

Sistemas Distribuidos

Lunes, 23 de enero

- Objetos y componentes distribuidos

Objetos distribuidos

- Beneficios:
 - Encapsulamiento
 - Abstracción de datos provee una clara separación entre la especificación y la implementación
 - Soluciones más dinámicas y extensibles

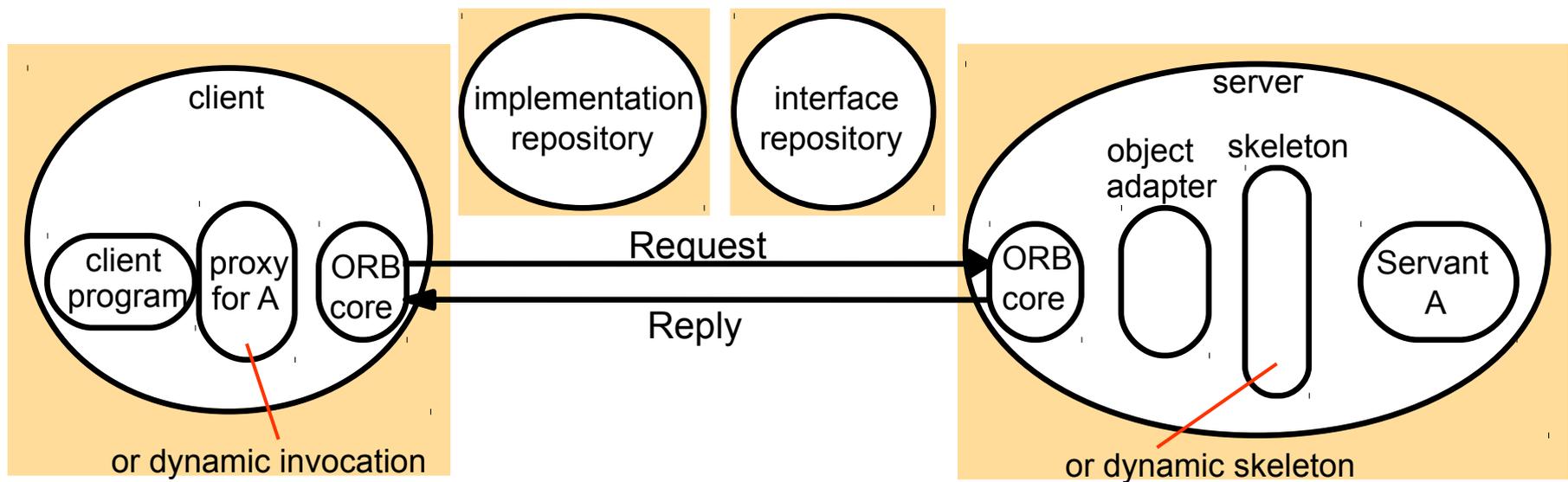
Objetos vs componentes

- Limitaciones de los objetos distribuidos:
 - Dependencias implícitas
 - Complejidad en la programación
 - Falta de separación de intereses
 - Falta de soporte para la implementación
- Componentes distribuidos:
 - Evolución natural de los objetos distribuidos

Objetos distribuidos

<i>Objects</i>	<i>Distributed objects</i>	<i>Description of distributed object</i>
Object references	Remote object references	Globally unique reference for a distributed object; may be passed as a parameter.
Interfaces	Remote interfaces	Provides an abstract specification of the methods that can be invoked on the remote object; specified using an interface definition language (IDL).
Actions	Distributed actions	Initiated by a method invocation, potentially resulting in invocation chains; remote invocations use RMI.
Exceptions	Distributed exceptions	Additional exceptions generated from the distributed nature of the system, including message loss or process failure.
Garbage collection	Distributed garbage collection	Extended scheme to ensure that an object will continue to exist if at least one object reference or remote object reference exists for that object, otherwise, it should be removed. Requires a distributed garbage collection algorithm.

CORBA



Ejemplo de arq. de componentes

