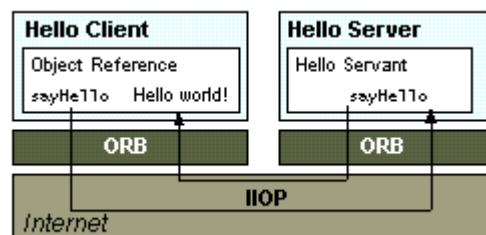


## SHORT NOTES / JAVA IDL

1. JAVA IDL is for CORBA programmers, this is business as usual
2. Using JAVA IDL java adds CORBA capability to JAVA language giving interoperability and connectivity based on Standards
3. JAVA IDL enables a distributed , web enabled java application to transparently invoke operations on remote network services using industry standards OMG IDL and IIOP
4. JAVA IDL technology is for programmers who needs to program for CORBA interfaces
5. Each language that support CORBA has mapping for CORBA based interfaces



- 6.
7. On the client side application includes a reference for the REMOTE object
8. That object reference has a STUB method , which is a stand in for the method being called remotely
9. The STUB is actually wired into the ORB, so that it invokes the ORB'S connection capabilities which forwards the invocation to the server
10. On the server side, the ORB uses SKELETON code to translate the REMOTE invocation into a METHOD CALL on the LOCAL object. The skeleton translates the cal and any parameters to their implementation specific format

11. When the method returns , the SKELETON code TRANSFORMS results or errors and sends them back to the client via the ORBS
12. Between ORB(S) communication proceeds by means of s SHARED PROTOCOL. IOP (Internet Inter ORB Protocol)
13. IOP defines HOW CORBA compliant ORB(S) pass information back and forth
14. CORBA object are described by IDL (Interface Definition Language)
15. If a JAVA client is invoking CORBA server , the server has an IDL interface describing the object
16. CORBA can be difficult to use if SERVER / CLIENT is behind a firewall or network address translation being used
17. CORBA does not support DISTRIBUTED GARBAGE COLLECTION
18. RMI supports DISTRIBUTED GARBAGE COLLECTION
19. CORBA usually provides following services
  - a. Object life cycle - Defines how CORBA objects are created , removed , moved and copied
  - b. Naming - Defines how CORBA objects can have friendly symbolic names
  - c. Events - Decouples the communication between distributed objects
  - d. Relationships - Provides arbitrary typed n-ary relationships between CORBA objects
  - e. Externalization - Coordinates the transformation of CORBA object to and from external media
  - f. Transactions - Coordinates atomic access to CORBA objects

LIYANA ARACHCHIGE RANIL

- g. Concurrency Control - Provides a locking service for CORBA objects in order to ensure serializable access
- h. Property - supports the association of name value pairs
- i. Trader - Supports the finding of CORBA objects based on properties describing the service offered by the object
- j. Query - Supports queries on objects