Chapter 5

Question 1 Draw The principle of mobile IP

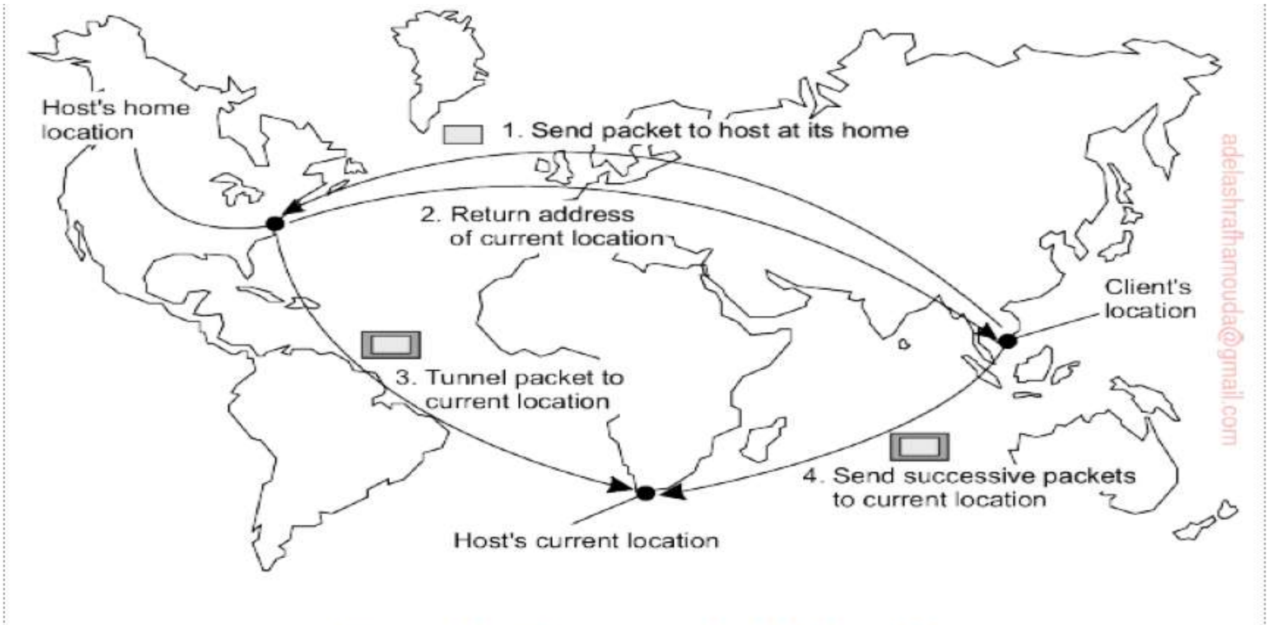
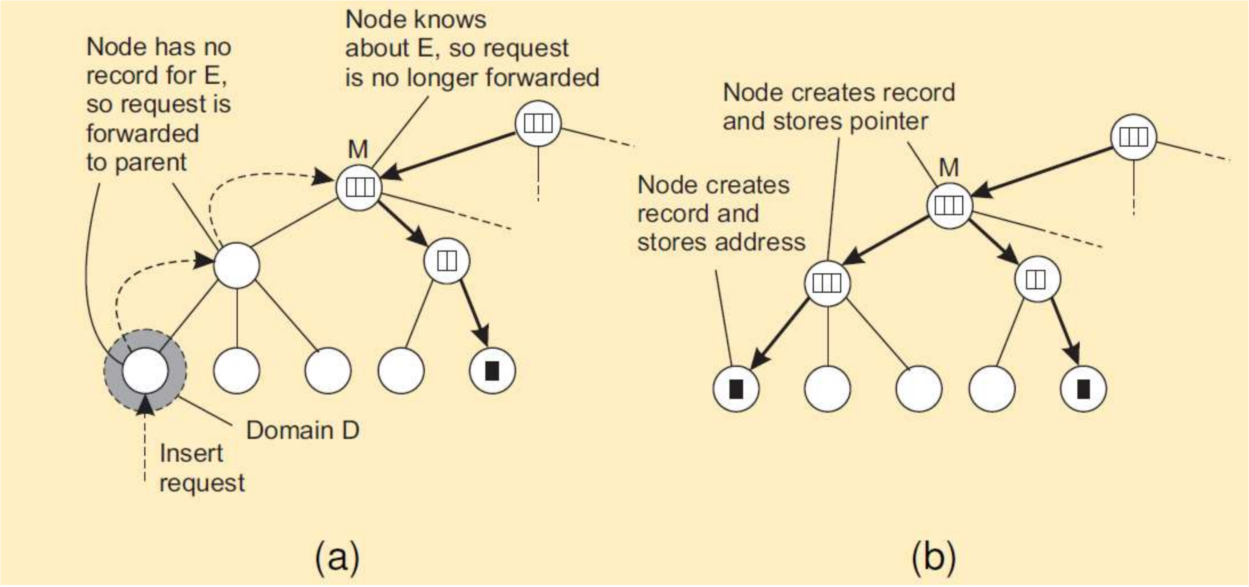

Host's home location
1. Send packet to host at its home
2. Return address of current location
Client's location
3. Tunnel packet to current location
4. Send successive packets to current location
Host's current location

Figure 5.3: The principle of Mobile IP.

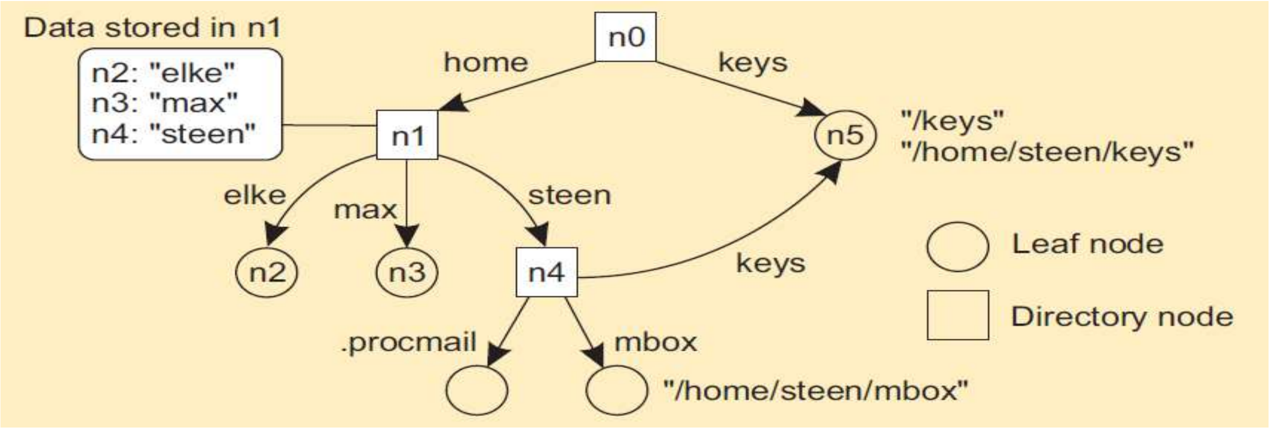
Question 2 Draw HLS: Insert operation

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

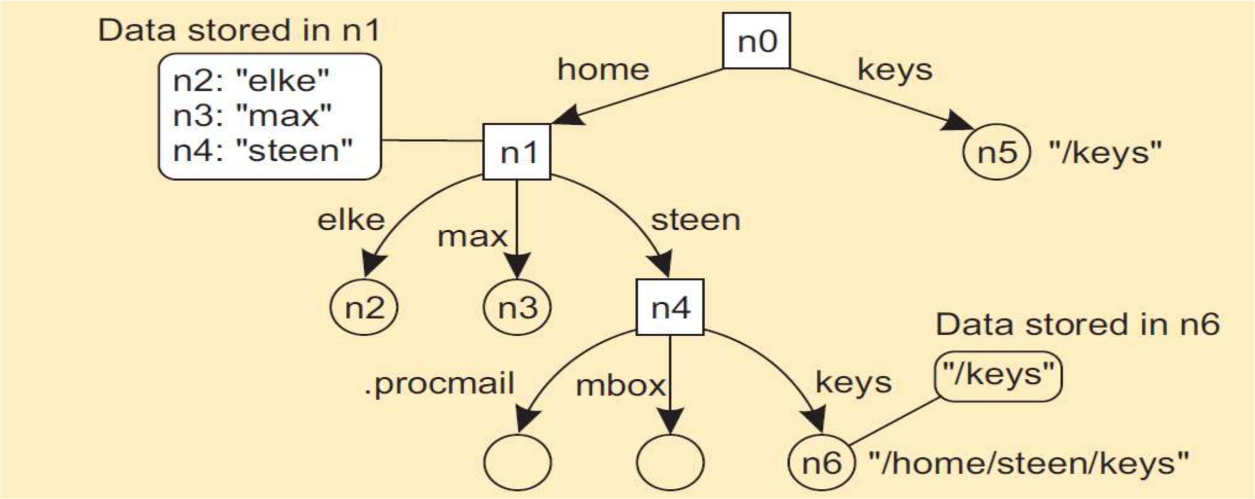
(b) A chain of forwarding pointers to the leaf node is created


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
M
Node creates record and stores address
(b)

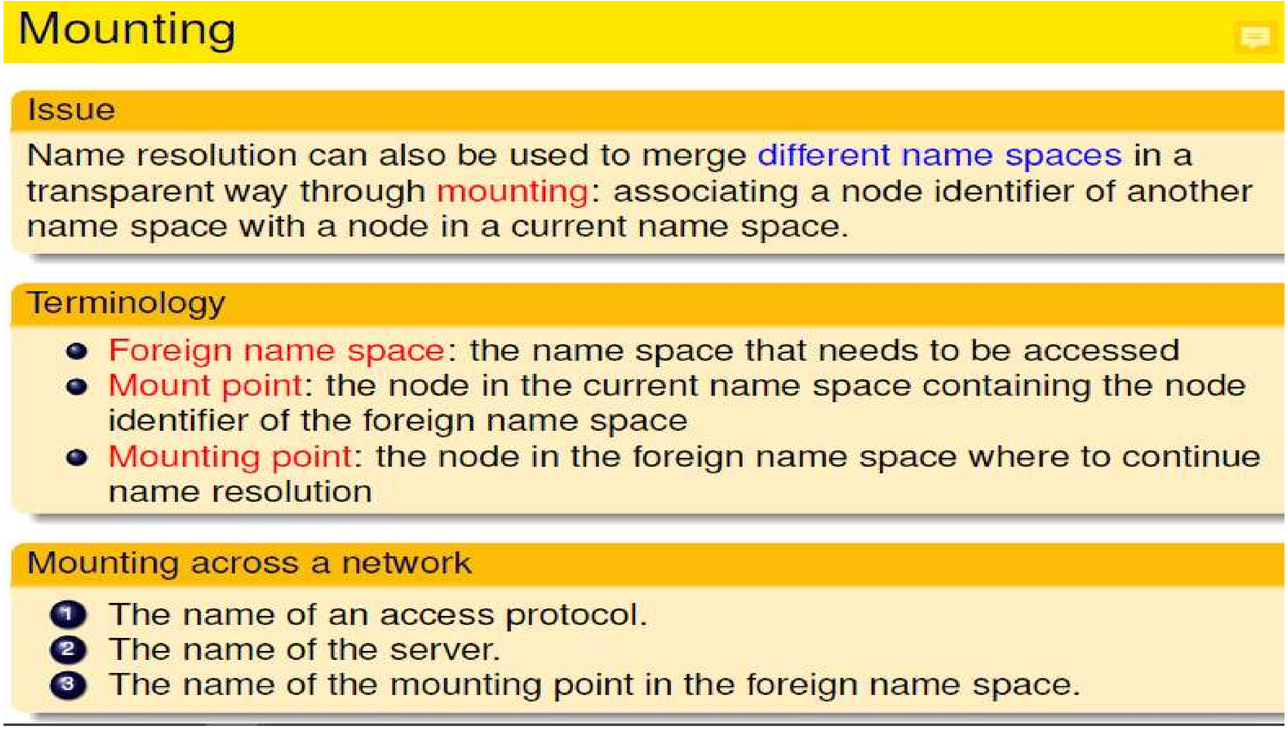
Question 3 Draw A general naming graph with a single root node



Question 4 Draw The concept of a symbolic link explained in a naming graph



Question 5 Fill



Mounting

Issue

Name resolution can also be used to merge different name spaces in a transparent way through mounting: associating a node identifier of another name space with a node in a current name space.

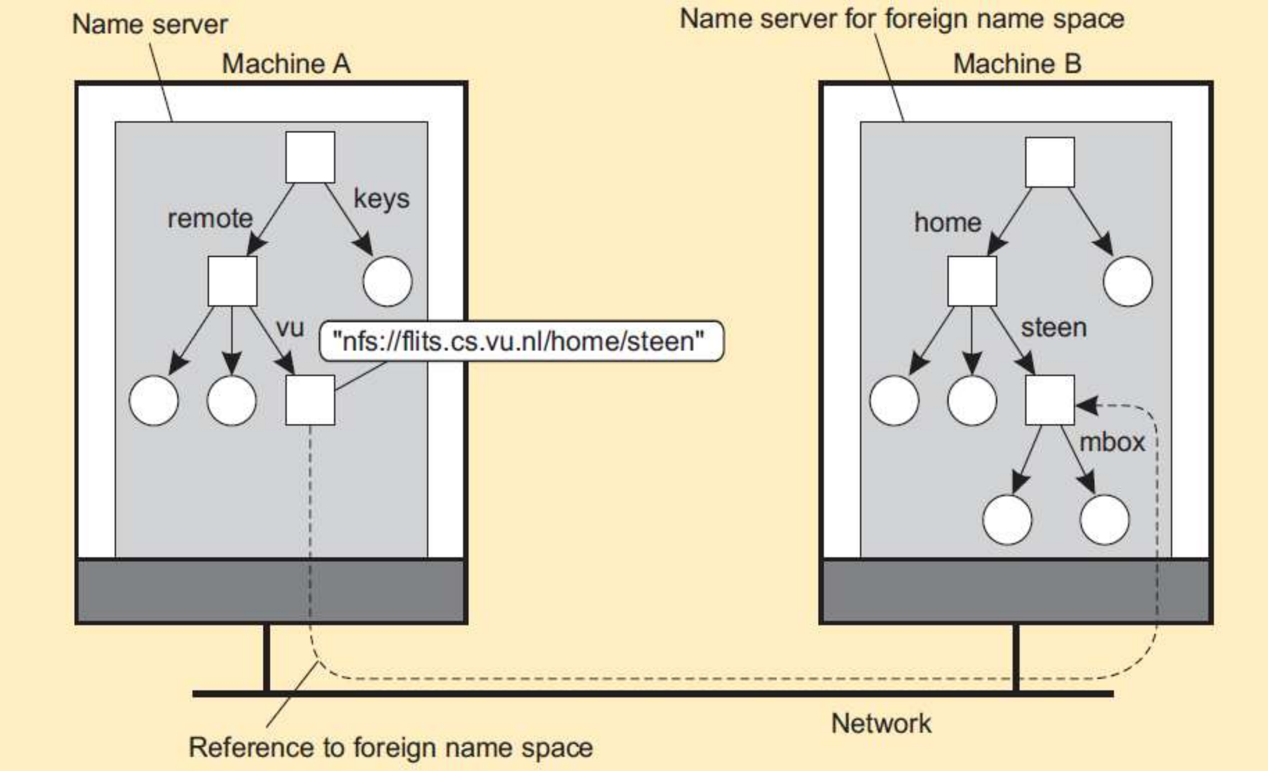
Terminology

- Foreign name space: the name space that needs to be accessed
- Mount point: the node in the current name space containing the node identifier of the foreign name space
- Mounting point: the node in the foreign name space where to continue name resolution

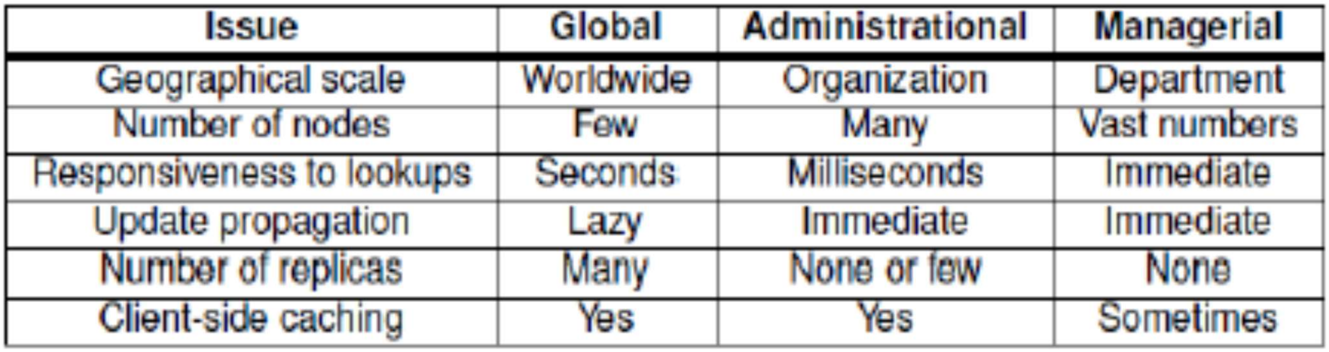
Mounting across a network

1. The name of an access protocol.
2. The name of the server.
3. The name of the mounting point in the foreign name space.

Question 6 Draw Mounting remote name spaces through a specific access protocol


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

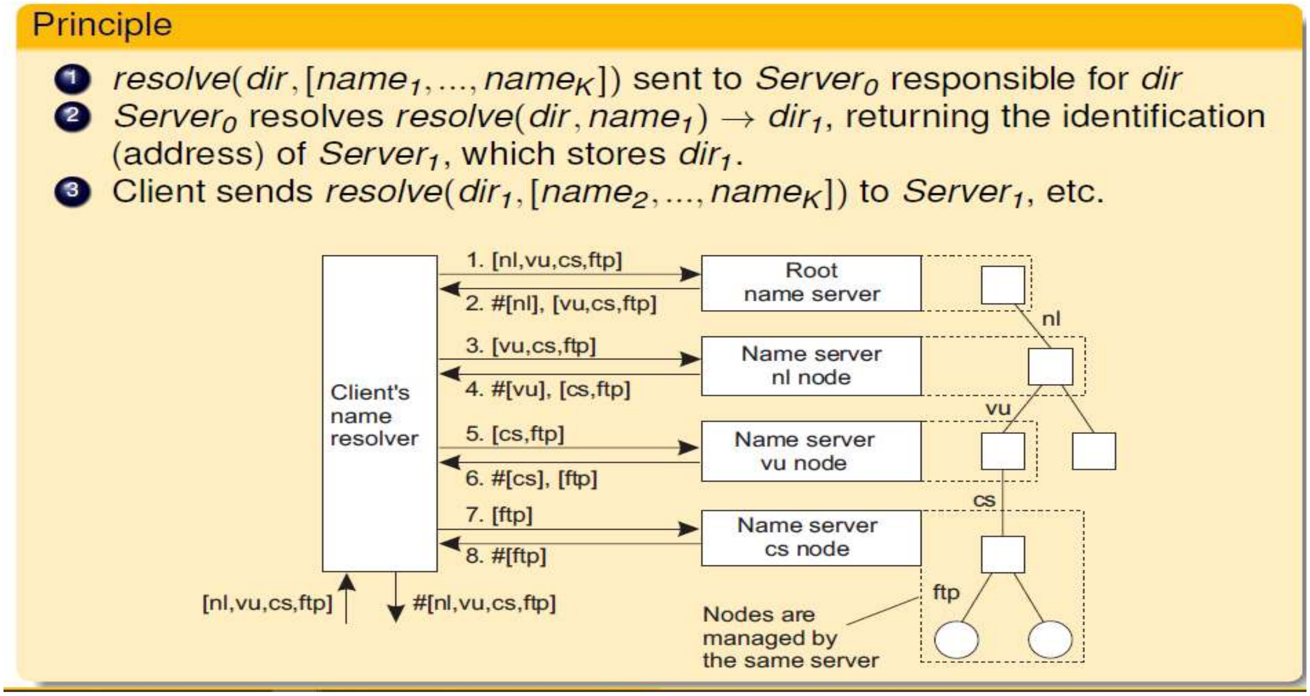
Question 7 Compare between name servers for implementing nodes in a name space



Issue	Global	Administrational	Managerial
Geographical scale	Worldwide	Organization	Department
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
Client-side caching	Yes	Yes	Sometimes

Figure 5.16: A comparison between name servers for implementing nodes from a large-scale name space partitioned into a global layer, an administrational layer, and a managerial layer.

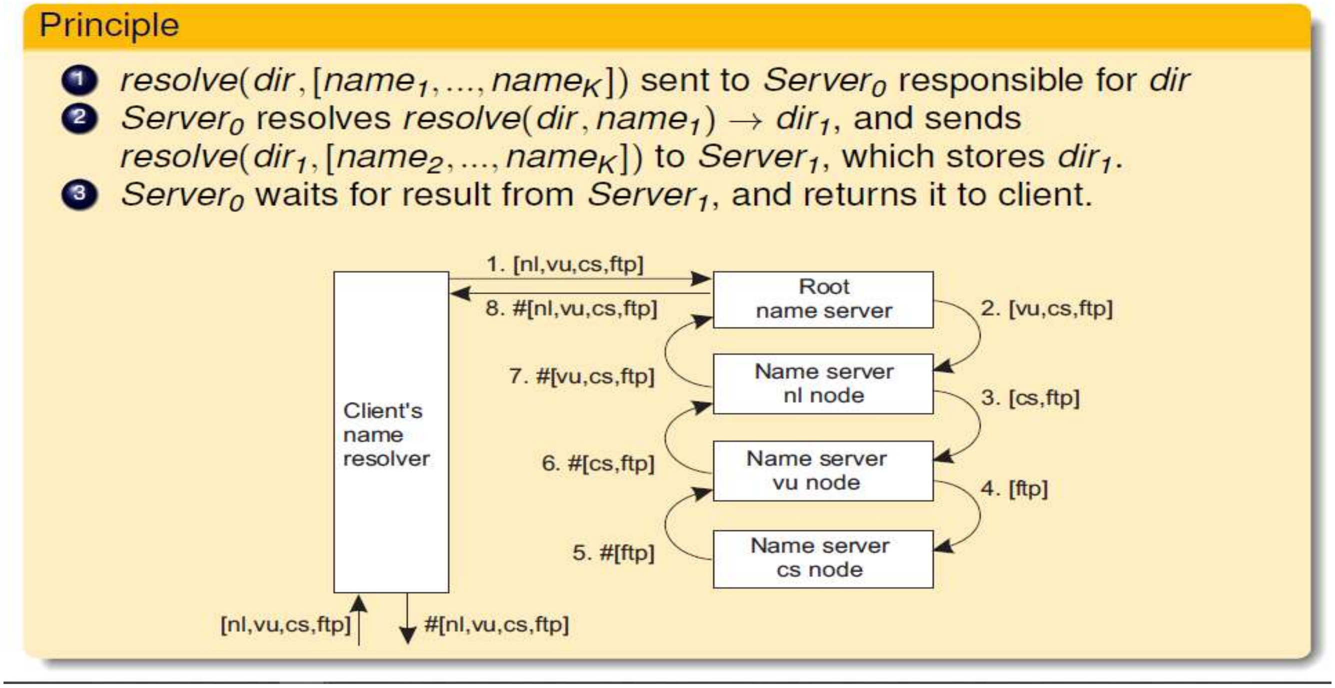
Question 8 Draw Iterative name resolution



Principle

1. $resolve(dir, [name_1, ..., name_K])$ sent to $Server_0$ responsible for dir
2. $Server_0$ resolves $resolve(dir, name_1) \rightarrow dir_1$, returning the identification (address) of $Server_1$, which stores dir_1.
3. Client sends $resolve(dir_1, [name_2, ..., name_K])$ to $Server_1$, etc.

Question 9 Draw Recursive name resolution



Principle

1. $resolve(dir, [name_1, ..., name_K])$ sent to $Server_0$ responsible for dir
2. $Server_0$ resolves $resolve(dir, name_1) \rightarrow dir_1$, and sends $resolve(dir_1, [name_2, ..., name_K])$ to $Server_1$, which stores dir_1.
3. $Server_0$ waits for result from $Server_1$, and returns it to client.

Question 10

Example on attribute-based naming: **Microsoft's active directory service**

Question 11

Each index is to be mapped to a server, who keeps a reference to the associated entity. One possible solution: **use a Distributed hash tables (DHT)**