Mehrstufige Systemarchitektur für mobile spontan vernetzte Geräte

Clemens Cap
FB Informatik
Institut für Technische Informatik

Dirk Timmermann
FB Elektrotechnik u. Informationstechnik
Institut für Angewandte Mikroelektronik und Datentechnik

Agenda

- Derzeitiger Status
- Projektziele und –struktur
- Spontane Vernetzung, Java, Jini
- Systemarchitektur
- Smartcards, Java Prozessoren
- Aufgaben

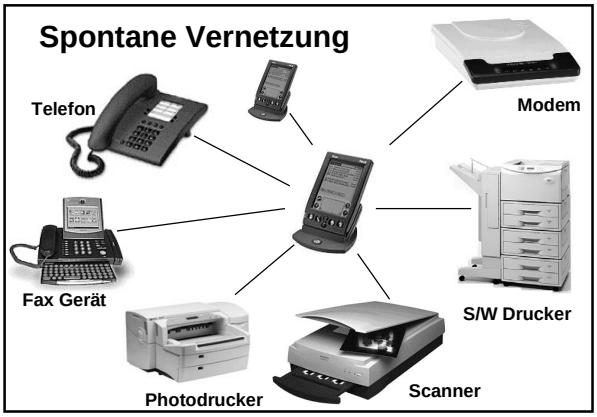
Projektziele

- Untersuchung der *ungelösten Sicherheitsprobleme* im Bereich *mobiler* Systeme bei *spontaner* Vernetzung über *drahtlose* Verbindungen
- Ganzheitliche theoretische und praktische Untersuchungen im Soft- *und* Hardwarebereich
- Sichere Authentifizierung, Verschlüsselung und Zugriffssteuerung trotz geringerer Prozessorleistung und niedriger Stromaufnahme auf mobilen Systemen
- Entwicklung von Techniken der *Anonymisierung* und der individuellen Konfiguration durch den Anwender
- Prototypische Implementierung eines
  - gestuften Client-Server Systems, bestehend aus
    - mobilem PersonalCard-Client aus Javaprozessor + Kryptofähigkeit + drahtloser Schnittstelle
    - PC-basiertem Server
  - Beispielanwendung als Referenzszenario, z.B. Info-Badge, digitaler Assistent

Projektstruktur

- ***Cap***
  - Sichere Protokolle für drahtlose Kommunikation
  - Mobile sowie spontanvernetzte Einsatzbereiche
  - Sicherheitsarchitektur für spontane Vernetzung
  - Implementierung einer Beispielanwendung
- ***Timmermann***
  - Hardwaredesign der PersonalCard
  - Java-Prozessor mit Kryptofähigkeit
  - Sicherheitsarchitektur und Protokolle
  - I/O und drahtlose Kommunikation
- ***Gegenseitiges Cross-Auditing der Sicherheitsaspekte***

Spontane Vernetzung



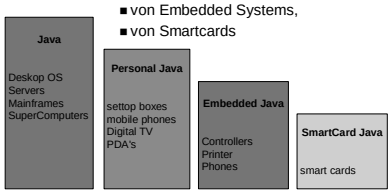
Spontane Vernetzung

- **Aufsetzen der Basisparameter zur Kommunikation (Bsp: IP / DHCP, IP / BOOTP)**
- **Kryptographisches „Booten“**
- **Auffinden von Dienstverzeichnissen**
- **Auffinden von Diensten**
- **Nutzen von Diensten (auf Geräte & Anwendungsebene)**
  - Installieren gerätespezifischer Treiber
  - Reservierung von Betriebsmitteln
  - Verteilen von Ereignissen
  - Durchführen von Transaktionen

  Jini Funktionalität
- **Sicherheitsaspekte**
- **Ubiquität bei spontaner Bindung**
- **Mobilität bei langandauernder, wechselnder Bindung**

Java

- **Java als Programmiersprache**
  - des World Wide Web,
  - verteilter Anwendungen,
  - von Embedded Systems,
  - von Smartcards



Sicherheitsaspekte

- **Sicherheit des Verzeichnisses**
- **Zugang**
  - Authentisieren
  - Autorisieren
  - Anonymisieren
- **Bezahlen**
- **Profilerstellung**
- **Multiparty-Kommunikation**

Eigentümer der Daten bestimmt deren Nutzung

„Jini over SSL“ reicht nicht aus

Gedanken zur Applikation

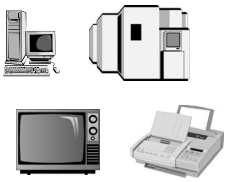
Special Purpose Maschine
Mobil & Ubiquitäres Umfeld
Leichtgewichtige Dienste

- **Einfache Nutzung eines Dienstes**
  - Anonym / Persönlich bei Autorisierung, Bezahlung, Profilerstellung
  - Bsp: Nachrichtendienst, Zutrittskontrolle (Mitbestimmungs-konform)
- **Multicast von Informationen**
  - Identitätsabhängiger Zugang ohne Roaming
  - Bsp: Börsenticker
- **Koppelung mehrerer Nutzer**
  - Profilabhängige Kommunikation, bidirektional, Multicast oder Konferenz
  - Durch Vermittlung von zentralem Server
  - Server lernt aus Kommunikation möglichst wenig
  - Bsp: Messe, active badge, drahtlose Abstimmungen und Auktionen, Verkehrskoordinationssystem (Taxi-Sharing)

Zweistufige Architektur für Mobilität

Mobiles Gerät / Badge | **Schnittstelle** | **Serviceprovider**

Infrarot
Bluetooth
Wireless LAN
GSM
UMTS



- **Eigenschaften**
  - Knappe Ressourcen: Größe, Performance, Stromverbrauch, Bandbreite, Reichweite
  - Jini zu „schwer“
  - Konsequenzen für Konfigurierbarkeit, Anonymität, Sicherheit

Mehrstufige Architektur für Mobilität

Mobiles Gerät / Badge | **Schnittstelle** | **Lokaler Serviceprovider LSP** | **Zentraler Serviceprovider ZSP**

Infrarot
Bluetooth
Wireless LAN
GSM
UMTS

Protokoll

z. B. Desktop auf Messe
oder
PDA + LSP

Protokoll

z.B. Desktop zu Hause, enthält Profil des Users

knapp
reichlich
Ressourcen

- **Mobilgerät (intell. Badge) sehr einfach**
- **Enthält nur IP-# oder URL und 'etwas' Kryptofähigkeit**
- **Systemarchitektur für mobile Sicherheit**
- **mehrstufiges Konzept nutzt jeweils verfügbare Ressourcen**

Zu entwickeln

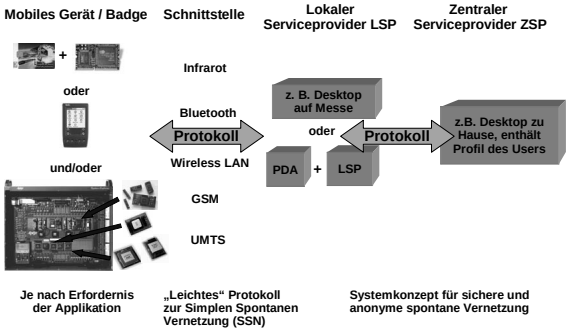
Mobiles Gerät / Badge | **Schnittstelle** | **Lokaler Serviceprovider LSP** | **Zentraler Serviceprovider ZSP**

Java Smart Card + **µC**
oder
Palm / PDA
und/oder
VHDL – Module auf Emulator

Java Proc | **Krypto Proc** | **evtl. RF-IF**

Je nach Erfordernis der Applikation

Zu entwickeln



Java auf Smartcards Besonderheiten

- **Einschränkungen der Sprache**
  - keine Strings, Chars, Fließkommazahlen, Threads
  - keine <clinit>-Methoden, (java.*), mehrdimensionalen Arrays
- **Erweiterungen der Klassenbibliothek**
  - neue Packages (javacard.* , javacardx.*)
- **Einschränkungen der JVM (u.a.)**
  - nicht alle Bytecodes verwendet
  - Stack: 127 Byte vs. 64K Worte
- **Größe nur 14 KByte ROM !**
- **Für PersonalCard:**
  - Anpassung und *Hardware*umsetzung der Smartcard-JVM
  - Bereits teilweise erfolgt:
    - Java Prozessor für Bytecode der Spezifikation Java SmartCard 2.1
    - Erweitert um I/O Befehle
    - KAFFEMASCHINE zur Entwicklung des Bytecodes aus Java, Simulationsumgebung

Deliverables

- **Sicherheitskonzept für Dienste in einem mobilen, spontan vernetzten Umfeld**
- **Umsetzung des Sicherheitskonzepts für Anwendungsprogrammierer in einer Java-API**
- **Lightweight Simple Spontaneous Networking Protokolle**
- **PersonalCard mit Java-Prozessor, Kryptofähigkeit und drahtloser Kommunikationsschnittstelle**
- **Anwendungsprogramm im Referenzszenario**