Red Hat Directory Server 12

规划和设计目录服务器

用于规划有效目录服务的概念和配置选项

Last Updated: 2024-11-23

Red Hat Directory Server 12 规划和设计目录服务器

用于规划有效目录服务的概念和配置选项

法律通告

摘要

了解目录设计的各个方面，包括目录树、架构、拓扑、复制和安全性的设计。查找有关目录服务的好处和选项的更多信息、目录服务器实施的策略和高级配置示例。

目录

提供有关红帽目录服务器的反馈

我们感谢您对我们文档和产品的输入信息。请让我们了解如何改进文档。要做到这一点：

- 要通过 JIRA 提交有关红帽目录服务器文档的反馈（需要帐户）：
  1. 转至 红帽问题跟踪程序。
  2. 在 **Summary** 字段中输入描述性标题。
  3. 在 **Description** 字段中输入您对改进的建议。包括文档相关部分的链接。
  4. 点对话框底部的 **Create**。
- 通过 JIRA 提交有关红帽目录服务器产品的反馈（需要帐户）：
  1. 转至 红帽问题跟踪程序。
  2. 在 **Create Issue** 页面上，单击 **Next**。
  3. 填写 **Summary** 字段。
  4. 在 **Component** 字段中选择组件。
  5. 填写 **Description** 字段，包括：
     a. 所选组件的版本号。
     b. 重现问题的步骤或您的建议以改进。
  6. 点 **Create**。

第 1 章 目录服务简介

红帽目录服务器提供集中目录服务。目录服务器与现有系统集成并充当整合员工、客户、供应商和合作伙伴信息的集中存储库。使用目录服务器，您可以管理用户配置文件和身份验证。

在以下章节中了解在设计目录之前需要了解的内容。

1.1. 关于目录服务

目录服务 是存储企业信息的软件、硬件和进程的集合，提供对此信息的访问。目录服务由至少一个目录服务器实例和一个目录客户端应用程序组成。客户端应用程序可以访问存储在 目录中的名称、电话号码、地址和其他数据。

目录服务的一个示例是域名系统(DNS)服务器。DNS 将主机名映射到 IP 地址。DNS 客户端向 DNS 服务器发送请求，并且服务器回复 server.example.com 具有哪个 IP 地址。因此，所有主机都成为 DNS 服务器的客户端。此外，用户可以通过记住主机名而不是 IP 地址在网络中轻松定位计算机。DNS 服务器的一个限制是，它只存储两种类型的信息：主机名和 IP 地址。true 目录服务存储出几乎无限的信息。

在 Red Hat Directory Server 中，您可以将以下数据存储在一个可从网络访问的存储库中：

- 物理设备信息，如组织中打印机的数据：位置、制造商、购买日期和序列号。
- 公共员工信息：名称、电子邮件地址和部门。
- 私人员工信息：工资、政府身份证明号、主页地址、电话号码和付款等级。
- 合同或帐户信息：客户端的名称、最终交付日期、投标信息、合同号和项目日期。

目录服务器提供标准协议和应用程序编程接口(API)，以访问其包含和满足许多应用需求的信息。

1.1.1. 关于全局目录服务

红帽目录服务器通过向各种应用程序提供信息来提供全局目录服务。直到最近，许多应用程序都与自己的专有用户数据库捆绑在一起，包含特定于该应用的用户的信息。虽然专有数据库如果您只使用一个应用程序，但如果数据库管理相同的信息，则多个数据库会变得管理负担。

例如，公司运行三个不同的专有电子邮件系统，每个电子邮件系统都有自己的专有目录服务。如果用户更改某个目录中的密码，则更改不会自动复制到其他目录中。在不同地方管理相同的信息会增加硬件和人员成本。增加的维护开销被称为 *n+1 目录问题*。

全局目录服务通过提供可由任何应用程序访问的集中存储库来解决 n+1 目录问题。但是，为各种应用程序提供对目录服务的访问需要基于网络的方法在应用程序和目录服务之间进行通信。

红帽目录服务器使用 LDAP 的应用程序访问其全局目录服务。

1.1.2. 关于 LDAP

LDAP 提供了客户端应用程序和服务器用于相互通信的通用语言。LDAP 是 ISO X.500 标准描述的目录访问协议(DAP)的一个"轻量级"版本。

DAP 通过可扩展且强大的信息框架访问目录，但以较高的管理成本为目录提供。DAP 使用了一个不是互联网标准协议的通信层，并具有复杂的目录命名惯例。

LDAP 在降低管理成本的同时保留 DAP 的最佳功能。LDAP 使用通过 TCP/IP 和简化的编码方法运行的开放目录访问协议。它保留数据模型，可在中型投资的硬件和网络基础架构中支持数百万条目。

1.2. 目录服务器简介

红帽目录服务器具有多个组件。目录核心是实现 LDAP 协议的服务器。您可以在 Red Hat Directory Server 中使用 LDAP SDK 编写的不同 LDAP 客户端、第三方和自定义应用程序。

Red Hat Directory Server 安装包括以下元素：

- 核心目录服务器 LDAP 服务器、LDAP v3 兼容网络守护进程(**ns-slapd**)以及所有相关的插件、用于管理服务器及其数据库的命令行工具，以及其配置和模式文件。在 配置目录数据库、配置 和架构参考中 了解更多信息。
- Web 控制台，一种图形管理控制台，可简化目录服务设置和维护。使用 Web 控制台 登录目录服务器， 了解更多信息。
- SNMP 代理使用简单网络管理协议(SNMP)来监控目录服务器。使用 SNMP 在监控目录服务器中 了解更多信息。

目录服务器在没有其他 LDAP 客户端应用程序的情况下为 I intranet 或 extranet 提供了一个基础。兼容服务器应用程序使用目录作为共享服务器信息的中央存储库，如员工、客户、供应商和合作伙伴数据。目录服务器管理用户身份验证、访问控制、用户首选项。在托管环境中，合作伙伴、客户和供应商可以管理其目录部分，从而降低管理成本。

目录服务器依赖于 *插件来添加功能*，如数据库层、复制和链数据库。您可以禁用与核心目录服务操作无关的插件。

1.2.1. 目录服务器前端概述

目录服务器是一个多线程应用程序。这意味着多个客户端可以同时绑定到同一网络的服务器。当目录服务扩展为包含大量条目或地理分布式客户端时，服务还包括放置于网络的战略性位置的多个目录服务器。

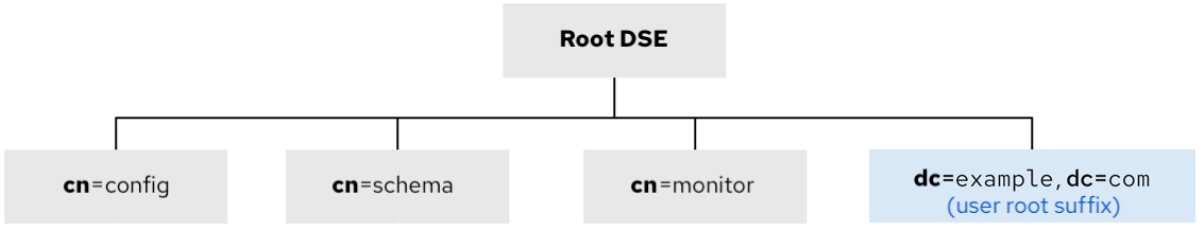
目录服务器的服务器前端使用 LDAP 通过 TCP/IP 和 LDAP 通过 Unix 套接字(LDAPI)管理与目录客户端应用程序的通信。

目录服务器可以与传输层安全(TLS)建立安全（加密）连接，具体取决于客户端是否协商以使用 TLS 连接。如果客户端是签发的证书，目录服务器可以使用 TLS 来确认客户端有权访问服务器。此外，TLS 用于执行其他安全活动，如消息完整性检查、数字签名和服务器之间的相互身份验证。

1.2.2. 基本目录服务器树概述

目录树也称为目录信息树(DIT)，镜像大多数文件系统使用的树模型。在安装过程中，Directory 服务器会创建一个默认的目录树。

默认目录树



安装后，该目录包含以下根后缀和子树：

- **Root DSE** (Root DSA 特定条目)是 LDAP 服务器的一个特殊条目。**Root DSE** 可分辨名称(DN)是零长度字符串。
- **cn=config** 包含有关服务器内部配置的信息。
- **cn=monitor** 包含服务器和数据库监控统计信息。
- **cn=schema** 包含目前在服务器中载入的 schema 元素。
- **用户根** 后缀，目录服务器在设置过程中创建的默认用户数据库的后缀。您可以在创建 Directory Server 实例时定义用户根后缀名称。用户 root 后缀通常具有 **dc** 命名惯例，如 **dc=example,dc=com**，或者为组织使用 **o** 属性，如 **o=example.com**。有关命名用户后缀的详情，请参考 Choosing the suffix。root 用户后缀与 **userRoot** 数据库关联。您稍后通过导入 LDIF 文件或创建条目来填充数据库。

您可以通过添加与目录安装相关的任何数据来扩展默认的目录树。有关目录树的更多信息，请参阅 指定目录树。

1.3. 目录服务器数据存储

数据库是存储、性能、复制和索引的基本单元。您可以在数据库上进行导入、导出、备份、恢复、索引等操作。目录服务器将数据存储在 **LDAP 数据库管理器** (LDBM)数据库中。LDBM 数据库作为插件实现，该插件会自动随目录一起安装，并默认启用。

默认情况下，Directory 服务器为根后缀使用一个后端数据库实例，一个数据库足以包含目录树。此数据库可以管理数百万条目。此数据库支持高级方法来备份和恢复数据，以最大程度降低数据丢失。

但是，您可以使用多个数据库来支持整个目录服务器部署来管理比存储在单个数据库中更多的数据。

1.3.1. 关于目录条目

LDAP 数据交换格式(LDIF) 是用于描述目录条目的标准基于文本的格式。条目由 LDIF 文件中的多个行组成，并包含有关对象的信息，如机构中的人员或网络上的打印机。

有关条目的信息以一组属性及其值表示。每个条目都有一个对象类属性，用于指定条目描述的对象类型，并定义其包含的额外属性集合。每个属性都描述了条目的特定特征。

例如，一个条目可以具有 **organizationalPerson** 对象类，表明该条目代表一个机构中个人。这个对象类支持 **givenName** 和 **telephoneNumber** 属性。分配给这些属性的值定义条目所代表的人的名称和电话号码。

目录服务器也使用服务器 *计算的只读操作属性*。管理员可以为访问控制和其他服务器功能手动设置这些操作属性。

执行目录条目的搜索

目录树以分级结构存储条目。LDAP 支持为条目查询数据库并请求目录树中分支下的所有条目的工具。分支子树的根称为基础区分名称或 *基本 DN*。例如，如果执行指定 base DN 为 **ou=people, dc=example,dc=com** 的 LDAP 搜索请求，则搜索操作仅检查 dc=example,dc=com 目录树中的 **ou=people** 子树。

默认情况下，LDAP 搜索不会返回所有条目，并排除具有 **ldapsubentry** 对象类的管理条目。管理条目可用于定义角色或服务类。要在搜索响应中包含这些条目，客户端应用程序还需要额外搜索带有 **ldapsubentry** 对象类的条目。

其他资源

- 关于目录服务器中的角色
- 使用命令行查找条目(ldapsearch)

1.3.2. 分布目录数据

如果您将目录树的部分存储在单独的数据库中，目录可以并行处理客户端请求来提高性能。您甚至可以将数据库存储在不同的机器上以进一步提高性能。要连接目录的一部分，目录服务器使用数据库链接和链。有关数据库链接和链的更多信息，请参阅在 目录服务器中使用引用。

其他资源

- 分发目录数据

1.4. 设计流程概述

1. 规划目录数据
   目录包含数据，如用户名、电话号码和组详情。本计划章节可帮助您分析机构中各种数据源，并了解其关系。了解目录可以存储和要执行的数据类型，以设计目录服务器的内容。

2. 设计目录模式
   目录旨在支持一个或多个启用了目录的应用程序。这些应用程序对目录存储的数据（如文件格式）的要求。目录架构决定了此数据的特征。了解目录服务器附带的标准模式、如何自定义架构的描述以及维护一致性模式的提示。

3. 设计目录树
   决定在读取数据层次结构设计和示例概述后如何组织和引用存储数据。

4. 设计目录拓扑
   如果您计划将目录划分为多个物理目录服务器以及这些服务器如何相互通信，了解拓扑设计。

5. 设计复制过程
   了解复制概念、copyble 数据类型、各种复制场景和高可用性目录服务提示。

6. 设计安全目录
   了解您可以如何保护您的目录。了解安全威胁、安全方法概述、分析安全性中的步骤以及设计访问控制来保护目录数据完整性的提示。

7. 设计同步
   在混合平台基础架构中，请考虑与 Microsoft Active Directory 数据库中存储的信息同步。

1.5. 部署目录

首先，安装测试服务器实例，以确保服务可以处理用户负载。如果服务没有最佳的初始配置，请调整设计并再次进行测试。调整设计，直到它满足企业需求。

创建并调优成功测试目录服务器实例后，按照以下事项开发将目录服务移至生产环境的计划：

- 所需资源的估算
- 计划需要完成的内容，以及时间
- 一组用于衡量部署是否成功的条件

1.6. 其他资源

目录、LDAP 和 LDIF 的关键资源：

- RFC 2849： LDAP 数据交换格式(LDIF)技术规格
- RFC 2251： 轻量级目录访问协议(v3)

第 2 章 规划目录数据

目录数据可以包含用户名、电子邮件地址、电话号码、用户组和其他信息。您希望在目录中存储的数据类型决定了目录结构、授予数据的访问以及请求和授予此访问权限的方式。

2.1. 目录数据简介

目录的适当数据具有以下特征：

- 数据被读取频率超过写入的频率。
- 数据可以使用 attribute-value 格式表示，如 **surname=jensen**。
- 数据不仅适用于一个个人或组。例如，多个人和应用程序可以使用员工名称或打印机位置。
- 从多个物理位置访问数据。

例如，软件应用的员工首选项设置对该目录并不适合，因为只有单个应用程序实例需要访问这些信息。但是，如果应用程序可以读取该目录的首选项设置，并且用户希望根据不同站点的偏好使用应用程序，那么在目录中包括此类设置非常有用。

2.1.1. 包括在目录中的信息

您可以在条目中添加有关个人或资产作为属性的有用信息。例如：

- 联系信息，如电话号码、物理地址和电子邮件地址。
- 描述性信息，如员工号码、工作标题、经理或管理员识别以及与作业相关的兴趣。
- 组织联系信息，如电话号码、物理地址、管理员标识和业务说明。
- 设备信息，如打印机物理位置、打印机类型以及打印机每分钟可以生成的页面数。
- 有关公司交易合作伙伴、客户以及客户的联系和账单信息。
- 合同信息，如客户名称、到期日期、工作说明和定价信息。
- 个人的软件首选项或软件配置信息。
- 资源站点，如指向特定文件或应用程序的 Web 服务器的指针或文件系统。

将目录服务器用于服务器管理之外，需要规划要在目录中存储哪些其他类型的信息。例如，您可以包括以下信息类型：

- 合同或客户端帐户详情
- 工资数据
- 物理设备信息
- 主页联系信息
- 办公室联系企业中不同站点的信息

2.1.2. 要从目录中排除的信息

Red Hat Directory Server 管理客户端应用程序读取和偶尔更新的大型数据卷，但目录服务器不是处理大型、非结构化对象（如镜像或其他介质）而设计的。您应该在文件系统中维护这些对象。但是，目录可以使用 FTP、HTTP 和其他 URL 类型将指针存储到这些类型的应用程序。

2.2. 定义目录需求

在设计目录数据时，您只能考虑当前所需的数据，以及目录（和组织）可能会随时间变化。在设计过程中考虑目录的将来需求会影响目录中数据的结构化和分发方式。

考虑以下几点：

- 您现在想要在目录中拥有什么操作？
- 通过部署目录，您希望解决哪些立即问题？
- 您使用的支持目录的应用程序的即时需求是什么？
- 您希望在不久的将来添加到目录中什么？例如，企业使用目前不支持 LDAP 的会计软件包，但此核算软件包将在几个月内启用了 LDAP。识别 LDAP 兼容应用程序使用的数据，并在可行时计划将数据迁移到目录中。
- 您要在以后存储在目录中哪些信息？例如，托管公司可以拥有与当前客户不同的数据要求，如存储镜像或媒体文件等。通过这种方式规划，您可以识别您尚未考虑的数据源。

2.3. 执行站点调查

站点调查 是发现和特征目录内容的正式方法。计划更多执行调查的时间，因为准备对目录架构至关重要。站点调查由以下任务组成：

- 识别使用目录的应用程序。
  确定您在企业中部署的启用了目录的应用程序及其数据需求。
- 识别数据源。
  调查企业并识别数据源，包括 Active Directory、其他 LDAP 服务器、PBX 系统、人工资源数据库和电子邮件系统。
- 特征目录需要包含的数据。
  确定目录中应位于哪些对象（例如，人员或组）以及这些对象在目录中要维护哪些属性（如用户名和密码）。
- 确定要提供的服务级别。
  决定客户端应用程序的目录数据可用性，并相应地设计架构。目录可用性会影响您如何配置数据复制和串联策略，以连接存储在远程服务器上的数据。
- 识别数据供应商。
  数据供应商包含目录数据的主要来源。您可以将此数据镜像到其他服务器，以实现负载平衡和恢复。确定每个数据的数据供应商。
- 确定数据所有权。
  对于每个数据，确定负责数据更新的人员。
- 确定数据访问。
  从其他来源导入数据时，为批量导入和增量更新开发策略。作为此策略的一部分，尝试在单个位置管理数据，并限制可以更改数据的应用程序数量。另外，限制写入任何给定数据的人员数量。较小的组可确保数据完整性，同时减少管理开销。

- 记录站点调查。

如果目录对多个机构产生影响，请考虑创建一个目录部署团队，其中包含来自每个受影响机构的代表进行站点调查。

公司通常都会拥有人工资源部门、会计部门或客户收帐部门、制造企业、销售组织和开发组织。包括来自每个机构的代表可帮助执行调查过程，以及从本地数据存储迁移到集中式目录。

2.3.1. 识别使用目录的应用程序

访问目录的应用程序以及这些应用程序的数据需求指导目录内容的规划。使用该目录的各种常见应用程序包括：

- *目录浏览器应用程序，如在线电话书*。决定用户需要哪些信息，并将其包含在目录中。
- *电子邮件应用程序，特别是电子邮件服务器*.所有电子邮件服务器都需要在 目录中提供一些路由信息。但是，有些信息可能需要更高级的信息，如存储用户邮箱的磁盘的位置、假通知详情和协议信息，例如 IMAP 和 POP。
- *启用目录的人工资源应用程序*.这需要其他个人信息，如政府身份识别号、家地址、家电话号码、出生日期、工资和职务。
- *Microsoft Active Directory*.通过 Windows User Sync，可以集成 Windows 目录服务，以便与 Directory Server 一起工作。这两个目录都可以存储用户信息和组信息。在现有 Windows 服务器部署后配置目录服务器部署，以便用户、组和其他目录数据可以同步。

在评估将使用目录的应用程序时，请考虑每个应用程序使用的信息类型。下表提供了应用程序的示例以及应用程序使用的信息：

表 2.1.应用程序数据需要示例

Application	数据类别	data
电话	人员	名称、电子邮件地址、电话号码、用户 ID、密码、部门号码、经理、邮件停止
Web 服务器	人员、组	用户 ID、密码、组名称、组成员、组所有者
日历服务器	人员、会议室	Name, user ID, cube number, room room name

当您识别每个应用程序使用的应用程序和信息时，您将了解多个应用程序会使用哪些类型的数据。规划中的这一步可以防止目录中的数据冗余，并明确显示与数据目录相关的应用程序需要什么。

以下因素会影响目录中维护的数据类型以及将信息迁移到目录时的最终决定：

- 各种传统应用程序和用户所需的数据
- 传统应用程序与 LDAP 目录通信的能力

2.3.2. 识别数据源

要确定要包含在目录中的所有数据，请执行对现有数据存储的调查。该调查应包括以下内容：

- 识别提供信息的组织。
  找到管理关键信息的所有组织，如信息服务、人力资源、工资和会计部门。
- 识别信息源的工具和流程。
  信息的常见来源包括网络操作系统（如 Windows、Novell Netware、UNIX NIS）、电子邮件系统、安全系统、PBX （手机切换）系统和人力资源应用程序。
- 确定对数据进行中央化如何影响数据管理。
  集中式数据管理可能需要新的工具和新的流程。在某些情况下，中央化可能需要机构中的员工和不满意。

在调查过程中，开发一个列表来标识企业中的所有信息源，如下表中所示：

表 2.2.信息源示例

数据源	数据类别	data
人力资源数据库	人员	名称、地址、电话号码、部门号码、经理
电子邮件系统	人员、组	名称、电子邮件地址、用户 ID、密码、电子邮件首选项
设施系统	设施	构建名称、指纹、现有数字、访问代码

2.3.3. 特征目录数据

使用以下方法将您要包含在目录中的数据特征：

- 格式
- 大小
- 各种应用程序中出现的次数
- 数据所有者
- 与其他目录数据的关系

在目录中查找您要包含的数据中的常见特征。这有助于在架构设计阶段节省时间，如 指定目录模式。

请考虑以下表，其特征是目录数据：

表 2.3。目录数据特征

data	格式	大小	所有者	相关
员工名称	文本字符串	128 个字符	人员资源	用户条目

data	格式	大小	所有者	相关
传真号	电话号码	14 个数字	设施	用户条目
电子邮件地址	文本	多个字符	IS 部门	用户条目

2.3.4. 确定服务级别

您提供的服务级别取决于依赖支持目录的应用程序的人员的预期。要确定每个应用程序所需的服务级别，请确定如何使用应用程序。

随着目录的演进，目录可能需要支持从生产到关键任务级别的不同服务级别。在目录部署后提高服务级别比较困难，因此请确保初始设计满足未来需求。

例如，要消除总故障的风险，请使用多层次配置，其中有多个供应商处理同一数据。

2.3.5. 考虑数据供应商

数据 供应商是提供数据的服务器。在多个位置存储相同的信息会降低数据完整性。数据供应商确保存储在多个位置的所有信息都是一致且准确的。以下场景需要数据供应商：

- 在目录服务器间复制
- Directory 服务器和 Active Directory 之间的同步
- 访问目录服务器数据的独立客户端应用程序

使用多层次复制时，目录服务器可以包含在多个服务器上的信息的主副本。多个供应商仍保持更改日志，并安全地解决冲突。您可以配置有限的供应商服务器，可以接受更改并将数据复制到副本或消费者服务器[1].如果服务器脱机，几个数据供应商服务器提供安全故障转移。有关多层次复制的更多信息，请参阅 TBA[Designing the replication]。

使用同步，您可以将目录服务器用户、组、属性和密码与 Microsoft Active Directory 用户、组、属性和密码集成。如果您有两个目录服务，则决定是否管理相同的信息、这些信息量将共享，哪些服务将提供数据。最好选择一个应用程序来管理数据，并允许同步过程在其他服务上添加、更新或删除条目。

如果您使用与目录间接通信的应用程序，请考虑数据的供应商源。使数据更改过程尽可能简单。决定管理数据的位置后，使用相同的位置来管理其中包含的所有其他数据。当数据库在企业之间丢失同步时，单一位置简化了故障排除。

您可以采用以下方法提供数据：

- 管理目录以及不使用目录的所有应用程序中的数据。
  维护多个数据供应商不需要自定义脚本传输数据。在这种情况下，某人必须更改所有其他站点中的数据，以防止企业的数据重新同步，但这针对目录目的。
- 在非目录应用程序中管理数据，并编写脚本、程序或网关来将该数据导入到目录中。
  当您已使用应用程序管理数据时，在非目录应用程序中管理数据是最理想的选择。此外，您还将对目录进行查找，例如，对于在线企业电话图书。

如何维护数据的主副本取决于特定目录的需求。但是，始终保持维护简单且一致。例如，请勿尝试在多个位置管理数据，然后在竞争应用程序之间自动交换数据。这样做会导致更新丢失并增加管理开销。

例如，该目录管理一个员工的家电话号码，该号码同时存储在 LDAP 目录和人工资源数据库中。人工资源应用程序是启用了 LDAP 的，可以将数据从 LDAP 目录自动传输到人工资源数据库，反之亦然。

如果您试图管理 LDAP 目录和人力资源数据库中该员工电话号码的更改，那么电话号码被更改的最后位置会覆盖其他数据库中的信息。只有编写数据的最后一个应用程序具有正确的信息时，才能接受。

如果该信息已过时（例如，由于人工资源数据是从备份中恢复），则将删除 LDAP 目录中的正确电话号码。

2.3.6. 确定数据所有权

*数据所有权*指的是负责确保数据最新状态的人员或组织。在数据设计阶段，决定谁可以将数据写入目录。以下是决定数据所有权的一些常见策略：

- 允许对除少量目录内容管理器之外的每个目录进行只读访问。
- 允许个人用户管理其战略性信息子集，如其密码、他们在机构中的角色、其自主许可证号码以及联系信息，如电话号码或办公室号码、描述信息。
- 允许个人经理编写该人员信息的战略子集，如联系信息或职位。
- 允许机构管理员创建和管理该机构的条目，使他们能够作为目录内容管理器运行。
- 创建赋予用户读取或写入访问权限组的角色。您可以为人力资源、财务或核算创建角色。允许这些角色对组需要的数据具有读取访问权限、写入访问权限或两者。这可包括工资信息、政府标识号以及主页电话号码和地址。

多个个人可能需要对同一信息进行写访问。例如，信息系统或目录管理组可能需要对员工密码进行写权限。此外，员工还需要对自己的密码进行写入访问权限。虽然多个人可以访问同一信息，但尽量使该组保持小且可识别性，以确保数据完整性。

其他资源

- 分组目录条目
- 设计安全目录

2.3.7. 确定数据访问

确定数据所有权后，决定谁获得读取每个数据的访问权限。例如，员工主页电话号码可以存储在 目录中。此数据对许多用户（包括员工经理和人力资源部门）非常有用。员工应该能够读取该信息以进行验证。但是，家联系信息可被视为敏感。

对于目录中存储的每个信息，请考虑以下几点：

- 某人是否可以匿名读取数据？
  LDAP 协议支持匿名访问，并允许轻松查找信息。但是，由于这种匿名情况（任何人都可以访问该目录），因此优先使用此功能。
- 某人能否广泛读取整个企业的数据？
  您可以设置访问控制客户端必须登录到（或绑定到）目录才能读取特定信息的方式。与匿名访问不同，这种类型的访问控制可确保只有机构成员有权访问目录信息。此外，Directory 服务器访问日志包含有关谁访问信息的记录。

  有关访问控制的更多信息，请参阅 指定访问控制。

- 是否有必须访问数据的人员或应用程序的可识别组？
  对数据具有写入特权的任何人都需要读访问权限（除密码写入访问权限除外）。目录也可以包含特定于特定机构或项目组的数据。识别这些访问权限需要有助于确定哪些组、角色和访问控制。

  有关组和角色的详情，请参考 指定目录树。有关访问控制的详情，请参考指定访问控制。

对每个目录数据进行这些决策会定义目录的安全策略。这些决策取决于站点的性质，以及网站上已提供的安全性。例如，允许防火墙或无法直接访问互联网意味着，如果目录直接放置在互联网上，就很难支持匿名访问。此外，一些信息可能只需要访问控制和验证措施来限制访问。其他敏感信息可能需要在数据库中加密。

大多数国家的数据保护法律规定了企业如何维护和访问个人信息。例如，法律可能会禁止匿名访问信息，或者要求用户能够查看和编辑代表它们的条目中的信息。与组织法律部门进行检查，以确保目录部署符合企业运营的国家的数据保护法律。

在设计 安全目录 中详细介绍创建安全策略及其实现方式。

在复制中，*使用者服务器* 或 *副本服务器* 接收来自供应商服务器或 hub 服务器的更新。

2.4. 记录站点调查

由于数据设计的复杂性，记录站点调查的结果。站点调查的每个步骤都可以使用简单表来跟踪数据。您可以构建一个概述决策和有效关注的供应商表。最好使用电子表格，您可以在其中轻松排序和搜索内容。

下表标识站点调查标识的每个数据的所有权和数据访问权限。

表 2.1. 示例：标记数据所有权和权限

数据名称	所有者	供应商服务器/应用程序	自助读取/写	全局读	HR Writable	IS Writable
员工名称	HR	PeopleSoft	只读	是（匿名）	是	是
用户密码	IS	Directory US-1	读/写	否	否	是
主页电话号码	HR	PeopleSoft	读/写	否	是	否
员工位置	IS	Directory US-1	只读	是（必须登录）	否	是
办公室电话号码	设施	电话交换机	只读	是（匿名）	否	否

表中的每一行指出正在评估的信息类型、对其感兴趣的部门，以及如何使用和访问信息。例如，在第一行中，*employee names* 数据具有以下管理注意事项：

- *Owner*。人员资源拥有此信息，因此负责其更新和更改。
- *Supplier Server/Application*。Soft 应用管理员工名称信息。
- *Self Read/Write*。可以读取自己的名称，但不能读取（或更改）它。

- *Global Read*。员工名称可以被有权访问该目录的任何人匿名读取。
- *HR 可写*。人员资源组成员可以更改、添加和删除目录中的员工名称。
- *IS 可写*.信息服务(IS)组成员可以更改、添加和删除目录中的员工名称。

2.5. 重复站点调查

您可能需要多个站点调查，特别是当一个企业在多个城市或国家设有办事处时。信息性需求可能很复杂，有些不同组织必须将其信息保存在本地办事处，而不是单一的集中站点。

在这种情况下，每个办公室都会保留一份主要信息副本才能执行自己的站点调查。完成站点调查后，每个调查的结果应返回给中央团队（包括每个办公室的代表），以便在整个企业的数据架构模型和目录树的设计中使用。

---

[1] 在复制中，*使用者服务器* 或 *副本服务器* 接收来自供应商服务器或 hub 服务器的更新。

第 3 章 设计目录模式

目录架构描述了目录中数据类型。当您知道目录中存储的数据表示时，您可以确定要使用哪种模式。在架构设计过程中，每个数据元素都映射到 LDAP 属性，相关元素则收集到 LDAP 对象类中。设计良好的模式有助于维护目录数据的完整性。

3.1. 架构设计流程概述

您可以在架构设计过程中选择并定义对象类和属性，以表示 Directory Server 存储的条目。以下步骤在模式设计过程中执行：

- 选择预定义的架构元素来满足数据要求。
- 扩展标准目录服务器架构以定义新的元素以满足要求。
- 规划架构维护。

您可以使用 Directory Server 提供的标准模式中定义的现有 schema 元素。标准架构元素有助于确保与启用了目录的应用程序兼容。架构由大量目录用户审核并同意，因为 schema 基于 LDAP 标准。

3.2. STANDARD 模式

目录架构通过对 data 值的大小、范围和格式设置约束来维护目录中存储的数据的完整性。该架构标识目录包含的不同类型条目（如人员、设备和组织）以及每个条目可用的属性。

目录服务器中的预定义模式包含标准的 LDAP 模式和应用程序特定模式，以支持服务器的功能。您可以通过添加新的对象类和属性来满足目录的唯一需求来扩展架构。

3.2.1. 模式格式

目录服务器的模式格式基于 LDAP 协议的版本 3 构建。此协议要求目录服务器通过 LDAP 本身发布其模式，允许目录客户端应用程序以编程方式检索架构并调整其行为。您可以在 **cn=schema** 条目中找到 Directory 服务器的全局模式集合。

目录服务器模式与 LDAPv3 模式不同，因为它使用其专有对象类和属性。另外，它使用名为 **X-ORIGIN 389 Directory Server** 的 schema 条目中的私有字段，它描述了最初定义了 schema 条目的位置。

当您在标准 LDAPv3 模式中定义模式条目时，**X-ORIGIN 389 Directory Server** 字段指的是 RFC 2252。如果红帽为 Directory Server 使用定义了该条目，则 **X-ORIGIN 389 Directory Server** 字段包含值 **389 Directory Server**。例如，标准 person 对象类出现在 schema 中：

```
# objectclasses: ( 2.5.6.6 NAME 'person' DESC 'Standard Person Object Class' SUP top
MUST (objectclass $ sn $ cn) MAY (description $ seeAlso $ telephoneNumber $ userPassword)
X-ORIGIN 'RFC 2252' )
```

这个模式条目状态：

- 类的对象标识符(OID) (**2.5.6.6**)
- 对象类的名称(**个人**)
- 类的描述(**标准角色**)
- 所需的属性(**objectclass、sn 和 cn**)

- 可选属性(**描述**、**seeAlso**、**telephoneNumber** 和 **userPassword**)

3.2.2. 标准属性

属性包含特定的数据元素，如名称或传真号码。目录服务器将数据表示为属性数据对，它是一个与特定信息片段关联的描述性 schema 属性。它们也称为 **attribute-value assertions** 或 **AVAs**。

例如，目录可以将一块数据（如人的名称）存储在带有 standard 属性的对中。名为 **Babs Jensen** 的人员的条目具有 properties-data 对 **cn: Babs Jensen**。

整个条目以一系列属性-数据对表示。Babs Jensen 的整个条目如下：

```
dn: uid=bjensen,ou=people,dc=example,dc=com
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
cn: Babs Jensen
sn: Jensen
givenName: Babs
givenName: Barbara
mail: bjensen@example.com
```

模式中的每个属性定义都包含以下信息：

- 唯一名称
- 属性的对象标识符(OID)
- 属性的文本描述
- 属性语法的 OID
- 表示以下内容：
  a. 属性是单值或多值
  b. 属性用于目录自己的使用
  c. 属性的来源
  d. 与属性关联的其他匹配规则。

cn 属性定义会出现在 schema 中，如下所示：

```
attributetypes: ( 2.5.4.3 NAME 'cn' DESC 'commonName Standard Attribute'
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )
```

使用属性的语法，您可以定义可以存储在属性中的值的格式。目录服务器支持所有标准属性语法。

其他资源

- 支持的 LDAP 属性语法

3.2.3. 标准对象类

对象类代表一个实际对象，如个人或传真机器。它用于对相关的信息进行分组。在使用对象类前，您必须在 schema 中识别对象类及其属性。默认情况下，该目录识别对象类的标准列表。

每个目录条目都至少属于一个对象类。当您将对象类放在条目上的 schema 中时，它会通知 Directory Server 具有一组特定的属性值，且必须具有另一个较小的所需属性值。

对象类定义中提供了以下信息：

- 唯一名称
- 对象标识符(OID)
- 一组强制属性
- 组允许或可选属性

模式中的标准 person 对象类：

```
objectclasses: ( 2.5.6.6 NAME 'person' DESC 'Standard Person Object Class' SUP top
    MUST (objectclass $ sn $ cn) MAY (description $ seeAlso $ telephoneNumber $ userPassword)
    X-ORIGIN 'RFC 2252' )
```

注意

您可以使用标准 LDAP 操作查询和更改目录模式，因为对象类是直接定义并存储在目录服务器中的。

其他资源

- 对象类的标准列表

3.3. 将数据映射到默认模式

您必须将站点调查期间标识的数据映射到现有的默认目录 schema。如果 schema 中的元素与现有默认模式不匹配，您可以创建自定义对象类和属性。

默认目录模式存储在 **/usr/share/dirsrv/schema/** 目录中，其中包含 Directory 服务器的所有通用模式。您可以在 **00core.ldif** 文件中找到 LDAPv3 标准用户和机构模式。您还可以在 **50ns-directory.ldif** 文件中找到早期版本的目录使用的配置模式。

警告

不要修改默认目录模式。

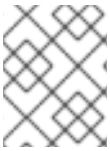
3.3.1. 与 schema 元素匹配的数据

您可以将站点调查中标识的数据映射到现有的目录模式。这个过程涉及以下步骤：

- 您必须识别数据所描述的对象类型。

有时，一种数据可以描述多个对象。确定在目录 schema 中是否需要记录区别。例如，电话号码可以描述员工的电话号码和会议房间电话号码。确定这些不同类型的数据是否需要被视为目录架构中不同的对象。

- 您必须从默认 schema 中选择一个类似的对象类。最好使用通用对象类，如组、人员和机构。
- 您必须从匹配的对象类中选择一个类似的属性。
- 您必须从站点调查中识别不匹配的数据。如果某些数据与默认目录架构定义的对象类和属性不匹配，请自定义 schema。



注意

目录服务器配置、命令和文件参考有助于确定可用于您的数据的属性。每个属性都列出接受它的对象类，每个对象类被列为必需属性和允许的属性。

其他资源

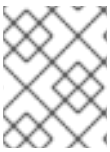
- Red Hat Directory Server 配置、命令和文件参考

3.4. 自定义模式

您可以通过添加属性和对象类，在 Directory Server 中使用 Web 控制台来扩展标准模式。您还可以创建 LDIF 文件并手动添加 schema 元素。

在自定义模式时，以下规则适用：

- 您必须保持架构简单。
- 您必须重复使用 schema 元素。
- 您必须最小化为每个对象类定义的强制属性数量。
- 不要为同一目的（数据）定义多个对象类或属性。
- 不要修改属性或对象类的任何现有定义。



注意

在自定义 schema 时，您无法删除或替换标准模式。这样做可能会导致与其他目录或 LDAP 客户端应用程序兼容性。

自定义对象类和属性在 **99user.ldif** 文件中定义。每个实例在 **/etc/dirsrv/slapd-*instance_name*/schema/** 目录中维护自己的 **99user.ldif** 文件。您还可以创建自定义模式文件，并将架构动态重新加载到服务器中。

当给定对象类无法存储机构的专用信息时，您可以扩展架构，而 Directory 服务器提供的对象类和属性应该满足最常见的企业需求。您还可以扩展架构，以支持支持支持支持支持支持支持支持支持 LDAP 应用的唯一数据需要的对象类和属性。

3.4.1. 分配对象标识符

您必须为每个 LDAP 对象类或属性分配唯一名称和 **对象标识符** (OID)。当您定义架构时，元素需要您的机构的唯一基本 OID。添加另一个层次结构级别，以便为属性和对象类创建新分支。在 schema 中获取和分配 OID 涉及以下步骤：

1. 从互联网分配号机构(**IANA**)或国家机构获取 OID。在某些国家/地区，公司已经为他们分配了 OID。

2. 创建 OID registry 来跟踪 OID 分配。OID registry 是目录 schema 中使用的 OID 和描述的列表。这样可确保没有 OID 用于多个目的。然后使用 schema 发布 OID 注册表。

3. 在 OID 树中创建分支以容纳 schema 元素。在 OID 分支或目录架构下至少创建两个分支，将 OID.**1** 用于属性，而 OID.**2** 用于对象类。根据需要添加新分支，以定义自定义匹配规则或控制（如 OID.'3）。

其他资源

- IANA

3.4.2. 定义新对象类的策略

您可以通过以下两种方式创建新对象类：

- 创建新对象类，一个用于可以添加属性的每个对象类结构。

- 创建一个对象类，它支持为目录创建的所有自定义属性。您可以通过将对象类定义为辅助对象类来创建此对象类。

您可以混合使用这两种方法。例如，您要创建属性 **exampleDateOfBirth**,**examplePreferredOS**,**exampleBuildingFloor**, 和 **exampleVicePresident**。简单的解决方案是创建多个对象类，允许其中的一些部分属性。

- **examplePerson** 对象类允许 **exampleDateOfBirth** 和 **examplePreferredOS**。**examplePerson** 的父对象是 **inetOrgPerson**。

- **exampleOrganization** 对象类允许 **示例BuildingFloor** 和 **exampleViceP located**。exampleOrganization 的父对象是 **机构** 对象类。

新对象类以 LDAPv3 模式格式出现，如下所示：

```
objectclasses: ( 2.16.840.1.117370.999.1.2.3 NAME 'examplePerson' DESC 'Example Person
Object Class'
    SUP inetorgPerson MAY (exampleDateOfBirth $ examplePreferredOS) )

objectclasses: ( 2.16.840.1.117370.999.1.2.4 NAME 'exampleOrganization' DESC 'Organization
Object Class'
    SUP organization MAY (exampleBuildingFloor $ exampleVicePresident) )
```

或者，您可以创建一个单一对象类来允许所有这些属性，并将其与需要它们的任何条目一起使用。单个对象类显示如下：

```
objectclasses: (2.16.840.1.117370.999.1.2.5 NAME 'exampleEntry' DESC 'Standard Entry Object
Class' SUP top
    AUXILIARY MAY (exampleDateOfBirth $ examplePreferredOS $ exampleBuildingFloor $
exampleVicePresident) )
```

新的 **exampleEntry** 对象类标记为 **AUXILIARY**，这意味着它可用于任何条目，而不考虑其结构对象类。

您可以根据机构环境来组织新对象类。决定新对象类的实现时请考虑以下几点：

- 如果架构中添加了超过两个或三个对象类，则必须使用单个对象类。
- 多个对象类需要严格的数据设计。严格数据设计强制注意对象类结构，其中放置每个数据的对象类结构很有用或繁琐。
- 当数据可应用于多个对象类（如人员和资产条目）时，您可以使用单个对象类使用数据。例如，您可以在个人和组条目上设置自定义 **preferredOS** 属性。单个对象类可以在这两类条目上允许此属性。
- 您必须避免新对象类所需的属性。当您指定 **require** 而不是 **允许** 新对象类中的属性时，它可以使架构变得不灵活。在定义了新对象类后，决定其允许和需要哪些属性，以及它继承属性的对象类。

3.4.3. 定义新属性的策略

您必须将标准属性用于应用程序兼容性和长期维护。您必须搜索默认目录中已存在的属性，并将它们与新对象类一起使用，或检查 Directory Server 架构指南。但是，如果标准模式不包含所有必要的信息，请添加新的属性和新对象类。

例如，一个个人条目可能需要比个人、**organizationalPerson** 或 **inetOrgPerson** 对象类默认支持更多的属性。标准目录服务器模式中没有属性来存储生日期。您可以在新的辅助对象类 **examplePerson** 中创建和设置新属性 **dateOfBirth** 作为允许的属性：

```
attributetypes: ( dateofbirth-oid NAME 'dateofbirth' DESC 'For employee birthdays'
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN 'Example defined')

objectclasses: ( 2.16.840.1.117370.999.1.2.3 NAME 'examplePerson' DESC 'Example Person
Object Class'
    SUP inetorgPerson MAY (exampleDateOfBirth $ cn) X-ORIGIN 'Example defined')
```

注意

您不能在标准 schema 元素中添加或删除自定义属性。如果目录需要自定义属性，请添加自定义对象类使其包含它们。

3.4.4. 删除模式元素

您无法删除目录服务器中默认包含的 schema 元素。未使用的模式元素并不代表操作或管理开销。删除标准 LDAP 模式的部分，可能会导致与将来的 Directory Server 安装和其他启用了目录的应用程序的兼容性问题。

但是，您可以删除未使用的自定义 schema 元素。在从 schema 中删除对象类定义前，请使用对象类修改每个条目。首先删除定义可能会阻止使用对象类的条目被修改。修改条目的 schema 检查也会失败，除非从条目中删除了未知对象类值。

3.4.5. 创建自定义模式文件

除了 Directory Server 提供的 **99user.ldif** 文件外，您还可以为 Directory 服务器创建自定义架构文件。这些架构文件具有特定于组织的新自定义属性和对象类。新架构文件位于 schema 目录中，**/etc/dirsrv/slapd-instance_name/schema/** 中。所有标准属性和对象类仅在加载自定义 schema 元素后加载。

注意

自定义架构文件不能以数字方式或字母顺序高于 **99user.ldif**。

创建自定义模式文件后，模式更改可以通过以下方式分布到所有服务器中：

- 您可以将这些自定义模式文件复制到实例的 schema 目录 **/etc/dirsrv/slapd-instance/schema** 并加载 schema，通过运行 **schema-reload.pl** 脚本来动态重新载入 schema。
- 您可以使用 LDAP 客户端（如 Web 控制台）或使用 **ldapmodify** 命令修改服务器上的模式。
- 使用复制时，所有复制模式元素都复制到使用者服务器 **99user.ldif** 文件中。要将架构保留在自定义架构文件中，如 **90example_schema.ldif**，必须手动将文件复制到消费者服务器。复制不会复制架构文件。

当您不将这些自定义模式文件复制到所有服务器时，只有在供应商服务器上的 schema 更改时，架构信息才会复制到消费者服务器中。当架构定义复制到尚不存在的消费者服务器时，它们会存储在 **99user.ldif** 文件中。

注意

目录无法跟踪存储架构定义的位置。如果只对供应商服务器维护 schema，您可以在消费者的 **99user.ldif** 文件中存储 schema 元素。

3.4.6. 自定义模式的最佳实践

以下建议可帮助您定义兼容和可管理的自定义模式。

命名架构文件

以数字方式命名自定义模式文件，按字母顺序低于 **99user.ldif**。**99user.ldif 文件包含** 带有 **X-ORIGIN** 值 'user defined' 的属性。目录服务器将所有"用户定义的"模式元素写入最高命名的文件，然后按字母顺序将。如果架构文件的名称是 **99zzz.ldif**，并且更新了 schema，则所有 **X-ORIGIN** 值为 'user defined' 的属性都会写入 **99zzz.ldif** 文件。因此，包含两个包含重复信息的 LDIF 文件，以及 **99zzz.ldif 文件中** 的一些信息可能会被删除。

在命名自定义模式文件时，使用以下命名格式 **：[00-99]yourName.ldif**。

使用 'user defined' 作为原始卷

不要在自定义模式文件的 **X-ORIGIN** 字段中使用 'user defined'，如 **60example.ldif**，因为在通过 LDAP 添加架构时，目录服务器在内部使用 'user defined'。

如果自定义架构元素直接添加到 **99user.ldif**，则使用 'user defined' 作为 **X-ORIGIN** 的值。如果设置了不同的 **X-ORIGIN** 值，则服务器只需覆盖它即可。

使用值 'user defined' 的 **X-ORIGIN** 会阻止 Directory Server 删除 **99user.ldif** 文件中的 schema 定义。

例如：

```
attributetypes: ( exampleContact-oid NAME 'exampleContact'
DESC 'Example Corporate contact'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
X-ORIGIN 'Example defined')
```

目录服务器加载 schema 条目后，它如下所示：

```
attributetypes: ( exampleContact-oid NAME 'exampleContact'
DESC 'Example Corporate contact'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
X-ORIGIN ('Example defined' 'user defined') )
```

在对象类前定义属性

当您添加新的 schema 元素时，请在对象类中使用所有属性前定义所有属性。属性和对象类可以在相同的架构文件中定义。

在单个文件中定义架构

在一个模式文件中定义每个自定义属性或对象类，以防止服务器在加载最近创建的模式时覆盖任何以前的定义。服务器首先以数字顺序加载模式，然后是字母顺序。决定如何在重复文件中保留模式：

- 请注意每个架构文件中包括哪些 schema 元素。
- 在命名和更新 schema 文件时要小心。通过 LDAP 工具编辑模式元素时，更改将按字母顺序自动写入最后一个文件。大多数架构更改都会写入默认的 **99user.ldif** 文件，而不是自定义模式文件，如 **60example.ldif**。**99user.ldif** 文件中的 schema 元素覆盖其他模式文件中的重复元素。
- 如果使用 Web 控制台管理架构，请将所有架构定义添加到 **99user.ldif** 文件中。

3.5. 致一致性模式概述

LDAP 客户端应用程序通过在目录服务器中使用一致的模式查找目录条目。您不能使用不一致的模式在目录树中找到信息，因为它使用不同的属性或格式来存储相同的信息。

您可以使用以下方法维护模式一致性：

- 您可以使用架构检查来确保属性和对象类确认架构规则。
- 您可以使用语法验证来确保属性值与所需属性语法匹配。
- 您可以选择并应用一致的数据格式。

3.5.1. 架构检查

模式检查会验证所有新的或修改的目录条目是否都符合 schema 规则。默认情况下，目录启用模式检查。当违反规则时，目录会拒绝请求的更改。

注意

架构检查会验证是否存在正确的属性。您可以使用语法验证来验证属性值是否采用正确的语法。不要禁用此功能。

启用架构检查后，您必须注意对象类定义的必需属性和允许的属性。当您向条目的对象类定义添加属性时，目录服务器可以返回对象类违反消息。

例如，如果将条目定义为使用 **organizationalPerson** 对象类，该条目需要通用名称(**cn**)和 surname (**sn**)属性。在创建条目时，必须设置这些属性的值。另外，还有可在该条目上使用的属性列表，包括 **telephoneNumber**、**uid**、**streetAddress** 和 **userPassword** 等描述性属性。

3.5.2. 语法验证概述

语法验证意味着目录服务器验证属性值是否与该属性的必要语法匹配。例如，语法验证可以确认新的 **telephoneNumber** 属性具有值的有效电话号码。它会被默认启用。

您可以选择为语法验证配置额外的设置，以记录有关语法违反情况的警告信息，然后拒绝修改或允许修改过程成功。

如果添加了新属性值，语法验证会检查 LDAP 操作。它不会处理通过复制等数据库操作添加的现有属性或属性。可以使用 **dsconf** schema validate-syntax 命令验证现有属性。

此功能验证除二进制语法和非标准语法之外的所有属性语法，它们没有定义的必要格式。语法会根据 **RFC 4514** 验证，但 DN 除外，它们会根据不太严格的 **RFC 1779** 或 **RFC 2253** 验证。

注意

您可以配置严格的 DN 验证。

3.5.2.1. 目录服务器操作的语法验证

语法验证适用于标准 LDAP 操作，如创建条目（添加）或编辑属性(modify)。当您验证属性语法时，它可能会影响其他目录服务器操作。

数据库加密

您可以在值写入 LDAP 操作数据库中前加密属性。这意味着，加密是在验证属性语法后执行的。您可以导入和导出加密的数据库。

注意

您必须使用标志- **encrypted (dsctl)** 执行导出和导入操作，这允许导入操作进行语法验证。

如果您在不使用加密 **标志（不支持）的情况下导出加密** 数据库（不支持），则会创建一个带有加密值的 LDIF。您无法验证加密的属性，会记录警告，在导入 LDIF 时跳过属性验证。

同步

对于 Windows Active Directory 条目和 Directory Server 条目中的属性，允许或强制实施语法。您无法同步 Active Directory 值，因为语法验证强制执行 Directory 服务器条目中的 RFC 标准。

复制

如果目录服务器 11.0 实例是将更改复制到消费者的供应商，您可以使用语法验证。但是，假设复制中的供应商是旧版本的目录服务器，或者禁用了语法验证。在这种情况下，无法对 11.0 使用者使用语法验证，因为 Directory 服务器 11.0 使用者可能会拒绝供应商允许的属性值。

3.5.3. 一致的数据格式

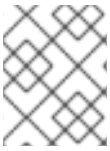
您可以使用 LDAP 模式将数据带有属性值来放置数据。但是，务必要通过选择适合 LDAP 客户端应用程序和目录用户的格式，将数据存储在目录树中。

您可以使用 LDAP 协议和目录服务器以 **RFC 2252** 中指定的数据格式表示数据。例如，在两个 **ITU-T** 建议文档中定义了电话号码的正确 LDAP 格式：

- **ITU-T 建议 E.123**.国家和国际电话号码的表示法。
- **ITU-T 建议 E.163**.国际电话服务的编号计划.例如，美国电话号码格式为 **+1 555 222 1717**。

再如，**postalAddress** 属性以多行字符串的形式有一个属性值，其使用美元符号($)作为行分隔符。正确格式化的目录条目如下所示：

```
postalAddress: 1206 Directory Drive$Pleasant View, MN$34200
```



注意

属性需要字符串、二进制输入、整数和其他格式。您可以在属性的 schema 定义中设置格式。

3.5.4. 关于在复制模式下保持一致性

编辑目录架构时，这些更改记录在 changelog 中。在复制过程中，会扫描 changelog 是否有更改，以及正在复制任何更改。在复制模式下保持一致性，复制可以在没有任何错误的情况下继续。

在复制环境中为保持一致性模式考虑以下点：

- 不要修改只读副本上的模式。
  当您修改只读副本中的模式时，它会在 schema 中引入不一致，并导致复制失败。
- 不要创建具有相同名称的属性，它们使用不同的语法。
  当您在读写副本中创建属性时，其名称与供应商副本上的属性相同，但使用不同于供应商的属性的语法时，复制将失败。

3.6. 其他资源

- RFC 2251：轻量级目录访问协议(v3)
- RFC 2252: LDAPv3 属性语法定义
- 用于 LDAPv3 的 X.500 用户 Schema 的 RFC 2256: 概述

第 4 章 设计目录树

您可以使用目录树查看存储在目录服务器中的数据。目录树的设计基于目录中存储的信息类型、企业的物理性质、用于目录的应用程序以及实施的复制类型。

4.1. 目录树简介

您可以使用目录树命名目录数据并引用客户端应用程序。目录树可以与其他设计决策交互，包括可用于分发、复制或控制对目录数据的访问的选项。您可以在部署之前设计目录树，以便在部署阶段和稍后操作期间减少时间和精力。

使用精心设计好的目录树，您可以：

- 只需维护目录数据即可。
- 灵活地创建复制策略和访问控制。
- 使用目录服务支持应用程序。
- 简化用户的目录导航。

目录树的结构遵循分层 LDAP 模型。目录树提供了一种方式来组织不同的逻辑方式，如按组、人员或位置来组织数据。您还可以使用目录树来确定如何在多个服务器间对数据进行分区。例如，每个数据库都需要在后缀级别上对数据进行分区。在没有正确的目录树结构的情况下，您无法有效地将数据分散到多个服务器上。

此外，复制受到使用的目录树结构的类型的限制。当您只想复制目录树的部分时，请在设计过程中考虑这一点。

4.2. 设计目录树

在规划目录树时，您可以做出以下主要决策：

- 您可以选择包含数据的后缀。
- 您可以通过创建目录树结构来确定数据条目之间的分级关系。
- 您可以命名目录树层次结构中的条目。

4.2.1. 选择后缀

后缀是目录树根目录下的条目名称，目录数据存储在其中。目录可以包含多个后缀。如果有两个或者多个没有自然常见的 root 信息树，您可以使用多个后缀。默认情况下，标准目录服务器部署包含多个后缀，一个用于存储数据，另一个用于存储内部目录操作所需的数据，如配置信息和目录模式。

命名后缀的惯例

您应该在常用基础条目(*根后缀*)中找到 目录中的所有条目。当您为根目录后缀选择名称时，要使名称有效，必须为：

- 全局唯一
- Static
- 短，以便您可以轻松读取其下的条目

- 便于人键入和记住

在单一企业环境中，您可以选择与企业 DNS 名称或互联网域名一致的目录后缀。例如，如果企业拥有 **example.com** 的域名，则目录后缀为 **dc=example,dc=com**。**dc** 属性通过将域名拆分到其组件部分来代表后缀。通常，您可以使用任何属性来命名根后缀。但是，对于托管机构，您必须将 root 后缀限制为以下属性：

dc

定义域名的组件。

c

包含代表国家名称的双位代码，由 ISO 定义。

l

标识该条目所在的计数、城市或其他地理位置，或者与该条目相关联。

st

标识条目所在的状态或省去。

o

标识条目所属机构的名称。

这些属性提供与订阅者应用程序的互操作性。例如，托管组织可以使用这些属性来创建根后缀 **o=example_a, st=Washington,c=US** 用于其其中一个客户端 **example_a**。

根据后缀的 **X.500** 命名约定，通常使用机构名称以及国家设计。

命名多个后缀

目录中的每个后缀都是唯一的目录树。您可以创建存储在单独的数据库目录服务器中的多个目录树。

例如，您可以创建单独的后缀，如 **example_a** 和 **example_b**，并将它们存储在单独的数据库中。

490_RHDS_0124

您可以根据资源限制在单一服务器或多个服务器上存储数据库。

4.2.2. 创建目录树结构

决定是否使用平面或分级树结构。尝试尽可能使目录树变为扁平。但是，以后在多个数据库间分区信息、准备复制或设置访问控制时，一定程度的层次结构非常重要。

树的结构涉及以下步骤和注意事项：

- 分支目录
- 识别分支点

- 复制注意事项
- 访问控制注意事项

4.2.2.1. 分支目录

命名空间必须尽量扁平，以避免有问题的名称更改。更容易对目录树、名称中的更多组件以及名称更有可能更改的等级。

使用以下准则来设计目录树层次结构：

- 将树分支，仅代表企业中最大的组织子部门。您应该限制分支指向部门，如公司信息服务、客户支持、销售和工程等。确保用于分支目录树的划分是稳定的。如果企业频繁重组，则不要执行此类分支。
- 对分支点使用功能或通用名称而不是实际的机构名称。当您重命名子树时，如果后缀有许多子项，则名称更改进程是资源密集型和长。例如，使用 **Engineering** 而不是 **Widget Research 和 Development**。
- 如果您有多个执行类似功能的组织，请尝试为该功能创建单个分支点。例如，即使有多个营销机构（每个机构都负责特定的产品行），创建一个 **ou=Marketing** 子树。然后，所有 marketing 条目都属于该树。

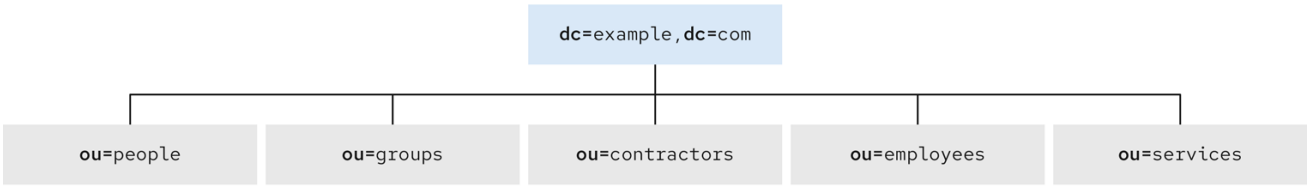
企业环境中的分支

如果您根据可能更改的信息规划目录树结构，可以避免名称更改。例如，如果您将结构基于树中的对象类型，而不是机构。

使用以下通用对象来定义结构：

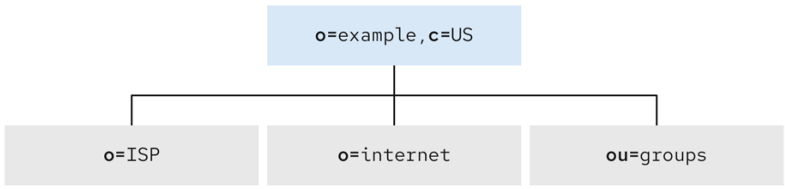
- **ou=people**
- **ou=groups**
- **ou=contracts**
- **ou=services**

下图显示了使用这些对象组织的目录树：



托管环境中的分支

对于托管环境，创建一个树，其中包含对象类 **组织的** 两个条目(**o**)，并在 root 后缀下包含对象类 **organizationalUnit** (**ou**)的条目。例如，名为 **Example ISP** 的互联网服务提供商按以下方式对目录进行分支：

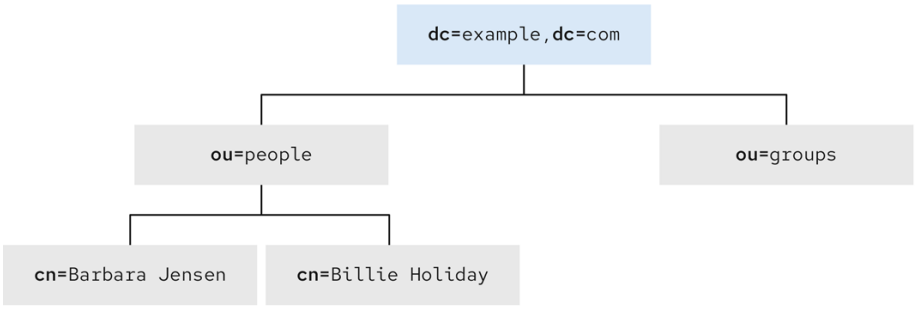


4.2.2.2. 识别分支点

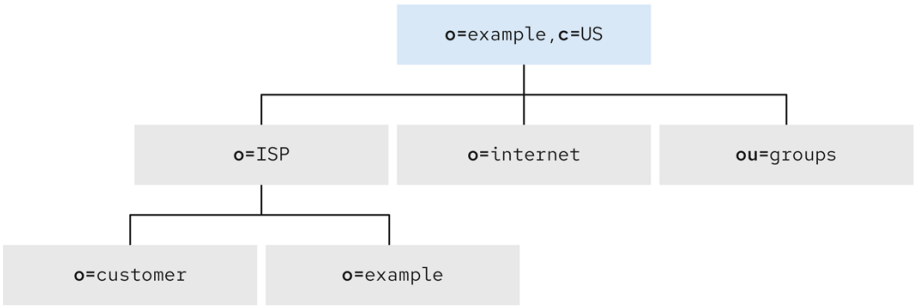
在规划目录树中的分支时，决定使用什么属性来识别分支点。分支点是一个属性数据对，如 **ou=people,l=Japan,cn=Barbara Jansen**。请记住，DN 是由这些属性数据对组成的唯一字符串。例如，Barbara Jensen 条目的 DN 是 Example Company 的员工，如下所示：

uid=bjensen,ou=people,dc=example,dc=com.

请参阅下图中带有 **ou=people,ou=groups,cn=Barbara Jensen,cn=Billie Holiday** 分支点的示例。



请参阅下图中的互联网供应商示例 ISP 的目录树示例：



在 root 后缀条目 **o=example,c=US** 下，树被分成三个分支。**o=ISP** 分支包含客户数据和 ISP 示例内部信息。**o=internet** 分支是域树。**ou=groups** 分支包含有关管理组的信息。

在为分支点选择属性时请考虑以下建议：

- **保持一致。**
  如果 DN 格式在目录树中不一致，一些 LDAP 客户端应用程序可能无法找到可分辨的名称(DN)。如果目录树的一个部分的 ou 下为 o，则确保 ou 在目录服务的所有其他部分下。

- 尝试只使用传统属性。
  当您使用传统属性时，它会增加目录服务器与第三方 LDAP 客户端应用程序兼容的的可能性。使用传统属性还意味着默认目录架构知道它们。

传统属性	描述
dc	域名的一个元素，如 **dc=example**。它通常以对的形式指定，甚至更长的时间，具体取决于域，如 **dc=example,dc=com** 或 **dc=mtv,dc=example,dc=com**。有关命名域名的更多信息，请参阅命名后缀 部分。
c	国家/地区名称。
o	机构名称。使用此属性代表大型部门分支，如公司部门、教育部门（人类、科学）、子公司或其他主要分支。您可以使用此属性代表域名。
ou	机构单元。使用此属性代表比机构较小的部门分支。组织单元通常属于上述组织。
st	状态或省名称。
L 或 **locality**	本地性，如城市、国家、办公室或设施名称。

注意

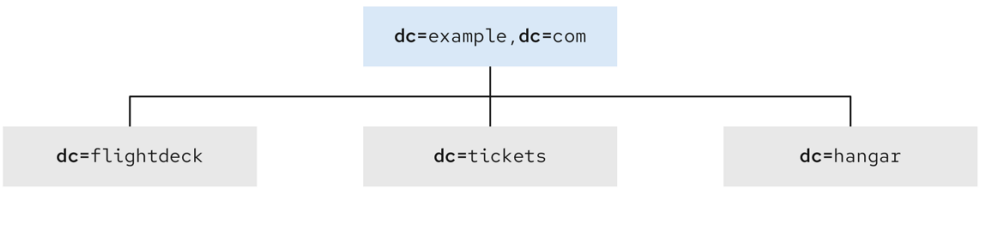
常见错误是假定根据可分辨名称中使用的属性搜索目录。区分名称只是目录条目的唯一标识符，不能用作搜索键。相反，请根据条目本身中存储的属性对搜索条目。因此，如果条目的可分辨名称为 uid=bjensen,ou=People,dc=example,dc=com，则搜索 dc=example 不匹配那个条目，除非在该条目中明确添加了 dc:example 作为属性。

4.2.2.3. 复制注意事项

规划您要复制的条目。您可以在子树顶部指定 DN，并复制其下的所有条目。此子树也与数据库对应，一个目录部分包含目录数据。

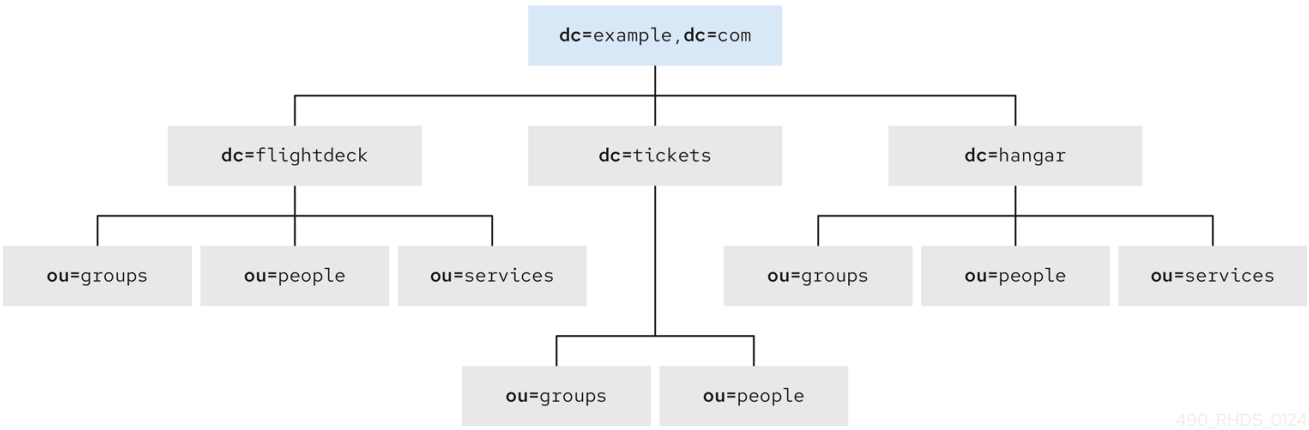
例如，在企业环境中，您可以组织目录树，使其与企业中的网络名称对应。网络名称往往 不会改变，因此目录树结构稳定。

例如，Example Company 有三个主要网络，称为 flightdeck.example.com、ticket.example.com 以及 hangar.example.com。公司最初将其目录树分到其主要组织部门的三个主要组中。参阅下图中目录树的初始分支：

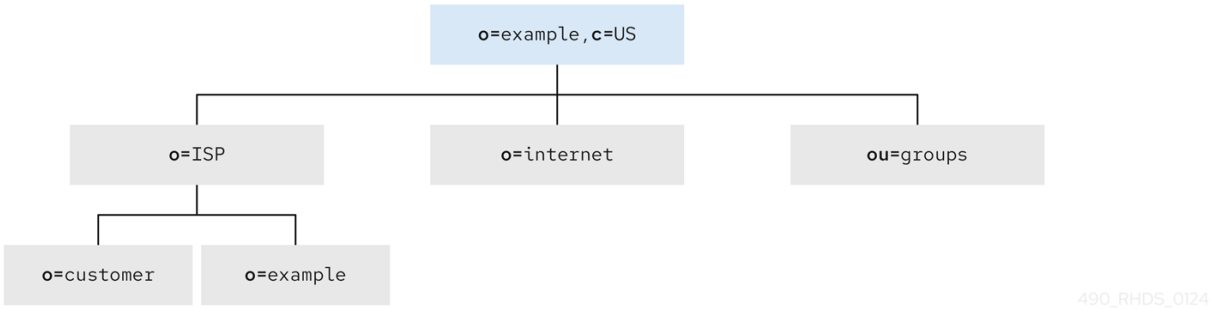


490_RHDS_0124

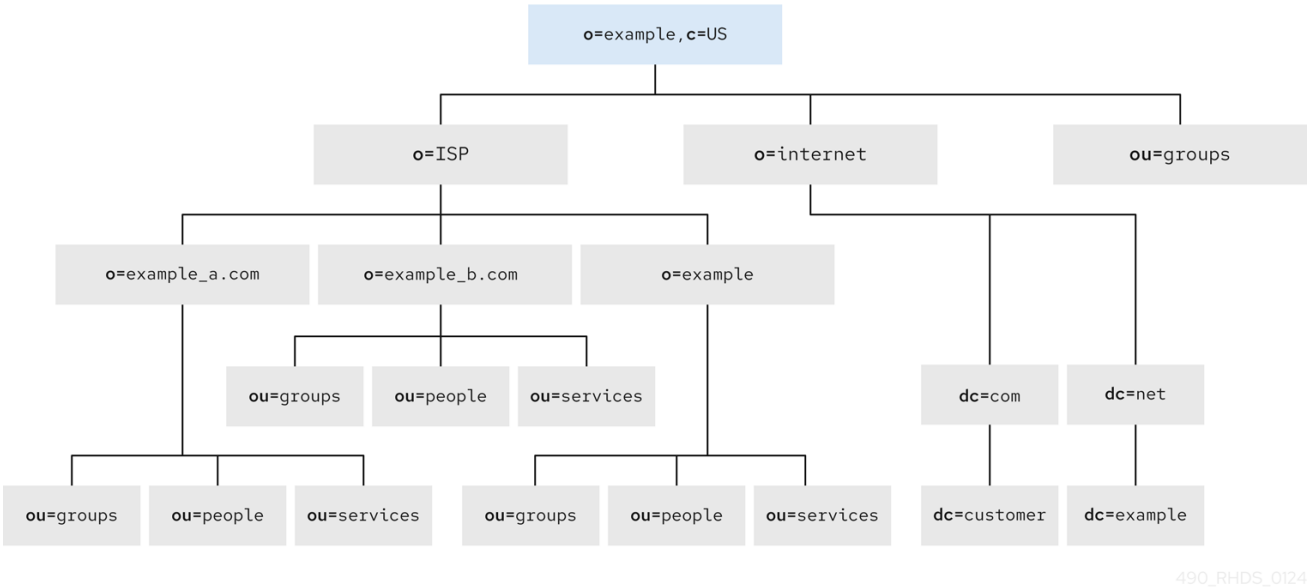
创建树的初始结构后，公司创建额外的分支。请参阅以下图像中的扩展分支：



在另一个示例中，互联网供应商示例 ISP 有以下初始分支来满足供应商需求：



之后，示例 ISP 为逻辑子组创建额外的分支。请参阅以下图像中的扩展分支：



企业示例 Company 和托管组织都根据不经常更改的信息设计其数据层次结构。

4.2.2.4. 访问控制注意事项

您可以使用目录树中的层次结构来启用某些类型的访问控制。与复制一样，可以更轻松地对类似的条目进行分组，然后从单个分支进行管理。

您可以通过分级目录树进行管理。例如，为管理员授予营销部门对营销条目的访问权限，以及来自销售部门对销售条目的管理员访问，根据这些部门对目录树进行设计。

另外，您可以根据目录内容而不是目录树设置访问控制。使用访问控制指令(ACI)机制，您可以允许特定条目访问包含特定属性值的所有条目。例如，设置一个 ACI，为 sales 管理员授予对包含属性值 ou=Sales 的所有条目的访问权限。

但是，ACI 可能很难管理。决定访问控制的最佳方法：组织树层次结构中的分支、ACI 或两者的组合。

4.2.3. 命名条目

在设计目录树的层次结构后，您需要决定在命名结构中的条目时使用哪些属性。当您选择一个或多个属性值时，您可以形成一个 *相对可分辨名称* (RDN)。RDN 是 DN 的最左侧部分，您选择的 *属性是 naming 属性*。naming 属性为条目设置唯一名称。例如，DN uid=bjensen,ou=people,dc=example,dc=com 具有 RDN uid=bjensen。

您选择的属性取决于您命名的条目类型。

在命名条目时请考虑以下几点：

- 您不应更改为命名选择的属性。
- 名称在目录中必须是唯一的。唯一名称可确保 DN 只引用目录中的一个条目。

当您创建条目时，在条目中定义 RDN。通过条目内定义的 RDN，该条目可以更轻松地找到该条目。这是因为，根据条目本身中存储的属性值搜索条目，而不基于实际的 DN。

属性名称具有含义，因此尝试使用与它所代表的输入类型匹配的属性名称。例如，不要使用 l（位置）代表机构，或使用 c (country)代表机构单元。

4.2.3.1. 命名目录树中的 person 条目

个人条目名称必须是唯一的。通常，要命名人员条目，您可以使用 commonName 或 cn、属性组成一个相对可分辨名称(RDN)。例如，名为 Babs Jensen 的人员的条目可能具有可分辨名称(DN)，作为 cn=Babs Jensen,dc=example,dc=com。

请注意，在 RDN 中使用通用名称可能不足使条目名称唯一，并可能会创建几个相同的条目，从而导致 DN 名称冲突。

通过在通用名称中添加唯一标识符来避免常见名称冲突，如 cn=Babs Jensen+employeeNumber=23,dc=example,dc=com。但是，这可能会导致大型目录的通用名称，很难维护。

更好的方法是，使用 cn 以外的某些属性来识别 person 条目。考虑使用以下属性之一：

uid
: 使用 uid 属性指定个人的一些唯一值，如用户登录 ID 或员工号码。通过uid 属性识别托管环境中的订阅者。

mail
: mail 属性包含始终唯一的个人电子邮件地址。此属性可能会导致包含重复属性值的 DN，如 mail=bjensen@example.com,dc=example,dc=com。只有在您找不到 uid 属性的一些唯一值时，才使用这个选项。例如，如果企业没有为临时或合同员工分配员工号码或用户 ID，则使用 mail 属性而不是 uid 属性。

employeeNumber
: 对于 inetOrgPerson 对象类的员工，请使用 employeeNumber 属性。

无论您用于个人条目 RDN 的属性数据对，请确保它们是唯一的永久值。人员条目 RDN 应该也是可读的。例如，DN uid=bjensen,dc=example,dc=com 比 uid=b12r56A,dc=example,dc=com 更首选，它简化了一些目录任务，如根据其可分辨名称更改目录条目。另外，一些目录客户端应用程序假设 uid 和 cn 属性使用人类可读的名称。

托管环境中个人条目的注意事项

如果个人是服务的订阅者，该条目应具有 inetUser 对象类，并且包含 uid 属性。此属性在客户子树中必须是唯一的。

如果个人是托管机构的一部分，请将 inetOrgPerson 属性与 nsManagedPerson 对象类一起使用。

将 person 条目放在目录树中

使用以下准则将 person 条目放置到目录树中：

- 在目录树中的组织条目下，找到企业中的人员。
- 为托管组织的 ou=people 分支找到下的托管组织的订阅者。

4.2.3.2. 目录树中的命名组条目

您可以使用以下方法代表组：

- *静态组* 明确定义其成员。groupOfNames 或 groupOfUniqueNames 对象类包含命名组成员的值。静态组适用于几个成员的组，如目录管理员组，不适用于具有数千个成员的组。
  静态组条目必须包含 uniqueMember 属性值，因为 uniqueMember 是 groupOfUniqueNames 对象的强制属性。此对象类需要 cn 属性，您可以使用它来组成组条目的 DN。
- *动态组* 指定过滤器，与过滤器匹配的所有条目都是此组的成员。
- *角色* 统一静态和动态组概念。

在托管环境中，请考虑使用 groupOfUniqueNames 对象类来包含命名目录管理中使用的组成员的值。

另外，在 ou=Groups 分支下查找用于目录管理的组条目。

其他资源

- 分组目录条目

4.2.3.3. 命名机构条目

组织条目名称必须是唯一的。当您将机构的法律名称与其他属性值一起使用时，有助于确保名称是唯一的，如 o=example_a+st=Washington,o=ISP,c=US。

您还可以使用商标，但它们可能不唯一。

在托管环境中，在机构条目中包含以下属性：

- o (organizationName)
- 带有 top,organization, 和 nsManagedDomain的值

4.2.3.4. 命名其他条目

该目录包含代表不同信息的条目，如本地城市、州、国家、设备、服务器、网络信息和其他数据类型。对于这些类型的条目，使用 RDN 中的 cn 属性。您还可以将组条目命名为 cn=administrators,dc=example,dc=com。

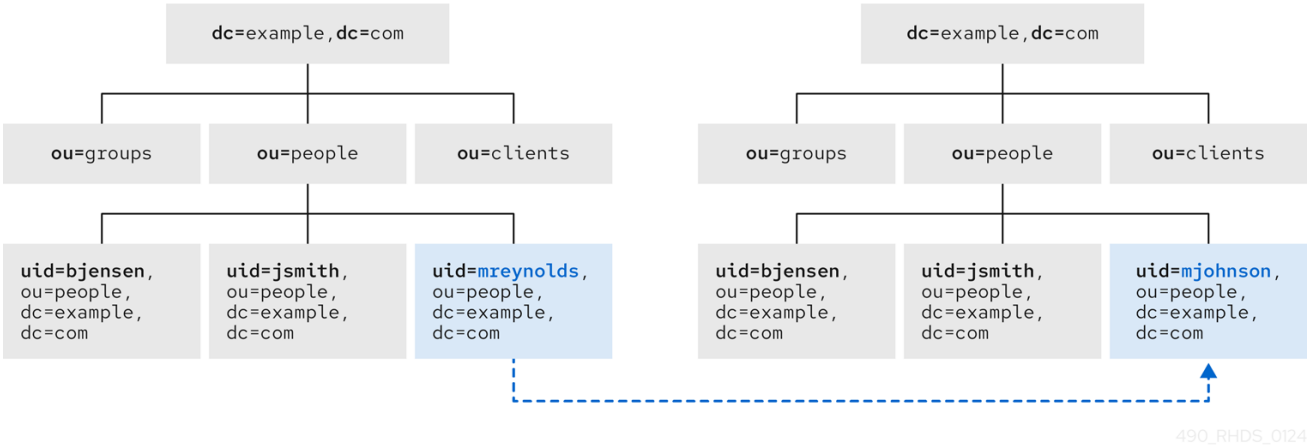
有时，条目对象类不支持 commonName 属性。反之，使用条目对象类支持的属性。naming 属性不必与您在条目中实际使用的属性对应。但是，如果您在条目中使用的 DN 属性和属性之间存在一些关联，则更轻松地管理目录树。

4.2.4. 重命名条目和子树

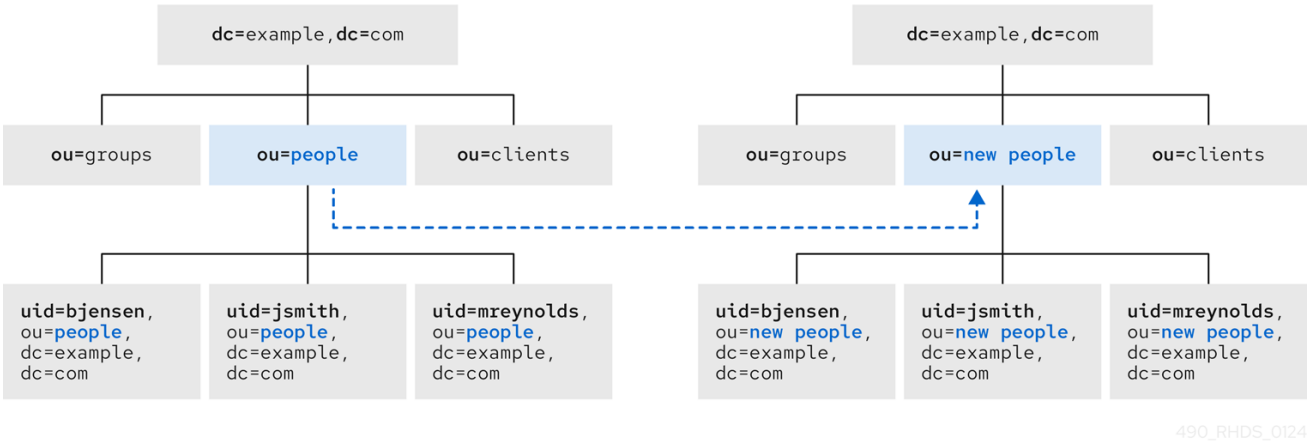
条目名称定义目录树结构。每个分支点在层次结构中创建一个新链接。

```
dc=example,dc=com  =>  root suffix
ou=People,dc=example,dc=com  =>  org unit
st=California,ou=People,dc=example,dc=com  =>  state/province
l=Mountain View,st=California,ou=People,dc=example,dc=com  =>  city
ou=Engineering,l=Mountain View,st=California,ou=People,dc=example,dc=com  =>  org unit
uid=jsmith,ou=Engineering,l=Mountain View,st=California,ou=People,dc=example,dc=com  =>  leaf entry
```

当您更改条目的 naming 属性时，条目 RDN，您执行 *modrdn* 操作。此修改操作会将条目移到目录树中。对于叶条目（无子项的项），modrdn 操作只更改 RDN 部分，父条目保持不变。



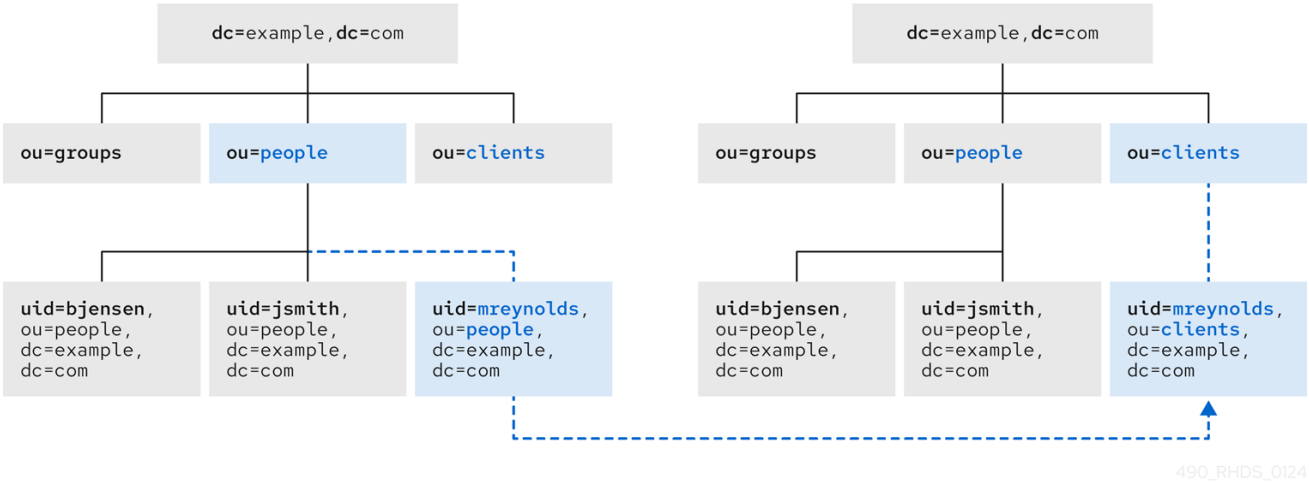
对于子树条目，modrdn 操作重命名子树条目本身，并更改子树下所有子条目的 DN 组件。



重要

子树 modrdn 操作也会移动和重命名子树条目下的所有子条目。对于大型子树，这可能是时间和资源密集型进程。以层次结构来规划对目录数的命名结构，使其不需要频繁对子树进行重命名操作。

重命名子树的类似操作是将条目从一个子树移到另一个子树。这种扩展类型的 modrdn 操作同时重命名条目，即使它相同名称，并且设置一个新的s uperior 属性，将条目从一个父项移到另一个父项。



目录服务器使用 entryrdn.db 索引执行新的高级和子树重命名操作。目录服务器通过自链接、父目录链接和任何子项链接来识别每个条目。entryrdn.db 索引将父项和子项作为条目的属性呈现给条目，并通过唯一 ID 及其 RDN 而不是完整的 DN 描述每个条目。

entryrdn.db 索引的格式如下：

```
numeric_id:RDN => self link
    ID: ; RDN: "rdn"; NRDN: normalized_rdn P:RDN => parent link
    ID: ; RDN: "rdn"; NRDN: normalized_rdn C:RDN => child link
    ID: #; RDN: "rdn"; NRDN: normalized_rdn
```

例如，ou=people 子树具有 dc=example,dc=com 父项和 uid=jsmith 子条目。entryrdn.db 索引包含以下内容：

```
4:ou=people
  ID: 4; RDN: "ou=people"; NRDN: "ou=people"
P4:ou=people
  ID: 1; RDN: "dc=example,dc=com"; NRDN: "dc=example,dc=com"
C4:ou=people
  ID: 10; RDN: "uid=jsmith"; NRDN: "uid=jsmith"
```

执行 rename 操作时请考虑以下几点：

- **您无法重命名根后缀。**
- **您不需要重新配置复制协议。目录服务器将复制协议应用到整个数据库，而不是数据库中的子树。**
- **您可能需要在子树重命名操作后重新配置所有同步协议。同步协议在后缀或子树级别上设置，因此重命名子树可能会破坏同步。**

- 您需要重新配置为子树设置的所有子树级 ACI，并为子树的子条目设置所有条目级 ACI。
- 您可以使用子项重命名子树，但您无法使用子项删除子树。
- 当您尝试更改子树的组件时，如从 ou 移到 dc 时，它可能会失败，并显示 schema 冲突。例如，organizationalUnit 对象类需要 ou 属性。如果操作尝试从 organizationalUnit 对象类中删除 ou 属性，则子树操作会失败。

4.3. 分组目录条目

要简化目录管理，可组您创建的条目。目录服务器支持以下对条目方法进行分组的方法：

- 组
- 角色

4.3.1. 关于目录服务器中的组

组是用户的集合。目录服务器有几个组类型，它们反映了允许的成员资格类型，如证书组、URL 组和唯一组，它们只有唯一的成员。您可以通过对象类（如 groupOfUniqueNames ）和对应的成员属性（如 uniqueMember ）来定义每种类型的组。

组的类型标识成员的类型。组配置取决于目录服务器如何将成员添加到组中。目录服务器有两个组类型：

静态组
: 静态组具有有限并定义的成员列表。您可以手动将成员添加到组条目。

动态组
: 动态组使用过滤器将成员添加到组中。因此，成员数量会持续更改，因为与组过滤器更改的条目数。

组不对条目执行任何操作，但 LDAP 客户端可以管理组来执行操作。

4.3.1.1. 列出用户条目中的组成员资格

组是用户 DN 的列表。默认情况下，只有组条目包含成员资格信息，用户条目不包含此信息。

MemberOf 插件使用组成员条目动态更新用户条目，并反映该用户所属的组。该插件会自动扫描带有指定成员属性的组条目，跟踪所有用户 DN，并使用组名称在用户条目中创建对应的 memberOf 属性。

用户所属每个组的名称列为 memberOf 属性，您可以管理 memberOf 属性的值。

注意

默认情况下，MemberOf 插件只搜索目录服务器存储在与组相同的数据库中的潜在成员。如果目录服务器将用户和组存储在不同的数据库中，则 MemberOf 插件不会更新用户条目，因为插件无法定义用户和组之间的关系。

启用 memberOfAllBackends 属性，将 MemberOf 插件配置为搜索所有配置的数据库。

您可以通过在插件条目中设置多值 memberofgroupattr 来配置单个 MemberOf 插件实例，以管理多种类型的组。

4.3.1.2. 向组中添加自动新条目

您可以根据组成员资格应用密码策略、访问控制列表和其他规则。使用组，您可以在目录中一致且可靠地应用策略。

创建新条目时，自动将新条目分配给组，确保目录服务器在没有管理员操作的情况下立即将适当的策略和功能应用到这些条目。

使用 *Automembership* 插件时，静态组可以充当动态组。Automembership 插件使用基于条目属性、目录位置和正则表达式的一组规则，将用户自动分配给指定的组。

根据其他属性的值，可能存在与 LDAP 搜索过滤器匹配的实例条目。例如，您需要根据 IP 地址或物理位置将机器添加到不同的组中。或者您需要根据用户的员工 ID 号将用户放置到不同的组中。

automember 定义是一组嵌套条目，以及 Auto Membership 插件容器，然后自动成员定义，然后是该定义的任何正则表达式条件。

注意

仅当新条目添加到目录服务器时，目录服务器才会自动将条目分配给组。对于您修改的现有条目或条目以满足自动成员规则，请运行 fix-up 任务来分配正确的组成员资格。

4.3.2. 关于目录服务器中的角色

角色同时充当静态和动态组。对于组，Directory 服务器会将条目添加到组条目中作为成员。使用角色时，Directory 服务器会向条目添加 role 属性，然后使用该属性自动识别角色条目中的成员。

使用角色，您可以使用以下方法组织用户：

- 明确列出角色成员。查看角色时，您可以看到此角色的完整成员列表。您可以查询角色来检查动态组无法实现的成员资格。

- 查看条目所属的角色。当您查看条目时，您可以看到条目所属的角色，因为 Directory 服务器根据条目中的属性决定角色成员资格。它与组的 memberOf 属性类似，唯一的区别是您不需要启用或配置插件实例才能使此功能正常工作。

- 分配适当的角色。目录服务器通过条目分配角色成员资格，而不是通过角色分配。因此，您可以在一个步骤中通过编辑条目来轻松地分配和删除用户所属的角色。

受管角色可以执行您可以使用静态组完成的所有操作。您可以使用过滤的角色过滤角色成员，类似于使用动态组的过滤。角色比组更容易使用，因为它们在实施过程中具有更大的灵活性，并降低客户端复杂性。

您可以使用角色类型显式或动态指定成员。目录服务器支持以下类型的角色：

受管角色

受管角色具有明确的成员列表。

过滤的角色

如果条目在角色中定义的特定属性，目录服务器会将条目分配给过滤的角色。角色定义指定目标属性的 LDAP 过滤器。与过滤器匹配的条目具有角色（位于成员）。

嵌套角色

嵌套角色是包含其他角色的角色。

您可以激活或停用只有一个操作中的整个条目组。您可以通过取消激活角色所属的角色来临时禁用角色的成员。

当您激活某个角色时，用户仍可以使用角色条目绑定到服务器。但是，用户无法使用属于此角色的任何条目绑定到服务器。属于 inactivated 角色的条目会将 nsAccountLock 属性设置为 true。

当您激活嵌套角色时，如果用户是嵌套角色内任何角色的成员，则无法绑定到服务器。属于嵌套角色的直接或间接成员的所有条目会将 nsAccountLock 设置为 true。在嵌套的任意点上激活嵌套角色，取消激活其下的所有角色和用户。

4.3.3. 决定组和角色之间的

角色和组可以实现相同的目标。受管角色可以执行静态组可以执行的操作，而过滤的角色则可过滤和识别成员，方式与动态组相同。角色和组都有优点和缺点。决定是否使用角色或组，还是混合取决于您的要求和服务器资源。

角色降低了客户端复杂性。使用角色时，客户端应用程序可以通过搜索条目中的 nsRole 操作属性来检查角色成员资格。此多值属性标识条目所属的每个角色。从客户端应用程序视图中，检查成员资格的方法统一并在服务器端执行。

但是，角色需要增加服务器复杂性。与评估组相比，评估角色对目录服务器的资源密集型，因为服务器能够为客户端应用工作。

组需要更智能且更复杂的客户端来有效地使用它们。例如，从应用程序角度而言，动态组不提供服务器中的支持，以提供组成员列表。相反，应用会检索组定义，然后运行过滤器。只有在配置适当的插件时，用户条目才会包含组成员资格信息。

注意

您可以使用 MemberOf 插件来平衡组成员资格。当用户添加到组时，MemberOf 插件会在用户条目中动态创建 memberOf 属性。客户端可以在组条目上运行单个搜索，以获取其所有成员的列表，或者对用户条目进行单个搜索来获取它所属的所有组的完整列表。

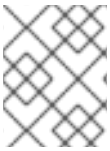
只有修改成员资格时，服务器才会有维护开销。由于目录服务器同时将指定 的成员 （组）和 memberOf （用户）属性存储在数据库中，所以搜索不需要额外的处理，这会使从客户端进行搜索非常高效。

其他资源

- 列出用户条目中的组成员资格
- 向组中添加自动新条目

4.4. 虚拟目录信息树视图

目录服务器支持 *虚拟目录信息树视图*、虚拟视图。虚拟视图是标准目录树之外的可选结构层，用于分类和搜索条目。



注意

虚拟视图与多个后端并不完全兼容。虚拟视图返回的条目必须位于同一后端中，因为搜索仅限于一个后端。

有关虚拟 DIT 视图的更多信息，请参阅使用视图创建虚拟目录层次结构

4.4.1. 虚拟 DIT 视图示例

下面的 LDIF 条目显示基于位置的虚拟视图层次结构。驻留在 dc=example,dc=com 下的任何条目，并适合在此视图中显示查看描述，并根据位置进行组织。

```
dn: ou=Location Views,dc=example,dc=com
objectclass: top
objectclass: organizationalUnit
objectclass: nsView
ou: Location Views
description: views categorized by location


dn: ou=Sunnyvale,ou=Location Views,dc=example,dc=com
objectclass: top
objectclass: organizationalUnit
objectclass: nsView
ou: Sunnyvale
nsViewFilter: (l=Sunnyvale)
description: views categorized by location


dn: ou=Santa Clara,ou=Location Views,dc=example,dc=com
objectclass: top
objectclass: organizationalUnit
objectclass: nsView
```

```
ou: Santa Clara
nsViewFilter: (l=Santa Clara)
description: views categorized by location


dn: ou=Cupertino,ou=Location Views,dc=example,dc=com
objectclass: top
objectclass: organizationalUnit
objectclass: nsView
ou: Cupertino
nsViewFilter: (l=Cupertino)
description: views categorized by location
```

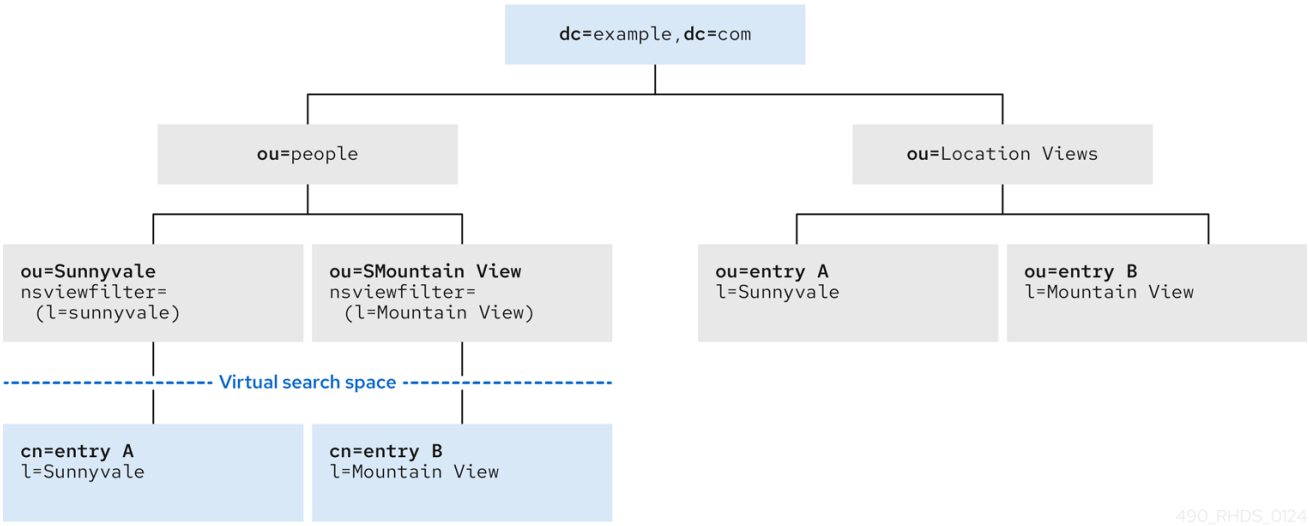
基于 ou=Location Views,dc=example,dc=com 的子树搜索返回 dc=example,dc=com 下的所有条目，它与过滤器 (l=Sunnyvale)、(l=Santa Clara) 或 (l=Cupertino) 匹配。但是，一个级别的搜索不会返回除子视图条目以外的条目，因为所有合格条目都位于三个下级视图中。

ou=Location Views,dc=example,dc=com view 条目本身不包含过滤器。此功能有助于组织分层，无需进一步限制视图中包含的条目。任何视图都可以省略该过滤器。

虽然示例过滤器非常简单，但您使用的过滤器可以根据需要复杂。您可以限制视图应包含的条目类型。例如，若要将此层次结构限制为仅包含人员条目，请将 nsfilter 属性添加到 ou=Location Views,dc=example,dc=com，其过滤器值 (objectclass=organizationalperson)。

每个带有过滤器的视图会限制所有下级视图的内容，而带有过滤器的下级视图也会限制其上级内容。例如，首先创建顶部视图 ou=Location Views 和上述新过滤器，会创建一个包含 机构 对象类的所有条目的视图。添加下级视图以进一步限制条目时，现在下级视图中显示的条目将从上级视图中删除。这演示了虚拟 DIT 视图如何模拟传统 DIT 的行为。

虽然虚拟 DIT 视图模拟传统 DIT 的行为，但视图可以执行传统 DIT 无法执行的任务：条目可能会出现在多个位置上。例如，要将 Entry B 与 Mountain View 和 Sunnyvale 关联，请将 Sunnyvale 值添加到 location 属性中，该条目会出现在这两个视图中。



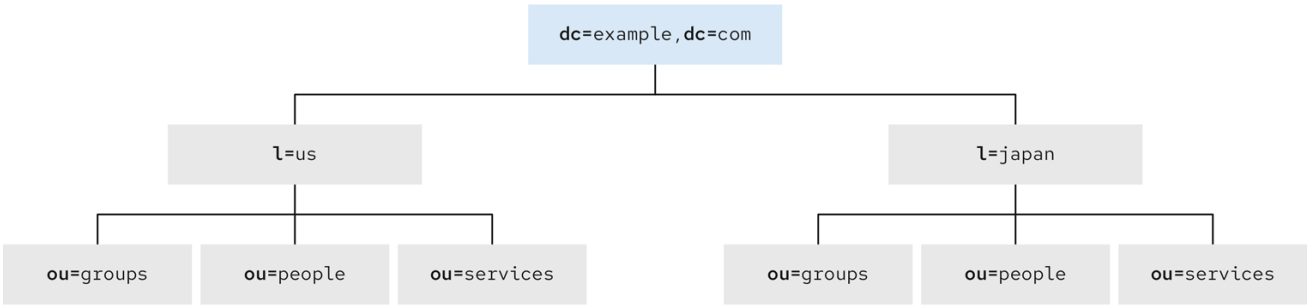
4.5. 目录树设计示例

查找国际企业和 ISP 的目录树示例。

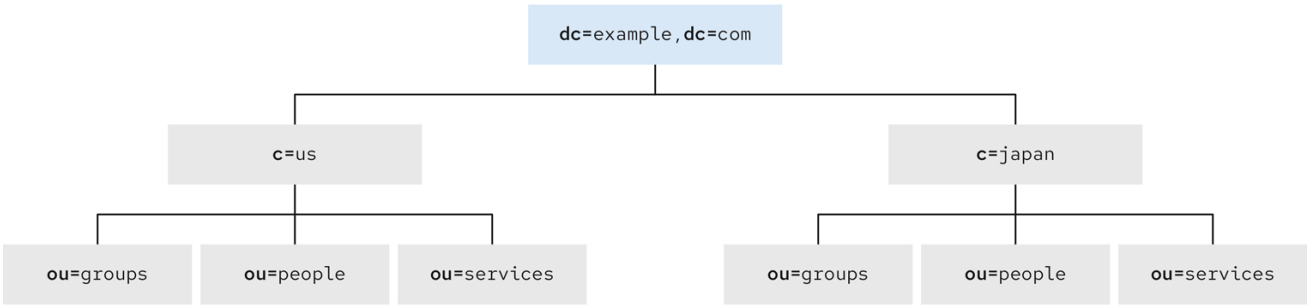
国际企业目录树

使用 Internet 域名作为目录树的根条目。然后，为企业有操作的每个国家的国家/地区分支根条目下面的树。

要代表不同的国家，请使用 l（位置）属性：



但是，c (country)属性也可以代表每个国家分支：



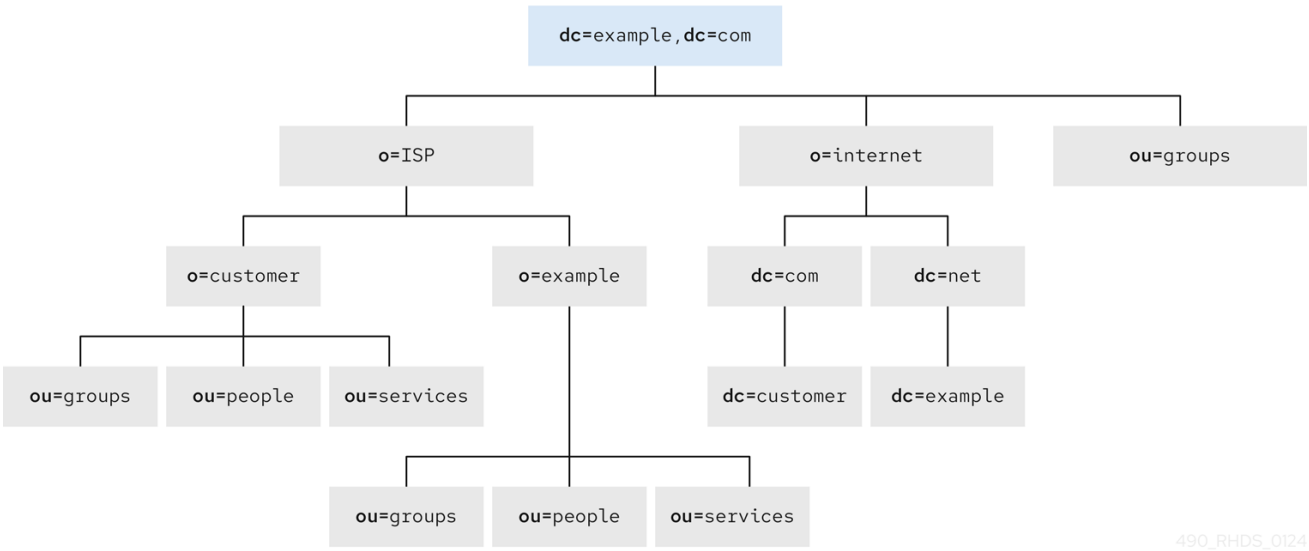
LDAP 对 DN 中的属性顺序没有限制。

ISP 的目录树

Internet 服务提供商(ISP)可以支持多种企业及其目录。ISP 应该把每个客户视为一个唯一的企业，并相应地设计其目录树。为安全起见，为每个客户提供唯一的后缀和独立的安全策略。

为每个客户分配一个单独的数据库，并将这些数据库存储在单独的服务器上。将每个目录树放在其自己的数据库中时，您可以备份和恢复每个目录树的数据，而不影响其他客户。

此外，分区减少了磁盘争用导致的性能问题，以及磁盘中断的客户数量可能影响。



4.6. 其他资源

- RFC 2247：在 LDAP/X.500 区分名称中使用域
- RFC 2253: LDAPv3，UTF-8 字符串代表可辨识的名称

第 5 章 设计目录拓扑

红帽目录服务器可以存储大量条目，因此您可能需要在多个服务器间分发您的条目。目录拓扑描述了如何在多个物理目录服务器中划分目录树以及如何链接这些服务器。

5.1. 拓扑概述

目录服务器支持 *分布式目录*，将您在 Designing-the-directory-tree 中设计的目录树分散到多个物理目录服务器中。如何在这些服务器间划分目录会影响以下与性能相关的点：

- 支持目录的应用程序的性能。
- 目录服务的可用性。
- 管理目录服务。

目录拓扑有以下键含义：

数据库

数据库 是作业的基本单元，如复制、备份和数据恢复。您可以将单个目录划分为多个部分，并将其分配到单独的数据库。然后您可以在服务器之间分发这些数据库，从而减少每台服务器的工作负载。您可以在单一服务器上存储多个数据库。例如，一个服务器可能包含三个不同的数据库。
有关多个数据库的更多详细信息，请参阅关于使用多个数据库。

后缀

当您将目录树划分到多个数据库时，每个数据库都包含目录树的一个部分，称为 *后缀*。例如，您可以使用一个数据库来仅存储目录树的 ou=people,dc=example,dc=com 后缀(branch)中的条目。
有关后缀的详情，请参阅关于后缀。

知识参考（引用和链）

目录服务器提供知识引用机制，如引用和链等，用于连接存储在不同数据库中的目录数据。
有关引用和串联的详情，请参阅使用引用 和使用 链。

5.2. 分发目录数据

通过分发目录数据，您可以在多个服务器上扩展目录，而无需在企业的每个服务器上物理包含目录条目。因此，分布式目录可以保存更多的条目数量，而不是单个服务器可能。

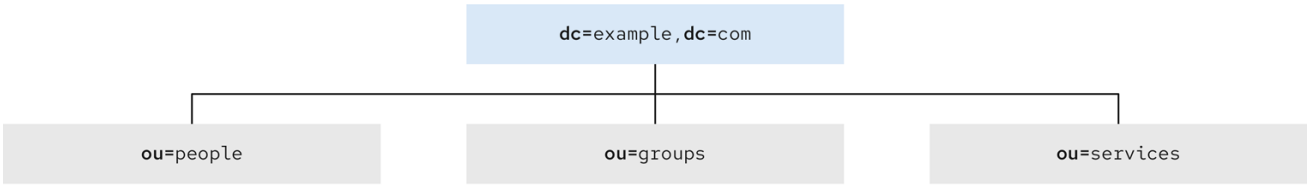
另外，您可以将目录配置为隐藏用户的发布详情。

5.2.1. 在目录服务器中使用多个数据库

目录服务器将数据存储在 Lightning Memory-Mapped Databases (LMDB)中。每个数据库由一组大型文件组成，其中包含分配给它的所有数据。

您可以将目录树的不同部分存储在不同的数据库中。例如，您的目录树可以通过以下方式出现：

图 5.1. 目录树示例



示例中的目录由以下三个子后缀组成：

- ou=people,dc=example,dc=com
- ou=groups,dc=example,dc=com
- ou=services,dc=example,dc=com

您可以通过以下方式将三个子后缀的数据存储在三个独立的数据库中：

图 5.2. 在单独的数据库中存储后缀数据

- DB1 用于 ou=people,dc=example,dc=com
- DB2 for ou=groups,dc=example,dc=com
- DB3 用于 ou=services,dc=example,dc=com

当您将目录树划分为多个数据库时，您可以在多个服务器之间分发这些数据库，以减少每台服务器上的工作负载。例如，您可以在两台服务器上存储三个数据库(DB 1、DB2 和 DB3)。

图 5.3. 在单独的服务器间划分后缀数据库

服务器 A 包含 DB1 和 DB2，而 Server B 包含 DB3。

目录服务器支持动态添加数据库，而不停止整个目录服务。

5.2.2. 目录服务器中的后缀

数据库包含特定后缀的数据（目录树的一个部分）。在 Directory Server 中，您可以创建 root 后缀或子后缀。

根后缀

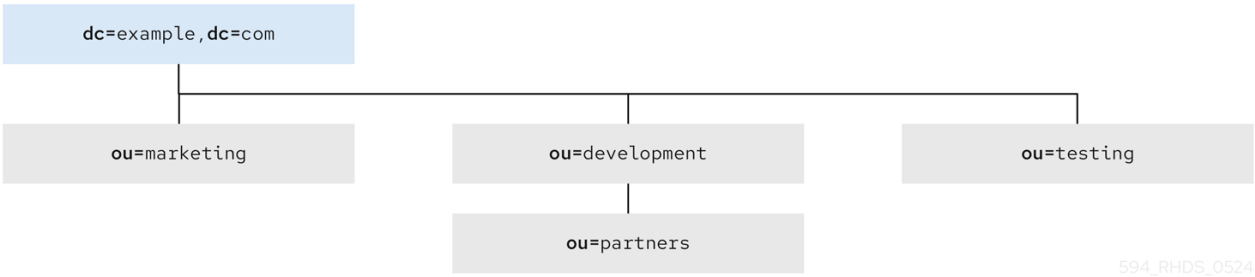
root 后缀是树顶部的条目。它可以是目录树的根目录，也可以是您为目录服务器设计的大型树的一部分。

sub-suffix

子后缀是根后缀下的分支。

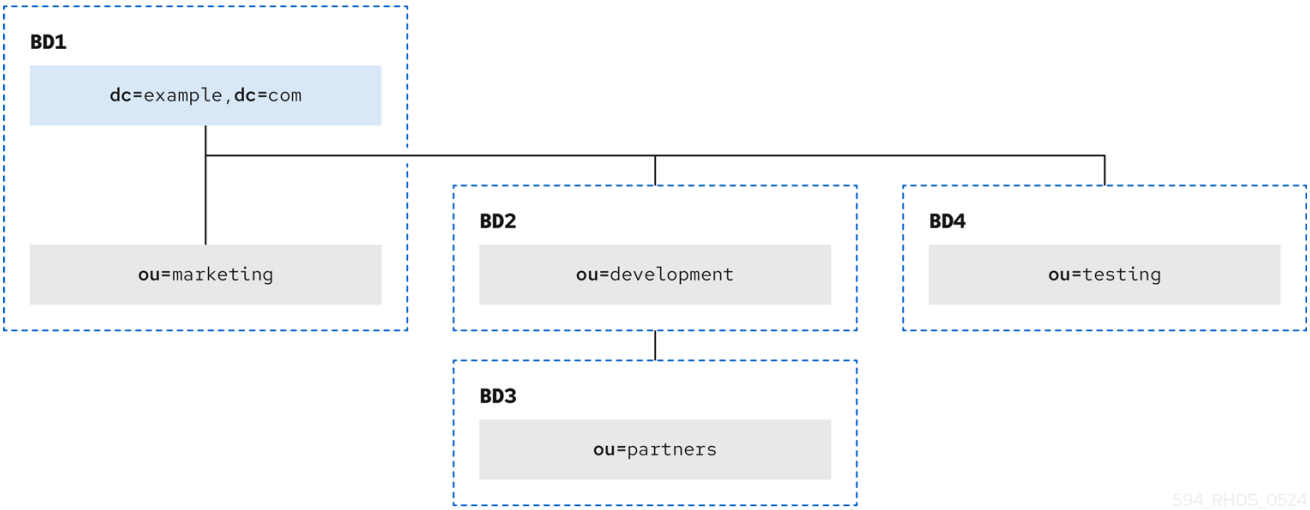
例如，ExampleCom 创建后缀来代表其目录数据的分布，如下所示：

图 5.4. ExampleCom 的目录树



ExampleCom 以以下方式将其目录树分散到四个不同的数据库中：

图 5.5. 分布在多个数据库中的目录树



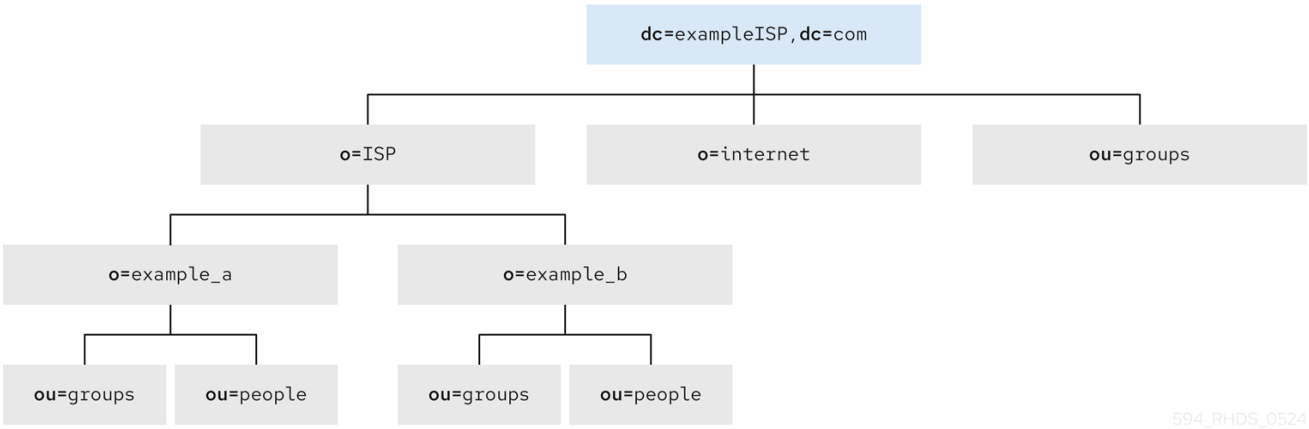
四个数据库包含以下后缀的数据：

- root 后缀 dc=example,dc=com.除了 dc=example,dc=com 数据外，此数据库还包含原始目录树的 ou= marketing,dc=example,dc=com 分支的数据。
- ou=testing,dc=example,dc=com 子后缀。
- ou=development,dc=example,dc=com 子后缀。
- ou= partner,ou=development,dc=example,dc=com 子后缀。

使用多个 root 后缀

目录服务可以包含多个根后缀。例如，一个 ISP 称为主机多个网站，一个用于 example_a.com，另一个用于 example_b.com。ExampleISP 具有以下目录结构：

图 5.6. 具有多个根后缀的目录树



ISP 创建以下根后缀：

- dc=exampleISP,dc=com，其中包含以下条目的数据：
  - dc=exampleISP,dc=com

- o=ISP,dc=exampleISP,dc=com
- o=internet,dc=exampleISP,dc=com
- ou=groups,dc=exampleISP,dc=com

- 带有以下条目的数据的 o=example_a.com ：
  - o=example_a.com,o=ISP,dc=exampleISP,dc=com
  - ou=people,o=example_a.com,o=ISP,dc=exampleISP,dc=com
  - ou=groups,o=example_a.com,o=ISP,dc=exampleISP,dc=com
- 带有以下条目的数据的 o=example_b.com ：
  - o=example_b.com,o=ISP,dc=exampleISP,dc=com
  - ou=people,o=example_b.com,o=ISP,dc=exampleISP,dc=com
  - ou=groups,o=example_b.com,o=ISP,dc=exampleISP,dc=com

其他资源

- 在单独的数据库中存储后缀

5.3. 目录服务器中的知识参考

知识参考定义了分布式数据之间的关系。知识引用是指向不同数据库中保存的目录信息的指针。目录服务器提供以下知识引用，用于将分布式数据链接到单个目录树：

引用

目录服务器向客户端应用程序返回信息，表示客户端应用程序需要联系另一个服务器来满足请求。

链

目录服务器代表客户端应用程序联系其他服务器，并在操作完成后将结果返回给客户端应用程序。

5.4. 在目录服务器中使用引用

Directory Server 返回的信息是目录服务器返回的信息，告知客户端应用程序要联系以继续请求。*当客户端应用程序请求本地服务器不包含的目录条目时，会发生此重定向机制。*

目录服务器支持以下引用类型：

默认引用

当客户端应用程序请求不属于本地树的条目时，该目录会返回默认的引用。您可以在服务器和后缀级别上配置默认引用。

智能引用

目录服务器在目录中存储智能引用条目。智能引用指向包含子树的服务器，其 DN 与包含智能引用的条目的 DN 匹配。

目录服务器返回 LDAP 统一资源 locator 或 LDAP URL 格式的所有引用。

其他资源

- *目录服务器中的默认引用*
- *目录服务器中的智能引用*

5.4.1. LDAP 引用的结构

目录服务器返回 LDAP URL 格式的所有引用。LDAP URL 包含以下信息：

- 要联系的服务器的主机名。
- 服务器配置为侦听 LDAP 请求的服务器上的端口号。
- 基本 DN（用于搜索操作）或目标 DN（用于添加、删除和修改操作）。

例如，客户端应用程序通过 dc=example,dc=com 分支搜索带有 surname Jensen 的条目。但是，目录树的一部分存储在欧洲服务器上。引用会将以下 LDAP URL 返回到客户端应用程序：

> ldap://europe.example.com:389/ou=people,l=europe,dc=example,dc=com

本引用指示客户端应用程序在端口 389 上联系主机 europe.example.com，并通过欧洲分支 ou=people,l=europe,dc=example,dc=com 提交一个新的搜索。

您使用的 LDAP 客户端应用程序决定了如何处理引用。有些客户端应用程序会自动重试服务器中的操作。其他客户端应用程序将引用信息返回给用户。红帽目录服务器提供的大多数 LDAP 客户端应用程序（如命令行工具）自动遵循引用。目录服务器使用初始目录请求中提供的同一绑定凭证来访问服务器。

大多数客户端应用程序都遵循有限的引用数，或 跃点。对引用数量的限制可减少客户端应用程序试图完成目录查找请求的时间，并有助于消除由循环引用模式导致的挂起进程。

5.4.2. 目录服务器中的默认引用

当联系的服务器或数据库不包含请求的数据时，目录服务器会返回 默认的引用。

例如，客户端请求以下目录条目： uid=bjensen,ou=people,dc=example,dc=com。

但是，服务器仅管理存储在 dc=europe,dc=example,dc=com 后缀下的条目。目录向客户端返回一个引用，其中包含要联系 dc=example,dc=com 后缀下的条目的信息。然后，客户端联系适当的服务器并重新提交原始请求。

您可以在服务器和后缀级别配置默认引用：

-

要设置服务器级别引用，请使用服务器级配置属性 nsslapd-referral。目录服务器在 dse.ldif 配置文件中存储属性值。当服务器不可用或者客户端没有访问本地服务器上的数据时，Directory 服务器会返回默认的引用。

- 要设置后缀级别引用，请使用后缀配置属性 nsslapd-referral 和 nsslapd-state。当整个后缀离线时，Directory 服务器会将引用返回到对该后缀发出的客户端请求。

5.4.3. 目录服务器中的智能引用

除了默认的引用外，Directory 服务器 还支持智能引用，它将目录条目或目录树与特定 LDAP URL 关联。因此，目录服务器可以将客户端请求转发到以下任意一个：

- 包括在不同服务器上的同一命名空间。
- 不同服务器上的不同命名空间。
- 同一服务器上的不同命名空间。

与默认引用不同，目录服务器将智能引用存储在目录中，而不是在配置文件中。

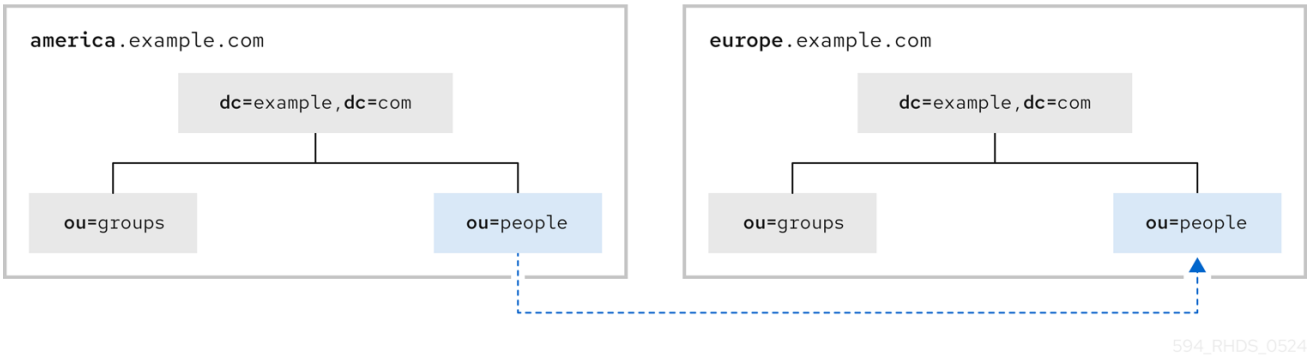
例如，ExampleCom 的美国办公室的目录包含 ou=people,dc=example,dc=com 目录分支点。

要将此分支上的请求重定向到 ExampleCom 欧洲办事处的 ou=people 分支，您可以在 ou=people 条目上指定一个智能引用。智能引用具有以下值：

ldap://europe.example.com:389/ou=people,dc=example,dc=com

美国目录的 ou=people 分支的请求将按照以下方式重定向到欧洲目录中：

图 5.7. 使用智能引用重定向请求

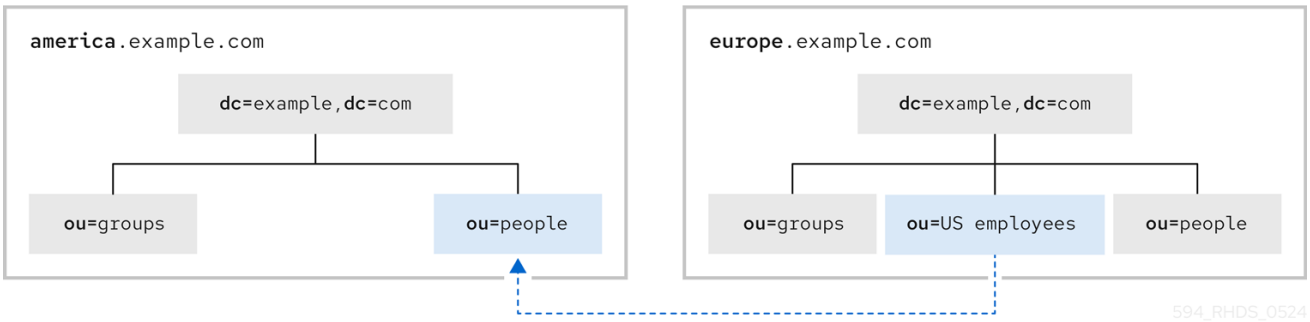


您可以使用相同的机制将查询重定向到使用不同的命名空间的不同服务器。例如，在 ExampleCom 办事处工作的员工为美国的 ExampleCom 员工的电话号码提出了欧洲目录服务的请求。目录服务器返回以下引用：

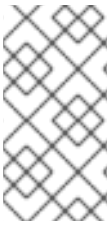
```
ldap://america.example.com:389/ou=people,dc=example,dc=com
```

下图显示了引用不同命名空间的工作方式：

图 5.8. 将查询重定向到不同的服务器和命名空间



最后，在同一服务器上提供多个后缀时，您可以将查询从一个命名空间重定向到同一服务器上提供的另一项。例如，要将本地服务器上的 o=example,c=us 的所有查询重定向到 dc=example,dc=com，请在 o=example,c=us 条目上设置 smart referral ldap:///dc=example,dc=com。LDAP URL 中的第三个斜杠表示 URL 指向同一服务器。



注意

从一个命名空间到另一个命名空间的引用仅适用于其搜索以可分辨名称的客户端。其他类型的操作（如 ou=people,o=example,c=US ）没有正确执行。

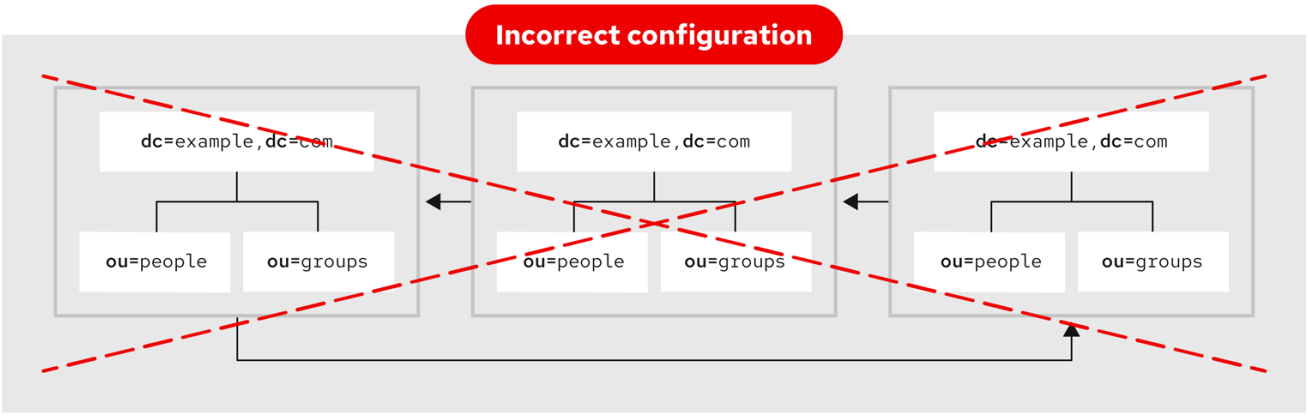
5.4.4. 使用智能引用时的注意事项

在使用智能引用前请考虑以下点：

- 保持设计简单。

  复杂的引用网络使管理变得困难。智能引用也可以导致循环引用模式。例如，一个指向一个 LDAP URL 的引用指向另一个 LDAP URL，以此类推，直到链中的某个位置指向原始服务器。下图显示了循环引用模式：

图 5.9. 圆形引用模式



- 重定向至主要分支点。

  为提高安全性并降低维护成本，请限制引用使用在后缀和主要分支点级别处理重定向。不要使用 smart 引用作为别名机制。

- 考虑安全隐患。

  访问控制不会跨引用边界。即使发送请求的服务器最初允许访问条目，智能引用也会向其他服务器发送客户端请求，客户端应用可能会被拒绝访问。

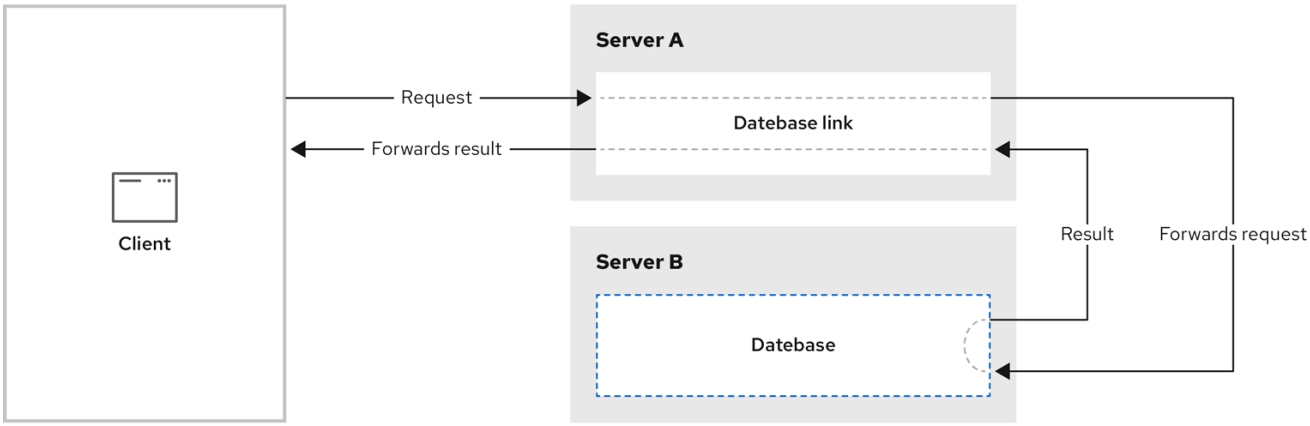
  此外，客户端应用程序需要凭据来向引用客户端的服务器进行身份验证。

5.5. 使用链

链是一种代表客户端应用程序将请求重定向到其他服务器的方法。链(Chaining)作为插件在服务器中实

施。插件被默认启用。使用这个插件，您可以创建数据库链接，指向远程存储的数据的特殊条目。当客户端应用程序从数据库链接请求数据时，数据库链接会从远程数据库检索数据并将其返回到客户端。

图 5.10. 使用链将客户端请求发送到服务器



每个数据库链接都与保留数据的远程服务器关联。您还可以配置包含数据库链接副本的备用远程服务器，以便在出现故障时使用。

有关配置数据库链接的更多信息，请参阅 创建和维护数据库链接。

数据库链接提供以下功能：

- 对远程数据的不可见访问

  数据库链接会解析客户端请求，从客户端完全隐藏数据分发。

- 动态管理

  您可以在整个系统一直供客户端应用程序使用时，在系统中添加或删除目录的一部分。您可以使用数据库链接来临时将引用返回给应用程序，直到您在目录中重新分发条目。

  您还可以使用返回引用的后缀而不是将客户端应用程序转发到数据库来实现。

- Access control

数据库链接模拟客户端应用程序，为远程服务器提供适当的授权身份。当不需要访问控制评估时，您可以禁用远程服务器上的用户模仿。

有关数据库链接和访问控制评估的更多信息，请参阅 数据库链接和访问控制评估。

5.6. 决定引用和链

根据您的目录的具体需求，在引用和串联之间进行选择。

- 链会降低客户端复杂性，从而降低服务器复杂性。但是，通过串联，客户端应用程序可以与单一服务器交互，并仍然访问存储在多个服务器上的数据。客户端应用不需要对请求链的服务器进行身份验证。

- 使用引用时，客户端应用必须找到引用并重新提交搜索结果。客户端还必须能够对服务器进行身份验证。

  另外，当公司网络使用代理时，有时引用会失败。例如，客户端应用程序可能只有与防火墙中的一个服务器通信的权限。如果该应用程序被称为其他服务器，则应用程序可能无法成功联系它。

  但是，参考人员为客户端应用程序提供了更大的灵活性，开发人员可以为分布式目录操作进度向用户提供更好的反馈。

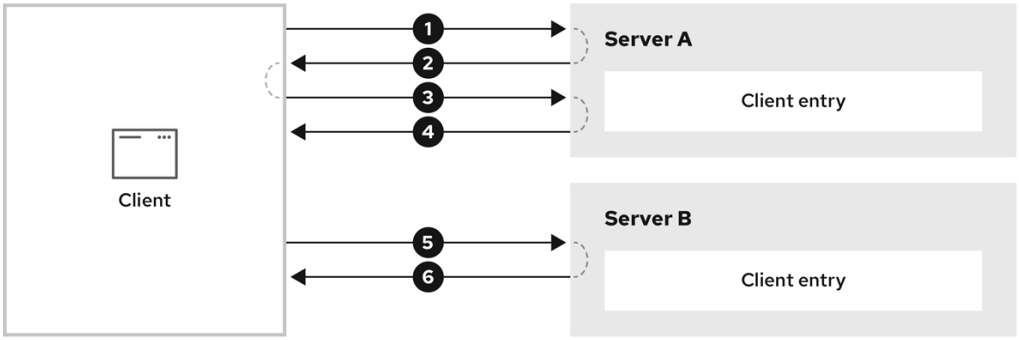
5.6.1. 评估访问控制

链评估访问控制与引用的不同。使用引用时，所有目标服务器上必须存在客户端条目(bind DN)。使用串联时，客户端条目不需要位于所有目标服务器上。

5.6.1.1. 使用引用执行搜索请求

下图显示了使用引用对服务器的客户端请求：

图 5.11. 使用引用向服务器发送客户端请求



594_RHDS_0524

搜索请求以以下方式进行：

1. *客户端应用首先与 Server A 绑定。*

2. *服务器 A 包含提供用户名和密码的客户端的条目，因此它返回绑定接受消息。为了让引用可以正常工作，客户端条目必须存在于 服务器 A 中。*

3. *客户端应用程序将操作请求发送到 服务器 A。*

4. *但是，服务器 A 不包含请求的信息。相反，Server A 会向客户端应用程序返回一个引用，指示它联系 Server B。*

5. *然后，客户端应用程序向 Server B 发送绑定请求。要成功绑定，Server B 还必须包含客户端应用程序的条目。*

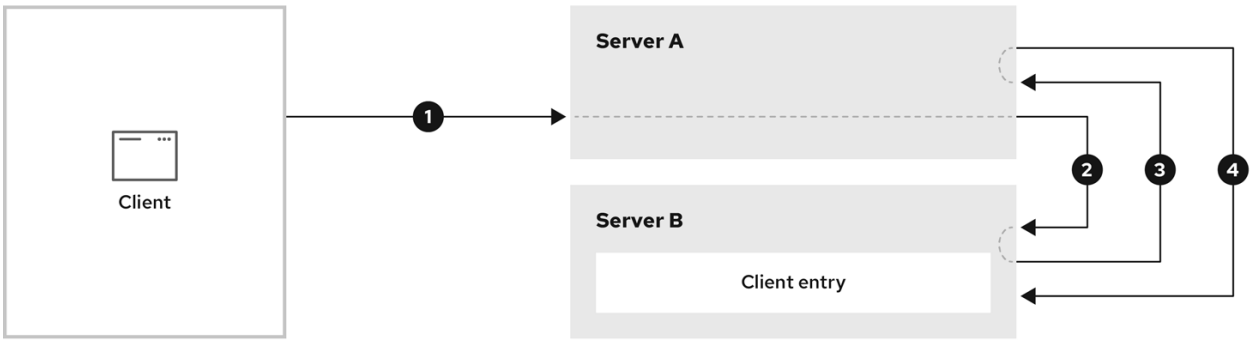
6. *绑定成功，客户端应用现在可以将其搜索操作重新提交到 Server B。*

此方法要求 Server B 具有来自 Server A 的客户端条目的复制副本。

5.6.1.2. 使用链执行搜索请求

您可以通过链解决跨服务器复制客户端条目的问题。

图 5.12. 使用链将客户端请求发送到服务器



搜索请求以以下方式进行：

1. *客户端应用与服务器 A 绑定，服务器 A 会尝试确认用户名和密码是否正确。*

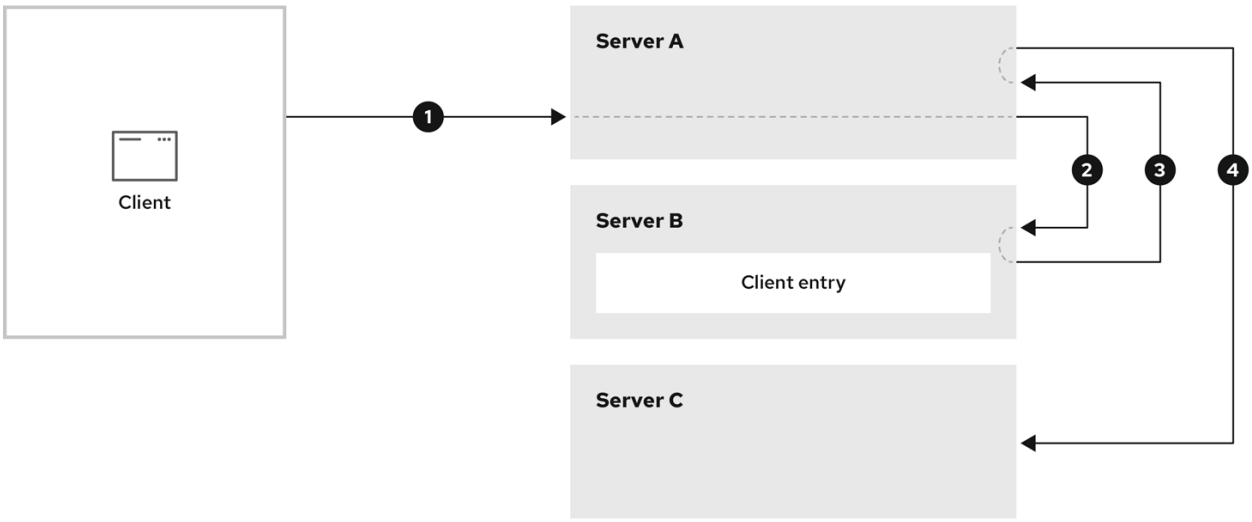
2. *服务器 A 不包含与客户端应用程序对应的条目。相反，它包含指向 Server B 的数据库链接，其中包含客户端的实际条目。服务器 A 将绑定请求 发送到服务器 B。*

3. *服务器 B 将接受响应 发送到服务器 A。*

4. *然后，服务器 A 使用数据库链接处理客户端应用程序请求。数据库链接联系位于 Server B 上的远程数据存储来处理搜索操作。*

在链系统中，与客户端应用程序对应的条目不需要位于与客户端请求的数据相同的服务器上。下图显示了如何使用两个串联服务器来完成客户端搜索请求。

图 5.13. 使用两个不同的服务器进行身份验证并检索数据



搜索请求以以下方式进行：

1. *客户端应用与服务器 A 绑定，服务器 A 确认用户名和密码正确。*

2. *服务器 A 不包含与客户端应用程序对应的条目。相反，它包含指向 Server B 的数据库链接，其中包含客户端的实际条目。服务器 A 将绑定请求 发送到服务器 B。*

3. *服务器 B 将接受响应 发送到服务器 A。*

4. *然后，服务器 A 使用另一个数据库链接处理客户端请求。数据库链接联系位于 Server C 上的远程数据存储来处理搜索操作。*

5.6.1.3. 不支持的访问控制

数据库链接不支持以下访问控制：

- *控制当用户条目位于其他服务器上时必须访问用户条目的内容。这包括基于组、过滤器和角色的访问控制。*

- *根据客户端 IP 地址或 DNS 域的控制可能会被拒绝。这是因为数据库链接在联系远程服务器时模拟客户端。如果远程数据库包含基于 IP 的访问控制，它会使用数据库链接域而不是原始客户*

端域来评估它们。

5.7. 使用索引提高数据库性能

根据数据库的大小，客户端应用程序执行的搜索可能需要大量时间和资源。因此，为了提高搜索性能，您可以使用索引。

索引是目录数据库存储的文件。为您的目录中的每个数据库维护单独的索引文件。每个文件根据其索引的属性命名。特定属性的索引文件可以包含多个类型的索引。例如，名为 cn.db 的文件包含通用名称(cn)属性的所有索引。

根据使用目录的应用程序类型，使用不同类型的索引。不同的应用程序可能会频繁搜索特定的属性，或者可能会以不同语言搜索目录，或者可能需要特定格式的数据。

5.7.1. 目录索引类型概述

目录服务器支持以下索引类型：

存在索引

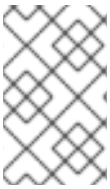
列出具有特定属性的条目，如 uid。

相等索引

列出包含特定属性值的条目，如 cn=Babs Jensen。

大约索引

允许搜索大约（或类似"sounds"）搜索。例如，条目可能包含 cn=Babs L. Jensen 的属性值。大约搜索将返回针对 cn~=Babs Jensen、cn~=Babs 和 cn~=Jensen 进行搜索的值。



注意

大约索引需要使用 ASCII 字符以英语编写。

子字符串索引

允许针对条目内的子字符串进行搜索。例如，您搜索 cn swigderson 将匹配常见名称，如 Bill Anderson、Norma Henderson 和 contains this string。

国际索引

提高了在国际目录中搜索信息的性能。您可以通过将区域设置（异步 OID）与您要索引的属性关联，将索引配置为应用匹配的规则。

浏览索引或虚拟列表视图(VLV)索引

提高了 web 控制台中条目的显示性能。您可以在任何目录树分支上创建 浏览索引，以提高显示性能。

其他资源

- *管理索引*

5.7.2. 评估索引成本

在使用索引来提高搜索性能时请考虑以下点：

- *索引增加修改条目所需的时间。*

  维护的更多索引，需要更长的时间来更新数据库。

- *索引文件使用磁盘空间。*

  您可以索引的更多属性，即您创建的更多文件。另外，如果您为包含长字符串的属性创建大约和子字符串的索引，索引文件可能会快速增长。

- *索引文件使用内存。*

  为了更有效地运行，目录服务器会将尽可能多的索引文件放入内存中。根据数据库缓存大小，索引文件使用池中可用的内存。大量索引文件需要更大的数据库缓存。

- *创建索引文件需要一些时间。*

  虽然索引文件在搜索过程中节省时间，但维护不必要的索引可能会浪费时间。在使用目录时，仅维护客户端应用程序所需的文件。

第 6 章 设计复制过程

复制目录信息会增加目录的可用性和性能。设计复制过程，以确保数据在何时和需要的位置可用。

6.1. 复制简介

复制是自动将目录数据从一个目录服务器复制到另一个目录服务器的机制。使用复制时，存储在自己的数据库(副本)中的任何目录树或子树都可以在服务器之间复制。保存信息的主副本的服务器会自动将任何更新复制到所有副本。

复制提供高可用性目录服务，并可在地理上分发数据。以下是复制优点列表：

- 容错和故障转移

  将目录树复制到多个服务器可确保您的目录可用，即使客户端应用程序因为硬件、软件或网络问题无法访问特定的目录服务器。客户端被称为另一个目录服务器进行读写操作。

  注意

  只有通过 多层次复制复制功能，才能添加、修改和删除 操作的故障转移。

- 负载平衡

  在服务器间复制目录树可减少任何给定服务器上的访问负载，从而改进了服务器响应时间。

- 更高的性能

  将目录条目复制到用户接近的位置可提高目录服务器性能。

- 本地数据管理

  通过复制，您可以在本地拥有和管理信息，同时与整个企业中的其他目录服务器共享。

6.1.1. 复制概念

在考虑实施复制时，回答以下基本问题：

- 您需要复制哪些信息？
- 哪些服务器持有该信息的主副本或供应商副本？
- 哪些服务器持有该信息的只读副本或消费者副本。
- 当消费者副本从客户端应用程序接收 修改 请求时会发生什么？哪些服务器必须重定向到哪个服务器？

了解提供了解目录服务器如何实现复制的概念：

- replica
- 复制单元
- 供应商和消费者
- 变更日志
- 复制协议

6.1.1.1. replica

副本 是参与复制的数据库。目录服务器支持以下类型的副本：

供应商副本（读写）

包含目录数据的主副本的读写数据库。只有供应商副本处理 修改 目录客户端的请求。

消费者副本（只读）

包含供应商副本上保存信息的另一副本的只读数据库。消费者副本可以处理目录客户端的搜索请求，但引用 修改 对供应商副本的请求。

目录服务器可以在复制中管理具有不同角色的多个数据库。例如，您可以在供应商副本中存储 dc=accounting,dc=example,dc=com 后缀，以及消费者副本中的 dc=sales,dc=example,dc=com 后缀。

6.1.1.2. 复制单元

复制的最小单元是后缀（命名空间）。复制机制要求一个后缀对应于一个数据库。目录服务器无法使用自定义分发逻辑复制通过两个或多个数据库分发的后缀。

6.1.1.3. 供应商和消费者

供应商服务器

供应商服务器是将更新复制到其他服务器的服务器。供应商服务器维护一个更改日志，其中包含每个更新操作的记录。

消费者服务器

消费者服务器是一个从其他服务器接收更新的服务器。

在以下情况下，服务器可以同时扮演供应商和消费者的角色：

- 在级联复制中，当某些服务器扮演 hub 服务器的角色时。如需更多信息，请参阅 删除复制。
- 在多层次复制中，当多个服务器管理供应商读写副本时。每台服务器从其他服务器发送和接收更新。如需更多信息，请参阅 多层次复制。

注意

在 Red Hat Directory Server 中，供应商服务器总是启动复制，而不是消费者。

供应商服务器必须执行以下操作：

- 响应从目录客户端读取请求和更新请求。
- 维护副本的状态信息和更改日志。供应商服务器始终负责记录对其管理的读写副本所做的更改。这样可确保任何更改被复制到消费者服务器中。
- 启动到消费者服务器的复制。

消费者服务器必须执行以下操作：

- 响应读取请求。
- 请参阅对副本的供应商服务器更新请求。当消费者服务器收到添加、删除或更改条目的请求时，请求被称为供应商服务器。然后供应商服务器执行请求并复制这些更改。

在级联复制的特殊情况下，hub 服务器执行以下操作：

- 响应读取请求。
- 请参阅对供应商服务器的更新请求。
- 启动到消费者服务器的复制。

6.1.1.4. 变更日志

每个供应商服务器均维护更改日志。changelog 是供应商副本中发生的修改记录。供应商服务器将这些修改推送到存储在其他服务器上的副本。

添加、修改或删除条目时，目录服务器会在 changelog 文件中记录执行的 LDAP 操作。

更改日志仅用于服务器内部使用。如果您的应用程序需要读取更改日志，则需要使用 Retro Changelog 插件来向后兼容。

有关 changelog 属性的详情，请参考 cn=changelog,cn=database_name,cn=ldbm database,cn=plugins,cn=config 下的数据库属性。

6.1.1.5. 复制协议

服务器使用复制协议来定义如何在两个服务器之间执行复制。复制协议描述了一个供应商 和一个 消费者之间的复制。协议在供应商服务器上配置，并确定以下信息：

- 要复制的数据库。
- 推送数据的使用者服务器。
- 复制可能会发生的时间。
- 供应商服务器必须在消费者上绑定的 DN 和凭证，称为 Replication Manager 条目或 供应商绑定 DN。
- 如何保护连接，例如 TLS、StartTLS、客户端身份验证、SASL 或简单身份验证。
- 要复制的属性。有关部分复制的详情，请参阅 Fractional 复制。

6.1.2. 数据一致性

数据一致性指的是复制数据库的内容在给定时间点上如何相互匹配。供应商决定消费者何时必须更新，并启动复制。复制只能在消费者初始化后启动。

目录服务器始终在一周的一天或一天的特定时间保持同步或调度更新。

持续同步副本

持续同步的副本提供更好的数据一致性，但它们会因为频繁更新而增加网络流量。

在以下情况下使用持续同步的副本：

- 在服务器间有可靠、高速的连接。
- 您的客户端应用程序主要发送 搜索，读取，并与目录服务器 进行比较，仅发送几个 更新操作。

用户计划更新

如果您的目录可能具有较低级别的数据一致性，并且您希望降低网络流量的影响，请选择调度更新。

在以下情况下使用调度的更新：

- 您有不可靠或定期可用的网络连接。
- 客户端应用程序主要向 目录服务器 发送添加和修改操作。
- 您需要降低连接成本。

multi-supplier 复制中的数据一致性

当您有多层次复制时，每个供应商都有松散一致的副本，因为在任何给定时间，供应商在存储的数据中可能会存在差异，即使副本不断同步。

松散一致性的主要原因如下：

- 在供应商之间传播 修改操作 具有延迟。
- 服务 修改操作 的供应商不会等待第二个供应商在向客户端返回"成功"消息前进行验证。

6.2. 常见复制场景

您可以使用以下常见场景来构建最适合您的需要的复制拓扑：

- 单层复制
- multi-supplier 复制
- 级联复制
- 混合环境

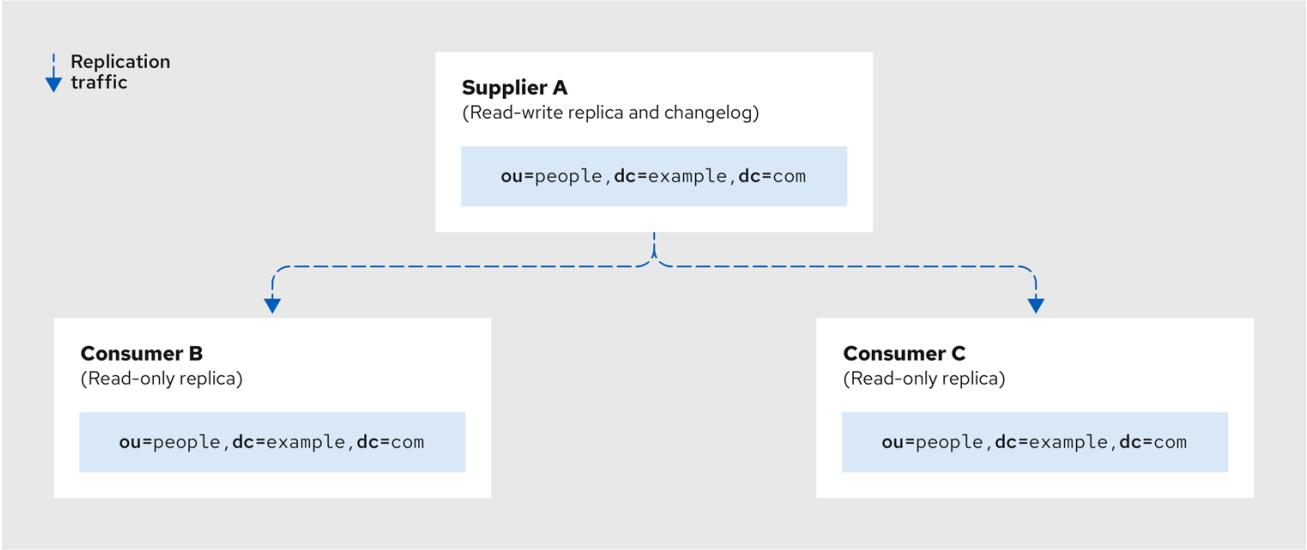
6.2.1. 单层复制

在单层次复制场景中，供应商服务器维护目录数据的主副本（读写副本），并将此数据的更新发送到一个或多个消费者服务器。所有目录修改都会在供应商服务器上的读写副本中进行，消费者服务器包含数据的只读副本。

供应商服务器维护一个 changelog，记录对供应商副本所做的任何更改。

下图显示了单层复制场景：

图 6.1. 单层复制



647_RHDS_0624

单一供应商服务器可以管理的使用者服务器总数取决于网络的速度以及每天修改的条目总数。

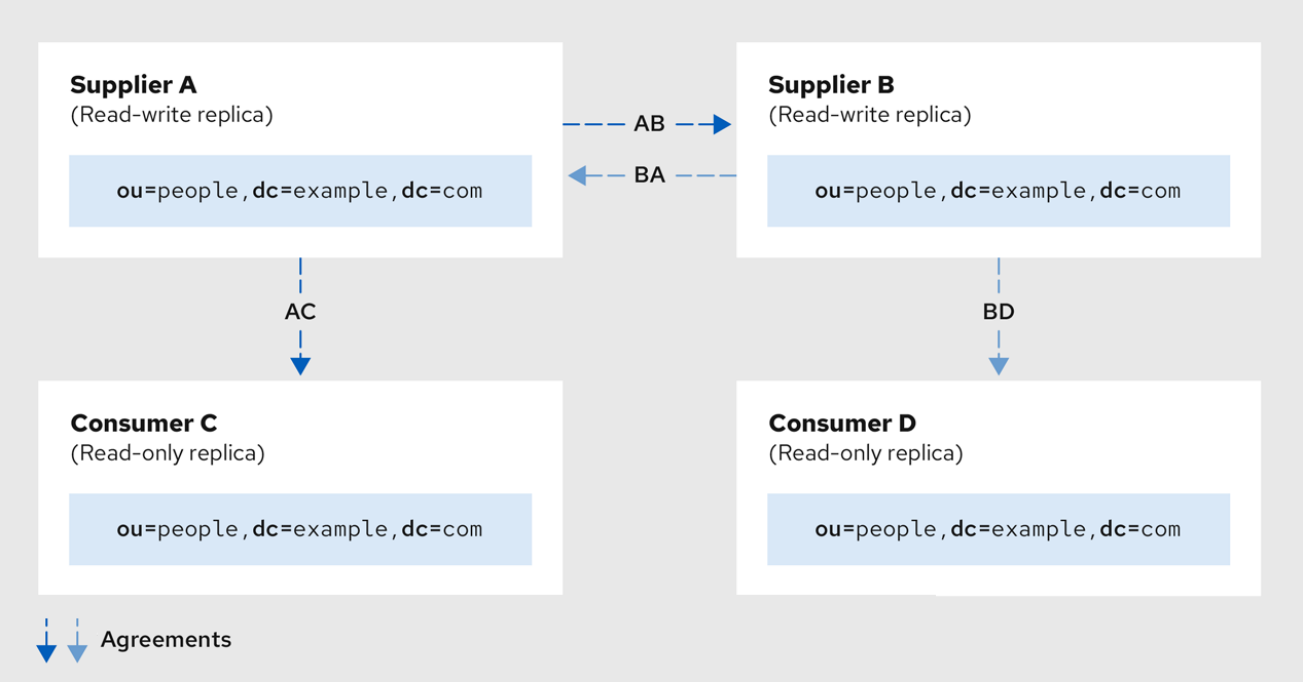
6.2.2. multi-supplier 复制

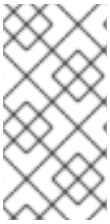
在多层次复制环境中，同一信息的主要副本可以存在于多个服务器上，并且可以在不同的位置同时更新目录数据。每台服务器上发生的更改复制到其他服务器上，这意味着每个服务器作为供应商和消费者的功能。

当在多个服务器上修改同一数据时，会发生复制冲突。使用冲突解析过程，目录服务器使用最新的更改作为有效的更改。

在多层次环境中，每个供应商都需要具有指向消费者和其他供应商的复制协议。例如，您使用两个供应商配置复制，供应商 A 和 Vendor B，以及两个消费者，Consumer C 和 Consumer D。另外，您决定一个供应商只更新一个消费者。在 Vendor A 上，您可以创建一个指向 供应商 B 和 Consumer C 的复制协议。在 Vendor B 上，您可以创建一个指向 供应商 A 和 Consumer D 的复制协议。下图演示了复制协议：

图 6.2. 使用两个供应商进行多层次复制





注意

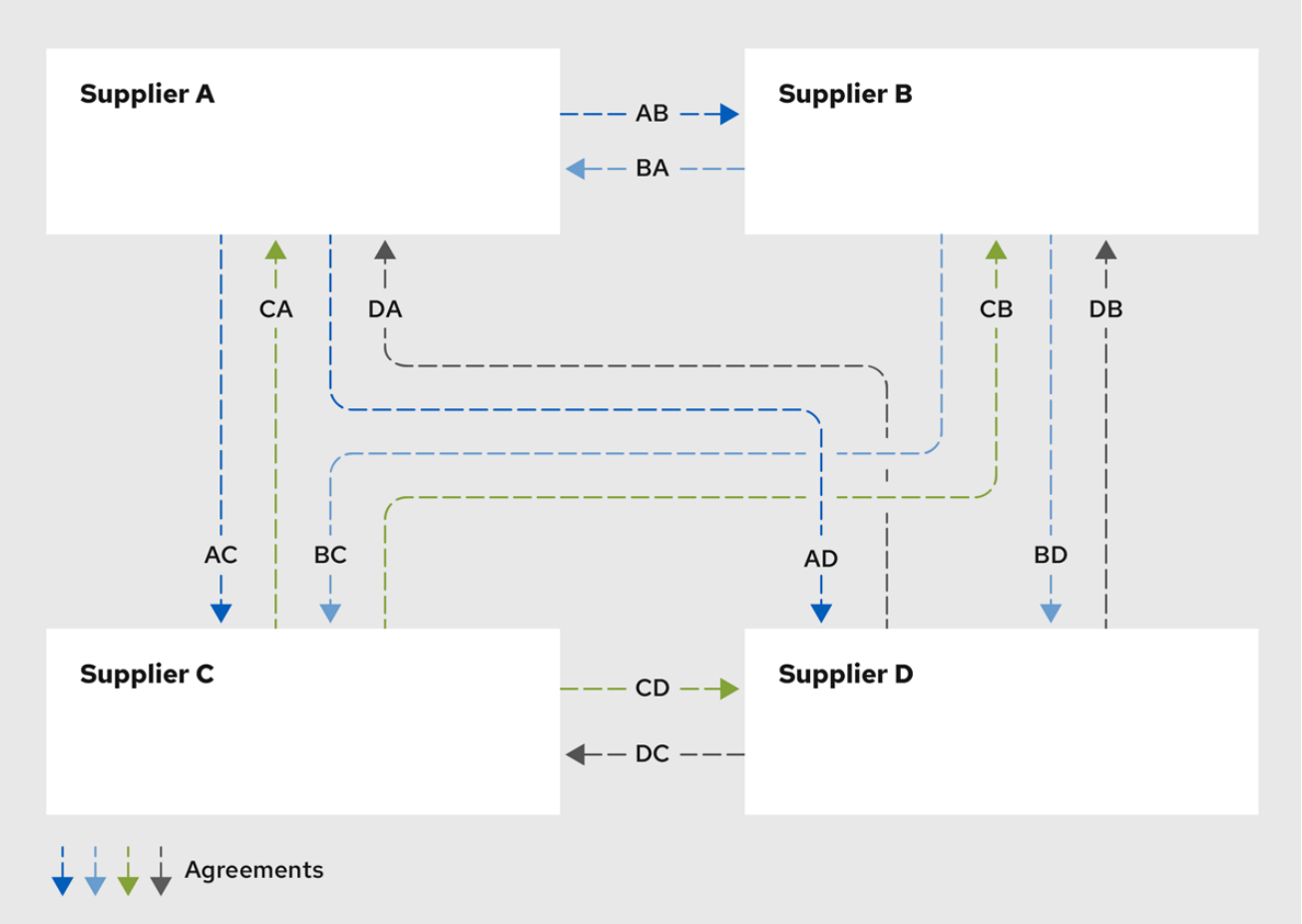
Red Hat Directory Server 在任何复制环境中支持最多 20 个供应商服务器，以及无限数量的 hub 和消费者服务器。

使用许多供应商需要创建一系列复制协议。此外，每个供应商都可以在不同的拓扑中配置，这意味着您的目录服务器环境可以有 20 个不同的目录树甚至模式差异。许多其他变量可能会对拓扑选择进行直接影响。

供应商可以将更新发送到所有其他供应商，或向供应商的一些子集发送更新。当更新发送到所有供应商时，更改会更快地传播，整个场景具有更好的容错能力。但是，它增加了供应商配置的复杂性，并带来高网络和高服务器需求。向供应商子集发送更新要更简单地配置和减少网络和服务器负载，但在出现多个服务器故障时会增加数据丢失的风险。

完全连接的网格拓扑

下图显示了一个完全连接的网格拓扑，其中四个供应商服务器将数据复制到所有其他供应商服务器。总之，在四个供应商服务器之间存在 Telve 复制协议，因为一个复制协议只描述了一个供应商和一个消费者之间的关系。

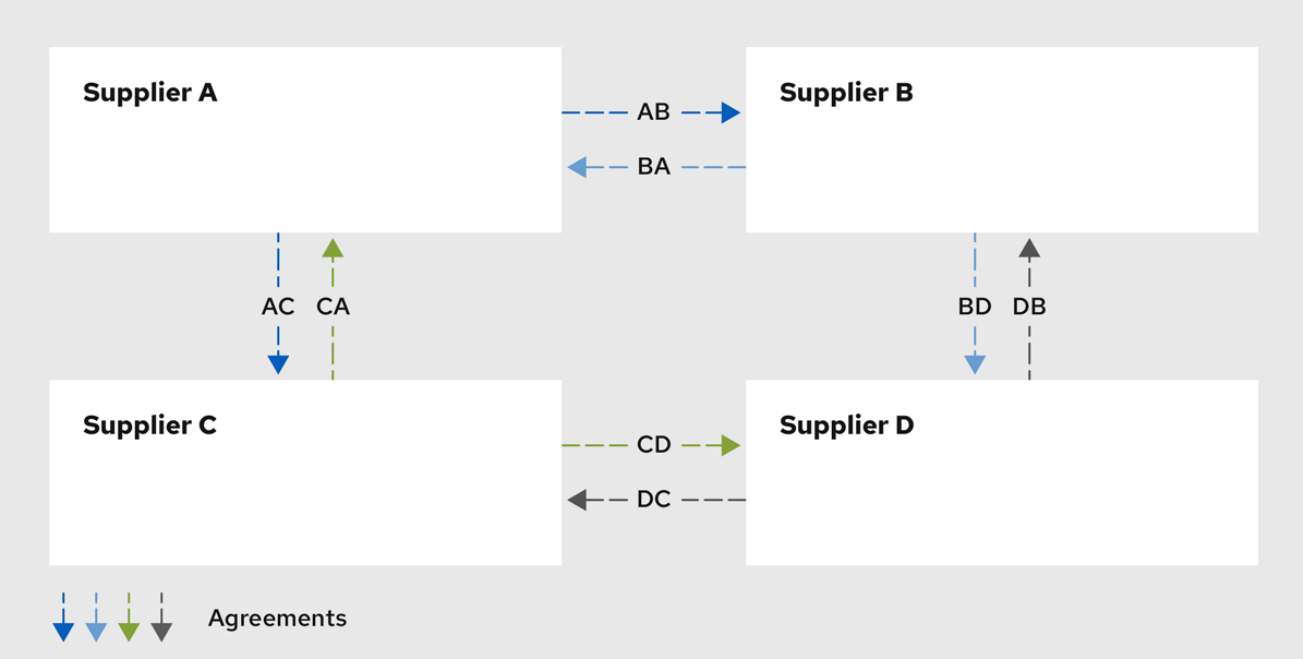


如果您有 20 个供应商，则需要在总计创建 380 个复制协议（每个有 19 个协议的 20 个供应商）。

如果两个或更多个服务器同时失败的可能性比较小或连接，请考虑使用部分连接的拓扑。

部分连接的拓扑

下图显示了每个供应商服务器将数据复制到两个供应商服务器的拓扑。与上例拓扑相比，只有 8 个复制协议存在于四个供应商服务器之间。



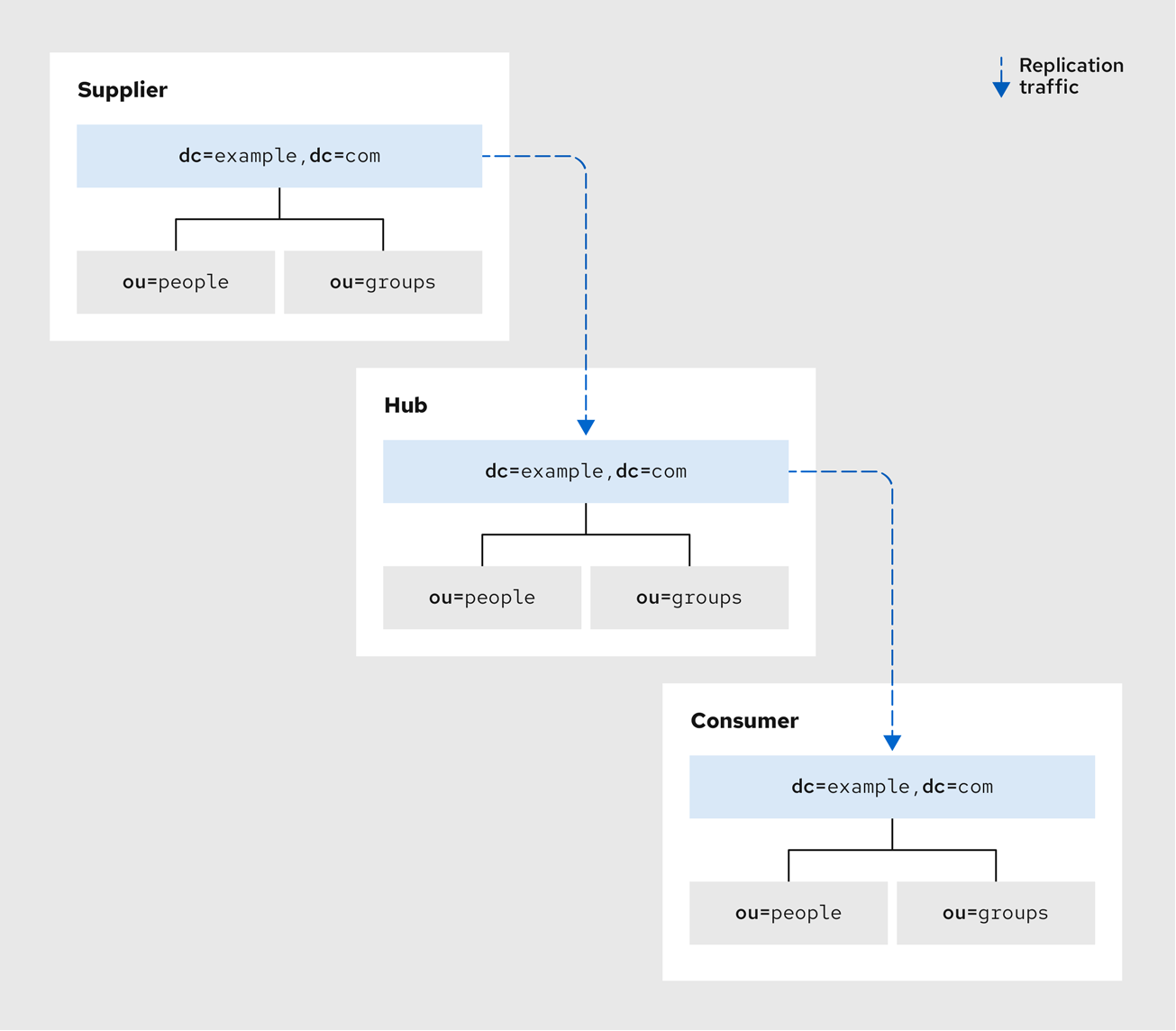
6.2.3. 级联复制

在级联复制场景中，hub 服务器从供应商服务器接收更新，并将这些更新发送到消费者服务器。hub 服务器是一个混合的，因为它包含只读副本，如典型的消费者服务器，它也像典型的供应商服务器一样维护一个更改日志。

Hub 服务器将供应商数据转发到消费者，并引用从目录客户端更新到供应商的请求。

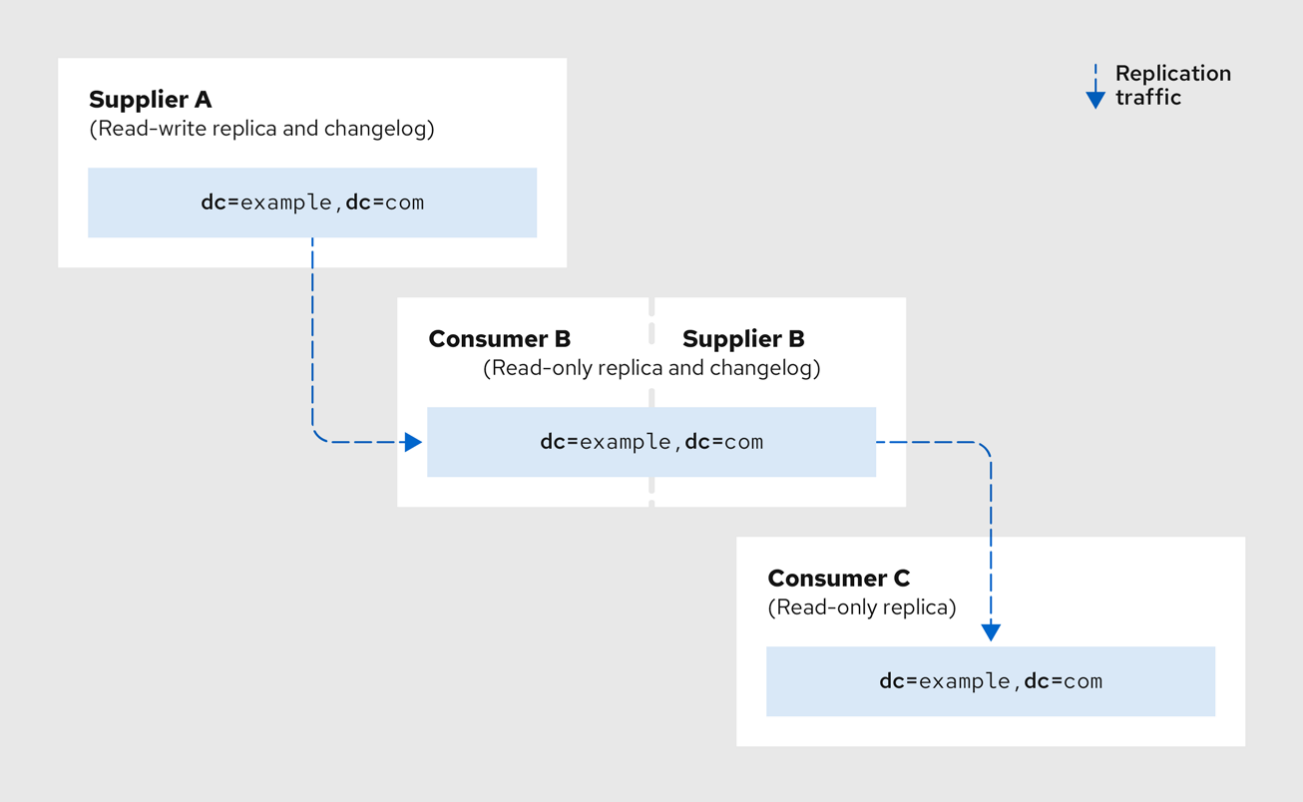
下图显示了级联复制场景：

图 6.3. 级联复制场景



下图显示了如何在每台服务器上配置副本（读写或只读），以及哪些服务器维护更改日志。

图 6.4. 在级联复制中复制流量和更改日志



在以下情况下级联复制很有用：

- *平衡繁重流量负载。由于复制拓扑中的供应商管理所有更新流量，因此可能会使它们负载过重，从而支持用户复制流量。您可以将复制流量重定向到可服务复制到大量消费者的 hub。*
- *通过在地理位置分散的环境中使用本地 hub 供应商来减少连接成本。*
- *以提高目录服务的性能。如果您将所有读取操作定向到消费者，并将所有更新操作都定向到供应商，您可以从 hub 服务器中删除所有索引（系统索引除外）。这将显著提高供应商和 hub 服务器之间的复制速度。*

其他资源

- *使用复制进行负载均衡*
- *使用复制来实现高可用性*

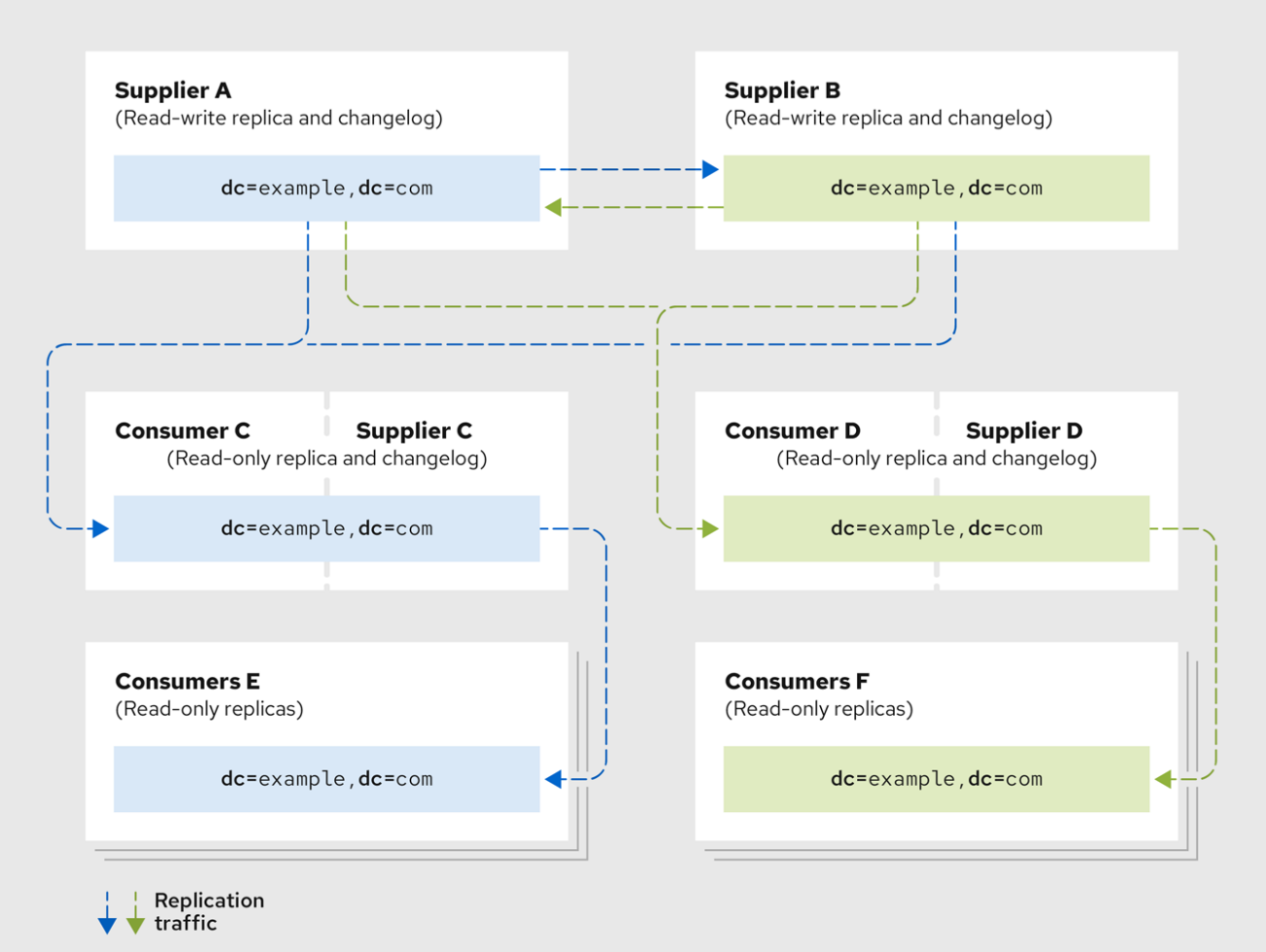
- *使用复制进行本地可用性*

6.2.4. 混合场景

可以合并任何复制场景，以满足网络和目录环境的需求。一个常见的组合是使用带有级联配置的多层次配置。

下图显示了混合场景的示例拓扑：

图 6.5. 合并多层次和级联复制



6.3. 定义复制策略

您可以根据您要提供的服务来确定复制策略。以下是您可以实现的常见复制策略：

- 如果高可用性是主要关注的，请在单一站点上创建一个包含多个目录服务器的数据中心。单层复制提供读取故障切换，而多层次复制则提供 write-failover。

如需了解更多详细信息，请参阅使用复制来实现高可用性。

- 如果本地可用性是主要关注，请使用复制将数据分布到位于全球本地办事处的目录服务器。您可以在单一位置（如公司总部）维护所有信息的主要副本，或者每个本地站点可以管理与它们相关的部分。

  如需了解更多详细信息，请参阅使用复制进行本地可用性。

- 要平衡目录服务器管理和避免网络拥塞的请求负载，请使用复制配置进行负载平衡。

  如需了解更多详细信息，请参阅使用复制进行负载平衡。

- 如果您将多个用户用于公司的不同位置或部分，或者某些服务器不安全，则使用 部分复制来排除 敏感或很少修改的信息，以在不损害敏感信息的情况下维护数据完整性。

  如需了解更多详细信息，请参阅 Fraction al 复制。

- 如果网络在多个站点和由多层次复制连接的本地数据供应商中扩展了多个目录服务器，请将复制配置用于广域网。

  如需了解更多详细信息，请参阅 跨各种网络复制。

要确定复制策略，首先对网络、用户、应用程序及其如何使用目录服务的调查开始。

6.3.1. 执行复制问卷调查

收集网络质量和使用信息，以帮助定义复制策略：

- 连接不同构建或远程站点以及可用带宽的 LAN 和 WAN 的质量。

- 用户的物理位置、每个站点有多少个用户，以及他们打算如何使用目录服务的使用模式。

  管理人工资源数据库或财务信息的网站通常会在目录上造成大量负载，而不是包含将目录用于电话的工程人员。

- 访问目录的应用程序数量以及 读、搜索 的相对百分比，并将 操作与 写入操作 进行比较。

  如果消息传递服务器使用 目录，请找出它处理的每个电子邮件消息所执行的操作数量。使用该目录的其他产品通常是身份验证应用程序或元目录应用程序等产品。对于每个应用程序，确定目录中执行的操作的类型和频率。

- 目录中存储条目的数量和大小。

6.3.2. 复制资源要求

复制需要资源。在定义复制策略时请考虑以下资源要求：

磁盘用量

在供应商服务器上，目录服务器在每个更新操作后写入更改日志。因此，接收许多更新操作的供应商服务器具有更高的磁盘用量。

服务器线程

每个复制协议都会创建一个专用的线程，CPU 负载取决于复制吞吐量。

文件描述符

服务器将一个文件描述符用于更改日志，以及每个复制协议的一个文件描述符。

6.3.3. 管理多层次复制所需的磁盘空间

在多层次拓扑中，供应商维护复制所需的额外日志，包括目录编辑的 changelog、更新条目的状态信息以及已删除条目的 tombstone 条目。由于这些日志文件可能会变得非常大，所以您必须定期清理这些文件，以避免不必要的磁盘空间使用。

在每个服务器上，您可以使用以下属性在复制环境中配置复制日志维护：

- nsslapd-changelogmaxage 属性设置 changelog 中条目的最长期限。当条目早于最长期限值后，Directory 服务器会删除该条目。设置条目的最长期限可防止更改日志无限期增长。

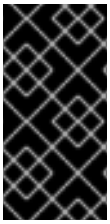
nsslapd-changelogmaxentries 属性设置 changelog 可以包含的最大条目数。请注意，nsslapd-changelogmaxentries 值必须足够大，以包含一组完整的目录信息。否则，多层次复制可能会出现问题。

- nsDS5ReplicaPurgeDelay 设置更改日志中 tombstone (deleted)条目和状态信息的最长期限。当 tombstone 或 state 信息条目早于该年龄后，Directory 服务器会删除该条目。nsDS5ReplicaPurgeDelay 值只适用于 tombstone 和状态信息条目，而 nsslapd-changelogmaxage 适用于 changelog 中的每个条目，包括目录修改。

- nsDS5ReplicaTombstonePurgeInterval 属性设置服务器运行清除操作的频率，以清理更改日志的 tombstone 和 state 条目。确保最长期限超过复制更新调度的最长期限。否则，在更新副本时可能会出现多层次复制问题。

6.3.4. 使用复制来实现高可用性

在单一服务器出现故障时，使用复制来防止目录不可用。至少，将本地目录树复制到至少一个备份服务器。您为容错复制的频率取决于您的要求。但是，根据目录所使用的硬件和网络的质量将此决策为基础。不可靠的硬件需要更多备份服务器。



重要

不要将复制用作常规数据备份策略的替代，因为复制和备份具有不同的目的。有关备份目录数据的详情，请参考 备份和恢复红帽目录服务器。

您可以选择以下策略来防止目录不可用：

- 要保证所有目录客户端的写入故障转移，请使用 多层次复制。

- 要保证 read-failover，请使用 single-supplier 复制。

LDAP 客户端应用程序通常配置为仅搜索一个 LDAP 服务器。如果您没有自定义客户端应用程序通过位于不同 DNS 主机名的 LDAP 服务器轮转，则您只能配置 LDAP 客户端应用程序以查看目录服务器的单一 DNS 主机名。因此，您可能需要使用 DNS 循环或网络排序来为备份目录服务器提供故障切换。

6.3.5. 使用复制进行本地可用性

根据网络的质量以及您的数据是关键任务，您可能需要将复制用于本地可用性。

将复制用于本地可用性，理由如下：

- 您需要一个本地的数据主副本。

  大型的跨国企业可能需要仅维护某个国家/地区的员工感兴趣的目录信息。此外，拥有本地数据主副本对于任何企业而言对于在部门或组织级别上控制数据的任何企业非常重要。

- 您有不可靠或间歇性可用的网络连接。

  国际网络具有不可靠的 WAN，导致网络连接不稳定。

- 您有定期的、会影响目录服务器性能的网络负载。

  具有老化网络的企业可能会在正常工作时间内遇到大量网络负载。

- 您要减少供应商上的网络负载和工作负载。

  即使网络可靠且可用，您可能想降低网络成本。

6.3.6. 使用复制进行负载均衡

复制目录数据的一个主要原因是平衡网络的工作负载，并提高目录性能。

因为目录条目的大小通常是 1 KB，每个目录搜索都会在您的网络负载中添加大约 1 KB。如果您的目录用户每天执行十个目录搜索，则每个目录用户大约需要 10 KB 的网络负载。如果您有一个缓慢、大量加载或不可靠的 WAN，您可能需要将目录树复制到本地服务器。

但是，确定本地可用数据是否值得复制导致的网络负载增加的成本。如果您将整个目录树复制到远程站点，您可能会在网络上增加更大的负载，与用户搜索中产生的流量进行比较。如果您的目录更改频繁，但当远程站点只有几个用户每天执行一些目录搜索时，这尤其如此。

下表将复制目录的负载影响与 100万个条目（每天 100,000 个条目）进行了比较，以及每天执行 10 个员工的小远程站点的负载影响。

表 6.1. 在网络上复制和远程搜索的影响

加载类型	access/day	平均条目大小	Load
复制	100,000	1KB	100MB/day
远程搜索	1,000	1KB	1MB/day

在不过载网络的情况下将数据提供给本地站点之间有妥协，就是使用调度复制。有关数据一致性和复制计划的更多信息，请参阅 数据一致性。

其他资源

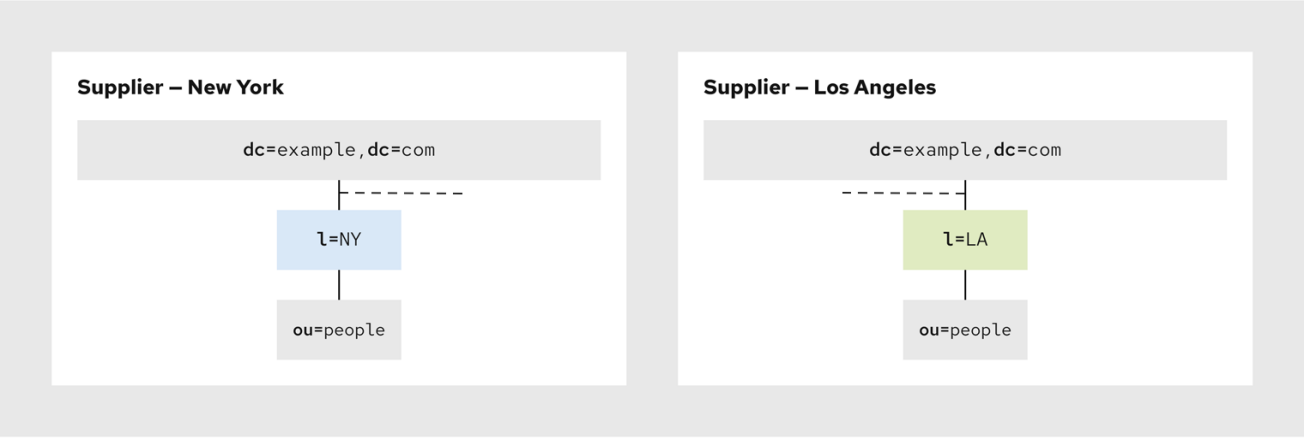
- 网络负载均衡示例
- 提高性能的负载平衡示例
- 小站点的复制策略示例
- 大型站点的复制策略示例

6.3.6.1. 网络负载均衡示例

这个示例描述了在 New York (NY)和 Los Angeles (LA)和 Los Angeles (LA)设有办事处的企业，每个办公室管理单独的子树。

下图显示了企业如何管理子树：

图 6.6. 企业 NY 和 LA 子树



647_RHDS_0624

每个办公室都包含一个高速网络，但两个城市之间的连接是不可靠的。要平衡网络负载，请使用以下策略：

- 为每个办公室选择一个服务器作为本地管理数据的供应商服务器。

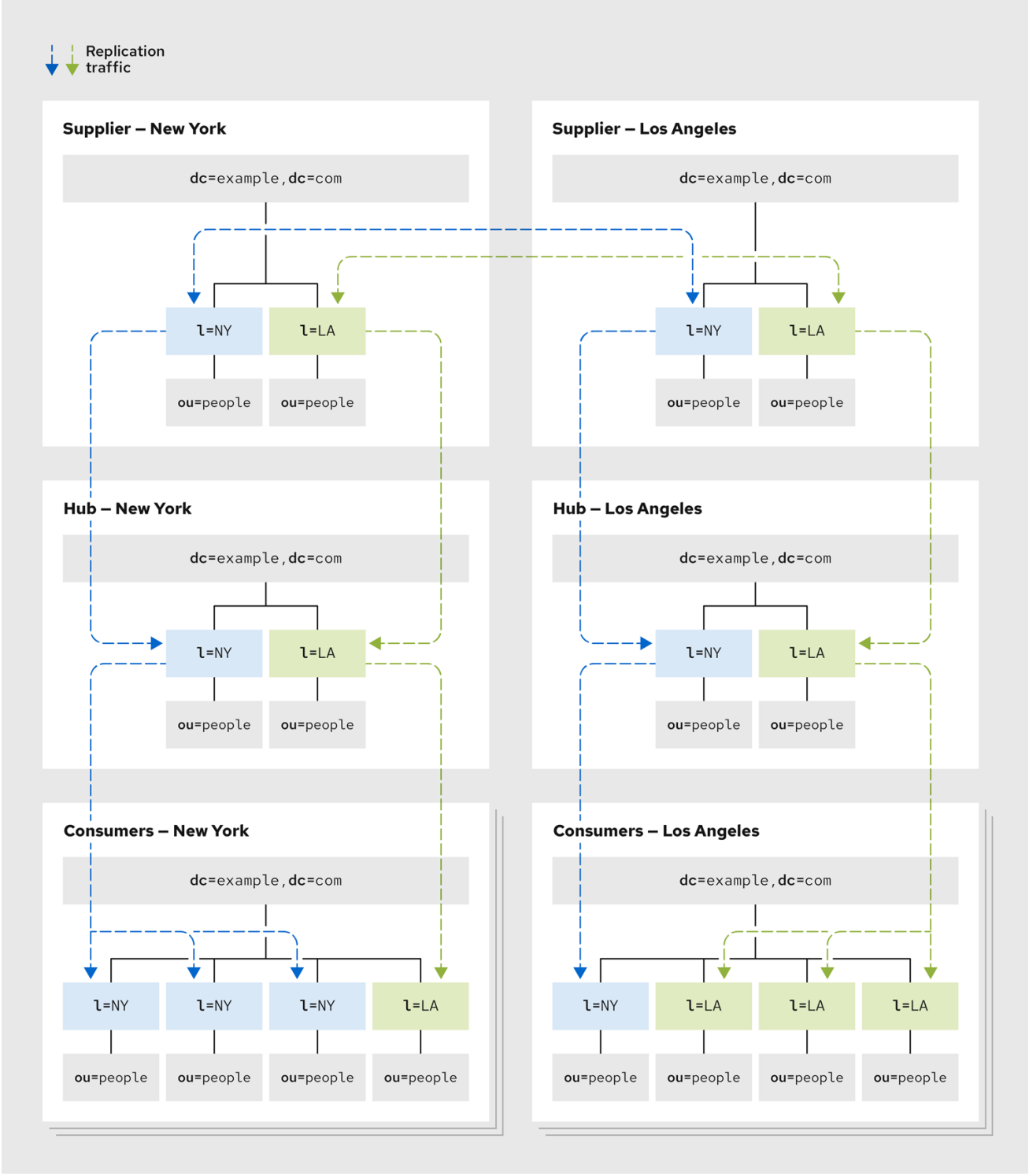
  将本地管理的数据从该供应商复制到远程办公室中的相应供应商服务器。当您在每个位置中有主要数据副本时，用户不会在不可靠的连接上执行更新和搜索操作。因此，性能会被优化。

- 将每个供应商服务器上的目录树（包括从远程办公室提供的数据）复制到至少一个本地目录服务器，以确保目录数据的可用性。

- 在每个位置配置级联复制，并增加用于搜索本地数据的消费者数量，以提供进一步的负载平衡。

  NY 办公室生成比 LA 特定搜索更具体的搜索。该示例显示了带有三个 NY 数据消费者和一个 LA 消费者的 NY 办事处。LA LA 办事处有 3 个 LA 数据消费者和一个 NY 数据消费者。

图 6.7. 企业负载平衡示例



其他资源

- *关于后缀*
- *cascading-replication*

- *multi-supplier 复制*

6.3.6.2. 提高性能的负载平衡示例

这个示例描述了具有以下特征的企业：

- *该目录包括支持 1,000,000 个用户的 1,500,000 个条目。*
- *每个用户每天都执行 10 个目录搜索。*
- *消息传递服务器每天处理 25,000,000 个邮件，并且执行五个目录搜索每个邮件。*
- *用户分布到四个时区。*

这相当于总计每天的 135,000,000 个目录搜索：

1,000,000 个用户 x 10 搜索 = 每天 10,000,000 名用户搜索

25,000,000 邮件 x 5 搜索 = 每天 125,000,000 邮件搜索

10,000,000 + 125,000,000 = 135,000,000 个每天搜索

*随着营业日的 8 小时工作日和用户分布到四个时区，在四个时区内，峰值使用量延长至 12 小时。因此，目录服务器必须在 12 小时内支持 135,000,000 目录搜索。此等于为 3,125 搜索每秒搜索 (135,000,000 /(60*60*12))。*

如果运行 Directory 服务器的硬件支持每秒读取 500 个，则必须使用至少 6 个或 7 个目录服务器来支持这个负载。对于具有一百万目录用户的企业，为本地可用性添加更多目录服务器。

在这种情况下，您可以使用以下复制策略：

- 将两个目录服务器放在多层次配置中，以一个城市处理所有写入流量。

  此配置假设您想对所有目录数据进行单点控制。

- 使用供应商服务器复制到一个或多个 hub。

  点消费者的 读取、搜索 和比较 请求，使供应商只能处理 写入请求。有关 hub 的更多信息，请参阅 Cascading-replication。

- 使用 hub 复制到整个企业的本地站点。

  复制到本地站点有助于平衡服务器和网络上的负载，并确保目录数据的高可用性。

- 在每个站点，至少复制一次以确保高可用性，至少用于 读取操作。

  使用 DNS sort 以确保本地用户始终找到他们可以用于目录搜索的本地目录服务器。

6.3.6.3. 小站点的复制策略示例

示例企业具有以下特征：

- 整个企业包含在单一构建中。
- 构建速度非常快（每秒100 Mb 每秒）和轻量级使用网络。
- 网络非常稳定，服务器硬件和操作系统平台是可靠的。
- 单个服务器可以轻松处理负载。

在这种情况下，您需要至少复制一次，以便在关闭主服务器进行维护或硬件升级时来确保可用性。另外，设置一个 DNS 循环，以便在其中一个目录服务器不可用时提高 LDAP 连接性能。

6.3.6.4. 大型站点的复制策略示例

从 Example 复制策略为小型站点 的示例企业已增长到更大的位置，现在具有以下特征：

- 公司包含在两个单独的构建中，构建 A 和构建 B。
- 在正常工作时间内，构建之间的连接速度缓慢且非常繁忙。
- 每个构建都有一个非常快（每秒100 Mb 每秒）和轻量级使用的网络。
- 每个构建中的网络非常稳定，服务器硬件和操作系统平台是可靠的。
- 单个服务器可在一个构建中轻松处理负载。

对于这样的条件，您的复制策略包含以下步骤：

- 在两个构建中选择一个服务器，以包含目录数据的主副本。

  将服务器置于构建中，其中包含负责目录数据主副本的最大人数，例如构建 A。
- 在构建 A 内至少复制一次，以获得目录数据的高可用性。

  使用 multi-supplier 复制 配置来确保写入故障切换。
- 在第二个构建 B 中创建两个副本。
- 如果您在供应商和消费者服务器之间不需要密切一致性，请仅在非高峰期内调度复制。

6.3.7. 部分复制

使用部分复制，您可以选择一组目录服务器不会从供应商复制到消费者或其他供应商的属性。因此，

可以在不复制数据库包含的所有信息的情况下复制数据库。

每个复制协议启用并配置部分复制。目录服务器对所有条目同样应用属性排除。排除的属性始终对消费者没有值。因此，对消费者服务器执行搜索的客户端永远不会看到排除的属性，即使搜索过滤器明确指定这些属性。

在以下情况下使用部分复制：

- 消费者服务器使用较慢的网络进行连接。很少更改的属性或较大的属性（如 jpegPhoto ）会减少网络流量。
- 消费者服务器放置在不受信任的网络中，如公共互联网。排除敏感属性（如电话号码）提供了额外的保护级别，以确保即使服务器访问控制措施被攻击者禁止访问敏感属性，也不会访问敏感属性。

6.3.8. 在广域网络间复制

广域网络(WAN)通常具有更高的延迟、更高的带宽延迟产品，速度低于局域网。当供应商和消费者使用广泛网络连接时，目录服务器支持有效的复制。

在以前的版本中，目录服务器使用的复制协议非常敏感，因为供应商只发送一个更新操作，然后等待消费者的响应。这会导致缩短延迟更高的吞吐量。

目前，供应商在不等待响应的情况下向消费者发送多个更新和条目，复制吞吐量与局域网的吞吐量类似。

在使用 WAN 时，请考虑以下性能和安全问题：

- 使用传输层安全(TLS)协议保护在公共网络（如互联网）中执行的复制。
- 对网络使用 T1 或更快的互联网连接。
- 避免在为通过 WAN 进行复制创建协议时在服务器之间持续同步。复制流量可能会消耗大量带宽，并减慢整个网络和互联网连接的速度。

其他资源

- *提高多层次复制环境中的延迟*

6.4. 对其他目录服务器功能使用复制

要更好地设计复制策略，了解复制和其他目录服务器功能之间的交互。

6.4.1. 复制和访问控制

目录存储访问控制指令(ACI)作为条目的属性，Directory 服务器会与其他目录内容一起复制这些 ACI。例如，要从特定主机限制对目录的访问，请使用 ACI 中的特定于主机的设置。否则，当 ACI 复制到其他服务器时，所有服务器上将拒绝对该目录的访问，因为 Directory 服务器在本地评估 ACI。

有关为目录设计访问控制的更多信息，请参阅 设计访问控制。

6.4.2. 复制和目录服务器插件

复制可用于 Directory 服务器提供的大多数插件。但是，以下插件在多层次环境中存在限制和例外：

- *属性唯一插件*

  Attribute Uniqueness 插件验证添加到本地服务器上的条目的属性值的唯一性。例如，公司需要 mail 属性对用户条目是唯一的。当两个不同的用户同时在两个不同的供应商服务器上添加 mail 属性的值时，Directory 服务器会将这些用户添加到目录中，因为没有命名冲突，因此不会发生复制冲突。属性唯一插件不会检查复制的更改，因此 mail 属性值将变为目录中的非唯一性。

- *参考完整性插件*

  当仅在多层次集中的一个供应商启用时，引用完整性可用于多层次复制。这样可确保仅在其中一个供应商服务器上发生引用完整性更新，并传播到其他服务器。

- *auto Membership 和 MemberOf 插件*

要使这两个插件在复制环境中正常工作，请配置插件以在每台服务器上本地执行更新。

注意

默认情况下，插件被禁用，您必须手动启用它们。

其他资源

- 服务器插件功能参考
- 列出用户条目中的组成员资格

6.4.3. 复制和数据库链接

当您使用链在目录中分发条目时，包含数据库链接的服务器指的是包含实际数据的远程服务器。在这种情况下，您无法复制数据库链接。但是，您可以复制包含远程服务器上实际数据的数据库。

6.4.4. 模式复制

在复制环境中，在参与复制的所有服务器上，该架构必须一致。为确保模式一致性，请只在单一供应商服务器上进行模式修改。

如果您在服务器间配置了复制，则默认发生模式复制。

Standard 模式

目录服务器使用以下场景进行标准模式复制：

1. 在将数据推送到消费者服务器前，供应商服务器会检查其模式版本是否与消费者服务器中保存的模式版本相同。
2. 如果供应商和消费者上的模式条目都相同，复制操作将继续。
3. 如果供应商模式版本比消费者模式版本更新，供应商服务器会在继续数据复制前将其模式复

制到消费者。

4. 如果供应商模式版本早于消费者模式版本，则复制可能会失败，或者服务器可能会在复制过程中返回错误，因为消费者中的模式不支持新数据。因此，切勿 更新消费者服务器上的模式。您必须只在复制拓扑中的供应商服务器上维护架构。

目录服务器使用 dsconf 命令、 Web 控制台、 LDAP 修改操作或直接发送到 99user.ldif 文件，将更改复制到架构。

如果您在两个供应商服务器上进行模式修改，消费者从两个供应商接收数据，每个供应商都具有不同的模式。消费者应用具有更新的 schema 版本的供应商的修改。在这种情况下，消费者的架构始终与其中一个供应商不同。要避免这种情况，请始终确保您只在一个供应商上进行模式修改。

您不需要创建特殊的复制协议来复制模式。但是，同一目录服务器可以包含供应商和消费者副本。因此，始终将充当 schema 的供应商的服务器标识，然后在复制环境中作为 schema 信息的用户在复制环境中设置复制协议。

有关标准模式文件的更多信息，请参阅 *标准模式*。

Custom 模式

如果您使用标准的 99user.ldif 文件作为自定义模式，目录服务器只会将自定义模式复制到所有消费者。目录服务器不会复制其他自定义模式文件或更改这些文件，即使您通过 Web 控制台或 dsconf 命令进行了更改。

如果使用其他自定义文件，则必须在供应商更改时手动将这些文件复制到拓扑中的所有服务器上。

其他资源

- *自定义模式*
- *目录服务器在复制环境中管理模式更新*。

第 7 章 设计安全目录

designing-rhds

红帽目录服务器如何保护数据会影响到之前的所有设计区域。任何安全设计都需要保护目录中的数据，并满足用户和应用程序的安全性和隐私需求。

了解如何分析安全需求以及如何设计目录以满足这些需求。

7.1. 关于安全威胁

目录可能面临潜在的安全威胁风险。了解最常见的威胁有助于概述整体安全设计。对目录安全性的威胁分为三个主要类别：

- 未授权访问
- 未授权的篡改
- 拒绝服务

7.1.1. 未授权访问

防止目录不受未授权访问的影响可能看似简单，但实施安全解决方案可能比先出现更加复杂。目录信息交付路径有多个潜在的访问点，未授权客户端可以访问数据。

以下场景只描述了一些未授权客户端如何访问目录数据的示例：

- 未授权的客户端可以使用另一个客户端凭证来访问数据。特别是当目录使用未保护的密码时。未授权的客户端也可以在合法客户端和目录服务器之间交换的信息。
- 未经授权的访问可以从公司内部进行，或者，如果公司连接到一个 extranet，还是从公司外部连接到互联网。

Directory 服务器提供的身份验证方法、密码策略和访问控制机制可以有效地防止未经授权的访问。

其他资源

- 选择适当的验证方法
- 设计密码策略
- 设计访问控制

7.1.2. 未授权的篡改

如果入侵者可以访问目录服务器和客户端应用程序之间的目录或截获通信，则他们可以修改或篡改目录数据。如果客户端不信任数据，或者目录本身无法信任它从客户端接收的修改和查询，则目录服务无需使用。

例如，如果目录无法检测到篡改，攻击者可以将客户端请求更改为服务器，或者不转发它，并将服务器响应更改为客户端。TLS 和类似技术可以通过在连接结束时签名信息来解决这个问题。

其他资源

- 有关在目录服务器中使用 TLS 的更多信息，请参阅 保护服务器连接

7.1.3. 拒绝服务

在拒绝服务攻击时，攻击者的目标是防止目录为其客户端提供服务。例如，攻击者可能会使用所有系统资源，因此防止其他人使用这些资源。目录服务器可以通过设置分配给特定绑定 DN 的资源的限制来防止拒绝服务攻击。有关根据用户绑定 DN 设置资源限值的更多信息，请参阅 用户管理和身份验证 指南。

7.2. 分析安全需求

分析环境和用户，以确定特定的安全需求。设计安全目录 一章中的站点调查说明了在目录中可读取和写入单个数据的基本决策。此信息构成了安全设计的基础。

目录服务用于支持业务的方式定义如何实施安全性。为 Intranet 提供服务的目录不需要与支持向互联网

打开的 extranet 或电子商务应用程序的目录相同。

如果目录只为内部网提供服务，请考虑信息需要什么级别的访问：

- 如何为用户提供和应用程序，并可访问执行其作业所需的信息。
- 如何保护员工或业务方面的敏感数据。

如果该目录服务于 extranet 或支持互联网上的电子商务应用程序，请考虑以下额外点：

- 如何为客户提供隐私保证。
- 如何保证信息的完整性。

7.2.1. 确定访问权限

数据分析确定了访问目录服务所需的信息用户、组、合作伙伴、客户和应用程序。可以通过两种方式之一授予权限：

- 尽可能多授予权限，同时仍然保护敏感数据。

  开放方法需要准确确定哪些数据对业务敏感或至关重要。

- 授予每个类别用户完成其作业所需的最小访问权限。

  限制的方法需要详细了解机构内每个用户的信息需求，以及可能外的内容。

无论用于确定访问权限的方法是什么，都创建一个一个简单的表，该表列出了机构中用户的类别以及授予每个权限的访问权限。考虑创建一个列出目录中保存敏感数据的表，并针对各个数据，这是保护数据所采取的步骤。

其他资源

- 有关检查用户身份的详情，请参考 选择适当的身份验证方法 部分。

- 有关限制访问目录信息的详情，请参考 设计访问控制 部分。

7.2.2. 确保数据保密性和完整性

当使用目录支持通过外部网与业务合作伙伴交换时，或者支持互联网上的电子商务应用程序，请确保交换数据的隐私性和完整性。

使用以下方法确保数据隐私性和完整性：

- 加密数据传输。

- 使用证书签署数据传输。

其他资源

- 有关加密方法目录服务器提供的详情，请参考 密码存储方案部分

- 有关签名数据的详情，请参考 保护服务器连接 部分。

- 有关在目录服务器数据库中加密敏感信息的详情，请参考 加密数据库 一节。

7.2.3. 执行常规审计

作为额外的安全措施，执行常规审计，通过检查日志文件和 SNMP 代理记录的信息来验证总体安全策略的效率。

其他资源

- 有关监控目录服务器的更多信息，请参阅监控服务器和数据库活动

- 有关日志文件的更多信息，请参阅 日志文件参考

7.2.4. 安全需要分析示例

这些示例展示了不可变 ISP 公司 example.com 如何分析其安全需求。example.com 提供 Web 托管和互联网访问。example.com 活动的一部分是托管客户端公司的目录。它还提供对多个单独订阅者的互联网访问。因此，example.com 在其目录中有三个主要的信息：

- example.com 内部信息
- 属于企业客户的信息
- 与个人订阅者相关的信息

example.com 需要以下访问控制：

- 向自己的目录信息提供托管公司的目录管理员（如 example_a 和 example_b ）的访问权限。
- 为托管公司目录信息实施访问控制策略。
- 为所有使用 example.com 进行互联网访问的单个客户端实施标准访问控制策略。
- 拒绝对所有外部的 example.com 公司目录的访问。
- 向世界授予 example.com 订阅者读取访问权限。

7.3. 安全方法概述

目录服务器提供多种设计符合特定需求的总体安全策略的方法。安全策略应该足够强大，以防止未经授权的用户修改或检索敏感信息，但也足以便于管理。复杂的安全策略会导致错误，导致人们无法访问或更糟糕的信息，允许人们修改或检索它们不应该被访问的目录信息。

表 7.1. 目录服务器中的可用安全方法

安全方法	描述
身份验证	验证其他方的身份。例如，客户端在 LDAP 绑定操作期间为目录服务器提供密码。
密码策略	定义密码必须满足的条件才能考虑此密码有效。例如，年龄、长度和语法。
Encryption	保护信息的隐私。当数据被加密时，只有接收者才能理解数据。
Access control	定制授予不同目录用户的访问权限，并提供指定所需凭证或绑定属性的方法。
帐户取消激活	禁用用户帐户、帐户组或整个域，以便目录服务器自动拒绝所有验证尝试。
安全连接	通过使用 TLS、StartTLS 或 SASL 加密连接来维护信息的完整性。如果在传输过程中加密信息，接收者可以确定在传输过程中没有被修改。设置最低安全强因素，从而需要安全连接。
Auditing	确定目录的安全性是否已被破坏。一个简单的审核方法是查看目录维护的日志文件。
SELinux	使用红帽目录服务器机器上的安全策略来限制和控制对目录服务器文件和流程的访问。

合并了在安全设计中维护安全性的各种工具，并纳入目录服务的其他功能，如复制和数据分发，以支持安全设计。

7.4. 选择适当的验证方法

有关安全策略的基本决定是用户如何访问该目录。匿名用户访问该目录，或者每个用户都需要使用用户名和密码（验证）登录目录？

了解目录服务器提供的身份验证方法。该目录对所有用户使用相同的身份验证机制，无论它们是人员还是 LDAP 感知应用程序。

7.4.1. 匿名和未经身份验证的访问

匿名访问提供了目录最简单的访问形式。通过匿名访问权限，连接到该目录的任何人都可以访问数

据。

当您配置匿名访问时，您无法跟踪谁执行哪种搜索类型，而只有某人执行搜索。您可以尝试阻止特定的用户或组访问某些类型的目录数据，但如果匿名访问允许该数据，则仍可以通过匿名绑定到目录来访问数据。

您可以限制匿名访问。通常，目录管理员仅允许匿名访问读、搜索和比较特权，而不用于写入、添加、删除或自我写入特权。通常，管理员限制对包含通用信息（如名称、电话号码和电子邮件地址）的属性子集的访问。您不应该允许匿名访问更敏感数据，如政府身份识别号，例如美国、家电话号码和地址以及工资信息。

如果您需要严格访问目录数据的规则，您可以完全禁用匿名访问。

当用户试图使用用户名绑定但没有用户密码属性时，未经身份验证的 绑定是。例如：

```
ldapsearch -x -D "cn=jsmith,ou=people,dc=example,dc=com" -b "dc=example,dc=com" "
(cn=joe)"
```

如果用户没有尝试提供密码，目录服务器会授予匿名访问权限。未经身份验证的绑定不需要绑定 DN 为现有条目。

与匿名绑定一样，您可以通过限制对数据库的访问来禁用未经身份验证的绑定以提高安全性。另外，您可以禁用未经身份验证的绑定，以防止客户端的绑定失败。有些应用程序可能认为它已成功验证到该目录，因为它收到了一个绑定成功消息，因此无法传递密码，而只是与未经身份验证的绑定连接。

7.4.2. 简单绑定和安全绑定

如果不允许匿名访问，用户必须对该目录进行身份验证，然后才能访问目录的内容。通过简单的密码身份验证，客户端通过发送可重复使用的密码来向服务器进行身份验证。

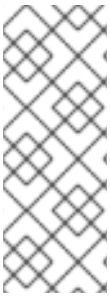
例如，客户端使用 bind 操作向目录进行身份验证，它提供可分辨名称和一组凭证。服务器在目录中查找与客户端 DN 对应的条目，并检查客户端给出的密码是否与以该条目存储的值匹配。如果存在，服务器会验证客户端。如果没有，身份验证操作会失败，客户端会收到错误消息。

绑定 DN 通常与个人的条目对应。但是，一些目录管理员更喜欢绑定为机构条目，而不是作为个人绑定。目录要求用于绑定的条目具有允许 userPassword 属性的对象类。这样可确保目录识别绑定 DN 和密码。

大多数 LDAP 客户端从用户隐藏绑定 DN，因为用户可能会发现要记住的 DN 字符的长字符串。当客户端尝试从用户隐藏绑定 DN 时，它会使用以下绑定算法：

1. 用户输入唯一标识符，如用户 ID。例如，fchen。
2. LDAP 客户端应用程序搜索该标识符的目录，并返回相关的可分辨名称。例如：uid=fchen,ou=people,dc=example,dc=com。
3. LDAP 客户端应用程序使用检索到的区分名称和用户提供的密码绑定到目录。

简单的密码身份验证提供了一种简单的方法来验证用户，但它需要额外的安全方法。考虑将其使用限制为组织内部网。要通过 extranet 与业务合作伙伴间的连接使用，或与互联网上的客户进行传输，可能最好需要安全（加密）连接。



注意

简单密码验证的缺陷是以纯文本形式发送密码。如果未授权用户正在侦听，这可能会破坏目录的安全性，因为该用户能够模仿授权用户。nsslapd-require-secure-binds 配置属性需要使用 TLS 或启动 TLS 在安全连接中进行简单的密码身份验证。这会有效地加密纯文本密码，使其不能被恶意参与者嗅探。

使用 nsslapd-require-secure-binds 配置属性，通过 TLS 或 Start TLS 打开安全连接。SASL 身份验证或基于证书的身份验证也可以进行。当目录服务器和客户端应用程序相互建立安全连接时，客户端通过不以明文传输密码来执行带有额外保护级别的简单绑定。

其他资源

- 保护服务器连接
- nsslapd-require-secure-binds 配置属性描述。

7.4.3. 基于证书的验证

另一种形式的目录身份验证涉及使用数字证书绑定到目录。当用户首次访问密码时，该目录会提示用户输入密码。但是，密码不与 目录中存储的密码不匹配，而是会打开用户证书数据库。

如果用户提供正确的密码，目录客户端应用程序会从证书数据库获取身份验证信息。然后，客户端应用程序和目录使用此信息通过将用户证书映射到目录 DN 来识别用户。目录允许或拒绝访问此身份验证过程中确定的目录 DN。

其他资源

- 保护红帽目录服务器

7.4.4. 代理身份验证

代理身份验证是一种特殊的身份验证形式，因为请求访问该目录的用户没有绑定到自己的 DN，而是使用 proxy DN。

代理 DN 是一个实体，它有适当的权限来执行用户请求。当个人或应用程序收到代理权限时，它们将任何 DN 指定为代理 DN，但目录管理器 DN 除外。

代理权限的一个主要优点是，LDAP 应用程序可以使用单个线程和一个绑定来服务多个用户对目录服务器发出请求。客户端应用程序使用代理 DN 绑定到目录服务器，而不是为每个用户绑定和身份验证。

代理 DN 在 LDAP 操作中指定，客户端应用提交。例如：

```
ldapmodify -D "cn=Directory Manager" -W -H ldap://server.example.com -X
"dn:cn=joe,dc=example,dc=com" -f mods.ldif
```

在这个版本中，管理器条目 cn=Directory Manager 会收到用户 cn=joe 的权限，以将修改应用到 mods.ldif 文件。管理器不需要提供用户密码来进行此更改。

注意

代理机制非常强大，您必须谨慎使用。代理权限在访问控制列表(ACL)范围内授予，当您授予用户代理权限时，这个用户可以代理目标下任何用户。您不能将代理权限限制为特定用户。

例如，如果条目具有对 dc=example,dc=com 树的代理权限，则此条目可以执行任何操作。因此，请确保在目录的最低可能级别上设置代理访问控制指令(ACI)。

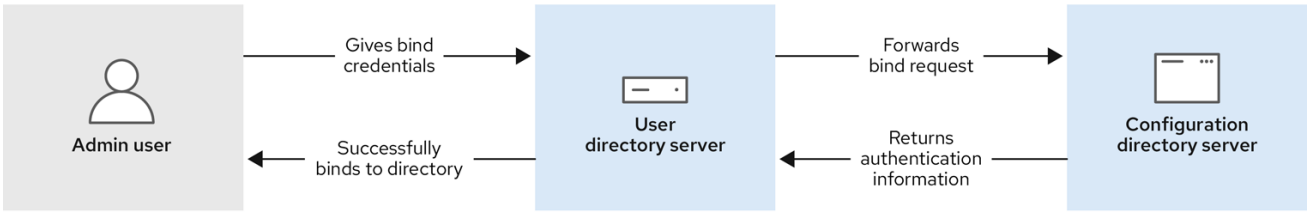
其他资源

- *管理访问控制*

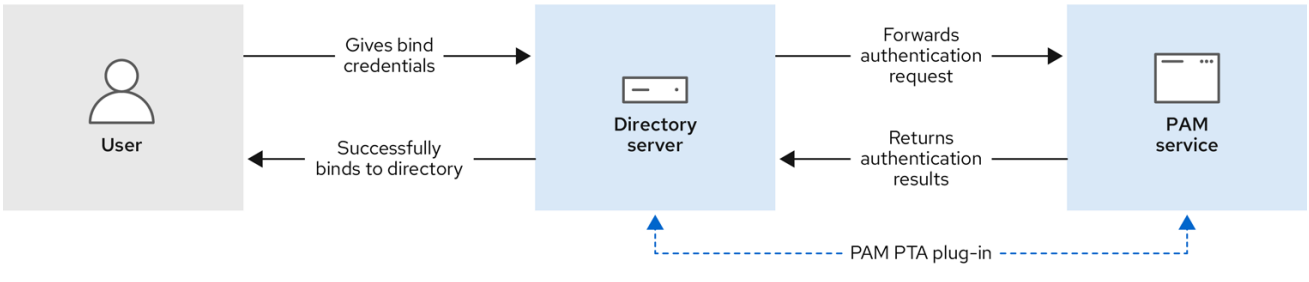
7.4.5. 直通身份验证(PTA)

直通身份验证(PTA) 是 Directory 服务器将任何身份验证请求从一台服务器转发到另一台服务器时。

例如，当目录服务器将实例的所有配置信息存储在另一个目录实例中时，目录服务器对 User Directory 服务器使用直通身份验证来连接到配置目录服务器。PTA 插件处理目录服务器到目录服务器直通身份验证。



许多系统已经具有 Unix 和 Linux 用户的身份验证机制，如可插拔验证模块(PAM)。您可以配置 PAM 模块，以告知 Directory 服务器为 LDAP 客户端使用现有验证存储。目录服务器与 PAM 服务交互，以使用 PAM 直通身份验证插件来验证 LDAP 客户端。



使用 PAM 直通身份验证时，当用户尝试绑定到目录服务器时，目录服务器会将凭据转发到 PAM 服务。如果凭据与 PAM 服务中的信息匹配，则用户可以成功绑定到 Directory 服务器，且所有目录服务器访问控制限制和帐户设置。

注意

您可以将目录服务器配置为使用 PAM，但您不能将 PAM 配置为使用 Directory 服务器进行身份验证。

您可以使用系统安全服务守护进程(SSSD)配置 PAM 服务。只需将 PAM 直通身份验证插件指向 SSSD 使用的 PAM 文件，如 /etc/pam.d/system-auth。SSSD 可以使用各种不同的身份提供程序，包括 Active Directory、红帽目录服务器或其他目录，如 OpenLDAP 或本地系统设置。

7.4.6. 免密码身份验证

身份验证会首先评估用户帐户是否可以进行身份验证。帐户必须遵循以下条件：

- 它必须处于活动状态。
- 不能锁定。
- 它必须根据任何适用的密码策略具有有效的密码。

有时，当用户不应该或无法绑定到目录服务器时，客户端应用需要执行用户帐户的身份验证。例如，系统可能使用 PAM 管理系统帐户，并将 PAM 配置为使用 LDAP 目录作为其身份存储。但是，系统使用免密码凭证，如 SSH 密钥或 RSA 令牌，且这些凭据无法传递给目录服务器。

红帽目录服务器支持用于 LDAP 搜索 的帐户 Usability Extension Control 扩展。此扩展会为每个返回的条目返回一个额外行，给出帐户状态以及有关该帐户密码策略的一些信息。然后，客户端或应用程序可以使用该状态来评估该用户帐户外的身份验证尝试。基本上，这控制信号用户是否被允许进行身份验证，而无需执行身份验证操作。

另外，您可以将此扩展与系统级服务（如 PAM）搭配使用，允许免密码登录，这些登录仍使用 Directory 服务器来存储身份，甚至控制帐户状态。

注意

默认情况下，只有目录管理器可以使用帐户 Usability Extension Control。要允许其他用户使用控制，请在支持的控制条目 oid=1.3.6.1.4.1.42.2.27.9.5.8,cn=features,cn=config 上设置适当的 ACI。

其他资源

- 检查帐户可用性以进行免密码访问

7.5. 设计帐户锁定策略

帐户锁定策略可通过防止未经授权或破坏对该目录的访问来保护目录数据和用户密码。目录服务器锁定或 取消激活、帐户后，该用户无法绑定到该目录，任何身份验证操作都会失败。

使用 nsAccountLock operational 属性来实现帐户取消激活。当条目包含 nsAccountLock 属性且值为 true 时，服务器会拒绝该帐户的绑定尝试。

目录服务器可以根据特定的自动条件定义帐户锁定策略：

- 目录服务器可以将帐户锁定策略与密码策略关联。当用户在指定次数后使用正确的凭证登录时，Directory 服务器会锁定帐户，直到管理员手动解锁它。

  这样的策略通过重复尝试猜测用户密码，防止尝试破坏该目录的恶意执行者。

- 目录服务器可以在过一定时间后锁定帐户。您可以使用此策略来控制临时用户的访问权限，如 interns、个人或 seasonal worker，它们会根据帐户创建的时间限时的访问权限。另外，您可以创建一个帐户策略，如果帐户在上一次登录时间不活动一段时间，则激活用户帐户。

  使用帐户策略插件实施基于时间的帐户锁定策略，并为该目录设置全局设置。您可以为不同的过期时间和类型创建多个帐户策略子条目，然后通过类服务将这些策略应用到条目。

其他资源

- 设计密码策略

7.6. 设计密码策略

密码策略是一组规则，用于管理在给定系统中如何使用密码。目录服务器密码策略指定密码必须满足的条件，如年龄、长度以及用户是否可以重复使用密码。

7.6.1. 密码策略如何工作

目录服务器支持精细的密码策略，这意味着 Directory 服务器在目录树中的任意点定义密码策略。目录服务器在以下级别上定义密码策略：

整个目录

这样的策略称为 全局 密码策略。当您配置和启用此策略时，Directory 服务器会将其应用到目录中的所有用户，但 Directory Manager 条目和启用了本地密码策略的用户条目除外。

此策略类型可以为所有目录用户定义通用、单一密码策略。

目录的特定子树

此类策略称为 子树级别或 本地密码策略。当您配置和启用此策略时，Directory 服务器会将其应用到指定子树下的所有用户。

此策略类型在托管环境中很好，为每个托管公司支持不同的密码策略，而不是为所有托管公司实施单一策略。

目录的特定用户

此类策略称为 用户级别或 本地密码策略。当您配置和启用此策略时，Directory 服务器只会将其应用到指定用户。

此策略类型可以为不同的目录用户定义不同的密码策略。例如，指定一些用户每天更改其密码，一些用户每月更改，而所有其他用户每六个月更改一次。

默认情况下，Directory 服务器包含与全局密码策略相关的条目和属性，这意味着将相同的策略应用于所有用户。要为子树或用户设置密码策略，请在子树或用户级别添加额外的条目，并启用 cn=config 条目的 nsslapd-pwpolicy-local 属性。此属性充当交换机，打开和关闭细粒度密码策略。

您可以使用命令行或 Web 控制台更改密码策略。在命令行中，dsconf pwpolicy 命令更改全局策略，dsconf localpwp 命令会更改本地策略。您可以在 配置密码策略部分找到 设置密码策略 的步骤。

密码策略检查过程

您添加到目录的密码策略条目决定了 Directory 服务器应强制执行的密码策略的类型(global 或 local)。

当用户尝试绑定到目录时，Directory 服务器决定是否为用户条目定义并启用本地策略。目录服务器按以下顺序检查策略设置：

1. *目录服务器决定是否启用精细的密码策略。服务器检查 cn=config 条目中 nsslapd-pwpolicy-local 属性的值(on 或 off)。如果值设为 off，服务器会忽略子树和用户级别定义的策略，并强制实施全局密码策略。*

2. *目录服务器确定是否为子树或用户定义本地策略。服务器检查对应用户条目中的 pwdPolicysubentry 属性：*

   a. *如果属性存在，服务器会强制为用户配置本地密码策略。如果条目具有属性，但值为空或无效（例如，指向不存在的条目），服务器会记录错误消息。*

   b. *如果在用户条目中没有找到 pwdPolicysubentry 属性，服务器会在达到顶部前检查父条目、grandparent 条目和其他上一级条目。*

   c. *如果任何上级条目中都未找到 pwdPolicysubentry 属性，则服务器会应用全局策略。*

3. *服务器将用户提供的密码与用户条目中指定的值进行比较，以确保它们匹配。服务器也使用密码策略定义的规则来确保密码在允许用户绑定到目录之前有效。*

除了绑定请求外，如果请求中存在 userPassword 属性，则在添加和修改操作过程中也会进行密码策略检查。

修改 userPassword 的值会检查两个密码策略设置：

- *激活密码过期策略。如果没有满足最低期限要求，服务器会返回 约束Violation 错误。密码更新操作失败。*

-

激活了密码历史记录策略。如果 userPassword 属性的新值位于密码历史记录中，或者与当前密码相同，服务器会返回 约束Violation 错误。密码更新操作失败。

添加和修改 userPassword 的值会检查为密码语法设置的密码策略：

- 激活密码最小长度策略。如果 userPassword 属性的新值小于所需的最小值长度，服务器会返回 约束Violation 错误。密码更新操作失败。
- 激活密码语法检查策略。如果 userPassword 的新值与条目的 another 属性相同，服务器会返回 约束Violation 错误。密码更新操作失败。

7.6.2. 密码策略属性

了解可用于为服务器创建密码策略的属性。目录服务器在 cn=config 条目中保存密码策略属性，您可以使用 dsconf 实用程序更改这些设置。

故障的最大数量

此设置在密码策略中启用基于密码的帐户锁定。如果用户尝试登录一定次数并失败，Directory 服务器会锁定该帐户，直到管理员解锁或有选择地通过了一定的时间。使用 passwordMaxFailure 配置参数设置最大故障数。

当登录尝试达到限制时，目录服务器有两种方法来计数登录尝试并锁定帐户：

- 当数字命中时，目录服务器会锁定帐户(n)
- 目录服务器仅在计数超过时锁定帐户(n+1)。

例如，如果失败限制是三次尝试，可以在第三个失败尝试时锁定帐户(n)，或者在第四次失败尝试时锁定(n+1)。n+1 行为是 LDAP 服务器的历史行为，因此被视为传统行为。较新的 LDAP 客户端预期有一个更严格的硬限制。默认情况下，Directory 服务器使用严格的限制(n)，但您可以在 passwordLegacyPolicy 配置参数中更改旧的行为。

重置后更改密码

目录服务器密码策略可以指定用户在首次登录时或管理员重置密码后更改密码。管理员设置的默认密码通常遵循公司惯例，如用户初始、用户 ID 或公司名称。如果发现这个惯例，这通常是恶意参与者试图

破坏系统的第一个值。因此，建议在管理员重置这些密码后要求用户更改密码。

如果您为密码策略配置此设置，则即使禁用了用户定义的密码，用户需要更改密码。如果密码策略不需要或者不允许用户更改密码，则管理员分配的密码不应遵循任何明显的惯例，应该很难发现。

默认配置不需要用户在重置后更改密码。

用户定义的密码

您可以设置密码策略来允许或不允许用户更改自己的密码。良好的密码是强密码策略的关键。良好的密码不应使用普通词语，如字典单词、pet 或子代的名称、生日、用户 ID 或任何可轻松发现的用户的信息（或存储在目录本身中）。良好的密码应包含字母、数字和特殊字符的组合。但是，为了方便起见，用户通常使用易于记住的密码。因此，一些企业选择为用户设置密码，以满足强大密码条件，也不允许用户更改密码。

为用户设置密码有以下缺陷：

- 它需要大量的管理员时间。
- 因为管理员指定的密码通常更难以记住，用户更有可能写出密码，从而增加发现的风险。

默认情况下，允许用户定义的密码。

密码过期

密码策略可允许用户无限期地使用相同的密码，或指定在给定时间后过期的密码。通常，使用较长的密码会被发现。但是，如果密码经常过期，用户可能很难记住它们，并采取措施写出密码。常见策略是使密码每 30 到 90 天过期。即使禁用密码过期时间，服务器也会记住密码过期规格。如果重新启用了密码过期，则密码只在最后一次禁用前设定的时间有效。例如，如果您将密码配置为每 90 天过期，然后禁用并重新启用密码过期，则默认的密码过期持续时间会保留 90 天。

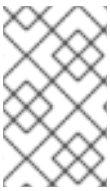
默认情况下，用户密码永不过期。

过期警告

如果您设置了密码过期周期，最好在用户密码过期前向用户发送警告。

当用户绑定到服务器时，目录服务器会显示警告。如果启用了密码过期，默认情况下，Directory 服务器通过使用 LDAP 消息向用户发送警告信息，在用户密码过期前一天。用户客户端应用应支持此功能。

密码过期警告的有效范围为从一到 24,855 天。



注意

在 Directory 服务器发送过期警告前，密码 永不 过期。

宽限期限制

过期的密码 宽限期 意味着用户仍然可以登录到系统，即使其密码已过期。要允许某些用户使用过期密码登录，请在密码过期后指定允许用户的宽限期尝试次数。

默认情况下，Directory 服务器不允许宽限期。

密码语法检查

密码语法检查强制实施密码字符串的规则，以便任何密码都必须满足或超过某些条件。所有密码语法检查都可以在全局、每个子树或每个用户应用。passwordCheckSyntax 属性管理密码语法检查。

默认密码语法要求最小密码长度为八个字符，且密码中没有使用任何普通词语。微小的词语是存储在 uid,cn,sn,givenName,ou, 或 mailattributes 中的任何值。

另外，您可以使用其他形式的密码语法强制，为密码语法提供不同的可选类别：

- 密码中需要的最小字符数(passwordMinLength)。
- 最小数字字符数，即零到 9 之间的数字(passwordMinDigits)。
- 最小 ASCII 字母字符数，包括大写和小写(passwordMinAlphas)。

- 最小大写 ASCII 字母字符数(passwordMinUppers)。
- 最小小写 ASCII 字母字符数(passwordMinLowers)。
- 最少的特殊 ASCII 字符数，如 !@#$ (passwordMinSpecials)。
- 最小 8 位字符数(passwordMin8 位)。
- 可以立即重复相同字符的次数上限，如 aaabbb (passwordMaxRepeats)。
- 密码需要的最小字符类别数；类别可以是大写或小写字母、特殊字符、数字或 8 位字符 (passwordMinCategories)。
- 目录服务器针对 CrackLib 字典(passwordDictCheck)检查密码。
- 目录服务器检查密码是否包含 palindrome (passwordPalindrome)。
- 目录服务器可防止设置同一类别中有更多连续字符的密码(passwordMaxClassChars)。
- 目录服务器可防止设置包含特定字符串的密码(passwordBadWords)。
- 目录服务器可防止设置包含管理员定义属性中设置的字符串的密码 (passwordUserAttributes)。

需要的更多语法类别，其密码越高。

默认情况下禁用密码语法检查。

密码长度

密码策略可能需要用户密码的最短长度。通常，较短的密码更易于破解。密码的建议最小长度为八个字符。这很难破解但很短，用户可以在不写出密码的情况下记住密码。此属性的有效值范围从两个到 512 个字符。

默认情况下，服务器没有最小密码长度。

密码最短期限

密码策略可以阻止用户更改指定时间的密码。当您将 passwordMinAge 属性与 passwordHistory 属性结合使用时，用户无法重复使用旧密码。例如，如果密码最短期限(passwordMinAge)属性为两天，则用户在单个会话期间无法重复更改密码。这可以防止它们通过密码历史记录加以利用，以便他们可以重复利用旧密码。

passwordMinAge 属性的值的有效范围从零到 24 855 天。值为零(0)表示用户可以立即更改密码。

密码历史记录

目录服务器可以在密码历史记录中存储两个到 24 密码。如果密码位于历史记录中，用户无法将其密码重置为旧密码。这可防止用户重复使用容易记住的一些密码。或者，您可以禁用密码历史记录，从而允许用户重复使用密码。

即使密码历史记录关闭，密码也会停留在历史记录中。如果重新打开密码历史记录，用户无法在禁用密码历史记录前重复使用历史记录中的密码。

默认情况下，服务器不会维护密码历史记录。

密码存储方案

密码存储方案指定了在目录中存储目录服务器密码的加密类型。目录服务器支持多个不同的密码存储方案：

基于密码的 Key Derivation Function 2 (PBKDF2_SHA256, PBKDF2-SHA1, PBKDF2-SHA256, PBKDF2-SHA512)

这是最安全的密码存储方案。默认存储方案为 PBKDF2-SHA512。

Salted Secure Hash Algorithm (SSHA、SSHA-256、SSHA-384 和 SSHA-512)

推荐的 SSHA 方案是 SSHA-256 或更强大的。

CLEAR

这意味着没有加密，它是 SASL Digest-MD5 唯一选项，因此使用 SASL 需要 CLEAR 密码存储方案。虽然目录存储的密码可以通过使用访问控制信息(ACI)指令来保护，但它仍然不是将纯文本密码存储在目录中的最佳选择。

安全哈希算法(SHA、SHA-256、SHA-384 和 SHA-512)

这比 SSHA 的安全性较低。

UNIX 加密

这个算法提供与 UNIX 密码的兼容性。

MD5

此存储方案比 SSHA 不太安全，但对于需要 MD5 的传统应用程序包括该存储方案。

Salted MD5

这个存储方案比普通 MD5 哈希更安全，但安全性仍低于 SSHA。此存储方案不包含与新密码搭配使用，但有助于从支持 salted MD5 的目录迁移用户帐户。

密码上次更改时间

passwordTrackUpdateTime 配置属性告知服务器记录最后一次目录服务器更新条目密码的时间戳。目录服务器将密码更改时间保存为用户条目中的操作属性 pwdUpdateTime，它与 modifyTimestamp 或 lastModified 操作属性分开。

默认情况下，服务器不会存储上次更改时间的密码。

其他资源

- cn=config 条目下的配置属性

7.6.3. 在复制环境中设计密码策略

目录服务器在复制环境中强制使用密码和帐户锁定策略，如下所示：

- 在数据供应商中强制执行密码策略。

- 在复制设置中的所有服务器上强制使用帐户锁定。

目录服务器在目录中复制密码策略信息，如密码年龄、帐户锁定计数器和过期警告计数器。但是，目录服务器不会复制配置信息，如密码语法和密码修改的历史记录。目录服务器在本地存储此信息。

在复制环境中配置密码策略时，请考虑以下点：

- 所有副本都发出 impending password expiration 的警告。目录服务器在每台服务器上保留此信息，因此如果用户依次绑定到多个副本，用户会多次收到相同的警告。另外，如果用户更改密码，副本可能需要一些时间才能接收此信息。如果用户更改密码，然后立即重新绑定，绑定可能会失败，直到副本注册更改为止。

- 同一绑定行为应该在所有服务器上发生，包括供应商和副本。始终在每台服务器上创建相同的密码策略配置信息。

- 帐户锁定计数器在多层次环境中可能无法按预期工作。

7.7. 设计访问控制

在决定身份验证方案后，决定如何使用这些方案来保护目录中所含的信息。访问控制可以指定某些客户端有权访问特定信息，而其他客户端则不能访问。

使用一个或多个 访问控制列表 (ACL)来定义访问控制。目录 ACL 由一个或多个 访问控制信息 (ACI)语句组成，允许或拒绝权限，如读、写、搜索和比较，与指定的条目及其属性进行比较。

使用 ACL，您可以在目录树的任何级别上设置权限：

- 整个目录

- 目录的特定子树

- 目录中的特定条目

- 特定的一组条目属性
- 任何与给定 LDAP 搜索过滤器匹配的条目

另外，您可以为特定用户、属于特定组的所有用户或目录的所有用户设置权限。您可以定义网络位置的访问，如 IP 地址(IPv4 或 IPv6)或 DNS 名称。

7.7.1. 关于 ACI 格式

在设计安全策略时，您需要了解 ACI 在目录中如何表示，以及您可以设置的权限。

目录 ACI 使用以下通用形式：

```
target permission bind_rule
```

ACI 变量有以下描述：

目标

指定条目，通常是子树，即 ACI 目标、其目标属性或两者。目标标识 ACI 应用到的目录元素。ACI 只能将一个条目为目标，但可以针对多个属性。此外，目标还可以包含 LDAP 搜索过滤器。您可以为包含通用属性值的广泛分散条目设置权限。

权限

标识 ACI 集合的实际权限。权限变量指出 ACI 允许或拒绝特定类型的目录访问，如读取或搜索，到指定的目标。

绑定规则

标识权限应用到的绑定 DN 或网络位置。绑定规则也可以指定 LDAP 过滤器，如果该过滤器被评估为绑定客户端应用程序，则 ACI 应用到客户端应用程序。

因此，对于目录对象目标，如果绑定规则为 true，ACI 允许或拒绝权限。

权限和绑定规则被设置为对，每个目标可以有多个权限绑定规则对。您可以有效地为任何给定目标设置多个访问控制。例如：

> *target (permission bind_rule)(permission bind_rule) ...*

其他资源

- *有关 ACI 格式的完整描述，请参阅管理访问控制*

7.7.1.1. 目标

ACI 可以以条目上的目录条目和属性为目标。

定位目录条目包括该条目及其在权限范围内的所有子条目。如果没有为 ACI 显式定义目标条目，则 ACI 目标到包含 ACI 语句的目录条目。ACI 只能以一个条目为目标，或者只有与单个 LDAP 搜索过滤器匹配的条目。

目标属性将权限仅应用到属性值的子集。当您以一组属性为目标时，指定 ACI 目标或 ACI 不明确目标的属性。排除目标设置权限中的属性，但对象类结构允许的一些属性。

其他资源

- *定位目录条目*
- *目标属性*

7.7.1.2. 权限

权限可以允许或拒绝访问。避免拒绝权限，请参阅 允许或拒绝访问

权限可以是在目录服务上执行的任何操作：

权限	描述
读	指明用户是否可以读取目录数据。
写	指明用户是否可以更改或创建目录。此外，此权限允许用户删除目录数据，但不能删除条目本身。但是，要删除整个条目，用户必须拥有删除权限。

权限	描述
搜索	指明用户是否可以搜索目录数据。这与 read 权限不同，如果用户作为搜索操作的一部分返回，则允许用户查看目录数据。 例如，如果您允许搜索通用名称(**cn**)并读取个人房间号码，则目录服务器可以将房间号码作为通用名称搜索的一部分返回。但是，用户无法使用房间号码作为搜索的主题。使用这个组合以防止人员搜索位于特定房间的人员。
比较	指明用户是否可以比较数据。比较权限表示搜索功能，但 Directory 服务器不会返回搜索实际目录信息。相反，Directory 服务器会返回一个简单的布尔值，指示比较值是否匹配。使用 compare 操作在目录身份验证期间匹配 **userPassword** 属性值。
自我写入	仅对组管理使用自我写入权限。使用这个权限，用户可以向组中添加或从组中删除自己。
添加	指明用户可以在目标条目下创建子条目。
删除	指明用户是否可以删除目标条目。
Proxy	表示用户可以使用除 Directory Manager 之外的任何其他 DN 访问此 DN 的权限的目录。

7.7.1.3. 绑定规则

*绑定规则定义 **ACI** 应用到的绑定 **DN** （用户）。它还可以指定 **bind** 属性，如当天或 **IP** 地址的时间。*

*另外，轻松绑定规则，使 **ACI** 仅适用于用户自己的条目。用户可以更新自己的条目，而无需运行更新另一个用户条目的风险。*

*绑定规则指示 **ACI** 适用时的以下情况：*

- *如果绑定操作到达特定的 **IP** 地址(**IPv4** 或 **IPv6**)或者 **DNS** 主机名。您可以使用它来强制从给定机器或网络域进行所有目录更新。*
- *如果用户匿名绑定。为匿名绑定设置权限意味着该权限适用于绑定到该目录的任何人。*

- 对于成功绑定到该目录的任何人。在阻止匿名访问时，您可以使用它来允许常规访问。
- 如果用户已绑定为条目的直接父项。
- 如果用户满足特定的 LDAP 搜索标准。

目录服务器为绑定规则提供以下关键字：

父

如果绑定 DN 是直接父条目，则绑定规则为 true。您可以授予允许目录条目管理其直接子条目的特定权限。

Self

如果绑定 DN 与请求访问的条目相同，则绑定规则为 true。您可以授予特定权限，以允许个人更新自己的条目。

All

对于成功绑定到该目录的任何人，绑定规则是 true。

Anyone

每个人都绑定规则都是 true。使用此关键字来允许或拒绝匿名访问。

7.7.2. 设置权限

默认情况下，Directory 服务器拒绝对所有用户的访问，但 Directory Manager 除外。因此，您必须设置 ACI，以便用户能够访问该目录。

7.7.2.1. 优先级规则

当用户尝试任何对目录条目的访问时，Directory 服务器会检查目录中设置的访问控制。要确定访问，目录服务器应用 优先级规则。此规则指出，当存在两个冲突的权限时，拒绝访问的权限优先于授予访问权限的权限。

例如，如果目录服务器拒绝目录根目录的写入权限，并且该权限适用于访问该目录的任何人，那么无论可能允许写出任何权限，任何用户都可以写入该目录。要允许特定用户对目录写入权限，您需要设置原

始 deny-for-write 的范围，使其不包括该用户。然后，您需要为用户设置额外的允许权限。

7.7.2.2. 允许或拒绝访问

您可以允许或拒绝对目录树的访问，但要小心地拒绝访问。由于优先级规则，如果目录服务器找到在更高级别拒绝访问的规则，它将拒绝较低级别的访问，无论可能授予访问权限的冲突权限是什么。

限制允许访问规则的范围只包括用户或客户端应用程序的最小可能子集。例如，您可以设置权限，以允许用户在其目录条目上写入任何属性，但拒绝除 Directory 管理员组成员以外的所有用户写入 uid 属性的权限。

或者，以以下方式编写允许写入访问的两个访问规则：

- 创建一个规则，允许向每个属性写入特权，但 uid 属性除外。此规则应适用于每个人。
- 创建一个规则，允许向 uid 属性写入特权。此规则应仅适用于 Directory Administrators 组的成员。

仅允许特权，避免设置显式拒绝特权。

7.7.2.3. 何时拒绝访问

很少需要设置显式拒绝权限，但在以下情况下非常有用：

- 您有一个大型目录树，其中包括复杂的 ACL。

  为了安全起见，目录服务器可能需要突然拒绝对特定用户、组或物理位置的访问。不必花费时间仔细检查现有的 ACL 以了解如何限制允许权限，而是临时设置显式拒绝特权，直到您有时间进行分析。如果 ACL 变得复杂，则拒绝 ACI 只在以后为管理开销增加成本。在可能的情况下，尽快取消相关的 ACL 以避免显式拒绝特权，然后简化整体访问控制方案。

- 您可以根据周中的一天或一天的一个小时设置访问控制。

  例如，目录服务器可以在 11:00 p.m 中拒绝从 Sunday 编写活动。(2300)至周一到周一下午 1 点。(0100).从管理的角度来看，管理 ACI 可能更容易地限制此类型的基于时间的访问，而不是

搜索所有 allow-for-write ACI 并限制其在这个时间段内的范围。

- 在委派目录管理机构到多个人时，您可以限制特权。

  要允许一个人或一组人管理目录树的某些部分，而无需修改树的某些方面，请使用明确拒绝特权。

  例如，要确保 Mail Administrators 不允许对通用名称(cn)属性进行写访问，请设置可明确拒绝对通用 name 属性的写访问的 ACI。

7.7.2.4. 放置访问控制规则的位置

您可以为目录中的任何条目添加访问控制规则。通常，管理员将访问控制规则添加到带有对象类 domainComponent,country,organization,organizationalUnit,inetOrgPerson, 或 group 的条目。将规则组织为尽可能多的组，以简化 ACL 管理。规则适用于其目标条目和所有条目子项。因此，最好将访问控制规则放在目录或目录分支点上，而不是跨各个叶条目（如个人）分散它们。

7.7.2.5. 使用过滤的访问控制规则

您可以使用 LDAP 搜索过滤器来设置与一组定义的条件匹配的任何目录条目的访问权限。例如，允许对包含 organizationalUnit 属性的任何条目的读取访问权限，该属性设置为 marketing。

过滤的访问控制规则允许预定义的访问级别。例如，该目录包含家地址和电话号码信息。有些人希望发布此信息，而其他人希望取消列出。

您可以使用以下方法配置访问：

1. 向名为 publishHomeContactInfo 的每个用户目录条目添加一个属性。
2. 设置一个访问控制规则，仅针对其 publishHomeContactInfo 属性设为 true 的条目授予对 homePhone 和 homePostalAddress 属性的读访问权限。使用 LDAP 搜索过滤器来表达此规则的目标。
3. 允许目录用户将自己的 publishHomeContactInfo 属性的值更改为 true 或 false。这样，目录用户可以决定此信息是否公开可用。

其他资源

LDAP 搜索过滤器

7.7.3. 查看 ACI ：获取有效的权限

get valid permissions (GER)是一个扩展的 ldapsearch 命令，该命令返回在条目中的每个属性上设置的访问控制权限。通过此搜索，LDAP 客户端可以决定服务器访问控制配置允许用户执行的操作。

访问控制信息被分为两组的访问权限：条目权利和属性权限。条目权限是仅限于该特定条目的权限，如修改或删除。属性权限是该目录中对该属性的每个实例的访问权限。

在以下情况下可能需要进行详细的访问控制：

- 您可以使用 GER 命令更好地组织目录的访问控制指令。与另一个组相比，通常需要限制一组用户可以查看或编辑的内容。例如，QA Managers 组的成员可能有权限搜索并读 manager 和 salary 等属性，当只有 HR Group 的成员有权限修改或删除它们。检查用户或组的有效权限是验证管理员是否设置适当的访问控制的方法。

- 您可以使用 GER 命令查看可以在您的个人条目上查看或修改哪些属性。例如，用户应具有对 homePostalAddress 和 cn 等属性的访问权限，但只能具有对 manager 和 salary 属性的读取访问权限。

其他资源

- 使用 Get Effective Rights search 检查条目的访问权限

- Get Effective Rights search 的常见场景

7.7.4. 使用 ACI：一些提示和技巧

以下提示有助于降低管理目录安全模型的管理负担，并改进目录性能特性：

- 最小化目录中的 ACI 数量。

虽然目录服务器可以评估超过 50,000 个 ACI，但难以管理大量 ACI 语句。大量 ACI 可让人工管理员立即决定特定客户端可用的目录对象。

目录服务器使用宏最小化目录中的 ACI 数量。使用宏在 ACI 目标或绑定规则或两者中代表 DN 或其部分。

- 平衡允许和拒绝权限。

  虽然默认规则是拒绝对任何没有特别授予访问权限的用户的访问，但最好使用一个 ACI 来减少 ACI 的数量，以允许访问树的根，以及少量拒绝 ACI 条目。这种情境可避免使用多个允许 ACI 接近叶条目。

- 识别 ACI 中的最小属性集合。

  当允许或拒绝对属性子集的访问时，请选择如果最小列表是允许或拒绝的属性集合。然后设置 ACI，以便它只需要管理最小列表。

  例如，person 对象类包含大量属性。要允许用户仅更新几个属性，请编写 ACI，仅允许对这些属性进行写入访问。但是，要允许用户更新除几个属性外的所有属性，请创建 ACI，允许除这几个命名的属性外进行写入访问。

- 仔细使用 LDAP 搜索过滤器。

  搜索过滤器不会直接命名您管理访问权限的对象。因此，使用它们可能会导致意外的结果。特别是，当目录变得更为复杂时。在 ACI 中使用搜索过滤器前，使用同一过滤器运行一个 ldapsearch 操作，从而使结果明确。

- 不要在目录树的不同部分中重复 ACI。

  防止重叠的 ACI。例如，如果目录根点上有一个 ACI，允许组对 commonName 和 givenName 属性进行写入访问权限，而另一个 ACI 则仅允许对 commonName 属性进行同一组写入访问权限，然后考虑更新 ACI，以便只有一个控制授予对组的写入访问权限。

  当目录增长更为复杂时，意外重叠的 ACI 的风险也会快速增加。通过避免 ACI 重叠，通过减少目录中包含的 ACI 总数来简化安全管理。

- 名称 ACI。

  虽然命名 ACI 是可选的，但为每个 ACI 提供一个简短的、有意义的名称有助于管理安全模型。

- 在目录中尽可能地对 ACI 进行分组。

  尝试将 ACI 位置限制为目录根点和主要目录分支点。对 ACI 进行分组有助于管理 ACI 的总列表，并帮助将目录中的 ACI 总数保持最小。

- 避免使用双负数，例如，如果绑定 DN 不等于 cn=Joe，则拒绝写入。

  虽然此语法对服务器完全可接受，但它不是人类可读的。

其他资源

- 使用宏访问控制指令

7.7.5. 将 ACI 应用到根 DN (Directory Manager)

通常，访问控制规则不适用于目录管理器用户。Directory Manager 在 dse.ldif 文件中定义，而不是在常规用户数据库中定义，ACI 目标不包括该用户。

目录管理器需要高级别的访问权限才能执行维护任务并响应事件。但是，您可以为 Directory Manager 授予一定级别的访问控制，以防止以 root 用户身份执行未经授权的访问或攻击。

使用 RootDN Access Control 插件设置特定于 Directory Manager 用户的特定访问控制规则：

- 基于时间的访问控制，允许或拒绝特定天数和特定时间范围的访问。

- IP 地址规则，用于允许或拒绝从定义的 IP 地址、子网和域的访问。

- 主机访问规则，以允许或拒绝对特定主机、域和子域的访问。

您只能为目录管理器设置一个访问控制规则。它位于插件条目中，它适用于整个目录。

重要

确保 Directory Manager 帐户具有适当的访问权限级别。此管理用户可能需要在非小时内或响应故障时执行维护操作。在这种情况下，设置太强的时间或一天规则可以防止 Directory Manager 用户有效地管理该目录。

其他资源

- 对目录管理器帐户 设置访问控制。

7.8. 加密数据库

数据库以纯文本形式存储信息。因此，访问控制措施可能无法完全保护一些非常敏感的信息，如政府身份识别号或密码。有可能获得服务器持久存储文件的访问权限，可以直接通过文件系统或访问丢弃的磁盘驱动器或存档介质。

使用数据库加密时，可以加密个别属性，因为它们存储在数据库中。配置后，特定属性的每个实例（甚至索引数据）都会加密，只能通过安全频道（如 TLS）访问。

其他资源

- 有关使用数据库加密的详情，请参考 管理属性加密 章节。

7.9. 保护服务器连接

在为识别的用户设计验证方案和用于保护目录中信息的访问控制方案后，下一步是为在服务器和客户端应用程序之间传递信息的完整性而设计的方法。

对于 server-to-client 连接和服务器到服务器的连接，目录服务器支持各种安全连接类型：

传输层安全性(TLS)

目录服务器可以通过 TLS 使用 LDAP 通过网络提供安全通信。为特定连接选择的加密方法是客户端应用程序和目录服务器之间的协商结果。

启动 TLS

目录服务器也支持启动 TLS，这是通过常规的未加密 LDAP 端口发起传输层安全(TLS)连接的方法。

简单身份验证和安全层(SASL)

SASL 是一个安全框架，可以用来配置不同的机制来向服务器验证用户，具体取决于您在客户端和服务器应用程序中启用的机制。另外，SASL 可以在客户端和服务器之间建立一个加密的会话。目录服务器使用 SASL 和 GSS-API，启用 Kerberos 登录，以及用于几乎所有服务器到服务器的连接，包括复制、串联和直通身份验证。目录服务器无法使用 SASL 与 Windows 同步。

对于处理敏感信息的任何操作（如复制）以及一些操作（如 Windows 密码同步），建议使用安全连接。目录服务器可以同时支持 TLS 连接、SASL 和非安全连接。

目录服务器可以同时支持 SASL 身份验证和 TLS 连接。例如，您配置了目录服务器实例，以要求 TLS 连接到服务器，并支持用于复制连接的 SASL 身份验证。这意味着不需要选择是否在网络环境中使用 TLS 或 SASL。

另外，您可以为到服务器的连接设置最低级别的安全性。安全强度因素措施在关键强度方面，如何实现安全连接的强度。您可以设置需要某些操作的 ACI，如密码更改，只有在连接是某个强度或更高强时进行。您还可以设置最小 SSF，它基本上可以禁用标准连接，并为每个连接需要 TLS、启动 TLS 或 SASL。目录服务器同时支持 TLS 和 SASL，服务器会计算所有可用连接类型的 SSF，并选择最强的连接类型。

其他资源

- *有关使用 TLS、启动 TLS 和 SASL 的更多信息，请参阅 保护红帽目录服务器*

7.10. 使用 SELINUX 策略

SELinux 是安全策略的集合，它为系统上的应用程序、进程和文件定义访问控制。安全策略是一组规则，告诉 SELinux 可以或无法访问哪些规则来防止未经授权的访问和篡改。

SELinux 对服务器上的文件、目录、端口、进程、用户和其他对象进行分类。SELinux 将每个对象放在适当的安全上下文中，以定义对象如何通过角色、用户和安全级别在服务器的行为中。SELinux 将对象的这些角色分组到域中，SELinux 规则定义如何允许一个域中的对象与另一个域中的对象交互。

目录服务器有以下域：

- Directory 服务器的 dirsrv_t
- SNMP 的 dirsrv_snmp_t
- LDAP 端口的 LDAP _port_t

这些域为 Directory 服务器的所有进程、文件、目录、端口、套接字和用户提供安全上下文：

- SELinux 为每个实例标记具有特定安全上下文的文件和目录。目录服务器使用的大多数主目录都有所有本地实例的子目录，无论有多少，SELinux 都会轻松地将单个策略应用到新实例。
- 每个实例的 SELinux 标签具有特定安全上下文的端口。
- SELinux 限制相应域中的所有目录服务器进程。
- 每个域具有特定的规则，它们定义域授权了哪些操作。
- 如果 SELinux 策略没有指定，SELinux 会拒绝对实例的任何访问。

SELinux 有三个不同的执行级别：

disabled

没有 SELinux

permissive

SELinux 处理规则被处理，但不会强制实施它们。

enforcing

SELinux 严格强制执行所有规则。

Red Hat Directory Server 定义了 SELinux 策略，允许它在严格的 SELinux enforcing 模式下以正常方式运行。目录服务器可以在不同的模式下运行，一个用于正常操作，一个用于数据库操作，如导入 (ldif2db 模式)。Directory 服务器的 SELinux 策略仅适用于正常模式。

默认情况下，Directory 服务器使用 SELinux 策略以正常模式运行。

其他资源

- *SELinux 的工作原理*

第 8 章 目录设计示例

目录服务的设计取决于企业的大小和性质。以下示例是开发实时目录服务部署计划的起点。

8.1. 本地企业设计示例

小公司 ExampleCom 是一个汽车部分制造商，有 500 名员工。ExampleCom 决定部署红帽目录服务器，以支持其使用的启用了目录的应用程序。

8.1.1. 本地企业的数据设计

为了决定目录要存储的数据类型，ExampleCom 会创建一个执行站点调查的部署团队。部署团队决定以下关键点：

- 消息传递服务器、Web 服务器、日历服务器、人工资源应用程序和白页应用将使用该目录。
- 消息传递服务器对 uid、mailServerName 和 mailAddress 等属性执行 精确 搜索。为提高数据库性能，ExampleCom 将维护这些属性的索引。

  有关使用索引的更多信息，请参阅使用索引提高数据库性能。
- 白页应用程序搜索用户名和电话号码。因此，目录必须处理大量频繁子字符串、通配符和返回大量结果集的模糊搜索。ExampleCom 公司决定维护以下索引：
  - cn,sn, 和 givenName 属性 的存在,equality,approximate, 和 substring 索引。
  - telephoneNumber 属性 的存在,equality, 和 substring 索引。
- 目录必须维护用户和组信息，以支持组织中部署的基于 LDAP 服务器的内部网。目录管理员组将管理大多数 ExampleCom 用户和组信息。但是，ExampleCom 需要一组单独的邮件管理员来管理电子邮件信息。
- 目录必须存储用户公钥证书，以支持公钥基础架构(PKI)应用，如 S/MIME 电子邮件。

8.1.2. 本地企业的架构设计

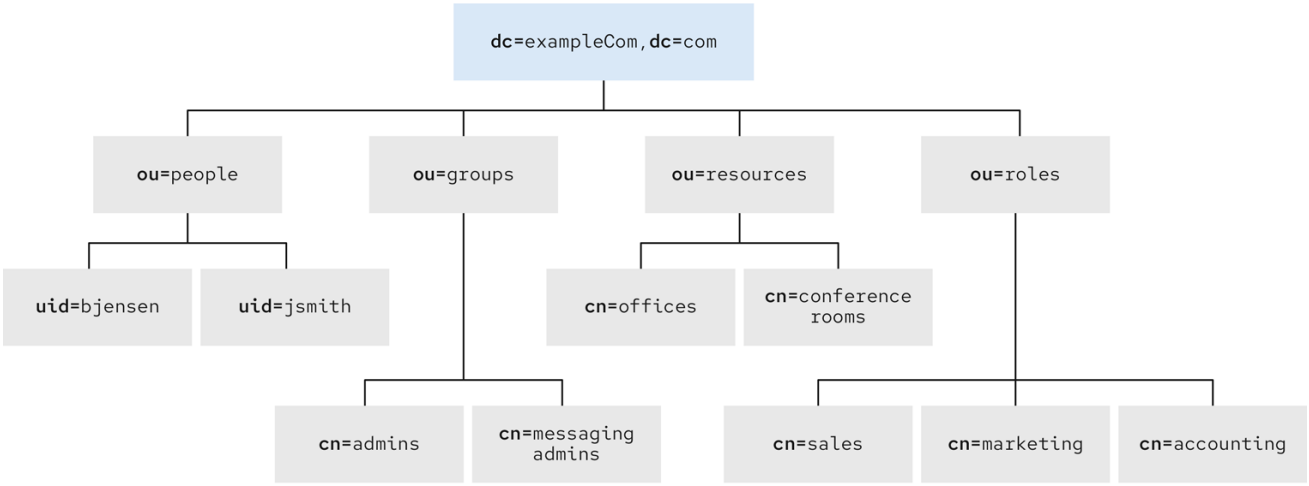
ExampleCom 目录支持的应用程序需要 userCertificate 和 uid (userID)属性。因此，ExampleCom 部署团队决定使用 inetOrgPerson 对象类代表目录中的条目，因为它允许这两个属性。

此外，ExampleCom 希望通过创建 examplePerson 对象类来代表员工来自定义默认目录模式。这个对象类来自 inetOrgPerson 对象类。examplePerson 允许一个 exampleID 属性。此属性包含分配给每个员工的特殊员工号码。以后，ExampleCom 可以在 examplePerson 对象类中添加新属性。

8.1.3. 本地企业目录树设计

根据准备的数据和模式设计，ExampleCom 会创建以下目录树：

图 8.1. ExampleCom 的目录树



- 目录树的根目录为 dc=example,dc=com，它是公司互联网域名。
- 目录树有四个分支点：
  - ou=people
  - ou=groups

- ou=resources
- ou=roles

- 所有 ExampleCom people 条目都是在 ou=people 分支下创建的。

  people 条目是 个人、OrganizationPerson、inetOrgPerson 和 examplePerson 对象类的所有成员。uid 属性唯一标识每个条目的可分辨名称(DN)。例如，公司包含 Babs Jensen 的条目 (uid=bjensen)和 Emily Stanton (uid=estanton)。

- 对于 ExampleCom 中的每个部门，将创建 销售、营销和 会计 角色。

  每个个人条目都包含一个 role 属性，用于标识个人所属的部门。现在，公司可以根据这些角色创建访问控制指令(ACI)。

  有关角色的更多信息，请参阅 第 4.3.2 节 “关于目录服务器中的角色”

- 以下组分支在 ou=groups 分支下创建：
  - cn=administrators 组包含管理目录内容的目录管理员的条目。
  - cn=messaging admins 组包含仅管理邮件帐户的邮件管理员。此组对应于消息传递服务器使用的管理员组。

- ou=resources 分支下的以下分支被创建：
  - 会议室的 ou=conference rooms 分支。
  - 办公室的 ou= offices 分支。

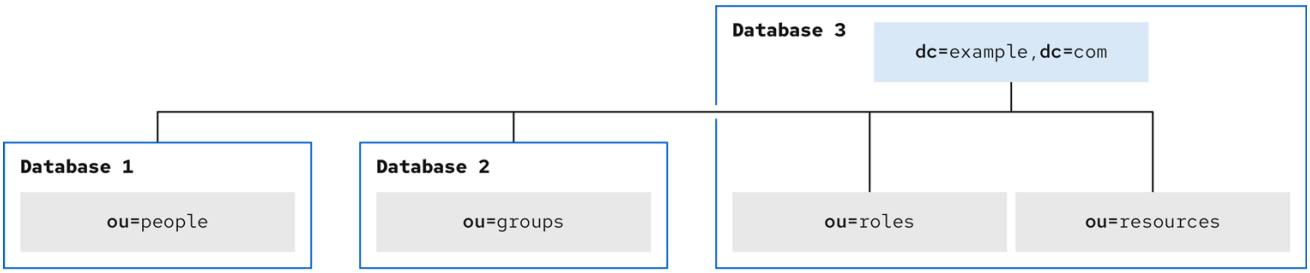
- 创建类服务(CoS)，根据条目是否属于管理组，为 mailquota 属性提供值。此 CoS 为管理员提供 100GB 邮件配额，而普通 ExampleCom 员工则具有 5GB 邮件配额。

8.1.4. 本地企业拓扑设计

ExampleCom 部署团队开始设计目录数据库和服务器拓扑。

examleCom 设计以下数据库拓扑：

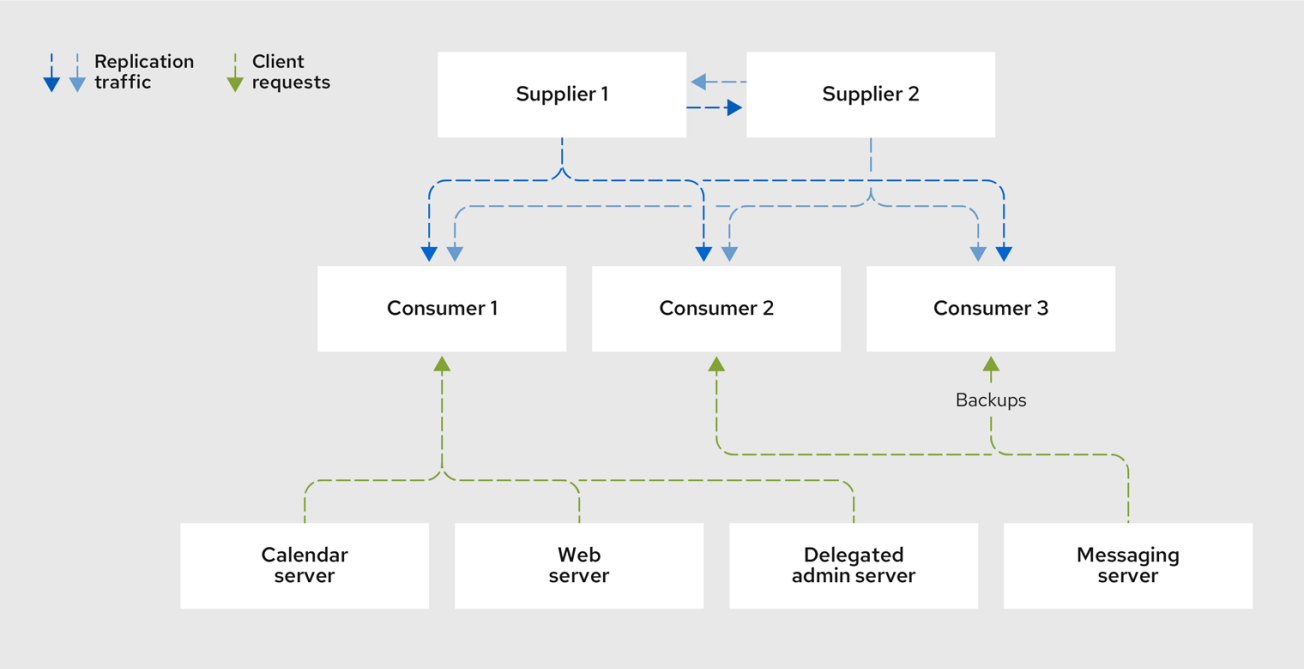
图 8.2. 本地企业数据库拓扑



- 数据库 1 存储 ou=people 分支。
- 数据库 2 存储 ou=groups 分支。
- 数据库 3 存储 ou=resources 和 ou=roles 分支，以及 dc=example,dc=com root 后缀。

examleCom 设计以下服务器拓扑：

图 8.3. 本地企业服务器拓扑



ExampleCom 决定具有两个供应商服务器和三个消费者服务器的服务器拓扑。两个供应商各自更新在目录服务器部署中所有三个用户。

消费者向一个消息传递服务器和其他服务器提供数据。修改来自兼容服务器的请求会路由到适当的消费者服务器。消费者服务器使用智能引用将请求路由到负责修改数据的主副本的供应商服务器。

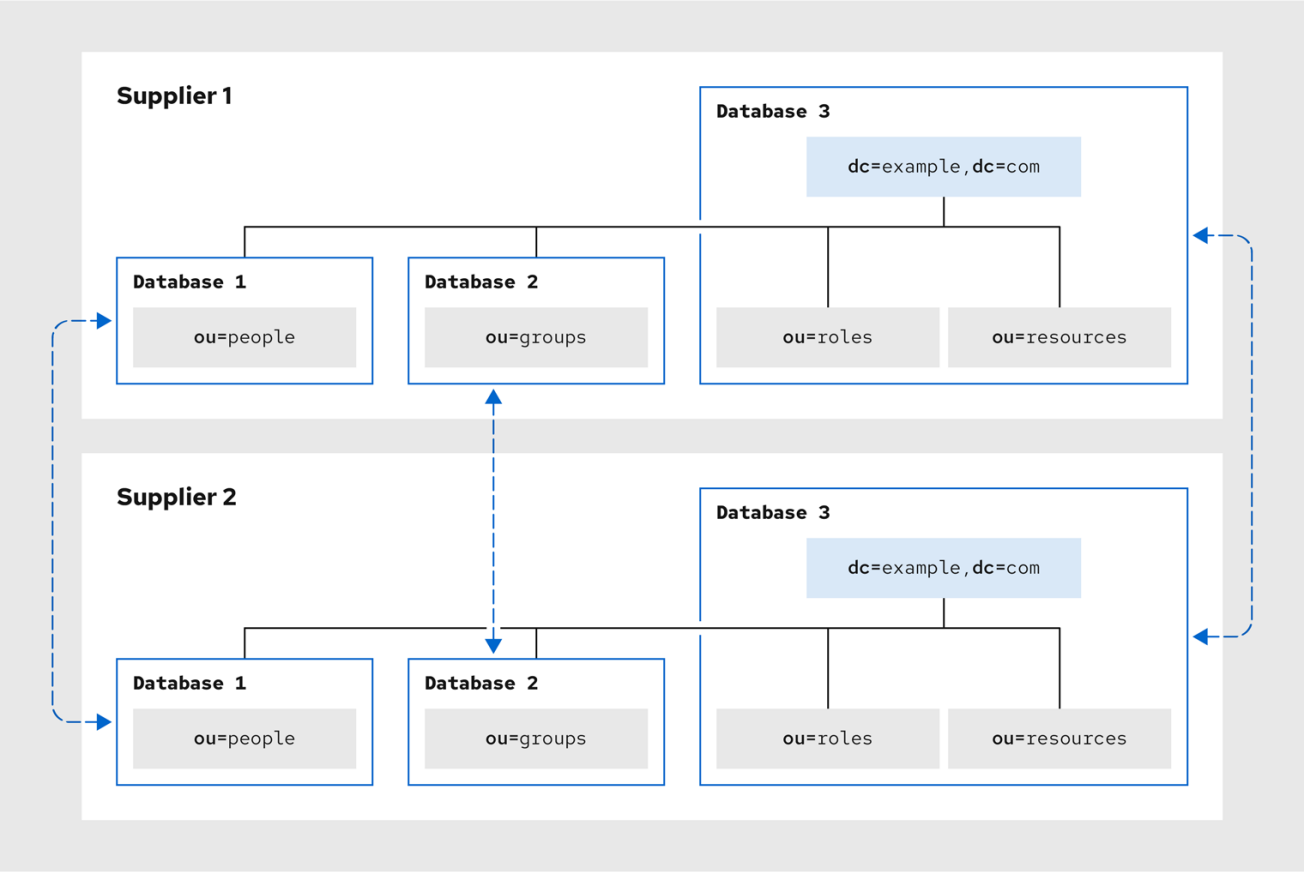
8.1.5. 本地企业复制设计

ExampleCom 决定使用多层次复制设计来确保其目录数据的高可用性。有关多层次复制的更多信息，请参阅 多层次复制。

multi-supplier 架构

ExampleCom 在多层次复制架构中使用两个供应商服务器。供应商更新另一个目录数据，以便目录数据保持一致。下图显示了 ExampleCom 的供应商层次架构。

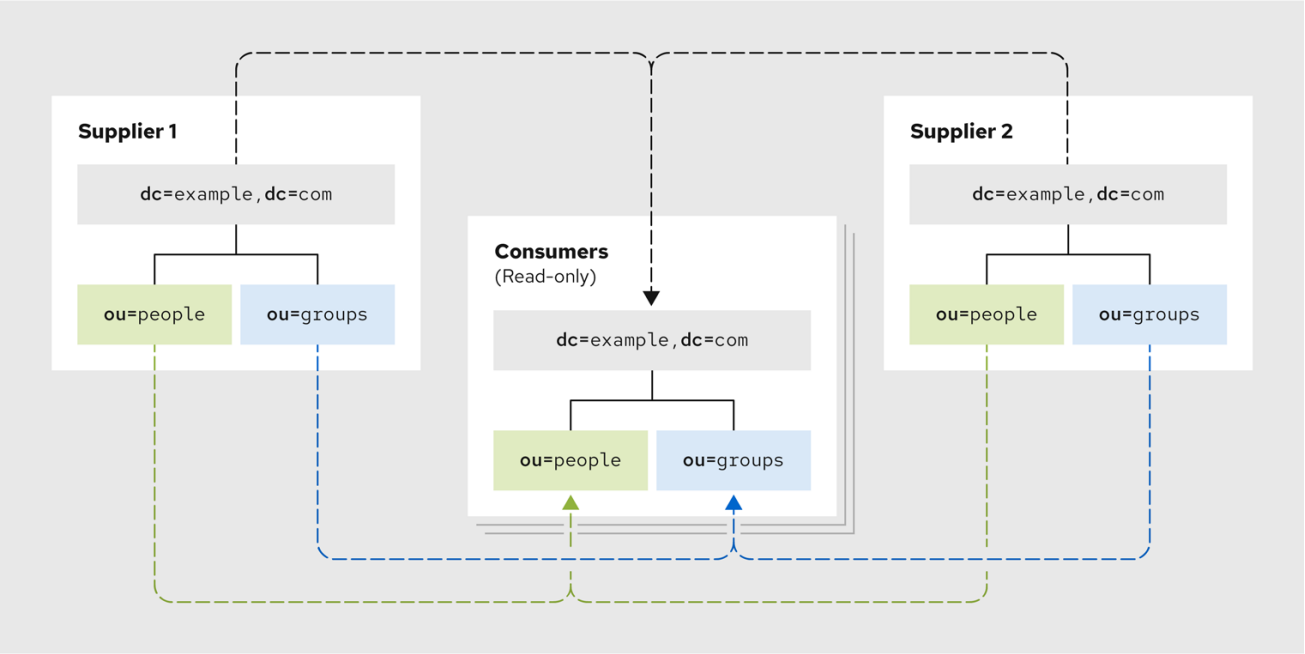
图 8.4. ExampleCom multi-supplier 架构



provider-consumer 架构

下图显示了供应商服务器如何在目录的 ExampleCom 部署中复制到每个消费者。

图 8.5. ExampleCom provider-consumer 架构



两个供应商服务器都会更新三个消费者服务器。这样可确保当其中一个供应商服务器失败时，消费者不会受到影响。

8.1.6. 本地企业安全设计

为了保护目录数据，ExampleCom 会创建以下访问控制指令(ACI)：

- ACI，允许员工修改其条目。用户可以修改除 uid、管理器 和部门属性 之外的所有 属性。
- 一个 ACI，只允许员工和员工经理查看员工主页地址和电话号码来保护员工数据的隐私。
- 目录树的根目录中的 ACI，为两个管理员组授予适当的目录权限：
  - 目录 administrator 组需要对 目录具有完全访问权限。
  - 消息传递管理员组需要 写入 和删除 对 mailRecipient 和 mailGroup 对象类的访问，以及这些对象类允许的属性，包括 mail 属性。ExampleCom 还授予消息传递管理员组、删除，以及向组 子目录添加权限以创建邮件组。

- 目录树根目录下的常规 ACI，允许匿名访问 读取、搜索 和比较 访问。另外，这个 ACI 拒绝匿名用户访问密码信息。

- 一个 ACI，为 accounting 角色授予所有 payroll 信息的访问权限。

另外，ExampleCom 决定以下安全措施：

- 为了防止服务器拒绝服务攻击和不当使用，ExampleCom 根据目录客户端用来绑定的 DN 设置资源限值：

  - 匿名用户可以一次接收 100 个条目，以响应搜索请求。

  - 消息传递管理员可以接收 1,000 个条目。

  - 目录管理员可以收到无限数量的条目。

- ExampleCom 创建一个密码策略，其中密码必须至少为八个字符，并在 90 天后过期。

  有关密码策略的更多信息，请参阅指定密码策略。

8.1.7. 本地企业的操作决策

公司就其目录的日常操作做出以下决策：

- 每晚备份数据库。

- 使用 SNMP 监控服务器状态。

- 自动轮转访问和错误日志。

- 监控错误日志，以确保服务器按预期执行。
- 监控访问日志，以指示可以索引的搜索。

其他资源

- 日志文件引用。

8.2. 跨国企业设计示例

ExampleCom 以前是本地 企业设计示例的小型公司，已发展为分布在三个地理位置的大型组织：美国、欧洲和亚太地区。公司现在拥有 20,000 多名员工，所有员工均在 ExampleCom 办事处所在的国家/地区工作。

ExampleCom 决定启动公司范围内的 LDAP 目录，以改进内部通信，以便更轻松地开发和部署 Web 应用程序并提高安全性和隐私。

在为国际公司设计目录树时，ExampleCom 需要找到以下问题的解决方案：

- 如何逻辑地收集目录条目？
- 如何支持数据管理？
- 如何支持全局扩展上的复制？

此外，ExampleCom 希望创建一个额外的网，即供应商和交易合作伙伴可以使用该 extranet 作为外部客户内部网的扩展。

8.2.1. 跨国企业的数据设计

ExampleCom International 创建一个部署团队来执行站点调查。部署团队决定站点调查中的以下关键点：

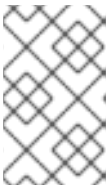
-

消息传递服务器用于为大多数 ExampleCom 站点提供电子邮件路由、交付和读取服务。企业服务器提供文档发布服务。所有服务器都在红帽目录服务器 12 上运行。

- ExampleCom International 需要允许管理员在本地管理数据。例如，欧洲站点负责管理目录的欧洲分支，以及此分支数据的主副本。

- 由于 ExampleCom International offices 的地理分布，用户和应用程序必须每天 24 小时访问目录。

- 特定数据元素的数据值必须采用多种语言。



注意

所有数据都使用 UTF-8 字符集。任何其它字符集都违反了 LDAP 标准。

另外，extranet 的数据设计必须确保满足以下条件：

- 部分供应商需要登录 ExampleCom International 目录来管理其公司的合同。部分供应商依赖于用于身份验证的数据元素，如名称和用户密码。

- 交易合作伙伴将使用 目录查找合作伙伴网络中人员的联系详细信息，如电子邮件地址和电话号码。

8.2.2. 跨国企业的 schema 设计

ExampleCom International 使用其原始模式设计，并添加了两个新的对象类来支持 extranet：

- exampleSupplier 对象类允许 exampleSupplierID 属性。此属性包含 ExampleCom International 分配给每个 automobile 部分供应商的唯一 ID。

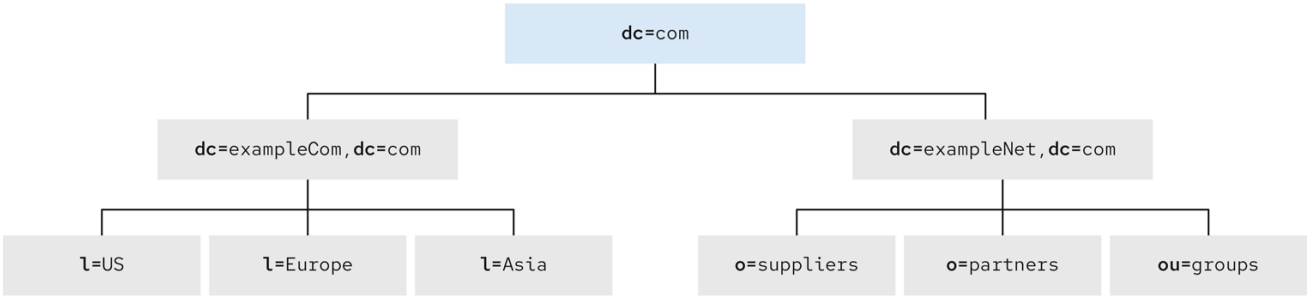
- examplePartner 对象类允许 examplePartnerID 属性。此属性包含 ExampleCom International 分配给每个交易合作伙伴的唯一 ID。

有关自定义默认目录模式的详情，请参考 自定义模式。

8.2.3. 跨国企业的目录树设计

ExampleCom International 创建以下目录树：

图 8.6. ExampleCom International 的基本目录树



611_RHDS_0524

dc=com 后缀是目录树的根目录。在此后缀下，公司会创建以下分支：

- dc=exampleCom,dc=com 分支，其中包含 ExampleCom International 的内部数据。
- dc=exampleNet,dc=com 分支，其中包含 extranet 的数据。

dc=exampleCom,dc=com 下的 I intranet 的目录树有三个主要分支。每个分支对应于 ExampleCom International has offices 的区域之一。这些分支通过使用 l (locality)属性标识。

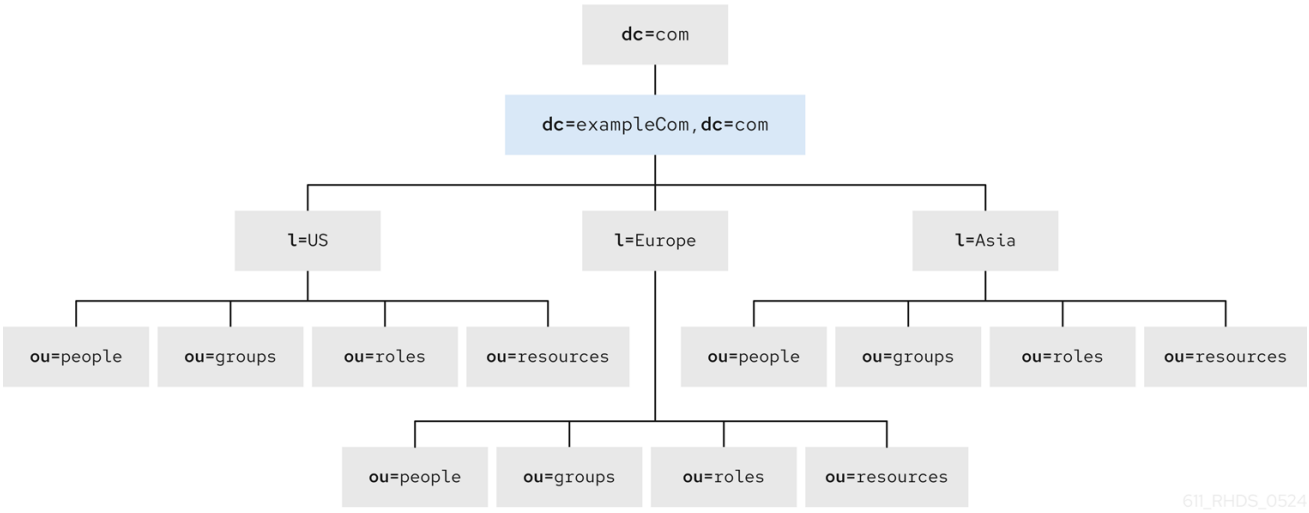
在 dc=exampleNet,dc=com 分支下，ExampleCom International 创建以下分支：

- 公司的供应商 o=suppliers 分支。
- 用于交易 合作伙伴的 o= Partners 分支。
- ou=groups 分支，其中包含 extranet 管理员的条目，以及用于合作伙伴订阅有关自动移动部分制造最新信息的邮件列表。

8.2.3.1. ExampleCom International 的 l intranet 设计

dc=exampleCom,dc=com 下的每个分支都 从本地企业示例的目录树设计 中重复 ExampleCom 的原始目录树设计。

图 8.7. intranet 的目录树示例



在每个地方，ExampleCom International 会创建以下分支点：

- ou=people
- ou=groups
- ou=roles
- ou=resources

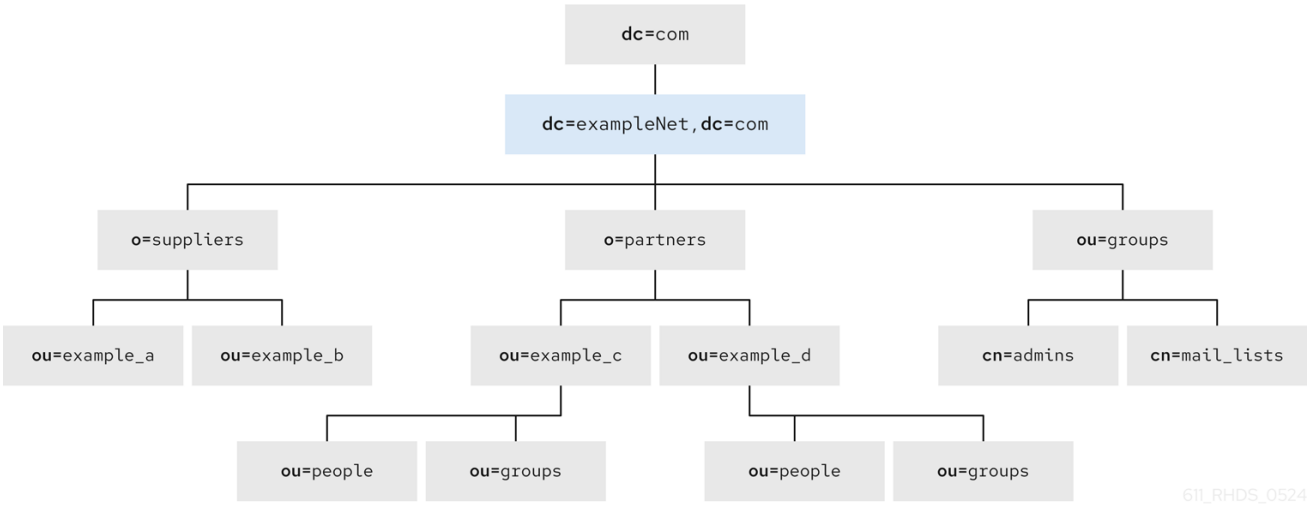
l=Asia locality 的条目会出现在 LDIF 中，如下所示：

```
dn: l=Asia,dc=exampleCom,dc=com
objectclass: top
objectclass: locality
l: Asia
description: includes all sites in Asia
```

8.2.3.2. ExampleCom International 的 extranet 设计

下图显示了 ExampleCom extranet 的目录树：

图 8.8. extranet 的目录树示例



8.2.4. 跨国企业的拓扑设计

ExampleCom International 部署团队开始设计目录数据库和服务器拓扑。

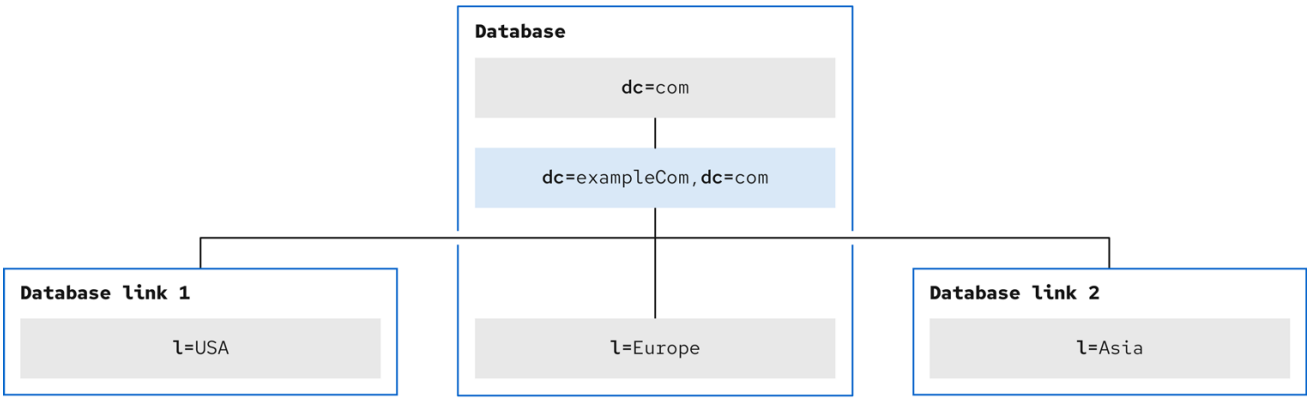
8.2.4.1. ExampleCom International 的数据库拓扑

ExampleCom International 将相同的拓扑设计用于其所有地方。但是，欧洲本地化存储以下分支的数据的主副本：

- dc=com root 条目
- dc=exampleCom,dc=com 下的 I intranet
- dc=exampleNet,dc=com 下的 extranet

下图显示了本地欧洲的数据库拓扑：

图 8.9. ExampleCom Europe 的数据库拓扑



l=Europe 数据库存储 dc=exampleCom,dc=com 和 dc=com 条目的主要副本。

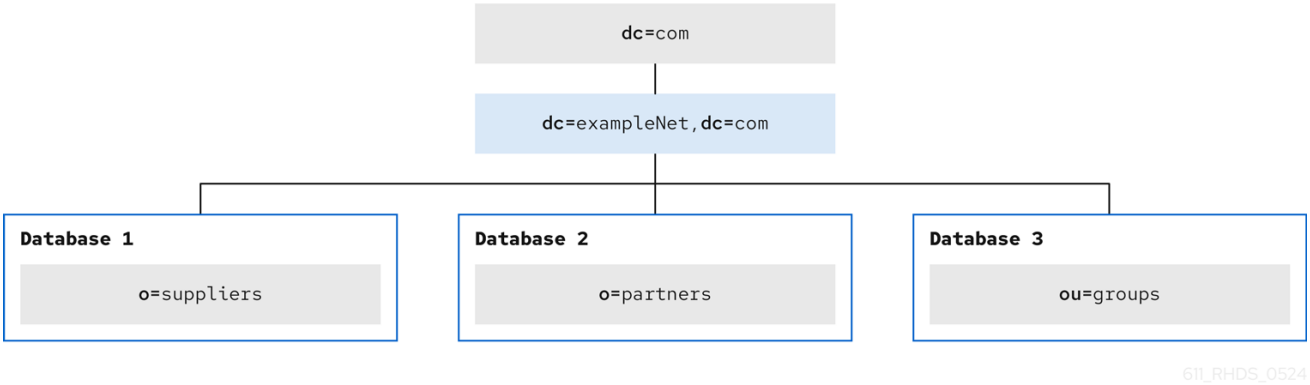
数据库链接 1 和 数据库链接 2 指向存储在每个国家的数据库。例如，在 l=USA 分支下收到的数据的示例Com Europe 服务器收到的操作请求由美国服务器上的数据库链接串联。有关数据库链接和链的更多信息，请参阅使用链。

欧洲服务器包含 extranet 数据的主要副本。extranet 数据以以下方式存储三个数据库：

- *数据库 1 存储 o=suppliers 分支的主要副本。*
- *数据库 2 存储 o= partners 分支的主要副本。*
- *数据库 3 存储 ou=groups 分支的主要副本。*

下图显示了 extranet 的数据库拓扑：

图 8.10. ExampleCom International Extranet 的数据库拓扑



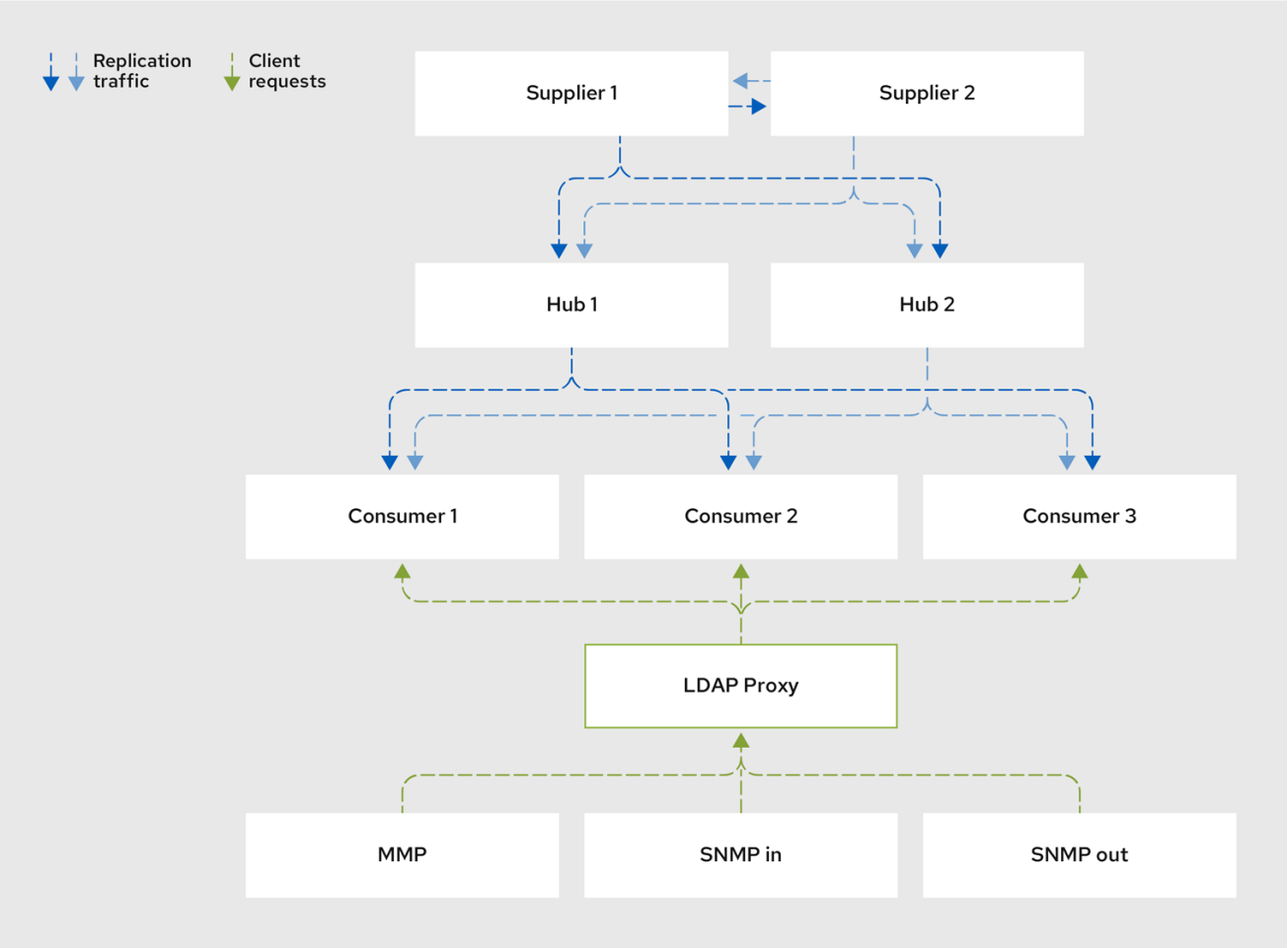
8.2.4.2. ExampleCom International 的服务器拓扑

ExampleCom International 开发以下类型的服务器拓扑：

- *企业内部网的拓扑。ExampleCom 决定拥有三个数据中心，每个主要地点分别对应一个：欧洲、美国和亚太地区。每个数据中心包含以下服务器：*
  - *两个供应商服务器。*
  - *两个 hub 服务器。*
  - *三个消费者服务器。*
- *合作伙伴 extranet 的拓扑。*

下图显示了 ExampleCom Europe 数据中心的架构：

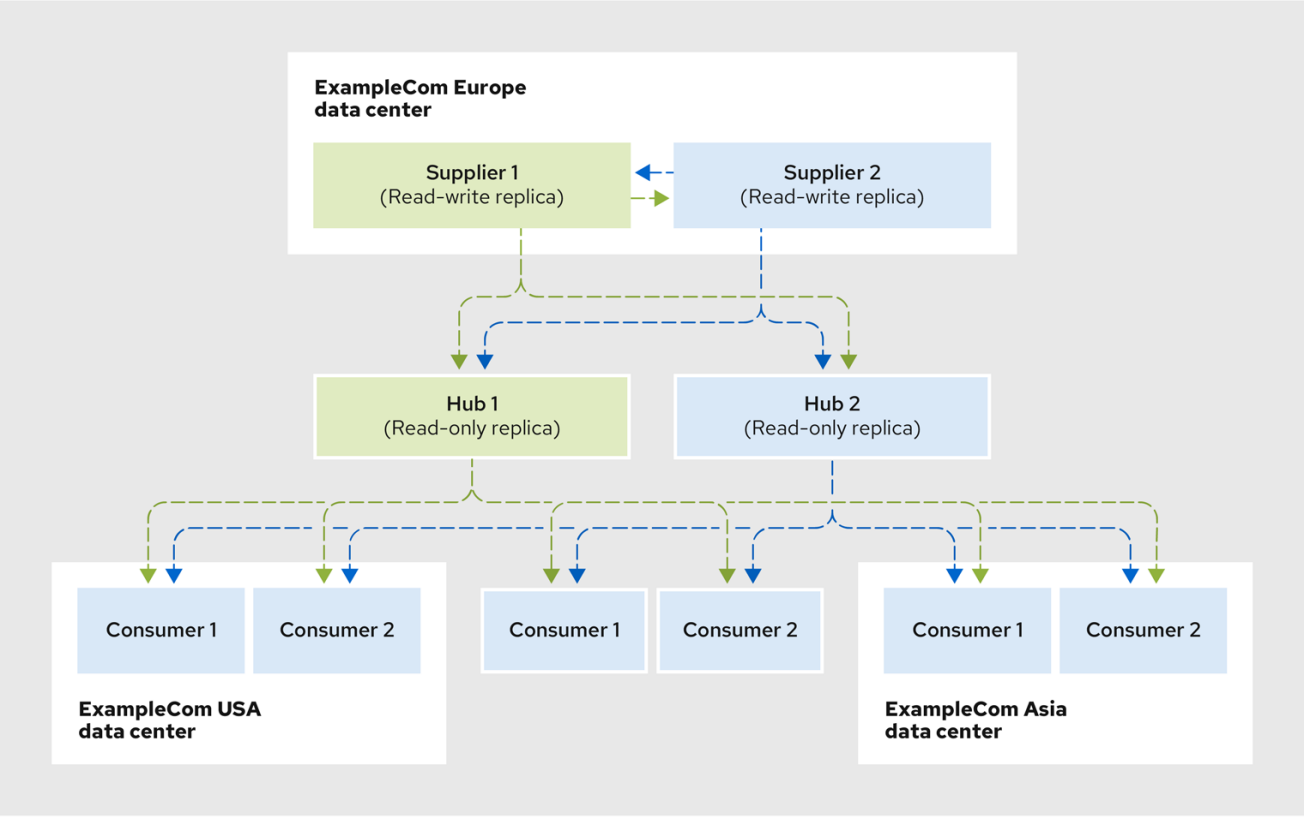
图 8.11. ExampleCom Europe 的服务器拓扑



欧洲数据中心包含 ExampleCom extranet 的主要副本。此数据复制到美国数据中心的两个消费者服务器，以及 Asia 数据中心中的两个消费者服务器。总体而言，ExampleCom 需要十个服务器来支持 extranet。

下图显示了 Europe 数据中心中 ExampleCom extranet 的服务器架构：

图 8.12. ExampleCom International extranet 的服务器拓扑



611_RHDS_0524

hub 服务器将数据复制到每个数据中心的两个消费者服务器中：欧洲、美国和 Asia。

8.2.5. 跨国企业复制设计

在为目录设计复制时，Com International 会考虑以下点：

- *数据由本地管理。*
- *网络连接的质量因站点到站点而异。*
- *数据库链接用于连接远程服务器上的数据。*
- *包含数据的只读副本的 hub 服务器用于将数据复制到消费者服务器。*

hub 服务器位于启用了目录的本地应用程序，如邮件服务器或 Web 服务器。

要让供应商服务器专注于写操作，只有 hub 服务器执行复制。

当 ExampleCom 扩展并需要添加更多消费者服务器时，额外的消费者不会影响供应商服务器的性能。

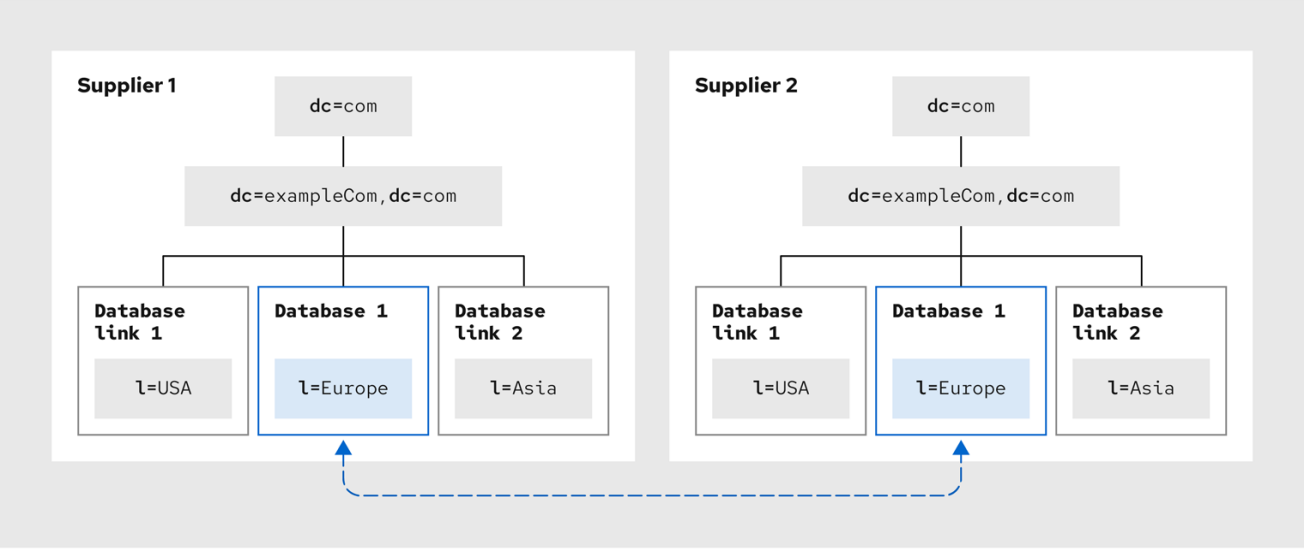
multi-supplier 架构

对于 ExampleCom intranet，每个位置都存储其数据的主副本，并使用数据库链接将数据链接到其他本地实体中的数据。

对于其数据的主要副本，每个本地复制架构都使用多层次复制架构。

下图显示了本地欧洲的多层次架构，其中包括 dc=exampleCom,dc=com 和 dc=com 分支：

图 8.13. ExampleCom Europe 的多层次架构



GII_RHDS_0524

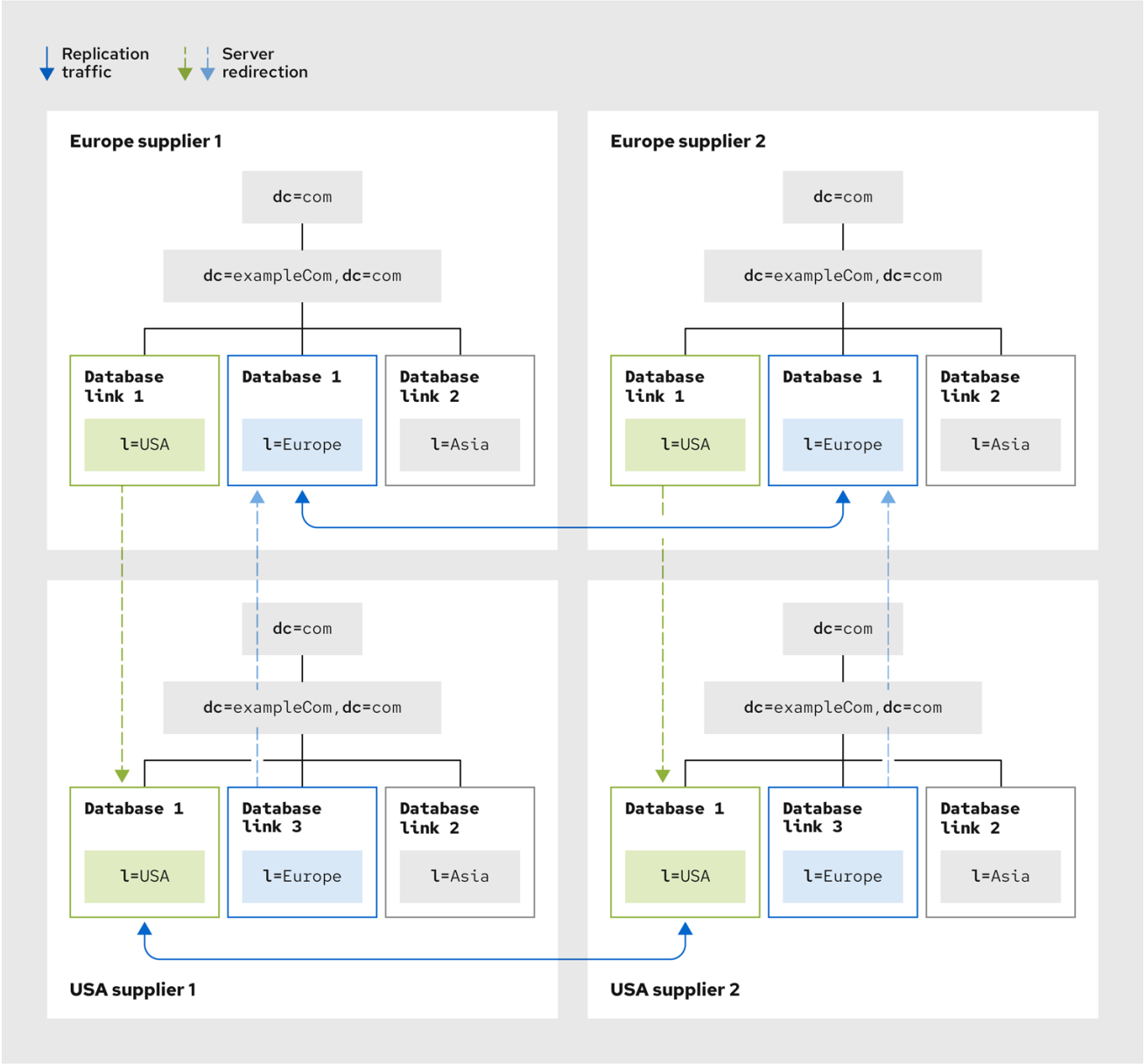
每个地方都包含两个供应商，它们共享该站点的数据的主副本。每个地方都负责其数据的主副本。

使用多层次架构可确保数据的可用性，并帮助平衡由每个供应商服务器管理的工作负载。

要降低总失败的风险，ExampleCom 在每个站点使用多个读写供应商目录服务器。

下图显示了欧洲和美国两个供应商服务器的两个供应商服务器之间的交互：

图 8.14. ExampleCom Europe 和 ExampleCom USA 的 multi-supplier 架构



611_RHDS_0524

ExampleCom USA 和 ExampleCom Asia 之间以及 ExampleCom Europe 和 ExampleCom Asia 之间存在相同的关系。

8.2.6. 跨国企业的安全设计

ExampleCom International 使用以前的安全设计添加以下访问控制来支持其新的跨国：

- ExampleCom 为 intranet 的根目录添加常规 ACI，在每个国家/地区创建更严格的 ACI，以及每个国家下的分支。
- ExampleCom 决定使用宏 ACI 来最小化目录中的 ACI 数量。

  ExampleCom 使用宏代表 ACI 中的目标或绑定规则部分的 DN。当目录获得传入的 LDAP 操作时，ACI 宏与 LDAP 操作目标的资源匹配。如果发生匹配项，Directory 服务器会将宏替换为目标资源的 DN 的值。

有关宏 ACI 的更多信息，请参阅使用宏访问控制说明。

ExampleCom 添加以下访问控制来支持其 extranet：

- ExampleCom 决定对所有外网活动使用基于证书的身份验证。登录 extranet 时，用户需要数字证书。目录存储证书。因此，用户可以通过查找保存在 目录中的公钥来发送加密的电子邮件。
- ExampleCom 创建一个 ACI，用于禁止对 extranet 进行匿名访问。这样可防止 extranet 免受拒绝服务攻击。
- ExampleCom 希望更新目录数据，使其仅来自 ExampleCom-hosted 应用。这意味着，使用 extranet 的合作伙伴和供应商只能使用 ExampleCom 提供的工具。通过将 extranet 用户限制为 ExampleCom 首选工具，ExampleCom 管理员可以使用审计日志来跟踪目录的使用，并限制 ExampleCom International 之外的 extranet 用户可能引入的问题类型。

第 9 章 目录服务器 RFC 支持

找到显著支持的与 LDAP 相关的 RFC 列表。请注意，它不是 Directory 服务器支持的 RFC 的完整列表。

9.1. LDAPV3 功能

技术规范图(RFC 4510)

这是一个跟踪文档，不包含要求。

协议(RFC 4511)

支持以下例外：

- RFC 4511 第 4.4.1 节.通知 Disconnection: Directory Server 在此例中终止连接。
- RFC 4511 第 4.5.1.3 节.SearchRequest.derefAliases: LDAP 别名不被支持。
- RFC 4511 第 4.13 节.IntermediateResponse Message

目录信息模型(RFC 4512)

支持以下例外：

- RFC 4512 第 2.4.2.structural Object Classes: Directory Server 支持具有多个结构对象类的条目。
- RFC 4512 第 2.6 节.别名条目
- RFC 4512 第 4.1.2 节.attribute Types：不支持属性类型 COLLECTIVE。
- RFC 4512 第 4.1.4 节.匹配规则使用

- RFC 4512 第 4.1.6 节.DIT 内容规则
- RFC 4512 第 4.1.7 节.DIT 结构规则和名称表单
- RFC 4512 第 5.1.1. 节.altServer

请注意，RFC 4512 允许 LDAP 服务器不支持之前列出的例外。详情请查看 RFC 4512 第 7.1 节。服务器指南。

身份验证方法和安全机制(RFC 4513)

支持。

字符串代表可辨识名称(RFC 4514)

支持。

搜索过滤器的字符串代表(RFC 4515)

支持。

统一资源定位器(RFC 4516)

支持。但是，此 RFC 主要侧重于 LDAP 客户端。

语法和匹配规则(RFC 4517)

支持。例外：

- directoryStringFirstComponentMatch
- integerFirstComponentMatch
- objectIdentifierFirstComponentMatch
- objectIdentifierFirstComponentMatch

- *keywordMatch*
- *wordMatch*

国际化字符串准备(RFC 4518)

支持。

User Applications 的 schema (RFC 4519)

支持。

entryUUID Operational Attribute (RFC 4530)

支持。

内容同步操作(RFC 4533)

支持。

9.2. 身份验证方法

匿名 SASL 机制(RFC 4505)

不支持。请注意，RFC 4512 不需要 ANONYMOUS SASL 机制。但是，Directory 服务器支持 LDAP 匿名绑定。

外部 SASL Mechanism (RFC 4422)

支持。

普通 SASL 机制(RFC 4616)

不支持。请注意，RFC 4512 不需要 PLAIN SASL 机制。但是，Directory 服务器支持 LDAP 匿名绑定。

SecurID SASL Mechanism (RFC 2808)

不支持。但是，如果 Cyrus SASL 插件存在，目录服务器可以使用它。

Kerberos V5 (GSSAPI) SASL Mechanism (RFC 4752)

支持。

CRAM-MD5 SASL Mechanism (RFC 2195)

支持。

Digest-MD5 SASL Mechanism (RFC 2831)

支持。

一次性密码 SASL Mechanism (RFC 2444)

不支持。但是，如果 Cyrus SASL 插件存在，目录服务器可以使用它。

9.3. X.509 证书模式和属性支持

X.509 证书的 LDAP 架构定义 (RFC 4523)

- *属性类型和对象类：支持。*
- *语法：不支持。目录服务器使用二进制和八进制语法。*
- *匹配规则：不支持。*