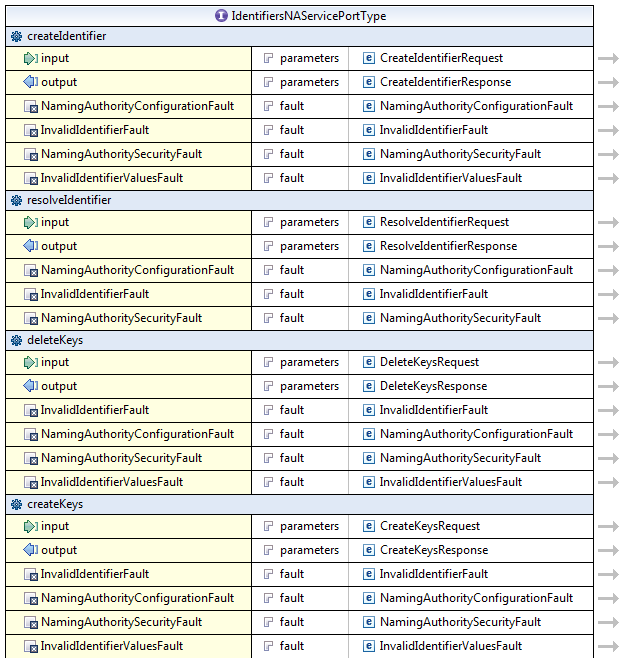
## Assumptions and Dependencies

|  |  |
| --- | --- |
| **Assumptions** | **Affects** |
| caGrid1.4 | This service is deployed in the caGrid 1.4 environment. |

|  |  |
| --- | --- |
| **Dependency** | **Description** |
| caGrid Production Environment | Service relies on caGrid Production environment for advertisement, discovery, and trust fabric synchronization |
| caGrid Index Service Server | The Grid Service relies on caGrid Index Service for the service registration and discovery |

## Service Interface

### Interface Model



### WSDL Messages and Port Types

The service shall conform to the messages and port types defined in the WSDL located at the following URL:

|  |  |  |
| --- | --- | --- |
| **Implemented Interface No.** | **Supported Interface Name** | **Link** |
| SIDM-INF1 | SpecimenIdentifierAuthority | IdentifiersNAService.wsdl |

### Operations Details for NamingAuthority Interface

The following sections describe service operations in detail.

#### resolveIdentifier

|  |  |
| --- | --- |
| **Description** | This operation returns the metadata associated with the given identifier. |
| **Pre-Conditions** | The identifier must exist and the requesting user must be authorized to resolve it. |
| **Security Controls** | A user can read a key from an identifier if any one of the below conditions are met:    (a) User is identifier's administrator  (b) User is listed as reader by the key’s policy identifier  (c) Key has no policy identifier and user is listed as reader or the identifier  (d) Key has no policy identifier and identifier has no security settings    Only the keys that the user is authorized to read are returned.  A security exception is thrown if the user can’t read any of the keys associated with the identifier. |
| **Inputs** | Identifier (Required) |
| **Outputs** | Identifier Metadata (keys and values) |
| **Post-Conditions** |  |
| **Exception Conditions** | InvalidIdentifierException  NamingAuthorityConfigurationException  NamingAuthoritySecurityException |
| **Notes** |  |

#### getKeyNames

|  |  |
| --- | --- |
| **Description** | This operation returns the metadata associated with the given identifier. |
| **Pre-Conditions** | The identifier must exist and the requesting user must be authorized to resolve it. |
| **Security Controls** | A user can read a key from an identifier if any one of the below conditions are met:  (a) User is identifier's administrator  (b) User is listed as reader by the key’s policy identifier  (c) Key has no policy identifier and user is listed as reader or the identifier  (d) Key has no policy identifier and identifier has no security settings    Only the keys that the user is authorized to read are returned.  A security exception is thrown if the user can’t read any of the keys associated with the identifier. |
| **Inputs** | Identifier (Required) |
| **Outputs** | Identifier Metadata (keys and values) |
| **Post-Conditions** |  |
| **Exception Conditions** | InvalidIdentifierException  NamingAuthorityConfigurationException  NamingAuthoritySecurityException |
| **Notes** |  |

#### getKeyData

|  |  |
| --- | --- |
| **Description** | This operation returns the metadata associated with the given identifier. |
| **Pre-Conditions** | The identifier must exist and the requesting user must be authorized to resolve it. |
| **Security Controls** | A user can read a key from an identifier if any one of the below conditions are met:    (a) User is identifier's administrator  (b) User is listed as reader by the key’s policy identifier  (c) Key has no policy identifier and user is listed as reader or the identifier  (d) Key has no policy identifier and identifier has no security settings    Only the keys that the user is authorized to read are returned.  A security exception is thrown if the user can’t read any of the keys associated with the identifier. |
| **Inputs** | Identifier (Required) |
| **Outputs** | Identifier Metadata (keys and values) |
| **Post-Conditions** |  |
| **Exception Conditions** | InvalidIdentifierException  NamingAuthorityConfigurationException  NamingAuthoritySecurityException |
| **Notes** |  |

### 

### Operations Details for SpecimenIdentifierAuthority Interface

#### registerGSID

Register a new Specimen Identifier.

|  |  |
| --- | --- |
| **Behavior Description** | This operation registers a Specimen Identifier. |
| **Pre-Conditions** |  |
| **Security Conditions** | A user can create identifiers if any one of the below  conditions are met:  (a) The user has been given explicit authorization to register identifiers. |
| **Inputs** | * String suggestedIdentifer * String[] parentIdentifiers |
| **Outputs** | Identifier |
| **Post-Conditions** |  |
| **Exception Conditions** | NamingAuthorityConfigurationException  InvalidIdentifierValuesException  InvalidIdentifierException  NamingAuthoritySecurityException |
| **Additional Implementation Details** |  |
| **Notes** |  |

#### addSite

Add the user as a Site that has information on the GSID.

|  |  |
| --- | --- |
| **Behavior Description** | Add the current user as an site that has information about the input specimen identifier. |
| **Pre-Conditions** |  |
| **Security Conditions** | A user can create identifiers if any one of the below  conditions are met:  (a) The user has been given explicit authorization to register identifiers. |
| **Inputs** | * String identifer |
| **Outputs** | Identifier |
| **Post-Conditions** |  |
| **Exception Conditions** | NamingAuthorityConfigurationException  InvalidIdentifierValuesException  InvalidIdentifierException  NamingAuthoritySecurityException |
| **Additional Implementation Details** |  |
| **Notes** |  |

#### generateIdentifiers

Create a batch of Identifiers.

|  |  |
| --- | --- |
| **Behavior Description** | Generates a batch of valid UUIDs. |
| **Pre-Conditions** | None |
| **Security Pre-Conditions** | A user can create identifiers if any one of the below  conditions are met:  (a) The user has been given explicit authorization to register identifiers. |
| **Inputs** | * int – Number of UUIDs to generate. |
| **Outputs** | String[] |
| **Post-Conditions** | None |
| **Alternate Conditions** | None |
| **Exception Conditions** | NamingAuthorityConfigurationException  NamingAuthoritySecurityException |
| **Additional Details** | None |
| **Notes** | None |

#### validateIdentifier

Validate Identifier.

|  |  |
| --- | --- |
| **Behavior Description** | Validates if the identifier can be registered as unique by the service. |
| **Pre-Conditions** | None |
| **Security Pre-Conditions** | A user can create identifiers if any one of the below  conditions are met:  (a) The user has been given explicit authorization to register identifiers. |
| **Inputs** | String identifier |
| **Outputs** | boolean |
| **Post-Conditions** | None |
| **Alternate Conditions** | None |
| **Exception Conditions** | NamingAuthorityConfigurationException  NamingAuthoritySecurityException |
| **Additional Details** | None |
| **Notes** | None |

#### getParentHierarchy

Return parents of a Specimen Identifier

|  |  |
| --- | --- |
| **Behavior Description** | Return the parent hierarchy of the input Specimen Identifier. |
| **Pre-Conditions** | None |
| **Security Pre-Conditions** | None |
| **Inputs** | String |
| **Outputs** | Tree |
| **Post-Conditions** | None |
| **Alternate Conditions** | None |
| **Exception Conditions** | NamingAuthorityConfigurationException  InvalidIdentifierValuesException  InvalidIdentifierException  NamingAuthoritySecurityException |
| **Additional Details** | None |
| **Notes** | None |

#### getChildHierarchy

Return children of a Specimen Identifier.

|  |  |
| --- | --- |
| **Behavior Description** | Return the parent hierarchy of the input Specimen Identifier. |
| **Pre-Conditions** | None |
| **Security Pre-Conditions** | None |
| **Inputs** | String |
| **Outputs** | Tree |
| **Post-Conditions** | None |
| **Alternate Conditions** | None |
| **Exception Conditions** | NamingAuthorityConfigurationException  InvalidIdentifierValuesException  InvalidIdentifierException  NamingAuthoritySecurityException |
| **Additional Details** | None |
| **Notes** | None |

#### registerSite

Register Site information for the user.

|  |  |
| --- | --- |
| **Behavior Description** | Registers site information for the user. |
| **Pre-Conditions** | None |
| **Security Pre-Conditions** | None |
| **Inputs** | String application  String applicationURL  String applicationVersion  String contactName  String contactEmail  String contactPhone  String organization |
| **Outputs** |  |
| **Post-Conditions** | None |
| **Alternate Conditions** | None |
| **Exception Conditions** | NamingAuthorityConfigurationException  InvalidIdentifierValuesException  InvalidIdentifierException  NamingAuthoritySecurityException |
| **Additional Details** | None |
| **Notes** | None |

#### addSite

Associate Site information to an identifier.

|  |  |
| --- | --- |
| **Behavior Description** | Associates registered site information to an identifier. |
| **Pre-Conditions** | None |
| **Security Pre-Conditions** | None |
| **Inputs** | String |
| **Outputs** |  |
| **Post-Conditions** | None |
| **Alternate Conditions** | None |
| **Exception Conditions** | NamingAuthorityConfigurationException  InvalidIdentifierValuesException  InvalidIdentifierException  NamingAuthoritySecurityException |
| **Additional Details** | None |
| **Notes** | None |

## Message Information Model

Communication protocol of the service is SOAP 1.1, so all messages of the service will contain appropriate SOAP envelope and will conform to this protocol standard.

### Information Model

The platform for this PSM is chosen to be Web Service, thus Platform Independent Model presented in the PIM specification document is transformed into Web Service Specific Model represented by an XML Schema

A section of the WSDL file that describes the messages that are exchanged by this system.

<types>

<schema xmlns=*"http://www.w3.org/2001/XMLSchema"* xmlns:xs=*"http://www.w3.org/2001/XMLSchema"* xmlns:wsrbf=*"http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-BaseFaults-1.2-draft-01.xsd"* targetNamespace=*"http://identifiers.cagrid.nci.nih.gov/IdentifiersNAService"* elementFormDefault=*"qualified"* attributeFormDefault=*"unqualified"*>

<import namespace=*"http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-BaseFaults-1.2-draft-01.xsd"* schemaLocation=*"../wsrf/faults/WS-BaseFaults.xsd"* />

<import namespace=*"gme://caGrid.caBIG/1.0/gov.nih.nci.cagrid.metadata.service"* schemaLocation=*"./xsd/cagrid/types/service/servicemodel.xsd"* />

<import namespace=*"http:*[*/identifiers.cagrid.nci.nih.gov/IdentifiersNAService/types"* sche](/identifiers.cagrid.nci.nih.gov/IdentifiersNAService/types%22%20sche)aLocation=*"./IdentifiersNAServiceTypes.xsd"* />

<import namespace=*"http:*[*/na.cagrid.org/1.0/NamingAuthority"* sche](/na.cagrid.org/1.0/NamingAuthority%22%20sche)aLocation=*"./org.cagrid.identifiers.namingauthority.xsd"* />

<import namespace=*"gme://caGrid.caBIG/1.0/gov.nih.nci.cagrid.metadata.security"* schemaLocation=*"./xsd/cagrid/types/security/security.xsd"* />

<import namespace=*"gme://caGrid.caBIG/1.0/gov.nih.nci.cagrid.metadata.common"* schemaLocation=*"./xsd/cagrid/types/common/common.xsd"* />

<import namespace=*"gme://caGrid.caBIG/1.0/gov.nih.nci.cagrid.metadata"* schemaLocation=*"./xsd/cagrid/types/caGridMetadata.xsd"* />

<element name=*"CreateIdentifierRequest"*>

<complexType>

<sequence>

<element name=*"identifierData"*>

<complexType>

<sequence>

<element ref=*"ns0:IdentifierData"* minOccurs=*"1"* maxOccurs=*"1"* />

</sequence>

</complexType>

</element>

</sequence>

</complexType>

</element>

<element name=*"CreateIdentifierResponse"*>

<complexType>

<sequence>

<element ref=*"ns0:Identifier"* minOccurs=*"1"* maxOccurs=*"1"* />

</sequence>

</complexType>

</element>

<element name=*"ResolveIdentifierRequest"*>

<complexType>

<sequence>

<element name=*"identifier"*>

<complexType>

<sequence>

<element ref=*"ns0:Identifier"* minOccurs=*"1"* maxOccurs=*"1"* />

</sequence>

</complexType>

</element>

</sequence>

</complexType>

</element>

<element name=*"ResolveIdentifierResponse"*>

<complexType>

<sequence>

<element ref=*"ns0:IdentifierData"* minOccurs=*"1"* maxOccurs=*"1"* />

</sequence>

</complexType>

</element>

<element name=*"DeleteKeysRequest"*>

<complexType>

<sequence>

<element name=*"identifier"*>

<complexType>

<sequence>

<element ref=*"ns0:Identifier"* minOccurs=*"1"* maxOccurs=*"1"* />

</sequence>

</complexType>

</element>

<element name=*"keyNames"*>

<complexType>

<sequence>

<element ref=*"ns0:KeyName"* minOccurs=*"1"* maxOccurs=*"unbounded"* />

</sequence>

</complexType>

</element>

</sequence>

</complexType>

</element>

<element name=*"DeleteKeysResponse"*>

<complexType>

<sequence />

</complexType>

</element>

<element name=*"CreateKeysRequest"*>

<complexType>

<sequence>

<element name=*"identifier"*>

<complexType>

<sequence>

<element ref=*"ns0:Identifier"* minOccurs=*"1"* maxOccurs=*"1"* />

</sequence>

</complexType>

</element>

<element name=*"identifierData"*>

<complexType>

<sequence>

<element ref=*"ns0:IdentifierData"* minOccurs=*"1"* maxOccurs=*"1"* />

</sequence>

</complexType>

</element>

</sequence>

</complexType>

</element>

<element name=*"CreateKeysResponse"*>

<complexType>

<sequence />

</complexType>

</element>

<element name=*"ReplaceKeyValuesRequest"*>

<complexType>

<sequence>

<element name=*"identifier"*>

<complexType>

<sequence>

<element ref=*"ns0:Identifier"* minOccurs=*"1"* maxOccurs=*"1"* />

</sequence>

</complexType>

</element>

<element name=*"identifierValues"*>

<complexType>

<sequence>

<element ref=*"ns0:IdentifierValues"* minOccurs=*"1"* maxOccurs=*"1"* />

</sequence>

</complexType>

</element>

</sequence>

</complexType>

</element>

<element name=*"ReplaceKeyValuesResponse"*>

<complexType>

<sequence />

</complexType>

</element>

<element name=*"GetKeyNamesRequest"*>

<complexType>

<sequence>

<element name=*"identifier"*>

<complexType>

<sequence>

<element ref=*"ns0:Identifier"* minOccurs=*"1"* maxOccurs=*"1"* />

</sequence>

</complexType>

</element>

</sequence>

</complexType>

</element>

<element name=*"GetKeyNamesResponse"*>

<complexType>

<sequence>

<element ref=*"ns0:KeyName"* minOccurs=*"1"* maxOccurs=*"unbounded"* />

</sequence>

</complexType>

</element>

<element name=*"GetKeyDataRequest"*>

<complexType>

<sequence>

<element name=*"identifier"*>

<complexType>

<sequence>

<element ref=*"ns0:Identifier"* minOccurs=*"1"* maxOccurs=*"1"* />

</sequence>

</complexType>

</element>

<element name=*"keyName"*>

<complexType>

<sequence>

<element ref=*"ns0:KeyName"* minOccurs=*"1"* maxOccurs=*"1"* />

</sequence>

</complexType>

</element>

</sequence>

</complexType>

</element>

<element name=*"GetKeyDataResponse"*>

<complexType>

<sequence>

<element ref=*"ns0:KeyNameData"* minOccurs=*"1"* maxOccurs=*"1"* />

</sequence>

</complexType>

</element>

<element name=*"RegisterGSIDRequest"*>

<complexType>

<sequence>

<element name=*"suggestedIdentifer"* type=*"xs:string"* minOccurs=*"1"* maxOccurs=*"1"* />

<element name=*"parentIdentifiers"* type=*"xs:string"* minOccurs=*"1"* maxOccurs=*"unbounded"* />

</sequence>

</complexType>

</element>

<element name=*"RegisterGSIDResponse"*>

<complexType>

<sequence>

<element name=*"response"* type=*"xs:string"* minOccurs=*"1"* maxOccurs=*"1"* />

</sequence>

</complexType>

</element>

<element name=*"AddSiteRequest"*>

<complexType>

<sequence>

<element name=*"identifier"* type=*"xs:string"* minOccurs=*"1"* maxOccurs=*"1"* />

</sequence>

</complexType>

</element>

<element name=*"AddSiteResponse"*>

<complexType>

<sequence>

<element ref=*"ns0:Identifier"* minOccurs=*"1"* maxOccurs=*"1"* />

</sequence>

</complexType>

</element>

<element name=*"GenerateIdentifiersRequest"*>

<complexType>

<sequence>

<element name=*"numOfIdentifiers"* type=*"xs:int"* minOccurs=*"1"* maxOccurs=*"1"* />

</sequence>

</complexType>

</element>

<element name=*"GenerateIdentifiersResponse"*>

<complexType>

<sequence>

<element name=*"response"* type=*"xs:string"* minOccurs=*"1"* maxOccurs=*"unbounded"* />

</sequence>

</complexType>

</element>

<element name=*"GetParentHierarchyRequest"*>

<complexType>

<sequence>

<element name=*"identifier"* type=*"xs:string"* minOccurs=*"1"* maxOccurs=*"1"* />

</sequence>

</complexType>

</element>

<element name=*"GetParentHierarchyResponse"*>

<complexType>

<sequence>

<element ref=*"ns0:Tree"* minOccurs=*"1"* maxOccurs=*"1"* />

</sequence>

</complexType>

</element>

<element name=*"GetChildHierarchyRequest"*>

<complexType>

<sequence>

<element name=*"identifier"* type=*"xs:string"* minOccurs=*"1"* maxOccurs=*"1"* />

</sequence>

</complexType>

</element>

<element name=*"GetChildHierarchyResponse"*>

<complexType>

<sequence>

<element ref=*"ns0:Tree"* minOccurs=*"1"* maxOccurs=*"1"* />

</sequence>

</complexType>

</element>

<element name=*"RegisterSiteRequest"*>

<complexType>

<sequence>

<element name=*"application"* type=*"xs:string"* minOccurs=*"1"* maxOccurs=*"1"* />

<element name=*"applicationURL"* type=*"xs:string"* minOccurs=*"1"* maxOccurs=*"1"* />

<element name=*"applicationVersion"* type=*"xs:string"* minOccurs=*"1"* maxOccurs=*"1"* />

<element name=*"contactName"* type=*"xs:string"* minOccurs=*"1"* maxOccurs=*"1"* />

<element name=*"contactEmail"* type=*"xs:string"* minOccurs=*"1"* maxOccurs=*"1"* />

<element name=*"contactPhone"* type=*"xs:string"* minOccurs=*"1"* maxOccurs=*"1"* />

<element name=*"organization"* type=*"xs:string"* minOccurs=*"1"* maxOccurs=*"1"* />

</sequence>

</complexType>

</element>

<element name=*"RegisterSiteResponse"*>

<complexType>

<sequence />

</complexType>

</element>

<element name=*"IdentifiersNAServiceResourceProperties"*>

<complexType>

<sequence>

<element ref=*"ns3:ServiceMetadata"* />

</sequence>

</complexType>

</element>

</schema>

</types>

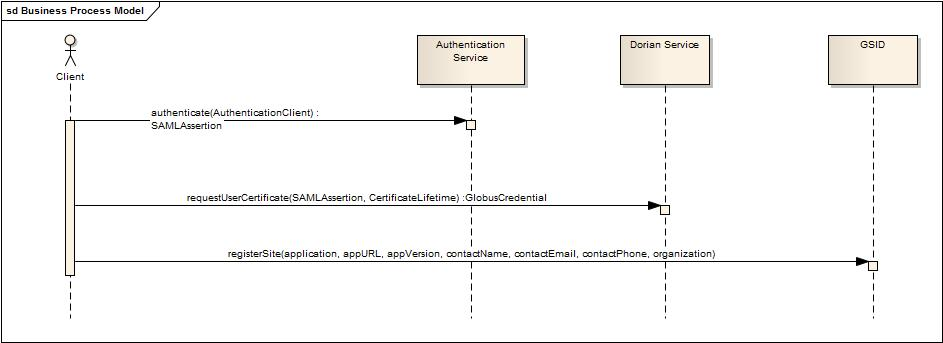
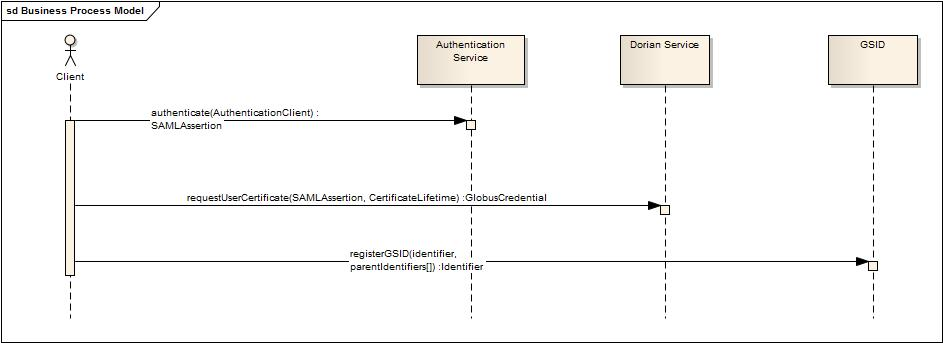
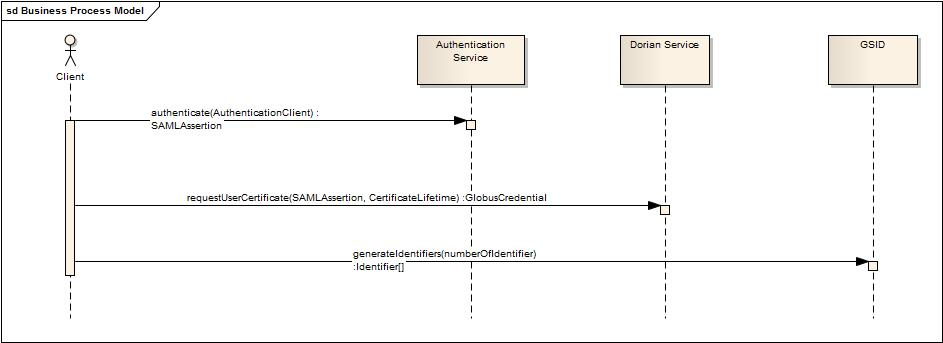
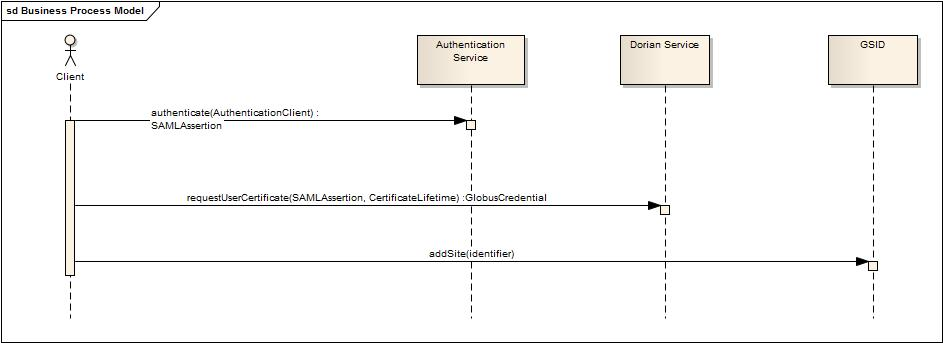
## Service Interactions

### Interaction Details

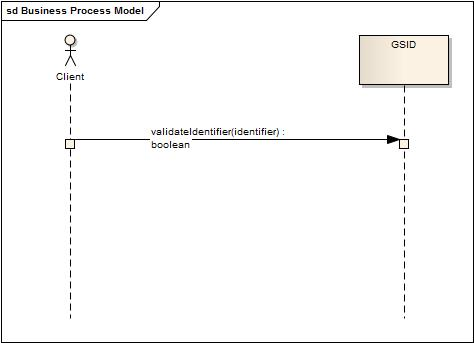
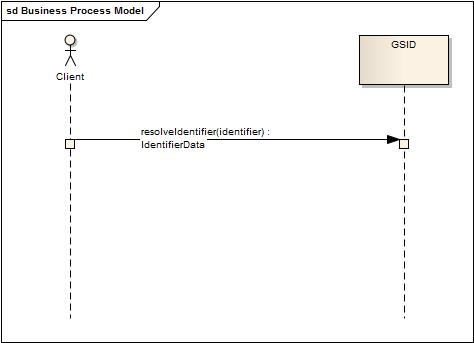
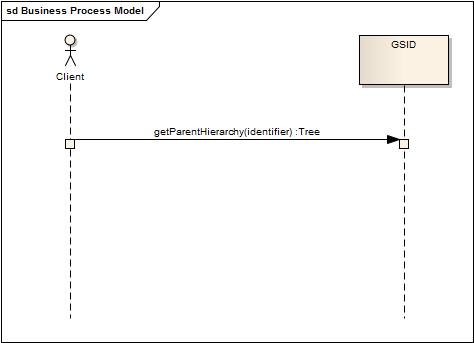
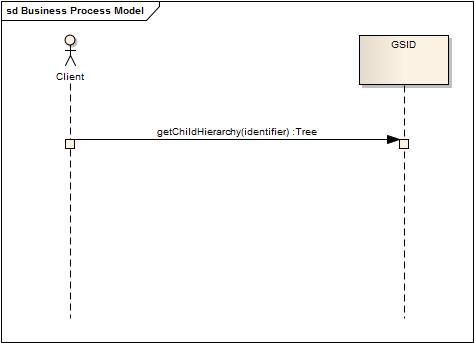


The above diagram highlights the interaction of various services with the Specimen Identifier Management service. A Specimen Management system receives security credentials using the Security Framework. Using these credentials it registers a Specimen Identifier with the Specimen Identifier Management service. At a later time a client (typically another specimen management system) accesses the Specimen Identifier Service to resolve a specimen identifier. Typically the client already knows of the Specimen Identifier Management service, but in the event that it does not, it uses a prefix Authority to discover the service. Once having access to Specimen Identifier Management service it can retrieve dependency information of a Specimen Identifier or it can access further information by using the Site information associated with a Specimen Identifier. The Site information should provide information as to which information service is aware of the biospecimen and using the Specimen Identifier, one may use it as a key to retrieve additional information from the information service.

The following diagrams depict how the service in invoked through using the Grid Services framework. The following four diagrams show an authenticated user performing operations on the service.



The following diagrams show a non-authenticated user performing allowed operations on the service.



## Implementation Considerations

### Security

|  |  |
| --- | --- |
| **Access Control** | |
| Does the Service require Access Control mechanism to be in place to restrict access to only authenticated users or systems? | Yes |
| If Yes then provide the following in detail: | |
| The service uses standard grid infrastructure to authenticate Globus grid credentials. | |
| **Application (Service) Security [Access Policy]** | |
| Does the Service incorporate any security controls (Authorization) to ensure that access to information is granted to only the authorized users / systems? | Yes |
| If Yes then provide the following in detail: | |
| A default security policy has been implemented and included in the appendixes. | |
| **Cryptography** | |
| Does the Service require encryption of data transmitted to and from it? | No |
| If Yes then provide the following in detail: | |
|  | |
| **Information Security and Risk Management** | |
| Is the information served by the service confidential or privileged? And if yes, is it at risk from any external threats or vulnerabilities? | No |
| If Yes then provide the following in detail: | |
|  | |
| **Legal, Regulations, Compliance and Investigations** | |
| Does the information served by the service fall under any legal / regulatory compliance either at federal, state, local or institutional level ? | No |
| If Yes then provide the following in detail: | |
|  | |
| **Telecommunications and Network Security** | |
| Does the service need any network or transport level security such as SSL, Firewall protection etc. | No |
| If Yes then provide the following in detail: | |
| The framework does not require encryption, but it can be configured when a secure deployment is required. | |

### Auditing

N/A

### Privacy

N/A

### Error Handling

#### Overview

The framework interface throws typed exceptions. The resolution over HTTP may produce standard HTTP errors.

#### Error Object Details

|  |  |
| --- | --- |
| **Error Code** | **Error/Exception Description** |
| HTTP 403 | HTTP request to resolve identifier has been denied (user is not authorized). |
| NamingAuthorityConfigurationException | A configuration error has been detected by the naming authority while serving a request. |
| NamingAuthoritySecurityException | Request denied due to authorization error. |
| InvalidIdentifierException | The identifier provided is invalid or does not exist. |
| InvalidIdentifierValuesException | The identifier metadata provided is invalid. |

## Deployment Considerations

### Deployment Modes

The service will be deployed as a caGrid service using SOAP 1.1 binding over HTTPS in a WS-I Basic Profile v1.1-compliant container.

### Performance

The table below lists, per operation, the maximum average performance conformant implementations must meet, under the load given in the next section. Times are expressed in seconds, and should be measured using clients on networks local to the deployed services. Roundtrip times are for full message delivery, but do not include network latency or client serialization/de-serialization overhead.

|  |  |
| --- | --- |
| **Operation** | **Maximum Average Roundtrip** |
| getKeyNames | < 2.0s |
| getKeyData | < 2.0s |
| resolveIdentifier | < 2.0s |
| addSite | < 2.0s |
| registerGSID | < 2.0s |
| generateIdentifiers | < 2.0s |
| validateIdentifier | < 2.0s |
| getParentHierarchy | < 2.0s |
| getChildHierarchy | < 2.0s |

### Scalability

Conformant services will support the performance metrics described in the previous section under a sustained maximum load of 2 requests per second. The requests should be a random mix of the available operations.

### Discovery

The service posts all specifications in a WSDL and is accessible via web service endpoints. The specification does not cover how interested parties discover available web service endpoints. It is conceivable that there will be global and local registries, such as UDDI, that list available service providers in the context of further metadata aiding the discovery of relevant services. Clients can invoke the web service via any SOAP-based service client applications.

### Uptime

The expected uptime for this service is to be 99.9% during business hours.

### Failover

Failover should be provided for endpoints in the case of failing server software (servlet container, database) or hardware. Recommended is a second server (hardware) that can either be switched in when the primary hardware fails or concurrently deployed in a load balancing setup.

## Constraints

No known constraints at this time.

## Known Issues

No known issues at this time.

# Recommendations for Conformance and Compliance

## Conformance Assertions

| Name | Type | Viewpoint | Description | Test method |
| --- | --- | --- | --- | --- |
| Observable behavior | Prohibition | Computational | Service clients are prohibited from making any assumptions based on the internal implementation details of the Service. As a corollary, all observable behavior must be made explicit in the Service specification. | 1. Design Review  2. Change of implementation not affect clients |
| Functional Profile | Obligation | Computational | All functional profiles must be implemented. | 1. Design review  2. Inspection of WSDL and XSD |
| Operations required | Obligation | Computational | Designers/implementers are obligated to implement all of the operations of both identified functional profiles. | 1. Test cases include all identified operations |
| Information Model Conformance | Obligation | Information | The Information Model representation will conform to the Information model provided in Section 3.5.1. | 1. Design Review  2. Test cases to be defined to test specified information model conformance |
| Multiple Jurisdictions | Obligation | Enterprise | Implementers will provide an implementation that supports multiple jurisdictional domains to support different Coordinating Center and Sites as well as multi-site trials | 1. Test cases include multiple domain scenarios |
| Query Performance | Obligation | Enterprise | The performance of the “registerGSID” operation should be within 0.2 second to return a single record from the underlying database | 1. Test cases to include performance testing. |
| Auditing | Obligation | Engineering | Designers/implementers are obligated to decide on and document the approach for logging or auditing of operation calls | 1. Design Review |

# Appendix B – Relevant Standards

Please refer to the Section 1.3 of this document.

# Appendix C – Glossary

Please refer to the caBIG shared glossary located at <https://wiki.nci.nih.gov/display/Suite/10+-+Glossary+v2.2>.