
Stream: Internet Engineering Task Force (IETF)
RFC: [8909](#)
Category: Standards Track
Published: November 2020
ISSN: 2070-1721
Author: G. Lozano
ICANN

RFC 8909

Registry Data Escrow Specification

Abstract

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

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at <https://www.rfc-editor.org/info/rfc8909>.

Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction
2. Terminology
3. Problem Scope
4. Conventions Used in This Document
 - 4.1. Date and Time
5. Protocol Description
 - 5.1. Root Element <deposit>
 - 5.2. Rebuilding the Registry from Data Escrow Deposits
6. Formal Syntax
 - 6.1. RDE Schema
7. Internationalization Considerations
8. IANA Considerations
9. Security Considerations
10. Privacy Considerations
11. Example of a Full Deposit
12. Example of a Differential Deposit
13. Example of an Incremental Deposit
14. References
 - 14.1. Normative References
 - 14.2. Informative References

[Acknowledgments](#)

[Author's Address](#)

1. Introduction

Registry Data Escrow (RDE) 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 of the objects related to registered domain names, e.g., names, contacts, name servers.

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 also the users of services relying on the registry data.

In the context of domain name registries, registration data escrow is a requirement for generic Top-Level Domains (gTLDs) (e.g., Specification 2 of the ICANN Base Registry Agreement; see [\[ICANN-GTLD-RA-20170731\]](#)), and some country code TLD (ccTLD) 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. An independent specification is required for each type of registry/set of objects that is expected to be escrowed.

The format for data escrow deposits is specified using version 1.0 of the Extensible Markup Language (XML) as described in [\[W3C.REC-xml-20081126\]](#), and XML Schema notation as described in [\[W3C.REC-xmlschema-1-20041028\]](#) and [\[W3C.REC-xmlschema-2-20041028\]](#).

Readers are advised to read [Section 2](#) ("Terminology") carefully to understand the precise meanings of Differential and Incremental Deposits, as the definitions used in this document are different from the definitions typically used in the domain of data backups.

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: There are three kinds of deposits: Full, Differential, and Incremental. For all three kinds of deposits, the universe of registry objects to be considered for data escrow is comprised of any objects required to offer the registry services.

Differential Deposit: A 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 the definition of "domain name" in [\[RFC8499\]](#).

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

Full Deposit: A 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: An 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 (i.e., one or more Incremental or Differential Deposits exist for the period between the Timeline Watermark of a Full Deposit and an Incremental or Differential Deposit), the more recent deposit **MUST** contain all of the transactions of the earlier deposit.

Registrar: See the definition of "registrar" in [\[RFC8499\]](#).

Registry: See the definition of "registry" in [\[RFC8499\]](#).

Third-Party Beneficiary: A 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: The Timeline Watermark is the point in time on which to base the collecting of database objects for a deposit. Deposits are expected to be consistent with that point in time.

Top-Level Domain (TLD): See the definition of "Top-Level Domain" in [\[RFC8499\]](#).

3. Problem Scope

In the past few years, the issue of registry continuity has been carefully considered in the gTLD and ccTLD spaces. 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 domain name industry has been the main target for this specification, it has been designed to be as general as possible.

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 and 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 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 (Key Signing Key / Zone Signing Key) 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 for what purposes the data may be used, are outside the scope of this document.

Parties using this specification shall use a signaling mechanism to control the transmission, reception, and validation of data escrow deposits. The definition of such a signaling mechanism is outside the scope of this document.

4. Conventions Used in This Document

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 prefixes "rdeObj1" and "rdeObj2", with the corresponding namespaces "urn:example:params:xml:ns:rdeObj1-1.0" and "urn:example: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 parameter specified as "Z".

5. Protocol Description

The format for data escrow deposits as produced by a registry is defined below. The deposits are represented in XML ([Section 6](#)). Only the format of the objects deposited is defined. This document does not prescribe 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 [[W3C.REC-xmlschema-1-20041028](#)] of the XML Schema element to define the actual elements of an object to be escrowed.

The specification for each object to be escrowed **MUST** declare the identifier to be used to reference the object to be deleted or added/modified.

5.1. Root Element <deposit>

The container or root element for a Registry Data Escrow deposit is <deposit>.

The <deposit> element contains the following attributes:

- A **REQUIRED** "type" attribute that is used to identify the kind of deposit:
 - FULL: Full.
 - INCR: Incremental.
 - 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.
- A "prevId" attribute that can be used to identify the previous Incremental, Differential, or Full Deposit. This attribute is **REQUIRED** in Differential Deposits ("DIFF" type), is **OPTIONAL** in Incremental Deposits ("INCR" type), and is not used in Full Deposits ("FULL" 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 either (1) is omitted or (2) **MUST** be "0". If a deposit needs to be generated again, the attribute **MUST** be set to "1", and so on.

The <deposit> element contains the following child elements:

5.1.1. Child <watermark> Element

A **REQUIRED** <watermark> element contains the date-time [[RFC3339](#)] corresponding to the Timeline Watermark of the deposit.

5.1.2. Child <rdeMenu> Element

This element contains auxiliary information regarding the data escrow deposit.

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

- A **REQUIRED** <version> element that identifies the RDE protocol version. This value **MUST** be 1.0.
- One or more <objURI> elements that contain namespace URIs representing the <contents> and <deletes> element objects.

5.1.3. Child <deletes> Element

For Differential Deposits, this element contains the list of objects that have been deleted since the previous deposit of any type. For Incremental Deposits, this element contains the list of objects that have been deleted since the previous Full Deposit.

This section of the deposit **MUST NOT** be present in Full Deposits.

5.1.4. Child <contents> Element

For Full Deposits, this element contains all objects. For Differential Deposits, this element contains the list of objects that have been added or modified since the previous deposit of any type. For Incremental Deposits, this element contains the list of objects that have been added or modified since the previous Full Deposit.

5.2. Rebuilding the Registry from Data Escrow Deposits

When applying Incremental or Differential Deposits (when rebuilding the registry from data escrow deposits), the relative order of the <deletes> and <contents> elements is important because dependencies may exist between the objects. All of the <deletes> elements **MUST** be applied first, in the order in which they appear. All of the <contents> elements **MUST** be applied next, in the order in which they appear.

If an object is present in the <contents> or <deletes> 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. An object **SHOULD NOT** exist multiple times in either the <contents> or <deletes> elements in a single deposit.

When rebuilding a registry, the <deletes> section **MUST** be ignored if present in a Full Deposit.

6. Formal Syntax

RDE is specified in XML Schema notation. The formal syntax presented here is a complete schema representation of RDE suitable for automated validation of RDE XML instances.

The <CODE BEGINS> and <CODE ENDS> tags are not part of the schema; they are used to note the beginning and ending of the schema for URI registration purposes.

6.1. RDE Schema


```
<CODE BEGINS>
<?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"
        minOccurs="0"/>
    </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 minOccurs="0" 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>

  </schema>
<CODE ENDS>
```

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 the use of an "encoding" attribute in an `<?xml?>` declaration, the 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 for the RDE namespace:

URI: urn:ietf:params:xml:ns:rde-1.0
Registrant Contact: IESG
XML: None. Namespace URIs do not represent an XML specification.

Registration for the RDE XML schema:

URI: urn:ietf:params:xml:schema:rde-1.0
Registrant Contact: IESG

See [Section 6](#) ("Formal Syntax") of this document.

9. 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 parties **SHOULD** take all necessary precautions, such as encrypting the data at rest and in transit to avoid inadvertent disclosure of private data. Regardless of the precautions taken by the parties regarding data at rest and in transit, authentication credentials **MUST NOT** be escrowed.

Authentication of the parties passing data escrow deposit files is also of the utmost importance. The escrow agent **MUST** properly authenticate the identity of the registry before accepting data escrow deposits. Similarly, the registry **MUST** authenticate the identity of the escrow agent before submitting any data.

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

Note: If Transport Layer Security (TLS) is used when providing an escrow service, the recommendations in [\[RFC7525\]](#) **MUST** be implemented.

10. 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 upon by the parties, and such a legal document must ensure that privacy-sensitive and/or personal data receives the required protection.

11. Example of a Full Deposit

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:example:params:xml:ns:rdeObj1-1.0"
  xmlns:rdeObj2="urn:example:params:xml:ns:rdeObj2-1.0"
  type="FULL"
  id="20191018001">
  <rde:watermark>2019-10-17T23:59:59Z</rde:watermark>
  <rde:rdeMenu>
    <rde:version>1.0</rde:version>
    <rde:objURI>urn:example:params:xml:ns:rdeObj1-1.0</rde:objURI>
    <rde:objURI>urn:example: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>
```

12. Example of a Differential 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:example:params:xml:ns:rdeObj1-1.0"
  xmlns:rdeObj2="urn:example:params:xml:ns:rdeObj2-1.0"
  type="DIFF"
  id="20191019001" prevId="20191018001">
  <rde:watermark>2019-10-18T23:59:59Z</rde:watermark>
  <rde:rdeMenu>
    <rde:version>1.0</rde:version>
    <rde:objURI>urn:example:params:xml:ns:rdeObj1-1.0</rde:objURI>
    <rde:objURI>urn:example:params:xml:ns:rdeObj2-1.0</rde:objURI>
  </rde:rdeMenu>
  <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>
```

13. Example of an Incremental 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:example:params:xml:ns:rdeObj1-1.0"
  xmlns:rdeObj2="urn:example:params:xml:ns:rdeObj2-1.0"
  type="INCR"
  id="20200317001" prevId="20200314001">
  <rde:watermark>2020-03-16T23:59:59Z</rde:watermark>
  <rde:rdeMenu>
    <rde:version>1.0</rde:version>
    <rde:objURI>urn:example:params:xml:ns:rdeObj1-1.0</rde:objURI>
    <rde:objURI>urn:example: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>
```

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>>.

- [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>>.
- [W3C.REC-xml-20081126] Bray, T., Ed., Paoli, J., Ed., Sperberg-McQueen, C.M., Ed., Maler, E., Ed., and F. Yergeau, Ed., "Extensible Markup Language (XML) 1.0 (Fifth Edition)", REC-xml-20081126, November 2008, <<https://www.w3.org/TR/2008/REC-xml-20081126/>>.
- [W3C.REC-xmlschema-1-20041028] Thompson, H.S., Ed., Beech, D., Ed., Maloney, M., Ed., and N. Mendelsohn, Ed., "XML Schema Part 1: Structures Second Edition", REC-xmlschema-1-20041028, October 2004, <<https://www.w3.org/TR/2004/REC-xmlschema-1-20041028/>>.
- [W3C.REC-xmlschema-2-20041028] Biron, P. V., Ed. and A. Malhotra, Ed., "XML Schema Part 2: Datatypes Second Edition", REC-xmlschema-2-20041028, October 2004, <<https://www.w3.org/TR/2004/REC-xmlschema-2-20041028/>>.

14.2. Informative References

- [ICANN-GTLD-RA-20170731] ICANN, "Base Registry Agreement", 31 July 2017, <<https://newgtlds.icann.org/sites/default/files/agreements/agreement-approved-31jul17-en.pdf>>.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.
- [RFC7525] Sheffer, Y., Holz, R., and P. Saint-Andre, "Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)", BCP 195, RFC 7525, DOI 10.17487/RFC7525, May 2015, <<https://www.rfc-editor.org/info/rfc7525>>.

Acknowledgments

Special suggestions that were 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, Bhadrash Modi, and Alexander Mayrhofer.

Shoji Noguchi and Francisco Arias participated as coauthors through version 07 of draft-arias-noguchi-registry-data-escrow (the precursor to this document) and provided invaluable support for this document.

Author's Address

Gustavo Lozano

Internet Corporation for Assigned Names and Numbers

12025 Waterfront Drive, Suite 300

Los Angeles, CA 90292

United States of America

Phone: [+1.310.823.9358](tel:+13108239358)

Email: gustavo.lozano@icann.org