Liberty Alliance Project:

Liberty ID-SIS Presence Service Implementation Guidelines
Version: 1.0-12

Editors:
Justin Uberti, AOL

Contributors:
David Castellanos, Ericsson
Salima Fazal-Karim, France Télécom
Jukka Kainulainen, Nokia
Peter Saint-Andre, Jabber
Yoshitsugu Tsuchiya, NTT
Rob Lockhart, IEEE-ISTO

Abstract:

This document is an informative guide to implementers of the Liberty Alliance Presence Service, as described in the Liberty ID-SIS Presence Service Specification. It provides basic examples and explanations of how to perform common tasks when using this specification.

The reader is expected to be familiar with the Liberty ID-WSF Web Services Framework Overview, XML, SOAP, and presence services in general.

Filename: draft-liberty-id-sis-presence-guidelines-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 America Online, Inc.; American Express Travel Related Services; Ericsson; France Télécom; The
International Security, Trust, and Privacy Alliance (ISTPA); Jabber, Inc.; Nippon Telegraph and Telephone
Corporation; Nokia Corporation; Sun Microsystems, Inc.; and Vodafone Group Plc. All rights reserved.
Contents

1. Introduction ................................................................. 4
2. Recommended Liberty Presence Service Implementation .................................. 5
3. Privacy Considerations .................................................................. 10
References .................................................................................. 13
1. Introduction

This document provides a rationale and guidance for implementers of the Liberty Presence (ID-SIS-PRES) Service and focuses on guidelines for Presence Service Providers (PSPs) and Presence Service Clients (PSCs). A companion document, Liberty ID-SIS Presence Service Technical Specification [LibertyPresence], normatively describes the ID-SIS-PRES. This document is intended to provide recommendations to developers for matters that are not explicitly formalized in [LibertyPresence]. Definitions for Liberty-specific terms can be found in the [LibertyGlossary].

If there is disagreement between this document and [LibertyPresence], 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 presence services in general. 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 [LibertyPresence] 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], and
- Privacy and Security Best Practices [LibertyPrivacy]

1.2. High Level View of ID-SIS-PRES

The ID-SIS-PRES is provided by an attribute provider (AP) [LibertyIDWSFGuide], which we will refer to as a PSP in this document. The PSP is an ID-WSF web service that hosts the ID-SIS-PRES. PSCs, which may be Service Providers (SPs) themselves, make requests to the PSP on behalf of the Principal [LibertyIDWSFGuide].
2. Recommended Liberty Presence Service Implementation

As mentioned in [LibertyPresence], ID-SIS-PRES is a wrapper of sorts around existing presence protocols. Specifically, the service allows a PSC to make [LibertyDST] requests to a PSP concerning the presence information of a Principal; contained in those requests and in the associated replies are XML elements from either the SIMPLE PIDF [RFC3863] or XMPP [RFC3921] protocols specified by the IETF, or the IMPS [PA11] protocol specified by the Open Mobile Alliance. Examples of this encapsulation will be discussed in this section.

In each [LibertyDST] request, the target Principal is identified in the Resource element. It may also be the case with certain encapsulated protocols that the target is also identified in the encapsulated elements. In this case, both occurrences of the ID should match. If not, the value in the encapsulated protocol should be ignored.

2.1. Transaction Type Examples

The following are examples of [LibertyDST] operations for presence data. For clarity, the standard SOAP envelope is omitted from these examples; only the SOAP body contents are shown.

2.1.1. Queries for Presence Information

To query for presence information, the client will send a [LibertyDST] Query to the PSP for the Principal for which presence is to be retrieved. The XML namespace used for the query identifies which presence protocol is to be used, and the ResourceID element specifies a URI that references the aforementioned Principal. The PSP will reply with a [LibertyDST] QueryResponse back to the client with a Status element indicating the result of the request and, if successful, a Data element that contains XML elements from the chosen presence protocol.

2.1.1.1. SIMPLE Query/QueryResponse

```xml
  <ResourceID>pres:user@example.com</ResourceID>
  <QueryItem/>
</Query>

(QueryResponse xmlns="urn:liberty:id-sis:pres:simple:2005-07">
  <Status code="OK"/>
  <Data>
    <presence entity="pres:user@example.com" xmlns="urn:ietf:params:xml:ns:pidf">
      <tuple id="bs35r9">
        <status>
          <basic>open</basic>
        </status>
      </tuple>
    </presence>
  </Data>
</QueryResponse>
```

2.1.1.2. XMPP Query/QueryResponse

```xml
<Query xmlns="urn:liberty:id-sis-pres:xmpp:2005-07">
  <ResourceID>pres:user@example.com</ResourceID>
  <QueryItem/>
</Query>

<QueryResponse xmlns="urn:liberty:id-sis-pres:xmpp:2005-07">
  <Status code="OK"/>
  <Data>
    <presence from="user@example.com/bs35r9" xmlns="urn:jabber:client"/>
  </Data>
</QueryResponse>
```
2.1.1.3. IMPS Query/QueryResponse

2.1.2. Modification of Presence Information

To modify presence information (typically for itself), the client will send a [LibertyDST] Modify to the PSP. The XML namespace used for the request identifies the type of data contained inside, the ResourceID element specifies a URI that references the Principal, and the NewData element contains the XML elements from the chosen presence protocol that represent the new presence state of the Principal. The PSP will reply with a [LibertyDST] ModifyResponse back to the client with a Status element indicating the result of the request.

2.1.2.1. SIMPLE Modify/ModifyResponse

2.1.2.2. XMPP Modify/ModifyResponse
<NewData>
  <presence from="user@example.com/bs35r9" type="unavailable" xmlns="urn:jabber:client"/>
</NewData>
</Modification>
</Modify>
<ModifyResponse xmlns="urn:liberty:id-sis-pres:xmpp:2005-07">
  <Status code="OK"/>
</ModifyResponse>

2.1.2.3. IMPS Modify/ModifyResponse

<Modify xmlns="urn:liberty:id-sis-pres:imps:2005-07">
  <ResourceID>pres:user@example.com</ResourceID>
  <Modification>
    <NewData>
      <UserAvailability xmlns="http://www.wireless-village.org/PA1.1">
        <PresenceValue>NOT_AVAILABLE</PresenceValue>
      </UserAvailability>
    </NewData>
  </Modification>
</Modify>
<ModifyResponse xmlns="urn:liberty:id-sis-pres:imps:2005-07">
  <Status code="OK"/>
</ModifyResponse>

2.1.3. Subscriptions for Presence Information

In the interest of efficiency, PSCs/PSPs can support subscriptions for presence information so that the PSC is notified by the PSP whenever the presence information for the subscribed-to Principal changes, instead of having to ask the PSP continually for updates. To start a subscription, the PSC will send a [LibertyDST] Subscribe request, again specifying the XML namespace to identify the presence protocol and the ResourceID to identify the user to watch, along with an Endpoint URL to indicate where the notifications should be sent. The response to this is a [LibertyDST] SubscribeResponse, which contains a Status code indicating the success or failure of the request, and, if successful, a Notification element, which is described below.

2.1.3.1. SIMPLE Subscribe/SubscribeResponse

<Subscribe xmlns="urn:liberty:id-sis-pres:simple:2005-07">
  <ResourceID>pres:user@example.com</ResourceID>
  <Subscription expires="2005-10-27T16:49:29Z" invokeID="798fewq">
    <NotifyTo>
        http://notify.example.com/sink
      </sb-ext:Endpoint>
    </NotifyTo>
  </Subscription>
</Subscribe>
  <Status code="OK"/>
  <Notification invokeID="798fewq" subscriptionID="ui87" expires="2005-10-27T16:49:29Z">
    <Data>
      <presence entity="pres:user@example.com" xmlns="urn:ietf:params:xml:ns:pidf">
        <tuple id="bs35r9">
      </Data>
    </Notification>
</SubscribeResponse>
2.1.4. Async Notification of Presence Information

When the presence state of a Principal changes, the PSP will notify all subscribed PSCs. This is done by sending a [LibertyDST] Notify request to the PSC, which contains a Notification element with details about the subscription, and, within it, a Data element that contains the XML elements of the chosen presence protocol indicating the subscribed-to Principal’s current presence information. Upon receipt of this notification, the PSC will reply with a NotifyResponse, indicating that it processed the Notify properly (or not).

2.1.4.1. SIMPLE Notify/NotifyResponse

```xml
  <Notification subscriptionID="ui87" expires="2005-10-27T16:49:29Z">
    <Data>
      <presence entity="pres:user@example.com" xmlns="urn:ietf:params:xml:ns:pidf">
        <tuple id="bs35r9">
          <status>
            <basic>closed</basic>
          </status>
        </tuple>
      </presence>
    </Data>
  </Notification>
</Notify>
```

```
<NotifyResponse xmlns="urn:liberty:id-sis-pres:simple:2005-07">
  <Status code="OK"/>
</NotifyResponse>
```

2.2. Suggested Mapping of Presence Elements

As mentioned above, each of the three supported presence protocols have their own format for their presence information. When handling responses in different formats, it may be desirable to have a way to unify the various responses so that the individual presence protocol details are hidden from the higher layers of the application. Table 1 contains a simple, non-normative mapping for some common presence elements.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability: Status</td>
<td><code>&lt;basic&gt;</code> element (values: &quot;open&quot; or &quot;closed&quot;)</td>
<td><code>&lt;type</code> attribute (values: &quot;unavailable,&quot; if absent; assumed to be &quot;available&quot;)</td>
<td><code>&lt;OnlineStatus&gt;</code> element (values: &quot;T,&quot; &quot;F&quot;)</td>
<td><code>&lt;Address&gt;</code> element</td>
</tr>
<tr>
<td>Availability Address</td>
<td><code>&lt;contact&gt;</code> element</td>
<td><code>&lt;from</code> attribute</td>
<td><code>&lt;Address&gt;</code> element</td>
<td></td>
</tr>
<tr>
<td>Availability Free-form status</td>
<td><code>&lt;note&gt;</code> element</td>
<td><code>&lt;status</code>, or <code>&lt;show</code> element</td>
<td><code>&lt;StatusText&gt;</code> element</td>
<td></td>
</tr>
</tbody>
</table>

Liberty Alliance Project
Mappings for less commonly-used elements are contained in Table 2 below. Note that the SIMPLE elements mentioned are from the most recent [RPID] and [GEOPRIV] Internet Drafts, and may be subject to change.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Motion status summary</td>
<td>[RPID]&lt;activities&gt; element</td>
<td>[XMPP_user_activity]</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>&lt;activity&gt; element</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Geolocation</td>
<td>[GEOPRIV]&lt;geopriv&gt;</td>
<td>&lt;geoloc&gt;</td>
<td>GeoLocation element</td>
</tr>
<tr>
<td>Mood</td>
<td>[RPID]&lt;Mood&gt; element</td>
<td>[XMPP_user_mood]</td>
<td>&lt;StatusMood&gt; element</td>
</tr>
<tr>
<td></td>
<td>&lt;mood&gt; element</td>
<td></td>
<td></td>
</tr>
<tr>
<td>User Icon</td>
<td>[RPID]&lt;status-icon&gt; element</td>
<td>&lt;photo&gt; element</td>
<td>N/A</td>
</tr>
</tbody>
</table>

It is worth noting that some of these common fields in the supported presence protocols overlap with services offered in the Liberty framework. Among these services are the Liberty Personal Profile service [LibertyIDPP], which contains information about how to contact the user, and the Liberty Geolocation service [LibertyGL]. Both of these services offer information that is richer from what would be contained in the presence document and should be used, when possible.
3. Privacy Considerations

Presence information, by its very nature, is fairly sensitive. Even a simple on/off presence indication allows a watcher to have an idea of where a user may be and his or her typical schedule. When detailed status information is added to the presence info, control of the presence information becomes even more of a concern.

3.1. Authorization of PSCs

At a minimum, a user should be able to specify which PSCs are able to request presence data for him or her. The consequence of this to the PSP implementer is that incoming presence requests must be checked to ensure that their requesting PSC is authorized by the target Principal to receive his or her presence information.

Basic authorization of PSCs can be accomplished making use of the Liberty ID-WSF-defined SOAP headers:

- Liberty ID-WSF `<Provider>` header block, as defined by [LibertySOAPBinding], provides a means for a sender to claim that it is represented by a given `providerID` value.
- Liberty ID-WSF `<ResourceAccessStatement>` element within the `<saml:Assertion>` element of the Liberty ID-WSF `<wsse:Security>` header block, as defined by [LibertySecMech], conveys sufficient information regarding the accessing entity and the resource for which access is being attempted.

While the implementation and maintenance of any authorization lists is specific to the individual PSP, Liberty supplies some mechanisms to indicate the requestor of the presence data and how the data will be used.

3.1.1. Consent

The [LibertySOAPBinding] specification defines a `<Consent>` header block that allows PSCs to indicate to the PSP that they have obtained consent to query or modify the presence data for a Principal. PSPs can use this mechanism only to allow requests from clients that have explicitly been authorized by the Principal.

3.1.1.1. Example

The sample message below shows the `<Consent>` header block in a SOAP message requesting a particular Principal’s presence information:

```xml
<S:Envelope>
  <S:Header>
    <Consent id="A124395732495743"
      uri="urn:liberty:consent:obtained"
      timestamp="2112-03-15T11:12:10Z"/>
  </S:Header>
  <S:Body>
    <Query xmlns="urn:liberty:id-sis-pres:simple:2005-07">
      <ResourceID>pres:user@example.com</ResourceID>
      <QueryItem/>
    </Query>
  </S:Body>
</S:Envelope>
```

3.1.2. Usage Directives

[LibertySOAPBinding] also provides a `<UsageDirective>` header block that allows PSCs and PSPs to specify limits on how the user’s presence data shall be used. The semantics for this block have not yet been fully specified, but one possible usage would involve a URL being specified in the `<UsageDirective>` block in the presence response, and this URL would reference a policy that describes a set of obligations that the requesting PSC must fulfill or be in breach.
Alternatively, the PSC could use the `<UsageDirective>` to specify a URL that indicates its policy for handling the presence data.

### 3.1.2.1. Example

```xml
<S:Envelope>
    <S:Header>
        <UsageDirective S:mustUnderstand="1">
            <example:PrivacyPolicyReference>
                http://example.com/policies/presence
            </example:PrivacyPolicyReference>
        </UsageDirective>
    </S:Header>
    <S:Body>
        <Query xmlns="urn:liberty:id-sis-pres:simple:2005-07">
            <ResourceID>pres:user@example.com</ResourceID>
            <QueryItem/>
        </Query>
    </S:Body>
</S:Envelope>
```

### 3.1.3. User Interaction

The [LibertyInteract] specification is an ID-WSF specification that defines schemas and profiles that enable a PSP to interact with the owner of the resource that is exposed by that PSP. [LibertyInteract] defines a profile that enables a PSC and a PSP to cooperate in redirecting the resource owner to the PSP and back to the PSC as well as elements, processing rules, and the WSDL that, together, define an identity-based interaction service that can be made available temporarily by the PSC 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 or WAP Push. Upon receiving the above request from the PSP, the Interaction Service is responsible for "rendering" a "form" to the Principal appropriate to the interaction channel.

In the context of ID-SIS-PRES, the user-interaction mechanisms defined in [LibertyInteract] will be primarily used when the PSP can not accurately decide on the release of the requested presence information.

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 ID-SIS-PRES, this might be applicable only for specific presence information (e.g., "Availability: Free From Status"), but not all, since a PSP should be able to determine some of the Principal's presence information itself (e.g., "Status: Online/Offline").

### 3.1.3.1. Example

An example of an InteractionRequest sent to the Interaction Service by the PSP that needs to obtain consent for the release of the corresponding Principal’s presence information to a specific PSC is shown below.

```xml
<InteractionRequest xmlns="urn:liberty:is:2003-08">
    <ResourceID data:d8ddw6dd7m28v628</ResourceID>
    <Inquiry title="Profile Provider Question">
        <Help moreLink="http://presence.example.com/help/consent">
            Example.com is requesting your presence information. 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 presence information again.
        </Help>
    </Inquiry>
</InteractionRequest>
```
<Select name="presencechoice">
  <Label>Do you want to share your presence information with Example.com?</Label>
  <Value>no</Value>
  <Item label="Not this time" value="no">
    <Hint>We won’t give out your presence information this time but we’ll ask you again next time</Hint>
  </Item>
  <Item label="Yes, once" value="yes">
    <Hint>We will share your presence information and will ask again next time.</Hint>
  </Item>
  <Item label="No, never" value="never">
    <Hint>We won’t give out your presence information and won’t ask you again</Hint>
  </Item>
  <Item label="Yes, always" value="always">
    <Hint>We will share your presence information now and in the future with Example.com</Hint>
  </Item>
</Select>
References

Informative


