

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 17, 2013

J. Peterson
NeuStar, Inc.
July 16, 2012

**A Framework and Data Model for Queries about Telephone-Related Queries
(TeRQ)
draft-peterson-terq-01**

Abstract

As telephone services migrate to the Internet, Internet applications require access to diverse information about telephone numbers. ENUM, for example, applied the DNS to the problem of finding URIs for telephone services on the Internet. The intrinsic limitations in the query/response semantics of the DNS, however, have often been strained by the requirements for accessing information about telephone numbers. This document therefore proposes a protocol-independent framework and data model for querying and responding to requests concerning telephone numbers and call routing that allows a richer expression of both questions and answers.

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 <http://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 January 17, 2013.

Copyright Notice

Copyright (c) 2012 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 (<http://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.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Table of Contents

1.	Terminology	4
2.	Motivation	5
3.	Overview of the Framework	7
4.	Transport Independence	8
5.	The Data Model	9
5.1.	Source	9
5.1.1.	Query Source	9
5.1.2.	Query Intermediary	9
5.1.3.	Route Source	10
5.2.	Subject	10
5.2.1.	Telephone Number	10
5.2.2.	Service Provider Identifier	11
5.3.	Attributes	11
5.3.1.	Routing Attributes	11
5.3.2.	Administrative Attributes	12
5.4.	Records	12
5.4.1.	Attributes	12
5.4.2.	Authority	12
5.4.3.	Priority	12
5.4.4.	Expiration	12
5.5.	Response Code	12
6.	Bindings	14
7.	Encodings	16
8.	Profiles	17
9.	Security Considerations	18
10.	IANA Considerations	19
11.	Acknowledgements	20
12.	Informative References	21
	Author's Address	22

1. Terminology

In this document, the key words "MAY", "MUST", "MUST NOT", "SHOULD", and "SHOULD NOT", are to be interpreted as described in [[RFC2119](#)].

2. Motivation

Telephone numbers remain the worldwide standard identifier for routing calls and text messages over the Public Switched Telephone Network (PSTN). As identifiers, however, telephone numbers differ fundamentally from the identifiers commonly used by Internet applications. Email, the web and native Voice over IP (VoIP) systems typically use identifiers that rely on the Domain Name System (DNS) to resolve a domain portion of the identifier to a particular IP address; commonly, Uniform Resource Indicators (URIs) with a user and host component serve this purpose. In order to bridge this gap between the PSTN and the Internet, the ENUM effort specified a DNS profile for translating telephone numbers into URIs.

While the ENUM approach suffices for simple number translations, more complex routing and administrative functions can strain the capabilities of the DNS. Many of these problems result from the limiting simplicity of the DNS query string. DNS queries have a fairly rigid syntax oriented towards the resolution of an atomic name in a hierarchical namespace. Telephone call routing, however, may require compound queries that operate on several distinct query elements that are difficult to cast hierarchically. Many of the complex query/response mechanisms used in the PSTN are not tied directly to call routing or establishment, such as finding the caller's name (CNAM) when a call is received. Moreover, the centralized and authoritative hierarchy of the DNS proved a poor match for the actual procedures used to route telephone calls. This led to work on "infrastructure" ENUM, which assumed private DNS implementations, each of which could give a different answer to the same request to translate a telephone number depending on who asked, or other internal factors. The framework of the SPEERMINT working group, expanding on these requirements, differentiated the mapping of a telephone number to a target network (the "Look-up Function") from the mapping made by the originating network to the proper next-hop to reach such a target network (the "Location Routing Function"). While the LUF can be centralized and authoritative, the LRF is necessarily subjective and localized. In the SPEERMINT model, the routing of a call may involve an intermediate lookup that operates on a Service Provider Identifier (SPID) rather than a telephone number. Mapping these capabilities to ENUM requires security and administrative practices that further complicate its DNS implementation. The underlying architectural issues that give rise to all these problems are detailed in [draft-iab-dns-applications](#).

Despite these difficulties, the need for solutions in this space is pressing, as many carriers worldwide contemplate migrating their entire PSTN infrastructure onto the Internet within the next decade. Further pressures come from emerging Internet communications

providers who never invested in PSTN infrastructure in the first place, but want access to services related to telephone numbers. These different communities have diverse requirements. In some environments, there are performance constraints that would require a very lightweight binary protocol; in others, applications might prefer human-readable markup languages suitable for interfacing with existing APIs.

Therefore, this document proposes a reconsideration of telephone routing and administration services on the Internet based on a framework that considers queries and responses in an abstract architecture. This document specifies no particular syntax or encoding for queries or responses, but instead describes an extensible data model for the semantics of queries and responses that future specifications might encode in accordance with application needs.

3. Overview of the Framework

This framework specifies an abstract query/response protocol that enables a Client to send Queries to a Service about telephone numbers or related telephone services. Queries may pass through one or more Intermediaries on their way from a Client to a Service; for example, through aggregators or service bureaus. A Client establishes the Subject of a Query, and optionally specifies one or more Attributes of particular interest in order to narrow the desired response. When a Service receives a Query, it performs any necessary authorization and policy decisions based on the Source. If policy permits, the Service generates a Response, which will consist of a Response Code and one or more Records associated with the Subject. The Service then sends the Response through the same path that the Query followed; transactional identifiers set by the Client and Service correlate the Query to the Response and assist any intermediary routing.

4. Transport Independence

The data model provided for Queries and Responses in this framework is independent of any underlying transport or encoding. Future specifications will define Bindings that specify particular transports and Encodings for Queries and Responses. In some deployment environments, for example, a binary encoding and lightweight transport might be more appropriate than the use of a web protocol. This specification provides a template of requirements that must be addressed by any encoding scheme.

It is a design goal of this work that the semantics of Queries and Responses survive interworking through translations from one encoding to another; for example, when an Intermediary receives a binary query from a Client, it should be able to transcode it to an XML format to send to a Service without discarding any of the original semantics.

[TBD]

5. The Data Model

Every query has a Source and a Subject, and may have one or more Attributes. Every Response has a Response Code, one or more Records (containing Attributes), and may have a Subject (if the Subject differs from that of the Query).

5.1. Source

The Source is a required element in Queries. In this specification, three categories of Sources are defined: Query Source, Query Intermediary, and Route Source. At least one of these Sources must be present in a Query, and multiple Sources are permitted. Responses do not contain a Source.

Future specifications may extend the set of Source types.

5.1.1. Query Source

Every Query generated by a client has a Query Source, which identifies the originator of the Query. This represents the logical identity of the user or service provider who first sent the Query, rather than the identity of any Intermediate entity. This field is provided in the Source to authenticate the poser of the Query, so that the Service can make any necessary authorization decisions as it formulates a Response.

In some service deployments, an Intermediary may wish to mask the Query Source from a Service. The removal of the Query Source is permitted by TeRQ, but any Intermediary that removes the Query Source must provide a Query Intermediary for the Source element.

A Query Source element has a Type, which indicates how the logical identity of the originator of the Query has been represented. The Type field of the Query Source is extensible. Initial values include a domain name, a URI and a telephone number.

The Type element of the Query Source is followed by a Value, which contains the identity. The format of the identity is determined by the Type.

5.1.2. Query Intermediary

Optionally, Queries may contain one or more Query Intermediary elements in the Source. A Query Intermediary resides between the originator of the Query (the Client) and the Service, where it may aggregate queries, proxy them, transcode them, or provide any related relay function to assist the delivery of Queries to the Service.

The Query Intermediary element, like the Query Source, contains the logical identity of the service that relayed the Query. This field is provided in the Source for those deployments in which the Service makes an authorization decision based on the identity of the Intermediary rather than a Query Source.

A Query Intermediary element has a Type, which indicates how the logical identity of the Intermediary has been represented. The Type element of the Query Intermediary is extensible. Initial values include a domain name or a URI.

The Type of the Query Intermediary element is followed by a Value, which contains the identity. The format of the identity is determined by the Type.

5.1.3. Route Source

Optionally, Queries may contain a Route Source which identifies a reference point in the network from which any Routing Attributes in the response should be calculated. It therefore always designates a network element, though depending on the circumstances, it may be an endpoint, a gateway, a border device, or any other agent that makes forwarding decisions for telephone calls and related services.

A Route Source element has a Type, which indicates how the network element has been represented. The Type field of the Query Source is extensible. Initial values include a domain name, an IP address or a trunk group.

The Type of the Route Source element is followed by a Value, which designates the network element. The format of the identity is determined by the Type.

5.2. Subject

All Queries contain a Subject. The Subject contains the resource for which the originator of the Query is asking the Service to return Attributes. Responses only contain a Subject if the Subject of the Response differs from that of the original Query, which may occur when (for example) the Subject contains a broad range, and the Service replies with a more narrow Subject. Future specifications may define alternative Subject elements.

5.2.1. Telephone Number

The Telephone Number element of the Subject contains an encoding of a telephone number or a telephone number fragment.

A Telephone Number has a Type which designates which sort of telephone number the element contains. Types defined by this specification include: E.164, Service Code, Short Code, Prefix, Nationally-Specific and Unknown.

The Type of the Telephone Number element is followed by a Value, which contains the telephone number itself. The format of the identity is determined by the Type.

5.2.2. Service Provider Identifier

A Service Provider Identifier (SPID) may also be the Subject of the Query, if, for example, in a SPEERMINT-like architecture an initial resolution has already translated a telephone number into a SPID, and now the client wishes to find routes or other information related to the SPID.

A Service Provider Identifier has a Type which designates the sort of SPID the element contains. SPID types defined by this specification include: SPID.

5.3. Attributes

Attributes in this data model are all specified as having a Name and a Value. Individual attributes may contain their own subtyping mechanisms as required.

Queries optionally contain Attributes; a Query with no specified Attributes requests that the Service return any Attributes associated with the Subject. In a Query, the presence of one or more Attributes limits the scope of the Query to Records about the Subject containing those Attributes.

Responses contain Attributes within the one or more Record elements. At least one Record element will always be present in a successful Response, and thus at least one Attribute will be as well.

Attributes are broadly divided between Routing Attributes and Administrative Attributes. Routing Attributes provide information required to route communications, including URIs.

5.3.1. Routing Attributes

Routing Attributes defined by this document include: voip, sms [[TBD](#)]

5.3.2. Administrative Attributes

Administrative Attributes defined by this document include: CNAM, SPID, dialplan [[TBD](#)]

5.4. Records

The Record element appears only in Responses. It exists primarily as a means to deliver Attributes in answer to Queries, grouping together Attributes with an Authority and any expiry and preferential data recommended by the Service.

5.4.1. Attributes

A Record contains an Attribute, which may be either a Routing or Administrative Attribute.

5.4.2. Authority

The Authority subelement of a Record specifies the source of the data: either the entity that provisioned the data with the Service or the external source from which the Service collected the data. Like the "Query Source" element, the Authority element ideally gives a logical identity of the source of the data. The format has a Type followed by a Value, where the format of Values is defined by the Type. Types defined by this document include: domain name and IP address.

5.4.3. Priority

Optionally, a Service may specify a weighted Priority associated with a Record. Priorities are between 0 and 1, with a value of 1 having the highest priority.

5.4.4. Expiration

Optionally, a Service may specify an absolute time at which a Record will no longer be valid, should a client or intermediary wish to cache a Record. In the absence of an Expiration element, Records may be cached for a maximum of twenty-four hours.

5.5. Response Code

All Responses contain a Response Code.

Response Codes defined by this document include: Success, Subject Does Not Exist, No Suitable Records Exist for Subject, Subject Syntax Error, Unknown Attribute, Unauthorized Source, Route Source Topology

Unavailable.

[TBD]

6. Bindings

A TeRQ Binding is an underlying protocol that carries TeRQ Queries and Responses. Future specifications may define Bindings in accordance with the procedures in the IANA Considerations sections of this document.

By underlying protocol, this specification means both transport-layer protocols as well as any application-layer protocols that the Binding requires. Thus an example Binding might specify a combination of TCP, TLS, HTTP and SOAP as the underlying transport for TeRQ. Alternatively, it might only specify a very lightweight underlying protocol like UDP. A Binding may be specific to a particular Encoding, or it may be independent of any Encoding.

Bindings must specify whether they are continuous, transactional or non-transactional. A continuous Binding creates a persistent connection between two TeRQ entities over which many, potentially unrelated, Queries and Responses might flow. Many Bindings defined for use between an Intermediary and a Service will have this property, as Intermediaries may aggregate on behalf of many Clients, and opening a separate transport-layer connection for each new Query would be inefficient. A transactional Binding creates a temporary connection between two TeRQ entities for the purpose of fulfilling a single Query; any Responses to the Query will use the same connection to return to the sender of the Query. Finally, a non-transactional Binding does not rely on any sort of connection semantics: the senders of Queries and Responses will always initiate a new instance of the Binding to send a message.

This document makes no provision for discovering the Bindings supported by a TeRQ Client, Intermediary or Service. Intermediaries may transcode between Bindings if necessary when acting to connect a Client and a Service, especially if the Client and Service support no Bindings in common.

A Binding specification must enumerate all categories of metadata required to establish a connection using a Binding. For some Bindings, this might comprise solely an IP address and a port; for other Bindings, this might instead require higher-layer application identifiers like a URI. This metadata includes any identifiers necessary for correlating Queries to Responses in a continuous or non-transactional Binding; any Encoding making use of these Bindings must specify how it carries those elements.

Bindings must also describe the security services they make available. If a Binding incorporates TLS, for example, the host authentication that TLS can provide should be described in the

Binding specification, so that Encodings can potentially make use of this service to provide some of the semantics of TeRQ.

7. Encodings

A TeRQ Encoding specifies how the Query and Reponse are constructed syntactically. An Encoding may be specific to a particular Binding, or it may be specified independently of any Binding.

An Encoding may define an object format; for example, an XML or JSON object, described with any appropriate schemas, or an ABNF description. An Encoding might alternatively specify a mapping of the semantic elements of Queries and Responses on to the existing fields of headers of a protocol, especially when that protocol has been defined as an underlying protocol Binding.

Every Encoding must specify how each semantic element of a Query and Response will be represented. For all baseline TeRQ Attributes and elements, the Encoding specifies whether values will be text or binary, how they will be encoded. Many baseline element types (such as telephone numbers) can appear in different places in a TeRQ message; Encodings need only specify these common element types once. [TBD - extract common element types] Due to the extensibility of TeRQ, however, future specifications might define element types that an Encoding does not address. Profiles using those extensions and Encodings must explain their interaction.

8. Profiles

For particular deployment environments, only one particular Binding, Encoding and set of Attributes or other extended elements may be meaningful. Future specifications may therefore define TeRQ Profiles, which describe a particular deployment environment and the Binding, Encoding and set of Attributes or elements it requires.

Profiles may be extensible, but any Attributes or elements required to negotiate support for extensions must be defined within the Profile.

9. Security Considerations

[TBD]

10. IANA Considerations

This document creates a registry of element and subelements for use with this framework. This registry is extensible, with an IANA Registration policy of Specification Required. Any new element registered must supply the name of the element, the name of the parent element in the data model, and a code point as described below.

To facilitate interoperability of binary encodings, this document specifies a code points associated with all registered elements and subelements in Queries and Responses. Any new specification that extends the list of elements or subelements of this framework must provide a binary code point for their element.

The initial population of this registry is: [[TBD](#)]

This document furthermore creates a registry of Response Codes. [[TBD](#)]

[11.](#) Acknowledgements

12. Informative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.

Author's Address

Jon Peterson
NeuStar, Inc.

Email: jon.peterson@neustar.biz