

16/7/22

→ Distributed middle ware:-

Applications

Domain specific services

Common middle ware service

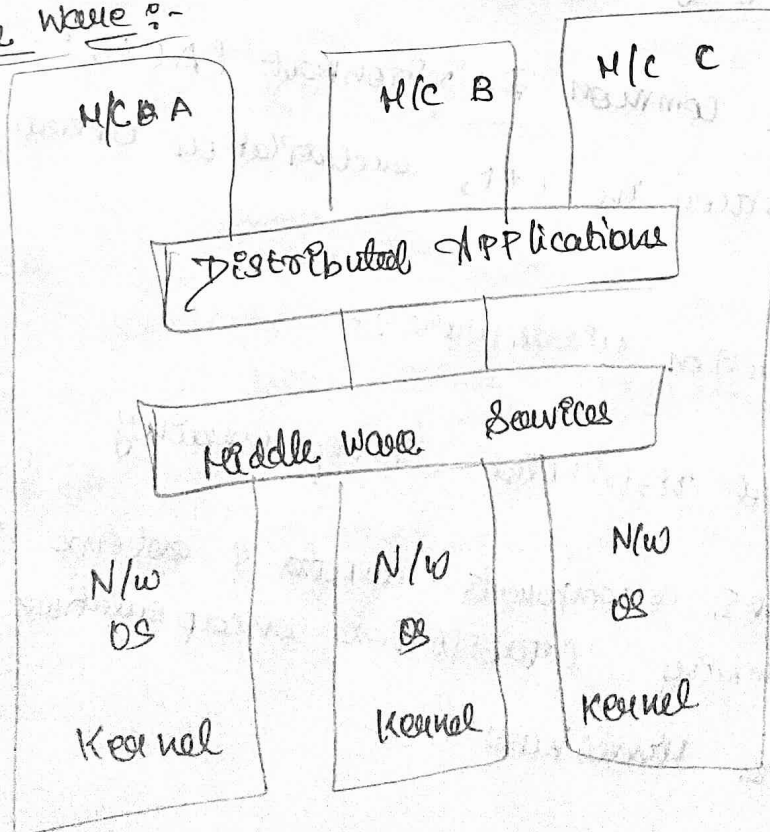
Distributed middle ware service

Host Infrastructure middle ware

OS & Protocols

Hardware devices

→ Middle ware :-



→ Host Infra structure :-

& encapsulate & enhance native OS & concurrency mechanism to create reusable.

\* Networking component

reactors, monitors objects, acceptors, active objects.

→ Sun JVM → platform independent

→ execute code → by abstraction different OS & CPU architecture.

→ NET :-

\* Microsoft platform xML service.  
\* Designed to connect information devices

common language Routine (CLR) - middle ware.

→ Adaptive Common Environment (ACE) :-

written in C++, encapsulation operating system

↳ Distribution middle ware :-

& higher level distributed programming

& reusable API & components automates & extend native OS  
N/w programming capabilities encapsulation by host

Infrastructure middle ware.

→ OH

→ Sun

→ Micro

\* component

→ SOA

\* I  
\* S

→ Com

Program  
Develop  
Directly

→ OMG CORBA :-

\* Common Object Request Broker Architecture.

\* object + + component Interoperate across middle ware.

→ SUN JAVA RMI :- (Remote Method Invocation)  
\* It is the methods invoked from other JVMs.

→ Message Oriented Middleware :- Component Object Model (COM)  
\* software component communicate via Remote component.

→ SOAP ⇒ (Simple Object Access Protocol) :-  
\* It enable auto web service  
\* shared open web infrastructure.

→ Common UI Services :-  
\* allow application Developer to contribute Program business logic without writing codes to Develop Distributed system use low level middle ware directly.

→ ORBI CORBA:- event notification logging, Multimeter  
streaming security, Global RT, RT scheduling,  
fault tolerance concurrency ENT & transaction

→ Sun Extended Java Bean:-  
\* Distributed system → prebuilt  
software & services called Beans (not code from  
scratch)

→ NET:-  
Application specific into components domain

→ Domain specific middleware:-  
\* used for particular domain such as telecom,  
economic, e-commerce, healthcare, process automation  
programmable hardware mach devices:-

→ ORBI:- (object mgmt group)

→ DTF (domain task forces)

→ E-commerce DTF:-  
a development of economic & electronic  
banking screens medical.

→ SV

18/7/92

\* coll

\* Dat

\* Time

\* clock

→ cc

\* S

force:

such

csp &

that

\* csp

that.

→ col

\* csp



→ Smartest Useful solution Organis!

\* Medical Image include ultrasound, URG, CP SCAN etc...

18/7/22

- \* collaborative signal processing
- \* Data gathering
- \* Time Dependent Systems
- \* clock Synchronization

→ collaborative signal processing (CSP)

\* Signal processing (SP) is an Electrical Engg sub field forces on analysing, modifying & Synthesise signals such as sound, image, scientific measurement

CSP & SP → process signals ~ to Extract info from that collaboration means work together

\* CSP → process signal to extract information from that.

→ collaboration :- means work together.

\* CSP :- It is a process signal received from group of Element.

\* Reasons - CSP :-

- \* unity is straight
- \* goal → Identify, Detect & track any object.
- \* Each sensor → process event for certain things
- \* Detect & store → local info

→ In Sensor Network:-

Elements → sensors

Target → location of target.

→ Time Dependent & Clock Sync :-

\* Distributed System (DS) :-

It is a collection of computer

via high speed common system.

\* In Distributed system, Hardware & software components communicate, coordinate by message passing.

\* Each node share information with other nodes.

\* Need proper allocation of Resources & help coordinator between different processes.

\* To resolve conflicts sync is used.

\* Sync achieved by using clock.

\* Time set based on universal time coordination (UTC) ▲

Clock Sync Ways

- External clock sync
- Internal

1) External clk syn :-

External clock is present at  
node for reference. for all nodes adjust the time  
accordingly.

ii) Internal clk syn :-

\* Each node has its own local time  
of share with other nodes.

\* all other nodes set time the

accordingly.

Two types of clock syn algorithm :-

① centralized

② Distributed.

① centralized :-

\* Time server used as reference.

\* Single time server propagates its  
time to all the nodes.

\* Single time systems  $\rightarrow$  Node fails  $\rightarrow$   
It lose sync

Eg:- Berkeley algo, Passive time service, Active  
time service etc...

② Distributed :-

- \* NO centralized time server
- \* Node adjust by local time.
- \* Overcome issues as centralized like scalability, single point failure etc.

avg

Eg:-

Global avg algo, Localized avg algo,  
Network time Protocol (NTP).