Systèmes Distribués

Master MIAGE 1

Andrea G. B. Tettamanzi
Université de Nice Sophia Antipolis
Département Informatique
andrea.tettamanzi@unice.fr

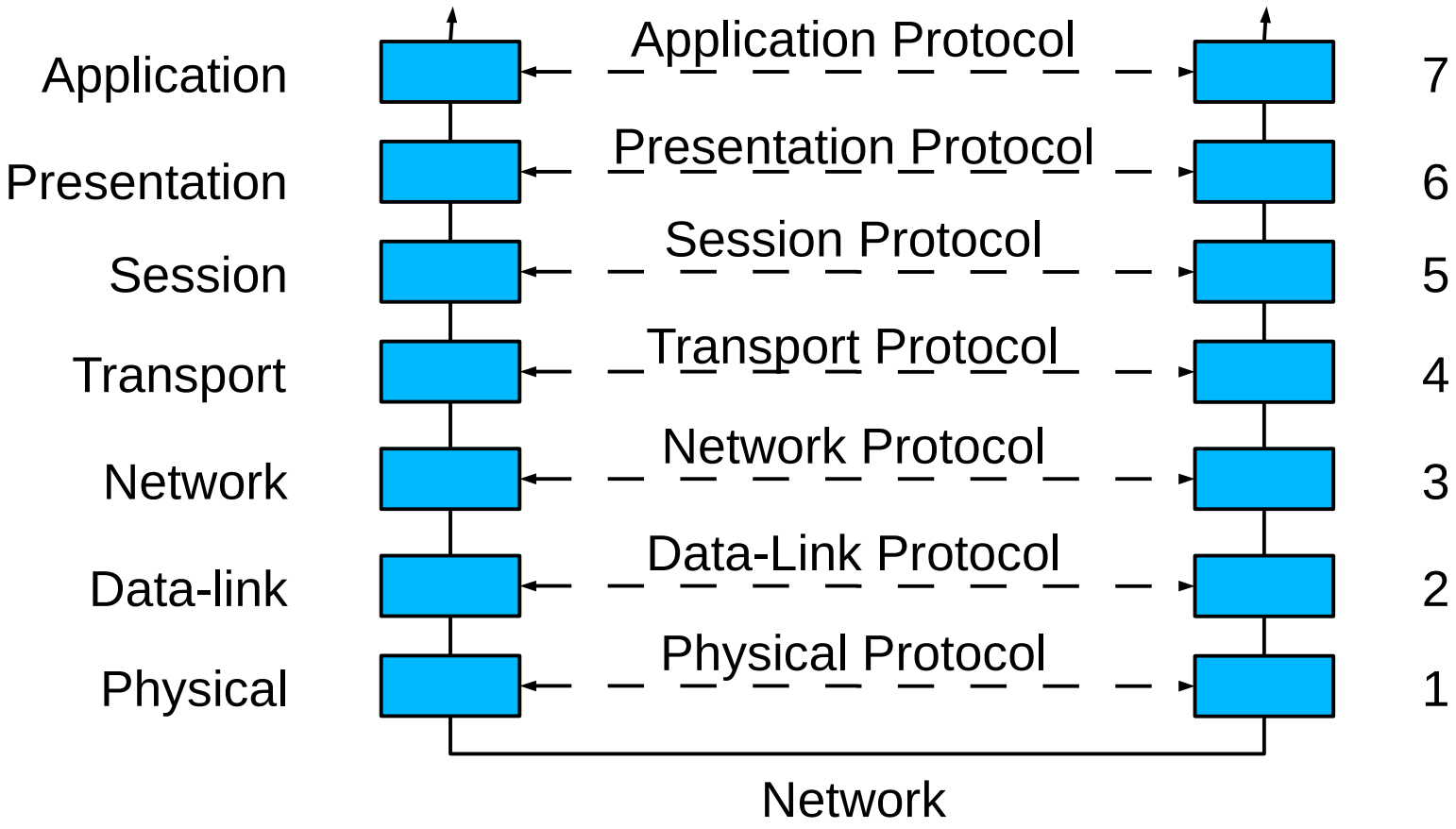
CM - Séance 3 – Partie I

Communication
Concepts de base et RPC

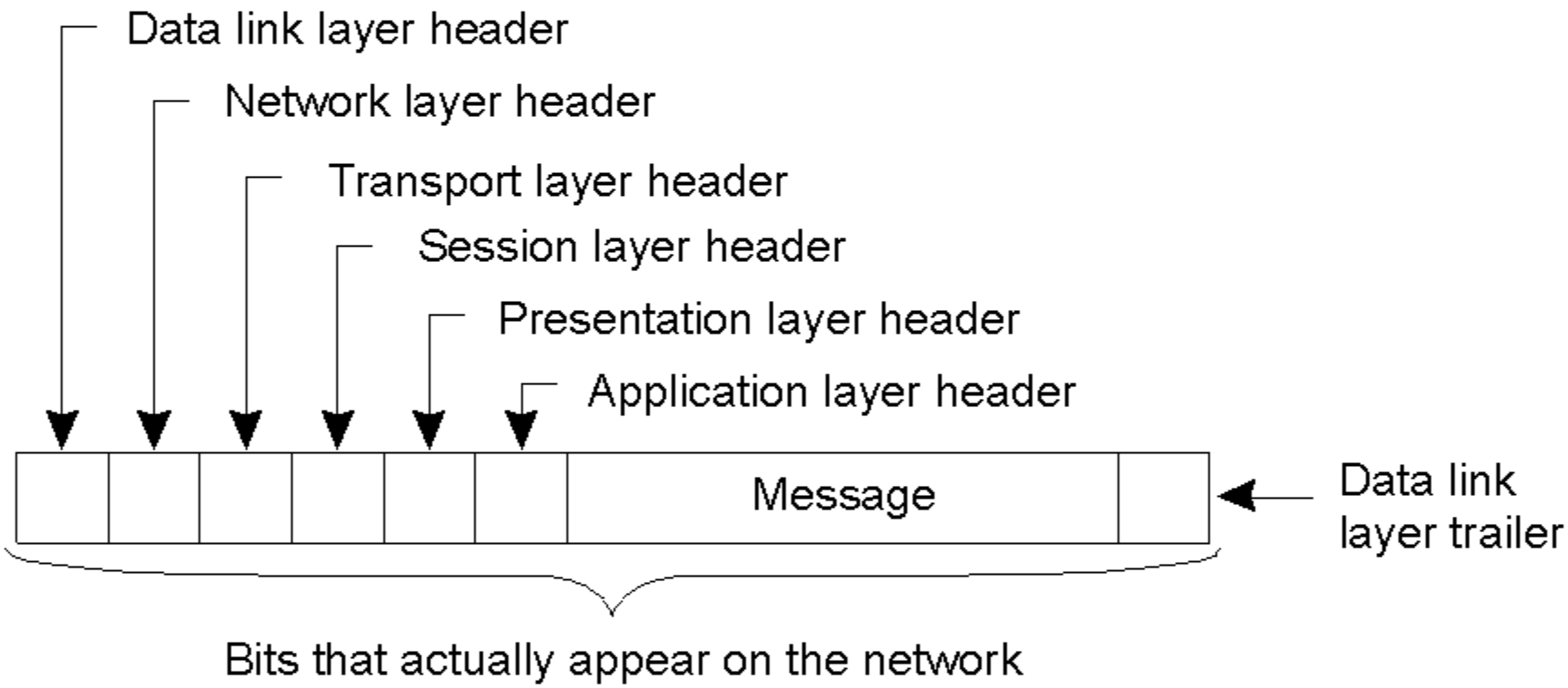
Communication

Chapter 4

Layered Protocols: the ISO/OSI Stack

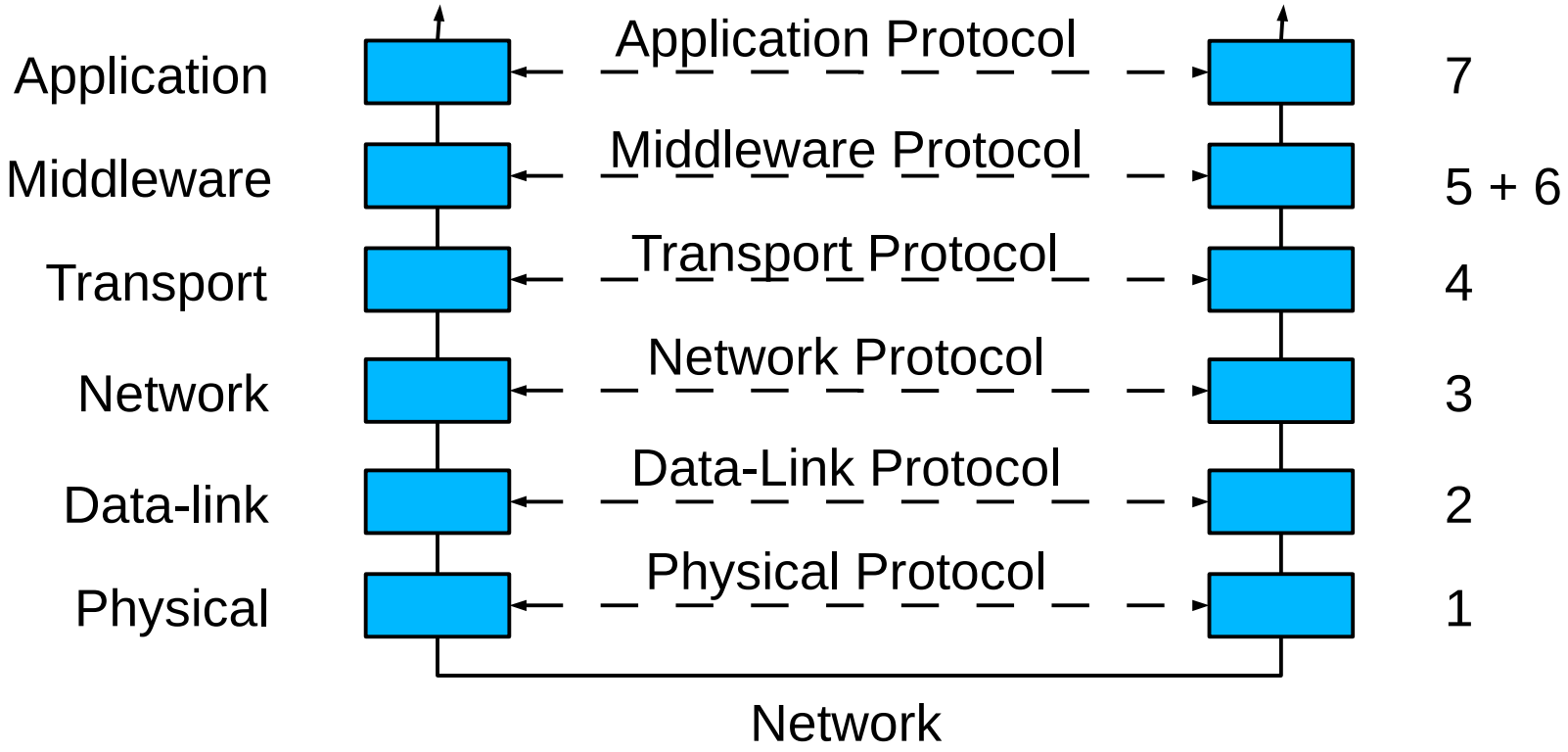


Layered Protocols: Messages

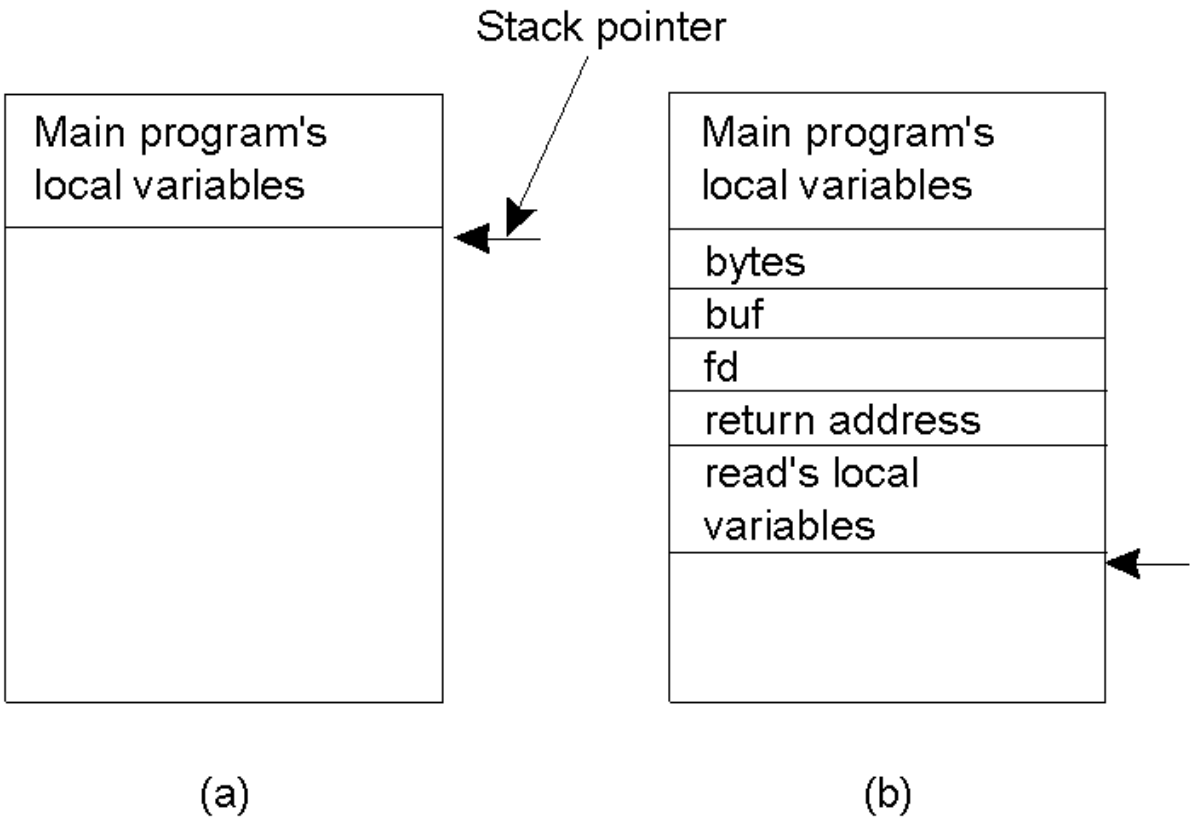


A typical message as it appears on the network.

Middleware Protocols: An adaptation of the ISO/OSI Stack



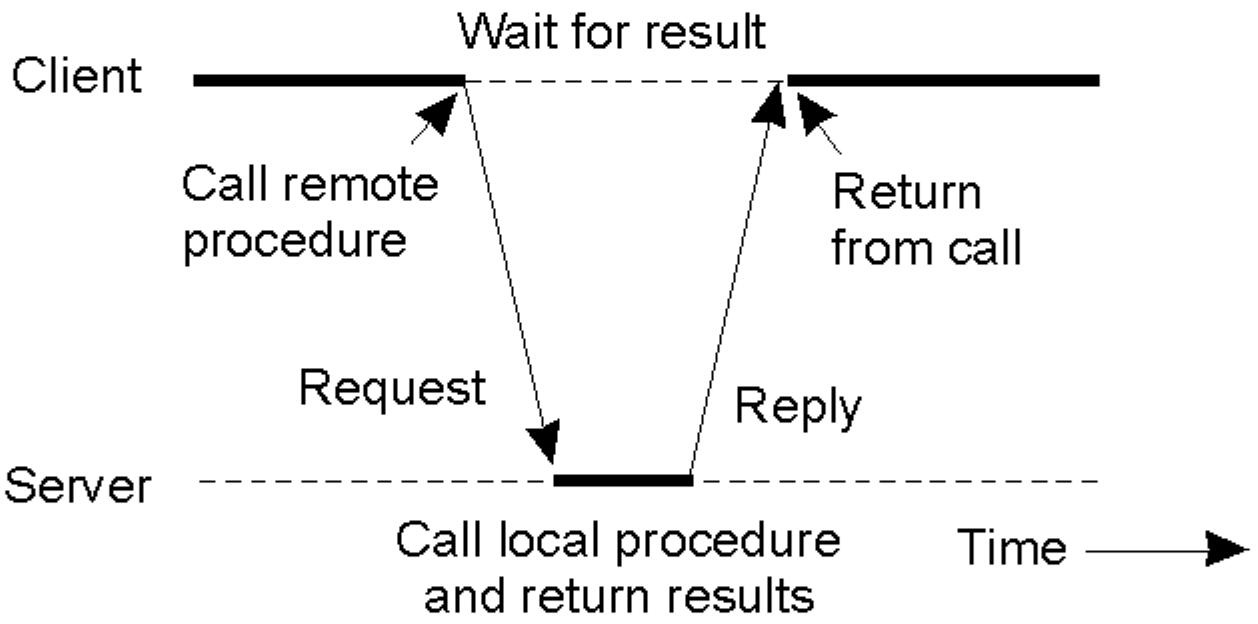
Conventional Procedure Call



a) Parameter passing in a local procedure call: the stack before the call

b) The stack while the called procedure is active

Client and Server Stubs

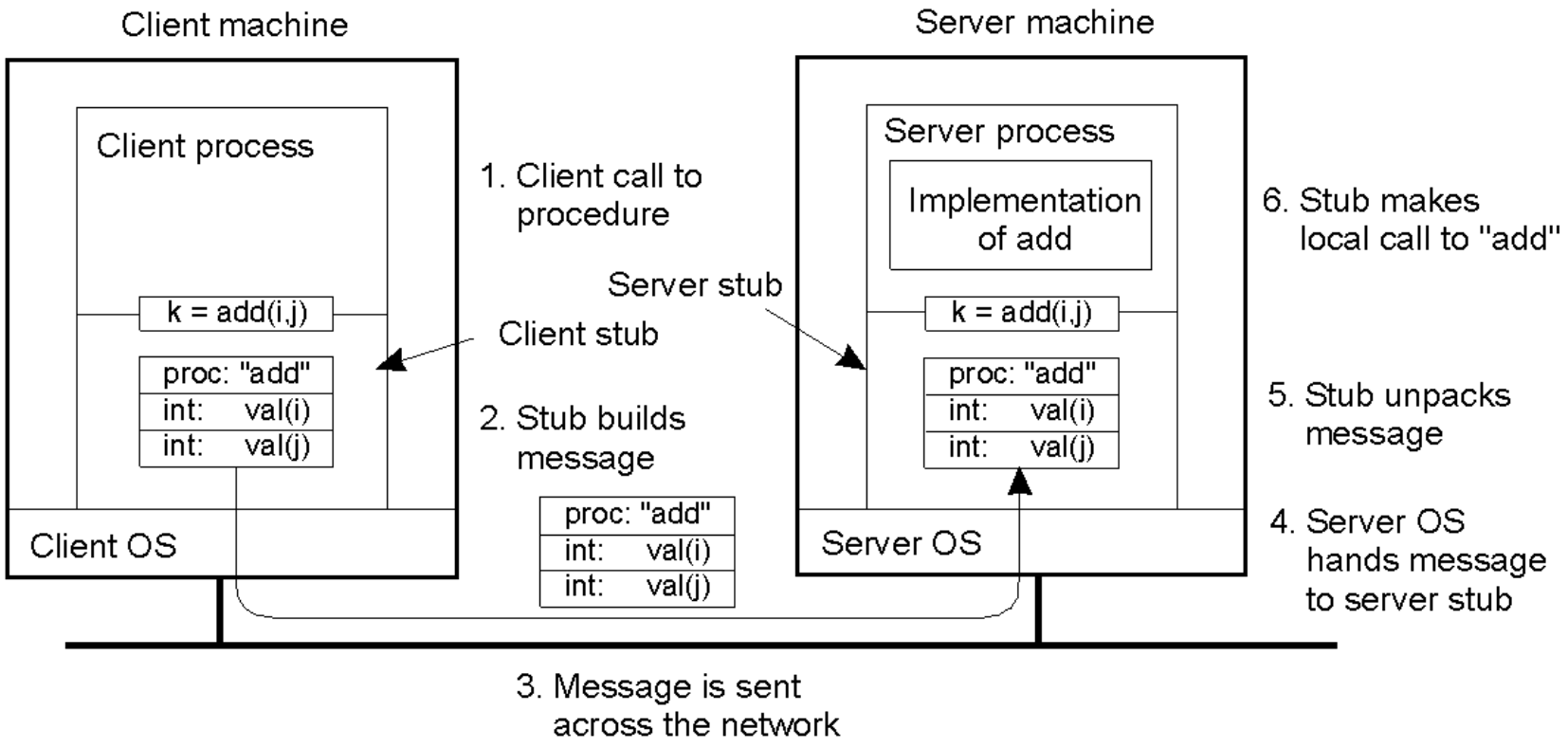


Principle of RPC between a client and server program.

Steps of a Remote Procedure Call

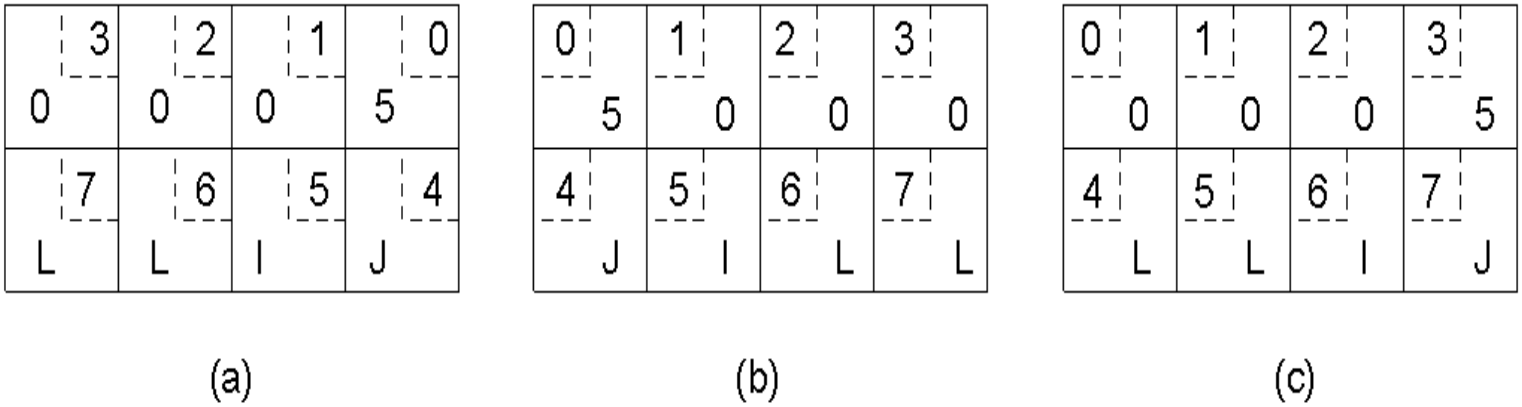
1. Client procedure calls client stub in normal way
2. Client stub builds message, calls local OS
3. Client's OS sends message to remote OS
4. Remote OS gives message to server stub
5. Server stub unpacks parameters, calls server
6. Server does work, returns result to the stub
7. Server stub packs it in message, calls local OS
8. Server's OS sends message to client's OS
9. Client's OS gives message to client stub
10. Stub unpacks result, returns to client

Passing Value Parameters (1)



Steps involved in doing remote computation through RPC

Passing Value Parameters (2)



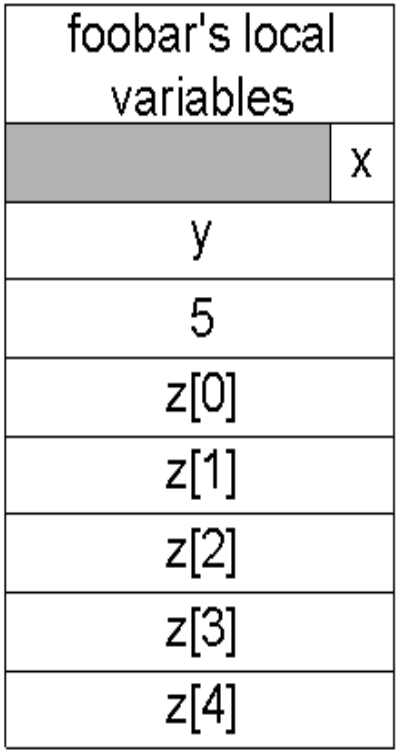
a) Original message on the Pentium
b) The message after receipt on the SPARC
c) The message after being inverted. The little numbers in boxes indicate the address of each byte

Parameter Specification and Stub Generation

a) A procedure
b) The corresponding message.

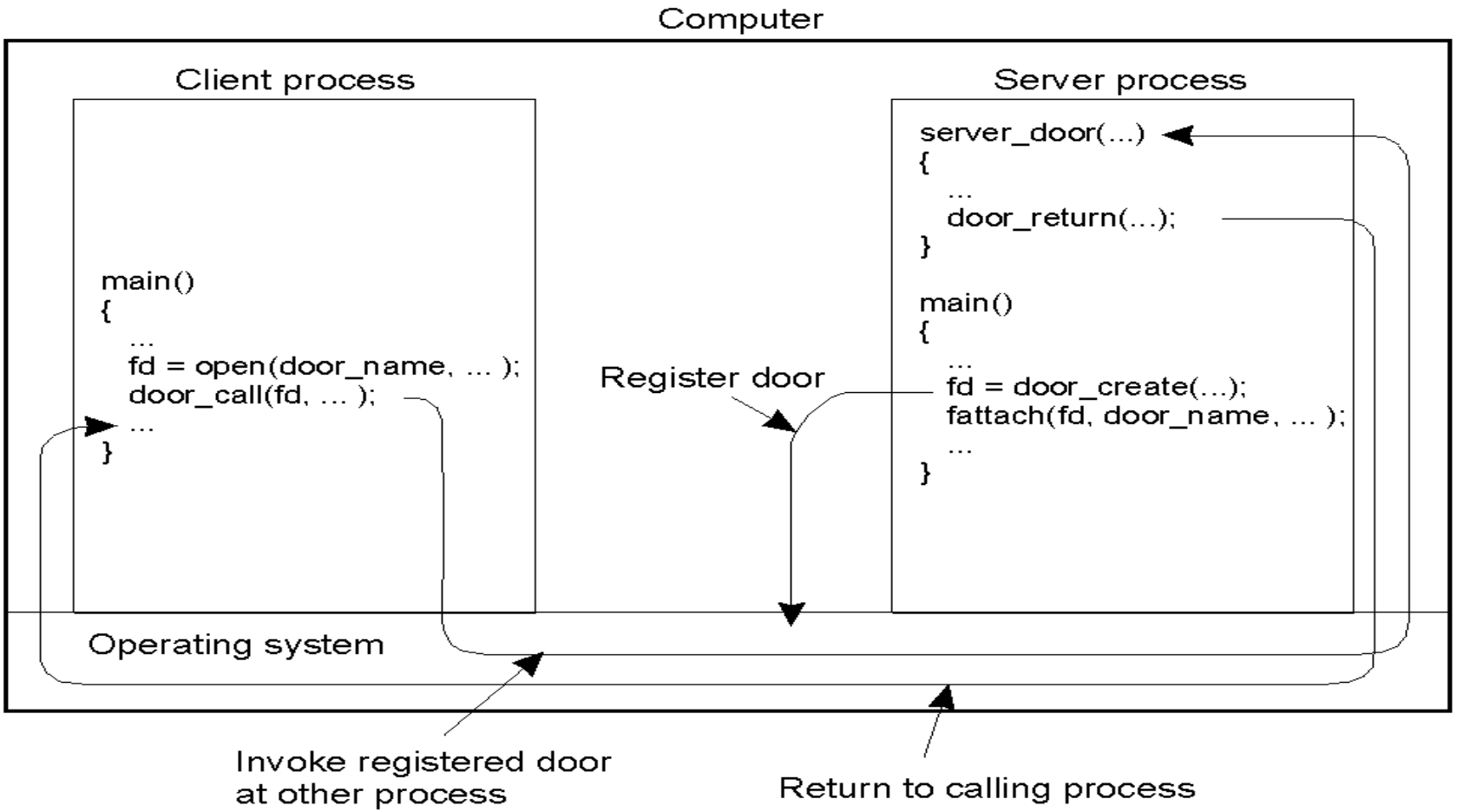
```
foobar( char x; float y; int z[5] )
{
   ....
}
```

(a)



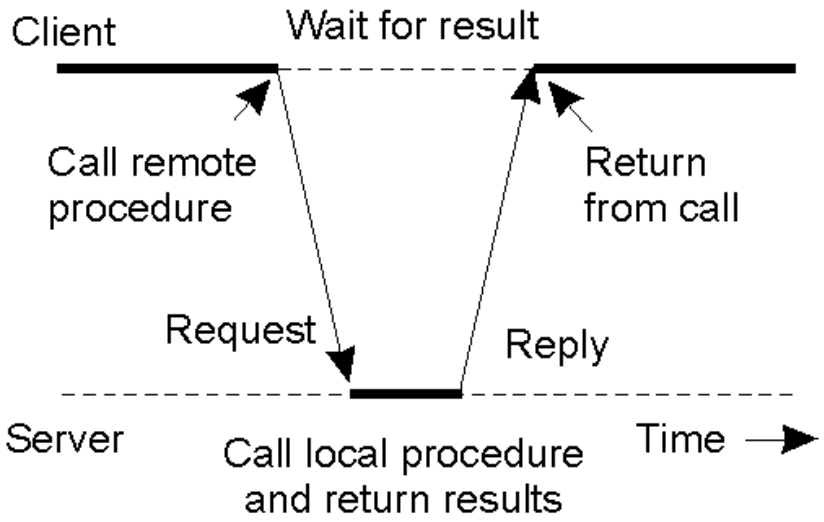
(b)

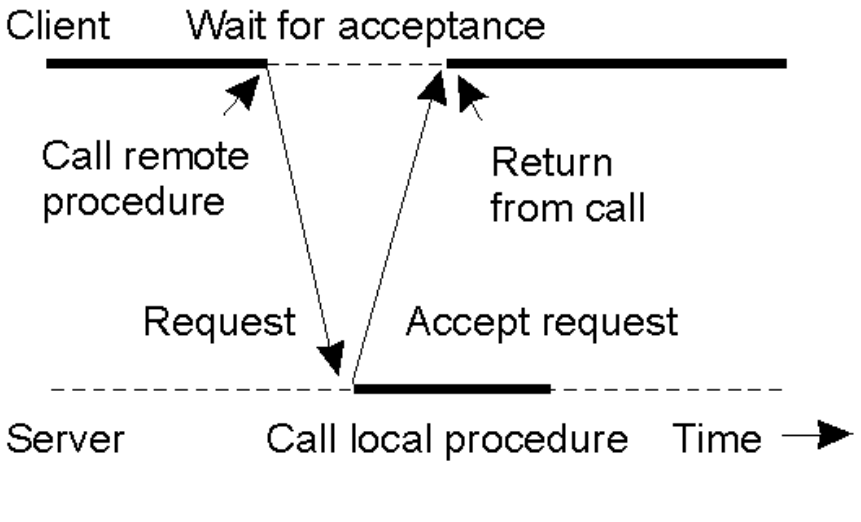
Doors



The principle of using doors as IPC mechanism.

Asynchronous RPC (1)

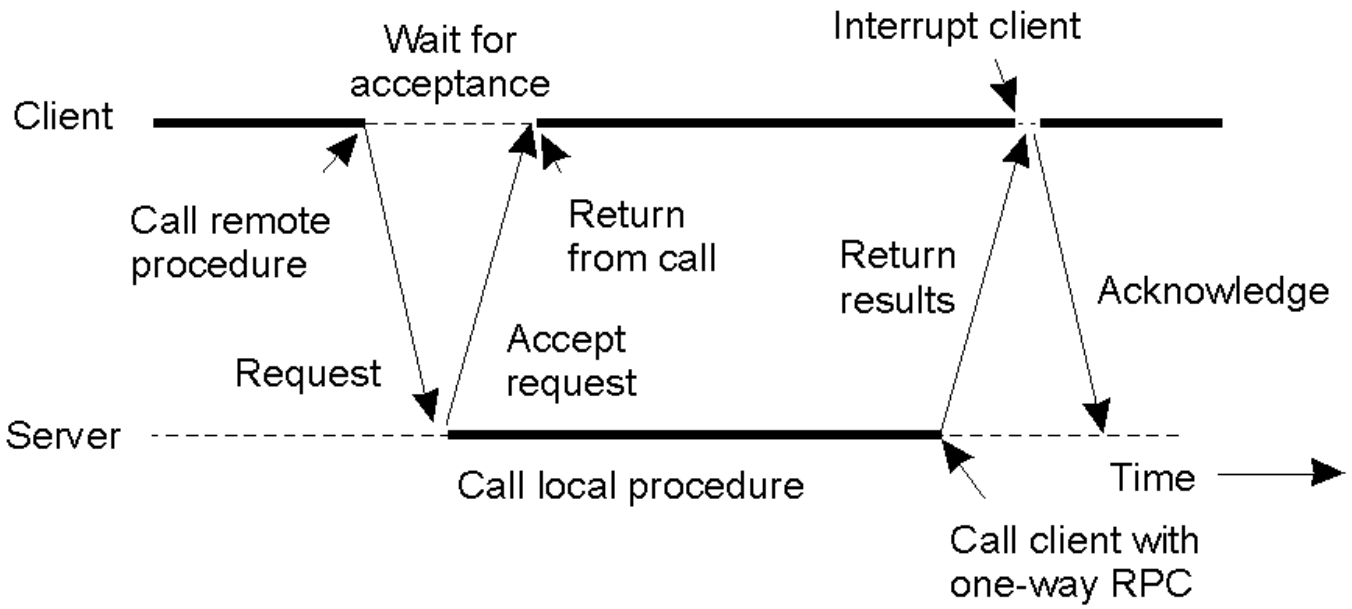




a) The interconnection between client and server in a traditional RPC

b) The interaction using asynchronous RPC

Asynchronous RPC (2)

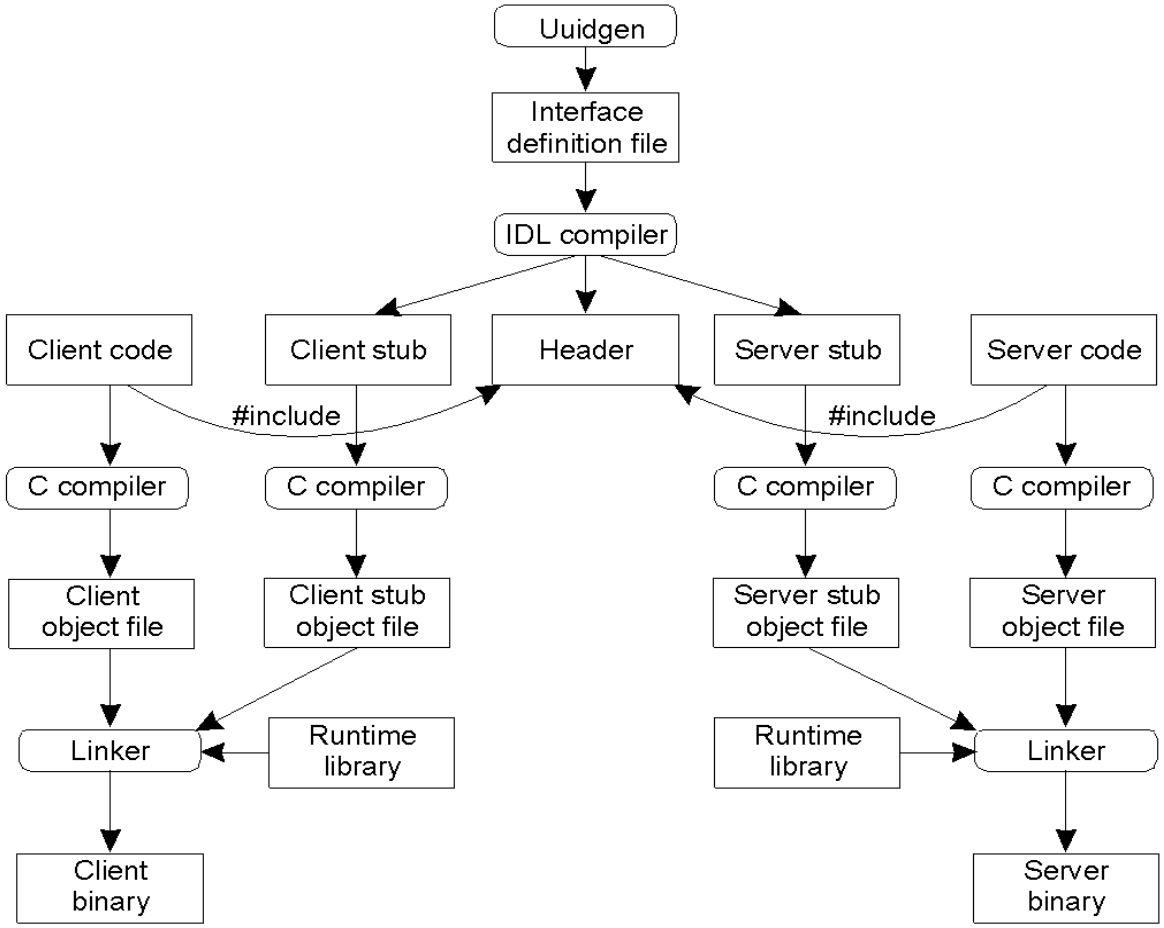


A client and server interacting through two asynchronous RPCs

DCE

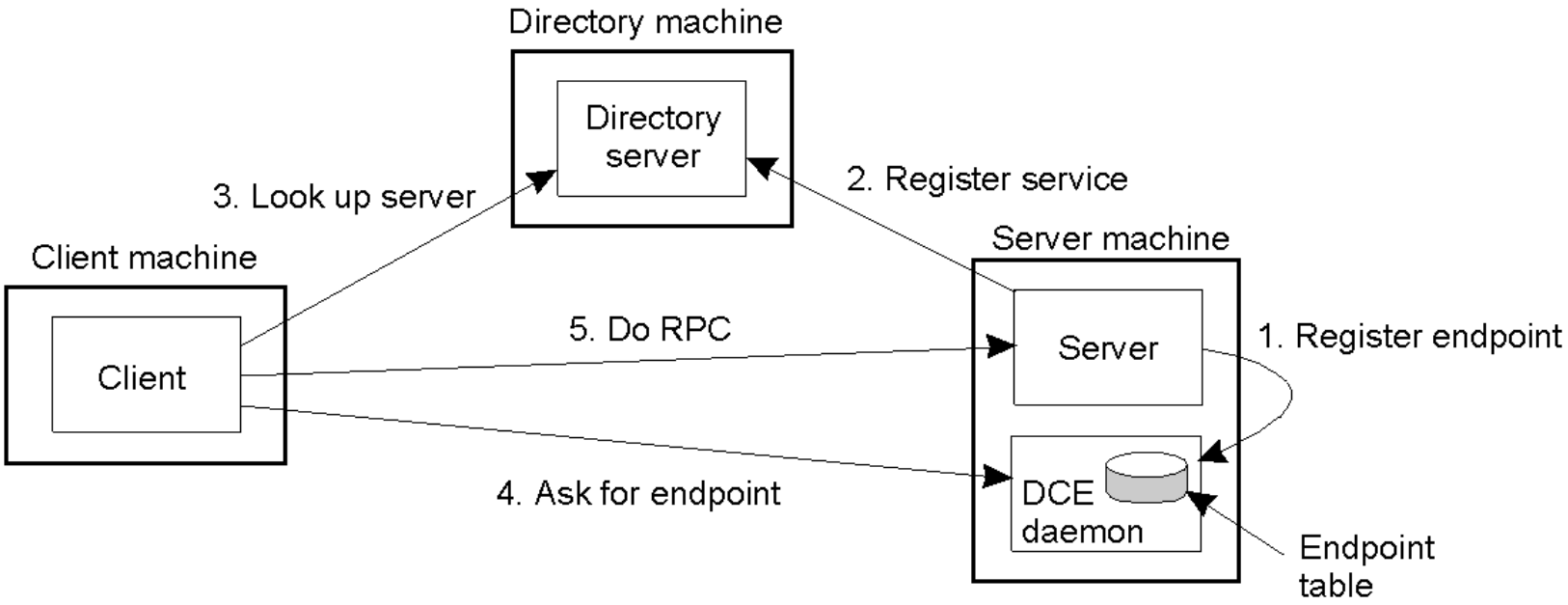
- DCE = Distributed Computing Environment
- Developed in the early '90 by a consortium of Apollo Computer (later acquired by HP), IBM, DEC, and others
- DCE supplies a framework for client/server applications
- The framework includes :
  - DCE/RPC, a remote procedure call mechanism
  - A naming service
  - A time service
  - An authentication service
  - DCE/DFS, a distributed file system
- Now OpenDCE: http://www.opengroup.org/dce/

Writing a Client and a Server



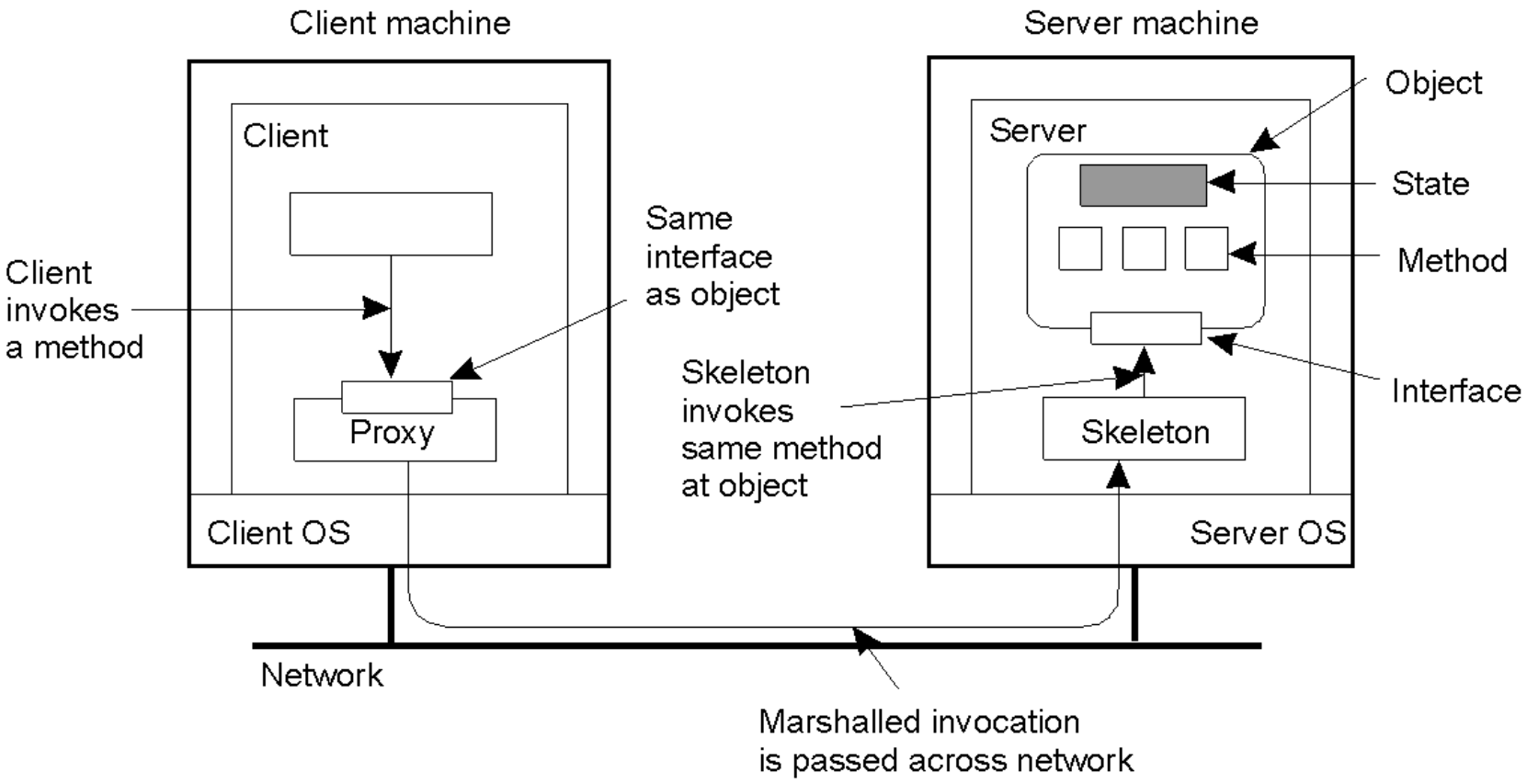
The steps in writing a client and a server in DCE RPC.

Binding a Client to a Server



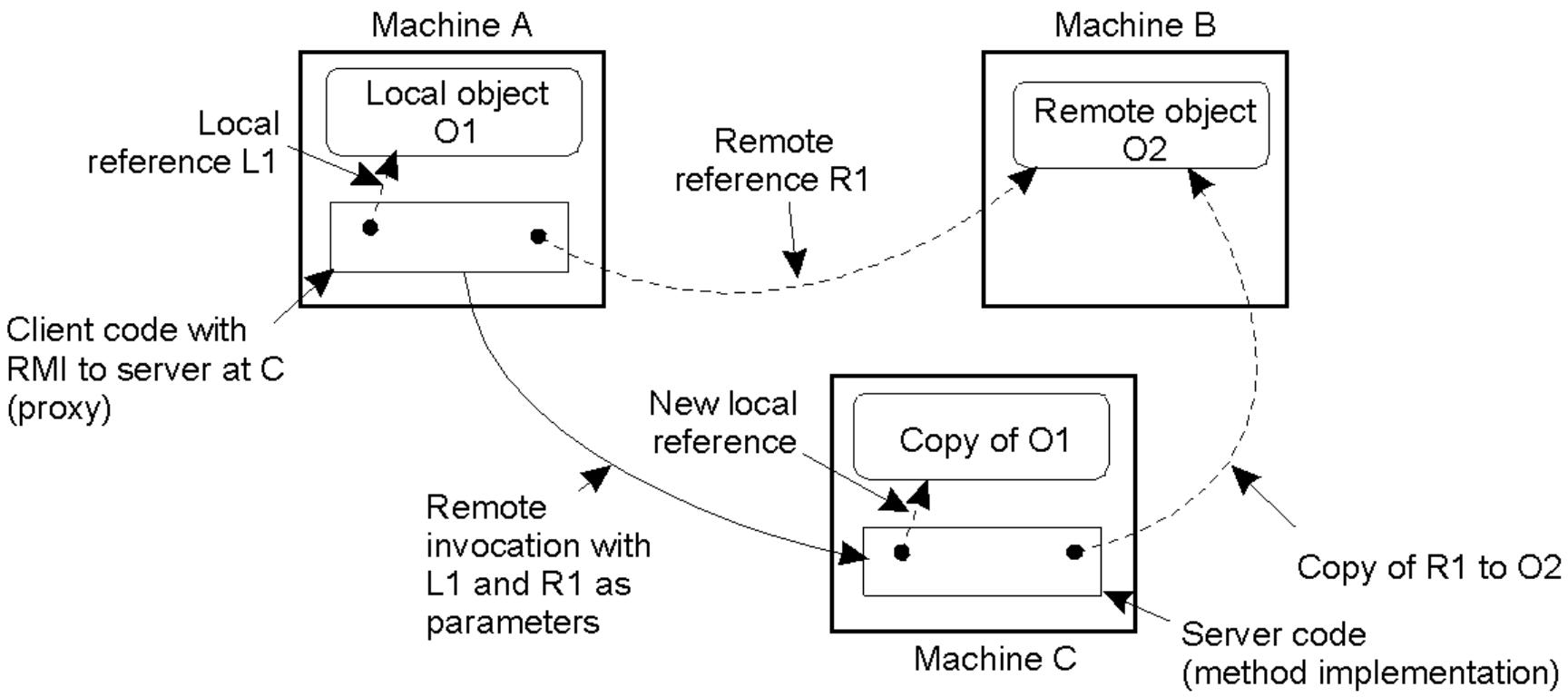
Client-to-server binding in DCE.

Distributed Objects



Common organization of a remote object with client-side proxy.

Parameter Passing



The situation when passing an object by reference or by value.

Merci de votre attention