

# Sistem Terdistribusi

Naming and Directory Service

# Introduction

- System manages a wide collection of **entities**
- How to **identify** them?
- Naming schema:
  - Files: /boot/vmlinuz
  - Process: 123
  - Devices: /dev/hda
  - Users: anton
- Tiga varian
  - Human Oriented Names
  - System Oriented Names
  - Combination between human/system names

# System oriented name

- Represented in *machine readable form*
- 32/64 bit strings
- Structured/unstructured
- Easy to store
  - Computer only
- Not easy to remember
- Hard for humans to use
- Example: **inode (0x00245dad)**

# Human oriented name

- Variable length character strings
- Usually structured
- Easy to remember and distinguish
- Hard for machine to process
- Example:
  - URL (Uniform Resource Locator)
  - URI (Uniform Resource Identifier)
  - URN (Uniform Resource Name)

# The role of names and name services

- Resources are accessed using *identifier* or *reference*
  - An identifier can be stored in **variables** and retrieved from database quickly
  - Identifier includes or can be transformed to an **address** for an **object**
    - E.g. NFS file handle, CORBA remote object reference
  - A *name* is human-readable value (usually a string) that can be *resolved* to an identifier or address
    - Internet domain name, file pathname, process number
    - E.g. `/etc/passwd`, `http://www.cdk4.net/`
- For many purposes, **names are preferable** to identifiers
  - because the binding of the named resource to a physical location is deferred and can be **changed**
  - because they are more **meaningful** to users
- Resource names are *resolved* by name services
  - to give identifiers and other useful attributes

# URI

- Uniform Resource Identifier
- Identify resource on the Web
- Include image, document, video, audio on the web, and protocol
- URL is part of URI
- Disadvantages: **Possible of broken link**
- Example:
  - <http://lecturer.ukdw.ac.id/anton/sister.php>

# Protokol yang didukung oleh URI RFC 1738

ftp	File Transfer Protocol
http	Hypertext Transfer Protocol
gopher	The Gopher Protocol
mailto	Email address
news	USENET news
nnntp	USENET news using NNTP access
telnet	Reference to interactive sessions
wais	Wide-Area Information Servers
file	Host-specific file name
prospero	Prospero Directory Service

- "http://" host [ ":" port] [ "/" path] [ "?" search]
- "ftp://" [user ":" password "@" host] [ ":" port] \*[ "/" directoryname]  
[ "/" filename]

# URN

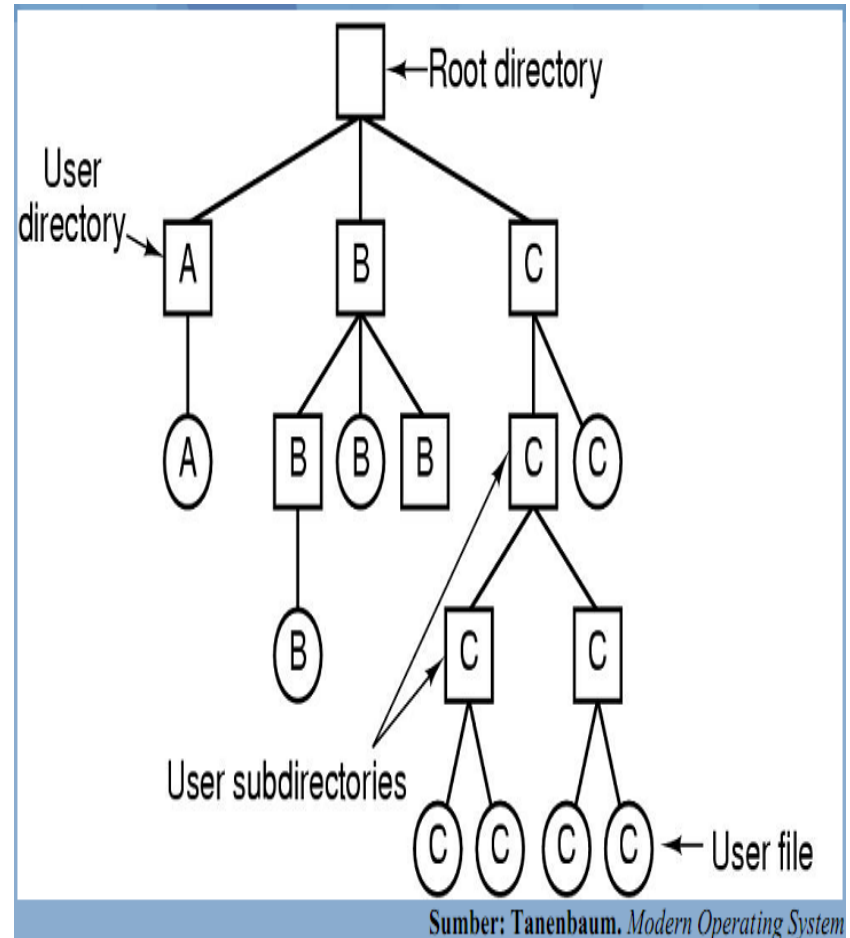
- Provide better approach to find resources on the Web
- Idea: Each resource has persistence URN
- Format: **urn:nameSpace:nameSpace-specificName**
- Example: ISBN
  - urn:ISBN:0-201-62433-8
- *urn:dcqs.qmul.ac.uk:TR2000-56*

# URC

- Uniform Resource Characteristics
- merupakan subset dari URN untuk mendeskripsikan suatu resources Web dengan suatu **atribut**,
- contoh pada META  
'author=anton',  
'keywords=sister,alpro,strukdat',  
'description=lecturer web'

# Penamaan di Linux

- Menggunakan konsep tree
- Tiap direktori dapat mengandung file dan subdirektori
- Path (absolut path) adalah urutan direktori yang berasal dari MFD (master file directory)
- Working dir. (relative path) adalah path yang berasal dari current directory
- Current directory adalah direktori yang baru-baru ini digunakan
- Contoh absolut path : /C/C/C/C



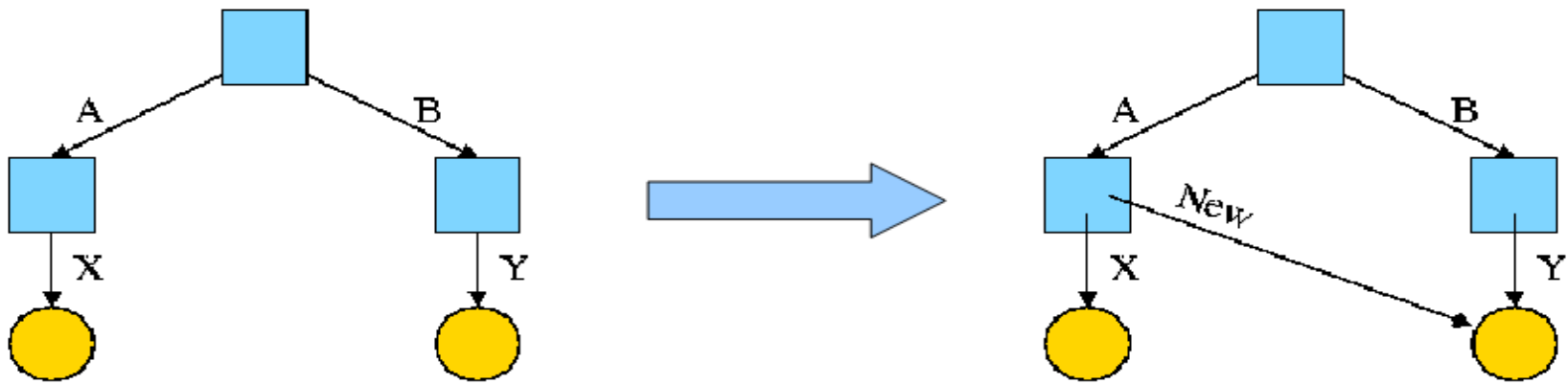
# Contoh Directory di Linux

- "/" direktori root
- /bin perintah binari esensial
- /boot file –file konfigurasi boot loader
- /dev device files
- /etc konfigurasi sistem host-specific
- /lib shared libraries essential dan modul kernel
- /mnt mount point untuk me-mount suatu file system sementara
- /opt tambahan paket aplikasi piranti lunak
- /sbin sistem binari esensial
- /tmp tempat file sementara
- /usr secondary hierarchy of system
- /var data variabel

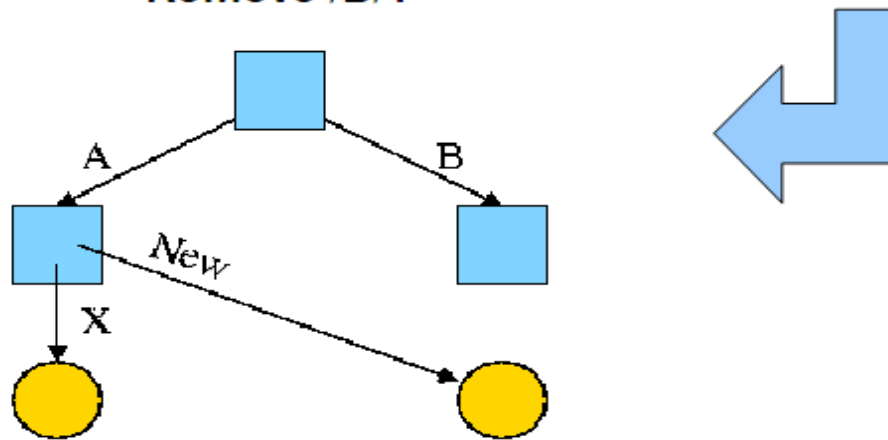
# Aliasing di Linux

- **Another name** for an entity
- Hard link
  - Refer to same **inode on disk**
  - Refer to file/directory
  - **One owner**
- Soft link – symbolic link
  - Pointer to other entities
  - shortcut

# Hard link

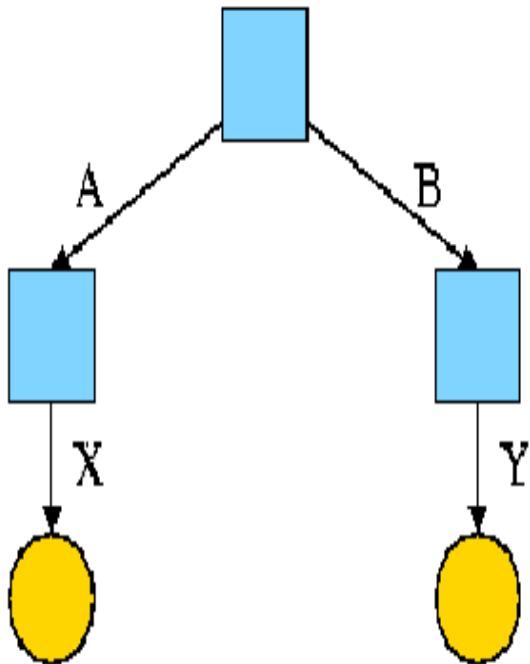


**Remove /B/Y**

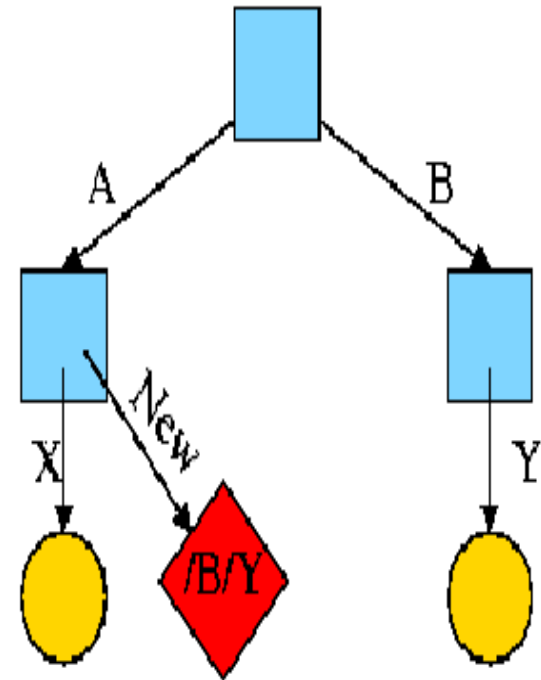


# Soft link (symlink)

Initial tree



Create Symlink (ln -s /B/Y /A/New)



# Pengertian Name Service

- Dalam sistem terdistribusi, **nama** digunakan untuk menunjuk ke suatu resource yang beragam dan tersebar seperti komputer, layanan (services), file, dan remote object
- Merupakan layanan penamaan yang berfungsi untuk menyimpan **naming context**, yakni kumpulan **binding** nama dengan objek, tugasnya untuk **me-resolve nama**.
  - Naming context: yaitu sehimpunan keterkaitan antara nama dan atribut objek: seperti user, komputer, *services dan remote object*.

# Pengertian (2)

- Pengaksesan **resource** pada sistem terdistribusi yang memerlukan:
  - Nama resource (untuk pemanggilan),
  - Alamat (lokasi resource tsb),
  - Pemetaan antara nama dan alamat sebenarnya
  - Rute (bagaimana mencapai lokasi tsb)

# Manfaat penamaan

- **komunikasi** : misal>nama domain sebagai bagian dari email
- **resource sharing** : proses tidak dapat mengakses suatu resource, jika resource tersebut tidak diberi nama
- **location independence**: perubahan lokasi tidak menuntut perubahan nama, asalkan lokasi tidak menjadi bagian dari nama resource tsb.
- **Security** : jika sebuah nama dipilih secara acak dari himpunan besar integer, maka nama tsb hanya bisa diketahui dari legitimate source, sulit jika menebak.
  - Jadi jika seseorang mengetahui nama obyek tsb, maka dia memang diberitahu, karena sulit sekali menebak nama tsb.

# Kebutuhan terhadap name service

- Untuk penamaan unik yang standard
- Untuk scalability
- Untuk consistency
- Untuk performance dan availability
- Agar mudah menyesuaikan terhadap perubahan
- Untuk perlindungan kegagalan

# Jenis nama

- **Pure name** : nama yang tidak perlu di terjemahkan, karena pada nama tersebut sudah menunjuk alamat objek langsung.
  - Contoh : IP
- **non-pure name** : dalam nama mengandung suatu informasi (atribut misalnya) tentang suatu objek.
  - Contoh : URL, alamat email, X.500 Directory Service, IOR (Interoperability Object Reference).

# Istilah-istilah

- **Name resolution:**
  - Nama ditranslasikan ke data ttg resource/object tsb.
- **Binding:**
  - Asosiasi / Pengikatan antara nama & obyek.
  - Biasanya nama diikat (bound) ke attributes dari suatu obyek tertentu.
- **Address:** alamat dari sebuah entitas dalam sistem terdistribusi
- **Attribute:** nilai suatu object property.

# Contoh Pemetaan Atribut

- **DNS** : memetakan dari nama ke atribut alamat IP host
- **X.500 directory service** : memetakan suatu nama seseorang ke beberapa atribut, seperti email, telepon, dsb.
- **CORBA Naming Service** yang memetakan nama *remote objek* ke *remote object reference*.

# Tnameserv - CORBA

\$ tnameserv &

Initial Naming Context:

IOR:000000000000002b49444c3a6f6d672e6f72672f436f734e616d696e672f4e616d696e67436f  
6e746578744578743a312e300000000000100000000000007c00010200000000a3132372e302e  
302e3100038400000035afabcb00000000205ff51852000000010000000000000010000000d544e  
616d6553657276696365000000000000004000000000a000000000000100000001000000200000  
000000010001000000020501000100010020000101090000000100010100

TransientNameServer: setting port for initial object references to: 900

# Composed naming domains used to access a resource from a URL

URL

http://www.cdk4.net:8888/WebExamples/earth.html

DNS lookup

Resource ID (IP number, port number, pathname)

55.55.55.55

8888

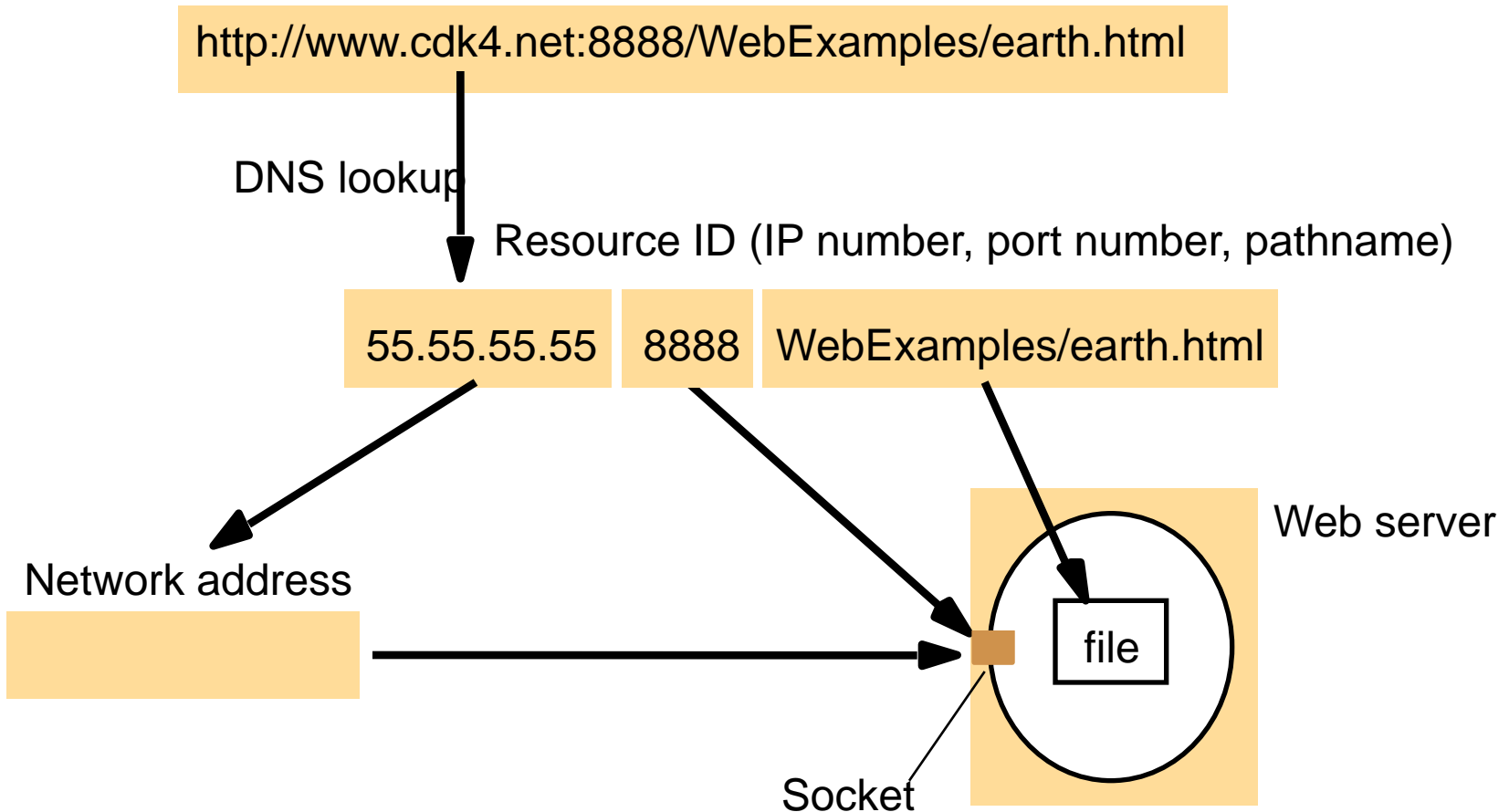
WebExamples/earth.html

Network address

Web server

file

Socket



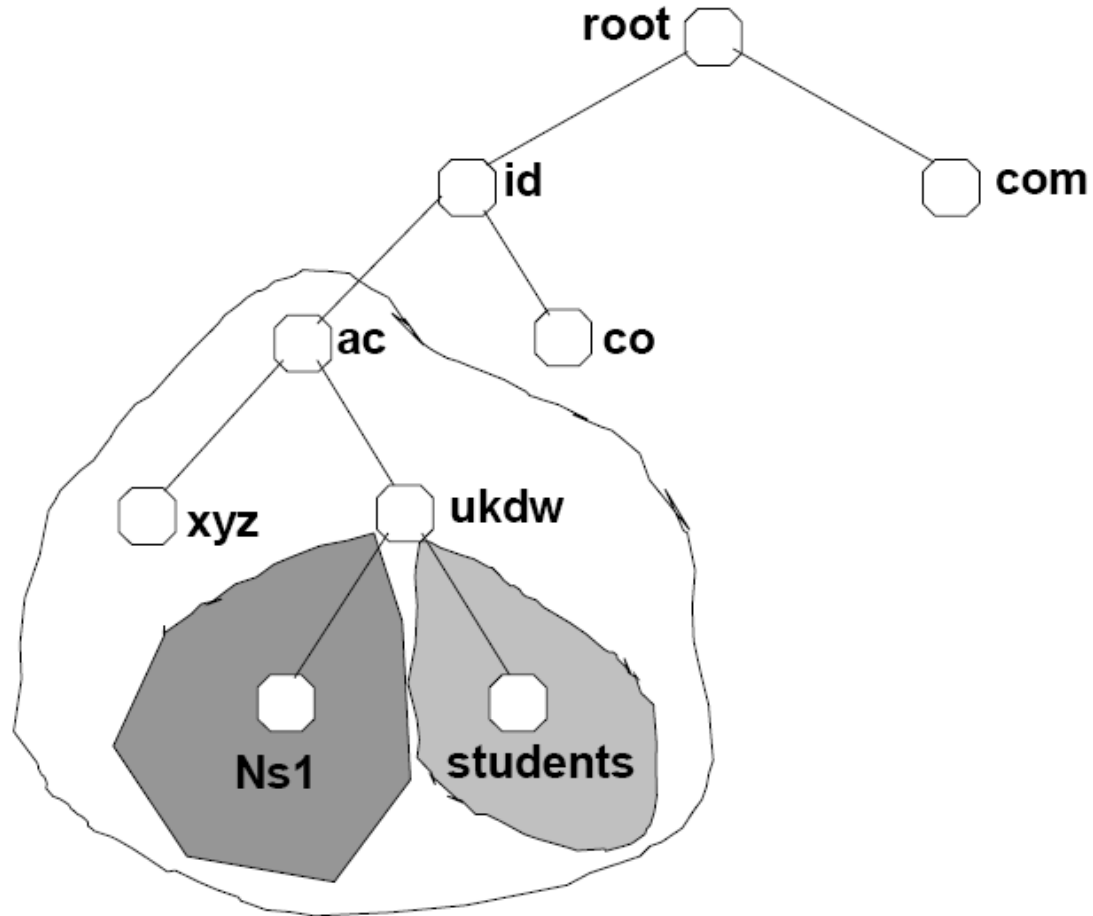
# Yang dilakukan Name Server

- **Partitioning**
  - tidak ada satu name server yang dapat menyimpan seluruh nama dan atribut untuk seluruh jaringan
  - data nama dipartisi berdasarkan domain
- **Replication**
  - sebuah domain biasanya memiliki lebih dari satu name server
  - untuk meningkatkan availability dan performance
- **Caching**
  - sebuah name server dapat melakukan mekanisme caching terhadap data nama dari name server lain
  - hal ini dilakukan untuk mencegah operasi permintaan sama berulang-ulang

# Name resolution (Pemetaan)

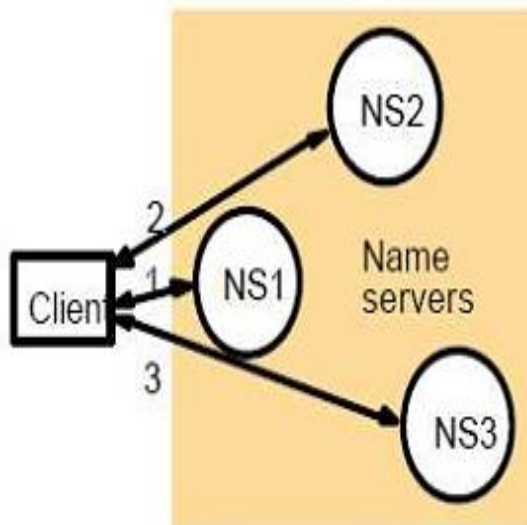
- **Resolution**
  - Mapping a name onto a node referred by name
  - Interested in data stored by the node
- **Path Name Resolution**
  - Begin at first node
    - Root node for absolute name
    - Directory node for relative name
  - End with data from the last node

# Pemetaan

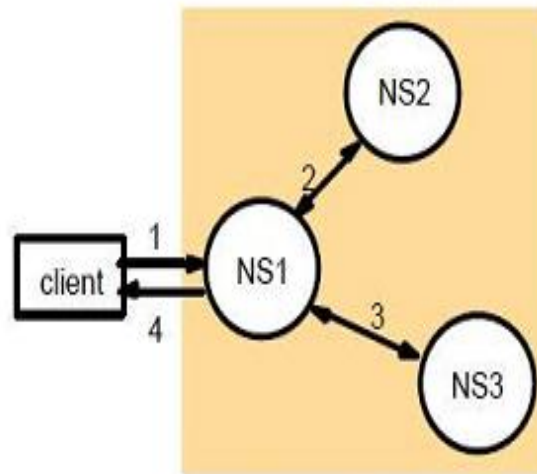


# Jenis Name Resolution

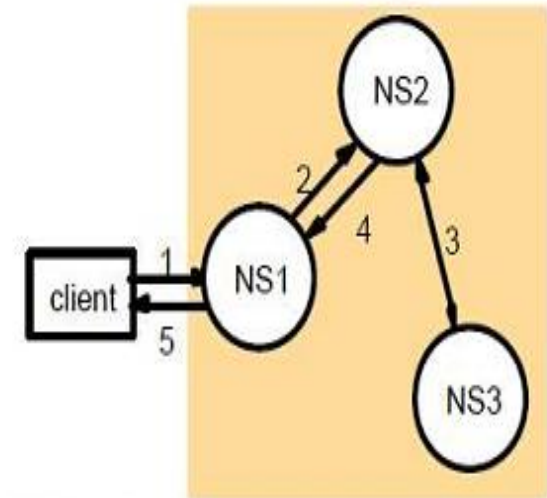
- Iterative Navigation
- non-recursive, server-controlled
- recursive, server-controlled



iterative navigation



Non-recursive  
server-controlled



Recursive  
server-controlled

Wesley Publishers 2000

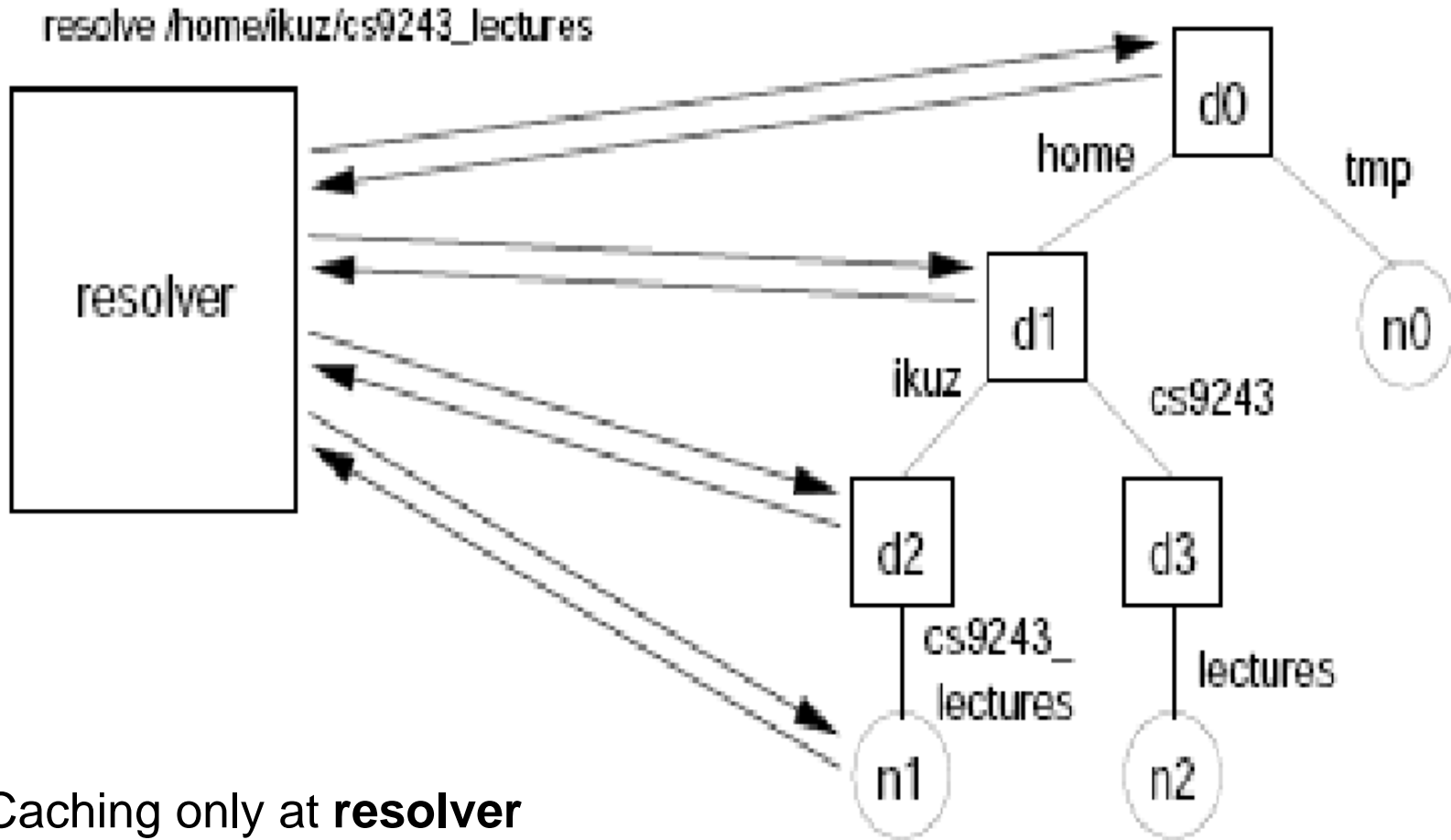
A client iteratively contacts name servers NS1–NS3 in order to resolve a name  
A name server NS1 communicates with other name servers on behalf of a client

# NFS iterative name resolution

## Reason:

- This is because the file service may encounter a **symbolic link** (i.e. an *alias*) when resolving a name.
- A symbolic link must be interpreted in the client's file system name space because it may point to a file in a directory stored at **another server**.
- The client computer must determine **which server this is**, because **only the client knows** its mount points.

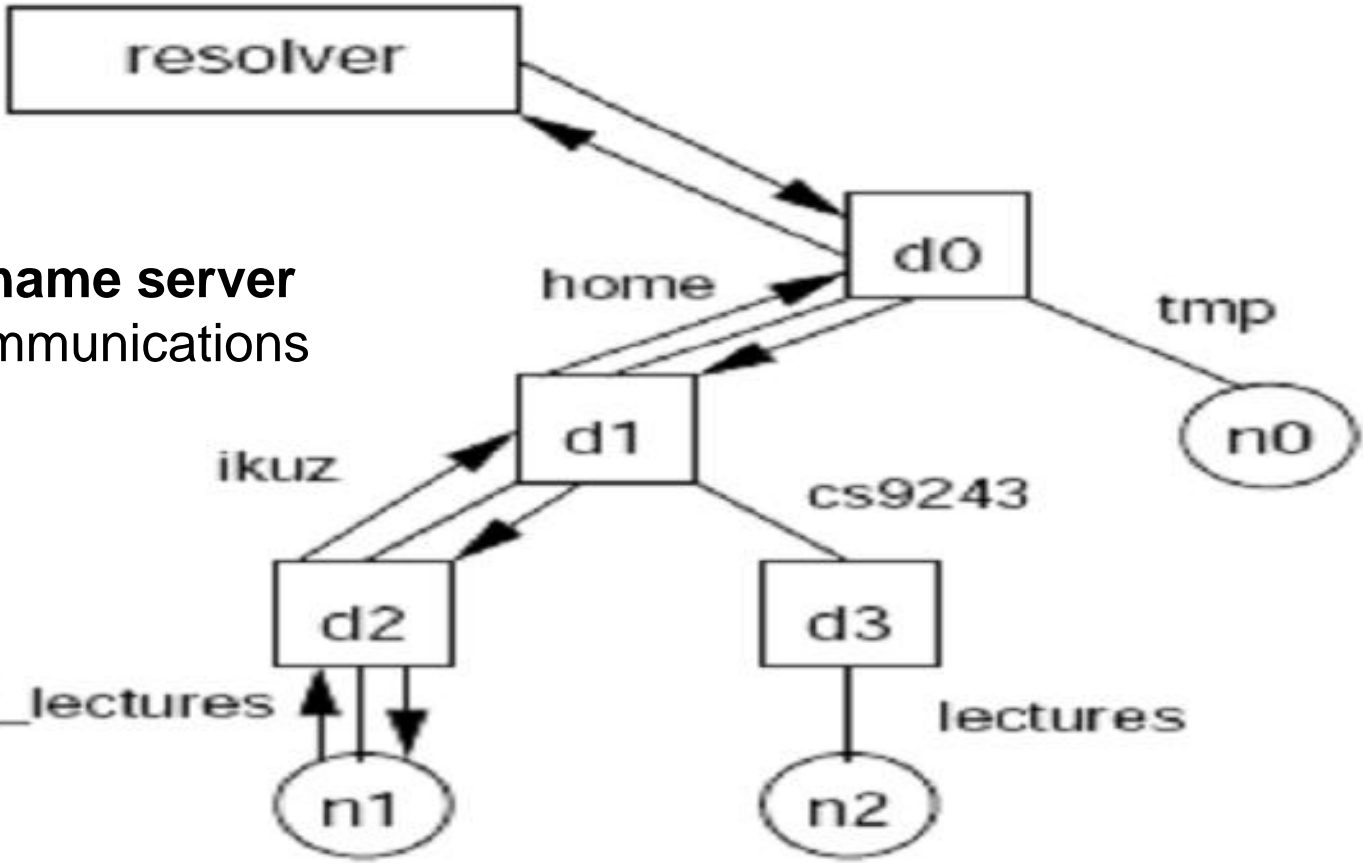
# Iterative / Non Recursive



- Caching only at **resolver**
- Lots of communications (overhead)

# Recursive

resolve /home/ikuz/cs9243\_lectures



- Caching on **name server**
- Reduced communications

# DNS

- Dibuat oleh Paul Mockapetris (1987)
- Berbentuk **tree**
- merupakan **standard** penamaan pada Internet
- database DNS diterapkan dengan sistem **partitioning** yang terbagi-bagi dalam suatu zone berdasar domainnya.
- Pada tiap zone memiliki 2 name server yang menyediakan **authoritative naming information** (replikasi)

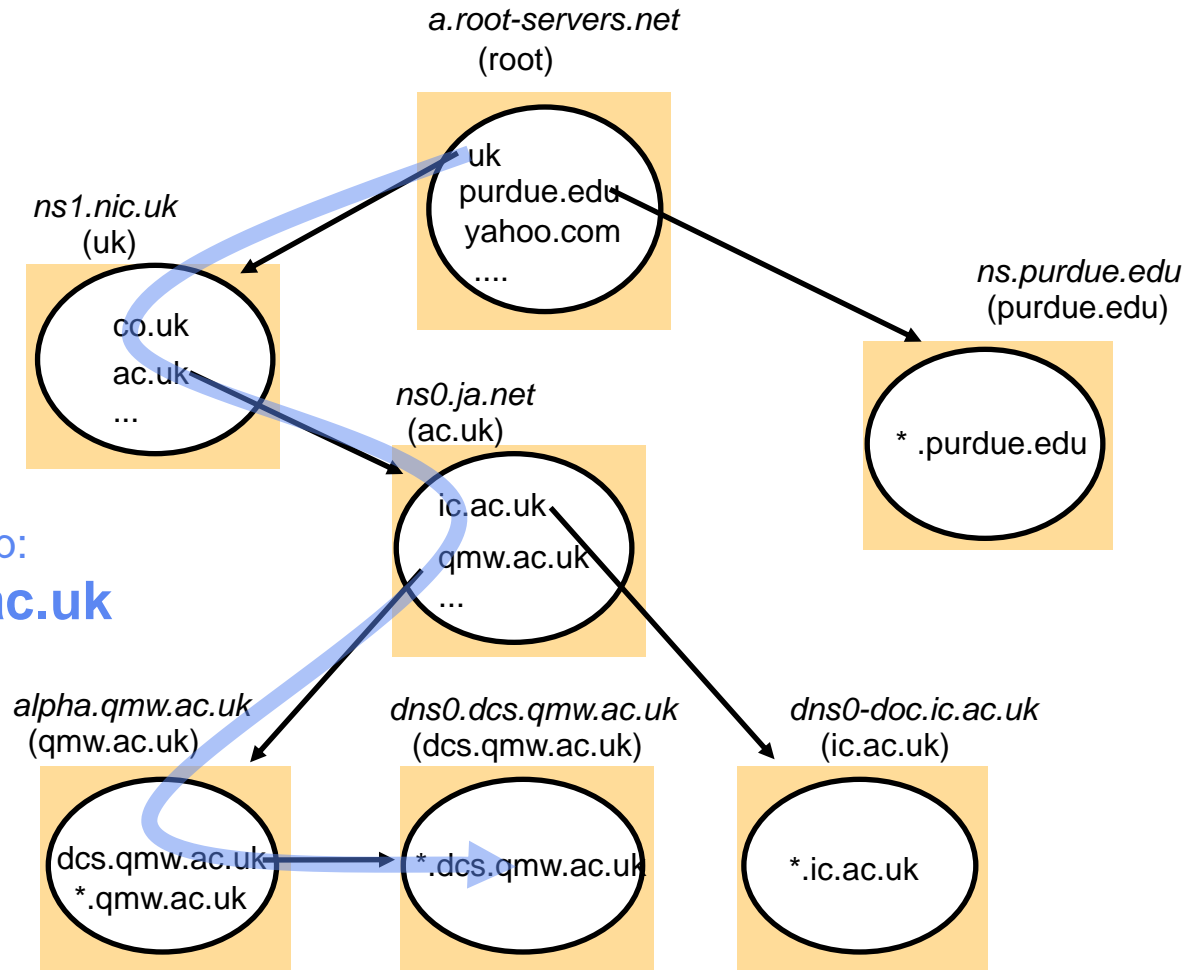
# DNS

- Suatu **authoritative DNS Server** ada 2 jenis
  - **Primary server** : membaca data langsung dari database nama utama untuk zone tersebut
  - **Secondary Server** : secara periodik mendownload data zone dari database utama
- Selain itu DNS juga menerapkan mekanisme **caching** yang selalu menyimpan informasi hasil resolve name yang sudah dilakukan

# DNS name servers

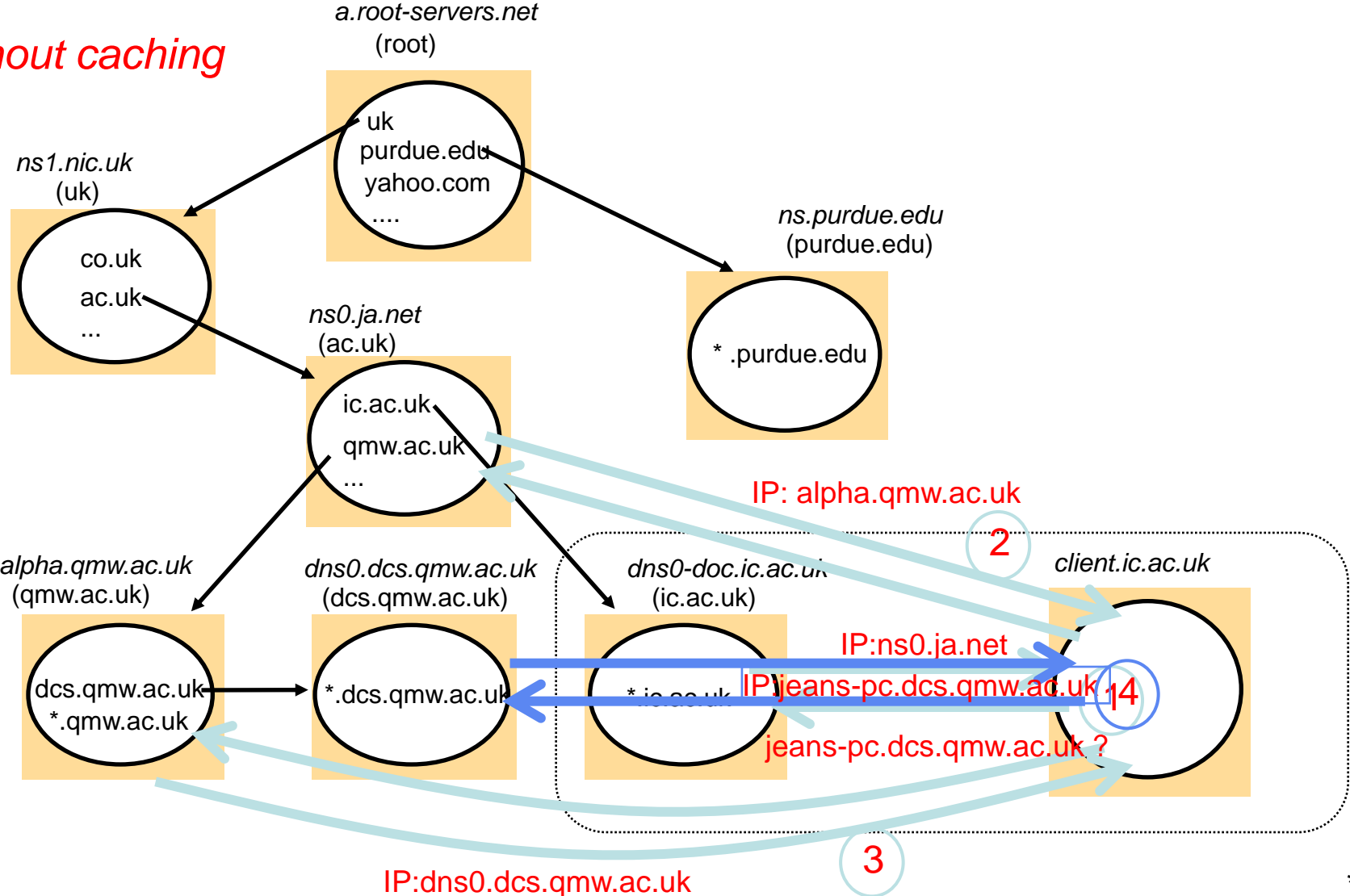
*Note:* Name server names are in italics, and the corresponding domains are in parentheses. Arrows denote name server entries

authoritative path to lookup:  
**jeans-pc.dcs.qmw.ac.uk**

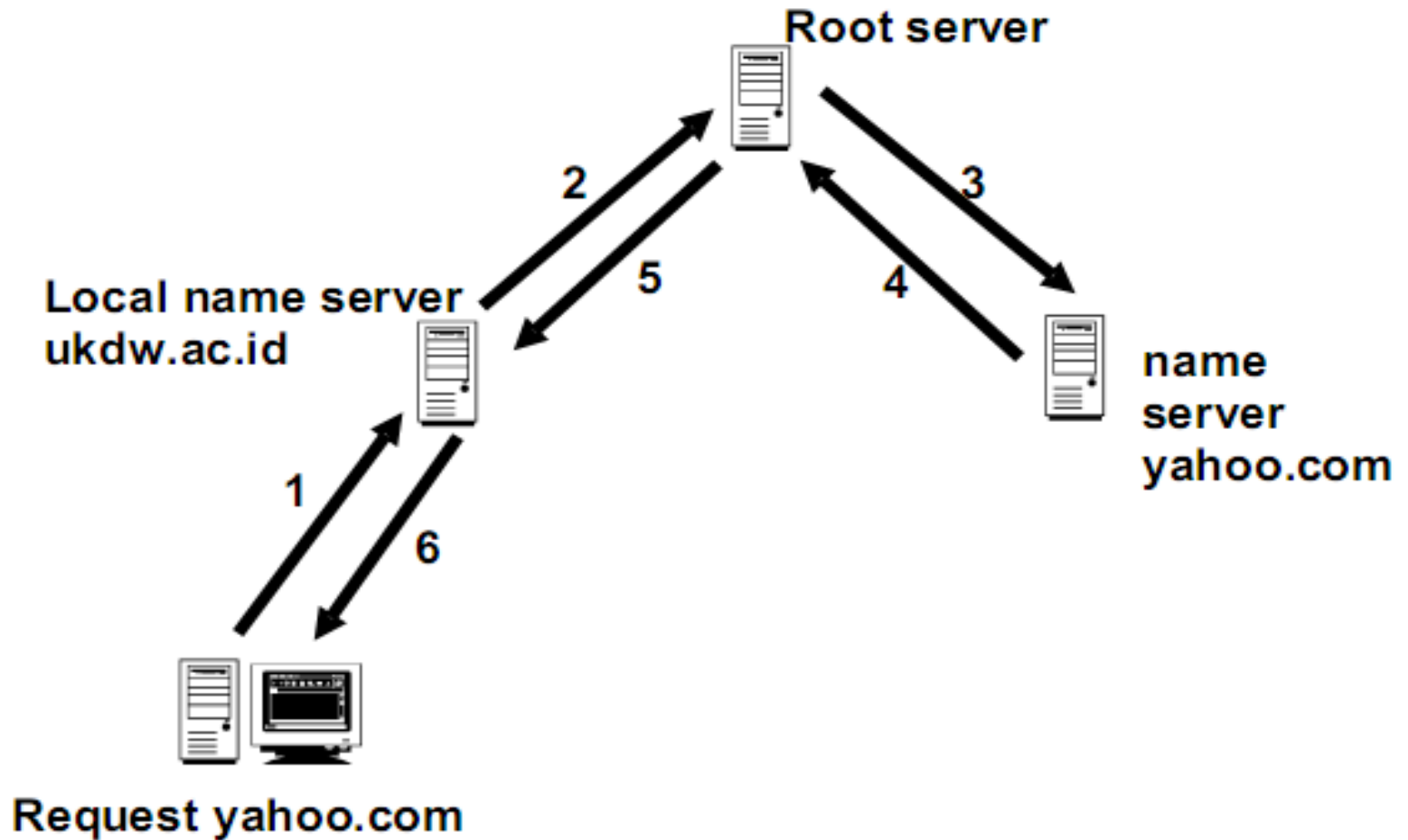


# DNS in typical operation

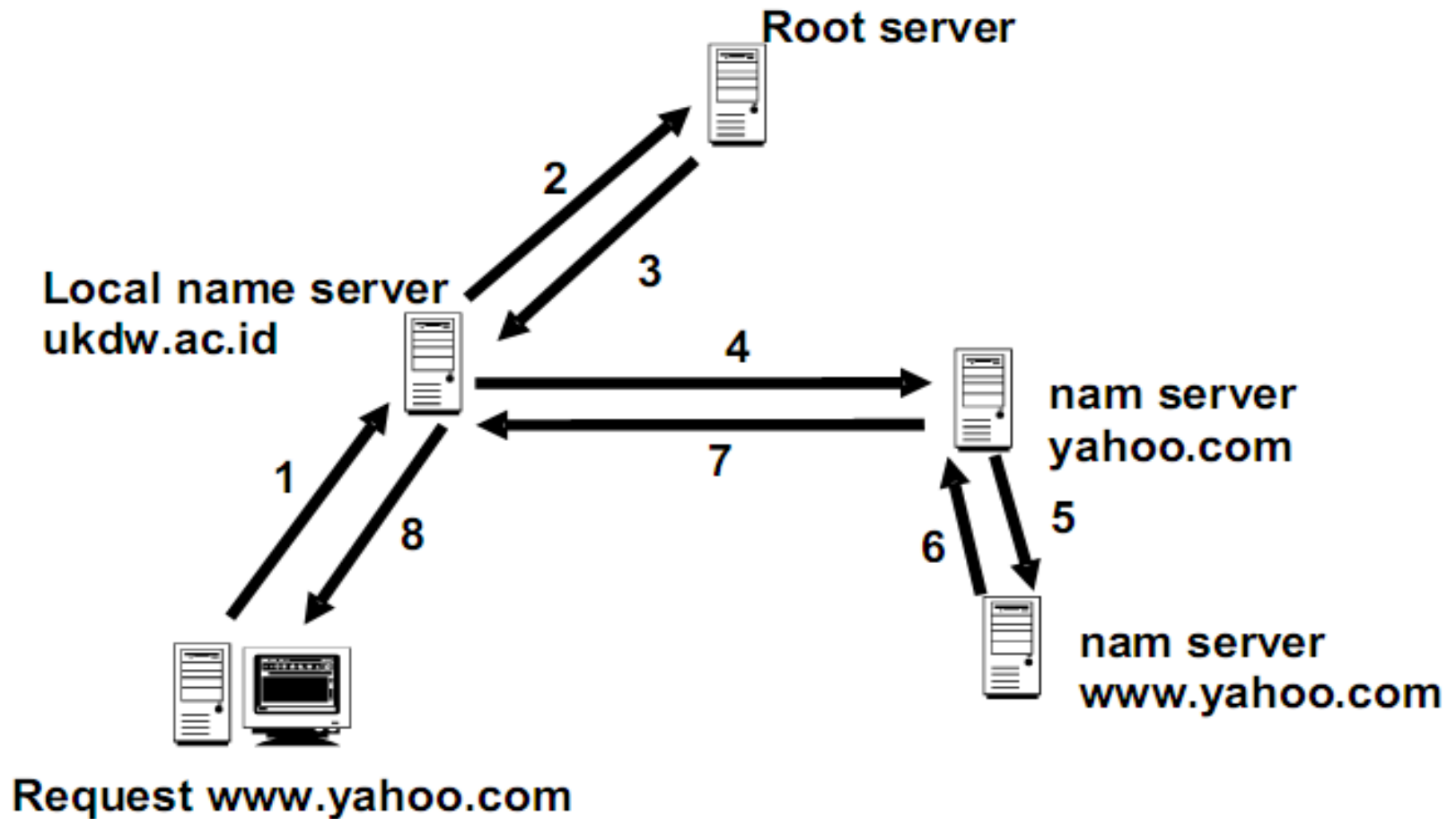
*Without caching*



# Cara kerja DNS (iteratif)



# Cara kerja DNS (rekursif)



# DNS management / layer

- **Global Layer**
  - Stable (rarely changed)
  - Managed by different administrators
- **Administrational**
  - Represents entitles belonging to same organization
  - Relatively stable
  - Maintained by organization's administrators
- **Managerial**
  - Changes regularly
  - Maintained by users/administrators

# DNS resource records

---

<i>Record type</i>	<i>Meaning</i>	<i>Main contents</i>
A	A computer address	IP number
NS	An authoritative name server	Domain name for server
CNAME	The canonical name for an alias	Domain name for alias
SOA	Marks the start of data for a zone	Parameters governing the zone
WKS	A well-known service description	List of service names and protocols
PTR	Domain name pointer (reverse lookups)	Domain name
HINFO	Host information	Machine architecture and operating system
MX	Mail exchange	List of <preference, host> pairs
TXT	Text string	Arbitrary text

---

# DNS resource records

- **Tipe record NS (Name Server)**  
IN NS ns1.anton.com
- **Tipe record MX (Mail eXchange)**  
IN MX 10 ns1.anton.com
- **Tipe record A (Address)**  
ns1 IN A 192.168.0.2
- **Tipe record PTR (Domain Name Pointer)**  
192.168.0.2 IN PTR www.anton.com
- **CNAME (Canonical Name)**  
www IN CNAME ns1.anton.com
- Agar mesin dapat melakukan resolve ke DNS Server, pada file /etc/resolv.conf diisi:  
**domain antonie.net**  
**nameserver 192.33.4.10**

# DNS Query Processing

- DNS Client = **Resolver**
  - Receive queries
  - Format message in DNS Protocol
  - Communicate with other name server
- Most case use: UDP protocol
- Multi query on each request

# Contoh

```
antonie@ubuntu904:~$ nslookup ukdw.ac.id
```

Non-authoritative answer:

Name: ukdw.ac.id

Address: 222.124.22.21

```
antonie@ubuntu904:~$ resolveip www.ukdw.ac.id
```

IP address of www.ukdw.ac.id is 222.124.22.21

# Directory Service

- Sebuah direktori service merupakan sebuah **database**, yang pengelolaan informasinya di dasarkan pada **atribut** setiap item datanya.
- Informasi pada direktori lebih banyak akan **dibaca** daripada operasi update (add, modify, ataupun delete).
- Sehingga pada suatu directory service tidak menerapkan transaksi atau skema roll-back yang kompleks seperti halnya yang ditemui pada database service.

# Directory Service

- Pengubahan informasi pada direktori terjadi pada semua atau beberapa atribut suatu item direktori.
- Dengan sifatnya yang sebagian besar diterapkan operasi pembacaan, maka suatu directory service akan menerapkan suatu model **replikasi** untuk dapat memberikan kehandalan yang lebih baik.
- Suatu directory service akan menerapkan protokol **LDAP** (Lightweight Directory Access Protocol)
- Contoh server: OpenLDAP
  - ldap://antonie.com/dc=x,dc=com
  - ldap://argle.bargle.com/dc=bar,dc=com??sub?uid=barney

# LDAPAdmin

The screenshot shows the LDAP Admin: Test application window. The left pane displays a tree view of the LDAP directory structure for the domain `dc=igh.dc=com [192.168.16.200]`. The tree includes organizational units (OUs) such as `ou=Computers`, `ou=Groups`, `ou=Users`, `ou=Management`, `ou=Marketing`, and `ou=Sales`. Under `ou=Users`, the entry `uid=samba user` is selected.

The right pane displays the details for the selected entry, showing a table of attributes and their values:

Attribute	Value
cn	samba.user
displayName	User, Samba
gidNumber	513
givenName	Samba
homeDirectory	/home/samba user
loginShell	/bin/false
objectClass	posixAccount
objectClass	top
objectClass	inetOrgPerson
objectClass	sambaSamAccount
objectClass	shadowAccount
sambaAcctFlags	[U]
sambaDomainName	MYDOMAIN
sambaHomeDrive	H:
sambaHomePath	\\server\samba user
sambaLogonTime	2147483647
sambaUser	samba user
sambaUserNumber	543-3884341 251-1015615633-125512
uidNumber	62256

A context menu is open over the `uid=samba user` entry in the tree view. The menu options include: `Set Password...`, `New` (with a sub-menu open), `Edit Entry...`, `Copy...`, `Move...`, `Delete`, `Refresh` (F5), `Search...` (Ctrl+F), and `Properties...`. The `New` sub-menu is open, showing options for creating new entries: `Entry...`, `User...`, `Computer...`, `Group...`, `Mailing list...`, `Transport table...`, `Organizational unit...` (highlighted), `Host...`, and `Locality...`.

# NEXT

- Replication and Consistency