From an OpenLDAP back-end for Samba to a Samba back-end for OpenLDAP

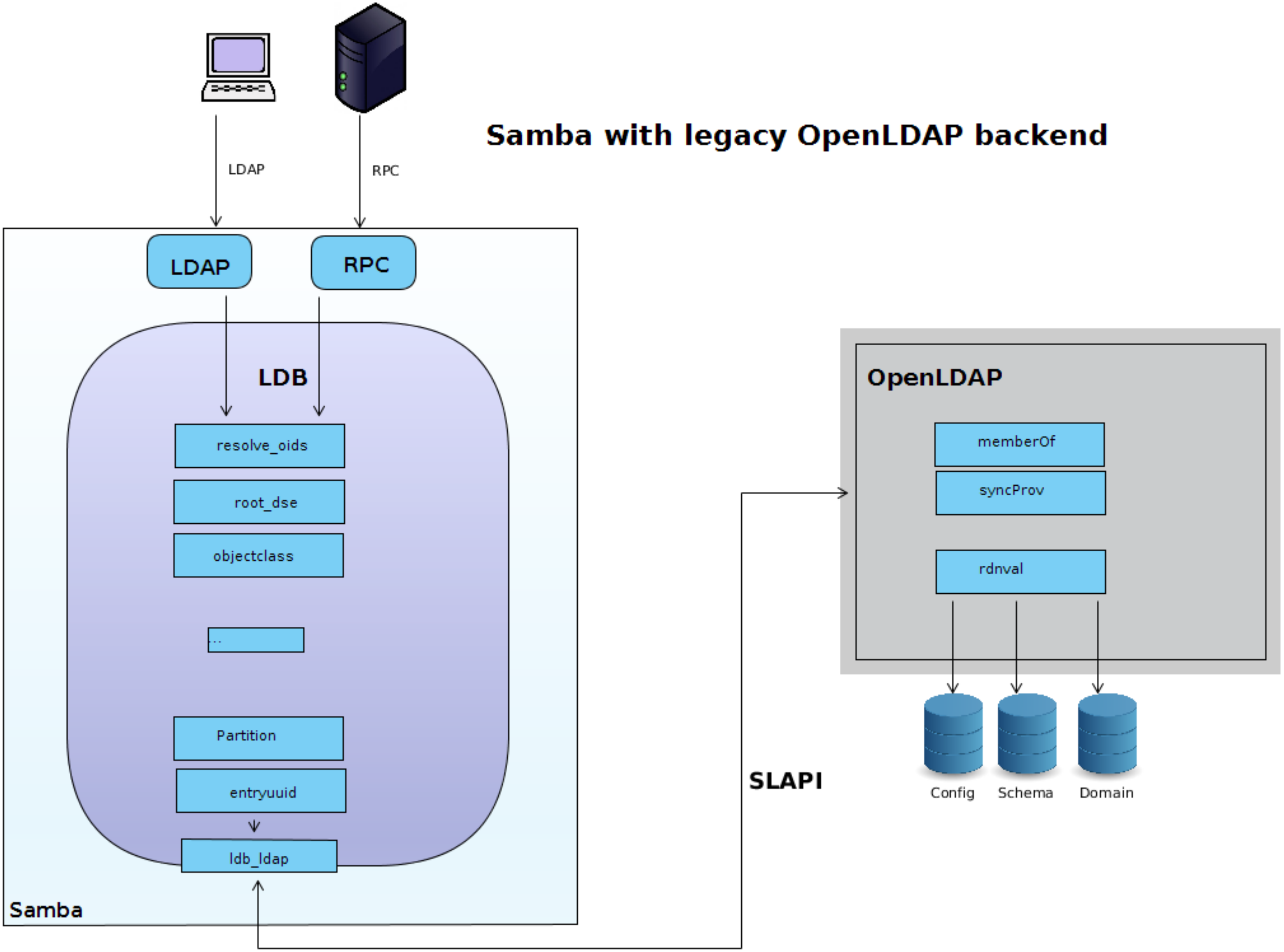
Nadezhda Ivanova
Software Developer @ Symas Corp

SambaXP 2023, Gottingen

A new back-end for Samba 4

- Integrate Samba 4’s AD implementation with the speed and scalability of OpenLDAP
- Samba 4 (used to) have a built-in size limitation due to use of TDB
- Samba 4 (used to) have a slow LDAP service.
- Combine OpenLDAP's excellence with Samba's know-how.
- LDAP traffic should be handled by the one best suited for the job – OpenLDAP itself.
  - Move the LDB modules that implement AD specific operations to OpenLDAP whenever needed.
  - RPC and other protocols will still be handled by Samba
- “Relieve” Samba of its LDAP server.

Samba with legacy OpenLDAP backend



Samba provisioning with Legacy OpenLDAP

- Samba provisioning scripts created slapd.conf
  - cn=Schema
  - cn=Configuration
  - Domain
  - 2 DNS application partitions
  - Refint and memberOf configuration to implement linked attributes
  - Indexing configuration
- Provisioning script created a schema definition file for OpenLDAP – backend.schema
- Populated the created databases with the necessary initial data, including cn=Schema

top

(2.5.6.0 NAME 'top'

"DESC 'top of the superclass chain' "

"ABSTRACT MUST objectClass)"

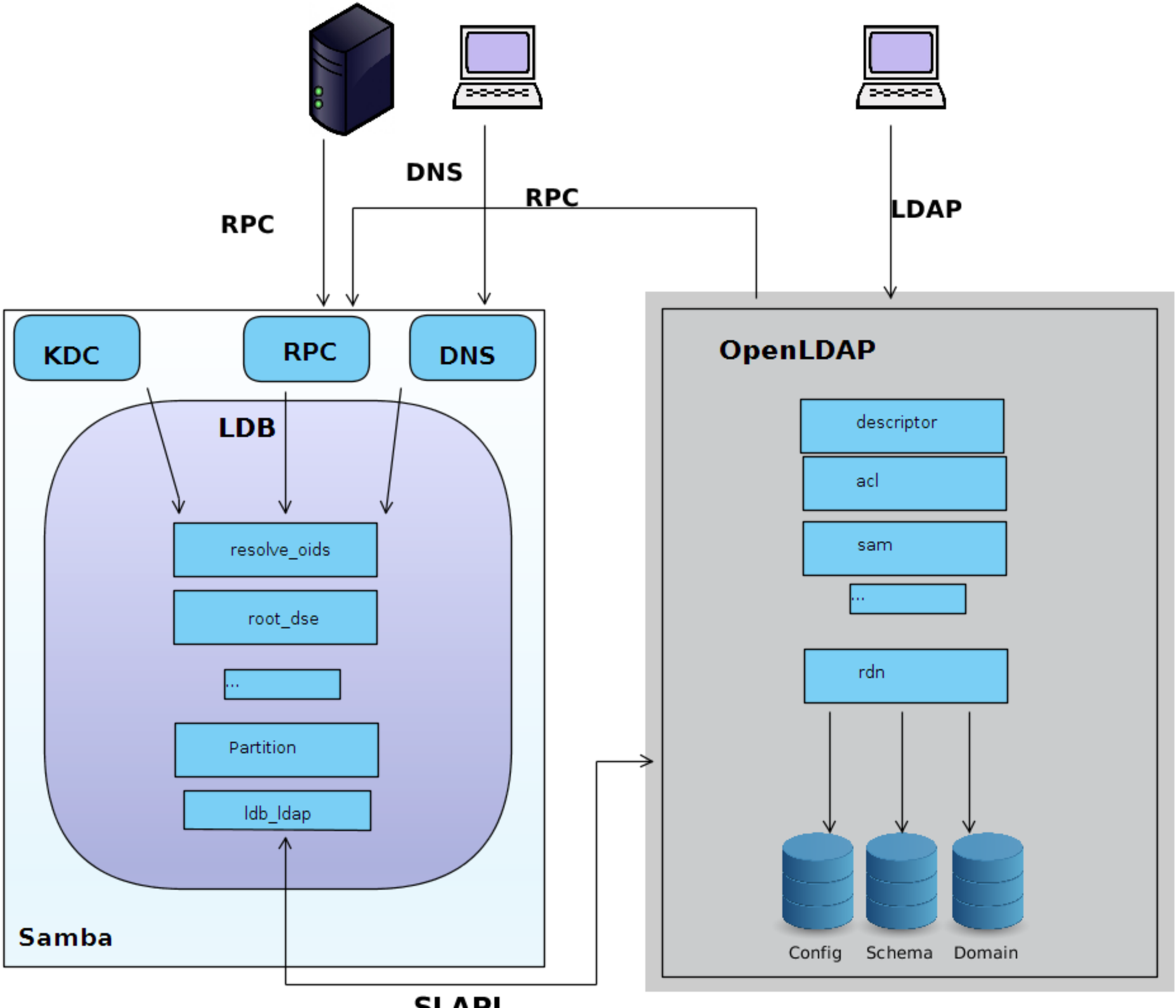
"top", "(2.5.6.0 NAME 'top' "

"DESC 'top of the superclass chain' "

"ABSTRACT MUST (objectClass) "

MAY (instanceType $ nTSecurityDescriptor $ objectCategory $ adminDescription $ adminDisplayName $ allowedAttributes $ allowedAttributesEffective $ allowedChildClasses $ allowedChildClassesEffective $ bridgeheadServerListBL $ canonicalName $ cn $ description $ directReports $ displayName $ displayNamePrintable $ dSASignature $ dSCorePropagationData $ extensionName $ flags $ fromEntry $ frsComputerReferenceBL $ fRSMemberReferenceBL $ fSMORoleOwner $ isCriticalSystemObject $ isDeleted $ isPrivilegeHolder $ lastKnownParent $ managedObjects $ masteredBy $ mS-DS-ConsistencyChildCount $ mS-DS-ConsistencyGuid $ msCOM-PartitinSetLink $ msCOM-UserLink $ msDS-Approx-Immed-Subordinates $ msDs-masteredBy $ msDS-MembersForAzRoleBL $ msDS-NCReplCursors $ msDS-NCReplInboundNeighbors $ msDS-NCReplOutboundNeighbors $ msDS-NcType $ msDS-NonMembersBL $ msDS-ObjectReferenceBL $ msDS-OperationsForAzRoleBL $ " "msDS-OperationsForAzTaskBL $ msDS-ReplAttributeMetaData $ msDS-ReplValueMetaData $ msDS-TasksForAzRoleBL $ msDS-TasksForAzTaskBL $ name $ netbootSCPBL $ nonSecurityMemberBL $ objectVersion $ otherWellKnownObjects $ ownerBL $ parentGUID $ partialAttributeDeletionList $ partialAttributeSet $ possibleInferiors $ proxiedObjectName $ proxyAddresses $ queryPolicyBL $ replPropertyMetaData $ replUpToDateVector $ repsFrom $ repsTo $ revision $ sDRightsEffective $ serverReferenceBL $ showInAdvancedViewOnly $ siteObjectBL $ subRefs $ systemFlags $ url $ uSNDSALastObjRemoved $ USNIntersite $ uSNLastObjRem $ uSNSource $ wbemPath $ wellKnownObjects $ wWWHomePage $ msSFU30PosixMemberOf $ msDFSR-ComputerReferenceBL $ msDFSR-MemberReferenceBL $ msDS-EnabledFeatureBL $ msDS-LastKnownRDN $ msDS-HostServiceAccountBL $ msDS-OIDToGroupLinkBl $ msDS-LocalEffectiveRecycleTime $ msDS-LocalEffectiveDeletionTime $ isRecycled $ msDS-PSOApplied $ msDS-PrincipalName $ msDS-RevealedListBL $ msDS-AuthenticatedToAccountlist $ msDS-IsPartialReplicaFor $ msDS-IsDomainFor $ msDS-IsFullReplicaFor $ msDS-RevealedDSAs $ msDS-KrbTgtLinkBl $ whenCreated $ whenChanged $ uSNCreated $ uSNChanged $ subschemaSubEntry $ structuralObjectClass $ objectGUID $ distinguishedName $ modifyTimeStamp $ memberOf $ createTimeStamp $ msDS-NC-RO-Replica-Locations-BL))"

New Samba OpenLDAP Backend



Implementation approach

- We started by using the legacy back-end and replacing individual modules
- But:
  - Samba modules are interconnected and often communicate with each other via internal controls
  - Ldb modules ≈ 40 000 lines of C
  - They rely on being executed in a specific order, and not all of them can be removed
  - Sometimes RPC traffic is initiated from inside a module, e.g samldb and replmetadata

Where did we get by this

- I learned to write OpenLDAP overlays ;)
- Modified OpenLDAP internal schema so that Samba4Top mapping is unnecessary. This required rewriting OpenLDAP's internal schema code
- InstanceType
- showDeleted
- Some constructed and operation attributes, special and secret attributes
- Some attempts at access checks

Where did we get?

- Secdescriptor overlay
  - Collects the necessary data – parent SD, default security descriptor.
  - Calculates the new descriptor using some Samba library functions and adds it to the new entry.
  - Recalculates the SD's of the modified object and all of its children.
  - Handles the sDFlags control
  - Gets the security token as a control from Samba

A new back-end for Samba 4, take two

- Switch to separate implementation of functionality within OpenLDAP, with manual testing via OpenLDAP directly, until LDAP behavior is as desired
- Use Samba’s provisioning script to populate a database, then rely on that to gradually add functionality to OpenLDAP
- Determine how and if to remove or modify Samba modules later, after RPC tests

Active Directory Schema

- Defined by objects of type attributeSchema and classSchema
- Schema updates are performed by added new objects of this type in the cn=Schema,cn=Configuration partition
- Schema objects cannot be deleted, only set to “defunct”
- Schema objects contain additional data, necessary for AD operation
- Some standard classes have additional non-standard attributes – e.g “top”

Where did we get?

- ad_schema overlay - registers the attributeSchema and classSchema attributes in OpenLDAP schema
  - Maps the AD style syntax to LDAP syntax
  - creates schema definition for the class or attribute that is registered in OpenLDAP schema
  - Adds the additional schema data to the expanded AttributeType and objectClass data
  - If the attribute is indexed, creates an index value for it in cn=config
  - If the attribute is linked, creates a memberOf configuration entry
- Removed most attribute and object-class mappings, as the required attributes and object classes are supported by OpenLDAP
- Slapd.conf – no longer uses backend.schema, index, refint or memberOf configurations

Well...

- Samba does not stand still, and it is hard to keep up
- Samba switched to a multi-process model
- Samba implemented (and made default) an LMDB back-end
- Changes to LDB broke the ability to provision the legacy OpenLDAP back-end

Samba/AD Class definitions

```
objectclass (
2.5.6.14
 NAME 'device'
  SUP top
  STRUCTURAL
  MUST ( cn )
  MAY ( bootFile $ bootParameter $ cn $
description $ ipHostNumber $
   l $ macAddress $ manager $ msSFU30Aliases
$ msSFU30Name $ msSFU30NisDomain $
nisMapName $ o $ ou $ owner $ seeAlso $
serialNumber $ uid ))


extendedClassInfo: ( '2.5.6.14' NAME 'device'
CLASS-GUID
'8E7A96BFE60DD011A28500AA003049E2' )
```

```
cn: Device
ldapDisplayName: device
governsId: 2.5.6.14
objectClassCategory: 0
rdnAttId: cn
subClassOf: top
auxiliaryClass: ipHost, ieee802Device, bootableDevice
systemMustContain: cn
mayContain: msSFU30Name, msSFU30NisDomain, nisMapName,
msSFU30Aliases
systemMayContain: serialNumber, seeAlso, owner, ou, o, l
systemPossSuperiors: domainDNS, organizationalUnit,
organization,container
schemaIdGuid:bf967a8e-0de6-11d0-a285-00aa003049e2
defaultSecurityDescriptor: D:
(A;;RPWPCRCCDCLCLORCWOWDSDDTSW;;;DA)
(A;;RPWPCRCCDCLCLORCWOWDSDDTSW;;;SY)(A;;RPLCLORC;;;AU)
defaultHidingValue: TRUE
systemOnly: FALSE
defaultObjectCategory:
CN=Device,CN=Schema,CN=Configuration,<RootDomainDN>
systemFlags: FLAG_SCHEMA_BASE_OBJECT
```

Samba/AD Attribute definitions

```
attributetype (
  1.2.840.113556.1.4.656
  NAME 'userPrincipalName'
  EQUALITY caseIgnoreMatch
  SUBSTR caseIgnoreSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
  SINGLE-VALUE
  )
```

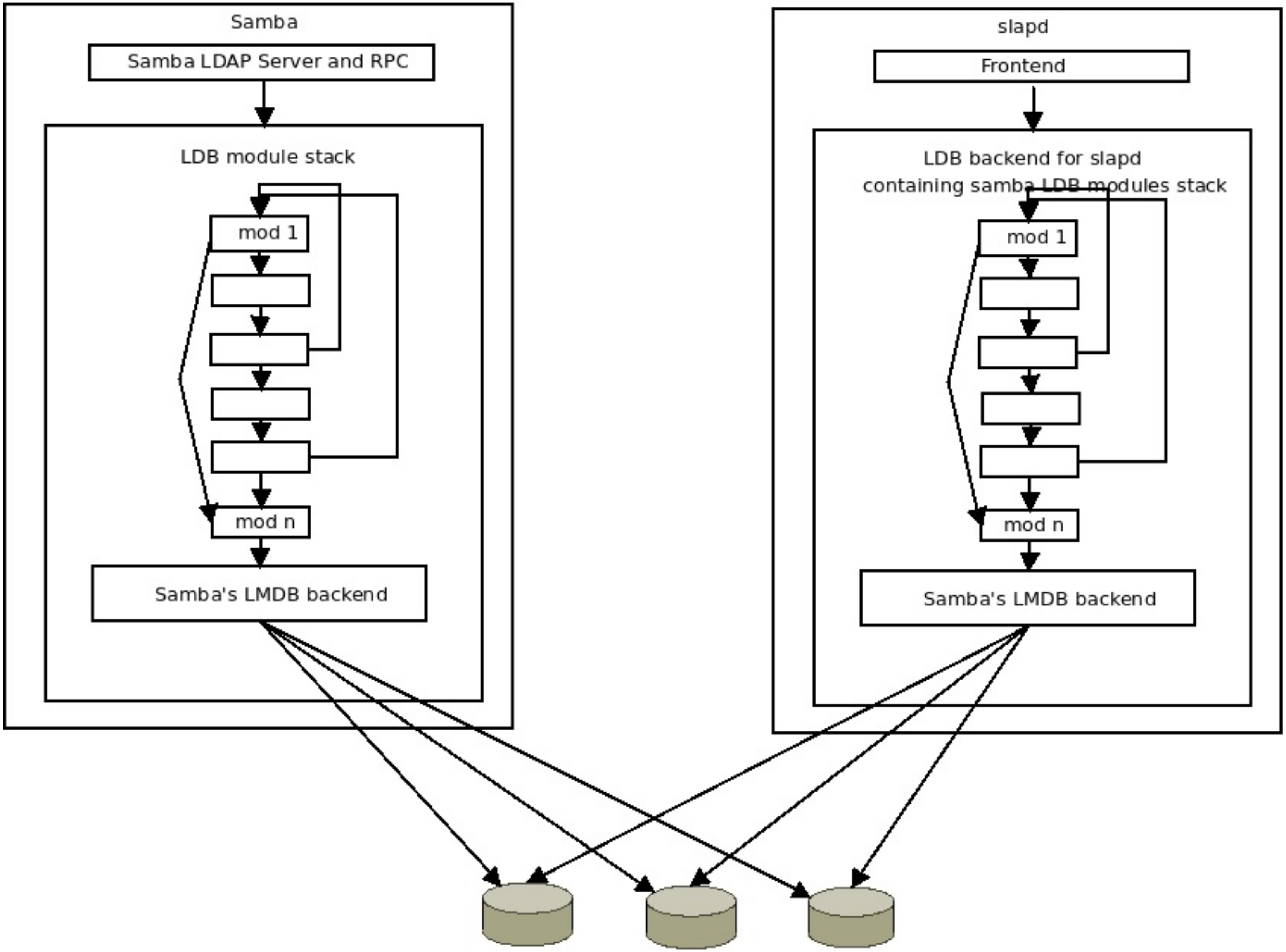
```
extendedAttributeInfo:
( '1.2.840.113556.1.4.656' NAME
'userPrincipalName' RANGE-UPPER '1024'
PROPERTY-GUID
'BB0E6328D541D111A9C10000F80367C1'
PROPERTY-SET-GUID
'54018DE4F8BCD111870200C04FB96050'
INDEXED )
```

```
cn: User-Principal-Name
ldapDisplayName: userPrincipalName
attributeId: 1.2.840.113556.1.4.656
attributeSyntax: 2.5.5.12
omSyntax: 64
isSingleValued: TRUE
schemaIdGuid: 28630ebb-41d5-11d1-a9c1-0000f80367c1
systemOnly: FALSE
searchFlags: fATTINDEX
rangeUpper: 1024
attributeSecurityGuid: e48d0154-bcf8-11d1-8702-00c04fb96050
isMemberOfPartialAttributeSet: TRUE
systemFlags: FLAG_SCHEMA_BASE_OBJECT | FLAG_ATTR_REQ_PARTIAL_SET_MEMBER
schemaFlagsEx: FLAG_ATTR_IS_CRITICAL
```

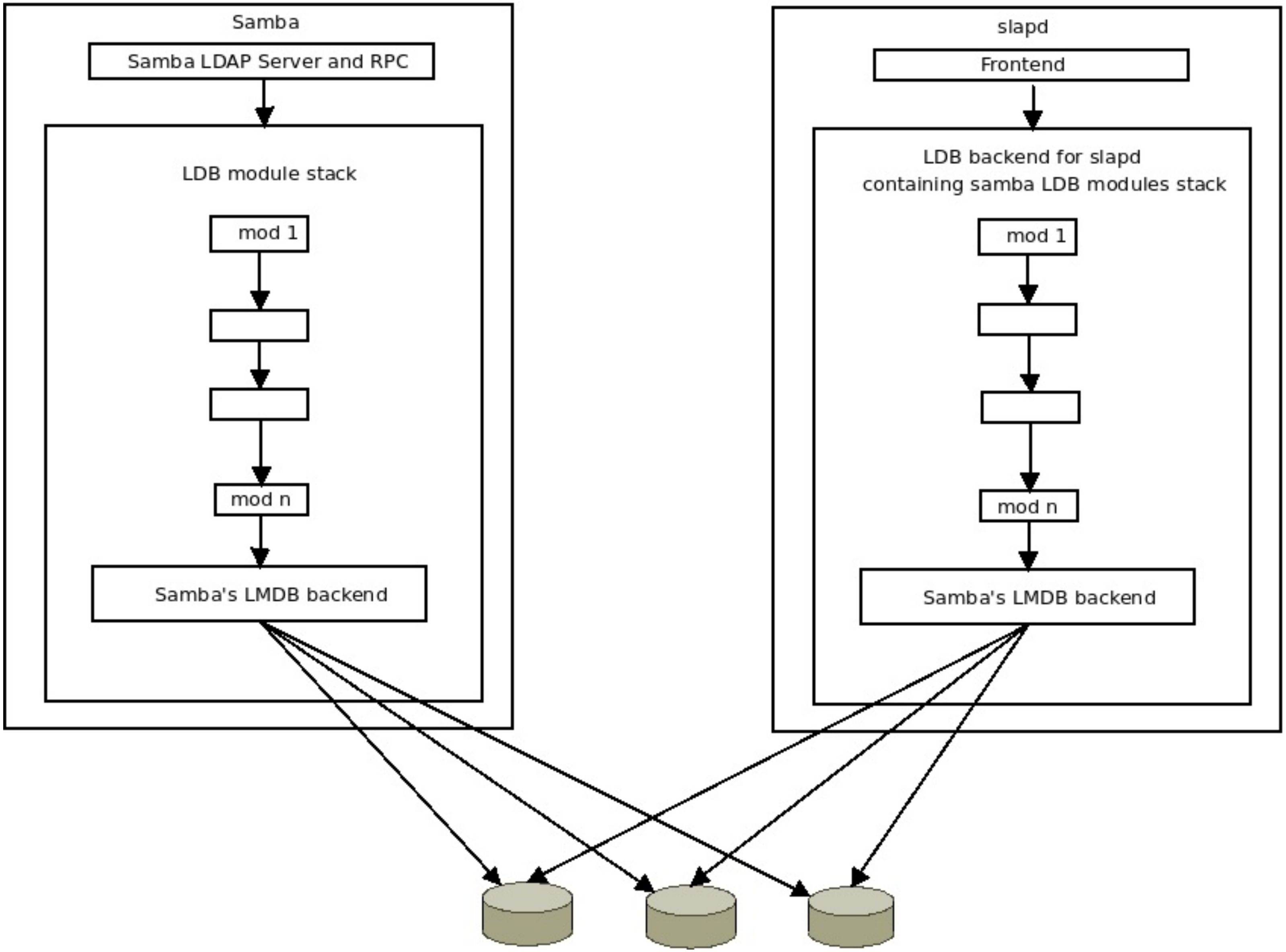
A Samba back-end for OpenLDAP

- Integrate the LDB module stack as an OpenLDAP backend
- Re-factor the LDB stack so that modules become truly independent
- Develop he ability to wrap LDB modules inside overlays, tuning the LDB stack into an overlay stack, while still using Samba code

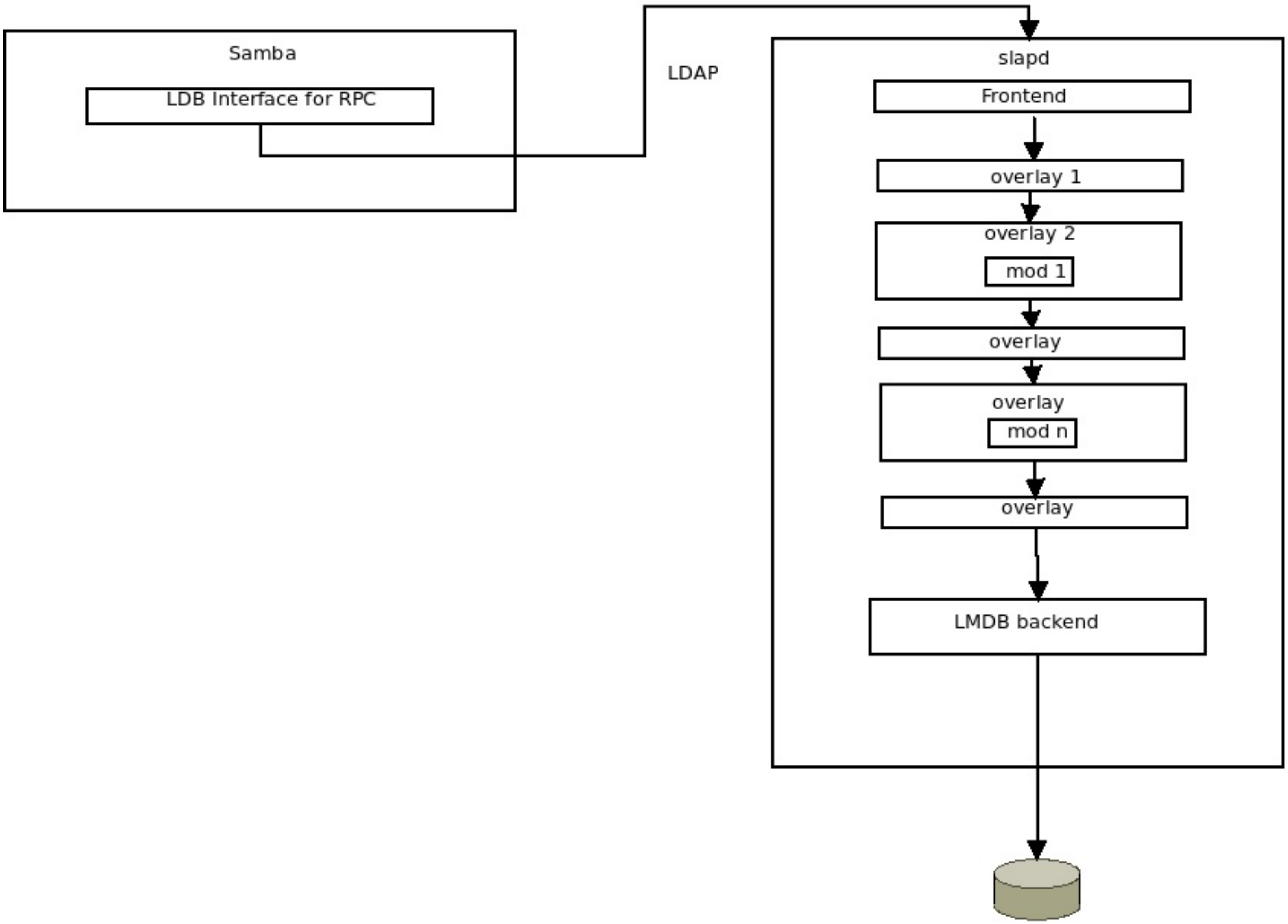
Step 1


Samba
Samba LDAP Server and RPC
LDB module stack
mod 1
mod n
Samba's LMDB backend
slapd
Frontend
LDB backend for slapd
containing samba LDB modules stack
mod 1
mod n
Samba's LMDB backend

Step 2


Samba
Samba LDAP Server and RPC
LDB module stack
mod 1
mod n
Samba's LMDB backend
slapd
Frontend
LDB backend for slapd
containing samba LDB modules stack
mod 1
mod n
Samba's LMDB backend

Step 3


Samba
LDB Interface for RPC
LDAP
slapd
Frontend
overlay 1
overlay 2
mod 1
overlay
overlay
mod n
overlay
LMDB backend

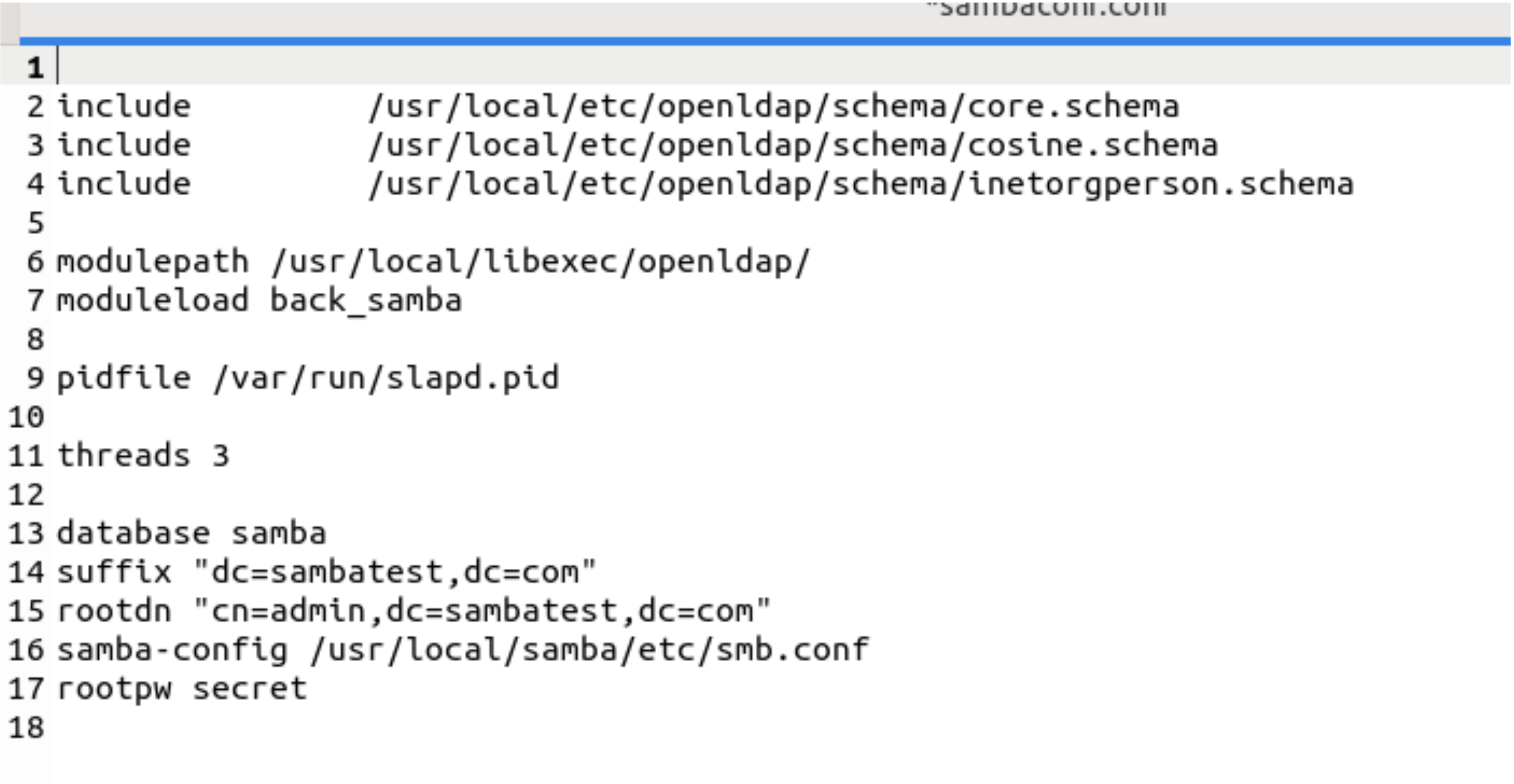
Why?

- It will lead to a much better integration of OpenLDAP and Samba
- Unless we move on to rewriting the LDB modules as overlays, we will be able to collaborate with the Samba Team in the support of the code
- We would be able to deliver a use-able solution faster
- Users will be able to use new Samba features faster
- Samba has an LMDB backend
- Some serious performance improvements in Samba
- Samba is now mature enough that module refactoring can be attempted

back-samba

- Loads a loadparm_context based on the smb.conf
- On bind, creates a system_session and an ldb context that is connection-specific(?)
- Connects to sam.ldb
- Implements LDAP operation handlers that translate OpenLDAP operations to ldb operations, using ldb_build_xxxxx_req
- Uses samba libraries   -lldb -ltalloc -lsamba-hostconfig -lcmdline-s4-samba4 -lsamdb -lsamba-sockets-samba4
- “lives” in contrib/slapd-modules (https://gitlab.symas.net/nivanova/back-samba)

slapd.conf



```

include          /usr/local/etc/openldap/schema/core.schema
include          /usr/local/etc/openldap/schema/cosine.schema
include          /usr/local/etc/openldap/schema/inetorgperson.schema

modulepath /usr/local/libexec/openldap/
moduleload back_samba

pidfile /var/run/slapd.pid

threads 3

database samba
suffix "dc=sambatest,dc=com"
rootdn "cn=admin,dc=sambatest,dc=com"
samba-config /usr/local/samba/etc/smb.conf
rootpw secret

```

Current workflow

- Build and install samba
- Build and install back-samba module for OpenLDAP
- Provision samba using samba-tool so that the databases are created
- Start OpenLDAP
- Perform LDAP requests using ldap tools or python scripts

Demo… ish?

When your program is a complete mess, but it does its job

Next todo’s

- RootDSE
- Schema
- Authentication (gensec?)
- Better memory management – avoid memory duplication if possible
- Bench-marking strategies - how much “better” is enough?

There be dragons

LDB THREAD SAFETY
PROPER AUTHENTICATION
THINGS WE HAVEN'T EVEN CONSIDERED...
imgflip.com

Suggestions are welcome and appreciated!

(nivanova@samba.org)