

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 7, 2016

J. Peterson
Neustar, Inc.
July 6, 2015

A Framework and Information Model for Telephone-Related Queries (TeRQ)
draft-peterson-terq-04

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 information 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 7, 2016.

Copyright Notice

Copyright (c) 2015 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	3
2. Motivation	3
3. Use Cases	5
3.1. Number Translation with Multiple Authorities	5
3.2. Customer name queries	5
3.3. Pre-port validation	6
3.4. Caller-ID Spoofing prevention	6
3.5. Prefix-based route caching	7
3.6. Inventory search	7
3.7. Motivation for Extensions	7
3.7.1. SPEERMINT Number Translation	7
4. Overview of the Framework	8
5. Transport Independence	8
5.1. Bindings	9
5.2. Encodings	10
5.3. Profiles	11
6. The Information Model	11
6.1. Source	11
6.1.1. Query Source	11
6.1.2. Query Intermediary	12
6.1.3. Route Source	12
6.2. Subject	13
6.2.1. Telephone Number	13
6.3. Attributes	13
6.4. Records	14
6.4.1. Attributes	14
6.4.2. Authority	14
6.4.3. Priority	14

6.4.4. Expiration	14
6.5. Response Code	14
6.6. Signature	15
7. Element Types	15
7.1. Telephone Number Type	15
7.1.1. TN Range Type	15
7.2. Domain Name Type	15
7.3. Uniform Resource Indicator (URI) Type	15
7.4. Internet Protocol (IP) Address Type	16
7.5. Service Provider Identifier (SPID) Type	16
7.6. Trunk Group Type	16
7.7. Display Name Type	16
7.8. Expiry Type	16
7.9. Priority Type	16
7.10. Extension Type	17
8. Attributes	17
8.1. Routing Attributes	17
8.2. Administrative Attributes	17
9. Security Considerations	17
10. IANA Considerations	18
11. Acknowledgements	18
12. Informative References	18
Author's Address	20

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 such as SIP ([RFC3261]) 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 ([RFC6116]) 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 ([RFC5067]), 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 ([RFC6406]), 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 [RFC6950].

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 details queries and responses in an abstract architecture. This document specifies no particular syntax or encoding for queries or responses, but instead describes an extensible information model for the semantics of queries and responses that future specifications might encode in accordance with application needs.

3. Use Cases

This section records several motivating use cases for the TeRQ framework.

3.1. Number Translation with Multiple Authorities

An Internet-based VoIP client places a call destined for a telephone number. An end user and a service provider both want to provision data against the same telephone number; for example, a service provider might want to provision an endpoint address on an Internet gateway for the number, whereas an end user might want to provision the preferred voicemail service for the number. A directory service can permit multiple authorities to provision data for the same telephone number; clients query this service. Clients who query for this data might have a trust relationship with either authority or both. When a client launches a query, it should receive in response any records that authorities authorize the client to receive, allowing the client to decide what it should trust and use. As the multiple authorities provision records at the directory, they sign those records, and when the client receives a response it validates the signatures on the records and trusts those records, or not, based on its association with the signer, independent of any security relationship with the directory.

Translations should be available for nationally specific numbers, including freephone numbers.

A very similar use case could also be constructed for SMS routing (including short codes).

3.2. Customer name queries

An Internet gateway receives a call from the PSTN. The gateway wants to put the calling number (IAM CIN) into the username portion of the From header field value of a SIP request, and also to populate the display-name of that header field. The gateway therefore launches a query to a CNAM service, which may or may not be the same as any services used for number translation. The CNAM service only accepts requests from authorized parties with whom it has a billing relationship. Since the Internet gateway launching the query is only one of many gateways in its administrative domain, not every gateway will have a trust relationship with the CNAM service. Instead, the gateways send their requests to a local intermediary which aggregates requests and maintains a trust relationship with the CNAM service.

Ideally, if the gateway uses the same service for number translation and for CNAM, it should be able to place both requests in the same

message: one for the called number, flagged for translation, and one for the calling number, flagged for CNAM.

Under high volumes, the intermediary maintains a transport connection to the CNAM service, rather than opening a new socket and re-negotiating security for each individual request. The intermediary may also bundle multiple numbers into a single request, and expect to get back a response with multiple records associated with those numbers. In both cases, a transaction number is used to match requests to responses.

Finally, the intermediary authenticates sources of traffic and authorizes only gateways to receive responses, as CNAM data is sensitive and the CNAM service may charge for transactions.

3.3. Pre-port validation

A mall kiosk that sells cellular telephones has a customer that wants to purchase a new phone and port their old number onto the phone. Porting needs to be validated on the spot and typically completed in a very short time frame (say within fifteen minutes). The new service provider for the number needs to make a query to an intercarrier communications process (ICP) service to validate the customer with the old service provider. In order to validate the port, the new service provider needs to submit the telephone number, the customer's name and customer's zip code. The ICP needs to respond either confirming that the customer information is correct for the number in question or not.

The responses to ICP queries are potentially privacy-sensitive. It is not feasible for every mall kiosk to have a direct relationship with this database, therefore requests go through an intermediary which has a trust relationship with the ICP service.

3.4. Caller-ID Spoofing prevention

An SMS service bureau receives an SMS message from a particular telephone number. It wants to be able to consult an authoritative service to ascertain whether or not that calling number is allocated and SMS capable. The bureau sends a request to the service to determine if the number in question exists and has an SMS capability. Only if a record is returned proving that the number is SMS capable does the bureau forward the SMS to its destination.

A similar use case could be constructed for voice calls. For more on these similar use cases, see [RFC7340].

3.5. Prefix-based route caching

A soft client on a tablet attempts to call out to a telephone number. The client has a pre-existing association with a service that performs number translation on its behalf; the client knows the address of an intermediary belonging to the service, and has security credentials to pass requests through that intermediary. When the intermediary forwards the request to the service, the service returns a response indicating that the entire thousand-block to which that number belongs is routed to an enterprise with an Internet PBX. The intermediary receives this response along with a time-to-live and caches the response locally. When subsequent requests come in from clients, the intermediary can match the requests against this prefix, and return the appropriate response without needing to consult the service.

3.6. Inventory search

A Internet service provider provisions many telephone numbers within a given number range. The provider later wants to verify which numbers are associated with the address of a particular SMSC, perhaps an SMSC that has experienced a failure. The service provider thus wants to formulate a search query across the entire number range, requesting only those numbers that have that association. The service where the numbers are provisioned must be able to authenticate the service provider as this sort of search operation would not be authorized for end users.

3.7. Motivation for Extensions

While the current version of this specification focuses on a small core set of features, the TeRQ framework should be extensible to support use cases with alternative identifiers and scopes.

3.7.1. SPEERMINT Number Translation

An Internet gateway receives a call from the PSTN destined for a telephone number. The gateway resides in a walled garden that has numerous peering points with other administrative domains, including through a number of clearinghouses, typical of a SPEERMINT architecture. The gateway queries two services to determine where it should deliver the call. The gateway first makes a number translation request of a public directory, which returns a service provider identifier (SPID) of the network to which the call should be delivered (LUF). The gateway then makes a query to a private directory, internal to its walled garden, to translate that SPID into the address of the proper point-of-interconnection to exit the walled garden (LRF).

In this case, the SPID might take the form of a numerical identifier, a domain name or other identifier; behind the scenes, the internal private directory may contain links between several different forms of identifiers.

The internal private directory may respond with a different POI depending on which gateway is asking - a USA West Coast gateway might get a different answer than an East Coast gateway. The directory therefore authenticates incoming queries to identify the originating gateway and serve a customized answer.

Although the internal private directory is inherently trusted by the gateway, the public directory (which returns the SPID) is not directly trusted by the gateway. The data in the public directory, however, is provisioned by authorities, including the number owners. As they provision records at the public gateway, they sign those records, and when the gateway receives a response it validates the signatures on the records and trusts those records, or not, based on its association with the signer, independent of any security relationship with the directory.

4. 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.

5. Transport Independence

The information 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.

5.1. 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. Bindings must have a means of providing mutual authentication, integrity and confidentiality between Clients, Intermediaries and Services. If a Binding supports TLS, for example, these features can be provided by using TLS in an appropriate deployment environment.

5.2. Encodings

A TeRQ Encoding specifies how the Query and Response 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. Encodings must also define whether or not they provide a bundling feature that allows multiple Queries to be carried within particular objects or mappings.

Every Encoding must specify how each semantic Element Type of a Query and Response will be represented. For all baseline TeRQ Attributes and Element Types, 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. 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.

Encodings must also describe the security services they make available. In particular, encodings must describe a means of providing authentication of the Sources and Authorities of Queries and Responses respectively, as well as an integrity check over critical elements including the Subject of Queries and the Record of Responses.

[TBD - we may define more about the computation of this signature, including canonicalization of elements, in this framework, and make it a requirement for encodings to support this mechanism]

5.3. Profiles

For particular deployment environments, only one 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.

6. The Information 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.

6.1. 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.

6.1.1. 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 by an intermediary 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.

6.1.2. 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.

6.1.3. 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.

6.2. Subject

All Queries have 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, including Profiles, may define alternative Subject elements.

6.2.1. Telephone Number

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

A Telephone Number has a Type which designates which sort of telephone number the element contains. Types defined by this specification include: telephone number and telephone number range.

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.

6.3. Attributes

Attributes in this information model have a Name, which may optionally be associated with a Type and Value.

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 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. The format of the elements contained in the Attributes is given below in Section 7.

6.4. 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.

6.4.1. Attributes

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

6.4.2. 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.

6.4.3. 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.

6.4.4. 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.

6.5. 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]

6.6. Signature

A Record optionally concludes with a Signature element. The Signature element contains a signature over the concatenation of the other elements given the Record. Signatures are provided by the Authority responsible for the Record.

[Syntax TBD]

7. Element Types

7.1. Telephone Number Type

The telephone number type conforms to the telephone number syntax given in [RFC3966] Section 3, in the ABNF for "telephone-subscriber."

Type Code: T

[TBD - need for subtying? E.164, Service Code, Short Code, Prefix, Nationally-Specific and Unknown.]

7.1.1. TN Range Type

The TN range type consists of a prefix of a telephone number (per [RFC3966] "telephone-subscriber"), and is semantically equivalent to all syntactically-valid telephone numbers below that prefix. For example, in the North American Numbering plan, the prefix 157143454 would be equivalent to all numbers ranging from 15714345400 to 15714345499.

[TBD - identify alternative ways of specifying ranges, potentially as separate element types]

Type Code: R

7.2. Domain Name Type

The domain name type conforms to the syntax of RFC1034 Section 3.5 and Section 2.1 of [RFC1123].

Type Code: D

7.3. Uniform Resource Indicator (URI) Type

The Uniform Resource Indicator (URI) type conforms to the syntax for URIs given in [RFC3986] (see Section 3).

Type Code: U

7.4. Internet Protocol (IP) Address Type

The IP Address type conforms to the ABNF syntax of either the IPv4address given in RFC3986 (Appendix A) or the IPv6reference of [RFC5954].

Type Code: I

7.5. Service Provider Identifier (SPID) Type

The SPID type consists of a four-digit number.

[TBD - introduce other elements for alternative SPID syntaxes]

Type Code: S

7.6. Trunk Group Type

The trunk group type conforms to the "trunk-group-label" ABNF given in [RFC4904] (Section 5).

Type Code: G

7.7. Display Name Type

The display name is a string of Unicode characters, UTF-8 encoded, with a maximum length of fifty octets.

Type Code: N

7.8. Expiry Type

The Expiry type is an absolute time conformant to the syntax of [RFC3339].

Type Code: E

7.9. Priority Type

The Priority type contains a number between 0 and 1, conforming to the specification of the "q" parameter of the Contact header field in [RFC3261].

Type Code: P

7.10. Extension Type

This code is reserved for future use.

Type Code: X

8. Attributes

All attributes have a Name, which consists of a string. Optionally an Attribute may take a Value, in which case it also has a Type. Broadly, Attributes are here divided into two categories: Routing Attributes and Administrative Attributes.

When an Attribute is specified, if it requires a Value which does not have a Type in the base TeRQ specification, that Type must be defined along with the Attribute.

8.1. Routing Attributes

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

8.2. Administrative Attributes

Administrative Attributes defined by this document include: CNAM (Type Display Name), SPID (Type SPID), dialplan (Type ?) [TBD]

9. Security Considerations

The framework of this document differs from previous efforts to manage telephone numbers on the Internet largely by offering a much richer set of security services. In particular, it requires that three entities be capable of authenticating themselves to one another at the layer of a binding: Clients, Intermediaries and Services. It furthermore requires object security at the encoding layer so that Sources and Authorities can sign data in order to authenticate Queries and Responses that may pass through Intermediaries, and moreover so that Authorities can prove to Clients that their Records are authoritative even when the Authority does not operate the Service. The requirements that bindings and encodings must satisfy to meet these security needs are specified in Section 5.

[TBD - more]

10. IANA Considerations

This specification defines several registries: A registry of Elements, a registry of Element Types, a registry of Attributes, and a registry of Response Codes.

This document creates a registry of Elements 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 information model, and a code point. [TBD]

This specification pre-provisions the Element Types registry with the entries given in Section 6. These elements are indexed by their Type Code. This registry is extensible, with an IANA Registration policy of Specification Required. Any new Element Type registered must supply the name of the Element Type, the name of the parent element in the information model, and a Type Code.

This specification creates an Attribute registry which is indexed by Attribute names. This registry is extensible, with an IANA Registration policy of Specification Required. Any new element registered must supply the name of Attribute, and list all Element Types that may be associated with Values of the Attribute.

This document furthermore creates a registry of Response Codes. This registry is pre-provisioned with the values given in Section 5.5. [TBD]

11. Acknowledgements

The authors would like to thank Paul Kyzviat and Dale Worley for their input into this specification.

12. Informative References

- [RFC1123] Braden, R., "Requirements for Internet Hosts - Application and Support", STD 3, RFC 1123, October 1989.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

- [RFC3324] Watson, M., "Short Term Requirements for Network Asserted Identity", RFC 3324, November 2002.
- [RFC3325] Jennings, C., Peterson, J., and M. Watson, "Private Extensions to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks", RFC 3325, November 2002.
- [RFC3339] Klyne, G., Ed. and C. Newman, "Date and Time on the Internet: Timestamps", RFC 3339, July 2002.
- [RFC3966] Schulzrinne, H., "The tel URI for Telephone Numbers", RFC 3966, December 2004.
- [RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, January 2005.
- [RFC4474] Peterson, J. and C. Jennings, "Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP)", RFC 4474, August 2006.
- [RFC4904] Gurbani, V. and C. Jennings, "Representing Trunk Groups in tel/sip Uniform Resource Identifiers (URIs)", RFC 4904, June 2007.
- [RFC4916] Elwell, J., "Connected Identity in the Session Initiation Protocol (SIP)", RFC 4916, June 2007.
- [RFC5039] Rosenberg, J. and C. Jennings, "The Session Initiation Protocol (SIP) and Spam", RFC 5039, January 2008.
- [RFC5067] Lind, S. and P. Pfautz, "Infrastructure ENUM Requirements", RFC 5067, November 2007.
- [RFC5727] Peterson, J., Jennings, C., and R. Sparks, "Change Process for the Session Initiation Protocol (SIP) and the Real-time Applications and Infrastructure Area", BCP 67, RFC 5727, March 2010.
- [RFC5954] Gurbani, V., Carpenter, B., and B. Tate, "Essential Correction for IPv6 ABNF and URI Comparison in RFC 3261", RFC 5954, August 2010.
- [RFC6116] Bradner, S., Conroy, L., and K. Fujiwara, "The E.164 to Uniform Resource Identifiers (URI) Dynamic Delegation Discovery System (DDDS) Application (ENUM)", RFC 6116, March 2011.

- [RFC6406] Malas, D. and J. Livingood, "Session PEERing for Multimedia INTERconnect (SPEERMINT) Architecture", RFC 6406, November 2011.
- [RFC6461] Channabasappa, S., "Data for Reachability of Inter-/Intra-Network SIP (DRINKS) Use Cases and Protocol Requirements", RFC 6461, January 2012.
- [RFC6698] Hoffman, P. and J. Schlyter, "The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA", RFC 6698, August 2012.
- [RFC6950] Peterson, J., Kolkman, O., Tschofenig, H., and B. Aboba, "Architectural Considerations on Application Features in the DNS", RFC 6950, October 2013.
- [RFC7340] Peterson, J., Schulzrinne, H., and H. Tschofenig, "Secure Telephone Identity Problem Statement and Requirements", RFC 7340, September 2014.

Author's Address

Jon Peterson
Neustar, Inc.

Email: jon.peterson@neustar.biz