



# LMT

(An online tutor who teaches you as a friend)

[www.lastmomenttutions.com](http://www.lastmomenttutions.com)

## *Notes*

# *Last moment tuitions*

ek online friend hai jo apko dost ki tarah padhata hai  
jaise exam time dost padhate hai  
aur exam me kaise PASS hona hai uska jugaad  
apko yeh jaroor milenga

**Note :** pehle website or youtube pe videos dekhlo  
fir apko notes bahut ache se samjhenge

[www.Lastmomenttuitions.com](http://www.Lastmomenttuitions.com)

# Distributed System

- What is Distributed System (Basics)
- Goals of Distributed System
- Issue in Distributed System
- Software concept of distributed system
  - DOS (Distributed OS)
  - NOS (Network OS)
  - Middleware
- Models of Middleware
  - RPC (Remote Procedure call)
  - MOM (Message Oriented Middleware)
  - DO (Distributed Object)
  - Peer to Peer
  - Vertical and Horizontal

# Distributed System:-

Amrinder

1) What is Distributed System?

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

Two aspects of definition: → First one deals with Hardware  
↳ the machines are autonomous.

→ Second one deals with software  
↳ users think they are dealing with a single system.

Simple si baat hote hai 4 aadami par dikhta Ek hi hai aise hote hai bohat si systems par user ko lagta hai Ek hi hai.

→ To support heterogeneous computers and networks while offering a single-system view, distributed systems are often organized by means of a layer of software that is logically placed between a higher level layer consisting of users and applications.

→ A layer underneath consisting of operating systems, as shown in below Fig. (a) known as middleware.



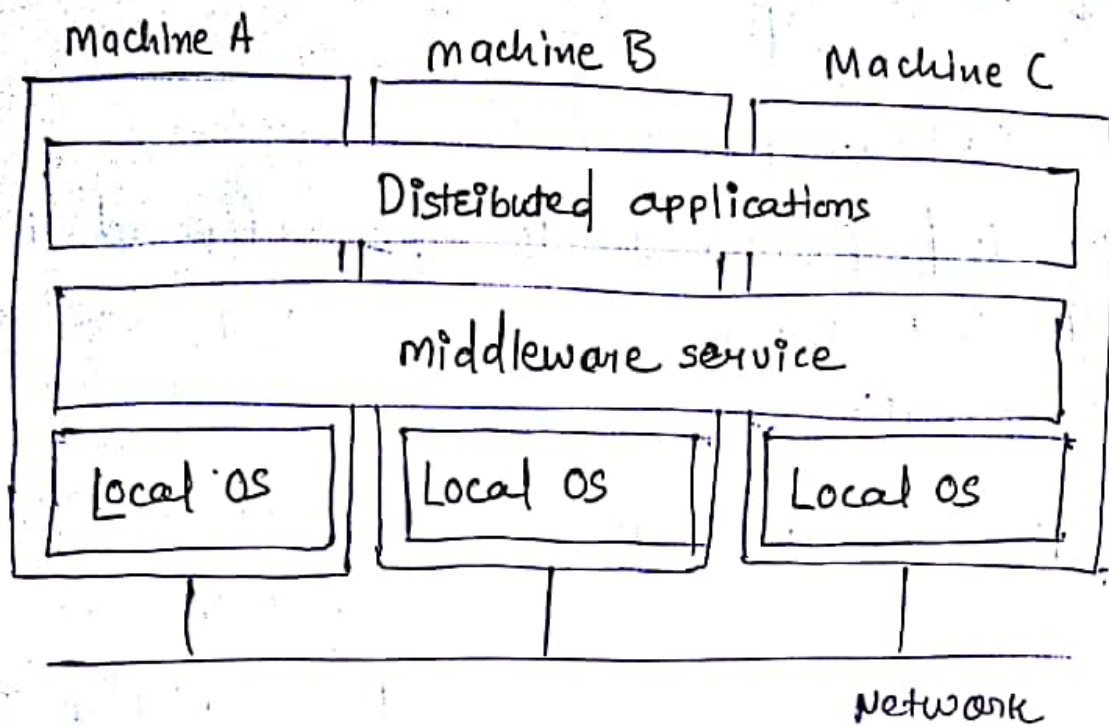


Fig (a)

### Examples of Distributed systems:—

- ↳ 1) Network of workstations in a university or company department.
- ↳ 2) Workflow information system that supports the automatic processing of orders.

Sabse agra example.. WWW (World Wide Web)  
 WWW pe kitne sare servers connected hai kisko  
 pata par user ko toh ek hi jagah www type  
 karne se sab mil jata hai.

Q) What are goals of distributed system.

- The four important goals that should be met to make building a distributed system worth the effort
- A distributed system should make resources easily accessible, it should reasonably hide the fact that resources are distributed across a network, it should be open and it should be scalable

1. Making Resources Accessible : The main goal of distributed system is

to make it easy for the user (and application) to access remote resources and to share them in a controlled and efficient way.

Example: Printer used by a network of computers

- Resources can be just about anything, but typical examples include things like printer, computers, storage facilities, data, files, web pages, etc.

2. Distribution transparency: An important goal of a distributed system is to hide the fact that its Processes and resources are physically distributed across multiple computers

- A distributed system that is able to present itself to users and applications as if it were only a single computer system is said to be transparent

3. Openness: The openness of distributed system is determined primarily by the degree to which new resources sharing services can be added and be made available for use by variety of client Program.

- An open distributed system is a system that offers services according to standard rules that describe the syntax and semantic of those services
- An open distributed system should also be extensible



4) Scalability: Distributed system operate effectively and efficiently at many different set scales ranging from a small intranet to the internet

- A system is described as scalable if it will remain effective when there is a significant increase in number of resources and number of users



Q] What are the issues in designing a distributed system?

⇒ 1] Openness: The Openness of a computer system is the characteristic that determine whether the system can be extended or reimplemented in various ways.

- The openness of distributed system is determined primarily by the degree to which new resource-sharing services can be added and be made available for use by variety of client Programs

2] Security: Many of the information resources that are made available and maintained in distributed system have high intrinsic value to the users

- Their security is therefore of considerable importance
- Security for information & resources has three component: Confidentiality, integrity, availability

3) Scalability: Distributed system operate effectively and efficiently at many different scales ranging from a small intranet to the internet

- A System is described as scalable if it will remain effective when there is a significant increase in the number of resources and number of user.

4) Flexibility: flexibility is considered as the ability which can be easily modified

- It includes:

- Ease of modification: It should be easy to incorporate changes in the system

- Ease of enhancement: It should be easy to new functionality into the system

5) Reliability: One of the original goal of distributed system was to make them more reliable than single Processor system

- The idea is that if a machine goes down some other machine takes over the job
- For high reliability, the fault handling mechanism of DS must be design Properly to avoid fault, to tolerate fault, to detect and to recover from faults

5) Heterogeneity: A heterogeneous system consists of interconnected sets of dissimilar hardware or software of both.

- The system should be designed in such a way that it should cope with the heterogeneous environment.

7) Performance: If a distributed system is used its performance must be as good as centralized system

- That is when a particular application is run on distributed system, its overall performance should be better than at least to that of the running the same application on a single processor system.
- Some design principles should be considered for better performance as follows
  - Take advantage of fine-grained parallelism for multiprocessing
  - minimize copying of data minimize network traffic

8) Transparency: Next Page,



## \* Transparency:-

→ An important goal of a distributed system is to hide the fact that its processes and resources are physically distributed across multiple computers.

→ A ~~single~~ distributed system that is able to present itself to users and applications as if it were only a single computer system is said to be transparent.

Several aspects of Transparency.

### 1) Access Transparency:-

→ It deals with hiding differences in data representation and the way that resources can be accessed by users.

→ Differences in naming conventions, as well as how files can be manipulated, should all be hidden from users and applications.

### 2) Location Transparency:-

Resource kahan pada, hai vo Gupadeta hai.  
location se usko matlab nahi hona chahiye.

→ Users can not tell where a resource is physically located in the system. Naming plays an important role in achieving location transparency. In particular, location transparency can be achieved by assigning only logical names to resources.



### 3) Relocation Transparency:-

- Distributed system in which resources can be relocated while they are being accessed without the user or application noticing anything. In such cases, the system is said to support relocation transparency.

### 4) Migration Transparency:-

- Distributed systems in which resources can be moved without affecting how that resources can be accessed are said to provide migration transparency.

### 5) Replication Transparency:-

Bahot sahi copies padli hai agar to bhi pata na chle user ko

- Replication transparency deals with hiding the fact that several copies of a resource exist. To hide replication from users, it is necessary that all copies have the same name.

### 6) Concurrency Transparency:-

- Two independent users may each have stored their files on the same file server or may be accessing the same tables in a shared database.
- In such cases, it is important that each user does not notice that the other is making use of the resource. This phenomenon is called Concurrency transparency.

### 7) Failure Transparency:-

Kus andar fail ho gaya system ko resource bandh ho gaya toh vo bhi chhupa de. Kaa.



Making a distributed system failure transparent means that a user does not notice that a resource fails to work properly, and that the system subsequently recovers from the failure.

### 8) Persistence Transparency:-

→ It deals with masking whether a resource is in volatile memory or perhaps somewhere on a disk.

→ For example, many object oriented databases provide facilities for directly invoking methods on stored objects

### \* Degree of Transparency:

Although distribution transparency is generally preferable for any distributed system, there are situations in which attempting to blindly hide all distribution aspects from users is not always a good idea.

An example is requesting your electronic newspaper to appear in your mailbox before 7 A.M. local time, as usual, while you are currently at the other end of the world living in a different time zone. Your morning paper will not be the morning paper you are used to.

The conclusion is that aiming for distribution transparency is a nice goal when designing and implementing distributed systems, but that it should be considered together with other issues such as performance.

# Software Concepts of distributed system

To select different operating system for distributed environment is slow concept.

Three largely used operating system

1) Distributed OS (DOS)

2) Network OS (NOS)

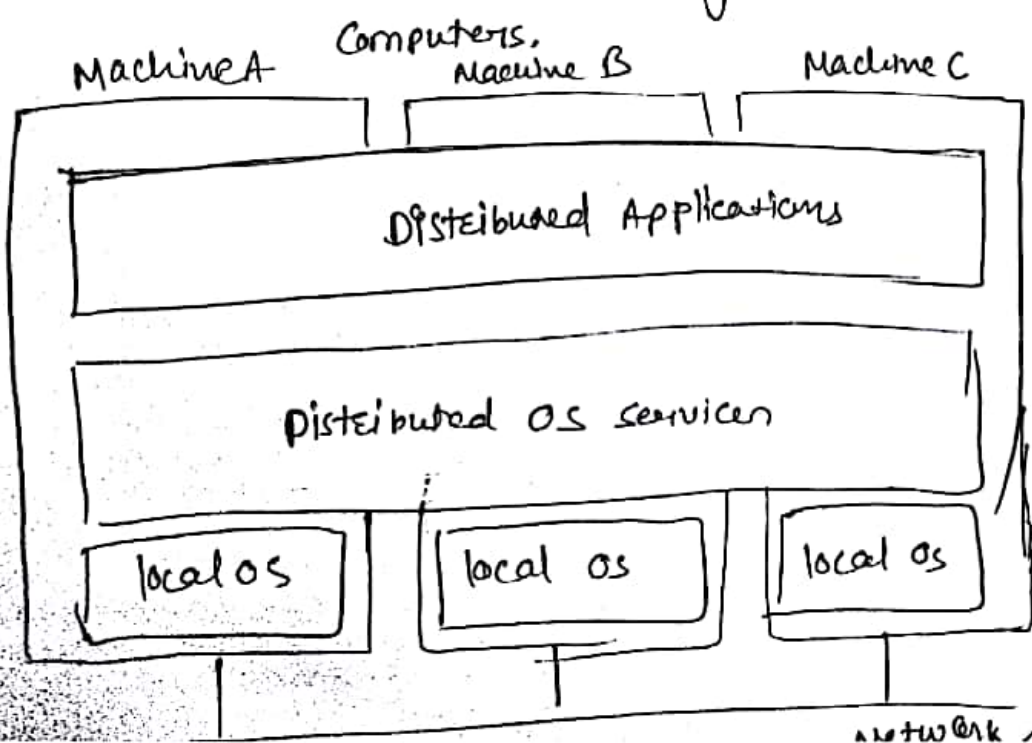
3) Middleware

## ① DOS

Isko samajhne kiliye pehle multiprocessor / multicomputer dekh lete hain

**Multiprocessor** - uses different system services to manage resources connected in a system and use system calls to communicate with the processor.

**multicomputer** - the distributed operating system uses a separate uniprocessor OS on each computer for communicating between different





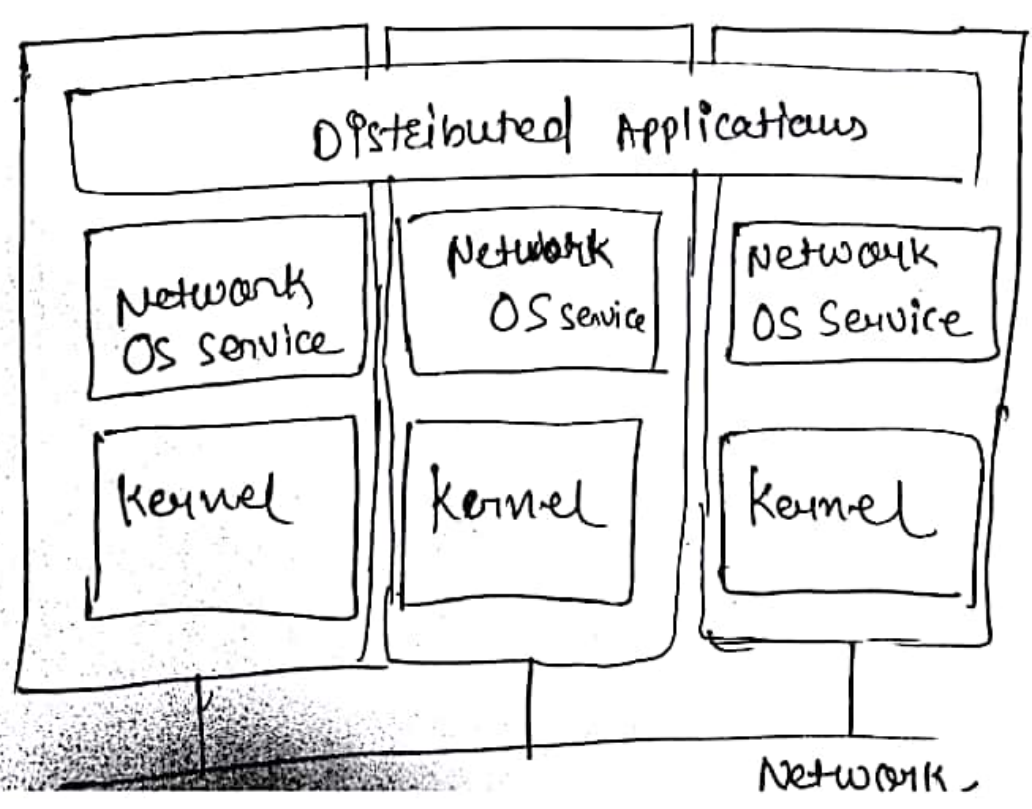
It communicates with all the computer using message passing interface. and it follows the tightly coupled architecture pattern.

Example Automated banking system, Railway Reservation

## 2) NOS

(Simply ye Networked data hai jab ki Dos me users ko machine ke network connection koi pata nhi rehta isme rehta hai)

- A NOS is made up of sw and associated protocols that allow a set of computer network to be used together.
- Environment users are aware of multiplicity of machines.
- Performance is badly affected if certain part of the h/w starts malfunctioning.
- It follows loosely coupled architecture pattern





## Middleware :-

(Ye Dos or Nos k bich me karta hai efficiency provide krne k liye)

As distributed operating system has lack of scalability and network operating system fails to provide a single coherent view so this is new layer formed between them known as middleware layer.

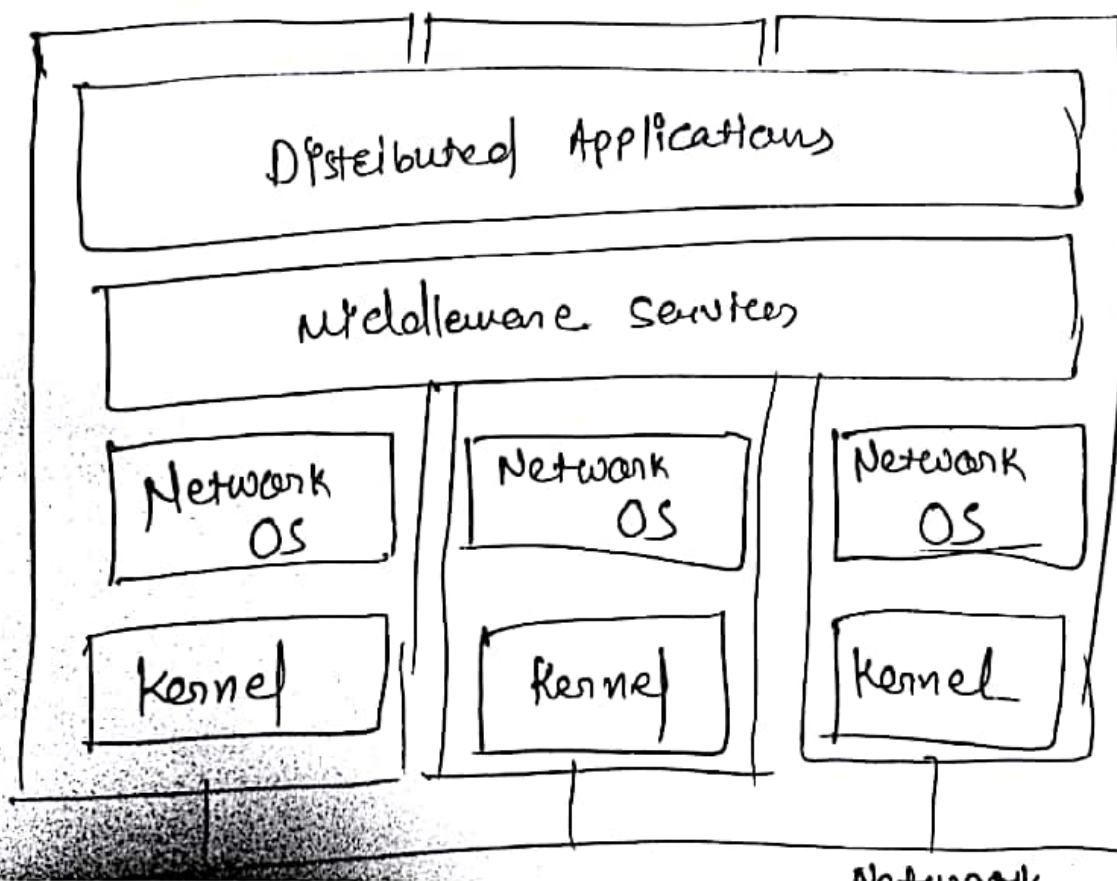
For local Application → Common set of services

Remote Application → Independent set of services

It provide the services as →

- locating objects
- finding location of objects
- handling the protocol information
- synchronization
- concurrency and security

Example Data Converter, Communication Controllers

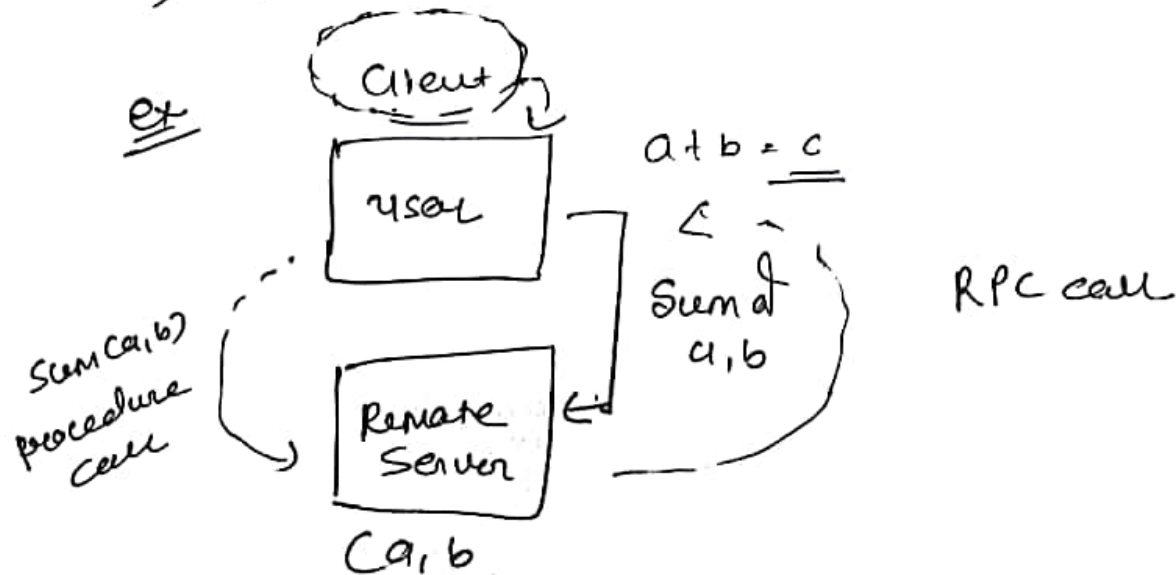


# Models of Middleware

- middleware refers to the software that is common to multiple applications and builds on the network transport services to enable ready development of new applications and network services
- The first middleware model was started with the distributed file system, where the files were stored and distributed over the network

## Model 1: Remote Procedure Call.

- ↳ one of the successful middleware models used in modern distributed applications for communication,
- ↳ It uses local call to call a procedure residing on the remote machine to get the result.
- ↳ client-server communication is hidden.



Therefore ~~the~~ though the method was executed remotely it appears like a local to the called machine.

This is synchronous technology Client/server both should be present.

## 2) Message oriented middleware (MOM)

- ↳ It uses data structures like queue to store and retrieve messages.
- ↳ Queue is used to get rid of misplaced messages



- ↳ It is asynchronous message can be sent if one of the sender/receiver is not present.

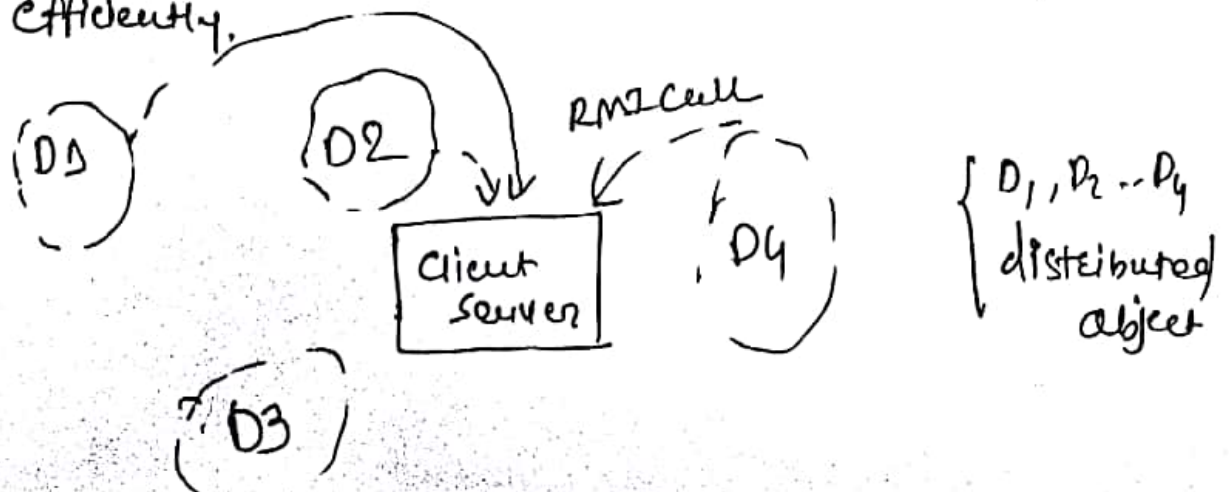
Ex. Email.

## Model 3:- Distributed Object

- In this objects are distributed to the remote server to facilitate client

Ex. RMI and CORBA

- This mechanism hides the communication interfaces and their details to provide access to the remote object efficiently.



## Model 4: peers to peers

↳ All processes play similar roles, interacting cooperatively as peers to perform a distributed activity or computation without any distinction between clients and servers.

↳ This model is fully distributed and parallel

ex: Process Migration  
Remote memory access  
P2P file exchange

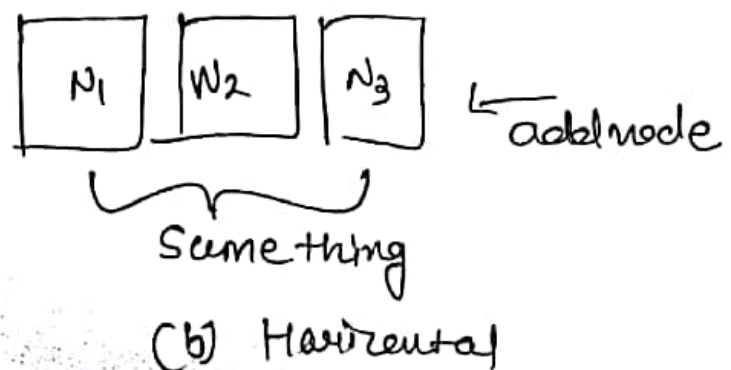
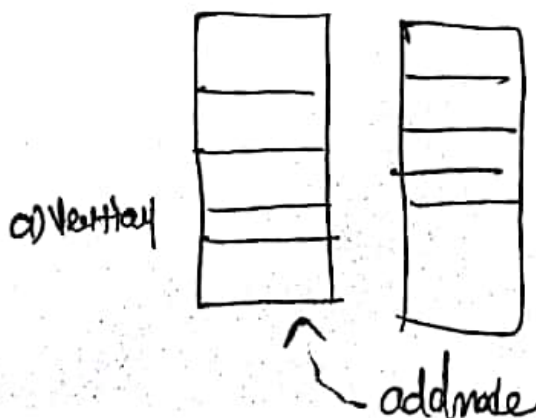
## Model 5: Vertical And Horizontal

Vertical:- Vertical distribution servers/machines on one node Authentication node server and on other node verification node server.

→ It handles the high load as we can add nodes separately for different functionality.

Horizontal → All machines do same thing for one service.

When load is high we have to add another node that can handle same task.





Thank you so much 😊  
videos and notes dekhne ko

if you have any doubt  
you can contact us at

website: [www.lastmomenttutions.com](http://www.lastmomenttutions.com)

email: [lastmomenttutions@gmail.com](mailto:lastmomenttutions@gmail.com)

whatsapp: 7038604912

facebook: [www.facebook.com/lastmomenttution](http://www.facebook.com/lastmomenttution)

Instagram: [www.instagram.com/last\\_moment\\_tutions](http://www.instagram.com/last_moment_tutions)

Strongly suggest to see  
*Sandeep makehwari*  
*videos*



*To  
change  
your  
life*