                    WHITE PAGES MEETING REPORT

STATUS OF THIS MEMO

   This memo provides information for the Internet community. This memo
   does not specify an Internet standard of any kind. Distribution of
   this memo is unlimited.

INTRODUCTION

   This report describes the results of a meeting held at the November
   IETF (Internet Engineering Task Force) in Houston, TX, on November 2,
   1993, to discuss the future of and approaches to a white pages
   directory services for the Internet.

   As proposed to the National Science Foundation (NSF), USC/Information
   Sciences Institute (ISI) conducted the meeting to discuss the
   viability of the X.500 directory as a practical approach to providing
   white pages service for the Internet in the near future and to
   identify and discuss any alternatives.

   An electronic mail mailing list was organized and discussions were
   held via email for two weeks prior to the meeting.

1. EXECUTIVE SUMMARY

   This report is organized around four questions:

   1) What functions should a white pages directory perform?

      There are two functions the white pages service must provide:
      searching and retrieving.

      Searching is the ability to find people given some fuzzy
      information about them. Such as "Find the Postel in southern
      California". Searches may often return a list of matches.

      While the idea of indexing has been around for some time, such as
      the IN-ADDR tree in the Domain Name System (DNS), a new
      acknowledgment of its importance has emerged from these

discussions. Users want fast searching across the distributed
database on attributes different from the database structure.
Pre-computed indices satisfy this desire, though only for
specified searches.

Retrieval is obtaining additional information associated with a
person, such as an address, telephone number, email mailbox, or
security certificate.

Security certificates (a type of information associated with an
individual) are essential for the use of end-to-end
authentication, integrity, and privacy in Internet applications.
The development of secure applications in the Internet is
dependent on a directory system for retrieving the security
certificate associated with an individual. For example, the
privacy enhanced electronic mail (PEM) system has been developed
and is ready to go into service, and is now hindered by the lack
of an easily used directory of security certificates. An open
question is whether or not such a directory needs to be internally
secure.

2) What approaches will provide us with a white pages directory?

It is evident that there are and will be several technologies in
use. In order to provide a white pages directory service that
accommodates multiple technologies, we should promote
interoperation and work toward a specification of the simplest
common communication form that is powerful enough to provide the
necessary functionality. This "common ground" approach aims to
provide the ubiquitous WPS (White Pages Service) with a high
functionality and a low entry cost.

3) What are the problems to be overcome?

It must be much easier to be part of the Internet white pages than
to bring up a X.500 DSA (Directory Service Agent), yet we must
make good use of the already deployed X.500 DSAs. Simpler white
pages services (such as Whois++) must be defined to promote
multiple implementations. To promote reliable operation, there
must be some central management of the X.500 system. A common
naming scheme must be identified and documented. A set of index-
servers, and indexing techniques, must be developed. The storage
and retrieval of security certificates must be provided.

4) What should the deployment strategy be?

   Some central management must be provided, and easy to use user
   interfaces (such as the Gopher "gateway"), must be widely
   deployed. The selection of a naming scheme must be documented.
   We should capitalize on the existing infrastructure of already
   deployed X.500 DSAs. The "common ground" model should be adopted.
   A specification of the simplest common communication form must be
   developed. Information about how to set up a new server (of
   whatever kind) in "cookbook" form should be made available.

RECOMMENDATIONS

   1. Adopt the common ground approach. Encourage multiple client and
       server types, and the standardization of an interoperation
       protocol between them. The clients may be simple clients,
       front-ends, "gateways", or embedded in other information access
       clients, such as Gopher or WWW (World Wide Web) client programs.
       The interoperation protocol will define message types, message
       sequences, and data fields. An element of this protocol should
       be the use of Universal Record Locators (URLs).

   2. Promote the development of index-servers. The index-servers
       should use several different methods both for gathering data for
       their indices, and for searching their indices.

   3. Support a central management for the X.500 system. To get the
       best advantage of the effort already invested in the X.500
       directory system it is essential to provide the relatively small
       amount of central management necessary to keep the system
       functioning.

   4. Support the development of security certificate storage and
       retrieval from the white pages service. One practical approach
       is initially to focus on getting support from the existing X.500
       directory infrastructure. This effort should also include
       design and development of the storage and retrieval of security
       certificates for other white pages services, such as Whois++.

2. HISTORY

   In February 1989, a meeting on Internet white pages service was
   initiated by the FRICC (Federal Research Internet Coordinating
   Committee) and the ensuing discussions resulted in RFC 1107 [1] that
   offered some technical conclusions. Widespread deployment was to
   have taken place by mid-1992.

        RFC 1107: K. Sollins, "Plan for Internet Directory Services",
        [1].

   Several other RFCs have been written suggesting deployment strategies
   and plans for an X.500 Directory Service.

   They are:

        RFC 1275: S. Hardcastle-Kille, "Replication Requirements to
        provide an Internet Directory using X.500", [2].

        RFC 1308: C. Weider, J. Reynolds, "Executive Introduction to
        Directory Services Using the X.500 Protocol", [3].

        RFC 1309: C. Weider, J. Reynolds, S. Heker, "Technical Overview
        of Directory Services Using the X.500 Protocol", [4].

        RFC 1430: S. Hardcastle-Kille, E. Huizer, V. Cerf, R. Hobby &
        S. Kent, "A Strategic Plan for Deploying an Internet X.500
        Directory Service", [5].

   Also, a current working draft submitted by A. Jurg of SURFnet
   entitled, "Introduction to White pages services based on X.500",
   describes why we need a global white pages service and why X.500 is
   the answer [6].

   The North America Directory Forum (NADF) also has done some useful
   work setting conventions for commercial providers of X.500 directory
   service. Their series of memos is relevant to this discussion. (See
   RFC 1417 for an overview of this note series [7].) In particular,
   NADF standing document 5 (SD-5) "An X.500 Naming Scheme for National
   DIT Subtrees and its Application for c=CA and c=US" is of interest
   for its model of naming based on civil naming authorities [8].

   Deployment of a X.500 directory service including that under the PSI
   (Performance Systems International) White Pages Pilot Project and the
   PARADISE Project is significant, and continues to grow, albeit at a
   slower rate than the Internet.

3. QUESTIONS

   Four questions were posed to the discussion list:

      1) What functions should a white pages directory perform?

      2) What approaches will provide us with a white pages directory?

      3) What are the problems to be overcome?

      4) What should the deployment strategy be?

3.A. WHAT FUNCTIONS SHOULD A WHITE PAGES DIRECTORY PERFORM?

   The basic function of a white pages service is to find people and
   information about people.

   In finding people, the service should work fast when searching for
   people by name, even if the information regarding location or
   organization is vague. In finding information about people, the
   service should retrieve information associated with people, such as a
   phone number, a postal or email address, or even a certificate for
   security applications (authentication, integrity, and privacy).
   Sometimes additional information associated with people is provided
   by a directory service, such as a list of publications, a description
   of current projects, or a current travel itinerary.

   Back in 1989, RFC 1107 detailed 8 requirements of a white pages
   service: (1) functionality, (2) correctness of information, (3) size,
   (4) usage and query rate, (5) response time, (6) partitioned
   authority, (7) access control, (8) multiple transport protocol
   support; and 4 additional features that would make it more useful:
   (1) descriptive naming that could support a yellow pages service, (2)
   accountability, (3) multiple interfaces, and (4) multiple clients.

   Since the writing of RFC 1107, many additional functions have been
   identified. A White Pages Functionality List is attached as Appendix
   1. The problem is harder now, the Internet is much bigger, and there
   are many more options available (Whois++, Netfind, LDAP (Lightweight
   Direct Access Protocol), different versions of X.500 implementations,
   etc.)

   A white pages directory should be flexible, should have low resource
   requirements, and should fit into other systems that may be currently
   in use; it should not cost a lot, so that future transitions are not
   too costly; there should be the ability to migrate to something else,
   if a better solution becomes available; there should be a way to
   share local directory information with the Internet in a seamless

fashion and with little extra effort; the query responses should be
reliable enough and consistent enough that automated tools could be
used.

3.B. WHAT APPROACHES WILL PROVIDE US WITH A WHITE PAGES DIRECTORY?

People have different needs, tastes, etc. Consequently, a large part
of the ultimate solution will include bridging among these various
solutions. Already we see a Gopher to X.500 gateway, a Whois++ to
X.500 gateway, and the beginnings of a WWW to X.500 gateway. Gopher
can talk to CSO (a phonebook service developed by University of
Illinois), WAIS (Wide Area Information Server), etc. WWW can talk to
everything. Netfind knows about several other protocols.

Gopher and WAIS "achieved orbit" simply by providing means for people
to export and to access useful information; neither system had to
provide ubiquitous service. For white pages, if the service doesn't
provide answers to specific user queries some reasonable proportion
of the time, users view it as as failure. One way to achieve a high
hit rate in an exponentially growing Internet is to use a proactive
data gathering architecture (e.g., as realized by archie and
Netfind). Important as they are, replication, authentication, etc.,
are irrelevant if no one uses the service.

There are pluses and minuses to a proactive data gathering method.
On the plus side, one can build a large database quickly. On the
minus side, one can get garbage in the database. One possibility is
to use a proactive approach to (a) acquire data for administrative
review before being added to the database, and/or (b) to check the
data for consistency with the real world. Additionally, there is
some question about the legality of proactive methods in some
countries.

One solution is to combine existing technology and infrastructure to
provide a good white pages service, based on a X.500 core plus a set
of additional index/references servers. DNS can be used to "refer"
to the appropriate zone in the X.500 name space, using WAIS or
Whois++, to build up indexes to the X.500 server which will be able
to process a given request. These can be index-servers or centroids
or something new.

Some X.500 purists might feel this approach muddles the connecting
fabric among X.500 servers, since the site index, DNS records, and
customization gateways are all outside of X.500. On the other hand,
making X.500 reachable from a common front-end would provide added
incentive for sites to install X.500 servers. Plus, it provides an
immediate (if interim) solution to the need for a global site index
in X.500. Since the goal is to have a good white pages service,

X.500 purity is not essential.

It may be that there are parts of the white pages problem that cannot
be addressed without "complex technology". A solution that allows
the user to progress up the ladder of complexity, according to taste,
perceived need, and available resources may be a much healthier
approach. However, experience to date with simpler solutions
(Whois++, Netfind, archie) indicates that a good percentage of the
problem of finding information can be addressed with simpler
approaches. Users know this and will resist attempts to make them
pay the full price for the full solution when it is not needed.
Whereas managers and funders may be concerned with the complexity of
the technology, users are generally more concerned with the quality
and ease of use of the service. A danger in supporting a mix of
technologies is that the service may become so variable that the
loose constraints of weak service in some places lead users to see
the whole system as too loose and weak.

Some organizations will not operate services that they cannot get for
free or they cannot try cheaply before investing time and money.
Some people prefer a bare-bones, no support solution that only gives
them 85 percent of what they want. Paying for the service would not
be a problem for many sites, once the value of the service has been
proven. Although there is no requirement to provide free software
for everybody, we do need viable funding and support mechanisms. A
solution can not be simply dictated with any expectation that it will
stick.

Finally, are there viable alternative technologies to X.500 now or do
we need to design something new? What kind of time frame are we
talking about for development and deployment? And will the new
technology be extensible enough to provide for the as yet unimagined
uses that will be required of directory services 5 years from now?
And will this directory service ultimately provide more capabilities
than just white pages?

3.C. WHAT ARE THE PROBLEMS TO BE OVERCOME?

There are two classes of problems to be examined; technology issues
and infrastructure.

TECHNOLOGY:

How do we populate the database and make software easily available?

Many people suggest that a public domain version of X.500 is
necessary before a wide spread X.500 service is operational. The
current public domain version is said to be difficult to install and

to bring into operation, but many organizations have successfully
installed it and have had their systems up and running for some time.
Note that the current public domain program, quipu, is not quite
standard X.500, and is more suited to research than production
service. Many people who tried earlier versions of quipu abandoned
X.500 due to its costly start up time, and inherent complexity.

The ISODE (ISO Development Environment) Consortium is currently
developing newer features and is addressing most of the major
problems. However, there is the perception that the companies in the
consortium have yet to turn these improvements into actual products,
though the consortium says the companies have commercial off-the-
shelf (COTS) products available now. The improved products are
certainly needed now, since if they are too late in being deployed,
other solutions will be implemented in lieu of X.500.

The remaining problem with an X.500 White Pages is having a high
quality public domain DSA. The ISODE Consortium will make its
version available for no charge to Universities (or any non-profit or
government organization whose primary purpose is research) but if
that leaves a sizeable group using the old quipu implementation, then
there is a significant problem. In such a case, an answer may be for
some funding to upgrade the public version of quipu.

In addition, the quipu DSA should be simplified so that it is easy to
use. Tim Howes' new disk-based quipu DSA solves many of the memory
problems in DSA resource utilization. If one fixes the DSA resource
utilization problem, makes it fairly easy to install, makes it freely
available, and publishes a popular press book about it, X.500 may
have a better chance of success.

The client side of X.500 needs more work. Many people would rather
not expend the extra effort to get X.500 up. X.500 takes a sharp
learning curve. There is a perception that the client side also
needs a complex Directory User Interface (DUI) built on ISODE. Yet
there are alternative DUIs, such as those based on LDAP. Another
aspect of the client side is that access to the directory should be
built into other applications like gopher and email (especially,
accessing PEM X.509 certificates).

We also need data conversion tools to make the transition between
different systems possible. For example, NASA had more than one
system to convert.

Searching abilities for X.500 need to be improved. LDAP is great
help, but the following capabilities are still needed:

      -- commercial grade easily maintainable servers with back-end
         database support.

      -- clients that can do exhaustive search and/or cache useful
         information and use heuristics to narrow the search space in case
         of ill-formed queries.

      -- index servers that store index information on a "few" key
         attributes that DUIs can consult in narrowing the search space.
         How about index attributes at various levels in the tree that
         capture the information in the corresponding subtree?

   Work still needs to be done with Whois++ to see if it will scale to
   the level of X.500.

   An extended Netfind is attractive because it would work without any
   additional infrastructure changes (naming, common schema, etc.), or
   even the addition of any new protocols.

   INFRASTRUCTURE:

   The key issues are central management and naming rules.

   X.500 is not run as a service in the U.S., and therefore those using
   X.500 in the U.S. are not assured of the reliability of root servers.
   X.500 cannot be taken seriously until there is some central
   management and coordinated administration support in place. Someone
   has to be responsible for maintaining the root; this effort is
   comparable to maintaining the root of the DNS. PSI provided this
   service until the end of the FOX project [9]; should they receive
   funding to continue this? Should this be a commercial enterprise?
   Or should this function be added to the duties of the InterNIC?

   New sites need assistance in getting their servers up and linked to a
   central server.

   There are two dimensions along which to consider the infrastructure:
   1) general purpose vs. specific, and 2) tight vs. loose information
   framework.

   General purpose leads to more complex protocols - the generality is
   an overhead, but gives the potential to provide a framework for a
   wide variety of services. Special purpose protocols are simpler, but
   may lead to duplication or restricted scope.

   Tight information framework costs effort to coerce existing data and
   to build structures. Once in place, it gives better managability and
   more uniform access. The tight information framework can be

   subdivided further into: 1) the naming approach, and 2) the object
   and attribute extensibility.

   Examples of systems placed in this space are: a) X.500 is a general
   purpose and tight information framework, b) DNS is a specific and
   tight information framework, c) there are various research efforts in
   the general purpose and loose information framework, and d) Whois++
   employs a specific and loose information framework.

   We need to look at which parts of this spectrum we need to provide
   services. This may lead to concluding that several services are
   desirable.

3.D. WHAT SHOULD THE DEPLOYMENT STRATEGY BE?

   No solution will arise simply by providing technical specifications.
   The solution must fit the way the Internet adopts information
   technology. The information systems that have gained real momentum
   in the Internet (WAIS, Gopher, etc.) followed the model:

   -- A small group goes off and builds a piece of software that
      supplies badly needed functionality at feasible effort to
      providers and users.

   -- The community rapidly adopts the system as a de facto standard.

   -- Many people join the developers in improving the system and
      standardizing the protocols.

   What can this report do to help make this happen for Internet white
   pages?

   Deployment Issues.

   -- A strict hierarchical layout is not suitable for all directory
      applications and hence we should not force fit it.

   -- A typical organization's hierarchical information itself is often
      proprietary; they may not want to divulge it to the outside world.

      It will always be true that Institutions (not just commercial)
      will always have some information that they do not wish to display
      to the public in any directory. This is especially true for
      Institutions that want to protect themselves from headhunters, and
      sales personnel.

   -- There is the problem of multiple directory service providers, but
      see NADF work on "Naming Links" and their "CAN/KAN" technology
      [7].

      A more general approach such as using a knowledge server (or a set
      of servers) might be better. The knowledge servers would have to
      know about which server to contact for a given query and thus may
      refer to either service provider servers or directly to
      institution-operated servers. The key problem is how to collect
      the knowledge and keep it up to date. There are some questions
      about the viability of "naming links" without a protocol
      modification.

   -- Guidelines are needed for methods of searching and using directory
      information.

   -- A registration authority is needed to register names at various
      levels of the hierarchy to ensure uniqueness or adoption of the
      civil naming structure as delineated by the NADF.

It is true that deployment of X.500 has not seen exponential growth
as have other popular services on the Internet. But rather than
abandoning X.500 now, these efforts, which are attempting to address
some of the causes, should continue to move forward. Certainly
installation complexity and performance problems with the quipu
implementation need solutions. These problems are being worked on.

One concern with the X.500 service has been the lack of ubiquitous
user agents. Very few hosts run the ISODE package. The use of LDAP
improves this situation. The X.500-gopher gateway has had the
greatest impact on providing wide-spread access to the X.500 service.
Since adding X.500 as a service on the ESnet Gopher, the use of the
ESnet DSA has risen dramatically.

Another serious problem affecting the deployment of X.500, at least
in the U.S., is the minimal support given to building and maintaining
the necessary infrastructure since the demise of the Fox Project [9].
Without funding for this effort, X.500 may not stand a chance in the
United States.

4. REVIEW OF TECHNOLOGIES

   There are now many systems for finding information, some of these are
   oriented to white pages, some include white pages, and others
   currently ignore white pages. In any case, it makes sense to review
   these systems to see how they might fit into the provision of an
   Internet white pages service.

4.A. X.500

   Several arguments in X.500's favor are its flexibility, distributed
   architecture, security, superiority to paper directories, and that it
   can be used by applications as well as by humans. X.500 is designed
   to provide a uniform database facility with replication,
   modification, and authorization. Because it is distributed, it is
   particularly suited for a large global White Pages directory. In
   principle, it has good searching capabilities, allowing searches at
   any level or in any subtree of the DIT (Directory Information Tree).
   There are DUIs available for all types of workstations and X.500 is
   an international standard. In theory, X.500 can provide vastly
   better directory service than other systems, however, in practice,
   X.500 is difficult, too complicated, and inconvenient to use. It
   should provide a better service. X.500 is a technology that may be
   used to provide a white pages service, although some features of
   X.500 may not be needed to provide just a white pages service.

   The are three reasons X.500 deployment has been slow, and these are
   largely the same reasons people don't like it:

   1) The available X.500 implementations (mostly quipu based on the
      ISODE) are very large and complicated software packages that are
      hard to work with. This is partly because they solve the general
      X.500 problem, rather than the subset needed to provide an
      Internet white pages directory. In practice, this means that a
      portion of the code/complexity is effectively unused.

      The LDAP work has virtually eliminated this concern on the client
      side of things, as LDAP is both simple and lightweight. Yet, the
      complexity problem still exists on the server side of things, so
      people continue to have trouble bringing up data for simple
      clients to access.

      It has been suggested that the complexity in X.500 is due to the
      protocol stack and the ISODE base. If this is true, then LDAP may
      be simple because it uses TCP directly without the ISODE base. A
      version of X.500 server that took the same approach might also be
      "simple" or at least simpler. Furthermore, the difficulty in
      getting an X.500 server up may be related to finding the data to

put in the server, and so may be a general data management problem
rather than an X.500 specific problem.

There is some evidence that eventually a large percentage of the
use of directory services may be from applications rather than
direct user queries. For example, mail-user-agents exist that are
X.500 capable with an integrated DUA (Directory User Agent).

2) You have to "know a lot" to get a directory service up and running
   with X.500. You have to know about object classes and attributes
   to get your data into X.500. You have to get a distinguished name
   for your organization and come up with an internal tree structure.
   You have to contact someone before you can "come online" in the
   pilot. It's not like gopher where you type "make", tell a few
   friends, and you're up and running.

   Note that a gopher server is not a white pages service, and as
   noted elsewhere in this report, there are a number of issues that
   apply to white pages service that are not addressed by gopher.

   Some of these problems could be alleviated by putting in place
   better procedures. It should not any be harder to get connected
   to X.500 than it is to get connected to the DNS, for example.
   However, there is a certain amount of complexity that may be
   inherent in directory services. Just compare Whois++ and X.500.
   X.500 has object classes. Whois++ has templates. X.500 has
   attributes. Whois++ has fields. X.500 has distinguished names.
   Whois++ has handles.

3) Getting data to populate the directory, converting it into the
   proper form, and keeping it up-to-date turns out to be a hard
   problem. Often this means talking to the administrative computing
   department at your organization.

   This problem exists regardless of the protocol used. It should be
   easy to access this data through the protocol you're using, but
   that says more about implementations than it does about the
   protocol. Of course, if the only X.500 implementation you have
   makes it really hard to do, and the Whois++ implementation you
   have makes it easy, it's hard for that not to reflect on the
   protocols.

The fact that there are sites like University of Michigan, University
of Minnesota, Rutgers University, NASA, LBL, etc. running X.500 in
serious production mode shows that the problem has more to do with
the current state of X.500 software procedures. It takes a lot of
effort to get it going. The level of effort required to keep it
going is relatively very small.

The yellow pages problem is not really a problem. If you look at it
in the traditional phonebook-style yellow pages way, then X.500 can
do the job just like the phone book does. Just organize the
directory based on different (i.e., non-geographical) criteria. If
you want to "search everything", then you need to prune the search
space. To do this you can use the Whois++ centroids idea, or
something similar. But this idea is as applicable to X.500 as it is
to Whois++. Maybe X.500 can use the centroids idea most effectively.

Additionally, it should be noted that there is not one single Yellow
Pages service, but that according to the type of query there could be
several such as querying by role, by location, by email address.

No one is failing to run X.500 because they perceive it fails to
solve the yellow pages problem. The reasons are more likely one or
more of the three above.

X.500's extra complexity is paying off for University of Michigan.
University of Michigan started with just people information in their
tree. Once that infrastructure was in place, it was easy for them to
add more things to handle mailing lists/email groups, yellow pages
applications like a documentation index, directory of images, etc.

The ESnet community is using X.500 right now to provide a White Pages
service; users succeed everyday in searching for information about
colleagues given only a name and an organizational affiliation; and
yes, they do load data into X.500 from an Oracle database.

LBL finds X.500 very useful. They can lookup DNS information, find
what Zone a Macintosh is in, lookup departmental information, view
the current weather satellite image, and lookup people information.

LDAP should remove many of the complaints about X.500. Implementing
a number of LDAP clients is very easy and has all the functionality
needed. Perhaps DAP should be scrapped.

Another approach is the interfacing of X.500 servers to WWW (the
interface is sometimes called XWI). Using the mosaic program from
the NCSA, one can access X.500 data.

INTERNET X.500

The ISO/ITU may not make progress on improving X.500 in the time
frame required for an Internet white pages service. One approach is
to have the Internet community (e.g., the IETF) take responsibility
for developing a subset or profile of that part of X.500 it will use,
and developing solutions for the ambiguous and undefined parts of
X.500 that are necessary to provide a complete service.

Tasks this approach might include are:

1. Internet (IETF) control of the base of the core service white
   pages infrastructure and standard.

2. Base the standard on the 1993 specification, especially
   replication and access control.

3. For early deployment choose which parts of the replication
   protocol are really urgently needed. It may be possible to define
   a subset and to make it mandatory for the Internet.

4. Define an easy and stable API (Application Program Interface) for
   key access protocols (DAP, LDAP).

5. Use a standard knowledge model.

6. Make sure that high performance implementations will exist for the
   most important servers, roles principally for the upper layers of
   the DSA tree.

7. Make sure that servers will exist that will be able to efficiently
   get the objects (or better the attributes) from existing
   traditional databases for use at the leaves of the DSA tree.

4.B. WHOIS++

The very first discussions of this protocol started in July 1992. In
less than 15 months there were 3 working public domain
implementations, at least 3 more are on the way, and a Whois++
front-end to X.500. In addition, the developers who are working on
the resource location system infrastructure (URL/URI) have committed
to implementing it on top of Whois++ because of its superior search
capabilities.

Some of the main problems with getting a White Pages directory going
have been: (1) search, (2) lack of public domain versions, (3)
implementations are too large, (4) high start up cost, and (5) the
implementations don't make a lot of sense for a local directory,
particularly for small organizations. Whois++ can and does address
all these problems very nicely.

Search is built into Whois++, and there is a strong commitment from
the developers to keep this a high priority.

The protocols are simple enough that someone can write a server in 3
days. And people have done it. If the protocols stay simple, it
will always be easy for someone to whip out a new public domain
server. In this respect, Whois++ is much like WAIS or Gopher.

The typical Whois++ implementation is about 10 megabytes, including
the WAIS source code that provides the data engine. Even assuming a
rough doubling of the code as additional necessary functionality is
built in, that's still quite reasonable, and compares favorably with
the available implementations of X.500. In addition, WAIS is disk-
based from the start, and is optimized for local searching. Thus,
this requires only disk storage for the data and the indexes. In a
recent test, Chris Weider used a 5 megabyte source data file with the
Whois++ code. The indices came to about another 7 megabytes, and the
code was under 10 megabytes. The total is 22 megabytes for a Whois++
server.

The available Whois++ implementations take about 25 minutes to
compile on a Sun SPARCstation IPC. Indexing a 5 megabyte data file
takes about another 20 minutes on an IPC. Installation is very easy.
In addition, since the Whois++ server protocol is designed to be only
a front-end, organizations can keep their data in any form they want.

Whois++ makes sense as a local directory service. The
implementations are small, install quickly, and the raw query
language is very simple. The simplicity of the interaction between
the client and the server make it easy to experiment with and to
write clients for, something that wasn't true of X.500 until LDAP.
In addition, Whois++ can be run strictly as a local service, with
integration into the global infrastructure done at any time.

It is true that Whois++ is not yet a fully functional White Pages
service. It requires a lot of work before it will be so. However,
X.500 is not that much closer to the goal than Whois++ is.

Work needs to be done on replication and authentication of data. The
current Whois++ system does not lend itself to delegation. Research
is still needed to improve the system and see if it scales well.

4.C. NETFIND

   Right now, the white pages service with the most coverage in the
   Internet is Mike Schwartz' Netfind. Netfind works in two stages: 1)
   find out where to ask, and 2) start asking.

   The first stage is based on a database of netnews articles, UUCP
   maps, NIC WHOIS databases, and DNS traversals, which then maps
   organizations and localities to domain names. The second stage
   consists of finger queries, Whois queries, smtp expns and vrfys, and
   DNS lookups.

   The key feature of Netfind is that it is proactive. It doesn't
   require that the system administrator bring up a new server, populate
   it with all kinds of information, keep the information in sync, worry
   about update, etc. It just works.

   A suggestion was made that Netfind could be used as a way to populate
   the X.500 directory. A tool might do a series of Netfind queries,
   making the corresponding X.500 entries as it progresses.
   Essentially, X.500 entries would be "discovered" as people look for
   them using Netfind. Others do not believe this is feasible.

   Another perhaps less interesting merger of Netfind and X.500 is to
   have Netfind add X.500 as one of the places it looks to find
   organizations (and people).

   A search can lead you to where a person has an account (e.g.,
   law.xxx.edu) only to find a problem with the DNS services for that
   domain, or the finger service is unavailable, or the machines are not
   be running Unix (there are lots of VMS machines and IBM mainframes
   still out there). In addition, there are security gateways. The
   trends in computing are towards the use of powerful portables and
   mobile computing and hence Netfind's approach may not work. However,
   Netfind proves to be an excellent yellow-pages service for domain
   information in DNS servers - given a set of keywords it lists a set
   of possible domain names.

   Suppose we store a pointer in DNS to a white-pages server for a
   domain. We can use Netfind to come up with a list of servers to
   search, query these servers, then combine the responses. However, we
   need a formal method of gathering white-pages data and informal
   methods will not work and may even get into legal problems.

The user search phase of Netfind is a short-term solution to
providing an Internet white pages. For the longer term, the
applicability of the site discovery part of Netfind is more relevant,
and more work has been put into that part of the system over the past
2 years than into the user search phase.

Given Netfind's "installed customer base" (25k queries per day, users
in 4875 domains in 54 countries), one approach that might make sense
is to use Netfind as a migration path to a better directory, and
gradually phase Netfind's user search scheme out of existence. The
idea of putting a record in the DNS to point to the directory service
to search at a site is a good start.

One idea for further development is to have the DNS record point to a
"customization" server that a site can install to tailor the way
Netfind (or whatever replaces Netfind) searches their site. This
would provide sites a choice of degrees of effort and levels of
service. The least common denominator is what Netfind presently
does: DNS/SMTP/finger. A site could upgrade by installing a
customization server that points to the best hosts to finger, or that
says "we don't want Netfind to search here" (if people are
sufficiently concerned about the legal/privacy issues, the default
could be changed so that searches must be explicitly enabled). The
next step up is to use the customization server as a gateway to a
local Whois, CSO, X.500, or home grown white pages server. In the
long run, if X.500 (or Whois++, etc.) really catches on, it could
subsume the site indexing part of Netfind and use the above approach
as an evolution path to full X.500 deployment. However, other
approaches may be more productive. One key to Netfind's success has
been not relying on organizations to do anything to support Netfind,
however the customization server breaks this model.

Netfind is very useful. Users don't have to do anything to wherever
they store their people data to have it "included" in Netfind. But
just like archie, it would be more useful if there were a more common
structure to the information it gives you, and therefore to the
information contained in the databases it accesses. It's this common
structure that we should be encouraging people to move toward.

As a result of suggestions made at the November meeting, Netfind has
been extended to make use of URL information stored in DNS records.
Based on this mechanism, Netfind can now interoperate with X.500,
WHOIS, and PH, and can also allow sites to tune which hosts Netfind
uses for SMTP or Finger, or restrict Netfind from searching their
site entirely.

4.D. ARCHIE

   Archie is a success because it is a directory of files that are
   accessible over the network. Every FTP site makes a "conscious"
   decision to make the files available for anonymous FTP over the
   network. The mechanism that archie uses to gather the data is the
   same as that used to transfer the files. Thus, the success rate is
   near 100%. In a similar vein, if Internet sites make a "conscious"
   decision to make white-pages data available over the network, it is
   possible to link these servers to create a world-wide directory, such
   as X.500, or build an index that helps to isolate the servers to be
   searched, Whois++. Users don't have to do anything to their FTP
   archives to have them included in archie. But everybody recognizes
   that it could be more useful if only there were some more common
   structure to the information, and to the information contained in the
   archives. Archie came after the anonymous FTP sites were in wide-
   spread use. Unfortunately for white-pages, we are building tools,
   but there is no data.

4.E. FINGER

   The Finger program that allows one to get either information about an
   individual with an account, or a list of currently logged in users,
   from a host running the server, can be used to check a suggestion
   that a particular individual has an account on a particular host.
   This does not provide an efficient method to search for an
   individual.

4.F. GOPHER

   A "gateway" between Gopher and X.500 has been created so that one can
   examine X.500 data from a Gopher client. Similar "gateways" are
   needed for other white pages systems.

4.G. WWW

   One extension to WWW would be an attribute type for the WWW URI/URL
   with the possibility for any client to request from the X.500 server
   (1) either the locator (thus the client would decide to access or not
   the actual data), or (2) for client not capable of accessing this
   data, the data itself (packed) in the ASN.1 encoded result.

   This would give access to potentially any piece of information
   available on the network through X.500, and in the white pages case
   to photos or voice messages for persons.

This solution is preferable to one consisting of storing this
multimedia information directly in the directory, because it allows
WWW capable DUIs to access directly any piece of data no matter how
large. This work on URIs is not WWW-specific.

5. ISSUES

5.A. DATA PROTECTION

Outside of the U.S., nearly all developed countries have rather
strict data protection acts (to ensure privacy mostly) that governs
any database on personal data.

It is mandatory for the people in charge of such white pages
databases to have full control over the information that can be
stored and retrieved in such a database, and to provide access
controls over the information that is made available.

If modification is allowed, then authentication is required. The
database manager must be able to prevent users from making available
unallowed information.

When we are dealing with personal records the issues are a little
more involved than exporting files. We can not allow trawling of
data and we need access-controls so that several applications can use
the directory and hence we need authentication.

X.500 might have developed faster if security issues were not part of
the implementation. There is tension between quick lightweight
implementations and the attempt to operate in a larger environment
with business issues incorporated. The initial belief was that data
is owned by the people who put the data into the system, however,
most data protection laws appoint the organizations holding the data
responsible for the quality of the data of their individuals.
Experience also shows that the people most affected by inaccurate
data are the people who are trying to access the data. These
problems apply to all technologies.

5.B. STANDARDS

Several types of standards are needed: (1) standards for
interoperation between different white pages systems (e.g., X.500 and
Whois++), (2) standards for naming conventions, and (3) and standards
within the structured data of each system (what fields or attributes
are required and optional, and what are their data types).

The standards for interoperation may be developed from the work now
in progress on URLs, with some additional protocol developed to
govern the types of messages and message sequences.

Both the naming of the systems and the naming of individuals would
benefit from consistent naming conventions. The use of the NADF
naming scheme should be considered.

When structured data is exchanged, standards are needed for the data
types and the structural organization. In X.500, much effort has
gone into the definition of various structures or schemas, and yet
few standard schemas have emerged.

There is a general consensus that a "cookbook" for Administrators
would make X.500 implementation easier and more attractive. These
are essential for getting X.500 in wider use. It is also essential
that other technologies such as Whois++, Netfind, and archie also
have complete user guides available.

5.C. SEARCHING AND RETRIEVING

The main complaint, especially from those who enjoyed using a
centralized database (such as the InterNIC Whois service), is the
need to search for all the John Doe's in the world. Given that the
directory needs to be distributed, there is no way of answering this
question without incurring additional cost.

This is a problem with any distributed directory - you just can't
search every leaf in the tree in any reasonable amount of time. You
need to provide some mechanism to limit the number of servers that
need to be contacted. The traditional way to handle this is with
hierarchy. This requires the searcher to have some idea of the
structure of the directory. It also comes up against one of the
standard problems with hierarchical databases - if you need to search
based on a characteristic that is NOT part of the hierarchy, you are
back to searching every node in the tree, or you can search an index
(see below).

In general:

-- the larger the directory the more need for a distributed solution
   (for upkeep and managability).

-- once you are distributed, the search space for any given search
   MUST be limited.

-- this makes it necessary to provide more information as part of the
   query (and thus makes the directory harder to use).

   Any directory system can be used in a manner that makes searching
   less than easy. With a User Friendly Name (UFN) query, a user can
   usually find an entry (presuming it exists) without a lot of trouble.
   Using additional listings (as per NADF SD-5) helps to hide geographic
   or civil naming infrastructure knowledge requirements.

   Search power is a function of DSA design in X.500, not a function of
   Distinguished Naming. Search can be aided by addition in X.500 of
   non-distinguishing attributes, and by using the NADF Naming Scheme it
   is possible to lodge an entry anywhere in the DIT that you believe is
   where it will be looked for.

   One approach to the distributed search problem is to create another
   less distributed database to search, such as an index. This is done
   by doing a (non-interactive) pre-search, and collecting the results
   in an index. When a user wants to do a real time search, one first
   searches the index to find pointers to the appropriate data records
   in the distributed database. One example of this is the building of
   centroids that contain index information. There may be a class of
   servers that hold indices, called "index-servers".

5.D. INDEXING

   The suggestion for how to do fast searching is to do indexing. That
   is to pre-compute an index of people from across the distributed
   database and hold that index in an index server. When a user wants
   to search for someone, he first contacts the index-server. The
   index-server searches its index data and returns a pointer (or a few
   pointers) to specific databases that hold data on people that match
   the search criteria. Other systems which do something comparable to
   this are archie (for FTP file archives), WAIS, and Netfind.

5.E. COLLECTION AND MAINTENANCE

   The information must be "live" - that is, it must be used. Often one
   way to ensure this is to use the data (perhaps locally) for something
   other than white pages. If it isn't, most people won't bother to
   keep the information up to date. The white pages in the phone book
   have the advantage that the local phone company is in contact with
   the listee monthly (through the billing system), and if the address
   is not up to date, bills don't get delivered, and there is feedback
   that the address is wrong. There is even better contact for the
   phone number, since the local phone company must know that for their
   basic service to work properly. It is this aspect of directory
   functionality that leads towards a distributed directory system for
   the Internet.

One approach is to use existing databases to supply the white pages
data. It then would be helpful to define a particular use of SQL
(Structured Query Language) as a standard interface language between
the databases and the X.500 DSA or other white pages server. Then
one needs either to have the directory service access the existing
database using an interface language it already knows (e.g., SQL), or
to have tools that periodically update the directory database from
the existing database. Some sort of "standard" query format (and
protocol) for directory queries, with "standard" field names will be
needed to make this work in general. In a way, both X.500 and
Whois++ provide this. This approach implies customization at every
existing database to interface to the "standard" query format.

Some strongly believe that the white pages service needs to be
created from the bottom up with each organization supplying and
maintaining its own information, and that such information has to be
the same -- or a portion of the same -- information the organization
uses locally. Otherwise the global information will be stale and
incomplete.

One way to make this work is to distribute software that:

    - is useful locally,

    - fits into the global scheme,

    - is available free, and

    - works on most Unix systems.

With respect to privacy, it would be good for the local software to
have controls that make it possible to put company sensitive
information into the locally maintained directory and have only a
portion of it exported for outsiders.

5.F. NAMING STRUCTURE

We need a clear naming scheme capable of associating a name with
attributes, without any possible ambiguities, that is stable over
time, but also capable of coping with changes. This scheme should
have a clear idea of naming authorities and be able to store
information required by authentication mechanisms (e.g., PEM or X.509
certificates).

The NADF is working to establish a National Public Directory Service,
based on the use of existing Civil Naming Authorities to register
entry owners' names, and to deal with the shared-entry problem with a
shared public DIT supported by competing commercial service

   providers. At this point, we do not have any sense at the moment as
   to how [un]successful the NADF may be in accomplishing this.

   The NADF eventually concluded that the directory should be organized
   so entries can be found where people (or other entities) will look
   for them, not where civil naming authorities would place their
   archival name registration records.

   There are some incompatibilities between use of the NADF Naming
   Scheme, the White Pages Pilot Naming Scheme, and the PARADISE Naming
   Scheme. This should be resolved.

5.G. CLAYMAN PROPOSAL

   RFC 1107 offered a "strawman" proposal for an Internet Directory
   Service. The next step after strawman is sometimes called "clayman",
   and here a clayman proposal is presented.

   We assume only white pages service is to be provided, and we let
   sites run whatever access technologies they want to (with whatever
   access controls they feel comfortable).

   Then the architecture can be that the discovery process leads to a
   set of URLs. A URL is like an address, but it is a typed address
   with identifiers, access method, not a protocol. The client sorts
   the URLs and may discard some that it cannot deal with. The client
   talks to "meaningful URLs" (such as Whois, Finger, X.500).

   This approach results in low entry cost for the servers that want to
   make information available, a Darwinian selection of access
   technologies, coalescence in the Internet marketplace, and a white
   pages service will tend toward homogeneity and ubiquity.

   Some issues for further study are what discovery technology to use
   (Netfind together with Whois++ including centroids?), how to handle
   non-standard URLs (one possible solution is to put server on top of
   these (non-standard URLs) which reevaluates the pointer and acts as a
   front-end to a database), which data model to use (Finger or X.500),
   and how to utilize a common discovery technology (e.g., centroids) in
   a multiprotocol communication architecture.

   The rationale for this meta-WPS approach is that it builds on current
   practices, while striving to provide a ubiquitous directory service.
   Since there are various efforts going on to develop WPS based on
   various different protocols, one can envisage a future with a meta-
   WPS that uses a combination of an intelligent user agent and a
   distributed indexing service to access the requested data from any
   available WPS. The user perceived functionality of such a meta-WPS

will necessarily be restricted to the lowest common denominator. One
will hope that through "market" forces, the number of protocols used
will decrease (or converge), and that the functionality will
increase.

The degree to which proactive data gathering is permitted may be
limited by national laws. It may be appropriate to gather data about
which hosts have databases, but not about the data in those
databases.

6. CONCLUSIONS

   We now revisit the questions we set out to answer and briefly
   describe the key conclusions.

6.A. WHAT FUNCTIONS SHOULD A WHITE PAGES DIRECTORY PERFORM?

   After all the discussion we come to the conclusion that there are two
   functions the white pages service must provide: searching and
   retrieving.

   Searching is the ability to find people given some fuzzy information
   about them. Such as "Find the Postel in southern California".
   Searches may often return a list of matches.

   The recognition of the importance of indexing in searching is a major
   conclusion of these discussions. It is clear that users want fast
   searching across the distributed database on attributes different
   from the database structure. It is possible that pre-computed
   indices can satisfy this desire.

   Retrieval is obtaining additional information associated with a
   person, such as address, telephone number, email mailbox, and
   security certificate.

   This last, security certificates, is a type of information associated
   with an individual that is essential for the use of end-to-end
   authentication, integrity, and privacy, in Internet applications.
   The development of secure application in the Internet is dependent on
   a directory system for retrieving the security certificate associated
   with an individual. The PEM system has been developed and is ready
   to go into service, but is now held back by the lack of an easily
   used directory of security certificates.

   PEM security certificates are part of the X.509 standard. If X.500
   is going to be set aside, then other alternatives need to be
   explored. If X.500 distinguished naming is scrapped, some other
   structure will need to come into existence to replace it.

6.B.   WHAT APPROACHES WILL PROVIDE US WITH A WHITE PAGES DIRECTORY?

   It is clear that there will be several technologies in use. The
   approach must be to promote the interoperation of the multiple
   technologies. This is traditionally done by having conventions or
   standards for the interfaces and communication forms between the
   different systems. The need is for a specification of the simplest
   common communication form that is powerful enough to provide the
   necessary functionality. This allows a variety of user interfaces on
   any number of client systems communicating with different types of
   servers. The IETF working group (WG) method of developing standards
   seems well suited to this problem.

   This "common ground" approach aims to provide the ubiquitous WPS with
   a high functionality and a low entry cost. This may done by singling
   out issues that are common for various competing WPS and coordinate
   work on these in specific and dedicated IETF WGs (e.g., data model
   coordination). The IETF will continue development of X.500 and
   Whois++ as two separate entities. The work on these two protocols
   will be broken down in various small and focussed WGs that address
   specific technical issues, using ideas from both X.500 and Whois++.
   The goal being to produce common standards for information formats,
   data model and access protocols. Where possible the results of such
   a WG will be used in both Whois++ and X.500, although it is envisaged
   that several WGs may work on issues that remain specific to one of
   the protocols. The IDS (Integrated Directory Services) WG continues
   to work on non-protocol specific issues. To achieve coordination
   that leads to convergence rather than divergence, the applications
   area directorate will provide guidance to the Application Area
   Directors as well as to the various WGs, and the User Services Area
   Council (USAC) will provide the necessary user perspective.

6.C.   WHAT ARE THE PROBLEMS TO BE OVERCOME?

   There are several problems that can be solved to make progress
   towards a white pages service more rapid. We need:

   To make it much easier to be part of the Internet white pages than
   bringing up a X.500 DSA, yet making good use of the already deployed
   X.500 DSAs.

   To define new simpler white pages services (such as Whois++) such
   that numerous people can create implementations.

   To provide some central management of the X.500 system to promote
   good operation.

   To select a naming scheme.

To develop a set of index-servers, and indexing techniques, to
provide for fast searching.

To provide for the storage and retrieval of security certificates.

6.D.   WHAT SHOULD THE DEPLOYMENT STRATEGY BE?

We should capitalize on the existing infrastructure of already
deployed X.500 DSAs. This means that some central management must be
provided, and easy to use user interfaces (such as the Gopher
"gateway"), must be widely deployed.

-- Document the selection of a naming scheme (e.g., the NADF scheme).

-- Adopt the "common ground" model. Encourage the development of
   several different services, with a goal of interworking between
   them.

-- Develop a specification of the simplest common communication form
   that is powerful enough to provide the necessary functionality.
   The IETF working group method of developing standards seems well
   suited to this problem.

-- Make available information about how to set up new servers (of
   what ever kind) in "cookbook" form.

7. SUMMARY

   While many issues have been raised, there are just a few where we
   recommend the action be taken to support specific elements of the
   overall white pages system.

   RECOMMENDATIONS

   1. Adopt the common ground approach - give all protocols equal
       access to all data. That is, encourage multiple client and
       server types, and the standardization of an interoperation
       protocol between them. The clients may be simple clients,
       front-ends, "gateways", or embedded in other information access
       clients, such as Gopher or WWW client programs. The
       interoperation protocol will define some message types, message
       sequences, and data fields.   An element of this protocol should
       be the use of URLs.

   2. Promote the development of index-servers. The index-servers
       should use several different methods of gathering data for their
       indices, and several different methods for searching their
       indices.

   3. Support a central management for the X.500 system. To get the
       best advantage of the effort already invested in the X.500
       directory system it is essential to provide the relatively small
       amount of central management necessary to keep the system
       functioning.

   4. Support the development of security certificate storage and
       retrieval from the white pages service. The most practical
       approach is to initially focus on getting this supported by the
       existing X.500 directory infrastructure. It should also include
       design and development of the storage and retrieval of security
       certificates in other white pages services, such as Whois++.

8. REFERENCES

   [1] Sollins, K., "Plan for Internet Directory Services", RFC 1107,
        M.I.T. Laboratory for Computer Science, July 1989.

   [2] Hardcastle-Kille, S., "Replication Requirements to provide an
        Internet Directory using X.500, RFC 1275, University College
        London, November 1991.

   [3] Weider, C., and J. Reynolds, "Executive Introduction to
        Directory Services Using the X.500 Protocol", FYI 13, RFC 1308,
        ANS, USC/Information Sciences Institute, March 1992.

   [4] Weider, C., Reynolds, J., and S. Heker, "Technical Overview of
        Directory Services Using the X.500 Protocol", FYI 14, RFC 1309,
        ANS, USC/Information Sciences Institute,, JvNC, March 1992.

   [5] Hardcastle-Kille, S., Huizer, E., Cerf, V., Hobby, R., and S.
        Kent, "A Strategic Plan for Deploying an Internet X.500
        Directory Service", RFC 1430, ISODE Consortium, SURFnet bv,
        Corporation for National Research Initiatives, University of
        California, Davis, Bolt, Beranek, and Newman, February 1993.

   [6] Jurg, A., "Introduction to White pages services based on X.500",
        Work in Progress, October 1993.

   [7] The North American Directory Forum, "NADF Standing Documents: A
        Brief Overview", RFC 1417, The North American Directory Forum",
        NADF, February 1993.

   [8] NADF, An X.500 Naming Scheme for National DIT Subtrees and its
        Application for c=CA and c=US", Standing Document 5 (SD-5).

   [9] Garcia-Luna, J., Knopper, M., Lang, R., Schoffstall, M.,
        Schraeder, W., Weider, C., Yeong, W, Anderson, C., (ed.), and J.
        Postel (ed.), "Research in Directory Services: Fielding
        Operational X.500 (FOX)", Fox Project Final Report, January
        1992.

9. GLOSSARY

         API - Application Program Interface
         COTS - commercial off the shelf
         CSO - a phonebook service developed by University of Illinois
         DAP - Direct Access Protocol
         DIT - Directory Information Tree
         DNS - Domain Name System
         DUI - Directory User Interface
         DUA - Directory User Agent
         DSA - Directory Service Agent
         FOX - Fielding Operational X.500 project
         FRICC - Federal Research Internet Coordinating Committee
         IETF - Internet Engineering Task Force
         ISODE - ISO Development Environment
         LDAP - Lightweight Direct Access Protocol
         NADF - North American Directory Forum
         PEM - Privacy Enhanced Mail
         PSI - Performance Systems International
         SQL - Structured Query Language
         QUIPU - an X.500 DSA which is a component of the ISODE package
         UFN - User Friendly Name
         URI - Uniform Resource Identifier
         URL - Uniform Resource Locator
         WAIS - Wide Area Information Server
         WPS - White Pages Service
         WWW - World Wide Web

9. ACKNOWLEDGMENTS

   This report is assembled from the words of the following participants
   in the email discussion and the meeting. The authors are responsible
   for selecting and combining the material. Credit for all the good
   ideas goes to the participants. Any bad ideas are the responsibility
   of the authors.

        Allan Cargille   University of Wisconsin
        Steve Crocker   TIS
        Peter Deutsch   BUNYIP
        Peter Ford   LANL
        Jim Galvin   TIS
        Joan Gargano   UC Davis
        Arlene Getchell   ES.NET
        Rick Huber   INTERNIC - AT&T
        Christian Huitema   INRIA
        Erik Huizer   SURFNET
        Tim Howes   University of Michigan
        Steve Kent   BBN
        Steve Kille   ISODE Consortium
        Mark Kosters   INTERNIC - Network Solutions
        Paul Mockapetris   ARPA
        Paul-Andre Pays   INRIA
        Dave Piscitello   BELLCORE
        Marshall Rose   Dover Beach Consulting
        Sri Sataluri   INTERNIC - AT&T
        Mike Schwartz   University of Colorado
        David Staudt   NSF
        Einar Stefferud   NMA
        Chris Weider   MERIT
        Scott Williamson   INTERNIC - Network Solutions
        Russ Wright   LBL
        Peter Yee   NASA

10. SECURITY CONSIDERATIONS

   While there are comments in this memo about privacy and security,
   there is no serious analysis of security considerations for a white
   pages or directory service in this memo.

11. AUTHORS' ADDRESSES

     Jon Postel
     USC/Information Sciences Institute
     4676 Admiralty Way
     Marina del Rey, CA 90292

     Phone: 310-822-1511
     Fax: 310-823-6714
     EMail: Postel@ISI.EDU

     Celeste Anderson
     USC/Information Sciences Institute
     4676 Admiralty Way
     Marina del Rey, CA 90292

     Phone: 310-822-1511
     Fax: 310-823-6714
     EMail: Celeste@ISI.EDU

APPENDIX 1

   The following White Pages Functionality List was developed by Chris
   Weider and amended by participants in the current discussion of an
   Internet white pages service.

   Functionality list for a White Pages / Directory services

   Serving information on People only

   1.1 Protocol Requirements

      a) Distributability
      b) Security
      c) Searchability and easy navigation
      d) Reliability (in particular, replication)
      e) Ability to serve the information desired (in particular,
         multi-media information)
      f) Obvious benefits to encourage installation
      g) Protocol support for maintenance of data and 'knowledge'
      h) Ability to support machine use of the data
      i) Must be based on Open Standards and respond rapidly to correct
         deficiencies
      j) Serve new types of information (not initially planned) only
         only upon request
      k) Allow different operation modes

   1.2 Implementation Requirements

      a) Searchability and easy navigation
      b) An obvious and fairly painless upgrade path for organizations
      c) Obvious benefits to encourage installation
      d) Ubiquitous clients
      e) Clients that can do exhaustive search and/or cache useful
         information and use heuristics to narrow the search space in
         case of ill-formed queries
      f) Ability to support machine use of the data
      g) Stable APIs

   1.3 Sociological Requirements

      a) Shallow learning curve for novice users (both client and
         server)
      b) Public domain servers and clients to encourage experimentation
      c) Easy techniques for maintaining data, to encourage users to
         keep their data up-to-date
      d) (particularly for organizations) The ability to hide an
         organization's internal structure while making the data public.

Postel & Anderson   [Page 33]

         e) Widely recognized authorities to guarantee unique naming during
            registrations (This is specifically X.500 centric)
         f) The ability to support the privacy / legal requirements of all
            participants while still being able to achieve good coverage.
         g) Supportable infrastructure (Perhaps an identification of what
            infrastructure support requires and how that will be
            maintained)

   Although the original focus of this discussion was on White Pages,
   many participants believe that a Yellow Pages service should be built
   into a White Pages scheme.

   Functionality List for Yellow Pages service

   Yellow pages services, with data primarily on people

   2.1 Protocol Requirements

         a) all listed in 1.1
         b) Very good searching, perhaps with semantic support OR
         b2) Protocol support for easy selection of proper keywords to
            allow searching
         c) Ways to easily update and maintain the information required by
            the Yellow Pages services
         d) Ability to set up specific servers for specific applications or
            a family of applications while still working with the WP
            information bases

   2.2 Implementation Requirements

         a) All listed in 1.2
         b) Server or client support for relevance feedback

   2.3 Sociological Requirements

         a) all listed in 1.3

   Advanced directory services for resource location (not just people
   data)

   3.1 Protocol Requirements

         a) All listed in 2.1
         b) Ability to track very rapidly changing data
         c) Extremely good and rapid search techniques

3.2 Implementation Requirements

    a) All listed in 2.2
    b) Ability to integrate well with retrieval systems
    c) Speed, Speed, Speed

3.3 Sociological Requirements

    a) All listed in 1.3
    b) Protocol support for 'explain' functions: 'Why didn't this
       query work?'