

***Development of the Unified Modeling Language (UML) for GLOBAL Reference Architecture (GRA)***

***GRA-UML***

**Development Approach**

**Version 0.5 (Draft for Review)**

|  |  |
| --- | --- |
| Delivery On: | **May 2014** |
|  |  |
| Prepared For:  Prepared By: | **IJIS Institute**  **Model Driven Solutions**  **Open Networks, Inc.** |

Table of Contents

[Introduction 3](#_Toc388005438)

[Objective 3](#_Toc388005439)

[Overview 3](#_Toc388005440)

[Analysis 4](#_Toc388005441)

[Standards Review 4](#_Toc388005442)

[Process Modeling 5](#_Toc388005443)

[Scope Classification 5](#_Toc388005444)

[Redundancy Handling 7](#_Toc388005445)

[Development Approach 8](#_Toc388005446)

[Methodology 8](#_Toc388005447)

[Approach 8](#_Toc388005448)

[Layering 11](#_Toc388005449)

[Templates 14](#_Toc388005450)

[Appendix A: References 22](#_Toc388005451)

[Appendix B: Revision History 23](#_Toc388005452)

Table of Figures

[Figure 1: OMG RFP Component Overview 3](#_Toc387311823)

[Figure 2: GRA-UML Development Approach Overview 10](#_Toc387311824)

[Figure 3: Logical Service Profile Example 12](file:///C:\Users\Cory-c\Documents\Company\MDSSVN\Clients\IJIS\GRAUML\Delivery\docs\GRA%20UML%20Development%20Approach%20v4.1.docx#_Toc387311825)

[Figure 4: Activity Diagram Template 13](#_Toc387311826)

[Figure 5: Service Description Example 14](#_Toc387311827)

[Figure 6: Service Interface Document Example 15](#_Toc387311828)

[Figure 7: Technology Template Overview 16](file:///C:\Users\Cory-c\Documents\Company\MDSSVN\Clients\IJIS\GRAUML\Delivery\docs\GRA%20UML%20Development%20Approach%20v4.1.docx#_Toc387311829)

[Figure 8: UML Model with Template Process Overview 17](#_Toc387311830)

[Figure 9: Service Metadata Table 20](#_Toc387311831)

[Figure 10: Service Interaction Requirement - Specification Table 21](#_Toc387311832)

[Figure 11: Specification Hierarchy 22](#_Toc387311833)

# Introduction

## Objective

GRA-UML leverages a Model Driven Architecture (MDA) approach to provide a standard for modeling Global Information Sharing Initiative (Global) Reference Architecture (GRA) Services Specification Packages (SSPs) with UML. GRA-UML will define UML profiles for modeling GRA SSPs, including one profile to enable the development of Platform Independent Models (PIM), and another profile to provide the parameters required to transform the PIMs into conformant GRA SSP Platform Specific Models (PSMs). The specification will also define the generation of GRA specific artifacts from these profiles.

*Please note that specific examples and diagrams are notional as the specific profile is still undergoing development.*

## Overview

An OMG Model-Driven Architecture (MDA) approach governs the definition of the GRA-UML functionality and the development of structured models. The structured models represent a logical view of the underlying functionality in a platform-independent model (PIM), which is free of any technology designation, including programming languages, operating system and application environments.

The MDA approach leverages standard UML modeling to define the Business Model (PIM), which provides a software independent description of the services automated by a GRA SSP. A model of the SSP defines the interpretation of the PIM and metadata that fully define a valid SSP. A Query/View/Transformation specification (QVT), leveraging other OMG standards, bridges the logical and physical layer, providing a map from the Business Model and SSP model into the physical GRA SSP Technology artifacts.



Figure 1: OMG RFP Component Overview

GRA-UML generates a conformant reference[[1]](#footnote-1) GRA SSP that contains an artifact catalog, the definition of all service metadata, a full Service Description Document (SDD) and one or more Service Interface Definition Documents (SIDD). In addition the GRA-UML model produces the service definition (WSDL) and schema (XSD) technical artifacts.

## Analysis

### Standards Review

The GRA-UML profiles will allow for the specification of and/or reference to the business requirements found in the GRA SSP along with the GRA Web Services Service Interaction Profile (WS-SIP) and the GRA Reliable Secure Web Services Service Interaction Profile (RS WS-SIP). The profile standards define how the options included in the GRA SSP and GRA Web Service SIPs, referenced in Section 6.4.1 “GRA Specifications” of the OMG RFP, provision the reference profile, while not excluding future extension to other specifications, such as ebXML.

The nature of the GRA SSP is to bind the business model to a specific set of contractual obligations, governed by service standards, within a targeted implementation. As such, the scope of GRA is bounded in that the resulting specifications do not dictate the implementation technology, target platform or deployment mechanisms, beyond the specification of service interactions and policy.

GRA-UML will provide for the production of valid, conformant GRA service specifications from a UML model that defines the service contract and policy statements. The GRA-UML model fully specifies the policy information for services as “detailed requirements” down to what would be considered an abstract service description (WSDL) layer. The service definition will have all of the business context and information necessary to drive the production of documentation and the GRA metadata. This will also contain the detailed messaging requirements (e.g. non repudiation, confidentiality and integrity) as required at the business level.

The “double edged sword” of standards is that if they are underspecified achieving interoperability usually requires supplemental agreements/contracts. On the other hand, if standards are over specified the implementation is often complicated and restrictive. For example, within a secure network a given web service may not need any encryption, but over the internet the same web service may require message-level encryption. For many of the business requirements there are several standards that can be used to satisfy the condition. In some cases GRA provides insufficient detail as to how those specifications are to be used to enable interoperability. There are permutations and interactions among the various options. Despite all these options the vast majority of SSPs use basic web services, as such web services should be the baseline SIP.



GRA-UML will allow for the specification of business and exchange requirements in the GRA SSP, the GRA Web Services SIP (WS-SIP) and the GRA Reliable Secure Web Services SIP (RS WS-SIP).

### Process Modeling

The request for proposal requires that the resulting profile be based on elements from UML, SoaML and the UML profile for BPMN. Process modeling is referenced in the GRA specifications.

It has been stated by the GSC that there is a desire to separate processes from services in a SSP, that in fact the SSP should not include services. Due to this new information references to process considerations have been removed. If this implies any other changes to the required content (e.g. metadata) we request this information be provided in a timely manner.

### Scope Classification

GRA-UML endeavors to simply the production of conformant GRA specifications that meet the needs of the most typical system specifications. In order to define the typical system, the project will focus on a user-centric, top-down SOA profile that has sufficient information to generate a valid and useful set of GRA artifacts. To that end, the following discussion further outlines the issues and associated constraints.

The GRA is intentionally loosely specified and open ended. As a result, a general purpose representation for service definition policy (and other) extensions (e.g. WSDL extension) would be unwieldy for a modeler. However, it has been observed that there is, in practice, little or no variance between the policies applied across the spectrum of existing SSPs. In order to complete the GRA-UML profile, the key patterns of policy requirements will be identified, which will substantially simplify the modeling process and improve consistency across provisioned SSPs.

The SIP Specifications reference many independent web service extensions, each associated with a specification, namespace and schema. The SSP reference template contains approximately 10 various service definition (WSDL) extensions, of which 5 are not mentioned in the SIP Specifications. The set of published SSPs contain 9 different WSDL extensions, of which 6 are not mentioned in either the SIP Specification or the SSP reference template. An isomorphic representation of these extensions in UML would likely require numerous stereotypes, awkward modeling patterns, and much coupling between technology binding and business requirements.

In order to accommodate the open-ended, extensible nature of the GRA, the GRA-UML will model all GRA defined business requirements and then provide a template that covers the reliable secure web service interaction profile. The template will essentially be a parameterized set of SSP artifacts that are predefined to satisfy base SSP components as well as specific technologies in a given class of service interactions. The parameters will be generated from the UML model. The reference specifications will provide a guide to the default template. Refer to the “Template” section of the “Development Approach” below for a complete description of the template design.



The GRA-UML will be as flexible as possible in allowing various *technology templates*, which represent a specific set of GRA standards. As a result, the choice of the technology profile will be configurable via templates.

The objective of the SSP provisioning process is to produce an SSP conformant with the GRA architectural requirements, including the Reliable Secure Web Services SIP. Compliance with the RS-WS SIP conformance targets can be achieved with minimal modeling effort when the key policy application patterns are identified. The goal would be to cover the vast majority of use cases without requiring explicit technology bindings for each possible extension point.

The scope of the GRA-UML profile will be to produce GRA artifacts defined by the reference GRA specifications provided, specifically the Offender Transfer Notification Service (OTNS), the SORNA Inter-Jurisdiction Relocation Service (SIRS) and the Prescription Monitoring Program Information Exchange Service (PMIX) reference SSPs. In that way, all specifications and extensions included in the reference profiles will be included in the GRA-UML profile. If, on the other hand, a potential capability is not used in a reference specification, then that extension mechanism will not be considered an explicit requirement for the profile. As such, being able to consume every possible option of every existing GRA specification is secondary.



The GRA-UML profile will be scoped to the reference specifications to ensure that the most critical components of the GRA are included in the UML model while generating a usable framework that is not unwieldy for the implementer. Extension will be provided based on templates.

### Redundancy Handling

One of the problems with the suite of GRA specifications and templates is lack of precision, clarity and consistency of descriptions of concept, scope, purpose, name and description across required artifacts. An objective of the UML model is to ensure that each unique concept reflects the normative definition of the GRA concept currently distributed across multiple specifications/templates. There should be a single source of the truth for any GRA concept.

The GRA maintains multiple methods of representing the same or similar information, such as service names/namespaces, descriptions and summaries, which can be represented in the metadata, the SDD, the service description (WSDL), the data type definition (XSD) and/or the service interaction profile (SIP). The GRA does not contain any constraints that ensure the data is included or consistent across the various locations.



In order to provide for consistency and simplicity, the GRA-UML profile will represent each element of service information with a singular model element, thereby eliminating redundancy in the model and ensuring consistency in the generated specification packages.

The existence of redundant sets of schemas for an IEPD and the SIP will be mitigated by the new GRA Service Specification Guidelines. However, there are many other artifacts in an SSP which relate a set of common concepts across different viewpoints. In the case of many of the current SSPs, the identity, description and relationships of those concepts are not consistent across different viewpoints. This is to be expected in an environment where artifact production is manually implemented, at different points of time, by different providers and with no means of reliably synchronizing the various forms of artifacts. An objective of the UML model is to ensure that those concepts are semantically consistent, non-redundant and at the suitable level of abstraction to enable provisioning of a consistent set of required artifacts.

# Development Approach

## Methodology

The GRA-UML development approach will follow an iterative development methodology that allows the project team to identify the tasks, establish a prioritized backlog and then work through the specification addressing the highest priority tasks first. As the scope of GRA is quite broad, an iterative approach ensure that the GRA-UML Profile is continually updated in a manner that leaves the profile in a stable, usable state. The methodology not only ensures that the highest priority (most critical) aspects of the profile are added first, but also leaves the model ready for testing at the end of each component cycle.

## Approach

Multiple levels of abstraction leading from business requirement to technical implementation need to be tractable, constrained and validated. The more abstract aspects of the model need to be identified and isolated from the technology bindings expressible in a specific specification.

The major model components of a specification are categorized as follows:

* UML Profiles

The GRA-UML will consist of 2 UML Profiles: a Logical Service Profile and a GRA SSP Profile, while leveraging other existing OMG Profiles such as NIEM-UML and SoaML.

* Baseline Template

The combination of the models, defined in the above profiles, and a template will produce a SSP. The baseline template will provide a set of artifacts with substitutable parameters. The provisioning process will “fill in” these parameters with information derived from the model. The baseline template will implement the GRA Reliable Secure Web Services SIP (RS WS-SIP). Users will be able to modify the baseline template or provide their own the define specific technology or stylistic choices.

* QVT Transformations

QVT transformations, perhaps combined with other technologies such as the “OMG MOF 2 Text” standard, will be used to produce the information used to parameterize the templates. QVT is an OMG specification for defining model to model transforms. MOF 2 Text is an OMG standard for producing textual artifacts from models.



Figure 2: GRA-UML Development Approach Overview

In order to define the major model components, the user defined GRA-UML model will be partitioned into the following forms of model resources:

* Business model.

The model would be technology agnostic, encompassing those concept abstractions from business requirements, capabilities and contracts to Information Modeling.  The same Business model or business model elements could ultimately target multiple SSP targets. The business model will focus on the service definition.

* SSP model.

The model will be relatively simple, assuming the existence of the templates.  A SSP model will represent the SSP plus template instances, which are deployed to realize specific aspects of the business model in the resulting SSP.  Refer to the Template section below for information pertaining to the baseline template specification.

### Layering

*Logical Services Profile (PIM Layer)*

* The Logical Services Profile will not be GRA specific, but will be tailored for GRA.
* The profile will consist of normal UML Interfaces that use NIEM data types in operations as well as signals (notification). Note: no extra layer is needed between NIEM types and the corresponding interfaces.
* The profile will also include participant components with ports that provide and/or use these interfaces – like SoaML with no extra stereotypes required.
* The following Logical Service model, based on SIRS, provides a notional example:

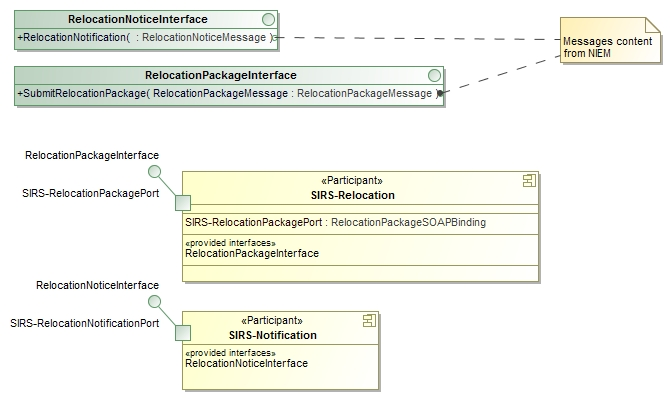


Figure 3: Logical Service Profile Example

*Choreography Diagrams (PIM Layer)*

* **Note: Due to changes in scope of an SSP, the activity diagram may be replaced with a sequence diagram or collaboration as these are more service specific.**
* Examples to be forthcoming.

*GRA Service Description*

* The GRA Service Description (PSM Layer) is where the model gets GRA specific, but is still mostly technology independent.
* The Service Description includes a component or artifact that represents a GRA service as well as nested components that represent the ports and interfaces.
* The service descriptions will contain stereotypes to capture metadata not provided by other model elements as well as business requirements such as the policy constraints enumerated in the service specification guidelines.
* SIP level components will capture technology specific choices and metadata.
* Each component of the Service Description realizes something in the logical layer.
  + GRA specific stereotypes map to GRA metadata including service requirements
* Ability to attach external document templates (e.g. SDD document) and document artifacts for specific purposes (e.g. business overview).
* References to one or more service interfaces (in the GRA sense).
* The following Service Description, based on SIRS SSP, provides a notional example:

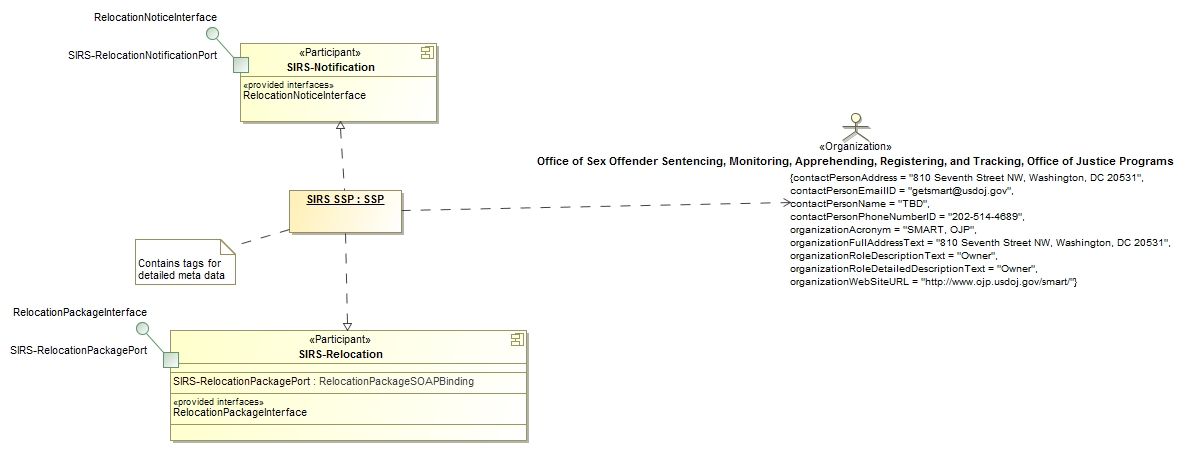


Figure 5: Service Description Example

*Service Interface Document (PSM Layer)*

* Provides technology binding, policy at the technology level.
* Also may have or augment components or artifacts representing GRA service as well as nested components for the ports and interfaces, corresponding most directly to the WSDL layers, however other technology bindings are possible using other templates.
* References a technology template. Template components able to include model content (either source model or derived XML) content to bind it to the particular SSP. At the technology level, most of the content is in metadata.xml and the WSDL.
* The following notional diagram, based on SIRS SSP, provides a notional example:

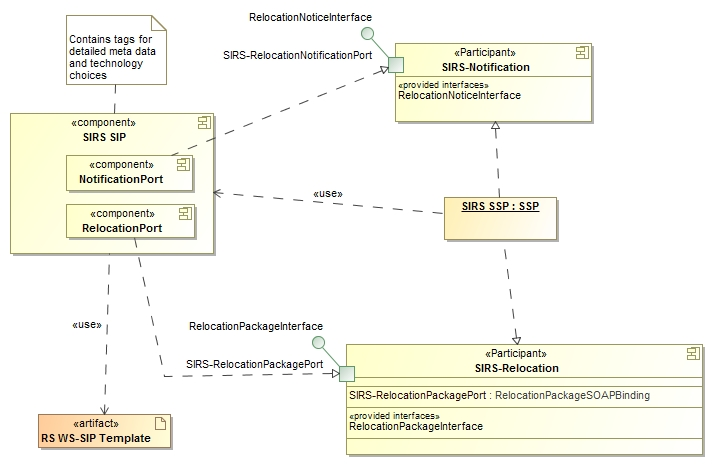


Figure 6: Service Interface Document Example

### Templates

A “technology template” will serve as a parameterized, SSP template in which the specific technologies are predefined to satisfy a given class of service interactions. The parameter values will be generated from the UML model. The template will be in the format of the desired SSP with a transformation section, which will use the SSP metadata from the UML model as input and produce the final service specification package. The templates will, in essence, move some of the provisioning from Query/View/Transformation (QVT) to platform specific technologies as defined by the templates. Therefore, if an implementer requires different technology choices, then they would be expected to produce a derivative template. This would also allow addition of “soft content” like logos and documentation sections.

Within the context of a model, GRA artifacts are a form of manifestation for the business model.  The template artifact instances constitute a specific configuration of properties for a targeted architecture.  Thus, the policy specification requirements for GRA can be represented as a suitably constrained setl of artifacts.

A reusable inventory of preconfigured templates (artifact instance patterns) could be easily and quickly applied to a business model.  The business model, SSP model and template are the resources to produce an SSP.

Figure 7: Technology Template Overview

The SSP model pattern can easily be extended to incorporate additional technology bindings, their constraints, properties, and relationships.  Similarly the inventory of re-usable templates could be augmented to accommodate common enterprise-specific technology bindings.

The following diagram depicts the UML modeling process leveraging technology templates to reflect the business parameters and specific technologies defined for the underlying service interactions. The diagram illustrates the following points:

1. A UML Tool will extract data from the GRA SSP UML Model and transfer the business data into the various parameterized placeholders in the template artifacts.
2. The Technology Template, as described above, represents the GRA SSP framework, which can be the GRA-UML baseline template or a user-defined template.
3. The process will result in the generation of GRA SSPs that have consistent information throughout the artifacts since the data is pulled from the UML Model. In addition, generated SSP artifacts will be consistent across disparate specifications.



Figure 8: UML Model with Template Process Overview

The standard and implementations would come with one template for secure WS that would produce a valid SSP, thus satisfying both requirements. The data from the model would both select the template and provide it with data.  As the model would be part of the SSP, all data would be visible, but the majority of the data will likely come from the simplified XML Metadata generated by the model.

The GRA UML profile will initially support a standard “baseline” technology template that represents the core SSP artifacts along with the parameters and constraints that govern the generation of a reliable secure web services SIP.

**Service Metadata**

|  |  |
| --- | --- |
| **Service Identification**  A means of uniquely identifying a service. | |
| **Service Name** | Human readable identification for this service |
| **Service Name Abbreviation Text** | Human readable abbreviation of the Service Name. |
| **Service Description**  A set of details describing a service. | |
| **Service Purpose Text** | Defines the purpose the service intends to perform / accomplish. |
| **Service Scope Description Text** | Defines the scope of the service. |
| **Service Description Summary Text** | Brief summary of this service for short display purposes— maximum of 160 characters including spaces. |
| **Service Description Text** | Human readable narrative description of this service—may contain as much detail as you think useful to those with a potential interest in this service and its business usage/application. |
| **Service Description Keyword Text** | Search terms that would not otherwise be in other metadata attributes (e.g., Child Support Warrant, Domestic Relations Warrant, Domestics). |
| **Service Capabilities** | An enumeration of the capabilities provided by a service. The capabilities will be described in a free text format. |
| **Service Real World Effects** | An enumeration of the Real World Effects Provided by the Service. The Real World Effects will be described in a free text format. |
| **Security Classification Text** | Any applicable classification of the security level of the information exchanged by the service, such as SBU, Secret, etc. If there is no strict classification this field can contain a brief statement regarding the security of the data. |
| **IEPD references**  This identifies all of the Information Exchange Package Documents which the service uses in its data model. | |
| **IEPD Name** | A human readable identification of the IEPD. |
| **IEPD URL** | A URL where the IEPD is posted and available. |

|  |  |
| --- | --- |
| **Version Information**  A structured representation of a version for something, such as a service, document, etc. | |
| **Major Version** | Service Major Version Number. |
| **Minor Version** | Service Minor Version Number. |
| **Revision Version** | Service Minor Version Number. |
| **Creation Date** | YYYY-MM. This is the date when the service was first created. Do NOT confuse with the date you submitted this service to a registry. |
| **Activation Date** | YYYY-MM. The date when the service was or will be first available in production. Do NOT confuse with the date submitted this service to a registry. |
| **Last Revision Date** | Year and month (YYYY-MM) this service information was last revised. Do NOT confuse with the date the service itself was last revised with generating new service documentation. |
| **Lifecycle Status** | Identifies the current stage of the service within the lifecycle. Valid values are; In Design, In Development, Release Candidate, Operational/Production, Deprecated. |

|  |  |
| --- | --- |
| **Related Organization Information**  A collection of organizations that are somehow related to the service. | |
| **Organization Name** | The full name of the agency. |
| **Organization Acronym** | The acronym for the agency. |
| **Organization Full Address** | A physical address of an agency in full text form. |
| **Organization Web Site URL** | Internet address of the agency's web site. |
| **Organization Role Description Text** | The organization role defined in free form text. That could be creator, provider, owner, maintainer, authorities source, etc. |
| **Organization Role Detailed Description** | More detailed text explaining the role of the organization responsibilities. |
| **Contact Person Name** | Person designated as the point of contact for the organization. |
| **Contact Person Email** | Email of the person designated as the point of contact for the organization. |
| **Contact Person Phone Number** | Phone number of the person designated as the point of contact. |
| **Contact Person Address** | Physical address of the person designated the point of contact. |

|  |  |
| --- | --- |
| **Business Context**  A collection of details describing the business context in which the service operates. | |
| **Domains** | Primary domains or line(s) of business that this service covers. |
| **Exchange Partner Types** | Types of organizations that would commonly use this service. |
| **Sponsors** | Names and acronyms of professional or governmental organization(s) that sponsored, contributed, or participated in the development of the service. |

|  |  |
| --- | --- |
| **Service Interface**  A set of details relating to the interface of a service. | |
| **Security Implemented** | Identifies if Security has been implemented to access this service. This is a Yes/No field. |
| **URI**  A complex element containing URIs information. | |
| **URI Address** | Fully qualified locator of the service interface potentially including version and environment. |
| **URI Description** | Description of the URI. This would provide better understanding of what is actually at the URI address. |
| **Service Action**  A complex element containing Actions information. | |
| **Action Name** | Service Action performed within this service interface. |
| **Action Purpose** | Description of the Service Action purpose performed by this service interface. |
| **Message Exchange Pattern** | Category of transaction this service is designed and used for: query/response, message, publish/subscribe, document, etc. |
| **SIP**  A complex element containing information about the Service Interaction Profile Implemented. | |
| **SIP Name** | Name of the Service Interaction Profile Implemented. |
| **SIP Version** | Version of the Service Interaction Profile Implemented. |

|  |  |
| --- | --- |
| **Service Level Agreements**  A collection of policies, agreements, licensing and any other governance or performance documentation | |
| **Service Policy And Contracts**  A collection of policies specifying constraints and any other details regarding the realization of a service. | |
| **Applicable Policies Indicator** | True when there are any applicable policies governing the use, administration, or implementation of a service. |
| **Applicable Policies** | A description or references to an applicable policy governing the use, administration, or implementation of a service. |
| **Performance/Quality of Service Metrics**  A collection of performance quality of service (QoS) characteristics | |
| **Service Response Time** | A description of the average response time for a service. The response time is calculated as the time input is provided to the service until the service completes its process or provides output for the consumer. |
| **Service Usage Details**  A collection of licensing & any other usage governance constraints regarding the realization of a service. | |
| **Approval Required Indicator** | Identifies if a consumer must first obtain the approval prior to using the service. (true=approval required or false=approval not required). |
| **Licensing Required Indicator** | True when a license is required to use the service; False otherwise. |

Figure 9: Service Metadata Table

|  |  |  |
| --- | --- | --- |
| **Requirements** | **Baseline** | **Specification** |
| Service Consumer Authentication | Yes | WS-I Basic Security Profile 1.0  Web Service Security (WSS) |
| Service Consumer Authorization | Yes | WS-I Basic Security Profile 1.0  Web Service Security (WSS) |
| Identity & Attribute Assertion Transmission | No | N/A |
| Service Authentication | No | N/A |
| Message Nonrepudiation | Yes | WS-I Basic Security Profile 1.0  Web Service Security [WSS]  XML Signature [XMLSIG]  Security Timestamp |
| Message Integrity | Yes | WS-I Basic Security Profile 1.0  Web Service Security [WSS]  XML Signature [XMLSIG] |
| Message Confidentiality | Yes | WS-I Basic Security Profile 1.0  Web Service Security [WSS]  XML Signature [XMLSIG]  XML Encryption [XMLENC] |
| Message Addressing | Yes | WS-Addressing 1.0 |
| Reliability | Yes | WS-ReliableMessaging 1.0 |
| Transaction Support | No | N/A |
| Service Metadata Availability | Yes | SSP metadata.xml file |
| Interface Description Requirements | Yes | WSDL 1.1 |

Figure 10: Service Interaction Requirement - Specification Table

**Specification Hierarchy**



Figure 11: Specification Hierarchy

In the course of time, various enterprises will evolve or refine their technology binding requirements.  In these cases, enterprise-specific artifacts and artifact instance libraries could be created and shared across the enterprise.  The libraries may refine, extend, or constrain the core libraries, similar in concept to how an EIEM facilitates reuse within a NIEM-based enterprise.

# Appendix A: References

**GRA** Global Reference Architecture (GRA) Specification, Version 1.9.1

<https://it.ojp.gov/gra>

**GRA** **SSG** GRA Service Specification Guideline, Version 1.0.0

<https://it.ojp.gov/gist/43/The-Global-Reference-Architecture--GRA--Service-Specification-Guideline-V-1-0-0>

**WS-SIP** GRA Web Services Service Interaction Profile, Version 1.3

https://it.ojp.gov/gist/56/Global-Reference-Architecture--GRA--Web-Services-Service-Interaction-Profile-Version-1-3

**RS WS-SIP** GRA Reliable Secure Web Services Service Interaction Profile, Version 1.2

https://it.ojp.gov/gist/85/Global-Reference-Architecture--GRA--Reliable-Secure-Web-Services-Service-Interaction-Profile

**WS-I BP** WS-I Basic Profile, Version 1.2, 2010-11-09,

<http://ws-i.org/profiles/basicprofile-1.2-2010-11-09.html>

**WS-I BSP** WS-I Basic Security Profile, Working Group Draft, March 30, 2007,

<http://www.ws-i.org/Profiles/BasicSecurityProfile-1.0.html>

**WS-I RSP** Reliable Secure Profile Version 1.0, 2010-11-09,

<http://www.ws-i.org/Profiles/ReliableSecureProfile-1.0-2010-11-09.html>

**UML** Object Management Group (OMG) Unified Modeling Language, Version 2.0

<http://www.uml.org/>

**BPMN** OMG Business Process Model & Notation, Version 2.0

<http://www.omg.org/bpmn/index.htm>

**SoaML** Service Oriented Architecture Modeling Language, Version 1.0.1

<http://www.omg.org/spec/SoaML/1.0.1/>

# Appendix B: Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Editor | Change |
| 4/2014 | 0.1 | Cory Casanave,  Model Driven Solutions | Provided Initial Scope and Approach Content |
| 4/2014 | 0.2 | Todd Seymour,  Open Networks | Drafted GRA-UML Development Approach Document |
| 4/2014 | 0.3 | Todd Seymour,  Open Networks | Expanded Template Approach documentation;  Reflected revisions from the team |
| 5/8/2014 | 0.4 | Cory Casanave | Edit for IJIS review |
| 5/16/2014 | 0.5 | Cory Casanave | Removed process per GSC |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

1. A reference SSP provides a template for an SSP that will ultimately be deployed [↑](#footnote-ref-1)