
Lección 8

Introducción a las llamadas a procedimientos remotos (RPC)



Objetivo

Extender a los sistemas distribuidos el mecanismo de llamadas a procedimientos y subrutinas de los lenguajes de programación.

Programa Principal

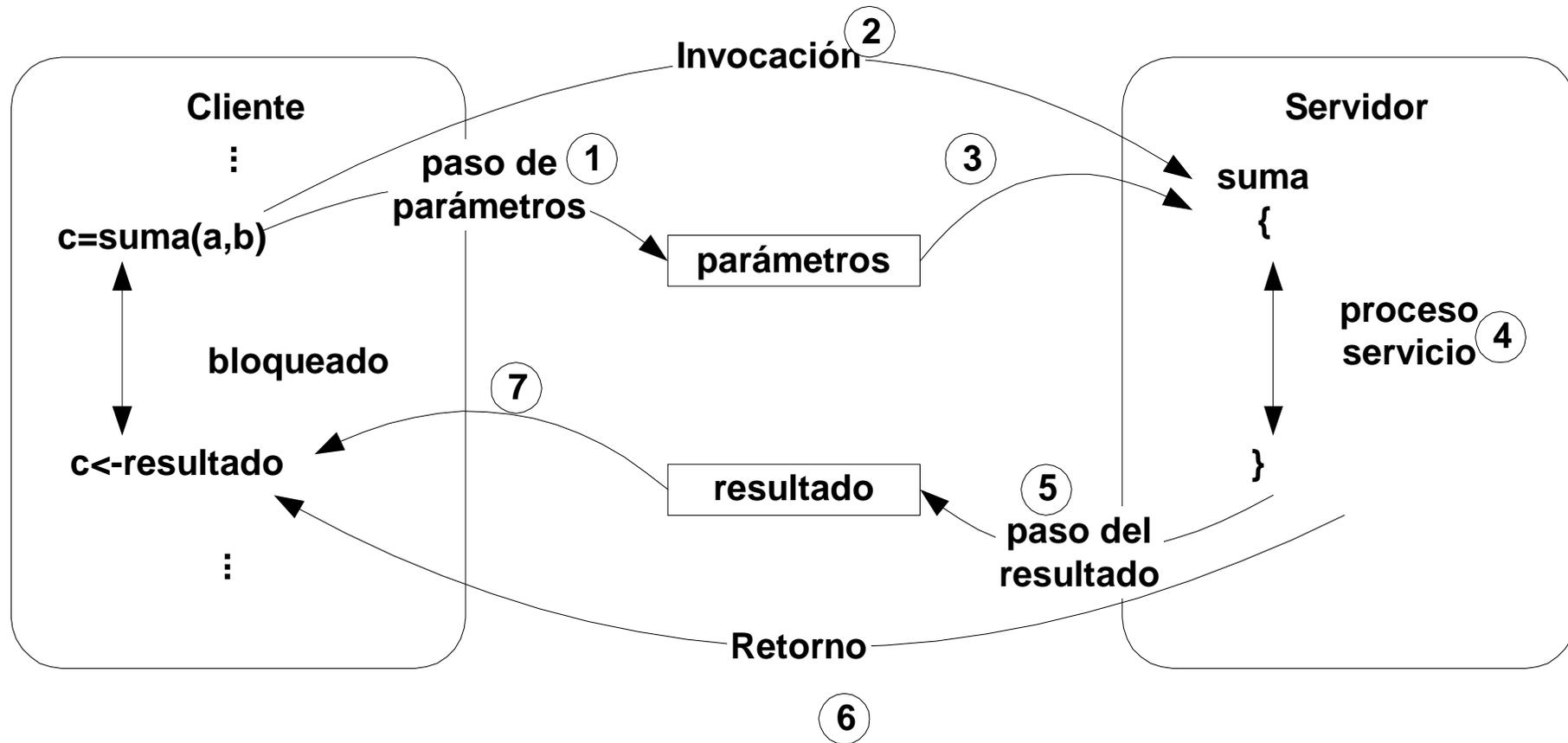
```
.....  
c=suma(a,b);  
.....
```

Subrutina

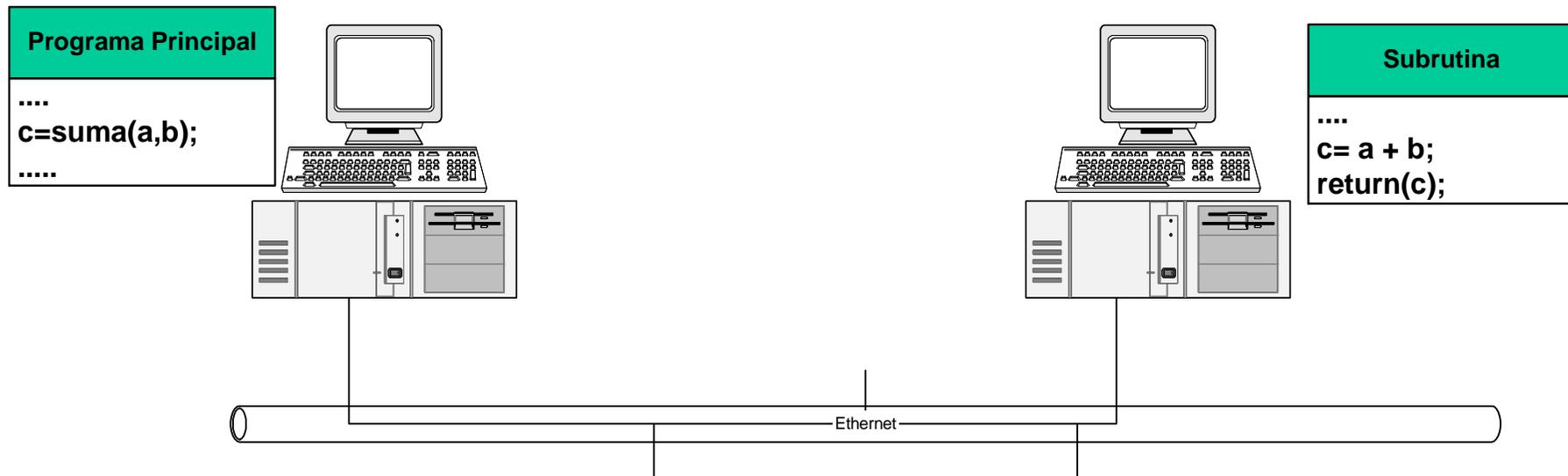
```
.....  
res= a + b;  
return(res);
```



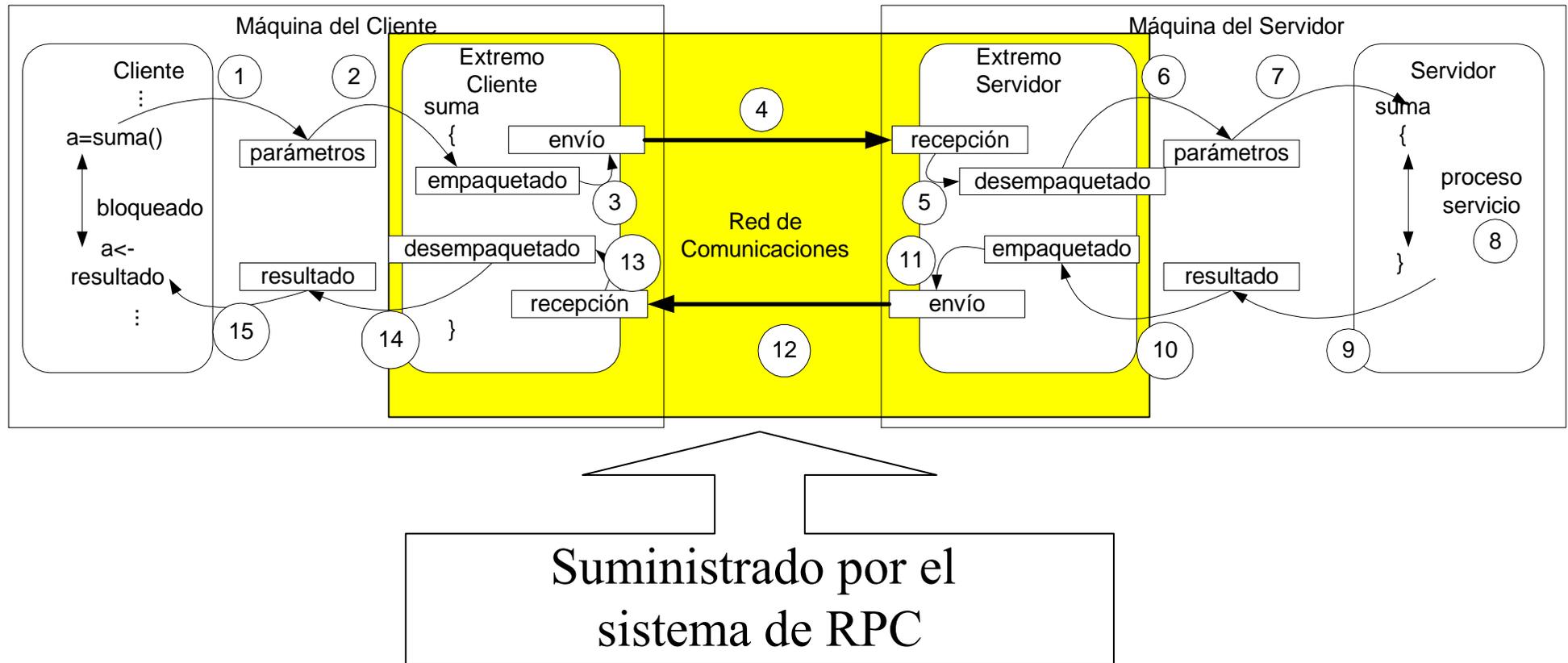
Mecanismo de la llamada a un procedimiento



Objetivo

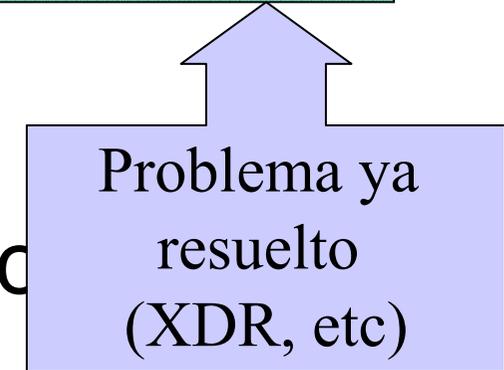


Mecanismo de las RPC



Problemática a resolver por las RPC

- Heterogeneidad de la información.
- Enlazado e identificación.
- Generación de los extremos cliente y del servidor.
- Transparencia.
- Tolerancia a fallos.



Enlazado e identificación

Problema: ¿dónde está el procedimiento? ¿Cómo sé que es el correcto?.

Solución en sistemas locales:

- ¿Dónde está?: Incorporado en el ejecutable en tiempo de compilación (enlazado estático) o en una librería a la que se accede en tiempo de ejecución (enlace dinámico).
- ¿Identificación? Por nombre en el momento de la compilación.



Enlazado e identificación

Sistemas distribuidos

Localización:

¿Dónde está el procedimiento? ¿Máquina? ¿Puerto de protocolo?

Identificación:

¿Cómo estoy seguro de que en ese lugar se encuentra el procedimiento que busco?

Ambos problemas se solucionan simultáneamente.



Enlazado e identificación

Necesidades:

1. Mecanismo predefinido de búsqueda.
2. Método de identificación del servicio:
 - Qué servicio realiza.
 - Qué parámetros recibe.
 - Qué resultado devuelve.
 - Qué versión del servicio es.

En un sistema distribuido, el cliente y el servidor pueden seguir procesos de actualización distintos que imposibiliten la interoperabilidad.



Enlazado e identificación

Identificación:

- El servidor especifica de manera formal su interfaz. Se utiliza un “**Lenguaje de Especificación del Interfaz**” (IDL) propio de cada sistema de RPC. Ejemplo:

```
Specification servidor_calculo, version 2.5:
{
    [out] int suma ([in] int a, [in] int b);
    [out] float divide ([in] int a, [in] int b);
}
```



Enlazado e identificación

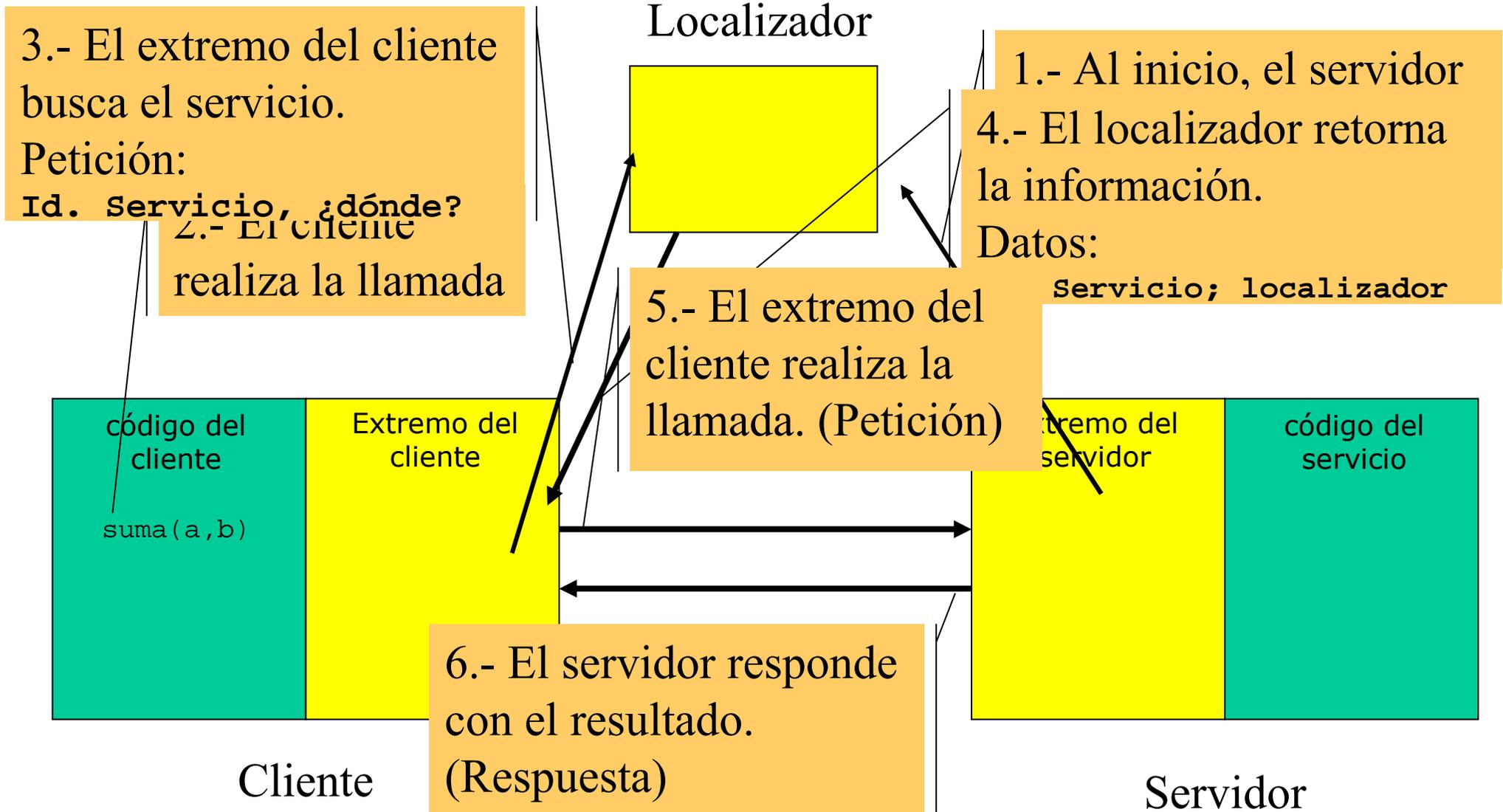
Localización:

- En tiempo de ejecución, el servidor *exporta* (hace pública) la especificación, junto con la localización (dónde se le puede encontrar).
- Exportar significa enviar el interface del procedimiento a un servicio de localización.
- El servicio de localización o localizador (*binder*) es un proceso que mantiene un registro de servicios y localizaciones. Ejemplo:

Servicio	Versión	Máquina	Puerto	protocolo
Suma	2.5	Orion	15346	Tcp
Divide	2.5	Orion	11927	Tcp

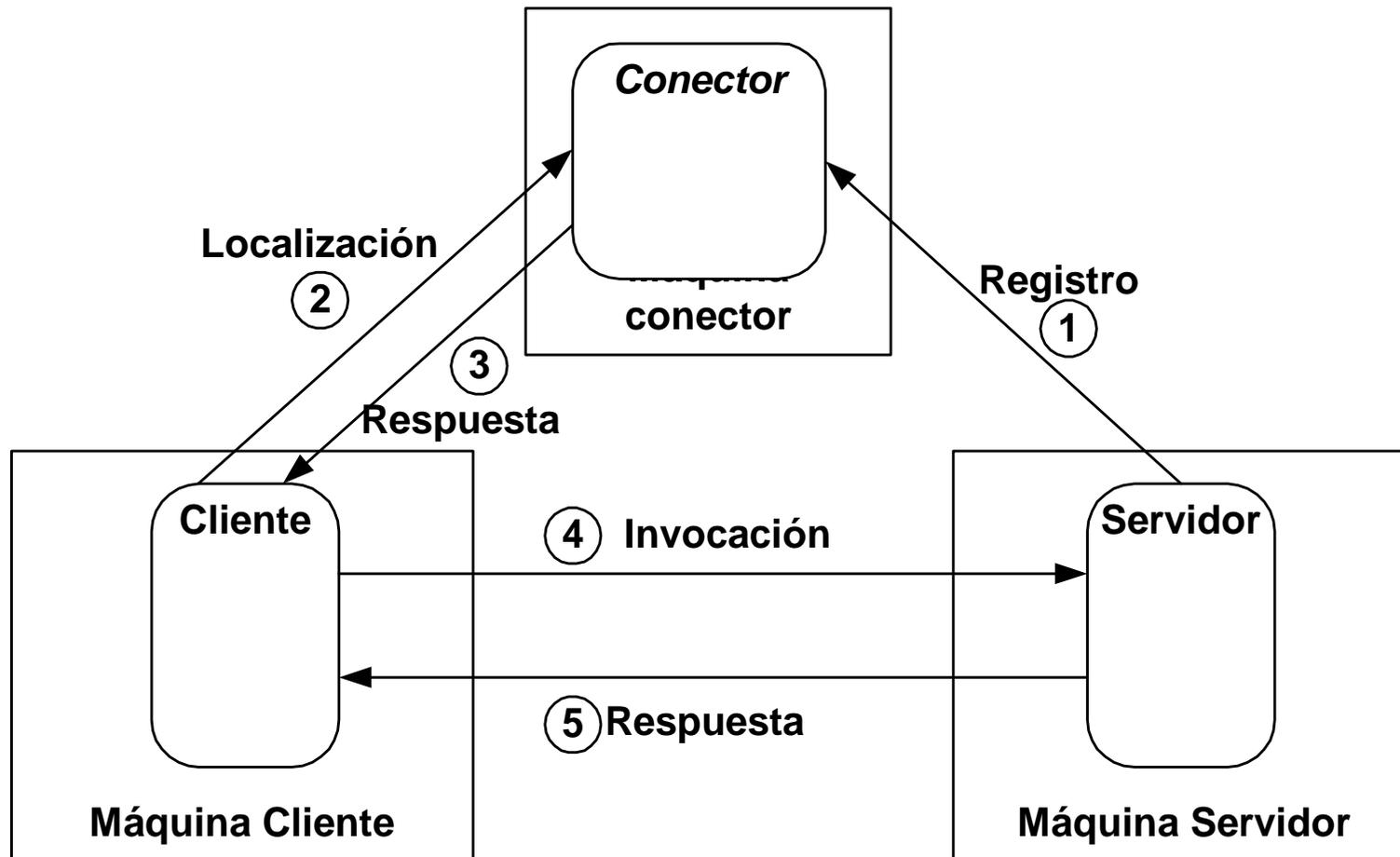


Enlazado. Uso del localizador



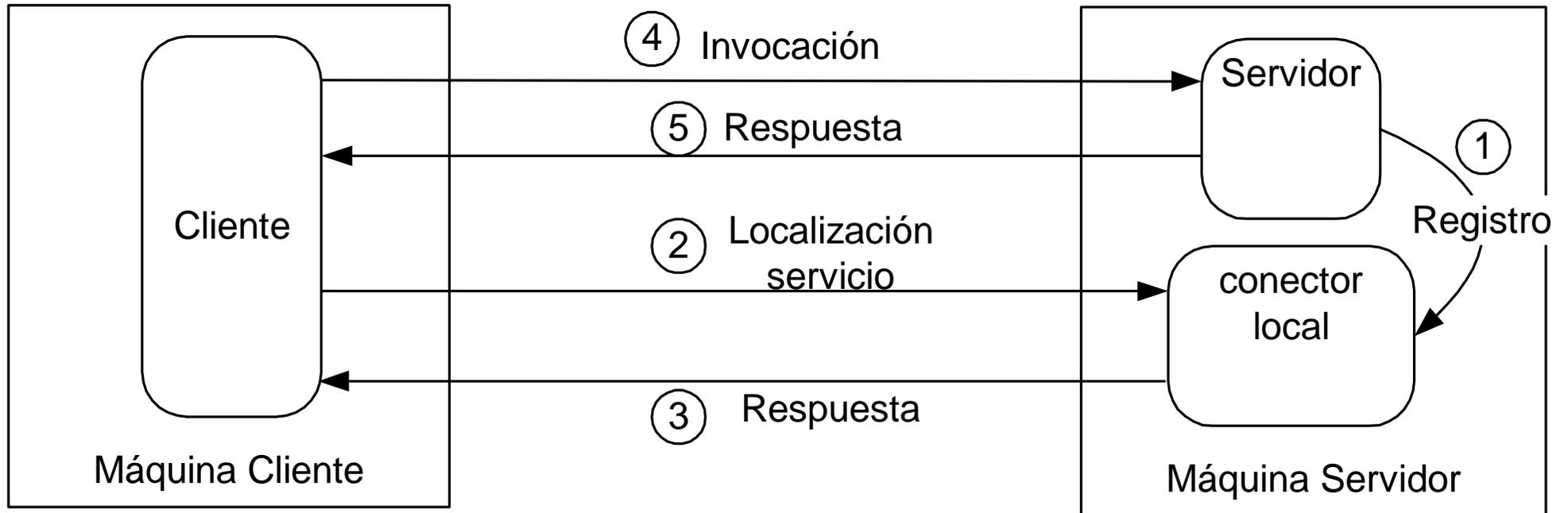
Tipos de localizadores

Localizador Centralizado (Páginas amarillas)



Tipos de localizadores

Localizador Local

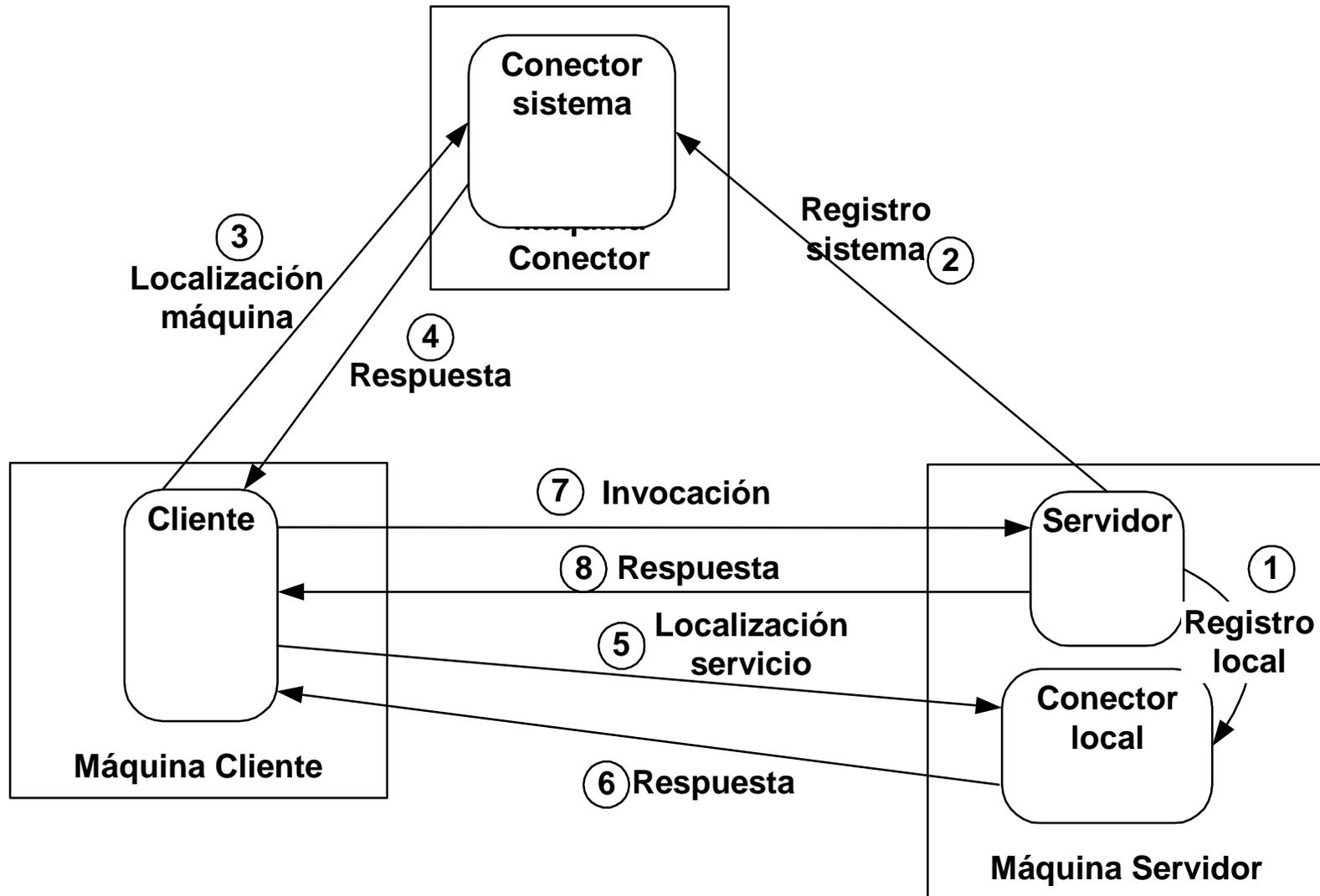


Exige conocer la máquina en la que se encuentra el servicio



Tipos de localizadores

Jerarquía de Localizadores



Generación de los extremos de cliente y servidor

- Los extremos los genera de manera automática el sistema de RPC basándose en la especificación realizada con el “**Lenguaje de Especificación del Interfaz**”.
- El trabajo lo realiza un **compilador** del Lenguaje de Especificación del Interfaz.
- Los extremos se encargarán del empaquetado y desempaquetado de los datos, el envío y la recepción a través de la red, así como la identificación, localización e invocación del servicio en la máquina remota.



Transparencia

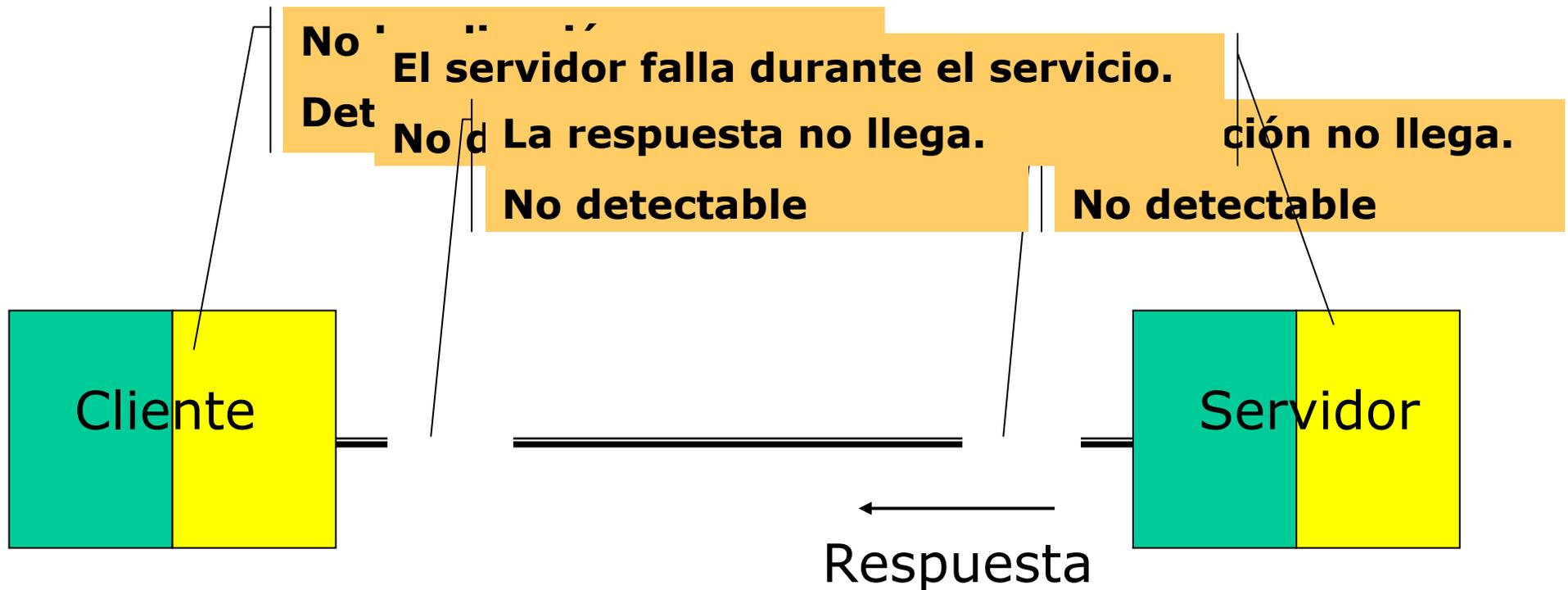
- La transparencia se refiere al grado de similitud entre una RPC y una llamada local.
- Aspectos sintácticos: Normalmente transparencia total. Indistinguibles.
- Utilización: Transparencia no total. Restricciones dependientes del sistema de RPC. Ejemplos:
 - **Acceso a variables globales.** Imposible para RPC.
 - **Paso de punteros como parámetros.** Cliente y servidor están en espacios de memoria distintos. Posible en algunos casos.



Tolerancia a fallos

Una RPC tiene más posibilidades de fallar que una llamada local y es más difícil de detectar el fallo.

Posibles Fallos



Tolerancia a fallos

- ¿Cómo detecta el cliente un fallo en una RPC?
 - Si la respuesta no llega **en un determinado plazo de tiempo ("timeout")**, la RPC se da por fallida.
- ¿Qué puede hacer el cliente en caso de fallo?.
Opciones:
 1. No hacer nada e informar al usuario.
 2. Reintentar la RPC.
- ¿Qué puede hacer el servidor en caso de fallo?



Tolerancia a fallos. Semánticas

Cliente	Servidor		Semántica de la RPC
Reintenta RPC	Filtra duplicados de RPC	Acción realizada	
No	No aplicable	No aplicable	<i>Quizás</i>
Sí	No	Reejecución servicio	<i>Al menos una vez</i>
Sí	Sí	Retransmisión respuesta	<i>A lo más una vez</i>

