Liberty ID-SIS Geolocation Service Implementation Guidelines

Version: 1.0-15

Editors:
Corina Grahm, Ericsson
David Castellanos, Ericsson
Jukka Kainulainen, Nokia

Contributors:
Paul Madsen, Entrust
Rachid Oulahal, France Télécom
Rob Lockhart, IEEE-ISTO

Abstract:

This document provides implementation guidelines supplemental to the Liberty ID-SIS Geolocation (ID-SIS-GL) Service Specification.

The reader is expected to be familiar with the Liberty ID-WSF Web Services Framework Overview, XML, SAML and SOAP. The Liberty ID-SIS-GL is a web service hosted by an application provider and usually discovered via a discovery service.

ID-SIS-GL offers geolocation information including the position of a Principal, speed and direction related information and information related to the quality of the data provided. ID-SIS-GL may also provide geolocation information in a more human readable format (e.g., street, city, region, country).

An ID-SIS-GL service is an instance of a data oriented (see ID-WSF Data Services Template) identity web service (see ID Web Services Framework). An ID-SIS-GL service, like all data services, is characterized by the ability to query and update attribute data as well as the ability to subscribe to receive notifications of location information updates. It relies on mechanisms from other specifications for access control and for conveying data validation information and usage directives.

Filename: draft-liberty-id-sis-gl-guidelines-v1.0-15.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.

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

Contents

1. Liberty ID-SIS Geolocation Service

Liberty ID-SIS Geolocation (ID-SIS-GL) defines a web service that offers geolocation information regarding a Principal. ID-SIS-GL is an instance of a data oriented identity web service using the Liberty ID-WSF Data Services Template [LibertyDST] and rest of the Liberty ID-WSF framework. The geolocation related data used in ID-SIS-GL is mostly adopted from the Mobile Location Protocol specified by the Open Mobile Alliance.

This document provides a rationale and guidance for implementers of the ID-SIS-GL. A companion document, Liberty ID-SIS Geolocation Service Technical Specification [LibertyGL], normatively describes the ID-SIS-GL.

If there is disagreement between present document and [LibertyGL], the Specification is prescriptive.

1.1. Document Audience

This document is intended for application developers and implementers. The reader is presumed to be familiar with XML, SAML, SOAP, and WSDL. The reader should be familiar, as well, with the Liberty ID-FF Architectural Overview [LibertyIDFFOverview] and the Liberty ID-WSF Web Services Framework Overview [LibertyIDWSFOverview].

Apart from this implementation guidelines document, readers and implementers of the [LibertyGL] specification will also benefit from the information contained in the following documents:

- Liberty ID-WSF Implementation Guidelines, [LibertyIDWSFGuide].
- Liberty ID-WSF Security and Privacy Overview, [LibertyIDWSFSecurityPrivacyGuidelines].
- Privacy and Security Best Practices, [LibertyPrivacy].

1.2. Architectural Context of the ID-SIS-GL

ID-SIS-GL service is an instance of a data-oriented identity service. The data-oriented aspect means that the service intends to provide attribute data structured in logical containers. This approach is used by other Liberty services as they share the methods and general framework as described in [LibertyDST].

The identity services in general require that Principal is directly or abstractly present in all transactions involving his identity or data, e.g., data that the Principal has gathered about other people. Thus the services that consult the ID-SIS-GL service use Liberty architectural framework to prove that they are acting on behalf of the Principal or that the Principal has somehow consented to sharing the data, for example by means of a standing order or subscription. The identity services are further described in [LibertyIDWSFOverview].

1.2.1. ID-SIS-GL as an Interface

Although the essence of the ID-SIS-GL service is attributes expressed as data, it should be understood that the technical implementation is actually a process, which handles data requests and computes responses. The specification defines a data interface to a geolocation service; no particular implementation is mandated. The specification can be considered to provide a "dictionary" of data and parameter fields, the specific fields used determined by the implementations and circumstances. The fact that the services are dynamic allows many powerful features such as flexible permission enforcement and supplying different responses to different service providers sending same requests, e.g., some data may not be provided to all service providers or some quality of positioning is not supported for all service providers.

1.2.2. Participants

The ID-SIS-GL is provided by an *attribute provider* (AP) [LibertyIDWSFGuide], sometimes referred to as a Location Service Provider (LSP) in this document. The LSP is an ID-WSF web service that hosts the ID-SIS-GL. The ID-SIS-GL is queried or updated by a *client*, which is usually a *service provider* (SP) [LibertyIDFFOverview] acting on behalf of the *Principal* (also known as the *Target*) [LibertyIDWSFGuide]. The client is sometimes referred to as a *web services client* (WSC) or as a Location Service Client (LSC). The [LibertyIDWSFGuide] describes the means by which the Principal can delegate to the LSC a right to invoke her ID-SIS-GL service, i.e., a service assertion. Before the LSC can access the ID-SIS-GL, it usually (but not necessarily) has to *discover* which AP hosts the ID-SIS-GL for the Principal. This is accomplished using a *discovery service* (DS) [LibertyDisco] that issues the service assertions.

The Target Principal will often be a human individual and normally will be specifying the privacy policies for her/himself. However other entities may specify additional policies to be met acting also as *Policy Owners*

The basic Use Case ID-SIS-GL is covering, implies that the entity for which location data is determined (Target) and the entity on whose behalf the LSC will request the Target's location data are the same Principal (*Requestor*).

1.3. Overview of Liberty ID-SIS Geolocation

Before discussing Implementation Guidelines for Liberty ID-SIS Geolocation Service, a typical scenario is presented in Figure 1

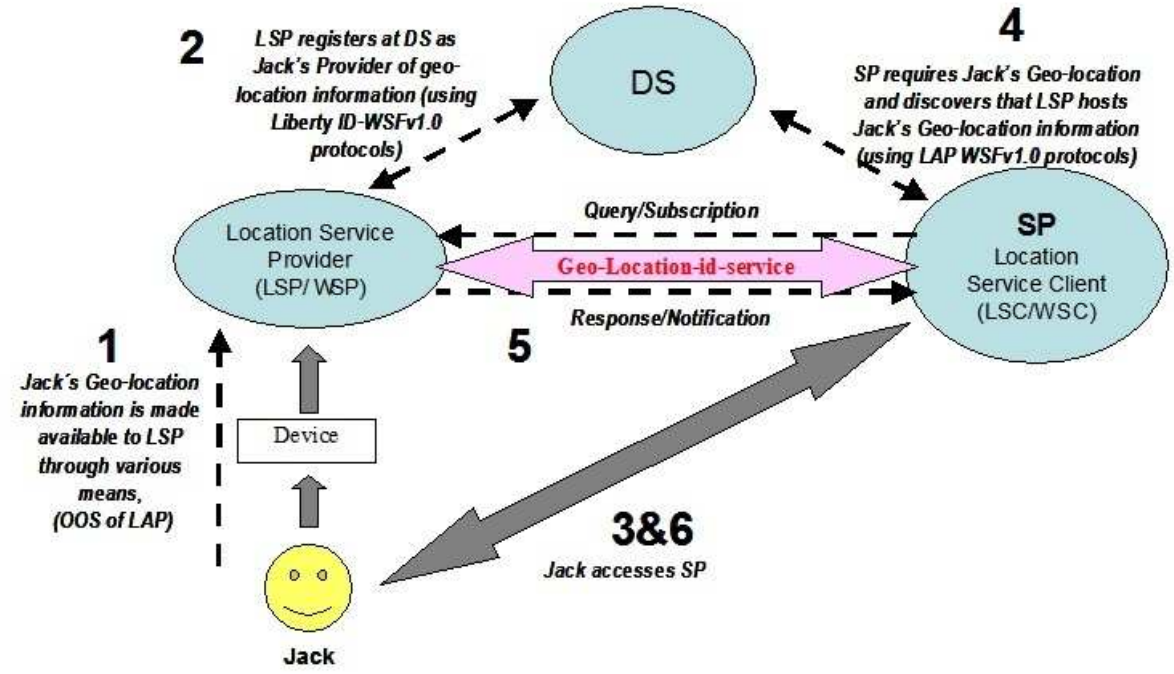

Figure 1. Liberty ID-SIS Geolocation Scenario

1. Jack's (Principal's) geolocation information is made available to a Location Service Provider (LSP) through various means. Jack selects LSP as the Provider of his Geolocation information and also at this time Jack would specify his privacy policy for access to his geolocation data.
   There are a multitude of mechanisms by which a Location Service Provider can determine the location of a principal (e.g., IP address, Mobile positioning, GPS). Use of such mechanisms or definition of new ones is, however, out of the scope of Liberty ID-SIS Geolocation Service.

2. LSP registers at Jack's Discovery Service (DS) so that future Location Service Clients (LSC) will be able to determine which LSP Jack is using.
   N.B. LSPs perform this step just once.

3. Later on, Jack accesses a Service Provider (SP) in order to make use of some location-based service. Single Sign On mechanisms available from Liberty Alliance Project might be leveraged by the SP for Jack's identification and authentication purposes.

4. SP requires Jack's geolocation information in order to deliver the service (or a more personalized service) to Jack and, acting as a LSC (WSC), queries the DS looking for Jack's LSP.

5. After the DS returns information of the LSP, the LSC can query Jack's location. LSC would then be able to query and even subscribe to Jack's geolocation information according to Liberty ID-SIS Geolocation service as specified in [LibertyGL].

6. Finally, SP (LSC/WSC), after processing Jack's geolocation information, would be able to deliver the service to Jack.

2. OMA Mobile Location Protocol

As [LibertyGL] shows the Liberty ID-SIS Geolocation service is very close to the Mobile Location Protocol specified by the Open Mobile Alliance [MLPv3.1]. This makes it relatively straightforward to implement a web service interface according to [LibertyGL] to location servers already supporting [MLPv3.1]. This chapter discusses the similarities and differences between [LibertyGL] and [MLPv3.1]. Some issues already discussed in [LibertyGL] are not repeated here.

2.1. Transport Protocol

[MLPv3.1] defines mapping for HTTP, how XML content specified in [MLPv3.1] is transported using HTTP. [LibertyGL] is based on the Liberty Identity Web Services Framework (ID-WSF) [LibertyIDWSFOverview] and uses SOAP on top of HTTP. Logically the difference is relatively small as [MLPv3.1] already defines the header and the body for messages and those map logically quite straightforward to the SOAP header and body.

2.2. Header

[MLPv3.1] defines own message header. [LibertyGL] doesn't define any message header as header blocks are defined by a number of specifications of the Liberty ID-WSF. Equivalent functionality provided by the [MLPv3.1] headers is also provided by the Liberty ID-WSF. This chapter describes how similar functionality can be achieved using Liberty ID-WSF.

2.2.1. OMA-MLP <requestor>, <client>, and <subclient> Header Elements

The `requestor` element of [MLPv3.1] indicates the initiator of the location request, so in this context besides a Service Provider it could also be an MS subscriber who is asking the position of another target MS. The identity of the `requestor` may be an MSISDN or any other identifier identifying the initiator of the location request.

The `subclient` elements (if present) of [MLPv3.1] identify the Service Providers, resellers, and portals in the chain of service providers between the network and the principal. The distinction between `client` and `subclient` elements is that the `client` element identifies the provider of the service that the Location Server has the initial relationship with, whereas the `subclient` elements identify the chain of other service providers up to the principal. The final service provider in the chain is identified as such (`last_client="YES"`).

In the scope of Liberty ID-WSF, the chain of entities involved in a location requests towards the Location Server is represented using:

- Liberty ID-WSF `<Provider>` header block as defined by [LibertySOAPBinding]. This header block provides a means for a sender to claim that it is represented by a given `providerID` value.
- `<ProxySubject>` and `<ProxyTransitedStatement>` elements within the `<saml:Assertion>` element of the Liberty ID-WSF `<wsse:Security>` header block as defined by [LibertySecMech]. These elements are used to identify the entities (if any) which actively participated in the message exchanges leading up to a given resource access.
- Liberty ID-WSF `<ResourceAccessStatement>` element within the `<saml:Assertion>` element of the Liberty ID-WSF `<wsse:Security>` header block as defined by [LibertySecMech]. The purpose of this statement is to convey sufficient information regarding the accessing entity and the resource for which access is being attempted.

2.2.2. OMA-MLP <client> and <sessionid> Header Elements

In turn, the `client` element of [MLPv3.1] can be comprised of `id`, `pwd`, `serviceid`, and `requestmode` elements.

- In a request, `id` and `pwd` represent the identity and password of the registered user performing a location request. In an answer, they represent the name and password of a location server.
  The header element `sessionid` of [MLPv3.1] is used to represent the current session between the Location Service Provider and the Location Service Client and normally it is also used to replace the `id` and `pwd` elements in subsequent requests, i.e., a WSC is authenticated normally for the first request and, as part of the response, the `sessionid` is returned. If the Location Service Provider does not return a `sessionid`, the Location Service Client shall continue to "login" for subsequent transactions. The Location Service Client may ignore the `sessionid`, if desired, and continue to "login" using `id` and `pwd` elements for subsequent transactions.
  Liberty ID-WSF, as defined by [LibertySecMech], specifies more advanced identification and authentication mechanisms that Location Service Clients and Providers may use in these cases. Additionally, the `<ServiceInstanceUpdate>` header block defined by [LibertySOAPBinding] returns a `<wsse:BinarySecurityToken>` that could be used as credentials in subsequent request(s). (See [LibertyIDWSFGuide] for examples.)

- `Serviceid` specifies an `id` that is used by an entity to identify the service or application that is accessing the network. A typical use of this element is to provide further information on the nature and purpose of the location service being used (e.g., Navigation). Liberty ID-WSF `<UsageDirectives>` header block defined by [LibertySOAPBinding] specifies a container that could include related information. Refer to Section 3.2.2, which includes an example showing how this information could be conveyed.

- `requestmode` indicates whether the request has been initiated by the end-user. Similar indications can be inferred looking into Liberty ID-WSF `<Consent>` and `<ProcessingContext>` header blocks.
  `<Consent>` is used to explicitly claim whether the Principal consented to the present interaction.
  `<ProcessingContext>` may be employed by a sender to signal to a receiver that the latter should add a specific additional facet to the overall processing context in which any action(s) is invoked as a result of processing any `ID-*` message also conveyed in the overall SOAP-bound `ID-*` message. [LibertySOAPBinding] defines three processing context facet URIs including one for situations when Principal is online and another one for offline situations.

2.3. Body

The body in [MLPv3.1] is either one of the request (e.g., `Standard Location Immediate Request` or `Triggered Location Request`) or a response or report (e.g., `Standard Location Immediate Answer` or `Triggered Location Report`). In [LibertyGL], the body is a SOAP body containing a message specified by [LibertyGL]. The table below shows how the message bodies map between [MLPv3.1] and [LibertyGL].

172 **Table 1. Message Body Mapping between [MLPv3.1] and [LibertyGL]**

OMA Mobile Location Protocol[MLPv3.1]	*Liberty ID-SIS Geolocation*[LibertyGL]
`Standard Location Immediate Request (SLIR)`	`Query` (or `Subscribe`, when asynchronous service is requested)
`Standard Location Immediate Answer (SLIA)`	`QueryResponse` (or `SubscribeResponse` when asynchronous service was requested)
`Standard Location Immediate Report (SLIR)`	`Notify`
`Emergency Location Immediate Request (EME_LIR)`	`Query`
`Emergency Location Immediate Answer (EME_LIA)`	`QueryResponse`
`Standard Location Report (SLREP)`	`Notify`
`Emergency Location Report (EMEREP)`	`Notify`
`Triggered Location Reporting Request (TLRR)`	`Subscribe`
`Triggered Location Reporting Answer (TLRA)`	`SubscribeResponse`
`Triggered Location Report (TLREP)`	`Notify`
`Triggered Location Reporting Stop Request (TLRSR)`	`Subscribe`
`Triggered Location Reporting Stop Answer (TLRSA)`	`SubscribeResponse`

173 The asynchronous services `<SLIR res_type="ASYNC">` and `<SLIA res_type="ASYNC">` of [MLPv3.1] maps
174 to subscriptions/notifications used by [LibertyGL] when the `duration` for the subscription equals to `zero` and
175 `returnCurrentValues` is set to `False` in the subscription request.

176 [LibertyGL] doesn't specify any emergency specific messages, but uses the same messages from [LibertyDST] for both
177 `Standard` and `Emergency Location Immediate services`. The only difference in contents is that `EME_LIA`
178 contains two more optional elements `<esrd>` and `<esrk>` compared to `SLIA`. When a WSP authenticates a WSC
179 sending a `<Query>` and notices that the request came from an emergency service, it knows to add those elements,
180 when applicable.

181 In the same way as in [MLPv3.1] for `Standard Location Report`, the needed "request" parameters including the
182 endpoint to which the notifications should be sent must be specified out-of-band as no request message has been sent to
183 get these reports. Similar as for the case of `Emergency Location Report`, but here the Location Service Provider
184 must also know to add elements `<esrd>` and `<esrk>` when applicable.

185 Inside the body, the parameters and the data are mostly the same, but there are some differences. Different type
186 definitions and structures are used for some data and [LibertyGL] defines some new parameters and data.

187 The example below shows how basic querying differs between [MLPv3.1] and [LibertyGL].

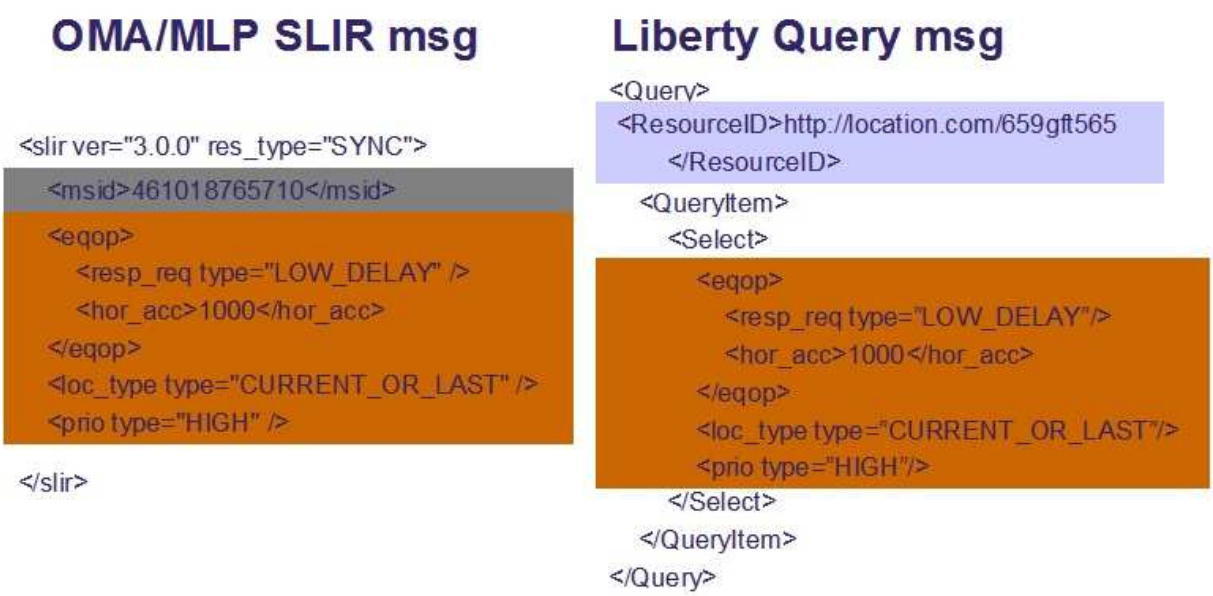

Figure 2. `Query` Message Mapping between [MLPv3.1] and [LibertyGL]

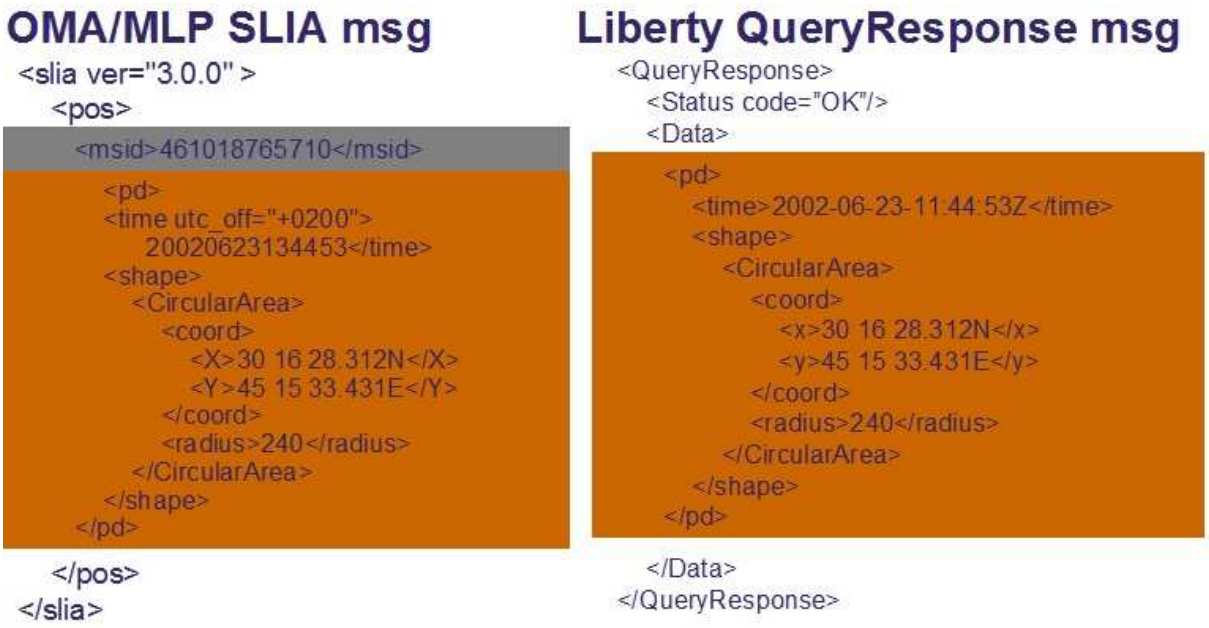

Figure 3. `QueryResponse` Message Mapping between [MLPv3.1] and [LibertyGL]

2.3.1. <ResourceID>

The `msid` element of [MLPv3.1] represents the identifier of a Mobile Subscriber being located. `msid` will normally come in the form of `MSISDN` (Mobile Subscriber ISDN Number as the default value) which is a unique and global identifier for a user, commonly used in GSM (Global System for Mobile communications) mobile phone networks.

In a scenario where multiple parties will provide services to an individual, it is not necessarily desirable (from a user privacy perspective) for a single, global identifier to be tied so closely to that individual, as multiple providers may then collude to determine the identity of the user.

To mitigate such a possibility, Liberty uses a `<ResourceID>` or `<EncryptedResourceID>` to identify the identity-based resource being accessed as shown above. `<ResourceID>` is used to provide privacy-protecting qualities in this case. Such an identifier may also be encrypted for transmission by some third-party, to prevent the third-party from being aware of the actual identifier value.

The type definitions for `<ResourceID>` and `<EncryptedResourceID>` elements are imported from the Liberty ID-WSF Discovery Service schema. For more information about resources, different types of resource identifiers, and encryption of resource identifiers, see [LibertyDisco].

[MLPv3.1] also offers the possibility to query for location information of a range of `MSISDN`s in one single location request. `msid`s element would be used in these cases. [LibertyDST] also offers the possibility of requesting location information of more than one principal by inserting multiple `<Query>` elements in the same request message. Implementers shall, however, assess performance impacts of this practice since potentially each `<Query>` element may have associated a different security assertion that will have to be analyzed and validated before being able to issue any response.

[LibertyDisco] defines mechanisms that facilitate discovery and invocation of resource offerings. This specification also shows how entities which authenticate principals using SAML (e.g. a Liberty ID-FF Identity Provider), may provide a Service Provider with the contact information of the discovery service containing identity services for the authenticated principal. Normatively with this mechanism, an SP acting on behalf of a particular principal will be only able to discover resource offerings of that particular principal (i.e. the principal requesting location information and the target principal being located are actually the same principal as depicted in the use case example in figure 1 above). Non-standard ways of obtaining Discovery Service contact information and resource offerings shall be employed for scenarios where the principal originating the location request is different from the target principal (e.g., "friend finder" location services).

In Liberty ID-WSF and when the principal initiator of the request is actually the resource owner, the Location Service Provider performs authorization of location requests based primarily on the identity of the requesting Location Service Client. Other authorization decision may imply the fact that the requesting principal has an open session with the Location Service Client or not. However, there is no actual verification of the identity of the actual user (if any) behind the requesting Location Service Client as it is assumed that it will actually be the owner of the resource (location information) being accessed. Additional mechanisms, specified by neither ID-WSF nor [LibertyGL], shall be employed to authorize location requests when the principal originating the location request is different from the target principal.

For example, the `codeword` element of [MLPv3.1] is an access code defined per `msid`. This code is used to protect location information of a mobile station against unwanted location requests. Only location requests with the correct codeword of a target `msid` are accepted. Similar techniques could be used as an alternative to authorize location requests when the principal originating the location request is different from the target principal. Relevant information optionally could be accommodated in the ID-SIS-GL schema, making use of the extension mechanisms defined by [LibertyGL]. However, the particularization of this mechanism to make it possible to be used within the Liberty ID-SIS-GL scenarios is out of the scope of Liberty ID-SIS-GL.

2.3.2. Additional Operations

[LibertyGL] provides some additional features compared to [MLPv3.1].

- Existing subscriptions can be queried and modified.

- Notifications can be acknowledged and separate end notifications can be used to indicate that a subscription is not valid anymore.

- The acknowledgement of a subscription request may also return the current location. In special applications, the geolocation information of a Principal can be modified through the provided web service interface.

- In addition to the coordinate format used by [MLPv3.1], [LibertyGL] also offers the possibility to return the position of a principal in the format of a street address (based on the `<Address>` element of [LibertyIDPP]).

- In addition to the possibility to subscribe to area-based notifications also provided by [MLPv3.1], [LibertyGL] offers the possibility for location requests to include a reference area against which the user's actual location can be compared. A value of `true` or `false` is returned depending on the comparison result.

These features are optional and [LibertyGL] can be implemented in a way that only features mapping to [MLPv3.1] features are implemented. Please note that [LibertyGL] can also be implemented without supporting all the features mapping to [MLPv3.1] features.

2.4. Result Codes

The result codes are reported in a different way in [MLPv3.1] and [LibertyGL]. [MLPv3.1] specifies `<result>` element, which is returned, when no data is returned, e.g., when `SLIA` returns the requested position information, no `<result>` element is included. Together with the `<result>` element an optional `<add_info>` element may also be used to provide more specific information, e.g., which element caused the problem. [LibertyGL] uses the status report specified in [LibertySOAPBinding] and [LibertyDST]. For errors in headers and major message faults an ID-* Fault message is returned [LibertySOAPBinding]. For return status of the body part is provided as specified by [LibertyDST] with some additional geolocation specific status codes defined in [LibertyGL]. Each response message contains a status code, even successful response including position data. In addition to the top level status code there can be one or more second level status codes giving more specific information as [LibertyDST] defines only three top level status codes `OK`, `Failed`, and `Partial`. The use of the more detailed second level status codes is optional, unless the top-level status code `Partial` is used. For more details, see [LibertyDST].

[MLPv3.1] and [LibertyGL] use different strategies for result/status codes. [MLPv3.1] has result codes indicating that there are certain type of problems with some element or attribute and then `<add_info>` element may refer to actual element or attribute. The status code values used by [LibertyGL] usually state the element causing the problem and the type of the problem, when used for second level status codes. The `ref` attribute should also be used to refer to the element causing the failure. On the other hand, the status code might point to a higher-level element than the exact element, e.g., `InvalidSelect`. The table below gives guidance how result and status codes between [MLPv3.1] and [LibertyGL] map to each other.

Table 2. Result/Status Code Mapping between [MLPv3.1] and [LibertyGL]

	OMA Mobile Location Protocol[MLPv3.1]	*Liberty ID-SIS Geolocation*[LibertyGL]
Resid	*Slogan*	*Status code*
0	OK	OK
1	SYSTEM FAILURE	– Use ID-* Fault message when the whole message fails and status code UnexpectedError, when at least one other request inside the same message succeeded.
2	UNSPECIFIED ERROR	UnspecifiedError
3	UNAUTHORIZED APPLICATION	ActionNotAuthorized
4	UNKNOWN SUBSCRIBER	InvalidResourceID
5	ABSENT SUBSCRIBER	AbsentSubscriber
6	POSITION METHOD FAILURE	PositionMethodFailure
101	CONGESTION IN LOCATION SERVER	– Use ID-* Fault message when the whole message fails and status code UnexpectedError, when at least one other request inside the same message succeeded.
102	CONGESTION IN MOBILE NETWORK	– Use ID-* Fault message when the whole message fails and status code UnexpectedError, when at least one other request inside the same message succeeded.
103	UNSUPPORTED VERSION	– Use ID-* Fault message when the whole message fails and status code UnexpectedError, when at least one other request inside the same message succeeded.
104	TOO MANY POSITION ITEMS	Use more specific status code referring to the actual problem, e.g., NoMultipleResources
105	FORMAT ERROR	Use more specific status code referring to the actual problem, e.g., InvalidSelect.
106	SYNTAX ERROR	Use more specific status code referring to the actual problem, e.g., NoMultipleAllowed
107	PROTOCOL ELEMENT NOT SUPPORTED	Use more specific status code referring to the actual problem, e.g., TypeNotSupported
108	SERVICE NOT SUPPORTED	Depending on the case either ActionNotSupported, if e.g., trying to subscribe to notifications, or more specific status code, if the problem is in smaller details, e.g., PeriodicNotificationsNotSupported
109	PROTOCOL ELEMENT ATTRIBUTE NOT SUPPORTED	Use more specific status code referring to the actual problem, e.g., ChangeHistoryNotSupported
110	INVALID PROTOCOL ELEMENT VALUE	Use more specific status code referring to the actual problem, e.g., InvalidResourceID
111	INVALID PROTOCOL ELEMENT ATTRIBUTE VALUE	Use more specific status code referring to the actual problem, e.g., InvalidExpires
112	PROTOCOL ELEMENT VALUE NOT SUPPORTED	Use more specific status code referring to the actual problem, e.g., RequestedGranularityNotSuported
113	PROTOCOL ELEMENT ATTRIBUTE VALUE NOT SUPPORTED	Use more specific status code referring to the actual problem, e.g., LocTypeNotAvailable
201	QOP NOT ATTAINABLE	QopNotAttainable
202	POSITIONING NOT ALLOWED	ActionNotAuthorized
203		
204	DISALLOWED BY LOCAL REGULATIONS	DisallowedByLocalRegulations
207	MISCONFIGURATION OF LOCATION SERVER	– Use ID-* Fault message when the whole message fails and status code UnexpectedError, when at least one other request inside the same message succeeded.
500 - 599		

3. Privacy Aspects of Liberty ID-SIS for Geolocation

The purpose of this chapter is to give an overview of the specific privacy aspects related to Location Based Services and to indicate how privacy issues are possible to address and resolve within the Liberty framework.

Location Based Services are applications that provide content or services to a person based on a combination of their registered personal profile and their location–often relative to some other location. Location Based Services will likely bring many advantages to end-users. Notwithstanding this potential value to users, such services introduce new privacy risks that must be addressed. The portability and increasing ubiquity of mobile devices, coupled with the ability to determine their location (and consequently the owning user) pose new risks for abuse.

While these applications promise significant benefit to end-users, the potentially sensitive nature of location information requires that the privacy issues be addressed. This chapter provides an overview of the types of Location Based Services that might be applicable and the privacy risks of sharing location information, as well as indicates how ID-SIS-GL, as part of the Liberty Alliance's architecture for permissions-based attribute sharing, can address these privacy requirements.

The Liberty ID-SIS Geolocation Service will be introduced as a standardized protocol for the sharing of a principal's geolocation data in a privacy-respecting manner.

3.1. Model

From a privacy point of view, the conceptual model for location sharing is shown in the diagram below.

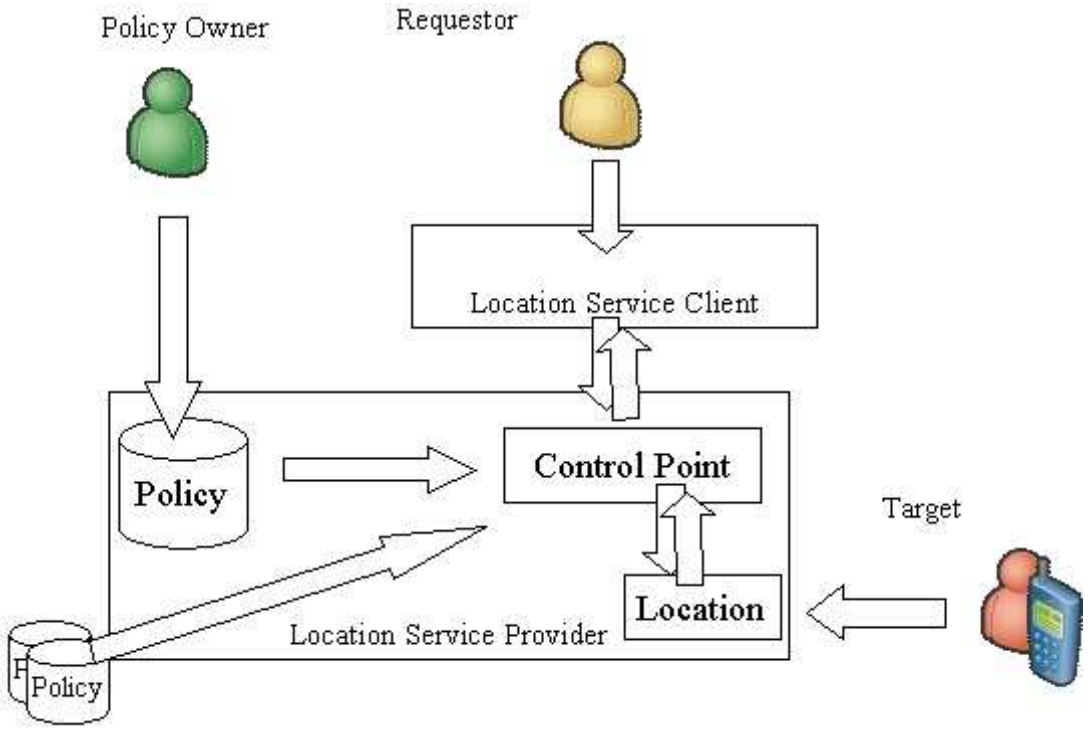

Figure 4. Conceptual Model for Location Sharing

1. A Requestor asks a Location Service Client to obtain the location data of a Target (here a Principal).

2. The Location Service Client makes use of [LibertyDisco] in order to discover resource offerings for location information of the specified principal. Here, the DS may already enforce policies controlling the access to the services (for a specific Principal). These policies may be controlled by the principal and/or by the Discovery Service Provider.
   When the Requestor is the Target, example applications might be "Where am I?" This is actually the case as depicted in Figure 1 and fully supported by [LibertyGL] in combination with the rest of available ID-WSF specifications.
   When the Requestor is different from the Target, example applications might be "Friend Finder" or "Alert me if somebody who likes jazz comes in the room." N.B. Non-standard ways of discovering resource offerings shall be employed in this case.

3. The Location Service Client then forwards the location request to a Location Service Provider that can determine the location of the target principal. Before releasing the location data, the Location Service Provider confirms that the privacy policy for that target (which may reflect the Requestor's identity and/or the Location Service Client identity) allows its release.
   Policies enforced by the Location Service Provider should specify the criteria for location release and/or rules governing notifications. Regulatory requirements may be expressed as external policy inputs.
   It is possible that the Policy Owner is different from the resource owner (target user being located). In this case, example applications might be "Tell me if my son leaves the city limits."
   The intended usages for the location data (potentially expressed in the request from the Location Service Client) may also feed into the decision to release the location data. It is up to the Location Service Provider to decide if intended usage for the location data is compulsory to be sent by the Location Service Client.

4. The location data is returned, finally, to the Location Service Client, which subsequently presents it to the Requestor in an appropriate format (e.g., as an overlay on a map, directions to take to get to the Target).

3.2. Liberty ID-SIS for Geolocation - Privacy Protections

Any system for sharing location information must be able to address the privacy issues/requirements identified in the previous section. In this section, we introduce the Liberty ID-SIS Geolocation Service and discuss how it accomplishes this.

There are a multitude of mechanisms by which the Mobile Service Network Operator can determine the location of a principal. If the Location Service Provider is also a Mobile Service Network Operator, then the location-based service can be delivered to the principal at once. However, if the Service Provider is not the Mobile Service Network Operator, then there is a disconnection between where the location information is held and where it is needed.

In other words, the Web Service Provider that offers some Location Based Services is not, in general, the same entity as the de-facto Location Provider.

The Liberty ID-SIS Geolocation Service addresses this issue. The [LibertyGL] defines an interoperability protocol by which Service Providers (acting as Location Service Clients) can request the location of a particular principal from the appropriate Location Service Provider (LSP) that has, or can determine, such location information.

As it is built on the Liberty ID-WSF infrastructure, [LibertyGL]automatically inherits the privacy enabling technologies defined within that framework. We discuss these mechanisms in the following sections.

3.2.1. Consent

Consent is fundamental to Liberty's model. Liberty Location Service Providers should offer Principals choice as to how, when, and to whom their location data is shared. Location Service Providers should also allow Principals to review, verify, or modify consents previously given.

ID-WSF-based entities may wish to claim whether they obtained the Principal's consent for carrying out any given operation, such as updating a Principal's Personal Profile entry. [LibertySOAPBinding] specification defines the `<Consent>` header block to allow Location Service Clients to indicate to the Location Service Provider that they have obtained the consent of the relevant Principal for the release of the location data.

The sample message below shows the `<Consent>` header block in a SOAP message requesting the release of a particular principal's location data.

```
<S:Envelope>
   <S:Header>
      <Consent id="A124395732495743"
         uri="urn:liberty:consent:obtained"
         timestamp="2112-03-15T11:12:10Z"/>
   </S:Header>
   <S:Body>
      Request for Location Data
   </S:Body>
</S:Envelope>
```

It is important to note that the `<Consent>` Header block as shown above is a claim made by the Location Service Client. The Location Service Provider's policy will determine if the claim is sufficient evidence of consent.

3.2.2. Usage Directives

[LibertySOAPBinding] provides a SOAP-based invocation framework for identity services. Within this framework, Liberty defines a usage directive container in which the policy requirements for attribute data, once released, can be carried.

Liberty ID-WSF has not yet defined particular semantics and/or processing rules for the `<UsageDirective>` Header block. As an example, even if the privacy policy for a principal were to allow their location data to be released, the Location Service Provider might include with the location data any obligations that the requesting Location Service Client must fulfill or be in breach. Similarly, the Location Service Client can use the same `<UsageDirective>` Header block on its request to indicate its intent for the location data, if released. This is shown below.

```
<S:Envelope>
   <S:Header>
      <UsageDirective S:mustUnderstand="1">
         <cot:PrivacyPolicyReference>
            http://circle-of-trust.com/policies/eu-compliant/location
         </cot:PrivacyPolicyReference>
         <serviceid>
            urn:liberty:sg:geoloc:purpose:emergency
         </serviceid>
      </UsageDirective>
   </S:Header>
   <S:Body>
      Request for Location Data
   </S:Body>
</S:Envelope>
```

The Location Service Client inserts a reference to a specific privacy policy for location data in a `PrivacyPolicyReference` element (defined by some Circle of Trust separate from Liberty). This information will feed into the Location Service Provider's decision to release the location data or not.

Additionally, the Location Service Client may insert an indication of the nature of the service that generated the location request (e.g., as requested by "`urn:liberty:sg:geoloc:purpose:emergency`" included in a `<serviceid>` element).

3.2.3. User Interaction

A Location Service Provider will sometimes need to interact with the principal for which location data is being requested in order to clarify privacy policy. [LibertyInteract] specification, is an ID-WSF specification that defines schemas and profiles that enable a Location Service Provider to interact with the owner of the resource that is exposed by that WSP.

[LibertyInteract] defines a profile that enables a WSC and a WSP to cooperate in redirecting the resource owner to the WSP and back to the WSC as well as elements, processing rules, and WSDL that together define an identity-based interaction service that can be made available temporarily by the WSC or offered on a more permanent basis by a party that has the necessary permanent channel to the Principal.

By definition, an Interaction Service is capable of interacting with the Principal at any time, for example, by using special protocols, mechanisms, or channels such as instant messaging, WAP Push, etc. Upon receiving the above request from the Location Service Provider, the Interaction Service is responsible for "rendering" a "form" to the Principal appropriate to the interaction channel.

An example of an `InteractionRequest` sent to the Interaction Service by the Location Service Provider that needs to obtain consent for the release of the corresponding principal's location to a specific Location Service Client is shown below.

```
<InteractionRequest xmlns="urn:liberty:is:2003-08">
   <ResourceID>data:d8ddw6dd7m28v628</ResourceID>
   <Inquiry title="Profile Provider Question">
      <Help moreLink="http://location.example.com/help/consent">
      Example.com is requesting your location. Please pick one of
      the provided options. Note that the last two options will ensure that
      you won't be asked this question when Example.com asks for your location again.
      </Help>
      <Select name="locationchoice">
         <Label>Do you want to share your location with Example.com?</Label>
         <Value>no</Value>
         <Item label="Not this time" value="no">
            <Hint> We won't give out your address but we'll ask you again next time
            </Hint>
         </Item>
         <Item label="Yes, once" value="yes">
            <Hint>We will share your address but will ask again next time.</Hint>
         </Item>
         <Item label="No, never" value="never">
            <Hint>We won't give out your address and won't ask you again</Hint>
         </Item>
         <Item label="Yes, always" value="always">
            <Hint>We will share your address now and in the future with Example.com
            </Hint>
         </Item>
      </Select>
   </Inquiry>
</InteractionRequest>
```

In the context of Liberty ID-SIS-GL, the user interaction mechanisms defined in [LibertyInteract] will be primarily used when the Location Service Provider cannot accurately decide on the release of the requested location information (e.g., if a Location Service Provider obtains location data that allows it to support different location aspects than for which it previously obtained the principal's privacy policy, it MUST obtain consent before release).

In general, interaction mechanisms also may be used in order to obtain the actual value of an attribute being requested. However, in the context of Liberty ID-SIS-GL, it will be very unlikely that a Location Service Provider will initiate a user interaction request with this purpose as the Location Service Provider should be able to determine principal's location itself.

3.2.4. Privacy Policies

Although a WSC requests some geolocation information related to a Principal, an LSP may decide not to return that information as Principal's privacy MUST always be protected. [LibertyDST] already sets some general requirements, but geolocation has some additional specific issues as it differs from basic services providing Principal's attributes. Global rules can not be specified as e.g., local regulations vary, but some issues are highlighted here.

- Normally, some information is either released or not as such, but, with geolocation, there might be additional options available for a Principal. The accuracy of the information can be modified. For example, instead of telling that a Principal is exactly at a certain place, an LSP may intentionally obfuscate the actual location of the principal by introducing semi-random errors, returning a bigger area in a `<shape>` element (not necessarily considering the actual location of the user in the center of that area), or just returning some sub-elements of the `<CivilData>` element instead of all (e.g., by returning `<St>` for State but not `<L>` for city information).

- Also, a Principal may define that her position MUST NOT be released, if she is in certain area, or her position MAY be released to certain requestors only during certain times, e.g., to employer only during working hours. If any of such privacy policies are defined, an LSP MUST follow them.

- On the other hand, there may be different local regulations causing safety issues to override user-defined privacy policies, e.g., in case of emergency services, the position of a Principal is needed regardless of the privacy settings. Also, parents may have the right to know where their children are. In this case, parents will be policy owners.

- In some cases, it may be normal if a WSC were to query, multiple times, for a particular Principal's location in a specific, possibly short, period of time (e.g., in fleet management or navigation scenarios). In most cases, though, it would be suspicious if a WSC were to do so since such multiple requests may provide more information than the Principal would be willing to reveal. If an LSP is able to detect these situations, the LSP may consider applying additional privacy policies to prevent an unauthorized WSC from tracking a particular Principal's location.

- [LibertyIDWSFSecurityPrivacyGuidelines], indicates that policies controlling the access to the attributes of a Principal are enforced in the attribute provider (the Location Service Provider in this case). This does not necessarily mean that the actual storage and evaluation of these policies needs to be performed at the Location Service Provider itself. It could rely on an external policy repository.
  However, details on the protocols and mechanisms for the interface between the Location Service Provider and that external policy repository are out of the scope of the Liberty Alliance and the ID-SIS-GL work in particular. In any case, it should be worth noting that any additional information required from Location clients in order, for example, to perform this external privacy checking functions could be optionally accommodated in the ID-SIS-GL SOAP messages using the extensions mechanisms defined by [LibertyGL].

3.2.5. Relative Location Requests

A WSC will sometimes 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. In other cases, it may be the policy of the LSP (as conducted by the principal) that might prevent the release of the actual location of the user while it still would be fine to inform whether the principal is within a particular area or not.

In either case, WSCs and LSPs will have the alternative to support requests for relative locations as defined by [LibertyGL]. A reference area against which the user's actual location can be compared is added in a request for a relative location and the LSP just returns `true` or `false` depending on the comparison result.

References

Informative

[LibertyDisco] Beatty, John, Hodges, Jeff, Sergent, Jonathan, eds. "Liberty ID-WSF Discovery Service Specification," Version 2.0-02, Liberty Alliance Project (24 Nov 2004). *http://www.projectliberty.org/specs*

[LibertyDST] "Liberty ID-WSF Data Services Template Specification," Version 2.0-06, Liberty Alliance Project (22 November, 2004). *http://www.projectliberty.org/specs* Kainulainen, Jukka, Ranganathan, Aravindan, eds.

[LibertyGL] Kainulainen, Jukka, eds. "Liberty ID-SIS Geolocation Service Specification," Version 1.0-12, Liberty Alliance Project (23 February, 2005). *http://www.projectliberty.org/specs*

[LibertyIDFFOverview] Wason, Thomas, eds. "Liberty ID-FF Architecture Overview," Version 1.2-errata-v1.0, Liberty Alliance Project (12 September 2004). *http://www.projectliberty.org/specs*

[LibertyIDPP] Kellomaki, Sampo, eds. "Liberty Identity Personal Profile Service Specification," Version 1.0, Liberty Alliance Project (12 November 2003). *http://www.projectliberty.org/specs*

[LibertyIDWSFGuide] Weitzel, David, eds. (13 January 2005). "Liberty ID-WSF Impelmentation Guideline," Draft version 2.0-02, Liberty Alliance Project *http://www.projectliberty.org/specs/*

[LibertyIDWSFOverview] Tourzan, Jonathan, Koga, Yuzo, eds. "Liberty ID-WSF Web Services Framework Overview," Version 1.0-errata-v1.0, Liberty Alliance Project (12 September 2004). *http://www.projectliberty.org/specs*

[LibertyIDWSFSecurityPrivacyGuidelines] Landau, Susan, eds. "Liberty ID-WSF Security and Privacy Overview," Version 1.0, Liberty Alliance Project (8 October 2003). *http://www.projectliberty.org/specs*

[LibertyInteract] Aarts, Robert, eds. "Liberty ID-WSF Interaction Service Specification," Version 2.0-01, Liberty Alliance Project (22 November 2004). *http://www.projectliberty.org/specs*

[LibertyPrivacy] Korentayer, E., eds. (14 April 2003). "Project Liberty Privacy and Security Best Practices," Release 2.0, Liberty Alliance Project *http://www.projectliberty.org/specs/Project_Liberty_Best_Practices4.14.03.pdf*

[LibertyReg] Kemp, John, eds. "Liberty Enumeration Registry Governance," Version 1.0, Liberty Alliance Project (12 November 2003). *http://www.projectliberty.org/specs*

[LibertySecMech] Ellison, Gary, Madsen, Paul, eds. "Liberty ID-WSF Security Mechanisms," Version 2.0-03, Liberty Alliance Project (22 November 2004). *http://www.projectliberty.org/specs*

[LibertySOAPBinding] Hodges, Jeff, Kemp, John, Aarts, Robert, eds. "Liberty ID-WSF SOAP Binding Specification," Version 2.0-01, Liberty Alliance Project (22 November 2004). *http://www.projectliberty.org/specs*

[MLPv3.1] "Mobile Location Protocol (MLP) Candidate Version 3.1," Open Mobile Alliance (16 March 2004). *http://member.openmobilealliance.org/ftp/public_documents/loc/Permanent_documents/OMA-LIF-MLP-V3_1-20040316-C.zip*