ANSI/NISO Z39.93-2013

ISSN: 1041-5653

The Standardized Usage Statistics Harvesting Initiative (SUSHI) Protocol

Abstract: This standard defines an automated request and response model for the harvesting of electronic resource usage data utilizing a Web services framework that can replace the user-mediated collection of usage data reports. It was designed as a generalized protocol extensible to a variety of usage reports. An extension designed specifically to work with COUNTER reports is provided.

The standard is built on SOAP (Simple Object Access Protocol) for transferring request and response messages. The `GetReport` method is used for transferring `ReportRequest` as the input message and returning `ReportResponse` as the output message.

The standard includes a versioned Web Services Description Language (WSDL) to describe the Web service namespace and operations, and an XML schema constraining the syntax of the SUSHI transaction. Rules for report naming are outlined and complemented by an external reports registry, which provides for the definition of both COUNTER and non-COUNTER reports.

An American National Standard
Developed by the
National Information Standards Organization

Approved February 20, 2013
by the
American National Standards Institute

NISO
How the information world
CONNECTS

About NISO Standards

NISO standards are developed by the Standards Working Groups of the National Information Standards Organization. The development process is a strenuous one that includes a rigorous peer review of proposed standards open to each NISO Voting Member and any other interested party. Final approval of the standard involves verification by the American National Standards Institute that its requirements for due process, consensus, and other approval criteria have been met by NISO. Once verified and approved, NISO Standards also become American National Standards.

These standards may be revised or withdrawn at any time. For current information on the status of this standard contact the NISO office or visit the NISO website at: www.niso.org.

Published by
National Information Standards Organization (NISO)
3600 Clipper Mill Road
Suite 302
Baltimore, MD 21211
www.niso.org

ISSN: 1041-5653 (National information standards series)

ISBN: 978-1-937522-13-1

Contents

Tables

Figures

Foreword

(This foreword is not part of ANSI/NISO Z39.93-2013, *The Standardized Usage Statistics Harvesting Initiative (SUSHI) Protocol*. It is included for information only.)

About This Standard

This standard was developed in response to the need in libraries to more efficiently collect COUNTER (Counting Online Usage of NeTworked Electronic Resources) compliant usage statistics. COUNTER was launched in March 2002 as an international initiative to help librarians and publishers in the recording and exchange of usage statistics for electronic resources. By following COUNTER's *Code of Practice*, vendors can provide library customers with Excel or CSV (comma delimited) files of usage data using COUNTER's standardized formats and data elements. As of September 2006, COUNTER has nearly 200 members and over 40 vendors have certified compliance to one or more *Code of Practice* releases.

COUNTER Usage Difficulties

The success of COUNTER created a new problem for librarians—the amount of time they spend retrieving, storing, and aggregating their COUNTER reports. For each supplier of COUNTER reports, a library must manually connect to the supplier's website and download the Excel or CSV files. Once retrieval is complete, the librarian is presented with multiple data files from multiple providers, which, while standardized, still require significant manipulation and normalization of the data to aggregate reports. Many libraries are creating or purchasing electronic resource management (ERM) systems to help them in storing and managing all this data. But the process for transferring the Excel/CSV data into the repository is either manual or requires custom programming.

Background on SUSHI Development

The idea for an automated method to solve the COUNTER report problem was first discussed by Adam Chandler and Ted Fons at the 2004 Charleston Conference. Fons is Product Manager for the Innovative Interfaces Inc. (III) Electronic Resource Management module. Chandler is Coordinator, Service Design Group, Information Technology and Technical Services at Cornell University Library, and is active in the Digital Library Federation's Electronic Resource Management Initiative (ERMI). The two agreed to push forward a Web service protocol for handling the XML version of the COUNTER reports. Tim Jewell (University of Washington) and Oliver Pesch (EBSCO) were recruited at the American Library Association meeting in June 2005 to further refine the concept. A project team was formed and from there the project was expanded to include the members listed below. Draft requirements and specifications for the protocol began in July 2005. In October 2005, NISO agreed to sponsor the initiative, the group was officially named SUSHI (Standardized Usage Statistics Harvesting Initiative), and work began on creating this standard defining the protocol.

The SUSHI Web Service

The Standardized Usage Statistics Harvesting Initiative (SUSHI) represents a "Web services" approach to solving the COUNTER retrieval and consolidation problem. The protocol is a SOAP (Simple Object Access Protocol) request/response Web services "wrapper" for the XML version of COUNTER reports. In the protocol, a transaction begins when a client service running as part of an application developed by a library—or running as part of a usage data consolidation service or ILS / ERM system—identifies itself, identifies the customer whose statistics are being requested, and specifies the desired report to the SUSHI server service running at a data provider. In response, the server provides the report in XML format, along with the requestor and customer information—or an appropriate error message.

The SUSHI developers envision a system in which the client system is programmed to automatically retrieve reports on a monthly schedule for all the COUNTER-compliant vendors with which the library does business. The ability for the client to manually trigger requests may also be desirable to allow for easier testing with new SUSHI implementations and for retrieving previous months' usage data.

Although developed primarily for COUNTER reports, SUSHI was written as a generalized protocol that accommodates extensions for customized non-COUNTER usage reports. The COUNTER-specific schema extension is provided to accompany the general protocol.

Trial Use

The SUSHI protocol was issued as a Draft Standard for Trial Use from September 20, 2006 through May 20, 2007. During the trial period, dozens of implementations of both the client or server sides of the protocol were successfully conducted. Minor revisions to the schemas were made to address issues identified during the trial. The current standard reflects those changes. The schema version at the time of the trial was 1.0; the schema version at the time of publication of this standard was 1.5. The most current version of the schema is maintained on the SUSHI schema website (http://www.niso.org/schemas/sushi/).

2013 Revision

At the time the standard was due for its five-year reaffirmation review, the SUSHI Standing Committee recommended that a revision be issued to make two minor changes:

- Addition to Table 17 of a new error code 1020 when a client exceeds the number of allowable requests to a server in a particular timeframe. This accommodates the need of some SUSHI server providers to set limits on access in order to manage server performance.
- Revised informative Appendix G on Security Considerations. Although security is not part of the standard protocol, security was considered in developing the standard and this appendix provides recommendations on security-related issues. It was updated to reflect technology changes and experience gained since the initial implementation of the SUSHI protocol.

Trademarks, Service Marks

Wherever used in this standard, all terms that are trademarks or service marks are and remain the property of their respective owners.

NISO Voting Members

At the time this standard was balloted, the following organizations were members of the NISO Voting Pool that approved this standard. The individuals listed were the voting representatives who voted on the ballot.

American Institute of Physics (AIP)
Terry Hulbert

American Library Association (ALA)
Nancy Kraft

American Psychological Association (APA)
Janice Fleming

American Society for Information Science & Technology (ASIS&T)
Mark Needleman

Cengage Learning
Marti Heyman

EBSCO Information Services
Oliver Pesch

Emerald Publishing Group
Dan Prendergast

Ex Libris, Inc.
Mike Dicus

Inera Inc.
Bruce Rosenblum

Innovative Interfaces, Inc.
John McCullough

ITHAKA/JSTOR/Portico
Amy Kirchhoff

John Wiley & Sons, Ltd.
Keith Webster

Library of Congress
John Zagas

Los Alamos National Laboratory
Miriam Blake

Lyrasis
Peter Murray

Minitex
Cecelia Boone

Music Library Association
Mark McKnight

National Archives and Records Administration
Marilyn Redman

National Library of Medicine (NLM)
Barbara Rapp

Polaris Library Systems
Eric Graham

Ringgold, Inc.
Donald Chvatal

Scholarly iQ
John Milligan

Serials Solutions
Ashley Bass

The Library Corporation (TLC)
Juli Marsh

NISO Business Information Topic Committee

The following individuals served on the Business Information Topic Committee at the time it approved this standard:

Karla Strieb, Chair
Ohio State University Libraries

Ivy Anderson
California Digital Library

Niels Dam
Reed Elsevier

Denise Davis
Sacramento Public Library

Herbert Gruttemeier
Institut de l'Information Scientifique et Technique

William Hoffman
Swets Information Services

Norm Medeiros
Haverford College

Gary Van Overborg
Scholarly iQ

Christine Stamison
Swets Information Services

SUSHI Standing Committee Members

The following individuals were members of the SUSHI Standing Committee that developed and approved this revision to the standard:

Bob McQuillan, Co-Chair
Innovative Interfaces, Inc.

Oliver Pesch, Co-Chair
EBSCO Information Services

Marie Kennedy
Loyola Marymount University

John Milligan
Scholarly iQ

Paul Needham
Cranfield University

James Van Mil
University of Cincinnati Libraries

Chan Li
California Digital Library

Acknowledgments

The Working Group members were assisted by the following individuals during the initial development of the 2007 standard:

Ben Burbridge
Innovative Interfaces, Inc.

Matthew Connolly
Cornell University Library

Cynthia Hodgson
NISO

Curt Kohler
Elsevier

Tony Li
Innovative Interfaces, Inc.

Joshua Santelli
Project Euclid

Rolf van der Tang
Swets Information Services

Dennis Vaux
Innovative Interfaces, Inc.

Petar Vucetin
EBSCO Information Services

James Wismer
Thomson Scientific

The Standardized Usage Statistics Harvesting Initiative (SUSHI) Protocol

1 Purpose

The SUSHI protocol is designed to provide an automated method for retrieving standardized usage statistics reports using a machine-processable XML container. The protocol utilizes the Web services Simple Object Access Protocol (SOAP).

2 Scope

The SUSHI protocol is generalized to accommodate any customized usage reports that conform to the protocol's requirements. Specific reports are addressed through the use of an extension to the general protocol. Since the original intent of the protocol was to work with COUNTER reports, a COUNTER-specific schema extension is provided with the general protocol. (See Appendix C for the COUNTER-SUSHI schema.)

The SUSHI protocol is a transport mechanism for usage reports and is not a repository for the retrieved usage data. This standard does not specify or define the repository for the usage data; therefore, the XML formatted data can be collected in spreadsheets, in-house databases, commercial electronic resource management (ERM) systems, or any other repository that the user establishes for that purpose.

3 References

This standard references the following documents:

The COUNTER Code of Practice for Journals and Databases. Edinburgh, UK: COUNTER. Release 2, published April 2005; valid from January 1, 2006. Available from: http://www.projectcounter.org/code_practice.html

The COUNTER Code of Practice for Books and Reference Works. Edinburgh, UK: COUNTER. Release 1, March 2006. Available from: http://www.projectcounter.org/code_practice.html

COUNTER Register of Vendors. http://www.projectcounter.org/articles.html

The Unicode Consortium, *The Unicode Standard*, Version 5.0.0. Boston, MA, Addison-Wesley, 2007. ISBN 0-321-48091-0. Online version available from: http://www.unicode.org/versions/Unicode5.0.0/

W3C Note-datetime, *Date and Time Formats*. http://www.w3.org/TR/NOTE-datetime

W3C Recommendation. *Simple Object Access Protocol (SOAP),* version 1.2. World Wide Web Consortium, June 24, 2003. Available from: http://www.w3.org/TR/soap/

W3C Recommendation. *Extensible Markup Language (XML)*, version 1.0 (Third Edition). World Wide Web Consortium, February 4, 2004. http://www.w3.org/TR/2004/REC-xml-20040204/

4 Definitions

The following terms, as used in this standard, have the meanings indicated.

Term	Definition
client	A computer system that connects over a network to a SUSHI server in order to retrieve usage statistics.
COUNTER (**C**ounting **O**nline **U**sage of **N**etworked **E**lectronic **R**esources)	An international initiative to facilitate the recording and exchange of online usage statistics. As used in this standard, refers to the reports defined by the initiative.
electronic resources	Digital information sources (e.g., online journal issues or databases) accessible through a networked client computer.
ERM (Electronic Resource Management [system])	A database and software system dedicated to handling the cataloging, licensing, accessing, and other aspects of electronic resource use.
GUID (Global Unique Identifier)	An identifier for an entity that is guaranteed to be unique within a global namespace, with a goal of unambiguously identifying that entity regardless of the context of the system or vendor.
ILS (Integrated Library System)	A set of automated library services—such as an online catalog, circulation management, and serial tracking—that share a common database.
release	A version of a COUNTER Code of Practice and the reports defined within that Code.
request	A SOAP message sent from a client to a server requesting usage statistics for a specified customer and a desired report type.
requestor	An entity requesting usage statistics on behalf of a customer. The requestor may be, but is not necessarily, the customer itself.
response	A SOAP message sent from a server to a client, containing the usage statistics for the customer specified in a request.
server	A computer system that responds to SUSHI SOAP messages and supplies usage statistics to a client over a network.
SOAP (Simple Object Access Protocol)	A protocol that specifies a self-contained messaging system used to exchange data and access services across a network (commonly the Internet).
transaction	A complete SUSHI exchange, comprised of a client-to-server request and a server-to-client response.
usage statistics	Reports detailing the use of a customer's electronic resources over a given period of time.

Term	Definition
Web service	A server-based software system that provides information or functionality in response to XML-based messages sent through the Internet.
WSDL (Web Service Description Language)	An XML protocol that completely defines the functionality and data provided by a Web service, along with the message formats and input data required to use the service.
XML (Extensible Markup Language)	A protocol for coding text or data with meaningful information in the form of enclosing tags.

5 Element Reference Guide

5.1 Element Listings

Section 6 of this standard defines the XML schema elements of the SUSHI protocol. For each element, a table of attributes is provided. Table 1 defines each of the columns used in Section 6 to describe the schema elements.

Table 1: Description of Element Tables

Element	Name of the element.
Type	Specifies the type of data allowed. See Section 5.2 for the list of types.
Req'mt (Requirement)	M = mandatory MA = mandatory if applicable R = recommended O = optional
Syntax	Describes the required syntax of the attribute variable.
Value Examples	Provides one or more examples of how a valid attribute would appear.

5.2 Data Types

The allowable data types for the elements described in this standard are listed and defined in Table 2. All data should conform to the Unicode UTF-8 character set.

Note: UTF-8 is the byte-oriented encoding form of Unicode.

Table 2: Data Type Definitions

Data Type	Definitions
AnyURI	Any valid Uniform Resource Identifier (URI).
AnyType	Any data type; no restrictions.
ComplexType	A type made by combining elements.

Data Type	Definitions
Date	Date in the granularity of year-month-day recorded in compliance with the *W3C Note-datetime*. The format is: YYYY-MM-DD
DateTime	Date and time recorded in compliance with the *W3C Note-datetime* in the format: YYYY-MM-DDThh:mm:ss.sTZD (with hours 0-24, and where the Time Zone Designator is set to the constant "Z" to reflect that all times are expressed in UTC).
Enumerated	Values must come from an enumerated list that is specified in the Syntax column of the Element tables.
Integer	A positive integer.
String	One or more alphanumeric characters.
sushi:Exception	A complex type defined by the SUSHI schema.

6 SUSHI Protocol

The SUSHI protocol consists of an XML schema and a WSDL (Web Services Description Language) that represent the data contract and service contract between client and server operating in a business-to-business environment. The Data Contract (XML Schema) consists of the `ReportRequest` and `ReportResponse`. These constructs are discussed in more detail in Section 6.2. Section 6.3 covers the Service Contract (the WSDL) and defines the functions supported by the SUSHI protocol.

6.1 Namespace

Table 3 illustrates how the XML messages within the SUSHI protocol will declare the schema and its namespace.

Table 3: SUSHI Protocol Header

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:sushi="http://www.niso.org/schemas/sushi/1_5"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.niso.org/schemas/sushi/1_5"
elementFormDefault="qualified" attributeFormDefault="unqualified"
version="1.5">
```

The SUSHI data contract declares the name space at the beginning of the schema document. The namespace is used to separate data types from different schemas. The SUSHI namespace is:

`http://www.niso.org/schemas/sushi/[version no.]`

The `version` element, [version no.], following the namespace designates the version of the `sushi.xsd` in use. Since the protocol uses fully qualified type names, the version is preceded with `elementFormDefault="qualified"`.

6.2 Data Contract

The Data Contract represents the data elements that are passed in a SOAP request response pattern when interacting with service operations. It consists of the *ReportRequest* (Section 6.2.1), *ReportResponse* (Section 6.2.2), and *Exceptions and Errors* (Section 6.2.3). The version of the XML schema that was current at the time of publication is reproduced in Appendix A.

6.2.1 ReportRequest

`ReportRequest` (Table 4) is a set of service input parameters describing the `Requestor` (Section 6.2.1.1), `CustomerReference` (Section 6.2.1.2), and `ReportDefinition` (Section 6.2.1.3).

The `Requestor` is intended to identify the organization requesting usage information for the customer. Customer information is passed via the `CustomerReference` element. Details about which report and data filter to apply are passed in the `ReportDefinition` element.

The variable elements of the `ReportRequest` are defined in Table 5. The entire `ReportRequest` element, including the subparts described in Sections 6.2.1.1 – 6.2.1.3, is illustrated in Figure 1.

Table 4: SUSHI Protocol ReportRequest Schema

```
<xs:complexType name="ReportRequest">
    <xs:sequence>
        <xs:element name="Requestor" type="sushi:Requestor">
        </xs:element>
        <xs:element name="CustomerReference" type="sushi:CustomerReference">
        </xs:element>
        <xs:element name="ReportDefinition" type="sushi:ReportDefinition">
        </xs:element>
    </xs:sequence>
    <xs:attribute name="Created" type="xs:dateTime" use="required"/>
    <xs:attribute name="ID" type="xs:string" use="required"/>
</xs:complexType>
```

Table 5: Elements for `ReportRequest`

Element	Type	Req'mt	Syntax	Value Example(s)
`Created`	DateTime	M	YYYY-MM-DD Thh:mm:ss.sZ	2001-12-17T09:30:47.0Z
`ID`	String	M	Any unique ID.	7e36aaf7-ee8d-4364-97b5-e06daabb362b

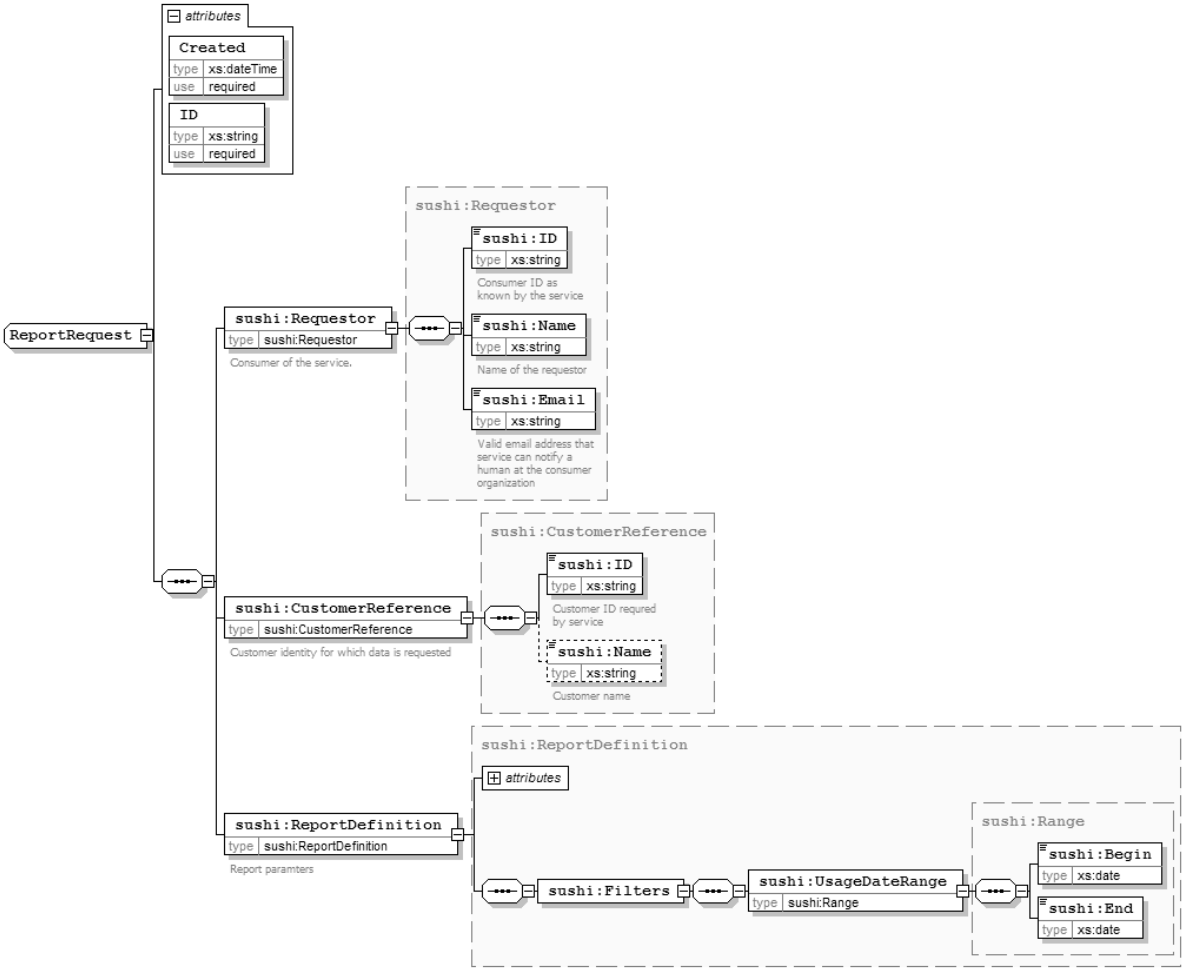

Figure 1: Diagram of SUSHI `ReportRequest`

6.2.1.1 Requestor

The `Requestor` element (Table 6) is used by the service to control access to customer usage statistics. The `Requestor` is used to identify the organization making the request, which may not be the same organization as the customer for whom the statistics are being harvested. The `Requestor ID` plays a central role in securing a SUSHI transaction. At their option, Service Providers can limit access to only known `Requestor IDs` or even known `Requestor IDs` that a customer has approved. (More information on securing SUSHI is provided in Appendix G).The variable elements of the `Requestor` are defined in Table 7.

NOTE: In cases where the requestor is also the customer (e.g., the customer has a locally hosted usage consolidation application), the `Requestor ID` could be the same as the `CustomerReference ID`.

Table 6: SUSHI Protocol Requestor Schema

```
<xs:complexType name="Requestor">
    <xs:sequence>
        <xs:element name="ID" type="xs:string">
        </xs:element>
        <xs:element name="Name" type="xs:string">
        </xs:element>
        <xs:element name="Email" type="xs:string">
        </xs:element>
    </xs:sequence>
</xs:complexType>
```

Table 7: Elements for Requestor

Element	Type	Req'mt	Syntax	Value Example(s)
ID	String	M	A valid unique identifier string assigned by the service provider.	60099445-3B1C-4862-8B15-B2EB72E6BE4B
Name	String	R	Internationally recognized organization name.	ScholarlyStats Cornell University Library
Email	String	R	Standard internet e-mail format of "string.domain". Can be a list of e-mails delimited with a semicolon.	sushiSupport@cornell.edu

NOTE 1: Service providers are encouraged to issue globally unique Requestor IDs (such as a GUID, for example: 48D537C3-B240-4d58-ABC9-BA052FFADB5F).

NOTE 2: Use of the Name and Email attributes in the Requestor element is optional but it is strongly encouraged for clients to send this information as it can be helpful in troubleshooting problems or sending automated e-mails. It is recommended that the e-mail address be of a service administrator who can facilitate technical as well as data problems.

6.2.1.2 CustomerReference

The CustomerReference element (Table 8) contains the identification information about the customer that is necessary in order for the customer to be recognized by the service provider to allow generation of the specified report. The actual mechanism for creating and providing customer IDs is the responsibility of the service provider. The variable elements of the CustomerReference are defined in Table 9.

Table 8: SUSHI Protocol CustomerReference Schema

```
<xs:complexType name="CustomerReference">
    <xs:sequence>
        <xs:element name="ID" type="xs:string">
        </xs:element>
        <xs:element name="Name" type="xs:string" nillable="true" minOccurs="0">
        </xs:element>
        </xs:sequence>
</xs:complexType>
```

Table 9: Elements for `CustomerReference`

Element	Type	Req'mt	Syntax	Value Example(s)
`ID`	String	M	Any token or identifier recognized by the service provider that will identify the customer within the service provider's system.	12345 JBC12345
`Name`	String	R	Internationally recognized organization name.	Cornell University Library
NOTE: Although the `Name` of the customer is not required, clients are strongly encouraged to provide the internationally known name of the organization for which data is requested. This information is useful when troubleshooting problems.				

6.2.1.3 ReportDefinition

The `ReportDefinition` element (Table 10) describes which report the client is requesting, using the `Name` and `Release` attributes of the element as well as the range of filters to apply for the requested usage data. See Section 7 for requirements related to report names and release numbers. The variable elements of the `ReportDefinition` are defined in Table 11.

Only one kind of filter is currently supported by the protocol and that is the `UsageDateRange` filter. The variable elements of the `UsageDateRange` are defined in Table 12. This filter defines the beginning and the end of the date range for the requested usage information.

If a service provider does not support the requested report name or release (version), the service must send an exception back to the client (see Section 6.2.3).

Table 10: SUSHI Protocol `ReportDefinition` Schema

```
<xs:complexType name="ReportDefinition">
    <xs:sequence>
        <xs:element name="Filters">
            <xs:complexType>
                <xs:sequence>
                    <xs:element name="UsageDateRange" type="sushi:Range"/>
                </xs:sequence>
            </xs:complexType>
        </xs:element>
    </xs:sequence>
    <xs:attribute name="Name" type="xs:string" use="required">
    </xs:attribute>
    <xs:attribute name="Release" type="xs:string" use="required"/>
</xs:complexType>
```

Table 11: Elements for ReportDefinition

Element	Type	Req'mt	Syntax	Value Example(s)
Filters	Complex Type	M	Varies as defined by the individual filters.	See Table 12.
Name	String	M	Valid Report Name (see Section 7).	JR1 DB2
Release	String	M	Valid Release Numbers of the Usage Report (see Section 7).	1 2
NOTE: COUNTER uses the terminology of “release” to indicate a version number.				

Table 12: Elements for UsageDateRange

Element	Type	Req'mt	Syntax	Value Example(s)
Begin	Date	M	YYYY-MM-DD	2003-07-01
End	Date	M	YYYY-MM-DD	2004-07-31
NOTE 1: The UsageDateRange is inclusive of the specified begin and end dates. NOTE 2: There is no restriction on the span of dates from the Begin to End values. If a service provider does not support the requested date range or cannot supply all requested data, the service will respond with the data it can provide and must send a warning exception back to the client.				

6.2.2 ReportResponse

After a service receives and processes the request, the service must send a ReportResponse (Table 13) element that includes a repeat of Request elements as well as an XML fragment that should be a strongly typed payload of the Report element. The SUSHI schema provides a template report response in the ReportResponse element. This template implements a generic Report element that can return any type of usage report. The service must implement the ReportResponse element by extension, overriding the Report element with either a strongly typed payload or by leaving the anyType definition if the service desires to have open-ended, multi-payload response.

The COUNTER-SUSHI schema extension is an example of a strongly typed payload for delivering COUNTER reports.

The COUNTER payload schemas are created to correspond with the COUNTER *Code of Practice* releases. These payload schemas can be found at: http://www.niso.org/schemas/sushi/index.html#COUNTER.

Figure 2 illustrates the ReportResponse element of the SUSHI schema and Table 14 describes the variable elements of the ReportResponse.

Table 13: SUSHI Protocol ReportResponse Schema

```
<xs:complexType name="ReportResponse">
        <xs:sequence>
            <xs:element name="Exception" type="sushi:Exception" nillable="true"
minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="Requestor" type="sushi:Requestor"/>
            <xs:element name="CustomerReference" type="sushi:CustomerReference"/>
            <xs:element name="ReportDefinition" type="sushi:ReportDefinition"/>
            <xs:element name="Report" type="xs:anyType" block="substitution"
form="qualified" nillable="true"/>
        </xs:sequence>
        <xs:attribute name="Created" type="xs:dateTime" use="required"/>
        <xs:attribute name="ID" type="xs:string" use="required"/>
      </xs:complexType>
```

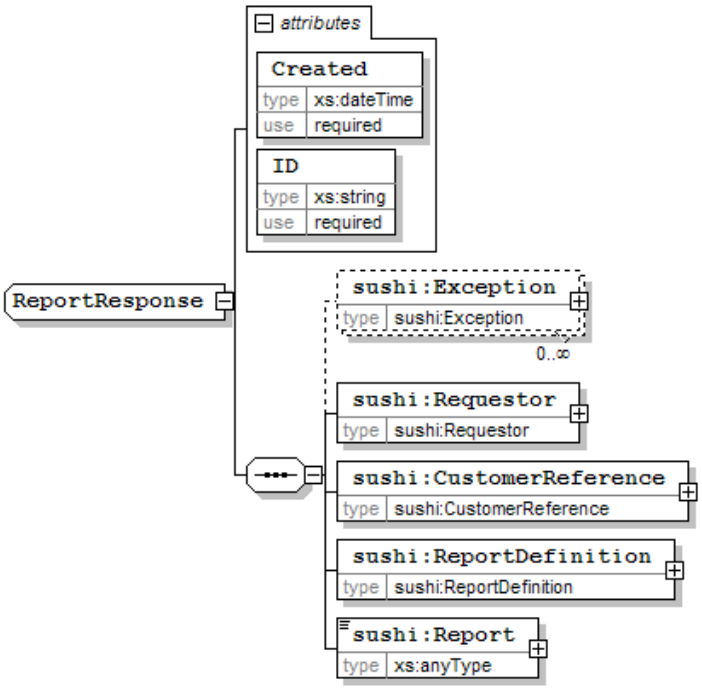

Figure 2: Diagram of SUSHI ReportResponse

NOTE: `Requestor`, `CustomerReference`, and `ReportDefinition` elements are echoed back from the `ReportRequest`.

Table 14: Elements for ReportResponse

Element	Type	Req'mt	Syntax	Value Example(s)
Created	DateTime	M	YYYY-MM-DD Thh:mm:ss.sZ	2006-10-17T09:30:47.0Z
ID	String	M	A valid unique identifier string repeated from the ReportRequest.	60099445-3B1C-4862-8B15-B2EB72E6BE4B
Exception	sushi:Exception	MA	See Section 6.2.3, *Exceptions and Errors*.	
Report	AnyType	M	Any valid XML.	Report element can be null.

6.2.3 Exceptions and Errors

Exception (Table 15) is a mandatory element, when applicable, that must be sent in the ReportResponse message to provide explanatory information to the client in the event that the server encounters an error. Server applications will check for errors and inconsistencies in the ReportRequest and report these inconsistencies in the Exceptions element of the ReportResponse.

Table 16 defines the elements that make up the Exception element, and Figure 3 graphically illustrates the element.

The list of exceptions that may be reported by a SUSHI server when processing a request is in Table 17.

Table 15: SUSHI Protocol Exception Reporting Schema

```
<xs:complexType name="Exception">
    <xs:sequence>
        <xs:element name="Number" type="xs:int">
        </xs:element>
        <xs:element name="Severity" type="sushi:ExceptionSeverity">
        </xs:element>
        <xs:element name="Message" type="xs:string">
        </xs:element>
        <xs:element name="HelpUrl" type="xs:anyURI" nillable="true" minOccurs="0">
        </xs:element>
        <xs:element name="Data" type="xs:anyType" nillable="true" minOccurs="0">
        </xs:element>
    </xs:sequence>
    <xs:attribute name="Created" type="xs:dateTime" use="required"/>
</xs:complexType>
```

Table 16: Elements for Exception

Element	Type	Req'mt	Syntax	Value Example(s)
Created	DateTime	M	YYYY-MM-DD Thh:mm:ss.sZ	2006-10-17T09:30:47.0Z
Number	Integer	M	A one- to four-digit integer.	2000
Severity	Enumerated	M	Info, Debug, Warning, Error, Fatal	
Message	String	M	Any valid string.	Access denied. Account '60099445-3B1C-4862-8B15-B2EB72E6BE4B' cannot access Resource 'B03BA8AD-CF3C-4E10-ABA6-2963F46E7BD0' for customer 'Cornell University Library'.
HelpUrl	AnyURI	R	A help URL.	http://sushi.ebsco.com/somehelppage.html
Data	AnyType	O	Any standard or custom type.	

NOTE 1: The Severity types are defined as follows:

Info – provides more information about the response.

Debug – provides debugging information about the response.

Warning – does not interrupt the completion of the transaction, but gives more information about the response.

Error – server unable to process request due to problem with request – returns no payload.

Fatal – server processing failed – returns no payload.

NOTE 2: The Data element is mostly used for debugging if the server needs to send more data to the client, e.g., a stack trace.

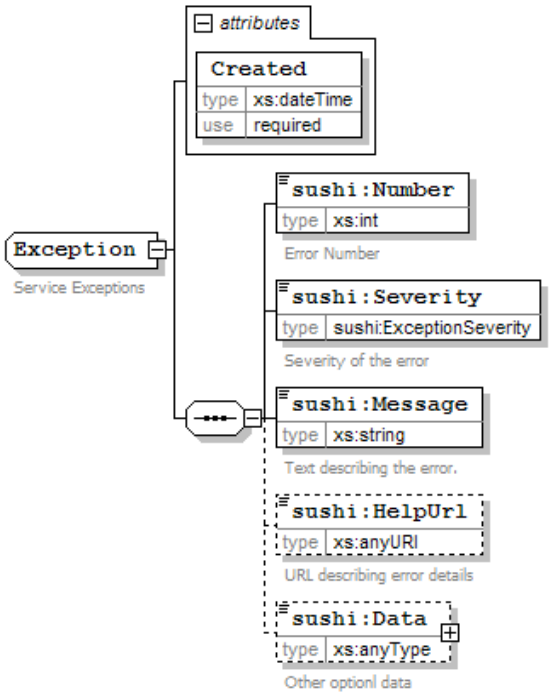

Figure 3: Diagram of SUSHI Exception Reporting

Table 17: Standard SUSHI Exceptions

Exception	Severity Level	Exception Number	Invocation Conditions
Info or *Debug*	Info Debug	0	Any. These messages will never be standardized and service providers can design them as they see fit.
Warnings	Warning	1-999	Any. This range is reserved for the use of service providers to supply their own custom warnings.
Service Not Available	Fatal	1000	Service is executing a request, but due to internal errors cannot complete the request. Service must return `ReportResponse` and no payload.
Service Busy	Fatal	1010	Service is too busy to execute the incoming request. Service must return `ReportResponse` with this exception and no payload. Client should retry the request after some reasonable time.
Client has made too many requests	Fatal	1020	If the server sets a limit on the number of requests a client can make within a given timeframe, the server will return this error when the client exceeds that limit. The server would provide an explanation of the limit in the Message of the error. E.g., "Client has made too many requests. This server allows only 5 requests per day per `RequesterID` and `CustomerID`."
Requestor Not Authorized to Access Service	Error	2000	If `Requestor ID` is not recognized or not authorized by the service.
Requestor is Not Authorized to Access Usage for Institution	Error	2010	If `Requestor` has not been authorized to harvest usage for the institution identified by the `CustomerReference ID`, or if the `CustomerReference ID` is not recognized.
Report Not Supported	Error	3000	The requested report name, version, or other means of identifying a report that the service can process is not matched against the supported reports.
Report Version Not Supported	Error	3010	Requested version of the data is not supported by the service.
Invalid Date Arguments	Error	3020	Any format or logic errors involving date computations, e.g., end date cannot be less than begin date.

Exception	Severity Level	Exception Number	Invocation Conditions
No Usage Available for Requested Dates	Error	3030	Service did not find any data for the date range specified.
Partial Data Returned	Warning	3040	Request could not be fulfilled in its entirety. Data that was available was returned.

NOTE 1: An `Error` does not interrupt completion of the transaction (in the sense of a programmatic failure), although it may not return the expected report for the reason that is identified. A `Fatal` exception does not complete the transaction, however the problem may be temporary and a retry could be successful.

NOTE 2: Optional response for "No usage available for requested dates": Service may respond with the additional exception of `Info` level and include in the message additional information. E.g., if the client is requesting data for a date range where the begin date is before what the service offers, the service might include a `HelpURL` that can provide more information about supported dates.

6.3 Service Contract (WSDL)

The Service Contract represents a set of operations with input and output parameters that are agreed upon between the client and the service. These are defined in the core SUSHI WSDL. The version of the core WSDL that was current at the time of publication is reproduced in Appendix B. It is a common practice for a service to provide a WSDL document dynamically by querying the service, e.g., http://sushi.ws.ebsco.com/1_5/counterreports.asmx?wsdl.

6.3.1 Service Contract Definitions and Types

Each service provider must implement SUSHI operations and messaging protocol and, at a minimum, reference the SUSHI data contract directly or indirectly through another data contract.

Table 18: SUSHI Service Contract Definitions and Types

```
<definitions xmlns:tns="SushiService"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:import0="http://www.niso.org/schemas/sushi/1_5"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="SushiService"
targetNamespace="SushiService" xmlns="http://schemas.xmlsoap.org/wsdl/">

  <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" />
  <types>
    <xsd:schema>
      <xsd:import schemaLocation="http://www.niso.org/schemas/sushi/sushi_1_5.xsd"
namespace="http://www.niso.org/schemas/sushi/1_5" />
    </xsd:schema>
  </types>
```

6.3.2 Service Contract Operation

The SUSHI service contract implements only one operation: `GetReport`. This operation expects `ReportRequest` and returns `ReportResponse`, as illustrated in Table 19.

Table 19: SUSHI Service Contract Operation

```
<portType name="SushiServiceInterface">
    <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" />
    <operation name="GetReport">
        <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" />
        <input message="tns:GetReportIn" />
        <output message="tns:GetReportOut" />
    </operation>
</portType>
```

6.3.3 Service Contract Messages

The input and output message definitions of the service contract are listed in Table 20.

Table 20: SUSHI Service Contract Messages

```
<message name="GetReportIn">
    <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" />
    <part name="messagePart" element="import0:ReportRequest" />
</message>

<message name="GetReportOut">
    <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" />
    <part name="messagePart" element="import0:ReportResponse" />
</message>
```

6.3.4 COUNTER-SUSHI Service Contract

The service contract definition for the COUNTER-SUSHI (extended schema) service is listed in Table 21. It is shown here to illustrate an extension to the schema for a specific report type, in this case for COUNTER reports. Items that are significant are marked in bold. Types used for the data exchange are declared in the types section of the WSDL and are specific to the strongly typed payload that the data service will provide.

Table 21: COUNTER-SUSHI Service Contract Definitions and Types

```
<definitions xmlns:tns="SushiService"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:import0="http://www.niso.org/schemas/sushi/1_5/counter"
xmlns:import1="http://www.niso.org/schemas/sushi/1_5"
xmlns:import2="http://www.niso.org/schemas/sushi/counter/jd02_1_0"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="SushiService"
targetNamespace="SushiService" xmlns="http://schemas.xmlsoap.org/wsdl/">
       <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"/>
       <types>
              <xsd:schema>
                     <xsd:import
schemaLocation="http://www.niso.org/schemas/sushi/counter_sushi_1_5.xsd"
namespace="http://www.niso.org/schemas/sushi/1_5/counter"/>
                     <xsd:import schemaLocation="
http://www.niso.org/schemas/sushi_1_5.xsd"
namespace="http://www.niso.org/schemas/sushi/1_5"/>
                     <xsd:import schemaLocation="
http://www.niso.org/schemas/sushi/counter_jd02_1_0.xsd"
namespace="http://www.niso.org/schemas/sushi/counter/jd02_1_0"/>
              </xsd:schema>
       </types>
```

7 Report Naming

7.1 Types of Reports

There are three types of reports that can be requested using the SUSHI protocol.

1) COUNTER Reports

   SUSHI was originally conceived of as a method for delivering COUNTER reports. SUSHI is closely coupled, therefore, to developments at COUNTER. All COUNTER reports will be listed in the registry (see Section 7.3).

2) SUSHI Registered Non-COUNTER Reports

   The SUSHI protocol will support non-COUNTER reports that meet the requirements listed in Section 7.2. Registration for non-COUNTER reports will be voluntary, but is strongly recommended for any reports that might cross over business partners. (See Section 7.3.)

3) Other Reports

   SUSHI is designed to handle any report that is formatted in XML. Interested parties are encouraged to use the protocol for other purposes where it is appropriate.

7.2 Report Naming Requirements

The SUSHI XML schema requires that reports be uniquely identified by a name and a release. Both of these elements must conform to the type `string`. There is no restriction on length.

7.3 Registry of Reports

An official SUSHI Registry of Reports is maintained at the NISO SUSHI website: http://www.niso.org/committees/SUSHI/reports.html

All COUNTER reports will be listed in the registry by the SUSHI maintainers.

Registration of non-COUNTER reports is voluntary, but registration is strongly recommended for any reports that might cross over business partners. Registration will be on a first-come, first-serve basis for reserving a report name. NISO members may register non-COUNTER reports on behalf of themselves or their business partners.

8 Versions and Extensions

8.1 SUSHI Versioning

In order to remove future ambiguity as the SUSHI protocol evolves, this standard explicitly defines the versioning rules for the SUSHI data contract and operations. While the versioning of the actual report payload is outside the scope of this document, the Working Group's intent is addressed briefly in Section 8.2.

SUSHI versioning is as follows:

1) The following items must be versioned using the NISO standard revision process:
   a) All SUSHI data types as recorded in the SUSHI schema (XSD). This schema is versioned as a whole.

b) SUSHI service contract document (WSDL) elements:
   - `Message` element
   - `PortTypes` and `Operations` elements
   - `Binding` elements

NOTE: The `Types` section of the SUSHI service contract is excluded from the versioning process as long as it contains the reference to a SUSHI data type (see 6.3.1) or to an extension of the data contract.

2) Any changes to SUSHI data types or service elements will result in a schema that is assigned the next major version number (e.g., 1.0, 2.0, 3.0, etc.).
3) NISO will maintain the SUSHI data types schema (XSD) and will version the schema up to the next major version if the schema contains substantive changes.
4) NISO will issue a new WSDL document only when schemas, messages, or operations are added or modified. The implementers of this new service contract should publish their service using the new version number.
5) The URI format for each item is:

   http://www.niso.org/schemas/sushi/sushi<major-ver>.xsd

NOTE: It is strongly recommended that service providers utilize a unique URI when a new version of SUSHI is implemented. This will allow clients who have not yet upgraded to the new version to continue working and retrieving reports.

8.2 Extending Data Contract for Additional Reports

The one extensible part of the `ReportResponse` message is the `Report` element, which can accept any data type. As an example of an extension, Figure 4 and Figure 5 illustrate the COUNTER-SUSHI extension that was written to retrieve COUNTER reports. See Appendix C for the complete COUNTER-SUSHI schema.

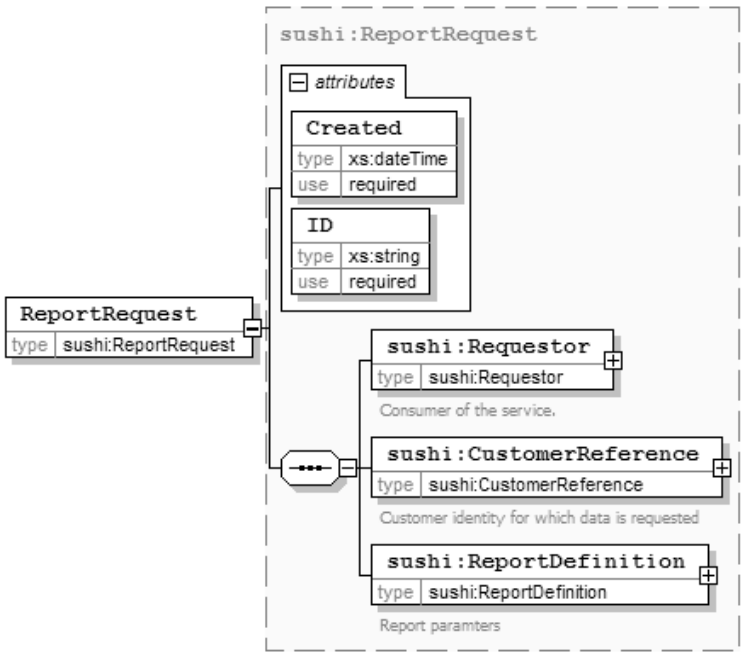

Figure 4: Diagram of COUNTER-SUSHI Extension `ReportRequest`

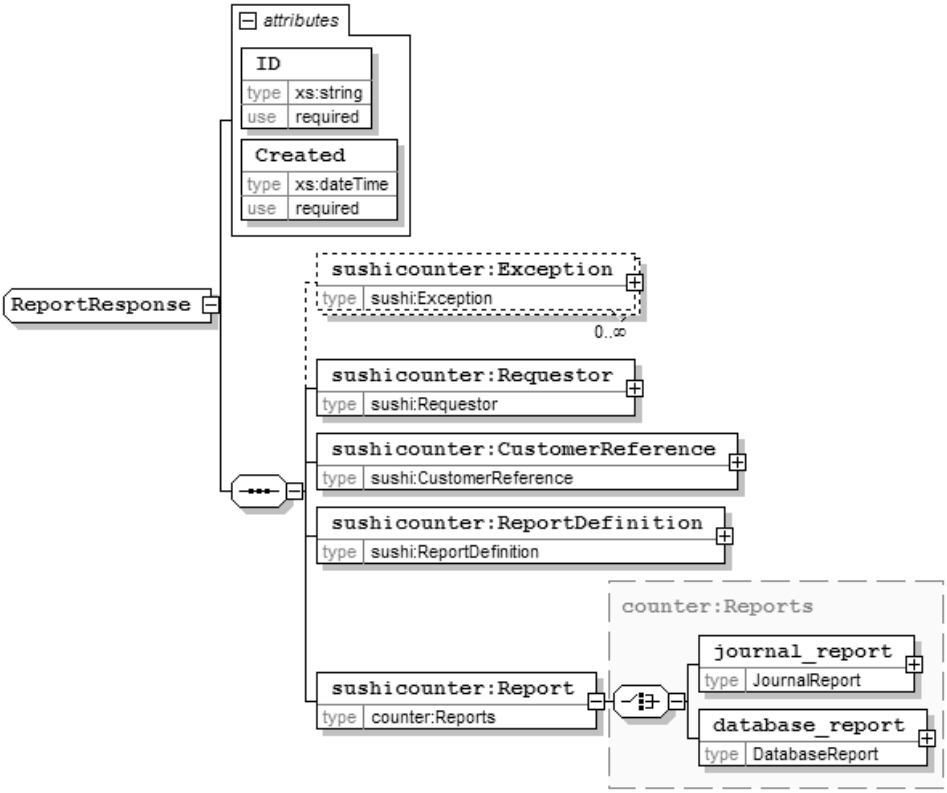

Figure 5: Diagram of COUNTER-SUSHI Extension `ReportResponse`

Implementers of the protocol would reference the appropriate data contract (schema) for the `Report` element in their particular implementation.

Appendix D provides further discussion on extending the `Reports` element within the standard.

Appendix A
(informative)
SUSHI Protocol XML Schema

(This appendix is not part of ANSI/NISO Z39.93-2013, *The Standardized Usage Statistics Harvesting Initiative (SUSHI) Protocol*. It is included for information only.)

The current XML schema at the time of publication of this standard (version 1.5) is included here for information only. The most current version of the schema is maintained on the SUSHI schema website (http://www.niso.org/schemas/sushi/).

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:sushi="http://www.niso.org/schemas/sushi/"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.niso.org/schemas/sushi/" elementFormDefault="qualified"
attributeFormDefault="unqualified" version="1.6">
    <xs:element name="ReportRequest" type="sushi:ReportRequest"/>
    <xs:element name="ReportResponse" type="sushi:ReportResponse"/>
    <xs:complexType name="ReportResponse">
        <xs:sequence>
            <xs:element name="Exception" type="sushi:Exception" nillable="true" minOccurs="0"
maxOccurs="unbounded"/>
                <xs:annotation>
                    <xs:documentation>One or more exceptions found in the request or while
processing the request.</xs:documentation>
                </xs:annotation>
            <xs:element name="Requestor" type="sushi:Requestor"/>
                <xs:annotation>
                    <xs:documentation>Repeated from the ReportRequest. The service or instance
of the service requesting the report. </xs:documentation>
                </xs:annotation>
            <xs:element name="CustomerReference" type="sushi:CustomerReference"/>
                <xs:annotation>
                    <xs:documentation>Repeated from the ReportRequest. The customer whose
usage is being reported.</xs:documentation>
                </xs:annotation>
            <xs:element name="ReportDefinition" type="sushi:ReportDefinition"/>
                <xs:annotation>
                    <xs:documentation>Repeated from the ReportRequest. Defines the report
being requested.</xs:documentation>
                </xs:annotation>
        </xs:sequence>
        <xs:attribute name="Created" type="xs:dateTime" use="required"/>
            <xs:annotation>
                <xs:documentation>Date/time the response was created.</xs:documentation>
            </xs:annotation>
        <xs:attribute name="ID" type="xs:string" use="required"/>
            <xs:annotation>
                <xs:documentation>If the ReportRequest had an ID, return in this
attribute.</xs:documentation>
            </xs:annotation>
    </xs:complexType>
    <xs:complexType name="ReportRequest">
        <xs:sequence>
            <xs:element name="Requestor" type="sushi:Requestor">
                <xs:annotation>
                    <xs:documentation>Consumer of the service (client).</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element name="CustomerReference" type="sushi:CustomerReference">
                <xs:annotation>
```

```
                    <xs:documentation>Identity of the customer for which data is
requested.</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element name="ReportDefinition" type="sushi:ReportDefinition">
                <xs:annotation>
                    <xs:documentation>Report parameters including report name, version, and
filters (e.g., date ranges).</xs:documentation>
                </xs:annotation>
            </xs:element>
        </xs:sequence>
        <xs:attribute name="Created" type="xs:dateTime" use="required"/>
            <xs:annotation>
                <xs:documentation>Date/time the request was created using format "yyyy-mm-dd
hh:mm:ss".</xs:documentation>
            </xs:annotation>
        <xs:attribute name="ID" type="xs:string" use="required"/>
            <xs:annotation>
                <xs:documentation>Identifier may be provided by client application for
internal use/diagnostics.</xs:documentation>
            </xs:annotation>
    </xs:complexType>
    <xs:complexType name="Requestor">
        <xs:annotation>
            <xs:documentation>
            <xs:documentation>Service consumer (client). Identifies the customer or 3rd party
acting on behalf of customer.</xs:documentation>
        </xs:annotation>
        <xs:sequence>
            <xs:element name="ID" type="xs:string">
                <xs:annotation>
                    <xs:documentation>Identifier by which the service (server) knows the
service consumer (client). For some services this may be the same as the Customer
ID.</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element name="Name" type="xs:string">
                <xs:annotation>
                    <xs:documentation>Name of the requestor organization. (This is not the
name of a person.)</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element name="Email" type="xs:string">
                <xs:annotation>
                    <xs:documentation>Valid e-mail address that service can use to contact a
human at the requestor organization.</xs:documentation>
                </xs:annotation>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="ReportDefinition">
        <xs:annotation>
            <xs:documentation>Input arguments for the report</xs:documentation>
        </xs:annotation>
        <xs:sequence>
            <xs:element name="Filters">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="UsageDateRange" type="sushi:Range"/>
                                <xs:annotation>
                                        <xs:documentation>The date range for the report. For
COUNTER reports statistics are pulled by month; therefore, the date range should cover a
range of months.</xs:documentation>
                                </xs:annotation>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
        <xs:attribute name="Name" type="xs:string" use="required">
            <xs:annotation>
                <xs:documentation>See http://www.niso.org/workrooms/sushi/reports for a list
of registered report names and releases.</xs:documentation>
            </xs:annotation>
        </xs:attribute>
        <xs:attribute name="Release" type="xs:string" use="required"/>
            <xs:annotation>
```

```
                <xs:documentation>The release number corresponding to the COUNTER Code of
Practice when the current definition of the report was introduced.</xs:documentation>
            </xs:annotation>
    </xs:complexType>
    <xs:complexType name="Range">
        <xs:annotation>
            <xs:documentation>The Date Range Type made up of Begin and End dates. Note:
COUNTER usage is only reported at the month level; therefore, this represents the range of
months.</xs:documentation>
        </xs:annotation>
        <xs:sequence>
            <xs:element name="Begin" type="xs:date"/>
                <xs:annotation>
                    <xs:documentation>Beginning date of range formatted as "yyyy-mm-dd".
Specify 'dd' as the first day of the month for COUNTER reports.</xs:documentation>
                </xs:annotation>
            <xs:element name="End" type="xs:date"/>
                <xs:annotation>
                    <xs:documentation>Ending date of range formatted as "yyyy-mm-dd". Specify
'dd' as the last day of the month for COUNTER reports.</xs:documentation>
                </xs:annotation>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="CustomerReference">
        <xs:annotation>
            <xs:documentation>A reference to a customer for which data is
requested.</xs:documentation>
        </xs:annotation>
        <xs:sequence>
            <xs:element name="ID" type="xs:string">
                <xs:annotation>
                    <xs:documentation>The Customer ID (usually proprietary) that the service
uses to identify the customer. The value of this ID for a given organization will be
different for each service.</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element name="Name" type="xs:string" nillable="true" minOccurs="0">
                <xs:annotation>
                    <xs:documentation>Customer name</xs:documentation>
                </xs:annotation>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="Exception">
        <xs:annotation>
            <xs:documentation>Service Exceptions</xs:documentation>
        </xs:annotation>
        <xs:sequence>
            <xs:element name="Number" type="xs:int">
                <xs:annotation>
                    <xs:documentation>Error Number.</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element name="Severity" type="sushi:ExceptionSeverity">
                <xs:annotation>
                    <xs:documentation>Severity of the error.</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element name="Message" type="xs:string">
                <xs:annotation>
                    <xs:documentation>Text describing the error.</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element name="HelpUrl" type="xs:anyURI" nillable="true" minOccurs="0">
                <xs:annotation>
                    <xs:documentation>URL describing error details.</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element name="Data" type="xs:anyType" nillable="true" minOccurs="0">
                <xs:annotation>
                    <xs:documentation>Other optional data.</xs:documentation>
                </xs:annotation>
            </xs:element>
        </xs:sequence>
        <xs:attribute name="Created" type="xs:dateTime" use="required"/>
    </xs:complexType>
    <xs:simpleType name="ExceptionSeverity">
        <xs:annotation>
```

```
            <xs:documentation>Service exception levels: Info, Debug, Warning,
Error</xs:documentation>
        </xs:annotation>
        <xs:restriction base="xs:string">
            <xs:enumeration value="Info">
                <xs:annotation>
                    <xs:documentation source="Info – type of severity that provides more
information about the response. Can be included in a chain of other severity levels"/>
                </xs:annotation>
            </xs:enumeration>
            <xs:enumeration value="Debug">
                <xs:annotation>
                    <xs:documentation source="Debug – type of severity that provides debugging
information about the response. Can be included in a chain of other severity levels"/>
                </xs:annotation>
            </xs:enumeration>
            <xs:enumeration value="Warning">
                <xs:annotation>
                    <xs:documentation source="Warning – type of severity that does not
interrupt the completion of the transaction (defined as: request->data->response) but gives
more information about the response. It could be followed by more exceptions (Info,
Debug)."/>
                </xs:annotation>
            </xs:enumeration>
            <xs:enumeration value="Error">
                <xs:annotation>
                    <xs:documentation>Error – type of severity that would return no payload.
It could be followed by more exceptions (Info, Debug).</xs:documentation>
                </xs:annotation>
            </xs:enumeration>
            <xs:enumeration value="Fatal">
                <xs:annotation>
                    <xs:documentation>Fatal – type of severity that would return no payload.
It could be followed by more exceptions (Info, Debug)</xs:documentation>
                </xs:annotation>
            </xs:enumeration>
        </xs:restriction>
    </xs:simpleType>
    <xs:complexType name="Activity">
        <xs:sequence/>
    </xs:complexType>
</xs:schema>
```

Appendix B
(informative)
Core SUSHI WSDL

(This appendix is not part of ANSI/NISO Z39.93-2013, *The Standardized Usage Statistics Harvesting Initiative (SUSHI) Protocol*. It is included for information only.)

The core SUSHI WSDL at the time of publication of this standard is included here for information only. The most current version of the WSDL is maintained on the SUSHI website (http://www.niso.org/schemas/sushi/).

```
<definitions xmlns:tns="SushiService" xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:import0="http://www.niso.org/schemas/sushi/1_6"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="SushiService"
targetNamespace="SushiService" xmlns="http://schemas.xmlsoap.org/wsdl/">

  <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" />
  <types>
    <xsd:schema>
      <xsd:import schemaLocation="http://www.niso.org/schemas/sushi/sushi_1_6.xsd"
namespace="http://www.niso.org/schemas/sushi/1_6" />
    </xsd:schema>
  </types>

  <message name="GetReportIn">
    <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" />
    <part name="messagePart" element="import0:ReportRequest" />
  </message>

  <message name="GetReportOut">
    <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" />
    <part name="messagePart" element="import0:ReportResponse" />
  </message>

  <portType name="SushiServiceInterface">
    <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" />
    <operation name="GetReport">
      <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" />
      <input message="tns:GetReportIn" />
      <output message="tns:GetReportOut" />
    </operation>
  </portType>

  <binding name="SushiService" type="tns:SushiServiceInterface">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
    <operation name="GetReport">
      <soap:operation soapAction="SushiService:GetReportIn" style="document" />
      <input>
        <soap:body use="literal" />
      </input>
      <output>
        <soap:body use="literal" />
      </output>
    </operation>
  </binding>
  <binding name="SushiServiceSoap12" type="tns:SushiServiceInterface">
    <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" />
    <operation name="GetReport">
      <soap12:operation soapAction="SushiService:GetReportIn" style="document" />
      <input>
        <soap12:body use="literal" />
      </input>
      <output>
        <soap12:body use="literal" />
```

```
      </output>
    </operation>
  </binding>
  <service name="SushiService">
    <port name="SushiServicePort" binding="tns:SushiService">
          <documentation>Location is intentionally left blank for the client to fill
in.</documentation>
      <soap:address location="" />
    </port>
  </service>
</definitions>
```

Appendix C
(informative)
COUNTER-SUSHI Extension for COUNTER reports

(This appendix is not part of ANSI/NISO Z39.93-2013, *The Standardized Usage Statistics Harvesting Initiative (SUSHI) Protocol*. It is included for information only.)

While the SUSHI schema was written as a generalized protocol to support a variety of potential usage reports, the main intent of the developers was to support COUNTER (Counting Online Usage of NeTworked Electronic Resources) reports. Thus, a COUNTER-SUSHI schema extension was developed for use with the general SUSHI protocol.

The current XML COUNTER-SUSHI schema at the time of publication of this standard (version 3.0) is included here for information only. The most current version of the schema is maintained on the SUSHI schema website (http://www.niso.org/schemas/sushi/).

NOTE: The namespace for the COUNTER-SUSHI schema is:

`http://www.niso.org/schemas/sushi/counter`

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:sushicounter="http://www.niso.org/schemas/sushi/counter"
xmlns:counter="http://www.niso.org/schemas/sushi/counter"
xmlns:sushi="http://www.niso.org/schemas/sushi"
targetNamespace="http://www.niso.org/schemas/sushi/counter" elementFormDefault="qualified"
attributeFormDefault="unqualified" version="3">
        <xs:import namespace="http://www.niso.org/schemas/counter"
schemaLocation="counter3_0.xsd"/>
        <xs:import namespace="http://www.niso.org/schemas/sushi/1_5"
schemaLocation="sushi_1_6.xsd"/>
        <xs:element name="ReportResponse" type="sc:CounterReportResponse">
                <xs:annotation>
                        <xs:documentation>The report response for COUNTER reports
(CounterReportResponse).</xs:documentation>
                </xs:annotation>
        </xs:element>
        <xs:element name="ReportRequest" type="s:ReportRequest">
                <xs:annotation>
                        <xs:documentation>Request from SUSHI Client requesting a COUNTER
report.</xs:documentation>
                </xs:annotation>
        </xs:element>
        <xs:complexType name="CounterReportResponse">
                <xs:annotation>
                        <xs:documentation>This is the actual response expected from a
COUNTER_SUSHI service. A combination of the SUSHI ReportResponse with the COUNTER "Report"
added. NOTE: The ReportRequest and ReportResponse are defined in the SUSHI1_6.xsd and the
COUNTER report is defined in the Counter3_0.xsd.</xs:documentation>
                </xs:annotation>
                <xs:complexContent>
                        <xs:extension base="s:ReportResponse">
                                <xs:sequence>
                                        <xs:element name="Report" type="c:Reports"
nillable="true">
                                                <xs:annotation>
```

```
                                                        <xs:documentation>This represents the
payload of the ReportResponsen by referencing the "Reports" element in the COUNTER
schema.</xs:documentation>
                                                  </xs:annotation>
                                        </xs:element>
                                  </xs:sequence>
                          </xs:extension>
                  </xs:complexContent>
          </xs:complexType>
</xs:schema>
```

Appendix D
(informative)
Utilizing SUSHI to Harvest Additional Reports

(This appendix is not part of ANSI/NISO Z39.93-2013, *The Standardized Usage Statistics Harvesting Initiative (SUSHI) Protocol*. It is included for information only.)

As stated in Section 8.2, only the `sushi.xsd` and `sushi.wsdl` are considered when versioning the SUSHI protocol. A conscious decision has been made to exclude the evolution and versioning of the report payload from the standard. This decision was made in order to provide a degree of flexibility in the protocol to handle the introduction of new reports without needing to amend the standard to address them.

Within the standard, reports are retrieved via a generic `ReportRequest` operation, using free form textual identifiers. The associated SUSHI report registry (see Section 7.3) provides a list of possible report/version combinations that may be supported by a given SUSHI implementation. Should a client issue a request for a report that is not supported by a specific SUSHI implementation, the service will return an appropriate `Exception` structure with the associated error information. Essentially, the SUSHI protocol handles the report as an opaque entity. As a result, there is nothing about the SUSHI protocol that needs to change in order to support a change in a report payload.

This is not to say that supporting a new or changed report will not require coding changes in the client or server applications.

- The service provider's implementation must be changed to recognize and produce a new or updated report. However, these changes can be managed in a manner that is transparent to the service's clients with respect to service version. The most client-friendly manner of implementing a report change would be to replace the old copy of the service with the enhanced copy at the same WSDL endpoint, but providers are not required to do so.
- Similarly, a client's integration must be modified to request the new/modified report and to process the resulting XML document. However, since the client must specify a specific report name and version to retrieve, it is insulated from any report changes at the service provider until it is ready to process them. If the report change at the provider were to remove the specific version of the report being requested, the client would receive a standard `Exception` response with the error details.

Appendix E
(informative)
SUSHI Maintenance

(This appendix is not part of ANSI/NISO Z39.93-2013, *The Standardized Usage Statistics Harvesting Initiative (SUSHI) Protocol*. It is included for information only.)

NISO has established a Standing Committee with responsibility for ongoing maintenance of the SUSHI standard. Questions and comments about this standard should be sent via www.niso.org/contact/; they will be forwarded to the Standing Committee.

The designated Standing Committee will be responsible for:

1) Responding to queries on the interpretation of this standard.
2) Reviewing suggestions for the improvement of the standard.
3) Maintaining a current list of inquiries and responses that may be used for potential future enhancements of this standard.
4) Maintaining the following components of the SUSHI protocol:
    a) SUSHI WSDL – The official sushi.wsdl and any corrections, changes, or extensions as allowed by the standard (see Section 8).
    b) SUSHI schema – The official `sushi.xsd` and any corrections, changes, or extensions as allowed by the standard (see Section 8).
    c) COUNTER-SUSHI schema – The official `ReportResponse` extension `counter_sushi.xsd` and any corrections, changes, or extensions as allowed by the standard (see Section 8)
    d) COUNTER payload schema – NISO has signed a Memorandum of Understanding with COUNTER that gives responsibility for creating and maintaining the COUNTER XML payload schema to the SUSHI developers. The intent is to keep the COUNTER XML schema synchronized with new releases of the COUNTER *Codes of Practice*. COUNTER's current policy is to limit changes in the *Code of Practice* to every two years. There are currently two codes that are updated in alternate years.
    e) SUSHI website – The SUSHI website includes the main page with general information, the schema webpage, the report registry webpage, and the FAQ. These will need to be updated to reflect any ongoing changes or needed explanatory information.
5) Initiating revisions to the standard as required in order to meet the needs of the SUSHI user community. Such revisions will be done in accordance with NISO's procedures for developing and revising standards.

Appendix F
(informative)
SUSHI Data Exchange Examples

(This appendix is not part of ANSI/NISO Z39.93-2013, *The Standardized Usage Statistics Harvesting Initiative (SUSHI) Protocol*. It is included for information only.)

F.1 Request

```
      <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
        <soap:Body>
          <ReportRequest ID="3D5C8B09-1720-4ff7-856A-2BDE09F95568"
xmlns="http://www.niso.org/schemas/sushi/1_5">
            <Requestor>
              <ID>3A520BFF-3EB8-4ed6-8A13-686C0932B6B0</ID>
              <Name>EBSCO</Name>
              <Email>pvucetin@ebsco.com</Email>
            </Requestor>
            <CustomerReference>
              <ID>1</ID>
              <Name>Web Development</Name>
            </CustomerReference>
            <ReportDefinition Release="2" Name="JR1">
              <Filters>
                <UsageDateRange>
                  <Begin>2003-07-01</Begin>
                  <End>2003-08-01</End>
                </UsageDateRange>
              </Filters>
            </ReportDefinition>
          </ReportRequest>
        </soap:Body>
      </soap:Envelope>
```

F.2 ReportResponse

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <soap:Body>
        <ReportResponse ID="3D5C8B09-1720-4ff7-856A-2BDE09F95568" Created="2007-07-
10T20:47:49.9712767Z" xmlns="http://www.niso.org/schemas/sushi/1_5/counter">
            <Requestor>
                <ID xmlns="http://www.niso.org/schemas/sushi/1_5">3A520BFF-3EB8-4ed6-8A13-
686C0932B6B0</ID>
                <Name xmlns="http://www.niso.org/schemas/sushi/1_5">EBSCO</Name>
                <Email
xmlns="http://www.niso.org/schemas/sushi/1_5">pvucetin@ebsco.com</Email>
            </Requestor>
            <CustomerReference>
                <ID xmlns="http://www.niso.org/schemas/sushi/1_5">1</ID>
                <Name xmlns="http://www.niso.org/schemas/sushi/1_5">Web Development</Name>
            </CustomerReference>
            <ReportDefinition Release="2" Name="JR1">
                <Filters xmlns="http://www.niso.org/schemas/sushi/1_5">
                    <UsageDateRange>
                        <Begin>2003-07-01</Begin>
                        <End>2003-08-01</End>
                    </UsageDateRange>
                </Filters>
            </ReportDefinition>
            <Report>
                <journal_report cop_report="JR1" cop_version="2" id="JR1:2"
xmlns="http://www.niso.org/schemas/sushi/counter/jd02_1_0">
                    <header>
```

```
                        <title>JR1</title>
                        <timestamp>2007-07-10T20:47:50.7837663Z</timestamp>
                        <vendor>
                                <vend_name>EBSCO Industries, INC</vend_name>
                                <vend_imprint>imprint</vend_imprint>
                                <vend_site>http://www.ebsco.com/</vend_site>
                                <vend_contact>ejsfeedback@ebsco.com</vend_contact>
                                <vend_logo url="http://www.ebsco.com/home/images/eislogo.gif"/>
                        </vendor>
                        <customer>
                                <cust_name>Web Development</cust_name>
                                <cust_ref>1</cust_ref>
                        </customer>
                    </header>
                    <journal_data>
                        <journal name="AACN Clinical Issues: Advanced Practice in Acute and
Critical Care" online_issn="" print_issn="1079-0713" publisher="Lippincott Williams &amp;
Wilkins" platform="EBSCOhost EJS">
                                <requests start="2003-07-01" end="2003-07-31"
type="ft_pdf">1</requests>
                                <requests start="2003-07-01" end="2003-07-31"
type="ft_html">0</requests>
                                <requests start="2003-07-01" end="2003-07-31"
type="ft_total">1</requests>
                                <requests start="2003-08-01" end="2003-08-31"
type="ft_pdf">1</requests>
                                <requests start="2003-08-01" end="2003-08-31"
type="ft_html">0</requests>
                                <requests start="2003-08-01" end="2003-08-31"
type="ft_total">1</requests>
                        </journal>
                        <journal name="Abacus" online_issn="" print_issn="0001-3072"
publisher="Blackwell Publishing Ltd" platform="EBSCOhost EJS">
                                <requests start="2003-07-01" end="2003-07-31"
type="ft_pdf">9</requests>
                                <requests start="2003-07-01" end="2003-07-31"
type="ft_html">0</requests>
                                <requests start="2003-07-01" end="2003-07-31"
type="ft_total">9</requests>
                                <requests start="2003-08-01" end="2003-08-31"
type="ft_pdf">2</requests>
                                <requests start="2003-08-01" end="2003-08-31"
type="ft_html">0</requests>
                                <requests start="2003-08-01" end="2003-08-31"
type="ft_total">2</requests>
                        </journal>
                    </journal_data>
                </journal_report>
            </Report>
        </ReportResponse>
    </soap:Body>
</soap:Envelope>
```


F.3 ReportResponse with Exception Elements


```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing" xmlns:wsse="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-
1.0.xsd">
        <soap:Header>
                <wsa:Action>SushiService:GetReportInResponse</wsa:Action>
                <wsa:MessageID>urn:uuid:2e9b4584-a14d-474f-9e47-02872b8abe4d</wsa:MessageID>
                <wsa:RelatesTo>urn:uuid:e4d4ae54-b1e9-4851-80a0-cfa873b0b977</wsa:RelatesTo>

        <wsa:To>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:To>
                <wsse:Security>
                        <wsu:Timestamp wsu:Id="Timestamp-13a0eb98-cc6b-42df-bc0b-
795c914c6fd3">
                                <wsu:Created>2007-07-10T20:57:09Z</wsu:Created>
                                <wsu:Expires>2007-07-10T21:02:09Z</wsu:Expires>
                        </wsu:Timestamp>
                </wsse:Security>
        </soap:Header>
        <soap:Body>
                <ReportResponse ID="3D5C8B09-1720-4ff7-856A-2BDE09F95568" Created="2007-07-
10T20:57:07.2675913Z" xmlns="http://www.niso.org/schemas/sushi/1_5/counter">
```

```
                        <Exception Created="2007-07-10T20:57:07.5488503Z">
                                <Number
xmlns="http://www.niso.org/schemas/sushi/1_5">2010</Number>
                                <Severity
xmlns="http://www.niso.org/schemas/sushi/1_5">Error</Severity>
                                <Message xmlns="http://www.niso.org/schemas/sushi/1_5">Access
denied.  Account '3A520BFF-3EB8-4ed6-8A13-686C0932B6B0' cannot access Resource 'B03BA8AD-
CF3C-4E10-ABA6-2963F46E7BD0' for customer '-1'</Message>
                                <HelpUrl
xmlns="http://www.niso.org/schemas/sushi/1_5">http://ws.tempuri.com/errorDescription.aspx?id=
2010</HelpUrl>
                                <Data xsi:nil="true"
xmlns="http://www.niso.org/schemas/sushi/1_5"/>
                        </Exception>
                        <Requestor>
                                <ID xmlns="http://www.niso.org/schemas/sushi/1_5">3A520BFF-
3EB8-4ed6-8A13-686C0932B6B0</ID>
                                <Name
xmlns="http://www.niso.org/schemas/sushi/1_5">EBSCO</Name>
                                <Email
xmlns="http://www.niso.org/schemas/sushi/1_5">pvucetin@ebsco.com</Email>
                        </Requestor>
                        <CustomerReference>
                                <ID xmlns="http://www.niso.org/schemas/sushi/1_5">-1</ID>
                                <Name xmlns="http://www.niso.org/schemas/sushi/1_5">Invalid
Customer Test</Name>
                        </CustomerReference>
                        <ReportDefinition Release="2" Name="JR1">
                                <Filters xmlns="http://www.niso.org/schemas/sushi/1_5">
                                        <UsageDateRange>
                                                <Begin>2003-07-01</Begin>
                                                <End>2003-08-01</End>
                                        </UsageDateRange>
                                </Filters>
                        </ReportDefinition>
                </ReportResponse>
        </soap:Body>
</soap:Envelope>
```

Appendix G
(informative)
Security Considerations

(This appendix is not part of ANSI/NISO Z39.93-2013, *The Standardized Usage Statistics Harvesting Initiative (SUSHI) Protocol*. It is included for information only.)

The SUSHI standard, which encompasses sushi.xsd and sushi.wsdl, does not include an integrated security mechanism; however, security was a consideration in developing this standard. The SUSHI Standing Committee recognizes that, for many institutions, usage data and other information about their collections are considered confidential and are covered by institutional security policies. Strong security is possible without building a security mechanism into the standard protocol. This appendix discusses three levels of security that can be implemented by SUSHI clients and servers. These are:

1) Securing the data communication channel
2) Authenticating the requesting organization (the SUSHI client software)
3) Validating the rights of a requesting organization to access usage data for a specific customer

G.1 Securing the Communications Channel

SUSHI is built on top of SOAP (Simple Object Access Protocol), which in turn uses either HTTP or HTTPS for transmission between client and server applications. By using HTTPS, the communication between client and server is encrypted using SSL (Secure Sockets Layer), thereby preventing any third party from intercepting the transmission and discovering its content.

It is recommended that all SUSHI developers implement their Web services for HTTP and HTTPS.

G.2 Authenticating the Requesting Organization

Using HTTPS, the communication channel is secure, however, there is still a possibility that an unauthorized software application could access a SUSHI server and request usage data. To prevent this, the content provider can implement a security layer within the SUSHI server. Following are three options:

- `Requestor ID` validation
- IP authentication
- Username/password

The simplest form of authentication would be to validate the `Requestor ID`. Unless the `Requestor` is known to the server, the request would not be processed. The `Requestor ID` can be a simple number or code identifying the client, or, if stronger security is desired, the value for the `Requestor ID` could include encrypted information, such as the domain of the client so that when the client submits the SUSHI `ReportRequest`, the server can decrypt the `Requestor ID` to verify that the client is legitimate.

Adding IP authentication is another option to create a much stronger level of security. The requesting organization would need to register the IP address of the computer running their SUSHI client software with the service provider(s). The service provider would only process requests for recognized IP addresses. When implementing IP authentication for SUSHI clients, be aware that

many of institutions will use a hosted usage harvesting service, which means the same client with the same IP address may be making requests on behalf of many institutions.

NOTE: To ensure interoperability of clients and servers, do not use WS-Security extensions or similar mechanisms to introduce username/password authentication to the SOAP or HTTP level.

G.3 Validating Rights of a Requesting Organization to Access Specific Customer Data

Most content providers will store usage data for a large number of institutions and will also have a large number of requesting organizations looking to harvest the usage data. The content provider can introduce another security layer to restrict authorized requesting organizations to certain customer data.

The SUSHI `ReportRequest` contains the `Requestor ID` (identifying the requesting organization) and the `CustomerReference ID` (identifying the organization whose usage is to be harvested). Service providers can fairly easily set up a system that requires their customers to “authorize” requesting organizations to harvest their data. If the service provider registers the requesting organizations, then it can present their customers with a simple user interface that gives them the option to “activate” SUSHI harvesting, then identify the requesting organization(s) allowed to do the harvesting. The result is a mapping between `CustomerReference IDs` and `Requestor IDs`, allowing the SUSHI server to verify that the data harvesting is permitted before processing is continued.

For service providers who are using IP authentication for the requesting organization, a simpler model could be implemented when the requestor and the customer are the same. The SUSHI server could verify that the IP address of the requestor is included in the IP range registered for the customer and, if so, processing of the request would continue.

G.4 Summary of Security Considerations

Even though security is not part of the standard, SUSHI can be a very secure protocol. Using HTTPS, the communication channel is secure. Service providers can authenticate the requesting organizations using IP addresses and `Requester IDs`. Service providers can allow customers to have their data further protected by first requiring them to opt in to SUSHI harvesting, then by identifying which requesting organizations (i.e., SUSHI client implementations registered with the service provider) are allowed to harvest their data.

Appendix H
(informative)
Creating Proprietary SUSHI Extensions

(This appendix is not part of ANSI/NISO Z39.93-2013, *The Standardized Usage Statistics Harvesting Initiative (SUSHI) Protocol*. It is included for information only.)

While every effort has been made to create and define a comprehensive framework for SUSHI, it is expected that the standard protocol will not address all possible future contingencies. It is the Working Group's intention to allow consumers and providers to extend their SUSHI implementation(s) with proprietary extensions that address these needs. In order to ensure that the core SUSHI specification is not impacted by such extensions, the following conventions must be observed when extending the specification. Implementations that are extended in accordance with these conventions are still considered to be SUSHI compliant, even though the extension itself does not become part of the SUSHI standard.

1) Proprietary WSDL extensions must be created from a copy of the sushi.wsdl with a filename that utilizes a second decimal place number increment to the version of the SUSHI protocol it is extending. For example:

   `sushi_1_5_1.wsdl`

   is an extension of version 1.5 of the sushi.wsdl.
2) No version of `sushi.xsd` may be modified, nor may any other version of `sushi.xsd` be substituted for the official SUSHI-maintained version in the extended WSDL.
3) No operation from the official SUSHI-maintained `sushi.wsdl` may be removed.
4) No operation from the official SUSHI-maintained `sushi.wsdl` may have its parameters modified or overloaded.
5) Proprietary extension WSDLs may import other schemas and utilize the data types and elements in `sushi.xsd` as needed to implement the extended functionality.
6) Proprietary operations should be named in such a way as to avoid potential naming conflicts with other provider's extensions.

   NOTE: A naming scheme of `X_<provider/client/groupName>_<operationName>` is recommended.

Table 22 represents an example of "extending" the SUSHI schema by importing a new schema. Figure 6 illustrates how the extended report requests are introduced in this new schema.

Table 22: Example of Extending SUSHI

```
<xs:schema
      xmlns="http://schemas.vendor.com/schemas/sushi/copy_sushi1_5" <- New schema
      targetNamespace="http://schemas.vendor.com/schemas/sushi/copy_sushi1_5" <-
New schema
      xmlns:xs="http://www.w3.org/2001/XMLSchema"
      xmlns:ns1="http://www.niso.org/schemas/sushi/1_5" <- Imported SUSHI schema
      elementFormDefault="qualified"
      attributeFormDefault="unqualified">

      <xs:import namespace="http://www.niso.org/schemas/sushi/1_5"
schemaLocation="sushi1_5.xsd"/> <- Imported SUSHI XSD
      <xs:element name="Root"> <- New TYPE extending original SUSHI type.
            <xs:annotation>
                  <xs:documentation>Comment describing your root
element</xs:documentation>
            </xs:annotation>
      </xs:element>
      <xs:complexType name="ExtendedReportRequest">
            <xs:complexContent>
                  <xs:extension base="ns1:ReportRequest">
                        <xs:sequence>
                              <xs:element name="MyNewProperty" type="xs:string"/>
                        </xs:sequence>
                  </xs:extension>
            </xs:complexContent>
      </xs:complexType>
</xs:schema>
```

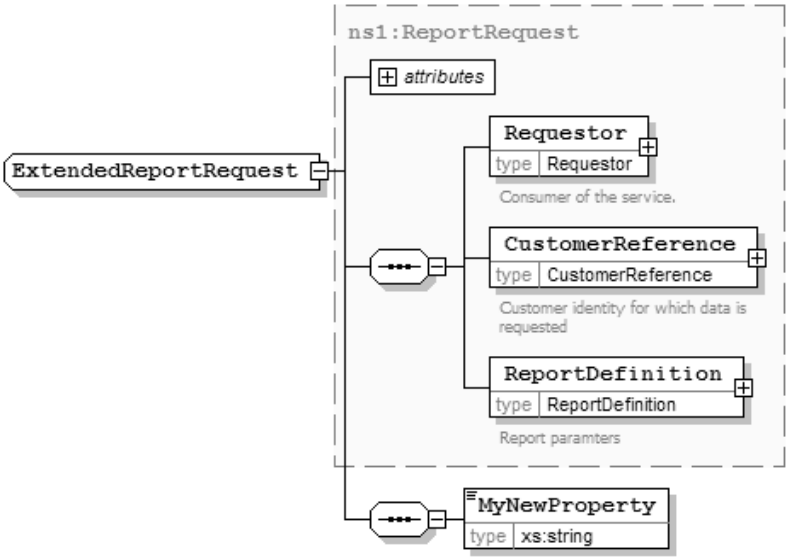

Figure 6: Diagram of "Extended" SUSHI Schema

Bibliography

(This bibliography is not part of ANSI/NISO Z39.93-2013, *The Standardized Usage Statistics Harvesting Initiative (SUSHI) Protocol*. It is included for information only.)

SUSHI Resources

Standardized Usage Statistics Harvesting Initiative (SUSHI) homepage:
www.niso.org/workrooms/sushi/

SUSHI FAQ
www.niso.org/workrooms/sushi/faq/

SUSHI Schema webpage:
http://www.niso.org/schemas/sushi/

SUSHI Reports Registry
www.niso.org/workrooms/sushi/reports/

SUSHI Server Registry
https://sites.google.com/site/sushiserverregistry/

SUSHI Server Recommendations
www.niso.org/workrooms/sushi/server/

COUNTER-SUSHI Implementation Profile (NISO RP-14-2012)
www.niso.org/apps/group_public/project/details.php?project_id=97

SUSHI Toolkits and Other Aids
www.niso.org/workrooms/sushi/tools/

Web Services Resources

Axis: Apache <Web Services /> Project [an open source, Java-based SOAP implementation]
http://ws.apache.org/axis/

Codehaus XFire [an open source, Java web-service framework]
http://xfire.codehaus.org/

Microsoft Developers Network, Web Services and Other Distributed Technologies
http://msdn.microsoft.com/webservices/

W3C Recommendation. **Simple Object Access Protocol (SOAP)**, version 1.2. World Wide Web Consortium, June 24, 2003. Available from: http://www.w3.org/TR/soap/