



On line Training Material on AAI development

Introduction and AAI concepts

Diego Reis | RNP | 27-02-16 | WP2



What is Identity Management?



- A system of standards, procedures and technologies that provides electronic credentials to individuals.
- Maintains authoritative information about individuals.
- Establishes the trust needed for transactions.
- Facilitates and controls user access to online applications or resources.



Identity Management



Who are you? (identification)

- Collect personally identifying information to prove you are who you say you are (identity proofing), such as drivers license, passport, or biometric data
- Assign attributes [(name, address, college or university, department, role (faculty, staff, student), major, email address)]

How can you prove it? (authentication)

- Verifying that the person seeking access to a resource is the one previously identified and approved



Identity Management



Authentication does not verify that the identity proofing is correct. It establishes that the previously identified person is the same one who is seeking access to a resource.



Key Entities



Three entities involved in gaining access to a resource:

1. Subject (i.e. user) – The person identified and the subject of assertions (or claims) about his or her identity.
2. Identity Provider – Typically the university or organization that maintains the identity system, identity-proofs the subject and issues a credential. Also provides assertions or claims to the service provider about a subject's identity.
3. Service Provider (sometimes called the relying party) – Owner/provider of the protected resource to which the subject would like to access. Consumes the assertion from the identity provider and makes an authorization decision.

Key Terms



Authentication – Verification (via a user ID and password) that a subject is associated with an electronic identifier. This is the responsibility of the identity provider.

Authorization – Determining whether a subject is eligible to gain access to a resource or service. The authorization decision is made by the service provider and is based on the attributes provided by the identity provider.

Attribute – A single piece of information associated with an electronic identity database record, such as name, phone number, group affiliation, email address, major.



The Problem

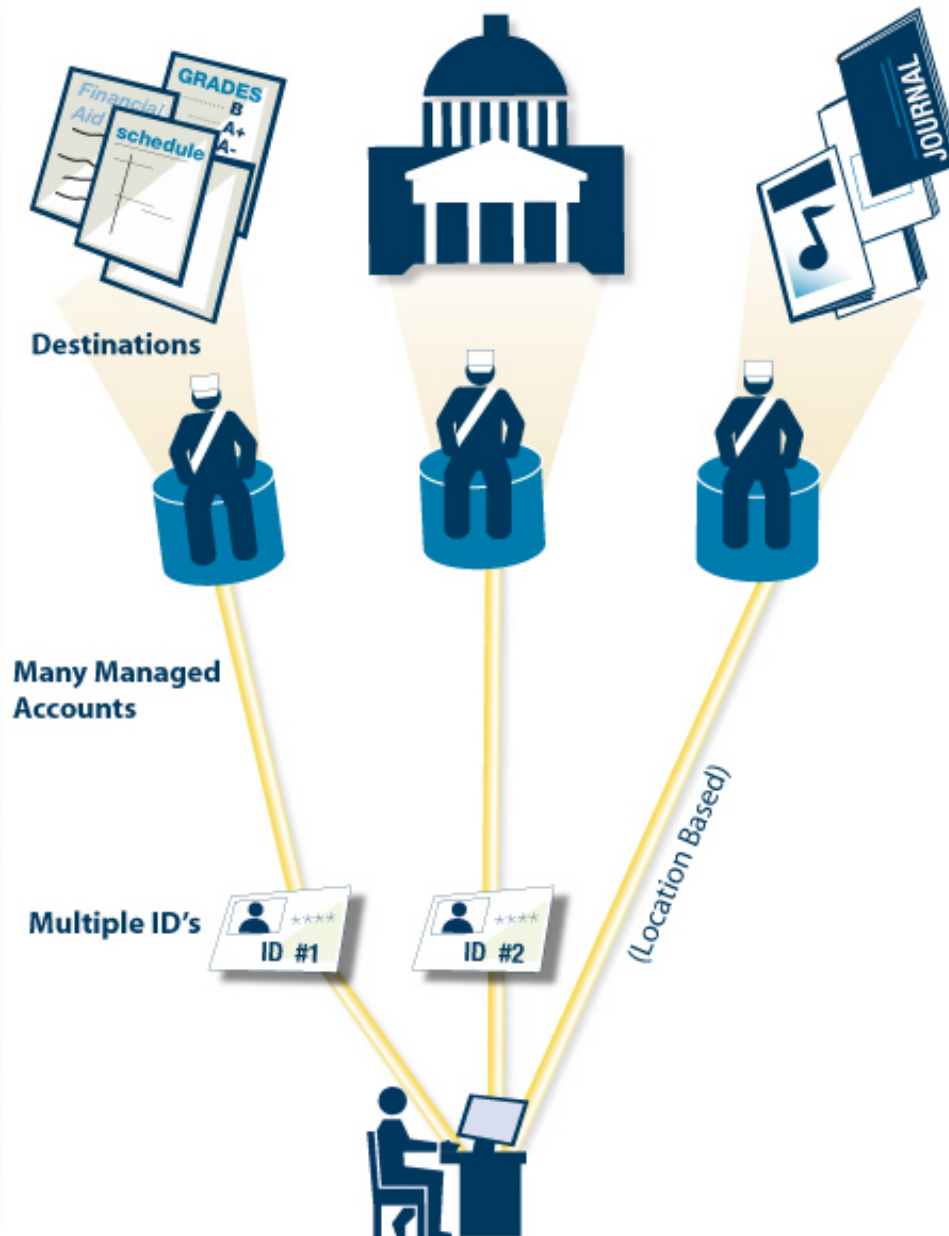


The system of authentication and authorization, and the passing of attributes, requires that the identity provider and service provider agree on policies and procedures.

When you have one identity provider working with many service providers – or one service provider working with many identity providers – things get complicated.

Individual service providers keep subject information in their own databases, or may want direct access to an identity provider's database, or may require frequent batch uploads of identity information.





1. Tedious user registration at all resources
2. Unreliable and outdated user data at resources
3. Different login process at each resource
4. Many different passwords
5. Identity provider may need to support multiple custom authentication methods and/or be asked for access to its identity database

The Problem



- Growing number of applications – on-campus and outsourced or hosted
- All of these service providers must:
 - Verify the identity of users (faculty, staff, students, others)
 - Know who's eligible to access the service
 - Know the student is active and hasn't left school
- Increase in outsourced or cloud services raises concerns about the security and privacy of the identity data

A Solution: Federated Identity Management



Federation: An association of organizations that come together to exchange information, as appropriate, about their users and resources in order to enable collaborations and transactions.

All participants in a federation agree on the same policies and procedures related to identity management and the passing of attributes.

Instead of one-to-one relationships, the federation allows one-to-many relationships.



Federated Identity Management



- Parties agree to leverage the identity provider's database, rather than creating separate data stores
- Users no longer register with the service provider, using their university credentials for transactions
- Single sign-on convenience for users
- Identity provider does the authentication; service provider does the authorization
- Attributes are the key – maintain privacy and security

Federated authentication and authorization infrastructure



- **Motivation**

- Dissemination of technologies and tools which stimulate sharing resources, information services and inter-institutional

- **Challenges**

- Develop secure and scalable environment to enable the envisioned cooperation actually happens



Federated authentication and authorization infrastructure



- **Examples of inner services:**
 - Project registration, registration of students, record notes, document sharing etc.
- **Examples of outer services :**
 - Access to digital libraries, resource sharing (CPU cycles, storage space), distance learning etc.
- A federation offers to institutions the authentication and authorization infrastructure necessary to interconnect people and share resources, information and services



Federated authentication and authorization infrastructure



- **What is a federation?**

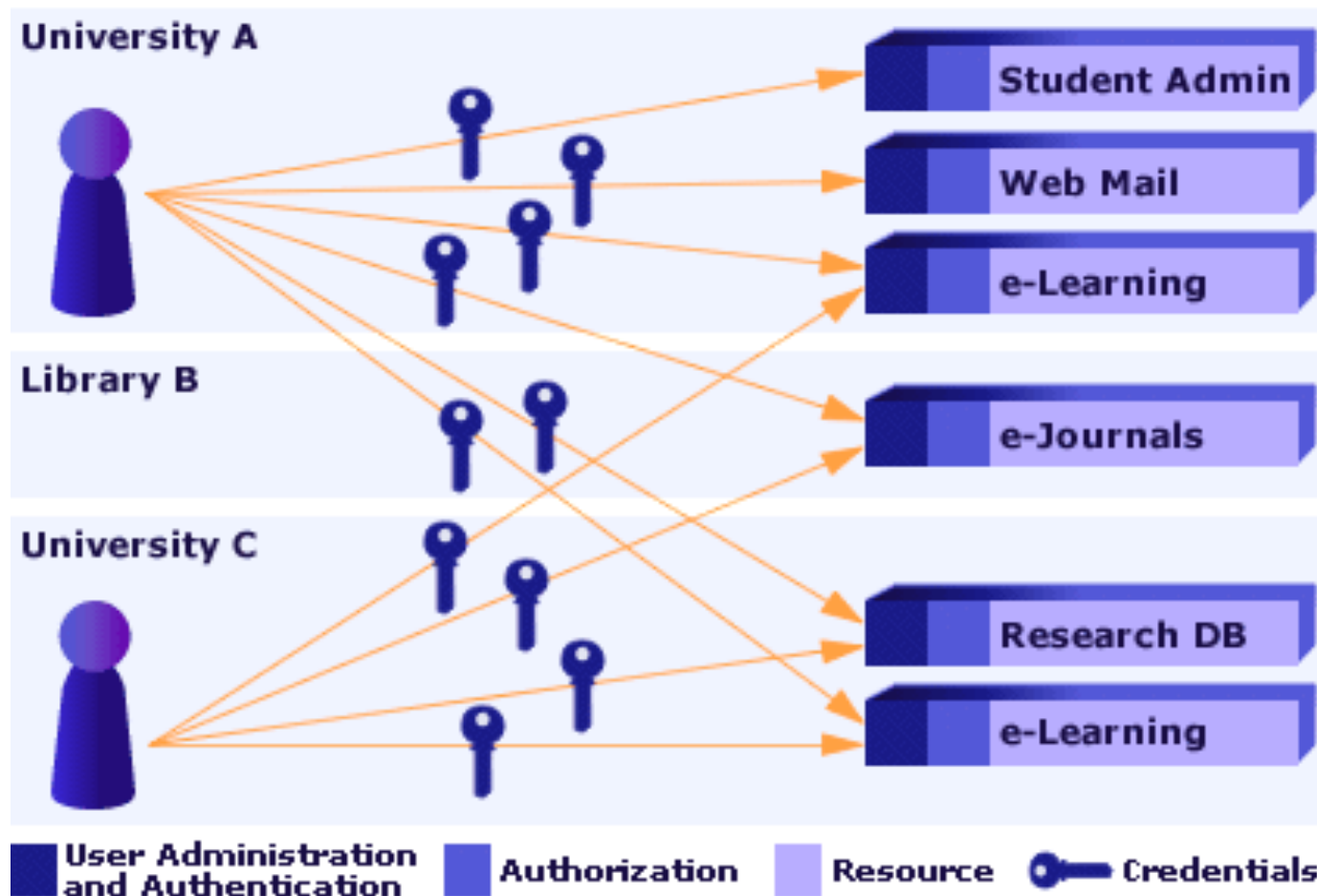
- Reliable network type that reduces bilateral contracts between users and service providers
- Implement the principle of federated identity:
 - Institutions implement different authentication methods, while maintaining interoperability



Federated authentication and authorization infrastructure

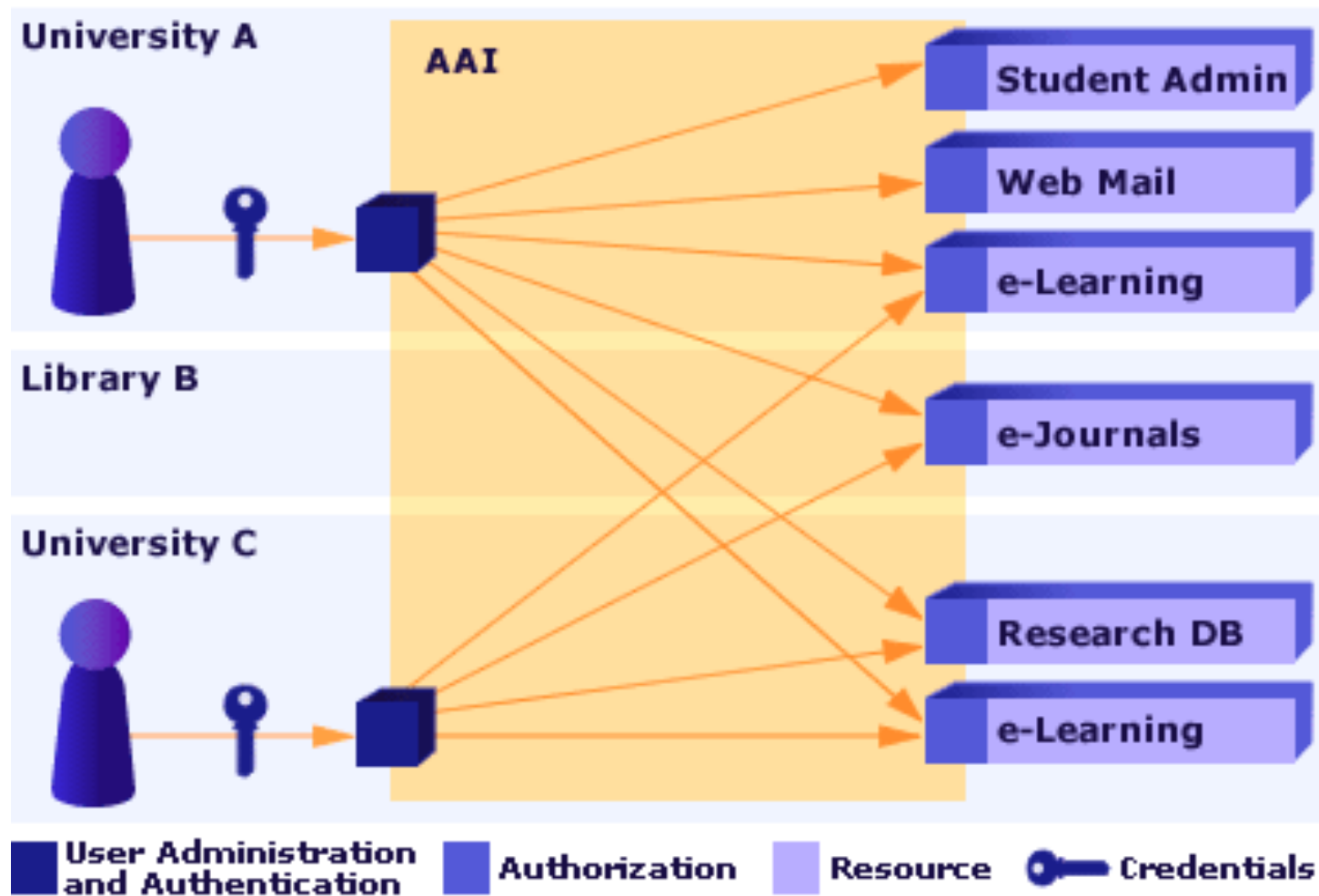


Fonte: SWITCH AAI-Federation



Federated authentication and authorization infrastructure

Fonte: SWITCH AAI-Federation



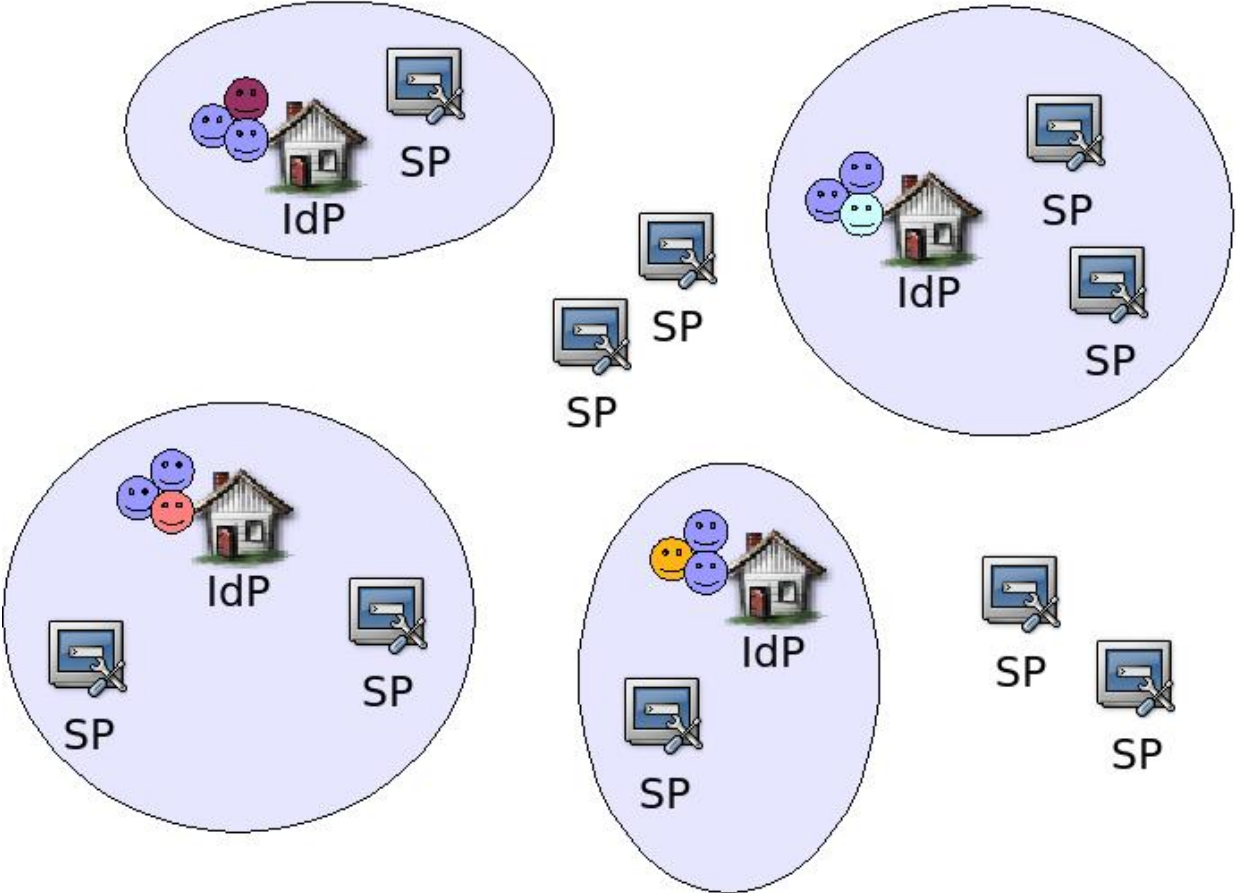
Elements of a federation



- **A federation includes two elements:**
 - Identity Provider(IdP)
 - Service Provider(SP)
- **Actors in a federation:**
 - User: wants to use a protected resource
 - Resource provider: application with an embedded SP
 - User's institution: has an IDP and an internal authentication process



Elements of a federation



Additional component of a federation



- **Where Are You From (WAYF) / Discovery Service (DS)**
 - Element that centralizes informations about identity providers from a federation



Identity Providers



- **Implement internal identity management policies of a institution**
 - User's attributes
 - Name, entry date, office, enrollment, etc.†
 - Authentication method
 - Login/password, certificates, etc.†
 - Unique identifier for each person linked to institution



Directories



- A directory is a database designed to cater mainly to large amounts of consultation and not the large volumes of updates.

The screenshot shows the Active Directory Explorer interface. The left pane displays a tree view of the directory structure. The right pane shows the properties of the selected object, 'OU=OUtest2'.

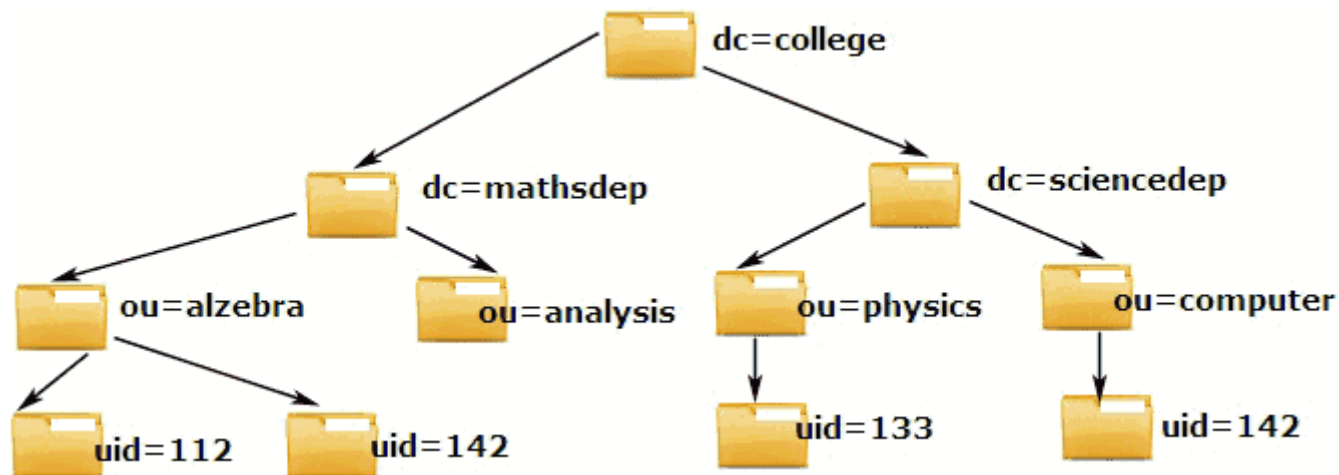
Name	Value
CN	Bob Nitz4
CN	OUtestGroup2
objectClass	top
objectClass	organizationalUnit
ou	OUtest2
distinguishedName	OU=OUtest2,OU=OUtest,DC=pa
instanceType	4
whenCreated	20100209004142.0Z
whenChanged	20100209004142.0Z
uSNCreated	189512
uSNChanged	189512
name	OUtest2
objectGUID	D2 DE F9 EA 7A 10 09 40 AF
objectCategory	CN=Organizational-Unit,CN=Sche
dSCorePropagationData	20101020233130.0Z
dSCorePropagationData	16010101000001.0Z
createTimeStamp	20100209004142.0Z
modifyTimeStamp	20100209004142.0Z
subSchemaSubEntry	CN=Aggregate,CN=Schema,CN=



Directories



- The directories store your information in a hierarchical form, not using data tables, like an ordinary relational base.
- Instead, the data are organized in a DIT (Directory Information Tree), where each vertex is a record, and each record is a collection of information about a object that we want to store.



Directories



- The basic unit of information stored in the directory entry is called entry.
- Entries represent objects of interest in the real world, as people, servers or organizations.
- Entries are composed of collections of attributes that contain information about the object.



Directories



- An entry example

The screenshot shows the Apache Directory Studio interface. The left pane displays a tree view of the directory structure, with the entry 'cn=PentahoAdmin' selected under 'ou=roles'. The right pane shows the details for this entry, including its DN and a table of attributes.

Attribute Description	Value
objectClass	groupOfUniqueNames (structural)
objectClass	top (abstract)
cn	PentahoAdmin
uniqueMember	cn=Nigel Pond,ou=users,o=mojo
uniqueMember	cn=Test User,ou=users,o=mojo



The LDAP Protocol



- The LDAP protocol (Lightweight Access Directory Protocol) defines a group of messages used by servers and clients from a directory, being a communication protocol.
- It is an open standard that defines a method for accessing and updating directory informations.



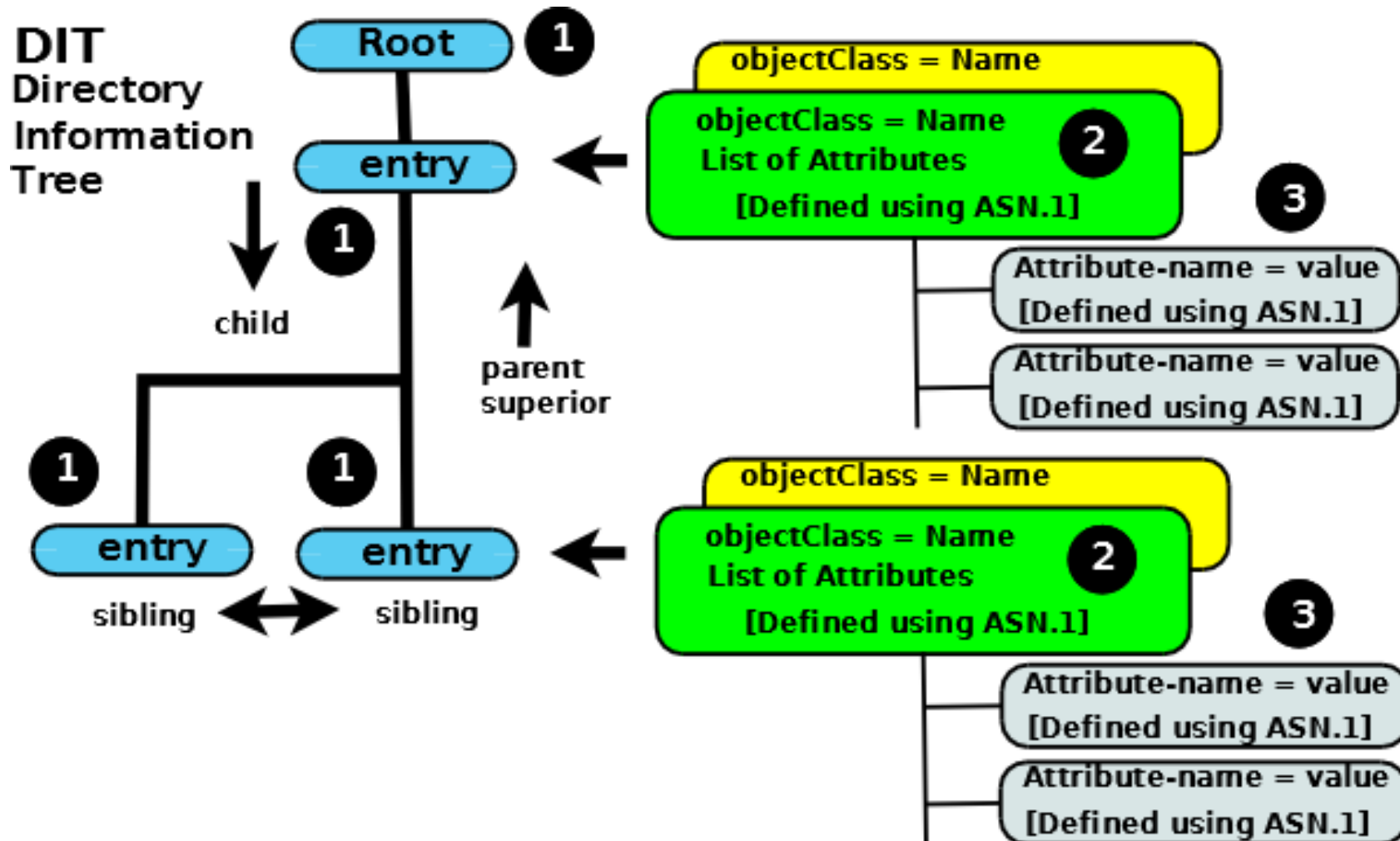
The LDAP Protocol



- The LDAP protocol (Lightweight Access Directory Protocol) defines a group of messages used by servers and clients from a directory, being a communication protocol. It is an open standard that defines a method for accessing and updating directory informations.

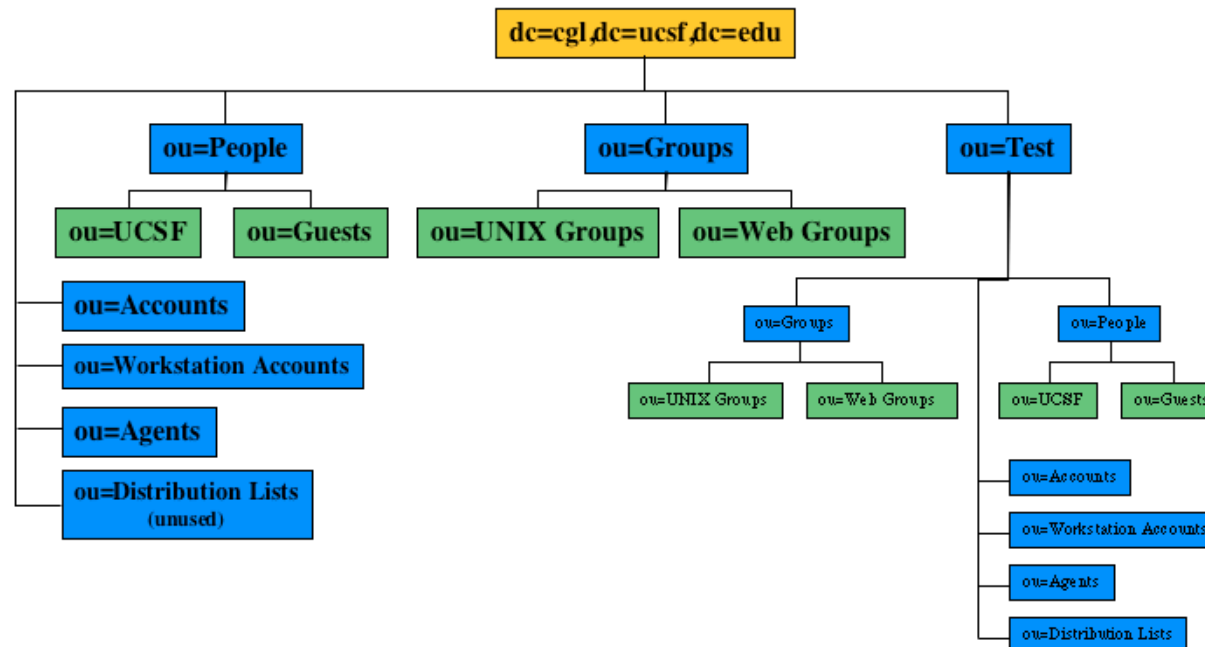


The LDAP Protocol



The LDAP Protocol

- Everything in LDAP is hierarchical.
- Schemas provide the packaging units that group together related objectclasses and attributes



The LDAP Protocol



- Rules
 1. Schemas are simply packaging units
 - All objectclasses and attributes are defined inside schemas
 - All the schemas which include the objectclasses and attributes used in any LDAP implementation must be known to the LDAP server
 - An attribute defined in one schema can be used by an objectclass defined in another schema



The LDAP Protocol



- Rules
 - objectClasses group sets of attributes:
 - objectclasses are defined inside schemas.
 - objectclasses may be organised in a hierarchy in which case they inherit all the properties of their parents or SUPERior in the LDAP jargon.
 - objectclasses may be STRUCTURAL, in which case they can be used to create [entries](#) (data objects), AUXILIARY in which case they may be added into any convenient [entry](#), or ABSTRACT - a non-existent 'thingie'. The most common ABSTRACT objectclass is top, which forms the highest level of every objectclass hierarchy, and terminates any hierarchy.
 - If an objectclass is part of a hierarchy it must be the same type (STRUCTURAL, AUXILIARY) as any SUPERior objectClass. The exception to this rule is if the SUPERior is the top ABSTRACT type which as noted is used to terminate any hierarchy.
 - objectclasses are the means for including attributes (they are attribute containers in the jargon).
 - objectclasses define whether an attribute is mandatory (MUST be present) or optional (MAY be present) within the objectClass.
 - objectclasses are defined using [ASN.1](#) notation.



The LDAP Protocol



- **Rules**

- Attributes typically contain data:

- Every attribute is defined in a schema.
 - Every attribute is included in one or more objectclasses.
 - To use an attribute in an entry, its objectclass must be included in the entry definition and its objectclass must be included in a schema. In turn, the schema must be identified to the LDAP server.
 - An attribute's characteristics are defined using ASN.1 notation.
 - An attribute can appear once in any instance of its containing ObjectClass (SINGLE-VALUE) or can appear more than once in any instance of its containing ObjectClass (MULTI-VALUE). MULTI-VALUE is the default. Thus, for example, it is perfectly reasonable to have more than one instance of an email address (attribute mail) but it would be slightly confusing to have more than one instance of a password (attribute userPassword). Not even your mother could sort that out.
 - An attribute definition may be part of a hierarchy, in which case it inherits all the properties of its parents. For example, commonName (cn), givenName (gn) and surname (sn) are all children of the name attribute. Unlike the objectClass, attribute hierarchies are not terminated with a top equivalent. In the attribute case it is the absence of a SUPERior definition which indicates, surprisingly, that this is the end of the hierarchy.
 - An attribute definition includes its type (or SYNTAX), for example, a string or number, and how it behaves in certain conditions, for instance, whether comparison operations are case-sensitive or case-insensitive using what are called matchingRules (more on this later, much later).



The LDAP Protocol



- Rules
 - entries group sets of objectclasses within a DIT:
 - entries must contain one, and only one, STRUCTURAL objectClass. A STRUCTURAL objectClass may have a SUPERior (may be part of a hierarchy) which is also STRUCTURAL and thus the hierarchy may be viewed as a single STRUCTURAL objectClass.
 - entries may contain any number of AUXILIARY objectClasses.
 - entries can have child entries which appear below them in the address (naming) hierarchy
 - entries can have parent entries which appear above them in the address (naming) hierarchy
 - entries can have sibling entries which appear at the same level as them in the address hierarchy. sibling entries share a common parententry.



LDAP Models



- LDAP Models
 - Describe the information that can be stored in the directory and what can be done with them
- LDAP Schemas
 - Define the structure of an entry in a directory and the attributes that can be inserted in it



LDAP Models



- Information model
 - Describes the structure of information in the LDAP directory
 - Basic units of information are objects called entries
 - Entries are composed of a collection of attributes
 - Entries are arranged in a tree structure called Directory Information Tree (DIT)



LDAP Models

- Information model
 - DIT

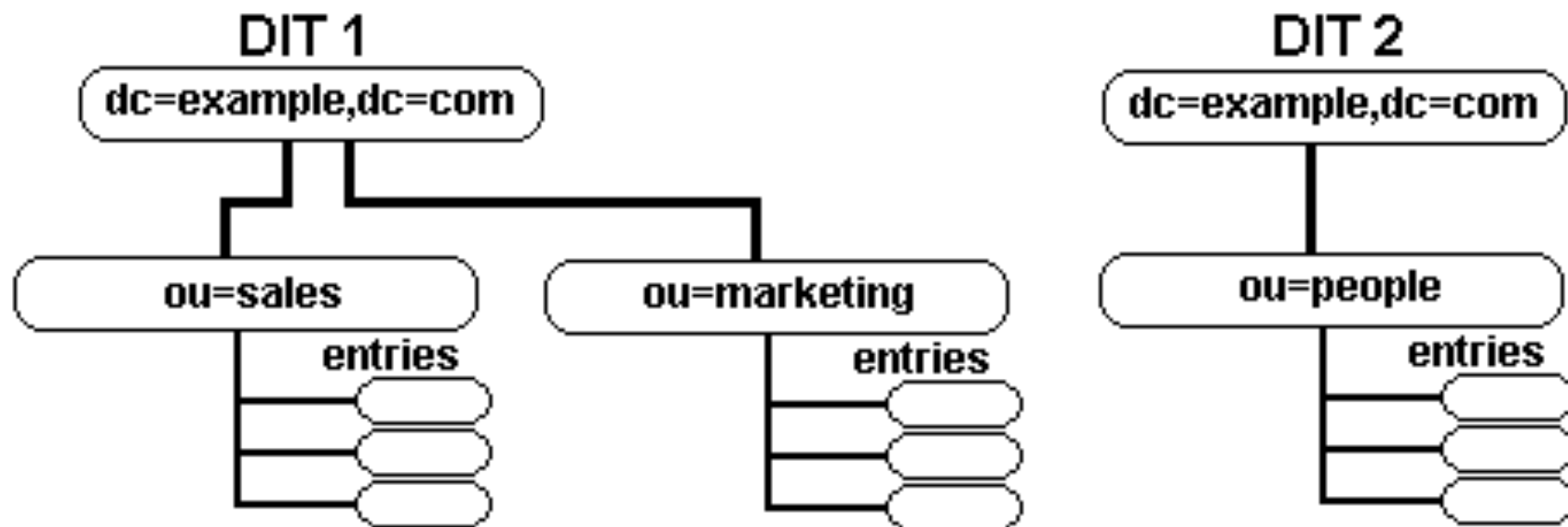


Figura 2.1

LDAP Models



- Information model
 - Entries (Objects)
 - Each entry has a unique name (DN)
 - In general, any input using an abstract class, at least one structural and may have helper classes
 - Have only attributes defined in object classes
 - Object classes
 - Define which attributes are required or optional
 - Can be abstract, structural or auxiliary
 - Can inherit properties from other classes



LDAP Models



- Information model
 - Classes de objects

```
objectclass ( 2.5.6.6 NAME 'person`  
  SUP top STRUCTURAL  
  MUST ( sn $ cn )  
  MAY ( userPassword $ telephoneNumber $  
  seeAlso $ description ) )
```



LDAP Models



- Information model
 - Classes de objects
 - As one more exemple, suppose one class called *person* was defined including one attribute *surname*.
 - The object class *organizationalPerson* could be defined as a subclass of *person*.
 - Tha class *organizationalPerson* would have the same attributes of class *person* and and could add other attributes, like title.
 - The *person* object class could be considered superior of *organizationalPerson* class.



LDAP Models



- Information model
 - Object classes

```
objectclass ( <OID of object class>
  [ "NAME" <name of of object class> ]
  [ "DESC" <object class description> ]
  [ "OBSOLETE" ]
  [ "SUP" <OID of ancestral object class> ]
  [ ( "ABSTRACT" | "STRUTURAL" | "AUXILIARY" ) ]
  [ "MUST" <required attributes> ]
  [ "MAY" <optional attributes> ]
)
```



LDAP Models



- Information model
 - attributes

```
attributetype ( <attribute`s OID>
  [ "NAME" <attribute`s name> ]
  [ "DESC" <attribute`s descriptions> ]
  [ "OBSOLETE" ]
  [ "SUP" <ancestral class` OID> ]
  [ "EQUALITY" <rule of comparison> ]
  [ "ORDERING" < rule of comparison]
  [ "SUBSTR" < rule of comparison]
  [ "SYNTAX" <syntax`s OID> ]
  [ "SINGLE-VALUE" ]
  [ "COLLECTIVE" ]
  [ "NO-USER-MODIFICATION" whsp ]
  [ "USAGE" whsp attributeUsage ] )
```



LDAP Models



- Information model
 - attributes

```
attributetype ( 2.5.4.20 NAME 'telephoneNumber'  
    DESC 'RFC2256: Telephone Number'  
    EQUALITY telephoneNumberMatch  
    SUBSTR telephoneNumberSubstringsMatch  
    SYNTAX  
    1.3.6.1.4.1.1466.115.121.1.50{32} )
```



LDAP Models



- Information model
 - IANA Standard to OIDs
 - Each attribute and object class has a unique identifier OID recorded in the IANA
 - <http://www.iana.org>
 - To create new attributes and object classes you need to order the registration of the institution with the IANA
 - RNP acquired OID 1.3.6.1.4.1.15996, and new attributes and objects can be numbered from it



LDAP Models



- Information model
 - Examples de syntaxes:
 - Boolean: 1.3.6.1.4.1.1466.115.121.1.7
 - DN: 1.3.6.1.4.1.1466.115.121.1.12
 - Char UTF-8: 1.3.6.1.4.1.1466.115.121.1.15
 - Integer: 1.3.6.1.4.1.1466.115.121.1.27
 - Numeric Char: 1.3.6.1.4.1.1466.115.121.1.36
 - Postal Address: 1.3.6.1.4.1.1466.115.121.1.41
 - Audio: 1.3.6.1.4.1.1466.115.121.1.4
 - Certificate: 1.3.6.1.4.1.1466.115.121.1.8
 - JPEG: 1.3.6.1.4.1.1466.115.121.1.28



LDAP Models



- Information model
 - Examples de regras de comparação:

Nome	Tipo	Descrição
<i>BooleanMatch</i>	<i>equality</i>	Boolean
<i>CaseIgnoreMatch</i>	<i>equality</i>	Not case sensitive
<i>CaseIgnoreOrderingMatch</i>	<i>ordering</i>	Not case sensitive
<i>CaseIgnoreSubstringsMatch</i>	<i>substrings</i>	Not case sensitive
<i>CaseExactMatch</i>	<i>equality</i>	Differentiates uppercase and lowercase
<i>NumericStringOrderingMatch</i>	<i>ordering</i>	Numeric
<i>NumericStringMatch</i>	<i>equality</i>	Numeric



LDAP Models



- Model name
 - Entries are named according to their position in the DIT
 - DNs are composed of Relative Distinguished Names (RDN) which has the form:

`<attribute name> = <value>`

- While DNs uniquely identifies an entry in the directory, RDNs do the same within a directory level



LDAP Models



- Model name
 - DN e RDN

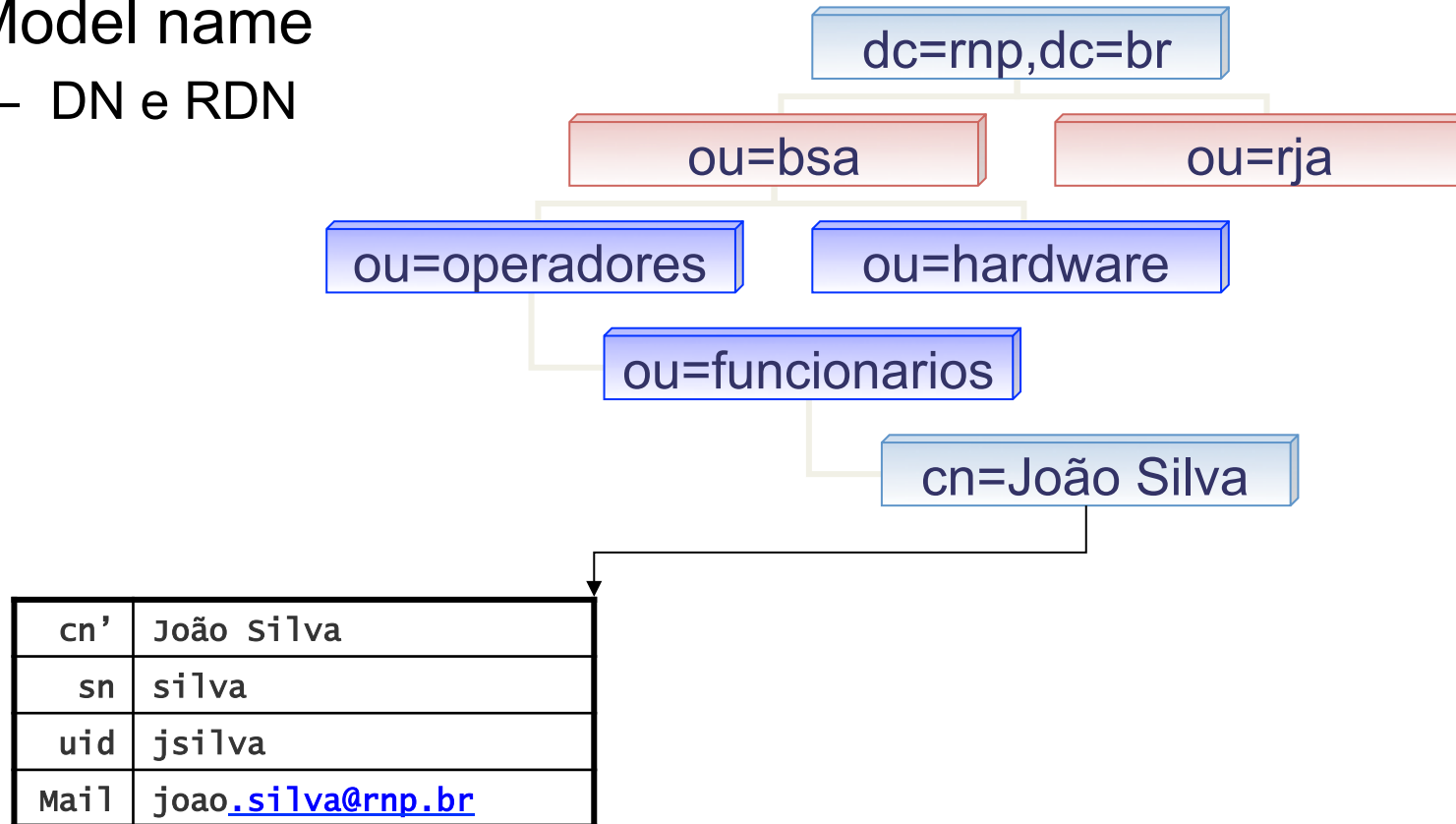


Figura 2.2



LDAP Models



- Model name
 - Representation by strings

`cn=João Silva,ou=funcionarios,ou=operadores,ou=bsa,dc=rnp,dc=br`

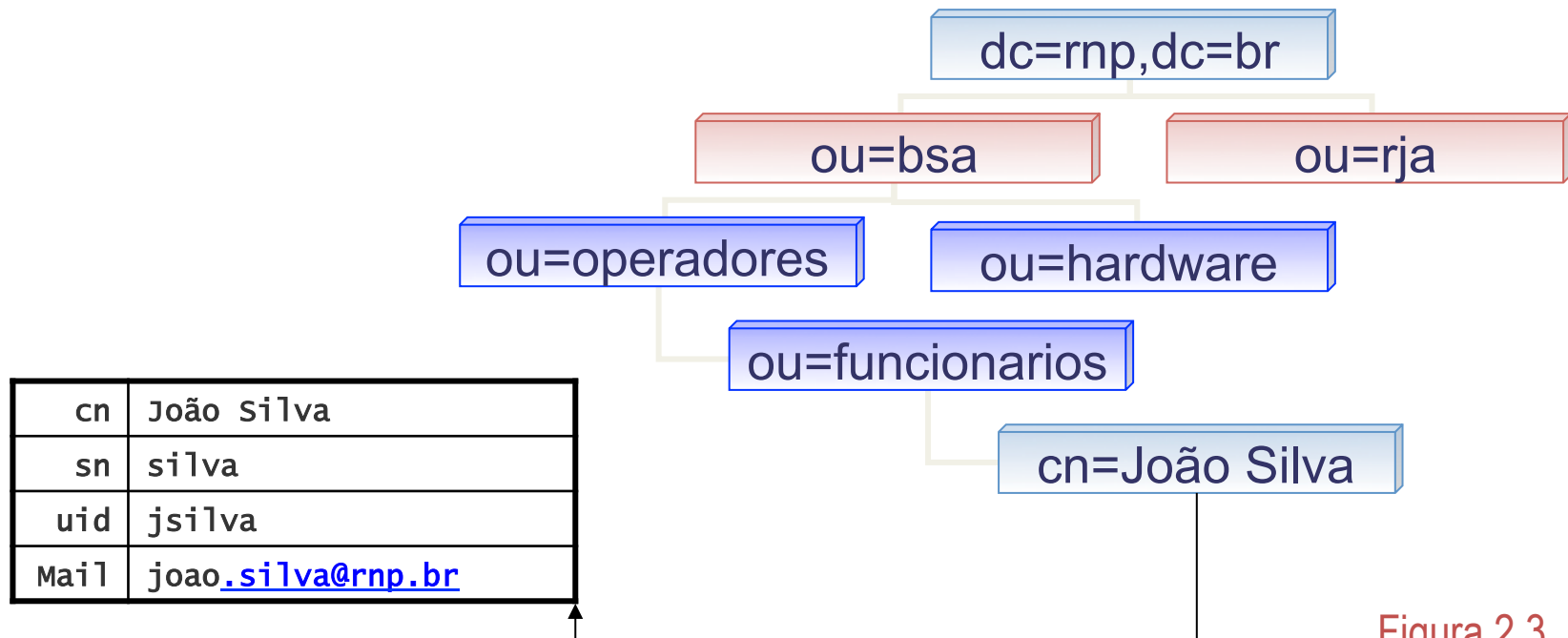


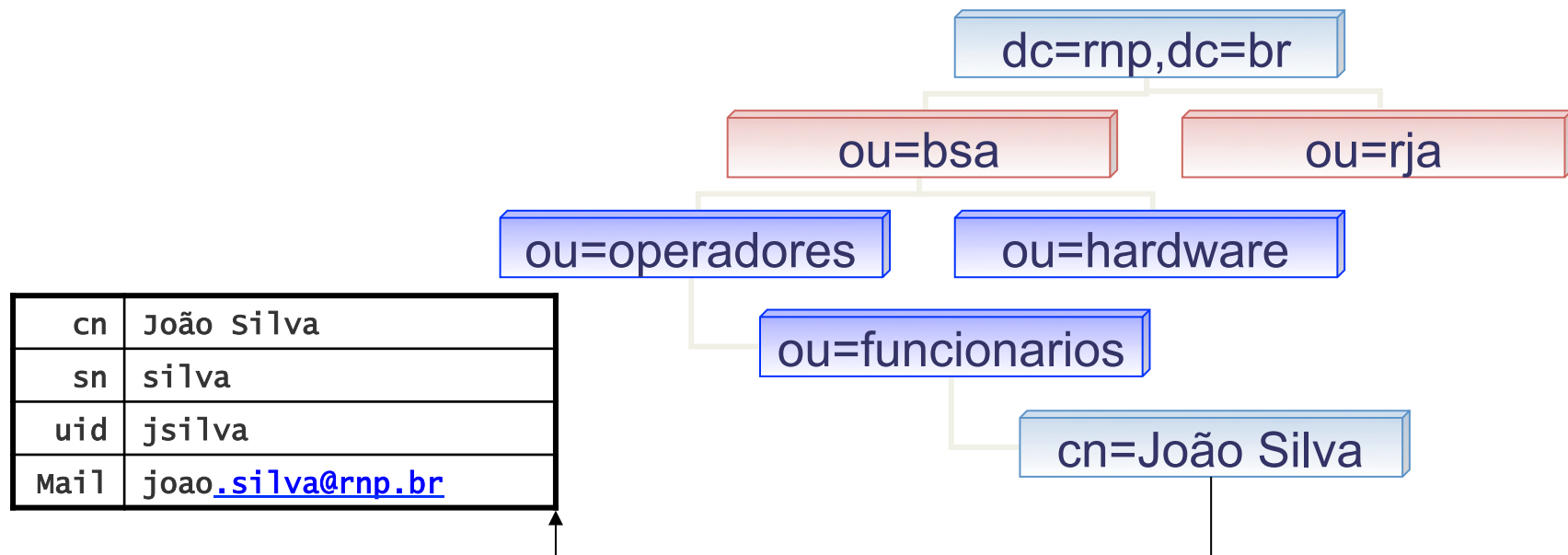
Figura 2.3

LDAP Models



- Name model
 - URL representation

`ldap://server/cn=João Silva,ou=funcionarios,
ou=operadores,ou=bsa,dc=rnp,dc=br?uid`



LDAP Models



- Functional Model
 - Three operation category that should be performed in LDAPv3
 - Authentication:
 - bind
 - unbind
 - abandon
 - Search:
 - search
 - compare
 - Update:
 - add, modify, delete e modifyRDN



LDAP Models



- Functional Model
 - Search
 - Base
 - Scope
 - Search filter
 - Attributes to return
 - Limit



LDAP Models



- Functional Model
 - Search filters
 - attribute operador value

Operador	Exemplo
&	(& (cn=joao) (sn=silva))
	((uid=joao) (uid=silva))
!	(! (uid=joao))
=	gidNumber=100
~=	sn~=silv
>=	uidNumber>=5000
<=	Sn<=silva
*	*

LDAP Schemas



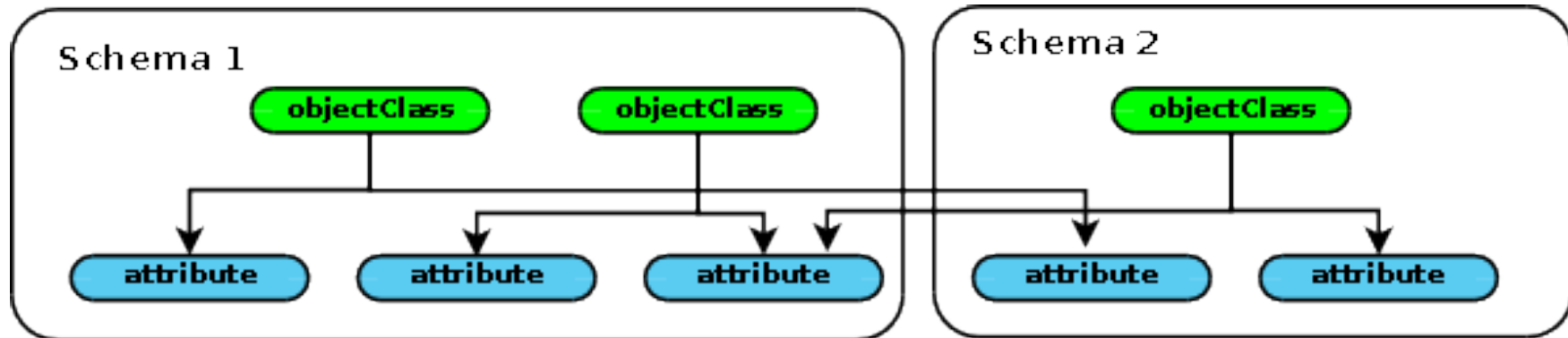
- An LDAP schema is a convenient packaging unit for containing broadly similar objectClasses and attributes
- Every attribute or objectclass, including its superior objectclass or attribute, used in an LDAP implementation must be defined in a **schema**, and that schema must be **known** to the LDAP server
- In OpenLDAP OLC (cn=config) the installed schemas are located under cn=schema, cn=config



LDAP Schemas



- Additional schemas or conversion of a schema format may be done manually (if not large)
- For large files, using the slaptest utility with a couple of manual edits may be the quickest method



LDIF Representation



- LDAP Data Interchange Format
 - Entries group description
 - Description of update sentences



LDIF Representation



- Description group of entries

```
dn: <distinguished name>
```

```
<attrdesc>: <attrvalue>
```

```
<attrdesc>: <attrvalue>
```

```
<attrdesc>:: <base64-encoded-value>
```

```
<attrdesc>:< <URL>
```

```
...
```



LDIF Representation



- Description group of entries

```
dn: cn=Joao Silva  
    ,dc=rnp,dc=br  
objectclass: top  
objectclass: person  
cn: Joao Silva  
sn: Silva  
cn:: IGJlZ2lucyB3aXRoIGEgc3BhY2U=  
cn:< file:///tmp/arquivo
```



LDIF Representation



- Description of update sentences

```
dn: <distinguishedname>
changetype: <[modify|add|delete|modrdn]>
<[modify|add|delete|modrdn]>: <attributetype>
<attrdesc>: <value1>
...
-
<[modify|add|delete|modrdn]>: <attributetype>
<attrdesc>: <value1>
<attrdesc>: <value2>
...
-
```



LDIF Representation



- Description of update sentences

```
dn: cn=Joao Silva,dc=rnp,dc=br
changetype: add
objectclass: person
objectclass: inetorgperson
cn: Joao
cn: Joao Silva
sn: Silva
```

```
dn: cn=Joao Silva,dc=rnp,dc=br
changetype: modify
add: givenName
givenName: jo
givenName: Joao
-
replace: description
description: Funcionario Joao
```



Shell commands and graphical tool



- Principal LDAP commands:
 - Idapadd <options> -f <arquivo LDIF>
 - Add directory entries
 - Idapmodify <options> -f <LDIF file>
 - Modify data in directory, either by modifying entries or adding them
 - Idapdelete <options> <list DNs | -f file>
 - Delete directories entries
 - Idapsearch <options> <search filter>
 - Perform searches in directory according specific criteria



Shell commands and graphical tool



- Shell command examples

```
ldapadd -x -H ldap://servidor.ldap -D  
"cn=admin,dc=curso,dc=ldap" -W -f arquivo.ldif  
ldapmodify -x -D "cn=admin,dc=esr,dc=rnp,dc=br" -W -f  
arquivo.ldif  
ldapsearch -x -D "cn=admin,dc=esr,dc=rnp,dc=br" -W -b  
"dc=curso,dc=ldap" uid=00123456  
ldapdelete -x -D "cn=admin,dc=esr,dc=rnp,dc=br" -W  
"uid=dijkstra,ou=people,dc=esr,dc=rnp,dc=br"
```



Shell commands and graphical tool



- Graphical tool – Apache Directory Studio

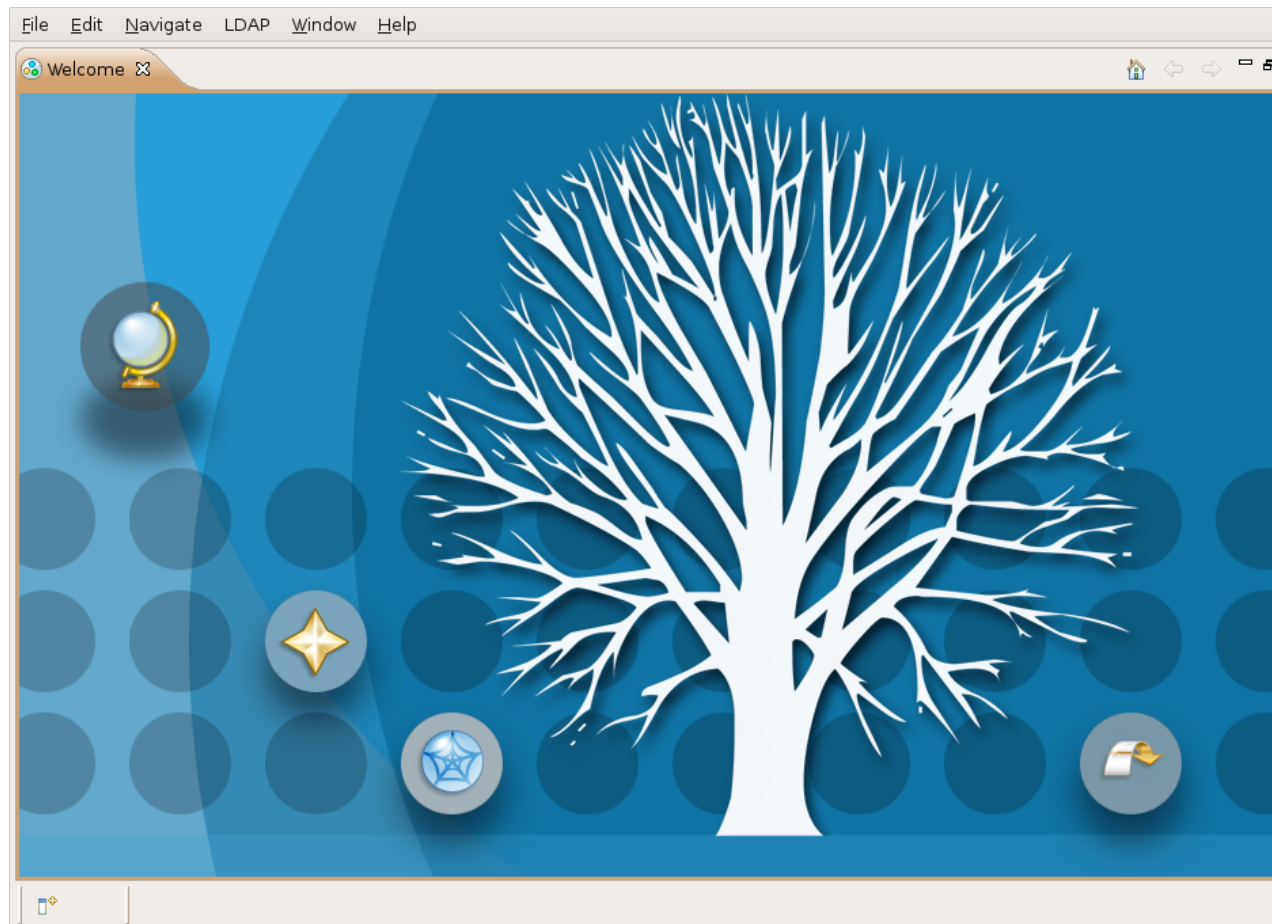
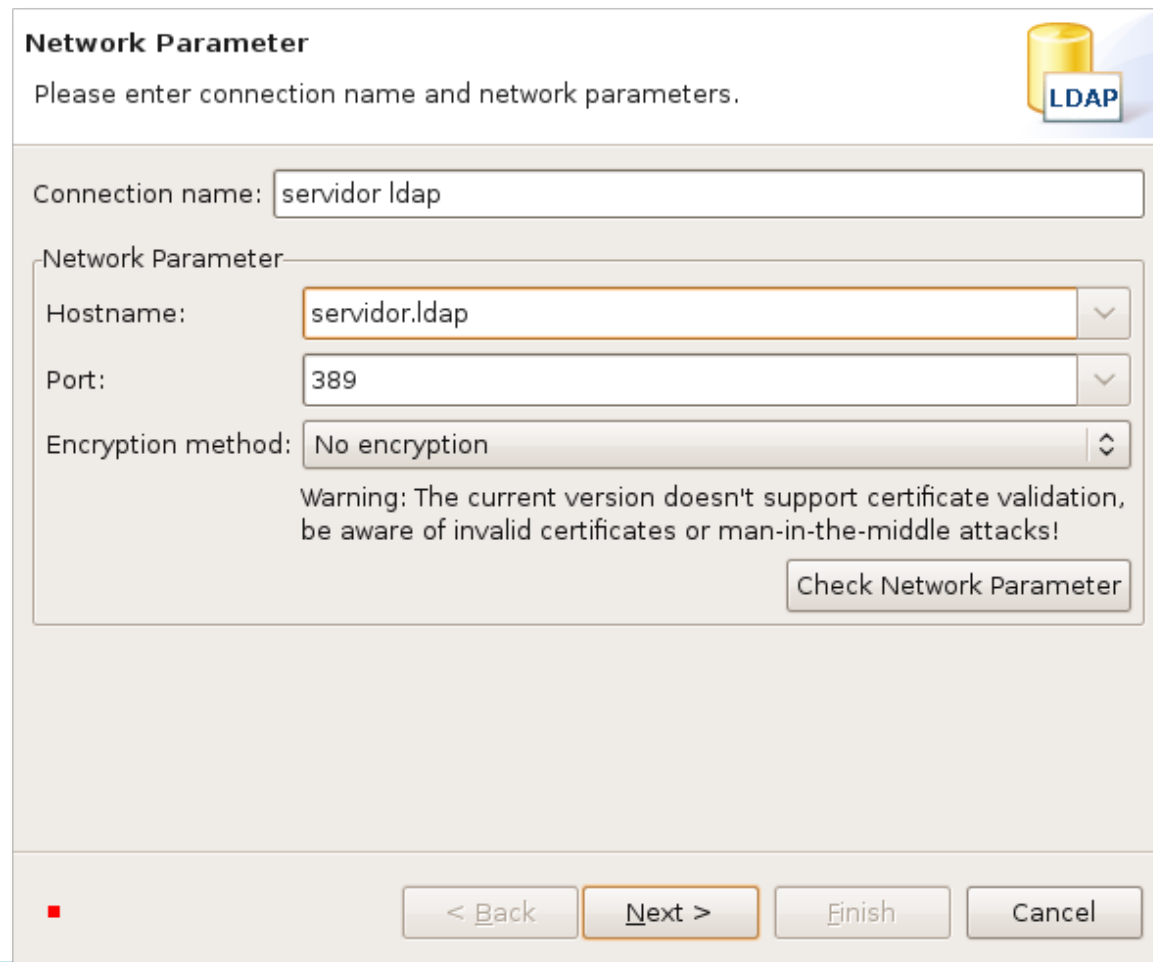


Figura 2.4



Shell commands and graphical tool

- Connection with LDAP server



Network Parameter

Please enter connection name and network parameters.

Connection name:

Network Parameter

Hostname:

Port:

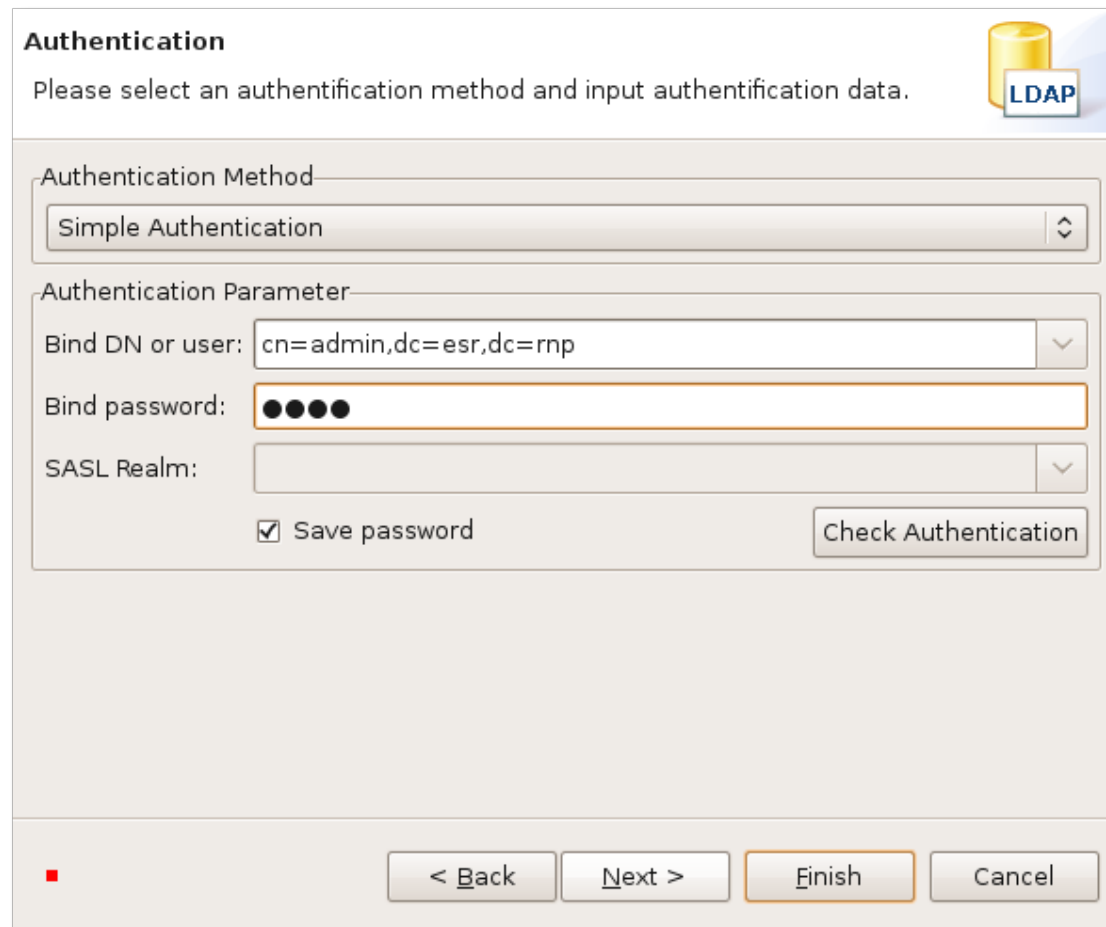
Encryption method:


Warning: The current version doesn't support certificate validation, be aware of invalid certificates or man-in-the-middle attacks!

Figura 2.5

Shell commands and graphical tool

- LDAP database administrator



Authentication 

Please select an authentication method and input authentication data.

Authentication Method: Simple Authentication

Authentication Parameter:

Bind DN or user: cn=admin,dc=esr,dc=rnp

Bind password: ●●●●

SASL Realm:

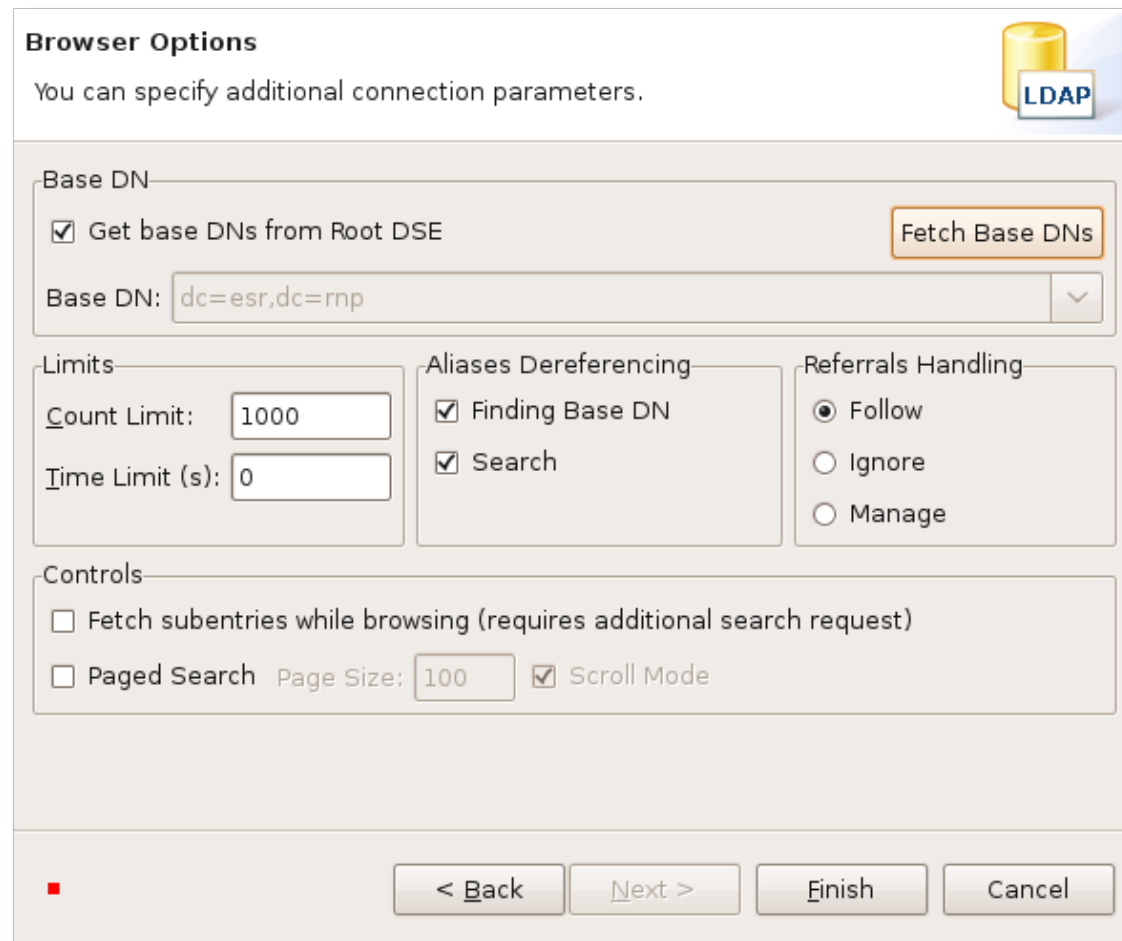
Save password


< Back Next > Finish Cancel

Figura 2.6

Shell commands and graphical tool

- Navigation options in the directory



Browser Options 

You can specify additional connection parameters.

Base DN

Get base DN from Root DSE Fetch Base DN

Base DN:

Limits

Count Limit:

Time Limit (s):

Aliases Dereferencing

Finding Base DN

Search

Referrals Handling

Follow

Ignore

Manage

Controls

Fetch subentries while browsing (requires additional search request)

Paged Search Page Size: Scroll Mode

Figura 2.7

Shell commands and graphical tool



- Main Screen ApacheDS

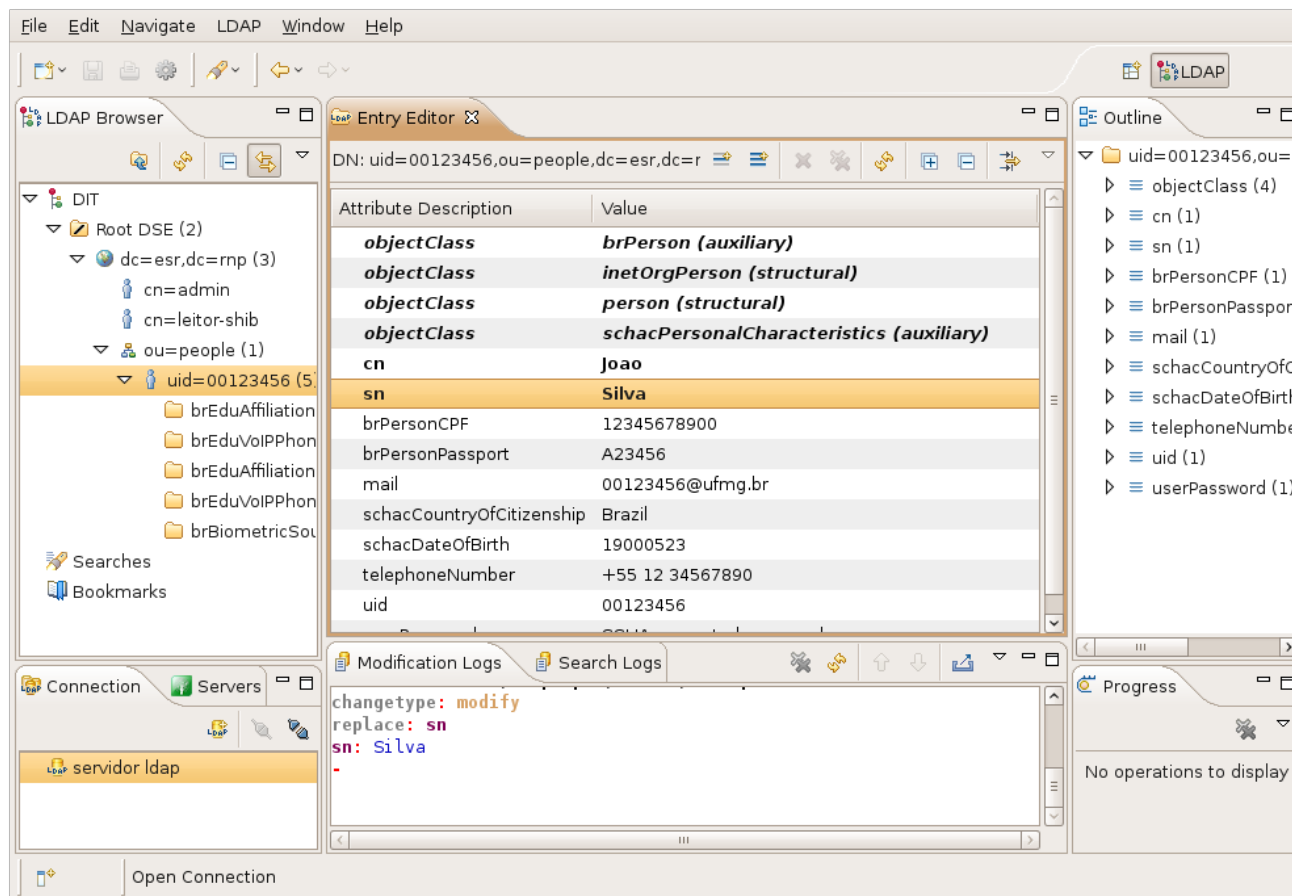


Figura 2.8

OpenLDAP



- Open source implementation of LDAP v3
- Platform independent
- Authentication strong mechanisms SASL
- Data confidentiality and integrity with the use of SSL / TLS
- Internationalization through the use of Unicode
- Guidelines and continued
- Revelation schemes
- Extended controls and operations



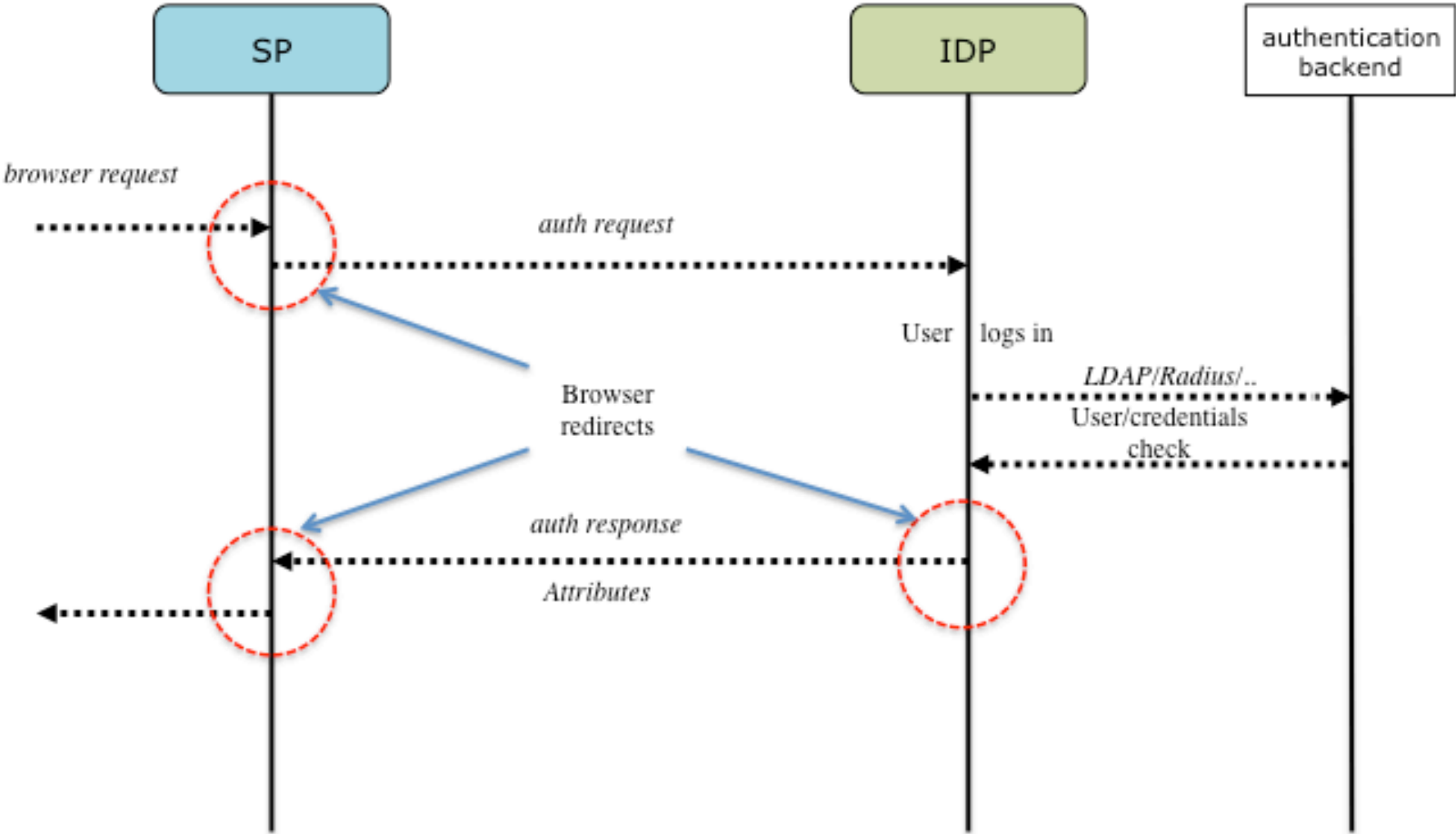
Service Providers



- Implement services that must be available to person linked to institution, and requires:
 - Authentication:
 - Identification of the service users
 - Authorization:
 - Additional user's attributes that ensure access privileges
- Focus on the implementation of the service, not the maintenance of user records



Interaction between the federation elements



Federation Metadata

- An XML document that describes every federation entity
- Contains
 - Unique identifier for each entity known as the entityID
 - Endpoints where each entity can be contacted
 - Certificates used for signing and encrypting data
- May contain
 - Organization and person contact information
 - Information about which attributes an SP wants/needs
- Metadata is usually distributed by a public HTTP URL
 - The metadata should be digitally signed
 - Bilateral metadata exchange scales very badly
- Metadata **must** be kept up to date so that
 - New entities can work with existing ones
 - Old, or revoked, entities are blocked



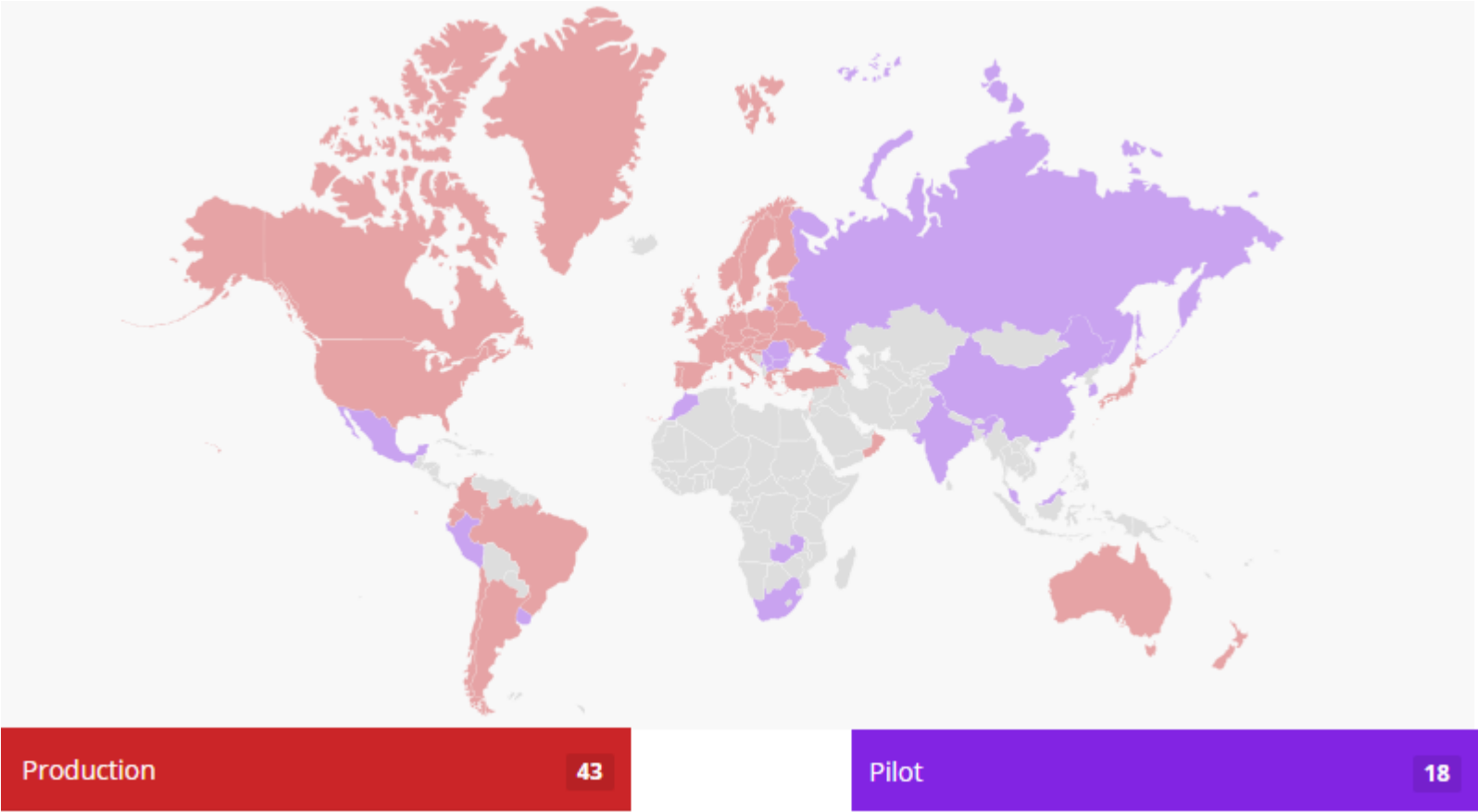
NREN Federations



- CAFE
 - Brazil
- InCommon
 - EUA
- Feide
 - Norway
- Switch
 - Swiss
- SDSS
 - United Kingdom



Federations Map



Schema eduPerson

- eduPerson is a LDAP schema designed to include widely-used person and organizational attributes in higher education
- It was created in the context of the Internet2 project to describe people from the academic community
- The eduPerson assumes only one entry per person, with multi-valued fields describing, for example, the various links of a user with the institution
- This model can not meet the needs of all federations, and may be customized according to each particular case
- The eduPerson object class provides a common list of attributes and definitions, drawing on the existing standards in higher education



Schema eduPerson



- Development of eduPerson was supported with funding from Internet2.
- The Internet2 **MACE-Dir Working Group** releases new versions of the eduPerson specification.



Schema eduPerson



- eduPerson specification intended to support LDAP operations include an auxiliary object class for campus directories designed to facilitate communication among higher education institutions
- It consists of a set of data elements or attributes about individuals within higher education, along with recommendations on the syntax and semantics of the data that may be assigned to those attributes



Schema eduPerson



- It is recommended that person entries have the person, organizationalPerson and inetOrgPerson object classes defined
- Based in part on RFC4512 and RFC4519
- EduPerson attributes would be brought into the person entry as appropriate from the auxiliary eduPerson object class



Schema eduPerson



- The most common and useful personal attributes are identifiers
- Identifier is an information element that is specifically designed to distinguish each entry from its peers in a particular set
- While almost any information in an entry may contribute to differentiating it from similar entries, identifiers are intentionally designed to do this



Schema eduPerson



- Scope
 - The eduPersonPrincipalName, eduPersonPrincipalNamePrior, eduPersonScopedAffiliation, and eduPersonUniqueid attribute definitions found below make use of the concept of scope
 - The meaning of scope is specific to the attribute to which it is attached and can vary from one attribute to another.



Schema eduPerson

- **eduPerson Object Class Definition**

- All eduPerson-defined attribute names are prefaced with "eduPerson." The eduPerson auxiliary object class contains all of them as "MAY" attributes:

- (1.3.6.1.4.1.5923.1.1.2
NAME 'eduPerson'
AUXILIARY
MAY (eduPersonAffiliation \$
eduPersonNickname \$
eduPersonOrgDN \$
eduPersonOrgUnitDN \$
eduPersonPrimaryAffiliation \$
eduPersonPrincipalName \$
eduPersonEntitlement \$
eduPersonPrimaryOrgUnitDN \$
eduPersonScopedAffiliation \$
eduPersonTargetedID \$
eduPersonAssurance \$
eduPersonPrincipalNamePrior \$
eduPersonUniqueid)



Scheme Personalization - brEduPerson



- Schema classes brEduPerson
 - Scheme proposed for members of higher education institutions in Brazil
 - It is divided into:
 - General informations about any citizen
 - General informations about members of an institution
 - Specific informations about employees and students
- Modeled relationships in hierarchical structure



Scheme Personalization - brEduPerson



- Objects classes and attributes
 - brPerson
 - brPersonCPF, brPersonPassport
 - brEduPerson
 - brEduAffiliationType, brEntranceDate, brExitDate, brEduAffiliation
 - brBiometricData
 - brCaptureDate, brBiometricSource, brBiometricData



Scheme Personalization - brEduPerson



- Object classes and attributes
 - brEduVoIP
 - brEduVoIPalias
 - brEduVoIPtype
 - brEduVoIPadmin
 - brEduVoIPcallforward
 - brEduVoIPaddress
 - brEduVoIPexpiryDate
 - brEduVoIPbalance
 - brEduVoIPcredit
 - brEduVoIPphone



Name model for use in the CAFe Federation



- Need to reflect in the database the fact that the same person play different roles within the institution or has more than one VoIP number, each with its own characteristics, or store biometric data from different sources
- Examples:
 - The same student in more than one course, with date of entry and different course
 - A teacher performing different functions in certain periods
 - Course Coordination
 - Unit Director



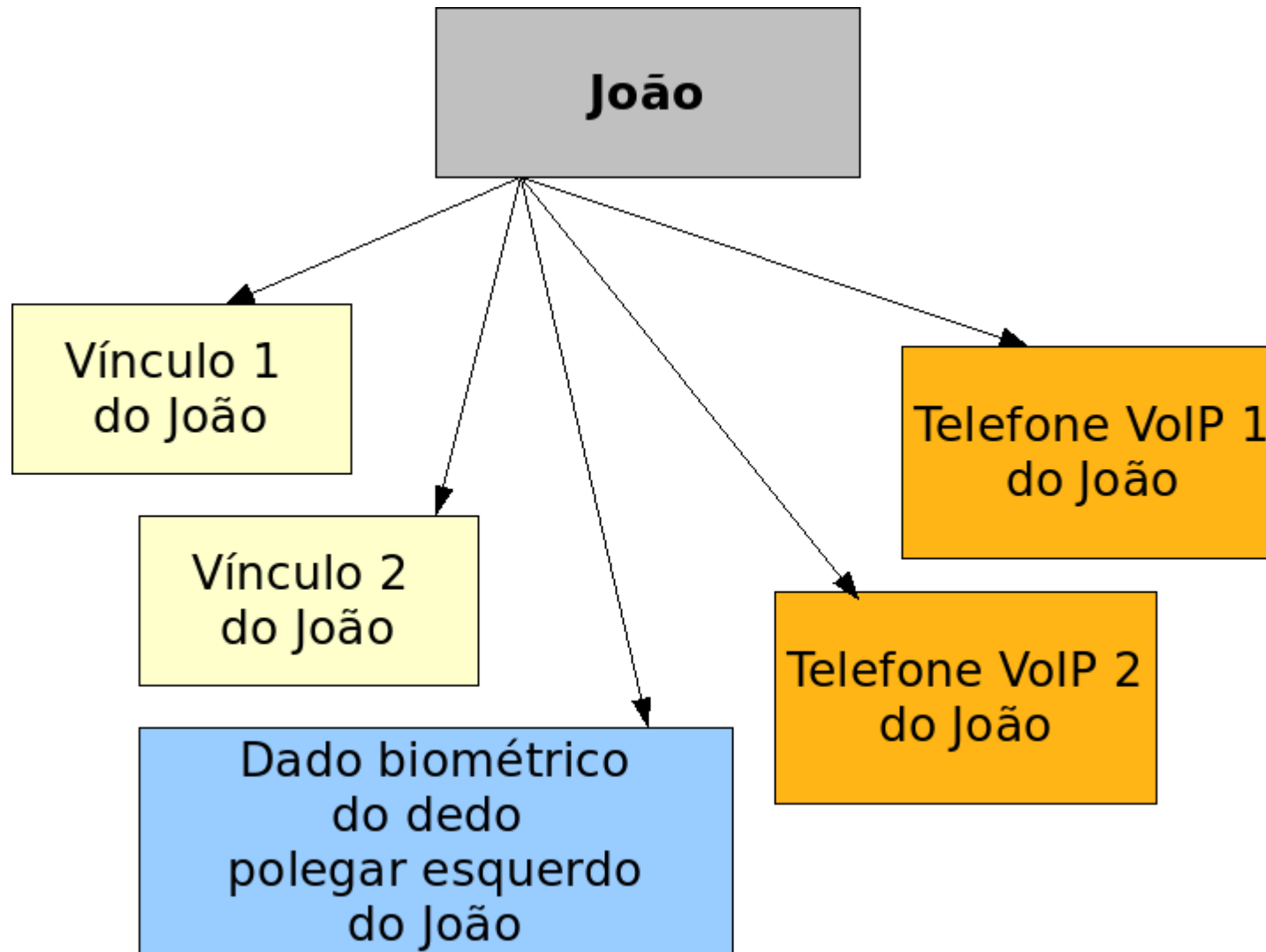
Name model for use in the CAFe Federation



- Proposed Model:
 - The main item - person of an institution - will be treated as a container below which we will appear with related information distinct ties to the institution. examples:
 - Teacher, student, employee
 - VoIP phones
 - biometric sources



Name model for use in the CAFederation



Name model for use in the CAFederation



- Entry example:

```
dn: uid=silvana,ou=people,dc=uff,dc=br
objectClass: person
objectClass: inetOrgPerson
objectClass: brPerson
objectClass: schacPersonalCharacteristics
uid: silvana
brcpf: 12345678900
brpassport: A23456
schacCountryOfCitizenship: Brazil
telephoneNumber: +55 22 81389199
cn: Silvana
userPassword: *****
```



Name model for use in the CAFe Federation



- Entry example

```
dn: braff=1,uid=silvana,ou=people,dc=uff,dc=br
objectclass: brEduPerson
braff: 1
brafftype: faculty
brEntranceDate: 20070205
dn:braff=2,uid=silvana,ou=people,dc=uff,dc=br
objectclass: brEduPerson
braff: 2
brafftype: student
brEntranceDate: 20070205
brExitDate: 20080330
```



Name model for use in the CAFederation



- Entry example

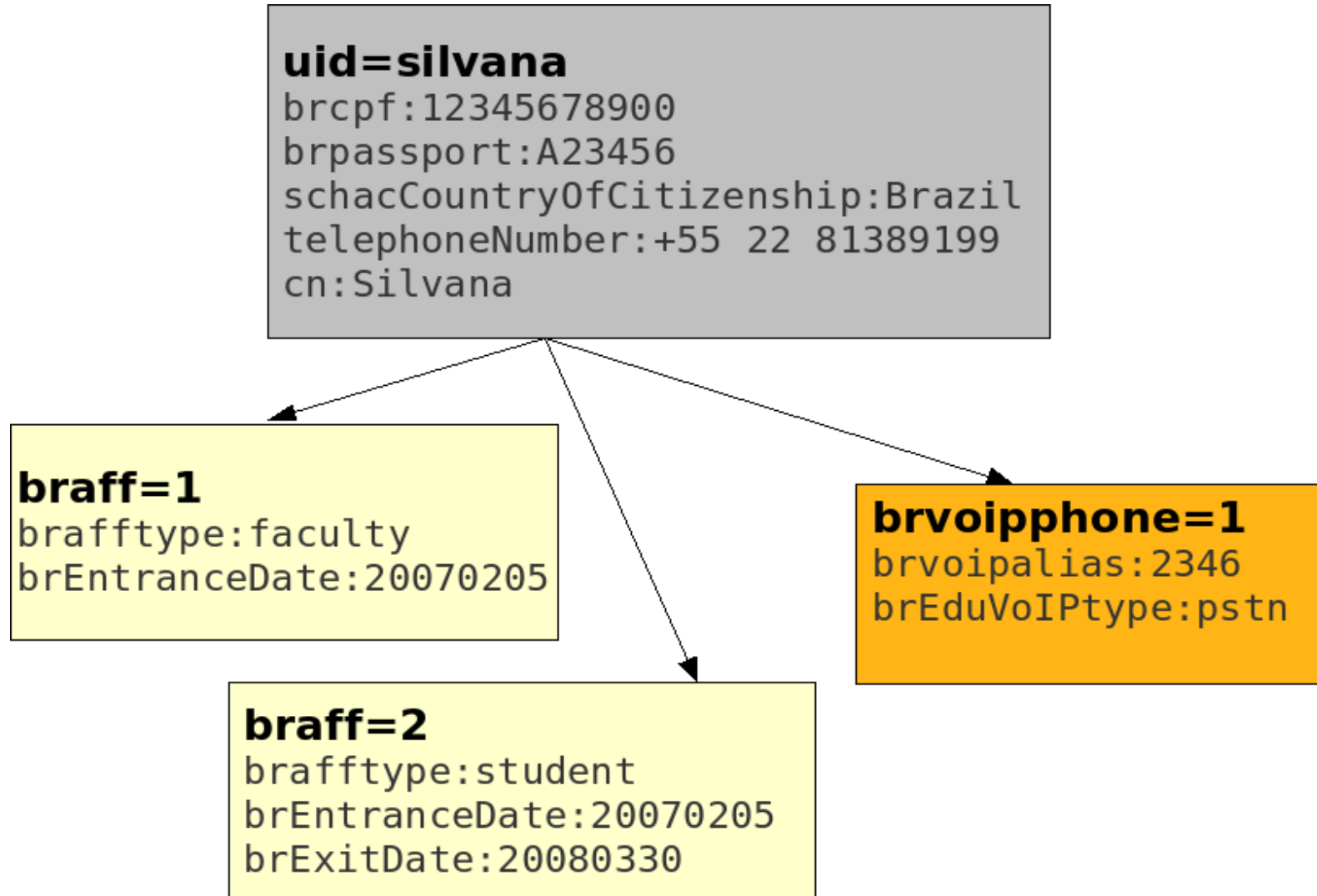
```
dn:brvoipphone=1,uid=silvana,ou=people,dc=uff,dc=br
objectclass: brEduVoIP
brvoipphone: 1
brvoipalias: 2346
brEduVoIPtype: pstn
brEduVoIPadmin:uid=admin,ou=people,dc=uff,dc=br
```



Name model for use in the CAFe Federation



Magic
Middleware for collaborative Applications
and Global virtual Communities



Quiz Time



Magic

Middleware for collaborative Applications
and Global virtual Communities



Quiz Time



1. Which of the following statements are true in Federated Identity Management?

- a) Only the IdP holds the user credentials
- b) Federations route credentials to SPs
- c) Per service credentials are held in applications
- d) The SP needs all information about a user to be released

2. What is eduPersonScopedAffiliation?

- a) A way to describe your role within an organisation.
- b) Your name.
- c) A description of what you can access.
- d) A number assigned to you.



Quiz Time



3. **What are the motivation to implement a federated authentication and authorization infrastructure?**
4. **A federation can personalize the eduPerson schema?**
 - a) Yes
 - b) No
5. **Which of the following is NOT a directory information tree component?**
 - a) Entry
 - b) Object Class
 - c) Attribute
 - d) LDAP



Sources



- **Basic Federation / AAI Training Module**
 - Part1-AAI-Fundamentals (<https://wiki.geant.org/pages/viewpage.action?pageId=50399132>)
 - LDAP protocol (<https://tools.ietf.org/html/rfc4511#page-3>)

