

Lehrstuhl für Mobile und Verteilte Systeme

Übung Verteilte Systeme/ Ubiquitous Computing

Übungsblatt 5

Johannes Martens



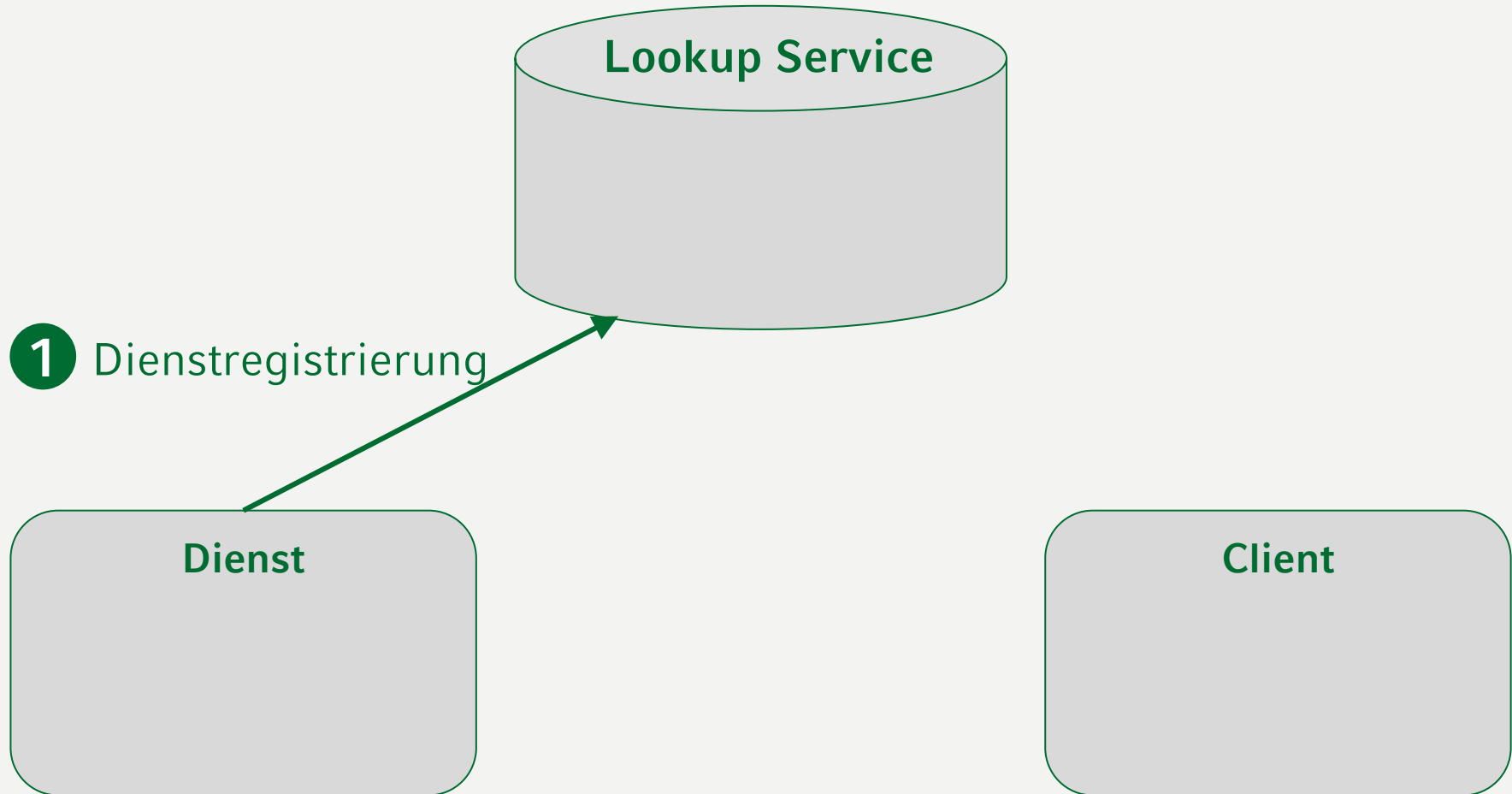
Aufgabe 15: Dienstvermittlungssysteme in der Praxis



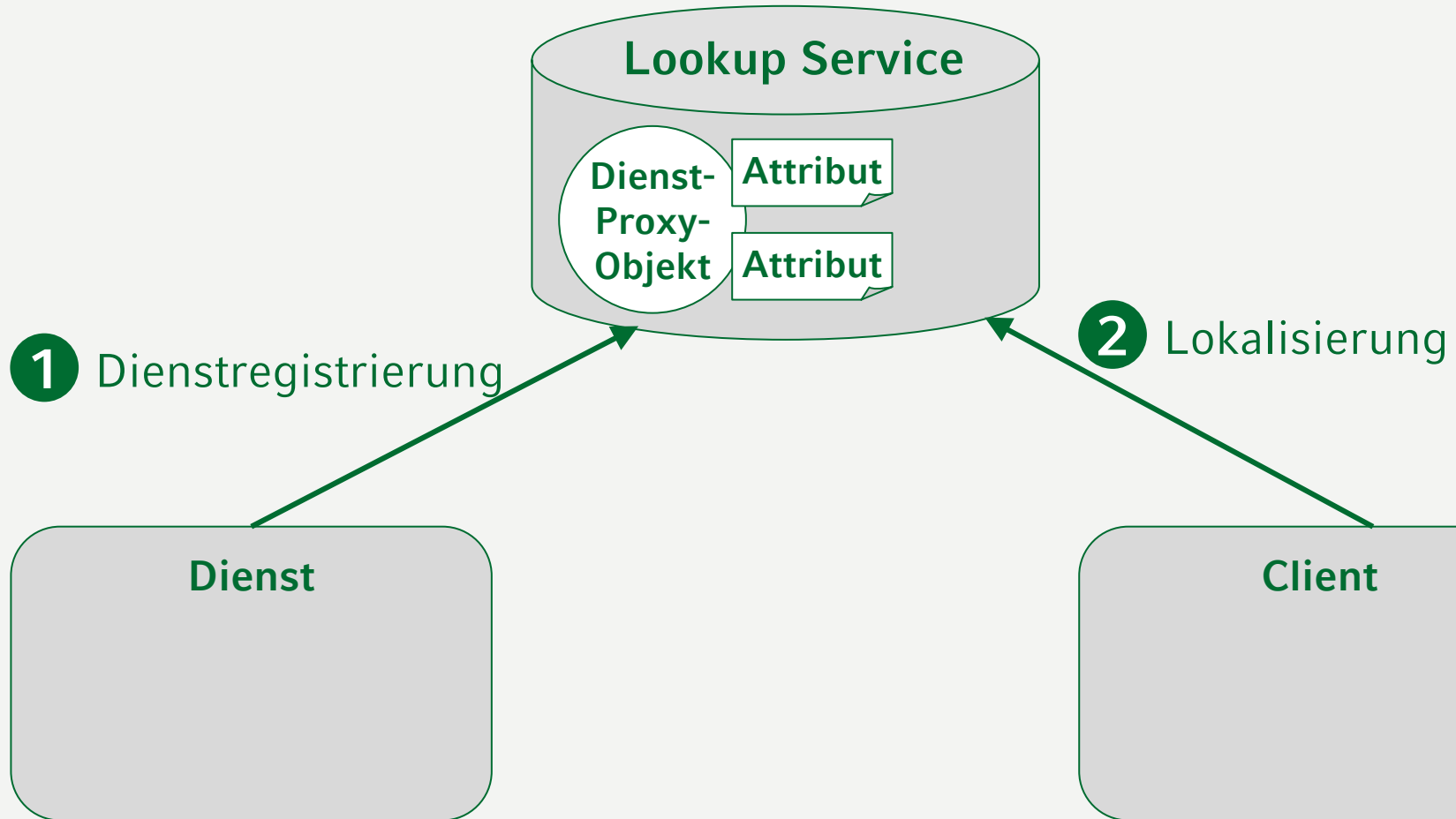
- Entwicklung der Firma Sun Microsystems
- Basiert auf Java Technologie
 - Plattformunabhängigkeit
- Features
 - Transaktionsmanagement
 - Persistenzverwaltung
 - Dienstvermittlung



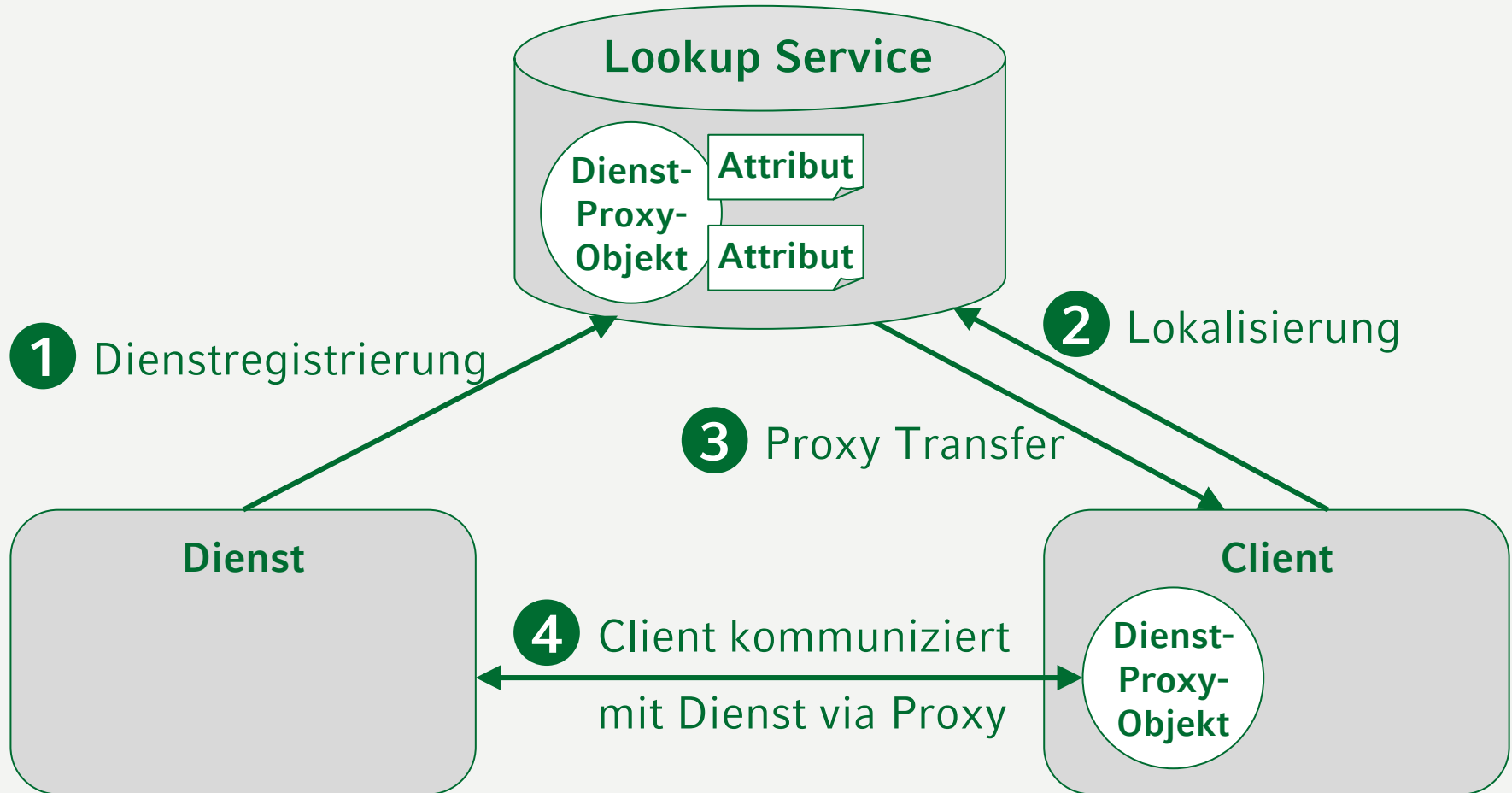
- Komponententypen eines Jini-Systems
 - Clients
 - Dienstinstanzen
 - Lookup Service (LUS)
- Netzwerkprotokoll
 - Keine Einschränkung durch Jini Spezifikation
 - Gegenwärtig nur Implementierung für TCP/IP
 - Voll konfiguriertes Netzwerk wird benötigt (im Gegensatz zu UPnP)
- Kommunikation erfolgt über RMI



- Lokalisierung eines zuständigen Lookup Services (LUS)
 - meist via Multicast (Discovery-Nachrichten)
- LUS sendet Objektreferenz (*ServiceRegistrar*)
 - Implementiert Managementschnittstelle für Registrierung von Diensten
 - Auch als „*Discovery*“-Prozess im Rahmen von Jini bezeichnet
- Dienstregistrierung durch Dienstinstanz
 - Dienst-Proxy-Objekt
 - Dienstattribute (Entries)
 - Vorgang als „*Join*“ bezeichnet



- Lokalisierung eines zuständigen Lookup Services (LUS)
 - meist via Multicast (Discovery-Nachrichten)
- Anforderungen mittels *ServiceTemplates* konkretisiert
 - Typ eines Dienstes
 - Menge von Attributen



- Ziel: Aktualität des Dienstangebots zu gewährleisten
- Verfahren:
 - Referenzen auf Objekte nur für eine vordefinierte Zeitspanne gültig
 - Objektreferenzen auch als *Leases* bezeichnet
- Auswirkungen:
 - Leases müssen periodisch verlängert werden
 - Nichtverfügbarkeit eines Dienstes wird automatisch erkannt und Dienst wird aus Angebot entfernt
 - Leases können auch vor Ablauf als ungültig markiert werden

- Spezifiziert durch das UPnP-Forum
 - Ursprünglich von Microsoft initiiert
 - Mittlerweile knapp 800 Mitglieder
- Konzept:
 - Kontrollpunkt (z. B. Mobiles Endgerät) kann Dienste von UPnP-Geräten (z. B. Drucker) ohne Interaktion des Benutzers auffinden und nutzen.
- Zentrale Merkmale von UPnP
 - IP-basierte Kommunikation
 - Betriebssystem- und Programmiersprachenunabhängig
 - Zentrale oder dezentrale Kontrolle möglich
 - herstellerspezifische Erweiterungen möglich



0. Adressierung (Addressing)

- Gültige IP-Adresse für spätere Schritte benötigt
- via DHCP oder AUTO-IP/Zeroconf

1. Lokalisierung (Discovery)

- Meldung an Kontrollpunkte via UDP
 - Multicast-Adresse: 239.255.255.250:1900
 - Kontrollpunkte können auch nach UPnP-Geräten suchen
- Simple Service Discovery Protocol (SSDP)
 - HTTP-basiertes Dienstvermittlungssystem
 - Passende Geräte antworten mit ihrer URL
 - Leasing-Konzept (analog zu Jini)
- Discovery-Message
 - Wichtigste Angaben zum Gerät und Dienst (Namen, Typ, ...)
 - URL zur genaueren Beschreibung des Geräts/Diensts

2. Beschreibung (Description)

- Kontrollpunkt ruft Beschreibung des Geräts per HTTP über TCP/IP von der URL ab
- Beschreibung in Form eines XML-Dokument
 - Informationen über den Hersteller
 - Seriennummer
 - URL-Adressen für die Steuerung, Ereignisse und die Präsentation
 - Spezifikation von Kommandos und Aktionen sowie Datentypen und Datenbereiche

3. Steuerung (Control)

- Über SOAP-Messages an Steuerungs-URL

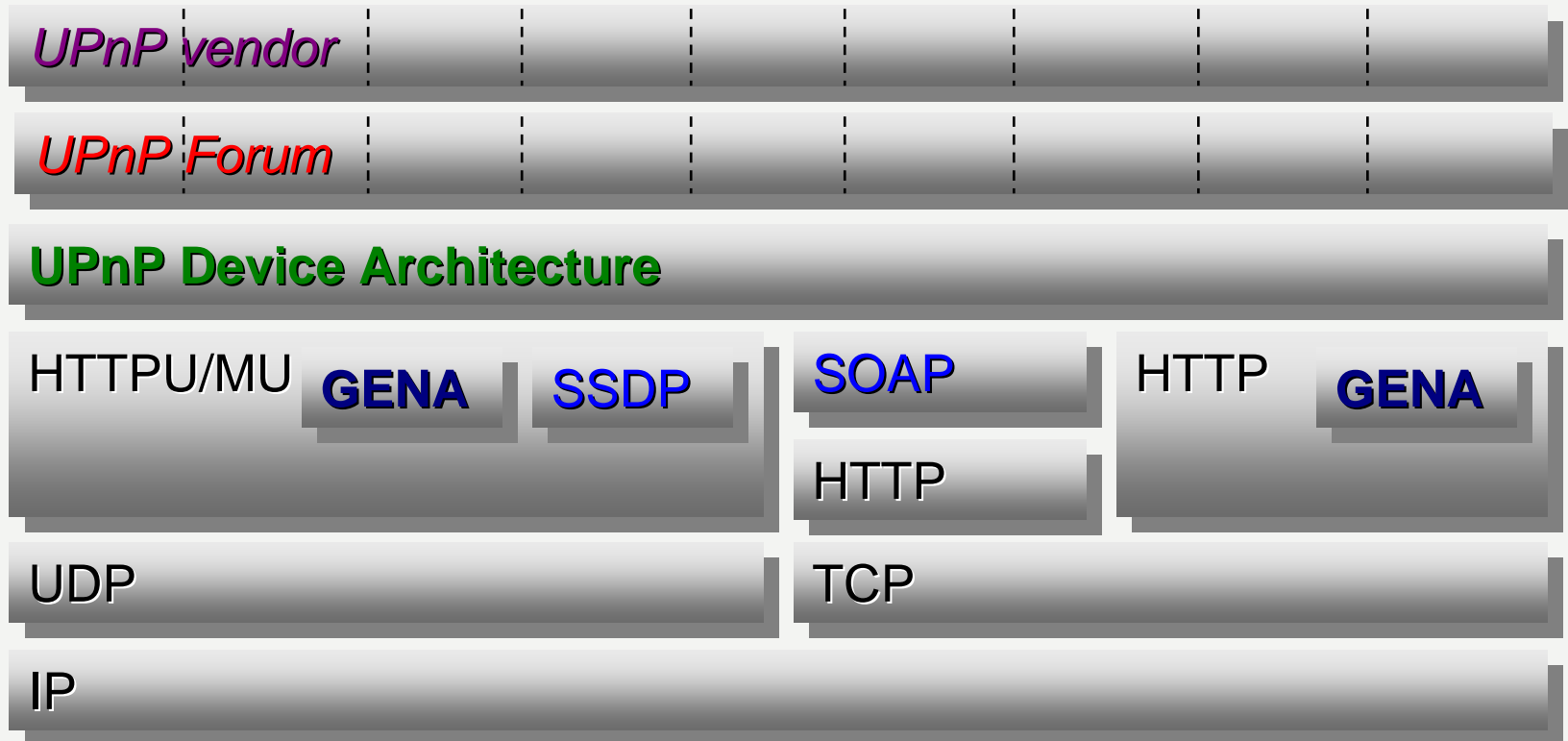
4. Ereignismeldungen (Event Notification)

- Ziel: Push-Kommunikation ermöglichen
- General Event Notification Architecture (GENA)
 - XML-basiert
 - Ermöglicht Abonnements von Statusvariablen
 - Bei Änderung: Verschicken einer „event message“ mit dem aktuellen Zustand der Statusvariable

5. Präsentation (Presentation)

- Alternative zur Steuerung und Ereignismeldungen
- Ziel: alternative Benutzeroberfläche neben dem standardisierten Zugriff via UPnP
- Zugriff mittels Webbrowser über die presentation URL

Vendor-specific API above



Vendor-specific OS below

- Spezifiziert durch die Internet Engineering Taskforce (IETF)
 - Drei Versionen: SLP, SLPv2 und SLPv3
 - RFC 2608, RFC 2609, RFC 2165, RFC 224
- Offener Standard zur Beschreibung von
 - Diensten
 - Protokollen zwischen Systemkomponenten
- System- und Programmiersprachenunabhängig (analog zu UPnP)
- Praktischer Einsatz
 - Novell Netware
 - CUPS

- User Agent (UA)
 - Realisieren Dienstsuche im Auftrag eines Clients
 - Multicast-basierender Discovery-Prozess
 - Lokalisieren Directory Agent
- Service Agent (SA)
 - Verbreiten Nachricht über Existenz eines Dienstes
 - Vermittelt Dienstbeschreibung
 - Registriert und deregistriert Dienste bei DA
- Directory Agent (DA)
 - Funktionalität eines zentralen Dienstverzeichnisses
 - Mehrere DAs möglich (Replikation sinnvoll)
 - Liefert Liste von Dienstreferenzen mit deren URL (Lookup-Prozess)

- Basiert auf IP
 - paket-orientiert (meist UDP, manchmal auch TCP)
 - Intensive Nutzung von Multicast (Gerätesuche und Beitreten eines neuen Netzes)
 - Wiederholung von Multicast-Nachrichten wg. Unzuverlässigkeit von UDP
 - Alle Agenten verwenden Port 427 mit UDP (SA und DA auch TCP)
- Verhalten von SLP abhängig von Existenz eines Directory Agents
 - Client sendet Multicast-Anfrage an DA nach Verbindungsherstellung mit dem Netzwerk
 - Erhält er keine Antwort, so wechselt er in „Non-DA-Mode“
 - Nachträgliches Hinzufügen eines DA möglich (Heartbeat-Multicast)
 - SAs registrieren daraufhin alle Dienste beim DA

- DA im Netz nicht vorhanden
 - UA sendet Multicast UDP Nachricht mit der Anfrage
 - Alle SAs mit passenden Diensten senden ein UDP-Paket an den UA
 - Sollte die Antwort nicht in ein UDP-Paket passen
 - UDP-Paket als „overflown“ markiert
 - UA kann Anfrage direkt per TCP an SA senden (keine Begrenzung der Nachrichtengröße)
- DA im Netz vorhanden
 - UA sendet Anfragenachricht per UDP oder TCP an DA
 - DA wertet die Anfrage bei sich aus und sendet Antwort direkt an UA zurück

- URL

Abstract Service Type

`service:printer:lpr://myprinter/myqueue`

Service Type

- Attribute

(printer-name=Hugo),
(printer-natural-language-configured=en-us),
(printer-location=In my home office),
(printer-document-format-supported=application/postscript)