NETWORK ADMINISTRATION

Lecture 2

Logical organization of the computer network. Directory services - concept and architectures.

Assoc. Prof. Rosen Radkov PhD

NETWORK ADMINISTRATION

Main components of the computer network

Hardware: desktops and laptops, servers, mobile devices, printers, network devices, etc .;

Software: system and application software;

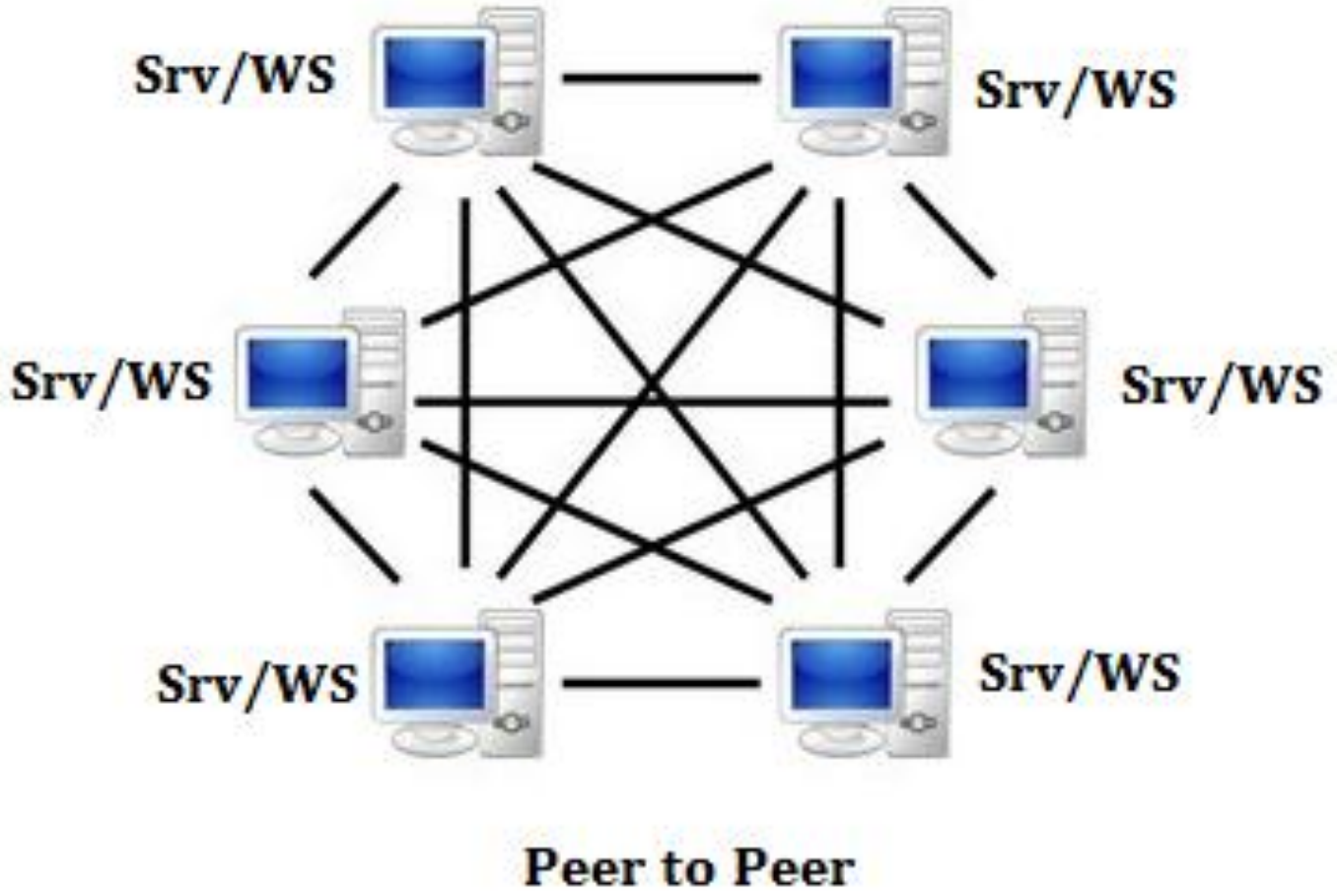
Resources: files, printers, networks, services, etc .;

Accounts: names through which the access to the resources is regulated;

Assoc. Prof. Rosen Radkov PhD

NETWORK ADMINISTRATION

Networks with distributed resources (peer-to-peer)



Assoc. Prof. Rosen Radkov PhD

NETWORK ADMINISTRATION

Main characteristics

- For networks with a small number of devices
- Designed for home or micro enterprise companies
- Simplified architecture
- They use desktop operating systems
- Ability to connect devices to different operating systems

Assoc. Prof. Rosen Radkov PhD

Advantages

- Simplified administration
- Low cost of the initial investment
- Low maintenance costs

Disadvantages

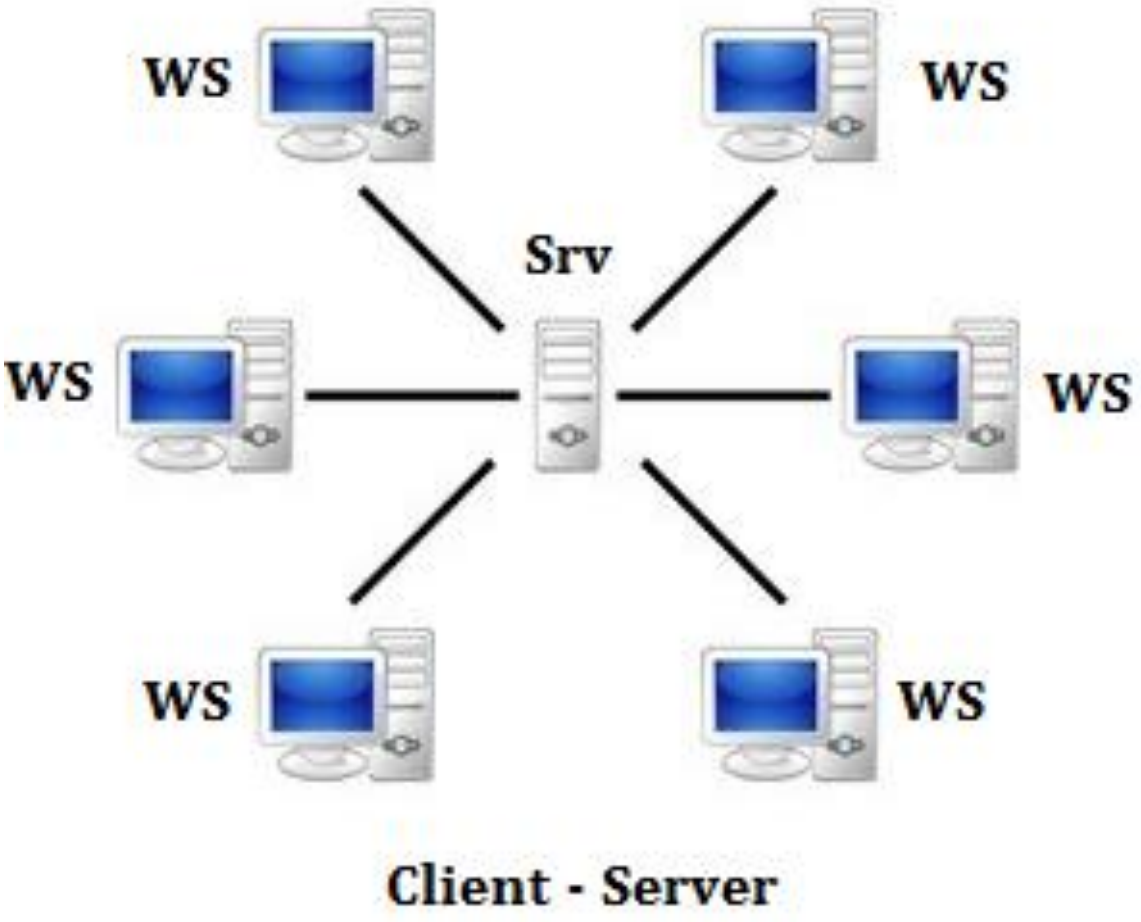
- All devices need to be administered
- Inability to use common names of users and groups
- It is impossible to impose common or group policies
- If a change is required, it must be performed on all affected devices
- Low reliability
- Low information security

Disadvantages

- Problems organizing remote access to resources
- Lack of centralized management
- Difficulties in finding the resources available to a user
- Difficulties when needing to service new business processes

NETWORK ADMINISTRATION

Client-Server networks



Assoc. Prof. Rosen Radkov PhD

NETWORK ADMINISTRATION

Main characteristics

- For networks with a large number of devices
- Designed for small, medium or large enterprise type companies
- Complicated architecture
- Server and desktop operating systems are also used
- Ability to connect devices running operating systems from different manufacturers
- High reliability
- Centralized service management

NETWORK ADMINISTRATION

Advantages

- Centralized administration and management is possible
- Opportunity to impose common and group policies
- Easy organization of access control resources
- Available resources are easy to find
- It is possible to achieve high reliability
- Opportunities for achieving high information security

Advantages

- Remote resources access is possible
- Easy scaling

Disadvantages

- Complex IT infrastructure
- Complicated administration
- Initial investment has a high cost
- High cost of current maintenance
- Need for a competent and experienced IT staff

Differences between desktop computers and servers (recap)

- Amount and type of processors;
- RAM volume;
- Disk system type: RAID arrays; different type of discs;
- Reliability of components : MTBF;
- Diagnostic tools;
- Power supply redundancy;
- Operation under pressure - hot swap;
- etc.

Assoc. Prof. Rosen Radkov PhD

Differences between desktop and server OS (recap)

- Number of processors supported;
- Supported volume of RAM;
- Supported disk space;
- Supported services: DHCP, DNS, Active directory, Print server, Domain controller, Web server and others;
- Simultaneous operation mode of many users;
- Support for remote access technologies;
- Virtualization platform support;
- etc.

Conclusions

There is a need for a service that allows for:

- organizing a catalog of users, groups, resources and services;
- resource access control management;
- reliable and secure verification and management of identities;

Concept of a directory

What is the directory?

- A service, i.e. software;
- A service, that maintains a list of objects, arranged in a certain way, of an organization (users, e-mail addresses, digital certificates, computers, resources, etc.) in a specialized database (repository).
- Service that provides adequate response to inquiries about the objects in the maintained list;

Concept of a directory

- Allows users or applications to find resources by certain characteristics of theirs (for example, to find the name of the employee, his email address, phone or office number, etc., by the use of his username as a query);

- It differs from other databases in that it is more often queried (read operation) than updated (write operation) and therefore optimized for reading;

Concept of a directory

- Most directories do not support transactions or are limited to performing operations on the database itself;
- Directories do not support common ways to access them, such as SQL. They work with simplified and optimized access protocols. This allows the client and server part to be simplified as well;
- Sometimes it is necessary for the directory service to turn to other services to obtain the necessary information, i.e. it is only one of the services needed for an organization;

Concept of a directory

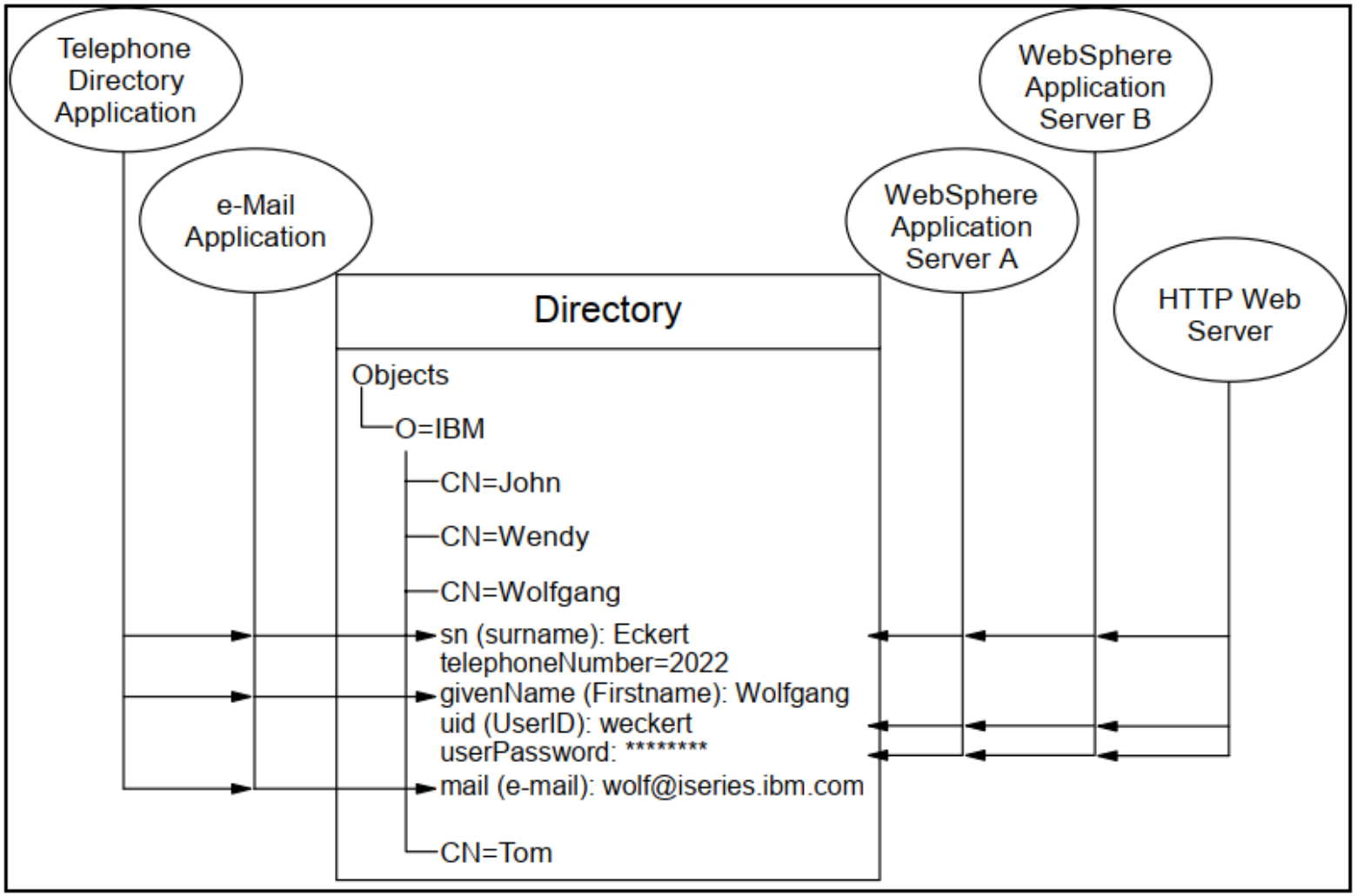
- Directory services must allow the management of connections between sites;

- The directory service must be:

- flexible – to store the required data set;
- protected – to provide access from both the Intranet and the Internet;
- scalable according to business needs;
- use an open standards-based protocol;

NETWORK ADMINISTRATION

Concept of a directory



Directory evolution (1)

- 1988 г. – X.500 of the ITU, based on the ISO model, not on TCP / IP, and uses Directory Access Protocol (DAP) - complex and includes more features than necessary for objects in a directory;

- 1990 г. - (ISO 9594 Data Communication Network Directory), the recommendations X.500-X.521 refer to it;

- 1993 г. - the first version of the Lightweight Directory Access Protocol (LDAP) published in RFC1487 (X.500 Lightweight Access Protocol) - practically never got used (developed by a team of scientists from the University of Michigan);

Directory evolution (2)

- 1995 - LDAPv2 - before that LDAP was used primarily as a gateway between X.500 servers. At the end of 1995 the first LDAP server is launched in Michigan;

- 1997 - An important update to LDAP was published in RFC2251, which provided several new and significant features that made it a sufficiently reliable and flexible protocol. Since then, many companies such as Sun, Novell (now NetIQ), Microsoft and in the past Netscape have developed directories based on LDAP;

- 2002 - LDAPv3 is published in RFC3377, which summarizes all the "main" LDAP RFC documents;

Directory evolution (3)

Accompanying RFCs:

- Lightweight Directory Access Protocol (v3): Attribute Syntax Definitions (RFC 2252)
- _x005F_x005F_x0002_Lightweight Directory Access Protocol (v3): UTF-8
- String Representation of Distinguished Names (RFC 2253)
- The String Representation of LDAP Search Filters (RFC 2254)
- The LDAP URL Format (RFC 2255)

Directory evolution (4)

- A Summary of the X.500(96) User Schema for use with LDAPv3 (RFC 2256)
- _x005F_x005F_x0002_Authentication Methods for LDAP (RFC 2829)
- LDAPv3: Extension for Transport Layer Security (RFC 2830)
- Lightweight Directory Access Protocol (v3): Technical Specification (RFC 3377)

LDAP applications

- Microsoft Active Directory
- NetIQ eDirectory
- Apache Directory Server
- Red Hat Directory Server
- Oracle Internet Directory
- IBM Tivoli Directory Server
- OpenLDAP
- IBM Lotus Domino

What is LDAP?

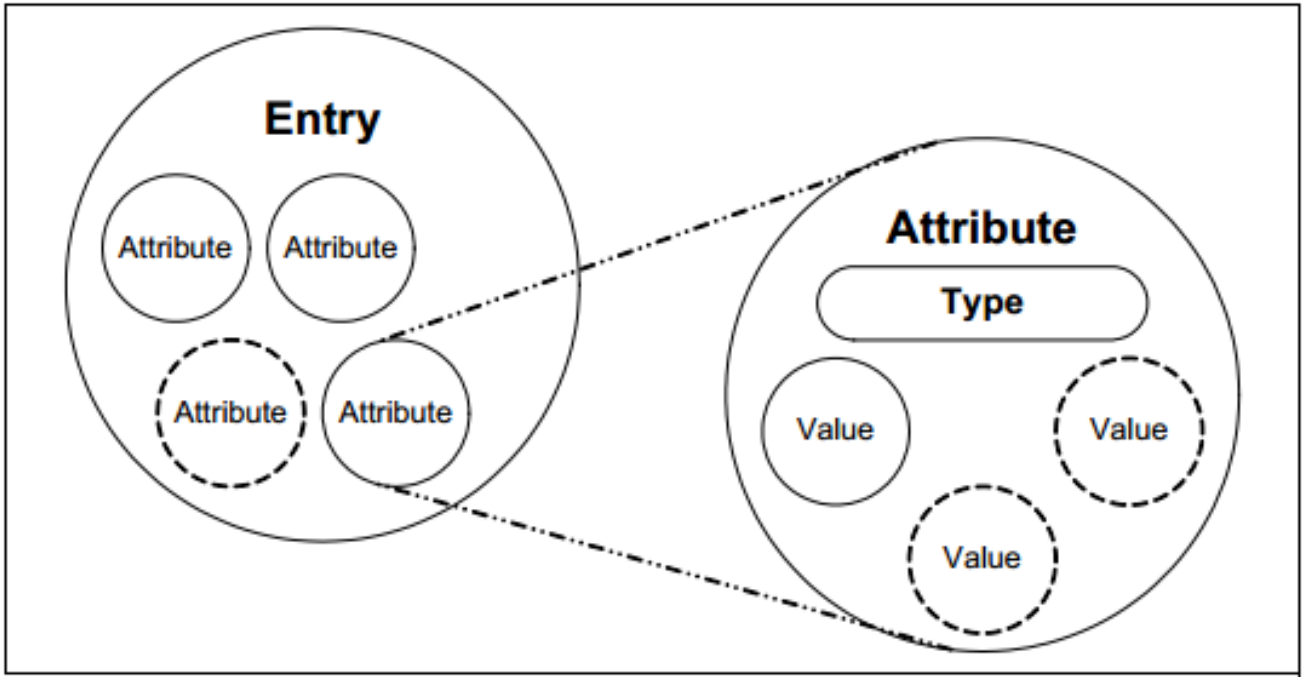
- An open standard that defines a standard method for accessing and updating information in the directory. APIs for various programming languages have been developed;
- Defines a communication protocol, i.e. the format of the messages and the transport of the messages exchanged between the client and the directory server;
- Uses TCP port 636 for secure communication and 389 for unsecured communication;
- LDAP is a protocol, not a directory!!!

Communication between LDAP client and server includes :

- The client creates a session with an LDAP server – binding
- The client can use a username and password for authentication or build an anonymous session with default rights
- The client performs operations (adding, deleting, changing) with the database, with the possibility of using filters
- When the client completes the queries, it closes the session - unbinding

Concept and architecture of the LDAP (1)

- Informational model



Concept and architecture of the LDAP (2)

- Attributes syntax

Syntax	Description
bin	Binary information
ces	String (case sensitive)
cis	String (not case sensitive)
tel	Telephone number (text)
dn	Distinct name
Generalized Time	Date and time (string)
Postal Address	Postal address (rows separated by $)

Concept and architecture of the LDAP (3)

- Common LDAP attributes

Attribute	Syntax	Description	Example
CommonName, cn	cis	Object name	Ivan Ivanov
surname, sn	cis	Family name	Ivanov
telephoneNumber	tel	Telephone number	+35952777777
OrganizationalUnitName, ou	cis	Department name	NetAdmin
owner	dn	Distinct name of a person owning the object	cn=Ivan Ivanov, o=CompanyNet, c=bg
organization, o	cis	Organization name	CompanyNet

Concept and architecture of the LDAP (4)

- Schemas define the type of object and the list of its attributes that can be stored in the directory. Each server can have its own schema, but a standard is expected to exist in order to interact between directories.

Object classes and required attributes

Object class	Description	Required attribute
InetOrgPerson	Defines employee records	commonName (cn) surname (sn) objectClass
OrganizationalUnit	Defines records for the organizational unit	ou objectClass
Organization	Defines records for the organization	o objectClass

Concept and architecture of the LDAP (5)

- Object attributes can be required and optional, for example *Surname* is required, but *Description* is not.
- The schema has a validation mechanism that verifies that all mandatory attributes are entered before storing the object record.
- Optional attributes can be entered at any time.

Concept and architecture of the LDAP (6)

- Subclassing: Objects can be extracted from other objects, this is called subclassing. If an object called *person* is defined that has an attribute *surname* (and others), then the class of the *organizationalPerson* object can be defined as a subclass of person. Then it will have the attributes defined for person as well as others, if such exist.
- The person object class is called "superior".
- A special class named top does not have a "superior" class. This class must include the objectClass attribute.
- The attributes of the top class are present in all directory entries.

Concept and architecture of the LDAP (7)

- Each directory entry has a special attribute called objectClass. The value of the objectClass attribute is a list of two or more names in the schema. The schema determines what type of object(s) the record represents. One of the values must be either top or alias. Alias is used if the record is an alias for another record, otherwise top is used. The objectClass attribute determines what attributes the record should have and what it can have.

- *extensibleObject* – allows any attribute to be saved in the record.

Concept and architecture of the LDAP (8)

- LDAP Data Interchange Format (LDIF) – for processing large volumes of data, i.e. many records at once.

Format:

dn: <distinguished name>
<attrtype> : <attrvalue>
<attrtype> : <attrvalue>
...

Concept and architecture of the LDAP (9)

LDIF file example:

```
dn: o=ibm.com
objectclass: top
objectclass: organization
o: ibm.com

dn: ou=People, o=ibm.com
objectclass: organizationalUnit
ou: people

dn: ou=marketing, o=ibm.com
objectclass: organizationalUnit
ou: marketing

dn: cn=John Smith, ou=people, o=ibm.com
objectclass: top
objectclass: organizationalPerson
cn: John Smith
sn: Smith
givenname: John
uid: jsmith
ou: marketing
```

Concept and architecture of the LDAP (10)

LDAP scheme

◆ *Objectclass* – indicates the type of object

- Typical object types are: *person, organization, organizational unit, domain component* и *groupOfNames*

- There are object classes that determine the relationship of the object to other objects. For example, the top class indicates that this object has child objects below it in a hierarchical tree structure

- Object classes can be declared as: *abstract*, *structural*, *auxiliary*

Concept and architecture of the LDAP (11)

- LDAP object classes define a set of standard attributes, which are called mandatory attributes (*must contain*) and optional attributes (*may contain*).

Example of an LDAP object definition :

objectclass: top
objectclass: person
objectclass: organizationalPerson
objectclass: inetOrgPerson
objectclass: eDominoAccount

Concept and architecture of the LDAP (12)

Example:

objectclass: top
objectclasses=(2.5.6.0 NAME 'top' DESC 'Standard ObjectClass' ABSTRACT MUST (ObjectClass))

objectclass: person
objectclasses=(2.5.6.6 NAME 'person' DESC 'Defines entries that generically represent people.' SUP 'top' STRUCTURAL MUST (cn $ sn) MAY (userPassword $ telephoneNumber $ seeAlso $ description))

objectclass: organizationalPerson
objectclasses=(2.5.6.7 NAME 'organizationalPerson' DESC 'Defines entries for people employed by or associated with an organization.' SUP 'person' STRUCTURAL MAY (title $ x121Address $ registeredAddress $ destinationIndicator $ preferredDeliveryMethod $ telexNumber $ teletexTerminalIdentifier $ internationalISDNNumber $ facsimileTelephoneNumber $ street $ postalAddress $ postalCode $ postOfficeBox $ physicalDeliveryOfficeName $ ou $ st $ l))

Concept and architecture of the LDAP (13)

objectclass: inetOrgPerson
objectclasses=(2.16.840.1.113730.3.2.2 NAME 'inetOrgPerson' DESC 'Defines entries representing people in an organizations enterprise network.' SUP 'organizationalPerson' STRUCTURAL MAY (audio $ businessCategory $ carLicense $ departmentNumber $ employeeNumber $ employeeType $ givenName $ homePhone $ homePostalAddress $ initials $ jpegPhoto $ labeledURI $ mail $ manager $ mobile $ pager $ photo $ preferredLanguage $ roomNumber $ secretary $ uid $ userCertificate $ userSMIMECertificate $ x500UniqueIdentifier $ displayName $ o $ userPKCS12))

Concept and architecture of the LDAP (14)

OIDs are a string of numbers and are designed to be globally unique (ISO). They are formed using a unique numeric string (eg 1.3.4.7.4.17) and additional digits are added in a unique way (such as 1.3.4.7.4.17.1, 1.3.4.7.4.17.2, 1.3.4.7. 3, etc.).

A company or organization can acquire a *branch* from a *root* in the OID tree. OIDs terminology uses *arc* instead of *branch*, and *vertex* instead of *root*. In the above example, the branch is - 1.3.4.7.4.17).

The organization can then extend the arc (called an arc), as shown above, to create additional OIDs and arcs.

Concept and architecture of the LDAP (15)

Example: OIDs of IBM

1 (ISO-assigned OID)

1.3 (ISO-identified organization)

1.3.18 (IBM)

1.3.18.0 (IBM Objects)

Concept and architecture of the LDAP (16)

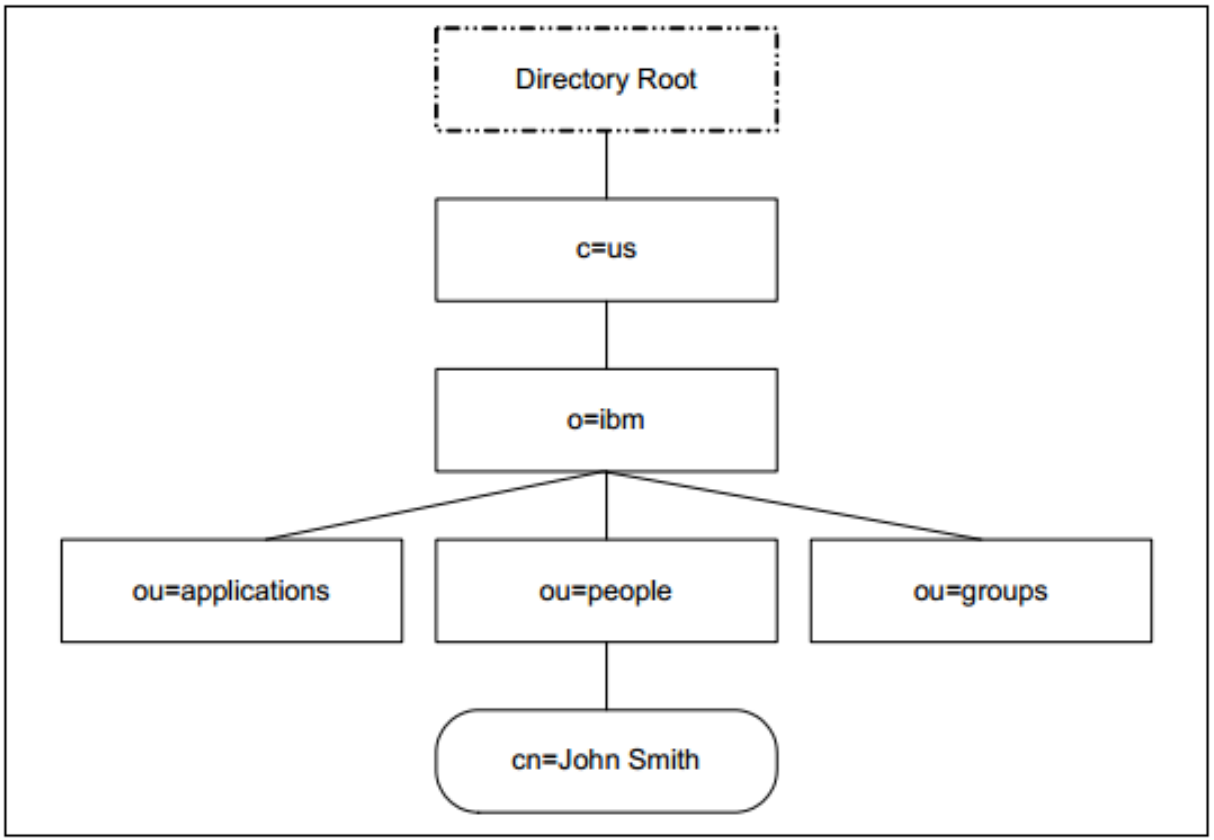
- Attributes of objects. The object class defines the attributes and the type of data contained in the object. Typical attributes are: *cn (common name), sn (surname), givenName, mail, uid, userPassword*
Like object classes, they are defined by a unique OID number.

attribute: name
attributetypes=(2.5.4.41 NAME 'name' DESC 'The name attribute type is the attribute supertype from which string attribute types typically used for naming may be formed. It is unlikely that values of this type itself will occur in an entry.' EQUALITY 1.3.6.1.4.1.1466.109.114.2 SUBSTR 2.5.13.4 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 USAGE userApplications)

attribute: sn
attributetypes=(2.5.4.4 NAME ('sn' 'surName') DESC 'This is the X.500 surname attribute, which contains the family name of a person.' SUP 2.5.4.41 EQUALITY 2.5.13.2 ORDERING 2.5.13.3 SUBSTR 2.5.13.4 USAGE userApplications)

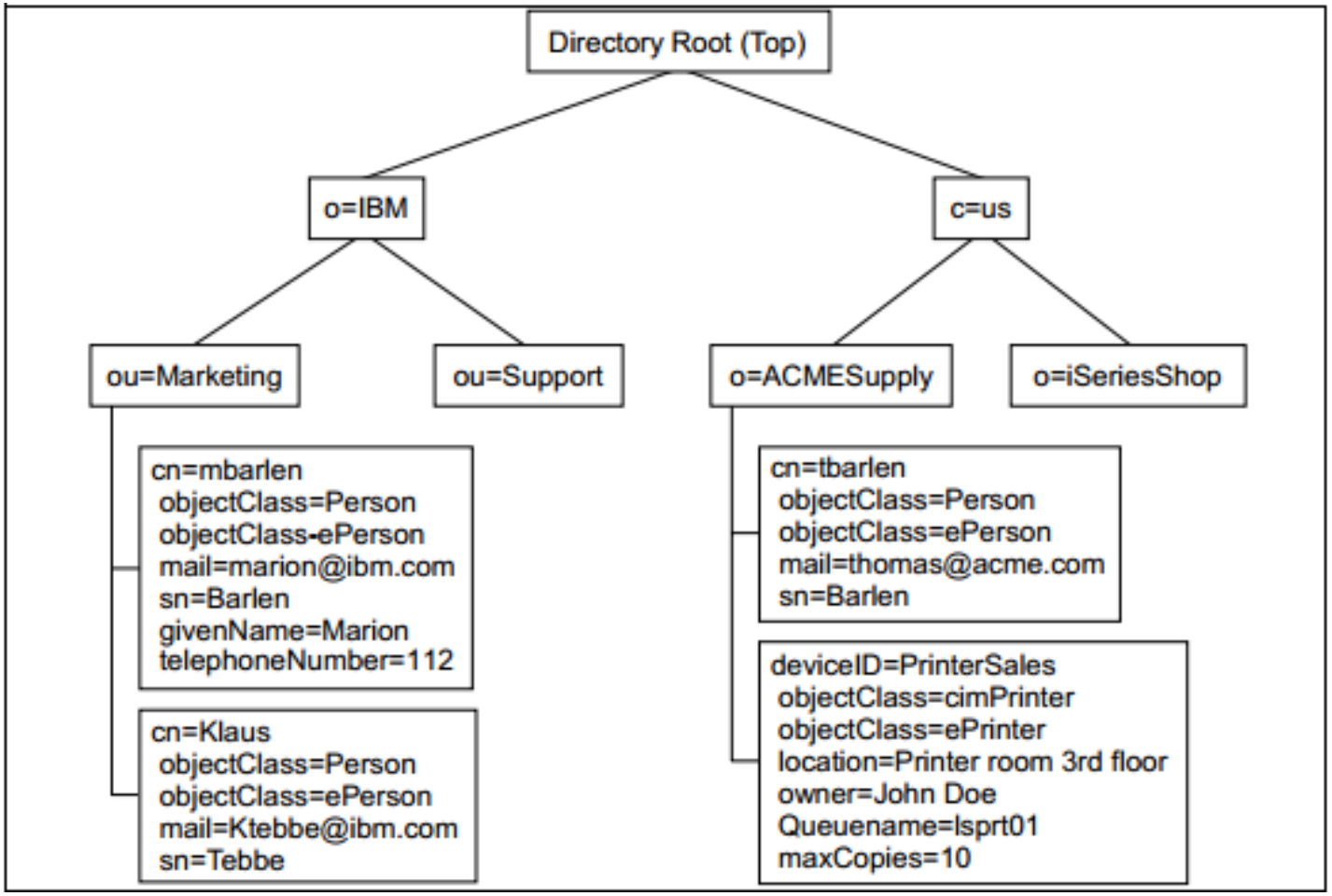
Concept and architecture of the LDAP (17)

- Model name in LDAP - organized in the form of a tree called Directory Information Tree (DIT)



NETWORK ADMINISTRATION

Example for a Directory Information Tree (DIT)



Assoc. Prof. Rosen Radkov PhD

Concept and architecture of the LDAP (18)

- LDAP Distinguished Name (DNs)
- Relative Distinguished Name (RDN)

```
cn=Sandy Brown,ou=marketing,o=ibm,c=US
cn=Leslie Jones,ou=development,o=ibm,c=US
```

Синтаксис:

```
<name> ::= <name-component> ( <spaced-separator> )
         | <name-component> <spaced-separator> <name>

   <spaced-separator> ::= <optional-space>
                          <separator>
                          <optional-space>

   <separator> ::=  "," | ";"

   <optional-space> ::= ( <CR> ) *( " " )
```

Concept and architecture of the LDAP (19)

```
<name-component> ::= <attribute>
        | <attribute> <optional-space> "+"
          <optional-space> <name-component>

<attribute> ::= <string>
        | <key> <optional-space> "=" <optional-space> <string>

<key> ::= 1*( <keychar> ) | "OID." <oid> | "oid." <oid>
<keychar> ::= letters, numbers, and space

<oid> ::= <digitstring> | <digitstring> "." <oid>
<digitstring> ::= 1*<digit>
<digit> ::= digits 0-9

<string> ::= *( <stringchar> | <pair> )
         | '"' *( <stringchar> | <special> | <pair> ) '"'
         | "#" <hex>


<special> ::= "," | "=" | <CR> | "+" | "<" |  ">"
         | "#" | ";"

<pair> ::= "\" ( <special> | "\" | '"')
<stringchar> ::= any character except <special> or "\" or '"'


<hex> ::= 2*<hexchar>
<hexchar> ::= 0-9, a-f, A-F
```

Concept and architecture of the LDAP (20)

- Functional model of the LDAP

- authentication: *Bind, Unbind, Abandon*

- query: *Search* и *Compare*

The start point and range must be defined for the search: *baseObject*, *singleLevel* или *wholeSubtree*

Example: *Find the postal address for cn=John Smith,o=IBM,c=DE*

- update: *Add, Delete, Modify, ModifyRDN*

Concept and architecture of the LDAP (21)

- LDAP security model - based on the operation : *Bind*

- Authentication

- Integrity

- Confidentiality

- Authorization

NETWORK ADMINISTRATION

THANK YOU FOR THE ATTENTION!

Assoc. Prof. Rosen Radkov PhD