Name Services

- Introduction
- Name services
- Domain Name System
- Directory and discovery services
- Case study of the Global Name Service
- Summary

What is Name Service?

- Names: used to refer resources.
- Descriptive attributes also useful means of identification.
- E.g. file name: /etc/passwd or URL: www.cdk4.net
- E.g. Remote object reference or remote file handles.
- A distinct service that is used by client to obtain attributes such as the addresses of resources or objects when given their names
- Fundamental Trios:
  - Names
  - Addresses
  - Routes (path)

Name and Address

- Name
  - A human readable string
- Address
  - Bits used by machines to locate an object
- Bind
  - Association between a name and an address
- Resolve
  - Translate from a name to an address
  - Example

Attributes

Attributes: Value of a property associated with an object.

- DNS
  - IP address
- X.500
  - Person's email address and telephone number
- CORBA Naming Service
  - Remote object reference

Uniform Resource Identifiers

- URL (Uniform Resource Location)
  - Addresses of web resources
  - Dangling problems: a resource may be moved
- URN (Uniform Resource Name)
  - Pure resource names
  - Intend to solve the dangling problems
  - URN lookup service: mapping from URN to URL
  - E.g. urn:ISBN:0-201-62433-8 identifies books that bear the name 0-201-62433-8 in the standard ISBN naming scheme

General name service requirements

- Supports to resolve a name i.e. to lookup attributes from a given name.
- Motivation for name management:
  - Unification: It is often convenient for resources managed by different services to use the same naming scheme. URIs are a good example of this.
  - Integration: Common name service is required in DS because of sharing. (Different Domains)
- Scalability
  - Arbitrary number of names
  - Arbitrary number of administrative organizations
- Flexibility
  - A long lifetime
  - Accommodate variations on the organization of the set of names

General name service requirements (2)

- High availability
  - Most other systems depend upon it
- Fault isolation
  - Isolate location failures from entire service
- Tolerance of mistrust
  - Not all clients are trusted by all components of the system

- When you hand a ***name*** to a naming system to resolve, it must …… *look it up* to find the corresponding object or entity
  - Resolve

- When you hand an ***address*** to a system, it already knows how to find it.
  - E.g., an IP address, an *i-node* of a file system

- When you hand a ***path*** (i.e., a *route*) to a system, you are giving it a sequence of steps it knows how to follow
  - Iteratively or recursively

Name spaces

- A collection of all valid names recognized by a particular service
- Require a syntactic definition
- Hierarchic Name spaces:
  - Each part of name is resolved relative to a separate context.
  - E.g. file system- directory represents a context.

Internal structure of a name

- Hierarchic structure, e.g. */etc/passwd*
  - Resolve relative to a separate context
  - Potentially infinite
  - Different context managed by different people
- Alias

Naming domain

- A single administrative authority
  - A name space for which there exists a single authority
  - E.g. www.pku.edu
- Be stored in a separated server
  - Naming domains are in general stored in different name servers

Combining name spaces

- Homogeneous/ heterogeneous name spaces
- Merging
  - E.g. mount file system in Unix and NFS
  - E.g. create a higher-level root context
- How?
  - Each computer has its own root, with overlapping file names. For example, /etc/passwd refers to one file on red and a different file on blue.

- The obvious way to merge the file systems is to replace each computer's root with a 'super root' and mount each computer's file system in this super root, say as /red and /blue. Users and programs can then refer to/red/etc/passwd and /blue/etc/passwd.

- Heterogeneity:
  - DS allows heterogeneous name spaces to be embedded within it.
  - DCE names may contain junctions, which are similar to mount points in NFS and UNIX except that they allow heterogeneous name spaces to be mounted.
  - Ex: */.../dcs.qmul.ac.uk/principals/Jean.Dollimore.* → **Junction**

Cell

Customizing name spaces

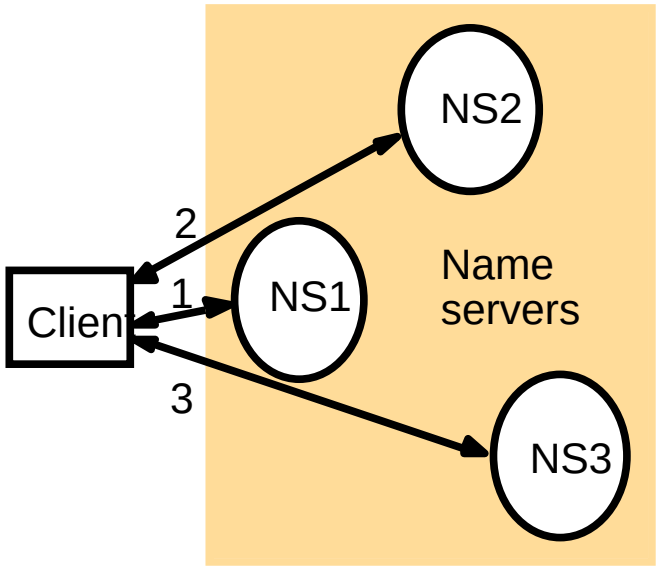
- One file with different names
  - E.g. a NFS directory mounted on different machines
- One name refer to different files
  - E.g. install configuration for multi-platform
- One name space per people

Name resolution

- Resolution is an iterative process, where a name is repeatedly presented the naming context.
- Name space is partitioned in different name servers(very large database)
- Navigation: process of locating naming domain from more than one name server is order to resolve name.
- Iterative navigation
  - Client controlling
  - E.g., DNS, NFS
- Maps given name

onto attributes.

e.g. /etc/passwd



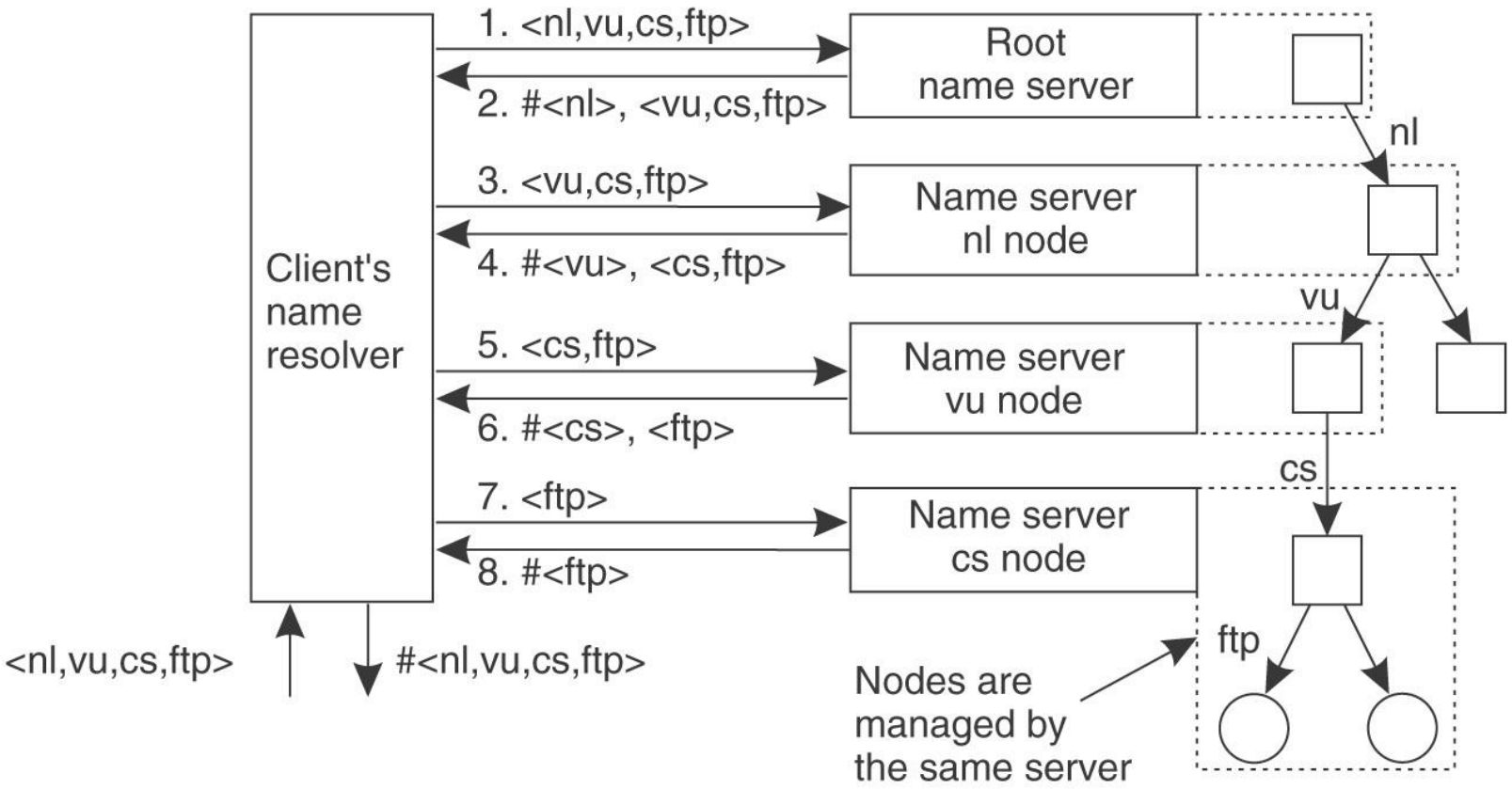
A client iteratively contacts name servers NS1–NS3 in order to resolve a name

- To resolve a name, it is first presented to some initial naming context; resolution iterates as long as further contexts and derived names are output.

- Example: /etc/passwd, in which ‘etc’ is presented to the context ‘/’, and then ‘passwd’ is presented to the context ‘/etc’.

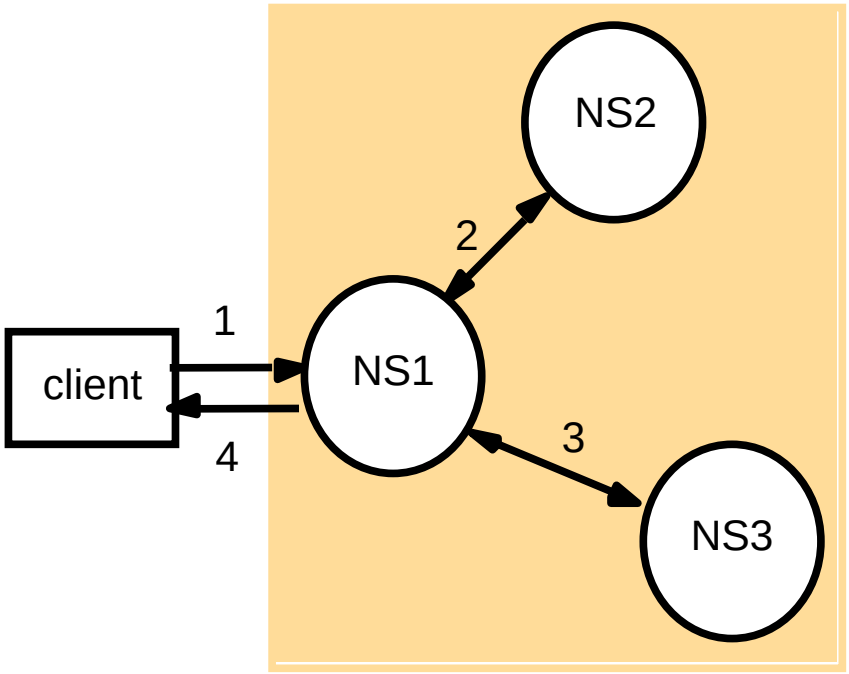
- One navigation model that DNS supports is known as iterative To resolve a name, a client presents the name to the local name server, which attempts to resolve it.
- The local name server has the name, it returns the result immediately. If it does not, it will suggest another server that will be able to help.
- Resolution proceeds at the new server, with further navigation as necessary until the name is located or is discovered to be unbound.

E.g., *ftp.cs.vu.nl*

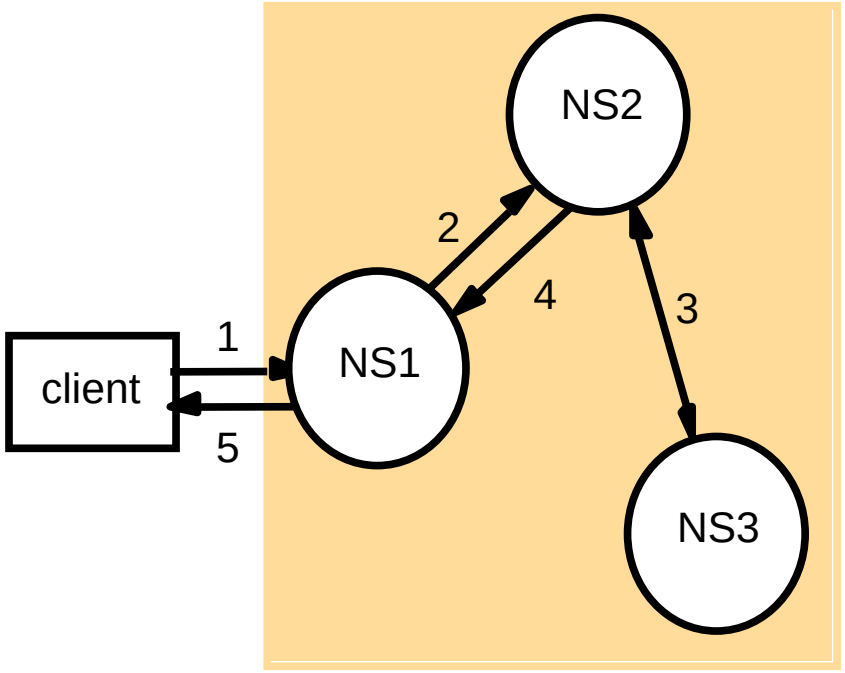


Server controlled navigation

- Non-recursive/Recursive
  - Recursive type is suitable to environment where there are administrative domain prohibits



Non-recursive
server-controlled



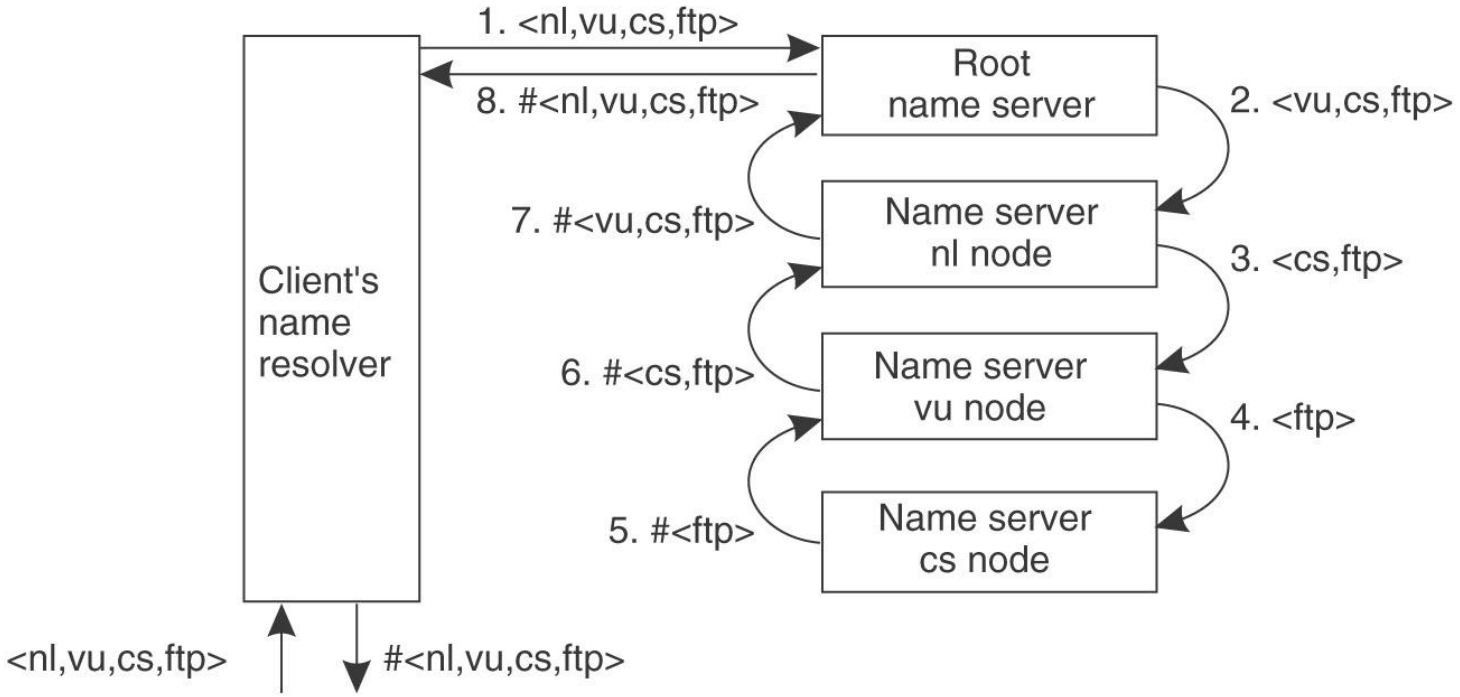
Recursive
server-controlled

A name server NS1 communicates with other name servers on behalf of a client

- Under non-recursive server-controlled navigation, any name server may be chosen by the client.
- This server communicates by multicast or iteratively with its peers in as it were a client.
- Under recursive server-controlled navigation, the client once more contacts a single server.
- If this server does not store the name, the server contacts a peer storing a (larger) prefix of the name, which in turn attempts to resolve it.
- This procedure continues recursively until the name is resolved.

- If a name service spans distinct administrative domains, then clients executing in one administrative domain may be prohibited from accessing name servers belonging to another such domain.
- Moreover, even name servers may be prohibited from discovering the disposition of naming data across name servers in another administrative domain.
- Then, both client-controlled and non-recursive server-controlled navigation are inappropriate, and recursive server-controlled navigation must be used.

E.g., *ftp.cs.vu.nl*



Caching tech. in name resolution

- To store the result of previous name resolutions.
- Enhance response time
- Eliminate the workload of high-level name servers
- Isolate the failures of high-level name servers

The Domain Name System

- Original Internet Naming scheme
  - A central master files
  - Download to all hosts by FTP
- Domain names [1987]
  - Name space is partitioned both organizationally and according to geography
  - Convert domain names into IP addresses
- Hierarchical partitioning of the name database, replication of naming data & Caching.

Parts of a URL

The DNS name space: Top Level Domains

Com – Commercial organizations

Edu – Universities and other educational institutions

Gov – US governmental agencies

Mil – US military organizations

Net – Major network support centres

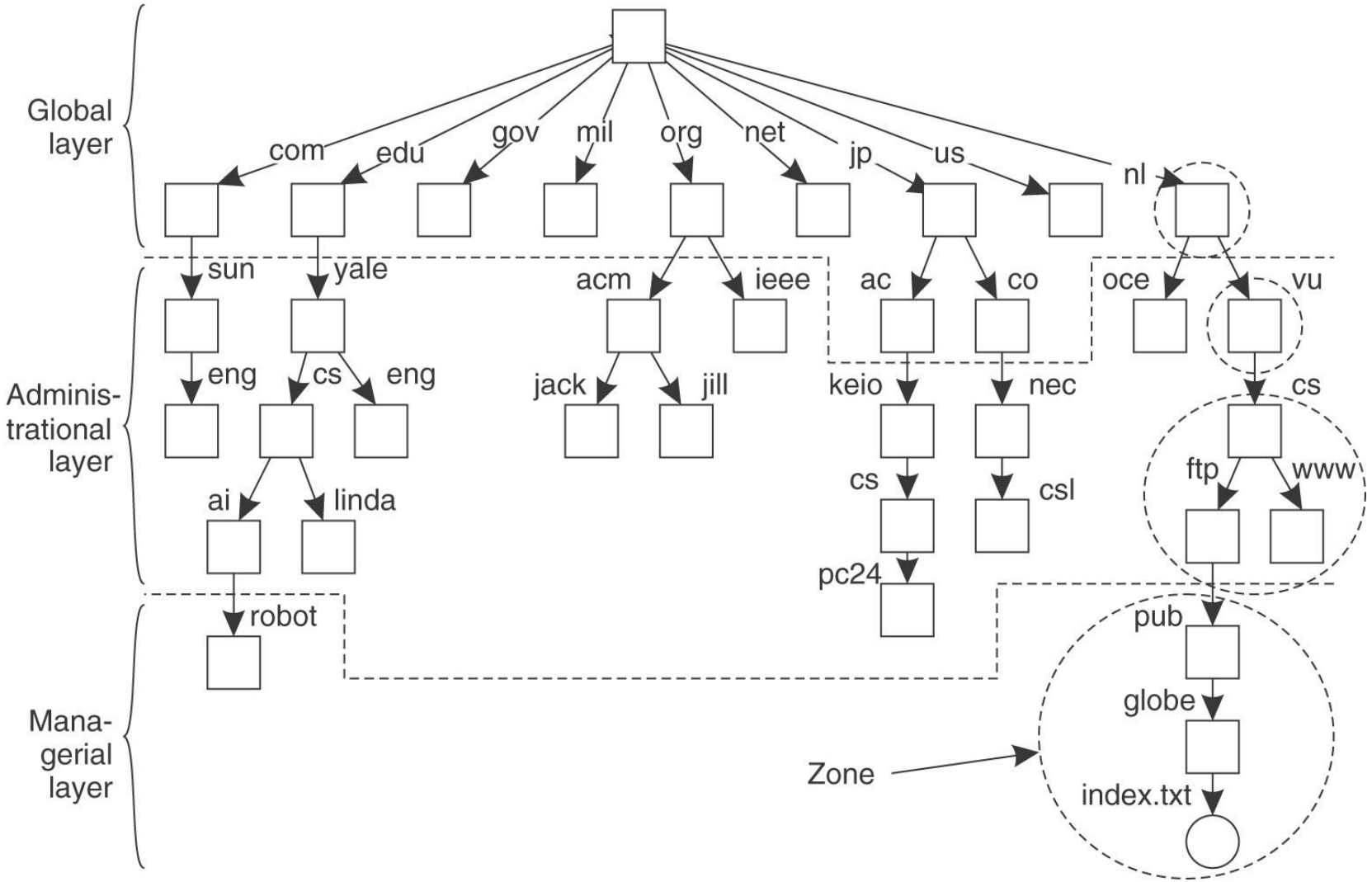
Org – Organizations not mentioned above

Int – International organizations

Us – united States

Uk – United Kingdom

Cn - China


Global layer
Adminis-trational layer
Mana-gerial layer
com
edu
gov
mil
org
net
jp
us
nl
sun
yale
acm
ieee
ac
co
oce
vu
eng
cs
eng
jack
jill
keio
nec
cs
ai
linda
cs
csl
ftp
www
pc24
robot
pub
globe
index.txt
Zone

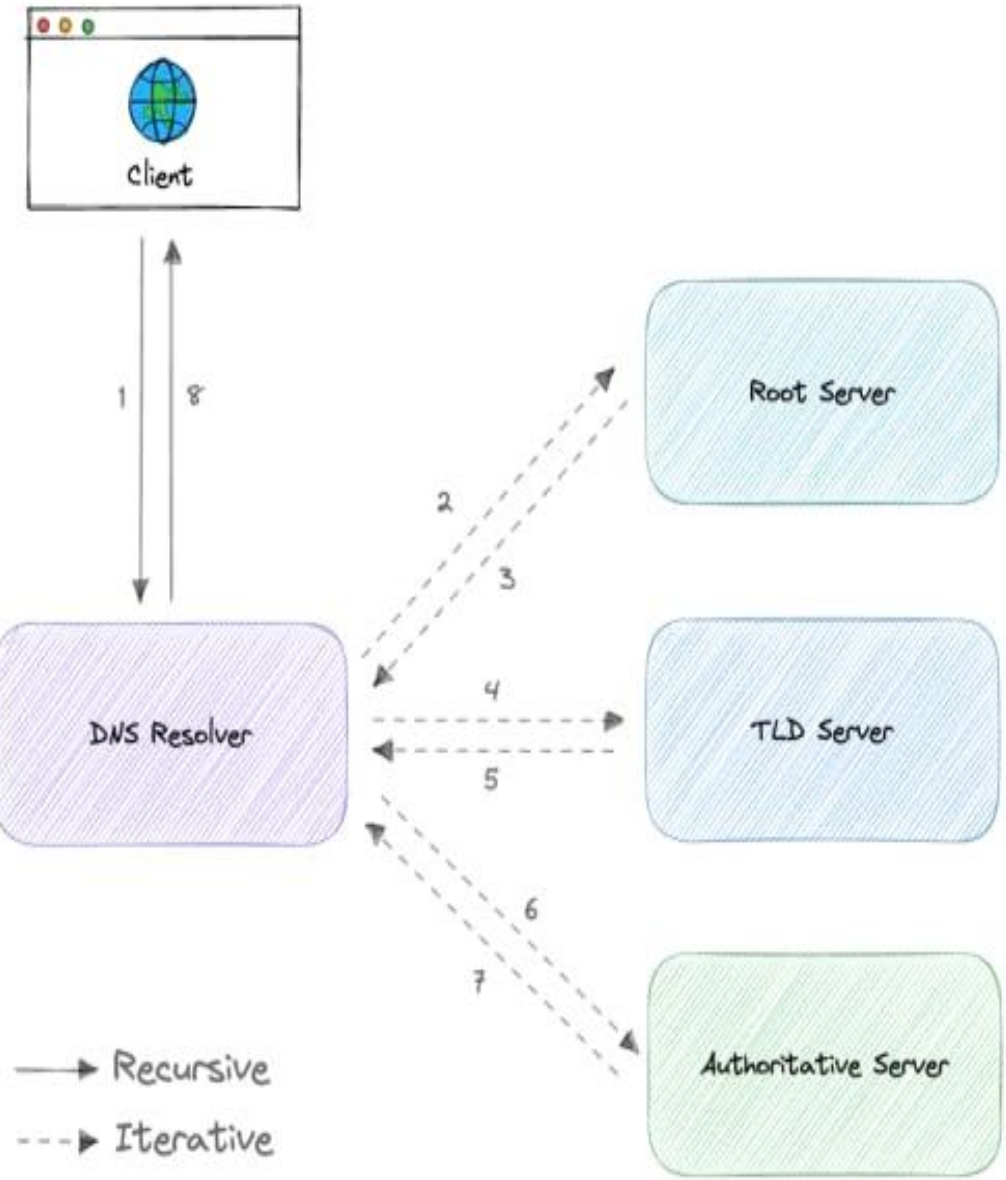
DNS queries

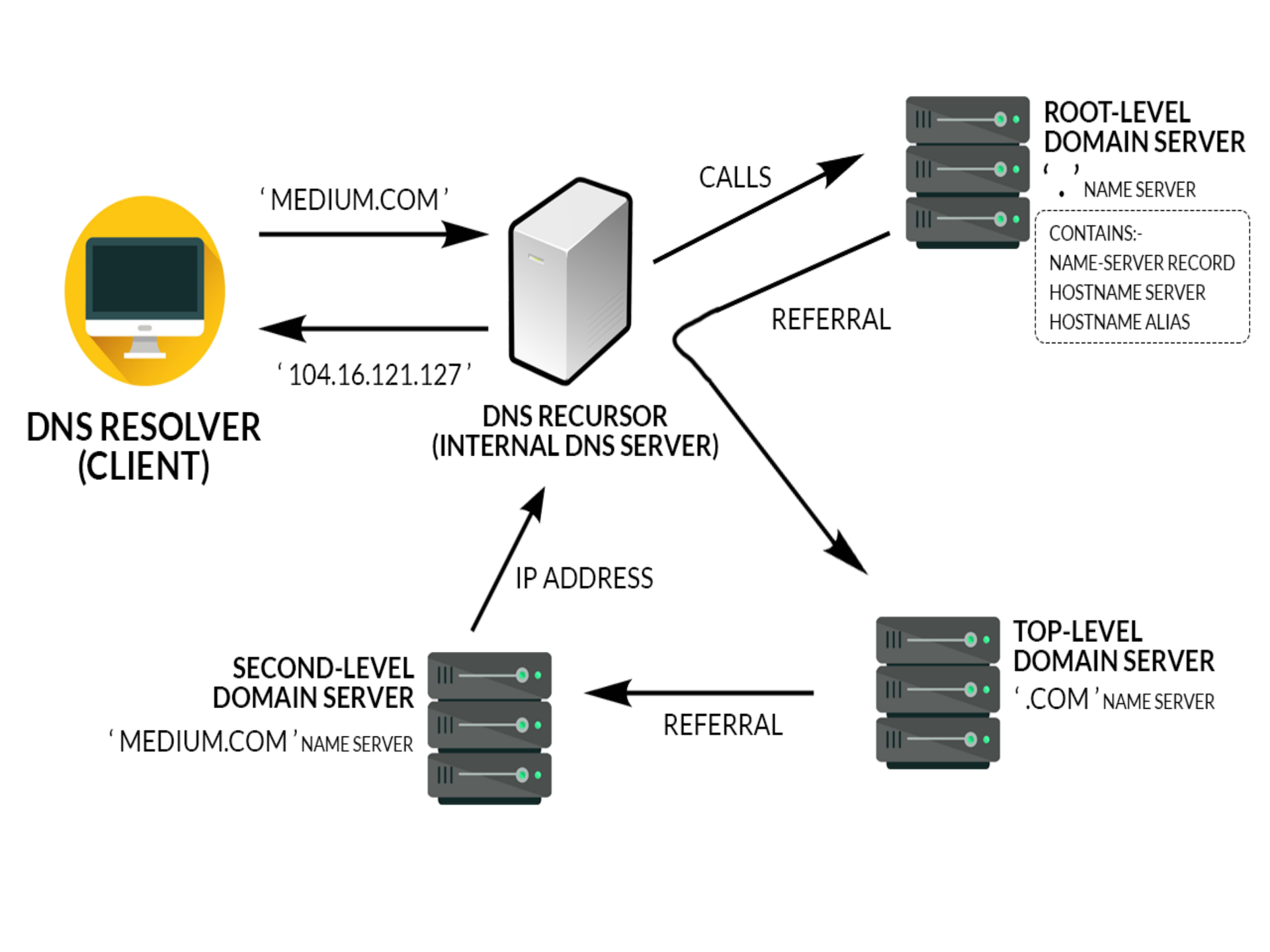
- Internet DNS is primarily used for simple host name resolution and for looking up electronic mail hosts, as follows:

1. Host name resolution: From URL to IP address
2. Mail host location
   - Given a domain name, return a list of domain names of hosts that can accept the mail
   - E.g. tom@dcs.rnx.ac.uk DNS is queried with the address dcs.rnx.ac.uk and the type designation ‘mail’. It returns a list of domain names of hosts that can accept mail for dcs.rnx.ac.uk, if such exist.
3. Reverse resolution: From IP to URL

DNS queries …*continued*

4. Host information
   - E.g. the architecture type or operating system of a machine
5. Well-known services
   - A list of the services run by a computer (telnet/ ftp)
   - Protocol used to obtain them (UDP & TCP)


Client
1
8
Root Server
2
3
DNS Resolver
4
5
TLD Server
6
7
Authoritative Server
Recursive
Iterative


ROOT-LEVEL
DOMAIN SERVER
' '
. NAME SERVER
CONTAINS:-
NAME-SERVER RECORD
HOSTNAME SERVER
HOSTNAME ALIAS
CALLS
REFERRAL
' MEDIUM.COM '
' 104.16.121.127 '
DNS RESOLVER
(CLIENT)
DNS RECURSOR
(INTERNAL DNS SERVER)
IP ADDRESS
TOP-LEVEL
DOMAIN SERVER
' .COM ' NAME SERVER
REFERRAL
SECOND-LEVEL
DOMAIN SERVER
' MEDIUM.COM ' NAME SERVER

DNS name servers

- DNS names are divided into Zones
- Zone
  - Include names in the domain, less any sub-domains(a zone could contain data for Queen Mary, University of London – *qmul.ac.uk* – less the data held by departments (for example the Department of Computer Science – dcs.qmul.ac.uk).
  - At least two name servers for the zone
  - Hold name servers for the sub-domains
- Each server hold zero or more Zones
  - Zero zone: the caching name server

DNS name servers …*continued*

- Servers that a name server holds
  - Lower-level name servers
    - Child name servers
  - High-level name servers
    - One or more root name servers
    - Parent name server

DNS name resolution

- Iterative navigation / recursive navigation
- Example
- DNS resource types

Navigation and Query Processing

- Resolver: Client that access name server.
- Resolver specifies which type of navigation is required.
  - Recursive and Iterative Navigation
- BIND: Berkeley Internet Name Domain
  - Implementation of DNS on UNIX
  - Supports primary, secondary and cache-based servers

DNS performance

- Replication
  - Zone data are replicated on at least two name servers
  - Master server / secondary server
    - Synchronize periodically
- Cache
  - Any server is free to cache data
  - Time-to-live value

DNS performance (2)

- Availability & Scalability
  - Achieved by a combination of replication, cache and partition
- Acceptable inconsistent naming data

Directory Services

- A special kind of naming service
  - Searching attributes (no name but knows only attributes)
  - E.g. What is the name of user who is phone number is 0250-22224533
  - Which computer in this building is running on MAC OS?
- Directory Service: a service that stores collections of bindings between names and attribute and looks up entries that match attribute-based specifications.
- Entries
  - Each entry is concerned with a set of <attribute, value> pairs
- Query
  - Lookup by known attributes
  - Return interested attributes
  - E.g. query one's telephone No. by his name

Directory services (2)

- Yellow page / white page
  - Directory service / Conventional naming service
- Directory servers and navigation
  - Similar to name service
- Example
  - Active Directory Service
  - X.500
  - LDAP

Discovery services

- A special kind of directory service
- Register the services provided in a spontaneous network.(can connect and disconnect unpredictably.)
- General operations
  - Register / lookup / de-register
- E.g. a registered printer

ResourceClass=printer, type=laser, colour=yes, resolution=600dpi,

Location=room101, url=http://www.hotelDuLac.com/services/printer57

Introduction to GNS(Global Name Service)

- Designed by DEC lab [lampson 1986]
- To provide facilities for resource location, mail addressing and Authentication
- Design objectives of GNS:
  - Large Size: Millions of computer names
  - Long life time: accommodate changes
  - High Availability
  - Fault Isolation: Local failures don't cause the entire system to fail.
  - Tolerance of Mistrust

Architecture of GNS

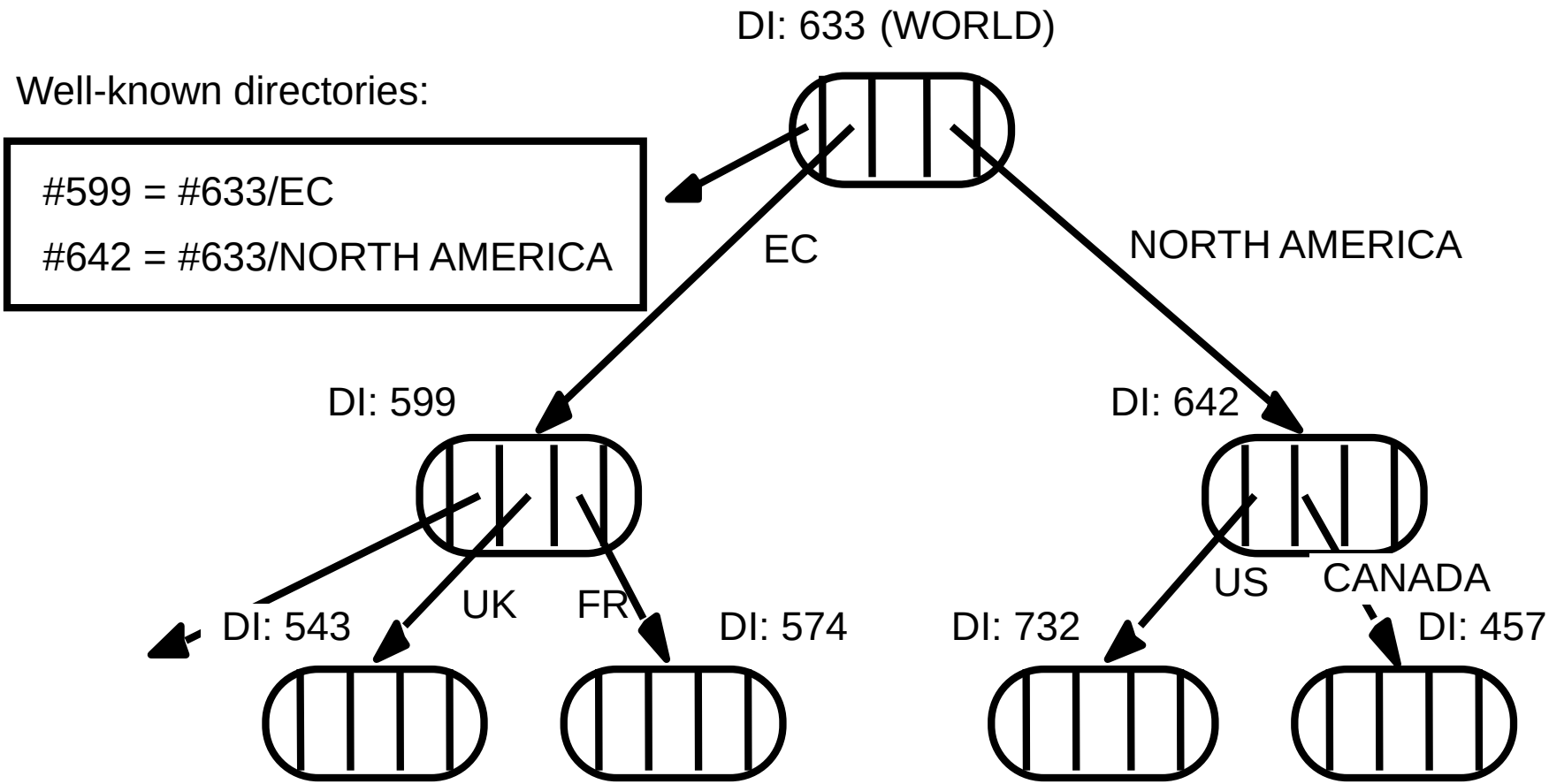
- Directory tree / value tree
- Two levels:
  - Client Level: hierarchical names & their values with operations
  - Admin Level: Synchronized copies of database
- Directory identifier (DI)
  - Unique identifier of a directory
- Client side: Name of an entry
  - <directory name, value name>
  - E.g. <EC/UK/AC/QMW, Peter.Smith/password>

Architecture of GNS (2)

- Multiple name servers
  - Directory tree is partitioned and stored in many servers
- Replication
  - Each partition is replicated in several servers
- Cache
  - Inconsistency cache data is acceptable

How does GNS accommodate changes?

- Merge two name space by a super-root
  - How to it transparent to client applications?
  - E.g. how to locate /uk/ac/qmw ?



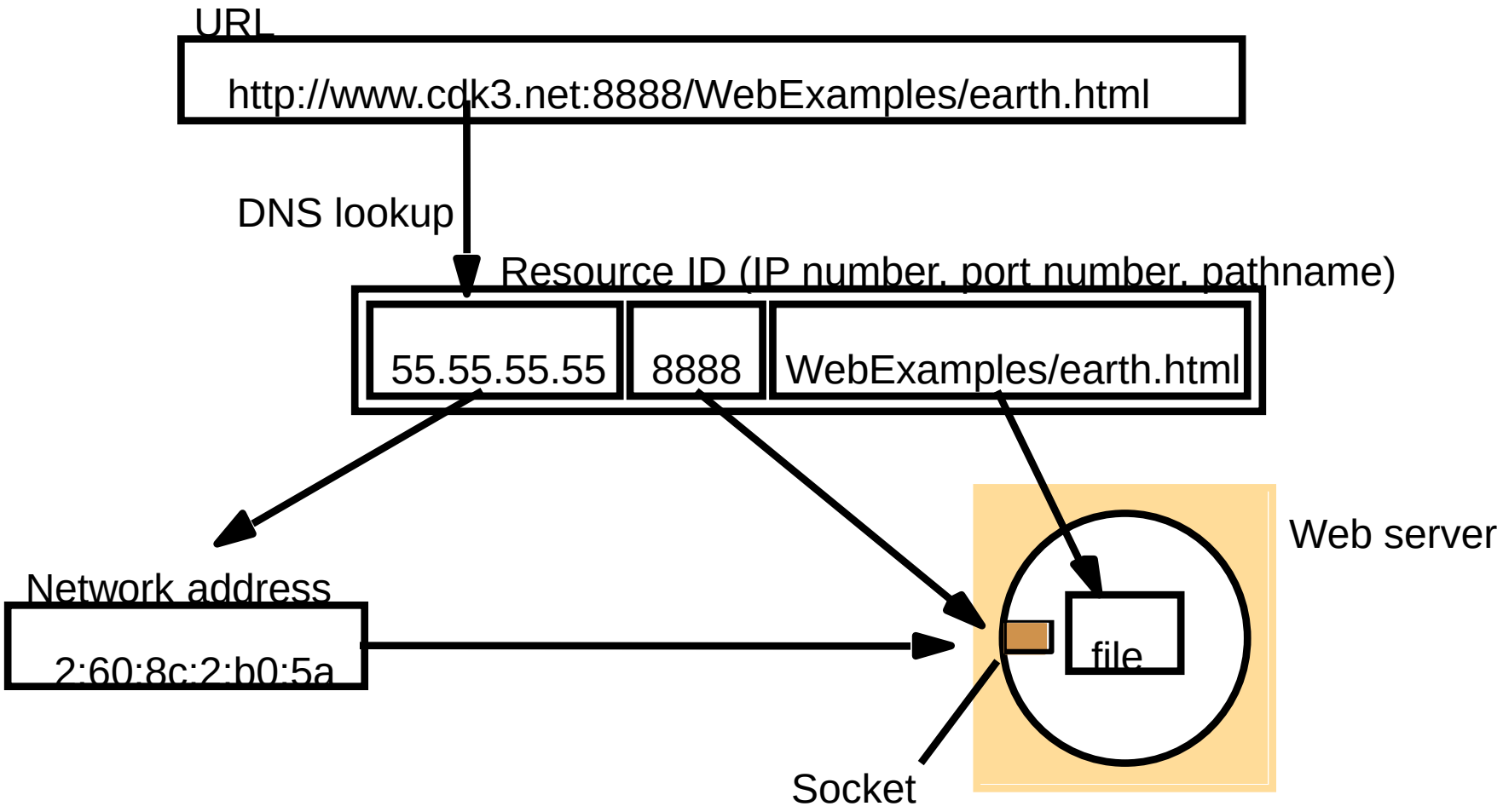
Working root DI

- Client agent
  - Store the DI of working root by client agent
  - E.g., for </UK/AC/QMW, Peter.Smith>, client agent stores #599 which is the DI of "/", i.e. EC
- Resolve name
  - Working root DI + relative path
    - Uniquely refer to a name in the merged tree
  - E.g. <#599/UK/AC/QMW, Peter.Smith>

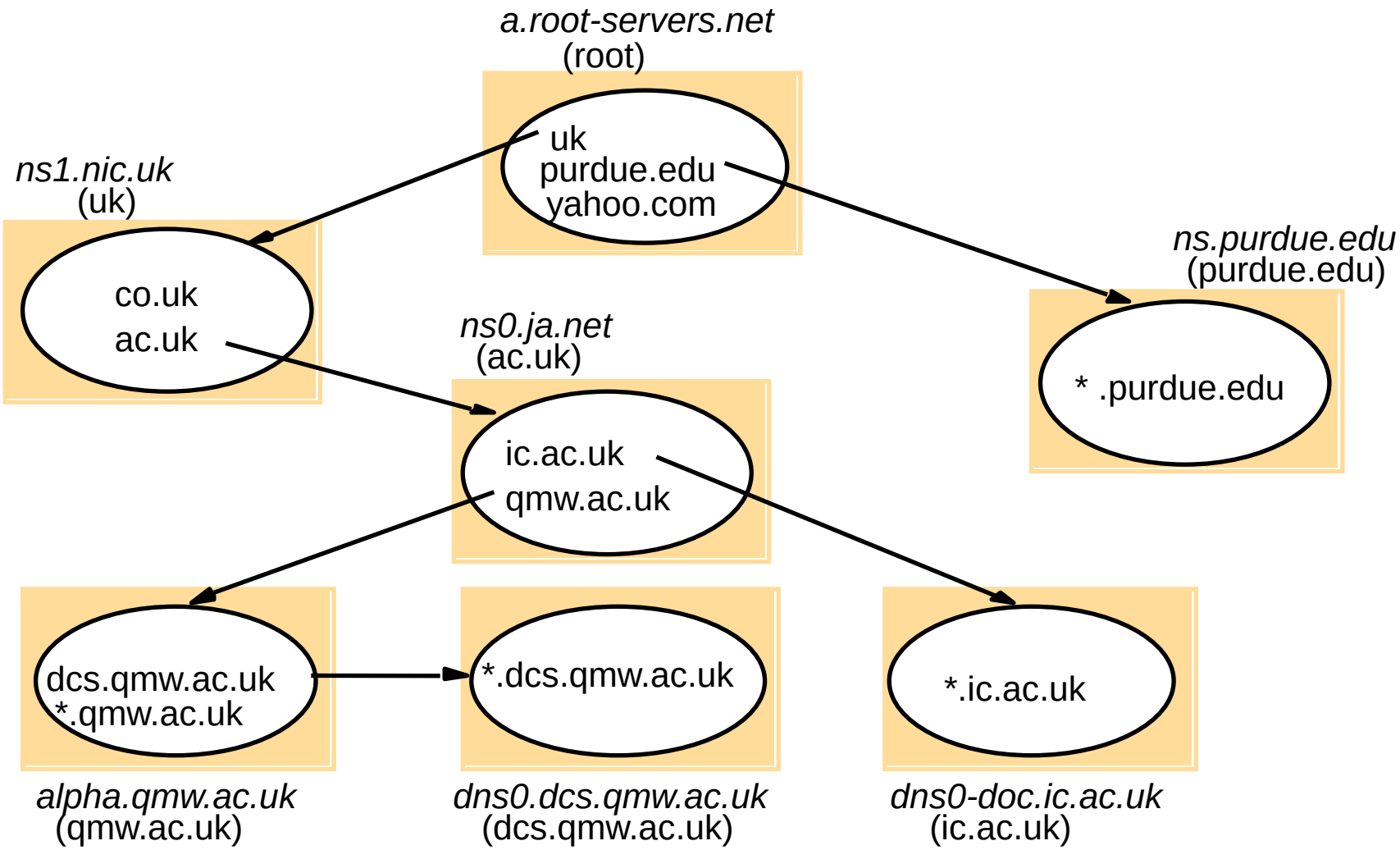
Well-known directories

- A table of well-know directories
  - Mapping between working root DI to new absolute path
- Replication
  - Well-know directories are replicated at each nodes
  - Bottleneck of consistency
- Examples

Composed naming domains used to access a resource from a URL



DNS name servers



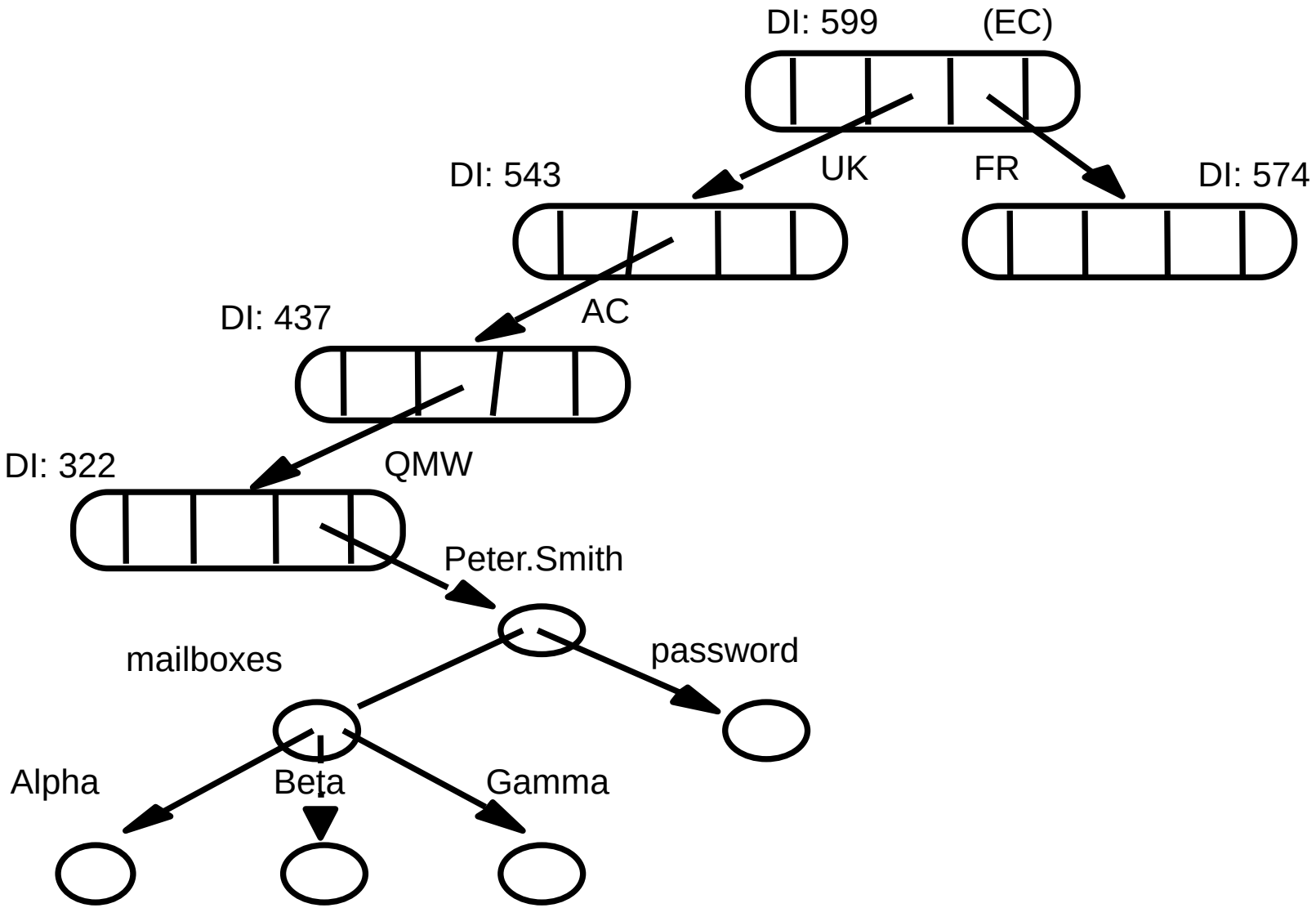
Name server names are in italics, and the corresponding domains are in parentheses. Arrows denote name server entries

DNS resource records

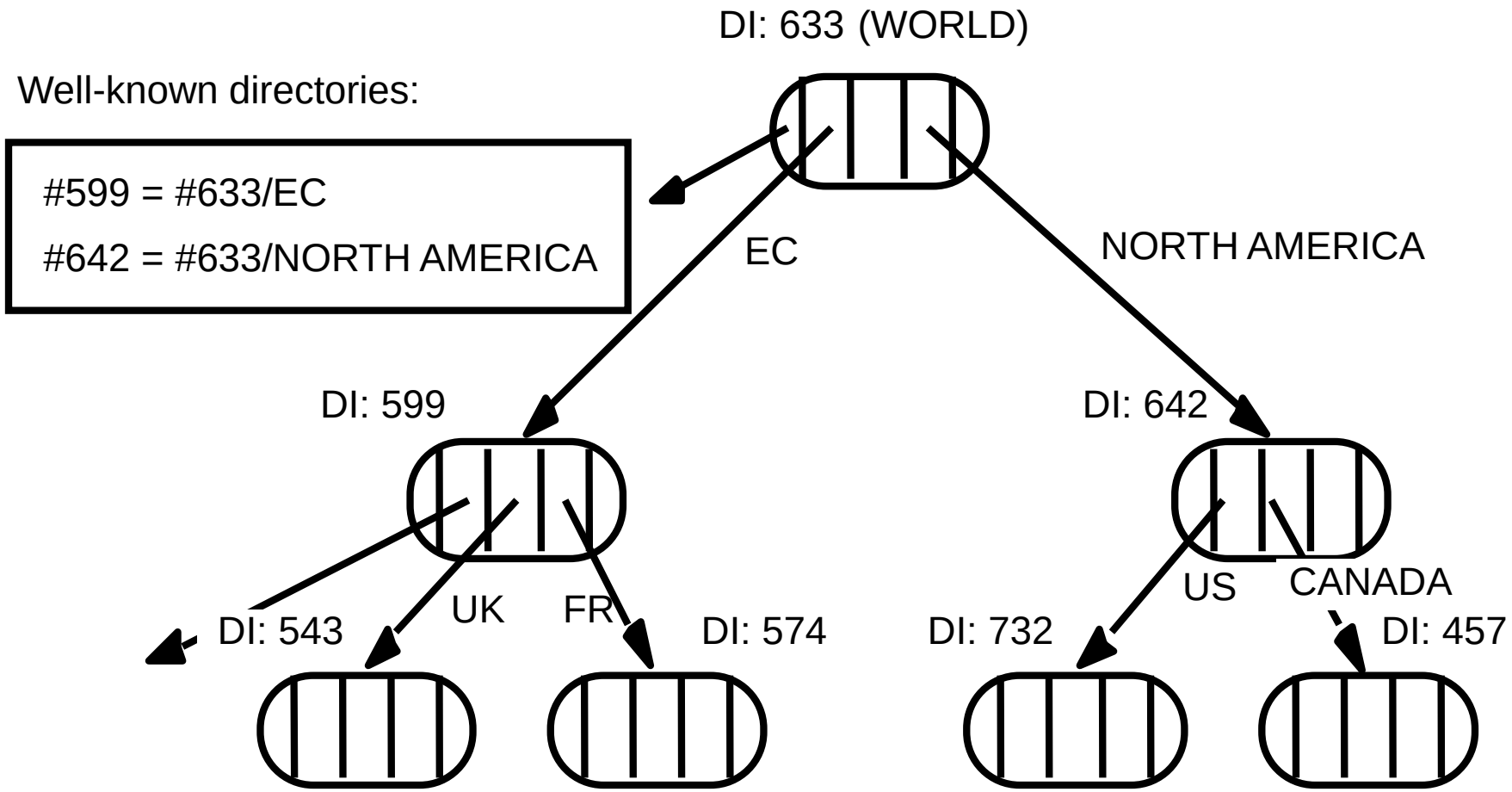
Record type	*Meaning*	*Main contents*
A	A computer address	IP number
NS	An authoritative name server	Domain name for server
CNAME	The canonical name for an alias	Domain name for alias
SOA	Marks the start of data for a zone	Parameters governing the zone
WKS	A well-known service description	List of service 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

Zone data are stored by name servers in files in one of several fixed types of resource record.

GNS directory tree and value tree for user Peter.Smith



Merging trees under a new root



Restructuring the directory