



GRANTS.GOVSM

FIND. APPLY. SUCCEED.SM

Applicant Web Services Integration

June 5, 2009

Applicant Web Services Integration

June 5, 2009

Table of Contents

1. Introduction.....	4
Objective.....	4
System Overview	4
Technical Summary.....	4
Audience.....	4
Related Documents	4
Assumptions	5
2. When to Develop a System-To-System (S2S) Interface	5
Application Volume	5
Business Case	5
Integration with Grants Management Systems.....	6
Decision Selection Criteria	6
3. Technical Requirements.....	6
Client Side Requirements	6
Supported Platforms	6
Technical Staff	7
4. Web Services Interface	7
What It Provides	7
What It Does Not Provide.....	7
Authorized Organization Representative (AOR).....	7
Grants.gov XML Schemas	7
Opportunity Schema	7
Form Schema	8
Header Schema.....	8
Attachment Schema	8
Change Management	8
Finding Opportunities on Grants.gov.....	9
Retrieving Grant Opportunities from Grants.gov.....	9
Schema Location Requirements in the Grants.gov XML	9
Submitting Electronic Applications to Grants.gov	9
Preparing Grant Application XML.....	9
Grants.gov Hashing Standard	10
Preparing and Hashing Attachments	10
Hashing the Grant Forms XML.....	10
Electronic Submission	11
Tracking the Status of Electronic Grant Application Submissions	11
Filter Types for Querying GetApplicationList	12
Processing Status.....	12
Uniquely Identifying Submissions	12
Retrieving Detailed Reasons for a Rejection.....	13
Other Notification Mechanisms.....	13

Applicant Web Services Integration

June 5, 2009

Accessing Grants.gov Test Environment	13
5. Web Services Security	13
Mutual Authentication and SSL	13
6. Grantee organization Options for Web Services Implementation	14
Web Server Description	14
Web Application Servers	14
7. Grants.gov Application Processing	14
Lifecycle of an Application	14
8. Appendix A	15
Grants.gov Contact	15
Acronyms and Definitions	15
Web Resources on SOAP, WSDL and Other Tools and Technologies	15
9. Appendix B	16
Reference Implementation	16
Dynamic Binding to the Grants.gov Applicant WSDL	16
Static Binding to the Grants.gov Applicant WSDL	16
10. Appendix C	16
Sample Opportunity Schema	17

Applicant Web Services Integration

June 5, 2009

1. Introduction

Objective

The objective of this document is to provide grantee organizations the information necessary to accomplish system integration with Grants.gov Applicant Web Services. The document covers both the business and technical requirements of the system. Decision makers are provided decision criteria and an analysis of alternative solutions for submitting electronic applications to Grants.gov. System architects receive comprehensive technical detail on standards and specifications needed to implement client systems.

System Overview

Grants.gov provides applicants a web services interface for automated submission of completed grants applications. The system is designed for secure e-business transaction processing among multiple trading partners. Web Services is the specification implemented by Grants.gov to facilitate an electronic business network. The overall design reflects the best practices and standards defined by the open standards body World Wide Web Consortium (W3C) <http://www.w3.org>. As a result, the maximum flexibility is extended to designers of grantee systems.

Technical Summary

- Operating System and Software Language Agnostic
- Open-specifications Driven
- W3C standards for Web Services
- Extensible Markup Language (XML) <http://www.w3.org/XML>
- Simple Object Access Protocol (SOAP) <http://www.w3.org/TR/soap>
- Hyper Text Transfer Protocol (HTTP) <http://www.w3.org/Protocols/>
- Web Services Description Language (WSDL) <http://www.w3.org/TR/wsdl>
- SSL and Mutual Certificate Based Authentication for security
- Delegation of responsibility to the applicant system for verifying AOR authority to submit applications to Grants.gov
- Push Submission Method
- Reference Implementation

Audience

The audience for this specification includes:

- Project managers
- Software Engineers
- Software Testers
- Other Interested Groups

Related Documents

Applicant Web Services Integration

June 5, 2009

Please refer to the following documents all available in the Reference Implementation
<http://www.grants.gov/DoingBusinessApplicant>

- Applicant Web Services Reference Implementation Guide
 - A complete user guide for the applicant reference implementation
- Applicant Web Services Security
 - Rich source of information related to Grants.gov security including: SSL, Mutual Authentication and Digital Certificates
- Applicant Web Services Definition Language (WSDL)
 - XML document that describes the capabilities of the Grants.gov Webservices as collections of communication endpoints capable of exchanging messages.

Assumptions

- Organizations selecting to participate in the electronic exchange have the technical capacity to do so.
- The information exchange is based on web services, where the Grants.gov system device is listening for messages at all times.
- The communication exchange is based on an agreed upon standard protocol and standard transaction definition.
- Grants.gov web services will process application submission requests one at a time; a separate submission request must be made for each application.
- Authentication is performed using mutual certificate based authentication.

2. When to Develop a System-To-System (S2S) Interface

Developing a Web Services S2S client is a complex task requiring significant resources to develop and maintain. There are many factors to consider before deciding to develop an automated system. Such factors include application volume, a defined business case for automation, and integration with existing grants management systems. The sections below describe each of these considerations.

Application Volume

The volume of applications that an applicant system submits should be a major factor in an organizations decision to develop a system-to-system interface. Organizations that submit less than 100 applications per year should seriously think about the investment in time and money that is required to develop and *maintain* a system-to-system interface.

Business Case

Grantee-organizations are often faced with overwhelming challenges when applying for federal government grants, regardless of which agency they address. Various program requirements and administrative differences necessitate constant updating of procedures, and disparate data standards and business processes require redundant data entry, often resulting in inaccurate

Applicant Web Services Integration

June 5, 2009

application submissions. As a result, a need for automation has evolved. Web Services can assist applicant organizations in accomplishing automation by:

- Eliminating the burden of redundant or disparate electronic and paper-based data collection requirements by capturing grant applications in XML format.
- Simplify and standardize data definitions and application forms via XML Schema.
- Protect the confidentiality, availability, and integrity of data via certificate based mutual authentication and SSL.
- Standardize the collection of financial and progress report data in support of audit and performance measurement activities.
- Improve direct feed into the back office system, facilitating workflow.

Integration with Grants Management Systems

Applicant organizations that have existing grants management systems may leverage web services to submit grant applications electronically to Grants.gov. Applicant organizations can programmatically submit applications in an XML format known as SOAP with Attachments.

Since XML is platform independent, data elements can be mapped using any technology that the applicant organization chooses. As a result, Web Services integration with grants management systems can provide the following benefits:

- Eliminates human data entry errors
- Provides ability to integrate with existing workflow processes
- Provides better management and reporting capabilities

Decision Selection Criteria

This section provides some initial criteria for management personnel in selecting an integration solution. If applicant organizations have low volume or limited IT resources and are not seeking to automate the submission process then they should consider with the person-to-system solution. The system-to-system solution requires a significant investment on the part of grantee organizations. The person-to-system offers a lower risk and lower cost alternative such as the Grants.gov provided Adobe Reader solution (http://www.grants.gov/help/download_software.jsp#adobe811).

3. Technical Requirements

Client Side Requirements

The system-to-system interface requires grantee organizations to develop a W3C compliant SOAP client to interact with the Grants.gov web services interface. Integrators shall comply with the W3C Web Services (HTTP, SOAP v1.1, SOAP with attachments XML, and WSDL). In addition, the client system must support certificate based mutual authentication over 128-bit SSL.

Supported Platforms

Due to the use of standards-based communications methods, web services can be developed in various programming languages, including Java, .NET, C++, VBScript, JavaScript, PHP, or PERL. More importantly, web services are “platform-independent”; i.e., web services can execute on virtually all platforms, including Linux, NT/2000/XP, HP-UX, and Solaris.

Applicant Web Services Integration

June 5, 2009

Technical Staff

Technical staff should have knowledge of web services and the W3C technology standards. These specifications provide a software architect the foundation for building a client interface to Grants.gov.

4. Web Services Interface

What It Provides

The Web Services interface provides a platform independent messaging service that follows the SOAP with attachments specification. Below is a brief overview of the Web Services methods that are currently available:

- *GetOpportunityList* – Web service to query for grant opportunities that are available on Grants.gov for electronic submission
- *SubmitApplication* – Web service to submit application packages to Grants.gov for the DUNS number(s) associated with the client's digital certificate
- *GetApplicationList* – Web Service to obtain a list of the Grant Applications that have been received by Grants.gov for the DUNS number(s) associated with the client's digital certificate
- *GetApplicationStatusDetail* – Web Service to retrieve a detailed status for a specific application that has been received by Grants.gov for the DUNS number(s) associated with the client's digital certificate

What It Does Not Provide

The Web Services interface of Grants.gov does not support:

- Verification of AOR authority to submit is not validated by Grants.gov. The grantee organization is responsible for ensuring that electronic submissions accurately reflect organizational rules governing such permissions.
- Applicant user management, posting of opportunities, and the communication with applicants or grant making agencies.
- Software to populate the Grants.gov application XML shall be developed by the grantee organization. Client integrators shall develop the software to produce XML that conforms to Grants.gov [Opportunity Schemas](#).

Authorized Organization Representative (AOR)

The AOR is a delegated authority to submit on behalf of an institution by the E-Biz Point of Contact (POC). From a business perspective, the POC is solely responsible for determining who has authority to submit. The POC could delegate such authority to many people, or to one other person, or to retain such authority solely to him/herself. Grants.gov shall validate and authenticate the grantee organization's system when accessing Grants.gov web services, but Grants.gov shall **not** validate the AOR name on the grant application XML. The grantee organization is responsible for ensuring that electronic submissions accurately reflect organizational rules governing such permissions.

Grants.gov XML Schemas

Opportunity Schema

Applicant Web Services Integration

June 5, 2009

Each grant opportunity that is published on Grants.gov has its own opportunity XML schema. An opportunity schema defines the required and optional form schemas for a single grant opportunity. Opportunity schemas can be obtained using the GetOpportunityList web service. For a sample of an opportunity schema, see [Appendix C](#).

Form Schema

A Form schema contains the actual data elements that comprise the grant application (e.g., SF- 424 and R&R). Form schemas are included by reference in the opportunity schemas. All Form schemas are published and available for download via the Grants.gov Resource Directory (<http://apply.grants.gov/system/MetaGrantApplication>). Grantee organizations should focus their development on the core set of Form schemas (e.g., SF-424 and R&R) that relate to their highest volume of grant applications.

Header Schema

The Header schema is required when submitting opportunities to Grants.gov. The schema can be accessed by the following URL <http://apply.grants.gov/system/schemas/Header-V1.0.xsd> The following is a list of the required elements:

- HashValue – The hash value of the Grant Application XML (computation shall be based on the “grant:Forms” sub-element in the Grant Application XML, this excludes the header and footer XML and attachments). Grants.gov currently supports the Secure Hash Algorithm (SHA-1) for computing the Hash value.
- OpportunityID – The opportunity ID for the Grant Application. This data is CASE SENSITIVE. To avoid any potential conflicts, applicant systems should utilize the opportunity meta-data returned by the GetOpportunityList web service when populating the Header schema XML.
- CompetitionID – The Competition ID for the Grant Application (Required only if available)
- CFDA Number – The CFDA Number associated with the Grant Application (Required only if available)

Attachment Schema

The Attachments schema is required when adding attachments to an electronic grant application package. Some form schemas may allow you to attach files for certain fields and some may even require attachments. When attaching a file to a form you must include the file details defined in the Attachments schema. The schema can be accessed by the following URL

<http://apply.grants.gov/system/schemas/Attachments-V1.0.xsd>

- FileName – The file name for that attachment
- MimeType – The mime content-type of the file
- FileLocation href attribute – This attribute should match the content-id that is assigned to the SOAP Attachment part. This allows the attachments to be associated in the grant form xml.
- HashValue – The hash value of the attachment.
 - hashAlgorithm – Attribute to indicate the algorithm used. Must be set to ‘SHA-1’ or the application will be rejected.

Change Management

To assist grantee organizations in the change management process of XML schemas Grants.gov will:

Applicant Web Services Integration

June 5, 2009

- Develop and enforce a change management policy on form schemas so that adequate leadtime is provided for applicant systems to comply.
- Provide a test environment for introducing new schemas and new versions of existing schemas.

Finding Opportunities on Grants.gov

Grantee organizations are responsible for obtaining the Opportunity Number (often referred to as Funding Number) from the find function on Grants.gov -

<http://www.grants.gov/FindGrantOpportunities?search=basic>

The CFDA number and Competition ID may be required if the opportunity includes them. The search may return opportunities that are not posted on Grants.gov. In other words, not all opportunities on Grants.gov are available on Find. To verify that the opportunity is available on Grants.gov, click on the link titled "Grant" beside the target opportunity. Next, scroll to the bottom of the opportunity and click the "Apply for Grant Electronically" button to view the confirmation of availability on Grants.gov.

Retrieving Grant Opportunities from Grants.gov

An opportunity XML schema defines the required and optional form schemas for a particular grant opportunity. The opportunity instructions provide the associated agency business rules for a particular opportunity. The GetOpportunityList function provides grantee organizations a webservice for retrieving the opportunity schema and instructions.

Schema Location Requirements in the Grants.gov XML

The use of xsi:schemaLocation attributes in XML submitted to Grants.gov shall conform to the rules identified in this section. Applicant organizations need NOT supply xsi:schemaLocations in the grants.gov XML. If an applicant submits XML to grants.gov containing an xsi:schemaLocation the URL must reside on the respective grants.gov servers (Acceptance Testing or Production) or the application will be rejected. The Form and System schema location

URL's are provided in the MetaGrantApplication schema

(<http://apply.grants.gov/system/schemas/applicant/MetaGrantApplication.xsd>). Opportunity schema URL's are returned in the GetOpportunityList web service. Applications containing DTD's will also be rejected.

Submitting Electronic Applications to Grants.gov

Preparing Grant Application XML

Grantee systems shall produce XML that conforms to Grants.gov Opportunity and Form schemas. The Grant Application XML shall pass strict XML schema validation defined by the Opportunity and Form schemas. Moreover, the opportunity may require additional business rules defined within the opportunity instructions. Opportunity schemas also require a Header XML document that conforms to the [Header schema](#).

The Grant Application XML shall be validated by the Grantee system prior to submitting the application to Grants.gov. The Grantee system shall include an implementation of the XML Cataloging specification (<http://www.oasis-open.org/committees/download.php/2384/cs-entityxml->

Applicant Web Services Integration

June 5, 2009

[catalogs-1.0.html](#)) that will resolve external URI requests for schemas to local schema files downloaded by the Grantee from the <http://apply.grants.gov/system/MetaGrantApplication> site.

Grants.gov Hashing Standard

Grants.gov uses the Secure Hash Algorithm (SHA-1) (<http://www.itl.nist.gov/fipspubs/fip180-1.htm>) for computing hash values. The resulting hash value shall be encoded using the Base64 data encoding specification (<http://www.ietf.org/rfc/rfc3548.txt>). The resulting value will be populated in the global schema 'HashValue' element. Grants.gov requires grantee organizations to hash the <grant:Forms> xml node and each SOAP attachment. When creating the attachment XML it is IMPORTANT to specify 'SHA-1' in the Global Schema 'glob:hashAlgorithm' attribute. If a value other than 'SHA-1' is received in the XML, Grants.gov will reject the application. The next few sections describe the process in greater detail.

Preparing and Hashing Attachments

The following sub-section is intended to provide clarification on sending attachments to Grants.gov via SOAP with Attachments. The Attachments schema is required when adding attachments to an electronic grant application package. The schema can be accessed by the following URL <http://apply.grants.gov/system/schemas/Attachments-V1.0.xsd>. The schema contains a field named 'FileLocation'. This element represents the Content-Id (CID) for the attachment. The href attribute of the 'FileLocation' element should be populated with the appropriate URI. Moreover, the CID in the schema should match the CID contained within the SOAP attachment header in the SOAP message. The only restriction is that the file names **must** be unique. The attachment shall be hashed using the [Grants.gov hashing standards](#) and the value placed in the 'HashValue' xml element in the attachment xml. For more information on Content-Id and SOAP attachments, please refer to the following links below:

- RFC 2111 - <http://www.ietf.org/rfc/rfc2111.txt>
- IBM Article - <http://www-106.ibm.com/developerworks/xml/library/x-tippass.html>

Hashing the Grant Forms XML

Once the Grant Application XML document is prepared, the Grantee organization shall compute using the [Grants.gov hashing standards](#). The hash value should be computed over the "grant:Forms" sub-element and not the entire application XML document. The <grant:Forms> element and its sub-elements shall be canonicalized before hashing to guarantee equivalence of hash values for "logically" equivalent Grant application XML documents. The canonicalization standard to use is the "Exclusive XML Canonicalization" W3C specification (<http://www.w3.org/TR/2001/REC-xml-c14n-20010315>) because the hash is taken over a subset of nodes. This specification, when applied, will produce XML that has the exact same lexical structure for all XML Node inputs that are "logically" equivalent. The specification will not include namespace specific attributes that are in ancestor Nodes of canonicalized sub-elements. Nor will it include the namespace declarations of ancestor Nodes that are not used by the sub elements to be canonicalized. This shields the canonicalized sub-elements from being affected by namespace declarations in ancestor Nodes that are not to be canonicalized.

There are three important points to take note of in the exclusive canonicalization specification:

Applicant Web Services Integration

June 5, 2009

1) white spaces between XML elements will not be normalized; 2) different namespace prefixes that are bound to the same namespace are not resolved; 3) comments in the XML are excluded. In the case of the first point, applicant and agency users must recognize that white spaces between XML elements are “retained” by canonicalization and thus will produce different hash values if white space values are changed. The hash value will also differ if namespace prefixes do not stay consistent between the times when the user created the hash value and when the Grants.gov WebServices consumer checks it.

The resulting hash value computed using the [Grants.gov hashing standard](#) shall be populated in the ‘HashValue’ element within the Header Schema. The entire XML document, including the Form schemas and Header Schema, shall be placed in the SOAP body for transport to Grants.gov.

Electronic Submission

Once grantee systems have prepared the SOAP message containing the Grant Application XML and any existing Attachments, the electronic application shall be submitted to Grants.gov by invoking the SubmitApplication web service. A successful transaction will receive a response that contains the Grants.gov tracking ID and a receipt date/time stamp. If an unexpected condition occurs, a SOAP fault will be returned containing a detailed error message. Validation of Electronic Submissions at Grants.gov will perform the following validation on all electronic submissions that are received via Grants.gov applicant web services. If the application fails to meet any of the validation rules below, the application will be assigned a rejected status. Grantee organizations shall use the GetApplicationStatusDetail web service to retrieve the descriptive reasons for the rejection. The service takes the Grants.gov tracking ID in the request, and returns a detailed message with the reasons for the rejection status.

- The system shall ensure that the application is a complete well-formed XML document.
- The system shall perform strict XML schema validation of the Grant Application XML against the opportunity XML schema.
- The system shall verify that the application is for a known Funding Opportunity Number (i.e., a Funding Opportunity Number previously registered on the Grants.gov system). The data values are CASE SENSITIVE. To avoid any potential conflicts, applicant systems should utilize the opportunity meta data returned by the GetOpportunityList web service when populating the Header schema XML. The data returned in the GetOpportunityList web service will always accurately reflect the data stored in the Grants.gov system.
- The system shall verify that the opportunity was submitted within the Opening and Closing timeframe
- The system shall verify that the DUNS number in the xml submission matches the DUNS number for the applicant organization (that is, the DUNS number of the submitter).
- The system shall verify the CFDA number(s) present on the application match known CFDA catalog numbers.
- The system shall verify the hash values for the grant:Forms and any attachments. The hashAlgorithm attribute must be set to ‘SHA-1’ or the application will be rejected.
- The system shall check any attached files to verify that they are virus free.

Tracking the Status of Electronic Grant Application Submissions

Applicant Web Services Integration

June 5, 2009

The GetApplicationList and GetApplicationStatusDetail web services provide grantee organizations a mechanism for verifying the status of electronic applications submitted to Grants.gov. The application information that is returned from web services at Grants.gov will reflect the applications submitted by the DUNS number(s) associated with the grantee organizations digital certificate. In other words, the grantee organization will only have access to view data about grant applications that were submitted by their system.

Filter Types for Querying GetApplicationList

Applications can be queried by using any combination of the following filtering criteria on the GetApplicationListRequest:

- Processing Status – The current processing status associated with the application
- CFDA Number – The Catalog of Federal Domestic Assistance number
- Opportunity Number – The opportunity number
- SubmissionTitle – SubmissionTitle xml element from the [header schema](#) (see [Uniquely Identifying Submissions](#) for more information)

Processing Status

The following is a list of application processing status in the Grants.gov system:

- Received – Grants.gov acknowledged receipt of the application
- Processing – Grants.gov is validating the application package and performing internal processes
- Validated – Grants.gov has validated the contents of the application submission
- Rejected with Errors – Grants.gov has rejected application submission. Grantee organizations shall use the GetApplicationStatusDetail web service to retrieve the descriptive reasons for the rejection
- Received by Agency – The Grant Making Agency has acknowledged receipt of the application from Grants.gov
- Agency Tracking Number Assigned -- The Grant Making Agency has assigned an internal tracking number

Uniquely Identifying Submissions

Grantee organizations are provided two ways to track electronic applications, the Grants.gov tracking ID and the 'SubmissionTitle' element. Grants.gov creates and returns the Grants.gov tracking ID anytime an electronic grant application is successfully submitted. The SubmitApplicationResponse message is sent in response to submitting an application via the SubmitApplication web service. The response message will contain the Grants.gov tracking ID. The 'SubmissionTitle' element contained in the Header schema may be used by grantee organizations to assign an organization specific tracking IDs to grant applications submitted to

Grants.gov. The unique submission tracking capability is purely optional and is provided as a convenience to the grantee organization; however the capability of uniquely identifying applications with a grantee tracking ID is particularly critical if the grantee system fails while processing a SubmitApplicationResponse message and loses the Grants.gov tracking ID. In this event, the grantee system could search the Grants.gov GetApplicationList web service with the 'SubmissionTitle' filter using the grantee tracking ID sent on the grant application header.

Applicant Web Services Integration

June 5, 2009

Retrieving Detailed Reasons for a Rejection

Grantee organizations shall use the GetApplicationStatusDetail web service to retrieve the descriptive reasons for the rejection. The service takes the Grants.gov tracking ID in the request, and returns a detailed message with the reasons for the rejection status.

Other Notification Mechanisms

Email is an additional mechanism for receiving notifications from Grants.gov. Whenever the application status changes, a notification email will be sent.

Accessing Grants.gov Test Environment

The system-to-system interface requires grantee organizations to develop a W3C compliant SOAP client to interact with the Grants.gov web services interface. Organizations interested in testing the Applicant web services may do so without the need to register with Grants.gov. The following two options for connecting are provided:

- Download and install the applicant web services reference implementation (<http://www.grants.gov/WebServices>). This tool contains everything needed to connect to the Grants.gov acceptance test server and execute web services. The reference implementation will also contain an implementation of the XML Cataloging specification provided by the Apache foundation.
- To test with an existing W3C compliant SOAP client, integrators may use the SSL client keystore provided with the reference implementation. The digital certificate contained in the SSL keystore authorizes applicant organizations to access the applicant web services. Since the certificate is shared with the entire community, the GetApplicationList webservice will return many applications.

5. Web Services Security

Mutual Authentication and SSL

Please refer to the Grants.gov [Applicant Web Services Security](#) document for in-depth information on security. Grants.gov uses Secure Socket Layer (SSL) and mutual authentication for managing the security of a message transmission over the Internet. Grants.gov requires both server-side and client-side authentication. Web Services need to be implemented securely; i.e., the data transferred over the Web Service needs to be safe from interception tampering, and unauthorized access. Some of the most common security threats to a distributed information system include:

- An authorized user of the system gaining access to information that should be hidden from him/her.
- A user masquerading as someone else, and obtaining access to whatever that second user is authorized to do. Actions will be attributed to the wrong person.
- Eavesdropping on a communication line, gaining access to confidential data.
- Tampering with communication between objects: modifying, inserting, and deleting items.
- Lack of accountability, due, for example, to inadequate identification of users. The most important problem is repudiation, meaning denial of responsibility for an action.

Applicant Web Services Integration

June 5, 2009

6. Grantee organization Options for Web Services Implementation

Grantee organizations may use a variety of Commercial Off The Shelf (COTS) and public domain tools and utilities to integrate with the Grants.gov application retrieval Web Services interfaces. Some of the more popular products are listed below:

Web Server Description

[Apache](#) (Apache) Apache has been the most popular web server on the Internet since April of 1996. The June 2003 [Netcraft Web Server Survey](#) found that 63% of the web sites on the Internet are using Apache, thus making it more widely used than all other web servers combined. [Internet Information Server](#) (Microsoft) The second most widely used Web Server today. IIS 6.0 leverages the latest Web standards such as Microsoft ASP.NET, XML, and Simple Object Access Protocol (SOAP) for the development, implementation, and management of Web applications.

Web Application Servers

Application servers offer developers an integrated Web development environment that allows them to connect and manage a variety of enterprise resources such as Web servers, databases, and legacy application systems. Application servers interface directly with the web server and the backend systems; they are where the business logic of web-based applications that access enterprise resources is embedded. The following tables list a few of the commercial and public domain application servers. See the [Application Server Comparison Matrix](#) provided by TheServerSide.com for a more comprehensive list.

7. Grants.gov Application Processing

Lifecycle of an Application

Listed below is the workflow to submit an electronic application to Grants.gov and then retrieval by the agencies system from the Grants.gov system:

- Applicant system creates an electronic application by populating an XML document that conforms to an opportunity schema published on Grants.gov and adding any attachments (if necessary).
- Applicant system constructs a SOAP message and invokes the SubmitApplication web service on the Grants.gov server to submit the application.
- Grants.gov server responds with a SOAP message containing the tracking ID and receipt date and time.
- Grants.gov system validates and processes the application
- The agency system invokes the Grants.gov web service to get a list of applications
- The Grants.gov web service responds with containing a list of applications with their Grants.gov tracking numbers.
- The agency system invokes the Grants.gov web service to download (pull) a specific application.
- The Grants.gov system responds with the application, including any native documents in binary format.
- The agency system invokes the Grants.gov web service confirming the successful retrieval (download) of an application.

Applicant Web Services Integration

June 5, 2009

- The agency system can optionally add an internal tracking ID.

8. Appendix A

Grants.gov Contact

Vincent Sprouls, Vincent.Sprouls@hhs.gov

Acronyms and Definitions

Acronym Description

AOR Authorized Organization Representative
CCR Central Contractor Registry
CFDA Catalog of Federal Domestic Assistance
COTS Commercial Off-the-Shelf
DUNS Data Universal Numbering System

Acronym Description

EFT Electronic Funds Transfer
GUI Graphical User Interface
HTTP Hyper Text Transfer Protocol
HTTPS Hyper Text Transfer Protocol Secure Sockets
IT Information Technology
J2EE Java 2 platform, Enterprise Edition. A Java platform designed for the intensive computing needs in large enterprises.
JAVA A general purpose, high-level, object-oriented, cross-platform programming language.
MIME Multipurpose Internet Mail Extensions
OMB Office of Management and Budget
PDF Portable Document Format
POC Point of Contact
SHA Secure Hash Algorithm
S2S System to System
SOAP Simple Object Access Protocol
SSL Secure Socket Layer
SWA Simple Object Access Protocol with Attachment
URI Uniform Resource Identifiers
W3C World Wide Web Consortium
WSDL Web Services Description Language

Acronym Description

XML eXtensible Markup Language
XSL eXtensible Stylesheet Language

Web Resources on SOAP, WSDL and Other Tools and Technologies

[SOAP] [SOAP \(Simple Object Access Protocol\) Version 1.1](#)

Applicant Web Services Integration

June 5, 2009

(<http://www.w3.org/TR/SOAP/>)

[XML] Extensible Markup Language (XML) 1.0 (<http://www.w3.org/TR/1998/REC-xml-19980210>)

[ebXML] Electronic Business using Extensible Markup Language (ebXML)

(<http://www.ebxml.org>)

[ebMS2] ebXML messaging specification 2.0 <http://www.ebxml.org/specs/ebMS2.pdf>

[MultipartRelated] The MIME Multipart/Related Content-type

(<http://www.ietf.org/rfc/rfc2387.txt>)

[MIME1] Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies (<http://www.ietf.org/rfc/rfc2045.txt>)

[CID] Content-ID and Message-ID Uniform Resource Locators

<http://www.ietf.org/rfc/rfc2111.txt>

[URI] Uniform Resource Identifiers (URI): Generic Syntax

(<http://www.ietf.org/rfc/rfc2396.txt>)

[RFC2557] MIME Encapsulation of Aggregate Documents, such as HTML (MHTML)

(<http://www.ietf.org/rfc/rfc2557.txt>)

[XMLBASE] XML Base <http://www.w3.org/TR/xmlbase>

[Hypertext Transfer Protocol -- HTTP/1.1 (<http://www.ietf.org/rfc/rfc2616.txt>)

Application Server Comparison Matrix

[WSDL] Web Services Description Language -- <http://www.w3.org/TR/wsdl>

9. Appendix B

Reference Implementation

The reference implementation provided by Grants.gov is a complete client/server example to demonstrate how Applicants can submit applications to Grants.gov via web services and J2EE. It contains the complete source code, detailed documentation, and self-signed digital certificates for the testing purposes. Organizations utilizing the Java platform are free to use the code as a starting point or a complete SOAP client solution. The reference implementation can be downloaded at <http://www.grants.gov/WebServices>

Dynamic Binding to the Grants.gov Applicant WSDL

Document Style Web Services are loosely coupled with the server-side components. The client sends the parameter to the Web Service as an XML document, instead of discrete set of parameter values in RPC style. The Web Service processes the document, executes the operation and constructs and sends the response to the client as an XML document. Thus there is no direct mapping between the server objects (parameters, method calls etc) and the values in XML documents that prevents dynamic binding to the agency WSDL.

Static Binding to the Grants.gov Applicant WSDL

Grants.gov utilizes static WSDL binding in the applicant system-to-system reference implementation. WSDL2Java is an Apache tool used at Grants.gov for building Java proxies and skeletons from the Applicant WSDL.

10. Appendix C

Applicant Web Services Integration

June 5, 2009

Sample Opportunity Schema

URL:

http://apply.grants.gov/opportunities/schemas/oppNSF-1008-RR_Package-cfda10.100-cid1234.xsd

XSD File:

```
<?xml version="1.0"?>
<xsd:schema
targetNamespace="http://apply.grants.gov/system/MetaGrantApplication"
xmlns:grant="http://apply.grants.gov/system/MetaGrantApplication"
xmlns:header="http://apply.grants.gov/system/Header-V1.0"
xmlns:footer="http://apply.grants.gov/system/Footer-V1.0"
xmlns:att="http://apply.grants.gov/system/Attachments-V1.0"
xmlns:RR_SF424="http://apply.grants.gov/forms/RR_SF424-V1.0"
xmlns:RR_PersonalData="http://apply.grants.gov/forms/RR_PersonalData-V1.0"
xmlns:RR_PerformanceSite="http://apply.grants.gov/forms/RR_PerformanceSite-V1.0"
xmlns:RR_OtherProjectInfo="http://apply.grants.gov/forms/RR_OtherProjectInfo-V1.0"
xmlns:RR_KeyPerson="http://apply.grants.gov/forms/RR_KeyPerson-V1.0"
xmlns:RR_Budget="http://apply.grants.gov/forms/RR_Budget-V1.0"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified"
attributeFormDefault="qualified">
<!--System Data Types-->
<xsd:import namespace="http://apply.grants.gov/system/Header-V1.0"
schemaLocation="http://apply.grants.gov/system/schemas/Header-V1.0.xsd"/>
<xsd:import namespace="http://apply.grants.gov/system/Footer-V1.0"
schemaLocation="http://apply.grants.gov/system/schemas/Footer-V1.0.xsd"/>
<xsd:import namespace="http://apply.grants.gov/system/Attachments-V1.0"
schemaLocation="http://apply.grants.gov/system/schemas/Attachments-V1.0.xsd"/>
<!--Included Mandatory Forms-->
<xsd:import namespace="http://apply.grants.gov/forms/RR_SF424-V1.0"
schemaLocation="http://apply.grants.gov/forms/schemas/RR_SF424-V1.0.xsd"/>
<!--Included Optional Forms-->
<xsd:import namespace="http://apply.grants.gov/forms/RR_PersonalData-V1.0"
schemaLocation="http://apply.grants.gov/forms/schemas/RR_PersonalData-V1.0.xsd"/>
<xsd:import namespace="http://apply.grants.gov/forms/RR_PerformanceSite-V1.0"
schemaLocation="http://apply.grants.gov/forms/schemas/RR_PerformanceSite-V1.0.xsd"/>
<xsd:import namespace="http://apply.grants.gov/forms/RR_OtherProjectInfo-V1.0"
schemaLocation="http://apply.grants.gov/forms/schemas/RR_OtherProjectInfo-V1.0.xsd"/>
<xsd:import namespace="http://apply.grants.gov/forms/RR_KeyPerson-V1.0"
schemaLocation="http://apply.grants.gov/forms/schemas/RR_KeyPerson-V1.0.xsd"/>
Applicant Web Services Integration 32
<xsd:import namespace="http://apply.grants.gov/forms/RR_Budget-V1.0"
schemaLocation="http://apply.grants.gov/forms/schemas/RR_Budget-V1.0.xsd"/>
<xsd:element name="GrantApplication">
```

Applicant Web Services Integration

June 5, 2009

```
<xsd:complexType>
<xsd:sequence>
<xsd:element ref="header:GrantSubmissionHeader"/>
<xsd:element name="Forms">
<xsd:complexType>
<xsd:all>
<!--Included Mandatory Forms-->
<xsd:element ref="RR_SF424:RR_SF424"/>
<!--Included Optional Forms-->
<xsd:element ref="RR_PersonalData:RR_PersonalData" minOccurs="0"/>
<xsd:element ref="RR_PerformanceSite:RR_PerformanceSite" minOccurs="0"/>
<xsd:element ref="RR_OtherProjectInfo:RR_OtherProjectInfo" minOccurs="0"/>
<xsd:element ref="RR_KeyPerson:RR_KeyPerson" minOccurs="0"/>
<xsd:element ref="RR_Budget:RR_Budget" minOccurs="0"/>
</xsd:all>
</xsd:complexType>
</xsd:element>
<xsd:element ref="footer:GrantSubmissionFooter"/>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
</xsd:schema>
```