



# Liberty Identity Personal Profile Service Specification

Version: 1.0-18

## **Editors:**

Sampo Kellomäki, Vodafone

## **Contributors:**

Carolina Canales, Ericsson  
Andy Feng, AOL  
Ariel Gordon, France Telecom  
Vincent Guesdon, France Telecom  
Jukka Kainulainen, Nokia  
Lena Kannappan, France Telecom  
Bronislav Kavsan, RSA  
John Kemp, IEEE-ISTO  
John Linn, RSA  
David del Ser, Vodafone  
Jonathan Sergent, Sun

## **Abstract:**

The Liberty ID Personal Profile (ID-PP) is a web service, hosted by an application provider and discovered via discovery service. It offers basic profile information regarding principal, including name, legal identity, and minimal set of contact information such as legal domicile, home, and work addresses as well as 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 arbitrary other data to be included. ID-PP only stores information regarding the principal himself and does not target contact management or e-commerce applications (future contact book may address those requirements better). A typical principal has two ID-PP service instances, one for her work identity, and another for her private identity. ID-PP is an instance of data oriented (see ID-WSF Data Services Template) identity web service (see ID Web Services Framework). ID-PP, like all data services, is characterized by ability to query and update attribute data and incorporates mechanisms for access control and conveying data validation information and usage directives.

Copyright © 2003 Liberty Alliance Project

## 1 Notice

2 Copyright © 2003 ActivCard; American Express Travel Related Services; America Online, Inc.; Bank of America;  
3 Bell Canada; Cingular Wireless; Cisco Systems, Inc.; Citigroup; Communicator, Inc.; Consignia; Deloitte & Touche  
4 LLP; Earthlink, Inc.; Electronic Data Systems, Inc.; Entrust, Inc.; Ericsson; Fidelity Investments; France Telecom;  
5 Gemplus; General Motors; Hewlett-Packard Company; i2 Technologies, Inc.; Internet2; Intuit Inc.; MasterCard  
6 International; NEC Corporation; Netegrity; NeuStar; Nextel Communications; Nippon Telegraph and Telephone  
7 Company; Nokia Corporation; Novell, Inc.; NTT DoCoMo, Inc.; OneName Corporation; Openwave Systems Inc.;  
8 Phaos Technology; PricewaterhouseCoopers LLP; Register.com; RSA Security Inc; Sabre Holdings Corporation;  
9 SAP AG; SchlumbergerSema; SK Telecom; Sony Corporation; Sun Microsystems, Inc.; Trustgenix; United Airlines;  
10 VeriSign, Inc.; Visa International; Vodafone Group Plc; Wave Systems;. All rights reserved.

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

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

24  
25 Liberty Alliance Project  
26 Licensing Administrator  
27 c/o IEEE-ISTO  
28 445 Hoes Lane  
29 Piscataway, NJ 08855-1331, USA  
30 [info@projectliberty.org](mailto:info@projectliberty.org)

## Revision History

**Revision:** 00 (early draft) **Date:** 23.7.2002 Sampo Kellomäki (sampo@symlabs.com)

Created first draft

**Revision:** 01 (draft) **Date:** 5.8.2002 Sampo

Converted to formal document and fleshed out many sections.

**Revision:** 02 (draft) **Date:** 7.8.2002 Sampo

Improved formatting, added confidentiality note.

**Revision:** 03 (draft) **Date:** 20.8.2002 Sampo

Edited some typos, class -> container change, added vCard feedback

**Revision:** 04 (draft) **Date:** 29.8.2002 Sampo

29.8. conf call, discussions with David del Ser (VF).

**Revision:** 05 (draft) **Date:** 19.9.2002 Sampo

Total rewrite based on feedback from Bedford Face to Face meeting. Renamed from draft-liberty-core-profile.lyx to draft-liberty-pip-and-ab-svc.lyx. Separated many services to draft-liberty-fast-track-svc.lyx. Adopted approach of including pictures, examples, and schema fragments from generated files.

**Revision:** 06 (draft) **Date:** 29.9.2002 Sampo

Renamed from draft-liberty-pip-and-ab-svc.lyx. Incorporated feedback from Jukka Kainulainen (Nokia). Feedback from conf call. Split issues document apart per suggestion from Lena Kannappan (FT).

**Revision:** 07 (draft) **Date:** 25.11.2002 Sampo

Renamed from draft-liberty-pip-abook-svc to draft-vf-cip-cb-svc. Complete rewrite based on feed back from Kyoto and after.

**Revision:** 08 (draft) **Date:** 9.12.2002 Sampo

Fixed some typos

**Revision:** 09 (draft) **Date:** 10.12.2002 Sampo

Incorporated feedback from MRD F2F via James Vanderbeek, incl. EIP, rename back to PIP.

**Revision:** 10 (draft) **Date:** 11.12.2002 Sampo

Feedback from SVC group and Tech pleantry; pruned out all CB material, renamed to draft-liberty-pip-svc

**Revision:** 11 (draft) **Date:** 19.12.2002 Sampo

Added XPATH and keyword sections, removed homeaddresscard in favour of addresscard with type element, renamed roles to types

**Revision:** 12 (draft) **Date:** 3.2.2003 Sampo

DocBook conversion, incorporated feedback from Miami, hierarchy change

**Revision:** 13 (draft) **Date:** 9.2.2003 Sampo

Some namespace and XML schema cleanups

**Revision:** 14 (draft) **Date:** 18.2.2003 Sampo

Incorporated comments by Jukka and Andy Feng, cleaned up schema per Jonathan. Sanitized DocBook notation to conform to T. Wason's guidelines.

**Revision:** 15 (proposal) **Date:** 27.2.2003 Sampo

Improved schema such that any leaf element can appear at top level. Reword document such that XPATH queries do not return data wrapped to the containers of the context. Changed containers to use dstContainerCommonAttributes attribute group. This removes the acc and modifier XML attributes from the containers, among others. Single

PostalAddress container with dollar as separator. Selection by script or lang XML attributes dropped. Clarify computed attribute discussion to state that its just an implementation detail. The ID attribute for PIPAddressCard and MsgContact fixed by referencing attribute from DST where it is fixed. Global ID is a privacy problem, see DST. Enumerator registry issue solved by referencing a nonexistent document TBD. Added language to explain that transactional guarantee is MUST, but will effectively be achieved for minimal PIP because it can only do one modify at a time. Enhanced PIP to contain group information (community/FunctionalRole). Merged EIP to PIP as EmploymentInformation container, see draft-EIP\_high\_level\_requirements\_\_2003-02-27\_.doc. Added subaccount and msgmethod to msgcontact. Added FunctionalRole. Refined discussion of emergency contacts, moved the container to top level.

**Revision:** 16 (draft) **Date:** 10.2.2003 Sampo

Included language from Ariel Gordon regarding employment and private life identity. Reworked revision history to better conform to Liberty DocBook. Fixed leafAttributes in schema. Comments from Vincent.Guesdon@francetelecom.com. Renamed from PIP to ID-PP. Reworked explanation of AnalyzedName, added MN and LMN. Labelled cultural portability and naming attribute chapters as non-normative. Renamed EmploymentIdentity back to IDPPEmploymentIdentity and removed EIP. Removed Permissions and EmergencyContact containers. Clarified multiple localizable attributes discussion to forbid multiple instances with the same lang and script.

**Revision:** 17 **Date:** 21.3.2003 John Kemp

Formatted better

**Revision:** 18 **Date:** 12.4.2003 John Kemp

Updated legal notice; revised to use no namespace for DST

## Contents

93		
94	1. Introduction .....	7
95	1.1. Private life and work ID-PP will coexist .....	7
96	1.2. Architectural Context of the ID-PP .....	7
97	1.3. Notational conventions .....	8
98	1.4. XML document confusion (non-normative) .....	8
99	2. Generalities .....	9
100	2.1. Namespaces .....	10
101	2.2. Extension and Namespace Reservation .....	10
102	2.3. Versioning .....	11
103	2.4. Derivation from Data Services Template .....	11
104	3. Discovery and Queries .....	11
105	3.1. Rationale (non-normative) .....	12
106	3.2. Discovery Option Keywords .....	12
107	3.3. Supported XPATH expressions for queries .....	15
108	3.4. Examples of minimal XPath Queries (non-normative) .....	17
109	3.5. Supported XPATH expressions for Modifies .....	17
110	4. Processing rules and other considerations .....	18
111	4.1. Query does not have to report same data every time .....	19
112	4.2. Only one Modification needs to be supported .....	19
113	4.3. Implementation need not support entire ID-PP schema to be compliant .....	19
114	5. Cultural Portability (non-normative) .....	19
115	6. Name attributes and identity (non-normative) .....	21
116	7. Qualifying Attributes with XML Attributes .....	23
117	7.1. ID-PP specific XML attributes .....	24
118	7.2. Details of ID-PP specific XML attributes .....	24
119	7.3. Use of idpp:id XML attribute .....	25
120	8. Overview of the Personal Identity Profile Service .....	25
121	8.1. IDPP .....	26
122	8.2. CommonName .....	26
123	8.3. AnalyzedName .....	26
124	8.4. LegalIdentity .....	26
125	8.5. IDPPEmploymentIdentity .....	26
126	8.6. IDPPAddressCard .....	26
127	8.7. Address .....	27
128	8.8. MsgContact .....	27
129	8.9. Facade .....	27
130	8.10. IDPPDemographics .....	27
131	9. Containers and Attributes of the ID-PP .....	27
132	9.1. IDPP .....	28
133	9.2. InformalName .....	30
134	9.3. LInformalName .....	30
135	9.4. CommonName .....	30
136	9.5. CN .....	31
137	9.6. LCN .....	31
138	9.7. AltCN .....	32
139	9.8. LAltCN .....	32
140	9.9. AnalyzedName .....	32
141	9.10. PersonalTitle .....	33
142	9.11. LPersonalTitle .....	34
143	9.12. FN .....	34
144	9.13. LFN .....	34
145	9.14. SN .....	34

146	9.15. LSN .....	34
147	9.16. MN .....	35
148	9.17. LMN .....	35
149	9.18. LegalIdentity .....	35
150	9.19. LegalName .....	36
151	9.20. LLegalName .....	37
152	9.21. VAT .....	37
153	9.22. AltID .....	37
154	9.23. DOB .....	37
155	9.24. Gender .....	38
156	9.25. MaritalStatus .....	38
157	9.26. IDPPEmploymentIdentity .....	39
158	9.27. JobTitle .....	40
159	9.28. LJobTitle .....	40
160	9.29. IDPPAddressCard .....	41
161	9.30. IDPPAddrType .....	42
162	9.31. Address .....	42
163	9.32. PostalAddress .....	43
164	9.33. LPostalAddress .....	44
165	9.34. PostalCode .....	44
166	9.35. L .....	44
167	9.36. LL .....	45
168	9.37. St .....	45
169	9.38. LSt .....	45
170	9.39. C .....	45
171	9.40. Nick .....	45
172	9.41. LNick .....	46
173	9.42. MsgContact .....	46
174	9.43. Comment .....	48
175	9.44. MsgType .....	48
176	9.45. MsgMethod .....	49
177	9.46. MsgTechnology .....	49
178	9.47. MsgProvider .....	50
179	9.48. MsgAccount .....	50
180	9.49. MsgSubaccount .....	51
181	9.50. MsgLimit .....	51
182	9.51. Facade .....	52
183	9.52. MugShot .....	53
184	9.53. WebSite .....	53
185	9.54. NamePronounced .....	53
186	9.55. GreetSound .....	53
187	9.56. GreetMeSound .....	53
188	9.57. IDPPDemographics .....	54
189	9.58. DisplayLanguage .....	55
190	9.59. Language .....	56
191	9.60. Birthday .....	56
192	9.61. Age .....	56
193	9.62. TimeZone .....	57
194	9.63. SignKey .....	57
195	9.64. EncryptKey .....	57
196	10. XML schema for ID-PP .....	57
197	11. Normative References .....	62
198	12. Informational References .....	63
199	13. Glossary .....	64

## 1. Introduction

The ID Personal Profile is a Liberty identity service that deals with identity information regarding the principal itself, be it in private or work capacity. It does not aim to be fully generic contact book and does not pretend to address all requirements of e-commerce applications. It is chartered to be the least common denominator for holding information about principal itself. Other services, such as wallet and contact book will address specific applications in a more complete and general way.

(\*\*\* <<cip-svc-hilevel.svg>>)

### 1.1. Private life and work ID-PP will coexist

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

Under such circumstances it is natural that the consumer oriented ID-PP service providers need not hold the IDPPEmploymentIdentity container while an enterprise could provide a ID-PP service for its employees and this service would be maintained by the human resources department, potentially limiting principal's control over the data held according to the policies of the company.

A principal having two such ID-PP services would have both of them registered in the discovery service. An implementation dependent mechanism in the discovery service could allow the principal to choose on SP by SP or transaction by transaction basis which ID-PP service to use. Such choice amounts to 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-PP will most probably be surrendered voluntarily by the principal and is not likely to be validated to high standards, whereas the contents of the employee ID-PP are likely to be validated by the HR department. Thus when employee uses his work ID-PP and this is served by an attribute provider hosted by the employer, the employer is vouching for employees identity and attributes and these are secured by the digital signature of the employer (because employer runs the AP and in Liberty AP will sign the attribute response).

The private and work ID-PPs may have some of the same attributes populated, but this specification does not mandate that the values should necessarily be the same. Synchronization may exist, but there can be situations where this is not desirable and at any rate it is likely that the two are hosted by different organizations so certainly mandating synchronization is infeasible.

### 1.2. Architectural Context of the ID-PP

The Liberty Identity Personal Profile service is an instance of a data oriented identity service. The data oriented aspect means that the service aims to provide attribute data structured in containers. This same approach is used by some other Liberty services (e.g. Business Identity Profile and Contact Book) and they all share the methods and general framework as described in [ID-DST].

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-PP service must 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 [ID-WSF].

#### 1.2.1. ID-PP is an interface, no particular implementation is mandated

Although the essence of the ID-PP service is data expressed as attributes, it should be understood that the technical implementation is actually a process which handles data requests and computes responses. 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-PP often comes from backend systems that may serve other purposes as well. For example an enterprise hosting a ID-PP service for its employees may choose to use their human resources database, or the ID-PP backend may also be used by the contact book service. Such sharing of backends is considered normal practise 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-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 specified behaviour against conceptual XML document, the implementation have to behave as if the document existed, but do not necessarily have to implement it in concrete terms.

## 1.2.2. Players and compliance

The ID-PP is provided by an *attribute provider* (AP) [Overview] also sometimes called a *ID-PP provider*. The AP is a ID-WSF web service that hosts the ID-PP. ID-PP is queried or updated by a *client*, which is usually a *service provider* (SP) [Overview] acting on behalf of the *principal* [Overview]. The client is sometimes called *web services client* (WSC). [ID-WSF] provides the means by which the principal can delegate to the SP a right to invoke her ID-PP service, i.e. a service assertion. Before the SP can access the ID-PP it usually has to *discover* which AP hosts the ID-PP for the principal. This is accomplished using *discovery service* [ID-DISCO] that issues the service assertions.

This specification specifies an interface to which *implementation* and *instance* (deployment) of ID-PP service must conform. The implementation may be a software product offered by a *vendor*. Typically such product, if configured and operated correctly, can provide ID-PP service. However for an AP to be ID-PP compliant it must operate correctly a ID-PP compliant implementation. Thus ID-PP compliance testing will need to address both implementations and instances.

## 1.3. Notational conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this specification are to be interpreted as described in IETF RFC 2119 [RFC2119]: "they MUST only be used where it is actually required for interoperation or to limit behavior which has potential for causing harm (e.g., limiting retransmissions)"

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

## 1.4. XML document confusion (non-normative)

The XML documents that are specified by the schemas are most general wire representations of the responses from ID-PP service. By most general we mean that document could have that schema if all data has been provisioned and no permissions filtering occurs. After filtering what goes on wire may no longer be conformat to this schema. Thus implementors may need to tweak this schema before using it to implement services. Generally the tweaking 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 that are understood from the context. Therefore the XML schema permits any and every element of ID-PP as a top level element. This is done to ensure that on-the-wire representations can always be



287 compatible with the ID-PP schema. This does not imply that the underlying conceptual XML document could have  
288 just any element at the top level. The underlying conceptual XML document is always understood to be rooted on a  
289 single ID-PP container.

290 One more confusion is that as requests to ID-PP service are actually SOAP documents, there is one schema for the  
291 SOAP layer and another for the document that is returned inside the SOAP response. The present document does not  
292 concern itself with the SOAP schemas.

## 2. Generalities

### 2.1. Namespaces

The namespace for the ID-PP service is designated by URI

```
urn:liberty:idpp:1.0
```

This namespace is abbreviated as "idpp:" in this document. If namespace has been omitted in some part of this document, then "idpp:" should be taken to be the default namespace. The namespace URI is also used as `ServiceType` designator.

For enumerator URNs we usually drop the version number. As enumerator URNs are completely separate from XML this does not have adverse effect.

#### Table 1.

#### Referenced XML namespaces

Abbreviation	URI	Description
ds:	<a href="http://www.w3.org/2000/09/xmldsig">http://www.w3.org/2000/09/xmldsig</a>	XML DSIG [RFC3275] (for SignKey type)
xml:	<a href="http://www.w3.org/TR/REC-xml">http://www.w3.org/TR/REC-xml</a>	XML Definition [XML] (for xml:lang)
xs:	<a href="http://www.w3.org/2002/XMLSchema">http://www.w3.org/2002/XMLSchema</a>	XML Schema Definition [XSD]

Please note that the schema for this specification includes the DST schema, by use of an `<include>` statement. Elements and attributes from the DST schema file populate the `idpp` namespace.

### 2.2. Extension and Namespace Reservation

ID-PP is designed to be extensible. Extension can happen in six ways:

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

For enumerators, attribute names, and container names the extensions MUST use their own XML name space unless the extension is adopted by Liberty. The default name space of the ID-PP is reserved for use by Liberty Alliance standardization efforts.

It is expected that some extensions will eventually get standardized and move to "main stream". This will, unfortunately, create situations where same attribute may exist in some experimental namespace and in the official ID-PP namespace. Implementations SHOULD be programmed right from start to accept both variants, but MUST NOT emit attributes using the official namespace until approved. This way once an extension becomes official, it can be just "flipped on" and all attribute consumers will 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.

ID-PP elements that are enumerations use URIs as enumerators. Each element's description details the authority for adopting new official enumerators. In some cases, such as country and language codes, a well established international standards body assigns the enumerators. In other cases this specification specifies some enumerators and stipulates that Liberty Registry [LibReg] may assign additional official enumerators. At any rate, organizations and industry consortia are free to define and manage their own extensions.

For discovery option keywords, see `Option` element in [ID-DISCO], the extendibility is handled by using URIs as keywords. Organizations are encouraged to include their domain name in the keyword URI for the extended keywords they create.

All containers and elements defined in ID-PP schema have a `xs:any` element which permits arbitrary schema extension, should an implementation decide to support this. Implementation MAY support schema extension but is not required to do so. If an implementation does support schema extension then it MAY register `urn:liberty:idpp:can:extend` discovery option keyword. If an implementation does support schema extension, it is usually convenient to also register extended discovery option keywords and support a richer vocabulary of XPATH expressions.

## 2.3. Versioning

Once an attribute has been published and taken in common use, its syntax and semantics can not change. For interoperability it is extremely important that this rule is respected. If evolution in syntax or semantics is needed, new versions MUST be assigned different names, effectively creating new attributes with the new semantics, so that they do not conflict with the original attribute.

If the evolution is semantic the new attribute name should reflect the new semantics. If the changes are purely syntactical a convention of suffixing the attribute name by a number can be adopted.

## 2.4. Derivation from Data Services Template

The ID-PP service is an instance of Data Services Template [ID-DST] and all stipulations of [ID-DST] are hereby incorporated unless expressly waived or modified in this document.

The ID-PP service uses the WSDL declaration specified in [ID-DST].

## 3. Discovery and Queries

### 3.1. Rationale (non-normative)

The ID-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-PP, that a service provider sees, is bound to be incomplete because

- the instance of the ID-PP has chosen to only support some subset of ID-PP. For example a consumer oriented ID-PP service might choose to not support `IDPPEmploymentIdentity` container while a ID-PP service run by an enterprise for its employee might not support `Facade` container.
- not all information about the principal was ever provisioned
- national legislation forbids some information from being collected or shared across international boundaries
- attribute provider's policy forbids the SP from having some bits of 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 simply a myth.

It is more fruitful to approach the Personal Identity Profile from the perspective of "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 guide lines for fair information use and, I am sure, would be well appreciated by many privacy advocates.

Considering the possible applications and their minimal information needs quite naturally leads to collecting the attributes into containers that correspond to these information needs. These containers are very useful abstractions because they are an open ended mechanism 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, it is not very useful for the end users to think in terms of the individual attributes like street address, city, and state and whether some e-commerce company should be allowed to see them. It is much more meaningful to assign the permissions at the `AddressCard` level. By the same token, it is more convenient that the service providers express their information needs in terms of containers of attributes.

Attribute containers group the attributes in meaningful units, but there are quite many uninteresting small containers. Therefore the discovery service operates at granularity of a option keyword. Option keywords, which are defined in this document, are used to discover existence or support for containers or groups of containers in a way meaningful to applications. See [ID-DISCO] for generic definition of the Discovery Service and the processing rules for discovering by keyword.

### 3.2. Discovery Option Keywords

ID-PP defines a number of discovery keywords to be included as `Option` elements in discovery registrations and queries, see [ID-DISCO]. Some of these express availability of data and others the ability to update data. An attribute provider may advertise ability to update data even if it currently does not have a given data item populated for the principal. Effectively the update keywords express APs willingness to host the data.

#### 3.2.1. Data availability discovery option keywords

The data availability oriented keywords extract selected components from the profile as if the given XPATH expression was applied. The implementation does not have to use XPATH as long as the results are equivalent. Presence of the

keyword implies that there is high probability that the corresponding data can be obtained if queried. However, there is no guarantee: the data may still be unavailable due to permissions or race conditions between data removal and updates to the discovery service.

**Table 2.**

**Data availability discovery option keywords**

Keyword	Equivalent XPATHs	Meaning
urn:liberty:idpp	/idpp	Has some ID-PP data
urn:liberty:idpp:domicile	/IDPP/IDPPAddressCard [IDPPAddrType="urn:liberty:idpp:addrType:domicile"]	Has some address card data corresponding to the domicile
urn:liberty:idpp:home	/IDPP/IDPPAddressCard [IDPPAddrType="urn:liberty:idpp:addrType:home"]	Has some address card data corresponding to the home address
urn:liberty:idpp:work	/IDPP/IDPPAddressCard [IDPPAddrType="urn:liberty:idpp:addrType:work"], /IDPP/MsgContact [MsgType="urn:liberty:idpp:msgType:work"]	Has some address card or messaging contact data corresponding to the office address
urn:liberty:idpp:personal	/IDPP/IDPPMsgContact [MsgType="urn:liberty:idpp:msgType:personal"]	Has some messaging contact data corresponding to personal contact
urn:liberty:idpp:mobile	/IDPP/IDPPMsgContact [MsgType="urn:liberty:idpp:msgType:mobile"]	Has some messaging contact data for mobile contact
urn:liberty:idpp:vacation	/IDPP/IDPPMsgContact [MsgType="urn:liberty:idpp:msgType:vacation"]	Has some messaging contact data for vacation contact
urn:liberty:idpp:address	/IDPP/IDPPAddressCard	Has some address card data
urn:liberty:idpp:cn	/IDPP/CommonName	Has some common name data
urn:liberty:idpp:informalName	/IDPP/InformalName, /IDPP/LInformalName	Has informal name
urn:liberty:idpp:legal	/IDPP/LegalIdentity	Has some legal identity data
urn:liberty:idpp:employment	/IDPP/IDPPEmploymentIdentity	Has some employment identity data
urn:liberty:idpp:facade	/IDPP/Facade	Has some facade data
urn:liberty:idpp:keys	/IDPP/SignKey, /IDPP/EncryptKey	Has either or both keys
urn:liberty:idpp:demographics	/IDPP/IDPPDemographics	Has some demographics data
urn:liberty:idpp:permissions	/IDPP/Permissions	Has some permissions data
urn:liberty:idpp:emergency	/IDPP/EmergencyContact, /IDPP/IDPPAddressCard [IDPPAddrType="urn:liberty:idpp:addrType:emergency"]	Has some emergency contact date
urn:liberty:idpp:extend	n/a	Has some extended data

An attribute provider MUST NOT register a data availability discovery option keyword if it is implausible that the data will be available. For example if an AP does not yet have the data, it must not opportunistically register the keyword with hopes of being able to gather the data by the time it is requested by someone or to be able to gather it via Resource Owner Interaction protocol. An attribute provider SHOULD NOT register a keyword if the principal has set such permissions on the data that it can not be released under any plausible circumstances.

### 3.2.2. Data update discovery option keywords

The data update discovery option keywords express the willingness and ability of the attribute provider to store at least some data corresponding to the given XPATH expression. These keywords do not imply that the AP currently has any data regarding the containers referenced by the keyword.

**Table 3.**

**Data update discovery option keywords**

Keyword	Equivalent XPATHs	Meaning
urn:liberty:idpp:can	/IDPP	Can store some ID-PP data
urn:liberty:idpp:can:domicile	/IDPP/IDPPAddressCard [IDPPAddrType="urn:liberty:idpp:addrType:domicile"]	Can store some address card data corresponding to the domicile
urn:liberty:idpp:can:home	/IDPP/IDPPAddressCard [IDPPAddrType="urn:liberty:idpp:addrType:home"]	Can store some address card data corresponding to the home address
urn:liberty:idpp:can:work	/IDPP/IDPPAddressCard [IDPPAddrType="urn:liberty:idpp:addrType:work"],  /IDPP/MsgContact [MsgType="urn:liberty:idpp:msgType:work"]	Can store some address card or messaging contact data corresponding to the office address
urn:liberty:idpp:can:personal	/IDPP/IDPPMsgContact [MsgType="urn:liberty:idpp:msgType:personal"]	Can store some messaging contact data corresponding to personal contact
urn:liberty:idpp:can:mobile	/IDPP/IDPPMsgContact [MsgType="urn:liberty:idpp:msgType:mobile"]	Can store some messaging contact data for mobile contact
urn:liberty:idpp:can:vacation	/IDPP/IDPPMsgContact [MsgType="urn:liberty:idpp:msgType:vacation"]	Can store some messaging contact data for vacation contact
urn:liberty:idpp:can:address	/IDPP/IDPPAddressCard	Can store some address card data
urn:liberty:idpp:can:cn	/IDPP/CommonName	Can store some common name data
urn:liberty:idpp:can:informalName	/IDPP/InformalName, /IDPP/LInformalName	Can store informal name
urn:liberty:idpp:can:legal	/IDPP/LegalIdentity	Can store some legal identity data
urn:liberty:idpp:can:employment	/IDPP/IDPPEmploymentIdentity	Can store some employment identity data
urn:liberty:idpp:can:facade	/IDPP/Facade	Can store some facade data
urn:liberty:idpp:can:keys	/IDPP/SignKey, /IDPP/EncryptKey	Can store either or both keys
urn:liberty:idpp:can:demographics	/IDPP/IDPPDemographics	Can store some demographics data
urn:liberty:idpp:can:permissions	/IDPP/Permissions	Can store some permissions data
urn:liberty:idpp:can:emergency	/IDPP/EmergencyContact	Can store some emergency data
urn:liberty:idpp:can:extend	n/a	Can accept schema extension and corresponding data

An implementation MUST NOT register a data update discovery option keyword unless at least some Modify request regarding the data referenced by the keyword can plausibly succeed. For example if AP is read only it must not register any data update discovery option keywords. Similarly if the underlying database is incapable of storing the data, then the keyword must not be advertised. An implementation that registers a data update discovery option keyword SHOULD be capable of accepting any Modify request (subject to permissions) regarding that category of data and SHOULD support all elements specified in ID-PP schema for that category.

### 3.2.3. Other discovery option keywords

Sometimes it is convenient to use the discovery service to register capabilities and features of the AP, especially if the AP is more than minimally compliant. As some SPs might need to find an AP that supports some optinal feature, it is useful to register these features in the discovery service. Thus following additional discovery option keywords are defined:

**Table 4.**

## Other discovery option keywords

Keyword	Meaning
urn:liberty:idpp:fullXPath	Full XPATH syntax is supported
urn:liberty:idpp:multipleModification	Multiple Modification elements may appear in Modify request

### 3.2.4. Ambiguity in case of multiple AP hosting the same data

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

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

## 3.3. Supported XPATH expressions for queries

The [ID-DST] specifies `Query` element that contains potentially several `QueryItem` elements which in turn contain `Select` element, but leaves the actual select expression open. ID-PP implementation MAY support full XPATH expressions [XPATH] as select expression. If it does, it MAY advertise `urn:liberty:idpp:fullXPath` discovery option keyword. Minimally compliant implementation of ID-PP MUST support following XPATH expressions as select expressions

1. slash separated path to any depth. The path is always anchored at document root and may not contain wild cards or empty nodes. Although ID-PP may be extended, currently the complete set of all possible slashed paths is as follows

```
/IDPP
/IDPP/InformalName
/IDPP/LInformalName
/IDPP/CommonName
/IDPP/CommonName/CN
/IDPP/CommonName/LCN
/IDPP/CommonName/AltCN
/IDPP/CommonName/LAltCN
/IDPP/CommonName/AnalyzedName
/IDPP/CommonName/AnalyzedName/PersonalTitle
/IDPP/CommonName/AnalyzedName/LPersonalTitle
/IDPP/CommonName/AnalyzedName/FN
/IDPP/CommonName/AnalyzedName/LFN
/IDPP/CommonName/AnalyzedName/SN
/IDPP/CommonName/AnalyzedName/LSN
/IDPP/CommonName/AnalyzedName/MN
/IDPP/CommonName/AnalyzedName/LMN
/IDPP/LegalIdentity
/IDPP/LegalIdentity/LegalName
/IDPP/LegalIdentity/LLegalName
/IDPP/LegalIdentity/AnalyzedName
/IDPP/LegalIdentity/AnalyzedName/PersonalTitle
/IDPP/LegalIdentity/AnalyzedName/LPersonalTitle
/IDPP/LegalIdentity/AnalyzedName/FN
/IDPP/LegalIdentity/AnalyzedName/LFN
/IDPP/LegalIdentity/AnalyzedName/SN
/IDPP/LegalIdentity/AnalyzedName/LSN
/IDPP/LegalIdentity/AnalyzedName/MN
/IDPP/LegalIdentity/AnalyzedName/LMN
```

467 /IDPP/LegalIdentity/VAT  
468 /IDPP/LegalIdentity/AltID  
469 /IDPP/LegalIdentity/DOB  
470 /IDPP/LegalIdentity/Gender  
471 /IDPP/LegalIdentity/MaritalStatus  
472 /IDPP/IDPPEmploymentIdentity  
473 /IDPP/IDPPEmploymentIdentity/JobTitle  
474 /IDPP/IDPPEmploymentIdentity/LJobTitle  
475 /IDPP/IDPPEmploymentIdentity/CommonName  
476 /IDPP/IDPPEmploymentIdentity/CommonName/CN  
477 /IDPP/IDPPEmploymentIdentity/CommonName/LCN  
478 /IDPP/IDPPEmploymentIdentity/CommonName/AltCN  
479 /IDPP/IDPPEmploymentIdentity/CommonName/LAltCN  
480 /IDPP/IDPPEmploymentIdentity/CommonName/AnalyzedName  
481 /IDPP/IDPPEmploymentIdentity/CommonName/AnalyzedName/PersonalTitle  
482 /IDPP/IDPPEmploymentIdentity/CommonName/AnalyzedName/LPersonalTitle  
483 /IDPP/IDPPEmploymentIdentity/CommonName/AnalyzedName/FN  
484 /IDPP/IDPPEmploymentIdentity/CommonName/AnalyzedName/LFN  
485 /IDPP/IDPPEmploymentIdentity/CommonName/AnalyzedName/SN  
486 /IDPP/IDPPEmploymentIdentity/CommonName/AnalyzedName/LSN  
487 /IDPP/IDPPEmploymentIdentity/CommonName/AnalyzedName/MN  
488 /IDPP/IDPPEmploymentIdentity/CommonName/AnalyzedName/LMN  
489 /IDPP/IDPPAddressCard  
490 /IDPP/IDPPAddressCard/IDPPAddrType  
491 /IDPP/IDPPAddressCard/Address  
492 /IDPP/IDPPAddressCard/Address/PostalAddress  
493 /IDPP/IDPPAddressCard/Address/LPostalAddress  
494 /IDPP/IDPPAddressCard/Address/PostalCode  
495 /IDPP/IDPPAddressCard/Address/L  
496 /IDPP/IDPPAddressCard/Address/LL  
497 /IDPP/IDPPAddressCard/Address/St  
498 /IDPP/IDPPAddressCard/Address/LSt  
499 /IDPP/IDPPAddressCard/Address/C  
500 /IDPP/IDPPAddressCard/Nick  
501 /IDPP/IDPPAddressCard/LNick  
502 /IDPP/MsgContact  
503 /IDPP/MsgContact/Nick  
504 /IDPP/MsgContact/LNick  
505 /IDPP/MsgContact/Comment  
506 /IDPP/MsgContact/MsgType  
507 /IDPP/MsgContact/MsgMethod  
508 /IDPP/MsgContact/MsgTechnology  
509 /IDPP/MsgContact/MsgProvider  
510 /IDPP/MsgContact/MsgAccount  
511 /IDPP/MsgContact/MsgSubaccount  
512 /IDPP/MsgContact/MsgLimit  
513 /IDPP/Facade  
514 /IDPP/Facade/MugShot  
515 /IDPP/Facade/WebSite  
516 /IDPP/Facade/NamePronounced  
517 /IDPP/Facade/GreetSound  
518 /IDPP/Facade/GreetMeSound  
519 /IDPP/IDPPDemographics  
520 /IDPP/IDPPDemographics/DisplayLanguage  
521 /IDPP/IDPPDemographics/Language  
522 /IDPP/IDPPDemographics/Birthday  
523 /IDPP/IDPPDemographics/Age  
524 /IDPP/IDPPDemographics/TimeZone  
525 /IDPP/SignKey  
526 /IDPP/EncryptKey

Note that when using slash separated paths, only the elements  
that are inside the container selected by the path are returned.  
The result will not have the parent containers because they  
are understood from the context of the query.

## 1. Selection of IDPPAddressCard by exact match on contents of leaf element for the following leaf elements



a. idpp:IDPPAddrType

b. idpp:Nick

c. idpp:LNick

Only one of the tests needs to be supported in any one slashed path. Such bracketed expression may appear also in middle of a slashed path.

2. Selection of `MsgContact` by exact match on contents of leaf element for the following leaf elements

a. idpp:Nick

b. idpp:LNick

c. idpp:MsgTechnology

d. idpp:MsgMethod

e. idpp:MsgType

`Nick` and `LNick` need only be testable in isolation. `MsgTechnology`, `MsgMethod`, and `MsgType` can be tested in isolation or simultaneously combined with an AND operator. Other operators need not be supported. Such bracketed expression may appear also in middle of a slashed path.

3. Selection by `idpp:id` XML attribute of `IDPPAddressCard` or `MsgContact`. This test may appear in middle of a slashed path and need not be combinable with any other test.

4. Selection of `/IDPP/LegalIdentity/AltID` by exact match against XML attribute `idpp:altIDType`

XML namespaces MUST be fully supported in the XPath expressions by all implementations of ID-PP, the minimal ones included. The XML namespace mechanism provides flexibility that allows any extension attributes to coexist with standard attributes.

### 3.4. Examples of minimal XPath Queries (non-normative)

All following are legal queries:

```
/idpp:IDPP/idpp:IDPPAddressCard[idpp:IDPPAddrType="urn:liberty:idpp:addrType:domicile"]  
/idpp:IDPP/idpp:IDPPAddressCard[idpp:IDPPAddrType="urn:liberty:idpp:addrType:domicile"]/Address/C  
/idpp:IDPP/idpp:MsgContact[idpp:MsgTechnology="urn:liberty:idpp:msgTechnology:voice"]
```

### 3.5. Supported XPATH expressions for Modifies

For Modify requests, the following slashed paths MUST be supported in `Select` elements (see [ID-DST]). Implementation MAY support full XPATH. The same XPATH expressions as for querying MUST be supported.

```
/IDPP  
/IDPP/InformalName  
/IDPP/LInformalName  
/IDPP/CommonName  
/IDPP/LegalIdentity  
/IDPP/IDPPEmploymentIdentity  
/IDPP/IDPPAddressCard  
/IDPP/MsgContact  
/IDPP/Facade  
/IDPP/IDPPDemographics
```

578 /IDPP/SignKey  
579 /IDPP/EncryptKey  
580  
581

582 Effectively this minimal set of slashed paths defines the minimal granularity of updates that must be supported. If a  
583 client needs to update an attribute at a finer granularity than defined here, then it should first query the element and  
584 then make a Modify with queried values and the value it wants to change. It is recognized that this approach has  
585 inherent problems

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

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

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

## 4. Processing rules and other considerations

### 4.1. Query does not have to report same data every time

A ID-PP instance is NOT REQUIRED to report same results to two instances of the same query. A ID-PP instance SHOULD report same results to the same query made by the same client, unless an update (Modify or out-of-band) has happened meanwhile. A ID-PP instance MAY use the Resource Owner Interaction protocol [ID-ROI] or out-of-band means to determine what data to return or even to query the data on the fly.

In general it is up to the ID-PP provider, guided by its policies, the permissions the principal has set, and the interaction with the principal, to decide what data to return on query and clients should use the data based on it's semantic meaning as specified here and further qualified by the `acc` (Attribute Collection Context) XML attributes [ID-DST] that may be present in the query response. A client SHOULD NOT attempt to use ID-PP as a transparent data store as there can be multiple updates, permission, and policy reasons that break the transparency.

### 4.2. Only one Modification needs to be supported

Modify operation works as described in [ID-DST] with the additional relaxation that minially compliant ID-PP implementation MAY refuse a `Modify` request with multiple `Modification` elements as long as all processing rules specified in [ID-DST] are followed (namely the 3rd one regarding failure to support multiple `Modification` elements). Thus minimally compliant implementation does not have to support multiple `Modification` elements.

Implementations SHOULD support multiple `Modification` elements whenever feasible. If implementation does support multiple `Modification` elements it MAY register `urn:liberty:idpp:multipleModification` discovery option keyword. Note that [ID-DST] requires multiple `Modification` elements to behave transactionally, i.e. either all `Modification` elements must either succeed or fail as a group. If an implementation has difficulty in guaranteeing the transactional semantics, it may be better to only support one `Modification` element for which the transactional semantics are trivial.

Note that minimally compliant ID-PP implementation still must support multiple `QueryItem` elements as specified in [ID-DST].

### 4.3. Implementation need not support entire ID-PP schema to be compliant

As ID-PP service instances may exist for varying purposes, such as holding attributes relating private life identity or holding attributes relating to the principal acting in her job function, a minimally compliant ID-PP implementation MAY opt not to support some ID-PP containers or elements or some features. Such implementation may still be labelled as "ID-PP implementation" provided that publicly available documentation about the implementation clearly discloses which parts of the schema and which features are not supported. All other features and schema are assumed to be supported. A deployment of an implementation that is not capable of supporting the full schema SHOULD only register the discovery option keywords that accurately reflect its capabilities.

An implementation that is capable (a deployment may still administratively restrict schema and features) of supporting all of the schema and features specified in this document MAY be labelled as "full ID-PP implementation". An implementation that falls short in any feature or part of schema MUST NOT be labelled as "full ID-PP implementation".

A deployment that supports all of the schema and features specified in this document MAY be labelled as "full ID-PP deployment" or "full ID-PP service". To meet full ID-PP deployment status all of the schema and features MUST be supported for all principals wishing to use them, barring policy decision excluding some principal.

A deployment that only supports some subset of ID-PP may still be labelled as "ID-PP deployment" or "ID-PP service" provided that the deployment publicly discloses, on its web site, for example, which subset it supports.

## 5. Cultural Portability (non-normative)

When considering character set issues, we must first distinguish between what is visible to the end user and what is for computer consumption. For the latter we should stick the 7bit US ASCII or stricter character set. This will keep it simple and easy for programmers having to implement these specifications.

Large proportion of the attributes are for computer consumption. This is mainly due to the fact that we usually want to discourage free form input of user preferences and other data. With constrained input, most data can be condensed to code tokens, enumerations, which are, for purposes of display, looked up from a localization database. 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.

Only attributes that actually 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 nonissue as arabic numerals are universally used

We need to consider representation of the names and the addresses. Unfortunately the fact is that these appear to be highly culture dependent. The end users pay a lot of attention to these and may be offended if we are not able to capture all nuances of their mother tongue and culture, so it behooves to get these right.

Furthermore, we are designing for the internet, so the underlying assumption must be that 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 with some variations.

To cater for both local custom and the international interaction, names and addresses have to be represented in both the local writing system and as 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 for these people why they should use latin alphabet in communications between themselves.

Default character set of the ID-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 most optimal solution for some far eastern scripts. Other encoding can be specified in the XML header. In practise 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 for Americas and most of the 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 uppercase el (L). The local script attributes are further qualified using XML attribute `xml:lang` which indicates which writing system the attribute uses. Sometimes this is further refined with XML attribute `idpp:script` which can be used to make the difference if the same language can be written using two different writing system (e.g. kanji and kana systems are in parallel use for japanese). It is possible to have various local script versions of an attribute, each properly qualified with the XML attributes `xml:lang` and `idpp:script`. Thus a person could have her name represented simultaneously in latin alphabet, hindi, and chinese if she so chooses.

677 Note, however, that it is not advisable to create multiple instances of localizable attribute with same `xml:lang` and  
678 `idpp:script` XML attributes as this creates ambiguity as to which one to use. For example in case there are multiple  
679 `LPostalAddress` lines, one should use the line separation mechanism that is provided (i.e. the dollar separator)  
680 rather than create multiple instances of the attribute. If multiple ambiguous instances exist, an implementation may  
681 return them in an unpredictable order.

682 The parallel attributes approach allows people to communicate in their own writing system with their countrymen  
683 while simultaneously engaging in international transactions using the latin alphabet transliterations of their names and  
684 addresses.

## 6. Name attributes and identity (non-normative)

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 other people close to that person (e.g. post man knows to deliver 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 even has nicknaming conventions that are usually used in the common names, e.g. "William" == "Bill".

The cleanest approach is to explicitly and separately represent the common name and the legal name. This approach 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.

Structure of names varies greatly by culture. Many other schemas unfortunately show great anglosaxon bias in specifying that a person must have first name, middlename, 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 legal identity:

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

Antonio José Navarro 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

-> M<sup>a</sup> Adelaide Oliveira

General conclusion from this is that for legal names it is impossible to find culture neutral structure other than just "legalname" where the entire legal name appears with all necessary given names and surnames. For legal identity we capture user's name as accurately as possible using the LegalName attribute and we make no attempt to impose any structure to this name.

For common name we take the approach 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 tend to try to choose some approximation (e.g. the Spanish convention discussed above), but if their names can not be correctly approximated, they tend to get frustrated and start abusing the first or last name fields to concatenate additional information. For example:

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

Any such munging around obviously ruins any legal value the name may have. Worse, the rigid systems do not let the humans achieve the effect they want 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 `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 full ability to get the effect 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 become interesting to know user's given name distinct from his surname. A such 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 for these special cases the user's legal and common names MAY be captured using `AnalyzedName` container which has culture dependent break down of the name, typically consisting of given name and surname and possibly other components as well.

Some ID-PP providers may not wish to burden their users with asking separately for `LegalName`, `CN`, and `AnalyzedName` (i.e. `FN` and `SN`). These implementations have several options

1. ask only for `LegalName` and populate `CN` with it
2. offer `LegalName` as default for `CN` and let user customize it
3. ask only for `CN` and do not populate `FN` and `SN` at all
4. ask for `FN` and `SN` and then automatically formulate `CN` from these (assuming they can handle all culture dependencies correctly for their own customer base)

If only `CN` is available and `FN` and `SN` are absent, the applications should use `CN` where 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-PP provider.

## 7. Qualifying Attributes with XML Attributes

### 7.1. ID-PP specific XML attributes

XML attributes associated with containers of profiles are authoratively discussed in [ID-DST]. Additional ID-PP specific XML attributes are used to qualify some of the profile attributes.

**Table 5.**

**ID-PP specific XML attributes**

XML attribute	Elements that use	Syntax	Example	Description
nameScheme	AnalyzedName	enum	firstlast	Scheme for analyzing name to components
script	localizable attr.	enum	kana	Additional writing system used for localized attribute
altIDType	VAT,AltID	enum	fialv	Type of identification number stored in VAT or AltID

### 7.2. Details of ID-PP specific XML attributes

#### 7.2.1. altIDType

**Synopsis** Type of identification number stored in VAT or AltID attribute

**Enumeration** urn:liberty:idpp:altIDType:itcif, urn:liberty:idpp:altIDType:ptnif,  
urn:liberty:idpp:altIDType:esnif, urn:liberty:idpp:altIDType:fialv,  
urn:liberty:idpp:altIDType:xxx-tbd

**Example** urn:liberty:idpp:altIDType:itcif

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 enumerators are URIs to facilitate future expansion. Currently following enumerators are defined:

```
urn:liberty:idpp:altIDType:ukvat
urn:liberty:idpp:altIDType:itcif
urn:liberty:idpp:altIDType:ptnif
urn:liberty:idpp:altIDType:esnif
urn:liberty:idpp:altIDType:fialv
```

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

#### 7.2.2. nameScheme

**Synopsis** Scheme for analyzing name to components

**Enumeration** urn:liberty:idpp:nameScheme:firstlast, urn:liberty:idpp:nameScheme:firstmiddlelast

**Example** urn:liberty:idpp:nameScheme:firstlast



Schemes are culture dependent and it expected that list will be expanded. The enumerators are URIs to facilitate expansion without conflice. Currently following enumerators are defined

```
urn:liberty:idpp:nameScheme:firstlast  
urn:liberty:idpp:nameScheme:firstmiddlelast
```

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

### 7.2.3. script

Synopsis	Refinement of writing system used for localized attribute
----------	---

Enumeration	urn:liberty:idpp:script:kana, urn:liberty:idpp:script:kanji
-------------	---

Example	urn:liberty:idpp:script:kana
---------	------------------------------

In some countries more than one writing system is in use, thus lang XML-attribute alone is not sufficient. script XML-attribute is used to provide further refinement. This is an enumeration where enumerators are URIs to facilitate future expansion. Currently following enumerators are defined:

```
urn:liberty:idpp:script:kana  
urn:liberty:idpp:script:kanji
```

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

## 7.3. Use of idpp:id XML attribute

idpp:id XML attribute is generally allowed for any element, but it is mandatory for IDPPAddressCard and MsgContact containers, because there can be multiple of each as children of the IDPP container. Thus the idpp:id XML attribute provides disambiguation. It has uniqueness properties as specified in [ID-DST], i.e. unique amongst elements within the conceptual underlying XML document.

For IDPPAddressCard and MsgContact containers the client that adds these containers MUST supply idpp:id when making the Modify request. A client MAY supply a idpp:id XML attribute for any element as part of the Modify request.

If IDPPAddressCard and MsgContact containers are populated out-of-band, the idpp:id XML attribute MUST be provisioned, although this can be done in an implementation dependent way. Out-of-band provisioning MAY supply idpp:id XML attribute for any element.

Once idpp:id XML attribute has been provisioned for an element, whether via Modify or by out-of-band update, it MUST remain unaltered and be returned by queries regarding the element until another Modify or out-of-band update occurs. Modifies and out-of-band updates that do not create element, but merely modify it's contents, SHOULD not alter the idpp:id XML attribute.

## 8. Overview of the Personal Identity Profile Service

(\*\*\* <<prof/lib-IDPP-short.svg>>)

### 8.1. IDPP

Attribute	Example	Synopsis
InformalName	theWanderer	Screen name of the principal
CommonName	(container)	The way the user likes to be called in every day situations
LegalIdentity	(container)	Official legal identification of the principal
IDPPEmploymentIdentity	(container)	Minimal Employer and employment details
IDPPAddressCard	(container)	A limited address card
MsgContact	(container)	Generic phone, email, or instant messaging contact
Facade	(container)	Principal's look and sound facade
IDPPDemographics	(container)	Base level demographics used by ID-PP
SignKey		Principal's public key or certificate for signing
EncryptKey		Principal's public key or certificate for encryption

### 8.2. 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

### 8.3. AnalyzedName

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

### 8.4. 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	502677123	Fiscal identification number
AltID	12451211	Other identification number(s)
DOB	1982-04-15	Date of Birth
Gender	urn:liberty:idpp:gender:f	Gender of the principal
MaritalStatus	urn:liberty:idpp:maritalstatus:divorced	Marital status such as single or married

### 8.5. IDPPEmploymentIdentity

Attribute	Example	Synopsis
JobTitle	CIO	Job title in latin script
CommonName	(container)	Usual every day name of the company or employer

### 8.6. IDPPAddressCard

Attribute	Example	Synopsis
IDPPAddrType	urn:liberty:idpp:addrType:domicile	Marks the role of a IDPPAddressCard
Address	(container)	Commonly used bundle of postal address fields
Nick	Joe Work	Nick name for identifying item in user interface

## 8.7. Address

Attribute	Example	Synopsis
PostalAddress	c/o Carolyn Lewis\$2378 Madrona Beach Way North	Detailed local address, e.g. street or block address with house n° etc.
PostalCode	98503-2341	Postal code, such as zip code
L	Olympia	Locality or city
St	wa	State or province, if applicable
C	us	Country

## 8.8. MsgContact

Attribute	Example	Synopsis
Nick	Joe Work	Nick name for identifying item in user interface
Comment	This is very important	Private comment about a data object
MsgType	urn:liberty:idpp:msgType:mobile	Usage role of the messaging contact
MsgMethod	urn:liberty:idpp:msgMethod:im	Messaging method associated with this contact
MsgTechnology	urn:liberty:idpp: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
MsgLimit	160	Size limit for the message

## 8.9. 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

## 8.10. IDPPDemographics

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

\the\_end

## 9. Containers and Attributes of the ID-PP

### 9.1. IDPP

Synopsis ID Personal Profile

Cardinality 0-1

XML-attributes

**Table 6.**

#### Contents of IDPP

Attribute	Localized	Type	Synopsis
InformalName	LInformalName	cis	Screen name of the principal
CommonName	n/a	(container)	The way the user likes to be called in every day situations
LegalIdentity	n/a	(container)	Official legal identification of the principal
IDPPEmploymentIdentity	n/a	(container)	Minimal Employer and employment details
IDPPAddressCard	n/a	(container)	A limited address card
MsgContact	n/a	(container)	Generic phone, email, or instant messaging contact
Facade	n/a	(container)	Principal's look and sound facade
IDPPDemographics	n/a	(container)	Base level demographics used by ID-PP
SignKey	n/a	ds:KeyInfoType	Principal's public key or certificate for signing
EncryptKey	n/a	ds:KeyInfoType	Principal's public key or certificate for encryption

#### Description

This is the top level container returned by the ID-PP service. The purpose of the ID-PP is to provide registration information strictly about the principal himself. The address information is on purpose held in limited form, as "mini" contact cards, to discourage principals from using them to hold, e.g. girl friend's address.

The IDPP container is the sole top level, or root, element of the underlying conceptual XML document hosted by the ID-PP service.

#### Example

```

<IDPP>
  <InformalName>theWanderer</InformalName>
  <CommonName>
    <CN>Zita Lopes</CN>
    <LCN xml:lang=" " script="urn:liberty:idpp:script:kana">Lk343asas</LCN>
    <AltCN>Maria Lopes</AltCN>
    <AltCN>Zita Mª Lopes</AltCN>
    <AnalyzedName nameScheme="urn:liberty:idpp:nameScheme:firstlast">
      <PersonalTitle>Dr.</PersonalTitle>
      <FN>Zita</FN>
      <SN>Lopes</SN>
      <MN>Lopes</MN>
    </AnalyzedName>
  </CommonName>
  <LegalIdentity>
    <LegalName>Zita Maria Oliveira da Figueira Lopes</LegalName>
    <AnalyzedName nameScheme="urn:liberty:idpp:nameScheme:firstlast">
      <PersonalTitle>Dr.</PersonalTitle>
      <FN>Zita</FN>
      <SN>Lopes</SN>
      <MN>Lopes</MN>

```

```
867     </AnalyzedName>
868     <VAT altIDType="urn:liberty:idpp:altIDType:itcif">502677123</VAT>
869     <AltID altIDType="urn:liberty:idpp:altIDType:itcif">12451211</AltID>
870     <AltID altIDType="urn:liberty:idpp:altIDType:itcif">55312222</AltID>
871     <DOB>1982-04-15</DOB>
872     <Gender>urn:liberty:idpp:gender:f</Gender>
873     <MaritalStatus>urn:liberty:idpp:maritalstatus:divorced</MaritalStatus>
874 </LegalIdentity>
875 <IDPPEmploymentIdentity>
876   <JobTitle>CIO</JobTitle>
877   <CommonName>
878     <CN>Mercnet Corp.</CN>
879     <LCN xml:lang=" " script="urn:liberty:idpp:script:kana">LKj343asas</LCN>
880     <AltCN>Mercnet Enterprises</AltCN>
881     <AnalyzedName nameScheme="urn:liberty:idpp:nameScheme:firstlast">
882       <PersonalTitle>Dr.</PersonalTitle>
883       <FN>Zita</FN>
884       <SN>Lopes</SN>
885       <MN>Lopes</MN>
886     </AnalyzedName>
887   </CommonName>
888 </IDPPEmploymentIdentity>
889 <IDPPAddressCard>
890   <IDPPAddrType>urn:liberty:idpp:addrType:domicile</IDPPAddrType>
891   <Address>
892     <PostalAddress>c/o Carolyn Lewis$2378 Madrona Beach Way North</PostalAddress>
893     <PostalCode>98503-2341</PostalCode>
894     <L>Olympia</L>
895     <St>wa</St>
896     <C>us</C>
897   </Address>
898   <Nick>Joe Work</Nick>
899 </IDPPAddressCard>
900 <MsgContact>
901   <Nick>Joe Work</Nick>
902   <Comment>This is very important</Comment>
903   <Comment>but may change in future. (example comment)</Comment>
904   <MsgType>urn:liberty:idpp:msgType:mobile</MsgType>
905   <MsgMethod>urn:liberty:idpp:msgMethod:im</MsgMethod>
906   <MsgTechnology>urn:liberty:idpp:msgTechnology:icq</MsgTechnology>
907   <MsgProvider>AOL</MsgProvider>
908   <MsgAccount>123435234</MsgAccount>
909   <MsgSubaccount>1</MsgSubaccount>
910   <MsgLimit>160</MsgLimit>
911 </MsgContact>
912 <Facade>
913   <MugShot>http://fotoserver.com/~joedoe/face.gif</MugShot>
914   <WebSite>http://provider.com/~user</WebSite>
915   <NamePronounced>http://fotoserver.com/~joedoe/name.wav</NamePronounced>
916   <GreetSound>http://fotoserver.com/~joedoe/greet.wav</GreetSound>
917   <GreetMeSound>http://fotoserver.com/~joedoe/greetme.wav</GreetMeSound>
918 </Facade>
919 <IDPPDemographics>
920   <DisplayLanguage>pt-br</DisplayLanguage>
921   <Language>pt</Language>
922   <Language>fi</Language>
923   <Language>en</Language>
924   <Birthday>-05-09</Birthday>
925   <Age>18</Age>
926   <TimeZone>+05:00</TimeZone>
927 </IDPPDemographics>
928 </IDPP>
929
930
931
```

## XML schema

```
933
934
935 <element name="IDPP" type="idpp:IDPPType"/>
936 <complexType name="IDPPType">
937   <sequence>
938     <element ref="idpp:InformalName" minOccurs="0" maxOccurs="1"/>
939     <element ref="idpp:LInformalName" minOccurs="0" maxOccurs="10"/>
940     <element ref="idpp:CommonName" minOccurs="0" maxOccurs="1"/>
```

```

941 <element ref="idpp:LegalIdentity" minOccurs="0" maxOccurs="1"/>
942 <element ref="idpp:IDPPEmploymentIdentity" minOccurs="0" maxOccurs="1"/>
943 <element ref="idpp:IDPPAddressCard" minOccurs="0" maxOccurs="10"/>
944 <element ref="idpp:MsgContact" minOccurs="0" maxOccurs="30"/>
945 <element ref="idpp:Facade" minOccurs="0" maxOccurs="1"/>
946 <element ref="idpp:IDPPDemographics" minOccurs="0" maxOccurs="1"/>
947 <element ref="idpp:SignKey" minOccurs="0" maxOccurs="1"/>
948 <element ref="idpp:EncryptKey" minOccurs="0" maxOccurs="1"/>
949 <any namespace="##other" processContents="skip" minOccurs="0"/>
950 </sequence>
951 <attributeGroup ref="idpp:commonAttributes"/>
952 </complexType>
953
954
955

```

## 9.2. InformalName

Synopsis	Screen name of the principal
Data type	cis
Cardinality	0-1
Example	<b>&lt;InformalName&gt;</b> theWanderer <b>&lt;/InformalName&gt;</b>

An informal "handle" the principal may want to be known by. Often this could be a screen name, but note that user may also have different screen names at different services.

## 9.3. LInformalName

Synopsis	Localized screen name of the principal
Data type	cis
Cardinality	0-10
Example	

## 9.4. CommonName

Synopsis	The way the user likes to be called in every day situations
Cardinality	0-1
XML-attributes	

## Table 7.

### 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

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

### Example

```
<CommonName>
  <CN>Mercnet Corp.</CN>
  <LCN xml:lang=" " script="urn:liberty:idpp:script:kana">LKj343asas</LCN>
  <AltCN>Mercnet Enterprises</AltCN>
  <AnalyzedName nameScheme="urn:liberty:idpp:nameScheme:firstlast">
    <PersonalTitle>Dr.</PersonalTitle>
    <FN>Zita</FN>
    <SN>Lopes</SN>
    <MN>Lopes</MN>
  </AnalyzedName>
</CommonName>
```

### XML schema

```
<element name="CommonName" type="idpp:CommonNameType"/>
<complexType name="CommonNameType">
  <sequence>
    <element ref="idpp:CN" minOccurs="0" maxOccurs="1"/>
    <element ref="idpp:LCN" minOccurs="0" maxOccurs="10"/>
    <element ref="idpp:AltCN" minOccurs="0" maxOccurs="10"/>
    <element ref="idpp:LAltCN" minOccurs="0" maxOccurs="10"/>
    <element ref="idpp:AnalyzedName" minOccurs="0" maxOccurs="1"/>
    <any namespace="##other" processContents="skip" minOccurs="0"/>
  </sequence>
  <attributeGroup ref="idpp:commonAttributes"/>
</complexType>
```

## 9.5. CN

Synopsis	Every day name in latin writing system
Data type	cis
Cardinality	0-1
Example	<CN>Zita Lopes</CN>

cn specifies the preferred every day name of the principal, often concatenation of preferred first and surname. The principal may want to be known under other names as well, e.g. for purposes of a white pages listing. These are covered by AltCN and LAltCN.

## 9.6. LCN

Synopsis	Every day name in local writing system
Data type	cis
Cardinality	0-10
Example	<LCN>LKj343asas</LCN>

## 9.7. AltCN

Synopsis	Additional every day names in latin writing system
Data type	cis
Cardinality	0-10
Example	<AltCN>Maria Lopes</AltCN>

<AltCN>Zita M<sup>a</sup> Lopes</AltCN>

User's preferred every day name is CN, but she may want to be known under other names as well, e.g. for white pages listing, thus we capture those names here. AltCN is optional.

## 9.8. LAltCN

Synopsis	Additional every day names in local writing system
Data type	cis
Cardinality	0-10
Example	

## 9.9. AnalyzedName

Synopsis	Name analyzed into bits and pieces
Cardinality	0-1
XML-attributes	nameScheme



**Table 8.**

**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 intial

## 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, but following elements are RECOMMENDED to be used:

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)

if they fit the deployment's requirements. Deployments are encouraged to use the schema extension mechanism to add any other elements that are deemed necessary.

## Example

```
<AnalyzedName nameScheme="urn:liberty:idpp:nameScheme:firstlast">
  <PersonalTitle>Dr.</PersonalTitle>
  <FN>Zita</FN>
  <SN>Lopes</SN>
  <MN>Lopes</MN>
</AnalyzedName>
```

## XML schema

```
<element name="AnalyzedName" type="idpp:AnalyzedNameType"/>
<complexType name="AnalyzedNameType">
  <sequence>
    <element ref="idpp:PersonalTitle" minOccurs="0" maxOccurs="1"/>
    <element ref="idpp:LPersonalTitle" minOccurs="0" maxOccurs="10"/>
    <element ref="idpp:FN" minOccurs="0" maxOccurs="1"/>
    <element ref="idpp:LFN" minOccurs="0" maxOccurs="10"/>
    <element ref="idpp:SN" minOccurs="0" maxOccurs="1"/>
    <element ref="idpp:LSN" minOccurs="0" maxOccurs="10"/>
    <element ref="idpp:MN" minOccurs="0" maxOccurs="1"/>
    <element ref="idpp:LMN" minOccurs="0" maxOccurs="10"/>
    <any namespace="##other" processContents="skip" minOccurs="0"/>
  </sequence>
  <attribute name="nameScheme" type="idpp:URI" use="optional"/>
  <attributeGroup ref="idpp:commonAttributes"/>
</complexType>
```

## 9.10. PersonalTitle

1086	Synopsis	Personal or honorary title
1087	Data type	cis
1088	Cardinality	0-1
1089	Example	<PersonalTitle>Dr.</PersonalTitle>

## 1090 9.11. LPersonalTitle

1091	Synopsis	Personal title in local writing system
1092	Data type	cis
1093	Cardinality	0-10
1094	Example	

## 1095 9.12. FN

1096	Synopsis	First name, Given name
1097	Data type	cis
1098	Cardinality	0-1
1099	Example	<FN>Zita</FN>

## 1100 9.13. LFN

1101	Synopsis	First name in local writing system
1102	Data type	cis
1103	Cardinality	0-10
1104	Example	

## 1105 9.14. SN

1106	Synopsis	Surname (familyname)
1107	Data type	cis
1108	Cardinality	0-1
1109	Example	<SN>Lopes</SN>

## 9.15. LSN

1110

1111	Synopsis	Surname in local writing system
1112	Data type	cis
1113	Cardinality	0-10
1114	Example	

## 9.16. MN

1115

1116	Synopsis	Middle name or intial
1117	Data type	cis
1118	Cardinality	0-1
1119	Example	<MN>Lopes</MN>

## 9.17. LMN

1120

1121	Synopsis	Middle name in local writing system
1122	Data type	cis
1123	Cardinality	0-10
1124	Example	

## 9.18. LegalIdentity

1125

1126	Synopsis	Official legal identification of the principal
1127	Cardinality	0-1
1128	XML-attributes	

**Table 9.**

**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	ces	Fiscal identification number
AltID	n/a	altid	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="urn:liberty:idpp:nameScheme:firstlast">
    <PersonalTitle>Dr.</PersonalTitle>
    <FN>Zita</FN>
    <SN>Lopes</SN>
    <MN>Lopes</MN>
  </AnalyzedName>
  <VAT altIDType="urn:liberty:idpp:altIDType:itcif">502677123</VAT>
  <AltID altIDType="urn:liberty:idpp:altIDType:itcif">12451211</AltID>
  <AltID altIDType="urn:liberty:idpp:altIDType:itcif">55312222</AltID>
  <DOB>1982-04-15</DOB>
  <Gender>urn:liberty:idpp:gender:f</Gender>
  <MaritalStatus>urn:liberty:idpp:maritalstatus:divorced</MaritalStatus>
</LegalIdentity>

```

**XML schema**

```

<element name="LegalIdentity" type="idpp:LegalIdentityType"/>
<complexType name="LegalIdentityType">
  <sequence>
    <element ref="idpp:LegalName" minOccurs="0" maxOccurs="1"/>
    <element ref="idpp:LLegalName" minOccurs="0" maxOccurs="10"/>
    <element ref="idpp:AnalyzedName" minOccurs="0" maxOccurs="1"/>
    <element ref="idpp:VAT" minOccurs="0" maxOccurs="1"/>
    <element ref="idpp:AltID" minOccurs="0" maxOccurs="unbounded"/>
    <element ref="idpp:DOB" minOccurs="0" maxOccurs="1"/>
    <element ref="idpp:Gender" minOccurs="0" maxOccurs="1"/>
    <element ref="idpp:MaritalStatus" minOccurs="0" maxOccurs="1"/>
    <any namespace="##other" processContents="skip" minOccurs="0"/>
  </sequence>
  <attributeGroup ref="idpp:commonAttributes"/>
</complexType>

```

## 9.19. LegalName

**Synopsis**

Full legal name in latin writing system

1175	Data type	cis
1176	Cardinality	0-1
1177	Example	<b>&lt;LegalName&gt;</b> Zita Maria Oliveira da Figueira Lopes <b>&lt;/LegalName&gt;</b>
1178	Full legal name written using latin script. If principal's legal name is written using system other than latin, it should	
1179	appear in llegalname and legalname may be left unspecified. Even in these cases the legalname may be specified if	
1180	there is an official latin transliteration (e.g. in passport).	
1181	For example, in Japan legal names are usually in kanji and as such should be stored in llegalname. For the many	
1182	Japanese that do not have a passport the legalname is left unpopulated, but those that do have passport also have	
1183	official transliterated version of their name, which may be stored in legalname.	
1184	The assumption is that the principal only has one official legal name. If there actually can be multiple, please pick one	
1185	and inform the maintainer, Sampo Kellomäki <b>&lt;sampo@symlabs.com&gt;</b> , about the requirement to have multiple legal	
1186	names.	

## 1187 9.20. LLegalName

1188	Synopsis	Full legal name in local writing system
1189	Data type	cis
1190	Cardinality	0-10
1191	Example	

## 1192 9.21. VAT

1193	Synopsis	Fiscal identification number
1194	Data type	ces
1195	Cardinality	0-1
1196	Example	<b>&lt;VAT&gt;</b> 502677123 <b>&lt;/VAT&gt;</b>
1197	Optional, only if permitted by national legislation. Fiscal identification number most useful for invoicing and e-	
1198	commerce (often Value Added Tax number). There can only be one vat (this is to simplify e-commerce apps).	

## 1199 9.22. AltID

1200	Synopsis	Other identification number(s)
1201	Data type	altid
1202	Cardinality	0-n
1203	Example	<b>&lt;AltID&gt;</b> 12451211 <b>&lt;/AltID&gt;</b>

**<AltID>55312222</AltID>**

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 is country dependent, but the altIDType XML attribute 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 altIDType designations are used. They are encouraged to communicate these to [LibReg].

## 9.23. DOB

Synopsis	Date of Birth
Data type	date
Cardinality	0-1
Example	<b>&lt;DOB&gt;1982-04-15&lt;/DOB&gt;</b>

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

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

## 9.24. Gender

Synopsis	Gender of the principal
Enumeration	urn:liberty:idpp:gender:m, urn:liberty:idpp:gender:f
Cardinality	0-1
Example	<b>&lt;Gender&gt;urn:liberty:idpp:gender:f&lt;/Gender&gt;</b>

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:idpp:gender:m (male)  
urn:liberty:idpp:gender:f (female)

List is not expected to be extended, but see [LibReg].

## 9.25. MaritalStatus

Synopsis	Marital status such as single or married
Enumeration	urn:liberty:idpp:maritalstatus:single, urn:liberty:idpp:maritalstatus:married, urn:liberty:idpp:maritalstatus:commonlawmarriage, urn:liberty:idpp:maritalstatus:separated, urn:liberty:idpp:maritalstatus:divorced, urn:liberty:idpp:maritalstatus:widowed, urn:liberty:idpp:maritalstatus:dead, urn:liberty:idpp:maritalstatus:notapplicable

1238	Cardinality	0-1
1239	Example	<b>&lt;MaritalStatus&gt;</b> urn:liberty:idpp:maritalstatus:divorced <b>&lt;/MaritalStatus&gt;</b>
1240	Marital status or civil state (estado civil, siviilisääty). Not all enuerators are expected to be applicable in registries of	
1241	all countries.	
1242	This is an enumeration where enumerators are URIs. Following are possible enumerators:	
1243	urn:liberty:idpp:maritalstatus:single	
1244	urn:liberty:idpp:maritalstatus:married	
1245	urn:liberty:idpp:maritalstatus:commonlawmarriage	
1246	urn:liberty:idpp:maritalstatus:separated	
1247	urn:liberty:idpp:maritalstatus:divorced	
1248	urn:liberty:idpp:maritalstatus:widowed	
1249	urn:liberty:idpp:maritalstatus:dead	
1250	urn:liberty:idpp:maritalstatus:notapplicable	
1251		
1252		
1253	The list of enumerators can be extended as described in [LibReg].	

## 1254 9.26. IDPPEmploymentIdentity

1255	Synopsis	Minimal Employer and employment details
1256	Cardinality	0-1
1257	XML-attributes	

**Table 10.**

**Contents of IDPPEmploymentIdentity**

Attribute	Localized	Type	Synopsis
JobTitle	LJobTitle	cis	Job title in latin script
CommonName	n/a	(container)	The way the user likes to be called in every day situations

**Description**

Very minimal employment details, no EIP derived attributes. N.B. CommonName here refers to the doing-business-as name of the employer

**Example**

```
<IDPPEmploymentIdentity>
  <JobTitle>CIO</JobTitle>
  <CommonName>
    <CN>Mercnet Corp.</CN>
    <LCN xml:lang=" " script="urn:liberty:idpp:script:kana">LKj343asas</LCN>
    <AltCN>Mercnet Enterprises</AltCN>
    <AnalyzedName nameScheme="urn:liberty:idpp:nameScheme:firstlast">
      <PersonalTitle>Dr.</PersonalTitle>
      <FN>Zita</FN>
      <SN>Lopes</SN>
      <MN>Lopes</MN>
    </AnalyzedName>
  </CommonName>
</IDPPEmploymentIdentity>
```

**XML schema**

```
<element name="IDPPEmploymentIdentity" type="idpp:IDPPEmploymentIdentityType"/>
<complexType name="IDPPEmploymentIdentityType">
  <sequence>
    <element ref="idpp:JobTitle" minOccurs="0" maxOccurs="1"/>
    <element ref="idpp:LJobTitle" minOccurs="0" maxOccurs="10"/>
    <element ref="idpp:CommonName" minOccurs="0" maxOccurs="1"/>
    <any namespace="##other" processContents="skip" minOccurs="0"/>
  </sequence>
  <attributeGroup ref="idpp:commonAttributes"/>
</complexType>
```

**9.27. JobTitle**

Synopsis	Job title in latin script
Data type	cis
Cardinality	0-1
Example	<b>&lt;JobTitle&gt;CIO&lt;/JobTitle&gt;</b>



## 9.28. LJobTitle

Synopsis	Job title in local writing system
Data type	cis
Cardinality	0-10
Example	

## 9.29. IDPPAddressCard

Synopsis	A limited address card
Cardinality	0-10
XML-attributes	

### Table 11.

#### Contents of IDPPAddressCard

Attribute	Localized	Type	Synopsis
IDPPAddrType	n/a	enum	Marks the role of a IDPPAddressCard
Address	n/a	(container)	Commonly used bundle of postal address fields
Nick	Nick	cis	Nick name for identifying item in user interface

### Description

### Example

```
<IDPPAddressCard>
  <IDPPAddrType>urn:liberty:idpp:addrType:domicile</IDPPAddrType>
  <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>
</IDPPAddressCard>
```

### XML schema

```
<element name="IDPPAddressCard" type="idpp:IDPPAddressCardType"/>
<complexType name="IDPPAddressCardType">
  <sequence>
    <element ref="idpp:IDPPAddrType" minOccurs="0" maxOccurs="10"/>
    <element ref="idpp:Address" minOccurs="0" maxOccurs="1"/>
    <element ref="idpp:Nick" minOccurs="0" maxOccurs="1"/>
    <element ref="idpp:LNick" minOccurs="0" maxOccurs="10"/>
    <any namespace="##other" processContents="skip" minOccurs="0"/>
  </sequence>
```

```
1341     <attribute ref="idpp:id" />
1342     <attributeGroup ref="idpp:commonAttributes" />
1343 </complexType>
1344
1345
1346
```

## 9.30. IDPPAddrType

Synopsis	Marks the role of a IDPPAddressCard
Enumeration	urn:liberty:idpp:addrType:domicile, urn:liberty:idpp:addrType:home, urn:liberty:idpp:addrType:work
Cardinality	0-10
Example	<b>&lt;IDPPAddrType&gt;</b> urn:liberty:idpp:addrType:domicile <b>&lt;/IDPPAddrType&gt;</b>

Describes what is the role or the relation of the idppaddresscard to the principal.

More than one IDPPAddressType may appear if the same address can serve multiple functions. The type enumerators, which are URIs for extensibility, have following meaning

```
urn:liberty:idpp:addrType:domicile legal residence
urn:liberty:idpp:addrType:home everyday home
urn:liberty:idpp:addrType:work work address, the office where the person works
urn:liberty:idpp:addrType:emergency Emergency contact, see also EmergencyContact
```

N.B. It is perfectly possible for a IDPPAddressCard not to have any type.

The number of IDPPAddrTypes has been limited to bare minimum needed to convey the legal aspects an address may have. See AddressCard and AddressType for a more complete and flexible definition that is likely to address wider set of application needs.

## 9.31. Address

Synopsis	Commonly used bundle of postal address fields
Cardinality	0-1
XML-attributes	

**Table 12.**

## Contents of Address

Attribute	Localized	Type	Synopsis
PostalAddress	LPostalAddress	cis	Detailed local address, e.g. street or block address with house n° etc.
PostalCode	n/a	cis	Postal code, such as zip code
L	LL	cis	Locality or city
St	LSt	cis	State or province, if applicable
C	n/a	ISO 3166 country code	Country

## Description

Address container essentially represents a customary snail mail postal address. Obviously there are thousands of standards in this area, but we have opted to only represent the well standardized components in structured form while keeping the rest in "blob" form in `PostalAddress` element.

It is expected that at times knowing only country and zip code would be sufficient, but no provision for this has been made at the data model level. Our recommendation is that AP implements appropriate policies to only give out this information when nothing more is needed ("need to know" principle).

## Example

```

<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>

```

## XML schema

```

<element name="Address" type="idpp:AddressType"/>
<complexType name="AddressType">
  <sequence>
    <element ref="idpp:PostalAddress" minOccurs="0" maxOccurs="1"/>
    <element ref="idpp:LPostalAddress" minOccurs="0" maxOccurs="10"/>
    <element ref="idpp:PostalCode" minOccurs="0" maxOccurs="1"/>
    <element ref="idpp:L" minOccurs="0" maxOccurs="1"/>
    <element ref="idpp:LL" minOccurs="0" maxOccurs="10"/>
    <element ref="idpp:St" minOccurs="0" maxOccurs="1"/>
    <element ref="idpp:LSt" minOccurs="0" maxOccurs="10"/>
    <element ref="idpp:C" minOccurs="0" maxOccurs="1"/>
    <any namespace="##other" processContents="skip" minOccurs="0"/>
  </sequence>
  <attributeGroup ref="idpp:commonAttributes"/>
</complexType>

```

## 9.32. PostalAddress

### Synopsis

Detailed local address, e.g. street or block address with house n° etc.

1413	Data type	cis
1414	Cardinality	0-1
1415	Example	<PostalAddress>c/o Carolyn Lewis\$2378 Madrona Beach Way
1416		North</PostalAddress>

1417 PostalAddress represents the detailed local address component of a snail mail postal address. Depending on local  
1418 convention this usually includes street name, house number, apartment or suite letter or number, etc. In Japan, China,  
1419 Korea, and possibly other countries, a block address may appear instead. It also includes building name and may  
1420 include willage or suburb name depending on local convention. PostalAddress includes any mailstops or department  
1421 designators within an enterprise or may include "care of" specification. In countries where postal addresses are  
1422 represented as "turn left after the second maple tree" any such descriptions are included as lines of postaladdress.  
1423 If the address is a post office box (e.g. PO Box or postfach), then postaladdress is used to indicate the post office box.

1424 Multiline address is represented by using dollar sign ("\$\$") to mark the line breaks.

### 1425 9.33. LPostalAddress

1426	Synopsis	Street address in local writing system
1427	Data type	cis
1428	Cardinality	0-10
1429	Example	

### 1430 9.34. PostalCode

1431	Synopsis	Postal code, such as zip code
1432	Data type	cis
1433	Cardinality	0-1
1434	Example	<PostalCode>98503-2341</PostalCode>

1435 Although some countries, especially those in the British Common Wealth, have adopted postal codes that include latin  
1436 alphabet letters, most (all?) of the rest of the world represents postal codes as arabic numerals plus punctuation and  
1437 whitespace. Thus we feel that this attribute does not need to be localizable. It is inherent in the international postal  
1438 code system that standardization was a goal, so we feel that risk of having missed a requirement is low.

### 1439 9.35. L

1440	Synopsis	Locality or city
1441	Data type	cis
1442	Cardinality	0-1
1443	Example	<L>Olympia</L>

There is a gray line between suburb, locality, region, province, and state. Generally the suburb or building designation belongs in one of the postaladdress elements while the region or province may appear in either locality or st depending on local convention. For example in Spain locality might be

<L>Torrejon de Ardoz (MADRID)</L>

because the the local convention is to include the "communidad" in parenthesis as part of the locality name, while in US the standard way of approaching this is

<L>Olympia</L>  
<St>WA</St>

## 9.36. LL

Synopsis	Locality or city in local writing system
Data type	cis
Cardinality	0-10
Example	

## 9.37. St

Synopsis	State or province, if applicable
Data type	cis
Cardinality	0-1
Example	<St>wa</St>

Please see discussion of different granularities of locality in description of L.

## 9.38. LSt

Synopsis	State or province in local writing system
Data type	cis
Cardinality	0-10
Example	

## 9.39. C

Synopsis	Country
----------	---------

1474	Data type	ISO 3166 country code
1475	Cardinality	0-1
1476	Example	<C>us</C>

## 1477 **9.40. Nick**

1478	Synopsis	Nick name for identifying item in user interface
1479	Data type	cis
1480	Cardinality	0-1
1481	Example	<Nick>Joe Work</Nick>

1482 The nick name is intended to support identifying persons and address cards in user interfaces presented to the principal.  
1483 Typical use would be to present the nick names of all addresscards as a list for the principal to choose a shipping address  
1484 from. It is inappropriate to use nick names beyond the user interface context. For example nick name **SHOULD NOT**  
1485 be printed in address label. The nick name can not and **SHOULD NOT** be used as machine readable identification of  
1486 any data item as they are not guaranteed to be unique and may not even exist (if the user did not want to supply any).

1487 Nick corresponds by and large to the "Display As" field of Microsoft Outlook.

## 1488 **9.41. LNick**

1489	Synopsis	Local script version of nick name
1490	Data type	cis
1491	Cardinality	0-10
1492	Example	

## 1493 **9.42. MsgContact**

1494	Synopsis	Generic phone, email, or instant messaging contact
1495	Cardinality	0-30
1496	XML-attributes	

**Table 13.**

**Contents of MsgContact**

Attribute	Localized	Type	Synopsis
Nick	Nick	cis	Nick name for identifying item in user interface
Comment	n/a	cis	Private comment about a data object
MsgType	n/a	enum	Usage role of the messaging contact
MsgMethod	n/a	URI	Messaging method associated with this contact
MsgTechnology	n/a	enum	Messaging technology or protocol associated with this contact
MsgProvider	n/a	ces	Service provider or domain that provides messaging services
MsgAccount	n/a	ces	Account or address information within messaging provider
MsgSubaccount	n/a	ces	Subaccount within messaging account, such as voice mail box under phone number
MsgLimit	n/a	integer	Size limit for the message

**Description**

Principal may have various messaging addresses. Liberty has decided to be technology independent, thus we represent each messaging address as triplet of technology, provider, and account. We feel this is the most generic approach as it allows evolution in technologies (e.g. new instant messaging protocols) and evolution in the market place (e.g. new providers).

As people use their messaging contacts in different ways and as it is at least broadly necessary to contact the user by appropriate means, we have also included MsgType indication which, being an enum, lends itself easily to automatic interpretation.

For purposes of user interfaces we also optionally include Nick which allows identification of a given messaging address in a technology independent way. The Nick works much the same way as Microsoft Outlook's "Display As" field. The Comment allows principal to remember something about the contact.

Typical queries against MsgContact include (but are not limited to)

1. ~MsgType="urn:liberty:idpp:msgType:personal"~ to get all personal life messaging contacts irrespective of technology.
2. ~MsgMethod="urn:liberty:idpp:msgMethod:voice"~ to get all voice messaging contacts. This includes both work and home phones be they POTS or VoIP
3. ~MsgMethod="urn:liberty:idpp:msgMethod:im" && MsgType="urn:liberty:idpp:msgType:work"~ to get all work instant messaging contacts irrespective of technology
4. ~MsgTechnology="urn:liberty:idpp:msgTechnology:icq" && MsgType="urn:liberty:idpp:msgType:work"~ to get all work instant messaging contacts using ICQ technology (here ICQ implies IM)

## Example

```
<MsgContact>
  <Nick>Joe Work</Nick>
  <Comment>This is very important</Comment>
  <Comment>but may change in future. (example comment)</Comment>
  <MsgType>urn:liberty:idpp:msgType:mobile</MsgType>
  <MsgMethod>urn:liberty:idpp:msgMethod:im</MsgMethod>
  <MsgTechnology>urn:liberty:idpp:msgTechnology:icq</MsgTechnology>
  <MsgProvider>AOL</MsgProvider>
  <MsgAccount>123435234</MsgAccount>
  <MsgSubaccount>1</MsgSubaccount>
  <MsgLimit>160</MsgLimit>
</MsgContact>
```

## XML schema

```
<element name="MsgContact" type="idpp:MsgContactType"/>
<complexType name="MsgContactType">
  <sequence>
    <element ref="idpp:Nick" minOccurs="0" maxOccurs="1"/>
    <element ref="idpp:LNick" minOccurs="0" maxOccurs="10"/>
    <element ref="idpp:Comment" minOccurs="0" maxOccurs="1"/>
    <element ref="idpp:MsgType" minOccurs="0" maxOccurs="1"/>
    <element ref="idpp:MsgMethod" minOccurs="1" maxOccurs="1"/>
    <element ref="idpp:MsgTechnology" minOccurs="1" maxOccurs="1"/>
    <element ref="idpp:MsgProvider" minOccurs="0" maxOccurs="1"/>
    <element ref="idpp:MsgAccount" minOccurs="1" maxOccurs="1"/>
    <element ref="idpp:MsgSubaccount" minOccurs="0" maxOccurs="1"/>
    <element ref="idpp:MsgLimit" minOccurs="0" maxOccurs="1"/>
    <any namespace="##other" processContents="skip" minOccurs="0"/>
  </sequence>
  <attributeGroup ref="idpp:commonAttributes"/>
</complexType>
```

## 9.43. Comment

Synopsis	Private comment about a data object
Data type	cis
Cardinality	0-1
Example	<Comment>This is very important</Comment>



**<Comment>**but may change in future. (example comment)**</Comment>**

Comments are free format and exists for the sole benefit of the principal. Although principal may use a comment to remember something about a data object in a unstructured way, comments are not meant to be an extension mechanism. Service providers and applications that need to store additional data are encouraged to use the schema extension mechanisms to store their additional data in attributes created for that purpose.

N.B. As the comment is private to the principal, it may be written using whatever writing system the principal chooses and feels comfortable with. Therefore there is no need for LComment attribute.

## 9.44. MsgType

**Synopsis** Usage role of the messaging contact

**Enumeration** urn:liberty:idpp:msgType:personal, urn:liberty:idpp:msgType:work,  
urn:liberty:idpp:msgType:mobile, urn:liberty:idpp:msgType:vacation

**Cardinality** 0-1

**Example** **<MsgType>**urn:liberty:idpp:msgType:mobile**</MsgType>**

The principal role for which the messaging contact is used. Generally people make distinction between their working and private lives. The mobile type represents (potentially expensive) ubiquitous contactability.

This is an enumeration where enumerators are URIs for extendibility. Additional enumerators can be assigned as specified in [LibReg].

Currently following enumerators are specified:

```
urn:liberty:idpp:msgType:personal
urn:liberty:idpp:msgType:work
urn:liberty:idpp:msgType:mobile
urn:liberty:idpp:msgType:vacation
```

## 9.45. MsgMethod

**Synopsis** Messaging method associated with this contact

**Data type** URI

**Cardinality** 1

**Example** **<MsgMethod>**urn:liberty:idpp:msgMethod:im**</MsgMethod>**

Technology or protocol identifier of a messaging address. Although this is an enumeration, the list of enumerators is expected to evolve over time. Therefore enumerators are URIs and the list of enumerators may be extended as specified in [LibReg]. Currently following enumerators are specified:

```
urn:liberty:idpp:msgMethod:voice
urn:liberty:idpp:msgMethod:fax
urn:liberty:idpp:msgMethod:email
urn:liberty:idpp:msgMethod:pager
urn:liberty:idpp:msgMethod:im
```

## 9.46. MsgTechnology

**Synopsis** Messaging technology or protocol associated with this contact

**Enumeration** urn:liberty:idpp:msgTechnology:voice, urn:liberty:idpp:msgTechnology:fax, urn:liberty:idpp:msgTechnology:email, urn:liberty:idpp:msgTechnology:pager, urn:liberty:idpp:msgTechnology:aol, urn:liberty:idpp:msgTechnology:icq, urn:liberty:idpp:msgTechnology:yahoo, urn:liberty:idpp:msgTechnology:msn, urn:liberty:idpp:msgTechnology:mms, urn:liberty:idpp:msgTechnology:mim, urn:liberty:idpp:msgTechnology:irc

**Cardinality** 1

**Example** <MsgTechnology>urn:liberty:idpp:msgTechnology:icq</MsgTechnology>

Technology or protocol identifier of a messaging address. Although this is an enumeration, the list of enumerators is expected to evolve over time. Therefore enumerators are URIs. Additional enumerators can be assigned as specified in [LibReg]. Currently following enumerators are specified:

```
urn:liberty:idpp:msgTechnology:pots
urn:liberty:idpp:msgTechnology:voip
urn:liberty:idpp:msgTechnology:fax
urn:liberty:idpp:msgTechnology:email
urn:liberty:idpp:msgTechnology:mms
urn:liberty:idpp:msgTechnology:pager
urn:liberty:idpp:msgTechnology:aol
urn:liberty:idpp:msgTechnology:icq
urn:liberty:idpp:msgTechnology:yahoo
urn:liberty:idpp:msgTechnology:msn
urn:liberty:idpp:msgTechnology:mim
urn:liberty:idpp:msgTechnology:irc
```

## 9.47. MsgProvider

**Synopsis** Service provider or domain that provides messaging services

**Data type** ces

**Cardinality** 0-1

**Example** <MsgProvider>AOL</MsgProvider>

This element is effectively an enumeration within a msgtechnology. The msgprovider may appear empty if the addressing of the messaging technology is such that the provider does not matter. For example phone numbers and internet email addresses are provider neutral.

Typically instant messaging systems run on some provider's servers and the addressing is only meaningful within that provider. For these technologies provider must be specified.

## 9.48. MsgAccount

Synopsis Account or address information within messaging provider

Data type ces

Cardinality 1

Example <MsgAccount>123435234</MsgAccount>

Account name or number. If there is an unambiguous way to address the account in either internet domain or in public switched telephone network, then the account SHOULD be represented in that form. Otherwise msgAccount will be meaningful only within a provider of given technology.

For email, the account is the principal's internet domain email address, e.g. sampo@symlabs.com

For POTS or other public telephony, the account is simply the telephone number in international E.123 format (e.g. see <http://www.ngi.org/enum/pub/DraftRecE123.htm>).

The E.123 specification provides a mechanism, the "ext" keyword, for representing subaccounts such as might be family voice mailboxes where you dial one additional digit to reach family member's voice mail box.

## 9.49. MsgSubaccount

Synopsis Subaccount within messaging account, such as voice mail box under phone number

Data type ces

Cardinality 0-1

Example <MsgSubaccount>1</MsgSubaccount>

Subaccount such as might be family voice mailboxes where you dial one additional digit to reach family member's voice mail box.

## 9.50. MsgLimit

Synopsis Size limit for the message

Data type integer

Cardinality 0-1

Example <MsgLimit>160</MsgLimit>

Size limit for the messaging contact. Unit and maximum value for this can be inferred from the messaging technology, but user may want to limit the size of messages that he wants to receive using a given messaging account. For example, mobile based mail account might not want to receive a 2MB attachment.

Following units are suggested

Technology	Unit
voice	seconds
fax	pages
email	octet
gsmsms	octet
pager	octet
aol	octet
icq	octet
yahoo	octet
msn	octet
mms	octet
mim	octet
irc	octet

## 9.51. Facade

Synopsis Principal's look and sound facade

Cardinality 0-1

XML-attributes

**Table 14.**

### Contents of Facade

Attribute	Localized	Type	Synopsis
MugShot	n/a	URL	Face photo
WebSite	n/a	URI	Web site of the principal
NamePronounced	n/a	URL	User's common name pronounced (usually by the user)
GreetSound	n/a	URL	Greeting sound, user saying "Hello" to someone else
GreetMeSound	n/a	URL	Sound for user interface to greet the user

### Description

#### Example

```
<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>
```

### XML schema

```
<element name="Facade" type="idpp:FacadeType"/>
```

```
1707 <complexType name="FacadeType">
1708   <sequence>
1709     <element ref="idpp:MugShot" minOccurs="0" maxOccurs="1"/>
1710     <element ref="idpp:WebSite" minOccurs="0" maxOccurs="1"/>
1711     <element ref="idpp:NamePronounced" minOccurs="0" maxOccurs="1"/>
1712     <element ref="idpp:GreetSound" minOccurs="0" maxOccurs="1"/>
1713     <element ref="idpp:GreetMeSound" minOccurs="0" maxOccurs="1"/>
1714     <any namespace="##other" processContents="skip" minOccurs="0"/>
1715   </sequence>
1716   <attributeGroup ref="idpp:commonAttributes"/>
1717 </complexType>
1718
1719
1720
```

## 9.52. MugShot

Synopsis	Face photo
Data type	URL
Cardinality	0-1
Example	<b>&lt;MugShot&gt;</b> <a href="http://fotoserver.com/~joedoe/face.gif">http://fotoserver.com/~joedoe/face.gif</a> <b>&lt;/MugShot&gt;</b>

## 9.53. WebSite

Synopsis	Web site of the principal
Data type	URI
Cardinality	0-1
Example	<b>&lt;WebSite&gt;</b> <a href="http://provider.com/~user">http://provider.com/~user</a> <b>&lt;/WebSite&gt;</b>

Specifies the "home" page of the principal

## 9.54. NamePronounced

Synopsis	User's common name pronounced (usually by the user)
Data type	URL
Cardinality	0-1
Example	<b>&lt;NamePronounced&gt;</b> <a href="http://fotoserver.com/~joedoe/name.wav">http://fotoserver.com/~joedoe/name.wav</a> <b>&lt;/NamePronounced&gt;</b>

## 9.55. GreetSound

Synopsis	Greeting sound, user saying "Hello" to someone else
Data type	URL

1740	Cardinality	0-1
1741	Example	<code>&lt;GreetSound&gt;http://fotoserver.com/~joedoe/greet.wav&lt;/GreetSound&gt;</code>
1742	<b>9.56. GreetMeSound</b>	
1743	Synopsis	Sound for user interface to greet the user
1744	Data type	URL
1745	Cardinality	0-1
1746	Example	<code>&lt;GreetMeSound&gt;http://fotoserver.com/~joedoe/greetme.wav&lt;/GreetMeSound&gt;</code>

## 1747 **9.57. IDPPDemographics**

1748	Synopsis	Base level demographics used by ID-PP
1749	Cardinality	0-1
1750	XML-attributes	

**Table 15.**

**Contents of IDPPDemographics**

Attribute	Localized	Type	Synopsis
DisplayLanguage	n/a	ISO Language code	The language the principal prefers for displayed user interfaces
Language	n/a	ISO Language code	Languages the principal is able to understand
Birthday	n/a	gMonthDay	Birthday without year
Age	n/a	number	Age of the principal in years
TimeZone	n/a	(hh,mm) = /^[+-](\d\d):(\d\d)\$/	Time zone of the principal

**Description**

Some demographic data like language are so fundamental for operation of service like ID-PP that it has been included in the IDPPDemographics container. Other forms of demographic data is expected to live in a profile service of its own.

**Example**

```
<IDPPDemographics>
  <DisplayLanguage>pt-br</DisplayLanguage>
  <Language>pt</Language>
  <Language>fi</Language>
  <Language>en</Language>
  <Birthday>-05-09</Birthday>
  <Age>18</Age>
  <TimeZone>+05:00</TimeZone>
</IDPPDemographics>
```

**XML schema**

```
<element name="IDPPDemographics" type="idpp:IDPPDemographicsType"/>
<complexType name="IDPPDemographicsType">
  <sequence>
    <element ref="idpp:DisplayLanguage" minOccurs="0" maxOccurs="1"/>
    <element ref="idpp:Language" minOccurs="0" maxOccurs="10"/>
    <element ref="idpp:Birthday" minOccurs="0" maxOccurs="1"/>
    <element ref="idpp:Age" minOccurs="0" maxOccurs="1"/>
    <element ref="idpp:TimeZone" minOccurs="0" maxOccurs="1"/>
    <any namespace="##other" processContents="skip" minOccurs="0"/>
  </sequence>
  <attributeGroup ref="idpp:commonAttributes"/>
</complexType>
```

## 9.58. DisplayLanguage

**Synopsis** The language the principal prefers for displayed user interfaces

**Data type** ISO Language code

**Cardinality** 0-1

Example `<DisplayLanguage>pt-br</DisplayLanguage>`

We keep the user interface language distinct from understood languages because many "translated" web sites are not as good as the versions in some major language. Thus many users that understand a major language will prefer the smoother user interfaces available in that language over their native language.

## 9.59. Language

Synopsis Languages the principal is able to understand

Data type ISO Language code

Cardinality 0-10

Example `<Language>pt</Language>`

`<Language>fi</Language> <Language>en</Language>`

These languages are appropriate for call routing in call centers. They may also be used in case displaylanguage is not available.

## 9.60. Birthday

Synopsis Birthday without year

Data type gMonthDay

Cardinality 0-1

Example `<Birthday>-05-09</Birthday>`

The intended applications of this attribute are computation of zodiac or birthday reminder services. The point is to reveal the day without the year so dob is not given out.

N.B. Age and Birthday attributes together allow DOB to be determined, thus caution should be exercised not to reveal both of these simultaneously or to the same party unless that party is trusted enough for disclosing DOB.

Although Birthday is often computed from DOB, an implementation may actually store it in a database as long as it is automatically kept in synchrony with the DOB.

It is typical for an implementation to ignore modification of Birthday. Modify the DOB instead for more reliable behaviour.

## 9.61. Age

Synopsis Age of the principal in years

Data type number

Cardinality 0-1

Example `<Age>18</Age>`



Age of the principal in years. In many cases it is only relevant to know whether the principal is above a certain age. Thus we can improve privacy by not giving out birth date, but just the age.

N.B. Age and Birthday attributes together allow DOB to be determined, thus caution should be exercised not to reveal both of these simultaneously or to the same party unless that party is trusted enough for disclosing DOB.

Although Age is usually computed from DOB, an implementation MAY use some form of batch job to periodically update a database. If This approach is chosen, the skew from DOB SHOULD not exceed one week. The update MUST be done such that Age never portrays the principal older than she really is.

Age should not be modified directly with Modify request and an implementation is free to ignore such modification. Modify the DOB instead.

## 9.62. TimeZone

Synopsis Time zone of the principal

Data type (hh,mm) = /^[+-](\d\d):(\d\d)\$/

Cardinality 0-1

Example <TimeZone>+05:00</TimeZone>

Expresses timezone as offset from UTC. Time zone is often computed from geolocation and local laws (e.g. summer time).

## 9.63. SignKey

Synopsis Principal's public key or certificate for signing

Data type ds:KeyInfoType

Cardinality 0-1

Example

Public key or certificate that the principal may use for applications such as email signing (PGP based or S/MIME) or some enterprise applications.

N.B. this certificate is not necessarily related to any certificate based Liberty authentication contexts.

This key distribution mechanism does not address certificate validity and revocation issues.

SignKey follows the definition of KeyInfo container in RFC3275, section 4.4. The extension mechanisms that are defined for KeyInfo apply.

## 9.64. EncryptKey

Synopsis Principal's public key or certificate for encryption

Data type ds:KeyInfoType

Cardinality 0-1

1853 Example

1854 See entry for `SignKey` for discussion

1855 \the\_end

## 10. XML schema for ID-PP

1856

```
1857 <?xml version="1.0" encoding="UTF-8"?>
1858 <xs:schema
1859   targetNamespace="urn:liberty:idpp:1.0"
1860   xmlns="urn:liberty:idpp:1.0"
1861   xmlns:xs="http://www.w3.org/2001/XMLSchema"
1862   xmlns:idpp="urn:liberty:idpp:1.0"
1863   xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
1864   xmlns:xml="http://www.w3.org/XML/1998/namespace"
1865   elementFormDefault="qualified">
1866 <xs:import namespace="http://www.w3.org/2000/09/xmldsig#" schemaLocation="http://www.w3.org/TR/xmldsig-core/xmldsig-core-
1867 schema.xsd"/>
1868 <xs:import namespace="http://www.w3.org/XML/1998/namespace" schemaLocation="http://www.w3.org/2001/03/xml.xsd"/>
1869
1870 <!-- Adding include for DST schema -->
1871 <xs:include schemaLocation="lib-svc-dst.xsd"/>
1872 <!-- end include -->
1873
1874 <xs:complexType name="String">
1875   <xs:simpleContent>
1876     <xs:extension base="xs:string">
1877       <xs:attributeGroup ref="idpp:commonAttributes"/>
1878     </xs:extension>
1879   </xs:simpleContent>
1880 </xs:complexType>
1881
1882 <xs:complexType name="LocalizableString">
1883   <xs:simpleContent>
1884     <xs:extension base="idpp:String">
1885       <xs:attribute ref="xml:lang" use="optional"/>
1886       <xs:attribute name="script" type="xs:anyURI" use="optional"/>
1887     </xs:extension>
1888   </xs:simpleContent>
1889 </xs:complexType>
1890
1891 <xs:complexType name="Integer">
1892   <xs:simpleContent>
1893     <xs:extension base="xs:integer">
1894       <xs:attributeGroup ref="idpp:commonAttributes"/>
1895     </xs:extension>
1896   </xs:simpleContent>
1897 </xs:complexType>
1898
1899 <xs:complexType name="URI">
1900   <xs:simpleContent>
1901     <xs:extension base="xs:anyURI">
1902       <xs:attributeGroup ref="idpp:commonAttributes"/>
1903     </xs:extension>
1904   </xs:simpleContent>
1905 </xs:complexType>
1906
1907 <xs:complexType name="AltIDType">
1908   <xs:simpleContent>
1909     <xs:extension base="idpp:String">
1910       <xs:attribute name="altIDType" type="xs:anyURI" use="optional"/>
1911     </xs:extension>
1912   </xs:simpleContent>
1913 </xs:complexType>
1914
1915 <xs:complexType name="KeyInfoType" mixed="true">
1916   <xs:simpleContent>
1917     <xs:extension base="ds:KeyInfoType">
1918       <xs:attributeGroup ref="idpp:commonAttributes"/>
1919     </xs:extension>
1920   </xs:simpleContent>
1921 </xs:complexType>
1922
1923 <xs:complexType name="Date">
1924   <xs:simpleContent>
1925     <xs:extension base="xs:date">
1926       <xs:attributeGroup ref="idpp:commonAttributes"/>
1927     </xs:extension>
1928   </xs:simpleContent>
1929 </xs:complexType>
1930
```

```
1931 <xs:complexType name="MonthDay">
1932   <xs:simpleContent>
1933     <xs:extension base="xs:gMonthDay">
1934       <xs:attributeGroup ref="idpp:commonAttributes"/>
1935     </xs:extension>
1936   </xs:simpleContent>
1937 </xs:complexType>
1938
1939 <xs:element name="IDPP" type="idpp:IDPPType"/>
1940 <xs:complexType name="IDPPType">
1941   <xs:sequence>
1942     <xs:element ref="idpp:InformalName" minOccurs="0" maxOccurs="1"/>
1943     <xs:element ref="idpp:LInformalName" minOccurs="0" maxOccurs="10"/>
1944     <xs:element ref="idpp:CommonName" minOccurs="0" maxOccurs="1"/>
1945     <xs:element ref="idpp:LegalIdentity" minOccurs="0" maxOccurs="1"/>
1946     <xs:element ref="idpp:IDPPEmploymentIdentity" minOccurs="0" maxOccurs="1"/>
1947     <xs:element ref="idpp:IDPPAddressCard" minOccurs="0" maxOccurs="10"/>
1948     <xs:element ref="idpp:MsgContact" minOccurs="0" maxOccurs="30"/>
1949     <xs:element ref="idpp:Facade" minOccurs="0" maxOccurs="1"/>
1950     <xs:element ref="idpp:IDPPDemographics" minOccurs="0" maxOccurs="1"/>
1951     <xs:element ref="idpp:SignKey" minOccurs="0" maxOccurs="1"/>
1952     <xs:element ref="idpp:EncryptKey" minOccurs="0" maxOccurs="1"/>
1953     <xs:any namespace="##other" processContents="skip" minOccurs="0"/>
1954   </xs:sequence>
1955   <xs:attributeGroup ref="idpp:commonAttributes"/>
1956 </xs:complexType>
1957
1958 <xs:element name="InformalName" type="idpp:String"/>
1959 <xs:element name="LInformalName" type="idpp:LocalizableString"/>
1960
1961 <xs:element name="CommonName" type="idpp:CommonNameType"/>
1962 <xs:complexType name="CommonNameType">
1963   <xs:sequence>
1964     <xs:element ref="idpp:CN" minOccurs="0" maxOccurs="1"/>
1965     <xs:element ref="idpp:LCN" minOccurs="0" maxOccurs="10"/>
1966     <xs:element ref="idpp:AltCN" minOccurs="0" maxOccurs="10"/>
1967     <xs:element ref="idpp:LAltCN" minOccurs="0" maxOccurs="10"/>
1968     <xs:element ref="idpp:AnalyzedName" minOccurs="0" maxOccurs="1"/>
1969     <xs:any namespace="##other" processContents="skip" minOccurs="0"/>
1970   </xs:sequence>
1971   <xs:attributeGroup ref="idpp:commonAttributes"/>
1972 </xs:complexType>
1973
1974 <xs:element name="CN" type="idpp:String"/>
1975 <xs:element name="LCN" type="idpp:LocalizableString"/>
1976 <xs:element name="AltCN" type="idpp:String"/>
1977 <xs:element name="LAltCN" type="idpp:LocalizableString"/>
1978
1979 <xs:element name="AnalyzedName" type="idpp:AnalyzedNameType"/>
1980 <xs:complexType name="AnalyzedNameType">
1981   <xs:sequence>
1982     <xs:element ref="idpp:PersonalTitle" minOccurs="0" maxOccurs="1"/>
1983     <xs:element ref="idpp:LPersonalTitle" minOccurs="0" maxOccurs="10"/>
1984     <xs:element ref="idpp:FN" minOccurs="0" maxOccurs="1"/>
1985     <xs:element ref="idpp:LFN" minOccurs="0" maxOccurs="10"/>
1986     <xs:element ref="idpp:SN" minOccurs="0" maxOccurs="1"/>
1987     <xs:element ref="idpp:LSN" minOccurs="0" maxOccurs="10"/>
1988     <xs:element ref="idpp:MN" minOccurs="0" maxOccurs="1"/>
1989     <xs:element ref="idpp:LMN" minOccurs="0" maxOccurs="10"/>
1990     <xs:any namespace="##other" processContents="skip" minOccurs="0"/>
1991   </xs:sequence>
1992   <xs:attribute name="nameScheme" type="xs:anyURI" use="optional"/>
1993   <xs:attributeGroup ref="idpp:commonAttributes"/>
1994 </xs:complexType>
1995
1996 <xs:element name="PersonalTitle" type="idpp:String"/>
1997 <xs:element name="LPersonalTitle" type="idpp:LocalizableString"/>
1998 <xs:element name="FN" type="idpp:String"/>
1999 <xs:element name="LFN" type="idpp:LocalizableString"/>
2000 <xs:element name="SN" type="idpp:String"/>
2001 <xs:element name="LSN" type="idpp:LocalizableString"/>
2002 <xs:element name="MN" type="idpp:String"/>
2003 <xs:element name="LMN" type="idpp:LocalizableString"/>
2004
2005 <xs:element name="LegalIdentity" type="idpp:LegalIdentityType"/>
2006 <xs:complexType name="LegalIdentityType">
2007   <xs:sequence>
```

```
2008 <xs:element ref="idpp:LegalName" minOccurs="0" maxOccurs="1"/>
2009 <xs:element ref="idpp:LLegalName" minOccurs="0" maxOccurs="10"/>
2010 <xs:element ref="idpp:AnalyzedName" minOccurs="0" maxOccurs="1"/>
2011 <xs:element ref="idpp:VAT" minOccurs="0" maxOccurs="1"/>
2012 <xs:element ref="idpp:AltID" minOccurs="0" maxOccurs="unbounded"/>
2013 <xs:element ref="idpp:DOB" minOccurs="0" maxOccurs="1"/>
2014 <xs:element ref="idpp:Gender" minOccurs="0" maxOccurs="1"/>
2015 <xs:element ref="idpp:MaritalStatus" minOccurs="0" maxOccurs="1"/>
2016 <xs:any namespace="##other" processContents="skip" minOccurs="0"/>
2017 </xs:sequence>
2018 <xs:attributeGroup ref="idpp:commonAttributes"/>
2019 </xs:complexType>
2020
2021 <xs:element name="LegalName" type="idpp:String"/>
2022 <xs:element name="LLegalName" type="idpp:LocalizableString"/>
2023 <xs:element name="VAT" type="idpp:String"/>
2024 <xs:element name="AltID" type="idpp:AltIDType"/>
2025 <xs:element name="DOB" type="idpp:Date"/>
2026 <xs:element name="Gender" type="idpp:URI"/>
2027 <xs:element name="MaritalStatus" type="idpp:URI"/>
2028
2029 <xs:element name="IDPPEmploymentIdentity" type="idpp:IDPPEmploymentIdentityType"/>
2030 <xs:complexType name="IDPPEmploymentIdentityType">
2031 <xs:sequence>
2032 <xs:element ref="idpp:JobTitle" minOccurs="0" maxOccurs="1"/>
2033 <xs:element ref="idpp:LJobTitle" minOccurs="0" maxOccurs="10"/>
2034 <xs:element ref="idpp:CommonName" minOccurs="0" maxOccurs="1"/>
2035 <xs:any namespace="##other" processContents="skip" minOccurs="0"/>
2036 </xs:sequence>
2037 <xs:attributeGroup ref="idpp:commonAttributes"/>
2038 </xs:complexType>
2039
2040 <xs:element name="JobTitle" type="idpp:String"/>
2041 <xs:element name="LJobTitle" type="idpp:LocalizableString"/>
2042
2043 <xs:element name="IDPPAddressCard" type="idpp:IDPPAddressCardType"/>
2044 <xs:complexType name="IDPPAddressCardType">
2045 <xs:sequence>
2046 <xs:element ref="idpp:IDPPAddrType" minOccurs="0" maxOccurs="10"/>
2047 <xs:element ref="idpp:Address" minOccurs="0" maxOccurs="1"/>
2048 <xs:element ref="idpp:Nick" minOccurs="0" maxOccurs="1"/>
2049 <xs:element ref="idpp:LNick" minOccurs="0" maxOccurs="10"/>
2050 <xs:any namespace="##other" processContents="skip" minOccurs="0"/>
2051 </xs:sequence>
2052 <xs:attributeGroup ref="idpp:commonAttributes"/>
2053 </xs:complexType>
2054
2055 <xs:element name="IDPPAddrType" type="idpp:URI"/>
2056
2057 <xs:element name="Address" type="idpp:AddressType"/>
2058 <xs:complexType name="AddressType">
2059 <xs:sequence>
2060 <xs:element ref="idpp:PostalAddress" minOccurs="0" maxOccurs="1"/>
2061 <xs:element ref="idpp:LPostalAddress" minOccurs="0" maxOccurs="10"/>
2062 <xs:element ref="idpp:PostalCode" minOccurs="0" maxOccurs="1"/>
2063 <xs:element ref="idpp:L" minOccurs="0" maxOccurs="1"/>
2064 <xs:element ref="idpp:LL" minOccurs="0" maxOccurs="10"/>
2065 <xs:element ref="idpp:St" minOccurs="0" maxOccurs="1"/>
2066 <xs:element ref="idpp:LSt" minOccurs="0" maxOccurs="10"/>
2067 <xs:element ref="idpp:C" minOccurs="0" maxOccurs="1"/>
2068 <xs:any namespace="##other" processContents="skip" minOccurs="0"/>
2069 </xs:sequence>
2070 <xs:attributeGroup ref="idpp:commonAttributes"/>
2071 </xs:complexType>
2072
2073 <xs:element name="PostalAddress" type="idpp:String"/>
2074 <xs:element name="LPostalAddress" type="idpp:LocalizableString"/>
2075 <xs:element name="PostalCode" type="idpp:String"/>
2076 <xs:element name="L" type="idpp:String"/>
2077 <xs:element name="LL" type="idpp:LocalizableString"/>
2078 <xs:element name="St" type="idpp:String"/>
2079 <xs:element name="LSt" type="idpp:LocalizableString"/>
2080 <xs:element name="C" type="idpp:String"/>
2081 <xs:element name="Nick" type="idpp:String"/>
2082 <xs:element name="LNick" type="idpp:LocalizableString"/>
2083
2084 <xs:element name="MsgContact" type="idpp:MsgContactType"/>
```

```
2085 <xs:complexType name="MsgContactType">
2086   <xs:sequence>
2087     <xs:element ref="idpp:Nick" minOccurs="0" maxOccurs="1"/>
2088     <xs:element ref="idpp:LNick" minOccurs="0" maxOccurs="10"/>
2089     <xs:element ref="idpp:Comment" minOccurs="0" maxOccurs="1"/>
2090     <xs:element ref="idpp:MsgType" minOccurs="0" maxOccurs="1"/>
2091     <xs:element ref="idpp:MsgMethod" minOccurs="1" maxOccurs="1"/>
2092     <xs:element ref="idpp:MsgTechnology" minOccurs="1" maxOccurs="1"/>
2093     <xs:element ref="idpp:MsgProvider" minOccurs="0" maxOccurs="1"/>
2094     <xs:element ref="idpp:MsgAccount" minOccurs="1" maxOccurs="1"/>
2095     <xs:element ref="idpp:MsgSubaccount" minOccurs="0" maxOccurs="1"/>
2096     <xs:element ref="idpp:MsgLimit" minOccurs="0" maxOccurs="1"/>
2097     <xs:any namespace="##other" processContents="skip" minOccurs="0"/>
2098   </xs:sequence>
2099   <xs:attributeGroup ref="idpp:commonAttributes"/>
2100 </xs:complexType>
2101
2102 <xs:element name="Comment" type="idpp:String"/>
2103 <xs:element name="MsgType" type="idpp:URI"/>
2104 <xs:element name="MsgMethod" type="idpp:URI"/>
2105 <xs:element name="MsgTechnology" type="idpp:URI"/>
2106 <xs:element name="MsgProvider" type="idpp:String"/>
2107 <xs:element name="MsgAccount" type="idpp:String"/>
2108 <xs:element name="MsgSubaccount" type="idpp:String"/>
2109 <xs:element name="MsgLimit" type="idpp:Integer"/>
2110
2111 <xs:element name="Facade" type="idpp:FacadeType"/>
2112 <xs:complexType name="FacadeType">
2113   <xs:sequence>
2114     <xs:element ref="idpp:MugShot" minOccurs="0" maxOccurs="1"/>
2115     <xs:element ref="idpp:WebSite" minOccurs="0" maxOccurs="1"/>
2116     <xs:element ref="idpp:NamePronounced" minOccurs="0" maxOccurs="1"/>
2117     <xs:element ref="idpp:GreetSound" minOccurs="0" maxOccurs="1"/>
2118     <xs:element ref="idpp:GreetMeSound" minOccurs="0" maxOccurs="1"/>
2119     <xs:any namespace="##other" processContents="skip" minOccurs="0"/>
2120   </xs:sequence>
2121   <xs:attributeGroup ref="idpp:commonAttributes"/>
2122 </xs:complexType>
2123
2124 <xs:element name="MugShot" type="idpp:URI"/>
2125 <xs:element name="WebSite" type="idpp:URI"/>
2126 <xs:element name="NamePronounced" type="idpp:URI"/>
2127 <xs:element name="GreetSound" type="idpp:URI"/>
2128 <xs:element name="GreetMeSound" type="idpp:URI"/>
2129
2130 <xs:element name="IDPPDemographics" type="idpp:IDPPDemographicsType"/>
2131 <xs:complexType name="IDPPDemographicsType">
2132   <xs:sequence>
2133     <xs:element ref="idpp:DisplayLanguage" minOccurs="0" maxOccurs="1"/>
2134     <xs:element ref="idpp:Language" minOccurs="0" maxOccurs="10"/>
2135     <xs:element ref="idpp:BirthDay" minOccurs="0" maxOccurs="1"/>
2136     <xs:element ref="idpp:Age" minOccurs="0" maxOccurs="1"/>
2137     <xs:element ref="idpp:TimeZone" minOccurs="0" maxOccurs="1"/>
2138     <xs:any namespace="##other" processContents="skip" minOccurs="0"/>
2139   </xs:sequence>
2140   <xs:attributeGroup ref="idpp:commonAttributes"/>
2141 </xs:complexType>
2142
2143 <xs:element name="DisplayLanguage" type="idpp:String"/>
2144 <xs:element name="Language" type="idpp:String"/>
2145 <xs:element name="BirthDay" type="idpp:MonthDay"/>
2146 <xs:element name="Age" type="idpp:Integer"/>
2147 <xs:element name="TimeZone" type="idpp:String"/>
2148 <xs:element name="SignKey" type="idpp:KeyInfoType"/>
2149 <xs:element name="EncryptKey" type="idpp:KeyInfoType"/>
2150 </xs:schema>
2151
```

## 11. Normative References

[ID-WSF] Identity Services Framework \*\*\*  
[ID-DST] Data Services Template \*\*\*  
[ID-DISCO] ID-WSF Discovery service  
[RFC2119] MUST, MAY, etc.  
[RFC3275] XML-Signature Syntax and Processing, section 4.4 (keyInfo)  
[XPath] XPath specification  
[LibReg] Liberty Registry. Mechanism or body for assigning enumerators. (\*\* TBD)

## 12. Informational References

- 2169 [Overview] Some ISF overview document
- 2170
- 2171 [ID-FF] ID Federation Framework
- 2172
- 2173 [ID-ROI] ID-WSF Resource Owner Interaction protocol
- 2174
- 2175 [LDAP] RFC2251
- 2176
- 2177 [BIP] Liberty Business Identity Profile. To be published.
- 2178
- 2179 [CB] Liberty Contact Book Service. To be published.
- 2180
- 2181



## 13. Glossary

2182		
2183	addressing	See XML addressing
2184	AP	see attribute provider
2185	application	Generally a customer facing service. From core profile prespective
2186		synonymous with service.
2187	attribute	Basic unit of information about the principal. Roughly similar
2188		to LDAP attribute and mostly synonymous with XML element (as
2189		opposed to container). Not to be confused with XML-attribute.
2190	attribute class	depreceted term for attribute container
2191	attribute container	a collection of attributes grouped together according to expected
2192		usage patterns. Some early work referred to these as "bundles"
2193	attribute provider	a web service that allows querying and updating of attribute data.
2194		Typically AP also enforces permissions.
2195	bundle	deprecated term for attribute container
2196	ces	Case Exact String. A term used in attribute descriptions to refer to
2197		free form strings. Exact Match means that if a search against this
2198		attribute is made, the comparisons are case sensitive. See also cis.
2199	ces	Case Inexact String. A term used in attribute descriptions to refer to
2200		free form strings. Inexact Match means that if a search against this
2201		attribute is made, the comparisons are case insensitive. See also ces.
2202	discovery service	A Liberty service for locating attribute providers and brokers
2203	end point	Colloquial term for entry point
2204	entry point	A SOAP (RPC) address and function name that can be used to
2205		obtain some service. In Liberty entry points are what Discovery
2206		Service allows one to discover
2207	IdP	Identity Provider. See [ID-FF].
2208	PIP	Historical name for ID-PP
2209	postal addressing	Address system used for snail mail
2210	POTS	Plain Old Telephone Service. Conventional analog voice telephone.
2211		Contrast with VoIP.
2212	service	A collection of entry points designed to offer some service or to
2213		provide some information. See also attribute provider.
2214	VoIP	Voice Over IP. An emerging voice telephony technology that uses
2215		IP networks. Contrast with POTS.
2216	XML addressing	A method for locating and refereing to data located in another
2217		service

2218	XML-attribute	attribute of an XML-element. Usually used to convey metadata
2219		regarding the core profile attribute represented by the XML element.
2220		Not to be confused with "attribute".
2221	XML-container	A special XML-element that contains other XML-elements. In
2222		the core profile, XML-containers are used to represent attribute
2223		containers.
2224	XML-element	A XML tag. Usually XML elements are used to represent core
2225		profile attributes.