Session Peering Provisioning Framework Analysis

Yoram Haddad *, Mickael Marrache *,†, Samuel Melloul * and David Schwartz †
department:Jerusalem College of Technology, Israel
department:XConnect Global Networks, London, UK

Abstract—Most computer-based systems require the ability to store information continuously in order to meet life cycle requirements of the data. The simplest way to meet these requirements typically involves using a centralized data store (relational or otherwise). In this paper, we analyze these requirements as they present themselves in the Voice over IP (VoIP) field, discuss the approach proposed by the Internet Engineering Task Force (IETF) to address the requirements, and, finally, suggest some improvements to this proposal to be incorporated in a future revision of the standard.

I. INTRODUCTION

VoIP (Voice over IP) as a technology is supported by a set of protocols that allow voice call (session) establishment and completion over the Internet. Specifically, the protocols address session establishment and media exchange. In terms of the rendezvous problem, early thinking proposed an email-like approach [1] involving SIP (Session Initiation Protocol) for session establishment and URIs (Uniform Resource Identifier, similar to email addresses). The approach was simple, storage of the remote party identifier (i.e. URI) could leverage the worldwide DNS (Domain Name System) infrastructure and routing information would be inherent in the identifier (e.g. sip:john@somecompany.com would entail finding an A record of somecompany.com and route the call to there). In practice, however, this approach was never seriously adopted for all the same reasons email is being replaced today by closed social group alternatives. In addition, people were just too used to using phone numbers for phone calls and the notion of calling an email address seemed foreign.

Instead, the identifier that emerged was none other than that all too familiar E.164 number (or Plain Old Telephone Number) [2] with all the philosophical issues (e.g. who really owns the number), network issues (e.g. how to route), and architectural issues (e.g. how to store, retrieve, provision, etc.) that the use of this type of identifier entails. Without addressing the philosophical issues, early attempts at solving the practical issues revolved around a technology known as ENUM (E.164 Number Mapping) [3] whereby the phone number is encoded in order to take advantage of DNS infrastructure. While ENUM addressed networking and some aspects of the architectural issues (store and retrieve), the fundamental difference between a URI and the whole ownership and delegation nature of phone numbers made the provisioning of these identifiers in the public domain (a.k.a Public or User ENUM) a colossal failure.

Instead, providers started using these identifiers, initially internally and eventually in closed federations or clubs (a.k.a carrier ENUM), necessitating the missing piece of the puzzle that is provisioning of these identifiers by the various members of the federations or clubs. This paper focuses on this provisioning process as it relates to the data store known as a numbering registry, or registry for short.

II. BACKGROUND

More and more enterprises today support voice and video services (as well as other session-based services) over the Internet. In order for this to be possible, the calling enterprise or Service Provider (SP for short) needs some information about the receiving enterprise or Service Provider. This information is known as Session Establishment Data (SED) and includes attributes such as the routing address of the ingress point at the receiving end (think DNS A record). The data provisioned for session establishment (SED) is typically used by various downstream SIP signaling systems to route a call to the next hop associated with the called domain. More specifically, the SED data is the set of parameters that the outgoing signaling path border elements (SBEs) need to initiate the session. See [4] for more details.

As mentioned in the introduction, the lookup process uses the phone number of the receiving party as the key to the query and for the purpose of this paper, the database storing this information is known as the registry. You will note that this lookup is done at call time and the information retrieved may be highly dynamic in nature.

Figure 1 presents a sequence diagram depicting the interaction between the components participating in call establishment using SIP. In this figure one can see that SIP proxy server A of one service provider needs routing related information to establish the session to SIP proxy server B of another service provider. The central component in the figure is explained later in this section. The difference between this flow and a regular SIP call establishment flow as defined in [5] is shown in red.

While the main piece of Session Establishment Data (SED) information is the routing address of the remote SIP proxy server, SED can include additional attributes of the call such as a cost that is associated with a routing option. This attribute would for instance allow a remote party to control access to its SIP services by specifying an originating party specific cost. In essence, the relatively straightforward rendezvous concept can be extended to include business rules that alter or filter this information based on the identity of the querying party. Throughout this document, these rules are commonly referred to peering or peering policies.

In terms of the registry, service providers may mutually define peering policies in the registry to represent routing information sharing agreements (generally commercial agreements). Peering policies are relationships between service providers for sharing routing information.
Multiple architectures exist to manage the information (i.e., routing related and peering related information). In one option, the service providers store this information locally and maintain it on their own via bilateral relationships with all other providers. Using this approach, there is one registry per service provider and besides the duplication of data, the biggest disadvantage of this approach is that each service provider needs to manage all this information locally, exposing himself to serious scalability issues when the number of related service providers grows.

Figure 2 illustrates a more efficient approach that consists of using a centralized registry whereby all participating providers provision (upload) their data to a shared location that manages the information on behalf of all the participants. In this case, a single registry is maintained that contains both routing and peering-related information for all the service providers. In essence by centralizing the information, better efficiency is achieved with no duplication of data, not to mention the reduced architectural complexity.

The main concern of this paper is how the different service providers interact with a centralized registry for data provisioning. Until now, the interface allowing for this sort of interaction was proprietary. Specifically, a service provider was forced to design its provisioning system according to the proprietary interface, remaining tightly coupled to this particular solutions provider. A standardized solution would let service providers avoid this undesirable coupling and be able to switch to another solutions provider easily.

In the past, the protocol OMA-IMPS had been issued by GSMA to ensure interworking between service providers for IM (Instant Messaging) services. A specification proposing a public solution for the particular problem presented in this document had been published by Cable Television Laboratories Inc [6]. Although this solution was publicly defined, it had never been standardized.

Currently, there is undergoing work at IETF within the
DRINKS working group to standardize a solution to this problem. The solution consists of a framework called SPPF (Session Peering Provisioning Framework) that defines a data model representing the information and a set of specifications describing how registry provisioning is performed. The DRINKS working group proposed as well one possible implementation of the registry provisioning protocol using SOAP (Simple Object Access Protocol) technology: SPPP (Session Peering Provisioning Protocol) over SOAP.

In this paper we present the preliminary conclusions of the drafts [7] and [8] for respectively SPPF and its SOAP implementation (both not yet finalized as RFC) together with some new use case illustrations. Thanks to the collaboration between an academic institute and a high-tech company specialized in this area, the team came up to some challenging issues arising from the current drafts and contributed to improving the developing standard.

Specifically, modifications and improvements were proposed which may lead to more efficient operation for products that would implement the future standard. We believe this is a good example where collaboration between research and industry lead to modification ahead of the product deployment.

The remainder of this paper is organized as follows. In the next section, we describe briefly the use cases on which the solution was built. In sections IV and V we present the interfaces and data model defined by the framework, respectively. Then interactive sharing mechanism, source based routing and egress routes are presented in sections VI, VII and VIII, respectively. Finally, we present an extensive analysis of the open issues of the standard and propose some solutions in section IX.

It is important to note that SPPF is only a provisioning protocol and not a query protocol. Representation of the information in the registries is defined by SPPF and is adapted for provisioning. However, it is not adapted for use by a query protocol. See [9] for more details.

III. USE CASES

Use cases are documented in [10] and vis-a-vis this document the following interconnect use cases need to be described:

A. Inter-Service Provider

Service providers create peering relationships with other service providers in order to establish interconnects. Establishing these interconnects involves, among other things, communicating and enabling the points of ingress and other establishment data used to establish sessions.

B. Direct and Indirect Peering

Some inter service provider peering relationships are created to enable the establishment of sessions to the public identifiers for which a service provider is the carrier-of-record. This is referred to as direct peering. Other inter service provider peering relationships are created to enable the establishment of sessions to public identifiers for which a service provider is a transit provider. This is referred to as indirect peering. Some service providers take into consideration a service provider’s role as a transit or carrier-of-record provider when selecting a route to a public identifier.

C. Intra-Service Provider

Service providers support the establishment of sessions between their own public identifiers, not just to other service providers public identifiers. Enabling this involves, among other things, communicating and enabling intra service provider signaling points and other establishment data that can differ from inter-service provider signaling points and establishment data.

D. Selective Peering

Service providers create peering relationships with other service providers in order to establish interconnects. However, service provider peering relationships often result in different points of ingress or other establishment data for the same set of public identifiers. This is referred to as selective peering and is done on a route group basis.

E. Delegated Hierarchy

A service provider may decide to maintain its own infrastructure to contain the route records that constitute the terminal step in the Lookup Function (LUF). In such cases, the service provider will provision registries to direct queries for the service providers’ public identifiers to its own infrastructure rather than provisioning the route records directly. For example, in the case of DNS-based route records, such a delegated hierarchy would make use of NS and CNAME records, while a flat structure would make use of NAPTR resource records.

IV. THE DIFFERENT INTERFACES

Figure 3 presents the different interfaces that are associated with the provisioning process.

As explained previously, a service provider provisions his SED information to the registry using an SPPF implementation. This is represented by the interface 1 in Figure 3 and it is this interface that is the main concern of this paper.

Although out of scope for this document, for the sake of completeness the other two interfaces are presented as well. A distribution mechanism (interface 2) is needed for two main reasons. Firstly, the components that access the data
Organization formation peering: organizations. The we will explain an important concept that is the basis of in-

duction, namely the two categories of information present in the registry which are routing information and peering

relationships between them. We deliberately omitted the attributes for each element since we are only presenting a high level introduction of the framework (more details can be found in [7] and [8]).

In this figure, we can see the concepts defined in intro-
duction, namely the two categories of information present in the registry which are routing information and peering information. Before introducing each one of these elements, we will explain an important concept that is the basis of information peering: organizations. The **Organization** element represents a party involved in the registry. A good example of organization is the aforementioned service provider. In the context of SPPF, an organization may play any combination of three roles: **registrant**, **registrar** and **peering organization**. A registrant represents an organization that owns some information in the registry. A registrar represents an organization that provisions some information to the registry on behalf of a registrant organization. The registrar concept is mainly used for tracking operations done on the registry and for authorization. Finally, a peering organization represents an organization authorized by a registrant organization, to access some of its provisioned information. A short example is now presented to illustrate these concepts. Assume that service provider A provisioned service provider B’s routing information (generally concerning its own subscribers) into the registry. Service providers A and B are said registrar and registrant of this information, respectively, since service provider A has just performed the provisioning operation while service provider B is the owner of the information. Now, let’s assume B has decided to share its routing information to service provider C. In this case, B should provision the registry in peering information declaring that it shares its routing information to C, where C is now referred to as a peering organization (i.e. an organization to whom information is shared).

We define now the concept of public identifier. As mentioned in introduction, an organization is interested to get the information needed to establish a session. For example, a service provider may want to establish a voice call to an endpoint identified by the telephone number 1-(212)-123-4567. Thus, it needs to query the registry (or the LDR for better efficiency) to retrieve the SED associated to this telephone number. In this case, the key of the query is the called telephone number. SPPF defines a generic type for these keys: **Public Identifier**.

A public identifier may be one of the following elements: a **single telephone number** represented by the **TN** type, a **telephone number prefix** represented by the **TNP** type, a **telephone number range** represented by the **TNR** type or a **routing number** represented by the **RN** type. The latter is an identifier very similar to a telephone number and is used in case of number portability to identify the switch to which the session should be redirected. The other types of identifiers except the TN type allow representing ranges of telephone numbers. A TNP represents a modulo-10 range of telephone numbers (e.g. 1-(212)-123) whereas a TNR gives the possibility to represent a more fine-grained range (e.g. from 1-(212)-123-0000 to 1-(212)-123-1111).

After defining what a public identifier is, we have to describe how the results to these queries are represented by SPPF. These results are generally routes. SPPF often calls these routes ingress SBE (Session Border Element) of a target SSP since they identify the SBE (which is an ingress or way in component) of the service provider associated to the called endpoint. These routes are represented by the **Route Record** type.

Finally, there is also a replication mechanism that defines how the data contained by one registry should be replicated to another registry (interface 2). This feature is mainly used to provide high availability of the registry information (e.g. in case of node failure).

V. THE DATA MODEL

In Figure 4, we present a conceptual diagram representing the different elements defined in the data model and the relationships between them. We deliberately omitted the attributes for each element since we are only presenting a high level introduction of the framework (more details can be found in [7] and [8]).

![SPPF Data Model](image)

Fig. 4. SPPF Data Model

A public identifier may be one of the following elements: a **single telephone number** represented by the **TN** type, a **telephone number prefix** represented by the **TNP** type, a **telephone number range** represented by the **TNR** type or a **routing number** represented by the **RN** type. The latter is an identifier very similar to a telephone number and is used in case of number portability to identify the switch to which the session should be redirected. The other types of identifiers except the TN type allow representing ranges of telephone numbers. A TNP represents a modulo-10 range of telephone numbers (e.g. 1-(212)-123) whereas a TNR gives the possibility to represent a more fine-grained range (e.g. from 1-(212)-123-0000 to 1-(212)-123-1111).

After defining what a public identifier is, we have to describe how the results to these queries are represented by SPPF. These results are generally routes. SPPF often calls these routes ingress SBE (Session Border Element) of a target SSP since they identify the SBE (which is an ingress or way in component) of the service provider associated to the called endpoint. These routes are represented by the **Route Record** type.

There is more than one type of routes, namely: **URI** for
Uniform Resource Identifier, NAPTR for Name Authority Pointer and NS for Name Server, where each one represents a different concept. NAPTR and NS are two types of DNS (Domain Name System) records. A querying organization may get the result to its query using successive DNS queries based on these records as following. If the result to the query is a NS record [11], the querying organization is asked to renew its query to the authoritative DNS server identified by the record. If the result to the query is a NAPTR record [12], the querying organization indicates directly (or through one or more other queries) the route needed to establish the session. A URI (e.g. a SIP URI) is an element that represents an identity. For example, a SIP URI may be associated to a telephone number and a session may be established to this URI using SIP.

In order to avoid defining associations between each public identifier/route record pair, SPPF provides grouping facilities. It defines two other types for grouping public identifiers and route records: destination groups and route groups. A destination group is a set of public identifiers that share common routes. A route group is a set of routes that are generally associated together to public identifiers. Finally, destination groups and route groups are associated to form valuable information that may be used to establish sessions. Indeed, a querying organization generally wants to know the routes associated to a specific public identifier. The first step would consist of finding all the destination groups that contain this public identifier, then to get all the route groups associated to these destination groups. A possible use case for using multiple route groups to the same destination can be for example to categorize routes ordering to a quality criterion. For example, one service provider would be interested to separate routes that may all be used to establish a session to a certain endpoint, in two categories that would describe good QoS (Quality of Service) and excellent QoS routes. Then, it may share good QoS routes to one peering organization and excellent QoS routes to another peering organization, by using different pricing strategies.

VI. SETTING PEERING POLICIES INTERACTIVELY

The peering feature provided by SPPF is represented in Figure 4. It involves only two elements: route groups and organizations. A registrant organization that owns some routing information may want to share it with another organization. As explained in section V, the latter plays the role of peering organization. But, how does this registrant organization declare its willingness to share this information? SPPF provides an offering mechanism allowing interactive peering between organizations.

Indeed, a specific type is dedicated to represent this sort of an offer: Route Group Offer\(^1\). As can be deduced from the name, the element that may be offered is the route group. By offering a route group to a peering organization, the registrant organization declares that it allows the other organization to receive the routes associated to the route group, in response to its queries.

\(^1\)This type doesn’t appear in Figure 4 since the latter only represents the static structure of the data.

The peering organization to which the route group had been offered has two possibilities: it may accept the offer or decline it. If it accepts the offer, all the public identifiers associated to routes through the route group may now be resolved by this peering organization. If it declines the offer, the routes remain inaccessible to the peering organization. To illustrate this offering concept, we present the following example. Assume that service provider A has provisioned the registry with its routing information. For example, it has added a single telephone number 1-(212)-123-4567 to the registry and included it in a destination group DG-1. Then, it has added a NAPTR record to the registry and included it in a route group RG-1. Finally, it associated DG-1 to RG-1 (i.e. routes included in RG-1 may be used to establish a session to the endpoints included in DG-1). Let’s assume now service provider A decided (e.g. after a commercial agreement) to share this route group to service provider B. It then sends a route group offer identifying the involved route group RG-1, to service provider B. The latter has the choice to accept or reject it. Let’s assume service provider B accepted the offer. It then returns to service provider A an accept message, indicating it accepts the offer. After receiving the accept message, service provider A adds a peering policy to the registry by associating the route group to the organization representing service provider B. At this point, the route group is shared. If service provider B needs to establish a call to the endpoint identified by the telephone number 1-(212)-123-4567, it may query the registry (or generally its LDR) and would get as a result the routes associated to RG-1. In this case, the associated NAPTR record would be returned as result. In figure 5 we show a sequence diagram representing the main scenario of the offer mechanism.

VII. SOURCE BASED ROUTING

As explained in the previous section, a registrant organization may send a route group offer to another organization. It then proposes the peering organization to access the routes associated to the route group. This is referred to source based routing since specific routes are shared to a specific organization that would send queries on this information (i.e. it is the source of these queries). But what if we want to set more specific routing rules? For instance, what if we wish...
to share a route group to a subset of a specific organization (e.g. only when it queries the registry from a specific IP)? The framework includes an attribute named sourceIden in the route group element. It consists of a set of regular expressions that are matched at resolution time (i.e. during resolution of a query sent by a peering organization) against some of the client attributes. These attributes may be: the IP address of the client, the root domain name(s) of the client and/or the calling party URI. If at least one of the regular expressions matches successfully, the routes associated to the concerned route group may be returned as results of the query. Again, let’s illustrate with a sample scenario. We will follow the same example presented in section VI. Assume that service provider B has two SIP signaling systems. Each one of these systems has one interface with a specific IP address: the first one is identified by the IP address 11.11.11.11 and the second one is identified by the IP address 22.22.22.22. Let’s assume also that service provider A has set the sourceIden attribute of RG-1 with the regular expression “11.11.11.11” to match against the client IP address. In this case, only the SIP signaling system of service provider B identified by the 11.11.11.11 IP address would be authorized to access the routes associated to the route group RG-1 (i.e. the NAPTR record). If the other system sends the same query, the regular expression will not be successfully matched against the 22.22.22.22 IP address and the system would not let it accessing this same NAPTR record. A possible use case for this can be when a registrant organization wishes to return specific routes according to the querying component (of a peering organization) identity. For example, a registrant organization could sell access to information per component. In this case, the peering organization would buy two licenses to access certain information and would provide the identity of each component to the registrant organization. The latter has just to set the sourceIden attribute of the shared route group to enable the feature.

VIII. EGRESS ROUTES

Egress routes are used to support a feature for the benefit of peering organizations. An egress route identifies the egress SBE of a source SSP. Figure 6 illustrates the scenario.

A peering organization (e.g. SSP B in the figure) may have more than one egress path (e.g. through SBE 1 or SBE 2 in the figure) toward a certain route (e.g. ingress SBE 4 of the target SIP service provider A). In this case, a peering organization would want to customize the route to choose the egress SBE to use for this route. A good way to implement this feature is using regular expressions that rewrite the routes returned as results of peering organization’s queries. Let’s introduce this concept by illustrating this scenario according to Figure 6. Let’s assume we have two SIP service providers, A and B. Provider A, the registrant of certain routing information, shares it with B (peering organization). The shared information is the route that identifies SBE 4. But, service provider B has two SBE (SBE 1 and SBE 2 in the figure). SSP B may want to prioritize all the traffic sent to the route identifying SBE 4, through SBE-1. SPPF gives to SSP B the possibility to store this rule in the registry. The peering organization only needs to add an egress route to the registry and to associate it to the concerned route. The egress route consists of a regular expression that prepends to the route the identifier of SBE 1. In this case, each time SSP B gets as result to its query the route corresponding to SBE 4, it will know that he has to establish the session through SBE 1. This feature may be very useful for a service provider, especially to enable load sharing over its different SBE.

IX. RESEARCH HIGHLIGHTS FOR STANDARDIZATION

A. Issues Arising from Non-Standardization

As mentioned above, SPPF defines only the interface used by a client to provision the registry. The two other interfaces, i.e. interface 2 (registry-LDR) and 3 (registry-registry) of Figure 4, are currently not defined. We present in the following why we believe it is important to standardize these interfaces and what are the challenging interoperability issues arising from the current lack of standardization on these interfaces. The registry-registry interface may not be necessary if the two registries are maintained by the same third-party organization. Indeed, the point of SPPF is to centralize the information, not to distribute it. But, for the scenario where the two registries are maintained by two different third-party organizations, centralization is not possible since each organization holds its own managed data. Under some circumstances, however, it would be very useful and profitable for these two organizations to share the information present in their registry. Until now, we spoke about interconnection between service providers by a third-party organization. The same idea may be applied to third-party organizations. But, it is not necessary to interconnect these organizations by another super third-party organization. The only requirement would be to define a standardized interface to enable interoperable communication between registries of different organizations. We think this kind of architecture may be very profitable and would enable to create a super network of organizations that will share the session information they maintain. Each organization may define its own protocol to communicate with its registries but this solution will tend to create many silos of organizations that use incompatible solutions.

The same problem appears with the registry-LDR interface although this interface belongs to a different level. It concerns the distribution mechanism of information present in the registry to the clients local cache (LDR). Standardizing this interface would allow better interoperability between the client’s network components and the registries maintained by third-party organizations.
B. Proposal for Improving Operation

We propose here to add an important functionality to the standard that could allow better overall performance. The solution proposed emerged from an initial research idea arising from collaboration of research theoretical ideas and the experience from the industry. Although the peering data offering mechanism described in section VI may be useful in some situations, a more common use case is that a given organization wants to request a registrant organization to share some desired session information. This is the inverse of the offering mechanism currently being standardized. The following example illustrates a real scenario. Every day, different network based issues may present themselves when users on one network attempt calls to users of other networks. For example, the route used to establish the session may be broken or may have a poor quality rendering it unusable. These unsuccessful calls are audited and actions may be taken by network operators to improve the service. One possible action would be to look for an alternative route to reach the destination. In this scenario, were the extension we are proposing available, a peering organization (i.e. the one that is served by the registry) could potentially initiate a request for additional routes to the destination from other members of the registry. This mechanism may be introduced in two steps. The first step consists of giving the possibility to every registered organization to retrieve the names and registrant organizations of the SED groups that serve a particular public identifier. The requesting organization would then filter the retrieved SED groups according to some private rules (e.g. excluding a SED group owned by an organization that the requesting organization does not want to peer with). This first part of the mechanism allows an organization to discover the organizations that own the desired session establishment data (i.e. routes). The second step consists of giving the possibility to every registered organization to request a registrant organization to share one or more SED groups. In this case, this is the registrant organization that will decide to accept or reject the sharing request. If the registrant organization accepts the request, the requesting organization will automatically be added to the peering organization list of the shared SED group.

C. The Importance of Choice of Terminology in the Standard

By simply changing terms used in the standard to define data types we can broaden the impact of the standard on new use cases. The current draft introduces the egress route concept specifically for the scenario involving multiple egress SBE. Using an egress route, a peering organization (originating SSP) has the possibility of customizing the ingress route so that it can select the egress SBE to use to exit the originating network. This customization is only possible for the peering organizations associated to a SED group. But, it is also common that a registrant organization wants to customize its shared routes taking into account the party asking for the routes. For example, part of the DNS record returned as response may depend on the peering organization that asked for it. Clearly, this is not possible according to the current model. The solution we propose consists of abstracting the egress route construct, replacing it by a more generic rewrite rule construct. This is justified by the fact this construct only includes information used to rewrite an ingress route.

Using our proposal, it should be possible for peering organizations and registrant organizations to associate rewrite rules to ingress routes. On the one hand, a peering organization would use this construct to add the egress SBE to use when exiting its network to the ingress route - this is the original use case proposed by SPPF. On the other hand, a registrant organization would use this construct to add some source specific information to the ingress route. By just abstracting the construct, the original functionality is preserved and the new useful one is added without difficult modifications of the model.

D. Scalability Issues of the Standard and Possible Improvements

The operations defined by SPPF only address the need to add few records to the registry in a single transaction. But, registrant organizations often need to add a huge number of records in the registry (perhaps millions or even tens of millions of records for an initial provisioning of the registry). Provisioning of destination groups, SED groups and SED may not be considered as batch operations since these operations are often performed interactively. But, provisioning of public identifiers is definitely not an interactive operation. Every day, modifications related to public identifiers (e.g. new numbers, ported numbers) have to be persisted to maintain the database in a correct state. It is clear that the communication model defined by SPPF is not sufficient due to its synchronous nature. A client cannot wait the transaction to finish (which may take a long time) in a synchronous manner.

One possible solution is the introduction of an asynchronous model using notifications when bulk transactions are done. When HTTP is used as application protocol, this task is more difficult since HTTP follows a request-response model. We thought it is important to raise this point in this paper since it is a lack of a very important and useful feature.

X. CONCLUSION

In this paper we presented an overview of the data model and flow defined by the SPPF framework. SPPF is a very useful framework for the VoIP industry because it solves an interoperability issue that has occurs in recent years due to VoIP service providers sharing information actively to provide reliable access to their services. We have shown that although the SPPF framework provides the main functionality required by this type of service, i.e. routing, collaboration with researchers shows there are still some important open issues.

ACKNOWLEDGMENT

The work on this project would not have been possible without the support of many people. The authors wish to express their gratitude to the members of the DRINKS working group, including Jeremy Barkan who tragically passed away during the project. Finally, we would also like to thank Dean Willis and Syed W. Ali for their assistance and huge help during the study.
REFERENCES


