# Deploying OpenLDAP

John H Terpstra, CTO
PrimaStasys Inc.
jht@primastasys.com & jht@samba.org

# Agenda

- Understand basics of OpenLDAP
- Discover how to deploy it quickly
- Look at ways LDAP can be used
- Overview LDAP utilities
- Know what LDAP servers are in common use

# Assumptions

- Use commercial Linux distribution that includes OpenLDAP
  - We will NOT compile and build OpenLDAP
- You want to get OpenLDAP running quickly
  - Before understanding all the technical details
  - Learn by experimentation
    - Modify working system
    - See what breaks
- When you have learned enough
  - Start with a fresh, new installation
- Expect to start from scratch for final solution

# BASICS

# Basics

- What is LDAP?
- What is OpenLDAP?
- How can one get started quickly?

  Assumptions:

  - It is easier to comprehend what is happening if you can see an example in operation
  - Experience accelerated learning through observation
  - It is easier to modify an working system than to build it from a cold start

# LDAP Defined

- Lightweight Directory Access Protocol
  - Lightweight protocol for accessing directory service
    - X.500-based
    - Runs over connection oriented network protocols
      - TCP/IP
      - Defined in RFC2251
      - Technical specification RFC3377

# Directory Definition

- What is the difference between a directory and a database?
  - A directory is a specialized database optimized for reading, browsing and searching
  - A directory has database and index files
    - Optimized for rapid information retrieval
    - Is not optimized for transaction oriented work, has no roll-back ability
    - Information in a directory is generally of a descriptive nature
- The Internet Domain Name System is an example of a directory

# Use of directories

- Identity management:
  Note: Single-Sign-On requires additional services / utilities
  - Unified back-end for authentication data
    - Mailing system user and list management
    - FTP Server user access control
    - Samba password backend
    - Replacement for NIS/YP database
    - Web access control backend
- Back-end for DNS and DHCP information
- Much more ...

# Directory Terms

- Distinquished Name (DN)

  jht@example.com becomes:

  *dn = uid=jht,dc=example,dc=com*

- Relative Distinguished Name (RDN)

  *uid=jht* is an example of an RDN

- Common Name (CN)

  jht@example.com can also be expressed as:

  dn = cn=jht,dc=example,dc=com

  Could also just contain the User's common name, eg:

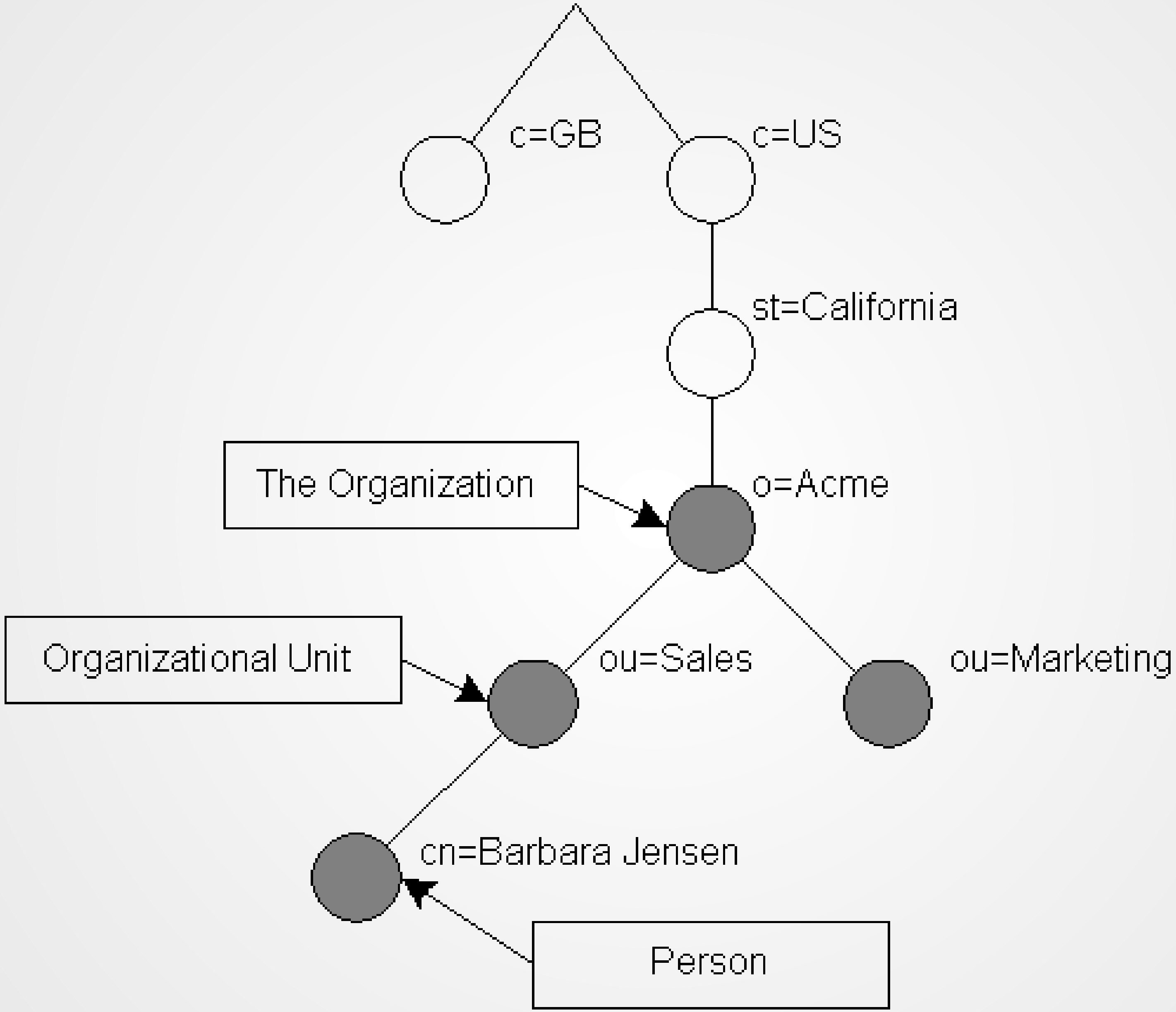
  cn = John H Terpstra

- Many more - discussed later

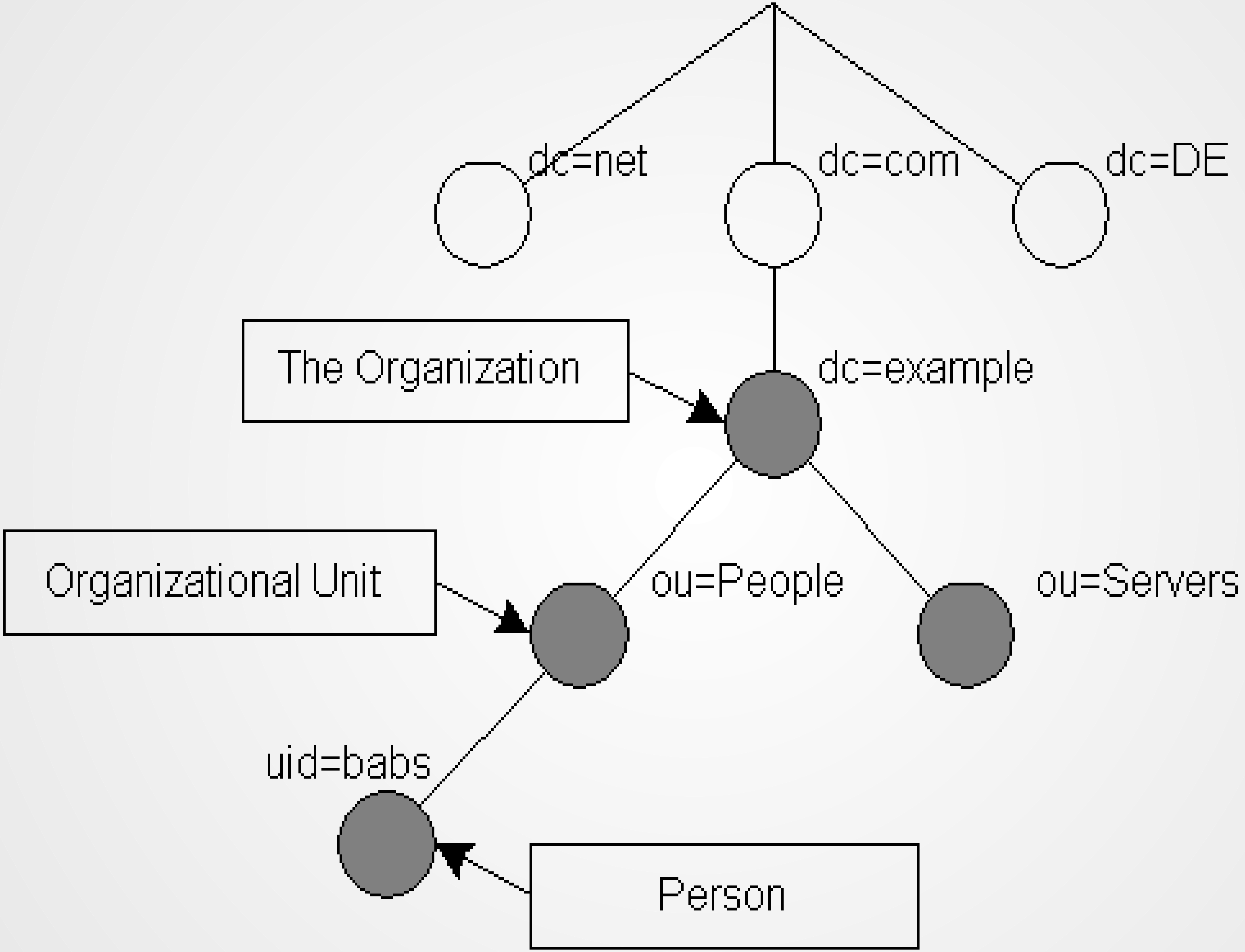
# Directory Arrangement

- A directory should be arranged to suit the purpose for which information is required
  - Consider:
    - What type of information will be retrieved most often
    - Performance requirements
    - Future use
    - Expansion / contraction possibilities
    - Filters that may be needed to locate information
  - Must be optimised
  - Must be appropriately indexed

# Example: Organizational Directory



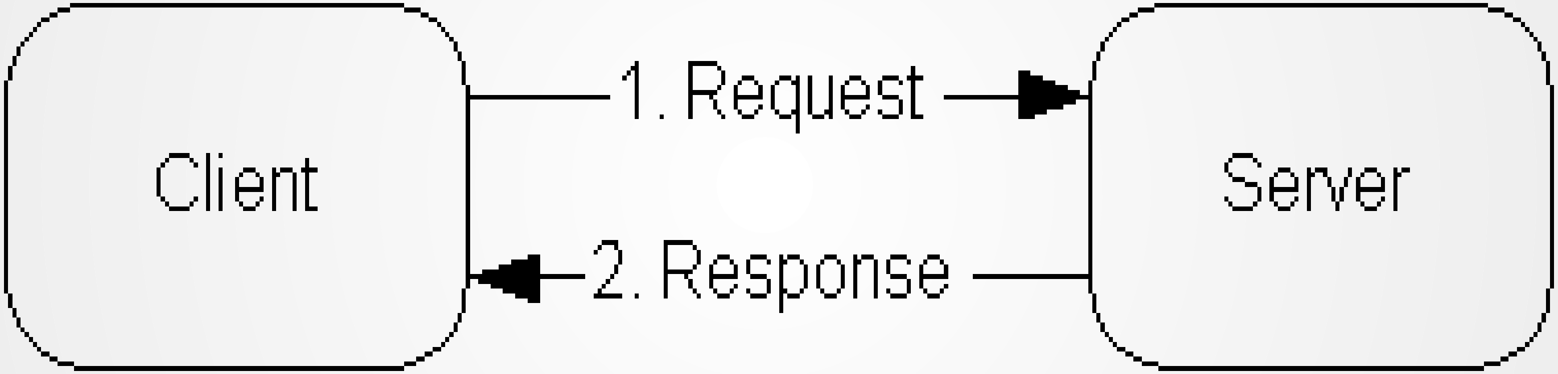
# Example: DNS Type Directory



# Directory Organization

- Singel Master
  - Local Directory
- Multi-Master
  - Local Directory with Referrals
  - Replicated Directory Services
  - Distributed Directory Service
- Multiple Directories may involve:
  - Superior directories
  - Subordinate directories

# Local Directory



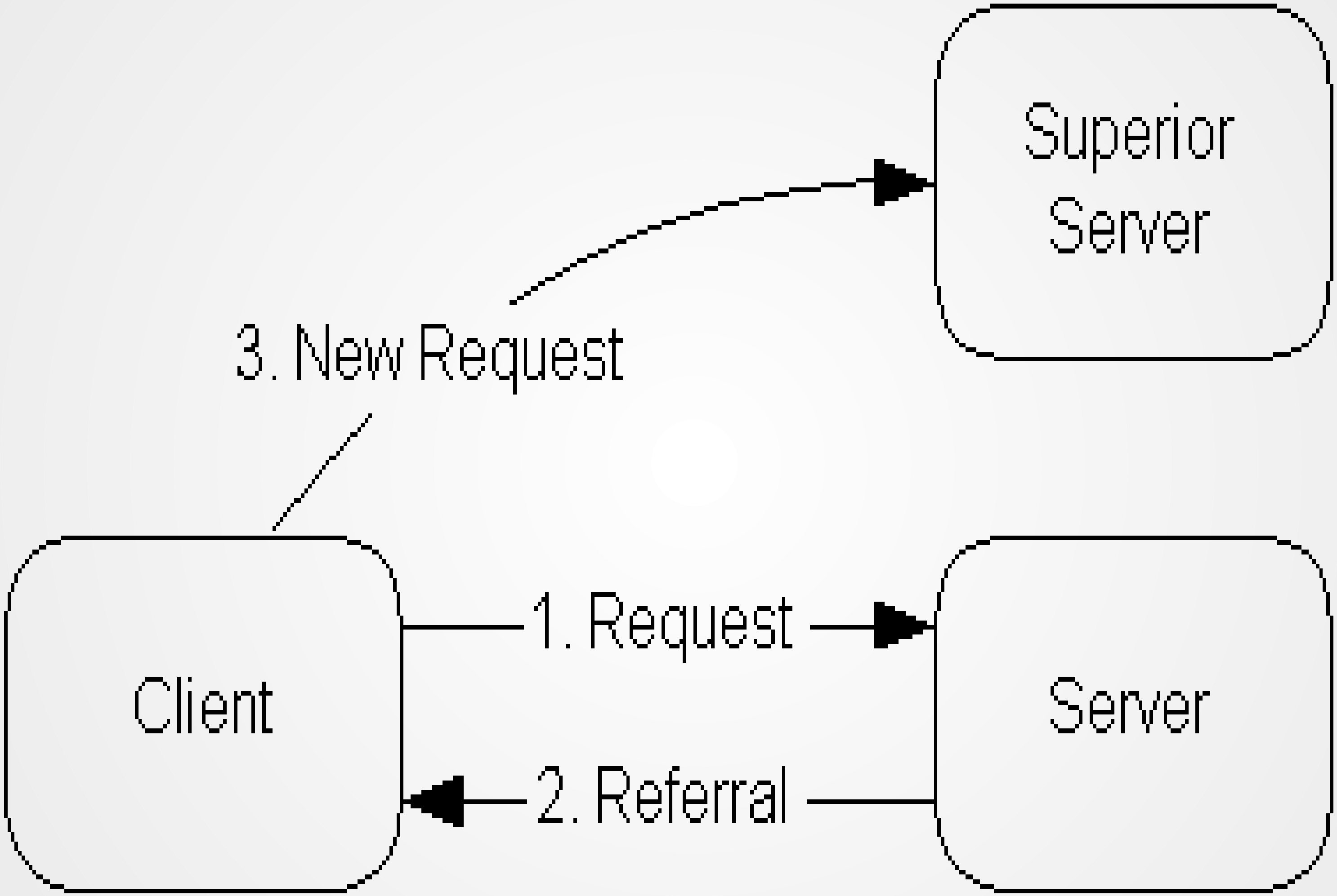
Client

1. Request

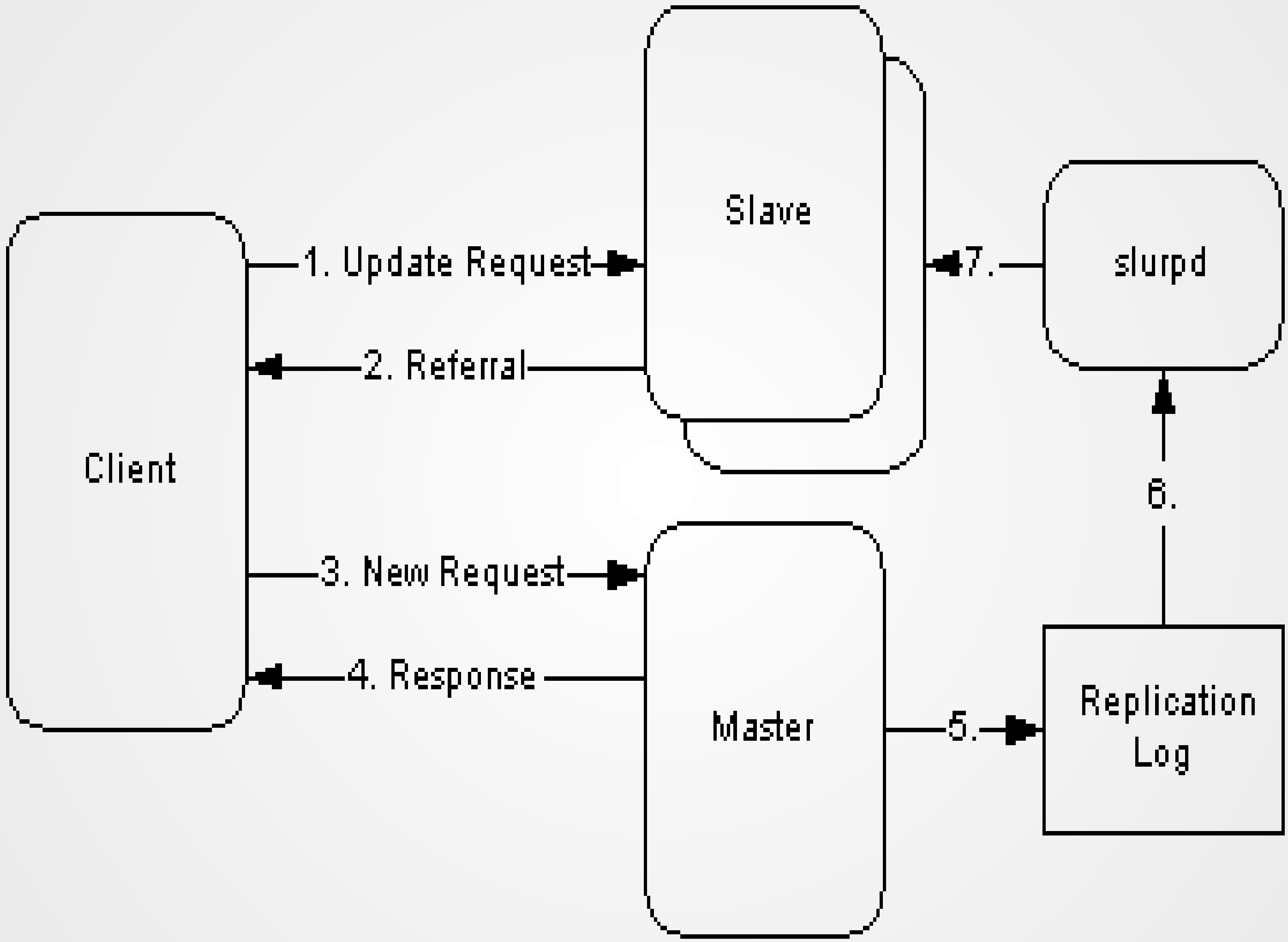
2. Response

Server

# Local Directory with Referrals



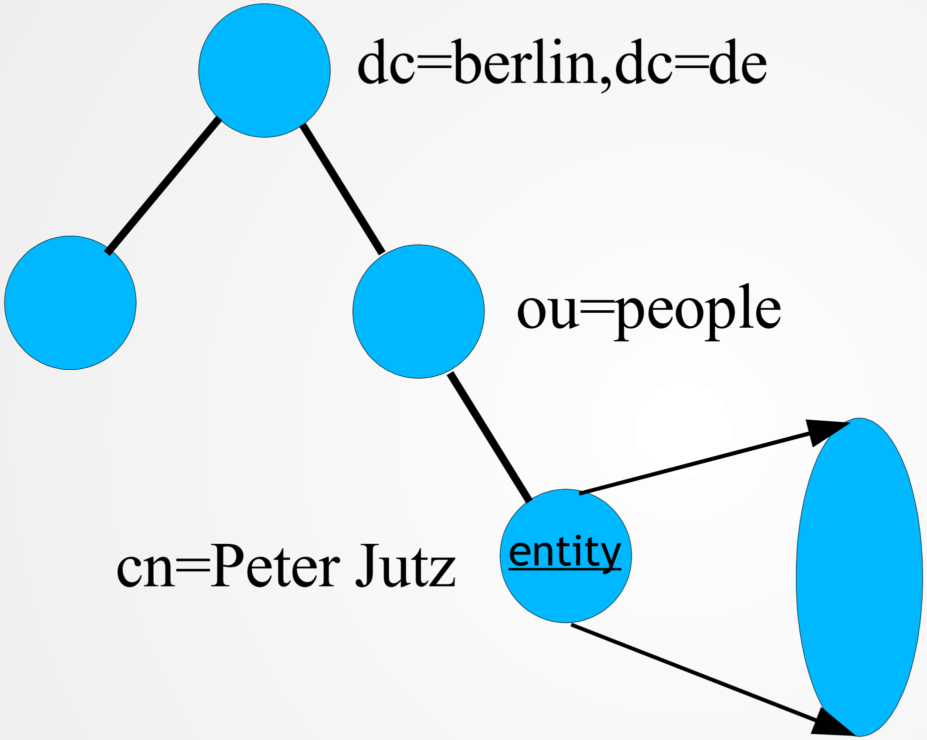
# Replicated Directory



# Distributed Directory

- Uses a mixture of *superior* and *subordinate* directories
  - With a mixture of referrals and replication
- Proprietary vendors invent own terms to describe complex directory structures
  - Microsoft
    - *ADS Forests* - Contain *Multiple Domains*
    - Domains within a forest can be *trusted*
    - *Foreign* Domains can be *trusted*
    - *Foreign* forests can be *trusted*
  - Representation schematics of directories varies

# Description of Directory Data



ObjectClass: Person
cn: Peter Jutz
telephoneNumber: 0162 323 6123
address: Ausfaht 10
...

# Information Storage

- The full DN format is described in RFC2253

  "Lightweight Directory Access Protocol (v3): UTF-8 String Representation of Distinguished Names."

- Common Terms:

  dn, cn, dc, etc. are defined in the directory *schema* in *ObjectClasses* and are subject to *rules* that specify the type of data that may be stored

- There is a selection of common schemas
- Part of the configuration involves design of the directory
  - schemas, access controls, action controls, etc.

# Common Schema Files

| *Purpose* | *Schema File* |
|---|---|
| Corba RFC2714 | corba.schema |
| Basic RFC2251-2256 | core.schema |
| X.400 RFC1274 | cosine.schema |
| DHCP Server Schema | dhcp.schema |
| DNS Zone Schema | dnszone.schema |
| ? | dyngroup.schema |
| Ximian Evolution Schema | evolutionperson.schema |
| InetOrgPerson RFC2798 | inetorgperson.schema |
| Java Objects RFC2713 | java.schema |
| Miscellaneous Objects | misc.schema |
| NIS RFC2307 | nis.schema |
| Experimental OpenLDAP Schema | openldap.schema |
| NIS RFC2703bis | rfc2307bis.schema |
| Samba-3 – Caution: Still changing | samba3.schema |
| SuSE OpenExchange Schema | suse-mailserver.schema |
| SuSE YaST2 Schema | yast.schema |

# Break Point - 1

Log onto your computer:

```
cd /etc/openldap/schema
```

Now examine the contents of each file

# Getting Started: Configuration & Initiation

# Getting Started

- Edit */etc/openldap/slapd.conf*
- Pre-requirements
  - Decide what type of directory you want
    - ie: Organizational, Domain Class, etc.
  - What type of information must be stored
    - ie: Schema components needed
  - Security requirements
  - Indexing requirements
- Start *slapd*
- Initialize the Directory

# Telephone Directory

- Attributes
  - Common Name (cn)
  - Last Name (sn)
  - Address (
  - Town / City
  - Post Code
  - Telephone Number
  - email Address

# Use of the OID

- core.schema:

```
attributetype ( 2.5.4.2 NAME 'knowledgeInformation'
        DESC 'RFC2256: knowledge information'
        EQUALITY caseIgnoreMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{32768} )
```

OID = Object Identifier