

INTERNET-DRAFT
August 1, 2001
Expires February 1, 2002

Y. Arrouye
V. Parikh
N. Popp
RealNames Corp.

Keyword Lookup Systems As a Class of Naming Systems
[draft-arrouye-cls-00.txt](#)

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of [Section 10 of RFC2026](#).

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/1id-abstracts.html>

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

Copyright Notice

Copyright (C) The Internet Society (2001). All Rights Reserved.

Abstract

This document emphasizes the convergence of thinking regarding Internet naming in the industry and the technical community.

There is strong consensus for establishing a layer above the current DNS to address some of its current limitations. At the same time, the community stresses the necessity for the new layer to go beyond the sole needs of the domain name system to realize an architecture capable of accommodating diverse services, with separate ownerships, different scopes and distinct operating models. In that context, interoperability is critical.

This Internet-Draft introduces critical requirements for supporting multiple namespaces, in particular the crucial need for discovering namespaces across service providers. Acknowledging the direction

opened by a reflection on the DNS [[DNSROLE](#), [DNSSEARCH](#)] as well as the need to support flexible naming for various services [[SLS](#)], we describe the Keywords systems as a unique class of Service Lookup Systems [[SLS](#)].

1. Introduction

During the course of the past few years, a number of companies have looked into providing an easier user interface for Web navigation than that of typing a URI into a browser's address line. Examples of such companies are RealNames (worldwide network), Netword (North America), Netpia (Korea), and 3721 (China).

All of these companies recognized the fact that URIs are unwieldy at best for users: they are hard to remember, due to their syntax, and as far as Web users are concerned, the only URIs that they commonly type into a browser are those forms that consist of the sole DNS name of the server they want to contact, eventually preceded by a valid protocol scheme. Therefore, identifiers like "www.ietf.org," "www.yahoo.com," and others, have replaced URIs in daily interaction with the Web.

Unfortunately, even these short names do not address users' needs. For example, while "www.ford.com" does belong to Ford Motor Corporation, it is "www.fordvehicles.com" that shows vehicles of the Ford brand. Not only is it unfortunate that people think of one name ("ford") and need to use another one ("fordvehicles"), but the DNS labels are not user-friendly names: "fordvehicles" has no space between the words, for example.

While companies have been very creative in fitting their brand names into DNS labels (as can be seen by the company "Bird & Bird" getting the DNS name "twobirds.com"), if one looks at the Web in general, there is a big disconnect between the DNS labels that are used to identify Web sites, and the actual names by which people refer to these sites, which are typically brand or product names.

This should not come as a surprise considering that the role of DNS is to map unique identifiers to network resources, not to give human friendly names to information or services. DNS has been abused to fulfill naming needs on the Web for a few years. It is actually a testimony to the strength of DNS that it has not crumbled under the additional requirements it has been put through.

User behavior and expectations are just one part of the problem that DNS was not designed to and cannot be expected to address. In addition, the Internet world is truly multilingual, bringing in a whole new set of names that are used to refer to information and

services. People in different parts of the world want to use their own languages and scripts to refer to things, on the Web, in e-mail, on the Internet. So do companies. A Chinese company does not want to offer its Chinese products to its Chinese customers via a URI in Roman Script.

Because DNS is the visible tip of the naming iceberg, the one that is already working and freely accessible (obtaining a domain name is pretty easy), people want to add capabilities to support DNS names that are truly multilingual, and the existence of the IETF Internationalized Domain Name (IDN) working group is a recognition of that desire.

Unfortunately, a number of limitations of today's domain names still apply to IDNs as they are being considered now. Amongst them, the inability to use a space to separate words, strong limitations in the length of labels, and the lack to carry enough expression power to handle the visual requirements of names in Persian, for example.

The technical community has begun to recognize that naming for humans is something that needs to be addressed outside of DNS, because DNS is a system of identifiers for network resources, not a system of easy-to-use names. Companies selling keywords and other friendly names have been the first to address this need. One only needs to be aware of the existence of John Klensin's drafts about the role of DNS and how to extend it by adding other layers [[DNSROLE](#), [DNSSEARCH](#)] to acknowledge that this view is becoming a well-regarded one.

Keywords systems, which are based on natural language names in native languages, were the first to turn the recognition about the paradigm shift that happened in the use of DNS names into a new product and protocol layer.

Their existence has fostered the creation of the Common Name Resolution Protocol [[CNRP](#)] in 1999. CNRP was an acknowledgment that users and applications wanted to use simple names to denote arbitrary resources on the Web. CNRP did not address the existence of many namespaces that were task-oriented.

SLS builds on this foundation. It recognizes that people primarily use services such as the DNS, the Web, or email, and that those services use names to denote information. Using SLS, and thus CNRP, one can get from human-friendly names to network or information resource identifiers for a number of applications.

[2. Keyword Systems and Lessons Learned](#)

The primary goal of a keyword system is to make it easy for users to

refer to things by their common, or human-friendly name in their local language. Keyword systems also take into account what users commonly expect - that the network will have information about them and use that context on their behalf to navigate them to the "best" destinations on the Web based on their query.

It is a fact that a given name may refer to different things: "Woolworth" is a dime store in the USA but a high-end clothing store in South Africa; there may be one "Joe's Pizza" per city, or even more; and trademarks of the same name, but different applicability fields, coexist in a given country. Users, however, are typically only interested in giving one name to get some information.

What RealNames learned from this is that while it is important for names to co-exist and be differentiated by facets(e.g. country, city, language, ...), it is also important that this differentiation is done transparently for the user wherever possible.

Let's look at the RealNames keyword system for example. A Keyword has many facets, amongst them a name, a country, a language, a description, a URI, and a service code. Service codes (also called content codes) differentiate between the "web" and "mobile phone web" for which content is typically formatted differently. Countries let people refer to a global brand such as "Coca-Cola" by using the same name in different places. Language addresses the needs of countries with various ethnic populations, or official languages, to cater to the language requirements of their population. Description is a purely informative property. Finally, through a URI, one can point to the exact physical address associated to a given name in a given context.

We've just used the word "context" and it is worth examining it. If names need to be associated to some meta-data to be useful and not fall into the flat namespace trap that the DNS has fallen into with gTLDs like .com, and, if users are going to expect satisfactory results using just names but not all the meta-data when using a keyword, then it is clear that the remaining meta-data must be supplied from the user's context.

As namespaces will become more specialized, allowing for more identical names to exist in different dimensions, the reliance on context will increase. As we have operated our Keyword system these past four years, we have found that obtaining more meta-data is a difficult task. It is a formidable chicken and egg situation. If users' applications and devices cannot supply the relevant data, then name buyers will not be willing to supply the appropriate ones when registering keywords. Keyword name buyers will only supply meta-data if they see a benefit. Benefits are derived from applications that

perform more efficient navigation based on meta-data. No specification, regardless of how robust it may be, can alter this dynamic by itself. This combination of meta-data requirements and reluctance to provide meta-data cannot be ignored as we lay down the foundation for a new standard.

At the same time, once a name buyer uses a Keyword system the importance of meta-data becomes clear and they are willing to provide appropriate meta-data to support the names they buy.

However, name buyers are generally unwilling to supply more than a name at the time of use of the system (primarily because they take the rest of context for granted), so the burden of supplying the contextual information falls solely on the application or device.

One key to resolving this problem is mining and using whatever context can be supplied. For example, as devices get smarter, they can also supply more context. For example, a Web browser typically knows about the user's language (from its language preferences, which then translate into an HTTP header for language negotiation), and can make a (risky) assumption about the country from the language. But a cellular phone nowadays can also know about the user's location to some reasonable accuracy. The existence of both devices also means that the kind of device is part of the available context.

Another lesson from the requirement for context is that not all the properties of a keyword are useful in order to use the keyword as a way to go get some information (through the URI associated to it). The URI, which is what one wants to get, is definitely not one of these properties. Also, in a system where keywords are associated to a description, one does not want to require the description in order to enable direct resolution.

In terms of supplying meta-data, the description associated with a keyword in the RealNames system is not necessary for lookups, while it is very useful for displaying entries in a directory of keywords. Some properties, like an industry category, are useful in both cases, but cannot be used today for navigation because of difficulty of establishing their value as part of a user's context.

We call unique key the set of properties that are required to match a keyword and get a single result from a lookup. These properties translate into required context if an application wants to offer direct navigation from a name to a destination through a keywords system. Unique keys may differ from one keywords system to another (for example, RealNames uses a few properties while Netword only uses a name today). Unique keys may, and will, change, depending on the sophistication of the devices that are used to access a given

service.

As unique keys become smarter and able to know more about the context of their uses, more discriminating namespaces can come into existence and proliferate. Today, given the simplicity of the devices we use, keywords systems use short unique keys for the emerging namespaces that Keywords systems are.

Because the unique key lets one access a record with other meta-data, it can be used not only to enable applications to fetch a URI and use it, but also to give one a place to store state.

3. The Importance of Multiple Namespaces Above DNS

A keyword system is an example of a namespace made up of records which consist of the unique keys and other helpful information. This namespace is dedicated to providing keyword-based natural language navigation. We believe that this is only one of many types of namespaces that will emerge as the Internet continues the significant transformation it is now undergoing.

Like many in the industry, we believe that the limitations of DNS will be best addressed by facilitating the emergence of many interoperable namespaces. These multiple namespaces will be largely distributed, with distinct owners and administrative rules. In fact, that has already occurred, to the good of Internet users and service providers. We believe the right course is to encourage this innovation, and that industry and the technical community should work together to define a set of standard protocols through which all of these namespaces can interoperate.

As the network continues its evolution, namespaces will offer different types of content and information to users and will be based on a continuum of business models. Although a namespace is likely to focus on one or a few types of network resources, it will be necessary for applications to interact with many namespaces at the same time in the course of a user session. Clearly, a flexible, lightweight protocol needs to define and facilitate interaction.

This interaction is likely to encompass at the minimum registration, resolution (lookup) and discovery (search) services. Application developers must be able to rely on a common set of protocols for interacting with namespaces both to reduce the burden of adding new services to the network and also to enable users to freely choose between a rich offering of providers.

Based on current industry development, it is not difficult to anticipate tomorrow's importance of companies, people, and products

namespaces whose services will be delivered to user applications through the network and across multiple devices ranging from desktops to PDAs and data enabled cell phones.

Many important instances of namespaces have already appeared in the last few years, and the fast-growing concept of Web services is likely to increase the reliance of user application on these namespaces as a core component. The increased tie and reliance between applications and namespaces accessible through the network exacerbates the need for open standards and interoperability.

For example, if ICQ was one of the first major community namespace, Firefly was the first company to identify the need for separate specialized namespaces. Passport and its Hailstorm incarnation will be a namespace for people as network end-points that will not only provide unique identity (email address) to people but also personal context storage, and presence detection. Namespaces for people as network end-points are likely to become fundamental elements of all user-centric applications. Namespaces will likely encompass many aspects of our user experience. Napster was the first P2P music namespace to profoundly impact the way users lookup and discover songs on the network and MusicNet is a more recent attempt to create a commercial music namespace that can be distributed through multiple applications.

Other namespaces will be business enablers. UDDI is an industry-wide attempt to create a namespace for small and medium companies to discover each other and conduct business across the Internet and Keywords systems like RealNames are specialized toward human-friendly access to companies, products and services. There will obviously be many more valuable namespaces in the future.

Since one cannot and does not want to anticipate any single one of them to dominate, it is important to facilitate the interaction of applications with multiple namespaces through a common sets of protocols accessible across multiple devices and operating systems.

4. Requirements Arising From Multiple Independent Namespaces

The growing proliferation of many independent namespaces places important constraints and requirements on the standard that will be created. We believe that some of these requirements may have been absent or minimized in previous proposals. Such critical constraints include:

- The definition of a mechanism for discovering Namespaces.
- The standardization of all core namespace services: resolution

(lookup), discovery (search) and registration.

- The need to simplify Klensin's layer 2 [[DNSROLE](#)] in order to facilitate the adoption of the standard by most client applications and service providers.
- The support for different rules of uniqueness and disambiguation across namespaces.

The first requirement is about enabling applications to discover all available Namespaces.

The second is about service interoperability. In a world of many namespaces, these requirements are straightforward (and the solutions non trivial).

The third requirement underlines the need to minimize the definition of the schema that Klensin's layer 2 services must all implement for interoperability. It is mainly driven by the need to simplify client implementations. As we impose more facets on this layer, we impose more burden on application developers and the user-interface (especially considering that as pointed out above, the user context carried by applications today is fairly minimal). This model cannot be successful in practice.

Application developers' adoption of the standard is obviously essential since the prospect of increased distribution will be the main driver for namespace owners to implement the standard and make their service interoperable.

However, we also recognize that from a service provider standpoint, the complexity of the Klensin's layer 2 schema is not a major impediment since facets within a query can simply be ignored. For example, all things being equal, a service that would not support category, for example, could return the same results whatever value of the category facet the user may specify. Obviously, one would expect that the ability to accurately match results to the query context passed by the client to be a strong service differentiator, hence a market force to drive providers to fully implement the required schema.

This last observation leads to the third point. Like the authors of [[SLS](#)], we do not think that it is possible to impose a single context of uniqueness to all namespaces (we define uniqueness of a namespace as the minimum set of facets required to uniquely identify a record within the namespace). We too, believe that the proper notion of uniqueness is tied to the problem space tackled by the namespace, and hence, will vary from one provider to another.

At the same time, we do not believe that it is possible to define a Klensin's layer 2 schema capable of providing all namespaces with sufficient context to ensure uniqueness and disambiguation for lookups.

Instead, we see a practical solution in the middle. We believe that some namespaces will require narrower context of uniqueness than the complete schema that they expose to an application, using the extra facets to facilitate disambiguation by end-users through an interactive process (e.g. a user typing "Joe" could be asked to disambiguate between the unique Keyword "Joe's seafood" in the restaurant category and the unique Keyword "Joe body repair shop" in the "body shop" category).

If this is true, defining the aforementioned layer as the ceiling to provide disambiguation for lookup is not very useful. Indeed, it is easy to envision that whichever standard set of facets we eventually agree upon for this layer, a new namespace will emerge with new requirements for uniqueness and disambiguation to prove us wrong.

In that case, we propose that it might be more appropriate to consider defining the sets of facets that all namespaces should support in order to provide interoperability, while at the same time enabling each namespace to advertise to the applications the exact set of facets that it really needs for uniqueness. The definition of facets should be consistent, but each application should be able to dynamically choose which facets to use. As applications exist and will emerge that have different needs than DNS envisioned, use of facets should be determinable by each application, not by a protocol designed to suit the needs of the existing DNS system.

To illustrate that last point, let us consider a music namespace that supports all the layer 2 facets proposed by John Klensin and a new one called "singer" (the common name in this namespace is the title of the song). Let us assume now that in this database of songs, the context provided by the song and the singer's name is enough to pinpoint a unique record. It is easy to see that:

- It would not be sufficient for an application to only use the standardized layer 2 facets to disambiguate between songs (the "singer" property is missing).
- It would be bad to prompt the user for the entire context (language, geography, and category) supported by the namespace for accessing a specific song, considering that in this case, only 2 facets would suffice.
- It would not be efficient for the application to store the full

context of the record if the goal is to bookmark a song for future reference.

This observation leads to separating the notion of schema for interoperability (the Klensin's layer 2 schema) from the notion of uniqueness. There seems to be an agreement in the technical community that one of the main reasons for introducing a new layer above DNS is the lack of support for context in DNS names. At the same time, there is no clear consensus on what scope of uniqueness should be standardized.

Our feeling is that agreeing on a fixed context for uniqueness would be an error. Context will always vary across providers and namespaces, and it is not, nor should it be, a requirement to ensure interoperability.

However, if we agree to let service providers disagree on rules of uniqueness, we believe it is still important for applications to be able to discover these rules at the service level. In fact, applications need to understand how much context needs to be captured from the user in order to disambiguate a query (lookup), and how much context needs to be saved to be able to reference the same unique record (bookmark). Without such capability, applications would have to default to using layer one identifiers for uniquely referencing a record.

Hence, there is a need for formalizing the notion of context-based identifiers. Context-based identifiers or unique keys (to use a database terminology) would allow a namespace to publish its own context of uniqueness (the sets of facets that it uses for uniquely identifying a record). Service providers would advertise one or many of these unique keys.

Unique keys could subsequently be recognized and used by applications to uniquely reference a record. We anticipate that applications will take advantage of unique keys to minimize the user interface and the interaction with the user for disambiguation during the lookup process. As explained above, in a keyword system, such interaction is kept to a minimum. The user simply types the common name (keyword) although the application transparently passes the required context (user country, language and the target service) for disambiguation. If the name is unambiguous, the user accesses the resource directly; otherwise a list of records is returned and presented to the user for an interactive disambiguation.

5. Unique Keys

To formalize the notion of a context-based identifier or unique key

and enable each namespace to publish its rules of uniqueness, we propose to introduce to CNRP a new mechanism to declare which facets are part of the unique key. This mechanism will be used to express the set of facets necessary to uniquely identify a single record for a given service (possibly within each different service type). In CNRP, a reference to facets is expressed using the "propertyreference" property. This makes it relatively straightforward for a service to describe its unique keys.

To illustrate this last point, let us consider a namespace of people uniquely identified by their telephone number very much like the type of service provided by ENUM [[RFC2916](#)]. Such service could advertise its unique identifier comprised of the telephone number and the SLS service target as follows (the common name is the telephone number):

```
<?xml version="1.0"?>
  <!DOCTYPE cnrp PUBLIC "-//IETF//DTD CNRP 1.0//EN"
    "http://ietf.org/dtd/cnrp-1.0.dtd">
  <cnrp>
    <results>
      <service>
        <serviceuri>urn:foo:bar</serviceuri>
        <servers>
          <server>
            <serveruri>http://enum.example.com:4321</serveruri>
          </server>
        </servers>
        <description>This is the definition of an ENUM like
          SLS service</description>
        <!-- This schema defines the service as SLS compliant. -->
        <property name="cnrp-service-type">sls</property>
        <!-- We define all the properties of a keyword system. -->
        <propertyschema>
          <propertydeclaration id="i1">
            <propertyname>commonname</propertyname>
            <propertytype>E.164</propertytype>
          </propertydeclaration>
          <propertydeclaration id="i2">
            <propertyname>service</propertyname>
            <propertytype>sls</propertytype>
          </propertydeclaration>
          <propertydeclaration id="i3">
            <propertyname>userpublicprofile</propertyname>
            <propertytype>freeform</propertytype>
          </propertydeclaration>
        </propertyschema>
        <!-- This is where we define the scope of uniqueness. -->
        <uniquekey>
```



```

        <propertyreference required="yes" ref="i1" />
        <propertyreference required="yes" ref="i2" />
    </uniquekey>
    <!-- We define all the properties in the query. -->
    <queryschema>
        <propertyreference required="yes" ref="i1" />
        <propertyreference required="yes" ref="i2" />
    </queryschema>
    <!-- We define all the properties in the response. -->
    <resourcedescriptorschema>
        <propertyreference required="yes" ref="i1" />
        <propertyreference required="yes" ref="i2" />
        <propertyreference required="no" ref="i3" />
    </resourcedescriptorschema >
</service>
</results>
</cnrp>

```

To put things in prospective, a typical query-response interaction between an application and the service would look like this (in this example, the application is looking for the user's email address):

```

C:<?xml version="1.0"?>
C: <!DOCTYPE cnrp PUBLIC "-//IETF//DTD CNRP 1.0//EN"
C:   "http://ietf.org/dtd/cnrp-1.0.dtd">
C: <cnrp>
C:   <query>
C:     <commonname>+330493656172</commonname>
C:     <property name="service" type="sls">email</property>
C:   </query>
C: </cnrp>

S:<?xml version="1.0"?>
S: <!DOCTYPE cnrp PUBLIC "-//IETF//DTD CNRP 1.0//EN"
S:   "http://ietf.org/dtd/cnrp-1.0.dtd">
S: <cnrp>
S:   <results>
S:     <service id="i0">
S:       <serviceuri>http://enum.example.com</serviceuri>
S:     </service>
S:     <resourcedescriptor>
S:       <commonname>+330493656172</commonname>
S:       <resourceuri>mailto://jd@compuserve.com</resourceuri>
S:       <property name="service" type="sls">email</property>
S:       <property name="userpublicprofile" type="freeform">This is
S:         the best email address for Jean Dupont, Pediatrician
S:         in Nice, France</property>
S:     </resourcedescriptor>

```



```
S:      </results>
S:    </cnrp>
```

6. Standardization of Keyword Systems as a Class of Namespaces

This section describes a keyword system as a specific class of namespaces. Although we use CNRP and the SLS proposal to formalize the definition of a keyword system as a service type, the main goal is to identify the elements of a namespace that require standardization in order to define an interoperable class (aside from the requirements identified previously).

Initiated by the CNRP effort, the standardization of keyword system is a goal that has been revived by the Multilingual Internet Name Consortium due to the rapid emergence of competing yet non-interoperable services in Asia, notably in Korea and China. We hope that this approach can establish a path toward the standardization of keyword systems that would proceed in conjunction with the definition of the broader directory layer above DNS.

We feel that the standardization of keyword system must imply the following:

1. The definition of a list of service targets relevant to all keyword systems (e.g. web, email, mobile-web, etc...)
2. For each of these target services, the definition of the unique key common to all keyword systems.

For example, the standard could stipulate that the unique key for keyword systems for the "web" target service be the combination of a common name, a language, and a country. On the other hand, for the target service "email", it is highly conceivable that the standard would be defined on a very different set of facets (for example, "organization name" could be one of the required facets).

7. Formalization of KLS Using The SLS Notation

Using CNRP, a client application could discover that a CNRP service is a keyword system by issuing the standard CNRP "servicequery":

```
<?xml version="1.0"?>
  <!DOCTYPE cnrp PUBLIC "-//IETF//DTD CNRP 1.0//EN"
    "http://ietf.org/dtd/cnrp-1.0.dtd">
  <cnrp>
    <servicequery />
  </cnrp>
```


The service would then return a service object expressing the characteristics common to all keyword systems (also note the suggested use of the cnrp-service-type property as defined in the SLS proposal):

```
<?xml version="1.0"?>
  <!DOCTYPE cnrp PUBLIC "-//IETF//DTD CNRP 1.0//EN"
    "http://ietf.org/dtd/cnrp-1.0.dtd">
  <cnrp>
    <results>
      <service>
        <serviceuri>urn:foo:bar</serviceuri>
        <servers>
          <server>
            <serveruri>http://keywords.ex.com:4321</serveruri>
          </server>
        </servers>
        <description>This is the definition of a keyword system as
          an SLS service</description>
        <!-- We define the service as an SLS service. -->
        <property name="cnrp-service-type">sls</property>
        <!-- We also define the service as a keyword system. -->
        <property name="cnrp-service-type">keywords</property>
        <!-- We define all the properties of this keyword system. -->
        <propertyschema>
          <propertydeclaration id="i0">
            <propertyname>commonname</propertyname>
            <propertytype>freeform</propertytype>
          </propertydeclaration>
          <propertydeclaration id="i1">
            <propertyname>language</propertyname>
            <propertytype>RFC1766</propertytype>
          </propertydeclaration>
          <propertydeclaration id="i2">
            <propertyname>geography</propertyname>
            <propertytype>ISO3166-1</propertytype>
          </propertydeclaration>
          <propertydeclaration id="i3">
            <propertyname>service</propertyname>
            <propertytype>sls</propertytype>
          </propertydeclaration>
        </propertyschema>
        <!-- We define all the properties admissible and required
          in the query. -->
        <querschema>
          <propertyreference required="yes" ref="i0" />
          <propertyreference required="yes" ref="i1" />
          <propertyreference required="yes" ref="i2" />
```



```
<propertyreference required="yes" ref="i3" />
</queryschema>
<!-- We define all the properties admissible and
      required in the response. -->
<resourcedescriptorschema>
  <propertyreference required="yes" ref="i0" />
  <propertyreference required="yes" ref="i1" />
  <propertyreference required="yes" ref="i2" />
  <propertyreference required="yes" ref="i3" />
</resourcedescriptorschema >
<!-- This is where we define the scope of uniqueness. -->
<uniquekey>
  <property name="service" type="sls">web</property>
  <propertyreference required="yes" ref="i0" />
  <propertyreference required="yes" ref="i1" />
  <propertyreference required="yes" ref="i2" />
  <propertyreference required="yes" ref="i3" />
</uniquekey>
</service>
</results>
</cnrp>
```

Note that the queryschema is only one path for an application to discover that a CNRP service is a keyword system. Since the service object is also embedded within the response to any query, the application can simply recognize a keyword system by parsing the information contained in the service object before processing the query results.

8. Conclusion

As the network transforms, meta-data that provides context is critical, and a flexible set of standards regarding collection and use of meta-data must be defined. This will allow a wide proliferation of namespaces to quickly develop that meet the needs of the diverse worldwide population that uses the Internet.

9. References

[CNRP] N. Popp, M. Mealling, and M. Moseley, Common Name Resolution Protocol (CNRP), [draft-ietf-cnrp-10.txt](#), June 2001.

[DNSROLE] J. Klensin, Role of the Domain Name System, [draft-klensin-dns-role-01.txt](#), May 2001.

[DNSSEARCH] J. Klensin, A Search-based access model for the DNS, [draft-klensin-dns-search-01.txt](#), July 2001.

[RFC2916] P. Faltstrom, E.164 number and DNS, [RFC 2916](#), September 2000.

[SLS] M. Mealling and L. Daigle, Service Lookup System (SLS), [draft-mealling-sls-00.txt](#), July 2001.

Author's Address

Yves Arrouye
RealNames Corporation
150 Shoreline Drive
Redwood City, CA 94065

Phone: (650) 486-5503

E-mail: yves@realnames.com

Keyword: RealNames

Web: <http://www.realnames.com/>

Vishesh Parikh
RealNames Corporation
150 Shoreline Drive
Redwood City, CA 94065

Phone: (650) 486-5507

E-mail: vparikh@realnames.com

Keyword: RealNames

Web: <http://www.realnames.com/>

Nico Popp
RealNames Corporation
150 Shoreline Drive
Redwood City, CA 94065

Phone: (650) 486-5549

E-mail: nico@realnames.com

Keyword: RealNames

Web: <http://www.realnames.com/>

Full Copyright Statement

Copyright (C) The Internet Society (2001). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS 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.

Acknowledgement

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

