An OpenLDAP backend for Samba 4

Nadezhda Ivanova
Software Engineer @ Symas Corp

symas

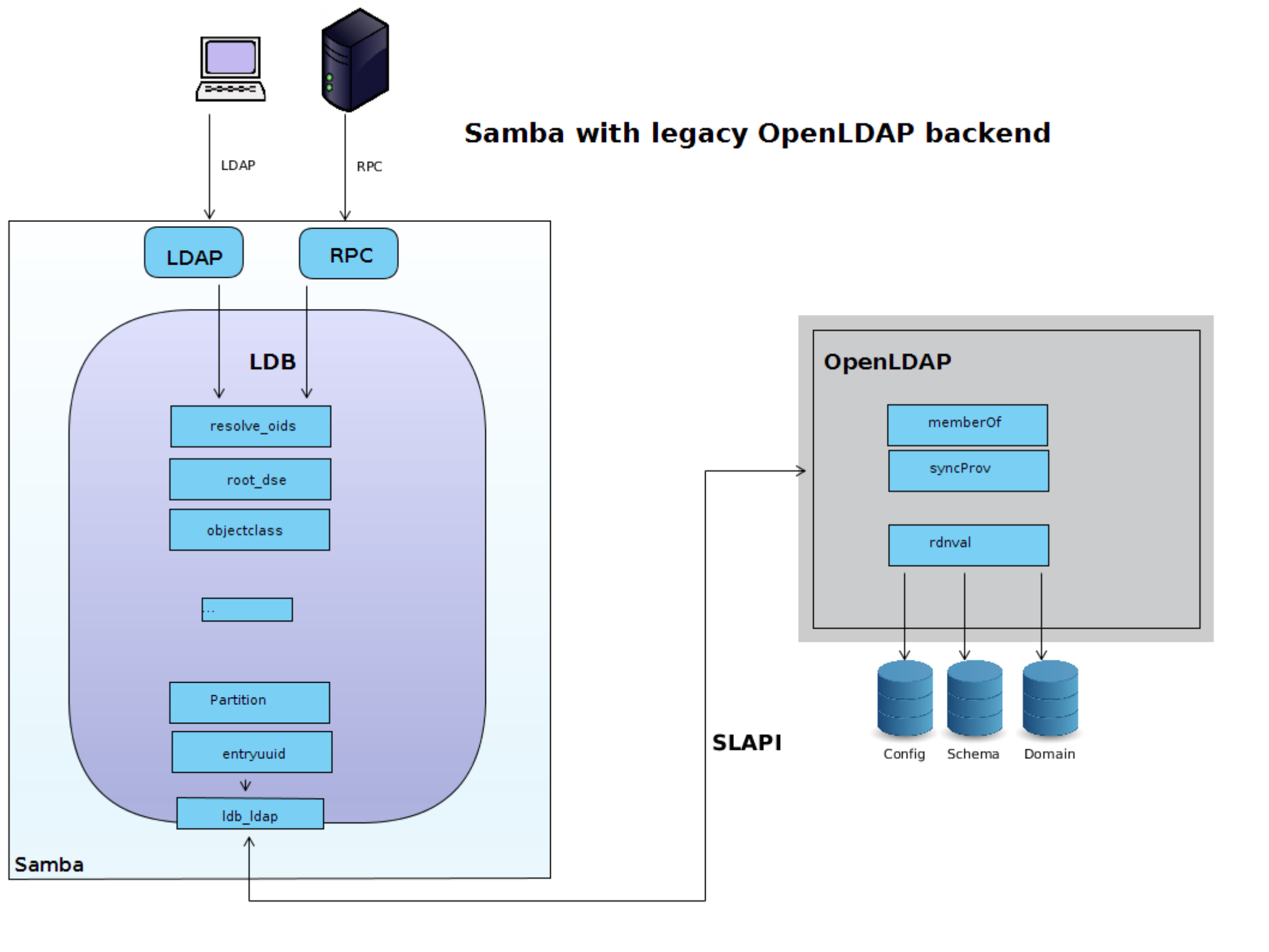
About Samba4

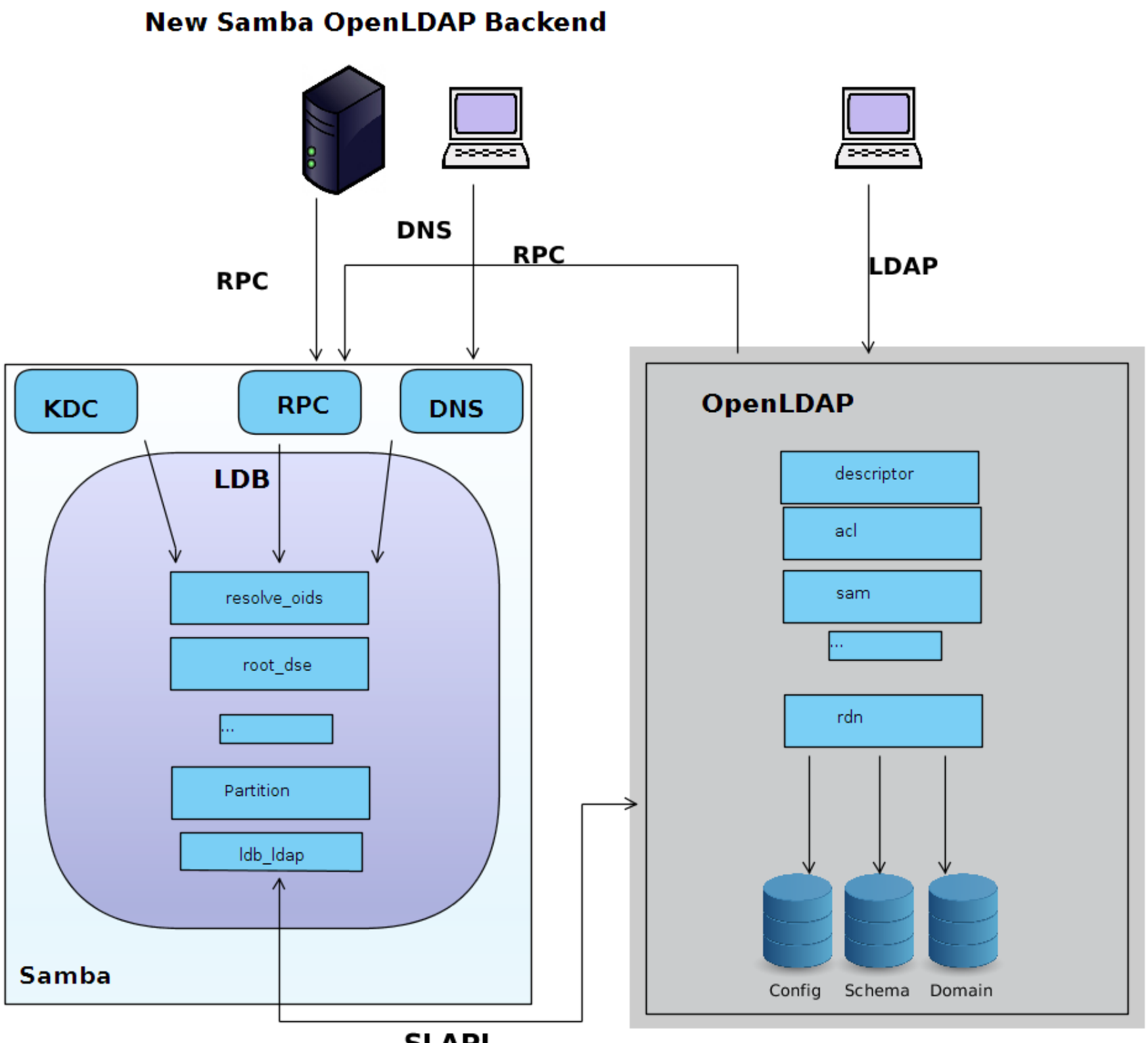
- Combines the file sharing service of Samba with a fully AD compatible Domain controller
- Can be a standalone Domain Controller
- Can join an existing Windows Active Directory domain as a member server, or an RODC
- Supports all FSMO roles
- Domain member machines running Windows work with Samba4 transparently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

symas

About Samba4

- Released in 2013 after more than 10 years in development
- New releases added every 6-9 months
- Successfully deployed by small to mid-sized companies
- Functionality is developed as separate LDB modules, similar in structure to OpenLDAP overlays
- Has its own internal DNS server, or can work with BIND9 using a BIND_DLZ module.
- Has its own Kerberos KDC
- Used to have an OpenLDAP back-end, which got retired due to lack of resources and technical difficulties


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


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

Implementation approach

- We started by replacing individual modules, but:
  - Samba modules are interconnected and often communicate with each other via internal controls
  - 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
- Switch to separate implementation of functionality within OpenLDAP, with manual testing via OpenLDAP directly, until LDAP behavior is as desired
- Determine how and if to remove or modify Samba modules later, after RPC tests

symas

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”

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))"

Samba provisioning with Legacy OpenLDAP

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

symas

ad_schema overlay

- Registers the attributeSchema and classSchema attributes in OpenLDAP schema
- On LDAP_ADD, from the incoming entry:
  - 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

symas

ad_schema overlay – cont.

- LDAP_SEARCH – creates the values for attributeInfo, classInfo, extendedAttributeInfo and extendedClassInfo for the subschema (cn=Aggregate, cn=Schema)
- On db_open, loading of the schema data from the database, as it is no longer stored in a file (WIP)
- TODO – implement schema modification restrictions – such as modifying base schema objects, consistency checks, etc.

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

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

symas

Syntax mapping

attributeSyntax	oMSyntax	oMObjectClass	LDAP OID
2.5.5.8 (Boolean)	1		1.3.6.1.4.1.1466.115.121.1.7
2.5.5.9 (Integer)	2, 10		1.3.6.1.4.1.1466.115.121.1.27
2.5.5.16 (Large Int)	65		1.2.840.113556.1.4.906
2.5.5.11 (UTC-Time)	23		1.3.6.1.4.1.1466.115.121.1.53
2.5.5.11(Generalized-Time)	24		1.3.6.1.4.1.1466.115.121.1.24
2.5.5.14(DN-String)	127	0x2A 0x86 0x48 0x86 0xF7 0x14 0x01 0x01 0x01 0x0C	1.2.840.113556.1.4.904
2.5.5.14(Access-Point)	127	0x2B 0x0C 0x02 0x87 0x73 0x1C 0x00 0x85 0x3E	1.3.6.1.4.1.1466.115.121.1.2

symas

Indexing and linked attributes

- Linked attributes are distinguished by the linkID attribute in their attributeSchema entry.
- LinkID%2 = 0 – forward link, forward link +1 back link
- searchFlags: fATTINDEX – the attribute is indexed

symas

Constructed and Operational attributes

- An attribute that is returned only when requested by name (systemFlags:FLAG_ATTR_IS_OPERATIONAL)
- Constructed – FLAG_ATTR_IS_CONSTRUCTED
- Special Attributes - not defined as operational or constructed but behave as such, or are created by the DS during object creation, yet not defined as constructed
- Secret attributes – a “hard-coded” list of attributes that is never exposed via LDAP

symas

objectguid

- Constructs and adds the objects':
  - GUID – a randomly generated unique identifier got the object
  - SID – Security Identifier of a security principal.
  - InstanceType
  - WhenCreated, whenChanged

symas

Demo

symas

Active Directory Authorization Data

- Security Principal (securityPrincipal) – any entity that needs access to an object or resource. Can be a user, group, or a computer account. Uniquely identified bu a Security Identifier (SID).
- Security context – a list of SIDs for all groups that the principal is member of, + the principle's own SID
- Security token – A data structure, representing the security context of a principle (MS_DTYP).
- Security Descriptor – Contains access control information about the object that it is associated with – nTSecurityDescriptor atribute of an object.

symas

sectoken

- LDAP_BIND
  - Retrieve the user entry, and for each value of its memberOf attribute, retrieve the SID and add it to the list of SIDS
  - Do the above recursively
  - Set the Privileges flags
  - Attach the resulting structure to the ConnExtra of the connection
- LDAP_UNBIND – destroy the security token structure

Security Descriptor

dn: CN=Users

O:DA

G:DA

D:AI(A;;RPWPCRCCDCLCLORCWOWDSDDTSW;;;SY)

(A;;RPWPCRCCDCLCLORCWOWDSW;;;DA)

(OA;;CCDC;bf967aba-0de6-11d0-a285-00aa003049e2;;AO)

(OA;;CCDC;bf967a9c-0de6-11d0-a285-00aa003049e2;;AO)(

(A;;RPLCLORC;;;AU)

(OA;CIIOID;RP;b7c69e6d-2cc7-11d2-854e-00a0c983f608;bf967a86-0de6-11d0-a285-00aa003049e2;ED)

(OA;CIID;RPWPCR;91e647de-d96f-4b70-9557-d63ff4f3ccd8;;PS)
(A;CIID;RPWPCRCCDCLCLORCWOWDSDDTSW;;;EA)
(A;CIID;RPWPCRCCLCLORCWOWDSDSW;;;BA)

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

secdescriptor

- LDAP_ADD
  - 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.
- LDAP_MODIFY
  - Recalculates the SD's of the modified object and all of its children.
- LDAP_MODRDN
  - Similar to LDAP_MODIFY, still work in progress.
- LDAP_SEARCH
  - Handles the sDFlags control

symas

Required access for LDAP operations

- Search
  - LIST_CHILDREN 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

Some changes to Samba

- 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

symas

Next Challenges

- Full implementation of the SAMDB
- Kerberos authentication of OpenLDAP users using the Samba KDC
- “Catching up” with Samba – changes to the way LDB uses event contexts, and changes to the Samba process model

symas