Liberty ID-SIS Personal Profile Service Implementation Guidelines

Version: 1.1

Editors:
Sampo Kellomäki, Symlabs, Inc.
Rob Lockhart, IEEE-ISTO

Contributors:
Rajeev Angal, Sun Microsystems, Inc.
Carolina Canales-Valenzuela, Ericsson
David del Ser, Vodafone Group Plc
Andy Feng, America Online, Inc.
Ariel Gordon, France Télécom
Vincent Guesdon, France Télécom
Jukka Kainulainen, Nokia Corporation
Lena Kannappan, France Télécom
Bronislav Kavsan, RSA Security Inc.
John Linn, RSA Security Inc.
Jonathan Sergent, Sun Microsystems, Inc.
John Kemp, IEEE-ISTO
Thomas Wason, IEEE-ISTO

Abstract:

This document provides implementation guidelines supplemental to the Liberty ID-SIS Personal Profile (ID-SIS-PP) specification. It is also the general guideline for Liberty Profiles. The reader is expected to be familiar with the Liberty ID-WSF Web Services Framework Overview, XML, SAML and SOAP. ID-SIS-PP is a web service hosted by an application provider and usually discovered via a discovery service. It offers basic profile information regarding Principal, including name, legal identity, and a minimal set of contact information such as legal domicile, home, and work addresses. The profile may also contain phone numbers, emails, and other online contact information. Some basic demographics and presentation information and employment and public key details may also be included. An extension mechanism allows other arbitrary data to be included. An ID-SIS-PP service only stores information regarding the Principal and does not target contact management or e-commerce applications. A typical Principal has two ID-SIS-PP service instances, one for a work identity, and another for a private identity. An ID-SIS-PP 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-PP service, like all data services, is characterized by an ability to query and update attribute data. It incorporates mechanisms from other specifications for access control and for conveying data validation information and usage directives.

Filename: liberty-idsis-pp-guidelines-v1.1.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. Introduction

The ID-SIS Personal Profile specification defines a Liberty identity service that supports identity information regarding the Principal itself, be it in a private or work capacity. It is not intended to be a fully generic contact book and may not address all requirements of e-commerce applications. It is intended to be the least common denominator for holding information about the Principal him or herself. Other services, not necessarily defined by Liberty such as wallet and contact book, will address specific applications in a more comprehensively.

This document provides a rationale and guidance for implementers of the ID Personal Profile. A companion document, Liberty Identity Personal Profile Service Specification [LibertyIDPP], normatively describes the ID Personal Profile. If there is disagreement between present document and [LibertyIDPP], 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 and SOAP. The reader should be familiar with the Liberty ID-FF Architectural Overview ([LibertyIDFFOverview]) and the Liberty D-WSF Web Services Framework Overview ([LibertyIDWSFOverview])

1.2. Co-Existence of Private and Work Profiles

The ID-SIS-PP contains many types of information; not all of it is appropriate in all contexts. It is expected that a Principal who is employed typically will have at least two ID-SIS-PP services: one for holding information appropriate while acting in a private capacity and another while acting as an employee of a company. The two ID-SIS-PPs could be attached to two different identities, one within a consumer-oriented Identity Provider (IdP) and another within an employer's private IdP. Alternatively, the two profiles could be contained within one identity (e.g., if employer chose to outsource the IdP function to some IdP that also accepts consumers).

The consumer-oriented ID-SIS-PP service providers need not hold the `EmploymentIdentity` container, while an enterprise could provide a ID-SIS-PP service for its employees and this service would be maintained by the human resources department, potentially limiting a Principal's control over the data held according to the policies of the company.

A Principal having two such ID-SIS-PP services would usually have both of them registered in the discovery service. An implementation-dependent mechanism in the discovery service could allow the Principal to choose which ID-SIS-PP service to use on an SP-by-SP or transaction-by-transaction basis. Such a choice amounts to a Principal being able to decide whether she wants to act in her personal or professional capacity in any given situation.

The information in the private life ID-SIS-PP may be surrendered voluntarily by the Principal. This information is not likely to be validated to high standards. By contrast, the contents of the employee ID-SIS-PP are likely to be validated by an HR department. Thus when an employee uses his work ID-SIS-PP and this information is served by an attribute provider (AP) hosted by the employer, the employer is vouching for employee's identity and attributes. These are secured by the digital signature of the employer (because employer runs the AP and in a Liberty implementation the AP will sign the attribute response).

The private and work (employee) ID-SIS-PPs may use some of the same attributes, but this specification does not require that the values should be the same. Synchronization may exist, but there can be situations where this is not desirable. It is likely that the two profiles are hosted by different organizations therefore requiring synchronization is not feasible.

1.3. Architectural Context of the ID-SIS-PP

The Liberty Identity Personal Profile 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 may be used by other Liberty services as they all 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-PP 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].

1.3.1. ID-SIS-PP as an interface

Although the essence of the ID-SIS-PP 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 profile service; no particular implementation is mandated. The specification can be considered to provide a "dictionary" of data 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 data for same attributes to different service providers. Thus an implementation may choose to hold some of the attributes in a database while obtaining others on the fly or computing them.

The data accessible through ID-SIS-PP often comes from backend systems that may serve other purposes as well. For example, an enterprise hosting an ID-SIS-PP service for its employees may choose to use their human resources database, or the ID-SIS-PP backend may also be used by a contact book service. Such sharing of backends is considered normal practice and may cause one service to update data in another "out-of-band." Out-of-band updates are expressly allowed, but are considered out of scope for purposes of ID-SIS-PP specification.

This specification, at formal and conceptual level, specifies a XML document. However, this does not mean that data is necessarily stored as a XML document. The data could just as well be computed on the fly or fetched from a directory (LDAP) or relational database (SQL) server and formatted into XML only for purposes of speaking Liberty protocols. When this document specifies behavior against conceptual XML document, the implementation has to behave as if the document existed, but does not necessarily have to implement it in concrete terms.

1.3.2. Participants and compliance testing

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

ID-SIS-PP compliance testing addresses both implementations and instances. ID-SIS-PP specifies an interface to which an *implementation* and an *instance* (deployment) of ID-SIS-PP service conform. The implementation may be a software product offered by a *vendor*. Typically such a product, if configured and operated correctly, will provide an ID-SIS-PP service instance. For an AP instance to be ID-SIS-PP compliant, it must use correctly an ID-SIS-PP compliant implementation.

1.4. XML Document Instantiations

An ID-SIS-PP service may respond to a query with an XML instantiation of a Profile schema. The XML documents that are specified by the Liberty Personal Profile XML schemas are the most general serial representations of the information. The expression "most general" means that a document could fully instantiate that schema if all data has been provisioned and no permissions filtering occurs. After filtering, the transmitted content may no longer conform to this schema. Thus implementers may need to adjust this schema before using it to implement services. Generally the adjustments will involve setting all minOccurs specifications to zero.

When queries that point to interior elements of the conceptual XML document are applied, the returned data does not contain the higher level containers. It contains the queried element and its contents. The specific higher level containers are to be inferred from the context provided by the query. Therefore the XML schema permits any and every element of ID-SIS-PP to serve as a top level element. This ensures that serial representations can always be compatible with the ID-SIS-PP schema. This does not imply that the underlying conceptual XML document could have any arbitrary element at the top level. The underlying conceptual XML document is always considered to be rooted on a single ID-SIS-PP container.

A potential confusion is that as requests to ID-SIS-PP service are actually SOAP documents, there is one schema for the SOAP layer and another for the document that is returned inside the SOAP response. The Liberty ID-SIS-PP specification does not define the SOAP schemas.

1.5. Extension mechanisms

There are six methods for extending the ID-SIS-PP specification:

1. by adding more enumerator URIs to existing attributes
2. by adding new attributes to existing containers
3. by creating new containers
4. by creating new discovery option keywords (URIs)
5. by extending the supported subset of XPATH expressions
6. by schema extension

For attribute names and container names the extensions use their own XML namespace. If a component that was formerly an extension is adopted by Liberty, it is no longer an extension. The adoption of extensions is an intended path for the evolution of the Liberty Profiles.

If an implementation supports schema extension, it is usually convenient to also register extended discovery option keywords and support a richer vocabulary of XPATH expressions as well.

It is expected that some extensions will eventually become adopted, moving into the "main stream" of the Liberty specifications. This will, unfortunately, create situations where the same attribute may exist at the same time in an experimental namespace and in the official ID-SIS-PP namespace. Implementations SHOULD be programmed to accept both variants, but MUST NOT emit attributes using the official namespace until approved. This strategy allows an extension to be toggled into the Liberty namespace and structure if it becomes adopted by Liberty. This strategy allows all attribute consumers to automatically recognize the new attribute. As a transitional measure the attribute provider MAY emit an attribute twice: once in the experimental form and once in the official form. There is no guarantee that all extensions will become adopted.

ID-SIS-PP elements that are enumerations use URIs as enumerators (values). Each element's description details the authority for adopting new official enumerators. In some cases, such as country and language codes, enumerators have been assigned by a well-established international standards body. In other cases, this specification defines some enumerators and stipulates that a registry [LibertyReg] may assign additional official enumerators. Organizations and industry consortia are allowed to define and manage their own extensions.

2. Overview of the Data Model

The following is a summary of the major sections, or "containers," of the Personal Profile data structure. The containers have been illustrated graphically. Specific details defining the elements and data types are contained in the ID-SIS-PP Specification [LibertyIDPP] All top level containers are optional (obligation = optional), some may be repeated. The specification defines data capabilities, it does not define the specific data that any particular implementation must support. An implementation should publicly reveal the portions of the specifications that is supports. Additionally, an implementation may extend the Liberty data model using well-defined mechanisms.

2.1. Structure of the PP Data Model

Table 1. Structure of the PP Data Model

Attribute	Oblig.	Example	Synopsis
`InformalName`	Optional	theWanderer	Screen name of the Principal
`CommonName`	Optional	(container)	The way the user likes to be called in every day situations
`LegalIdentity`	Optional	(container)	Official legal identification of the Principal
`EmploymentIdentity`	Optional	(container)	Minimal Employer and employment details
`AddressCard`	Optional	(container)	An address card for ID-SIS-PP
`MsgContact`	Optional	(container)	Generic phone, email, or instant messaging contact
`Facade`	Optional	(container)	Principal's look and sound facade
`Demographics`	Optional	(container)	Base level demographics used by ID-PP
`SignKey`	Optional		Principal's public key or certificate for signing
`EncryptKey`	Optional		Principal's public key or certificate for encryption
`EmergencyContact`	Optional	Contact spouse Mary Lee at ...	Next of kin or other person to contact if Principal has medical emergency

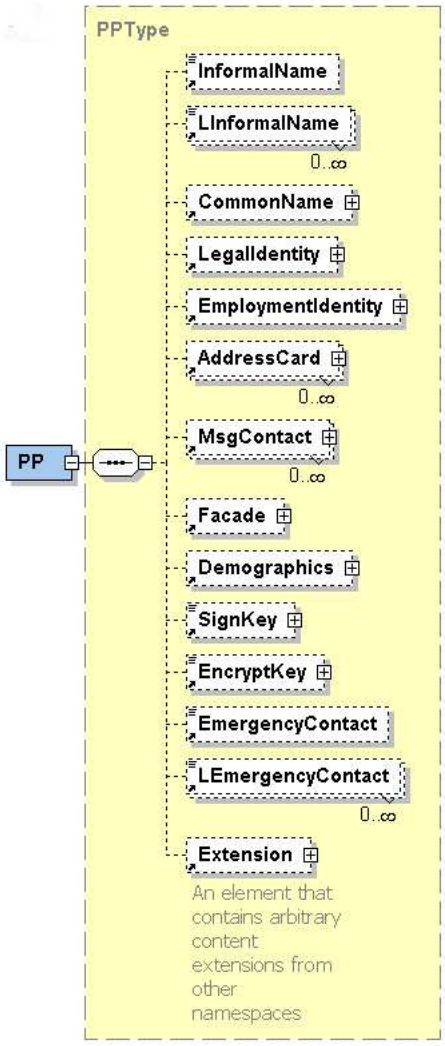


Figure 1. Top Level Personal Profile structure.

Example

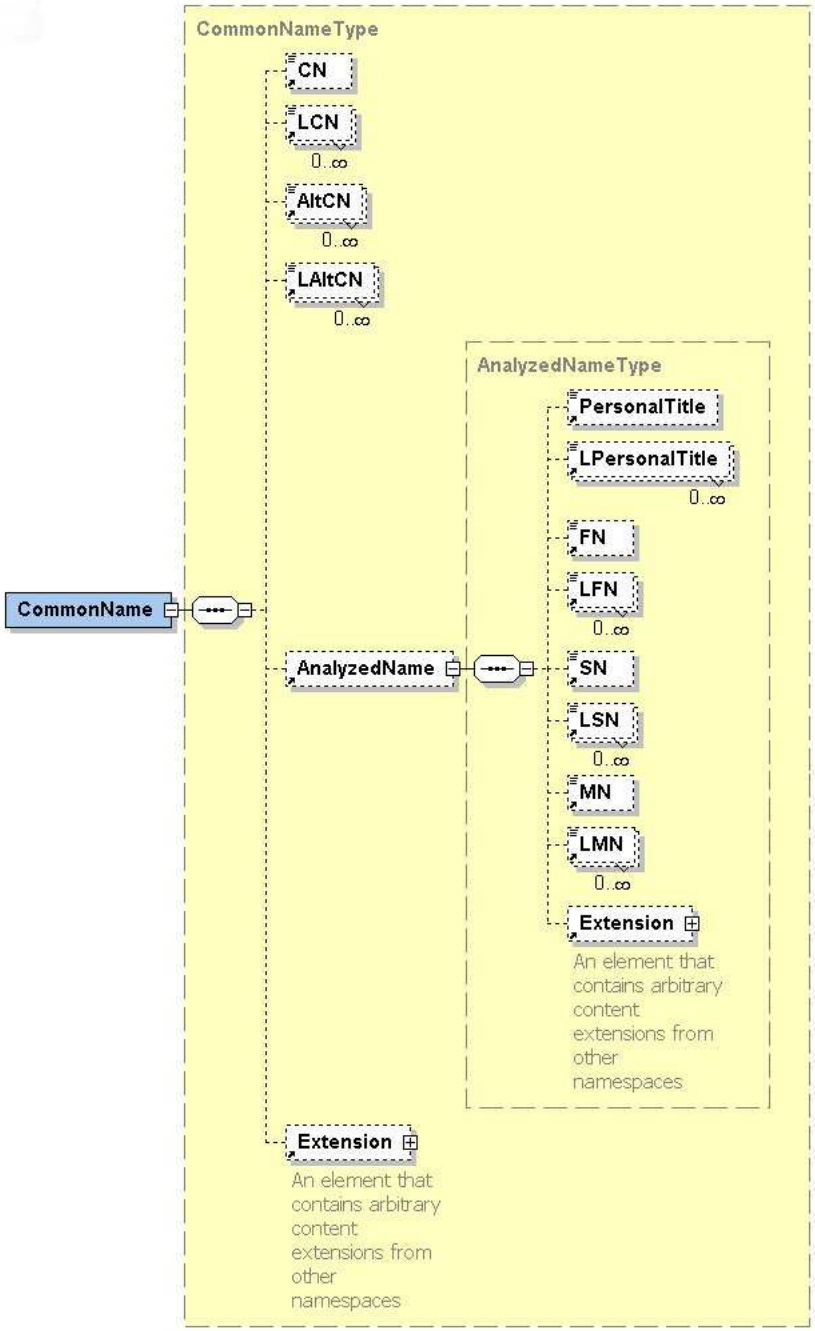
```
<PP>
   <InformalName>theWanderer</InformalName>
   <CommonName>
      <CN>Zita Lopes</CN>
      <LCN xml:lang="es">LKj343asas</LCN>
      <AltCN>Maria Lopes</AltCN>
      <AltCN>Zita Lopes</AltCN>
      <AnalyzedName nameScheme="">
         <PersonalTitle>Dr.</PersonalTitle>
         <FN>Zita</FN>
         <SN>Lopes</SN>
         <MN>Maria</MN>
      </AnalyzedName>
   </CommonName>
   <LegalIdentity>
      <LegalName>Zita Maria Oliveira da Figueira Lopes</LegalName>
      <AnalyzedName nameScheme="">
         <PersonalTitle>Dr.</PersonalTitle>
         <FN>Zita</FN>
         <SN>Lopes</SN>
```

```
         <MN>Maria</MN>
      </AnalyzedName>
      <VAT>
         <IDValue>502677123</IDValue>
         <IDType>urn:liberty:id-sis-pp:IDType:itcif</IDType>
      </VAT>
      <AltID>
         <IDValue>502677123</IDValue>
         <IDType>urn:liberty:id-sis-pp:IDType:itcif</IDType>
      </AltID>
      <DOB>1982-04-15</DOB>
      <Gender>urn:liberty:id-sis-pp:gender:f</Gender>
      <MaritalStatus>urn:liberty:id-sis-pp:maritalstatus:divorced</MaritalStatus>
   </LegalIdentity>
   <EmploymentIdentity>
      <JobTitle>CIO</JobTitle>
      <O>Mercnet Corp.</O>
      <LO>Mercnet Corp.</LO>
      <AltO>Mercnet Corp.</AltO>
      <AltLO>Mercnet Corp.</AltLO>
   </EmploymentIdentity>
   <AddressCard>
      <AddrType>urn:liberty:id-sis-pp:addrType:domicile</AddrType>
      <Address>
         <PostalAddress>c/o Carolyn Lewis$2378 Madrona Beach Way North</PostalAddress>
         <PostalCode>98503-2341</PostalCode>
         <L>Olympia</L>
         <St>wa</St>
         <C>us</C>
      </Address>
      <Nick>Joe Work</Nick>
   </AddressCard>
   <MsgContact>
      <Nick>Joe Work</Nick>
      <Comment>This is very important</Comment>
      <Comment>but may change in future. (example comment)</Comment>
      <MsgType>urn:liberty:id-sis-pp:msgType:mobile</MsgType>
      <MsgMethod>urn:liberty:id-sis-pp:msgMethod:im</MsgMethod>
      <MsgTechnology msgLimit="160">urn:liberty:id-sis-pp:msgTechnology:icq</MsgTechnology>
      <MsgProvider>AOL</MsgProvider>
      <MsgAccount>123435234</MsgAccount>
      <MsgSubaccount>1</MsgSubaccount>
   </MsgContact>
   <Facade>
      <MugShot>http://fotoserver.com/~joedoe/face.gif</MugShot>
      <WebSite>http://provider.com/~user</WebSite>
      <NamePronounced>http://fotoserver.com/~joedoe/name.wav</NamePronounced>
      <GreetSound>http://fotoserver.com/~joedoe/greet.wav</GreetSound>
      <GreetMeSound>http://fotoserver.com/~joedoe/greetme.wav</GreetMeSound>
   </Facade>
   <Demographics>
      <DisplayLanguage>pt-br</DisplayLanguage>
      <Language>pt</Language>
      <Language>fi</Language>
      <Language>en</Language>
      <Birthday>--05-09</Birthday>
      <Age>18</Age>
      <TimeZone>+05:00</TimeZone>
   </Demographics>
   <EmergencyContact>Contact spouse Mary Lee at ...</EmergencyContact>
   <LEmergencyContact>Contact spouse Mary Lee at ...</LEmergencyContact>
</PP>
```

2.2. CommonName

277 **Table 2. CommonName**

Attribute	Example	Synopsis
`CN`	Zita Lopes	Every day name in latin writing system
`AltCN`	Maria Lopes	Additional every day names in latin writing system
`AnalyzedName`	(container)	Name analyzed into bits and pieces



278

279 **Figure 2. CommonName Container**

280 The `AnalyzedName` container will be widely used, so it is detailed separately in the Name and Identity Management
281 section. See the discussion of name and identity management below.

2.3. LegalIdentity

See Section 6 on Managing the Principal's Name and Identity

Table 3. LegalIdentity

Attribute	Example	Synopsis
`LegalName`	Zita Maria Oliveira da Figueira Lopes	Full legal name in latin writing system
`AnalyzedName`	(container)	Name analyzed into bits and pieces
`VAT`	(container)	Fiscal identification number
`AltID`	(container)	Other identification number(s)
`DOB`	1982-04-15	Date of Birth
`Gender`	`urn:liberty:id-sis-pp:gender:f`	Gender of the Principal
`MaritalStatus`	`urn:liberty:id-sis-pp:maritalstatus:divorced`	Marital status such as single or married

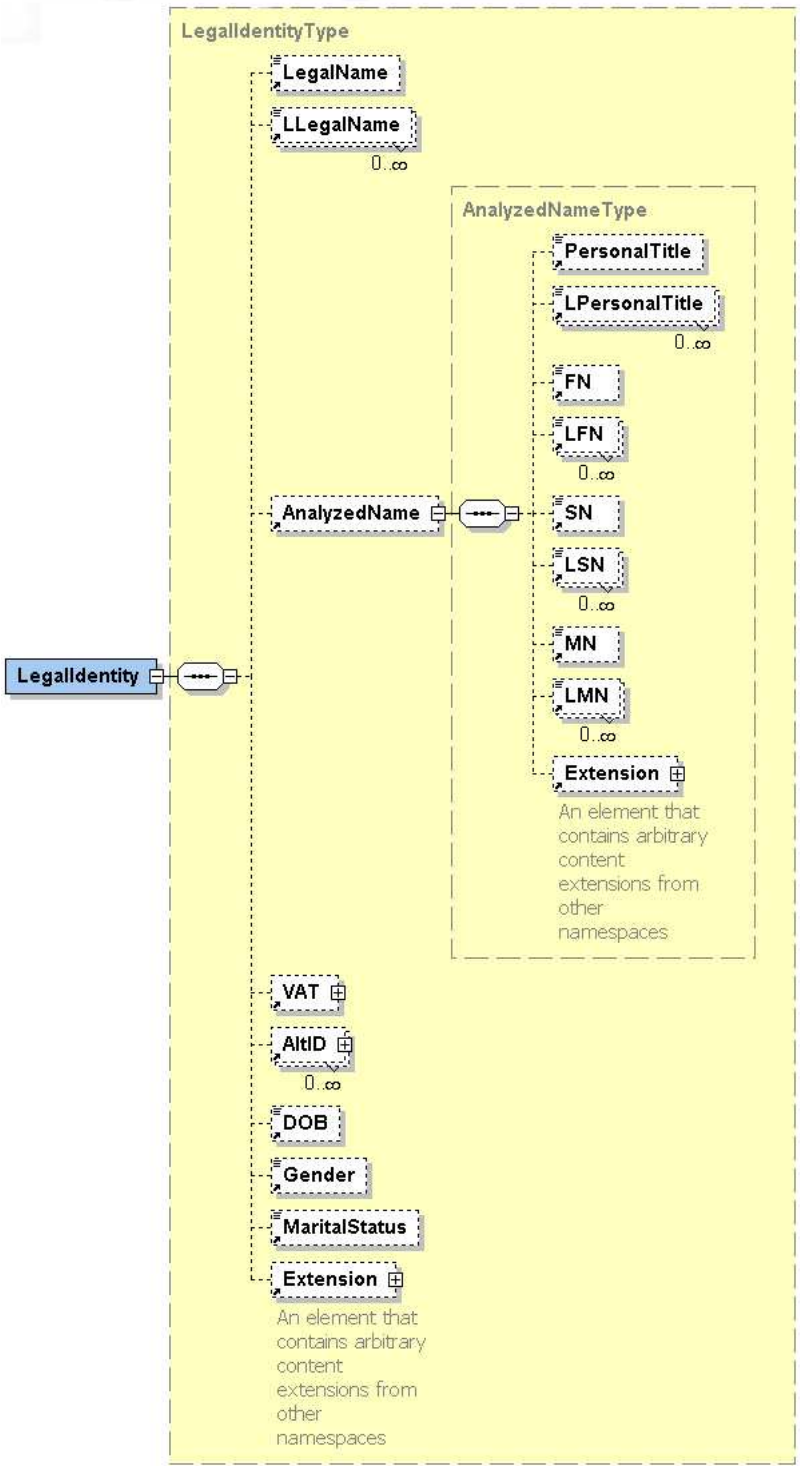
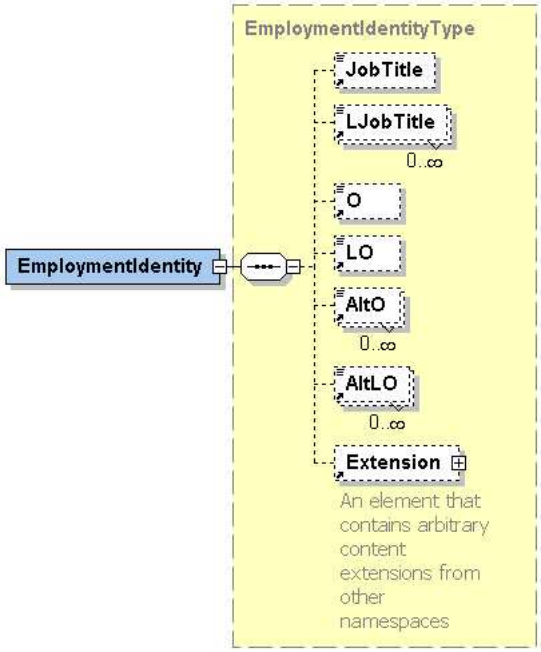


Figure 3. LegalIdentity Container

2.4. EmploymentIdentity

288 **Table 4. EmploymentIdentity**

Attribute	Example	Synopsis
`JobTitle`	CIO	Job title in latin script
`O`	Mercnet Corp.	Informal name of an organization
`AltO`	Mercnet Corp.	Additional informal names of an organization



289

290 **Figure 4. EmploymentIdentity Container**

291 2.5. AddressCard

292 **Table 5. AddressCard**

Attribute	Example	Synopsis
`AddrType`	`urn:liberty:id-sis-pp:addrType:domicile`	Marks the role of an AddressCard
`Address`	(container)	Commonly used bundle of postal address fields
`Nick`	Joe Work	Nick name for identifying item in user interface

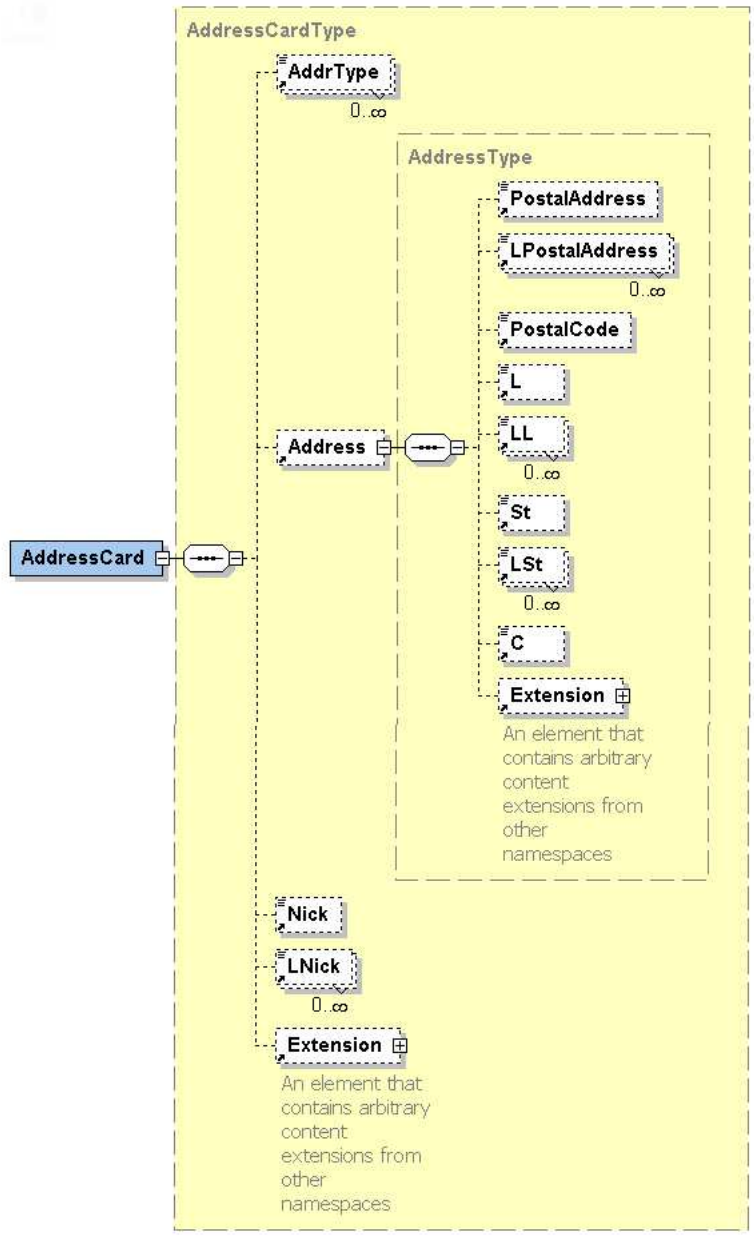
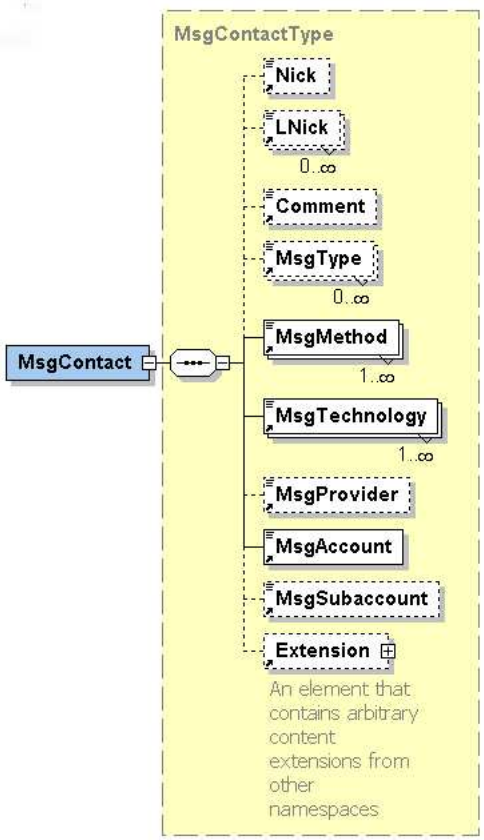


Figure 5. AddressCard Container

2.6. MsgContact

296 **Table 6. MsgContact**

Attribute	Example	Synopsis
`Nick`	Joe Work	Nick name for identifying item in user interface
`LNick`	Joey	Localized nick name for identifying item in user interface
`LComment`	This is very important	Private comment about a data object, localized
`MsgType`	`urn:liberty:id-sis-pp:msgType:mobile`	Usage role of the messaging contact
`MsgMethod`	`urn:liberty:id-sis-pp:msgMethod:im`	Messaging method associated with this contact
`MsgTechnology`	`urn:liberty:id-sis-pp:msgTechnology:icq`	Messaging technology or protocol associated with this contact
`MsgProvider`	AOL	Service provider or domain that provides messaging services
`MsgAccount`	123435234	Account or address information within messaging provider
`MsgSubaccount`	1	Subaccount within messaging account, such as voice mail box under phone number



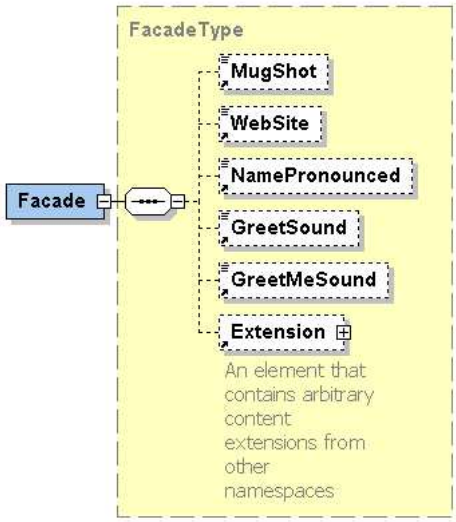
297

298 **Figure 6. MsgContact Container**

299 2.7. Facade

300 **Table 7. Facade**

Attribute	Example	Synopsis
MugShot	http://fotoserver.com/~joedoe/face.gif	Face photo
WebSite	http://provider.com/~user	Web site of the Principal
NamePronounced	http://fotoserver.com/~joedoe/name.wav	User's common name pronounced (usually by the user)
GreetSound	http://fotoserver.com/~joedoe/greet.wav	Greeting sound, user saying "Hello" to someone else
GreetMeSound	http://fotoserver.com/~joedoe/greetme.wav	Sound for user interface to greet the user



301

302 **Figure 7. Facade Container**

303 2.8. Demographics

304 **Table 8. Demographics**

Attribute	Example	Synopsis
DisplayLanguage	pt-br	The language the Principal prefers for displayed user interfaces
Language	pt	Languages the Principal is able to understand
Birthday	–05-09	Birthday without year
Age	18	Age of the Principal in years
TimeZone	+05:00	Time zone of the Principal

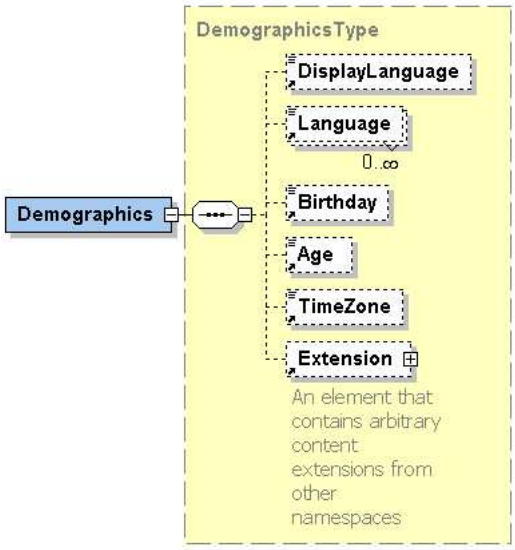


Figure 8. Demographics Container

3. Security Considerations

For the most part ID-SIS-PP relies on standard privacy and security mechanisms provided by ID-FF and ID-WSF. Of these the following are considered to be of paramount importance:

1. Ability to have several ID-SIS-PP service instances per principal. This allows the principal to have effective control over who holds which data about her; consequently, the existence of some piece of data in one place does not imply that other pieces of data need to be kept in the same place. This is especially important when considering that many principals are expected to want to maintain separation between their work and private lives, combined with the fact that an employer is likely to mandate that the work related profile be hosted on attribute provider it controls. The most important element supporting several ID-SIS-PP service instances is the ID-WSF Discovery Service, particularly its discovery option keyword registration feature.

2. Flexible permissions enforcement. It is important that Liberty recognizes that permissions enforcement will happen at all layers and is under control of the principal, even if, technically speaking, Liberty has framed permissions enforcement mechanisms as out of scope for the standardization effort.

3. Usage directives. They are a logical companion and combined with digital signatures provide the necessary audit trail and accountability so that abusers can be kept in check and the system can enjoy wide public confidence.

4. Solid architectural foundation so that above mentioned higher level mechanisms can be relied to work effectively. A solid foundation includes things such as transport layer security, application of digital signatures to both requests and response and a flawless crypto system and protocol design.

Most ID-SIS-PP specific privacy concerns can be addressed by properly configuring the permissions mechanisms.

1. Reliable control of access to see various ID numbers that may be held in ID-SIS-PP. The permissions for the IDs needs to be tightly maintained. Most of these IDs are in the LegalIdentity container. The permissions need to take in consideration both principal's preference and the legal obligations that may vary from jurisdiction to jurisdiction.

2. Tight control of the principal's full legal name, date of birth, gender, and other attributes that are customarily used for formal identification purposes.

3. The date of birth can be discovered by combining the principal's age and birth day. The latter two attributes exist separately to allow avoiding disclosing a date of birth to entertainment services that are either performing an age check or are sending greetings cards, but their simultaneous disclosure will effectively disclose the date of birth. The permissions should be set such that this type of disclosure can not occur inadvertently.

4. Most services that request profile information have a narrow scope of interest. An administrator of the ID-SIS-PP provider should be able to determine what information can legitimately be needed for implementing a particular service. The default permissions should take this into consideration so that information is only disclosed on "need to know" basis rather than through a blanket disclosure.

5. Some pieces of private life information may not be appropriate in working life. This should be reflected in the permissions.

4. Discovery and Queries

4.1. Rationale

The ID-SIS-PP is intended to be the "least common denominator" of information available about the Principals. However, even the "least" is seldom available to all service providers all of the time. The ID-SIS-PP that a service provider sees is apt to be incomplete because:

- the instance of the ID-SIS-PP has chosen to support only some subset of ID-SIS-PP. For example, a consumer oriented ID-SIS-PP service might choose to not support the `EmploymentIdentity` container while a ID-SIS-PP service run by an enterprise for its employees might not support the `Facade` container.
- not all information about the Principal was provisioned.
- national legislation forbids some information from being collected or shared across international boundaries.
- an attribute provider's policy forbids the SP from having some information.
- the permissions that the Principal sets forbids the SP from getting parts of the information.

Therefore the concept of "the one core profile" is not realized.

It is more fruitful to approach the ID Personal Profile from the perspective of a "need to know." For any given service it ought to be possible to determine what is the minimal set of information needed to provide the service. The need to know principle is consistent with guidelines for fair information use.

A consideration of the possible applications and their minimal information needs results in collecting the attributes into several containers that correspond to these information needs. These containers are useful abstractions because they are open ended mechanisms for grouping attributes and for assigning permissions to them. The container grouping also provides natural basis for requesting meaningful groups of attributes and discovering which attribute provider provides them.

For example, frequently it is not useful for the end users to think of access permissions by an e-commerce company to an address in terms of the individual attributes of street address, city, and state. It is much more meaningful to assign the permissions at the `AddressCard` level. Similarly, it is more convenient that service providers express their information needs in terms of containers of attributes.

Attribute containers aggregate the attributes into meaningful blocks; however, there are many containers, some of which may not be of interest to a particularly discovery service. Therefore the discovery service operates at granularity of an option keyword. Option keywords are used to discover the existence or support for particular containers or groups of containers in a way meaningful to applications. See [LibertyDisco] for generic definition of the Discovery Service and the processing rules for discovering by keyword.

4.2. Ambiguity if multiple APs host the same data

If two attribute providers (APs) register to host the same data, the choice of which AP will be used is implementation dependent. The first point of control is the discovery service which, despite multiple registrations, may still choose to return only one AP in an implementation dependent manner. If the discovery service returns multiple APs then the SP decides, according to its implementation, which one(s) to use. For example, an SP may use the first AP, may prompt user to choose, or may query all of the APs and combine the information.

It should be noted that due to private life - employee dichotomy it is quite likely that any given Principal will have at least two ID-SIS-PP services with largely overlapping sets of attributes; consequently, a basic discovery service is likely to prove inadequate. Discovery service implementations are encouraged to provide features that allow Principals to choose which ID-SIS-PP to use in each context. This could involve recording preferences or even prompting the Principal using the [LibertyInteract] or other means.

4.3. Examples of minimal XPath Queries

[LibertyIDPP] describes a minimal set of XPATH queries that must be supported and also supplies mapping from discovery option keywords to such XPATH expressions. Implementers are encouraged to test all queries specified for the discovery option keywords. In addition, we provide in this section some additional XPATH expressions that are legal under the definition of minimal compliance. We expect this list to be expanded as more corner cases are uncovered. Please contact the Liberty Alliance Project (http://www.projectliberty.org).

```
/pp:PP/pp:AddressCard[pp:AddrType="urn:liberty:id-sis-pp:addrType:domicile"]
/pp:PP/pp:AddressCard[pp:AddrType="urn:liberty:id-sis-pp:addrType:domicile"]/pp:Address/C
/pp:PP/pp:MsgContact[pp:MsgTechnology="urn:liberty:id-sis-pp:msgTechnology:voice"]
```

4.4. Supported XPATH expressions for Modifies and update granularity

Effectively the minimal set of supported Modify XPATH expressions (slashed paths) defines the minimal granularity of updates that need to be supported. If a client needs to update an attribute at a finer granularity than defined here, then it should first query the element and then execute a `Modify` with queried values and the value it wants to change. It is recognized that this approach has inherent problems:

1. other update between Query and Modify: client should deal with this race condition in an implementation dependent way, such as making a second query to verify that the update succeeded or by ignoring the possibility of race.

2. Query may return incomplete data due to permissions. Presumably under these circumstances the corresponding Modify will fail for similar reasons.

Updates to the containers listed above should be atomic whenever feasible. For example, if the underlying database technology is [LDAP], it is advisable to model each of the above-listed containers as an entry so that the directory server provides atomicity of update.

5. Processing rule rationale

Note that [LibertyDST] requires multiple `Modification` elements to behave in a transactional fashion, i.e., all `Modification` elements must either succeed or fail as a group. If an implementation has difficulty guaranteeing the transactional semantics, it may be preferable to support only one `Modification` element, for which the transactional semantics are trivial.

6. Managing the Principal's Name and Identity

The Principal's name and identity are complex and sensitive information. A person often considers his or her name as *synonymous* with his or her legal identity. People want their names properly represented and recorded. As a person's name is associated with his or her legal identity, both are the subject of this section. Additionally, names and legal identities exist in both local and global contexts. A particular local context is called a "localization."

The semantics of name attributes are particularly perilous as there is great cultural variance in both appearance and legal meaning of names. Generally people will have a "common name" which will usually be understood correctly by people close to that person (e.g., the postman knows how to deliver a letter even if just common name is used), but which often is abbreviated and as such does not have legal value in some cultures. Other cultures attach legal meaning even to commonly used abbreviated names. Usually this common name is what the person prefers to be called and many cultures have nicknaming conventions that are usually used in the common names, e.g., "William" = "Bill."

The straightforward approach is to separately represent the common name and the legal name. This allows both to be entered into the system and made available to those service providers that have need for each type of name. The Principal will be able to attribute separate permissions to his legal identity as opposed to his common name.

The Liberty Personal Profiles supports a wide variety of forms of names. The structure of names varies greatly by culture (see also Section 7 on Cultural Portability, below). Many schemas unfortunately have an Anglo Saxon bias in specifying that a person must have first name, middle name, and surname. This structure simply is not valid in many cultures. It is instructive to realize that in Latin cultures people typically have several first and last names, only some of which are used in the daily life, but all of which are needed for a legal identity:

In Spain the common name is usually the user's first given name and first surname as in

```
Antonio Joscavarro Cano --> Antonio Navarro
```

In Portugal it is usually user's first given name and last surname as in

```
Zita Maria Oliveira Lopes --> Zita Lopes
```

But if her first given name is "Maria" then often the second given name is used.

```
Maria Adelaide Cordeiro Oliveira --> Adelaide Oliveira
```

or

```
--> Mª Adelaide Oliveira
```

The general conclusion is that for legal names it is impossible to find culturally neutral structure other than a single field, "legalname," where the entire legal name appears with all necessary given names and surnames. For legal identity the user's name is captured as accurately as possible using the `LegalName` attribute. The Liberty specification does not attempt to impose a structure on this name.

For the common name, Liberty presumes that it is a user preference and no legal meaning should be attached to it. In systems that have rigid first name, last name structure, people whose names do not conform to this convention frequently attempt to use an approximation (e.g., the Spanish convention discussed above). If their names can not be correctly approximated, they may mis-use the first or last name fields to concatenate additional information. For example:

```
Cristina del Amo --> FN: Cristina, SN: delAmo
```

Misuse of the specific name component attributes obviously ruins any legal value the name may have. Worse, the rigid systems do not let the humans achieve the desired result so they try to circumvent the system in more or less successful ways and may perceive the system as foreign import.

The most flexible approach is to capture the Principal's full common name as a `CN` attribute. This attribute contains both first name and surname along with any possible middle initials, generation indicators, etc. all in the form the user wants them to appear. As there is only one field where all names can be put in any order, the user has the ability to get the result he wants. For many applications CN alone is sufficient: it is appropriate way to salute the Principal in user interface and it can be used to informally refer to the Principal in interactions with other humans. It can also be used as part of mailing address or as comment field in email addresses.

In some cultures and in some applications it may be of interest to know user's given name as distinct from his surname. Such an application could be "white pages" where users are alphabetized by surname (but note that in some countries white pages are alphabetized by first name). A German e-banking application might salute a customer as "Herr Kellomäki" to avoid being too informal (compare with "Herr Sampo Kellomäki" derived from `CN`). To cater to these special cases the user's legal and common names MAY be captured using the `AnalyzedName` container which has a culture-dependent break down of the name, typically consisting of a given name, a surname, and possibly other components.

Various strategies may be used for populating the name fields. Some ID-SIS-PP providers may not wish to burden their users with separate requests for a `LegalName`, a `CN`, and a `AnalyzedName` (i.e., `FN` and `SN`). There are several reasonable several options:

1. Ask for only the `LegalName` and use it to populate the `CN`;
2. Offer the `LegalName` as the default for `CN` and allow the user customize it;
3. Ask for only the `CN` and do not populate `FN` and `SN`;
4. Ask for the `FN` and `SN` and then automatically formulate the `CN` from these (assuming they can handle all culture dependencies correctly for the customer base).

If only the `CN` is available and the `FN` and `SN` are absent, the applications should use `CN` where a given name or surname would have been needed. The application may also opt to ask the Principal for this missing information and provision it back to the ID-SIS-PP provider.

6.1. InformalName

The **InformalName** is an informal "handle" the Principal may want to be known by. This could be a screen name, but the user may also have different screen names at different services. The localized version of InformalName is **LInformatlName**. See Section 7 on Cultural Portability in this document for details on the implementation of localized values. See the [LibertyIDPP] specification for technical details.

Example: **<InformalName>**theWanderer**</InformalName>**

6.2. CommonName

The **CommonName** is the name the Principal prefers in normal situations. [LibertyIDPP] contains the details for this container. Note that all attributes may be represented in global and/or local elements.

Table 9. Contents of CommonName

Attribute	Localized	Type	Synopsis
`CN`	`LCN`	cis	Every day name in latin writing system
`AltCN`	`LAltCN`	cis	Additional every day names in latin writing system
`AnalyzedName`	n/a	(container)	Name analyzed into bits and pieces

Description

The `CN` SHOULD appear to ensure wide interoperability. At least the `CN` or `LCN` MUST appear.

Example

```
<CommonName>
   <CN>Zita Lopes</CN>
   <LCN>LKj343asas</LCN>
   <AltCN>Maria Lopes</AltCN>
   <AltCN>Zita Lopes</AltCN>
   <AnalyzedName nameScheme="">
      <PersonalTitle>Dr.</PersonalTitle>
      <FN>Zita</FN>
      <SN>Lopes</SN>
      <MN>Maria</MN>
   </AnalyzedName>
</CommonName>
```

6.2.1. AnalyzedName

The **AnalyzedName** contains elements defining the fragments that compromise the complete name. These fragments may be expressed in local and/or local representations in appropriate elements. Local elements have an "L" prefix.

Table 10. Contents of AnalyzedName

Attribute	Localized	Type	Synopsis
PersonalTitle	LPersonalTitle	cis	Personal or honorary title
FN	LFN	cis	First name, Given name
SN	LSN	cis	Surname (familyname)
MN	LMN	cis	Middle name or initial

Description

This container allows names to be analyzed to arbitrary detail. Note that often CN, which is unstructured, is more portable and preferred.

This specification does not mandate any particular schemes; however, the following elements are RECOMMENDED for use if they suit the deployment's requirements:

```
PersonalTitle and LPersonalTitle for representing the title
   FN and LFN for representing the first name(s)
   MN and LMN for representing the middle name(s) or intial(s)
   SN and LSN for representing the surname(s)
```

Deployments are encouraged to use the schema extension mechanism to add any other elements that are deemed necessary. See the Extension Mechanisms Section 1.5, above, for explanation of the extension mechanism.

Example

```
   <AnalyzedName nameScheme="">
      <PersonalTitle>Dr.</PersonalTitle>
      <FN>Zita</FN>
      <SN>Lopes</SN>
      <MN>Maria</MN>
   </AnalyzedName>
```

6.3. LegalIdentity

The **LegalIdentity** contains the elements that define the Official legal identification of the Principal. That which constitutes "Official" is not defined by the Liberty specifications, but is left to the discretion of the implementation and to the Principal.

Table 11. Contents of LegalIdentity

Attribute	Localized	Type	Synopsis
LegalName	LLegalName	cis	Full legal name in Latin writing system
AnalyzedName	n/a	(container)	Name analyzed into bits and pieces
VAT	n/a	(container)	Fiscal identification number
AltID	n/a	(container)	Other identification number(s)
DOB	n/a	date	Date of Birth
Gender	n/a	enum	Gender of the Principal
MaritalStatus	n/a	enum	Marital status such as single or married

Description

At least LegalName or LLegalName MUST appear.

Example

```
<LegalIdentity>
   <LegalName>Zita Maria Oliveira da Figueira Lopes</LegalName>
   <AnalyzedName nameScheme="">
      <PersonalTitle>Dr.</PersonalTitle>
      <FN>Zita</FN>
      <SN>Lopes</SN>
      <MN>Maria</MN>
   </AnalyzedName>
   <VAT>
      <IDValue>502677123</IDValue>
      <IDType>urn:liberty:id-sis-pp:IDType:itcif</IDType>
   </VAT>
   <AltID>
      <IDValue>502677123</IDValue>
      <IDType>urn:liberty:id-sis-pp:IDType:itcif</IDType>
   </AltID>
   <DOB>1982-04-15</DOB>
   <Gender>urn:liberty:id-sis-pp:gender:f</Gender>
   <MaritalStatus>urn:liberty:id-sis-pp:maritalstatus:divorced</MaritalStatus>
</LegalIdentity>
```

6.3.1. LegalName

The **LegalName** element contains the full legal name of the Principal in latin writing system, e.g., **<LegalName>**Zita Maria Oliveira da Figueira Lopes**</LegalName>**., Details are enumerated in the LibertyIDPP [LibertyIDPP] specification. As is true of all elements use is optional; however, use of LegalName is recommended.

LegalName is the full legal name written using Latin script. If the Principal's legal name is written using a character system other than Latin, it should appear in LLegalName and LegalName may be left unspecified. Even in these cases the LegalName may be specified if there is an official Latin transliteration (e.g., in a passport).

As an example, in Japan legal names are usually in kanji and as such should be stored in LLegalName. For the many Japanese that do not have a passport the LegalName will be left unpopulated, but those that do have a passport also have official transliterated version of their name, which may be stored in LegalName.

It is assumed that the Principal has only one official legal name. If there actually can be multiple legal names, please pick one and inform the Liberty Alliance Project about the requirement to have multiple legal names.

6.3.1.1. LLegalName

The **LLegalName** contains the full legal name in a local writing system. It may be substituted for, or used in addition to, the LegalName.

6.3.2. AnalyzedName

Table 12. AnalyzedName

Attribute	Example	Synopsis
PersonalTitle	Dr.	Personal or honorary title
FN	Zita	First name, Given name
SN	Lopes	Surname (familyname)
MN	Maria	Middle name or initial

6.3.3. VAT

The **VAT** element contains a fiscal identification number. Its use is optional.

Table 13. Contents of VAT

Attribute	Localized	Type	Synopsis
IDValue	n/a	ces	Identification number value
IDType	n/a	enum	Type of identification number stored in VAT or AltID attribute

Description

The VAT is optional, used only if permitted by national legislation. The fiscal identification number is most useful for invoicing and e-commerce (often Value Added Tax number). There can only be one VAT (this is to simplify e-commerce applications).

Example

```
<VAT>
   <IDValue>502677123</IDValue>
   <IDType>urn:liberty:id-sis-pp:IDType:itcif</IDType>
</VAT>
```

6.3.3.1. IDValue

The **IDValue** element contains an identification number value, e.g., **<IDValue>**502677123**</IDValue>**

6.3.3.2. IDType

The IDType element contains a designation of the type of identification number stored in the VAT or AltID attribute. Its use is optional. The value is a URI, e.g.,

<IDType>urn:liberty:id-sis-pp:IDType:itcif**</IDType>**

Although the semantics of VAT mandate that it should be the ID number most useful for e-commerce, it is sometimes necessary to know the exact type of id number involved, thus this attribute. This attribute can be used to select one of the AltIDs. This is an enumeration where the values are URIs to facilitate future expansion. Currently following enumerators are defined:

```
urn:liberty:id-sis-pp:IDType:ukvat
urn:liberty:id-sis-pp:IDType:itcif
urn:liberty:id-sis-pp:IDType:ptnif
urn:liberty:id-sis-pp:IDType:esnif
urn:liberty:id-sis-pp:IDType:fialv
urn:liberty:id-sis-pp:IDType:rfid
```

Additional enumerators can be defined as specified in [LibertyReg].

6.3.4. AltID

The **AltID** element contains an alternate identification number. The element may be used multiple times, each one containing one identification number. Its use is optional.

Table 14. Contents of AltID

Attribute	Localized	Type	Synopsis
`IDValue`	n/a	ces	Identification number value
`IDType`	n/a	enum	Type of identification number stored in VAT or AltID attribute

Description

There can be multiple `AltIDs`, as needed. `AltID` provides a placeholder for other ID numbers that may be needed in some countries or situations. The possible values are country-dependent, but the `IDType` element should be used to indicate the type of ID being stored. National standards bodies are encouraged to set standards regarding which IDs are held and which `IDType` designations are used. They are encouraged to communicate these through the mechanism given in [LibertyReg].

Storage of ID attributes is highly regulated in many countries. If an AP chooses to keep `AltID` attributes, the AP must implement sufficient permissions enforcement, policies, audit trail, and usage directives to ensure that `AltID` is only used for legitimate purposes.

Example

```
<AltID>
   <IDValue>502677123</IDValue>
   <IDType>urn:liberty:id-sis-pp:IDType:itcif</IDType>
</AltID>
```

6.3.5. DOB

The **DOB** element contains the Date of Birth. Its use is optional, but may be used no more than once. Example:

<**DOB**>1982-04-15</**DOB**>

In countries where national ID numbers can not be collected, the date of birth may be an important differentiating factor between people with the same name. The date of birth is expressed using the Gregorian calendar. Applications may, for display purposes, map this to other calendar systems, depending on cultural context.

Note

The DOB is stored using the Gregorian calendar. User interfaces are encouraged to compute from the Gregorian calendar to a local calendar representation of the Principal (or SP), e.g., Emperor's calendar, Muslim calendar, or Julian calendar.

6.3.6. Gender

The **Gender** element contains the gender of the Principal using an enumerated vocabulary. It may be used no more than once. Example:

<**Gender**>urn:liberty:id-sis-pp:gender:f</**Gender**>

In some cases gender can be used to differentiate between people with same name, especially in countries where national ID numbers can not be collected. Possible enumerators, which are URIs, are

```
urn:liberty:id-sis-pp:gender:m (male)
urn:liberty:id-sis-pp:gender:f (female)
```

The list is not expected to be extended, but see [LibertyReg].

6.3.7. MaritalStatus

The **MaritialStatus** element contains the marital status, such as single or married, for example:

<**MaritalStatus**>urn:liberty:id-sis-pp:maritalstatus:divorced</**MaritalStatus**>

Marital status or civil state (estado civil, siviilis_y). Not all enumerators are expected to be applicable in the registries of all countries.

Marital status is an enumerated list where the values are URIs. Following are possible enumerators:

```
urn:liberty:id-sis-pp:maritalstatus:single
urn:liberty:id-sis-pp:maritalstatus:married
urn:liberty:id-sis-pp:maritalstatus:commonlawmarriage
urn:liberty:id-sis-pp:maritalstatus:separated
urn:liberty:id-sis-pp:maritalstatus:divorced
urn:liberty:id-sis-pp:maritalstatus:widowed
urn:liberty:id-sis-pp:maritalstatus:dead
urn:liberty:id-sis-pp:maritalstatus:notapplicable
```

The list of enumerators can be extended as described in [LibertyReg].

7. Cultural Portability

An Internet environment is the underlying assumption for the systems designs; end users will venture to web sites outside their own culture and interact with other users and businesses in foreign countries. This calls for a common language. A large part of the world, but not the entire world, has standardized on the use of the Latin alphabet (character set) with some variations.

When considering character set issues, it is necessary to distinguish between what is visible to the end user and what is for computer consumption. For the latter we should use the 7bit US ASCII or stricter character set. This will simplify the programming of applications that implement these specifications.

A large proportion of the attributes are for computer consumption. This serves to discourage free form input of user preferences and other data. With constrained input, most data can be condensed to code tokens or enumerations, which may be looked up from a localization database for purposes of display. This approach greatly facilitates creation of multilingual user interfaces as the data does not have a language dependency - this is handled by the presentation layer. The localization database does not need to be standardized in the context of Liberty.

The only attributes that need to be directly visible to the end user are:

- names
- addresses of all sorts (postal, email, phone number, etc.)
- some numeric attributes representing limits, but these are a nonissue as Arabic numerals are universally used

Consideration needs to be given to the representation of names and addresses. These appear to be culture dependent. The end users attend to these and may be offended if all nuances of their mother tongue and culture are not captured properly; it behooves the implementer to execute these properly.

To support both local custom and the international interaction, names and addresses should be represented in both the local writing system and as a Latin transliteration. People living in cultures that do not use Latin alphabet are accustomed to the idea that their names and addresses need to be transliterated to Latin alphabet and many local conventions exist for achieving this. Never-the-less, it is difficult to justify to these people why they should use Latin alphabet in communications between themselves.

The default character set of the ID-SIS-PP is ISO-10646, which is consistent with XML. ISO-10646 is able to represent nearly all human languages of the world and aims at supporting all human languages of the world. The encoding is by default UTF-8. UTF-8 can represent all characters of ISO-10646 so it is sufficient, although it is not the optimal solution for some far Eastern scripts. Other encoding can be specified in the XML header. In practice using encoding other than UTF-8 may lead to interoperability problems.

For nonlocalizable attributes or Latin versions of localizable attributes, the Latin 1 character set should be used as this caters to the Americas and most of Europe without having to make compromises (e.g., accents of Spanish, Portuguese, French and German can be represented in this character set). However, for every name and address attribute a parallel version using local writing system should be provided.

The Latin versions of attributes are named with plain names. The local script versions are names with same name prefixed with an uppercase el (L). The following table summarizes the Personal Profile attributes that have local representations. Case exact strings (ces) may be evaluated with case sensitivity, hence character case should maintained in storage and transmittal. Case inexact strings (cis) may be evaluated without case sensitivity. Following this summary is a discussion of some of the major issues involved in representation for the cultural portability of Profiles data.

687 **Table 15. Global and Localized elements**

Attribute	Localized	Type	Synopsis
CN	LCN	cis	Every day name in Latin writing system
AltCN	LAltCN	cis	Additional every day names in Latin writing system
InformalName	LInformalName	cis	Screen name of the Principal
PersonalTitle	LPersonalTitle	cis	Personal or honorary title
FN	LFN	cis	First name, Given name
SN	LSN	cis	Surname (familyname)
MN	LMN	cis	Middle name
LegalName	LLegalName	cis	Full legal name
JobTitle	LJobTitle	cis	Job title
O	LO	cis	Informal name of an organization
AltO	LAltO	cis	Alternate Informal name of an organization
PostalAddress	LPostalAddress	cis	Street address
L	LL	cis	Locality or city
St	LSt	cis	State or province
Nick	LNick	cis	Nick name for identifying item in user interface
EmergencyContact	LEmergencyContact	ces	Next of kin or other person to contact if Principal has medical emergency

688 It is possible to have multiple local script versions of an attribute, each properly qualified with the XML attributes
689 `xml:lang` and `dst:script`. The local script attributes are further qualified using the XML attribute `xml:lang`
690 which indicates which writing system the attribute uses. This may be further refined with the XML attribute
691 `dst:script` which can differentiate systems if the same language can be written using two different writing system
692 (e.g., kanji and kana systems are in parallel use for Japanese). Thus a person could have her name represented
693 simultaneously in Latin alphabet, Hindi, and Chinese if she so chooses.

694 It is not advisable to create multiple instances of a localizable attribute with the same `xml:lang` and `dst:script`
695 XML attributes as this creates an ambiguity. For example, if there are multiple `LPostalAddress` lines, one should
696 use the line separation mechanism that is provided (i.e., the dollar separator) rather than create multiple instances of
697 the attribute. If multiple ambiguous instances exist, an implementation may return them in an unpredictable order.

698 The use of parallel attributes allows people to communicate in their own writing system with their countrymen while
699 simultaneously engaging in international transactions using the Latin alphabet transliterations of their names and
700 addresses.

References

Informative

[LibertyIDPP] Kellomäki, Sampo, Lockhart, Rob, eds. "Liberty ID-SIS Personal Profile Service Specification," Version 1.1, Liberty Alliance Project (29 September, 2005). *http://www.projectliberty.org/specs*

[LibertyIDWSFOverview] Tourzan, Jonathan, Koga, Yuzo, eds. "Liberty ID-WSF Web Services Framework Overview," Version 1.1, Liberty Alliance Project (14 December 2004). *http://www.projectliberty.org/specs*

[LibertyIDWSFGuide] Weitzel, David, eds. (26 April 2004). "Liberty ID-WSF Impelmentation Guideline," Draft version 1.0-08, Liberty Alliance Project *http://www.projectliberty.org/specs/*

[LibertyDST] "Liberty ID-WSF Data Services Template Specification," Version 1.1, Liberty Alliance Project (14 December 2004). *http://www.projectliberty.org/specs* Kainulainen, Jukka, Ranganathan, Aravindan, eds.

[LibertyDisco] Sergent, Jonathan, eds. "Liberty ID-WSF Discovery Service Specification," Version 1.2, Liberty Alliance Project (12 December 2004). *http://www.projectliberty.org/specs*

[LibertyProtSchema] Cantor, Scott, Kemp, John, eds. "Liberty ID-FF Protocols and Schema Specification," Version 1.2-errata-v3.0, Liberty Alliance Project (14 December 2004). *http://www.projectliberty.org/specs*

[LibertyInteract] Aarts, Robert, eds. "Liberty ID-WSF Interaction Service Specification," Version 1.1, Liberty Alliance Project (14 December 2004). *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*

[LibertyCB] Kellomäki, Sampo, eds. "Liberty ID-SIS Contact Book Service Specification," Version 1.0-10, Liberty Alliance Project (20 February, 2005). *http://www.projectliberty.org/specs*

[LibertyReg] Kemp, John, eds. "Liberty Enumeration Registry Governance," Version 1.1, Liberty Alliance Project (14 December, 2004). *http://www.projectliberty.org/specs*

[RFC2119] Bradner, S., eds. "Key words for use in RFCs to Indicate Requirement Levels," RFC 2119, The Internet Engineering Task Force (March 1997). *http://www.ietf.org/rfc/rfc2119.txt [March 1997]*.

[RFC3275] Eastlake , D., Reagle, J., Solo, D., eds. (March 2002). "(Extensible Markup Language) XML-Signature Syntax and Processing," RFC 3275, The Internet Engineering Task Force *http://www.ietf.org/rfc/rfc3270.txt [March 2002]*.

[XPATH] Clark , J., DeRose , S., eds. (16 November 1999). "XML Path Language (XPath) Version 1.0 ," Recommendation, W3C *http://www.w3.org/TR/xpath [August 2003]*.

[LDAP] Wahl, M., Howes, T., Kille, S., eds. (December 1997). "Lightweight Directory Access Protocol (Version 3), ," RFC2251, Internet Engineering Task Force *http://www.rfc-editor.org/rfc/rfc2251.txt [August 2003]*.