Definitions for talking about directories

Status of this Memo

Copyright Notice

Abstract

   When discussing systems for making information accessible through the
   Internet in standardized ways, it may be useful if the people who are
   discussing it have a common understanding of the terms they use.

   For example, a reference to this document would give one the power to
   agree that the DNS (Domain Name System) is a global lookup repository
   with perimeter integrity and loose, converging consistency. On the
   other hand, a LDAP (Lightweight Directory Access Protocol) directory
   server is a local, centralized repository with both lookup and search
   capability.

   This document discusses one group of such systems which is known
   under the term, "directories".

1. Introduction and basic terms

   We suggest using the following terms for the remainder of this
   document:

   - Information: Facts and ideas which can be represented (encoded) as
      data in various forms.

   - Data: Information in a specific physical representation, usually a
      sequence of symbols that have meaning; especially a representation
      of information that can be processed or produced by a computer.
      (From [SEC].)

   - Repository: An amount of data that is accessible through one or
      more access methods.

- Requester: Entity that may (try to) access data in a repository. Note that no assumption is made that the requester is animal, vegetable, or mineral.

- Maintainer: Entity that causes changes to the data in the repository. Usually, all maintainers are requesters, since they need to look at the data too, however, the roles are distinct.

- Access method: Well-defined series of operations that will cause data available from a repository to be obtained by the requester.

- Site: Entity that hosts all or part of a repository, and makes it available through one or more access methods. A site may in various contexts be a machine, a datacenter, a network of datacenters, or a single device.

This document is not intended to be either comprehensive or definitive, but is intended to give some aid in mutual comprehension when discussing information access methods to be incorporated into Internet Standards-Track documents.

2. Dimensions of classification

2.1 Uniqueness and scope

Some information systems are global, in the sense that only one can sensibly exist in the world.

Others are inherently local, in that each locality, site or even box will run its own information store, independent of all others.

The following terms are suggested:

- Global repository: A repository that there can be only one of in the world. The world itself is a prime example; the public telephone system's number assignments according to E.164 is another.

- Local repository: A class of repository of which multiple instances can exist, each with information relevant to that particular repository, with no need for coordination between them.

- Centralized repository: A repository where all access to data has to pass through some single site.

- Distributed repository: A repository that is not centralized; that is, access to data can occur through multiple sites.

- Replicated repository: A distributed repository where all sites
  have the same information.

- Cooperative repository: A distributed repository where not all
  sites have all the information, but where mechanisms exist to get
  the info to the requester, even when it is not available to the
  site originally asked.

Note: The term "global" is often a matter of social or legal context;
for instance, the E.164 telephone numbering system is global by
international treaty, while the debate about whether the Domain Name
System is global in fact or just a local repository with ambitions
has proved bait for too many discussions to enumerate.

Some claim that globality is in the eye of the beholder; "everything
is local to some context". When discussing technology, it may be
wise to use "very widely deployed" instead.

Note: Locating the repositories changes with the scale of
consideration. For instance, the global DNS system is considered a
distributed cooperative repository, built out of zone repositories
that themselves may be distributed, and are always replicated when
distributed.

2.2 Search, Lookup, Query and Notify

A different consideration when describing repositories is the types
of method they offer to find information.

The chief classifications are:

- Lookup methods require the user to know or guess some exact value
  before asking for information, sometimes called a "lookup key" or
  "identifier" and sometimes called a "name". The word "name" is
  NOT recommended, since it conflicts with other uses of that word
  The response to a successful lookup is a single group of
  information, often called "information about the identified
  entity". A lookup method is binary (yes/no) in recall: It either
  returns one result or no result; if it returns a result, that
  result is the right result for that lookup key, so it is also of
  binary precision (no info or completely relevant info).

- Search methods require the user to know some approximate value of
  some information. They usually return zero, one, or more
  responses that match the information supplied according to some
  algorithm. Where the repository is structured around "entities",
  the information can be about zero, one, or many entities.

In database terms, a lookup method corresponds to a query exactly matching a unique key on a table; all other database queries would be classified as "search" methods.

In general, repositories that offer more flexible search methods may also give room for ad-hoc queries, refinements from a previous query, approximate matching and other aids; this may lead to many different combinations of precision and recall.

One may define terms to enumerate what one gets out of these repositories:

   . Precision is the degree to which what you asked for is what you
      wanted (no extraneous information)

   . Recall is the ability to assure oneself that all relevant data
      from the repository is returned

   . Type I errors occurs when relevant data exists in the
      repository, but is not returned

   . Type II errors occur when irrelevant data is returned with a
      query result

Note that these concepts can only be applied when the property "relevance" is well defined; that is, it depends on what the repository is used for. A further discussion of these topics is found in [KORFHAGE].

An orthogonal dimension has to do with time:

- Query repositories will answer a request with a response, and once
   that is over with, will do nothing more.

- Notify repositories will get a request from a user to have
   information returned at some later time when it becomes available,
   current or whatever, and will respond at that time with a
   notification that information is available.

- Subscription repositories are like notify repositories, but will
   transfer the actual information when available.

2.3 Consistency models

Consistency (or the lack thereof) is a property of distributed repositories; for this particular discussion, we ignore the subject of semantically inconsistent data (such as occurrences of pregnant men), and focus on the problem of consistency where inconsistency is

defined as having the same request, using the same credentials, be
answered with different data at different sites.

Distributed repositories may have:

- Strict consistency, where the problem above never arises. This is
  quite difficult; repositories that exhibit this property are
  usually quite constrained and/or quite expensive.

- Strict internal consistency, where the replies always reflect a
  consistent picture of the total repository, but some sites may
  reflect an earlier version of the repository than others.

- Loose, converging consistency, where different parts of the
  repository may be updated at different times as seen from a single
  site, but the process is designed in such a way that if one stops
  making changes to the repository, all sites will sooner or later
  present the same information.

- Inconsistency, where no guarantee can be made whatsoever

One interesting variant is subset consistency, where the system is
consistent (according to one of the definitions above), but not all
questions will be answered at all sites; possibly because different
sites have different policies on what they make available (NetNews),
or because different sites only need different subsets of the "whole
picture" (BGP).

2.4 Security models

Its harder to describe security models in a few sentences than other
properties of information systems. There also exists a large
specialized literature on terminology for security, including [SEC].

Some thoughts, though:

On trust in data: Why do we trust a piece of data to be correct?

- Because it's in the repository (and therefore must have been
  authorized).

  This is perimeter (or Eggshell) integrity.

- Because it contains internal integrity checks, usually involving
  digital signatures by verifiable identities. This is item
  integrity; the granularity of the integrity and the ability to do

integrity checks on the relationships between objects is extremely
important and extremely hard to get right, as is establishing the
roots of the trust chains.

- Because it fits other available information, and causes the right
  things to happen when I use it.

  This is hopeful integrity.

Which integrity model to choose is a matter of evaluating the cost of
implementing the integrity (cost), the value to you of integrity of
the resource being protected (value), and the impact of cost on doing
business (risk).

On access to information, the usual categories apply:

- Open access: Anyone can get the information.

- Property-based access: Access because of what you are, or where
  you are. For example limited to "same network", "physically
  present", or "resolvable DNS name"

- Identity-based access: Access because of who you are (or
  successfully claim to be). (I.e., username/password, personal
  certificates or other verifiable information.)

  These are then backed up by a layer specifying what the identity
  you have proven yourself to be has access to.

- Token-based access: Access because of what you have. Hardware
  tokens, smartcards, certificates, or capability keys.

  In this case, access is given to all who can present that
  credential, without caring about their identity.

The most common approaches are identity-based and open access;
however, "what you have" access is commonly used informally in, for
example, password-protected FTP or Web sites where the password is
shared between all members of a group.

2.5 Update models

A few examples:

- Read-only repositories have no standard means of changing the
  information in them. This is usually accomplished through some
  other interface than the standard interface.

   - Read-mostly repositories are designed based on a theory that reads
      will greatly outnumber updates; this may, for instance, be
      reflected in relatively slow consistency-updating protocols.

   - Read-write repositories assume that the updates and the read
      operations are of the same order of magnitude.

   - Write-mostly repositories are designed to store an incoming stream
      of data, and when needed reproduce a relevant piece of data from
      the stream. Typical examples are insurance company databases and
      audit logs.

2.6 The term "Directory"

   The definitions above never used the term "Directory".

   In most common usages, the properties that a repository must have in
   order to be worthy of being called a directory are:

   - Search

   - Convergent consistency

   All the other terms above may vary across the set of things that are
   called "directories".

3. Classification of some real systems

3.1 The Domain Name System

   The DNS [DNS] is a global cooperative lookup repository with loose,
   converging consistency and query capability only.

   It is either strictly read-only or read-mostly (with Dynamic DNS),
   has an open access model, and mainly perimeter integrity (some would
   say hopeful integrity). DNSSEC [DNSSEC] aims to give it item
   integrity.

   The DNS is built out of zone repositories that themselves may be
   distributed, and are always replicated when distributed.

   Note that like many other systems, the DNS has some features that do
   not fit neatly in the classification; for instance, there is a
   (deprecated and not widely used) function called IQUERY, which allows
   a very limited query capability.

If one opens up the box and looks at the relationship between primary
and secondary nameservers, that can be seen as a limited form of
notify capability, but this is not available to end-users of the
total system.

3.2 The (imagined) X.500 Global Directory

X.500 [X500] was intended to be a global search repository with
loose, converging consistency.

It was intended to be read-mostly, perimeter secure and query-
capable.

3.3 The Global BGP Routing Information Database

The Global or top-level BGP routing information database [BGP1] is
often viewed as a global read-write repository with loose, converging
subset consistency (not all routes are carried everywhere) and very
limited integrity control, mostly intended to be perimeter integrity
based on, "access control based on what you are".

One can argue that BGP [BGP2] is better viewed as a global mechanism
for updating a set of local read/write repositories, since far from
all routing information is carried everywhere, and the decision on
what routes to accept is always considered a local policy matter.
But from a security model perspective, a lot of the controls are
applied at the periphery of the routing system, not at each local
repository; this still makes it interesting to consider properties
that apply to the BGP system as a whole.

3.4 The NetNews system

NetNews [NEWS] is a global read-write repository with loose (non-
converging) subset consistency (not all sites carry all articles, and
article retention times differ). Between sites it offers
subscription capability; to users it offers both search and lookup
functionality.

3.5 SNMP MIBs

An SNMP [SNMP] agent can be thought of as a local, centralized
repository offering lookup functionality.

With SNMPv3, it offers all kinds of access models, but mostly,
"access because of what you have", seems popular.

4. Security Considerations

   Security is a very relevant question when considering information
   access systems.

      Some issues to consider are:

      - Controlled access to information

      - Controlled rights to update information

      - Protection of the information path from provider to consumer

      - With personal information, privacy issues

      - Interactions between multiple ways to access the same
        information

   It is probably a Good Thing to consider carefully the security models
   from section 2.4 when designing repositories or repository access
   protocols.

5. Acknowledgement

   The author wishes to thank all who contributed to this document,
   including Patrik Faltstrom, Eric A. Hall, James Benedict, Ted Hardie,
   Urs Eppenberger, John Klensin, and many others.

6. References

   [SEC]   Shirey, R., "Internet Security Glossary", FYI 36, RFC
               2828, May 2000.

   [DNS]   Mockapetris, P., "Domain names - concepts and
               facilities", STD 13, RFC 1034, November 1987.

   [DNSSEC]   Eastlake, D., "Domain Name System Security Extensions",
               RFC 2535, March 1999.

   [E164]   ITU-T Recommendation E.164/I.331 (05/97): The
               International Public Telecommunication Numbering Plan.
               1997.

   [BGP1]   "Analyzing the Internet's BGP Routing Table", published in
               "The Internet Protocol Journal", Volume 4, No 1, April
               2001. At the time of writing, available at
               http://www.telstra.net/gih/papers/ipj/4-1-bgp.pdf

   [BGP2]   Rekhter, Y. and T. Li, "A Border Gateway Protocol 4
                 (BGP-4)", RFC 1771, March 1995.

   [NEWS]   Kantor, B. and P. Lapsley, "Network News Transfer
                 Protocol", RFC 977, February 1986.

   [SNMP]   Case, J., Mundy, R., Partain, D. and B. Stewart,
                 "Introduction to Version 3 of the Internet-standard
                 Network Management Framework", RFC 2570, April 1999.

   [X500]   Weider, C. and J. Reynolds, "Executive Introduction to
                 Directory Services Using the X.500 Protocol", FYI 13, RFC
                 1308, March 1992.

   [KORFHAGE] "Information Storage and Retrieval", Robert R. Korfhage,
                 Wiley 1997. See page 194 for "precision" and "recall"
                 definitions.

7. Author's Address

   Harald Tveit Alvestrand
   Cisco Systems
   Weidemanns vei 27
   N-7043 Trondheim
   NORWAY

   Phone: +47 41 44 29 94
   EMail: Harald@alvestrand.no

8. Full Copyright Statement

Acknowledgement