Distributed Systems

6. Name Services

Werner Nutt

Naming Concepts

Names = strings used to identify objects (files, computers, people, processes, objects)

- Textual names (human readable)
  - used to identify individual services, people
    - email address: Hans.Mair@inf.unibz.it
    - URL: www.google.com
  - or groups of people or objects
    - mailing lists: professors@unibz.it
    - mail domains (if there are several mail exchangers)

Naming Concepts (cntd)

- Numeric addresses (identify the location of an object)
  - locate individual resources, e.g.

    193.206.186.100 (IP host address)
  - special case: group addresses, e.g.

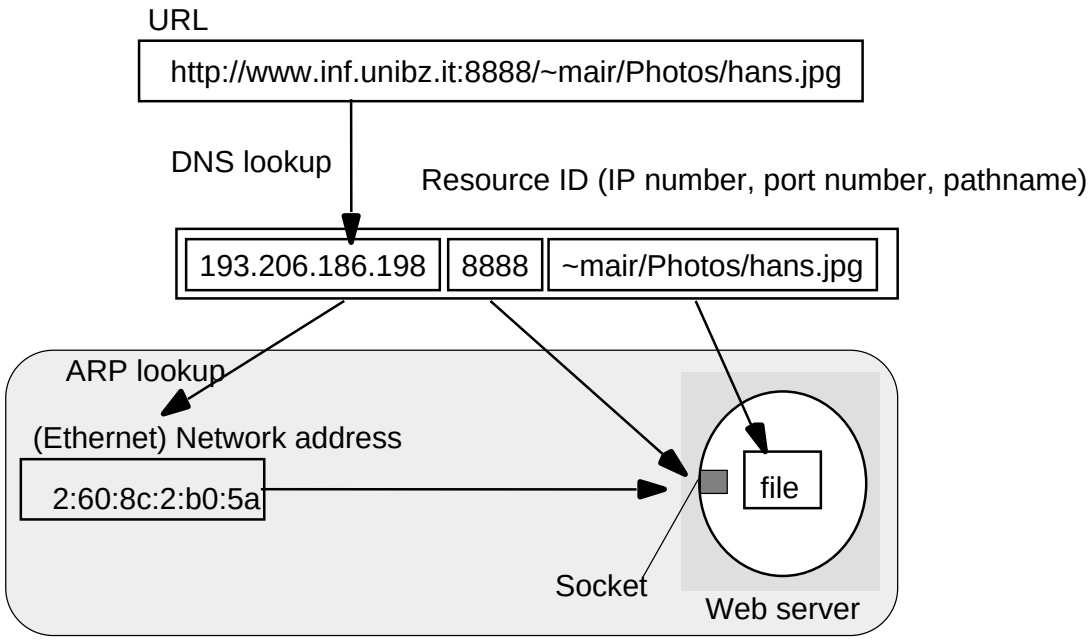
    multicast and broadcast addresses: IP Multicast, Ethernet
- Object identifiers
  - "pure" names (=bit patterns), usually numeric and large
    - never reused (include timestamp)
    - used for identification purposes

No real distinction between names and addresses.
Both must be looked up to obtain lower-level data (= name resolution).

Examples of Name Services

- File system
  - maps file name to file
- RMI registry
  - binds remote objects to symbolic names
- DNS (=Domain Name Service)
  - maps domain names to IP addresses
  - scalable, can handle change
- X.500/LDAP directory service
  - maps person's name to email address, phone number

Name Resolution on the WWW



Names and Resources

Currently, different name systems are used for each type of resource:

resource	*name*	*identifies*
file	pathname	file within a given file system
process	process id	process on a given computer
port	port number	IP port on a given computer

Uniform Resource Identifiers (URI) identify arbitrary resources:

- Uniform Resource Locator (URL): locates resource
  - typed by the scheme field (http, ftp, nfs, etc.)
  - part of the name is service-specific
  - resources cannot be moved between domains
- Uniform Resource Name (URN): names resource

Name Spaces

- Name space = collection of all valid names recognised by a service with
  - a syntax for specifying names, and
  - rules for resolving names (e.g., left to right)
- Naming context = maps a name to primitive attributes directly, or to another context and derived name (usually by prefixing)
  - telephone no: country, area, number
  - Internet host names: contexts = domains
  - Unix file system: contexts = directories

Name Spaces (cntd)

- Binding
  - associating a name to an object
  - binding names to attributes, one of which may be address
- Naming domain
  - has an authority that assigns names to objects within a name space or context
    - sysadmin assigns login names
    - Host names are assigned in a domain
  - object may be registered more than once within context
- Multiple names
  - alias (alternative name for an object, e.g. www, ftp, etc.)
  - symbolic name (alternative name which maps to a path name in the name space, e.g., symbolic link for file)

Hierarchic Name Spaces

- Sequence of name tokens resolved in different context
  - syntax: name token (text string) + delimiter
  - DNS: inf.unibz.it
  - Unix: /usr/bin
- Name structure reflects organisational structure
  - name changes if object migrates
  - names can be used relative to context or absolute
  - local contexts managed in a distributed fashion
- Examples
  - domain names, Unix file system

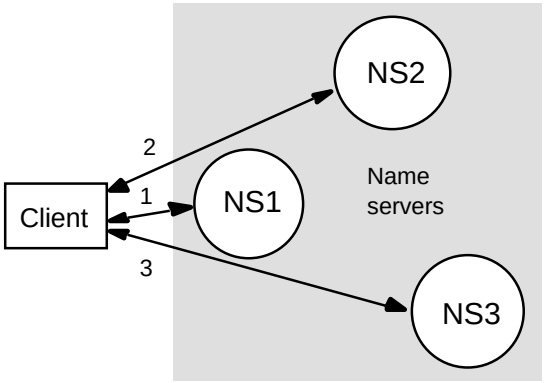
Flat Name Spaces

- Single global context and naming authority for all names
  - computer serial number
  - Ethernet address
  - remote object reference
    (IP address, port, time, object number, interface id)
- Names not meaningful
  - difficult to resolve (no tree hierarchy)
  - easy to create
  - easy to ensure uniqueness (timestamps)

Name Resolution

- Iteratively present name to a naming context
  - start with initial naming context
  - repeat as long as contexts + derived names are returned
  - aliases can introduce cycles
    (abandon after threshold no of resolutions or ensure no cycles)
- Replication/Caching
  - used for improved fault-tolerance on large services
    (more than one server, cf. DNS)
- Navigation
  - organising the access to several servers

Iterative Navigation

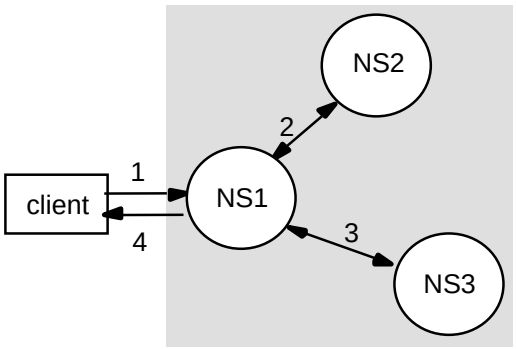


e.g., in DNS

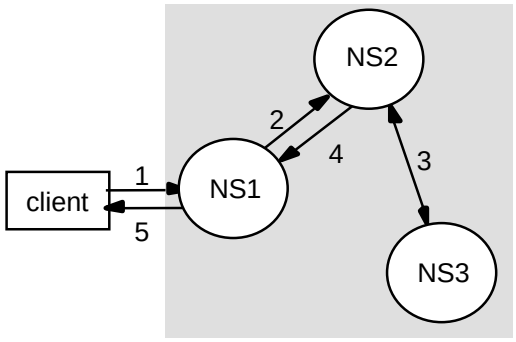
- The database is distributed over servers for different domains
- A client contacts servers NS1–NS3 one after the other in order to resolve a name
- Server returns attributes if it knows the name, otherwise suggests another server

Server-controlled Navigation

Name server communicates with other name servers on the client's behalf



Non-recursive
server-controlled



Recursive
server-controlled

In DNS, iterative navigation is the standard.
Recursive navigation is an option that is necessary in domains that limit client access to their DNS information for security reasons

Replication and Caching

Replicate some directories for performance and availability.

- Updates
  - Approach 1: write to single master, master propagates updates
  - Approach 2: write to any replica, later merge updates (timestamps)
  - Result: weak consistency (some entries out of date)
- Look-ups
  - try any local server, then go to root and down the tree
- Caching
  - names and addresses of recently used objects

Internet Domain Name System (DNS)

Maps host names to IP addresses (basically)

Design dates back to 1987 (Mockapetris)

Before

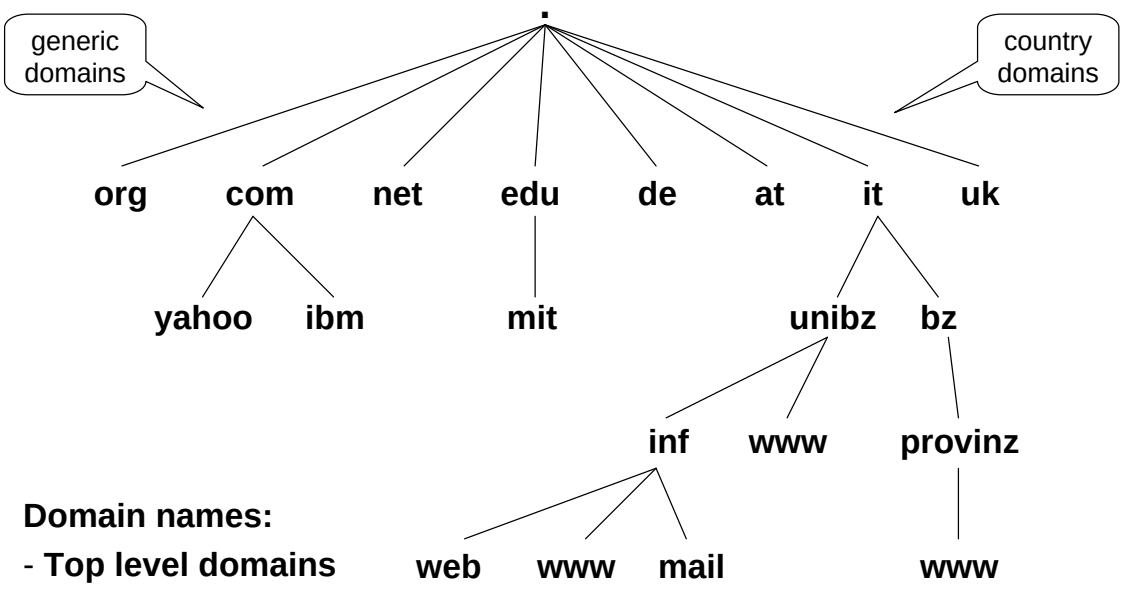
- all host names and addresses in one large master file
- stored on one central host
- downloaded by computers that needed to resolve names

What were the drawbacks of that approach?

Internet Domain Name System (cntd)

- Distributed naming database
- Hierarchical name structure reflects administrative structure of the Internet
- Rapidly resolves domain names to IP addresses
  - exploits caching heavily
  - typical query time ~100 milliseconds
- Scales to millions of computers
  - partitioned database
  - caching
- Resilient to failure of a server
  - replication (e.g., 13 root servers, 6 servers for .it, etc.)

The DNS Name Space



Domain names:

- **Top level domains**
- **2nd level domains**
- **…**

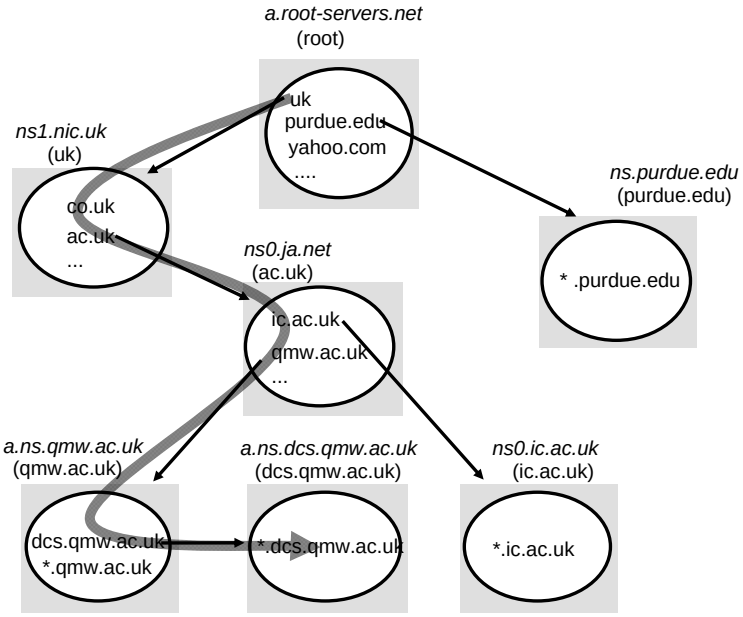
Computer names

DNS Server Functions

- Main function:
  - resolves domain names for computers, i.e. gets their IP addresses
  - caches the results of previous searches until they pass their “time to live”
- Info offered:
  - host IP addresses and canonical names
  - name servers for a domain
  - mail exchangers for a domain
  - host information - type of hardware and OS
  - well-known services - a list of well-known services offered by a host
- Other functions:
  - reverse resolution - get domain name from IP address

Example: DNS Servers

Look up IP-address of
`www.dcs.qmw.ac.uk`



- Name server names are in *italics*
- (Corresponding domains are in parentheses)
- ⟶ denotes a name server entry

DNS Servers and Zones

- The DNS namespace consists of zones:
  - zone = domain minus sub-domains, administered independently
- Every zone must have at least two name servers
  - exactly one master (= primary) server: contains the only writable copy of the “zone file”
  - one or more secondary (= slave) servers: copies its zone file from the master
  - both, master and slaves, are “authoritative” for the zone
  - set up should guarantee that slaves never hold information that is out of date

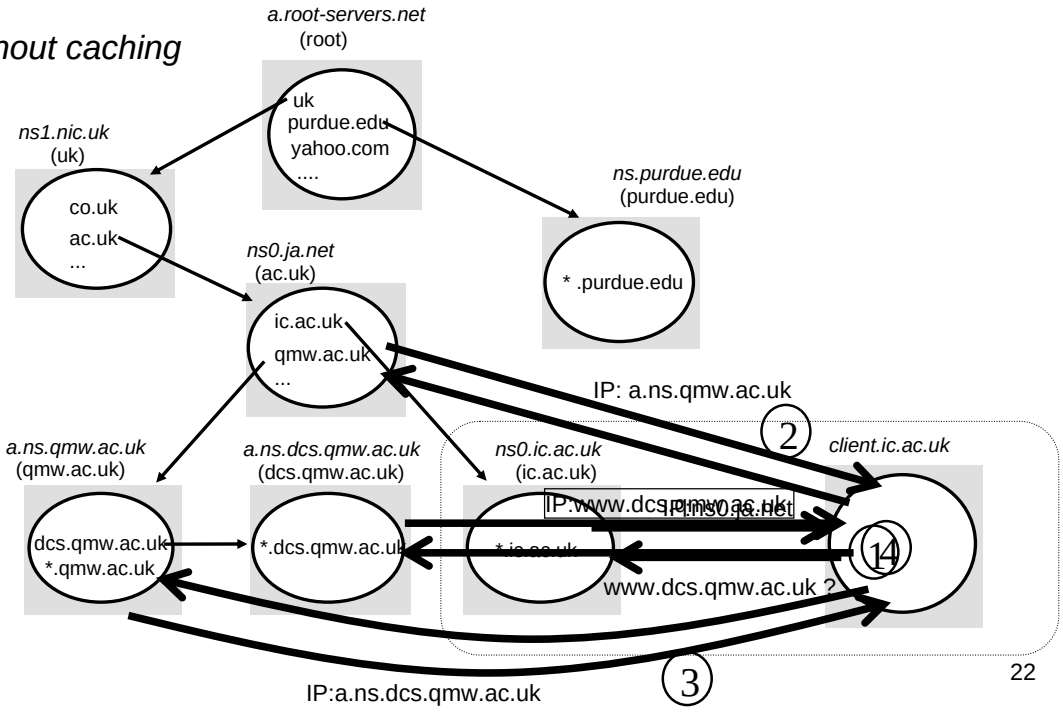
DNS Name Resolution

Basic algorithm

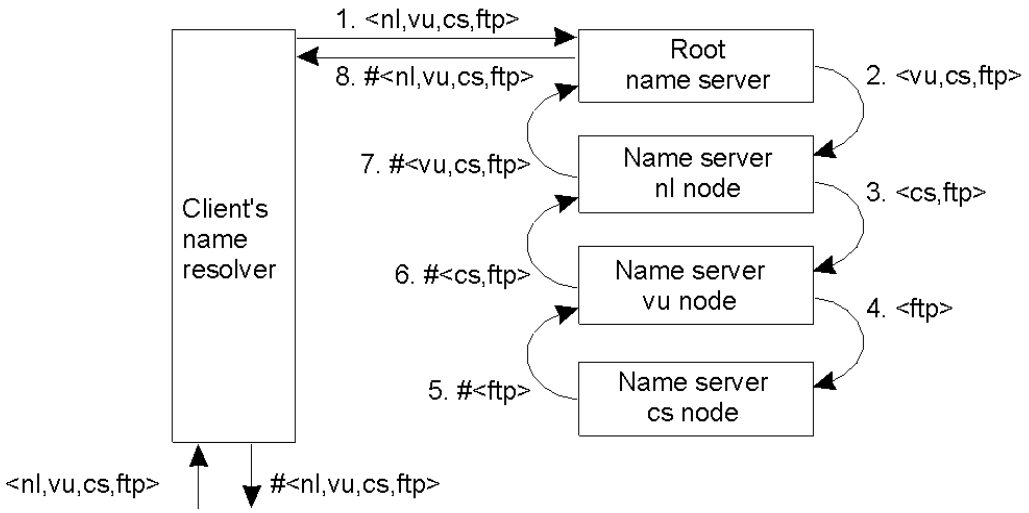
- Look for the name in the local cache
- Try a superior DNS server, which responds with:
  - another recommended DNS server
  - the IP address (which may not be entirely up to date)

DNS Iteration

Without caching



Recursive Name Resolution in DNS



Types of 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

DNS Serves organize their info in "resource records"

Name Server Content

An excerpt from the DNS database for the zone *cs.vu.nl*.

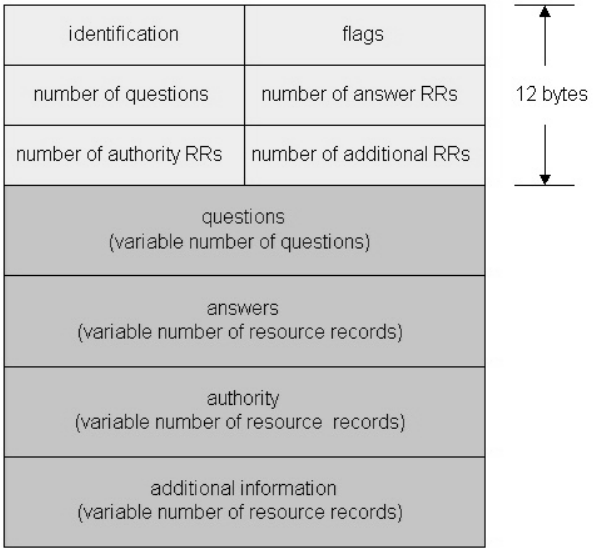
Name	Record type	Record value
cs.vu.nl	SOA	star (1999121502,7200,3600,2419200,86400)
cs.vu.nl	NS	star.cs.vu.nl
cs.vu.nl	NS	top.cs.vu.nl
cs.vu.nl	NS	solo.cs.vu.nl
cs.vu.nl	TXT	"Vrije Universiteit - Math. & Comp. Sc."
cs.vu.nl	MX	1 zephyr.cs.vu.nl
cs.vu.nl	MX	2 tornado.cs.vu.nl
cs.vu.nl	MX	3 star.cs.vu.nl
star.cs.vu.nl	HINFO	Sun Unix
star.cs.vu.nl	MX	1 star.cs.vu.nl
star.cs.vu.nl	MX	10 zephyr.cs.vu.nl
star.cs.vu.nl	A	130.37.24.6
star.cs.vu.nl	A	192.31.231.42
zephyr.cs.vu.nl	HINFO	Sun Unix
zephyr.cs.vu.nl	MX	1 zephyr.cs.vu.nl
zephyr.cs.vu.nl	MX	2 tornado.cs.vu.nl
zephyr.cs.vu.nl	A	192.31.231.66
www.cs.vu.nl	CNAME	soling.cs.vu.nl
ftp.cs.vu.nl	CNAME	soling.cs.vu.nl
soling.cs.vu.nl	HINFO	Sun Unix
soling.cs.vu.nl	MX	1 soling.cs.vu.nl
soling.cs.vu.nl	MX	10 zephyr.cs.vu.nl
soling.cs.vu.nl	A	130.37.24.11
laser.cs.vu.nl	HINFO	PC MS-DOS
laser.cs.vu.nl	A	130.37.30.32
vucs-das.cs.vu.nl	PTR	0.26.37.130.in-addr.arpa
vucs-das.cs.vu.nl	A	130.37.26.0

DNS Message Format

Queries and replies have the same format (using UDP)

Header

- identification: 16 bit number set in query, matching reply with same number
- flags: 1 bit each, e.g.,
  - query or reply
  - authoritative answer
  - recursion desired
  - recursion available



DNS Message Format (cntd)

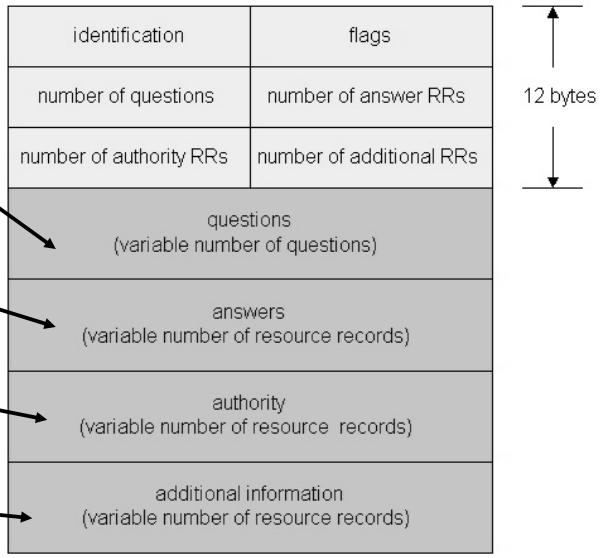
The message body consists of resource records

Domain names (or IP adds), type of records requested *(incomplete records)*

Resource records answering the query

Records pointing to authoritative servers

Typically, address records of the authoritative servers



Implementations of DNS

- De facto standard for UNIX is BIND (= Berkeley Internet Name Domain)
  - Client programs acting as resolver link in library software (i.e., no process on client)
  - Server machines run a daemon (“named”)
  - Server can be configured as one of three categories:
    - primary, secondary, caching-only
- Microsoft’s Active Directory supports DNS

Access to DNS

- **`host`**
  - command for name resolution and reverse resolution
- **`nslookup`**
  - command/tool to query DNS servers for arbitrary info
- **`dig`**
  - similar to nslookup, without some of the deficiencies of the former

- /etc/resolv.conf
  - file containing IP address of default name server

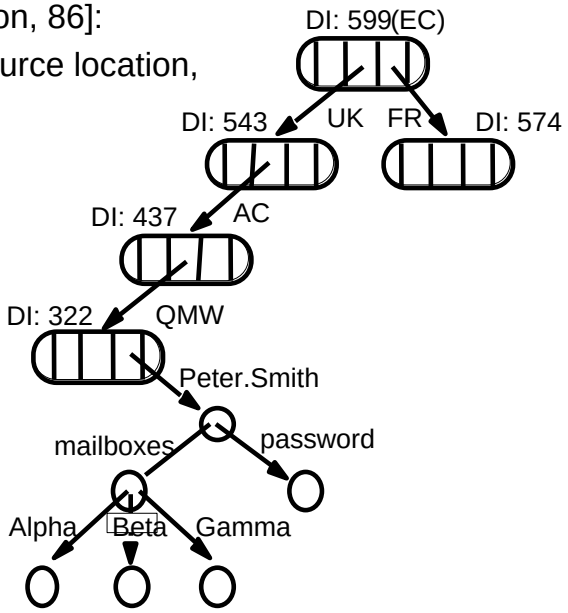
- Java JNDI (= Java Naming and Directory Interface)
  - provides interface for querying DNS

Global Name Service (GNS)

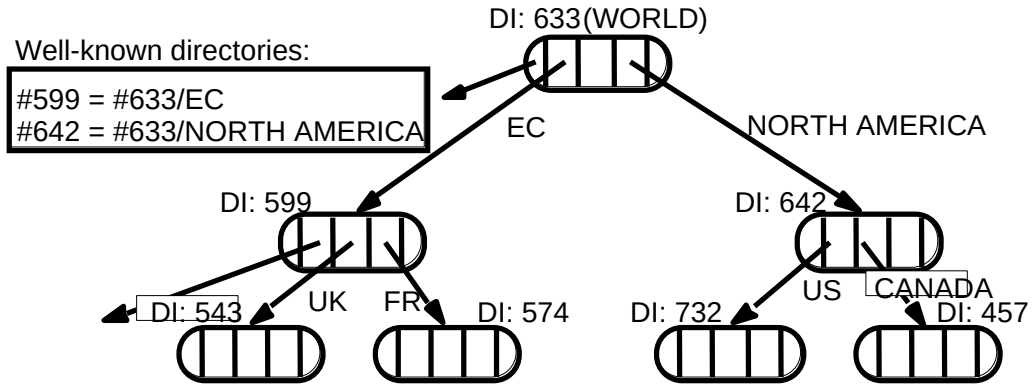
A proposal from research [B. Lampson, 86]:
GNS is more flexible system for resource location, mail addressing and authentication

- Structured leafs: "Value trees"
- Directory nodes have a unique directory identifier ID
- Names in GNS have two parts
  <directory name, value name>

GNS accommodates change:
use directory identifiers
to identify old roots

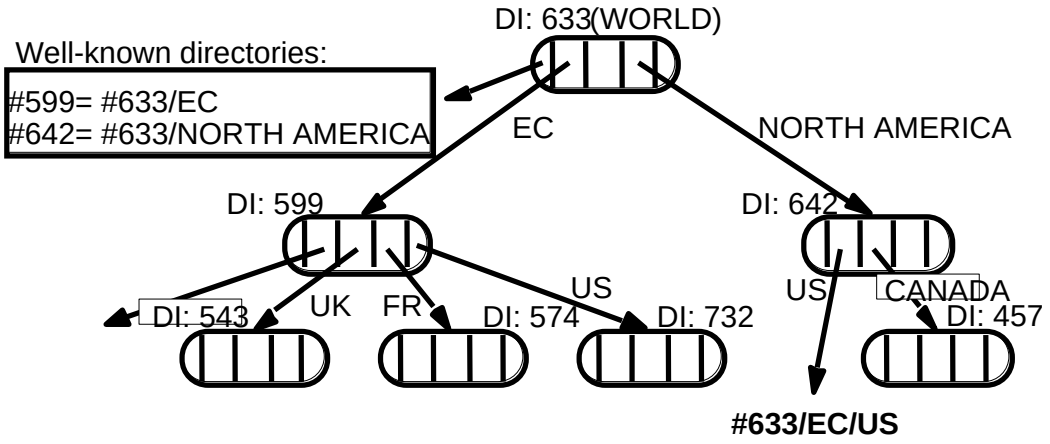


Merging Trees Under a New Root



Old “working roots” (like #599 (EC)) can be found in the new tree, using the “well-known” directories table of #633 (WORLD)

Restructuring the Directory



- The US becomes part of the EU:
  a symbolic link (#633/EC/US) points to the new location

GNS gains flexibility at the cost of accumulating additional data after reconfiguration