

Universität Erlangen-Nürnberg
Informatik 7:

Profilbasierte Rekonfiguration kleiner Endgeräte



ROSES

Gerhard Fuchs,
Sébastien Truchat, Falko Dressler

gerhard.fuchs@informatik.uni-erlangen.de

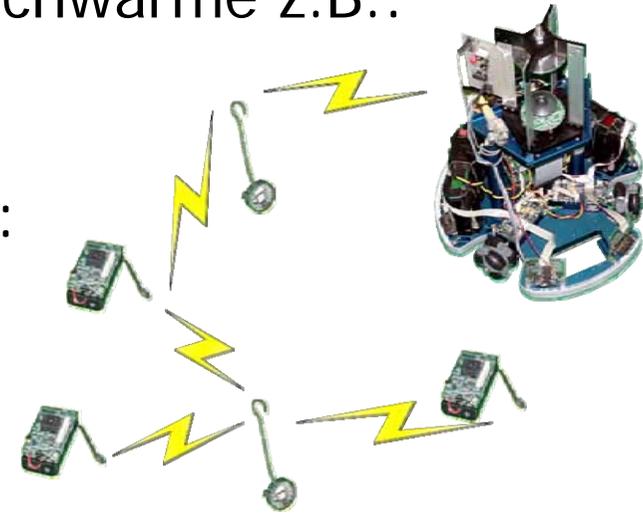
**2. Fachgruppengespräch
„Ortsbezogene Anwendungen und
Dienste“**

Inhalt

- ROSES-Projekt (RObot Assisted SEnsor NetworkS)
- Ziel
- Szenario
- Entwickelter Profile-Matching-Mechanismus
- Benötigte Anpassungen
- Zusammenfassung

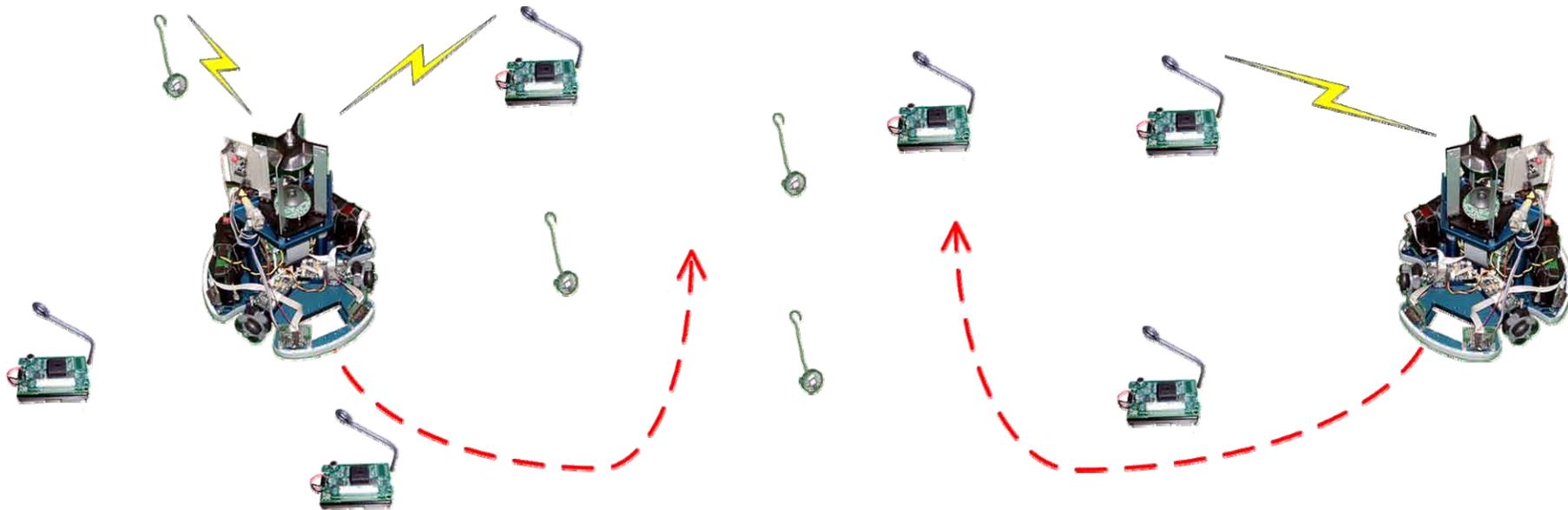
ROSES-Projekt

- Sensornetzwerk unterstützte Roboterschwärme z.B.:
 - ▶ Lokalisierung, Navigation
 - ▶ Kommunikationsinfrastruktur
- Roboter unterstützte Sensornetze z.B.:
 - ▶ Wartung, Reparatur, Rekonfiguration
 - ▶ Kommunikationsrelais
- Forschungsziele
 - ▶ Mobile Sensornetze
 - ▶ Optimierte **Aufgabenverteilung** und Kommunikation nach Anwendungs- und Energiegesichtspunkten
 - ▶ **Energieeffiziente** Arbeitsweise, Kommunikation und Navigation
 - ▶ Sensorunterstützte **Lokalisierung und Navigation**
 - ▶ **Dienstgütegewahre** Kommunikation in heterogenen mobilen Netzen mit dynamischer Topologie
 - ▶ **Sichere** Kommunikation und Datenmanagement in mobilen Sensornetzen

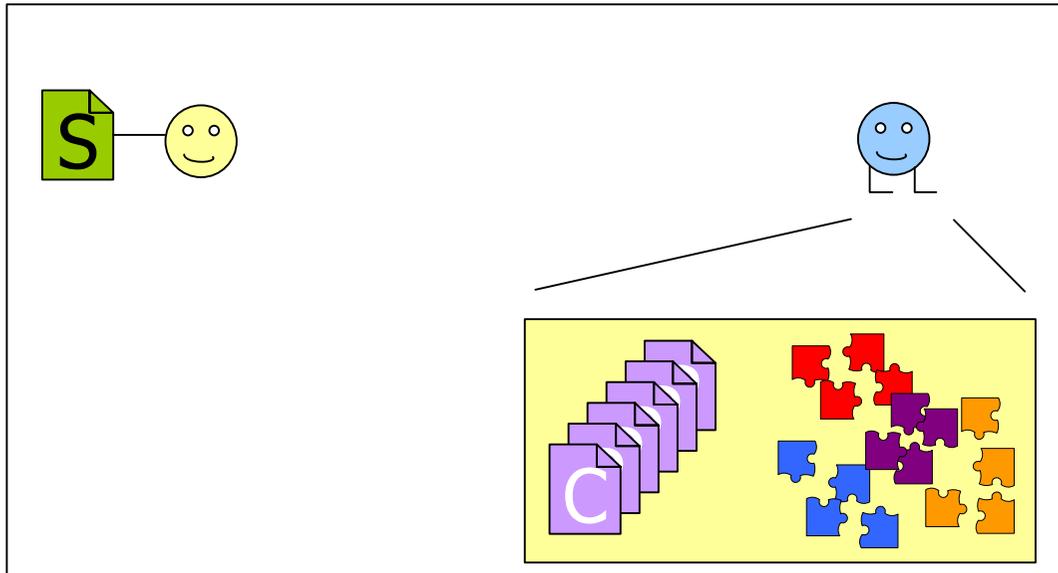


Ziel

- Wartungsarbeiten am Sensornetzwerk ausgeführt von Robotern
 - ▶ Konfiguration
 - ▶ Rekonfiguration
 - ▶ Programmierung



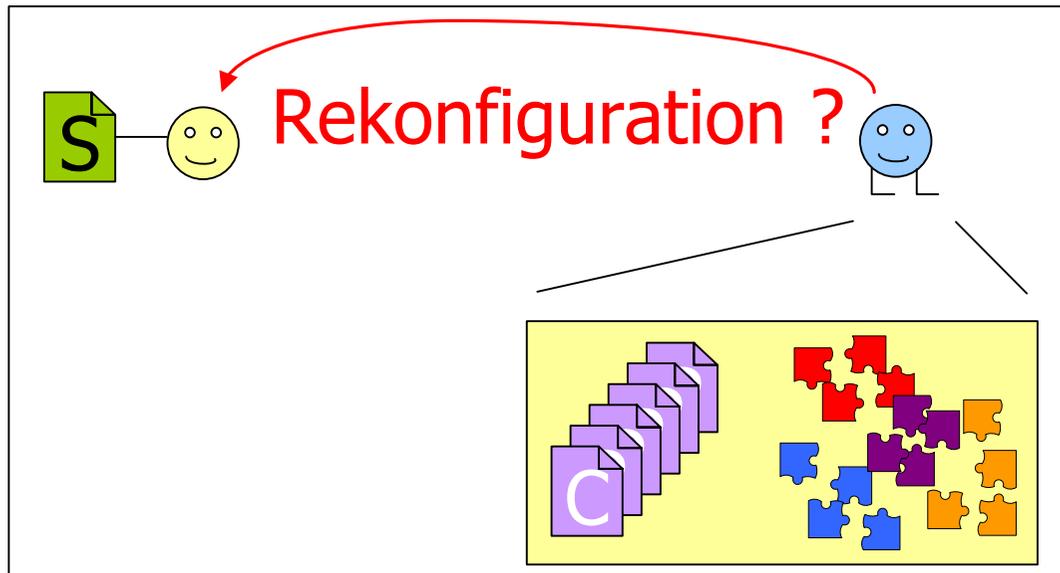
Szenario – Ausgangssituation



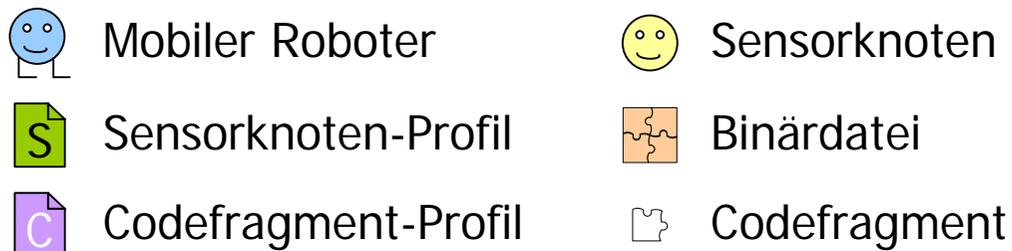
- Sensorknoten sind mit Profilen beschrieben (Typ, Sensoren ...)
- Auf dem Roboter sind Codefragmente mit entsprechenden Profilen



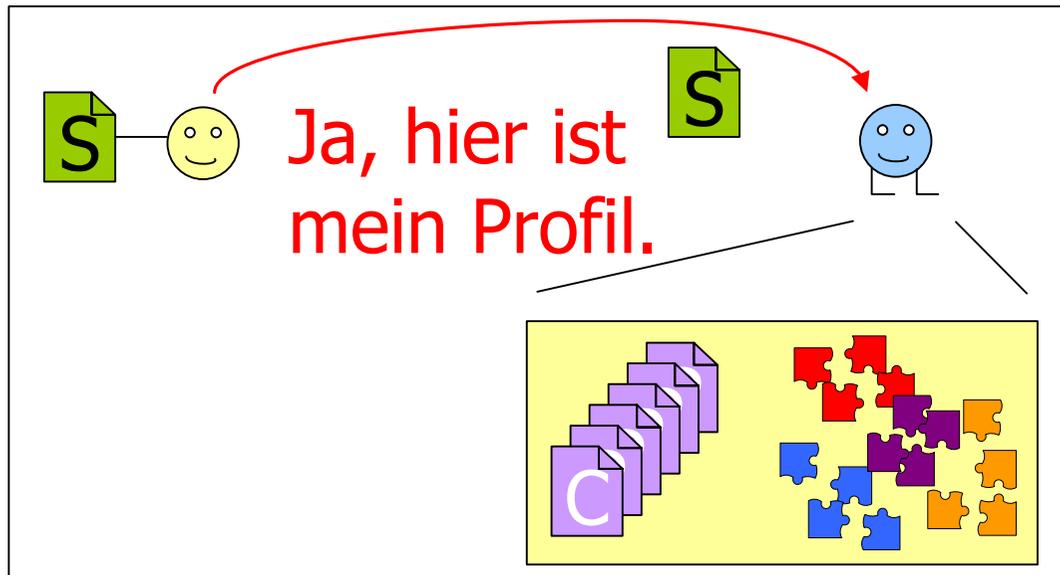
Szenario – Initiierung



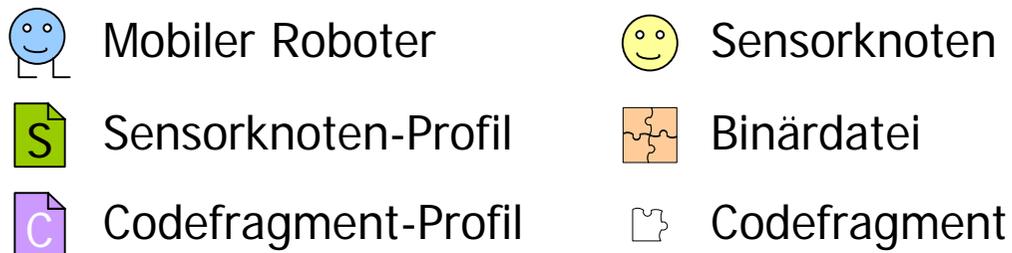
- Roboter verkündet, dass eine Rekonfiguration nötig ist



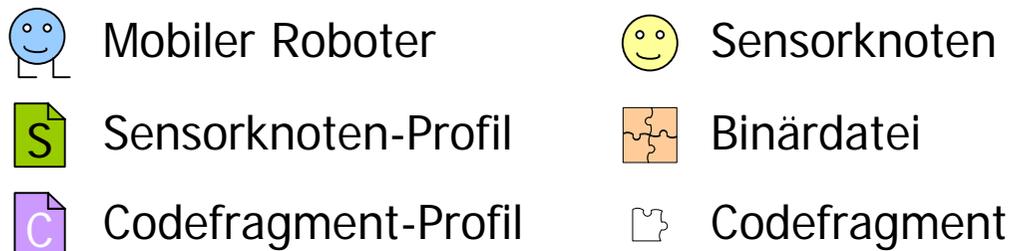
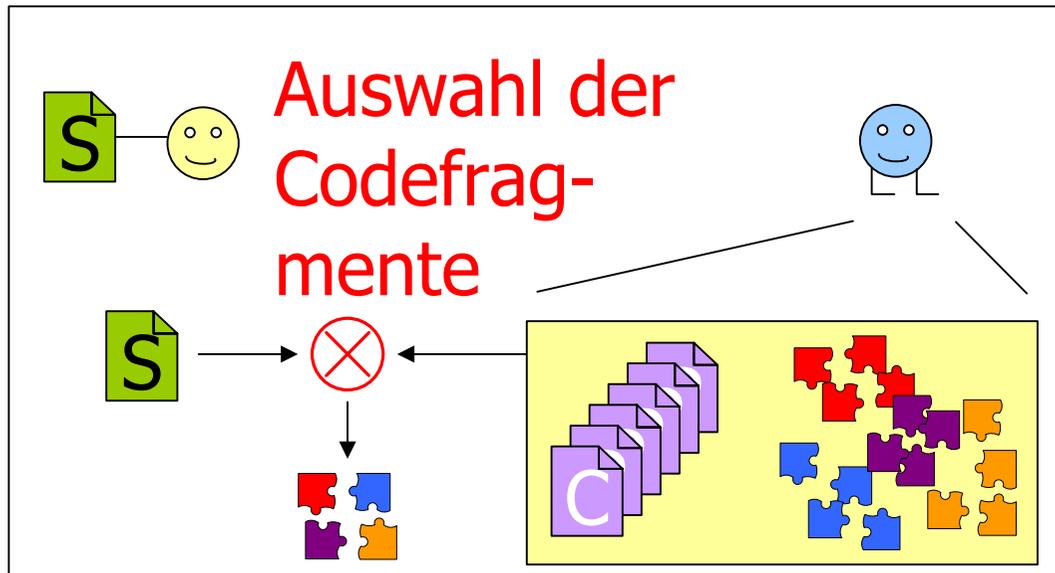
Szenario – Bestätigung



- Sensorknoten sendet sein Profil an den Roboter

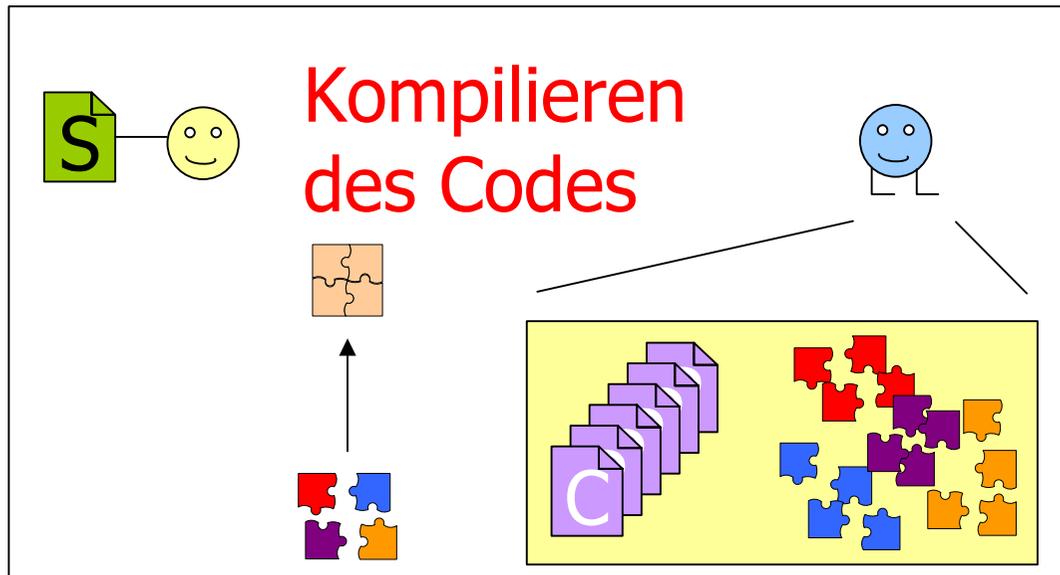


Szenario – Profile Matching

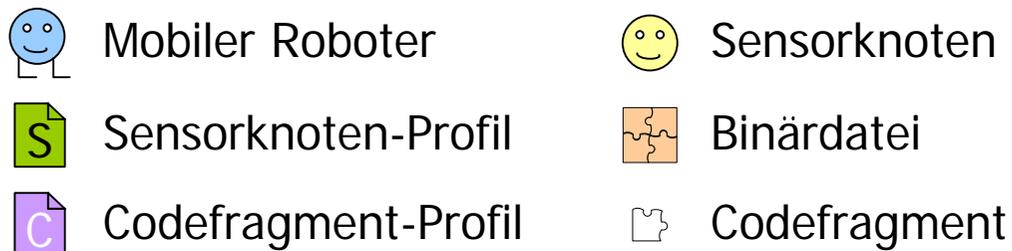


- Auswahl der Codefragmente durch den Roboter
- Anforderungen der Codefragmente müssen von den Eigenschaften des Sensorknotens erfüllt werden

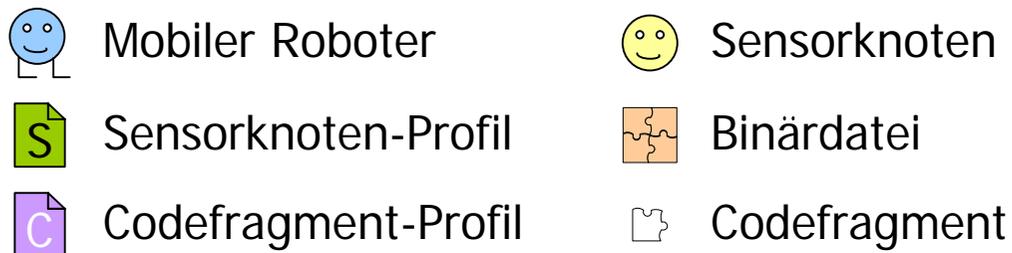
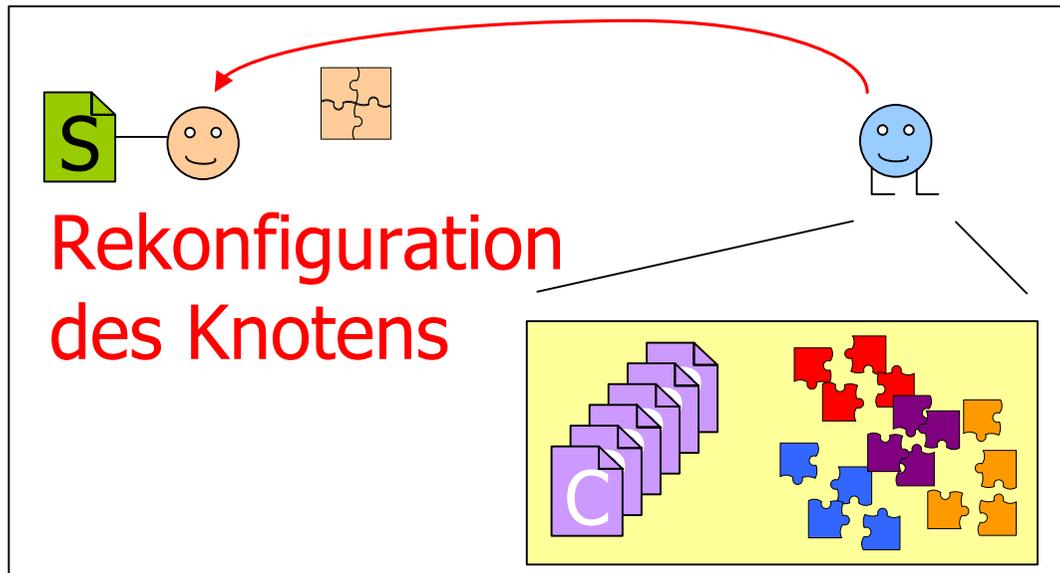
Szenario – Content Adaption



- Kompilieren der ausgewählten Codefragmente
- Ergebnis: Für den Sensorknoten angepasste Software

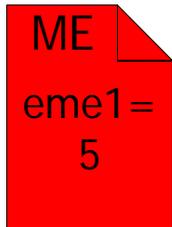


Szenario – Rekonfiguration

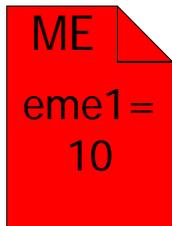


- Binärdatei wird auf den Sensorknoten übertragen
- Neustart des Sensorknotens
- Rekonfiguration ist abgeschlossen

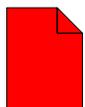
Entwickelter Profile-Matching-Mechanismus



beschreibt
→



beschreibt
→



Hardwareprofil

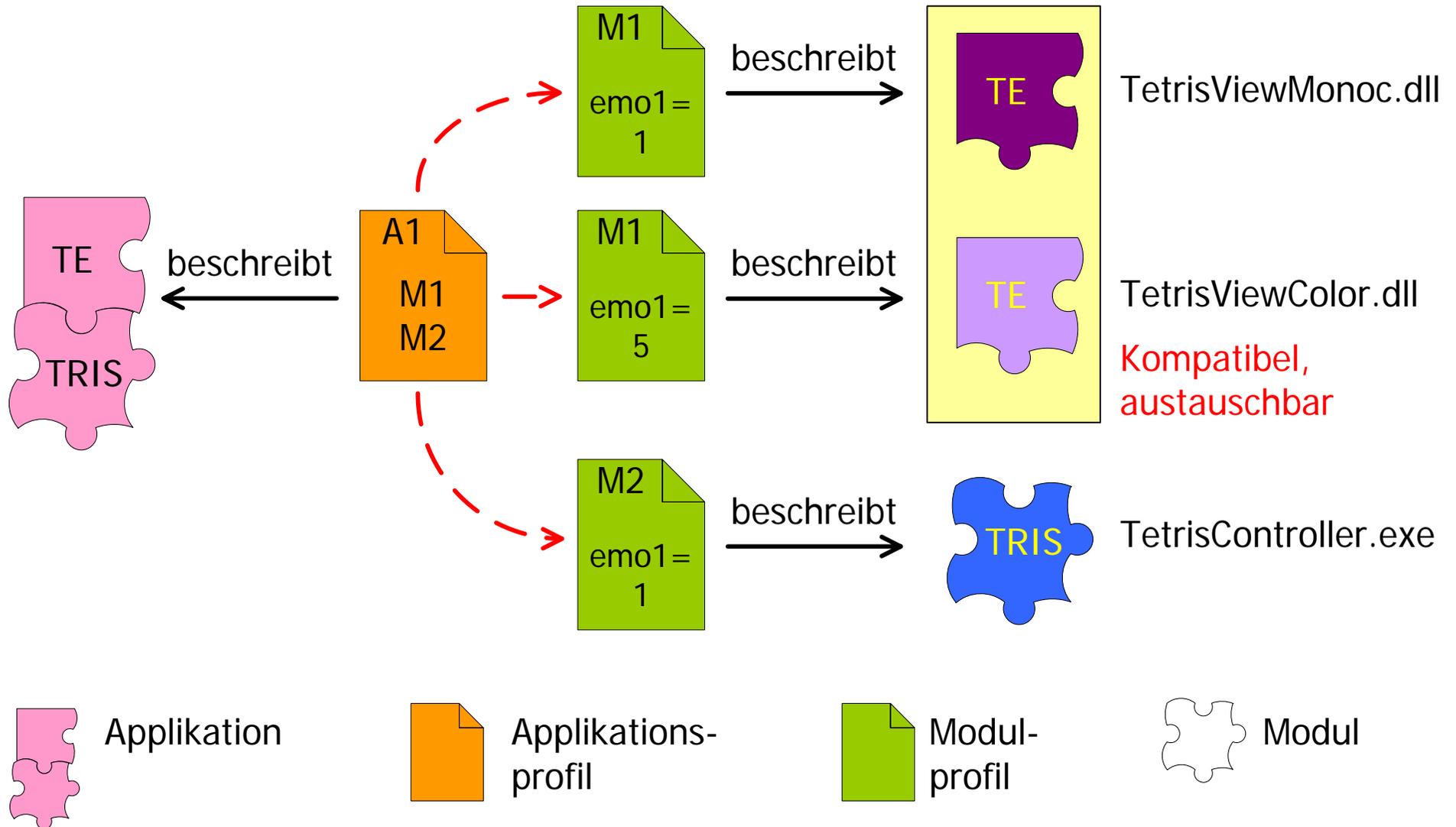
■ Ein Profil

- ▶ beschreibt ein Objekt
- ▶ mit Eigenschaften
- ▶ und / oder Anforderungen

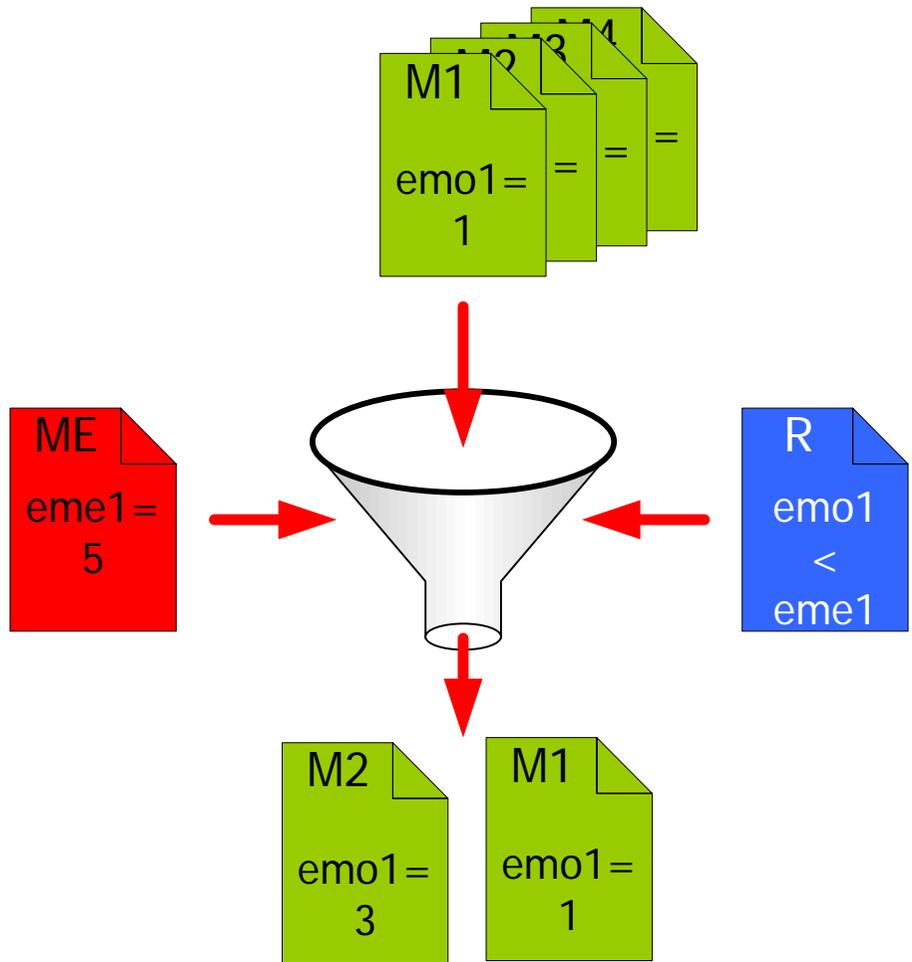
■ Beispiel

- ▶ Objekt = PDA
- ▶ Profil – Eigenschaften:
 - RAM = 64MB
 - RES = 640 x 480 dpi
 - OS = Windows CE

Entwickelter Profile-Matching-Mechanismus



Entwickelter Profile-Matching-Mechanismus



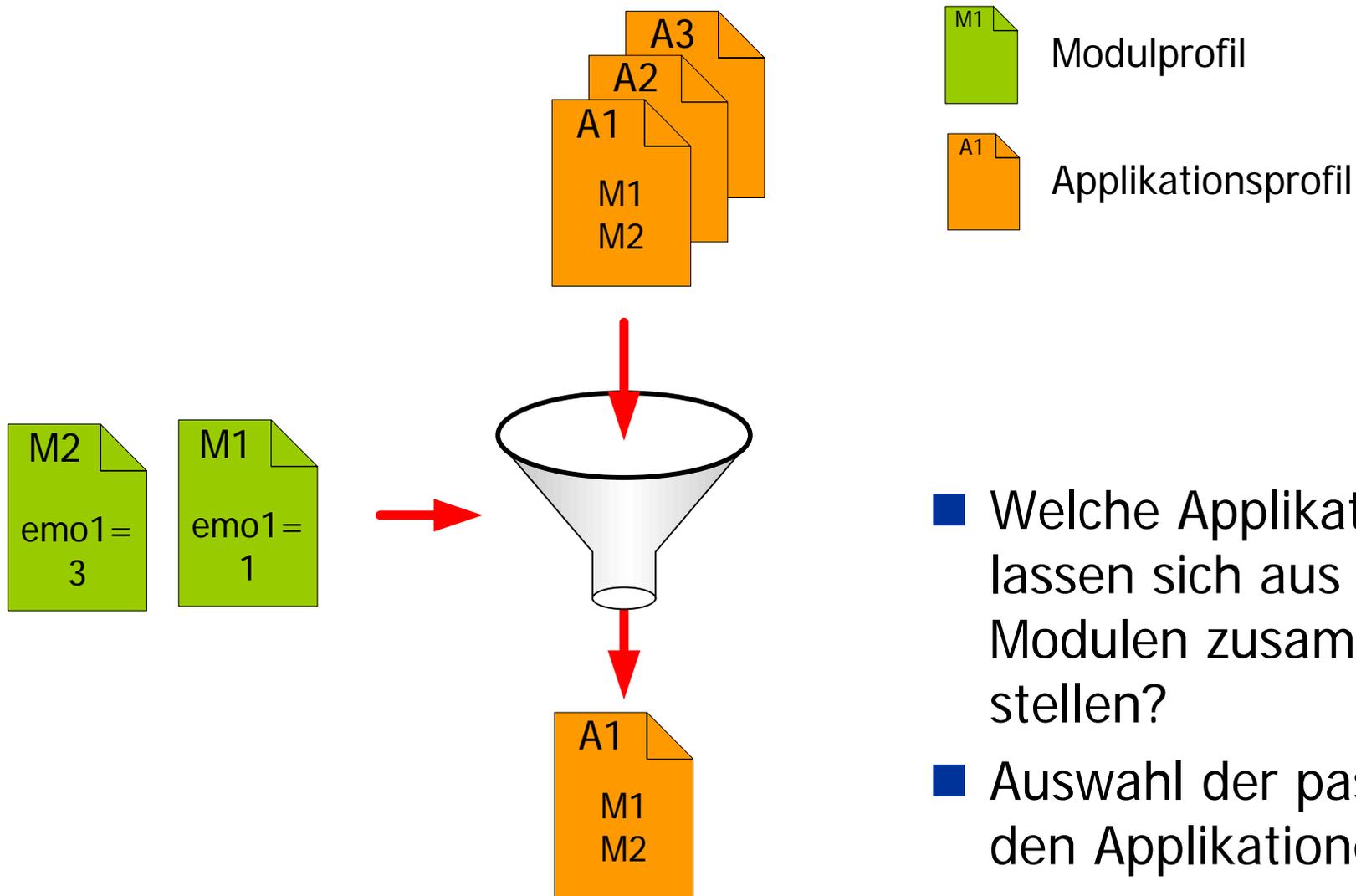
Matching-Regeln

Hardwareprofil

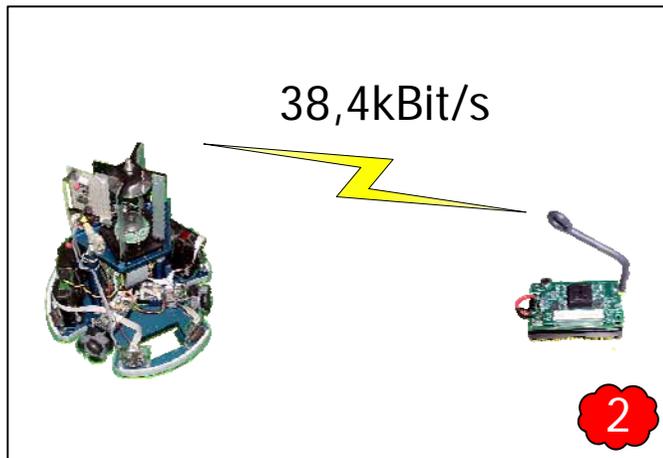
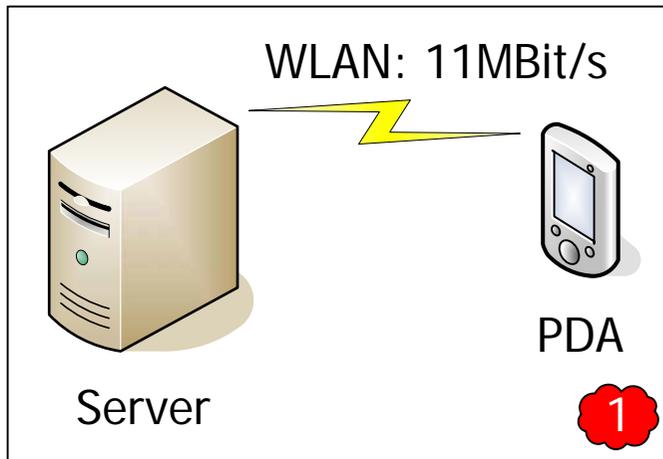
Modulprofil

- Welche Module laufen auf dem Endgerät?
- Auswahl der passenden Module

Entwickelter Profile-Matching-Mechanismus



Benötigte Anpassungen



■ Prototyp: 1

- ▶ Server (stationär) – Java
- ▶ PDA (mobil) – Java (64MB)
- ▶ WLAN 11MBit/s
- ▶ Profile CC/PP (Composite Capabilities / Preferences Profil)

■ Neues Szenario: 2

- ▶ Server (mobil) – Embedded Linux
- ▶ Sensorknoten (stationär)
 - 128kB Flash Memory
 - AT45DB041B
 - TinyOS
- ▶ Komm.: 38,4kBit/s

Zusammenfassung

- Ortsabhängige Rekonfiguration als Prototyp realisiert
- Ressourcenschonenderes Datenformat für die Profile nötig
- Möglichst vollständige Verlagerung des Profile-Matchings auf den Server
- Java → Kompilieren für jeden einzelnen Knoten

➔ **Ressourcenschonendere** Umsetzung + Content Adaption ist nötig!