

# CUDB Data Distribution

---

## FACILITY DESCRIPTION

**Copyright**

© Ericsson AB 2016, 2017. All rights reserved. No part of this document may be reproduced in any form without the written permission of the copyright owner.

**Disclaimer**

The contents of this document are subject to revision without notice due to continued progress in methodology, design and manufacturing. Ericsson shall have no liability for any error or damage of any kind resulting from the use of this document.

**Trademark List**

All trademarks mentioned herein are the property of their respective owners. These are shown in the document Trademark Information.



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Scope	1
1.2	Revision Information	1
1.3	Target Groups	1
1.4	Prerequisites	1
1.5	Typographic Conventions	1
<b>2</b>	<b>Overview</b>	<b>3</b>
2.1	Prerequisites	3
2.2	Architecture	3
2.3	Description	7
2.4	Dependencies and Interactions	11
<b>3</b>	<b>Operation and Maintenance</b>	<b>13</b>
3.1	Configuration	13
3.2	Fault Management	14
3.3	Performance Management	14
3.4	Security	14
3.5	Logging	14
	<b>Glossary</b>	<b>15</b>
	<b>Reference List</b>	<b>17</b>





# 1 Introduction

This document provides a description for the different data distribution features in the Ericsson Centralized User Database (CUDB).

## 1.1 Scope

The purpose of this document is to describe the data distribution features supported by CUDB.

## 1.2 Revision Information

**Rev. A** This document is based on 6/155 34-HDA 104 03/9 with the following changes:

- Section 2.2.1 on page 3: Updated LDAP Data Views information.

**Rev. B** Editorial changes only.

**Rev. C** Editorial changes only.

## 1.3 Target Groups

This document is intended for system administrators and users working with the data distribution.

## 1.4 Prerequisites

This section is not applicable to this feature.

## 1.5 Typographic Conventions

Typographic conventions can be found in the following document:

- *Typographic Conventions*





## 2 Overview

From the perspective of processing and data structure, CUDB is a distributed database. Data is partitioned and distributed across different memory clusters called Data Store Units. These DS units are distributed over a set of CUDB Nodes that comprise the CUDB system and contain a portion of the full database.

Seamless access to the full data is provided by any CUDB node, no matter which CUDB node actually hosts the DS unit containing the requested data.

### 2.1 Prerequisites

This section is not applicable to this feature.

### 2.2 Architecture

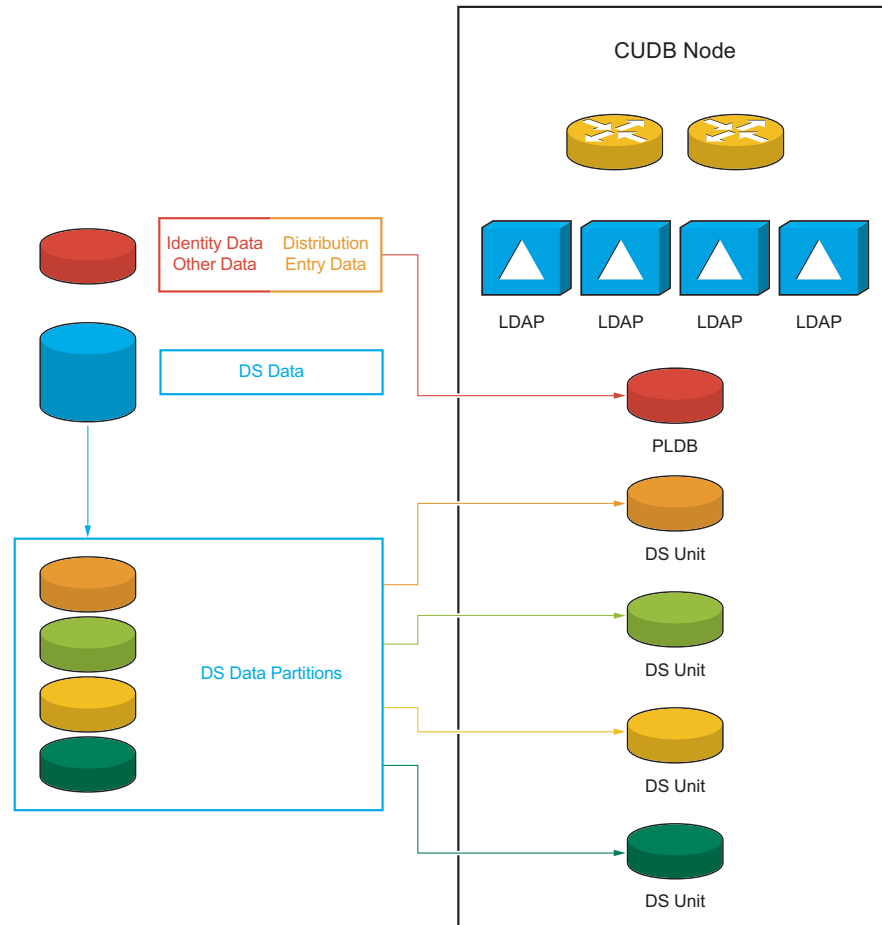
This section describes the architecture of the data distribution feature.

#### 2.2.1 CUDB Data Types and Data Architecture

Following are different types of CUDB data, depending on the nature of data, and distribution property:

- Data Store (DS) data:  
  
Subscription data for the different application Front Ends (FEs) accessing the CUDB. The subscriber data is the minimum set of data that cannot be partitioned and which is accessed by one or more unique identifiers.
- Processing Layer (PL) data:
  - Identity data: set of identifiers to access the subscriber data. This can be any kind of identifier such as IMSI, IP address, URI, and so on.
  - Distribution Entry (DE) data: contains internal distribution data used by CUDB to locate DS data within the distributed system.
  - Other PL data.

Figure 1 shows the data types and the CUDB node data architecture.



*Figure 1 CUDB Data Types and CUDB Node Data Architecture*

The CUDB node data architecture is based on a set of “in memory databases” storing all these types of data in the following ways:

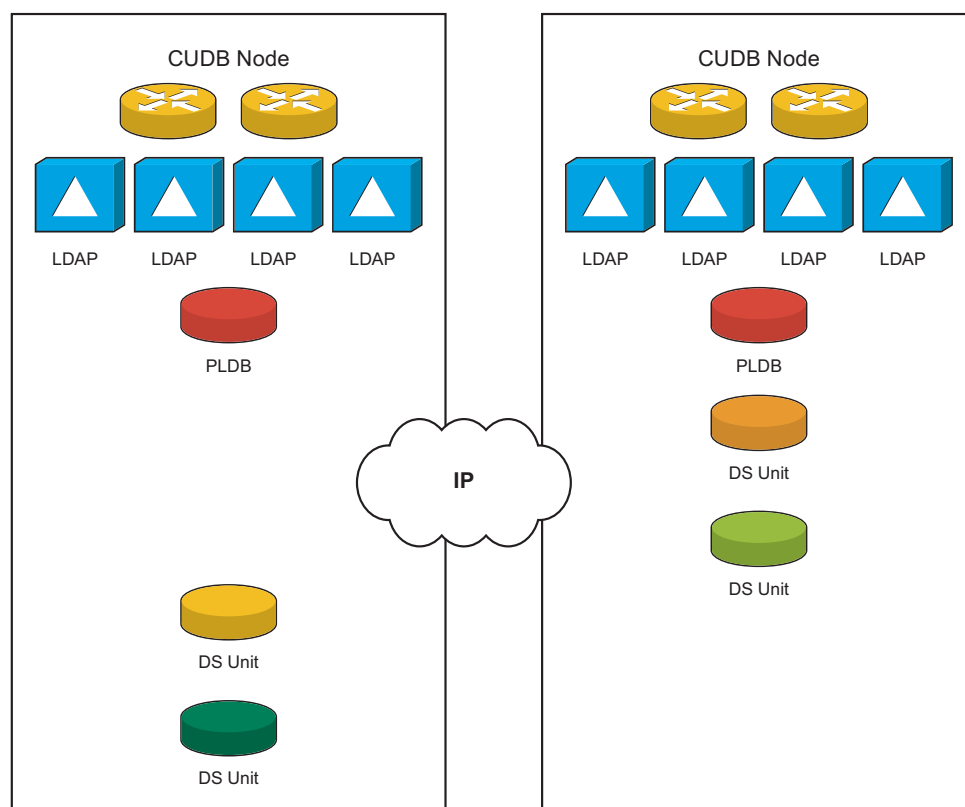
- DS data is partitioned and distributed across the different databases hosted in the same or in different CUDB Nodes. These databases are called DS units. In this way, different CUDB nodes store different partitions of the subscriber data and there can be CUDB nodes containing not all of the partitions but only a set of the whole distributed database.
- PL Data is stored in a unique database called the Processing Layer Database (PLDB).

For more information on capacity configuration and scalability, refer to *CUDB Data Storage Handling*, Reference [2].



DS data, once stored, can be redistributed between the different DS units in the system. For more information, refer to *CUDB Subscription Reallocation*, Reference [3].

The CUDB system is a set of CUDB Nodes interconnected by the IP Backbone as shown in Figure 2.



*Figure 2 CUDB Data Distribution*

The CUDB system provides seamless access to the whole distributed database through any CUDB node which has connected an application FE (established Lightweight Directory Access Protocol (LDAP) session).

If CUDB nodes are located in different geographical sites, then distributed data is also geographically distributed.

CUDB supports geographical redundancy so that geographically distributed data can be replicated in one or two other DS units hosted in different CUDB sites within the CUDB system.

The set of DS units storing same replicating data is a DS Unit Group (DSG).



For more information on geographical redundancy, refer to *CUDB High Availability*, Reference [4].

The LDAP Data Views function supports accessing stored data through customizable views.

Virtual entries are composed of real attributes mapped into virtual attributes, which are grouped into virtual object classes, and those into virtual entries. LDAP users that have a view assigned to them cannot see the core Directory Information Tree (DIT), only the virtual tree. Access to CUDB data through a view supports the same LDAP requests as through the core DIT, but restrictions apply on write operations to ensure data and schema rules integrity. For more details, refer to *CUDB LDAP Data Views*, Reference [5].

**Note:** The LDAP Data Views function can only be used if the Application Facilitator Value Package is available.

## 2.2.2

### CUDB LDAP DIT DEs

CUDB provides an LDAP interface. The data stored in CUDB is managed as an LDAP DIT. Different type of data is stored in different branches in the DIT.

DIT entries whose descendants store distributed data are called DEs.

CUDB identifies DEs in the DIT based on the branch where the entry instance is stored. The branches can be configured differently.

Figure 3 shows an example of the DIT.

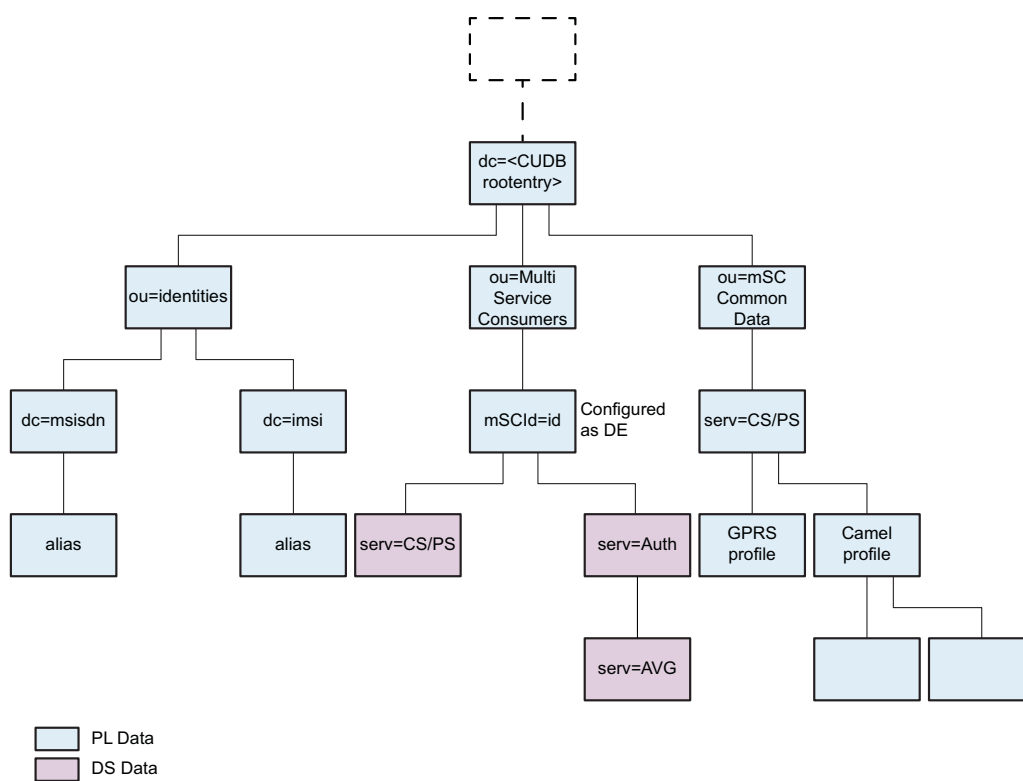


Figure 3 Data Information Tree and Data Distribution

For more information on CUDB DIT and DE configuration, refer to *CUDB LDAP Interwork Description*, Reference [6].

## 2.3 Description

Every time an entry is to be created in the DIT, CUDB analyzes whether this entry is a DE or not. If yes, the entry itself is stored in the PLDB and a valid DSG is selected from the ones defined in the system to store its distributed data. This selection is determined by the Data Distribution mechanisms provided by CUDB.

Distributed data, which are children of different DEs, can be stored in the same or a different DSG.

The distribution mechanism is applied only once when a specific DE entry is first created. Once the entry is identified as a DE, the mechanism is applied and a target DSG is obtained as result. The DE itself is stored in the PLDB containing the data of the target DSG where distributed data of the DE is stored when created.

The supported data distribution mechanisms are described in the following sections.



The behavior of the system does not allow to store distributed data into the DSG0 (that is, PLDB). All DEs are stored in the DSG0 and all their distributed data is stored in a DSG different from 0.

### 2.3.1 Default Data Distribution

The default distribution mechanism selects a DSG using a probability function where every DSG is assigned a weight based on its level of occupancy. The occupancy level of a DSG corresponds to the memory level of the most occupied replica.

This distribution mechanism allows the specification, at provisioning time, of a geographical zone (if GeoZones is enabled).

This is specified by the `ZoneId` attribute received in the LDAP request, as follows:

- If the `ZoneId` attribute is received and the specified zone is valid, then only the DSGs belonging to that zone are evaluated.
- If the `ZoneId` attribute is not received and a default zone is configured in the system, only the DSGs belonging to the default zone are evaluated.
- If the `ZoneId` attribute is not received and a default zone is not configured in the system, all DSGs in the system are evaluated.

For more information on CUDB multiple geographical zones, refer to *CUDB Multiple Geographical Areas*, Reference [7].

For each available DSG, the following value is calculated:

$$(\text{random-number} \bmod (\text{max\_mem\_usage} - \text{DSGx occupancy})) + 1$$

See the following definitions of the terms above:

**DSGx occupancy**

The percentage of memory used in the DSG.

**max\_mem\_usage**

Fixed value, set to 95%.

The system selects the DSG with the highest value obtained.

### 2.3.2 Custom Data Distribution

The custom data distribution mechanism allows the specification of a customized dynamic library that is loaded at system start-up and whose custom function results a target DSG where that specific DE instance must be assigned.



The custom function receives the following set of information from the CUDB:

- Values of the attributes received in the distributed data to be stored. This changes depending on the entry.
- Dynamic information:
  - Memory use of each DSG.
  - List of DSGs that are disabled for provisioning.
- Configuration data:
  - Zone definitions
  - Default zone
  - Maximum occupancy level for a DSG.

This function can be extended to use any information from any external source.

For more details on the use of dynamic library for distribution, refer to *CUDB Dynamic Library for Distribution*, Reference [8].

### 2.3.3 Data Distribution Policy Overwrite

Data distribution policy overwrite mechanism allows the specification, at provisioning time, of destination DSG where descendants of that specific DE entry is to be allocated.

This is specified by `DSUnitGroup` attribute received in the LDAP request. For more information on CUDB DIT and DE configuration, refer to *CUDB LDAP Interwork Description*, Reference [6].

The reception of this attribute overrides any other mechanism.

### 2.3.4 Data Distribution Mechanism Interactions

The evaluation of the distribution is a combination of the different mechanisms detailed in the following steps:

1. If `DSUnitGroup` is received, then no other distribution mechanism is applied. No further actions are taken but a general check including the validity and availability of the received DSG id.
2. If no DSG is received and dynamic library is configured, then system applies the custom distribution function.

There are three possible actions, depending on the result of the function:

- A specific DSG is obtained. If the DSG does not exist, then the default distribution algorithm is applied.



- No target DSG is obtained, but the default distribution algorithm is applied.
  - No target DSG is obtained, and the LDAP operation is rejected with error code 53 `LDAP unwilling to perform`.
3. The default mechanism if the dynamic library is not configured is as follows:
- If the `ZoneId` is received, then the system checks that the geographical zones are configured, and that the specified zone exists. Depending on the result of the check, the following can happen in this case:
    - If no geographical zones are configured, then the LDAP operation is rejected with error code 19 `LDAP constrain violation`.
    - If geographical zones are configured, but the specified zone does not exist, then the LDAP operation is rejected with error code 19 `LDAP constrain violation`.
    - If the specified zone is valid, it is assigned as a target zone.
  - If the `ZoneId` is not received, but geographical zones are configured, then the target zone is the default zone.

Once a target zone has obtained valid `ZoneId`, or default zone or none (whole system), default selection mechanism is applied between all the configured DSGs within the target zone.

In general, once the target DSG is obtained by the former mechanisms the following can happen:

- If a DSG is not found, then the LDAP operation is rejected with error code 53 `LDAP unwilling to perform`.
- If a DSG is found but its occupancy level is over the maximum, then LDAP operation is rejected with error code 53 `LDAP unwilling to perform`.
- Else if resulting DSG is valid as target DSG, then DSG is assigned to the DE and the process continues.

### 2.3.5 CUDB Subscription Reallocation

CUDB subscription reallocation applies once data is distributed and stored between the different DSGs in the system.

CUDB supports administrative redistribution between DSGs in the system for the distributed data stored in the CUDB system. That is, CUDB administratively moves stored data from one DSG to other DSGs. It is also possible to use it for higher level movements than DSG (node, site, or zone) by a proper handling of origin and target DSGs.



For more details on this feature, refer to *CUDB Subscription Reallocation*, Reference [3].

### 2.3.6 CUDB Data Store Group Provisioning

CUDB supports the disabling and enabling of a specific DSG for provisioning.

A provisioning-disabled DSG is not involved in any of the mechanisms for data distribution.

Since the DSG0 is the PL, it must be avoided as a possible candidate for storing distributed data. For this reason, the parameter must not accept DSG0 as a possible value.

Specifically in overwrite mechanism, if the DSG id received in the LDAP operation identifies a DSG which is provisioning-disabled the operation is rejected with an error code 53 LDAP unwilling to perform.

## 2.4 Dependencies and Interactions

This section is not applicable to this feature.





## 3 Operation and Maintenance

For information on subscription reallocation and multiple geographical zones operation and maintenance, refer to *CUDB Subscription Reallocation*, Reference [3] and *CUDB Multiple Geographical Areas*, Reference [7].

### 3.1 Configuration

This section describes the configuration of the data distribution feature.

#### 3.1.1 DE Configuration

The `distributedEntryPattern` attribute, class `CudbLdapAccess`, configures the DEs. It specifies which entries are considered DEs in the CUDB.

When a DE is provisioned, both of the optional attributes `ZoneId` and `DSUnitGroup` must be defined in any of the `objectClasses` belonging to that entry.

#### 3.1.2 Custom Distribution

The use of custom distribution is enabled or disabled.

The `customDistributionPolicyEnabled` attribute, class `CudbLdapAccess`, indicates that a custom distribution policy library is loaded. When this attribute is changed to `true`, the library is loaded and when changed to `false`, the library is unloaded.

For more information on attributes, refer to *CUDB Node Configuration Data Model Description*, Reference [9] and for further instructions, refer to *CUDB System Administrator Guide*, Reference [1].

#### 3.1.3 DSG Provisioning Management

CUDB supports the disabling and enabling of a specific DSG for provisioning by `cudbDsgProvisioningManage` command. For more information on this command, refer to *CUDB Node Commands and Parameters*, Reference [10].

#### 3.1.4 Configured DSG Checking

For further instructions on how to check the DSGs configured in the system, refer to *CUDB System Administrator Guide*, Reference [1].



## 3.2 Fault Management

This section is not applicable to this feature.

## 3.3 Performance Management

This section is not applicable to this feature.

## 3.4 Security

This section is not applicable to this feature.

## 3.5 Logging

Table 1 lists the logged messages related to the data distribution (only error and warning logs are considered).

*Table 1 Data Distribution Logs*

Severity	Message Information	Trigger Event
WARNING	No space left in DSG <dsg_id>	Impossible to add an entry because the DS cluster is full.
WARNING	DSG <dsg_id> returned by distribution external library does not exist. Default distribution algorithm will be applied	The DSG returned by distribution external library does not exist.
WARNING	Zone <zone_id> does not exist	The zone does not exist.
WARNING	No space left in CUDB System	The CUDB system is full.
WARNING	No space left in zone <zone_id>	The zone is full.

For more information, refer to *CUDB Node Logging Events*, Reference [11].



## Glossary

For the terms, definitions, acronyms, and abbreviations used in this document, refer to *CUDB Glossary of Terms and Acronyms*, Reference [12].





## Reference List

### **CUDB Documents**

- [1] *CUDB System Administrator Guide*
- [2] *CUDB Data Storage Handling*
- [3] *CUDB Subscription Reallocation*
- [4] *CUDB High Availability*
- [5] *CUDB LDAP Data Views*
- [6] *CUDB LDAP Interwork Description*
- [7] *CUDB Multiple Geographical Areas*
- [8] *CUDB Dynamic Library for Distribution*
- [9] *CUDB Node Configuration Data Model Description*
- [10] *CUDB Node Commands and Parameters*
- [11] *CUDB Node Logging Events*
- [12] *CUDB Glossary of Terms and Acronyms*