Introduction

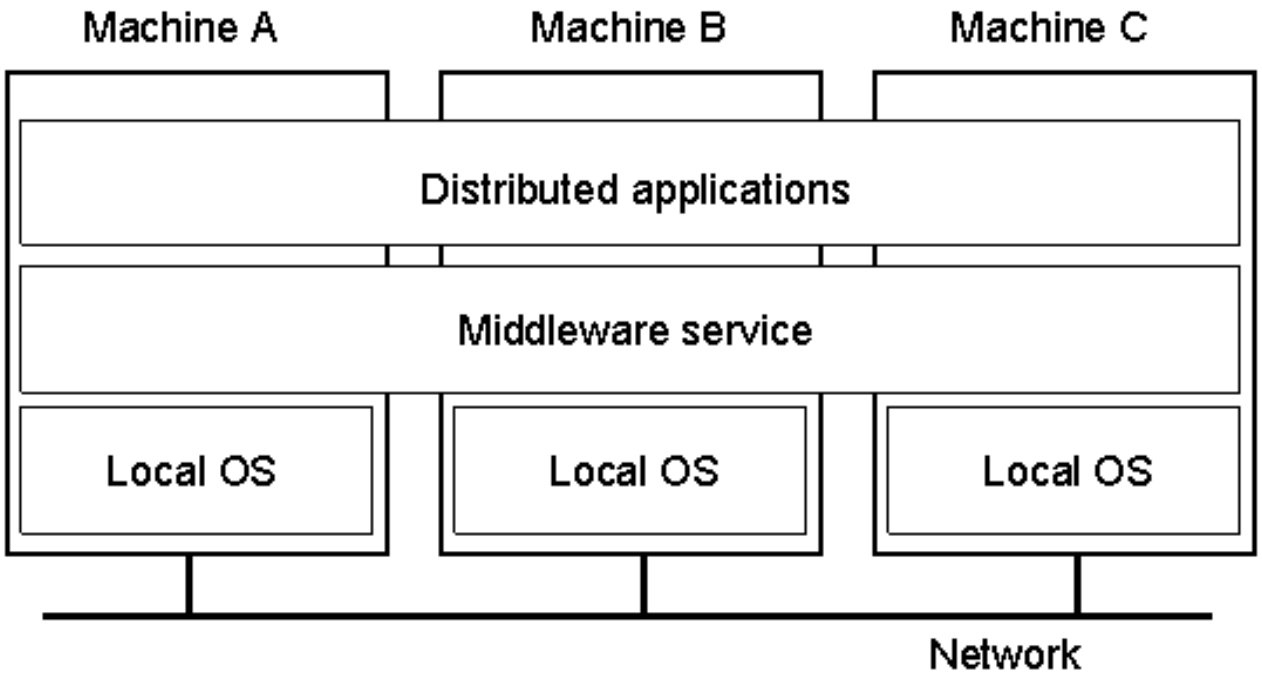
Chapter 1

Definition of a Distributed System (1)

A distributed system is:

A collection of independent computers that appears to its users as a single coherent system.

Definition of a Distributed System (2)



A distributed system organized as middleware.
Note that the middleware layer extends over multiple machines.

Examples of Distributed Systems

A network of workstations.

A workflow information system.

A flight reservation system.

The World Wide Web

Goals of Distributed Systems

Connecting users and resources.

Transparency.

Openness.

Scalability.

Transparency in a Distributed System

Transparency	Description
Access	Hide differences in data representation and how a resource is accessed
Location	Hide where a resource is located
Migration	Hide that a resource may move to another location
Relocation	Hide that a resource may be moved to another location while in use
Replication	Hide that a resource may be shared by several competitive users
Concurrency	Hide that a resource may be shared by several competitive users
Failure	Hide the failure and recovery of a resource **(see next slide)**
Persistence	Hide whether a (software) resource is in memory or on disk

Different forms of transparency in a distributed system.

Distributed System

Lamport definition:

"*you know you have one when the crash of a computer you have never heard of stops you from getting any work done.*"

Scalability Problems

Concept	Example
Centralized services	A single server for all users
Centralized data	A single on-line telephone book
Centralized algorithms	Doing routing based on complete information

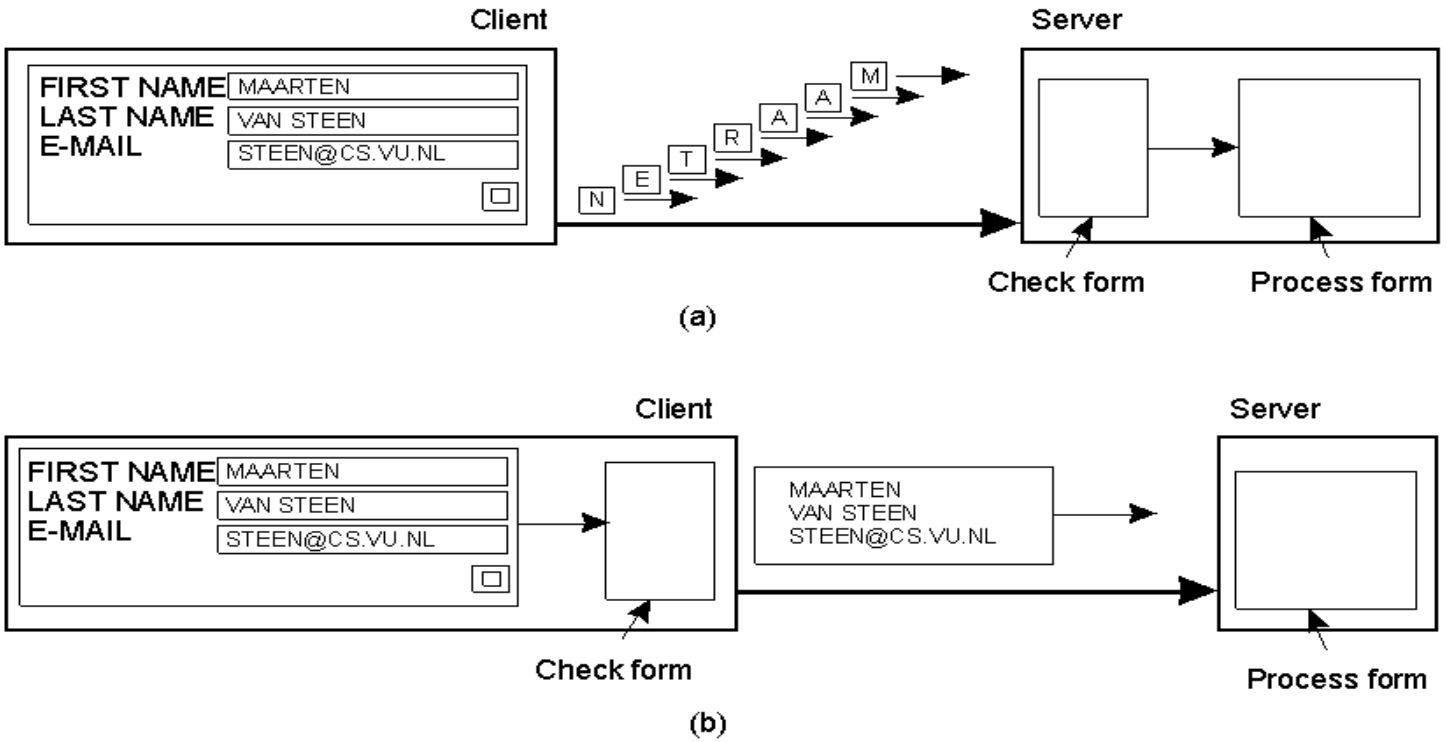
Examples of scalability limitations.

Decentralized algorithms

- No machine has complete information about the system state.
- Machines make decisions based only on local information.
- Failure of one machine does not ruin the algorithm.
- There is no implicit assumption that a global clock exists.

The larger the system, the larger the uncertainty.

Scaling Techniques (1)

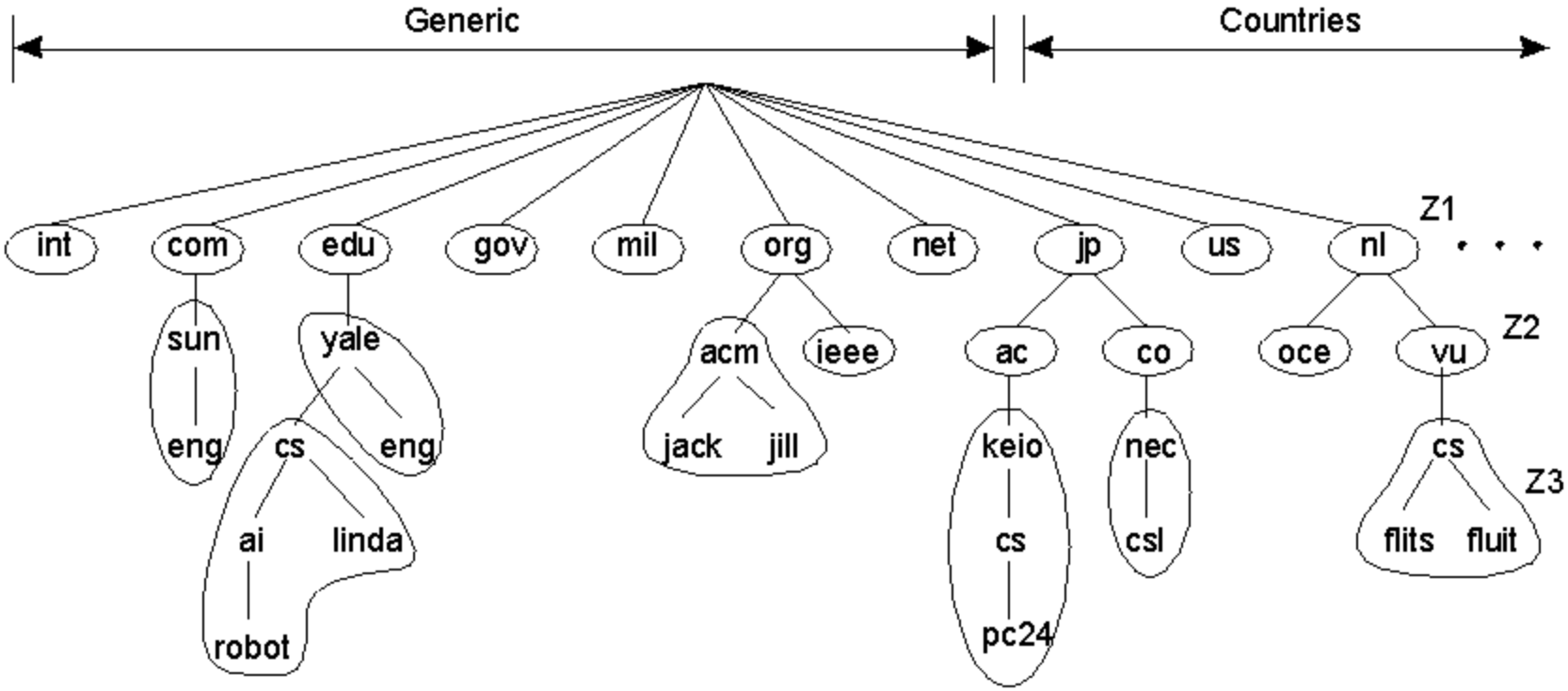


The difference between letting:

a) a server or

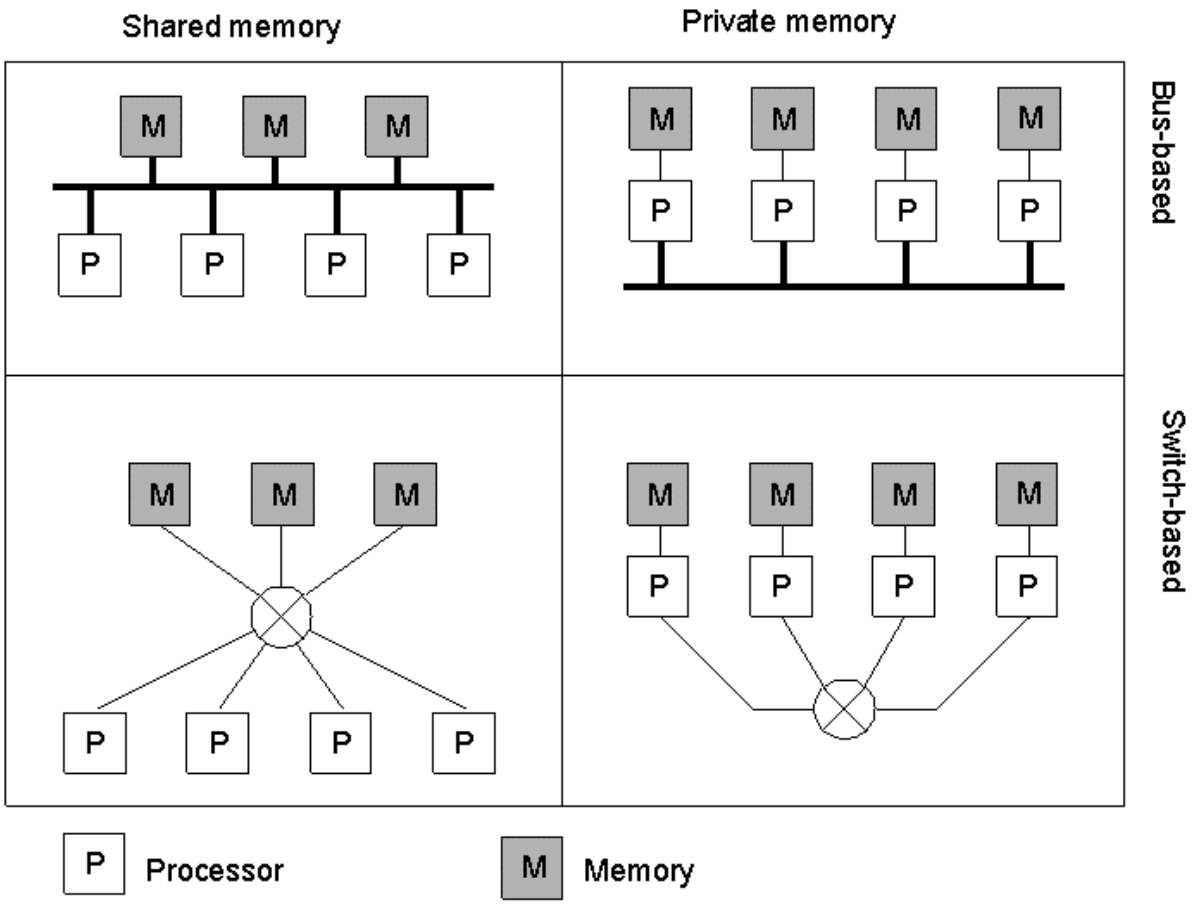
b) a client check forms as they are being filled

Scaling Techniques (2)



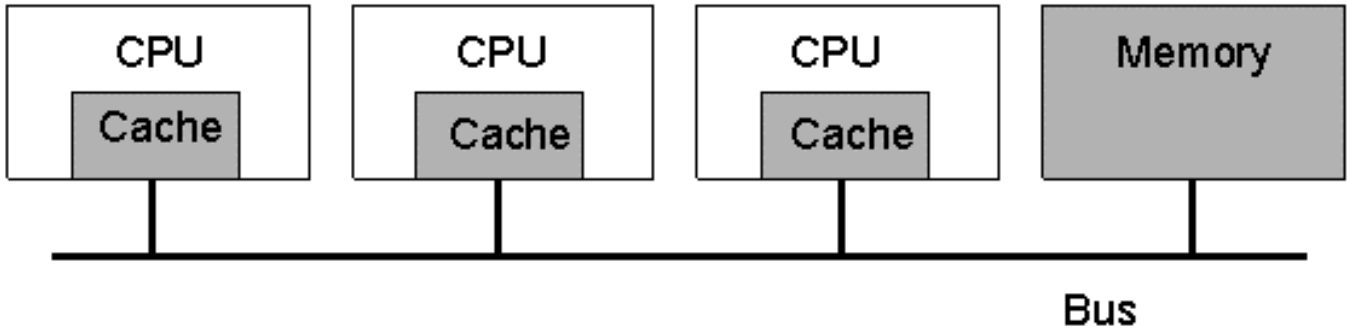
An example of dividing the DNS name space into zones.

Hardware Concepts



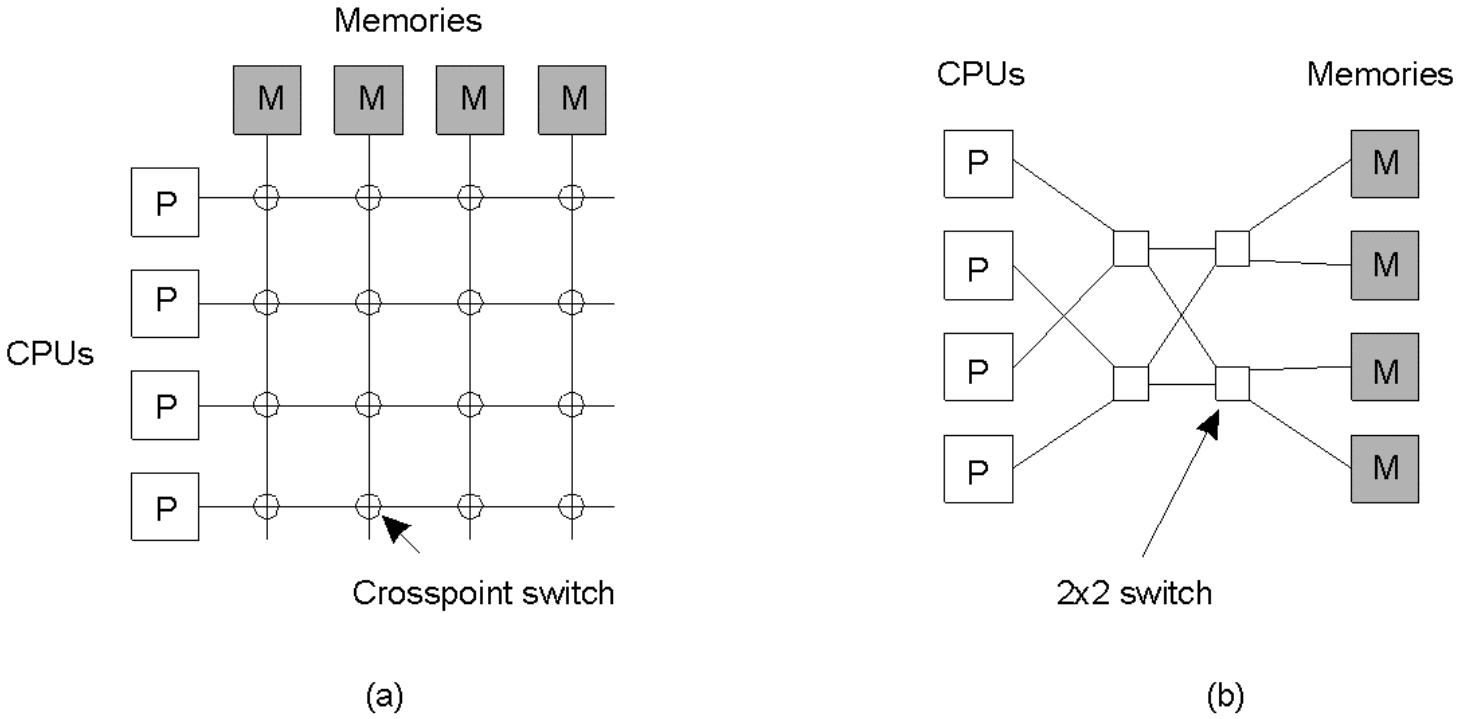
Different basic organizations and memories in distributed computer systems

Multiprocessors (1)



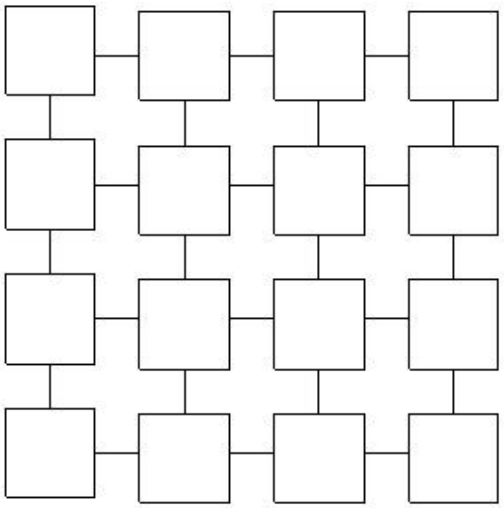
A bus-based multiprocessor.

Multiprocessors (2)

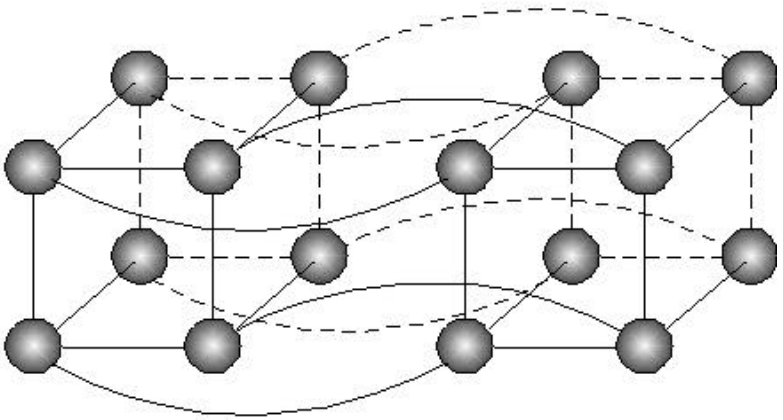


a) A crossbar switch
b) An omega switching network

Homogeneous Multicomputer Systems



(a)



(b)

a) Mesh
b) Hypercube

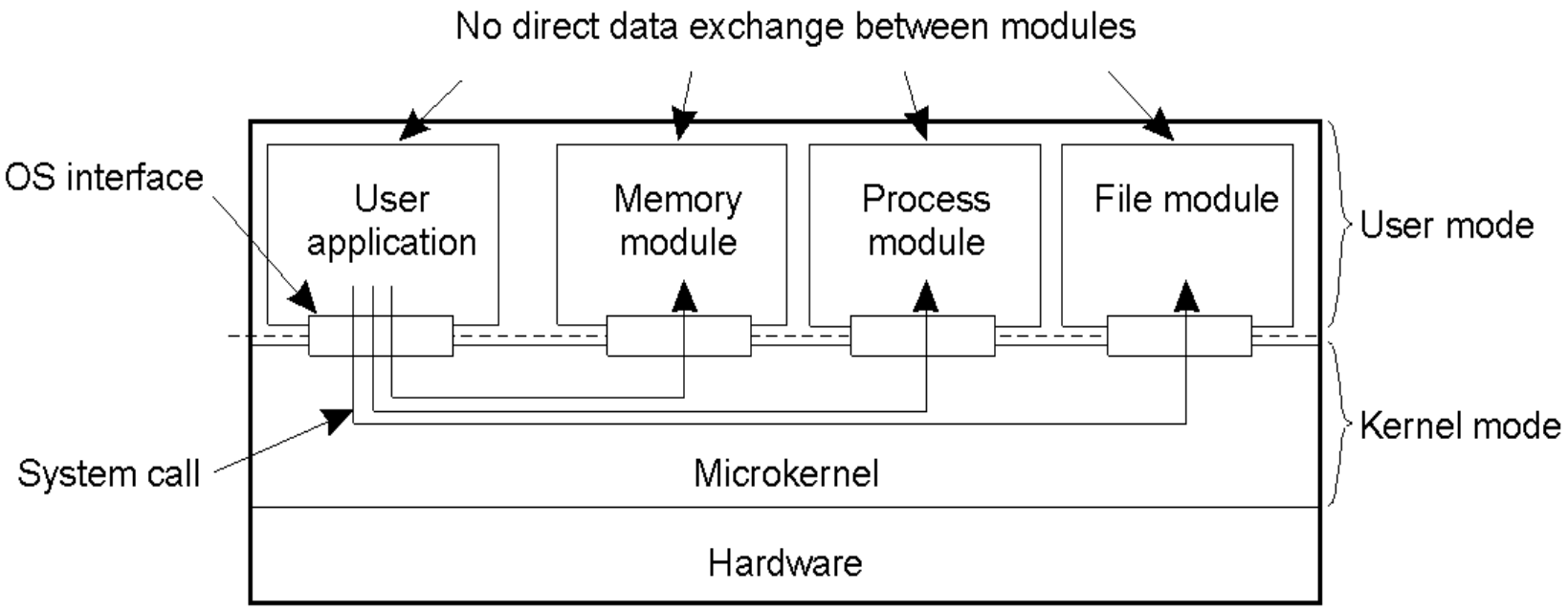
Software Concepts

System	Description	Main Goal
DOS	Tightly-coupled operating system for multi-processors and homogeneous multicomputers	Hide and manage hardware resources
NOS	Loosely-coupled operating system for heterogeneous multicomputers (LAN and WAN)	Offer local services to remote clients
Middleware	Additional layer atop of NOS implementing general-purpose services	Provide distribution transparency

An overview between

- DOS (Distributed Operating Systems)
- NOS (Network Operating Systems)
- Middleware

Uniprocessor Operating Systems

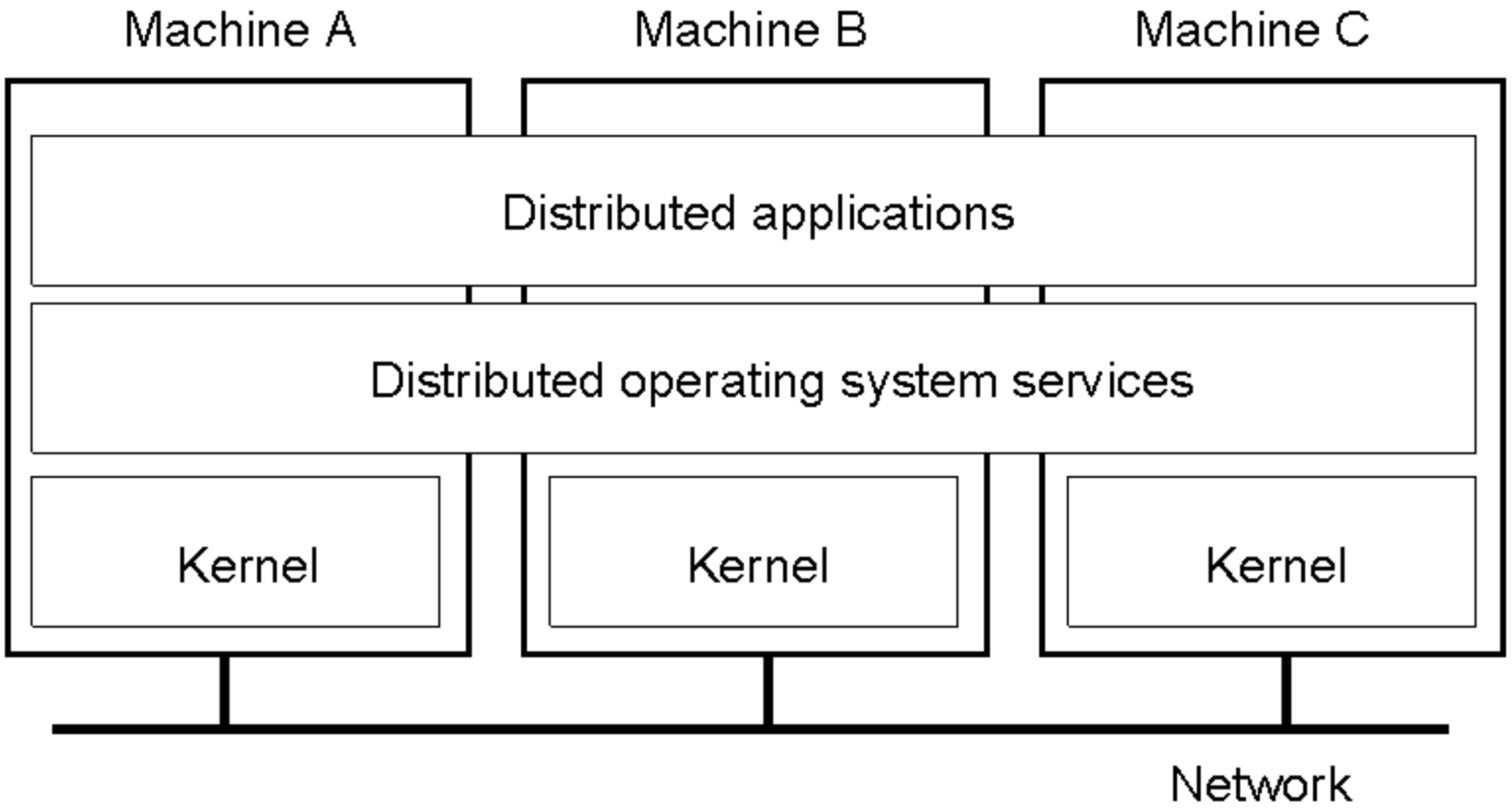


Separating applications from operating system code through a microkernel.

Multiprocessor Operating Systems (1)

- Using many CPUs and a shared memory.
- Handling concurrent access to the shared memory.
- Mechanisms: Semaphores and Monitors to protect against concurrent access.
- Easy to extend Uniprocessor Operating Systems developed as a set of concurrent processes.

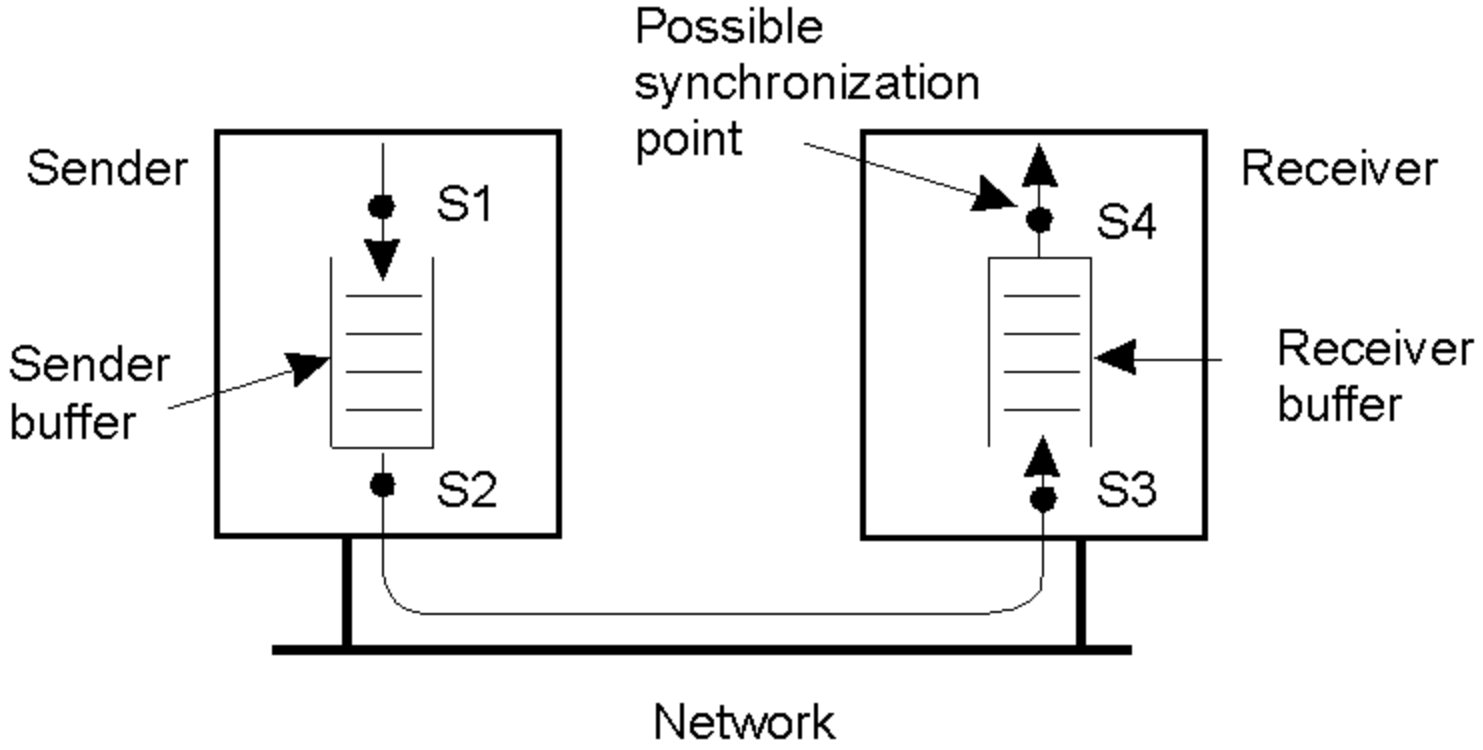
Multicomputer Operating Systems (1)



General structure of a multicomputer operating system

Multicomputer Operating Systems (2)

Message Passing



Alternatives for blocking and buffering in message passing.

Multicomputer Operating Systems (3)

Synchronization point	Send buffer	Reliable comm. guaranteed?
Block sender until buffer not full	Yes	Not necessary
Block sender until message sent	No	Not necessary
Block sender until message received	No	Necessary
Block sender until message delivered	No	Necessary

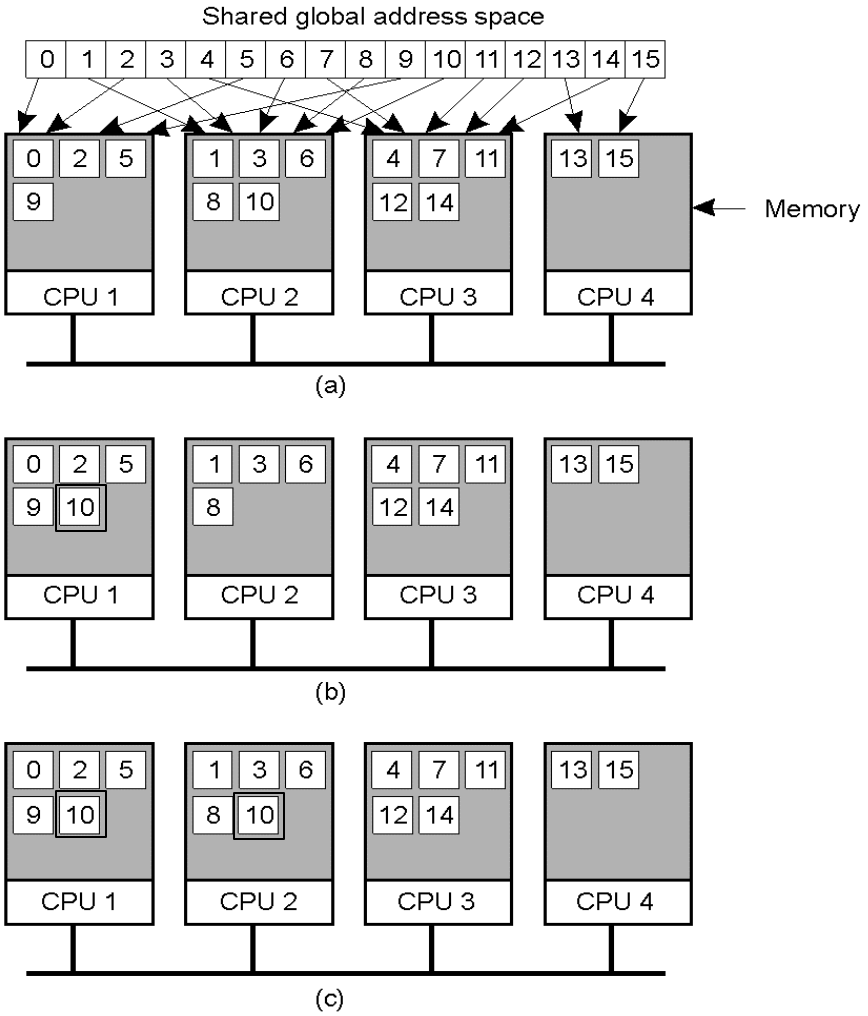
Relation between blocking, buffering, and reliable communications.

Distributed Shared Memory Systems (1)

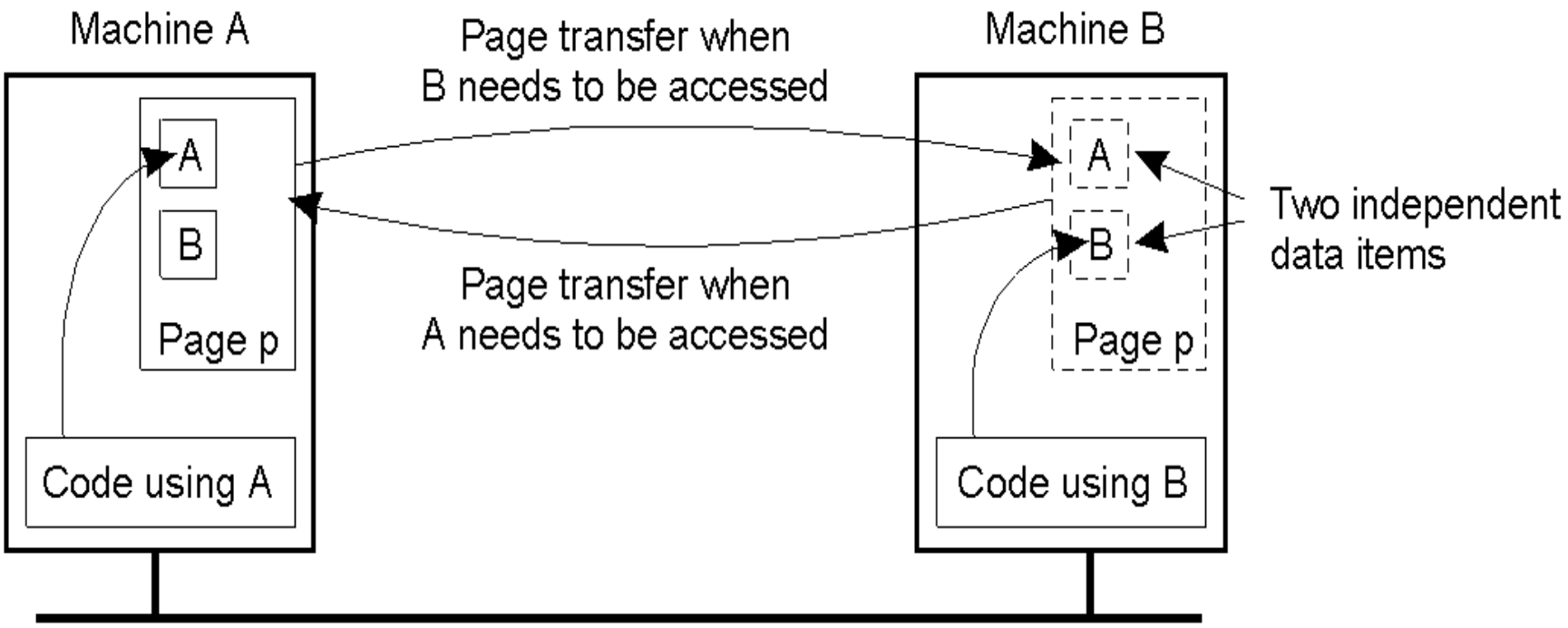
a) Pages of address space distributed among four machines

b) Situation after CPU 1 references page 10

c) Situation if page 10 is read only and replication is used

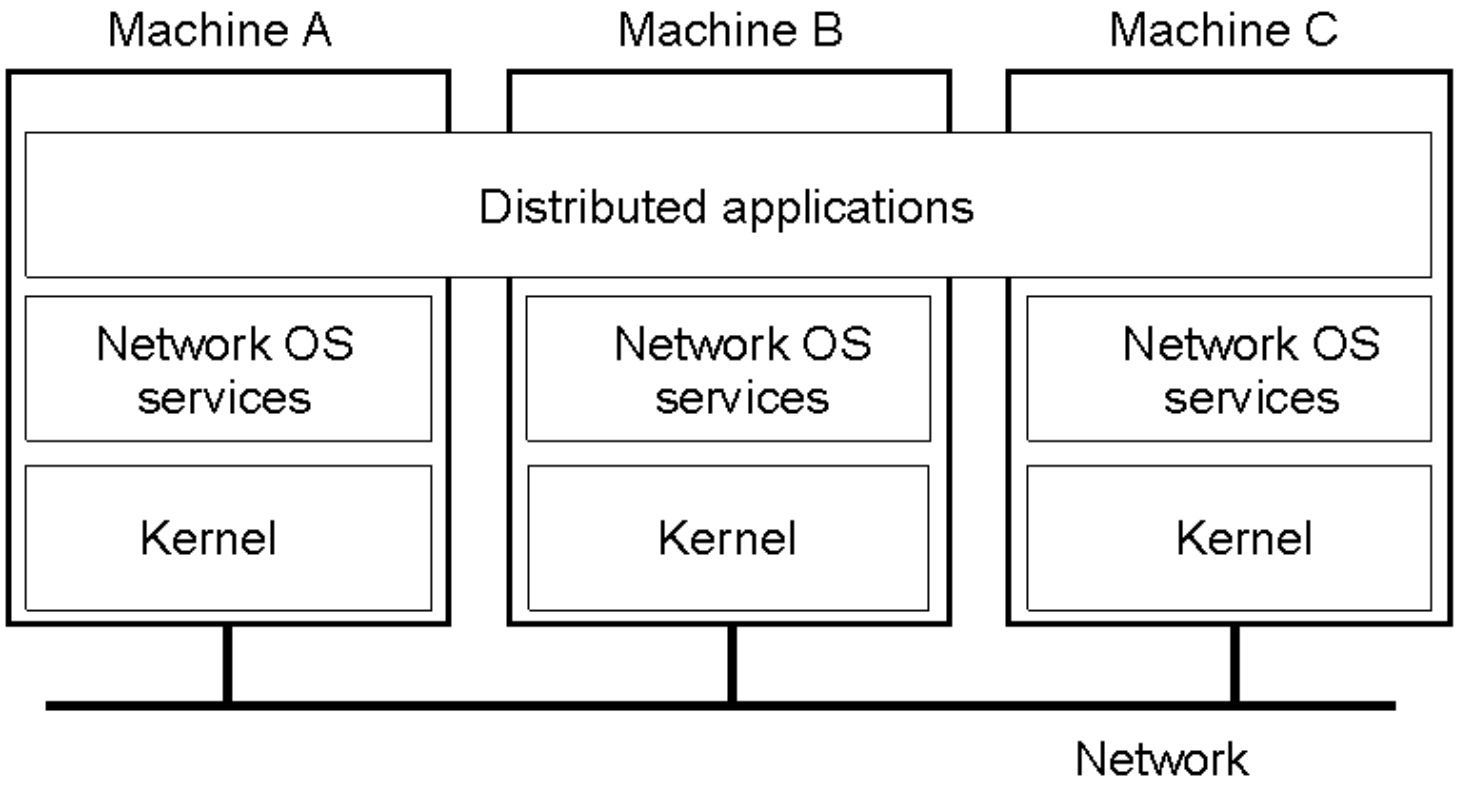


Distributed Shared Memory Systems (2)



False sharing of a page between two independent processes.

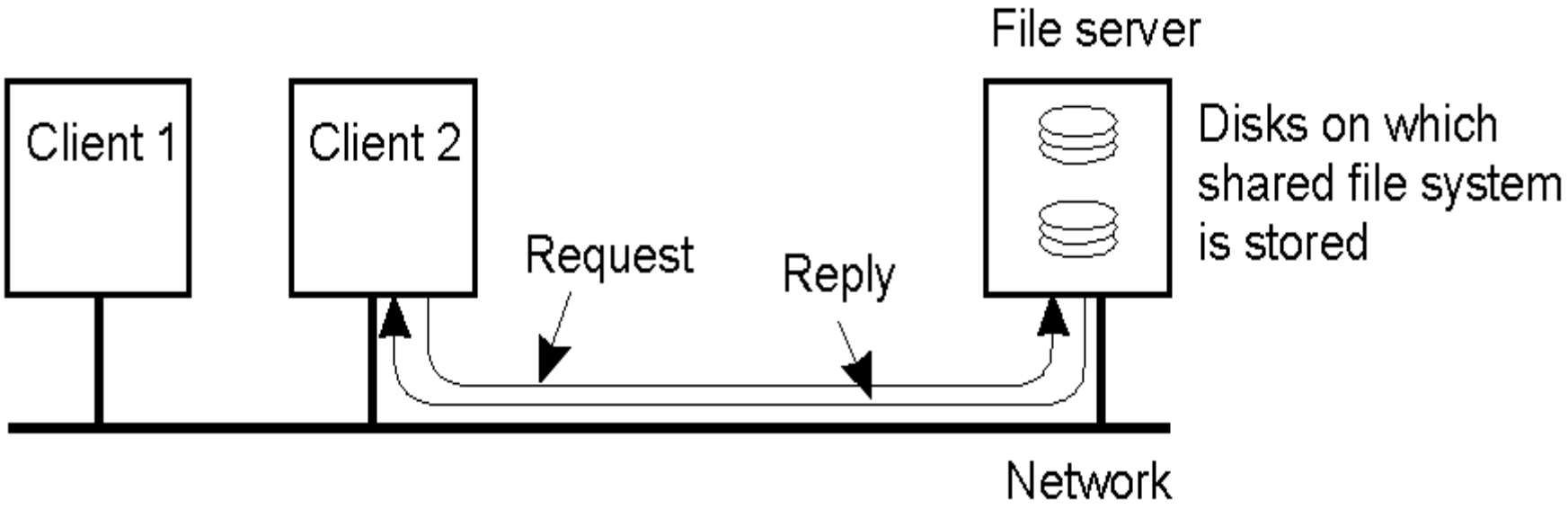
Network Operating System (1)



General structure of a network operating system.

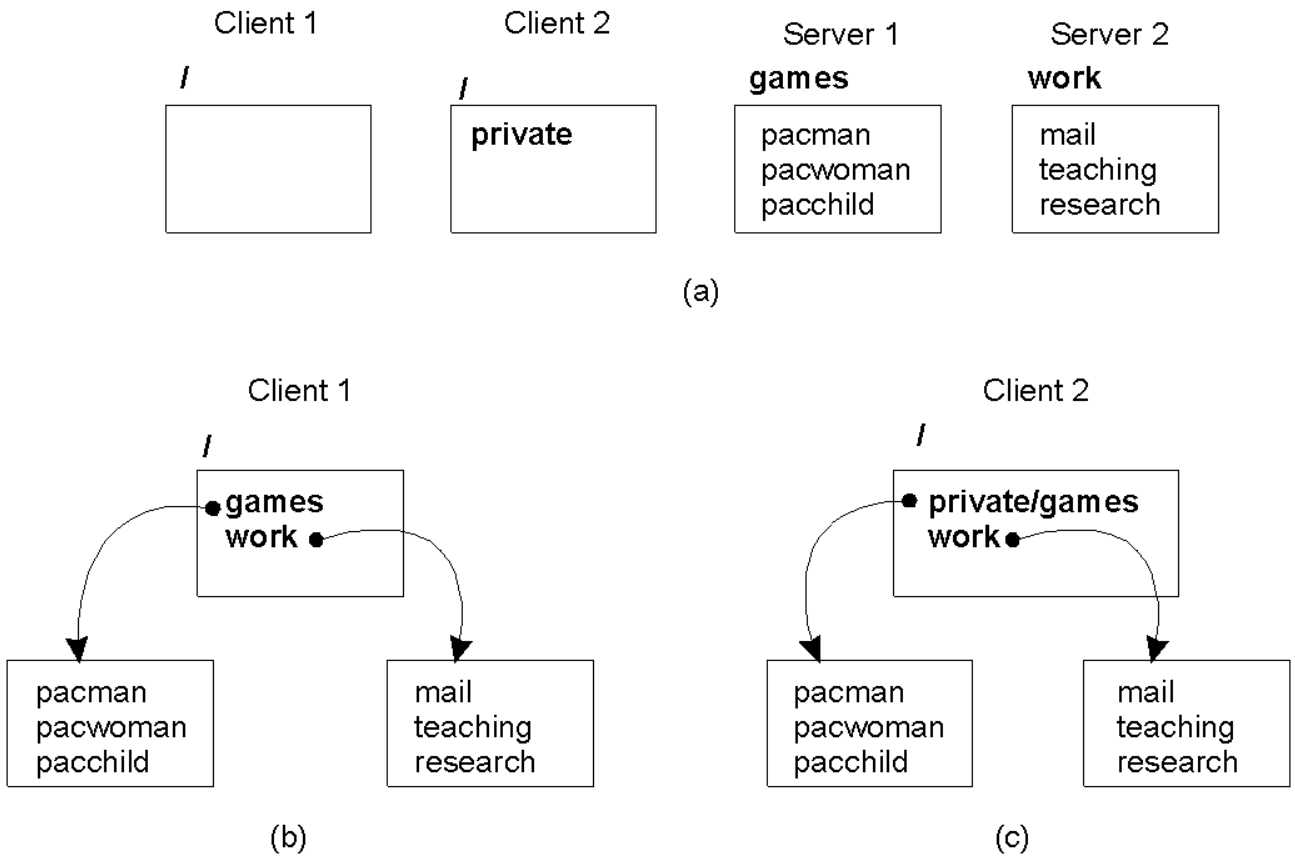
Command examples: rlogin, rcp, rsh

Network Operating System (2)



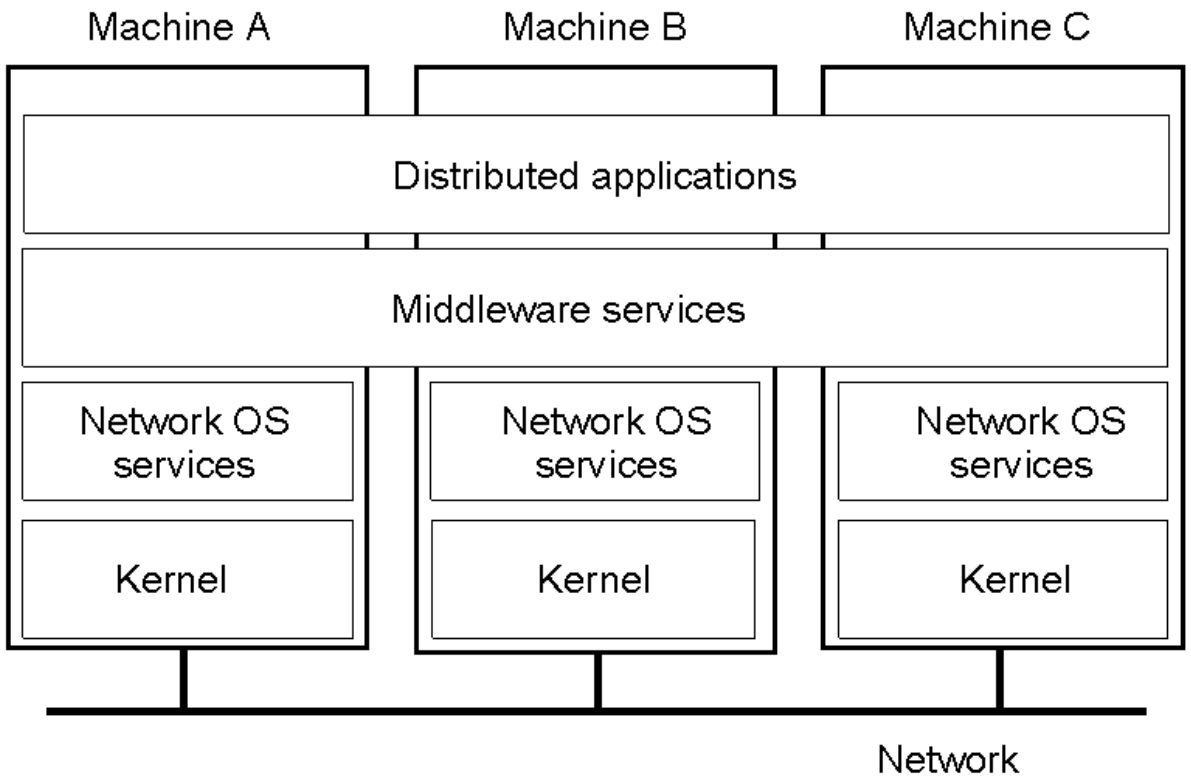
Two clients and a server in a network operating system.

Network Operating System (3)



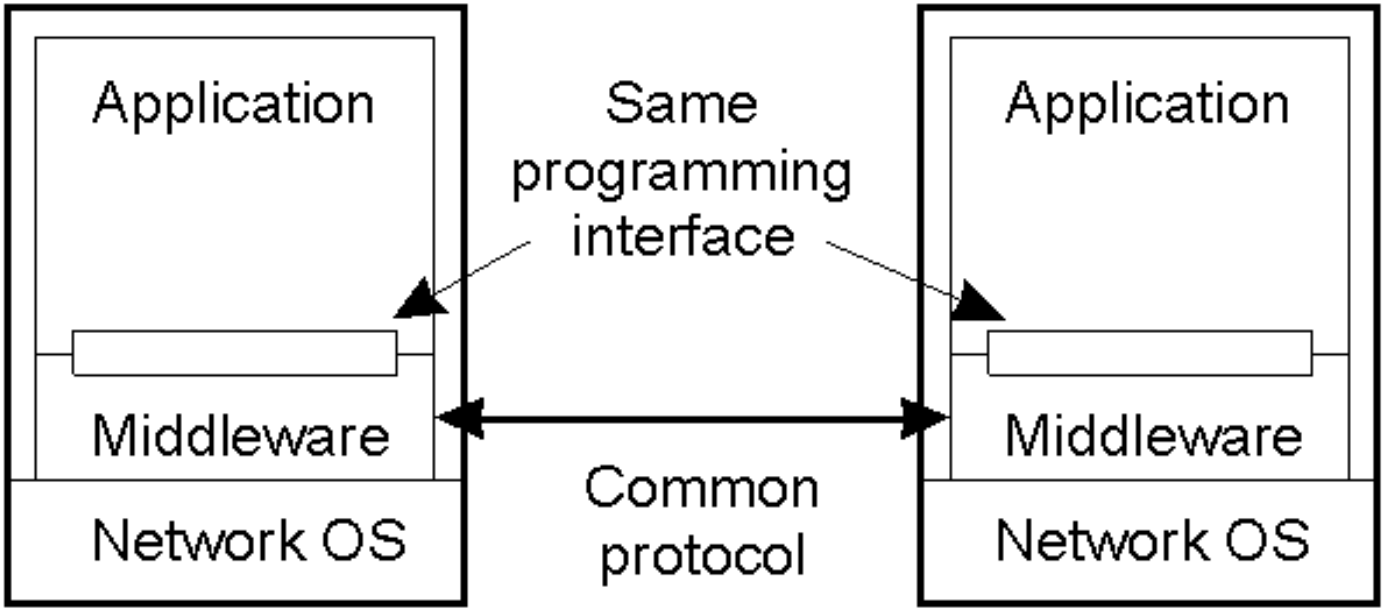
Different clients may mount the servers in different places.

Positioning Middleware



General structure of a distributed system as middleware.

Middleware and Openness



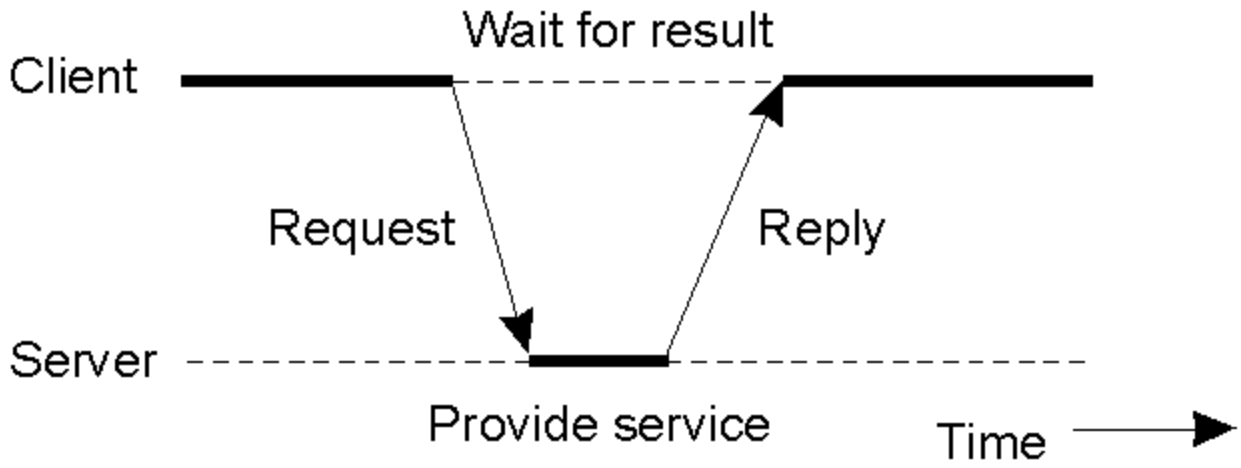
In an open middleware-based distributed system, the protocols used by each middleware layer should be the same, as well as the interfaces they offer to applications.

Comparison between Systems

Item	Distributed OS		Network OS	Middleware-based OS
	Multiproc.	Multicomp.		
Degree of transparency	Very High	High	Low	High
Same OS on all nodes	Yes	Yes	No	No
Number of copies of OS	1	N	N	N
Basis for communication	Shared memory	Messages	Files	Model specific
Resource management	Global, central	Global, distributed	Per node	Per node
Scalability	No	Moderately	Yes	Varies
Openness	Closed	Closed	Open	Open

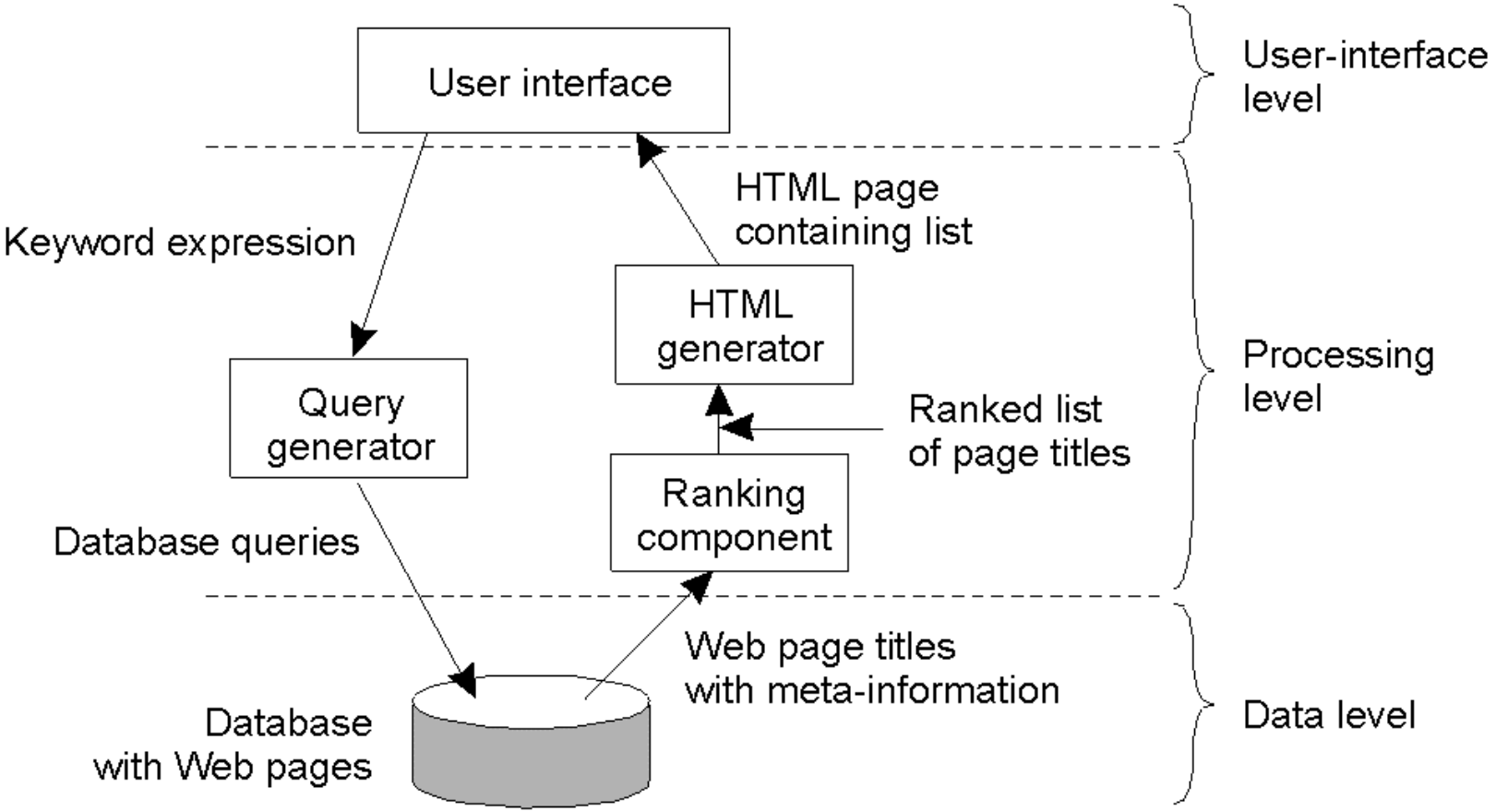
A comparison between multiprocessor operating systems, multicomputer operating systems, network operating systems, and middleware based distributed systems.

Clients and Servers



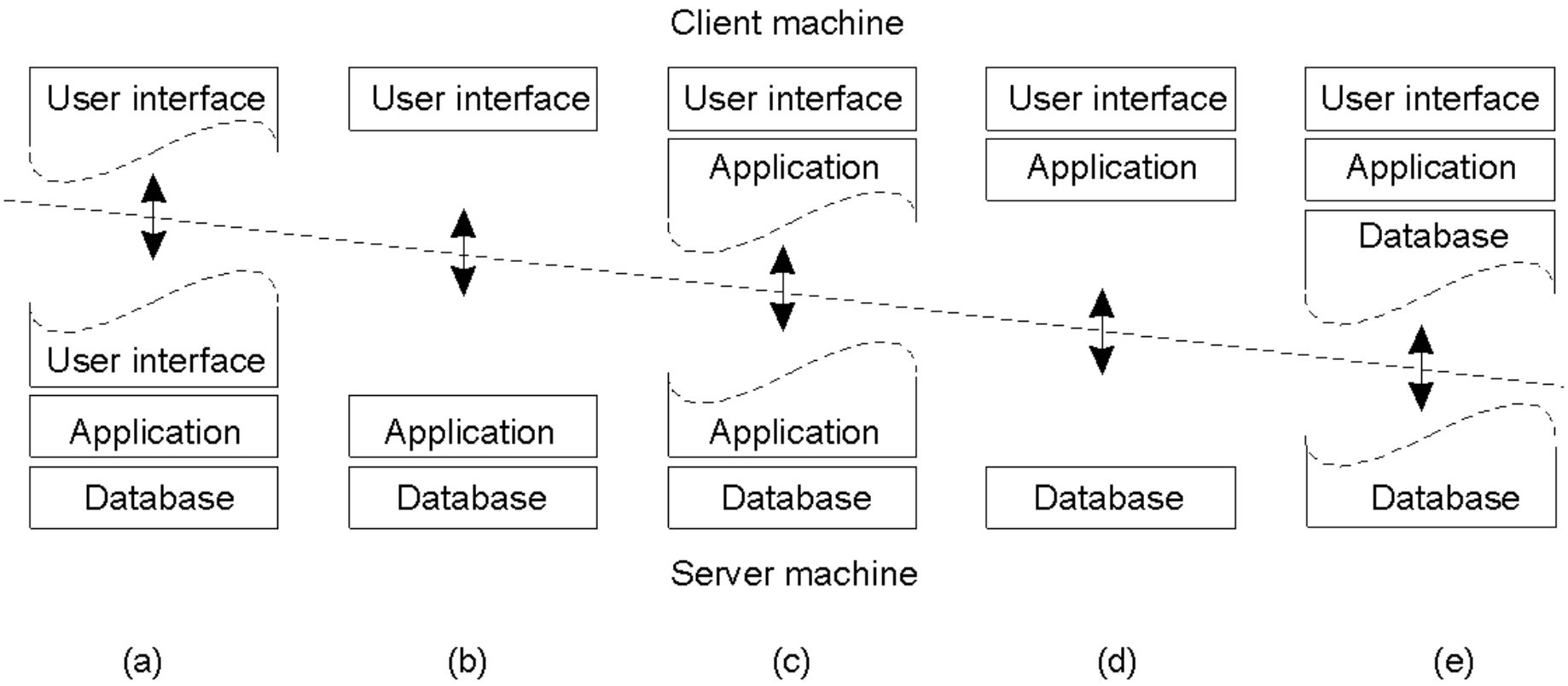
General interaction between a client and a server.

Processing Level



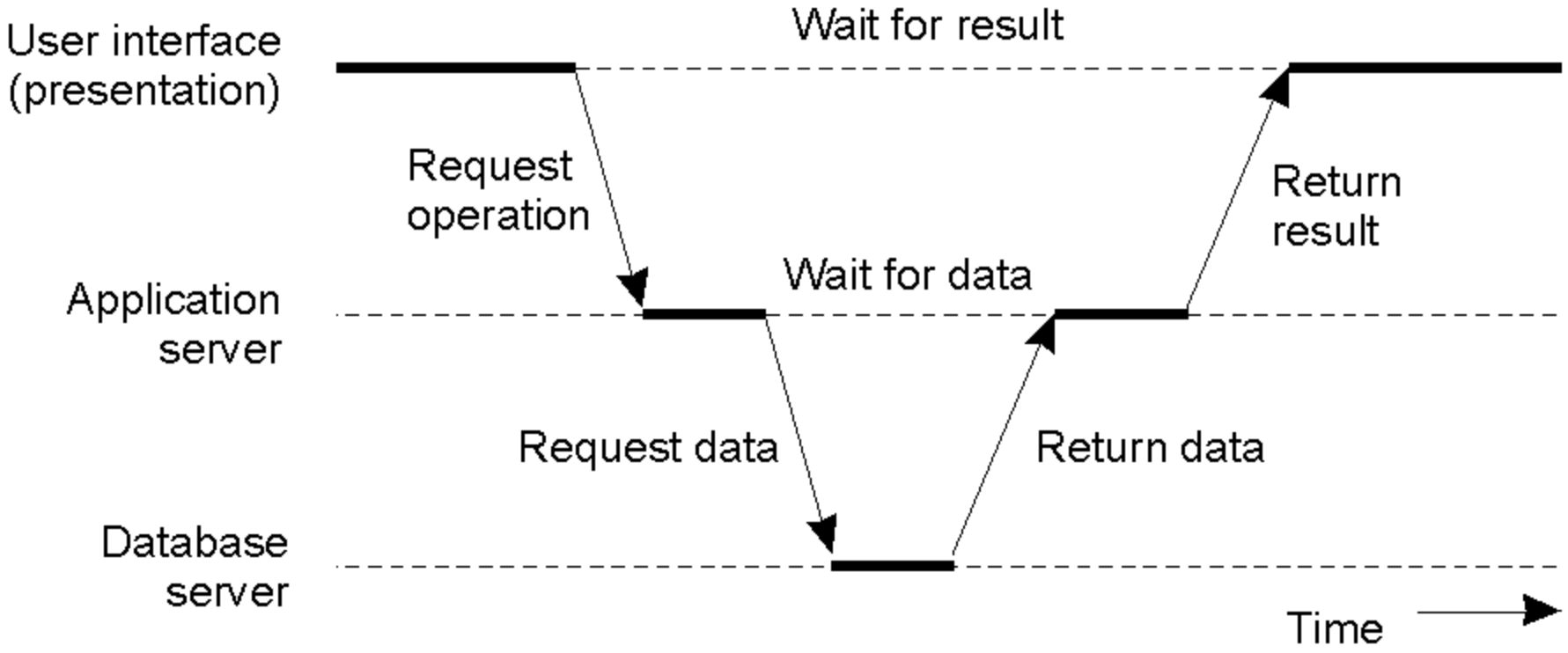
The general organization of an Internet search engine into three different layers

Multitiered Architectures (1)



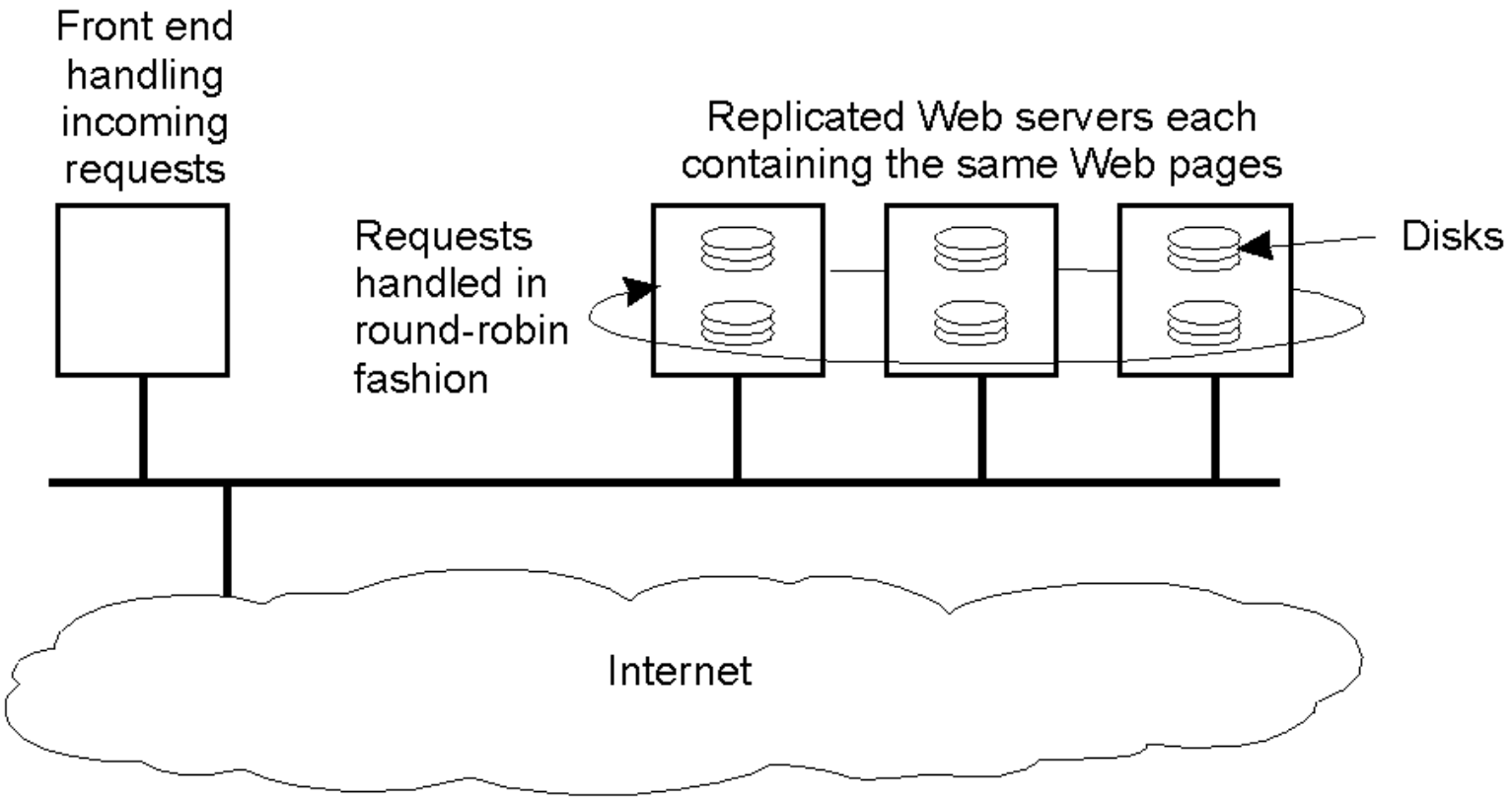
Alternative client-server organizations (a) – (e).

Multitiered Architectures (2)



An example of a server acting as a client.

Modern Architectures



An example of horizontal distribution of a Web service.