

# Object Brokers

Tecnologias de Middleware 2004/2005

André Santos

# Resumo

- O que são Object Brokers?
  - Como surgiu o conceito?
  - CORBA
    - Exemplos de utilização
    - Comparação com Java RMI
- Actualidade (J2EE, .NET)

# O que são Object Brokers?

- object - conceito de Objecto em linguagens de programação
- broker - “...one that acts as an agent for others...” (digamos em português, “agente”)
- object brokers - middleware que constitui uma infraestrutura (distribuída) para interoperabilidade entre Objectos

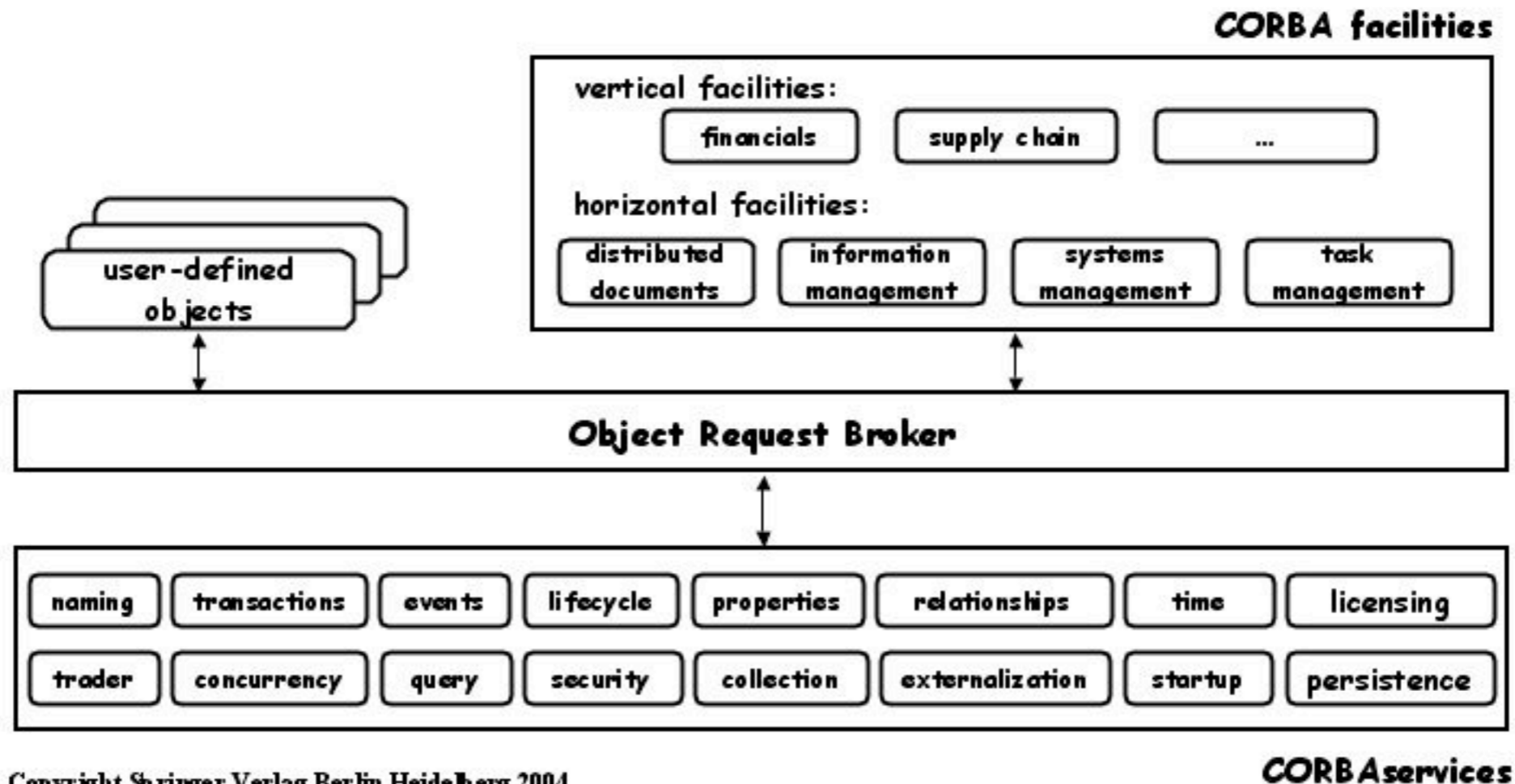
# Como surgiu o conceito?

- Surgiram no início dos anos 90
- Transposição do conceito de RPCs para aplicações orientadas a objectos
- Suporte de herança e polimorfismo
- Geralmente associado a CORBA (OMG)
- DCOM, COM+ também são Object Brokers (Microsoft)

# CORBA

- CORBA - Common Object Request Broker Architecture
- independente da plataforma e linguagem (Common)
- ORB - Object Request Broker
- Services (e.g. Naming)
- Facilities (e.g. Distributed Documents)

# CORBA: Componentes



# CORBA:IDL

Exemplos de correspondências

<u>IDL</u>	<u>Java</u>
octet	byte
enum	class
long	int
long long	long

IDL

```
module HelloApp
{
  interface Hello
  {
    string sayHello();
  };
};
```

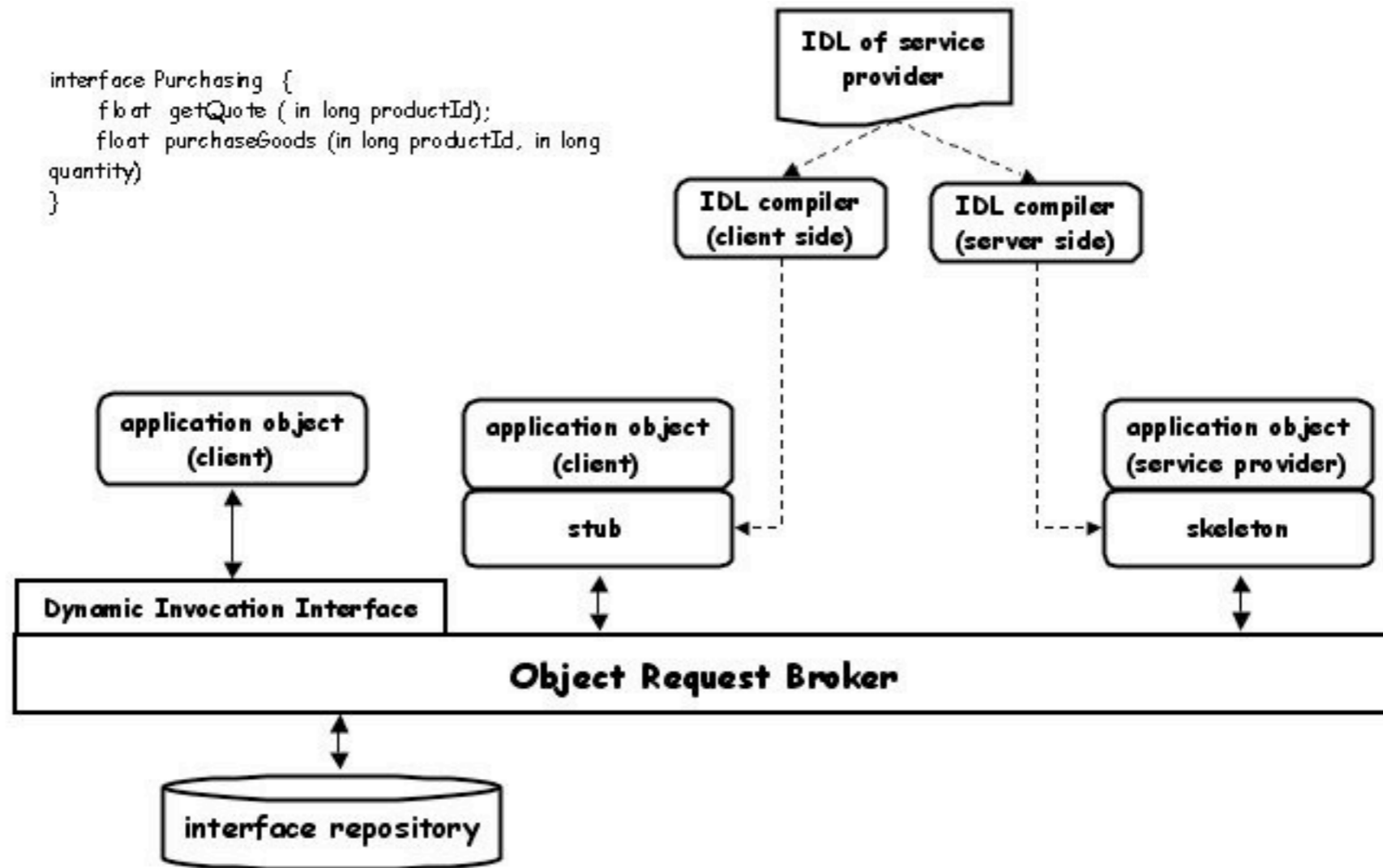
---

```
public interface Hello extends
  HelloOperations,
  org.omg.CORBA.Object,
  org.omg.CORBA.portable.IDLEntity
{
}
```

Java

```
public interface HelloOperations
{
  String sayHello ();
}
```

# CORBA:Arquitectura





# CORBA:

## Encapsulamento

- um cliente não tem que saber qual a linguagem e plataforma do servidor
- ambos os lados podem mudar a concretização e sua lógica
- transformações IDL standard
- transparência na localização (Naming Service)

# CORBA: História

## (resumo)

- 1991 - v1.0: versão inicial; modelo de objectos; IDL; invocação dinâmica
- 1996 - v2.0: GIOP, IIOP
- 1998 - v2.2: transformações IDL para Java
- 2000 - v2.4: Tempo-real
- 2002 - v3.0: Modelo de componentes (ao estilo de J2EE); integração com Java

GIOP - General Inter-ORB Protocol  
IIOP - Internet Inter-ORB Protocol

# Utilização (cliente)

```
try{

    // Criação e inicialização do ORB
    ORB orb = ORB.init(args, null);

    // Obter a raiz do serviço de nomes
    org.omg.CORBA.Object objRef = orb.resolve_initial_references("NameService");
    NamingContext ncRef = NamingContextHelper.narrow(objRef);

    // Obter a referência para o objecto servidor através do serviço de nomes
    NameComponent nc = new NameComponent("Hello", "");
    NameComponent path[] = {nc};
    Hello helloRef = HelloHelper.narrow(ncRef.resolve(path));

    // Invocar um método no servidor
    String s = helloRef.sayHello();

    System.out.println(s);

}
catch(Exception e) {
    System.out.println("ERRO: " + e);
    e.printStackTrace(System.out);
}
```

# Utilização (servidor)

```
try{

    // Criação e inicialização do ORB
    ORB orb = ORB.init(args, null);

    // Instanciar o objecto servidor e regista-lo no ORB
    HelloServant helloRef = new HelloServant();
    orb.connect(helloRef);

    // Obter a raiz do serviço de nomes
    org.omg.CORBA.Object objRef = orb.resolve_initial_references("NameService");
    NamingContext ncRef = NamingContextHelper.narrow(objRef);

    // Associar o objecto servidor ao nome
    NameComponent nc = new NameComponent("Hello", "");
    NameComponent path[] = {nc};
    ncRef.rebind(path, helloRef);

    // Esperar por invocações
    java.lang.Object sync = new java.lang.Object();
    synchronized(sync){
        sync.wait();
    }
}
catch(Exception e) {
    System.err.println("ERROR: " + e);
    e.printStackTrace(System.out);
}

class HelloServant extends _HelloImplBase
{
    public String sayHello() {
        return "olá!";
    }
}
```

# Invocação Dinâmica

- torna possível a “descoberta” e sua invocação de novos métodos em tempo de execução
- estes não foram incluídos estaticamente no stub/skeleton
- problema: a semântica dos “novos” serviços

# Utilização (invocação dinâmica)

(...)

```
org.omg.CORBA.Request request = aircraft._request("sayHello");  
request.set_return_type(orb.get_primitive_tc(org.omg.CORBA.TCKind.tk_string));  
request.invoke();  
String hello = request.result().value().extract_string();  
System.out.println (hello);
```

# CORBA vs. Java RMI

## RMI

- + Portável (entre várias plataformas)
- - só suporta plataformas com Java

## CORBA

- + útil para integração de sistemas legados (e.g. Cobol)
- - não suporta transferência de objectos ou código

# CORBA: Exemplo de caso de sucesso

- The Weather Channel (TWC)
  - Integração de dados (meteorológicos) provenientes de várias fontes diferentes
  - Difusão (por TV) de diferentes dados consoante a localização geográfica
  - Tecnologia: RedHat Linux 6.2, MICO (implementação de CORBA livre), Python



# Actualidade

- Actualmente, a utilização de CORBA para desenvolvimento de novos sistemas tem vindo a decrescer
- Aparecimento de J2EE e mais tarde .NET
- CORBA 3 introduz interoperabilidade com Java e um modelo de componentes ao estilo de J2EE
- Prevê-se que as especificações CORBA sejam substituídas por outros standards

# CORBA, J2EE, COM+, .NET

	várias linguagens	várias plataformas
CORBA	X	X
J2EE		X
COM+	X*	
.NET	X	X

\* - embora todas proprietárias da Microsoft

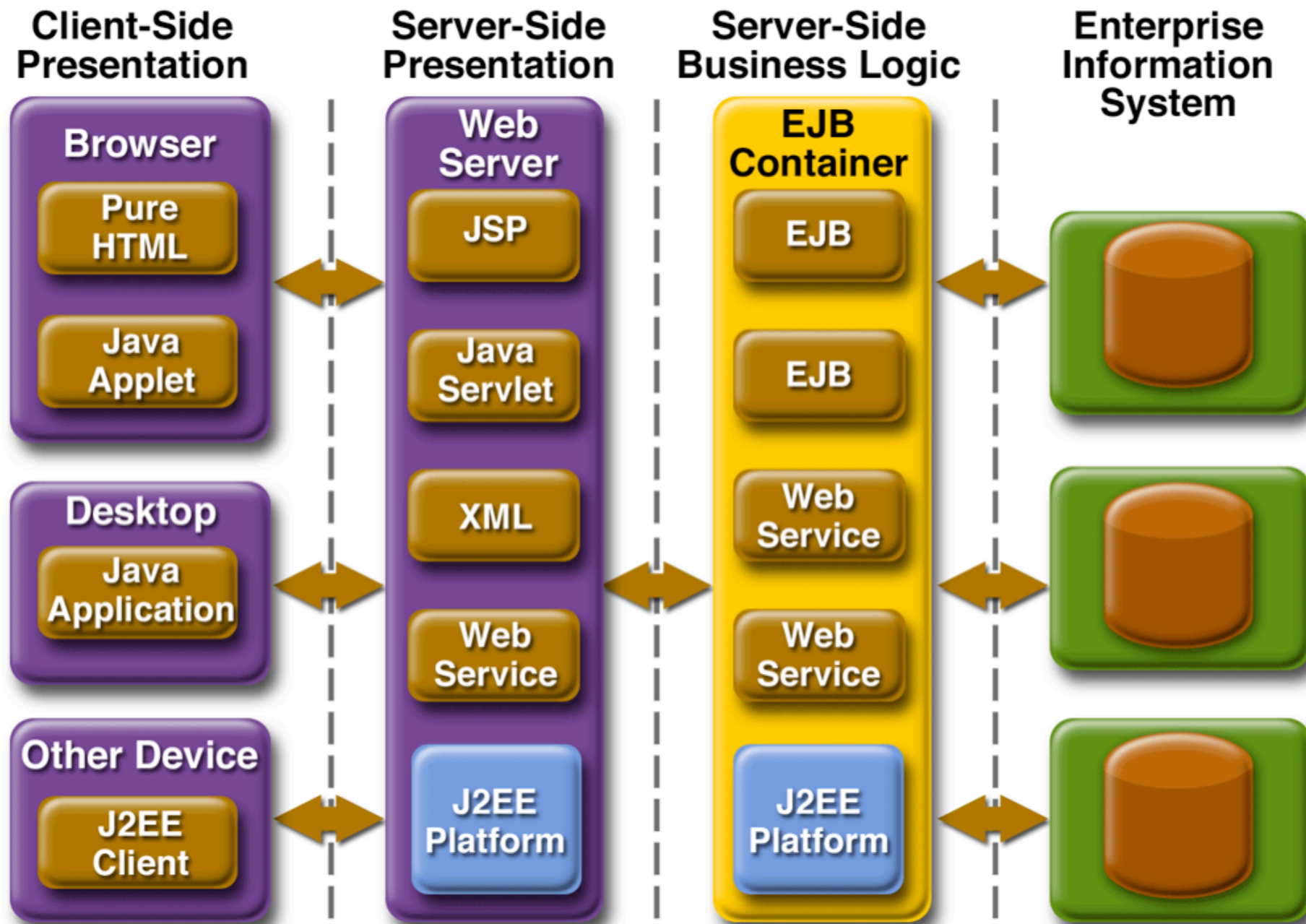
# Porquê o “abandono” de CORBA?

- Alguns serviços do CORBA foram meras especificações
  - (e.g.) Não contem monitor transaccional
- Ineficiência das implementações de Object Brokers
- IIOP não eficiente na Internet

# J2EE

- J2EE - Java 2 Enterprise Edition (Sun Microsystems)
- Infraestrutura de middleware, adequada ao desenvolvimento de aplicações para a Web
- Componentes desenvolvidos em Java, que podem executar em plataformas que corram JVM (Java Virtual Machine)

# J2EE



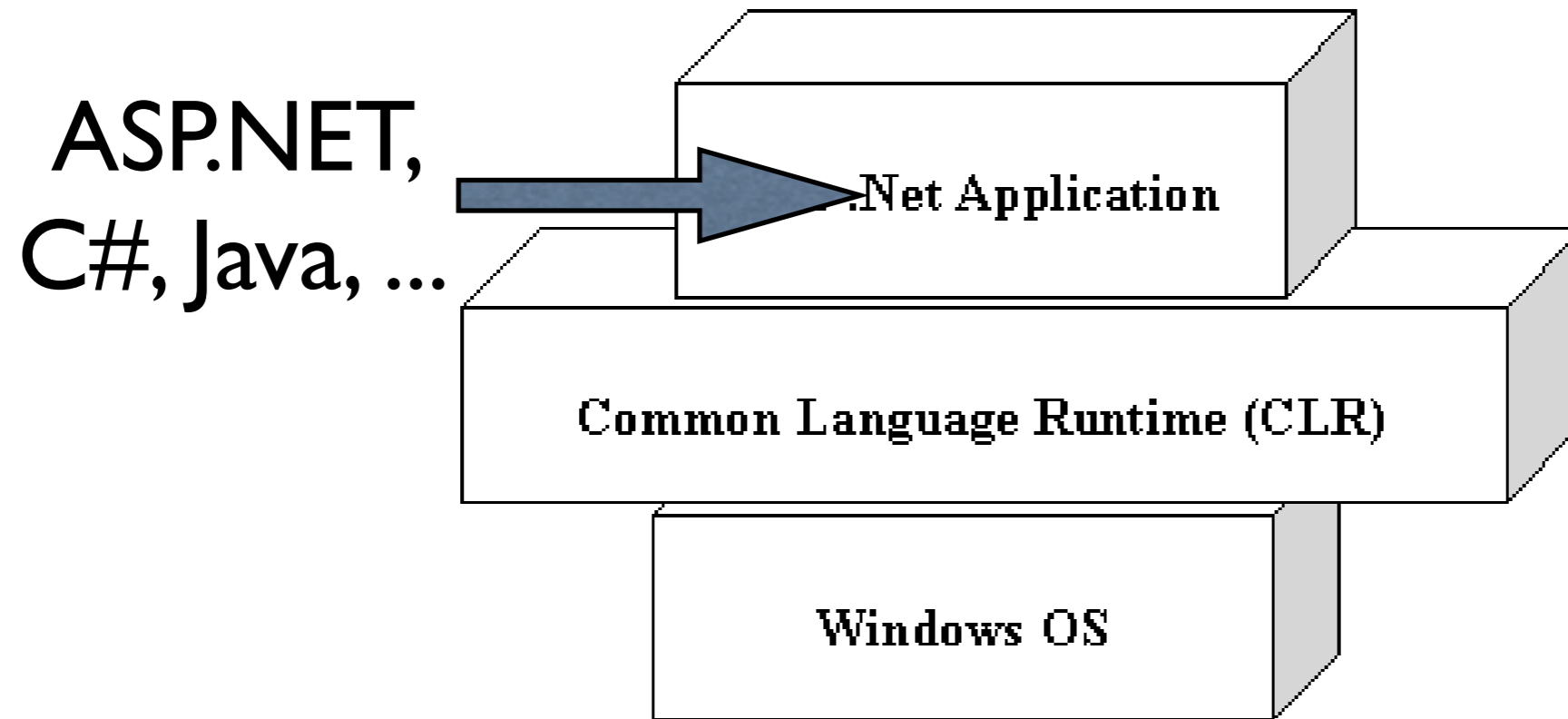
# J2EE

- Semelhanças ao CORBA:
  - Orientada a objectos
  - Interoperabilidade entre objectos: RMI
  - Exportação de serviços (entre diferentes plataformas/linguagens): Web Services
  - Suporte de serviço de nomes
  - Suporte de transacções
- Interoperavel com CORBA

# .NET

- Substitui COM+, continuando compatível
- Common Language Runtime (CLR)
- Permite utilizar várias linguagens (e.g. Java) que compilam em MSIL (Microsoft Intermediate Language) que corre na CLR

# .NET





# .NET

- Semelhanças com CORBA:
  - Orientada a objectos
  - Interoperabilidade entre objectos: RPC
  - Exportação de serviços (entre diferentes plataformas/linguagens): Web Services

# Saber mais...

- História do CORBA: [http://www.omg.org/gettingstarted/history\\_of\\_corba.htm](http://www.omg.org/gettingstarted/history_of_corba.htm)
- Utilizar CORBA com Java: <http://java.sun.com/j2se/1.3/docs/guide/idl/index.html>
- CORBA vs. Java RMI: [http://www.javacoffeebreak.com/articles/rmi\\_corba/](http://www.javacoffeebreak.com/articles/rmi_corba/)
- .NET vs. J2EE: [http://java.oreilly.com/news/farley\\_0800.html](http://java.oreilly.com/news/farley_0800.html) (de 2000, mas dá uma boa ideia)  
<http://www.theserverside.com/articles/article.tss?I=J2EE-vs-DOTNET> (de 2001)

# Links úteis

- **OMG:** <http://www.omg.org> (CORBA)
- **Microsoft .NET:** <http://www.microsoft.com/net/>
- **Sun J2EE:** <http://java.sun.com/j2ee/>