International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

X.1255

(09/2013)

SERIES X: DATA NETWORKS, OPEN SYSTEM COMMUNICATIONS AND SECURITY

Cyberspace security – Identity management

Framework for discovery of identity management information

Recommendation ITU-T X.1255

ITU-T X-SERIES RECOMMENDATIONS

DATA NETWORKS, OPEN SYSTEM COMMUNICATIONS AND SECURITY

PUBLIC DATA NETWORKS	X.1–X.199
OPEN SYSTEMS INTERCONNECTION	X.200–X.299
INTERWORKING BETWEEN NETWORKS	X.300–X.399
MESSAGE HANDLING SYSTEMS	X.400–X.499
DIRECTORY	X.500–X.599
OSI NETWORKING AND SYSTEM ASPECTS	X.600–X.699
OSI MANAGEMENT	X.700–X.799
SECURITY	X.800–X.849
OSI APPLICATIONS	X.850–X.899
OPEN DISTRIBUTED PROCESSING	X.900–X.999
INFORMATION AND NETWORK SECURITY	
General security aspects	X.1000–X.1029
Network security	X.1030–X.1049
Security management	X.1050–X.1069
Telebiometrics	X.1080–X.1099
SECURE APPLICATIONS AND SERVICES	
Multicast security	X.1100–X.1109
Home network security	X.1110–X.1119
Mobile security	X.1120–X.1139
Web security	X.1140–X.1149
Security protocols	X.1150–X.1159
Peer-to-peer security	X.1160–X.1169
Networked ID security	X.1170–X.1179
IPTV security	X.1180–X.1199
CYBERSPACE SECURITY	
Cybersecurity	X.1200–X.1229
Countering spam	X.1230–X.1249
Identity management	**X.1250–X.1279**
SECURE APPLICATIONS AND SERVICES	
Emergency communications	X.1300–X.1309
Ubiquitous sensor network security	X.1310–X.1339
CYBERSECURITY INFORMATION EXCHANGE	
Overview of cybersecurity	X.1500–X.1519
Vulnerability/state exchange	X.1520–X.1539
Event/incident/heuristics exchange	X.1540–X.1549
Exchange of policies	X.1550–X.1559
Heuristics and information request	X.1560–X.1569
Identification and discovery	X.1570–X.1579
Assured exchange	X.1580–X.1589

For further details, please refer to the list of ITU-T Recommendations.

Recommendation ITU-T X.1255

Framework for discovery of identity management information

Summary

The purpose of Recommendation ITU-T X.1255 is to provide an open architecture framework in which identity management information can be discovered. This IdM information will necessarily be represented in different ways and supported by various trust frameworks or other IdM systems using different metadata schemas. This framework will enable, for example, entities operating within the context of one IdM system to have identifiers from other IdM systems accurately resolved. Without the capability for discovering such information, users and organizations (or programs operating on their behalf) are left to determine how best to establish the credibility and authenticity of a suitable identity, whether for a user, a system resource, information or other entities. Based on this information, it is up to the user or organization to determine whether or not to rely on a given trust framework or other IdM system for such purposes. The core components of the framework set forth in this Recommendation include: 1) a digital entity data model, 2) a digital entity interface protocol, 3) one or more identifier/resolution systems and 4) one or more metadata registries. These components form the basis of the open architecture framework.

History

Edition	Recommendation	Approval	Study Group
1.0	ITU-T X.1255	2013-09-04	17

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure, e.g., interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database at http://www.itu.int/ITU-T/ipr/.

Table of Contents

Recommendation ITU-T X.1255

Framework for discovery of identity management information

1 Scope

Discovery of identity management information deals with the fact that one must have the ability to obtain relevant information about identifiers, including those utilizing e-mail address syntax and those that are URLs, as well as persistent identifiers. Such discovery is a key element for enabling interoperability across heterogeneous information systems.

The scope of this Recommendation is for a framework that:

- enables the discovery of identity-related information and its provenance, including information being identified such as services, processes and entities;
- enables the discovery of identity-related information attributes including, but not limited to visual logos and human-readable site names;
- enables the discovery of attributes and the functionality of applications;
- describes a data model and a protocol to enable meta-level interoperability for representation, access and discovery of the information referenced above in heterogeneous IdM environments.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ISO 8601] ISO 8601:2004, *Data elements and interchange formats – Information interchange – Representation of dates and times.*

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 entity [b-ITU-T Y.2720]: Anything that has a separate and distinct existence that can be uniquely identified. In the context of IdM, examples of entities include subscribers, users, network elements, networks, software applications, services and devices. An entity may have multiple identifiers.

3.1.2 identity provider [b-ITU-T Y.2720]: An entity that creates, maintains and manages trusted identity information of other entities (e.g., users/subscribers, organizations and devices) and offers identity-based services based on trust, business and other types of relationship.

3.1.3 relying party [b-ITU-T Y.2720]: An entity that relies on an identity representation or claim by a requesting/asserting entity.

3.1.4 trust [b-ITU-T Y.2720]: A measure of reliance on the character, ability, strength or truth of someone or something.

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

3.2.1 association: A relationship, if any, between two identified entities.

3.2.2 digital entity: An entity represented as, or converted to, a machine-independent data structure consisting of one or more elements in digital form that can be parsed by different information systems; the structure helps to enable interoperability among diverse information systems in the Internet.

3.2.3 discovery: The act or process of seeking or locating target information, i.e., obtaining knowledge pertaining to the target.

3.2.4 element: Part of a digital entity consisting of a type-value pair, where the type is represented by a resolvable persistent identifier and the value is the relevant digital information for that type.

3.2.5 federated registries: A collection of interoperable registries that register metadata and participate in a common set of methods to share information reliably and in a commonly understood format.

3.2.6 identifier: A sequence of bits used to obtain state information about the digital entity being identified; typically, this is done via an appropriate resolution system.

3.2.7 identity management: A means by which identity management information, whether for a user, a system resource, information or other entities, can be validated.

3.2.8 identity management information: Identity-related information including all types of metadata associated with identity, provenance, association and trust.

3.2.9 metadata: Structured information that pertains to the identity of users, systems, services, processes, resources, information or other entities.

3.2.10 persistent identifier: A unique identifier that resolves to state information about a digital entity and that is resolvable for at least as long as the digital entity exists.

3.2.11 provenance: Information pertaining to any source of information including the party or parties involved in generating it, introducing it and/or vouching for it.

3.2.12 registry: A mechanism for registering metadata about digital entities and storing metadata schemas, and which provides an ability to search the registry for persistent identifiers based on the use of the metadata schemas.

3.2.13 repository: An interface that accepts deposits of digital entities, enables their retention, and provides secure access to the digital entities via their identifiers.

3.2.14 resolution system: A system that accepts identifiers known to the system as input, and provides relevant state information about the entity being identified.

3.2.15 touch point: A registry within a system of federated registries that is selected to interface with a designated registry in another federation, typically for the purposes of peering.

3.2.16 trust framework: An IdM system where a set of verifiable commitments are made by each of the various parties in a transaction to their counter parties, and these commitments necessarily include: (a) controls to help ensure commitments are met and (b) remedies for failure to meet such commitments.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

API Application Program Interface

Bits Binary Digits

BNF Backus Normal Form

DE Digital Entity

DEIP Digital Entity Interface Protocol

DNA Deoxyribonucleic acid

HTTP Hypertext Transfer Protocol

ID Identifier

IdM Identity Management

IdP Identity Provider

MAC Media Access Control

P2P Peer-to-Peer

PKI Public Key Infrastructure

RP Relying Party

TCP Transmission Control Protocol

TF Trust Framework

URL Uniform Resource Locator

XML Extensible Markup Language

5 Conventions

None.

6 Recommendation

This Recommendation is for an open architecture framework to support the discovery of identity management information. It addresses the following subjects:

a) the concept of trust, which is an important aspect of identity management;

b) trust information, which may be used to determine how much reliance to place on any piece of IdM information;

c) federated registries for discovery;

d) an interoperability architecture for federation; and

e) a discussion of both hierarchical and peer-to-peer federations.

Discovery of identity management information is based on the use of metadata obtained from a registry or a system of federated registries. The framework includes the existence of a means of resolving persistent identifiers. In general, federated registries will be operated by multiple parties and shall support the digital entity data model to represent metadata records and the digital entity interface protocol to achieve the interoperability of such registries. The use of multiple schemas is presumed and each registry shall provide details of its publicly and/or privately supported metadata schemas by their respective persistent identifiers. The persistent identifiers of privately supported

schema may be made known publicly, if desired, or they may be maintained privately along with the associated metadata schemas for limited use within restricted communities.

Appendices I and II respectively provide an overview of scenarios of usage and an example BNF description of a type record (BNF is a standard notation for representing context-free grammars).

6.1 Notions of trust

The word 'trust' is a term of art and carries a number of connotations. To trust a person or process generally means to have some level of confidence in a certain outcome of events, even if those events are not precisely specified. Building discovery systems however, will require added specification. To simply state that A can trust B does not mean that A can trust B for all possible event outcomes. Trusting B to return a service for a given payment is not the same as trusting B to keep that payment secret or to not publicize the names of everyone who makes such payments.

The most important issue in trust frameworks is identity management, that is, are the parties in any given transaction really who they claim to be. But confidence in the outcome of a given transaction with a given party is dependent not only on the identity of that party but also on other attributes of the claims and assertions made by that party. To evaluate these attributes in a consistent fashion requires a vocabulary or set of metrics that can be applied to these attributes. Such measures and descriptions could be applied by third parties serving as trust framework rating agencies, or they could be averaged across groups of ratings from users, as is done in recommendation systems and other 'crowd sourcing' applications. Recommended categories for these measures and descriptions are described below.

Strength: Level of trustworthiness. How likely is this party to do what they say they will do? How likely is it that the assertion of identity (e.g., I am the author of that software and the royalty belongs to me) is correct? This is likely to be expressed as a numeric or letter grade.

Classification: What type of trustworthiness is being asserted? Can standard categories be established? Identity is a category unto itself. Other categories would include financial trustworthiness (e.g., how likely is a given party to perform as promised in a financial transaction), privacy (how likely is a given party to keep private information that it claims will not be released), and authoritativeness (e.g., how likely is it that the information received from a given party is accurate). Other high-level classes are possible and finer levels of granularity are possible for each class.

Length of trust chain: Some trust transactions depend on a chain of trust, frequently thought of in terms of a certificate hierarchy or layers of digitally-signed software. The concept is generally applicable to all areas of trust: the longer the chain of trusted assertions which go back to a trust anchor, the weaker the final level of trust. A measure of the length of this chain would be a key measure of trustworthiness of any identity or other assertion.

All of these attributes (and many others) would be candidates for inclusion in metadata records describing identity providers, relying parties and any other components involved in trusted transactions.

6.2 Trust information

Three distinct aspects of trust information are discussed below in relation to the functions and actions involved in federated discovery and those of the participating constituent entities.

6.2.1 Trust information in a discovery response

Enabling the discovery of identity management information is a primary objective of the open architecture set forth in this Recommendation; however, the determination of trust is left to the user to determine. Additional functionality or services can be supported within the architecture in the form of optional components/modules (software and/or hardware). In this sense, the architecture

could include a trust framework as an optional capability, as well as enhance/enrich a discovery response with trust information or even support a trust determination. External entities would have the ability to determine whether they want to receive this trust information directly. They could choose to deactivate the feature and seek to collect trust information on their own or even from their own sources to make a trust determination.

6.2.2 Trust in the discovery system

The discovery system must be trusted, so that external parties feel confident in using its capabilities to register identity management information or to access it. This type of trust can be achieved through various means, including the establishment of specific methods, along with associated policies and procedures for reliability purposes. These may include evaluation of the components that are implemented as part of the framework, actions (measures) taken for misbehaving components or external parties, and adaptation of strong security and privacy frameworks. However, defining the exact methodology to achieve this kind of trust is outside the scope of this Recommendation.

6.2.3 Trusting external parties

The architecture must support policies and procedures that encourage reliable external parties to use it for the purpose of registering information. For security reasons, component entities that are directly involved in registering identity management information must be able to control the data inserted in the system and detect and/or avoid cases where malicious parties try to register false information.

Anonymous requests should be supported, but many of them may not lead to useful responses unless the identity of the individual requesters is known beforehand through some other means. In such cases, the capability for validating identity should be provided by an identity management capability that applies to all components, including users/requesters. Within the framework, each component is allotted a unique persistent identifier by an authorized and recognized identity provider that can be resolved to relevant information about the component. As indicated earlier, no assumption is made about how any component of a specific instance will determine whether to trust this information.

For many requests, the identity of the requester (an external party that issued a discovery request) must be validated before issuing a response. In the general case, all requesters would be evaluated prior to taking any further action. The architecture does not assume any decision-making capability is invoked within it; however, before constructing the final response to a discovery request, an external decision support mechanism may be invoked in order for permission to be obtained in advance from the identity producers.

6.3 Federated registries for discovery

A system of federated registries for discovery is described in this Recommendation with the goal of enabling metadata and other information about identifiers, as well as trust frameworks and other IdM systems to be found and evaluated. Federated registries can work together to share their metadata entities subject to any restriction that may apply. The actual information which corresponds to this metadata may be stored in their respective registries (if such storage is allowed), in one or more distributed repositories and, in some cases the information corresponding to this metadata may not be accessible in the Internet at all. In the latter case, this limitation would typically be determined by resolving the identifier to relevant state information about the entity; however, a registry could choose to provide that information as well.

Within a system of federated registries, a given registry can contribute a metadata record for a given entity to a second registry either as a full copy of the original metadata record or as a summary of that original record. The submission would have the original record, or a variant of the original, described in such a way that identified from where it was obtained, and which characterized the

type of community or domain represented by that registry. That same initial registry could thus serve as a cross-domain collection point for many other registries and provide a search service that could refer searchers to other registries to gather additional information. Such collection points are sometimes referred to as touch points.

The registry component of the architecture is designed around several key concepts. In addition to requiring every registered digital entity to be allotted an identifier, metadata records in the registry are themselves structured as digital entities, each having an associated identifier. This allows the metadata records to be referenced separately, and their identifiers will resolve to current state information about the metadata entities, even if the records move from one registry to another, or are available from multiple registries.

Nothing in the architecture limits the number of metadata entities that can be registered for a single digital entity. It may be desirable to generate multiple metadata entities for the same information when seen from different perspectives, for different audiences, and so on. The management of these metadata entities is greatly simplified by the use of unique and persistent identifiers: for example, it can easily be determined if two metadata records do or do not reference the same underlying information. Additional entities can also be created to relate individual metadata entities to each other in ways that would not otherwise result from searching across the individual entities.

The architecture allows for many-to-many relationships, in both directions, between repositories and registries. A given repository can contribute metadata for the same entities to multiple registries, and a given registry can accept metadata from multiple repositories. Collecting metadata from multiple repositories into a single registry enables the federation of these repositories. Allowing these repositories to contribute metadata about the same entities to multiple registries enables a single repository to be part of multiple federations, distinguished perhaps by serving different communities, using different metadata schemas, different approaches to indexing and searching, and other capabilities.

Finally, an instance of a registry can be federated with other registries. Multiple registries can push their metadata entities, or entities that are a function of those original metadata records, to each other. A given registry, call it Reg1, can contribute a metadata record for a given object to a second registry, call it Reg2, either as a full copy of the original metadata record or as a summary of that original record. The submission would have the original record, or a variant of the original, incorporated in a digital entity in such a way that identified it as coming from Reg1, and characterized the type of community or domain represented by Reg1. If Reg1 always federates with Reg2, then Reg2 could serve as a cross-domain collection point for many other registries like Reg1 and provide a search service that could refer searchers to other registries or directly to the DEs themselves, depending on the approach to combining and indexing the potentially heterogeneous metadata records.

While the focus in this Recommendation is on identity management, such a system can also serve to discover other kinds of information in complex distributed systems in the Internet such as those involving "Cloud Computing" or the "Internet of Things". The resolution information in a system of federated registries is obtained from individual IdM systems. Use of the federation's discovery mechanism will enable interoperability of IdM systems, more generally, and provide information suitable for an entity to learn about other IdM systems and to aid in developing trust in the use of identifiers from those systems.

Basic registry technology is in use by a large number of groups, some of whom have made use of open source versions and others who have developed proprietary customizations based on commonly understood specifications. Federation is achieved via protocols for information sharing. An important part of future work based on this Recommendation will be to describe and then formalize these specifications, to define suitable protocols and procedures along with appropriate metadata schemas, and to determine a commonly acceptable approach to maintain privacy, where

appropriate. How one decides to select or rely on a given IdM system is considered outside the scope of this Recommendation.

7 Interoperability architecture for federated registries

The system of federated registries in this Recommendation is based on an open architecture that enables interoperability across arbitrary information systems. It provides a means for information authentication and for access to information structured as digital entities and stored in most types of standard storage systems. A digital entity is a common data structure that enables interoperability among systems in the Internet; the elements of a DE are digital material, namely typed data, including a unique persistent identifier for this material.

Three architectural components are used in managing DEs. Each of these components can be used on their own, but they complement each other and together they provide a distributed and scalable information management capability for the Internet. The components are:

a) a scalable and distributed identifier system for the identification of DEs and for identifier resolution;

b) repositories for access to and management of digital entities; and

c) registries for federated search and discovery. Using these components, the resulting distributed system can be managed through interface specifications and protocols instead of through the on-going maintenance of specific components.

Digital entities are the core element around which all other components and services are built and managed. Digital entities do not replace existing formats and data structures, but instead provide a common means for representing these formats and structures, allowing them to be uniformly interpreted and thus moveable in and out of various heterogeneous information systems and across changes in systems over time. This model, though simple at its core, is non-trivial in its detailed implementation, and includes a protocol for interacting with DEs through repositories. In this Recommendation, all metadata will conform to the DE data model for purposes of interoperability and ease of reference.

The DE data model and the digital entity interface protocol for accessing DEs, described below, combined with an identifier and/or resolution system and a registry/repository approach to accessing DEs, provide the core of the open architecture. Together these components enable the long-term management of information structured as digital entities by uniquely and persistently identifying them, providing a method for obtaining current state information about the objects, providing a service for obtaining or otherwise using the entities, and a means of determining the identifiers of DEs based on information in metadata registries.

7.1 Digital entity data model

The DE data model described herein provides a uniform means to represent metadata records as DEs, and can also be used to represent other types of information as DEs. It is a logical model that allows for multiple forms of encoding and storage, and enables a single point of reference (i.e., the identifier) for many types of information that may be available in the Internet. Each DE has an intrinsic set of attributes, a user-defined set of attributes, embodied in one or more elements and zero or more additional elements containing information such as text, video or images represented in digital form. All of these elements can be made available through a precisely defined DEIP specification (see clause 7.2), which incorporates the capability for authentication using public key security, and perhaps other means of authentication using higher-level APIs, as might be implemented by DE repositories. This provides access with privacy and security to DEs.

The essential fixed attribute of a DE is its associated unique persistent identifier, which can be resolved to current state information about the DE, including its location(s), access controls, and validation, by submitting a resolution request to the resolution system. Examples of other intrinsic

DE element attributes are: date last modified, date created, and size. User extensible attributes may be set by the users with appropriate permissions.

Attributes that are not specifically addressed by the basic DE data model include ownership, authentication and access terms and conditions. These attributes will be an important part of most DE implementations; however, a single solution seems unlikely. Ownership and access control information will likely be contained in user extensible DE attributes or in separate data elements. This provides a common way to deal with various ownership and information management schemes, as well as multiple authentication and authorization schemes, without making the assumption that a single approach will be used across all domains and user communities.

The combination of a standard data model, a defined protocol for interacting with that data model, and an identifier/resolution system, provides a key ingredient for the coherent long-term management of information in the Internet. The resolution system should be a distributed, secure, high-performance resolution system designed to enable persistent reference to digital entities over long periods of time and over changes in location, access methods, ownership and other mutable attributes.

The core capability for discovery of IdM information results from the use of the registry component, which includes the repository. The function of an individual registry is to federate across collections of DEs, enabling end users and applications to search through and navigate the universe of registered entities. Repositories that contain collections of DEs can contribute metadata about the DEs for which they are responsible to one or more registries. A single registry can collect metadata from multiple repositories, and a single repository can send metadata to multiple registries. The registries can provide search and reporting functions over the represented entities and provide an entry point into the structured world of DEs and repositories.

There may be situations in which the registries are not, strictly speaking, needed, e.g., in the case where a direct reference to a DE, in the form of its identifier, is embedded in another DE or in a message or other document. In many cases, however, the end user, or automated process acting on behalf of a user, will not know the identifier to begin with, and will have to use some variety of search or sorting process to discover the needed reference. Even if a user knows the identifier, the user may not know how to resolve it, or how to interpret the resolution results. Recording the existence of DEs in registries can help to solve that problem in a very general way.

By defining operations that interact with a specified data model, digital entities can be constructed and used to represent most types of structured information. These are discussed in the next clause. A standard digital entity data model is illustrated in Figure 1. Representation of the entities in a form that is independent of the implementation details of the relevant storage system is an essential interoperability feature, as it allows multiple storage formats and approaches to be normalized to a single logical model.

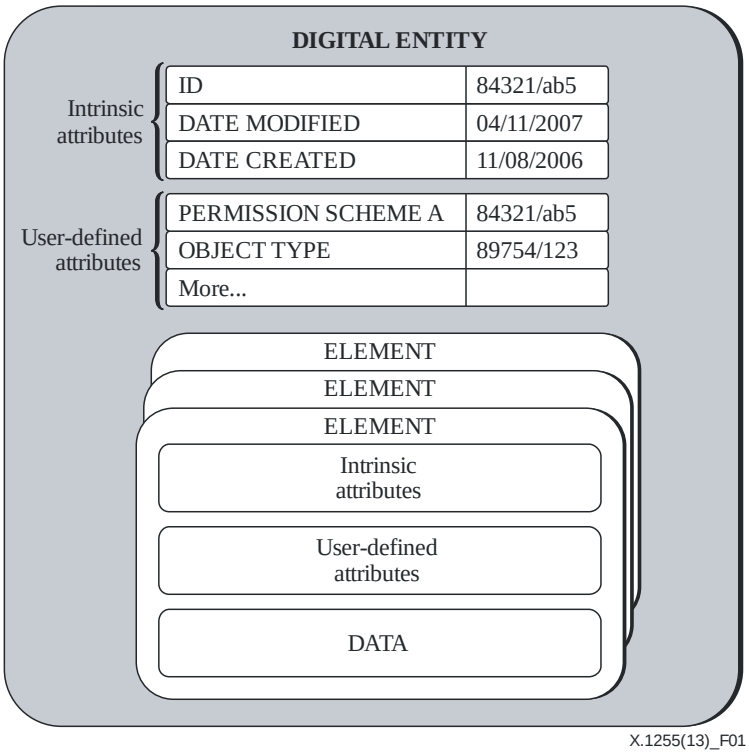

Figure 1 – Illustrative example of a digital entity

Except for the persistent identifier at the top, all data shown in Figure 1 is conceptual only. Each element of a digital entity can take different forms, i.e., digital entity references by identifier, an actual digital entity, plain local data suitably typed.

Registries may use or incorporate repositories to store metadata records; and repositories are information management systems that provide access to collections of DEs via the digital entity interface protocol. Repositories may generally be thought to incorporate the digital entities to which they provide access. A more detailed view however, would show them as portals into various storage and information systems, mapping the raw data into digital entities that may be stored locally or remotely. This could be as simple as a file system holding the data for a given DE in one or more files that are not known or visible to the user. Alternatively, especially for complex digital entities, data may be spread across multiple locations and systems and brought together in DE form only on demand, with one storage component holding the 'map' of the entity and the bulk of the data held in other systems. This technique of interacting with existing systems is key to federation, as the information in an arbitrarily complex information system can be logically divided into DEs, and those DEs made available in a standardized fashion, using an instance of a DEIP within user-centric applications.

A DE client can locate one or more repositories for a given DE by resolving its identifier. The resolution request will return the location of one or more relevant repositories with which the client can initiate a DE transaction.

The DE repository software normally provides multiple network interfaces for performing operations on digital entities, namely, the digital entity interface protocol for interacting with the DE itself, as well as locally desirable interfaces as determined by current technology options. The various interfaces each have their own benefits in terms of security, compatibility with proxy servers and the use of ubiquitous client software. Redundancy is built into the digital entity interface protocol, along with strong individual and group authentication. Redundancy is supported by a mirroring system in which each DE repository communicates with the others to ensure that

replicated entities are kept in sync. Authentication is based on either secret or public/private keys or other authentication mechanisms.

Other notable features include replication, allowing easy mirroring across repositories and extensibility through a plug-in mechanism. Plug-ins could be built to manage both entity type-specific activities, e.g., parsing a video format and dispensing a requested section, or activities oriented to network services, e.g., contributing metadata to a DE registry.

7.2 Digital entity interface protocol

Each interaction with an instance of a DE consists of invoking or applying an operation on the DE. Identity management information about the entity, each operation, and the target of the operation are all uniquely and persistently identified. In addition, resources of various kinds are identified entities, and the relevant state information about the resource may include, among other entries, its public key.

Operations are applied to digital entities by repositories which are themselves digital entities, and which provide access to the entities they contain. The digital entity interface protocol defines the method by which the entity communicates with a repository for the purpose of invoking operations on the digital entities for which the repository provides access. These operations can be used, in particular, to access specific metadata records by their identifiers; but such records can also be accessed semantically through other means such as dedicated registry "apps" and web browsers.

An operation on a digital entity involves the following elements:

- EntityID: the identifier of the digital entity requesting invocation of the operation;
- TargetEntityID: the identifier of the digital entity to be operated upon;
- OperationID: the identifier that specifies the operation to be performed;
- Input: a sequence of bits containing the input to the operation, including any parameters, content or other information; and
- Output: a sequence of bits containing the output of the operation, including any content or other information.

Identity management information may be accompanied by or communicated as part of a certificate that makes an explicit or implicit trust assertion about the information. However, the recipient may or may not accept the certificate if it was not created by an acceptable trust authority. Tokens may also be used in place of certificates to affect a similar trust result. Such certificates or tokens increase the likelihood of accurate identity information being conveyed; however, intrinsic security mechanisms that may be implemented as part of this open architecture can independently validate that the entity making use of identity management information possesses the appropriate private key which can be used to validate the identified digital entity. Either party to a transaction request involving identified digital entities may request the other party to encrypt a string with its private key and return it to the requesting party for validation. The parties to any transaction within the system may invoke other means of authentication, but there is no *a priori* need to negotiate such other means. The default mechanism indicated below is to use public/private key pairs, which is an integral capability of an instance of the DEIP. However, other authentication mechanisms may be used, if desired, by agreement of the parties. Invocation of an instance of the DEIP shall entail, at a minimum, the following non-optional steps:

a) Establishing an association between party A and party B, namely the two parties to the transaction, unless one already exists that they can use for this purpose.

b) Optionally, party A may request party B to validate itself to party A, for example, by using a PKI method.

c) Party A then issues a specific request to party B, as appropriate.

d) Optionally, party B may request party A to validate itself to party B, for example by using a PKI method.

e) Party B fulfils or denies the request, as appropriate.

f) The transaction is ended and either a new request is generated or the association is terminated, if appropriate.

A possible example of a specification of a DEIP is described at [b-DOIP] (see also [b-DO Repo]), but it is not a formal part of this Recommendation.

7.3 Interactions with a registry

Each interaction with a registry involves an identified digital entity that may be an individual or a system resource, and each has a persistent identifier that can be used to authenticate the digital entity. During setup, a registry can be pre-configured to trust any client or clients identified in a specific way via their identifiers. Clients can also choose to authenticate registries using the same procedure. Furthermore, specific clients can be configured to operate as specifically required in a federation process. This would allow a specific operation on the registry in addition to the ones that are commonly available to all trusted clients. When clients interact with a registry, the registry issues a challenge-response to verify that the client has the matching private key. Once this is verified, the registry verifies that the identifier belongs to the digital entity.

The following operations are supported by the registry interface:

- **Register a digital entity**: The registration information may consist only of metadata, but it can also be metadata combined with a DE to which the metadata applies. The registry manages the registered digital entity using its internal repository. Furthermore, the registry indexes the information structured as a digital entity using pre-configured rules that determine how to parse, tokenize and index the held information. When required, the registry creates an identifier for the digital entity and causes it to be inserted into the resolution system.
- **De-register a previously registered digital entity**: The registry deletes the digital entity from its internal repository, de-indexes it, and updates the resolution system to record the delete status of the entity.
- **Retrieve a previously registered digital entity via its identifier**: The registry serializes the digital entity managed in its internal repository and forwards it to the client.
- **Search**: The registry parses the search expression for keywords, exact matches or range queries to match against indexed digital entities, and returns the identifiers of matched digital entities. More advanced searching techniques, such as natural language queries, can be easily integrated, if research results permit.
- **Get latest transaction number**: The registry, which assigns numbers in a serial fashion to each register and de-register operation performed on it, returns the last such number to a client that is configured to participate in a federation process with the registry. This would allow potential clients (other registries participating in the federation process) to determine the state of the registry in order to push registered entities depending on the configured federation topology and chosen aggregation level.

Although authentication can be turned off, it is advisable that the registry authenticates the client and vice-versa. The encoding of exchanged messages can vary and is an implementation detail. Messages can be encoded as digital entity repository operations, at which point a register transaction will be a series of repository operations, e.g., create digital entity, add element. Alternately, messages may be encoded using third party data-encoding libraries, provided both the source and recipient agree (probably in advance) on the use of the same library.

7.4 Resolution systems

A component of the framework is the resolution system (of which there can be more than one) that can map identifiers to useful state information about the digital entity being identified, such as its location in the Internet, or authentication information for that digital entity, or a public key associated with the identifier. The open architecture nature of the framework enables the interoperability of resolution systems, which is a desirable objective of this Recommendation. The state information can be changed as needed to reflect the current state of the identified digital entity without changing its identifier, thus allowing the identifier of the item to persist over changes in location and other related state changes.

If the identifier of a needed resource is known, the resolution system and set of repositories provide what is needed for an authorized end user or process to view or otherwise access the digital entity. When the identity of the needed resource is unknown however, it will have to be discovered. In library and information science terms, the first case is called a "known-item" search (i.e., you know what you want and need to know how to get it). The second case typically requires a subject search and the goal of the tools used in a subject search is to reduce it to a known-item search. The digital entity registry enables this role to be performed.

While an instance of a registry can operate stand-alone, it can only satisfy discovery requests that it knows about. By federating multiple registries, it can know about digital entities registered elsewhere, and, thus, a broader search is possible across the entire collection of digital entities. The ability to determine which registries may contain relevant identity-related information is an important aspect of discovery of identity management information. Information available in one system may need to be discovered by another system, possibly of a different design. Assuming some entity has defined a way to associate such different systems and the information they contain, then the discovery framework should enable such associations to be discovered. However, this Recommendation does not discuss who or what is responsible for making such associations, what kind of information can be associated, or how the association is made and accessed. These issues will, in general, vary from context to context, and thus this Recommendation does not propose any kind of association practices. To clarify this matter, the term "association" is added to the definitions and the concept has been included in the definition of identity management information.

In many cases, privacy is critical and this is managed through the use of IdM techniques based on identifiers for individuals, groups, roles and resources, as well as the terms and conditions that are obtained from stored metadata.

7.5 Distributed queries and aggregated metadata in federated registries

The system of federated registries set forth in this Recommendation should be widely accessible and forms the basis of an open architecture discovery system. The system provides a uniform way to discover identity management information. A system of federated registries allows multiple IdM providers to participate in the provision of interoperable registries and to determine what information they are willing to share with other registries.

The registry technology provides a means by which the parties responsible for creating digital entities in the Internet, including services and other entities, can register the existence of a given set of such entities, accompany that registration with descriptive and structural metadata about the entities, including provenance information, and thus enhance the discoverability of the entities either for the public at large or for a defined community. One key piece of metadata that must be registered with the digital entity is its persistent identifier, and each such identifier must be resolvable in the Internet. For those entities not already identified, the registry can be configured to create identifiers as part of the registration process and provide the tools needed for the digital entity administrators to maintain the resolution information.

The system of federated registries will achieve four major objectives for discovery. First, it will enable uniform selection policies to be applied across participating trust frameworks and other IdM systems. Second, it will enable a user to access the registry information that the user is permitted to access without having to deal directly with multiple registries. Third, it provides infrastructure support for privacy and other access restrictions set by individual IdM systems. And fourth, it will enable semantic access to the registries to support multilingualism.

The concept of federated registries is based on the open architecture set forth herein that offers the following benefits:

- Unified selection policies: Registries and their associated trust frameworks or other IdM systems more generally, can be selected for querying based on properties of the information they purport to contain. An IdM system that does a certain level of background investigation to substantiate its information would normally be selected. Alternatively, a minimal trust framework or other IdM system that only checks credit card information or drivers' licences can be selected. At another extreme, an IdM system that does DNA testing of individuals may be selected. An organization that maintains policies to ensure the integrity of systems can be selected. By these means, a uniform method of selection can thus be applied across the universe of registries and their associated IdM systems.
- Shared metadata: The information made available in a system of federated registries is referred to as shared metadata. A generic template is associated with shared metadata, which identifies how the metadata is represented, and thus how it can be accessed for subsequent processing. The template does not contain specific entries.
- Federated access: If one registry does not have the information desired it may be accessible from one or more other registries. Indeed, in normal operation, the system will function so as to make such information readily available to the user independent of which registry might have contained it. Such access may be enabled by a variety of means, including hierarchical federation and peer-to-peer systems.
- Private access: Some registries will be restricted to certain user groups, application types, or roles associated with making use of the system, and some registries may be open to all. A means of restricting access to the discovery information based on criteria for such restriction is integral to the system. One or more mechanisms, agreed to by participating registries, will be used to maintain privacy within the system.
- Semantic access: A typing system is used to interpret inserted "types". IdPs can designate types of their own choosing in accordance with specification guidelines. This will enable semantic access to relevant information regardless of where it may be stored in the system, and it can help with multilingual requirements.

Metadata in such a system is available as structured data, with an associated unique persistent identifier that exists as long as the digital entity exists. Metadata from various registries may differ in subject area and/or metadata schema, making it difficult to provide a simple yet coherent search across the aggregation of all the records. If the metadata from different schemas or subject areas is reduced to a lowest common denominator schema, which is one solution to aggregating this type of data, then an optimal search strategy may be to identify those registries that would be the best candidates for a more detailed search. The transformation to the lowest common denominator schema could be done by the source registries or by the collector registry. Alternatively, the search itself could be mapped in some fashion to query the various metadata schemas appropriately, resulting in a set of queries that branch out from the original.

While aggregating metadata entities for discovering information from multiple domains is one possibility, issuing distributed queries across multiple registries each managing metadata entities of its domain is another. The landscape of federation across registries includes various other possibilities, as illustrated in a three-dimensional space in Figure 2.

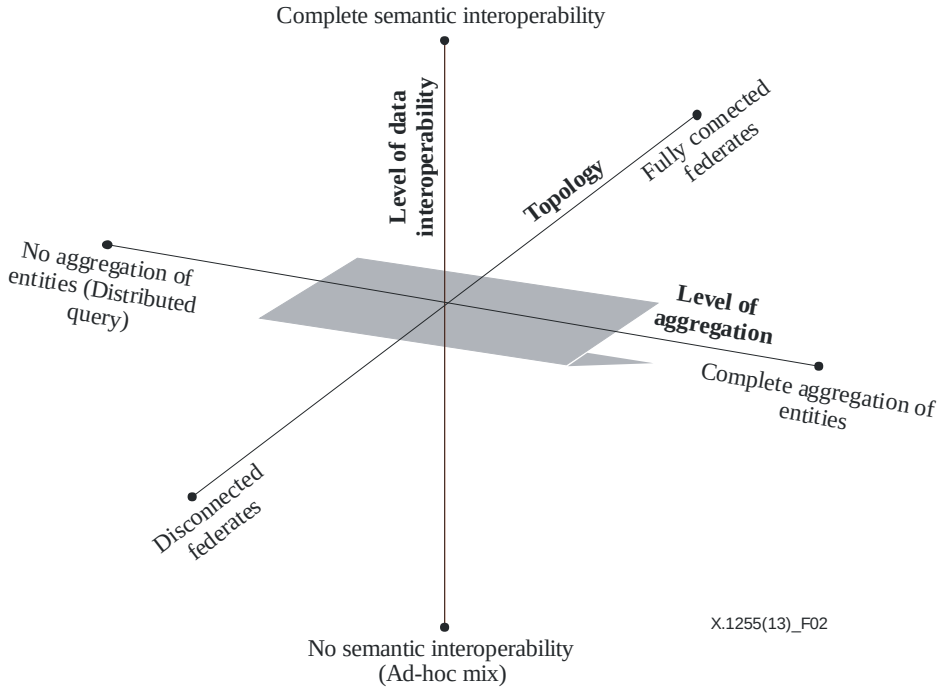

Figure 2 – Distributed queries across multiple registries

In Figure 2, there are three axes shown. One axis indicates the level of aggregation of registry metadata from no aggregation to full aggregation. The second axis indicates the degree of topological connectivity between the registries. The third axis relates to interoperability of the information from different registries. Each of these axes is described further below.

The level of aggregation axis indicates the degree to which a prior arrangement is made across registries to engage in the aggregation of metadata entities. The extreme left point on the axis indicates entities that are not aggregated across registries ahead of any query, while the point on the extreme right on the axis indicates that all entities are aggregated ahead of any query. Points along the axis represent other possibilities including aggregation of least common denominator metadata information, aggregation of search indices etc. As we move from left to right, the currency of the aggregated information is also reduced; the distributed query produces more up-to-date results, while the currency of the aggregated metadata entities depends on when they were aggregated last.

The topology axis indicates the degree to which the registries are connected. On one extreme end of the axis, the registries have no network connectivity to each other, thus resulting in no information sharing; the other extreme end indicates that the registries are fully connected to each other. Note that "how" they are connected is still determined by the level of aggregation, and topology only determines the possible linkages.

The level of data interoperability axis indicates the degree to which metadata entities from a registry that caters to a certain information domain is interoperable with metadata entities from another registry catering to a different information domain. In other words, the metadata schemas adopted by one registry may or may not be interoperable with schemas adopted by the other registry. Sometimes, transformation of the metadata entities is necessary to achieve a certain level of interoperability, if not full interoperability. In other cases, when the schemas are too far apart semantically, no amount of transformation will achieve a useful level of interoperability.

Note that not all points in the three-dimensional space shown in Figure 2 are valid. For instance, a distributed query on disconnected nodes implies no distribution of queries at all. Likewise, complete aggregation of non-interoperable entities implies a system of incoherent entities. To the extent that the points in the illustrated three-dimensional space are valid, the basic design of the registry should allow for such configuration possibilities, Also, whether the mapping is done in the

transformation of the metadata records or in the search, and whether the metadata records are transformed by the contributing registries or by the collecting registry, are all implementation details. There may be significant performance consequences, but the basic design should allow for implementation variations.

The approach taken in this Recommendation does not in and of itself solve the problem of search and retrieval across heterogeneous information systems, but it does provide a common framework in which different approaches can be used. Indeed, it is likely that there is no single solution to the problem and that the optimized approaches may vary with community of practice and subject area.

7.6 Metadata schemas

A major objective of this Recommendation is to provide a basis for defining a set of "high-level" metadata schemas to support the discovery of information on: a) identifiers used in various IdM systems; b) identity providers; c) relying parties; and d) trust frameworks and other IdM systems at all levels, including policies, procedures and underlying technical infrastructure. The necessary elements of these metadata schemas will be driven by specific usage scenarios, but will have to be extensible at both the element and the schema level in order to support growth and change in a dynamic area.

The various entities involved in identity management can each define their own specific schemas and, as needed, map them into these high-level standardized metadata schemas to describe their services, policies and procedures, and register these descriptions in one or more of a set of federated registries. These registries would support discovery services across the registered entities.

While it is possible that a single metadata schema could be created to accommodate all aspects of IdM technologies, relevant organizations and associated policies and procedures, it is proposed to begin with a single schema for each type of entity involved. The process of arriving at an agreed upon set of metadata schemas will become a collaborative process in which the interested parties will contribute their knowledge of the attributes which must be covered by the schemas; the evolving schemas can then be tested against various usage scenarios to see if they do indeed provide the needed information to support the discovery processes, and could then be augmented, if appropriate.

7.7 Metadata interoperability

Identifiers are one important ingredient to achieving metadata interoperability. However, certain other aspects of metadata interoperability, including those involving human definition and context of descriptions, are outside the scope of this Recommendation. Other attributes specified in metadata, such as those describing or enabling a particular configuration e.g., a specific connection mode and aggregation approach fall within the scope of registry operation. For the purposes of managing metadata entities across various registries, metadata interoperability will be facilitated if the collaborating parties decide on common metadata schemas. Metadata, then, will be managed as homogeneous entities, with registries interpreting and processing them in a consistent fashion. Clause 9 below illustrates two specific federation cases within the context of level of aggregation and topology, two dimensions that are typically applicable to this framework.

8 Types and type attributes

Registries provide metadata records in the form of digital entities that are intended to be exchanged with other federated registries. Each such record consists of a set of elements, each of which contains a "type" field and a "value" field. Understanding the meaning of each type is critical to manifesting the associated values in a form other than an opaque sequence of bits, or set of sequences of bits.

To understand what a type means, types are represented by persistent identifiers that may be resolved to useful information about the type. While the description of types is intended to be created by individuals, a standard means of describing and representing types is needed.

The specific aspects and attributes of what will ultimately constitute a type definition are expected to evolve over time, but the following four aspects are considered essential:

- The first category of attributes is the simplest and consists of human-readable descriptions of the purpose of the type. These descriptions are intended to describe the purpose of the type, the resources and concepts it describes, and its usage. These attributes will support descriptions in multiple languages.
- The second category of attributes of a type description consists of its provenance information. Every type definition should include its date of creation, last updated date, its contributors, its status and any alias identifier that it might have.
- The third category of attributes is related to describing the categorization of types, as well as the ability of types to leverage other types.
- The fourth category of attributes provides various systems with the ability to dynamically act on a resource of specific type.

The last three categories are described in more detail below.

A type is typically used to describe a specific category of resources and/or concepts according to a specific set of characteristics. This category represents a type's domain of applicability and is called a type's genre. For example, a character encoding type used to specify how a character is to be represented in a binary format would have an encoding genre. A data formatting type used to specify how to represent a structure as a set of bits would have a formatting genre.

Every type description will include an attribute that specifies its genre. The type genre description attribute provides a simple classification scheme that will normalize the development of new types and help type users discover existing types. A type genre is itself a type and new type genres can be added as needed to extend the type classification.

To maximize the reuse of types and to minimize the creation of duplicates, each type will be able to describe itself in terms of existing types. If, for example, a new type needs to specify that its resource is serialized in XML, it should do so by including a reference to the existing XML serialization type. Types can leverage other types by extension or by instantiation.

Each type should include any and all of the types that it leverages and how it does this. The ability of types to define themselves in term of other types will not only reduce the duplication of types but will allow type users to determine their understanding of a particular type with more granularity.

Finally, a type description should enable various systems to dynamically acquire the ability to act on any typed resource. The type description should include attributes that specify the location of network service bindings and/or specific module implementations, their platforms, and their associated interfaces. This will allow a generic type-processing library to dynamically and securely bind to such service, or to acquire, load and run the type's respective implementation module and process the resource.

Types, as discussed above, are uniquely identified. Resolving these type identifiers in some pre-defined resolution system will return a type record. An example of a BNF notation for a type record is shown in Appendix II that conceptually defines the group of entities that form such a type record.

Four sections are minimally required to unambiguously and coherently define a type, namely description, provenance, genre and processing.

The description section is a sequence of one or more human-readable descriptions that define the purpose and use of the type among other things. The language, which may conform to [b-IETF RFC 1766], in which these descriptions are made, shall be uniquely represented by its type and precede the descriptions.

Provenance captures the creation data, last modification date, contributors, aliases (or alternate identifiers) and status. Dates should conform to the [ISO 8601] standard. Contributors are names of personnel or organizations that contributed to the creation or registration of a type in a designated type registry. Aliases are captured to reference their prior registration in other local type registries that are declared here for the purposes of establishing the context of the defined type. Status identifies whether the type is in use or deprecated or obsolete.

Genre captures the essence of a type. Defining new types based on existing ones is a powerful notion that is key to defining complex types. A final notion regarding genres is to specify whether the genre information is a building block for defining other types or is defining a particular manifestation of a type. For instance, a binary encoding type by itself is a building block that allows defining other types.

Information may be passed to a service that knows how to parse and process the type information. Clients invoke a service to synthesize the given information. The service definition should identify where to reach the service, how to invoke the service, and what to expect as results from that service. No particular notation is recommended for defining such a service.

9 Hierarchical federation and peer-to-peer federation

In a hierarchical approach, a master registry is used to track the information held in multiple registries for convenience so only a single registry need be addressed. There could be multiple master registries, but they would all have to be known and consulted for a complete search.

In a peer-to-peer approach, certain registries choose to peer with selected other registries. The reasons for selected peering arrangements will vary. Organizational policies surrounding the management of registries, trust policies that prohibit or support federation touch points between registries, and availability/reliability of registries participating in a P2P network are a few of the reasons that could determine which registries are selected for peering by a given registry.

Both hierarchical and P2P formations however, imply only the topology of federation, and do not determine the level of aggregation chosen for either of the scenarios. While a variety of aggregation levels are applicable, two specific examples are given below for the purposes of illustration. Table 1 highlights the pros (indicated with a plus sign) and cons (indicated with a minus sign) of the two federation systems when either metadata entities are completely aggregated beforehand or queries are propagated real-time across registries in response to an IdM system query.

Table 1

	Hierarchical federation	Peer-to-peer federation
Complete aggregation of metadata entities at the collector registry	+ Complete cross domain discovery achieved through definitive aggregation process + Guaranteed relevance of cross domain information achieved through normalization of metadata entities during aggregation + Efficient performance due to localized search and retrievals – Rigid formation. Requires thorough setup processes that may interfere with organizational policies – Single points of failure, either at the master collector, or at intermediary collectors – Possibilities of obsolete information introduced by slow aggregation refresh rates – Possible scalability issues at the highest level in the hierarchy	+ Allows for flexible, non-rigid, groupings that cater to specific areas of interest + No single points of failures, as multiple routes for federation may be enabled + Guaranteed relevance of cross domain information achieved through normalization of metadata entities during aggregation + Efficient performance due to localized search and retrievals – No guarantee of completeness of cross domain discovery, unless formation is fully connected – High cost de-duplication efforts are required when registries can federate via multiple routes – Security concerns unless all touch points are trusted.
Query propagation across registries	+ Currency of metadata entities and information in these entities + Scalable system – Completeness of cross domain discovery is not guaranteed due to a likely non-availability of registry nodes at the time of query propagation – Relevance ranking of results is compromised due to runtime merge of results – Rigid formation. Requires thorough setup processes that may interfere with organizational policies – Single points of failure, either at the master registry node, or at intermediary registry nodes that propagate queries downward and pushes results upward – Performance issues due to non-robust hardware used for registry deployment	+ Currency of metadata entities and information in these entities + Scalable system – Completeness of cross domain discovery is not guaranteed due to a likely non-availability of registry nodes at the time of query propagation, even with redundant routes of federation – Relevance ranking of results is compromised due to runtime merge of results – High cost de-duplication efforts are required when registries can federate via multiple routes – Performance issues due to non-robust hardware used for registry deployment

Registry software supports and allows different combinations from the matrix shown in Table 1. Some combinations present greater challenges to implement than others. Scalability issues are addressed by the use of repository technology that abstracts the actual storage systems, and allows the simultaneous use of multiple storage systems. Replication and load balancing of registries is also provided, which alleviates the scalability issue. Duplicate detection, which would otherwise be

a problem in many-to-many relationships among registries, can be greatly alleviated through the use of persistent identifiers.

Data aggregation, as opposed to distributed query, assumes that the registry that initiates the movement of metadata records is pushing them, as opposed to responding to received requests. Also, the registry supplying records is referred to as the source and the recipient is referred to as the recipient. In federation, the recipient would be the touch point for the system of federated registries. The source registry forwards successfully executed changes in metadata, e.g., creations or edits of metadata records, to the recipients. These transactions cover changes to the state of a digital entity, namely, create, modify, alias, delete, as well as add/remove/replace relationships. Each registration record submitted to a registry is translated into register and de-register actions inside the registry core. Each such action is a transaction, which has a transaction identifier – a number that would normally get incremented starting from zero. This approach is equally applicable to scenarios in which registries are arranged in a peer-to-peer fashion as well as in a hierarchical fashion.

Configuring individual registries to target and propagate queries to selected registries, whether the formation is hierarchical or P2P, enables query propagations. Digital entity registries, in addition to supporting community specific interfaces, also support the digital entity interface protocol as a default interface. Queries can be propagated to other registries based on the use of this protocol.

Appendix I

Scenarios of usage

(This appendix does not form an integral part of this Recommendation.)

To illustrate the usage of a system of federated registries, a few scenarios would be helpful. A few scenarios are described below along with possible attributes that would need to be registered to enable the discovery process(es) to work acceptably.

- A client, either human or machine, would like to obtain a service from a service provider in the Internet. The service provider requires proof of identity and accepts identity credentials from any of a set of identity providers. The client (generally software) must be able to determine which IdPs are acceptable, whether or not the client already possesses the relevant credentials from one or more of the accepted IdPs, and, if not, how to obtain them.

  The service provider must advertise which IdPs are accepted for the relevant service(s). This could either be done directly by the service provider, in a standardized fashion, or by reference to a registry, also in a standardized fashion. In either case, the IdPs must be uniquely and precisely identified. The client can then match current participation in an IdP organization or otherwise have knowledge of how to match the IdP requirements, and present the relevant credentials to the service provider. In the case of direct advertising by the service provider, with unique and precise identification of the relevant IdP(s), and with client ability to provide IdP-specific credentials, no registry would be required. In all other cases, however, some level of information about the accepted IdPs must be discoverable. Given a unique and persistent identifier, this can be a direct look-up in a registry of IdP particulars. To meet the requirements of this usage scenario, the metadata describing a given IdP would need to provide clients with the information needed to determine if a given IdP was a reasonable choice for their use of the given service. Relevant attributes would include the unique persistent identifier of the IdP itself, for potential cross-referencing e.g., to review sites, trust frameworks in which the IdP participates, policies and procedures, legal requirements, required software, any fee schedules and so on. Some of this information would be in the form of a second level of indirection, e.g., many of the technical and policy details for any given IdP would be defined by the IdP's participation in one or more defined trust frameworks.
- The same details that would allow a client to discover the appropriateness of a given IdP would also allow a service provider to discover one or more IdPs whose credentials they would accept, and could thus be added to their list of acceptable IdPs. The IdP metadata would cover both of these use cases.
- In the reverse of the first scenario, a client accesses a service and presents an identity credential that the service has never before encountered. On the assumption that this credential presentation consists of, or at least begins with, an identifier for the IdP, the service must decide whether or not to accept the credential, to further investigate the possibility of accepting such a credential, or to simply reject it without further investigation. The registry in this scenario, would have to discover general information about the type of identifier and the IdP that it represents. This could in turn, lead to further registry look-ups on specific technologies used by the IdP, including relevant trust frameworks.
- The various entities involved in identity management, either explicitly or implicitly, would typically be members of one or more trust frameworks or other IdM system. Specification of the attributes of each IdM system would be required for the creation of a metadata schema describing that trust framework. Several important questions arise here. Is an IdM system described by the organization that provides it, a set of standards that implement it, a way of measuring compliance with standards, and so forth? Regardless of the answer to

these questions, it is clear that some IdPs and even some RPs would be usefully connected to higher-level descriptions, such as InCommon, Kantara, Safe-BioPharma and OIX organizations that would be discoverable within a registry or federation of registries.

In Figure I.1, we illustrate the way in which a system of federated registries (the "system") could be used with a specific usage scenario.

Step 1: In this example the end user requests a service from a relying party.

Step 2: The relying party responds with the identity of one or more trust frameworks that the RP trusts, in this case a single framework (TF1).

Step 3: The end user goes to the system with the identifier of TF1.

Step 4: The system responds with the relevant record for TF1. The information for TF1 includes the minimum attributes that will be required for trust within that framework.

Step 5: The end user evaluates these minimum requirements to determine whether it will be possible to gain the trust of the RP. Here we assume that the end user evaluation is positive and shows that the end user can meet the minimum attributes, e.g., driver's licence.

Step 6: The end user can now go back to the system requesting the IdPs that fall within TF1 and which can accommodate the protocol(s) which that end user supports e.g., HTTP and email, which we show here simply as protocol X.

Step 7: The system finds identity providers (IdPs) that match protocol X are within TF1.

Step 8: The system responds to the end user with the set of IdPs that match the requirements for both the relying party and the end user.

Step 9: The end user evaluates the set of IdPs returned from the system and makes a selection (IdP1).

Step 10: The end user, having the attributes required by IdP1 and speaking a protocol understood by IdP1, engages in challenge/response interaction with IdP1.

Step 11: A successful challenge/response interaction results in IdP1 delivering an authentication credential to the RP.

Step 12: The relying party, which now trusts the end users, delivers the requested service.

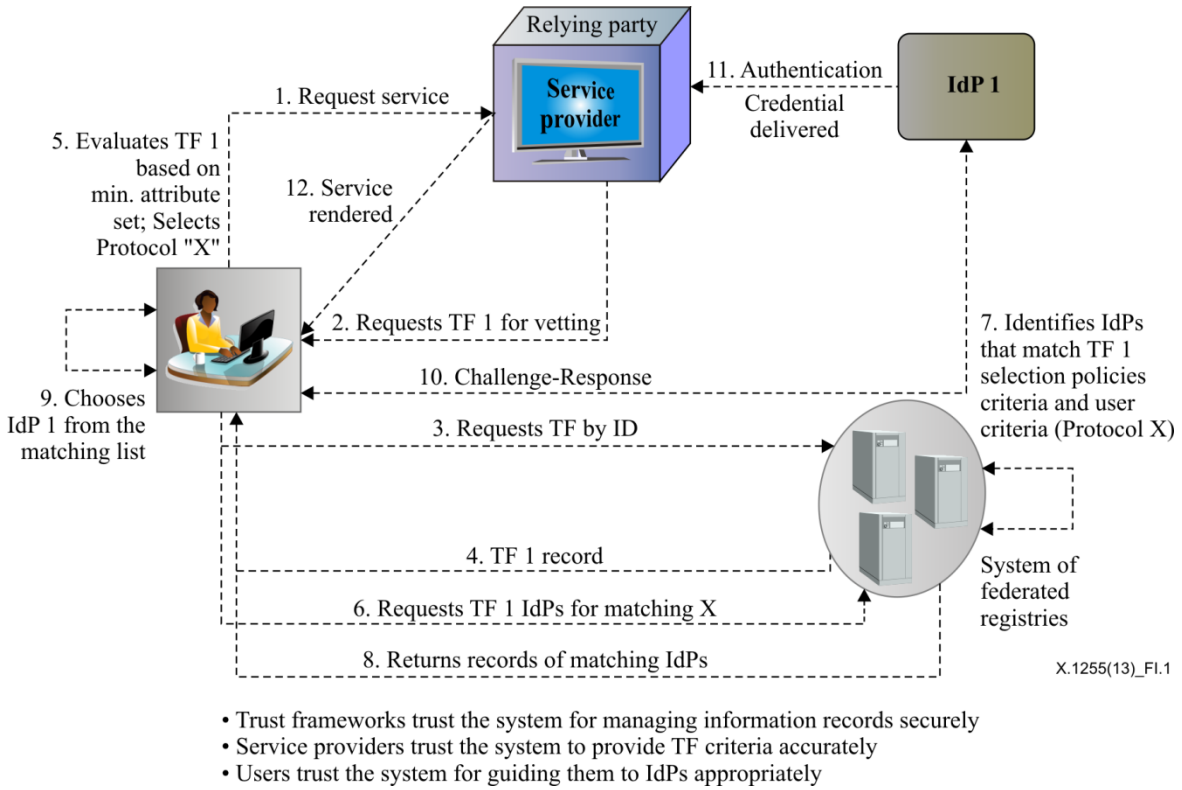

Figure I.1 – Authentication involving trust frameworks

In Figure I.2 we show high-level schemas of the records held by the system of federated registries that would enable the transaction described in Figure I.1, as well as the other usage scenarios described above. These records would be held by the system as digital entities, each with a persistent identifier. Each would have to be further built out into specific schemas in order for any prototyping to take place.

Each trust framework would have an identifier, a general description of the framework, the set of attributes used for authentication, and pointers to one or more IdP selection policies, which are themselves separate digital entities held in the system. These serve as an additional level of indirection such that all of the IdPs that fit within a single TF can be grouped by criteria instead of enumeration. Each IdP selection policy entity would have an identifier, a general description, a list of acceptable technologies (e.g., protocols supported), a list of organizational accreditation bodies, (e.g., a governmental organization), and any special operational constraints. The relationship between TFs and IdPs' selection policies would be many-to-many in both directions, i.e., a given TF could accommodate multiple IdP's selection policies and a given IdP selection policy could be used by multiple TFs.

The remaining two proposed entity types held by the system are IdPs and relying parties. Each IdP would have a persistent identifier, a general description, required user attributes, an evaluation policy for these attributes, specific protocols and accreditations accepted, and specific endpoints e.g., the location of the IdP in the form of the accepted protocols. Each relying party would have a persistent identifier, a general description, the set of TFs upon which it will rely, and any specific operational restrictions.

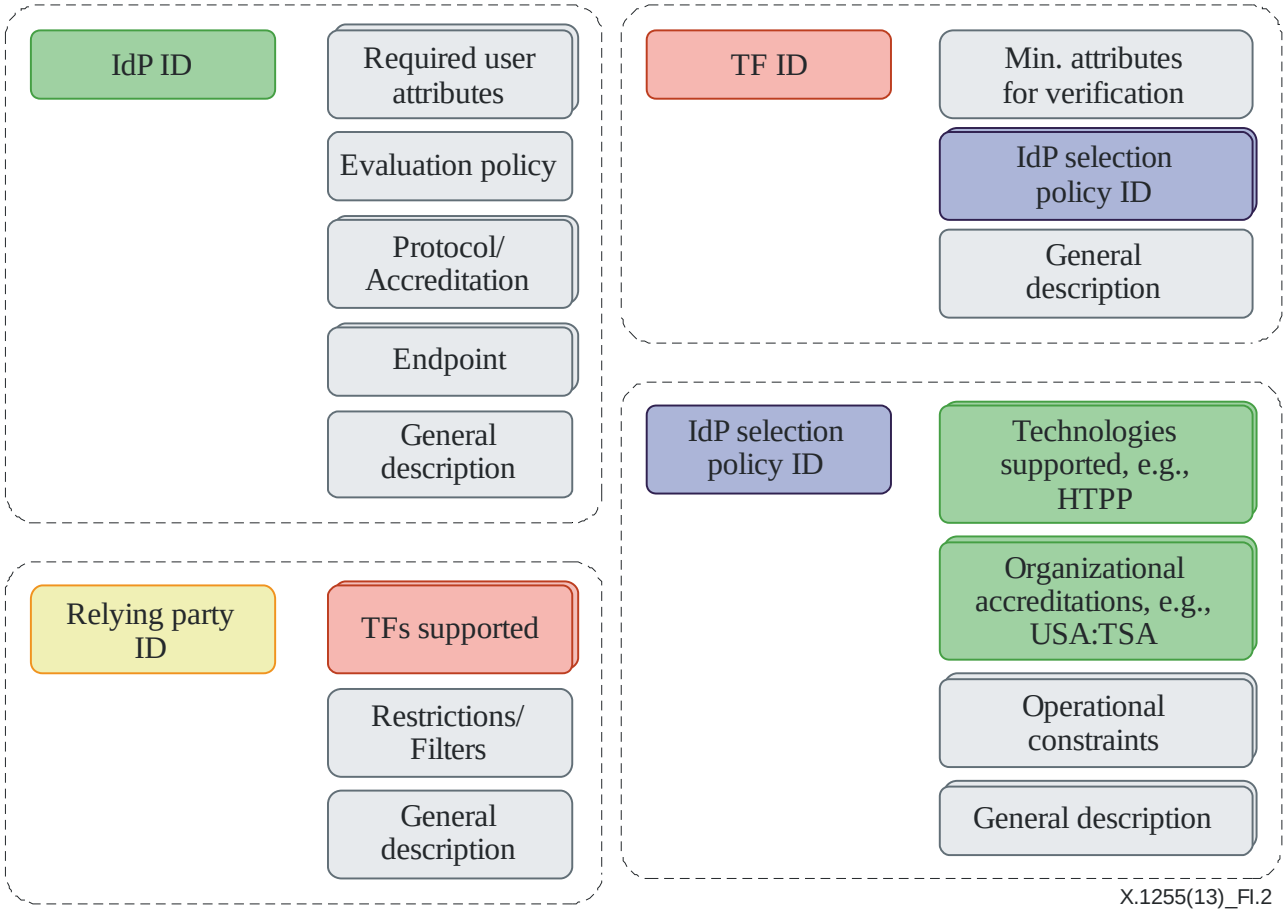

Figure I.2 – High level schemas

Appendix II

BNF notation for a Type record

(This appendix does not form an integral part of this Recommendation.)

BNF for a Type record:

```
<type identifier> := <unicode string>
<type> := <description section> <section delimiter>
    <provenance section> <section delimiter>
    <genre section> <section delimiter>
    <processing section>
```

```
<description section> := <language> '=' <human readable description>
    [<repetition delimiter> <description section>]
<language> := Any item from RFC 1766
<human readable description> := <unicode string>
```

```
<provenance section> := <creation date> <list delimiter>
    <last modified date> <list delimiter>
    <contributors> <list delimiter>
    <aliases> <list delimiter>
    <status>
<creation date> := Conforms to ISO 8601
<last modified date> := Conforms to ISO 8601
<contributors> := <unicode string>
    [<repetition delimiter> <contributors>]
<aliases> := <unicode string>
    [<repetition delimiter> <aliases>]
<status> := 'in use' | 'deprecated' | 'obsolete'
```

```
<genre section> := <genre> '=' <genre details>
    [<repetition delimiter> <genre section>]
<genre> := 'data structure' | 'encoding' | 'format'
<genre details> := <human readable description>
    [<list delimiter> <genre subsection>]
<genre subsection> := 'form='<form> <list delimiter>
    'relationship=' <relationship> <list delimiter>
    'related to=' <type identifier>
    [<repetition delimiter> <genre subsection>]
<form> := 'expression' | 'manifestation'
<relationship> := 'is equivalent to' | 'is derived from' |
    'is informed from'
```

```
<processing section> := <processor type> '=' <processor>
    [<repetition delimiter> <processing section>]
<processor type> := 'network service' | 'downloadable program' |
    'parsing function'
<processor> := <network service type> '=' <network service binding> |
   <compatible platform> <list delimiter>
   <program network location> <list delimiter>
   <program arguments> |
   <pseudo code>
<compatible platform> := 'Linux' | 'Windows' | 'Mac OS'
<program arguments> := <type>
     {<list delimiter> <unicode string>}
<pseudo code> := <unicode string>
```

```
<unicode string> := <visible character> [<unicode string>] |
    <whitespace character> <[unicode string>]
<visible character> = Any visible character in Unicode presumably encoded in
UTF-8
<whitespace character> := Any whitespace character in Unicode presumably encoded
in UTF-8
```

Notes:

1. <type identifier> issued to global resolution system resolves to a <type> record.
2. All delimiters, namely <section delimiter>, <repetition delimiter> and <list delimiter>, are implementation-specific details and purposely not defined here.
3. <network service type> is not defined here, but it should cover the popular network services as deemed fit by the implementing agency.
4. <network service binding> is also not defined here, but it should be based on the network service type. Actual definitions that conform to each of the service types should be stated here.
5. <program network location> is also not defined here, but it should identify the network protocol that the client should use to download the program from the network.
6. <compatible platform> may be extended or specified in more detail than defined here.

Bibliography

[b-ITU-T Y.2720] Recommendation ITU-T Y.2720 (2009), *NGN identity management framework.*

[b-IETF RFC 1766] IETF RFC 1766 (1995), *Tags for the Identification of Languages*. <http://www.ietf.org/rfc/rfc1766.txt>

[b-DO Repo] Reilly, S. and Tupelo-Schneck, R. (2010), *Digital Object Repository Server: A Component of the Digital Object Architecture,* D-Lib Magazine, Vol. 16, No. 1/2. <http://dx.doi.org/10.1045/january2010-reilly>

[b-DOIP] Reilly, S. (2009), *Digital Object Protocol Specification, Version 1.0,* Corporation for National Research Initiatives. <http://hdl.handle.net/4263537/5045>

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Terminals and subjective and objective assessment methods
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	**Data networks, open system communications and security**
Series Y	Global information infrastructure, Internet protocol aspects and next-generation networks
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2013