Network Working Group
Internet-Draft
Intended status: Informational
Expires: January 8, 2017

J. Peterson
T. McGarry
NeuStar, Inc.

July 7, 2016

Modern Problem Statement, Use Cases, and Framework
draft-ietf-modern-problem-framework-01.txt

Abstract

The functions of the public switched telephone network (PSTN) are rapidly migrating to the Internet. This is generating new requirements for many traditional elements of the PSTN, including telephone numbers (TNs). TNs no longer serve simply as telephone routing addresses, they are now identifiers which may be used by Internet-based services for a variety of purposes including session establishment, identity verification, and service enablement. This problem statement examines how the existing tools for allocating and managing telephone numbers do not align with the use cases of the Internet environment, and proposes a framework for Internet-based services relying on TNs.

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 8, 2017.

Copyright Notice

Copyright (c) 2016 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. Problem Statement 3
2. Definitions 4
2.1. Actors 5
2.2. Data Types 7
2.3. Data Management Architectures 8
3. Framework 9
4. Use Cases 10
4.1. Acquisition 11
4.1.1. CSP Acquires TNs from Registrar 11
4.1.2. User Acquires TNs from CSP 12
4.1.3. CSP Delegates TNs to Another CSP 12
4.1.4. User Acquires TNs from a Delegate 13
4.1.5. User Acquires Numbers from Registrar 13
4.2. Management 13
4.2.1. Management of Administrative Data 13
4.2.1.1. CSP to Registrar 14
4.2.1.2. User to CSP 14
4.2.1.3. User to Registrar 15
4.2.2. Management of Service Data 15
4.2.2.1. CSP to other CSPs 15
4.2.2.2. User to CSP 16
4.2.3. Managing Change 16
4.2.3.1. Changing the CSP for an Existing Communications Service 16
4.2.3.2. Terminating a Service 16
4.3. Retrieval 17
4.3.1. Retrieval of Public Data 17
4.3.2. Retrieval of Semi-restricted Administrative Data . . 18
4.3.3. Retrieval of Semi-restricted Service Data 18
4.3.4. Retrieval of Restricted Data 18
5. Acknowledgments 19
6. IANA Considerations 19
7. Security Considerations 19
8. Informative References 20
Authors' Addresses 22

1. Problem Statement

The challenges of utilizing telephone numbers (TNs) on the Internet have been known for some time. Internet telephony provided the first use case for routing telephone numbers on the Internet in a manner similar to how calls are routed in the public switched telephone network (PSTN). As the Internet had no service for discovering the endpoints associated with telephone numbers, ENUM [3] created a DNS-based mechanism for resolving TNs in an IP environment, by defining procedures for translating TNs into URIs for use by protocols such as SIP [2]. The resulting database was designed to function in a manner similar to the systems that route calls in the PSTN. Originally, it was envisioned that ENUM would be deployed as a global hierarchical service, though in practice, it has only been deployed piecemeal by various parties. Most notably, ENUM is used as an internal network function, and is hardly used between service provider networks. The original ENUM concept of a single root, e164.arpa, proved to be politically and practically challenging, and less centralized models have thus flourished. Subsequently, the DRINKS [4] framework showed ways that authorities might provision information about TNs at an ENUM service or similar Internet-based directory. These technologies have also generally tried to preserve the features and architecture familiar to the PSTN numbering environment.

Over time, Internet telephony has encompassed functions that differ substantially from traditional PSTN routing and management, especially as non-traditional providers have begun to utilize numbering resources. An increasing number of enterprises, over-the-top Voice over IP providers, text messaging services, and related non-carrier services have become heavy users of telephone numbers. An enterprise, for example, could deploy an IP PBX that receives a block of telephone numbers from a carrier and then in turn distribute those numbers to new IP telephones when they associate with the PBX. Internet services offer users portals where they can allocate new telephone numbers on the fly, assign multiple "alias" telephone numbers to a single line service, implement various mobility or find-me-follow-me applications, and so on. Peer-to-peer telephone networks have encouraged experiments with distributed databases for telephone number routing and even allocation.

This dynamic control over telephone numbers has few precedents in the traditional PSTN outside of number portability. Number portability has been implemented in many countries, and the capability of a user to choose and change their service provider while retaining their TN is widely implemented now. However, TN administration processes rooted in PSTN technology and policies dictate that this be an exception process fraught with problems and delays. Originally, processes were built to associate a specific TN to a specific service

provider and never change it. With number portability, the industry had to build new infrastructure, new administrative functions and processes to change the association of the TN from one service provider to another. Thanks to the increasing sophistication of consumer mobile devices as Internet endpoints as well as telephones, users now associate TNs with many Internet applications other than telephony. This has generated new interest in models similar to those in place for administering freephone services in the United States, where a user purchases a number through a sort of number registrar and controls its administration (such as routing) on their own, typically using Internet services to directly make changes to the service associated with telephone numbers.

Most TNs today are assigned to specific geographies, at both an international level and within national numbering plans. Numbering practices today are tightly coupled with the manner that service providers interconnect, as well as how TNs are routed and administered: the PSTN was carefully designed to delegate switching intelligence geographically. In interexchange carrier routing in North America, for example, calls to a particular TN are often handed off to the terminating service provider close to the geography where that TN is assigned. But the overwhelming success of mobile telephones has increasing eroded the connection between numbers and regions. Furthermore, the topology of IP networks is not anchored to geography in the same way that the telephone network is. In an Internet environment, establishing a network architecture for routing TNs could depend little on geography. Adapting TNs to the Internet requires more security, richer datasets and more complex query and response capabilities than previous efforts have provided.

This document will create a common understanding of the problem statement related to allocating, managing, and resolving TNs in an IP environment. It outlines a framework and lists motivating use cases for creating IP-based mechanisms for TNs. It is important to acknowledge at the outset that there are various evolving international and national policies and processes related to TNs, and any solutions need to be flexible enough to account for variations in policy and requirements.

2. Definitions

This section provides definitions for actors, data types and data management architectures as they are discussed in this document. Different numbering spaces may instantiate these roles and concepts differently: practices that apply to non-geographic freephone numbers, for example, may not apply to geographic numbers, and practices that exist under one Numbering Authority may not be permitted under another. The purpose of this framework is to

identify the characteristics of protocol tools that will satisfy the diverse requirements for telephone number acquisition, management, and retrieval on the Internet.

2.1. Actors

The following roles of actors are defined in this document:

Numbering Authority: A regulatory body within a country that manages that country's TNs. The Numbering Authority decides national numbering policy for the nation, region, or other domain for which it has authority, including what TNs can be allocated, and which are reserved.

Registry: An entity that administers the allocation of TNs based on a Numbering Authority's policies. Numbering authorities can act as the Registries themselves, or they can outsource the function to other entities. There are two subtypes of Registries: an Authoritative Registry and a Distributed Registry. The general term Registry in this document refers to both kinds of Registries.

Authoritative Registry: An authoritative Registry is a single entity with sole responsibility for specific numbering resources.

Distributed Registry: Distributed Registries are multiple Registries responsible for the same numbering resources.

Registrar: An entity that distributes the telephone numbers administered by a Registry; typically, there are many Registrars that can distribute numbers from a single Registry, through Registrars may serve multiple Registries as well. A Registrar has business relationships with its assignees and collects administrative information from them.

Communication Service Provider (CSP): A provider of communications services, where those services can be identified by TNs. This includes both traditional telephone carriers or enterprises as well as service providers with no presence on the PSTN who use TNs. This framework does not assume that any single CSP provides all the communications service related to a particular TN.

Service Enabler: An entity that works with CSPs to enable communication service to a User; perhaps a vendor, or third-party integrator.

User: An individual reachable through a communications service; usually a customer of a communication service provider.

Government Entity: An entity that, due to legal powers deriving from national policy, has privileged access to information about number administration under certain conditions.

Note that an individual, company or other entity may act in one or more of the roles above; for example, a company may be a CSP and also a Registrar. Although Numbering Authorities are listed as actors, they are unlikely to actually participate in the protocol flows themselves, though in some situations a Numbering Authority and Registry may be the same administrative entity.

All actors that are recipients of numbering resources, be they a CSP, Service Enabler, or User, can also be said to have a relationship to a Registry of either an assignee or delegate:

Assignee: An actor that is assigned a TN directly by a Registrar; an assignee always has a direct relationship with a Registrar.

Delegate: An actor that is delegated a TN from an assignee or another delegate, who does not necessarily have a direct relationship with a Registrar. Delegates may delegate one or more of their TN assignment(s) to one or more further downstream subdelegates.

As an example, consider a case where a Numbering Authority also acts as a Registry, and it issues 10,000 blocks of TNs to CSPs, which in this case also act as Registrars. CSP/Registrars would then be responsible for distributing numbering resources to Users and other CSPs. In this case, an enterprise deploying IP PBXs also acts as a CSP, and it acquires number blocks for its enterprise seats in chunks of 100 from a CSP acting as a Registrar with whom the enterprise has a business relationship. The enterprise is in this case the assignee, as it receives numbering resources directly from a Registrar. As it doles out individual numbers to its Users, the enterprise delegates its own numbering resources to those Users and their communications endpoints. The overall ecosystem might look as follows.

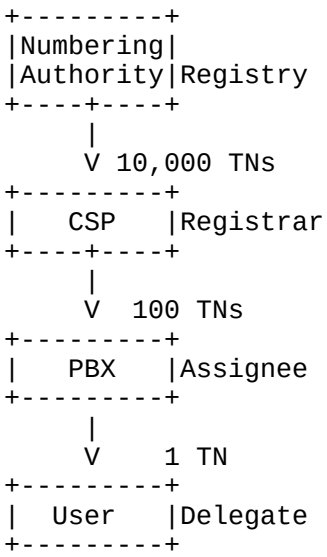


Figure 1: Chain of Number Assignment

2.2. Data Types

The following data types are defined in this document:

Administrative Data: assignment data related to the TN and the relevant actors; it includes TN status (assigned, unassigned, etc.), contact data for the assignee or delegate, and typically does not require real-time performance as access to this data is not required for ordinary call or session establishment.

Service Data: data necessary to enable service for the TN; it includes addressing data, service features, and so on, and typically does require real-time performance, in so far as this data typically must be queried during call set-up.

Administrative and service data can fit into three categories:

Public: data that anyone can access, for example a list of which numbering resources (unallocated number ranges) are available for acquisition from the Registry.

Semi-restricted: data that a subset of actors can access, for example CSPs may be able to access other CSP’s service data.

Restricted: data that is only available to a small subset of actors, for example a Government Entity may be able access contact information for a User.

While it might appear there are really only two categories, public and restricted based on requestor, the distinction between semi-restricted and restricted is helpful for the use cases below.

2.3. Data Management Architectures

This framework generally assumes that administrative and service data is maintained by CSPs, Registrars, and Registries. The role of a Registry described here is a "thin" one, where the Registry manages basic allocation information for the numbering space, such as information about whether or not the number is assigned, and if assigned, by which Registrar. It is the Registrar that in turn manages detailed administrative data about those assignments, such as contact or billing information for the assignee. In some models, CSPs and Registrars will be composed (the same administrative entity), and in others the Registry and Registrar may similarly be composed. Typically, service data resides largely at the CSP itself, though in some models a "thicker" Registry may itself contain a pointer to the servicing CSP for a number or number block. In addition to traditional centralized Registries, this framework also supports environments where the same data is being managed by multiple administrative entities, and stored in many locations. A distribute registry system is discussed further in [16].

Data store: a service that stores and enables access to administrative and/or service data.

Reference Address: a URL that dereferences to the location of the data store.

Distributed data stores: refers to administrative or service data being stored with multiple actors. For example, CSPs could provision their service data to multiple other CSPs.

Distributed Registries: refers to multiple Registries managing the same numbering resource. Actors could interact with one or multiple Registries. The Registries would update each other when change occurs. The challenge is to ensure there are no clashes, e.g., two Registries assigning the same TN to two different actors.

3. Framework

The framework outlined in this document requires three Internet-based mechanisms for managing and resolving telephone numbers (TNs) in an IP environment. These mechanisms will likely reuse existing protocols for sharing structured data; it is unlikely that new protocol development work will be required, though new information models specific to the data itself will be a major focus of framework development. Likely candidates for reuse here include work done in DRINKS [4] and WEIRDS [12], as well as the TeRI [13] framework.

These protocol mechanisms are scoped in a way that makes them likely to apply to a broad range of future policies for number administration. It is not the purpose of this framework to dictate number policy, but instead to provide tools that will work with policies as they evolve going forward. These mechanisms therefore do not assume that number administration is centralized, nor that number allocations are restricted to any category of service providers, though these tools must and will work in environments with those properties.

The three mechanisms are:

Acquisition: a protocol mechanism for acquiring TNs, including an enrollment process.

Management: a protocol mechanism for associating data with TNs.

Retrieval: a protocol mechanism for retrieving data about TNs.

The acquisition mechanism will enable actors to acquire TNs for use with a communications service. The acquisition mechanism will provide a means to request numbering resources from a service operated by a Registrar, CSP or similar actor. TNs may be requested either on a number-by-number basis, or as inventory blocks. Any actor who grants numbering resources will retain metadata about the assignment, including the responsible organization or individual to whom numbers have been assigned.

The management mechanism will let actors provision data associated with TNs. For example, if a User has been assigned a TN, they may select a CSP to provide a particular service associated with the TN, or a CSP may assign a TN to a User upon service activation. In either case, a mechanism is needed to provision data associated with the TN at that CSP.

The retrieval mechanism will enable actors to learn information about TNs, typically by sending a request to a CSP. For some information,

an actor may need to send a request to a Registry rather than a CSP. Different parties may be authorized to receive different information about TNs.

As an example, a CSP might use the acquisition interface to acquire a chunk of numbers from a Registrar. Users might then provision administrative data associated with those numbers at the CSP through the management interface, and query for service data relating to those numbers through the retrieval interface of the CSP.

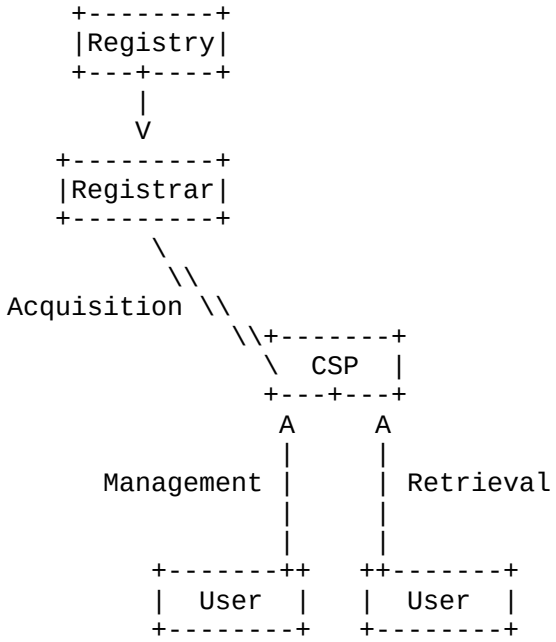


Figure 2: Example of the Three Interfaces

4. Use Cases

The high-level use cases in this section will provide an overview of the expected operation of the three interfaces in the MODERN problem space.

4.1. Acquisition

There are various scenarios for how TNs can be acquired by the relevant actors: a CSP, Service Enabler, or User. There are three actors from which numbers can be acquired: a Registrar, a CSP and a User (presumably one who is delegating to another party). It is assumed that Registrars are either composed with Registries, or that Registrars have established business relationships with Registries that enable them to distribute the numbers that the Registries here administer. In these use cases, a User may acquire TNs either from a CSP or a Registry, or from an intermediate delegate.

4.1.1. CSP Acquires TNs from Registrar

The most fundamental and traditional numbering use case is one where a CSP, such as a carrier, requests a block of numbers from a Registrar to hold as inventory or assign to customers.

Through some out-of-band business process, a CSP develops a relationship with a Registrar. The Registrar maintains a profile of the CSP and what qualifications they possess for requesting TNs. The CSP may then request TNs from within a specific pool of numbers in the authority of the Registry; such as region, mobile, wireline, tollfree, etc. The Registrar must authenticate and authorize the CSP, and then either grant or deny a request. When an assignment occurs, the Registry creates and stores administrative information related to the assignment such as TN status and Registrar contact information, and removes the specific TN(s) from the pool of those that are available for assignment. As a part of the acquisition and assignment process, the Registry provides any necessary credentials (for example, STIR certificates [14]) to the Registrar to be used to prove the assignment for future transactions.

Before it is eligible to receive TN assignments, per the policy of a national authority, the CSP may need to have submitted (again, through some out-of-band process) additional qualifying information such as current utilization rate or a demand forecast.

There are two scenarios under which a CSP requests resources; they are requesting inventory, or they are requesting for a specific User or delegate. TNs assigned to a User are always considered assigned, not inventory. The CSP will associate service information for that TN, e.g., service address, and make it available to other CSPs to enable interoperability. The CSP may need to update the Registrar regarding this service activation (this is part of the "TN status" maintained by the Registrar).

4.1.2. User Acquires TNs from CSP

Today, a User typically acquires a TN from CSP when signing up for communications service or turning on a new device. In this use case, the User becomes the delegate of the CSP.

A User creates or has a relationship with the CSP, and subscribes to a communications service which includes the use of a TN. The CSP collects and stores administrative data about the User. The CSP then activates the User on their network and creates any necessary service data to enable interoperability with other CSPs. The CSP could also update public or privileged databases accessible by other Actors. The CSP provides any necessary credentials to the User (for example, a STIR certificate [14]) to prove the assignment for future transactions. Such credential could be delegated from the one provided by the Registrar to the CSP to continue the chain of assignment.

The CSP could assign a TN from its existing inventory or it could acquire a new TN from the Registrar as part of the assignment process. If it assigns it from its existing inventory it would remove the specific TN from the pool of those available for assignment. It may also update the Registrar about the assignment so the Registrar has current assignment data.

4.1.3. CSP Delegates TNs to Another CSP

A reseller or a service bureau might acquire a block of numbers from a CSP to be issued to Users.

In this case, the delegate CSP has a business relationship with the assignee CSP. The assignee CSP collects and stores administrative data about the delegate. The assignee then activates the delegate on their network and creates any necessary service data to enable interoperability with other CSPs. The CSP could also update public or privileged databases accessible by other Actors. The CSP provides any necessary credentials to the delegate CSP (for example, a STIR certificate [14]) to prove the assignment for future transactions. Such credentials could be delegated from the one provided by the Registry to the CSP to continue the chain of assignment.

The CSP could assign a block from its existing inventory or it could acquire new TNs from the Registrar as part of the assignment process. If it assigns it from its existing inventory it would remove the specific TN from the pool of those available for assignment. It may also update the Registrar about the assignment so the Registrar has current assignment data. The Delegate may need to provide

utilization and assignment data to the Registry, either directly or through the CSP.

4.1.4. User Acquires TNs from a Delegate

Acquiring a TN from a delegate follows the process in Section 4.1.2, as it should be similar to how a User acquires TNs from a CSP. In this case, the delegate re-delegating the TNs would be performing functions done by the CSP, e.g., providing any credentials, collecting administrative data, creative service data, and so on.

4.1.5. User Acquires Numbers from Registrar

Today, a user wishing to acquire a freephone number may browse the existing inventory through one or more Registrars, comparing their prices and services. Each such Registrar either is a CSP, or has a business relationship with a CSP to provide services for that freephone number.

Acquiring a TN from a Registrar follows the process in Section 4.1.1, as it should be similar to how a CSP acquires TNs from a Registrar. In this case, the User must establish some business relationship directly to a Registrar, similarly to how such functions are conducted today when Users purchase domain names. For the purpose of status information kept by the Registry, TNs assigned to a User are always considered assigned, not inventory.

In this use case, after receiving a number assignment from the Registrar, a User will then obtain communications service from a CSP, and provide to the CSP the TN to be used for that service. The CSP will associate service information for that TN, e.g., service address, and make it available to other CSPs to enable interoperability.

4.2. Management

The management protocol mechanism is needed to associate administrative and service data with TNs, and may be used to refresh or rollover associated credentials.

4.2.1. Management of Administrative Data

Administrative data is primarily related to the status of the TN, its administrative contacts, and the actors involved in providing service to the TN. Protocol interactions for administrative data will therefore predominantly occur between CSPs and Users to the Registrar, or between Users and delegate CSPs to the CSP.

Most administrative data is not a good candidate for a distributed data store model. Access to it does not require real-time performance therefore local caches are not necessary. And it will include sensitive information such as user and contact data.

Some of the data could lend itself to being publicly available, such as CSP and TN assignment status. In that case it would be deemed public information for the purposes of the retrieval interface.

4.2.1.1. CSP to Registrar

After a CSP acquires a TN or block of TNs from the Registrar (per Section 4.1.1 above), it then provides administrative data to the Registrar as a step in the acquisition process. The Registrar will authenticate the CSP and determine if the CSP is authorized to provision the administrative data for the TNs in question. The Registry will update the status of the TN, i.e., that it is unavailable for assignment. The Registrar will also maintain administrative data provided by the CSP.

Changes to this administrative data will not be frequent. Examples of changes would be terminating service (see Section 4.2.3.2) and changing a CSP or delegate. Changes should be authenticated by a credential to prove administrative responsibility for the TN.

In a distributed Registry model, TN status, e.g., allocated, assigned, available, unavailable, would need to be provided to other Registries in real-time. Other administrative data could be sent to all Registries or other Registries could get a reference address to the host Registry’s data store.

4.2.1.2. User to CSP

After a User acquires a TN or block of TNs from a CSP, the User will provide administrative data to the CSP. The CSP commonly acts as a Registrar in this case, maintaining the administrative data and only notify the Registry of the change in TN status. In this case, the Registry maintains a reference address to the CSP/Registrar’s administrative data store so relevant actors have the ability to access the data. Alternatively a CSP could send the administrative data to an external Registrar to store. If there is a delegate between the CSP and user, they will have to ensure there is a mechanism for the delegate to update the CSP as change occurs.

4.2.1.3. User to Registrar

If the User has a direct relationship with the Registrar, then naturally the user could provision administrative data associated with their TN directly to the Registrar. This is the case, for example, with the freephone example, where a User has a business relationship with its freephone provider, and the freephone provider maintains account and billing data. While delegates necessarily are not assignees, some environments as an optimization might want to support a model where the delegate updates the Registrar directly on changes, as opposed to sending that data to the CSP or through the CSP to the Registrar. As stated already, the protocol should enable Users to acquire TNs directly from a Registrar, which Registrar may or may not also act as a CSP. In these cases the updates would be similar to that described in Section 4.2.1.1.

4.2.2. Management of Service Data

Service data is data required by an originating or intermediate CSP to enable communications service to a User: a SIP URI is an example of one service data element commonly used to route communications. CSPs typically create and manage service data, however it is possible that delegates and Users could as well. For most use cases involving individual Users, it is anticipated that lower-level service information changes would be communicated to CSPs via existing protocols (like the baseline SIP REGISTER [2] method) rather than through any new interfaces defined by MODERN.

4.2.2.1. CSP to other CSPs

After a User enrolls for service with a CSP, in the case where the CSP was assigned the TN by a Registrar, the CSP will then create a service address (such as a SIP URI) and associate it with the TN. The CSP needs to update this data to enable service interoperability. There are multiple ways that this update can occur, though most commonly service data is exposed through the retrieval interface (see Section 4.3. For certain deployment architectures, like a distributed data store model, CSPs may need to provide data directly to other CSPs.

If the CSP is assigning a TN from its own inventory it may not need to perform service data updates as change occurs because the existing service data associated with inventory may be sufficient once the TN is put in service. They would however likely update the Registry on the change in status.

4.2.2.2. User to CSP

Users could also associate service data to their TNs at the CSP. An example is a User acquires a TN from the Registrar (as described in Section 4.1.5) and wants to provide that TN to the CSP so the CSP can enable service. In this case, once the user provides the number to the CSP, the CSP would update the Registry or other actors as outlined in Section 4.2.2.1.

4.2.3. Managing Change

This section will address some special use cases that were not covered in other sections of 4.2.

4.2.3.1. Changing the CSP for an Existing Communications Service

A User who subscribes to a communications service, and received their TN from that CSP, wishes to retain the same TN but move their service to a different CSP. The User provides their credential to the new CSP and the CSP initiates the change in service.

In the simplest scenario, where there’s an authoritative composed Registry/Registrar that maintains service data, the new CSP provides the new service data with the User’s credential to the Registry/ Registrar, which then makes the change. The old credential is revoked and a new one is provided. The new CSP or the Registrar would send a notification to the old CSP, so they can disable service. The old CSP will undo any delegations to the User, including invalidating any cryptographic credentials (e.g. STIR certificates [13]) previously granted to the User. Any service data maintained by the CSP must be removed, and similarly, the CSP must delete any such information it provisioned in the Registry.

In a similar model to common practice in some environments today, the User could provide their credential to the old CSP, and the old CSP initiates the change in service.

If there was a distributed Registry that maintained service data, the Registry would also have to update the other Registries of the change.

4.2.3.2. Terminating a Service

A User who subscribes to a communications service, and received their TN from the CSP, wishes to terminate their service. At this time, the CSP will undo any delegations to the User, including invalidating any cryptographic credentials (e.g. STIR certificates [13]) previously granted to the User. Any service data maintained by the

CSP must be removed, and similarly, the CSP must delete any such information it provisioned in the Registrar.

The TN will change state from assigned to unassigned, the CSP will update the Registry. Depending on policies the TN could go back into the Registry, CSP, or delegate's pool of available TNs and would likely enter an ageing process.

In an alternative use case, a User who received their own TN assignment directly from a Registrar terminates their service with a CSP. At this time, the User might terminate their assignment from the Registrar, and return the TN to the Registry for re-assignment. Alternatively, they could retain the TN and elect to assign it to some other service at a later time.

4.3. Retrieval

Retrieval of administrative or service data will be subject to access restrictions based on the category of the specific data; public, semi-restricted or restricted. Both administrative and service data can have data elements that fall into each of these categories. It is expected that the majority of administrative and service data will fall into the semi-restricted category: access to this information may require some form of authorization, though service data crucial to reachability will need to be accessible. In some environments, it's possible that none of the service data will be considered public.

The retrieval protocol mechanism for semi-restricted and restricted data needs a way for the receiver of the request to identify the originator of the request and what is being requested. The receiver of the request will process that request based on this information.

4.3.1. Retrieval of Public Data

Under most circumstances, a CSP wants its communications service to be publicly reachable through TNs, so the retrieval interface supports public interfaces that permit clients to query for service data about a TN. Some service data may however require that the client by authorized to receive it, per the use case in Section 4.3.3 below.

Public data can simply be posted on websites or made available through a publicly available API. Public data hosted by a CSP may have a reference address at the Registry.

4.3.2. Retrieval of Semi-restricted Administrative Data

A CSP is having service problems completing calls to a specific TN, so it wants to contact the CSP serving that TN. The Registry authorizes the originating CSP to access this information. It initiates a query to the Registry, the Registry verifies the requestor and the requested data and Registry responds with the serving CSP and contact data.

Alternatively that information could be part of a distributed data store and not stored at the Registry. In that case, the CSP has the data in a local distributed data store and it initiates the query to the local data store. The local data store responds with the CSP and contact data. No verification is necessary because it was done when the CSP was authorized to receive the data store.

4.3.3. Retrieval of Semi-restricted Service Data

A User on a CSP’s network calls a TN. The CSP initiates a query for service data associated with the TN to complete the call, and will receive special service data because the CSP operates in a closed environment where different CSPs receive different responses, and only authorized CSPs may access service data. The query and response must have real-time performance. There are multiple scenarios for the query and response.

In a distributed data store model each CSP distributes its updated service data to all other CSPs. The originating CSP has the service data in its local data store and queries it. The local data store responds with the service data. The service data can be a reference address to a data store maintained by the serving CSP or it can be the service address itself. In the case where it’s a reference address the query would go to the serving CSP and they would verify the requestor and the requested data and respond. In the case where it’s the service address it would process the call using that.

In some environments, aspects of the service data may reside at the Registry itself (for example, the assigned CSP for a TN), and thus a the query may be sent to the Registry. The Registry verifies the requestor and the requested data and responds with the service data, such as a SIP URI containing the domain of the assigned CSP.

4.3.4. Retrieval of Restricted Data

In this case, a Government Entity wishes to access information about a particular User, who subscribes to a communications service. The entity that operates the Registry on behalf of the National Authority in this case has some pre-defined relationship with the Government

Entity. When the CSP acquired TNs from the National Authority, it was a condition of that assignment that the CSP provide access for Government Entities to telephone numbering data when certain conditions apply. The required data may reside either in the CSP or in the Registrar.

For a case where the CSP delegates a number to the User, the CSP might provision the Registrar (or itself, if the CSP is composed with a Registrar) with information relevant to the User. At such a time as the Government Entity needs information about that User, the Government Entity may contact the Registrar or CSP to acquire the necessary data. The interfaces necessary for this will be the same as those described in Section 4.3; the Government Entity will be authenticated, and an authorization decision will be made by the Registrar or CSP under the policy dictates established by the National Authority.

5. Acknowledgments

We would like to thank Henning Schulzrinne for his contributions to this problem statement and framework, and to thank Pierce Gorman for detailed comments.

6. IANA Considerations

This memo includes no instructions for the IANA.

7. Security Considerations

The acquisition, management, and retrieval of administrative and service data associated with telephone numbers raises a number of security issues.

Any mechanism that allows an individual or organization to acquire telephone numbers will require a means of mutual authentication, of integrity protection, and of confidentiality. A Registry as defined in this document will surely want to authenticate the source of an acquisition request as a first step in the authorization process to determine whether or not the resource will be granted. Integrity of both the request and response is essential to ensuring that tampering does not allow attackers to block acquisitions, or worse, to commandeer resources. Confidentiality is essential to preventing eavesdroppers from learning about allocations, including the personally identifying information associated with the administrative or technical contracts for allocations.

A management interface for telephone numbers has similar requirements. Without proper authentication and authorization

mechanisms in place, an attack could use the management interface to disrupt service data or administrative data, which could deny service to users, enable new impersonation attacks, prevent billing systems from operating properly, and cause similar system failures.

Finally, a retrieval interfaces has its own needs for mutual authentication, integrity protection, and for confidentiality. Any CSP sending a request to retrieve service data associated with a number will want to know that it is reaching the proper authority, that the response from that authority has not been tampered with in transit, and in most cases the CSP will not want to reveal to eavesdroppers the number it is requesting or the response that it has received. Similarly, any service answering such a query will want to have a means of authenticating the source of the query, and of protecting the integrity and confidentiality of its responses.

8. Informative References

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

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

[3] Bradner, S., Conroy, L., and K. Fujiwara, "The E.164 to Uniform Resource Identifiers (URI) Dynamic Delegation Discovery System (DDDS) Application (ENUM)", RFC 6116, DOI 10.17487/RFC6116, March 2011, <http://www.rfc-editor.org/info/rfc6116>.

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

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

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

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

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

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

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

[11] Peterson, J., Jennings, C., and R. Sparks, "Change Process for the Session Initiation Protocol (SIP) and the Real-time Applications and Infrastructure Area", BCP 67, RFC 5727, DOI 10.17487/RFC5727, March 2010, <http://www.rfc-editor.org/info/rfc5727>.

[12] Newton, A. and S. Hollenbeck, "Registration Data Access Protocol (RDAP) Query Format", RFC 7482, DOI 10.17487/RFC7482, March 2015, <http://www.rfc-editor.org/info/rfc7482>.

[13] Peterson, J., "A Framework and Information Model for Telephone-Related Information (TeRI)", draft-peterson-modern-teri-00 (work in progress), October 2015.

[14] Peterson, J. and S. Turner, "Secure Telephone Identity Credentials: Certificates", draft-ietf-stir-certificates-06 (work in progress), July 2016.

[15] Barnes, M., Jennings, C., Rosenberg, J., and M. Petit-Huguenin, "Verification Involving PSTN Reachability: Requirements and Architecture Overview", draft-jennings-vipr-overview-06 (work in progress), December 2013.

[16] Bellur, H. and C. Wendt, "Distributed Registry Protocol", draft-wendt-modern-drip-00 (work in progress), October 2015.

[17] Rosenberg, J. and H. Schulzrinne, "Session Initiation Protocol (SIP): Locating SIP Servers", RFC 3263, DOI 10.17487/RFC3263, June 2002, <http://www.rfc-editor.org/info/rfc3263>.

Authors' Addresses

Jon Peterson
Neustar, Inc.
1800 Sutter St Suite 570
Concord, CA 94520
US

Email: jon.peterson@neustar.biz

Tom McGarry
Neustar, Inc.
1800 Sutter St Suite 570
Concord, CA 94520
US

Email: tom.mcgarry@neustar.biz

Network Working Group
Internet-Draft
Intended status: Informational
Expires: September 6, 2018

J. Peterson
T. McGarry
NeuStar, Inc.

March 5, 2018

Modern Problem Statement, Use Cases, and Framework
draft-ietf-modern-problem-framework-04.txt

Abstract

The functions of the public switched telephone network (PSTN) are rapidly migrating to the Internet. This is generating new requirements for many traditional elements of the PSTN, including telephone numbers (TNs). TNs no longer serve simply as telephone routing addresses: they are now identifiers which may be used by Internet-based services for a variety of purposes including session establishment, identity verification, and service enablement. This problem statement examines how the existing tools for allocating and managing telephone numbers do not align with the use cases of the Internet environment, and proposes a framework for Internet-based services relying on TNs.

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 https://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 September 6, 2018.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of

publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Problem Statement 2
2. Definitions 4
2.1. Actors 5
2.2. Data Types 7
2.3. Data Management Architectures 8
3. Framework 9
4. Use Cases 11
4.1. Acquisition 11
4.1.1. Acquiring TNs from Registrar 12
4.1.2. Acquiring TNs from CSPs 13
4.2. Management 14
4.2.1. Management of Administrative Data 14
4.2.1.1. Managing Data at a Registrar 14
4.2.1.2. Managing Data at a CSP 15
4.2.2. Management of Service Data 15
4.2.2.1. CSP to other CSPs 15
4.2.2.2. User to CSP 16
4.2.3. Managing Change 16
4.2.3.1. Changing the CSP for an Existing Service 16
4.2.3.2. Terminating a Service 17
4.3. Retrieval 17
4.3.1. Retrieval of Public Data 18
4.3.2. Retrieval of Semi-restricted Administrative Data . . 18
4.3.3. Retrieval of Semi-restricted Service Data 18
4.3.4. Retrieval of Restricted Data 19
5. Acknowledgments 19
6. IANA Considerations 20
7. Privacy Considerations 20
8. Security Considerations 20
9. Informative References 21
Authors' Addresses 23

1. Problem Statement

The challenges of utilizing telephone numbers (TNs) on the Internet have been known for some time. Internet telephony provided the first use case for routing telephone numbers on the Internet in a manner similar to how calls are routed in the public switched telephone network (PSTN). As the Internet had no service for discovering the

endpoints associated with telephone numbers, ENUM [3] created a DNS-based mechanism for resolving TNs in an IP environment, by defining procedures for translating TNs into URIs for use by protocols such as SIP [2]. The resulting database was designed to function in a manner similar to the systems that route calls in the PSTN. Originally, it was envisioned that ENUM would be deployed as a global hierarchical service, though in practice, it has only been deployed piecemeal by various parties. Most notably, ENUM is used as an internal network function, and is rarely used between service provider networks. The original ENUM concept of a single root, e164.arpa, proved to be politically and practically challenging, and less centralized models have thus flourished. Subsequently, the DRINKS [4] framework showed ways that service providers might provision information about TNs at an ENUM service or similar Internet-based directory. These technologies have also generally tried to preserve the features and architecture familiar to the PSTN numbering environment.

Over time, Internet telephony has encompassed functions that differ substantially from traditional PSTN routing and management, especially as non-traditional providers have begun to utilize numbering resources. An increasing number of enterprises, over-the-top voice-over-IP (VoIP) providers, text messaging services, and related non-carrier services have become heavy users of telephone numbers. An enterprise, for example, can deploy an IP PBX that receives a block of telephone numbers from a carrier and then in turn distribute those numbers to new IP telephones when they associate with the PBX. Internet services offer users portals where they can allocate new telephone numbers on the fly, assign multiple "alias" telephone numbers to a single line service, implement various mobility or find-me-follow-me applications, and so on. Peer-to-peer telephone networks have encouraged experiments with distributed databases for telephone number routing and even allocation.

This dynamic control over telephone numbers has few precedents in the traditional PSTN outside of number portability. Number portability allows the capability of a user to choose and change their service provider while retaining their TN; it has been implemented in many countries; either for all telephony services or for subsets such as mobile. However, TN administration processes rooted in PSTN technology and policies dictate that this be an exception process fraught with problems and delays. Originally, processes were built to associate a specific TN to a specific service provider and never change it. With number portability, the industry had to build new infrastructure, new administrative functions and processes to change the association of the TN from one service provider to another. Thanks to the increasing sophistication of consumer mobile devices as Internet endpoints as well as telephones, users now associate TNs with many Internet applications other than telephony. This has

generated new interest in models similar to those in place for administering freephone (non-geographic toll free numbers) services in the United States, where a user purchases a number through a sort of number registrar and controls its administration (such as routing) on their own, typically using Internet services to directly make changes to the service associated with telephone numbers.

Most TNs today are assigned to specific geographies, at both an international level and within national numbering plans. Numbering practices today are tightly coupled with the manner that service providers interconnect, as well as how TNs are routed and administered: the PSTN was carefully designed to delegate switching intelligence geographically. In interexchange carrier routing in North America, for example, calls to a particular TN are often handed off to the terminating service provider close to the geography where that TN is assigned. But the overwhelming success of mobile telephones has increasingly eroded the connection between numbers and regions. Furthermore, the topology of IP networks is not anchored to geography in the same way that the telephone network is. In an Internet environment, establishing a network architecture for routing TNs could depend little on geography, relying instead on network topologies or other architectural features. Adapting TNs to the Internet requires more security, richer datasets and more complex query and response capabilities than previous efforts have provided.

This document attempts to create a common understanding of the problem statement related to allocating, managing, and resolving TNs in an IP environment, the focus of the IETF MODERN (Managing, Ordering, Distributing, Exposing, and Registering telephone Numbers) working group. It outlines a framework and lists motivating use cases for creating IP-based mechanisms for TNs. It is important to acknowledge at the outset that there are various evolving international and national policies and processes related to TNs, and any solutions need to be flexible enough to account for variations in policy and requirements.

2. Definitions

This section provides definitions for actors, data types and data management architectures as they are discussed in this document. Different numbering spaces may instantiate these roles and concepts differently: practices that apply to non-geographic freephone numbers, for example, may not apply to geographic numbers, and practices that exist under one Numbering Authority may not be permitted under another. The purpose of this framework is to identify the characteristics of protocol tools that will satisfy the diverse requirements for telephone number acquisition, management, and retrieval on the Internet.

2.1. Actors

The following roles of actors are defined in this document:

Numbering Authority: A regulatory body within a region that manages that regions TNs. The Numbering Authority decides national numbering policy for the nation, region, or other domain for which it has authority, including what TNs can be allocated, which are reserved, and which entities may obtain TNs.

Registry: An entity that administers the allocation of TNs based on a Numbering Authority's policies. Numbering authorities can act as the Registries themselves, or they can outsource the function to other entities. Traditional registries are single entities with sole authority and responsibility for specific numbering resources, though distributed registries (see Section 2.3) are also in the scope of this framework.

Credential Authority: An entity that distributes credentials, such as certificates that attest the authority of assignees (defined below) and delegates. This document assumes that one of more credential authorities may be trusted by actors in any given regulatory environment; policies for establishing such trust anchors are outside the scope of this document.

Registrar: An entity that distributes the telephone numbers administered by a Registry; typically, there are many Registrars that can distribute numbers from a single Registry, though Registrars may serve multiple Registries as well. A Registrar has business relationships with number assignees and collects administrative information from them.

Communication Service Provider (CSP): A provider of communications services, where those services can be identified by TNs. This includes both traditional telephone carriers or enterprises as well as service providers with no presence on the PSTN who use TNs. This framework does not assume that any single CSP provides all the communications service related to a particular TN.

Service Enabler: An entity that works with CSPs to enable communication service to a User; perhaps a vendor, a service bureau, or third-party integrator.

User: An individual reachable through a communications service; usually a customer of a communication service provider.

Government Entity: An entity that, due to legal powers deriving from national policy, has privileged access to information about number administration under certain conditions.

Note that an individual, organization, or other entity may act in one or more of the roles above; for example, a company may be a CSP and also a Registrar. Although Numbering Authorities are listed as actors, they are unlikely to actually participate in the protocol flows themselves, though in some situations a Numbering Authority and Registry may be the same administrative entity.

All actors that are recipients of numbering resources, be they a CSP, Service Enabler, or User, can also be said to have a relationship to a Registry of either an assignee or delegate:

Assignee: An actor that is assigned a TN directly by a Registrar; an assignee always has a direct relationship with a Registrar.

Delegate: An actor that is delegated a TN from an assignee or another delegate, who does not necessarily have a direct relationship with a Registrar. Delegates may delegate one or more of their TN assignment(s) to one or more further downstream subdelegates.

As an example, consider a case where a Numbering Authority also acts as a Registry, and it issues blocks of 10,000 TNs to CSPs, which in this case also act as Registrars. CSP/Registrars would then be responsible for distributing numbering resources to Users and other CSPs. In this case, an enterprise deploying IP PBXs also acts as a CSP, and it acquires number blocks for its enterprise seats in chunks of 100 from a CSP acting as a Registrar with whom the enterprise has a business relationship. The enterprise is in this case the assignee, as it receives numbering resources directly from a Registrar. As it doles out individual numbers to its Users, the enterprise delegates its own numbering resources to those Users and their communications endpoints. The overall ecosystem might look as follows.

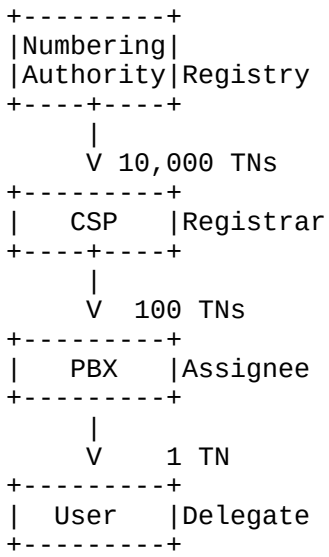


Figure 1: Chain of Number Assignment

2.2. Data Types

The following data types are defined in this document:

Administrative Data: assignment data related to the TN and the relevant actors; it includes TN status (assigned, unassigned, etc.), contact data for the assignee or delegate, and typically does not require real-time access as this data is not required for ordinary call or session establishment.

Service Data: data necessary to enable service for the TN; it includes addressing data and service features. Since this data is necessary to complete calls, it must be obtained in real time.

Administrative and service data can fit into three access categories:

Public: Anyone can access public data. Such data might include a list of which numbering resources (unallocated number ranges) are available for acquisition from the Registry.

Semi-restricted: Only a subset of actors can access semi-restricted data. For example CSPs may be able to access other CSP’s service data in some closed environment.

Restricted: Only a small subset of actors can access restricted data. For example a Government Entity may be able access contact information for a User.

While it might appear there are really only two categories, public and restricted based on requestor, the distinction between semi-restricted and restricted is helpful for the use cases below.

2.3. Data Management Architectures

This framework generally assumes that administrative and service data is maintained by CSPs, Registrars, and Registries. The terms "registrar" and "registry" are familiar from DNS operations, and indeed the DNS provides an obvious inspiration for the relationships between those entities described here. Protocols for transferring names between registries and registrars have been standardized in the DNS space for some time (see [14]). Similarly, the division between service data acquired by resolving names with the DNS protocol vs. administrative data about names acquired through WHOIS [15] is directly analogous to the distinction between service and administrative data described in Section 2.2. The major difference between the data management architecture of the DNS and this framework is that the distinction between the CSP and User, due to historical policies of the telephone network, will often not exactly correspond to the distinction between a name service and a registrant in the DNS world - a User in the telephone network is today at least rarely in a direct relationship with a Registrar comparable to that of a DNS registrant.

The role of a Registry described here is a "thin" one, where the Registry manages basic allocation information for the numbering space, such as information about whether or not the number is assigned, and if assigned, by which Registrar. It is the Registrar that in turn manages detailed administrative data about those assignments, such as contact or billing information for the assignee. In some models, CSPs and Registrars will be combined (the same administrative entity), and in others the Registry and Registrar may similarly be composed. Typically, service data resides largely at the CSP itself, though in some models a "thicker" Registry may itself contain a pointer to the servicing CSP for a number or number block. In addition to traditional centralized Registries, this framework also supports environments where the same data is being managed by multiple administrative entities, and stored in many locations. A distributed registry system is discussed further in [19]. To support those use cases, it is important to distinguish the following:

Data store: A Data Store is a service that stores and enables access to administrative and/or service data.

Reference Address: A Reference Address is a URL that dereferences to the location of the data store.

Distributed data stores: In a Distributed Data Store, administrative or service data can be stored with multiple actors. For example, CSPs could provision their service data to multiple other CSPs.

Distributed Registries: Multiple Registries can manage the same numbering resource. In these architectures, actors could interact with one or multiple Registries. The Registries would update each other when change occurs. The Registries have to ensure that data remains consistent, e.g. that the same TN is not assigned to two different actors.

3. Framework

The framework outlined in this document requires three Internet-based mechanisms for managing and resolving telephone numbers (TNs) in an IP environment. These mechanisms will likely reuse existing protocols for sharing structured data; it is unlikely that new protocol development work will be required, though new information models specific to the data itself will be a major focus of framework development. Likely candidates for reuse here include work done in DRINKS [4] and WEIRDS [12], as well as the TeRI [16] framework.

These protocol mechanisms are scoped in a way that makes them likely to apply to a broad range of future policies for number administration. It is not the purpose of this framework to dictate number policy, but instead to provide tools that will work with policies as they evolve going forward. These mechanisms therefore do not assume that number administration is centralized, nor that number allocations are restricted to any category of service providers, though these tools must and will work in environments with those properties.

The three mechanisms are:

Acquisition: a protocol mechanism for acquiring TNs, including an enrollment process.

Management: a protocol mechanism for associating data with TNs.

Retrieval: a protocol mechanism for retrieving data about TNs.

The acquisition mechanism will enable actors to acquire TNs for use with a communications service by requesting numbering resources from a service operated by a Registrar, CSP or similar actor. TNs may be requested either on a number-by-number basis, or as inventory blocks.

Any actor who grants numbering resources will retain metadata about the assignment, including the responsible organization or individual to whom numbers have been assigned.

The management mechanism will let actors provision data associated with TNs. For example, if a User has been assigned a TN, they may select a CSP to provide a particular service associated with the TN, or a CSP may assign a TN to a User upon service activation. In either case, a mechanism is needed to provision data associated with the TN at that CSP, and to extend those data sets as CSPs (and even Users) require.

The retrieval mechanism will enable actors to learn information about TNs. For real-time service data, this typically involves sending a request to a CSP; for other information, an actor may need to send a request to a Registry rather than a CSP. Different parties may be authorized to receive different information about TNs.

As an example, a CSP might use the acquisition interface to acquire a chunk of numbers from a Registrar. Users might then provision administrative data associated with those numbers at the CSP through the management interface, and query for service data relating to those numbers through the retrieval interface of the CSP.

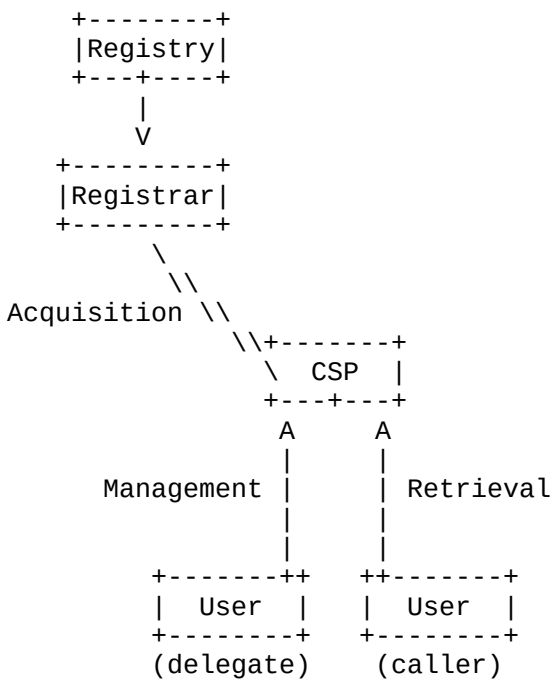


Figure 2: Example of the Three Interfaces

4. Use Cases

The high-level use cases in this section will provide an overview of the expected operation of the three interfaces in the MODERN problem space:

4.1. Acquisition

There are various scenarios for how TNs can be acquired by the relevant actors, that is, a CSP, Service Enabler, and a User. There are three actors from which numbers can be acquired: a Registrar, a CSP and a User (presumably one who is delegating to another party). It is assumed that Registrars are either the same entity as Registries, or that Registrars have established business relationships with Registries that enable them to distribute the numbers that the Registries administer. In these use cases, a User may acquire TNs either from a CSP or a Registry, or from an intermediate delegate.

4.1.1. Acquiring TNs from Registrar

The most traditional number acquisition use case is one where a CSP, such as a carrier, requests a block of numbers from a Registrar to hold as inventory or assign to customers.

Through some out-of-band business process, a CSP develops a relationship with a Registrar. The Registrar maintains a profile of the CSP and assesses whether or not CSPs meet the policy restrictions for acquiring TNs. The CSP may then request TNs from within a specific pool of numbers in the authority of the Registry; such as region, mobile, wireline, or freephone. The Registrar must authenticate and authorize the CSP, and then either grant or deny a request. When an assignment occurs, the Registry creates and stores administrative information related to the assignment such as TN status and Registrar contact information, and removes the specific TN(s) from the pool of those that are available for assignment. As a part of the acquisition and assignment process, the Registry provides to the Registrar any tokens or other material needed by a Credential Authority to issue credentials (for example, STIR certificates [17]) used to attest the assignment for future transactions. Depending on the policies of the Numbering Authorities, Registrars may be required to log these operations.

Before it is eligible to receive TN assignments, per the policy of a Numbering Authority, the CSP may need to have submitted (again, through some out-of-band process) additional qualifying information such as current utilization rate or a demand forecast.

There are two scenarios under which a CSP requests resources; they are requesting inventory, or they are requesting for a specific User or delegate. For the purpose of status information, TNs assigned to a User are always considered assigned, not inventory. The CSP will associate service information for that TN, e.g., service address, and make it available to other CSPs to enable interconnection. The CSP may need to update the Registrar regarding this service activation; this is part of the "TN status" maintained by the Registrar.

There are also use cases in which a User can acquire a TN directly from a Registrar. Today, a user wishing to acquire a freephone number may browse the existing inventory through one or more Registrars, comparing their prices and services. Each such Registrar either is a CSP, or has a business relationship with one or more CSPs to provide services for that freephone number. In this case, the User must establish some business relationship directly with a Registrar, similarly to how such functions are conducted today when Users purchase domain names. In this use case, after receiving a number assignment from the Registrar, a User will then obtain

communications service from a CSP, and provide to the CSP the TN to be used for that service. The CSP will associate service information for that TN, e.g., service address, and make it available to other CSPs to enable interconnection. The user will also need to inform the Registrar about this relationship.

4.1.2. Acquiring TNs from CSPs

Today, a User typically acquires a TN from CSP when signing up for communications service or turning on a new device. In this use case, the User becomes the delegate of the CSP. A reseller or a service bureau might also acquire a block of numbers from a CSP to be issued to Users.

Consider a case where a User creates or has a relationship with the CSP, and subscribes to a communications service which includes the use of a TN. The CSP collects and stores administrative data about the User. The CSP then activates the User on their network and creates any necessary service data to enable connectivity with other CSPs. The CSP could also update public or privileged databases accessible by other Actors. The CSP provides any tokens or other material needed by a Credential Authority to issue credentials to the User (for example, a STIR certificate [17]) to prove the assignment for future transactions. Such credentials could be delegated from the one provided by the Credential Authority to the CSP to continue the chain of assignment. CSPs may be required to log such transactions, if required by the policy of the Numbering Authority.

Virtually the same flow would work for a reseller: it would form a business relationship with the CSP, at which point the CSP would collect and store administrative data about the reseller and give the reseller any material needed for the reseller to acquire credentials for the numbers. A user might then in turn acquire numbers from the reseller: in this case, the delegate re-delegating the TNs would be performing functions done by the CSP, e.g., providing any credentials, collecting administrative data, or creative service data.

The CSP could assign a TN from its existing inventory or it could acquire a new TN from the Registrar as part of the assignment process. If it assigns it from its existing inventory, it would remove the specific TN from the pool of those available for assignment. It may also update the Registrar about the assignment so the Registrar has current assignment data. If a reseller or delegate CSP is acquiring the numbers, it may have the same obligations to provide utilization data to the Registry as the assignee, per Section 4.1.1.

4.2. Management

The management protocol mechanism is needed to associate administrative and service data with TNs, and may be used to refresh or rollover associated credentials.

4.2.1. Management of Administrative Data

Administrative data is primarily related to the status of the TN, its administrative contacts, and the actors involved in providing service to the TN. Protocol interactions for administrative data will therefore predominantly occur between CSPs and Users to the Registrar, or between Users and delegate CSPs to the CSP.

Some administrative data may be private, and would thus require special handling in a distributed data store model. Access to it does not require real-time performance therefore local caches are not necessary. And it will include sensitive information such as user and contact data.

Some of the data could lend itself to being publicly available, such as CSP and TN assignment status. In that case it would be deemed public information for the purposes of the retrieval interface.

4.2.1.1. Managing Data at a Registrar

After a CSP acquires a TN or block of TNs from the Registrar (per Section 4.1.1 above), it then provides administrative data to the Registrar as a step in the acquisition process. The Registrar will authenticate the CSP and determine if the CSP is authorized to provision the administrative data for the TNs in question. The Registry will update the status of the TN, i.e., that it is unavailable for assignment. The Registrar will also maintain administrative data provided by the CSP.

Changes to this administrative data will not be frequent. Examples of changes would be terminating service (see Section 4.2.3.2), changing the name or address of a User or organization, or changing a CSP or delegate. Changes should be authenticated by a credential to prove administrative responsibility for the TN.

In some cases, such as the freephone system in North America today, the User has a direct relationship with the Registrar. Naturally, these users could provision administrative data associated with their TNs directly to the Registrar, just as a freephone provider today maintains account and billing data. While delegates may not ordinarily have a direct relationship to a Registrar, some environments as an optimization might want to support a model where

the delegate updates the Registrar directly on changes, as opposed to sending that data to the CSP or through the CSP to the Registrar. As stated already, the protocol should enable Users to acquire TNs directly from a Registrar, which Registrar may or may not also act as a CSP. In these cases the updates would be similar to that described in Section 4.2.1.1.

In a distributed Registry model, TN status, e.g., allocated, assigned, available, unavailable, would need to be provided to other Registries in real-time. Other administrative data could be sent to all Registries or other Registries could get a reference address to the host Registry's data store.

4.2.1.2. Managing Data at a CSP

After a User acquires a TN or block of TNs from a CSP, the User will provide administrative data to the CSP. The CSP commonly acts as a Registrar in this case, maintaining the administrative data and only notifies the Registry of the change in TN status. In this case, the Registry maintains a reference address (see Section 2.3) to the CSP/ Registrar's administrative data store so relevant actors have the ability to access the data. Alternatively, a CSP could send the administrative data to an external Registrar to store. If there is a delegate between the CSP and user, they will have to ensure there is a mechanism for the delegate to update the CSP as change occurs.

4.2.2. Management of Service Data

Service data is data required by an originating or intermediate CSP to enable communications service to a User: a SIP URI is an example of one service data element commonly used to route communications. CSPs typically create and manage service data, however, it is possible that delegates and Users could as well. For most use cases involving individual Users, it is anticipated that lower-level service information changes (such as an end-user device receiving a new IP address) would be communicated to CSPs via existing protocols. For example, the baseline SIP REGISTER [2] method, even for bulk operations [13], would likely be used rather than through any new interfaces defined by MODERN.

4.2.2.1. CSP to other CSPs

After a User enrolls for service with a CSP, in the case where the CSP was assigned the TN by a Registrar, the CSP will then create a service address such as a SIP URI and associate it with the TN. The CSP needs to update this data to enable service interoperability. There are multiple ways that this update can occur, though most commonly service data is exposed through the retrieval interface (see

Section 4.3). For certain deployment architectures, like a distributed data store model, CSPs may need to provision data directly to other CSPs.

If the CSP is assigning a TN from its own inventory it may not need to perform service data updates as change occurs because the existing service data associated with inventory may be sufficient once the TN is put in service. They would however likely update the Registry on the change in status.

4.2.2.2. User to CSP

Users could also associate service data to their TNs at the CSP. An example is a User acquires a TN from the Registrar (as described in Section 4.1.1) and wants to provide that TN to the CSP so the CSP can enable service. In this case, once the user provides the number to the CSP, the CSP would update the Registry or other actors as outlined in Section 4.2.2.1.

4.2.3. Managing Change

This section will address some special management use cases that were not covered above.

4.2.3.1. Changing the CSP for an Existing Service

Consider the case where a User who subscribes to a communications service, and received their TN from that CSP, wishes to retain the same TN but move their service to a different CSP.

In the simplest scenario, where there’s an authoritative combined Registry/Registrar that maintains service data, the User could provide their credential to the new CSP and let the CSP initiate the change in service. The new CSP could then provide the new service data with the User’s credential to the Registry/Registrar, which then makes the change. The old credential is revoked and a new one is provided. The new CSP or the Registrar would send a notification to the old CSP, so they can disable service. The old CSP will undo any delegations to the User, including contacting the Credential Authority to revoke any cryptographic credentials (e.g., STIR certificates [17]) previously granted to the User. Any service data maintained by the CSP must be removed, and similarly, the CSP must delete any such information it provisioned in the Registry.

In a model similar to common practice in environments today, the User could alternatively provide their credential to the old CSP, and the old CSP initiates the change in service. Or, a User could go

directly to a Registrar to initiate a port. This framework should support all of these potential flows.

Note that in cases with a distributed Registry that maintained service data, the Registry would also have to update the other Registries of the change.

4.2.3.2. Terminating a Service

Consider a case where a user who subscribes to a communications service, and received their TN from the CSP, wishes to terminate their service. At this time, the CSP will undo any delegations to the User, which may involve contacting the Credential Authority to revoke any cryptographic credentials (e.g., STIR certificates [17]) previously granted to the User. Any service data maintained by the CSP must be removed, and similarly, the CSP must delete any such information it provisioned in the Registrar. However, per the policy of the Numbering Authority, Registrars and CSPs may be required to preserve historical data that will be accessible to Government Entities or others through audits, even if it is no longer retrievable through service interfaces.

The TN will change state from assigned to unassigned, the CSP will update the Registry. Depending on policies the TN could go back into the Registry, CSP, or delegate’s pool of available TNs and would likely enter an ageing process.

In an alternative use case, a User who received their own TN assignment directly from a Registrar terminates their service with a CSP. At this time, the User might terminate their assignment from the Registrar, and return the TN to the Registry for re-assignment. Alternatively, they could retain the TN and elect to assign it to some other service at a later time.

4.3. Retrieval

Retrieval of administrative or service data will be subject to access restrictions based on the category of the specific data: public, semi-restricted or restricted. Both administrative and service data can have data elements that fall into each of these categories. It is expected that the majority of administrative will fall into the semi-restricted category: access to this information may require some form of authorization, though service data crucial to reachability will need to be accessible. In some environments, it’s possible that none of the service data necessary to initiate communications will be useful to an entity on the public Internet, say, or that all that service data will have dependencies on the origination point of calls.

The retrieval protocol mechanism for semi-restricted and restricted data needs a way for the receiver of the request to identify the originator of the request and what is being requested. The receiver of the request will process that request based on this information.

4.3.1. Retrieval of Public Data

Either administrative or service data may be made publicly available by the authority that generates and provisions it. Under most circumstances, a CSP wants its communications service to be publicly reachable through TNs, so the retrieval interface supports public interfaces that permit clients to query for service data about a TN. Some service data may however require that the client be authorized to receive it, per the use case in Section 4.3.3 below.

Public data can simply be posted on websites or made available through a publicly available API. Public data hosted by a CSP may have a reference address at the Registry.

4.3.2. Retrieval of Semi-restricted Administrative Data

Consider a case in which a CSP is having service problems completing calls to a specific TN, so it wants to contact the CSP serving that TN. The Registry authorizes the originating CSP to access this information. It initiates a query to the Registry, the Registry verifies the requestor and the requested data and Registry responds with the serving CSP and contact data. However, CSPs might not want to make those administrative contact points public data: they are willing to share them with other CSPs for troubleshooting purposes, but not to make them available to general communication.

Alternatively that information could be part of a distributed data store and not stored at a monolithic Registry. In that case, the CSP has the data in a local distributed data store and it initiates the query to the local data store. The local data store responds with the CSP and contact data. No verification is necessary because it was done when the CSP was authorized to receive the data store.

4.3.3. Retrieval of Semi-restricted Service Data

Consider a case where a User on a CSP’s network calls a TN. The CSP initiates a query for service data associated with the TN to complete the call, and will receive special service data because the CSP operates in a closed environment where different CSPs receive different responses, and only participating CSPs can initiate communications. This service data would be flagged as semi-restricted. The query and response have real-time performance requirements in that environment.

Semi-restricted service data also works in a distributed data store model, where each CSP distributes its updated service data to all other CSPs. The originating CSP has the service data in its local data store and queries it. The local data store responds with the service data. The service data in the response can be a reference address to a data store maintained by the serving CSP, or it can be the service address itself. In the case where the response gives a reference address, a subsequent query would go to the serving CSP, who would in turn authorize the requestor for the requested data and respond appropriate. In the case where the original response contains the service address, the requestor would use that service address as the destination for the call.

In some environments, aspects of the service data may reside at the Registry itself (for example, the assigned CSP for a TN), and thus the query may be sent to the Registry. The Registry verifies the requestor and the requested data and responds with the service data, such as a SIP URI containing the domain of the assigned CSP.

4.3.4. Retrieval of Restricted Data

A Government Entity wishes to access information about a particular User, who subscribes to a communications service. The entity that operates the Registry on behalf of the Numbering Authority in this case has some pre-defined relationship with the Government Entity. When the CSP acquired TNs from the Numbering Authority, it was a condition of that assignment that the CSP provide access for Government Entities to telephone numbering data when certain conditions apply. The required data may reside either in the CSP or in the Registrar.

For a case where the CSP delegates a number to the User, the CSP might provision the Registrar (or itself, if the CSP is composed with a Registrar) with information relevant to the User. At such a time as the Government Entity needs information about that User, the Government Entity may contact the Registrar or CSP to acquire the necessary data. The interfaces necessary for this will be the same as those described in Section 4.3; the Government Entity will be authenticated, and an authorization decision will be made by the Registrar or CSP under the policy dictates established by the Numbering Authority.

5. Acknowledgments

We would like to thank Henning Schulzrinne and Adam Roach for their contributions to this problem statement and framework, and to thank Pierce Gorman for detailed comments.

6. IANA Considerations

This memo includes no instructions for the IANA.

7. Privacy Considerations

This framework defines two categories of information about telephone numbers: service data and administrative data. Service data describes how telephone numbers map to particular services and devices that provide real-time communication for users. As such, service data could potentially leak resource locations and even lower-layer network addresses associated with these services, and in rare cases, with end-user devices. Administrative data more broadly characterizes who the administrative entities are behind telephone numbers, which will often identify CSPs, but in some layers of the architecture could include personally identifying information (PII), even WHOIS-style information, about the end users behind identifiers. This could conceivably encompass the sorts of data that carriers and similar CSPs today keep about their customers for billing purposes, like real names and postal addresses. The exact nature of administrative data is not defined by this framework, and it is anticipated that the protocols that will perform this function will be extensible for different use cases, so at this point, it is difficult to characterize exactly how much PII might end up being housed by these services.

As such, if an attacker were to compromise the registrar services in this architecture which maintain administrative data, and in some cases even service data, this could leak PII about end users. These interfaces, and the systems that host them, are a potentially attractive target for hackers and need to be hardened accordingly. Protocols that are selected to fulfill these functions must provide the security features described in [Sec Cons].

Finally, this framework recognizes that in many jurisdictions, certain government agencies have a legal right to access service and administrative data maintained by CSPs. This access is typically aimed at identifying the users behind communications identifiers in order to enforce regulatory policy. Those legal entities already have the power to access the existing data held by CSPs in many jurisdictions, though potentially the administrative data associated with this framework could be richer information.

8. Security Considerations

The acquisition, management, and retrieval of administrative and service data associated with telephone numbers raises a number of security issues.

Any mechanism that allows an individual or organization to acquire telephone numbers will require a means of mutual authentication, of integrity protection, and of confidentiality. A Registry as defined in this document will surely want to authenticate the source of an acquisition request as a first step in the authorization process to determine whether or not the resource will be granted. Integrity of both the request and response is essential to ensuring that tampering does not allow attackers to block acquisitions, or worse, to commandeer resources. Confidentiality is essential to preventing eavesdroppers from learning about allocations, including the personally identifying information associated with the administrative or technical contracts for allocations.

A management interface for telephone numbers has similar requirements. Without proper authentication and authorization mechanisms in place, an attack could use the management interface to disrupt service data or administrative data, which could deny service to users, enable new impersonation attacks, prevent billing systems from operating properly, and cause similar system failures.

Finally, a retrieval interfaces has its own needs for mutual authentication, integrity protection, and for confidentiality. Any CSP sending a request to retrieve service data associated with a number will want to know that it is reaching the proper authority, that the response from that authority has not been tampered with in transit, and in most cases the CSP will not want to reveal to eavesdroppers the number it is requesting or the response that it has received. Similarly, any service answering such a query will want to have a means of authenticating the source of the query, and of protecting the integrity and confidentiality of its responses.

9. Informative References

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

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

[3] Bradner, S., Conroy, L., and K. Fujiwara, "The E.164 to Uniform Resource Identifiers (URI) Dynamic Delegation Discovery System (DDDS) Application (ENUM)", RFC 6116, DOI 10.17487/RFC6116, March 2011, <https://www.rfc-editor.org/info/rfc6116>.

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

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

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

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

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

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

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

[11] Peterson, J., Jennings, C., and R. Sparks, "Change Process for the Session Initiation Protocol (SIP) and the Real-time Applications and Infrastructure Area", BCP 67, RFC 5727, DOI 10.17487/RFC5727, March 2010, <https://www.rfc-editor.org/info/rfc5727>.

[12] Newton, A. and S. Hollenbeck, "Registration Data Access Protocol (RDAP) Query Format", RFC 7482, DOI 10.17487/RFC7482, March 2015, <https://www.rfc-editor.org/info/rfc7482>.

[13] Roach, A., "Registration for Multiple Phone Numbers in the Session Initiation Protocol (SIP)", RFC 6140, DOI 10.17487/RFC6140, March 2011, <https://www.rfc-editor.org/info/rfc6140>.

[14] Hollenbeck, S., "Generic Registry-Registrar Protocol Requirements", RFC 3375, DOI 10.17487/RFC3375, September 2002, <https://www.rfc-editor.org/info/rfc3375>.

[15] Daigle, L., "WHOIS Protocol Specification", RFC 3912, DOI 10.17487/RFC3912, September 2004, <https://www.rfc-editor.org/info/rfc3912>.

[16] Peterson, J., "An Architecture and Information Model for Telephone-Related Information (TeRI)", draft-peterson-modern-teri-03 (work in progress), July 2017.

[17] Peterson, J. and S. Turner, "Secure Telephone Identity Credentials: Certificates", RFC 8226, DOI 10.17487/RFC8226, February 2018, <https://www.rfc-editor.org/info/rfc8226>.

[18] Barnes, M., Jennings, C., Rosenberg, J., and M. Petit-Huguenin, "Verification Involving PSTN Reachability: Requirements and Architecture Overview", draft-jennings-vipr-overview-06 (work in progress), December 2013.

[19] Wendt, C. and H. Bellur, "Distributed Registry Protocol (DRiP)", draft-wendt-modern-drip-02 (work in progress), July 2017.

[20] Rosenberg, J. and H. Schulzrinne, "Session Initiation Protocol (SIP): Locating SIP Servers", RFC 3263, DOI 10.17487/RFC3263, June 2002, <https://www.rfc-editor.org/info/rfc3263>.

Authors’ Addresses

Jon Peterson
Neustar, Inc.
1800 Sutter St Suite 570
Concord, CA 94520
US

Email: jon.peterson@neustar.biz

Tom McGarry
Neustar, Inc.
1800 Sutter St Suite 570
Concord, CA 94520
US

Email: tom.mcgarry@neustar.biz

Network Working Group
Internet-Draft
Intended status: Informational
Expires: May 4, 2017

J. Peterson
Neustar

October 31, 2016

A JSON Binding and Encoding for TeRI
draft-peterson-modern-teri-json-00.txt

Abstract

The Telephone-Related Information (TeRI) framework defines an information model for data objects related to the acquisiton, management, and retrieval of telephone numbers and information related to them via the Internet. TeRI provides an abstract framework that must be instantiated by a particular binding and encoding. This document defines an HTTP binding for TeRI and a JavaScript Object Notation (JSON) encoding for TeRI.

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 May 4, 2017.

Copyright Notice

Copyright (c) 2016 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. Terminology 2
3. HTTP TeRI Binding 3
4. JSON TeRI Encoding 3
4.1. TeRI Requests 3
4.2. TeRI Responses 4
4.3. TeRI Records 4
5. Acknowledgments 4
6. IANA Considerations 5
7. Security Considerations 5
8. Informative References 5
Author's Address 5

1. Introduction

The Telephone-Related Information (TeRI) framework [I-D.peterson-modern-teri] defines an information model for data objects related to the acquisiton, management, and retrieval of telephone numbers and information related to them via the Internet. TeRI provides an abstract framework that must be instantiated by a particular binding and encoding, as described in [I-D.peterson-modern-teri] Section 6.2 and 6.3, respectively. This document defines an HTTP binding and JavaScript Object Notation (JSON) [RFC7159] encoding for TeRI. It does not however define any particular profile or deployment environment for using TeRI in this fashion; this only demonstrates an instantiation of the baseline TeRI specification using JSON.

This is an early stage Internet-Draft that serves primarily as a vehicle to give examples of a potential syntax for TeRI Requests and Responses in order to facilitate discussion.

2. Terminology

In this document, the key words "MAY", "MUST, "MUST NOT", "SHOULD", and "SHOULD NOT", are to be interpreted as described in [RFC2119]. This document also incorporates the terminology of the MODERN Framework [I-D.ietf-modern-problem-framework].

3. HTTP TeRI Binding

This specification defines a RESTful interface for getting and putting JSON objects related to TeRI at a web service.

TBD.

4. JSON TeRI Encoding

This specification defines separate JSON objects to carry TeRI Requests and Responses. All JSON objects begin with a "TeRI" element, which has a value stating whether the object contains a Request or a Response.

4.1. TeRI Requests

Per TeRI [I-D.peterson-modern-teri], all requests will have a Source and a Subject. Optionally, a block of Attributes will also appear in the Request.

This simplest TeRI request will therefore have the following form:

```
{ "TeRI":"Request",
  "Source":{"Request":"example.com"},
  "Subject":{"T":"12125551111"} }
```

A Request may have two types of Sources: a "Request" Source or an "Intermediary" Source. The "Source" element is given as an array here because a Request may have multiple Sources, including one "Request" Source and one or more "Intermediary" Sources. All Requests have a single Subject. The encoding of a Subject follows the Service Types given in [I-D.peterson-modern-teri] Section 4.2.1. Here, Type "T" signifies a single telephone number.

Most of the complexity in Requests comes from Attributes. Attributes appear in their own JSON array. An Attribute typically serves to qualify a request. In this example, the TeRI request is only asking for an Internet-based SMS service (see [I-D.peterson-modern-teri] Section 5.5.2) associated with the telephone number "12125551000".

```
{ "TeRI":"Request",
  "Source":{"Request":"example.com"},
  "Subject":{T":"12125551000" } ,
  "Attribute":{
      "Service":"sms"}
  }
```

4.2. TeRI Responses

All TeRI responses will give a Response Code. The simplest TeRI responses are therefore simple failure responses.

```
{ "TeRI":"Response",
  "Code":"Unauthorized Source" }
```

[TBD: Note that TeRI has not allocate Response Codes yet; these will accompany the human-readable response indicators like "Unauthorized Source"]

Most interesting TeRI responses contain Records, which are specified in Section 4.3.

4.3. TeRI Records

A TeRI Record consists of a JSON array containing a set of elements as defined in [I-D.peterson-modern-teri]. Records may appear in both TeRI Requests and Responses; for Retrieval Operations, it would be most common for Records to appear in Responses, when the Request indicates a Subject that the Source would like receive Records related to.

```
{ "TeRI":"Response",
  "Code":"Success",
  "Record:{
        "Identifier":"dc9a25c5-e44a-4dc1-9ff7-609da04bd694",
        "Authority":{"x5u":"http://example.com/cert.cert"},
        "Contact":"admin@example.com",
        "Service":{"U":"sip:alice@example.com"},
        "Signature":"dBjftJeZ4CVP-mB92K27uhbUJU1p1r_wW1gFWFOEjXk" }
        }
```

This is a simple example of a Record; more complex records may contain Priorities or Expiries. Note that the Service Type here follows the [I-D.peterson-modern-teri] Section 4.2.1 Types, as did the Attributes in the Request above.

5. Acknowledgments

We would like to thank you for your contributions to this problem statement and framework.

6. IANA Considerations

This memo includes no request to IANA.

7. Security Considerations

TBD.

8. Informative References

[I-D.ietf-modern-problem-framework]
Peterson, J. and T. McGarry, "Modern Problem Statement, Use Cases, and Framework", draft-ietf-modern-problem-framework-01 (work in progress), July 2016.

[I-D.peterson-modern-teri]
Peterson, J., "An Architecture and Information Model for Telephone-Related Information (TeRI)", draft-peterson-modern-teri-01 (work in progress), July 2016.

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

[RFC7159] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", RFC 7159, DOI 10.17487/RFC7159, March 2014, <http://www.rfc-editor.org/info/rfc7159>.

Author’s Address

Jon Peterson
Neustar, Inc.
1800 Sutter St Suite 570
Concord, CA 94520
US

Email: jon.peterson@neustar.biz

Network Working Group
Internet-Draft
Intended status: Informational
Expires: January 4, 2018

J. Peterson
Neustar

July 3, 2017

A JSON Binding and Encoding for TeRI
draft-peterson-modern-teri-json-01.txt

Abstract

The Telephone-Related Information (TeRI) framework defines an information model for data objects related to the acquisiton, management, and retrieval of telephone numbers and information related to them via the Internet. TeRI provides an abstract framework that must be instantiated by a particular binding and encoding. This document defines an HTTP binding for TeRI and a JavaScript Object Notation (JSON) encoding for TeRI.

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 4, 2018.

Copyright Notice

Copyright (c) 2017 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. Terminology 2
3. HTTP TeRI Binding 3
4. JSON TeRI Encoding 3
4.1. TeRI Requests 3
4.2. TeRI Responses 4
4.3. TeRI Records 4
5. Acknowledgments 4
6. IANA Considerations 5
7. Security Considerations 5
8. Informative References 5
Author's Address 5

1. Introduction

The Telephone-Related Information (TeRI) framework [I-D.peterson-modern-teri] defines an information model for data objects related to the acquisiton, management, and retrieval of telephone numbers and information related to them via the Internet. TeRI provides an abstract framework that must be instantiated by a particular binding and encoding, as described in [I-D.peterson-modern-teri] Section 6.2 and 6.3, respectively. This document defines an HTTP binding and JavaScript Object Notation (JSON) [RFC7159] encoding for TeRI. It does not however define any particular profile or deployment environment for using TeRI in this fashion; this only demonstrates an instantiation of the baseline TeRI specification using JSON.

This is an early stage Internet-Draft that serves primarily as a vehicle to give examples of a potential syntax for TeRI Requests and Responses in order to facilitate discussion.

2. Terminology

In this document, the key words "MAY", "MUST, "MUST NOT", "SHOULD", and "SHOULD NOT", are to be interpreted as described in [RFC2119]. This document also incorporates the terminology of the MODERN Framework [I-D.ietf-modern-problem-framework].

3. HTTP TeRI Binding

This specification defines a RESTful interface for getting and putting JSON objects related to TeRI at a web service.

TBD.

4. JSON TeRI Encoding

This specification defines separate JSON objects to carry TeRI Requests and Responses. All JSON objects begin with a "TeRI" element, which has a value stating whether the object contains a Request or a Response.

4.1. TeRI Requests

Per TeRI [I-D.peterson-modern-teri], all requests will have a Source and a Subject. Optionally, a block of Query Restrictions may also appear in the Request.

This simplest TeRI request will therefore have the following form:

```
{ "TeRI":"Retrieval",
  "Source":[{"Request":"example.com"}],
  "Subject":{"T":"12125551111"}
  }
```

The "Source" element is given as an array here because a TeRI Request may have multiple Sources, including one "Request" Source and one or more "Intermediary" Sources. All Requests have a single Subject. The encoding of a Subject follows the Service Types given in [I-D.peterson-modern-teri] Section 4.2.1. Here, Type "T" signifies a single telephone number.

Most of the complexity in Requests comes from Query Restrictions. Query Restrictions appear in their own JSON array, and typically serve to qualify a request. In this example, the TeRI request is only asking for Service Records, for an SMS service (see [I-D.peterson-modern-teri] Section 5.5.2) associated with the telephone number "12125551000".

```
{ "TeRI":"Retrieval",
  "Source":{"Request":"example.com"},
  "Subject":{T":"12125551000"} ,
  "Restriction":[
          ("Service":"sms"} ]
  }
```

4.2. TeRI Responses

All TeRI responses will give a Response Code. The simplest TeRI responses are therefore simple failure responses.

```
{ "TeRI":"Response",
  "Code":"Unauthorized Source"
  }
```

[TBD: Note that TeRI has not allocate Response Codes yet; these will accompany the human-readable response indicators like "Unauthorized Source"]

Most interesting TeRI responses contain Records, which are specified in Section 4.3.

4.3. TeRI Records

A TeRI Record consists of a JSON array containing a set of elements as defined in [I-D.peterson-modern-teri]. Records may appear in both TeRI Requests and Responses; for Retrieval Operations, it would be most common for Records to appear in Responses, when the Request indicates a Subject that the Source would like receive Records related to.

```
{ "TeRI":"Response",
  "Code":"Success",
  "Record:{
        "Identifier":"dc9a25c5-e44a-4dc1-9ff7-609da04bd694",
        "Authority":{"x5u":"http://example.com/cert.cert"},
        "Subject":[{T":"12125551000"}],
        "Access":"Public",
        "Service":{"sms":"sip:alice@example.com"},
        "Signature":"dBjftJeZ4CVP-mB92K27uhbUJU1p1r_wW1gFWFOEjXk" }
        }
```

This is a simple example of a Record; more complex records may contain Priorities or Expiries. Note that the Service Type here follows the [I-D.peterson-modern-teri] Section 4.2.1 Types, as did the Restrictions in the Request above.

5. Acknowledgments

We would like to thank YOU for your contributions to this specification.

6. IANA Considerations

This memo includes no request to IANA.

7. Security Considerations

TBD.

8. Informative References

[I-D.ietf-modern-problem-framework]
Peterson, J. and T. McGarry, "Modern Problem Statement, Use Cases, and Framework", draft-ietf-modern-problem-framework-02 (work in progress), March 2017.

[I-D.peterson-modern-teri]
Peterson, J., "An Architecture and Information Model for Telephone-Related Information (TeRI)", draft-peterson-modern-teri-02 (work in progress), October 2016.

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

[RFC7159] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", RFC 7159, DOI 10.17487/RFC7159, March 2014, <http://www.rfc-editor.org/info/rfc7159>.

Author’s Address

Jon Peterson
Neustar, Inc.
1800 Sutter St Suite 570
Concord, CA 94520
US

Email: jon.peterson@neustar.biz

modern
Internet-Draft
Intended status: Standards Track
Expires: May 4, 2017

C. Wendt
Comcast

October 31, 2016

Identity Registry (idreg)
draft-wendt-modern-identity-registry-00

Abstract

This document will describe an approach for how a distributed identity registry model might look. It will consider both public registry components of the data model necessary for routing calls from one globally routable identity to another. It will also consider part of the private registry components a provider may need to manage associations with users or customers. Other topics include provider associations, application or service association, and the ability to support multiple identities associated with a user/ subscriber (e.g. telephone number and e-mail identity).

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 May 4, 2017.

Copyright Notice

Copyright (c) 2016 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. Terminology 3
3. Overview 3
3.1. Identity Data Model 3
3.2. Other identity registry attributes 4
4. Message and Control Flows 5
4.1. Queries 5
4.2. Allocation/Assignment 5
4.2.1. Example 5
4.3. Update Entry/Port 6
4.4. Removal/de-allocation 6
5. Security Considerations 6
6. Acknowledgements 6
7. References 6
7.1. Normative References 6
7.2. Informative References 6
Author’s Address 7

1. Introduction

There are many useful VoIP and user to user communications applications that desire the ability to provide services that don’t depend on a single entity or provider to manage the end-to-end identities associated with that application. For example, using the VoIP protocol, SIP [RFC3261], the telephone network provides a federated mechanism that using a publicly known identity, the telephone number, a customer of a telephone provider A can call a customer of telephone provider B based on managed routing databases and routing rules. XMPP [RFC6120] is another example of a protocol that allowed federation of communications based on the username and domain of the host of the XMPP server. Each of these examples uses service specific databases or registries that are generally protocol or application specific, however today application providers general provide many applications or services for a user which generally share the use of common communications identities like telephone numbers, e-mail identities, or identities associated with web based IdPs.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

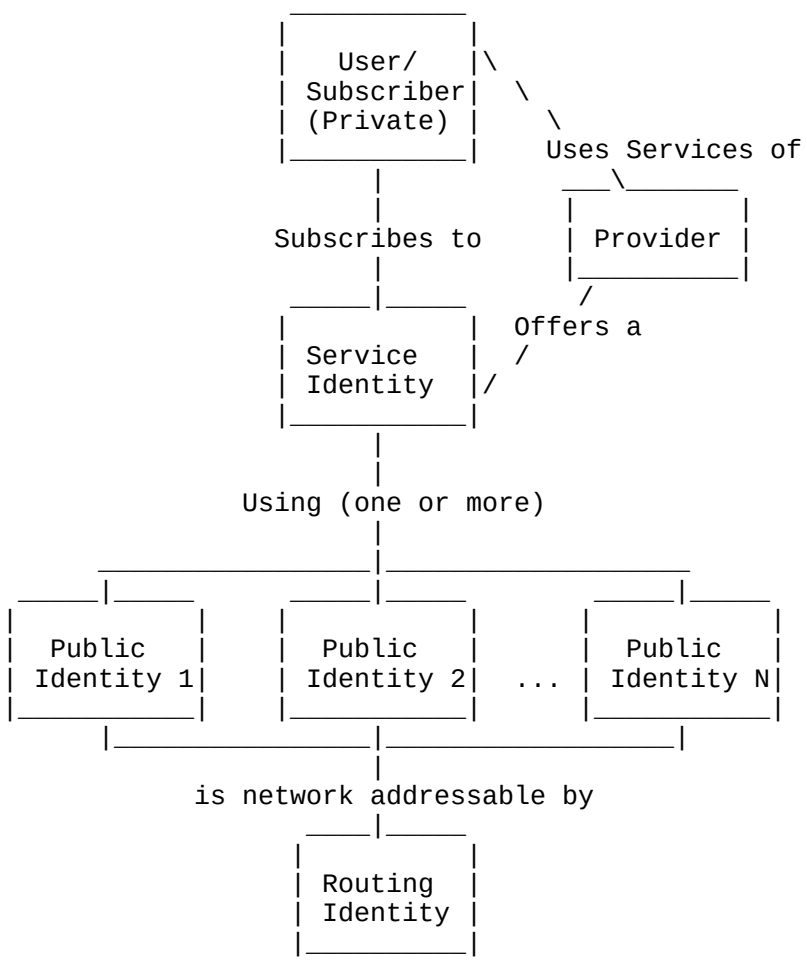
3. Overview

The identity registry model proposed in this document supports the model where there are a few actors in the model relevant to providing communications services.

- Provider - An entity that provide a service to customers and manages there identity in the network.
- User/Subscriber - The entity that is using the services of the provider.
- Service Identity - A globally unique identifier representing a application or service being made available to users/subscribers by multiple providers.
- Public Identity - A publicly known identity that the user associates with a service. This identity must be globally unique to a user/subscriber. It must also be provably associated to a given user/subscriber that claims the association.
- Routing Identity - A uniquely and globally routable identity used specifically in signaling calls between users.

This data model can be used to build the shared data between providers that support the federated service in order for users that are associated with one provider to call another provider.

3.1. Identity Data Model



3.2. Other identity registry attributes

The identity registry MUST support functions such as the following:

- The ability to query for available/unused identities for the purposes of either identifying conflicts before committing to the registry or identify unused identities that are part of a pool (e.g. telephone numbers)

- The ability to allocate identities for future use at individual levels or at block levels, such as NPA-NXX level telephone numbers or perhaps wildcard identities, e.g. *@example.com.

- The ability to update/transfer/port identities from one provider to another provider.

- The ability to digitally sign transactions to a provider for validation of legitimate transactions. Or forensic analysis of illegitimate transactions.

It is anticipated that this identity registry would be used with [I-D.wendt-modern-drip] for supporting a continuously and timely updated local registry for a given service identity the provider is offering.

4. Message and Control Flows

4.1. Queries

Typical queries for finding a globally routable identity should be in the context of a public identity and service identity for an allocated routing identity.

4.2. Allocation/Assignment

When a provider customer has decided to allocate a given single or block level set of telephone numbers there is a PUT command that allocates the number, given the number wasn't already allocated between the GET and the PUT. As a result of a successful allocation, the telephone number will be removed from the unallocated bucket.

4.2.1. Example

As part of the allocation, the service provider will be required to provide following information:

- publicID: telephone number in e.164 format (e.g., +12155551212).

- serviceID: "voip" by default, other services potentially in future.

- routingID: SIP URI with telephone number + domain representing service provider of record (e.g., sip:+12155551212@voip.example.com).

- timestamp: a timestamp retrieved from a common NTP server representing time of allocation, used for validating which service provider allocated first in race condition scenarios, and just for logging and historical reference in general.

- x5u: used for validation of signature

- o signature: using a provider level [RFC5280] based private key/ certificate, the provider MUST sign the information above to validate the change to the registry.

4.3. Update Entry/Port

If a provider needs to update information related to an allocated entry, such as adding a publicID, modify routingID, etc. or if there is a port where a new service provider will overwrite the entry with new information, the API should be the same.

There is a GET operation to read the current entry information, if the provider needs this information, (e.g., read/modify/write). There also is a PUT operation that will write the updated entry information. This will require a new timestamp and signature to validate the security of the operation and logging/historical purposes.

4.4. Removal/de-allocation

If a provider wants to remove an entry for the case where a customer removes his service and no longer wants to own or associate a public identity, a DELETE operation will be provided that will delete the entry, and for the case of a telephone number, will put the telephone number back in the pool of unallocated numbers.

5. Security Considerations

TBD

6. Acknowledgements

Thanks to Harsha Bellur for collaboration on developing this model and it’s implementation.

7. References

7.1. Normative References

[I-D.wendt-modern-drip]
Bellur, H. and C. Wendt, "Distributed Registry Protocol", draft-wendt-modern-drip-01 (work in progress), July 2016.

7.2. Informative References

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

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

[RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, DOI 10.17487/RFC5280, May 2008, <http://www.rfc-editor.org/info/rfc5280>.

[RFC6120] Saint-Andre, P., "Extensible Messaging and Presence Protocol (XMPP): Core", RFC 6120, DOI 10.17487/RFC6120, March 2011, <http://www.rfc-editor.org/info/rfc6120>.

Author’s Address

Chris Wendt
Comcast
One Comcast Center
Philadelphia, PA 19103
USA

Email: chris-ietf@chriswendt.net

modern
Internet-Draft
Intended status: Standards Track
Expires: September 14, 2017

C. Wendt
Comcast

March 13, 2017

Identity Registry (idreg)
draft-wendt-modern-identity-registry-01

Abstract

This document will describe an approach for how a distributed identity registry model might look. It will consider both public registry components of the data model necessary for routing calls from one globally routable identity to another. It will also consider part of the private registry components a provider may need to manage associations with users or customers. Other topics include provider associations, application or service association, and the ability to support multiple identities associated with a user/ subscriber (e.g. telephone number and e-mail identity).

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 September 14, 2017.

Copyright Notice

Copyright (c) 2017 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. Terminology 3
3. Overview 3
3.1. Identity Data Model 3
3.2. Other identity registry attributes 4
4. Message and Control Flows 5
4.1. Queries 5
4.2. Allocation/Assignment 5
4.2.1. API definition 5
4.2.2. Example 7
4.3. Update Entry/Port 7
4.3.1. API definition 7
4.4. Removal/de-allocation 8
4.4.1. API definition 8
5. Security Considerations 9
6. Acknowledgements 9
7. References 9
7.1. Normative References 9
7.2. Informative References 9
Author’s Address 9

1. Introduction

There are many useful VoIP and user to user communications applications that desire the ability to provide services that don’t depend on a single entity or provider to manage the end-to-end identities associated with that application. For example, using the VoIP protocol, SIP [RFC3261], the telephone network provides a federated mechanism that using a publicly known identity, the telephone number, a customer of a telephone provider A can call a customer of telephone provider B based on managed routing databases and routing rules. XMPP [RFC6120] is another example of a protocol that allowed federation of communications based on the username and domain of the host of the XMPP server. Each of these examples uses service specific databases or registries that are generally protocol or application specific, however today application providers general provide many applications or services for a user which generally share the use of common communications identities like telephone numbers, e-mail identities, or identities associated with web based IdPs.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

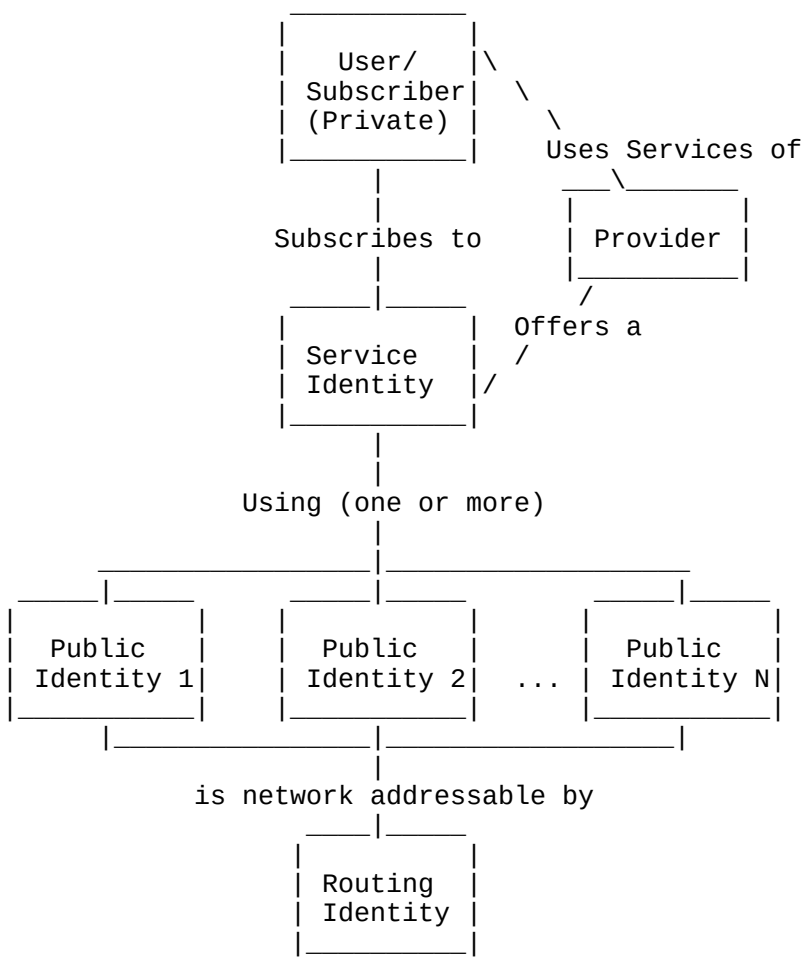
3. Overview

The identity registry model proposed in this document supports the model where there are a few actors in the model relevant to providing communications services.

- Provider - An entity that provide a service to customers and manages there identity in the network.
- User/Subscriber - The entity that is using the services of the provider.
- Service Identity - A globally unique identifier representing a application or service being made available to users/subscribers by multiple providers.
- Public Identity - A publicly known identity that the user associates with a service. This identity must be globally unique to a user/subscriber. It must also be provably associated to a given user/subscriber that claims the association.
- Routing Identity - A uniquely and globally routable identity used specifically in signaling calls between users.

This data model can be used to build the shared data between providers that support the federated service in order for users that are associated with one provider to call another provider.

3.1. Identity Data Model



3.2. Other identity registry attributes

The identity registry MUST support functions such as the following:

- The ability to query for available/unused identities for the purposes of either identifying conflicts before committing to the registry or identify unused identities that are part of a pool (e.g. telephone numbers)

- The ability to allocate identities for future use at individual levels or at block levels, such as NPA-NXX level telephone numbers or perhaps wildcard identities, e.g. *@example.com.

- The ability to update/transfer/port identities from one provider to another provider.

- The ability to digitally sign transactions to a provider for validation of legitimate transactions. Or forensic analysis of illegitimate transactions.

It is anticipated that this identity registry would be used with [I-D.wendt-modern-drip] for supporting a continuously and timely updated local registry for a given service identity the provider is offering.

4. Message and Control Flows

4.1. Queries

Typical queries for finding a globally routable identity should be in the context of a public identity and service identity for an allocated routing identity.

4.2. Allocation/Assignment

When a provider customer has decided to allocate a given single or block level set of telephone numbers there is a PUT command that allocates the number, given the number wasn’t already allocated between the GET and the PUT. As a result of a successful allocation, the telephone number will be removed from the unallocated bucket.

4.2.1. API definition

Request:
PUT /idreg/createidentity

Pass the following object (JSON) in the body.

Property	Type	Description
user_type	string	(MAND) Type representing user/sub
user_type_id	string	(MAND) ID associated with user Example: accountID of user
user_info	stringified JSON	(OPT) User specific metadata
service_id	string	(MAND) Service type identifier Example: "pstn", "voip". "volte"
public_id	string	(MAND) User associated service identifier. Example: telephone number

An Authorization Header MUST be included with a JWT including timestamp, x5u, and signature that will be associated with this transaction.

Response:

Code	Status
201	user profile created, associate public id, returns new routing ID
200	user profile and public id association already exists returns the same routing ID (Idempotent)
204	service identifier not found
400	input errors
401	unauthorized API access - Signature validation failed
5xx	errors related to DB access and other system anomalies

For HTTP/1.1 200 OK and HTTP/1.1 201 Created responses:

Property	Type	Description
user_id	string	Globally Unique ID (UUID) for user.
routing_id	string	routing ID

4.2.2. Example

As part of the allocation, the service provider will be required to provide following information:

- publicID: telephone number in e.164 format (e.g., +12155551212).
- serviceID: "voip" by default, other services potentially in future.
- routingID: SIP URI with telephone number + domain representing service provider of record (e.g., sip:+12155551212@voip.example.com).
- timestamp: a timestamp retrieved from a common NTP server representing time of allocation, used for validating which service provider allocated first in race condition scenarios, and just for logging and historical reference in general.
- x5u: used for validation of signature
- signature: using a provider level [RFC5280] based private key/ certificate, the provider MUST sign the information above to validate the change to the registry.

4.3. Update Entry/Port

If a provider needs to update information related to an allocated entry, such as adding a publicID, modify routingID, etc. or if there is a port where a new service provider will overwrite the entry with new information, the API should be the same.

There is a GET operation to read the current entry information, if the provider needs this information, (e.g., read/modify/write). There also is a PUT operation that will write the updated entry information. This will require a new timestamp and signature to validate the security of the operation and logging/historical purposes.

4.3.1. API definition

The PUT /idreg/createidentity API can be used for updates to entries as it’s an idempotent API. For porting of telephone numbers either createidentity or a combination of the delete described in the next section and createidentity can be used.

4.4. Removal/de-allocation

If a provider wants to remove an entry for the case where a customer removes his service and no longer wants to own or associate a public identity, a DELETE operation will be provided that will delete the entry, and for the case of a telephone number, will put the telephone number back in the pool of unallocated numbers.

4.4.1. API definition

Request:

DELETE /idreg/identitymapping/serviceid/:id/publicid/:id

Pass the following object (JSON) in the body.

Property	Type	Description
service_id	string	(MAND) Service type identifier Example: "pstn", "voip". "volte"
public_id	string	(MAND) User associated service identifier. Example: telephone number

An Authorization Header MUST be included with a JWT including timestamp, x5u, and signature that will be associated with this transaction.

Response:

Code	Status
200	public ID association deleted
204	record with service_id and public_id in request URI not found
400	input errors
401	unauthorized API access - Signature validation failed
5xx	errors related to DB access and other system anomalies

For HTTP/1.1 200 OK and HTTP/1.1 201 Created responses:

Property	Type	Description
user_id	string	Globally Unique ID (UUID) for user.
routing_id	string	routing ID

5. Security Considerations

TBD

6. Acknowledgements

Thanks to Harsha Bellur for collaboration on developing this model and it’s implementation.

7. References

7.1. Normative References

[I-D.wendt-modern-drip]
Bellur, H. and C. Wendt, "Distributed Registry Protocol", draft-wendt-modern-drip-01 (work in progress), July 2016.

7.2. Informative References

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

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

[RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, DOI 10.17487/RFC5280, May 2008, <http://www.rfc-editor.org/info/rfc5280>.

[RFC6120] Saint-Andre, P., "Extensible Messaging and Presence Protocol (XMPP): Core", RFC 6120, DOI 10.17487/RFC6120, March 2011, <http://www.rfc-editor.org/info/rfc6120>.

Author’s Address

Chris Wendt
Comcast
One Comcast Center
Philadelphia, PA 19103
USA

Email: chris-ietf@chriswendt.net