Business Document Metadata Service Location Version 1.0

OASIS Standard

01 August 2017

Specification URIs

This version:

http://docs.oasis-open.org/bdxr/BDX-Location/v1.0/os/BDX-Location-v1.0-os.odt (Authoritative)
http://docs.oasis-open.org/bdxr/BDX-Location/v1.0/os/BDX-Location-v1.0-os.html
http://docs.oasis-open.org/bdxr/BDX-Location/v1.0/os/BDX-Location-v1.0-os.pdf

Previous version:

http://docs.oasis-open.org/bdxr/BDX-Location/v1.0/cos01/BDX-Location-v1.0-cos01.odt (Authoritative)
http://docs.oasis-open.org/bdxr/BDX-Location/v1.0/cos01/BDX-Location-v1.0-cos01.html
http://docs.oasis-open.org/bdxr/BDX-Location/v1.0/cos01/BDX-Location-v1.0-cos01.pdf

Latest version:

http://docs.oasis-open.org/bdxr/BDX-Location/v1.0/BDX-Location-v1.0.odt (Authoritative)
http://docs.oasis-open.org/bdxr/BDX-Location/v1.0/BDX-Location-v1.0.html
http://docs.oasis-open.org/bdxr/BDX-Location/v1.0/BDX-Location-v1.0.pdf

Technical Committee:

OASIS Business Document Exchange (BDXR) TC

Chair:

Kenneth Bengtsson (kenneth@alfa1lab.com), Alfa1lab

Editors:

Dale Moberg (dmoberg@axway.com), Axway Software
Pim van der Eijk (pvde@sonnenglanz.net), Sonnenglanz Consulting

Additional artifacts:

This prose specification is one component of a Work Product that also includes:

- JSON example files*:* http://docs.oasis-open.org/bdxr/BDX-Location/v1.0/os/examples/

Related work:

This specification is related to:

- *Collaboration-Protocol Profile and Agreement Specification Version 2.0*. OASIS Standard. September 23, 2002. http://www.oasis-open.org/committees/ebxml-cppa/documents/ebcpp-2.0.pdf.
- *ebXML Collaboration-Protocol Profile and Agreement Specification Version 3.0*. Edited by Dale Moberg and Marty Sachs. 09 October 2009. Work in progress. https://www.oasis-open.org/committees/download.php/34606/ebcppa-v3.0-Spec-wd-r01-en-pete4.odt.
- *OASIS ebCore Party Id Type Technical Specification Version 1.0*. Edited by Dale Moberg and Pim van der Eijk. Latest version. http://docs.oasis-open.org/ebcore/PartyIdType/v1.0/PartyIdType-1.0.html.

Abstract:

This specification defines service discovery method values for use in DNS Resource Record service fields. A method is first specified to query and retrieve a URL for metadata services. Two metadata service types are then defined. Also an auxiliary method pattern for discovering a registration service to enable access to metadata services is described. The methods defined here are instances of the generic pattern defined within IETF RFCs for Dynamic Delegation Discovery Services (DDDS). This specification therefore defines DDDS applications for metadata and metadata-registration services.

Status:

This document was last revised or approved by the membership of OASIS on the above date. The level of approval is also listed above. Check the “Latest version” location noted above for possible later revisions of this document.

Technical Committee members should send comments on this Work Product to the Technical Committee’s email list. Others should send comments to the Technical Committee by using the "Send A Comment" button on the Technical Committee’s web page at https://www.oasis-open.org/committees/bdxr/.

This OASIS Standard is provided under the Non-Assertion Mode of the OASIS IPR Policy, the mode chosen when the Technical Committee was established. For information on whether any patents have been disclosed that may be essential to implementing this Work Product, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the Technical Committee web page (https://www.oasis-open.org/committees/bdxr/ipr.php).

Note that any machine-readable content (Computer Language Definitions) declared Normative for this Work Product is provided in separate plain text files. In the event of a discrepancy between any such plain text file and display content in the Work Product's prose narrative document(s), the content in the separate plain text file prevails.

Citation format:

When referencing this Work Product the following citation format should be used:

[BDX-Location-v1.0]

Business Document Metadata Service Location Version 1.0. Edited by Dale Moberg and Pim van der Eijk. 01 August 2017. OASIS Standard. http://docs.oasis-open.org/bdxr/BDX-Location/v1.0/os/BDX-Location-v1.0-os.html. Latest version: http://docs.oasis-open.org/bdxr/BDX-Location/v1.0/BDX-Location-v1.0.html.

Notices

All capitalized terms in the following text have the meanings assigned to them in the OASIS Intellectual Property Rights Policy (the "OASIS IPR Policy"). The full Policy may be found at the OASIS website.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published, and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this section are included on all such copies and derivative works. However, this document itself may not be modified in any way, including by removing the copyright notice or references to OASIS, except as needed for the purpose of developing any document or deliverable produced by an OASIS Technical Committee (in which case the rules applicable to copyrights, as set forth in the OASIS IPR Policy, must be followed) or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

OASIS requests that any OASIS Party or any other party that believes it has patent claims that would necessarily be infringed by implementations of this OASIS Committee Specification or OASIS Standard, to notify OASIS TC Administrator and provide an indication of its willingness to grant patent licenses to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification.

OASIS invites any party to contact the OASIS TC Administrator if it is aware of a claim of ownership of any patent claims that would necessarily be infringed by implementations of this specification by a patent holder that is not willing to provide a license to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification. OASIS may include such claims on its website, but disclaims any obligation to do so.

OASIS takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on OASIS' procedures with respect to rights in any document or deliverable produced by an OASIS Technical Committee can be found on the OASIS website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this OASIS Committee Specification or OASIS Standard, can be obtained from the OASIS TC Administrator. OASIS makes no representation that any information or list of intellectual property rights will at any time be complete, or that any claims in such list are, in fact, Essential Claims.

The name "OASIS" is a trademark of OASIS, the owner and developer of this specification, and should be used only to refer to the organization and its official outputs. OASIS welcomes reference to, and implementation and use of, specifications, while reserving the right to enforce its marks against misleading uses. Please see https://www.oasis-open.org/policies-guidelines/trademark for above guidance.

Table of Contents

1 Introduction

1.1 IPR Policy

This OASIS Standard is provided under the Non-Assertion Mode of the OASIS IPR Policy, the mode chosen when the Technical Committee was established.

For information on whether any patents have been disclosed that may be essential to implementing this Work Product, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the Technical Committee web page (https://www.oasis-open.org/committees/bdxr/ipr.php).

1.2 Terminology

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in [RFC2119].

1.3 Normative References

[DNSCORE] Mockapetris, P., **“**Domain Names – Concepts and Facilities”, RFC 1034, November 1987. http://tools.ietf.org/rfc/rfc1034

[DNSSEC1] Arends, R., Austein, R. Larson, M. Massey, D., Rose, S., **“**DNS Security Introduction and Requirements”, RFC 4033, March 2005. http://tools.ietf.org/rfc/rfc4033

[DNSSEC2] Arends, R., Austein, R. Larson, M. Massey, D., Rose, S., **“**Resource Records for the DNS Security Extensions”, RFC 4034, March 2005. http://tools.ietf.org/rfc/rfc4034

[DNSSEC3] Arends, R., Austein, R. Larson, M. Massey, D., Rose, S., “Protocol Modifications for the DNS Security Extensions”, RFC 4035, March 2005. http://tools.ietf.org/rfc/rfc4035

[ebCorePartyId] Moberg, D., Van Der Eijk, P. “OASIS ebCore Party Id Type Technical Specification Version 1.0. OASIS Committee Specification”, September 2010, https://docs.oasis-open.org/ebcore/PartyIdType/v1.0/PartyIdType-1.0.odt

[ebCPPA2] “Collaboration-Protocol Profile and Agreement Specification Version 2.0”. OASIS Standard, September, 2002. https://www.oasis-open.org/committees/ebxml-cppa/documents/ebcpp-2.0.pdf

[RFC2119] Bradner, S., “Key words for use in RFCs to Indicate Requirement Levels”, BCP 14, RFC 2119, March 1997. http://www.ietf.org/rfc/rfc2119.txt

[RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P.,Berners-Lee, T.,“Hypertext Transfer Protocol – HTTP/1.1”, RFC 2616, June 1999. http://tools.ietf.org/rfc/rfc2616

[RFC2782] Gulbrandsen, A., Vixie, P. Esibov, L., “A DNS RR for specifying the location of services (DNS SRV)”, RFC 2782, February 2000. http://tools.ietf.org/rfc/rfc2782

[RFC3401] Mealling, M., “Dynamic Delegation Discovery System (DDDS) Part One: The Comprehensive DDDS”, RFC 3401, October 2002. http://tools.ietf.org/html/rfc3401

[RFC3402] Mealling, M., “Dynamic Delegation Discovery System (DDDS) Part Two: The Algorithm”, RFC 3402, October 2002. http://tools.ietf.org/html/rfc3402

[RFC3403] Mealling, M., “Dynamic Delegation Discovery System (DDDS) Part Three: The Domain Name System (DNS) Database**”**, RFC 3403, October 2002. http://tools.ietf.org/rfc/rfc3403

[RFC3404] Mealling, M., “Dynamic Delegation Discovery System (DDDS)Part Four: The Uniform Resource Identifiers (URI) Resolution Application**”,** RFC 3404, October 2002. http://tools.ietf.org/rfc/rfc3404

[RFC3405] Mealling, M., “Dynamic Delegation Discovery System (DDDS) Part Five: URI. ARPA Assignment Procedures”, RFC 3405, October 2002. http://tools.ietf.org/rfc/rfc3405

[RFC3833] Atkins, D., Austein, R., “Threat Analysis of the Domain Name System (DNS)”, RFC 3833, August 2004. http://tools.ietf.org/rfc/rfc3833

[RFC3958] Daigle, L., Newton, A., “Domain-Based Application Service Location Using SRV RRs and the Dynamic Delegation Discovery Service (DDDS)”, RFC 3958, January 2005. http://tools.ietf.org/rfc/rfc3958

[RFC3986] Berners-Lee, T., Fielding, R., Masinter, L., **“**Uniform Resource Identifier (URI): Generic Syntax”, RFC 3968, January 2005. http://tools.ietf.org/rfc/rfc3986

[RFC4848] Daigle, L., “Domain-Based Application Service Location Using URIs and the Dynamic Delegation Discovery Service (DDDS)”, RFC 4848, April 2007. http://tools.ietf.org/rfc/rfc4848

[RFC5936] Lewis, E., Hoenes, A., “DNS Zone Transfer Protocol (AXFR)”, RFC 5936, June 2010. http://tools.ietf.org/rfc/rfc5936

[RFC6895] Eastlake, D., “Domain Name System (DNS) IANA Considerations”, BCP 42, RFC 6895, April 2003. http://tools.ietf.org/rfc/rfc6895

[URN] Moats, R., “URN Syntax”*,* RFC 2141, May 1997. http://tools.ietf.org/rfc/rfc2141

1.4 Non-Normative References

[ebCPPA3] Moberg, D. “OASIS ebXML Collaboration-Protocol Profile and Agreement Specification Version 3”. OASIS Working Draft, May 2007. https://www.oasis-open.org/committees/download.php/34606/ebcppa-v3.0-Spec-wd-r01-en-pete4.odt

[**ODataURL**] Pizzo, M,. Handl, R. and Zurmuehl , M. “OData Version 4.0 Part 2: URL Conventions”. OASIS Committee Specification, August 2013. http://docs.oasis-open.org/odata/odata/v4.0/cs01/part2-url-conventions/odata-v4.0-cs01-part2-url-conventions.html

[RFC1912] Barr, D., “Common DNS Operational and Configuration Errors”, RFC 1912, February, 1996. http://tools.ietf.org/rfc/rfc1912

[RFC2915] Mealling, M., Daniel, R., **“**The Naming Authority Pointer (NAPTR) DNS Resource Record*”,* RFC 2915, September 2000. http://tools.ietf.org/rfc/rfc2915

[RFC4398] Josefsson, S., “Storing Certificates in the Domain Name System (DNS)”,RFC 4398, March 2006. http://tools.ietf.org/rfc/rfc4398

[RFC4697] Larson, M.. Barber, P., “Observed DNS Resolution Misbehavior”, BCP 123, RFC 4697, October 2006. http://tools.ietf.org/rfc/rfc4697

[RFC5011] StJohns, M., “Automated Updates of DNS Security (DNSSEC) Trust Anchors”, RFC 5011, September 2007. http://tools.ietf.org/rfc/rfc5011

[RFC5625] Bellis, R., “DNS Proxy Implementation Guidelines”, RFC 5625, BCP 152, August 2009. http://tools.ietf.org/rfc/rfc5625

[RFC5731] Hollenbeck,S., *“*Domain Name System (DNS) Security Extensions Mapping for the Extensible Provisioning Protocol (EPP)”, RFC 5625, BCP 152, August 2009. http://tools.ietf.org/rfc/rfc5731

[RFC6781] Kolkman, O., Mekking, W., Gieben, R., “DNSSEC Operational Practices”, RFC 6781. http://tools.ietf.org/rfc/rfc6781

[SML] Sylvest, G., Andersen, J.J., Pedersen, K.V., Brun, M.H., Edwards, M. “PEPPOL Transport Infrastructure Service Metadata Locator (SML)”. February 2010.

https://www.oasis-open.org/committees/download.php/47488/ICT-Transport-SML_Service_Specification-101.pdf

[WSDL] Chinnici, R. Moreau, J.J., Ryman, A. Weerawarana, S. “Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language”. W3C Recommendation, June 2007. http://www.w3.org/TR/wsdl20/.

2 Business Interaction Metadata Services and Location Discovery

2.1 Introduction

A metadata service for business interactions provides information about what kinds of data transactions, and what kinds of enabling technologies for those transactions, are available for specific business process participants. While a web site that offers natural language “implementation guides” counts as a metadata service broadly viewed, in this document the focus will be on approaches that have emerged to automate the setup and life-cycle management of business interactions.

The location of a metadata service in this document refers primarily to a URL-specified endpoint identifier [RFC3986,2616] For the location of a human-readable and web-browsable document, a link in an email invitation, on a web page or in the results of a web search engine might provide the service location information. For an automated process, however, it is desirable to have a specific way to publish and retrieve location information. Ideally, the procedure for metadata service location would be as reliable and pervasive as the ability to find the numerical IP address for a given path of domain name system labels. Interestingly enough, the DNS system itself has specified record types that can store URLs [RFC2915,RFC4848], and whose use in building a location finding infrastructure for service location (of any kind) has been specified [RFC3401,3402,3403,3404].

The goal of this specification is to provide ways to find the location of a metadata service of a given type, to enable the use of information provided by that service to setup and start-up business interactions. Some specific services will be provided with standard designated values for use in DNS records; patterns for values of the DNS service field are provided to enable extension points for future community implementation standardizations or other standards bodies defining metadata.

2.2 Overview of the Core Service Location Discovery System

The goal of a Dynamic Delegation Discovery System (DDDS) application for metadata service discovery is to find URLs of specific types of metadata services, by using a DNS query string that represents an identity of a person or organization. In other words, the goal is to retrieve one or more URLs from the DNS that will enable finding out more about an entity’s enabled metadata services.

The framework for such a DDDS application sets out what kinds of information must be specified:

- A DNS query string which is a string of concatenated, dot-separated labels.
- Types of DNS records to be retrieved.
- Values for fields within DNS record types that are used.
- Processing steps to be taken.

Because the goal is to find URLs, the NAPTR RR with “U” value in its Flags field (U-NAPTR), provides the core framework for producing the desired DDDS application [RFC4848]. The service field of a U-NAPTR can distinguish distinct kinds of metadata services, or supporting services such as registration services to obtain access authorization for metadata services. DNS domain names of organizations or persons provide the core solution for query strings that may be used when trying to locate metadata services for persons or organizations.

So when “example.com” wants to announce the URL for metadata services, its IT services group can add a U-NAPTR record configuration such as:

```
IN NAPTR 100 10 "U" "Meta:CPPA" "!^.*$!https://example.com/cppa!" .
```

or

```
 IN NAPTR 100 10 "U" "Meta:SMP" "!^.*$!https://example.com/smp!" .
```

The service name (such as, Meta:SMP) identifies a kind of metadata service for the organization with the domain name “example.com,” and the URL obtained from the Regexp field provides a secured endpoint to contact. The Regexp field can be used by applying its value to the DNS label-path to carry out string manipulations that produce the URL. However, if the URL alone is needed, a vacuous match using a regular expression such as "^.*$" is applied with the URL returned without modification.

If a registration service is needed to arrange for authorized access, an additional NAPTR record can be added such as:

```
IN NAPTR 100 10 "U" "Register:CPPA" "!^.*$!https://example.com/register!" .
```

In essence, the above approach provides a simple but general approach to retrieving URLs for an organization’s metadata and registration services that is usable for both programmatic and human access.

2.3 Identifying Business Interaction Participants

2.3.1 Recommendations for Special Use Cases

A DDDS approach to metadata and metadata-registration service location obviously is most straightforwardly applied when the interacting participants all make use of DNS domain names associated with their businesses. This is because the business's domain name itself can serve as the business identifier supplied in the DNS query string when retrieving service locations.

However, there are several situations that benefit from more elaborate approaches to what query strings need to be used to retrieve service locations. For these situations, additional conventions are needed for converting other (non-DNS) identification formats into DNS query strings.

The special situations and purposes include the following:

- Service providers may have agreed upon “special” domains within which all metadata service endpoints will be located [SML]. In this case, information encoding the specific person or organization may need to be prefixed to the special domain names to form the DNS query string. The motivation for this is to allow multiple hosts for metadata or registration services, and to allow specific persons or organizations to announce their service hosts by U-NAPTR records within the “special” domains. The PEPPOL and GS1 networks use special domains, within which DNS addresses for service hosts can be published.

- Identity may be symbolically represented in any number of naming authority formats and values. The ISO registered naming authorities have been mapped into URNs in ebXML’s **OASIS ebCore Party Id Type Technical Specification Version 1.0**. [ebCorePartyID] There is, additionally, a defined BCP (Best Current Practice) RFC that specifies how URNs can be mapped to query strings. The resulting conventions can define one way to store NAPTR RRs for metadata service information.

- Identity may be associated with other identifying addresses, such as email addresses. Privacy of information concerns may require a specialized registration service for converting email addresses in the email domain to URLs for metadata services of the email identified domain.

- If privacy concerns allow use of DNS RRs, nevertheless, authorization for access to the metadata services may be requested. In that case, additional prerequisite registration services may be required so that the seeker of metadata information may be authorized through an authorization request to the “owner” of the metadata information provided by a service.

These more complex use cases will receive additional discussion.

2.3.2 Service Provider Domains

The PEPPOL network is based on a pattern wherein each business participant has one (or more) service providers; in this four-corner model, business document exchange does not require that there is any common “end-to-end” business document exchange protocol between a business sender and business receiver.

It is possible that the business customer of a service provider has its own DNS domain, and in that case the customer can add U-NAPTR records that provide the URL for the customer's metadata service located on a host in its service provider's domain.

It is also possible that the customer with a DNS name may not manage its own name servers, or has no convenient way to manage U-NAPTR records for metadata service or registration service locations.

It is then possible to place the metadata service location within the service provider's domain by creating a specialized DNS label for the customer. Such a path can encode a name and naming authority identifier within a prefix label for the DNS query path. Then U-NAPTR records can be retrieved for that prefixed DNS query string that return the URL for the metadata (or other) service.

In [SML], for example, the following URL pattern has been utilized:

http://<hash over recipientID>.<schemeID>.<SMLdomain>/<recipientID>/services/<documentType>

Furthermore, [SML] illustrates the hash over a recipientID as follows: "An example participant ID is "0010:5798000000001" ... for which the MD5 hash is "e49b223851f6e97cbfce4f72c3402aac". It is noted that the hash value is prefixed with a "B-" string.

Then a U-NAPTR for a DNS query string "B-e49b223851f6e97cbfce4f72c3402aac.sid.peppol.eu to a SMP metadata service hosted at "serviceprovider.peppol.eu" might be:

```
IN NAPTR 100 10 "U" "Meta:SMP" "!^.*$!https://serviceprovider.peppol.eu/e49b223851f6e97cbfce4f72c3402aac/!" .
```

Or, utilizing the regexp capability for group extraction from query strings,

```
IN NAPTR 100 10 "U" "Meta:SMP" "!^B-(+[0-9a-fA-F]).sid.peppol.eu$!https://serviceprovider.peppol.eu/\\1!" .
```

should yield the URL, "https://serviceprovider.peppol.eu/e49b223851f6e97cbfce4f72c3402aac". (Implementations should beware of varying server-side implementations of regexp backslash escaping.)

Notice that U-NAPTR URLs are not really intended to provide detailed URL paths to specific resources, or to add details that might be found in URL query parameters. Instead a URL provided by the NAPTR is one that provides the "service root" for some service type. The service details may then be appended to ther service root as path elements and/or query parameters. The conventions for service details may be generic REST ones, or they may have a machine processable description ([WSDL] or the like), or have a data model for interaction that supports a system of URL conventions for access to data records [ODataURL]. In other words, either paths and/or query parameters may need to be appended to the service root URL when actually using the service located at the U-NAPTR service location URL.

Since PEPPOL location URL templates combined a base service root with additional path information, retrieving the service location from a U-NAPTR record should be regarded as only providing the service root. More detailed URL path or query parameter conventions can then be separately specified in accordance with one of the previously mentioned approaches to service specification details.

2.3.3 Non-DNS Participant Identifiers

Many naming authorities exist that provide and manage business identifiers.

Ways to provide standardized URNs for various registered or unregistered naming authorities and person or organization name values are developed in **[**ebCorePartyId**]**:

"*Using the IETF/IANA registered NID format* `urn:oasis:names:tc:ebcore:partyid-type:` *a variety of ways to specify an organizational name are outlined. For example, the DUNS naming authority type can appear as*

`urn:oasis:names:tc:ebcore:partyid-type:DataUniversalNumberingSystem:0060`

`urn:oasis:names:tc:ebcore:partyid-type:iso9735:1`"

For each of these types, a DUNS value would appear as a suffix and identify a specific organization or business.

The IETF Best Current Practice [RFC3405] established top level domains (such as "URN.ARPA") within whose authority any registered NID may place NAPTR records that can be retrieved by queries using DNS

label paths. URNs are converted to DNS query paths by appending the top level domain “urn.arpa” to the registered NID (“oasis”). [See [URN] for URN syntax and the concepts of NID (Namespace ID) and NSS (Namespace Specific String).]

Then the URN’s NSS is prepended to obtain the full DNS query string:

myOrgIDValue:0060:iso6523:partyid-type:ebcore:tc:names.oasis.urn.arpa.

The registered NAPTRs for this string then provide URLs for metadata services when the metadata services have protected access, such as that enabled by HTTP basic authentication.

2.3.4 Protocol Specific Names and Addresses

The organization or person identifier formats for Email, IM, and SIP VOIP all make use of a format that links registered DNS names with a user name part (User@Domain).

When there are no concerns about publishing metadata service DNS records (which are intrinsically open and not access-control protected), the user name can be prefixed (as a label) to the domain name to form a DNS query string that may be used to retrieve U-NAPTRs for registration and metadata services. Some technical limitations on label length, and overall DNS query length exist. And in practice, some limitations in octets allowed in labels probably exist.

For new deployments, technical and practical implementations may be worked around by restricting the strings allowed for the new “User” part to be acceptable as DNS labels. For workarounds for existing non-conforming user names, conforming aliases may be created for the non-conforming names, and then used when metadata service RRs are created.

2.3.5 Registration Services

If a registration service is a prerequisite for access to the location of a metadata service, its URL can be published in the DNS using a U-NAPTR RR. This specification leaves the interaction with the registration service as implementation dependent.

In accordance with the ABNF in [RFC4848: 4.5 Service Parameters], the convention for a Registration service is to concatenate the app-service, a colon “:”, and the app-protocol (such as “CPPA” or “SMP”.)

IN NAPTR 100 10 "U" "Register:SMP" "!^.*$!https://example.com/register!"3

Other metadata services can of course be introduced with IANA registered or experimental values.

2.4 Location Discovery Flows

Given a DNS name formed from DNS labels separated by single dots, both metadata service location discovery, as well as associated registration service locations, can be returned as U-NAPTR records.

To support the use of U-NAPTR in obtaining service location URLs, an implementation of this specification MUST minimally provide an application programming interface that given a DNS name returns an array or similar collection of U-NAPTR resource records containing all defined fields.

More useful implementations SHOULD define and specify functionality for a broader range of inputs, and return types.

Some of these enhanced functions and utilities include:

- functions with inputs for selecting fields of the RRs returned for DNS label-path of the organization, person, or business document exchange network returning metadata or registration service locations.
- utility functions that provide transformations to create the DNS label that is prepended to the hosting domain path to form a distinctive DNS query path.

- utility functions constructing URNs for the participant identities within some naming authority in accordance with [ebCorePartyId] and found in the reserved urn.arpa domain.
- functions combining the above URN with a service hosting domain path to create a DNS label that is prepended to the hosting domain to form the DNS query path.
- functions with configurable filters of RRs (using service values, order, preference or other constraints on U-NAPTR RR fields).
- function requests to return particular formats of returned values or RRs such as XML, JSON, or others.

Applications or modules that make use of the DDDS-based discovery of metadata or registration services MUST provide either core support or enhanced support. Core support amounts to the ability to query for a specific organizational or personal domain and return results for all U-NAPTRs for that domain. Enhanced support will provide the capability to make queries and return results such as those enumerated above.

Specific APIs for core or enhanced support are not defined by this specification. An illustrative example for a web module is provided in appendix B.

It is worth noting that enhanced implementation APIs SHOULD allow for inputs of a sequence of strings together with concatenation and other common string operations. Also, various standardized hashing algorithms, such as sha-1, SHOULD be supported to yield short DNS labels allowed by the practical restrictions normally made for domain labels.

3 Security Considerations

DNS information is accessible over the public internet, and is not subject to any authorization restrictions.

While DNS is subject to some security threats [RFC3833], there are several available security enhancements available to establish both the origin of DNS RR data, and the integrity of RR data using DNSSEC security. [RFC4033,4034,4035]

DNS cache poisoning threats can be mitigated by following best practices for DNS data management [RFC5936,6781,6895]

4 Conformance

Both implementations and interaction communities may be in conformance with this specification.

Communities that use metadata CPPA or SMP conform to this specification simply by using one or more of the normative service field values in 2.2, namely, the values: Meta:SMP. Meta:CPPA, Register:SMP, Register:CPPA.

Other communities using a U-NAPTR for their metadata standards conform by naming their services using the patterns of section 2.2, such as Meta:[metadata standard identifier].

This specification also distinguishes two conformance levels for metadata service-location discovery implementations. An implementation here means an API defined in some programming language.

For either level of API conformance, it is assumed that the DNS record service behavior is fully defined by IETF DNS [RFC4848]. Essentially, U-NAPTR resource records MUST be supported on the server side.

Core Implementation Conformance: An implementation exhibits core conformance when it can be given a DNS-based personal or organizational identifier as a DNS label-path and return zero or more DNS RRs in a format containing at least the U-NAPTR RR fields [RFC4848]: order, pref, flags, service, regexp.

A public core-conformant implementation must expose interface and format descriptions and at least one invocation process that enables its use in at least one programming environment. Illustrations of core-conformant implementations are provided in Appendix A.

Enhanced Implementation Conformance: An implementation exhibits enhanced conformance when it is core-conformant and additionally is defined for a richer set of inputs, as enumerated in section 2.4 . For an implementation to exhibit enhanced conformance, it must also be a public implementation and define interface and format descriptions as well as one or more invocation procedures usable in one or more programming environments.

Appendix A Acknowledgments

The following individuals have participated in the creation of this specification and are gratefully acknowledged:

Participants:

Cruellas, Juan	Departamento de Arquitectura de Computadores, Univ Politecnica de Cataluna
Eijk, Pim van der	Sonnenglanz Consulting
Fieten, Sander	Individual
Forsberg, Martin	Swedish Association of Local Authorities & Regions
Lodi, Giorgia	Agency for Digital Italy
McGrath, Tim	Document Engineering Services Limited
Moberg, Dale	Axway Software
Pedersen, Klaus	Difi-Agency for Public Management and eGovernment
Raia, Alfio	Agency for Digital Italy
Rasmussen, Sven	Danish Agency for Digitisation, Ministry of Finance
Wigard, Susanne	Land Nordrhein-Westfalen

Appendix B Illustrative Core Conformance

1. Unix Shell cli core implementation:

```
#!/usr/bin/bash
if [ $# == 1 ]
then
dig -t NAPTR $1 | sed 's/^;.*//'  | grep $1  > rrs
awk '/IN/ {print $4, $5, $6, $7, $8, $9, $10}' < rrs
else
echo "Missing DNS path."
fi
```

2. Nodejs core interface description:

Input: meta1.metanet

Json output:

```
[ '{"name":"meta1.metanet","type":"NAPTR","flags":"u","order":10,"preference":100,"service":"meta:smp",
    "value":"http://www.meta1.metanet/smp"}',
  '{"name":"meta1.metanet","type": "NAPTR","flags":"u","order":10,"preference":100,"service":"meta:cppa",
    "value":"http://www.meta1.metanet/cppa"}' ]
```

Sample implementation of javascript invocation mechanisms for browsers via Ajax and for shell command lines are provided in a zipped package accompanying this specification.

Appendix C Revision History

Revision	Date	Editor	Changes Made
1	October 2012	Dale Moberg	Initial draft
2	December 2012	Dale Moberg	Conformance levels
3	July 2013	Dale Moberg	Security remarks from TC
4	August 2013	Dale Moberg	Sample core conformance interface illustrations
5	September 2013	Dale Moberg	https://lists.oasis-open.org/archives/bdxr/201308/msg00002.html Rework Peppol section 2.2.1 U-Naptr, adding regexp utilization example.
6	February 2014	Dale Moberg	The wd-06 version addresses the mainly edito-rial comments received by our TAB reviewer.
7	March 2014	Dale Moberg Pim van der Eijk	Resolve TAB comments with editorial changes.
8	March 2014	Pim van der Eijk	Bibliography, editorial changes.