



CORBA =  
**interoperabilidade  
+ serviços**



## Transpatências CORBA

- ◉ Transparência de **Acesso**
  - A interação A-B é idêntica seja B local ou remoto (representação de dados).
- ◉ Transparência de **Localização**
  - A implementação da interação A-B é independente da localização exata de B.

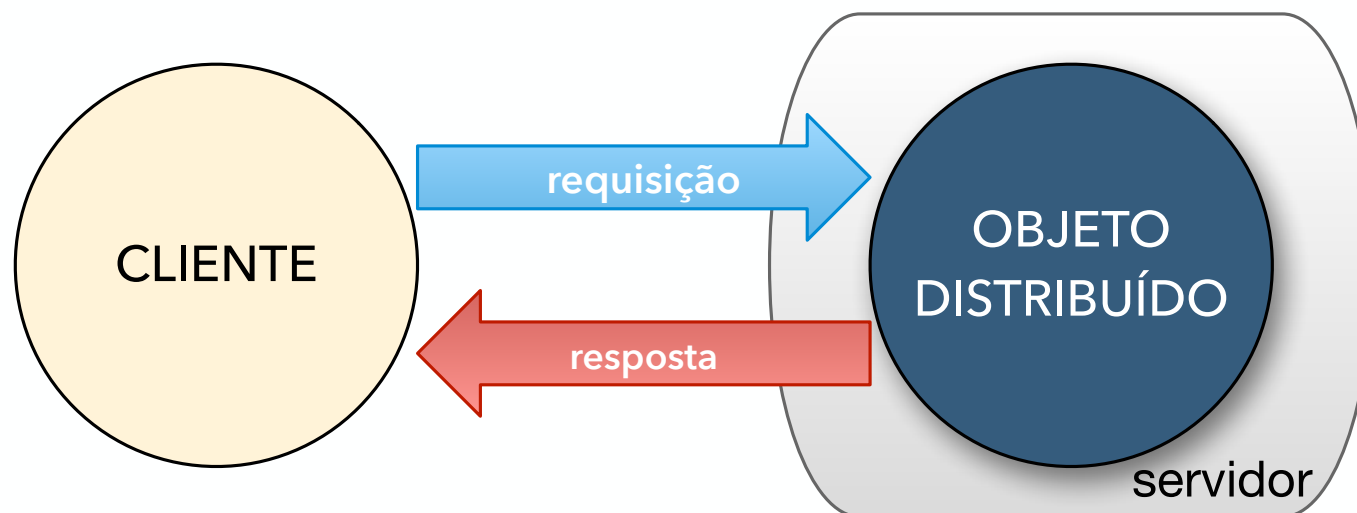


## Interações entre Objetos

- Objetos CORBA:
  - Podem estar em **qualquer lugar** da rede.
  - São agrupados em **componentes binários** acessados por meio da invocação de métodos.
- Linguagem de implementação, compilador usado e localização do objeto são **transparentes** para o cliente.
- Clientes precisam conhecer somente a **interface** publicada pelos objetos servidores.



# O Modelo de Programação





## Modelo de Programação

- **Objeto** distribuído
  - Encapsula estado interno
- Operação/**método**
  - Ponto de acesso a um procedimento executável
- **Interface**
  - Conjunto de operações/métodos (única por objeto)
- **Interações**
  - Cliente/servidor (síncronas ou assíncronas)



# Interfaces

Objetos Distribuidos - CORBA



## *Interface Definition Language (IDL)*

- ◉ Para padronizar a **aparência externa** dos objetos.
- ◉ Especificação da interface (e não do código) em uma linguagem neutra – **IDL**.
  - IDL é uma linguagem **puramente declarativa** (i.e., sem detalhes de implementação).
  - IDL define **interfaces contratuais** entre clientes e servidores.



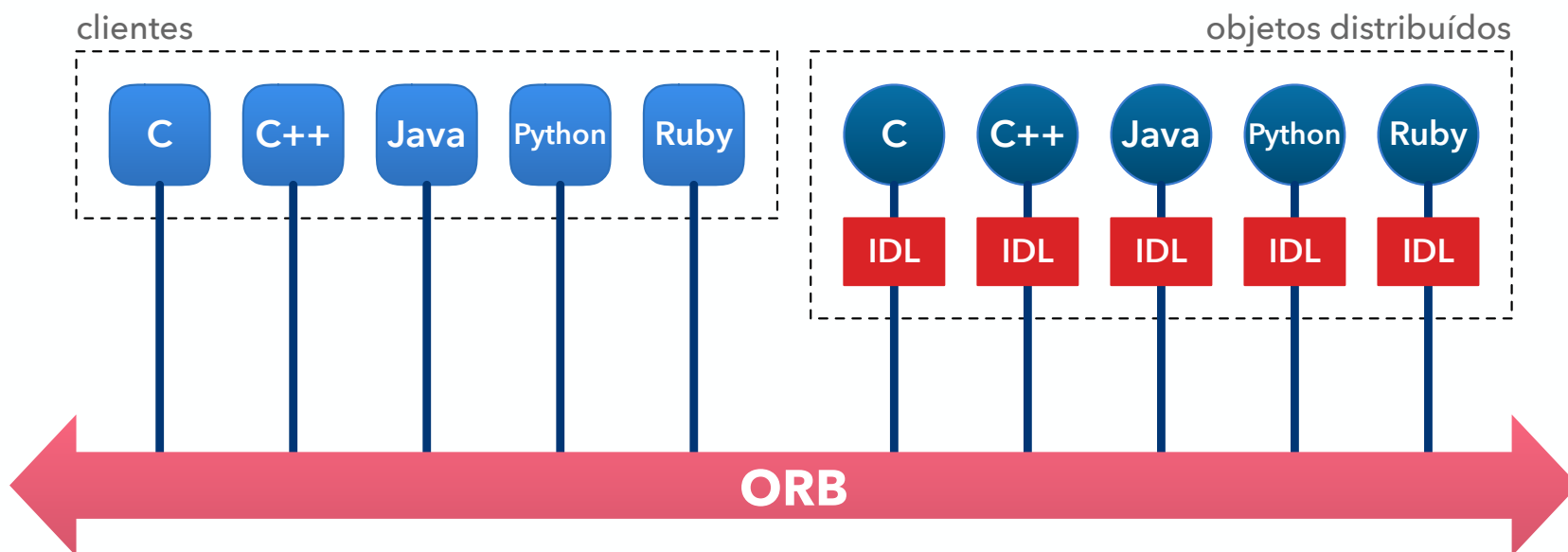
IDL permite a definição da interface **independentemente** de sistema operacional, linguagem de programação e rede para todos os serviços e componentes do ambiente CORBA.

*IDL = Interface Definition Language*





## Independência de IDL da Linguagem de Programação



**Arquivos IDL podem ser compilados!**



- Compilação:
  - para gerar código de comunicação (*stubs* e *skeletons*).
- Várias **linguagens-alvo**: C/C++, Java, Ruby, Python, Smalltalk, COBOL, etc.
- Tipos IDL são mapeados em tipos e código auxiliar na linguagem-alvo.