

# Introduction:

Our world relies on complex applications, and behind the scenes, distributed systems play an essential role. These systems break down large tasks and distribute them across multiple computers, making everything faster and more efficient.

But how do these computers, potentially speaking different programming languages, communicate effectively? That's where CORBA comes in!

Hi everyone, I am Noor Fatima along with my groupmates Aqsa Saeed, Subhan Amjad and Zohaib and Today we are going to explain CORBA, Common Object Request Broker Architecture, it is a standard developed to facilitate communication between diverse and distributed computing systems. By enabling different applications written in various programming languages to work together seamlessly, CORBA plays a crucial role in integrating complex computing environments.

OVER TO SUBHAN

## History:

Now, let's talk about the history of CORBA.

In the late 1980s to early 1990s, the computing environment was becoming increasingly complex. To address these challenges, the Object Management Group (OMG) was founded in 1989. This led to the development of CORBA (Common Object Request Broker Architecture), a standard designed to enable seamless communication between diverse systems and programming languages.

## Architecture:

NOW, ME AQSA & NOOR WILL EXPLAIN THE ARCHITECTURE.

### **Object Request Broker (ORB)**

The ORB is the core of CORBA, facilitating communication between distributed objects. It handles client requests, routes them to server objects, and ensures seamless interoperability across different languages and platforms.

- AQSA WILL SPEAK NOW -

### **Interface Definition Language (IDL)**

IDL defines object interfaces in a language-neutral way, specifying operations, parameters, and return types. It creates a contract between clients and servers, and compiles into stubs for communication.

- NOOR WILL SPEAK NOW -

### **Client and Server Stubs**

Client stubs act as proxies, packaging method calls into CORBA requests. Server stubs, or skeletons, unpack these requests and forward them to server objects, handling data marshaling and unmarshaling.

- AQSA WILL SPEAK NOW -

### **Portable Object Adapter (POA)**

POA manages CORBA object lifecycles, including activation, deactivation, and persistence. It provides location transparency, enabling dynamic object management and seamless client interactions.

- SUBHAN WILL SPEAK NOW -

### **CORBA Services**

CORBA supports distributed computing with services like Naming for locating objects, Event for asynchronous communication, Transaction for data consistency, Security for secure communication, and Persistence for object storage.

NOW, ZOHAIB WILL EXPLAIN THE WORKING OF CORBA

# ***WORKING***

OVER TO AQSA

## **Advantages**

### **Interoperability**

CORBA enables different systems and applications to communicate seamlessly, regardless of programming language or platform. This is essential for integrating diverse systems.

### **Language and Platform Independence**

Supporting multiple languages and operating systems, CORBA ensures flexibility and adaptability, allowing developers to use the best tools without compatibility concerns.

### **Reusability**

CORBA allows the reuse of existing components across applications, reducing development time and costs, and enhancing efficiency.

### **Scalability**

Designed for efficient handling of large, distributed systems, CORBA scales from small applications to complex enterprise solutions without compromising performance.

OVER TO SUBHAN

## **Complexities**

### **Complex Configuration and Deployment**

CORBA requires extensive setup and configuration, making deployment challenging. Managing multiple components and services can be complex and time-consuming.

### **Performance Overhead**

The middleware layer in CORBA can introduce latency, affecting performance. This often necessitates performance tuning and optimization to ensure efficient operation.

### **Interoperability Issues**

Despite its goal of seamless communication, CORBA can face compatibility problems between different implementations, complicating interoperability.

NOOR WILL GIVE THE CONCLUSION.

## **Conclusion:**

In short, CORBA has been super important for letting different computers talk to each other, even if they speak different languages or run on different systems. It helps different applications work together smoothly. But, like everything, it has its ups and downs. Knowing both its good sides and the problems it can bring is key to using it well in today's computer systems.

We are now open to questions and discussions.