Parallelism

Master 1 International

Andrea G. B. Tettamanzi

Université de Nice Sophia Antipolis

Département Informatique

andrea.tettamanzi@unice.fr

Lecture 3 – Part a

Naming

Names

- Bit or character string → entity
- Entity:
  - Resource (host, printer, disk, file, …)
  - Process
  - User
  - Mailbox, web page, window, etc.

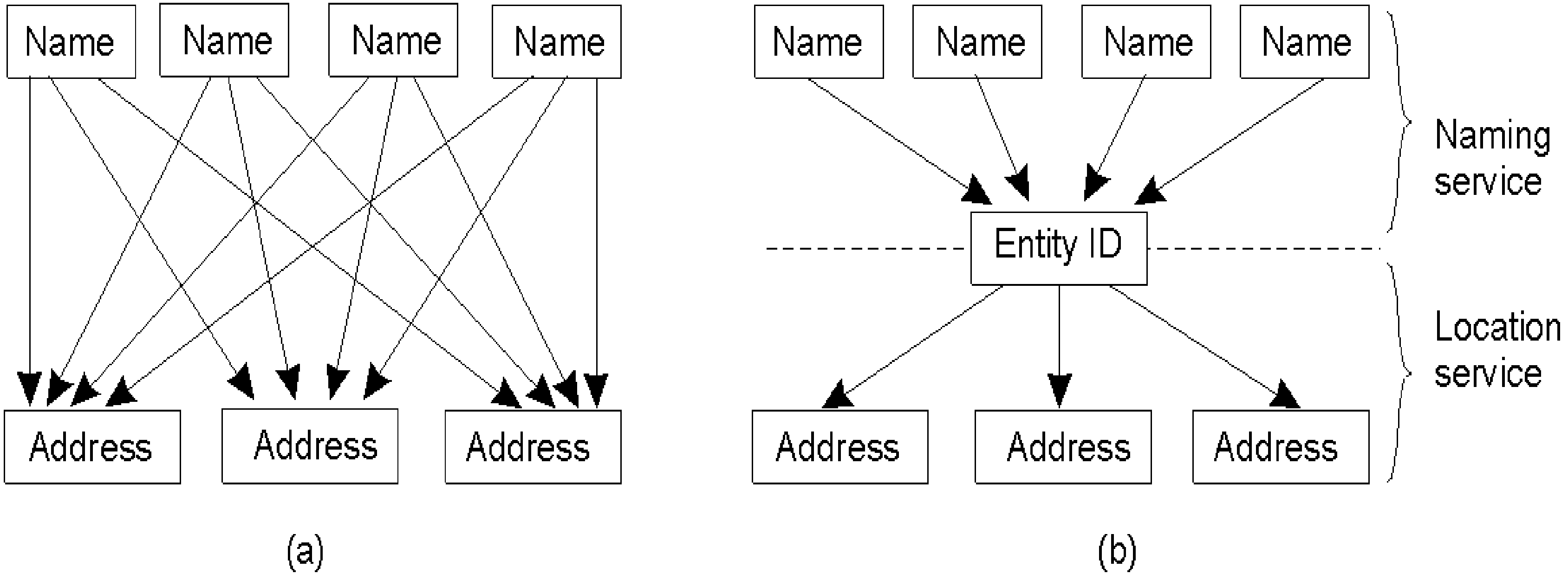
Addresses

- Every entity has one or more **access points**
- Address = name of an access point
- Access points are not fixed
- Entity ↔ Address?

Identifier

- A special kind of name
- Satisfies the following properties:
  - An identifier refers to at most one entity
  - Every entity has at most one identifier
  - Any given identifier always refers to the same entity (= not reused)

Naming versus Locating Entities



a) Direct, single level mapping between names and addresses.
b) Three-level mapping using identities.

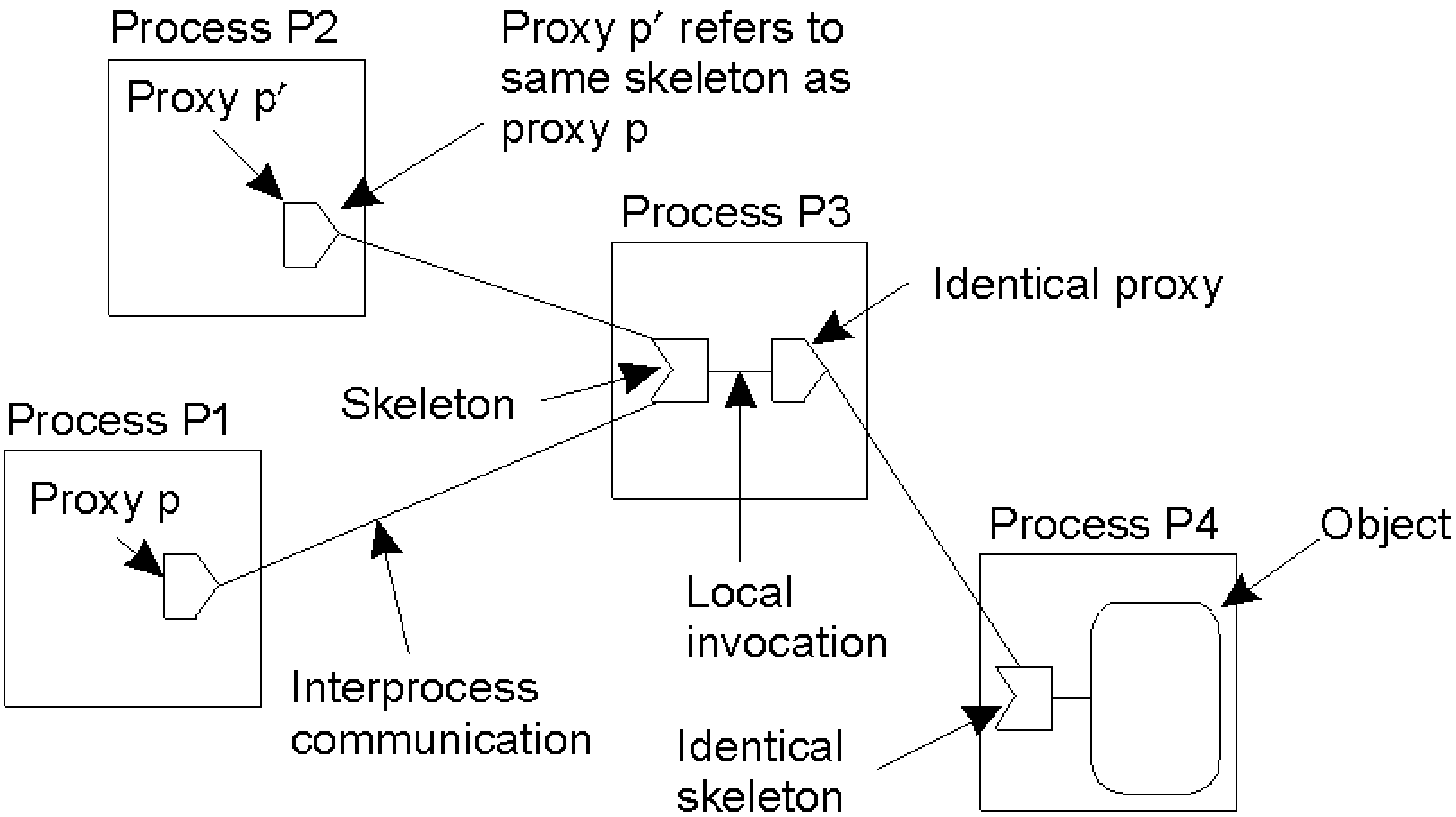
Flat naming

- Unstructured names

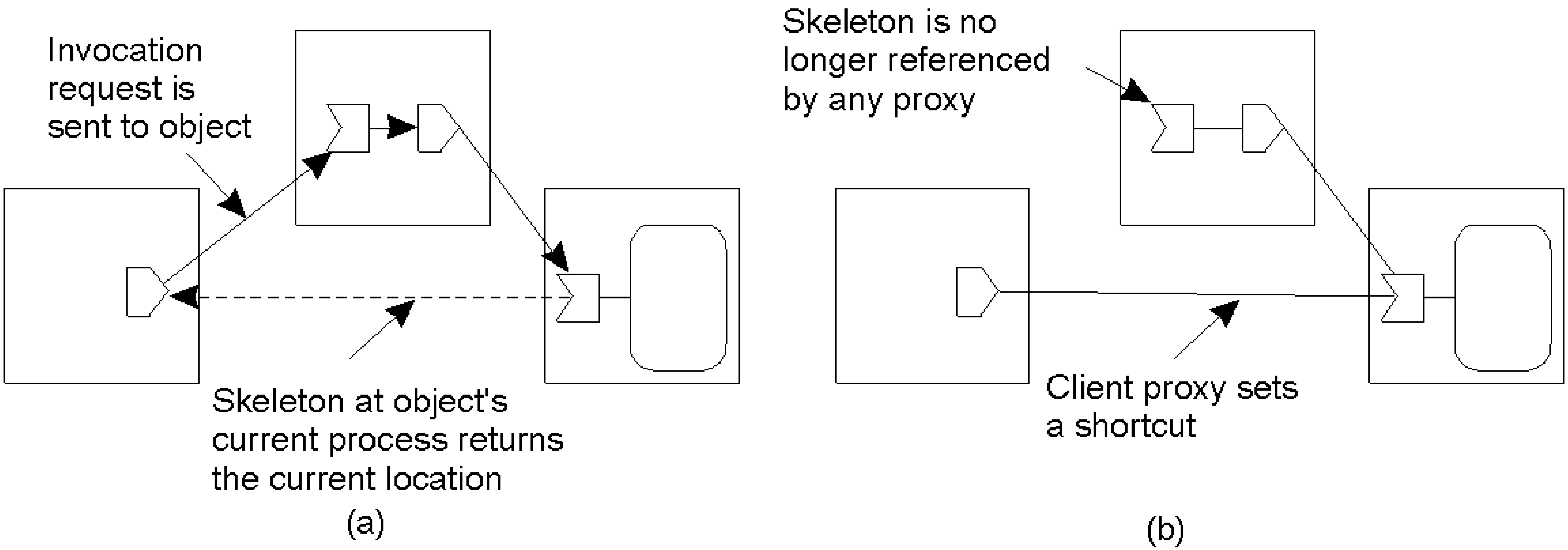
- Problem: locating mobile entities

- Simple solutions:
  - Broadcast (e.g., Address resolution protocol, ARP)
  - Multicast

Forwarding Pointers (1)

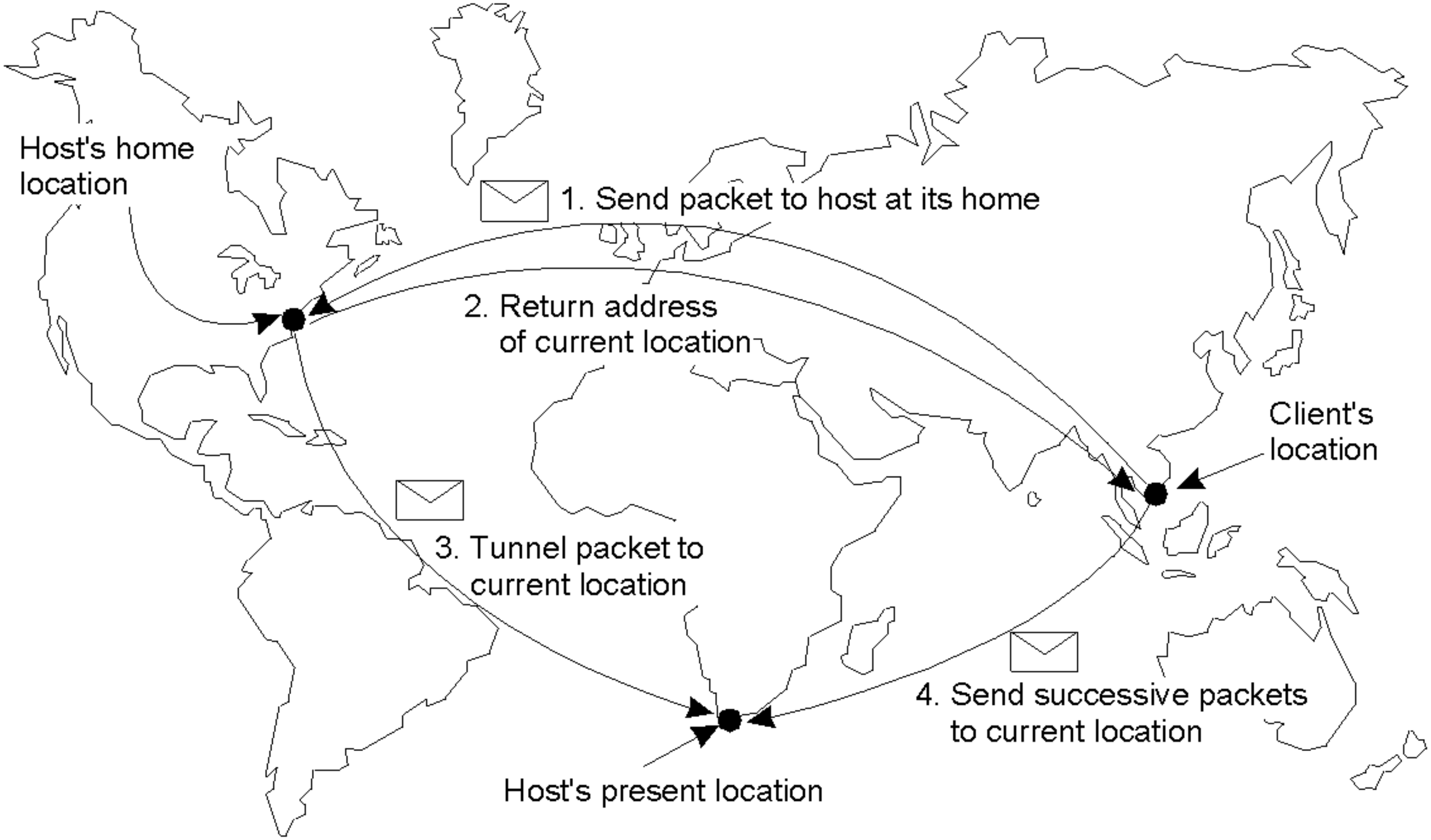


Forwarding Pointers (2)

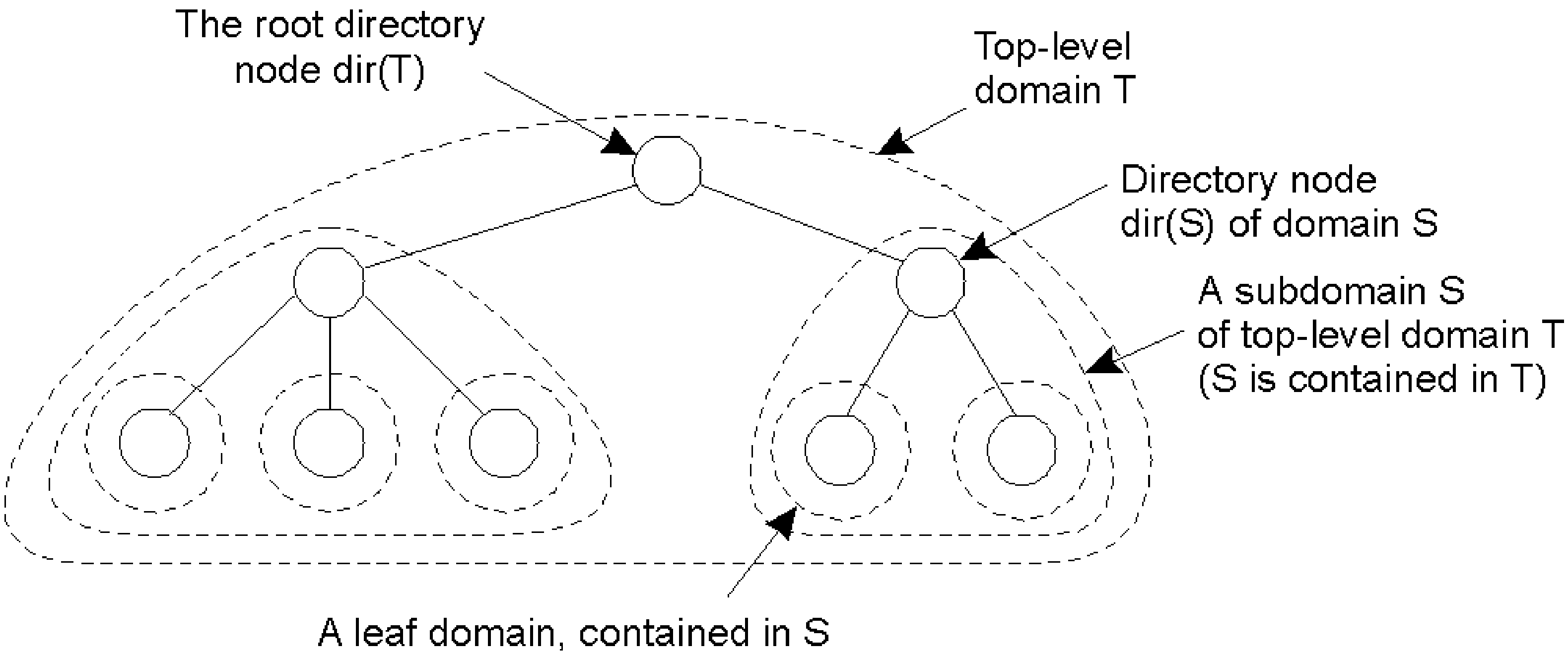

Invocation request is sent to object
Skeleton at object's current process returns the current location
(a)
Skeleton is no longer referenced by any proxy
Client proxy sets a shortcut
(b)

Redirecting a forwarding pointer, by storing a shortcut in a proxy.

Home-Based Approaches

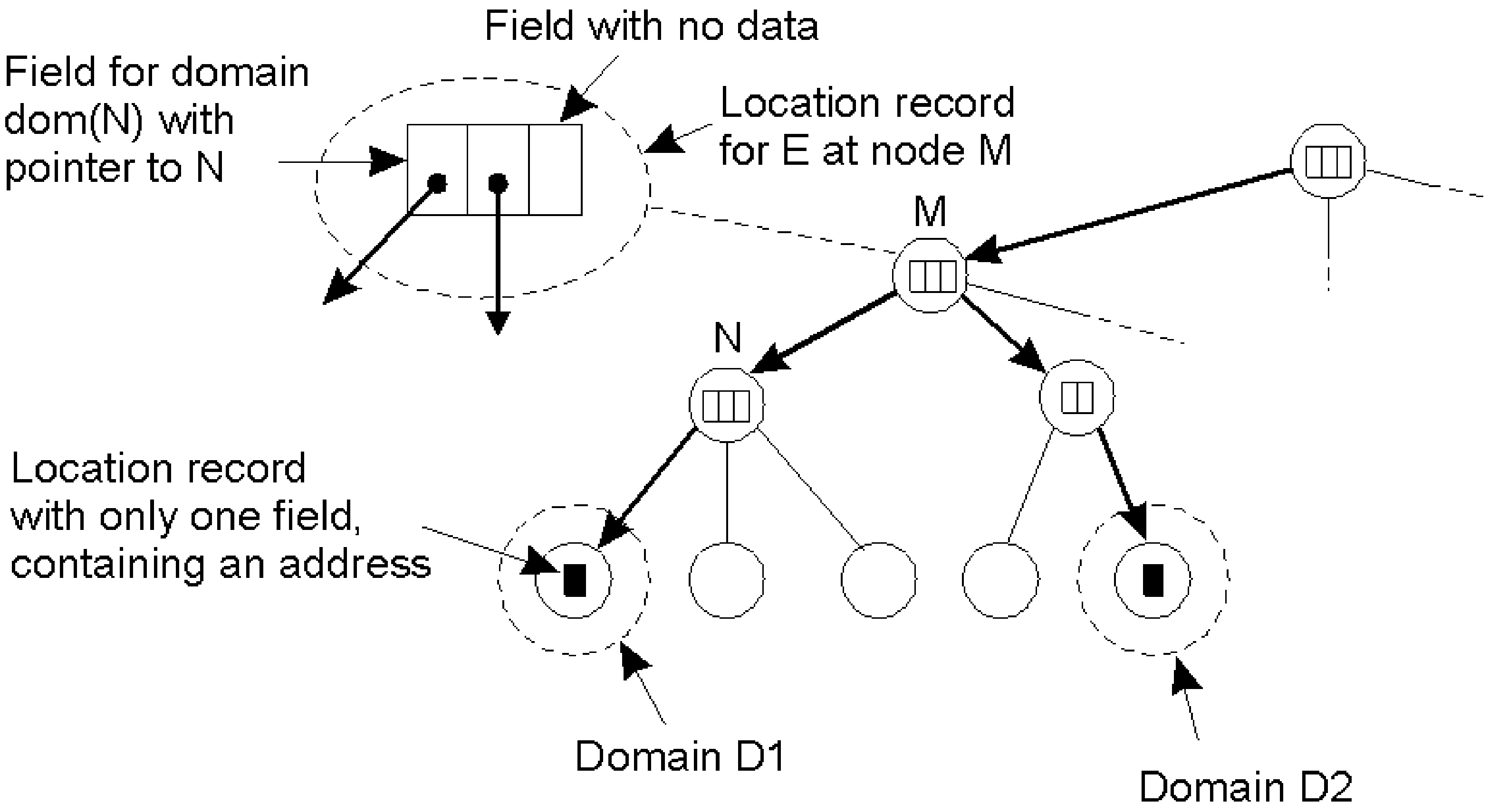

Host's home location
1. Send packet to host at its home
2. Return address of current location
3. Tunnel packet to current location
4. Send successive packets to current location
Client's location
Host's present location

Hierarchical Approaches (1)


The root directory node dir(T)
Top-level domain T
Directory node dir(S) of domain S
A subdomain S of top-level domain T (S is contained in T)
A leaf domain, contained in S

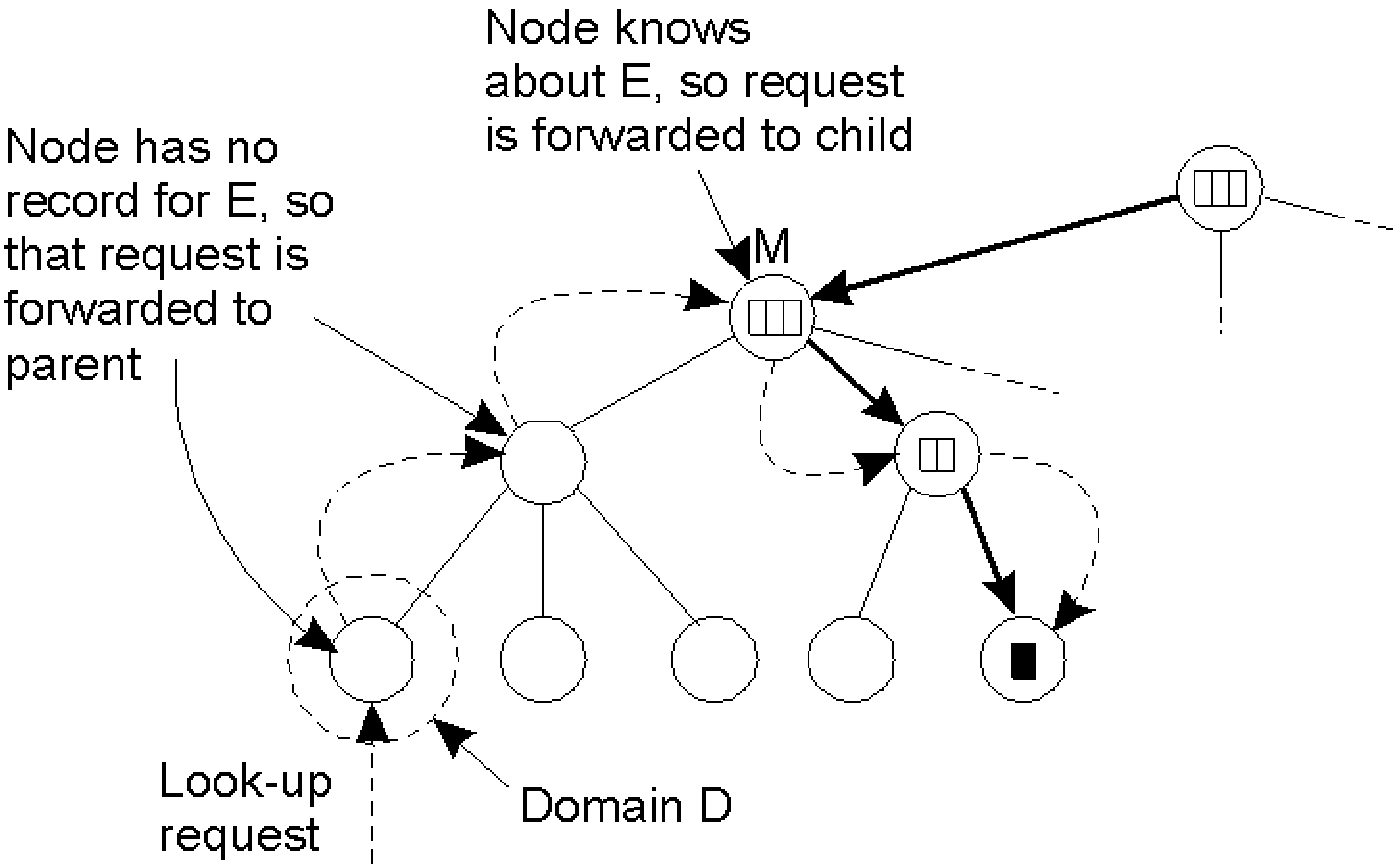
Hierarchical organization of a location service into domains, each having an associated directory node.

Hierarchical Approaches (2)


Field with no data
Field for domain dom(N) with pointer to N
Location record for E at node M
M
N
Location record with only one field, containing an address
Domain D1
Domain D2

An example of storing information of an entity having two addresses in different leaf domains.

Hierarchical Approaches (3)



Node knows about E, so request is forwarded to child

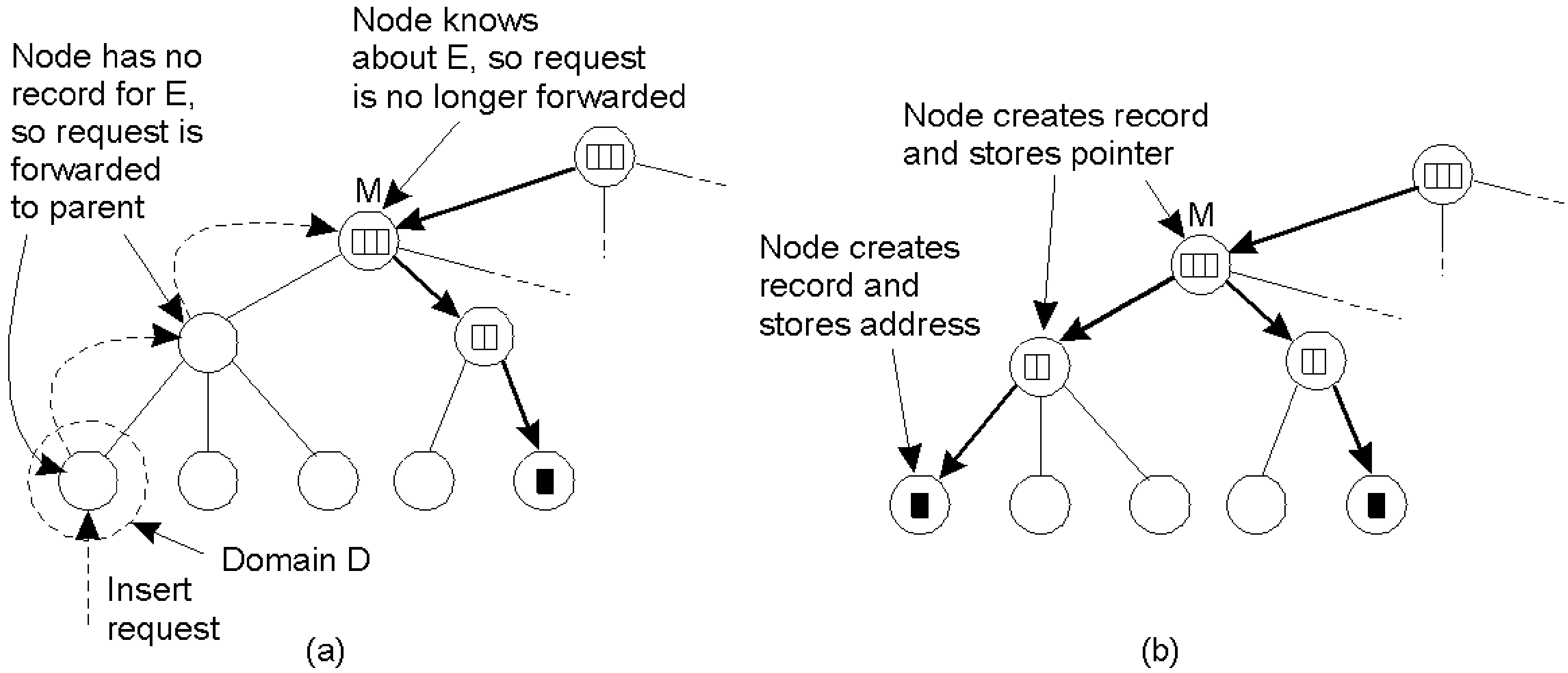
Node has no record for E, so that request is forwarded to parent

Look-up request

Domain D

Looking up a location in a hierarchically organized location service.

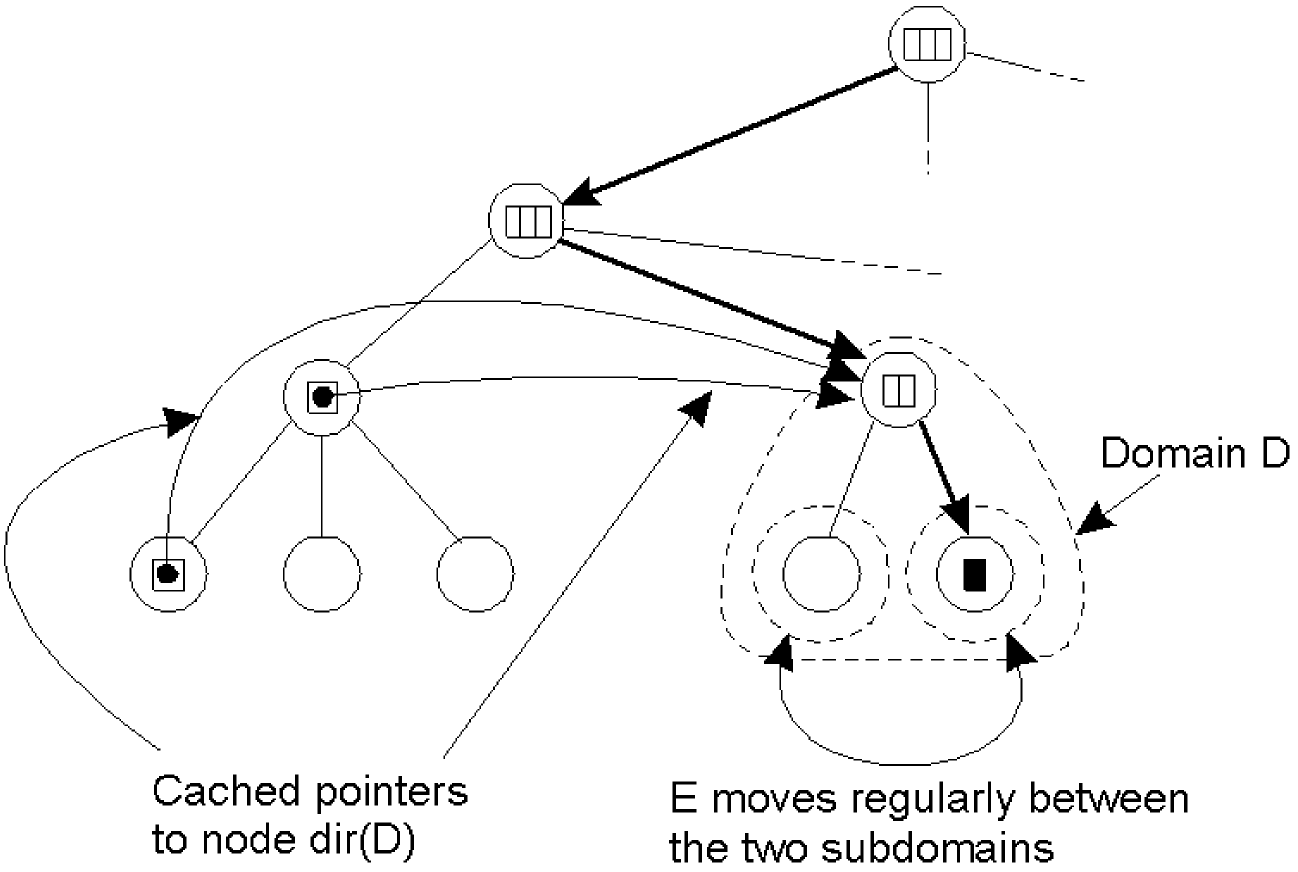
Hierarchical Approaches (4)


Node has no record for E, so request is forwarded to parent
Node knows about E, so request is no longer forwarded
M
Domain D
Insert request
(a)
Node creates record and stores pointer
Node creates record and stores address
M
(b)

a) An insert request is forwarded to the first node that knows about entity *E*.

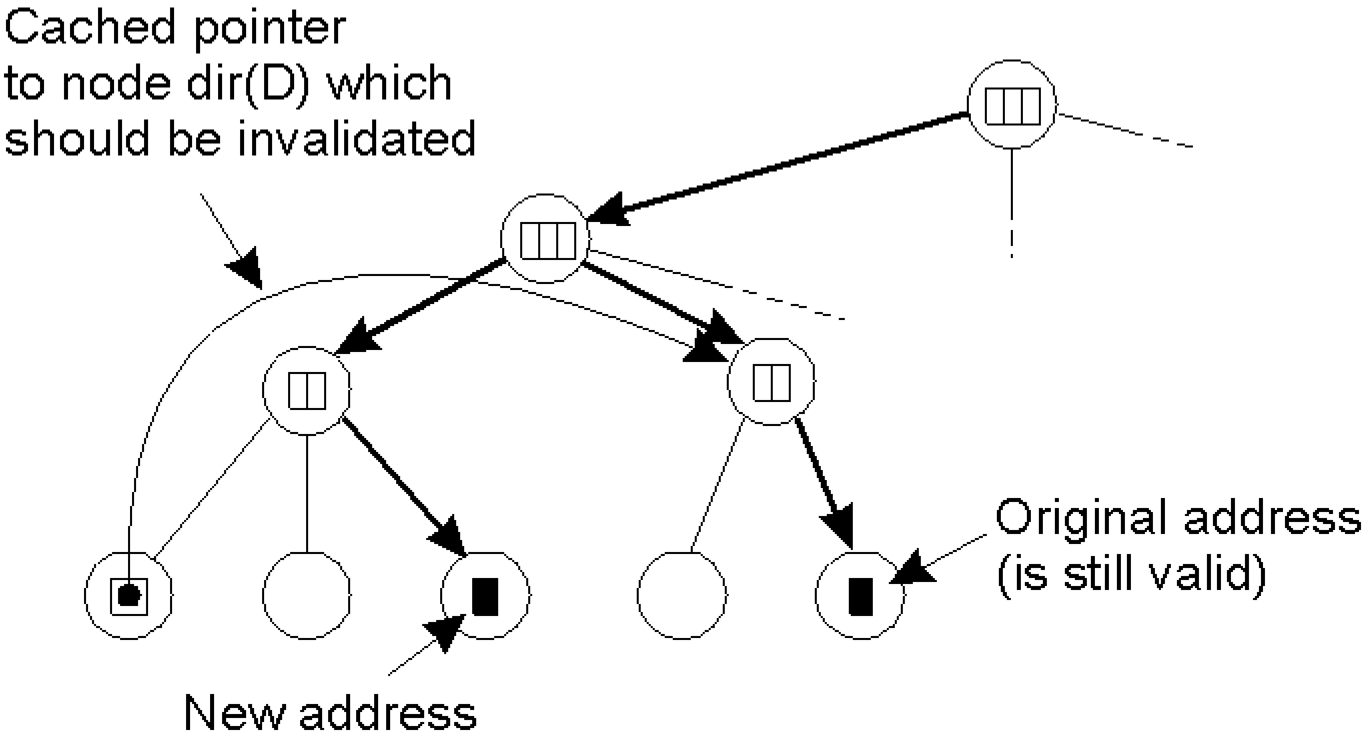
b) A chain of forwarding pointers to the leaf node is created.

Pointer Caches (1)



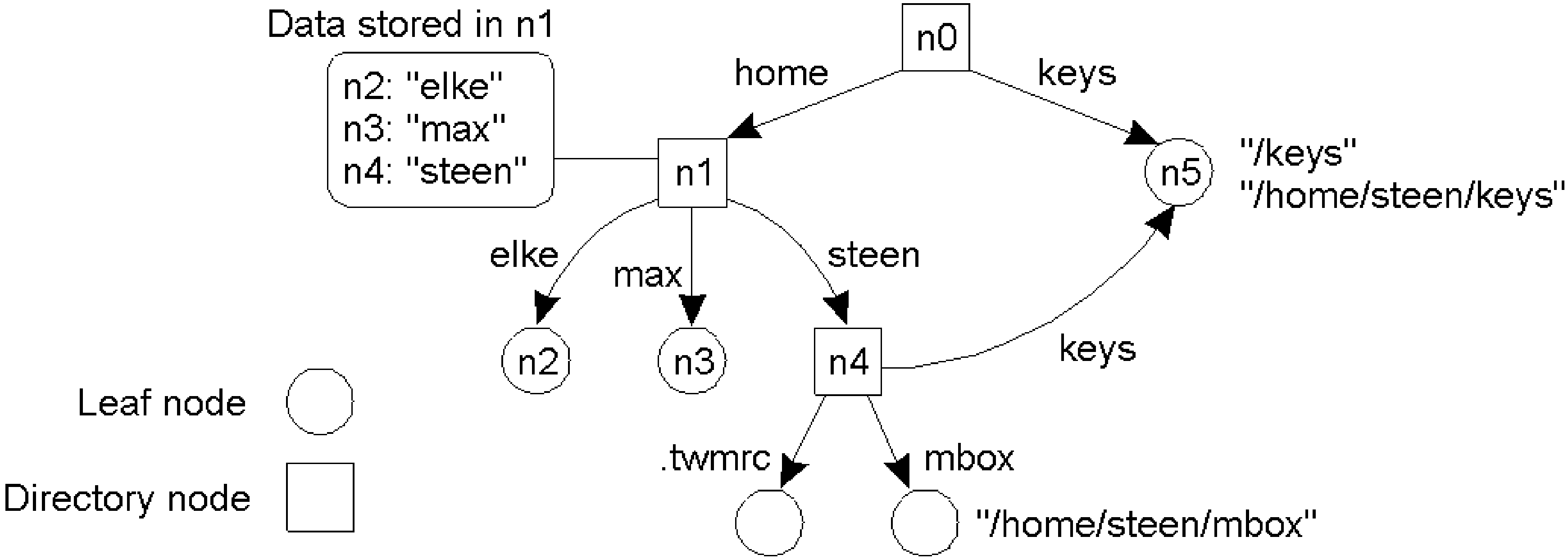
Caching a reference to a directory node of the lowest-level domain in which an entity will reside most of the time.

Pointer Caches (2)



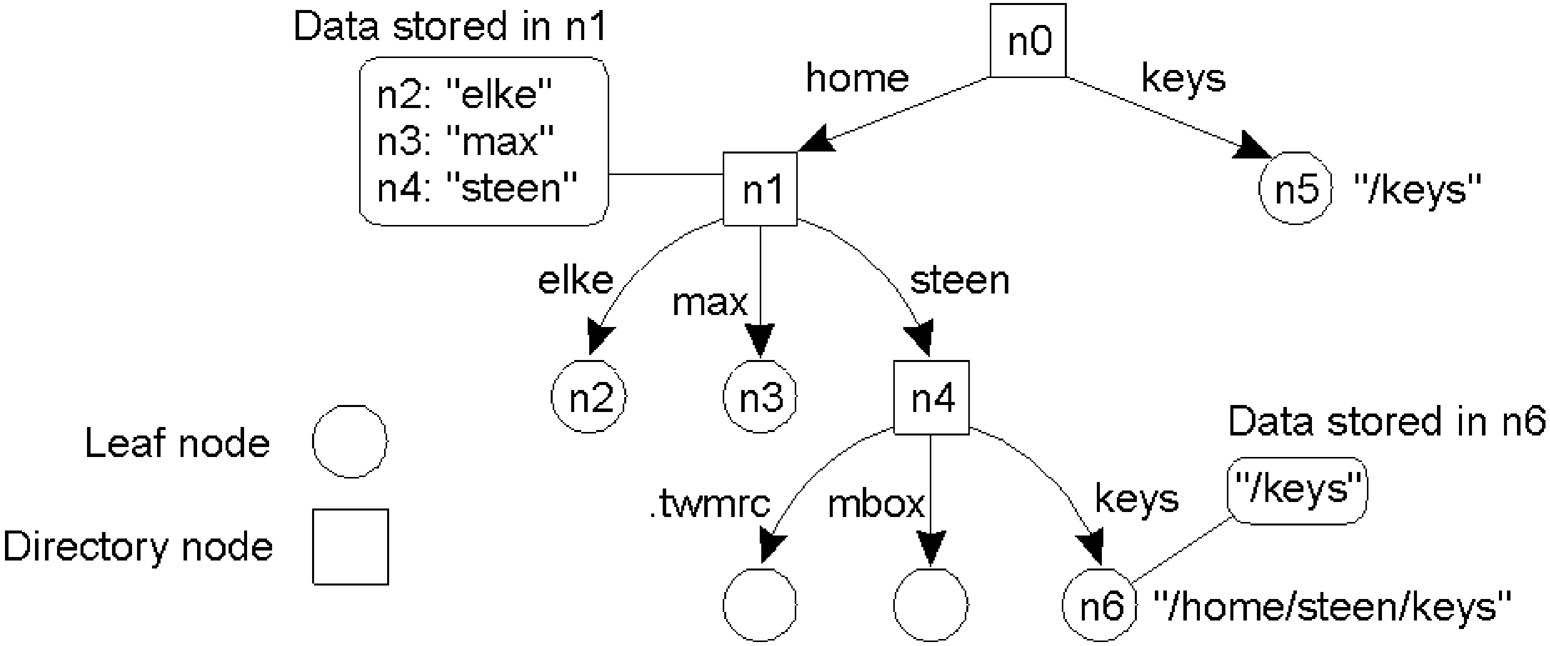
A cache entry that needs to be invalidated because it returns a nonlocal address, while such an address is available.

Name Spaces (1)



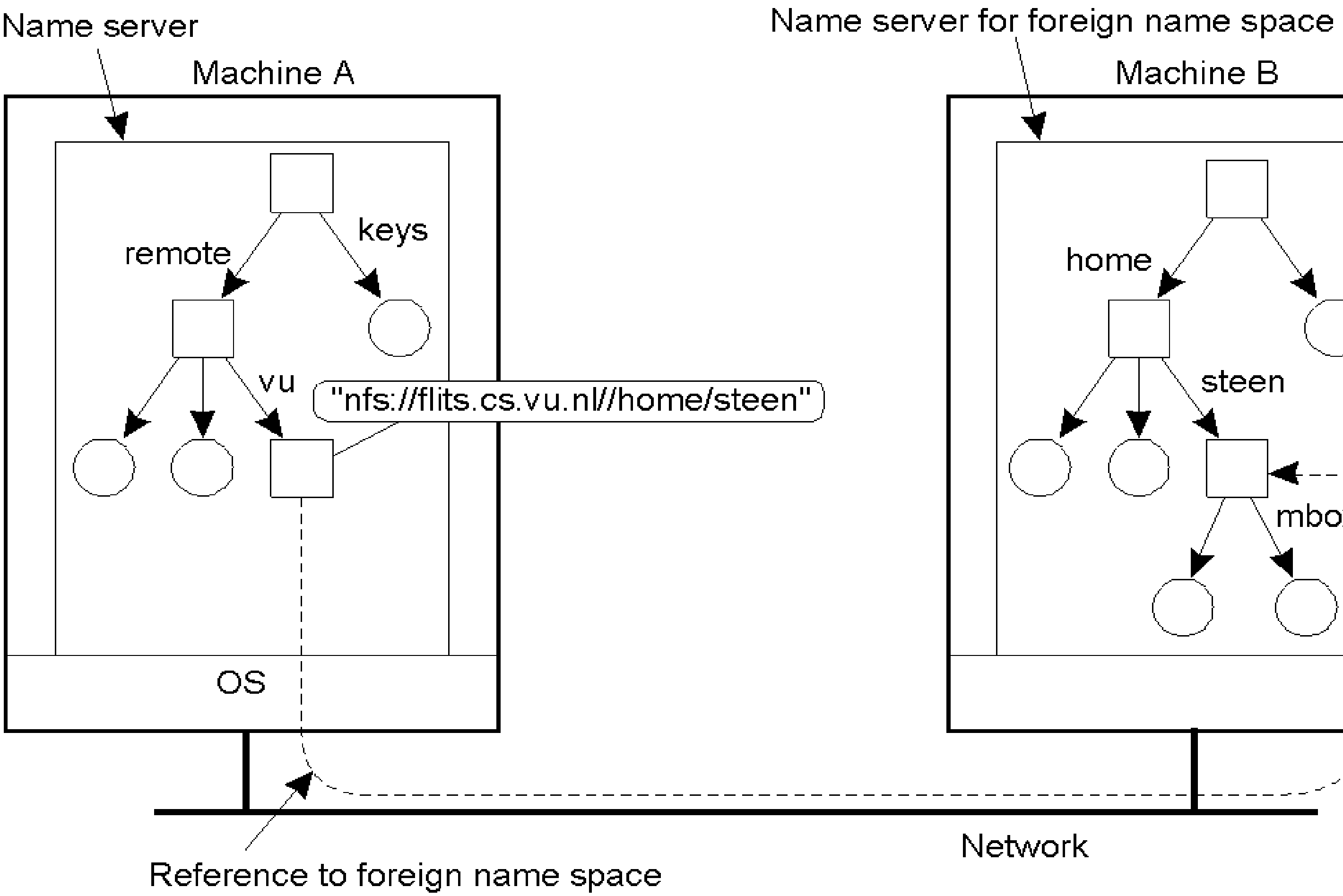
A general naming graph with a single root node.

Linking and Mounting (1)



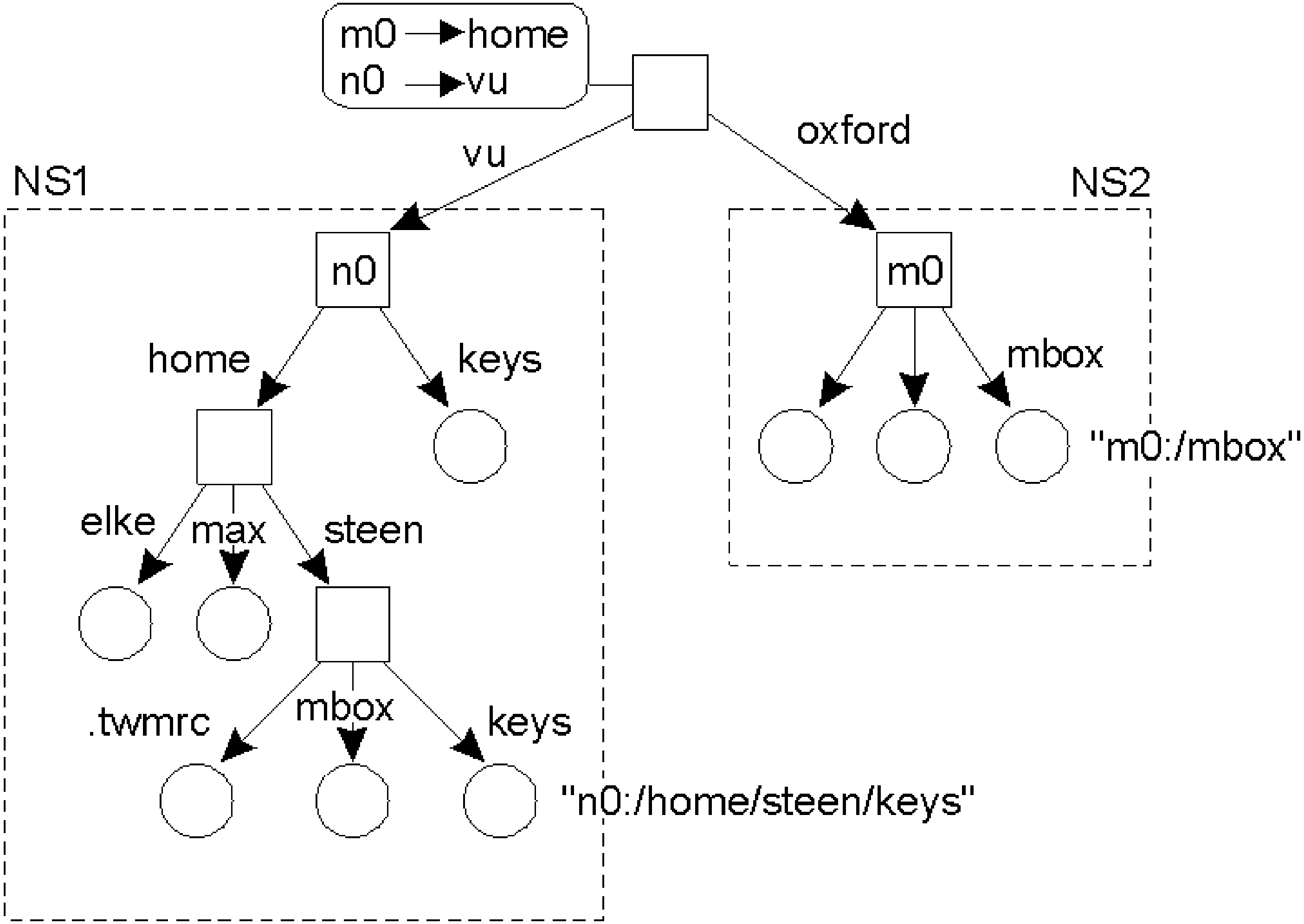
The concept of a symbolic link explained in a naming graph.

Linking and Mounting (2)


Name server
Machine A
Name server for foreign name space
Machine B
keys
remote
vu
"nfs://flits.cs.vu.nl//home/steen"
home
steen
mbox
OS
Network
Reference to foreign name space

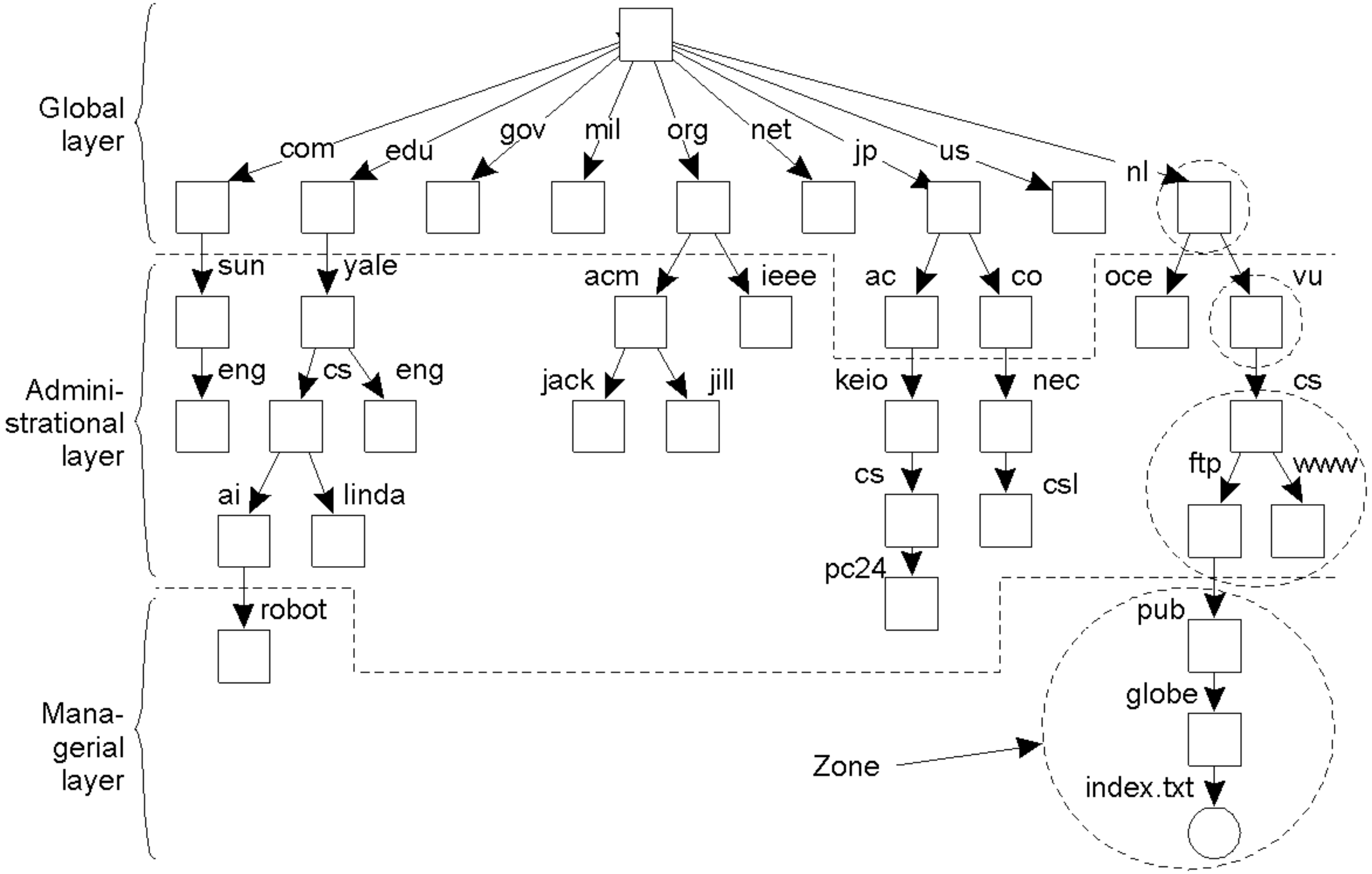
Mounting remote name spaces.

Linking and Mounting (3)



Organization of the DEC Global Name Service

Name Space Distribution (1)



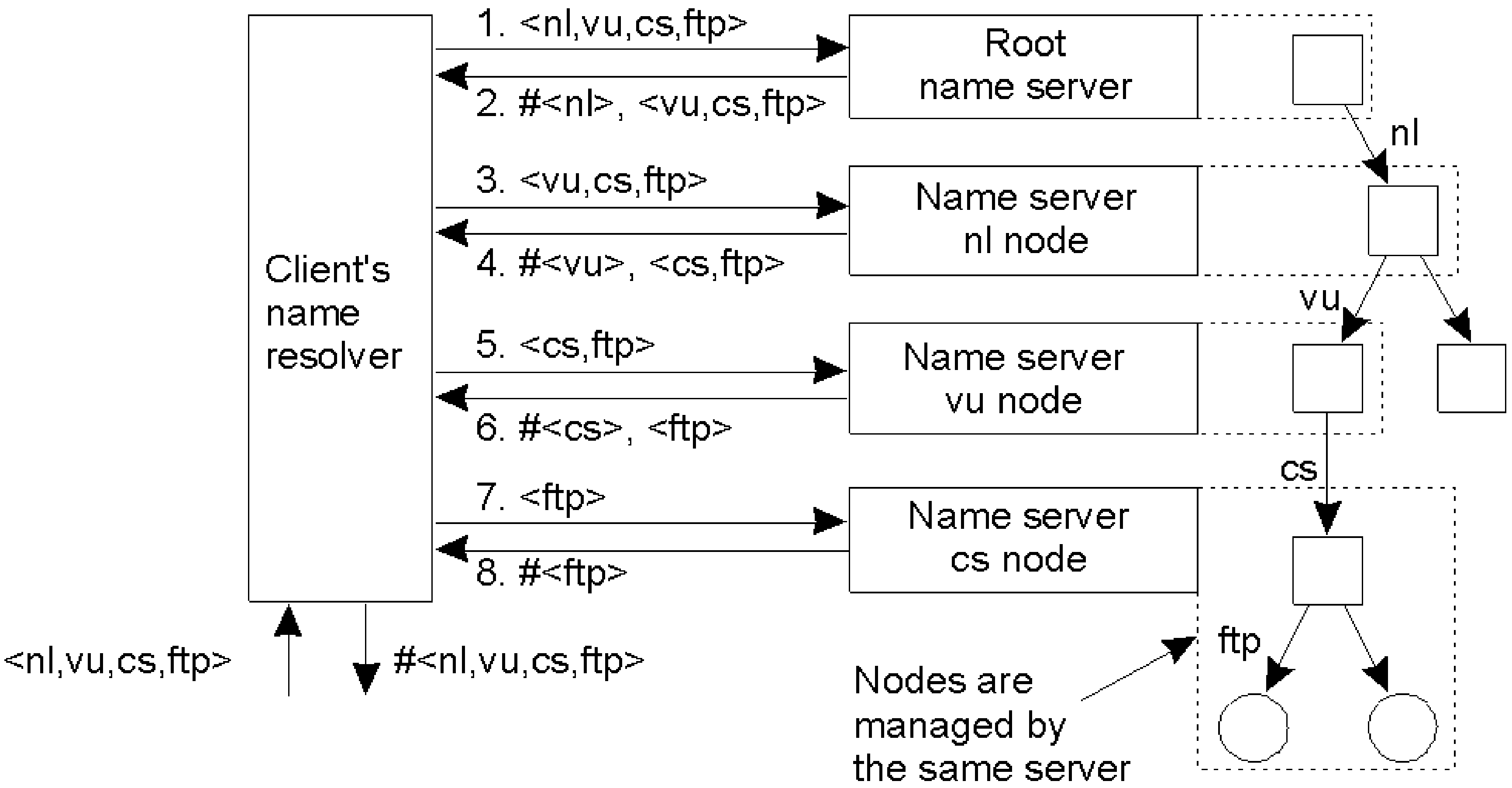
An example partitioning of the DNS name space, including Internet-accessible files, into three layers.

Name Space Distribution (2)

Item	Global	Administrational	Managerial
Geographical scale of network	Worldwide	Organization	Department
Total number of nodes	Few	Many	Vast numbers
Responsiveness to lookups	Seconds	Milliseconds	Immediate
Update propagation	Lazy	Immediate	Immediate
Number of replicas	Many	None or few	None
Is client-side caching applied?	Yes	Yes	Sometimes

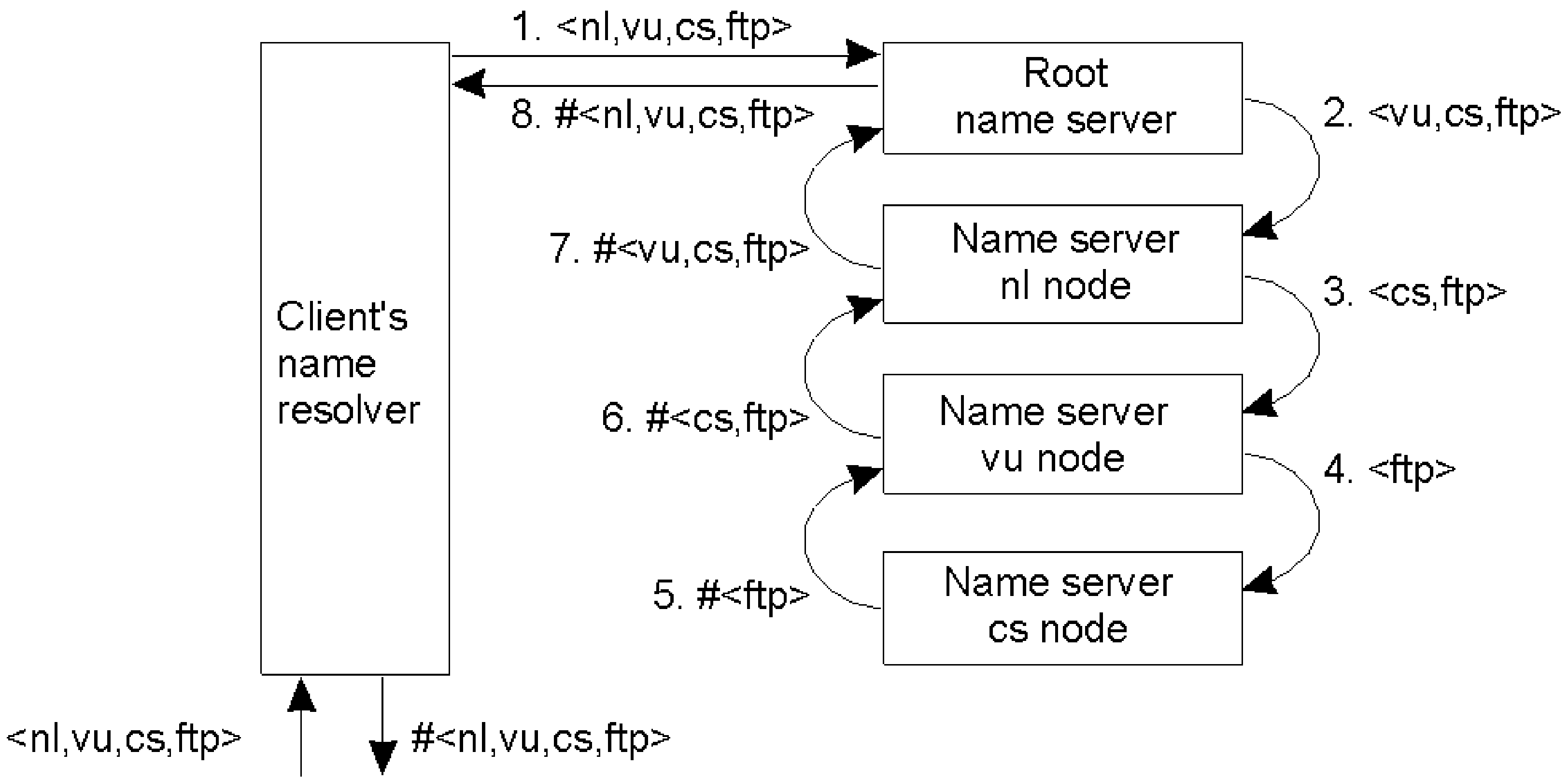
A comparison between name servers for implementing nodes from a large-scale name space partitioned into a global layer, as an administrational layer, and a managerial layer.

Name Resolution Implementation (1)



The principle of iterative name resolution.

Name Resolution Implementation (2)



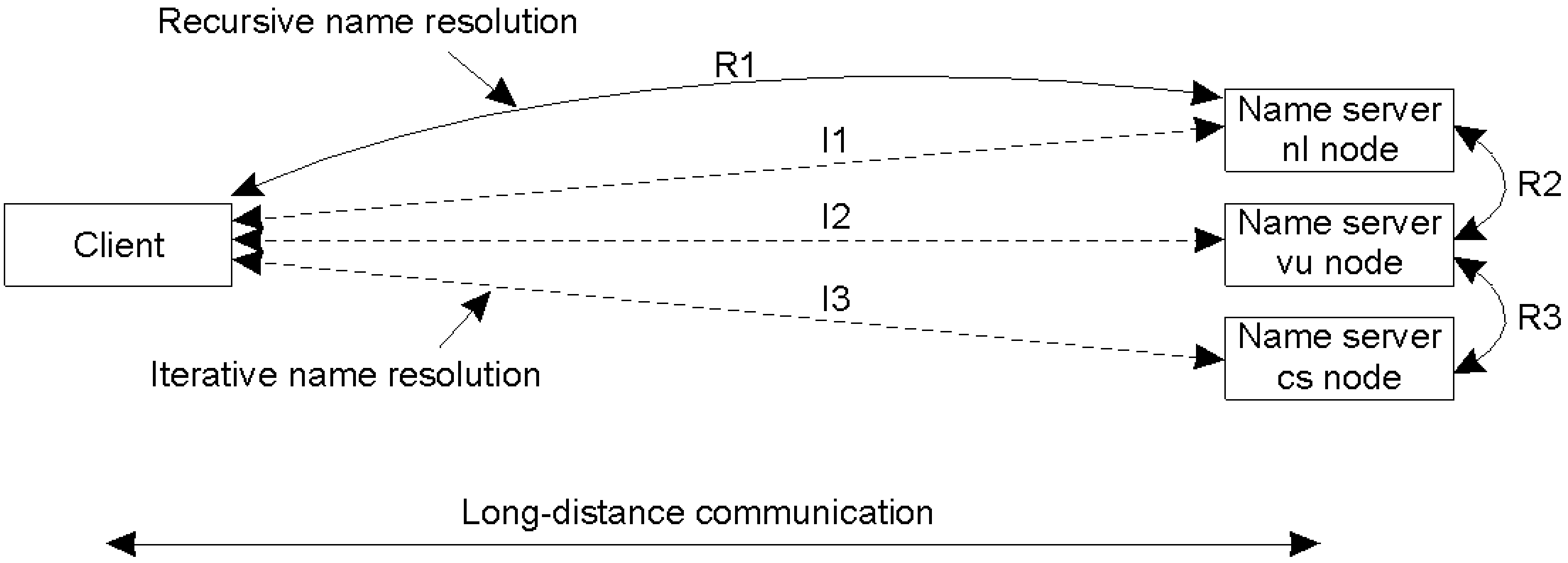
The principle of recursive name resolution.

Name Resolution Implementation (3)

Server for node	Should resolve	Looks up	Passes to child	Receives and caches	Returns to requester
cs	<ftp>	#<ftp>	--	--	#<ftp>
vu	<cs,ftp>	#<cs>	<ftp>	#<ftp>	#<cs> #<cs, ftp>
nl	<vu,cs,ftp>	#<vu>	<cs,ftp>	#<cs> #<cs,ftp>	#<vu> #<vu,cs> #<vu,cs,ftp>
root	<nl,vu,cs,ftp>	#<nl>	<vu,cs,ftp>	#<vu> #<vu,cs> #<vu,cs,ftp>	#<nl> #<nl,vu> #<nl,vu,cs> #<nl,vu,cs,ftp>

Recursive name resolution of <*nl, vu, cs, ftp*>. Name servers cache intermediate results for subsequent lookups.

Name Resolution Implementation (4)



Recursive name resolution
R1
I1
I2
I3
R2
R3
Client
Name server nl node
Name server vu node
Name server cs node
Iterative name resolution
Long-distance communication

The comparison between recursive and iterative name resolution with respect to communication costs.

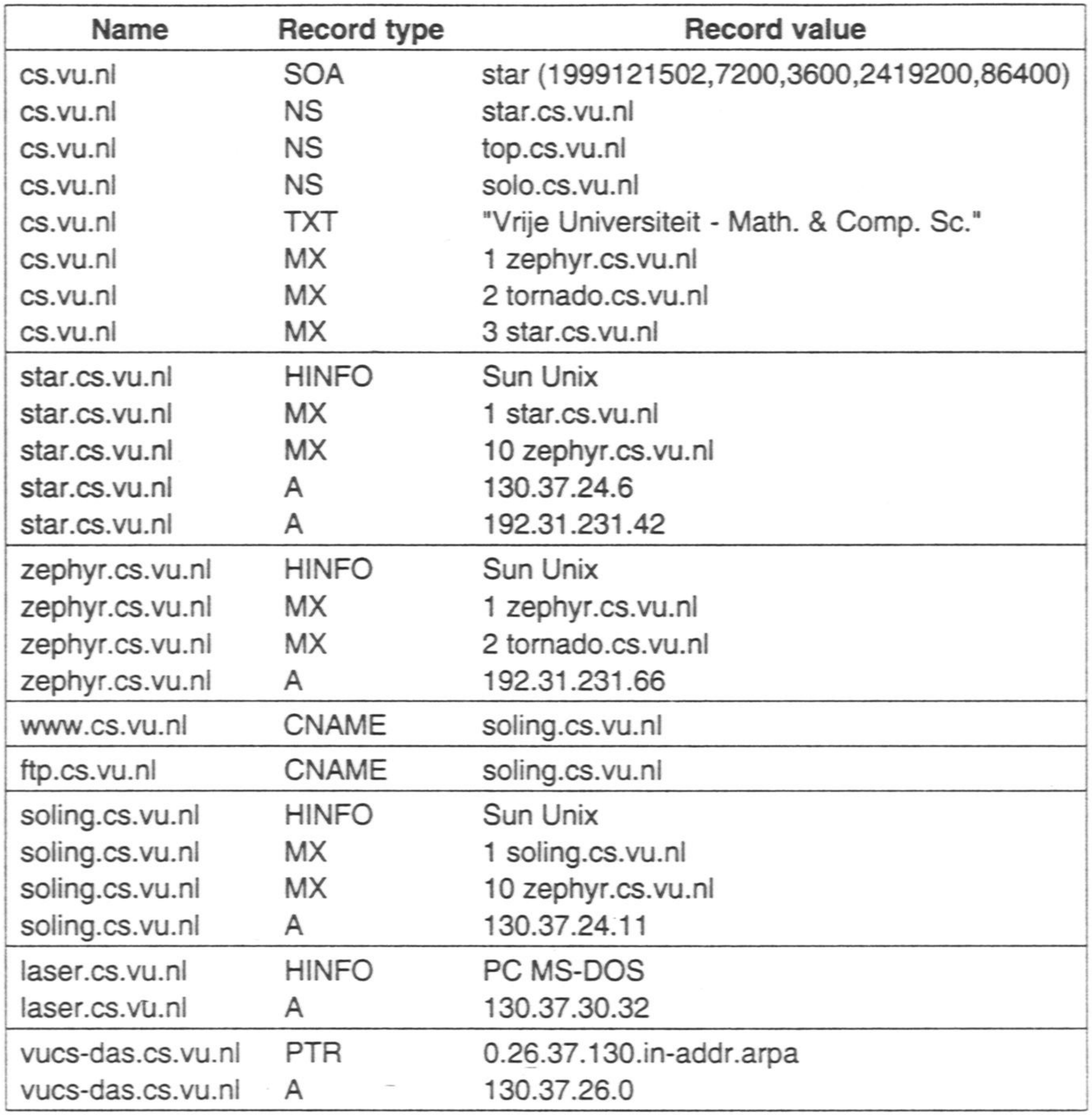
The DNS Name Space

Type of record	Associated entity	Description
SOA	Zone	Holds information on the represented zone
A	Host	Contains an IP address of the host this node represents
MX	Domain	Refers to a mail server to handle mail addressed to this node
SRV	Domain	Refers to a server handling a specific service
NS	Zone	Refers to a name server that implements the represented zone
CNAME	Node	Symbolic link with the primary name of the represented node
PTR	Host	Contains the canonical name of a host
HINFO	Host	Holds information on the host this node represents
TXT	Any kind	Contains any entity-specific information considered useful

The most important types of resource records forming the contents of nodes in the DNS name space.

DNS Implementation (1)

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 Implementation (2)

Name	Record type	Record value
cs.vu.nl	NIS	solo.cs.vu.nl
solo.cs.vu.nl	A	130.37.21.1

Part of the description for the *vu.nl* domain which contains the *cs.vu.nl* domain.

Attribute-Based Naming

- Underlying idea:
  - Instead of identifying a particular entity
  - Describe the entity that is needed

- Entity Description:
  - (attribute, value) pairs

Directory Services

- Support attribute-based naming
- Every entity is described by a number of (attribute, value) pairs
- Resolving a name = executing a query
- An entity description language is needed

Resource Description Framework

- RDF
- (Resource, Property, Value)

- Resource (= subject): URI
- Property (= predicate): label
- Value (= object): literal or URI

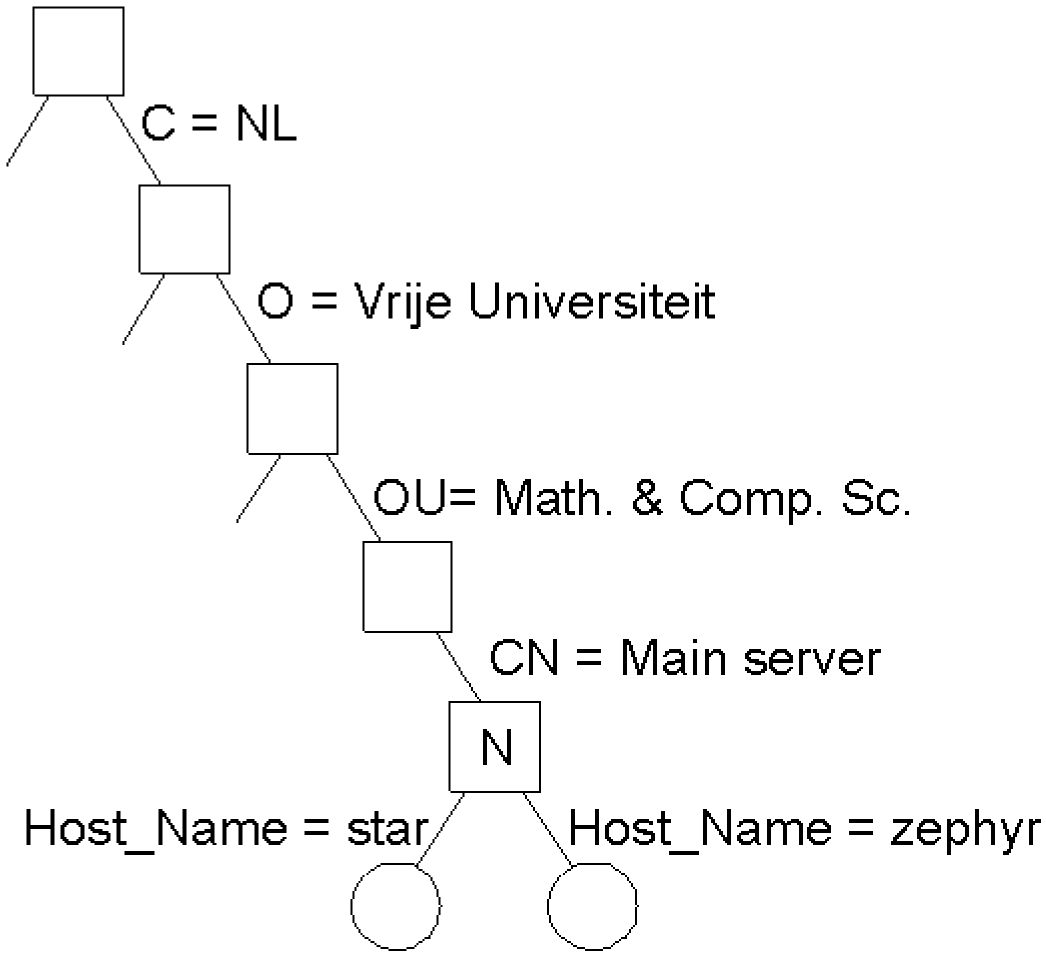
- http://www.w3.org/RDF/

The LDAP (X.500) Name Space (1)

Attribute	Abbr.	Value
Country	C	NL
Locality	L	Amsterdam
Organization	O	Vrije Universiteit
OrganizationalUnit	OU	Math. & Comp. Sc.
CommonName	CN	Main server
Mail_Servers	--	130.37.24.6, 192.31.231,192.31.231.66
FTP_Server	--	130.37.21.11
WWW_Server	--	130.37.21.11

A simple example of a X.500 directory entry using X.500 naming conventions.

The LDAP Name Space (2)



Part of the directory information tree.

The LDAP Name Space (3)

Attribute	Value	Attribute	Value
Country	NL	Country	NL
Locality	Amsterdam	Locality	Amsterdam
Organization	Vrije Universiteit	Organization	Vrije Universiteit
OrganizationalUnit	Math. & Comp. Sc.	OrganizationalUnit	Math. & Comp. Sc.
CommonName	Main server	CommonName	Main server
Host_Name	star	Host_Name	zephyr
Host_Address	192.31.231.42	Host_Address	192.31.231.66

Two directory entries having *Host_Name* as RDN.

Attribute-Based Naming in Grids

- Recently: Web Services, Grid Computing
- Directory Services
- Universal Directory and Discovery Integration (UDDI)

- Decentralized Implementations
  - On DHTs
  - On Semantic Overlay Networks

Attribute-Based Naming in DHTs

- Support for (attr, value) searches
  - With singular-valued attributes
  - With interval-valued attributes
- AVTree → hash value for each path
- Information on resource held on all the nodes responsible for the hash value of each path in the AVTree

Thank you for your attention!