



Objetos Distribuídos em Java

Sérgio Teixeira de Carvalho
sergiocarvalho@anhanguera.edu.br

Objetos Distribuídos em Java

Conceituação

Tecnologias

Sistema Distribuído

?

Sistema Distribuído

Definição 1:

Coleção de processadores fracamente acoplados e interconectados por uma rede de comunicação.

Sistema Distribuído

Definição 2:

Coleção de computadores autônomos interligados em uma rede e equipados com software de sistema distribuído.

Sistema Distribuído

Software de Sistema Distribuído:

O software permite a coordenação das atividades dos computadores e o compartilhamento de recursos do sistema – hardware, software e outros. Os usuários têm a percepção de um sistema único e com facilidades integradas, mesmo se implementadas por muitos computadores.

Objetos Distribuidos

?

Sistema Distribuído

X

Sistema Centralizado

Sistema Distribuído

!

Sistema Centralizado

Sistema Distribuído



!

Sistema Centralizado

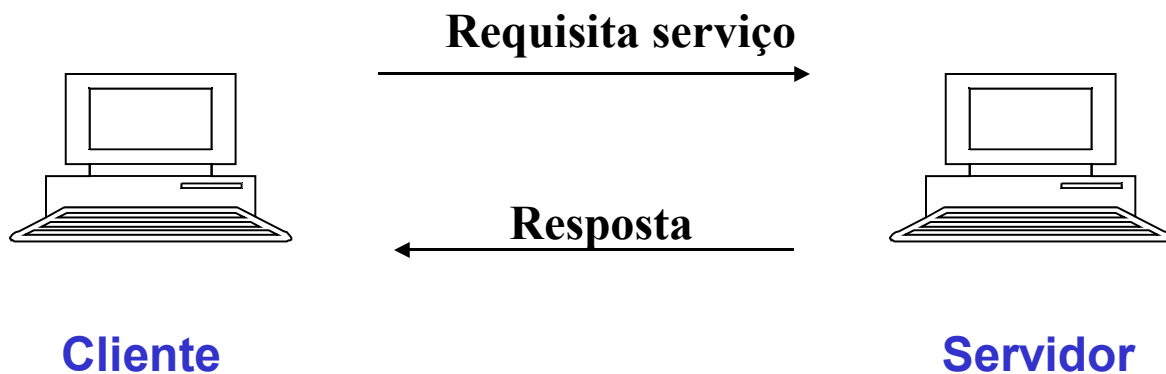


Comunicação entre Processos (IPC)

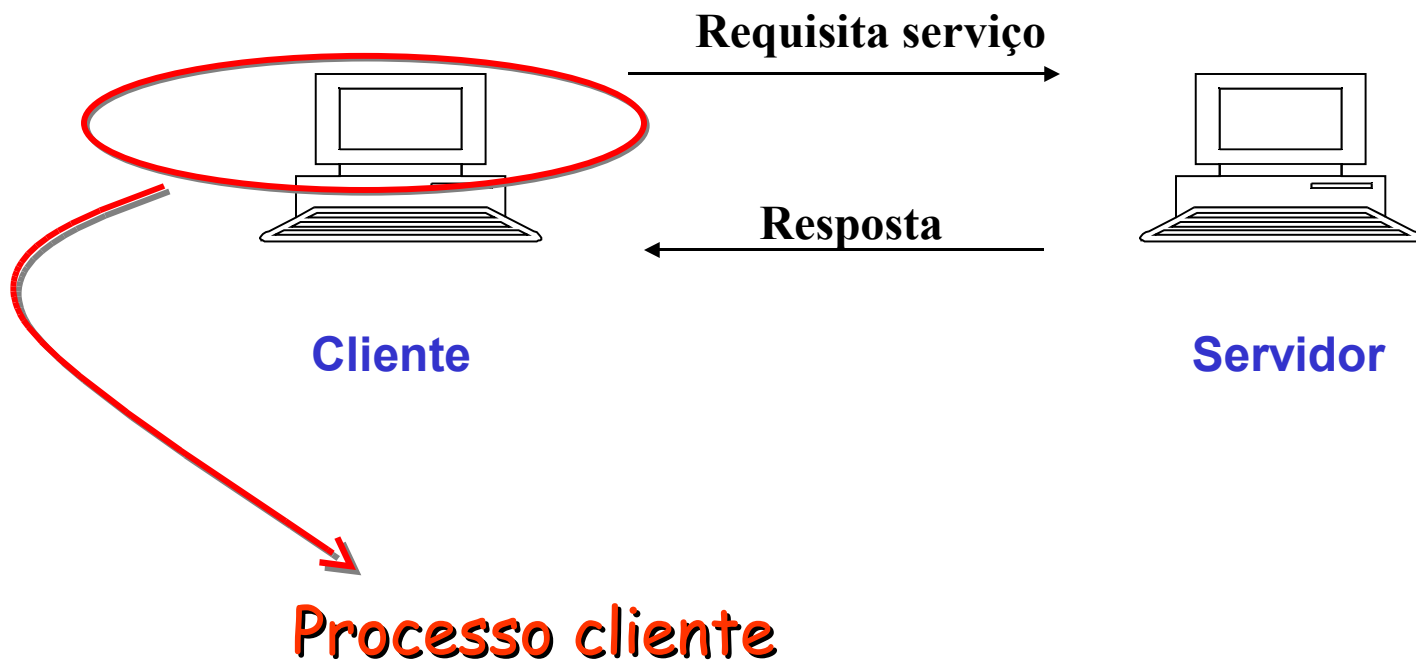
Comunicação Entre Processos (IPC)

- Comunicação Cliente-Servidor
- Chamada Remota a Procedimento (RPC)

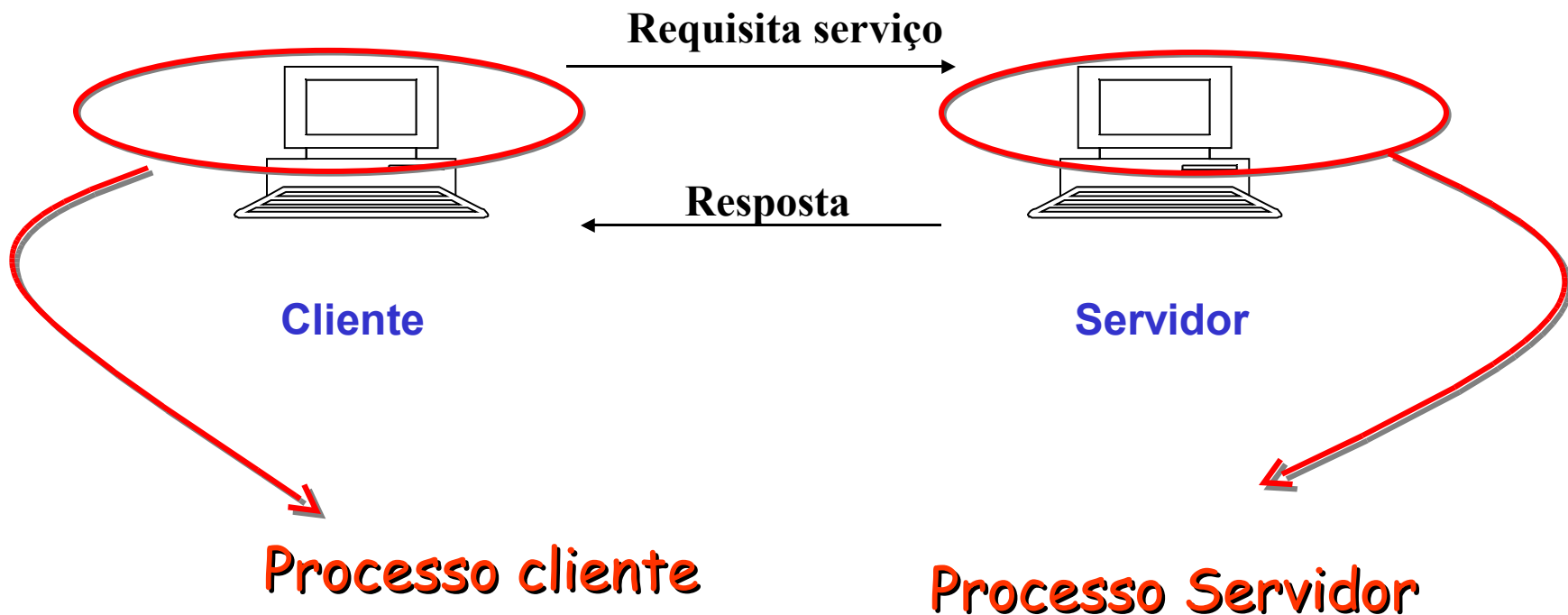
Comunicação Cliente/Servidor



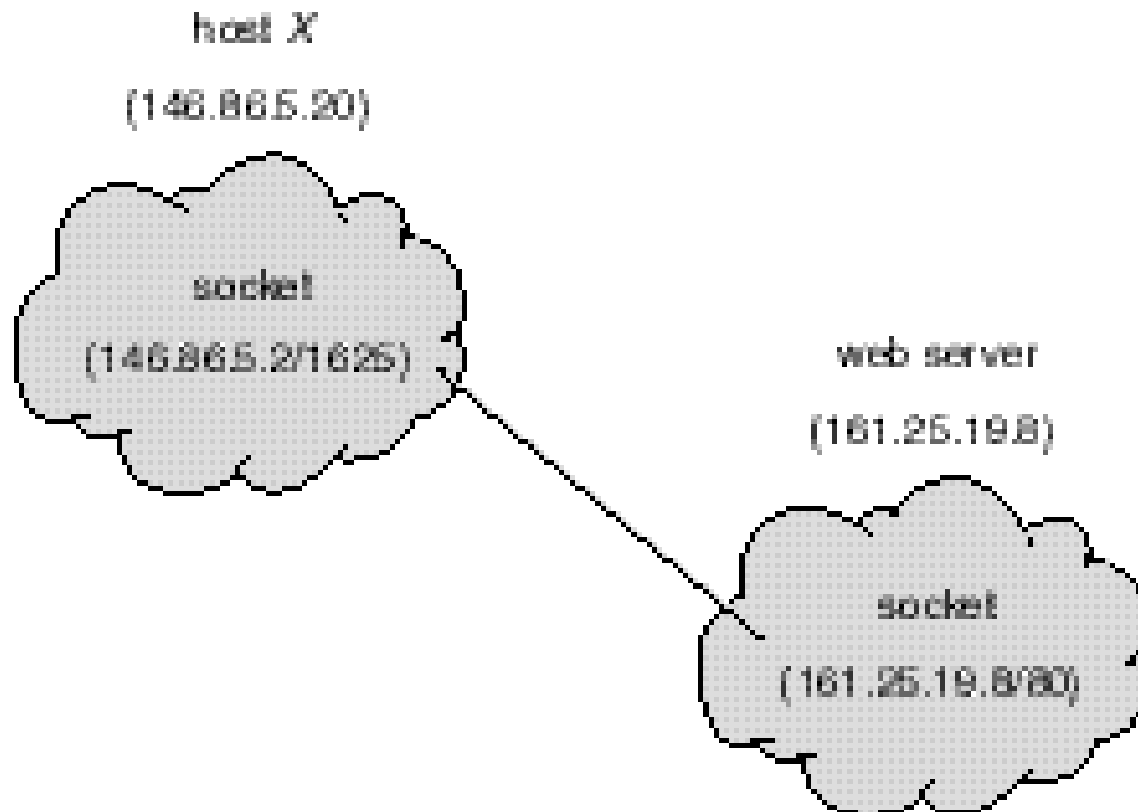
Comunicação Cliente/Servidor



Comunicação Cliente/Servidor



Comunicação com Sockets



Sockets em Java

- Servidor usa

```
s = new ServerSocket(5155)
```

para criar o socket na porta 5155

- Para aceitar conexões de clientes:

```
Socket cliente = s.accept()
```

- Conexões são normalmente servidas em threads separadas

Sockets em Java

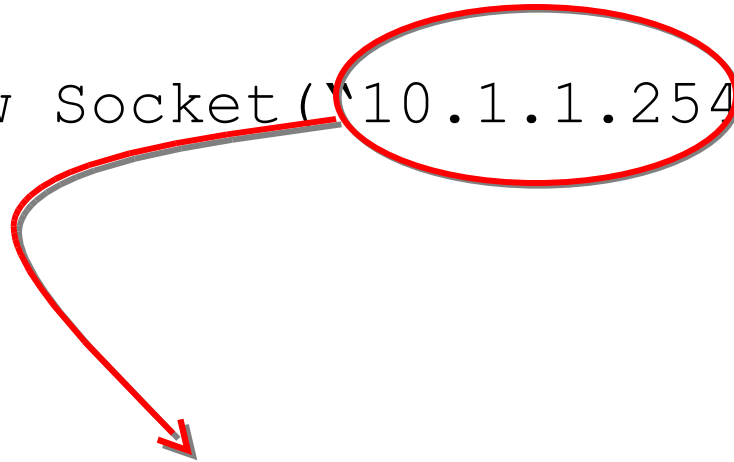
- O Cliente se conecta ao Servidor usando:

```
Socket s = new Socket("127.0.0.1", 5155);
```

Sockets em Java

- O Cliente se conecta ao Servidor usando:

```
Socket s = new Socket("10.1.1.254", 5155);
```

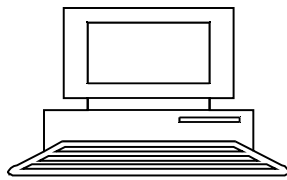


Endereço IP do Servidor

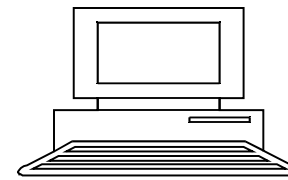
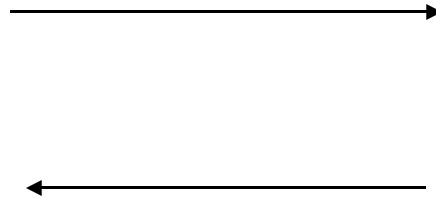
Chamada Remota de Procedimento (RPC)

- Sockets são considerados de baixo nível
- RPC oferece um nível de comunicação mais alto
- Cliente faz uma chamada a um servidor, usando mecanismos de chamada a procedimentos

Chamada Remota de Procedimento (RPC)



Cliente



Servidor

```
...  
m();  
...
```

```
void m() {  
    ...  
}
```

Chamada Remota de Procedimento (RPC)

Características:

- procedimentos com parâmetros
- procedimento remoto é executado em ambiente diferente do chamador

Chamada Remota de Procedimento (RPC)

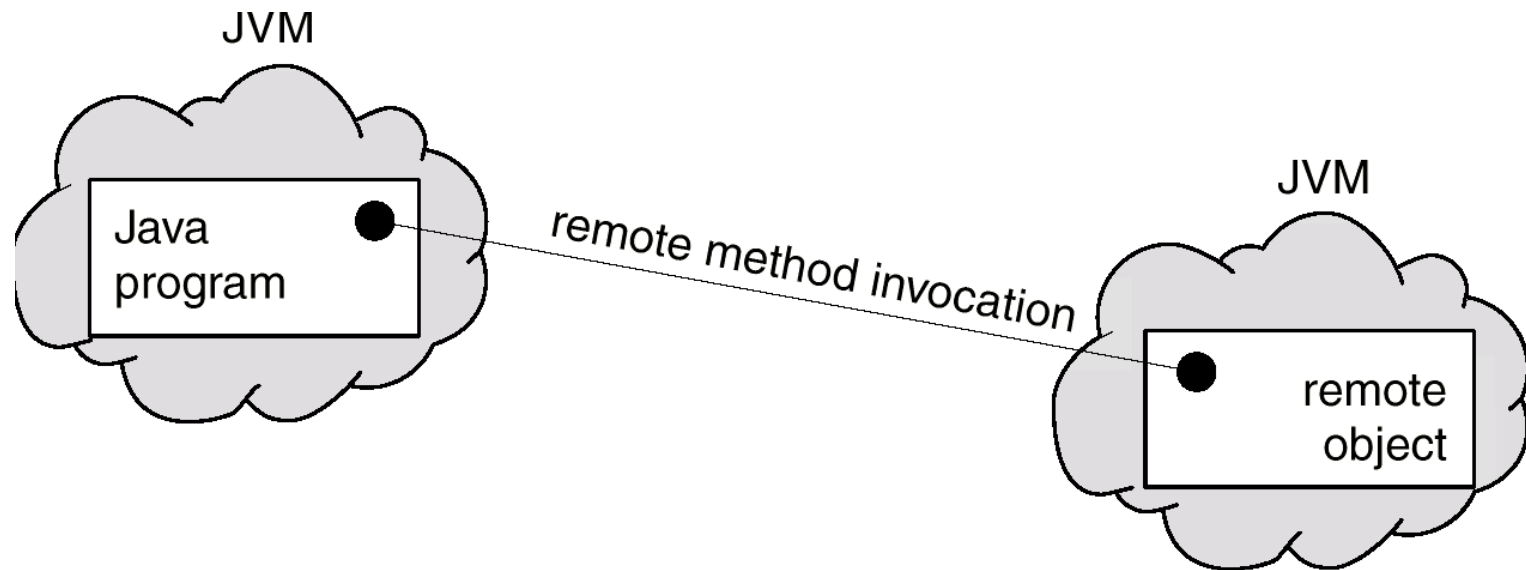
Suporte de um software de RPC:

- Processamento da Interface
- Tratamento da Comunicação
- Ligação (*bind*)

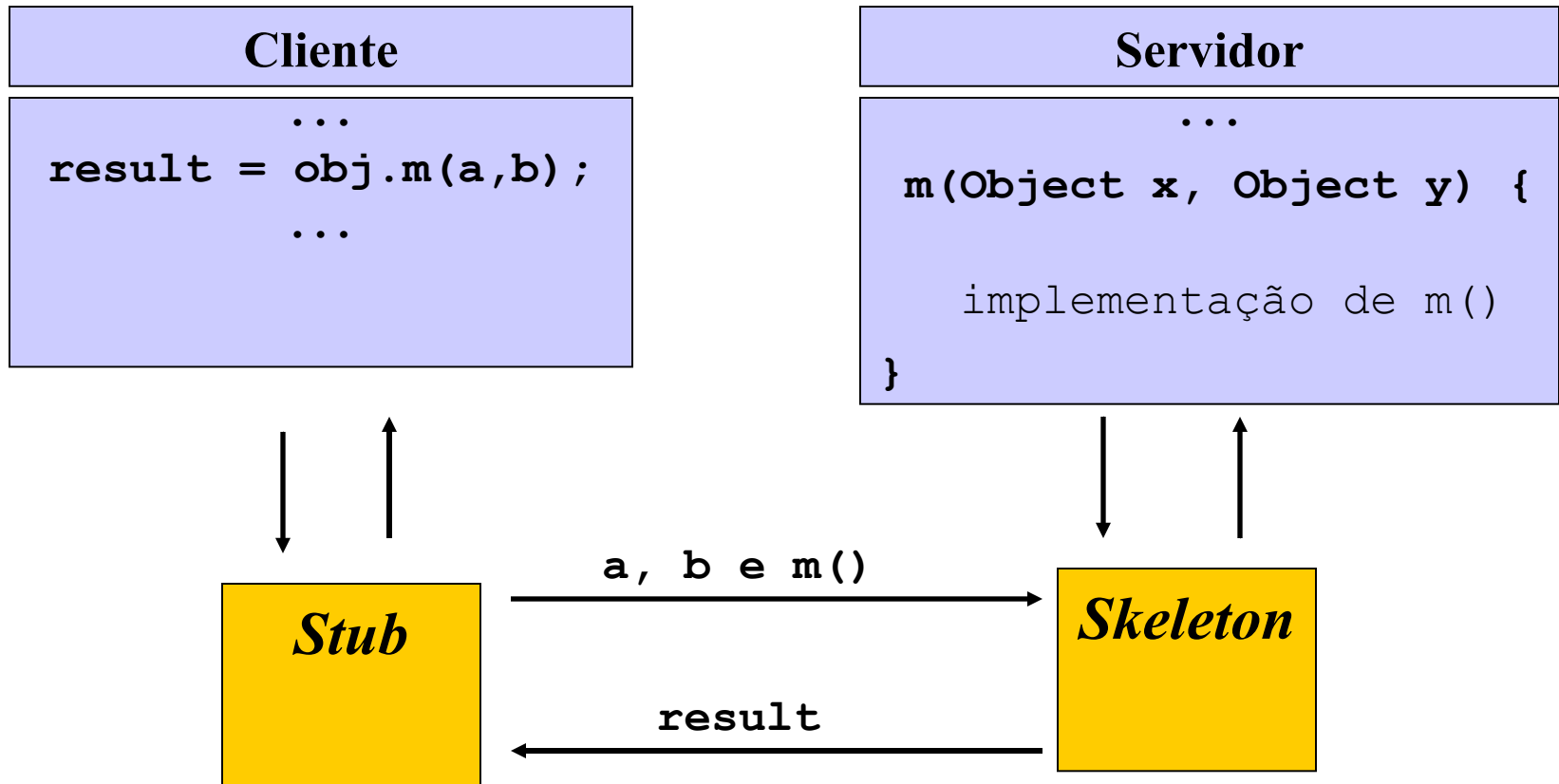
Chamada Remota de Método (RMI)

- Versão Java do RPC
- Um objeto é considerado remoto se reside em uma JVM remota

Chamada Remota de Método (RMI)



RMI



compilador RMI (*rmic*) gera *stub* e *skeleton*

RMI x RPC

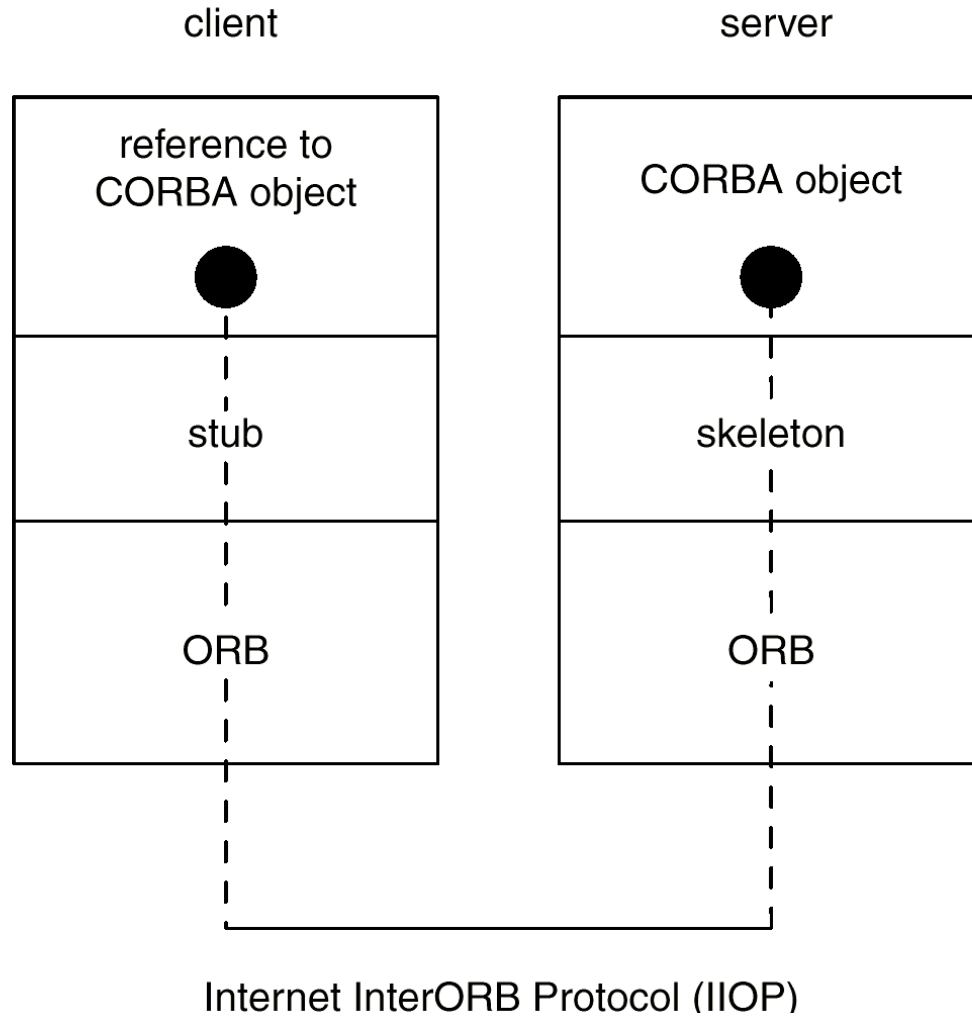
- RPC suporta estilo de programação procedural
- RMI suporta programação orientada a objetos
- Parâmetros no RPC são estruturas de dados
- Parâmetros no RMI são objetos

CORBA

(Common Object Request Broker Architecture)

- Middleware que permite a comunicação entre Clientes e Servidores heterogêneos
- IDL (Interface Definition Language) permite a descrição genérica da Interface de um serviço que um objeto remoto fornece
- ORB permite a comunicação entre Clientes e Servidores através da IDL
- IIOP (Internet InterORB Protocol) especifica como os ORBs podem se comunicar

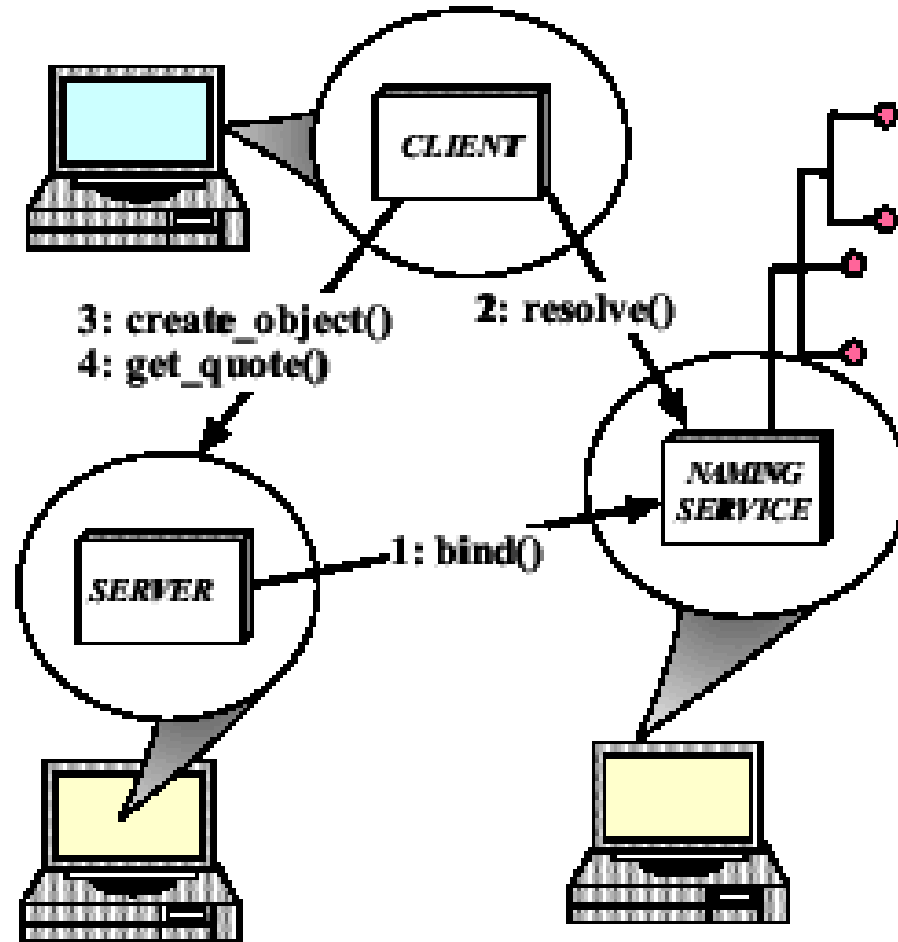
CORBA



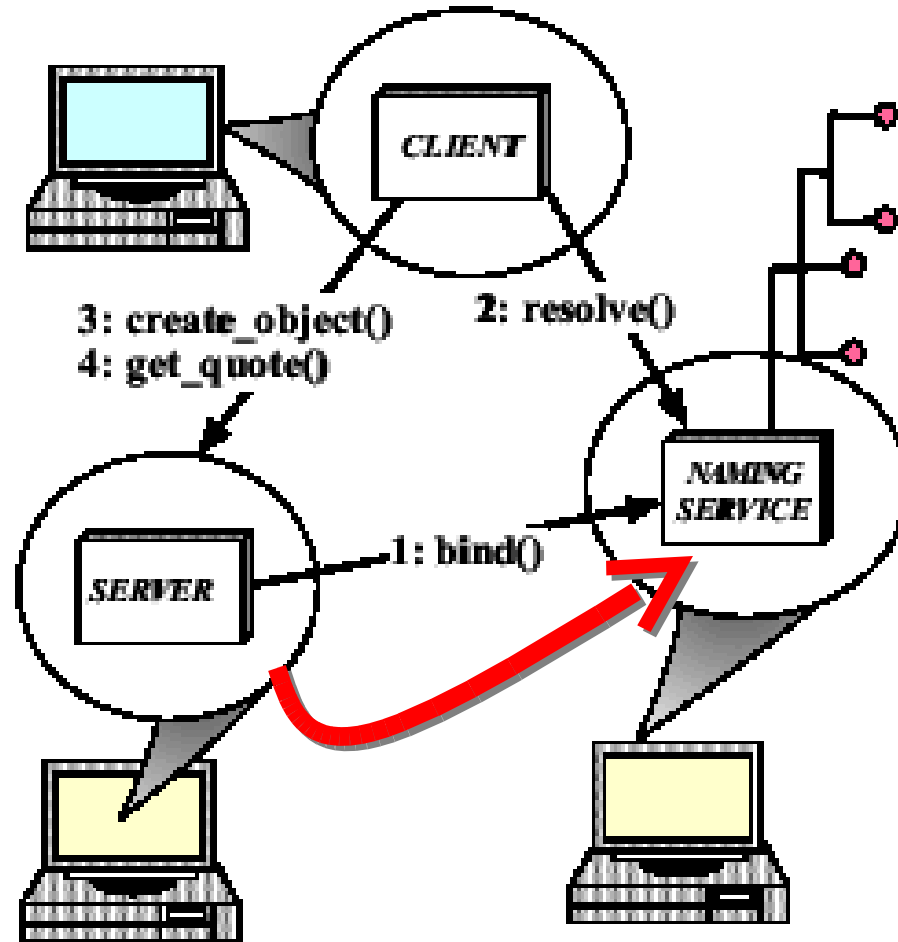
Serviço de Nomes

- Um Serviço de Nomes permite que objetos remotos registrem seus serviços
- *Binding*: associação de um nome com um objeto
- RMI e CORBA precisam de um Serviço de Nomes

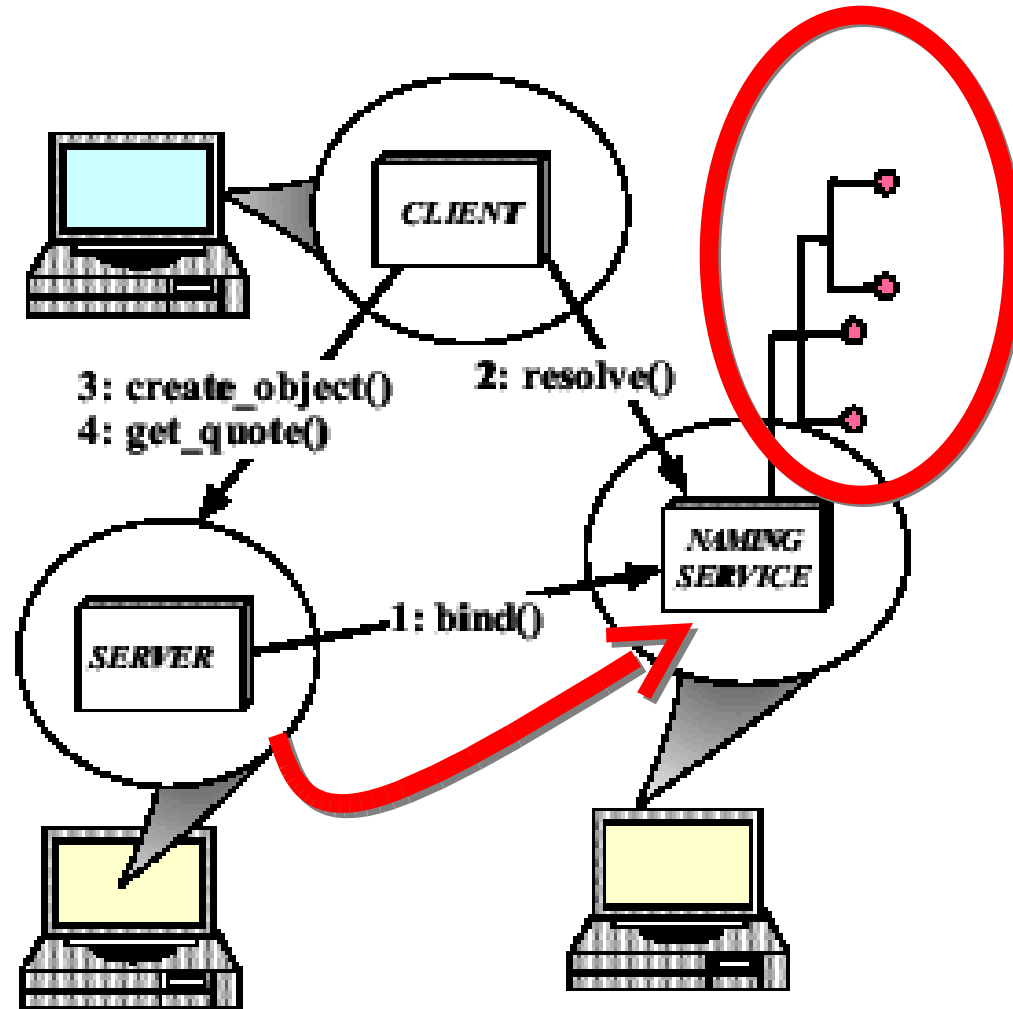
Serviço de Nomes



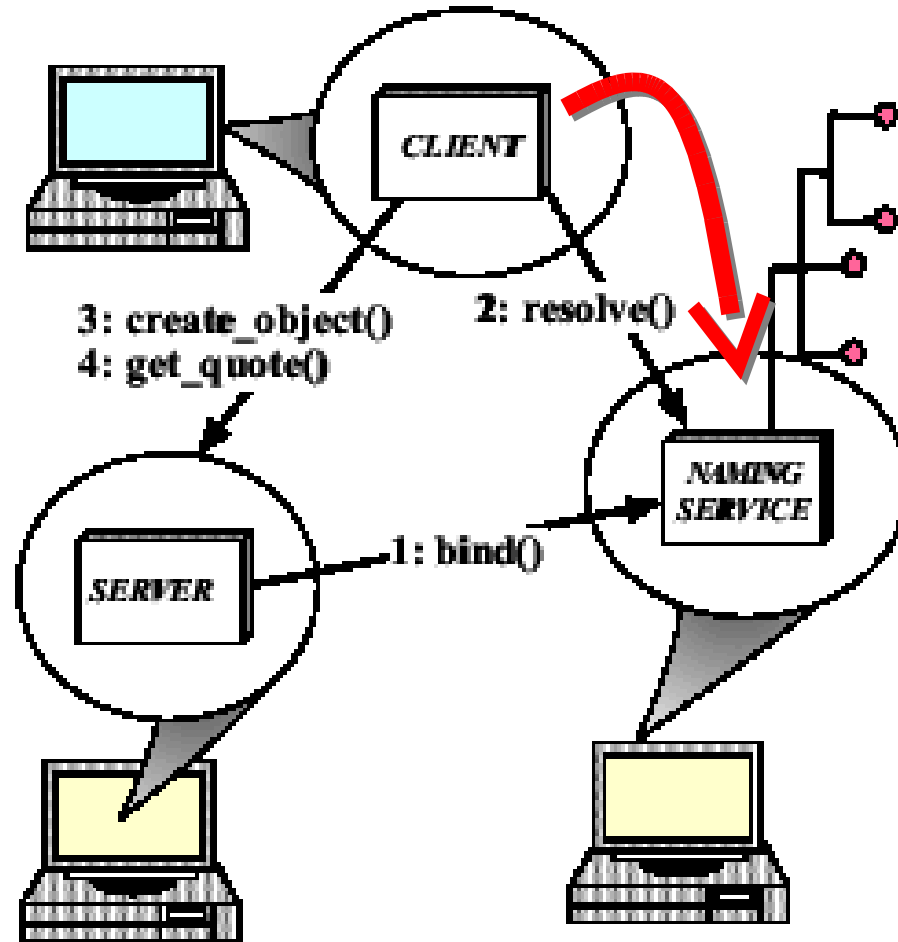
Serviço de Nomes



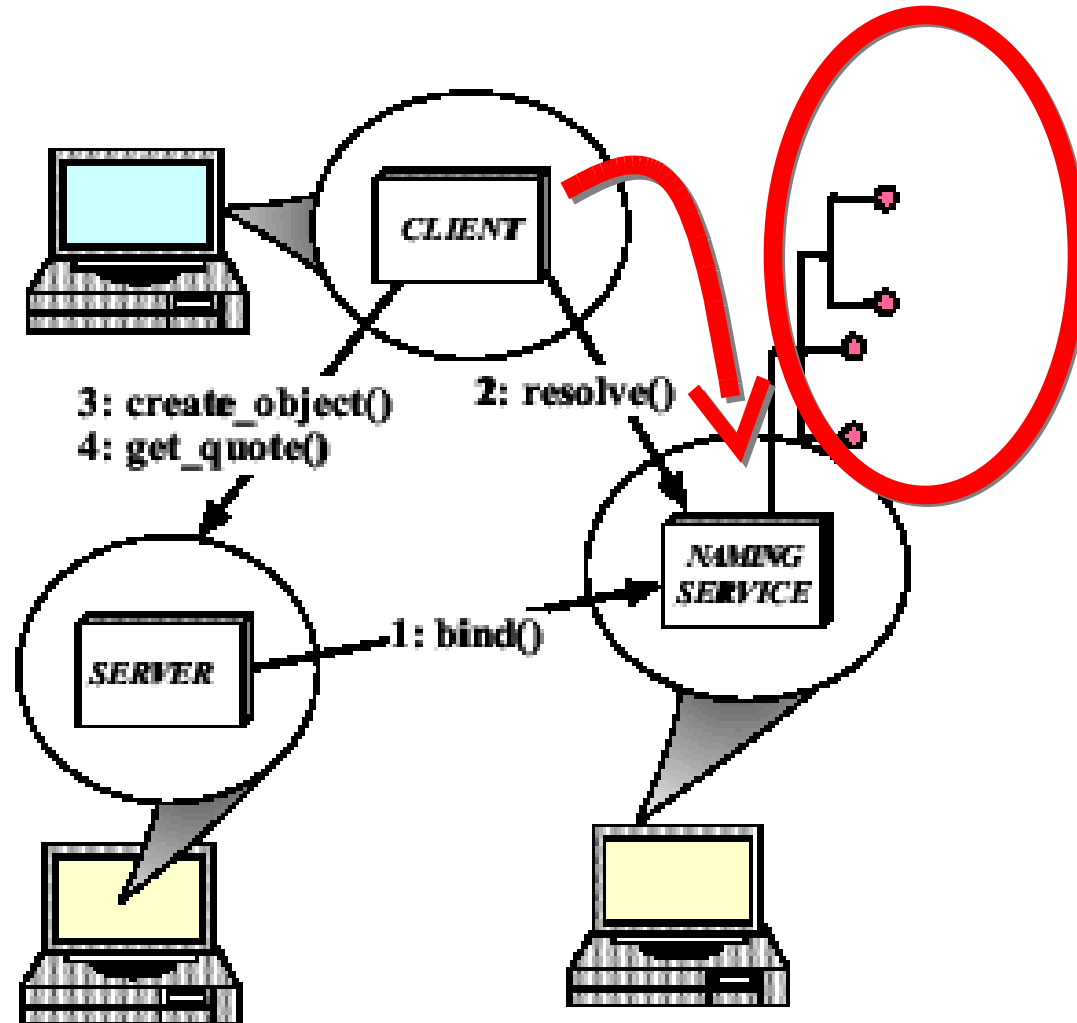
Serviço de Nomes



Serviço de Nomes



Serviço de Nomes



JNDI

Java Naming and Directory Interface

-Serviço de Nomes: associação de nomes a objetos

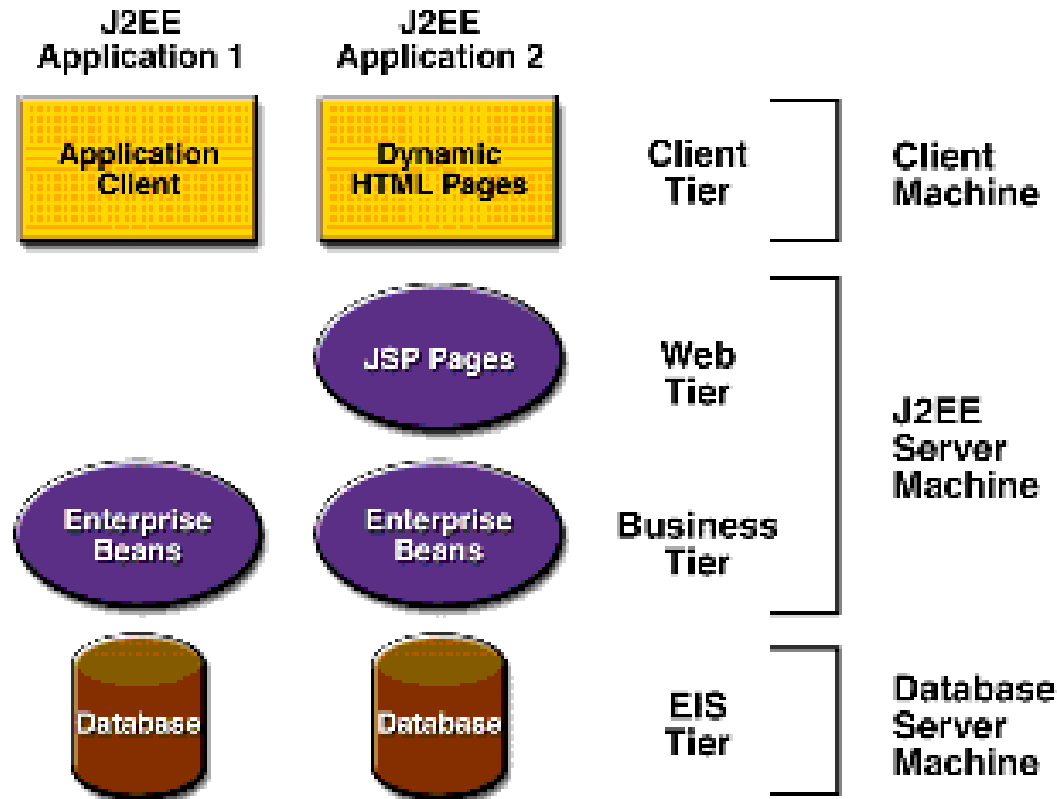
Serviço de Diretório: permite que objetos armazenados tenham atributos

-JNDI fornece funcionalidades de nome e diretório para aplicações em Java

-Independente de qualquer diretório específico

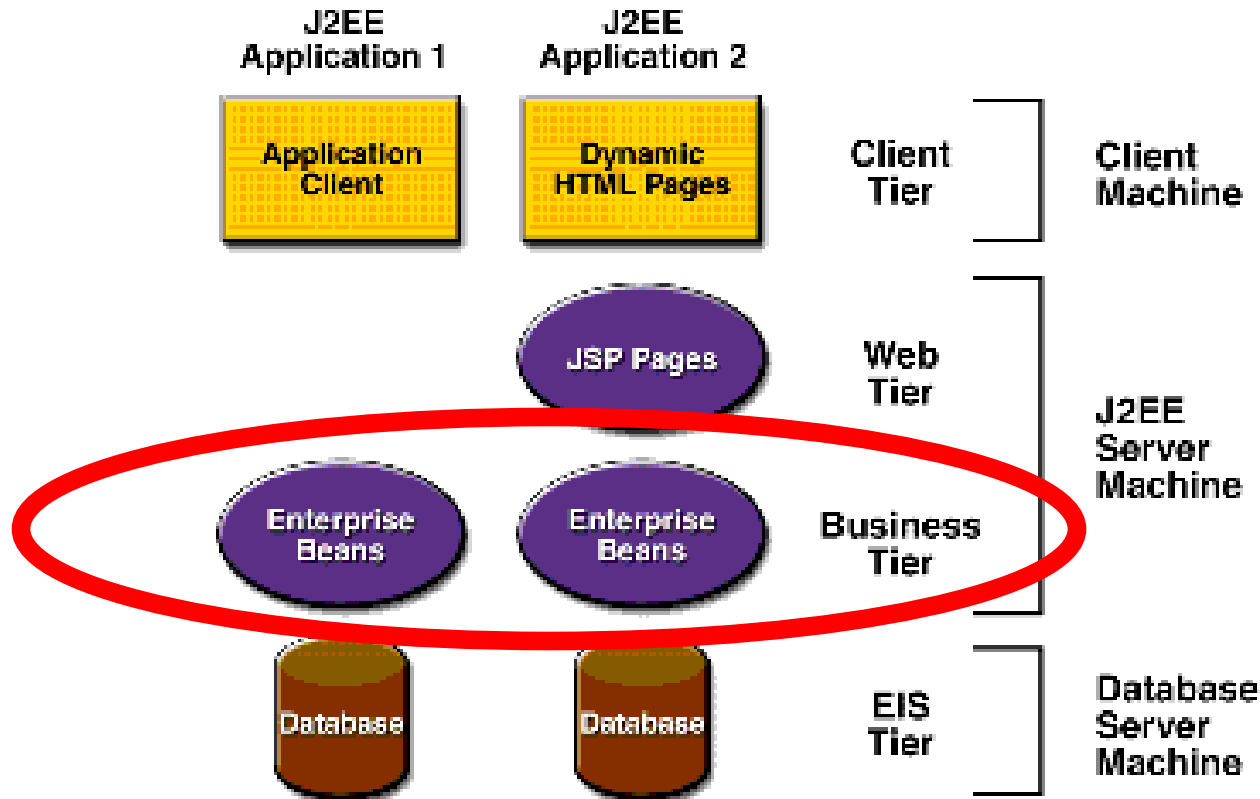
J2EE

Java 2 Platform, Enterprise Edition

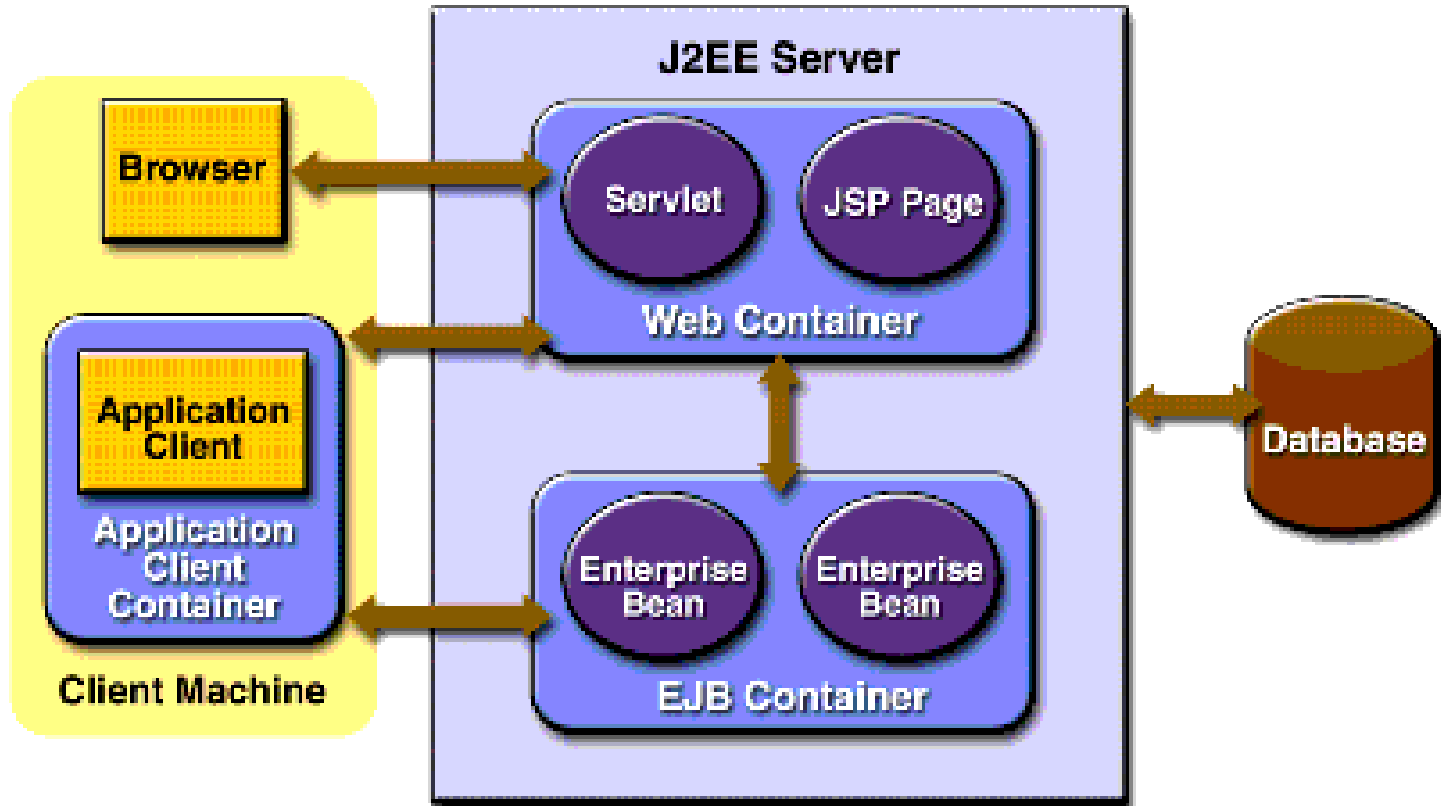


J2EE

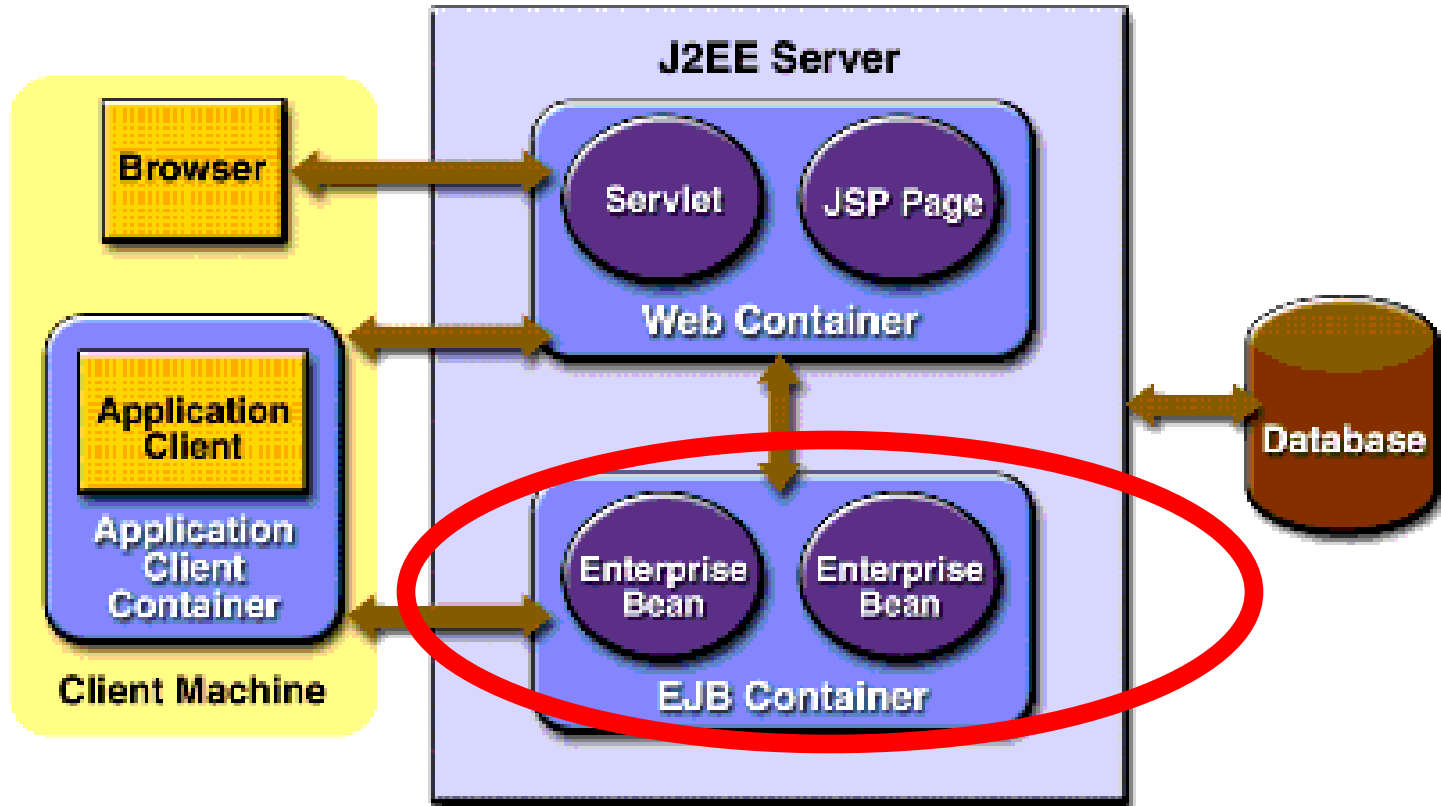
Java 2 Platform, Enterprise Edition



J2EE



J2EE



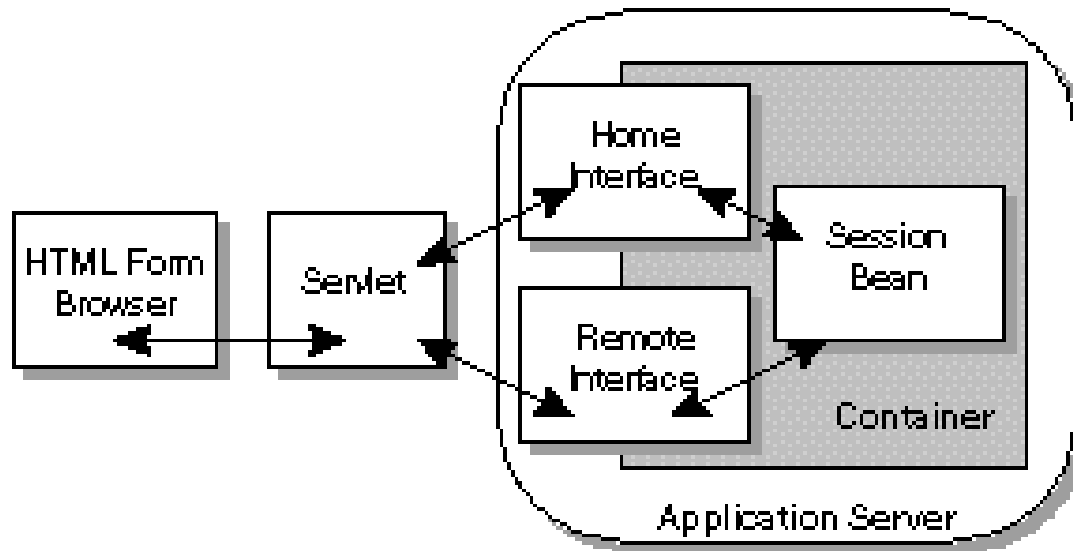
EJB

Enterprise Java Beans

- Código com atributos e métodos que implementam regras de negócio.
- Pode ser utilizado sozinho ou com outros EJBs para executar a lógica de negócio no servidor J2EE.

EJB

Enterprise Java Beans



Objetos Distribuídos em Java

RMI

EJB

CORBA

Sockets

JNDI

Prof. Sérgio Teixeira de Carvalho

sergiocarvalho@anhanguera.edu.br

Slides

<http://www.sergiocarvalho.net>

