

Projekt Erdbebenfrühwarnung im WiSe 2010/11



Entwicklung verteilter eingebetteter Systeme

Prof. Dr. Joachim Fischer
Dipl.-Inf. Ingmar Eveslage
Dipl.-Inf. Frank Kühnlenz

fischer|eveslage|kuehnlenz@informatik.hu-berlin.de

6. *SDL-Konzepte (Präzisierung)*

1. Modellstruktur
2. Einfacher Zustandsautomat: Triggerarten
3. Nachrichtenadressierung
4. Dynamische Prozessgenerierung
5. Prozeduren
6. Lokale Objekte
7. Ersetzungsmodelle
8. Semaphore
9. Spezialisierung von Zustandsautomaten

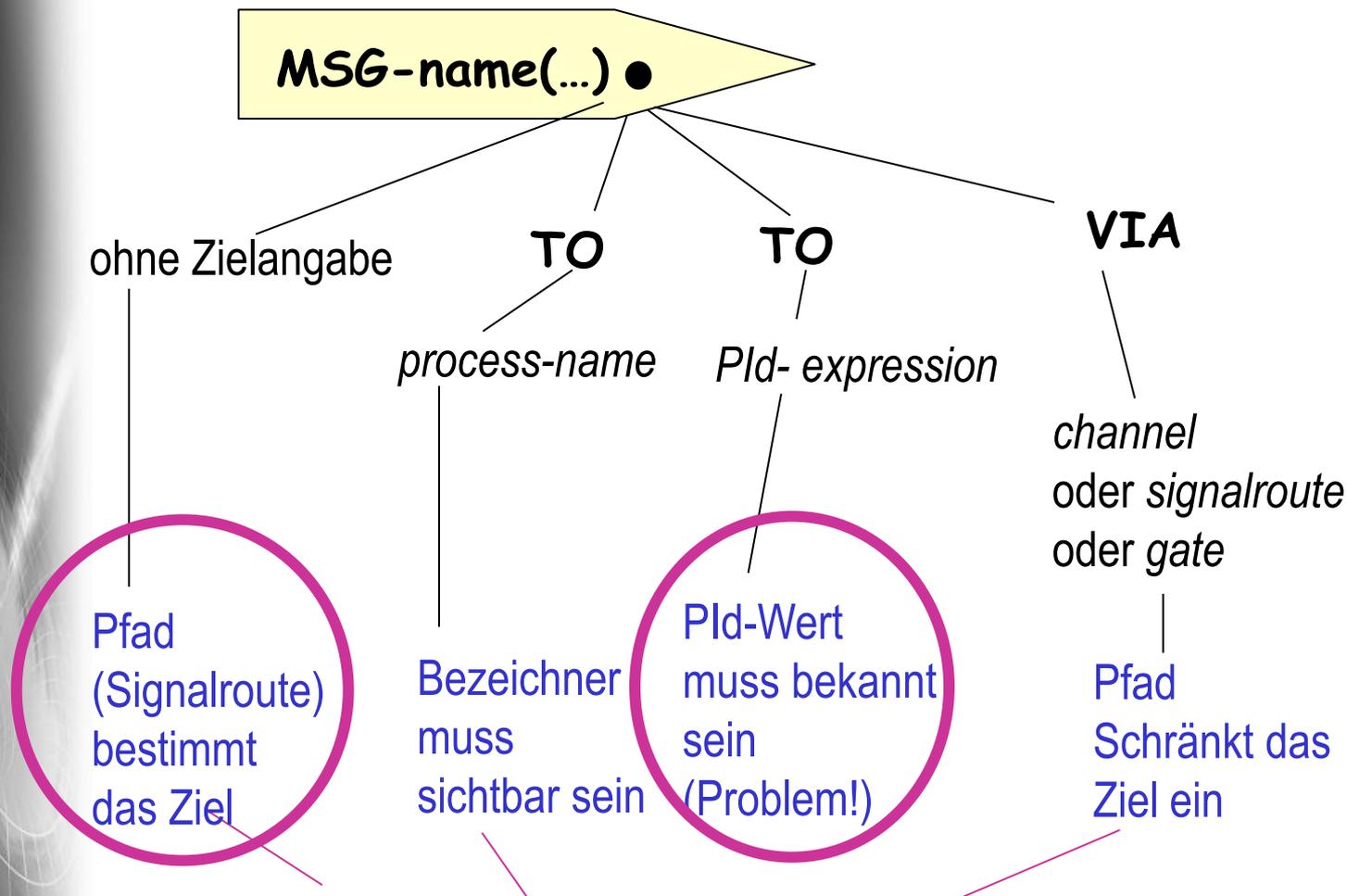
Zwei Adressierungsarten

(1) **OUTPUT** *msg-name (param-List)* **TO_X** *receiver*

(2) **OUTPUT** *msg-name (param-List)* **VIA** *medium*

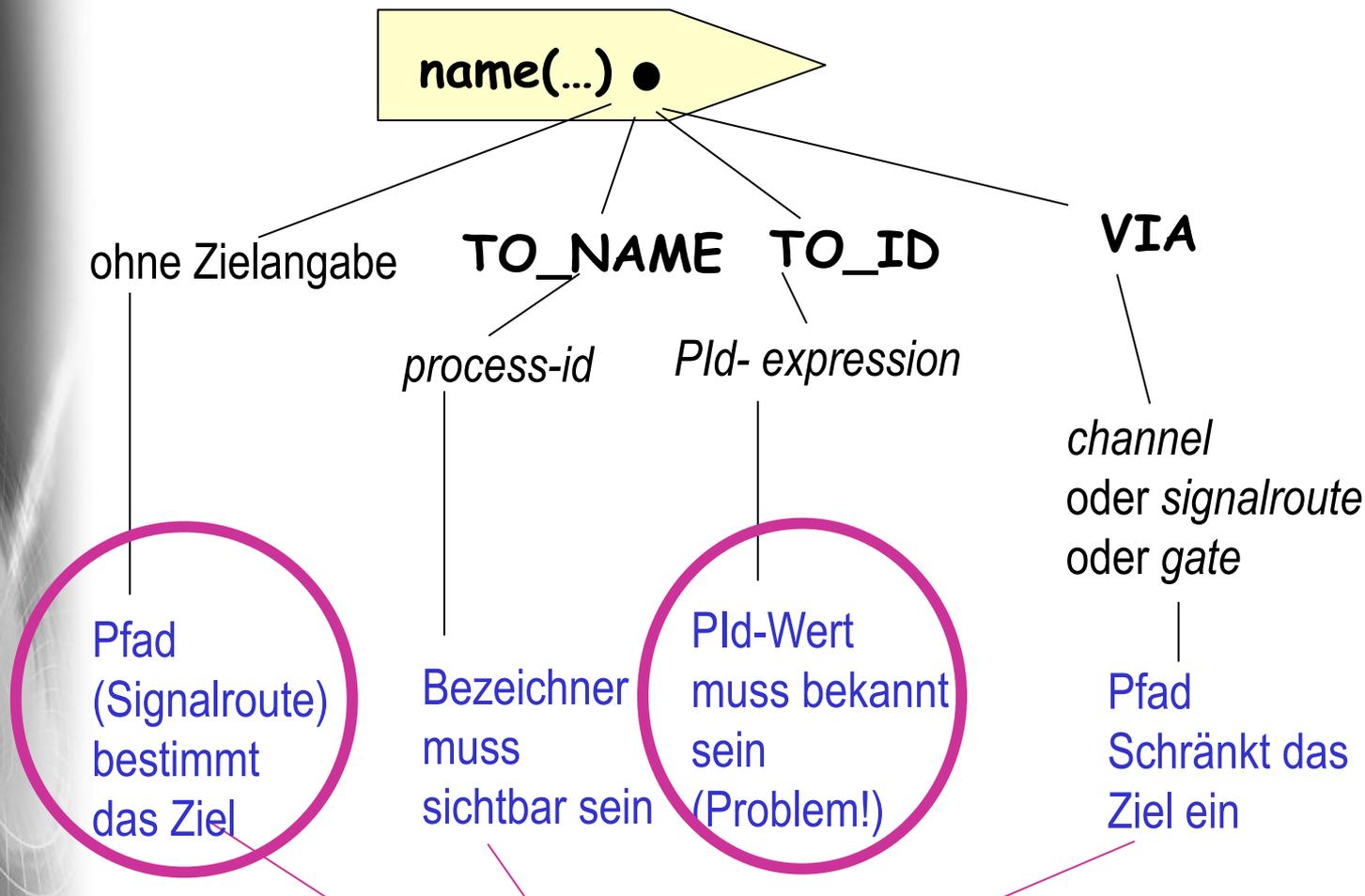
- Parameter: Werte, Referenzen
bei Referenzen ist wichtig: Welcher Prozess legt Lebenslauf fest.
häufig: Sender besitzt Datum nicht mehr, nachdem gesendet,
Empfänger gibt Speicherplatz frei (daraus entstehen Probleme)
- **TO_X:**
 - **ID:** PARENT, SELF, OFFSPRING, SENDER, oder RTDS_QueueId
 - **NAME:** erster generierter Prozess einer Menge
(nicht benutzen, wenn Name nicht 1-deutig)
 - **ENV**
- **medium:**
 - Name von Channel oder gate (verbunden mit aktuellem Prozess)
keine Mehrdeutigkeit zulässig!

SDL-Standard: Nachrichten-Adressierungsarten



bei Mehrdeutigkeiten der Empfänger-Instanz erfolgt eine nichtdeterminierte Auswahl

SDL-RT: Nachrichten-Adressierungsarten



bei Mehrdeutigkeiten der Empfänger-Instanz erfolgt eine nichtdeterminierte Auswahl

SDL-RT: Zeiger als Parameter

```
MESSAGE  
  ConReq(unsigned char *),  
  ConConf,  
  DisReq(myStruct *);
```

```
long          myDataLength;  
unsigned char *myData;  
myStruct      *pData;
```

```
ConReq  
(myDataLen,  
myData)
```

```
ConConf
```

```
DisReq  
(pData)
```

unsigned char* man kann optional auf die Länge zugreifen
(ist als erster Parameter zu übergeben)

SDL/RT- Adressierungsarten

```
MESSAGE
  ConReq(unsigned char *),
  ConConf,
  DisReq(myStruct *);
```

```
long          myDataLength;
unsigned char *myData;
myStruct      *pData;
```

```
ConReq
(256, myData)
TO_ID PARENT
```

```
ConConf TO_ID
aCalculatedReceiver
```

```
DisReq
(pData) TO_ID
PARENT
```

Adressierungsart **TO_NAME**: wird rudimentär unterstützt werden

Spezifische Adressierungsklauseln:

- TO_ID (PID-Ausdruck)
- TO_NAME
- TO_ENV

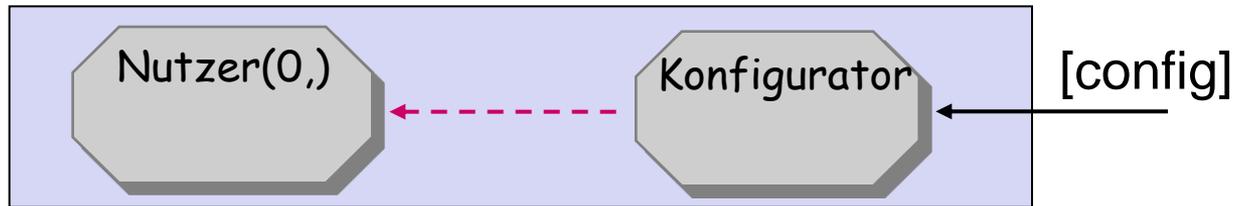
6. *SDL-Konzepte (Präzisierung)*

1. Modellstruktur
2. Einfacher Zustandsautomat: Triggerarten
3. Nachrichtenadressierung
4. Dynamische Prozessgenerierung
5. Prozeduren
6. Lokale Objekte
7. Ersetzungsmodelle
8. Semaphore
9. Spezialisierung von Zustandsautomaten

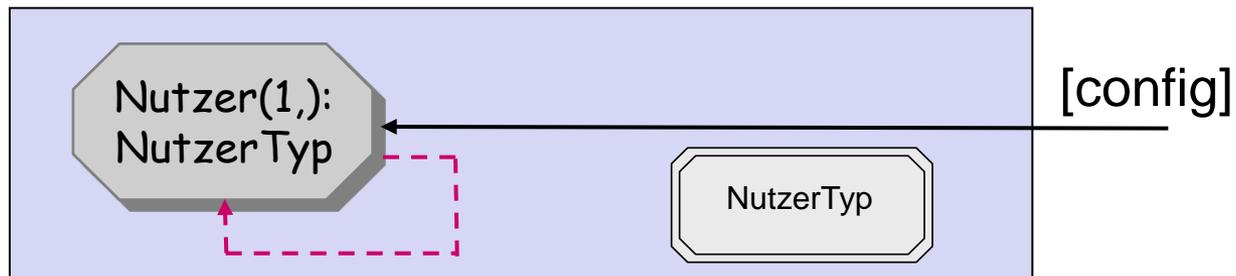
A-2: Dynamische Konfiguration von Nutzern

- 1. Variante: expliziter Generator

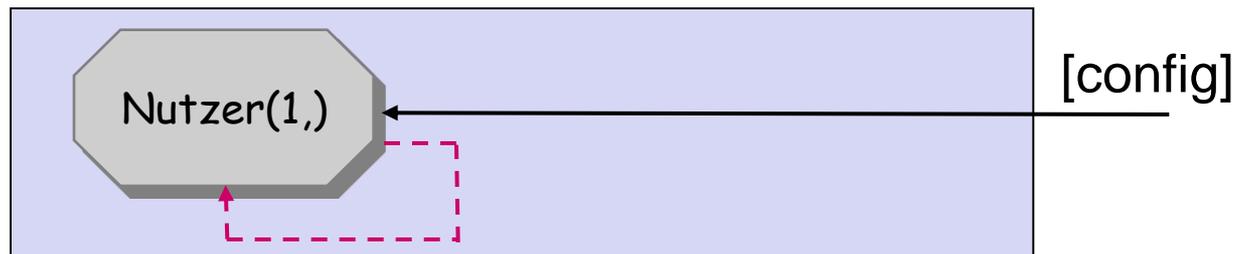
signal config (Natural):



- 2. Variante: Nutzertyp-Instanz ist in der Lage, Kopien von sich selbst in der jeweiligen Instanzmenge (hier Nutzer) zu erzeugen



- 3. Variante: Nutzer ist in der Lage, Kopien von sich selbst zu erzeugen



6. SDL-Konzepte (Präzisierung)

1. Modellstruktur
2. Einfacher Zustandsautomat: Triggerarten
3. Nachrichtenadressierung
4. Dynamische Prozessgenerierung
5. Prozeduren
6. Lokale Objekte
7. Ersetzungsmodelle
8. Semaphore
9. Spezialisierung von Zustandsautomaten

Kap. 7 INRES-Protokoll

später

7. *Protokollentwicklung in SDL*

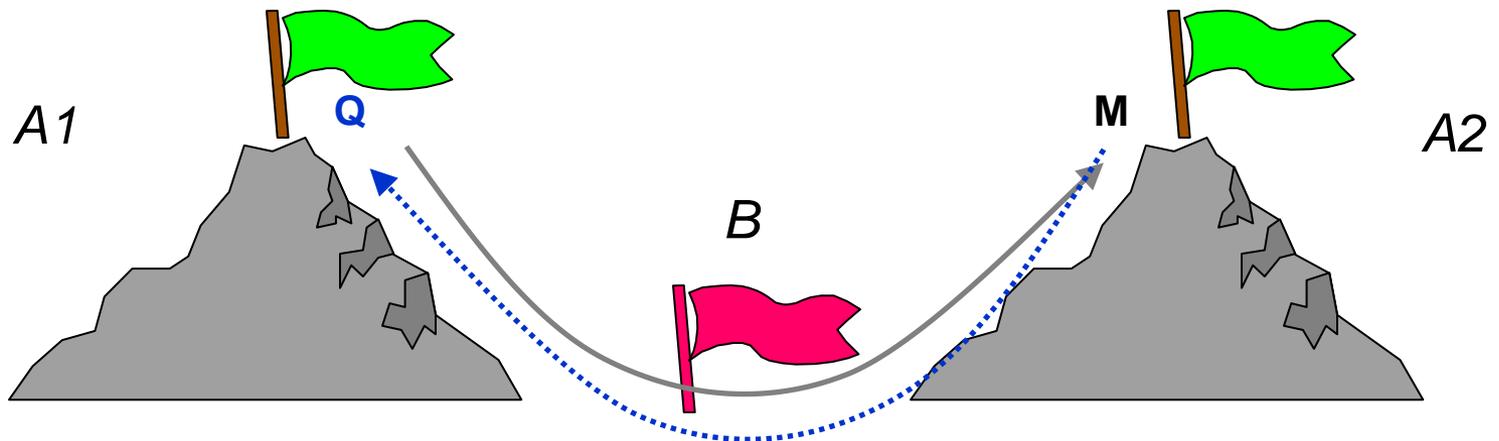
- OSI-Schichtenmodell (Konzept)
- InRes-Protokoll (Pseudo-Rechnernetzprotokoll)
- Umsetzung in SDL/RT

Protokolle im Altertum

Übertragung von Nachrichten über unsichere „Kanäle“
bei Quittieren dieser Nachrichten

Ziel

Absprache eines gemeinsamen Angriffstermins der verteilten grünen Armee



Fall: A1 erhält tatsächlich eine Bestätigung Q für Nachricht M von A2
Wie sicher können sich A1 und A2 sein, dass Termin gilt ?

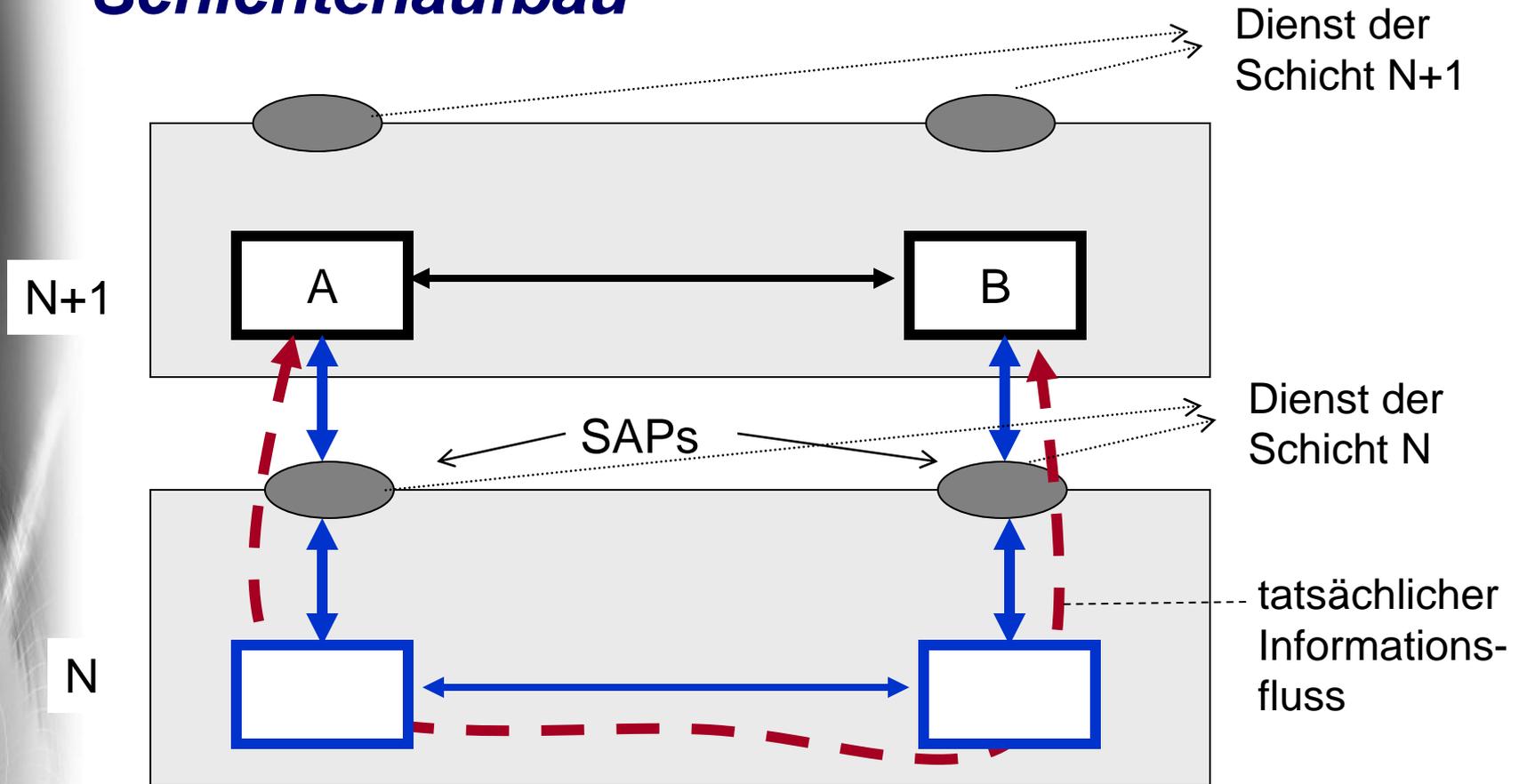
Protokoll (Protocol)

- Regeln zur Kommunikation zwischen Partnerinstanzen innerhalb einer Kommunikationsebene
 - Wer muss wann aus welchem Grund eine Information an seinen Partner schicken
 - Vereinbarungen von Struktur und Codierungen der auszutauschenden Informationen
- Schichten-Struktur zur Beherrschung komplexer Übertragungsdienste für verbundene Rechner (als Knoten in einem Rechnernetz)
 - in realen Systemen gibt es nur eine einzige physisch vorhandene horizontale Verbindung, die anderen sind abstrakt / virtuell
- ISO-Standard zum Aufbau von Kommunikationsprotokollen: **Open System Interconnection**
 - Konzepte unabhängig von ihrer Implementierung)

Dienst (Service)

- Eine Partnerinstanz einer Schicht N (*layer N*) bietet der darüber liegenden Schicht (*layer N+1*) eine Dienstleistung (*Service*)
- Kommandos zur Dienstinanspruchnahme heißen Dienstelemente/Dienstprimitive (*Service Primitives*, SP)
- Schnittstelle einer Schicht bzgl. eines Dienstes heißt Dienstzugangspunkt (*Service Access Point*, SAP)
- es besteht eine vertikale Instanzbeziehung zwischen den Dienst-Beteiligten unterschiedlicher Schichten (Dienstverbinder-Dienstnutzer)

Schichtenaufbau

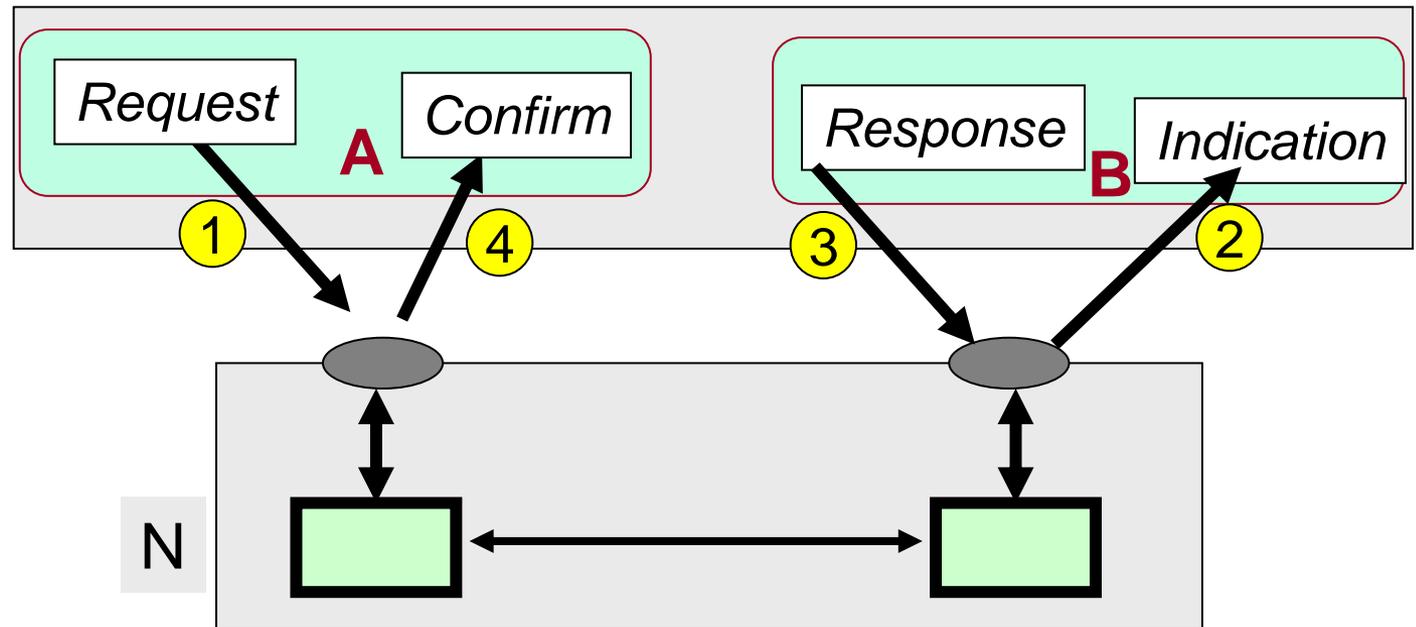


- A und B sind Kommunikationspartner
- Sie haben komplexe Informationen auszutauschen **ohne** physische Verbindung – sie vereinbaren ein Protokoll
- Sie sind Nutzer von Diensten der Schicht N, die physische Übertragung elementarer Informationen unterstützt

Identifikation von Dienstprimitiv-Klassen

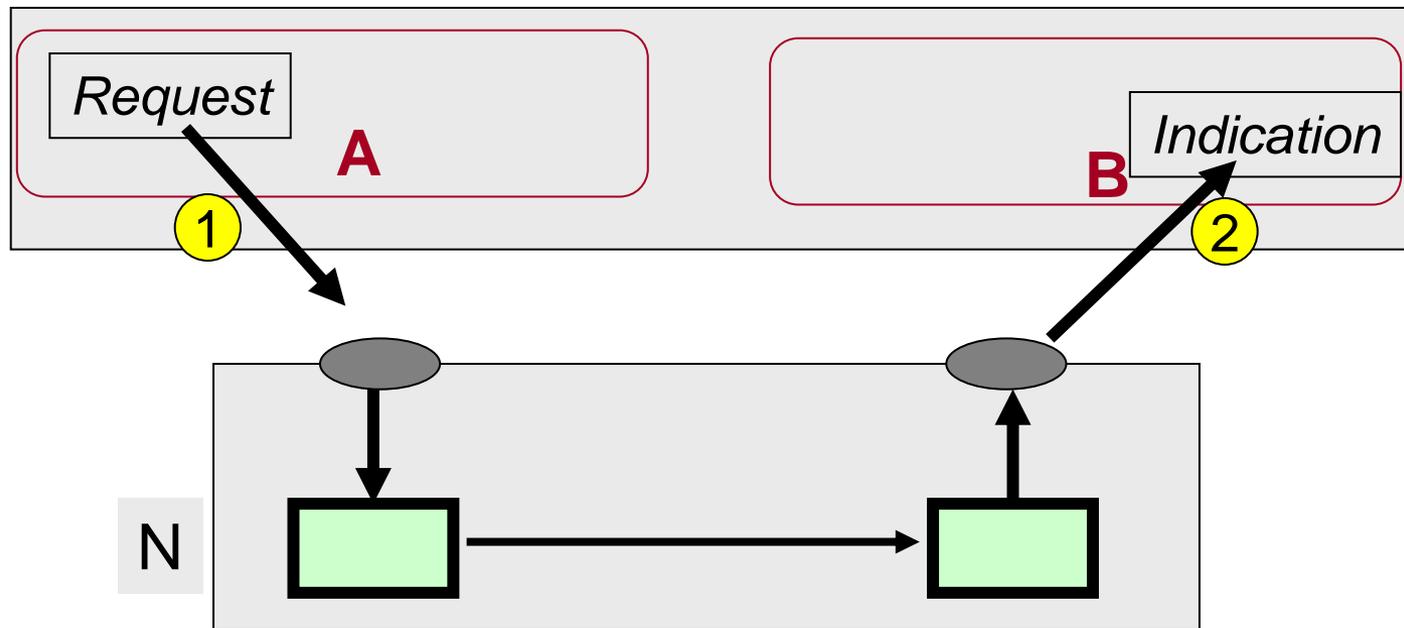
vier Grundtypen von Aktionen bei **einer** gerichteten Kommunikation zwischen **zwei** Partnern

Beispiel: A ist Auslöser, sein Anliegen wird von B bestätigt



Identifikation von Dienstprimitiv-Klassen

Beispiel: B ist Auslöser, sein Anliegen soll von A nicht bestätigt werden



7. *Protokollentwicklung in SDL*

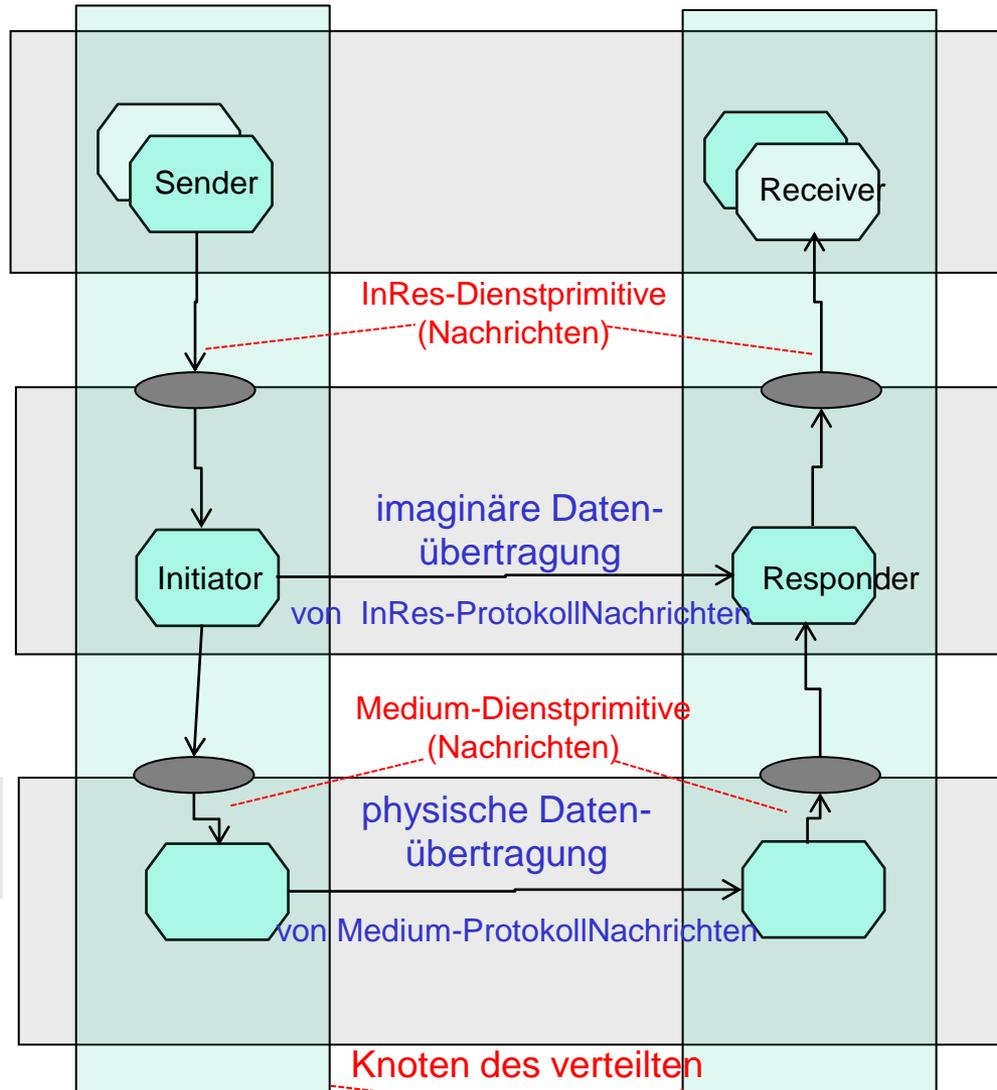
- OSI-Schichtenmodell (Konzept)
- InRes-Protokoll (Pseudo-Rechnernetzprotokoll)
- Umsetzung in SDL/RT

InRes-Schichtenarchitektur

Protokolleinheiten
(Prozesse/Zustands-
Maschinen)

ServiceAccessPoints
(Gates, Ports)

Anwendungs-
schicht



InRes-Dienste

überwachte
(sichere)
Datenübertragung

Medium-Dienste

unsichere
Datenübertragung

InRes-
Schicht

InRes-Protokoll:
unsymmetrisch,
verbindungsorientiert

Medium-
Schicht

Medium-Protokoll:
symmetrisch,
verbindungslos

InRes-Kommunikation

InRes = Initiator-Responder

(als asymmetrische 2-Schichten Kommunikationsvariante)

InRes-Dienst (der **InRes-Schicht**)

- sichere Datenübertragung über ein unsicheres Medium
- wird realisiert über das verbindungsorientierte INRES-Protokoll unter Nutzung eines Dienstes zur Datenübertragung (Medium-Dienst)
- Phasen:
Verbindungsaufbau, Datenübertragung, Verbindungsabbau

Medium-Dienst (der **Medium-Schicht**)

- wird realisiert über das verbindungslose Medium-Protokoll als physische Datenübertragung
- physische Übertragung mit Datenverlust

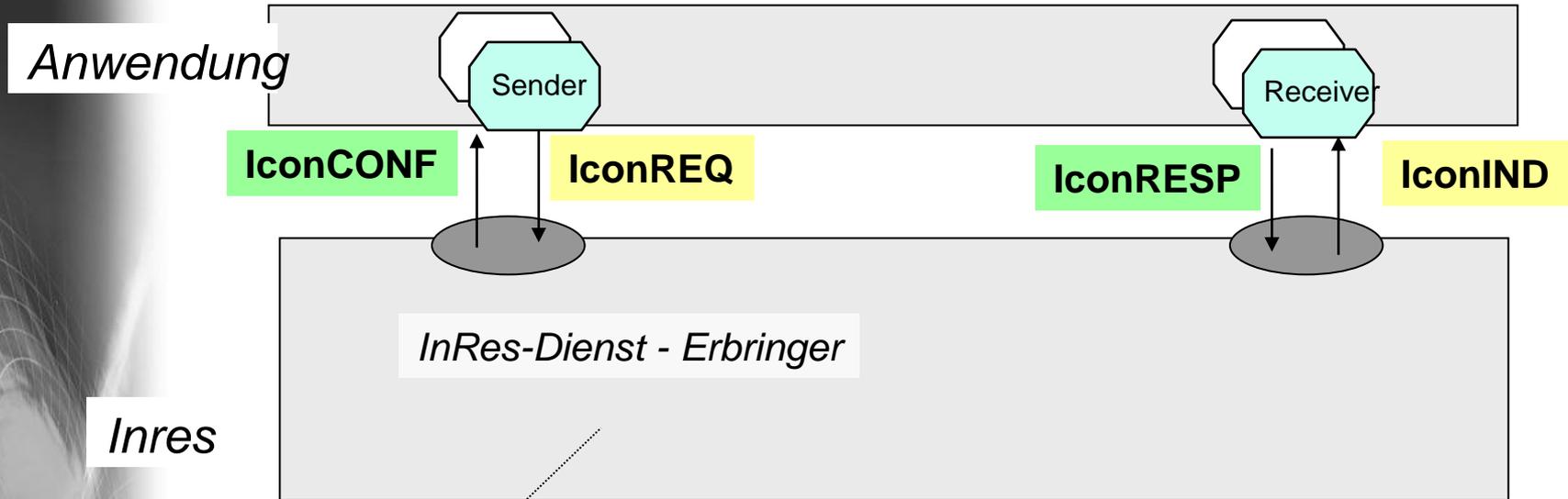
Dienstprimitive und Protokollnachrichten

Konzept	Kürzel	voller Name	Parameter
InRes-Dienstprimitive (Nachrichten)	IconREQ	InRes-Connection-Request	
	IconIND	InRes-Connection-Indication	
	IconRESP	InRes-Connection-Response	
	IconCONF	InRes-Connection-Confirm	
	IconERR	InRes-Connection-Error	ErrorType
	IdisREQ	InRes-Disconnection-Request	
	IdisIND	InRes-Disconnection-Indication	SequenceNumber
	IdatREQ	Inres-Data-Request	ISDUType
	IdatIND	InRes-Data-Indication	ISDUType

InRes-Service-Data-Unit

InRes-Dienst: verbindungsorientierte sichere Datenübertragung

Erfolgreicher **Verbindungsaufbau** bei Anwendung des InRes-Dienstes



Phasen:

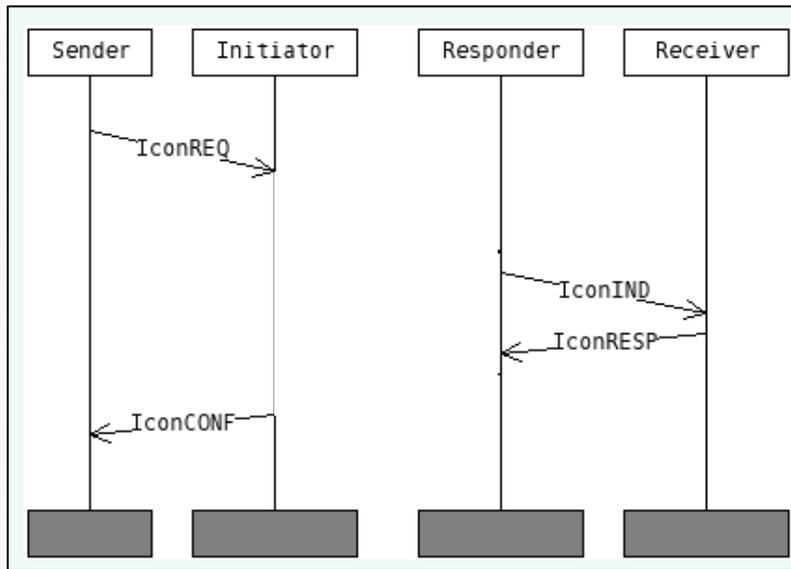
Verbindungsaufbau → Datenübertragung → Verbindungsabbau

Vereinfachung:
unsymmetrisches Protokoll

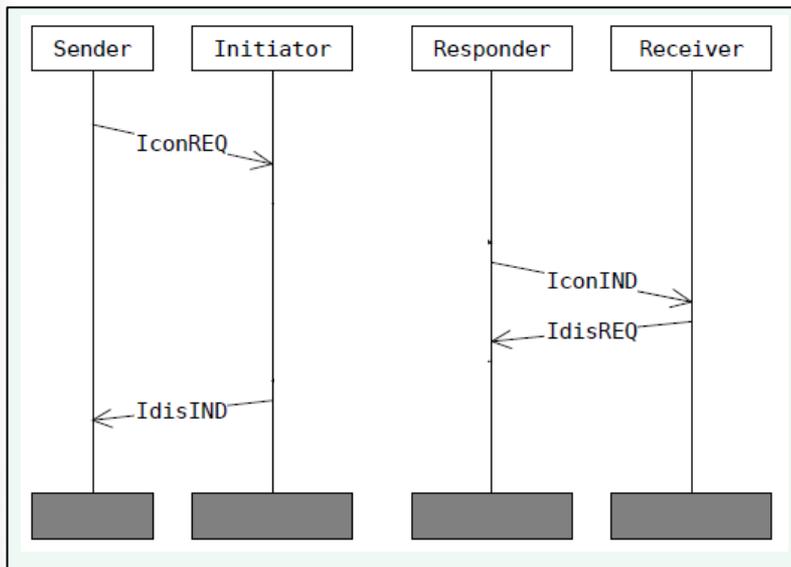
nur Sender

Empfänger,
InRes-Schicht

InRes-Dienst: Verbindungsaufbau

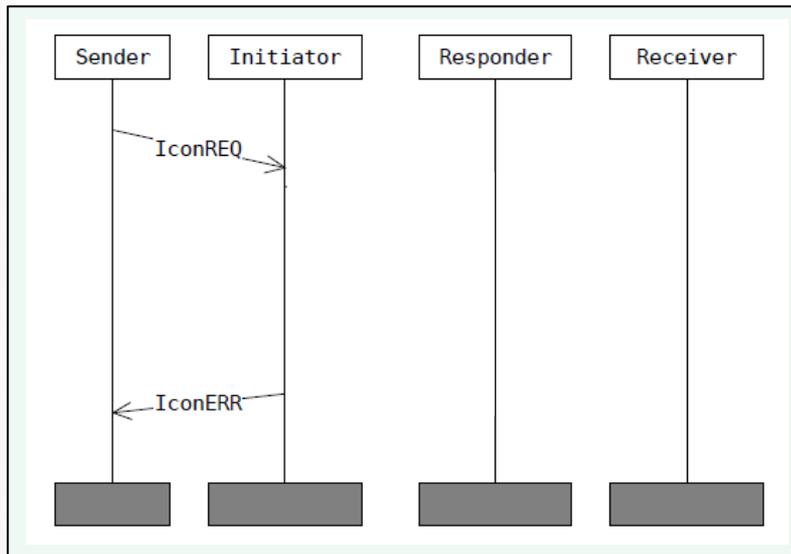


Erfolgreicher Verbindungsaufbau

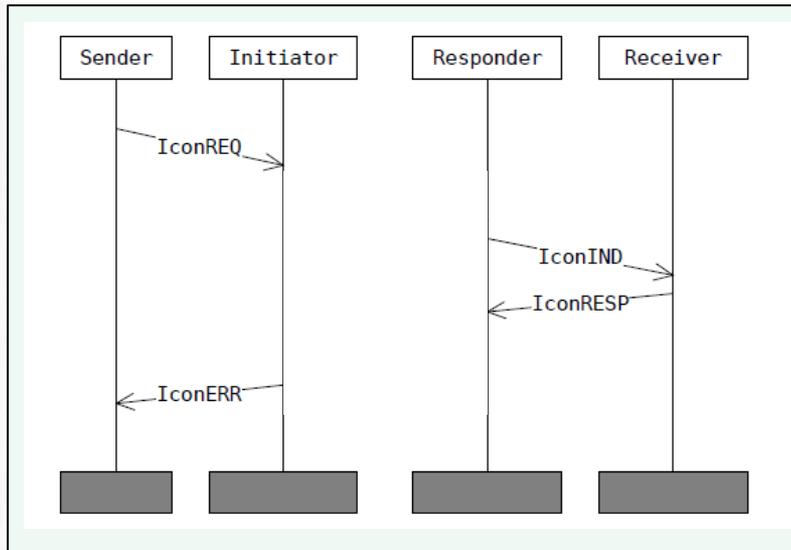


Erfolgloser Verbindungsaufbau
Ablehnung durch Empfänger

InRes-Dienst: Verbindungsaufbau

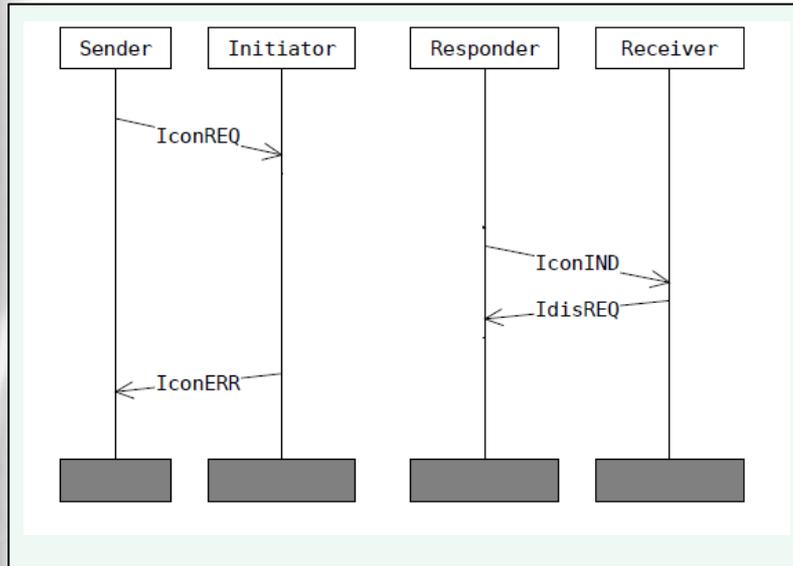


Erfolgloser Verbindungsaufbau
Übertragungsfehler
auf dem Weg zum Empfänger



Erfolgloser Verbindungsaufbau
Fehler bei der Übertragung der Bestätigung
auf dem Rückweg vom Empfänger

InRes-Dienst: Verbindungsaufbau

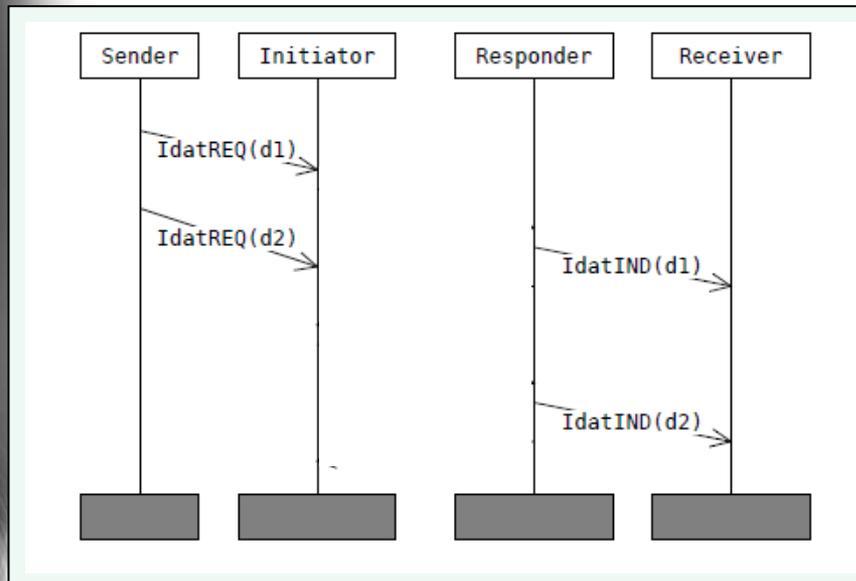


Erfolgloser Verbindungsaufbau

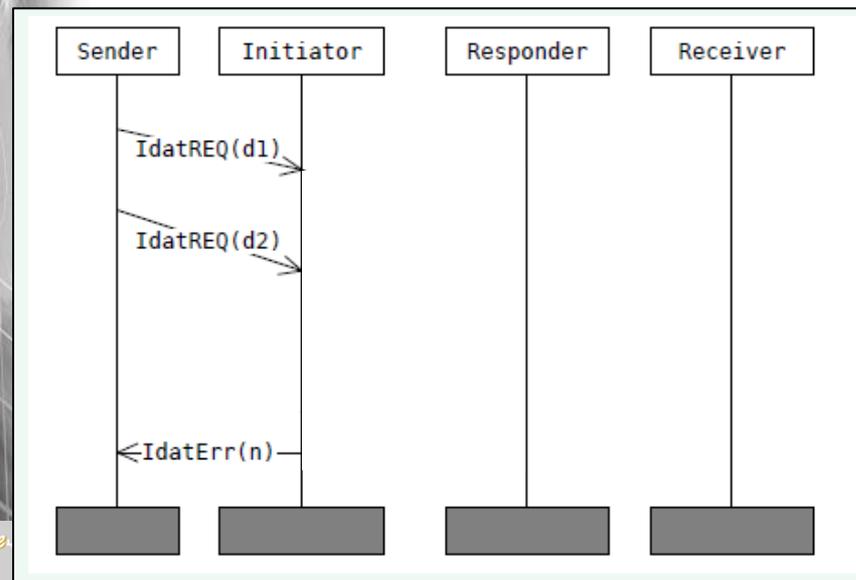
Fehler bei der Übertragung der Ablehnung
auf dem Rückweg vom Empfänger

drei verschiedene Ursachen für IconERR

InRes-Dienst: Datenübertragung

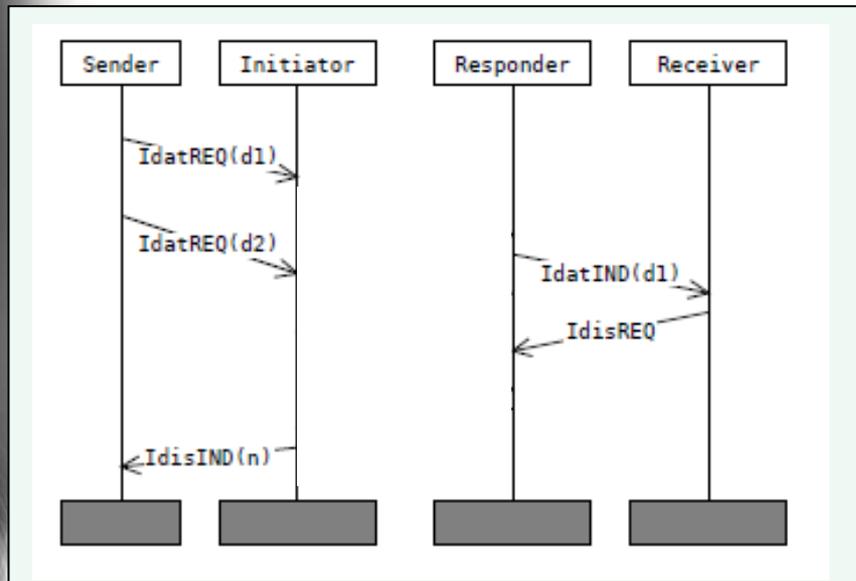


Erfolgreiche Datenübertragung



Erfolgreiche Datenübertragung
(= Verbindungsabbruch)

InRes-Dienst: Verbindungsabbau



Erfolgreicher Verbindungsabbau

Phasen:

Verbindungsaufbau → Datenübertragung → Verbindungsabbau

nur vom Sender

nur vom Empfänger

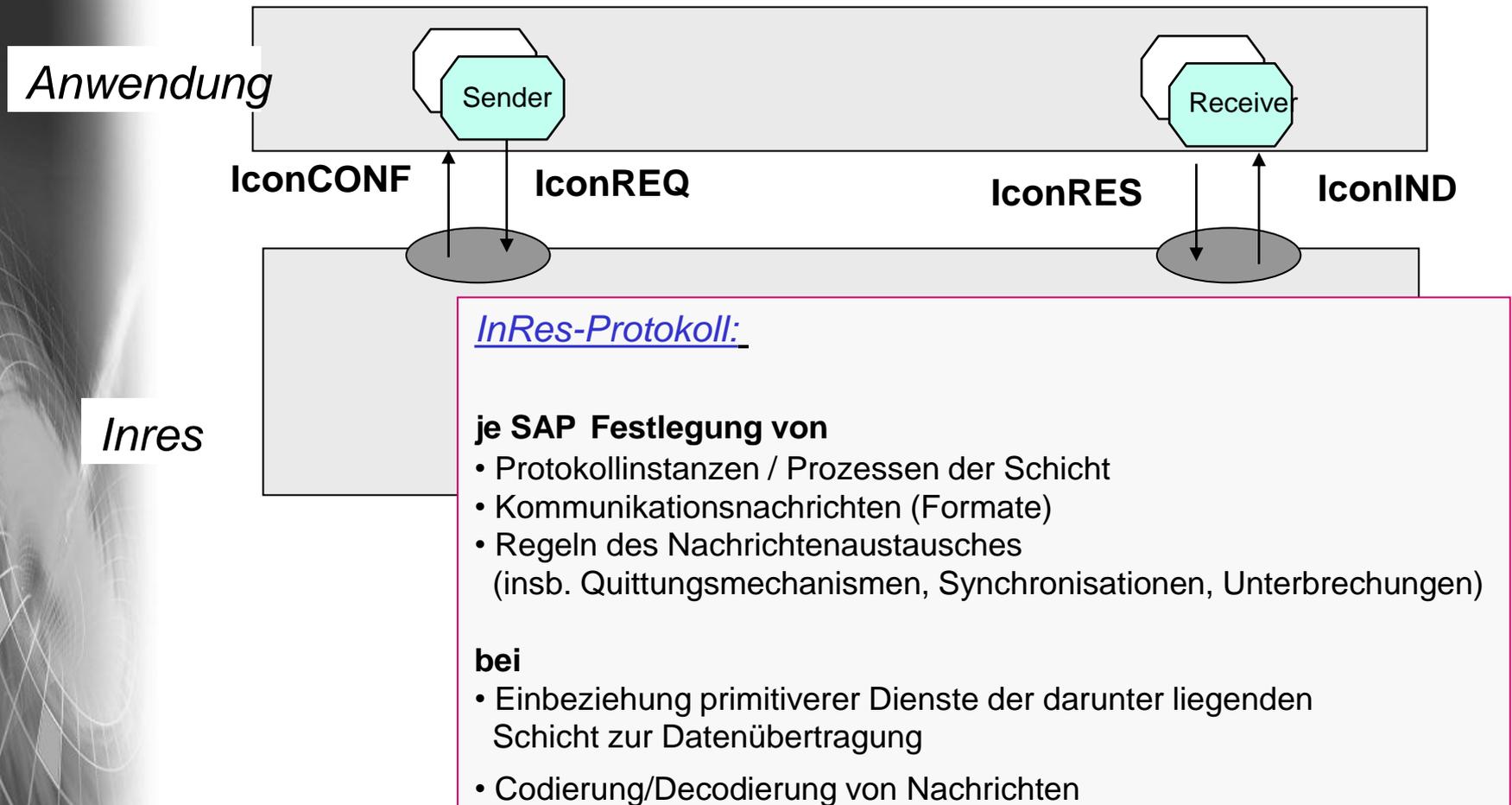
Dienstprimitive und Protokollnachrichten

Konzept	Kürzel	voller Name	Parameter
InRes-Dienstprimitive (Nachrichten)	IconREQ IconIND IconRESP IconCONF IconERR	InRes-Connection-Request InRes-Connection-Indication InRes-Connection-Response InRes-Connection-Confirm InRes-Connection-Error	ErrorType
	IdisREQ IdisIND	InRes-Disconnection-Request InRes-Disconnection-Indication	SequenceNumber
	IdatREQ IdatIND	Inres-Data-Request InRes-Data-Indication	ISDUType ISDUType
InRes-Protokolldateneinheiten	CR CC DAT AK DR	Connection-Request Connection-Confirm Data Acknowledge Disconnection-Request	SequenceNumber, ISDUType SequenceNumber SequenceNumber
Medium-Dienstprimitive	MdatREQ MdatIND	Medium-Data-Request Medium-Data-Indication	MSDUType MSDUType

InRes-Service-Data-Unit

Medium-Service-Data-Unit

InRes-Protokoll (1): zur Erbringung des InRes-Dienstes

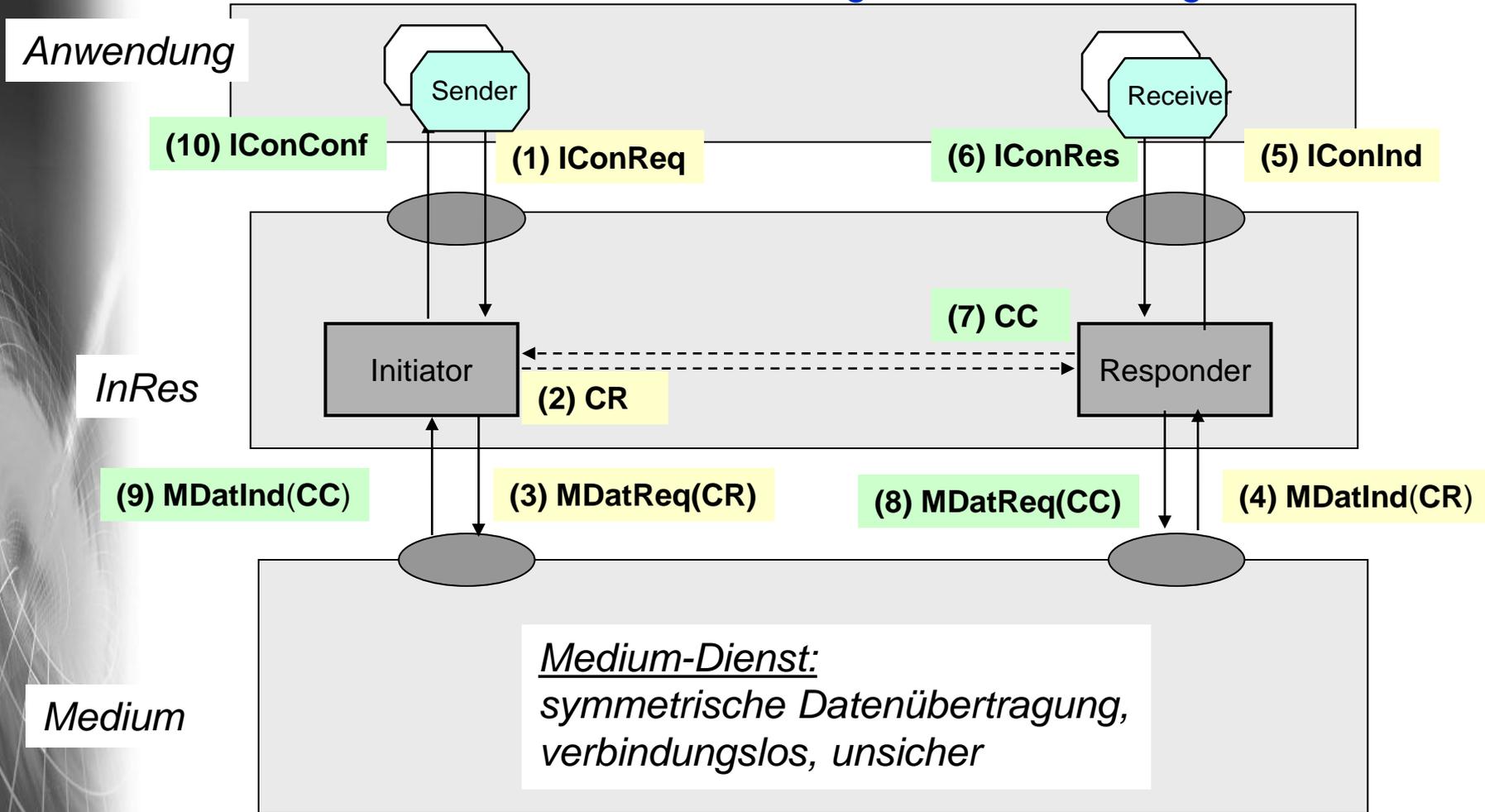


Protokolleinheiten müssen Einhaltung von folgendem Szenario sichern:

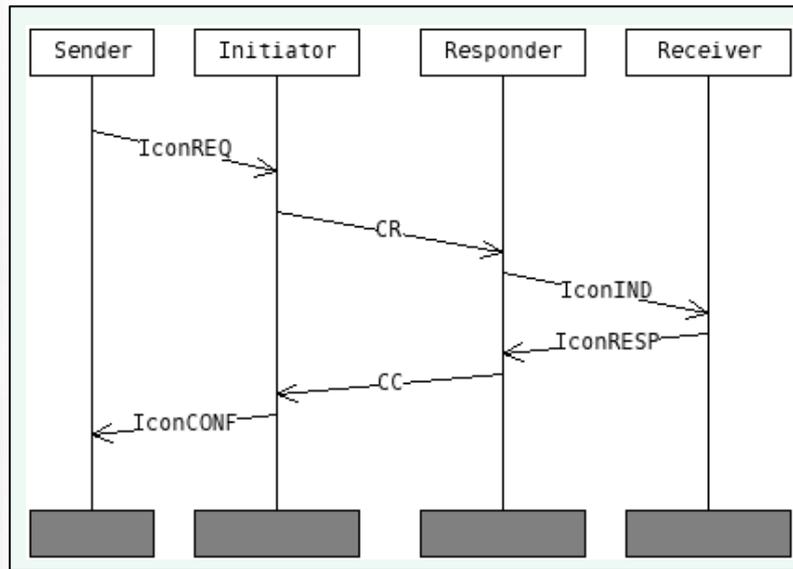
Verbindungsaufbau → Datenübertragung → Verbindungsabbau

InRes-Protokoll (2): zur Erbringung des InRes-Dienstes

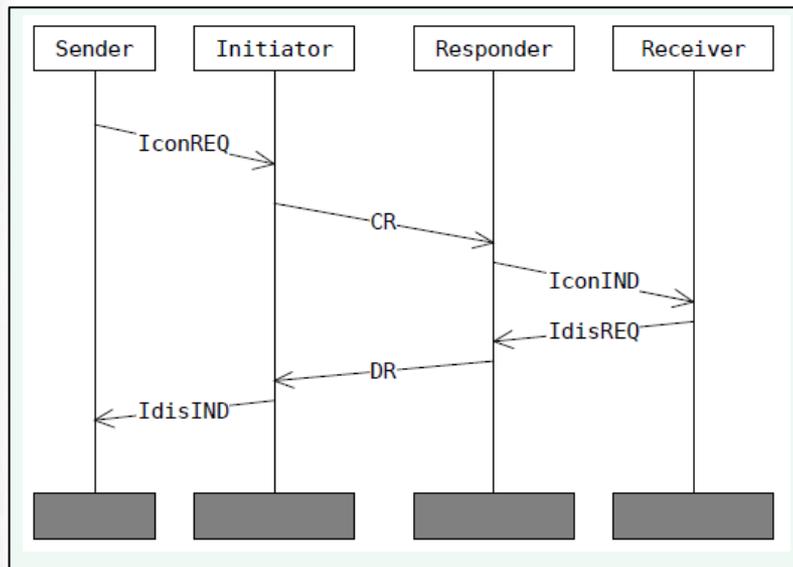
InRes-Protokoll: Erfolgreicher Verbindungsaufbau



InRes-Protokoll: Verbindungsaufbau

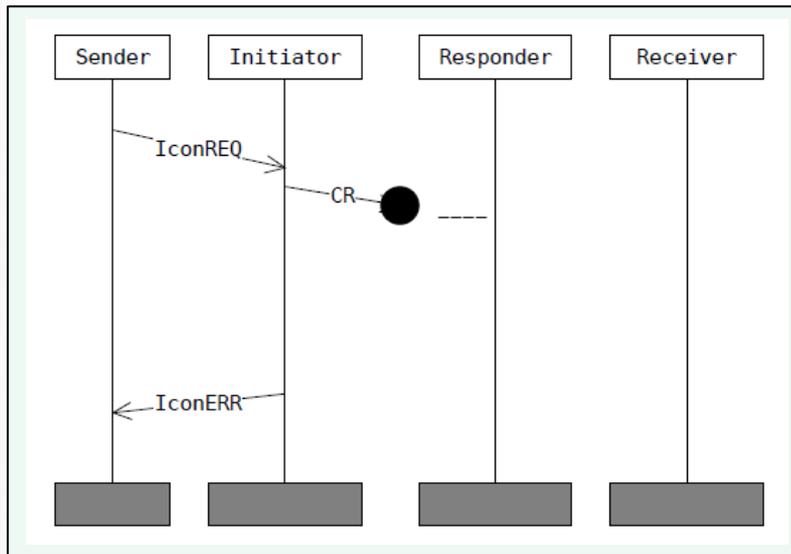


Erfolgreicher Verbindungsaufbau



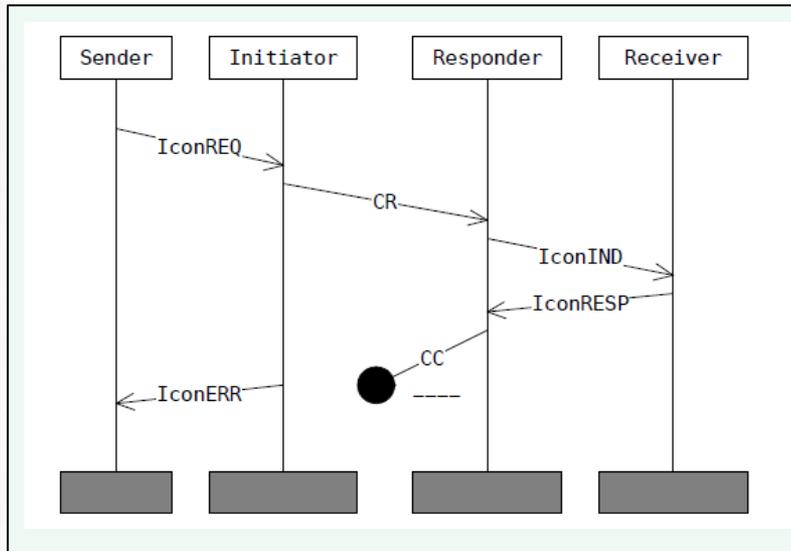
Erfolgreicher Verbindungsaufbau
Ablehnung durch Empfänger

InRes-Protokoll: Verbindungsaufbau



Erfolgloser Verbindungsaufbau
Übertragungsfehler
auf dem Weg zum Empfänger

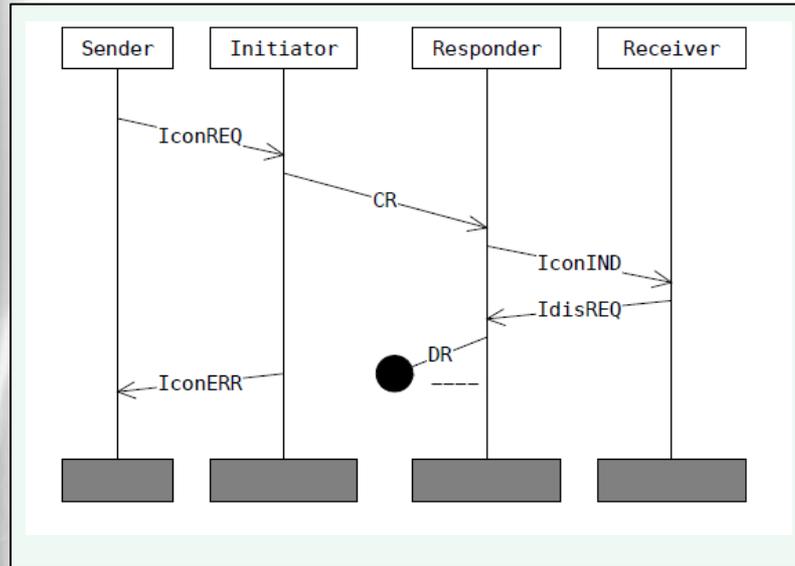
→ nach 3 Time-out-überwachten
Fehlversuchen
wird abgebrochen



Erfolgloser Verbindungsaufbau
Fehler bei der Übertragung der Bestätigung
auf dem Rückweg vom Empfänger

→ s.o.

InRes-Protokoll: Verbindungsaufbau

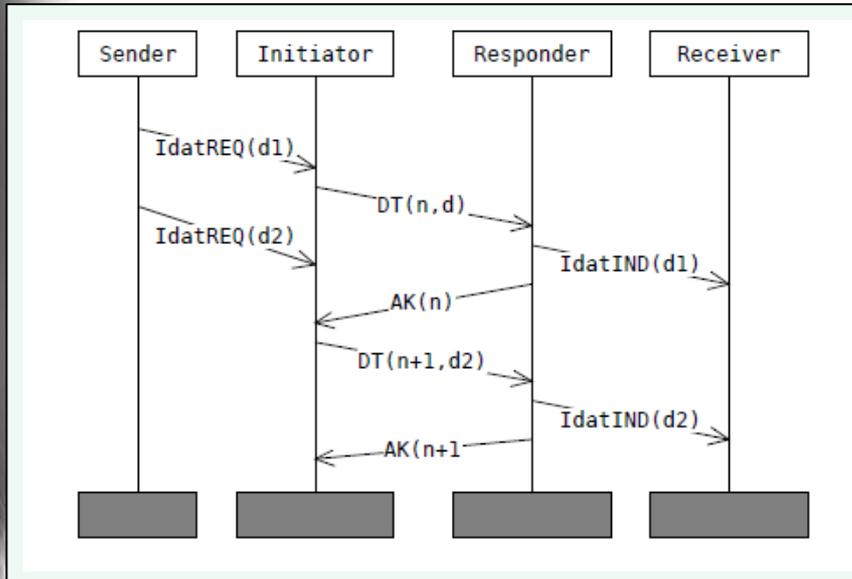


Erfolgreicher Verbindungsaufbau

Fehler bei der Übertragung der Ablehnung
auf dem Rückweg vom Empfänger

→ nach 3 Time-out-überwachten
Fehlversuchen
wird abgebrochen

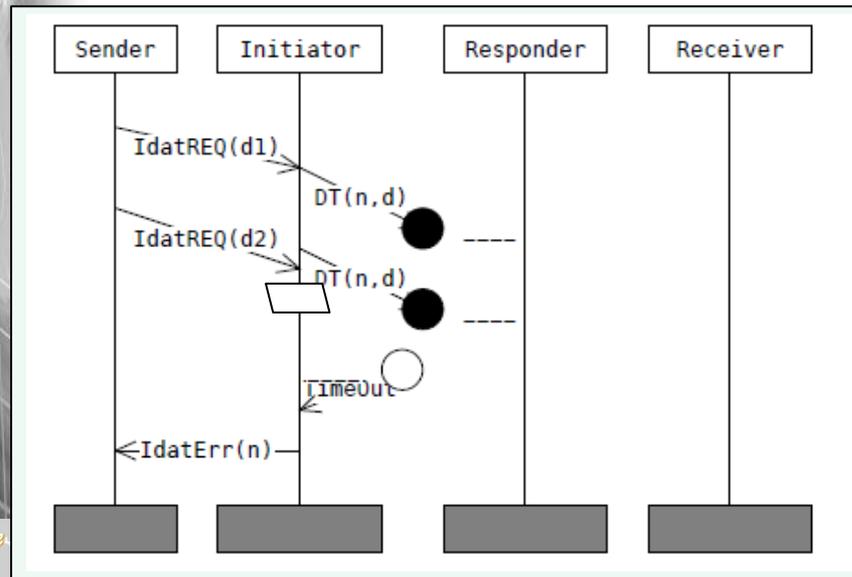
InRes-Dienst: Datenübertragung



Erfolgreiche Datenübertragung

Daten werden mit fortlaufender **Nummer** verschickt,
Nächste Daten werden erst dann verschickt,
Wenn eine **Quittung** den Erhalt durch die
Nummer des letzten Paketes bestätigt

→ Zwischenzeitlich ankommende Daten
müssen **gespeichert** werden

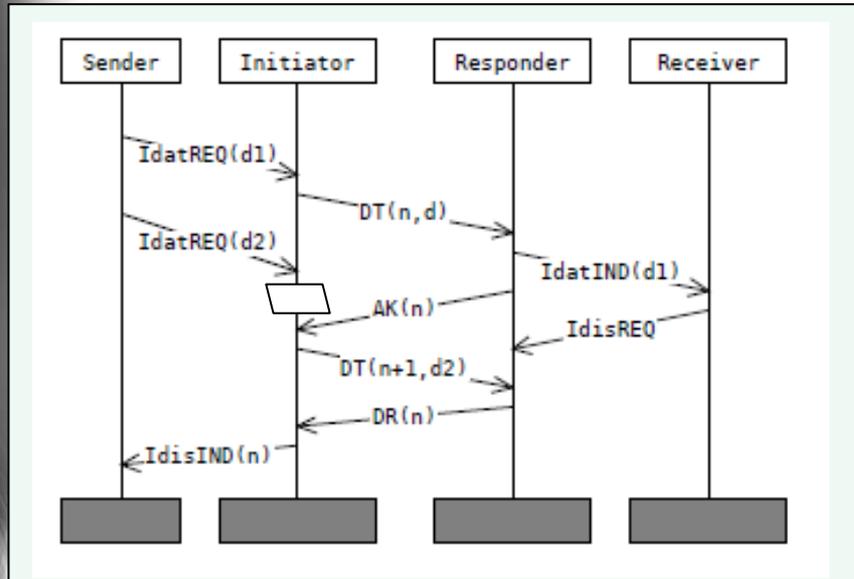


Erfolgreiche Datenübertragung (= Verbindungsabbruch)

→ nach 3 Time-out-überwachten
Fehlversuchen
wird **abgebrochen**

→ Sender erhält Information zur Anzahl
erfolgreicher Datenübertragungen

InRes-Dienst: Verbindungsabbau



Erfolgreicher Verbindungsabbau

7. *Protokollentwicklung in SDL*

- OSI-Schichtenmodell (Konzept)
- InRes-Protokoll (Pseudo-Rechnernetzprotokoll)
- Umsetzung in SDL/RT

Allgemeine Charakteristik des Inres-Protokolls

1. Verbindungsaufbau

- Nutzer (am ISAPini) sendet seinem Partner der Anwendungsschicht eine **Verbindungsaufbauanforderung**, die bestätigt oder abgelehnt werden kann
- Protokolleinheit am ISAPini (der Initiator) sendet timeoutüberwacht eine Protokollnachricht (CR) an den Partner der InRes-Schicht mit maximal **3-facher** Wiederholung
- Initiator informiert seinen Nutzer über das Resultat der Verbindungsaufnahme:
 - a) **IconIND** bei erfolgreicher Verbindungsaufbau
 - b) **IerrIND** bei erfolglosem Verbindungsaufbau (ohne Antwort der Partners)
 - c) **IdisIND** nach Ablehnung durch den Partner

2. Datenübertragung nach erfolgreichem Verbindungsaufbau

Nutzer (am ISAPini) sendet in **beliebiger Frequenz** Daten

- **Initiator** muss alle zwischenspeichern und in der Empfangsreihenfolge weiterschicken
- **Responder** wartet u.U. ewig auf Daten
- **Responder** schickt empfangene Daten nur **einmal** an seinen Nutzer (ISAPresp) unabhängig, wie oft er diese tatsächlich empfangen haben sollte

Allgemeine Charakteristik des Inres-Protokolls (2)

2. Datenübertragung

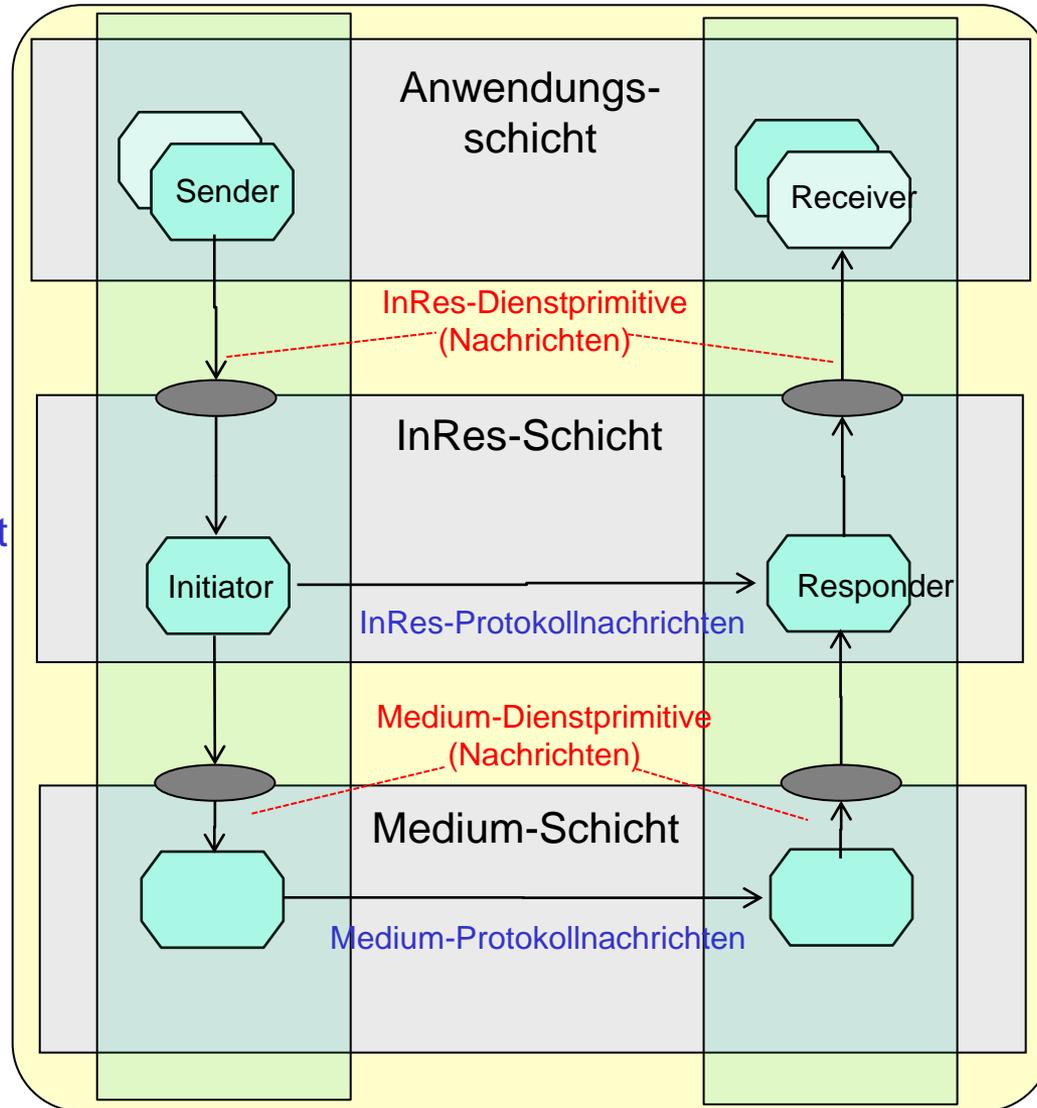
Art der Synchronisation der Protokolleinheiten: **Handshake-Verfahren**

- **Initiator** wartet mit der Übertragung des nächsten Signals solange bis er eine Quittung mit der Nummer des gesendeten Datums vom **Responder** erhält
→ schlechte Performanz
- bei **Timeout** oder falscher **Quittung**: sendet Initiator erneut (max. 3mal)

3. Verbindungsabbau

- Auslöser: nur **Empfänger** oder **Initiator** (im Fehlerfall)
praktisch unzureichend:
Sender kann nicht die Verbindung bei kompletter Datenübertragung beenden

InRes-Schichtenarchitektur



InRes-Protokoll:
 unsymmetrisch,
 verbindungsorientiert

Medium-Protokoll:
 symmetrisch,
 verbindungslos

InRes-Dienste

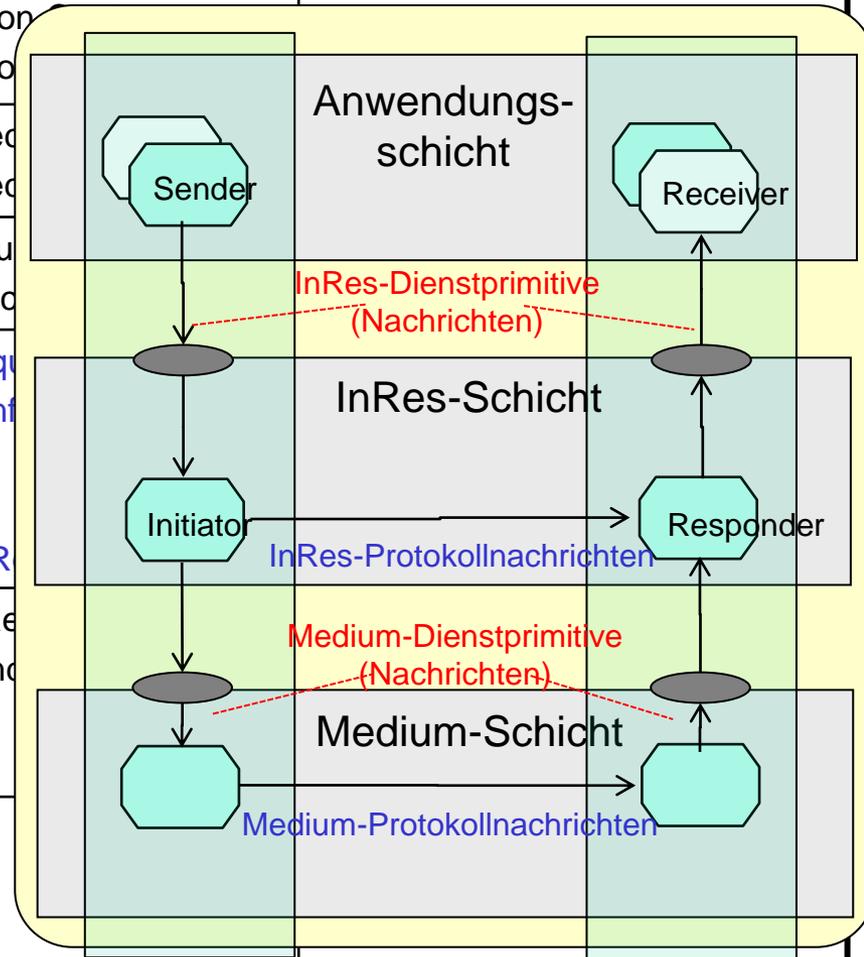
überwachte
 (sichere)
 Datenübertragung

Medium-Dienste

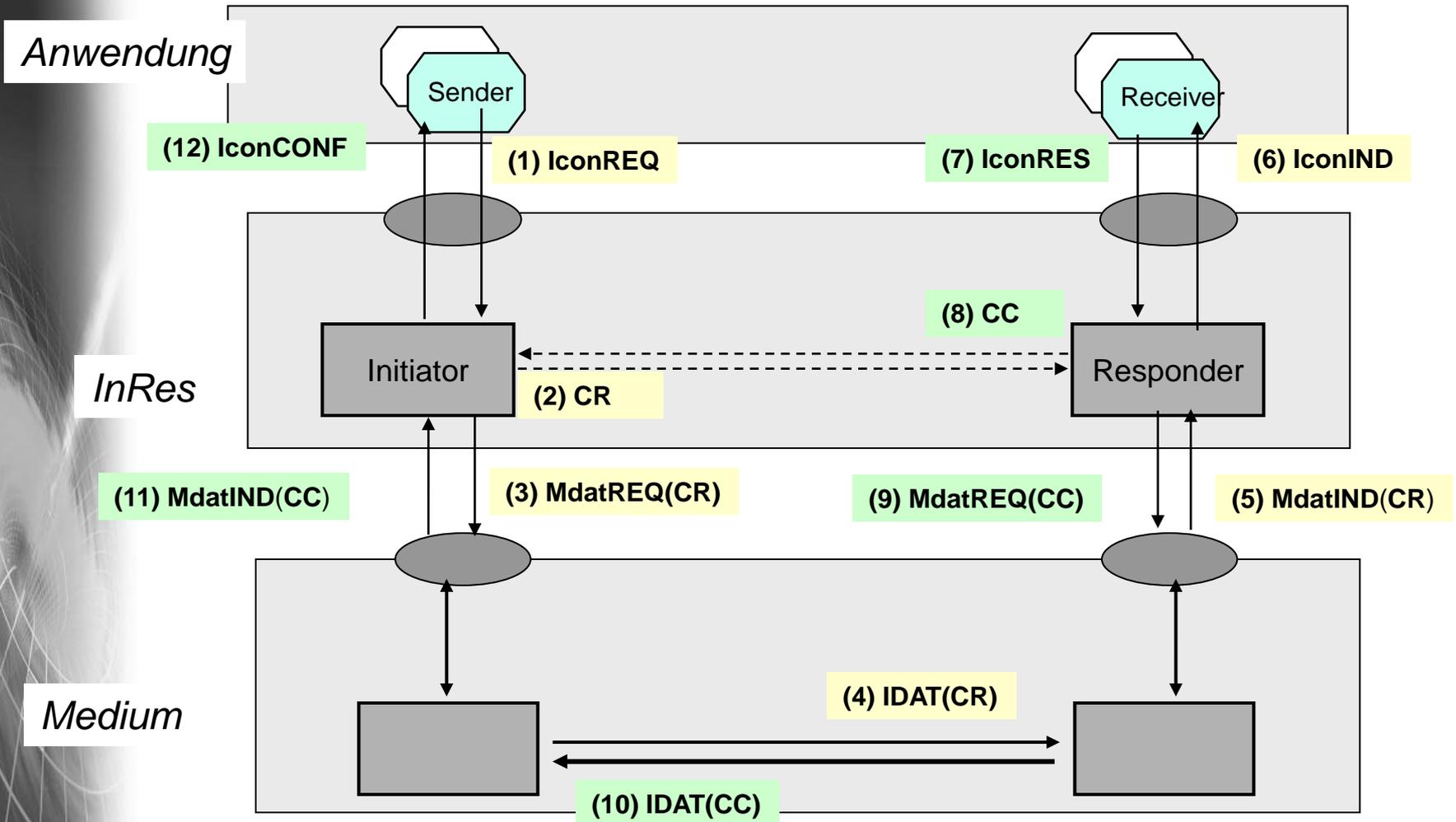
unsichere
 Datenübertragung

Dienstprimitive und Protokollnachrichten

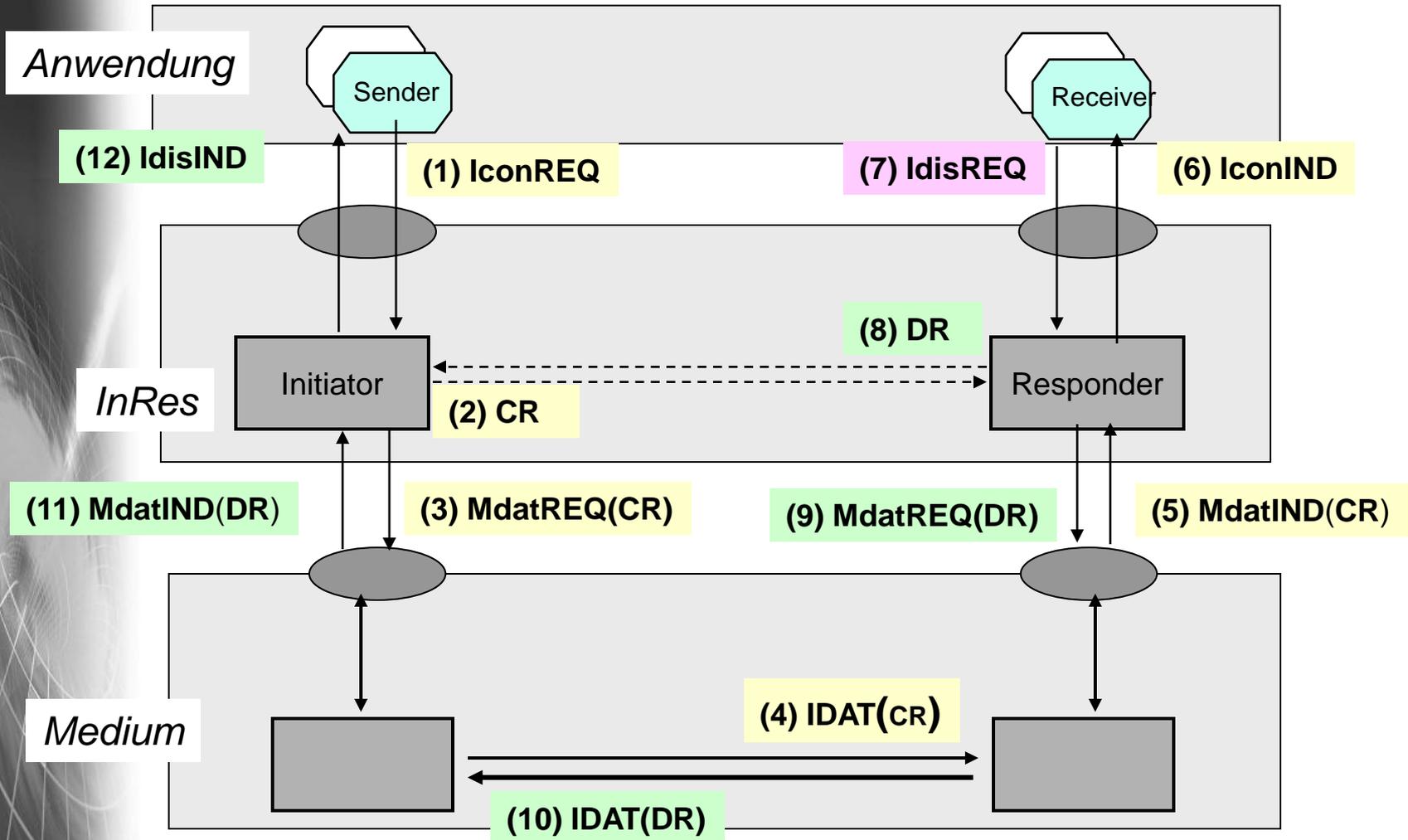
Konzept	Kürzel	voller Name	Parameter
InRes-Dienstprimitive (Nachrichten zwischen Anwendungsschicht und InRes-Schicht)	IconREQ	InRes-Connection-Request	
	IconIND	InRes-Connection-Indication	
	IconRESP	InRes-Connection-Response	
	IconCONF	InRes-Connection	
	lerrIND	InRes-Connection	
	IdisREQ	InRes-Disconnected	
	IdisIND	InRes-Disconnected	
	IdatREQ	Inres-Data-Request	
	IdatIND	InRes-Data-Indication	
InRes-Protokolldateneinheiten (Nachrichten der Inres-Schicht zur Initiator-Responder-Kommunikation)	CR	Connection-Request	
	CC	Connection-Conf	
	DAT	Data	
	AK	Acknowledge	
	DR	Disconnection-R	
Medium-Dienstprimitive (Nachrichten zwischen Anwendungsschicht und InRes-Schicht)	MdatREQ	Medium-Data-Request	
	MdatIND	Medium-Data-Indication	
Medium-Protokolldateneinheit (Nachrichten zwischen InRes-Schicht und Medium-Schicht)	IDAT	InRes-Data	



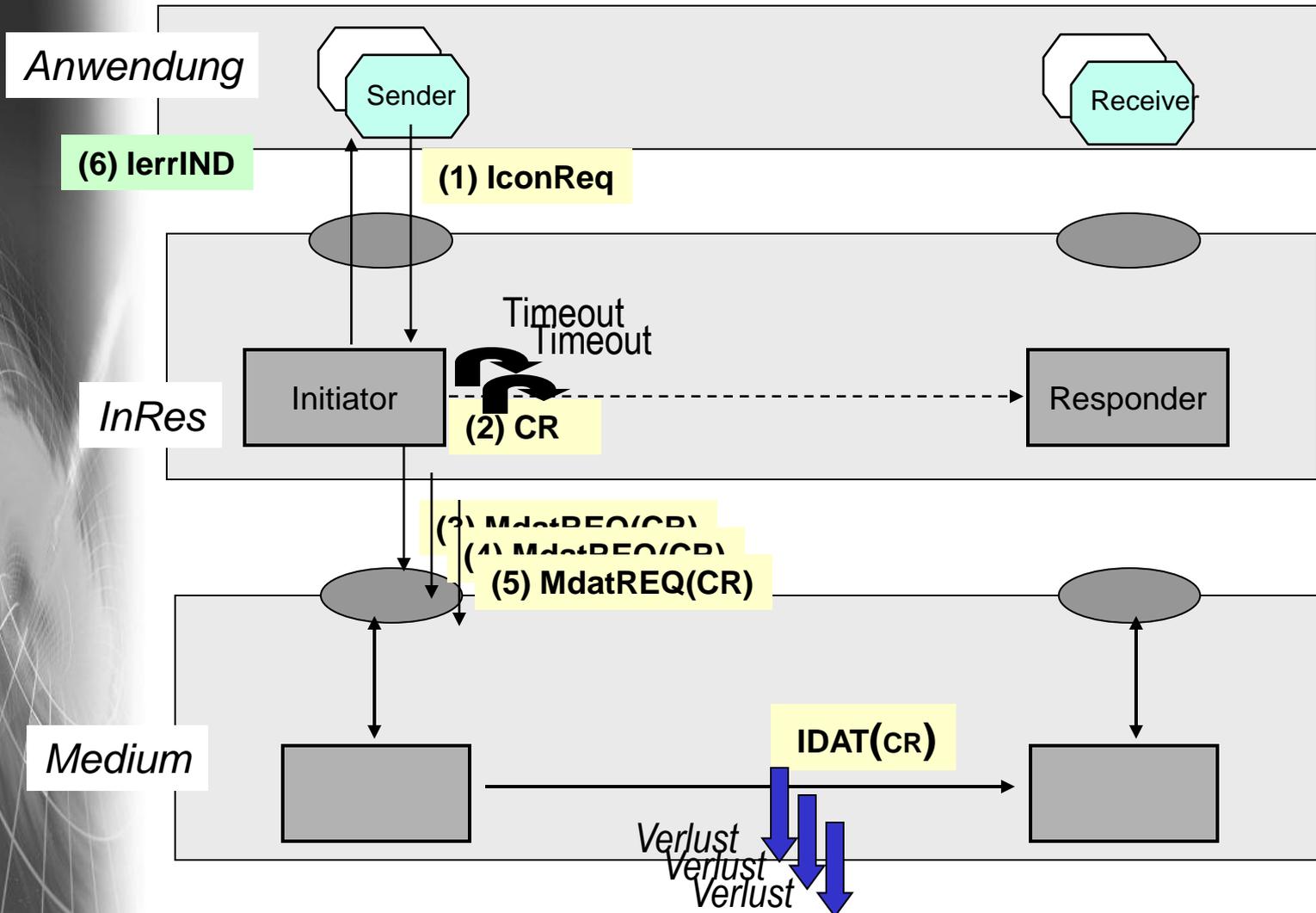
Erfolgreicher Verbindungsaufbau



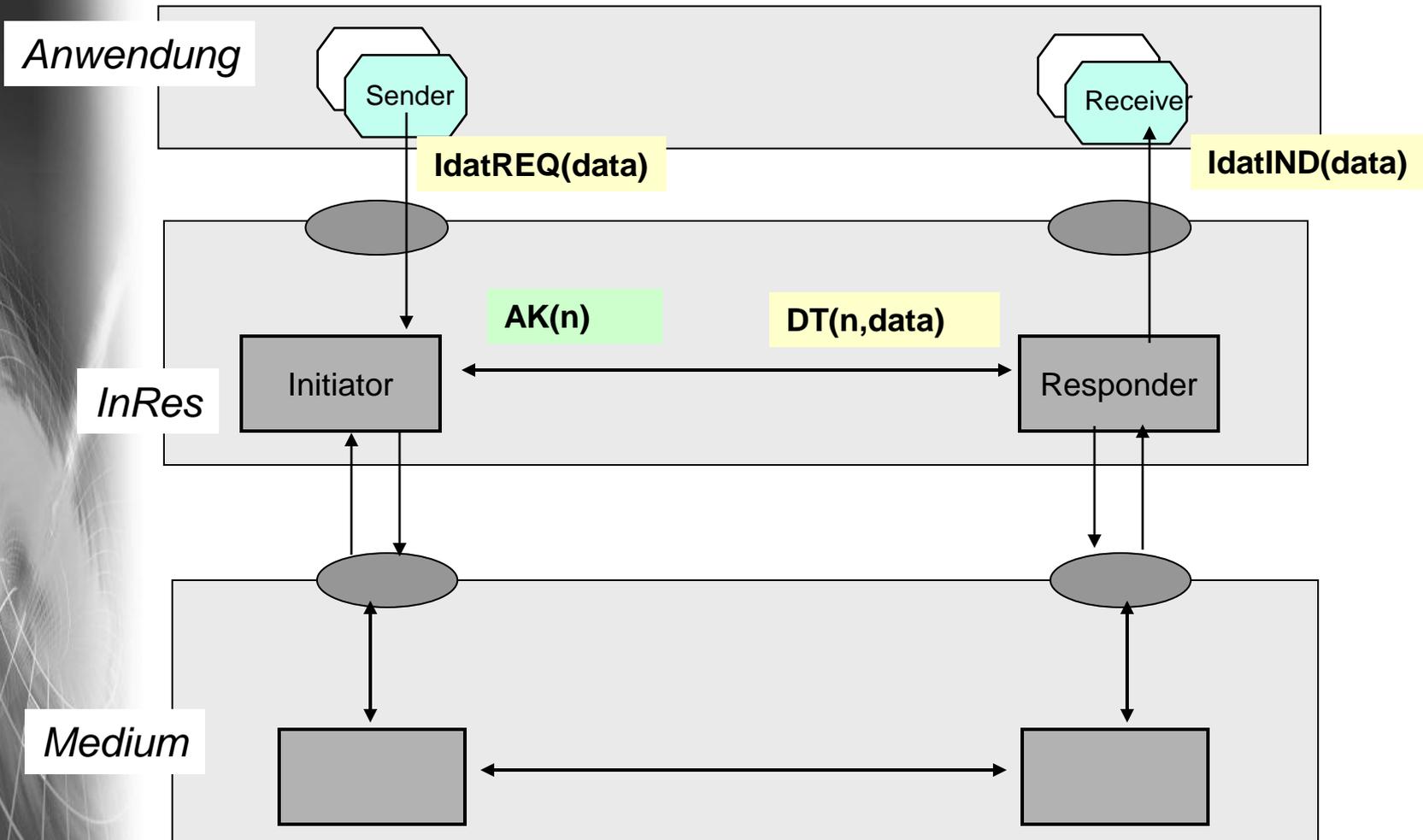
Abgelehnter Verbindungsaufbau



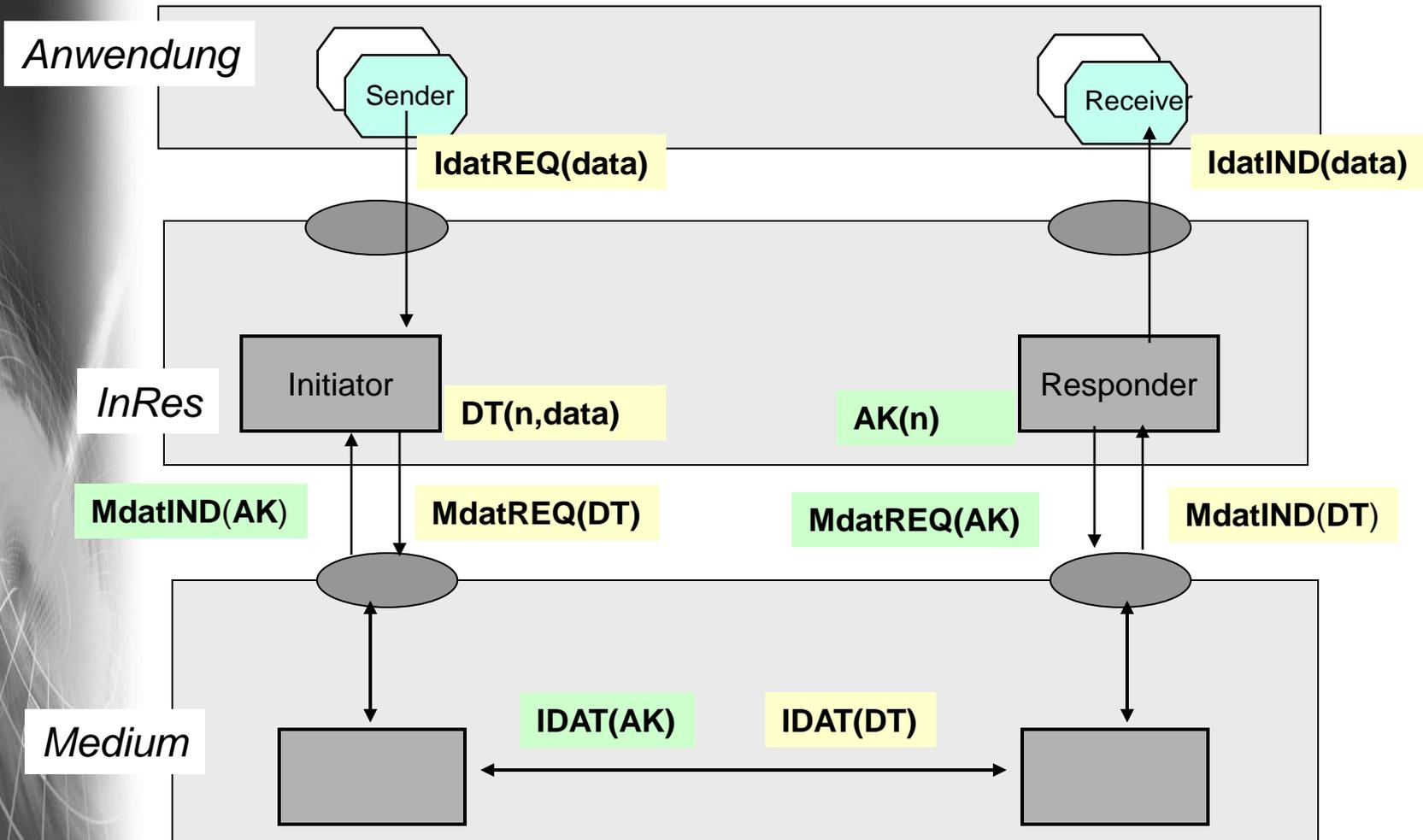
Abgebrochener Verbindungsaufbau



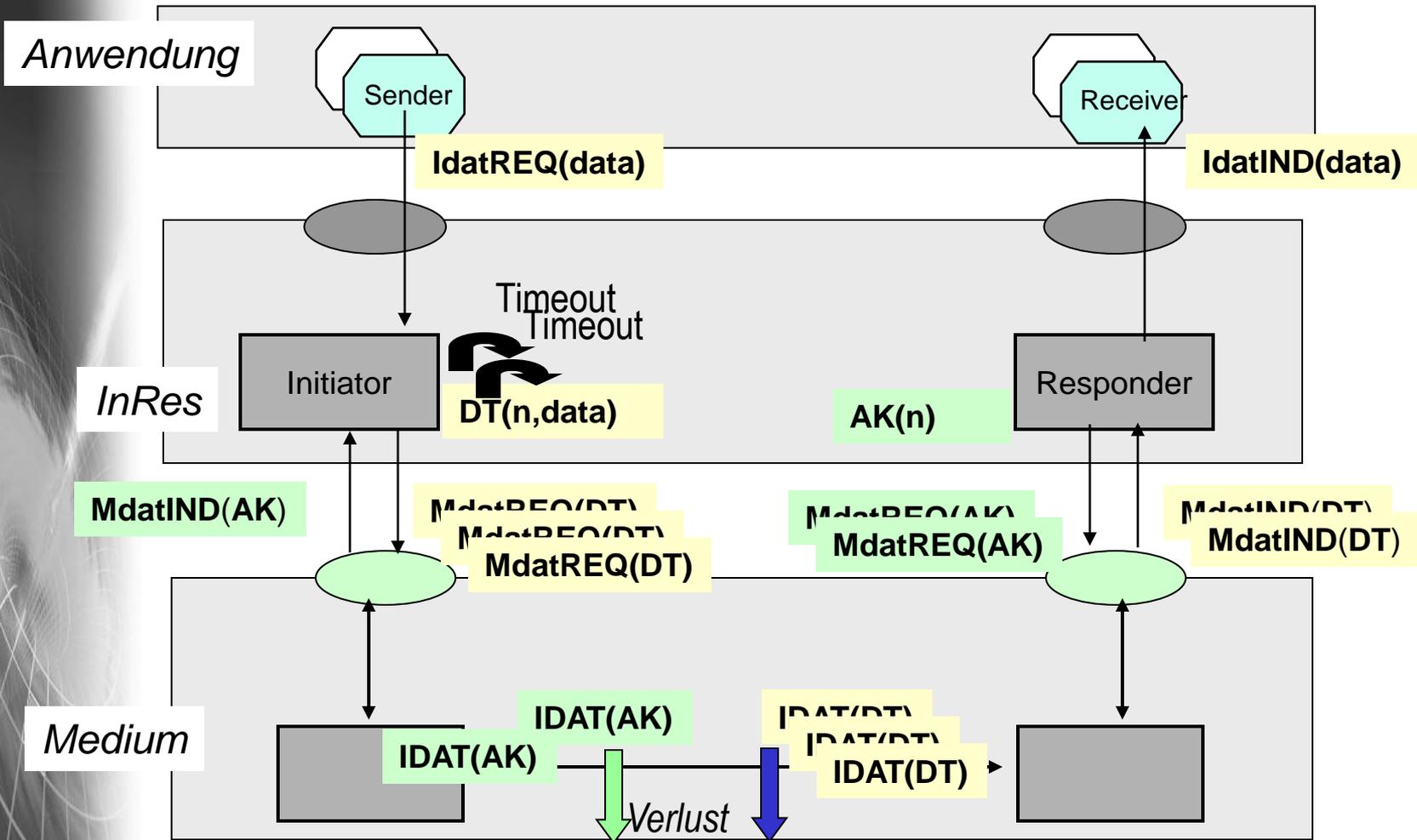
InRes-Dienst: Sichere Datenübertragung



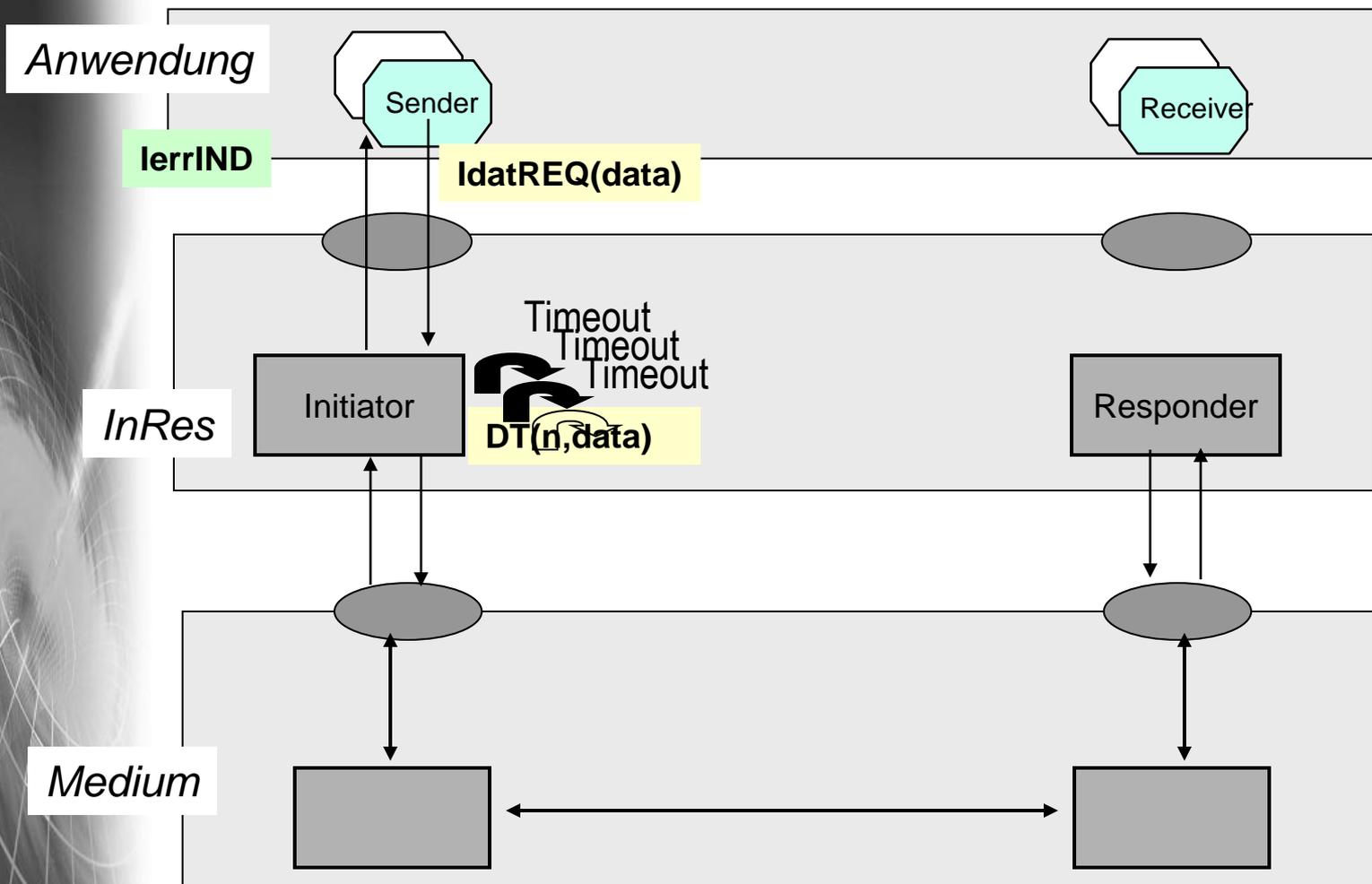
InRes-Protokoll: Sichere Datenübertragung



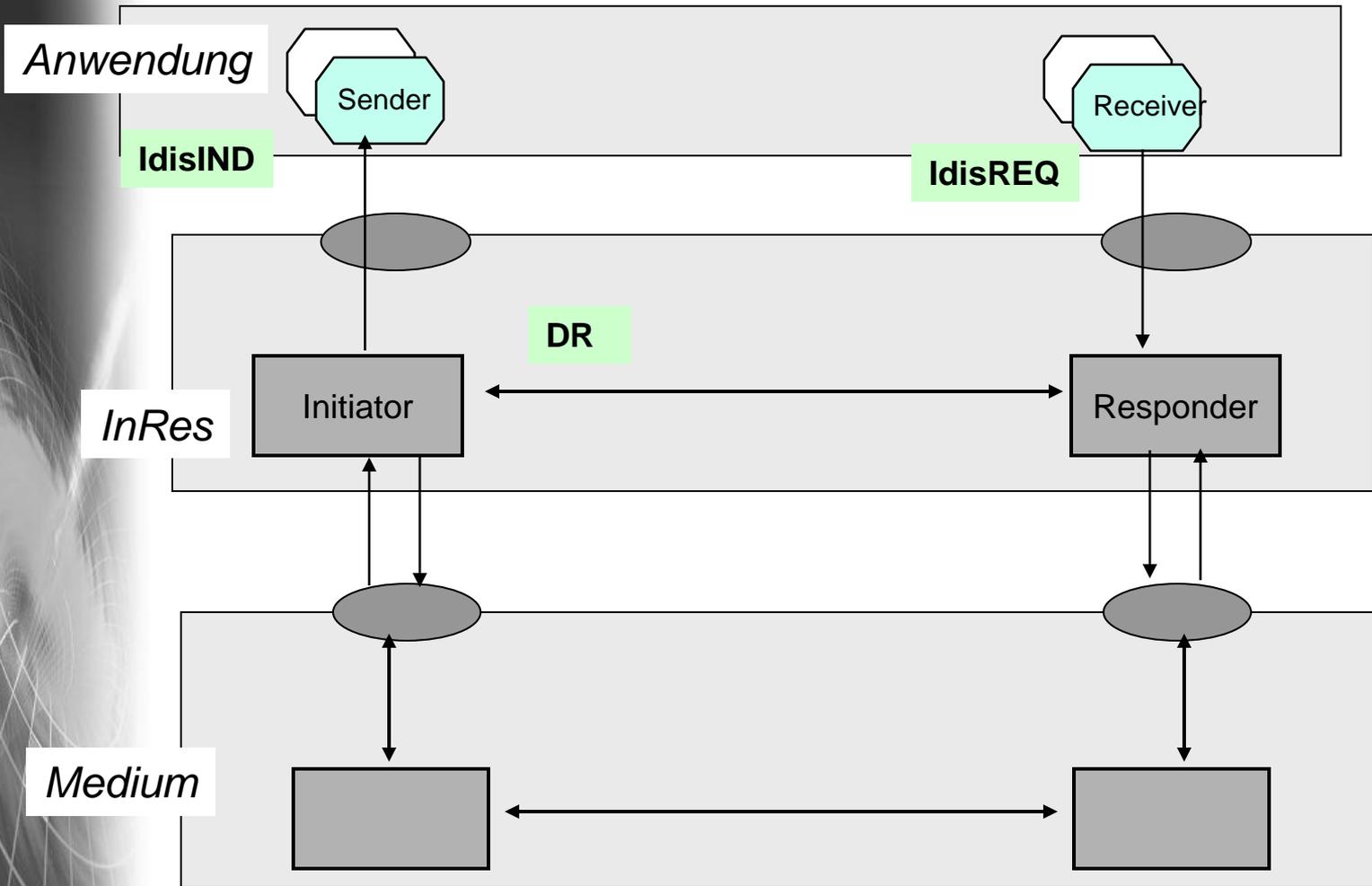
InRes-Protokoll: Sichere Datenübertragung



InRes-Protokoll: Sichere Datenübertragung



InRes-Dienst: Verbindungsabbau



Was passiert, wenn DR verloren geht?

Allgemeine Protokollarchitektur

