DESIRE II: Project Deliverable

Project Number:	RE 4004 (RE)
Project Title:	DESIRE II - Development of a European Service for Information on Research and Education II
Deliverable Type:	Internal

Deliverable Number:	D3.3c2
Title of Deliverable:	LDAP index exchange requirements and methods
Workpackage(s) contributing to the Deliverable:	WP 3.3
Author:	Peter Gietz
Contact Details:	DANTE, Francis House, 112 Hills Road, Cambridge, CB2 1PQ, UK Peter.Gietz@dante.org.uk
Other Authors:	Peter Valkenburg, Henny Bekker
URL	http://www.desire.org/html/research/deliverables/D?_?/D?_?.html

Abstract	This document describes the overall concept of a distributed indexing system based on the Common Indexing Protocol (CIP), as it will be implemented in the DESIRE II project and afterwards maintained as a service by the European academia. Although the system is designed for multi purpose usage, the main focus of this document lies on its application as a directory white pages index. The main subjects are previous work, introduction to CIP, the over all concept, distribution of the index objects, security and privacy issues and crawler policy.
Keywords	Indexing, CIP, LDAP, X.500, TIO, security, PGP, privacy

TABLE OF CONTENTS

PART II

DOCUMENT CONTROL

Issue Number	Issue Date	Reason for Change
V.0.8	5.3.1999	Initial version, with inclusion of parts of two documents dated 4.8.98 and 20.10.98
V.1	8.6.99	First complete version, added CIP introduction and security chapter

EXECUTIVE SUMMARY

The document begins with a summary of previous and parallel work on Directory indexing and an introduction to the Common Indexing Protocol (CIP). The main aim of the paper is to define a directory indexing system that is deployable in a European context, making directory information available to the research community of all participating countries. Although aimed at Directory indexing the model should be applicable to other indexing problems.

To implement an indexing system on such a large scale, a hierarchical index creation and distribution is necessary for overall performance and scalability issues. The whole indexing system of gathering and distributing and searching index objects should be managed by the server side. The mechanisms proposed in this document can be described as a subset of the Common Index Protocol (CIP), which is seen as the future standard for indexing in the Internet. Though not all features defined in CIP are planed to be implemented the overall structure should be compatible with this standard. After detailed remarks about the index object format, and the process of gathering and distributing index objects as well as the query routing, the over all concept of the index system is defined. Remarks on security and privacy issues, which favour a solution based on PGP conclude this document.

SCOPE STATEMENT

This document is accompanied by a more technical document “Generic distributed indexing architecture using CIP/TIO & HTTP” about the server side of the indexing system, as well as two documents on “LDAP Client requirements” and on “LDAP Client architecture” for the client side. Parts of this document have been released earlier.

PART III

GLOSSARY

ACL
Access control list

CIP
Common Indexing Protocol

DSA
Directory System Agent, Directory server

DISP
Directory Information Shadowing Protocol, replication mechanism in X.500

DSI
Data Set Identifier

DSP
Directory System Protocol

FTP
File Transfer Protocol

HTTP
Hyper Text Transport Protocol

LDAP
Lightweight Directory Access Protocol

MIME
Multipurpose Internet Mail Extensions

PGP
Pretty Good Privacy

SLP
Service Location Protocol

S/MIME
Secure MIME

SMTP
Simple Mail Transfer Protocol

SSL
Secure Socket Layer

TIO
Tagged Index Object

URI
Uniform Resource Identifier

X.500
International Directory standard defined by ISO and ITU

PART IV

1 INTRODUCTION

A Development that already has taken place in the World Wide Web has started in the Directory world as well: Its success, i.e. the huge amount of Data that have accumulated, creates the need of new technologies for managing and retrieving the information. In the WWW the standard technology, i.e. hyperlink collections, were complemented by WWW search services, a meta data aware second generation of which is the current state of affairs.

1.1 Historical background

In Directory the standard means of making the data better accessible to the end user was replication. Every part of the Directory Information Tree, a logical structure that contains the data, is mastered by one Directory System Agent (DSA), a directory server. The actual information is stored in attributes. These attributes belong to entries, which are defined by the special attribute named objectClass (e.g. objectClass Person) and which represent a node in the DIT. Because of the interconnection of all DSAs, to the end user the Directory looks like one single entity. He only has to connect to any of the DSAs and can retrieve all data from all other DSAs as well, since these can interchange data via a dedicated DSA-DSA protocol called Directory System Protocol (DSP). Since this communication can happen over long distances (e.g. if a user from Europe, and thus connecting to a European DSA, wants to have Directory information from New Zealand), data retrieval can be very slow. The solution for this problem is replication or shadowing: Via a dedicated protocol, the Directory Information Shadowing Protocol (DISP) DSAs can exchange the whole data subtree of the DIT, which they master. (E.g., A DSA in UK could shadow the Data of a DSA in New Zealand). This redundant of information of course has more advantages than just accelerating user access; it also has its merits in terms of general availability in the case of failures of single DSAs. One problem that replication technology couldn’t solve though was searches all over the world. Since in theory all DSAs would have to be queried only "mega DSA’s", shadowing all other DSAs would accelerate such searches. But such a DSA would transcend current available software and hardware as well as counteract the distributedness of the Directory.

1.2 Previous work

In the White Pages IETF Meeting in Nov 1993, as documented in [RFC1588], the need for Directory indexing technologies was seen for the first time. Paul Barker took up this challenge for X.500, with a concept of replicating only selected attributes (e.g. email address) of selected entries (selected by the object class) to build so called Index DSAs [Barker]. Although this reduces the problems of the before mentioned "mega DSA" this approach still would not scale. Another approach had been developed in the whois++ world [RFC1913], being to reduce the information, by tokenisation into the so called centroids, which were managed in a hierarchical structured mesh of whois++ index server [RFC1914]. David Chadwick took up this idea and defined a similar approach for the X.500 world, defining Centroid-DSAs holding information about normal DSAs [Chadwick]. Both update approaches based on the classical centroid model have as drawback that the information about interconnection between several attribute values get lost through the tokenisation. The final Approach for Directory indexing was made in the frame of the IETF WG Find, and is called Common Indexing Protocol (CIP) which defines an overall index server model [CIP arch], MIME objects for index object exchange [CIP mime], a transport protocols [CIP trans] as well as an index object type dedicated for directory information [CIP TIO].

1.3 DESIRE I

Since the indexing work in DESIRE I project was exclusively web oriented, the problem of Directory indexing was not addressed. Nevertheless some of the work in DESIRE I [Lundberg] has impact on the current approach in two ways. First the general concept of DESIRE I - crawl data, index data, retrieve data - is applicable to directory indexing as well. The second impact of DESIRE I (and of the ongoing web indexing work performed in DESIRE II) is the idea to define a general index tool box that is capable of dealing with web information as well as with Directory data. Another area of common interest is the definition of metadata. On the side of the Directory part it takes place on the level of object class and attribute definitions.

1.4 Summary of CIP

CIP is the attempt of the IETF find WG to define a common indexing mechanism for a multitude of Internet data protocols. It defines the process of passing indexing information from server to server to make them capable of routing client queries to the servers that have the apropriate data. The only prerequisite for a data access protocol to benefit from CIP is to have some kind of referral mechanism, like, e.g. X.500, LDAP and Whois++.

1.4.1 Index objects

To facilitate application domain specific indices, CIP index objects are abstract and have to be defined according to the specific usage. The different requirements for index objects are handled by encapsulating the index objects within MIME wrappers [MIME] as a standardized way to specify the differences. The respective MIME type tree is application/index. The basic protocols for moving index objects are common to all objects. In general an index object contains a subset of data in a compressed form (through tokenization).

1.4.2 Tagged Index object (TIO)

A separate CIP document [CIP-TIO] defines the TIO, which is an enhancement of the Whois++ Centroid concept. The added tags are attribute identifiers that allow to reconstruct the whole entry from the attributes contained in the index object, i.e. to „tie the individual object attributes back to an object as a whole“. The data necessary for an exchange of index objects are defined as well. A TIO consists of three parts: a header, the schema specification and the actual index payload. The schema part includes the definition of the type of tokenization.

1.4.3 Data set identifier (DSI) and Base-URI

The DSI uniquely identifies a given dataset and is chosen from any part of the ISO/ITU OID space and thus consists of an unbounded sequence of unbounded integers.

Wrapped CIP index objects include a base-URI, which is needed for referral generation based on searches in the index object. The base-URI will include information about the server from which the index object was created. Base-URI and DSI are both not part of the TIO, but are given as parameters to the MIME types and thus exist in the MIME transport wrapper.

1.4.4 CIP transport protocols

As CIP relies on interchange of standard MIME messages for all requests and replies, the CIP messages are passed over a bidirectional transport system. CIP defines three modes for the transport of the index objects: 1. The stream transport, where CIP messages are transmitted over bi-directional TCP connections via a simple text protocol; 2. Internet mail infrastructure as transport, using the Reply-To header to establish the return path; 3. HTTP transport, where transaction is performed by using the POST method to send requests and responses.

1.4.5 Security options

CIP defines several possibilities to secure the transport of tagged index objects: PGP/MIME [RFC2015], S/MIME [S/MIME], SSLv3 [RFC2246]. They should be chosen in accordance with the transport mechanism. The first two security mechanisms effect the application level and thus can be used independent of the transport protocol.

1.4.6 Problems of CIP

There are a few features of CIP and the TIO that might result in implementation problems:

- The attributes that identify the data server (DSI and base-URI) are not part of the index object but part of the MIME wrapper. This makes index object aggregation very problematic, since the DSI would be needed for identifying the original servers. So even if a an index object was aggregated

the unaggregatet index objects are needed to get the apropriate data to construct the referral. This chain of indirect referrals could increase the search time.
- The DN as part of the index object is handy for incremental index object updates, as it functions as unique identifier. On the other hand its untokenized inclusion causes a significant increase of the size of the index objects.
- Handling differently tokenized data is also problematic, since search results depend on it. Too much tokenization results in negative referrals, too litle results in increase of index object size. For a discussion see [Panotzki].

1.4.7 Deployment of CIP

CIP is to be released as RFCs very soon. There is already a CIP based index system from the Swedish TISDAG project, currently in a pilot phase that uses CIP mechanisms, like the TIO [TISDAG]. The concept differs from the current project in at least two points: 1. The TISDAG Referral server lies in a multiprotocol environment, where protocol conversion allows a client of protocol x to retrieve data from a server of protocol y. 2. The scope for the data is confined to directory data whereas the DESIRE system is designed to facilitate other types of data as well. Besides these differences, there are a lot of common aims with the DESIRE II indexing system.

2 THE MAIN CONCEPT

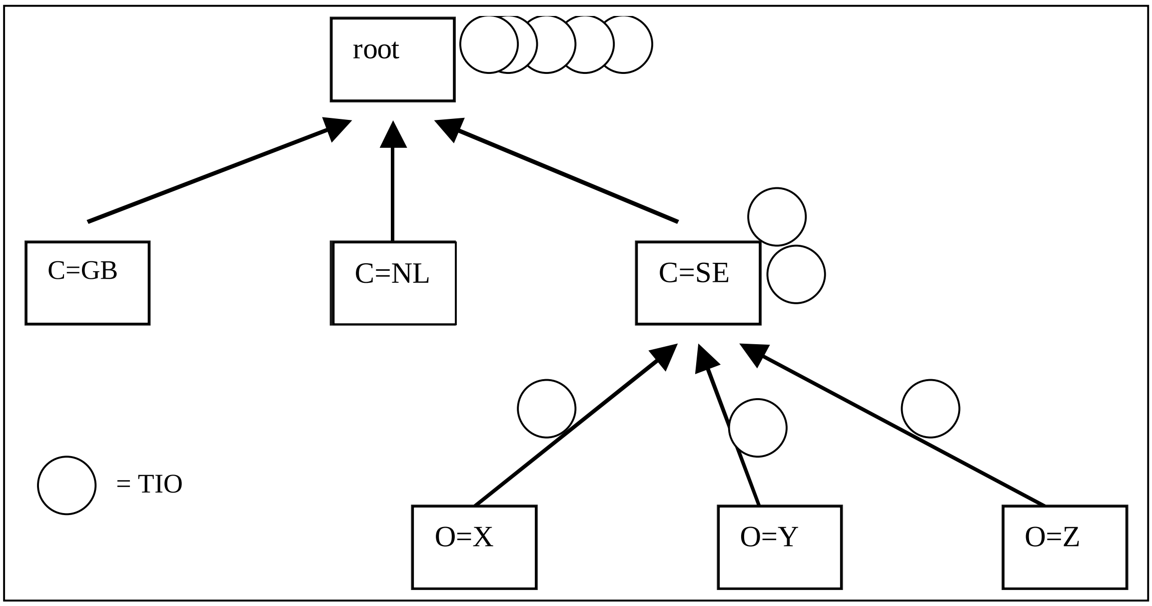
The main aim of this work package is to define a directory indexing system that is deployable in a European context, making directory information available to the research community of all participating countries. Although aimed at Directory indexing the model should be applicable to other indexing problems.

To implement an indexing system on such a large scale, a hierarchical index creation and distribution is necessary for overall performance and scalability issues. In such a model index servers located at higher levels of the hierarchy gather the index objects of server located on lower levels of the hierarchy. For example the index server of an organisation collects the index objects of all departmental directory servers, the index server of a country collects all index objects of the organisational index server. This ends up in one root index server that includes the index objects of all country level index servers that are part of the indexing system. Since it is not advisable to have one single point of information retrieval to which all clients that want to retrieve index information would have to connect to, the collection of index objects has to be redistributed downwards the same hierarchy. Since the management of such a big collection of index objects requires a considerable amount of hardware power they will not be distributed down to the single server, but might only reach to country level.

The whole indexing system of gathering and distributing and searching index objects should be managed by the server side. Clients should not need to have special features for retrieving the index information, which means that an index server has to respond to a client the same way a normal server would do, in case it doesn’t have the requested data: It just gives back a referral to a server that might have them. In the case of an index server the probability that the referral points to a server that has the data is very high. That is the only difference for the client. Although every client capable of chasing referrals could be used in the proposed indexing system, a client that includes special index related features is favourable for this project out of several reasons. First of all by now there is no publicly available LDAPv3 client that could be used for a proof of concept. Secondly special problems of index query, like the possibility of a huge amount of referrals have to be dealt with. And last but not least an index aware client can provide a better user interface that gives index specific information. Out of these reasons a dedicated client is in development, which is described in two further documents, one on specific requirements [Client-Req] and one on the client architecture, taking these into account [Client-Arch].

The mechanisms proposed in this document can be described as a subset of the Common Index Protocol (CIP), which is seen as the future standard for indexing in the Internet. Though not all features defined in CIP are planed to be implemented the overall structure should be compatible with this standard.

2.1 Gathering of Index objects



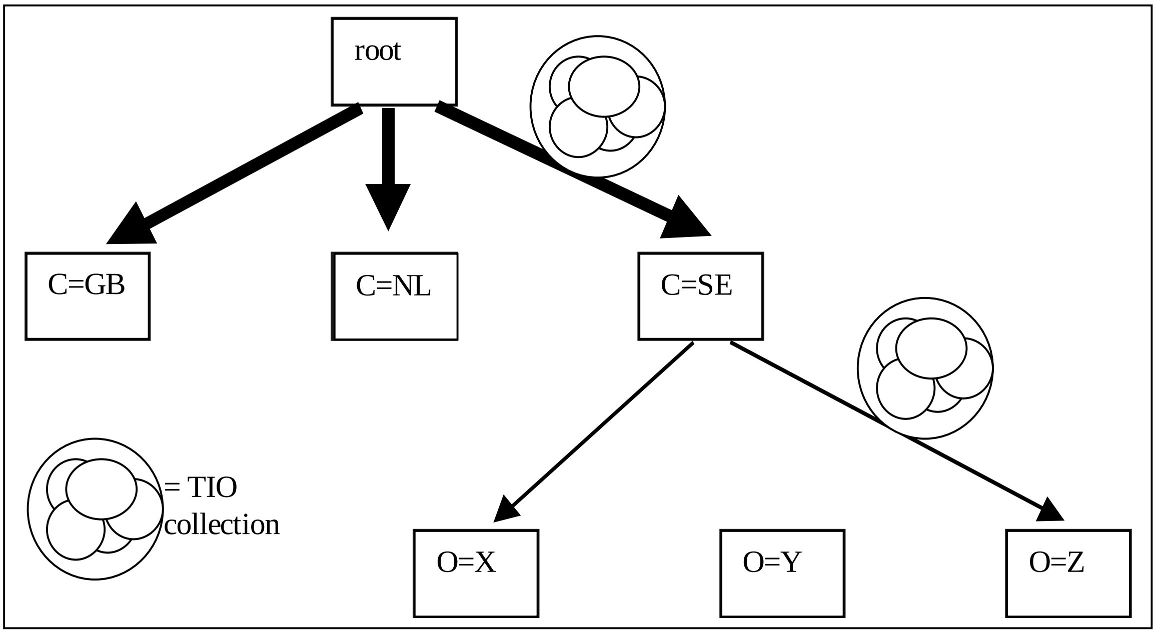
The atomic entities of the indexing system in its first stage are the index objects of the single server that are included in the indexing system. These index objects will not be modified in their content while their transport up and down the hierarchy; they will not be aggregated to bigger index objects. Although such an aggregation is defined in CIP, it produces in combination with the TIO hardly manageable problems. Through aggregation the tags of the TIO would change, which makes the retrieval more difficult. Since the index object includes header information about the data server (DSI, base URI), retrieval would have to follow back the steps of aggregation to finally reach the LDAP server. The update of index objects again would be difficult in terms of retrieving the right index entries in the right index objects, where again the whole aggregation path has to be followed. If, as proposed here, the index objects are not changed, the case of an update is quite straightforward: a new index object is to be produced and the old index object just has to be replaced in the index object collections. The DSI provides a perfect means for the identification of the index object to be replaced. Incremental update of single index objects is included in theTIO definition, which allows you to specify data blocks for add, delete and update operations. To unambigously identify the record for the delete and update oprations a unique identifier of the entry must be included in the index object. In the case of LDAP directories this identifier would be the whole untokenized DN. In a first approach the DESIRE II index system will not use this feature of incremental updates.

The index objects can be built by dedicated crawlers that crawl through the DIT subtree of one server to collect the data. A TIO converter can then in a second step produce the index object from those data. The decision which of the entries to crawl and which attribute values to collect, has to be done by each participating organisation, the maintainer of the single server respectively. These definitions should be made via crawler access policies stored in the directory itself and understood by the crawler. A separate document will define the mechanisms and the storage model for such a crawler access policy, similar to the robots.txt and robot meta tags defined for the WWW [Crawler]. To make sure that only crawlers compliant to this policy mechanism are able to get the data, the crawler has to authenticate itself. In a first stage, the crawler could be directed via access control mechanisms inherent in the Directory (acls). With such a mechanism in place it becomes irrelevant in terms of privacy issues, who will maintain and run such a crawler. It could either be the organisation itself, the NRN for all or a subset of organisations in a country, or even the maintainer of the central index objects at the root of the system.

The single servers that are part of the index system will be registered. Registered server will be put in a list, which will be accessed by the crawler or the maintainer of the crawler to retrieve knowledge about host and port of the server. Automatic server location as defined by the Service Location Protocol [SLP] will not be used, although such a technology could well be a future replacement of the registration process. The details of the registration process will be defined in a separate document [Registration].

The format of the index should be the TIO. The advantage of the TIO is, that all the indexed attributes of one directory entry can be identified, and search filters including more than one attribute can be used. The Dataset Identifier (DSI) should be used to uniquely identify a given dataset among all datasets indexed. All index objects should contain a base-URI in its header, which is crucial for generating referrals to the complete data of an indexed entry.

2.2 Distribution



To prevent a single index entry point, where all the worlds' clients would connect to, the gathered index objects (TIO collections) have to be distributed downwards again. Every country level should provide index servers for the respective country index, as well as for the mega index. If appropriate, the country level indices could be distributed to several index servers at different locations in the respective country.

The downward distribution of the indices, as well as the upward sending of the indices to be aggregated can be performed via simple FTP transfer for a proof of concept. More advanced transport mechanisms defined in the CIP Transport Protocols draft can be used instead eventually.

2.3 Query routing

The clients should not have to provide special features for using the index system. It connects to an index server in the same way it would connect to any other directory server. The access protocol is plain LDAP (v3). The server should then perform the following algorithm:

Perform a search in the locally stored dataset, and return the data if found. If no data matched the search filter, the server should consult the index server to search for appropriate entries and return the referrals to the entries, based on the base-URI found in the index.

The user could influence this algorithm by adding a base DN which defines the entry point and limits the search. The user can herewith, e.g. start the search from the root level, or from any other level in the hierarchy. In any case the client does not have to know anything about the indexing system except the hostname and port number of one nearby server, which is a part of the index system.

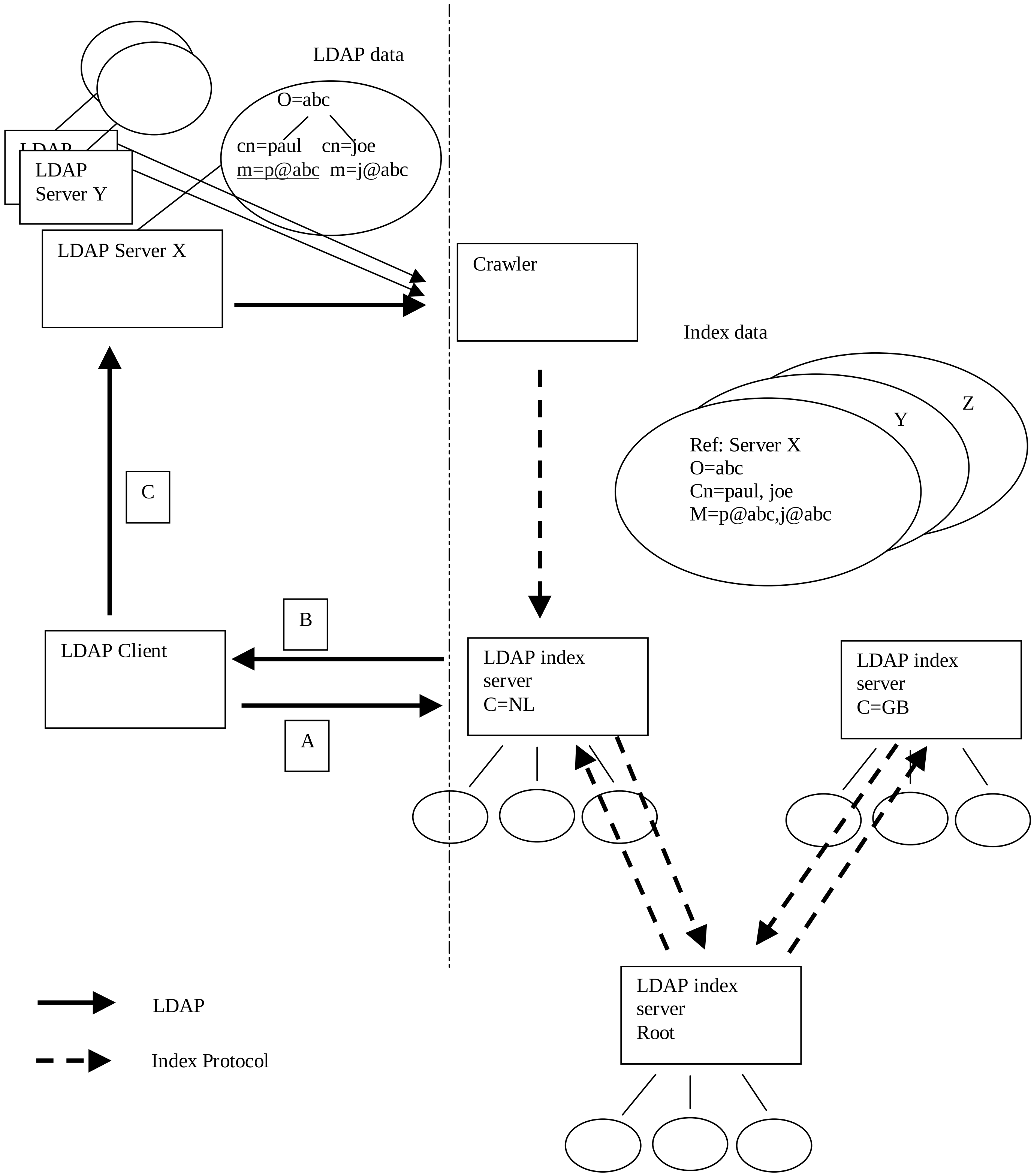
For more information about the client side see [client-req] and [client-arch].

2.4 The over all concept

For an illustration of the following see the Fig.

- A crawler collects the to be indexed data from standard organisational LDAP servers using LDAP searches.
- The crawler (or a program started by the crawler) builds Tagged Index Objects of these servers, which have to include knowledge for referrals (e.g., Base-URI) in the MIME wrapper.
- The crawler or TIO converter passes them on to a country level LDAP index server, using one of the CIP defined transport protocols (e.g., HTTP).
- The LDAP index server stores the index objects.
- The country level LDAP index servers distribute their index objects to a root LDAP index server.
- The root LDAP index server distributes the index objects of all other countries back to the country level index servers.
- A LDAP client (dedicated client, web browser, mail agent, etc.) sends an LDAP search to the country level LDAP index servers [A].
- The country level LDAP index server builds LDAP referral(s) from the index objects, which include data matching the search.
- The country level LDAP index server gives back the referral(s) to the Client [B].
- The Client interprets the referral(s) and retrieves the data from the original LDAP server [C].

The left side of the figure describes standard LDAPv3 products. The development of a dedicated LDAP client is only needed to make sure a proof of concept, including the handling of multiple referrals. The right side describes the special featured items our work package additionally has to produce namely the v3 version of the crawler, the country level LDAP index server and the root level LDAP index server. The main special feature of these will be index object creation and distribution, as well as creation of referrals from the index objects. Fo a more detailed description of this architecture see [Index-Arch].


LDAP data
O=abc
cn=paul
cn=joe
m=p@abc
m=j@abc
LDAP
Server Y
LDAP Server X
Crawler
Index data
Z
Y
Ref: Server X
O=abc
Cn=paul, joe
M=p@abc,j@abc
C
B
A
LDAP Client
LDAP index
server
C=NL
LDAP index
server
C=GB
LDAP index
server
Root
LDAP
Index Protocol

3 SECURITY CONSIDERATIONS

3.1 Personal data and privacy legislation

Since white pages directories contain personal data (i.e. e.g. name, email address, telephone number), it is important to conform to European privacy legislation. Even if all the data are public data and published in the directory with the consent of the affected persons, it is against that legislation to make available a bulk of such data. While transfered from one server to the other the index objects are vulnarable to get stolen by commercial data brokers and spammers. It is therefore neccessary to protect the index object data while transfering them on the net.

3.2 Encryption of the index objects

To secure the index object distribution process the data should be encrypted. Since CIP data are MIME encoded a MIME compatible encryption method is preferable, because then the security feature is independend of the transport protocol, let it be HTTP or FTP or email. The CIP authors advise to use PGP encrypted S/MIME as defined in [RFC2015]. PGP has a variety of advantages.

- It is commonly used in the Internet.
- It is easy to include into a MIME application..
- It provides means for public key asymmetrical encryption
- It provides means for symmetrical encrytion as well.
- In addition it provides a means of signing the data in a way that even one missing byte in the data makes the signature invalid
- All PGP functionality can be activated by a programm without human interference
- If implemented with care the passphrase that has to be inputed to the PGP programm can be securely stored and used without the possibility of snooping from outside.

3.3 Authentication between servers

All servers included in the indexing system are known due to a registration process. The maintainer of the data servers can define which data are to be included into the index. The index servers and the crawlers that take part in the index object gathering and distribution are also known. To prevent wrong index objects to be included into the index server, index object supplying programms should authenticate themselves. Servers could provide special applications entries with passwords to bind to before sending the data. A better method of authentication would be the signing of the data via a digital signature. This again could be implemented with a public key infrastructure like PGP.

4 REFERENCES

[Barker] Barker, P., "X.500 Index DSAs", Dante in Print 13, June 1995.

[Chadwick] Chadwick, D., "IndeX.500", Dante in Print 19, May 1996.

[Crawler] Gietz, P., "LDAP crawler policy information model", (work in progress)

[CIP-ARCH] Allen, J., Mealling, M., "The Architecture of the Common Indexing Protocol (CIP)", draft-ietf-find-cip-arch-02.txt (work in progress), November 1998.

[CIP-MIME] Allen, J., Mealling, M., "MIME Object Definitions for the Common Indexing Protocol (CIP)", draft-ietf-find-cip-mime-03.txt (work in progress), November 1998.

[CIP-TIO] Hedberg, R., Greenblatt, B., Moats, R. and M. Wahl, "A Tagged Index Object for use in the Common Indexing Protocol", draft-ietf-find-cip-tagged-07.txt (work in progress), March 1998.

[CIP-TRANS] Allen, J., Leach, P. J. "CIP Transport Protocols", draft-ietf-find-cip-trans-01.txt (work in progress), April 1999

[Client-Req] Mahl, D., "LDAP Client Requirements", (work in progress)

[Client-Arch] Mahl, D., "LDAP Client Architecture", (work in progress)

[Index-Arch] Bekker, H., "Generic distributed indexing architecture using CIP/TIO & HTTP", DESIRE internal project deliverable D3.3.c, no date.

[Lundberg] Lundberg, S., "Specification for Indexing Approach. The European Web Index: An Internet Search Service for the European Higher Education, Research and Development Communities", Deliverable D3.1, no date.

[Panotzki] Panotzki, P., „Complexity of the Common Indexing Protocol. Predicting Search Times in Index Server Meshes", (http://www.bunyip.com/research/papers/cip/cip.html), September 1996.

[Registration] N.N., "Registration process for the participation of LDAP servers in the DESIRE II LDAP index pilot", (work in progress)

[RFC1588] Postel, J, Anderson, C., "White Pages Meeting Report", RFC 1588, February 1994.

[RFC1913] Weider, C., Fullton, J and S. Spero, "Architecture of the Whois++ Index Service", RFC 1913, February 1996.

[RFC1914] Faltstrom, P., Schoultz, R., and C. Weider, "How to Interact with a Whois++ Mesh", RFC 1914, February 1996.

[RFC2015] Elkins, M., "MIME Security with Pretty Good Privacy (PGP)", RFC 2015, October 1996.

[RFC2165] Veizades, J., Guttman, E., Perkins, C., and S. Kaplan, "Service Location Protocol", RFC 2165, June 1997.

[RFC2246] Dierks, T., Allen, C., "The TLS Protocol Version 1.0", RFC 2246, January 1999.

[S/MIME] Ramsdell, B., "S/MIME Version 3 Message Specification", draft-ietf-smime-msg-08.txt (work in progress), April 1999.

[TISDAG] Daigle, Leslie L., “TISDAG: Technical Infrastructure for Swedish Directory Access Gateways, Technical Report“, Draft 4.1, December 1997.