IBM Education Assistance for z/OS V2R2

Items: Activity Log Enhancements
Compatibility Level Upgrade Without LDAP Outage
Dynamic Group Performance Enhancements
Replication of Password Policy Attributes from Read-Only Replica

Element/Component: IBM Tivoli Directory Server for z/OS (LDAP)

Material current as of May 2015

Agenda

- Trademarks
- Presentation Objectives
- For each of the enhancements, as needed:
  - Overview
  - Usage & Invocation
  - Migration & Coexistence Considerations
- Presentation Summary
- Appendix

Trademarks

Presentation Objectives

- Activity log enhancements: This provides additional data in the logs, specifically requested by customers
- LDAP Server compatibility level updates without an outage: This is related to a specific customer request to avoid a sysplex wide outage when enabling new capabilities in the LDAP servers in the sysplex group.
- Dynamic group performance enhancements: In prior releases, group determination for a user did not scale well when the directory contained thousands of dynamic groups. This is a scalability/performance improvement which allows wider use of dynamic groups in large directories through indexing.
- Replication of password policy operational attributes from a read-only replica: In prior releases, using advanced replication and password policy together, updates to user entries which occurred during failed authentication on read-only replicas were not reflected throughout the replication topology. This caused out of sync conditions and allowed users the ability to exceed policy restrictions by accessing multiple replicas. V2R2 provides an improvement to drive the updates to the supplier and throughout the topology.

Overview: Activity Log Enhancements

- Problem Statement / Need Addressed
  - Customers want better tracking of requests made to z/OS LDAP for either auditing purposes or problem determination
  - The activity log enhancement should not break the existing customer utilities by introducing changes to the existing records
- Solution
  - Add new activity log record types for previously unrecorded LDAP requests or events.
  - Add useful information to existing activity log records
  - Add an option in configuration file for activity log file version to provide compatibility with current version and future enhanced versions.
  - Add configuration options for each of the existing activity log environment variables
- Benefit / Value
  - Customer can get more useful information from the activity log for auditing or problem determination
  - z/OS LDAP can enhance the activity log without impacting existing activity log function
  - Customer can configure activity log more easily

Usage & Invocation: Activity Log Enhancements

- The activity log records client activities which can be analyzed to determine the client operations handled by the server. The activity log can contain information about operations handled by the server, client IP addresses, messages generated by the server, and hourly activity summary statistics. The LDAP server activity logging support has a number of features that allow the LDAP root administrator to customize the client activity to be logged.

- The existing configuration for activity log
  - Configuration file options
    logfile
    logFileFilter
    logFileRolloverTOD
    logFileRolloverDirectory
    logFileRolloverSize

  - Environment Variables

GLDLOG_MICROSECONDS	{ON | *anything_else*}
GLDLOG_MSG	{MSGS | NOMSGS}
GLDLOG_OPS	{WRITEOPS | ALLOPS | SUMMARY}
GLDLOG_TIME	{TIME | NOTIME | MERGEDRECORD}

Usage & Invocation: Activity Log Enhancements

- New Activity Log features: (these pertain to the new V1 format records)
  - new records for connection start and end
  - existing add request records changed to include the attribute names in the request
  - existing modify request records changed to include the attribute name and an indication of it being added, deleted or replaced
  - existing compare request records changed to include pwdpolicy update indicator
  - new record for abandon requests
  - existing request records changed to include the msgid so that abandon requests can be related to the request they affect
  - new record for unknown requests

- New activity log configuration options

– logFileVersion **1**	indicates new format of records, with expanded information
– logFileRecordType	replaces GLDLOG_TIME environment variable
– logFileMicroseconds	replaces GLDLOG_MICROSECONDS environment variable
– logFileMsgs	replaces GLDLOG_MSG environment variable

- SMF auditing related updates (for consistency)
  - provide the same events in activity logging and SMF auditing
  - modify the RACF SMF unload utility to support a new abandon event type

Usage & Invocation: Activity Log Enhancements

- New records for connection start and end events
  - Past reporting of binds and unbinds does not properly identify client connect and disconnect events
- Merged and non-merged record types are supported. Examples

Thu Oct 8 08:37:59 2009 mergedRecord **Connect**: connid = 1374, clientIP = 1.2.3.4, listen = ldap://[fe00::f4f7:0:0:7442:7510]:389

Thu Oct 8 08:50:59 2009 mergedRecord **Disconnect**: connid = 1374, clientIP = 1.2.3.4, bind = 'cn=john', cause = 1

Thu Oct 8 08:43:43.424715 2009 **Connect**: connid = 1, listen = ldap://[fe00::f4f7:0:0:7442:7510]:389, IP = 1.2.3.4,

Thu Oct 8 08:49:45.424716 2009 **Disconnect**: connid = 1, DN = 'cn=john', cause = 1, IP = 1.2.3.4

Usage & Invocation: Activity Log Enhancements

- Include “msgid” field in all LDAP request records:
  - “msgid” (along with “connid”) helps to macth up with abandon requests, when present.
- New “attrs” field on ADD and MODIFY records
  - “attrs” is a string of attribute names separated by spaces
- Examples

Thu Oct 8 08:49:45.424716 2009 End **Add**: connid = 1, DN = cn=john, rc = 0, **msgid = 789, attrs = cn objectclass description,** IP = 1.2.3.4

Thu Oct 8 08:37:59 2009 mergedRecord **Add**: connid = 1374, clientIP = 1.2.3.4, bind = 'cn=john', rc = 0, time = 49268usec, controls = 1.3.6.1.4.1.42.2.27.8.5.1, target = ou=zos,o=ibm,c=us, **msgid = 789, attrs = cn objectclass description,** rsn = NA

Thu Oct 8 08:49:45.424716 2009 End **Modify**: connid = 1, DN = 'cn=john,ou=zos,o=ibm,c=us', rc = 0, **msgid = 789, attrs = -telephone +mobilephone !socsecuritynumber,** IP = 1.2.3.4

Thu Oct 8 08:37:59 2009 mergedRecord **Modify**: connid = 1374, clientIP = 1.2.3.4, bind = 'cn=john,ou=zos,o=ibm,c=us', rc = 0, time = 49268usec, controls = 1.3.6.1.4.1.42.2.27.8.5.1, target = 'ou=zos,o=ibm,c=us', **msgid = 789, attrs = -telephone +mobilephone !socsecuritynumber,** rsn = NA

Usage & Invocation: Activity Log Enhancements

- Add new record for abandon request:
  - “msgid” is the message identifier of the abandon request itself
  - “targetmsgid” is the message identifier of the request to be abandonded
  -

- Examples
  -

```
Thu Oct 8 08:43:43.424715 2009 Abandon: connid = 1, msgid = 458, targetmsgid = 456, IP = 1.2.3.4

Thu Oct 8 08:49:45.424716 2009 End Abandon: connid = 1, msgid = 458, targetmsgid = 456, IP = 1.2.3.4


Thu Oct 8 08:37:59 2009 mergedRecord Abandon : connid = 1374, clientIP = 9.12.47.67, time = 711usec,
msgid = 458, targetmsgid = 456
```

Usage & Invocation: Activity Log Enhancements

- SMF Audit record 83 subtype 3 Changes:
  - New “abandon” request record
    - Common relocates (100-114) 201-207 222
    - new relocate 222
  -
  - New “LDAP_DISCONNECT_CAUSE” relocate for “disconnect” request record

DISCONNECT_TIMEOUT	0x00000001
DISCONNECT_NIC_FAILED	0x00000002
DISCONNECT_CLOSE_ABNORMAL	0x00000004
DISCONNECT_CLOSE_CLIENT	0x00000008
DISCONNECT_MESSAGE_INVALID	0x00000010
DISCONNECT_ERROR_INTERNAL	0x00000020
DISCONNECT_ERROR_WRITE	0x00000040
DISCONNECT_ERROR_SSL	0x00000080

Overview: Compatibility Level Upgrade Without LDAP Outage

- Problem Statement / Need Addressed
  - The server compatibility level is an LDAP server configuration option: serverCompatLevel.
  - This option is used to limit the functions supported by the LDAP server, so that new version server can work together with old version server within a sysplex group.
  - All LDAP servers must run at the same server compatibility level within a sysplex group.
  - All LDAP servers must have the same backend settings within a sysplex group.
  - Changing compatibility level or backend setting requires a sysplex-wide LDAP outage.

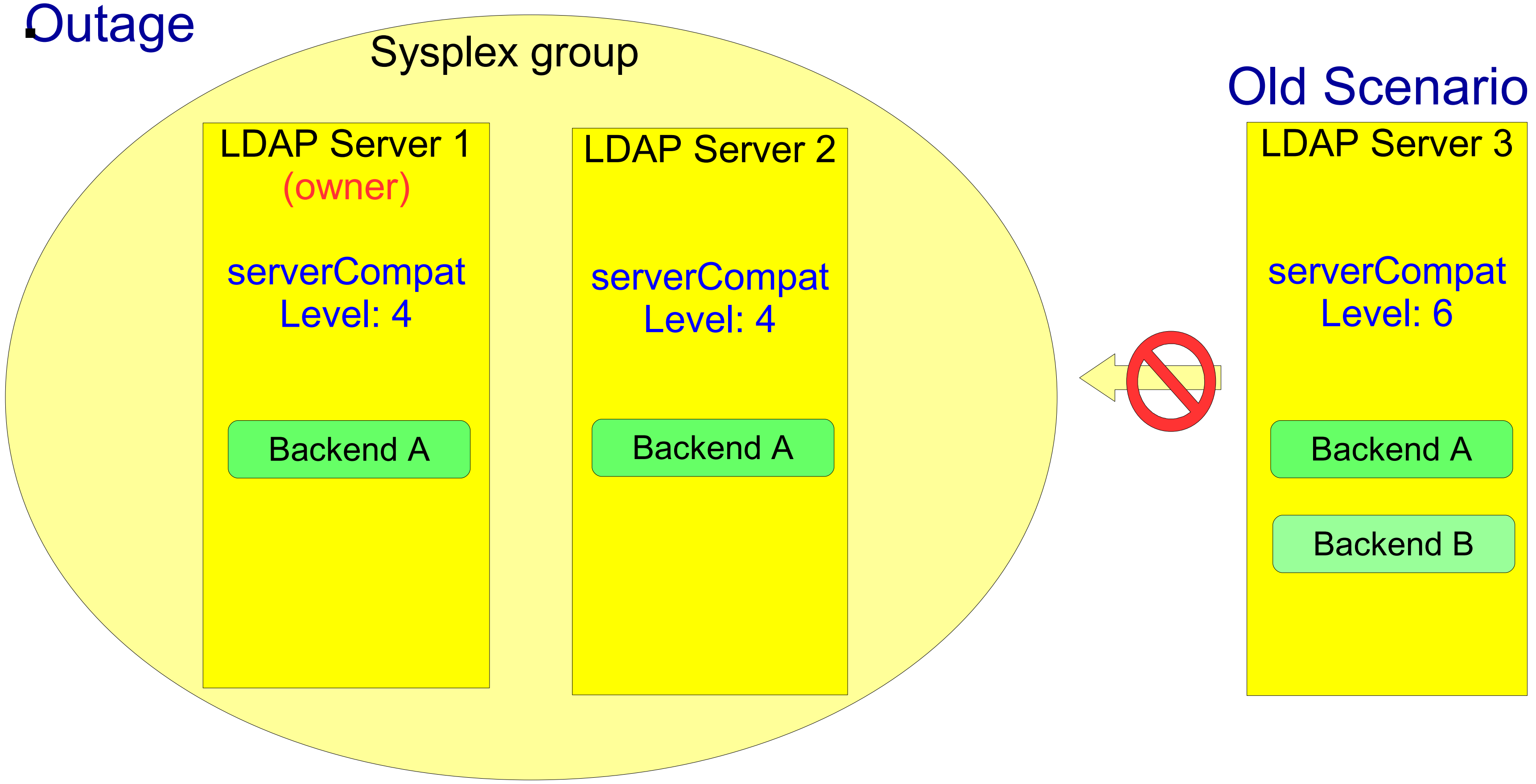
- Solution
  - Add a new run mode (transition mode) to use during migration to a new level

- Benefit / Value
  - Upgrade/downgrade compatibility level without full outage.
  - Change backend setting without full outage.

Usage & Invocation: Compatibility Level Upgrade Without LDAP Outage

- Create a copy of sysplex configuration and update serverCompatLevel and / or backend settings for transition server, then start it with “-T” parameter to join the sysplex in transition mode.
- Transition server first retrieves sysplex owner's compatibility level and backend settings for local validation, then uses the owner's level to initialize its backends. Any backend not defined on sysplex owner is flagged as private and initialized in a special way.
- Transition server operates at sysplex owner's level until it becomes the sysplex owner. Terminating all the other instances in the sysplex triggers transition server to complete transition by applying the new compatibility level and refreshing backends. Finally, transition server becomes a normal sysplex owner, making sysplex available for replicas with the same setting to join.
- It's recommended to restart the previous transition server after instances with new configuration join the sysplex.

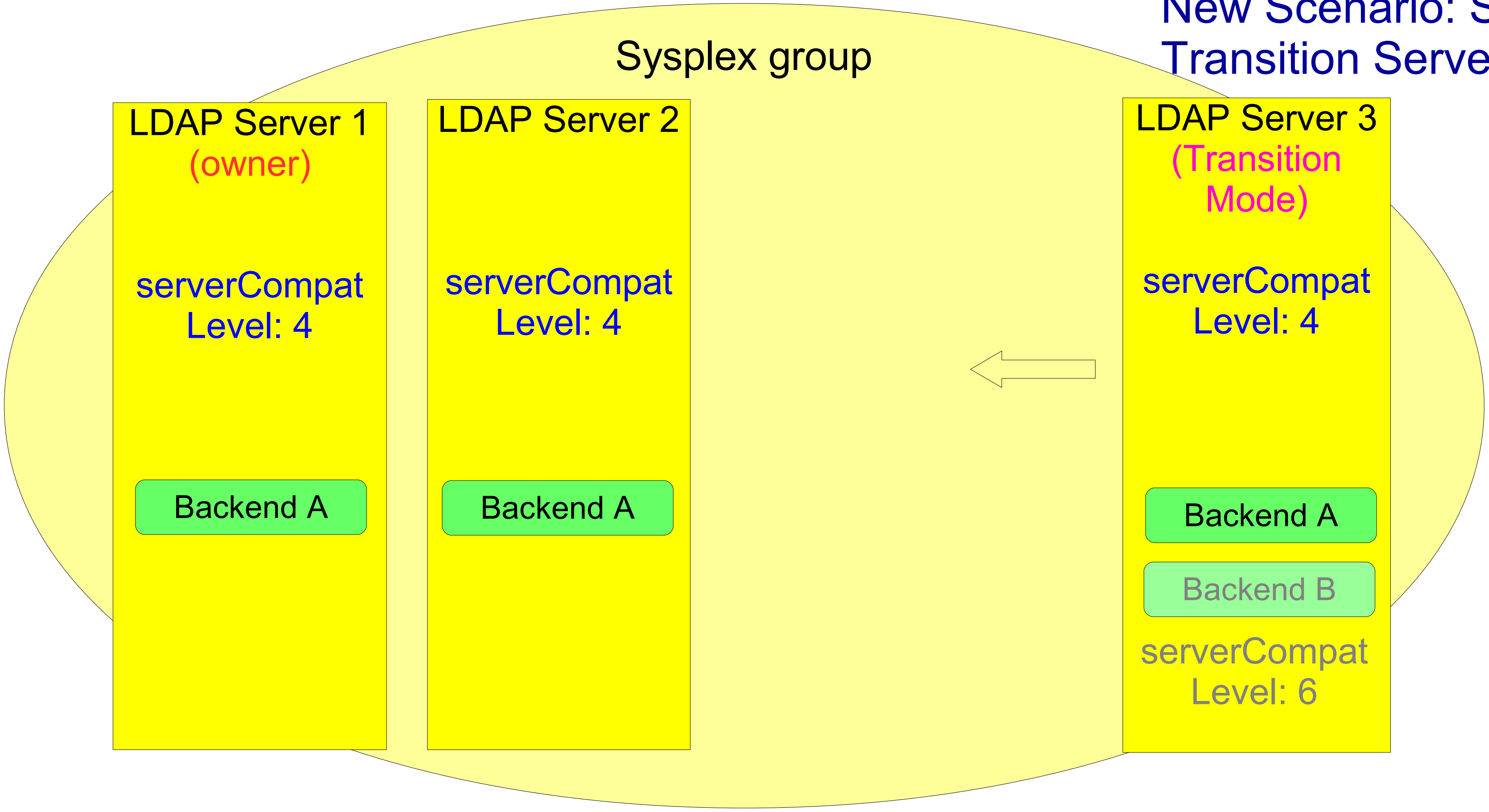
Usage & Invocation: Compatibility Level Upgrade Without LDAP Outage



When LDAP servers in a sysplex group are started, the first server to start becomes the sysplex owner of all the shared backends, including the schema.

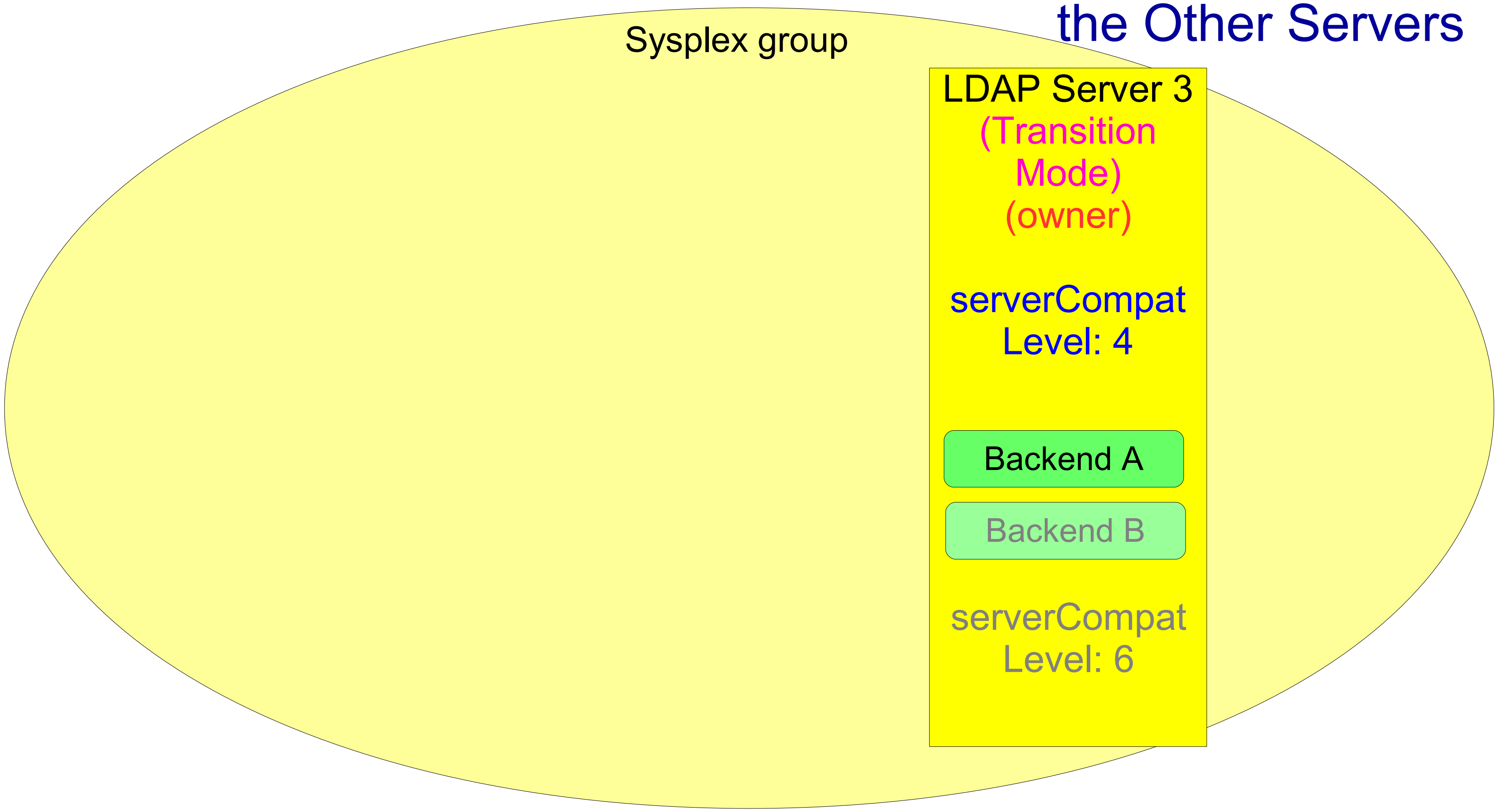
Usage & Invocation: Compatibility Level Upgrade Without LDAP Outage

New Scenario: Start Transition Server



Usage & Invocation: Compatibility Level Upgrade Without LDAP Outage

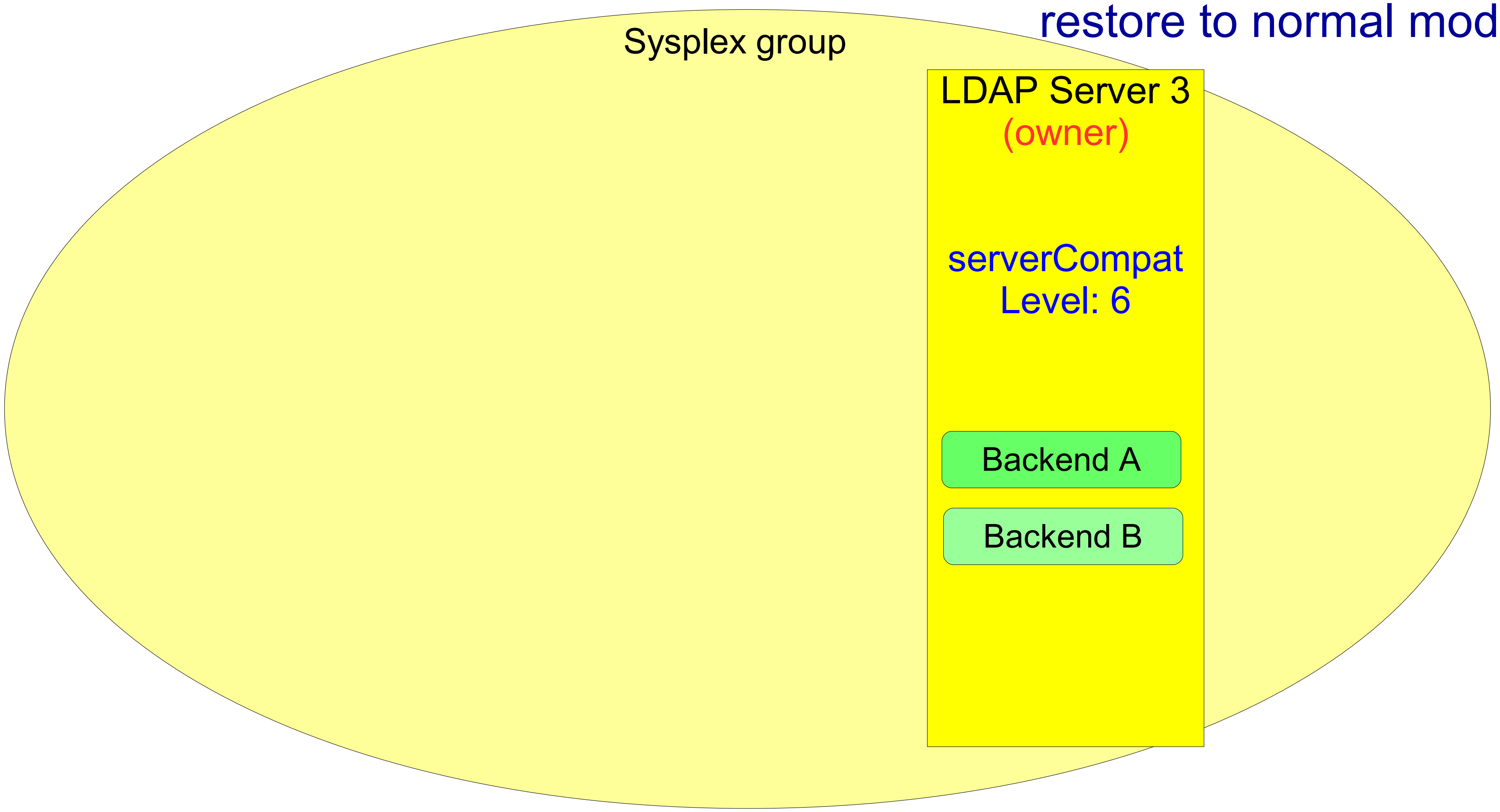
New Scenario: Shut down the Other Servers



If the sysplex owner ends, another LDAP server in the sysplex group becomes the owner.

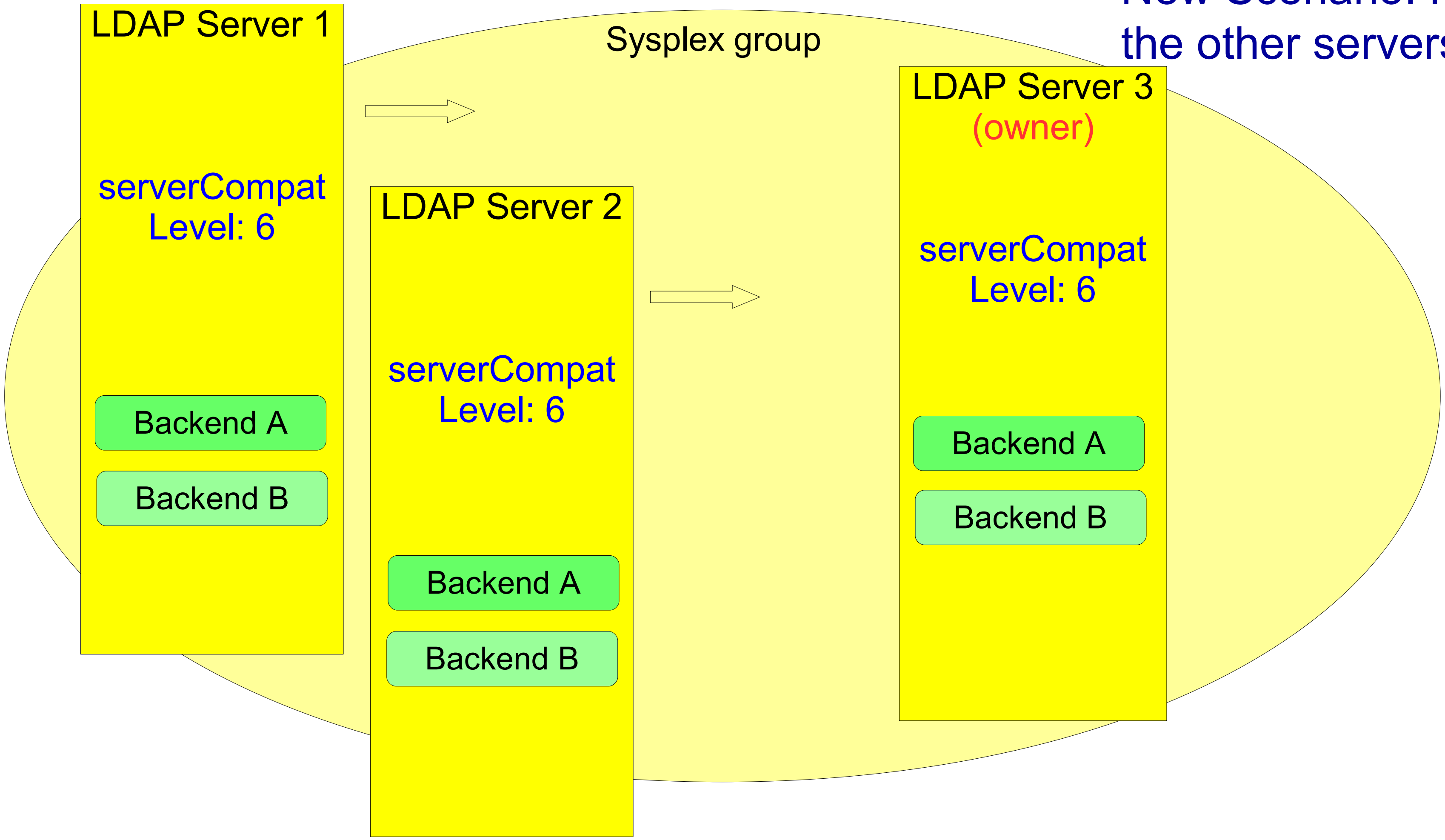
Usage & Invocation: Compatibility Level Upgrade Without LDAP Outage

New Scenario: Upgrade and restore to normal mode



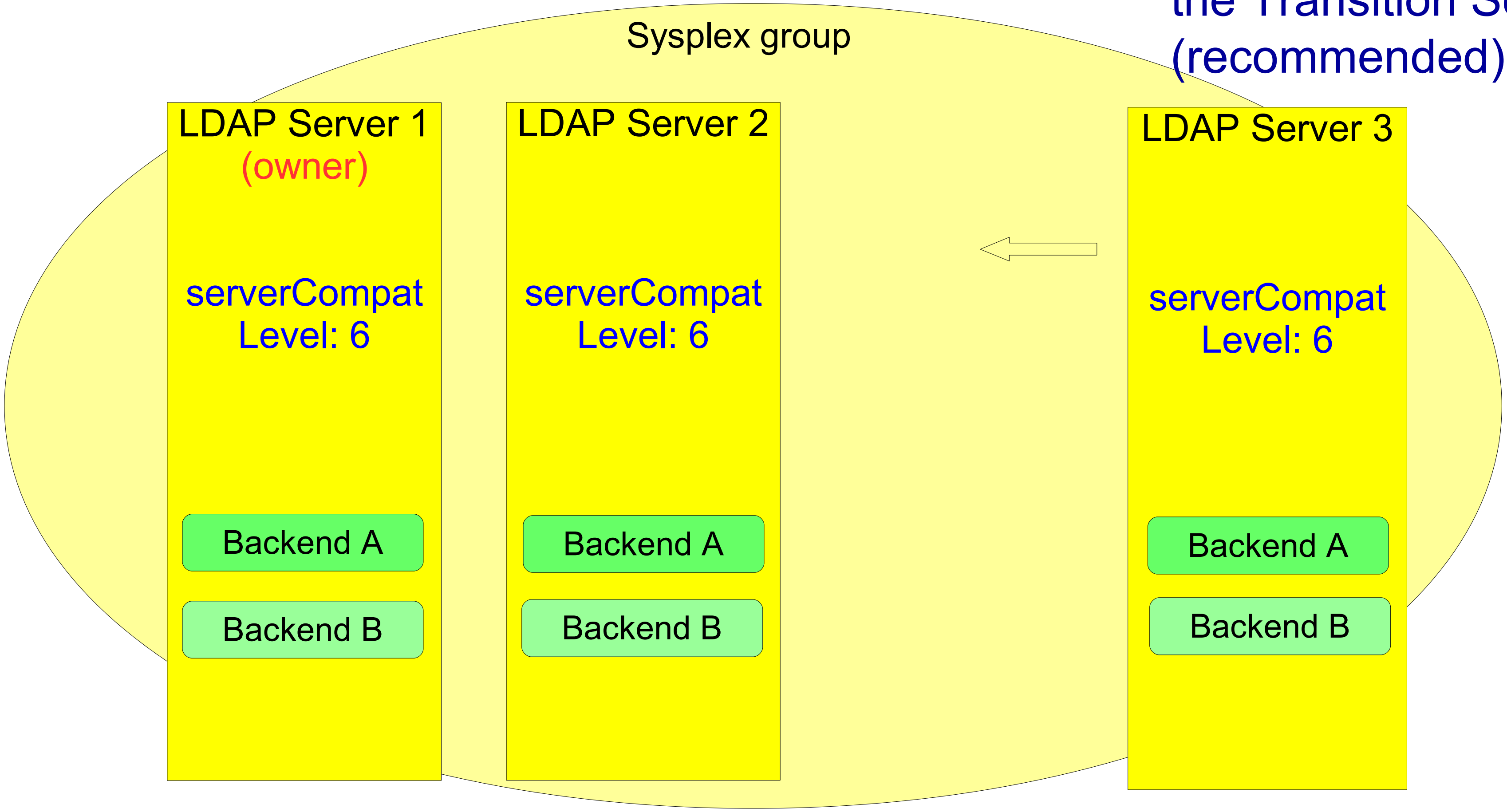
Usage & Invocation: Compatibility Level Upgrade Without LDAP Outage

New Scenario: Restart the other servers



Usage & Invocation: Compatibility Level Upgrade Without LDAP Outage

New Scenario: Restart the Transition Server (recommended)



The sysplex owner changes after the LDAP server 3 restarts.

Usage & Invocation: Compatibility Level Upgrade Without LDAP Outage

- Operator command option LEVEL displays not only server version and server level, but also compatibility level.
  - Sample command: f dssrv,display level
  - Sample output:
    - GLD1001I LDAP server version 3.22, Service level xxxxxx.
    - GLD1315I LDAP server compatibility level 7.
- Operator command option XCF displays which server is in transition mode.
  - Sample command: f dssrv, display xcf
  - Sample output:
    - GLD1135I Sysplex status
    - Server System Status
    - DCEIMGVV0066-BF31AD8E-2EFB4F4E DCEIMGVV0066 ACTIVE/REPLICA/TRANSITION
    - DCEIMGVV004E-BF31AD83-ACE95202 DCEIMGVV004E ACTIVE/OWNER
  - The TRANSITION status flag is cleared when the transition server completes transition process successfully.
  -
- New attribute in RootDSE: “ibm-slapdServerCompatibilityLevel”

Migration & Coexistence Considerations: Compatibility Level Upgrade Without LDAP Outage

- All servers in the sysplex must first be running z/OS V2R2 before this function can be utilized.
- Once V2R2 is installed everywhere, the LDAP serverCompatLevel can be updated to implement new function without shutting down all LDAP servers in the sysplex group simultaneously.
- The servers must be at serverCompatLevel 4 or higher to use this function initially.

Overview: Dynamic Group Performance Enhancements

- Problem Statement / Need Addressed
  - Customers experience high CPU in large directories with many dynamic groups defined. Group gathering via ibm-allgroups searches grows proportionally to the number of dynamic group memberURLs defined in the directory.
  - For example, defining dynamic groups based on organizations (such as department) in large corporations with thousands of departments creates heavy load. Groups are often gathered in conjunction with user logins.
- Solution
  - Improve the approach used to gather dynamic groups for a user, through better caching and indexing of dynamicURL filters.
- Benefit / Value
  - With the improved scalability, customers can continue to exploit the administrative benefits of dynamic groups even with large directories. Many thousands of dynamic groups can be defined, without growth in CPU during normal login operations.

Usage & Invocation: Dynamic Group Performance Enhancements

▪

- Dynamic group URLs are used to determine membership
  - Presented as attribute **memberURL** in dynamic group entries
  - Syntax: *ldap:///<base DN>??<scope>?<filter>*
  - Base DN and scope determine the search scope
  - Filter defines the matching rule
  - A user entry only needs to match one of the group's URLs to be in the group
- Dynamic groups have administrative benefits over static groups
  - Can be set up once based on search parameters and then be maintenance free. Static groups tend to be volatile, requiring updates as new users appear in the directory.
  - Extermely large static groups in TDBM present numerous performance issues, including partition skew in DB2, and issues trying to delete the group.

Usage & Invocation: Dynamic Group Performance Enhancements

- In the past, determining a user's groups for dynamic groups required matching the user entry against each URL in the directory. This is costly and does not scale.
- This enhancement introduces a partial index. Dynamic group memberURLs are indexed, when possible, on filter predicates:
  - equal and present filter predicates within the URL are eligible for indexing.
  - AND filters with at least one eligible predicate are also eligible for indexing.
  - OR filters are only eligible for indexing if all predicates are eligible.
  - Other filter types are not eligible: substring, greater or equal, less or equal, and NOT filters.
- High frequency filters predicates are not indexed. These provide little benefit if they drive too many URL evaluations.
- The memberURLs are separated into indexed and non-indexed. If the indexed memberURLs cannot be found by looking up the user's attribute/values in the filter index, they do not match. Large numbers of indexed memberURLs are therefore bypassed for evaluation. The remaining non-indexed URLs are evaluated in the old way (although additional optimizations have been made in this processing as well).

Usage & Invocation: Dynamic Group Performance Enhancements

- Example

Assume we have 1001 departmental dynamic groups with these URLs defined:

```
ldap:///o=ibm??sub?(&(dept=0001)(objectclass=person))
ldap:///o=ibm??sub?(&(dept=0002)(objectclass=person))
... ...
ldap:///o=ibm??sub?(&(dept=1000)(objectclass=person))
ldap:///o=ibm??sub?(objectclass=person)                ............. This is the "anyone" group
```

And the user entry is defined as:

```
cn=user050,c=us,o=ibm
objectclass=person
dept=0050
```

Each of the “dept” filter predicates occurs once. These are low frequency, and get indexed. To locate the user's groups, we lookup in the index for “dept=0050”. Only 1 of the 1000 URLs is identified as a candidate. URL matching is performed and we bypass 999 URLs.

In addition, the objectclass filter predicate occurs 1001 times, and is high frequency. However, 1000 of the URLs which use it are indexed by “dept”. Only the “anyone” group needs to go in the non-indexed URL list. Assuming users are only in one department, we typically only need to do 2 URL evaluations instead of 1001. Even if there are 10,000 or 1,000,000 groups, in this arrangement, we need only evaluate 2 dynamic URLs for any given user.

Usage & Invocation: Dynamic Group Performance Enhancements -- Dynamic group suggestions

In some cases, there may be flexibility in how dynamic groups could be defined. Some choices may be better than others.

- Suppose you have 3 departments that start with “Accounting”, such as Accounting1, Accounting2, Accounting3.
  - A URL filter like (dept=Accounting*) cannot be indexed.
  - A URL filter like (|(dept=Accounting1)(dept=Accounting2)(dept=Accounting3)) can be indexed. If there are hundreds of such accounting departments, you might still prefer the substring approach for simplicity and to avoid error.

- Consider using consistency in URLs to allow better internal consolidation

  ldap:///c=us??sub(&(objectclass=Person)(dept=accounting1))
  ldap:///c=us??sub(&(Objectclass=person)(DEPT=accounting1))
  ldap:///c=us??sub(&(dept=accounting1)(objectclass=Person))
  ... these are all equivalent, but slightly different, and will not be consolidated into a single URL

- Keep filters simple
  - (&(objectclass=person)(&(dept=accounting1))) ... internal AND (“&”) not needed
  - (&(objectclass=person)(dept=accounting1)) this is equivalent, but simpler

Usage & Invocation: Dynamic Group Performance Enhancements -- Dynamic group suggestions

If existing attributes are not well structured and dynamic group filters require too much complexity, a specific attribute can be placed in user entries to identify the group, and match an indexable URL:

- The IBM-provided schema includes an objectclass and attribute to place in the user entries, which can then be used to directly match a dynamic group:

```
User Entry
    dn:  cn=John,ou=poklab,ou=stg,c=us,o=ibm
    objectclass: person
    objectclass: ibm-dynamicMember
    ibm-group: Developers

Dynamic Group
    dn: cn=Development Group,ou=stglab,c=us,o=ibm
    objectclass: groupOfURLs
    memberURL: ldap:///c=us,o=ibm??sub?(ibm-group=Developers)
```

- In the example above, the filter is a simple equal filter, and is eligible for indexing.
- Customers can also define their own schema attributes and objectclasses, and use a similar approach.

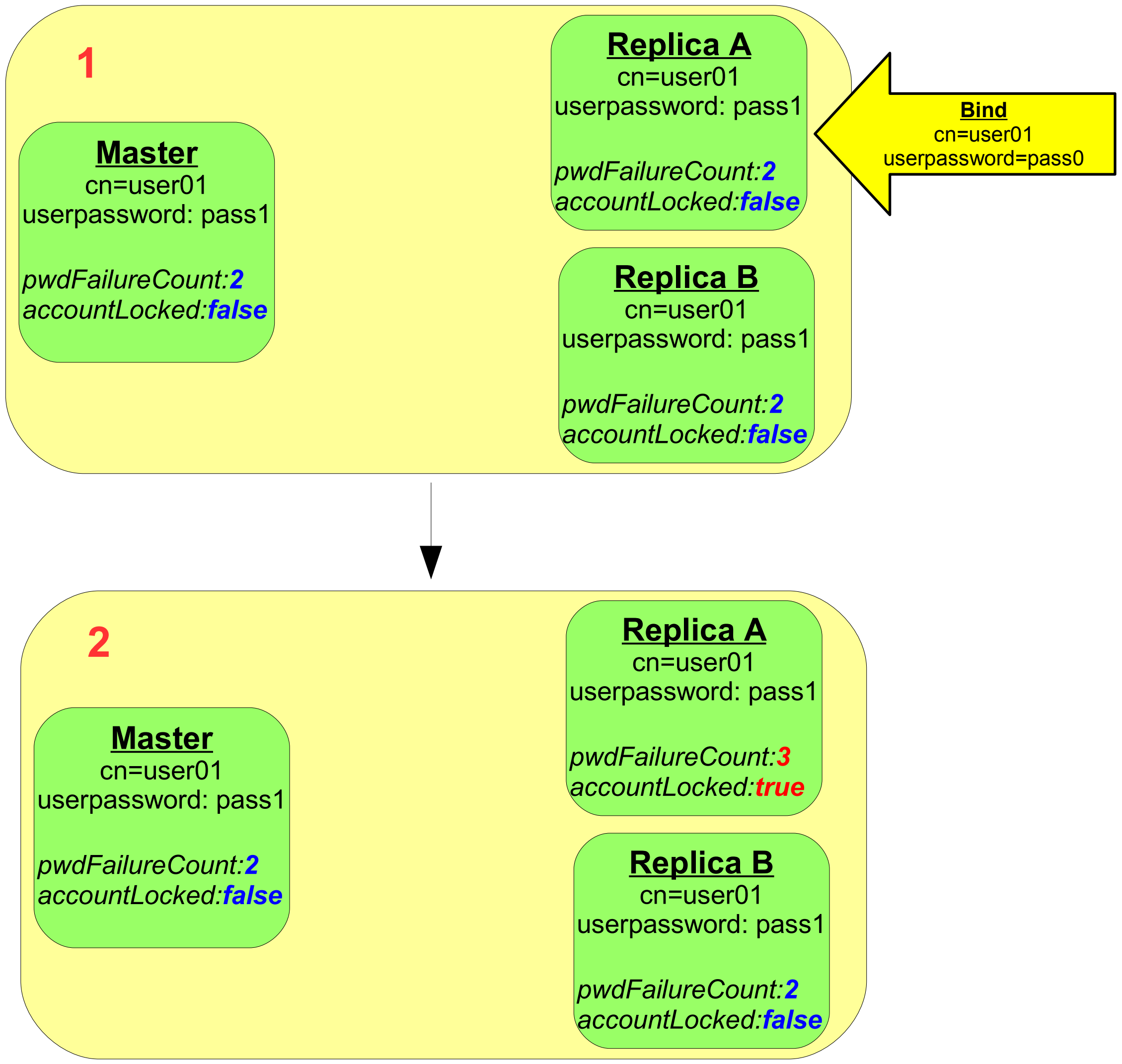
Migration & Coexistence Considerations: Dynamic Group Performance Enhancements

- If running in a mixed environment (sysplex sharing between V2R2 and V2R1), the downlevel server will continue to exhibit its past performance characteristics. Consider waiting until V2R2 is installed throughout the sysplex if you are considering deploying large numbers of dynamic groups.

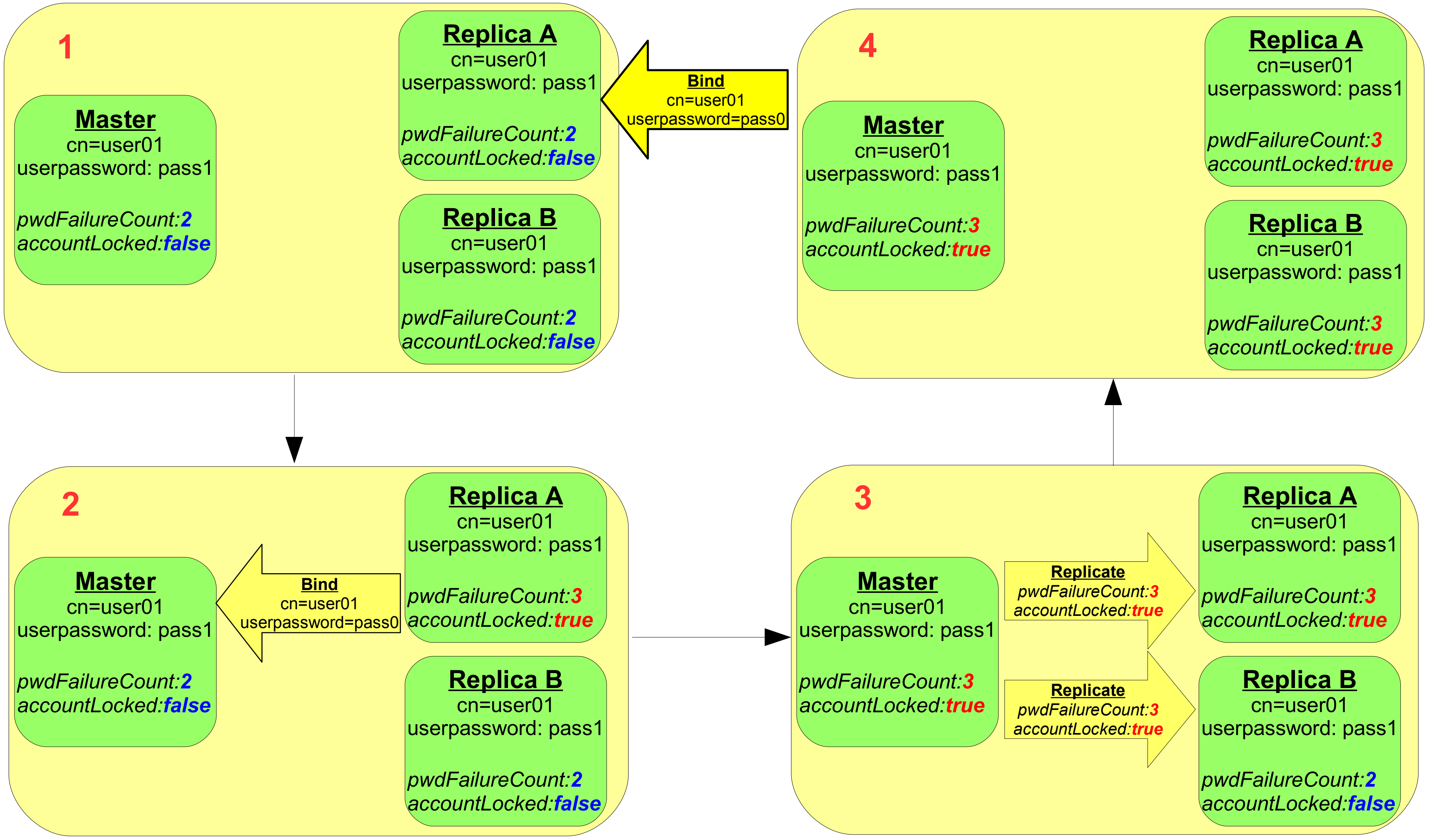
Overview: Replication of Password Policy Attributes from a Read-Only Replica

- Problem Statement / Need Addressed / User Stories
  - Read only replica server can only execute read operations (bind, search, compare).
  - If password policy is enabled, bind or compare operations may cause password policy operational attributes to be updated on the read-only replica. These updates cannot be replicated to the supplier and other servers in the same replication topology.
  - This could lead to a security issue where a user can do more login attempts than allowed by the specified password policy.
- Solution
  - Propagate password policy operational attributes from the read only replica to the supplier by sending a bind request.
- Benefit / Value
  - Provides consistent password policy operational attributes for each entry on all servers.

Overview: Bind to Read-Only Replica (without new feature)



Usage and Invocation: Bind to Replica, with New Feature Enabled



Usage & Invocation: Replication of Password Policy Attributes...

- This enhancement improves the consistency across the replication topology of password policy operational attributes updated during authentication attempts
  - pwdFailureTime
    - read-only replica propagates pwdFailureTime when the user account is not locked and authentication fails.
  - pwdExpirationWarned
    - read-only replica propagates pwdExpirationWarned when authentication succeeds and receives the password expiration warning for the first time.
  - pwdGraceUseTime
    - read-only replica propagates pwdGraceUseTime when the authentication succeeds, the password is expired, but the grace logins limit is not exceeded.
  - pwdAccountLockedTime
    - read-only replica propagates pwdAccountLockedTime when the authentication fails, and the account exceeds the number of failures allowed.

Usage & Invocation: Replication of Password Policy Attributes...

- Configure the following options on **all** servers in the replication topology:
  - Set serverCompatLevel = 8.
  - Enable and configure advanced replication.
    - ***Note:*** *This function is only supported with advanced replication, basic replication is NOT supported.*
  - Enable password policy.

    Note: *Password policies in CDBM should also be replicated, to ensure the policy rules are consistently implemented on the user entries throughout the replication topology.*
  - Specify “ibm-replicaReferralURL” in the replication context to point to the master server. The bind request is sent to the server specified in this attribute.
  - Add new attribute “ibm-slapdReplicateSecurityAttributes” to “cn=replication, cn=configuration” with a value of “TRUE”.

- If serverCompatLevel=8, ibm-slapdReplicateSecurityAttributes=TRUE and password policy is enabled, Root DSE search indicates this feature is enabled on this server by returning:

  ibm-enabledcapabilities=1.3.18.0.2.32.105

Usage & Invocation: Replication of Password Policy Attributes...

The following requests can trigger the read-only replica to propagate to the master:

- Simple bind request
  - When a simple bind request updates password policy operational attributes on a read-only replica, it triggers the replication action: Sending a bind request to the master using the same user DN and password.

    Note: *Though CRAM-MD5 and DIGEST-MD5 bind requests can update password policy operational attributes on the read-only replica, they can NOT trigger a replication action on the read-only replica because the user password is not available from the original client request.*
- Compare request involving the userPassword attribute value
  - A compare request involving the userPassword attribute value can also update the password policy operational attributes. On z/OS ITDS, this also triggers the replication action: Sending a **bind** request using the entry DN and password provided in the client request. A bind request must be sent to the master instead of a compare request because the control is only defined for bind requests.

    Note: *Distributed TDS does NOT currently support this feature when compare is used to authenticate to the read-only replica.*

Migration & Coexistence Considerations: Replication of Password Policy Attributes...

- All z/OS servers in the replication topology must be running V2R2 and must be running serverCompatLevel 8.
- If “IBM Security Directory Server” exists in the replication topology, the version must be 6.3.1 with latest fix pack or higher.

Presentation Summary

- At the end of this presentation, you should have an understanding of the following enhancements to the z/OS IBM Tivoli Directory Server in V2R2:
  - Activity Log Enhancements
  - Compatibility level upgrade without LDAP outage
  - Dynamic Group Performance Enhancements
  - Replication of password policy attributes from read-only replica

Appendix

- Publication References
  - SC23-6788 *IBM Tivoli Directory Server Administration and Use for z/OS*
  - SA23-2296 *IBM Tivoli Directory Server Messages and Codes for z/OS*