

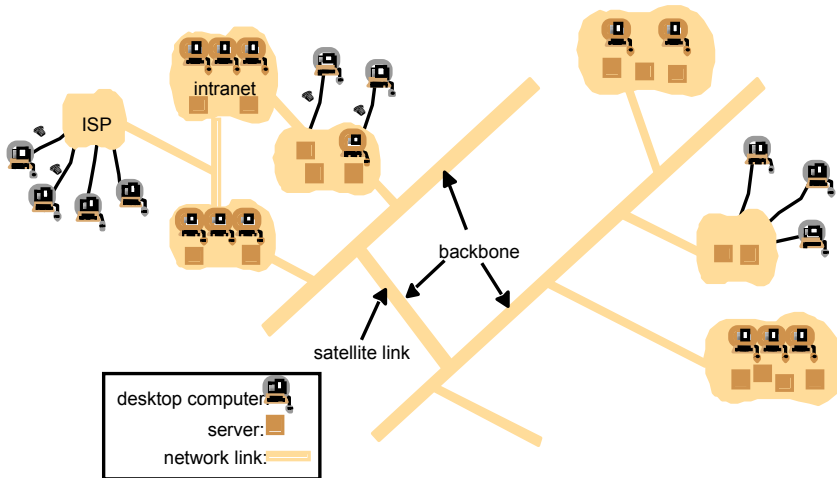
## Introduction to Distributed Computing

Operating Systems  
Prof. Sanjeev Setia

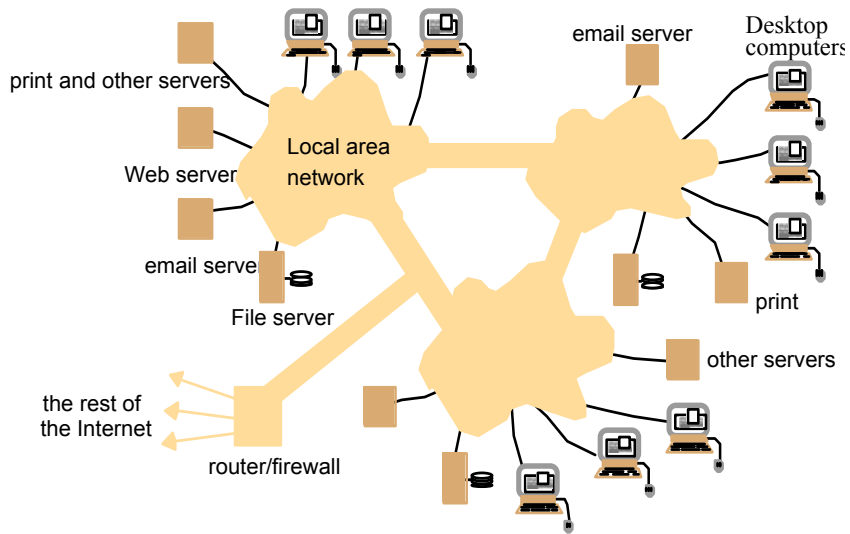
## Distributed systems

- ❑ "Workgroups"
- ❑ ATM (bank) machines
- ❑ WWW
- ❑ Multimedia conferencing
- ❑ Computing landscape will soon consist of ubiquitous network-connected devices
  - "The network is the computer"

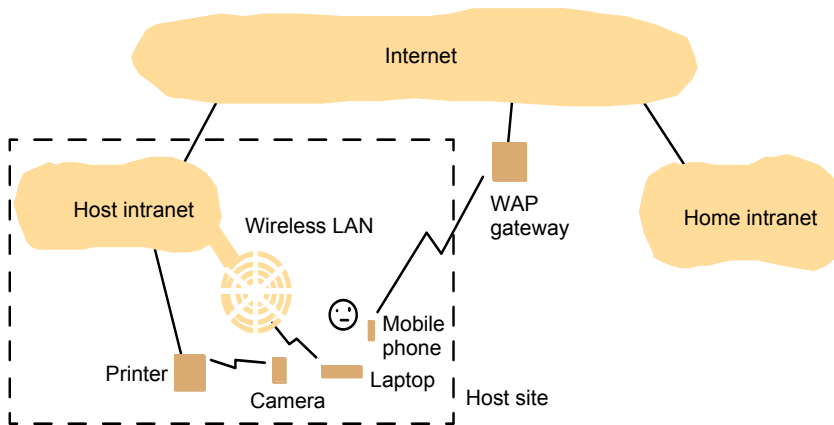
## A typical portion of the Internet



## A typical intranet



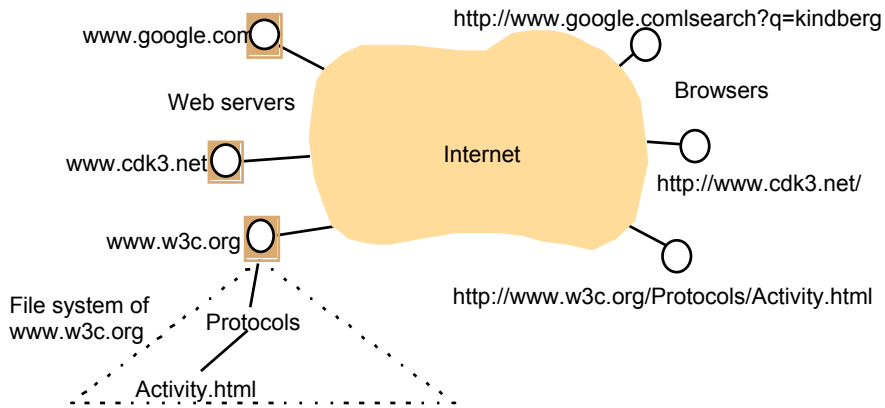
## Portable and handheld devices in a distributed system



## Distributed applications

- ❑ Applications that consist of a set of processes that are distributed across a network of machines and work together as an ensemble to solve a common problem
- ❑ In the past, mostly "client-server"
  - Resource management centralized at the server
- ❑ "Peer to Peer" computing represents a movement towards more "truly" distributed applications

## Web servers and web browsers



## Benefits

- ❑ Performance
  - Parallel computing a subset of distributed computing
- ❑ Scalability
- ❑ Resource sharing
- ❑ Fault tolerance and availability

## Challenges(Differences from Local Computing)

- ❑ Heterogeneity
- ❑ Latency
  - Interactions between distributed processes have a higher latency
- ❑ Memory Access
  - Remote memory access is not the same as local memory access
    - Local pointers are meaningless outside address space of process

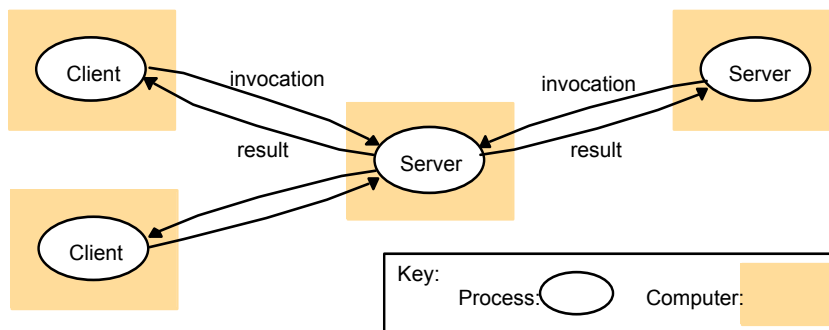
## Challenges cont'd

- ❑ Synchronization
  - Concurrent interactions the norm
- ❑ Partial failure
  - Applications need to adapt gracefully in the face of partial failure
  - Lamport once defined a distributed system as "One on which I cannot get any work done because some machine I have never heard of has crashed"

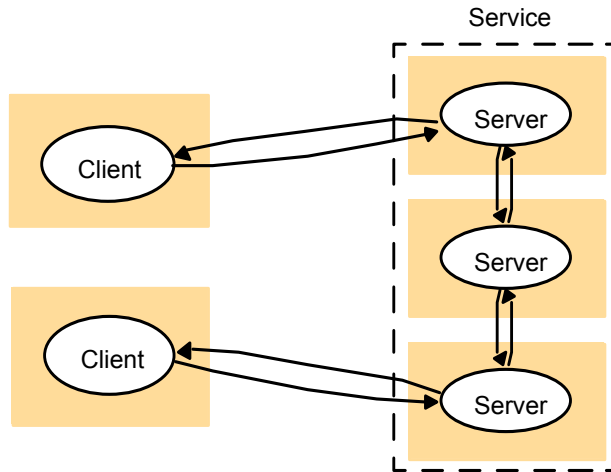
## Communication Patterns

- Client-server
- Group-oriented/Peer-to-Peer
  - Applications that require reliability
- Function-shipping/Mobile Code/Agents
  - Postscript, Java

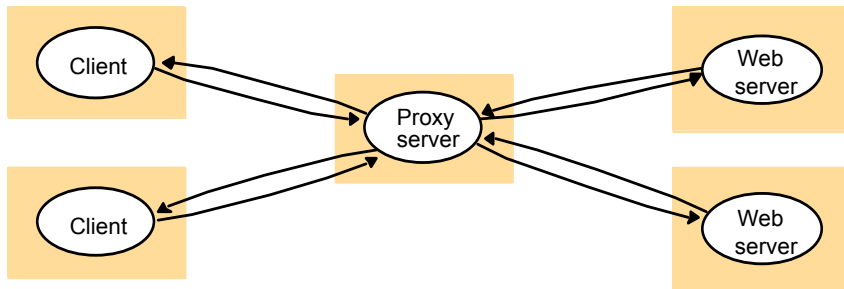
## Clients invoke individual servers



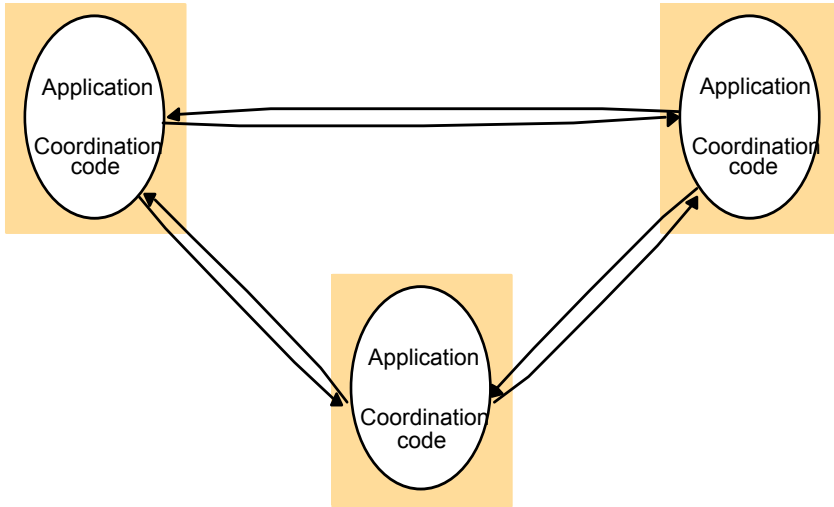
### A service provided by multiple servers



### Web proxy server

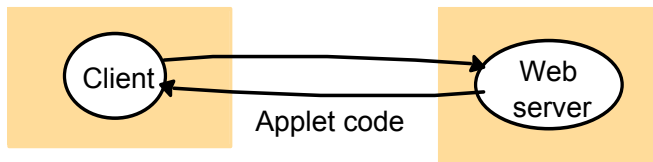


A distributed application based on peer processes



Web applets

a) client request results in the downloading of applet code

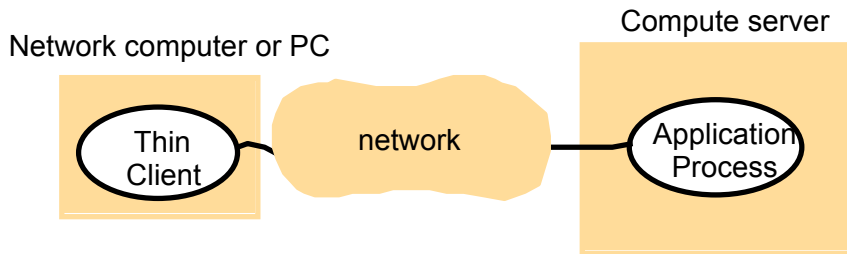


b) client interacts with the applet

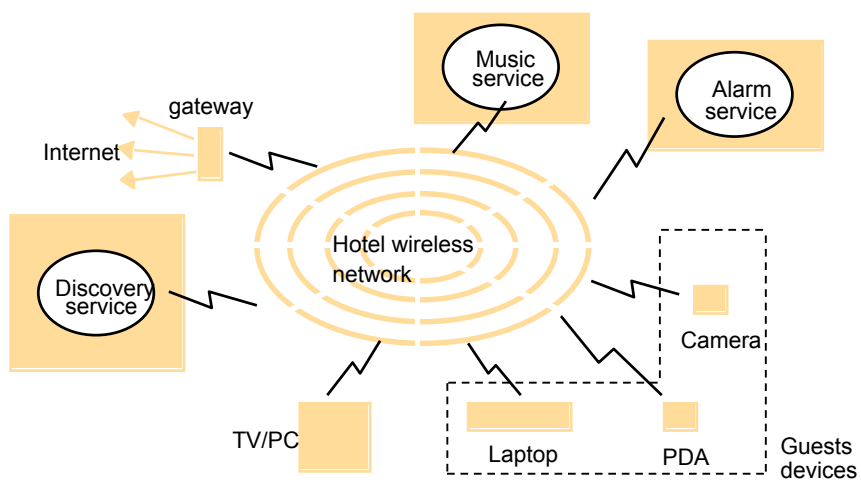




## Thin clients and compute servers



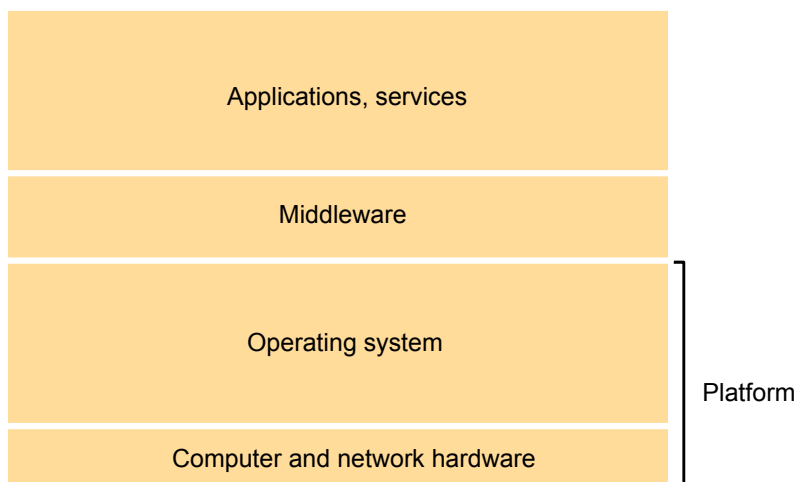
## Spontaneous networking in a hotel



## Distributed Software: Goals

- ❑ Middleware handles heterogeneity
- ❑ Higher-level support
  - Old view: Make distributed nature of application **transparent** to the user/programmer
    - Remote Procedure Calls, eg. SUN RPC, DCE RPC
    - RPC + Object orientation, e.g. DCOM, CORBA
  - Current view: Provide higher-level support **BUT expose** remote objects, partial failure, etc. to the programmer
    - Java, JINI, Javaspaces
- ❑ Scalability

## Software and hardware service layers in distributed systems



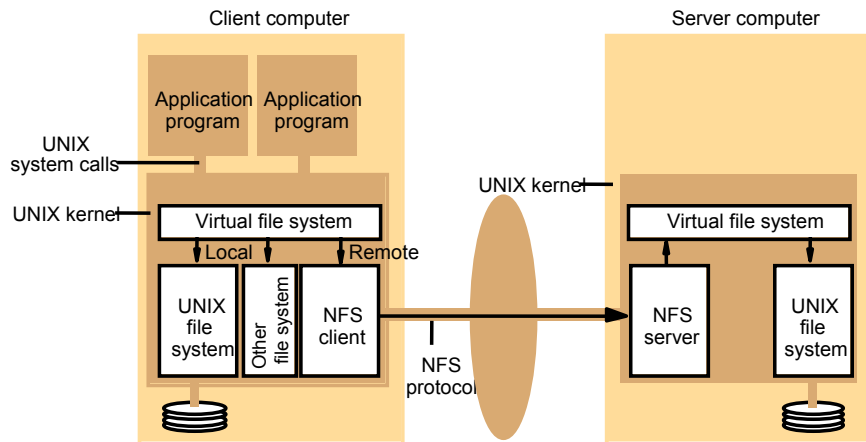
## Transparency

- Access - local and remote objects are accessed using identical operations
- Location - no knowledge of location of resource
- Concurrency - several processes can operate concurrently on shared objects without interference
- Replication - no knowledge of replicas
- Failure - graceful degradation
- Parallelism - tasks automatically parallelized

## Example: NFS

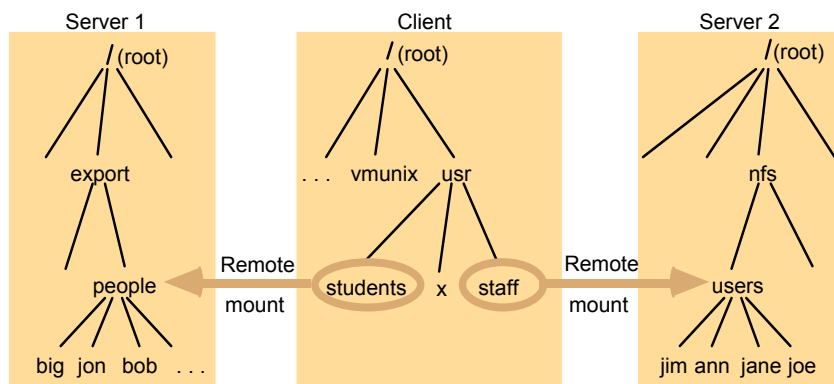
- A very successful distributed “application” based on RPC
  - Illustrates both arguments regarding the benefits and dangers of making the distributed nature of the system transparent to the user
- Interface for remote files same as interface for local files
- Soft mounts vs Hard mounts
  - Hard mounts force application to hang until server recovers
    - Unacceptable to users, lead to the introduction of soft mounts
  - Soft mounts expose network or server failures

## NFS architecture



Operating Systems CS 571 23

## Local and remote file systems accessible on an NFS client



Note: The file system mounted at `/usr/students` in the client is actually the sub-tree located at `/export/people` in Server 1; the file system mounted at `/usr/staff` in the client is actually the sub-tree located at `/nfs/users` in Server 2.

Operating Systems CS 571 24

## Scalability

- ❑ Becoming increasingly important because of the changing computing landscape
- ❑ Key to scalability: decentralized algorithms and data structures
  - No machine has complete information about the state of the system
  - Machines make decisions based on locally available information
  - Failure of one machine does not ruin the algorithm
  - There is no implicit assumption that a global clock exists

## Readings

- ❑ Chapter 1-2 of CDK3
- ❑ "A Note on Distributed Computing" - Waldo, Wyant, Wollrath, Kendall
  - Link on class web page