ECRIT
Internet-Draft
Expires: February 8, 2007

H. Schulzrinne
Columbia U.

August 7, 2006

Location-to-URL Mapping Architecture and Framework
draft-ietf-ecrit-mapping-arch-00

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with Section 6 of BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on February 8, 2007.

Copyright Notice

Copyright (C) The Internet Society (2006).

Abstract

This document describes an architecture for a global, scalable, resilient and administratively distributed system for mapping geographic location information to URLs. The architecture generalizes well-known approaches found in hierarchical lookup systems such as DNS. The architecture does not depend on using a specific protocol, but does require that protocols can summarize the coverage region of a node.

Table of Contents

1. The Mapping Problem

One of the central problems of providing emergency services to Internet systems is to map geographic location to a set of emergency services, represented by PSAPs, that can provide assistance for that particular location. This is a mapping problem, where a geographic location is translated into a set of URIs that allow the Internet system to contact an appropriate network entity. Other services may also find such location-to-URI mappings of use.

The overall emergency calling architecture separates mapping from placing calls or otherwise invoking the service, so the same mechanism can be used to verify that a mapping exists ("address validation") or to obtain test service URIs.

Mapping locations to URIs describing services requires a distributed, scalable and highly resilient infrastructure. Authoritative knowledge about such mappings is distributed among a large number of autonomous entities that may have no direct knowledge of each other. In this document, we describe an architecture for such a global service. It allows significant freedom to combine and split functionality among actual servers and imposes few requirements as to who should operate particular services.

Besides determining the PSAP URI, end systems also need to determine the local emergency dial strings. As discussed in Appendix A, the architecture described here can also address that problem.

The architecture described below does not depend on a particular mapping protocol, but naturally assumes that such protocols provide certain features, such as the ability to discover the coverage region of tree nodes. In this introduction, we describe the four participants in the system at a high level. Each role will later be introduced in more detail.

2. Terminology

In this document, the key words "MUST", "MUSTNOT", "REQUIRED", "SHALL", "SHALLNOT", "SHOULD", "SHOULDNOT", "RECOMMENDED", "MAY", and "OPTIONAL" are to be interpreted as described in RFC 2119 [1] and indicate requirement levels for compliant implementations.

3. Definitions

[Note: The terminology below is still evolving and needs refinement.]

In addition to the terms defined in [11], this document uses the

following terms to describe LoST:

authoritative mapping server (AMS): Resolver that can provide the authoritative answer to a particular set of queries, e.g., covering a set of PIDF-LO civic labels or a particular region described by a geometric shape. In some (rare) cases of territorial disputes, two resolvers may be authoritative for the same region. An AMS may redirect or forward a query to other AMS within the tree.

caching resolver: A caching resolver is contacted by a seeker, consults a forest mapping server and then resolves the query using an appropriate tree.

child: A child is a resolver that is authoritative for a subregion of a particular server. A child can in turn be parent.

cluster: A cluster is a group of resolver (servers) that all share the same mapping information and return the same results for queries. Clusters provide redundancy and share query load. Clusters are fully-meshed, i.e., they all exchange updates with each other.

complete: A civic mapping region is considered complete if it covers a set of hierarchical labels in its entirety, i.e., there is no other resolver that covers parts of the same region. (A complete mapping may have children that cover strict subsets of this region.) For example, a region spanning the whole country is complete, but a region spanning only some of the streets in a city is not.

forest guide: A forest guide has knowledge of the coverage region of all trees.

mapping: A mapping is a short-hand for 'mapping from a location object to one or more URLs describing either another mapping server or the desired PSAP URLs.'

parent: A mapping server that covers the region of all of its children. A mapping server without a parent is a root resolver.

peer: A resolver maintains associations other resolvers, called peers. Peers synchronize their region maps.

seeker: The resolver, ESRP or end system requesting a mapping.

region map: A data object describing a contiguous area covered by a resolver, either as a subset of a civic address or a geometric object.

root region map: A data object describing a contiguous area covered by a resolver, with no parent map.

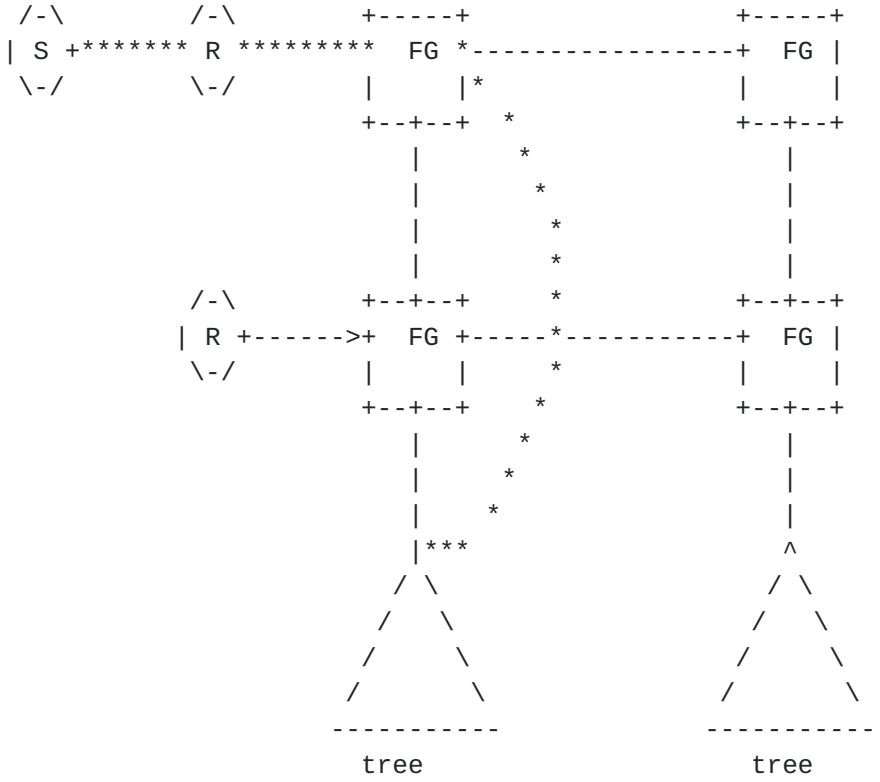
resolver: The server providing (part of) the mapping service. Resolvers cooperate to offer the mapping service to seekers.

tree: A tree consists of a hierarchy of authoritative mapping servers. Each tree exports its coverage region to the forest mapping servers.

4. Introduction

4.1 Overview of Operation

In short, end users of the mechanism, called seekers, contact resolvers that cache query results and know one or more "forest guides". Forest guides know the coverage region of trees and direct queries to the node at the top of the appropriate tree. Trees maintain the authoritative mapping information. Figure 1 shows the interaction of the components.



Architecture diagram, showing seekers (S), resolvers (R), forest guides (FG) and trees. The star (*) line indicates the flow of the query and responses in recursive mode.

Figure 1

4.2 Seekers: The Users of the Mapping System

Clients desiring mappings are known as seekers. Thus, seekers are the end users of the mapping information. Examples of such clients include SIP proxy servers or SIP end systems wishing to place an emergency call. Seekers provide location information describing a small geographic area and obtain one or more URIs describing the

service. Seekers may need to obtain this information in several steps, i.e., they may obtain pointers to intermediate servers that lead them closer to the final mapping. Seekers MAY cache query results for later use, but otherwise have no obligations to other entities in the system.

4.3 Trees: Authoritative Knowledge

The architecture assumes that authoritative knowledge about the mapping data is distributed among many independent administrative entities, but clients (seekers) needing the information may potentially need to find out mapping about any spot on earth. (Extensions to extra-terrestrial applications are left for future exploration.) Different types of services may divide responsibility differently and are independent of each other. Each node participating in the system has authoritative knowledge about mappings within its coverage region, typically, but not necessarily, a contiguous geographic region described by a polygon in geospatial coordinates or a set of civic address descriptors (e.g., "country = DE, A1 = Bavaria"). These coverage regions may be aligned with political boundaries, but that is not required. In most cases, to avoid confusion, only one node is responsible for a particular geographic or civic location, but the system can also deal with cases where coverage regions overlap.

The architecture assumes that knowledge about mappings is hierarchical, represented as a tree. Each tree node knows the coverage region of its children and sends queries to the appropriate server "down" the tree. There are no assumptions about the coverage region of a tree. For example, a tree could cover a single city, or a state/province or a whole country. Nodes within a tree need to loosely coordinate their operation, but they do not need to be operated by the same administrator.

Thus, the mapping function for the world is divided among trees. The collection of trees may not cover the whole world and trees are added and removed as the organization of mapping data changes. We call the collection of trees a forest. There is no limit on the number of trees within the forest, but the author pictures that the number of trees will likely be somewhere between a few hundred and a few thousand. The lower estimate would apply if each country operates one tree. We assume that tree coverage information changes relatively slowly, on the order of a few changes per year per tree, although the system imposes no specific threshold. (To be sure, information within a tree is likely to change much more frequently.)

4.4 Forest Guides: Finding the Right Tree

Unfortunately, just having trees covering various regions of the world is not sufficient as a client of the mapping protocol would not generally be able to keep track of all the trees in the forest. To facilitate orientation among the trees, we introduce a "forest guide". It is a server that keeps track of the coverage regions of the trees. For scalability and reliability, there will need to be a large number of forest guides, all providing the same information. A seeker can contact any forest guide and will then be directed to the right tree or, rarely, set of trees.

4.5 Resolvers: Finding Forest Guides and Caching Data

A seeker can contact a forest guide directly, but may not be able to easily locate such a guide. In addition, seekers in the same geographic area may already have asked the same question. Thus, it makes sense to introduce another entity, a resolver, that knows how to contact one or more forest guides and caches earlier queries to accelerate the response to mapping queries.

4.6 Minimal System Architecture

It is possible to build a functioning system consisting only of seekers and resolvers if these resolvers have other means of obtaining mapping data. For example, a company acting as a mapping service provider could collect mapping records manually and make them available to their customers through the resolver. While feasible as a starting point, such an architecture is unlikely to scale globally. Among other problems, it becomes very hard for providers of authoritative data to ensure that all such providers have up-to-date information. If new trees are set up, they would somehow make themselves known to these providers. Such a mechanism would be similar to the old "hosts.txt" mechanism for distributing host information in the early Internet.

5. Seeker

Seekers are consumers of mapping data and originate queries. Seekers do not answer queries. They contact either forest guides or resolvers to find the appropriate tree that can authoritatively answer their questions. As noted in the introduction, seekers can be end systems or call routing entities such as SIP proxy servers.

Seekers need to be able to identify appropriate resolvers. The mechanism for providing seekers with that information is likely to differ depending on who operates the resolvers. For example, if the voice service provider operates the resolver, it might include the

location of the resolver in the SIP configuration information it distributes to its user agents. An Internet access provider might provide a pointer to a resolver via DHCP. In an ad-hoc or zero-configuration environment, appropriate service directories may advertise resolvers.

For emergency calling, seekers could issue queries at boot time, periodically when cached information expires or only when placing an emergency call. It is probably unnecessary to continuously update mapping information for seekers representing a small user population, e.g., a single phone or residential SIP proxy.

Like other entities in the system, seekers can cache responses. This is particularly useful if the response describes the result for a region, not just a point. For example, for mobile nodes, seekers would only have to update their resolution results when they leave the coverage area of a PSAP and can avoid polling for this information. This will likely be of particular benefit for seekers representing a large user population, such as the outbound proxy in a corporate network. For example, rather than having to query separately for each cubicle, information provided by the authoritative node may indicate that the whole campus is covered by the same PSAP.

6. Resolver

Resolvers mediate between seekers and forest guides. Their primary role is to avoid having seekers find forest guides on their own. Unlike forest guides, resolvers do not store worldwide coverage maps, but they may cache regions returned as part of query results.

As noted earlier, seekers can contact forest guides directly. From a protocol perspective, a resolver acts in the same way as a seeker, except that it knows one or more forest guide.

ISPs or VSPs would include the address of a suitable resolver in their configuration information, i.e., in SIP configuration for a VSP or DHCP for an ISP. Resolvers are manually configured with the name of one or more forest guides.

7. Trees

7.1 Basic Operation

As noted in the introduction, trees are the authoritative source of mapping data. Each tree can map a location described by civic and geographic coordinates for one type of service (such as 'police' or 'fire'), although nothing prevents re-using the same tree for

multiple different services. The collection of trees for one service is known as a forest.

The tree architecture is roughly similar to the domain name system (DNS), except that delegation is not by label, but rather by region. (Naturally, DNS does not have the notion of forest guides.) One can also draw analogies to LDAP, when deployed in a distributed fashion.

Tree nodes maintain two types of information, namely coverage regions and mappings. Coverage regions describe the region served by a child node in the tree and point to a child node for further resolution. Mappings contain an actual service URI leading to a PSAP or another signaling server representing a group of PSAPs. To provide redundancy, a mapping entry can also contain multiple URLs, indicating both primary and backup services. For example, it might contain both a local PSAP and a state agency that takes over if the local PSAP fails. Unlike DNS NAPTR and SRV facilities, these can survive DNS failures and thus provide an additional, complementary mechanism to introduce redundancy services.

Leaf nodes, i.e., nodes without children, only maintain mappings, while tree nodes above the leaf nodes only maintain coverage regions. An example of a leaf node entry is shown below, indicating how queries for three towns are directed to different PSAPs.

```
country   A1 A2         A3        resource
US        NJ Bergen     Leonia    sip:psap@leonianj.gov
US        NJ Bergen     Fort Lee  sip:emergency@fortleenj.org
US        NJ Bergen     Teaneck   sip:police@teanecknjgov.org
....
```

Coverage regions are described by sets of polygons enclosing contiguous geographic areas or by descriptors enumerating groups of civic locations.

For example, a state-level tree node for New Jersey in the United States may contain the following coverage region entries, indicating that any query matching a location in Bergen County, for example, would be redirected or forwarded to the node located at bergen.nj.example.org. There is no requirement that all child nodes cover the same level within the civic hierarchy. As an example, in the table below, the city of Newark has decided to be listed directly within the state node, rather than through the county. Longest-match rules allow partial coverage, so that for queries for all other towns within Essex county would be directed to the county node for further resolution. In the example below, we use a fictitious URL scheme, 'rp', to identify the resolution protocol. In actual use, each entry would have one or more URLs pointing to resolution servers for

different protocols. Each entry may also contain multiple URLs for the same protocol to indicate primary and backup services.

```
C  A1 A2         A3     resource
US NJ Atlantic   *      rp://atlantic.nj.example.org/sos
US NJ Bergen     *      rp://bergen.nj.example.org/sos
US NJ Monmouth   *      rp://monmouth.nj.example.org/sos
US NJ Essex      *      rp://essex.nj.example.org/sos
US NJ Essex      Newark rp//newark.example.com/sos
....
```

Thus, there is no substantial difference between coverage region and mapping data. The only difference is that coverage regions return mapping protocol URLs, while mapping entries contain PSAP URLs. Mapping entries may be specific down to the house or floor level or may only contain street-level information. For example, in the United States, civic mapping data is generally limited to address ranges ("MSAG data"), so initial mapping databases may only contain street-level information.

To automate operations, a suitable mapping protocol would thus need to be able to query nodes for their coverage region. In the example above, the state-run node would query the county nodes and thus aggregate the coverage data. Conversely, nodes could also contact their parent nodes. There is some benefit of child nodes contacting their parents, as this allows changes in coverage region to propagate quickly up the tree.

7.2 Answering Queries

Within a tree, the basic operation is straightforward: A query reaches the root of the tree. That node determines which coverage region matches that request and forwards the request to the URL indicated in the coverage region record, returning a response to the querier when it in turns receives an answer (recursion). Alternatively, the node returns the URL of that child node to the querier. This process applies to each node, i.e., a node does not need to know whether the original query came from a parent node, a seeker, a forest guide or a resolver.

For efficiency, a node MAY return region information instead of a point answer. Thus, instead of returning that a particular geospatial coordinate maps to a service or mapping URL, it MAY return a polygon indicating the region for which this answer would be returned, along with expiration time (time-to-live) information. The querying node can then cache this information for future use.

For civic coordinates, trees may not include individual entries for

each floor, house number or street. To avoid giving the wrong indication that a particular location has been found valid, the protocol SHOULD return an indication which parts of the location information have actually been mapped.

7.3 Overlapping Coverage Regions

In some cases, coverage regions may overlap, either because there is a dispute as to who handles a particular geographic region or, more likely, since the resolution of the coverage map may not be sufficiently high. For example, a node may "shave some corners" off its polygon, so that its coverage region appears to overlap with its geographic neighbor. For civic coordinates, houses on the same street may be served by different PSAPs. The mapping mechanism needs to work even if a coverage map is imprecise or if there are disputes about coverage.

The solution for overlapping coverage regions is relatively simple. If a query matches multiple coverage regions, the node returns all URLs, in redirection mode, or queries both children, if in recursive mode. If the overlapping coverage is caused by imprecise coverage maps, only one will return a result and the others will return an error indication. If the particular location is disputed territory, the response will contain all answers, leaving it to the querier to choose the preferred solution or trying to contact all services in turn.

7.4 Scaling and Reliability

Since they provide authoritative information, tree nodes need to be highly reliable. Thus, while this document refers to tree nodes as logical entities within the tree, an actual implementation would likely replicate node information across several servers, forming a cluster. Each such node would have the same information. Standard techniques such as DNS SRV records can be used to select one of the servers. Replication within the cluster can use any suitable protocol mechanism, but a standardized incremental update mechanism makes it easier to spread those nodes across multiple independently-administered locations. The techniques developed for meshed SLP [7] are applicable here.

8. Forest Guides

Forest guides distribute records describing the coverage region for trees. For authenticity, the records are digitally signed. They are used by resolvers and possibly seekers to find the appropriate tree for a particular area. All forest guides should have consistent information, i.e., a collection of all the coverage regions of all

the trees. A tree node at the top of a tree can contact any forest guide and inject new coverage region information into the system. One would expect that each tree announces its coverage to more than one forest guide. Each forest guide peers with one or more other guides and distributes new coverage region announcements to all other guides.

Forest guides fulfill a similar role to root servers in DNS. However, their number is likely to be larger, possibly counted in hundreds. They distribute information, signed for authenticity, offered by trees.

Forest guides can, in principle, be operated by anybody, including voice service providers, Internet access providers, dedicated services providers and enterprises.

As in routing, peering with other forest guides implies a certain amount of trust in the peer. Thus, peering is likely to require some negotiation between the administering parties concerned, rather than automatic configuration. The mechanism itself does not imply a particular policy as to who gets to advertise a particular coverage region.

9. Security Considerations

The architecture addresses the following security issues, usually through the underlying transport security associations:

Server impersonation: Seekers, cluster members and peers can assure themselves of the identity of the remote party by using the facilities in the underlying channel security mechanism, such as TLS.

Query or query result corruption: To avoid that an attacker can modify the query or its result, the architecture RECOMMENDS the use of channel security, such as TLS. Results SHOULD also be digitally signed, e.g., using XML digital signatures. Note, however, that simple origin assertion may not provide the end system with enough useful information as it has no good way of knowing that a particular signer is authorized to represent a particular geographic area. It might be necessary that certain well-known Certificate Authorities (CAs) vet sources of mapping information and provide special certificates for that purpose. In many cases, a seeker will have to trust its local resolver to vet information for trustworthiness; in turn, the resolver may rely on trusted forest guides to steer it to the correct information.

Region corruption: To avoid that a third party or an untrustworthy member of a server population introduces a region map that it is not authorized for, any node introducing a new region map MUST sign the object by encapsulating the data into a CMS wrapper. A recipient MUST verify, through a local policy mechanism, that the signing entity is indeed authorized to speak for that region. Determining who can speak for a particular region is inherently difficult unless there is a small set of authorizing entities that participants in the mapping architecture can trust. Receiving systems should be particularly suspicious if an existing region map is replaced with a new one with a new mapping address. In many cases, trust will be mediated: A seeker will have a trust relationship with a resolver. The resolver, in turn, will contact a trusted forest guide.

Additional threats that need to be addressed by operational measures include denial-of-service attacks.

10. IANA Considerations

Since this document describes an architecture, not a protocol, it does not ask IANA to register any protocol constants.

11. References

11.1 Normative References

[1] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[2] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, March 1997.

[3] Gulbrandsen, A., Vixie, P., and L. Esibov, "A DNS RR for specifying the location of services (DNS SRV)", RFC 2782, February 2000.

[4] Peterson, J., "A Presence-based GEOPRIV Location Object Format", RFC 4119, December 2005.

[5] Rosen, B., "Dialstring parameter for the Session Initiation Protocol Uniform Resource Identifier", draft-rosen-iptel-dialstring-04 (work in progress), June 2006.

11.2 Informative References

[6] Guttman, E., Perkins, C., Veizades, J., and M. Day, "Service Location Protocol, Version 2", RFC 2608, June 1999.

[7] Zhao, W., Schulzrinne, H., and E. Guttman, "Mesh-enhanced Service Location Protocol (mSLP)", RFC 3528, April 2003.

[8] Newton, A. and M. Sanz, "IRIS: The Internet Registry Information Service (IRIS) Core Protocol", RFC 3981, January 2005.

[9] Krochmal, M. and S. Cheshire, "DNS-Based Service Discovery", draft-cheshire-dnsext-dns-sd-03 (work in progress), July 2005.

[10] Petrie, D., "A Framework for Session Initiation Protocol User Agent Profile Delivery", draft-ietf-sipping-config-framework-08 (work in progress), March 2006.

[11] Schulzrinne, H. and R. Marshall, "Requirements for Emergency Context Resolution with Internet Technologies", draft-ietf-ecrit-requirements-10 (work in progress), June 2006.

Author's Address

Henning Schulzrinne
Columbia University
Department of Computer Science
450 Computer Science Building
New York, NY 10027
US

Phone: +1 212 939 7004
Email: hgs+ecrit@cs.columbia.edu
URI: http://www.cs.columbia.edu

Appendix A. Configuring Emergency Dial Strings

The section below is not directly related to the problem of determining service location, but is an instance of the more generic problem solved by this architecture, namely mapping location information to URLs.

For the foreseeable future, some user devices and software will emulate the user interface of a telephone, i.e., the only way to enter call address information is via a 12-button keypad with digits and the asterisk and hash symbol. Also, emergency numbers are likely to be used until essentially all communication devices feature IP connectivity and an alphanumeric keyboard. Unfortunately, more than 60 emergency numbers are in use throughout the world, with many of those numbers serving non-emergency purposes elsewhere, e.g., identifying repair or directory services. Countries also

occasionally change their emergency numbers to conform to regional agreements. An example is the introduction of 112 for countries in Europe.

Thus, a system that allows devices to be used internationally to place emergency calls needs to allow devices to discover emergency numbers automatically. In the system proposed, these numbers are strictly of local significance and are generally not visible in call signaling messages.

For simplicity of presentation, this section assumes that emergency numbers are valid throughout a country, rather than, say, be restricted to a particular city. This appears likely to be true in countries that have sufficiently advanced infrastructure to contemplate deploying IP-based emergency calling solutions. In addition, the solution proposed also works if certain countries do not use a national emergency number. There is no requirement that a country uses a single emergency number for all emergency services, such as fire, police, or rescue.

For the best user experience, systems should be able to discover two sets of numbers, namely those used in the user's home country and in the country the user is currently visiting. The user is most likely to remember the former, but a companion borrowing a device in an emergency may only know the local emergency numbers.

Determining home and local emergency numbers is a configuration problem, but unfortunately, existing configuration mechanisms are ill-suited for this purpose. For example, a DHCP server might be able to provide the local emergency number, but not the home numbers. When virtual private networks (VPNs) are used, even DHCP may provide numbers of uncertain origin, as a user may contact to the home network or some local branch office of the corporate network. Similarly, SIP configuration would be able to provide the numbers valid at the location of the SIP service provider, but even a SIP service provider with national footprint may serve customers that are visiting any number of other countries.

Since dial strings are represented as URLs [5], the problem of determining local and home emergency numbers is a problem of mapping locations to a set of URLs, i.e., exactly the problem that the mapping architecture is solving already.

The mapping operation is almost exactly the same as for determining the emergency service URL. The only difference is that if a seeker knows the civic location at least to the country level, it will use a query where the PIDF-LO only includes the country code. If it only knows its geospatial location, it has to include that longitude and

latitude. The seeker uses the service identifiers "dialstring.sos", "dialstring.sos.fire", etc. The resolver returns the appropriate set of URLs and, if a geospatial location was used in the query, the current region map for the country.

Within the mapping system, emergency calling regions are global information, i.e., they are distributed using the forest guide replication mechanism described earlier. Thus, every forest guide has access to all region mappings. This makes it possible that a seeker can ask any resolver for this information, reducing the privacy threat of revealing its location outside of an emergency call. The privacy threat is further reduced by the long-lived nature of the information, i.e., in almost all cases, the seeker will have already cached the national boundary information or country information on its first visit to the country.

Appendix B. Acknowledgments

Jong Yul Kim, Andrew Newton, Richard Stastny, Hannes Tschofenig provided helpful comments.

Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Disclaimer of Validity

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Copyright Statement

Copyright (C) The Internet Society (2006). This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.