

„Kommunikation und Austausch von Multimediadaten in spontanen Bluetooth-Netzwerken mittels J2ME“

Diplomarbeit Nepomuk Karbacher

Präsentation: 12.MAI.2005

Inhalt der Präsentation

Die Applikation „n3po“:

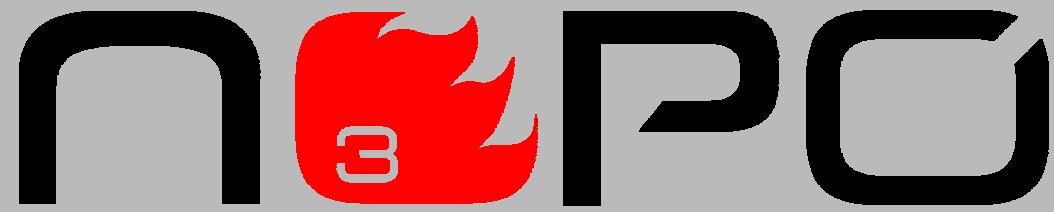
- Funktionalität
- Architektur/Implementierung

Test auf den Endgeräten:

- Nokia 6230
- Siemens S 65



Die Applikation:



WWW.N3PO.COM

Applikation: **Einsatzgebiet**

Suche von Personen mit bestimmten Merkmalen im direkten Umfeld mittels Bluetooth.

- Suchen von anderen Bluetooth-Geräten in einstellbaren Intervallen
- Suchen des eigenen Services auf gefundenen Geräten
- Prüfen der Merkmale
- Herstellung einer direkten Verbindung zwischen zwei Endgeräten
- Kommunikation und Übertragung von Multimediadaten
- Wiederherstellung der Verbindung nach Abbruch

Applikation: **Technische Voraussetzungen**

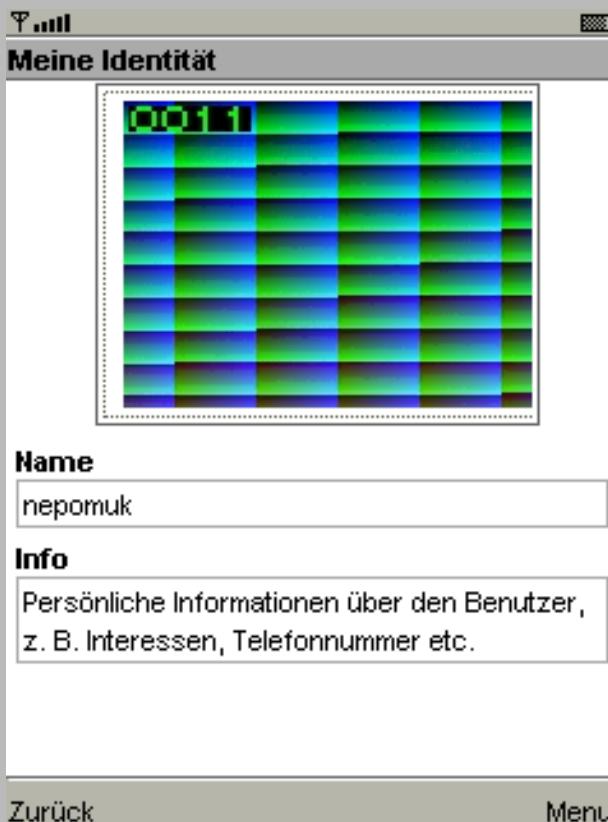
J2ME-fähiges Mobiltelefon mit **CLDC 1.1** und **MIDP 2.0**

optionale Pakete:

- Mobile Media API (**MMAPI**)
- Wireless Messaging API (**WMA**)
- Bluetooth API (**JABWT**)
- FileConnection API (**optional**)

Größe des .jar-Archivs incl. Grafiken und Audio: ~140 KB

Applikation: Funktionsweise



Identität:

Erstellen der eigenen Identität, bestehend aus:

- Foto
- Name
- Infotext

Das Foto kann über die interne Kamera aufgenommen, oder über das Filesystem eingelesen werden.

Applikation: Funktionsweise



Kennwortsuche:

Suche über individuelles Kennwort z. B:

Mein Kennwort: SucheKarte

(unter diesem Begriff werde ich gefunden)

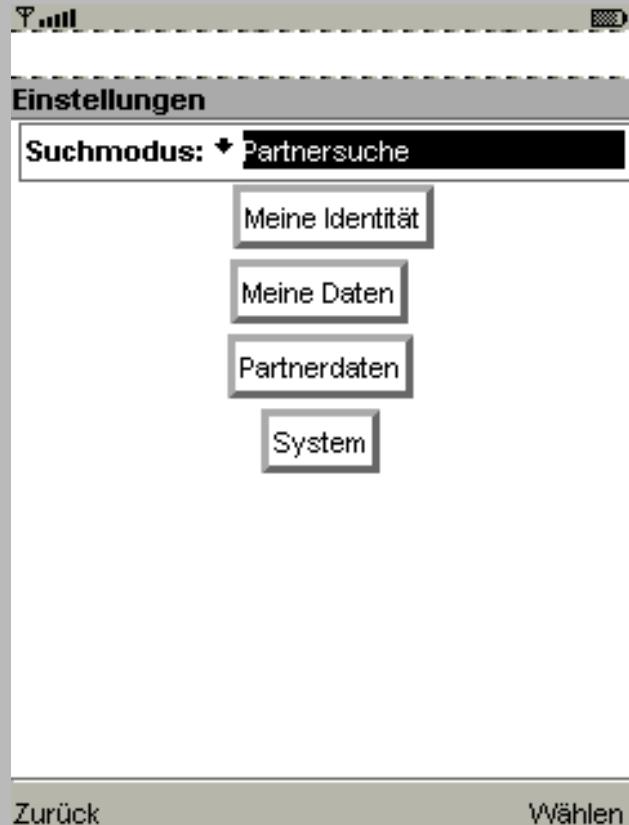
PartnerKennwort: HabeKarte

(nach diesem Begriff suche ich)

Vor Festivals oder Konzerten.

Zahlreiche andere Anwendungsmöglichkeiten können über die Kennwortsuche universell abgedeckt werden.

Applikation: Funktionsweise

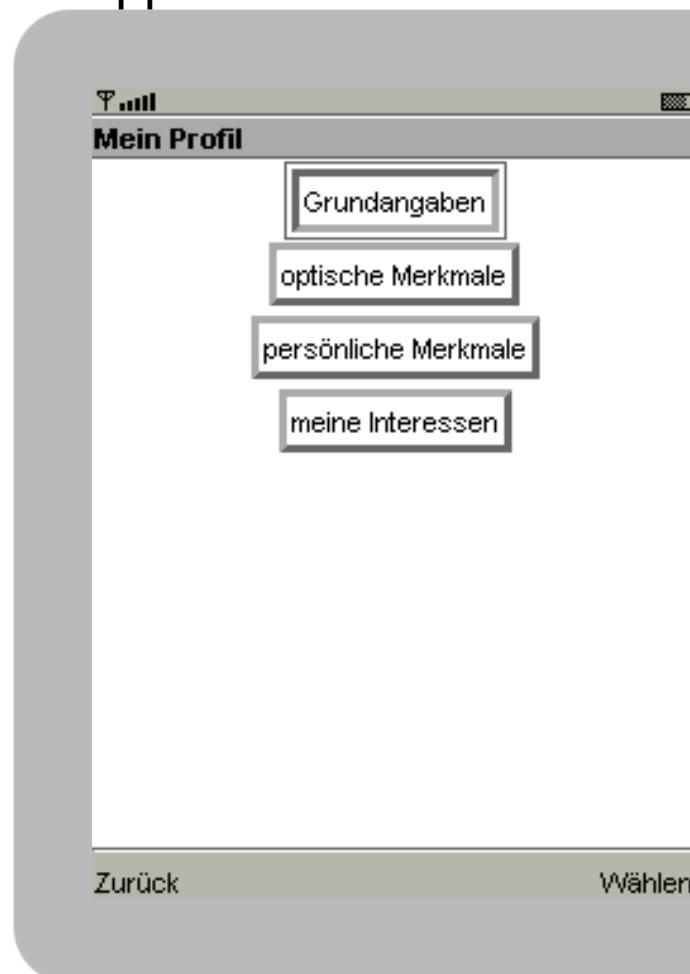


The screenshot shows a mobile application interface with a navigation bar at the top. Below the bar, the text "Einstellungen" is displayed. A dropdown menu is open, showing the text "Suchmodus: * Partnersuche" followed by four options: "Meine Identität", "Meine Daten", "Partnerdaten", and "System". The "Partnersuche" option is highlighted with a black bar and a white asterisk. At the bottom of the screen, there are two buttons: "Zurück" on the left and "Wählen" on the right.

Partnersuche-Modus:

- Statt Kennwörtern werden Personenprofile miteinander verglichen.
- Angelehnt an Internet-Partnerbörsen.
- Erstellung der Profile und Vergleich derer direkt auf dem Endgerät.
- Keine zwischengeschalteten Server
- > keine weiteren Kosten

Applikation: Funktionsweise



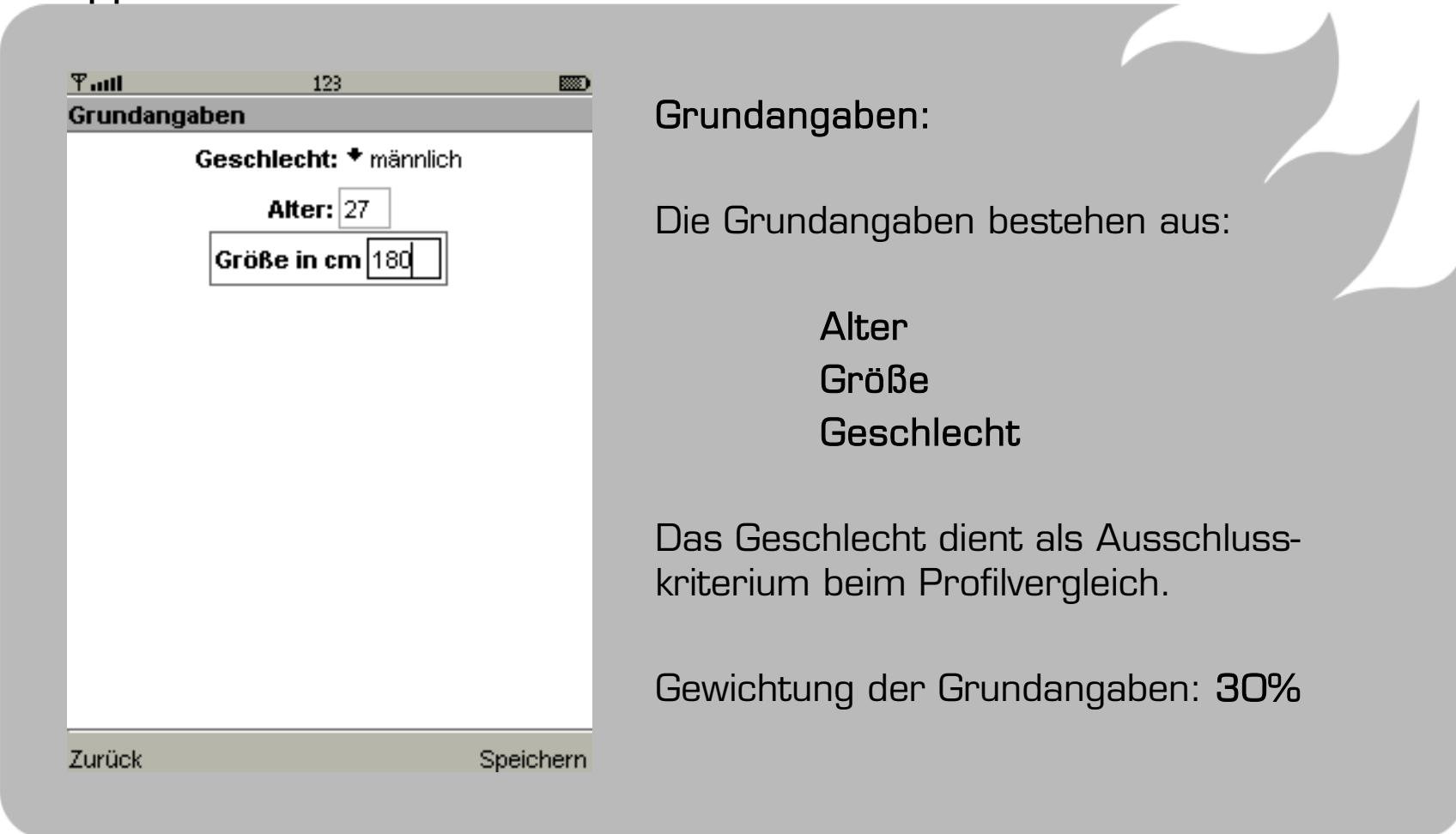
Persönliches Profil:

Beschreibt die eigenen Eigenschaften.

Kategorien:

Grundangaben
optische Merkmale
persönliche Merkmale
meine Interessen

Applikation: Funktionsweise



Grundangaben:

Die Grundangaben bestehen aus:

- Alter
- Größe
- Geschlecht

Das Geschlecht dient als Ausschlusskriterium beim Profilvergleich.

Gewichtung der Grundangaben: 30%

Grundangaben

Geschlecht: männlich

Alter: 27

Größe in cm: 180

Zurück **Speichern**

Applikation: Funktionsweise

The screenshot shows a software interface with a title bar 'optische Merkmale'. Below it is a list of four items: 'Figur' (slim), 'Haarfarbe' (blonde), 'Augenfarbe' (blue), and 'Frisur' (medium-length hair). The 'Frisur' item is highlighted with a black border. At the bottom are buttons for 'Zurück' (back) and 'Speichern' (store).

Optische Merkmale:

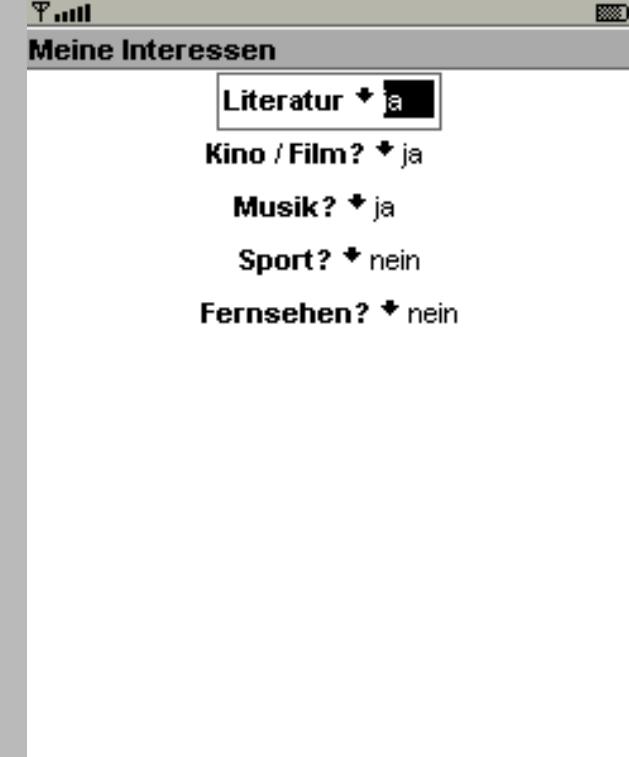
Die optischen Merkmale bestehen aus:

- Figur
- Haarfarbe
- Augenfarbe
- Frisur

Optionen vordefiniert, Popup-Menues

Gewichtung der optischen Merkmale: 20%

Applikation: Funktionsweise



Meine Interessen:

Die wählbaren Interessen beschränken sich in dieser Version auf:

- Literatur
- Kino bzw. Film
- Musik
- Sport
- Fernsehen

Optionen: „ja“ oder „nein“
„nein“ ist die Voreinstellung

Gewichtung der Interessen: 25%

Applikation: Funktionsweise

Meine Merkmale

Raucher? nein

Fitness? durchschnittlich

Familienstand? ledig

Kinder? 0

Schulabschluss? Abitur

Zurück **Speichern**

Persönliche Merkmale:

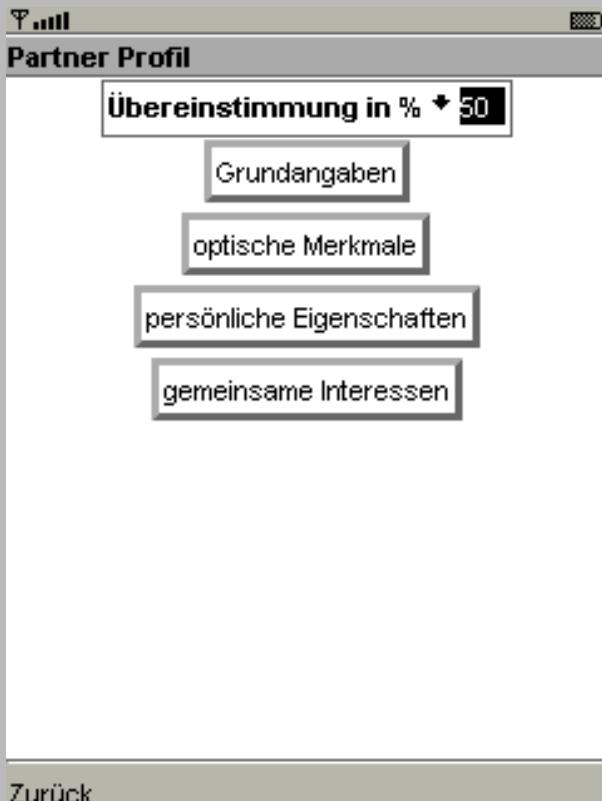


Die persönlichen Merkmale bestehen aus:

- Rauchstatus
- Fitness
- Familienstand
- Kinderzahl
- Schulabschluss.

Gewichtung: 25%

Applikation: Funktionsweise



Partner-Profil:

Das Partner-Profil legt die beim Partner gewünschten Eigenschaften fest.

Kategorien analog zum eigenen Profil.

Übereinstimmungsprozentsatz

(wie viele der gewünschten Eigenschaften muss der potentielle Partner erfüllen?)

Applikation: Funktionsweise

123

Grundangaben

Geschlecht: ♀ weiblich

Alter von: 22

Alter bis: 29

Größe von: 170

Größe bis: 180

Zurück **Speichern**

PartnerProfil: Grundangaben

Bestehend aus:

- gesuchtem Geschlecht
- minimales Alter
- maximales Alter
- minimale Größe
- maximale Größe

Applikation: Funktionsweise

Screenshot of a software application window titled "Gemeinsame Interessen". The window lists several interests with dropdown menus:

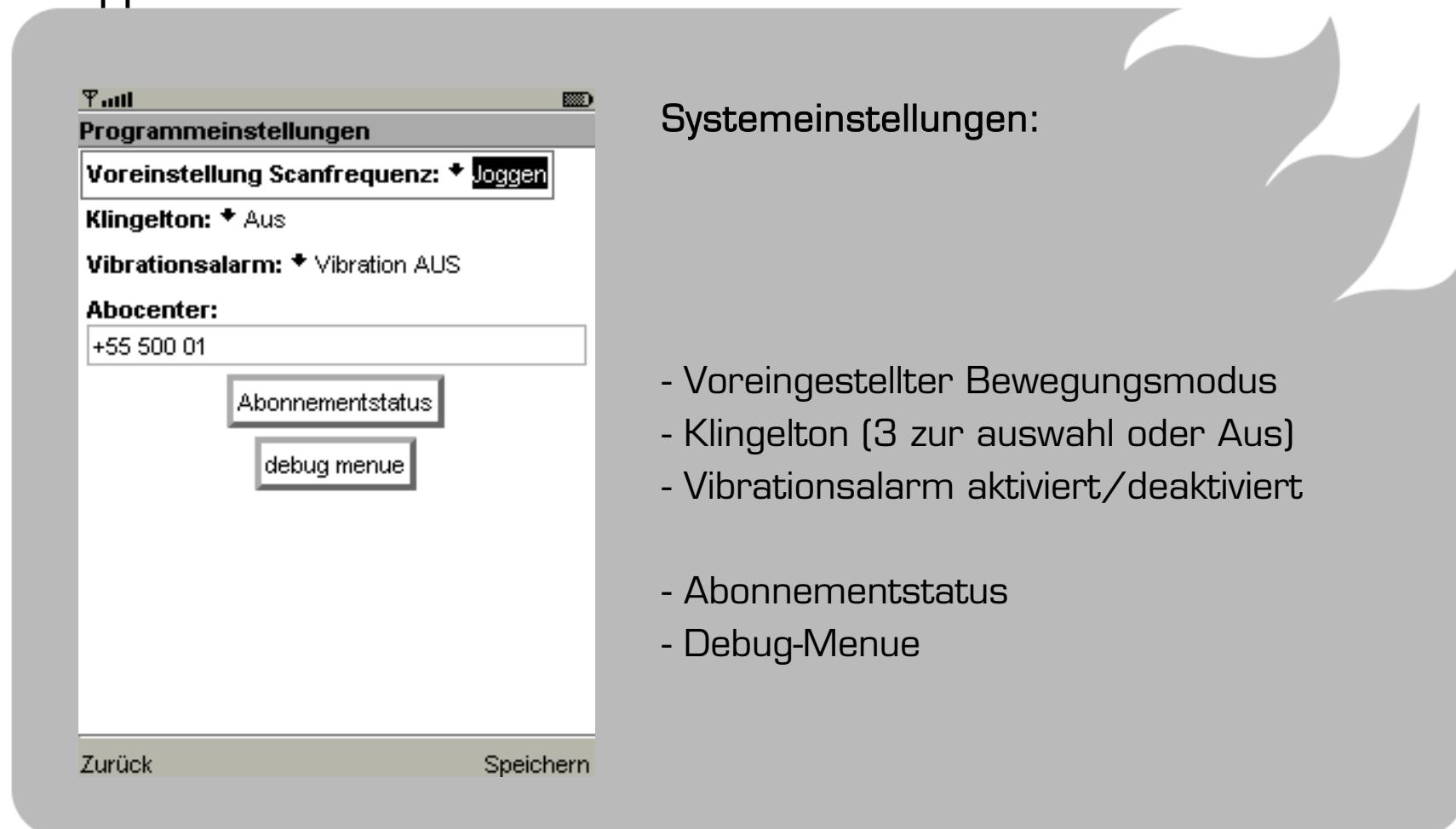
- Literatur: ▾ egal
- Kino / Film?: ▾ egal
- Musik?: ▾ ja
- Sport?: ▾ nein
- Fernsehen: ▾ ja ▾ nein ▾ egal

At the bottom of the window is a "Cancel" button.

Andere Dialoge im PartnerProfil:

Bei den Eigenschaften kann neben „ja“ und „nein“ auch „egal“ ausgewählt werden.

Applikation: Funktionsweise



Programmeinstellungen

Voreinstellung Scanfrequenz: ▾ **Joggen**

Klingelton: ▾ **Aus**

Vibrationsalarm: ▾ **Vibration AUS**

Abocenter:
+55 500 01

Abonnementstatus

debug menue

Zurück Speichern

Systemeinstellungen:

- Voreingestellter Bewegungsmodus
- Klingelton (3 zur auswahl oder Aus)
- Vibrationsalarm aktiviert/deaktiviert
- Abonnementstatus
- Debug-Menue

Applikation: Funktionsweise



Abonnementstatus:

Ihr Abo läuft seit: Fri Mar 18 11:04:01 UTC 2005

Ihr Abo läuft noch: 6 Tage

Abo verlängern

Zurück

Abonnementstatus:

Bezahlmodell der Applikation:

Abonnement für einen Monat

Bezahlung direkt im Programm über „Premium-SMS“ an Abocenter-Nummer.

Applikation: Funktionsweise



Abonnement verlängern:

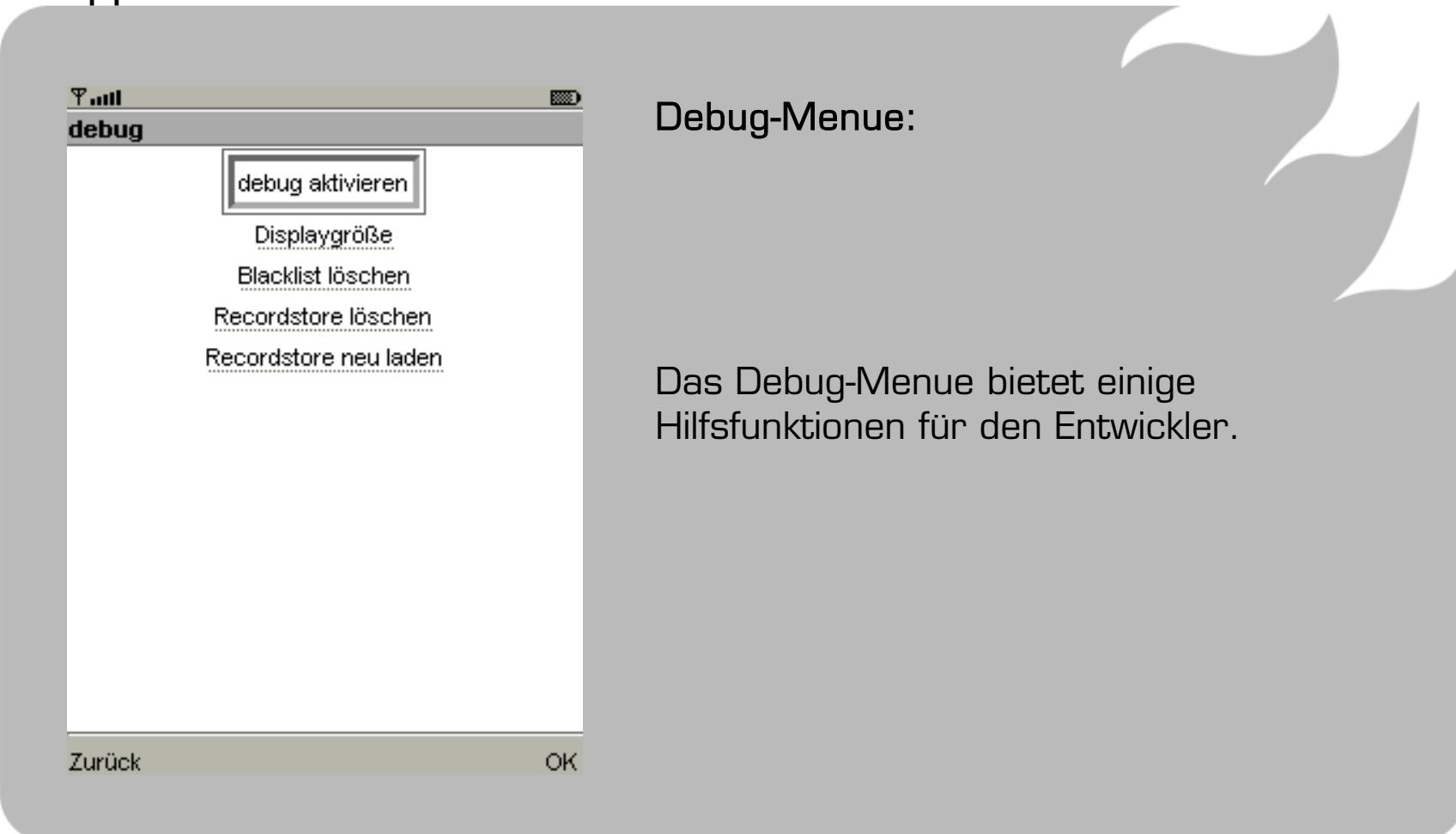
Freischaltcode für einen Monat als Antwort auf die Premium-SMS.

Bei Ablauf des Abonnements:

- Benachrichtigung des Benutzers
- Suche kann nicht mehr aktiviert werden

Programm lässt sich weiterhin starten, Menues bleiben benutzbar.

Applikation: Funktionsweise



Debug-Menue:

Das Debug-Menue bietet einige Hilfsfunktionen für den Entwickler.

Applikation: Funktionsweise

```
CoreMidlet: uid vorhanden nepomuk
CoreMidlet: Abostatus ok
Netlayer:: invoke init()
CoreMidlet: Bluetooth gestartet
DebugUtil: Print Service Record (# of element: 5
DebugUtil: Print Service Record URL btsp://0123
DebugUtil: i = 0DataElement[ServiceAvailability]
DebugUtil: i = 1DataElement[ServiceDescription]
DebugUtil: i = 2DataElement[ProtocolDescriptorLi
DebugUtil: i = 2  DataElement[ProtocolDescriptor
DebugUtil: i = 3DataElement[ServiceName] BTAuth
DebugUtil: i = 4DataElement[ServiceclassIDList]
DebugUtil: i = 4 DataElement[ServiceclassIDList
DebugUtil: i = 4 DataElement[ServiceclassIDList
Netlayer:: Server wartet auf Verbindung
```

Zurück

Debug-Modus:

Wenn der debug-Modus aktiviert wird, werden die Ausgaben auf eine Canvas gezeichnet, um möglichst viel Text auf dem Display darstellen zu können.

Applikation: Funktionsweise



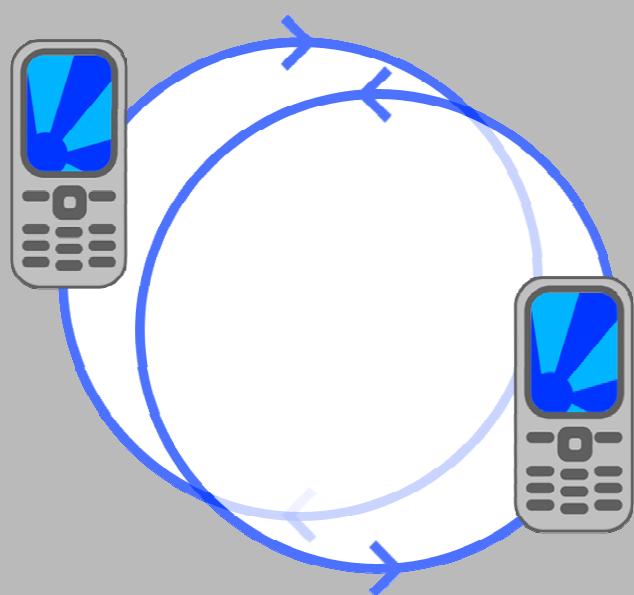
Aktivieren der Suche:

Vollständigkeit der Identität wird geprüft

Temporären Bewegungsmodus einstellen
(Joggen, Gehen, Sitzen)

„aktivieren“-Button auswählen

Applikation: Funktionsweise



Suche aktiviert:

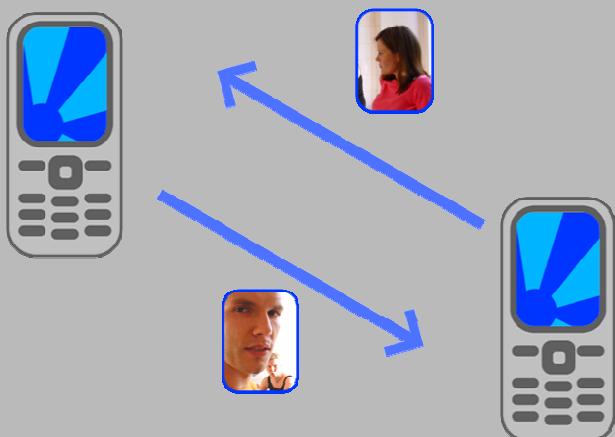
- Suche nach Bluetooth-Geräten
- in Intervallen (Bewegungsmodus)

Wenn ein Gerät gefunden wurde, wird der Service der „n3po“-Applikation darauf gesucht.

Wird der Service nicht gefunden:

weitere Geräte suchen

Applikation: Funktionsweise



Verbindung aufbauen:

Wird n3po-Service gefunden:

- Profil/Kennwort prüfen

Wenn Profil/Kennwort passt:

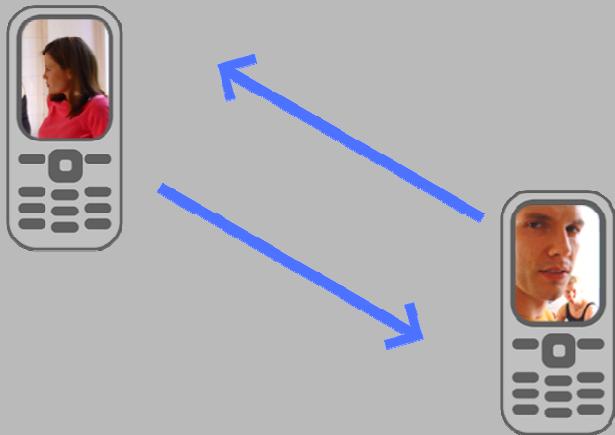
- Datensatz verschicken/empfangen
(Profil/Kennwort, Foto, Infotext, Name)

Wenn Profil/Kennwort nicht passt:

- Gerät auf „KnownDevices“ – Liste
- Gerätesuche fortsetzen

Applikation: Funktionsweise

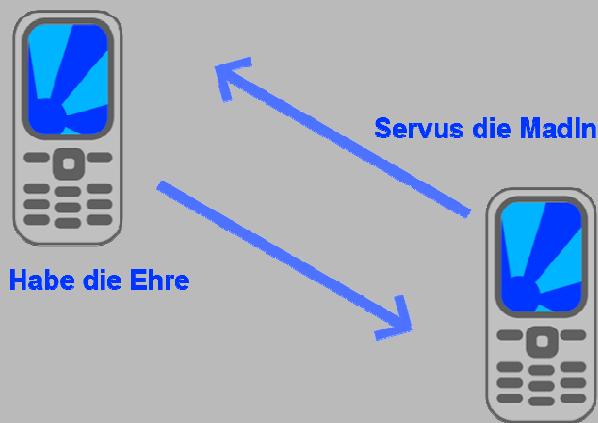
Verbindung aufgebaut:



- Benachrichtigen des Benutzers
(Klingelton / Vibration / Backlight)
- Darstellen von Foto, Name, Infotext

Person ist max. 10 – 100m entfernt und
mittels Foto identifizierbar.

Applikation: Funktionsweise

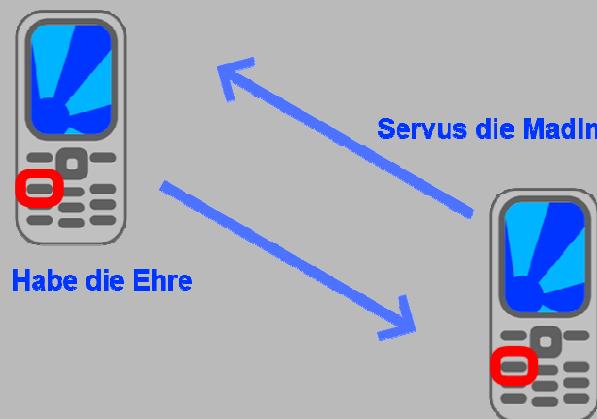


Text-Chat per Bluetooth:

Kostenlose Kommunikation via Textnachrichten
z. B. wenn Person nicht direkt sichtbar.
- bis 500 Zeichen pro Nachricht
- Bestätigungston

Applikation: Funktionsweise

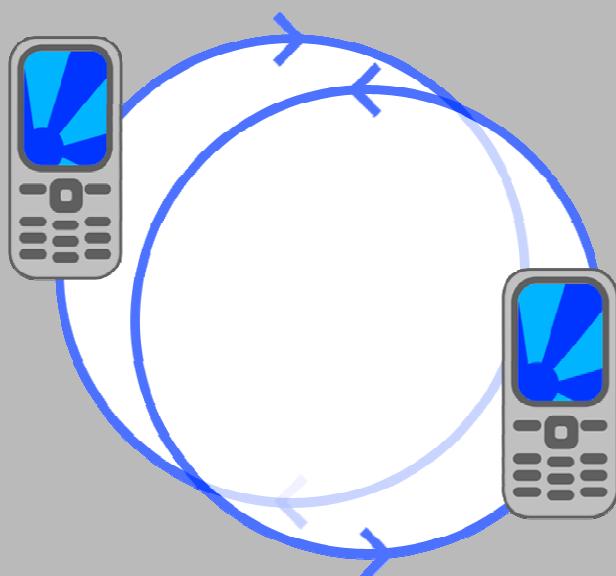
VoiceChat via Bluetooth



Ähnlich Walkie-Talkie/Push to talk

- Aufnahme solange Taste gedrückt ist.
- Verschicken der Audiodaten
- Abspielen über Lautsprecher des anderen Geräts
- Bestätigungston

Applikation: Funktionsweise



Wiederherstellen der Verbindung:

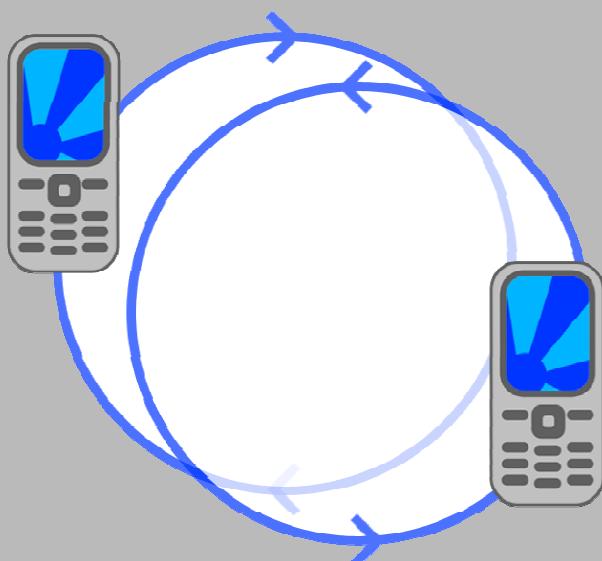
Bluetooth: Instabile Verbindung

Nach Störung oder Überschreitung der Reichweite:

- Programm bemerkt Verbindungsabbruch
- Versucht, Verbindung wiederherzustellen
- Solange Verbindung gestört: **Fehlerton**

Applikation: Funktionsweise

Beenden der Kommunikation:



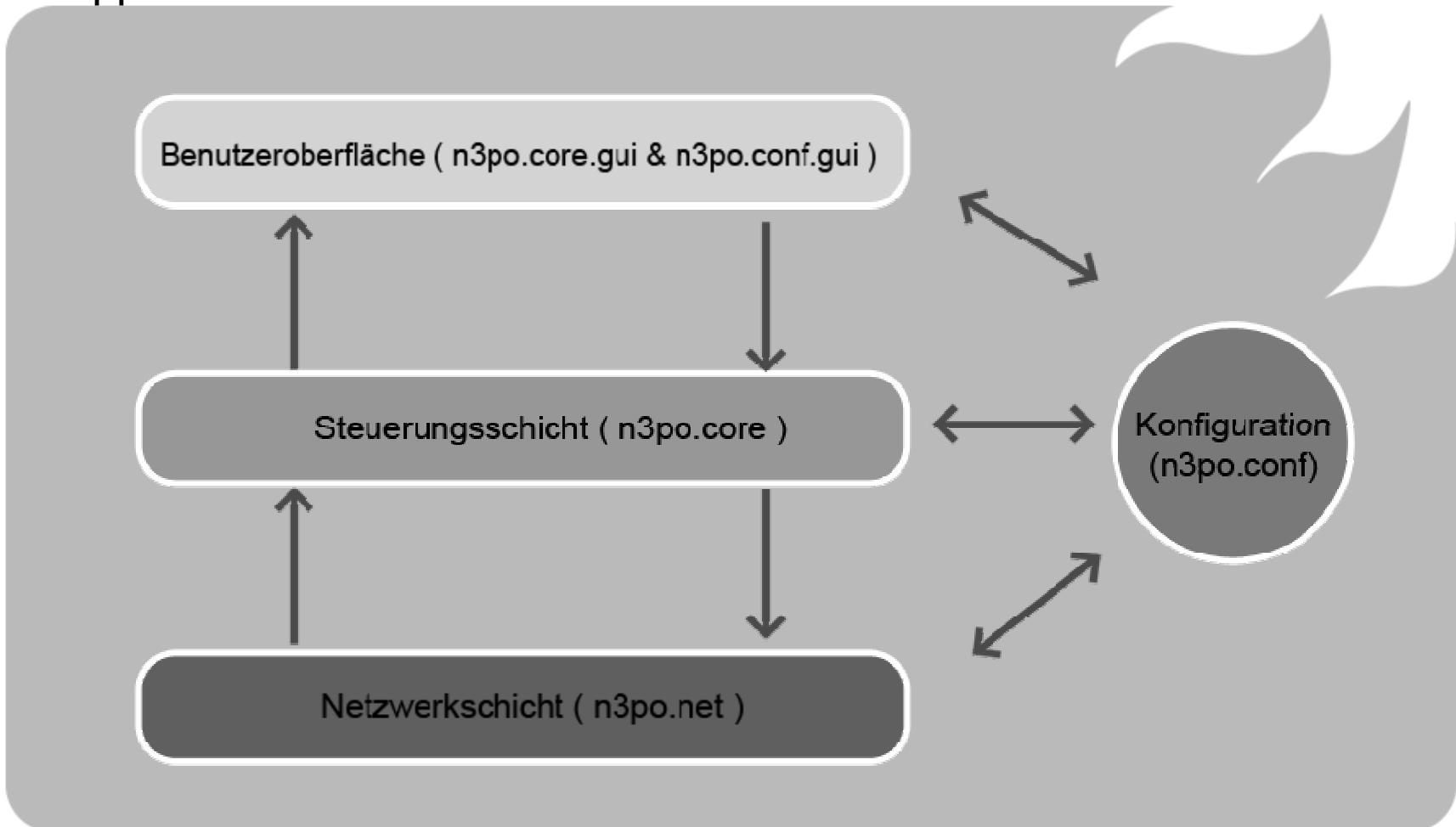
- **Blacklist-Funktion** zum Speichern unerwünschter Kommunikationspartner (speichern der Bluetooth-MAC)
- Wird die Kommunikation durch einen Benutzer beendet, wird die Suche nach anderen Geräten fortgesetzt.

Applikation: Architektur



IMPLEMENTIERUNG/ARCHITEKTUR

Applikation: Architektur



Applikation: Benutzeroberfläche

Klassen der Benutzerschnittstelle sind als Singletons implementiert, da von jeder Klasse maximal eine Instanz benötigt wird. `getInstance()`

Anforderungen können mit vorgefertigten Items gut abgedeckt werden.
- GUI-Klassen bauen auf den Form-Klassen der High-Level-UI auf.

Zwei Ausnahmen: VoiceChat und Textausgabe im Debug-Modus
Bauen auf Low-Level-UI Klassen auf ([Canvas](#))

Vorteil: Vertikal scrollbarer Text, Abfrage des Zustands von Gerätetasten

Applikation: **Steuerungsschicht**

Vermittelt zwischen Eingabe des Benutzers über die GUI-Klassen und der Netzwerkschicht.

Klasse **CoreMidlet** enthält Methoden zum:

- Starten, Pausieren und Beenden der Applikation
- Verschicken von Nachrichten/Daten
- Programmlogik (Profiltausch/Überprüfung)

Auf Ereignisse der Netzwerkschicht wird über **Callback-Methoden** reagiert, welche die Klasse **CoreMidlet** implementiert.

Klasse **MMUTIL**: Anpassung der Fotogröße an Handydisplay, Abspielen und Aufnahme von Audiodaten und MIDI

Klasse **DebugUtil**: Protokollierung des Programmablaufs

Applikation: **Netzwerkschicht**

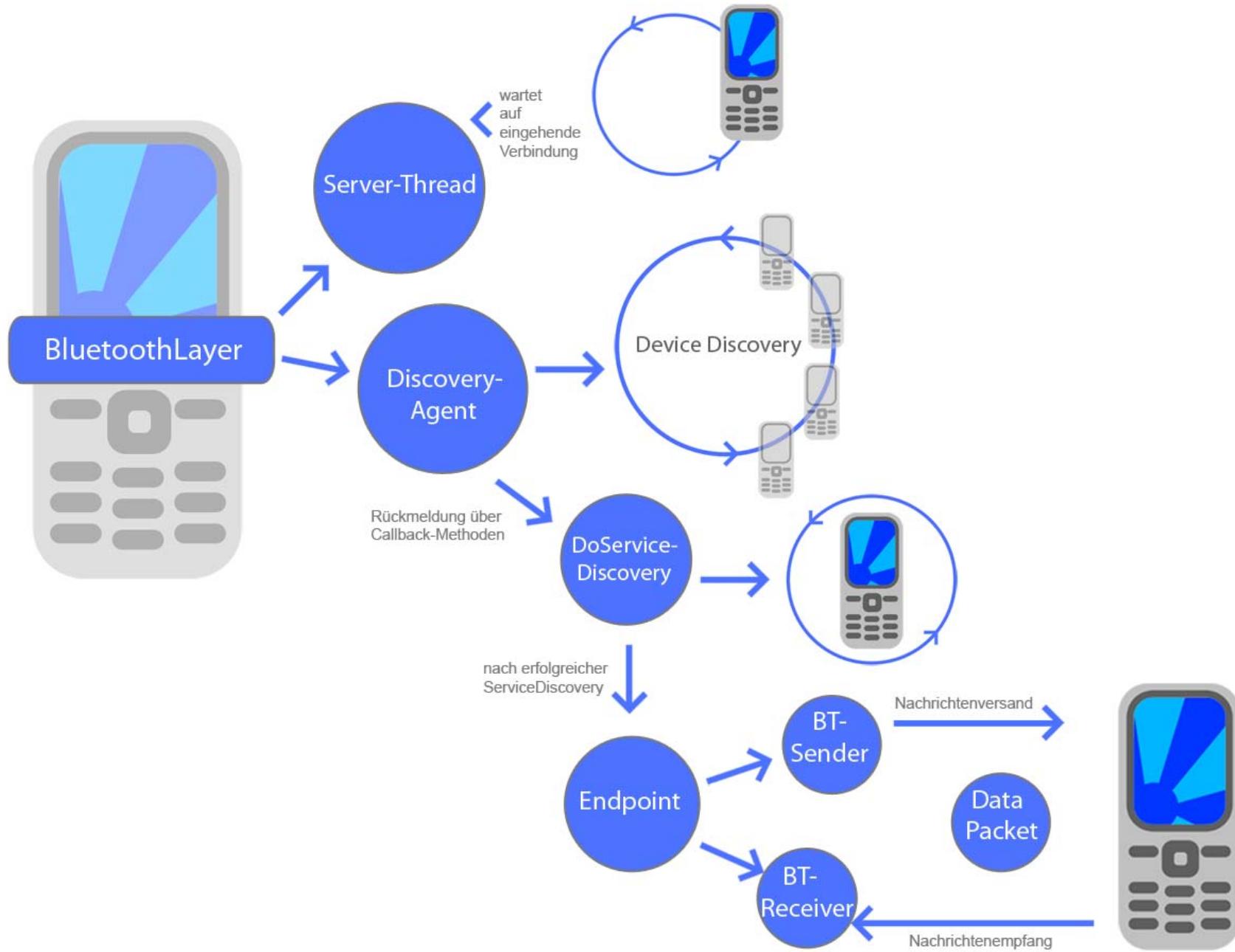
Alle Klassen für die Kommunikation über Bluetooth und SMS

Klasse Bluetoothlayer:

- erzeugt Server-Thread der auf eingehende Verbindung wartet
- startet die Device- und Servicediscovery
- prüft ob mit Gerät bereits kommuniziert wurde
- prüft ob Gerät auf Blacklist ist

Nach erfolgreicher Suche:

- erzeugt Endpoint-Objekt (Verbindung zu anderem Gerät)
- startet Reader- und Senderthread für Kommunikation



Die Endgeräte:



DIE ENDGERÄTE

Die Endgeräte: Nokia 6230



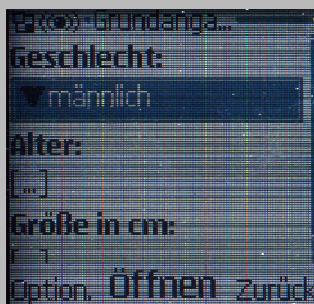
- Kam Ende 2003 auf den Markt
- Abmessungen: 103 x 44 x 20 mm
- 16 Bit Display
- Auflösung 128 x 128 Pixel
- Interne Kamera mit 640 x 480 Pixel
- 3,5 MB Speicher
- davon 512 KB als Heap nutzbar
- CLDC 1.1, MIDP 2.0
- Wireless Messaging API
- Mobile Media API
- Bluetooth API
- Nokia UI API

Die Endgeräte: Siemens S65



- Kam im Herbst 2004 auf den Markt
- Abmessungen 109 x 48 x 18 mm
- 16 Bit Display
- Auflösung 132 x 176 Pixel
- Interne Kamera 1 Megapixel
- 11 MB Speicher
- davon 1,5 MB als Heap nutzbar
- CLDC 1.1, MIDP 2.0
- Wireless Messaging API
- Mobile Media API
- 3D API
- Location API
- Bluetooth API

Die Probleme: Nokia 6230/MMAPI



Wenn man versucht ein Foto der eigenen Person zu machen stürzt das Programm sofort mit einer ClassCastException ab.

```
java.lang.ClassCastException: com.sun.media.sound.RealTimeSequencer
at com.nokia.phone.sdk.concept.mirrors.mma.media.player.MidiPlayer.doRealize(MidiPlayer.java)
at com.nokia.phone.sdk.concept.mirrors.mma.media.player.TonePlayer.doRealize(TonePlayer.java)
at com.nokia.phone.sdk.concept.mirrors.mma.media.player.BasicPlayer.realize(BasicPlayer.java)
at com.nokia.phone.sdk.concept.mirrors.mma.media.MMAManager.init(MMAManager.java)
at com.nokia.phone.sdk.concept.mirrors.mma.media.MMAManager.createPlayer(MMAManager.java)
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:39)
at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:25)
at java.lang.reflect.Method.invoke(Method.java:585)
at com.nokia.phone.sdk.concept.gateway.e.a(e.java)
at com.nokia.phone.sdk.concept.gateway.e.e.run(e.java)
at java.lang.Thread.run(Thread.java:595)
```

Die Probleme: Nokia 6230/MMAPI



MM API-Implementierung von Nokia unvollständig.

Keinerlei Unterstützung für:

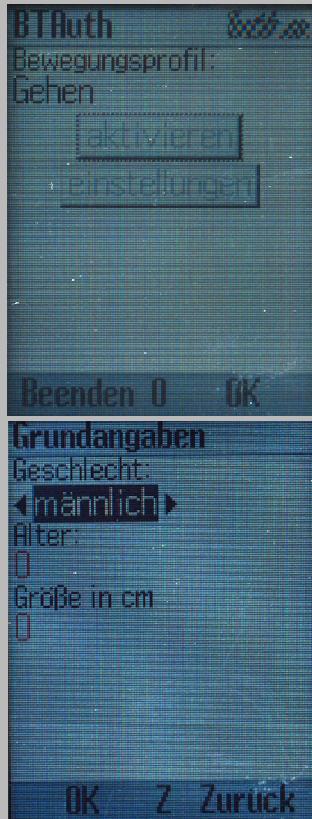
- Aufnahme von Foto oder Video
- Aufnahme von Audiodaten
- Abspielen von „echten“ Audiodaten

Einige Funktionalität: Abspielen von MIDI
(audio/x-tone-seq, audio/midi, audio/sp-midi)

Da die FileConnection API nicht unterstützt wird, auch
kein Nachladen des Fotos von Speicherplatte möglich.

>> Nokia 6230 für die Applikation unbrauchbar!

Die Probleme: Siemens S65/MMAPI



Implementiert die MMAPI (fast) vollständig, jedoch nicht fehlerfrei

- Aufnahme nur im JPG-Format
- Kleiner Fehler beim Darstellen des Videoltems
- ItemCommands funktionieren nicht bei Videoltem
- Audioaufnahme nur im AMR – Format

Schwerer Fehler bei Audioaufnahme:

Werden Audiodaten in ein byte[] aufgezeichnet, sind die Daten korrupt und nicht abspielbar. Es wird eine Exception wegen des unbekannten Datenformats geworfen.

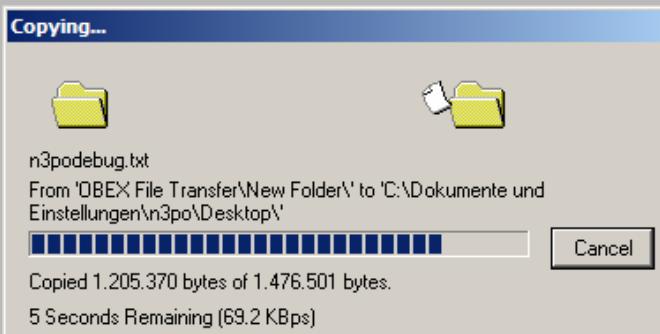
Die Probleme: Siemens S65/MMAPI

Lösungsmöglichkeit Audio-Bug:

- Siemens File API der 45 und 55-Generation existiert und funktioniert auch auf S65
- Audiodaten ins Filesystem schreiben
- File in byte[] auslesen
- Daten verschicken
- Audiodaten abspielen

Somit ist der VoiceChat via Bluetooth mit dem Siemens S65, wenn auch über den Umweg über das Filesystem, voll funktionsfähig.

Die Probleme: Siemens S65/Bluetooth



Auffällig niedrige Übertragungsraten
via Bluetooth bei Siemens S65



- Nokia mit 69,2KBps
- Siemens mit 7,2 KBps

Die Probleme: Siemens S65/Bluetooth

Probleme mit der Siemens Bluetooth-API:

Zu Beginn mit Firmware v12:

Absturz des Programms mit „Error Code 4“ sobald auf die Bluetooth-Funktionalität zugegriffen wird.

Nach Update auf Firmware v25:

Läuft, aber instabil.

Zeigt nach einiger Zeit einen „internen Bluetooth-Fehler“ an, welcher die BT-Funktionalität völlig deaktiviert.

Übertragung von Text funktioniert einwandfrei, bei größeren Daten (Foto) bricht die Übertragung ab.

Die Probleme: Siemens S65/Bluetooth



Siemens S65 mit offensichtlich korrupten
Bilddaten

Fehlersuche:

- Foto nach Aufnahme ins Filesystem speichern
- Logfiles ins Dateisystem schreiben
- Auswertung der Daten am Desktop-Rechner

Die Probleme: Siemens S65/Bluetooth

Ergebnis: Ab 500 – 800 byte: reproduzierbare Abbrüche

Lösung:

Daten wurden in kleinere Blöcke zerteilt (1024, 512, 256, 128 byte) und verschickt.
Erst als byte für byte einzeln verschickt wurden, kamen die Daten fehlerfrei an.

Neuer Fehler:

- Übertragung vom Client zum Server ausreichend schnell
- Vom Server zum Client unerträglich langsam

Die Probleme: Siemens S65

Daten Senden:

...

```
BTSender:: daten geschrieben: 2300 bytes. Zeit: 3688 ms
BTSender:: daten geschrieben: 2400 bytes. Zeit: 3808 ms
BTSender:: daten geschrieben: 2500 bytes. Zeit: 3932 ms
BTSender:: daten geschrieben: 2600 bytes. Zeit: 4071 ms
BTSender:: daten geschrieben: 2700 bytes. Zeit: 4191 ms
BTSender:: daten geschrieben: 2800 bytes. Zeit: 4316 ms
```

Daten Empfangen:

...

```
BTReceiver:: daten empfangen: 3300 bytes. Zeit: 123646 ms
BTReceiver:: daten empfangen: 3400 bytes. Zeit: 127431 ms
BTReceiver:: daten empfangen: 3500 bytes. Zeit: 131105 ms
BTReceiver:: daten empfangen: 3600 bytes. Zeit: 134903 ms
BTReceiver:: daten empfangen: 3700 bytes. Zeit: 138563 ms
```

Die Probleme: Siemens S65/Bluetooth

Geschwindigkeit vom Client zum Server: **ca 640 byte/s**

Geschwindigkeit vom Server zum Client: **ca 28 byte/s**

-> Übertragung eines ca. 4 Kb großen Fotos in über 2 Minuten!

-> Update der Firmwareversion auf v43

Die Probleme: Siemens S65/Bluetooth

Nach Update auf v43:

Übertragung ca. 1 KB/s in beide Richtungen

- Übertragung in max. 100 byte großen Blöcken, sonst Abbrüche

Somit funktioniert die Übertragung sämtlicher Daten mit dem S65, auch der VoiceChat per Bluetooth.

Weitere Bugs in der Siemens Bluetooth API verhindern (derzeit) jedoch den Einsatz der Applikation im Alltag:

Die Probleme: Siemens S65/Bluetooth

Die Sicherheitseinstellungen der BT-Verbindung werden ignoriert
(authenticate=false ; authorize=false;)

Deshalb: Pairing mit jedem Gerät, mit dem Kommuniziert wird (auch Profilüberprüfung), ausserdem Bestätigung der Verbindung durch den Benutzer.

Abhilfe: Profil bzw. Kennwort in Bluetooth-Service-Description eintragen.
Diese ist vor dem Verbindungsauftbau abfragbar.

Somit ist eine Bestätigung und Pairing nur nötig, wenn zwei Personen zueinander passen.

- > Leider ist die ServiceDescription auf dem S65 immer null.
- > Keine weitere Möglichkeit das Problem zu lösen.

Fazit



**DIE IDEE WAR GUT,
ABER DIE WELT NOCH NICHT BEREIT.**