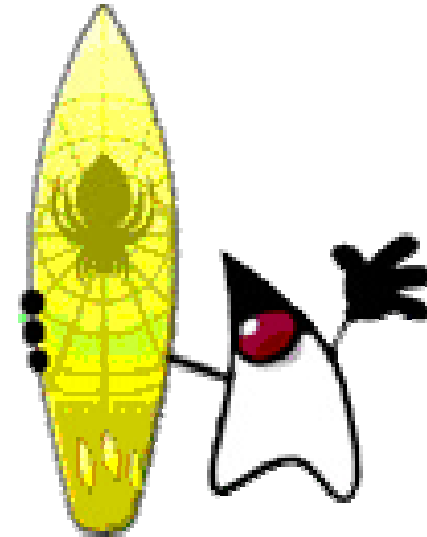


Proseminar „Mobil Computing“

Java Intelligent Network Infrastructure

06.07.2000

Bernd Dutkowski
bernd@dutkowski.net

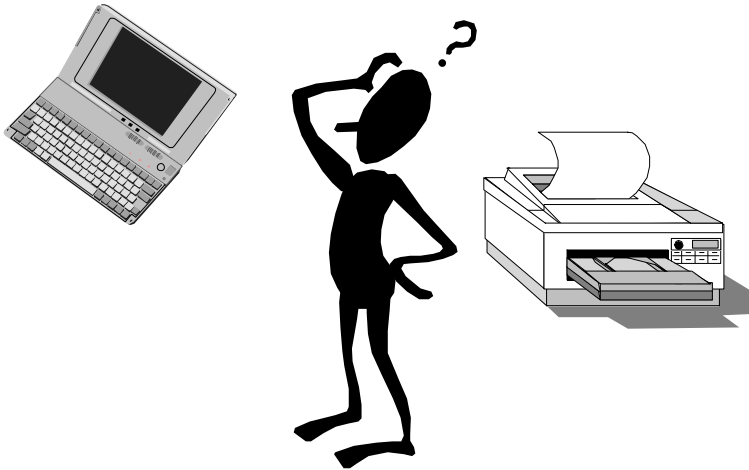


Übersicht

- Szenarien
- Probleme die durch JINI gelöst werden sollen
- Aufbau und Funktionsweise
- Probleme
- UPnP



Szenario: Laptop und Drucker



Probleme:

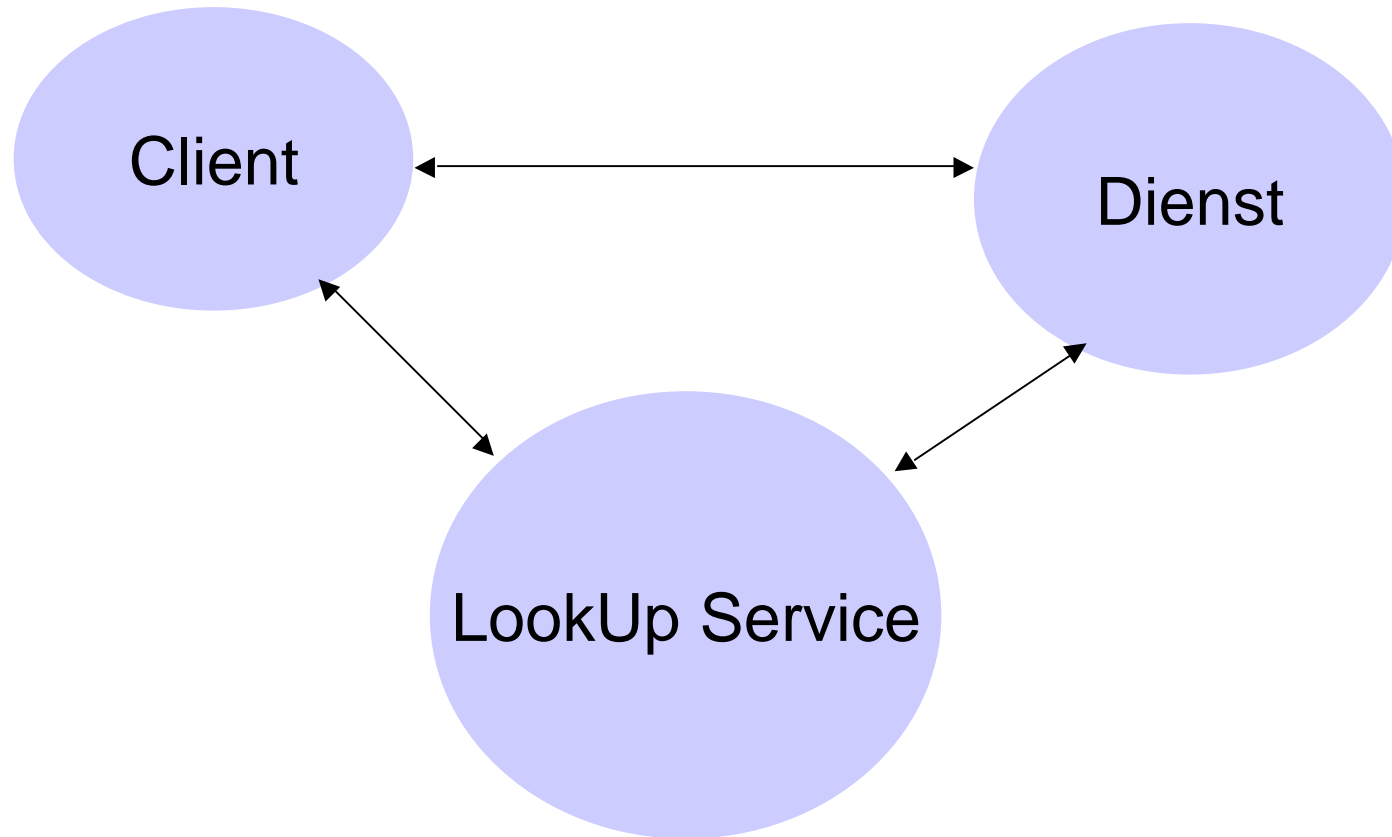
- Treiber
- Administration
- Wo ist welcher Dienst
- Sicherheit

Szenerien

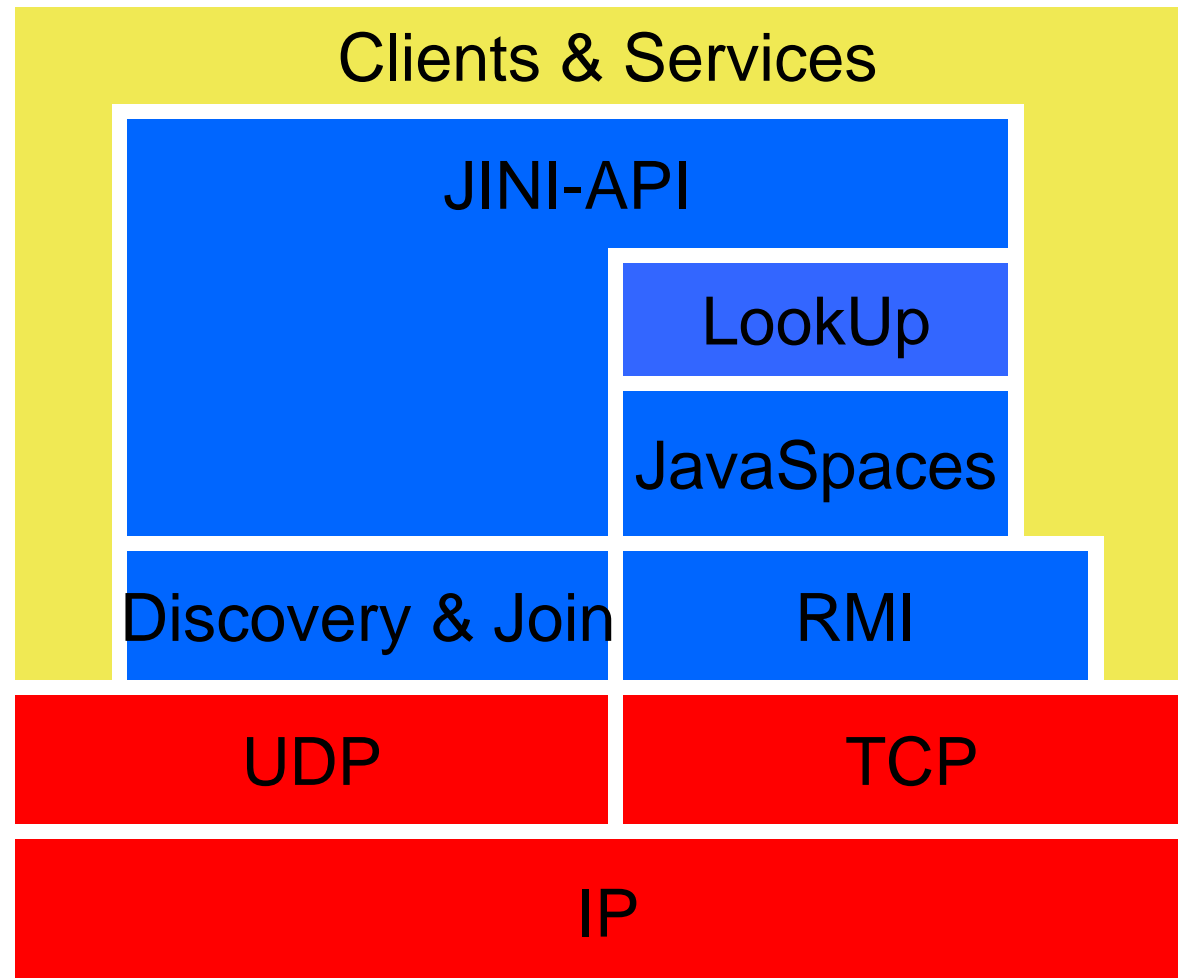
- Drucker und Laptop
- Computer Farm
- LAN Party
- JINI-Kaffeemaschine



JINI Universum

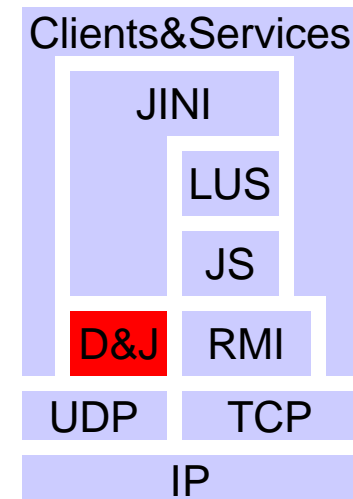


Schichtmodell



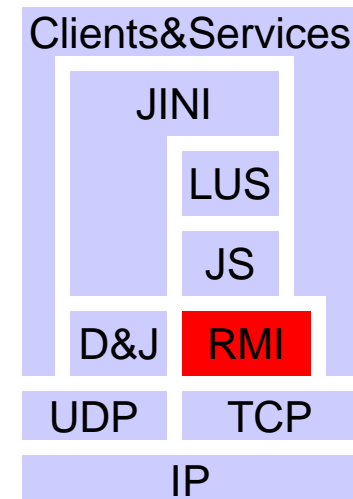
Discovery & Join

- Multicast Request Protocol
 - Client oder Dienst suchen per Broadcast LUS
- Multicast Announcement Protocol
 - LUS meldet seine Verfügbarkeit
- Unicast Discovery Protocol
 - wie „Multicast Request Protocol“
 - LUS-IP ist aber bekannt



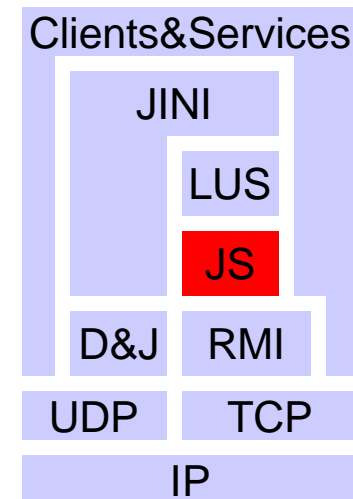
Remote Methode Invocation (RMI)

- Entfernter Methodenaufruf
- Transport von Code und Daten



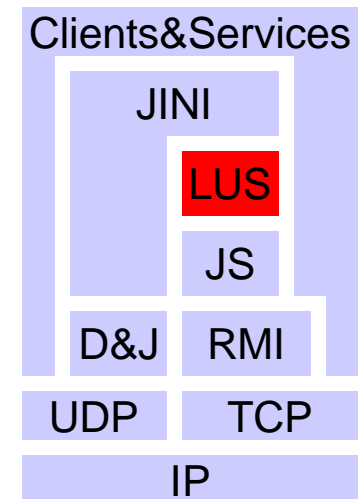
JavaSpaces

- Objekt-Peristenz-Service
- basiert auf der Idee der „Tupelsysteme“
- einfaches Interface mit 4 Operationen
 - (write(), notify(), read(), take())
- Leases
- transaktionsfähig



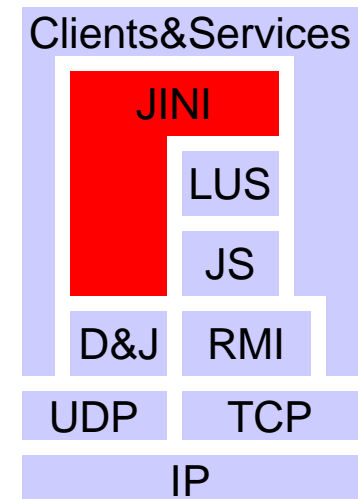
Lookup Service (LUS)

- Dienste melden sich hier an
- Clients können nach Diensten suchen



JINI-API

- Kapselt die darunter liegenden Module



Aufbau eines JINI-Dienstes

Schnittstelle

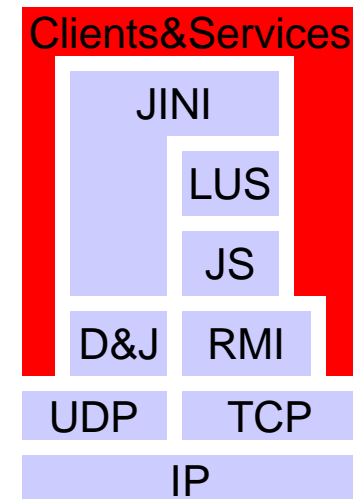
```
interface ...  
{  
    ...  
}
```

Implementierung

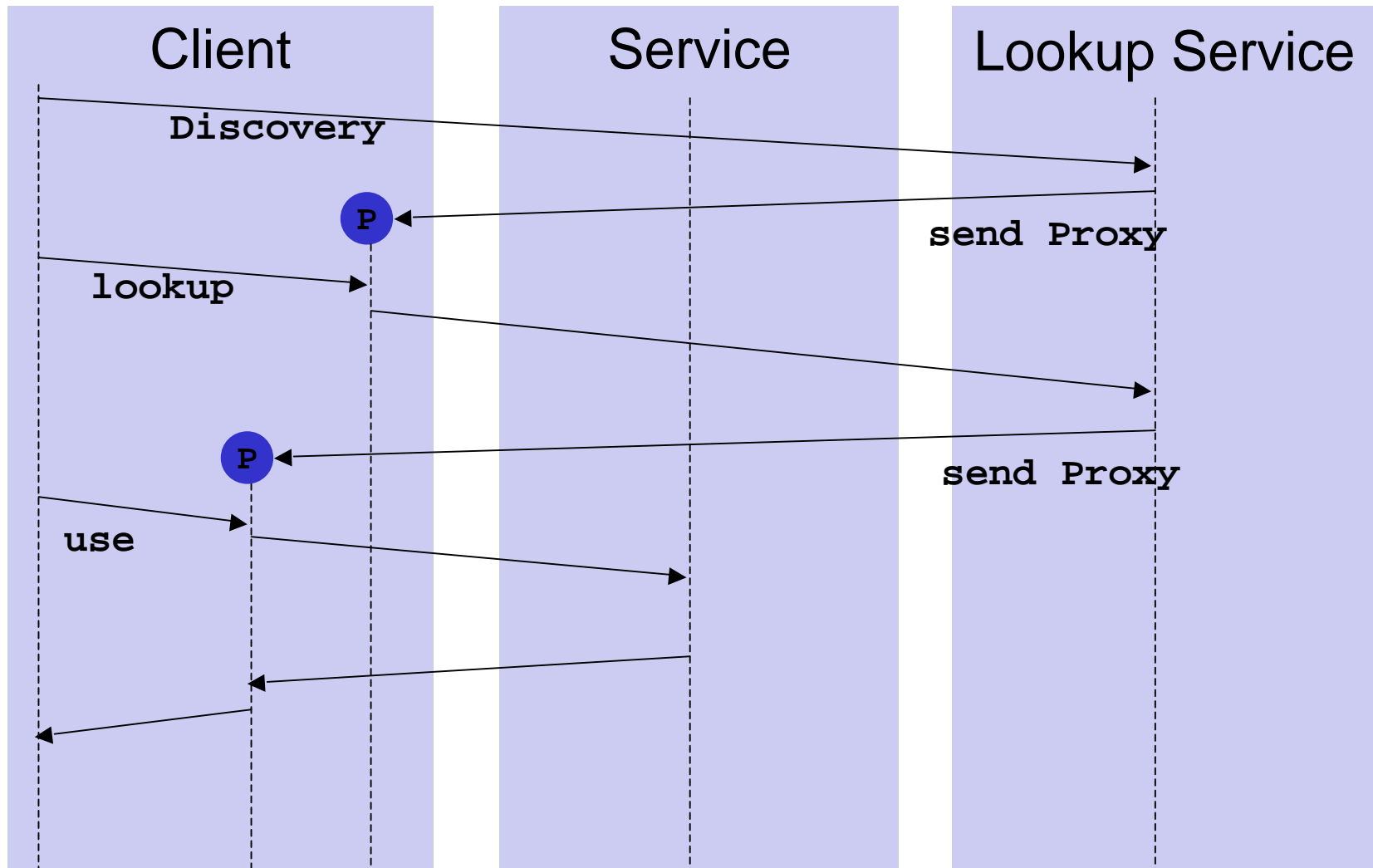
```
public class ... implements ...  
{  
    ...  
}
```

Kommunikation
(mit dem Gerät, ...)

Eigenes Protokoll
RMI
CORBA
...



Zusammenspiel



Probleme

- VM „paßt“ noch nicht in jedes Gerät
- Noch keine Produkte auf dem Markt
- Hohe Netzlast
- Dienst-Schnittstellen sind nicht genormt
- auf IP-Netze festgelegt
- Probleme mit Firewalls
- Microsoft wird JINI nicht unterstützen



JINI vs. UPnP

- Mobiler Code
- LUS

- keinen mobilen Code
- benötigt keinen zentralen LUS
- verwendet Offene Standards (XML, http)
- Microsoft macht sich dafür stark



Fazit

- Sehr viele gute Ideen
- bei kleinen Installationen praktikabel
- bei großen Installationen nicht praktikabel
- Dienst-Schnittstellen müssen genormt werden
- großer Hype und wenig konkretes

Ende

<http://www.jini.org/>



Voraussetzungen für JINI

- Java VM 1.2
- IP-Netz
 - IPv4 (+ Mobile IP)
 - IPv6



LUS startet

- beim RMI-Nameserver anmelden
- per „ Multicast Announcement Protocol“ die laufenden Dienste und Clients benachrichtigen
- auf Anfragen von Clients und Diensten warten



Dienst meldet sich an

- Sucht per „Multicast Request Protocol“ LUS
- meldet sich bei den gefundenen LUS an



Client benutzt Dienst

- Sucht per „Multicast Request Protocol“ LUS
- Sucht im LUS nach Dienst
- benutzt Dienst

