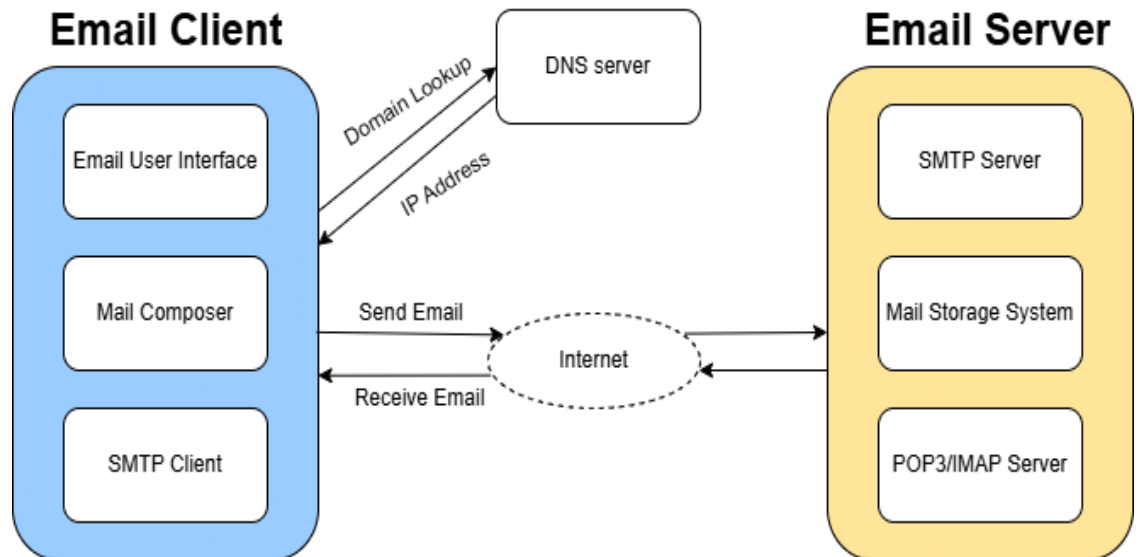


- 1) An email address consists of a user's name and a domain name. Sending a mail to a user does not involve knowing their physical or network location. Nor does the procedure to send an email message depend upon the location of the recipient.
 - i) Identify the 2 forms of transparency from this scenario
 - A) Location Transparency
 - B) Access Transparency
 - ii) **Email Server Architecture**



- 2) State and explain any three problems facing distributed systems' designers.
 - a) **Scalability**- The system must handle growth in users, data and resources without performance degradation. Designing systems that scale efficiently is difficult.
 - b) **Heterogeneity**- Distributed systems consist of different hardware, operating systems, networks, and programming languages. Ensuring smooth communication between diverse components is challenging.
 - c) **Fault Tolerance**- Failures such as network failures and server crashes are inevitable. For this reason, designers must ensure the system continues operating even when some components fail.
- 3) State and explain FOUR design concerns in the implementation of distributed systems
 - a) **Transparency**- The system should hide complexity such as location, replication, and failures from users.
 - b) **Security**- Protecting data from unauthorized access, attacks, and breaches is critical.
 - c) **Reliability and Availability**- The system should be dependable and accessible even in the presence of failures.
 - d) **Performance**- Minimizing latency and maximizing throughput despite communication delays.

- 4) A client gets a timestamp of 4:12:30.500 from a time server. The elapsed time between the request and response was 20 msec (0.020 sec). Using Cristian's algorithm, what time should the client set its clock to?

Cristian's Algorithm

Time received from the server= 4:12:30.500
 Round-trip time(RTT)= 20ms= 0.020 seconds
 One-way delay= RTT/2= 0.020/2= 0.010 seconds

$$\begin{aligned} \text{Client time} &= 4:12:30.500 + 0.010 = 4:12:30.510 \\ &= 4:12:30.510 \end{aligned}$$

- 5) A client attempts to synchronize with a time server. It records the round-trip times and timestamps returned by the server in the table below. Calculate time to which the client computers adjust

| |
|-----------------------|
| Round Trip ms() |
| 22 |
| 25 |
| 20 - (Minimum) |

RTT=200ms = 0.020s
 Time = 10:54:28.342

Cristian's Algorithm

$$\begin{aligned} \text{Client time} &= T_{\text{server}} + (\text{RTT}/2) \\ &= 10:54:28.342 + 0.020/2 \\ &= 10:54:28:342 \end{aligned}$$

- 6) High Availability -Users can access the voting system even when some nodes in the network cannot be reached
 Disaster Recovery - Protects against data loss through causes like fire since replicas are stored in geographically distributed data centers

7) Lamport Logical Clocks

a → b then a happens before b

P1, P2, P3

A vector clock has the form:

$[V_1, V_2, V_3]$

| Event | Process | Lamport Time | Vector Clock |
|-------|---------|--------------|--------------|
| e1 | P1 | 1 | [1,0,0] |
| e2 | P2 | 1 | [0,1,0] |
| e3 | P1 | 2 | [2,0,0] |
| e4 | P3 | 1 | [0,0,1] |
| e5 | P2 | 2 | [0,2,0] |

Causal event

$VC(a) \leq VC(b)$

$[1,0,0] < [2,0,0]$

e1 causally precedes e3

Concurrent event

Neither $VC(a) \leq VC(b)$ nor $VC(b) \leq VC(a)$

e2 = [0,1,0]

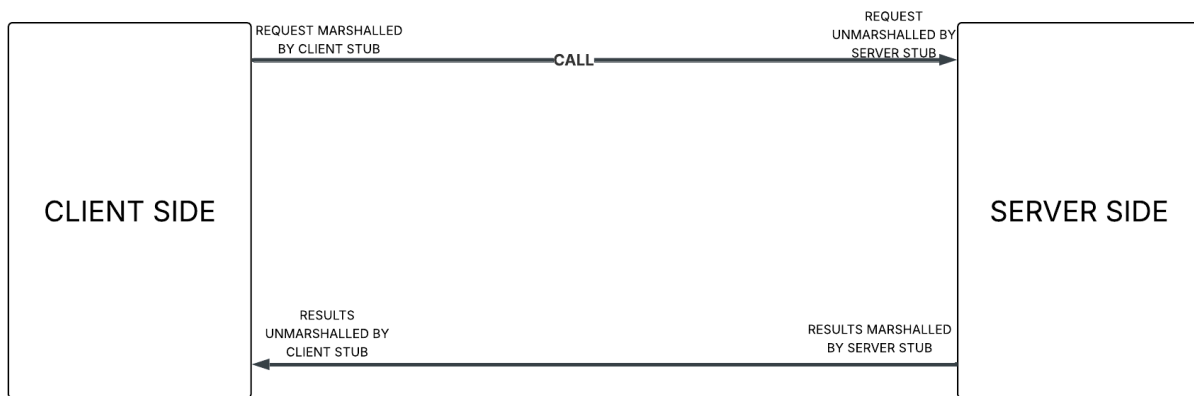
e4 = [0,0,1]

e2 and e4 are concurrent

- 8) Lack of Global Knowledge - Nodes only have local information, making agreement difficult
- Process Failures and Crash Handling - The leader may fail during the election or even after the election
- Concurrency of elections - Multiple nodes are capable of initiating elections simultaneously

- 9) Location Transparency - hides the location where resources are located
 Migration Transparency - hides the fact that a resource or a process can be moved
 Relocation Transparency - hides the movement of a resource or process while it is actively being used.

10) The RPC model has been basically used to demonstrate how messages may be passed between two or more remotely connected computers. With the help of a diagram, describe what takes place between the two machines during this process



11) You are asked to design a stock trading system where thousands of buy/sell orders are processed per second. Which architecture style would you choose and why? Event-Based Architecture; This is because of loose coupling meaning I do not need to know where the traders are in order to trigger an event, it is highly scalable thus many traders can subscribe to the system and it is very flexible as I can modify a component without affecting the entire system

12) Discuss four disadvantages of distributed systems

- Lack of a Global Clock: In distributed systems, computers run on different clocks, making it difficult to perfectly synchronize events. This can cause problems with event ordering, logging, coordination, and consistency of operations across the system.
- Inconsistencies (Data Consistency Issues): Because data is stored across multiple machines, updates may not appear immediately everywhere. This can lead to users seeing different versions of data, especially during network delays or failures.
- Security Challenges: Distributed systems involve many machines communicating over networks, which increases the attack surface. Data transmission can be intercepted, and unauthorized access is harder to control across multiple nodes.

- Management and Administration Difficulties: Monitoring, maintaining, updating, and troubleshooting many distributed nodes is complex. Failures are harder to detect and manage compared to a single centralized system.

13) In June 5 2019, a denial of service attack was launched against the 13 DNS servers. The attack managed to disable at least three of the 13 yet there was relatively little impact on the Internet at large. Give and explain two reasons why DNS was able to withstand this assault.

Redundancy through Distributed Architecture (No Single Point of Failure)

DNS is designed as a highly distributed system. Each of the 13 root servers is not a single machine but a globally distributed cluster of servers. Because of this architecture, when some nodes failed, others continued providing the service, demonstrating fault tolerance, a key property of distributed systems.

Replication of Services Across Many Nodes

DNS data (root zone information) is replicated across many servers worldwide. This means the same information exists in multiple locations, so even if several servers are unavailable, clients can still access the same data from other replicas

14) An e-commerce website is deployed across multiple data centers. When network delay occurs between data centers, different users temporarily see different stock values for the same product.

A. Identify the fault type

Network fault: The delay between data centers causes nodes to be temporarily unable to synchronize updates, leading to inconsistent system state.

B. Explain why users see different values

Because the system is distributed and replicated, each data center maintains its own copy of the stock data. When updates cannot propagate quickly due to network delay, replicas become temporarily inconsistent, so users connected to different data centers see different stock values.

C. How does the system finally recover consistency

The system recovers consistency through replica synchronization. Once the network delay is detected and communication is restored (failure detection), the replicas across the data centers exchange updates and reconcile differences. Using replication mechanisms, the system propagates the correct stock value to all nodes until every copy converges to the same final state (eventual consistency).

D. Outline two fault tolerance mechanisms used in this system

Replication: The system keeps multiple copies of data across different data centers. If one replica becomes unavailable or slow, other replicas can continue serving users, ensuring availability and reliability.

Failure Detection and Recovery: The system monitors components to detect failures (e.g., unreachable data center or delayed node) and then recovers by re-synchronizing replicas once communication is restored. This allows the system to return to a consistent state after faults.