# Distributed Data Storage and Management
# Part IX: Directory Access Protocols

Saptarshi Pyne
Assistant Professor
Department of Computer Science and Engineering
Indian Institute of Technology Jodhpur, Rajasthan, India 342030

**CSL4030 Data Engineering Lecture 24**
**October 6th, 2023**

# What we discussed in the last class

Heterogeneous distributed databases

- Multidatabase systems (MDBSes)
- Cloud storage services
  - The cloud architecture and its challenges

# Network directories

- A network **directory** stores information about network resources (servers, users, services, applications, etc.). It is a special type of distributed database where information is stored in a hierarchical fashion similar to the way files are organized in a file system.

- Directory access protocols = Standardized algorithms for accessing a directory.

# Why can not we access a directory just like we query other distributed databases?

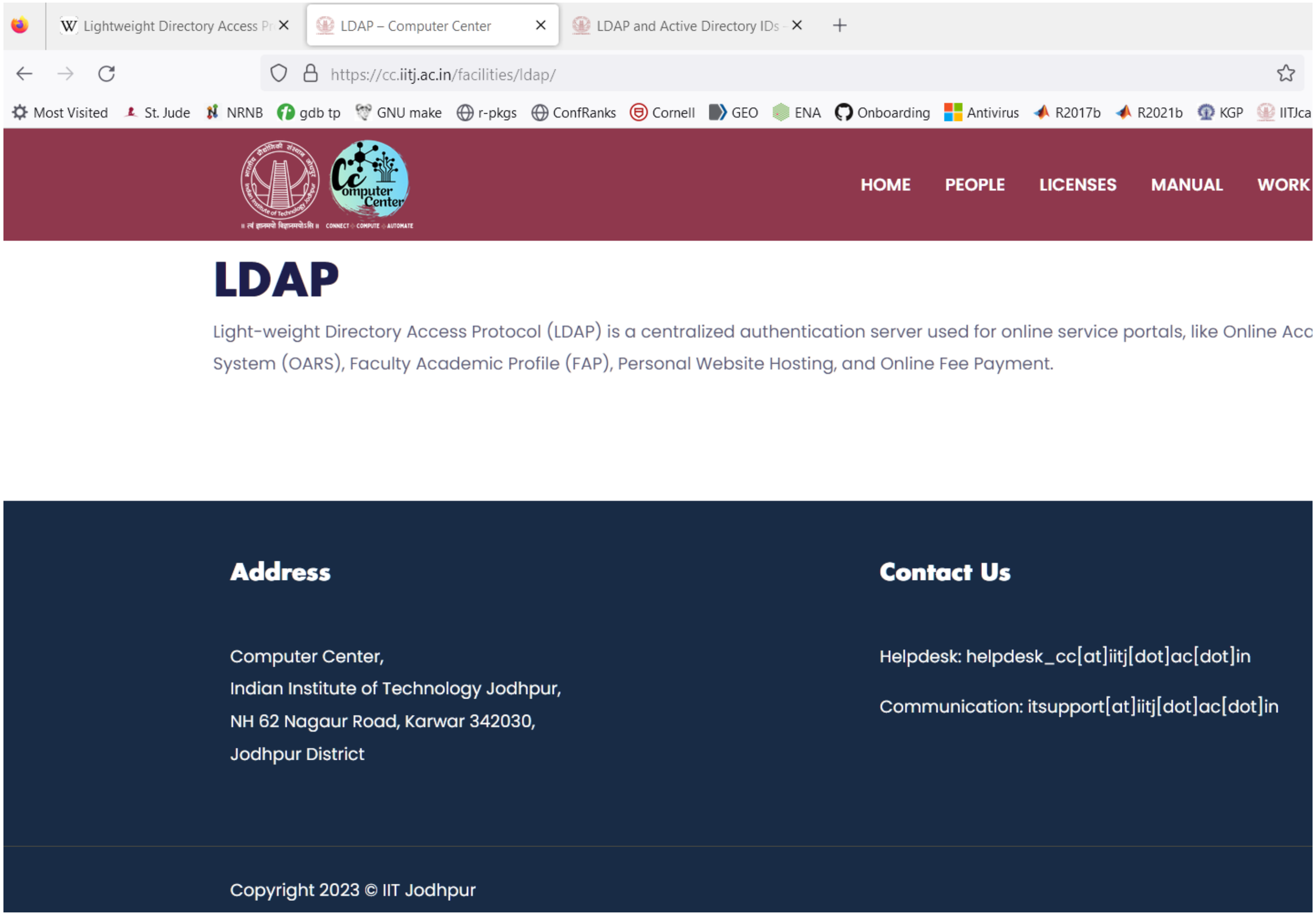
- A directory is a simple data structure. We do not need all the features of a DBMS. We can design simpler and more efficient protocols customized for accessing only directories.

- We can **name directories in a hierarchical fashion** just like the file system directories. That way we can design protocols to automatically forward queries from one directory to other directories.

# Examples of directory access protocols

- X.500 directory access protocol
- Lightweight directory access protocol (LDAP)
  - Widely used. Provides less features and a simpler API than X.500.

# The LDAP authentication server of IITJ



## LDAP

Light-weight Directory Access Protocol (LDAP) is a centralized authentication server used for online service portals, like Online Acc System (OARS), Faculty Academic Profile (FAP), Personal Website Hosting, and Online Fee Payment.

### Address

Computer Center,
Indian Institute of Technology Jodhpur,
NH 62 Nagaur Road, Karwar 342030,
Jodhpur District

### Contact Us

Helpdesk: helpdesk_cc[at]iitj[dot]ac[dot]in

Communication: itsupport[at]iitj[dot]ac[dot]in

Copyright 2023 © IIT Jodhpur

# LDAP data model

Each record is called an **entry**. It is usually stored in a binary format. For data transfer, an entry can be represented in the following plain text format know as LDAP data interchange format (LDIF).

```
dn: cn=JohnDoe, dc=ac, dc=in
o=iitj
ou=cse
cn: John Doe
givenName: John
sn: Doe
telephoneNumber: +1 888 555 6789
telephoneNumber: +1 888 555 1232
mail: john@example.com
manager: cn=Barbara Doe,dc=example,dc=com
objectClass: inetOrgPerson
objectClass: organizationalPerson
objectClass: person
objectClass: top
```

# LDAP data model (contd.)

- Each entry is identified by its **distinguished name (DN)**. A DN is composed of multiple relative DNs (**RDNs**) like relative file paths.

- Entries are organized into a hierarchical structure known as the **directory information tree (DIT)**.

- A directory system can be distributed over multiple servers. Each server contains a **subtree** of the DIT.

# LDAP data model (contd.)

- An entry/node in a server’s subtree can refer to another server’s subtree. This is called a **referral**. Referrals are the key ingredients in developing a distributed directory system.

- Client systems can access/query the directory system using APIs, such as the Apache LDAP API. The APIs use LDIF or URIs to submit an access request.

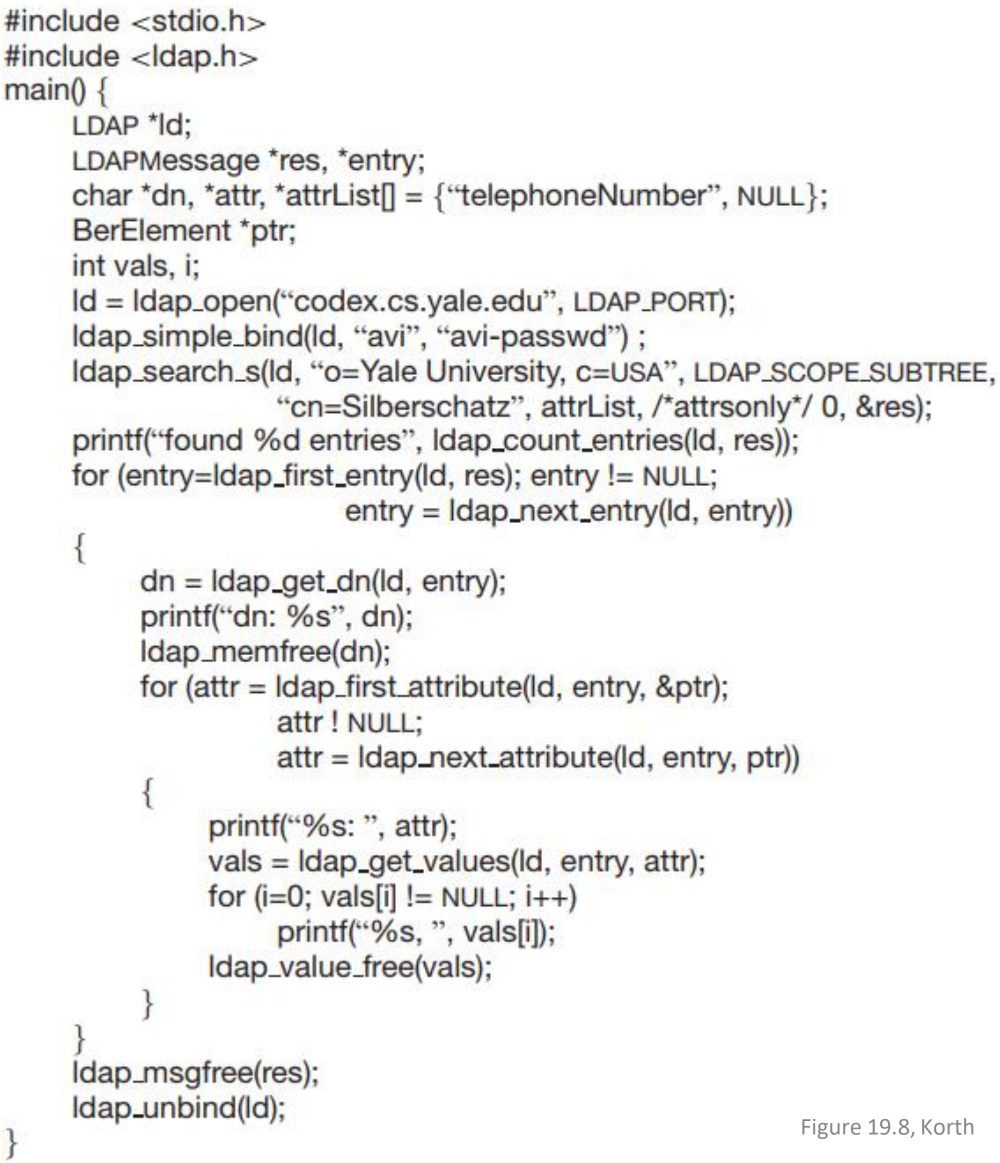
# LDAP URI access requests

Syntax:
ldap://host:port/DN?attributes?scope?filter?extensions

Example:

ldap:://codex.cse.iitj.ac.in/o=iitj,c=in?telephoneNumber?sub?cn=JohnDoe

# LDAP API access requests using C



```
#include <stdio.h>
#include <ldap.h>
main() {
        LDAP *ld;
        LDAPMessage *res, *entry;
        char *dn, *attr, *attrList[] = {“telephoneNumber”, NULL};
        BerElement *ptr;
        int vals, i;
        ld = ldap_open(“codex.cs.yale.edu”, LDAP_PORT);
        ldap_simple_bind(ld, “avi”, “avi-passwd”) ;
        ldap_search_s(ld, “o=Yale University, c=USA”, LDAP_SCOPE_SUBTREE,
                        “cn=Silberschatz”, attrList, /*attrsonly*/ 0, &res);
        printf(“found %d entries”, ldap_count_entries(ld, res));
        for (entry=ldap_first_entry(ld, res); entry != NULL;
                                entry = ldap_next_entry(ld, entry))
        {
                dn = ldap_get_dn(ld, entry);
                printf(“dn: %s”, dn);
                ldap_memfree(dn);
                for (attr = ldap_first_attribute(ld, entry, &ptr);
                                attr ! NULL;
                                attr = ldap_next_attribute(ld, entry, ptr))
                {
                        printf(“%s: ”, attr);
                        vals = ldap_get_values(ld, entry, attr);
                        for (i=0; vals[i] != NULL; i++)
                                printf(“%s, ”, vals[i]);
                        ldap_value_free(vals);
                }
        }
        ldap_msgfree(res);
        ldap_unbind(ld);
}
```

Figure 19.8, Korth

# References

- A. SILBERSCHATZ, H.F. KORTH, S. SUDARSHAN (2011), Database System Concepts, McGraw Hill Publications, 6th Edition.
  - Chapter 19. Distributed Databases

- https://en.wikipedia.org/wiki/Lightweight_Directory_Access_Protocol

# Thank you