Exim and LDAP

Making Exim Talk to an LDAP Server

UKUUG Summer 2009 Conference
Birmingham, UK
August 2009

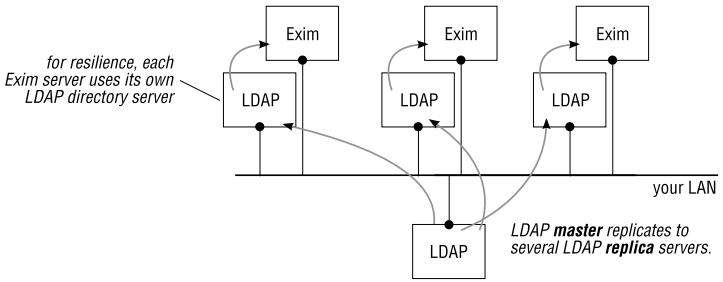
Jan-Piet Mens
mens.de

Overview

- Why Exim and LDAP are a good idea.
- Short LDAP refresher.
- Preparing Exim to use LDAP.
- How Exim uses LDAP.
- Summary.

Why LDAP with Exim?

- Centralized and distributed management.
- Automatic replication of configuration data.
- Centralized backup.
- Distributed data management. (ISP)
- Multiple servers share same configuration.



What can Exim do with LDAP?

- Look up e-mail addresses (people, aliases, groups).
- Conditionally route e-mail messages.
- Retrieve configuration settings.
- Virtualize domains. (E-mail "toaster".)
- Consolidate companies. (Groupware.)
- Authenticate users for SMTP.

LDAP Refresher

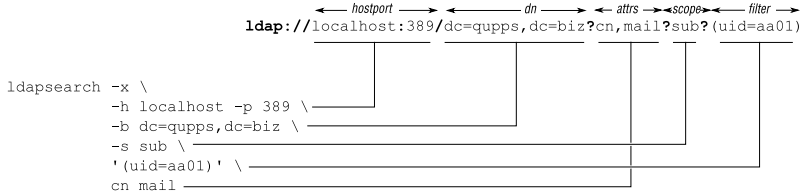
- Data organized hierarchically (authorization, replication).
- Tree structure contains entries (objects).
- Objects are structural or auxiliary.
- Distinguished Name (DN) composed of Relative DNs:

```
uid=janej, ou=People, dc=fupps, dc=com
```

- Top level of directory tree is the Base DN.
- Object classes define set of attributes allowed in entry.
- Attributes (types of information):
  - mandatory or optional
  - single-valued or multi-valued
- Objects inherit properties of parent classes.
- Standardized API.
- LDAP directories support fast reads.

Searching in LDAP

- Comparison: ldapsearch vs. LDAP URL:

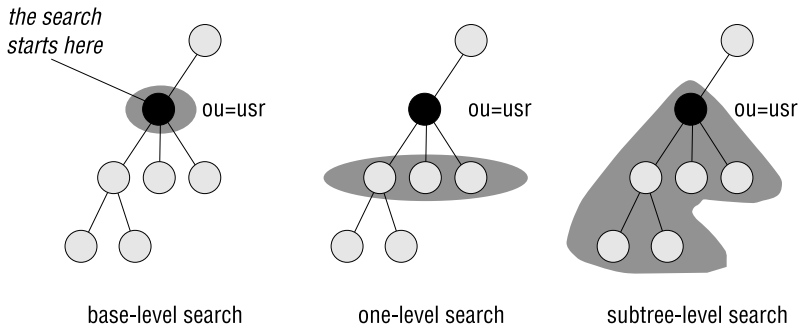


- Lightweight Directory Interchange Format:

```
dn: cn=Anne Mara,ou=Users,dc=qupps,dc=biz
cn: Anne Mara
mail: anne.mara@example.com
mail: am@qupps.biz
```

LDAP Search Scopes

- Search base defines where search will start.
- Search scopes:



Exim

- Exim is a Mail Transfer Agent (MTA).
- Developed at Cambridge by Philip Hazel.
- Variety of database look-ups
  - cdb, dbm, dsearch, lsearch, nis, dnsdb, mysql, ..., ldap
- Steps
  - Build Exim with LDAP support
  - Overview Exim configuration
  - How are LDAP queries used?
  - Test string expansions
  - Debug LDAP look-up functions
  - Debug LDAP connections

Build Exim with LDAP Support

- Enable LDAP in Local/Makefile:

```
LOOKUP_LSEARCH=yes
LOOKUP_LDAP=yes
...
```

- Specify type of LDAP libraries:

```
LDAP_LIB_TYPE=OPENLDAP2
```

- Add libraries and include directories:

```
LOOKUP_INCLUDE=-I/usr/include ...
LOOKUP_LIBS=-lldap -llber
```

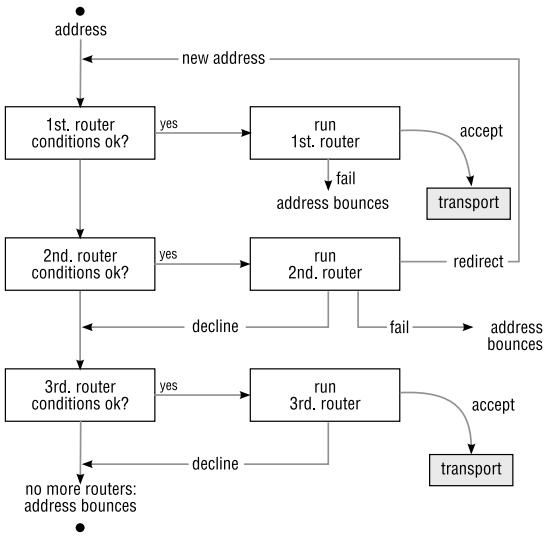
Exim configuration basics

- Exim configuration file is divided into sections:

```
qualify_domain = mta.example.com

BASEDN = dc=fupps,dc=com
BINDDN = uid=exim,ou=machines,BASEDN
```

- Access Control Lists
  - check recipients, authenticated, relay, etc.
- Routers
  - dnslookup, system aliases, forwarding, local users
- Transports
  - remote smtp, local delivery, pipe, file
- Retry
- Rewrites
- Authenticators

Exim: Routers and Transports



LDAP queries in Exim

- In database lookup and string expansions
  - ldap
  - ldapdm
  - ldapdn
- LDAP entries without attributes are considered non-existent.
- LDAP quoting

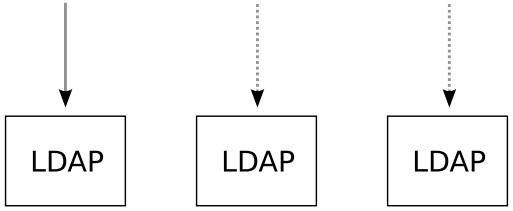
```
${quote_ldap:${local_part}}
${quote_ldap_dn: ... }
```

LDAP URLs in Exim

- Lookups use LDAP URLs

```
ldap://<hostname>:<port>/...
ldaps://<hostname>:<port>/...
ldapi://<pathname>/...
```

```
ldap_default_servers = \
   localhost : ldap2::563 : ldap3
```



Debugging: Exim

- Test Exim's string expansions:

```
# exim -be
> I am $primary_hostname
I am mta.example.net

> Hello, ${lookup ldap{
ldap:///dc=fupps,dc=com?cn?sub?sn=jolie }}
Hello, Jane Jolie

> ${lookup
ldap{ldap:///dc=fupps,dc=com?uid,cn?sub?uid=janej}}
cn="Jane Jolie" uid="janej"

> ${extract{uid}{ ${lookup
ldap{ldap:///dc=fupps,dc=com?uid,cn,mail?sub?uid=janej}}
}}
janej
```

Debugging: Exim's lookup functions

- View Exim doing look-ups:

```
# exim -bt -d+lookup postmaster@fupps.com
--------> ldap_aliases router <--------
local_part=postmaster domain=fupps.com
calling ldap_aliases router
rda_interpret (string): ${lookup ldapm
{user=uid=exim,dc=fupps,dc=com pass="shhh"

ldap:///dc=fupps,dc=com?rfc822MailMember?sub?(&(objectClass=
   nisMailAlias)(cn=${quote_ldap:$local_part}))}
{$value} fail }
...
extract item: jpm
extract item: post.master@example.org
extract item: root-example.com@example.com
extract item: root-megacor.biz@megacor.biz
extract item: root-qupps.biz@qupps.biz
```

Debugging: LDAP connections

- Keep an eye on LDAP server's log file. OpenLDAP:
  - slapd.conf

```
loglevel stats2 stats
```

  - slapd.log

```
conn=4 SRCH base="dc=fupps,dc=com" scope=2 \
        filter="(uid=janej)"
conn=4 SRCH attr=uid cn mail
conn=4 ENTRY dn="cn=jane jolie,dc=fupps,dc=com"
conn=4 SEARCH RESULT tag=101 err=0 nentries=1 text=
conn=4 UNBIND
conn=4 fd=12 closed
```

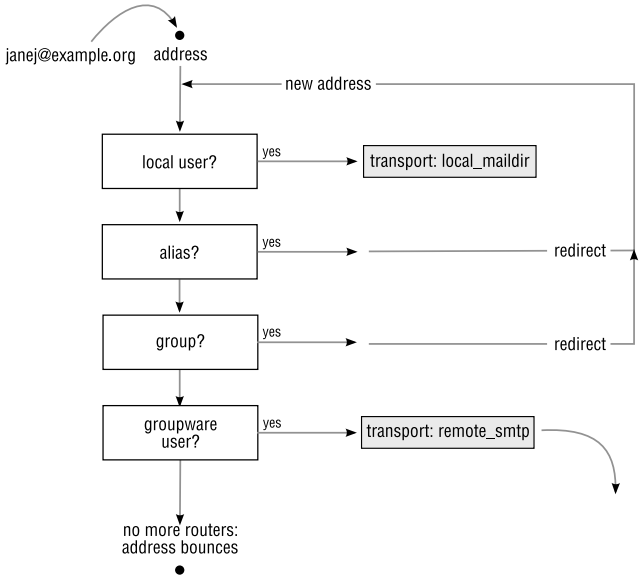
  - Warning only:

```
<= bdb_equality_candidates: (eximCfActive) not indexed
```

Using LDAP queries in Exim

- What we'll be covering now:
  - Addressing users.
  - Aliasing.
  - Using groups.
  - Dynamic LDAP queries.
  - Conditional routing.
  - Virtual hosting.
  - Smart host with custom schema.
  - SMTP authentication.
  - Groupware integration.

Routers and transports revisited



Addressing users

- Decide how to find user
  - e-mail address, username, other? (Know your data!)
- inetOrgPerson object has multi-valued 'mail' attribute, which says nothing much about routing.
- If not local, which target e-mail server?
- Add 'inetLocalMailRecipient' in such cases.
  - 'mailRoutingAddress' is the address of final delivery.

```
dn: uid=aa01, ...
mailLocalAddress: aa@my-home.org
mailLocalAddress: a.a@example.org
mail: anne@gmail.com
mailRoutingAddress: anna.m@e3.example.com
```

- Documentation and examples at

```
http://www.sendmail.org/m4/ldap_routing.html
```

Addressing: inetLocalMailRecipient router

- Example router for inetLocalMailRecipient

```
ldap_mailRouting:
 driver = redirect
 allow_fail
 allow_defer
 data = ${lookup ldapm {\
    ldaps:///BASEDN?mailRoutingAddress?sub?\
    (&\
    (objectClass=inetLocalMailRecipient)\
    (mailLocalAddress=\
      ${quote_ldap:$local_part@${domain}}\
    ))}{$value} fail }
```

Aliasing

- An alias is a forwarding e-mail address.
  - hostmaster, postmaster
- Reminder; from a file:

```
data = ${lookup{$local_part}lsearch{/etc/aliases}}
```

- inetLocalMailRecipient
  - auxiliary object class
  - add to any other object (e.g. person, account)
  - expired draft (Lachman-laser)
  - used by sendmail
- qmailGroup
- nisMailAlias
  - structural object class
- Roll your own?

Aliases with nisMailAlias

- Multivalued ‘rfc822MailMember’

```
dn: cn=postmaster,ou=Aliases,dc=fupps,dc=com
objectClass: top
objectClass: nisMailAlias
cn: postmaster
rfc822MailMember: jpm
rfc822MailMember: post.master@example.org
```

- Recommend one container for catchall aliases.
- Separate rest into "domain" containers (virtual hosting).

```
... ,ou=${domain}, ...
```

Aliases with nisMailAlias: router

- Example:

```
ldap_aliases:
 driver = redirect
 allow_fail
 allow_defer
 data = ${lookup \
   ldapm { \
           user=BINDDN \
           pass=BINDPW \
           ldap:///BASEDN?rfc822MailMember?sub?\
           (&\
             (objectClass=nisMailAlias)\
             (cn=${quote_ldap:$local_part}))\
           } {$value} fail }
 file_transport = address_file
 pipe_transport = address_pipe
```

Using groups

- Groups or distribution lists.
- posixGroup

```
memberUid: janej
```

- groupOfNames

```
member: cn=John Duck,ou=Users,o=example.net
```

- rfc822MailGroup

```
member: any.body@example.org
```

- mailGroup

```
uniqueMember: cn=Daisy Quack,ou=Users,o=example.org
```

- (any others?)

- We discuss one example: groupOfNames

Using groups: groupOfNames

- The group:

```
dn: cn=g2,ou=Groups,dc=fupps,dc=com
objectClass: groupOfNames
cn: g2
member: cn=Jane Jolie,ou=Users,dc=fupps,dc=com
member: cn=John Duck,ou=Users,dc=fupps,dc=com
```

- The member:

```
dn: cn=John Duck,ou=Users,dc=fupps,dc=com
objectClass: inetOrgPerson
cn: John Duck
uid: johnd
mail: john.duck@fupps.com
memberOf: cn=g2,ou=Groups,dc=fupps,dc=com
```

- The pain.
- Referential integrity needs maintaining...

Using groups: groupOfNames

- OpenLDAP has overlays (bits of code) that modify behaviour of back-end (slapo-<overlay name>).
- "memberof"-overlay updates attribute (memberOf) whenever membership attribute changes.
- In slapd.conf:

```
database bdb
suffix "dc=example,dc=net"
overlay memberof
```

- Exim router to handle memberOf

```
ldap_groups:
 driver = redirect
 data = ${lookup ldapm{ldap:///PEOPLEB\
   ?mail?sub?(memberof=${lookup ldapdn{ \
      ldap:///GROUPBASE\
      ??sub?(cn=${quote_ldap:$local_part})}})\
 }}
```

About groups

- Have to be created and maintained (by admins).
- Users create their own groups/lists in MUA.
- Often outdated.
- You have it all in LDAP anyway, so why not use it?

Dynamic LDAP queries

- Use an LDAP search-string as an e-mail address?
- All chemistry students?

```
{&{etype=student}{ou=Chem}}@example.net
```

- People in data centre except Mr. Zech?

```
{&{etype=*}{ou=Computing}{!{sn=Zech}}}@example.net
```

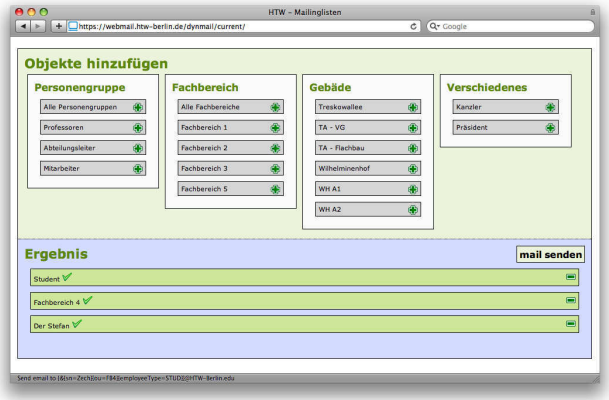
- Allow from within organisation only. (Spam.)
- Created by Stefan Zech of HTW Berlin.

```
http://tinyurl.com/l76mys
```

- End-user can't really write LDAP URLs, can she?
- Use Web-frontend or other custom application.

```
https://webmail.htw-berlin.de/dynmail/current/
```

Dynamic LDAP queries (2)



- MUA is invoked with mailto:-URL

```
{&{sn=Zech}{ou=FB4}{employeeType=STUD}}@HTW-Berlin.edu
```

Dynamic LDAP queries (3)

- The code. Two macros:

```
DYN_FILTER = \
    ${sg {${sg {$local_part}{\{}{(}}}{\}}{)}}

LDAP_DYN_SEARCH = ${lookup ldapm {\
        ldap:///LDAP_BASE?mail?sub?DYN_FILTER}}
```

- And the router:

```
ldap_dyn_search:
  condition = ${if eq {\
        ${length_1:$local_part}}{\{}{yes}{no}}
  driver = redirect
  data = LDAP_DYN_SEARCH
  headers_add = X-HTW-LDAP: DYN_FILTER
```

- That's it. (Brilliant, and I told Stefan as much.)

Dynamic LDAP queries (4)

- Possibly take it all a step further.
- Provide interface to store queries as LDAP entries.

```
cn=dyn-managers-1st-floor, ou=Canned-Queries
cn: dyn-managers-1st-floor
dynQuery: {&{roomNumber=1-*}{eType=Boss}}@example.net
```

- Exim router condition local part begins with "dyn-*" and then redirect.
- OpenLDAP has the 'dynlist' overlay:

```
dn: cn=All-M,ou=Groups,dc=fupps,dc=com
objectClass: groupOfURLs
cn: All-M
memberURL: ldap:///o=ex.org?mail?sub?(sn=M*)
```

Conditional routing: Ask LDAP if it's OK

- Use or skip a router
- Route a message depending on LDAP result
- Process messages specially on a per/user basis

```
dn: cn=Anne Mara,ou=Users,dc=qupps,dc=biz
cn: Anne Mara
service: dspam
service: ftp
service: internet
```

Conditional routing: Example

- Route to content scanner.

```
dspam_router:
 condition = "${if and { \
 {!def:h_X-Spam-Flag:} \
 {!eq {$received_protocol}{local}} \
 { <= {$message_size}{512k}} \
 {!eq {\
  ${lookup ldap {ldaps:///PEOPLEB?uid?sub?\
      (&(mail=${quote_ldap:$local_part@$domain})\
        (service=dspam)) \
  } {$value} {}}\
  {} } \
 }\
 {1}{0}}"
 driver = accept
 transport = dspam_spamcheck
```

Virtual hosting

- Set up a directory container per domain.

```
ou=Cloud
   ou=example.com
      ou=Usr
        cn=Jane Doe
      ou=Aliases
        cn=postmaster
   ou=megacor.biz
      ou=Usr
      ou=Aliases
```

- Exim has to know if domain is "local". Populate local domains ...

```
domainlist local_domains = @ : \
    ldapm;ldaps:///CLOUDBASE?ou?one
```

- ... or check for a single domain.

```
domainlist local_domains = @ : \
    ldap;ldaps:///ou=${domain},CLOUDBASE??base
```

Virtual hosting: router

- Is this user part of our virtual host?

```
LDAP_LOCALUSER  = \
    user=BINDDN \
    pass=BINDPW \
    ldaps:///ou=Usr,ou=${domain},CLOUDBASE?mail?sub?\
     (mail=${quote_ldap:${local_part}}@${domain})
```

- Yes, so route the message (here with a local transport)

```
ldapuser:
 driver      = accept
 condition   = ${lookup ldap{LDAP_LOCALUSER}}
 transport   = local_maildir
```

Virtual hosting: transport

- Determine user's mailDir directory, using posixAccount class.

```
L_HOME = ldaps:///ou=${domain},CLOUDBASE?\
    homeDirectory?sub?\
    (mail=${quote_ldap:${local_part}}@${domain})
```

- The Exim transport:

```
local_maildir:
 driver = appendfile
 maildir_format = true
 directory = ${lookup \
   ldap{L_HOME}{${sg{$value}{\$}{/Maildir}}} }
 maildir_tag = ,S=$message_size
 user = 7600
 group = mail
```

Smart host a la Carte

- A smart host is a mail relay server.
- Post configuration of an appliance.
- So far only standard object classes; here we add our own.
- Schema defines object classes in DIT.
- Typically contained in files.
- Schema is extensible: you can create your own objects.
- Elements identified by Object Identifier (OID).

```
1.3.6.1.4.1.1466.115.121.1.15
```

- Attributes are associated with:
  - a syntax
  - a matching rule

Custom schema: Example

- Schema file:

```
attributetype ( 1.3.6.1.4.1.7637.30.1.1.2
  NAME 'eximCfActive'
  DESC 'Config setting enabled?'
  EQUALITY booleanMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.7
  SINGLE-VALUE
  )

objectclass ( 1.3.6.1.4.1.7637.30.1.2.1
  NAME 'eximConf'
  SUP top STRUCTURAL
  DESC 'Exim configuration stanza'
  MUST ( cn $ eximCfActive )
  MAY ( eximCfValue $ description $ seeAlso )
  )
```

Custom schema: LDIF

- LDIF for our smart host:

```
dn: cn=exim-smarthost,ou=Conf,dc=fupps,dc=com
cn: exim-smarthost
objectclass: eximConf
eximCfActive: FALSE
eximCfValue: my-smarthost.example.org
description: If you need a smart host for
 outgoing mails from this host, set its
 hostname in eximCfValue and enable smart-host
 processing by setting eximCfActive.
```

Custom schema: Smarthost in Exim

- A macro

```
SMARTHOST = ldap:///ou=Conf,BASEDN?eximCfValue?\
  one?(&\
    (objectclass=eximConf)\
    (cn=exim-smarthost)\
    (eximCfActive=TRUE)\
  )
```

- The corresponding Exim router

```
ldap_smart_route:
 driver = manualroute
 domains = !+local_domains
 condition  = ${lookup ldap{SMARTHOST}}
 transport = remote_smtp
 route_list = * "${lookup ldap{SMARTHOST}}"
```

SMTP Authentication

- Plain, Login
  - Authentication data plain text. Use TLS encryption!
- CRAM-MD5
  - Challenge/response. Server needs access to unencrypted password.
- How should users authenticate?
  - e-mail address: jdoe@example.com
  - username: jdoe
- Authentication with LDAP requires DN.
- Can you afford to "construct" the DN?

```
user="uid=$auth1,ou=People,dc=fupps,dc=com"
```

- Better to search for the DN:

```
user="${lookup ldapdn {\
  ldaps:///BASEDN?dn?sub?\
  (&(uid=${quote_ldap:$auth1})(mail=*))}}"
```

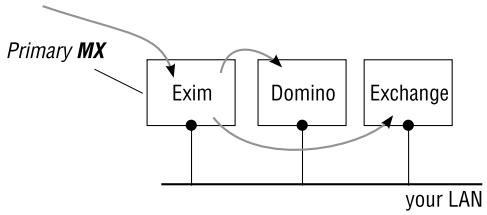
SMTP Authentication: server_condition

- Plain authentication with LDAP

```
USER = ldaps:///BASEDN?dn?sub?(&\
   (|(mail=${quote_ldap:$auth2})\
   (mailAlternateAddress=${quote_ldap:$auth2})))

ldap_plain:
 driver = plaintext
 public_name = PLAIN
 server_condition = ${if ldapauth {\
    user="${lookup ldapdn {USER}{$value}fail}" \
    pass=${quote:$auth3} \
    ldap:///BASEDN/ \
    }{yes} {no} \
    }
 server_set_id = ${sg{$ldap_dn}{\s+}{}}
```

Integrate your Groupware Server

- Groupware
  - IBM/Lotus Domino, Microsoft Exchange, Novell Groupwise



- All have LDAP directory -- enable it.
- Know their data: test LDAP queries with ldapsearch.
- Use OpenLDAP's ldap/meta back-end as a proxy?
- Specifically query foreign LDAP in Exim router.
- Before or after your "local" users?

Groupware server: How to integrate

- Sample Domino LDIF:

```
dn: CN=John Doe,OU=marketing,O=fupps.com
cn: John Doe
mail: jdoe@fupps.com
objectclass: dominoPerson
mailsystem: 1
uid: john.doe
uid: john.q.doe
mailserver: CN=JP510m,O=fupps.com
mailfile: mail\jdoe.nsf
```

- "Replicate" foreign directory data to your LDAP.
- Massage foreign server names into DNS names.
- Create "complete" inetLocalMailRecipient entries.

Groupware server: Exim router

- E.g. route to IBM/Lotus Domino if user exists there:

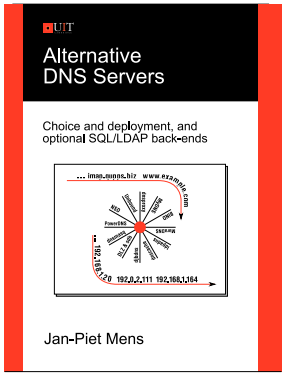
```
MEGACORPUSER = ldap://172.16.153.130/\
   ?mail?sub?\
   (mail=${quote_ldap:${local_part}}@${domain})

domino_split:
  driver = manualroute
  domains = megacorp.info
  condition = ${lookup ldap{MEGACORPUSER}}
  route_list = * "domino.megacorp.info"
  transport = remote_smtp
```

Summary

- Exim and LDAP make a very flexible system.
- Use good tools for managing LDAP. I use
  `http://www.lichteblau.com/ldapvi/`
  `http://directory.apache.org/studio/`
- Deploy more than one LDAP server; keep them "close" to Exim.
- Use selective replication (OpenLDAP syncrepl).
- But: Watch out for moving parts.
  - Once upon a time: MTA, DNS, files
  - Today: MTA, DNS, files, TLS, LDAP, RBL, Groupware, content-scanners
- Not a limitation of LDAP.
- Create static databases (e.g. CDB) from LDAP?
  `Net::LDAPapi, Net::LDAP`

Further reading

- Philip Hazel, 2007, The Exim SMTP Mail Server, 2nd ed.
  `http://uit.co.uk/content/exim-smtp-mail-server`
- Jan-Piet Mens, 2009, Alternative DNS Servers, UIT
  `http://uit.co.uk/altdns`



Thank you

Questions?