

Manila LDAP Migration

Technical Design

Table of Contents

[1 Version Control 5](#_Toc9515047)

[1.1 Revision History 5](#_Toc9515048)

[1.2 Review History 5](#_Toc9515049)

[1.3 Approval History 5](#_Toc9515050)

[1.4 References 5](#_Toc9515051)

[1.5 Terms 6](#_Toc9515052)

[2 Introduction 7](#_Toc9515053)

[2.1 Executive Summary 7](#_Toc9515054)

[2.2 Intended Audience 7](#_Toc9515055)

[2.3 Scope 7](#_Toc9515056)

[2.3.1 Out of Scope 7](#_Toc9515057)

[2.4 Assumptions 8](#_Toc9515058)

[2.5 Risk 8](#_Toc9515059)

[2.6 Design Principles and Considerations 8](#_Toc9515060)

[2.6.1 Adopt best practices 8](#_Toc9515061)

[2.6.2 Optimize processes 8](#_Toc9515062)

[2.6.3 Secure by default 8](#_Toc9515063)

[2.6.4 Provide resilience 9](#_Toc9515064)

[3 Data Architecture 10](#_Toc9515065)

[3.1 Directory Information Tree (DIT) 10](#_Toc9515066)

[3.1.1 Root Suffix 10](#_Toc9515067)

[3.1.2 User Details 10](#_Toc9515068)

[3.1.3 Service Accounts and administrators 11](#_Toc9515069)

[3.1.4 Global groups 11](#_Toc9515070)

[3.1.5 Application specific 11](#_Toc9515071)

[3.2 Access Control 11](#_Toc9515072)

[3.3 Schema 11](#_Toc9515073)

[3.3.1 Users 11](#_Toc9515074)

[3.3.2 Groups 12](#_Toc9515075)

[3.4 Index 13](#_Toc9515076)

[3.5 Access Control 13](#_Toc9515077)

[4 Physical Architecture 14](#_Toc9515078)

[4.1 Topology 14](#_Toc9515079)

[4.1.1 Replication 14](#_Toc9515080)

[4.1.2 Replication Integrity 15](#_Toc9515081)

[4.1.3 Availability 16](#_Toc9515082)

[4.1.4 Scalability 16](#_Toc9515083)

[4.2 Environment 16](#_Toc9515084)

[4.2.1 DEV Environment Architecture (Shore side) 16](#_Toc9515085)

[4.2.2 Stage Environment Physical Architecture (Shore side) 17](#_Toc9515086)

[4.2.3 Production Environment Physical Architecture (Shore side) 18](#_Toc9515087)

[5 Integration Architecture 20](#_Toc9515088)

[6 Security Architecture 21](#_Toc9515089)

[6.1 Network 21](#_Toc9515090)

[6.1.1 Transport security (SSL) 21](#_Toc9515091)

[6.2 Server and Operating System 21](#_Toc9515092)

[6.3 Installation and configuration 21](#_Toc9515093)

[6.4 Authentication 22](#_Toc9515094)

[6.5 Password Management 22](#_Toc9515095)

[6.5.1 Password Storage 22](#_Toc9515096)

[6.5.2 Password Change and Expiry 23](#_Toc9515097)

[6.5.3 Enterprise Password policy for Standard Account. 23](#_Toc9515098)

[6.5.4 Enterprise Password policy for Admin user. 23](#_Toc9515099)

[6.6 Authorization 24](#_Toc9515100)

[6.7 Data Security 24](#_Toc9515101)

[7 Operation Architecture 25](#_Toc9515102)

[7.1 Deployment 25](#_Toc9515103)

[7.1.1 Installation 25](#_Toc9515104)

[7.1.2 Data Migration 25](#_Toc9515105)

[7.1.2.1 Schema 25](#_Toc9515106)

[7.1.2.2 DIT 25](#_Toc9515107)

[7.1.2.3 Data 26](#_Toc9515108)

[7.1.2.4 Other 26](#_Toc9515109)

[7.2 Monitoring 26](#_Toc9515110)

[7.2.1 Monitoring of Directory service availability 26](#_Toc9515111)

[7.2.2 Monitoring active users 27](#_Toc9515112)

[7.2.3 Monitor replication: 27](#_Toc9515113)

[7.2.4 Monitoring Replication Delay over LDAP: 27](#_Toc9515114)

[7.2.5 Monitoring the work queue 28](#_Toc9515115)

[7.2.6 Monitoring Replication Connectivity 28](#_Toc9515116)

[7.2.7 Monitoring of Disk Space: 28](#_Toc9515117)

[7.2.8 Monitoring Certificate Expiration 28](#_Toc9515118)

[7.2.9 LDAP Search Monitoring Command: 29](#_Toc9515119)

[7.3 Audit and Logging 30](#_Toc9515120)

[7.3.1 Auditing 30](#_Toc9515121)

[7.3.2 Splunk Integration 30](#_Toc9515122)

[7.4 Backup and Restoration 30](#_Toc9515123)

[7.5 Reporting 31](#_Toc9515124)

[8 Design Decisions 32](#_Toc9515125)

[9 Open Items 33](#_Toc9515126)

# Version Control

## Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Comments |
| 1.0 | 5/9/2019 | KPMG | Draft version |
| 1.1 | 5/23/2019 | KPMG | Updated based on input from RCL |

## Review History

|  |  |  |
| --- | --- | --- |
| Version | Reviewed By | Date |
| 1.0 | Srini Nambiar | 5/20/2019 |
| 1.0 | Aman Kakkad | 5/21/2019 |
| 1.0 | Frank Ramirez Castaneda | 5/22/2019 |

## Approval History

|  |  |  |
| --- | --- | --- |
| Version | Approved By | Date |
| 1.0 | RCL | TBD |
|  |  |  |

## References

|  |  |
| --- | --- |
| Document | Author |
| Installation and Admin Guide | ForgeRock |
| LDAP migration recommendation | KPMG |
| LDAP Architecture | KPMG |

## Terms

Table 1: Terms

|  |  |  |
| --- | --- | --- |
| Abbreviation | Term | Description |
| Base DN | Base Distinguished Name | The starting point of where LDAP searches begin. |
| DS | Directory Server | This refer to new ForgeRock Directory server created for Manila LDAP |
| DIT | Directory Information Tree | Data represented in a hierarchical tree-like structure consisting of the Distinguished Names of directory service entries. |
| DN | Distinguished Name | An attribute that both uniquely identifies an entry and describes its position in the DIT. |
| LB | Load Balancer | A load balancer is a device that distributes network or application traffic across a cluster of servers |
| HTTP | Hypertext Transfer Protocol | An application protocol for distributed, collaborative, and hypermedia information systems. |
| HTTPS | Hypertext Transfer Protocol Secure | An application protocol for distributed, collaborative, and hypermedia information systems through an encrypted channel. |
| SSL | Secure Sockets Layer | A standard security technology for establishing an encrypted link between a server and a client. |
| LDAP | Lightweight Directory Access Protocol | A software protocol used to access and manage directory information. |
| LDAPS | Lightweight Directory Access Protocol Secure | A software protocol used to access and manage directory information through an encrypted channel. |

# Introduction

## Executive Summary

Royal Caribbean Cruises Ltd. (RCL) would like to migrate from RedHat 389 directory server to ForgeRock directory server to improve scalability, increase performance while modernizing directory service platform. This directory server is presently being used by MyRCL Home Portal and other applications for authentication and storing user details.

This document covers the logical, physical, integration and operational architectures as well as the design details, decisions and approaches taken to setup a new Directory service known as **forthecrews** in Culpeper Datacenter and migrate the user’s data from Manila LDAP to new Directory service. This document details the solution design across all environments: DEV, STAGE, and PROD and should be considered as the sole extensive design resource that is the foundation for all environment setup for new Directory server and to migrate the existing Manila LDAP user’s data to new Directory server, superseding all other documentation (if any).

## Intended Audience

Intended audience of this document includes Manila LDAP migration project stakeholders responsible for the solution migration, including architecture, infrastructure, network, security and implementation.

## Scope

The scope of this design document is limited to the following outcomes:

1. Setup new development, stage and production environment for Manila LDAP in highly available (HA) mode.
2. Migration of existing schema, users and their data from Manila LDAP to new instance of production LDAP.

### Out of Scope

Any outcome not identified in scope is deemed out of scope including but not limited to:

1. Changes to existing process which creates users in Manila LDAP for creating users in new LDAP instance.
2. Changes in pre-check application to authenticate against new production LDAP.
3. Migration of existing users to the development and stage environment.

## Assumptions

This section describes the assumptions that affect the new Directory Service design. Any changes to the assumptions can affect the validity of the requirements, and alter the parameters which govern this design.

1. Technical development changes in the target applications and server changes will be executed by appropriate RCL engineering teams. KPMG team will co-ordinate with RCL’s teams for these changes.
2. RCL team is responsible for resolving technical environmental issues (i.e., application servers, databases, network routing, etc.).
3. RCL team is responsible for any data cleanup activities need to be done before migrating the users in directory server.

## Risk

The following risks have been identified for the delivery of this solution

1. Infrastructure availability and build support is required from RCL infrastructure team.

## Design Principles and Considerations

The Manila LDAP Implementation team was guided by following principles for all design decisions and approaches while development of this document.

### Adopt best practices

Adopt industry best practices and standards while implementing. Security threats are ever evolving and so are security standards safeguarding against them.

### Optimize processes

Reduce manual processes as they can be error prone, introduce new risks and costly to maintain. Encourage automation wherever it is practical and cost-efficient.

### Secure by default

Features and data can only be accessed when access has been explicitly assigned and are protected from unintended or unauthorized access, change or destruction by leveraging principle of least privilege. This includes protection from unplanned events and natural disasters. The solution defaults to a secure version of transports protocols for information transport and should encrypt all information while it is at rest.

### Provide resilience

Infrastructure must be implemented such that there is no single point of failure and provides services at and exceeding the standard expected load to ensure availability of integrated applications.

# Data Architecture

This section describes the data structure of new Directory service

## Directory Information Tree (DIT)

The following DIT is recommended for the new Directory server (**forthecrews**)

ou=people

uid=<user id>

ou=groupss

cn=<group name>

ou=app\_groups

ou=<application>

cn=<group name >

dc=rccl, dc=com

dc

ou=accounts

uid=<user id>

### Root Suffix

The root suffix is configured as “**dc=rccl, dc=com**” that will be root of the entire DIT to reflect the name of company.

**Note:**

1. All the directory data under root suffix will be stored in a single backend database since there is no need to split data for replication purpose.
2. Backend database will not be encrypted to reduce impact on performance. File level access will ensure that rogue user does not have access to database.
3. Backend database file will be maintained on a separate file mount (/apps) in stage and production environment. This will ensure that there is no impact to OS operations due to any disk space constraints caused by directory server.

There is no specific requirement to automatically delete expired entries

### User Details

User details should be stored under “**ou=people, dc=rccl, dc=com**”. This branch is currently designed to store information about all the users.

**Note:**

The design above will not support data partitioning based on the user’s location.

### Service Accounts and administrators

All the service accounts and IT administrators will be stored under “ou=accounts, dc=rccl, dc=com”.

### Global groups

All the global groups would be stored under “ou=groups, dc=rccl, dc=com”.

### Application specific

Each application will have specific branch designed to store its group membership details. Application specific service account, administrators and group administrators will have access to this branch.

## Access Control

See Section 6.6 Authorization for more details about the access control model for directory information tree.

All the access to directory server would be logged and monitored as defined in section 7.3 Audit and Logging.

## Schema

Directory server will be designed to store the following types of data

1. Users – Represent the end-users and service accounts.
2. Groups – Represent collection of users.

### Users

The user will be created as “**inetOrgPerson**” with “metadata” object class attached to add RCL specific Information. The “metadata” object class will be created as “**Auxiliary**” object class to ensure that it can be added to the **inetOrgPerson** entries as needed to identify users associated with RCL.

The following attributes will be used to define users in directory

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Attribute Name | Object Class | Type | Mandatory | Matching Rules | Description |
| 1 | E1ID | metadata | Numeric String | No | N/A | User employee ID |
| 2 | Nationality | metadata | Country String | No | Equality Match:caseExactMatch | Employee nationality |
| 3 | ShipCode | metadata | String | No | Equality Match:caseExactMatch | Employee's current ship assignment |
| 4 | CompanyCode | metadata | String | No | N/A | Employee's current ship assignment (name representation) |
| 5 | CompanyName | metadata | String | No | Equality Match:caseExactMatch | Employee's current brand assignment (name representation) |
| 6 | Group | metadata | String | No | Equality Match:caseExactMatch | Employee classification (contextual) |
| 7 | Role | metadata | String | No | Equality Match:caseExactMatch | Employee classification (contextual) |
| 8 | ShipName | metadata | String | No |  | Employee's current ship assignment (name representation) |

### Groups

The groups will be represented by “**groupofuniquenames**” object class and the users will be assigned to these groups through “member” attribute.

Please note that group should be designed based on specific access criteria to ensure that limited number of users are part of the group.

**Note:**

During group definition phase, it may be appropriate to evaluate whether group should be modelled as an attribute to reduce impact on groups.

## Index

Directory server uses attribute indexes to improve the performance. These indexes are defined based on the usage pattern for a particular application. Additional indexes as needed will be created during application on-boarding phase. See Section 5 Integration Architecture for more details.

## Access Control

Access to the data stored in directory server is an important aspect of directory design to ensure data security. Please refer to Section 6.6 Authorization for more details.

# Physical Architecture

This section details the deployment topology and physical architecture of the AM Solution in Development (DEV), Stage (STAGE), and Production (PROD) environments.

## Topology

Each datacenter will consist of the following setup

1. All the servers have globally unique server-id set.
2. All the servers will have synchronized clocks to prevent issues with replications.
3. At least two servers running an instance of ForgeRock directory servers (with replication enabled) on port 1389 (LDAP) and port 1636 (LDAPS).
4. ForgeRock directory servers will be fronted by a pair of load balancer that:
   * 1. Exposes a single IP address that applications can connect to (through corresponding domain name).
     2. Load-balances incoming traffic across two or more directory servers.
     3. Monitors the health of directory servers. See section 7.2.1 Monitoring of Directory service availability for specific details about how the health of directory server can be monitored.
     4. If directory server failure is detected, stops directing traffic to the failed server.
     5. Once a failed directory server is available, adds the directory server automatically back to load balancing pool.

### Replication

Replication is the process of copying updates between DS servers such that all directory servers converge on identical copies of directory data. Replication is designed to let convergence happen over time by default. Letting convergence happen over time means that different replicas can be momentarily out of sync. It also means that if you lose an individual server or even an entire data center, your directory service can keep on running, and then get back in sync when the servers are restarted or the network is repaired. ForgeRock DS replication does not operate in active-passive mode. Instead, it supports ability to read and write to any running server. Replication replays your changes as soon as possible. After setting up replication, replicated DS directory servers publish an external change log over LDAP. This change log allows authorized client applications to read changes to directory data.

Below point will be considered while setting up the replication:

1. Replication per Suffix
2. Replication Connection Selection

The replication will be setup such that:

1. Directory servers will have built-in replication server.
2. All the servers within data-center are part of single replication group so that directory servers automatically connect to replication server within the data-center.
3. Directory servers in the data-center will be integrated with directory server in other data-centers in a mesh (i.e. each directory server will be connected to all the other directory servers in the other data-centers).

**Note:**

At this time, no sub-tree or fractional (attributes) replication is being setup.

### Replication Integrity

Replication on ForgeRock Directory Servers can generate two types of conflicts: modify and naming conflicts. Modify conflicts involve concurrent modifications to the same entry. Naming conflicts involve other operations that affect the DN of the entry.

Replication can resolve **modify conflicts** and many naming conflicts automatically by replaying the changes in the correct order. To determine the relative order in which changes occurred, replicas retain historical information for each update. The historical information is purged after configured (property replication-purge-delay with value of 3 days) duration.

The following conflicts are resolved through this process:

1. The attributes of a given entry are modified concurrently in different ways on different replicas.
2. An entry is renamed on one replica while being modified on another replica.
3. An entry is renamed on one replica while being renamed in a different way on another replica.
4. An entry is deleted on one replica while being modified on another replica.
5. An entry is deleted and another entry with the same DN added on one replica while the same entry is being modified on another replica.

Replication cannot resolve the following **naming conflicts**, so they must be resolved manually:

1. Different entries that share the same DN are added concurrently on multiple replicas.
2. An entry on one replica is moved (renamed) to use the same DN as a new entry concurrently added on another replica.
3. A parent entry is deleted on one replica while a child entry is added or renamed concurrently on another replica.

Please refer to Section [7.2.2](#_Replication) Replication for additional details for conflict resolution.

### Availability

The topology has been designed with two directory servers with a local load-balancer pair in front. The setup will ensure that

1. Application connects to single IP address front-ended by Load balancer pair. There is no change needed in application in case of any topology changes behind load-balancer
2. Load-balancer will continue to monitor availability of both the servers and in case of failure of one node automatically redirect the future traffic to another server till the failed node is back up.
3. In case of the failure of one of the nodes, the monitoring system will try to automatically restart the service and then server. If the system is not available, it will notify the Operator for further action.

### Scalability

The initial architecture is scoped for 1.3M users. The architecture provides for future scalability to accommodate expected growth as follows:

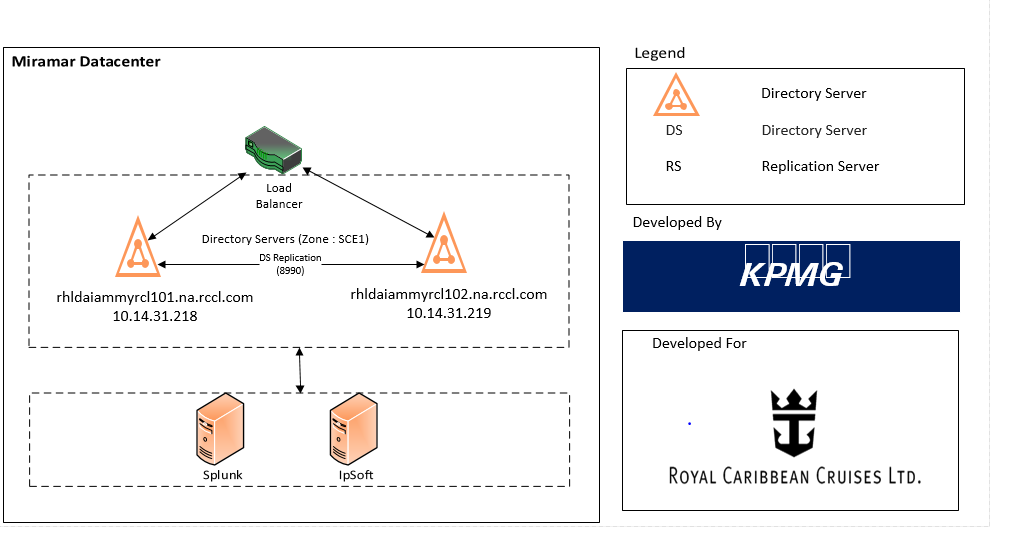
1. The infrastructure is designed to scale horizontally by adding new directory server and configuring load-balancer to distribute traffic to the new server.
2. Infrastructure is designed to scale by deploying it across multiple data-centers with the ability to reach eventual data consistency through replication process.

## Environment

This section describes the high-level architecture diagram for dev, stage and production environment.

### DEV Environment Architecture (Shore side)

This section explains the physical architecture of ForgeRock DS (**forthecrews**) in the DEV environment (Miramar, Florida, US) on shoreside.

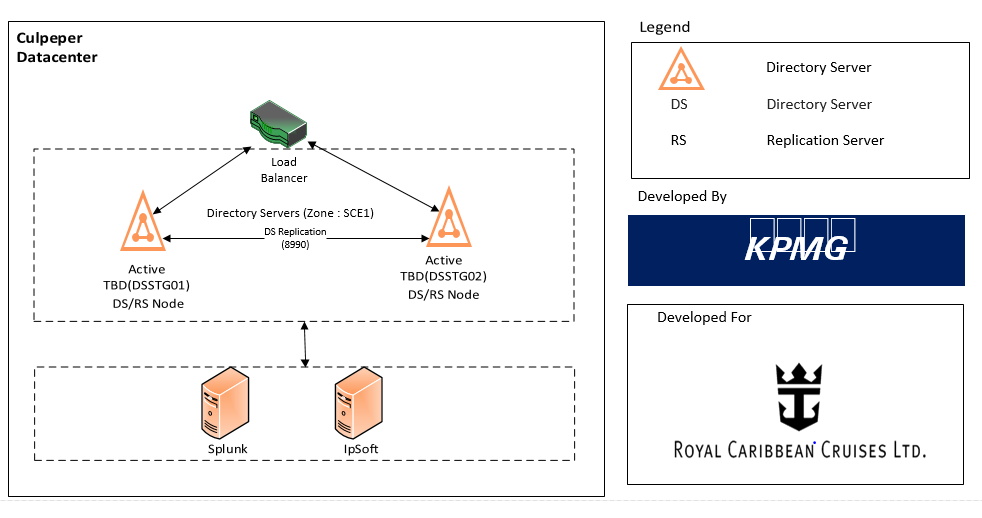


Directory Server DEV Physical Architecture (Shoreside)

ForgeRock directory (**forthecrews**) server deployment on shoreside DEV consists of two ForgeRock Directory Server nodes (with replication capability enabled) deployed in HA mode. Each DS node replicates user data to and receives updates from other directory server. Directory server nodes connect to each other to communicate replication data (changelog data). Replication happens on a secure channel. Directory server nodes purge delta after a pre-defined interval of 3 days.

### Stage Environment Physical Architecture (Shore side)

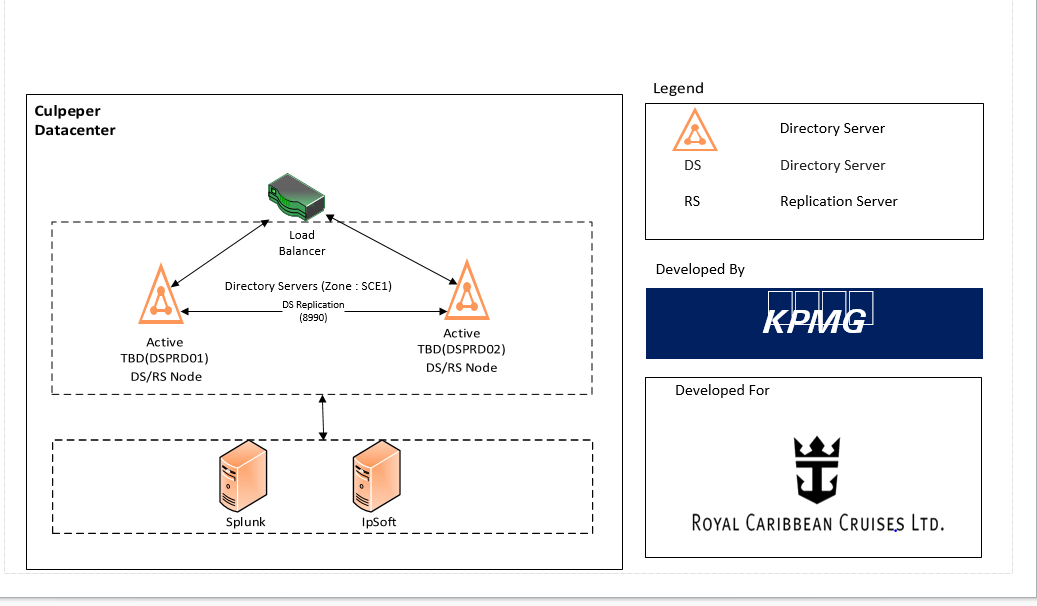
This section explains the physical architecture of ForgeRock DS (**forthecrews**) in the STAGE environment (Culpeper, Florida, US) on shoreside.



ForgeRock Directory server deployment on shoreside STAGE consists of two ForgeRock Directory Server (DS) (with replication capability enabled). The directory servers are fronted by a load-balancer. Directory Server nodes replicates changes and receive updates from other directory servers. Replication happens on a secure channel. Directory server nodes purge delta after a pre-defined interval. Constant connectivity between directory nodes is very critical for consistency.

### Production Environment Physical Architecture (Shore side)

This section explains the physical architecture of ForgeRock DS (**forthecrews**) in the Production environment (Culpeper, Florida, US) on shoreside.



ForgeRock Directory server deployment on shoreside PROD consists of two ForgeRock Directory Server (DS) nodes (with replication enabled). The directory servers are fronted by a load-balancer. Directory Server nodes replicates changes to and receive updates from other directory servers. Replication happens on a secure channel. Replication nodes purge delta after a pre-defined interval. Constant connectivity between replication nodes is very critical.

# Integration Architecture

This section details the approach to integrate various applications and other services with the directory service.

The primary approach for integrating with directory service will be through LDAPS protocol. This approach will involve the following steps:

1. Establishment of an application specific service account with appropriate access (See Section 6.5 Authorization for more details about access model).
2. Understanding of the operation that application needs to perform.
3. Based on the search criteria, identify the attributes that should be indexed and setting up the index.
4. Configure application to connect to directory server over LDAPS protocol using service account and password and perform relevant operations.
5. Deploy the application.

Please note that due to the nature of directory service, application should consider the following:

1. Directory service is based on concept of eventual consistency which implies that application should assume/workaround the concept that updates may not be available in other regions/data-centers or other LDAP Server immediately after updating the record on one server.
2. Directory service is designed for high read and low write. Application should not store data associated with use-cases that do not fall into the sweet spot of the service.

# Security Architecture

Due to sensitive nature of the data stored, security architecture of directory service is paramount.

## Network

The following controls will be setup to ensure network security of the deployment

1. All the LDAP server components would be located in secure computing environment SCE 1.
2. Access to SSH port will be limited to the jump server (or as defined by enterprise security policy).
3. Access to the LDAPS port (1636) will be limited to specific applications that need to connect to these server (as defined by enterprise security). Access to the LDAP port (1389) will be limited from the machine and specific IP addresses for debugging purpose.
4. Administrators can access LDAP (1389) & LDAPS (1636) port using LDAP tools directly. This access will be replaced by access through Privileged Access Management solution once it is available.
5. Security scan will be performed to ensure that there are no known vulnerabilities.

### Transport security (SSL)

Applications will connect to directory server over LDAPS (port 1636) protocol. The server authentication will be performed using public-CA signed certificate (DEV, STAGE and PROD) as agreed by RCL team in line with enterprise certificate policy.

**Note:**

1. There is no requirement to support mutually authenticated SSL.
2. There is no specific requirement to restrict the protocols & cipher suites used to connect to server over SSL.

Internal communication between directory server components within and across data-center will be performed over secured channel.

## Server and Operating System

All the appropriate security controls as defined by enterprise security policy must be implemented. Latest version of supported operating system will be used to build the directory servers.

## Installation and configuration

Access to installed files and directories will be limited to service account. The process will be running as service account without any additional privileges (e.g. root access to listen on port < 1024).

## Authentication

Access to the server over SSH will be limited to approved users in line with enterprise security Policy. Access to server over LDAP will be limited to all the users with **inetOrgPerson** entry in directory server including:

1. End-users on-boarded to directory service as entry.
2. Administrators who need to manage access.
3. Service accounts of application that need to connect to LDAP for performing various operations.

In addition to that internal accounts like “**cn=Directory Manager**” used to setup replication will have full control.

**Note:**

1. Server will reject any simple bind request with DN and no password.
2. There is no limit to number of concurrent clients connection to this server
3. Server will not reject any request received from a client that has not yet been authenticated, whose last authentication attempt was unsuccessful, or whose last authentication attempt used anonymous authentication.
4. Server will return extra information about why bind failed.
5. SASL authentication will be supported and preferred over simple text.

## Password Management

DS password policies govern not only passwords, but also account lockout, and how DS servers provide notification about account status.

DS servers support password policies as part of the server configuration, and subentry password policies as part of the (replicated) user data.

By default, DS directory servers include two password policy configurations, one default for all users, and another for the default directory superuser, **cn=Directory Manager**.

**Note:**

Password policy configuration updates must be applied to each replica in the deployment.

### Password Storage

Password will be stored using salted SHA512 hashing algorithm. Since most of the password will be migrated from existing application, the password will initially be stored as MD5 (current setup for Manila LDAP) and then migrated to Salted SHA512 whenever the user authenticates for the first time. This will be performed by setting “deprecated-password-storage-scheme” on the password policy to “MD5”.

### Password Change and Expiry

The system will enable the following capabilities (in line with enterprise password policy)

1. All the authentication and password change operations will be performed over secured channel (LDAPS)
2. Allow users to change their own password (important for application that authenticate and change password)
3. Password change by user will require current password.
4. Password should expire after 90 days
5. Allow users to change expired password
6. Expire password without any warning
7. Don’t force password change on adding a new user or password reset operation
8. Don’t allow grace login after password expiry.
9. Don’t maintain history of previous password

### Enterprise Password policy for Standard Account.

The system will enforce the following capabilities (in line with enterprise password policy)

1. Account will be locked out if user provides invalid password 5 times.
2. Account will be automatically unlocked after 30 mints.
3. Minimum password Length-9 characters
4. Password should be mixture upper, lower alpha numeric and special characters.
5. Password should not contain ship or brand name.

### Enterprise Password policy for Admin user.

The system will enforce the following capabilities (in line with enterprise password policy)

1. Account will be locked out if user provides invalid password 5 times.
2. Account will be automatically unlocked after 30 mints.
3. Minimum password Length-15 characters
4. Password should be mixture upper, lower alpha numeric and special characters.
5. Password should not contain ship or brand name

## Authorization

An ACI target statement specifies the entry, attributes, or set of entries and attributes for which you want to control access. The ACI for directory server will be setup such that:

1. User are able to bind using their credentials and read all attributes (typically used by RCL home Portal application for authentication)
2. Directory Server administrators have full access to the root suffix and its sub-tree.   
     
   **Note:**

Access by the directory server administrators will be restricted from IP address of Privileged Access Management server after the PAM on-boarding is complete. This would ensure that administrators can perform administrative operations only through PAM solution.

1. Applications will use corresponding service account to connect to directory server. These service accounts will have read and write access to **ou=people** and application specific directory tree.

## Data Security

The data storage on disk will not be encrypted due to performance impact. Access to the runtime data will be restricted to limit access to the backend files. In addition to that backup and archival process will be expected to encrypt the files prior to storage.

# Operation Architecture

This section explains the operational architecture of the new directory setup and the migration of existing Manila LDAP user’s data.

## Deployment

### Installation

As part of migration process a custom script will be developed which will do the silent installation and configuration of the directory server stack. Please refer to Section 4.2 Environment to understand all the components of Directory Structure in DEV and STAGE and PROD environments.

This utility will install following components based on supplied input parameters:

1. ForgeRock directory server (with replication capability enabled)
2. ForgeRock directory nodes configured to synchronize through replication.

The utility runs as a user (service account) to perform the installation operations.

### Data Migration

This project is focused on migrating the existing Manilla directory infrastructure to new ForgeRock Directory Server infrastructure.

The following process will be used to migrate data from existing Manila LDAP to FR Directory server.

#### Schema

The current 389 AD directory server uses the following types of object classes

1. Standard LDAP object classes like organization Unit (OU), inetOrgPerson.
2. Custom LDAP object class metadata.

The custom object class “metadata” is currently defined of type “Structural”. This will be changed to “Auxiliary” class to ensure that it is compliant with standard LDAP guidelines. All the attributes defined and used are in line with standard LDAP schema.

#### DIT

A new DIT, as defined in Section 3.1 Directory Information Tree, will be created to separate the different types of users, accounts, groups and application specific entries.

Existing application will have to be updated to accommodate the new directory information tree. In addition to that application will be expected to use application specific service account to bind to directory server over LDAPS protocol.

#### Data

In addition to schema and DIT, user data will be migrated to the new platform.

The following approach will be used for migration

1. User data will be exported from 389 AD server. This data will be transferred over secure channel from 389 AD server to new directory server.
2. Script will be run to create an import LDIF file such that
   * 1. All records identified for removal is removed.
     2. All non-operational attributes (except userPassword) are migrated as it is.
     3. Attribute “userPassword” will be base64 decoded to convert it to format supported for import to directory server.
     4. Base DN will be updated to reflect new DIT.
3. Converted LDIF file will be imported.
4. Server password policy attribute “deprecated-password-storage-scheme” will be set to “MD5” and “default-password-storage-scheme” will be set to “Salted SHA-512” to ensure that current MD5 password would be migrated to SSHA-512 password hashing mechanism when the user authenticates next time
5. If user’s decrypted passwords are available, a script will be run to authenticate user against new directory server to migrate password to new hashing algorithm where possible.

#### Other

No other details will be migrated from 389 AD server.

## Monitoring

This section explains the liveness monitoring of directory server service, replication and infrastructure. SNMP based monitoring will be setup to monitor out of box configuration.

### Monitoring of Directory service availability

ForgeRock DS provides a mechanism for validating DS health over HTTP and LDAP. A proper heartbeat check is progress through Connect 🡪 Bind 🡪 Unbind 🡪 Disconnect to fully verify the status of the DS server. Following command is executed which connects, binds, issues a base (scope) level search on the baseDN of "", unbinds and finally disconnects.

./ldapsearch --port <DS\_ADMIN\_PORT> --bindDN <BIND\_DN> --useSsl --trustAll --baseDN "" --searchScope base "(objectClass=\*)" 1.1

Above LDAP search operation returns the result as "dn:" (only). The LDAP request attribute "[1.1.](https://tools.ietf.org/html/rfc4511#section-4.5.1.8)" returns the distinguished name only. ForgeRock Directory server also provides following HTTP endpoints for verifying server health. Both these endpoints will also be monitored.

1. **/alive**: Provides an endpoint to check whether the server is currently alive, meaning that its internal checks have not found any errors that would require administrative action.
2. **/healthy**: Provides an endpoint to check whether the server is currently healthy, meaning that it is alive and any replication delays are below a configurable threshold.

### Monitoring active users

Monitoring users who are currently connected to the DS server using the following command:

$. /ldapsearch --port <DS\_ADMIN\_PORT> --bindDN <BIND\_DN> --bindPassword <password> --baseDN "cn=monitor" --searchScope sub "(objectClass=ds-monitor-connection\*)" \\*

### Monitor replication:

Below ‘dsreplication’ status command to give an overall view of the replication topology, including whether the servers are synchronized:

./dsreplication status --adminUID admin --adminPassword password --hostname <ds1.example.com> --port 4444 –trustAll

### Monitoring Replication Delay over LDAP:

Below is the default monitor user account to check the delay in replication

$ ldapsearch --port 1389 --bindDN uid=monitor --bindPassword password \

--baseDN cn=monitor "(ds-mon-current-delay=\*)" ds-mon-current-delay

The delay reflects the time between the latest update that the replica has received and the latest update that the replica has replayed. This metric is accurate only when the replica receives updates quickly. In the event of a network partition, the delay cannot accurately reflect updates happening on the other side of the partition.

### Monitoring the work queue

Below search command will be provide the information about the work queue to monitor

./ldapsearch --port <bindport> --bindDN "<BIND\_DN> " --bindPassword password --baseDN "cn=Work Queue,cn=monitor" --searchScope sub "(objectClass=\*)" \\*

### Monitoring Replication Connectivity

In order to ensure that replication across data-center is working, the replication work queue is monitored. In addition to that, connectivity between the replication server and replication server in other data-centers should be monitored.

### Monitoring of Disk Space:

The command below will be use to check whether the server is running out of disk space

curl --user <monitor>:<password> http://<HostName>:8080/metrics/prometheus 2>/dev/null | grep disk

### Monitoring Certificate Expiration

The command below will be used to check whether the server certificate is due to expire soon:

curl --user <monitor>:<password> http://opendj.example.com:8080/metrics/prometheus 2>/dev/null | grep cert

### LDAP Search Monitoring Command:

Below are some basic LDAP search commands against the cn=monitor entry and sub-entries to provide a variety of statistics that are useful for monitoring ForegRock Directory service:

|  |  |
| --- | --- |
| **BaseDN** | **Details** |
| cn=monitor | Provides general server information (an example of the type of information returned is shown below). |
| cn=Disk Space Monitor,cn=monitor | Provides information about disks, including disk location and space available. |
| cn=Work Queue,cn=monitor | Provides information about the work queue, including its backlog, average and max backlog. |
| cn=jvm,cn=monitor | Provides information about the system and the JVM, including memory usage. |
| cn=LDAP,cn=connection handlers,cn=monitor | Provides information about all open client connections.  Replaces cn=Client Connections,cn=monitor |
| ds-cfg-backend-id=userRoot,cn=Backends,cn=monitor | Provides monitoring information about the Berkeley DB Java Edition backend.  Replaces cn=userRoot Database Environment,cn=monitor |
| cn=System Information,cn=monitor | Provides information about the system and the JVM. |
| cn=JVM Memory Usage,cn=monitor | Provides information about memory usage in the JVM. |
| cn=Client Connections,cn=monitor | Provides information about all open client connections. |
| cn=userRoot Database Environment,cn=monitor | Provides monitoring information about the Berkeley DB Java Edition backend. |

## Audit and Logging

### Auditing

This section explains the audit features designed for DS. ForgeRock DS stack supports a comprehensive audit logging framework that captures key auditing events, critical for system security, troubleshooting, and regulatory compliance. By default, the server stores the following files under the <install directory>/**logs** directory.

1. Access logs for messages about clients accessing the server. One form of access log is a directory audit log. An audit log records changes to directory data in LDIF.
2. Debug logs for messages tracing internal server events.
3. Error logs for messages tracing server events.
4. Replication logs for messages used to help repair problems in data replication.
5. A server.out log for messages about server events since startup. Messages in this file have the same format as error log messages.
6. A server.pid process ID file when the server is running.

Logs are handled by log publishers. Log publishers determine which messages to publish, where to publish them, and what output format to use. The server's logging system supports extensibility through the ForgeRock Common Audit event framework. The log message formats are compatible for all products using the framework. The framework uses transaction IDs to correlate requests as they traverse the platform. This makes it easier to monitor activity and to enrich reports. Although the ForgeRock Common Audit event framework supports multiple topics, DS software currently supports handling only access events. DS software divides access events into **ldap-access** events and **http-access events**.

### Splunk Integration

This section explains the integration of DS with Splunk. DS will send audit logging data to a Splunk HTTP event collector REST endpoint. **DS configuration** document will contain the details on Splunk configuration.

## Backup and Restoration

This section explains the backup and restoration design for ForgeRock Directory Server.

ForgeRock directory servers (DS) back up data either in compressed, binary format, or in LDIF file(s). A DS backup process backs up the backend files, schema files and task LDIF files. In addition to these config.ldif, which contains main configuration for the DS instance, including replication data, and admin-backend.ldif, which contains the certificate for each replicated instance and the admin entry used for replication, are also backed up. Keystores / truststores, which contains client (truststore and keystore), replication (ads-truststore) and administration (admin-truststore and admin-keystore), key pairs are also backed up.

The backup will be performed using script/commands in line with enterprise backup policy.

## Reporting

We don’t have any non-functional requirement for reporting matrix.

# Design Decisions

The following decision matrix summarizes the key decisions factors in choosing and designing the transition strategy:

1. Coexistence of both Directory Server at the same time.
2. Data Consistency Level should be high.
3. Performance of new Directory Server should be high.
4. High Availability.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Issue | Approach | Decision | Justification |
| 1 | External Changelog | Approach 1: Enable external changelog that would allow external application to monitor changes.  Approach 2: Disable external changelog to reduce data storage requirements. | Approach 2 | External changelog capability will not be enabled at this time since there is no significant expectation of recovery from user error or need to track changes by connecting applications. |

# Open Items

This section lists down any items that are open concerning the IDM solution design.

Table <#>: IDM Open Items

|  |  |  |  |
| --- | --- | --- | --- |
| # | Item Description | Owner | Status |
| 1 | Password Policy |  |  |