

Konzepte von Betriebssystem- Komponenten Middleware



Jini

Vortrag
von
Philipp Sommer

Inhalt

1. Einführung

- Hintergrund
- Was ist Jini?
- Ziele von Jini

2. Jini Technology

- Komponenten
- Client, Server und Dienste
- Discovery und Join
- Client Lookup
- Proxies
- Leasing

3. Abschließende Bemerkung

4. Beispiel



Hintergrund

- 1994: Erster in Java geschriebene Webbrowser hat die Fähigkeit Code von ausführbaren Programmen herunterzuladen
- Idee: Diese Fähigkeit auf alle Geräte anwenden, und somit ein selbst kommunizierendes Netzwerk aufbauen
- Projekt bekommt die Bezeichnung **Jini**



Was ist Jini? Welche Ziele verfolgt Jini?

- Jini ist eine Netzwerkarchitektur, mit deren Hilfe sich Dienste finden, miteinander kommunizieren und sich gegenseitig nutzen können

Ziele:

- Einfachheit
- Zuverlässigkeit
- Skalierbarkeit



Inhalt

1. Einführung

- Hintergrund
- Was ist Jini?
- Ziele von Jini

2. Jini Technology

- Komponenten
- Client, Server und Dienste
- Discovery und Join
- Client Lookup
- Proxies
- Leasing

3. Abschließende Bemerkung

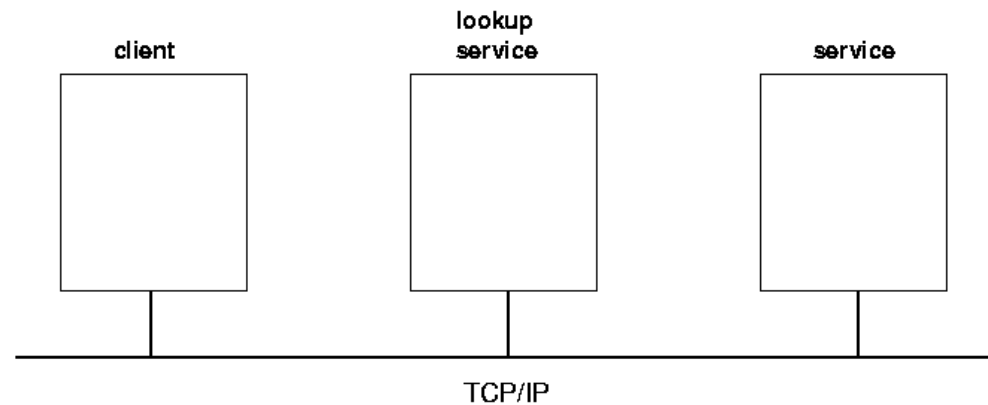
4. Beispiel



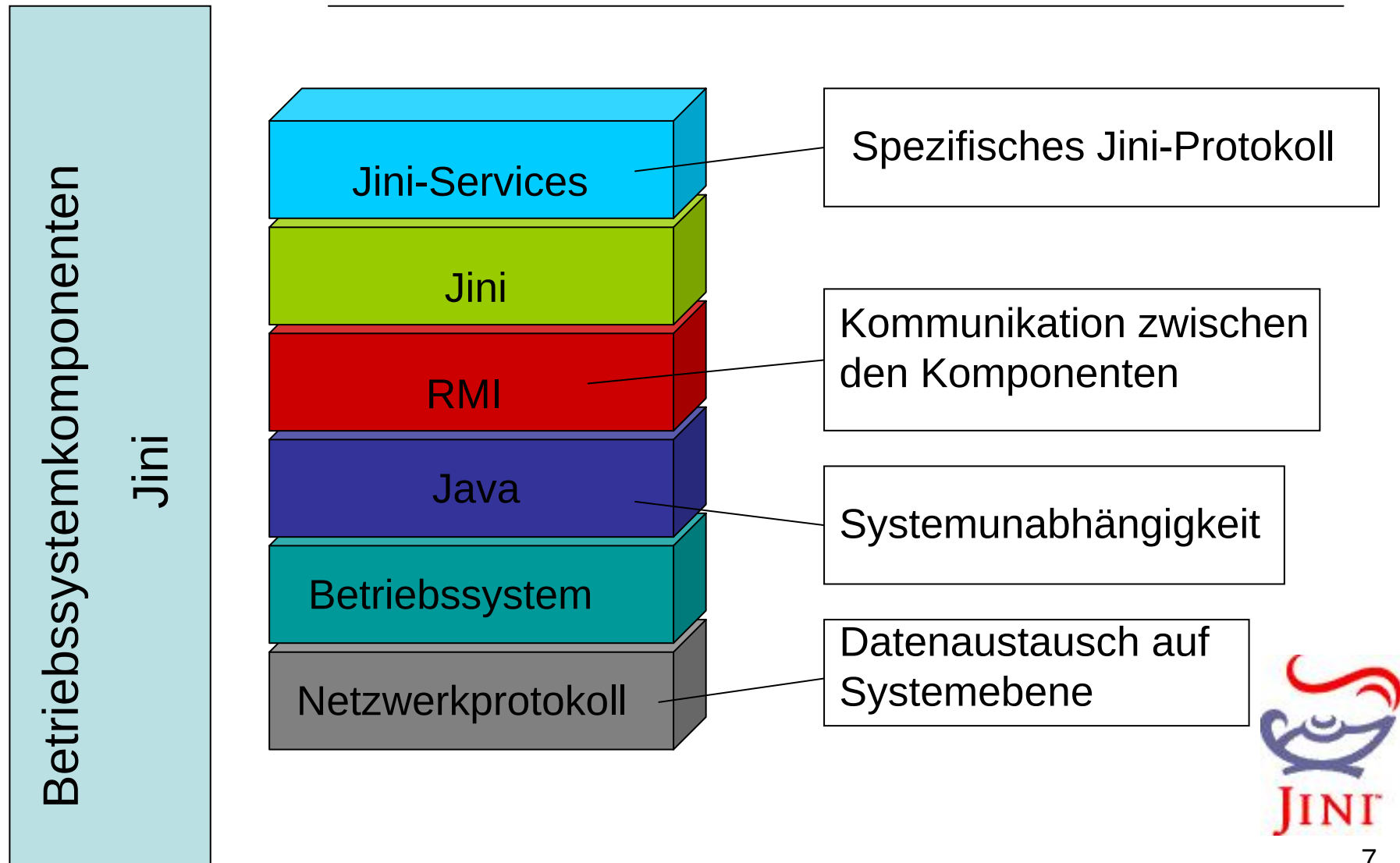
Komponenten

Drei Hauptkomponenten in einer Jini-Architektur:

- **Dienst** (z.B. Drucker): stellt allen anderen Netzwerkteilnehmern eine bestimmte Funktion zur Verfügung.
- **Client** (Dienstnutzer): nutzt einen oder mehrere Dienste eines Dienstanbieters.
- **Lookup-Service**: vermittelt zwischen Dienst und Client.



Protokollschichten



Client, Server und Dienst

Jini ist ein Framework für verteilte Computersysteme;

➔ Teilnehmer heißen Client und Server.

Aber: Server präsentieren der Außenwelt ein Interface (Dienstinterface), den sog. **Dienst**

- Dienstanbieter erzeugt ein Service-Objekt, das den Dienst implementiert
- Client nutzt den Dienst durch das Objekt



Dienstanbieter



Service-Objekt

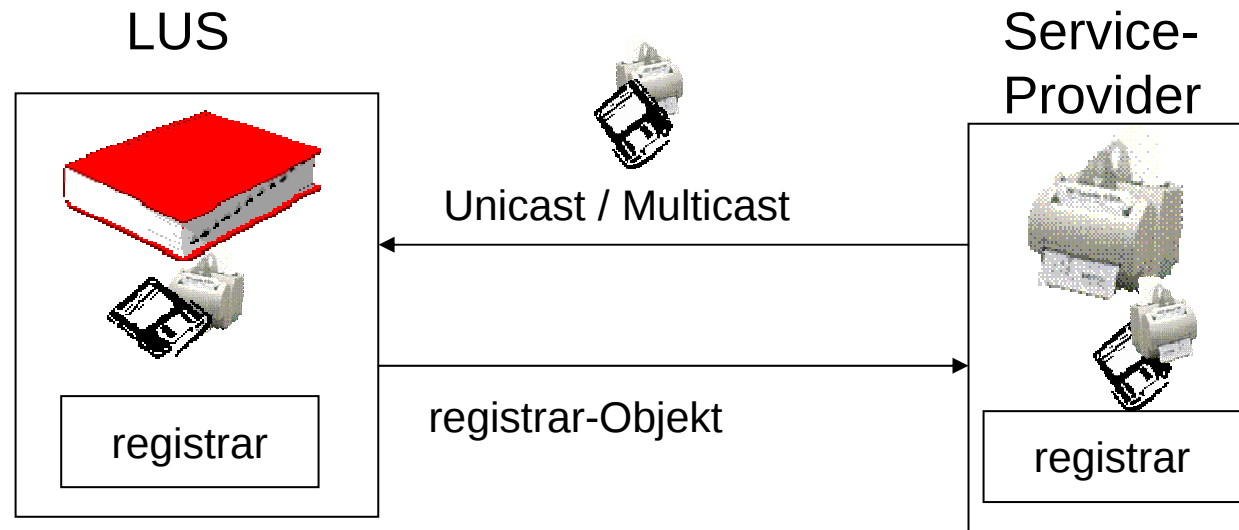


Discovery und Join

- **Discovery:** Auffinden des Lookup-Services durch den Dienst oder Client.
- **Joining:** Beitritt einer Jini-Gemeinschaft
 - **Unicast Request Protocol:**
 - Lookup-Service ist bekannt, Verbindung über TCP/IP
 - **Multicast Request Protocol:**
 - Lookup-Service ist unbekannt, Versenden von UDP Datagrammen;
 - **Multicast Announcement Protocol:**
 - Lookup-Service informiert mittels UDP alle Interessierten über sein Vorhandensein



Discovery und Join



1. Suche via Unicast oder Multicast
2. Antwort durch Senden eines registrar-Objekts
3. Hochladen des Service-Objekts

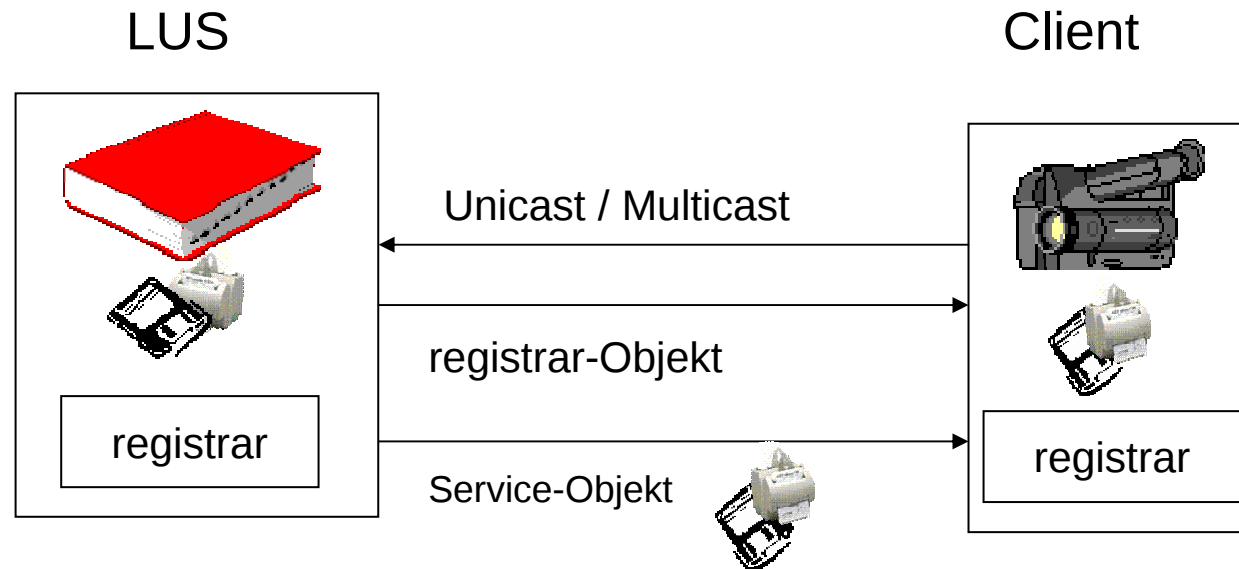


Client Lookup

- Registrierung erfolgt auf gleichem Weg, wie beim Dienstanbieter
- Client stellt Suchanfrage an den Lookup-Service
- Client versucht eine Kopie des Service-Objekts in seine JVM zu bekommen



Discovery und Join



1. Suche via Unicast oder Multicast
 2. Antwort durch Senden eines registrar-Objekts
- Dienstanbieter, Lookup-Service und Client haben das Service-Objekt in ihrer jeweiligen JVM



Proxies

Der Begriff Proxy stammt aus dem Englischen und bedeutet soviel wie Stellvertreter

- Proxies sind in der Regel in Software implementiert
- nehmen Anfragen entgegen und leiten diese weiter
- Das Service-Objekt dient dem Client also als **Proxy** zum Dienst



Proxies

Unterscheidung zwischen verschiedenen Situationen:

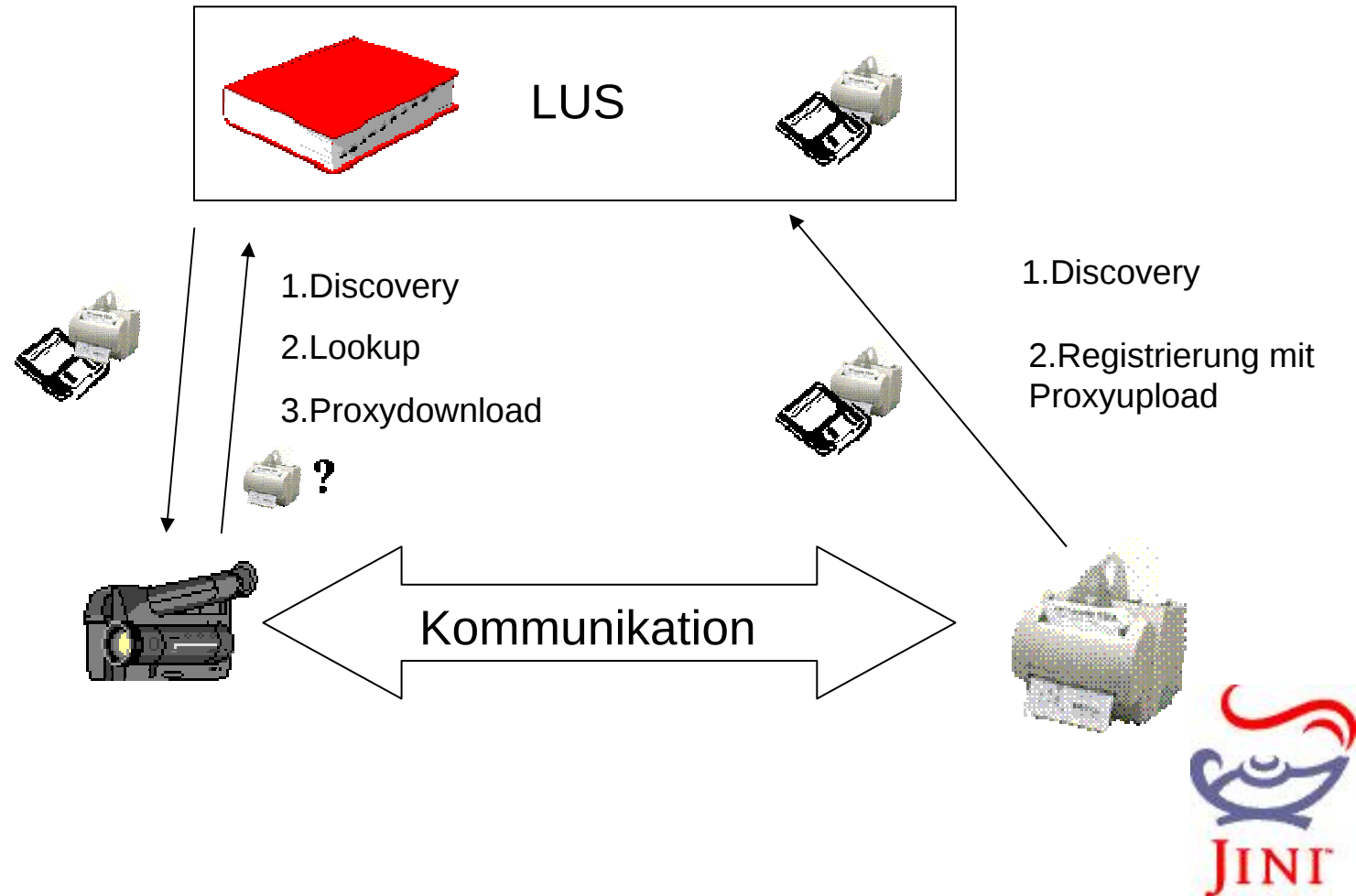
1. Proxy ist Service und wird komplett auf dem Client ausgeführt
2. Proxy ist das Interface zum Service. Er dient zur Kommunikation mit dem eigentlichen Dienst



Ablauf

Betriebssystemkomponenten

Jini



Leasing

Lease: Vertrag zwischen zwei Parteien, der die Nutzung einer Ressource zeitlich beschränkt

Ablauf:

- Lease-Bewerber (Dienst) verlangt vom Lease-Anbieter (Lookup-Service) für eine bestimmte Zeit einen Lease
- Anbieter bestimmt ob und wie lange er einen Lease gewähren will
- Anbieter schickt den Bewerber ein Lease Objekt
- Bewerber kann Lease abbrechen, erneuern oder verfallen lassen



Abschließende Bewertung

- Jini ist nur eine von vielen Architekturen (z.B. CORBA, T-Spaces und UPnP), bietet aber gegenüber anderen einige Vorteile:
 - keine Treiberinstallation
 - „selbstheilendes“ System
 - elegant für spontane Netzwerke

