Introduction to LDAP

Identity Management Workshop

Victoriano Giralt

Central Computing Facility
University of Málaga, Spain

Chişinău, Replublic of Moldova
May, 15th 2007

Let's set things straight from the beginig

LDAP is ***NOT*** a directory

Let's set things straight from the beginig

LDAP is ***NOT*** a directory

it is a protocol
to access the directory

Let's set things straight from the beginig

LDAP is ***NOT*** a directory

it is a protocol
to access the directory

X.500 *IS* the directory

Let's set things straight from the beginig

LDAP is ***NOT*** a directory

it is a protocol
to access the directory

X.500 *IS* the directory

Don't pannic

Let's set things straight from the beginig

LDAP is ***NOT*** a directory

it is a protocol
to access the directory

X.500 *IS* the directory

Don't pannic, I'm not going to resurrect the OSI stack

Let's set things straight from the beginig

LDAP is ***NOT*** a directory

it is a protocol
to access the directory

X.500 *IS* the directory

Don't pannic, I'm not going to resurrect the OSI stack

Let's make LDAP = LDAP accesable directory system

Overview

Overview

Overview

Overview

Overview

What is a directory?
from different points of view

Linguistics

According to D.R.A.E.

Directory

5. m. Roster of people belonging to a group, with indication of diverse information about them, such as role, location data, phone numbers, etc.

What is a directory?
from different points of view

Computer Science

Directory

Information objects organized hierarchycally.
Like:

- Storage file systems
- Domain Name System (DNS)
- X.500, *the* Directory

What is an *Entry*

An entry is *simply* an individual object stored in the directory

What is an *Entry*

An entry is *simply* an individual object stored in the directory
Examples of entries are

What is an *Entry*

An entry is *simply* an individual object stored in the directory
Examples of entries are

- A person

What is an *Entry*

An entry is *simply* an individual object stored in the directory
Examples of entries are

- A person
- An organization

What is an *Entry*

An entry is *simply* an individual object stored in the directory
Examples of entries are

- A person
- An organization
- A department

What is an *Entry*

An entry is *simply* an individual object stored in the directory
Examples of entries are

- A person
- An organization
- A department
- A network node

What is an *Entry*

An entry is *simply* an individual object stored in the directory
Examples of entries are

- A person
- An organization
- A department
- A network node
- The description of a classification code

What is an *Attribute*

An attribute is a name value pair of a characteristic of an object stored in a directory entry.

What is an *Attribute*

An attribute is a name value pair of a characteristic of an object stored in a directory entry.
Attributes can be

What is an *Attribute*

An attribute is a name value pair of a characteristic of an object stored in a directory entry.
Attributes can be

- Univalued. Only one same name value pair per entry.

What is an *Attribute*

An attribute is a name value pair of a characteristic of an object stored in a directory entry.
Attributes can be

- Univalued. Only one same name value pair per entry.
- Multivalued. Any number of same name value pairs.

What is an *Attribute*

An attribute is a name value pair of a characteristic of an object stored in a directory entry.
Attributes can be

- Univalued. Only one same name value pair per entry.
- Multivalued. Any number of same name value pairs.
- Operational. Managed by the server.

What is an *Attribute*

An attribute is a name value pair of a characteristic of an object stored in a directory entry.
Attributes can be

- Univalued. Only one same name value pair per entry.
- Multivalued. Any number of same name value pairs.
- Operational. Managed by the server.

Examples of attributes are

What is an *Attribute*

An attribute is a name value pair of a characteristic of an object stored in a directory entry.
Attributes can be

- Univalued. Only one same name value pair per entry.
- Multivalued. Any number of same name value pairs.
- Operational. Managed by the server.

Examples of attributes are

- A person's Surname (family name)
- A system's IP address
- A telephone number
- createTimestamp

What is an *ObjectClass*

An objectClass is a logical grouping of attributes that entries may have

What is an *ObjectClass*

An objectClass is a logical grouping of attributes that entries may have
According to objectClass definition, attributes are

What is an *ObjectClass*

An objectClass is a logical grouping of attributes that entries may have
According to objectClass definition, attributes are

- Required. The atribute must exist in an entry of that kind.

What is an *ObjectClass*

An objectClass is a logical grouping of attributes that entries may have
According to objectClass definition, attributes are

- Required. The atribute must exist in an entry of that kind.
- Optional. The atribute may (or *may not*) exist.

What is an *ObjectClass*

An objectClass is a logical grouping of attributes that entries may have
According to objectClass definition, attributes are

- Required. The atribute must exist in an entry of that kind.
- Optional. The atribute may (or *may not*) exist.

Examples of objectClasses are

What is an *ObjectClass*

An objectClass is a logical grouping of attributes that entries may have
According to objectClass definition, attributes are

- Required. The atribute must exist in an entry of that kind.
- Optional. The atribute may (or *may not*) exist.

Examples of objectClasses are

- Person
- innetNode
- schacLinkageIdentifiers

The Directory Information Tree

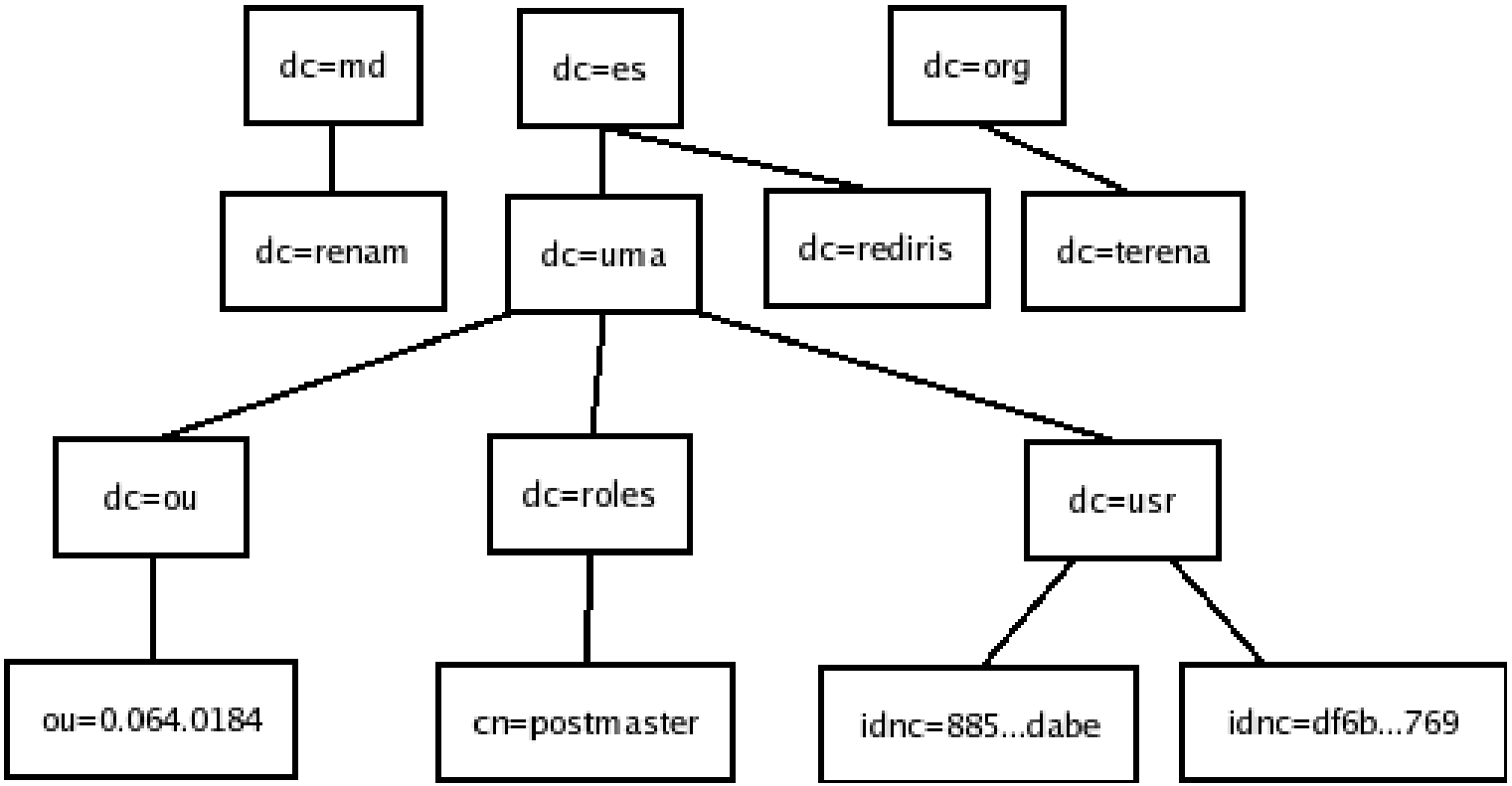
a.k.a. DIT

The tree that organises a hyerarchy of directory objects

The Directory Information Tree

a.k.a. DIT

The tree that organises a hyerarchy of directory objects



What is a *DistinguishedName*

It's my very own personal name in the directory

The components of a Distinguished Name

What is a *DistinguishedName*

It's my very own personal name in the directory

The components of a Distinguished Name

- DN

Distinguished Name

It is is an special attribute that uniquelly identifies an entry in a DIT branch. This value is composed of one or more attributes from the entry itself, and the DN of the branch where it resides. For example: dn=cn=postmaster,ou=roles,dc=renam,dc=md

What is a *DistinguishedName*

It's my very own personal name in the directory

The components of a Distinguished Name

- DN
- RDN

Relative Distinguished Name

It is the part of the DN that singles out an entry inside its branch in the DIT.
I.e.: The entry's DN minus the DN of its parent.
In the previous example: cn=postmaster

What is a *DistinguishedName*

It's my very own personal name in the directory

The components of a Distinguished Name

- DN
- RDN
- Base DN

Base DN

It is the DN of the DIT root.

For example: dc=renam,dc=md

Singularities of directories

a model closer to the real world

Best known database systems are relational ones
we will describe LDAP directories in comparison to them

Singularities of directories
a model closer to the real world

Best known database systems are relational ones
we will describe LDAP directories in comparison to them

Schema

Relational DBMS
No standard
table schema

LDAP directory
International standards
for persons and
organizations,
more so in Academia

Singularities of directories
a model closer to the real world

Best known database systems are relational ones
we will describe LDAP directories in comparison to them

Schema
Organization

Relational DBMS

One logical entity stored in several tables

LDAP directory

One logical entity
One object
One node = entry in DIT


Singularities of directories
a model closer to the real world

Best known database systems are relational ones
we will describe LDAP directories in comparison to them

Relational DBMS		LDAP directory
New table or several fields	Schema	All stored in one attribute
	Organization	As many as needed
	Multi value	

Singularities of directories
a model closer to the real world

Best known database systems are relational ones
we will describe LDAP directories in comparison to them

Relational DBMS		LDAP directory
Restricted set	Schema	Unlimited number through the use of syntaxes
	Organization	
	Multi value	
	Datatypes	

Singularities of directories
a model closer to the real world

Best known database systems are relational ones
we will describe LDAP directories in comparison to them

Relational DBMS

Matching rules outside the data model.
Implemented in programs

Schema
Organization
Multi value
Datatypes
Matching

LDAP directory

Matching rules in the data model.
Defined with the schema

Singularities of directories

a model closer to the real world

Best known database systems are relational ones
we will describe LDAP directories in comparison to them

Relational DBMS

Changes in schema
require big efforts.
Affect program logic

Schema
Organization
Multi value
Datatypes
Matching
Flexibility

LDAP directory

Granular
schema modifications
Up to the entry level.
Need attribute $->$
add objectClass

Singularities of directories
a model closer to the real world

Best known database systems are relational ones
we will describe LDAP directories in comparison to them

Relational DBMS

Network access
not standardised

Schema
Organization
Multi value
Datatypes
Matching
Flexibility
Access

LDAP directory

Standard network access: LDAP
Easy distribution of data on the Net

Singularities of directories
a model closer to the real world

Best known database systems are relational ones
we will describe LDAP directories in comparison to them

Relational DBMS

Only proprietary AuthN mechanisms

Schema
Organization
Multi value
Datatypes
Matching
Flexibility
Access
AuthN

LDAP directory

Various standard AuthN mechanisms that work over the network

Singularities of directories
a model closer to the real world

Best known database systems are relational ones
we will describe LDAP directories in comparison to them

Relational DBMS

Only proprietary
Overly complicated

Schema
Organization
Multi value
Datatypes
Matching
Flexibility
Access
AuthN
Replication

LDAP directory

Standard protocols
that are included
from design

Database systems

directory data has to be put onto storage

It is possible to use different database backends to store the directory data on the filesystem.

Database systems

directory data has to be put onto storage

It is possible to use different database backends to store the directory data on the filesystem.

- Sleepy Cat Berkeley DB
  It is the most used backed for LDAP

Database systems

directory data has to be put onto storage

It is possible to use different database backends to store the directory data on the filesystem.

- Sleepy Cat Berkeley DB
  It is the most used backed for LDAP
- Relational database systems
  It is possible to use this databases as backends for LDAP enabled directories.
  There are even RPMs to do that with OpenLDAP.

Database systems

directory data has to be put onto storage

It is possible to use different database backends to store the directory data on the filesystem.

- Sleepy Cat Berkeley DB
  It is the most used backed for LDAP
- Relational database systems
  It is possible to use this databases as backends for LDAP enabled directories.
  There are even RPMs to do that with OpenLDAP.
- GNU DBM

Connecting to the directory

for performing other tasks

These are the operations that control the connection to the directory

Connecting to the directory
for performing other tasks

These are the operations that control the connection to the directory

1. StartTLS

Secure connections

It should be mandatory
Encrypted transports protect the data

Connecting to the directory
for performing other tasks

These are the operations that control the connection to the directory

1. StartTLS
2. Bind

AuthN

The principal proves identity to the service
Thus, access controls can be applied to other operations
Some operations maybe made inside anonymous sessions

Connecting to the directory
for performing other tasks

These are the operations that control the connection to the directory

1. StartTLS
2. Bind
3. Unbind

End the session

The principal tells the service it has finished working
In practice, this closes de connection

Connecting to the directory
for performing other tasks

These are the operations that control the connection to the directory

1. StartTLS
2. Bind
3. Unbind

   Abandon

End operations

LDAP protocols allows for operations to be asynchronous
Then, it is possible to abandon operations before they finish

Searching in LDAP directories

powerful, but not for the faint of heart

Search operations using the LDAP protocol
are very powerful, thus they need many parameters

Searching in LDAP directories

powerful, but not for the faint of heart

Search operations using the LDAP protocol
are very powerful, thus they need many parameters

Base DN

a.k.a. Base object

The node in the DIT that is the starting point for the search

Searching in LDAP directories

powerful, but not for the faint of heart

Search operations using the LDAP protocol
are very powerful, thus they need many parameters

scope

Base DN

search depth

- base:
  Search only the base DN object
- onelevel:
  Search first level below base DN
- subtree:
  Search all levels below base DN

Searching in LDAP directories

powerful, but not for the faint of heart

Search operations using the LDAP protocol
are very powerful, thus they need many parameters

derefAliases

scope

Base DN

follow pointers

- neverDerefAlias
- derefInSearching:
  scope onelevel or sub
- derefFindingBaseObject:
  scope base
- derefAways

Searching in LDAP directories

powerful, but not for the faint of heart

Search operations using the LDAP protocol
are very powerful, thus they need many parameters

- size limit
- derefAliases
- scope
- Base DN

how many results

Limit the number of entries the server will return in the search result

Searching in LDAP directories

powerful, but not for the faint of heart

Search operations using the LDAP protocol
are very powerful, thus they need many parameters

- time limit
- size limit
- derefAliases
- scope
- Base DN

how long to wait

Limit the time the server will spend performing the search

Searching in LDAP directories

powerful, but not for the faint of heart

Search operations using the LDAP protocol
are very powerful, thus they need many parameters

- attrsOnly
- time limit
- size limit
- derefAliases
- scope
- Base DN

just the names

Return only the names of the attributes, no values

Searching in LDAP directories

powerful, but not for the faint of heart

Search operations using the LDAP protocol
are very powerful, thus they need many parameters

- filter
- attrsOnly
- time limit
- size limit
- derefAliases
- scope
- Base DN

search expression

The query itself
The format is (*condition*)

Searching in LDAP directories

powerful, but not for the faint of heart

Search operations using the LDAP protocol
are very powerful, thus they need many parameters

- attributes
- filter
- attrsOnly
- time limit
- size limit
- derefAliases
- scope
- Base DN

search results

A list of the attributes (and possibly values)
the server should return
from the entries in the result set.
* means all attributes

Search operations
the art of query building

LDAP queries consist of comparison conditions combined by boolean operations.

Search operations
the art of query building

LDAP queries consist of comparison conditions combined by boolean operations.

Comparison operations

- Equality

Exact match

The attribute value must match the search string

Search operations
the art of query building

LDAP queries consist of comparison conditions combined by boolean operations.

Comparison operations

- Equality
- Substring

Substring match

The attribute value should
start (string*) or end (*string) with
or contain (*string*) the search string

Search operations
the art of query building

LDAP queries consist of comparison conditions combined by boolean operations.

Comparison operations

- Equality
- Substring
- Presence

Any value *

The search retrieves entries that have the attribute, whatever its value.

Search operations
the art of query building

LDAP queries consist of comparison conditions combined by boolean operations.

Comparison operators
(attribute operator searchstring)

- =

Equal

The attribute value shall equal the result of the comparison operation

Search operations
the art of query building

LDAP queries consist of comparison conditions combined by boolean operations.

Comparison operators
(attribute operator searchstring)

- =
- <=

Less than or equal

The attribute value ordering position shall be lower or equal that that of the result of the comparison operation

Search operations
the art of query building

LDAP queries consist of comparison conditions combined by boolean operations.

Comparison operators
(attribute operator searchstring)

- =
- <=
- ~=

Approximate

The attribute value should sound similar, in English, to the comparison operation. Only equality is accepted here.


Search operations
the art of query building

LDAP queries consist of comparison conditions combined by boolean operations.

Boolean operators
(operator(condition)...)

- &

And

All conditions must be true for the entry to be added to the result set

(&(mail=user@renam.md)(schacUserStatus=*:mail:active))


Search operations
the art of query building

LDAP queries consist of comparison conditions combined by boolean operations.

Boolean operators
(operator(condition)...)

- &
- |

Or

If any of the conditions is true,
the entry is added to the result set

(|(irisMailMainAddress=user@renam.md)(irisMailAlternateAddress=user@renam.md))


Search operations
the art of query building

LDAP queries consist of comparison conditions combined by boolean operations.

Boolean operators
(operator(condition)...)

- &
- |
- !

Not

The entry must not meet the condition to be added to the result set

(!(schacUserStatus=*:locked))


Updating the directory

the usual operations, with a twist

All update operations require the DN of the entry
and are atomic, i.e. searches get the whole old or new one

Updating the directory

the usual operations, with a twist

All update operations require the DN of the entry
and are atomic, i.e. searches get the whole old or new one

- Add

Insert a new entry

Creates a new entry with the provided DN,
as long as there is no other entry with the same DN

and all required attributes are present

Updating the directory

the usual operations, with a twist

All update operations require the DN of the entry
and are atomic, i.e. searches get the whole old or new one

- Add
- Delete

Remove the named entry

Deletes the entry with the provided DN

Updating the directory

the usual operations, with a twist

All update operations require the DN of the entry
and are atomic, i.e. searches get the whole old or new one

- Add
- Delete
- Modify

Modify the entry attributes

The attributes listed in the operation are altered
in the entry whose DN is provided.
Attribute operations are:

- Add attribute-value pair
- Remove attribute-pair
- Delete attribute
- Replace current value

Updating the directory

the usual operations, with a twist

All update operations require the DN of the entry
and are atomic, i.e. searches get the whole old or new one

- Add
- Delete
- Modify
- Rename

Modify the entry's DN

This operation changes the entry's DN
A copy of the original entry is first created

- rename: delete original, new in same branch
- move: delete original, new in other branch
- copy: keep original

Directory indexing
getting ready for finding entries

LDAP is read optimized
Thus proper indexing is of capital importance

Directory indexing
getting ready for finding entries

LDAP is read optimized
Thus proper indexing is of capital importance

- More indexes $=>$ faster searches

Directory indexing

getting ready for finding entries

LDAP is read optimized

Thus proper indexing is of capital importance

- More indexes $=>$ faster searches
- More indexes $=>$ slower updates

Directory indexing
getting ready for finding entries

LDAP is read optimized
Thus proper indexing is of capital importance

- More indexes $=>$ faster searches
- More indexes $=>$ slower updates
- Which attributes to index

Directory indexing
getting ready for finding entries

LDAP is read optimized
Thus proper indexing is of capital importance

- More indexes $=>$ faster searches
- More indexes $=>$ slower updates
- Which attributes to index
- How to index them

Directory indexing
getting ready for finding entries

LDAP is read optimized
Thus proper indexing is of capital importance

- More indexes => faster searches
- More indexes => slower updates
- Which attributes to index
- How to index them, it depends on the schema

Directory indexing
getting ready for finding entries

LDAP is read optimized
Thus proper indexing is of capital importance

- More indexes $=>$ faster searches
- More indexes $=>$ slower updates
- Which attributes to index
- How to index them, it depends on the schema
  - equality

Directory indexing
getting ready for finding entries

LDAP is read optimized
Thus proper indexing is of capital importance

- More indexes $=>$ faster searches
- More indexes $=>$ slower updates
- Which attributes to index
- How to index them, it depends on the schema
  - equality
  - presence

Directory indexing
getting ready for finding entries

LDAP is read optimized
Thus proper indexing is of capital importance

- More indexes => faster searches
- More indexes => slower updates
- Which attributes to index
- How to index them, it depends on the schema
  - equality
  - presence
  - substring

Access Control Lists

Controlling Access to entries and attributes

LDAP offers a very fine grained control on who can access what

Access Control Lists

Controlling Access to entries and attributes

LDAP offers a very fine grained control on who can access what

- Anonymous or bound sessions

Access Control Lists

Controlling Access to entries and attributes

LDAP offers a very fine grained control on who can access what

- Anonymous or bound sessions
- Requested attributes

Access Control Lists

Controlling Access to entries and attributes

LDAP offers a very fine grained control on who can access what

- Anonymous or bound sessions
- Requested attributes
- Search, read or write

Access Control Lists

Controlling Access to entries and attributes

LDAP offers a very fine grained control on who can access what

- Anonymous or bound sessions
- Requested attributes
- Search, read or write
- Attribute values

Access Control Lists

Controlling Access to entries and attributes

LDAP offers a very fine grained control on who can access what

- Anonymous or bound sessions
- Requested attributes
- Search, read or write
- Attribute values
- Complex search filters

Access Control Lists

Controlling Access to entries and attributes

LDAP offers a very fine grained control on who can access what

- Anonymous or bound sessions
- Requested attributes
- Search, read or write
- Attribute values
- Complex search filters
- Originating IP address

Some URLs

Wikipedia	http://en.wikipedia.org/wiki/Ldap
OpenLDAP	http://www.openldap.org/
Fedora DS	http://directory.fedoraproject.org/
RFCs	http://www.rfc-editor.org/rfcsearch.html search for LDAP