

OpenLDAP & Meta-Directory

Pierangelo Masarati <ando@sys-net.it>
SysNet s.n.c., Italy

OpenLDAP Developers' Day
July 18, 2003, Wien

Index

- back-meta
- slapd architecture
- motivations
- scenarios
- meta-directory
- issues
- roadmap

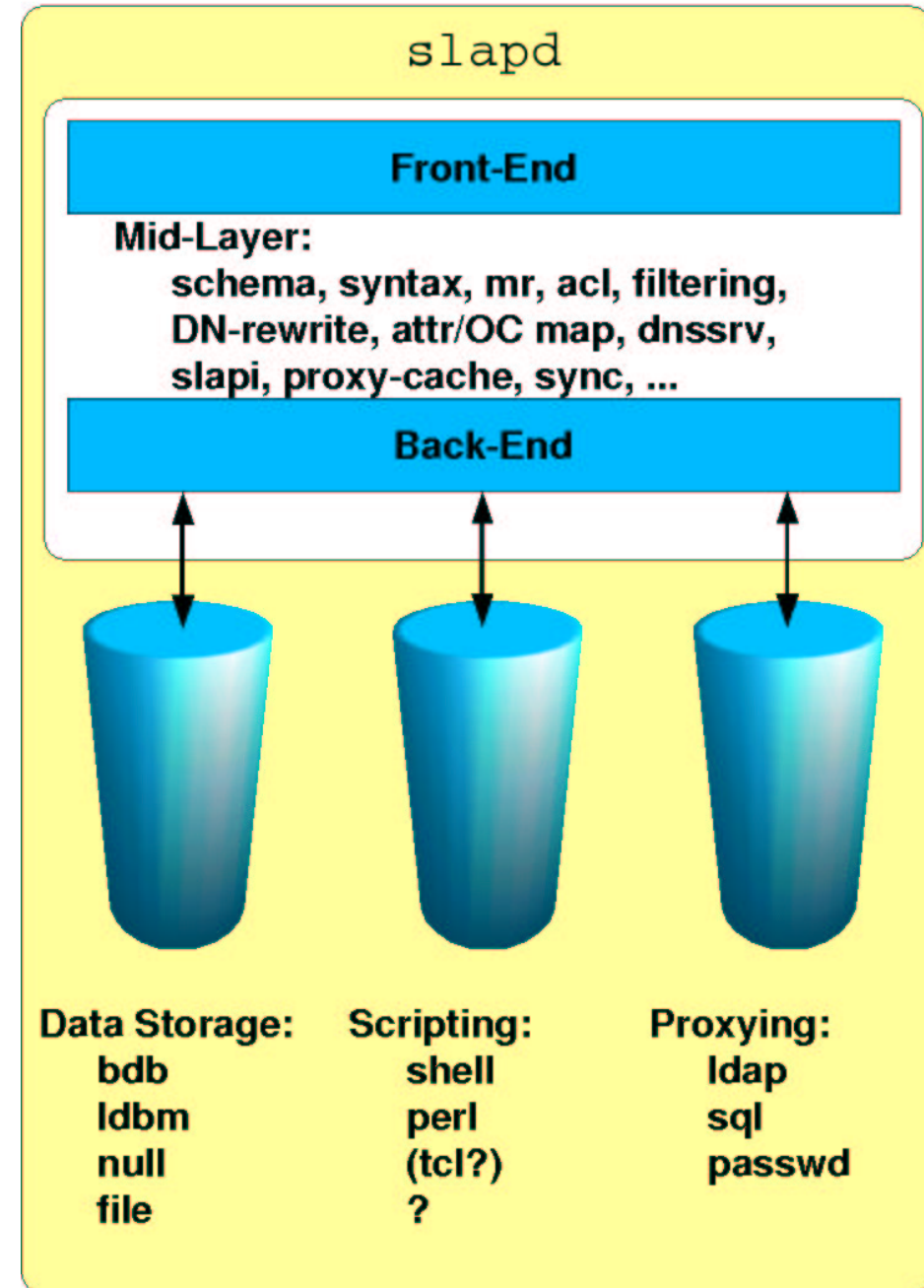
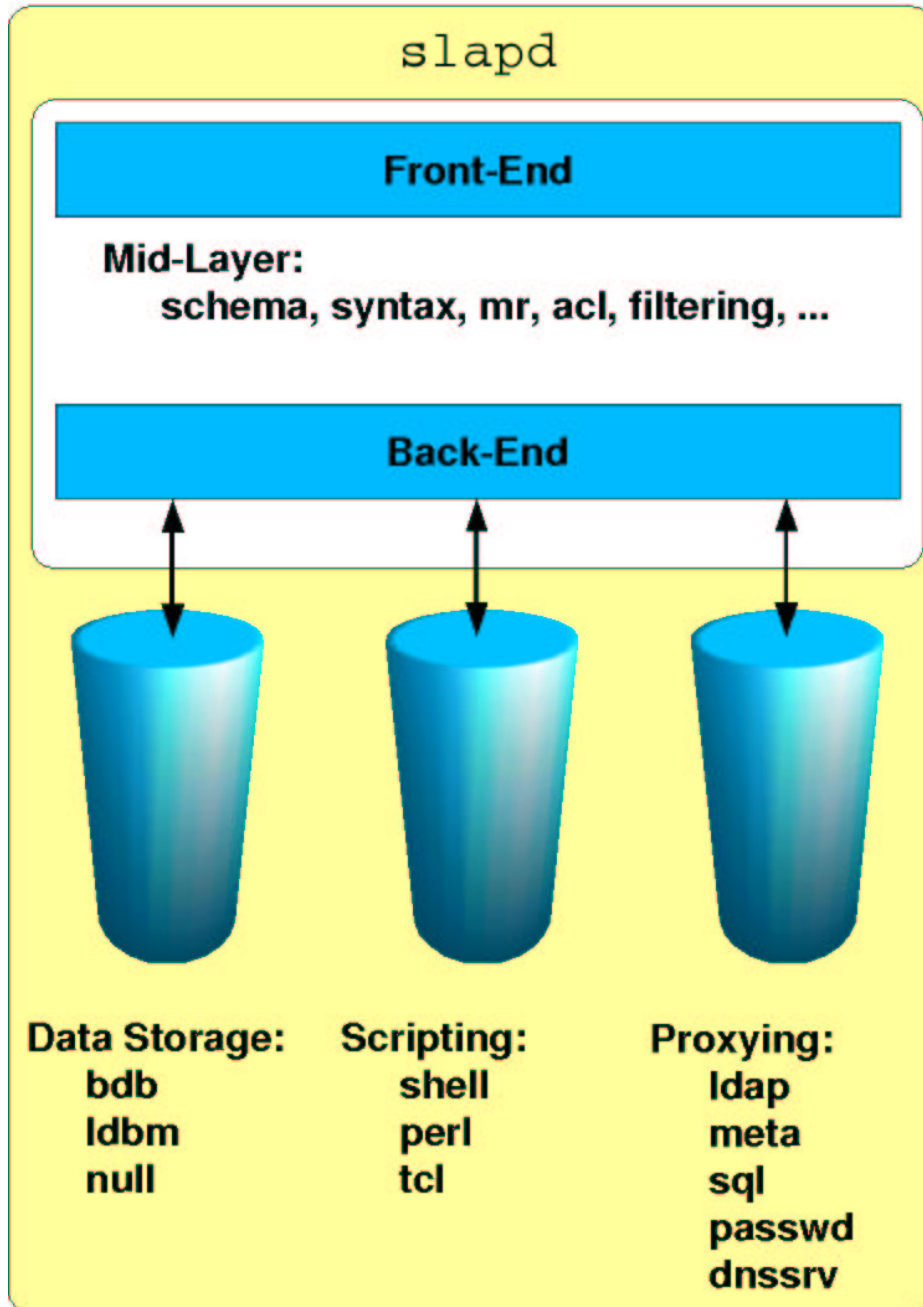
back-meta

- History:
 - from back-ldap to back-meta
 - DN rewriting
 - attribute/objectClass mapping
- Provides:
 - remote directory glueing
 - DN & operation-related rewriting
 - a place for slapd hacking (proxy-cache?)

back-meta (...)

- Limitations:
 - no multiple DN check/countermeasures
 - no entry join/merge
- Possible Evolution:
 - implement join/merge
 - implement/adapt caching (proxy-cache?)
 - overcome limitations/define operative limits

Slapd architecture



Back-Meta Functionality

- replace back-meta with:
 - glueing
 - multiple instances of back-`{ldap|sql|bdb|...}`
 - attr/OC map, DN rewrite moved to mid-layer
 - slapi for special features (e.g. customization)
- proxy-cache:
 - decouple from back-`{ldbm,meta}`
 - move to mid-layer

Scenarios

- A) Etherogeneous sources of user (/application/whatever) profiling that should be glued together and presented as a unique directory server
 - pull operation mode
- B) Homogeneous, specialized directory services that should also be presented as a unique directory server
 - pull/push operation mode

Motivation

- Improve flexibility and scalability (in enterprise terms)
- Allow OpenLDAP deployment in rigid infrastructures (in enterprise terms: limited open source penetration, ...)
- ...
- Customers like the term “Meta-Directory” ;)

Motivation (...)

- Hot topic is SQL interoperability, where RDMBS don't offer native solutions
- back-sql is an important part of the picture
- back-sql needs improvements:
 - syntax compliance (“reasonable” equivalences)
 - UTF-8 support?
- Meta-Directory solution:
 - doesn't require ALL data to be on one database
 - allows to publish databases not under control

Meta-Directory Functionality

need to implement a join engine to merge partial entries; issues:

- join criteria (DN? rewritten DN? attr(s)? filter?)
- schema consistency
- temporary storage before sending results
- conflicts due to successive target modifications
- tracking of common attrs authoritative source
- sync/async cache update

Meta-Directory Strategies

- Caching is almost mandatory:
forget about back-meta!
- Synchronous (impractical):
 - cache is updated on request
 - poor performance
 - consistency “guaranteed”
- Asynchronous:
 - cache is updated/sync'd on regular basis
 - excellent performance
 - loose consistency

Meta-Directory Issues

- filtering: entries must be already merged (filter templates as in proxy-cache?)
- access control consistent with sources (a “philosophical” problem, same as with back-ldap)
- cache synchronization of common attrs/OC (reference counter and source tags?)
- write operations (hints/templates for data split? readonly?)

Tentative Roadmap

- Figure out possible solutions to the different set of detailed problems
- Select and prioritize feasible/desirable ones
- Perform anticipated architectural changes
- Exploit existing software by callbacks:
 - merge engine from proxy-cache
 - source gathering from back-glue
 - cache storage from bdb

Conclusions

- key metadirectory feature: entry merge/join
- suggest back-meta out of slapd
- suggest proxy-cache w/o back-`{ldbm,meta}`
- possible scenario(s) for asynchronous join engine, with key issues to be faced yet