Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 28, 2020

G. Lozano
ICANN

Nov 25, 2019

Registry Data Escrow Specification
draft-ietf-regext-data-escrow-02

Abstract

This document specifies the format and contents of data escrow deposits targeted primarily for domain name registries. However, the specification was designed to be independent of the underlying objects that are being escrowed, therefore it could be used for purposes other than domain name registries.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on May 28, 2020.

Copyright Notice

Table of Contents

1. Introduction

Registry Data Escrow is the process by which a registry periodically submits data deposits to a third-party called an escrow agent. These deposits comprise the minimum data needed by a third-party to resume operations if the registry cannot function and is unable or unwilling

to facilitate an orderly transfer of service. For example, for a domain name registry or registrar, the data to be deposited would include all the objects related to registered domain names, e.g., names, contacts, name servers, etc.

The goal of data escrow is higher resiliency of registration services, for the benefit of Internet users. The beneficiaries of a registry are not just those registering information there, but all relying parties that need to identify the owners of objects.

In the context of domain name registries, registration data escrow is a requirement for generic top-level domains and some country code top-level domain managers are also currently escrowing data. There is also a similar requirement for ICANN-accredited domain registrars.

This document specifies a format for data escrow deposits independent of the objects being escrowed. A specification is required for each type of registry/set of objects that is expected to be escrowed.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

Deposit. Deposits can be of three kinds: Full, Differential or Incremental. For all kinds of deposits, the universe of registry objects to be considered for data escrow are those objects necessary in order to offer the registry services.

Differential Deposit. Contains data that reflects all transactions involving the database that were not reflected in the last previous Full, Incremental or Differential Deposit, as the case may be. Differential Deposit files will contain information from all database

objects that were added, modified or deleted since the previous deposit was completed as of its defined Timeline Watermark.

Domain Name. See definition of Domain name in [RFC8499].

Escrow Agent. The organization designated by the registry or the third-party beneficiary to receive and guard data escrow deposits from the registry.

Full Deposit. Contains the registry data that reflects the current and complete registry database and will consist of data that reflects

the state of the registry as of a defined Timeline Watermark for the deposit.

Incremental Deposit. Contains data that reflects all transactions involving the database that were not reflected in the last previous Full Deposit. Incremental Deposit files will contain information from all database objects that were added, modified or deleted since the previous Full Deposit was completed as of its defined Timeline Watermark. If the Timeline Watermark of an Incremental Deposit were to cover the Timeline Watermark of another (Incremental or Differential) Deposit since the last Full Deposit, the more recent deposit MUST contain all the transactions of the earlier deposit.

Registrar. See definition of Registrar in [RFC8499].

Registry. See definition of Registry in [RFC8499].

Third-Party Beneficiary. Is the organization that, under extraordinary circumstances, would receive the escrow deposits the registry transferred to the escrow agent. This organization could be a backup registry, registry regulator, contracting party of the registry, etc.

Timeline Watermark. Point in time on which to base the collecting of database objects for a deposit. Deposits are expected to be consistent to that point in time.

Top-Level Domain. See definition of Top-Level Domain (TLD) in [RFC8499].

3. Problem Scope

In the past few years, the issue of registry continuity has been carefully considered in the gTLD and ccTLD space. Various organizations have carried out risk analyses and developed business continuity plans to deal with those risks, should they materialize.

One of the solutions considered and used, especially in the gTLD space, is Registry Data Escrow as a way to ensure the continuity of registry services in the extreme case of registry failure.

So far, almost every registry that uses Registry Data Escrow has its own specification. It is anticipated that more registries will be implementing escrow especially with an increasing number of domain registries coming into service, adding complexity to this issue.

It would seem beneficial to have a standardized specification for Registry Data Escrow that can be used by any registry to submit its deposits.

While the main motivation for developing this specification is rooted on the domain name industry, the specification has been designed to be as general as possible. This allows other types of registries to use this base specification and develop their own specifications covering the objects used by other registration organizations.

Specifications covering the objects used by registration organizations shall identify the format and contents of the deposits a registry has to make, such that a different registry would be able to rebuild the registration services of the former, without its help, in a timely manner, with minimum disruption to its users.

Since the details of the registration services provided vary from registry to registry, specifications covering the objects used by registration organizations shall provide mechanisms that allow its extensibility to accommodate variations and extensions of the registration services.

Given the requirement for confidentiality and the importance of accuracy of the information that is handled in order to offer registration services, parties using this specification shall define confidentiality and integrity mechanisms for handling the registration data.

Specifications covering the objects used by registration organizations shall not include in the specification transient objects that can be recreated by the new registry, particularly those of delicate confidentiality, e.g., DNSSEC KSK/ZSK private keys.

Details that are a matter of policy should be identified as such for the benefit of the implementers.

Non-technical issues concerning data escrow, such as whether to escrow data and under which purposes the data may be used, are outside of scope of this document.

4. General Conventions

The XML namespace prefix "rde" is used for the namespace "urn:ietf:params:xml:ns:rde-1.0", but implementations MUST NOT depend on it; instead, they should employ a proper namespace-aware XML parser and serializer to interpret and output the XML documents.

The XML namespace prefix "rdeObj1" and "rdeObj2" with the corresponding namespace "urn:ietf:params:xml:ns:rdeObj1-1.0" and "urn:ietf:params:xml:ns:rdeObj2-1.0" are used as example data escrow objects.

4.1. Date and Time

Numerous fields indicate "dates", such as the creation and expiry dates for objects. These fields SHALL contain timestamps indicating the date and time in UTC, specified in Internet Date/Time Format (see [RFC3339], Section 5.6) with the time-offset specified as "Z".

5. Protocol Description

The following is a format for data escrow deposits as produced by a

registry. The deposits are represented in XML. Only the format of the objects deposited is defined, nothing is prescribed about the method used to transfer such deposits between the registry and the escrow agent or vice versa.

The protocol intends to be object agnostic allowing the "overload" of abstract elements using the "substitutionGroup" attribute of the XML Schema element to define the actual elements of an object to be escrowed.

5.1. Root element <deposit>

The container or root element for a Registry Data Escrow deposit is <deposit>. This element contains the following child elements: <watermark>, <rdeMenu>, <deletes> and <contents> elements. This element also contains the following attributes:

- A REQUIRED "type" attribute that is used to identify the kind of deposit: FULL (Full), INCR (Incremental) or DIFF (Differential).
- A REQUIRED "id" attribute that is used to uniquely identify the escrow deposit. Each registry is responsible for maintaining its own escrow deposits identifier space to ensure uniqueness.
- An OPTIONAL "prevId" attribute that can be used to identify the previous Incremental, Differential or Full Deposit. This attribute MUST be used in Differential Deposits ("DIFF" type).
- An OPTIONAL "resend" attribute that is incremented each time the escrow deposit failed the verification procedure at the receiving party and a new escrow deposit needs to be generated by the registry for that specific date. The first time a deposit is generated the attribute is either omitted or MUST be "0". If a

deposit needs to be generated again, the attribute MUST be set to "1", and so on.

Example of a Full Deposit with the two example objects rdeObj1 and rdeObj2:

```
<?xml version="1.0" encoding="UTF-8"?>
<rde:deposit
```

```
    xmlns:rde="urn:ietf:params:xml:ns:rde-1.0"
    xmlns:rdeObj1="urn:ietf:params:xml:ns:rdeObj1-1.0"
    xmlns:rdeObj2="urn:ietf:params:xml:ns:rdeObj2-1.0"
    type="FULL"
    id="20191017001">
    <rde:watermark>2019-10-18T00:00:00Z</rde:watermark>
    <rde:rdeMenu>
      <rde:version>1.0</rde:version>
      <rde:objURI>urn:ietf:params:xml:ns:rdeObj1-1.0</rde:objURI>
      <rde:objURI>urn:ietf:params:xml:ns:rdeObj2-1.0</rde:objURI>
    </rde:rdeMenu>
    <rde:contents>
      <rdeObj1:rdeObj1>
        <rdeObj1:name>EXAMPLE</rdeObj1:name>
      </rdeObj1:rdeObj1>
      <rdeObj2:rdeObj2>
        <rdeObj2:id>fsh8013-EXAMPLE</rdeObj2:id>
      </rdeObj2:rdeObj2>
    </rde:contents>
  </rde:deposit>
```

Example of a Differential Deposit with the two example objects rdeObj1 and rdeObj2:


```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<rde:deposit
  xmlns:rde="urn:ietf:params:xml:ns:rde-1.0"
  xmlns:rdeObj1="urn:ietf:params:xml:ns:rdeObj1-1.0"
  xmlns:rdeObj2="urn:ietf:params:xml:ns:rdeObj2-1.0"
  type="DIFF"
  id="20191017001" prevId="20191016001">
  <rde:watermark>2019-10-18T00:00:00Z</rde:watermark>
  <rde:rdeMenu>
    <rde:version>1.0</rde:version>
      <rde:objURI>urn:ietf:params:xml:ns:rdeObj1-1.0</rde:objURI>
      <rde:objURI>urn:ietf:params:xml:ns:rdeObj2-1.0</rde:objURI>
  </rde:rdeMenu>
  <rde:deletes>
    <rdeObj1:delete>
      <rdeObj1:name>EXAMPLE1</rdeObj1:name>
    </rdeObj1:delete>
    <rdeObj2:delete>
      <rdeObj2:id>fsh8013-EXAMPLE</rdeObj2:id>
    </rdeObj2:delete>
  </rde:deletes>
  <rde:contents>
    <rdeObj1:rdeObj1>
      <rdeObj1:name>EXAMPLE2</rdeObj1:name>
    </rdeObj1:rdeObj1>
    <rdeObj2:rdeObj2>
      <rdeObj2:id>sh8014-EXAMPLE</rdeObj2:id>
    </rdeObj2:rdeObj2>
  </rde:contents>
</rde:deposit>
```

Example of an Incremental Deposit with the two example objects rdeObj1 and rdeObj2:

```
<?xml version="1.0" encoding="UTF-8"?>
<rde:deposit
  xmlns:rde="urn:ietf:params:xml:ns:rde-1.0"
  xmlns:rdeObj1="urn:ietf:params:xml:ns:rdeObj1-1.0"
  xmlns:rdeObj2="urn:ietf:params:xml:ns:rdeObj2-1.0"
  type="INCR"
  id="20191017001" prevId="20191010001">
  <rde:watermark>2019-10-18T00:00:00Z</rde:watermark>
  <rde:rdeMenu>
    <rde:version>1.0</rde:version>
    <rde:objURI>urn:ietf:params:xml:ns:rdeObj1-1.0</rde:objURI>
    <rde:objURI>urn:ietf:params:xml:ns:rdeObj2-1.0</rde:objURI>
  </rde:rdeMenu>
  <rde:deletes>
    <rdeObj1:delete>
      <rdeObj1:name>EXAMPLE1</rdeObj1:name>
    </rdeObj1:delete>
    <rdeObj2:delete>
      <rdeObj2:id>fsh8013-EXAMPLE</rdeObj2:id>
    </rdeObj2:delete>
  </rde:deletes>
  <rde:contents>
    <rdeObj1:rdeObj1>
      <rdeObj1:name>EXAMPLE2</rdeObj1:name>
    </rdeObj1:rdeObj1>
    <rdeObj2:rdeObj2>
      <rdeObj2:id>sh8014-EXAMPLE</rdeObj2:id>
    </rdeObj2:rdeObj2>
  </rde:contents>
</rde:deposit>
```

5.2. Child <watermark> element

A REQUIRED <watermark> element contains the data-time corresponding to the Timeline Watermark of the deposit.

5.3. Child <rdeMenu> element

This element contains auxiliary information of the data escrow deposit.

A REQUIRED <rdeMenu> element contains the following child elements:

- A REQUIRED <version> element that identifies the RDE protocol version.

- One or more <objURI> elements that contain namespace URIs representing the <contents> and <deletes> element objects.

5.4. Child <deletes> element

This element SHOULD be present in deposits of type Incremental or Differential. It contains the list of objects that were deleted since the base previous deposit. Each object in this section SHALL contain an ID for the object deleted.

This section of the deposit SHOULD NOT be present in Full Deposits. When rebuilding a registry it SHOULD be ignored if present in a Full Deposit.

The specification for each object to be escrowed MUST declare the identifier to be used to reference the object to be deleted.

5.5. Child <contents> element

This element of the deposit contains the objects in the deposit. It MUST be present in all type of deposits. It contains the data for the objects to be escrowed. The actual objects have to be specified individually.

In the case of Incremental or Differential Deposits, the objects indicate whether the object was added or modified after the base previous deposit. In order to distinguish between one and the other, it will be sufficient to check existence of the referenced object in the previous deposit.

When applying Incremental or Differential Deposits (when rebuilding the registry from data escrow deposits) the relative order of the <deletes> elements is important, as is the relative order of the <contents> elements. All the <deletes> elements MUST be applied first, in the order that they appear. All the <contents> elements MUST be applied next, in the order that they appear.

If an object is present in the <contents> section of several deposits (e.g. Full and Differential) the registry data from the latest deposit (as defined by the Timeline Watermark) SHOULD be used when rebuilding the registry.

6. Formal Syntax

6.1. RDE Schema


```
BEGIN
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="urn:ietf:params:xml:ns:rde-1.0"
```

```
    xmlns:rde="urn:ietf:params:xml:ns:rde-1.0"
    xmlns="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="qualified">

    <annotation>
      <documentation>
        Registry Data Escrow schema
      </documentation>
    </annotation>

    <!-- Root element -->
    <element name="deposit" type="rde:escrowDepositType"/>

    <!-- RDE types -->
    <complexType name="escrowDepositType">
      <sequence>
```



```
        <element name="watermark" type="dateTime"/>
        <element name="rdeMenu" type="rde:rdeMenuType"/>
        <element name="deletes" type="rde:deletesType" minOccurs="0"/>
        <element name="contents" type="rde:contentsType"/>
      </sequence>
      <attribute name="type" type="rde:depositTypeType" use="required"/>
      <attribute name="id" type="rde:depositIdType" use="required"/>
      <attribute name="prevId" type="rde:depositIdType"/>
      <attribute name="resend" type="unsignedShort" default="0"/>
    </complexType>

    <!-- Menu type -->
    <complexType name="rdeMenuType">
      <sequence>
        <element name="version" type="rde:versionType"/>
        <element name="objURI" type="anyURI" maxOccurs="unbounded"/>
      </sequence>
    </complexType>

    <!-- Deletes Type -->
    <complexType name="deletesType">
      <sequence minOccurs="0" maxOccurs="unbounded">
        <element ref="rde:delete"/>
      </sequence>
    </complexType>
```

```
    <element name="delete" type="rde:deleteType" abstract="true" />
    <complexType name="deleteType">
      <complexContent>
        <restriction base="anyType"/>
      </complexContent>
    </complexType>

    <!-- Contents Type -->
    <complexType name="contentsType">
      <sequence maxOccurs="unbounded">
        <element ref="rde:content"/>
      </sequence>
    </complexType>

    <element name="content" type="rde:contentType" abstract="true" />
    <complexType name="contentType">
      <complexContent>
        <restriction base="anyType"/>
      </complexContent>
    </complexType>

    <!-- Type of deposit -->
```



```
    <simpleType name="depositTypeType">
      <restriction base="token">
        <enumeration value="FULL"/>
        <enumeration value="INCR"/>
        <enumeration value="DIFF"/>
      </restriction>
    </simpleType>

    <!-- Deposit identifier type -->
    <simpleType name="depositIdType">
      <restriction base="token">
        <pattern value="\w{1,13}"/>
      </restriction>
    </simpleType>

    <!-- A RDE version number is a dotted pair of decimal numbers -->
    <simpleType name="versionType">
      <restriction base="token">
```

```
        <pattern value="[1-9]+\.[0-9]+"/>
        <enumeration value="1.0"/>
      </restriction>
    </simpleType>

    <!-- Auxiliary element to identify a registrar -->
    <simpleType name="clIDType">
      <restriction base="token">
        <minLength value="3"/>
        <maxLength value="16"/>
      </restriction>
    </simpleType>

    <complexType name="rrType">
      <simpleContent>
        <extension base="rde:clIDType">
          <attribute name="client" type="rde:clIDType"/>
        </extension>
      </simpleContent>
    </complexType>
  </schema>
  END
```

7. Internationalization Considerations

Data escrow deposits are represented in XML, which provides native support for encoding information using the Unicode character set and its more compact representations including UTF-8. Conformant XML processors recognize both UTF-8 and UTF-16. Though XML includes provisions to identify and use other character encodings through use

of an "encoding" attribute in an <?xml?> declaration, use of UTF-8 is RECOMMENDED.

8. IANA Considerations

This document uses URNs to describe XML namespaces and XML schemas conforming to a registry mechanism described in [RFC3688]. Two URI assignments have been registered by the IANA.

Registration request for the RDE namespace:

URI: urn:ietf:params:xml:ns:rde-1.0

Registrant Contact: See the "Author's Address" section of this document.

XML: None. Namespace URIs do not represent an XML specification.

Registration request for the RDE XML schema:

URI: urn:ietf:params:xml:schema:rde-1.0

Registrant Contact: See the "Author's Address" section of this document.

See the "Formal Syntax" section of this document.

9. Implementation Status

Note to RFC Editor: Please remove this section and the reference to RFC 7942 [RFC7942] before publication.

This section records the status of known implementations of the protocol defined by this specification at the time of posting of this Internet-Draft, and is based on a proposal described in RFC 7942 [RFC7942]. The description of implementations in this section is intended to assist the IETF in its decision processes in progressing drafts to RFCs. Please note that the listing of any individual implementation here does not imply endorsement by the IETF. Furthermore, no effort has been spent to verify the information presented here that was supplied by IETF contributors. This is not intended as, and must not be construed to be, a catalog of available implementations or their features. Readers are advised to note that other implementations may exist.

According to RFC 7942 [RFC7942], "this will allow reviewers and working groups to assign due consideration to documents that have the benefit of running code, which may serve as evidence of valuable

experimentation and feedback that have made the implemented protocols more mature. It is up to the individual working groups to use this information as they see fit".

9.1. Implementation in the gTLD space

Organization: ICANN

Name: ICANN Registry Agreement

Description: the ICANN Base Registry Agreement requires Registries, Data Escrow Agents, and ICANN to implement this specification. ICANN receives daily notifications from Data Escrow Agents confirming that more than 1,200 gTLDs are sending deposits that comply with this specification. ICANN receives on a weekly basis per gTLD, from more than 1,200 gTLD registries, a Bulk Registration Data Access file that also complies with this specification. In addition, ICANN is aware of Registry Service Provider transitions using data files that conform to this specification.

Level of maturity: production.

Coverage: all aspects of this specification are implemented.

Version compatibility: versions 03 - 08 are known to be implemented.

Contact: gustavo.lozano@icann.org

URL: https://www.icann.org/resources/pages/registries/registries-agreements-en

10. Security Considerations

This specification does not define the security mechanisms to be used in the transmission of the data escrow deposits, since it only specifies the minimum necessary to enable the rebuilding of a registry from deposits without intervention from the original registry.

Depending on local policies, some elements or most likely, the whole deposit will be considered confidential. As such the registry transmitting the data to the escrow agent SHOULD take all the necessary precautions like encrypting the data itself and/or the transport channel to avoid inadvertent disclosure of private data.

It is also of the utmost importance the authentication of the parties passing data escrow deposit files. The escrow agent SHOULD properly authenticate the identity of the registry before accepting data

escrow deposits. In a similar manner, the registry SHOULD authenticate the identity of the escrow agent before submitting any data.

Additionally, the registry and the escrow agent SHOULD use integrity checking mechanisms to ensure the data transmitted is what the source intended. Validation of the contents by the escrow agent is RECOMMENDED to ensure not only the file was transmitted correctly from the registry, but also the contents are also "meaningful".

11. Privacy Considerations

This specification defines a format that may be used to escrow personal data. The process of data escrow is governed by a legal document agreed by the parties, and such legal document must regulate the particularities regarding the protection of personal data.

12. Acknowledgments

Special suggestions that have been incorporated into this document were provided by James Gould, Edward Lewis, Jaap Akkerhuis, Lawrence Conroy, Marc Groeneweg, Michael Young, Chris Wright, Patrick Mevzek, Stephen Morris, Scott Hollenbeck, Stephane Bortzmeyer, Warren Kumari, Paul Hoffman, Vika Mpisane, Bernie Hoeneisen, Jim Galvin, Andrew Sullivan, Hiro Hotta, Christopher Browne, Daniel Kalchev, David Conrad, James Mitchell, Francisco Obispo, Bhadresh Modi and Alexander Mayrhofer.

Shoji Noguchi and Francisco Arias participated as co-authors until version 07 providing invaluable support for this document.

13. Change History

[[RFC Editor: Please remove this section.]]

13.1. Changes from 00 to 01

1. Included DNSSEC elements as part of the basic <domain> element as defined in RFC 5910.

2. Included RGP elements as part of the basic <domain> element as defined in RFC 3915.

3. Added support for IDNs and IDN variants.

4. Eliminated the <summary> element and all its subordinate objects, except <watermarkDate>.

5. Renamed <watermarkDate> to <watermark> and included it directly under root element.

6. Renamed root element to <deposit>.

7. Added <authinfo> element under <registrar> element.

8. Added <roid> element under <registrar> element.

9. Reversed the order of the <deletes> and <contents> elements.

10. Removed <rdeDomain:status> minOccurs="0".

11. Added <extension> element under root element.

12. Added <extension> element under <contact> element.

13. Removed <period> element from <domain> element.

14. Populated the "Security Considerations" section.

15. Populated the "Internationalization Considerations" section.

16. Populated the "Extension Example" section.

17. Added <deDate> element under <domain> element.

18. Added <icannID> element under <registrar> element.

19. Added <eppParams> element under root element.

20. Fixed some typographical errors and omissions.

13.2. Changes from 01 to 02

1. Added definition for "canonical" in the "IDN variants Handling" section.

2. Clarified that "blocked" and "reserved" IDN variants are optional.

3. Made <rdeRegistrar:authInfo> optional.

4. Introduced substitutionGroup as the mechanism for extending the protocol.

5. Moved <eppParams> element to be child of <contents>

6. Text improvements in the Introduction, Terminology, and Problem Scope per Jay's suggestion.

7. Removed <trDate> from <rdeDomain> and added <trnData> instead, which include all the data from the last (pending/processed) transfer request

8. Removed <trDate> from <rdeContact> and added <trnData> instead, which include all the data from the last (pending/processed) transfer request

9. Fixed some typographical errors and omissions.

13.3. Changes from 02 to 03

1. Separated domain name objects from protocol.

2. Moved <extension> elements to be child of <deletes> and <contents>, additionally removed <extension> element from <rdeDomain>,<rdeHost>, <rdeContact>,<rdeRegistrar> and <rdeIDN> elements.

3. Modified the definition of <rde:id> and <rde:prevId>.

4. Added <rdeMenu> element under <deposit> element.

5. Fixed some typographical errors and omissions.

13.4. Changes from 03 to 04

1. Removed <eppParams> objects.

2. Populated the "Extension Guidelines" section.

3. Fixed some typographical errors and omissions.

13.5. Changes from 04 to 05

1. Fixes to the XSD
2. Extension Guidelines moved to dnrd-mappings draft
3. Fixed some typographical errors and omissions.

13.6. Changes from 05 to 06

1. Fix resend definition.

13.7. Changes from 06 to 07

1. Editorial updates.
2. schemaLocation removed from RDE Schema.

13.8. Changes from 07 to 08

1. Ping update

13.9. Changes from 08 to 09

1. Ping update.

13.10. Changes from 09 to 10

1. Implementation Status section was added

13.11. Changes from 10 to 11

1. Ping update.

13.12. Changes from 11 to REGEXT 00

1. Internet Draft (I-D) adopted by the REGEXT WG.

13.13. Changes from version REGEXT 00 to REGEXT 01

1. Privacy consideration section was added

13.14. Changes from version REGEXT 01 to REGEXT 02

1. Updated the Security Considerations section to make the language normative

2. Updated the rde XML schema to remove the dependency with the eppcom namespace reference

3. Editorial updates

4. Remove the reference to RFC 5730

5. Added complete examples of deposits

14. References

14.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC3339] Klyne, G. and C. Newman, "Date and Time on the Internet: Timestamps", RFC 3339, DOI 10.17487/RFC3339, July 2002, <https://www.rfc-editor.org/info/rfc3339>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

14.2. Informative References

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <https://www.rfc-editor.org/info/rfc3688>.

[RFC7942] Sheffer, Y. and A. Farrel, "Improving Awareness of Running Code: The Implementation Status Section", BCP 205, RFC 7942, DOI 10.17487/RFC7942, July 2016, <https://www.rfc-editor.org/info/rfc7942>.

[RFC8499] Hoffman, P., Sullivan, A., and K. Fujiwara, "DNS Terminology", BCP 219, RFC 8499, DOI 10.17487/RFC8499, January 2019, <https://www.rfc-editor.org/info/rfc8499>.

Author's Address

Gustavo Lozano
Internet Corporation for Assigned Names and Numbers
12025 Waterfront Drive, Suite 300
Los Angeles 90292
United States of America

Phone: +1.310.823.9358
Email: gustavo.lozano@icann.org