Open Geospatial Consortium Inc.

Date: 12 January 2011

Reference number of this OGC® project document: **OGC 10-069r2**

Category: OGC® Public Engineering Report

Editor: Panagiotis (Peter) A. Vretanos

OWS 7 Engineering Report -- Geosynchronization service

Copyright notice

Warning

This document is not an OGC Standard. This document is an OGC Public Engineering Report created as a deliverable in an OGC Interoperability Initiative and is not an official position of the OGC membership. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an OGC Standard. Further, any OGC Engineering Report should not be referenced as required or mandatory technology in procurements.

Document type:	OGC® Publicly Available Document
Document subtype:	Engineering report
Document stage:	Public
Document language:	English

Contents

List of Tables

List of FIGURES

i. Preface

The service described in this candidate standard has its origins in the Canadian Geospatial Data Infrastructure (CGDI) Pilot. The CGDI Pilot was an initiative of OGC's Interoperability Program to test the feasibility of using open, standards-based technology to improve the management and dissemination of geospatial data in Canada. In particular, this project demonstrated that technology from multiple vendors based on the Open Geospatial Consortium's Web Feature Service (WFS) standard could interoperate to provide access to the most current and authoritative data; thereby maintaining currency, avoiding versioning and minimizing duplication of the data. The main development of CGDI was conducted from April 2007 to January 2008

Additional concepts described in this candidate standard were developed during the OGC Web Service Phase 5 project and a project in Canada with the Ministries of Natural Resources and Transportation in the province of Quebec. Both projects tested and extended the concepts of a federated Geosynchronization system.

ii. Submitting organizations

The following organizations submitted this Implementation Specification to the Open Geospatial Consortium Inc. as a Request For Comment (RFC).

a) CubeWerx Inc.

b) The Carbon Project

c) Ministère des resources naturelles et de la faune

d) Ministère des transports du Québec

e) U.S. Army Geospatial Center (AGC) (details T.B.D)

iii. Submission contact points

All questions regarding this submission should be directed to the editor or the contributors:

Name	Email	Organization
Panagiotis (Peter) A. Vretanos (ed.)	pvretano[at]cubewerx.com	CubeWerx Inc.
Nuke Goldstein	ngoldstein[at]thecarbonproject.com	The Carbon Project
Jeff Harrison	jharrison[at]thecarbonproject.com	The Carbon Project
Edric Keighan	ekeighan[at]cubewerx.com	CubeWerx Inc.
Mark Mattson	mmattson[at]thecarbonproject.com	The Carbon Project
Gaetan Poulin	gaetan.poulin[at]mtq.gouv.qc.ca	MTQ
Romain Métivier	romain.metivier[at]mrnf.gouv.qc.ca	MRNF
Glenn Stowe	gstowe[at]cubewerx.com	CubeWerx Inc.

iv. Changes to the OGC® Abstract Specification

The OGC® Abstract Specification does not require changes to accommodate this OGC® candidate standard.

Foreword

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. Open Geospatial Consortium Inc. shall not be held responsible for identifying any or all such patent rights. However, to date, no such rights have been claimed or identified.

Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the specification set forth in this document, and to provide supporting documentation.

Introduction

Agile spatial data infrastructures, based on OGC standards, allow data provider organizations to deploy increasingly efficient networks capable of responding to dynamic requirements for using geospatial data. In this context, data provider organizations are being more and more called upon to deliver current, timely and verified data over the World Wide Web.

In order to satisfy these requirements, data providers must collaborate with outside entities to collect new data and/or update their existing data holdings. This may, for example, mean synchronizing their data with closest-to-source providers as might be the case between municipal, state/provincial and/or federal levels of government. This may also mean crowd-sourcing (see http://en.wikipedia.org/wiki/Crowdsourcing) their data production or supporting volunteer geographic information (VGI). Crowd-sourcing or VGI has proven to be a very effective tool for emergency management / disaster relief situations as is evidenced by the response to the Haiti earthquake of 12-JAN-2010.

Regardless of the nature of the collaboration, there is a need for a service to mediate the interaction between data providers and outside entities acting as data collectors. The service must support data entry with validation, notification of changes to interested parties and allow replication of the data provider's features.

A Geosynchronization service, deployed by a data provider, sits between the provider's data store(s) and data collectors. It allows data collectors to submit new data or make modifications to existing data without directly affecting the data in the provider's data store(s) until validation has been applied thus ensuring that the data published by the provider is of high quality.

This candidate standard makes no assumption about the nature of the validation or the quality criteria that a provider has established; only that the Geosynchronization service allows the data to be evaluated for quality before being applied to the provider's data store(s). For example, changes submitted by trusted data collectors may be applied directly to the provider's data stores while data obtained from external sources would be subjected to more rigorous validation. Validation might involve a manual process where authorised and certified reviewers check each submitted piece of data to assess its quality. Similarly, automated heuristics might be applied to assess the quality of the data. In any case, the data is validated before being applied to the target feature types.

Interested parties (or subscribers) must be able to flexibly track which changes have been submitted by data collectors, what the disposition of those changes was (i.e. whether the changes were accepted or rejected by the validation process) and also track the actual changes made to a provider's features for the purpose of replication. This flexible notification capability is achieved by allowing subscribers to specify predicates that precisely identify about which events they are interested in being notified. For example, a subscriber might register that they are interested in receiving notifications about proposed changes made within a specific geographic area.

Finally, a Geosynchronization service must support replication of the data provider's features by maintaining an event channel containing a log of all changes made to the features. This channel may be used to support several modes of replication including having the Geosynchronization service read the change log and apply the changes to one or more target feature types that have been subscribed to for replication. Replication may be full or partial possibly involving a schema translation from the provider's feature schema to the schema of the target feature.

1 Scope

This candidate standard describes a service that allows data collectors to propose changes to be made to a data provider's features. A change proposal can be made to create new data or to modify/delete existing data. Proposed changes are reviewed (either manually or automatically) an are either accepted or rejected. Accepted changes are applied to the feature(s). The service also maintains a log of all changes applied to each feature that can be used for replication.

The service maintains three ATOM feeds to support these activities. The change feed is a log of all proposed changes. The resolution feed is a log of the dispositions of all proposed changes. Finally, the replication feed is a log of all changes made to the data store. Operations are described to manage feed entries and this candidate standard describes the mandatory content of entries for each of the three feeds.

This candidate standard also describes a query method for the feeds that supports spatial and temporal predicates and is compatible with the OpenSearch standard.

Finally, this document describes a publish/subscribe API that allows interested parties to subscribe to any of the feeds for the purpose of notification and to the replication feed for the purpose of synchronizing features.

This candidate standard assumes that all features that participate in a Geosynchronization federation are accessible using the WFS-T API. Furthermore, all changes to be made to features are described as using the WFS Transaction API.

This candidate standard also assumes that some role-based identity management systems is available to the service to authenticate users of the service and determine their roles as described in this standard.

The candidate standard assumes that the service implements some notification mechanism. For example, a Geosynchronization service may access and Web Notification Service (see OGC 07-071) to perform the actual notification.

This candidate standard defines 13 operations:

- GetCapabilities
- GetEntries
- Insert (feed entries)
- Update (feed entries)
- Delete (feed entires)
- CreateTopic
- RemoveTopic
- ListTopics
- Subscribe
- ListSubscriptions

- PauseSubscription
- CancelSubscription

2 Conformance

2.1 Conformance classes

Table 1 lists the conformance classes defined in this candidate standard. Each class is characterized by an API and protocol binding resulting in 28 conformance classes being defined.

The content of each cell in Table 1 identifies the clauses in this document where that particular combination of API and protocol binding is described.

Table 1 – Conformance Classes

			PROTOCOL BINDINGS			
CONFORMANCE CLASS	**API**	**OPERATIONS**	**OGC POX**	**OGC KVP**	**REST**	**SOAP**
Basic	Discovery	GetCapabilities[1]	9.3.1.5.1.1	9.3.1.5.1.2	N/A	9.3.1.5.2
	Transaction	Insert,Update,Delete	9.3.2.4.1.1	N/A	9.3.2.4.2	9.3.2.4.3
	Query	GetEntries	9.3.3.4.1.1	9.3.3.4.1.2	9.3.3.4.2	9.3.3.4.3
	Change Management	AcceptChange RejectChange ReviewChanges	9.3.4.2.4.1.1, 9.3.4.3.4.1.1, 9.3.4.4.4.1.1	9.3.4.2.4.1.2, 9.3.4.3.4.1.2, N/A	9.3.4.2.4.2, 9.3.4.3.4.2, N/A	9.3.4.2.4.2, 9.3.4.3.4.3, 9.3.4.4.4.3
Topics	Topic Management	CreateTopic RemoveTopic ListTopics	9.3.5.2.4.1.1, 9.3.5.3.4.1.1, 9.3.5.4.4.1.1	9.3.5.2.4.1.2, 9.3.5.3.4.1.2, 9.3.5.4.4.1.2	9.3.5.2.4.2, 9.3.5.3.4.2, 9.3.5.4.4.2	9.3.5.2.4.3, 9.3.5.3.4.3, 9.3.5.4.4.3
Active Notification	Active Notification	Subscribe ListSubscriptions PauseSubscription ResumeSubscription CancelSubscription	9.3.5.5.4.1.1, 9.3.5.6.4.1.1, 9.3.5.7.4.1.1, 9.3.5.8.4.1.1, 9.3.5.9.4.1.1	9.3.5.5.4.1.2, 9.3.5.6.4.1.2, 9.3.5.7.4.1.2, 9.3.5.8.4.1.2, 9.3.5.9.4.1.2	9.3.5.5.4.2, 9.3.5.6.4.2, 9.3.5.7.4.2, 9.3.5.8.4.2, 9.3.5.9.4.2	9.3.5.5.4.3, 9.3.5.6.4.3, 9.3.5.7.4.3, 9.3.5.8.4.3, 9.3.5.9.4.3
Active Synchronization	Synchronization[2]	Subscribe	9.3.5.5.4.1.1	9.3.5.5.4.1.2	9.3.5.5.4.2	9.3.5.5.4.3

1. This class includes the GetCapabilities operation that generates an OGC capabilities document, the AtomPub service document accessible via a published URL and the OpenSearch description document also accessible via published URL.

2. The active synchronization class is an implementation of the Subscribe operation, from the Active Notification class, with support for the "sync" delivery method (see 9.3.5.5.2.1 & Table 17).

All implementations of this candidate standard shall implement the Basic class. The specific protocol bindings implemented for each API can be mixed and matched.

Optionally, implementations of this candidate standard may implement the Topics, Active Notification and/or Active Synchronization classes. Again, the specific protocol bindings implemented for each supported API can be mixed and matched.

2.2 OWS-7 project

Table 2 lists the APIs and protocol bindings that where implemented for the OWS-7 project.

Table 2 – OWS-7 APIs and protocol bindings

API	Protocol Binding
Discovery	OGC KVP with HTTP GET
Transaction	REST (i.e. AtomPub)
Query	OGC KVP (with mapping to OpenSearch) with HTTP GET
Change Management	OGC KVP with HTTP GET
Topic Management	OGC KVP with HTTP GET
Active Notification	OGC KVP with HTTP GET
Synchronization	OGC KVP with HTTP GET

3 Normative references

The following normative documents contain provisions which, through reference in this text, constitute provisions of this part of OGC 10-069. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this part of OGC 10-069 are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. For undated references, the latest edition of the normative document referred to applies.

IETF RFC 2368, The mailto URL Scheme (July 1998)

IETF RFC 2616, Hypertext Transfer Protocol -- HTTP/1.1 (June 1999)

IETF RFC 3920, Extensible Messaging and Presence Protocol (XMPP): Core (October 2004)

IETF RFC 3986, Uniform Resource Identifier (URI): Generic Syntax (January 2005)

IETF RFC 4287, The Atom Syndication Format (December 2005)

IETF RFC 5023, The Atom Publishing Protocol (October 2007)

OGC 04-094, OpenGIS Web Feature Service (WFS) Implementation Specification

OGC 04-095, OpenGIS Filter Encoding Implementation Specification

OGC 06-049r1, OpenGIS GML 3.1.1 Simple Features Profile

OGC 06-050r3, An Introduction to GeoRSS: A Standards Based Approach for Geo-enabling RSS feeds (19 July 2006)

OGC 06-121r3, OpenGIS Web Service Common Implementation Specification

OGC 07-004, OWS-4 GeoDDS Mass Market (formerly GeoRSS) Interoperability Program Report

OGC 07-036, Geography Markup Language (GML)

OGC 10-032, OpenGIS® Catalogue Service Implementation Specification, Version 3.0, Part 4: OpenSearch Query Interface (February 2010)

OpenSearch, OpenSearch v1.1 (Draft 4)

W3C XML, Extensible Markup Language (XML) 1.0 (Third Edition), W3C Recommendation (4 Feb 2004)

W3C XML Namespace, Namespaces in XML, W3C Recommendation (14 JAN 1999)

W3C XML Schema Part 1, XML Schema Part 1: Structures, W3C Recommendation (2 May 2001)

W3C XML Schema Part 2, XML Schema Part 2: Datatypes, W3C Recommendation (2 May 2001)

W3C WSDL, Web Service Description Language (WSDL) 1.1 (15 March 2001)

4 Terms and definitions

4.1
attribute
name-value pair contained in an element

4.2
category document
documents that describe the categories allowed in Collection

4.3
change feed
collection of ATOM entries that describe changes to a data store expressed using the WFS Transaction syntax (see OGC 04-094)

4.4
client
software component that can invoke an operation from a server

4.5
collection
resource that contains a set of member resources

NOTE In this candidate standard, collection are implemented as ATOM feeds (see IETF 4287).

4.6
collector
a person or entity that proposes changes to data

4.7
coordinate
one of a sequence of n numbers designation the position of a point in n-dimensional space

4.8
coordinate reference system
coordinate system that is related to an object by datum

4.9
element
basic information item of an XML document containing child element, attribute and character data

4.10
entry resource
members of a collection that are represented as ATOM entry documents (see IETF RFC 4287)

4.11
event
any detectable or discernable occurrence that has significance for the management of an SDI

4.12
feature
abstraction of real work phenomena

4.13
feature identifier
identifier that uniquely designates a feature instance

4.14
filter expression
predicate expression encoded in XML (see OGC 04-095)

4.15
follower
person or process that accesses or subscribes to the replication feed of a GSS for the purpose of data synchronization

4.16
integrator
person or process that reviews proposed data changes and then makes a determination (based on established criteria) if the proposed change is acceptable or not

4.17
interface
named set of operations that characterize the behaviour of an entity

4.18
local resource
resource that is under the direct control of a system

4.19
member resource
resource whose IRI is listed in a Collection with a atom:link element with a relation of "edit" or "edit-media"

4.20
Multipurpose Internet Mail Extensions (MIME) type
media type and subtype of data in the body of a message that designates the native representation (canonical form) of such data

4.21
namespace
collection of names, identified by a URI reference which are used in XML documents as element names and attribute names

4.22
operation
specification of a transformation or query that an object may be called to execute

4.23
property
facet or attribute of an object, referenced by name

4.24
publisher
synonym for collector (see 4.6)

4.25
remote resource
a resource that is not under the direct control of a system

4.26
replication feed
collection of ATOM entries containing a log of changes that have been applied to a data store that can be used for the purpose of replicating or synchronizing with that data store

4.27
representation
entity included with a request or response (see IETF RFC 2616)

4.28
request
invocation of an operation by a client

4.29
resolution feed
collection of ATOM entries describing the disposition of proposed changes listed in a change feed

4.30
resource
asset or means that fulfils a requirement

NOTE In this candidate standard a resource is a network-accessible data object or service identified by an IRI, as defined in [RFC2616]

4.31
response
result of an operation returned from a server to a client

4.32
reviewer
synonym for integrator (see 4.15)

4.33
schema
format description of a model

4.34
schema
<XML Schema>
collection of schema components within the same target namespace

4.35
server
particular instance of a service

4.36
service
distinct part of the functionality that is provided by an entity through an interface

4.37
service metadata
metadata describing the operations and information available at a server

4.38
service document
XML document that describes the location and capabilities of one or more Collections grouped into Workspaces

4.39
topic
collection of ATOM entries that satisfy some query predicates

NOTE: this is also referred to as a filtered feed because a topic is generated by querying a base feed and applying some predicate; for example a topic could consist of all the entries that lie within some defined boundary

4.40
Uniform Resource Identifier
unique identifier for a resource, structured in conformance with IETF RFC 3986

NOTE The generate syntax is <scheme>::<scheme-specified-part>. The hierarchical syntax with a namespace is <scheme://<authority><path>?<query>

4.41
Workspace
named group of collections

5 Abbreviated terms

AtomPub	ATOM Publishing Protocol

CGDI	Canadian Geospatial Data Infrastructure
CRS	Coordinate reference system
DCP	Distributed Computing Platform
EPSG	European Petroleum Survey Group
FES	Filter Encoding Specification
GML	Geography Markup Language
GSS	GeoSynchronization Service
HTTP	Hypertext Transfer Protocol
HTTPS	Secure Hypertext Transfer Protocol
IETF	Internet Engineering Task Force
KVP	Keyword-value pairs
MIME	Multipurpose Internet Mail Extensions
OGC	Open Geospatial Consortium
OWS	OGC Web Service
SDI	Spatial Data Infrastructure
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
URN	Uniform Resource Name
VSP	Vendor Specific Parameter
WNS	Web Notification Service
WFS	Web Feature Service
XML	Extensible Mapkup Language

6 Abstract model

6.1 Introduction

This clause describes, in an abstract way, the resources and operations required to support the operation of a Geosynchronization service. The discussion in this section is intended to be agnostic of any particular implementation. Clause 9, describes a concrete implementation of this candidate standard.

6.2 Identity management

6.3 Resources

6.3.1 Events

This document defines three event types that are handled by a GSS. These are: Change Proposal events, Resolution events and Replication events.

A change proposal event indicates that a collaborating party has signaled a desire to add new data to or modify/delete existing data from a data provider's data store.

A resolution event signals to a collaborating party or data publisher that their proposed change has been reviewed and has either been accepted or rejected.

A replication event signals that a change has been made to a data provider's data store and describes what that change is. Interested parties can then retrieve the description of the change and apply that change to their local copy of the data thus synchronizing their data store with the data provider's data store.

Table 3 lists the minimum set of properties that may be used to describe an event.

Table 3 – Event properties

Property Name	Cardinality	Description
id	1	A unique identifier for the event.
author	0 or N	A description of the entity or entities signaling the event.
updated	1	A timestamp indicating when the event was created or last changed.
title	1	A human-readable title for the event.
summary	0 or 1	A human-readable narrative describing the nature of the event.
content	0 or 1	Any payload that needs to be associated with the event. For example, a WFS Transaction describing a proposed change.
category	0 or N	A tag that classifies the event according to some classification scheme. An event may be classified many times using different classification schemes.
link	0 or N	A link to related resources or events.
where	0 or 1	A geometry that geo-locates the event.

The specific type for each of these properties is implementation dependant.

6.3.2 Event channels

An event channel is a conduit for the dissemination of events.

A Geosynchronization service shall implement three event channels: a Proposed Change event channel, a Resolution event channel and a Replication event channel.

The Proposed Change channel allows data publishers to accept updates to their data from collaborative participants or data publishers. Participants register proposed changes to the Proposed Change event channel. A proposed change can propose the creation of new data or the modification/deletion of existing data.

Proposed changes are reviewed in order to ensure that each change satisfies quality or correctness criteria established by the data provider. The review process can be performed manually or automatically the end result being that each proposed change is either accepted or rejected. The result of the review process – to accept or reject the change – is registered in the Resolution event channel. Events registered in the Resolution channel shall be cross-linked with events registered in the Proposed Change channel so that an interested party can track the disposition of one or more proposed changes.

In the event that a proposed change is accepted, it shall be applied to the data provider's data store and an event shall be registered to the Replication event channel describing the exact change that was made to each affected feature. Replication events always describe a change made to a single feature. Interested parties, who wish to maintain a synchronized copy of all or part of a data provider's data store can read the events from the Replication channel and apply them to their local copy.

Table 4 lists the minimum set of properties that may be used to characterize and event channel. Specific implementations of this specification may include additional properties.

Table 4 – Event channel properties

Property Name	Cardinality	Description
id	1	A unique identifier for the event channel.
title	1	A human-readable title for the event channel.
updated	1	A timestamp indicating where the event channel was created of the last time it was updated.
event	0 or N	Zero or more event entries.

6.3.3 Topics

A topic is an ad-hoc event channel or view of an event channel that has been filtered in order to identify events that satisfy some user-specified predicates. For example, a topic might be created that identifies events that occur within a specific geographic region by applying a spatial predicate (e.g. Within) to an event channel.

Topics are read-only objects since they are derived by querying the primary event channels that a GSS manages (see 6.3.2).

Figure 1 defines the structure of a Topic.

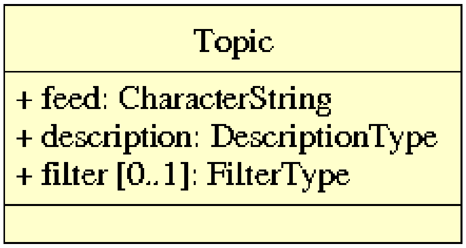


Figure 1 – Topic UML class diagram

6.3.4 Subscriptions

A subscription is used to signal a desire to be notified whenever an event appears in an event channel or topic. Subscriptions are considered resources because they can be created, destroyed and operated upon.

Figure 2 defines the structure of a Subscription.

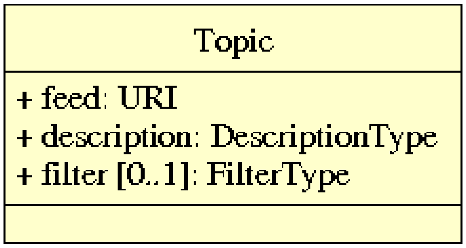


Figure 2 – Subscription UML class diagram

6.4 Identity management

6.4.1 Users

The GSS is a secure service and this document assumes that users have to be authenticated in order to use the server.

This candidate standard further assumed that users shall also be assigned roles that limit the type and scope of GSS operations that can be executed.

Thus, this candidate standard assumes that some identity management system is implemented by or available to the GSS. This candidate standard makes no assumption about which identity management system the server implements or uses. However, the user's credentials, including any roles assigned to the user, shall be accessible by the GSS.

6.4.2 Roles

This candidate standard assumes that the following user-roles are available:

1. Normal user – can read any event channel

2. Data publisher – can create entries in the Proposed Changes event channel

3. Reviewer – can create entries in the Resolution event channel

6.4.3 Permission model

A GSS shall define a root or system user who is omnipotent within the system.

Any user with any role shall be able to read any event from an event channel.

Only users with data publisher or reviewer roles shall be able to create entries in the proposed change event channel.

Only users with the reviewer role shall be able to create entries in the Resolution event channel.

Only users with the reviewer role shall be able to create entries in the Replication event channel.

Only the owner of an event entry shall be able to modify or delete that entry.

Any user can create a Topic and only the owner of a topic can remove it.

When a user requests a list of topics from a GSS, they shall be able to see all available topics.

Any user can subscribe to any event channel or topic. Only a subscription owner can pause, resume or cancel that subscription.

When a user requests a list of subscription from the system, they shall only be able to see the subscriptions they own.

6.5 Service interface

6.5.1 Common types

Figure 3 defines types that are references in the remainder of sub-clause 6.5.

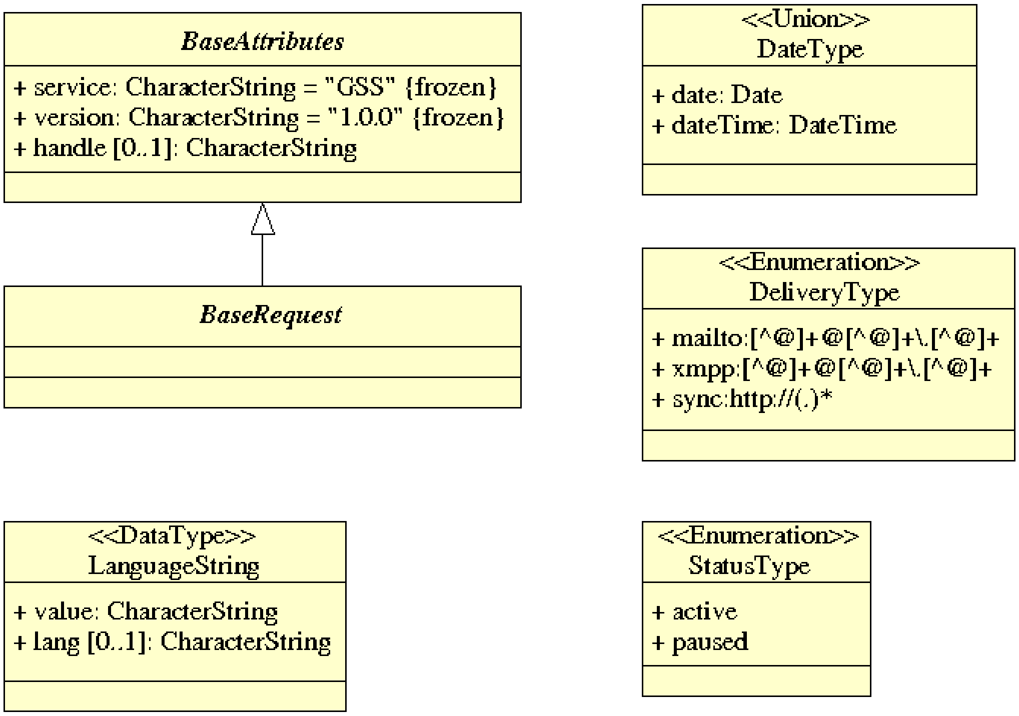


Figure 3 – Common types UML class diagram

6.5.2 Discovery

A Geosynchronization service shall, on demand, provide service metadata to a client application. The specific format of the metadata is implementation dependant. Such metadata may include a description of who is providing the service, the operations that the service implements, some light-weight metadata about the content of the service and perhaps some indication of the kinds of queries that can be performed by the service.

Figure 4 defines the operations of the Discovery service interface and their parameters.

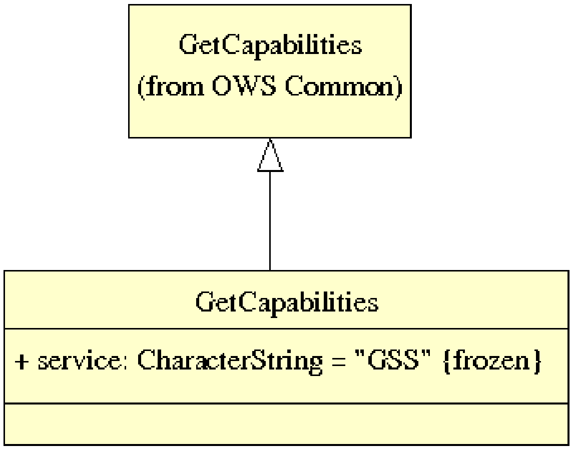


Figure 4 – Discovery UML class diagram

6.5.3 Transaction

A primary function of a Geosynchronization service is to manage the contents of event channels (see 6.3.2). As such, the service shall implement operations that allow events to be created, modified or removed from an event channel.

Figure 5 defines the operations of the Transaction service interface and their parameters.

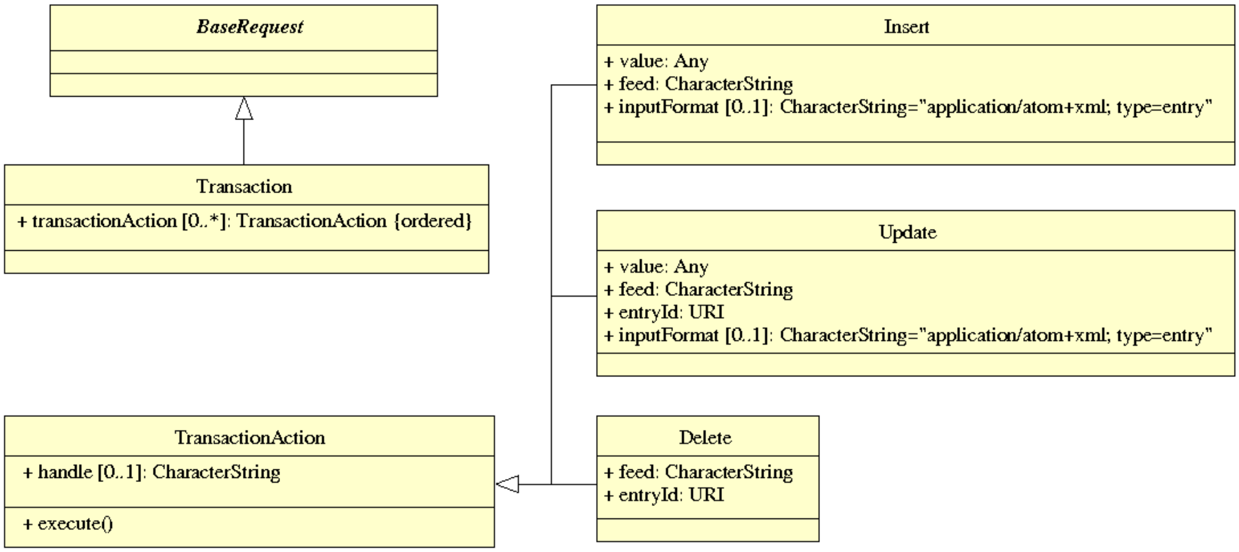

BaseRequest
Transaction
+ transactionAction [0..*]: TransactionAction {ordered}
TransactionAction
+ handle [0..1]: CharacterString
+ execute()
Insert
+ value: Any
+ feed: CharacterString
+ inputFormat [0..1]: CharacterString="application/atom+xml; type=entry"
Update
+ value: Any
+ feed: CharacterString
+ entryId: URI
+ inputFormat [0..1]: CharacterString="application/atom+xml; type=entry"
Delete
+ feed: CharacterString
+ entryId: URI

Figure 5 – Transaction UML class diagram

The response to a Transaction operation is defined in figure 6.

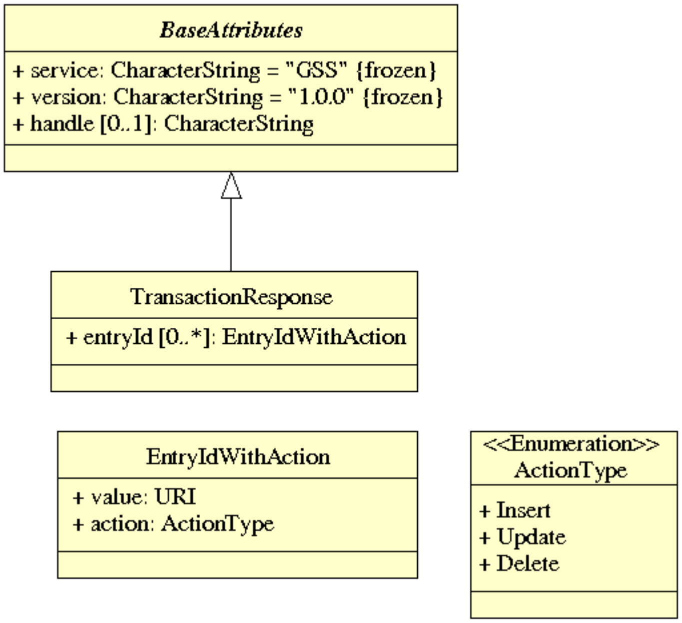

BaseAttributes
+ service: CharacterString = "GSS" {frozen}
+ version: CharacterString = "1.0.0" {frozen}
+ handle [0..1]: CharacterString
TransactionResponse
+ entryId [0..*]: EntryIdWithAction
EntryIdWithAction
+ value: URI
+ action: ActionType
<<Enumeration>>
ActionType
+ Insert
+ Update
+ Delete

Figure 6 – Transaction response UML class diagram

6.5.4 Query

A GSS shall offer the ability to query event channels using spatial and non-spatial predicates to identify a subset of events that satisfy the specified constraints.

Spatial predicates include both geometric and temporal constrains. Temporal constraints are particularly important since they allow clients to poll an event channel for new events by querying the channel for events that have occurred since the last time the channel was queried.

A GSS shall sort query responses in descending temporal order such that the most recent event in the result set is presented first.

Figure 7 defines the operations of the Query service interface and their parameters.

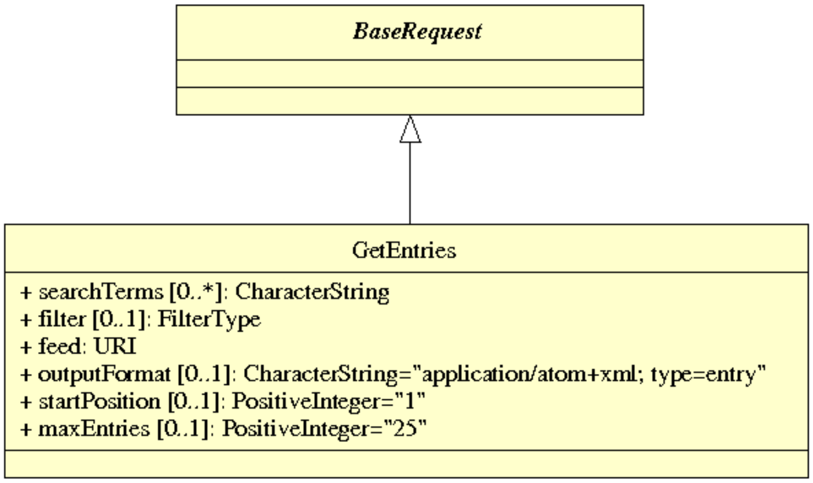

BaseRequest
GetEntries
+ searchTerms [0..*]: CharacterString
+ filter [0..1]: FilterType
+ feed: URI
+ outputFormat [0..1]: CharacterString="application/atom+xml; type=entry"
+ startPosition [0..1]: PositiveInteger="1"
+ maxEntries [0..1]: PositiveInteger="25"

Figure 7 – Query UML class diagram

6.5.5 Change management

6.5.5.1 Introduction

A primary function of a GSS is to allow authorized or automated reviewers to vet changes before they are applied to a data provider's data store. The change management interface defines operations for accepting or rejecting change proposals one at a time or in bulk.

Figure 8 defines the operations of the Change management service interface and their parameters.

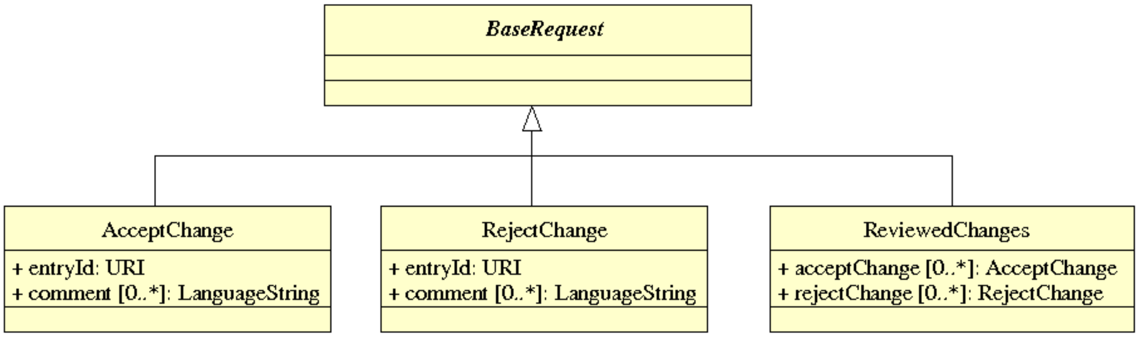

BaseRequest
AcceptChange
+ entryId: URI
+ comment [0..*]: LanguageString
RejectChange
+ entryId: URI
+ comment [0..*]: LanguageString
ReviewedChanges
+ acceptChange [0..*]: AcceptChange
+ rejectChange [0..*]: RejectChange

Figure 8 – Change management UML class diagram

The response to Change management operations is defined in figure 9.

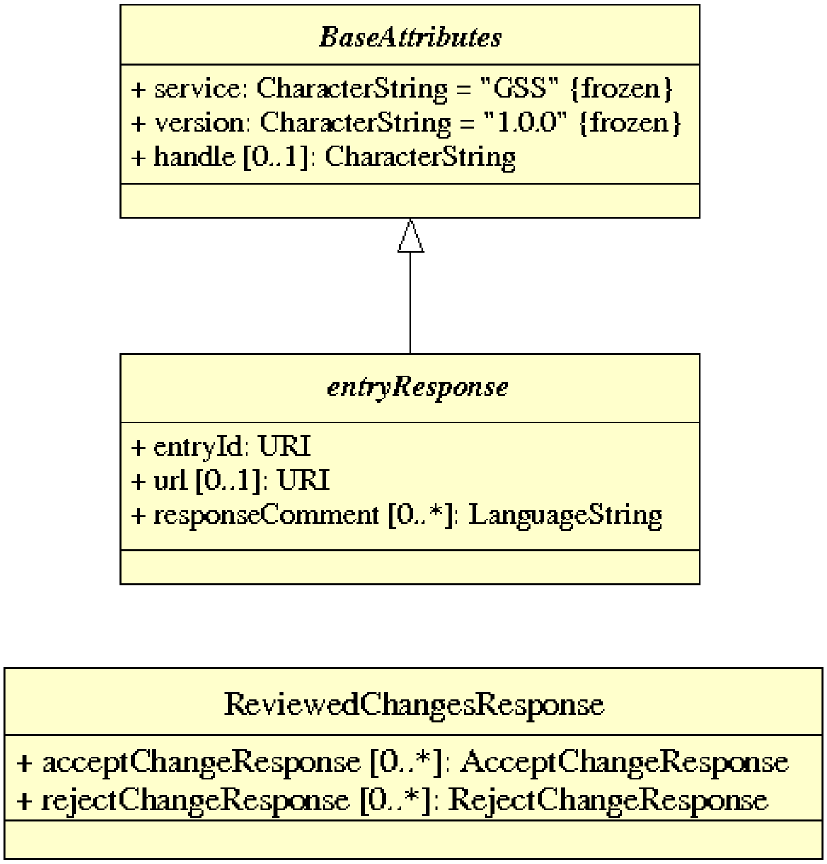

BaseAttributes
+ service: CharacterString = "GSS" {frozen}
+ version: CharacterString = "1.0.0" {frozen}
+ handle [0..1]: CharacterString
entryResponse
+ entryId: URI
+ url [0..1]: URI
+ responseComment [0..*]: LanguageString
ReviewedChangesResponse
+ acceptChangeResponse [0..*]: AcceptChangeResponse
+ rejectChangeResponse [0..*]: RejectChangeResponse

Figure 9 – Change management responses UML class diagram

6.5.5.2 Accept change proposal

The AcceptChange operation marks a change proposal event as accepted. The operation shall create a new event in the Resolution event channel, signaling that the proposed change was accepted.

The service shall cross link the corresponding events in the Change Proposal event channel and the Resolution event channel so that the disposition of the proposed change can be determined by following either event channel.

This operation shall apply the proposed change to the data store that is under the control of the GSS and shall create one or more events in the Replication event channel signaling the changes made.

6.5.5.3 Reject change proposal

The RejectChange operation is used to indicate that a change proposal event has been rejected. This means that a reviewer (human or otherwise) has determined that the proposal is not acceptable.

For a RejectChange operation, the service shall create a new event in the Resolution event channel signaling the rejection and the original change proposal event.

The service shall also cross link the corresponding change proposal events in the Change Proposal event channel and the Resolution event channel so that the disposition of the proposed change can be determined by following either event channel.

6.5.5.4 Bulk accept/reject processing

The ReviewedChanges operation contains one or more AcceptChange and/or RejectChange operations and enables bulk processing of change proposals.

6.5.6 Topic management

Identifying events to trigger notification is done through the use of Topics (see 6.3.3). Table 5 list the set of operations that may be used for create, discover and response topics.

Table 5 – Topic management API

Operation Name	Description
CreateTopic	Create an ad-hoc event channel based on one of the three Geosynchronization feeds. A topic can include non-spatial, spatial and temporal predicates to identify entries that satisfy certain criteria. For example, a topic can be created that filters the Proposed Change channel for entries that lie within a specific geographic area.
RemoveTopic	Delete a topic from the service.
ListTopics	Get a list of all available topics.

Figure 10 defines the operations and parameters for the Topics service interface which, in combination with the Active Notification service interface (see 6.5.7), allows users to be notified of events of interest.

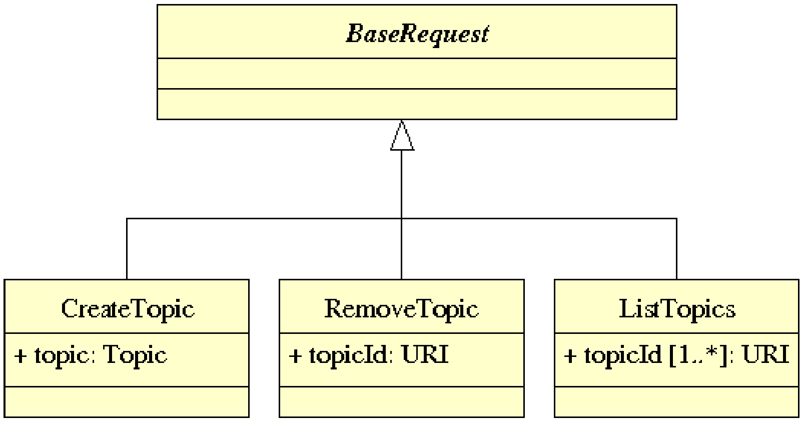


Figure 10 – Topic management UML class diagram

This candidate standard assumes that some mechanism for sending notification messages to subscribers is implemented by or available to the GSS. This candidate standard makes no assumption about which notification system the server implements or uses.

The response to the ListTopics operation of the Topics service interface is defined in figure 11.

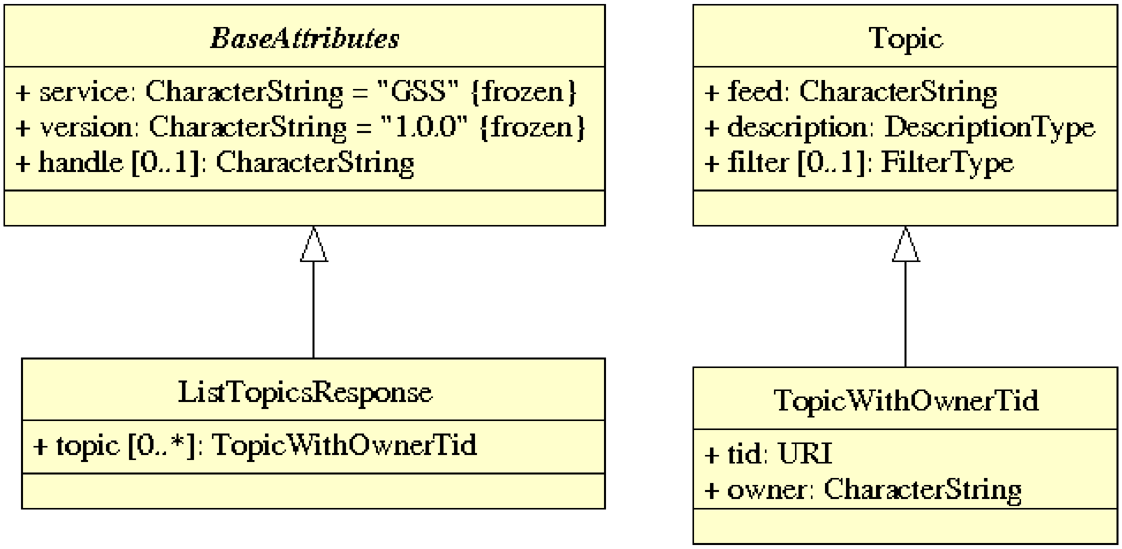

BaseAttributes
+ service: CharacterString = "GSS" {frozen}
+ version: CharacterString = "1.0.0" {frozen}
+ handle [0..1]: CharacterString
Topic
+ feed: CharacterString
+ description: DescriptionType
+ filter [0..1]: FilterType
ListTopicsResponse
+ topic [0..*]: TopicWithOwnerTid
TopicWithOwnerTid
+ tid: URI
+ owner: CharacterString

Figure 11 – ListTopicsResponse UML class diagram

The response to the other operations in the Topics service interface is defined in figure 12.

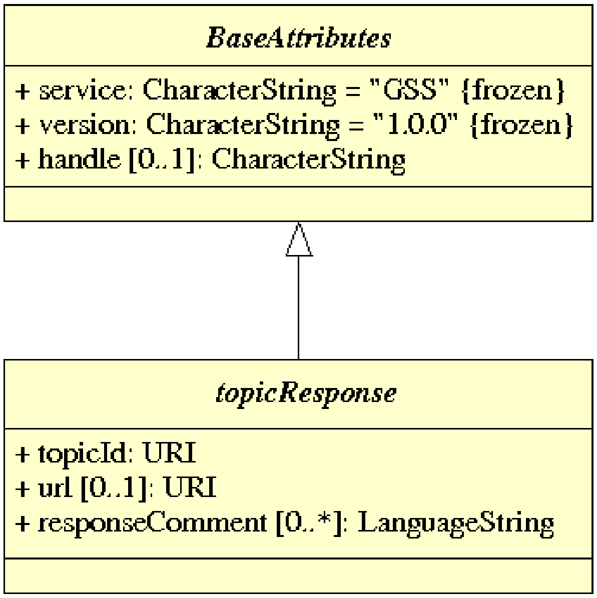

BaseAttributes
+ service: CharacterString = "GSS" {frozen}
+ version: CharacterString = "1.0.0" {frozen}
+ handle [0..1]: CharacterString
topicResponse
+ topicId: URI
+ url [0..1]: URI
+ responseComment [0..*]: LanguageString

Figure 12 – Topic management responses UML class diagram

6.5.7 Active notification

A Geosynchronization service manages three primary event channels (see 6.3.2) and any number of ad-hoc event channels or topics (see 6.3.3).

The subscription interface allows users to register their interest in being notified whenever an event is registered in an event channels that a GSS manages.

Table 6 lists the operations of the subscriptions interface and their descriptions.

Table 6 – Active notification API

Operation Name	Description
CreateTopic	Create an ad-hoc event channel based on one of the three Geosynchronization feeds. A topic can include non-spatial, spatial and temporal predicates to identify entries that satisfy certain criteria. For example, a topic can be created that filters the Proposed Change channel for entries that lie within a specific geographic area.
RemoveTopic	Delete a topic from the service.
ListTopics	Get a list of all available topics.
Subscribe	Subscribe to a topic for the purpose of notifications. Whenever a new entry is created in a base feed that satisfies the topic predicates, a notification action is performed by the Geosynchronization service to all subscribers.
ListSubscriptions	Get a list of subscriptions to which the current user has subscribed.
PauseSubscription	Pause notifications for the specified subscription.

ResumeSubscription	Restart notifications for the specified subscription.
CancelSubscription	Permanently stop receiving notifications for the specified subscription.

Figure 13 defines the operations of the Subscription service interface and their parameters.

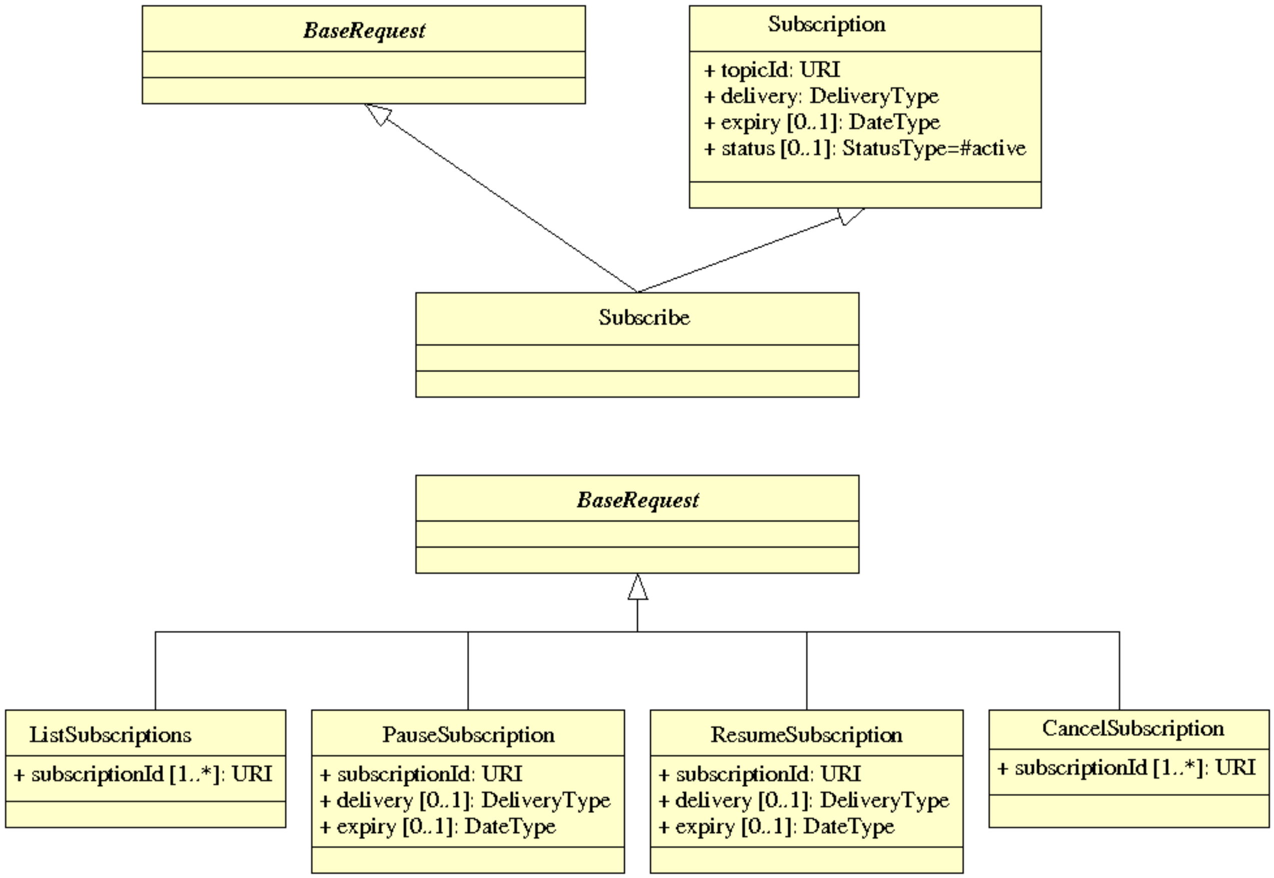

BaseRequest
Subscription
+ topicId: URI
+ delivery: DeliveryType
+ expiry [0..1]: DateType
+ status [0..1]: StatusType=#active
Subscribe
BaseRequest
ListSubscriptions
+ subscriptionId [1..*]: URI
PauseSubscription
+ subscriptionId: URI
+ delivery [0..1]: DeliveryType
+ expiry [0..1]: DateType
ResumeSubscription
+ subscriptionId: URI
+ delivery [0..1]: DeliveryType
+ expiry [0..1]: DateType
CancelSubscription
+ subscriptionId [1..*]: URI

Figure 13 – Active notification UML class diagram

The response to the ListSubscriptions operation of the Subscription service interface is defined in figure 14.

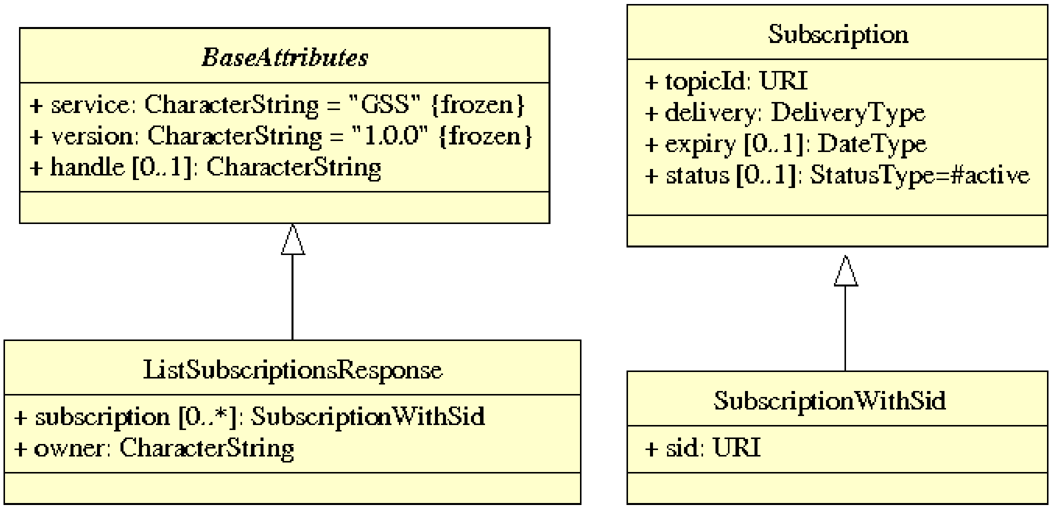

BaseAttributes
+ service: CharacterString = "GSS" {frozen}
+ version: CharacterString = "1.0.0" {frozen}
+ handle [0..1]: CharacterString
Subscription
+ topicId: URI
+ delivery: DeliveryType
+ expiry [0..1]: DateType
+ status [0..1]: StatusType=#active
ListSubscriptionsResponse
+ subscription [0..*]: SubscriptionWithSid
+ owner: CharacterString
SubscriptionWithSid
+ sid: URI

Figure 14 – ListSubscriptionsResponse UML class diagram

The response to the other operations in the Subscription service interface is defined in figure 15.

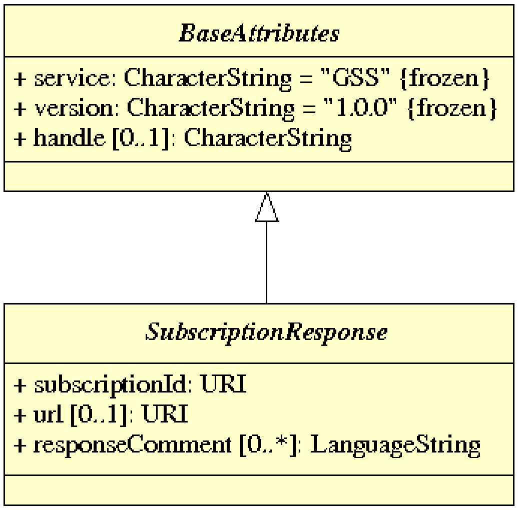

BaseAttributes
+ service: CharacterString = "GSS" {frozen}
+ version: CharacterString = "1.0.0" {frozen}
+ handle [0..1]: CharacterString
SubscriptionResponse
+ subscriptionId: URI
+ url [0..1]: URI
+ responseComment [0..*]: LanguageString

Figure 15 – Active notification responses UML class diagram

6.5.8 Synchronization

6.5.8.1 Description

Synchronization is the process whereby one or more target feature types maintain an up-to-date copy of data stored in a source feature type. All feature types shall be accessible via a WFS.

The desire to initiate synchronization is signaled using the Subscribe operation (see 6.5.7).

Synchronization is supported by the information maintained in the Replication event channel and can be characterized by what is synchronized and how it is synchronized.

Several modes of synchronization can be envisioned based on what is synchronized:

1. A target server can synchronize with a source server exactly mirroring the data and schema found in the source server.

2. A target server can synchronize with a source server exactly mirroring the schema but only synchronizing a portion of the source data. Which portion is synchronized may be specified by filtering the Replication event channel.

3. A target server can synchronize with a source server exactly mirroring the data from the source server but applying the changes from the source to a different schema in the target server.

4. A target server can synchronize with a source server synchronizing only a portion of the source server and applying the changes from the source server to a different schema in the target server.

There are also two ways that synchronization can be executed:

1. Events can be read from the Replication channel by some agent outside the GSS and then applied to the target server. This external agent would use the GSS to filter the Replication channel to obtain events of interest, it would manipulate them as required (e.g. transforming the schemas from the source data store to the target data store) and then applying them to the target server.

2. The GSS can act as the synchronization agent and automatically apply changes to one or more target servers that have subscribed to the Replication event channel or a topic created on the Replication channel.

6.5.8.2 Feature identification

An important aspect of synchronization is determining the equivalence of a feature in a source data store with a feature in a target data store.

There is a high probability the feature identifiers maintained in the source data store will not be the same as those found in the target data store and so some manner of determining equivalence is required.

Thus far two strategies have been tested:

1. Global feature identifiers – This approach has some external authority assigning a global identifier to each feature that both the source and target data stores are responsible for maintaining. This allows feature equivalence to be determined by simply comparing the global identifiers of the two features. If they identifiers are equal then the corresponding records in the source and target data store represent the same feature. This was the approach used in the CGDI Pilot.

2. Property comparison – This approach identifies corresponding features in the source and target data stores by comparing all the feature properties and most importantly the geometry. If all the properties are equal and the geometry of the features is the same then they are considered to be equivalent features. This approach has been tested by the Carbon Project

With the existence of the AcceptChange operation, another possibility presents itself:

1. Mapping tables -- A requirement of the AcceptChange operation is that the GSS behave like a client and apply each change event in the Replication event channel to each subscribed target data store. As such, the GSS has access to the feature identifiers of all newly created features and can thus maintain a mapping between the feature identifiers on a source data store and the feature identifiers on a synchronizing target data store.

6.5.8.3 Initial seeding

T.B.D. → There is a need to describe how a synchronizing target server is seeded with data before synchronization commences. Do we assume that the target starts empty, is seeded with data from the source and then synchronizes thereafter? Do we assume that the target already exists? In that case what does synchronization mean since the two data sets (source and target) might be quite different!

7 Basic service elements

7.1 Introduction

This clause specifies aspects of Geosynchronization server behavior that are independent of particular operations or are common to several operations.

7.2 Version numbering and negotiation

7.2.1 Version number form and value

The Geosynchronization Service defines a protocol version number. The version number applies to the XML schema and the request encodings defined in this document. The version number contains three non-negative integers, separated by decimal points, in the form "*x*.*y*.z". The numbers "*y*" and "z" shall not exceed 99.

Implementations of this service based on this document shall use the value "1.0.0" as the protocol version number.

7.2.2 Version number negotiation

As the first incarnation of this service, implementations based on this document will not perform any version negotiation.

7.3 Parameter encoding

For KVP-encoded requests, servers that implement this candidate standard shall consider parameter names as case in-sensitive. For example, the parameter names GetEntries and GETENTRIES shall be considered equivalent. Parameter values, on the other hand, shall be considered case sensitive. For example, the values CHANGEFEED and ChangeFeed should not be considered equivalent.

For XML-encoded requests, servers shall validate the requests against the XML-schema fragments defined in this candidate standard. In general this means that parameter names shall be case sensitive. However, as with KVP-encoded requests, parameter values shall be considered base sensitive.

7.4 SOAP

7.4.1 Declaring support

A GSS that suppose SOAP shall declare this support via the conformance declaration in the capabilities document (see 9.3.1.2.3).

7.4.2 SOAP envelope

In the SOAP interface, the XML element that contains the request data or response data of an operation shall be placed in the body of a SOAP envelope as specified in SOAP Version 1.2.

The following XML fragment illustrates a SOAP 1.2 envelope with an empty body:

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope
    xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
   <soap:Header>
   </soap:Header>
   <soap:Body>
   </soap:Body>
</soap:Envelope>
```

The soap:Body element contains the XML encodings of the GSS operation request data or response data as specified in sub-clause 9.3.

The optional soap:Header element is reserved for optional elements that are beyond the scope of the GSS specification, such as identity tokens or licensing information.

As recommended by [OGC 08-009r1], the document/literal style shall be used for the SOAP bindings of the operation requests.

This candidate standard does not support KVP-encoded requests encapsulated within a SOAP envelope.

7.4.3 Exceptions using SOAP

As described in [OGC 06-094r1], if a GSS encounters an error while processing an operation request encoded in a SOAP envelope, the server shall generate a SOAP 1.2 response message where the content of the Body element is a Fault element containing an ExceptionReport element (see 7.7). This shall be done using the following XML fragment:

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope
 xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
   <soap:Body>
      <soap:Fault>
         <soap:Code>
            <soap:Value>soap:Server</soap:Value>
         </soap:Code>
         <soap:Reason>
            <soap:Text>
            A server exception was encountered.
            </soap:Text>
         </soap:Reason>
```

```
            <soap:Detail>
               <ows:ExceptionReport
                  xmlns:ows="http://www.opengis.net/ows/1.1">
                  …
               </ows:ExceptionReport>
            </soap:Detail>
         </soap:Fault>
      </soap:Body>
   </soap:Envelope
```

The Value element (child of the Code element) shall have the value "soap:Server" (where “soap” stands for an arbitrary namespace prefix referring to the SOAP 1.2 namespace “http://www.w3.org/2003/05/soap-envelope”) indicating that this is a server exception. The Text element (child of the Reason element) shall have the value "A Server exception was encountered."

This fixed string is used since the details of the exception shall be specified in the Detail element using an ows:ExceptionReport element.

7.5 Base request type

7.5.1 XML encoding

The following XML Schema fragment defines the base request type:

```
<xsd:complexType name="BaseRequestType" abstract="true">
   <xsd:attributeGroup ref="gss:BaseAttributes"/>
</xsd:complexType>
<xsd:attributeGroup name="BaseAttributes">
   <xsd:attribute name="service" type="ows:ServiceType"
                  use="required" fixed="GSS"/>
   <xsd:attribute name="version" type="xsd:string"
                  use="required" fixed="1.0.0"/>
   <xsd:attribute name="handle"
                  type="xsd:string" use="optional"/>
</xsd:attributeGroup>
<xsd:simpleType name="DateType">
   <xsd:union memberTypes="xsd:date xsd:dateTime"/>
</xsd:simpleType>
```

The service parameter shall be set to "GSS" indicating that the Geosynchronization server is being invoked.

For this candidate standard, the version shall be set to "1.0.0".

The optional handle parameter may be used to associate a mnemonic with the request for the purpose of error reporting.

The gss:DateType allow dates to be specified with or without time. If time is not specified the implied time is 00:00:00Z.

All dates and time shall be specified in UTC or zulu time.

7.5.2 KVP encoding

The following table defines KVP parameters common to all requests except the GetCapabilities request.

Table 7 – Base request parameters

Parameter Name	Type	O/M	Default Value
SERVICE	string	M	Fixed value of GSS
VERSION	string	M	Fixed value of 1.0.0
HANDLE	string	O	

7.6 Base response types

7.6.1 Introduction

The base response types declare the schema of the response for most GSS operations. The response schemas allow operations, to indicate a subscription, topic or entry identifier in the response. They also allow a link to a related resource and a comment to be specified. Multiple gss:ResponseComment elements may appear in a response in order to support multiple languages

7.6.2 Subscription response

The following XML Schema fragment defines the XML-encoding for the response to the Subscribe, PauseSubscription, ResumeSubscription and CancelSubscription operations.

```
<xsd:complexType name="SubscriptionResponseType">
   <xsd:sequence>
      <xsd:element name="SubscriptionId" type="xsd:anyURI"/>
      <xsd:element name="Url" type="xsd:anyURI" minOccurs="0"/>
      <xsd:element name="ResponseComment" type="ows:LanguageStringType"
                   minOccurs="0" maxOccurs="unbounded"/>
   </xsd:sequence>
   <xsd:attributeGroup ref="gss:BaseAttributes"/>
</xsd:complexType>
```

The schema allows a subscription identifier to be specified in the response. It also allows a link to a related resource and a comment to be specified. Multiple gss:ResponseComment elements may appear in a response in order to support multiple languages

7.6.3 Topic response

The following XML Schema fragment defines the XML-encoding for the response to the CreateTopic and RemoveTopic operations.

```
<xsd:complexType name="TopicResponseType">
   <xsd:sequence>
      <xsd:element name="TopicId" type="xsd:anyURI"/>
      <xsd:element name="Url" type="xsd:anyURI" minOccurs="0"/>
      <xsd:element name="ResponseComment" type="ows:LanguageStringType"
```

```
                      minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
      <xsd:attributeGroup ref="gss:BaseAttributes"/>
   </xsd:complexType>
```

The schema allows a topic identifier to be specified in the response. It also allows a link to a related resource and a comment to be specified. Multiple gss:ResponseComment elements may appear in a response in order to support multiple languages

7.6.4 Entry response

The following XML Schema fragment defines the XML-encoding for the response to the AcceptChange and RejectChange operations.

```
   <xsd:complexType name="EntryResponseType">
      <xsd:sequence>
         <xsd:element name="EntryId" type="xsd:anyURI"/>
         <xsd:element name="Url" type="xsd:anyURI" minOccurs="0"/>
         <xsd:element name="ResponseComment" type="ows:LanguageStringType"
                      minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
      <xsd:attributeGroup ref="gss:BaseAttributes"/>
   </xsd:complexType>
```

The schema allows an entry identifier to be specified in the response. It also allows a link to a related resource and a comment to be specified. Multiple gss:ResponseComment elements may appear in a response in order to support multiple languages

7.7 Exception reporting

In the event that a Geosynchronization Service encounters an error while processing a request or receives an unrecognised request, it shall generate an XML document indicating that an error has occurred. The format of the XML error response is specified by, and shall validate against, the exception response schema defined in clause 8 of the OWS Common Implementation Specification [OGC 06-121r3].

Example

The following is an example of an exception report. This exception indicates that the first insert statement failed because of a missing closing XML tag in the request.

```
<?xml version="1.0" ?>
<ExceptionReport
version="1.1.0"
xmlns="http://www.opengis.net/ogc"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="owsExceptionReport.xsd">
<Exception code="999" locator="Line 101">
<ExceptionText>Feed Error:Invalid feed type specified</ExceptionText>
</Exception>
</ExceptionReport>
```

8 Geosynchronization overview

Figure 16 shows a wiring diagram that illustrates the basic use case for a Geosynchronization service.

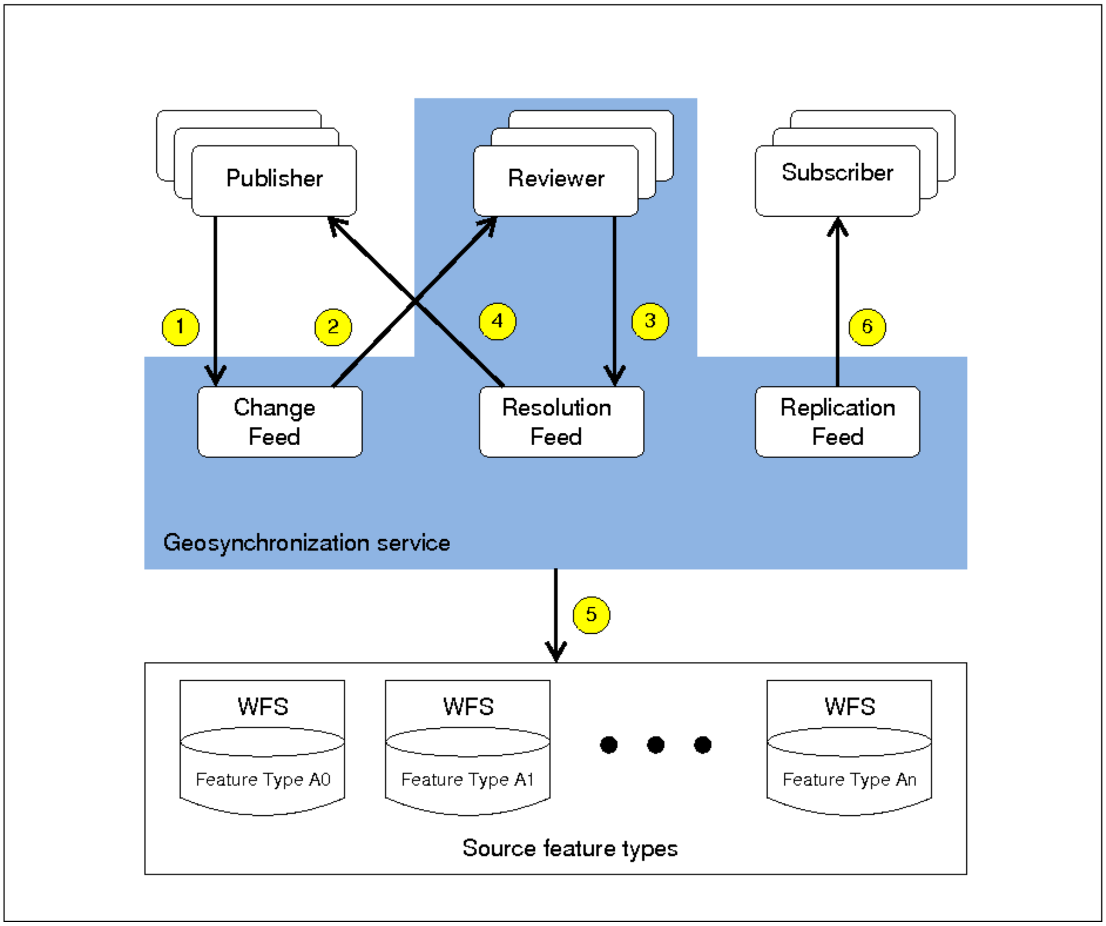


Figure 16 – Geosynchronization Wiring Diagram

This candidate standard assumes that a data provider's data store(s) is(are) accessible through a WFS interface and thus all feature types in the diagram are assumed to accessible via web feature services.

Within the figure the following actors are represented:

1. At the top left are collaborating participants or data publishers.

2. At the bottom, a set of feature types accessible via a web feature servers (i.e. A0, A1, …, An) that are within the control domain of the GSS. Transactional access to these feature types is mediated by the GSS.

3. At the top right, subscribers who are synchronizing feature types that are outside the control domain of the GSS. Subscribers either access the replication feed themselves of have the GSS process the replication feed to synchronize data between features types within the control domain of the GSS (i.e. A0, A1, … ,An) and feature types outside the control domain of the GSS.

With reference to the numbers within the yellow circles, the basic GSS use case proceeds as follows:

1. A data publisher posts a proposed change to the GSS for one of the feature types within the GSS's domain of control (i.e. A0, A1, … ,An).

2. The proposed change is read from the Change Feed and reviewed to ensure that it satisfies correctness and quality standards established by the data provider whose feature type is to be updated (i.e. one of A0, A1, … An in this diagram). The review process can be manual or automatic.

3. If the proposed change is rejected, that event is registered in the Resolution Feed and a notification is sent to all interested subscribers.

4. If the propose change is accepted, that event is registered in the Resolution Feed…

5. …and the GSS applies the proposed change to the intended feature type.

6. …and the GSS registers all the changes made to the target feature type (i.e. A0, A1, … ,An) in the Replication Feed. It also sends a notification to all interested subscribers that the proposed change has been accepted and that changes have been made to one of the feature types within the Geosynchronization server's control domain (i.e. A0, A1, … An).

7. Changes posted to the Replication feed may also be applied, by the GSS, to any external feature type that has subscribed for synchronization. The changes are first passed through the schema translation engine to, if necessary, convert them to the schema of each target feature type.

8. If a schema translation is required, transformation metadata is read by the GSS and the transaction is transformed to the schema of the target feature type …

9. …and then the change is applied to the target feature type via WFS.

9 Implementation

9.1 Introduction

As describe in the abstract model, a GSS service has four distinct functional capabilities.

It manages events and event channels for disseminating those events. This includes the capability to create ad-hoc event channels called topics.

It implements a change management API that allows proposed changes to be reviewed, either manually or automatically, and either be accepted or rejected.

It implements a query API that allows event channels to be queried based on spatial and non-spatial predicates where spatial predicates can include both geometric and temporal predicates.

It may implement a subscription interface to allow interested parties to be notified of events and also allows the GSS to act as a synchronization agent keeping one or more target feature type synchronized with a source feature type within the control domain of the GSS.

This clause describes an implementation of a GSS where:

- Events are implemented as ATOM entries.

- Event channels and topics are implemented as ATOM feeds.

- Feed management may be performed using the ATOM Publishing Protocol (see IETF RFC 5023) although other bindings are described as well.

- A query operation is defined that supports both spatial and non-spatial predicates. The operation is compatible with OpenSearch and the OpenSearch geo and time extensions.

- A subscription API is defined to create and manage subscriptions and to enable synchronization.

9.2 Resources

9.2.1 Introduction

Event channels shall be implemented as ATOM feeds (see IETF RFC 4287) with geographic components being encoded using GeoRSS (see OGC 06-050r3). Feeds are maintained by the Geosynchronization service on behalf of participants collaborating in an environment composed of distributed WFS servers. In addition to feeds, a GSS also maintains topics which are filtered feeds.

There are thee main ATOM feeds used to implement Geosynchronization. A Proposed Change feed, Resolution feed and Replication feed. Interested parties can subscribe to any of these feeds, or topics created on the feeds, to be notified whenever entries are created or modified.

9.2.2 Proposed change feed

9.2.2.1 Description

The Proposed Change Feed shall be used to record suggested or proposed changes to a data provider's data store. Any authorized user can post entries to the Proposed Change Feed. Changes shall be expressed as WFS transactions upon features in a data provider's data store.

9.2.2.2 Encoding

Table 8 describes how change feed entries shall be encoded. If an ATOM element does not appear in this table then is may be omitted or populated as described in the ATOM specification (see IETF RFC 4287).

Table 8 – Change Feed elements

ATOM Entry Element	M/O	Description
content	M	The content element shall contain a description of the proposed change. This can be in the form of narrative text or a GML fragment indicating what the state of the feature should be. If the proposed change is in the form of a narrative then the "type" attribute shall be set to "text/plain". If the proposed change is in the form of a GML fragment then the "type" attribute shall be set to "text/xml subtype=gml/X.X.X" where X.X.X indicates the version of GML being used and may include additional components for profiles. For this specification, the GML version shall be "text/xml; subtype=gml/3.1.1/profiles/gmlsf/1.0.0/0"
id	M	The Id element shall be present. The content of the Id element may be empty. If the content is empty then the server shall generate an id (in the form of a uuid) for the entry. If the content is not empty then it shall be an identifier for the entry generated by

ATOM Entry Element	M/O	Description
		the client in the form of a uuid. The server shall preserve client generated uuid's. Example: urn:uuid:26b3c614- b761-4f7c-a011-c2c15642cc3a.
link	M	There shall be a single link element with the value of the "rel" attribute set to "targetServer" and the value of the "href" attribute set to the GetCapabilities URL of the WFS to which this change is to be applied. Example: <atom:link rel="targetServer" href=http://demo.cubewerx.com/cubeserv/cubeserv.cgi?config=DEMO&datastore=DEMO&service=WFS&request=GetCapabilities/> This allows the change feed to be filtered for entries that apply to a particular server.
link	M	There shall be one or more link elements with the value of the "rel" attribute set to "targetFeature" and the value of the "href" attribute set to point to a feature that is being changed. There shall be one link for each feature changed in the proposed change. Example: <atom:link rel="targetFeature" href=http://demo.cubewerx.com/cubeserv/cubeserv.cgi?config=DEMO&datastore=DEMO&service=WFS&version=1.1.0&request=GetFeature&typeName=Roads&featureId=10/>
summary	O	May contain a short description about the change. Similar to a log entry someone might put into an svn or cvs log. Example: "Telephone pole moved 10m North as per observed position from onsite crew."
title	M	Shall contain a title describing what is being changed. For example: "Proposed change for feature INWATERA_1M.1013.".
updated	M	The date the change was made. All time shall be zulu time.
georss:where	O	If the feature being changed is non-spatial then the element shall not be specified. If the feature is spatial and includes a single point geometry then that geometry shall be used as the value of element. Otherwise a gml:Envelope containing the geometry shall be used as the value of the georss:where element.

9.2.3 Resolution feed

9.2.3.1 Description

The Resolution feed shall be used to provide feedback to users who have posted items to the Proposed Change Feed. When a proposed change is accepted or rejected, the Resolution Feed is updated to indicate the disposition of the change. Users can subscribe to the Resolution feed to be notified when a decision is reached about their proposed change(s).

9.2.3.2 Encoding

Table 9 describes how resolution feed entries shall be encoded. If an ATOM element does not appear in this table then is may be omitted or populated as described in the ATOM specification (see IETF RFC 4287).

Table 9 – Resolution feed elements

ATOM Entry Element	M/O	Description
author	M	Shall be populated with the contact information for the data custodian that may be contacted to ask questions about the resolution.
category	M	There shall be at one category element in the entry indicating the disposition of the originating change proposal. The "term" attribute shall have a value of "Accepted" if the originating change proposal was accepted or "Rejected" if the originating change proposal was rejected. The scheme attribute shall have the value: "http://www.opengis.org/geosync/resolutions"
content	M	If the change proposal was rejected then the content element may contain a narrative describing why it was rejected. If the change proposal was accepted but modified then the content element may contain a narrative describing the modifications.
id	M	An id for the resolution feed entry.
link	M	A link with the value of the "rel" attribute set to "disposition" and the value of the href attribute set to point to the change feed entry that has been accepted or rejected.
title	M	The value of the title element shall be 'Proposal Accepted' or 'Proposal Rejected' depending of the disposition of the originating change proposal.
updated	M	The date the change was made. All time shall be zulu time.
georss:where	O	If the originating change proposal was rejected, the georss:where element shall be omitted. If the originating change proposal was accepted, the georss:where element shall have a value according to the following rules: If the modified feature is non-spatial then the element shall not be specified. If the modified feature is spatial and includes a single point geometry then that geometry shall be used as the value of element. If the modified feature is spatial and includes a non-single point geometry then a gml:Envelope containing the geometry shall be used as the value of the <georss:where>element.

9.2.4 Replication feed

9.2.4.1 Description

The Replication Feed is a log of changes that have been applied to a data provider's data store. Changes shall be recorded as WFS transactions each describing a change to a single feature. Interested parties can then use the entries in the Replication feed to synchronize a local copy of the data with the data provider's data store. Synchronization can be performed by an agent external for the GSS or interested parties can subscribe to the Replication Feed or a topic thereof and have the GSS perform the synchronization.

9.2.4.2 Encoding

Table 10 describes how replication feed entries shall be encoded. If an ATOM element does not appear in this table then is may be omitted or populated as described in the ATOM specification (see IETF RFC 4287).

Table 10 – Replication feed elements

ATOM Entry Element	M/O	Description
author	M	Shall be populated with a responsible party at the source server that can be contacted in case of problems.
category	M	There shall be at one category element in the entry indicating the type change to be applied to a geographic feature. The "term" attribute shall have a value of "insert", "update" or "delete". The scheme attribute shall have the value: "http://www.opengis.org/geosync/actions"
content	M	The content element shall contain the WFS transaction action (Insert, Update or Delete) that has been applied to the feature. The value of the type attribute shall be "text/xml; subtype=gml/X.X.X" where X.X.X indicates the version of GML being used to encode the feature and may include additional components for profiles. For this specification, the GML version shall be "text/xml; subtype=gml/3.1.1/profiles/gmlsf/1.0.0/0"
id	M	The id element shall be present. The content of the id element may be empty. If the content is empty then the server shall generate an id (in the form of a uuid) for the entry. If the content is not empty then it shall be an identifier for the entry generated by the client in the form of a uuid. The server shall preserve client generated uuid's. Example: urn:uuid:26b3c614- b761-4f7c-a011-c2c15642cc3a.
link	O	There may be one link element in the entry, with the value of the "rel" attribute set to "feature", that fetches the feature from the source server. This element shall only be populated if the server supports versioning and can fetch the version of the feature that contains the change described by the entry. Otherwise the link shall not be populated.
summary	O	May contain a short description about the change. Similar to a log entry someone might put into an svn or cvs log.

ATOM Entry Element	M/O	Description
		Example: "Telephone pole moved 10m North as per observed position from onsite crew."
title	M	Shall be populated with the value: "XXX of feature YYY" where "XXX" is one of Insert, Update or Delete and YYY is the feature id of the feature being affected.
updated	M	The date the change was made. All time shall be zulu time.
georss:where	O	If the feature is non-spatial then the element shall not be specified. If the feature is spatial and includes a single point geometry then that geometry shall be used as the value of element. Otherwise a gml:Envelope containing the geometry shall be used as the value of the <georss:where>element.

9.2.5 Resource URIs

9.2.5.1 Feed URL

Each feed that the Geosynchronization server offers shall be identified by a URL.

A feed's URL shall be advertised in the server's AtomPub Service Document which can be obtained from the server's capabilities document or accessed directly as described in 9.3.1.

The final step in the feed URL shall be the name of the feed. For a GSS this will be one of CHANGEFEED, RESOLUTIONFEED or REPLICATIONFEED.

Example:

In the following example, the blue text is the server's base URL and the green text is the name of the feed.

```
http://demo.cubewerx.com/projects/ows7/gss/gss.cgi/ows7/GSS/1.0.0/OWS7/CHANGEFEED
```

9.2.5.2 Entry URL

Each entry in a feed shall be accessible via a URL that is a combination of the base URL of the service and the id of the feed entry.

Example:

In the following example, the blue text is the server's base URL, the green text is the name of the feed (CHANGEFEED in this case) and the black text at the end is the id of the entry.

```
http://demo.cubewerx.com/projects/ows7/gss/gss.cgi/ows7/GSS/1.0.0/OWS7/entry/CHANGEFEED/urn:uuid:e7b2da53-2092-48a8-ac31-dd461e99ad8f
```

9.2.5.3 Topic URL

The topic URL identifies the collections of topics offered by a GSS.

The topic URL shall be composed of the server's base URL, a path element named "topic" and the topic identifier assigned by the GSS when the topic was created.

Example:

In the following example, the blue text is the server's base URL, the red text is the path element "topic" and the back text at the end is the id of the topic.

```
http://demo.cubewerx.com/projects/ows7/gss/gss.cgi/ows7/GSS/1.0.0/OWS7/topic/ur
n:uuid:2dcef869-3472-429b-8ade-af57b0598bf8
```

9.2.5.4 Topic entry URL

Because a topic is simply a filtered view of a feed, topic entry URLs shall be the same as entry URLs (see 9.2.5.2).

9.2.5.5 Subscription URL

The subscription URL identifies the collection of subscriptions that the GSS is managing.

The subscription URL shall be composed of the server's base URL, the path element "subscription" and the subscription identifier assigned by the GSS when the subscription was created.

Example:

In the following example, the blue text is the server's base URL, the red text is the path element "topic" and the back text at the end is the id of the topic.

```
http://demo.cubewerx.com/projects/ows7/gss/gss.cgi/ows7/GSS/1.0.0/OWS7/subscrip
tion/urn:uuid:45ce3a31-3004-4992-a9b1-2732c66ed34d
```

9.2.6 Resource encoding

9.2.6.1 Feed and entry encoding

ATOM feeds and entries shall be encoded as described in IETF RFC 5023. In addition, servers shall follow the encoding rules described in sub-clauses 9.2.2, 9.2.3 and 9.2.4.

9.2.6.2 Topic encoding

The following XML Schema fragment shall define the XML encoding for a topic:

```
<xsd:element name="Topic" type="gss:TopicType"/>
<xsd:complexType name="TopicType">
   <xsd:sequence>
      <xsd:element name="Feed" type="xsd:anyURI"/>
      <xsd:element name="Description" type="ows:DescriptionType"/>
      <xsd:element ref="fes:Filter" minOccurs="0"/>
   </xsd:sequence>
</xsd:complexType>
```

The gss:Feed element encodes the resource URL of the feed (see 9.2.5.1) upon which the topic was created..

The gss:Description element shall be used include a human-readable description of the topic. At the very least, the ows:Title element shall be used to assign a human-readable title to the topic.

The fes:Filter element may be used to specify predicates that determine which feed entries appear in the topic. For example, a spatial predicate may be specified to restrict the entries presented to those that lie within an area of interest.

9.2.6.3 Subscription encoding

A subscription signals to a GSS that a subscriber wishes to be notified whenever new entries satisfy the predicates specified when the topic was created.

The following XML Schema fragment shall define the XML encoding for a subscription:

```
<xsd:element name="Subscription" type="gss:SubscriptionType"/>
<xsd:complexType name="SubscriptionType">
   <xsd:sequence>
      <xsd:element name="TopicId" type="xsd:anyURI"/>
      <xsd:element name="Delivery" type="gss:DeliveryMethodType"/>
      <xsd:element name="Expiry"
                   type="gss:DateType" minOccurs="0"/>
   </xsd:sequence>
   <xsd:attribute name="status" type="gss:SubscriptionStatusType"
                  use="optional" default="active"/>
</xsd:complexType>
<xsd:simpleType name="DeliveryMethodType">
   <xsd:restriction base="xsd:string">
      <xsd:pattern value="mailto:[^@]+@[^@]+\.[^@]+"/>
      <xsd:pattern value="xmpp:[^@]+@[^@]+\.[^@]+"/>
      <xsd:pattern value="sync:http://(.)*"/>
   </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="SubscriptionStatusType">
   <xsd:restriction base="xsd:string">
      <xsd:pattern value="active"/>
      <xsd:pattern value="paused"/>
   </xsd:restriction>
</xsd:simpleType>
```

The gss:TopicId element shall contain the identifier of the topic.

The gss:Delivery element indicates where a subscriber should be notified whenever new entries appear in the topic.

The gss:Expiry element sets a time limit on how long the server shall continue sending notifications to the subscriber. If not specified, notification shall continue until the subscription is explicitly cancelled.

The status attribute shall be used indicate the status notification status of the subscription. A value of "active" means that notifications are being sent. A value of "paused" indicates that the subscription is temporarily stopped and no notifications are being sent.

9.3 Service interface

9.3.1 Discovery

9.3.1.1 Introduction

In order to commence interacting with a Geosynchronization server, a client needs to discover metadata about the service that describes which protocol version(s) the server supports, what operations are available, what their access points are, which resource are available, etc.

OGC capabilities documents, AtomPub Service Documents and OpenSearch Description documents are designed to provide the necessary information although using different schemas and providing a different amount of detail about the server.

In order to promote a high level of compatibility with existing OGC services, Geosynchronization servers shall provide an OGC capabilities document.

Embedded within that capabilities document shall be an AtomPub Service Document and an OpenSearch Description document.

In order to conform to the AtomPub specification (see IETF RFC 5023), the embedded AtomPub Service Document shall also be independently addressable via URL.

In order to conform to the OpenSearch specification (see OpenSearch), the embedded OpenSearch Description document shall also be independently addressable via URL.

9.3.1.2 GetCapabilities operation

9.3.1.2.1 Introduction

The GetCapabilities operation generates a service metadata document describing a Geosynchronization service provided by a server.

All Geosynchronization services shall implement the GetCapabilities operation.

9.3.1.2.2 Request encoding

The request encoding of the GetCapabilities request shall be as specified in sub-clause 7.2.2 of OGC document 06-121r3.

9.3.1.2.3 Response encoding

The root element of the response to a GetCapabilities request is the gss:Capabilities element which is declared by the following XML Schema fragment:

```
<xsd:element name="Capabilities"
             type="gss:CapabilitiesType"/>
<xsd:complexType name="CapabilitiesType">
   <xsd:complexContent>
      <xsd:extension base="ows:CapabilitiesBaseType">
         <xsd:sequence>
            <xsd:any namespace="http://www.w3.org/2007/app"
                     processContents="lax"/>
            <xsd:any namespace="http://a9.com/-/spec/opensearch/1.1/"
                     processContents="lax"/>
            <xsd:element ref="gss:ConformanceDeclaration"/>
```

```
               <xsd:element ref="fes:Filter_Capabilities" minOccurs="0"/>
            </xsd:sequence>
         </xsd:extension>
      </xsd:complexContent>
   </xsd:complexType>
   <xsd:element name="ConformanceDeclaration">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element name="ConformanceClass"
                         minOccurs="28" maxOccurs="28">
               <xsd:complexType>
                  <xsd:simpleContent>
                     <xsd:extension base="xsd:boolean">
                        <xsd:attribute name="name"
                                       type="gss:ConformanceClassName"/>
                     </xsd:extension>
                  </xsd:simpleContent>
               </xsd:complexType>
            </xsd:element>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
   <xsd:simpleType name="ConformanceClassName">
      <xsd:restriction base="xsd:string">
         <xsd:pattern value="Discovery_OGC_POX"/>
         <xsd:pattern value="Discovery_OGC_KVP"/>
         <xsd:pattern value="Discovery_REST"/>
         <xsd:pattern value="Discovery_SOAP"/>
         <xsd:pattern value="Transaction_OGC_POX"/>
         <xsd:pattern value="Transaction_OGC_KVP"/>
         <xsd:pattern value="Transaction_REST"/>
         <xsd:pattern value="Transaction_SOAP"/>
         <xsd:pattern value="Query_OGC_POX"/>
         <xsd:pattern value="Query_OGC_KVP"/>
         <xsd:pattern value="Query_REST"/>
         <xsd:pattern value="Query_SOAP"/>
         <xsd:pattern value="ChangeManagement_OGC_POX"/>
         <xsd:pattern value="ChangeManagement_OGC_KVP"/>
         <xsd:pattern value="ChangeManagement_REST"/>
         <xsd:pattern value="ChangeManagement_SOAP"/>
         <xsd:pattern value="TopicManagement_OGC_POX"/>
         <xsd:pattern value="TopicManagement_OGC_KVP"/>
         <xsd:pattern value="TopicManagement_REST"/>
         <xsd:pattern value="TopicManagement_SOAP"/>
         <xsd:pattern value="ActiveNotification_OGC_POX"/>
         <xsd:pattern value="ActiveNotification_OGC_KVP"/>
         <xsd:pattern value="ActiveNotification_REST"/>
         <xsd:pattern value="ActiveNotification_SOAP"/>
         <xsd:pattern value="Synchronization_OGC_POX"/>
         <xsd:pattern value="Synchronization_OGC_KVP"/>
         <xsd:pattern value="Synchronization_REST"/>
         <xsd:pattern value="Synchronization_SOAP"/>
      </xsd:restriction>
   </xsd:simpleType>
```

The base type, ows:CapabilitiesBaseType, is defined in the OWS Common Implementation Specification (see OGC 06-121-r3, 7.2.4).

The elements ows:ServiceIdentification, ows:ServiceProvider and ows:OperationMetadata are inherited from the base type ows:CapabilitiesBaseType.

In addition to the sections defined in sub-clause 7.4 of OGC 06-121r3, the capabilities response document shall contain the following sections:

1. WSDL section:
   The WSDL element may be used to provide a link to a WSDL (see W3C WSDL) document describing the service interface.

2. AtomPub Service Document section:
   Embedded within the OGC capabilities document shall be a AtomPub Service Document describing the available workspaces and collections as per the IETF RFC 5023. A Geosynchronization service shall advertise a single workspace containing three collections: change feed, resolution feed, and replication feed.

3. OpenSearch Description Document section:
   The GetCapabilities document shall also include an embedded OpenSearch description document (see OpenSearch).

4. ConformanceDeclaration section:
   The gss:ConformanceDeclaration sections allows a server to declare which conformance classes (see Table 1) it implements. The gss:ConformanceDeclaration element shall contain 27 gss:ConformanceClass elements, one for each conformance class defined in Table 1. If the contents of the gss:ConformanceClass is "true", this shall indicate that the server implements that conformance class. Otherwise the contents of the gss:ConformanceClass element shall be "false" indicating that the server does not implement the corresponding conformance class.

5. Filter capabilities section:
   The schema of the Filter Capabilities Section is defined in ISO 19143 and is used to advertise the expressions that may be used to form query predicates

9.3.1.3 AtomPub Service Document

AtomPub resource discovery starts with the publication of a URI for a Service Document.

AtomPub defines a Service Document as an XML document consisting of one or more Workspaces. A Geosynchronization Service Document, however, *shall* contain one and only one Workspace. A Workspace shall have a title, and shall contain Collections, which reference the actual Geosynchronization feeds (i.e. change, resolution and replication feeds).

If a Collection is editable, it shall contain accept elements, which specify what media types may be added to a Collection. In Geosynchronization parlance, this corresponds to editing and posting Feed entries.

An AtomPub Service Document shall be embedded within the server's OGC capabilities document. However, a server shall also arrange for the AtomPub service document to be directly and independently addressable via URL.

Example:

The following example uses a vendor extension on the standard GetCapabilities operation to allow independent access to a servers AtomPub service document:

```
http://www.cubewerx.com/geosync.cgi?request=GetCapabilities&sections=app:servic
e&outputFormat=application/atomsvc+xml
```

A Geosynchronization service shall use the media type of `application/atomsvc+xml` when returning the AtomPub Service Document.

A Collection may also contain Categories, which define a controlled vocabulary on the Collection. A Geosynchronization service shall advertise the terms "Accept" and "Reject" as categories for the resolution feed and the terms "Insert", "Update" and "Delete" as categories for the replication feed.

For more information on Category documents, see AtomPub clause 7.

For a full explanation of the encoding of a Service Document, see AtomPub clause 8.

Example

The following XML fragment shows the AtomPub service document for the fictitious GSS at Acme Corp..

```
<?xml version="1.0" encoding="UTF-8"?>
<service xmlns="http://www.w3.org/2007/app"
        xmlns:atom="http://www.w3.org/2005/Atom">
   <workspace>
      <atom:title>Geosynchronization Service</atom:title>
      <collection href="http://www.acme.com/gss.cgi/CHANGEFEED">
         <atom:title>Change Feed</atom:title>
         <accept>application/atom+xml;type=entry</accept>
      </collection>
      <collection href="http://www.acme.com/gss.cgi/RESOLUTIONFEED">
         <atom:title>Resolution Feed</atom:title>
         <accept>application/atom+xml;type=entry</accept>
         <app:categories
              fixed="yes"
              scheme="http://www.opengis.org/geosync/categories/res">
            <atom:category term="Accepted" />
            <atom:category term="Rejected" />
         </app:categories>
      </collection>
      <collection href="http://www.acme.com/gss.cgi/REPLICATIONFEED">
         <atom:title>Replication Feed</atom:title>
         <accept>application/atom+xml;type=entry</accept>
         <app:categories
              fixed="yes"
              scheme="http://www.opengis.org/geosync/categories/rep">
            <atom:category term="Insert" />
            <atom:category term="Update" />
            <atom:category term="Delete" />
         </app:categories>
      </collection>
      <collection href="http://www.acme.com/gss.cgi/topic">
         <atom:title>Topics</atom:title>
         <accept>text/xml</accept>
      </collection>
      <collection href="http://www.acme.com/gss.cgi/subscription">
         <atom:title>Subscriptions</atom:title>
         <accept>text/xml</accept>
      </collection>
   </workspace>
```

```
</service>
```

In this example, the server is advertising collections for the three base feeds, CHANGEFEED, RESOLUTIONFEED and REPLICATIONFEED, as well as collections form topics and subscriptions.

9.3.1.4 OpenSearch description document

A Geosynchronization service shall make available an OpenSearch description document that can be used to describe the web interface of a search engine.

A Geosynchronization service shall use the media type of `application/opensearchdescription+xml` when returning the OpenSearch Description Document.

For a full explanation of the encoding of an OpenSearch Document, see http://www.opensearch.org.

Example

The following XML fragment is an example of an OpenSearch description document.

```
<?xml version="1.0" encoding="UTF-8"?>
<OpenSearchDescription xmlns="http://a9.com/-/spec/opensearch/1.1/">
   <ShortName>Geosynchronization Feed Search</ShortName>
   <Description>Search the feeds managed by a GSS.</Description>
   <Tags>Geosynchronization feed collaboration GSS search</Tags>
   <Contact>admin@acme.com</Contact>
   <Url type="application/atom+xml"
        template="http://www.pvretano.com/gss/cubeserv.cgi?
                  service=GSS&
                  version=1.0.0&
                  request=GetEntries&
                  feed=CHANGEFEED&
                  outputFormat=ATOM&
                  startPosition={startIndex?}&
                  maxEntries={count?}&
                  searchTerms={searchTerms?}&
                  bbox={geo:box?}&
                  starttime={time:start?}&
                  temporalOp=After"/>
   <LongName>Acme Geosynchronization Service Feed Search</LongName>
   <Image height="64" width="64"
          type="image/png">http://acme.com/websearch.png</Image>
   <Query role="example" count="25" />
   <Developer>Example.com Development Team</Developer>
   <Attribution>Copyright     2005,     Acme.com,     Inc.,     All     Rights
Reserved</Attribution>
   <SyndicationRight>open</SyndicationRight>
   <AdultContent>false</AdultContent>
   <Language>en-us</Language>
   <OutputEncoding>UTF-8</OutputEncoding>
   <InputEncoding>UTF-8</InputEncoding>
</OpenSearchDescription>
```

9.3.1.5 Bindings

9.3.1.5.1 OGC bindings

9.3.1.5.1.1 POX with POST binding

Sending an XML-encoded GetCapabilities request (see 9.3.1.2.2) to a server's base URL using the HTTP POST method shall generate an XML-encoded capabilities document (see 9.3.1.2.3).

9.3.1.5.1.2 KVP with GET binding

Sending an KVP-encoded GetCapabilities request to(see 9.3.1.2.2) a server's base URL using the HTTP GET method shall generate an XML-encoded capabilities document (see 9.3.1.2.3).

9.3.1.5.1.3 Base URL for GetCapabilities operation

A client must have a-priori knowledge of the server's base URL for the GetCapabilities request in order to initiate an interaction with the server. How that a-priori knowledge is acquired is outside the scope of this candidate standard but one common method is to consult a service catalogue.

9.3.1.5.2 REST binding

There is no direct REST binding for the GetCapabilities operation defines. However, the AtomPub service document and OpenSearch description documents can be obtained as described in sub-clauses 9.3.1.3 and 9.3.1.4.

9.3.1.5.3 SOAP binding

Sending a SOAP message, whose body contains an XML-encoded GetCapabilities request (see 9.3.1.2.2) , to a server's base URL using the HTTP POST method shall generate a SOAP response message whose body contains an XML-encoded capabilities document (see 9.3.1.2.3).

9.3.2 Feed management

9.3.2.1 Introduction

The feed management API is used to insert, update and delete entries from the various feeds managed by a GSS (see 9.2).

9.3.2.2 Request encoding

9.3.2.2.1 XML encoding

The following XML Schema fragment defines the XML encoding for the Transaction operation that may be used to insert, update and/or delete entries from a feed.

```
<xsd:element name="Transaction" type="gss:TransactionType"/>
<xsd:complexType name="TransactionType">
   <xsd:complexContent>
      <xsd:extension base="gss:BaseRequestType">
         <xsd:sequence>
            <xsd:sequence minOccurs="0" maxOccurs="unbounded">
               <xsd:element ref="gss:AbstractTransactionAction"/>
            </xsd:sequence>
         </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>

  <xsd:element name="AbstractTransactionAction"
             type="gss:AbstractTransactionActionType" abstract="true"/>
  <xsd:complexType name="AbstractTransactionActionType" abstract="true">
    <xsd:attribute name="handle" type="xsd:string"/>
  </xsd:complexType>

  <xsd:element name="Insert" type="gss:InsertType"
             substitutionGroup="gss:AbstractTransactionAction"/>
  <xsd:complexType name="InsertType">
    <xsd:complexContent>
      <xsd:extension base="gss:AbstractTransactionActionType">
        <xsd:sequence>
          <xsd:any namespace="##other" processContents="lax"/>
        </xsd:sequence>
        <xsd:attribute name="feed" type="xsd:string" use="required"/>
        <xsd:attribute name="inputFormat" type="xsd:string"
                       default="application/atom+xml; type=entry"/>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>

  <xsd:element name="Update" type="gss:UpdateType"
             substitutionGroup="gss:AbstractTransactionAction"/>
  <xsd:complexType name="UpdateType">
    <xsd:complexContent>
      <xsd:extension base="gss:AbstractTransactionActionType">
        <xsd:sequence>
          <xsd:any namespace="##other" processContents="lax"/>
        </xsd:sequence>
        <xsd:attribute name="feed" type="xsd:string" use="required"/>
        <xsd:attribute name="entryId" type="xsd:anyURI" use="required"/>
        <xsd:attribute name="inputFormat" type="xsd:string"
                       default="application/atom+xml; type=entry"/>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>

  <xsd:element name="Delete" type="gss:DeleteType"
             substitutionGroup="gss:AbstractTransactionAction"/>
  <xsd:complexType name="DeleteType">
    <xsd:complexContent>
      <xsd:extension base="gss:AbstractTransactionActionType">
        <xsd:attribute name="feed" type="xsd:string" use="required"/>
        <xsd:attribute name="entryId" type="xsd:anyURI" use="required"/>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
```

A Transaction operation contain one or more Insert, Update or Delete actions that create, modify or remove entries from a feed.

The feed attribute shall be used to identify the feed that shall be operated upon by each action.

The entryId attribute shall be used, for the Update and Delete action, to identify the entry to be modified or deleted.

The inputFormat attribute shall be used to assert the input encoding of the feed entry. All servers shall support the value application/atom+xml to indicate an ATOM feed entry. Other values may be supported but this candidate standard does not assign any meaning to those values. Servers that support additional inputFormat values shall advertise those values in the server's capabilities document using the ows:Parameter element (see 06-121r3, 7.4.6).

9.3.2.2.2 KVP encoding

No KVP encoding is defined for the feed management operations.

9.3.2.3 Response encoding

The following XML-schema fragment defines the XML-encoding of the response to a Transaction request:

```
<xsd:element name="TransactionResponse">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element name="EntryId" minOccurs="0" maxOccurs="unbounded">
               <xsd:complexType>
                  <xsd:simpleContent>
                     <xsd:extension base="xsd:anyURI">
                        <xsd:attribute name="action"
                                       type="gss:ActionType" use="required"/>
                        <xsd:attribute name="featureRef"
                                       type="xsd:anyURI" use="optional"/>
                     </xsd:extension>
                  </xsd:simpleContent>
               </xsd:complexType>
            </xsd:element>
         </xsd:sequence>
         <xsd:attributeGroup ref="gss:BaseAttributes"/>
      </xsd:complexType>
   </xsd:element>
   <xsd:simpleType name="ActionType">
      <xsd:restriction base="xsd:string">
         <xsd:enumeration value="Insert"/>
         <xsd:enumeration value="Update"/>
         <xsd:enumeration value="Delete"/>
      </xsd:restriction>
   </xsd:simpleType>
```

The response contains one or more gss:EntryId elements – one for each entry that is operated upon by the transaction. The content of each gss:EntryId element is the unique identifier of the affected entry. The value of the action attribute shall be set to "Insert", "Update" or "Delete" to indicate the kind of action performed on the entry. The optional featureRef attribute may contain a link to the feature(s) modified by the corresponding action.

In the event that the operation fails an exceptions shall be raised as described in 7.5.

9.3.2.4 Bindings

9.3.2.4.1 OGC bindings

9.3.2.4.1.1 POX with POST binding

9.3.2.4.1.1.1 Request

Sending an XML-encoded Transaction request to a server's base URL using the HTTP POST method shall create, modify or delete one or more feed entries. The base POST URL for the Transaction operation can be determined from the server's capabilities document (see 9.3.1.2).

9.3.2.4.1.1.2 Response

Upon successful execution of a Transaction operation the server shall respond by setting the HTTP status code to 200 and generating an XML document that contains a gss:TransactionResponse element (see 9.3.2.3).

9.3.2.4.1.2 KVP with GET binding

No KPV binding is defined.

9.3.2.4.2 REST binding

9.3.2.4.2.1 Introduction

The REST binding for publishing of feed entries to the Geosynchronization server shall be performed using the ATOM Publishing Protocol (see IETF RFC 5023). The AtomPub protocol uses the following HTTP methods to perform transactions: POST to create new entries (i.e. Insert), PUT to modify existing entries (i.e. Update) and DELETE to remove an entry (i.e. Delete). Unlike the POX binding (see 9.3.2.4.1.1) the REST binding can only operate on one entry at a time.

9.3.2.4.2.2 Creating a new feed entry (POST)

To add entries to a feed, a client shall send an HTTP POST requests to the URL of the feed (see 9.2.5.1). The content of the POST request *shall* be an ATOM entry. See AtomPub sub-clause 9.2 for further details on resource creation.

For resource creations a Geosynchronization service shall only accept the media type of `application/atom+xml;type=entry`. If any other resource type is encountered, the service shall raise an exception (see 7.7).

9.3.2.4.2.3 Modifying an existing feed entry (PUT)

To edit a feed entry, a client shall send a HTTP PUT request to the entries URL (see 9.2.5.2). To avoid unintentional loss of data when editing feed entries, clients should retrieve the existing entry from the Feed make their changes, and the put the entire entry back to the server.

9.3.2.4.2.4 Deleting a feed entry (DELETE)

To delete a feed entry, a client shall send an HTTP DELETE request to the entries URL (see 9.2.5.2).

9.3.2.4.2.5 Response encoding

The ATOM Publishing Protocol uses the response status codes defined in HTTP to indicate the success or failure of an operation. Consult the HTTP specification (see RFC2616) for detailed definitions of each status code and the AtomPub specification (see RFC 5023) for the content of each response. This information is summarized here.

In response to a POST, the server shall signals a successful entry creation with an HTTP status code of 201. The response shall include a Location header indicating the entry URI of the Atom Entry, and an ATOM representation of the entry shall be returned in the body of the response. The returned entry may not match the entry posted by the client. A server may change the values of various elements in the entry, such as the atom:id, atom:link, atom:updated, and atom:author values, and may choose to remove or add other elements and attributes, or change element content and attribute values.

In response to a PUT, the server shall signal a successful update with an HTTP status code of 200. An ATOM representation of the modified entry shall be returned in the body of response.

In response to a DELETE, the server shall signal a successful deletion with an HTTP status code of 200.

In the event that an exception is encountered, a Geosynchronization Service shall respond with an HTTP status code and an exception report as specified in sub-clause 7.7.

9.3.2.4.3 SOAP binding

Sending a SOAP message, whose body contains an XML-encoded Transaction request (see 9.3.2.2.1) , to a server's base URL using the HTTP POST method shall generate a SOAP response message whose body contains a gss:TransactionResponse element (see 9.3.2.3).

9.3.3 Query

9.3.3.1 Introduction

This clause defines a generalized query operation, called GetEntries, that allows feeds to be queried for entries that satisfy some set of spatial and non-spatial predicates where the spatial predicates can include geometric and temporal constraints. The response to this query operation shall itself be encoded as an ATOM feed but only containing the entries that satisfy the predicates.

The operation's parameters are intentionally designed to map the standard OpenSearch parameters as well as the parameters found in the geo and time extensions (see OGC 10-032).

9.3.3.2 Request encoding

9.3.3.2.1 XML encoding

The following XML Schema fragment defines the XML encoding for the GetEntries operation:

```
<xsd:element name="GetEntries">
   <xsd:complexType>
      <xsd:complexContent>
         <xsd:extension base="gss:BaseRequestType">
            <xsd:sequence>
               <xsd:element name="SearchTerms" type="xsd:string"
```

```
                         minOccurs="0" maxOccurs="unbounded"/>
                  <xsd:element ref="fes:Filter" minOccurs="0"/>
               </xsd:sequence>
               <xsd:attribute name="feed" type="xsd:anyURI"/>
               <xsd:attribute name="outputFormat" type="xsd:string"
                              use="optional"
                              default="application/atom+xml; type=entry"/>
               <xsd:attribute name="startPosition"
                              type="xsd:nonNegativeInteger"
                              use="optional" default="1"/>
               <xsd:attribute name="maxEntries" type="xsd:nonNegativeInteger"
                              use="optional" default="25"/>
            </xsd:extension>
         </xsd:complexContent>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="EntryId"
                type="gss:EntryIdType"
                substitutionGroup="fes:_Id"/>
   <xsd:complexType name="EntryIdType">
      <xsd:complexContent>
         <xsd:extension base="fes:AbstractIdType">
            <xsd:attribute name="eid" type="xsd:anyURI" use="required"/>
         </xsd:extension>
      </xsd:complexContent>
   </xsd:complexType>
```

The gss:SearchTerms element may be used to encode one or more search terms that the server should look for in each entry.

The ogc:Filter elements may be used to encode a more precise set of predicates that is applied by the query.

The implied logical operation when both the gss:SearchTerms and ogc:Filter elements are specified is AND.

The feed attribute shall be used to specify the name of the feed to query.

The outputFormat attribute may be used to assert the format of the response. The default value shall be "application/atom+xml; type=feed" indicating that the output shall be an ATOM feed. A server may implement other formats but this candidate standard does not assign any meaning to those formats. If additional formats are support they shall be advertised in the server's capabilities document using the ogc:Parameter element (see OGC 06-121r3, 7.4.6) in the operation metadata section.

The startPosition attribute may be used to indicate the start index at which the server shall begin presenting results.

The maxEntries attribute may be used to specify how many entries shall appear in the response.

The gss:EntryId element may be used in a Filter expression to identify specific entries in a GetEntries request.

9.3.3.2.2 KVP encoding

The following table defines the KVP-encoding for the GetEntries operation.

Table 11 – GetEntries KVP request encoding

Parameter Name	O / M	Type	Description
SERVICE	M	string	Fixed string GSS.
VERSION	M	string	Fixed string 1.0.0
REQUEST	M	string	Fixed string GetEntries
FEED	M	URI	Feed id to query.
OUTPUTFORMAT	O	string	Servers shall support the string applicaton/atom+xml indicating that the output shall be an ATOM feed. Other values are allowed but are not assigned any meaning by this candidate standard. The default value is application/atom+xml.
STARTPOSITION	O	integer	Positive integer indicating start index within result set. The default value is 1.
MAXENTRIES	O	integer	Indicates the maximum number of entries to present in the response. The default value is 25.
SEARCHTERMS	O	string	Comma separated list of terms to search for within each entry.
Spatial parameters:			
BBOX	O	string	Bounding box predicate encoded as specified in 06-121r3.
GEOM	O	string	WKT-encoded geometry specifying an area of interest.
SPATIALOP	O	string	One of: Equals, Disjoint, Touches, Within, Overlaps, Crosses, Intersects (d), Contains
CRS	O	anyURI	A URI asserting the CRS of the specified GEOM value. The default value shall be urn:ogc:def:crs:EPSG::4326.
Temporal parameters:			
STARTTIME	O	string	An RFC 3339 string encoding a start time.
ENDTIME	O	string	An RFC 3339 string encoding an end time
TEMPORALOP	O	string	One of: After (d), Before, Begins, BegunBy, TContains, During, EndedBy, Ends, TEquals, Meets, MetBy, TOverlaps, OverlappedBy
Identity predicate			
ENTRYID	O	string	Identifier of an entry to retrieve
Generalized predicate:			
FILTER	O	string	An XML fragment encoding a OGC filter that is applied to the entries.

NOTES:

1) The Spatial parametersjjjj and the FILTER parameter are mutually exclusive

2) The implied logical operation for the search parameters is AND.

3) Geometry values for the encoded using the WKT formats (see ANNEX B).

4) A single time instant shall be encoded using the STARTTIME parameter.

9.3.3.2.2.1 OpenSearch parameter mapping

The following table maps the GetEntries parameters to the standard set of OpenSearch parameters as well as the geo and time extensions.

Table 12 – Mapping of GetEntries parameters to OpenSearch

Parameter Name	OpenSearch parameter
SERVICE	Fixed in URL template to 'SERVICE=GSS'
VERSION	Fixed in URL template to 'VERSION=1.0.0'
REQUEST	Fixed in URL template to 'REQUEST=GetEntries'
FEED	Fixed in URL template to 'FEED=<feed id>' The value is the name of a base feed (i.e. CHANGEFEED, RESOLUTIONFEED or REPLICATIONFEED) or the name of a topic available on the server. The list of available topics can be obtained using the ListTopics operation.
OUTPUTFORMAT	Fixed in URL template to 'OUTPUTFORMAT=ATOM'
STARTPOSITION	startIndex
MAXENTRIES	Count
SEARCHTERMS	searchTerms
Spatial Parameters:	
BBOX	geo:box
GEOM	geo:geometry
SPATIALOP	geo:relation
CRS	Fixed in URL template to CRS=urn:ogc:def:crs:EPSG::4326.
Temporal parameters:	
STARTTIME	time:start
ENDTIME	time:end
TEMPORALOP	Fixed in url template to 'TEMPORALOP=X' where X is one of After, Before, Begins, BegunBy, TContains, During, EndedBy, Ends, TEquals, Meets, MetBy, TOverlaps, OverlappedBy
General predicate:	
FILTER	Fixed in URL template to 'FILTER=<xml fragment>'

9.3.3.2.2.2 Invoking GetEntries using a standard OpenSearch client

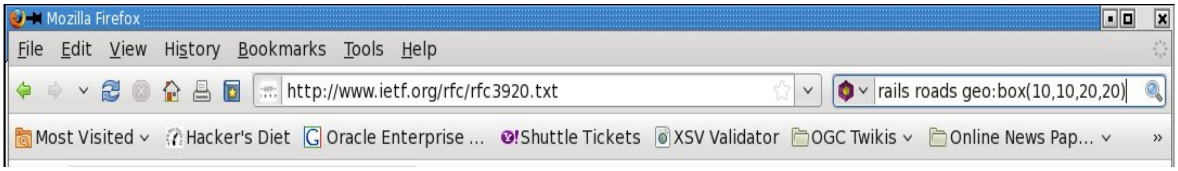
Many OpenSearch clients are not aware of the geo and time extensions of OpenSearch and they would definitely not be aware of any of the extensions proposed in this document. This presents a usage problem since the power of the GetEntries operation would be limited to specialized clients rather than standard OpenSearch clients such as Firefox or Internet Explorer.

Over time, standard OpenSearch clients will implement the geo and time extensions. However, until that time this candidate standard proposes overloading the searchTerms parameter to handle the geo and time extensions. The proposed syntax for embedding geo and time parameter within the searchTerms parameter is:

<namespace prefix>:<parameter>"(" value ")"

Example:

Consider a query that searches for entries containing the terms "rails" and "roads" and also includes the spatial predicate "bbox=10,10,20,20". The following figure illustrates how such a query would be embedded into the search box of a browser.



http://www.ietf.org/rfc/rfc3920.txt

rails roads geo:box(10,10,20,20)

Firefox would pass to the server the search terms 'rails', 'roads' and the string 'geo:box(10,10,20,20)'. It is then the responsibility of the Geosynchronization server to recognize the geo:box parameter and apply the spatial predicate to the query.

9.3.3.2.2.3 Description document

The following XML fragment illustrated how an OpenSearch URL template is created using the GetEntries operation:

```
<Url type="application/atom+xml"
     template="http://www.pvretano.com/gss/cubeserv.cgi?
                service=GSS&
                version=1.0.0&
                request=GetEntries&
                feed=CHANGEFEED&
                outputFormat=ATOM&
                startPosition={startIndex?}&
                maxEntries={count?}&
```

```
searchTerms={searchTerms?}&
bbox={geo:box?}&
starttime={time:start?}&
temporalOp=After"/>
```

GetEntries parameters, such as service and version that do not map to OpenSearch parameters are hard coded into the template. GetEntries parameters, such as bbox and starttime, that map to OpenSearch parameters are encoded using the substitution variable syntax described in the OpenSearch standard (see OpenSearch).

9.3.3.3 Response encoding

The response to a GetEntries operation, using the default output format, shall be an ATOM feed (see IETF RFC 4287) with the mime type of application/atom+xml. The feed shall contain all entries that satisfy the specified query predicates and presentation parameters. The entries in the response feed shall be sorted in descending temporal order so that the most recent entry is the first entry in the response.

9.3.3.4 Bindings

9.3.3.4.1 OGC bindings

9.3.3.4.1.1 POX with POST binding

9.3.3.4.1.1.1 Request

A query can be executed by sending an XML-encoded GetEntries request to a server's base URL using the HTTP POST method. The base POST URL for the GetEntries operation can be determined from the server's capabilities document (see 9.3.1.2).

9.3.3.4.1.1.2 Response

Upon successful execution of a GetEntries operation the server shall respond by setting the HTTP status code to 200 and generating an XML document that contains the entries that satisfy the query and conforms to the feed encoding defined in IETF RFC 4287.

9.3.3.4.1.2 KVP with GET binding

9.3.3.4.1.2.1 Request

A query can be executed by sending an HTTP GET request to a URL formed by appending (see OGC 06-121r3, 11.2) the GetEntries request parameters (see Table 11) to a server's base URL. The base GET URL for the GetEntries operation can be determined from the server's capabilities document (see 9.3.1.2).

9.3.3.4.1.2.2 Response

See sub-clause 9.3.3.4.1.1.2.

9.3.3.4.2 REST binding

9.3.3.4.2.1.1 Request

A query can be executed by sending an HTTP GET request to a URL formed by appending the GetEntries query parameters to the URL of the feed (see 9.2.5.1).

9.3.3.4.2.1.2 Response

See sub-clause 9.3.3.4.1.1.2.

9.3.3.4.3 SOAP binding

Sending a SOAP message, whose body contains an XML-encoded GetEntries request (see 9.3.3.2.1) , to a server's base URL using the HTTP POST method shall generate a SOAP response message whose body contains an ATOM feed (see IETF RFC 4287).

9.3.4 Change management

9.3.4.1 Introduction

The following set of operations allows changes to be reviewed and either accepted or rejected.

9.3.4.2 AcceptChange operation

9.3.4.2.1 Introduction

The AcceptChange operation shall signal to a GSS that the specified change has been reviewed and is accepted.

Upon receiving an AcceptChange request the server shall:

1. Apply the specified change to the target WFS.

2. If the change is successfully applied, the server shall:

   a) Create an entry is the Resolution feed indicating that the specified change has been accepted. The entry shall be encoded according to the content rules specified in Table 9.

   b) Create one or more entries in the replication feed each describing the change to a feature that was affected the change proposal. Each entry shall be encoded according to the content rules specified in Table 10.

   c) The server shall send a response to the client as specified in 8.3.5.2.2.

3. If and error is encountered applying the change to the target WFS, the server shall raise an exception as specified in 7.5.

9.3.4.2.2 Request encoding

9.3.4.2.2.1 XML encoding

The following XML Schema fragment defines the XML encoding for the AcceptChange request:

```
<xsd:element name="AcceptChange">
   <xsd:complexType>
      <xsd:complexContent>
         <xsd:extension base="gss:BaseRequestType">
            <xsd:sequence>
               <xsd:element name="EntryId" type="xsd:anyURI"/>
                <xsd:element name="Comment"
                             type="ows:LanguageStringType"
                             minOccurs="0" maxOccurs="unbounded"/>
                <xsd:element name="Revision" minOccurs="0">
                   <xsd:complexType>
                      <xsd:sequence>
                         <xsd:any namespace="##other"
                                  processContents="lax"/>
                      </xsd:sequence>
                   </xsd:complexType>
                </xsd:element>
            </xsd:sequence>
         </xsd:extension>
      </xsd:complexContent>
   </xsd:complexType>
</xsd:element>
```

The gss:EntryId element shall encode the id of the entry from the CHANGEFEED that is being accepted.

The gss:Comment element may be used to include an acceptance comment. Multiple gss:Comments elements may be used to express the same comment in languages.

If a proposed change is accepted with modification the AcceptChange operation shall include a gss:Revision element that shall contain the originally proposed WFS transaction with the modifications made by the reviewer or review process. It is also recommended that a gss:Comment element be included in the AcceptChange operation explaining the modifications made by the reviewer. When processing the AcceptChange operation the server shall apply to revised WFS transaction included with the operation rather than the WFS transaction found in the CHANGEFEED entry.

9.3.4.2.2.2 KVP encoding

The following tables defines the KVP-encoding for the AcceptChange operation

Table 13 – AcceptChange KVP-encoding

Parameter Name	Description	Mandatory ?
REQUEST	Fixed value of AcceptChange	Y
ENTRYID	The ID of an entry in the change feed.	Y
COMMENT	A human readable comment about the accepted change.	N

The KVP-encoding does not support accepting changes with revisions.

9.3.4.2.3 Response

The following XML-Schema fragment defines the response to an AcceptChange operation:

```
<xsd:element name="AcceptChangeResponse" type="gss:EntryResponseType"/>
```

See sub-clause 7.6.4 for the declaration of gss:EntryResponseType.

The server shall encode the entry id of the entry id of the accepted change using the EntryId element in the response.

The server may also include a comment about the operation using the optional gss:Description element.

In the event that an exception is encountered, the service shall respond with an exception report as specified in sub-clause 7.7.

9.3.4.2.4 Bindings

9.3.4.2.4.1 OGC bindings

9.3.4.2.4.1.1 POX with POST binding

9.3.4.2.4.1.1.1 Request

Sending an XML-encoded AcceptChange request to a server's base URL using the HTTP POST method shall signal that the specified CHANGEFEED entry has been accepted. The base POST URL for the AcceptChange operation can be determined from the server's capabilities document (see 9.3.1.2).

9.3.4.2.4.1.1.2 Response

Upon successful execution of an AcceptChange operation the server shall respond by setting the HTTP status code to 200 and generating an XML document that contains a gss:AcceptChangeResponse element (see 9.3.4.2.3).

9.3.4.2.4.1.2 KVP with GET binding

9.3.4.2.4.1.2.1 Request

Sending an HTTP GET request to a URL formed by appending (see OGC 06-121r3, 11.2) the AcceptChange request parameters (see Table 13) to a server's base URL shall signal that the specified CHANGEFEED entry has been accepted. The base GET URL for the AcceptChange operation can be determined from the server's capabilities document (see 9.3.1.2).

9.3.4.2.4.1.2.2 Response

See sub-clause 9.3.4.2.4.1.1.2.

9.3.4.2.4.2 REST binding

9.3.4.2.4.2.1 Request

Sending an HTTP PUT request, containing an XML-encoded AcceptChange message, to the CHANGEFEED entry URL (see 9.2.5.2) shall signal that the indicated change has been accepted.

The entry id in the AcceptChange message shall match the id of the CHANGEFEED entry being modified with the HTTP PUT method. The target CHANGEFEED entry shall not be modified by the operation but the server shall perform the operations described in sub-clause 9.3.4.2.1 for accepting a change.

9.3.4.2.4.2.2 Response

Upon successful execution of an AcceptChange operation the server shall respond by setting the HTTP status code to 200 and generating an XML document that contains a gss:AcceptChangeResponse element (see 9.3.4.2.3).

9.3.4.2.4.3 SOAP binding

Sending a SOAP message, whose body contains an XML-encoded AcceptChange request (see 9.3.4.2.2.1) , to a server's base URL using the HTTP POST method shall generate a SOAP response message whose body contains a gss:AcceptChangeResponse element (see 9.3.4.2.3).

9.3.4.3 RejectChange operation

9.3.4.3.1 Introduction

The RejectChange operation shall be used to signal that the specified entry in the CHANGEFEED has been reviewed and has been rejected.

Upon receiving a RejectChange request the server shall create an entry in the Resolution feed indicating that the change was rejected. The entry shall be encoded according to the encoding rules in Table 9.

9.3.4.3.2 Request encoding

9.3.4.3.2.1 XML encoding

The following XML Schema fragment defines the XML encoding to the RejectChange operation.

```
<xsd:element name="RejectChange">
   <xsd:complexType>
      <xsd:complexContent>
         <xsd:extension base="gss:BaseRequestType">
            <xsd:sequence>
               <xsd:element name="EntryId" type="xsd:anyURI"/>
                <xsd:element name="Comment"
                             type="ows:LanguageStringType"
                             minOccurs="0" maxOccurs="unbounded"/>
            </xsd:sequence>
         </xsd:extension>
      </xsd:complexContent>
   </xsd:complexType>
</xsd:element>
```

The gss:EntryId element shall encoding the identifier of the CHANGEFEED entry that has been rejected. Multiple gss:Comments elements may be used to express the same comment in languages.

9.3.4.3.2.2 KVP encoding

The following table defines the KVP-encoding for the RejectChange operation.

Table 14 – RejectChange KVP request encoding

Parameter Name	Description	Value	Mandatory ?
REQUEST	Name of operation	Fixed value of *RejectChange*	Y
ENTRYID	The ID of an entry in the change feed.	Obtained by querying the change feed.	Y
COMMENT	A human readable narrative describing why the change was rejected.	String	Y

9.3.4.3.3 Response

The following XML-schema fragment defines the response to a RejectChange operation:

```
<xsd:element name="RejectChangeResponse" type="gss:EntryResponseType"/>
```

See sub-clause 7.6.4 for the declaration of gss:EntryResponseType.

The server shall encode the entry id of the rejected change using the EntryId element in the response.

The server may also include a comment about the operation using the optional gss:Description element.

In the event that an exception is encountered, the service shall respond with an exception report as specified in sub-clause 7.7.

9.3.4.3.4 Bindings

9.3.4.3.4.1 OGC bindings

9.3.4.3.4.1.1 POX with POST binding

9.3.4.3.4.1.1.1 Request

Sending an XML-encoded RejectChange request to a server's base URL using the HTTP POST method shall signal that the specified CHANGEFEED entry has been rejected. The base POST URL for the RejectChange operation can be determined from the server's capabilities document (see 9.3.1.2).

9.3.4.3.4.1.1.2 Response

Upon successful execution of an RejectChange operation the server shall respond by setting the HTTP status code to 200 and generating an XML document that contains a gss:RejectChangeResponse element (see 9.3.4.3.3).

9.3.4.3.4.1.2 KVP with GET binding

9.3.4.3.4.1.2.1 Request

Sending an HTTP GET request to a URL formed by appending (see OGC 06-121r3, 11.2) the RejectChange request parameters (see Table 14) to a server's base URL shall signal that the specified CHANGEFEED entry has been accepted. The base GET URL for the AcceptChange operation can be determined from the server's capabilities document (see 9.3.1.2).

9.3.4.3.4.1.2.2 Response

See sub-clause 9.3.4.3.4.1.1.2.

9.3.4.3.4.2 REST binding

9.3.4.3.4.2.1 Request

Sending an HTTP PUT request, containing an XML-encoded RejectChange message, to the CHANGEFEED entry URL (see 9.2.5.2) shall signal that the indicated change has been accepted.

The entry id in the RejectChange message shall match the id of the CHANGEFEED entry being modified with the HTTP PUT method. The target CHANGEFEED entry shall not be modified by the operation but the server shall perform the operations described in sub-clause 9.3.4.3.1 for rejecting a change.

9.3.4.3.4.2.2 Response

See sub-clause 9.3.4.3.4.1.1.2.

9.3.4.3.4.3 SOAP binding

Sending a SOAP message, whose body contains an XML-encoded RejectChange request (see 9.3.4.3.2.1) , to a server's base URL using the HTTP POST method shall generate a SOAP response message whose body contains a gss:RejectChangeResponse element (see 9.3.4.3.3).

9.3.4.4 ReviewedChanges operation

9.3.4.4.1 Introduction

The ReviewedChanges operation allows a single request to be sent to a GSS that contains one or more AcceptChange and/or RejectChange operations. It allows for bulk processing of change proposals.

9.3.4.4.2 Request encoding

9.3.4.4.2.1 XML encoding

The following XML Schema fragment defines the XML encoding to the ReviewedChanges operation.

```
<xsd:element name="ReviewedChanges">
   <xsd:complexType>
      <xsd:complexContent>
         <xsd:extension base="gss:BaseRequestType">
            <xsd:choice maxOccurs="unbounded">
               <xsd:element ref="gss:AcceptChange"/>
               <xsd:element ref="gss:RejectChange"/>
            </xsd:choice>
         </xsd:extension>
      </xsd:complexContent>
   </xsd:complexType>
</xsd:element>
```

9.3.4.4.2.2 KVP encoding

No KVP encoding is defined for the ReviewedChanges operation.

9.3.4.4.3 Response encoding

The following XML-Schema fragment defines the response to a ReviewedChanges operation:

```
<xsd:element name="ReviewedChangesResponse">
   <xsd:complexType>
      <xsd:choice maxOccurs="unbounded">
         <xsd:element ref="gss:AcceptChangeResponse"/>
         <xsd:element ref="gss:RejectChangeResponse"/>
         <xsd:element ref="ows:ExceptionReport"/>
      </xsd:choice>
   </xsd:complexType>
</xsd:element>
```

The elements gss:AcceptChangeResponse and gss:RejectChangeResponse shall appear in the response in the same order in which the gss:AcceptChange and gss:RejectChange elements appear in the ReviewedChanges request.

In the event that one of the AcceptChange or RejectChange operations fails, the ows:ExpcetionReport element (see Clause 8, 06-121r3) shall appear in the response in place of the gss:AcceptChangeResponse or gss:RejectChangeResponse element that corresponds to the failed operation.

9.3.4.4.4 Bindings

9.3.4.4.4.1 OGC bindings

9.3.4.4.4.1.1 POX with POST binding

Sending an XML-encoded ReviewedChanges request to a server's base URL using the HTTP POST method shall signal that the specified AcceptChange and RejectChange operations shall be processed in bulk.

The base POST URL for the ReviewedChanges operation can be determined from the server's capabilities document (see 9.3.1.2).

9.3.4.4.4.1.2 KVP with GET binding

No KVP binding is defined for the ReviewedChanges operation.

9.3.4.4.4.2 REST binding

No REST binding is defined for the ReviewedChange operation.

9.3.4.4.4.3 SOAP binding

Sending a SOAP message, whose body contains an XML-encoded ReviewedChanges request (see 9.3.4.4.2.1) , to a server's base URL using the HTTP POST method shall generate a SOAP response message whose body contains a gss:ReviewedChangesResponse element (see 9.3.4.4.3).

9.3.5 Subscription

9.3.5.1 Introduction

The subscription API defines operations to create, and remove topics as well as operations so that clients may discover the list of available topics.

The subscription API also defines operations that allow interested parties to subscribe to a topic or base feed, get a list of user subscriptions and pause, resume or cancel a subscription.

9.3.5.2 CreateTopic operation

9.3.5.2.1 Introduction

The CreateTopic operation allows a client to create a filtered view of a base feed by optionally specifying spatial and/or non-spatial constraints to be applied to an Atom+GeoRSS feed. The new topic shall include a description containing a human-readable title and may also include a human-readable narrative describing the topic.

9.3.5.2.2 Request encoding

9.3.5.2.2.1 XML encoding

The following XML schema fragment defines the XML encoding for the CreateTopic operation:

```
<xsd:element name="CreateTopic">
<xsd:complexType>
      <xsd:complexContent>
         <xsd:extension base="gss:BaseRequestType">
            <xsd:sequence>
               <xsd:element ref="gss:Topic" />
            </xsd:sequence>
         </xsd:extension>
      </xsd:complexContent>
   </xsd:complexType>
</xsd:element>
```

The CreateTopic operation contains a gss:Topic element that defines the topic to create.

The gss:Topic element (see 9.2.6.2) contains the following components:

1. The gss:Feed element shall contain the name of the feed URL upon which the topic will be created (see 9.2.5.1).

2. The gss:Description element shall include an ows:Title element, containing a human readable title for the topic, and may include an ows:Abstract and ows:Keyword elements.

3. The fes:Filter element shall encode predicates to be applied to the feed to identify the set of entries for the topic.

9.3.5.2.2.2 KVP encoding

Table 15 defines the KVP parameters for the CreateTopic operation.

Table 15 – CreateTopic KVP request encoding

Parameter Name	O/M	Type	Description
FEED	M	URI	URL of the base feed (see 9.2.5.1)
TITLE	O	string	Human readable title for the subscription
ABSTRACT	O	string	Human readable narrative describing the subscription.
KEYWORDS	O	string	A comma-separated list of key words.
Spatial parameters:			
BBOX	O	string	Bounding box predicate encoded as specified in 06-121r3.
GEOM	O	string	WKT-encoded geometry specifying an area of interest.
SPATIALOP	O	string	One of: Equals, Disjoint, Touches, Within, Overlaps, Crosses, Intersects (d), Contains
CRS	O	anyURI	A URI asserting the CRS of the specified GEOM value. The default value shall be urn:ogc:def:crs:EPSG::4326.
Temporal parameters:			
STARTTIME	O	string	An RFC 3339 string encoding a start time.
ENDTIME	O	string	An RFC 3339 string encoding an end time
TEMPORALOP	O	string	One of: After (d), Before, Begins, BegunBy, TContains, During, EndedBy, Ends, TEquals, Meets, MetBy, TOverlaps, OverlappedBy
Identity predicate			
ENTRYID	O	string	Identifier of an entry to retrieve

Parameter Name	O/M	Type	Description
Generalized predicate:			
FILTER	O	string	An XML fragment encoding a OGC filter that is applied to the entries.
NOTES: 1) EntryId, Spatial parameters and FILTER are mutually exclusive 2) EntryId and Temporal parameters are mutually exclusive 3) The implied logical operation for the search parameters is AND. 4) Geometry values for the encoded using the WKT formats (see ANNEX B). 5) A single time instant shall be encoded using the STARTTIME parameter.			

9.3.5.2.3 Response encoding

The following XML-Schema fragment defines the response to a CreateTopic operation:

```
<xsd:element name="CreateTopicResponse" type="gss:TopicResponseType"/>
```

See sub-clause 7.6.3 for the declaration of gss:TopicResponseType.

The server shall generate an id for the newly created topic and shall encode it as the content of the TopicId element in the response. The topic id may be used in subsequent operations to reference the topic.

The server may also include a comment about the topic using the gss:ResponseComment element.

In the event that an exception is encountered, the service shall respond with an exception report as specified in sub-clause 7.7.

9.3.5.2.4 Bindings

9.3.5.2.4.1 OGC bindings

9.3.5.2.4.1.1 POX with POST binding

9.3.5.2.4.1.1.1 Request

Sending an XML-encoded CreateTopic request to a server's base URL using the HTTP POST method shall create a new topic. The base POST URL for the CreateTopic operation can be determined from the server's capabilities document (see 9.3.1.2).

9.3.5.2.4.1.1.2 Response

Upon successful execution of an CreateTopic operation the server shall respond by setting the HTTP status code to 201 and generating an XML document that contains a gss:CreateTopicResponse element (see 9.3.5.2.3).

9.3.5.2.4.1.2 KVP with GET binding

9.3.5.2.4.1.2.1 Request

Sending a KVP-encoded CreateTopic request to a server's base URL using the HTTP GET method shall create a new topic. The base GET URL for the CreateTopic operation can be determined from the server's capabilities document (see 9.3.1.2).

9.3.5.2.4.1.2.2 Response

See sub-clause 9.3.5.2.4.1.1.2.

9.3.5.2.4.2 REST binding

9.3.5.2.4.2.1 Request

Sending an HTTP POST request, containing a gss:Topic element, to a server's topic URL shall create a new topic. See sub-clause 9.2.5.3 for a description of the server's topic URL.

The following XML schema fragment defines the gss:Topic element:

```
   <xsd:element name="Topic" type="gss:TopicType"/>
```

The gss:TopicType type is defined is sub-clause 9.2.6.2.

Example:

The following example creates a tropic that monitors the change feed for entries that lie within the region of Gatineau, Quebec.

```
POST /gss.cgi/topic
Host: www.acme.com
User-Agent: geosyncclient/1.0
Authorization: Basic ZGFmZnk6c2VjZXJldA==
Content-Type: text/xml
Content-Length: nnn
Slug: fid435890453

<?xml version="1.0" encoding="UTF-8"?>
<gss:Topic feed="CHANGEFEED">
   <gss:Description>
      <ows:Title>Gatineau Changes</ows:Title>
      <ows:Abstract>This topic lists all the changes made in the Gatineau
region in Quebec, Canada</ows:Abstract>
   </gss:Description>
   <fes:Filter>
      <fes:Intersects>
         <fes:PopertyName>entry/georss:where</fes:PropertyName>
         <gml:Polygon> … <gml:Polygon>
      </fes:Intersects>
   </fes:Filter>
</gss:Topic>
```

9.3.5.2.4.2.2 Response

See sub-clause 9.3.5.2.4.1.1.2.

9.3.5.2.4.3 SOAP binding

Sending a SOAP message, whose body contains an XML-encoded CreateTopic request (see 9.3.5.2.2.1) , to a server's base URL using the HTTP POST method shall generate a SOAP response message whose body contains a gss:CreateTopicResponse element (see 9.3.5.2.3).

9.3.5.3 RemoveTopic operation

9.3.5.3.1 Introduction

The RemoveTopic operation allows a client to delete a topic previously created using the CreateTopic operation. The list of available topics can be discovered using the ListTopics operation. The topic to be removed is identified in the request using its id.

9.3.5.3.2 Request encoding

9.3.5.3.2.1 XML encoding

The following XML-Schema fragment defines the XML encoding for the RemoveTopic operation.

```
<xsd:element name="RemoveTopic">
   <xsd:complexType>
      <xsd:complexContent>
         <xsd:extension base="gss:BaseRequestType">
            <xsd:attribute name="topicId" type="xsd:anyURI" use="required"/>
         </xsd:extension>
      </xsd:complexContent>
   </xsd:complexType>
</xsd:element>
```

9.3.5.3.2.2 KVP encoding

Table 16 defines the parameters for the **RemoveTopic** operation.

Table 16 – RemoveTopic KVP request encoding

Parameter Name	Description	Value	Mandatory?
REQUEST	Name of operation	Fixed value of *RemoveTopic*	Y
TOPICID	The ID of an existing subscription	Possible values obtained using the ListTopics operation.	Y

9.3.5.3.3 Response encoding

The following XML-schema fragment defines the response to a RemoveTopic operation:

```
<xsd:element name="RemoveTopicResponse" type="gss:TopicResponseType"/>
```

See sub-clause 7.6.3 the declaration of gss:TopicResponseType.

The server shall encode the topic id of the removed topic using the TopicId element in the response.

The server may also include a comment about the operation using the gss:ResponseComment element.

In the event that an exception is encountered, the service shall respond with an exception report as specified in sub-clause 7.7.

9.3.5.3.4 Bindings

9.3.5.3.4.1 OGC bindings

9.3.5.3.4.1.1 POX with POST binding

9.3.5.3.4.1.1.1 Request

Sending an XML-encoded RemoveTopic request to a server's base URL using the HTTP POST method shall remove the topic. The base POST URL for the RemoveTopic operation can be determined from the server's capabilities document (see 9.3.1.2).

9.3.5.3.4.1.1.2 Response

Upon successful execution of a RemoveTopic operation the server shall respond by setting the HTTP status code to 200 and generating an XML document that contains a gss:RemoveTopicResponse element (see 9.3.5.3.3).

9.3.5.3.4.1.2 KVP with GET binding

9.3.5.3.4.1.2.1 Request

Sending a KVP-encoded RemoveTopic request to a server's base URL using the HTTP GET method shall remove the topic. The base GET URL for the RemoveTopic operation can be determined from the server's capabilities document (see 9.3.1.2).

9.3.5.3.4.1.2.2 Response

See sub-clause 9.3.5.3.4.1.1.2.

9.3.5.3.4.2 REST binding

9.3.5.3.4.2.1 Request

Sending an HTTP DELETE request to a topic's URL shall remove the topic. The ListTopics operation can be used to determine the URL of each topic.

Example:

The following example removes a tropic.

```
POST /gss.cgi/topic/urn:uuid: 177b04aa-80bb-45fd-b54f-454a31bc8913
Host: www.acme.com
Authorization: Basic ZGFmZnk6c2VjZXJldA==
```

9.3.5.3.4.2.2 Response

See sub-clause 9.3.5.3.4.1.1.2.

9.3.5.3.4.3 SOAP binding

Sending a SOAP message, whose body contains an XML-encoded RemoveTopic request (see 9.3.5.3.2.1) , to a server's base URL using the HTTP POST method shall generate a SOAP response message whose body contains a gss:RemoveTopicResponse element (see 9.3.5.3.3).

9.3.5.3.5 Termination notification

A described in the abstract model (see 6.5.6, 6.5.7), clients may subscribe to topics and be notified of any posted entries. In the event that a topic is removed that still has subscribers, a termination notification shall be sent to each subscriber. Thus subscribers should be prepared to deal with such an eventuality.

9.3.5.4 ListTopics operation

9.3.5.4.1 Introduction

The ListTopics operation allows a client to discover the available topics to which the client may subscribe using the Subscribe operation.

9.3.5.4.2 Request encoding

9.3.5.4.2.1 XML encoding

The following XML schema fragment defines the XML encoding for the ListTopics operation:

```
<xsd:element name="ListTopics">
   <xsd:complexType>
      <xsd:complexContent>
         <xsd:extension base="gss:BaseRequestType">
            <xsd:element name="TopicId" type="xsd:anyURI"
                         minOccurs="0" maxOccurs="unbounded"/>
         </xsd:extension>
      </xsd:complexContent>
   </xsd:complexType>
</xsd:element>
```

One or more gss:TopicId elements may be specified to get a listing of specific entries.

If no gss:TopicId element is specified then all topics shall be listed.

9.3.5.4.2.2 KVP encoding

Table 17 defines the parameters for the **ListTopics** operation.

Table 17 – ListTopics KVP request encoding

Parameter Name	Description	Value	Mandatory?
REQUEST	Name of operation	Fixed value of *ListTopics*	Y

Parameter Name	Description	Value	Mandatory?
TOPICID	Comma separated list of topic ids to list.		N

9.3.5.4.3 Response

The following XML-Schema fragment defines the response to a ListTopics operation:

```
<xsd:element name="ListTopicsResponse">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element name="Topic" minOccurs="0" maxOccurs="unbounded">
            <xsd:complexType>
               <xsd:complexContent>
                  <xsd:extension base="gss:TopicType">
                     <xsd:attribute name="tid"
                                    type="xsd:anyURI" use="required"/>
                     <xsd:attribute name="owner"
                                    type="xsd:string" use="required"/>
                  </xsd:extension>
               </xsd:complexContent>
            </xsd:complexType>
         </xsd:element>
      </xsd:sequence>
      <xsd:attributeGroup ref="gss:BaseAttributes"/>
   </xsd:complexType>
</xsd:element>
```

The response consists if a list of gss:Topic elements each describing a topic that is available on the server. The attribute tid shall encode the identifier for the topic. The owner attribute shall encode the name of the user who created the topic.

In the event that an exception is encountered, the service shall respond with an exception report as specified in sub-clause 7.7.

9.3.5.4.4 Bindings

9.3.5.4.4.1 OGC bindings

9.3.5.4.4.1.1 POX with POST binding

9.3.5.4.4.1.1.1 Request

Sending an XML-encoded ListTopics request to a server's base URL using the HTTP POST method shall list the requested topic(s). The base POST URL for the ListTopics operation can be determined from the server's capabilities document (see 9.3.1.2).

9.3.5.4.4.1.1.2 Response

Upon successful execution of a ListTopics operation the server shall respond by setting the HTTP status code to 200 and generating an XML document that contains a gss:ListTopicsResponse element (see 9.3.5.4.3).

9.3.5.4.4.1.2 KVP with GET binding

9.3.5.4.4.1.2.1 Request

Sending an HTTP GET request to a URL formed by appending (see OGC 06-121r3, 11.2) the ListTopics request parameters (see Table 17) to a server's base URL shall list all available topics. The base GET URL for the ListTopics operation can be determined from the server's capabilities document (see 9.3.1.2).

9.3.5.4.4.1.2.2 Response

See sub-clause 9.3.5.4.4.1.1.2.

9.3.5.4.4.2 REST binding

9.3.5.4.4.2.1 Request

Sending an HTTP GET request to a server's topic URL shall list all available topics. See sub-clause 9.2.5.3 which describes how to form the server's topic URL.

9.3.5.4.4.2.2 Response

See sub-clause 9.3.5.4.4.1.1.2.

9.3.5.4.4.3 SOAP binding

Sending a SOAP message, whose body contains an XML-encoded ListTopics request (see 9.3.5.4.2.1) , to a server's base URL using the HTTP POST method shall generate a SOAP response message whose body contains a gss:ListTopicsResponse element (see 9.3.5.4.3).

9.3.5.5 Subscribe operation

9.3.5.5.1 Introduction

The Subscribe operation allows a client to subscribe to an available topic. The list of available topics can be discovered using the ListTopics operation (see 9.3.5.4).

9.3.5.5.2 Request encoding

9.3.5.5.2.1 XML encoding

The following XML fragment defines the XML encoding for a Subscribe operation:

```
<xsd:element name="Subscribe">
   <xsd:complexType>
      <xsd:complexContent>
         <xsd:extension base="gss:SubscriptionType">
            <xsd:attributeGroup ref="gss:BaseAttributes"/>
         </xsd:extension>
      </xsd:complexContent>
   </xsd:complexType>
</xsd:element>
<xsd:complexType name="SubscriptionType">
   <xsd:sequence>
      <xsd:element name="TopicId" type="xsd:anyURI"/>
```

```
         <xsd:element name="Delivery" type="gss:DeliveryMethodType"/>
         <xsd:element name="Expiry"
                      type="gss:DateType" minOccurs="0"/>
      </xsd:sequence>
      <xsd:attribute name="status" type="gss:SubscriptionStatusType"
                     use="optional" default="active"/>
   </xsd:complexType>
   <xsd:simpleType name="DeliveryMethodType">
      <xsd:restriction base="xsd:string">
         <xsd:pattern value="mailto:[^@]+@[^@]+\.[^@]+"/>
         <xsd:pattern value="xmpp:[^@]+@[^@]+\.[^@]+"/>
         <xsd:pattern value="sync:http://(.)*"/>
      </xsd:restriction>
   </xsd:simpleType>
   <xsd:simpleType name="SubscriptionStatusType">
      <xsd:restriction base="xsd:string">
         <xsd:pattern value="active"/>
         <xsd:pattern value="paused"/>
      </xsd:restriction>
   </xsd:simpleType>
```

The gss:TopicId element shall identify the topic to which the request is subscribing. The list of available topics and their identifiers can be obtained using the ListTopics operation (see 9.3.5.4).

The gss:Delivery element shall be a string of the format "*protocol:address[,additional parameters]*" and indicates how notification of events shall be delivered. Servers shall support one or more of the mailto, xmpp or sync protocols. Which specific protocols a server supports shall be advertised in the server capabilities document using an ows:Parameter element within the description of the Subscribe operation. The mailto and xmpp protocols are well defined internet protocols (see IETF RFC 2368 and IETF RFC 3920). The sync protocol is defined in this candidate standard and has the format:

sync:<serverurl>[,<transformation url>]

where the serverurl is the GetCapabilities URL of a WFS that the Geosynchronization service shall keep synchronized based on the specified topic. In the event that source and target WFS schemas are not the same, an optional transformation URL may point to a document (e.g. XSLT document) or service that can transform a transaction as recorded in a Replication Feed entry to a transaction suitable for the target server.

Example:

sync:http://www.pvretano.com/wfs.cgi,http://www.pvretano.com/Convert.xslt

This delivery specification tells the Geosynchronization service to synchronize with the WFS at http://www.pvretano.com/wfs.cgi and use the Convert.xslt file to transform the source transaction from the ReplicationFeed to the schema of the target server.

The optional gss:Expiry element may be used to set an expiry date for the subscription after which not further notifications shall be sent.

The sid and owner attributes shall be ignored if set in a request. These are set by the server once the request has been processed. The sid encodes the subscription id once the subscription has been created and the owner encodes the identity of the user who created the subscription.

The status attribute shall be used to indicate the status of the subscription in response to a ListSubscriptions operation (see 9.3.5.6). It may also be used in the REST binding (see 9.3.5.7.4.2 & 9.3.5.8.4.2) to pause or resume a subscription.

9.3.5.5.2.2 KVP encoding

Table 18 defines the parameters for the **Subscribe** operation.

Table 18 – Subscribe KVP request encoding

Parameter Name	Description	Value	Mandatory ?
REQUEST	Name of operation	String Fixed value of *Subscribe*	Y
TOPICID	The ID of an existing subscription	URI Possible values obtained using the ListTopics operation.	Y
DELIVERY	Method of delivery for notification of new entries.	String With the format *protocol:address[,additional parameters]*. The protocol can be on of mailto, xmpp or sync.	Y
EXPIRY	Sets an expiry date for the subscriptions after which no further notifications shall be sent.	DateTime	N

Example: The following is an example of a minimal **Subscribe** request.

```
http://www.yourserver.com/geosync?REQUEST=Subscribe&TOPICID=urn:uuid:18fed003-
1ff0-480f-93f4-af7ca4b201e2&DELIVERY=mailto:xxx@yyy
```

9.3.5.5.3 Response encoding

The following XML-Schema fragment defines the response to a Subscribe operation:

```
   <xsd:element name="SubscribeResponse" type="gss:SubscriptionResponseType"/>
```

See sub-clause 7.6.2 for the declaration of gss:SubscriptionResponseType.

The server shall encode the subscription id using the SubscriptionId element in the response.

The server may also include a comment about the operation using the gss:Description element..

In the event that an exception is encountered, the service shall respond with an exception report as specified in sub-clause 7.7.

9.3.5.5.4 Bindings

9.3.5.5.4.1 OGC bindings

9.3.5.5.4.1.1 POX with POST binding

9.3.5.5.4.1.1.1 Request

Sending an XML-encoded Subscribe request to a server's base URL using the HTTP POST method shall create a subscription to the specified topic. The base POST URL for the Subscribe operation can be determined from the server's capabilities document (see 9.3.1.2).

9.3.5.5.4.1.1.2 Response

Upon successful execution of a Subscribe operation the server shall respond by setting the HTTP status code to 200 and generating an XML document that contains a gss:SubscribeResponse element (see 9.3.5.2.3).

9.3.5.5.4.1.2 KVP with GET binding

9.3.5.5.4.1.2.1 Request

Sending an HTTP GET request to a URL formed by appending (see OGC 06-121r3, 11.2) the Subscribe request parameters (see Table 18) to a server's base URL shall create a subscription to the specified topic. The base GET URL for the Subscribe operation can be determined from the server's capabilities document (see 9.3.1.2).

9.3.5.5.4.1.2.2 Response

See sub-clause 9.3.5.5.4.1.1.2.

9.3.5.5.4.2 REST binding

9.3.5.5.4.2.1 Request

Sending an HTTP POST request, with a gss:Subscription element as its contents, to the server's subscription URL (see 9.2.5.5) shall create a subscription to the specified topic.

9.3.5.5.4.2.2 Response

See sub-clause 9.3.5.5.4.1.1.2.

9.3.5.5.4.3 SOAP binding

Sending a SOAP message, whose body contains an XML-encoded Subscribe request (see 9.3.5.5.2.1) , to a server's base URL using the HTTP POST method shall generate a SOAP response message whose body contains a gss:SubscribeResponse element (see 9.3.5.5.3).

9.3.5.6 ListSubscriptions

9.3.5.6.1 Introduction

The ListSubscriptions operation generates a list of subscriptions to which an authenticated user has subscribed

9.3.5.6.2 Request encoding

9.3.5.6.2.1 XML encoding

The following XML Schema fragment defines the XML encoding for the ListSubscriptions operation

```
<xsd:element name="ListSubscriptions">
   <xsd:complexType>
      <xsd:complexContent>
         <xsd:extension base="gss:BaseRequestType">
            <xsd:element name="SubscriptionId"
                         type="xsd:anyURI"
                         minOccurs="0" maxOccurs="unbounded"/>
         </xsd:extension>
      </xsd:complexContent>
   </xsd:complexType>
</xsd:element>
```

The gss:SubscriptionId element can be used to specify one or more user subscriptions to present in the response. If the gss:SubscriptionId element is not specified then all user subscriptions shall be listed.

9.3.5.6.2.2 KVP encoding

.Table 19 defines the KVP parameters for the ListSubscriptions operation.

Table 19 – ListSubscriptions KVP request encoding

Parameter Name	Description	Value	Mandatory ?
REQUEST	Name of operation	Fixed value of *ListSubscriptions*	Y
SUBSCRIPTIONID	The id of a subscription to list.	Comma separated list of subscription ids.	N

9.3.5.6.3 Response encoding

The following XML-Schema fragment defines the response to a ListSubscriptions operation:

```
<xsd:element name="ListSubscriptionsResponse">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element name="Subscription"
                      minOccurs="0" maxOccurs="unbounded">
            <xsd:complexType>
               <xsd:complexContent>
                  <xsd:extension base="gss:SubscriptionType">
                     <xsd:attribute name="sid"
                                    type="xsd:anyURI" use="required"/>
                  </xsd:extension>
               </xsd:complexContent>
            </xsd:complexType>
         </xsd:element>
      </xsd:sequence>
      <xsd:attributeGroup ref="gss:BaseAttributes"/>
      <xsd:attribute name="owner"
                     type="xsd:string" use="required"/>
   </xsd:complexType>
</xsd:element>
```

The owner attribute on the gss:ListSubscriptionsResponse element shall contain the name of the owner of the listed subscriptions.

The response consists if a list of gss:Subscription elements (see 9.2.6.3) each describing a subscription. The attribute sid shall encode the identifier of each subscription.

In the event that an exception is encountered, the service shall respond with an exception report as specified in sub-clause 7.7.

9.3.5.6.4 Bindings

9.3.5.6.4.1 OGC bindings

9.3.5.6.4.1.1 POX with POST binding

9.3.5.6.4.1.1.1 Request

Sending an XML-encoded ListSubscriptions request to a server's base URL using the HTTP POST method shall generate a list of user subscriptions. The base POST URL for the Subscribe operation can be determined from the server's capabilities document (see 9.3.1.2).

9.3.5.6.4.1.1.2 Response

Upon successful execution of an ListSubscriptions operation the server shall respond by setting the HTTP status code to 200 and generating an XML document that contains a gss:ListSubscriptionsResponse element (see 9.3.5.6.3).

9.3.5.6.4.1.2 KVP with GET binding

9.3.5.6.4.1.2.1 Request

Sending an HTTP GET request to a URL formed by appending (see OGC 06-121r3, 11.2) the ListSubscriptions request parameters (see Table 19) to a server's base URL shall generate a list of user subscription. The base GET URL for the Subscribe operation can be determined from the server's capabilities document (see 9.3.1.2).

9.3.5.6.4.1.2.2 Response

See sub-clause 9.3.5.6.4.1.1.2.

9.3.5.6.4.2 REST binding

9.3.5.6.4.2.1 Request

Sending an HTTP GET request to a server's subscription URL (see 9.2.5.5) shall generate a list of user subscriptions.

9.3.5.6.4.2.2 Response

See sub-clause 9.3.5.6.4.1.1.2.

9.3.5.6.4.3 SOAP binding

Sending a SOAP message, whose body contains an XML-encoded ListSubscriptions request (see 9.3.5.6.2.1) , to a server's base URL using the HTTP POST method shall generate a SOAP response message whose body contains a gss:ListSubscriptionsResponse element (see 9.3.5.6.3).

9.3.5.7 PauseSubscription operation

9.3.5.7.1 Introduction

The **PauseSubscription** operation can be used to temporary suspend delivery of the specified subscription. The current list of user subscriptions can be discovered using the **ListSubscriptions** operation. The subscription to be pause is identified, in the request, using its id.

9.3.5.7.2 Request encoding

9.3.5.7.2.1 XML encoding

The following XML Schema fragment defines the XML encoding for PauseSubscription command:

```
<xsd:element name="PauseSubscription">
   <xsd:complexType>
      <xsd:complexContent>
         <xsd:extension base="gss:BaseRequestType">
            <xsd:sequence>
               <xsd:element name="SubscriptionId" type="xsd:anyURI"/>
               <xsd:sequence minOccurs="0">
                  <xsd:element name="Delivery"
                               type="gss:DeliveryMethodType" minOccurs="0"/>
                  <xsd:element name="Resume" type="gss:DateTime"/>
               </xsd:sequence>
            </xsd:sequence>
         </xsd:extension>
      </xsd:complexContent>
   </xsd:complexType>
</xsd:element>
```

The gss:SubscriptionId element specifies the id of the subscription to pause. No further notifications shall be sent until the subscription is resumed using the ResumeSubscription operation or the date specified by the gss:Resume element has elapsed.

The gss:Delivery element may be used to reset the delivery target when notification resumes on the time and date specified using the gss:Resume element. The gss:Delivery element may only be specified in the gss:Resume element has been specified.

Pausing an already pause paused subscription may be used to reset the resume time.

9.3.5.7.2.2 KVP encoding

Table 20 defines the parameters for the **PauseSubscription** operation.

Table 20 – PauseSubscription KVP request encoding

Parameter Name	Description	Value	Mandatory?
REQUEST	Name of operation	Fixed value of *PauseSubscription*	Y
SUBSCRIPTIONID	The ID of an existing subscription	Possible values obtained using the ListSubscriptions operation.	Y
DELIVERY	Method of delivery for notification of new entries	One of mailto, xmpp or sync.	N
RESUME	A date at which time notifications shall result.	DateTime	N
NOTE: The DELIVERY parameter may be specified only if the RESUME parameter has been specified.			

9.3.5.7.3 Response encoding

The following XML-Schema fragment defines the response to a PauseSubscription operation:

```
<xsd:element name="PauseSubscriptionResponse"
             type="gss:SubscriptionResponseType"/>
```

See sub-clause 7.6.2 for the declaration of gss:SubscriptionResponseType.

The server shall encode the subscription id of the paused subscription using the gss:SubscriptionId element in the response.

The server may also include a comment about the operation using the gss:ResponseComment element..

In the event that an exception is encountered, the service shall respond with an exception report as specified in sub-clause 7.7

9.3.5.7.4 Bindings

9.3.5.7.4.1 OGC bindings

9.3.5.7.4.1.1 POX with POST binding

9.3.5.7.4.1.1.1 Request

Sending an XML-encoded PauseSubscription request to a server's base URL using the HTTP POST method shall pause notification for the specified subscription. The base POST URL for the

PauseSubscription operation can be determined from the server's capabilities document (see 9.3.1.2).

9.3.5.7.4.1.1.2 Response

Upon successful execution of an PauseSubscription operation the server shall respond by setting the HTTP status code to 200 and generating an XML document that contains a gss:PauseSubscriptionResponse element (see 9.3.5.7.3).

9.3.5.7.4.1.2 KVP with GET binding

9.3.5.7.4.1.2.1 Request

Sending an HTTP GET request to a URL formed by appending (see OGC 06-121r3, 11.2) the PauseSubscription request parameters (see Table 20) to a server's base URL shall stop notification for the specified subscription. The base GET URL for the Subscribe operation can be determined from the server's capabilities document (see 9.3.1.2).

9.3.5.7.4.1.2.2 Response

See sub-clause 9.3.5.7.4.1.1.2.

9.3.5.7.4.2 REST binding

9.3.5.7.4.2.1 Request

Sending an HTTP PUT request, with a gss:Subscription element whose status attribute shall be set to "paused" as its contents, to the server's subscription URL (see 9.2.5.5) shall update the subscription and stop notification for the specified subscription.

9.3.5.7.4.2.2 Response

See sub-clause 9.3.5.7.4.1.1.2.

9.3.5.7.4.3 SOAP binding

Sending a SOAP message, whose body contains an XML-encoded PauseSubscription request (see 9.3.5.7.2.1) , to a server's base URL using the HTTP POST method shall generate a SOAP response message whose body contains a gss:PauseSubscriptionResponse element (see 9.3.5.7.3).

9.3.5.8 ResumeSubscription operation

9.3.5.8.1 Introduction

The ResumeSubscription operation can be used to resume notification from a previously paused subscription.

The ResumeSubscription operation can also be executed on an un-paused subscription in which case the operation can be used to reset the notification delivery endpoint and/or the expiry date.

9.3.5.8.2 Request encoding

9.3.5.8.2.1 XML encoding

The following XML Schema fragment defines the XML encoding for the ResumeSubscription operation:

```
<xsd:element name="ResumeSubscription">
   <xsd:complexType>
      <xsd:complexContent>
         <xsd:extension base="gss:BaseRequestType">
            <xsd:sequence>
               <xsd:element name="SubscriptionId" type="xsd:anyURI"/>
               <xsd:element name="Delivery"
                            type="gss:DeliveryMethodType" minOccurs="0"/>
               <xsd:element name="Expiry"
                            type="gss:dDteTime" minOccurs="0"/>
            </xsd:sequence>
         </xsd:extension>
      </xsd:complexContent>
   </xsd:complexType>
</xsd:element>
```

The gss:SubscriptionId element encodes the identifier of the subscription to be resumed.

The gss:Delivery element allows the delivery endpoint for notification to be reassigned upon resumption of notification (see 9.3.5.5.2.1).

The gss:Expiry element allows the expiry date of the subscription to be reset (see 9.3.5.5.2.1).

Resuming an already resumed subscription may be used to reset the delivery endpoint and the expiry time.

9.3.5.8.2.2 KVP encoding

Table 21 defines the parameters for the **ResumeSubscription** operation.

Table 21 – ResumeSubscription KVP request encoding

Parameter Name	Description	Value	Mandatory?
REQUEST	Name of operation	Fixed value of *ResumeSubscription*	Y
SUBSCRIPTIONID	The ID of an existing subscription	Possible values obtained using the ListSubscriptions operation.	Y
DELIVERY	Method of delivery for notification of new entries.	One of mailto, xmpp or sync.	N
EXPIRY	Expiry date for the subscription.	DateTime	N

9.3.5.8.3 Response encoding

The following XML-Schema fragment defines the response to a ResumeSubscription operation:

```
<xsd:element name="ResumeSubscriptionResponse"
             type="gss:SubscriptionResponseType"/>
```

See sub-clause 7.6.2 for the declaration of gss:SubscriptionResponseType.

The server shall encode the subscription id of the resumed subscription using the gss:SubscriptionId element in the response.

The server may also include a comment about the operation using the gss:ResponseComment element..

In the event that an exception is encountered, the service shall respond with an exception report as specified in sub-clause 7.7.

9.3.5.8.4 Bindings

9.3.5.8.4.1 OGC bindings

9.3.5.8.4.1.1 POX with POST binding

9.3.5.8.4.1.1.1 Request

Sending an XML-encoded ResumeSubscription request to a server's base URL using the HTTP POST method shall resume notification for the specified subscription. The base POST URL for the ResumeSubscription operation can be determined from the server's capabilities document (see 9.3.1.2).

9.3.5.8.4.1.1.2 Response

Upon successful execution of a ResumeSubscription operation the server shall respond by setting the HTTP status code to 200 and generating an XML document that contains a gss:ResumeSubscriptionResponse element (see 9.3.5.8.3).

9.3.5.8.4.1.2 KVP with GET binding

9.3.5.8.4.1.2.1 Request

Sending an HTTP GET request to a URL formed by appending (see OGC 06-121r3, 11.2) the ResumeSubscription request parameters (see Table 21) to a server's base URL shall resume notification for the specified subscription. The base GET URL for the ResumeSubscription operation can be determined from the server's capabilities document (see 9.3.1.2).

9.3.5.8.4.1.2.2 Response

See sub-clause 9.3.5.8.4.1.1.2.

9.3.5.8.4.2 REST binding

9.3.5.8.4.2.1 Request

Sending an HTTP PUT request, with a gss:ResumeSubscription element as its contents, to the server's subscription URL (see 9.2.5.5) shall stop notification for the specified subscription.

9.3.5.8.4.2.2 Response

See sub-clause 9.3.5.8.4.1.1.2.

9.3.5.8.4.3 SOAP binding

Sending a SOAP message, whose body contains an XML-encoded ResumeSubscription request (see 9.3.5.8.2.1) , to a server's base URL using the HTTP POST method shall generate a SOAP response message whose body contains a gss:ResumeSubscriptionResponse element (see 9.3.5.8.3).

9.3.5.9 Cancel Subscription operation

9.3.5.9.1 Introduction

The **CancelSubscription** operation can be used to permanently suspend delivery of and remove the specified subscription.

Upon cancelling a subscription all current subscribers shall be sent a termination notification indicating that no further notifications shall be received for the specified subscription.

9.3.5.9.2 Request encoding

9.3.5.9.2.1 XML encoding

The following XML Schema fragment defines the XML encoding for the CancelSubscription operation:

```
<xsd:element name="CancelSubscription">
   <xsd:complexType>
      <xsd:complexContent>
         <xsd:extension base="gss:BaseRequestType">
            <xsd:sequence>
               <xsd:element name="SubscriptionId" type="xsd:anyURI"/>
            </xsd:sequence>
         </xsd:extension>
      </xsd:complexContent>
   </xsd:complexType>
</xsd:element>
```

The gss:SubscriptionId element shall be used to encode the identifier of the subscription to be cancelled.

9.3.5.9.2.2 KVP encoding

Table 22 defines the parameters for the **CancelSubscription** operation.

Table 22 – CancelSubscription KVP request encoding

Parameter Name	Description	Value	Mandatory ?
REQUEST	Name of operation	Fixed value of *CancelSubscription*	Y
SUBSCRIPTIONID	The ID of an existing subscription	Possible values obtained using the ListSubscriptions operation.	Y

9.3.5.9.3 Response encoding

The following XML-Schema fragment defines the response to a CancelSubscription operation:

```
<xsd:element name="CancelSubscriptionResponse"
             type="gss:SubscriptionResponseType"/>
```

See sub-clause 7.6.2 for the declaration of gss:SubscriptionResponseType.

The server shall encode the subscription id of the cancelled subscription using the gss:SubscriptionId element in the response.

The server may also include a comment about the operation using the optional gss:ResponseComment element.

In the event that an exception is encountered, the service shall respond with an exception report as specified in sub-clause 7.7.

9.3.5.9.4 Bindings

9.3.5.9.4.1 OGC bindings

9.3.5.9.4.1.1 POX with POST binding

9.3.5.9.4.1.1.1 Request

Sending an XML-encoded CancelSubscription request to a server's base URL using the HTTP POST method shall stop further notification and delete the specified subscription. The base POST URL for the CancelSubscription operation can be determined from the server's capabilities document (see 9.3.1.2).

9.3.5.9.4.1.1.2 Response

Upon successful execution of a CancelSubscription operation the server shall respond by setting the HTTP status code to 200 and generating an XML document that contains a gss:CancelSubscriptionResponse element (see 9.3.5.9.3).

9.3.5.9.4.1.2 KVP with GET binding

9.3.5.9.4.1.2.1 Request

Sending an HTTP GET request to a URL formed by appending (see OGC 06-121r3, 11.2) the CancelSubscription request parameters (see Table 22) to a server's base URL shall stop further notification and delete the specified subscription. The base GET URL for the CancelSubscription operation can be determined from the server's capabilities document (see 9.3.1.2).

9.3.5.9.4.1.2.2 Response

See sub-clause 9.3.5.9.4.1.1.2.

9.3.5.9.4.2 REST binding

9.3.5.9.4.2.1 Request

Sending an HTTP DELETE request to the server's subscription URL (see 9.2.5.5) shall stop further notification and delete the specified subscription.

9.3.5.9.4.2.2 Response

See sub-clause 9.3.5.9.4.1.1.2.

9.3.5.9.4.3 SOAP binding

Sending a SOAP message, whose body contains an XML-encoded CancelSubscription request (see 9.3.5.9.2.1) , to a server's base URL using the HTTP POST method shall generate a SOAP response message whose body contains a gss:CancelSubscriptionResponse element (see 9.3.5.9.3).

Annex A – Abstract test suite (Normative)

T.B.D.

Annex B – BNF for WKT geometry encoding (Normative)

- values for the "geom" parameter encoded using OGC Well Known Text representation

- here is the BNF for WKT from OGC 09-049:

```
<Geometry Tagged Text> := <Point Tagged Text>
                        | <LineString Tagged Text>
                        | <Polygon Tagged Text>
                        | <MultiPoint Tagged Text>
                        | <MultiLineString Tagged Text>
                        | <MultiPolygon Tagged Text>
                        | <GeometryCollection Tagged Text>

<Point Tagged Text> := POINT <Point Text>
<LineString Tagged Text> := LINESTRING <LineString Text>
<Polygon Tagged Text> := POLYGON <Polygon Text>
<MultiPoint Tagged Text> := MULTIPOINT <Multipoint Text>
<MultiLineString Tagged Text>:=MULTILINESTRING <MultiLineString Text>
<MultiPolygon Tagged Text> := MULTIPOLYGON <MultiPolygon Text>
<GeometryCollection Tagged Text> := GEOMETRYCOLLECTION
                                    <GeometryCollection Text>

<Point Text> := EMPTY | ( <Point> )

<Point> := <x> <y>
<x> := double precision literal
<y> := double precision literal

<LineString Text> := EMPTY
   | ( <Point> {, <Point> }* )
<Polygon Text> := EMPTY
   | ( <LineString Text> {, < LineString Text> }*)
<Multipoint Text> := EMPTY
   | ( <Point Text> {, <Point Text> }* )
<MultiLineString Text> := EMPTY
   | ( <LineString Text> {, < LineString Text> }* )
<MultiPolygon Text> := EMPTY
   | ( < Polygon Text> {, < Polygon Text> }* )
<GeometryCollection Text> := EMPTY
   | ( <Geometry Tagged Text> {, <Geometry Tagged Text> }* )
```

Annex C – Consolidated schemas (Normative)

The consolidated schema for the Geosynchronization service can be found at:

http://www.pvretano.com/schemas/gss/1.0.0

This is a temporary location that shall eventually be updated to point to a location in the OGC schema repository.

Annex D– WSDL (normative)

The WSDL description of the GSS interface can be found at:

http://www.pvretano.com/gss/1.0.0/wsdl

This is a temporary location that shall eventually be updated to point to a location in the OGC schema repository.

Annex E – Examples (informative)

E.1 Topic encoding

The following XML fragment illustrated the XML-encoding of a topic.

```
<?xml version="1.0" ?>
<gss:Topic
   xmlns:gss="http://www.opengis.net/gss/1.0"
   xmlns:fes="http://www.opengis.net/ogc"
   xmlns:ows="http://www.opengis.net/ows/1.1"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.opengis.net/gss/1.0
                       http://www.pvretano.com/schemas/gss/1.0.0/gss.xsd">
   <gss:Feed>CHANGEFEED</gss:Feed>
   <gss:Description>
      <ows:Title>A Test Topic</ows:Title>
   </gss:Description>
   <fes:Filter>
      <fes:PropertyIsEqualTo>
         <fes:PropertyName>atom:title</fes:PropertyName>
         <fes:Literal>SomeTitle</fes:Literal>
      </fes:PropertyIsEqualTo>
   </fes:Filter>
</gss:Topic>
```

E.2 Subscription encoding

The following XML fragment illustrates the XML-encoding of a subscription.

```
<?xml version="1.0" ?>
<gss:Subscription
   xmlns:gss="http://www.opengis.net/gss/1.0"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.opengis.net/gss/1.0
                       http://www.pvretano.com/schemas/gss/1.0.0/gss.xsd">
   <gss:TopicId>urn:uuid:13e92f40-d45f-4b19-8866-91b271f165f8</gss:TopicId>
   <gss:Delivery>mailto:pvretano@cubewerx.com</gss:Delivery>
</gss:Subscription>
```

E.3 GetCapabilities response example

The following XML document illustrates a GetCapabilities response that a GSS might generate. The document includes an emended AtomPub service document and an embedded OpenSearch description document.

```
<?xml version="1.0" encoding="UTF-8"?>
<GSS_Capabilities
   xmlns="http://www.opengis.net/gss/1.0"
   xmlns:os="http://a9.com/-/spec/opensearch/1.1/"
   xmlns:gss="http://www.opengis.net/gss/1.0"
   xmlns:ows="http://www.opengis.net/ows/1.1"
   xmlns:gml="http://www.opengis.net/gml"
   xmlns:fes="http://www.opengis.net/ogc"
   xmlns:app="http://www.w3.org/2007/app"
   xmlns:atom="http://www.w3.org/2005/Atom"
   xmlns:xlink="http://www.w3.org/1999/xlink"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.opengis.net/gss/1.0
             http://www.pvretano.com/schemas/gss/1.0.0/gss.xsd
             http://www.w3.org/1999/xlink
             http://www.pvretano.com/schemas/xlink/1.0.0/xlinks.xsd
             http://www.opengis.net/ows/1.1
             http://www.pvretano.com/schemas/ows/1.1.0/owsAll.xsd
             http://www.opengis.net/ogc
             http://www.pvretano.com/schemas/filter/1.1.0/filter.xsd"
   version="1.0.0">
   <ows:ServiceIdentification>
      <ows:Title>GeoSynchronization Service</ows:Title>
      <ows:Abstract>GeoSynchronization is a method by which geospatial data
within a data store is kept current, and by which two or
more geospatial data stores can be kept "in-sync" with
each other, using existing OGC interfaces, and standards-based
subscription/notification models (RSS/ATOM).
The GeoSynchronization service interface uses the Atom Publishing
Protocol and Publish/Subscribe interface to provide functionality
that:

   -- advertises changes made to a geospatial database creates
         and edits GeoRSS feeds in support of data currency and
               replication, using the Atom Publishing Protocol

   -- alerts subscribers to the service of proposed changes,
         resolutions to those changes, and replication transactions
               available

   -- harvests those changes using the OpenSearch protocol

The data currency and replication capability of the GeoSynchronization
Service is accomplished by the use of feeds, managed by the Service.
Feeds are published by one data source, (the Publisher) and subscribed
to by another data source (the Subscriber).  The Publisher adds an entry
in its feed for any change made to its database.  The Subscriber
receives alerts when new transactions have occurred, retrieves their
contents, and applies them to its database.

The GeoSynchronization service uses the draft OpenSearch protocol to
provide discovery and query of its feeds.</ows:Abstract>
      <ows:Keywords>
         <ows:Keyword>geo</ows:Keyword>
         <ows:Keyword>synchronization</ows:Keyword>
         <ows:Keyword>ATOM</ows:Keyword>
         <ows:Keyword>REST</ows:Keyword>
         <ows:Keyword>WFS</ows:Keyword>
         <ows:Keyword>GSS</ows:Keyword>
         <ows:Keyword>feed</ows:Keyword>
         <ows:Keyword>entry</ows:Keyword>
         <ows:Keyword>GeoRSS</ows:Keyword>
      </ows:Keywords>
      <ows:ServiceType>GSS</ows:ServiceType>
      <ows:ServiceTypeVersion>1.0.0</ows:ServiceTypeVersion>
   </ows:ServiceIdentification>
   <ows:ServiceProvider>
      <ows:ProviderName>Acme Inc.</ows:ProviderName>
      <ows:ProviderSite xlink:href="http://www.acme.com"/>
      <ows:ServiceContact>
         <ows:IndividualName>Willey E. Coyote</ows:IndividualName>
         <ows:PositionName>Senior Developer</ows:PositionName>
         <ows:ContactInfo>
            <ows:Phone>
               <ows:Voice>(647)700-7701</ows:Voice>
               <ows:Facsimile>(647)700-7702</ows:Facsimile>
            </ows:Phone>
            <ows:Address>
               <ows:DeliveryPoint>123 Main St.</ows:DeliveryPoint>
               <ows:City>Certain Endroit</ows:City>
               <ows:AdministrativeArea>Quebec</ows:AdministrativeArea>
               <ows:PostalCode>J8Y 6N5</ows:PostalCode>
               <ows:Country>Canada</ows:Country>

<ows:ElectronicMailAddress>wecoyote@acme.com</ows:ElectronicMailAddress>
            </ows:Address>
            <ows:OnlineResource xlink:href="http://www.acme.com"/>
         </ows:ContactInfo>
      </ows:ServiceContact>
   </ows:ServiceProvider>
   <ows:OperationsMetadata>
      <ows:Operation name="GetCapabilities">
         <ows:DCP>
            <ows:HTTP>
               <ows:Get xlink:href="http://www.acme.com/gss.cgi?"/>
               <ows:Post xlink:href="http://www.acme.com/gss.cgi"/>
            </ows:HTTP>
         </ows:DCP>
         <ows:Parameter name="AcceptVersions">
            <ows:AllowedValues>
               <ows:Value>1.0.0</ows:Value>
            </ows:AllowedValues>
         </ows:Parameter>
         <ows:Parameter name="AcceptFormats">
            <ows:AllowedValues>
               <ows:Value>text/xml</ows:Value>
            </ows:AllowedValues>
         </ows:Parameter>
         <ows:Parameter name="Sections">
            <ows:AllowedValues>
               <ows:Value>ServiceIdentification</ows:Value>
               <ows:Value>ServiceProvider</ows:Value>
               <ows:Value>OperationsMetadata</ows:Value>
               <ows:Value>Service</ows:Value>
               <ows:Value>SupportsGMLObjectTypeList</ows:Value>
               <ows:Value>Filter_Capabilities</ows:Value>
            </ows:AllowedValues>
         </ows:Parameter>
      </ows:Operation>
      <ows:Operation name="Transaction">
         <ows:DCP>
            <ows:HTTP>
               <ows:Post xlink:href="http://www.acme.com/gss.cgi"/>
            </ows:HTTP>
         </ows:DCP>
      </ows:Operation>
      <ows:Operation name="GetEntries">
         <ows:DCP>
            <ows:HTTP>
               <ows:Get xlink:href="http://www.acme.com/gss.cgi?"/>
               <ows:Post xlink:href="http://www.acme.com/gss.cgi"/>
            </ows:HTTP>
         </ows:DCP>
         <ows:Parameter name="outputFormat">
            <ows:AllowedValues>
               <ows:Value>application/atom+xml</ows:Value>
            </ows:AllowedValues>
         </ows:Parameter>
      </ows:Operation>
      <ows:Operation name="AcceptChange">
         <ows:DCP>
            <ows:HTTP>
               <ows:Get xlink:href="http://www.acme.com/gss.cgi?"/>
               <ows:Post xlink:href="http://www.acme.com/gss.cgi"/>
            </ows:HTTP>
         </ows:DCP>
      </ows:Operation>
      <ows:Operation name="RejectChange">
         <ows:DCP>
            <ows:HTTP>
               <ows:Get xlink:href="http://www.acme.com/gss.cgi?"/>
               <ows:Post xlink:href="http://www.acme.com/gss.cgi"/>
            </ows:HTTP>
         </ows:DCP>
      </ows:Operation>
      <ows:Operation name="CreateTopic">
         <ows:DCP>
            <ows:HTTP>
               <ows:Get xlink:href="http://www.acme.com/gss.cgi?"/>
               <ows:Post xlink:href="http://www.acme.com/gss.cgi"/>
            </ows:HTTP>
         </ows:DCP>
      </ows:Operation>
      <ows:Operation name="RemoveTopic">
         <ows:DCP>
            <ows:HTTP>
               <ows:Get xlink:href="http://www.acme.com/gss.cgi?"/>
               <ows:Post xlink:href="http://www.acme.com/gss.cgi"/>
            </ows:HTTP>
         </ows:DCP>
      </ows:Operation>
      <ows:Operation name="ListTopics">
         <ows:DCP>
            <ows:HTTP>
               <ows:Get xlink:href="http://www.acme.com/gss.cgi?"/>
               <ows:Post xlink:href="http://www.acme.com/gss.cgi"/>
            </ows:HTTP>
         </ows:DCP>
```

```
</ows:Operation>
<ows:Operation name="Subscribe">
   <ows:DCP>
      <ows:HTTP>
         <ows:Get xlink:href="http://www.acme.com/gss.cgi?"/>
         <ows:Post xlink:href="http://www.acme.com/gss.cgi"/>
      </ows:HTTP>
   </ows:DCP>
</ows:Operation>
<ows:Operation name="ListSubscriptions">
   <ows:DCP>
      <ows:HTTP>
         <ows:Get xlink:href="http://www.acme.com/gss.cgi?"/>
         <ows:Post xlink:href="http://www.acme.com/gss.cgi"/>
      </ows:HTTP>
   </ows:DCP>
</ows:Operation>
<ows:Operation name="PauseSubscription">
   <ows:DCP>
      <ows:HTTP>
         <ows:Get xlink:href="http://www.acme.com/gss.cgi?"/>
         <ows:Post xlink:href="http://www.acme.com/gss.cgi"/>
      </ows:HTTP>
   </ows:DCP>
</ows:Operation>
<ows:Operation name="ResumeSubscription">
   <ows:DCP>
      <ows:HTTP>
         <ows:Get xlink:href="http://www.acme.com/gss.cgi?"/>
         <ows:Post xlink:href="http://www.acme.com/gss.cgi"/>
      </ows:HTTP>
   </ows:DCP>
</ows:Operation>
<ows:Operation name="CancelSubscription">
   <ows:DCP>
      <ows:HTTP>
         <ows:Get xlink:href="http://www.acme.com/gss.cgi?"/>
         <ows:Post xlink:href="http://www.acme.com/gss.cgi"/>
      </ows:HTTP>
   </ows:DCP>
</ows:Operation>
<ows:Parameter name="service">
   <ows:AllowedValues>
      <ows:Value>GSS</ows:Value>
   </ows:AllowedValues>
</ows:Parameter>
<ows:Parameter name="version">
   <ows:AllowedValues>
      <ows:Value>1.0.0</ows:Value>
   </ows:AllowedValues>
</ows:Parameter>
<ows:Constraint name="srsName">
   <ows:AllowedValues>
      <ows:Value>urn:ogc:def:crs:EPSG::4326</ows:Value>
      <ows:Value>urn:ogc:def:crs:EPSG::84</ows:Value>
      <ows:Value>urn:ogc:def:crs:EPSG::3857</ows:Value>
      <ows:Value>urn:ogc:def:crs:OGC::AUTO42001</ows:Value>
      <ows:Value>urn:ogc:def:crs:OGC::AUTO42002</ows:Value>
```

```
<ows:Value>urn:ogc:def:crs:OGC::AUTO42003</ows:Value>
<ows:Value>urn:ogc:def:crs:OGC::AUTO42004</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::4267</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::27</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::4269</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::83</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::26703</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::26704</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::26705</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::26706</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::26707</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::26708</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::26709</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::26710</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::26711</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::26712</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::26713</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::26714</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::26715</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::26716</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::26717</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::26718</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::26719</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::26720</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::26721</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::26722</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::26903</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::26904</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::26905</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::26906</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::26907</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::26908</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::26909</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::26910</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::26911</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::26912</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::26913</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::26914</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::26915</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::26916</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::26917</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::26918</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::26919</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::26920</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::26921</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::26922</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::26923</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::32601</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::32602</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::32603</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::32604</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::32605</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::32606</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::32607</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::32608</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::32609</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::32610</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::32611</ows:Value>
```

```
<ows:Value>urn:ogc:def:crs:EPSG::32612</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::32613</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::32614</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::32615</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::32616</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::32617</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::32618</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::32619</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::32620</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::32621</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::32622</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::32623</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::32624</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::32625</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::32626</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::32627</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::32628</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::32629</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::32630</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::32631</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::32632</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::32633</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::32634</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::32635</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::32636</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::32637</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::32638</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::32639</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::32640</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::32641</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::32642</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::32643</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::32644</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::32645</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::32646</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::32647</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::32648</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::32649</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::32650</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::32651</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::32652</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::32653</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::32654</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::32655</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::32656</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::32657</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::32658</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::32659</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::32660</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::32701</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::32702</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::32703</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::32704</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::32705</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::32706</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::32707</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::32708</ows:Value>
<ows:Value>urn:ogc:def:crs:EPSG::32709</ows:Value>
```

```
            <ows:Value>urn:ogc:def:crs:EPSG::32710</ows:Value>
            <ows:Value>urn:ogc:def:crs:EPSG::32711</ows:Value>
            <ows:Value>urn:ogc:def:crs:EPSG::32712</ows:Value>
            <ows:Value>urn:ogc:def:crs:EPSG::32713</ows:Value>
            <ows:Value>urn:ogc:def:crs:EPSG::32714</ows:Value>
            <ows:Value>urn:ogc:def:crs:EPSG::32715</ows:Value>
            <ows:Value>urn:ogc:def:crs:EPSG::32716</ows:Value>
            <ows:Value>urn:ogc:def:crs:EPSG::32717</ows:Value>
            <ows:Value>urn:ogc:def:crs:EPSG::32718</ows:Value>
            <ows:Value>urn:ogc:def:crs:EPSG::32719</ows:Value>
            <ows:Value>urn:ogc:def:crs:EPSG::32720</ows:Value>
            <ows:Value>urn:ogc:def:crs:EPSG::32721</ows:Value>
            <ows:Value>urn:ogc:def:crs:EPSG::32722</ows:Value>
            <ows:Value>urn:ogc:def:crs:EPSG::32723</ows:Value>
            <ows:Value>urn:ogc:def:crs:EPSG::32724</ows:Value>
            <ows:Value>urn:ogc:def:crs:EPSG::32725</ows:Value>
            <ows:Value>urn:ogc:def:crs:EPSG::32726</ows:Value>
            <ows:Value>urn:ogc:def:crs:EPSG::32727</ows:Value>
            <ows:Value>urn:ogc:def:crs:EPSG::32728</ows:Value>
            <ows:Value>urn:ogc:def:crs:EPSG::32729</ows:Value>
            <ows:Value>urn:ogc:def:crs:EPSG::32730</ows:Value>
            <ows:Value>urn:ogc:def:crs:EPSG::32731</ows:Value>
            <ows:Value>urn:ogc:def:crs:EPSG::32732</ows:Value>
            <ows:Value>urn:ogc:def:crs:EPSG::32733</ows:Value>
            <ows:Value>urn:ogc:def:crs:EPSG::32734</ows:Value>
            <ows:Value>urn:ogc:def:crs:EPSG::32735</ows:Value>
            <ows:Value>urn:ogc:def:crs:EPSG::32736</ows:Value>
            <ows:Value>urn:ogc:def:crs:EPSG::32737</ows:Value>
            <ows:Value>urn:ogc:def:crs:EPSG::32738</ows:Value>
            <ows:Value>urn:ogc:def:crs:EPSG::32739</ows:Value>
            <ows:Value>urn:ogc:def:crs:EPSG::32740</ows:Value>
            <ows:Value>urn:ogc:def:crs:EPSG::32741</ows:Value>
            <ows:Value>urn:ogc:def:crs:EPSG::32742</ows:Value>
            <ows:Value>urn:ogc:def:crs:EPSG::32743</ows:Value>
            <ows:Value>urn:ogc:def:crs:EPSG::32744</ows:Value>
            <ows:Value>urn:ogc:def:crs:EPSG::32745</ows:Value>
            <ows:Value>urn:ogc:def:crs:EPSG::32746</ows:Value>
            <ows:Value>urn:ogc:def:crs:EPSG::32747</ows:Value>
            <ows:Value>urn:ogc:def:crs:EPSG::32748</ows:Value>
            <ows:Value>urn:ogc:def:crs:EPSG::32749</ows:Value>
            <ows:Value>urn:ogc:def:crs:EPSG::32750</ows:Value>
            <ows:Value>urn:ogc:def:crs:EPSG::32751</ows:Value>
            <ows:Value>urn:ogc:def:crs:EPSG::32752</ows:Value>
            <ows:Value>urn:ogc:def:crs:EPSG::32753</ows:Value>
            <ows:Value>urn:ogc:def:crs:EPSG::32754</ows:Value>
            <ows:Value>urn:ogc:def:crs:EPSG::32755</ows:Value>
            <ows:Value>urn:ogc:def:crs:EPSG::32756</ows:Value>
            <ows:Value>urn:ogc:def:crs:EPSG::32757</ows:Value>
            <ows:Value>urn:ogc:def:crs:EPSG::32758</ows:Value>
            <ows:Value>urn:ogc:def:crs:EPSG::32759</ows:Value>
            <ows:Value>urn:ogc:def:crs:EPSG::32760</ows:Value>
            <ows:Value>urn:ogc:def:crs:OGC::41001</ows:Value>
            <ows:Value>urn:ogc:def:crs:OGC::42101</ows:Value>
            <ows:Value>urn:ogc:def:crs:OGC::42103</ows:Value>
            <ows:Value>urn:ogc:def:crs:OGC::42105</ows:Value>
         </ows:AllowedValues>
      </ows:Constraint>
      <ows:Constraint name="SupportsSOAP">
         <ows:AllowedValues>
            <ows:Value>FALSE</ows:Value>
         </ows:AllowedValues>
      </ows:Constraint>
   </ows:OperationsMetadata>
   <app:service xmlns="http://www.w3.org/2007/app"
      xmlns:atom="http://www.w3.org/2005/Atom">
      <app:workspace>
         <atom:title>Geosynchronization Server Feeds</atom:title>
         <app:collection
            href="http://www.acme.com/feed/CHANGEFEED">
            <atom:title>GSS Change Feed</atom:title>
         </app:collection>
         <app:collection
            href="http://www.acme.com/feed/REPLICATIONFEED">
            <atom:title>GSS Replication Feed</atom:title>
         </app:collection>
         <app:collection
            href="http://www.acme.com/feed/RESOLUTIONFEED">
            <atom:title>GSS Resolution Feed</atom:title>
         </app:collection>
         <collection href="http://www.acme.com/gss.cgi/topic">
            <atom:title>Topics</atom:title>
            <accept>text/xml</accept>
         </collection>
         <collection href="http://www.acme.com/gss.cgi/subscription">
            <atom:title>Subscriptions</atom:title>
            <accept>text/xml</accept>
         </collection>
      </app:workspace>
   </app:service>
   <os:OpenSearchDescription>
      <os:ShortName>Geosynchronization Feed Search</os:ShortName>
      <os:Description>Search the feeds managed by a GSS.</os:Description>
      <os:Tags>Geosynchronization feed collaboration GSS search</os:Tags>
      <os:Contact>admin@acme.com</os:Contact>
      <os:Url type="application/atom+xml"
              rel="collection"
              template="http://www.acme.com/gss.cgi?
                        service=GSS&amp;
                        version=1.0.0&amp;
                        request=GetEntries&amp;
                        feed=CHANGEFEED&amp;
                        outputFormat=ATOM&amp;
                        startPosition={startIndex?}&amp;
                        maxEntries={count?}&amp;
                        searchTerms={searchTerms?}&amp;
                        bbox={geo:box?}&amp;
                        starttime={time:start?}&amp;
                        temporalOp=After"/>
      <os:Url type="application/atom+xml"
              rel="collection"
              template="http://www.acme.com/gss.cgi?
                        service=GSS&amp;
                        version=1.0.0&amp;
                        request=GetEntries&amp;
                        feed=RESOLUTIONFEED&amp;
                        outputFormat=ATOM&amp;
```

```
                        startPosition={startIndex?}&amp;
                        maxEntries={count?}&amp;
                        searchTerms={searchTerms?}&amp;
                        bbox={geo:box?}&amp;
                        starttime={time:start?}&amp;
                        temporalOp=After"/>
      <os:Url type="application/atom+xml"
              rel="collection"
              template="http://www.acme.com/gss.cgi?
                        service=GSS&amp;
                        version=1.0.0&amp;
                        request=GetEntries&amp;
                        feed=REPLICATIONFEED&amp;
                        outputFormat=ATOM&amp;
                        startPosition={startIndex?}&amp;
                        maxEntries={count?}&amp;
                        searchTerms={searchTerms?}&amp;
                        bbox={geo:box?}&amp;
                        starttime={time:start?}&amp;
                        temporalOp=After"/>
      <os:LongName>Acme Geosynchronization Service Feed Search</os:LongName>
      <os:Image height="64" width="64"
                type="image/png">http://acme.com/websearch.png</os:Image>
      <os:Query role="example" count="25" />
      <os:Developer>Example.com Development Team</os:Developer>
      <os:Attribution>Copyright 2005, Acme.com, Inc.</os:Attribution>
      <os:SyndicationRight>open</os:SyndicationRight>
      <os:AdultContent>false</os:AdultContent>
      <os:Language>en-us</os:Language>
      <os:OutputEncoding>UTF-8</os:OutputEncoding>
      <os:InputEncoding>UTF-8</os:InputEncoding>
   </os:OpenSearchDescription>
   <gss:ConformanceDeclaration>
      <gss:ConformanceClass
           name="Discovery_OGC_POX">true</gss:ConformanceClass>
      <gss:ConformanceClass
           name="Discovery_OGC_KVP">true</gss:ConformanceClass>
      <gss:ConformanceClass
           name="Discovery_SOAP">false</gss:ConformanceClass>
      <gss:ConformanceClass
           name="Transaction_OGC_POX">false</gss:ConformanceClass>
      <gss:ConformanceClass
           name="Transaction_REST">true</gss:ConformanceClass>
      <gss:ConformanceClass
           name="Transaction_SOAP">false</gss:ConformanceClass>
      <gss:ConformanceClass
           name="Query_OGC_POX">true</gss:ConformanceClass>
      <gss:ConformanceClass
           name="Query_OGC_KVP">true</gss:ConformanceClass>
      <gss:ConformanceClass
           name="Query_REST">false</gss:ConformanceClass>
      <gss:ConformanceClass
           name="Query_SOAP">false</gss:ConformanceClass>
      <gss:ConformanceClass
           name="ChangeManagement_OGC_POX">true</gss:ConformanceClass>
      <gss:ConformanceClass
           name="ChangeManagement_OGC_KVP">true</gss:ConformanceClass>
      <gss:ConformanceClass
```

```
        name="ChangeManagement_REST">false</gss:ConformanceClass>
   <gss:ConformanceClass
        name="ChangeManagement_SOAP">false</gss:ConformanceClass>
   <gss:ConformanceClass
        name="TopicManagement_OGC_POX">true</gss:ConformanceClass>
   <gss:ConformanceClass
        name="TopicManagement_OGC_KVP">true</gss:ConformanceClass>
   <gss:ConformanceClass
        name="TopicManagement_REST">false</gss:ConformanceClass>
   <gss:ConformanceClass
        name="TopicManagement_SOAP">false</gss:ConformanceClass>
   <gss:ConformanceClass
        name="ActiveNotification_OGC_POX">true</gss:ConformanceClass>
   <gss:ConformanceClass
        name="ActiveNotification_OGC_KVP">true</gss:ConformanceClass>
   <gss:ConformanceClass
        name="ActiveNotification_REST">false</gss:ConformanceClass>
   <gss:ConformanceClass
        name="ActiveNotification_SOAP">false</gss:ConformanceClass>
   <gss:ConformanceClass
        name="Synchronization_OGC_POX">true</gss:ConformanceClass>
   <gss:ConformanceClass
        name="Synchronization_OGC_KVP">true</gss:ConformanceClass>
   <gss:ConformanceClass
        name="Synchronization_REST">false</gss:ConformanceClass>
   <gss:ConformanceClass
        name="Synchronization_SOAP">false</gss:ConformanceClass>
</gss:ConformanceDeclaration>
<fes:Filter_Capabilities>
   <fes:Spatial_Capabilities>
      <fes:GeometryOperands>
         <fes:GeometryOperand>gml:Point</fes:GeometryOperand>
         <fes:GeometryOperand>gml:LineString</fes:GeometryOperand>
         <fes:GeometryOperand>gml:Polygon</fes:GeometryOperand>
         <fes:GeometryOperand>gml:Envelope</fes:GeometryOperand>
      </fes:GeometryOperands>
      <fes:SpatialOperators>
         <fes:SpatialOperator name="BBOX"/>
         <fes:SpatialOperator name="Equals"/>
         <fes:SpatialOperator name="Disjoint"/>
         <fes:SpatialOperator name="Intersects"/>
         <fes:SpatialOperator name="Touches"/>
         <fes:SpatialOperator name="Crosses"/>
         <fes:SpatialOperator name="Within"/>
         <fes:SpatialOperator name="Contains"/>
         <fes:SpatialOperator name="Overlaps"/>
      </fes:SpatialOperators>
   </fes:Spatial_Capabilities>
   <fes:Scalar_Capabilities>
      <fes:LogicalOperators/>
      <fes:ComparisonOperators>
         <fes:ComparisonOperator>LessThan</fes:ComparisonOperator>
         <fes:ComparisonOperator>GreaterThan</fes:ComparisonOperator>
         <fes:ComparisonOperator>LessThanEqualTo</fes:ComparisonOperator>
         <fes:ComparisonOperator>GreaterThanEqualTo</fes:ComparisonOperator>
         <fes:ComparisonOperator>EqualTo</fes:ComparisonOperator>
         <fes:ComparisonOperator>NotEqualTo</fes:ComparisonOperator>
         <fes:ComparisonOperator>Like</fes:ComparisonOperator>
```

```
            <fes:ComparisonOperator>Between</fes:ComparisonOperator>
            <fes:ComparisonOperator>NullCheck</fes:ComparisonOperator>
         </fes:ComparisonOperators>
         <fes:ArithmeticOperators>
            <fes:SimpleArithmetic/>
         </fes:ArithmeticOperators>
      </fes:Scalar_Capabilities>
      <fes:Id_Capabilities>
         <fes:Id_Element>gss:EntryId</fes:Id_Element>
      </fes:Id_Capabilities>
   </fes:Filter_Capabilities>
</GSS_Capabilities>
```

A GSS must also arrange for the AtomPub service document and the OpenSearch description document to be independently accessible. This candidate standard does not specify how that is to be accomplished, but a service that implements the OGC POX binding may use a vendor extension of the GetCapabilities request, using the Section parameter to accomplish this. For example:

http://www.acme.com/gss.cgi?service=GSS&version=1.1.0&request=GetCapabilities&section=app:Service&outputFormat=application/atompubsvc+xml

The use of the outputFormat parameter is a vendor extension that tells the server to generate the service document alone without the gss:Capabilities wrapper.

E.4 CreateTopic example

E.4.1 POX with POST binding

The following XML-encoded request sent to the server's HTTP POST URL for the CreateTopic operation will create a new. The server's HTTP GET URL for the CancelSubscription operation can be found in its capabilities document (see 9.3.1.2.3).

This example creates a topic with a spatial predicate that generates a feed containing replication entries within Val D'Or, Quebec.

```
<?xml version="1.0" ?>
<gss:CreateTopic
   service="GSS"
   version="1.0.0"
   xmlns:gss="http://www.opengis.net/gss/1.0"
   xmlns:gml="http://www.opengis.net/gml"
   xmlns:fes="http://www.opengis.net/ogc"
   xmlns:ows="http://www.opengis.net/ows/1.1"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.opengis.net/gss/1.0
                    http://www.pvretano.com/schemas/gss/1.0.0/gss.xsd">
   <gss:Topic>
      <gss:Feed>REPLICATIONFEED</gss:Feed>
      <gss:Description>
         <ows:Title>Val D'or Syncronization Feed</ows:Title>
         <ows:Abstract>This feed filters the replication feed to identfy
changes in the region of Val D'or, Quebec.</ows:Abstract>
      </gss:Description>
      <fes:Filter>
         <fes:Intersects>
```

```
            <fes:PropertyName>atom:entry/georss:where</fes:PropertyName>
            <gml:Polygon srsName="urn:ogc:def:crs:EPSG::4326">
               <gml:exterior>
                  <gml:LinearRing>
                     <gml:posList>48.123246933032               -77.808265686035
48.115683488677      -77.811698913574      48.109723627023      -77.823371887207
48.100553259308      -77.824058532715      48.098031121351      -77.864570617676
48.085647922138      -77.865257263184      48.089087998818      -77.853240966797
48.07945520427      -77.848434448242      48.080143320849      -77.839508056641
48.091839894493      -77.842254638672      48.089317329082      -77.830924987793
48.075785093733      -77.824058532715      48.076014483311      -77.817192077636
48.081290161361      -77.816505432129      48.080143320849      -77.808609008789
48.071197087392      -77.807922363281      48.071655906434      -77.803115844726
48.05238198435 -77.795906066894 48.03769224747 -77.786293029785 48.028049830821
-77.777366638183    48.032182516036    -77.771186828613    48.051463998531    -
77.773246765137    48.069820605719    -77.776679992675    48.078767078486    -
77.785606384277    48.086565298418    -77.785949707032    48.086106612324    -
77.778396606445    48.090922612296    -77.775650024414    48.080143320849    -
77.768440246582    48.080372690996    -77.738914489746    48.09115193438    -
77.732734680176    48.096426060045    -77.74097442627    48.103992338935    -
77.730674743653    48.116141910942    -77.734451293946    48.116371120541    -
77.76123046875    48.125538666026    -77.765350341797    48.134246318893    -
77.771530151367    48.133558926294    -77.785606384277    48.125997000355    -
77.791786193848    48.122101028191    -77.793846130371    48.123246933032    -
77.808265686035</gml:posList>
                  </gml:LinearRing>
               </gml:exterior>
            </gml:Polygon>
         </fes:Intersects>
      </fes:Filter>
   </gss:Topic>
</gss:CreateTopic>
```

E.4.2 REST binding

The following example creates a new topic by posting an XML-encoded gss:Topic element to the server's topic URL (see 9.2.5.3).

```
POST /gss.cgi/topic HTTP/1.1
Host: www.acme.com
Authorization: Basic YWRtaW46YWRtaW4=
Content-Type: text/xml
Content-Length: nnn

<?xml version="1.0"?>
<gss:Topic>
   <gss:Feed>REPLICATIONFEED</gss:Feed>
   <gss:Description>
      <ows:Title>Val D'or Syncronization Feed</ows:Title>
      <ows:Abstract>This feed filters the replication feed to identfy changes
in the region of Val D'or, Quebec.</ows:Abstract>
   </gss:Description>
   <fes:Filter>
      <fes:Intersects>
         <fes:PropertyName>atom:entry/georss:where</fes:PropertyName>
         <gml:Polygon srsName="urn:ogc:def:crs:EPSG::4326">
            <gml:exterior>
               <gml:LinearRing>
```

```
                <gml:posList>48.123246933032 -77.808265686035 48.115683488677
-77.811698913574    48.109723627023    -77.823371887207    48.100553259308    -
77.824058532715    48.098031121351    -77.864570617676    48.085647922138    -
77.865257263184    48.089087998818    -77.853240966797    48.07945520427     -
77.848434448242    48.080143320849    -77.839508056641    48.091839894493    -
77.842254638672    48.089317329082    -77.830924987793    48.075785093733    -
77.824058532715    48.076014483311    -77.817192077636    48.081290161361    -
77.816505432129    48.080143320849    -77.808609008789    48.071197087392    -
77.807922363281    48.071655906434    -77.803115844726    48.05238198435     -
77.795906066894    48.03769224747     -77.786293029785    48.028049830821    -
77.777366638183    48.032182516036    -77.771186828613    48.051463998531    -
77.773246765137    48.069820605719    -77.776679992675    48.078767078486    -
77.785606384277    48.086565298418    -77.785949707032    48.086106612324    -
77.778396606445    48.090922612296    -77.775650024414    48.080143320849    -
77.768440246582    48.080372690996    -77.738914489746    48.09115193438     -
77.732734680176    48.096426060045    -77.74097442627     48.103992338935    -
77.730674743653    48.116141910942    -77.734451293946    48.116371120541    -
77.76123046875     48.125538666026    -77.765350341797    48.134246318893    -
77.771530151367    48.133558926294    -77.785606384277    48.125997000355    -
77.791786193848    48.122101028191    -77.793846130371    48.123246933032    -
77.808265686035</gml:posList>
               </gml:LinearRing>
            </gml:exterior>
         </gml:Polygon>
      </fes:Intersects>
   </fes:Filter>
</gss:Topic>
```

E.4.3 Response

The following XML fragment illustrates the GSS response to CreateTopic operation.

```
HTTP/1.1 201 Created
Date: Mon, 05 Apr 2010 14:46:32 GMT
Server: Apache/2.0.46 (Red Hat)
Content-Length: nnn
Content-Type: text/xml

<?xml version="1.0" ?>
<gss:CreateTopicResponse
   service="GSS"
   version="1.0.0"
   xmlns:gss="http://www.opengis.net/gss/1.0"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.opengis.net/gss/1.0
                    http://www.pvretano.com/schemas/gss/1.0.0/gss.xsd">
   <gss:TopicId>urn:uuid:57aeeeac-8111-4cfb-8bcf-2673ab9674d0</gss:TopicId>
   <gss:Url>http://www.acme.com/topic/urn:uuid:57aeeeac-8111-4cfb-8bcf-
2673ab9674d0</gss:Url>
   <gss:ResponseComment xml:lang="en">Topic created.</gss:ResponseComment>
   <gss:ResponseComment          xml:lang="fr">La          matiÃ¨re          a
crÃ©Ã©.</gss:ResponseComment>
   <gss:ResponseComment              xml:lang="el">Î¤Î¿              Î¸ÎÎ¼Î±
Î´Î·Î¼Î¹Î¿Ï…ÏÎ³Î®Î¸Î·ÎºÎµ.</gss:ResponseComment>
</gss:CreateTopicResponse>
```

E.5 Remove topic example

E.5.1 POX with POST binding

The following XML-encoded request sent to the server's HTTP POST URL for the RemoveTopic operation will remove an existing topic. The server's HTTP POST URL for the RemoveTopic operation can be found in its capabilities document (see 9.3.2.1.3).

```
<?xml version="1.0" ?>
<gss:RemoveTopic
   service="GSS"
   version="1.0.0"
   topicId="urn:uuid:57aeeeac-8111-4cfb-8bcf-2673ab9674d0"
   xmlns:gss="http://www.opengis.net/gss/1.0"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.opengis.net/gss/1.0
                      http://www.pvretano.com/schemas/gss/1.0.0/gss.xsd"/>
```

E.5.2 REST binding

Sending an HTTP DELETE request to the topic's URL will remove the topic from the collection of topics:

```
DELETE /gss.cgi/topic/urn:uuid:57aeeeac-8111-4cfb-8bcf-2673ab9674d0 HTTP/1.1
Host: www.acme.com
Authorization: Basic YWRtaW46YWRtaW4=
```

E.5.3 Response

The following is an example of a response to a RemoveTopic operation.

```
HTTP/1.1 200 OK
Date: Mon, 05 Apr 2010 14:46:32 GMT
Server: Apache/2.0.46 (Red Hat)
Content-Length: nnn
Content-Type: text/xml

<?xml version="1.0" ?>
<gss:RemoveTopicResponse
   service="GSS"
   version="1.0.0"
   xmlns:gss="http://www.opengis.net/gss/1.0"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.opengis.net/gss/1.0
                      http://www.pvretano.com/schemas/gss/1.0.0/gss.xsd">
   <gss:TopicId>urn:uuid:57aeeeac-8111-4cfb-8bcf-2673ab9674d0</gss:TopicId>
   <gss:ResponseComment xml:lang="en">Topic removed.</gss:ResponseComment>
   <gss:ResponseComment      xml:lang="fr">La      matiÃ¨re      a      Ã©tÃ©
enlevÃ©e..</gss:ResponseComment>
   <gss:ResponseComment              xml:lang="el">Î¤Î¿              Î¸ÎÎ¼Î±
Î±Ï†Î±Î¹Ï□ÎÎ¸Î·ÎºÎµ.</gss:ResponseComment>
</gss:RemoveTopicResponse>
```

E.6 ListTopics example

E.6.1 POX with POST binding

The following XML-encoded request sent to the server's HTTP POST URL for the ListTopics operation will generate a list of available topics. The server's HTTP POST URL for the ListTopics operation can be found in its capabilities document (see 9.3.2.1.3).

```
<?xml version="1.0" ?>
<gss:ListTopics
   service="GSS"
   version="1.0.0"
   xmlns:gss="http://www.opengis.net/gss/1.0"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.opengis.net/gss/1.0
                       http://www.pvretano.com/schemas/gss/1.0.0/gss.xsd"/>
```

E.6.2 KVP with GET binding

The following KVP-encoded request sent to the server's HTTP GET URL for the ListTopics operation will generate a list of available topics. The server's HTTP GET URL for the ListTopics operation can be found in its capabilities document (see 9.3.1.2.3).

```
http://www.acme.com/gss.cgi?service=GSS&version=1.0.0&request=ListTopics
```

E.6.3 REST binding

Sending an HTTP GET request to a server's topic URL will generate a list of all the topics offered by a GSS.

```
GET /gss.cgi/topic HTTP/1.1
Host: www.acme.com
Authorization: Basic YWRtaW46YWRtaW4=
```

E.6.4 Response

The following is an example of a response to a ListTopics operation:

```
HTTP/1.1 200 OK
Date: Mon, 05 Apr 2010 14:46:32 GMT
Server: Apache/2.0.46 (Red Hat)
Content-Length: nnn
Content-Type: text/xml

<?xml version="1.0" ?>
<gss:ListTopicsResponse
   service="GSS"
   version="1.0.0"
   xmlns:gml="http://www.opengis.net/gml"
   xmlns:gss="http://www.opengis.net/gss/1.0"
   xmlns:ows="http://www.opengis.net/ows/1.1"
   xmlns:fes="http://www.opengis.net/ogc"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.opengis.net/gss/1.0
                       http://www.pvretano.com/schemas/gss/1.0.0/gss.xsd">
   <gss:Topic
      owner="romain"
      tid="urn:uuid:512f31db-d82a-4a59-bd67-4f04725e75a4">
      <gss:Feed>CHANGEFEED</gss:Feed>
      <gss:Description>
         <ows:Title>A Test Topic</ows:Title>
      </gss:Description>
      <fes:Filter>
         <fes:PropertyIsEqualTo>
               <fes:PropertyName>atom:title</fes:PropertyName>
            <fes:Literal>SomeTitle</fes:Literal>
         </fes:PropertyIsEqualTo>
      </fes:Filter>
   </gss:Topic>
   <gss:Topic
      owner="gaetan"
      tid="urn:uuid:57aeeeac-8111-4cfb-8bcf-2673ab9674d0">
      <gss:Feed>REPLICATIONFEED</gss:Feed>
      <gss:Description>
         <ows:Title>Val D'or Syncronization Feed</ows:Title>
         <ows:Abstract>This  feed  filters  the  replication  feed  to  identfy
changes in the region of Val D'or, Quebec.</ows:Abstract>
      </gss:Description>
      <fes:Filter>
         <fes:Intersects>
            <fes:PropertyName>atom:entry/georss:where</fes:PropertyName>
            <gml:Polygon srsName="urn:ogc:def:crs:EPSG::4326">
               <gml:exterior>
                  <gml:LinearRing>
                     <gml:posList>48.123246933032               -77.808265686035
48.115683488677      -77.811698913574      48.109723627023      -77.823371887207
48.100553259308      -77.824058532715      48.098031121351      -77.864570617676
48.085647922138      -77.865257263184      48.089087998818      -77.853240966797
48.07945520427      -77.848434448242      48.080143320849      -77.839508056641
48.091839894493      -77.842254638672      48.089317329082      -77.830924987793
48.075785093733      -77.824058532715      48.076014483311      -77.817192077636
48.081290161361      -77.816505432129      48.080143320849      -77.808609008789
48.071197087392      -77.807922363281      48.071655906434      -77.803115844726
48.05238198435 -77.795906066894 48.03769224747 -77.786293029785 48.028049830821
-77.777366638183    48.032182516036    -77.771186828613    48.051463998531    -
77.773246765137    48.069820605719    -77.776679992675    48.078767078486    -
77.785606384277    48.086565298418    -77.785949707032    48.086106612324    -
77.778396606445    48.090922612296    -77.775650024414    48.080143320849    -
77.768440246582    48.080372690996    -77.738914489746    48.09115193438    -
77.732734680176    48.096426060045    -77.74097442627    48.103992338935    -
77.730674743653    48.116141910942    -77.734451293946    48.116371120541    -
77.76123046875    48.125538666026    -77.765350341797    48.134246318893    -
77.771530151367    48.133558926294    -77.785606384277    48.125997000355    -
77.791786193848    48.122101028191    -77.793846130371    48.123246933032    -
77.808265686035</gml:posList>
                  </gml:LinearRing>
               </gml:exterior>
            </gml:Polygon>
         </fes:Intersects>
      </fes:Filter>
   </gss:Topic>
</gss:ListTopicsResponse>
```

E.7 Subscribe example

E.7.1 POX with POST binding

The following XML-encoded request sent to the server's HTTP POST URL for the Subscribe operation will subscribe the current user to the specified topic. The server's HTTP POST URL for the Subscribe operation can be found in its capabilities document (see 9.3.1.2.3).

```
<?xml version="1.0" ?>
<gss:Subscribe
   service="GSS"
   version="1.0.0"
   xmlns:gss="http://www.opengis.net/gss/1.0"
   xmlns:gml="http://www.opengis.net/gml"
   xmlns:fes="http://www.opengis.net/ogc"
   xmlns:ows="http://www.opengis.net/ows/1.1"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.opengis.net/gss/1.0
                       http://www.pvretano.com/schemas/gss/1.0.0/gss.xsd">
   <gss:TopicId>urn:uuid:57aeeeac-8111-4cfb-8bcf-2673ab9674d0</gss:TopicId>
   <gss:Delivery>mailto:roadrunner@desert.com</gss:Delivery>
   <gss:Expiry>2010-10-13</gss:Expiry>
</gss:Subscribe>
```

E.7.2 KVP wth GET binding

The following KVP-encoded request sent to the server's HTTP GET URL for the Subscribe operation will subscribe the current user to the specified topic. The server's HTTP GET URL for the Subscribe operation can be found in its capabilities document (see 9.3.1.2.3).

```
http://www.acme.com/gss.cgi?service=GSS&version=1.0.0&request=Subscribe&topicId
=                                              urn:uuid:57aeeeac-8111-4cfb-8bcf-
2673ab9674d0&delivery=mailto:roadrunner@desert.com&expiry=2010-10-13
```

E.7.3 REST binding

Sending a HTTP POST request containing a gss:Subscribe element to a server's subscription URL (see 9.2.5.5) will subscribe the current user to the specified topic.

```
POST /gss.cgi/subscription HTTP/1.1
Host: www.acme.com
Authorization: Basic YWRtaW46YWRtaW4=
Content-Type: text/xml
Content-Length: nnn

<?xml version="1.0" ?>
<gss:Subscription
   xmlns:gss="http://www.opengis.net/gss/1.0"
   xmlns:gml="http://www.opengis.net/gml"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.opengis.net/gss/1.0
                       http://www.pvretano.com/schemas/gss/1.0.0/gss.xsd">
   <gss:TopicId>urn:uuid:57aeeeac-8111-4cfb-8bcf-2673ab9674d0</gss:TopicId>
   <gss:Delivery>mailto:roadrunner@desert.com</gss:Delivery>
   <gss:Expiry>2010-10-13</gss:Expiry>
</gss:Subscription>
```

E.7.4 Response

The following is an example response to a Subscribe operation. Attention is drawn to the fact that the HTTP status is set to 201, Created. The content of the response is a gss:SubscribeResponse element.

```
HTTP/1.1 201 Created
Date: Mon, 05 Apr 2010 14:46:32 GMT
Server: Apache/2.0.46 (Red Hat)
Content-Length: nnn
Content-Type: text/xml

<gss:SubscribeResponse
   service="GSS"
   version="1.0.0"
   xmlns:gss="http://www.opengis.net/gss/1.0"
   xmlns:gml="http://www.opengis.net/gml"
   xmlns:fes="http://www.opengis.net/ogc"
   xmlns:ows="http://www.opengis.net/ows/1.1"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.opengis.net/gss/1.0
                      http://www.pvretano.com/schemas/gss/1.0.0/gss.xsd">
   <gss:SubscriptionId>urn:uuid:5a933c0d-85ff-49cf-9c50-
d9b4883e46a4</gss:SubscriptionId>
   <gss:Url>http://www.acme.com/subscription/urn:uuid:5a933c0d-85ff-49cf-9c50-
d9b4883e46a4</gss:Url>
   <gss:ResponseComment                             xml:lang="en">Subscription
created.</gss:ResponseComment>
   <gss:ResponseComment        xml:lang="fr">L'abonnement        a        Ã©tÃ©
crÃ©Ã©.</gss:ResponseComment>
   <gss:ResponseComment            xml:lang="el">Î—            ÏƒÏ…Î½Î´ÏΠÎ¿Î¼Î®
Î´Î·Î¼Î¹Î¿Ï…ÏΠÎ³Î®Î¸Î·ÎºÎµ.</gss:ResponseComment>
</gss:SubscribeResponse>
```

E.8 ListSubscriptions example

E.8.1 POX with POST binding

The following XML-encoded request sent to the server's HTTP POST URL for the ListSubscriptions operation will list all subscriptions for the current user. The server's HTTP POST URL for the ListSubscriptions operation can be found in its capabilities document (see 9.3.1.2.3).

```
<?xml version="1.0" ?>
<gss:ListSubscriptions
   service="GSS"
   version="1.0.0"
   xmlns:gss="http://www.opengis.net/gss/1.0"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.opengis.net/gss/1.0
                      http://www.pvretano.com/schemas/gss/1.0.0/gss.xsd"/>
```

E.8.2 KVP with GET binding

The following KVP-encoded request sent to the server's HTTP GET URL for the ListSubscriptions operation will list all subscriptions for the current user. The server's HTTP GET URL for the ListSubscriptions operation can be found in its capabilities document (see 9.3.1.2.3).

```
http://www.acme.com/gss.cgi?service=GSS&version=1.0.0&request=ListSubscriptions
```

E.8.3 REST binding

Sending a HTTP GET request to a server's subscription URL lists all subscriptions for the current user:

```
GET /gss.cgi/topic HTTP/1.1
Host: www.acme.com
Authorization: Basic YWRtaW46YWRtaW4=
```

E.8.4 Response

The following is an example of a response to a ListSubscriptions operation:

```
HTTP/1.1 200 OK
Date: Mon, 05 Apr 2010 14:46:32 GMT
Server: Apache/2.0.46 (Red Hat)
Content-Length: nnn
Content-Type: text/xml

<?xml version="1.0" ?>
<gss:ListSubscriptionsResponse
   service="GSS"
   version="1.0.0"
   owner="roadrunner"
   xmlns:gss="http://www.opengis.net/gss/1.0"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.opengis.net/gss/1.0
                       http://www.pvretano.com/schemas/gss/1.0.0/gss.xsd">
   <gss:Subscription
      status="active"
      sid="urn:uuid:5a933c0d-85ff-49cf-9c50-d9b4883e46a4">
      <gss:TopicId>urn:uuid:57aeeeac-8111-4cfb-8bcf-2673ab9674d0</gss:TopicId>
      <gss:Delivery>mailto:roadrunner@desert.com</gss:Delivery>
      <gss:Expiry>2010-10-13</gss:Expiry>
   </gss:Subscription>
   <gss:Subscription
      status="active"
      sid="urn:uuid:abd12195-4afa-48cd-af98-fb2c29aff1c9">
      <gss:TopicId>urn:uuid:cf5fdc72-d55b-4ce7-8b4b-035b7197a8b4</gss:TopicId>
      <gss:Delivery>mailto:bbunny@holeintheground.com</gss:Delivery>
   </gss:Subscription>
   <gss:Subscription
      status="paused"
      sid="urn:uuid:abd12195-4afa-48cd-af98-fb2c29aff1c9">
      <gss:TopicId>urn:uuid:52d2c48c-3609-4ef3-b0b4-f13ea85217df</gss:TopicId>

<gss:Delivery>sync:http://www.planetmarscorp.com/server/wfs.cgi?service=WFS&amp
;version=1.1.0&amp;request=GetCapabilities</gss:Delivery>
   </gss:Subscription>
```

```
</gss:ListSubscriptionsResponse>
```

E.9 PauseSubscription example

E.9.1 POX with POST binding

The following XML-encoded request sent to the server's HTTP POST URL for the PauseSubscription operation will pause the specified subscription. The server's HTTP POST URL for the PauseSubscription operation can be found in its capabilities document (see 9.3.1.2.3).

```
<?xml version="1.0" ?>
<gss:PauseSubscription
   service="GSS"
   version="1.0.0"
   xmlns:gss="http://www.opengis.net/gss/1.0"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.opengis.net/gss/1.0
                       http://www.pvretano.com/schemas/gss/1.0.0/gss.xsd">
   <gss:SubscriptionId>urn:uuid:5a933c0d-85ff-49cf-9c50-
d9b4883e46a4</gss:SubscriptionId>
</gss:PauseSubscription>
```

E.9.2 KVP with GET binding

The following KVP-encoded request sent to the server's HTTP GET URL for the PauseSubscription operation will pause the specified subscription. The server's HTTP GET URL for the PauseSubscription operation can be found in its capabilities document (see 9.3.1.2.3).

```
http://www.acme.com/gss.cgi?service=GSS&version=1.1.0&request=PauseSubscription
&subscriptionId= urn:uuid:5a933c0d-85ff-49cf-9c50-d9b4883e46a4
```

E.9.3 REST binding

The following example sends an gss:Subscription element to a subscription's URL using the HTTP PUT method. The status attribute of the gss:Subscription element is set to "paused" indicating that the subscription should be paused.

```
PUT /gss.cgi/subscription/urn:uuid:5a933c0d-85ff-49cf-9c50-d9b4883e46a4 HTTP/1.1
Host: www.acme.com
Authorization: Basic YWRtaW46YWRtaW4=
Content-Type: text/xml
Content-Length: nnn

<?xml version="1.0" ?>
<gss:Subscription
   status="paused"
   xmlns:gss="http://www.opengis.net/gss/1.0"
   xmlns:gml="http://www.opengis.net/gml"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.opengis.net/gss/1.0
                       http://www.pvretano.com/schemas/gss/1.0.0/gss.xsd">
   <gss:TopicId>urn:uuid:57aeeeac-8111-4cfb-8bcf-2673ab9674d0</gss:TopicId>
   <gss:Delivery>mailto:roadrunner@desert.com</gss:Delivery>
   <gss:Expiry>2010-10-13</gss:Expiry>
</gss:Subscription>
```

E.9.4 Response

The following is an example of a response to a PauseSubscription operation:

```
HTTP/1.1 200 OK
Date: Mon, 05 Apr 2010 14:46:32 GMT
Server: Apache/2.0.46 (Red Hat)
Content-Length: nnn
Content-Type: text/xml

<?xml version="1.0"?>
<gss:PauseSubscriptionResponse
   service="GSS"
   version="1.0.0"
   xmlns:gss="http://www.opengis.net/gss/1.0"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.opengis.net/gss/1.0
                       http://www.pvretano.com/schemas/gss/1.0.0/gss.xsd">
   <gss:SubscriptionId>urn:uuid:5a933c0d-85ff-49cf-9c50-
d9b4883e46a4</gss:SubscriptionId>
   <gss:ResponseComment                               xml:lang="en">Subscription
paused.</gss:ResponseComment>
   <gss:ResponseComment   xml:lang="fr">L'abonnement    a    Ã©tÃ©    fait    une
pause.</gss:ResponseComment>
   <gss:ResponseComment            xml:lang="el">Î—            ÏƒÏ…Î½Î´Ï□Î¿Î¼Î®
ÏƒÏ„Î±Î¼Î±Ï„Î®Î¸Î·ÎºÎµ.</gss:ResponseComment>
</gss:PauseSubscriptionResponse>
```

E.10Resume subscription operation

E.10.1POX with POST binding

Sending the following XML-encoded ResumeSubscription request to a server's HTTP POST URL for the ResumeSubscription request will cause the server to resume sending notifications for the subscription. The server's HTTP POST URL for the ResumeSubscription operation can be found in it's capabilities document (9.3.1.2.3).

```
<?xml version="1.0" ?>
<gss:ResumeSubscription
   service="GSS"
   version="1.0.0"
   xmlns:gss="http://www.opengis.net/gss/1.0"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.opengis.net/gss/1.0
                       http://www.pvretano.com/schemas/gss/1.0.0/gss.xsd">
   <gss:SubscriptionId>urn:uuid:5a933c0d-85ff-49cf-9c50-
d9b4883e46a4</gss:SubscriptionId>
   <gss:Expiry>2010-12-01</gss:Expiry>
</gss:ResumeSubscription>
```

E.10.2 KVP with GET binding

The following KVP-encoded request sent to the server's HTTP GET URL for the ResumeSubscription operation will cause the server to resume sending notifications for the subscription. The server's HTTP GET URL for the ResumeSubscription operation can be found in its capabilities document (see 9.3.1.2.3).

```
http://www.acme.com/gss.cgi?service=GSS&version=1.0.0&request=ResumeSubscriptio
n&subscriptionId=  urn:uuid:5a933c0d-85ff-49cf-9c50-d9b4883e46a4&expiry=2010-12-
01
```

E.10.3 REST binding

The following example sends an gss:Subscription element to a subscription's URL using the HTTP PUT method. The status attribute of the gss:Subscription element is set to "active" indicating that subscription notification should be resumed. In addition, the gss:Expiry element is changed from 2010-10-13 to 2010-12-01 indicating that the expiry date should be updated too

```
PUT /gss.cgi/subscription/urn:uuid:5a933c0d-85ff-49cf-9c50-d9b4883e46a4 HTTP/1.1
Host: www.acme.com
Authorization: Basic YWRtaW46YWRtaW4=
Content-Type: text/xml
Content-Length: nnn

<?xml version="1.0" ?>
<gss:Subscription
   status="active"
   xmlns:gss="http://www.opengis.net/gss/1.0"
   xmlns:gml="http://www.opengis.net/gml"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.opengis.net/gss/1.0
                       http://www.pvretano.com/schemas/gss/1.0.0/gss.xsd">
   <gss:TopicId>urn:uuid:57aeeeac-8111-4cfb-8bcf-2673ab9674d0</gss:TopicId>
   <gss:Delivery>mailto:roadrunner@desert.com</gss:Delivery>
   <gss:Expiry>2010-10-13</gss:Expiry>
</gss:Subscription>
```

E.10.4 Response

The following is an example of a response to a ResumeSubscription operation:

```
HTTP/1.1 200 OK
Date: Mon, 05 Apr 2010 14:46:32 GMT
Server: Apache/2.0.46 (Red Hat)
Content-Length: nnn
Content-Type: text/xml

<?xml version="1.0"?>
<gss:ResumeSubscriptionResponse
   service="GSS"
   version="1.0.0"
   xmlns:gss="http://www.opengis.net/gss/1.0"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.opengis.net/gss/1.0
                       http://www.pvretano.com/schemas/gss/1.0.0/gss.xsd">
   <gss:SubscriptionId>urn:uuid:5a933c0d-85ff-49cf-9c50-
d9b4883e46a4</gss:SubscriptionId>
   <gss:ResponseComment  xml:lang="en">The  subscription  was  resumed  and  the
expiry was reset to 2010-12-31.</gss:ResponseComment>
   <gss:ResponseComment  xml:lang="fr">L'abonnement  a  Ã©tÃ©  repris  et  la  date
d'expiration d'Ã©chÃ©ance a Ã©tÃ© fixÃ©e Ã  2010-12-31.</gss:ResponseComment>
```

```
   <gss:ResponseComment            xml:lang="el">Î—            ÏƒÏ…Î½Î´Ï□Î¿Î¼Î®
ÎµÏ€Î±Î½Î±Î»Î®Ï†Î¸Î·ÎºÎµ ÎºÎ±Î¹ Î· Î·Î¼ÎµÏ□Î¿Î¼Î·Î½Î¯Î± Î»Î®Î¾Î·Ï‚ Ï„Î-Î¸Î·ÎºÎµ
2010-12-31.</gss:ResponseComment>
</gss:ResumeSubscriptionResponse>
```

E.11CancelSubscription example

E.11.1 POX with POST binding

Sending the following XML-encoded request to the server's HTTP POST URL for the CancelSubscription operation will cancel the subscription. The server's HTTP POST URL for the CancelSubscription operation can be found in its capabilities document (see 9.3.1.2.3).

```
<?xml version="1.0" ?>
<gss:CancelSubscription
   service="GSS"
   version="1.0.0"
   xmlns:gss="http://www.opengis.net/gss/1.0"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.opengis.net/gss/1.0
                       http://www.pvretano.com/schemas/gss/1.0.0/gss.xsd">
   <gss:SubscriptionId>urn:uuid:5a933c0d-85ff-49cf-9c50-
d9b4883e46a4</gss:SubscriptionId>
</gss:CancelSubscription>
```

E.11.2 KVP with GET binding

The following KVL-encoded request sent to the server's HTTP GET URL for the CancelSubscription operation will cancel the subscription. The server's HTTP GET URL for the CancelSubscription operation can be found in its capabilities document (see 9.3.1.2.3).

```
http://www.acme.com/gss.cgi?service=GSS&version=1.0.0&request=CancelSubscriptio
n&subscriptionId=urn:uuid:5a933c0d-85ff-49cf-9c50-d9b4883e46a4
```

E.11.3 REST binding

Sending an HTTP DELETE request to a subscription's URL will cancel the subscription.

```
DELETE /gss.cgi/subscription/urn:uuid:5a933c0d-85ff-49cf-9c50-d9b4883e46a4 HTTP/1.1
Host: www.acme.com
Authorization: Basic YWRtaW46YWRtaW4=
```

E.11.4 Response

The following is an example response to a CancelSubscription operation:

```
HTTP/1.1 200 OK
Date: Mon, 05 Apr 2010 14:46:32 GMT
Server: Apache/2.0.46 (Red Hat)
Content-Length: nnn
Content-Type: text/xml

<?xml version="1.0"?>
<gss:CancelSubscriptionResponse
   service="GSS"
   version="1.0.0"
   xmlns:gss="http://www.opengis.net/gss/1.0"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.opengis.net/gss/1.0
                       http://www.pvretano.com/schemas/gss/1.0.0/gss.xsd">
   <gss:SubscriptionId>urn:uuid:5a933c0d-85ff-49cf-9c50-
d9b4883e46a4</gss:SubscriptionId>
   <gss:ResponseComment xml:lang="en">The subscription was
cancelled.</gss:ResponseComment>
   <gss:ResponseComment xml:lang="fr">L'abonnement a Ã©tÃ©
dÃ©commandÃ©.</gss:ResponseComment>
   <gss:ResponseComment xml:lang="el">Î— ÏƒÏ…Î½Î´Ï□Î¿Î¼Î®
Î±Î°Ï…Ï□ÏŽÎ¸Î·Î°Îµ.</gss:ResponseComment>
</gss:CancelSubscriptionResponse>
```

E.12 AcceptChange example

E.12.1 POX with POST binding

The following XML-encoded request sent to the server's HTTP POST URL for the AcceptChange operation will cause the server to accept the change (see 9.3.4.2). The server's HTTP POST URL for the AcceptChange operation can be found in its capabilities document (see 9.3.1.2.3).

```
<?xml version="1.0" ?>
<gss:AcceptChange
   service="GSS"
   version="1.0.0"
   xmlns:gss="http://www.opengis.net/gss/1.0"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.opengis.net/gss/1.0
                       http://www.pvretano.com/schemas/gss/1.0.0/gss.xsd">
   <gss:EntryId>urn:uuid:8032ea88-069c-4fbc-8dfb-1a9a586df139</gss:EntryId>
   <gss:Comment>Change was accepted with a revision to the elevation. Latest
survey reports the elevation at the location is 2437m.</gss:Comment>
</gss:AcceptChange>
```

E.12.2 KVP with GET binding

The following KVP-encoded request sent to the server's HTTP GET URL for the AcceptChange operation will cause the server to accept the change (see 9.3.4.2). The server's HTTP GET URL for the AcceptChange operation can be found in its capabilities document (see 9.3.1.2.3).

```
http://www.acme.com/gss.cgi?service=GSS&version=1.0.0&request=AcceptChange&entr
yId= urn:uuid:8032ea88-069c-4fbc-8dfb-1a9a586df139
```

E.12.3 REST binding

```
T.B.D.
```

E.12.4 Response

The following is an example response to an AcceptChange operation.

```
HTTP/1.1 200 OK
Date: Mon, 05 Apr 2010 14:46:32 GMT
```

```
Server: Apache/2.0.46 (Red Hat)
Content-Length: nnn
Content-Type: text/xml

<?xml version="1.0"?>
<gss:AcceptChangeResponse
   service="GSS"
   version="1.0.0"
   xmlns:gss="http://www.opengis.net/gss/1.0"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.opengis.net/gss/1.0
                       http://www.pvretano.com/schemas/gss/1.0.0/gss.xsd">
   <gss:EntryId>urn:uuid:8032ea88-069c-4fbc-8dfb-1a9a586df139</gss:EntryId>
</gss:AcceptChangeResponse>
```

E.12.5 AcceptChange example with modifications

The following example shows an AcceptChange operation that includes a modification made by the reviewer. The modified transaction appears as the content of the gss:Revision element.

```
<?xml version="1.0" ?>
<gss:AcceptChange
   service="GSS"
   version="1.0.0"
   xmlns:gss="http://www.opengis.net/gss/1.0"
   xmlns:gml="http://www.opengis.net/gml"
   xmlns:wfs="http://www.opengis.org/wfs"
   xmlns:tor="http://www.toronto.ca/planning"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.opengis.net/gss/1.0
                       http://www.pvretano.com/schemas/gss/1.0.0/gss.xsd">
   <gss:EntryId>urn:uuid:8032ea88-069c-4fbc-8dfb-1a9a586df139</gss:EntryId>
   <gss:Comment>Change  was  accepted  with  a  revision  to  the  height  of  the
building.     The  latest  measurements  show  that  the  height  is  actually
49m.</gss:Comment>
   <gss:Revision>
      <wfs:Transaction>
         <wfs:Insert>
            <tor:Building>
               <tor:bid>urn:uuid:551b3fb4-f3cdf530588e</tor:bid>
               <tor:name>Skydome</tor:name>
               <tor:revisionDate>2010-10-24</tor:revisionDate>
               <tor:height uom="m">49</tor:height>
               <tor:footprint>
                  <gml:Polygon>
                     <gml:exterior>
                        <gml:LinearRing>
                           <gml:posList>43.642178026869         -79.390994310379
43.642721509483      -79.388269186020      43.640671777928      -79.387400150299
43.640632956580      -79.387550354004      43.640438849463      -79.387722015381
43.640306856265      -79.387872219086      43.640198155766      -79.388022422790
43.640143805444      -79.388269186020      43.640050633347      -79.388440847397
43.639972989823      -79.388676881790      43.639957461106      -79.389030933380
43.639941932385      -79.389245510102      43.639988518535      -79.389653205872
43.640011811597      -79.389739036560      43.640089455071      -79.389835596084
43.640058397694      -79.390189647674      43.642170262796      -79.390962123870
43.642178026869 -79.390994310379</gml:posList>
                        </gml:LinearRing>
```

```
                    </gml:exterior>
                 </gml:Polygon>
              </tor:footprint>
           </tor:Building>
        </wfs:Insert>
     </wfs:Transaction>
  </gss:Revision>
</gss:AcceptChange>
```

E.13 RejectChange example

E.13.1 POX with POST binding

The following XML-encoded request sent to the server's HTTP POST URL for the RejectChange operation will cause the server to reject the specified change (see 9.3.4.3). The server's HTTP POST URL for the RejectChange operation can be found in its capabilities document (see 9.3.1.2.3).

```
<?xml version="1.0" ?>
<gss:RejectChange
   service="GSS"
   version="1.0.0"
   xmlns:gss="http://www.opengis.net/gss/1.0"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.opengis.net/gss/1.0
                    http://www.pvretano.com/schemas/gss/1.0.0/gss.xsd">
   <gss:EntryId>urn:uuid:8032ea88-069c-4fbc-8dfb-1a9a586df139</gss:EntryId>
   <gss:Comment>Overlaying the proposed change on satelite imagery of vicinity
shows that the geometry of the revised feature deviates significatly from the
ground truth.</gss:Comment>
</gss:RejectChange>
```

E.13.2 KVP with GET binding

The following KVP-encoded request sent to the server's HTTP GET URL for the RejectChange operation will cause the server to reject the specified change (see 9.3.4.3). The server's HTTP GET URL for the RejectChange operation can be found in its capabilities document (see 9.3.1.2.3).

```
http://www.acme.com/gss.cgi?service=GSS&version=1.0.0&request=RejectChange&entr
yId= urn:uuid:8032ea88-069c-4fbc-8dfb-1a9a586df139
```

E.13.3 REST binding

T.B.D.

E.13.4 Response

The following is an example response to a RejectChange operation.

```
HTTP/1.1 200 OK
Date: Mon, 05 Apr 2010 14:46:32 GMT
Server: Apache/2.0.46 (Red Hat)
Content-Length: nnn
Content-Type: text/xml
```

```
<?xml version="1.0"?>
<gss:RejectChangeResponse
   service="GSS"
   version="1.0.0"
   xmlns:gss="http://www.opengis.net/gss/1.0"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.opengis.net/gss/1.0
                       http://www.pvretano.com/schemas/gss/1.0.0/gss.xsd">
   <gss:EntryId>urn:uuid:8032ea88-069c-4fbc-8dfb-1a9a586df139</gss:EntryId>
</gss:RejectChangeResponse>
```

E.14 GetEntries examples

E.14.1 Get a specific feed entry

The following GetEntries example fetches an entry using its identifier.

```
<?xml version="1.0" ?>
<gss:GetEntries
   service="GSS"
   version="1.0.0"
   feed="CHANGEFEED"
   xmlns:gss="http://www.opengis.net/gss/1.0"
   xmlns:fes="http://www.opengis.net/ogc"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.opengis.net/gss/1.0
                  http://www.pvretano.com/schemas/gss/1.0.0/gss.xsd
                  http://www.opengis.net/ogc
                  http://www.pvretano.com/schemas/filter/1.1.0/filter.xsd">
   <fes:Filter>
      <gss:EntryId eid="urn:uuid:c64563bc-1c9a-4ffd-a81e-9ee991d8ce40"/>
   </fes:Filter>
</gss:GetEntries>
```

E.14.2 Get the 25 most recent entries

The following GetEntries request fetches the 25 more recent entries from the RESOLUTIONFEED order by most-recent first.

```
<?xml version="1.0" ?>
<gss:GetEntries
   service="GSS"
   version="1.0.0"
   feed="RESOLUTIONFEED"
   maxEntries="25"
   xmlns:gss="http://www.opengis.net/gss/1.0"
   xmlns:gml="http://www.opengis.net/gml"
   xmlns:fes="http://www.opengis.net/ogc"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.opengis.net/gss/1.0
             http://www.pvretano.com/schemas/gss/1.0.0/gss.xsd"/>
```

E.14.3 GetEntries example with a temporal predicate

The following GetEntries example fetches all entries after 2010-10-13T02:10:00Z.

```
<?xml version="1.0" ?>
<gss:GetEntries
   service="GSS"
   version="1.0.0"
   feed="CHANGEFEED"
   xmlns:gss="http://www.opengis.net/gss/1.0"
   xmlns:fes="http://www.opengis.net/ogc"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.opengis.net/gss/1.0
                 http://www.pvretano.com/schemas/gss/1.0.0/gss.xsd
                 http://www.opengis.net/ogc
                 http://www.pvretano.com/schemas/filter/1.1.0/filter.xsd">
   <fes:Filter>
      <fes:PropertyIsGreaterThan>
         <fes:PropertyName>atom:entry/atom:updated</fes:PropertyName>
         <fes:Literal>2010-10-13T02:10:00Z</fes:Literal>
      </fes:PropertyIsGreaterThan>
   </fes:Filter>
</gss:GetEntries>
```

E.14.4 GetEntries example with a spatial predicate

The following GetEntries example fetches all entries whose geometry intersects with the specified geometry.

```
<?xml version="1.0" ?>
<gss:GetEntries
   service="GSS"
   version="1.0.0"
   feed="RESOLUTIONFEED"
   maxEntries="15"
   xmlns:gss="http://www.opengis.net/gss/1.0"
   xmlns:gml="http://www.opengis.net/gml"
   xmlns:fes="http://www.opengis.net/ogc"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.opengis.net/gss/1.0
             http://www.pvretano.com/schemas/gss/1.0.0/gss.xsd
             http://www.opengis.net/ogc
             http://www.pvretano.com/schemas/filter/1.1.0/filter.xsd">

   <fes:Filter>
      <fes:Intersects>
         <fes:PropertyName>atom:entry/georss:where</fes:PropertyName>
         <gml:Polygon srsName="urn:ogc:def:crs:EPSG::4326">
            <gml:exterior>
               <gml:LinearRing>
                  <gml:posList>43.579893442166  -79.543762207032  43.754233115516
-79.614486694336    43.858296779162    -79.153060913086    43.869188195488    -
79.137954711914    43.782992628906    -79.104995727539    43.727444586475    -
79.194946289063    43.696672605503    -79.223785400390    43.662904523837    -
79.270477294922    43.612713914499    -79.309616088867    43.600284023536    -
79.396133422851    43.621164773217    -79.407806396484    43.627129370169    -
79.449005126953    43.627626393202    -79.466171264648    43.615696706136    -
79.463424682617    43.582877861177    -79.505310058594    43.579396024615    -
79.545135498047 43.579893442166 -79.543762207032</gml:posList>
               </gml:LinearRing>
            </gml:exterior>
         </gml:Polygon>
      </fes:Intersects>
   </fes:Filter>
</gss:GetEntries>
```

E.15 Transaction example

E.15.1 POX with POST binding

The following XML-encoded request sent to the server's HTTP POST URL for the Transaction operation will insert, update or delete entries from the specified feed (see 9.3.2). The server's HTTP POST URL for the Transaction operation can be found in its capabilities document (see 9.3.1.2.3).

```
<?xml version="1.0" ?>
<gss:Transaction
   service="GSS"
   version="1.0.0"
   xmlns:gss="http://www.opengis.net/gss/1.0"
   xmlns:gml="http://www.opengis.net/gml"
   xmlns:wfs="http://www.opengis.org/wfs"
   xmlns:tor="http://www.toronto.ca/planning"
   xmlns:atom="http://www.w3.org/2005/Atom"
   xmlns:georss="http://www.georss.org/georss"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.opengis.net/gss/1.0
             http://www.pvretano.com/schemas/gss/1.0.0/gss.xsd">
   <gss:Insert feed="CHANGEFEED">
      <atom:entry>
         <atom:title>Insert         of         feature         urn:uuid:551b3fb4-
f3cdf530588e</atom:title>
         <atom:summary>
            The proposed change add the footprint of the newly
            built Skydome in Toronto, Ontario.
         </atom:summary>
         <atom:id>urn:uuid:1225c695-cfb8-4ebb-aaaa-80da344efa6a</atom:id>
         <atom:updated>2008-12-13T18:30:02Z</atom:updated>
         <atom:author>
            <atom:name>John Doe</atom:name>
            <atom:email>jdoe@toronto.ca</atom:email>
         </atom:author>
         <atom:category term="insert"
                        scheme="http://www.opengis.org/geosync/actions"/>
         <atom:link rel="targetServer"
                    href="http://www.toronto.com/landregistry/wfs.cgi"/>
         <atom:content type="application/gml+xml">
            <wfs:Transaction>
               <wfs:Insert>
                  <tor:Building>
                     <tor:bid>urn:uuid:551b3fb4-f3cdf530588e</tor:bid>
                     <tor:name>Skydome</tor:name>
                     <tor:revisionDate>2010-10-24</tor:revisionDate>
                     <tor:height uom="m">45</tor:height>
                     <tor:footprint>
                        <gml:Polygon>
                           <gml:exterior>
                              <gml:LinearRing>
```

```
                                 <gml:posList>43.642178026869   -79.390994310379
43.642721509483      -79.388269186020      43.640671777928      -79.387400150299
43.640632956580      -79.387550354004      43.640438849463      -79.387722015381
43.640306856265      -79.387872219086      43.640198155766      -79.388022422790
43.640143805444      -79.388269186020      43.640050633347      -79.388440847397
43.639972989823      -79.388676881790      43.639957461106      -79.389030933380
43.639941932385      -79.389245510102      43.639988518535      -79.389653205872
43.640011811597      -79.389739036560      43.640089455071      -79.389835596084
43.640058397694      -79.390189647674      43.642170262796      -79.390962123870
43.642178026869 -79.390994310379</gml:posList>
                              </gml:LinearRing>
                           </gml:exterior>
                        </gml:Polygon>
                     </tor:footprint>
                  </tor:Building>
               </wfs:Insert>
            </wfs:Transaction>
         </atom:content>
         <georss:where>
            <gml:Envelope>
               <gml:lowerCorner>-85.3486 10.1183</gml:lowerCorner>
               <gml:upperCorner>-10.8013 80.4831</gml:upperCorner>
            </gml:Envelope>
         </georss:where>
      </atom:entry>
   </gss:Insert>
</gss:Transaction>
```

E.15.2 REST example

The following example creates a new REPLICATIONFEED entry:

```
POST /gss.cgi/REPLICATIONFEED HTTP/1.1
Host: www.acme.com
User-Agent: geosyncclient/1.0
Authorization: Basic ZGFmZnk6c2VjZXJldA==
Content-Type: application/atom+xml;type=entry
Content-Length: nnn
Slug: fid435890453

<?xml version="1.0" encoding="UTF-8"?>
<entry xmlns="http://www.w3.org/2005/Atom"
       xmlns:gml="http://www.opengis.net/gml">
   <title>Insert of feature urn:uuid:551b3fb4-f3cdf530588e</title>
   <updated>2008-12-13T18:30:02Z</updated>
   <author>
      <name>John Doe</name>
      <email>jdoe@example.org</email>
   </author>
   <category scheme="http://www.opengis.org/geosync/actions" term="insert"/>
   <link rel="targetServer" href="http://www.someserver.com/wfs.cgi"/>
   <content type="text/xml; subtype=gml/3.1.1/profiles/gmlsf/1.0.0/0">
      <wfs:Transaction xmlns:wfs="http://www.opengis.org/wfs"
                       Xmlns:gb="http://example.org/roads">
         <wfs:Insert>
            <gb:RoadSegment>
               <gb:cgdiId>urn:uuid:551b3fb4-f3cdf530588e</gb:cgdiId>
               <gb:acquisitionProvider>n.a.</gb:acquisitionProvider>
               <gb:revisionDate>2007-10-24-04:00</gb:revisionDate>
               <gb:nationalRoadClass>Arterial</gb:nationalRoadClass>
               <gb:numberOfLanes>0</gb:numberOfLanes>
               <gb:pavementStatus>C</gb:pavementStatus>
               <gb:pavedRoadSurfaceType>GR</gb:pavedRoadSurfaceType>
               <gb:roadSegmentID>551b3fb4f330588e</gb:roadSegmentID>
               <gb:segment>
                  <gml:LineString
                     srsName="urn:ogc:def:crs:EPSG:6.11.2:4326">
                     <gml:coordinates decimal="." cs="," ts=" ">
                        -78.93475229153133,43.92722209162127
                        -78.93539395009117,…
                     </gml:coordinates>
                  </gml:LineString>
               </gb:segment>
            </gb:RoadSegment>
         </wfs:Insert>
      </wfs:Transaction>
   </content>
   <georss:where>
      <gml:Envelope>
         <gml:lowerCorner>-85.3486 10.1183</gml:lowerCorner>
         <gml:upperCorner>-10.8013 80.4831</gml:upperCorner>
      </gml:Envelope>
   </georss:where>
</entry>
```

E.15.3 Response

E.15.3.1 Non-REST response

The following is an example response to a RejectChange operation.

```
<?xml version="1.0" ?>
<gss:TransactionResponse
   service="GSS"
   version="1.0.0"
   xmlns:gss="http://www.opengis.net/gss/1.0"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.opengis.net/gss/1.0
             http://www.pvretano.com/schemas/gss/1.0.0/gss.xsd">
   <gss:EntryId        action="Insert">urn:uuid:        1225c695-cfb8-4ebb-aaaa-
80da344efa6a</gss:EntryId>
</gss:TransactionResponse>
```

E.15.3.2 REST (AtomPub) response

The following is the response to the POST request in D.15.2.

```
HTTP/1.1 201 Created
Date: Fri, 7 Oct 2005 17:17:11 GMT
Content-Length: nnn
Content-Type: application/atom+xml;type=entry;charset="utf-8"
Location:     http://www.acme.com/gss.cgi/REPLICATONFEED/urn:uuid:1225c695-cfb8-
4ebb-aaaa-80da344efa6a

<?xml version="1.0" encoding="UTF-8"?>
```

```
<entry xmlns="http://www.w3.org/2005/Atom"
       xmlns:gml="http://www.opengis.net/gml">
   <title>Insert of feature urn:uuid:551b3fb4-f3cdf530588e</title>
   <id>urn:uuid:1225c695-cfb8-4ebb-aaaa-80da344efa6a</id>
   <updated>2008-12-13T18:30:02Z</updated>
   <author>
      <name>John Doe</name>
      <email>jdoe@example.org</email>
   </author>
   <category scheme="http://www.opengis.org/geosync/actions" term="insert"/>
   <link rel="targetServer" href="http://www.someserver.com/wfs.cgi"/>
   <content type="text/xml; subtype=gml/3.1.1/profiles/gmlsf/1.0.0/0">
      <wfs:Transaction xmlns:wfs="http://www.opengis.org/wfs"
                       Xmlns:gb="http://example.org/roads">
         <wfs:Insert>
            <gb:RoadSegment>
               <gb:cgdiId>urn:uuid:551b3fb4-f3cdf530588e</gb:cgdiId>
               <gb:acquisitionProvider>n.a.</gb:acquisitionProvider>
               <gb:revisionDate>2007-10-24-04:00</gb:revisionDate>
               <gb:nationalRoadClass>Arterial</gb:nationalRoadClass>
               <gb:numberOfLanes>0</gb:numberOfLanes>
               <gb:pavementStatus>C</gb:pavementStatus>
               <gb:pavedRoadSurfaceType>GR</gb:pavedRoadSurfaceType>
               <gb:roadSegmentID>551b3fb4f330588e</gb:roadSegmentID>
               <gb:segment>
                  <gml:LineString
                     srsName="urn:ogc:def:crs:EPSG:6.11.2:4326">
                     <gml:coordinates decimal="." cs="," ts=" ">
                        -78.93475229153133,43.92722209162127
                        -78.93539395009117,…
                     </gml:coordinates>
                  </gml:LineString>
               </gb:segment>
            </gb:RoadSegment>
         </wfs:Insert>
      </wfs:Transaction>
   </content>
   <georss:where>
      <gml:Envelope>
         <gml:lowerCorner>-85.3486 10.1183</gml:lowerCorner>
         <gml:upperCorner>-10.8013 80.4831</gml:upperCorner>
      </gml:Envelope>
   </georss:where>
</entry>
```

Bibliography

1. OGC 07-004, OWS4 GeoDSS Mass Market IPR (Mar 2007)

2. OGC 07-071, Web Notification Service Implementation Specification (May 2007)

3. OpenSearch Geo extension
http://www.opensearch.org/Specifications/OpenSearch/Extensions/Geo/1.0/Draft_1

4. OpenSearch Time extension
http://www.opensearch.org/Specifications/OpenSearch/Extensions/Time/1.0/Draft_1

5. ebXML Registry Binding for REST

Annex F Revision history

Date	Release	Author	Description
10/30/2008	0.0.1	Glenn Stowe	Creation of document
11/25/2008	0.0.2	Peter Vretanos	Major revision to better define subscription and query APIs.
03/30/2010	0.8.0	Peter Vretanos	Yet another major overhaul. Lots of changes most notable is the additions of POX, KVP and REST bindings for all the operations. SOAP is still T.B.D.
05/31/2010	0.9.0	Peter Vretanos	Add conformance classes and a conformance class section to the Capabilities document. Define WSDL for the server and complete SOAP bindings.
06/04/2010	0.9.0	Peter Vretanos	Incorporate comments from OWS7 participants.