

Indice

Hora 1

1. Introducción
2. Asignación automática de parámetros de red con presencia de servidores
3. Zeroconf
 - 3.1 Asignación de direcciones IP: escenarios y requerimientos
 - 3.1.1 Conflict-detection allocation
 - 3.1.2 Conflict-free allocation
 - 3.1.3 Best effort allocation

Hora 2

- 3.2 Traducción de nombres
- 3.3 Asignación de direcciones IP multicast
- 3.4 Descubrimiento de servicios
4. Universal Plug and Play
5. Jini

3.2 Traducción de nombres

- ▶ Traducción entre nombre de *host* y dirección *IP*
 - Navegación web
 - Un URL típicamente contiene un nombre que ha de ser traducido a una dirección
 - Un archivo de *log* de accesos debe mapear las direcciones que contiene a nombres

Requerimientos

- Tiene que traducir un nombre de *host* a una dirección *IP*
- Tiene que traducir una dirección *IP* a un nombre de *host*
- Selección del nombre del *host*
 - Cómo un *host* es asignado a un nombre de dominio se determina por algún otro protocolo de configuración y no forma parte del objeto de Zeroconf
 - Un protocolo tiene que permitir averiguar a un *host* si su propio nombre es único

Requerimiento

- Tiene que permitir a un *host* saber si su nombre es único, y si no lo es, notificar al usuario o a algún software de configuración de interfaz *IP* para seleccionar otro nombre, y entonces repetir el proceso de verificación del nombre único.

Traducción de nombres

► Multicast DNS

- Peticiones DNS casi estándar en la que la dirección destino es multicast
 - 224.0.0.251 for IPv4
 - FF02::FB for IPv6
- Clientes mDNS usan puerto origen/destino 5353 y el formato del paquete cambia levemente.
- Nombres de máquinas únicos:
 - La máquina crea el nombre que desee usar.
 - Hace una petición para ver si hay conflicto.
 - Si otra máquina tiene ese nombre tendrá que pensar en otro nombre y repetir el proceso.
 - Si dos máquinas empiezan a usar el mismo nombre a la vez, aquella con menor dirección IP gana.
- Proceso
 - Todas las máquinas implementan el servidor mDNS, principalmente para servir información conocida por la propia máquina.
 - Clientes quiere resolver un nombre: petición multicast.
 - Uno o más miembros del grupo multicast contestan con la respuesta.

3.3 Asignación de direcciones IP multicast

- ▶ Las direcciones Source-Specific Multicast (SSM) son: 232.0.0.0/8
- ▶ Fuente múltiple
 - Un sistema de intercomunicación en el hogar es un ejemplo de aplicación con múltiples fuentes de *IP multicast*, pues diversos orígenes pueden estar enviando paquetes destinados a la misma dirección *IP multicast*
 - El problema se presenta cuando una dirección puede continuar siendo válida aún después de que el *host* que inició la asignación haya desaparecido del grupo, es decir haya caído o simplemente abandonado el grupo *multicast*
 - Requerimiento: Un *host* distinto del que asigna las direcciones tiene que “defender” el mantenimiento de la asignación de una dirección *multicast*
- ▶ O. Catrina et al., “*Zeroconf Multicast Address Configuration Protocol (ZMAAP)*,” Internet draft, October 2002
 - Asigna direcciones únicas y se encarga de su gestión
 - Previene la reasignación de direcciones asignadas

3.4 Descubrimiento de servicios

- ▶ En general reciben el nombre de Service Discovery Protocols (SDPs)
- ▶ Servicio de descubrimiento
 - Los protocolos de este servicio permiten a los usuarios seleccionar servicios y/o hosts por un nombre que es descubierto dinámicamente, mucho mejor que por un nombre y tipo que el usuario debiera conocer anticipadamente
- ▶ SLP (Service Location Protocol)
 - Por el IETF srvloc Working Group

Service Location Protocol

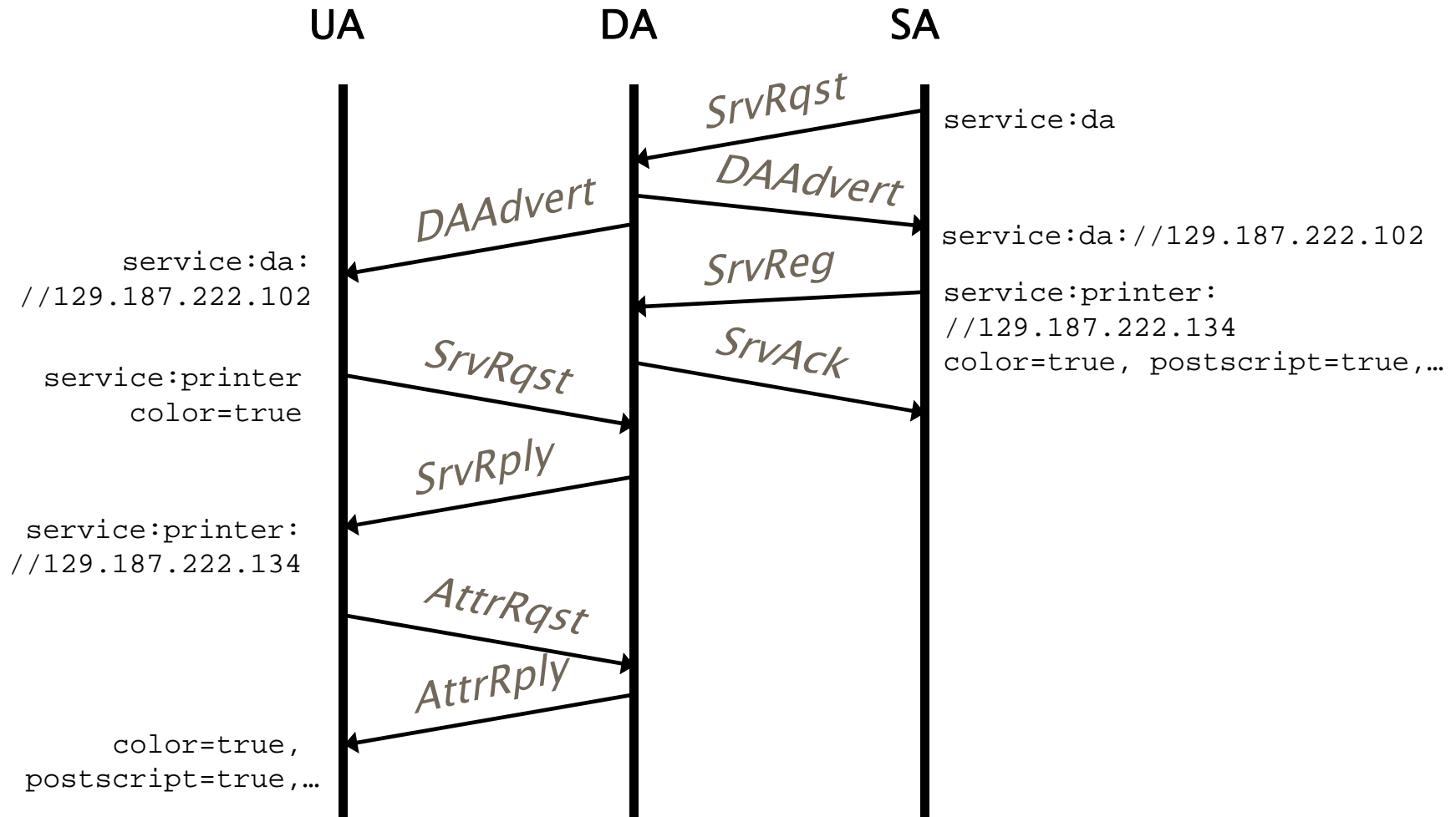
Componentes principales

- User Agent (UA)
 - Descubre los servicios que los dispositivos que representa están solicitando.
- Service Agent (SA)
 - Anuncia los servicios que él representa.
- Directory Agent (DA)
 - Acumula servicios de información de SA
 - Responde a peticiones de servicio de UA

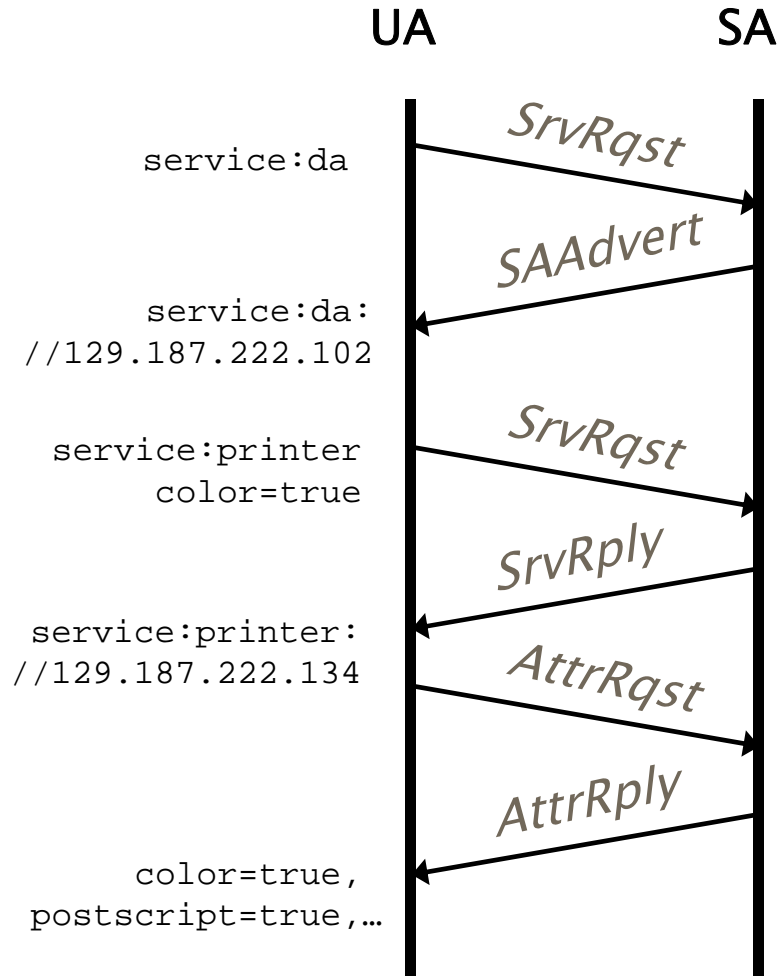
Implementación, dos variantes

- Con DA
- Sin DA

Service Location Protocol con DA

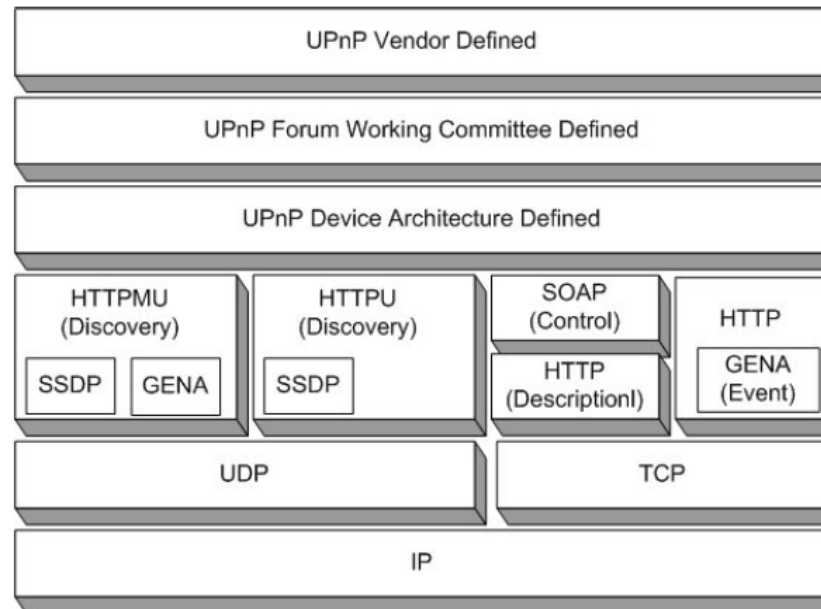


Service Location Protocol sin DA

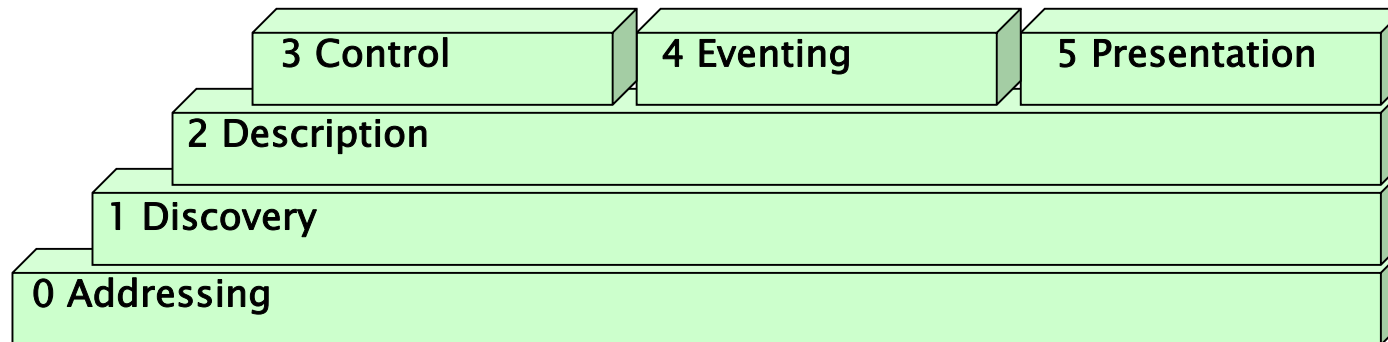


4. Universal Plug and Play

- ▶ Universal Plug and Play (UPnP)
 - Creado por el UPnP Forum, liderado por Microsoft
 - Pensado para entornos domésticos
 - No necesita de device drivers modificados
 - Independiente de la tecnología del medio (Ethernet, USB, Firewire)
 - Utiliza un protocolo HTTP sobre UDP con SOAP (web services)
 - Puerto UDP 1900 y el puerto TCP 2869



Universal Plug and Play



- ▶ Asignación de direcciones
 - Utiliza el AutoIP que está basado en el APIPA de zeroconf
- ▶ Componentes principales
 - Dispositivo (logical device)
 - Provee uno o más servicios.
 - Servicio (logical functional unit)
 - Exporta acciones y modelos del servicio junto a variables de estado del dispositivo.
 - Punto de control
 - Busca dispositivos

Universal Plug and Play

- ▶ Mucho más rígido que Zeroconf
 - Definición más rígida de los dispositivos y servicios compatibles
 - Necesidad de pasar una certificación de los equipos
 - Royalties por uso

Universal Plug and Play Networking

Paso 1. Discovery

Paso 2. Description

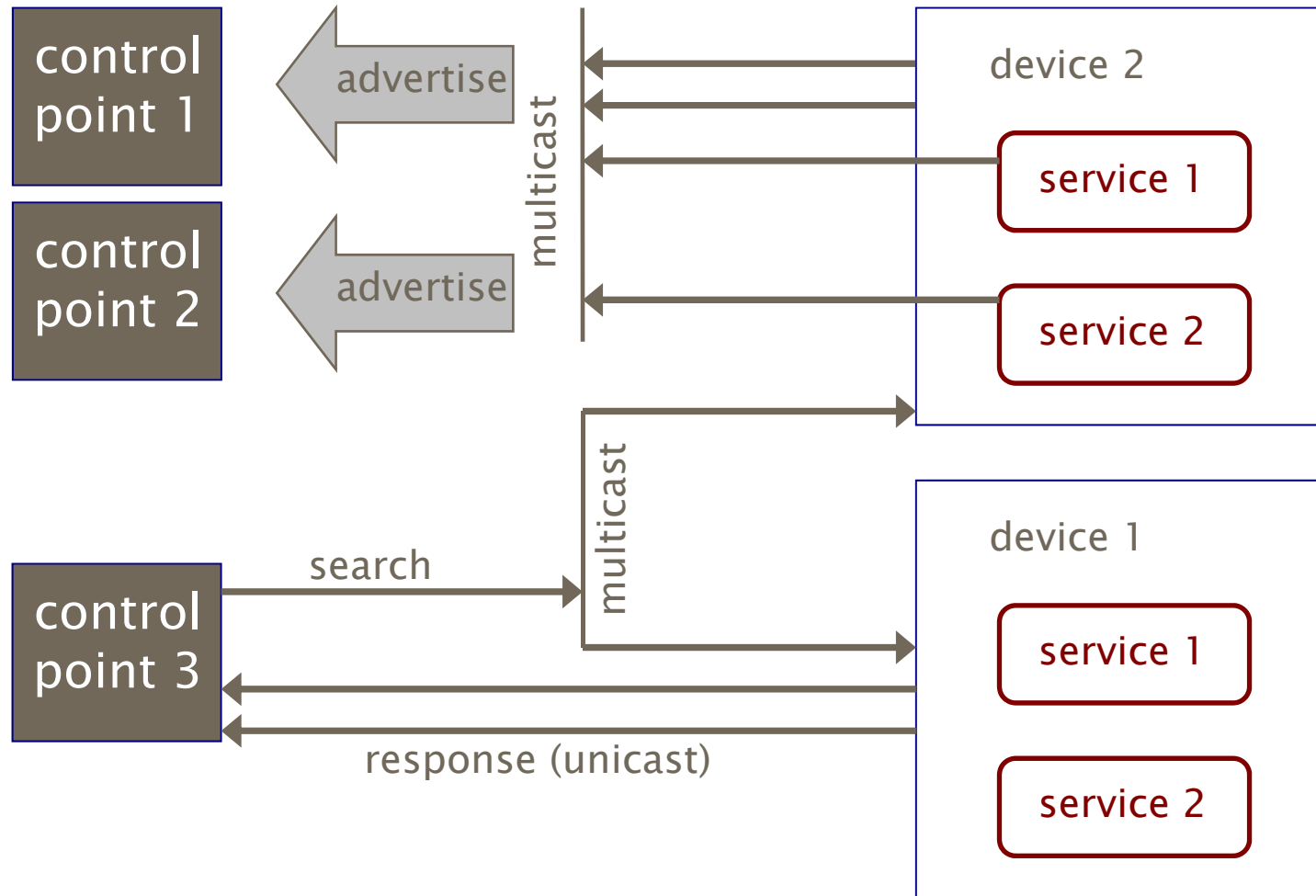
Paso 3. Control

Paso 4. Eventing

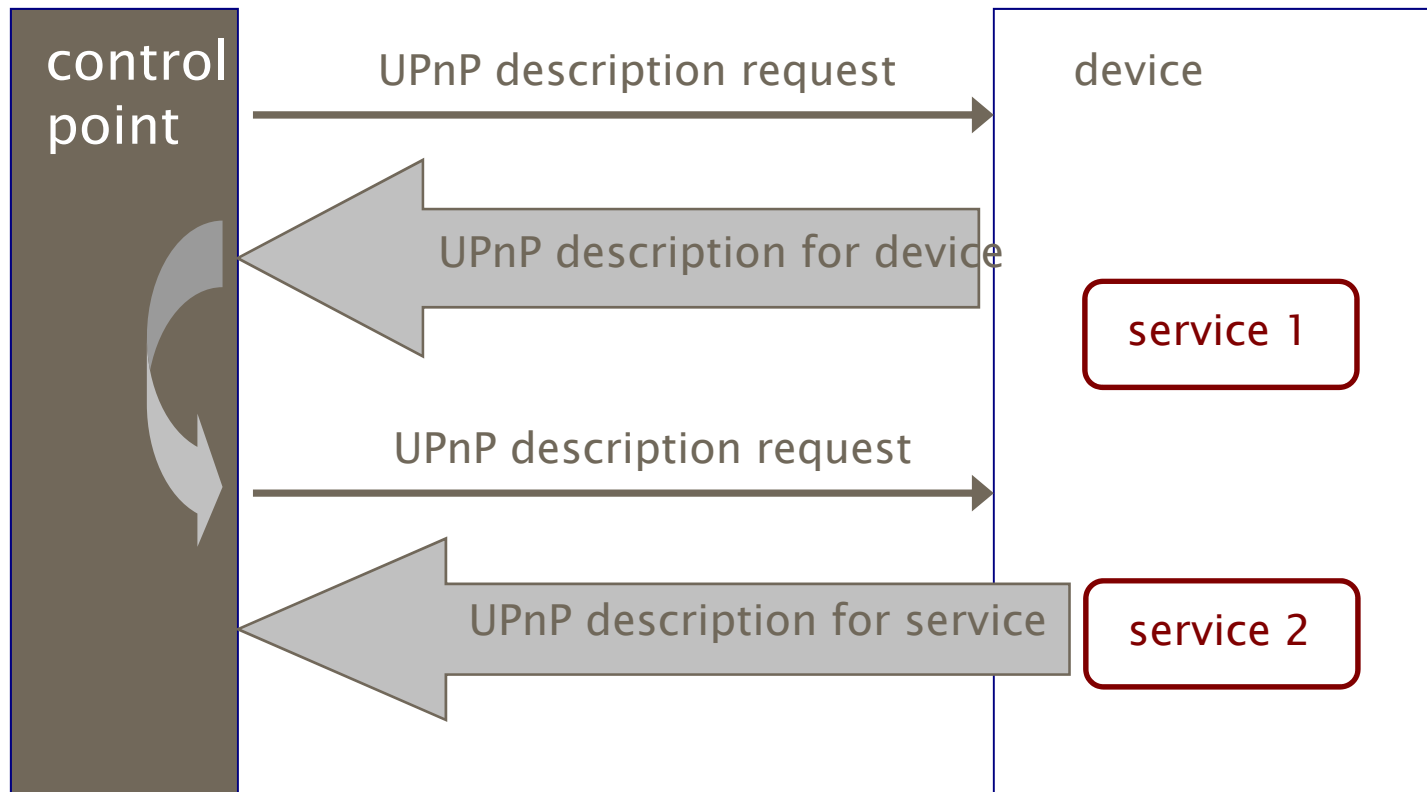
Paso 5. Presentation

Universal Plug and Play

Step 1. Discovery

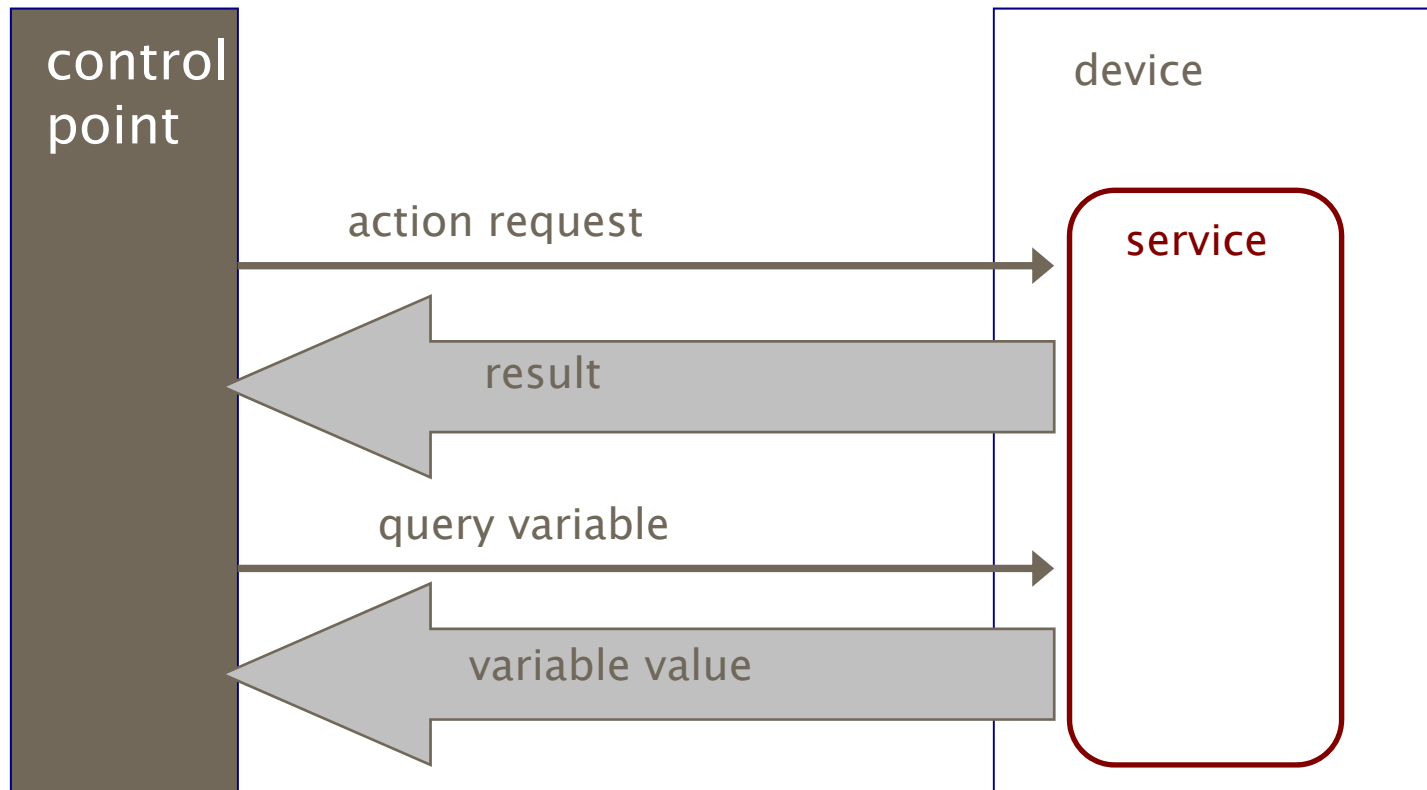


Universal Plug and Play Step 2. Description



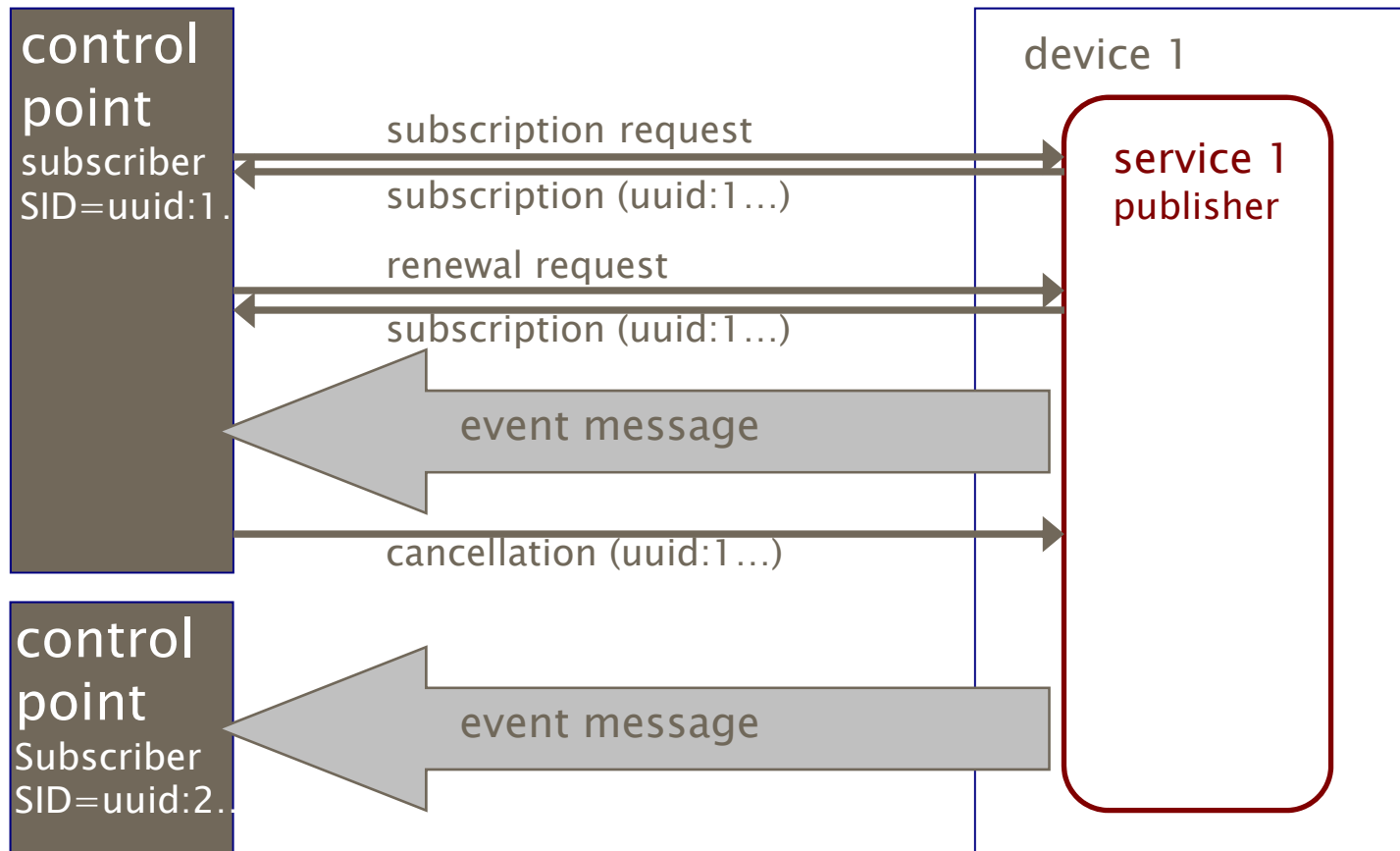
Universal Plug and Play

Step 3. Control



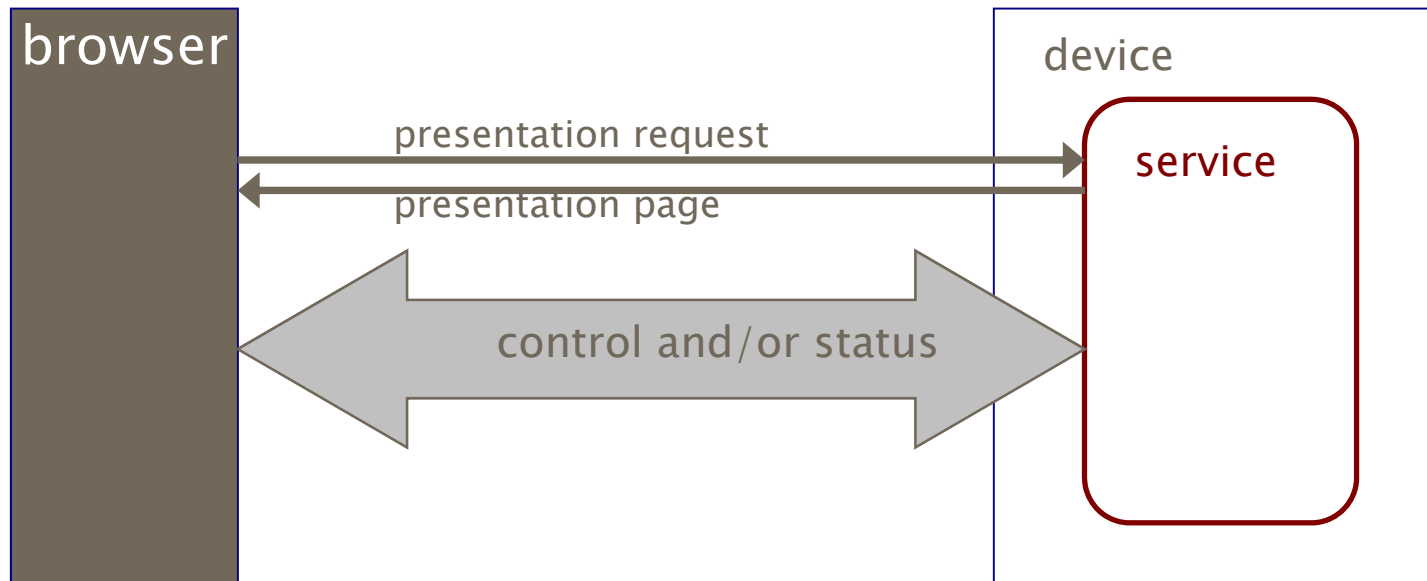
Universal Plug and Play

Step 4. Eventing



Universal Plug and Play

Step 5. Presentation



5. Jini

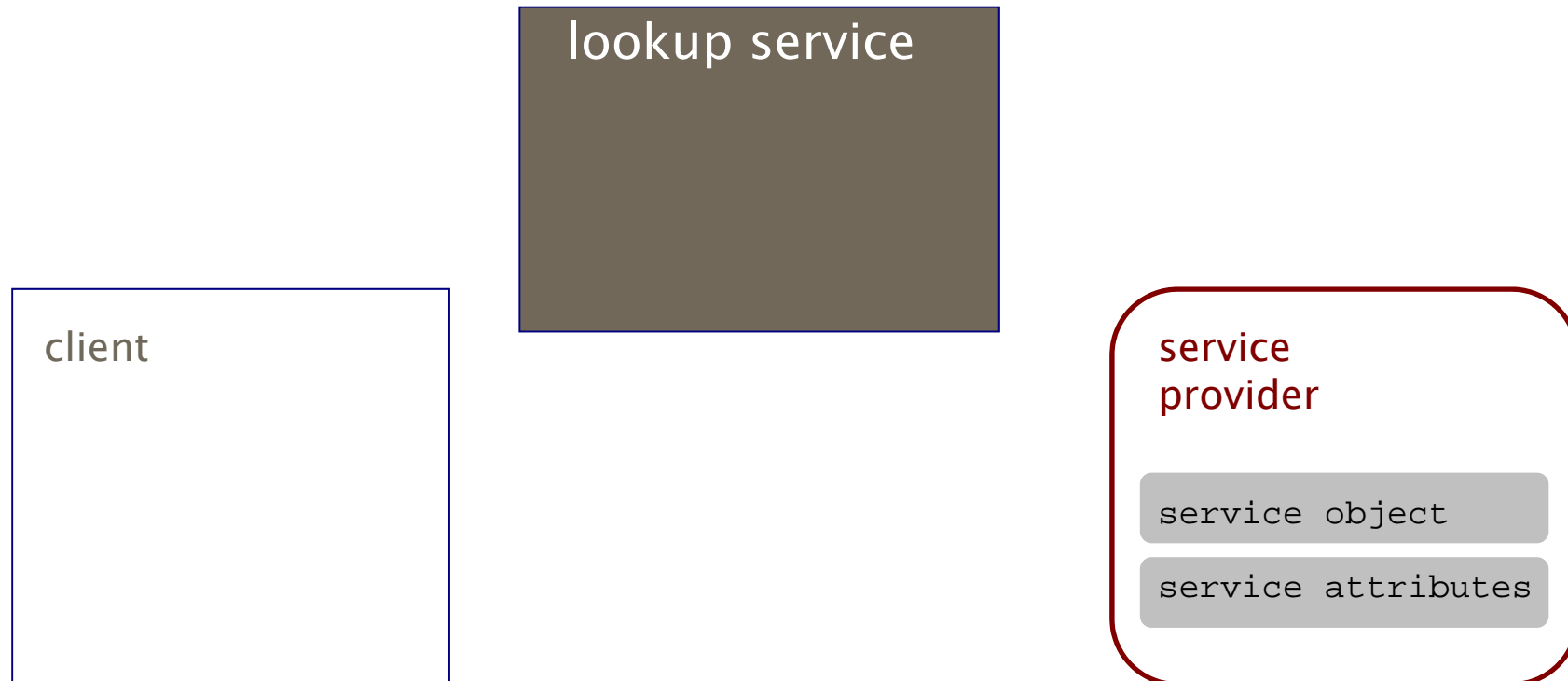
- ▶ Basado en RMI (Java Remote Invocation Method)
 - Propuesto por Sun Microsystems
- ▶ Servicios
 - Dispositivos (impresoras, pantallas, discos)
 - Software (aplicaciones, utilidades)
 - Information (base de datos, ficheros)
 - Usuarios del sistema.
- ▶ Servicio de búsqueda
 - Mapea los interfaces indicando la funcionalidad de los grupos de objetos que implementan el servicios.

Jini

- ▶ Objeto proxy de servicio
 - Puede ser una implementación de un servicio completo.
 - Una vez se localiza un servicio, su objeto proxy se sube al servicio de búsqueda.
 - Objetos cliente contactan al servicio de búsqueda para bajarse el objeto proxy.
- ▶ Eventos
 - Objetos se registran ante otros objetos para obtener notificaciones.
 - Cuando los servicios aparecen o desaparecen, se señala mediante eventos.

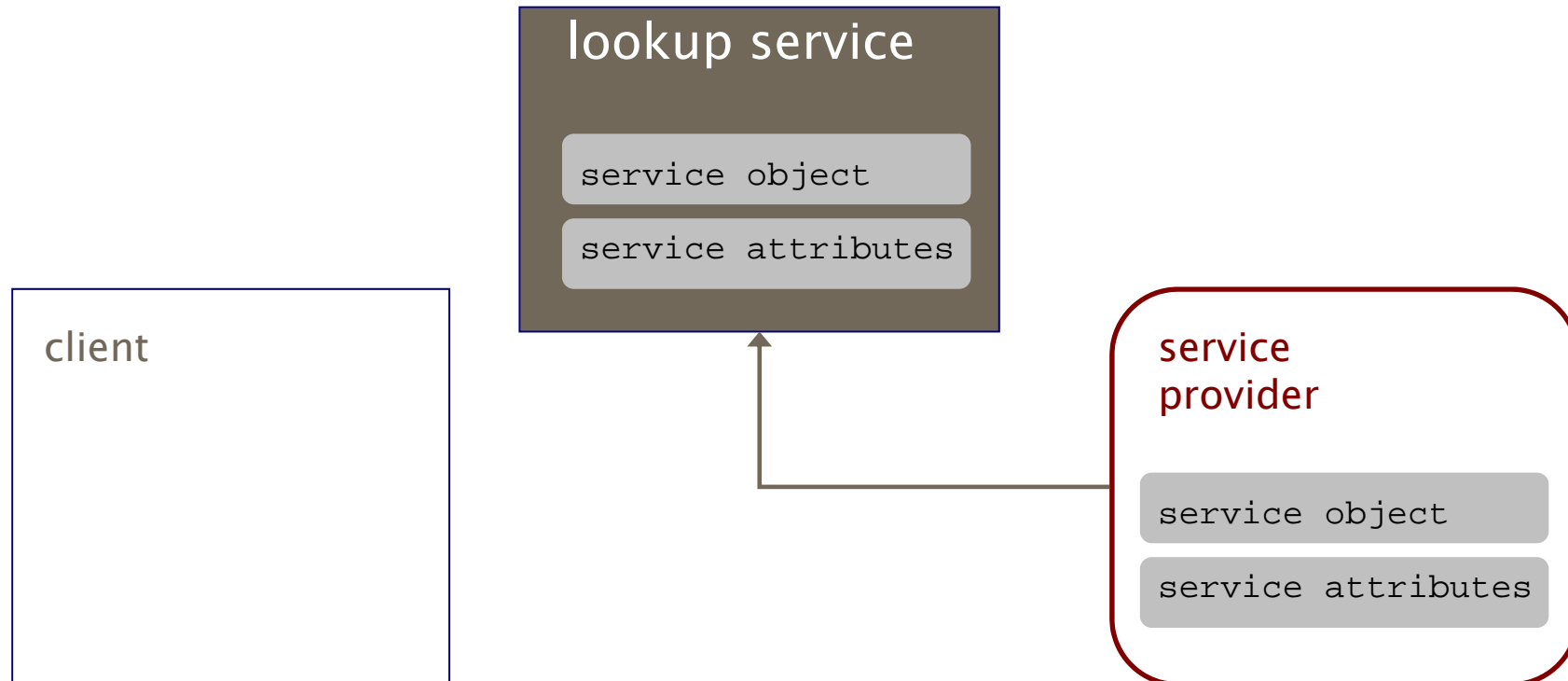
Jini Discovery

A service provider seeks a lookup service.



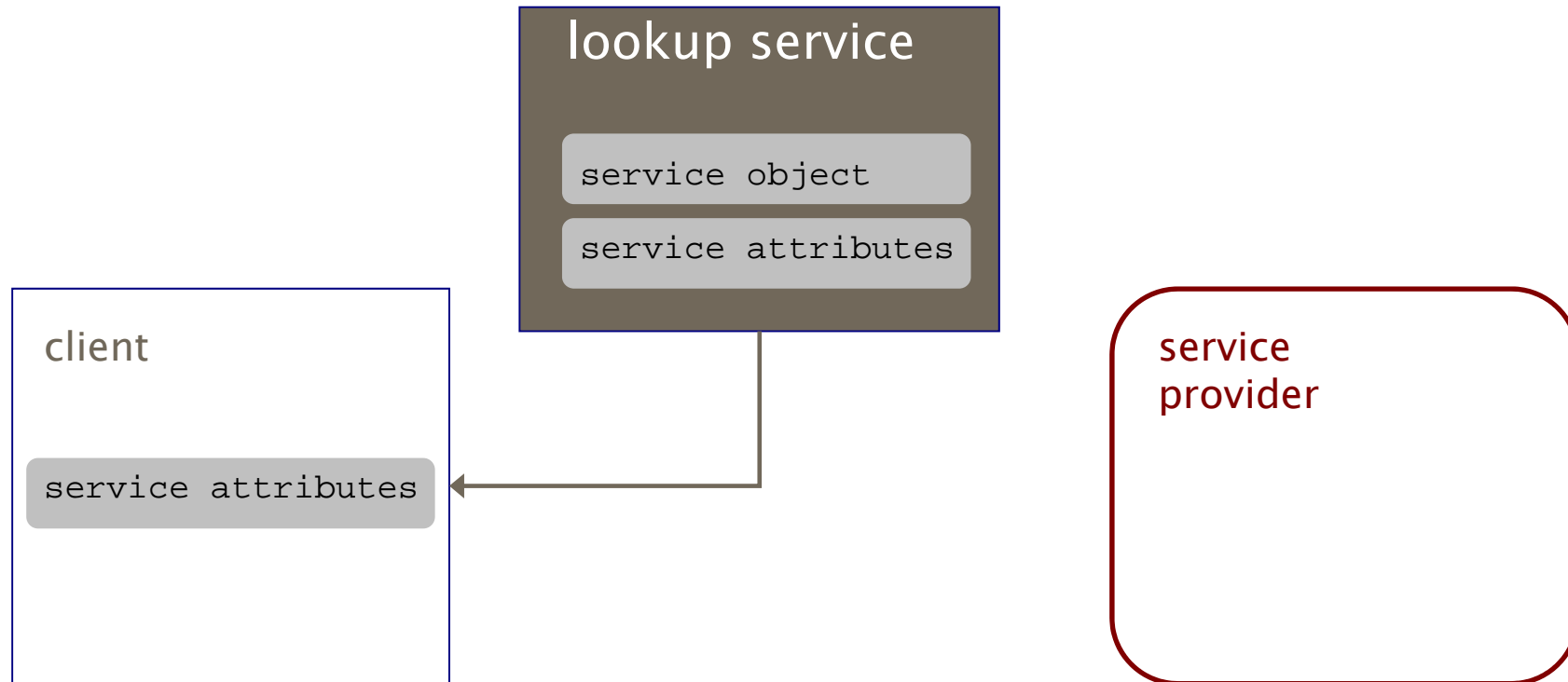
Jini Join

A service provider registers a service object (proxy) and its service attributes with the lookup service.



Jini Lookup

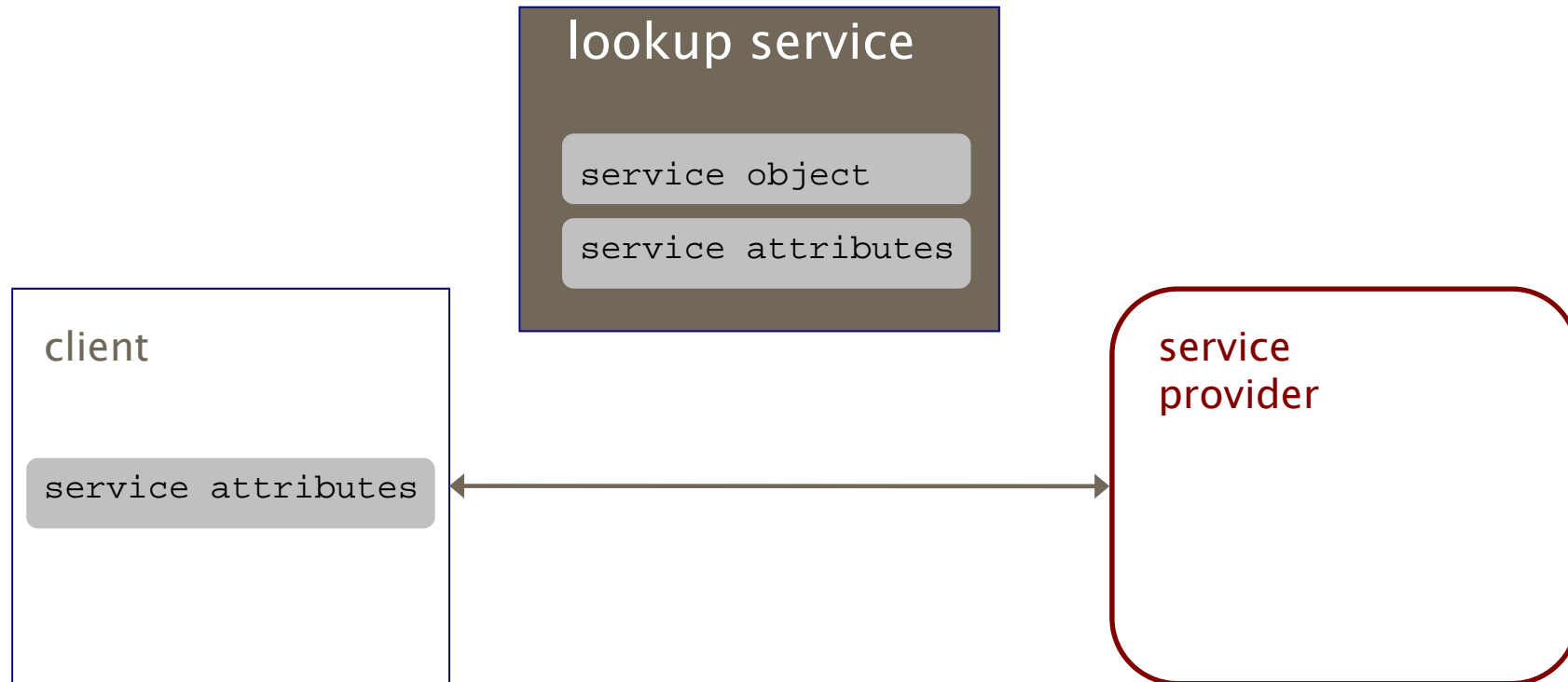
A client requests a service. A service object copy is moved to the client and used by the client to talk to the service.



Jini

Client Uses Service

The client interacts directly with the service provider via the service proxy object.



Referencias

- ▶ [Forouzan]
 - Capítulo 7
 - Capítulo 17
- ▶ [Stevens]
 - Capítulo 4
 - Capítulo 5
 - Capítulo 16
- ▶ Zeroconf
 - <http://developer.apple.com/networking/bonjour/specs.html>
 - <http://www.zeroconf.org>
- ▶ UPnP
 - <http://www.upnp.org/>