
# An OpenLDAP backend for Samba 4

Nadezhda Ivanova
Software Engineer @ Symas Corp

symas

# About Samba4

- Combines the file sharing service of Samba with a fully AD compatible Domain controller
- Can be a standalone Domain Controller
- Can join an existing Windows Active Directory domain as a member server, or an RODC
- Supports all FSMO roles
- Domain member machines work with Samba4 transparently
- Management can be done both with samba-tool and by installing Microsofts RSAT (Remote Server Administration Tools) on a Windows machine.

symas

# About Samba4

- Released in 2013 after more than 10 years in development
- Successfully deployed by small to mid-sized companies
- Functionality is developed as separate modules
- Microsoft Open Specifications Program (as of 2007)

symas

# A little light reading...

- https://wiki.samba.org - detailed instructions on how to setup a Samba4 DC
- [MS-ADTS]: Active Directory Technical Specification
- [MS-DRSR]: Directory Replication Service (DRS) Remote Protocol
- Windows Protocols Technical Specifications https://msdn.microsoft.com/en-us/library/jj712081.aspx

symas

# Samba 4 functionality

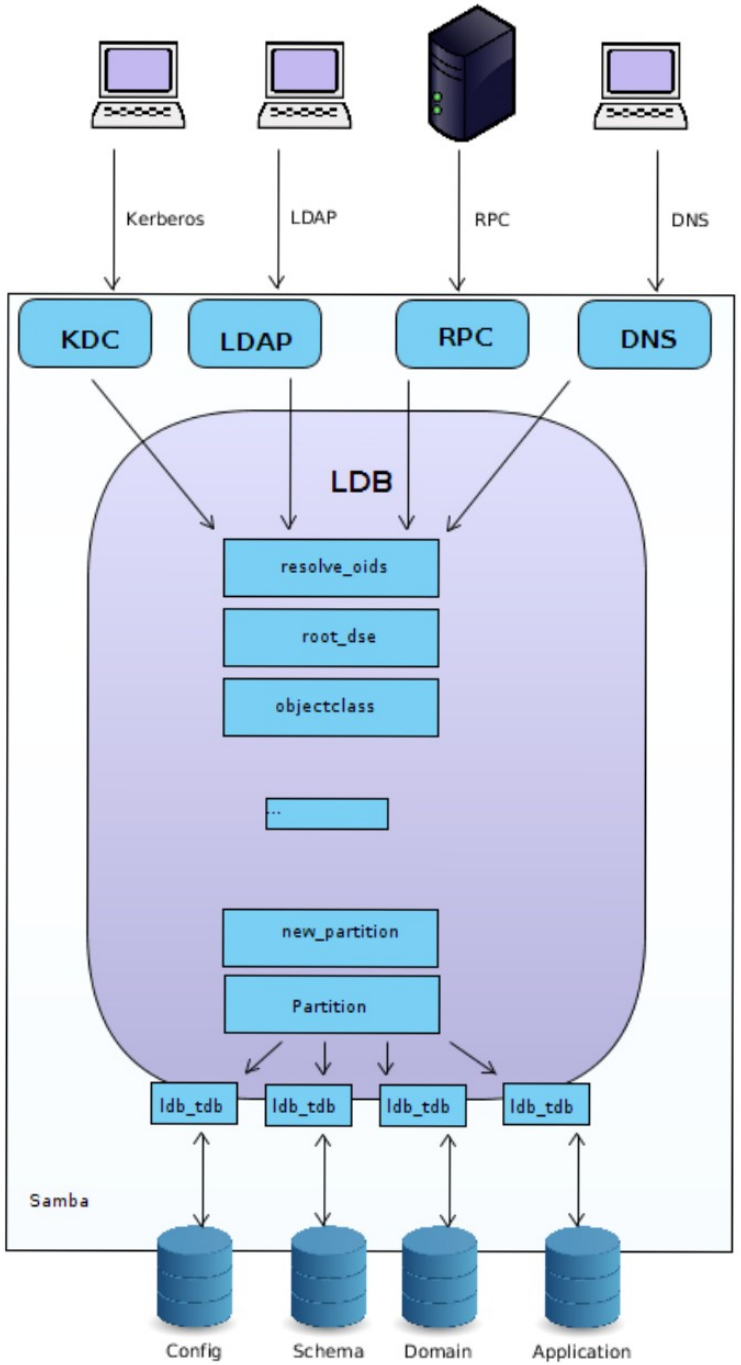
- LDAP – provides its own LDAP server, fully compatible with the AD flavor of LDAP and the AD schema.
- Kerberos KDC – integrated in Samba.
  - Heimdal Library
  - MIT Kerberos Library
- DNS
  - Internal Samba DNS
  - Bind
- RPC

symas

# RPC protocols

- Security Account Manager (SAMR)
- Local Security Authority (LSAR)
- DFSR – necessary to the AD compatibility because it is used to replicate Sysvol
- DRSR - Directory Replication Service – implements multi-master replication

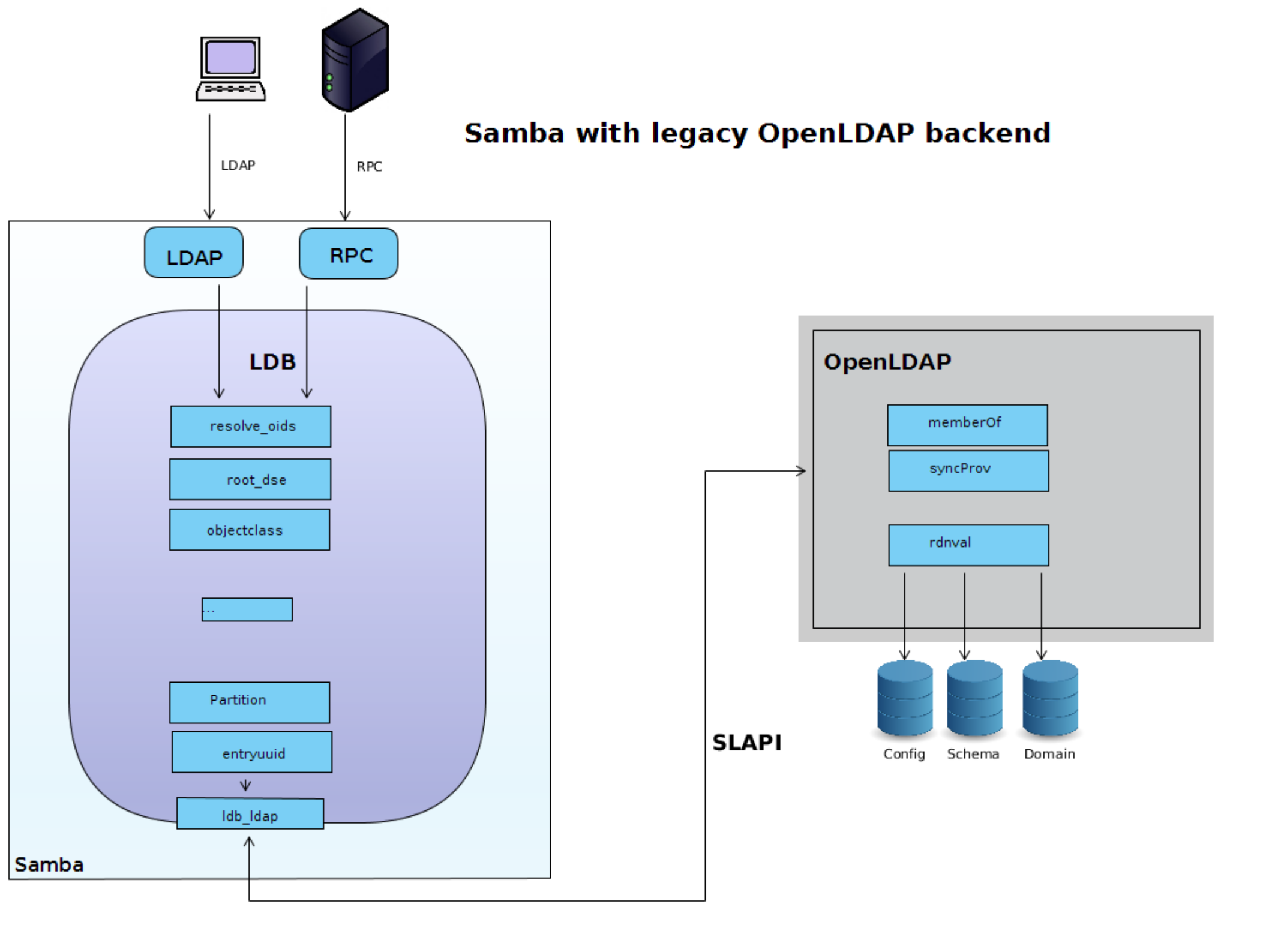
# Samba 4 with TDB



symas

# Problems of Samba 4 with TDB

- Scalability
  - Supported TDB version is 32 bit, which puts a 4GB limit on the database, equals around 300 000 objects depending on their size.
  - Work on the 64 bit is not progressing
- Performance
  - Initial Bulk load of 350.000 small User-Objects (LDIF, with unicodePwd) takes more than 6 hours on a real hardware machine.
  - The results are the same with direct LDB load, not dependent on network or protocol overhead.
  - A POC of MDB back-end for LDB was created by Jakub Hrozek, but oddly, it did not significantly improve performance.


Samba with legacy OpenLDAP backend
LDAP
RPC
LDAP
RPC
LDB
resolve_oids
root_dse
objectclass
...
Partition
entryuuid
ldb_ldap
Samba
SLAPI
OpenLDAP
memberOf
syncProv
rdnval
Config
Schema
Domain

# Samba provisioning with Legacy OpenLDAP

- Samba provisioning scripts creates slapd.conf
  - Only the basic partitions, no new partitions can be added
- Provisioning script creates a schema definition file for OpenLDAP
- Populates the created databases with the necessary initial data

symas

# Why not use the legacy OpenLDAP Back-end

- A “real” back-end – LDAP traffic goes through Samba, to make sure all the AD request processing specifics are implemented
- Incompatible with replication, as back then there was no transaction support
- Support was discontinued, since then Samba has made huge progress
  - Multi-master replication
  - DNS
- Conflicts with standard LDAPv3
  - Same attribute name, different OID
  - Object classes with changed definitions, attributes that in AD are operational
- This was resolved by adding additional modules to strip extended DN components, or to map attribute names
- Essentially, obsolete
- Would not solve all performance problems.
- Officially declared dead around 2010/2011

# 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 ) "

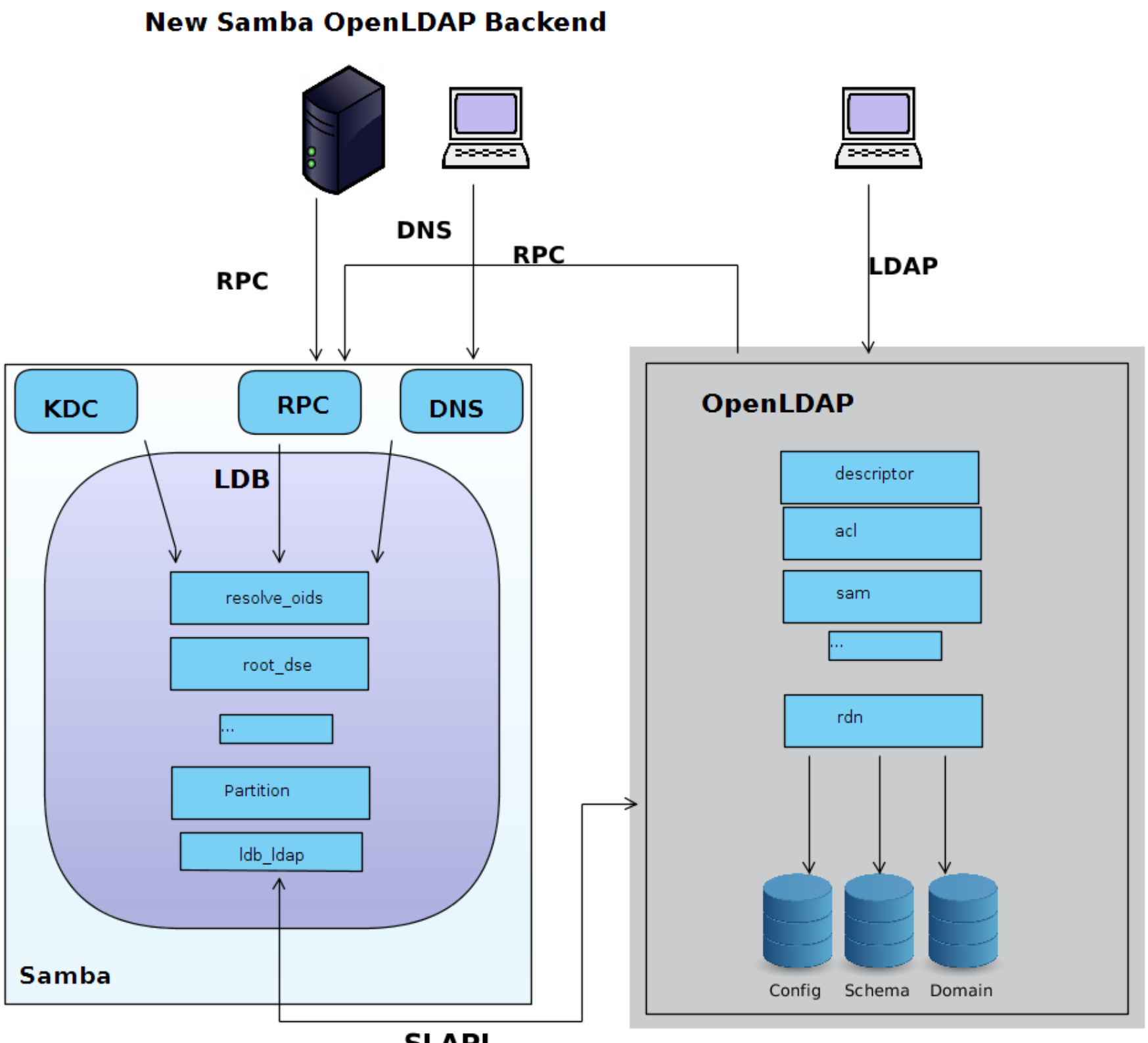
MAY ( instanceType $ nTSecurityDescriptor $ objectCategory $ adminDescription $ adminDisplayName $ allowedAttributes $ allowedAttributesEffective $ allowedChildClasses $ allowedChildClassesEffective $ bridgeheadServerListBL $ canonicalName $ cn $ description $ directReports $ displayName $ displayNamePrintable $ 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 ) )"

symas

symas

# More than a backend

- 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.


New Samba OpenLDAP Backend
DNS
RPC
RPC
LDAP
KDC
RPC
DNS
LDB
resolve_oids
root_dse
...
Partition
ldb_ldap
Samba
OpenLDAP
descriptor
acl
sam
...
rdn
Config
Schema
Domain
SLAPI

symas

# Challenges

- Ldb modules ≈ 40 000 lines of C
- We start by replacing individual modules, but:
  - Samba modules are interconnected and often communicate with each other via internal controls
  - Sometimes RPC traffic is initiated from inside a module, e.g samldb and replmetadata
- Alleviate the load by code reuse

symas

# Samba libraries in OpenLDAP

- Libclisecurity
  - SD generation
  - SDDL parsing
  - Access checks
- libsamba_schema
  - Additional schema data
  - Loading of AD schema LDIF

- libldb, libtalloc – necessary for the above

symas

# Work in progress

- Security descriptor generation
- Authorization
- InstanceType value checking
- Extended DN Control (<GUID=...>;<SID=...>;cn=Administrator)
- “Show Deleted” Control
- SAM – research phase
- A module to gather and maintain data necessary for request processing
- A module to load and maintain a Samba-type schema information

symas

# Operational attributes

- canonicalName
- primaryGroupToken
- tokenGroups
- parentGUID
- modifyTimestamp
- msDs-isRODC
- MsDS-userPasswordExpiryTime

# 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
  )
```

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

# 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 )
  )
```

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

symas

# Authorization

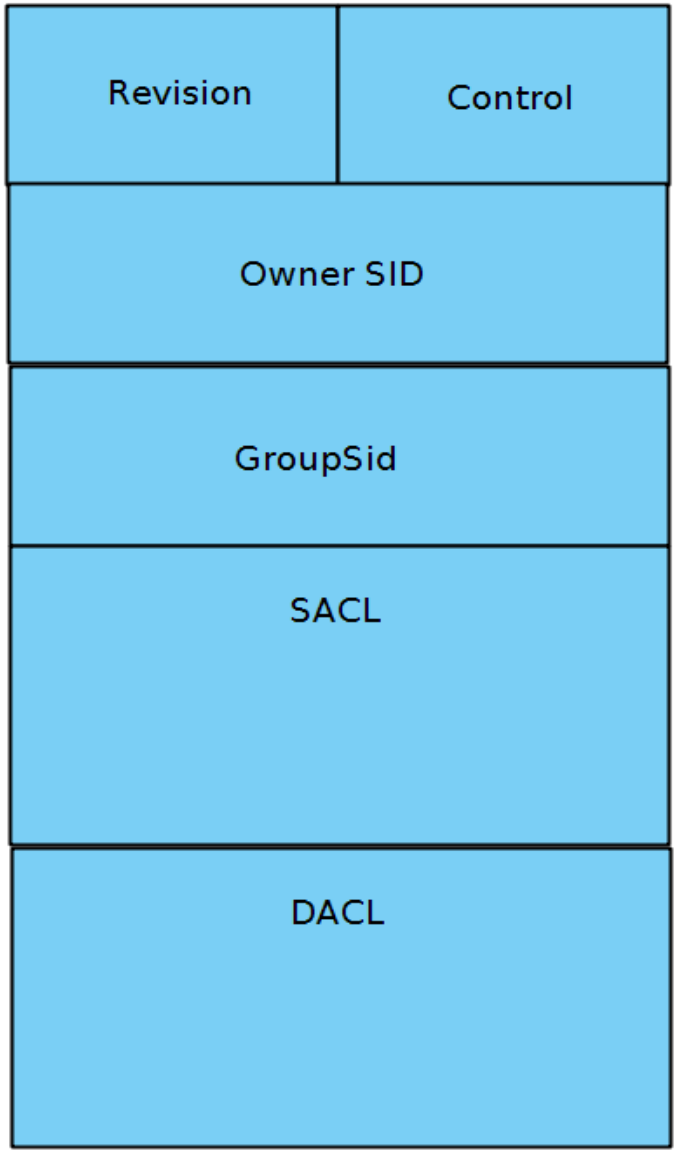
- Determines an account's rights over a specific object by comparing the security principal's security token with the object's security descriptor.
- Security token - a list of SIDs of every group the security principal is a member of, and the account SID

# Access Control Entries


- Grants particular access rights over the entire object, an object class or an attribute

symas

# Security descriptors



symas

# Calculating SD for a new object

- Input
  - SD of the parent container
  - SD provided by the client
  - Default SD (from defaultSecurityDescriptor attribute)
  - Session's security Token
- Output
  - Owner, Group, Explicit ACEs, Inherited ACEs

symas

# Required access for LDAP operations

- Search
  - LIST_CHLIDREN on the parent, READ_PROPERTY
- Add
  - CREATE_CHILD
- Modify
  - WRITE_PROPERTY
- Delete
  - DELETE_CHILD on the parent or DELETE on the object
- Rename
  - DELETE_CHILD on the parent, CREATE_CHILD on the new parent, WRITE_PROPERTY on the rdn attribute

symas

# Extended rights and Validated Writes

- ValidatedWrites – checks whether a user is allowed to enter an attribute value (e.g validateSPN)
- ExtendedRights – the rights to perform specific operations – e.g update the schema, modify or replicate from a replica, etc.

symas

# Constructed attributes

- AllowedAttributes – all attributes this object may have
- AllowedAttributesEffective – attributes that are permitted to be assigned to a class
- AllowedChildClasses – the particular object is a possible superior
- AllowedChildClassesEffective – the particular object is a possible superior AND the principal has the right to create child object of these classes
- sDRightsEffective

symas

# SAM

- Handles creation of objects that represent security principals
- Creates a SID for the new object
  - If we are not the RID master, initiates a RID pool allocation request
- Initializes user and group object attributes
- Handles userAccountControl

symas

# Next Steps

- Implement proper partition creation – this will allow proper provisioning and creation of application partitions.
- Reduce reliability on Samba libraries for reasons of performance.
- Incorporate schema data in OpenLDAP as part of the existing mechanism, rather than in a module.
- Finish porting the LDB module stack.
- Develop OpenLDAP to Samba communication mechanism – necessary for DRSR and SAM.

symas

# Testing

- Samba make test suite
  - Extensive coverage of LDAP functionality with Python Scripts
- Microsoft Documentation test suite
  - Developed to test documentation consistency
  - Very helpful in ensuring implementation compatibility

symas

# FAQ

- Is this a new version of Samba3 with an OpenLDAP domain controller?
- Will I be able to integrate an existing non-AD directory in an OpenLDAP server running in AD compatibility mode?
- Will I be able to combine using LDAP access lists with the AD access lists?

symas