Sistemas Distribuídos

CORBA

Edeyson Andrade Gomes

www.edeyson.com.br

Introdução

- Middleware
  - Camada de software que abstrai a complexidade e a heterogeneidade do ambiente distribuído

- Objetivos
  - Facilita a tarefa de projetar, programar e gerenciar aplicações distribuídas

  - Provê um ambiente de programação distribuído consistente e integrado

Introdução

- Middleware baseado em objetos
  - Aplicações são estruturadas em objetos distribuídos que interagem através da invocação de métodos

- Funcionalidades
  - Linguagem de Definição de Interface (IDL)
    - Abstração da linguagem de programação
  - *Broker* de requisições de objetos
    - Direciona as invocações de métodos para o objeto alvo apropriado de forma transparente
  - Conjunto de serviços
    - nomeação, transações, replicação, etc.

CORBA

- CORBA (Common Object Request Broker Architecture)
  - Proposto pela OMG em 1991
  - Descreve uma interface com mapeamento padronizado para diversas linguagens e um conjunto de serviços básicos

- CORBA 2 (1996)
  - Interoperabilidade entre desenvolvedores
    - Definição de padrões (GIOP – General Inter-ORB Protocol)
  - Protocolo IIOP (Internet Inter-ORB Protocol)
    - Implementação do GIOP para Internet (TCP/IP)

CORBA

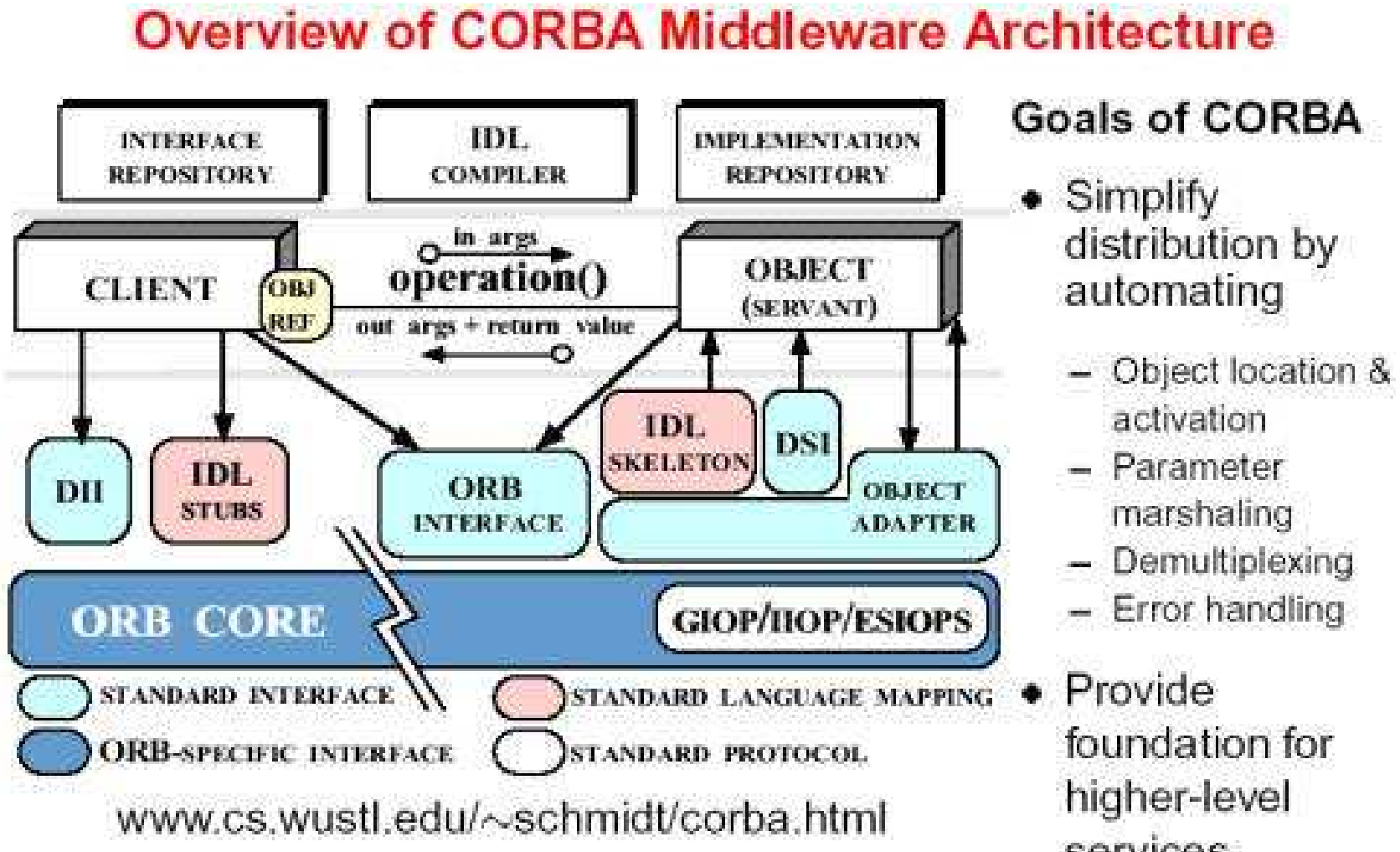
- *"The paradigm CORBA follows is a combination of two existing methodologies. The first is distributed client-server computing. It is based in part on message-passing systems, most commonly found in UNIX-based environments. The second methodology is object-oriented programming."*

  - *Gabriel Minton*

CORBA

- CORBA 3 (2002)
  - CORBA Component Model
    - Modificações realizadas no CORBA para inclusão de componentes (mudanças na IDL, repositório de interfaces, etc) – CIDL (Component Interface Definition Language)
  - Novos serviços
    - Fault-Tolerant CORBA, Quality of Service, Real-Time CORBA, Asynchronous Messaging, etc

Arquitetura



Goals of CORBA

- Simplify distribution by automating
  - Object location & activation
  - Parameter marshaling
  - Demultiplexing
  - Error handling
- Provide foundation for higher-level services

Arquitetura

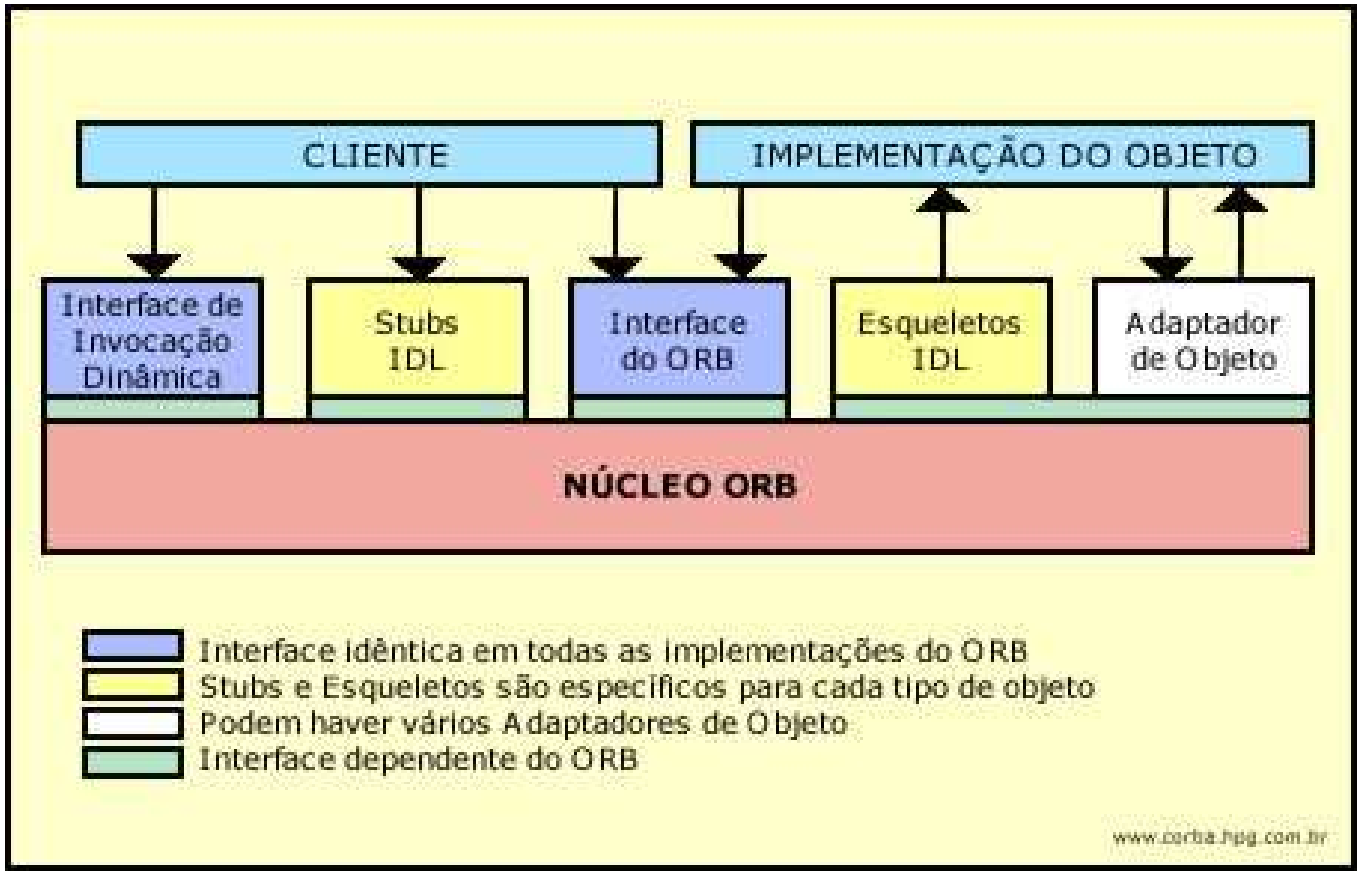
- O ORB é a estrutura mais importante da arquitetura CORBA.
  - Ele deve:
    - intermediar todas as transferências entre cliente e servidor;
    - fazer com que a transação seja transparente para cada uma das partes durante todo o processo;
    - Prover a localização do objeto alvo ao qual se destina a requisição;
    - envio dos parâmetros da requisição no formato aceito por este objeto;
    - Prover o retorno de parâmetros de saída da requisição para o cliente, se assim houver.

Arquitetura

- O ORB é a estrutura mais importante da arquitetura CORBA.
  - É responsabilidade do ORB a transferência de controle e de dados do cliente para o objeto, e em seguida, o retorno para o cliente.
  - No caso de uma invocação não se realizar perfeitamente o ORB informa ao cliente que uma exceção ocorreu.

Arquitetura

- Estrutura ORB



Arquitetura

- O cliente e a implementação do objeto estão separados por, pelo menos, três componentes
  - STUB IDL na ponta do cliente
  - Um ORB (ou vários)
  - SKELETON IDL na ponta do servidor.

  - Este isolamento dá uma grande flexibilidade e muitos benefícios.

Arquitetura

- Objetivo
  - Possibilitar que clientes invoquem métodos de objetos remotos em servidores CORBA
    - Ambos podem ser implementados em linguagens distintas
  - A linguagem de implementação do cliente não precisa ser necessariamente orientada a objetos

Arquitetura

- Funcionalidades
  - Invocações estáticas
    - Interface remota do objeto CORBA é conhecida em tempo de compilação (utilização dos stubs e skeletons)
    - Stubs
      - Interface de Invocação Estática (Static Invocation Interface - SII)
      - Para acessar uma operação remota, o cliente precisa estar estaticamente ligado ao stub correspondente.
      - Estão disponíveis na forma de bibliotecas e são invocados como procedimentos normais de uma linguagem de programação e gerados em tempo de compilação da descrição da interface do objeto.
      - Não pode acessar novos tipos de objetos que são adicionados ao sistema postriormente.

Arquitetura

- Funcionalidades
  - Invocações dinâmicas
    - Interface de invocação dinâmica – DII
    - Permite ao cliente invocar qualquer operação sobre qualquer objeto que ele possa acessar na rede.
    - Habilita um cliente, em tempo de execução, a descobrir novos objetos e fazer requisições a eles.
  - Passos:
    - 1. Identificar o objeto que queremos invocar (provavelmente através do Trader Service do CORBAservices);
    - 2. Recuperar sua interface (buscá-la no Repositório de Interfaces);
    - 3. Construir a invocação;
    - 4. Invocar a requisição, e receber os resultados.

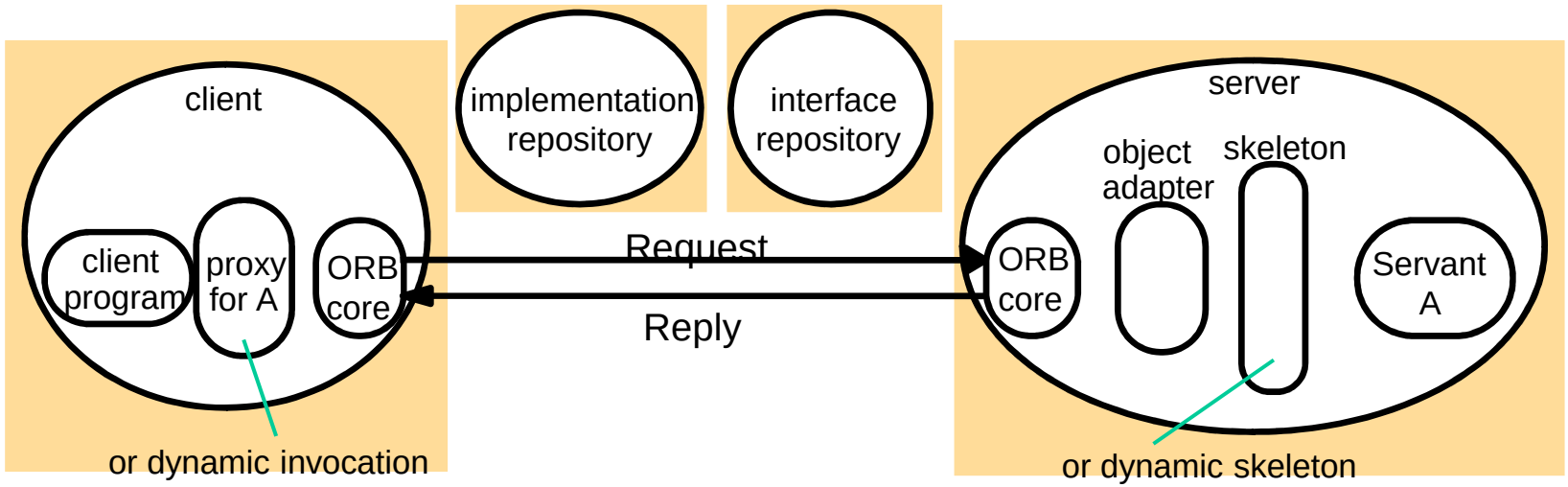
Arquitetura

- Funcionalidades
  - As implementações de objeto não podem detectar, quando recebem uma invocação, se esta foi feita ao ORB via SII ou via DII

Arquitetura

- Representação externa de dados
  - CDR (Common Data Representation)
- Semântica de invocação
  - Semântica at-most-once (default)
  - Semântica maybe (palavra-chave *oneway* na IDL)
    - Métodos sem resultados ou callbacks, o cliente não bloqueia na espera de uma resposta
- Objeto CORBA
  - Implementa uma interface - IDL
  - Possui uma referência de objeto remoto
  - Capaz de responder a invocações aos métodos de sua interface

Arquitetura

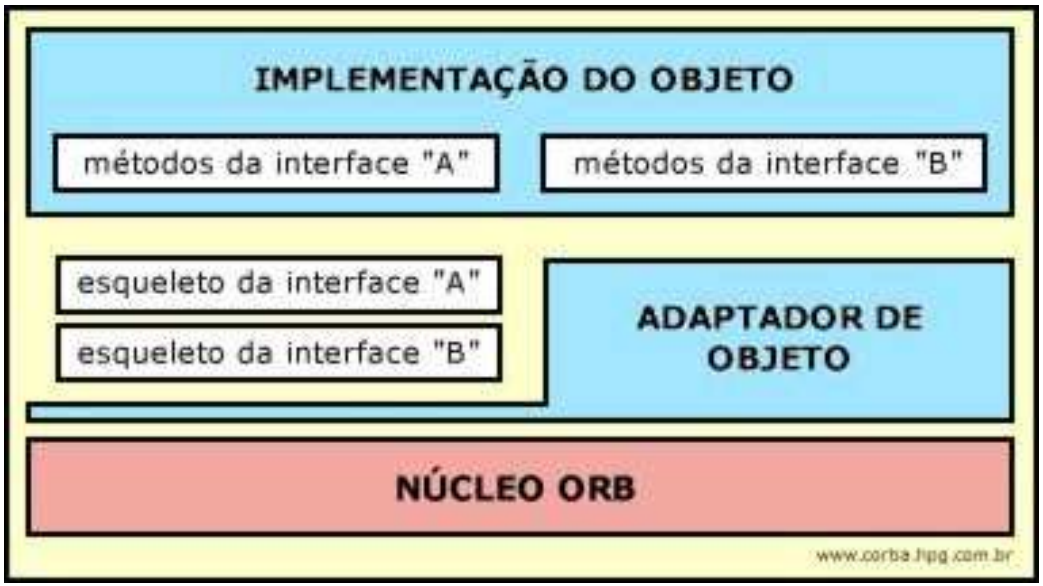


Arquitetura

- Adaptador de objeto (Object Adapter)
  - Interliga os objetos CORBA e as classes do servidor
  - Cria as referências a objetos remotos para os objetos CORBA
  - Despacha cada RMI através de um skeleton para o servidor apropriado
  - Ativa objetos (se necessário)
  - CORBA 2.2: POA (Portable Object Adapter)
    - Permite que aplicações e servidores sejam executados em ORBs produzidos por desenvolvedores diferentes

Arquitetura

- Adaptador de objeto (Object Adapter)
  - Exemplo com adaptador com função de ativar a implementação e autenticar a requisição recebida, verificando se cliente possui permissão para acessar o serviço requisitado.



Arquitetura

- Adaptador de objeto (Object Adapter)
  - É a interface através da qual uma implementação de objetos acessa as funções do ORB.
  - Cabe ao adaptador de objetos fazer a ativação e a desativação de implementações de objetos, manter o registro das implementações, promover a requisição de métodos e garantir a segurança da interação entre os elementos envolvidos na requisição da operação.

Arquitetura

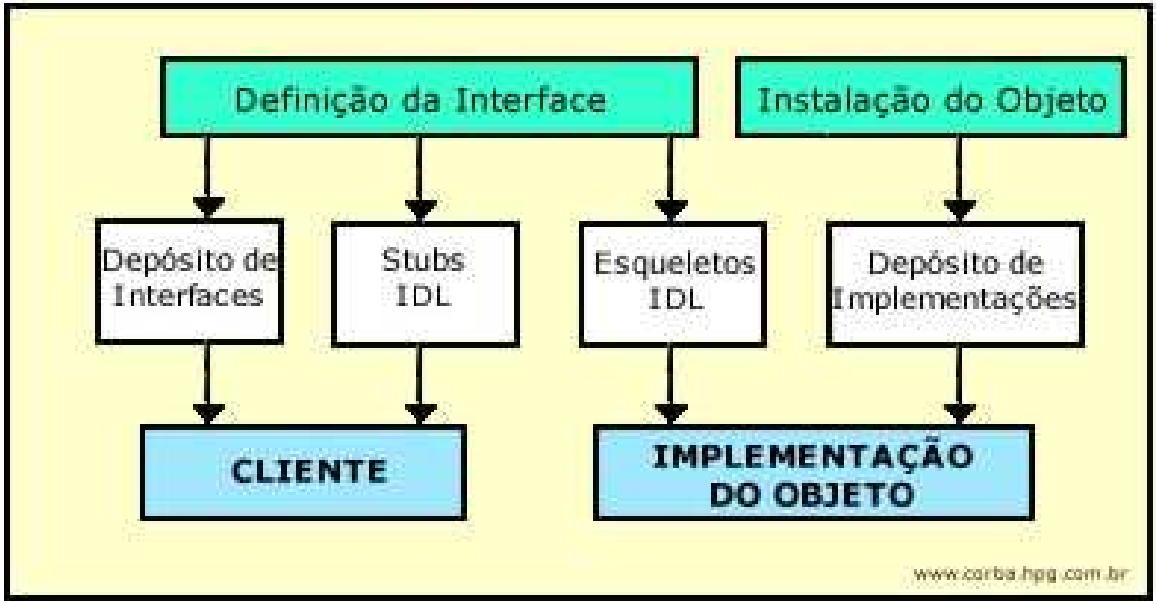
- Skeleton (servidor)
  - Gerada na linguagem do servidor
    - Compilador IDL
  - RMIs são despachadas através do skeleton apropriado a um servidor
  - marshaling e unmarshaling
- Stubs/proxies (cliente)
  - Geradas na linguagem do cliente
  - marshaling e unmarshaling

Arquitetura

- Repositório de implementação
  - Ativa por demanda os servidores registrados e localiza os servidores em execução
  - Armazena tabela com o mapeamento dos nomes dos adaptadores de objetos para suas implementações
  - Na ativação de objeto remoto, o *hostname* e o número da porta no servidor são adicionados na tabela
  - Possibilita armazenar outras informações sobre os servidores (e.g. controle de acesso)
  - Permite replicação
    - aumento de disponibilidade e tolerância a falhas

Arquitetura

- Repositório de interface
  - A definição da interface ocorre no repositório de interfaces e/ou na linguagem IDL
  - A instalação do objeto fornece informações referentes a sua implementação, que fica armazenada no repositório de implementações.



Arquitetura

- Repositório de interface
  - Provê informações sobre interfaces IDL registradas (e.g. métodos, argumentos, exceções)
  - Cliente sem proxy de objeto pode obter informações necessárias (métodos e argumentos)
  - Necessário para invocações dinâmicas
  - Nem todos os ORBs provêem um repositório de interfaces

  - O repositório de interfaces é o meio através do qual o ORB permite o acesso distribuído às implementações de objetos, colocando à disposição dos elementos da arquitetura as interfaces públicas dos objetos especificadas na linguagem IDL.

Arquitetura

- Interface de invocação dinâmica
  - Permite que clientes façam invocações dinâmicas a objetos CORBA desconhecidos
  - Cliente obtém informações necessárias sobre um objeto CORBA a partir do repositório de interfaces e as utiliza para construir uma invocação e enviá-la ao servidor
  - CORBA não permite o download de uma classe durante a execução como em Java RMI

Arquitetura

- Interface skeleton dinâmica
  - Permite um objeto CORBA aceitar invocações em uma interface sem skeleton
    - Interface não era conhecida em tempo de compilação
  - Skeleton dinâmico
    - Recebe a invocação
    - Inspeciona o conteúdo da requisição para descobrir o objeto destino, o método para ser invocado e os argumentos
    - Invoca o destino

CORBA IDL

- Facilidades para definir módulos, interfaces, tipos, atributos e métodos
- Sintaxe similar a C++ incluindo algumas palavras-chaves

- Módulos IDL
  - Permite agrupar interfaces e outros tipos IDL em unidades lógicas
  - Módulo define um nome de escopo (evita conflito entre nomes)
- Interfaces IDL
  - descrevem os métodos e atributos que são disponíveis por objetos CORBA que implementam a interface

CORBA IDL

- Métodos IDL

[oneway] <return_type> <method_name> (parameter1, ..., parameterL)

[raises (except1, ..., exceptN)]

  - parâmetros: <in | out | inout> <type> <parameter_name>
  - oneway: semântica maybe
  - raises: sinaliza que pode causar exceção

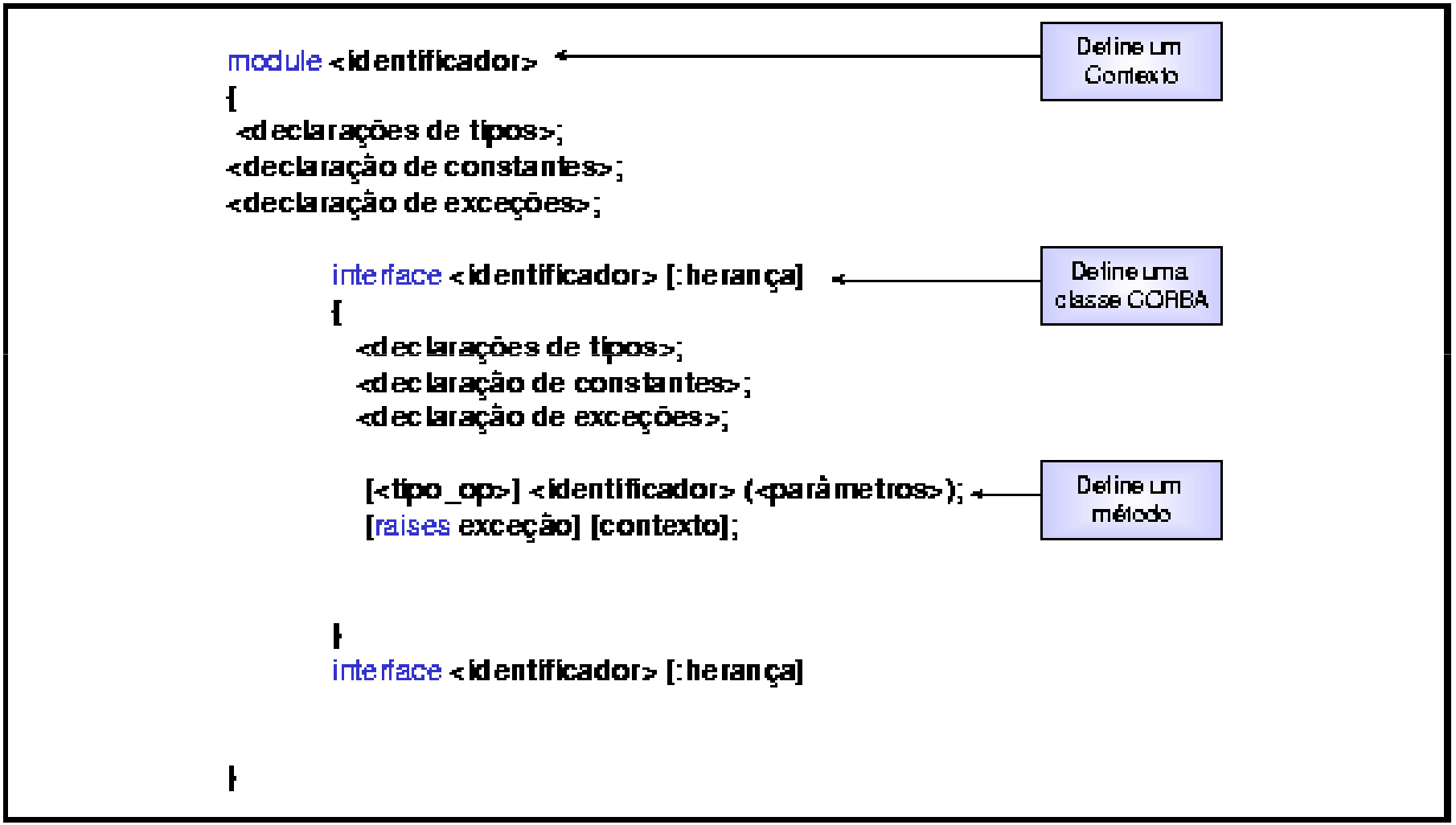
- Tipos IDL
  - 15 tipos primitivos: *short, long, unsigned short, unsigned long, float, double, char, boolean, octet, any*
  - tipos derivados: *sequence, string, array, record, enumerated, union*
  - tipo Object: objeto CORBA

Exemplo - CORBA IDL

```
exception DivisionByZero{};

interface calc
{
  long soma(in long a, in long b);
  long subtrai(in long a, in long b);
  long multiplica(in long a, in long b);
  long divide(in long a, in long b) raises (DivisionByZero);
};
```

Exemplo - CORBA IDL



Referência de Objeto Remoto

- IORs – Interoperable Object References (CORBA 2)
  - formato de referência a objetos remotos

IDL interface type name	Protocol and address details			Object key	
interface repository identifier	IIOP	host domain name	port number	adapter name	object name

  - **IDL interface type name**: nome da interface remota, igual ao identificador da interface no repositório de interfaces
  - **Protocol and address details**: protocolo (IIOP – Internet Inter-ORB protocol – usa TCP/IP), endereço e porta
  - **Object key**: identificação do objeto CORBA – nome do adaptador e nome do objeto

Referência de Objeto Remoto

- A criação de um objeto implica em:
  1. Registro de sua identidade junto ao ORB;
  2. ORB poder localizar uma implementação para o referido objeto;
  3. ORB poder enviar a esta implementação requisições de operações.

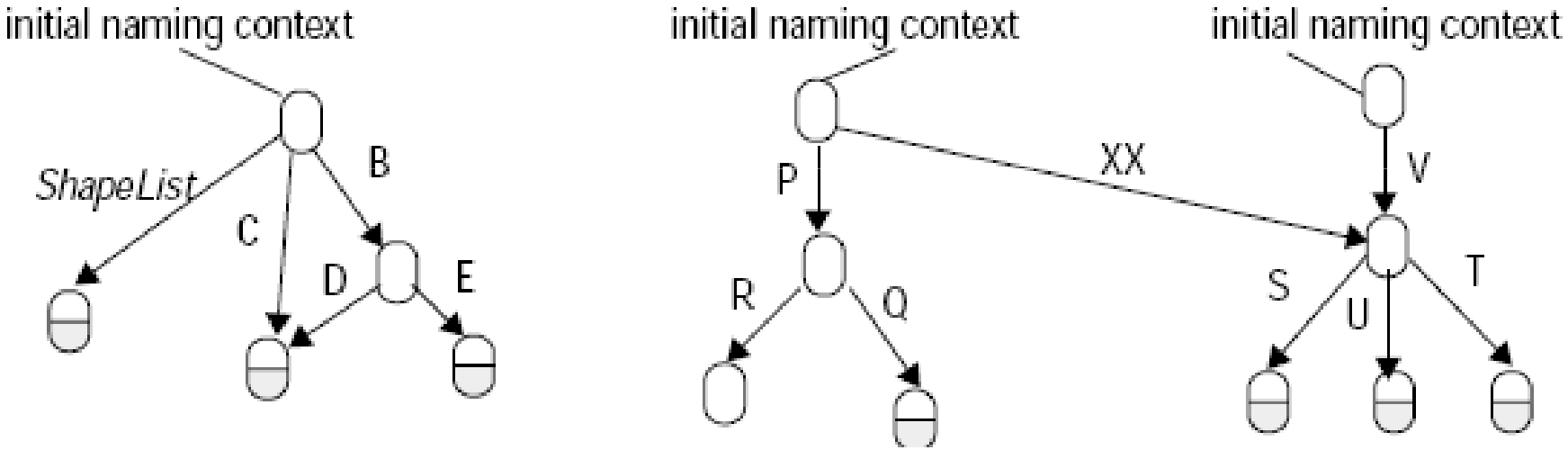
- Um objeto informa que implementa as operações descritas em uma determinada interface.

Serviços CORBA

- Serviço de nomes
  - mapeamento de nomes a referências a objetos remotos de objetos CORBA
  - provê operações para:
    - registrar referências de objetos remotos de objetos CORBA por nome
    - resolver referências de objetos remotos para clientes através do nome
  - definição de contextos: permite estrutura hierárquica

Serviços CORBA

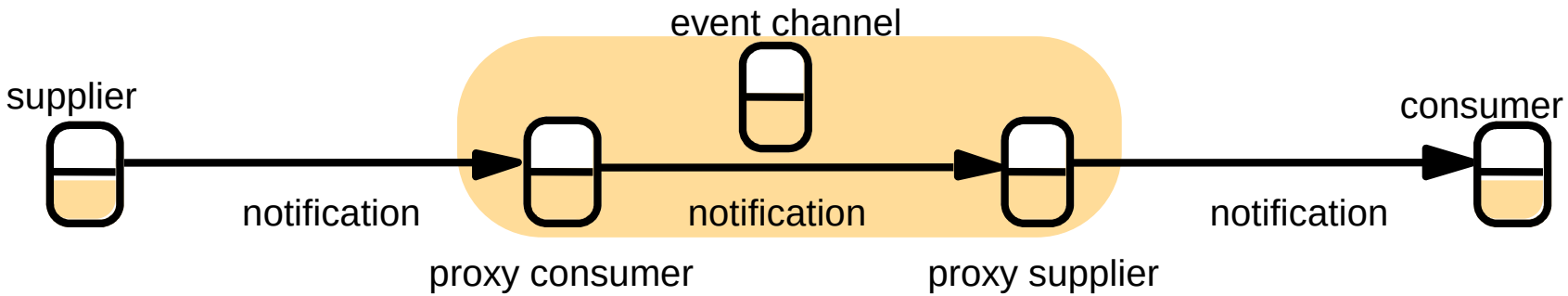
- Serviço de nomes



Serviços CORBA

- Serviço de eventos
  - define interfaces permitindo objetos de interesse, chamados **fornecedores**, comunicarem notificações a **consumidores**.
  - Propagação de notificações
    - push pelo fornecedor
    - pull pelo consumidor
  - **Canais de eventos**: objetos CORBA que são utilizados para permitir diversos fornecedores se comunicarem com diversos consumidores de forma assíncrona
    - atua como um buffer entre fornecedores e consumidores
    - pode usar multicast

Serviços CORBA



Serviços CORBA

- Serviço de notificação
  - extensão do serviço de eventos com novas funcionalidades
  - utilização de filtros pelos consumidores de eventos definindo exatamente os eventos de interesse
  - publicação dos eventos desejados pelos consumidores
    - fornecedores podem gerar somente os eventos desejados pelos consumidores
  - publicação dos eventos oferecidos pelos fornecedores
    - possiblita a "assinatura" a eventos assim que se tornam disponíveis
  - configuração de um canal ou evento particular
    - confiança de entrega de eventos
    - prioridade dos eventos

Serviços CORBA

- Serviço de segurança
  - autenticação dos usuários e servidores, geração de credenciais para os usuários e servidores (certificados garantindo seus direitos)
  - controle de acesso a objetos CORBA

Serviços CORBA

- Serviço de *trading*
  - permite que objetos sejam localizados a partir de atributos
  - possui uma base de dados com mapeamento dos tipos de serviço e atributos para referências a objetos remotos
  - clientes podem realizar consultas
    - tipo de serviço e atributos
    - preferências de ordem de recepção de ofertas que mapeiam as características desejadas

Serviços CORBA

- Serviço de transações
  - permite que objetos CORBA participem de transações
  - o cliente especifica uma transação como uma seqüência de chamadas RMI, iniciadas por begin e terminadas por commit ou rollback (abort)
  - ORB anexa um identificador de transações para cada invocação remota e lida com requisições de begin, commit e rollback
  - clientes podem suspender ou resumir transações
- Serviço de controle de concorrência
  - utilização de locks para aplicar controle de concorrência aos acesso realizados ao objeto CORBA
  - utilizado com transações ou de forma independente