        A Framework and Information Model for Telephone-Related Information
                                (TeRI)
                     draft-peterson-modern-teri-00

Abstract

   As telephone services migrate to the Internet, Internet applications
   require tools to access and manage information about telephone
   numbers. This document specifies a protocol-independent framework
   and information model for managing service and adminsitration data
   associated with telephone numbers.

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 April 21, 2016.

Table of Contents

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). Increasingly, real-time communications is migrating
   to the Internet, and bringing telephone numbers with it. As
   identifiers, however, telephone numbers differ fundamentally from
   those 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. To help telephone numbers work
   similarly on the Internet, a number of efforts have specified
   mechanisms to manage and retrieve information about telephone numbers
   via network services.

   For example, the ENUM ([RFC6116]) effort specified a public DNS
   profile for translating telephone numbers into URIs. Due to the
   difficulty of coordinating the public administration of telephone
   numbers in the DNS, this work transitioned to "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"). To provision the
   data associated with telephone numbers, the DRINKS working group
   ([RFC6461]) designed systems for uploading back-end data to the
   services that would answer ENUM queries.

None of the preceding efforts, however, encompassed the entire
lifecycle of a telephone number as an Internet identifier. They
focused largely on service data, on how to "resolve" a telephone
number to a location on the Internet, rather than on administrative
questions of how numbers are acquired, how the entities associated
with telephone numbers are authorized to provision data, and how what
kinds of systems need to be in place to allow a diverse community of
devices, applications and uses to manage numbers. Early
considerations were moreover based on overlapping, but not entirely
consistent, information models: intrinsic limitations in the DNS kept
the queries and responses of ENUM relatively simple, whereas the
DRINKS provisioning system considered a much richer syntax.

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. This includes devices, services, and
applications on the Internet that make use of telephone numbers and
need to distribute and manage numbering inventory: for example, an
Internet-enabled PBX that might want to automate the process for
allowing new connected phones to acquire numbers and provision
contact information for their users. 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. The use cases
associated with these functions are detailed in
[I-D.peterson-modern-problems].

Therefore, this document proposes a reconsideration of telephone
service and administration data on the Internet, based on an
information model that allows records associated with telephone
number to be created, modified and accessed through network
interfaces. This document specifies no particular syntax or encoding
for queries or responses, but instead describes an extensible
information model for the semantics of provisioning and querying
operations associated with a telephone number.

3. The Information Model

The fundamental building block of the TeRI model is the Record. A
Record is created by an Authority who has authority over a particular
telephone number or a set of numbers. There may be more than one
Authority who is authorized to create Records for a particular
telephone number, and a TeRI service may have multiple Records
corresponding to a single telephone number, including potentially

Records associated with a range of numbers including a particular
telephone number. Under various circumstances detailed in Section 4,
participants in the numbering ecosystem may create, read, update, and
modify Records.

Records contain Elements that hold data about the telephone number.
Elements in this information model have a Name, which may optionally
be associated with a Type and Value. Elements are grouped into
Service Elements and Administrative Elements.

3.1. Record Elements

A Record is made up of Elements, which may be either Service Data
Elements or Administrative Data Elements.

3.1.1. Identifier

Every Record has an Identifier, which is a globally unique identifier
of the Record. The Identifier will typically be created at the same
time as the Record itself, at a time when an assignment or delegation
has occurred (as described in [I-D.peterson-modern-problems]).

3.1.2. Authority

Every Record contains an Authority element 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. The
Authority element ideally gives a logical identity of the source of
the data. A public key value may also be designated by the Authority
element.

3.1.3. Contact

Every Record has at least one Contact. The Contact contains
administrative data about the assignee of the telephone number,
though additional Contacts may contain information about delegates
(as defined in [I-D.peterson-modern-problems]).

3.1.4. Subject

Every Record has a Subject. As the TeRI record concerns telephone
numbers, the Subject of a Record is either a telephone number type or
a telephone number range type.

3.1.5. Service

   Records optionally have one or more Service entries. A Service may
   be of any Service Type, as given in Section 3.2.1.

3.1.5.1. 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.

3.1.5.2. 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.

3.1.6. Signature

   Optionaly, a Record contains 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]

3.2. Element Value Types

   The remainder of a Record is made up of Elements. Elements types ae
   specified in this section. Every Element Type has a Type Code. A
   Type Code is used as a short form for the Element in a Record.

3.2.1. Service Types

3.2.1.1. 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.]

3.2.1.1.1. 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

3.2.1.2. 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

3.2.1.3. 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

3.2.1.4. 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

3.2.1.5. Trunk Group Type

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

Type Code: G

3.2.1.6. Service Provider Identifier (SPID) Type

The SPID type consists of a four-digit number.

[TBD - introduce other elements for alternative SPID syntaxes]

Type Code: ?

3.2.2. Public Key Type

The Credential type consists of a public key [encoding TBD].

Type Code: C

3.2.3. Contact Type

The contact type follows the conventions of jCard [RFC7095].

Type Code: C

3.2.4. Expiry Type

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

Type Code: E

3.2.5. 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

3.2.6. Record Identifier Type

The Record Identifier Type consists of a unique identifier for a
record [format TBD].

Type Code: U

3.2.7. Signature

[Syntax TBD]

Type Code: S

3.2.8. Extension Type

This code is reserved for future use.

Type Code: X

[4](#). Operations

   In this section are detailed the three TeRI Operations: Acquisition,
   Management, and Retrieval Operations.

[4.1](#). Common to All Operations

   All Operations in the TeRI model consist of Requests and Responses.
   A Request from a TeRI client to a service may attempt to create,
   read, update, or delete TeRI Records. Requests may focus only on
   particular parts of a TeRI record. A Response gives the result of
   the Operation back to the client, which may indicate success of
   failure.

[4.1.1](#). Requests

   All TeRI Requests have a Source, a Subject, and optionally a set of
   Attributes which further specify the nature of the Request. Some
   Requests will know the Identifier of the Record they concern, and may
   convey that in an Attribute; others will query for all Records
   matching a given Subject.

[4.1.1.1](#). Source

   The Source is a required element in all Requests. In this
   specification, two categories of Sources are defined: Request Source
   and Request Intermediary. At least one of these Sources must be
   present in a Retrieval Request, and multiple Sources are permitted.
   Responses do not contain a Source.

   Future specifications may extend the set of Source types.

[4.1.1.1.1](#). Request Source

   Every Request generated by a Client has a Request Source, which
   identifies the originator of the Request. This represents the
   logical identity of the user or service provider who first sent the
   Request, rather than the identity of any Intermediate entity. This
   field is provided in the Source to authenticate the poser of the
   Request, 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
   Request's Source from a Service. The removal of the Request's Source
   by an intermediary is permitted by TeRI, but any Intermediary that
   removes the Request Source must provide a Request Intermediary for
   the Source element.

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

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

4.1.1.1.2. Request Intermediary

Optionally, Requests may contain one or more Request Intermediary elements in the Source. A Request Intermediary resides between the originator of the Request (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 Requests to the Service.

The Request Intermediary element, like the Request Source, contains the logical identity of the service that relayed the Request. 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 Request Source.

A Request Intermediary element has a Type, which indicates how the logical identity of the Intermediary has been represented. The Type element of the Request Intermediary is extensible. Initial values include a domain name, an X.509 certificate subject, or a URI.

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

4.1.1.2. Subject

All Requests have a Subject. The Subject identifies the resource that the Request concerns. Responses only contain a Subject if the Subject of the Response differs from that of the original Request, 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.

4.1.1.2.1. Attributes

TeRI Attributes consist of a Name with an optional Type and an Optional Value. Most Attributes are specific to the Operation.

4.1.2. Responses

   All TeRI responses consist of a Response Code and optionally a set of
   Attributes which convey further information about the Operation.
   Most Attributes are specific to the Operation.

4.1.2.1. Response Code

   All Responses contain a Response Code.

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

   [TBD]

4.2. The Acquisition Operation

   An Acquisition Request has a Source and a Subject, and may have one
   or more Attributes. An Acquisition Response has a Response Code, and
   will contain one Record if it is succesful.

   The Subject of an Acquisition Request always specifies a Telephone
   Number Type or a Telephone Number Range Type. If the Subject
   contains a particular telephone number, then the Acquisition Request
   is a Request to acquire that particular telephone number. If it is a
   range, the Acquisition Request should be considered to be for the
   entire range, but Attributes of the Request may limit the scope of
   the resources requested. The Service will determine whether or not
   the Client is authorized to acquire the resources in question based
   on the Source of the Acqustion Request.

   The Response to an Acquisition Request will contain a Success
   Response Code if the resource can be allocated. The Subject of a
   Success Response will always contain the Telephone Number Type or
   Telephone Number Range that has been allocated. A successful
   Acqustion Response must contain a Record with a Identifier Element;
   that Record may also contain a Public Key attribute. By default,
   this Record will contain only Administrative Elements, without
   Service Elements. If a requested telephone number (or range) is
   already allocated, or a telephone number in the specified range is
   not available, then a Subject Conflict Response Code is returned.

4.3. The Management Operation

A Management Request comprises a Source, a Subject, and one or more
Records; it also may contain one or more Attributes. A Management
Request contains a Response Code, and optionally may contain a
Record.

The Subject of a Management Request always specifies a Telephone
Number Type or a Telephone Number Range Type. If the Subject
contains a particular telephone number, then the Acquisition Request
is a Request to acquire that particular telephone number. If it is a
range, the Acquisition Request should be considered to be for the
entire range.

A Management Request contains at least one Record; it may contain
multiple Records. Each Record in the Management Request must contain
a Record Identifier Element which designates the Record that the
Client is requesting that the Service replace with the Record
included in the Management Request. The Service will determine
whether or not the Client is authorized to modify the Record in
question via the Source of the Management Request.

4.4. The Retrieval Operation

Every Retrieval Request comprises a Source and a Subject, and may
have one or more Attributes. A Retrieval Response has a Response
Code, optionally one or more Records, and optionally a Subject, if
the Subject differs from that of the Request.

Retrieval Requests optionally contain Attributes; a Request with no
specified Attributes requests that the Service return any Attributes
associated with the Subject. In a Request, the presence of one or
more Attributes limits the scope of the Request to Records about the
Subject containing those Attributes. Typically an Attribute will
specify a Service or Service Type that the Client seeks Records for.

Retrieval Responses contain one or more Records. At least one Record
will always be present in a successful Response.

4.5. Common Attributes

Attributes are broadly divided between Service Attributes and
Administrative Attributes. Service Attributes provide information
required to route communications, including URIs. The format of the
elements contained in the Attributes is given in Section 3.2.

4.5.1. Administrative Attributes

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

4.5.2. Service Attributes

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

4.5.2.1. Route Source

   Optionally, Retrieval Requests may contain a Route Source Attribute
   which identifies a reference point in the network from which any
   Service 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 Request 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. Implementing Opertions

   This framework specifies an abstract Request/Response protocol that
   enables a Client to send Requests to a Service about telephone
   numbers or related telephone services. Requests 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 Request, and optionally includes one or
   more Attributes to focus the scope of the Request. When a Service
   receives a Request, 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 Request followed;
   transactional identifiers set by the Client and Service correlate the
   Request to the Response and assist any intermediary routing.

5.1. Transport Independence

   The information model provided for Requests 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 Requests 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 Requests and
   Responses survive interworking through translations from one encoding
   to another; for example, when an Intermediary receives a binary
   Request 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.

5.2. Bindings

   A TeRI Binding is an underlying protocol that carries Requests 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 TeRI.
   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 TeRI entities over which many, potentially
   unrelated, Requests 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
   Request would be inefficient. A transactional Binding creates a
   temporary connection between two TeRI entities for the purpose of
   fulfilling a single Request; any Responses to the Request will use
   the same connection to return to the sender of the Request. Finally,
   a non-transactional Binding does not rely on any sort of connection
   semantics: the senders of Requests 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 TeRI 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 Requests 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.

5.3. Encodings

A TeRI Encoding specifies how the Request 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 Requests 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 Requests to be carried within particular objects or
mappings.

Every Encoding must specify how each semantic Element Type of a
Request and Response will be represented. For all baseline TeRI
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 TeRI message; Encodings need only specify these common
element types once. Due to the extensibility of TeRI, 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 Requests
and Responses respectively, as well as an integrity check over
critical elements including the Subject of Requests 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]

5.4. 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 TeRI 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. 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
Requests 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 5.1.

[TBD - more]

7. 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]

8. Acknowledgements

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

9. Informative References

[I-D.peterson-modern-problems]
          Peterson, J. and T. McGarry, "Modern Problem Statement,
          Use Cases, and Framework", draft-peterson-modern-
          problems-01 (work in progress), July 2015.

[RFC1123] Braden, R., Ed., "Requirements for Internet Hosts -
          Application and Support", STD 3, RFC 1123,
          DOI 10.17487/RFC1123, October 1989,
          <http://www.rfc-editor.org/info/rfc1123>.

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

   [RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston,
              A., Peterson, J., Sparks, R., Handley, M., and E.
              Schooler, "SIP: Session Initiation Protocol", RFC 3261,
              DOI 10.17487/RFC3261, June 2002,
              <http://www.rfc-editor.org/info/rfc3261>.

   [RFC3324] Watson, M., "Short Term Requirements for Network Asserted
              Identity", RFC 3324, DOI 10.17487/RFC3324, November 2002,
              <http://www.rfc-editor.org/info/rfc3324>.

   [RFC3325] Jennings, C., Peterson, J., and M. Watson, "Private
              Extensions to the Session Initiation Protocol (SIP) for
              Asserted Identity within Trusted Networks", RFC 3325,
              DOI 10.17487/RFC3325, November 2002,
              <http://www.rfc-editor.org/info/rfc3325>.

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

   [RFC3966] Schulzrinne, H., "The tel URI for Telephone Numbers",
              RFC 3966, DOI 10.17487/RFC3966, December 2004,
              <http://www.rfc-editor.org/info/rfc3966>.

   [RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform
              Resource Identifier (URI): Generic Syntax", STD 66,
              RFC 3986, DOI 10.17487/RFC3986, January 2005,
              <http://www.rfc-editor.org/info/rfc3986>.

   [RFC4474] Peterson, J. and C. Jennings, "Enhancements for
              Authenticated Identity Management in the Session
              Initiation Protocol (SIP)", RFC 4474,
              DOI 10.17487/RFC4474, August 2006,
              <http://www.rfc-editor.org/info/rfc4474>.

   [RFC4904] Gurbani, V. and C. Jennings, "Representing Trunk Groups in
              tel/sip Uniform Resource Identifiers (URIs)", RFC 4904,
              DOI 10.17487/RFC4904, June 2007,
              <http://www.rfc-editor.org/info/rfc4904>.

   [RFC4916] Elwell, J., "Connected Identity in the Session Initiation
              Protocol (SIP)", RFC 4916, DOI 10.17487/RFC4916, June
              2007, <http://www.rfc-editor.org/info/rfc4916>.

   [RFC5039] Rosenberg, J. and C. Jennings, "The Session Initiation
              Protocol (SIP) and Spam", RFC 5039, DOI 10.17487/RFC5039,
              January 2008, <http://www.rfc-editor.org/info/rfc5039>.

   [RFC5067] Lind, S. and P. Pfautz, "Infrastructure ENUM
              Requirements", RFC 5067, DOI 10.17487/RFC5067, November
              2007, <http://www.rfc-editor.org/info/rfc5067>.

   [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, DOI 10.17487/RFC5727, March 2010,
              <http://www.rfc-editor.org/info/rfc5727>.

   [RFC5954] Gurbani, V., Ed., Carpenter, B., Ed., and B. Tate, Ed.,
              "Essential Correction for IPv6 ABNF and URI Comparison in
              RFC 3261", RFC 5954, DOI 10.17487/RFC5954, August 2010,
              <http://www.rfc-editor.org/info/rfc5954>.

   [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,
              DOI 10.17487/RFC6116, March 2011,
              <http://www.rfc-editor.org/info/rfc6116>.

   [RFC6406] Malas, D., Ed. and J. Livingood, Ed., "Session PEERing for
              Multimedia INTerconnect (SPEERMINT) Architecture",
              RFC 6406, DOI 10.17487/RFC6406, November 2011,
              <http://www.rfc-editor.org/info/rfc6406>.

   [RFC6461] Channabasappa, S., Ed., "Data for Reachability of Inter-
              /Intra-NetworK SIP (DRINKS) Use Cases and Protocol
              Requirements", RFC 6461, DOI 10.17487/RFC6461, January
              2012, <http://www.rfc-editor.org/info/rfc6461>.

   [RFC6698] Hoffman, P. and J. Schlyter, "The DNS-Based Authentication
              of Named Entities (DANE) Transport Layer Security (TLS)
              Protocol: TLSA", RFC 6698, DOI 10.17487/RFC6698, August
              2012, <http://www.rfc-editor.org/info/rfc6698>.

   [RFC6950] Peterson, J., Kolkman, O., Tschofenig, H., and B. Aboba,
              "Architectural Considerations on Application Features in
              the DNS", RFC 6950, DOI 10.17487/RFC6950, October 2013,
              <http://www.rfc-editor.org/info/rfc6950>.

   [RFC7095] Kewisch, P., "jCard: The JSON Format for vCard", RFC 7095,
              DOI 10.17487/RFC7095, January 2014,
              <http://www.rfc-editor.org/info/rfc7095>.

   [RFC7340] Peterson, J., Schulzrinne, H., and H. Tschofenig, "Secure
              Telephone Identity Problem Statement and Requirements",
              RFC 7340, DOI 10.17487/RFC7340, September 2014,
              <http://www.rfc-editor.org/info/rfc7340>.

Author's Address

   Jon Peterson
   Neustar, Inc.

   Email: jon.peterson@neustar.biz