Naming nei Sistemi Distribuiti

Naming (1)

- La risoluzione dei nomi permette ad un processo di accedere ad una entità in un sistema distribuito.
- Un sistema di naming è necessario per avere un modello comune di identificazione delle risorse.
- In un sistema distribuito il naming system è distribuito, per ragioni di scalabilità, efficienza, affidabilità, ecc.

Naming (2)

- In un sistema distribuito
  - Un **nome** è una stringa.
  - Una **entità** è una risorsa generica.

  - Un **access point** è una entità speciale.
  - Un nome di un access point è chiamato **address**.

- Esempi: *Telefono - numero, canale - frequenza*

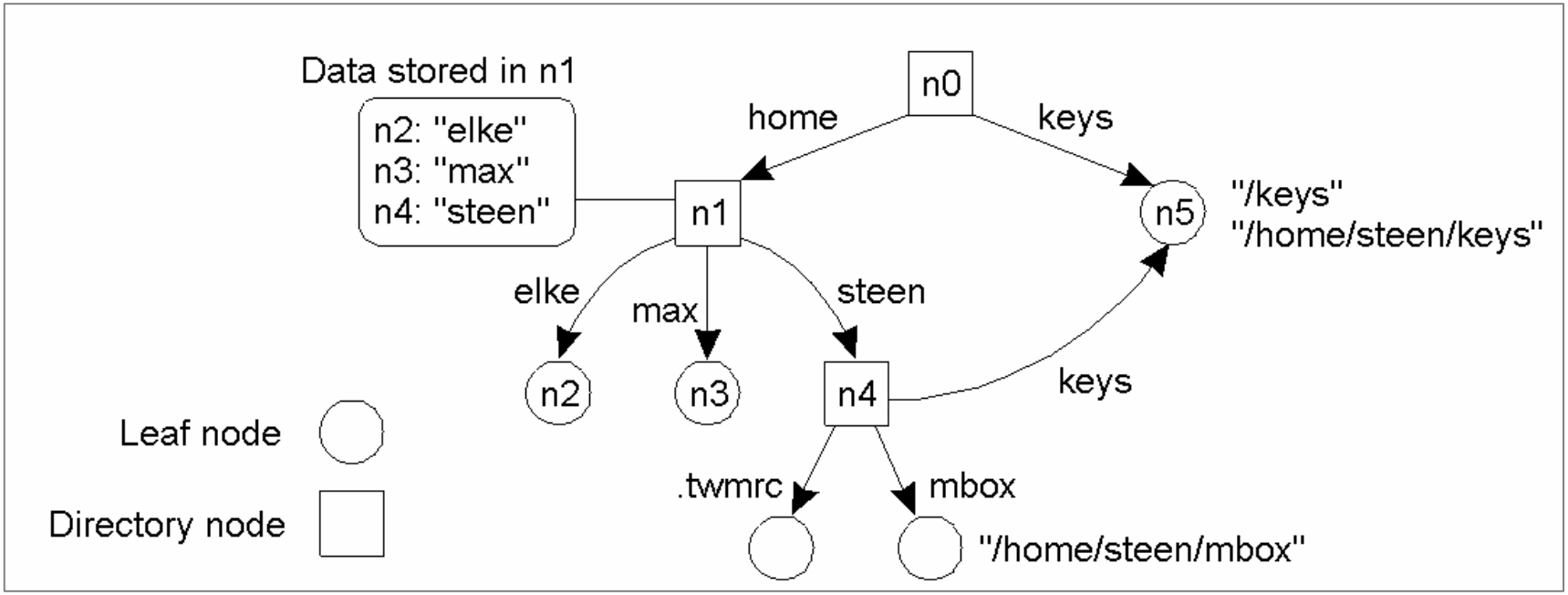
Naming (3)

- Una entità può avere più di un access point.
- Un nome di entità può essere **indipendente dalla locazione**.
- Quando
  - Un nome si riferisce ad una sola entità,
  - Ogni entità è riferita al più da un nome,
  - Un nome fa riferimento sempre ad una stessa entità (no riuso)

  Il nome è detto **identificatore** (**identifier**).

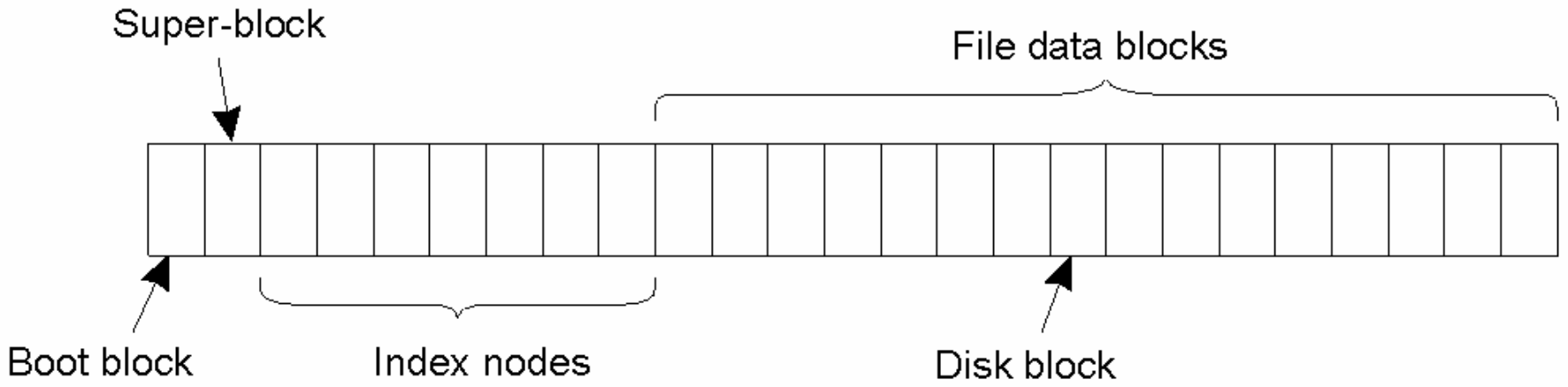
Name Spaces (1)

Un name space è una struttura che organizza i nomi.



Un grafo di naming generale con un solo nodo radice.

Name Spaces (2)



L'organizzazione generale della implementazione del file system di UNIX su un disco logico di blocchi contigui.

Name resolution

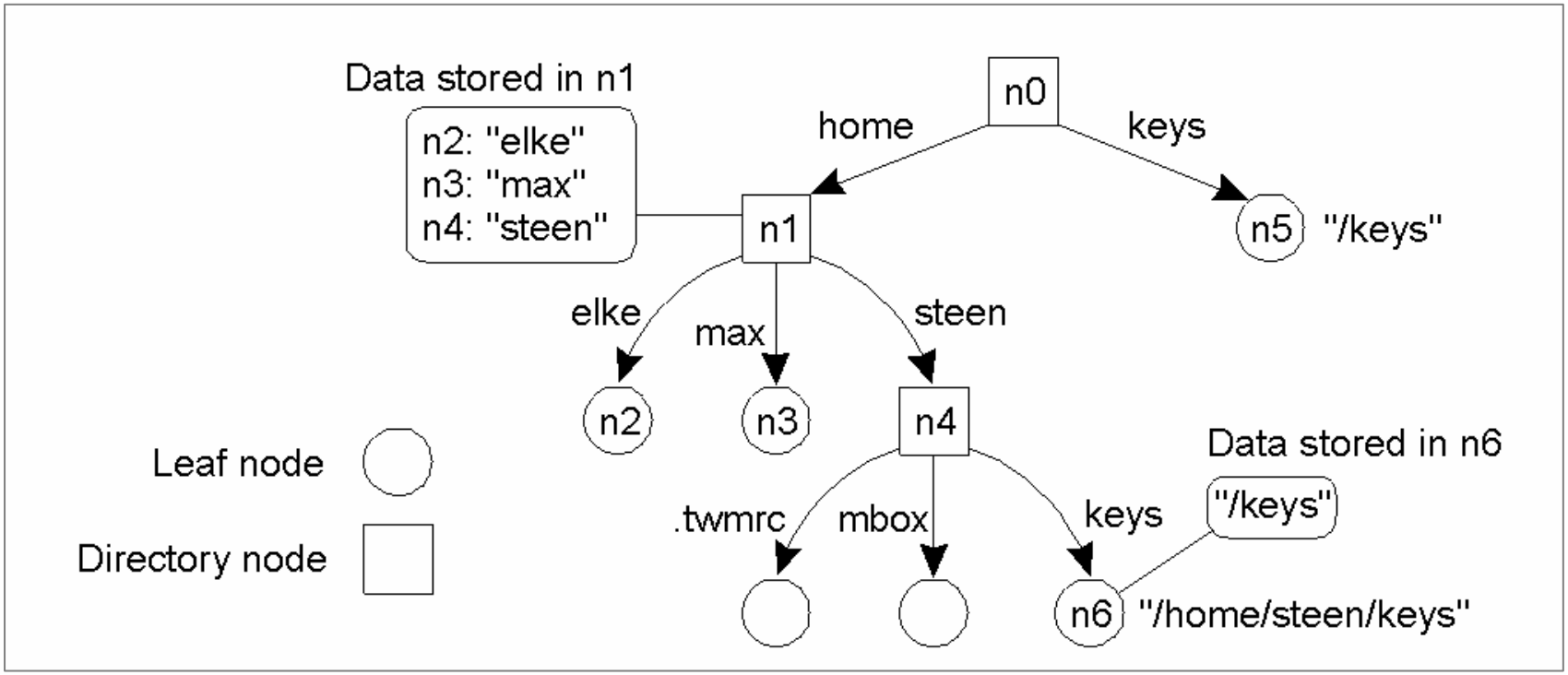
- Risoluzione dei nomi distribuita :

$$N:<l_1, l_2, \ldots, l_n>$$

- Closure mechanism: conosce da dove la risoluzione di un nome ha inizio.

- Vengono usati gli alias:
  - hard links
  - symbolic links.

Linking e Mounting (1)



Il concetto di link simbolico spiegato in un grafo di naming (n6 è un link simbolico)

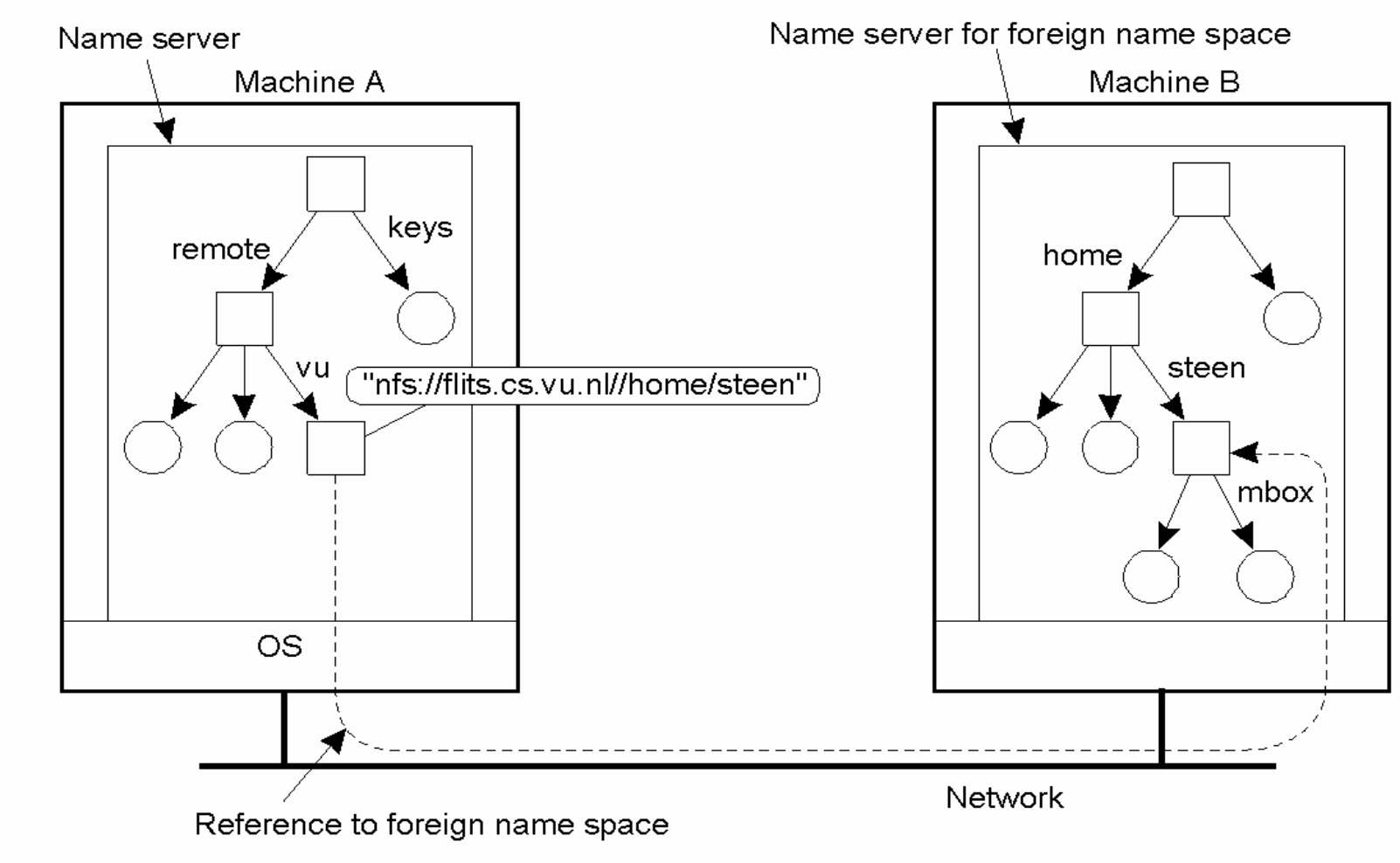
Linking e Mounting (2)

- Per il montaggio di un name space remoto in un sistema distribuito è necessario risolvere:

1. il nome del protocollo di accesso,
2. il nome del server,
3. il nome del punto di mounting nel name space remoto.

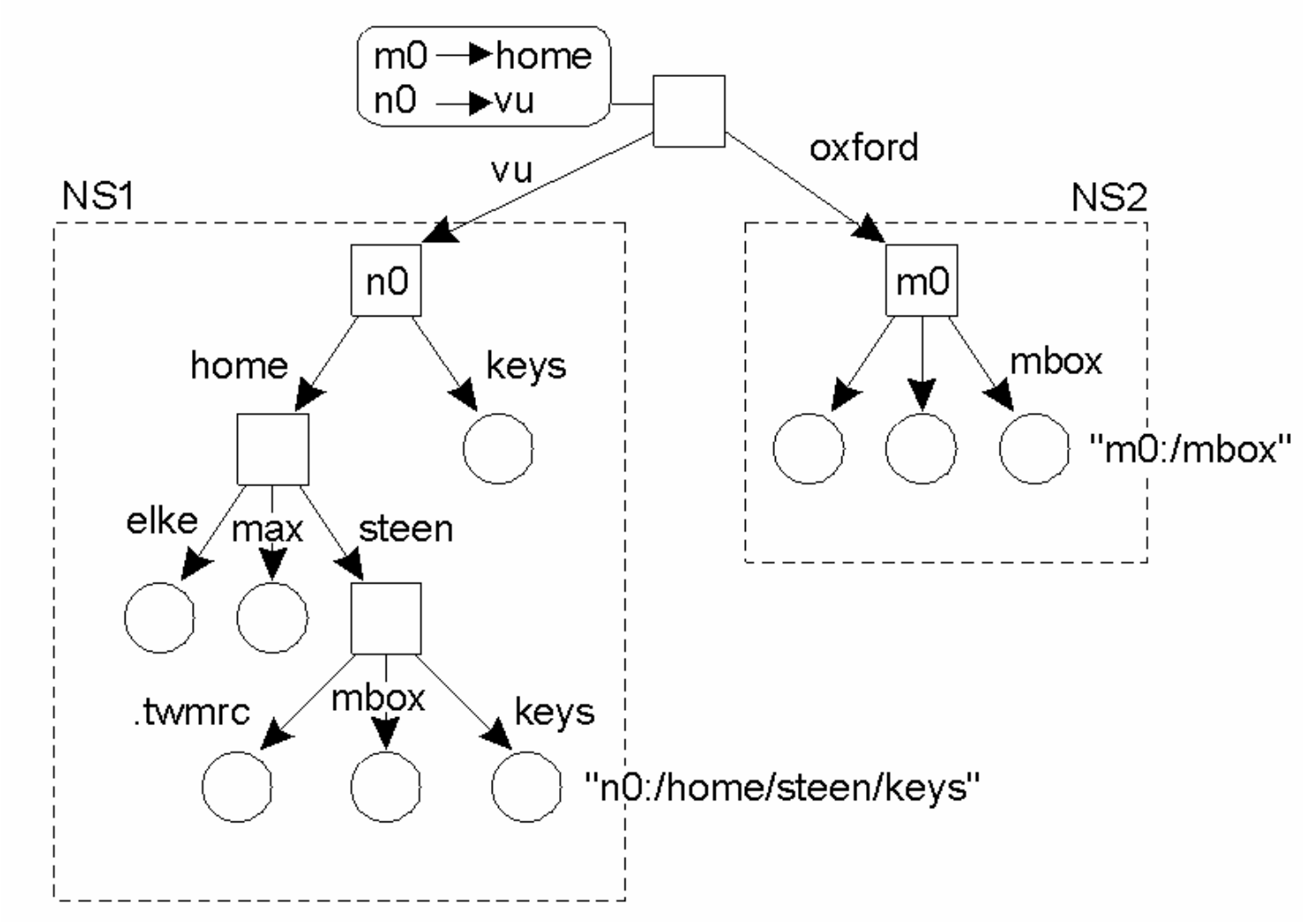
Esempio: nfs://flits.cs.vu.nl/home/steen

Linking e Mounting (3)



Mounting di un name space remoto attraverso un protocollo specifico.

Linking e Mounting (3)



Organizzazione del DEC Global Name Service

Name Space Distribuito

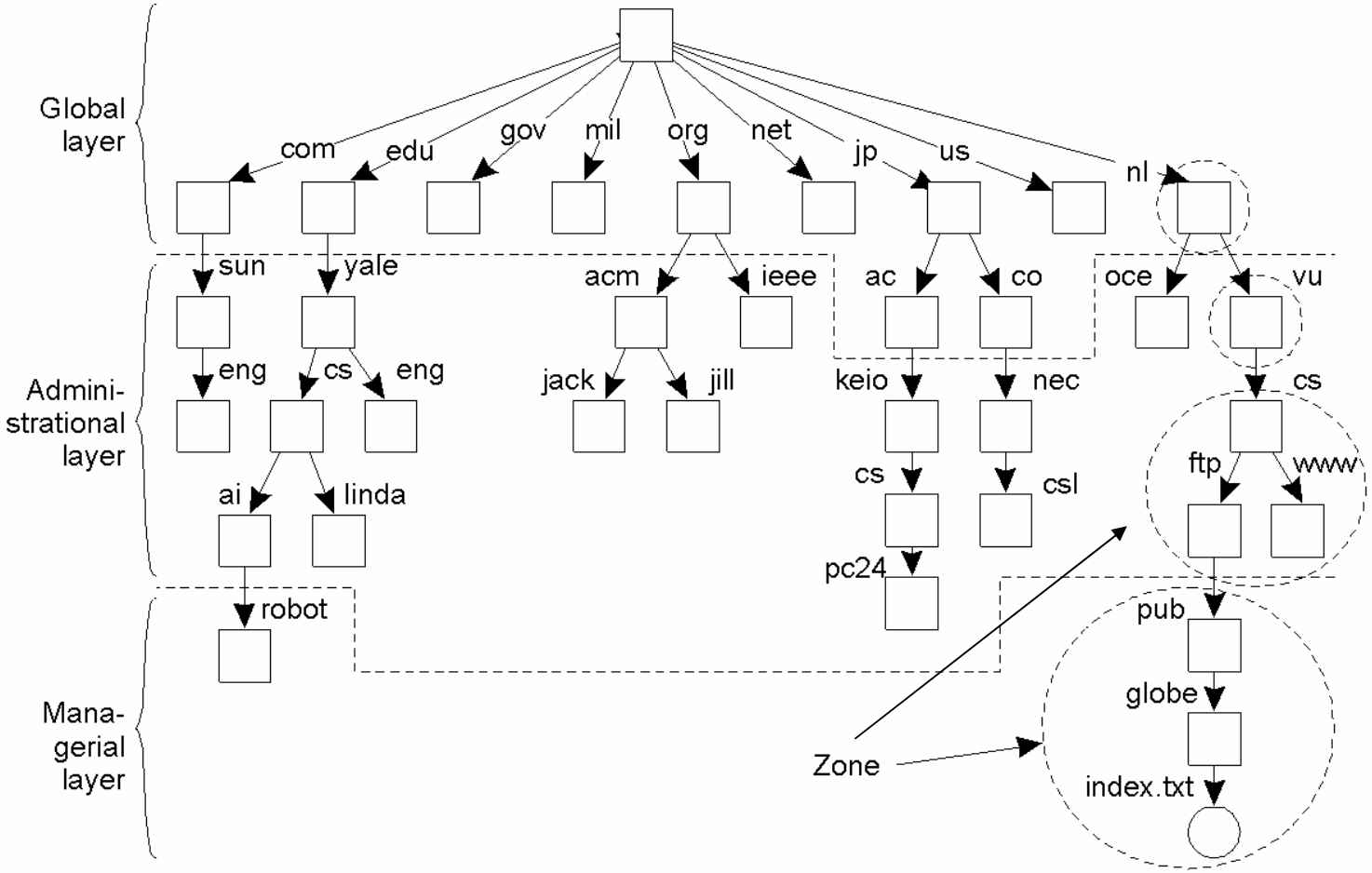
I sistemi distribuiti di grandi dimensioni usano name server gerarchici.

La replicazione dei name server può essere utile.

I name space possono essere suddivisi in più livelli logici:

- Livello globale,
- Livello di amministrazione,
- Livello di gestione.

Distribuzione del Name Space (1)



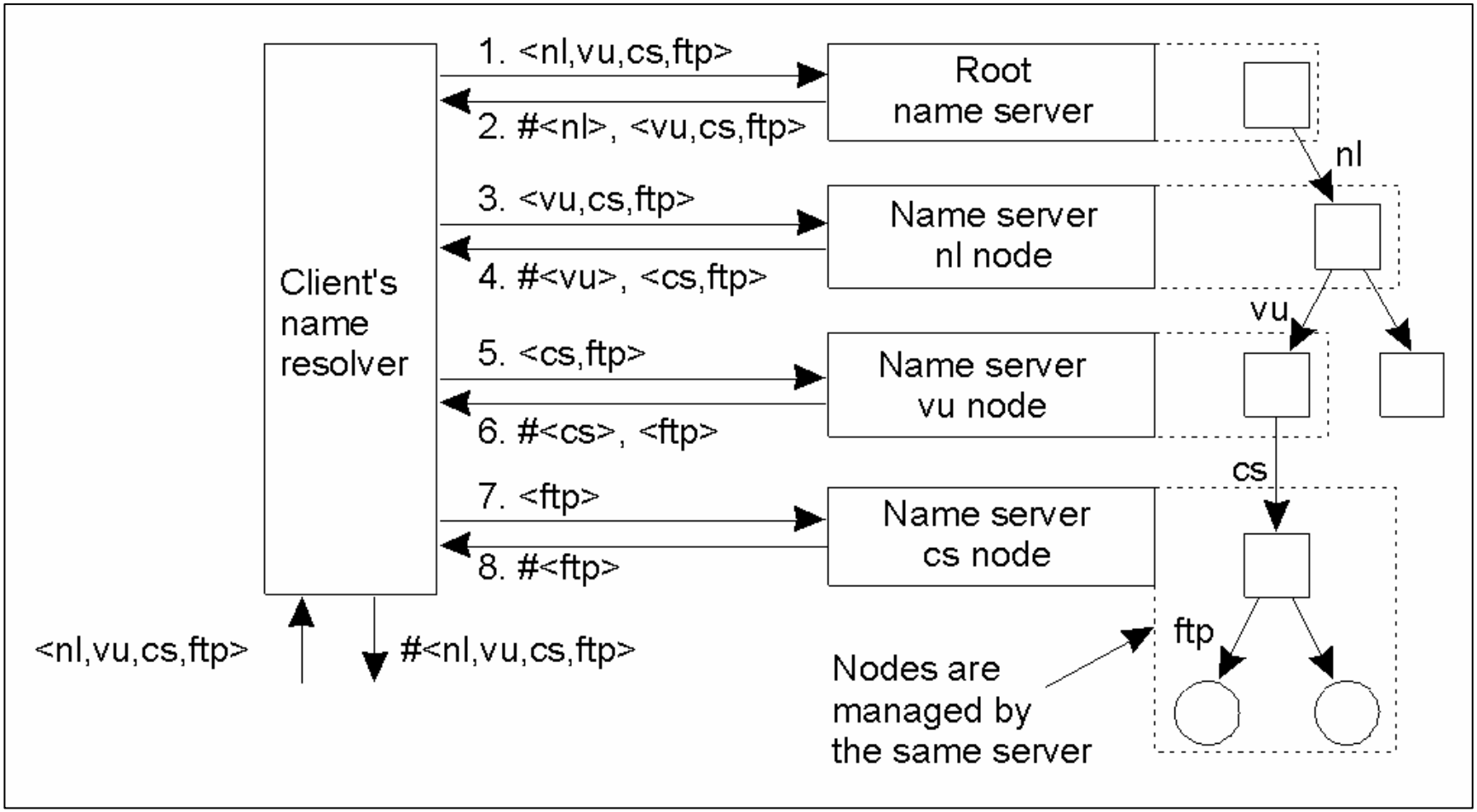
Un esempio di partizionamento del DNS name space a tre livelli che include file accessibili via Internet.

Distribuzione del Name Space (2)

	Globale	Amministrativo	Gestionale
Scala geografica della rete	Mondiale	Organizzazione	Dipartimento
Numero totale dei nodi	Pochi	Molti	Molti
Tempi di risposta	Secondi	Millisecondi	Immediati
Propagazione degli update	Lenta	Immediata	Immediata
Numero di repliche	Molti	Pochi o nessuna	Nessuna
E' usato caching nel client?	Si	Si	Talvolta

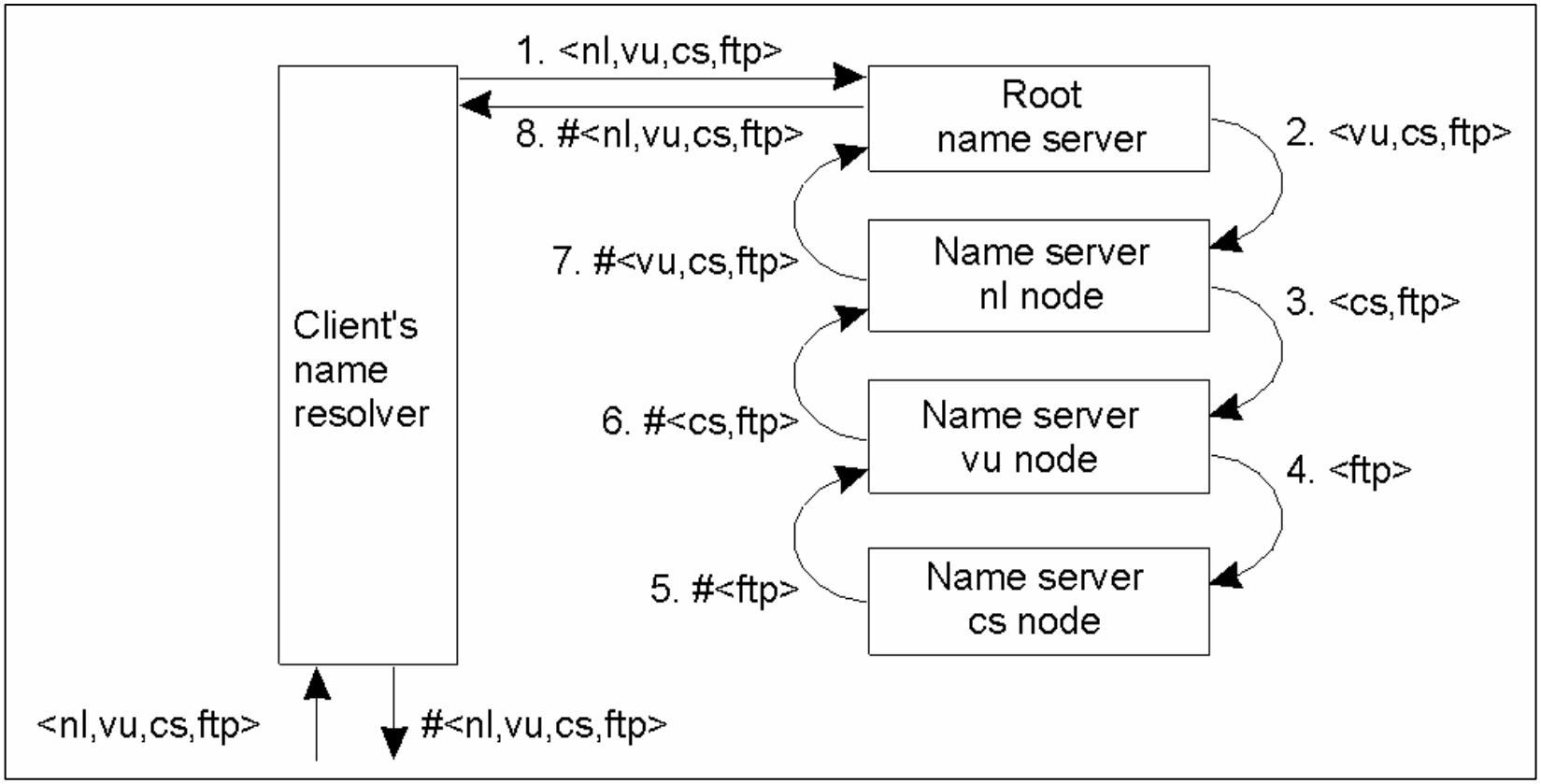
Un confronto tra le caratteristiche delle organizzazione dei name servers a diversi livelli di scala.

Name Resolution Iterativa



Lo schema di name resolution iterativa. Il cliente invia richieste ad ognuno dei name server. Ogni server svolge una parte della risoluzione e il client effettua le singole richieste.

Name Resolution Ricorsiva



Lo schema di name resolution ricorsiva. Il cliente invia richieste al name server radice. Ogni server svolge una parte della risoluzione e effettua le richieste ai server sottostanti.

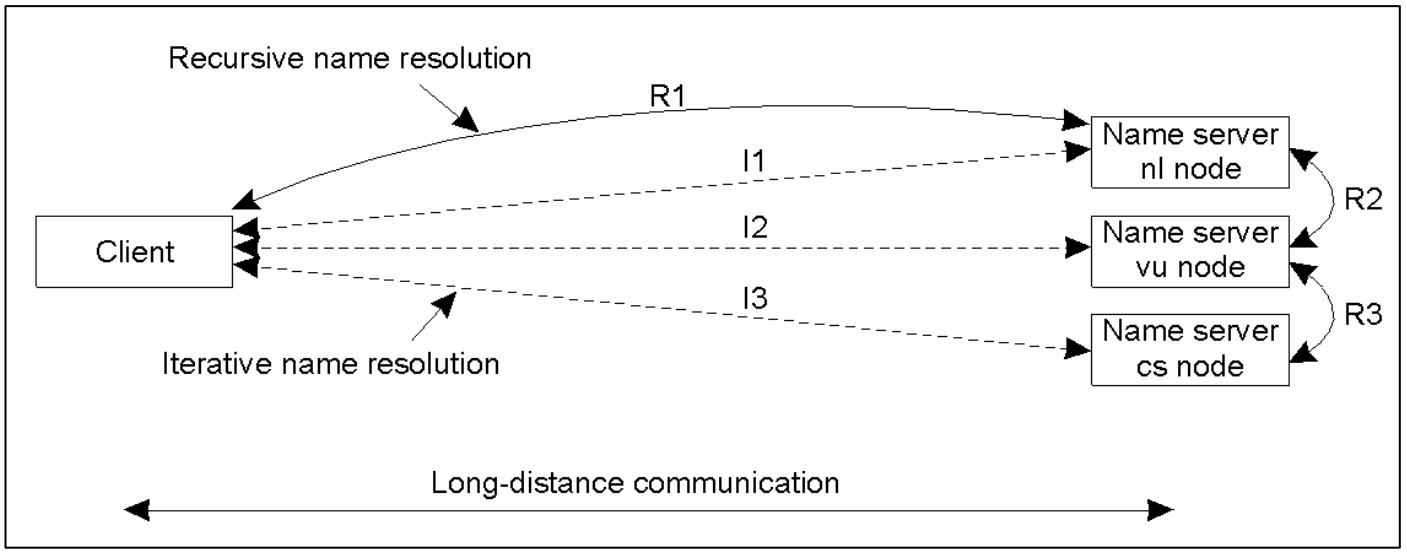
Name Resolution Ricorsiva

Server per nodo	Deve risolvere	Looks up	Passa al figlio	Riceve e memorizza	Ritorna al richiedente
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	<ni,vu,cs,ftp>	#<nl>	<vu,cs,ftp>	#<vu> #<vu,cs> #<vu,cs,ftp>	#<nl> #<nl,vu> #<nl,vu,cs> #<nl,vu,cs,ftp>

Risoluzione dei nomi ricorsiva di<*nl, vu, cs, ftp*>. I name server memorizzano nella cache i risultati intermedi per accessi successivi.

Implementazione della Risoluzione dei Nomi

Confronto tra name resolution ricorsiva e iterativa in relazione ai costi di comunicazione



Aspetti positivi della risoluzione ricorsiva:

- Minori costi di comunicazione
- Benefici dal caching dei nomi nei server

Il Name Space del DNS

- Il Domain Name System (**DNS**) di Internet è il più grande (?) servizio di naming (distribuito) esistente oggi.
- Il DNS ha una struttura gerarchica. Le risorse sono identificate da **pathname** che sono composti da **etichette**.
- Ogni etichetta è composta al massimo da 63 caratteri e un pathname da 255 caratteri.
- Un sottoalbero è chiamato **domain** e la sua radice **domain name**.
- Ogni nodo contiene una sequenza di **resource records**.

Il Name Space del DNS

Tipo di record	Entità Associata	Descrizione
SOA	Zona	Contiene informazioni sulla zona rappresentata
A	Host	Contiene un IP address dell' host che questo nodo rappresenta
MX	Dominio	Indica un mail server per gestire gli indirizzi di email del nodo
SRV	Dominio	Indica un server per gestire un servizio specifico del nodo
NS	Zona	Indica un name server che implementa la zona interessata
CNAME	Nodo	Link Simbolico con il nome primario della zona rappresentata
PTR	Host	Contiene il nome canonico dell'host
HINFO	Host	Contiene infomazioni sugli host che il nodo rappresenta
TXT	Ogni tipo	Contiene informazioni ritenute utili sull'entità

I tipi più importanti di record delle risorse formano i contenuti dei nodi nel name space del DNS.

Implementazione del DNS (1)

Un estratto del database DNS per la zona *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)

Nome	Record type	Record value
cs.vu.nl	NS	solo.cs.vu.nl
solo.cs.vu.nl	A	130.37.21.1

Parte della descrizione per il dominio *vu.nl* che contiene il dominio *cs.vu.nl*

Naming e Localizzazione di Entità

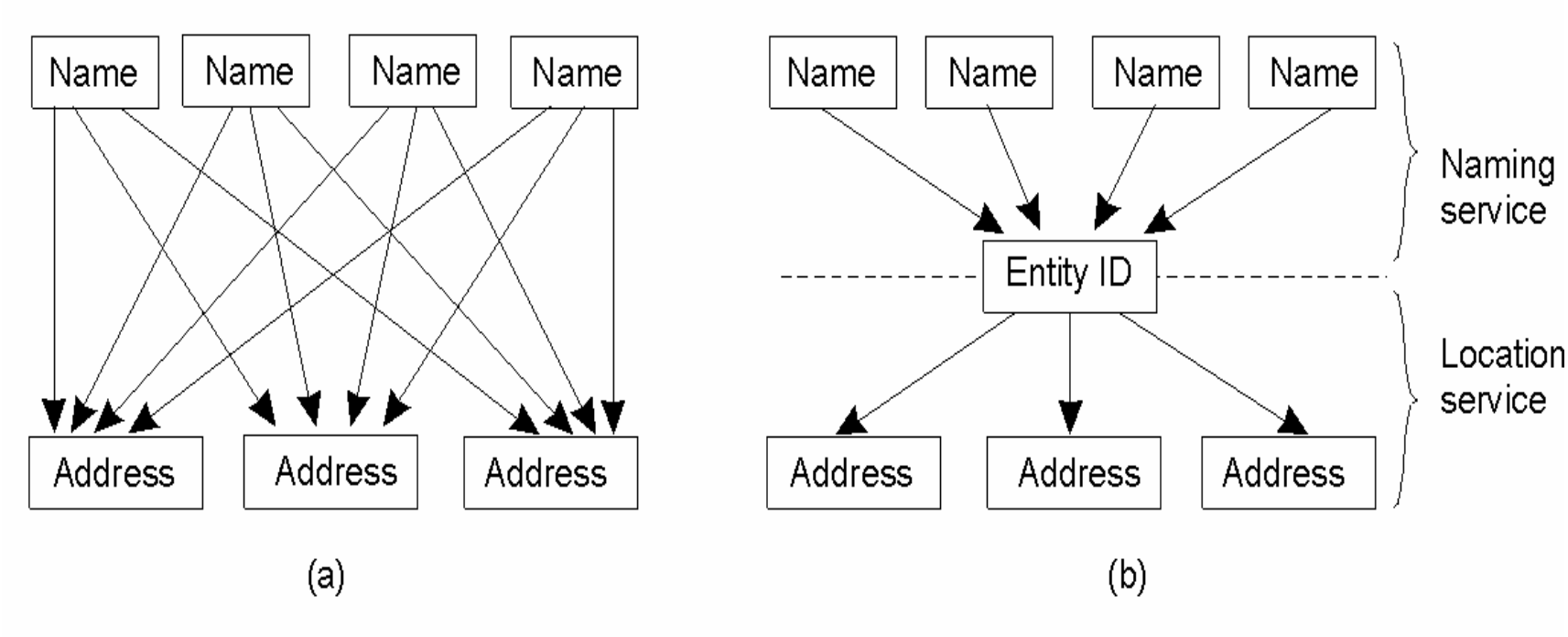
- Come gestire lo spostamento dei server in domini differenti?

  a) Memorizzare l'indirizzo della nuova macchina nel DNS entry della vecchia macchina.

  b) Memorizzare il nome della nuova macchina nel DNS entry della vecchia macchina.

- Un look up a più passi (multi-step) è necessario.

Naming e Localizzazione di Entità



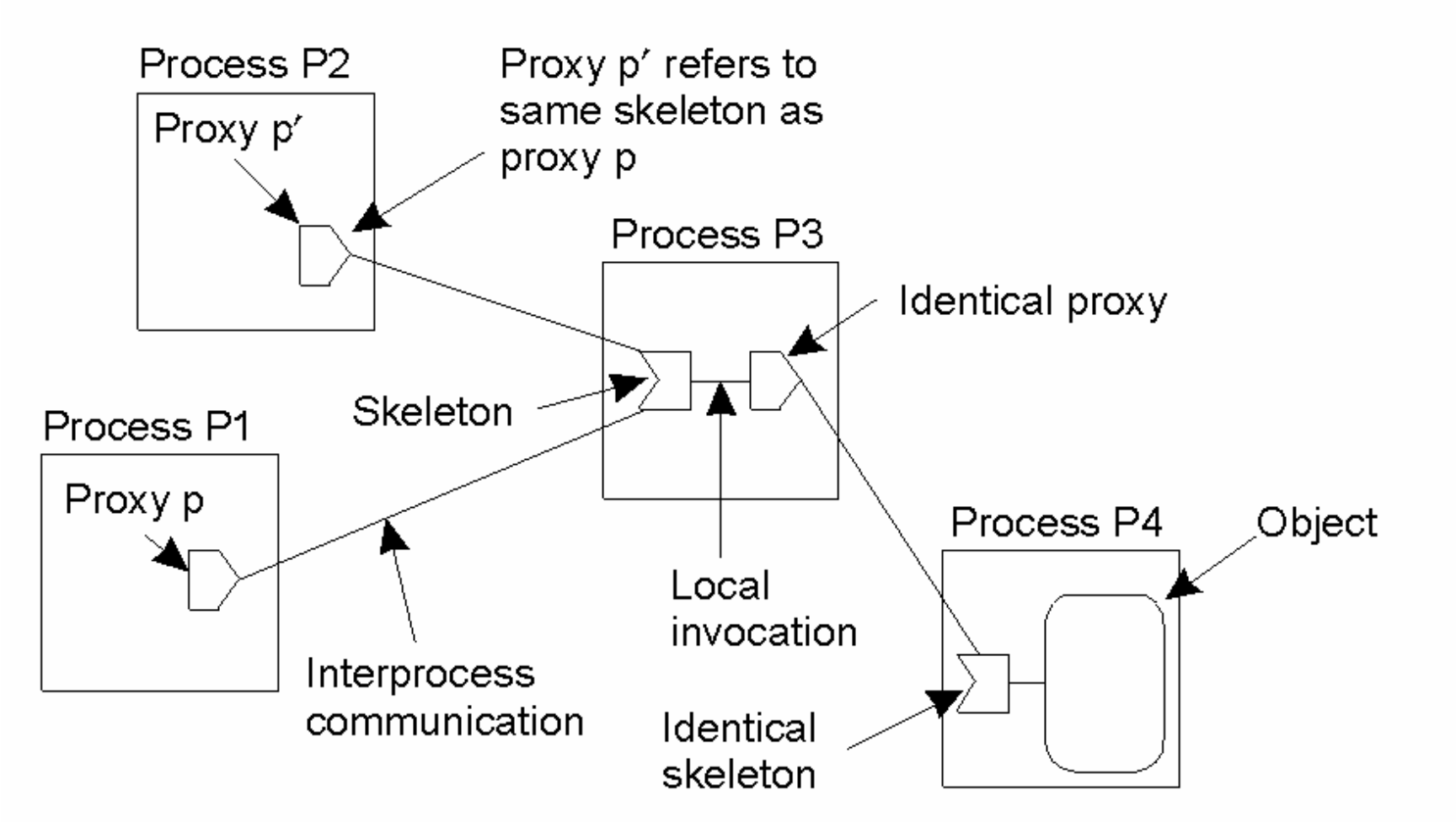
a) Mapping diretto, singolo livello tra nomi e indirizzi
   - Non adatto a risorse mobili

b) Mapping a due livelli usando identità
   - Flessibile e adatto a gestire risorse mobili

Localizzazione di entità: Broadcasting e Multicasting

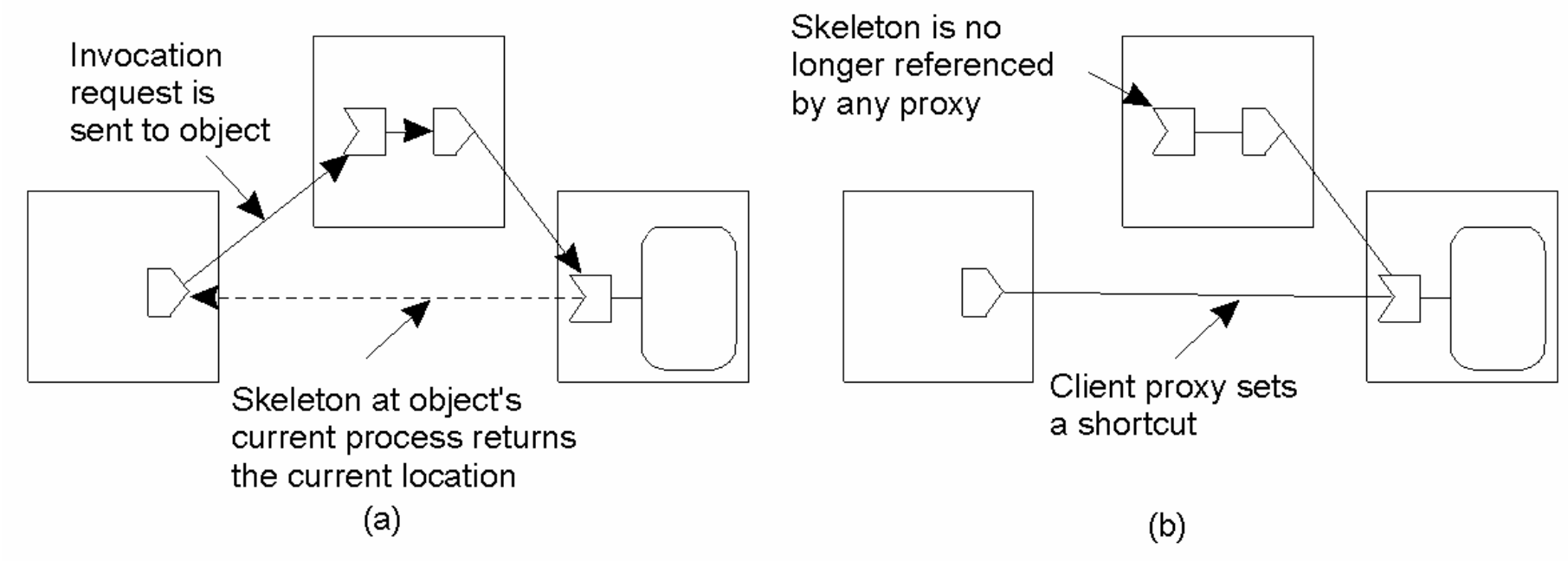
- In una LAN con pochi nodi può essere usato il meccanismo di broadcasting.
  - Un identificatore di entità è inviato ad ogni macchina chiedendo il controllo del proprietario dell'entità.
- Quando il numero dei nodi è elevato può essere usato il meccanismo di multicasting.
  - Si definiscono dei gruppi di nodi e si inviano le richieste a tutti i nodi di un gruppo.

Localizzazione di entità: Forwarding Pointers (1)



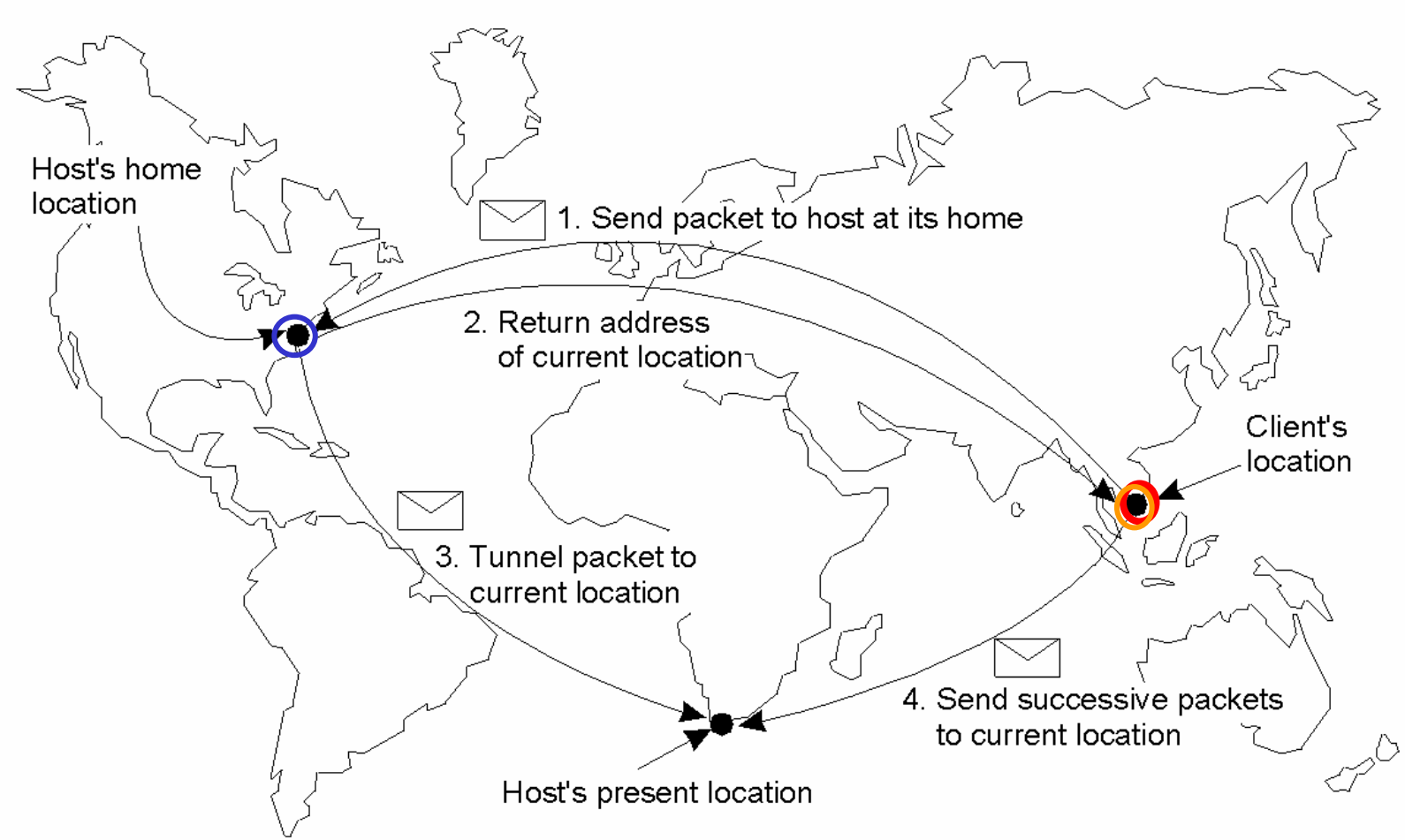
Il modello dei forwarding pointers usando le coppie (*proxy, skeleton*) per raggiungere una entità.

Localizzazione di entità: Forwarding Pointers (2)



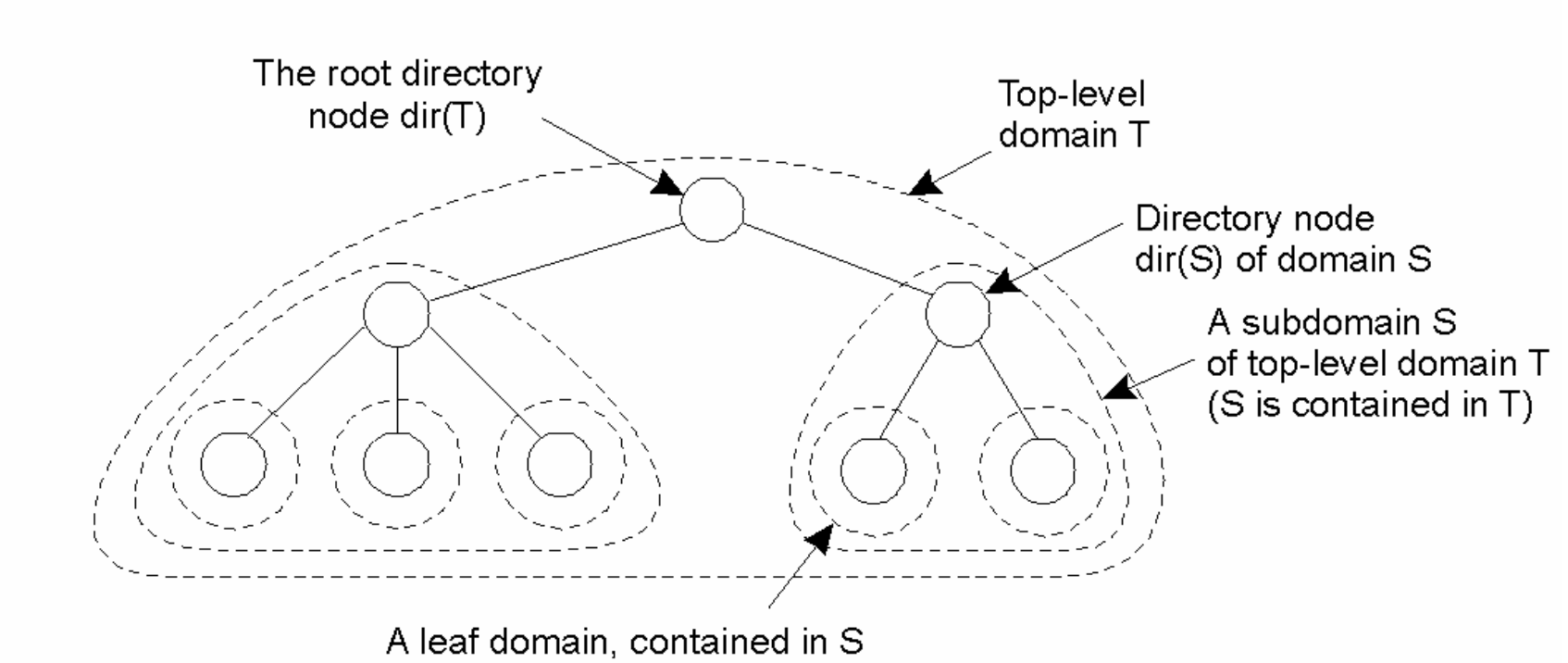
Redirezione di un forwarding pointer tramite la memorizzazione di un cammino in un proxy per ridurre le comunicazioni.

Localizzazione di entità: Approcci Home-Based



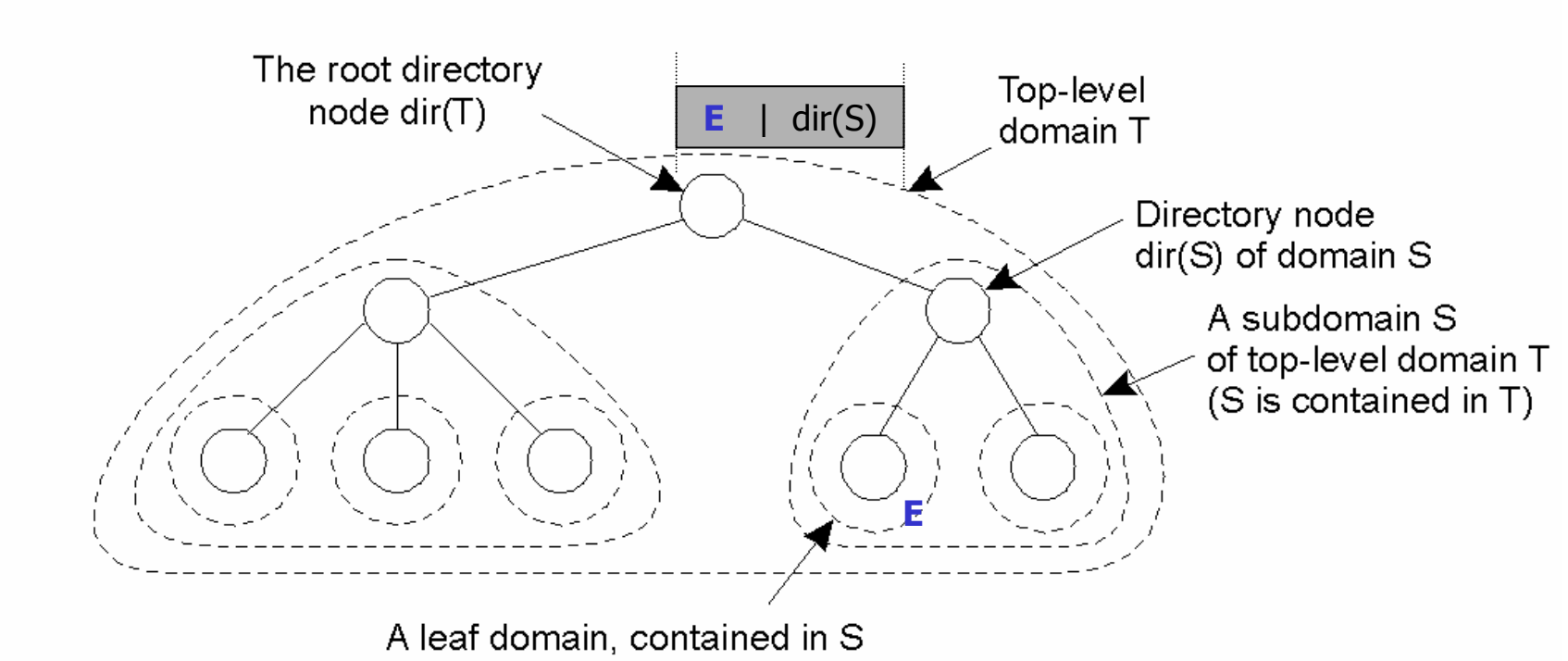
Il principio del Mobile IP.

Approcci Gerarchici (1)



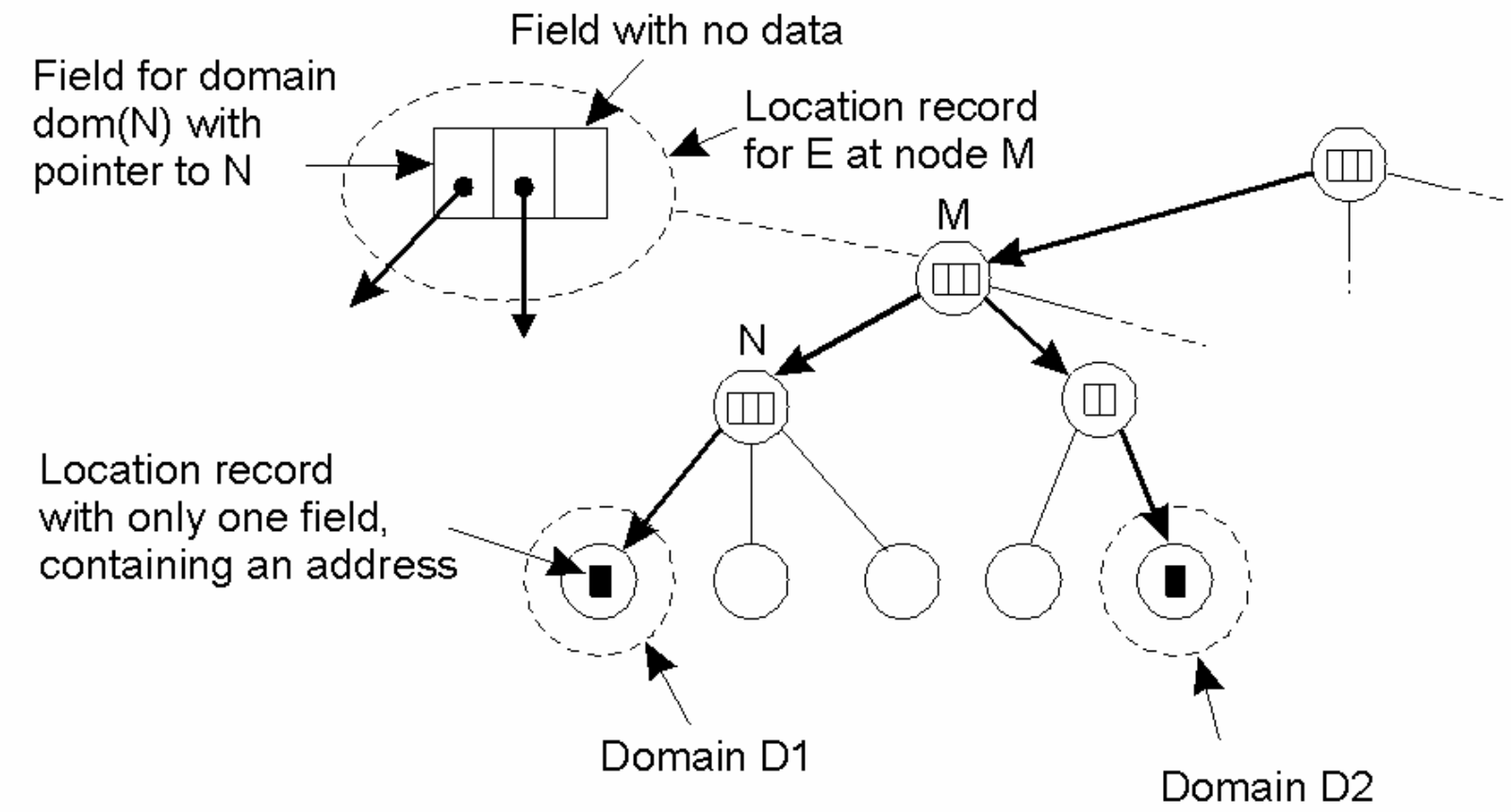
- Organizzazione gerarchica di un location service in domini, ognuno avente un directory node associato.
- Una entità in **D** è identificata da un location record in **dir(D)**.

Approcci Gerarchici (2)



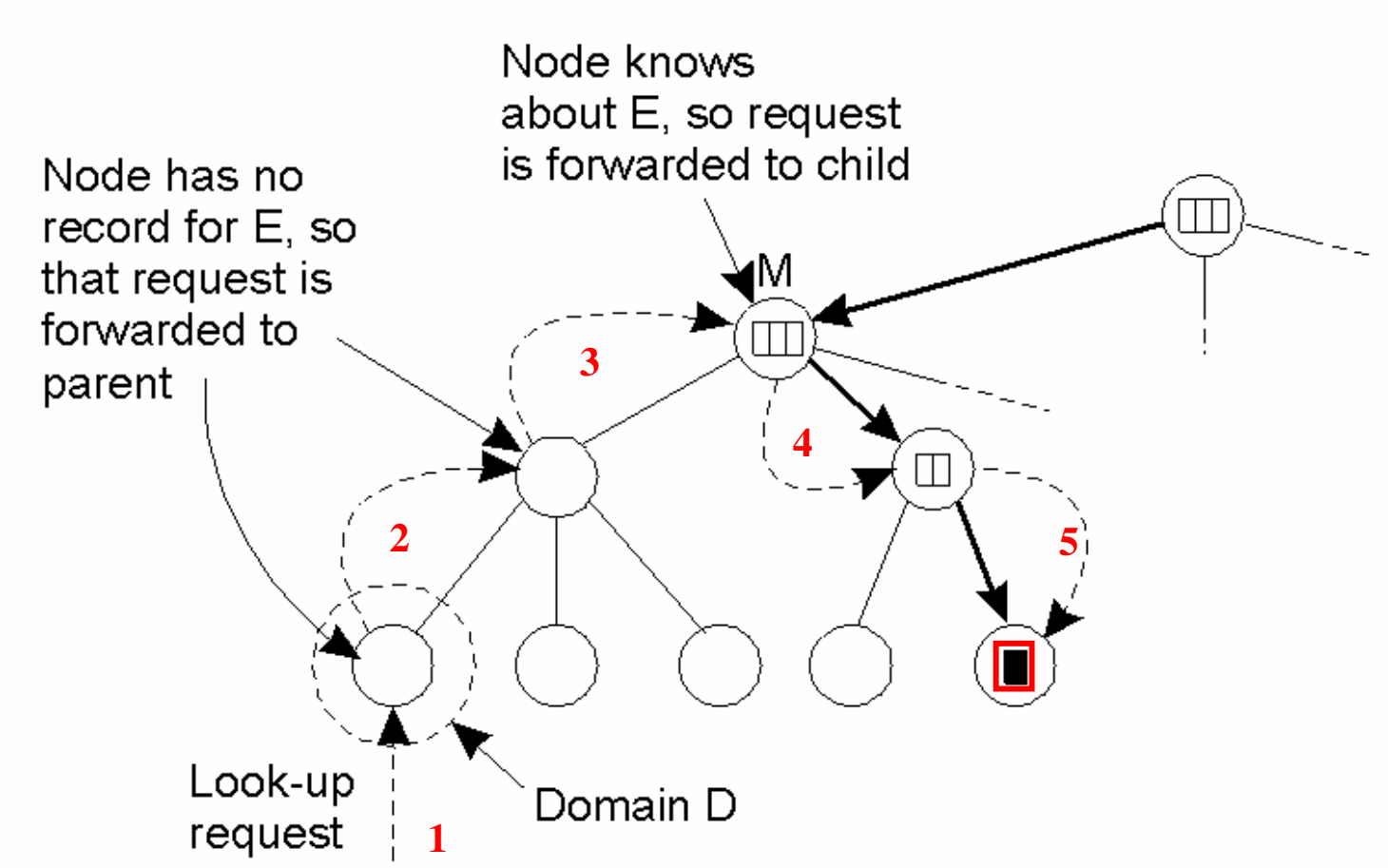
- Un nodo radice di un sotto-albero contiene una entry per ogni entità **E**.
- Il location record contiene un puntatore al directory node del successivo sotto-dominio del livello più basso che contiene l'entità **E.**

Approcci Gerarchici (3): Replicazione di entità



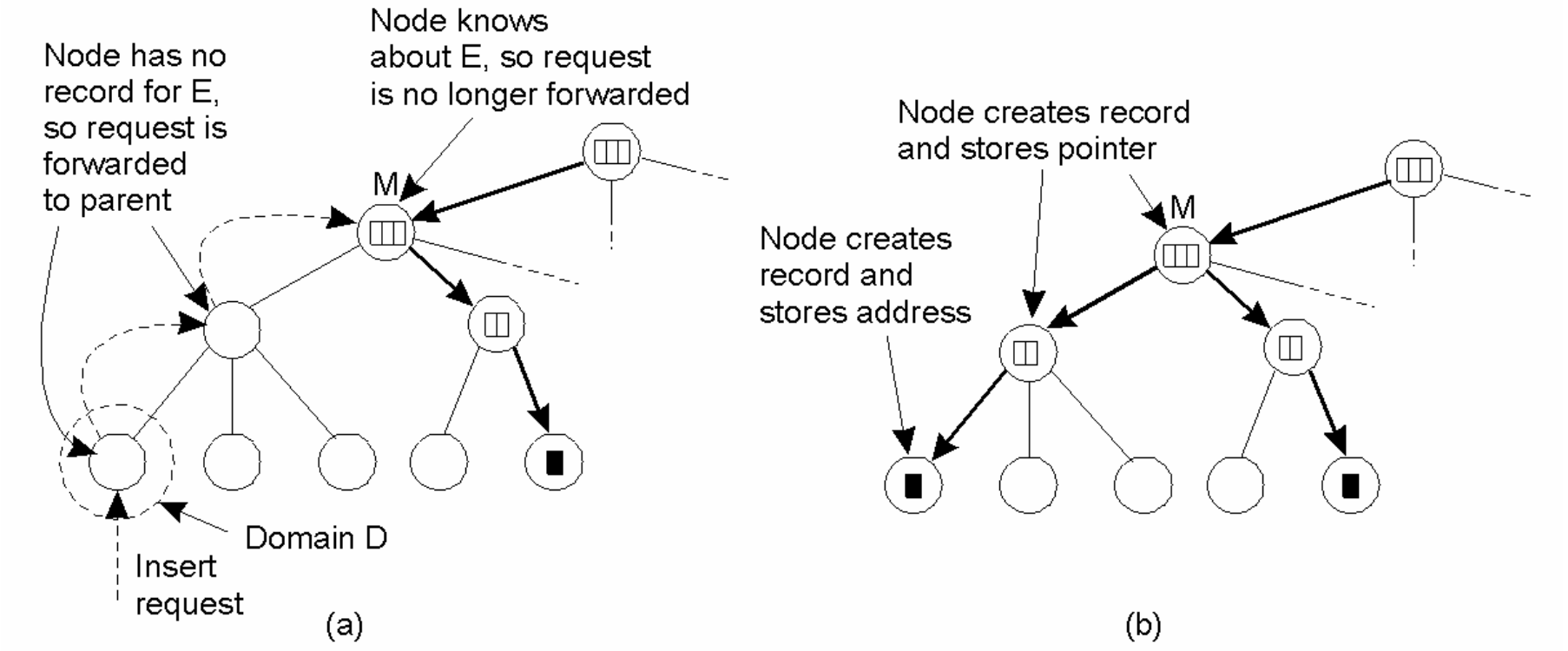
Un esempio di memorizzazione di informazione di una entità replicata che ha due indirizzi in differenti domini foglie.

Approcci Gerarchici (4): richiesta di accesso



Accesso ad una locazione in un location service organizzato gerarchicamente.

Approcci Gerarchici (5): richiesta di inserimento



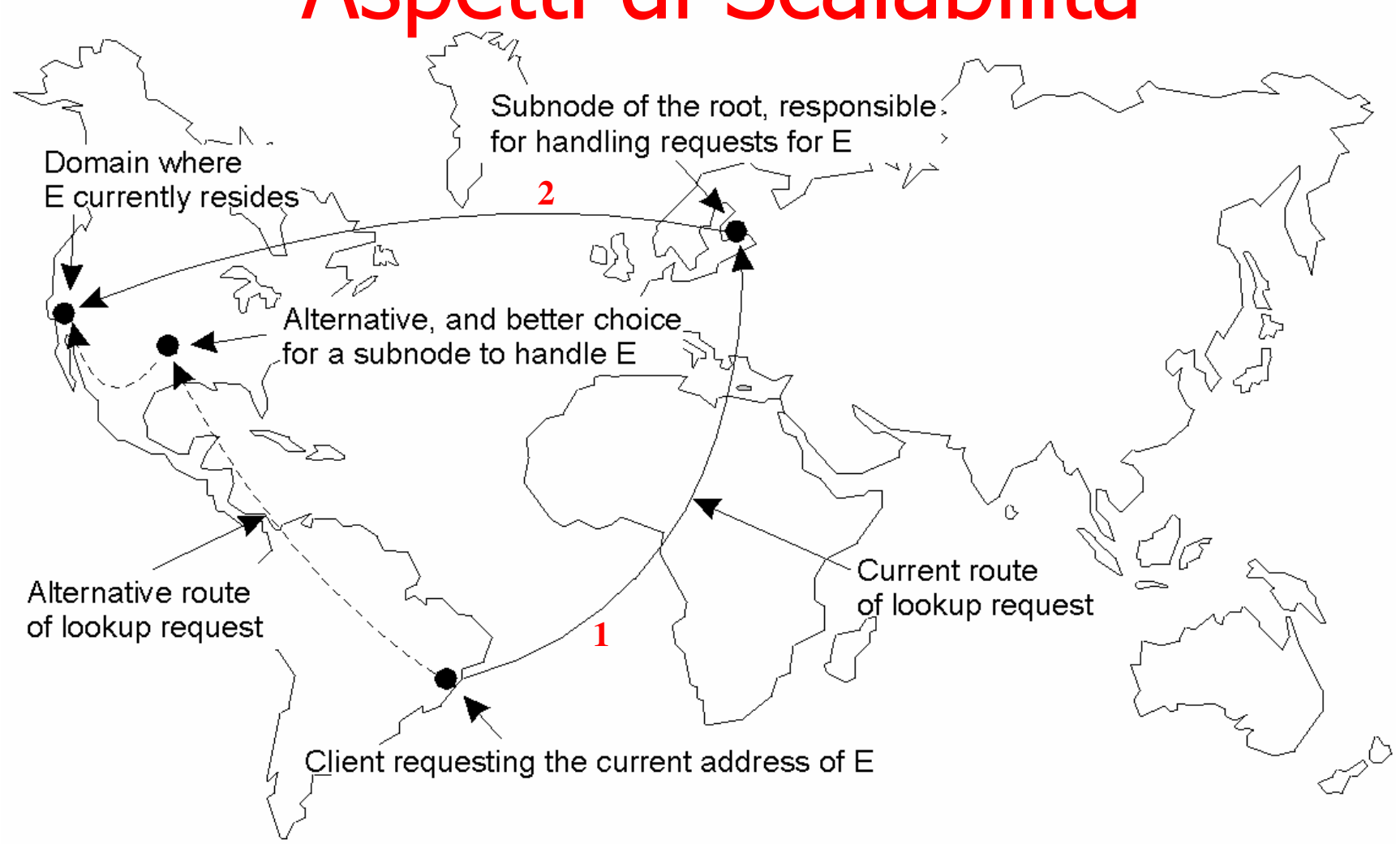
a) Una ***insert*** *request* è inviata al primo nodo che conosce l'entità *E*.

b) Viene creata una catena di forwarding pointers fino al nodo foglia.

Aspetti di Scalabilità

- In un servizio di locazione gerarchico il nodo radice deve memorizzare le entry per tutte le entità.
- Il nodo radice può diventare il collo di bottiglia del sistema.
- Può essere partizionato in un insieme di nodi che gestiscono un sottoinsieme di entità.
- Trovare il modo migliore per localizzare i nodi è molto complesso.

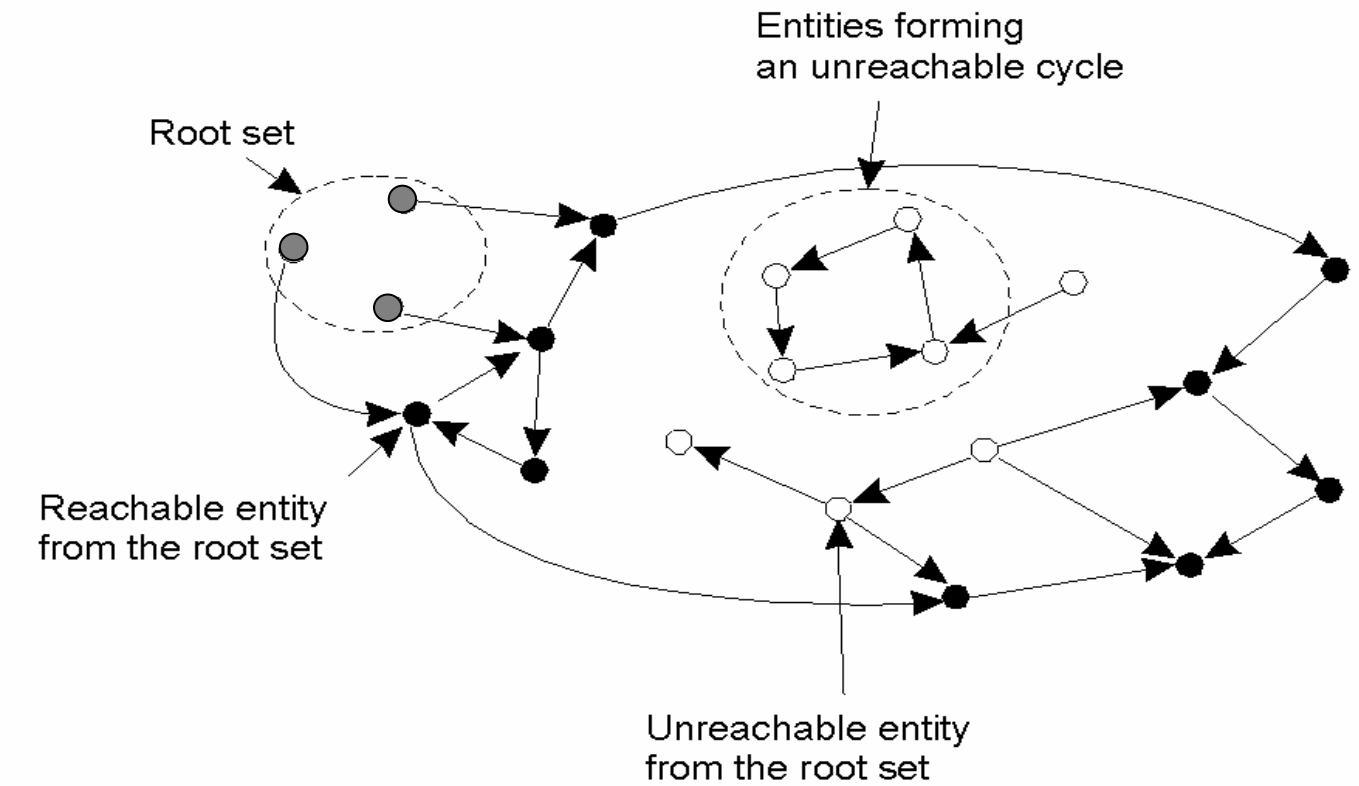
Aspetti di Scalabilità



Problemi di scalabilità relativi alla distribuzione uniforme di sotto-nodi di un nodo radice partizionato.

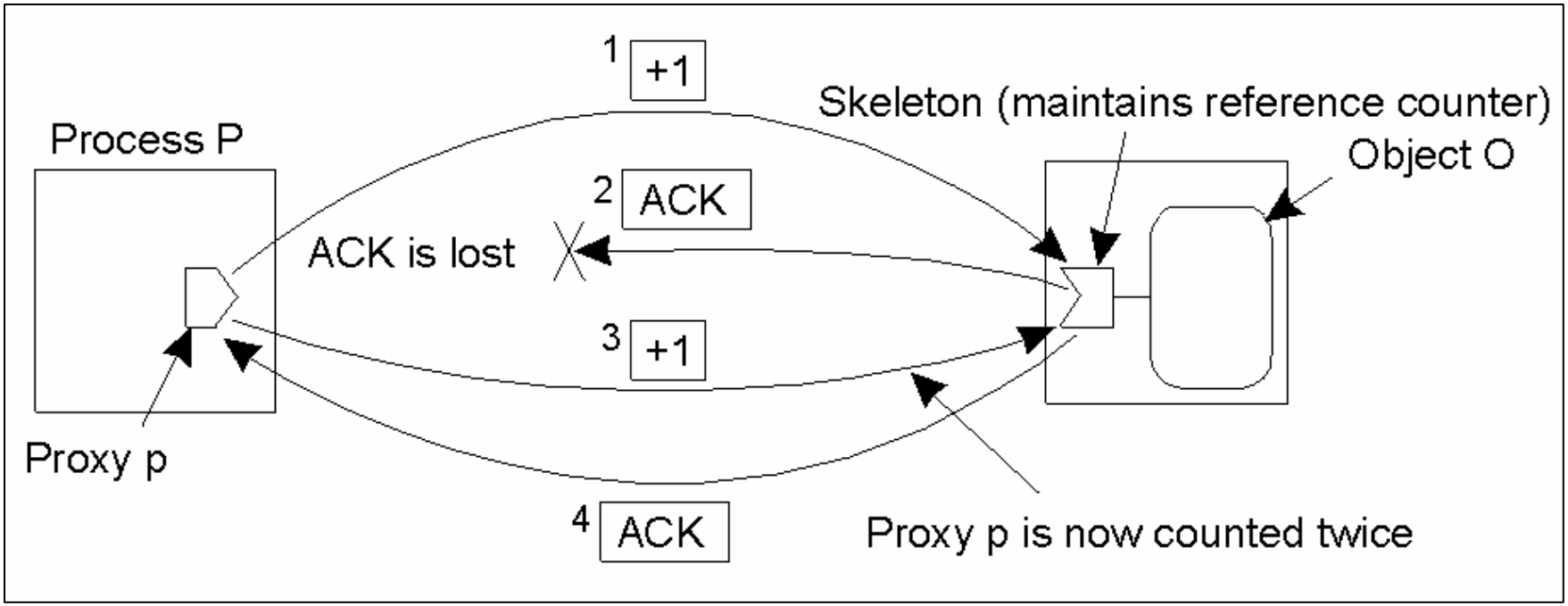
Problema di oggetti non referenziati

Soluzione: garbage collector distribuito.



Un esempio di un grafo che rappresenta oggetti contenenti riferimenti ad ogni altro. I nodi bianchi dovrebbero essere rimossi.

Reference Counting



Il problema di mantenere un corretto conteggio dei riferimenti in presenza di comunicazione non affidabile: *identificazione di messaggi duplicati*.

Reference Listing

- Il questo modello non si tiene conto del numero di riferimenti ma si mantiene una lista dei riferimenti.

- Questo approccio non richiede comunicazioni affidabili perché:
  - Aggiungere un riferimente esistente non ha alcun effetto sulla lista.
  - Togliere un riferimento non esistente non ha alcun effetto sulla lista.

- Il Reference Listing è usato in Java RMI: un processo invia il suo riferimento ad un oggetto remoto e l'oggetto aggiunge il processo alla sua reference list.