Strangers in Paradise

Bruno Koechlin <Bruno.Koechlin@inria.fr>
Katy Treca <Katy.Treca@urec.fr>
Paul-André Pays <Paul-Andre.Pays@inria.fr>

Abstract

Based on the authors' experience in working with heterogeneous implementations of X.500 [1] DSAs, this paper presents our current view on this crucial operational issue: functional and organizational interworking. The paper starts with the requirements for providing a widely accepted X.500 service for the R&D community. In the light of the PARADISE existing pilot and activities today's problems are analyzed in a generic way with special emphasis on all that is related to knowledge (model, representation, operational modes, aso...). Finally we propose a few recommendations which in our view should be taken into account if the R&D community wishes to really deploy an operational X.500 directory service in the short to medium term. The study is focused on first-level DSAs (eg. national masters) in terms of functionalities of the software but also in terms of organization and somewhat in terms of associated value added services.

I. Introduction and background

In the beginning was Christian Huitema and his X.400 and ASN.1, then out of a THORN came Pizarro and we started working. Then Paradise was invented across the Channel and began to spread, and our story begins at this point, when we, strangers, wanted to enter Paradise.

At this point all we knew was that God's name was Quipu and holy book of Paradise was RFC1276 [2] while ours was simply X.500 in its original '88 version.

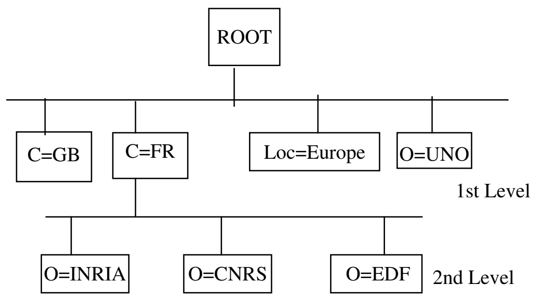
The work and problems we had with deploying OPAX (the French X500 pilot operation) and trying to install it in Paradise begat the VALUE PARADISE OIFP interworking activity.

Throughout this long story we had plenty of problems but also plenty of fun and we started to feel a little less dumb, so we would like to tell you this story and let you share a little bit in what we have discovered.

This paper thus presents the difficulties and sometimes the solutions when using a non RFC1276 compliant implementation in the Paradise White Pages Pilot. This will be presented from both theoretical (the standard and the OIFP work) and the practical (the OPAX French pilot) point of view.

II. Service requirements for a first level DSA

The ideas below are largely based on our experience with OPAX the French pilot which uses a Pizarro first level DSA and comprises three totally different implementations (Pizarro, Quipu and Marben).



A server associated with a first level naming context must provide an access path to any subordinate entry as well as a path to every first level server.

Each first level DSA must contain :

- the full root naming context which implies a reference path towards any others first level DSA,

- a path (subordinate reference) towards each of its subordinate DSAs.

The specific nature of a 1st level DSA is its naming context, it manages a simple DN with only one RDN.

A second level DSA has to contain :

- a superior reference to a first level DSA which enables a path towards any 1st level DSA

- a path towards each of its subordinate DSAs.

II.1 First level DSA services

The minimum compliant service has to provide :

- for its internal requests (from its subordinate DSAs), an access path to all the subordinates and to the other first level DSAs,

- for the external requests, an access path to all its subordinates.

It has the choice between chaining or returning a referral to answer the request. This service imposes to maintaining an accurate and up-to-date knowledge of the other 1st level DSA and of all its subordinates. Hopefully, this knowledge does not change too often.

II.2 For «an efficient operational service»

An operational service must involve more than just satisfying the minimum requirements:

- it must increase the level of service it provides to others

- it must enable others to be less dependent on the first level

Increasing the level of service:

Two keywords «response time» and «reliability» make it crucial that a first level DSA has:

- to have a maximum amount of knowledge (more than the required minimum) to avoid unnecessary chaining or referrals

- to answer the request as fast as possible and to keep a maximum of availibility, it must not deliver any other service (eg. to store other data than knowledge) and must have maximum connectivity (IP, X25...)

- to check the effective connectivity of its subordinates

- to be associated with one or more well synchronized backup systems

- to make sure the updating of the knowledge doesn't penalize the normal operational processing

- to directly or indirectly provide means to access all the subordinates taking into account the available network stacks and the potential network level filters.

Enabling others to be less dependent

The key idea is to avoid bottlenecks, which implies:

- providing means for its subordinates to connect (to bind) directly to as many other DSAs as possible without using the first level DSA. For this there must exist tools and procedures by which subordinates can retrieve and use all the first level knowledge,

- giving some help and documentation to newcomers so that their integration into the service is as easy as possible and without any negative side-effect on the existing base,

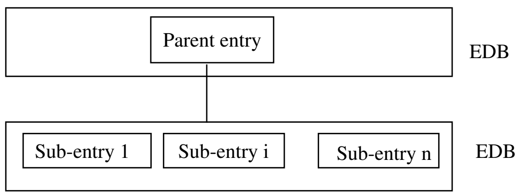
- automating as much as possible all these procedures, which implies providing and using tools for this purpose.

III. RFC 1276

RFC1276 has been written by S. Kille in order to define a replication mechanism which is not available in the X500-88 standard. This interim mechanism was intended to be simple in order to be easily implemented and deployed. To make it simple the X500 model was restricted. We will present only some key points useful to understand the rest of the presentation.

III.1 RFC 1276 Data Model

In the standard the unit of mastering data is the entry. In the RFC1276 model this unit is the Entry Data Block (EDB) which is the complete set of subordinate entries of a given entry (parent entry). An EDB must be entirely mastered by a DSA



For each subordinate entry the EDB contains :

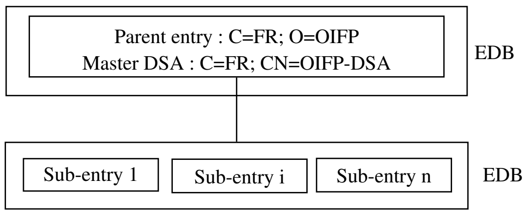
- its Relative Distinguished Name

- its Attributes

- the knowledge information associated with it

III.2 RFC1276 Knowledge Model and Representation

Knowledge information is represented in the DIT. Each entry whose siblings are in another DSA has a knowledge attribute which contains the distinguished name of the DSA which holds its siblings.



All the subordinate entries of C=FR; O=OIFP are held by DSA C=FR; CN=OIFP-DSA

An escape mechanism called «spot shadowing» allows you to have an entry of the EDB mastered by another DSA.

III.3 Proprietary replication mechanism

This mechanism allows a DSA to retrieve a complete EDB depending on the EDB version.

III.4 Consequences

Advantages

- RFC1276 defines a replication mechanism. The fact that entries are replicated reduces the chaining of operations and enhances the performance.

- Since all the subordinate entries should be in the same DSA (except if the spot shadowing mechanism is used), the Onelevel search operation is very efficient.

- A reference contains only the Distinguished Name of the DSA (not the Presentation Address) this makes the reference independent of the Presentation Address, the directory is used to retrieve the presentation address.

- This model provides a mechanism to organize the distribution of the knowledge between the first level DSAs.

Limitations

- This model imposes an internal representation of the knowledge references

- This mode of representation puts constraints on the DSA naming : the distinguished name of the DSA must be above the naming context the DSA is responsible for. This poses many problems for the use of «secure DSAs» to store certificates such as those required by PEM or X.509.

- In order to organize the distribution of the knowledge this model requires a root (level-0) DSA which has to contain all masters of the first level entries.

IV. A non RFC1276 DSA among an RFC1276 infrastructure

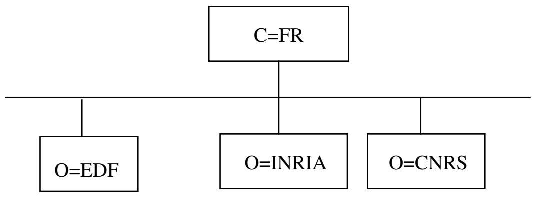
Different configurations are discussed below :

IV.1 Below a RFC1276 DSA

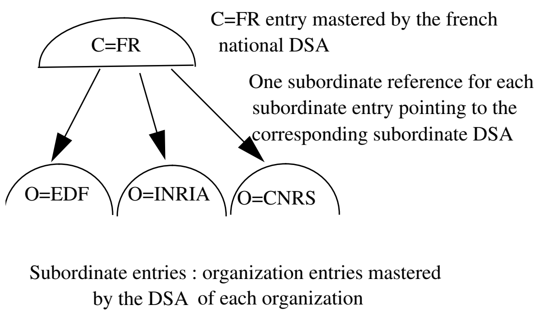
Theoretically, there should be no problems but in practice problems exist since the "spot shadowing" mechanism specified in RFC1276 has not been implemented. This mechanism is aimed at interworking with non RFC1276 implementations.

IV.2 Above an RFC1276 DSA

The subordinate entries of the C=FR entry are distributed according to the following scheme :



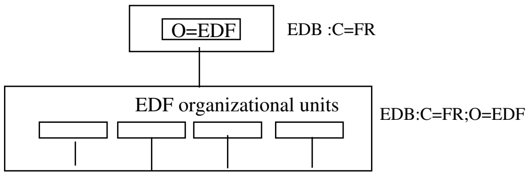
DIT



ENTRIES DISTRIBUTION

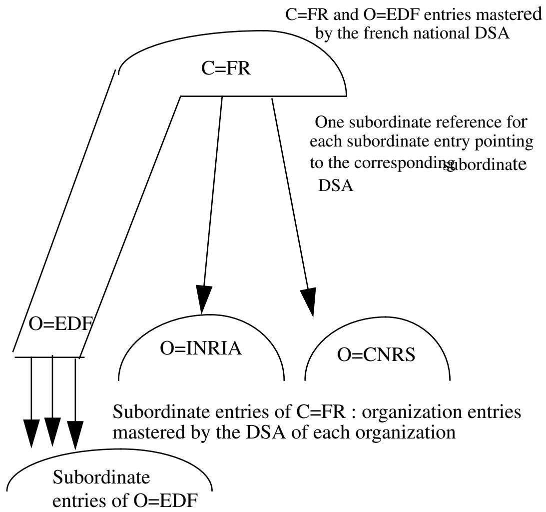
Among the subordinate DSAs we have RFC1276 DSAs one of them is for instance the EDF DSA.

If we put the entry O=EDF in the EDF RFC1276 DSA, we will have the following internal representation in the EDF DSA.



In this case, the EDF DSA will consider that C=FR has only one subordinate entry C=FR; O=EDF and will not be able to see the other subordinate entries of C=FR.

We could put the entry O=EDF in the French national DSA in which case the scheme would be the following :



ENTRIES DISTRIBUTION

In this case, subordinate references must be registered in the French national DSA for every O=EDF subordinate entry. When the EDF DSA manager wants to add or delete a subordinate entry of O=EDF, he must ask the French national DSA manager to add or delete the corresponding reference. Since EDF is a huge organization, it is difficult to manage.

X500 provides another type of reference which could solve this problem : the non-specific subordinate reference. Applied to the entry EDF, it would give the possibility to specify that subordinate entries of O=EDF can be found in the EDF DSA. This type of reference is optional in the standard and was not available in the implementation used in the French national DSA until the beginning of 1994. It is currently experimented in the framework of the OIFP project and is likely to be operational when this paper is released. In the products we have studied so far, this type of reference is seldom implemented.

IV.3 Knowledge discovery

The question is :

How a non RFC1276 DSA can retrieve and maintain knowledge and use it and make it usable by others ?

retrieve :

different models and internal representation of the knowledge are used by the different implementations thus one single possibility remains : to make use of returned referrals. The operation sequence is roughly : bind to a DSA, get a list of its entries, and read these entries but requiring «chaining prohibited», then, if a real entry is returned, it is a local entry. If not, a referral is returned back. Thus, we get the knowledge and are able to rebuild a knowledge tree. The company in charge of Pizarro has provided us with such a tool that we run periodically.

make available :

in order to avoid running this resource-consuming tool too often, OPAX DSA provides its users with a mail and FTP server where the retrieved and up-to-date knowledge information is available using a conventional external representation. Each product then requires a specific tool in order to load this common external representation from the server.

IV.4 Subordinate knowledge

The RFC1276 makes the one-level search very efficient since all the subordinate entries must be in the same DSA (except when the spot shadowing mechanism is used). One consequence is that the directory client's algorithms for browsing and searching designed in the framework of an RFC1276 infrastructure use mainly the one-level search operation.

Since all the subordinates of C=FR are distributed in separate DSAs, if a oneLevel search with baseObject=C=FR is to be performed, it must be chained to every subordinate DSA. A chained one-level search under C=FR is very slow compared to the same operation in an RFC1276 server. This is the reason why OPAXdsa is configured to return referrals whenever chaining would have to be performed to more than one single subordinate DSA. Since some other implementations are not processing returned referrals this has some interworking drawbacks.

In order to enhance the performance we suggest registering in the French national DSA copies of every subordinate entry of C=FR using an out-of-band mechanism. In this scheme the master entry of an organization under C=FR stays in the DSA of the organization while still providing the same level of one-level search performance as an RFC1276 conforming server. Previously, the software used in the French Master DSA could not hold a partial copy of a subtree. It assumed that it held the whole subtree. A new release, experimented in the framework of the OIFP project, provides this one-level copy mechanism and will be installed on the French Master DSA.

IV.5 Conformance to RFC1276

As mentioned in paragraph 4.1, RFC1276 is not fully implemented in PARADISE implementations. The "spot shadowing" mechanism is not available, consequently you cannot distribute the subordinate entries of an entry in different DSAs. So the PARADISE root DSA must hold the C=FR entry, if not the C=FR naming context would not be visible from PARADISE. The result is that we have two different C=FR master entries in the DIT, one in the root DSA, one in the C=FR DSA. It can work as long as the French DSA is not fully conformant to the standard and does not check the state of the operation received from the root DSA. The French DSA should return an "Invalid Reference" error and no operation could be chained from the root DSA to the French DSA.

From the user point of view, an organization may demand to have its master entry held in its own DSA for security or management reasons, for instance. This is not possible with RFC1276.

V. Evolution

V.1 Knowledge organization at the top of the DIT

Given that RFC1276 is only an interim solution and that it imposes too many constraints :

- Master copy of the first level entries held in the root DSA

- Demands a specific internal representation of the knowledge information

- Limitation on DSA naming

other solutions must be considered.

The standard specifies that :

- each first level DSA must be able to simulate the functions of a root DSA and then hold a copy of the root context

- the definition of the administrative procedure to replicate the root context is outside the scope of the standard

This means that each first level DSA must have at least a «reference path» for each first level entry pointing to a first level DSA.

How could this knowledge be distributed ?

We consider below two different approaches : external distribution outside X500, «root» servers using the X500 technology.

V.1.a External distribution outside X500

This approach requires :

- the definition of an external format to represent the knowledge information

- transfer protocols to distribute this information

- a coordination procedure to collect, integrate, distribute and use this information.

This could be an interim solution but we consider that X500 technologies can be used to provide this function.

V.1.b «Root» servers using the X500 technology

Root servers not fully integrated in the X500 infrastructure :

In this model we suggest setting up a set of «knowledge servers» not «fully» connected to the first level DSAs, these servers would be «specific DSAs» (ie. they would not be associated to any existing naming context but serve this knowledge sharing purpose) :

- the master knowledge information would be contained in a primary server

- secondary servers would contain a replication of this master knowledge information

- this knowledge information could be retrieved from the servers by tools using a protocol which could be DAP. An example of implementation could be :

```
LIST baseObject=root
```
to get all the first level entries
```
for each first level entry : READ object="first level entry" with chaining prohibited
```
to get the master DSA of each first level entry

This algorithm may not be considered powerful enough in the sense that it only makes it possible to retrieve the master DSA of each first level entry and not the slave DSAs which hold copies. The referral returned after a read operation can only contain one reference.

This algorithm could be enhanced in the following way :

```
LIST baseObject=root
```
to get all the first level entries
```
for each first level entry : SEARCH
baseObject="first level entry" with
chaining prohibited
```
to get the DSAs which hold this first level entry

This requires modifying the knowledge server so that :

- it sends a searchResult containing continuation references instead of a DSA referral

- in the set of continuation references returned, the first one must correspond to the master DSA for the tool to identify this master DSA, the following references correspond to the slave DSAs.

Root servers fully integrated in the X500 infrastructure

In this model servers would be DSAs talking DAP and DSP, connected to first level DSAs.

- The master knowledge information would be contained in a primary server.

- Secondary servers would contain a replication of this master knowledge information.

- These servers would only return referrals to provide high performance answers.

- The first level DSAs could retrieve the knowledge information using :

a knowledge discovery tool as presented above

a dynamic approach :

superior references to the root DSAs could be set to have referrals returned by the root DSAs. These referrals would then be cached.

replication :

this could be considered in the framework of the 92/93' standard since this version makes it possible to replicate knowledge. Each first level DSA would replicate the subordinate knowledge associated to the root entry.

V.2 Data organization at the top of the DIT

V.2.a Copies of the first level entries

The key point here is to decide whether first level DSAs should contain copies of the first level entries and in this case how this replication could be carried out. Two questions can be raised :

- Is it necessary to have DSAs holding copies of the first level entries ?

- Should these first level entries be replicated in the first level DSAs which are responsible for the distribution of the knowledge or should it be a service provided by other DSAs ?

It is necessary to have copies of the first level entries if the directory operation «on-level search baseObject=root» is considered as relevant. We can assume that :

- customers may not know the distinguished names of the first level entries (country code for instance),

- the knowledge of these first level entries will not always be integrated in their «user interface server».

We can conclude that DSAs containing copies of the first level entries are required.

Should this service be provided by the same DSAs ? The key point here is performance. Performing a search is more resource-consuming than getting knowledge references. Given the state of the pilot we can assume that for the moment it is not necessary to separate these two functions but in the future we could have :

- applications using the directory as a name server, that already know the distinguished name of the target object and require an efficient infrastructure to retrieve the attributes of the target object. These applications will not require search facilities but high speed knowledge servers (not loaded by search operations)

- applications using the directory for look up purposes like the white pages service with requires search facilities and thus requires servers containing copies of entries.

To avoid bottlenecks several copies of these servers must be available. I would be up to each administrative domain to operate servers containing copies of these different types of information.

V.2.b Replications of the first level entries

Root servers not fully integrated in the X500 infrastructure

Given that the first level DSA holds a copy of the root context, it holds the necessary knowledge information to be used by a tool using DAP to read all the first level entries. This type of replication would be partial since only the public attributes could be replicated. A full replication could only be achieve by a standardized protocol, independent of the implementation.

Root servers fully integrated in the X500 infrastructure

As long as a standardized protocol is not used, replication can only be carried out currently by :

- using a replication tool with the limitations already mentioned above
- using a proprietary replication protocol

The 92/93 X500 standard specifies a replication protocol. This protocol should allow us to achieve this function using the two following replication agreements :

1. the master root server would replicate the first level entries mastered by the first level DSA with which it has the replication agreement
2. a first level DSA would replicate all the entries except the one it masters, from a root server (primary or secondary) with which it has this replication agreement.

V.3 Operational aspects

V.3.a Operational mode of the first level DSAs regarding the chaining and referral mode

Given that first level DSAs must simulate the virtual root DSA, their main role is to provide knowledge. This can be done by replication or by returning references. Providing this service efficiently prohibits chaining since this would consume two many resources at this level of the DIT. This means that the «relay» function must be provide somewhere else.

V.3.b Relay servers

Customers may need «relay» servers for at least 3 main reasons :

1. All the different DSAs may not be fully connected to all the networks. In our community we have DSAs connected only through IP and others only through X25.
2. For access control reasons, DSAs (or the network equipment they are connected to) may accept calls coming only from a specific list of network addresses.
3. For X500 interworking reasons, DSA managers may wish to dialog only with one specific DSA and not to have to be faced with several different implementations spread throughout the world.

DSAs providing this service will have the following features :

1. a complete network connectivity
2. a rich knowledge to chain the operations efficiently
3. no data entries

Currently the X.500 standard does not alllow a DSA to behave as a relay since once the name resolution process has started it must progress in each DSA which chains the operation. Since a relay DSA does not hold data it cannot make the name resolution progress. This requires that

1. A relay-DSA must be modified so that it makes it possible that the name resolution does not progress
2. DSAs do not check in the trace part of a chained operation that the name resolution did not progress in a DSA.

VI. Conclusion

We can draw a few conclusions from our work :

- RFC1276 has demonstrated a lot of qualities but it is now time to forget it an use implementations truly conformant to the standard while preserving the efficiency of RFC1276.

- Effective wide deployment of X.500 based services will impose conformance to the '93 version of the standard. This will alleviate most of the interoperability and interworking problems we have been faced with so far, mostly because such key factors as the knowledge representation and the replication mechanism are now specified.

- Design and deployment of an operational X.500 service will still be a complex task.

- Knowledge distribution is a key issue. In our opinion the X.500 protocol should be improved so that :

  - multiple references related to the same naming context could be returned. A preference should be associated with each reference.
  - relay-DSAs could be achieved without having to "break" the standard

- A set of requirements about the "opening" of any X.500 service (comparable to the internet hosts requirements) should be established, for example :

  - no server exists without at least one back-up with a separate network access
  - no first-level server exists without at least a one-level copy of its subordinate entries
  - no distribution of a naming context exists without that same one-level replication in order to make all one-level searches extremely efficient
  - a set of requirements about acceptable and recommended behavior is established to provide a framework to clients' designer and developers in order to avoid poorly designed

clients breaking down the whole service.

We have identified other issues which are key points in order to deploy a directory service.

X.500 is an excellent technology but not a service per se, it has be complemented by a lot of additional services in order to be, for example, usable as the basis of an internet wide White Pages Service. These will certainly include :

- A complete set of specialized access servers acting as intermediaire (brokers) between the clients and the pure standard hierarchical DSA infrastructure. The access (or relay) servers will contain a lot of knowledge and most probably make use of the centroid (see whois++) concept in order to provide an indexed attribute based access to a hierarchical tree.

- An acceptable solution to the multiple providers issue for which we estimate that a combination of replication and of centroid front-ends is a far better solution than the existing NADF approach.

- New access protocols and access servers such as LDAP [3], CLDAP [4] and SOLO [5] best suited to TCP/IP network and stacks, which would be much simpler to use in order to develop plenty of clients (interactive or embedded in other applications).

- A good integration of X.500 with other existing information services is necessary. As regards the Internet, experimentations have already taken place :

  - New X.500 attributes containing URL have been defined. This makes it possible to point to any Internet resource from the directory. For instance, this mechanism is used to link the directory with multimedia databases like W3 databases.

  - A debate aimed at defining a «X.500 URL» has started. This URL would point to a X.500 resource (object, attribute) from the Internet.

Even if there is still plenty of work to be done, we are rather optimistic and confident about the future of X.500 as a basic technology for many directory demanding applications. We are confident because we have seen that X.500 has proven itself so far in pilot such as PARADISE and PASSWORD. Furthermore we feel that for any decent and general directory services the set of requirements will be as heavy as we have observed for X.500. There are good reasons to think that any competitor would have to be as complex as X.500.

On the other hand, it is clear that X.500 should be considered as an enabling technology and not as a service. Services might be designed and deployed using X.500 along with as many accompanying technologies as required. In case ISO/ITU insists on considering X.500 as a complete and closed service, the only option would be for the internet (IETF) to take over the control and responsability of the protocol and of the services.

VII. References

[1] The Directory - - - overview of concepts, models and services, December 1988. CCITT X.500 Series Recommendations.

[2] S.E. Hardcastle-Kille. Replication and Distributed Operation extensions to provide an Internet Directory using X.500. RFC1276, Department of Computer Science, University College London, November 1991.

[3] W. Yeong, T. Howes, S.E. Hardcastle-Kille. X.500 Lightweight Directory Access Protocol. RFC1487, July 1993.

[4] Alan Young. Connection-less Lightweight Directory Access Protocol. Internet Draft, draft-ietf-osids-cldap-oo.txt, October 1993.

[5] C. Huitema, P.A. Pays, A. Zahm, A. Woermann. Simple Object Look-up protocol (SOLO). Internet Draft, draft-huitema-solo-00.txt, INRIA, TS-E3X, December 1993.

VIII. Acknowledgements

This work has been made possible by

- All the people who have cooperate in the framework of PARADISE and OPAX

- The European Commission which funds the work we are doing in the framework of the VALUE X500 project (Operational Interworking Platform and Forum).

Author Information

Bruno Koechlin is engineer at INRIA (The French Institute in Computer-Science) where he takes part in national and international projects related to network applications (Email, Directory). He is mainly involved in the VALUE X500 OIFP project which deals with interworking between X.500 implementations.

Katy Treca is engineer at CNRS (The French Research Institute in Scientific Research) where she works in the Network Unit. She has taken part in the creation and organization of the French Research Network (RENATER). She is responsible with Jacqueline Denyset for the French Master DSA.

Paul-André Pays is leading the INRIA CSR/ Comiso team, whose task is to provide the french and european R&D community with value added network services and to experiment new services and technologies. The domains of activities of the team include Mail and Messaging, Directory services and distributed information service. Paul-André Pays is convenor of the RARE WG - NAP working group and heading OPAX the french X.500 pilot.