

# Recommended readings

---

- [LDAP for Rocket Scientists](#)
- [Basic LDAP Concepts](#)
- [Understanding the LDAP Protocol, Data Hierarchy, and Entry Components](#)

# Openldap server documentation

---

Exercises are based on the OpenLDAP server implementation.

Related material at <http://www.openldap.org>.

# What is LDAP anyway?

---

- Lightweight **D**irectory **A**ccess **P**rotocol
- Vendor independent
- IETF standard:

Clients interact with servers using a directory access protocol

# LDAP Server cli bind

Command	Result
<pre>ldapsearch \ -h localhost ❶ \ -D "cn=admin, dc=bet rayer, dc=com" ❷\ -w password -x ❸\ -b "dc=bet rayer, dc=com" ❹\ -s sub ❺ \ -LLL ❻</pre>	<pre>dn: dc=bet rayer, dc=com ❶ object Class: top object Class: dcObject object Class: organization o: Bet rayers heaven ❷ dc: bet rayer  dn: cn=admin, dc=bet rayer, dc=com ❸ object Class: simpleSecurityObject object Class: organizational Role cn: admin ❹ description: LDAP admini strator userPassword: : e1NT...dE53N1E= ❺</pre>

# Document Information Tree (DIT)

---

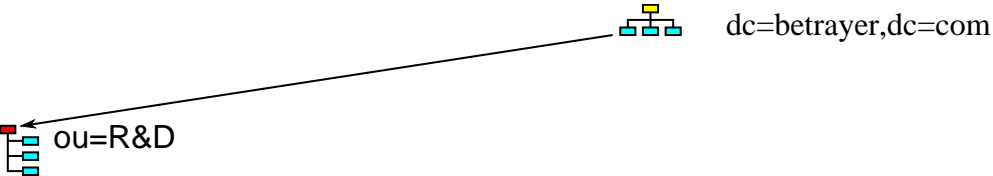


dc=betrayer,dc=com

ou=R&D

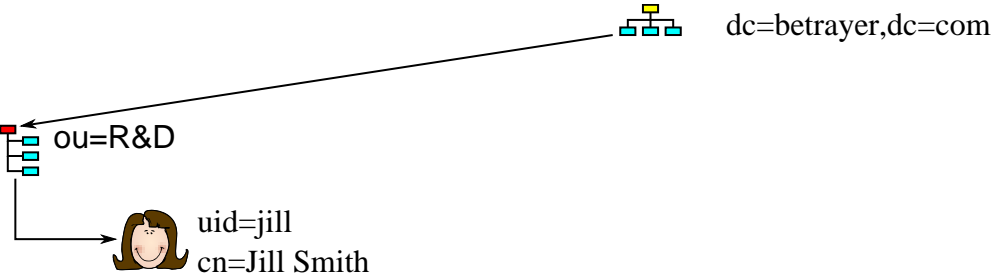
# Document Information Tree (DIT)

---

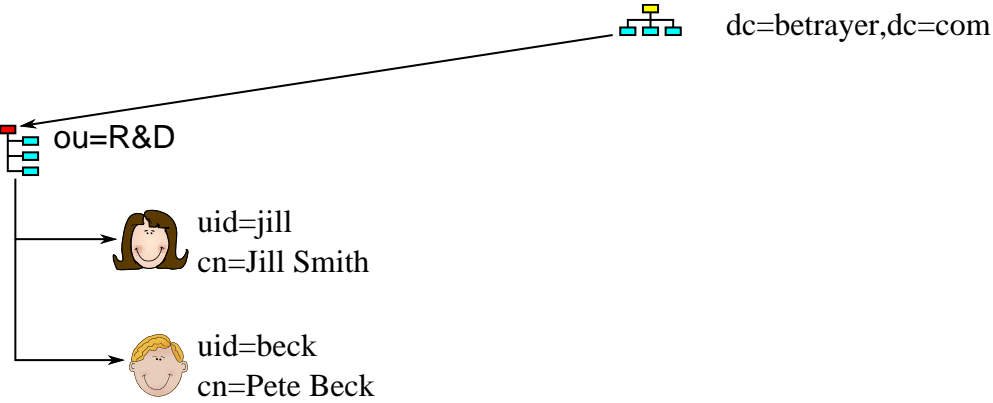


# Document Information Tree (DIT)

---

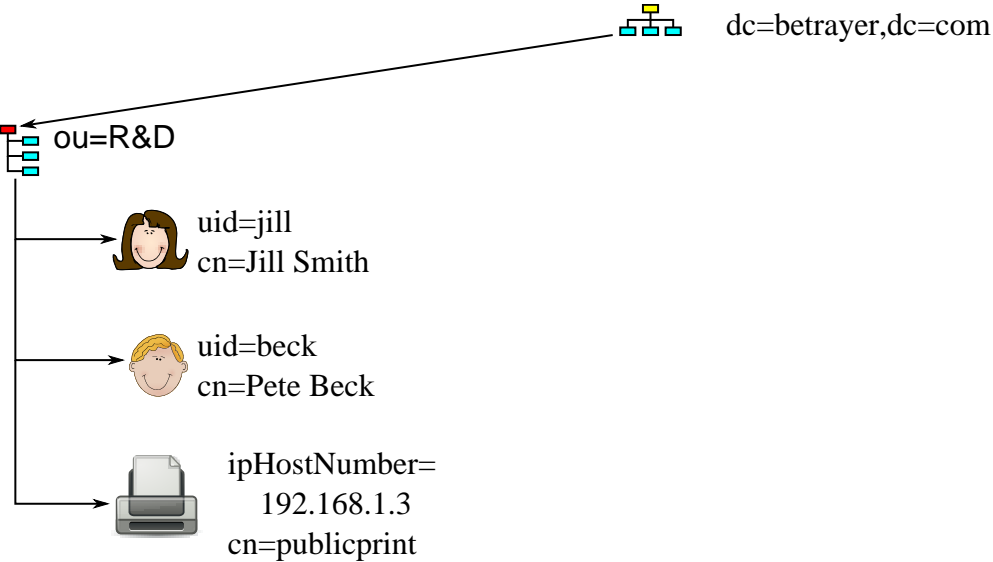


# Document Information Tree (DIT)

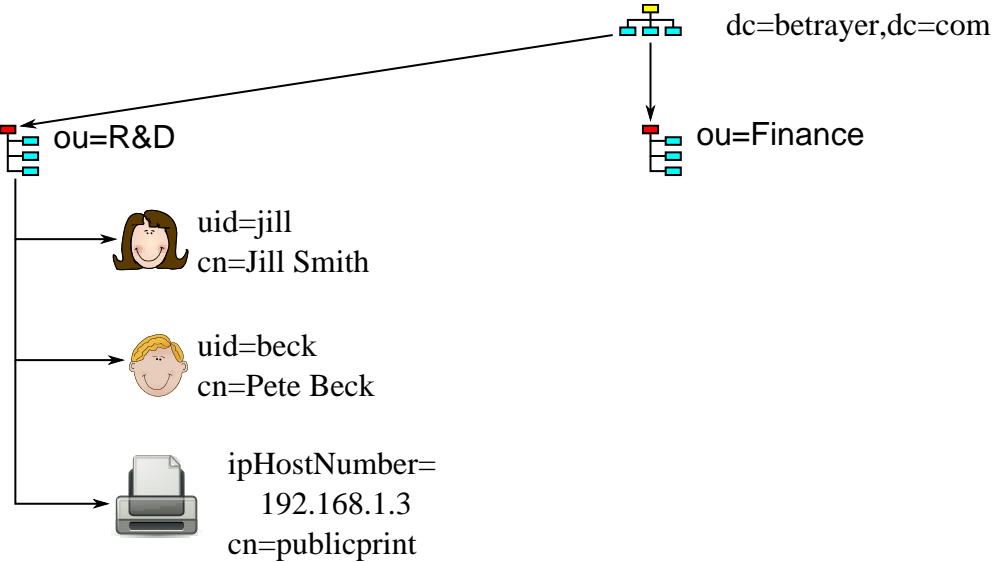




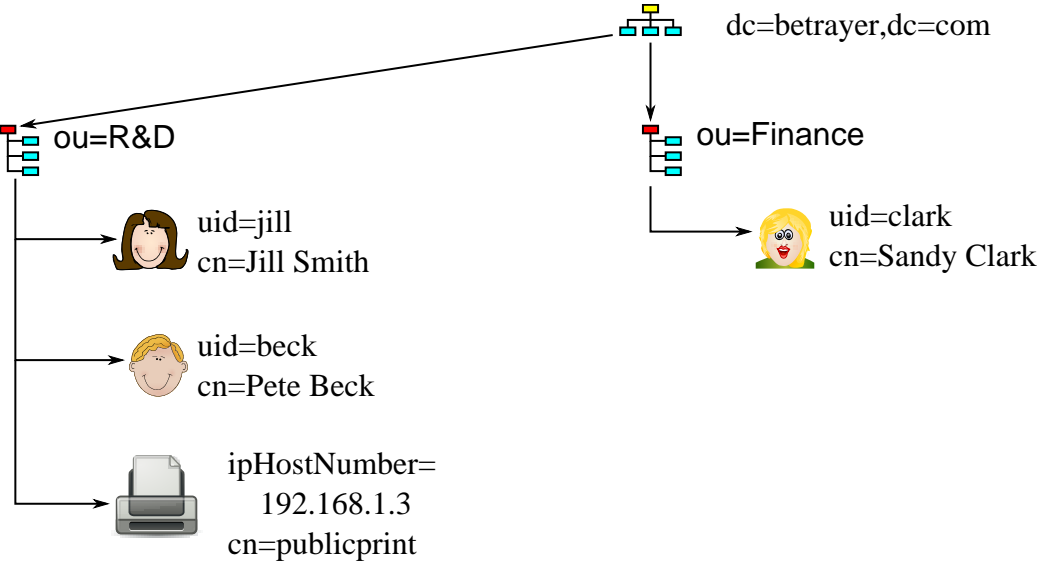
# Document Information Tree (DIT)



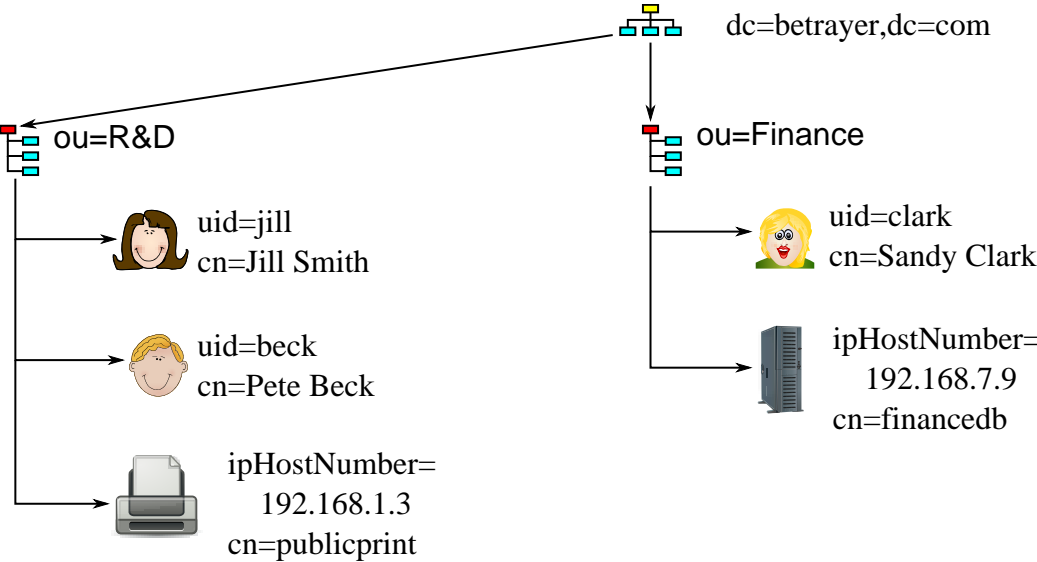
# Document Information Tree (DIT)



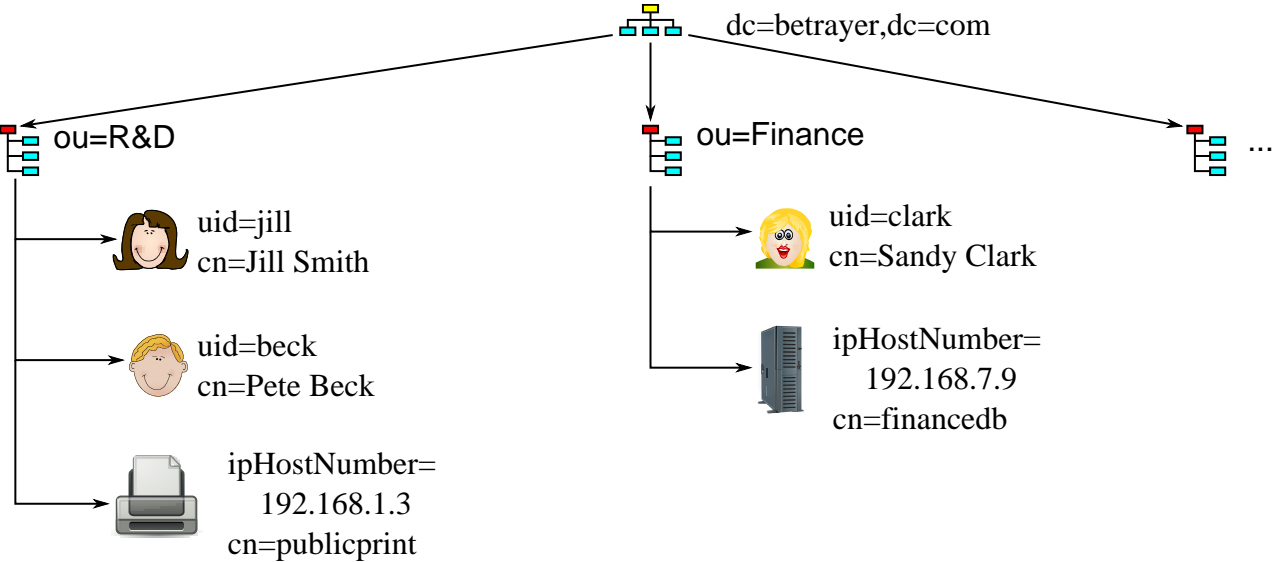
# Document Information Tree (DIT)



# Document Information Tree (DIT)



# Document Information Tree (DIT)



# Relative and absolute DNs

---

 **dc=betrayer,dc=de**

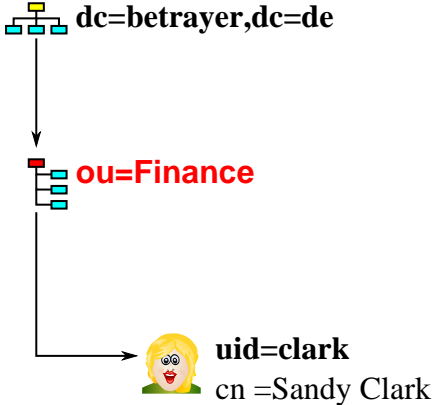
# Relative and absolute DNs

---



# Relative and absolute DNs

---

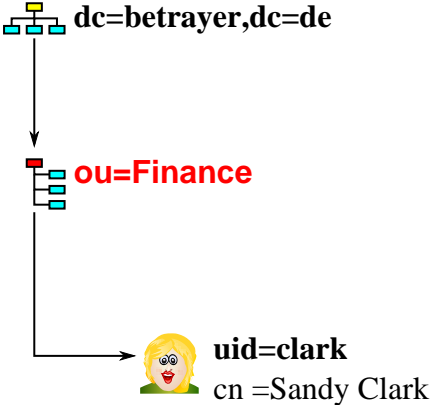




# Relative and absolute DNs

---

## Relative DN values



# Relative and absolute DNs

---

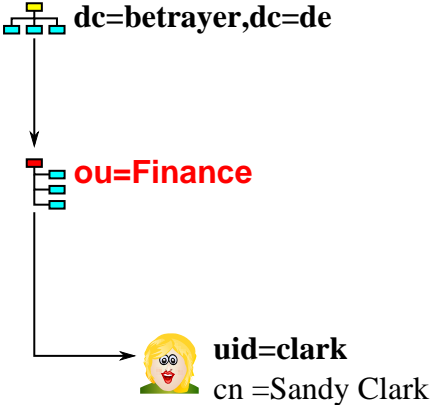
## Absolute DN values

**dc=betrayer,dc=de**

**ou=finance,dc=betrayer,dc=de**

**uid=clark,ou=finance,dc=betrayer,dc=de**

## Relative DN values



# Relative and absolute DNs

## Absolute DN values

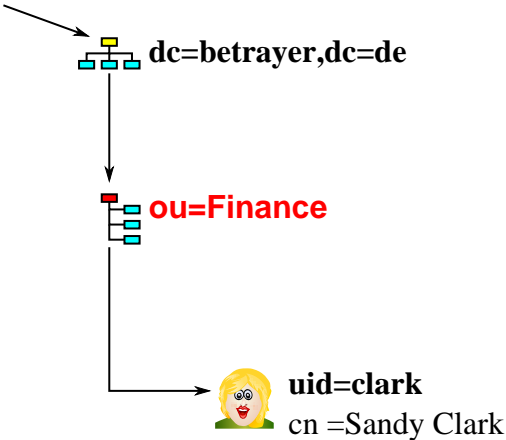
dc=betrayer,dc=de

ou=finance,dc=betrayer,dc=de

uid=clark,ou=finance,dc=betrayer,dc=de

## Relative DN values

Naming  
Context



# User example

---

dn: **ui d=cl ark, ou=fi nance, dc=bet rayer, dc=de** ❶

cn: Sandy Cl ark

homeDirectory: /home/cl ark

sn: Cl ark

**ui d: cl ark** ❷

ui dNumber: 21101

givenName: Sandy

loginShell: /bi n/bash

**mail: cl ark@bet rayer. com** ❸

**mail: fi nance@bet rayer. com**

postOfficeBox: 10G

userPassword: {SSHA} noneOf Your Busi ness

# objectClass

---

- Structuring LDAP entry data.
- Categories:
  - Structural
  - Auxiliary
  - Abstract

# objectClass clarifications

---

- Abstract classes: To be extended by other classes
- Structural classes:
- Each entry requires exactly one.
  - Specify the “main” type of object.
  - Must not inherit from auxiliary classes.

- Auxiliary classes:
- Provide non-conflicting supplementary information.
  - Think of (Java™) interfaces.
  - Must not inherit from structural classes.

# Augmenting inetOrgPerson by posixAccount

Class	Instance uid=clark, ou=finance, dc=betraye, dc=de
inetOrgPerson (structural)	
sn	sn: Clark
cn	cn: Sandy Clark
...	
posixAccount (auxiliary)	
cn	see above ⓘ
gidNumber	gidNumber: 23113
homeDirectory	homeDirectory: /home/clark
uid	uid: clark
uidNumber	uidNumber: 21101
userPassword	userPassword: {SSHA} noneOfYourBusiness
.....	

# Structural objectClass definitions

---

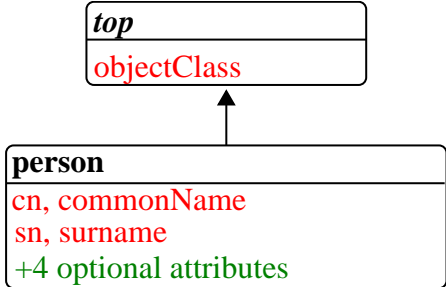
*top*

objectClass



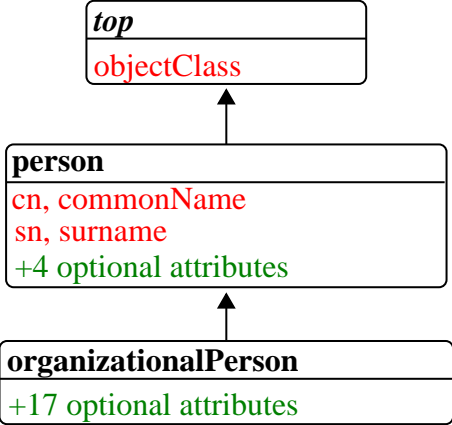
# Structural objectClass definitions

---



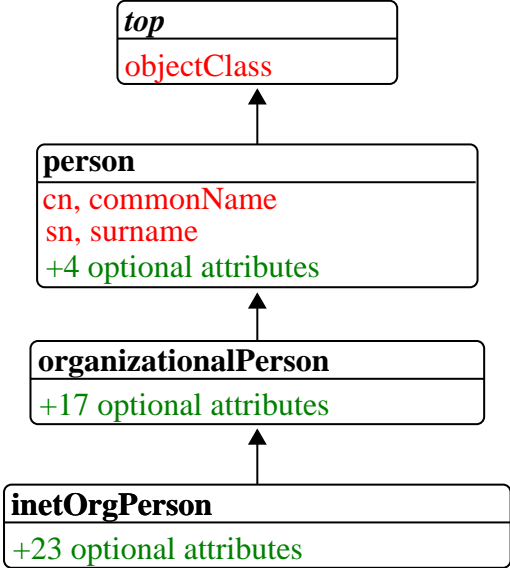
# Structural objectClass definitions

---



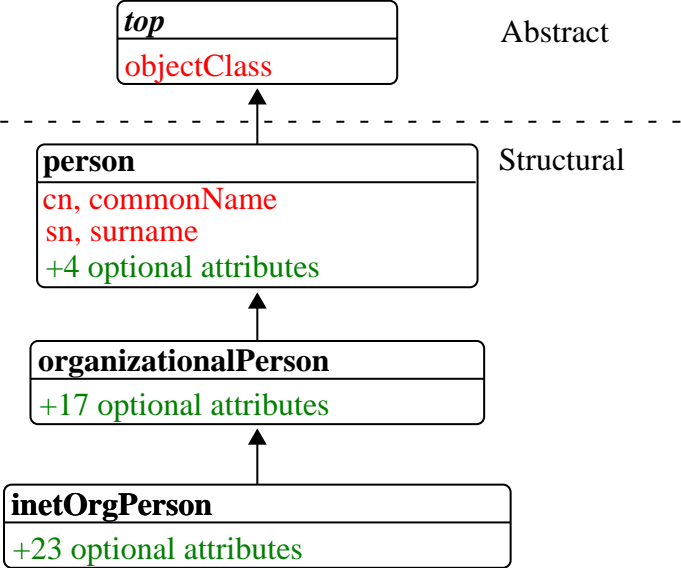
# Structural objectClass definitions

---

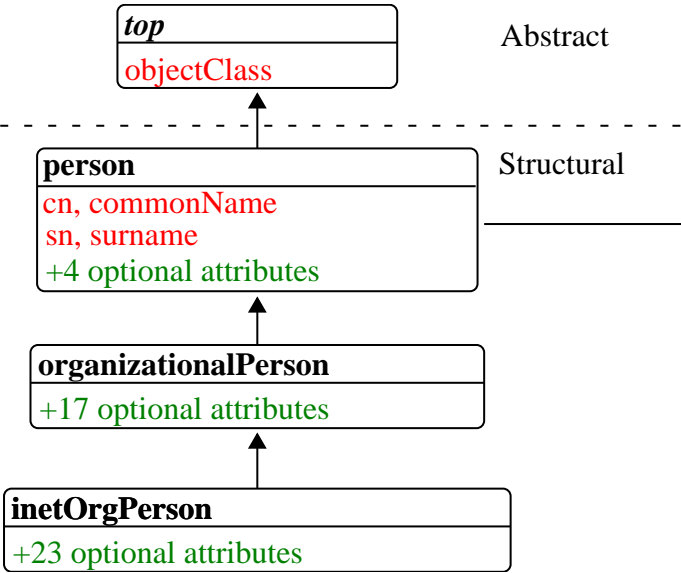


# Structural objectClass definitions

---



# Structural objectClass definitions



Abstract

Relational counterpart sketch:

```
CREATE TABLE person (
  cn VARCHAR NOT NULL,
  sn VARCHAR NOT NULL,
  telephoneNumber
  VARCHAR NULL,
  ...-- +3 more
)
```

# Search scopes

---

RFC 4520 defines three LDAP search scopes:

- baseObject (base)
- singleLevel (one)
- wholeSubtree (sub)

# Predicate based queries

---

RFC 4520 defines predicate based queries using RPN style:

- ( | ( cn=k\* ) ( ui dNumber < 2000 ) )

# LDAP bind types

---

- Anonymous bind: No user credentials.

Note: This typically provides limited privileges.

- Simple bind: User's DN + password:

DN: `uid=clark,ou=finance,dc=betray,dc=de`  
password: `123456789`



# LDIF exchange format

---

- **L**dap **D**ata **I**nterchange **F**ormat.
- Importing and exporting LDAP Data.
- Modifying existing entries (CRUD operations).
- Pure ASCII.

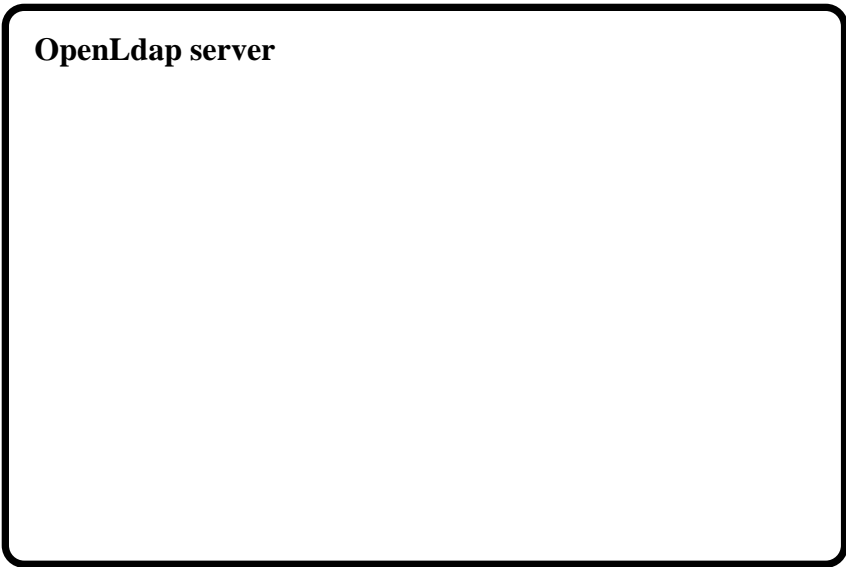
# LDIF sample

---

```
dn: uid=cl ark, ou=fi nance, dc=bet rayer, dc=de
obj ectCl ass: posi xAccount
obj ectCl ass: i net Or gPerson
cn: Sandy Cl ark
homeDi rectory: /home/cl ark
sn: Cl ark
uid: cl ark
uidNumber: 21101
gi venName: Sandy
l ogi nShell: /bi n/bash
nai l: cl ark@bet rayer. com
nai l: fi nance@bet rayer. com
post Of fi ceBox: 10G
user Password: {SSHA} noneOf Your Busi ness
```

# OpenLdap server architecture

---

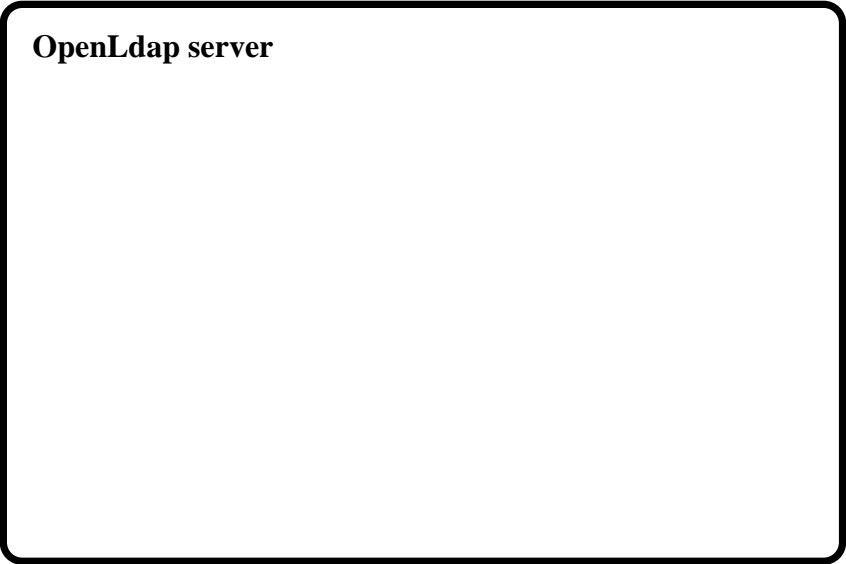


# OpenLdap server architecture

---



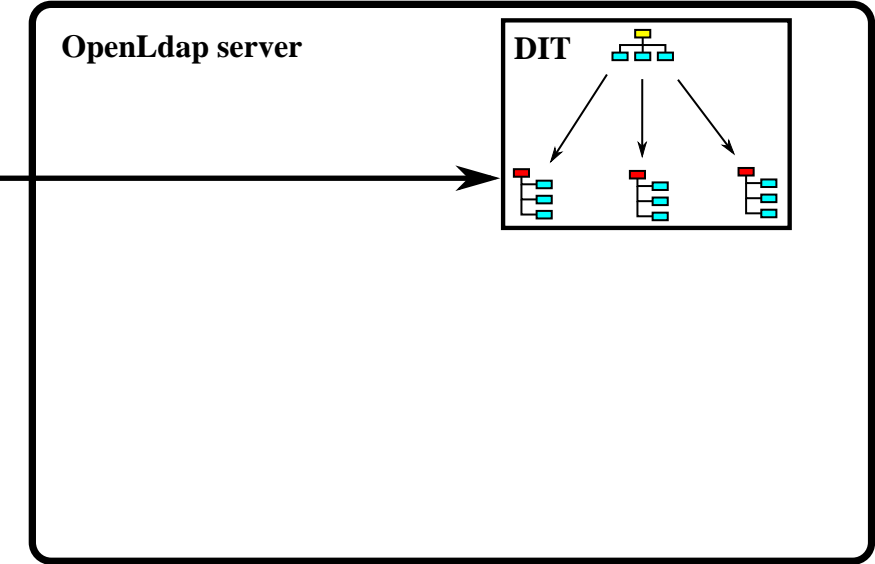
**dc=betrayer,dc=com**



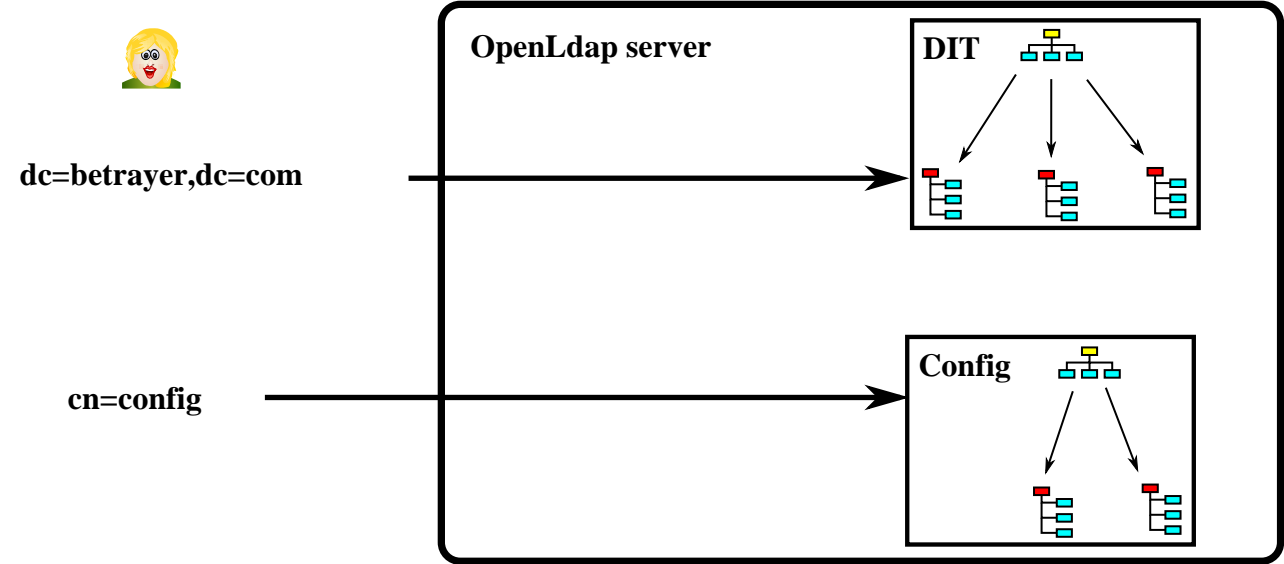
# OpenLdap server architecture



dc=betrayer,dc=com



# OpenLdap server architecture



# An example LDAP Tree

