Liberty ID-SIS Geolocation Service Specification
Version: 1.0-12

Editors:
Jukka Kainulainen, Nokia Corporation

Contributors:
David Castellanos, Ericsson
Corina Grahm, Ericsson
Rachid Oulahal, France Télécom
Rob Lockhart, IEEE-ISTO

Abstract:
The Liberty ID-SIS Geolocation (ID-SIS-GL) defines a web service. It offers geolocation information regarding a Principal. ID-SIS-GL is an instance of a data oriented identity web service. ID-SIS-GL is using the Liberty ID-WSF Data Services Template and readers of this document should be familiar with that as well as the rest of the Liberty ID-WSF framework. The geolocation related data is mostly from the Mobile Location Protocol version 3.1 specified by the Open Mobile Alliance and readers should be familiar with that also.

Filename: draft-liberty-id-sis-gl-v1.0-12.pdf
Notice

This document has been prepared by Sponsors of the Liberty Alliance. Permission is hereby granted to use the
document solely for the purpose of implementing the Specification. No rights are granted to prepare derivative works
of this Specification. Entities seeking permission to reproduce portions of this document for other uses must contact
the Liberty Alliance to determine whether an appropriate license for such use is available.

Implementation of certain elements of this document may require licenses under third party intellectual property
rights, including without limitation, patent rights. The Sponsors of and any other contributors to the Specification are
not, and shall not be held responsible in any manner for identifying or failing to identify any or all such third party
intellectual property rights. This Specification is provided "AS IS", and no participant in the Liberty Alliance
makes any warranty of any kind, express or implied, including any implied warranties of merchantability,
non-infringement of third party intellectual property rights, and fitness for a particular purpose. Implementors
of this Specification are advised to review the Liberty Alliance Project's website (http://www.projectliberty.org/) for
information concerning any Necessary Claims Disclosure Notices that have been received by the Liberty Alliance
Management Board.

Copyright © 2005 American Express Travel Related Services; Ericsson; France Télécom The International Security,
Trust, and Privacy Alliance; Nippon Telegraph and Telephone Corporation; Nokia Corporation; Sun Microsystems,
Inc.; Symlabs, Inc.; and Vodafone Group Plc. All rights reserved.

Liberty Alliance Project Licensing Administrator
c/o IEEE-ISTO
445 Hoes Lane
Piscataway, NJ 08855-1331, USA
info@projectliberty.org
## Contents

1. Overview ................................................................. 4
2. Service Parameters ...................................................... 6
3. Extensions ................................................................. 9
4. Discovery Option Keywords ............................................. 10
5. SelectType ............................................................... 12
6. TriggerType ............................................................. 18
7. Geolocation Information ............................................... 22
8. Examples ................................................................. 25
9. Schema ................................................................. 27
10. Abstract WSDL .......................................................... 33
References .......................................................................... 34
1. Overview

The Geolocation Service is a Liberty identity service which enables the sharing of the geolocation information of a Principal. The Geolocation Service (ID-SIS-GL) uses the protocols provided by the Liberty ID-WSF Data Services Template [LibertyDST] to carry the geolocation requests and responses. This specification must be read together with [LibertyDST]. This specification defines, normatively, how [LibertyDST] is used in the Geolocation Service and what geolocation information and parameters are supported by the Geolocation Service.

The geolocation information includes the position of a Principal, speed and direction related information, and information related to the quality of the position information.

A Web Service Provider (WSP) hosting an ID-SIS-GL service may support both querying a Principal’s geolocation information and subscribing to a Principal’s geolocation information. In special cases, a WSP may also support modifying a Principal’s geolocation information through the ID-SIS-GL interface. Usually, a WSP obtains the geolocation information by other means than input from a Web Service Consumer (WSC). The information is usually calculated by a network or a device.

For most of the geolocation-specific data and parameters, the definitions in [MLPv3.1] are used. For the corresponding XML elements and attributes, the normative semantic definitions are in [MLPv3.1], which is referred to in relevant places in this document. In this specification for those XML elements and attributes only, an informative description is given about their semantics. If there are any conflicts in those descriptions, the definition from [MLPv3.1] is used. As [MLPv3.1] uses DTD and this specification uses W3C XML schema [Schema1], the [XML] elements and attributes used from [MLPv3.1] are rewritten using the W3C XML Schema language in this specification. This specification provides the normative type definitions for those elements and attributes. Please note that there are some changes in the type definitions when more specific type definitions provided by the XML schema are used.

In addition to this normative specification, a separate implementation guidelines document is available [LibertyIDGLGuide]. It provides more information related to the ID-SIS-GL (e.g., it includes comparison to [MLPv3.1] and has important privacy notes).

This specification has four main parts. First, the use of [LibertyDST] is specified, how different alternatives features are used, and the needed parameters are defined. These define what kind of requests ID-SIS-GL supports. After that, the supported geolocation-specific parameters and data are defined (i.e., what can be received from a WSP hosting an ID-SIS-GL service and what can be sent as possible parameters in requests). After these two parts, some examples are shown and the full XML schema and an abstract WSDL for ID-SIS-GL are presented.

In case of disagreement between this document and any guidelines or XML description, this document is prescriptive. Any published errata are incorporated in this document by reference and, as such, are normative.

1.1. Notation

This specification uses schema documents conforming to W3C XML Schema (see [Schema1]) and normative text to describe the syntax and semantics of XML-encoded protocol messages. Note: Phrases and numbers in brackets [ ] refer to other documents; details of these references may be found at the end of this document.

The key words "MUST," "MUST NOT," "REQUIRED," "SHALL," "SHALL NOT," "SHOULD," "SHOULD NOT," "RECOMMENDED," "MAY," "MAY NOT," and "OPTIONAL" in this specification are to be interpreted as described in [RFC2119]: "they MUST only be used where it is actually required for interoperability or to limit behavior which has potential for causing harm (e.g., limiting retransmissions)."

These keywords are thus capitalized when used to specify, unambiguously, requirements over protocol and application features and behavior that affect the interoperability and security of implementations. When these words are not capitalized, they are meant in their natural-language sense.

The following namespaces are used in the schema definitions:

• The prefix xs: stands for the W3C XML schema namespace (http://www.w3.org/2001/XMLSchema) [Schema1].

• The prefix xml: stands for the W3C XML namespace (http://www.w3.org/XML/1998/namespace) [XML].

Please note that the DST schemas [LibertyDST] included do not define any namespace. Elements from those schemas will be in the ID-SIS-GL namespace when used for the ID-SIS-GL.

This specification uses the following typographical conventions in text: <Element>, <ns:ForeignElement>, attribute, Datatype, and OtherCode.

Definitions for Liberty-specific terms may be found in [LibertyGlossary].

1.2. Time Values

All Liberty time values have the type dateTime, which is built in to the W3C XML Schema Datatypes specification. Liberty time values MUST be expressed in UTC form, indicated by a "Z" immediately following the time portion of the value.

Liberty requesters and responders SHOULD NOT rely on other applications supporting time resolution finer than seconds, as implementations MAY ignore fractional second components specified in timestamp values. Implementations MUST NOT generate time instants that specify leap seconds.
2. Service Parameters

The Geolocation Service is an instance of a data service as described by [LibertyDST] and all stipulations of [LibertyDST] are hereby incorporated unless expressly waived or modified in this specification.

The following tables define how the different features of the Data Services Template are used in the Geolocation Service. The templates for these tables are copied from [LibertyDST].

Table 1. General Service Parameters (1/2)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;ServiceType&gt;</td>
<td>urn:liberty:id-sis-gl:2005-02</td>
</tr>
<tr>
<td>Discovery Options</td>
<td>See Section 4</td>
</tr>
<tr>
<td>Data Schema</td>
<td>For full schema see Section 9</td>
</tr>
<tr>
<td>Semantics of the &lt;Select&gt;</td>
<td>See Section 5.</td>
</tr>
</tbody>
</table>

Table 2. General Service Parameters (2/2)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Element uniqueness</td>
<td>Not applicable. No multiple elements with same name used.</td>
</tr>
<tr>
<td>Data Extension Supported</td>
<td>The &lt;Extension&gt; element MAY be used to add new geolocation-specific data, but the use is not specified by this specification. See Section 3.</td>
</tr>
</tbody>
</table>

Table 3. Query Parameters (1/2)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Support querying</td>
<td>Queries MUST be supported</td>
</tr>
<tr>
<td>Multiple &lt;Query&gt; elements</td>
<td>Multiple &lt;Query&gt; elements MAY be supported.</td>
</tr>
<tr>
<td>Multiple &lt;QueryItem&gt; elements</td>
<td>Multiple &lt;QueryItem&gt; elements MUST NOT be used.</td>
</tr>
<tr>
<td>Support sorting</td>
<td>Not applicable.</td>
</tr>
<tr>
<td>SortType definition</td>
<td>Not applicable as sorting is not supported, empty definition used:</td>
</tr>
<tr>
<td></td>
<td>&lt;xs:complexType name=&quot;SortType&quot;&gt;</td>
</tr>
<tr>
<td></td>
<td><a href="">xs:complexContent</a></td>
</tr>
<tr>
<td></td>
<td>&lt;xs:restriction base=&quot;EmptyType&quot;/&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;/xs:complexContent&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;/xs:complexType&gt;</td>
</tr>
<tr>
<td>Support changedSince</td>
<td>The changedSince attribute MAY be supported when a special need is found for it, but it is RECOMMENDED that it is not used as the geolocation information is very dynamic and rapidly changing and a client should just specify whether it wants current or last geolocation information.</td>
</tr>
<tr>
<td>Supported formats</td>
<td>When the changedSince attribute is used, the whole set of current information the requesting WSC is allowed to get is returned as the set is relatively small and usually all the data changes at the same time.</td>
</tr>
</tbody>
</table>

Table 4. Query Parameters (2/2)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Support includeCommonAttributes</td>
<td>MUST NOT be supported. Only one common attribute used and a WSP decides when to add it.</td>
</tr>
<tr>
<td>Support pagination</td>
<td>Not applicable. MUST NOT be used.</td>
</tr>
<tr>
<td>Support static sets</td>
<td>Not applicable. MUST NOT be used.</td>
</tr>
<tr>
<td>&lt;Extension&gt; in &lt;Query&gt;</td>
<td>The &lt;Extension&gt; element in &lt;Query&gt; MAY be used for new parameters, but the use is not specified by this specification. See Section 3.</td>
</tr>
</tbody>
</table>
### Table 5. Modify Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Support modification</td>
<td>Modifications MAY be supported in special cases. A WSC MUST NOT expect that in a general case.</td>
</tr>
<tr>
<td>Multiple &lt;Modify&gt; elements</td>
<td>Multiple &lt;Modify&gt; elements MAY be supported.</td>
</tr>
<tr>
<td>Multiple &lt;Modification&gt; elements</td>
<td>Multiple &lt;Modification&gt; elements MUST NOT be supported.</td>
</tr>
<tr>
<td>Support partial success</td>
<td>Not applicable as multiple &lt;Modification&gt; elements are not allowed.</td>
</tr>
<tr>
<td>Support notChangedSince</td>
<td>notChangedSince SHOULD NOT be supported.</td>
</tr>
<tr>
<td>&lt;Extension&gt; in &lt;Modify&gt;</td>
<td>The &lt;Extension&gt; element in &lt;Modify&gt; MAY be used to specify new parameters, but the use is not specified by this specification. See Section 3.</td>
</tr>
</tbody>
</table>

### Table 6. Subscribe Parameters (1/2)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Support subscribing to notifications</td>
<td>Subscriptions and notifications SHOULD be supported.</td>
</tr>
<tr>
<td>Use of the &lt;Subscribe&gt; element</td>
<td>If notification subscriptions are supported, modifying existing subscriptions MAY be supported, but canceling and renewing MUST be supported.</td>
</tr>
<tr>
<td>Multiple &lt;Subscribe&gt; elements</td>
<td>Multiple &lt;Subscribe&gt; elements MAY be supported.</td>
</tr>
<tr>
<td>Multiple &lt;Subscription&gt; elements</td>
<td>Multiple &lt;Subscription&gt; elements MAY be supported.</td>
</tr>
<tr>
<td>Use of the &lt;NotifyEndedTo&gt; element</td>
<td>The &lt;NotifyEndedTo&gt; element MUST be supported, if end notifications are used.</td>
</tr>
<tr>
<td>TypeType definition</td>
<td>The &lt;Type&gt; element is not used, so an empty definition is used for it:</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table 7. Subscribe Parameters (2/2)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>TriggerType definition</td>
<td>See Section 6 for the description and the type definition of the &lt;Trigger&gt; element.</td>
</tr>
<tr>
<td>Start of a subscription</td>
<td>If notifications are supported, the starts attribute MUST be supported.</td>
</tr>
<tr>
<td>Subscription expiration</td>
<td>Subscription expiration MUST be used.</td>
</tr>
<tr>
<td>Use of expires and duration attributes</td>
<td>Both expires and duration MUST be supported.</td>
</tr>
<tr>
<td>Support expires==starts</td>
<td>The same value for both the starts and the expires attribute MUST be supported.</td>
</tr>
<tr>
<td>Support zero duration</td>
<td>The value zero MUST be supported for the duration attribute.</td>
</tr>
<tr>
<td>&lt;Extension&gt; in &lt;Subscribe&gt;</td>
<td>The &lt;Extension&gt; element in &lt;Subscribe&gt; MAY be used for new parameters, but the use is not specified by this specification. See Section 3.</td>
</tr>
</tbody>
</table>

### Table 8. QuerySubscriptions Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Support querying existing subscrip-tions</td>
<td>MAY be supported.</td>
</tr>
<tr>
<td>Multiple &lt;QuerySubscriptions&gt; ele- ments</td>
<td>Multiple &lt;QuerySubscriptions&gt; elements MAY be supported.</td>
</tr>
<tr>
<td>&lt;Extension&gt; in &lt;QuerySubscriptions&gt;</td>
<td>The &lt;Extension&gt; element in &lt;QuerySubscriptions&gt; MAY be used for new parameters, but the use is not specified by this specification. See Section 3.</td>
</tr>
</tbody>
</table>
### Table 9. Notify Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Support notifications</td>
<td>Notifications SHOULD be supported.</td>
</tr>
<tr>
<td>Are notifications acknowledged</td>
<td>Notifications SHOULD be acknowledged.</td>
</tr>
<tr>
<td><code>&lt;Extension&gt;</code> in <code>&lt;Notify&gt;</code></td>
<td>The <code>&lt;Extension&gt;</code> element in <code>&lt;Notify&gt;</code> MAY be used to pass additional data, but the use is not specified by this specification. See Section 3.</td>
</tr>
</tbody>
</table>

### Table 10. EndNotify Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Support end notifications</td>
<td>End notifications MAY be supported.</td>
</tr>
<tr>
<td>Are end notifications acknowledged</td>
<td>End notifications SHOULD be acknowledged.</td>
</tr>
<tr>
<td><code>&lt;Extension&gt;</code> in <code>&lt;Ended&gt;</code></td>
<td>The <code>&lt;Extension&gt;</code> element in <code>&lt;Ended&gt;</code> MAY be used to pass additional data, but the use is not specified by this specification. See Section 3.</td>
</tr>
</tbody>
</table>
3. Extensions

The data schema has `<Extension>` elements which permit arbitrary schema extensions under them. An implementation MAY use those `<Extension>` elements to support data not defined in this specification. That new data MUST NOT be in the namespace of the ID-SIS-GL and it MUST be geolocation-related data.

If an implementation supports new data inside `<Extension>` elements, it MAY define new discovery option keywords for the extended data. New discovery option keywords MAY be specified also for other purposes.

There are also `<Extension>` elements inside the parameters for different requests. An implementation MAY use those `<Extension>` elements to support other parameters not defined in this specification. Possible other parameters MUST NOT be in the namespace of the ID-SIS-GL.

Although those `<Extension>` elements exist in the ID-SIS-GL schema, they are not mentioned in this specification when different XML elements are introduced, as the use of those elements is specified in this chapter. The possible content for the `<Extension>` elements SHOULD be specified so that they may also be ignored and such new data or features are then just not available or used. A WSC or a WSP, compliant to this specification, MAY ignore those `<Extension>` elements and the other party MUST be prepared for this.

For a number of elements and attributes, allowed values are enumerated in this specification. New values for those elements and attributes may be defined, see [LibertyReg].
4. Discovery Option Keywords

A number of discovery option keywords are defined to be included as `<Option>` elements in discovery registrations and queries, see [LibertyDisco]. A WSP may choose, which keywords it wants to use, if any, as that is an optional feature, but, when used, each keyword MUST be used only for the purpose for which it has been defined.

4.1. Data Availability Discovery Option Keywords

A data availability keyword describes what geolocation information might be available related to a Principal. A WSP MUST NOT register a discovery option keyword if it does not support the geolocation information to which a keyword refers. However, the geolocation information referred to by a keyword which a WSP has registered at a discovery service may not be available to a requesting WSC due to permissions or to the results of positioning operations. Also, a WSP may not register any keyword even if it supports all geolocation data specified in this document. A WSP MUST NOT register a data availability discovery option keyword if it is very likely that the data is not available.

The table below defines the data availability keywords for ID-SIS-GL. The left column lists the keywords and the right column lists the corresponding XML element names. When a keyword is registered at a discovery service, it means that the corresponding XML element should be available. See later chapters for definitions of the XML elements mentioned in the table.

<table>
<thead>
<tr>
<th>Keyword</th>
<th>The data supported</th>
</tr>
</thead>
<tbody>
<tr>
<td>urn:liberty:id-sis-gl:shape</td>
<td>&lt;shape&gt;</td>
</tr>
<tr>
<td>urn:liberty:id-sis-gl:speed</td>
<td>&lt;speed&gt;</td>
</tr>
<tr>
<td>urn:liberty:id-sis-gl:direction</td>
<td>&lt;direction&gt;</td>
</tr>
<tr>
<td>urn:liberty:id-sis-gl:heading</td>
<td>&lt;Heading&gt;</td>
</tr>
<tr>
<td>urn:liberty:id-sis-gl:alt</td>
<td>&lt;alt&gt;</td>
</tr>
<tr>
<td>urn:liberty:id-sis-gl:civil-data</td>
<td>&lt;CivilData&gt;</td>
</tr>
</tbody>
</table>

4.2. Other ID-SIS-GL Discovery Option Keywords

A position may be indicated using different coordinate reference systems. A WSP MUST support [WGS-84], but it MAY also support other coordinate reference systems, especially the ones used locally in the area a WSP serves. A WSP MAY register keywords indicating which other coordinate reference systems it supports in addition to the [WGS-84]. Those keywords have the format `urn:liberty:id-sis-gl:srs:xxx` where `xxx` is replaced by an identifier for the supported coordinate reference system using the same format as for the srsName attribute (e.g., `urn:liberty:id-sis-gl:srs:www.epsg.org:#4004`). (See [MLPv3.1] for the definition of the srsName attribute.)

Different WSPs may be able to provide the position in different levels of accuracy. A WSP MAY register one of the keywords defined in the table below at a discovery service to indicate what level of accuracy it should be able to reach for the position of a Principal. Please note that the actual accuracy reached at a specific moment varies based on the situation and also through such mechanisms as the use of the `<resp_req>` element in the request (see Section 5.1).

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>urn:liberty:id-sis-gl:accuracy:high</td>
<td>Better than 50 meters</td>
</tr>
<tr>
<td>urn:liberty:id-sis-gl:accuracy:medium</td>
<td>50-500 meters</td>
</tr>
<tr>
<td>urn:liberty:id-sis-gl:accuracy:low</td>
<td>Lower than 500 meters</td>
</tr>
</tbody>
</table>

A WSC will often be interested only in the position of a user relative to some other location (e.g., airport, store, other principal) rather than their actual location. A WSP MAY support these requests for relative locations. A reference area against which the user’s actual location can be compared is added in a request for a relative location and a WSP...
just returns true or false depending on the comparison result (see Section 5.5). If a WSP supports requests for relative location, it MAY register the following discovery option keyword urn:liberty:id-sis-gl:areacomparison. If a WSP supports only relative location queries and no other features like querying actual location, it MAY register the following discovery option keyword urn:liberty:id-sis-gl:onlyareacomparison.

### 4.3. DST Discovery Option Keyword

[LibertyDST] also defines a number of discovery option keywords. The following keywords defined in [LibertyDST] MAY be used by a WSP providing ID-SIS-GL service:

- urn:liberty:dst:extend
- urn:liberty:dst:can:extend
- urn:liberty:dst:multipleResources
- urn:liberty:dst:noModify
- urn:liberty:dst:noSubscribe
- urn:liberty:dst:noQuerySubscriptions
5. SelectType

[LibertyDST] recommends [XPATH] to be used for selecting the data a WSC wants to access. This is, however, not used in the ID-SIS-GL. The ID-SIS-GL also needs other parameters for queries and so a special XML structure is used for the <Select> element. Most of the parameters are from [MLPv3.1]. The <Select> element used in a request may define the geolocation data a WSC wants to have, but it also may define different accuracy and quality aspects a WSC requires from the response.

A <Select> element MAY contain one or more of the following elements: <eqop>, <geo_info>, <loc_type>, <prio>, and <AreaComparison>. Note: a basic query doesn’t need the <Select> at all as by default a WSP returns, as allowed by the privacy rule, all geolocation information it has related to a Principal, with the accuracy it is capable of achieving, and uses WGS-84 as the coordinate reference system.

The <Select> element MUST NOT be used in modifications as a WSC MUST provide all available geolocation data related to a Principal at once and not just modify one part of it.

5.1. <eqop> element

The <eqop> element specifies requested quality for the position information. The <eqop> element is from [MLPv3.1] with some modifications. The <eqop> element has the following elements:

- <ll_acc> defines longitude and latitude accuracy in seconds. For the normative semantic definition, see [MLPv3.1].
- <hor_acc> defines horizontal accuracy in meters and is an alternative to the <ll_acc> element. For the normative semantic definition, see [MLPv3.1].
- <alt_acc> defines altitude accuracy in meters. For the normative semantic definition, see [MLPv3.1].
- <max_loc_type> defines maximum allowed age in seconds for the location information. For the normative semantic definition, see [MLPv3.1].
- <resp_req> defines the response time requirement using type attribute, which has following enumeration values (for the normative semantic definition, see [MLPv3.1]):
  - NO_DELAY, a WSP should return any current location information it has related to a Principal immediately.
  - LOW_DELAY, a WSP should give higher priority to the response time than to the location accuracy.
  - DELAY_TOL, a WSP should give higher priority to the accuracy than to the response time. This is the default value.
All the elements inside an <eqop> element are optional as shown by the schema:

```xml
<xs:element name="eqop" minOccurs="0">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="resp_req" minOccurs="0">
        <xs:complexType>
          <xs:attribute name="type" type="xs:string" default="DELAY_TOL"/>
        </xs:complexType>
      </xs:element>
      <xs:choice minOccurs="0">
        <xs:element name="ll_acc" type="xs:float"/>
        <xs:element name="hor_acc" type="xs:float"/>
      </xs:choice>
      <xs:element ref="alt_acc" minOccurs="0"/>
      <xs:element name="max_loc_age" type="xs:integer" minOccurs="0"/>
      <xs:element ref="Extension" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

If a WSP knows that it can not meet the requirements set by a WSC using any of the elements <ll_acc>, <hor_acc>, <alt_acc>, or <max_loc_age>, it MUST NOT accept the request. For queries, this means that the processing of a <QueryItem> element MUST fail and a second level QopNotAttainable status code SHOULD be used in addition to the top level Failed status code. For subscriptions, this means that a subscription is not accepted if a WSP knows that, most probably, it will not meet the requirement in general. The same QopNotAttainable status code SHOULD be used, in addition to the top level Failed status code, when responding to a subscription request in this kind of a case. If a subscription had been accepted earlier, but a WSP, later, can not meet the requirement with individual positioning, it SHOULD still send a notification when the defined reasons to send a notification are met and it MUST add proper information indicating that the returned position may not meet the requirement (e.g., using the <lev_conf> element). The <resp_req> element does not define a hard requirement. It provides guidance concerning how a WSP SHOULD handle the request.

5.2. <geo_info> element

The <geo_info> element specifies two things: (1) what coordinate reference system should be used and (2) what geolocation information the requestor wants. Note that the default coordinate reference system and the only one which MUST be supported is [WGS-84], when coordinates are used. A <geo_info> element MUST contain one or more of the following elements: <CoordinateReferenceSystem>, <CivilData>, <shape>, <speed>, <alt>, <direction>, and <Heading>.

5.2.1. <CoordinateReferenceSystem> element

The <CoordinateReferenceSystem> element defines the coordinate reference system requested to be used in a response. For the normative semantic definition of the <CoordinateReferenceSystem> element, see [MLPv3.1]. A <CoordinateReferenceSystem> element contains an <Identifier> element, which identifies a coordinate reference system. The <Identifier> element contains three elements identifying the coordinate reference system. The first element is <code>, which is a unique identifier for the coordinate reference system as defined by the authority identified by the content of the <codeSpace> second element, which is also inside the <Identifier> element. The third element inside the <Identifier> element is the <edition> element, which defines the version of the coordinate reference system database managed by the authority defined by the <codeSpace> element.

If the <CoordinateReferenceSystem> element is used in a request and a WSP does not support the requested coordinate reference system, the processing of the request fails and a second level CrsNotSupported status code
should be returned in addition to the top level status code. If there is no `<CoordinateReferenceSystem>` element in a request, a WSP MUST use [WGS-84].

### 5.2.2. Elements requesting specific geolocation information

A `<geo_info>` element may also contain one or more of the elements `<CivilData>`, `<shape>`, `<speed>`, `<alt>`, `<direction>`, and `<heading>` to indicate that a WSC wants to have exactly those elements.

A WSP SHOULD return only the requested geolocation information, but, if it does not support the parameters in requests, it MAY also return all the geolocation information it has related to a Principal and is allowed to share with the requesting WSC, taking into account permissions and privacy policies. If a WSP supports the parameters in requests, but does not have all or part of the requested elements available at a time, it is not, in general, an error condition. That information is just missing from a response or a notification. If a WSP does not support some requested data at all, it SHOULD indicate this with a proper second level status code and return that part of the requested data it has. The status codes `CivilDataNotSupported`, `ShapeNotSupported`, `SpeedNotSupported`, `AltitudeNotSupported`, `DirectionNotSupported`, and `HeadingNotSupported` are available for this purpose. Note that there may be multiple second level status codes in a response.

When any of the elements `<shape>`, `<speed>`, `<alt>`, `<direction>`, or `<Heading>` are used inside a `<geo_info>` element, that element MUST NOT have any content (e.g., if a WSC wants to have the altitude, it SHOULD use `<alt>` inside the `<geo_info>` element or it MAY NOT specify any info in general and just expect to get all the geolocation information available and allowed). The `<CivilData>` element MAY contain other elements indicating which parts of the civil data a WSC wants to have. These subelements are only used to indicate what data a WSC wants and MUST NOT contain anything. For example, if a WSC wants to know in which country and city a Principal is, it may use following:

```xml
<geo_info>
  <CivilData>
    <L/>
  </CivilData>
</geo_info>
```

### 5.2.3. The Schema of the `<geo_info>` Element

```xml
<xs:element name="geoInfo" minOccurs="0">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="CoordinateReferenceSystem" minOccurs="0">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Identifier">
              <xs:complexType>
                <xs:sequence>
                  <xs:element name="code" type="xs:string"/>
                  <xs:element name="codeSpace" type="xs:string"/>
                  <xs:element name="edition" type="xs:string"/>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element ref="CivilData" minOccurs="0"/>
      <xs:element ref="shape" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```
5.3. <loc_type> element

The <loc_type> element specifies the type of geolocation information a WSC wants to have. For the normative semantic definition, see [MLPv3.1]. The <loc_type> element has a type attribute which can have the following values:

- CURRENT, a WSC wants to know the current geolocation information of a Principal. To meet this requirement, a WSP must not use geolocation information it knew before the time it received a request. It must obtain the current location.
- LAST, a WSC wants to know the last geolocation information a WSP has.
- CURRENT_OR_LAST, a WSP should find out the current geolocation information, but, if it can not get it, it may also return the latest information it has.
- INITIAL, this value is related to originating emergency calls and refers to the geolocation information from the time an emergency call was set-up.

The default value is CURRENT. If a WSP can not return the correct type of geolocation information in a response to a query, the processing of that <QueryItem> MUST fail and a second level LocTypeNotAvailable status code SHOULD be used in addition to the top level status code. In a subscription request, the value INITIAL MUST NOT be used. For notification subscriptions, usually, only the default value CURRENT makes sense and other values SHOULD NOT be used.

5.4. <prio> element

This element specifies the priority of the request. For the normative definition, see [MLPv3.1]. The <prio> element has a type attribute, which can have a value of NORMAL or HIGH. The default value is NORMAL. A WSP SHOULD prioritize the requests it is handling concurrently based on the priorities requested. A WSP MAY decide, itself, to whom it wants to provide high priority service. A WSP MAY not support different priorities at all and that is not an error condition, so it MUST process the request otherwise normally but ignoring the <prio> element.

5.5. <AreaComparison> Element

The <AreaComparison> element allows a WSC to ask for the location of a Principal relative to some specified location. The <AreaComparison> element contains a reference area in a form of <CivilData> or <shape>. A WSP compares this reference area to the location of the specified Principal and returns true, if the Principal is within the reference area, false, if the Principal is not within the reference area and unknown, if a WSP is not able to resolve the case. This way, the actual location of the Principal is not revealed as, by default, the location is not returned. The algorithm used to determine whether a Principal is within the specified area or not is out of the scope of this specification.

The true/false/unknown comparison result is returned in a <ComparisonResult> element as the first child element of the <Data> element in a response. A WSC may put more than one <AreaComparison> element in a request. In that case, each <AreaComparison> element MUST have an itemID attribute and each itemID
attribute MUST have a different value compared to other itemID attributes within the same message. In the response message, when multiple <AreaComparison> elements were used in the request, there MUST be one <ComparisonResult> element in the response for each <AreaComparison> element in the request. When there are multiple <ComparisonResult> elements in one response, each of them MUST have an itemIDRef attribute and the value of an itemIDRef attribute MUST be the value of the itemID attribute of the corresponding <AreaComparison> element.

A WSP might not support the format used for defining the reference area, not recognize the given values when <CivilData> is used, the specified reference area might be too small for a WSP to support it, or the Principal’s privacy preferences would preclude their location being pinpointed so exactly. In these cases, the processing of the <AreaComparison> element MUST fail and a suitable second level status code SHOULD be used in addition to the top level status code. If the actual location is requested simultaneously, processing of that should SHOULD NOT fail if the area comparison fails. The following status codes are defined for the mentioned purposes: ShapeNotSupported, SpecifiedShapeNotSupported, SpecifiedCivilDataNotRecognised, and TooSmallArea.

A WSP might not support the <AreaComparison> element at all. In that case, it MUST ignore the whole element, SHOULD use the second level status code AreaComparisonNotSupported in the response, and MUST process the request as a normal location query otherwise.

A normal query without the <AreaComparison> element inside the <Select> element is expected to return normal location data as specified by other parameters and processing rules. With the <AreaComparison> element, it is still possible to request the normal location data, but a WSC MUST set the returnLocation attribute to true inside the <AreaComparison> element to indicate that, in addition to the relative location, it wants to get the actual location, also. If the returnLocation attribute is set to false, the actual location data MUST NOT be returned, even if there are parameters related to a normal query inside the <Select> element. When the returnLocation attribute is set to true, after processing the relative position request, the WSP MUST process the normal location query as the <AreaComparison> element would not exist, e.g., even though for the area comparison a WSC submitted an area as <CivilData>, the location should be returned in the format specified in the <geo_info> parameter. Note that, if no other parameters than the <AreaComparison> element has been provided inside the <Select> element and the returnLocation attribute is set to true, it is also still a valid query for the actual location as all normal parameters are covered by default values and processing rules. Of course, the policies set by the Principal and/or the WSP might not allow returning the actual location and only the relative location information is given, if allowed.

The schema for the <AreaComparison> element:

```xml
<xs:element name="AreaComparison" minOccurs="0" maxOccurs="unbounded">
  <xs:complexType>
    <xs:choice>
      <xs:element ref="CivilData"/>
      <xs:element ref="shape"/>
      <xs:element ref="Extension"/>
    </xs:choice>
    <xs:attribute name="itemID" type="IDType"/>
    <xs:attribute name="returnLocation" type="xs:boolean" default="false"/>
  </xs:complexType>
</xs:element>
```

The schema for the <ComparisonResult> element:

```xml
<xs:simpleType name="ComparisonResultType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="true"/>
    <xs:enumeration value="false"/>
  </xs:restriction>
</xs:simpleType>
```
5.6. Common Error Handling

In addition to the error cases described related to different parameters and mentioned in [LibertyDST], there are some more geolocation-specific cases. The following second level status codes are defined for those cases:

- **AbsentSubscriber**, in case of a network-based positioning, the user (Principal) is currently not reachable.
- **PositionMethodFailure**, a WSP was not able to obtain the position of a Principal.
- **DisallowedByLocalRegulations**, the request is disallowed by local regulatory requirements.

5.7. Modifications

A WSP providing ID-SIS-GL service usually obtains the geolocation information by other means, but in special applications it MAY also support updating the geolocation information through the ID-SIS-GL web service interface it provides. When modifications are done using `<Modify>`, the requesting WSC MUST use only one `<Modification>` element. The `<Select>` element MUST NOT be used and the `<NewData>` element MUST contain all current content for the `<pd>` element. It is not allowed to modify only one part. All available information must be from the same moment of time.
6. TriggerType

There may be several different reasons triggering notifications. A WSC MUST define in a subscription request what should trigger notifications unless basic change notifications are used. This is done using the <Trigger> element in subscription requests. The ID-SIS-GL supports the following types of notifications:

- Notifications are sent when the geolocation information of a Principal changes.
- Notifications are sent periodically.
- Notifications are sent based on an event (e.g., when a mobile device a Principal is using and which is used for positioning a Principal is connected to a network).
- Notifications are sent, when the Principal enters or leaves a specified area.

A WSP may not support notifications at all or it may support only certain type of notifications. Notification may be delayed or not sent at all due to privacy reasons.

6.1. Change Notifications

The change notifications are the basic notifications supported by [LibertyDST]. The <Select> element is used normally to indicate what data a WSC is interested in. A WSC MAY also define the granularity (i.e., how small of a change should trigger a notification). This change is related only to the position (e.g., it is not possible to request notifications when the direction has changed by 90 degrees). The change granularity is specified by the <Granularity> element, which specifies the granularity in meters.

A WSP MAY NOT support the granularity requested by a WSC. In that case the processing of the subscription request MUST fail and the second level RequestedGranularityNotSupported status code SHOULD be used in addition to the top level status code. The minimum granularity supported by a WSP MAY be returned in the comment attribute of the second level <Status> element. When the minimum granularity is returned, the comment attribute MUST contain the text "Minimum granularity XXX meters." where the XXX is replaced with the right value. If a WSC does not put a <Trigger> element inside a subscription request, it is requesting a change notification without specifying any granularity. In this case, a WSP decides the default granularity to be used. A WSP MAY also implement a minimum filtering period during which a new notification is not sent, even if the position change has been big enough.

A WSP MAY support change notifications without supporting the <Granularity> element in subscription requests. If a WSP receives a request containing a <Granularity> element, although it does not support it, the processing of that request MUST fail and a second level GranularityNotSupported status code SHOULD be used in addition to the top level status code. If a WSP does not support change notifications in general and it receives a subscription request for such, the processing of that request MUST fail and the second level ChangeNotificationsNotSupported status code SHOULD be used in addition to the top level status code.

6.2. Periodic Notifications

For periodic notifications, a WSC MUST specify the interval in a subscription request. The interval is defined by an <Interval> element inside a <Trigger> element.

**xs:element name="Interval" type="xs:duration" minOccurs="0"*/
If a WSC specifies a too short interval from the point of view of a WSP, a WSP MAY reject the subscription request. If a WSP rejects a subscription request for this reason it SHOULD use a second level TooShortInterval status code in addition to the top level status code. It MAY also return the minimum interval in the comment attribute of the second level <Status> element. When the minimum interval is returned, the comment attribute MUST contain the text "Minimum interval XXX ." where the XXX is replaced with the minimum value using the format specified for xs:duration.

If a WSP supports notifications, but does not support periodic notifications, the processing of a subscription request MUST fail and a second level PeriodicNotificationsNotSupported status code SHOULD be used in addition to the top level status code.

6.3. Notifications Based On Other Events

A WSP may also support notifications triggered by a defined event. The only event defined at the moment is that of a mobile device becoming available to a network. If a WSC wants to get a geolocation notification when the mobile device a Principal is using becomes available to a network, it MUST add an <ms_event> element inside a <Trigger> element. See [MLPv3.1] for the normative semantic definitions of the <ms_event> element. The <ms_event> element MUST have a type attribute for which only the value MS_AVAIL is defined in this specification as inherited from [MLPv3.1].

A WSC might want to know when a Principal either enters, leaves, or is within a specified area. There is a new element defined for this purpose: <ChangeArea>. The actual area is specified either using a <shape> element, which contains a geometric shape defining an area, or by using a a <CivilData> element, which contains different elements of an address to the level needed to define an area (e.g., country and city). The <ChangeArea> element also contains a mandatory event attribute to specify what kind of an event related to the specified area should trigger a notification. The following values are defined for the event attribute: ENTERING, LEAVING, and WITHIN.
If a WSP does not support area notifications and it receives a subscription request for such, the processing of that subscription request MUST fail and the second level AreaNotificationsNotSupported status code SHOULD be used in addition to the top level status code. A WSP might not support the format used for defining the area, not recognize the given values when `<CivilData>` is used, or the specified area might be too small for a WSP to support it. In these cases, the processing of that subscription request MUST fail and a suitable second level status code SHOULD be used in addition to the top level status code. The following status codes are defined for the mentioned purposes: ShapeNotSupported, SpecifiedShapeNotSupported, CivilDataNotSupported, SpecifiedCivilDataNotRecognised, and TooSmallArea.

Note that a WSP may send an area notification much later than the actual event happened (e.g., it might not have been able to locate a Principal at the time when she entered or left a specified area). In this case, the notification is sent when a WSP becomes aware of that fact. A WSP also may not know about the whole event at all (e.g., a WSC has subscribed to get a notification when a Principal enters certain area, but a WSP may not be able to get the geolocation information of a Principal while the Principal was inside that specified area).

### 6.5. The Schema for the TriggerType

```xml
<xs:complexType name="TriggerType">
  <xs:sequence>
    <xs:element name="Granularity" type="xs:positiveInteger" minOccurs="0"/>
    <xs:element name="Interval" type="xs:duration" minOccurs="0"/>
    <xs:element name="ms_action" minOccurs="0">
      <xs:complexType>
        <xs:attribute name="type" type="xs:string" use="required"/>
      </xs:complexType>
    </xs:element>
    <xs:element name="ChangeArea" minOccurs="0" maxOccurs="unbounded">
      <xs:complexType>
        <xs:choice>
          <xs:element ref="CivilData"/>
          <xs:element ref="shape"/>
          <xs:element ref="Extension"/>
        </xs:choice>
        <xs:attribute name="event" type="xs:string" use="required"/>
      </xs:complexType>
    </xs:element>
    <xs:element ref="Extension" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
```

### 6.6. The reason for a notification

As with one request, a WSC may subscribe to multiple different types of notifications. A notification message must indicate what the reason for the notification was. This is indicated with a `<TriggerReason>` element inside the `<Data>` element of a notification. The following values are defined for the `<TriggerReason>`:

- PERIODIC
- CHANGE
- MS-AVAIL
- ENTERING
A WSP MAY support notification types not defined in this specification and there MAY be other reason codes for those notifications.

The `<TriggerReason>` MUST be before the `<pd>` element inside the `<Data>` element in a notification message.
7. Geolocation Information

Responses and notifications contain geolocation information inside <Data> elements provided by [LibertyDST]. The geolocation information is always inside a <pd> element. Most of the content of the <pd> element are from [MLPv3.1]. A <pd> element has the ACC attribute, which is defined in [LibertyDST]. In ID-SIS-GL, it is used to describe how the geolocation information has been obtained (e.g., is it calculated by a network or a device? entered by a human?). When the geolocation data is calculated by a network or a device (e.g., GPS), the value urn:liberty:dst:acc:challenge SHOULD be used to indicate this. When the data is entered by a human, a WSP does not know whether that data was calculated or entered or it does not want or is not allowed to release that information. The ACC MUST NOT be used or the value urn:liberty:dst:acc:unknown MUST be used in such cases. All the elements inside the <pd> element as well as the ACC attribute are optional as a <pd> element MAY have a different content depending, for example, on the request parameters, access rights and privacy policies, and the results of the positioning methods at each time.

The schema for the <pd> element:

```
<xs:element name="pd">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="time" type="xs:dateTime" minOccurs="0"/>
      <xs:element ref="CivilData" minOccurs="0"/>
      <xs:element ref="shape" minOccurs="0"/>
      <xs:sequence minOccurs="0">
        <xs:element ref="alt" type="xs:float" minOccurs="0"/>
        <xs:element ref="alt_acc" type="xs:float" minOccurs="0"/>
      </xs:sequence>
      <xs:element ref="speed" type="xs:float" minOccurs="0"/>
      <xs:element ref="direction" type="xs:float" minOccurs="0"/>
      <xs:element ref="heading" type="xs:float" minOccurs="0"/>
      <xs:element name="lev_conf" type="xs:float" minOccurs="0"/>
      <xs:element ref="Extension" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute ref="ACC"/>
  </xs:complexType>
</xs:element>
```

7.1. <time> element

The value of the <time> element defines the time when the information contained in a <pd> element was valid (e.g., calculated by a network). See [MLPv3.1] for a normative semantic definition for this element, but please note that the type definition in ID-SIS-GL is different as the xs:dateTime is used. A WSP SHOULD always include a <time> element in a response or a notification. It is especially important when a WSP is not returning the current geolocation information, but the last one it has related to a Principal.

7.2. <CivilData> element

The <CivilData> element contains the position of a Principal in the format of a street address. The same format used by the <Address> element in the [LibertyIDPP] is used as the base with the exception that only those common attributes needed for the localization are used. One new subelement containing the Mobile Network Code is added compared to the <Address> element in the [LibertyIDPP]. See [LibertyIDPP] for the definitions of the other elements inside the <CivilData> element. Please note that the Mobile Network Code values are country-specific, so it SHOULD be used only together with the country code.

```
<xs:element name="CivilData" type="CivilDataType"/>
```
<xs:complexType name="CivilDataType">
  <xs:sequence>
    <xs:element name="PostalAddress" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="LPostalAddress" type="LocalizedString" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="PostalCode" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="L" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="LL" type="LocalizedString" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="St" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="LSt" type="LocalizedString" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="C" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="MNC" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="LocalizedString">
  <xs:simpleContent>
    <xs:extension base="xs:string">
      <xs:attribute ref="xml:lang" use="required"/>
      <xs:attribute ref="script"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>

7.3. <shape> element

The <shape> element contains the position of a Principal as a geometric shape. The shape is defined by a subelement, which is inside a <shape> element. A Principal is somewhere in the area covered by that geometric shape. A WSP chooses which shape to use based on the position method it is using. As many WSCs may not be able to utilize all the different shapes properly, a WSP SHOULD try to return a shape which is easy for a WSC to use for its purposes (e.g., a point). See [MLPv3.1] for the normative semantic definitions as well as for the informative introduction to the different shapes.

<xs:element name="shape">
  <xs:complexType>
    <xs:choice minOccurs="0">
      <xs:element ref="Point"/>
      <xs:element ref="LineString"/>
      <xs:element ref="Polygon"/>
      <xs:element ref="Box"/>
      <xs:element ref="CircularArea"/>
      <xs:element ref="CircularArcArea"/>
      <xs:element ref="EllipticalArea"/>
      <xs:element ref="GeometryCollection"/>
      <xs:element ref="MultiLineString"/>
      <xs:element ref="MultiPoint"/>
      <xs:element ref="MultiPolygon"/>
      <xs:element ref="Extension"/>
    </xs:choice>
  </xs:complexType>
</xs:element>

7.4. <alt> and <alt_acc> elements

The <alt> element is used to indicate the altitude of a Principal in meters. See [MLPv3.1] for normative semantic definitions of the <alt> element. The altitude is in relation to the ellipsoid of the coordinate reference system used.
When an `<alt>` element is included in a response or a notification, the accuracy of that altitude may be described by an `<alt_acc>` element in the same response or notification. The accuracy of an altitude is also reported in meters.

7.5. `<speed>` element

When the speed of a Principal is returned in a response or a notification, it is reported in meters per second. See [MLPv3.1] for normative semantic definitions of the `<speed>` element.

7.6. `<direction>` element

A `<direction>` element indicates the direction towards which a Principal is moving. It is reported in degrees (0-360, 0=360=True North). See [MLPv3.1] for normative semantic definitions of the `<direction>` element.

7.7. `<Heading>` element

The `<heading>` is different than the `<direction>`. The `<heading>` defines to which direction a Principal is facing, so it may be defined also when a Principal is not moving. The `<heading>` is also reported in degrees (0-360, 0=360=True North).

7.8. `<lev_conf>` element

The `<lev_conf>` gives an estimate of the probability that a Principal is located at the area defined by the `<shape>` element returned in the same response or notification. See [MLPv3.1] for normative semantic definitions of the `<lev_conf>` element. When no `<shape>` element is included in a response or notification, but there is the `<CivilData>` element, the `<lev_conf>` element estimates the probability that a Principal is located at the area defined by the `<CivilData>` element.

7.9. Emergency service related elements

In addition to the position data inside a `<pd>` element, a response or a notification message MAY also contain one or two additional elements when a message is sent to an emergency service. These two elements are the Emergency Services Routing Digits `<esrd>` and the Emergency Services Routing Key `<esrk>`. See [MLPv3.1] for the definition of the semantics of these elements. These elements are added directly under the `<Data>` element at the same level as the `<pd>` element, when needed. When present, the order MUST be following: `<esrd>`, `<esrk>`, and `<pd>`. A WSP MUST know, by other means, when to add these elements in a response or a notification message, as request messages do not indicate, specifically, the need for these elements (i.e., a WSP MUST know that a message is related to an emergency service/call).

```xml
<xs:element name="esrd">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="type" type="xs:string"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>

<xs:element name="esrk">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="type" type="xs:string"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
```
8. Examples

The request message to query the location of a user with a number of parameters:

```
  <gl:QueryItem>
    <gl:Select>
      <gl:eqop>
        <gl:resp_req type="LOW_DELAY"/>
        <gl:hor_acc>1000</gl:hor_acc>
      </gl:eqop>
      <gl:loc_type type="CURRENT_OR_LAST"/>
      <gl:prio type="HIGH"/>
    </gl:Select>
  </gl:QueryItem>
</gl:Query>
```

An example response:

```
  <gl:Status code="OK"/>
  <gl:Data>
    <gl:pd>
      <gl:shape>
        <gl:CircularArea>
          <gl:coord>
            <gl:x>30 16 28.312N</gl:x>
            <gl:y>45 15 33.431E</gl:y>
          </gl:coord>
        </gl:CircularArea>
      </gl:shape>
    </gl:pd>
  </gl:Data>
</gl:QueryResponse>
```

Another response example for a case when it was not possible to attain the requested quality of positioning:

```
  <gl:Status code="Failed">
    <gl:Status code="QoPNotAttainable"/>
  </gl:Status>
</gl:QueryResponse>
```

In a basic query, no parameters need to be defined to get all of a Principal’s current geolocation information that is available and allowed. A request message for this may be simply:

```
</gl:Query>
```
In some cases, there is no need to specify the resource (see [LibertyDST]), so the request also may be simply:

```xml
```
9. Schema

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  targetNamespace="urn:liberty:id-sis-gl:2005-02"
  xmlns="urn:liberty:id-sis-gl:2005-02"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified" version="1.0-10">
  <xs:include schemaLocation="liberty-idwsf-dst-v2.0.xsd"/>
  <xs:include schemaLocation="liberty-idwsf-dst-dt-v2.0.xsd"/>
  <xs:complexType name="SortType">
    <xs:complexContent>
      <xs:restriction base="EmptyType"/>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="TypeType">
    <xs:complexContent>
      <xs:restriction base="EmptyType"/>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="SelectType">
    <xs:sequence>
      <xs:element name="AreaComparison" minOccurs="0" maxOccurs="unbounded">
        <xs:complexType>
          <xs:choice>
            <xs:element ref="CivilData"/>
            <xs:element ref="shape"/>
          </xs:choice>
          <xs:attribute name="itemID" type="IDType"/>
          <xs:attribute name="returnLocation" type="xs:boolean" default="false"/>
        </xs:complexType>
      </xs:element>
      <xs:element name="eqop" minOccurs="0">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="resp_req" minOccurs="0">
              <xs:complexType>
                <xs:attribute name="type" type="xs:string" default="DELAY_TOL"/>
              </xs:complexType>
            </xs:element>
            <xs:choice minOccurs="0">
              <xs:element name="ll_acc" type="xs:float"/>
              <xs:element name="hor_acc" type="xs:float"/>
            </xs:choice>
            <xs:element ref="alt_acc" minOccurs="0"/>
            <xs:element name="max_loc_age" type="xs:integer" minOccurs="0"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="geoinfo" minOccurs="0">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="CoordinateReferenceSystem" minOccurs="0">
              <xs:complexType>
                <xs:sequence>
                  <xs:element name="Identifier">
                    <xs:complexType>
                      <xs:sequence>
                        <xs:element name="code" type="xs:string"/>
                        <xs:element name="codeSpace" type="xs:string"/>
                        <xs:element name="edition" type="xs:string"/>
                      </xs:sequence>
                    </xs:complexType>
                  </xs:element>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```
<xs:element name="loc_type" minOccurs="0">
  <xs:complexType>
    <xs:attribute name="type" type="xs:string" default="CURRENT"/>
  </xs:complexType>
</xs:element>

<xs:element name="prio" minOccurs="0">
  <xs:complexType>
    <xs:attribute name="type" type="xs:string" default="NORMAL"/>
  </xs:complexType>
</xs:element>

<xs:element ref="Extension" minOccurs="0"/>
</xs:sequence>
</xs:complexType>

<xs:complexType name="ComparisonResultType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="true"/>
    <xs:enumeration value="false"/>
    <xs:enumeration value="unknown"/>
  </xs:restriction>
</xs:complexType>

<xs:element name="ComparisonResult">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="ComparisonResultType">
        <xs:attribute name="ItemIDRef" type="IDReferenceType"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>

<xs:complexType name="TriggerType">
  <xs:sequence>
    <xs:element name="Granularity" type="xs:positiveInteger" minOccurs="0"/>
    <xs:element name="Interval" type="xs:duration" minOccurs="0"/>
    <xs:element name="ms_action" minOccurs="0">
      <xs:complexType>
        <xs:attribute name="type" type="xs:string" use="required"/>
      </xs:complexType>
    </xs:element>
    <xs:element name="ChangeArea" minOccurs="0" maxOccurs="unbounded">
      <xs:complexType>
        <xs:choice>
          <xs:element ref="CivilData"/>
          <xs:element ref="shape"/>
          <xs:element ref="Extension"/>
        </xs:choice>
        <xs:attribute name="event" type="xs:string" use="required"/>
      </xs:complexType>
    </xs:element>
    <xs:element ref="Extension" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="TriggerReason" type="xs:string"/>
<xs:element name="pd">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="time" type="xs:dateTime" minOccurs="0"/>
      <xs:element ref="CivilData" minOccurs="0"/>
      <xs:element ref="shape" minOccurs="0"/>
      <xs:sequence minOccurs="0">
        <xs:element ref="alt" type="xs:float"/>
        <xs:element ref="alt_acc" type="xs:float" minOccurs="0"/>
      </xs:sequence>
      <xs:element ref="speed" type="xs:float" minOccurs="0"/>
      <xs:element ref="direction" type="xs:float" minOccurs="0"/>
      <xs:element ref="Heading" type="xs:float" minOccurs="0"/>
      <xs:element name="lev_conf" type="xs:float" minOccurs="0"/>
      <xs:element ref="Extension" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute ref="ACC"/>
  </xs:complexType>
</xs:element>
<xs:element name="shape">
  <xs:complexType>
    <xs:choice minOccurs="0">
      <xs:element ref="Point"/>
      <xs:element ref="LineString"/>  
      <xs:element ref="Polygon"/>
      <xs:element ref="Box"/>
      <xs:element ref="CircularArea"/>
      <xs:element ref="CircularArcArea"/>
      <xs:element ref="EllipticalArea"/>
      <xs:element ref="GeometryCollection"/>
      <xs:element ref="MultiLineString"/>
      <xs:element ref="MultiPoint"/>
      <xs:element ref="MultiPolygon"/>
      <xs:element ref="Extension"/>
    </xs:choice>
  </xs:complexType>
</xs:element>
<xs:element name="Box">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="coord"/>
      <xs:element ref="coord"/>
    </xs:sequence>
    <xs:attribute name="gid" type="xs:ID"/>
    <xs:attribute name="srsName" type="xs:string"/>
  </xs:complexType>
</xs:element>
<xs:element name="CircularArcArea">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="coord"/>
      <xs:element ref="inRadius"/>
      <xs:element ref="outRadius"/>
      <xs:element ref="startAngle"/>
      <xs:element ref="stopAngle"/>
      <xs:element ref="angularUnit" minOccurs="0"/>
      <xs:element ref="distanceUnit" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="gid" type="xs:ID"/>
    <xs:attribute name="srsName" type="xs:string"/>
  </xs:complexType>
</xs:element>
<xs:element name="CircularArea">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="coord"/>
      <xs:element ref="inRadius"/>
      <xs:element ref="outRadius"/>
      <xs:element ref="startAngle"/>
      <xs:element ref="stopAngle"/>
      <xs:element ref="angularUnit" minOccurs="0"/>
      <xs:element ref="distanceUnit" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="gid" type="xs:ID"/>
    <xs:attribute name="srsName" type="xs:string"/>
  </xs:complexType>
</xs:element>
<xs:element ref="radius"/>
<xs:element ref="distanceUnit" minOccurs="0"/>
</xs:sequence>
<xs:attribute name="gid" type="xs:ID"/>
<xs:attribute name="srsName" type="xs:string"/>
</xs:complexType>
</xs:element>
<xs:element name="EllipticalArea">
<xs:complexType>
<xs:sequence>
<xs:element ref="coord"/>
<xs:element ref="angle"/>
<xs:element ref="semiMajor"/>
<xs:element ref="semiMinor"/>
<xs:element ref="angularUnit"/>
<xs:element ref="distanceUnit" minOccurs="0"/>
</xs:sequence>
<xs:attribute name="gid" type="xs:ID"/>
<xs:attribute name="srsName" type="xs:string"/>
</xs:complexType>
</xs:element>
<xs:element name="GeometryCollection">
<xs:complexType>
<xs:sequence>
<xs:element ref="shape" maxOccurs="unbounded"/>
</xs:sequence>
<xs:attribute name="gid" type="xs:ID"/>
<xs:attribute name="srsName" type="xs:string"/>
</xs:complexType>
</xs:element>
<xs:element name="LineString">
<xs:complexType>
<xs:sequence>
<xs:element ref="coord"/>  
<xs:element ref="coord" maxOccurs="unbounded"/>
</xs:sequence>
<xs:attribute name="gid" type="xs:ID"/>
<xs:attribute name="srsName" type="xs:string"/>
</xs:complexType>
</xs:element>
<xs:element name="LinearRing">
<xs:complexType>
<xs:sequence>
<xs:element ref="coord" minOccurs="3" maxOccurs="unbounded"/>
</xs:sequence>
<xs:attribute name="gid" type="xs:ID"/>
<xs:attribute name="srsName" type="xs:string"/>
</xs:complexType>
</xs:element>
<xs:element name="MultiLineString">
<xs:complexType>
<xs:sequence>
<xs:element ref="LineString" maxOccurs="unbounded"/>
</xs:sequence>
<xs:attribute name="gid" type="xs:ID"/>
<xs:attribute name="srsName" type="xs:string"/>
</xs:complexType>
</xs:element>
<xs:element name="MultiPoint">
<xs:complexType>
<xs:sequence>
<xs:element ref="Point" maxOccurs="unbounded"/>
</xs:sequence>
<xs:attribute name="gid" type="xs:ID"/>
<xs:attribute name="srsName" type="xs:string"/>
</xs:complexType>
</xs:element>
<xs:element name="MultiPolygon">
  <xs:complexType>
    <xs:choice maxOccurs="unbounded">
      <xs:element ref="Polygon"/>
      <xs:element ref="Box"/>
      <xs:element ref="CircularArea"/>
      <xs:element ref="CircularArcArea"/>
      <xs:element ref="EllipticalArea"/>
    </xs:choice>
    <xs:attribute name="gid" type="xs:ID"/>
    <xs:attribute name="srsName" type="xs:string"/>
  </xs:complexType>
</xs:element>

<xs:element name="Point">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="coord"/>
    </xs:sequence>
    <xs:attribute name="gid" type="xs:ID"/>
    <xs:attribute name="srsName" type="xs:string"/>
  </xs:complexType>
</xs:element>

<xs:element name="Polygon">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="outerBoundaryIs"/>
      <xs:element ref="innerBoundaryIs" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="gid" type="xs:ID"/>
    <xs:attribute name="srsName" type="xs:string"/>
  </xs:complexType>
</xs:element>

<xs:element name="X" type="xs:string"/>
<xs:element name="Y" type="xs:string"/>
<xs:element name="Z" type="xs:string"/>
<xs:element name="angle" type="xs:string"/>
<xs:element name="angularUnit" type="xs:string"/>
<xs:element name="coord">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="X"/>
      <xs:element ref="Y" minOccurs="0"/>
      <xs:element ref="Z" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="distanceUnit" type="xs:string"/>
<xs:element name="inRadius" type="xs:string"/>
<xs:element name="innerBoundaryIs">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="LinearRing"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="outRadius" type="xs:string"/>
<xs:element name="outerBoundaryIs">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="LinearRing"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="radius" type="xs:string"/>
<xs:element name="semiMajor" type="xs:string"/>
<xs:element name="semiMinor" type="xs:string"/>
<xs:element name="startAngle" type="xs:string"/>
<xs:element name="stopAngle" type="xs:string"/>
<xs:element name="alt" type="xs:float"/>
<xs:element name="alt_acc" type="xs:float"/>
<xs:element name="speed" type="xs:float"/>
<xs:element name="direction" type="xs:float"/>
<xs:element name="Heading" type="xs:float"/>
<xs:element name="CivilData" type="CivilDataType"/>
<xs:complexType name="CivilDataType">
  <xs:sequence>
    <xs:element name="PostalAddress" type="xs:string" minOccurs="0"/>
    <xs:element name="LPostalAddress" type="LocalizedString" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="PostalCode" type="xs:string" minOccurs="0"/>
    <xs:element name="L" type="xs:string" minOccurs="0"/>
    <xs:element name="LL" type="LocalizedString" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="St" type="xs:string" minOccurs="0"/>
    <xs:element name="LSt" type="LocalizedString" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="C" type="xs:string" minOccurs="0"/>
    <xs:element name="MNC" type="xs:string" minOccurs="0"/>
    <xs:element ref="Extension" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="LocalizedString">
  <xs:simpleContent>
    <xs:extension base="xs:string">
      <xs:attribute ref="xml:lang" use="required"/>
      <xs:attribute ref="script"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
<xs:element name="esrd">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="type" type="xs:string"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
<xs:element name="esrk">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="type" type="xs:string"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
10. Abstract WSDL

<?xml version="1.0" encoding="UTF-8"?>
<!-- Abstract WSDL for Liberty ID-SIS Geolocation Interface Specification -->
<definitions name="id-sis-gl_2005-02_interface"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:gl="urn:liberty:id-sis-gl:2005-02">
    <types>
        <xsd:schema>
            <xsd:import schemaLocation="liberty-idsis-gl.xsd"/>
        </xsd:schema>
    </types>
    <message name="Query">
        <part name="body" element="gl:Query"/>
    </message>
    <message name="QueryResponse">
        <part name="body" element="gl:QueryResponse"/>
    </message>
    <message name="Modify">
        <part name="body" element="gl:Modify"/>
    </message>
    <message name="ModifyResponse">
        <part name="body" element="gl:ModifyResponse"/>
    </message>
    <message name="Subscribe">
        <part name="body" element="gl:Subscribe"/>
    </message>
    <message name="SubscribeResponse">
        <part name="body" element="gl:SubscribeResponse"/>
    </message>
    <message name="QuerySubscriptions">
        <part name="body" element="gl:QuerySubscriptions"/>
    </message>
    <message name="Subscriptions">
        <part name="body" element="gl:Subscriptions"/>
    </message>
    <portType name="GeolocPort">
        <operation name="GeolocQuery">
            <input message="typ:Query"/>
            <output message="typ:QueryResponse"/>
        </operation>
        <operation name="GeolocModify">
            <input message="typ:Modify"/>
            <output message="typ:ModifyResponse"/>
        </operation>
        <operation name="GeolocSubscribe">
            <input message="typ:Subscribe"/>
            <output message="typ:SubscribeResponse"/>
        </operation>
        <operation name="GeolocQuerySubscriptions">
            <input message="typ:QuerySubscriptions"/>
            <output message="typ:Subscriptions"/>
        </operation>
    </portType>
</definitions>
References

Normative


Informative