

# OSEK/VDX NM (Network Management)

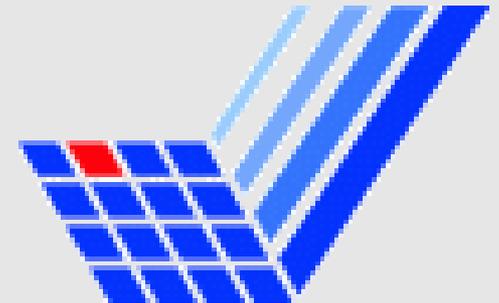
---

**Alexander Berger**

[alexander.berger@uni-dortmund.de](mailto:alexander.berger@uni-dortmund.de)

PG AutoLab

Seminarwochenende 21.-23. Oktober 2007



# Überblick

---

- Motivation
- Aufgaben des NM
- Architektur
- Konzept und Verhalten
  - Indirektes NM
  - Direktes NM
- Skalierbarkeit
- Ausblick
- Erfahrungen in der Massenproduktion
- Zusammenfassung

# Motivation

---

- Vernetzung von ECUs verschiedenster Hersteller
- standardisierte Schnittstellen und Protokolle sollen Funktionsfähigkeit zusichern
- Verhalten eines Knotens beeinflusst das Verhalten des gesamten Systems (und umgekehrt)
- Fehlfunktionen durch Beeinflussungen vermeiden

# Motivation

---

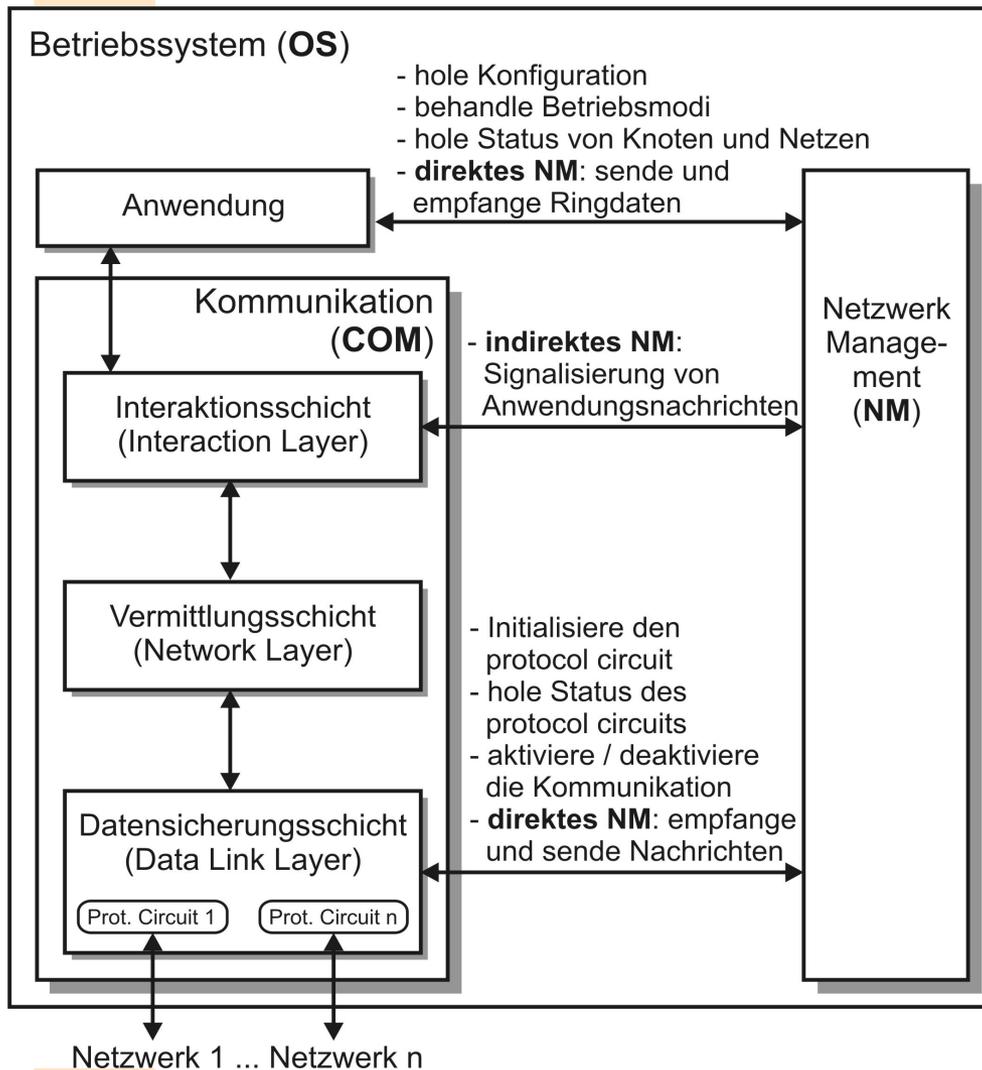
- Status des Netzwerks muss überwacht und ausgewertet werden
- Knoten müssen „Verhandlungen“ führen
- Ersparnis von Kosten und Entwicklungszeit
- Portabilität
- Auslagerung dieser Aufgaben in eine dedizierte Netzwerk-Management-Komponente

# Aufgaben des NM

---

- NM-Code von Anwendungscode entkoppeln
- Standard Management-Tasks übernimmt NM:
  - verschiedene Busse sind an einen  $\mu$ Controller anschließbar
  - Initialisierung der Kommunikationshardware
  - „start-up“ des Netzwerks & Steuerung der Betriebsmodi
  - Ermittlung der Netzwerk-Konfiguration
  - Unterstützung von Diagnose

# Architektur



## ■ NM hat Schnittstellen zur:

### ■ Anwendung

- API bietet Zugang zu NM-Funktionen

### ■ Interaktionsschicht

- Bietet NM Dienste zur Überwachung von Nachrichten der Anwendungen

### ■ Datensicherungsschicht

- Bietet NM Zugang zur Kommunikationshardware

# Konzept und Verhalten

---

- NM basiert auf dem Überwachen von Knoten
- Konfigurationsmanagement nutzt Knotenüberwachung zur Ermittlung der aktuellen Konfiguration
  - tatsächliche Konfiguration
  - Ziel-Konfiguration
  - „limp-home“-Konfiguration
- zwei verschiedene Methoden zur Netzwerküberwachung:
  - indirektes Knoten-Monitoring
  - direktes Knoten-Monitoring

# Indirektes NM – Knoten Monitoring

---

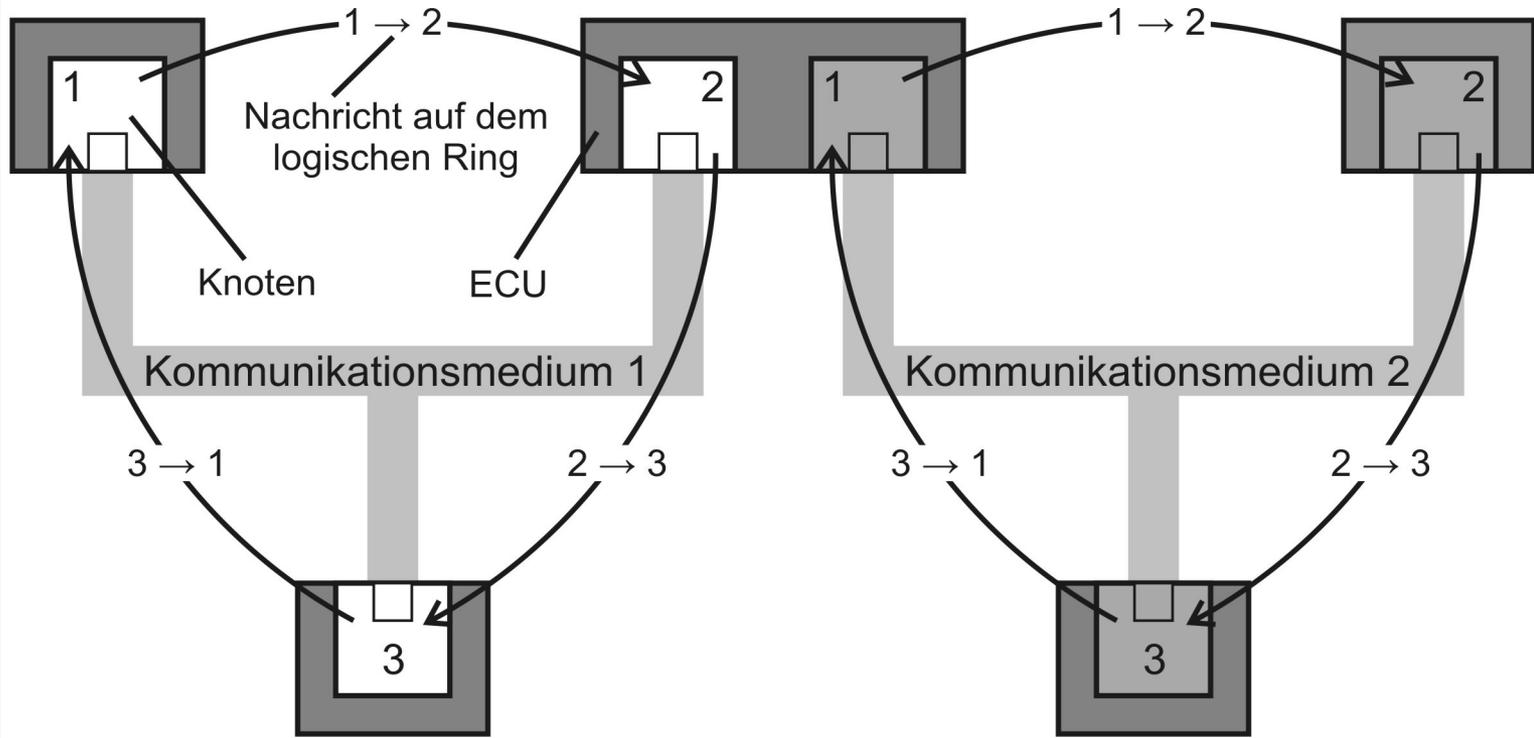
- Anwendungen tauschen Nachrichten in bestimmten Zeitabständen aus
- indirektes NM überwacht diesen normalen Nachrichtenaustausch
- Empfang und Senden von Nachrichten wird als Lebenssignal interpretiert
- wird über einen bestimmten Zeitraum keine Nachricht empfangen → Fehlfunktion des Knotens
- ideal bei sehr einfachen oder zeitkritischen Anwendungen
- Nachteil: Passiv → Nicht alle Aufgaben erfüllbar!

# Direktes NM – Knoten Monitoring

---

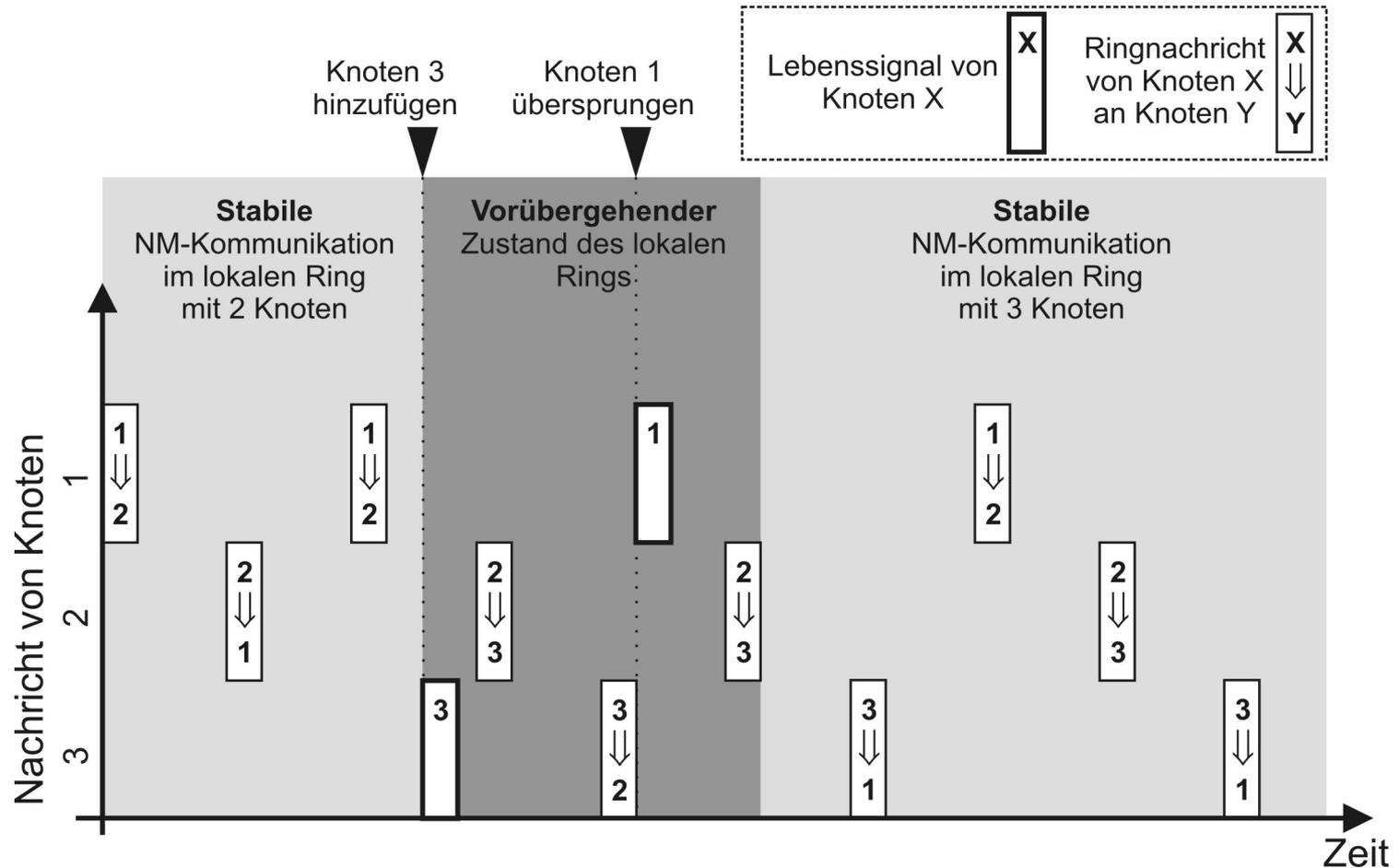
- Knoten überwachen sich gegenseitig
- Knoten senden und empfangen spezielle NM-Nachrichten
- Jeder Knoten sendet ein „Ich lebe“- („I am alive“-) Signal und empfängt die „Lebenssignale“ aller anderen Knoten.
- die „Lebenssignale“ werden „aufsummiert“  
→ gegenwärtige Konfiguration
- logischer Ring wird benutzt
- jeder Knoten bekommt eindeutige ID
- IDs folgen logischer Anordnung (z.B. 1,2,3,...)

# Direktes NM – logischer Ring

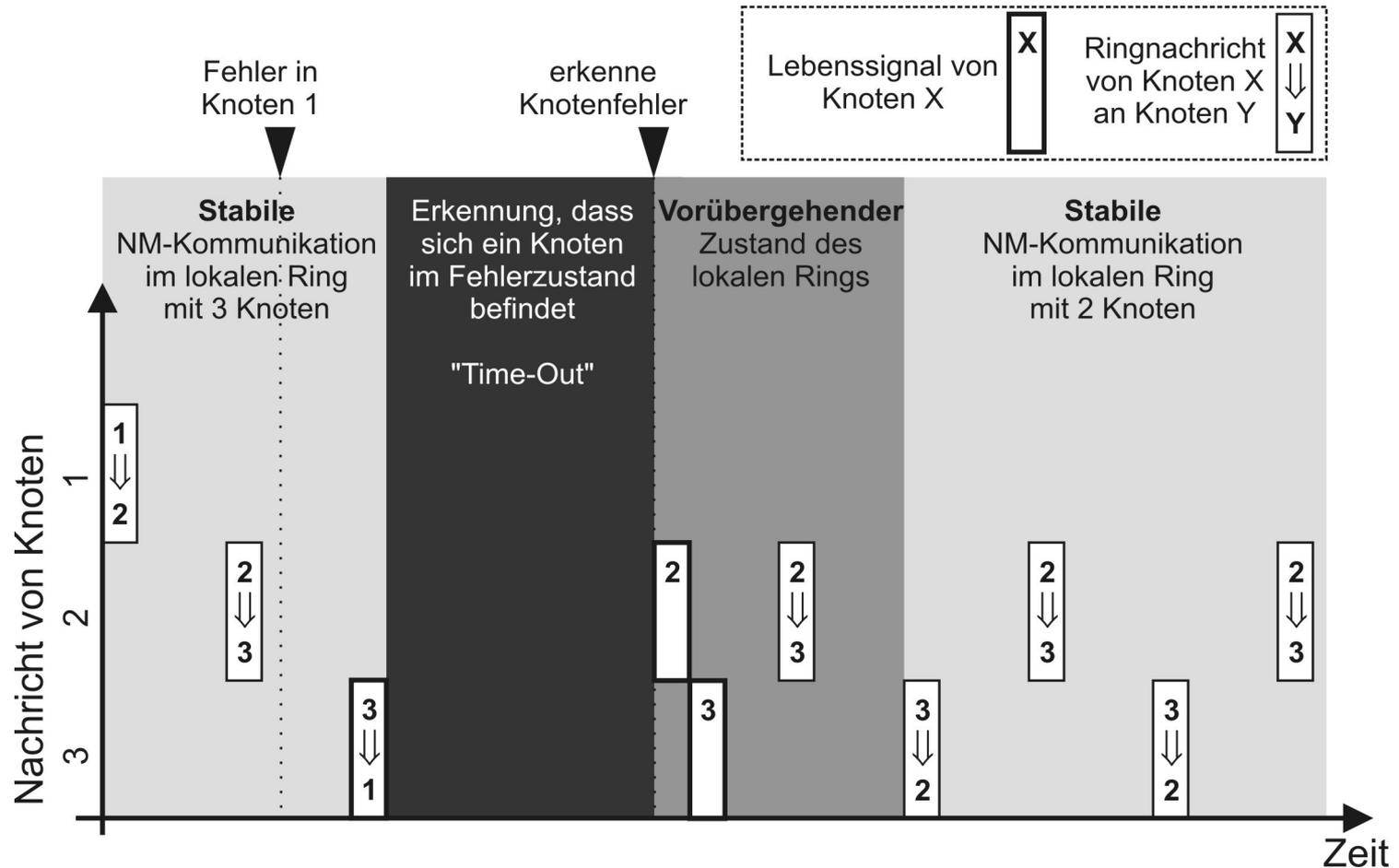


- Es gibt zwei Aktivitäten:
  - Integration von Knoten in den logischen Ring
  - Bestimmen von fehlerhaften Knoten und Neukonfiguration des logischen Rings

# Direktes NM – Verhalten des log. Rings



# Direktes NM – Verhalten des log. Rings



# Direktes NM – NM Protocol Data Unit (NMPDU)

- NM unterstützt die Interoperabilität von Knoten verschiedener Zulieferer
- Repräsentation von NM-Daten hat ein festgelegtes Format → NMPDU

Adressfeld		Steuerfeld	Datenfeld
Quell-ID	Ziel-ID	OpCode	Daten
Obligatorisch			Optional

Code	Bedeutung
0 0 0 0 0 0 0 1	Ringnachricht, Bussleep.ack = 0, Bussleep.ind = 0
0 0 0 0 0 1 0 1	Ringnachricht, Bussleep.ack = 0, Bussleep.ind = 1
0 0 0 0 0 x 1 1	Ringnachricht, Bussleep.ack = 1, Bussleep.ind = X
0 0 0 0 0 0 1 0	Lebenssignal, Bussleep.ind = 0
0 0 0 0 0 1 1 0	Lebenssignal, Bussleep.ind = 1
0 0 0 0 0 0 0 0	"limp home"-Nachricht, Bussleep.ind = 0
0 0 0 0 0 1 0 0	"limp home"-Nachricht, Bussleep.ind = 1

# Direktes NM – Übergang in Sleep-Mode

---

- jeder Knoten kann globalen Ruhezustand anstoßen
- sende Ringnachricht mit *Bussleep.ind* = 1
- andere Knoten im Netz:
  - reichen *Bussleep.ind* unverändert weiter
  - setzen *Bussleep.ind* zurück
- kommt *Bussleep.ind* beim Initiator unverändert (= 1) an  
→ sende Ringnachricht mit *Bussleep.ack* = 1
- alle Knoten empfangen diese Nachricht → wechseln (nach einer Wartezeit) in Ruhezustand

# Skalierbarkeit

---

- NM-Dienste auf praktisch jeder Art von Knoten verfügbar
  - low-cost Ein-Chip-Lösungen mit min. Speicher und Rechenleistung
  - high-tech-Lösungen mit praktisch unbegrenztem Speicher und gewaltiger Rechenleistung
- Spezifikation wurde in Kern-Dienste und optionale Dienste unterteilt
- Resultat: modulares NM, anpassbar an Speicher-  
ausstattung und Rechenleistung eines Knotens

# Ausblick

---

- Abbildung einer NMPDU auf eine CAN-Nachricht wird derzeit von den Kfz-Herstellern festgelegt
- zukünftige Anforderungen:
  - Subnetz-Betrieb
  - Gateway-Unterstützung
- Erste Schritte um diese Anforderungen zu erfüllen wurden bereits unternommen.

# Erfahrungen in der Massenproduktion

---

- Renault setzt **indirektes NM** ein
- benötigte Ressourcen:
  - 1 – 2 kByte ROM, inklusive Fehlerspeicher
  - 0% Buslast (keine NM-spezifischen Nachrichten)
- Mercedes-Benz setzt **direktes NM** ein
- benötigte Ressourcen:
  - 1,5 – 1,7 kByte ROM
  - 4 – 8 kByte RAM
  - 1% Buslast

# ODEK/VDX NM – Zusammenfassung

---

- Ersparnis von Kosten und Entwicklungszeit
- aktuelle Netzwerkkonfiguration bestimmen
- indirektes NM
  - passiv
  - für einfache Anwendungen
- direktes NM
  - dynamisch
  - passt sich an Änderungen im Netz an
  - Betriebsmodi ändern (z.B. globaler Ruhezustand)
- kontrolliertes Verhalten der ECUs (bzgl. Betriebsmodi)
- von Kfz-Herstellern erfolgreich eingesetzt

# OSEK/VDX NM (Network Management)

---

Vielen Dank für die Aufmerksamkeit!

# Quellen

---

- OSEK/VDX Portal.  
URL: <http://www.osek-vdx.org/> [Stand: 15.09.2007].
- OSEK/VDX NM Spezifikation Version 2.5.3.  
URL: <http://portal.osek-vdx.org/files/pdf/specs/nm253.pdf>  
[Stand: 15.09.2007].
- VDI/VDE-Gesellschaft Mess- und Automatisierungstechnik (Hg.):  
OSEK-VDX. open systems in automotive networks.  
Düsseldorf: VDI-Verlag, 2000.