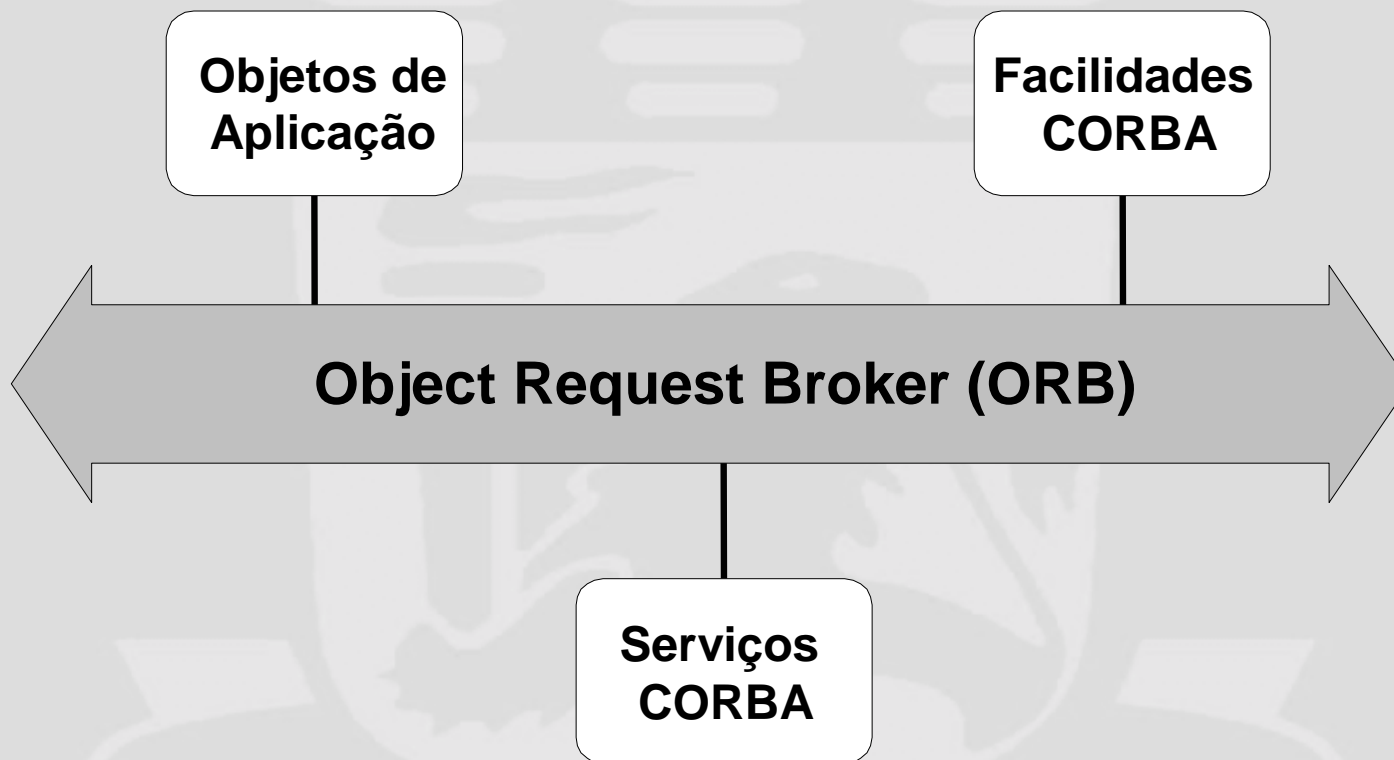

O Adaptador de Objetos e Interoperabilidade CORBA

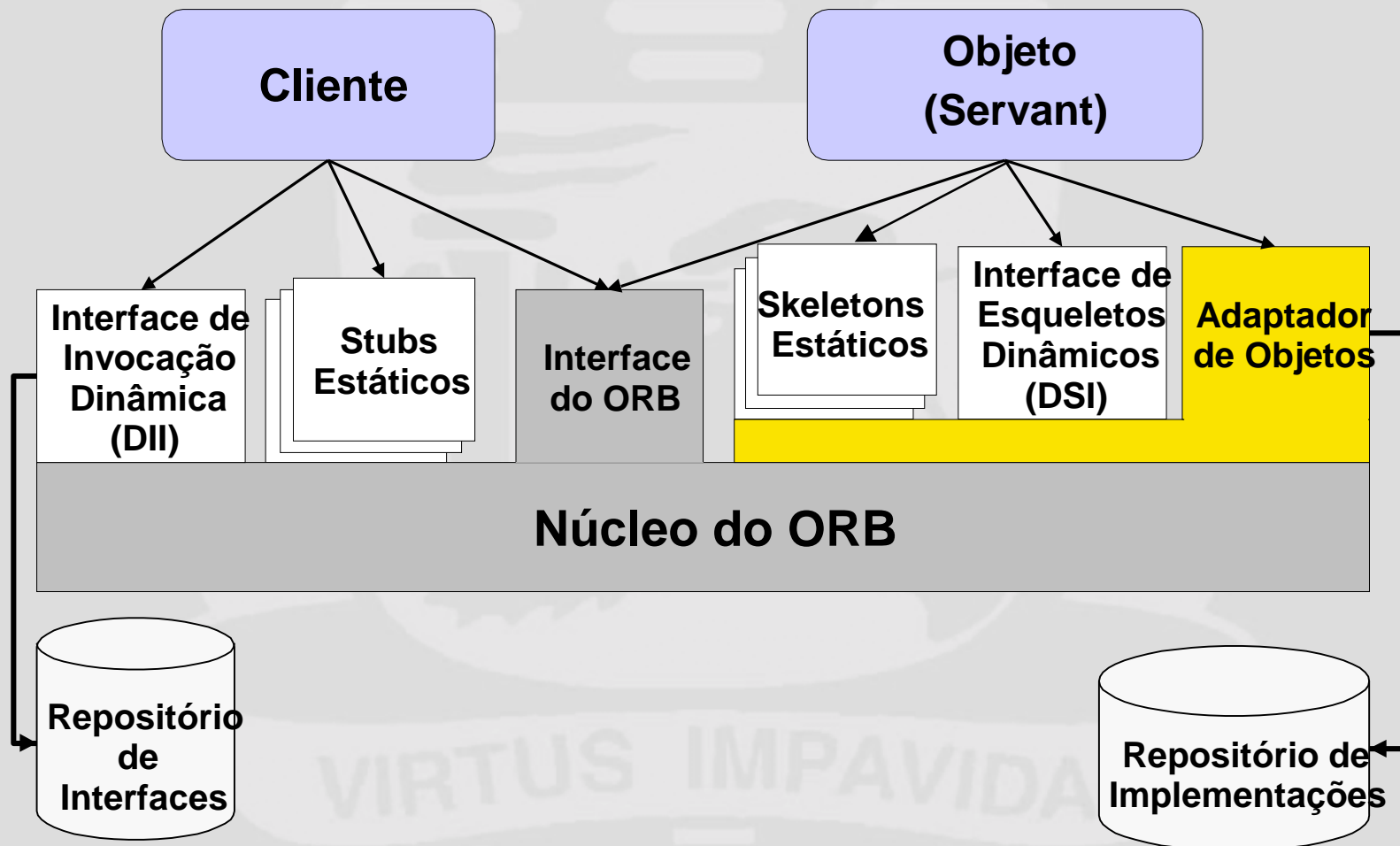
VIRTUS IMPAVIDA

O Modelo de Referência OMA



Object Management Architecture

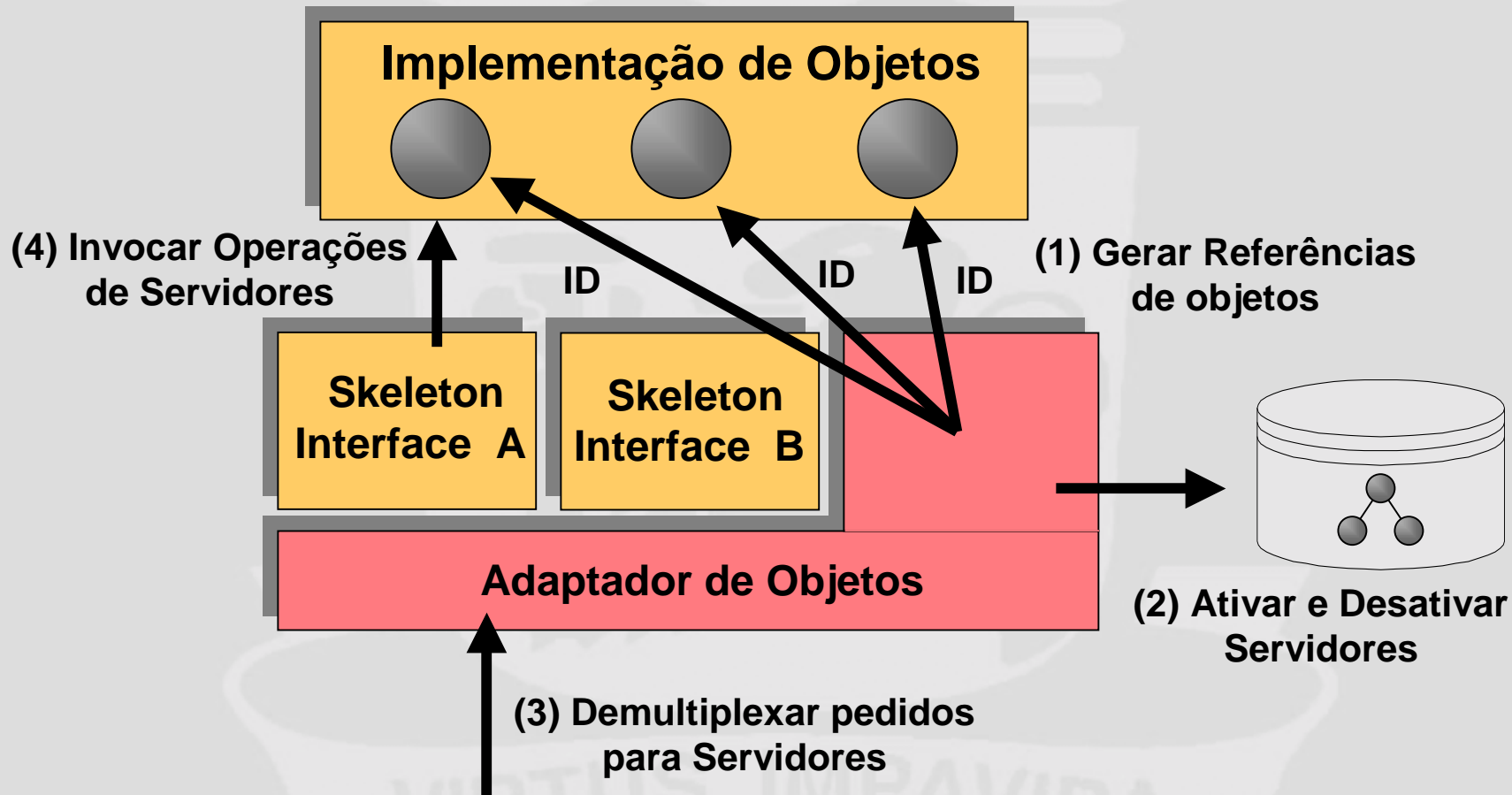
ORB CORBA



O Adaptador de Objetos

- ❑ O Adaptador de Objetos de CORBA é responsável por:
 - ❖ Gerar Referências de Objetos
 - ❖ Ativar e desativar servidores (Servants)
 - ❖ Demultiplexar pedidos para servidores
 - ❖ Colaborar com esqueletos IDL para invocar operações em servidores

O Adaptador de Objetos



Portable Object Adapter (POA)

- ❑ O POA (Portable Object Adapter) é a especificação atual para o adaptador de objetos CORBA
- ❑ Substitui a versão anterior denominada BOA (Basic Object Adapter)

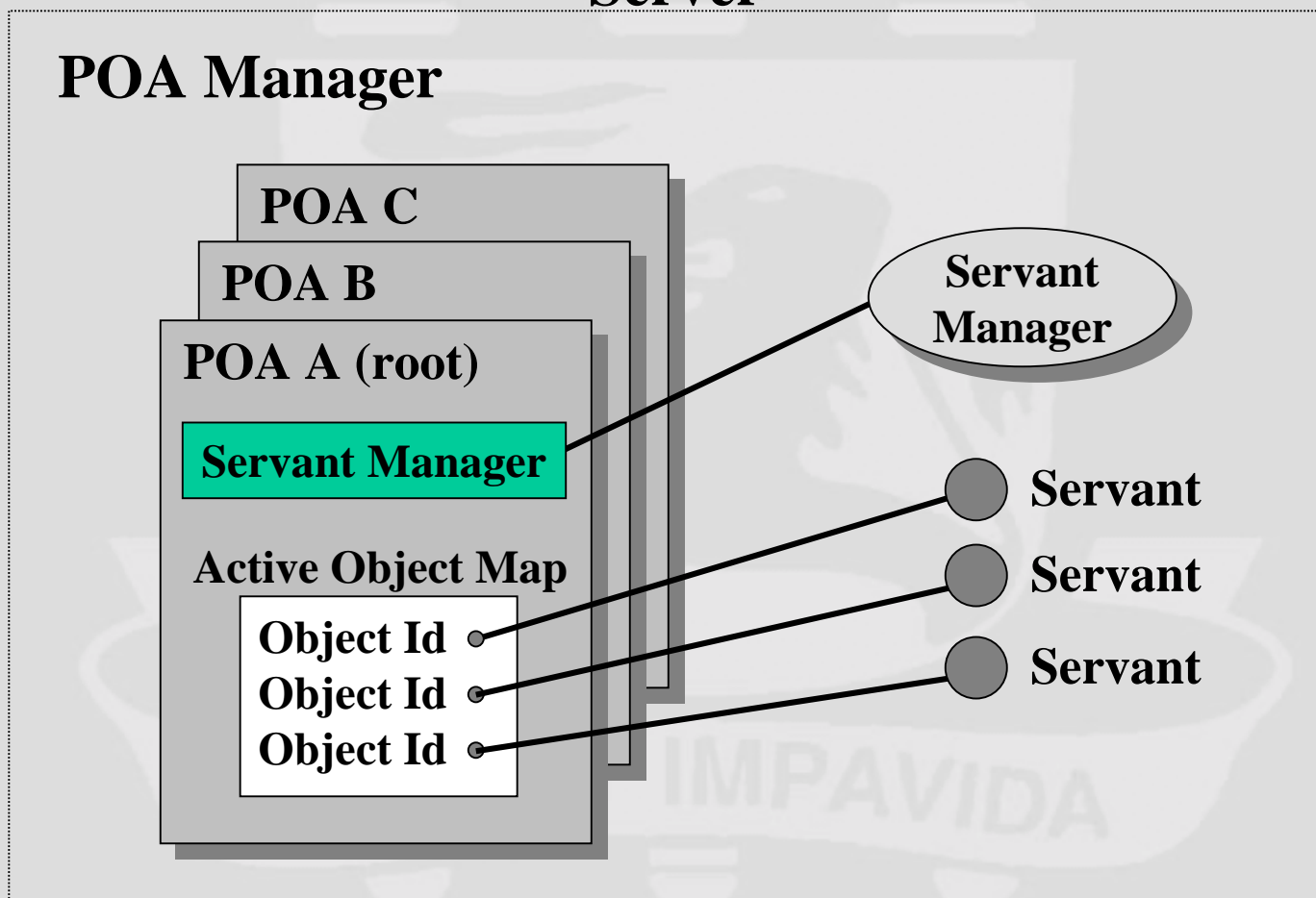
Portable Object Adapter (POA)

❑ Objetivos do POA

- ❖ Portabilidade: permitir a construção de servidores realmente portáteis entre ORBs de diferentes fabricantes
- ❖ Persistência: permitir a criação de objetos persistentes
- ❖ Utilização de Políticas: permitir a associação de políticas aos servidores
- ❖ Aninhamento de POAs: possibilitar a criação de múltiplas instâncias do POA no servidor

Arquitetura do POA

Server



Componentes do POA

❑ Client

- ❖ Um contexto computacional que realiza invocação de operações

❑ Server

- ❖ Um contexto computacional que implementa um objeto ou mais objetos

❑ Servant

- ❖ É um objeto de linguagem de programação que implementa pedidos em um ou mais objetos

Componentes do POA

❑ Object Id

- ❖ Identifica um objeto CORBA dentro de um servidor. Pode ser fornecido pelo usuário ou pelo próprio POA

❑ Object reference

- ❖ Estrutura de dados que identifica um objeto CORBA. Encapsula um Object Id e a identidade do POA que o controla

Componentes do POA

□ POA

- ❖ Entidade identificável no contexto de um servidor
- ❖ Fornece um *namespace* para *object ids* para outros POAs
- ❖ RootPOA pode ser obtido com `resolve_initial_references`

□ Policy

- ❖ É um objeto associado a um POA. Define características compartilhadas por todos os objetos implementados naquele POA

Componentes do POA

❑ POA manager

- ❖ É um objeto que encapsula o processamento de estados de um ou mais POAs

❑ Servant manager

- ❖ Um objeto que pode ser definido pelo usuário para permitir que o POA possa ativar e desativar servants. Existem dois tipos de Servant Managers
 - ServantActivator: para objetos persistentes
 - ServantLocator: para objetos transientes

Políticas do POA

- ❑ Políticas são utilizadas para controlar o ambiente de execução dos servants
- ❑ Os seguintes comportamentos podem ser gerenciados através de políticas:
 - ❖ Thread policy
 - ORB_CTRL_MODEL: O POA é responsável pelo gerenciamento de threads (default)
 - SINGLE_THREAD_MODEL: Apenas uma thread por servant

Políticas do POA (cont)

❑ Lifespan policy

- ❖ TRANSIENT: objetos não podem “sobreviver” fora do POA que os criou (Default)
- ❖ PERSISTENT: objetos podem “sobreviver” fora do POA que os criou

❑ Id assignment policy

- ❖ USER_ID: a aplicação fornece Object Ids
- ❖ SYSTEM_ID: o POA fornece os Object Ids (Default)

POA: Conclusão

- ❑ O POA permite o controle “total” do ambiente de execução do servidor
- ❑ É um padrão aceito e bastante completo para realizar as tarefas complexas do adaptador de objetos CORBA
- ❑ Introduce um esforço de programação maior, mas garante portabilidade das aplicações



Interoperabilidade entre ORBS
CORBA

Interoperabilidade

- ❑ A idéia de interoperabilidade entre ORBs CORBA surgiu com o CORBA 2.0
- ❑ A partir dessa versão foi introduzido um protocolo para permitir interoperabilidade

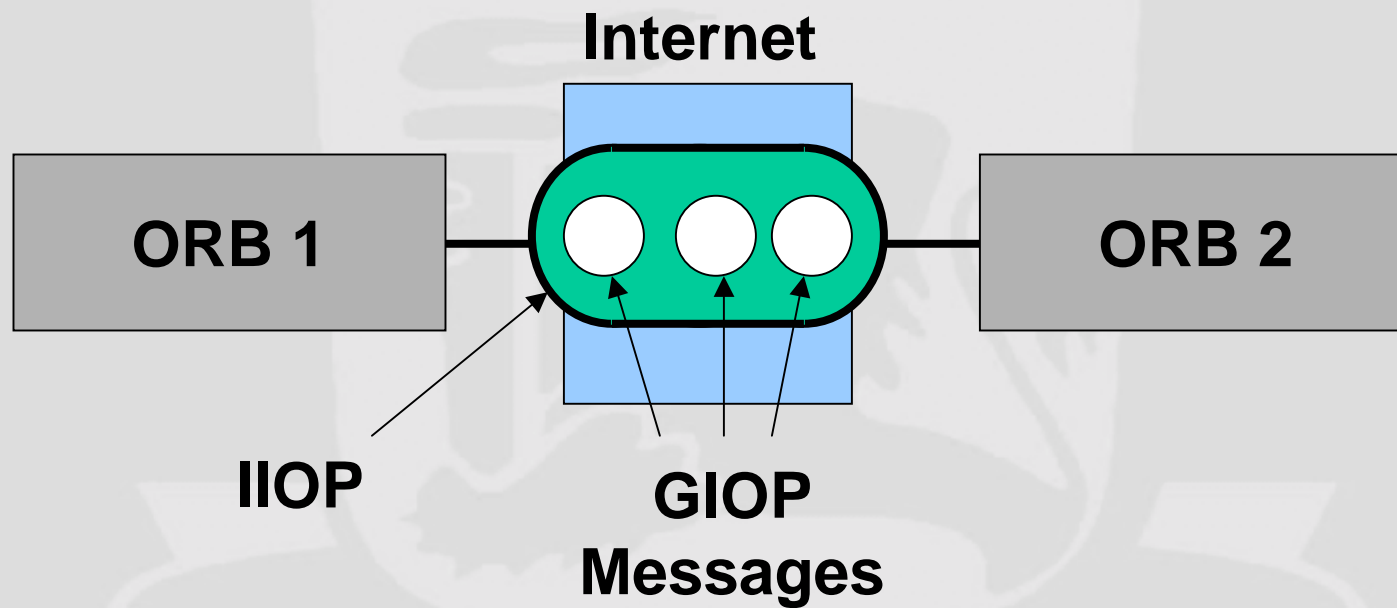
Interoperabilidade

- ❑ GIOP (General Inter-ORB Protocol) permite a interação entre ORBs de fabricantes diferentes
- ❑ GIOP é independente da arquitetura do ORB
- ❑ GIOP é executado sobre um protocolo de transporte orientado a conexão
- ❑ GIOP também é neutro com relação a esse protocolo de transporte

GIOP e IIOP

- ❑ O IIOP (Internet Inter-ORB Protocol) é uma especialização do GIOP para TCP/IP
- ❑ IIOP define como agentes abrem conexões TCP/IP e as utiliza para transferir mensagens GIOP

GIOP e IIOP



GIOP e IIOP

- ❑ GIOP e IIOP resolvem problemas de interoperabilidade da seguinte forma:
 - ❖ Permitindo que diferentes ORBs interajam escondendo fatores de implementação
 - ❖ Suportando todas as funcionalidades de CORBA através de diferentes ORBs
 - ❖ Preservando o conteúdo e a semântica das informações específicas entre os ORBs

O IIOP

❑ Formado por duas partes distintas

❖ Common Data Representation (CDR)

- Tipos de Dados CORBA
- Representação de baixo nível

❖ Mensagens GIOP

- Handshaking de comunicação

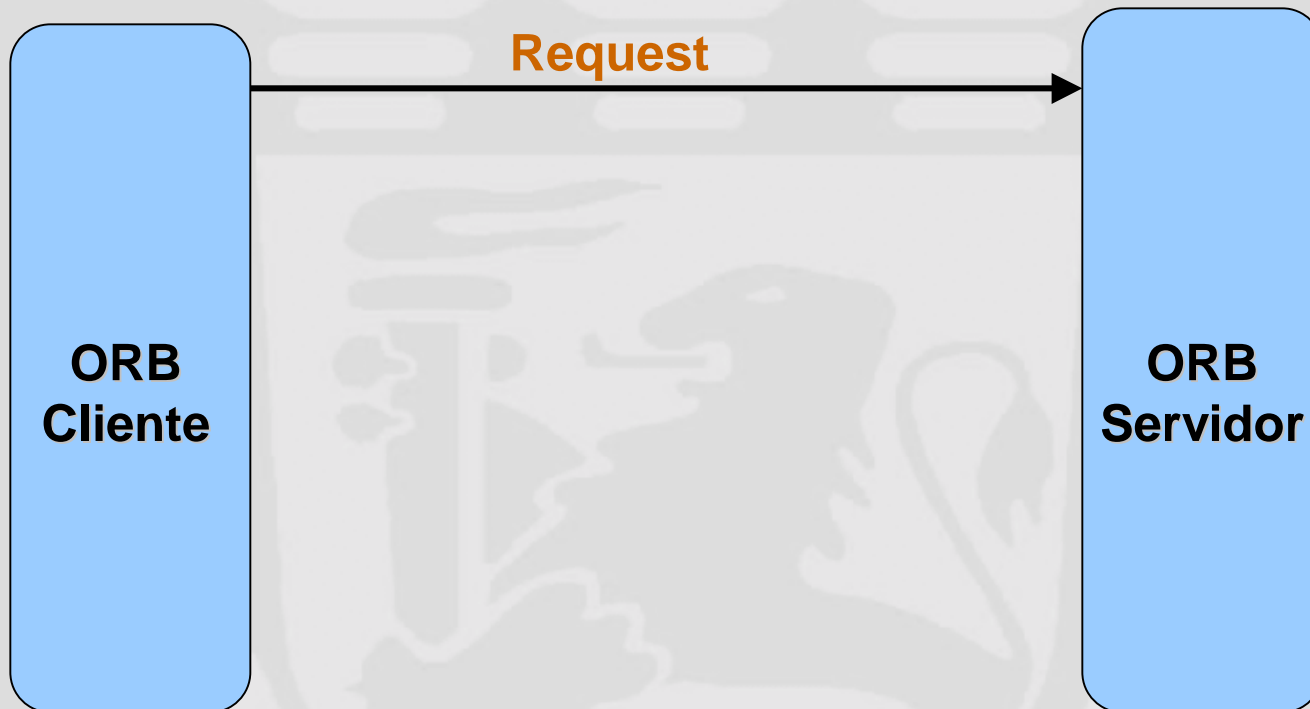
Common Data Representation

- ❑ CDR é a sintaxe utilizada para transferência de informações em comunicações IIOP
- ❑ Usada para mapear os tipos de dados CORBA IDL em uma representação comum de mais baixo nível
- ❑ Suporta:
 - ❖ Tipos Primitivos – char, short, float, boolean, etc.
 - ❖ Tipos Construídos – union, array, sequence etc.
 - ❖ Pseudo-objetos – exceptions
 - ❖ Referências de Objetos

Mensagens GIOP

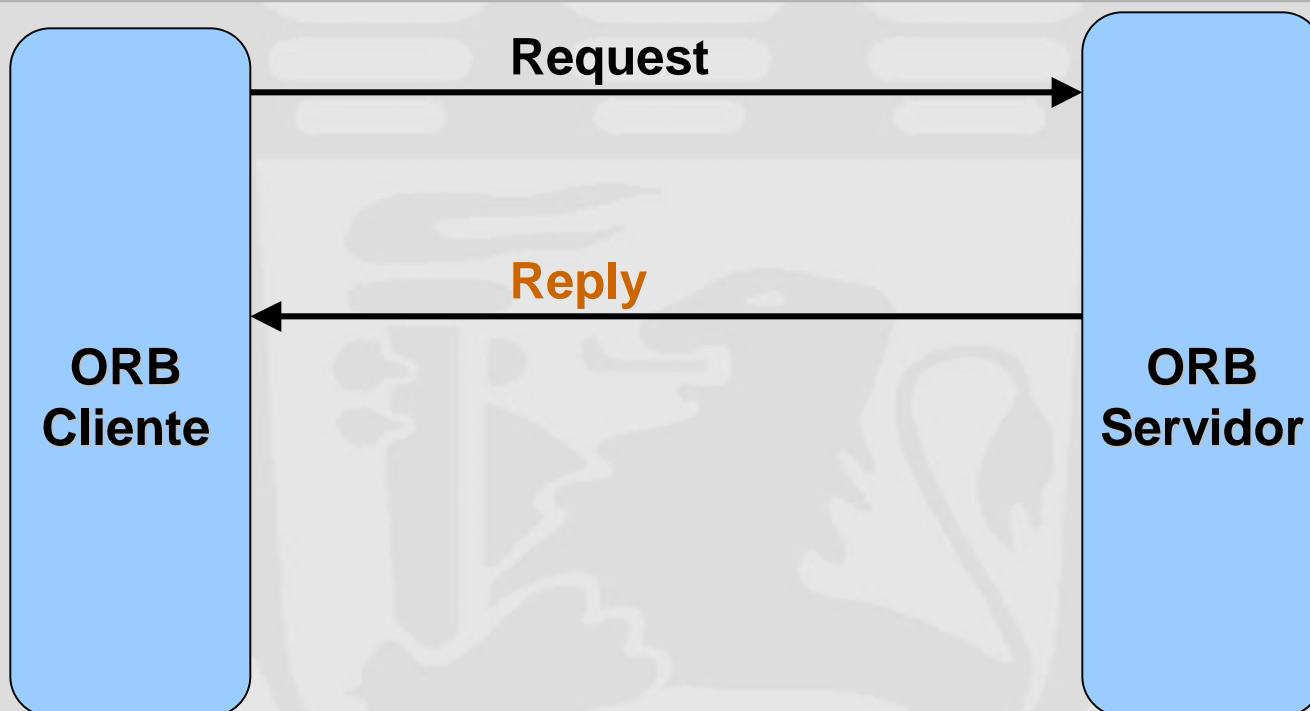
- ❑ 8 formatos de mensagens no total
- ❑ Mensagens geradas apenas pelo Cliente
 - ❖ Request msg
 - ❖ LocateRequest msg
 - ❖ CancelRequest msg
- ❑ Mensagens geradas apenas pelo Servidor
 - ❖ Reply msg
 - ❖ LocateReply msg
 - ❖ CloseConnection msg
- ❑ Cliente e Servidor suportam ambas
 - ❖ MessageError msg
 - ❖ Fragment msg

Mensagens GIOP



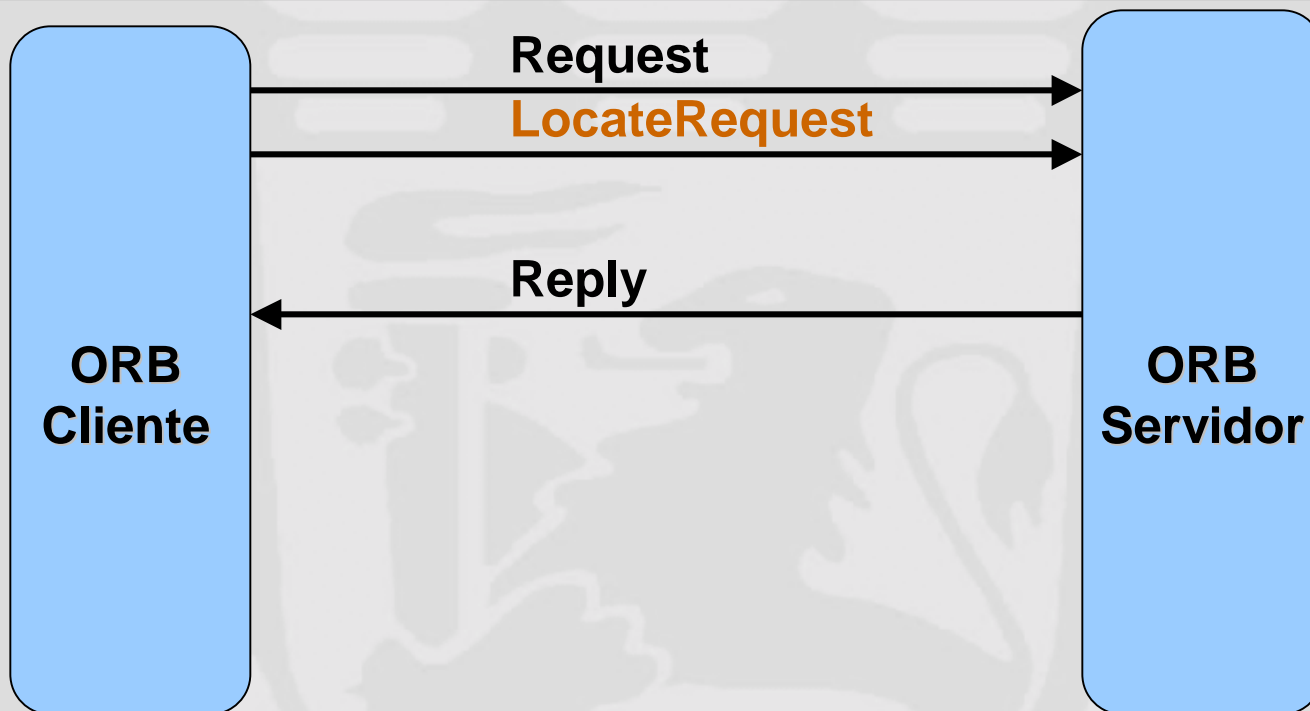
TCP/IP : GIOP message header, Request header (ID), Request body (parâmetros)

Mensagens GIOP



**GIOP header, Reply header (ID do Request),
Status Code (se no_exception retorna os parâmetros
out e inout)**

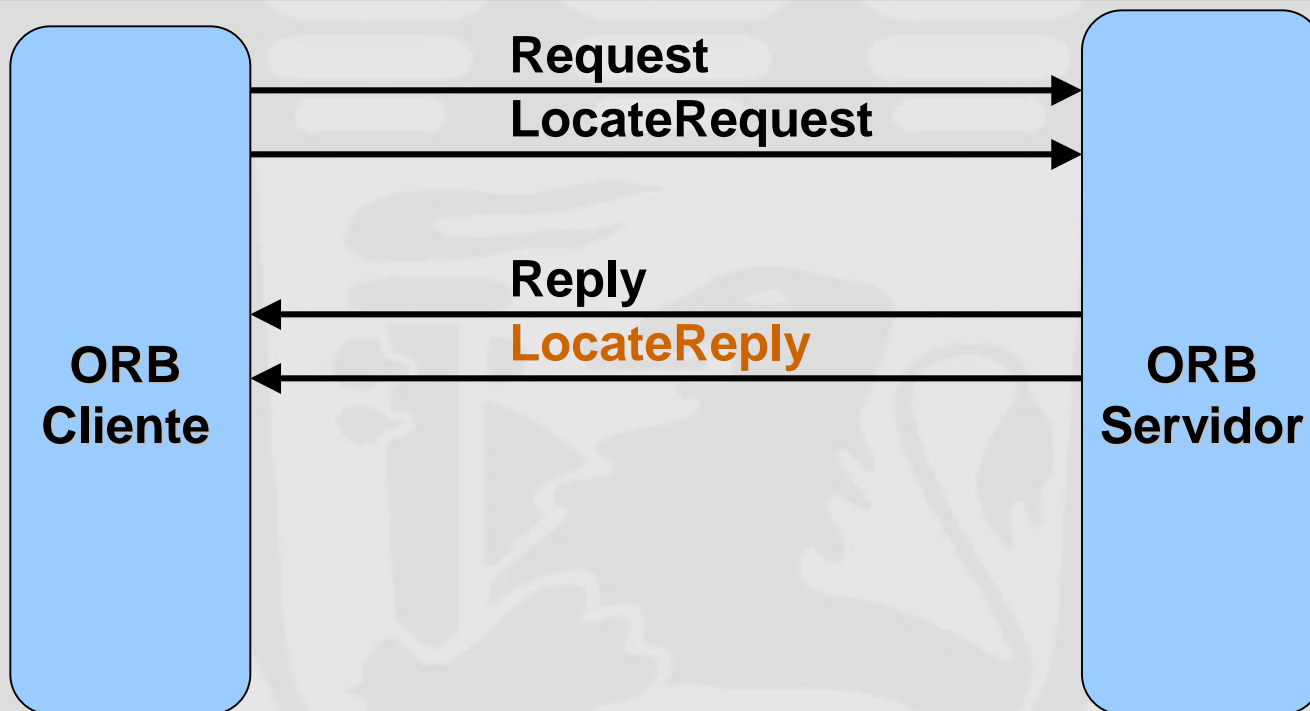
Mensagens GIOP



A referência de Objeto é válida ?

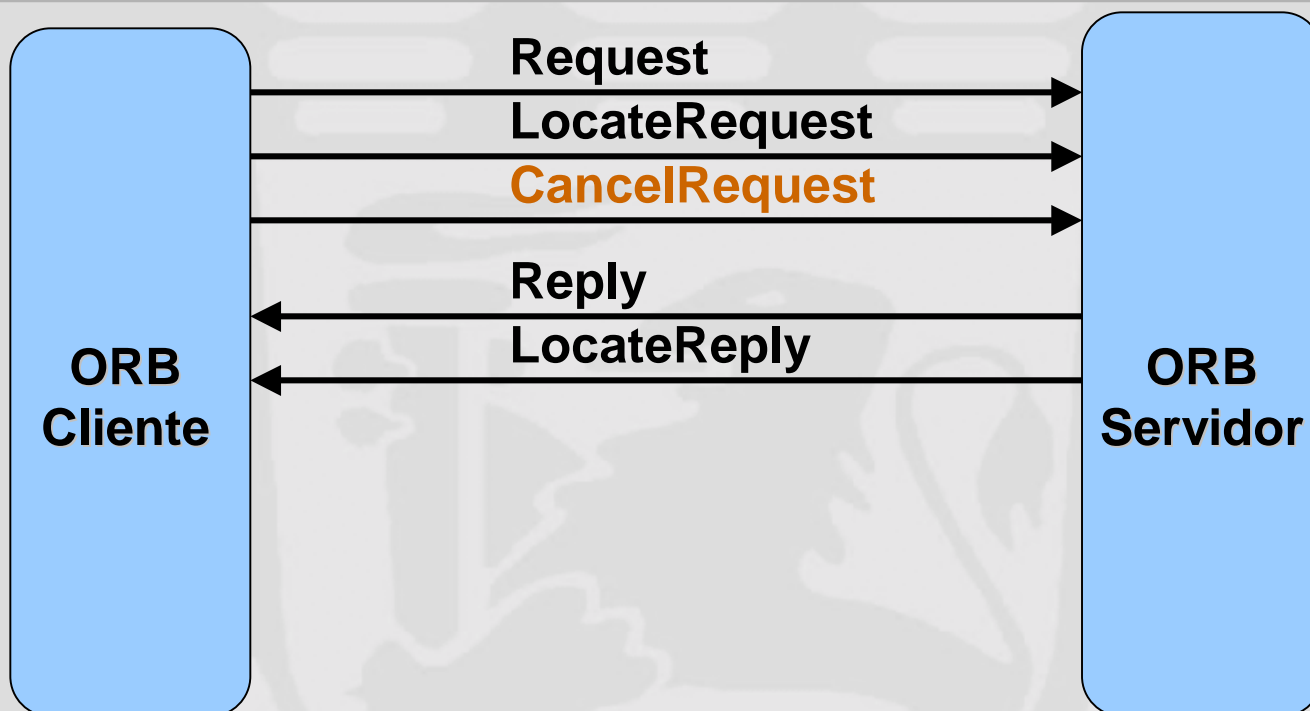
O Servidor está recebendo pedidos ? Se não, qual endereço a usar ?

Mensagens GIOP



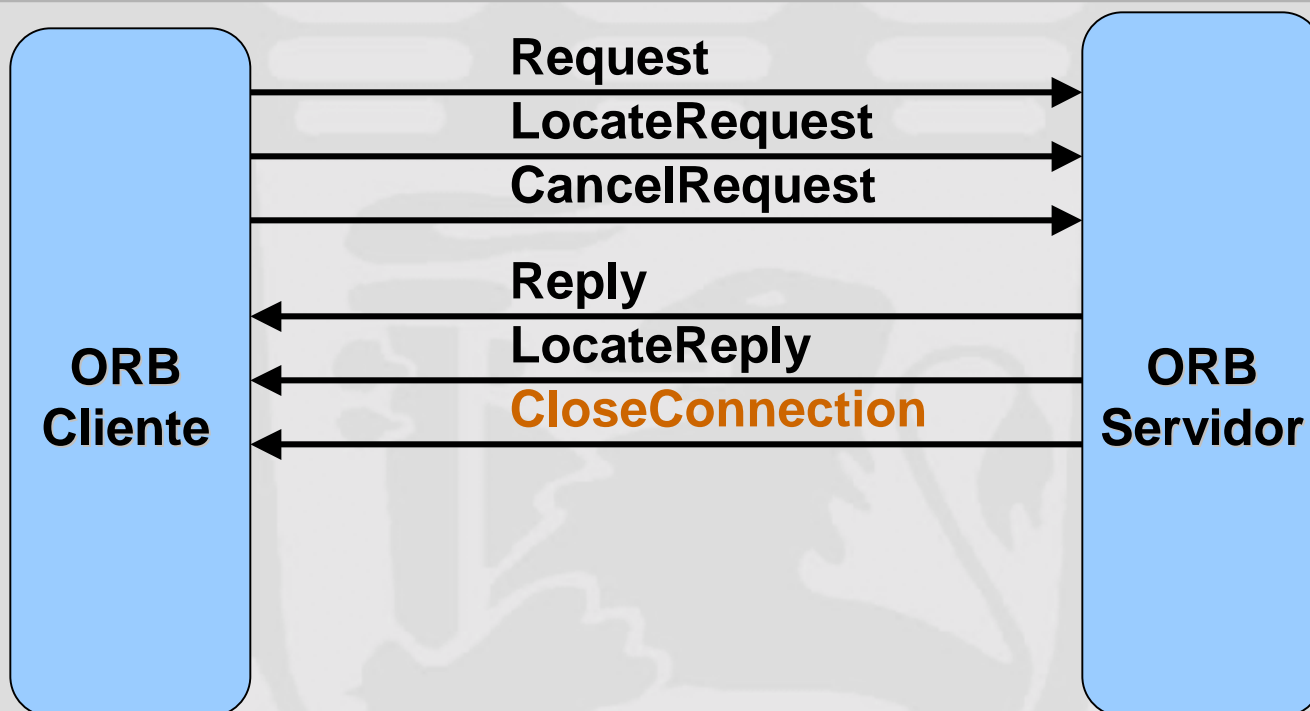
É a resposta para o LocateRequest. Se o endereço mudou, retorna o endereço para os próximos pedidos

Mensagens GIOP



Avisa ao servidor que o cliente não está mais interessado em receber reply de um Request específico

Mensagens GIOP



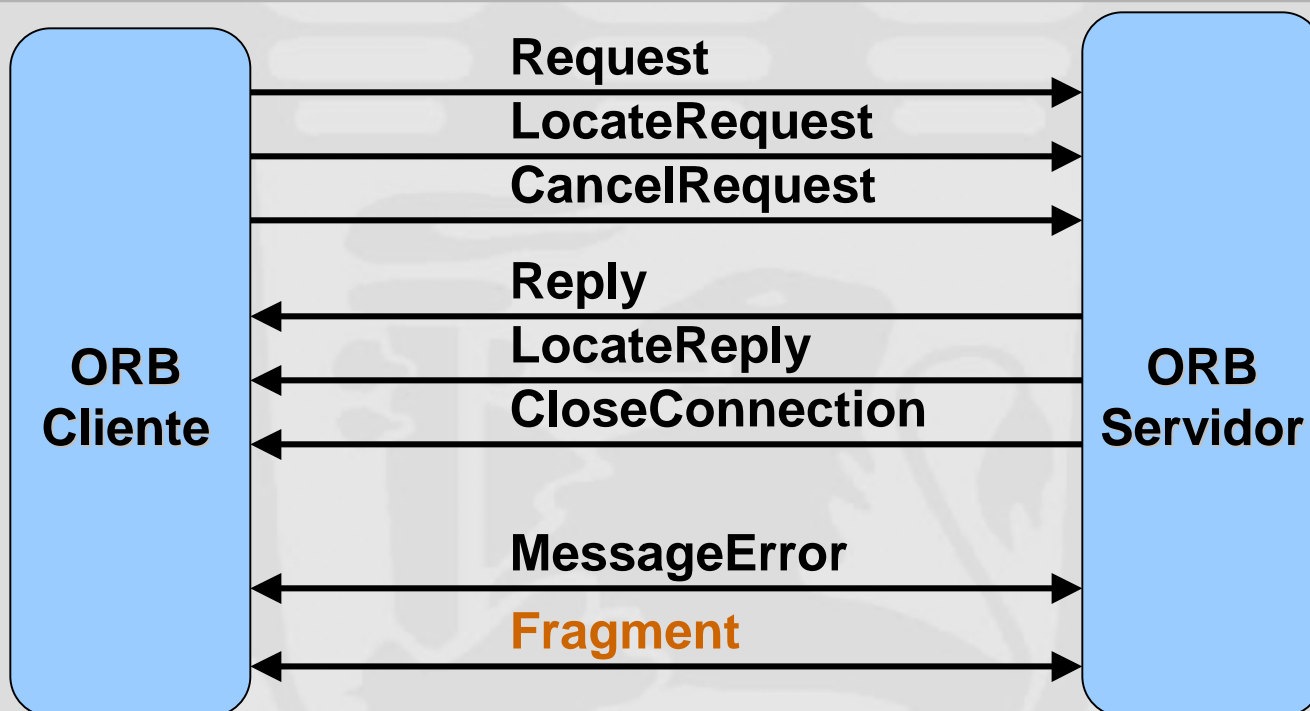
Informa ao cliente para não esperar mais nenhuma informação do servidor

Mensagens GIOP



Erros não interpretados pelo servidor, como versão de protocolo não reconhecida, etc.

Mensagens GIOP



Fragmentos de uma mensagem longa

GIOP: Conclusão

- ❑ GIOP é um protocolo independente de qualquer tipo de rede particular
- ❑ GIOP por si só é insuficiente para permitir comunicação cliente-servidor
- ❑ IIOP é o mapeamento de GIOP em redes TCP/IP
- ❑ IIOP é o protocolo que o OMG adotou como padrão e deve ser suportado por TODOS os ORBs CORBA, ou de forma nativa ou através de pontes

GIOP: Conclusão

- ❑ IIOP requer que agentes possam aceitar pedidos de objetos ou fornecer localização de objetos através de seus endereços IP
- ❑ Esses endereços são publicados através de IORs (Interoperable Object References)
- ❑ Clientes requerem serviços de objetos inicializando uma conexão através de um IOR específico