Directory Schema Registry Concept and Implementation Progress

TERENA Network Conference and
CARNet Users´ Conference
Zagreb, 19th May 2003

Peter Gietz, DAASI International GmbH
Peter.gietz@daasi.de

DAASI
International
Directory Applications
for Advanced Security
and Information Management

AGENDA

- Motivation
- Project Plan
- Survey of previous work on directory schema registry related technologies
- Existing LDAP schema
- Incorporation and usage policy
- Metadata format and DIT structure
- Software design
- Implemention progress
- Business Model

DAASI International

Directory Applications
for Advanced Security
and Information Management

Motivation

- Common schema (attributes and objec classes) are vital for directory interoperability
- There are a lot of standards allready out there, people may not know about
- There are even more good schema proposals not (yet) standardized
- People still tend to reinvent he wheel
- You can find information on the web but at different places
- Applications cannot retrieve schema information via LDAP

Project aims

- to set up a LDAP schema registry with
  - an easy browsable and searchable Web interface
  - an LDAP interface for retrieval
  - an interface based on MIME types defined in RFC 2927 for submissions of new schema
- to define a policy defining the standards for inclusion into the registry
- to search for all schema definitions made within the IETF and include them into the registry
- to develop a business model to keep the registry alive after the end of the project.

Project Funding body

- TERENA
  - (Trans-European-Research and Education Networkinc Association)
- JISC
  - (Joint Information Systems Committee, UK)
- REDIRIS
  - (Spanish National Research Network)
- CESNET
  - (Czech National Research Network)
- POZMAN SUPERCOMPUTING
  - (Poznan Supercomputing and Networking Center, Poland)
- DAASI International

DAASI International

Directory Applications
for Advanced Security
and Information Management

Project Documentation

- Project Proposal
- Deliverable B: Survey of previous work on directory schema registry related technologies and existing LDAP schema, version 0.91
- Deliverable B-2: Bibliography for the Directory Schema Registry Project, version 0.91
- Deliverable D: Definition of an incorporation and usage policy for a Directory Schema Registry, version, version 0.9
- Deliverable C: Definition of a metadata format and DIT structure, version 0.9
- Deliverable E: Software Spec (coming soon)

What is out there already

- The subschema mechanism defined in X.500
- The alternative mechanism of RFC 1804
- IANA procedures for registering LDAP elements
- The proposal of the IETF Schema Working Group
- LDAP Schema Viewer at http://ldap.akbkhome.com/
- Novell schema registry
- Object Identifier Registry of Harald Alvestrand at www.alvestrand.no//objectid
- The Object identifier tree of ASN.1.Information site at http://asn1.elibel.tm.fr/en/index.htm
- XML.org registry at http://www.xml.org/xml/registry.jsp
- Some more on Metadata and RDFS

Work that could be used

- IETF WG schema
  - provided specifications for a schema listing service for the directory technologies LDAP, Whois, Whois++ and Rwhois.
  - The idea was to provide a single point of discovery, to promote reuse, reduce duplication of effort and to promote interoperability.
  - This work is based on a document [RFC 2425] that defines a MIME Content-Type for holding directory information.

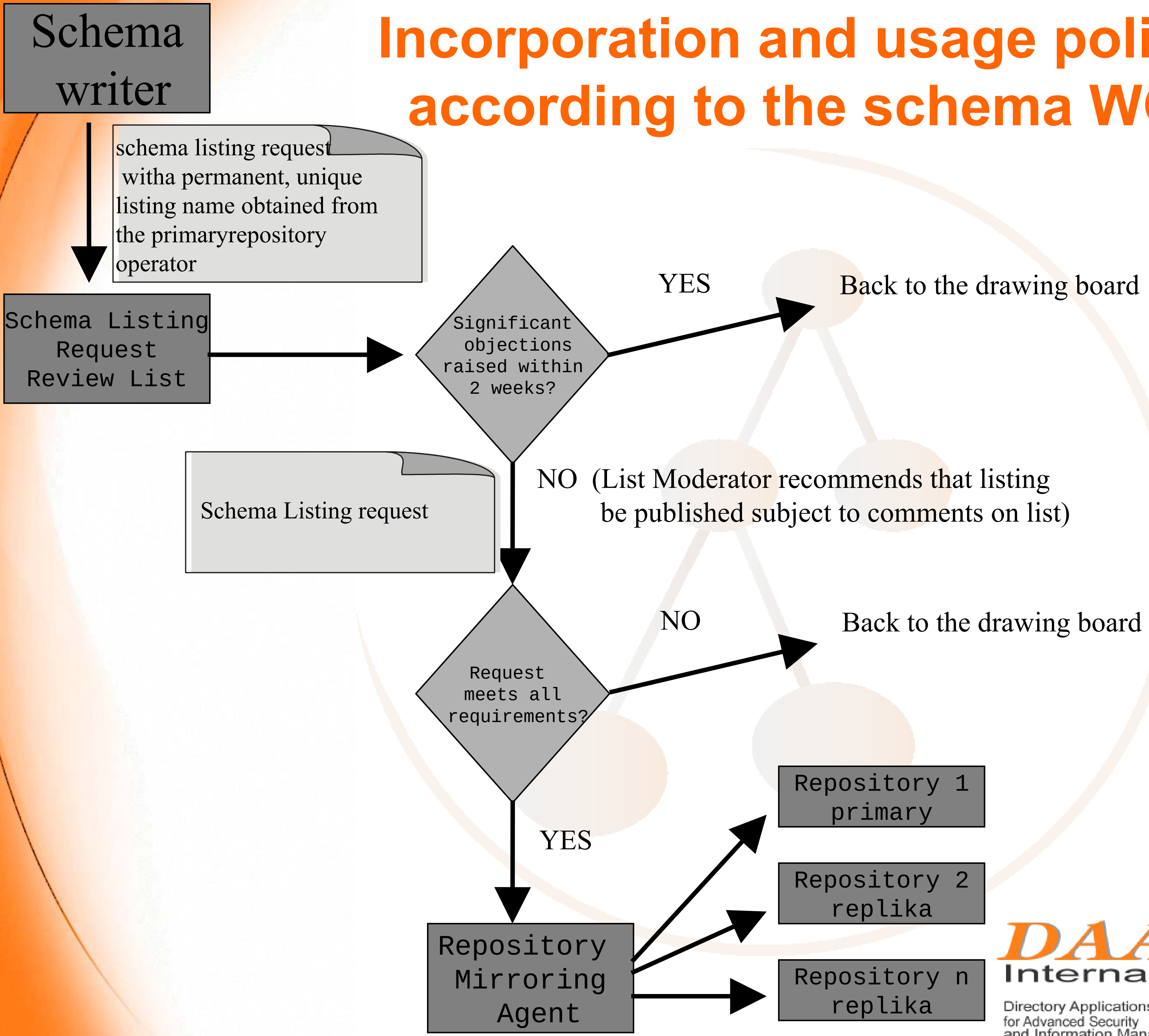
Schema WG docs

- Apple, C., "Directory Schema Listing File Names", <draft-ietf-schema-file-list-01.txt>, April 1998 (expired), http://www.watersprings.org/pub/id/draft-ietf-schema-file-list-01.txt
- Apple, C., "Directory Schema Listing Meta Data", <draft-ietf-schema-mime-metadata-01.txt>, April 1998, (expired), http://www.watersprings.org/pub/id/draft-ietf-schema-mime-metadata-01.txt
- Apple, C., "Directory Schema Listing Procedures", <draft-ietf-schema-proc-list-01.txt>, April 1998 (expired), http://www.watersprings.org/pub/id/draft-ietf-schema-proc-list-01.txt
- Apple, C., "Requirements for the Initial Release of a Directory Schema Listing Service", <draft-ietf-schema-rqmts-list-01.txt>, April 1998 (expired), http://www.watersprings.org/pub/id/draft-ietf-schema-rqmts-list-01.txt

Existing X.500/LDAP schema

- X.500 schema standards ([X.520] and [X.521])
- IETF LDAP schema standards (27 RFCs from RFC 1274 to RFC 3296)
- DMTF CIM LDAP
- Open Group LDAP DCE
- Internet 2/EDUCAUSE EduPerson, eduOrg
- Proprietary schema from Novell, Netscape, SUN, Microsoft
- LDAP schema of research projects
- LDAP Schema for Grid Computing (Globus Toolkit)
- LDAP schema of Open Source Projects

Incorporation and usage policy according to the schema WG



Schema writer

schema listing request witha permanent, unique listing name obtained from the primaryrepository operator

Schema Listing Request Review List

Significant objections raised within 2 weeks?

YES

Back to the drawing board

Schema Listing request

NO (List Moderator recommends that listing be published subject to comments on list)

Request meets all requirements?

NO

Back to the drawing board

YES

Repository Mirroring Agent

Repository 1 primary

Repository 2 replika

Repository n replika

DAASI International

Directory Applications for Advanced Security and Information Management

Policy of the
Direcory Schema Registry (DSR)

- Establishment of an open list for discussion about schema inclusion
- Specification of a moderator who interacts with the DSR operator
- Specification of the Policy Board
- Specification of the syntactical requirements for schema submission: formats, encoding, naming, process for checking syntax and OID.
- Specification of semantic requirements for schema submission defining a mandatory minimal set of metadata for single schema elements and a whole schema, bibliographical data and additional information on author and contact person
- Specification of a version control
- Specification of the registration process
- Specification of the comment mechanism
- Specification of the update process
- Specification of the actions and responsibilities of the DSR operator

DSR Policy Board

- finalise the decisions about the processes
- control the whole process
- appoint experts for review
- reassign responsibility for a schema
- moderate the discussion list
- decide about schema inclusion and classification of its status

DSR Operator

- Provide and run the technical infrastructure for operating the DSR (hardware, LAP-Server, Webgateway, Mailinglist)
- Provide OIDs and additional numbers for uniquely identifying schema submissions, including versioning
- Perform the specified schema checks.
- Forward schema registration requests to the policy board.
- Include schema according to the instructions of the policy board.
- Provide technical advice to the policy board.
- Contribute to the dissemination of project results and to Public Relations of the DSR.
- Act as a communication mediator between different interest groups.

What info will be stored

- Metadata on specification document
- LDAP compliant definitions of the schema elements
- Single parts of schema element definitions, e.g., MUST attributes in Object Classes
- Metadata as specified by the IETF WG schema
- Separate OID tree
- Additional metadata

LDAP Schema specified

- Metadata for bibliographical references
  - The Dublin Core Metadata set and its LDAP representation
  - Additional schema for person information
  - The front matter elements of RFC 2629
- Metadata specified by the IETF schema WG
  - MIME types for schema metadata and their LDAP representation (draft-ietf-schema-mime-metadata-01.txt)
  - MIME types for LDAP schema elements and their LDAP representation (RFC 2927)
- Additional schema for the DSR
  - Schema for additional schema elements not specified in RFC 2927
  - Schema for storing an OID tree
  - Schema for storing the single parts of schema element definitions
  - Schema for additional metadata

Dublin Core Metadata specification documents

```
objectclass ( 1.3.6.1.4.1.10126.1.7.4.1
    NAME 'dcContainerObject'
    DESC 'object containing the Dublin Core attributes'
    SUP top
    AUXILIARY
    MAY ( dcTitle $ dcCreator $ dcCreatorPointer $ dcSubject $
          dcDescription $ dcPublisher $ dcPublisherPointer $
          dcContributor $ dcContributorPointer $ dcDate $ dcType $
          dcFormat $ dcIdentifier $ dcSource $ dcLanguage $
          dcRelation $ dcCoverage $ dcRights $ dcAudience ) )
```

```
( 2.16.840.1.113730.3.2.2
    NAME 'dcCreatorInfoContentRule'
    DESC 'Profile for the use of object class inetOrgPerson to
          describe an author of a specification document'
    AUX dcPersonObject
    MUST ( cn $ sn )
    MAY ( c $ initials $ o $ street $ l $ st $ postalCode $
          telephoneNumber $ mail $ labeledURI $ displayName $
          givenName $ preferredLanguage )
```

Metadata of the schema WG

```
objectclass ( 1.3.6.1.4.1.10126.1.8.4.2
    NAME 'srSchemaPakMetadataObject'
    DESC 'object containing metadata on a schema according
          to the IETF schema WG specifications'
    SUP top
    AUXILIARY
    MUST ( srListingName $ srListingTitle $ srListingUse $
           srSchemaUnitSpecFiles $ srContactPersonPointer $
           srAuthContactPointer $ srListingSecurityNote $
           srSchemaUnitSpecURL $ srListingCreatedTime $ srPakMemberURI )
    MAY ( srRelatedToListing $ srMoreInfoURI $ srListingComment ) )
```

```
objectclass ( 1.3.6.1.4.1.10126.1.8.4.3
    NAME 'srSchemaPerson'
    DESC 'object containing metadata on a schema according
          to the IETF schema WG specifications'
    SUP top
    STRUCTURAL
    MUST ( srContactName $ srContactLanguage $ srContactEmail $
           srContactPhone $ srContactAddress )
    MAY ( description $ title ) )
```

DAASI International

Directory Applications
for Advanced Security
and Information Management

LDAP schema elements of RFC 2927 and their LDAP representation

```
( 1.3.6.1.4.1.1466.101.120.17
    NAME 'ldapSchemas'
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.56
    USAGE directoryOperation )

( 1.3.6.1.4.1.1466.115.121.1.56
    DESC 'LDAP Schema Definition' )
```

Values in this syntax are represented according to the following BNF:

```
    LdapSchema = "(" whsp
    numericoid whsp
    [ "NAME" qdescrs ]
    [ "OBSOLETE" whsp ]
    [ "IMPORTS" oids ]
    [ "CLASSES" oids ]
    [ "ATTRIBUTES" oids ]
    [ "MATCHING-RULES" oids ]
    [ "SYNTAXES" oids ]
    whsp ")"
```

```
attributetype ( 1.3.6.1.4.1.10126.1.11.3.1
    NAME 'srLdapSchemas'
    DESC 'user attribute for storing ldap schema definitions'
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.56
    SINGLE-VALUE )
```

RFC 2927 contd.

```
attributetype ( 1.3.6.1.4.1.10126.1.11.3.2
    NAME 'srLdapAttributeTypes'
    DESC 'user attribute for storing ldap attribute type definitions'
    EQUALITY objectIdentifierFirstComponentMatch
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.3 )
```

```
AttributeTypeDescription = "(" whsp
    numericoid whsp                ; AttributeType identifier
    [ "NAME" qdescrs ]             ; name used in AttributeType
    [ "DESC" qdstring ]            ; description
    [ "OBSOLETE" whsp ]
    [ "SUP" woid ]                 ; derived from this other
                                   ; AttributeType

    [ "EQUALITY" woid              ; Matching Rule name
    [ "ORDERING" woid              ; Matching Rule name
    [ "SUBSTR" woid ]              ; Matching Rule name
    [ "SYNTAX" whsp noidlen whsp ] ; see section 4.3
    [ "SINGLE-VALUE" whsp ]        ; default multi-valued
    [ "COLLECTIVE" whsp ]          ; default not collective
    [ "NO-USER-MODIFICATION" whsp ]; default user modifiable
    [ "USAGE" whsp AttributeUsage ]; default userApplications
    whsp ")"
```

Schema for storing schema element definitions

```
attributetype ( 1.3.6.1.4.1.10126.1.13.3.9
    NAME 'SRldapSyntaxPointer'
    DESC 'DN Pointer to an entry containing information about an
          LDAP syntax'
    EQUALITY distinguishedNameMatch
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
    SINGLE-VALUE )

objectclass ( 1.3.6.1.4.1.10126.1.13.4.1
    NAME 'srLdapSchemaObject'
    DESC 'LDAP Schema definition'
    STRUCTURAL
    MUST ( cn $ srLdapSchemas )
    MAY ( srLdapObsolete $ srLdapSyntaxPointer $
          srLdapSchemaDescriptor $ srLdapSchemaImports $
          srLdapOcPointer $ srLdapAttributeTypePointer $
          srLdapMatchingRulePointer )
objectclass ( 1.3.6.1.4.1.10126.1.13.4.2
    NAME 'srLdapElementObject'
    DESC 'toplevel objectclass of each LDAP schema object'
    ABSTRACT
    MAY ( srLdapElementDescriptor $ srLdapElementRdnDescriptor $
          srLdapElementDescription $ srLdapObsolete $ cn ) )
```

Schema for storing schema element definitions contd.

```
objectclass ( 1.3.6.1.4.1.10126.1.13.4.3
    NAME 'srLdapAttributeType'
    DESC 'LDAP Attribute Type definition'
    SUP  srLdapElementObject
    STRUCTURAL
    MUST ( srNumericOid $ srLdapAttributeTypes
    MAY  ( srLdapSup $
           srLdapEqualityMatchingRulePointer $
           srLdapSubstrMatchingRulePointer $
           srLdapOrderingMatchingRulePointer $
           srLdapSyntaxPointer $
           srLdapAttributeTypeUpperBound $
           srLdapSingleValue $
           srLdapCollective $
           srLdapNoUserModification $
           srLdapAttributeTypeUsage ) )
```

Additional metadata

```
objectclass ( 1.3.6.1.4.1.10126.1.14.4.1
    NAME 'srLdapAdditionalMetadataObject'
    DESC 'additional metadata provided for schema registry entries'
    AUXILIARY
    MAY  ( srlistingInformalComment $ SRreferencePointer $
           srSyntaxOK $ srOidOK ) )
```

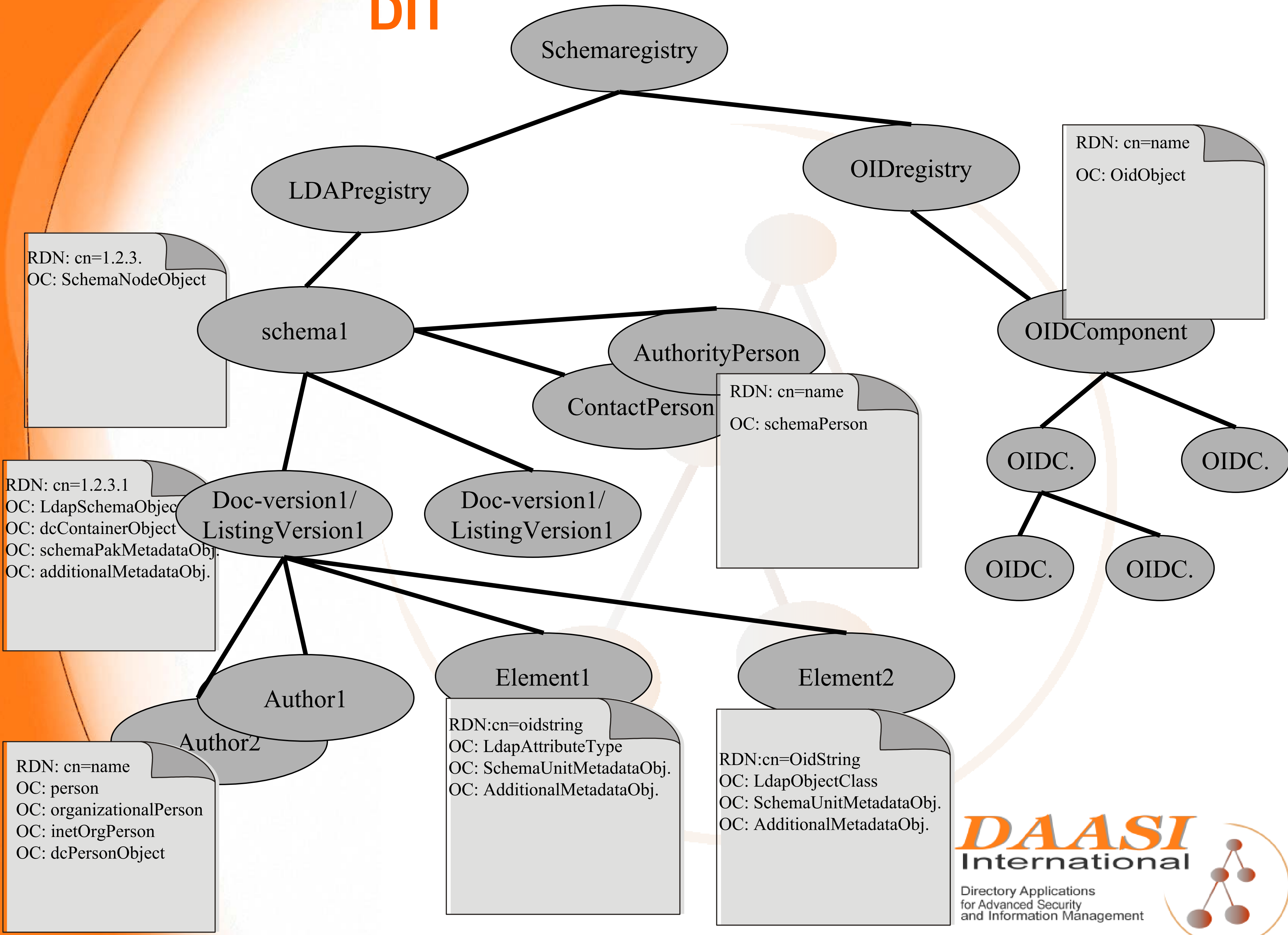
Additional schema for the DIT

```
objectclass ( 1.3.6.1.4.1.10126.1.8.4.4
    NAME 'srLdapSchemaNodeObject'
    DESC 'structural objectclass for a schema toplevel node'
    STRUCTURAL
    MUST cn
    MAY  ( srListingName $ srListingTitle $ srLdapSchemaDescriptor $
           srLdapSchemaVersions $ srLdapSchemaNewestVersion $ dcSubject $
           description)



   objectclass ( 1.3.6.1.4.1.10126.1.12.4.1
       NAME 'srOidObject'
       DESC 'Information about an ASN.1 Object Identifier'
       SUP  top
       STRUCTURAL
       MUST ( srOc $ srNumericOid )
       MAY  ( displayName $ description $ labeledURI $ mail $
              postalAddress $ srOidAuthName $ telephoneNumber ) )
```

DIT



Schemaregistry

LDAPregistry

OIDregistry

RDN: cn=name
OC: OidObject

RDN: cn=1.2.3.
OC: SchemaNodeObject

schema1

AuthorityPerson

ContactPerson

RDN: cn=name
OC: schemaPerson

OIDComponent

OIDC.

OIDC.

OIDC.

OIDC.

RDN: cn=1.2.3.1
OC: LdapSchemaObjec
OC: dcContainerObject
OC: schemaPakMetadataObj.
OC: additionalMetadataObj.

Doc-version1/ ListingVersion1

Doc-version1/ ListingVersion1

Author1

Author2

Element1

Element2

RDN: cn=name
OC: person
OC: organizationalPerson
OC: inetOrgPerson
OC: dcPersonObject

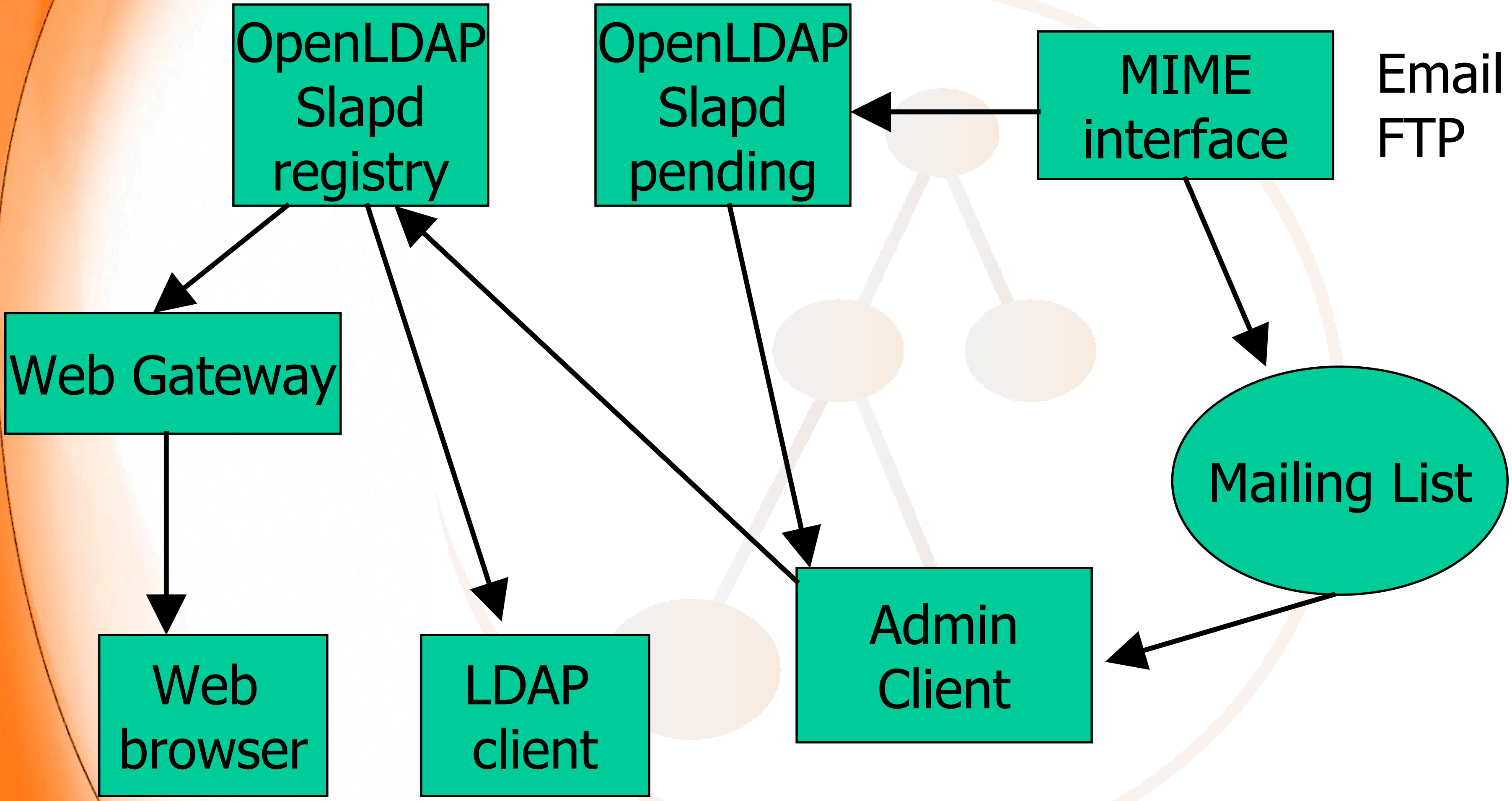
RDN:cn=oidstring
OC: LdapAttributeType
OC: SchemaUnitMetadataObj.
OC: AdditionalMetadataObj.

RDN:cn=OidString
OC: LdapObjectClass
OC: SchemaUnitMetadataObj.
OC: AdditionalMetadataObj.

DAASI International

Directory Applications
for Advanced Security
and Information Management

Workflow


OpenLDAP Slapd registry

OpenLDAP Slapd pending

MIME interface

Email
FTP

Web Gateway

Mailing List

Web browser

LDAP client

Admin Client

Business Modell

- After the project there has to be a funding modell for running the registry
- Either Organisations pay for registring their schema
- Or users pay for retrieving schema information
- Or Organisations just sponsor the registry
- It shouldn‘t be too costly to run the service
- Untill a solution is found DAASI will run it on ist own costs

Thank you for your attention

- More information at:
  - http://www.daasi.de/services/SchemaReg
  - Info@daasi.de

DAASI International

Directory Applications
for Advanced Security
and Information Management