



18. ITG Fachtagung Mobilkommunikation
Osnabrück, 16. Mai 2013
Heinz-Josef Eikerling, Nazar Selo Gavan



RULE-BASED LOCAL KNOWLEDGE NAVIGATION AND DECISION SUPPORT ON SMARTPHONES

Inhalt

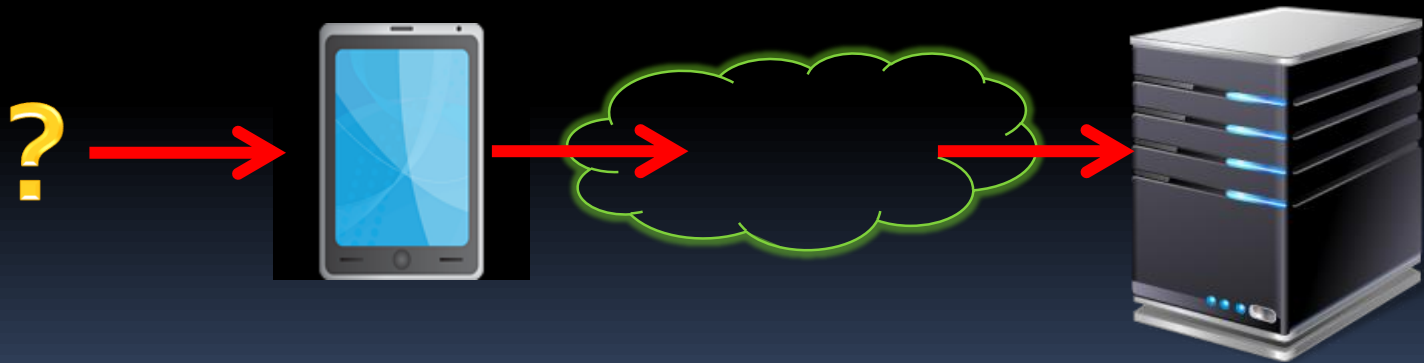
- Einführung
 - Motivation, Ziele / Anwendung
 - Herausforderungen
- Konzepte
 - Architektur
 - CLIPS / DSL
 - Integration Rule-Engine für Android
- Tests & Auswertungen
- Zusammenfassung
- Ausblick: Anwendungen

Motivation

- Mobile Geräte, insbesondere Smartphones, werden immer leistungsfähiger
 - Bsp.: Apple / Samsung A6 (**Dual Core**, 2 x ARM 9 Kern, 1,2 GHz → iPhone5), Qualcomm Snapdragon SoC (**Quad Core**, 4 x 1.5 GHz Krait → Google Nexus 4)
 - Speicher: 1 GB RAM (iPhone5), 2 GB RAM (Nexus 4), 16 – 64 GB Flash
- Nutzen: **kurzzeitig rechenintensive** Anwendungen können lokal ausgeführt werden

Local Knowledge Navigation

- Informationsabfragen via Smartphone
- Suchanfragen können komplex sein: *Wie wird das Wetter zum Segeln?*



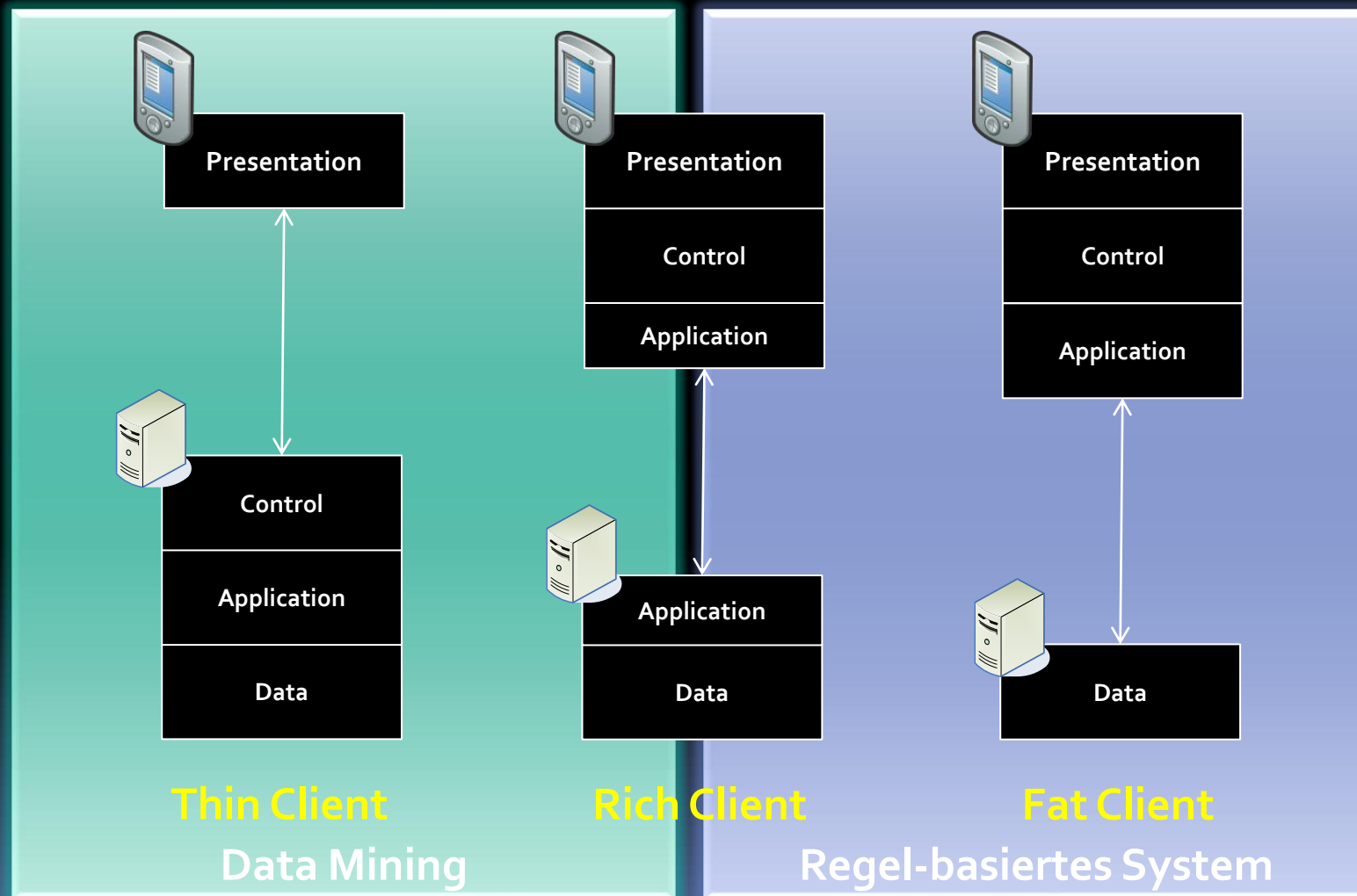
Local Knowledge Navigation

- Komplexe Suchanfragen erfordern die **Interpretation** von Suchanfragen
- Server-basierte Implementierungen:
 - iOS: Siri
 - Android: Google Mobile Search
- Problem:
 - Suchanfragen können protokolliert werden
 - Zumindest theoretisch ist eine **Zuordnung** zu einem Nutzer **möglich**
 - **Keine Anonymität, keine Privatsphäre!**

Möglichkeiten der Realisierung

- Case-based reasoning
- Logik höherer Ordnung
- Neuronale Netzwerke
- Data mining (→ **Suchmaschinen**):
 - Speicherung und Verarbeitung *großer Datenmengen* (= Wissen, Daten-Korpus)
 - Ableitung von Zusammenhängen aus Korrelationen der rel. Häufigkeiten von Schlagworten
 - Eigenschaften: on-line, server-basiert, hohe Wissensdiffusion
- **Regel-basierte Systeme:**
 - **Eigenschaften: off-line, server-unabhängig, geringe Wissensdiffusion**

Architektur-Alternativen



Rule-Engines auf mobilen Geräten

- Speziell interessant für Java / Android: JSR-94
- Kompatible Engines (JEE / Business Rule Engines):
 - Drools
 - JRules
 - Nicht lauffähig unter Android (iOS geht gar nicht)
- Lösung:
 - *C Language Integrated Production System (CLIPS)*

CLIPS Regeln

- Bedingungen (linke Seite) triggern Aktion (rechte Seite), LISP Syntax
- Format:

```
(defrule rule-name [<comment>]
  [<declaration>]           ; rule properties
    <condition>* ; left-hand side (LHS)
=>
    <action>* ) ; right-hand side (RHS)
```

CLIPS Datenstrukturen

- Datenstrukturen modellieren Fakten
- Beispiel für *Datenstruktur*:

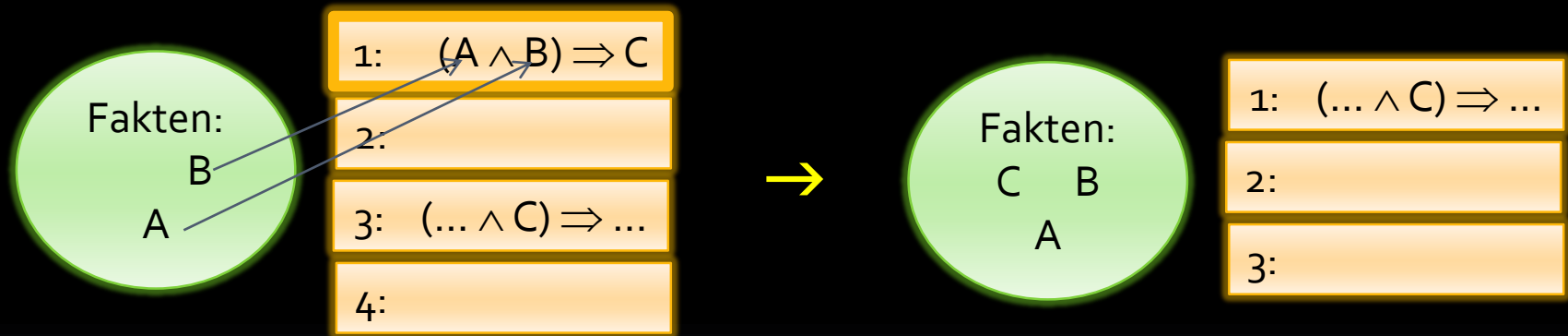
```
(deftemplate person "person template"  
  (slot name      (type  STRING ))  
  (slot age       (type  INTEGER))  
  (slot gender    (type  SYMBOL) (default Male))  
)
```

- Beispiel für *Faktum*:

```
(defacts persons  
  (person (name "Maria") (age 21) (gender  
Female)))  
)
```

CLIPS Eigenschaften

- **Vorwärtsverkettende** Rule-Engine:
 - Regelmenge wird kontinuierlich auf durch Fakten erfüllte Prämissen untersucht
 - Erfüllte Regel erzeugt neues Faktum (\rightarrow Aktion)
 - Reihenfolge der Regelauswertung wird über Agenda-festgelegt



- Implementiert den **RETE**-Algorithmus (C. Forgy, 1982)
- Definition von **Domain-Specific Languages** per ANTLR möglich:
 - Wurde für Testzwecke erstellt.

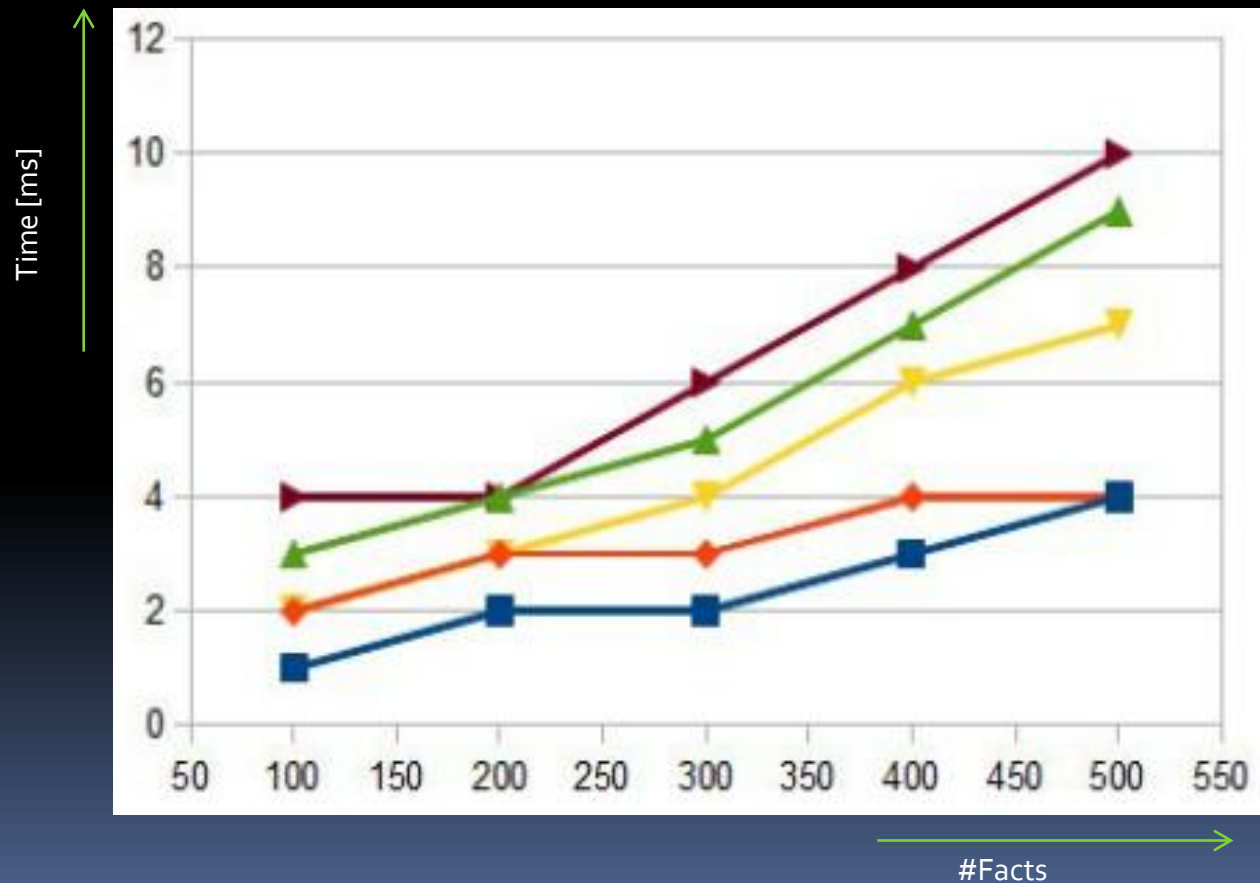
Rule-Engine: Effizienzbetrachtung

- Eigenschaften Vorwärtsverkettung:
 - Arbeitet Fakten-getrieben
 - Aufwand **linear** in der Größe der Wissensbasis
 - U.U. werden aber für die Anfrage irrelevante Ableitungen durchgeführt
- Rückwärtsverkettung:
 - Versuche, ein Faktum **q** aus der Wissensbasis abzuleiten
 - Analyse der Implikationen, die zu **q** führen
 - Arbeitet **ziel-gerichteter**, Aufwand für Ableitung vielfach **geringer als linear** in der Größe der Wissensbasis

→ **Besser für mobile Geräte (weniger Berechnungen / Energieverbrauch)**

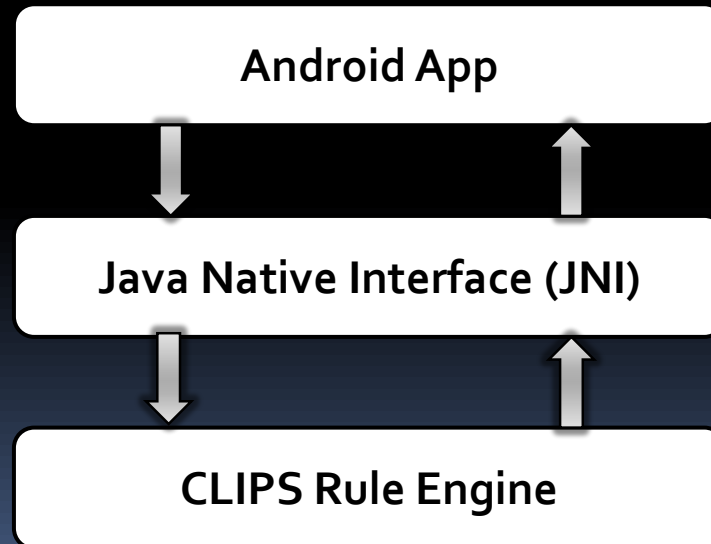
Tests

- Laufzeit steigt linear mit der Anzahl der Fakten / Regeln (für Samsung Galaxy, 1 GHz Single Core)



Integration CLIPS für Android (*ACLIPS*)

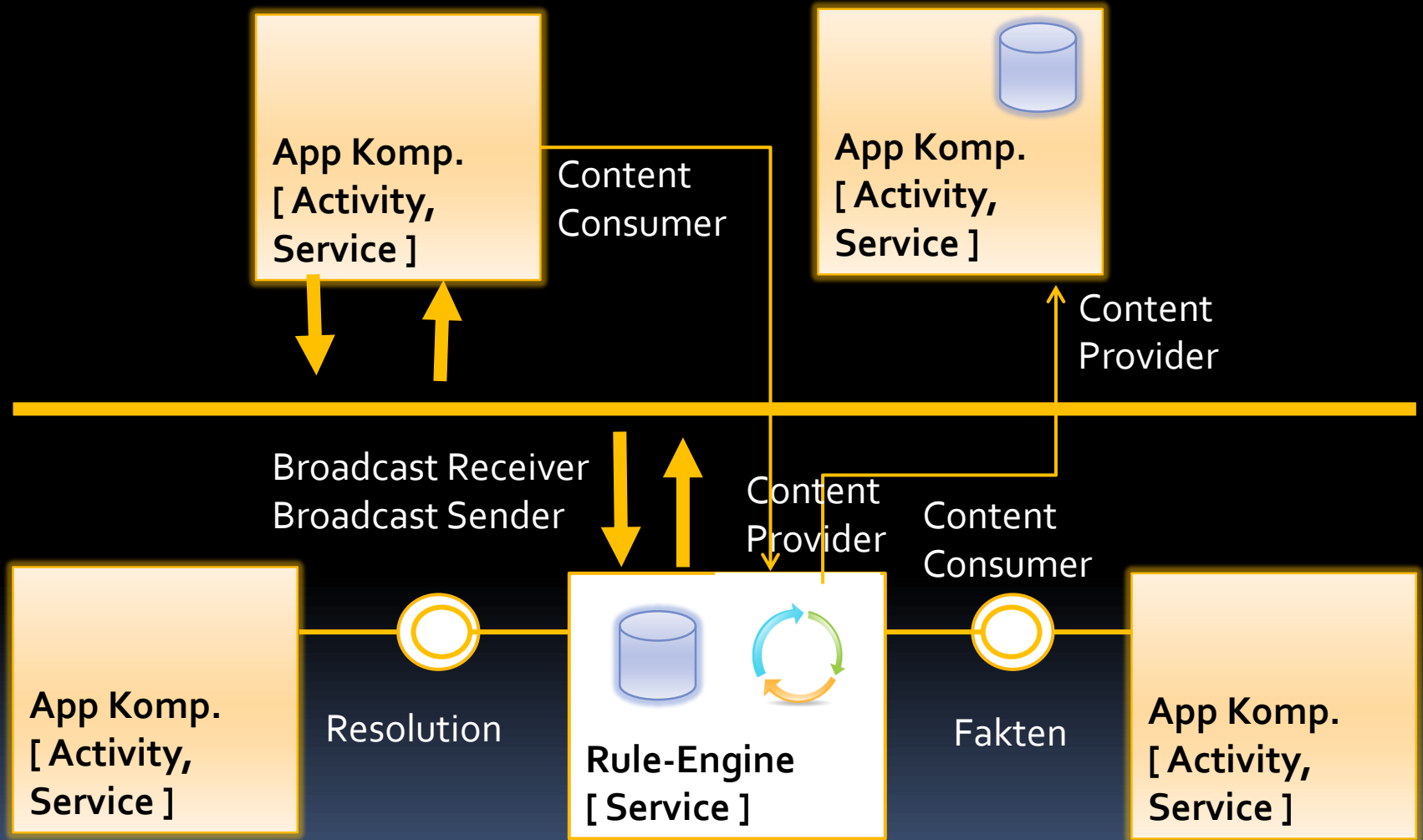
- Integration als **native** Bibliothek
- Wrapper-Klassen (Java) zum Aufruf der CLIPS-Funktionen



Einbettung ACLIPS in Apps

- **Service:**
 - Rule-Engine arbeitet im Hintergrund
 - Service-Zugriffe zur Einstellung von Fakten und Abfrage von Resolutionen
- **Broadcast Receiver:**
 - Applikationskomponenten registrieren sich für Ereignisse (Auslösung durch Rule-Engine)
 - Rule-Engine registriert sich für Ereignisse (= *Fakten*)
- **Content Provider:**
 - Resolutionen können per Query abgefragt werden
- **Content Consumer:**
 - Fakten werden von anderen Komponenten bereit gestellt

Einbettung ACLIPS in Apps



Fazit

- Ansatz (*Local Knowledge Assistant, ACLIPS*):
 - Regel-basiertes Software-System zur Entscheidungsunterstützung auf mobilen Geräten
 - Implementierung: Adaption von CLIPS für die Android Plattform
 - Entwicklung einer DSL zur Erstellung von Regel-Basen
- Weiterführende Arbeiten:
 - Extraktion von Regel-Sätzen aus großen Datenbeständen
 - Kompakte Repräsentation zur Anpassung an Gegebenheiten auf mobilen Geräten

Ausblick: Anwendungen

- Medizinische Anwendungen:
 - Überwachung von Risiko-Patienten
 - Überwachung von chronisch Kranken (Allergien etc.)
- Finanzbereich:
 - Unterstützung von Investitionsentscheidungen
- Wartung von komplexen technischen Produkten
- Intermodaler Transport