

Serviço de Nomes CORBA

Engenharia de Computação -
IFCE

Cidcley Teixeira de
Souza

cidcley@ifce.edu.br

Serviço de Nomes CORBA

- ▶ Páginas Brancas
 - ▶ Permite encontrar objetos através de nomes;
 - ▶ Nomes -> Referência de Objeto
 - ▶ Essa associação é denominada **name binding**;
 - ▶ Um **name context** é o espaço onde o nome do objeto é único;
 - ▶ Nomes são relativos a **name contexts**

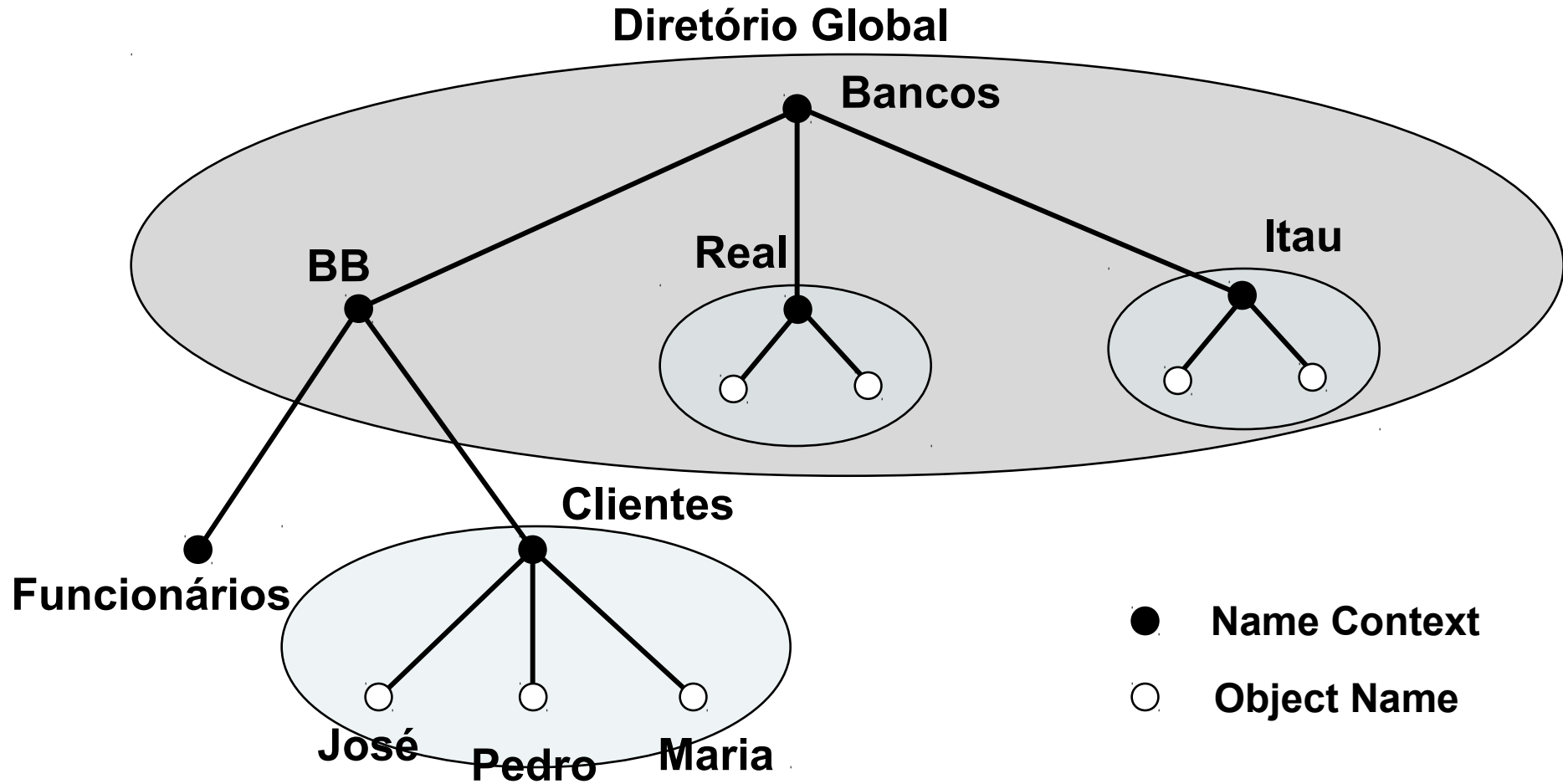
Serviço de Nomes CORBA

- ▶ Object Name Service
 - ▶ Padrão OMG desde 1993;
 - ▶ Encapsula os conceitos de serviços conhecidos, como X.500 e DNS;
 - ▶ A ideia é permitir a criação de hierarquias de nomes;
 - ▶ Clientes podem navegar em contextos de nomes procurando objetos;

Serviço de Nomes CORBA

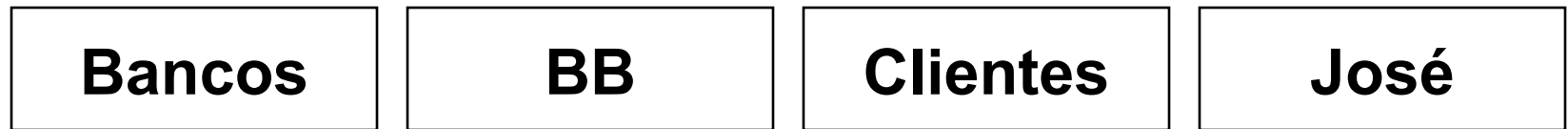
- ▶ CORBA Object Name
 - ▶ Uma sequência de nomes que formam uma hierarquia; (*Compound Name*)
 - ▶ Cada elemento é um nome de contexto, exceto o último que é o nome simples do objeto;
 - ▶ Os contextos são manipulados através da interface *Naming Context*;

Serviço de Nomes CORBA



Serviço de Nomes CORBA

- ▶ CORBA Object Name
 - ▶ Um *Compound Name* define o caminho a se seguir para se “resolver” nomes;

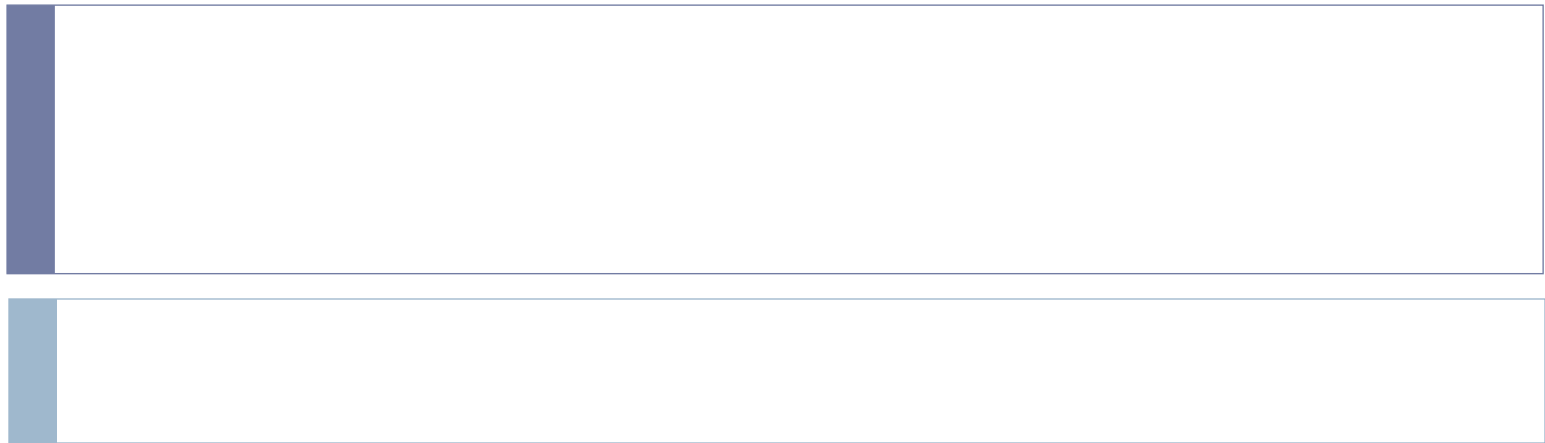


- ▶ Realizar um *bind* é criar uma associação nome/objeto em um determinado contexto;
- ▶ Os nomes são manipulados através de uma Struct IDL denominada *Name Component*;

Serviço de Nomes CORBA

- ▶ CORBA Object Name
 - ▶ Cada componente do nome é uma estrutura com dois atributos:
 - ▶ Identificador - String com o nome do Objeto;
 - ▶ Tipo - String que define o tipo do componente (utilizado informalmente pelas aplicações);

Funcionamento do Serviço de Nomes



Funcionamento do Serviço de Nomes

- ▶ Interface NamingContext
 - ▶ bind, rebind
 - ▶ Associa um objeto a um nome dentro de um contexto;
 - ▶ unbind
 - ▶ Remove um objeto de um contexto;
 - ▶ new_context
 - ▶ Retorna um contexto;

Funcionamento do Serviço de Nomes

- ▶ Interface NamingContext
 - ▶ bind_new_context
 - ▶ Cria um contexto e associa com um nome fornecido;
 - ▶ destroy
 - ▶ Apaga um contexto de nome;
 - ▶ resolve
 - ▶ Recupera um objeto através de um nome em um determinado contexto;

Funcionamento do Serviço de Nomes

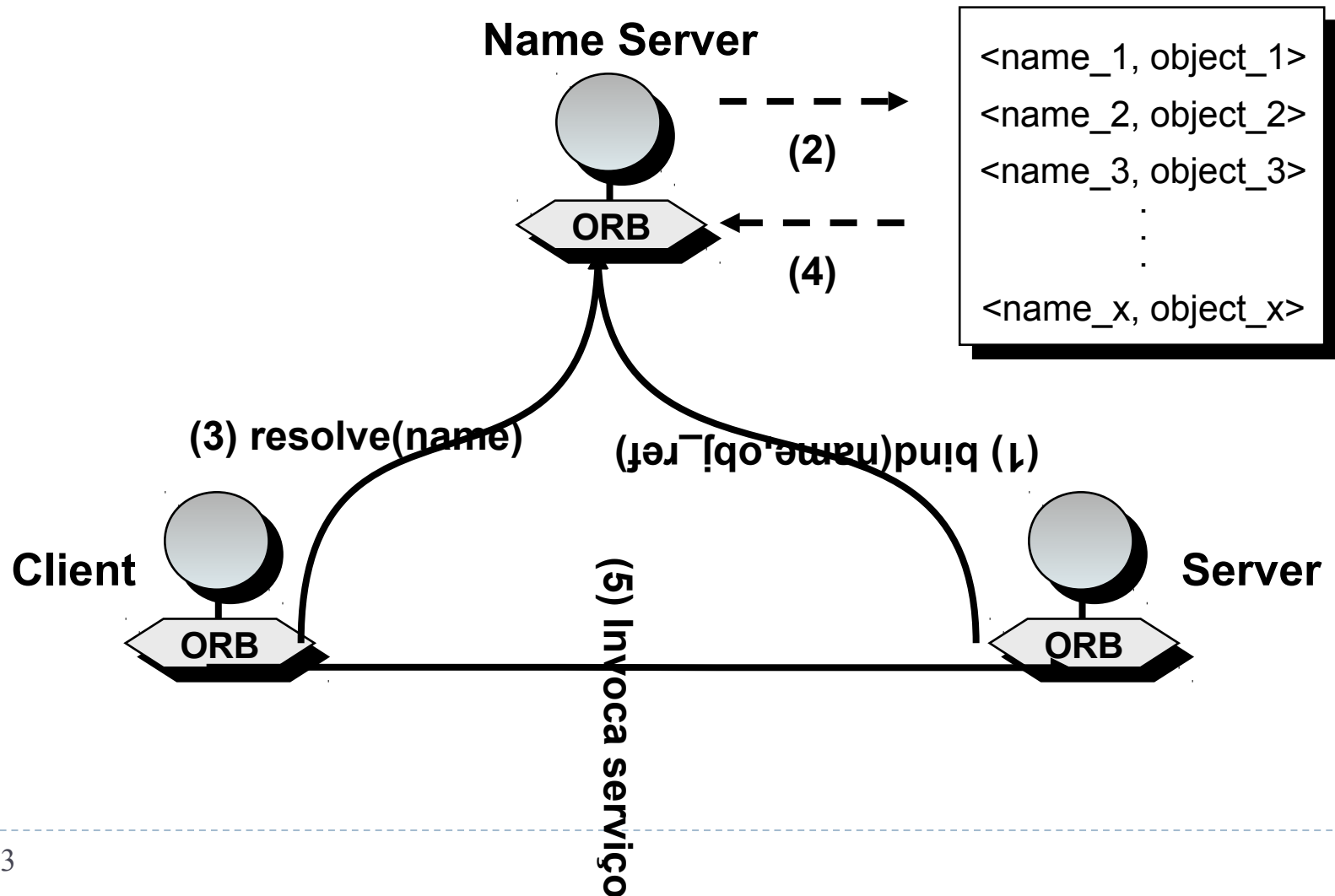
► Estrutura de Dados NameComponent

```
struct NameComponent{  
    Istring id;  
    Istring kind;  
};
```

Funcionamento do Serviço de Nomes

- ▶ Pacote Java
 - ▶ O pacote (package) java que implementa o serviço de nomes é o CosNaming;
 - ▶ Todos os ORBs java têm que utilizar esse package como interface para o serviço de nomes; (org.omg.CosNaming)

Funcionamento do Serviço de Nomes

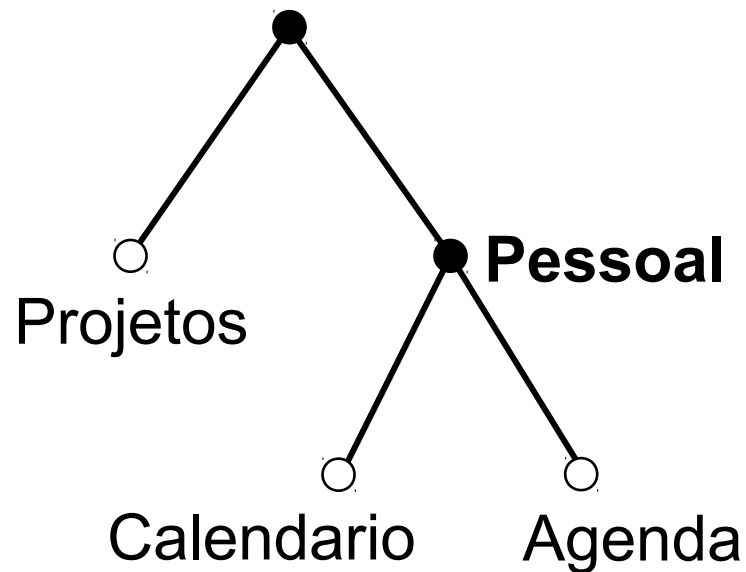


Cenários

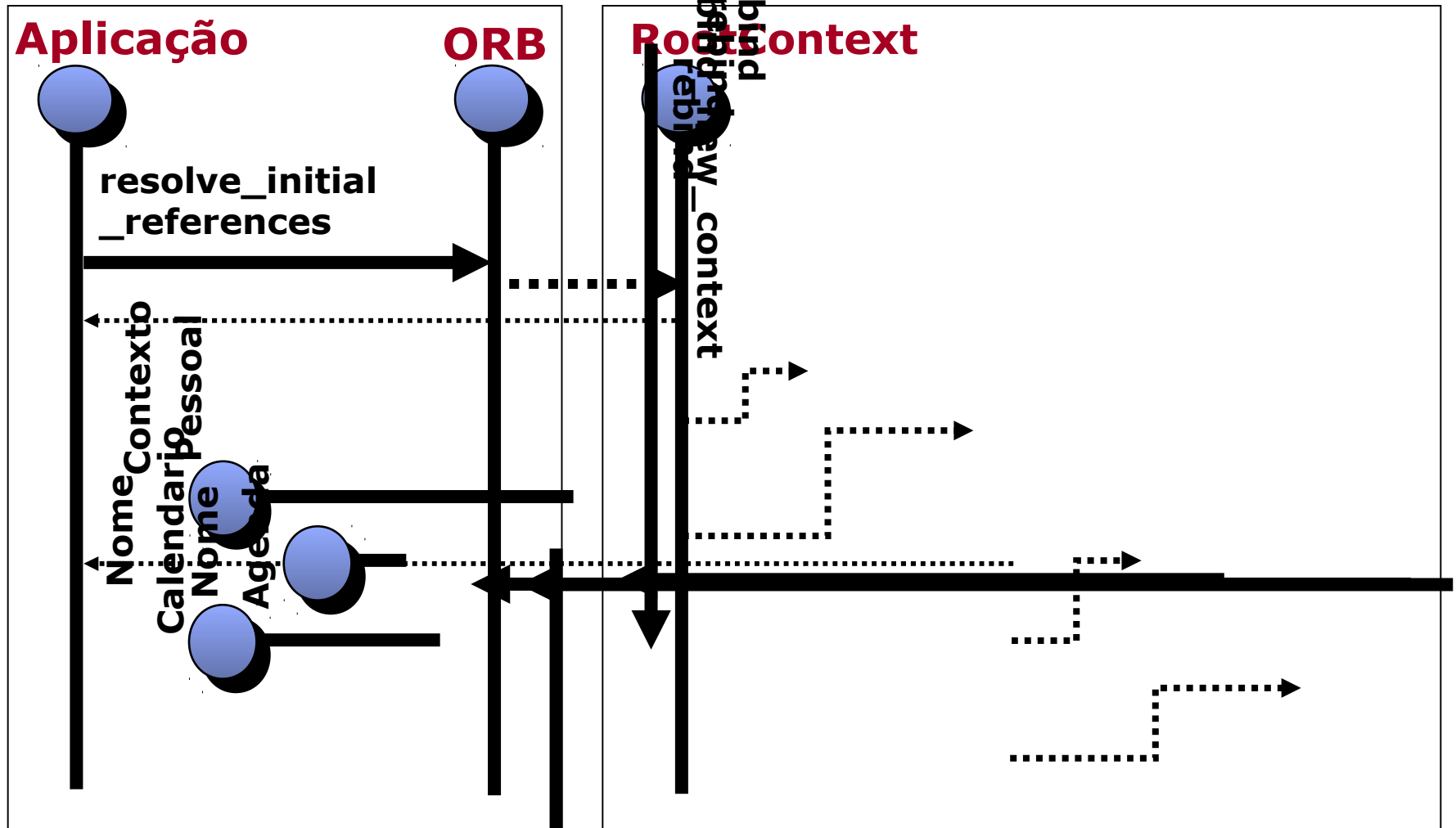
- Interação Cliente/Servidor
 - O Servidor invoca (bind/rebind) para associar um nome a uma referência de objeto;
 - O Servidor de Nomes adiciona essa referência/nome ao seu banco de dados (namespace);
 - A aplicação cliente invoca resolve para obter a referência do objeto;
 - O cliente utiliza a referência para invocar o métodos no objeto servidor;

Cenários - Criando Namespace

Contexto de Nomes Inicial (root)



Cenários - Criando Namespace



Cenários - Criando Namespace

- ▶ Obter o contexto de nomes inicial
 - ▶ Obtido por broadcast no momento da inicialização do servidor. Através do comando `resolve_initial_references()`;

```
org.omg.CORBA.Object objRef=  
orb.resolve_initial_references("NameService");  
  
NamingContext rootContext =  
    NamingContextHelper.narrow(objRef);
```

Cenários - Criando Namespace

- ▶ Criar o nome chamado "Projetos"
 - ▶ Inicialmente deve-se criar um NameComponent para Projetos;
 - ▶ Deve-se utilizar o construtor que recebe duas strings (id, kind)
 - ▶ `bind_new_context` deve ser invocado para se criar um novo NameContext
 - ▶ Será usado o tipo "Text" para esse nome;

Cenários - Criando Namespace

- ▶ Criar o nome chamado "Projetos"

```
NameComponent    comp1 = new  
    NameComponent("Projetos","Text");  
  
NameComponent[]  name1 = {comp1};  
  
rootContext.rebind(name1,objref);
```

Cenários - Criando Namespace

- ▶ Criar contexto chamado “Pessoal”
 - ▶ Criar um contexto chamado “Pessoal” relativo ao rootContext;
 - ▶ Inicialmente deve-se criar o nome “Pessoal” e conecta-lo ao contexto rootContext;
 - ▶ Será utilizado o tipo “diretorio” para esse contexto;

Cenários - Criando Namespace

- ▶ Criar contexto chamado "Pessoal"

```
NameComponent    comp2 = new  
    NameComponent("Pessoal", "diretorio");
```

```
NameComponent[] name2 = {comp2};
```

```
NamingContext PessoalContext =  
    rootContext.bind_new_context(name2);
```

Cenários - Criando Namespace

- ▶ Criar os nomes "Calendario" e "Agenda" e associa-los ao contexto "Pessoal"
- ▶ As referências dos objetos devem ser criadas inicialmente. No exemplo serão usados os valores objref3 e objref4 para representar essas referências (geradas pelo POA a partir dos Servants);
- ▶ Os nomes devem ser criados e associados às referências e depois conectados ao contexto devido;

Cenários - Criando Namespace

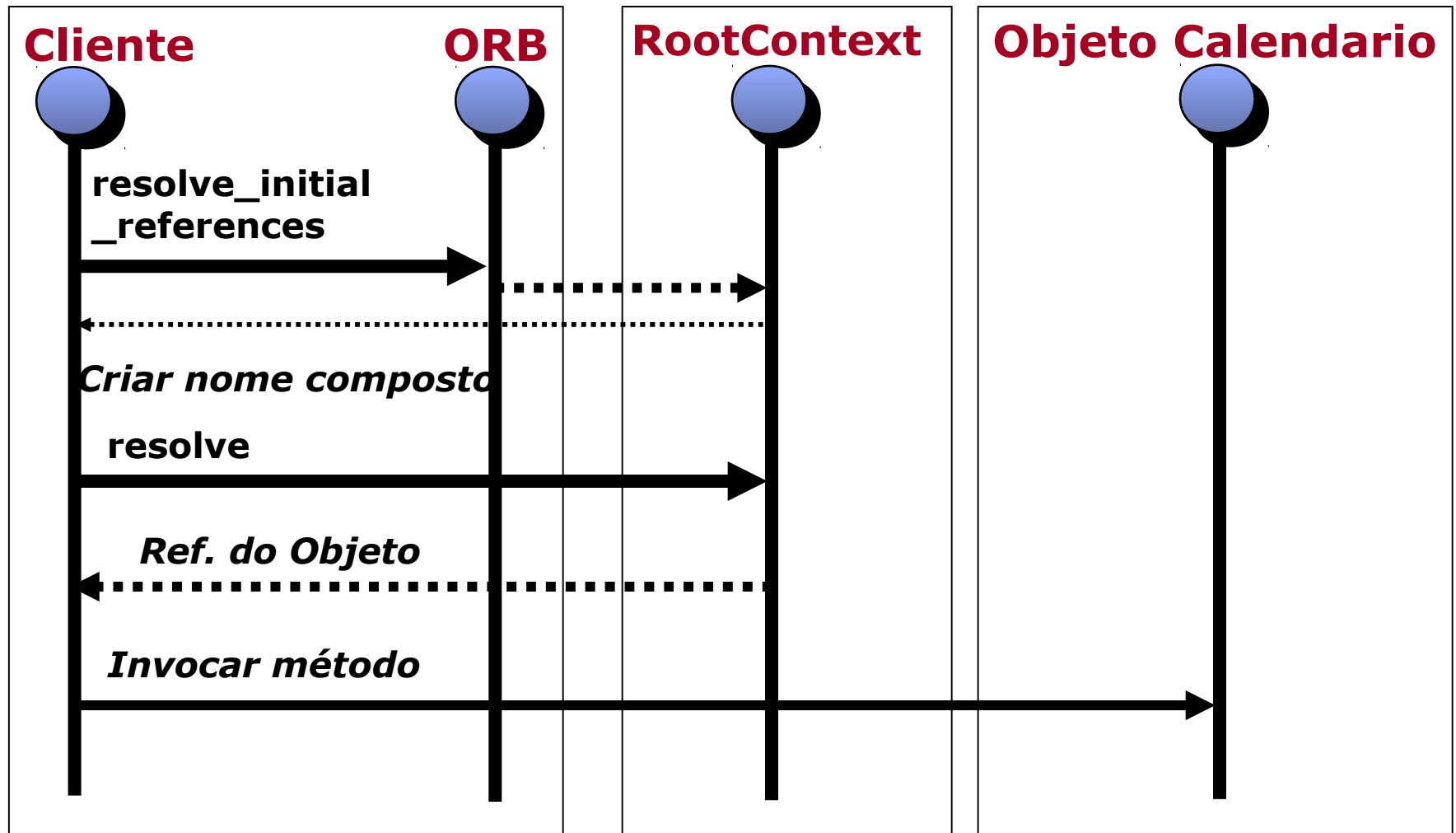
- Criar os nomes "Calendario" e "Agenda" e associa-los ao contexto "Pessoal"

```
NameComponent    comp3 = new  
    NameComponent("Calendario","text");  
NameComponent[]  name3 = {comp3};
```

```
NameComponent    comp4 = new  
    NameComponent("Agenda","text");  
NameComponent[]  name4 = {comp4};
```

```
PessoalContext.rebind(name3,objref3);  
PessoalContext.rebind(name4,objref4);
```

Cenários - Localizando Objetos



Cenários - Localizando Objetos

- ▶ Obter o contexto de nomes inicial
 - ▶ Obtido por broadcast no momento da inicialização do servidor. Através do comando `resolve_initial_references()`;

```
org.omg.CORBA.Object objRef =  
orb.resolve_initial_references("NameService");
```

```
NamingContext rootContext =  
    NamingContextHelper.narrow(objRef);
```

Cenários - Localizando Objetos

- Construir o nome composto a ser procurado.

```
NameComponent comp1 = new  
    NameComponent("Pessoal","diretorio");
```

```
NameComponent comp2 = new  
    NameComponent("Calendario","text");
```

```
NameComponent[] name = {comp1, comp2};
```

Cenários - Localizando Objetos

- ▶ Encontrar o Objeto com nome especificado

```
org.omg.CORBA.Object CalendarioRef =  
    rootContext.resolve(name) ;
```

Cenários - Localizando Objetos

- ▶ Invocar o método desejado
 - ▶ O método do objeto deve ser invocado como se este fosse local;

```
Calendario CalendarioObj =  
    CalendarioHelper.narrow(CalendarioRef) ;
```

Fim