

Network Working Group
Internet-Draft
Intended status: Informational
Expires: April 05, 2014

E. Ivov
Jitsi
P. Saint-Andre
Cisco Systems, Inc.
E. Marocco
Telecom Italia
October 02, 2013

CUSAX: Combined Use of the Session Initiation Protocol (SIP) and the
Extensible Messaging and Presence Protocol (XMPP)
draft-ivov-xmpp-cusax-09

Abstract

This document suggests some strategies for the combined use of the Session Initiation Protocol (SIP) and the Extensible Messaging and Presence Protocol (XMPP) both in user-oriented clients and in deployed servers. Such strategies, which mainly consist of configuration changes and minimal software modifications to existing clients and servers, aim to provide a single, full-featured, real-time communication service by using complementary subsets of features from SIP and from XMPP. Typically such subsets consist of telephony capabilities from SIP and instant messaging and presence capabilities from XMPP. This document does not define any new protocols or syntax for either SIP or XMPP, and by intent does not attempt to standardize "best current practices". Instead, it merely aims to provide practical guidance to those who are interested in the combined use of SIP and XMPP for real-time communication.

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 05, 2014.

Copyright Notice

Copyright (c) 2013 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.

Table of Contents

1. Introduction	2
2. Client Bootstrap	5
3. Operation	6
3.1. Server-Side Setup	7
3.2. Service Management	7
3.3. Client-Side Discovery and Usability	8
3.4. Indicating a Relationship Between SIP and XMPP Accounts	9
3.5. Matching Incoming SIP Calls to XMPP JIDs	10
4. Multi-Party Interactions	11
5. Federation	12
6. Summary of Suggested Strategies	13
7. IANA Considerations	14
8. Security Considerations	14
9. References	15
9.1. Normative References	15
9.2. Informative References	16
Appendix A. Acknowledgements	17
Authors' Addresses	18

1. Introduction

Historically SIP [RFC3261] and XMPP [RFC6120] have often been implemented and deployed with different purposes: from its very start SIP's primary goal has been to provide a means of conducting "Internet telephone calls". XMPP on the other hand, has, from its Jabber days, been mostly used for instant messaging and presence [RFC6121], as well as related services such as groupchat rooms [XEP-0045].

For various reasons, these trends have continued through the years even after each of the protocols had been equipped to provide the features it was initially lacking:

- o In the context of the SIMPLE working group, the IETF has defined a number of protocols and protocol extensions that not only allow for SIP to be used for regular instant messaging and presence but that also provide mechanisms for related features such as multi-party chat, server-stored contact lists, and file transfer [RFC6914].
- o Similarly, the XMPP community and the XMPP Standards Foundation have worked on defining a number of XMPP Extension Protocols (XEPs) that provide XMPP implementations with the means of establishing end-to-end sessions. These extensions are often jointly referred to as Jingle [XEP-0166] and arguably their most popular use case is audio and video calling [XEP-0167].

However, although SIP has been extended for messaging and presence and XMPP has been extended for voice and video, the reality is that SIP remains the protocol of choice for telephony-like services and XMPP remains the protocol of choice for IM and presence services. As a result, a number of adopters have found themselves needing features that are not offered by any single-protocol solution, but that separately exist in SIP and XMPP implementations. The idea of seamlessly using both protocols together would hence often appeal to service providers and users. Most often, such a service would employ SIP exclusively for audio, video, and telephony services and rely on XMPP for anything else varying from chat, contact list management, and presence to whiteboarding and exchanging files. Because these services and clients involve the combined use of SIP and XMPP, we label them "CUSAX" for short.

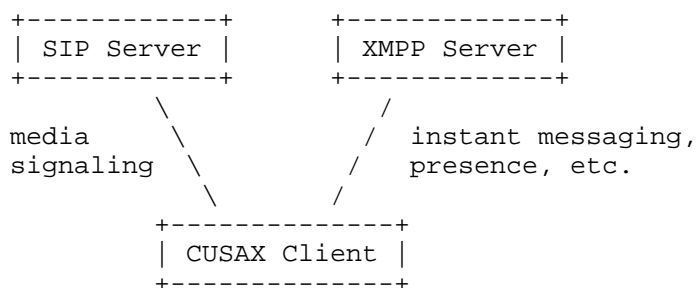


Figure 1: Division of Responsibilities

This document suggests different configuration options and minimal modifications to existing software so that clients and servers can

offer these hybrid services while providing an optimal user experience. It covers server discovery, determining a SIP Address of Record (AOR) while using XMPP, and determining an XMPP Jabber Identifier ("JID") from incoming SIP requests. Most of the text here pertains to client behavior but we also suggest certain server-side configurations and operational strategies. The document also discusses significant security considerations that can arise when offering a dual-protocol solution, and provides advice for avoiding security mismatches that would result in degraded communications security for end users.

Note that this document is focused on coexistence of SIP and XMPP functionality in end-user-oriented clients. By intent it does not define methods for protocol-level mapping between SIP and XMPP, as might be used within a server-side gateway between a SIP network and an XMPP network (a separate series of documents has been produced that defines such mappings). More generally, this document does not describe service policies for inter-domain communication (often called "federation") between service providers (e.g., how a service provider that offers a CUSAX service might communicate with a SIP-only or XMPP-only service), nor does it describe the reasons why a service provider might choose SIP or XMPP for various features.

This document concentrates on use cases where the SIP services and XMPP services are controlled by one and the same provider, since that assumption greatly simplifies both client implementation and server-side deployment (e.g., a single service provider can enforce common or coordinated policies across both the SIP and XMPP aspects of a CUSAX service, which is not possible if a SIP service is offered by one provider and an XMPP service is offered by another provider). Since this document is of an informational nature, it is not unreasonable for clients to apply some of the guidelines here even in cases where there is no established relationship between the SIP and the XMPP services (for example, it is reasonable for a client to provide a way for its users to easily start a call to a phone number or SIP URI found in a vCard or obtained from a user directory). However, the strategies to pursue in such cases are left to application developers.

This document makes a further simplifying assumption by discussing only the use of a single client, not use of and coordination among multiple endpoints controlled by the same user (e.g., user agents running simultaneously on a laptop computer, tablet, and mobile phone). Although user agents running on separate endpoints might themselves be CUSAX clients or might engage in different aspects of an interaction (e.g., a user might employ her mobile phone for audio and her tablet for video and text chat), such usage complicates the guidelines for developers of user agents and therefore is left as a matter of implementation for now.

It is important to note that this document does not attempt to standardize "best current practices" in the sense defined in the Internet Standards Process [RFC2026]. Instead, it collects together informational documentation about some strategies that might prove helpful to those who implement and deploy combined SIP-XMPP software and services. With sufficient use and appropriate modification to incorporate the lessons of experience, these strategies might someday form the basis for standardization of best current practices.

2. Client Bootstrap

One of the main problems of using two distinct protocols when providing one service is the impact on usability. Email services, for example, have long been affected by the mixed use of SMTP for outgoing mail and POP3 or IMAP for incoming mail. Although standard service discovery methods (such as the proper DNS records) make it possible for a user agent to locate the right host(s) for connect purposes, they do not provide the kind of detailed information that is needed to actually configure the user agent for use with the service. As a result, it is rather complicated for inexperienced users to configure a mail client and start using it with a new service, and as a result Internet service providers often need to provide configuration instructions for various mail clients. Client developers and communication device manufacturers on the other hand often ship with a number of so-called "wizard" interfaces that enable users to easily configure accounts with a number of popular email services. Although this may improve the situation to some extent, the user experience is still clearly sub-optimal.

While it should be possible for CUSAX users to manually configure their separate SIP and XMPP accounts (often using "wizards"), service providers offering CUSAX services to users of dual-stack SIP/XMPP clients ought to provide methods for online provisioning, typically by means of a web-based service at an HTTPS URL (naturally, single-purpose SIP services or XMPP services could offer such methods as well, but they can be especially helpful where the two aspects of the CUSAX service need to have several configuration options in common).

Although the specifics of such mechanisms are outside the scope of this document, they should make it possible for a service provider to remotely configure the clients based on minimal user input (e.g., only a user ID and password). As far as the authors are aware, no open protocol for endpoint configuration is yet available and adopted; however, application developers are encouraged to explore the potential for future progress in this space (e.g., perhaps based on technologies such as WebFinger [RFC7033]).

By default, when a CUSAX client is used in concert with SIP and XMPP accounts that have a CUSAX relationship (see Section 3.4), the client should disable audio and video calling over XMPP and disable instant messaging and presence over SIP. (It is a matter of implementation whether a CUSAX client allows a user to override these defaults in various ways, e.g., by domain, by individual contact, or by device.) The main advantage of this approach is that a client would employ the most relevant features from both SIP and XMPP when used in the context of a CUSAX service. Note that this default configuration does not apply to standalone SIP accounts or XMPP accounts, for which other settings are likely to be more appropriate (see Section 3.4 for details).

Once a client has been provisioned, it needs to independently log into the SIP account and XMPP account that make up the CUSAX "service" and then maintain both connections.

In order to improve the user experience, when reporting connection status a CUSAX client may wish to present the XMPP connection as an "instant messaging" or a "chat" account and the SIP connection as a "Voice and Video" or a "Telephony" connection. The exact naming is of course entirely up to implementers. The point is that, in cases where SIP and XMPP are components of a service offered by a single provider, such presentation could help users better understand why they are being shown two different connections for what they perceive as a single service. It could alleviate especially situations where one of these connections is disrupted while the other one is still active. Alternatively, the developers of a CUSAX client or the providers of a CUSAX service might decide to force a client to completely disconnect unless both aspects are successfully connected.

Clients may also choose to delay their XMPP connection until they have been successfully registered on SIP. This would help avoid the situation where a user appears online to her contacts but calling the user's client would fail because the user's client is still connecting to the SIP aspect of the CUSAX service.

3. Operation

Once a CUSAX client has been provisioned and authorized to connect to the corresponding SIP and XMPP services it would proceed by retrieving its XMPP roster.

The client should use XMPP for most forms of communication with the contacts from this roster, which will occur naturally because they were retrieved through XMPP. Audio/video features however, would typically be disabled in the XMPP stack, so media-related communication based on these features (e.g., direct calls, conferences, desktop streaming, etc.) would happen over SIP. The rest of this section describes deployment, discovery, usability and linking semantics that enable CUSAX clients to seamlessly use SIP for these features.

3.1. Server-Side Setup

In order for CUSAX to function properly, XMPP service administrators should make sure that at least one of the vCard [RFC6350] "tel" fields for each contact is properly populated with a SIP URI for the user's address at the SIP audio/video service provided by the CUSAX server. There are no limitations as to the form of that number. For example while it is desirable to maintain a certain consistency between SIP AORs and XMPP JIDs, that is by no means required. It is quite important however that the phone number or SIP AOR stored in the vCard be reachable through the SIP aspect of this CUSAX service. (The same considerations apply even if the directory storage format is not vCard storage over XMPP as described by [XEP-0054] or [XEP-0292].)

Administrators may also choose to include the "video" tel type defined in [RFC6350] for accounts that would be capable of handling video communication.

To ensure that the foregoing approach is always respected, service providers might consider validating the values of vCard "tel" fields before storing changes. Of course such validation would be feasible only in cases where a single provider controls both the XMPP and the SIP service since such providers would "know" (e.g., based on use of a common user database for both services) what SIP AOR corresponds to a given XMPP user.

3.2. Service Management

The task of operating and managing a standalone SIP service or XMPP service is not always easy. Combining the two into a unified service introduces additional challenges, including:

- o The necessity of opening additional ports on the client side if SIP functionality is added to an existing XMPP deployment or vice-versa.
- o The potential for important differences in security posture across SIP and XMPP (e.g., SIP servers and XMPP servers might support different TLS ciphersuites).
- o The need for, ideally, a common authentication backend and other infrastructure that is shared across the SIP and XMPP aspects of the combined service.
- o Coordinated monitoring and logging of the SIP and XMPP servers to enable the correlation of incidents and the pinpointing of problems.
- o The difficulty of troubleshooting client-side issues, e.g. if the client loses connectivity for XMPP but maintains its SIP connection.

Although separation of functionality (SIP for media, XMPP for IM and presence) can help to ease the operational burden to some extent, service providers are urged to address the foregoing challenges and similar issues when preparing to launch a CUSAX service.

Beyond the issues listed above, service providers might want to be aware of more subtle operational issues that can arise. For example, if a service provider uses different network operators for the SIP service and the XMPP service, end-to-end connectivity might be more reliable or consistent in one service than in the other service. Similar issues can arise when the media path and the signaling path go over different networks, even in standalone SIP or XMPP services. Providers of CUSAX services are advised to consider the potential for such topologies to cause operational challenges.

3.3. Client-Side Discovery and Usability

When rendering the roster for a particular XMPP account CUSAX clients should make sure that users are presented with a "Call" option for each roster entry that has a properly set "tel" field. This is the case even if calling features have been disabled for that particular XMPP account, as advised by this document. The usefulness of such a feature is not limited to CUSAX. After all, numbers are entered in vCards or stored in directories in order to be dialed and called. Hence, as long as an XMPP client has any means of conducting a call it may wish to make it possible for the user to easily dial any numbers that it learned through whatever means.

Clients that have separate triggers (e.g., buttons) for audio calls and video calls may choose to use the presence or absence of the "video" tel type defined in [RFC6350] as the basis for choosing whether to enable or disable the possibility for starting video calls (i.e., if there is no "video" tel type for a particular contact, the client could disable the "video call" button for that contact).

In addition to discovering phone numbers from vCards or user directories, clients may also check for alternative communication methods as advertised in XMPP presence broadcasts and Personal Eventing Protocol nodes as described in XEP-0152: Reachability Addresses [XEP-0152]. However, these indications are merely hints, and a receiving client ought not associate a SIP address and an XMPP address unless it has some way to verify the relationship (e.g., the vCard of the XMPP account lists the SIP address and the vCard of the SIP account lists the XMPP address, or the relationship is made explicit in a record provided by a trusted directory). Alternatively or in cases where vCard or directory data is not available, a CUSAX client could take the user's own address book as the canonical source for contact addresses.

3.4. Indicating a Relationship Between SIP and XMPP Accounts

In order to improve usability, in cases where clients are provisioned with only a single telephony-capable account they ought to initiate calls immediately upon user request without asking users to indicate an account that the call should go through. This way CUSAX users (whose only account with calling capabilities is usually the SIP part of their service) would have a better experience, since from the user's perspective calls "just work at the click of a button".

In some cases however, clients will be configured with more than the two XMPP and SIP accounts provisioned by the CUSAX provider. Users are likely to add additional stand-alone XMPP or SIP accounts (or accounts for other communications protocols), any of which might have both telephony and instant messaging capabilities. Such situations can introduce additional ambiguity since all of the telephony-capable accounts could be used for calling the numbers the client has learned from vCards or directories.

To avoid such confusion, client implementers and CUSAX service providers may choose to indicate the existence of a special relationship between the SIP and XMPP accounts of a CUSAX service. For example, let's say that Alice's service provider has opened both an XMPP account and a SIP account for her. During or after provisioning, her client could indicate that `alice@xmpp.example.com` has a CUSAX relationship to `alice@sip.example.com` (i.e., that they are two aspects of the same service). This way whenever Alice

triggers a call to a contact in her XMPP roster, the client would preferentially initiate this call through her example.com SIP account even if other possibilities exist (such as the XMPP account where the vCard was obtained or a SIP account with another provider). Similarly, the client would preferentially initiate textual chat sessions using her XMPP account.

If, on the other hand, no relationship has been configured or discovered between a SIP account and an XMPP account, and the client is aware of multiple telephony-capable accounts, it ought to present the user with the option of using XMPP Jingle as one method for engaging in audio and video interactions with a contact who has an XMPP address. This can help to ensure that a CUSAX user can complete audio and video calls with XMPP users who are not part of a CUSAX deployment.

3.5. Matching Incoming SIP Calls to XMPP JIDs

When receiving a SIP call, a CUSAX client may wish to determine the identity of the caller and a corresponding XMPP roster entry so that the receiving user could revert to chatting or other forms of communication that require XMPP. To do so, a CUSAX client could search the user's roster for an entry whose vCard has a "tel" field matching the originator of the call. In addition, in order to avoid the effort of iterating over the entire roster of the user and retrieving vCards for all of the user's contacts, the receiving client may guess at the identity of the caller based a SIP Call-Info header whose 'purpose' header field parameter has a value of "impp" as described in [RFC6993]. To enable this usage, a sending client would need to include such a Call-Info header in the SIP messages that it sends when initiating a call. An example follows.

```
Call-Info: <xmpp:alice@xmpp.example.com> ;purpose=impp
```

Note that the information from the Call-Info header should only be used as a cue: the actual AOR-to-JID binding would still need to be confirmed by the vCard of a contact in the receiving user's roster or through some other trusted means (such as an enterprise directory). If this confirmation succeeds the client would not need to search the entire roster and retrieve all vCards. Not performing the check might enable any caller (including malicious ones) to employ someone else's identity and perform various scams or Man-in-the-Middle attacks.

However, although an AOR-to-JID binding can be a helpful hint to the user, nothing in the foregoing paragraph ought to be construed as necessarily discouraging users, clients, or service providers from

accepting calls originated by entities that are not established contacts of the user (e.g., as reflected in the user's roster); that is a policy matter for the user, client, or service provider.

4. Multi-Party Interactions

CUSAX clients that support the SIP conferencing framework [RFC4353] can detect when a call they are participating in is actually a conference and can then subscribe to conference state updates as per [RFC4575]. A regular SIP user agent might also use the same conference URI for text communication with the Message Session Relay Protocol (MSRP). However, given that SIP's instant messaging capabilities would normally be disabled (or simply not supported) in CUSAX deployments, an XMPP Multi-User Chat (MUC) room [XEP-0045] associated with the conference can be announced/discovered through <service-uris> bearing the "groupextchat" purpose [I-D.ivov-groupextchat-purpose]. Similarly, an XMPP MUC room can advertise the SIP URI of an associated service for audio/video interactions using the 'audio-video-uri' field of the "muc#roominfo" data form [XEP-0004] to include extended information [XEP-0128] about the MUC room within XMPP service discovery [XEP-0030]; see [XEP-0045] for an example. These methods would enable a CUSAX-aware SIP conference server to advertise the existence of an associated XMPP chatroom, and for a CUSAX-aware XMPP chatroom to advertise the existence of an associated SIP conference server.

If a CUSAX client joins the MUC room associated with a particular call, it should not rely on any synchronization between the two. Both the SIP conference and the XMPP MUC room would function independently, each issuing and delivering its own state updates. Hence it is possible that that certain peers would temporarily or permanently be reachable in only one of the two conferences. This would typically be the case with single-stack clients that have only joined the SIP call or the XMPP MUC room. It is therefore important for CUSAX clients to provide a clear indication to users as to the level of involvement of the various participants: i.e., a user needs to be able to easily understand whether a certain participant can receive text messages, audio/video, or both.

At the level of the CUSAX service, it is also possible to enforce tighter integration between the XMPP MUC room and the SIP conference. Permissions, roles, kicks and bans that are granted and performed in the MUC room can easily be imitated by the conference focus/mixer into the SIP call. If, for example, a certain MUC member is muted, the conference mixer can choose to also apply the mute on the media stream corresponding to that participant. However, the details and exact level of such integration are entirely up to implementers and service providers.

The approach above describes one relatively lightweight possibility of combining SIP and XMPP multi-party interaction semantics without requiring tight integration between the two. As with the rest of this document, this approach is by no means normative. Implementations and future documents may define other methods or provide other suggestions for improving the unified communications user experience in cases of multi-user chats and conference calling.

5. Federation

In theory there are no technical reasons why federation (i.e., inter-domain communication) would require special behavior from CUSAX clients. However, it is worth noting that differences in administration policies may sometimes lead to potentially confusing user experiences.

For example, let's say atlanta.example.com observes the CUSAX policies described in this document. All XMPP users at atlanta.example.com are hence configured to have vCards that match their SIP identities. Alice is therefore used to making free, high-quality SIP calls to all the people in her roster. Alice can also make calls to the PSTN by simply dialing numbers. She may even be used to these calls being billed to her online account so she would be careful about how long they last. This is not a problem for her since she can easily distinguish between a free SIP call (one that she made by calling one her roster entries) from a paid PSTN call that she dialed as a number.

Then Alice adds xmpp:bob@biloxi.example.com. The Biloxi domain only has an XMPP service. There is no SIP server and Bob uses an XMPP-only client. However, Bob has added his mobile number to his vCard in order to make it easily accessible to his contacts. Alice's client would pick up this number and make it possible for Alice to start a call to Bob's mobile phone number.

This could be a problem because, other than the fact that Bob's address is from a different domain, Alice would have no obvious and straightforward cues telling her that this is in fact a call to the PSTN. In addition to the potentially lower audio quality, Alice may also end up incurring unexpected charges for such calls.

In order to avoid such issues, providers maintaining a CUSAX service for the users in their domain may choose to provide additional cues (e.g., a service-generated signal that triggers a user interface warning in a CUSAX client, an auditory tone, or a spoken message) indicating that a call would incur unexpected charges.

Another scenario arises when a SIP service allow communication only with intra-domain numbers; here Alice might be prevented from establishing a call with Bob's mobile phone. Providers should therefore make sure that calls to inter-domain numbers are flagged with an appropriate audio or textual warning.

6. Summary of Suggested Strategies

The following strategies are suggested for CUSAX user agents:

1. By default, prefer SIP for audio and video, and XMPP for messaging and presence.
2. Use XMPP for all forms of communication with the contacts from the XMPP roster, with the exception of features that are based on establishing real-time sessions (e.g. audio/video calls), for which SIP should be used.
3. Provide online provisioning options for providers to remotely setup SIP and XMPP accounts so that users wouldn't need to go through a multi-step configuration process.
4. Provide online provisioning options for providers to completely disable features for an account associated with a given protocol (SIP or XMPP) if the features are preferred in another protocol (XMPP or SIP).
5. Present a "Call" option for each roster entry that has a properly set "tel" field in the vCard or equivalent.
6. If the client is provisioned with only a single telephony-capable account, initiate calls immediately upon user request without asking users to indicate an account that the call should go through.
7. If no relationship has been configured or discovered between a SIP account and an XMPP account, and the client is aware of multiple telephony-capable accounts, present the user with the choice of reaching the contact through any of those accounts.
8. If known, indicate the existence of a special relationship between the SIP and XMPP accounts of a CUSAX service.
9. Optionally, present the XMPP connection as an "instant messaging" or a "chat" account and the SIP connection as a "Voice and Video" or a "Telephony" account.

10. Optionally, determine the identity of the audio/video caller and a corresponding XMPP roster entry so that the user could use textual chatting or other forms of communication that require XMPP.
11. Optionally, delay the XMPP connection until after a SIP connection has been successfully registered.
12. Optionally, check for alternative communication methods (SIP addresses advertised over XMPP, and XMPP addresses advertised over SIP).

The following strategies are suggested for CUSAX services:

1. Use online provisioning and configuration of accounts so that users won't need to setup two separate accounts for the CUSAX service.
2. Use online provisioning so that calling features are disabled for all XMPP accounts.
3. Ensure that at least one of the vCard "tel" fields for each XMPP user is properly populated with a SIP URI that is reachable through the SIP service.
4. Optionally, include the "video" tel type for accounts that are capable of handling video communication.
5. Optionally, provision clients with information indicating that specific SIP and XMPP accounts are related in a CUSAX service.
6. Optionally, attach a "Call-Info" header with an "impp" purpose to all SIP INVITE messages, so that clients can more rapidly associate a caller with a roster entry and display a "Caller ID".

7. IANA Considerations

This document has no actions for the IANA.

8. Security Considerations

Use of the same user agent with two different accounts providing complementary features introduces the possibility of mismatches between the security profiles of those accounts or features. Two security mismatches of particular concern are:

- o The SIP aspect and XMPP aspect of a CUSAX service might offer different authentication options (e.g., digest authentication for

SIP as specified in [RFC3261] and SCRAM authentication [RFC5802] for XMPP as specified in [RFC6120]). Because SIP uses a password-based method (digest) and XMPP uses a pluggable framework for authentication via the Simple Authentication and Security Layer (SASL) technology [RFC4422], it is also possible that the XMPP connection could be authenticated using a password-free method such as client certificates with SASL EXTERNAL even though a username and password is used for the SIP connection.

- o The Transport Layer Security (TLS) [RFC5246] ciphersuites offered or negotiated on the XMPP side might be different from those on the SIP side because of implementation or configuration differences between the SIP server and the XMPP server. Even more seriously, a CUSAX client might successfully negotiate TLS when connecting to the XMPP aspect of the service but not when connecting to the SIP aspect, or vice-versa. In this situation an end user might think that the combined CUSAX session with the service is protected by TLS, even though only one aspect is protected.

Security mismatches such as these (as well as others related to end-to-end encryption of messages or media) introduce the possibility of downgrade attacks, eavesdropping, information leakage, and other security vulnerabilities. User agent developers and service providers must ensure that such mismatches are avoided as much as possible (e.g., by enforcing common and strong security configurations and policies across protocols). Specifically, if both protocols are not safeguarded by similar levels of cryptographic protection, the user must be informed of that fact and given the opportunity to bring both up to the same level.

Section 5 discusses potential issues that may arise due to a mismatch between client capabilities, such as calls being initiated with costs that are not expected by the end user. Such issues could be triggered maliciously, as well as by accident. Implementers therefore need to provide necessary cues to raise user awareness as suggested in Section 5.

Refer to the specifications for the relevant SIP and XMPP features for detailed security considerations applying to each "stack" in a CUSAX client.

9. References

9.1. Normative References

- [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.

[RFC6120] Saint-Andre, P., "Extensible Messaging and Presence Protocol (XMPP): Core", RFC 6120, March 2011.

[RFC6121] Saint-Andre, P., "Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence", RFC 6121, March 2011.

9.2. Informative References

[I-D.ivov-groupchat-purpose]

Ivov, E., "A Group Text Chat Purpose for Conference and Service URIs in the Session Initiation Protocol (SIP) Event Package for Conference State ", draft-ivov-groupchat-purpose-03 (work in progress), June 2013.

[RFC2026] Bradner, S., "The Internet Standards Process -- Revision 3", BCP 9, RFC 2026, October 1996.

[RFC4353] Rosenberg, J., "A Framework for Conferencing with the Session Initiation Protocol (SIP)", RFC 4353, February 2006.

[RFC4422] Melnikov, A. and K. Zeilenga, "Simple Authentication and Security Layer (SASL)", RFC 4422, June 2006.

[RFC4575] Rosenberg, J., Schulzrinne, H., and O. Levin, "A Session Initiation Protocol (SIP) Event Package for Conference State", RFC 4575, August 2006.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.

[RFC5802] Newman, C., Menon-Sen, A., Melnikov, A., and N. Williams, "Salted Challenge Response Authentication Mechanism (SCRAM) SASL and GSS-API Mechanisms", RFC 5802, July 2010.

[RFC6350] Perreault, S., "vCard Format Specification", RFC 6350, August 2011.

[RFC6914] Rosenberg, J., "SIMPLE Made Simple: An Overview of the IETF Specifications for Instant Messaging and Presence Using the Session Initiation Protocol (SIP)", RFC 6914, April 2013.

- [RFC6993] Saint-Andre, P., "Instant Messaging and Presence Purpose for the Call-Info Header Field in the Session Initiation Protocol (SIP)", RFC 6993, July 2013.
- [RFC7033] Jones, P., Salgueiro, G., Jones, M., and J. Smarr, "WebFinger", RFC 7033, September 2013.
- [XEP-0004] Eatmon, R., Hildebrand, J., Miller, J., Muldowney, T., and P. Saint-Andre, "Data Forms", XSF XEP 0004, August 2007.
- [XEP-0030] Hildebrand, J., Millard, P., Eatmon, R., and P. Saint-Andre, "Service Discovery", XSF XEP 0030, June 2008.
- [XEP-0045] Saint-Andre, P., "Multi-User Chat", XSF XEP 0045, February 2012.
- [XEP-0054] Saint-Andre, P., "vcard-temp", XSF XEP 0054, July 2008.
- [XEP-0128] Saint-Andre, P., "Service Discovery Extensions", XSF XEP 0128, October 2004.
- [XEP-0152] Hildebrand, J. and P. Saint-Andre, "XEP-0152: Reachability Addresses", XEP XEP-0152, September 2013.
- [XEP-0166] Ludwig, S., Beda, J., Saint-Andre, P., McQueen, R., Egan, S., and J. Hildebrand, "Jingle", XSF XEP 0166, December 2009.
- [XEP-0167] Ludwig, S., Saint-Andre, P., Egan, S., McQueen, R., and D. Cionoiu, "Jingle RTP Sessions", XSF XEP 0167, December 2009.
- [XEP-0292] Saint-Andre, P. and S. Mizzi, "vCard4 Over XMPP", XSF XEP 0292, September 2013.

Appendix A. Acknowledgements

This draft is inspired by the "SIXPAC" work of Markus Isomaki and Simo Veikkolainen. Markus also provided various suggestions for improving the document.

The authors would also like to thank the following people for their reviews and suggestions: Sebastien Couture, Dan-Christian Bogos, Richard Brady, Olivier Crete, Aaron Evans, Kevin Gallagher, Adrian Georgescu, Saul Ibarra Corretge, David Laban, Gergely Lukacsy, Murray Mar, Daniel Pocock, Travis Reitter, and Gonzalo Salgueiro.

Brian Carpenter, Ted Hardie, Paul Hoffman, and Benson Schliesser reviewed the document on behalf of the General Area Review Team, the Applications Area Directorate, the Security Directorate, and the Operations and Management Directorate, respectively.

Benoit Claise, Barry Leiba, and Pete Resnick provided helpful and substantive feedback during IESG review.

The document shepherd was Mary Barnes. The sponsoring Area Director was Gonzalo Camarillo.

Authors' Addresses

Emil Ivov
Jitsi
Strasbourg 67000
France

Phone: +33-177-624-330
Email: emcho@jitsi.org

Peter Saint-Andre
Cisco Systems, Inc.
1899 Wynkoop Street, Suite 600
Denver, CO 80202
USA

Phone: +1-303-308-3282
Email: psaintan@cisco.com

Enrico Marocco
Telecom Italia
Via G. Reiss Romoli, 274
Turin 10148
Italy

Email: enrico.marocco@telecomitalia.it

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 25, 2013

J. Peterson
NeuStar, Inc.
October 22, 2012

A Framework and Data Model for Queries about Telephone-Related Queries
(TeRQ)
draft-peterson-terq-02

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 April 25, 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
4.1. Bindings	8
4.2. Encodings	9
4.3. Profiles	9
5. The Data Model	11
5.1. Source	11
5.1.1. Query Source	11
5.1.2. Query Intermediary	11
5.1.3. Route Source	12
5.2. Subject	12
5.2.1. Telephone Number	12
5.2.2. Service Provider Identifier	13
5.3. Attributes	13
5.4. Records	13
5.4.1. Attributes	14
5.4.2. Authority	14
5.4.3. Priority	14
5.4.4. Expiration	14
5.5. Response Code	14
6. Element Types	15
6.1. Telephone Number Type	15
6.1.1. TN Range Type	15
6.2. Domain Name Type	15
6.3. Uniform Resource Indicator (URI) Type	15
6.4. Internet Protocol (IP) Address Type	15
6.5. Service Provider Identifier (SPID) Type	15
6.6. Trunk Group Type	16
6.7. Display Name Type	16
6.8. Expiry Type	16
6.9. Priority Type	16
6.10. Extension Type	16
7. Attributes	17
7.1. Routing Attributes	17
7.2. Administrative Attributes	17
8. Security Considerations	18
9. IANA Considerations	19
10. Acknowledgements	20
11. 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 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 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.

4.1. 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. 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.

4.2. 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.

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.

4.3. 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.

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: 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.

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 format of the SPID. Types defined by this specification include: SPID and domain name.

5.3. Attributes

Attributes in this data model are all specified as having 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 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.

5.4. 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. Element Types

6.1. 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.]

6.1.1. TN Range Type

The TN range type consists of a set of two telephone number types, and semantically includes all numbers between those two numbers.

Type Code: R

6.2. 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

6.3. 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

6.4. 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

6.5. Service Provider Identifier (SPID) Type

The SPID type consists of a four-digit number.

Type Code: S

6.6. Trunk Group Type

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

Type Code: G

6.7. Display Name Type

The display name type conforms to the "display-name" ABNF given in RFC3261.

Type Code: N

6.8. Expiry Type

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

Type Code: E

6.9. Priority Type

The Priority type contains an integer between 0 and 1.

Type Code: P

6.10. Extension Type

This code is reserved for future use.

Type Code: X

7. 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.

7.1. Routing Attributes

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

7.2. Administrative Attributes

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

8. Security Considerations

[TBD]

9. 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 data 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 data 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]

10. Acknowledgements

11. 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

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