LDAP Basics, Schema, LDIF Files, and OpenLDAP

A Practical Introduction to Directory Services

This presentation explores the fundamental concepts, structure, and implementation of LDAP-based directory services, providing a comprehensive overview for IT professionals.

Chapter 1

Understanding LDAP Fundamentals

In this section, we'll explore the core concepts that underpin LDAP technology, including its purpose, structure, and common use cases in enterprise environments.

We'll examine how LDAP organises information hierarchically and why this structure is particularly well-suited for directory services.

What is LDAP?

The Lightweight Directory Access Protocol (LDAP) serves as a standardised method for accessing and maintaining distributed directory information services.

Core Characteristics

- Protocol for querying and modifying directory services over TCP/IP
- Optimised for read-heavy operations and hierarchical data storage
- Designed for efficient retrieval of structured information

Common Applications

- Centralised authentication and authorisation services
- Enterprise address books and user directories
- Network resource and service discovery
- Single sign-on (SSO) implementations

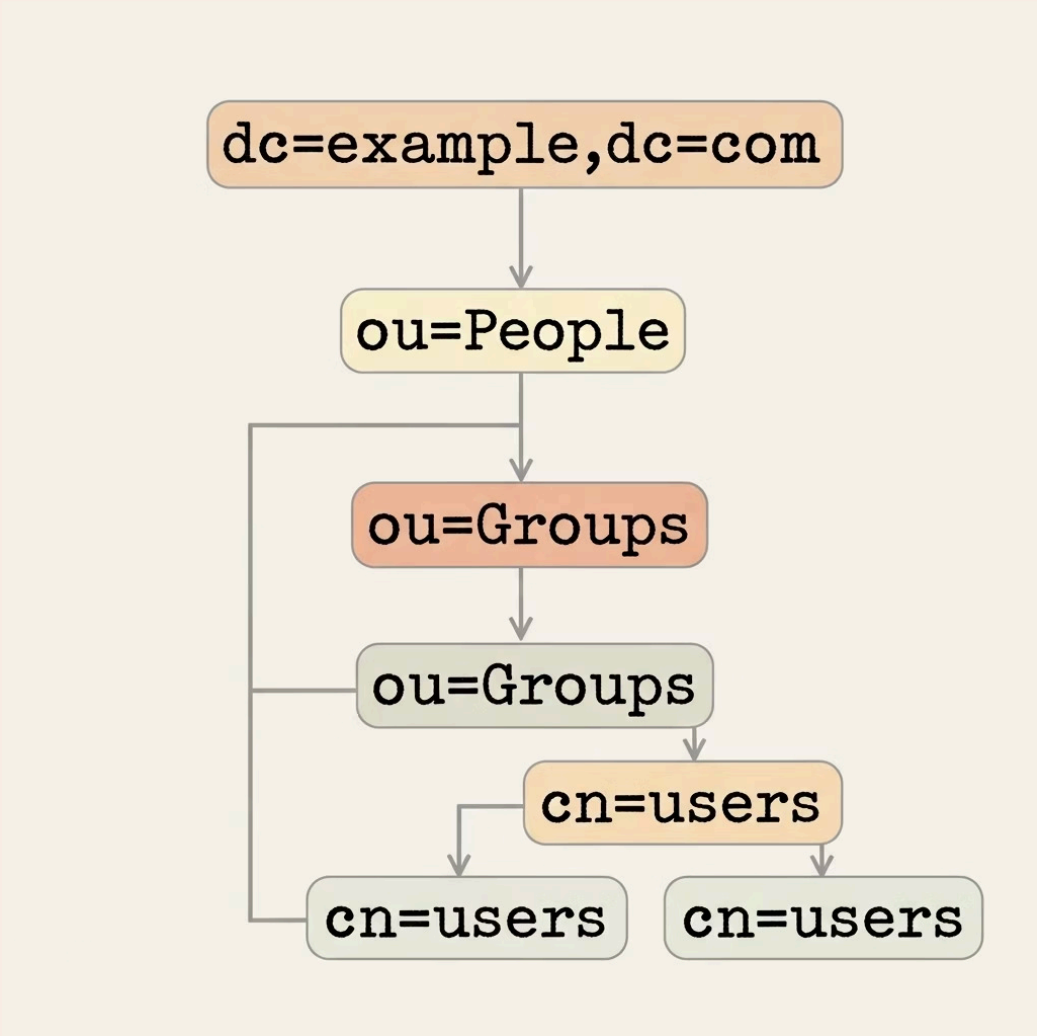
LDAP's lightweight nature makes it particularly suitable for environments where simplicity and performance are priorities, whilst still maintaining robust directory capabilities.

LDAP Directory Structure: The DIT

The Directory Information Tree (DIT) forms the backbone of LDAP's hierarchical structure, organising entries in a tree-like format that mirrors organisational structures.

Key Structural Elements:

- Entries are uniquely identified by Distinguished Names (DNs)
- Each entry contains attributes with corresponding values
- The structure flows from general to specific (top-down)



Example DN: cn=John Smith,ou=People,dc=example,dc=com

Common Attributes

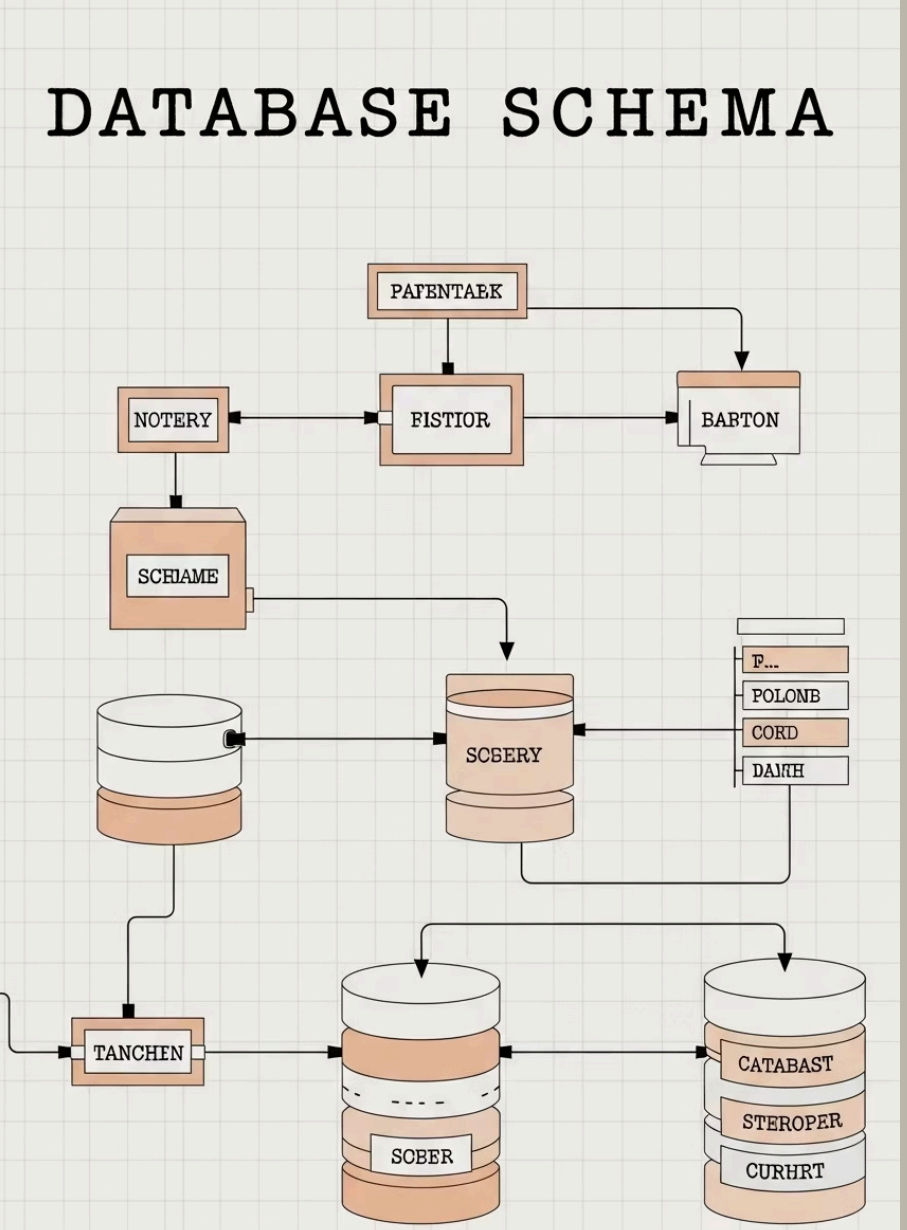
Each entry contains multiple attributes that define its characteristics:

- **cn** (Common Name): The person's full name
- **mail**: Email address
- **uid**: Unique identifier

Entry Types

Different types of entries represent various entities:

- **dc** (Domain Component): Used for DNS-style naming
- **ou** (Organisational Unit): Organises entries into groups
- **cn/uid**: Typically represents individual objects



Chapter 2

LDAP Schema Essentials

The schema defines the rules and structure of your LDAP directory, determining what types of entries can exist and what attributes they can contain.

Understanding schema design is crucial for building effective, maintainable directory services that serve your organisation's needs.

What is an LDAP Schema?

An LDAP schema functions as a blueprint that enforces data integrity by defining permissible structures and attributes within the directory.

1

ObjectClasses

Templates that define what attributes an entry can or must have:

- **Structural**: Primary class of an entry (e.g., person, organizationalUnit)
- **Auxiliary**: Additional attributes that can be added to an entry
- **Abstract**: Base classes that other classes inherit from

2

Attributes

Individual data elements with defined formats and constraints:

- **Syntax**: Data type (e.g., DirectoryString, Integer)
- **Matching Rules**: How values are compared
- **Single/Multi-valued**: Whether multiple values are allowed

3

OIDs

Object Identifiers uniquely identify schema elements:

- Globally unique numeric identifiers
- Hierarchical structure (e.g., 2.5.6.6 for person objectClass)
- Required for all schema elements

The schema ensures data consistency and enables applications to interact with directory data in a standardised way.

Schema Components in OpenLDAP

OpenLDAP implements schema components through a combination of standard and custom schema files, all structured to maintain directory integrity.

Core Schema Files:

- **core.ldif**: Basic attribute types and objectClasses
- **cosine.ldif**: COSINE and Internet X.500 schema
- **inetorgperson.ldif**: Internet/intranet person definitions
- **nis.ldif**: Network Information Services schema

Schema Management

- Modern OpenLDAP stores schema as configuration entries in cn=schema,cn=config
- Dynamic configuration allows runtime schema changes without server restart
- Custom schemas can extend functionality for specific applications

Schema Considerations

- Changes require careful planning to prevent data corruption
- Existing entries must remain compliant after schema modifications
- Schema extensions should use your organisation's OID allocation

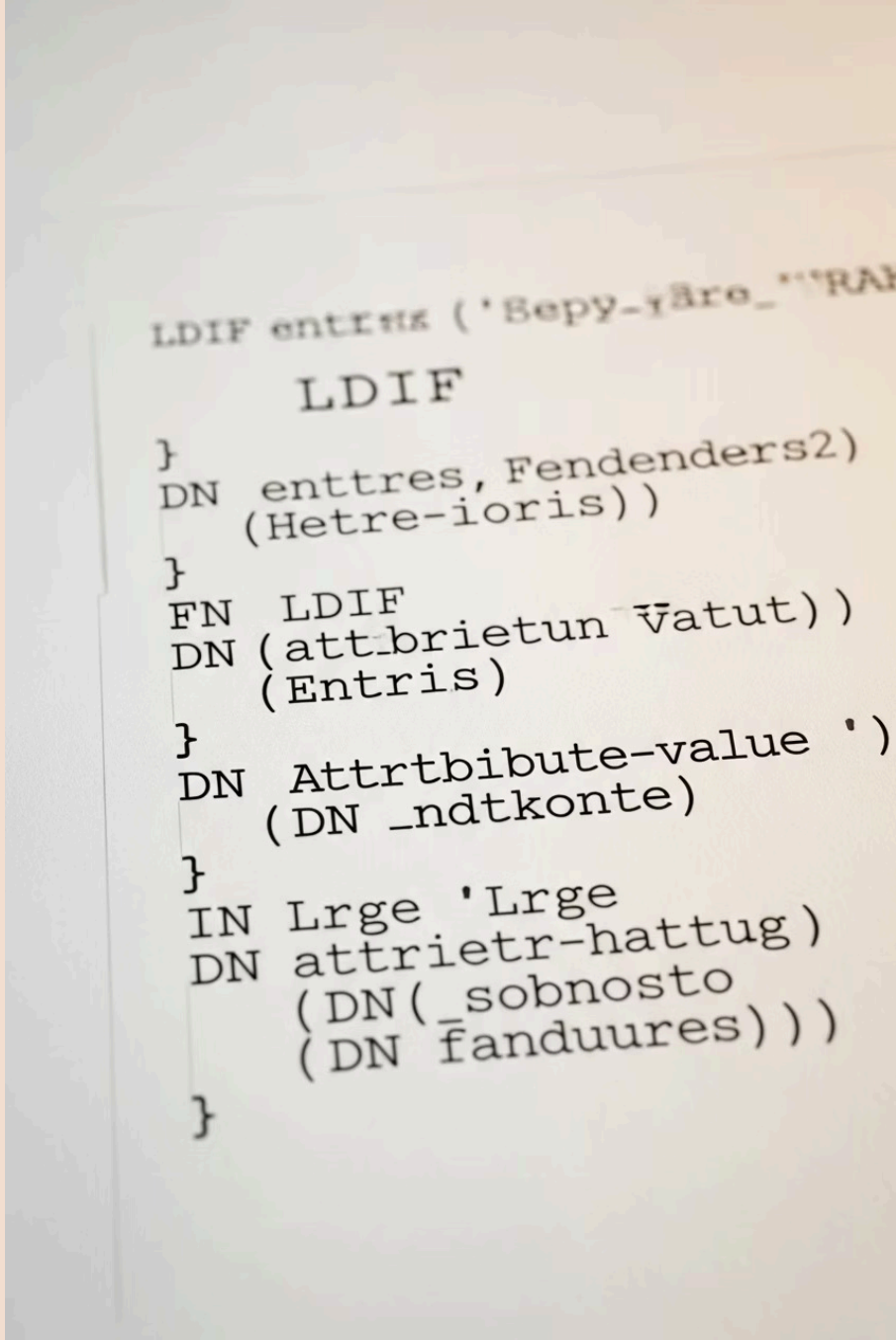
Schema modifications can have far-reaching impacts on directory functionality. Always test changes thoroughly in a non-production environment before implementation.

Chapter 3

LDIF Files – The Language of LDAP Data Exchange

LDIF (LDAP Data Interchange Format) provides a standardised text-based method for representing directory content and operations.

These files are essential for directory management, enabling administrators to perform bulk operations and exchange directory data between systems.



What is LDIF?

LDAP Data Interchange Format (LDIF) is the standard text-based format used to represent LDAP directory entries and directory update operations.

LDIF Structure

- Plain text format with key-value pairs
- Each entry starts with a DN (Distinguished Name)
- Attributes follow with their values
- Blank lines separate individual entries

Common LDIF Operations

- Adding new entries to the directory
- Modifying existing entries
- Deleting entries
- Renaming/moving entries

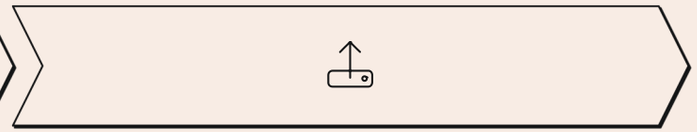
```
# Example LDIF file to add a user
dn: uid=jsmith,ou=People,dc=example,dc=com
objectClass: inetOrgPerson
objectClass: posixAccount
cn: John Smith
sn: Smith
uid: jsmith
uidNumber: 10001
gidNumber: 10001
homeDirectory: /home/jsmith
loginShell: /bin/bash
mail: jsmith@example.com
userPassword: {SSHA}hashed_password_here
```

Create LDIF File

Using a text editor, create a file containing the LDAP entries or modifications

Validate Syntax

Use tools like ldapmodify -Q -c to check for syntax errors before applying



Apply Changes

Use ldapadd or ldapmodify commands to apply the LDIF to the directory

Sample LDIF Entry to Add an Organizational Unit

Common LDIF Operations

1

Add Entry

Create new directory entries

2

Modify Entry

Change attributes of existing entries

3

Delete Entry

Remove entries from the directory

4

Rename/Move

Change DN of existing entries

Adding an Organizational Unit

```
# Create a new Organizational Unit
dn: ou=Engineering,dc=example,dc=com
objectClass: top
objectClass: organizationalUnit
ou: Engineering
description: Engineering Department

# Add a user to the new OU
dn: uid=engineer1,ou=Engineering,dc=example,dc=com
objectClass: inetOrgPerson
objectClass: posixAccount
cn: Lead Engineer
sn: Engineer
uid: engineer1
uidNumber: 10099
gidNumber: 10099
homeDirectory: /home/engineer1
loginShell: /bin/bash
mail: engineer1@example.com
```

This LDIF file first creates an organizational unit called "Engineering" and then adds a user to that unit. The operations would be executed sequentially when processed by OpenLDAP tools.

To import this LDIF file into an OpenLDAP directory, use the command:

```
ldapadd -x -D "cn=admin,dc=example,dc=com" -W -f engineering.ldif
```