

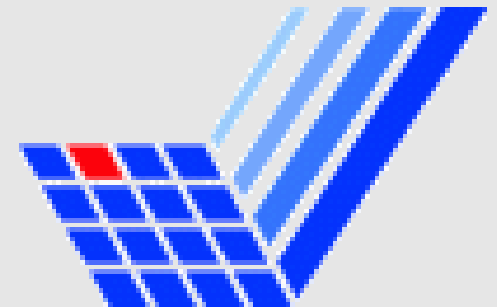
# OSEK / COM

---

**Florian Hohnsbehn**

[florian.hohnsbehn@cs.uni-dortmund.de](mailto:florian.hohnsbehn@cs.uni-dortmund.de)

PG AutoLab  
Seminarwochenende 21.-23. Oktober 2007



# Überblick

---

- **Einführung**
  - Was ist OSEK COM?
  - Einordnung in das Schichtenmodell von OSEK
- Nachrichten
- Anforderungen an untere Schichten
- Conformance Classes
- Fragen

# Einführung – Was ist OSEK / COM?

---

- Ziele:
  - Einheitliche Kommunikationsumgebung für automotive Steuergeräte
  - Protokollunabhängige Schnittstellen
- Anforderungen:
  - Kommunikation zwischen *Tasks* und/oder *Interrupt-Serviceroutinen*
  - Portierbarkeit, Wiederverwendbarkeit, Interoperabilität
  - Skalierbarkeit
  - Unterstützung von OSEK NM

# Einführung – Einordnung in OSEK

- *Interaction Layer (IL)*
  - stellt die OSEK COM API zur Verfügung
- *Network Layer*
  - Nicht durch OSEK COM spezifiziert, nur minimale Anforderungen
- *Data Link Layer*
  - Bietet oberen Schichten Dienste zum Datentransfer
  - Nicht von OSEK COM spezifiziert, minimale Anforderungen

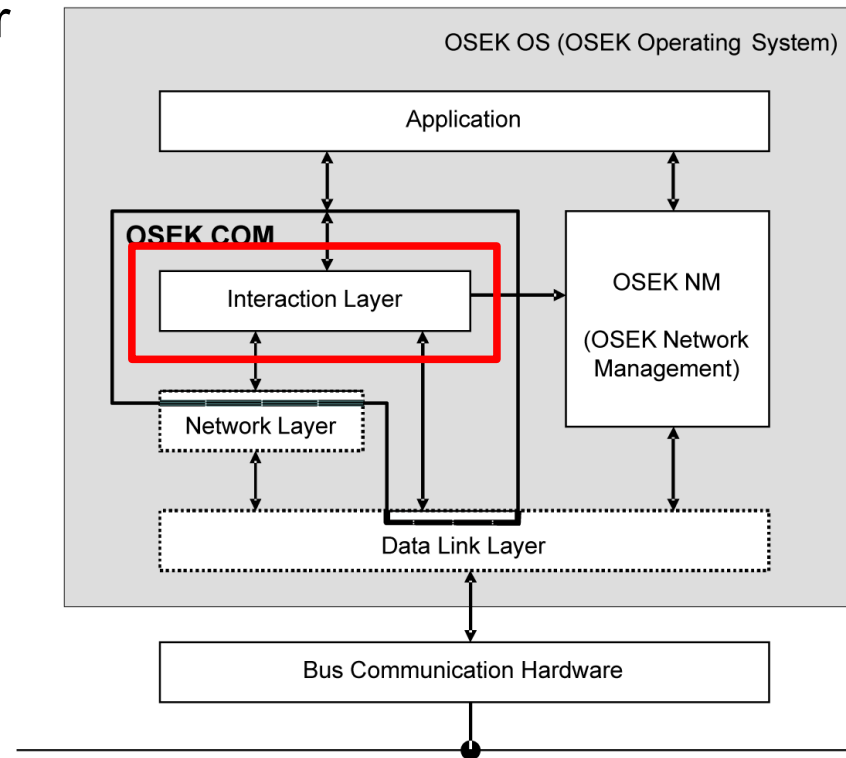


Figure 1-1: OSEK COM's layer model

# Nachrichten

---

- *Einführung*
- **Nachrichten**
  - Kommunikationskonzept
  - Übermittlung
  - Empfang
  - Filter
  - Deadline Monitoring
- Anforderungen an untere Schichten
- Conformance Classes
- Fragen

# Nachrichten - Kommunikationskonzept

---

- Nachrichten-(Signal-)basierte Kommunikation
  - Nachrichten und deren Eigenschaften statisch in OIL definiert
  - Nachrichtenlänge: statisch, dynamisch oder 0 (zero-length messages)
- Nachrichtenobjekte
  - Zur Verwaltung der Nachrichten
  - Identifier bei Systemerstellung zugewiesen
  - *Sending Message Object*
  - *Receiving Message Object*
    - Queued und Unqueued

# Nachrichten - Kommunikationskonzept

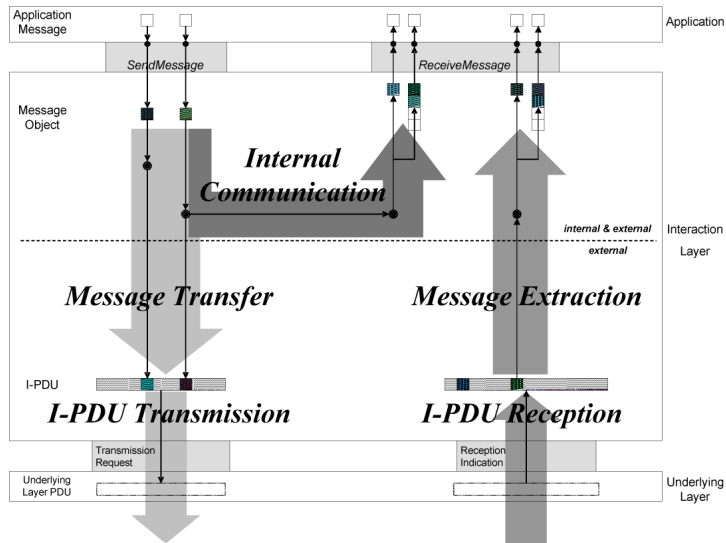


Figure 2-1: Simplified model for message transmission and reception in OSEK COM

- **Interne Kommunikation:**
  - Wird komplett in der *Interaction Layer* abgehandelt
    - Nachricht wird dem Empfänger unmittelbar zur Verfügung gestellt
- **Externe Kommunikation:**
  - IL benutzt Dienste der unteren Schicht
    - Eine oder mehrere Nachrichten in *IL Protocol Data Units (I-PDU)*
    - I-PDU wird an die untere Schicht kommuniziert

# Nachrichten - Kommunikationskonzept

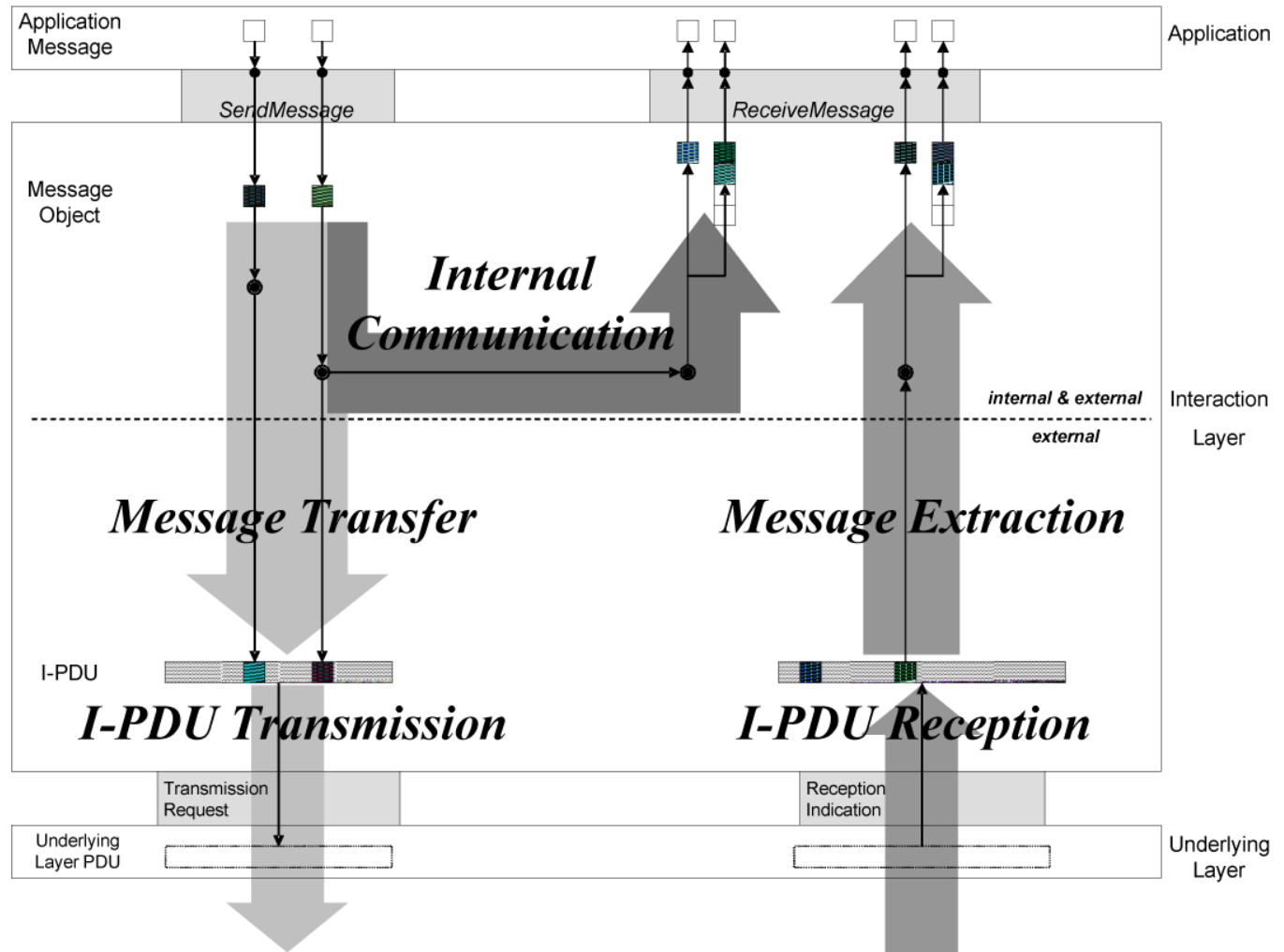


Figure 2-1: Simplified model for message transmission and reception in OSEK COM



# Nachrichten - Übermittlung

---

- API-Dienste: *SendMessage*, *SendDynamicMessage*, *SendZeroMessage*
- Ablauf (nur externe Kommunikation):
  - Filterung (nur statische Länge):
  - Packen von Nachricht in I-PDU
  - Übertragung der I-PDUs über untere Schicht
  - Optional: Bestätigung, ob Übertragung erfolgreich
- Interne Kommunikation:
  - Nachricht wird direkt ans *ReceiveMessageObject* der IL geroutet
  - Keine Nachrichten dynamischer Länge

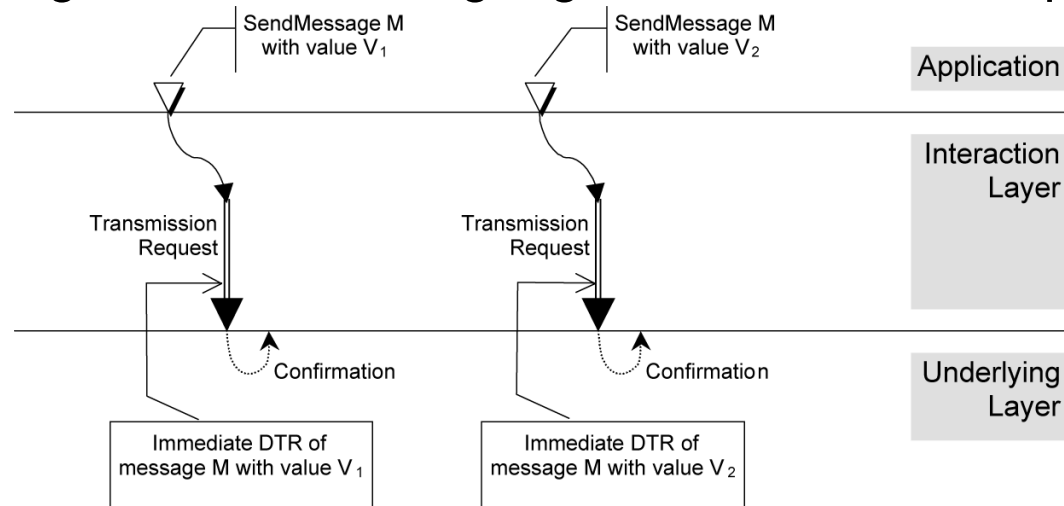
# Nachrichten - Übermittlung

---

- Transfer vom *SendMessageObject* zur I-PDU (nur extern)
  - Wenn Nachricht nicht ausgefiltert: wird in I-PDU geschrieben
  - Transfer properties bei Übertragung von Anwendung zu I-PDU:
    - Triggered:
      - Schreibt Nachricht in I-PDU
      - Veranlasst unmittelbare Übertragung der I-PDU, außer im *Periodic Transmission Mode*
      - *Zero-length messages* haben immer die Eigenschaft *Triggered*
      - Bei *Zero-length messages* werden keine Daten übertragen
    - Pending:
      - Schreibt Nachricht in I-PDU
      - Kein Transmission-Request

# Nachrichten – Übermittlung

- Übertragung der I-PDU
  - Drei Transmission Modes: Direct, Periodic, Mixed
- Direct Transmission Mode:
  - Ausgelöst unmittelbar nach I-PDU-Transfer mit Triggered-Property
  - Minimale Verzögerung (Minimum Delay Time,  $I\_TMD\_MDT \geq 0$ ) zwischen zwei Übertragungen definiert
  - Timer beginnt, sobald vorherige Übertragung bestätigt
  - Timer abgelaufen: Übertragung beim nächsten Request



DTR : Direct Transmission Request

# Nachrichten - Übermittlung

- Periodic Transmission Mode:
  - I-PDU wird mit jedem *SendMessage*-/*SendDynamicMessage*-Aufruf aktualisiert, ohne Transmission-Request
  - Ignoriert transfer property
  - Übertragung wird periodisch ausgelöst (Time Period, I\_TMP\_TPD)

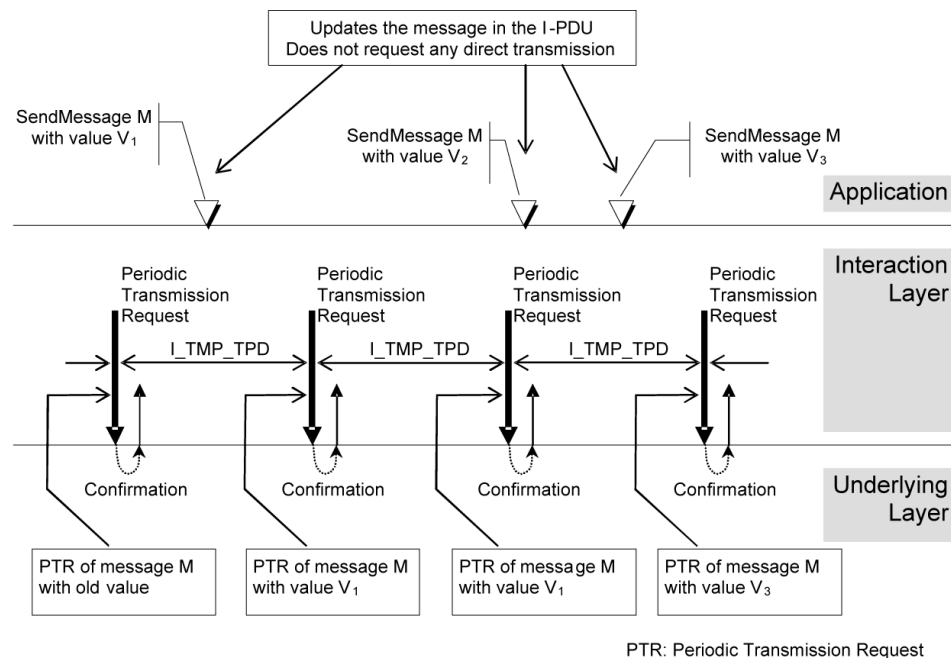


Figure 2-5: Periodic Transmission Mode

# Nachrichten - Übermittlung

- Mixed Transmission Mode:
  - Kombination aus direkter und periodischer Übertragung
  - Periodische Übertragung mit Time Period  $I\_TMM\_TPD$
  - Zwischen periodischen Übertragungen: direkte Übertragungen
  - Minimum Delay Time,  $I\_TMM\_MDT \geq 0$
  - MDT kann periodische Übertragung verzögern
  - Periode selbst wird nicht beeinträchtigt

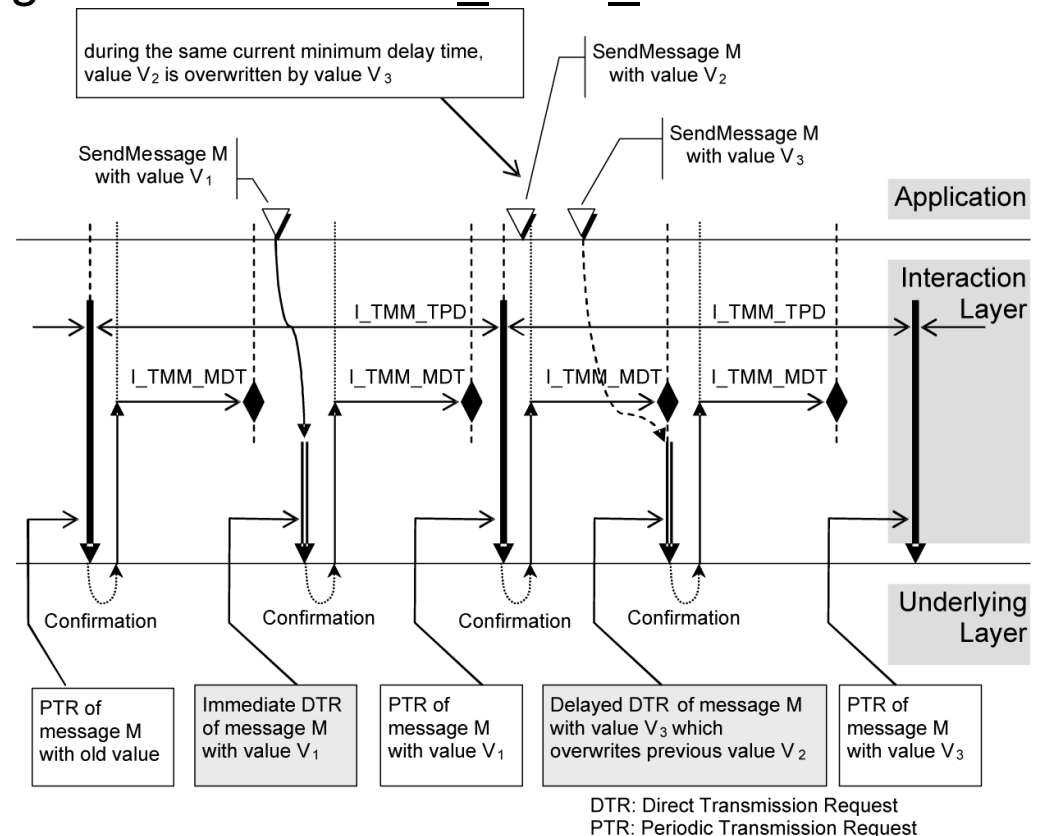


Figure 2-6: Mixed Transmission Mode with minimum delay time (simple cases)

# Nachrichten - Übermittlung

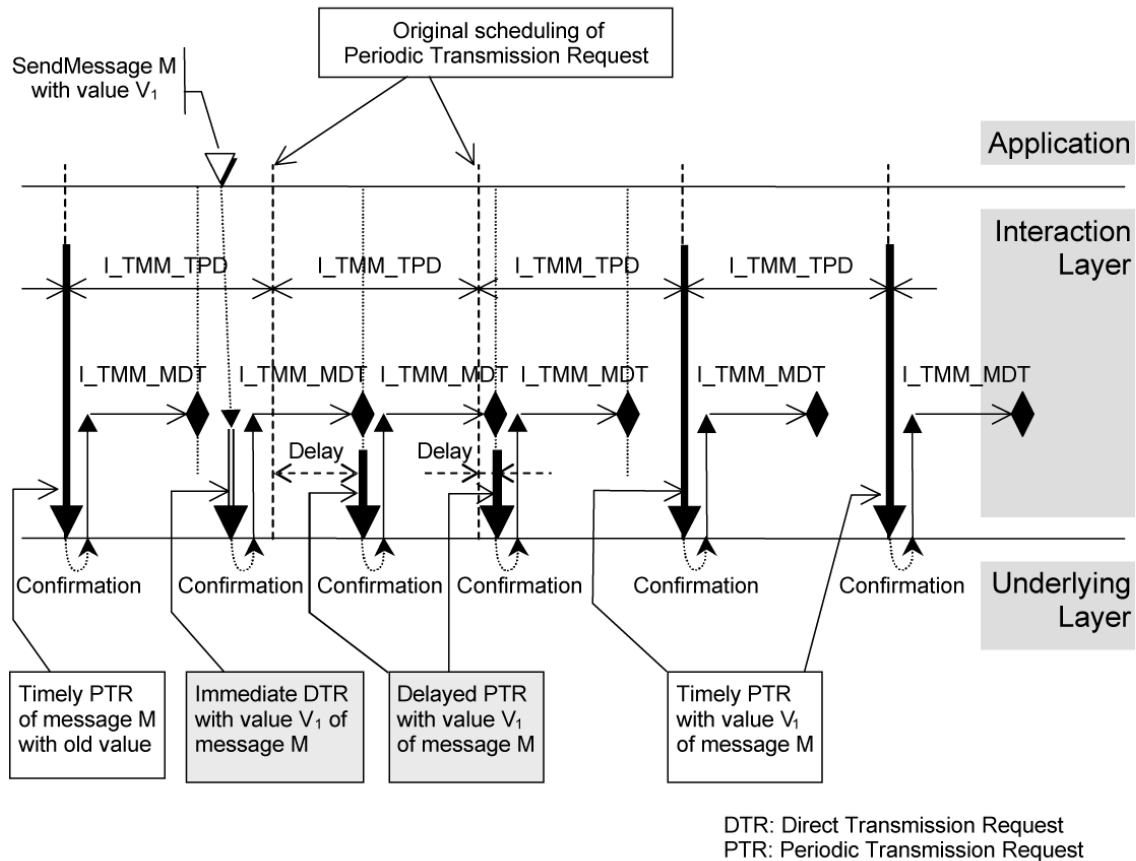


Figure 2-7: Mixed Transmission Mode with minimum delay time (MDT delays PTR)

# Nachrichten - Empfang

---

- Ablauf (nur externe Kommunikation):
  - *Indication*
  - Falls kein Fehler: Empfang erfolgreich, Übertragung der PDU in die I-PDU
  - Falls Fehler: *error indication*
  - *Reception Deadline Monitoring*
  - Nachrichtendaten aus I-PDUs entpackt
  
- Ablauf (intern und extern):
  - Reception filtering; kopieren in *ReceiveMessageObject*
  - *MessageReception* notification
  - Empfangen der Nachricht

# Nachricht - Filter

- Beim Senden und Empfangen: MessageObjects filtern
- Kein Filtern von dynamic-length und zero-length

- Attribute:

- *New value*: aktueller Wert
- *Old value*: vorheriger Wert
- *mask, x, min, max, period, offset*: konstante Werte
- *occurrence*: Zähler der Vorkommen der Nachricht

Algorithm Reference	Algorithm	Description
F_Always	True	No filtering is performed so that the message always passes
F_Never	False	The filter removes all messages
F_MaskedNewEqualsX	$(new\_value \& mask) == x$	Pass messages whose masked value is equal to a specific value
F_MaskedNewDiffersX	$(new\_value \& mask) != x$	Pass messages whose masked value is not equal to a specific value
F_NewIsEqual	$new\_value == old\_value$	Pass messages which have not changed
F_NewIsDifferent	$new\_value != old\_value$	Pass messages which have changed
F_MaskedNewEqualsMaskedOld	$(new\_value \& mask) == (old\_value \& mask)$	Pass messages where the masked value has not changed
F_MaskedNewDiffersMaskedOld	$(new\_value \& mask) != (old\_value \& mask)$	Pass messages where the masked value has changed
F_NewIsWithin	$min \leq new\_value \leq max$	Pass a message if its value is within a predefined boundary
F_NewIsOutside	$(min > new\_value) \text{ OR } (new\_value > max)$	Pass a message if its value is outside a predefined boundary
F_NewIsGreater	$new\_value > old\_value$	Pass a message if its value has increased
F_NewIsLessOrEqual	$new\_value \leq old\_value$	Pass a message if its value has not increased
F_NewIsLess	$new\_value < old\_value$	Pass a message if its value has decreased
F_NewIsGreaterOrEqual	$new\_value \geq old\_value$	Pass a message if its value has not decreased
F_OneEveryN	$occurrence \% period == offset$	Pass a message once every N message occurrences. Start: occurrence = 0. Each time the message is received or transmitted, occurrence is incremented by 1 after filtering. Length of occurrence is 8 bit (minimum).

Table 2-1: Message filter algorithms



# Nachrichten – Deadline Monitoring

## ■ Reception Deadline Monitoring

- Je Nachricht konfiguriert, je I-PDU mit Nachricht ausgeführt
- Überwacht, ob periodische Nachricht innerhalb Intervall ( $I\_DM\_RX\_TO$ ) empfangen
- Timer abgebrochen und neu gestartet, sobald I-PDU aus dem PDU mit Nachricht empfangen

- Time-Out: sofort Timerneustart
- Extra-Intervall für erstes Zeitintervall ( $I\_DM\_FRX\_TO$ )
- Auch auf Direct und Mixed Transmission Mode anwendbar
- Benachrichtigung an Anwendung oder OSEK Indirect NM möglich

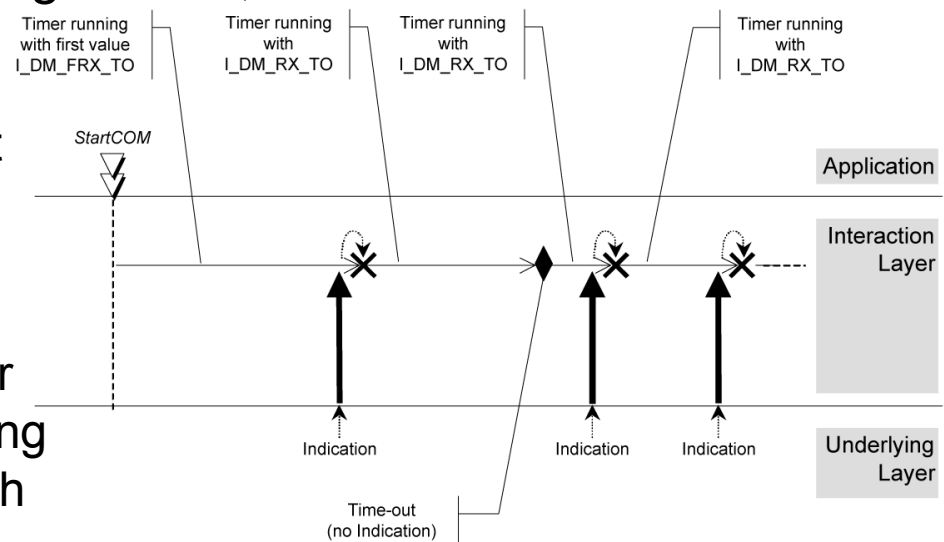


Figure 2-11: Deadline Monitoring for periodic reception

# Nachrichten – Deadline Monitoring

---

- Transmission Deadline Monitoring
  - Überwacht, ob transmission requests Übertragung zur Folge haben
  - Kann für jede Nachricht ein- oder ausgeschaltet werden
  - Überwachung erfolgt je I-PDU
  - Time-Out-Period ist Eigenschaft der I-PDU
  - Für alle Nachrichten mit Triggered Transfer Property möglich
  - Für Nachrichten mit Pending Transfer Property nur im Periodic Mode und im periodischen Teil des Periodic Mode möglich
- Deadline Monitoring im Direct Transmission Mode
  - Überprüft bei Aufruf von *SendMessage*, *SendDynamicMessage*, *SendZeroMessage*, ob im Zeitintervall (I\_DM\_TMD\_TO) Bestätigung erfolgt
  - Timer startet, sobald *SendMessage* o.ä. abgearbeitet sind
  - Bei Time-Out: Anwendung wird benachrichtigt
  - Anwendung entscheidet, ob Übertragung erneut versucht wird

# Nachrichten – Deadline Monitoring

- Deadline Monitoring im Periodic Transmission Mode
  - Überwacht, ob I-PDU in gegebenem Zeitintervall übertragen
  - Zeitintervall (I\_DM\_TMP\_TO) kann größer sein als die Periode
  - Timer startet nach jedem periodischen Transmission-Request, wenn er nicht schon läuft
  - Timer wird durch jede erhaltene Bestätigung abgebrochen
  - Bei Time-Out: Anwendung wird benachrichtigt

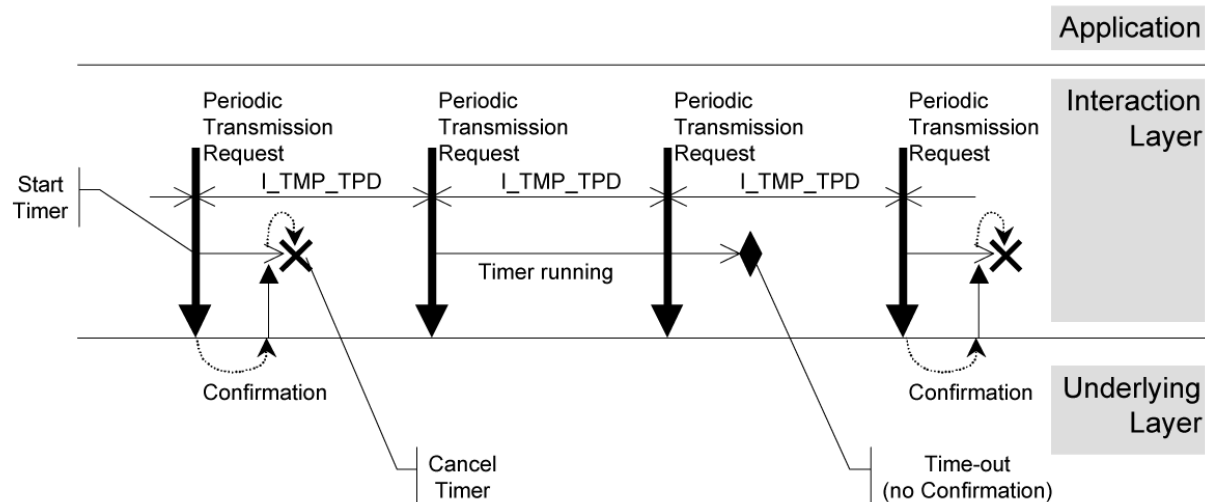


Figure 2-15: Periodic Transmission Mode: failed transmissions

# Nachrichten – Deadline Monitoring

- Deadline Monitoring im Mixed Transmission Mode
  - Überwacht, ob I-PDU in gegebenem Zeitintervall übertragen
  - Timer startet nach jedem Transmission-Request, wenn er nicht schon läuft
  - Timer wird durch jede erhaltene Bestätigung abgebrochen
  - Bei Time-Out: Anwendung wird benachrichtigt
  - Anwendung oder OSEK Indirect NM können benachrichtigt werden

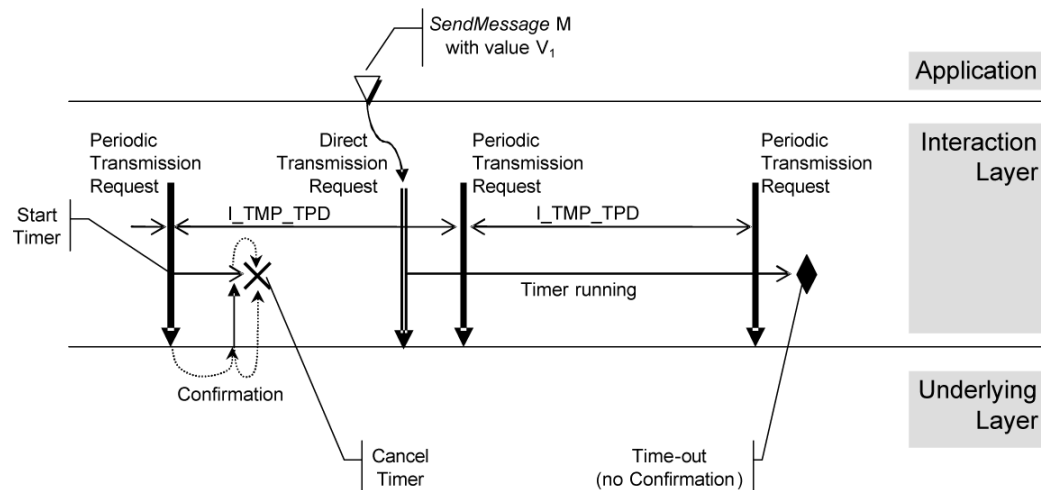


Figure 2-17: Mixed Transmission Mode: failed transmissions

# Nachrichten - Überblick

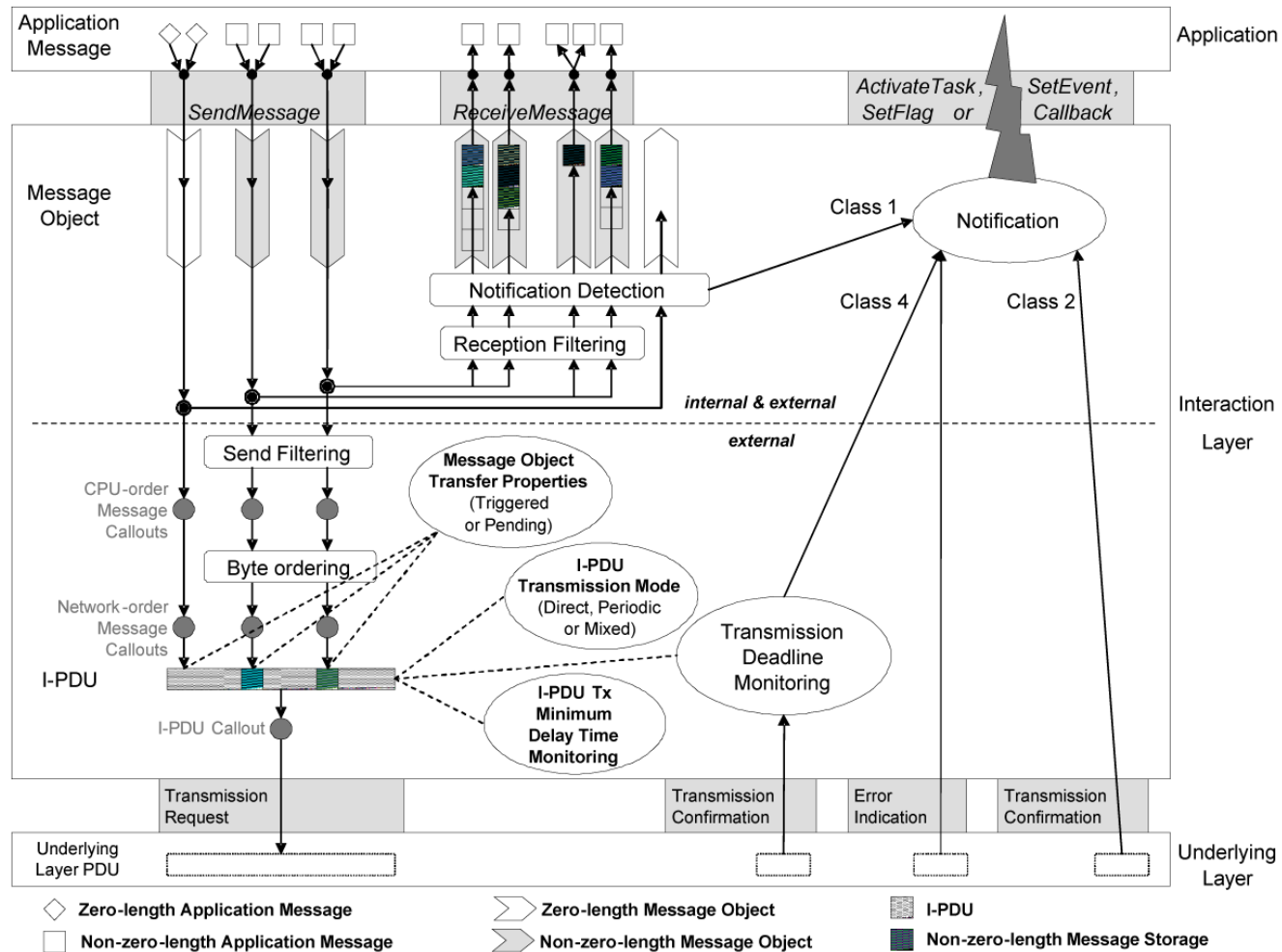


Figure 2-20: IL model for internal communication and external transmission

# Anforderungen an untere Schichten

---

- *Einführung*
- *Nachrichten*
- *Routinen der Interaction Layer*
- **Anforderungen an untere Schichten**
- Conformance Classes
- Fragen

# Anforderungen an untere Schichten

---

- I-PDUs mit fester oder dynamischer Länge verschicken:
  - drei Dienste erforderlich:
    - Request Service:
      - IL: Steuerinformationen und I-PDU an untere Schicht
      - schnellstmögliche Übermittlung der I-PDU
    - Confirmation Service:
      - Erfolg oder Fehler melden
    - Indication Service:
      - von unterer Schicht an Interaction Layer
      - Erfolg oder Fehler melden
  
- Broadcastübermittlung

# Conformance Classes

---

- *Einführung*
- *Nachrichten*
- *Anforderungen an untere Schichten*
- **Conformance Classes**
- Fragen



# Conformance Classes

## ■ Verschiedene Funktionalitätsstufen

### ■ CCCA:

- minimale Funktionalität, nur interne Kommunikation
- Nur unqueued Messages

### ■ CCCB:

- Nur interne Kommunikation
- Queued Messages
- Statusinformationen von Nachrichten

### ■ CCC0:

- Minimale Funktionalität für interne und externe Kommunikation

### ■ CCC1:

- Unterstützt alle Features von OSEK COM

Features	CCCA	CCCB	CCC0	CCC1
Unqueued messages	√	√	√	√
Notification Class 1	√ <sup>5</sup>	√	√	√
Queued messages		√		√
Message status information		√		√
External communication			√	√
Triggered Transfer Property			√	√
Notification Class 2			√	√
Byte order conversion			√	√
Direct Transmission Mode			√	√
Filtering				√
Pending Transfer Property				√
Zero-length messages				√
Dynamic-length messages				√
Periodic Transmission Mode				√
Mixed Transmission Mode				√
Minimum delay time				√
Deadline Monitoring				√
Notification Class 3				√
Notification Class 4				√
Callouts				√

Table 4-1: Definition of conformance classes

# Fragen

---

- *Einführung*
- *Nachrichten*
- *Anforderungen an untere Schichten*
- *Conformance Classes*
- **Fragen**

# Quellen

---

- OSEK-COM-Spezifikation:
  - <http://portal.osek-vdx.org/files/pdf/specs/osekcom303.pdf>