INTERNET-DRAFT   Eric A. Hall
Document: draft-ietf-crisp-firs-arch-03.txt   August 2003
Expires: March, 2004
Category: Standards-Track

The Federated Internet Registry Service: Architecture and Implementation Guide

Status of this Memo

Copyright Notice

Abstract

This document describes the architectural framework for the Federated Internet Registry Service (FIRS), a distributed service for storing, locating and transferring information about Internet resources using LDAPv3.

Table of Contents

1. Introduction

FIRS is intended to provide a distributed WHOIS-like information service, using the LDAPv3 specifications [RFC3377] for the data-formatting and query-transport functions.

1.1. Background

The original WHOIS service [RFC812] was provided as a front-end to a centralized repository of ARPANET resources and users. Over time, hundreds of WHOIS servers have been deployed across the public Internet, with each server providing general information about the particular network resources under the control of a specific organization.

Unfortunately, neither [RFC812] nor any of its successors define a strict set of data-typing or formatting requirements, and as a result, each of the different implementations provide different kinds of information in slightly different ways. Furthermore, each WHOIS server operates as a self-contained entity, with no standardized mechanisms to infer knowledge of any other servers, meaning that WHOIS servers cannot redirect clients to other servers for additional information. Another concern is that the WHOIS services which are being operated today offer no means of client authentication, requiring that server operators essentially publish all data with a single "world-readable" permission even though this single permission often conflicts with the privacy and security policies of specific jurisdictions.

There are many other secondary issues with the WHOIS service as it exists in current form. However, the largest problems are a lack of standardized data formats, a lack of widely-supported referral mechanisms, and a lack of privacy and security controls, as described in the preceding text.

FIRS attempts to address these issues by defining guidelines for the operation of a distributed and highly-structured WHOIS-like service, using LDAPv3 for the query/response transfer service, and using LDAP schema for the search inputs, answer data, and redirection mechanisms. In short, the intention of this approach is to provide an extensible and scalable WHOIS-like service by leveraging the inherent capabilities of LDAPv3.

1.2. Objectives

The principle objective behind FIRS is to offer structured information about distributed Internet resources in a model which reflects the federated delegations of those resources. This specifically includes centralized delegations from authorized governance bodies (such as DNS domains under top-level domains), but also includes delegations from authorized bodies further down the delegation path (such as leaf-node DNS domain names within the "corp.example.com" zone).

Furthermore, the FIRS service is intended to be used with a wide variety of resources. The core set of specifications define rules for handling the most-common resources (DNS domains, IP addresses, contact information, and so forth), but other types of resources may be grafted onto the architecture as needed. By extension, FIRS should be capable of providing the necessary support structure for any kind of information to be stored in a global mesh of FIRS-centric LDAP directories, and for the FIRS-specific clients and servers to be easily extended to accommodate that data.

Another critical objective is integration support, in that FIRS-specific data should be easily accessible to a wide number of applications. For example, if a network manager needs to retrieve information about a particular host or network which is displayed in a management application, it should be easy for that application to be extended so that the FIRS data can be fetched by that application, rather than always requiring the use of a FIRS-specific application.

Finally, the collection of specifications which define the Federated Internet Registry Service (FIRS) are intended to satisfy the CRISP Working Group requirements, as specified in draft-ietf-crisp-requirements-05, "Cross Registry Internet Service Protocol (CRISP) Requirements" [CRISP-REQ].

1.3. Overview

In order to achieve the stated objectives, the FIRS specifications collectively define an LDAP-specific application, including application-specific namespaces, object classes, attributes, syntaxes, matching filters, behavioral rules, and more. The framework defined in this document is intended to accommodate the specific resource-types and usages, while the other specifications define the technical details for the service as a whole or for the unique resource-types.

Cumulatively, the FIRS collection of specifications define the following service elements:

* Namespace Rules. The FIRS specifications define a layered namespace consisting of DNS-based delegation hierarchies, a FIRS-specific container entry, and resource-specific subordinate entries.

* Schema Definitions. The FIRS specifications reuse some existing LDAP schema definitions, and also define several FIRS-specific definitions, as needed.

* Query-Processing Rules. The FIRS specifications also reuse some existing processing rules, and define several additional rules as needed. Among these rules are requirements for normalizing data, locating servers, processing referrals, and more.

Meanwhile, the core collection of FIRS specifications define these naming, schema and processing rules for the following kinds of Internet resources:

* Partition Data. Each partition in the globally distributed database provides information about the partition itself, allowing users to get a sense of who is providing the data.

* Domain Name Resources. Any DNS domain name (including zone delegations and host-specific resources) can be tracked in the FIRS directory, with information about the domain resource being provided by registries, registrars, or operators, individually or collectively.

* Network Block Resources. Any IP address block (including v4 or v6 networks or host-specific addresses) or Autonomous System can be tracked in the FIRS directory, with information about the network resource being provided by regional registries, registrars, or operators, individually or collectively.

* Contacts. Each partition and network resource has multiple role-specific contact definitions, any of which can refer to generic role accounts or to actual persons, according to

policy and/or desire. Any partition can provide any degree of data about the contact entries under their control.

* Cross-Partition Pointers. Entries can act as aliases to other entries, or can point to other entries as sources of additional data. Meanwhile, attribute values for well-known resources can provide pointers to related data, such as providing a contact identifier that refers to a contact in another partition.

Cumulatively, this architecture provides a substrate of well-formed data which is highly-distributed across independent partitions and servers, while providing multiple "stovepipe" applications of that data.

2. Prerequisites and Terminology

The complete set of specifications in the FIRS collection cumulative define a structured and distributed information service using LDAPv3 for the data-formatting and transport functions. This specification should be read in the context of that set, which currently includes [FIRS-CORE], [FIRS-DNS], [FIRS-DNSRR], [FIRS-CONTCT], [FIRS-ASN], [FIRS-IPV4] and [FIRS-IPV6].

In order to fully understand FIRS, readers should be familiar with [RFC2247], [RFC2251], [RFC2252], [RFC2253], [RFC2254], [RFC2256], [RFC2798], [RFC3296]and [RFC3377].

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119.

3. Reference Example

For the purpose of subsequent discussion, a visual example of a typical partition is provided below. This data-set is not intended to provide a complete demonstration of all the capabilities in FIRS, but instead is only intended for illustration purposes. Each of the resource-specific specifications provide additional examples and illustrations which provide more detail.

Figure 1 below shows an example of a FIRS-specific data-set.

```
     dc=example,dc=com
     |
     +-cn=inetResources,dc=example,dc=com
       [top object class]
       [inetResources object class]
       |
       +-attribute: inetGeneralContacts
       | value: "admins@example.com"
       |
       +-cn=admins@example.com,cn=inetResources,dc=example,dc=com
       | [top object class]
       | [inetResources object class]
       | [inetOrgPerson object class]
       | |
       | +-attribute: mail
       |   value: "admins@example.com"
       |
       +-cn=example.com,cn=inetResources,dc=example,dc=com
       | [top object class]
       | [inetResources object class]
       | [inetDnsDomain object class]
       | [inetDnsRR object class]
       | |
       | +-attribute: inetDnsRRData
       |   value: "NS 86400 864000 ns1.example.net"
       |
       +-cn=www.example.com,cn=inetResources,dc=example,dc=com
         [top object class]
         [inetResources object class]
         [inetDnsDomain object class]
         [referral object class]
         |
         +-attribute: ref
           value: "ldap:///???(1.3.6.1.4.1.7161.1.3.0.1:=
                   www-1.example.net)"
```

Figure 1: The FIRS-specific data for Example Widgets.

As can be seen in Figure 1, entries use a FIRS-specific namespace in conjunction with FIRS-specific schema. FIRS clients use FIRS-specific queries to navigate and retrieve the data, as needed.

4. The FIRS Namespace

A critical aspect of FIRS is the use of an application-specific namespace which is imposed on all FIRS-based resources. The FIRS namespace rules facilitate the programmatic creation of searches, and help to ensure predictable results.

The FIRS namespace consists of three "layers", which are:

* A set of domainComponent relative distinguished names which cumulatively identify a specific partition of the global directory tree.

* A FIRS-specific container entry which segregates the resource-specific child entries from other LDAP data.

* The resource-specific entries which describe the managed resources in the selected partition.

The namespace follows a right-to-left order.

As an example, Figure 1 shows a DNS domain resource entry named "cn=example.com,cn=inetResources,dc=example,dc=com", which refers to the "example.com" domain resource within the "cn=inetResources" container under the "dc=example,dc=com" directory partition.

4.1. The domainComponent Hierarchy

The top-level of the namespace uses the domainComponent naming and mapping rules specified in RFC 2247 [RFC2247], which maps DNS domain names to domainComponent ("dc=") relative distinguished names (RDNs). The full sequence of domainComponent RDNs cumulatively represents a partition in the LDAP directory tree.

In this model, a sequence of domainComponent RDNs map to a domain name in the global DNS hierarchy, with a FIRS partition having an identical scope of authority as its domain name counterpart. Furthermore, the SRV resource records associated with those DNS domains also provide a mechanism for locating the authoritative LDAP servers associated with any particular resource in the global FIRS directory database.

Since the partition roots determine the scope of control over a set of resources, partitions which overlap also have overlapping scopes of control. For example, the "dc=com" and "dc=example,dc=com" partitions can both provide information about the "www.example.com" domain name resource. In order to reduce the amount of ambiguity which is naturally present in this kind of model, FIRS defines multiple bootstrapping models and also defines the default model which should be used for any given resource. For example, queries for centrally-delegated resources are supposed to ask the top-level partition for information about those resources, while queries for user-managed resources are supposed to ask the leaf-node partition for information about those resources.

Figure 1 shows the directory partition of "dc=example,dc=com" which maps to the "example.com" scope of authority from the DNS hierarchy, with the "dc=example,dc=com" sequence representing a distinct partition in the globally distributed directory database.

Note that each of the specifications which govern particular kinds of resources define their own partition-mapping rules, using different portions of the DNS hierarchy. Specifications are explicitly allowed to use whatever portion of the DNS namespace they wish for this service, but the absolute binding between partitions and DNS domains MUST be preserved in all cases. If an organization chooses to offer a private list of resources (such as advertising a list of networks which have been compromised), that organization is free to map the application-specific partition to any domain name it chooses (note that the use of SRV resource records for location information ensures that only a domain name under the control of a willing party can be used).

4.2. The inetResources Container

This specification requires the use of a mandatory LDAP container entry with the RDN of "cn=inetResources", which MUST exist at the root of every directory partition that provides FIRS services. All publicly-accessible resource-specific FIRS-related entries MUST be stored in the "cn=inetResources" container entry.

The primary motivation for this naming rule is for predictability, in that it allows searches to be formed programmatically (a search base for resources in "dc=example,dc=com" can be programmatically formed as "cn=inetResources,dc=example,dc=com", for example). Furthermore, the use of a single container entry for all of an organization's FIRS-related resources allows that branch of the directory database to be managed independently of other entries on the server, which facilitates better operational security and replication controls.

All told, the use of the inetResources container is important enough to justify the MANDATORY usage of this naming syntax.

4.3. Resource-Specific Entries

The FIRS collection of specifications define several Internet resource types, each of which have their own naming rules. However, each resource type follows a consistent naming principle, in that each specific resource has an RDN which uniquely identifies that resource within the inetResources container entry.

For example, Figure 1 shows an entry for the "www.example.com" domain name resource in the "cn=inetResources" container of the "dc=example,dc=com" partition, and also shows an entry for the "admins@example.com" contact resource in that same container and partition. Although the naming syntax is different for each resource type, the naming rules are consistent and facilitate predictable usage.

The naming rules for each of the distinct resource type are provided in the documents which govern those resource types.

4.4. Attribute References

Most of the core attributes that refer to one of the other core attributes provide entry names as data values. In this model, an attribute which needs to identify a particular resource provides the name of the target resource as the attribute value, with the client using this data to instantiate any new queries which may be requested by the user.

For example, many of the object classes provide "contact" attributes, each of which provide one or more contact identifiers as attribute values (such as providing "account@registry" or "user@domain", or both of these identifiers). If the user wishes to obtain more information about any of the listed contacts, they would use the identifier value to start a new query (in turn, these queries may further reference additional entries in the global system, either through the use of referrals or with additional attribute pointers).

4.5. Namespace Aliases

FIRS allows entries to alias for other entries through the use of referrals. Referrals represent one of the strongest capabilities of the FIRS architecture, in that they allow for a significant variety of cross-referencing among entries.

For example, a referral can be used to create a placeholder entry for specific resources (such as a web server), with that entry only existing as a referral for a resource which is managed in another partition (such as a web-hosting server at an ISP). This concept is illustrated in Figure 1, which shows an entry for "cn=www.example.com,cn=inetResources,dc=example,dc=com" that provides a referral with the inetDnsDomainMatch matching filter value of "cn=www-1.example.net", which would result in an entirely new query for that intDnsDomain entry being started (including new lookups for the authoritative partition, and so forth).

Referrals can also be created as subordinate entries underneath a canonical entry. In that model, any data for the local resource would be returned, and would also be accompanied by a referral to another entry on another server, where additional information about the named resource could be retrieved.

FIRS supports two different kinds of LDAP referrals, which are subordinate reference referrals and continuation reference referrals. Subordinate reference referrals indicate that the search base used in the query only exists as an alias to another partition or entry, meaning that the entire query must be restarted in order for any answer data to be retrieved. Meanwhile, continuation reference referrals indicate that some answer data is available, but that more information is available at some other location, and that the client should start new queries in order to retrieve all of the information.

Referrals are provided as URLs. FIRS specifically requires the use of LDAP URLs in order to ensure predictable automated processing. Refer to section 6.4.1 for a brief discussion on how these URLs are processed by FIRS clients.

4.6. Partition Replicas

All directory partitions which provide data for global Internet resources SHOULD be replicated across two or more servers. Each of the authoritative LDAP servers for the managed resource MUST be specified with a unique DNS SRV resource record.

Directory partitions which serve multiple organizations SHOULD also be replicated. For example, an ISP which provides FIRS services for their customers SHOULD also follow these same rules, since outages of those servers will affect multiple parties. Leaf-node directory partitions associated with user-managed resources MAY replicate their partitions, but are not required to do so.

Note that the most effective replication strategy will be for entities to replicate their directory partitions with their delegation parents, as this will allow queries for those resources to be processed by the parent servers (thereby eliminating the need for an immediate referral). In many cases, this will not be feasible (the servers for the "dc=com" directory partition cannot be expected to host replicas of every subordinate directory partition), but it is encouraged where practical.

It is also expected that certain servers will be configured to serve as multi-replica masters, effectively acting as large-scale caching servers for many different resources. When used in conjunction with the targeted bootstrap model described in section 6.4.1, this will allow clients to retrieve a significant amount of information without having to pursue a large number of referrals or redirects. This usage is expected and endorsed.

Note that the LDAP specifications do not currently provide cache timers or any other mechanisms which can indicate how accurate or timely any replicas may be. It is important for replicas to be synchronized frequently in order to avoid problems that may result from replicas going stale.

Further towards the objectives of reliability and redundancy, any referral URLs which include host identifier elements SHOULD provide multiple URLs, each of which identify different hosts. For leaf-node referrals and labeledURI [RFC2079] references, this behavior MAY be relaxed. Note that a host identifier MAY resolve to multiple addresses, and secondary IP addresses SHOULD be used if one of the addresses fails; clients SHOULD NOT give up on a host simply because one of its IP addresses appears to be unreachable.

5. FIRS Schema Definitions

Another critical aspect of FIRS is the use of well-known schema, including object classes, attributes, syntaxes and matching filters. Some of the schema definitions are for the global FIRS service and are usable by all entries (including resource-specific entries), while others are specific to particular resource-types.

For new services, pre-existing schema definitions SHOULD be reused if they are suitable, since this facilitates integration with other LDAP applications.

5.1. Global Schema

There are three global schema definitions which can be used by any of the entries within FIRS. These include:

* The "inetResources" master schema. All FIRS-related entries (including the inetResources container entry and all of the resource-specific subordinate entries) MUST use the inetResources structural object class and schema definitions defined in [FIRS-CORE]. The inetResources object class defines a variety of general-purpose attributes which are useful for general information about an organization and its resources.

* Associated resources. All FIRS-related entries MAY use the "inetAssociatedResources" auxiliary object class and schema definitions defined in [FIRS-CORE]. This object class provides cross-reference pointer attributes which allow an entry to reference other resources which may be of interest to other users or applications.

* Referral pointers. All FIRS-related entries MAY use the "referral" object class and schema definitions defined in [RFC3296]. This object class allows an entry to exist as a referral source, with queries for that resource being redirected to the referral target. Refer to section 4.4 for a discussion on the different kinds of referral mechanisms offered by FIRS, and section 6.4.1 for a discussion on the FIRS referral-processing mechanisms.

Figure 1 shows that all of the entries within and including the "cn=inetResources" container entry have the inetResources object class defined. Meanwhile, each of the resource-specific entries in that example also have their own resource-specific object classes, while the "cn=www.example.com" resource-specific entry also has the referral object class defined.

5.1.1. The inetResources schema

The inetResources object class is intended to provide summary information about a collection of resources under the control of a single organization or management body. Since this object class is also inherited by the resource-specific object classes, these attributes can be defined at each of the subordinate entries if a global set of attribute values is undesirable or unfeasible.

Since multiple directory partitions can use subordinate reference referrals to share a single common inetResources entry, it is important for the data to be applicable to all of the entries which refer to it. For example, it would be effective for a small private company to use a shared set of inetResources attributes for their DNS domain names and IP network blocks, but it would probably be counter-productive for a global ISP to share contact data across all of their hosted domains and routed networks. If separate contacts are required for each resource, the contact data should be specified within each entry, rather than being linked to the inetResources entry.

The inetResources object class provides several multi-valued contact-related attributes for a variety of well-known administrative roles. This model allows the inetResources entry and each of the subordinate managed resources to share a common set of administrative roles, or to have unique roles for each resource, as seen fit by the managing entity.

5.1.2. The inetAssociatedResources schema

The inetAssociatedResources object class defines attributes which are useful for providing general-purpose cross-referencing information with other resources. For example, a contact entry can list IPv4 networks or DNS domains as associated resources, thereby providing a simplistic cross-reference mechanism between an administrator and the resources he manages. In short, any of the common resource types can be associated with any other resource through the use of this object class.

5.1.3. The referral schema

The referral object class is used to redirect queries to other entries programmatically. This object class and its associated schema and rules provide the backbone of the aliasing mechanisms discussed in section 4.4.

5.2. Resource-Specific Schema

In addition to the global schema definitions, each of the resource-specific entries in FIRS MUST use the resource-specific schema definitions defined for use with that specific resource type. These object classes are defined in the specifications which govern the different resource-types. These include:

* DNS domains. Every domain name resource entry MUST use the inetDnsDomain object class and schema definitions defined in [FIRS-DNS]. These entries can refer to zone delegations, host-specific entries, reverse-lookup pointer entries, or any other domain name.

* DNS resource-records. Any domain name resource MAY use the inetDnsRR object class and schema definitions defined in [FIRS-DNSRR]. The inetDnsRR object class defines a single optional attribute for storing multiple DNS resource records as supplemental data to a domain name entry.

* IPv4 address blocks. Every IPv4 address block resource MUST use the inetIpv4Network object class and schema definitions defined in [FIRS-IPV4]. Entries can refer to entire networks or to single hosts, as needed.

* IPv6 address blocks. Every IPv6 address block resource MUST use the inetIpv6Network object class and schema definitions defined in [FIRS-IPV6]. Entries can refer to entire networks or to single hosts, as needed.

* Autonomous system numbers. Every autonomous system number resource MUST use the inetAsNumber object class and schema definitions defined in [FIRS-ASN].

* Contacts. Every contact entry MUST use the inetOrgPerson object class defined in [RFC2798], but MUST also use the additional schema definitions defined in [FIRS-CONTCT].

As was discussed in section 5.1, each resource-specific entry MAY exist as a referral source, or MAY have attributes which refer to additional (related) resources.

6. Query Processing Behaviors

Another critical aspect to FIRS is the query-processing behavioral rules which govern the ways in which a client parses an input string, locates a server which is authoritative for the resource being queried, generates LDAPv3 queries, and processes the resulting answer data. More specifically:

* Query pre-processing. Portions of this process require the client to determine the type of resource being queried for, and to determine the initial partition which should be used for the query. Since this process is different for each particular resource-type, the rules which govern this behavior are defined in each of the resource-specific specifications.

* Bootstrap processing. Once a resource-type and partition have been determined, the client must locate the LDAP servers which are authoritative for that partition. [FIRS-CORE] defines three different bootstrap models that clients can use as part of this process, while each of the resource-specific specifications define which of the models are to be used for each particular resource-type.

* Query processing. Once a server has been located, the client must submit the LDAP query which was formed during the pre-preprocessing phase. [FIRS-CORE] defines certain LDAPv3 query parameters which all FIRS clients MUST conform with, while the resource-specific specifications define resource-specific matching rules.

* Query post-processing. Response data frequently needs to be further processed. For example, referrals may need to be processed, or some kinds of data may need to be localized. These mechanisms and their behavioral rules are defined in [FIRS-CORE], while the resource-specific specifications may also describe supplemental rules.

Each of these phases are discussed in more detail below.

6.1. Query Pre-Processing

Client input is generally limited to a single well-formed unit of data, such as a domain name ("example.com") or an email address ("admins@example.com"), and this single piece of information must be used to subsequently build a fully-formed LDAPv3 query, including the assertion value, the search base, the matching filter, and so forth. All of these steps are part of the pre-processing phase.

Although the exact sequence of steps will vary according to the resource-type being queried, there are some commonalities between each of them. Among these steps:

* Determine the resource type. Different kinds of resources have different processing steps, validation mechanisms, and so forth, each of which require that the resource-type be appropriately identified. Clients MAY use any mechanisms necessary to force this determination.

* Validate and normalize the data. In all cases, the input data MUST be validated and normalized according to the syntax rules defined in the specification which governs the resource-type. As an example of this step, queries for internationalized domain names must be validated and normalized into a canonical UTF-8 [RFC2279] form before any other steps can be taken. Similarly, IP addresses are required to conform to specific syntax rules, with the input address possibly being expanded or compressed so as to comply with the syntax requirements.

* Determine the authoritative directory partition for the named resource. In most cases, the authoritative partition will be a variation of the input query string, but this is not always the case. For example, the default partition for an email address will be extrapolated from the domain component of the email address itself, while the authoritative partition for an autonomous system number uses a reserved (special-purpose) domain name. In some cases, the authoritative partition may change during the subsequent query-processing steps.

* Determine the search base for the query. Each resource type has resource-specific query-processing rules which will dictate how the authoritative partitions are mapped to the search base. In some cases, the cn=inetResources container entry in the authoritative partition will be used "as-is", while in other cases, the cn=inetResources container entry in a delegation parent of the authoritative partition will be used instead. In some cases, the search base may change during subsequent query-processing steps.

* Determine the assertion value for the query. The assertion value will usually be the normalized form of the input query. In some cases, the assertion value may change during subsequent query-processing steps.

* Determine the matching filter. Each resource-type has its own matching filter rules. For example, contact entries are matched with a simple equalityMatch comparison, while in

other cases the matching filter will be an extensibleMatch which is peculiar to the resource-type in use.

Once all of the pre-processing steps have been successfully completed, the client will have to locate an LDAPv3 server which is authoritative for the search base before it can submit the query. This process is described in section 6.2 below.

6.2. Bootstrap Processing

The bootstrap process uses DNS queries to locate the LDAP servers which should be used for a query. However, since different kinds of resources are managed through different delegation models, there are also different bootstrap models which have to be used to perform this process.

FIRS supports three different bootstrap models, which are:

* Targeted. The "targeted" bootstrap model has the client attempting to locate the LDAP servers associated with a specific domain name, such as a domain name which may be returned as referrals or URLs. If no servers can be found at that domain name, the client exits the query.

* Top-down. The "top-down" bootstrap model has the client attempting to locate the LDAP servers associated with a top-level partition in the delegation path to the authoritative partition, and then following any subsequent LDAP referrals which may be returned. If no servers can be found for the top-level domain, the client exits the query.

* Bottom-up. The "bottom-up" bootstrap model has the client attempting to locate the LDAP servers associated with the authoritative partition itself. If no servers can be found for that partition, the authoritative partition is reset to the immediate parent in the delegation hierarchy and new DNS queries are issued, with this process repeating until a server is found or there are no more domains in the delegation path which can be queried.

Each of the models are appropriate to different usages. For example, The targeted model is most useful when a particular piece of data is presumed to exist at a pre-determined location. Meanwhile, the top-down model is best suited for searches about global resources which are centrally managed and delegated (such as IP addresses and DNS domains), and where centrally-managed delegation information is critical. Finally, the bottom-up model is most appropriate for resources which are managed at a leaf-node (such as contact information).

6.3. Query Processing

Once an LDAP server has been located, the LDAPv3 query is submitted to that server.

Most of the values for the query will have been collected during the pre-processing phase, although [FIRS-CORE] defines some rules which govern all queries. For example, [FIRS-CORE] specifies a maximum time limit of 60 seconds for queries (among other similar kinds of restrictions) in order to prevent runaway searches which would otherwise match all entries.

[FIRS-CORE] also allows for authentication and access controls, in that FIRS servers are allowed to limit the depth and breadth of information that they provide to a specific client based on a variety of factors, including the level of authenticated access.

Another consideration which can arise during this phase of the process is protocol and schema versioning considerations. The [LDAP] specifications already define mechanisms for protocol version negotiation, and the use of these mechanisms is endorsed and encouraged in [FIRS-CORE].

Schema and capability negotiation is handled through the use of a "firsVersion" control (as defined in [FIRS-CORE]), which provides a list of the FIRS-specific object classes that are supported by the target server. If a server advertises support for any of the FIRS-specific object classes, then the server also commits to supporting all of the attributes and matching filters associated with that object class. Clients can then use this information to determine whether or not the current server is using the same schema as the client.

The client MAY also use this information to determine whether or not it will need to construct its own queries. Since it is somewhat likely that a particular server will not support all of the mechanisms required by the complete FIRS model (especially including all of the extended matching filters), then the client can use this information to determine if it needs to construct its own extended queries locally. Refer to the resource-specific documents for more information on this process.

6.4. Query Post-Processing

Once a query has been submitted and processed, the server will return answer data or some kind of referral, or possibly both. In general, FIRS clients are expected to display all of the answer data and process all of the referrals, although there are specific considerations which must be taken into account. In particular, there are considerations for handling the different kinds of referrals, and there are localizations issues for specific kinds of attribute data.

6.4.1. Referrals

As was discussed in section 4.4, there are two kinds of referral mechanisms which are used with FIRS, which are subordinate reference referrals and continuation reference referrals. More specifically:

* Subordinate reference referrals. Subordinate reference referrals are returned when the search base specified in a query exists as a referral to some other entry. This condition means that the current search operation cannot proceed, and that the search MUST be restarted using the search base specified in the referral message.

  Any of the FIRS-specific entries MAY be defined as subordinate reference referrals, although they are typically only used when the inetResources container entry in a partition is an alias for an inetResources container entry in another partition. Subordinate reference referrals and their schema are defined in [RFC3296] although there are additional restrictions placed on their usage as described in [FIRS-CORE].

* Continuation reference referrals. Continuation reference referrals are returned when a search operation has been successfully processed by the queried server, but the answer data also includes referrals to other entries. This condition means that the current search operation has succeeded, but that additional searches SHOULD be started in order for all of the answer data to be retrieved.

  These referrals are often provided as supplemental data to an answer set, although this is not required (a continuation reference referral can be the only response, but it won't be the only response in the common case).

Continuation reference referrals and their schema are also defined in [RFC3296], with additional restrictions placed on their usage as described in [FIRS-CORE].

Whenever a referral is received in response to a query, the client is required to display any answer data which has also been received and then process the referral.

LDAP referrals can use any kind of URL, although FIRS specifically requires the use of LDAP URLs. The client is required to parse the resulting URL for a host identifier, port number, search base, and assertion value elements, and then use these elements to construct and issue new queries.

Note that [RFC2251] defines a superior reference referral which is used as a "default referral" for out-of-scope searches. However, FIRS specifically excludes support for superior reference referrals. Any superior reference referrals which are encountered as part of this service are to be treated as errors.

6.4.2. Internationalization and localization

The FIRS model uses the internationalization and localization services which are inherent in LDAPv3. In many cases, this native support is sufficient to accommodate internationalization and localization considerations. However, there are several cases where additional and explicit support is required.

For example, the domainComponent attribute is specifically restricted to seven-bit character codes, and is traditionally interpreted as simple [US-ASCII]. This is problematic with internationalized domain names and the domainComponent attributes derived from them, since these attribute sequences are used in partition identifiers, search bases, and numerous other areas. In order to ensure interoperability, all DNS domain names which are mapped to domainComponent attributes MUST be reduced to their ASCII-compatible form using the ToASCII process defined in [RFC3490] before they are used for domainComponent sequences.

Similarly, although DNS is technically capable of storing eight-bit code-point values, the operational rules which govern DNS do not support this usage for domain names which are used as host identifiers (and this includes zone delegations). As a result, internationalized domain names which are to be used for DNS lookups (such as queries for SRV resource records) MUST be reduced to their ASCII-compatible form using the ToASCII process defined in [RFC3490] before these queries are issued.

In those cases where entries or attributes use normalized UTF-8 sequences inside of FIRS (specifically domain names and email addresses), FIRS clients SHOULD offer ASCII-compatible versions of those sequences, using the ToASCII process defined in [RFC3490]. This will ensure that clients are able to use these sequences with legacy (pre-IDNA) applications directly. For example, if an entry displays an inetAssociatedDomains attribute, the domain names in that attribute should be displayed in their default UTF-8 form (assuming that the client's operating system and application allows it), but should also be made available in their ASCII-compatible form (either as a clipboard option, command-line option, or some other user-selectable switch) in order to allow the data to be passed to a legacy application in a form which is understandable by the legacy application.

Attribute names are fixed, and can therefore be localized easily. As such, clients MAY choose to convert attribute names into a language appropriate to the local user for display purposes if this is desirable. However, clients MUST NOT localize attribute names which are used for query input. For example, clients MUST NOT convert "cn=" or "dc=" relative distinguished labels into a language-specific mapping and then use the mapped versions of these labels for assertion values in a subsequent query.

RFC 2277 [RFC2277] requires free-text data to be tagged with language tags. RFC 2596 [RFC2596] defines a mechanism for storing language tags and language-specific attribute values in LDAPv3, and these mechanisms SHOULD be supported by FIRS clients and servers. For example, an organization name could be provided in English and Arabic, with the language tags allowing the client to display the appropriate attribute value instance based on the locale settings of the user.

International postal regulations generally require that the recipient address on an envelope be provided in a language and charset which is native to the recipient's country, with the exception of the destination country name which should be provided in a language and charset that is native to the sender's country. This model ensures that the sender's post office will be able to route the mail to the recipient's country, while also ensuring that the destination country's post office will be able to perform local delivery. In order to facilitate this usage, the country attribute value SHOULD be localized to the local user's nomenclature for that country, but other postal address data SHOULD NOT be localized.

Notwithstanding the above, contact names SHOULD be provided in English in order to facilitate inter-party communications, using the mechanisms offered by [RFC2596]. For example, the default contact entry for a person in Japan SHOULD be provided in the native form for that person, but an English form SHOULD also be provided in order to allow non-Japanese users to properly address that person in subsequent communications. As stated in the preceding paragraph however, any postal communications for that person SHOULD use the native-language representation (at least on the envelope) in order to facilitate delivery.

Time and date strings in LDAP use the generalizedTime syntax, making them predictable and easily convertible if necessary. As such, dates MAY be localized for display purposes by client applications as necessary.

Finally, clients must recognize that some URL data is likely to be escaped, using at least one of the multiple rules which affect URLs and resource-specific data. For example, a URL which contains a domain name resource could theoretically have been escaped with three or four different syntax rules, and clients MUST be prepared to decode these URLs appropriately.

6.5. Query Restriction Mechanisms

[FIRS-CORE] defines several discrete rules that can be employed by server operators to limit the queries received from any particular client. Some of these restrictions include:

* Servers MAY refuse service to any client (5.3.3).

* Servers MAY impose arbitrary maximum limits on the number of queries issued by any particular client for any given time period, such as limiting clients to 50 queries-per-day or five queries-per-hour (5.3.3).

* Servers MAY impose arbitrary wait intervals between successive queries from any particular client, such as requiring clients to wait five minutes between queries (5.3.3).

* Servers MAY impose arbitrary limits on the maximum number of answers that they will return to a client over any given

time period (such as limiting clients to 100 answers-per-day), and MAY base this restriction on any type of answer data (5.3.3).

* Servers MAY restrict the resource-types that any particular client can query for, and MAY restrict the matching filters that any particular client can use for any of the resource-types that they are allowed to query (5.3.1).

* Servers MAY restrict the maximum length of time they spend processing any particular query (5.3.2).

* Servers MAY restrict the maximum number of matches that will be returned to any particular query (5.3.2).

* Server operators MAY define ACLs on entries and attributes in the database that restrict the data that is matched for any particular client (5.3.4).

Any of these restrictions MAY be defined by the operators of the server in question, according to the policies deemed necessary by the operators of that server. For example, operators MAY apply different restrictions on ranges of client addresses, or authenticated identity, or any other necessary metric, or any combination thereof.

7. Transition Issues

There are a handful of areas where FIRS does not fully compare with all of the existing WHOIS service offerings. These areas are discussed in more detail below.

7.1. NIC Handles

Legacy NIC handles in existing databases can be accommodated using two possible mechanisms:

* NIC handle output in legacy WHOIS systems may be replaced with contact identifier addresses, using domain elements which refer to the operator's domain. For example, the NIC handle of EH26 on Network Solutions' WHOIS server could be replaced with "eh26@firs.netsol.com" or something similar. This approach causes lookups for that email address to be directed towards the operator's FIRS servers, and facilitates fast coalescence around the FIRS system.

* Use the inetLocalIdentifier attribute defined in [FIRS-CORE]. This option provides a simple text string which can be used for private identifiers, but provides no integration with FIRS, other than allowing for attribute value searches.

Although both options can be used simultaneously, the former mechanism is especially preferred.

7.2. Change-Logs

Several WHOIS services provide pseudo change-logs in their response data, listing each unique modification event which has occurred for a particular resource. For example, RIPE and some of the ccTLDs provide WHOIS output which includes a series of "changed" fields that itemize every modification event ("updated", "added", etc.), the modifier, and the modification date, which cumulatively act as a change-log for the resource in question.

Organizations are certainly free to maintain this information on their internal systems. However, this information is not necessary for public view of the data in the FIRS service. Furthermore, where auditing of this information will be required, a format which is suitable to legal review will also be required.

Organizations who wish to make change-log information available should use an auxiliary LDAP schema for this purpose. An initial schema is available at http://www.ehsco.com/misc/draft-hall-ldap-audit-00.txt, although it has not been proposed as a standards-track effort, and should only be used as a starting point for other development.

7.3. Legacy System Support

Organizations which already provide WHOIS services over TCP port 43 have several migration options. At the lowest extreme, these organizations can continue to use and support those systems as-is, without any modifications. However, other organizations may choose to implement a FIRS client in a text-based application (such as a Perl script), with that application accepting typical queries over the legacy TCP/43 port, processing those queries through FIRS, and returning answer data back to the legacy WHOIS client. Another approach is described in draft-newton-whois-crisp-cohabitation-00.txt, which advocates the use of NAPTR and SRV resource records to redirect legacy clients to FIRS servers.

A similar range of options are available for back-end database integration. Organizations who do not wish to align their back-end databases to the LDAP/FIRS model can use basic scripts to generate LDIF files on a suitable schedule, and then populate their LDAP servers with this data. Meanwhile, other organizations may choose to provide an LDAP front-end to an existing database, while other organizations may choose to use a single LDAP repository for all of their applications.

In general terms, FIRS does not require or endorse any of these mechanisms, and they are only presented here so that operators are aware of the options.

8. Security Considerations

The FIRS collection of specifications describe an application of the LDAPv3 protocol, and as such it inherits the security considerations associated with LDAPv3, as described in section 7 of [RFC2251].

By nature, LDAP is a read-write protocol. As such, there are significant risks associated with unintentionally allowing unauthorized third-parties to update the underlying data. Moreover, allowing FIRS clients to update delegation data could result in network resources being stolen from their lawful operators. For example, if the LDAP front-end had update access to a domain delegation database, a malicious third-party could theoretically take ownership of a domain by exploiting an authentication weakness, thereby causing ownership of the domain to be changed to another party. For this reason, it is imperative that the FIRS service not be allowed to make critical modifications to delegated resources without ensuring that all possible precautions have been taken, potentially including strong authentication and encryption practices.

The query processing models described in these documents make use of DNS lookups in order to locate the LDAP servers associated with a particular resource. DNS is susceptible to certain attacks and forgeries which may be used to redirect clients to LDAP servers which are not authoritative for the resource in question.

This document provides multiple query models which will cause the same query to be answered by different servers (one would be processed by a delegation entity, while another would be processed by an operational entity). As a result, each of the servers may provide different information, depending upon the query type that was originally selected.

Some operators may choose to purposefully provide misleading or erroneous information in an effort to avoid responsibility for bad behavior. In addition, there are likely to be sporadic operator errors which will result in confusing or erroneous answers.

Neither this specification nor the LDAPv3 protocol currently provide cache timers or any other mechanisms which can indicate how accurate or timely any replicas may be. As a result, it is possible for a replica to become significantly outdated, even to the point of containing wholesale errors.

For all of the reasons listed above, it is essential that applications and end-users not make critical decisions based on the information provided by the FIRS service without having reason to believe the veracity of the information. Users should limit unknown or untrusted information to routine purposes.

Despite these disclaimers, however, it is very likely that the information presented through the FIRS service will be used for many operational and problem-resolution purposes. In order to ensure the veracity of the information, a minimal set of operational guidelines are provided herein. For the most part, these rules are designed to prevent unauthorized modifications to the data.

Note that these rules only apply to data which is willingly provided; no data is required to be entered, but where the data is provided, it SHOULD be validated as accurate on entry, and it MUST be secured against unauthorized modifications.

* The inetResources container entry and all of the resource-specific subordinate entries within every public DIT that provides FIRS resources SHOULD have anonymous read-only access permissions, and MUST NOT have anonymous add, delete or modify permissions.

* With the exception of contact-related attributes from the inetOrgPerson object class, each attribute MAY have whatever restrictions are necessary in order to suit local security policies, government regulations or personal privacy concerns. When the inetOrgPerson object class is used to provide contact details, the mail attribute MUST be defined, SHOULD be valid, SHOULD have anonymous read-only

access, and MUST NOT have anonymous add, delete or modify permissions.

By using the inetOrgPerson object class, it is expected that existing contact-related entries can be reused. If reusing these entries is undesirable or unfeasible, entries with the necessary access SHOULD be made available.

* End-users and implementers SHOULD provide anonymous access to the creatorsName, createTimestamp, modifiersName and modifyTimestamp operational attributes associated with each entry in the inetResources branch, since this information is useful for determining the age of the underlying data.

* Server operators MAY define additional add, delete or modify permissions for authenticated users, using any LDAPv3 authentication mechanisms they wish. In particular, delegation entities MAY provide for the remote management of delegated resources (such as assigning modification privileges to the managers of a particular delegated domain or address block), although this is entirely optional, and is within the sole discretion of the delegation body.

* In the general case, server operators SHOULD NOT offer clear-text authentication mechanisms over unencrypted connections.

Finally, there are physical security issues associated with any service which provides physical addressing and delivery information.

In summary, organizations MAY provide as much data as possible, although no information is required.

9. IANA Considerations

The FIRS collection of specifications define an application of the LDAPv3 protocol rather than a new Internet application protocol. As such, there are no protocol-related IANA considerations.

However, the FIRS collection of specifications do define several LDAP schema elements, including object classes, attributes, syntaxes and extensibleMatch filters, and these elements should be assigned OID values from the IANA branch. Furthermore, some of the specifications define their own status codes as attribute values, and IANA is expected to maintain the code-point mapping values associated with these attributes.

Finally, some of the specifications also describe public DNS and LDAP servers and data. It is expected that IANA will see to the establishment and maintenance of these servers and data.

10. Normative References

[CRISP-REQ] Newton, A. "Cross Registry Internet Service Protocol (CRISP) Requirements", draft-ietf-crisp-requirements-05, July 2003.

[FIRS-ASN] Hall, E. "Defining and Locating Autonomous System Numbers in the Federated Internet Registry Service", draft-ietf-crisp-firs-asn-03, August 2003.

[FIRS-CONTCT] Hall, E. "Defining and Locating Contact Persons in the Federated Internet Registry Service", draft-ietf-crisp-firs-contact-03, August 2003.

[FIRS-CORE] Hall, E. "The Federated Internet Registry Service: Core Elements", draft-ietf-crisp-firs-core-03, August 2003.

[FIRS-DNS] Hall, E. "Defining and Locating DNS Domains in the Federated Internet Registry Service", draft-ietf-crisp-firs-dns-03, August 2003.

[FIRS-DNSRR] Hall, E. "Defining and Locating DNS Resource Records in the Federated Internet Registry Service", draft-ietf-crisp-firs-dnsrr-02, July 2003.

[FIRS-IPV4] Hall, E. "Defining and Locating IPv4 Address Blocks in the Federated Internet Registry Service", draft-ietf-crisp-firs-ipv4-02, August 2003.

[FIRS-IPV6] Hall, E. "Defining and Locating IPv6 Address Blocks in the Federated Internet Registry Service", draft-ietf-crisp-firs-ipv6-02, August 2003.

[ISO10646] "ISO/IEC 10646-1:2000. International Standard -- Information technology -- Universal Multiple-Octet Coded Character Set (UCS) --

Part 1: Architecture and Basic Multilingual Plane"

[RFC2079] Smith, M. "Definition of an X.500 Attribute Type and an Object Class to Hold Uniform Resource Identifiers (URIs)", RFC 2079, January 1997.

[RFC2247] Kille, S., Wahl, M., Grimstad, A., Huber, R., and Sataluri, S. "Using Domains in LDAP/X.500 DNs", RFC 2247, January 1998.

[RFC2251] Wahl, M., Howes, T., and Kille, S. "Lightweight Directory Access Protocol (v3)", RFC 2251, December 1997.

[RFC2252] Wahl, M., Coulbeck, A., Howes, T., and Kille, S. "Lightweight Directory Access Protocol (v3): Attribute Syntax Definitions", RFC 2252, December 1997.

[RFC2253] Wahl, M., Kille, S., and Howes, T. "Lightweight Directory Access Protocol (v3): UTF-8 String Representation of DNs", RFC 2253, December 1997.

[RFC2254] Howes, T. "The String Representation of LDAP Search Filters", RFC 2254, December 1997.

[RFC2255] Howes, T., and Smith, M. "The LDAP URL Format", RFC 2255, December 1997.

[RFC2256] Wahl, M. "A Summary of the X.500(96) User Schema for use with LDAPv3", RFC 2256, December 1997.

[RFC2277] Alvestrand, H. "IETF Policy on Character Sets and Languages", BCP 18, RFC 2277, January 1998.

[RFC2279] Yergeau, F. "UTF-8, a transformation format of ISO 10646", RFC 2279, January 1998.

[RFC2596] Wahl, M., and Howes, T. "Use of Language Codes in LDAP", RFC 2596, May 1999.

[RFC2782] Gulbrandsen, A., Vixie, P., and Esibov, L. "A DNS RR for specifying the location of services (DNS SRV)", RFC 2782, February 2000.

[RFC2798]   Smith, M. "Definition of the inetOrgPerson LDAP Object Class", RFC 2798, April 2000.

[RFC3296]   Zeilenga, K. "Named Subordinate References in Lightweight Directory Access Protocol (LDAP) Directories", RFC 3296, July 2002.

[RFC3377]   Hodges, J., and Morgan, R. "Lightweight Directory Access Protocol (v3): Technical Specification", RFC 3377, September 2002.

[RFC3490]   Faltstrom, P., Hoffman, P., and Costello, A. "Internationalizing Domain Names in Applications (IDNA)", RFC 3490, March 2003.

[US-ASCII]   Cerf, V. "ASCII format for Network Interchange", RFC 20, October 1969.

11. Informational References

[RFC812]   Harrenstien, K., and White, V. "NICNAME/WHOIS", RFC 812, March 1982.

12. Changes from Previous Versions

draft-ietf-crisp-firs-arch-03:

* Several clarifications and corrections have been made.

* Added a discussion on the various query-restriction mechanisms that are available in the system as a whole.

* Clarified the use of referrals and added a discussion on attribute references.

draft-ietf-crisp-firs-arch-02:

* Several clarifications and corrections have been made.

draft-ietf-crisp-firs-arch-01:

* Several clarifications and corrections have been made.

draft-ietf-crisp-firs-arch-00:

* Restructured document set, separating the architectural discussion from the technical descriptions.

* Consolidated the security discussions.

draft-ietf-crisp-lw-core-00:

* As a result of the formation of the CRISP working group, the original monolithic document has been broken into multiple documents, with draft-ietf-crisp-lw-core describing the core service, while related documents describe the per-resource schema and access mechanisms.

* References to the ldaps: URL scheme have been removed, since there is no standards-track specification for the ldaps: scheme.

* An acknowledgements section was added.

draft-hall-ldap-whois-01:

* The "Objectives" section has been removed. [ir-dir-req] is now being used as the guiding document for this service.

* Several typographical errors have been fixed.

* Some unnecessary text has been removed.

* Figures changed to show complete sets of object classes, to improve inheritance visibility.

* Clarified the handling of reverse-lookup domains (zones within the in-addr.arpa portion of the DNS hierarchy) in the inetDnsDomain object class reference text.

* Referrals now use regular LDAP URLs (multiple responses with explicit hostnames and port numbers). Prior editions of this specification used LDAP SRV resource records for all referrals.

* The delegation status codes used by the inetDnsDelegationStatus, inetIpv4DelegationStatus, inetIpv6DelegationStatus and inetAsnDelegationStatus attributes have been condensed to a more logical set.

* Added an inetDnsAuthServers attribute for publishing the authoritative DNS servers associated with a domain. NOTE THAT THIS IS A TEMPORARY ATTRIBUTE THAT WILL EVENTUALLY BE

REPLACED BY GENERALIZED RESOURCE-RECORD ENTRIES AND ATTRIBUTES.

* Added an inetGeneralDisclaimer attribute for publishing generalized disclaimers.

* Added the inetAssociatedResources auxiliary object class for defining associated resources, and moved some of the IP addressing and ASN attributes to the new object class.

* Several attributes had their OIDs changed. NOTE THAT THIS IS AN INTERNET DRAFT, AND THAT THE OIDS ARE SUBJECT TO ADDITIONAL CHANGES AS THIS DOCUMENT IS EDITED.

13. Author's Address

Eric A. Hall
ehall@ehsco.com

14. Acknowledgments

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

Portions of this document were funded by VeriSign Labs.

The first version of this specification was co-authored by Andrew Newton of VeriSign Labs, and subsequent versions continue to be developed with his active participation. Edward Lewis and Peter Gietz also contributed significant feedback to this specification in the later stages of its developments.

15. Full Copyright Statement