



Distributed Systems

Aleardo Manacero Jr.



UNIVERSITY OF OREGON





Distributed File Systems



Introduction



- Distributed file systems can be seen as managers controlling files like distributed objects
- They add properties such as persistence, replication, and consistence which are not present in most systems
- In order to provide these properties some additional mechanisms have to be present in a distributed file system



Introduction



- Before going over DFS details, it is interesting to remember what modules are part of a FS

Directory	Relates file names to file IDs
File	Relates file IDs to particular files
Access control	Checks permission for operation requested
File access	Reads or writes file data or attributes
Block	Accesses and allocates disk blocks
Device	Performs disk I/O and buffering



Introduction



- Files have attributes, which form their metadata, along directory information
- A typical FS must provide operations to open, close, create, read, write, and seek a file
- All operations are executed based on rights of access and permissions, which are controlled by the access control module



Distributed File Systems requirements



- DFS must offer:
 - Transparency
 - Concurrent updates
 - File replication
 - Heterogeneity
 - Fault tolerance
 - Consistency
 - Security
 - Efficiency



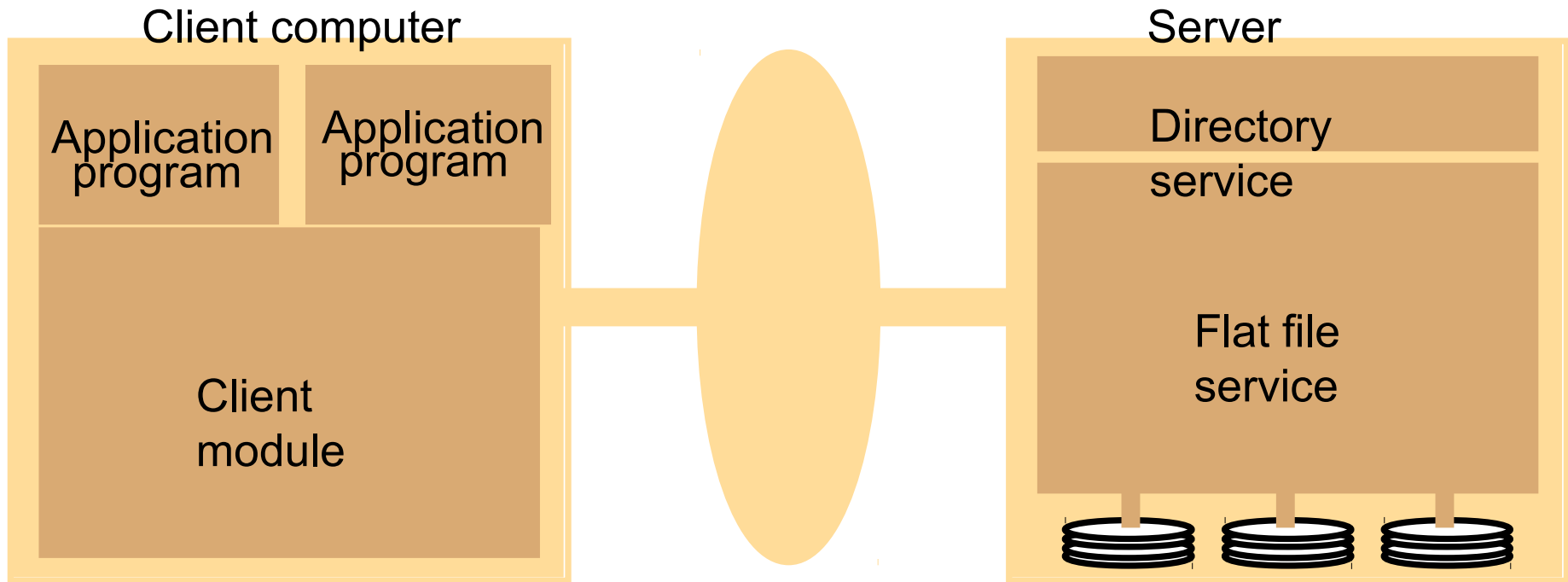
File service architecture



- A typical architecture for a DFS contains three major components:
 - Directory service (runs on server)
 - Flat file service (runs on server)
 - Client module (runs on client)



File service architecture



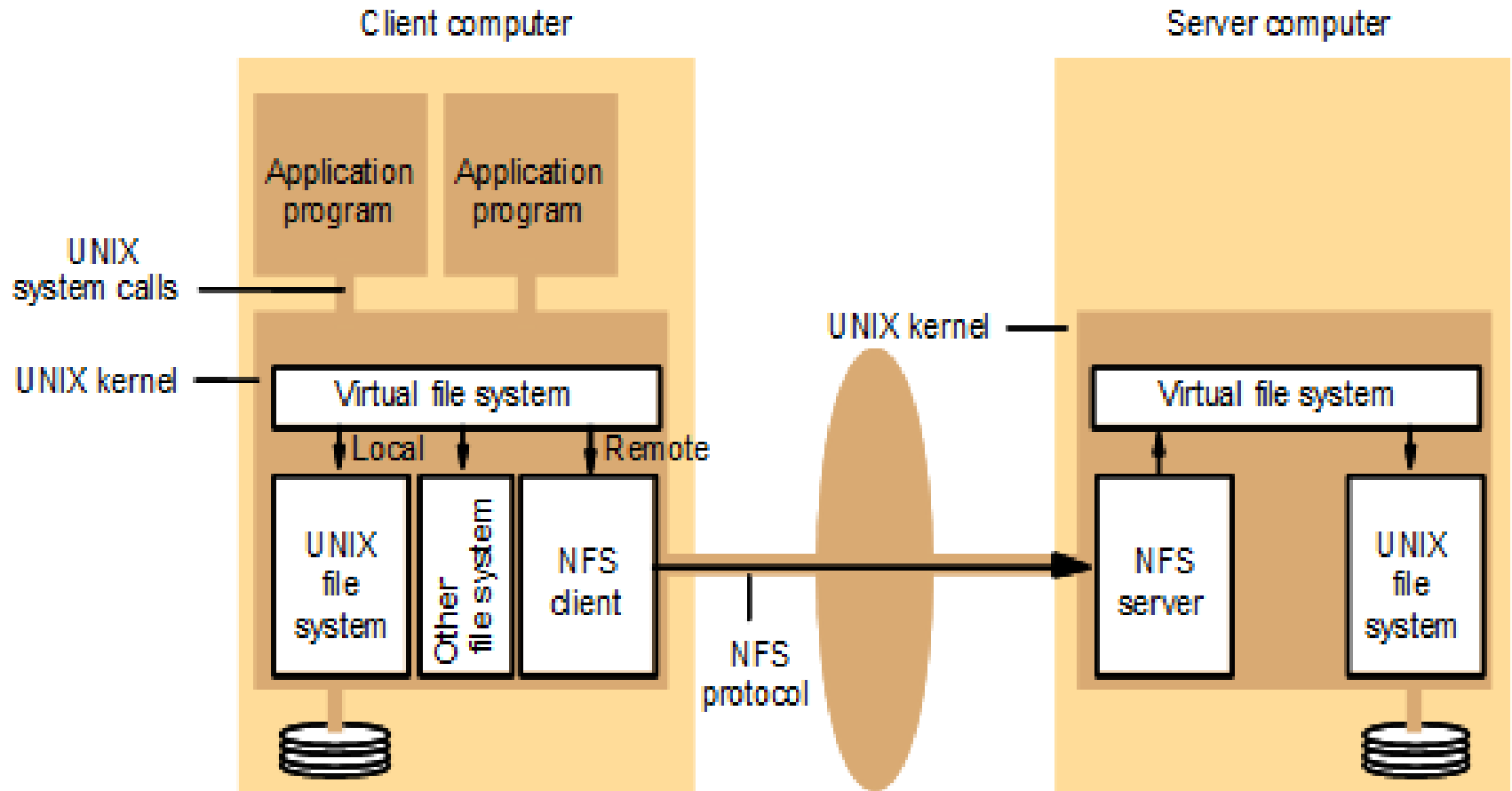
NFS architecture



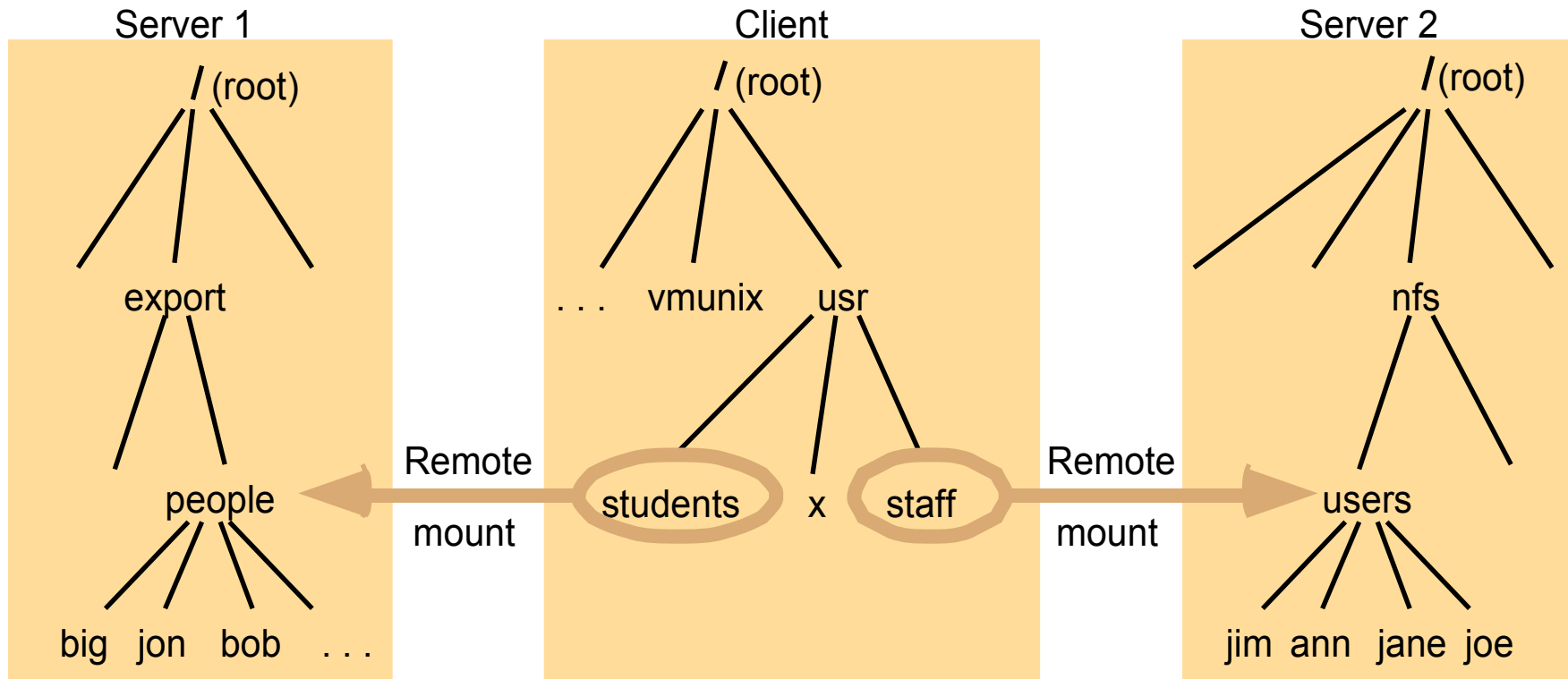
- One of the most known DFS is the Network File System – NFS – designed by Sun Microsystems
- NFS is built over the UNIX base, avoiding specific constructions to provide file access
- Specific characteristics include the authentication procedure (per access) and mounting (per client)



NFS architecture



Remote mounts



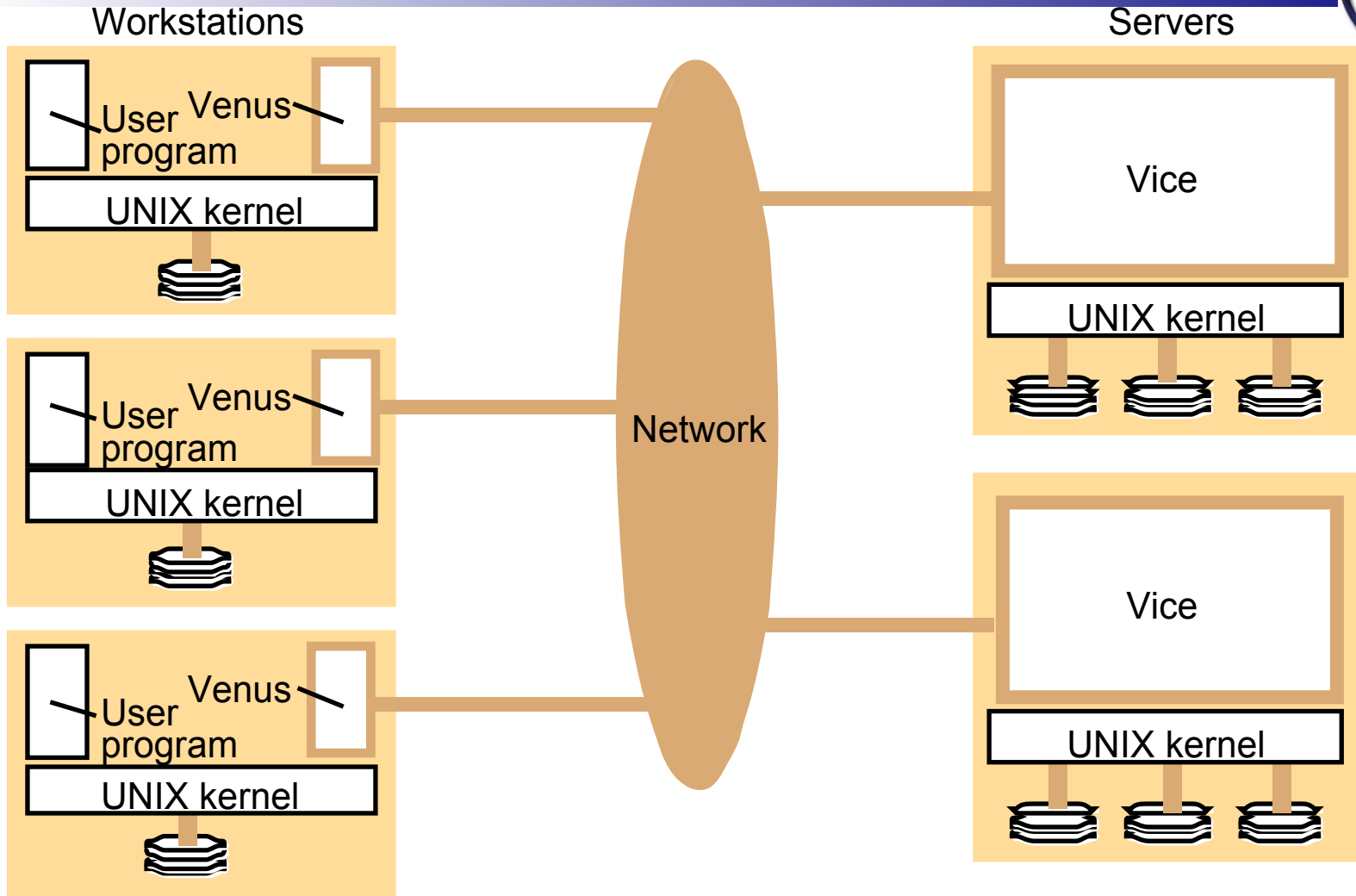
AFS architecture



- Another example of DFS is the Andrew File System – AFS – which differs from NFS by providing scalability as its major goal
- In order to have a better performance AFS provides caching of whole files
- Cached copies are created in the client disks and are permanent



AFS architecture

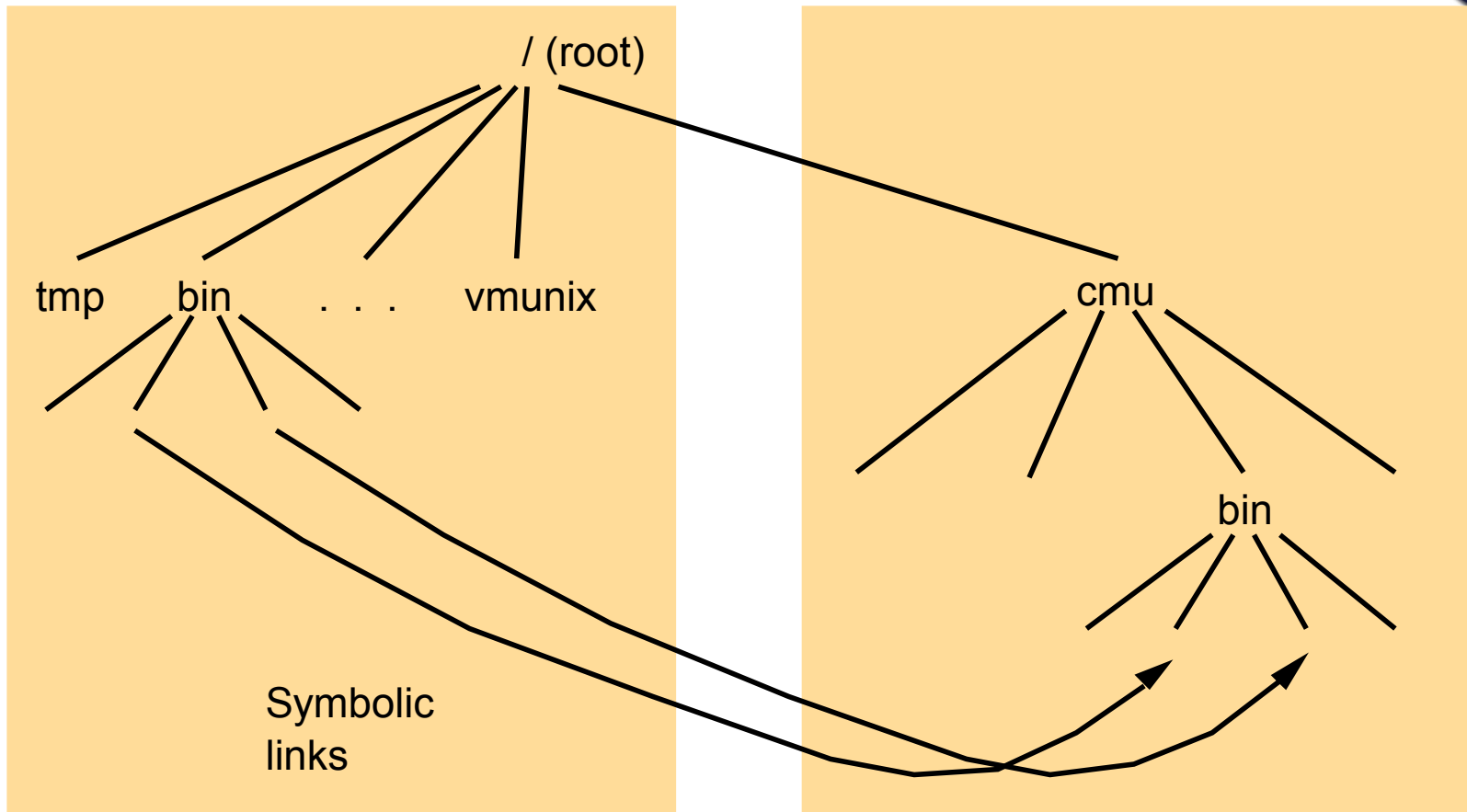


File name space in AFS



Local

Shared





Name services



Introduction



- Distributed files need to be identified and located in the system
- The same problem can be matched to hosts, services, objects, etc.
- Name services provide the solution for these problems
- DNS is one of the services providing naming



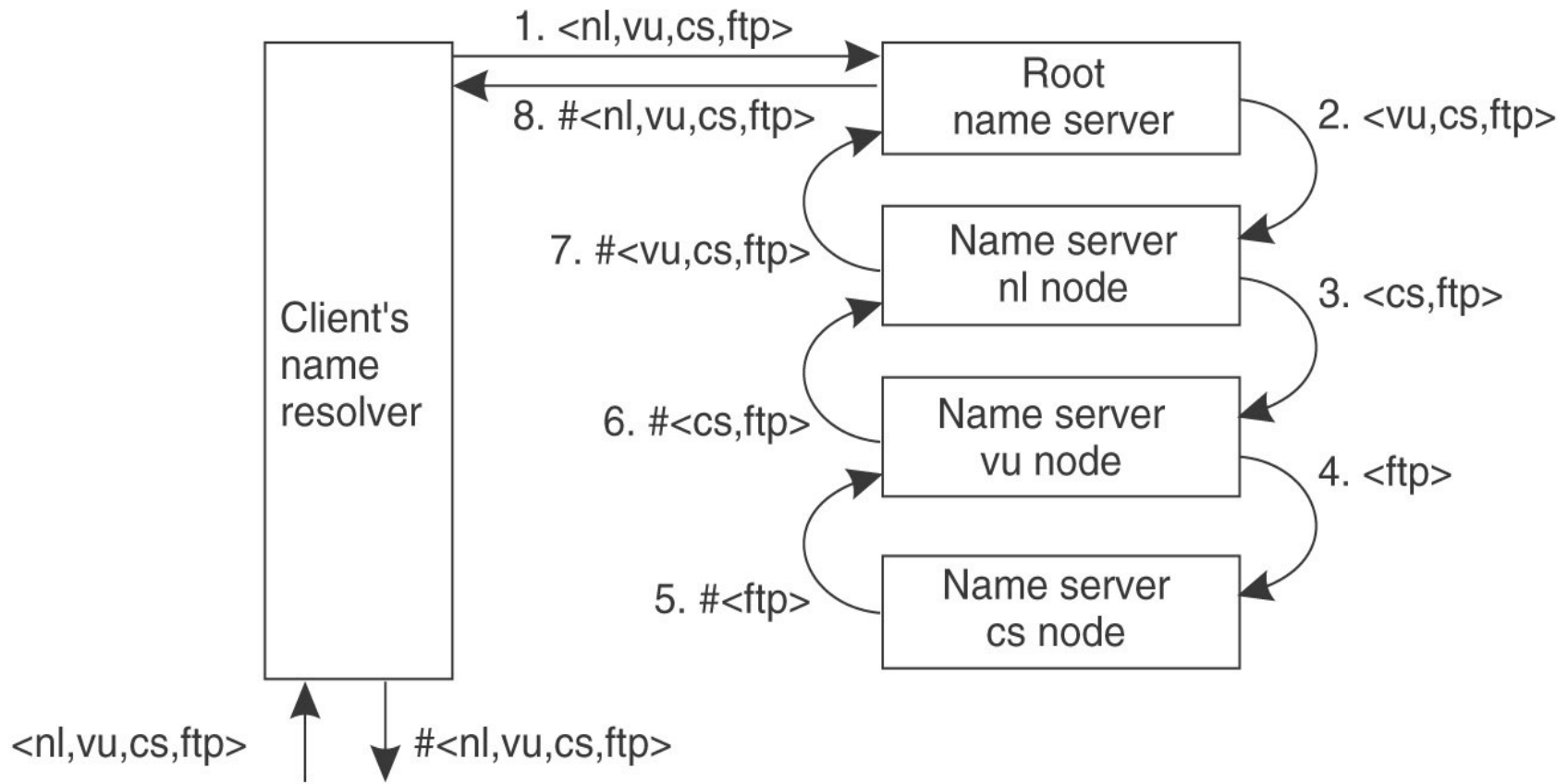
Names and addresses



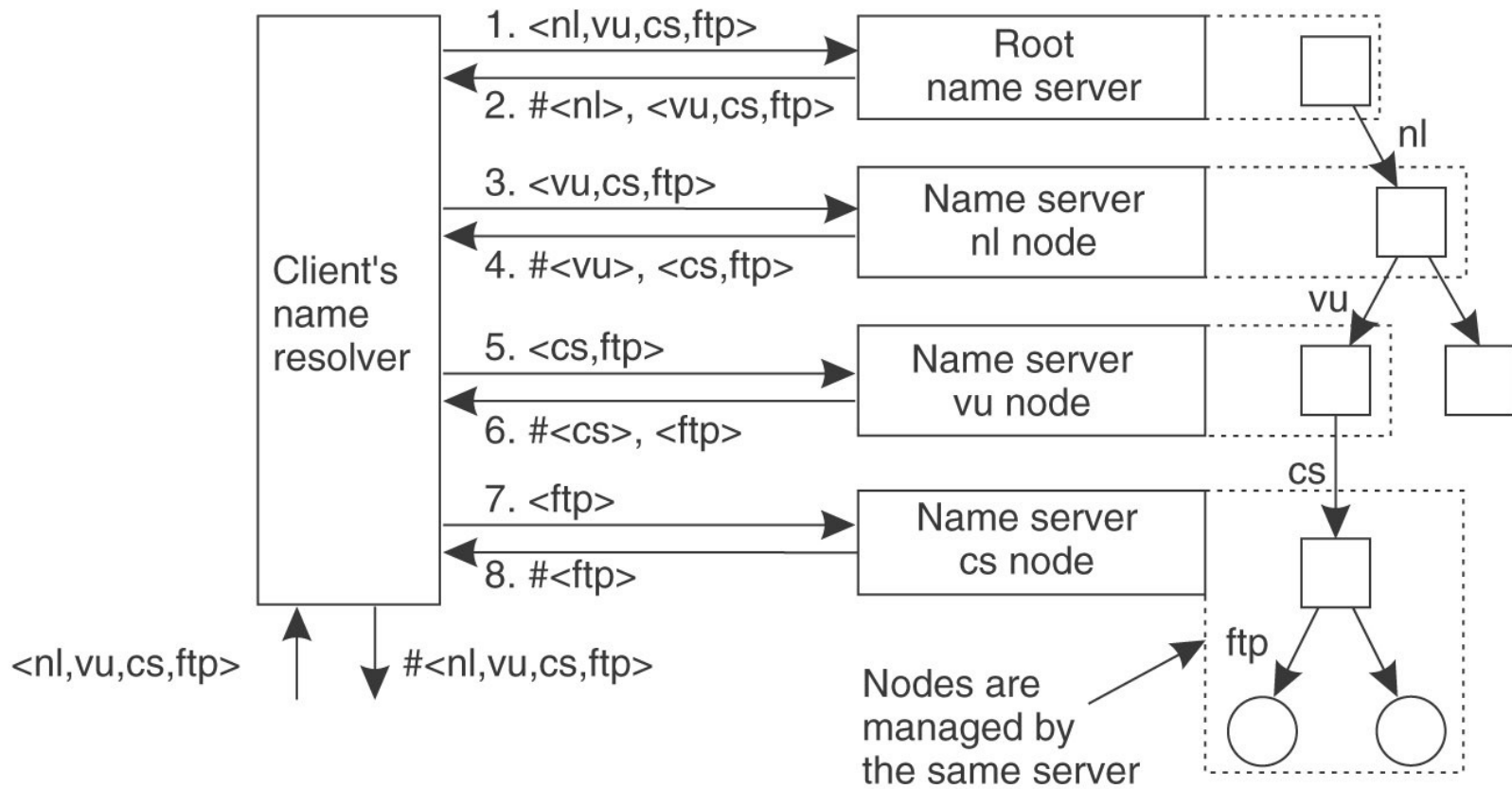
- Names are a mean to identify an object in the distributed system
- A name can be pure, formed by simple bit patterns, or non-pure, when it includes additional information about the object (location, for example)
- Name resolution is the procedure to link a name to a real file/object in the system



Name resolution – recursive approach



Name resolution – iterative approach



Name management in DNS



- In order to make the name resolution efficient, names are classified by zones, which tells:
 - Use hierarchies among domains/subdomains
 - The names and addresses for at least two authoritative name servers
 - The name servers holding authoritative data for subdomains
 - Management parameters, such as replication of zone data

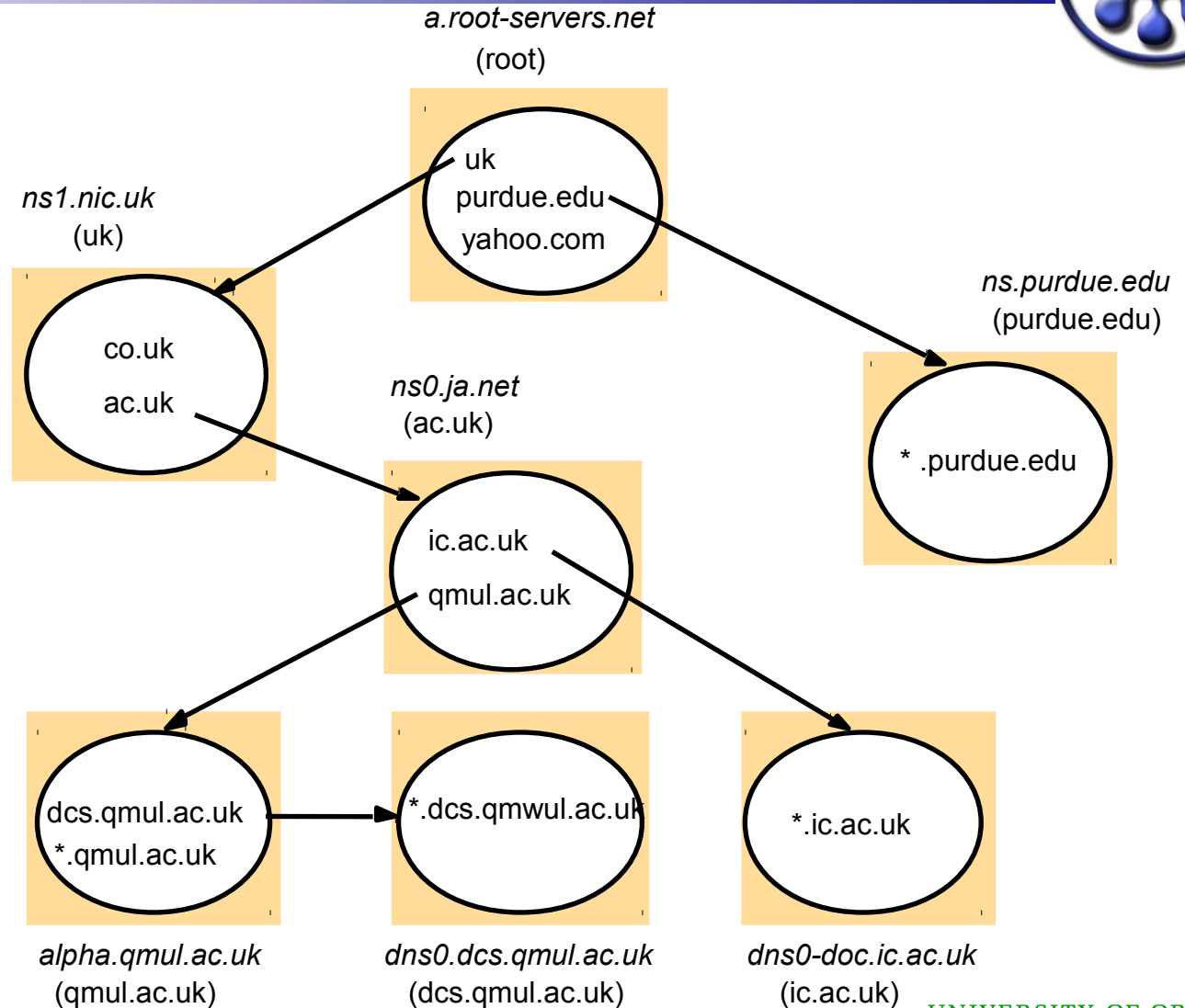


DNS name servers



Note: Name server names are in italics, and the corresponding domains are in parentheses.

Arrows denote name server entries



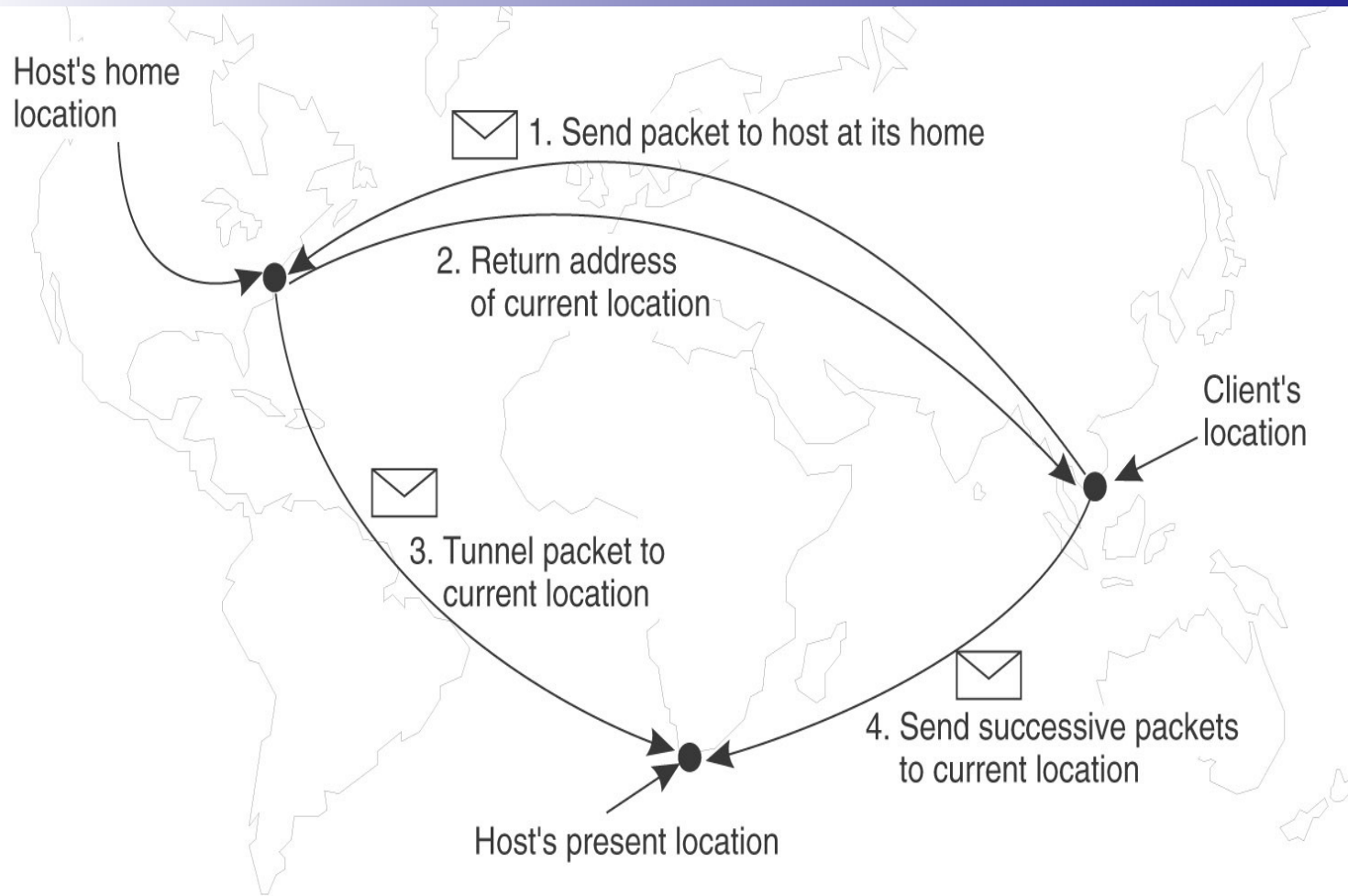
Name resolution and mobility



- Mobile devices pose a new challenge on name resolution
- Besides resolving name it is necessary to resolve location
- This can be done by making the current mobile address to be forwarded by the former “permanent” address



Name resolution and mobility





THAT'S IT FOR TODAY !!

