

# **Peer2peer Network Service Discovery for Ad-hoc Networks**

11.12.2003

Michael Dyrna  
dyrna@in.tum.de

# Outline

- I. Introduction
- II. Design of Service Discovery Protocols
- III. Mechanisms and Techniques
- IV. Existing Service Discovery Protocols
- V. Security Aspects
- VI. Summary & Outlook

# Outline

- I. Introduction
- II. Design of Service Discovery Protocols
- III. Mechanisms and Techniques
- IV. Existing Service Discovery Protocols
- V. Security Aspects
- VI. Summary & Outlook

# Traditional service management

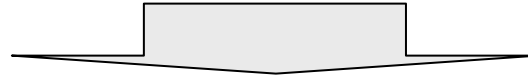
- traditional system administration not complex but time-consuming
- addresses must be assigned and published
- no standardised representation
- failures not handled
- drivers necessary on every client
- nomadic users?

# Background

proliferation of mobile networks



modularity is new goal of system development



peer-to-peer succeeds client/server



problem: service management



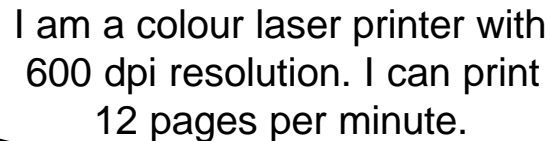
service discovery necessary

# Goals of Service Discovery

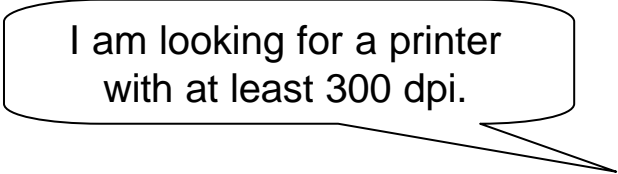
- enable modularity

therefore:

- services advertise themselves
- clients search for services



I am a colour laser printer with 600 dpi resolution. I can print 12 pages per minute.

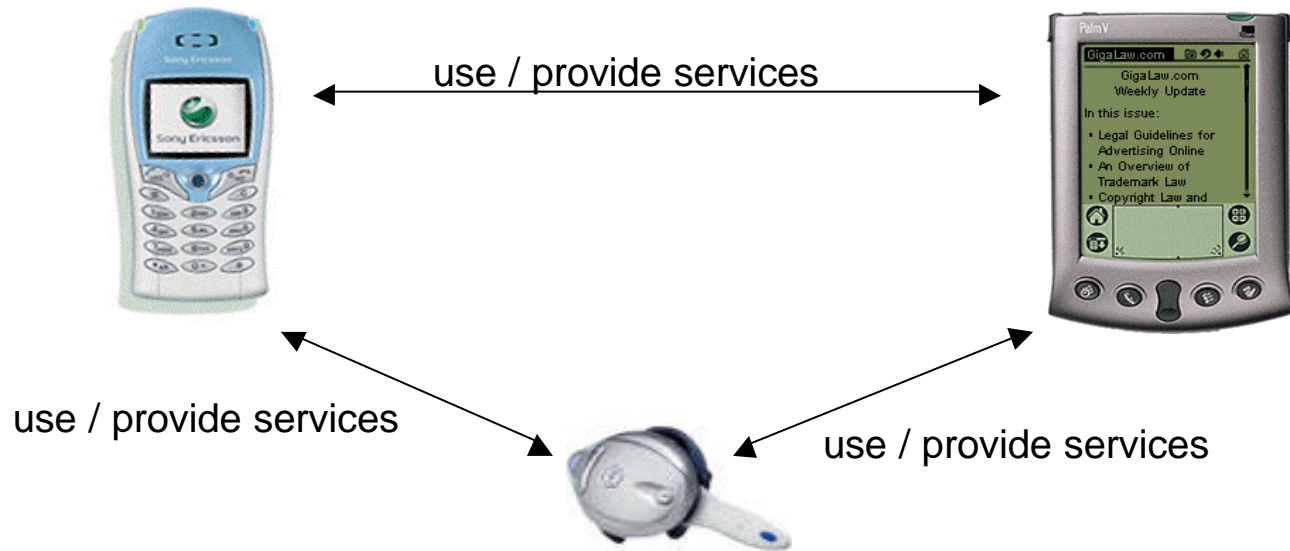


I am looking for a printer with at least 300 dpi.

# Scenario: Mobile phone, PDA and head phone

- SMS
- dial-up Internet connection
- audio input / output
- video input (camera)

- video output
- storage
- audio in-/output



- audio input
- audio output

# Outline

I. Introduction

II. Design of Service Discovery Protocols

III. Mechanisms and Techniques

IV. Existing Service Discovery Protocols

V. Security Aspects

VI. Summary & Outlook

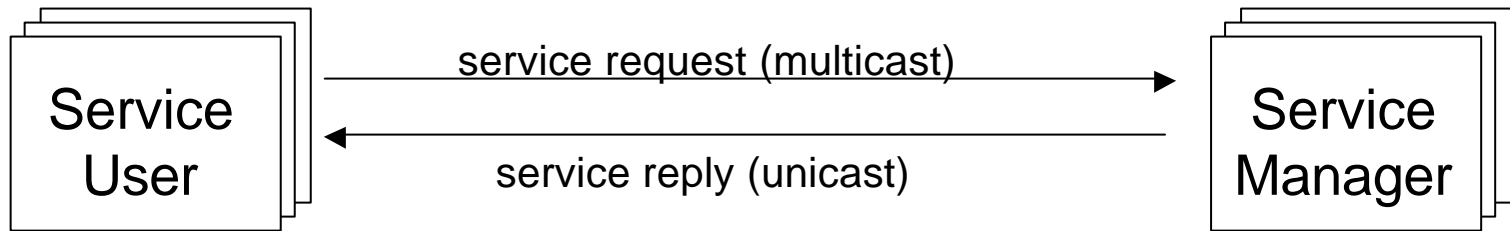


# Properties

- enable software components to find each other on a network
- provide a means for describing a service (for matching)
- techniques to detect changes in component availability
- maintain a consistent view of components

# Entities (1)

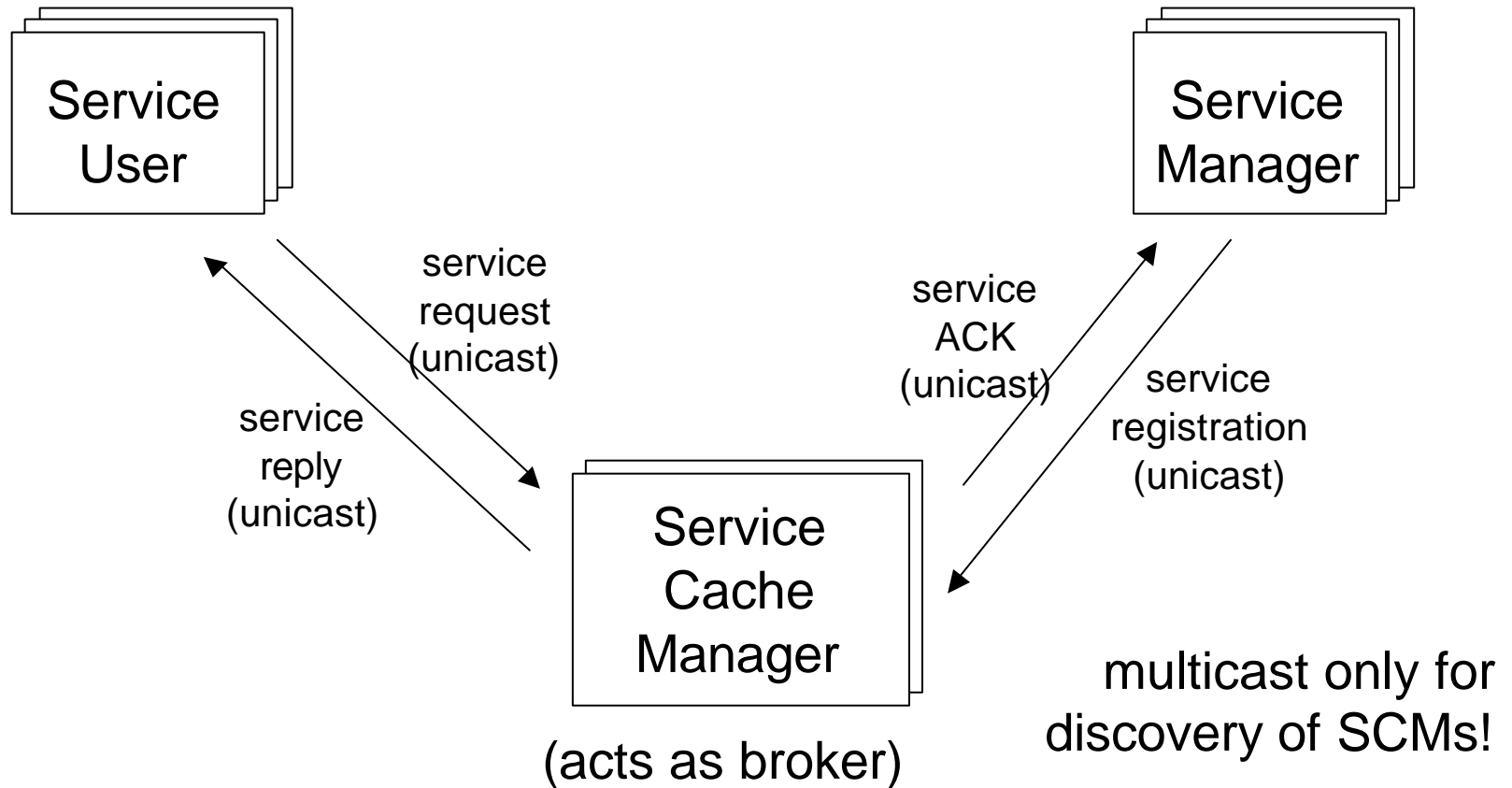
(a) without service cache manager



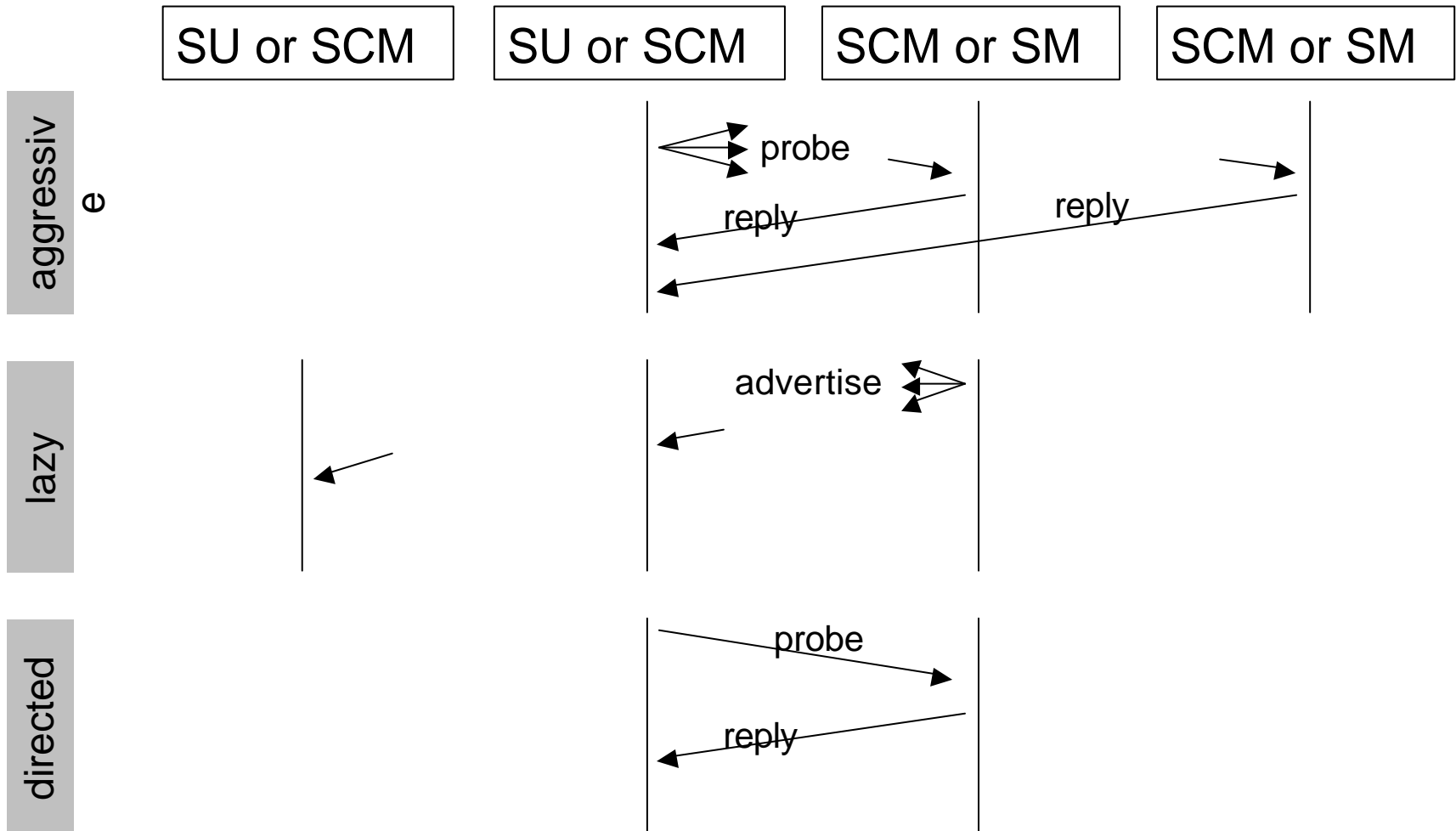
- SM holds information about services, attributes, interfaces
- SU queries for service and selects most appropriate one

## Entities (2)

(b) with service cache manager



# Discovery



# Registration

- service managers register their service descriptions with all discovered service cache managers
- service users can subscribe notification about changes in service descriptions of interest
- consistency is maintained by lease time (= period that limits validity of registration and subscription)

# Data Representation

- service identity (unique and location dependant, mandatory)
- service type (mandatory)
- optional: attributes
- optional: user interface
- optional: programme interface

Example:

Identity	192.168.8.15/mpool115
Type	modem
Attributes	baud=28800 phonenumberallowed=national dialprefix='0'

# Outline

- I. Introduction
- II. Design of Service Discovery Protocols
- III. Mechanisms and Techniques
- IV. Existing Service Discovery Protocols
- V. Security Aspects
- VI. Summary & Outlook

# Consistency Maintenance

- changes in topology on purpose or due to failure
- polling:
  - service user queries service manager or service cache manager to obtain up-to-date information
- notification:
  - service user registers with service manager to obtain up-to-date information
  - service manager notifies registered service users about changes
  - analogue: service manager <> cache manager



# Failure Detection & Recovery

- soft state persistence:
  - components send „heart beat messages“ regularly
  - no reception => assume remote component failed
  - next reception => recovery & update
- application level persistence
  - bounded retries: retry several times and throw exception
  - in case of exception, application can:
    - ignore it (good for polls/notifications)
    - retry operation (assume failure until success)
    - discard knowledge about component

# Outline

- I. Introduction
- II. Design of Service Discovery Protocols
- III. Mechanisms and Techniques
- IV. Existing Service Discovery Protocols
- V. Security Aspects
- VI. Summary & Outlook

# Service Location Protocol, Version 2 (SLP2)

- developed by IETF SrvLoc group (Sun, HP, Novell, IBM, ...)
- vendor and platform independent
- requires TCP/IP (uses mostly UDP)
- messages binary + string-based
- *User Agents, Service Agents, Directory Agents* (optional)
- DA address from DHCP or statically configured or by discovery
- *URL*: type://host:port/path
- protocol for using service is independent from SLP
- *service templates* by IANA
- scopes for administrative grouping

# Jini

- extension of Java
- developed by Sun, AOL + mobile equipment vendors
- for inter-connecting Java-enabled devices
- any component is service user and manager
- everything modelled as classes and objects
- *Lookup Table* can be on *Lookup Server* (optional)
- methods: *store*, *match*, *fetch*
- mobile code (Lookup Table contains *RMI* interface)
- “*discovery and join*”
- *groups* for administrative grouping

# Salutation

- developed by IBM, HP, Sun, Cisco, ...
- platform and network independence (any transport layer)
- can also handle access to services
- *Salutation Manager* (mandatory)
  - service broker
  - handles communication (via *Transport Manager*)
  - queries other Managers for services
- services register with only one Salutation Manager
- *Functional Units* = defined classes of devices and services
- *Service Session Management: Salutation Mode, Emulated Mode, Native Mode*

# Universal Plug and Play (UPnP)

- developed by Microsoft, Intel, Compaq, Cisco, ...
- implemented in Windows XP
- extension of Windows Plug&Play
- requires TCP/IP
- uses HTTP-over-UDP, SOAP, XML
- *Control Point, (controlled) device*
- only 2-party-architecture
- no consistency maintenance
- *control, eventing, presentation*

# Bluetooth Service Discovery Protocol (SDP)

- Bluetooth = short-range communication system
- describes all network layers, one of which is SDP
- developed by Microsoft, Intel, mobile equipment vendors
- pico nets with limited number of devices can overlap
- SDP server and client in every device
- no service cache manager
- search or browse
- consistency maintenance delegated to lower layers
- *Service Records* consist of *Service Attributes*

# Comparison

	SLP2	Jini	Salutation	UPnP	Bluetooth SDP
(main) developer	IETF	Sun	Salutation Consortium	Microsoft	Microsoft + Intel
network transport	TCP/IP	independent	independent	TCP/IP	Bluetooth
programming language	independent	Java	independent	independent	independent
OS and platform	dependent	independent	independent	dependent	independent
attributes searchable	yes	yes	yes	no	yes
service cache manager	optional	optional	mandatory	no	no
scoping	scopes	groups	no	no	not necessary
characteristics	very lightweight	code mobility, Java integration	network independence, service access	focus on hardware	high integration
qualification for ad- hoc	-	+	+	-	+
integration	-	+	-	-	++
scalability	+	+	+	-	--



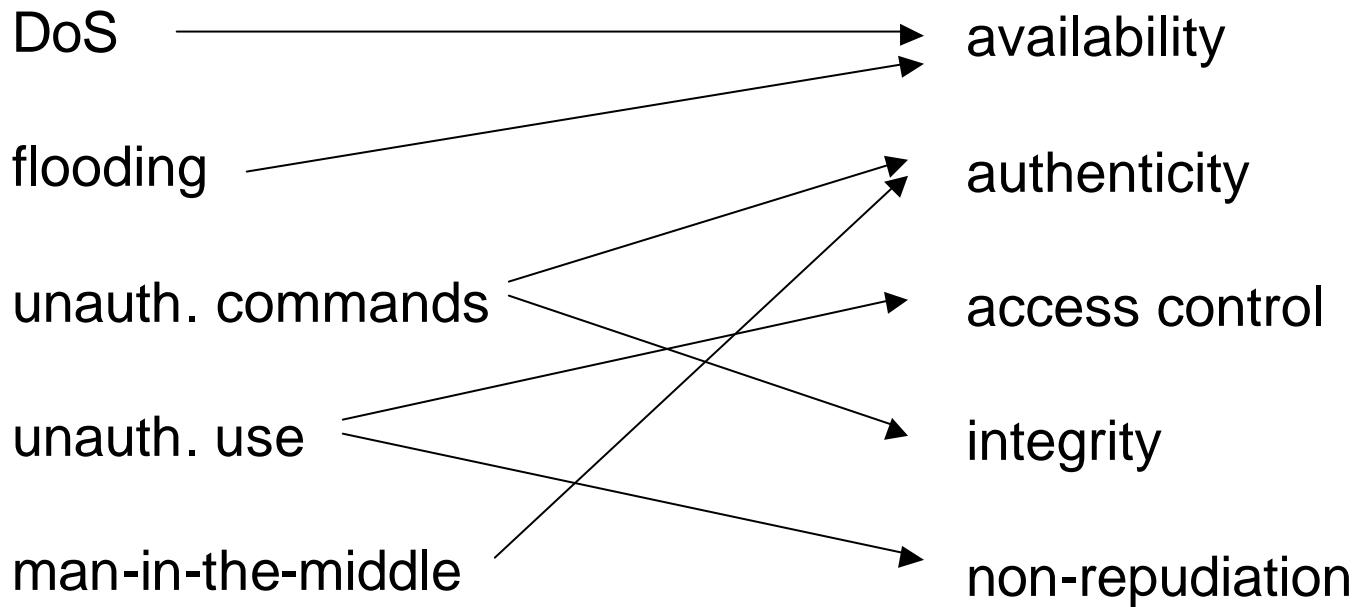
# Outline

- I. Introduction
- II. Design of Service Discovery Protocols
- III. Mechanisms and Techniques
- IV. Existing Service Discovery Protocols
- V. Security Aspects
- VI. Summary & Outlook

# Threats

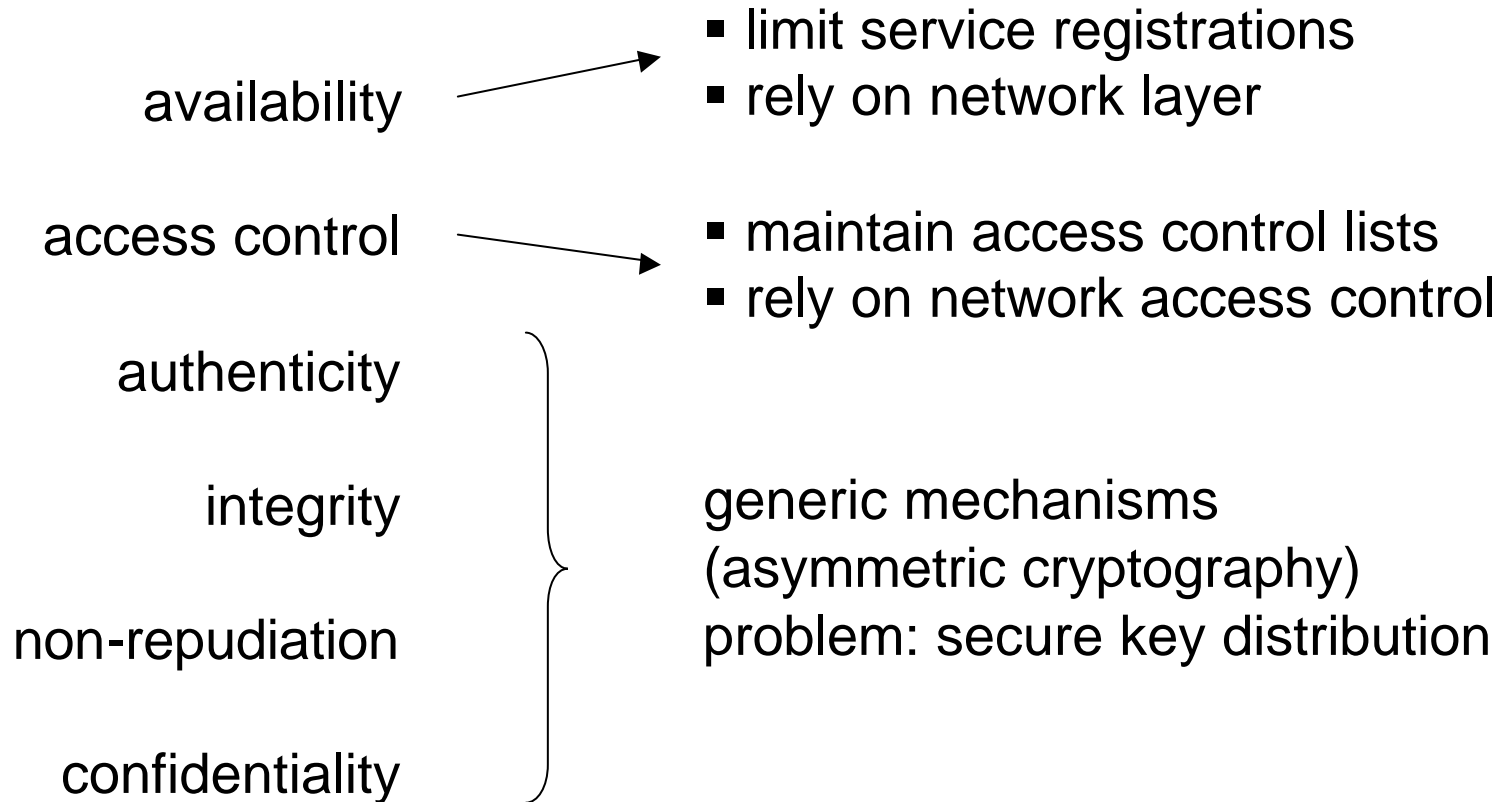
- perturbation of discovery
  - denial-of-service
  - flooding with nonsense-services
  - register and de-register services without permission
- unauthorised use of services
  - causes cost and / or enables access to sensitive data
- man-in-the-middle attack
  - eavesdrop sensitive service content (e.g. print service)
  - forward service data to actual service manager to disguise

# Requirements



not specific ← confidentiality

# Mechanisms



# Reality

- SLP2: optional authentication feature
  - but not for service users
  - manual key distribution
- Jini: Java provides most mechanisms, but mobile code causes new challenges:
  - server authentication and authorisation
  - trust problem client -> proxy (what does the proxy do?)
  - integrity only for messages, not for objects so far
- Salutation: optional authentication (username / password)
- UPnP: “Security considerations: to be determined”
- Bluetooth: usage protection and information confidentiality
  - SDP relies on these mechanisms

# Outline

- I. Introduction
- II. Design of Service Discovery Protocols
- III. Mechanisms and Techniques
- IV. Existing Service Discovery Protocols
- V. Security Aspects
- VI. Summary & Outlook

# Summary & Outlook

- trend to dynamic networks (like ad-hoc) => peer-to-peer approaches & service discovery necessary
- service discovery allows modularity
- consistency maintenance & failure handling important challenges
- SLP2 + Jini for large, Bluetooth for small homogeneous, Salutation for heterogeneous ad-hoc networks, UPnP at most for Windows notebooks ;-)
- co-existence + more protocols?
- security issue not well solved => future improvements?

**Questions? Answers!**

**Thank you for  
your attention!  
:-)**