Entwicklung von Software für drahtlose Sensor-Netzwerke

Jan Blumenthal, Dirk Timmermann
Universität Rostock

Informatik 2003
Frankfurt/Main, 2. Oktober 2003

Gliederung

- Projekt-Vorstellung
- Einführung zu Sensor-Netzwerken
- Software Engineering
- Middleware Ansatz
- Zusammenfassung

Projekt-Informationen

Projekt:

Middleware für mobile spontan vernetzte Sensornetzwerke

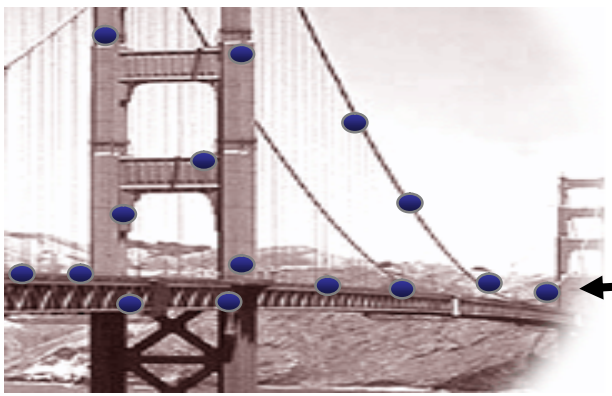
Ziele:

- Untersuchung von Verfahren:
  - Selbstkonfiguration von ad-hoc vernetzen Sensornetzwerken
  - Kooperatives Zusammenarbeiten unabhängiger Knoten
  - Fehlertoleranz
  - Kontextabhängigkeit der zur Verfügung stehenden Dienste
- Entlastung ressourcenarmer Sensorknoten
- Adaptierbare, wiederverwendbare Middleware für Sensor-Netzwerke
- Entwicklung einer Referenzinfrastruktur

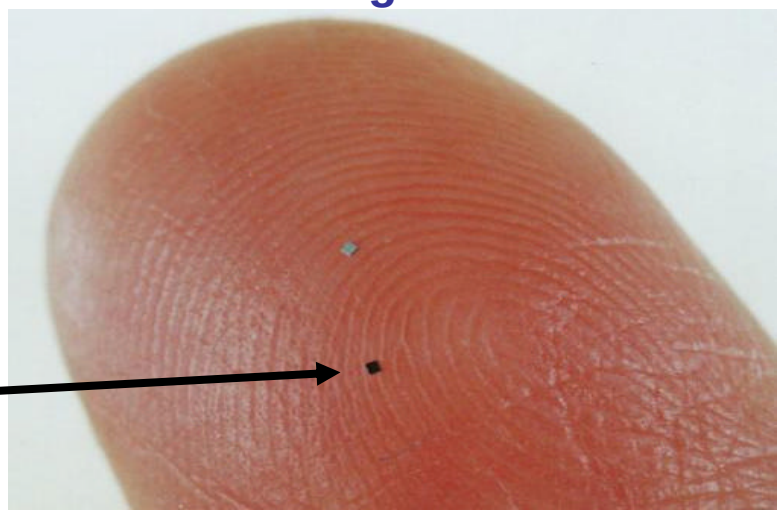
Charakteristik: Sensor-Knoten

- Limitationen:
  - Speicher (Flash/ROM: 8KB, RAM: 512 Byte)
  - Batterie (Lebenszeit: Tage – 10 Jahre)
  - Begrenzte Rechenfähigkeit
  - Übertragungsreichweite (5…20 m)
  - Datenraten: Bit/s … KB/s
- Übertragungsmethode:
  - Bluetooth
  - ZigBee
  - nanoNET
  - UWB (Ultra Wideband)

Beispiel: Materialüberwachung



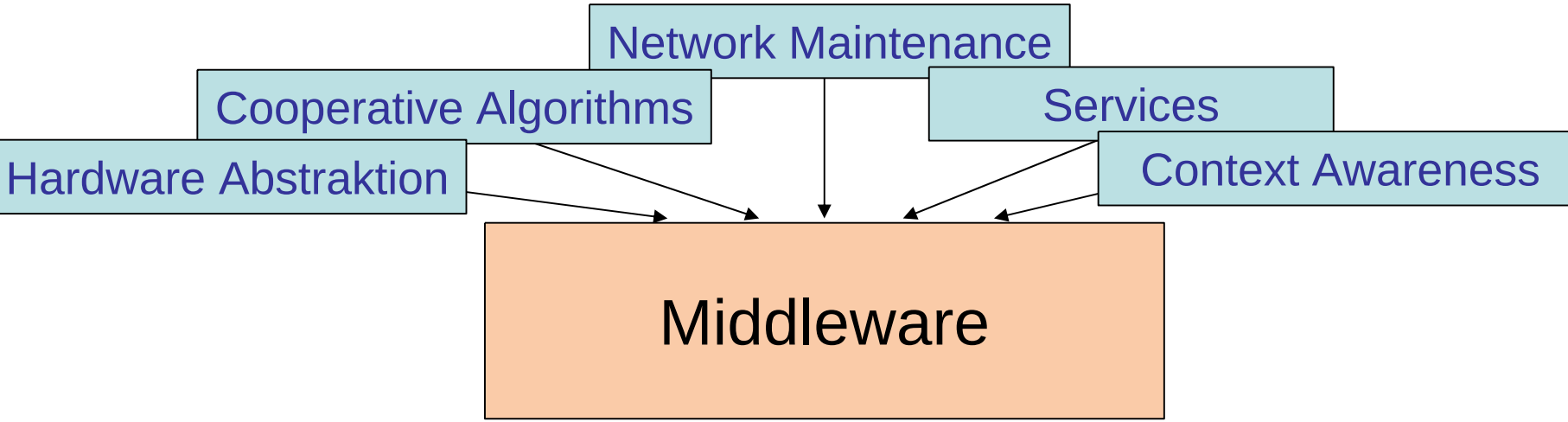
Zukünftige Sensorknoten



Charakteristik: Sensor-Netzwerk

1. Selbst-Organisation
   - Ad hoc Netzwerkbildung
   - Autonomer Verbindungsaufbau
   - Mobilitätsbehandlung
2. Netzwerk-Pflege
   - Adressierung von Knoten, Routing
   - Adaptive Reaktionen auf Umgebungsänderungen
   - Kompensation von Ausfällen
3. Kooperative Datenverarbeitung
   - Context awareness
   - Location positioning und location awareness
   - Messdatenreduktion

Software Engineering

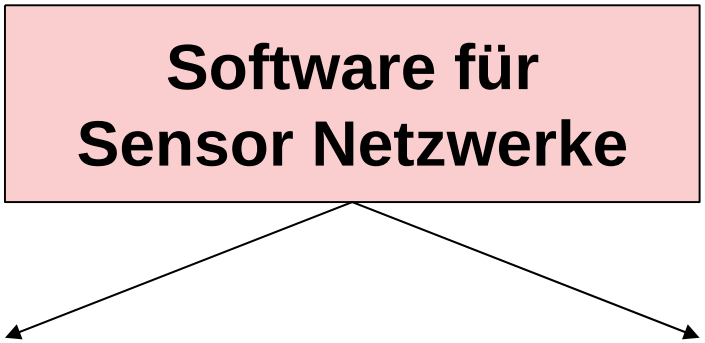


Middleware:

- Kapselung der Komplexität eines verteilten Systems
- Standardisierte Schnittstellen zur Knoten-Anwendung
- Remote-Access durch Services
- Anpassung der Software an dynamische Systemänderungen

Ziel: Ressourcen-optimierte service-orientierte Middleware für verschiedene Plattformen.

Software-Aspekte



Programmier-Aspekt

- Systemweites API
- Hardware-Abstraktion
- Verbergen der Heterogenität des verteilten Systems
- Optimierung von Schnittstellen

Pre-Deployment Phase

Verhaltens-Aspekt

- Zugriff auf entfernte Ressourcen ohne vorherige Kenntnis
- Anpassung der Anwendung an Umgebungsänderungen
- Task-Änderungen
- Evolution des Netzwerkes

Post-Deployment Phase

Middleware Eigenschaften

Skalierbare Middleware

- Anpassung von Datentypen während der Kompilierung
- Entfernen von ungenutzten Komponenten

Generische Middleware

- Plattform-unabhängige Software-Bibliotheken tendieren zu hoher Anzahl von komplexen Schnittstellen
- Schnittstellen-Optimierung (Interface-Optimization)

Nicht Generische Software

```
void setBaudrate(int handle, int baudrate)
{
  hardware_addr=getIOAddress(handle);
  hardware_addr->BTR0=baudrate;
}
```

Generische Software

```
void setBaudrate(int baudrate)
{
  // getIOAddress(handle);
  BTR0=baudrate;
}
```

Nur gültig bei Nutzung eines IO-Geräts

Einsparungen: Parameterübergabe, Funktionsaufruf, Stackoperation, Rückgabewert + Zuweisung, Feld-Operation

Middleware Eigenschaften II

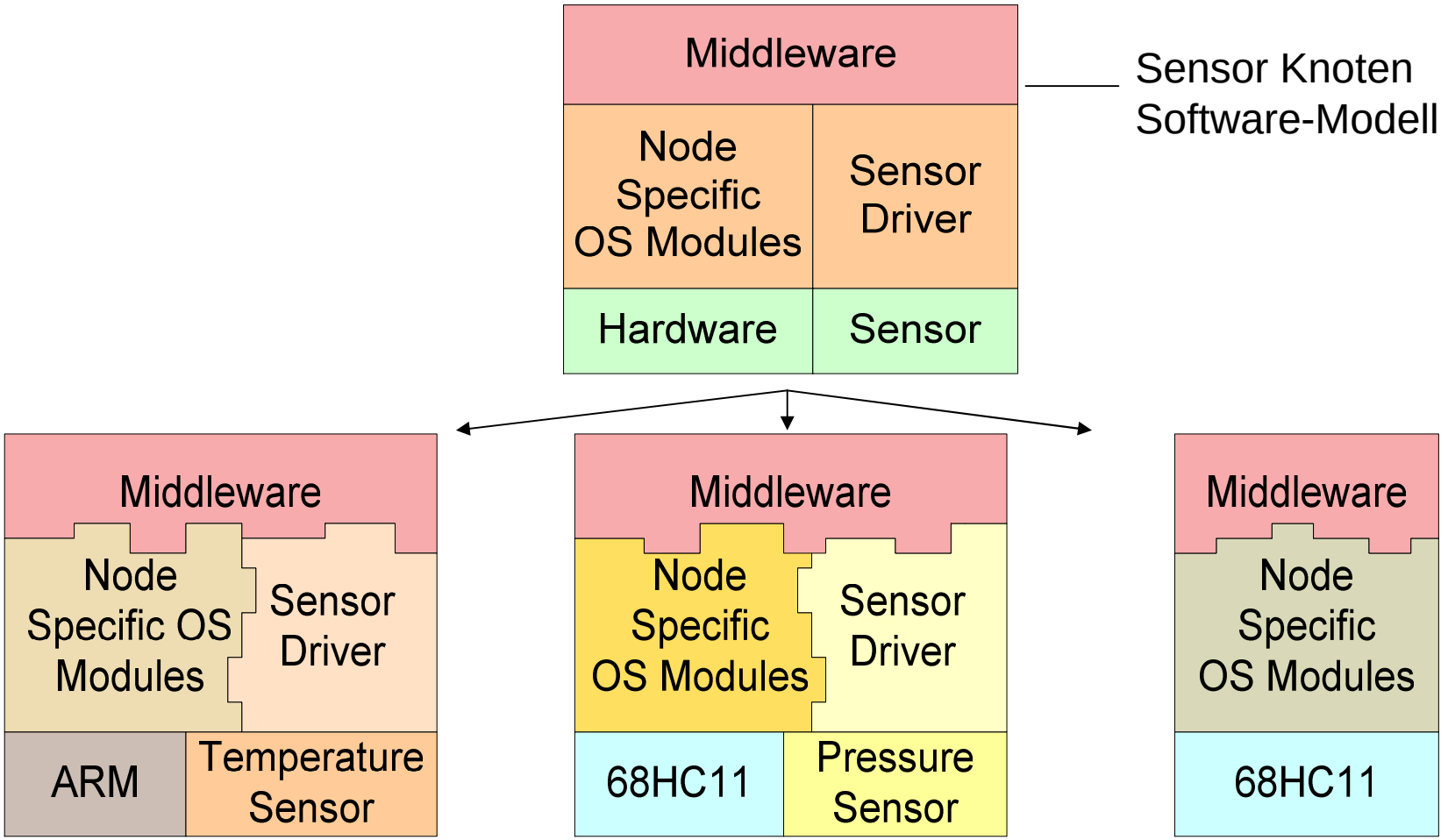
Adaptive Middleware

- Mobilität und Änderungen in der Umgebung erfordern Software-Anpassungen
  - Auswahl zum Clusterhead erfordert zusätzliche Routing-Software
  - Neue Messmethoden erfordern Programmänderungen
- Austausch und Ausführung von Komponenten zur Laufzeit
  - Start neuer Services
  - Download virtueller Programme

Reflektive Middleware

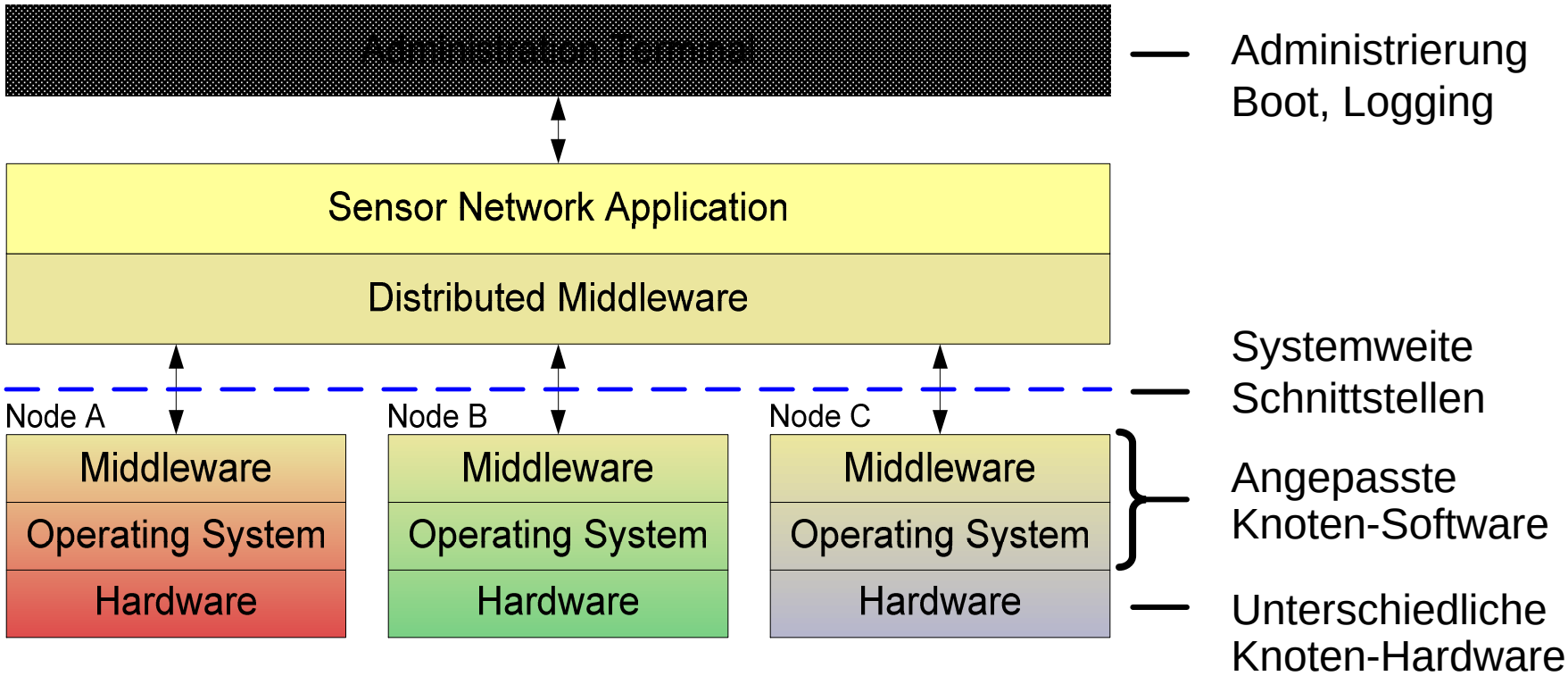
- Fähigkeit des Systems zur Selbstdiagnose und Einflussnahme
- Anwendungsschichten unbeeinflusst
- Inspection: Analysieren des Verhaltens via Logging / Debugging
- Adaptation: Anpassen des Verhaltens der internen Software-Schichten
- Beispiel: Änderung der Routing-Strategie in Abhängigkeit der Mobilität

Knoten-Anwendungen



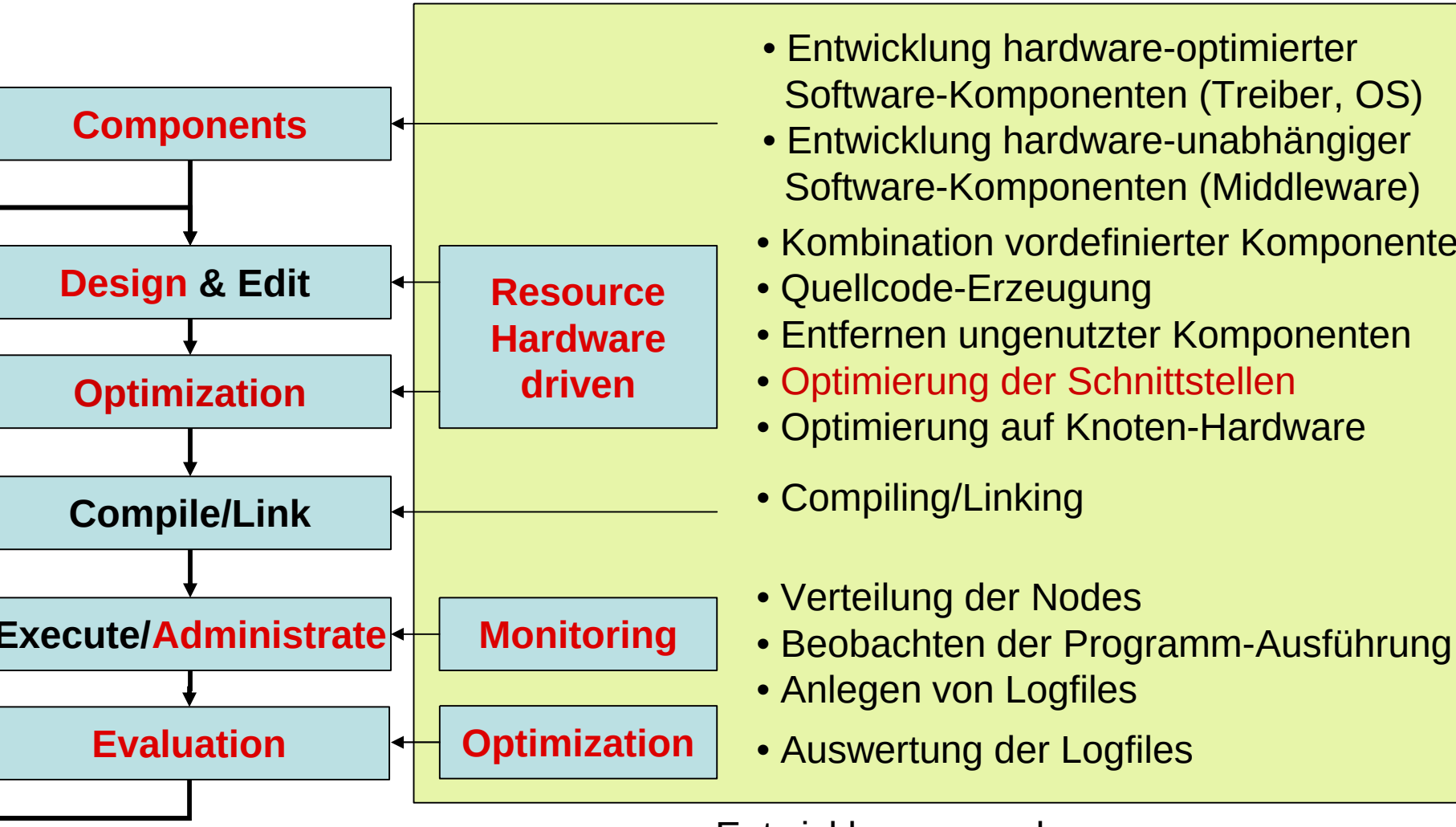
Änderungen der Schnittstellen an Bedürfnisse des Programms
anstatt
Änderungen des Programms an Bedürfnisse der Schnittstellen

Sensor Network Application



- Optimierung und Kompilierung der Middleware und des Betriebssystems für jede Hardware
- Sensor Network Application
  - Zusammensetzung aus allen Knoten-Services
- Optionales Administrierungs-Terminal zum Verwalten des Netzwerkes

Sensor Netzwerk Design



Änderungen gegenüber herkömmlichen Design-Methoden

Zusammenfassung

- Herausforderungen bei Software-Entwicklung für SN
- Design einer service-orientierten Middleware für SN
- Optimierungen für Knoten-Software basierend auf:
  - Interface-Design & Interface-Optimierung
  - Adaptive und reflektive Middleware
- Anliegende Arbeiten
  - Kriterien für Interface-Optimierung

Vielen Dank

Wo ist das Leck ?

- Feuchtigkeits-Sensorknoten in Sandsäcken
- Sammlung und Vorauswertung in Sensorknoten

Motto: „Nur der feuchte Sandsack kennt das Leck…