Modul OMSI-2
im SoSe 2010

Objektorientierte Simulation mit ODEMx

Prof. Dr. Joachim Fischer
Dr. Klaus Ahrens
Dipl.-Inf. Ingmar Eveslage
Dipl.-Inf. Andreas Blunk

fischer|ahrens|eveslage|blunk@informatik.hu-berlin.de

8. Protokollentwicklung in SDL

- OSI-Schichtenmodell (Konzept)
- InRes: allgemeine informale Beschreibung
- InRes-Dienstbeschreibung
- InRes-Protokollbeschreibung
- InRes-Semantik (Laufzeit-Betrachtung)
- Umsetzung in SDL/RT
- weiterer Ausbau

InRes-Kommunikation

InRes = Initiator-Responder
(als asymmetrische 2-Schichten Kommunikationsvariante)

InRes-Dienst (der InRes-Schicht)

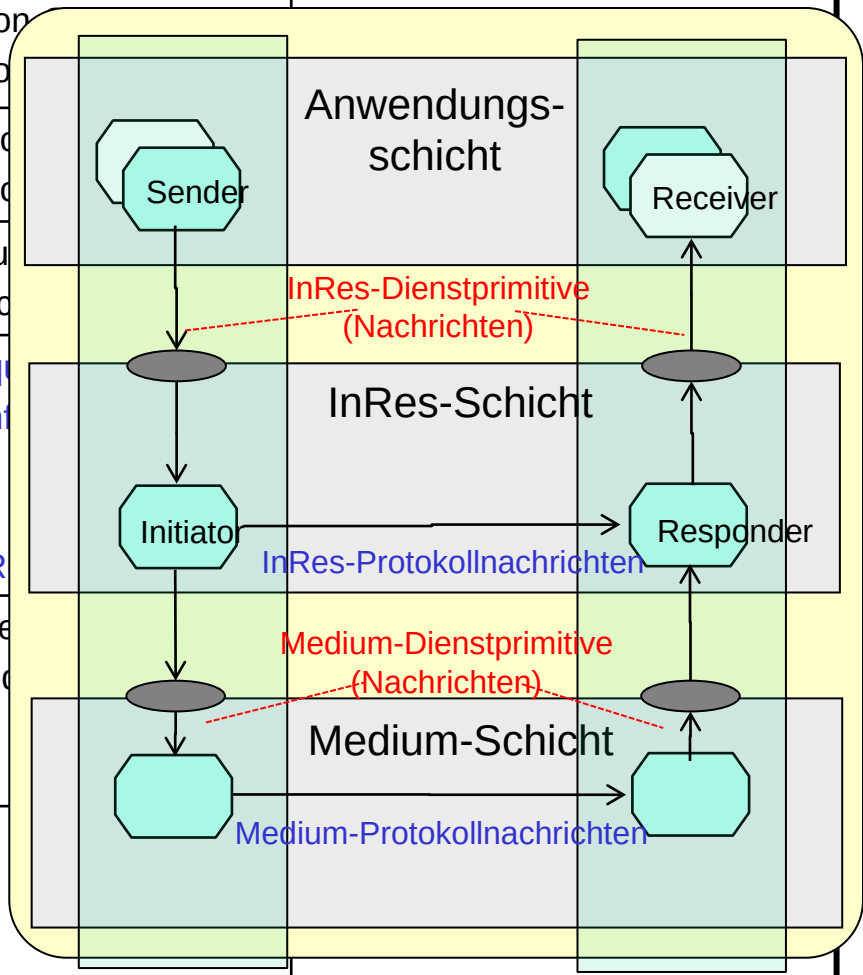
- als sichere Datenübertragung über ein unsicheres Medium
- wird realisiert über das verbindungsorientierte INRES-Protokoll unter
  Nutzung eines Low-Level-Dienstes zur unsichreren Datenübertragung (Medium-Dienst)
- Phasen:
  Verbindungsaufbau, Datenübertragung, Verbindungsabbau

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 zwischen Anwendungsschicht und InRes-Schicht)	IconREQ IconIND IconRESP IconCONF IerrIND	InRes-Connection-Request InRes-Connection-Indication InRes-Connection-Response InRes-Connection InRes-Connectio	
	IdisREQ IdisIND	InRes-Disconnec InRes-Disconnec	
	IdatREQ IdatIND	Inres-Data-Requ InRes-Data-Indic	
InRes-Protokolldateneinheiten (Nachrichten der Inres-Schicht zur Initiator-Responder-Kommunikation)	CR CC DAT AK DR	Connection-Requ Connection-Conf Data Acknowledge Disconnection-R	
Medium-Dienstprimitive (Nachrichten zwischen Anwendungsschicht und InRes-Schicht)	MdatREQ MdatIND	Medium-Data-Re Medium-Data-Ind	
Medium-Protokolldateneinheit (Nachrichten zwischen InRes-Schicht und Medium-Schicht)	IDAT	InRes-Data	



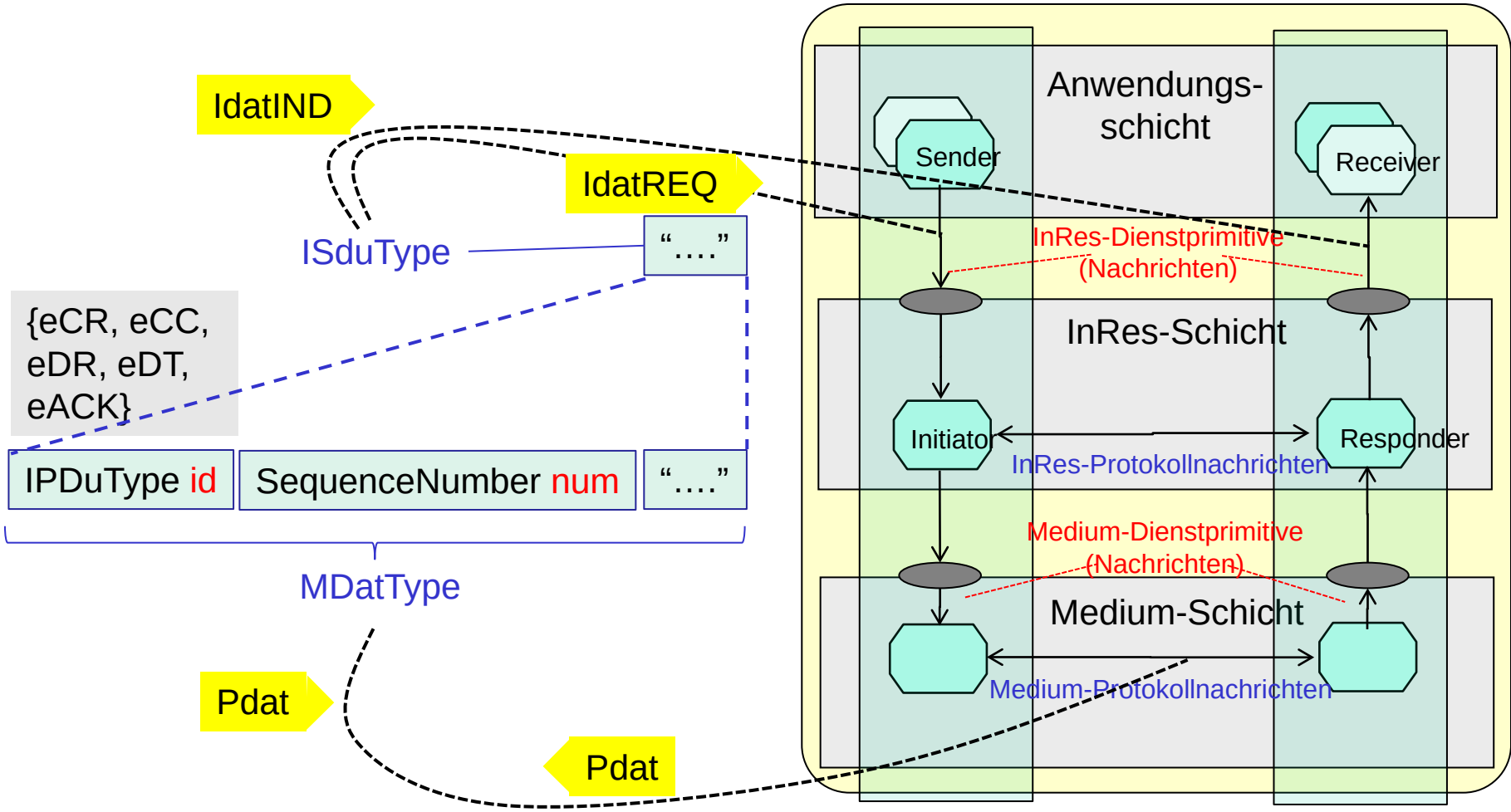
Dienstprimitive und Protokollnachrichten

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

Datenpakete: Nutzer, Übertragungsmedium

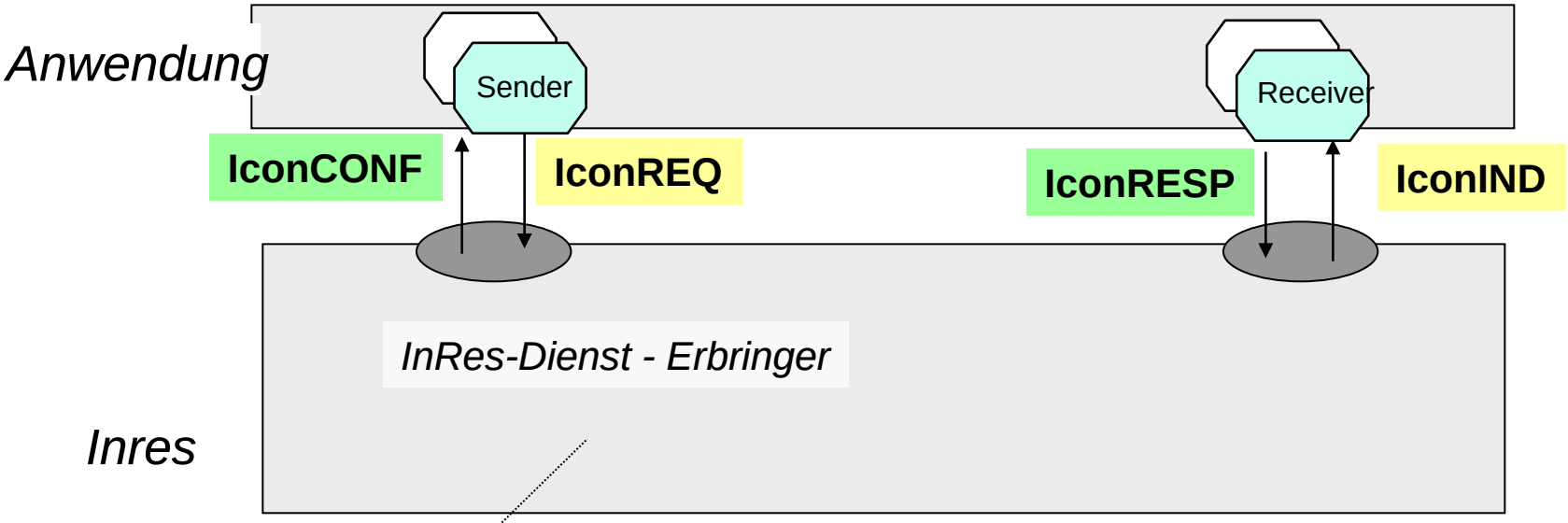


8. Protokollentwicklung in SDL

- OSI-Schichtenmodell (Konzept)
- InRes: allgemeine informale Beschreibung
- InRes-Dienstbeschreibung
- InRes-Protokollbeschreibung
- InRes-Semantik (Laufzeit-Betrachtung)
- Umsetzung in SDL/RT
- weiterer Ausbau

InRes-Dienst: *verbindungsorientierte sichere Datenübetragung*

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



Phasen:

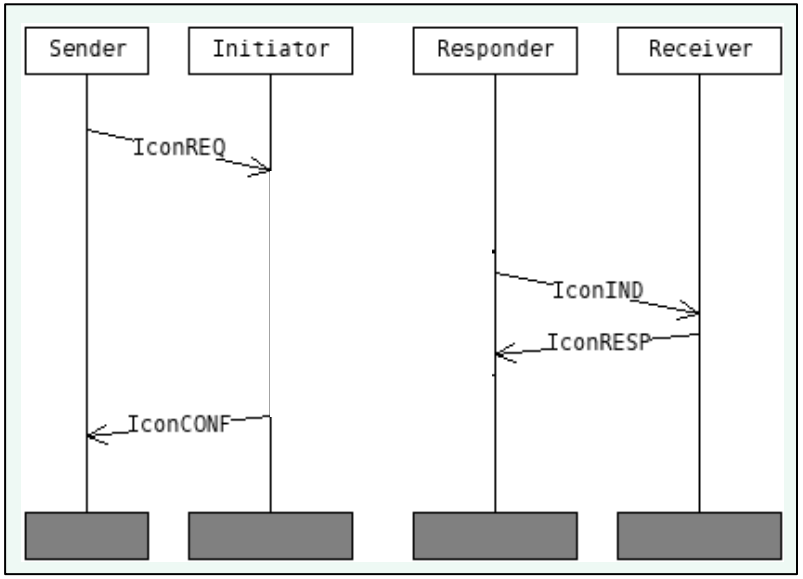
Verbindungsaufbau → Datenübertragung → Verbindungsabbau

Vereinfachung:
unsymmetrisches Protokoll

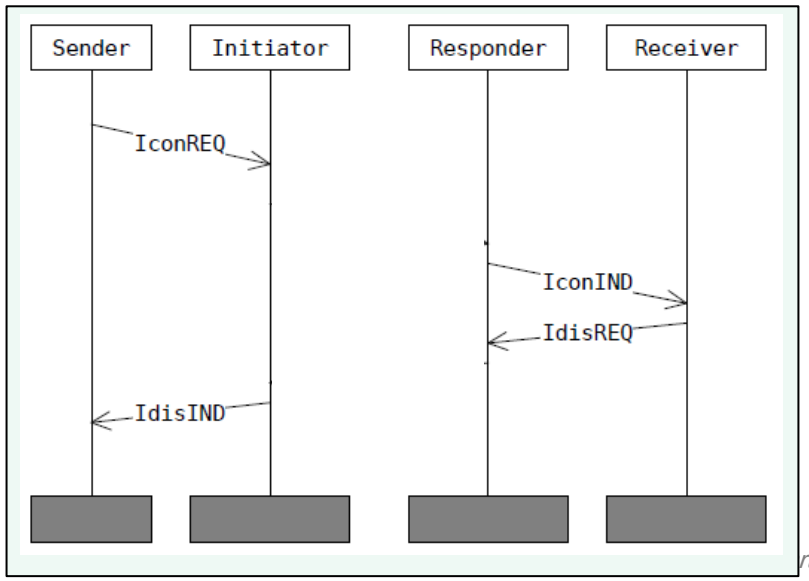
nur Sender

Empfänger,
InRes-Schicht

InRes-Dienst: Verbindungsaufbau

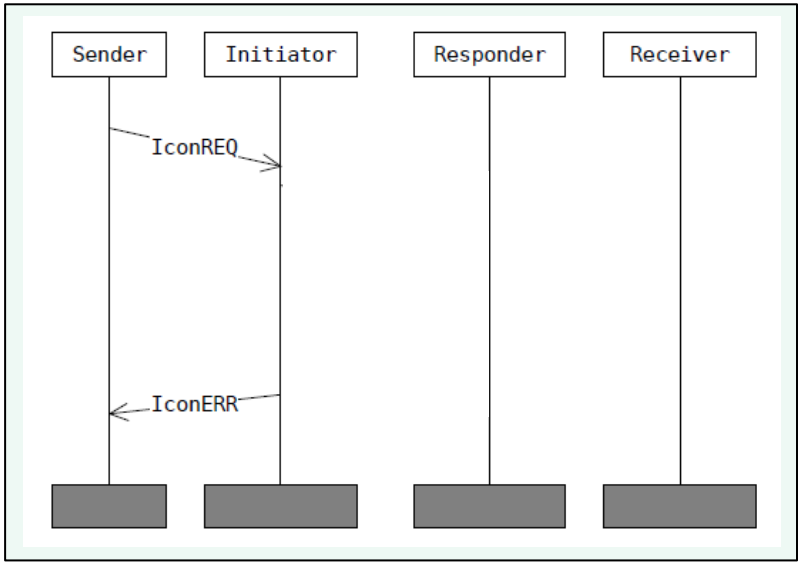


Erfolgreicher Verbindungsaufbau

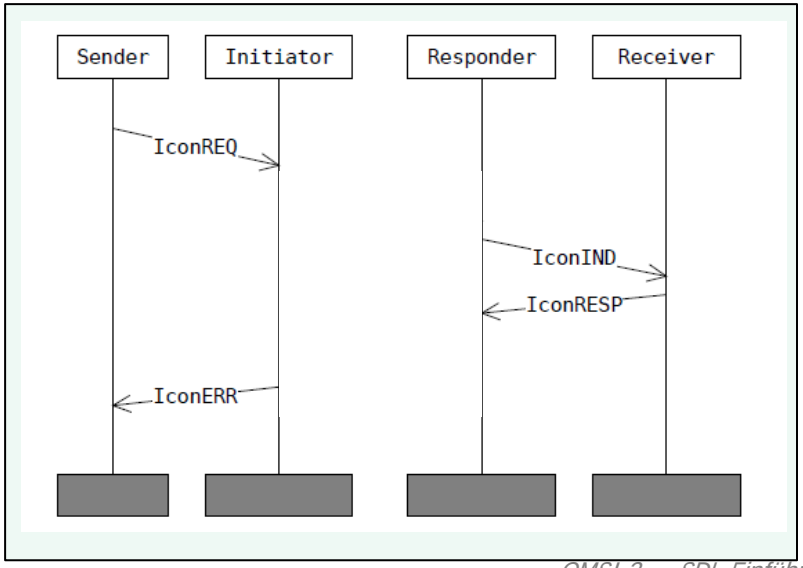


Erfolgloser Verbindungsaufbau
Ablehnung durch Empfänger

InRes-Dienst: Verbindungsaufbau

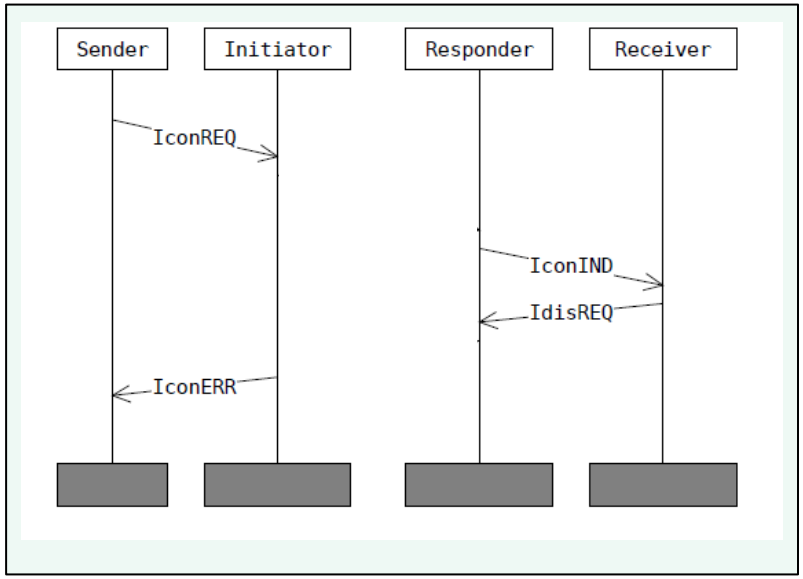


Erfolgloser Verbindungsaufbau
Übertragungsfehler
auf dem Weg zum Empfänger



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

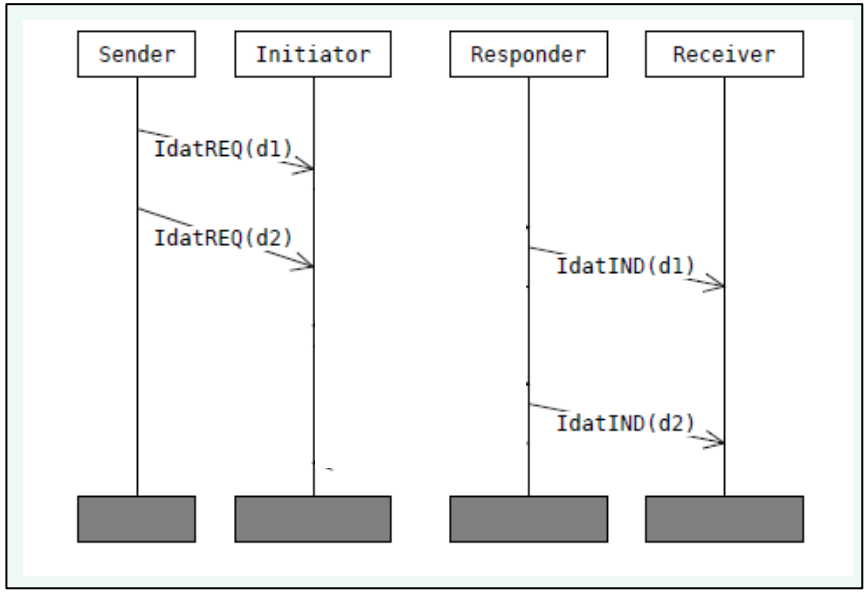
InRes-Dienst: Verbindungsaufbau



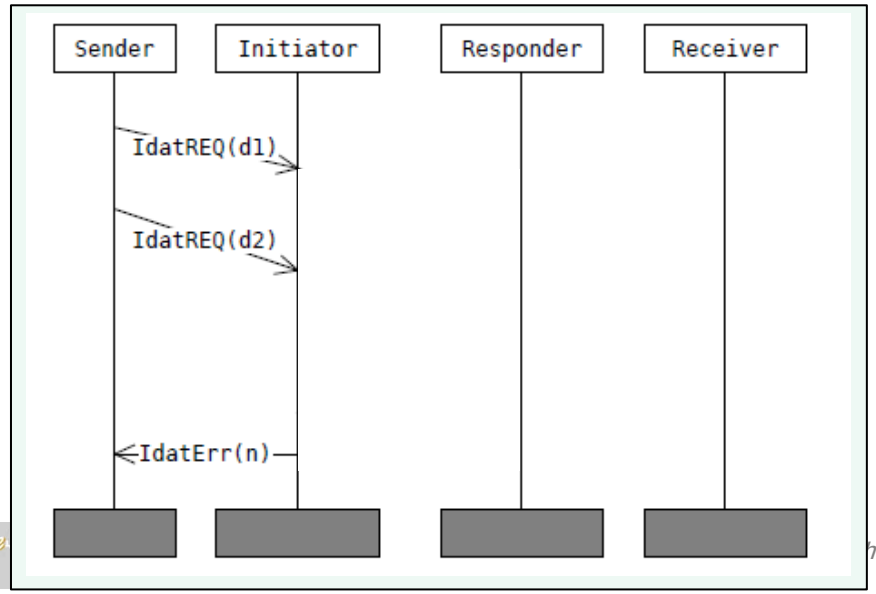
Erfolgloser Verbindungsaufbau
Fehler bei der Übertragung der Ablehnung auf dem Rückweg vom Empfänger

drei verschiedene Ursachen für IconERR

InRes-Dienst: Datenübertragung

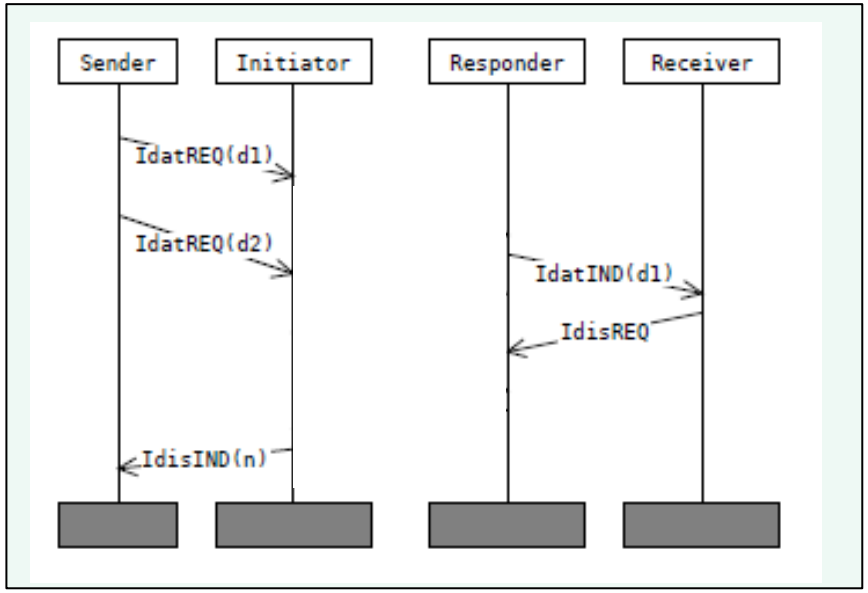


Erfolgreiche Datenübertragung



Erfolglose Datenübertragung
(= Verbindungsabbruch)

InRes-Dienst: Verbindungsabbau



Erfolgreicher Verbindungsabbau

Phasen:

Verbindungsaufbau → Datenübertragung → Verbindungsabbau

nur vom Sender

nur vom Empfänger

8. Protokollentwicklung in SDL

- OSI-Schichtenmodell (Konzept)
- InRes: allgemeine informale Beschreibung
- InRes-Dienstbeschreibung
- InRes-Protokollbeschreibung
- InRes-Semantik (Laufzeit-Betrachtung)
- Umsetzung in SDL/RT
- weiterer Ausbau

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

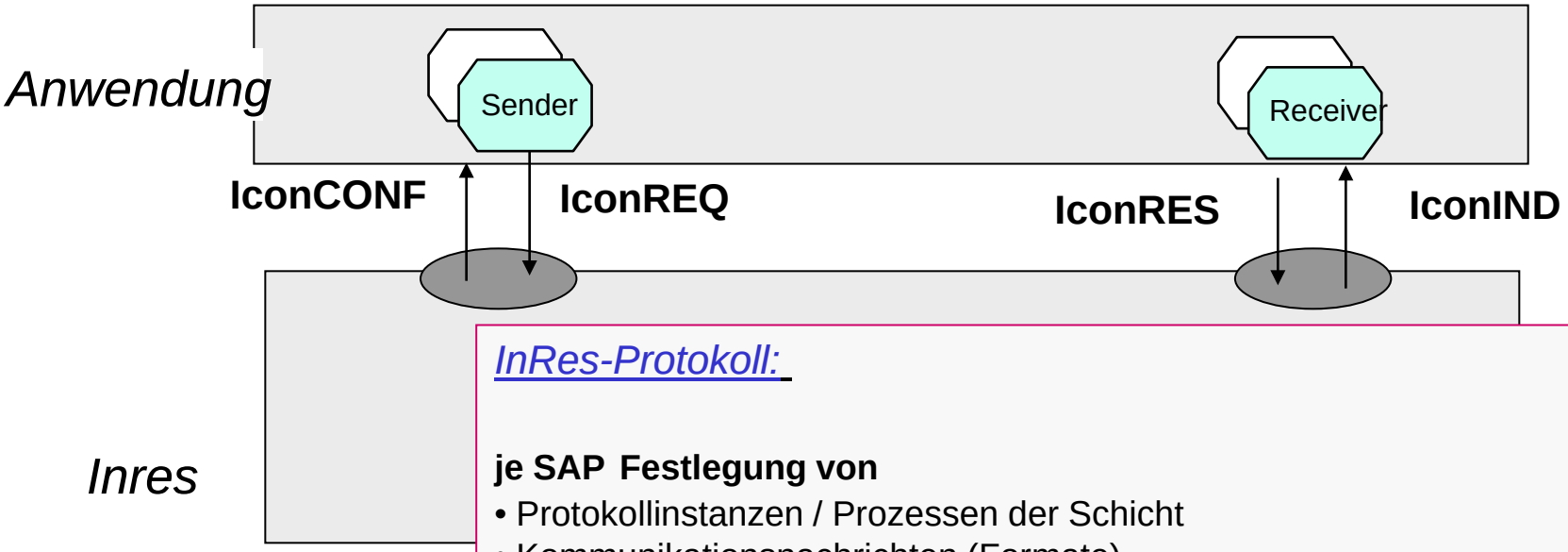
– bei Timeout oder falscher Quittung: sendet Initiator erneut (max. 3mal)

3. Verbindungsabbau

– Auslöser: nur Empfänger oder Initiator (im Fehlerfall)

  praktisch unzureichend:
  Sender kann nicht die Verbindung bei kompletter Datenübertragung beenden

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



InRes-Protokoll:

je SAP Festlegung von

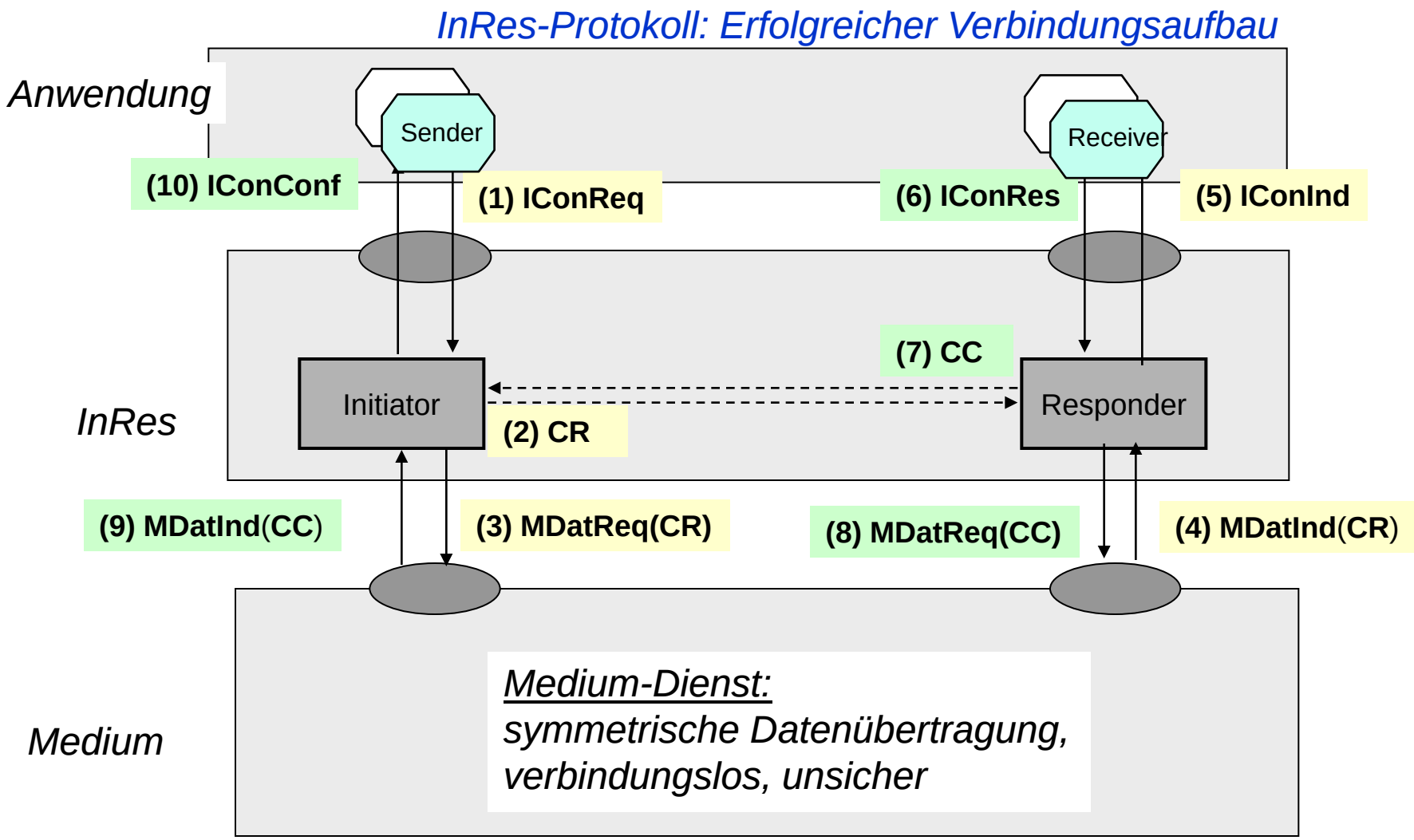
- Protokollinstanzen / Prozessen der Schicht
- Kommunikationsnachrichten (Formate)
- Regeln des Nachrichtenaustausches
  (insb. Quittungsmechanismen, Synchronisationen, Unterbrechungen)

bei

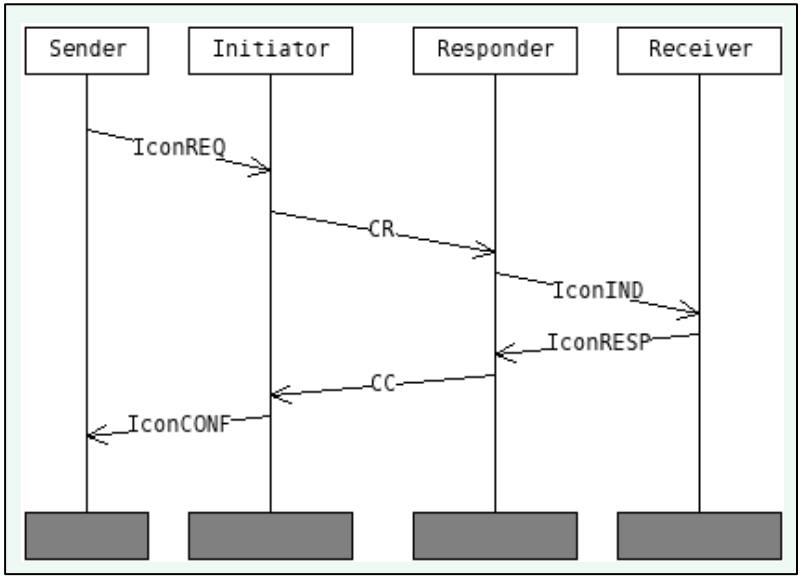
- Einbeziehung primitiverer Dienste der darunter liegenden Schicht zur Datenübertragung
- Codierung/Decodierung von Nachrichten

Protokolleinheiten müssen Einhaltung von folgendem Scenario sichern:
Verbindungsaufbau → Datenübertragung → Verbindungsabbau

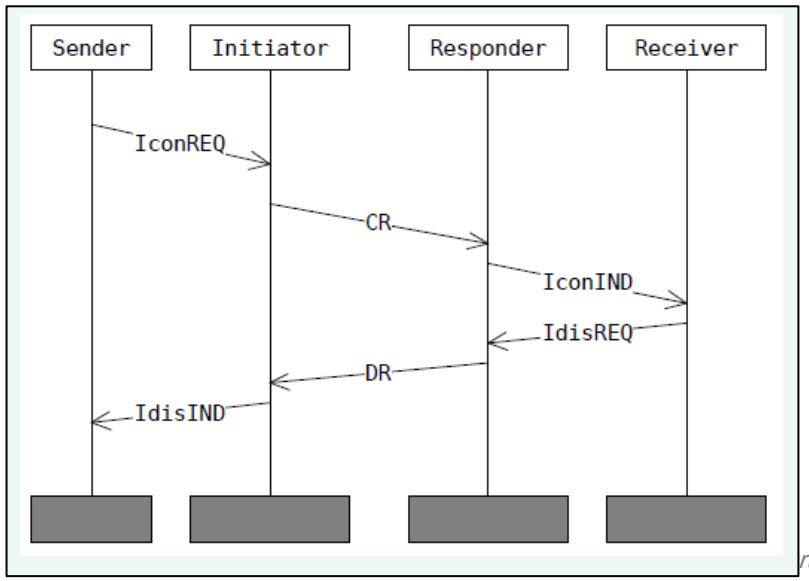
InRes-Protokoll (2): zur Erbringung des InRes-Dienstes



InRes-Protokoll: Verbindungsaufbau

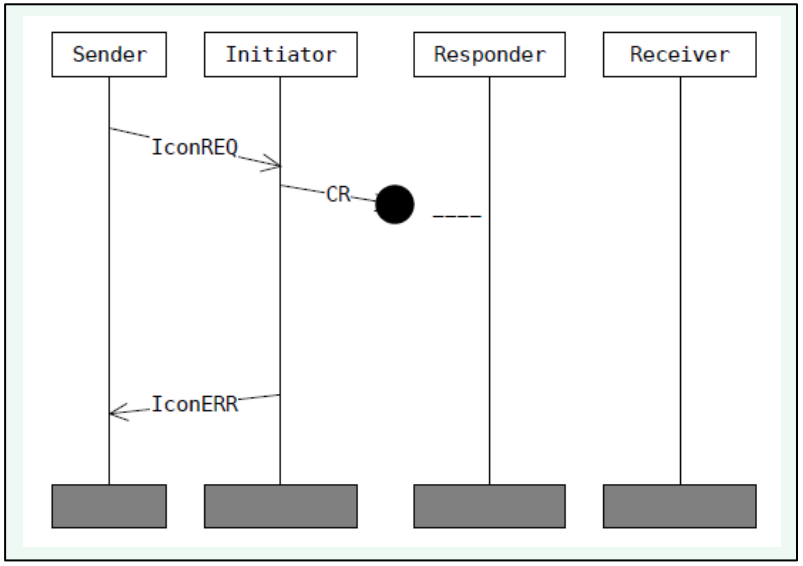


Erfolgreicher Verbindungsaufbau



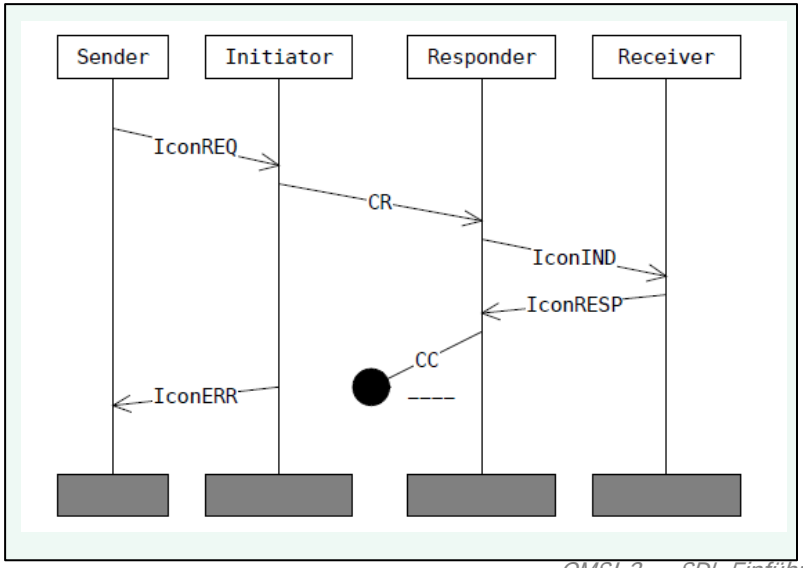
Erfolgloser Verbindungsaufbau
Ablehnung durch Empfänger

InRes-Protokoll: Verbindungsaufbau



Erfolgloser Verbindungsaufbau
Übertragungsfehler
auf dem Weg zum Empfänger

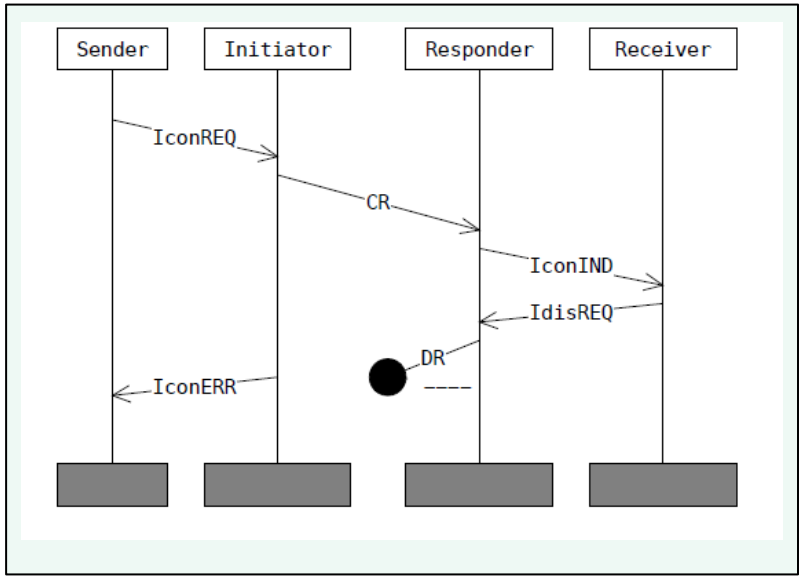
➔ nach 3 Time-out-überwachten Fehlversuchen wird abgebrochen



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

➔ s.o.

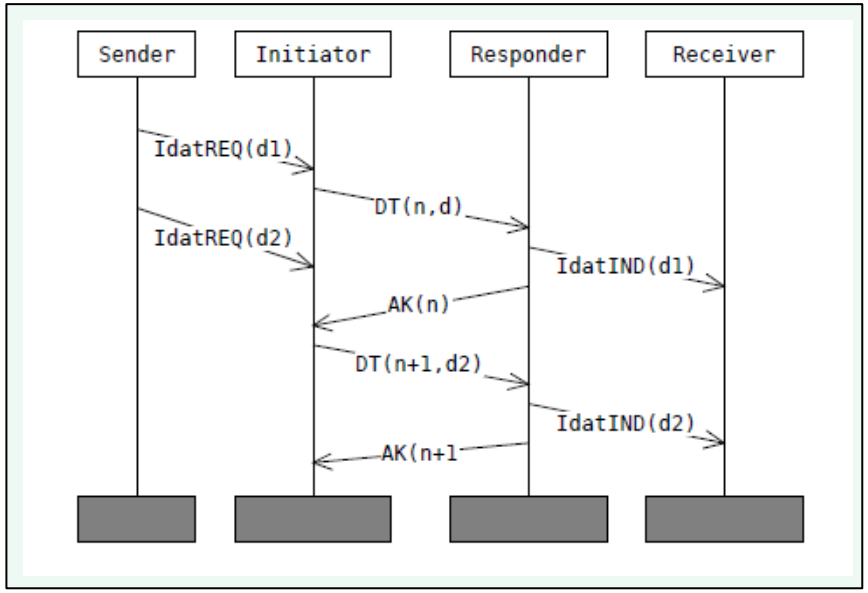
InRes-Protokoll: Verbindungsaufbau



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

➔ nach 3 Time-out-überwachten Fehlversuchen wird abgebrochen

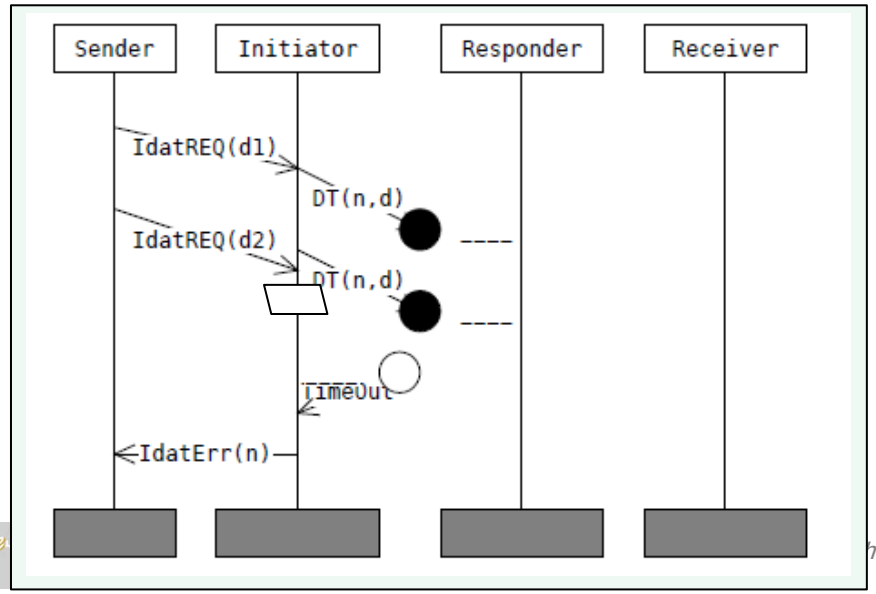
InRes-Dienst: Datenübertragung



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

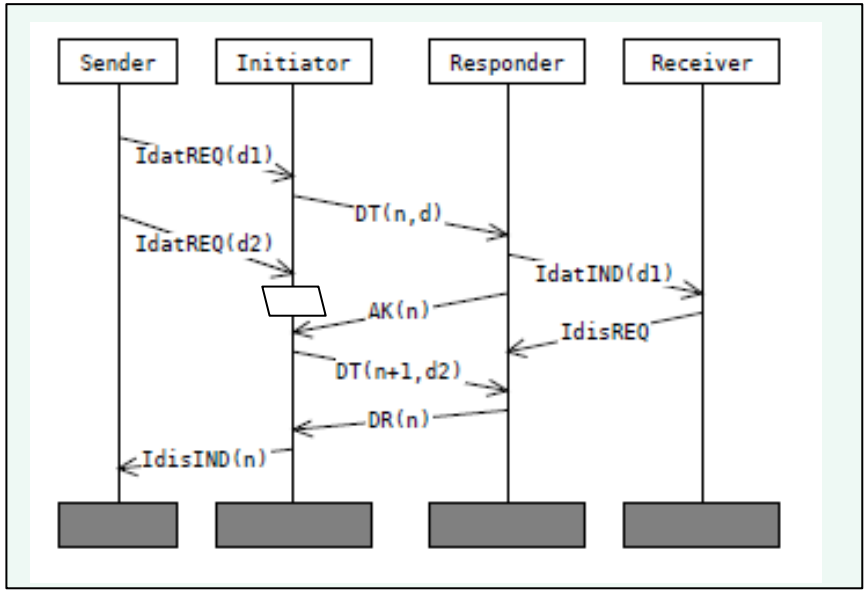


Erfolglose Datenübertragung
(= Verbindungsabbruch)

➔nach 3 Time-out-überwachten Fehlversuchen wird abgebrochen

➔ Sender erhält Information zur Anzahl erfolgreicher Datenübertragungen

InRes-Dienst: Verbindungsabbau

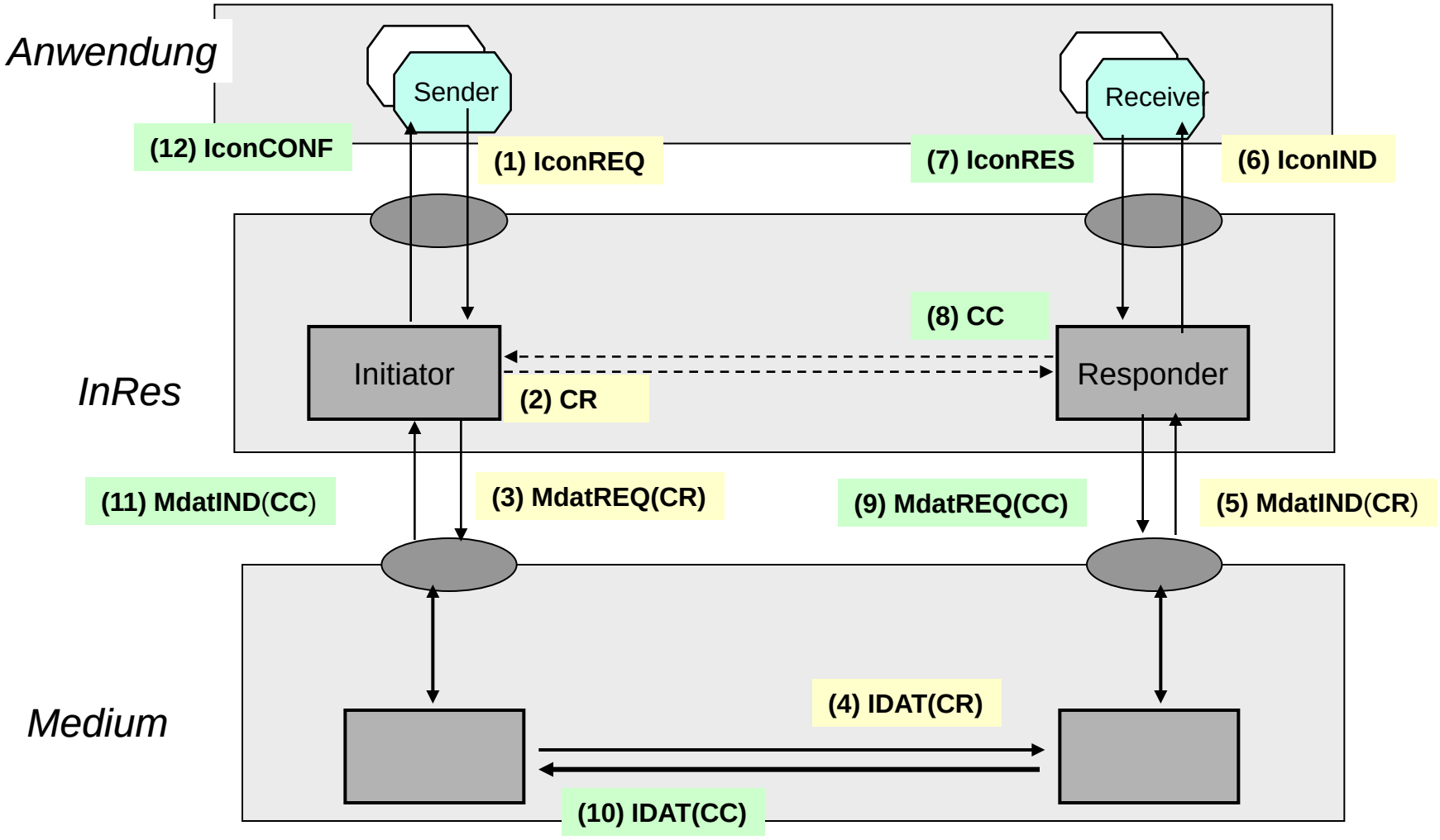


Erfolgreicher Verbindungsabbau

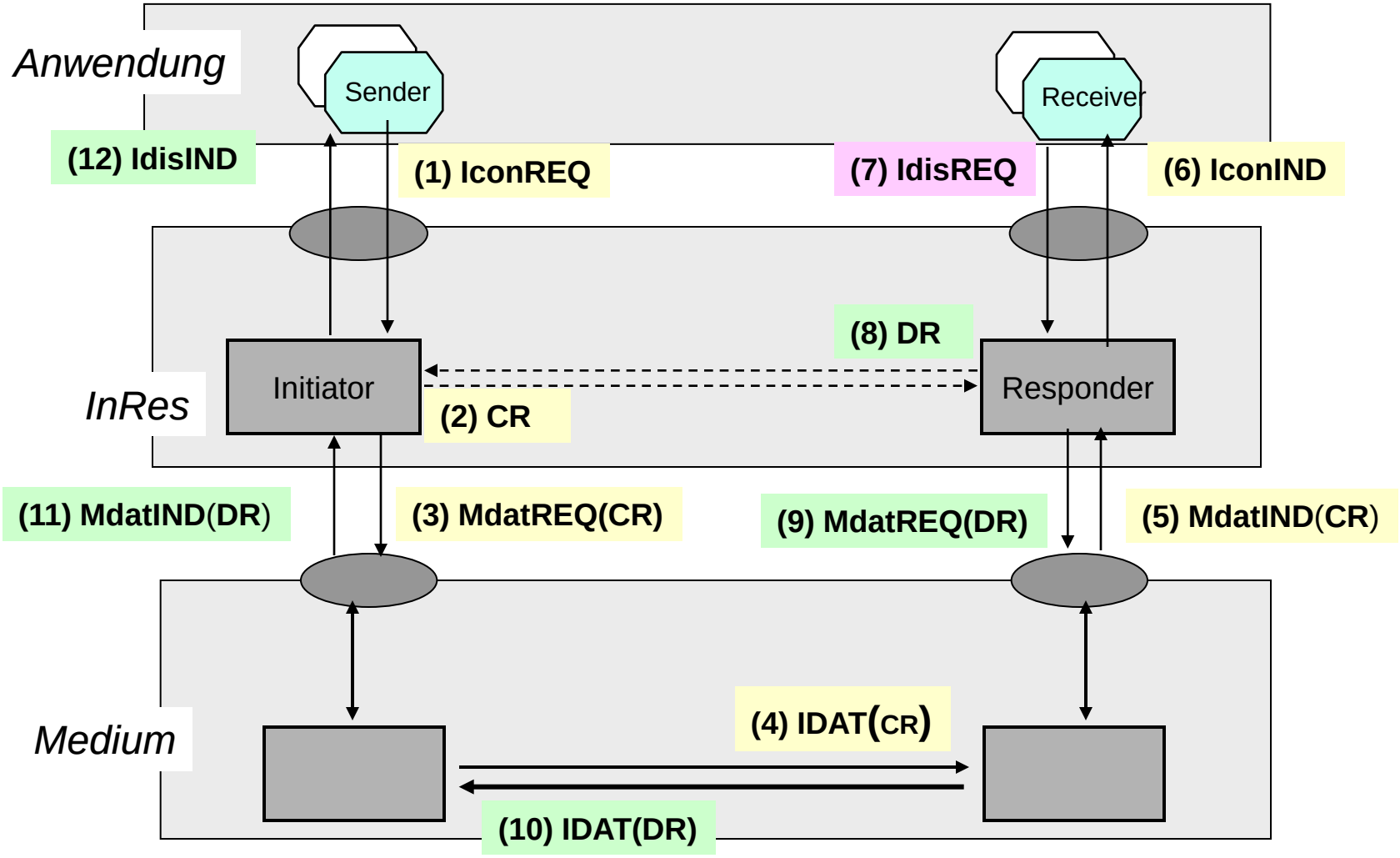
8. Protokollentwicklung in SDL

- OSI-Schichtenmodell (Konzept)
- InRes: allgemeine informale Beschreibung
- InRes-Dienstbeschreibung
- InRes-Protokollbeschreibung
- InRes-Semantik (Laufzeit-Betrachtung)
- Umsetzung in SDL/RT
- weiterer Ausbau

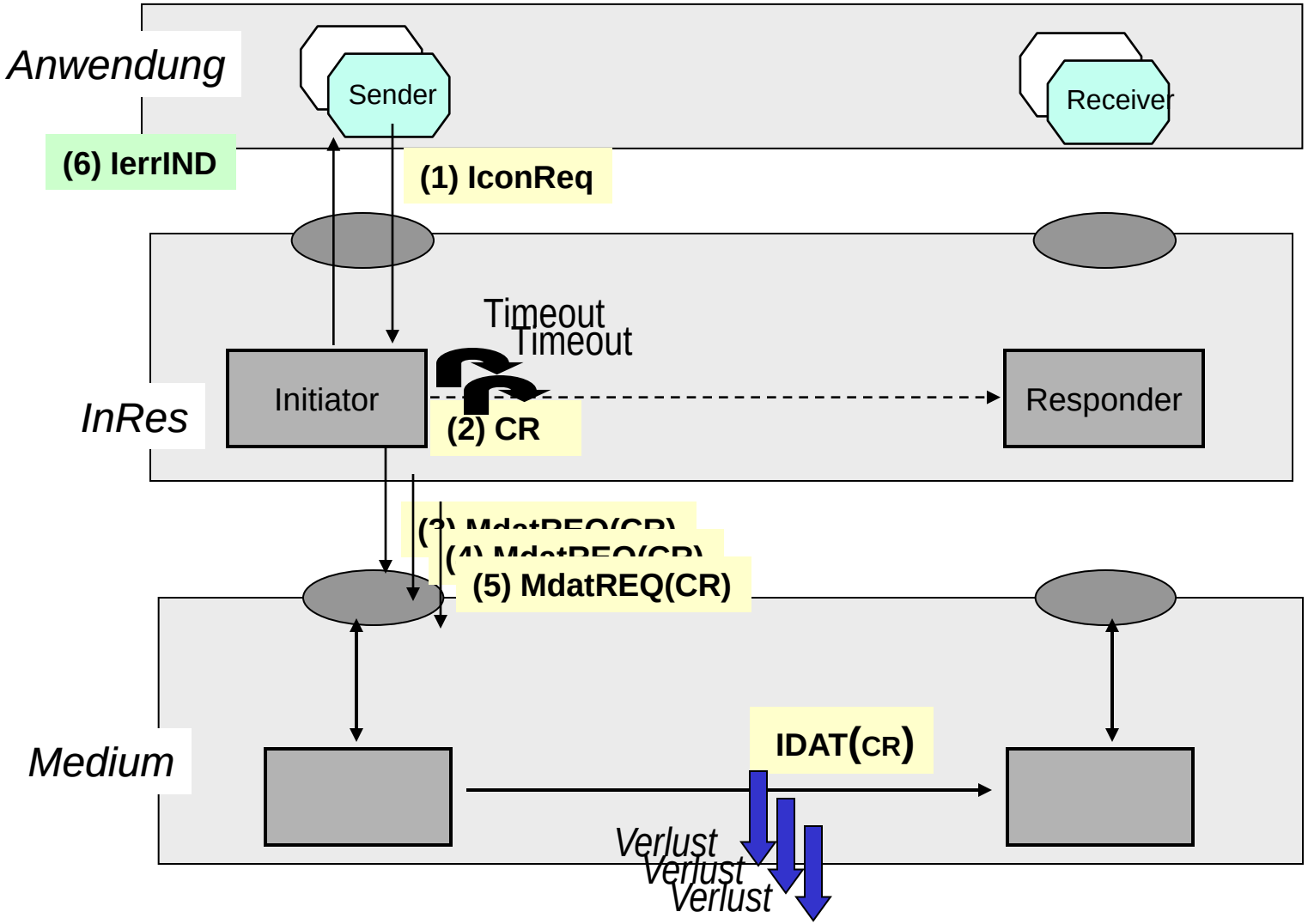
Erfolgreicher Verbindungsaufbau



Abgelehnter Verbindungsaufbau



Abgebrochener Verbindungsaufbau



8. Protokollentwicklung in SDL

- OSI-Schichtenmodell (Konzept)
- InRes: allgemeine informale Beschreibung
- InRes-Dienstbeschreibung
- InRes-Protokollbeschreibung
- InRes-Semantik (Laufzeit-Betrachtung)
- Umsetzung in SDL/RT
- weiterer Ausbau

RTDS- InRes-Projektfile: System-Modell

System: Netzkonfiguration (zwei Knoten)

```
USE Lib;
```

```
#include "RandDraw.h"
```

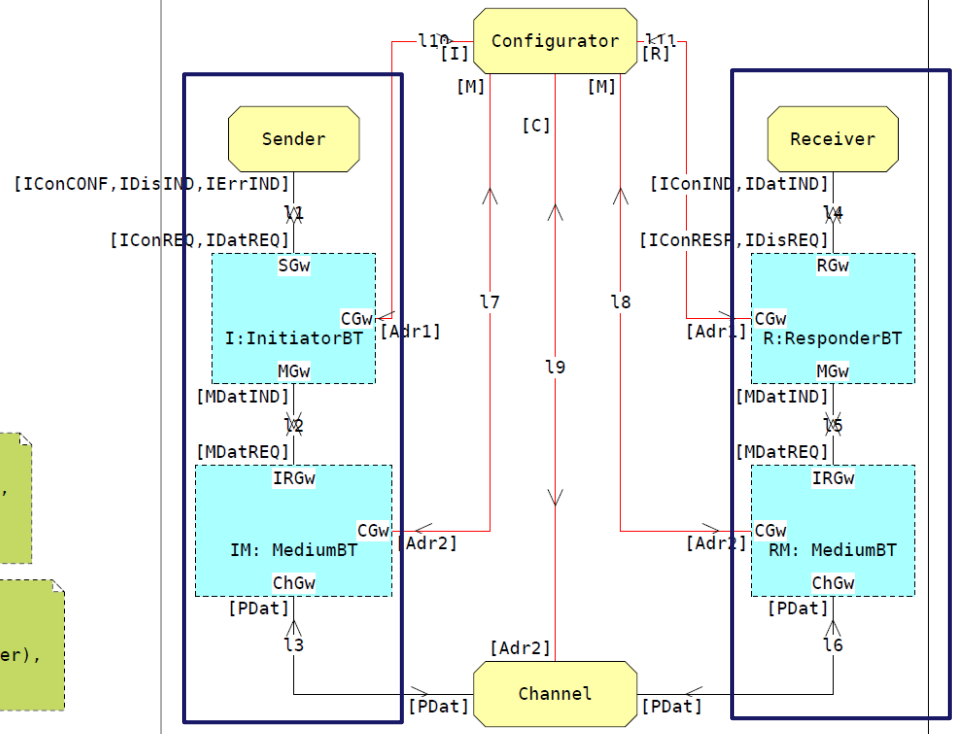
```
MESSAGE I,
        R,
        M,
        C,
        Adr1(RTDS_QueueId),
        Adr2(RTDS_QueueId,RTDS_QueueId);
```

```
MESSAGE IConREQ, IConIND, IConRESP, IConCONF, IErrIND,
        IDatREQ(char), IDatIND(char),
        IDisREQ, IDisIND;
```

```
MESSAGE CR, CC,
        DT(SequenceNumber,ISduType), ACK(SequenceNumber),
        DR;
```

```
MESSAGE MDatREQ(MDatType),
        MDatIND(MDatType);
```

```
MESSAGE PDat(MDatType);
```

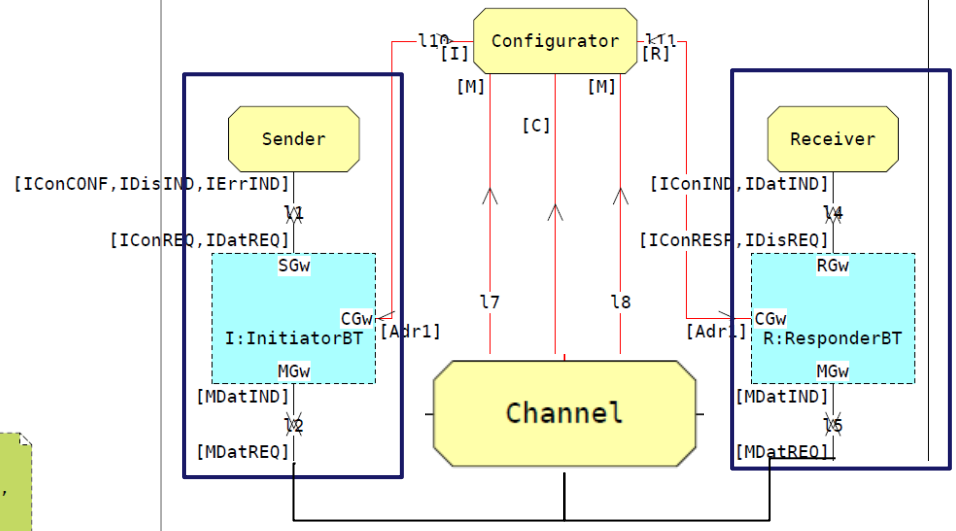


physischer Kanal

System: Netzkonfiguration (virtueller Kanal)

```
USE Lib;

#include "RandDraw.h"

MESSAGE I,
        R,
        M,
        C,
        Adr1(RTDS_QueueId),
        Adr2(RTDS_QueueId,RTDS_QueueId);

MESSAGE IConREQ, IConIND, IConRESP, IConCONF, IErrIND,
        IDatREQ(char), IDatIND(char),
        IDisREQ, IDisIND;

MESSAGE CR, CC,
        DT(SequenceNumber,ISduType), ACK(SequenceNumber),
        DR;

MESSAGE MDatREQ(MDatType),
        MDatIND(MDatType);

MESSAGE PDat(MDatType);
```

...Entwicklung\Inres\NetConfuguration.rdd - Part. 0 Page 1-1



virtueller Kanal
(Kurzschluss auf InRes-Schicht-Ebene)

...Entwicklung\Inres\NetConfuguration.rdd - Part. 0 Page 1-2

System: Netzkonfiguration (virtueller Kanal)

```
USE Lib;
```

```
#include "RandDraw.h"
```

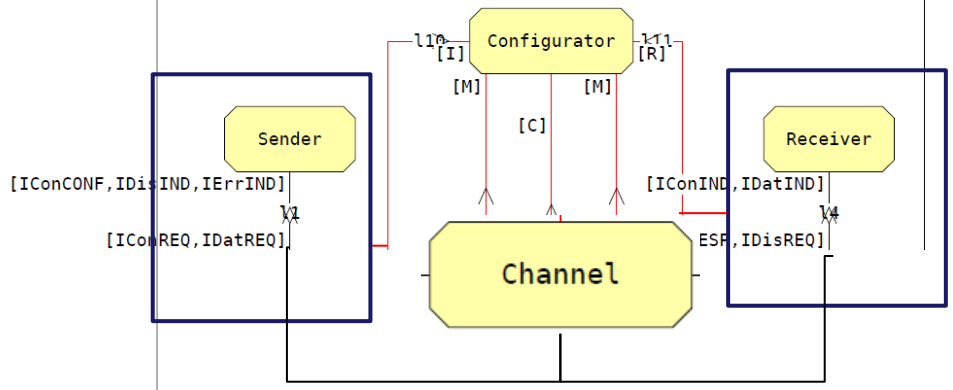
```
MESSAGE I,
        R,
        M,
        C,
        Adr1(RTDS_QueueId),
        Adr2(RTDS_QueueId,RTDS_QueueId);
```

```
MESSAGE IConREQ, IConIND, IConRESP, IConCONF, IErrIND,
        IDatREQ(char), IDatIND(char),
        IDisREQ, IDisIND;
```

```
MESSAGE CR, CC,
        DT(SequenceNumber,ISduType), ACK(SequenceNumber),
        DR;
```

```
MESSAGE MDatREQ(MDatType),
        MDatIND(MDatType);
```

```
MESSAGE PDat(MDatType);
```



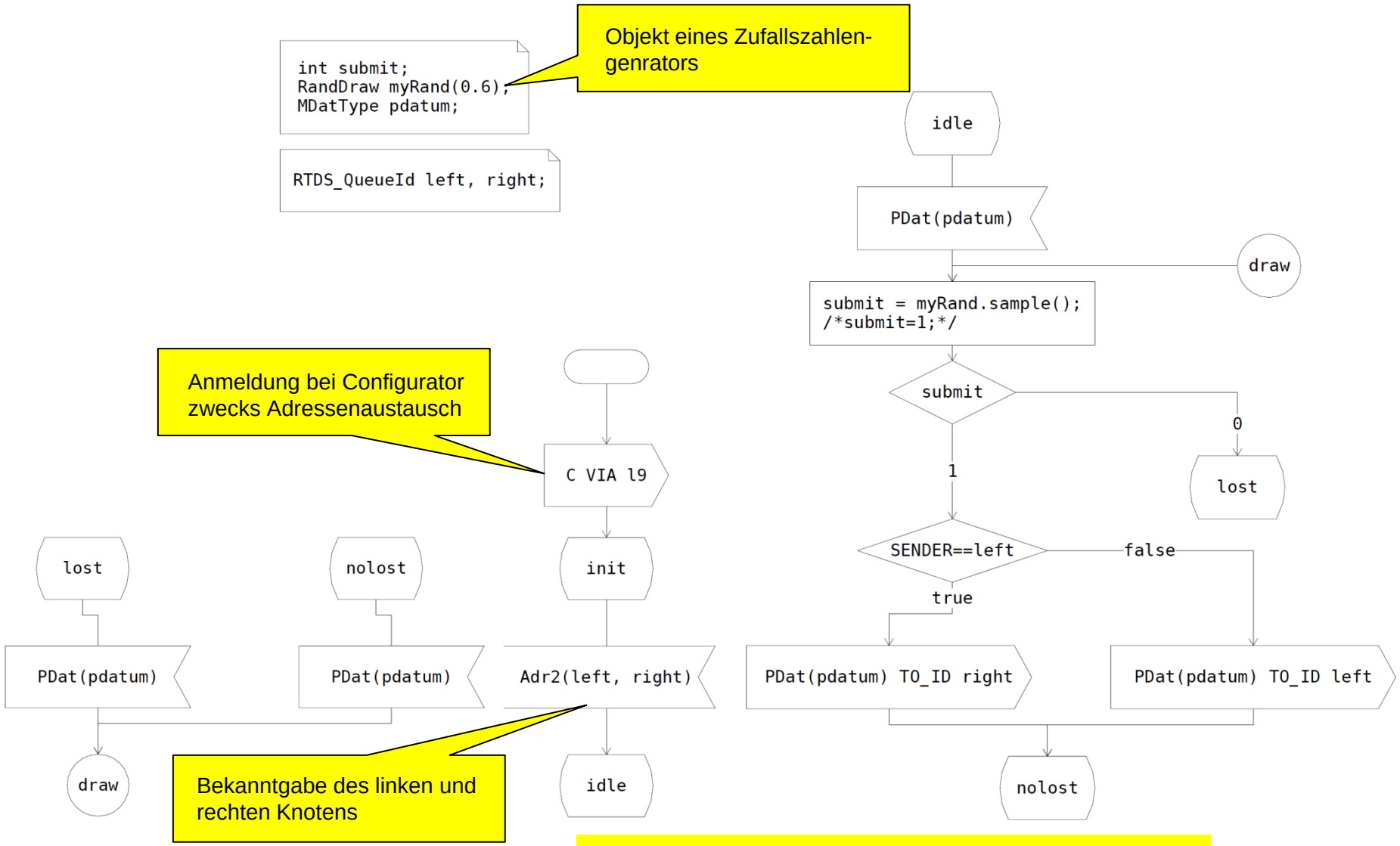
(Kurzschluss auf Anwendungsschicht-Ebene)

so kann Schicht für Schicht getestet werden

...Entwicklung\Inres\NetConfuguration.rdd - Part. 0 Page 1-1

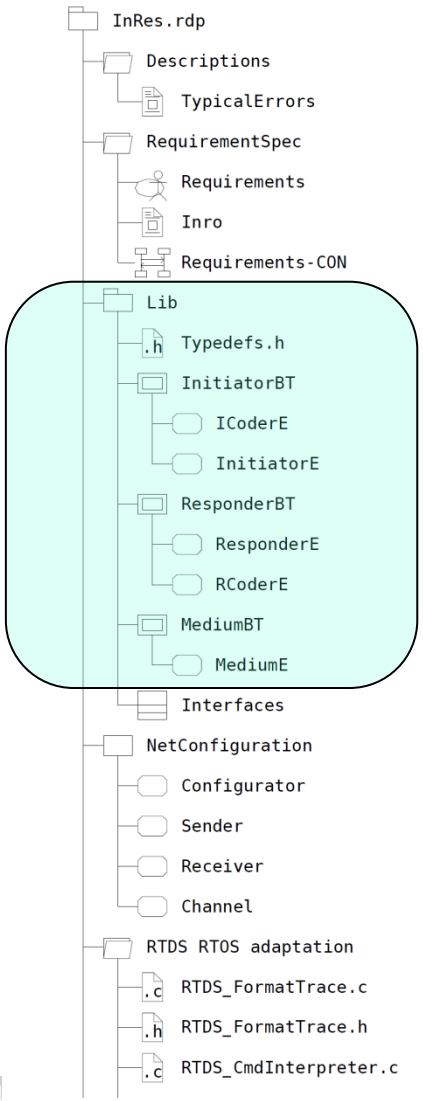
...Entwicklung\Inres\NetConfuguration.rdd - Part. 0 Page 1-2

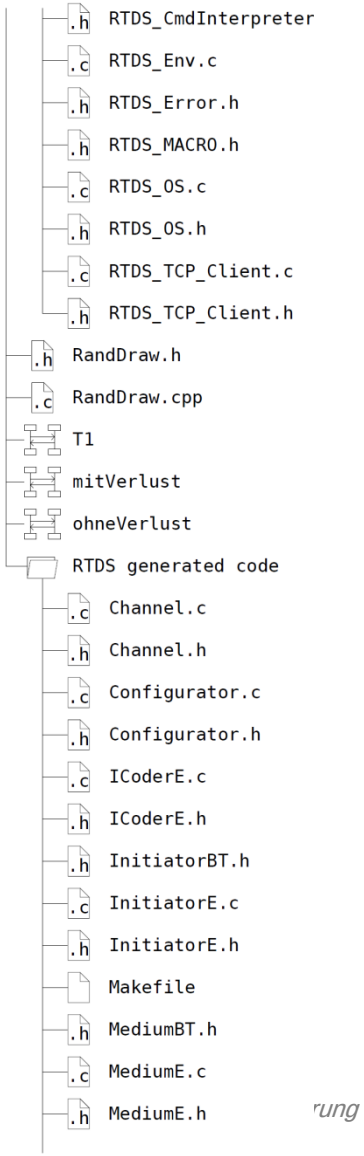
Übertragungskanal mit Verlust

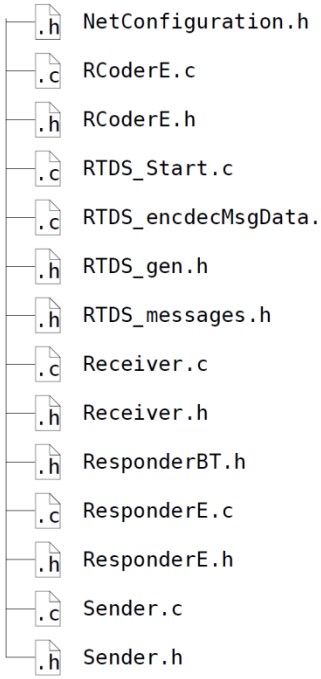


Wie Verdopplung und Vertauschung?

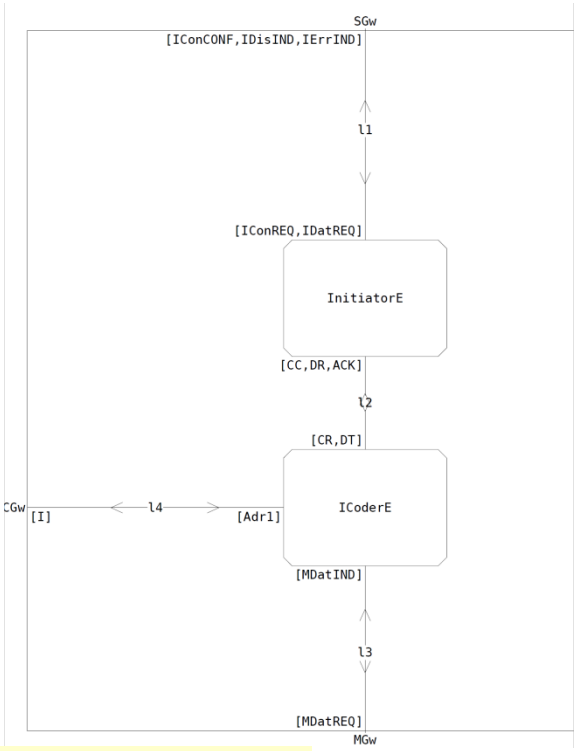
RTDS- InRes-Projektfile: Blocktypen

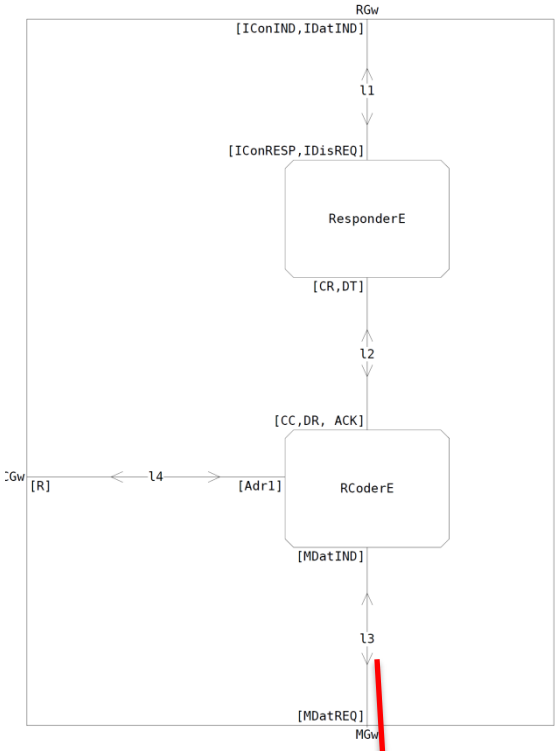




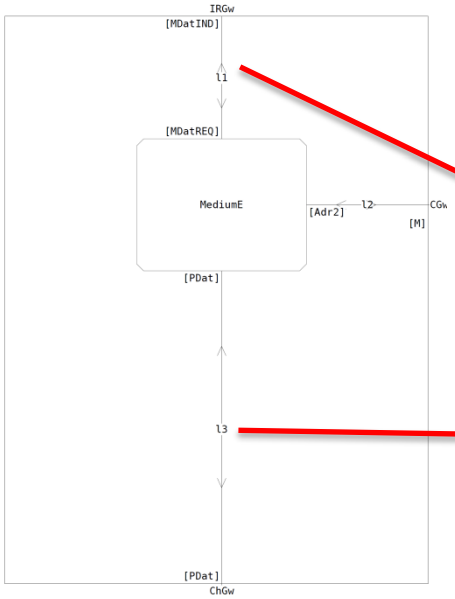


Protokolleinheiten der
- InRes-Schicht
  InitiatorBT, ResponderBT
- Medium-Schicht
  MediumBT





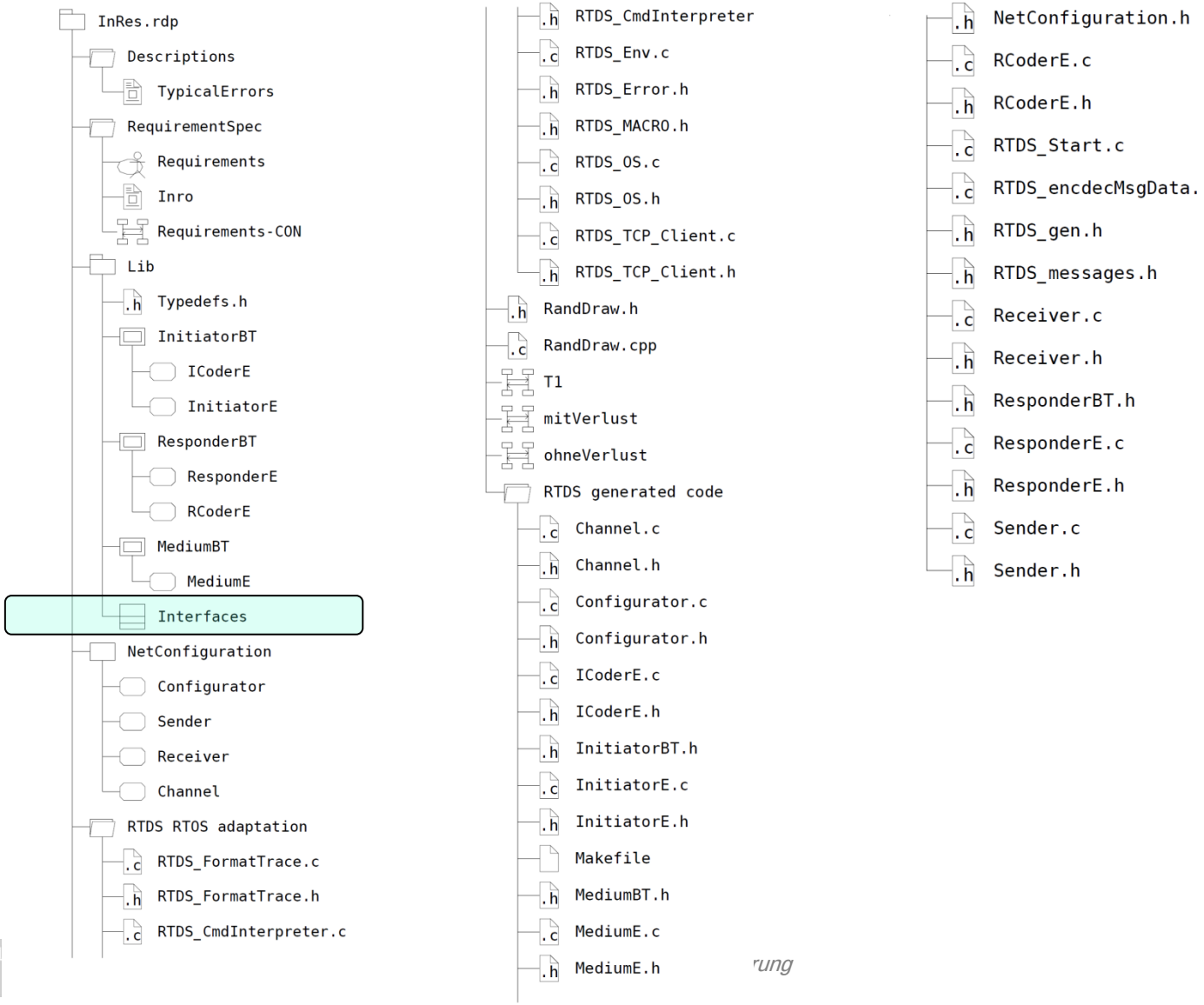

```
#ifndef TYPEDEFS_H_
#define TYPEDEFS_H_

typedef  char  ISduType;
typedef  int SequenceNumber;

typedef  enum
{eCR, eCC, eDR, eDT, eACK}  IPduType;

typedef  struct  MDatType {
    IPDuType id;
    SequenceNumber num;
    ISduType data ;
} MDatType;
#endif /*TYPEDEFS_H_*/
```



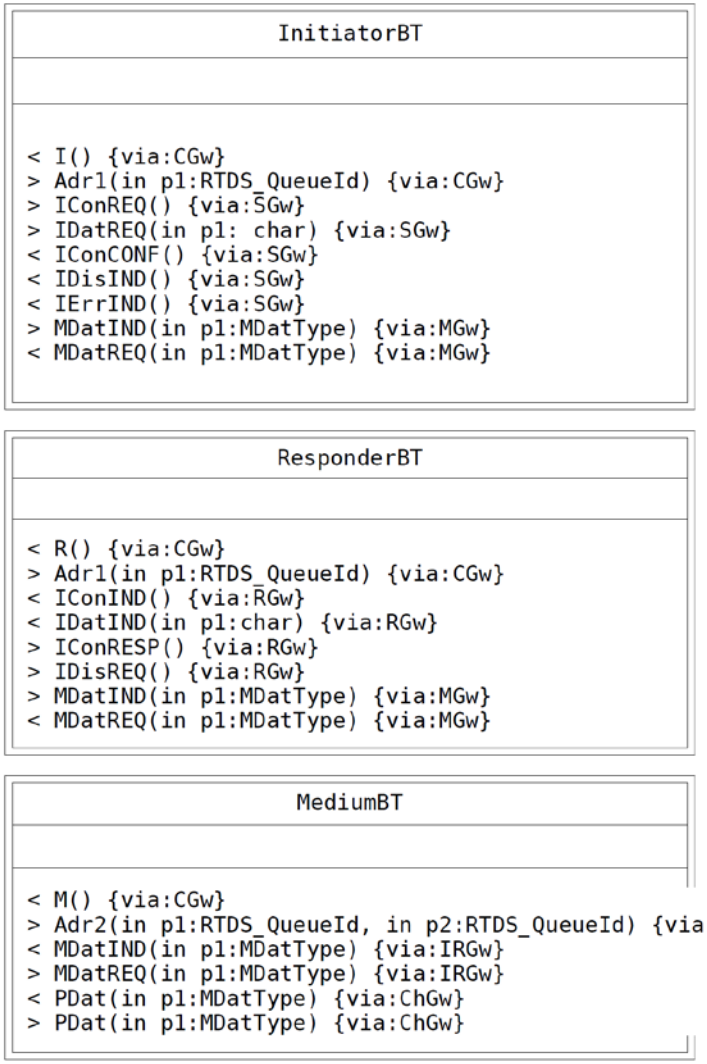

```
MESSAGE MDatREQ(MDatType),
        MDatIND(MDatType);
```

```
MESSAGE PDat(MDatType);
```

RTDS- InRes-Projektdatei: UML-Interface



Interfaces von Blocktypen



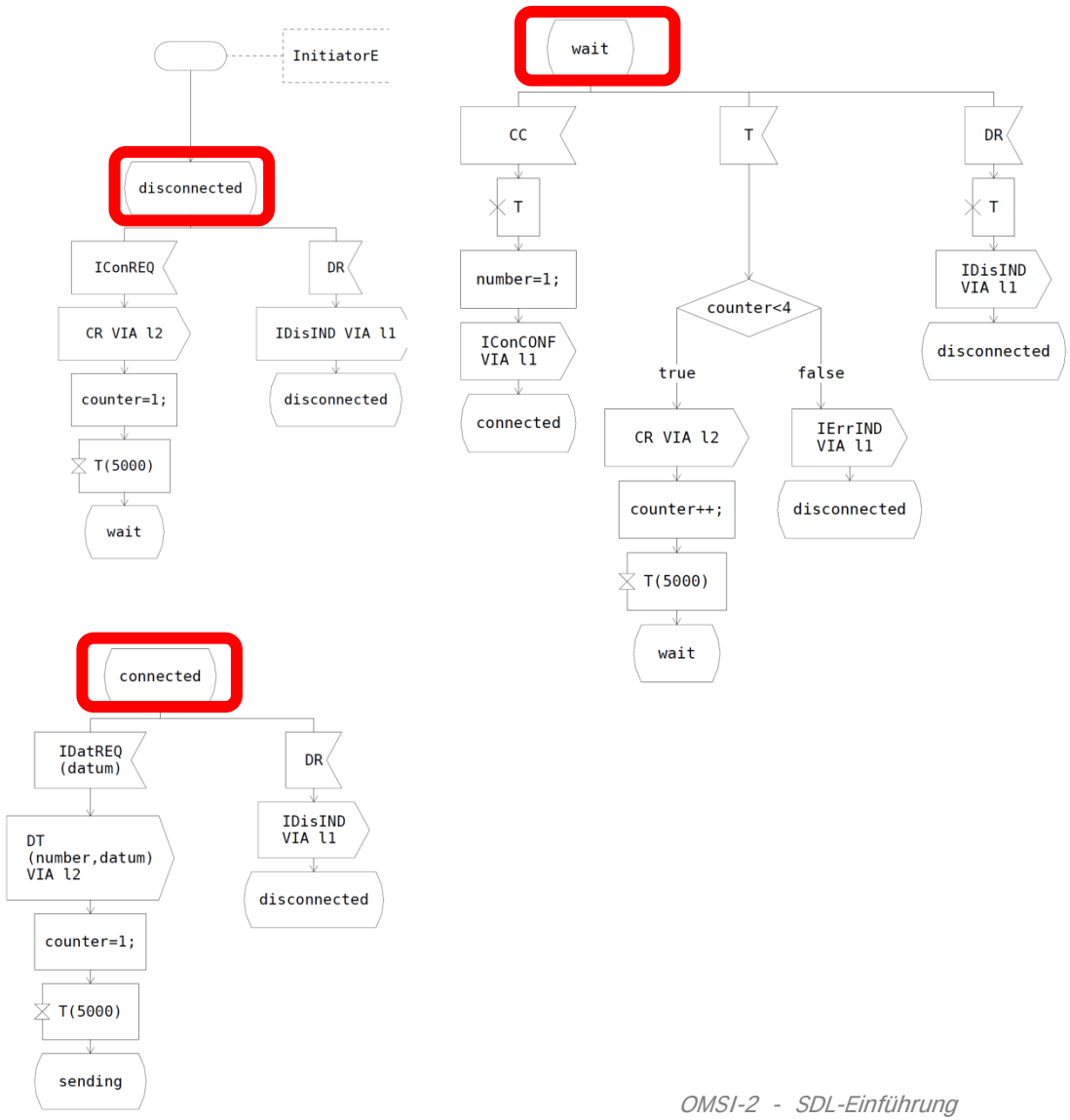
Protokolleinheiten wurden nicht mit expliziter Typangabe definiert

deshalb:

- entfällt deren Interface-Spec;
  sie sind ja bereits in einem (und nur einem) Kontext eingebettet

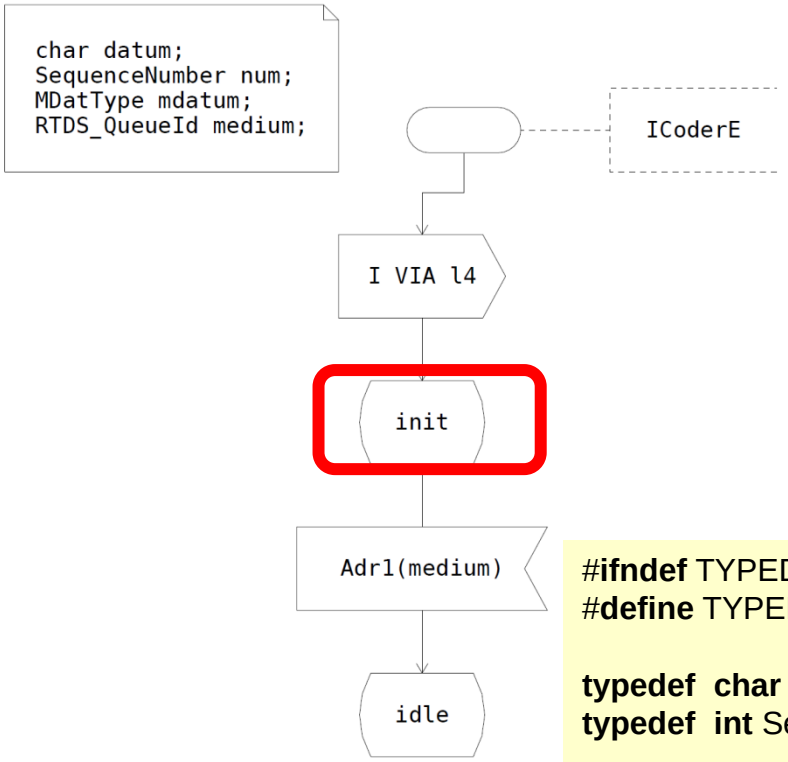
```
int counter=0;

char datum;
SequenceNumber num, number;
```

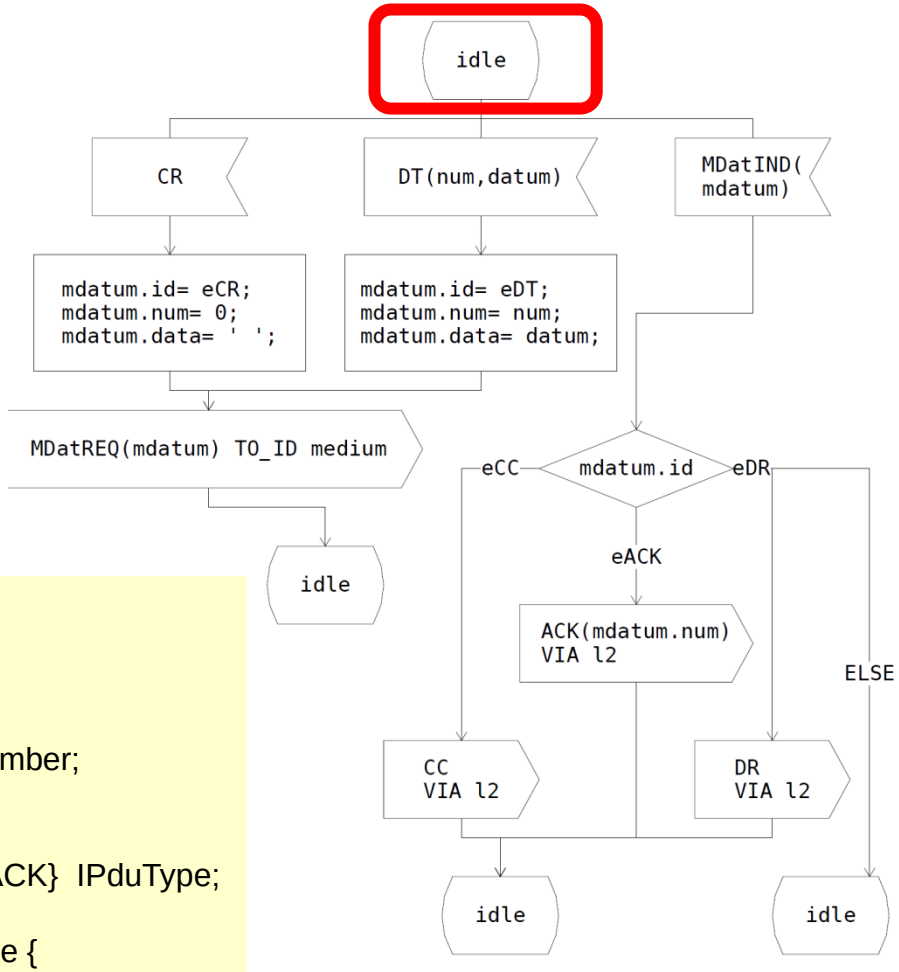
Initiator-Process



implizite (lokale) Definition von t als Timer

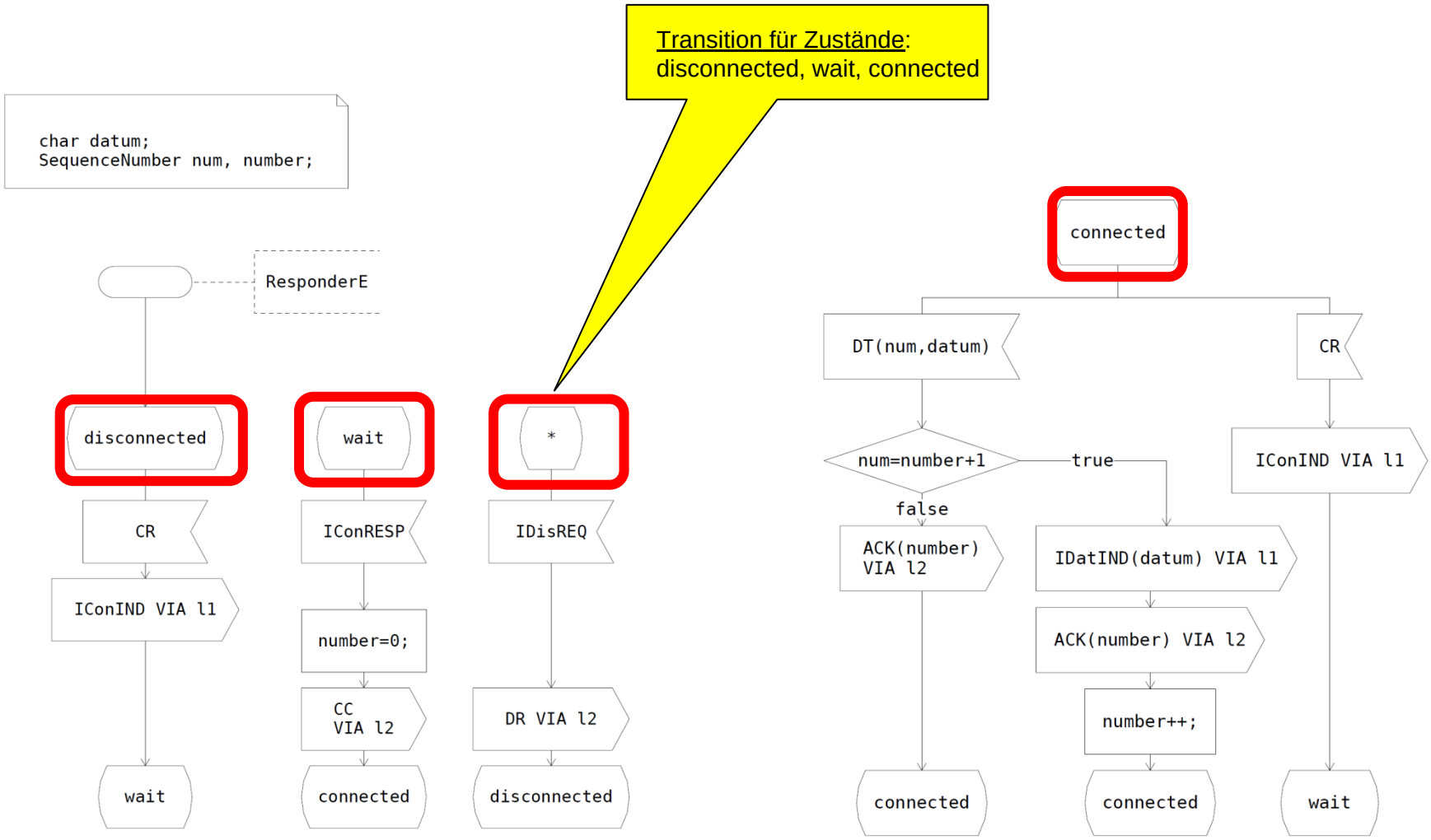
Codierer: (Initiator-seitig)



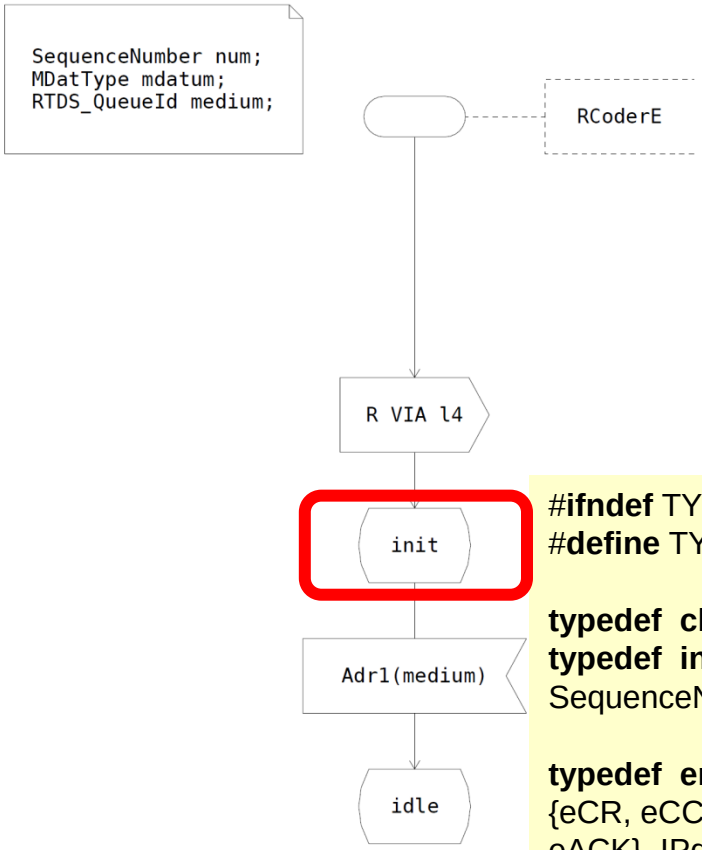



```
#ifndef TYPEDEFS_H_
#define TYPEDEFS_H_

typedef  char  ISduType;
typedef  int SequenceNumber;

typedef  enum
{eCR, eCC, eDR, eDT, eACK}  IPduType;

typedef  struct  MDatType {
    IPduType id;
    SequenceNumber num;
    ISduType data ;
} MDatType;
#endif /*TYPEDEFS_H_*/
```

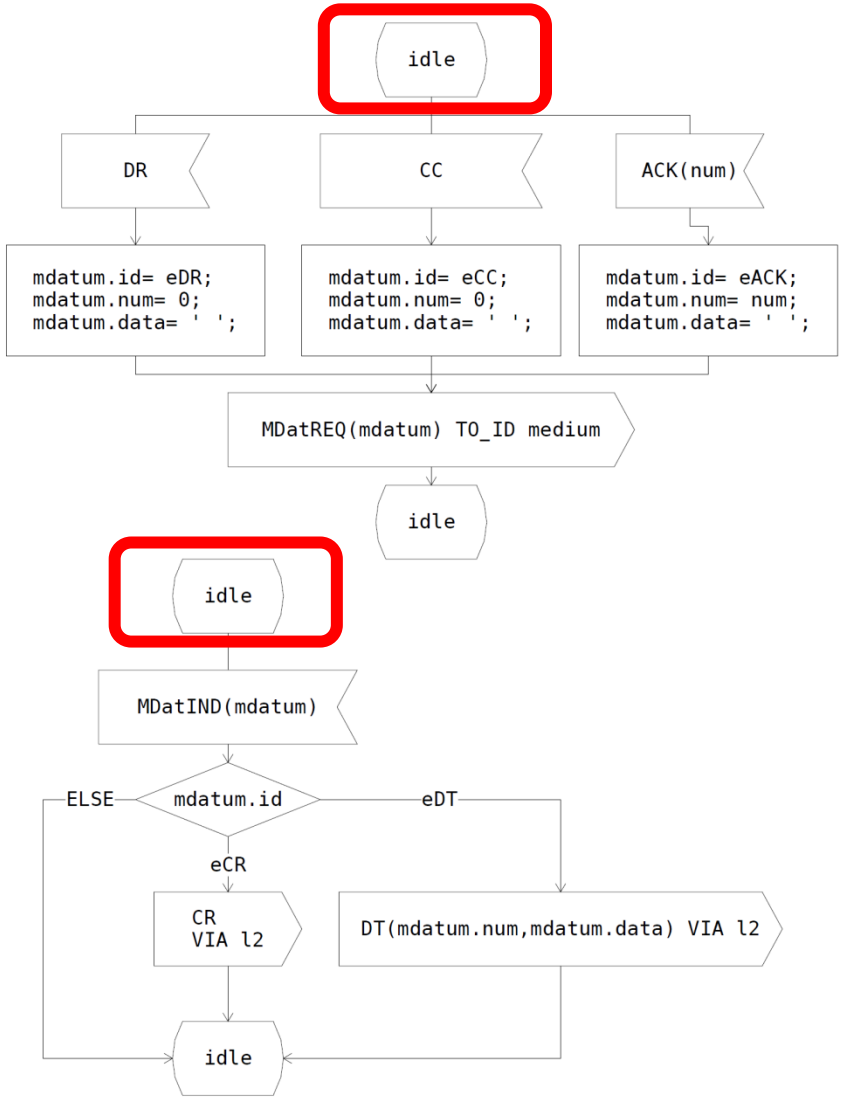
Responder-Process



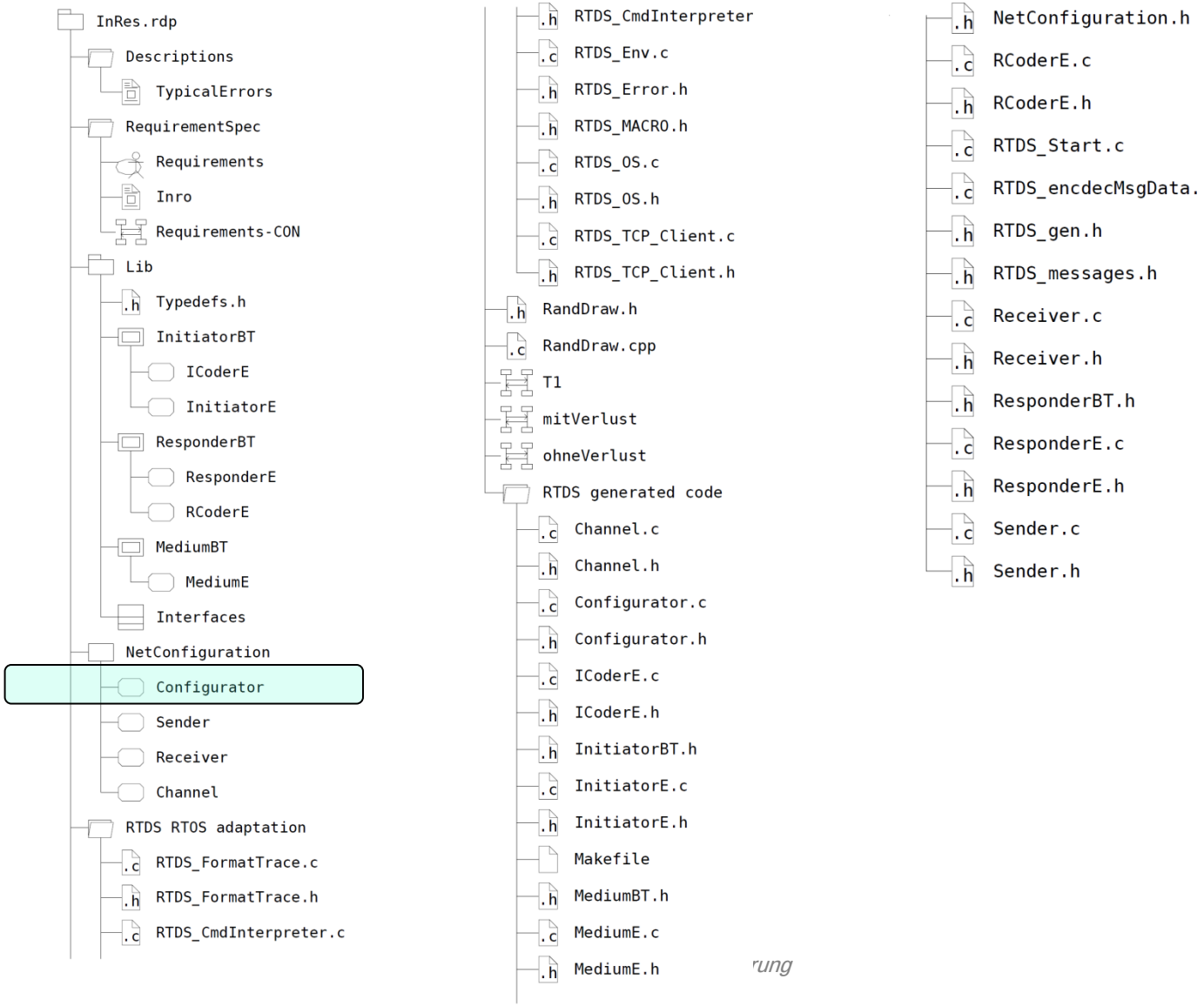
Codierer: (Responder-seitig)



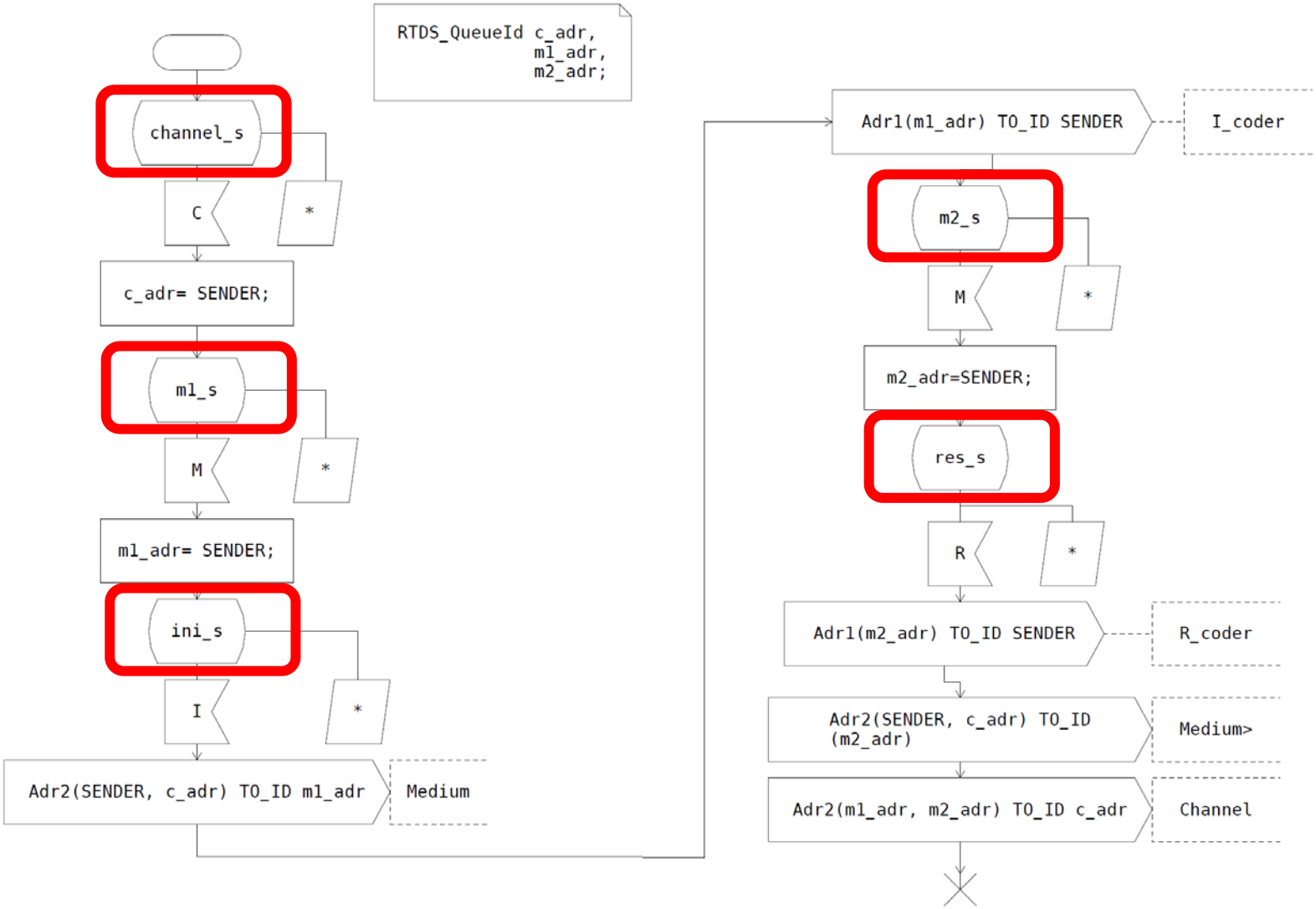

```
#ifndef TYPEDEFS_H_
#define TYPEDEFS_H_

typedef  char  ISduType;
typedef  int
SequenceNumber;

typedef  enum
{eCR, eCC, eDR, eDT,
eACK}  IPduType;

typedef  struct  MDatType {
    int id;
    SequenceNumber num;
    ISduType data ;
} MDatType;
#endif /*TYPEDEFS_H_*/
```



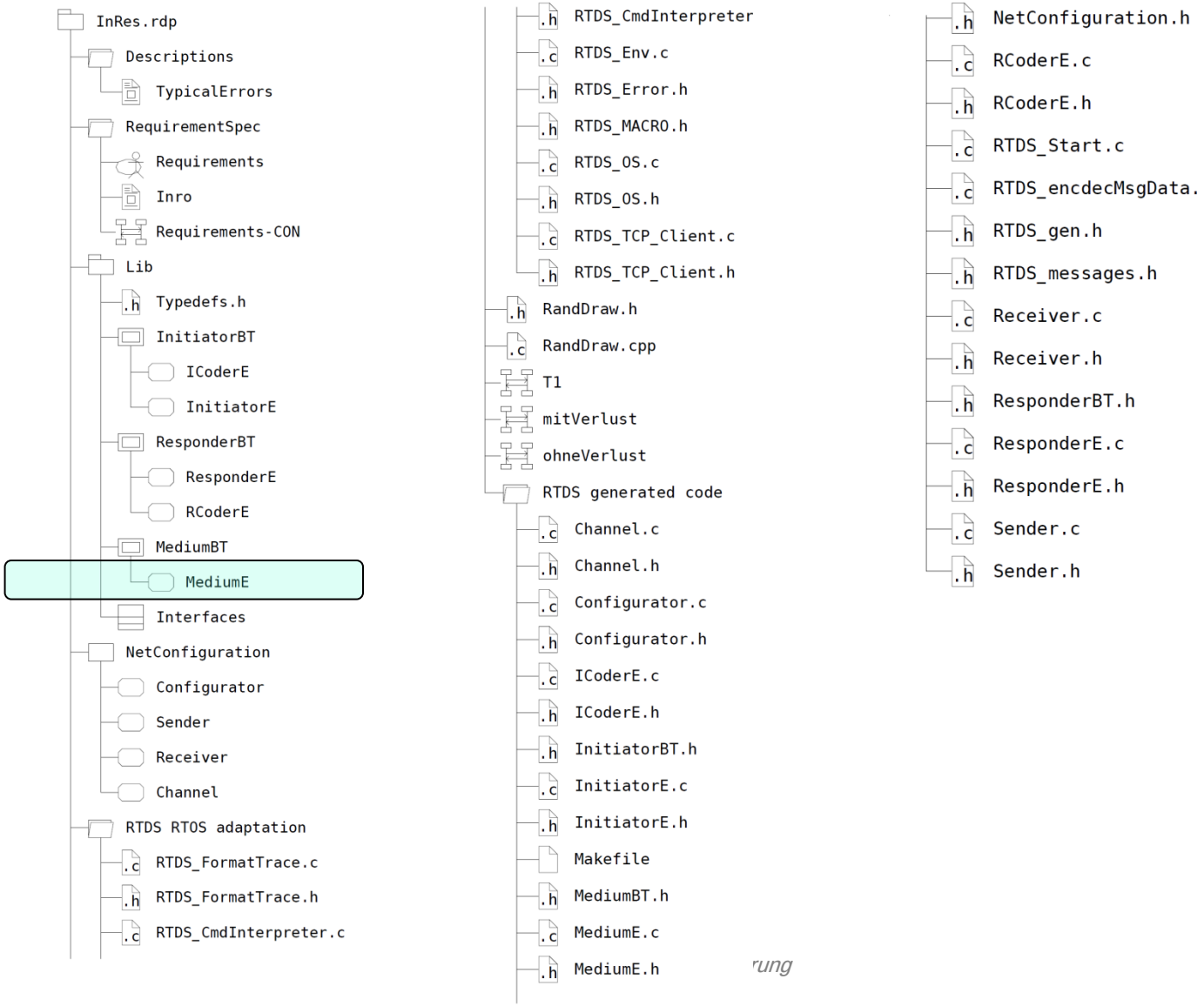
RTDS- InRes-Projektdatei: Configurator



Systemkonfigurator

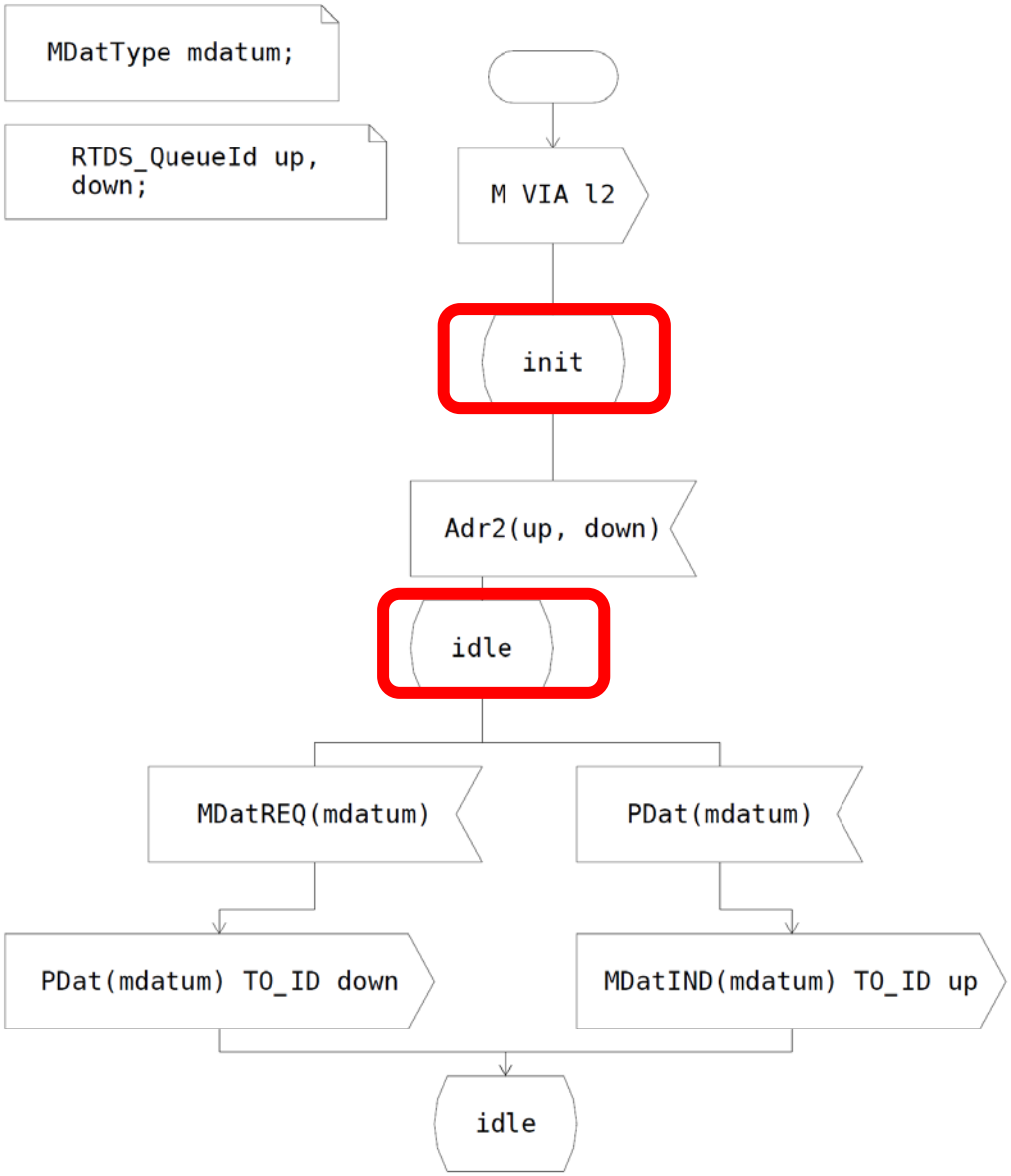


RTDS- InRes-Projektdatei: Medium-Protokolleinheit

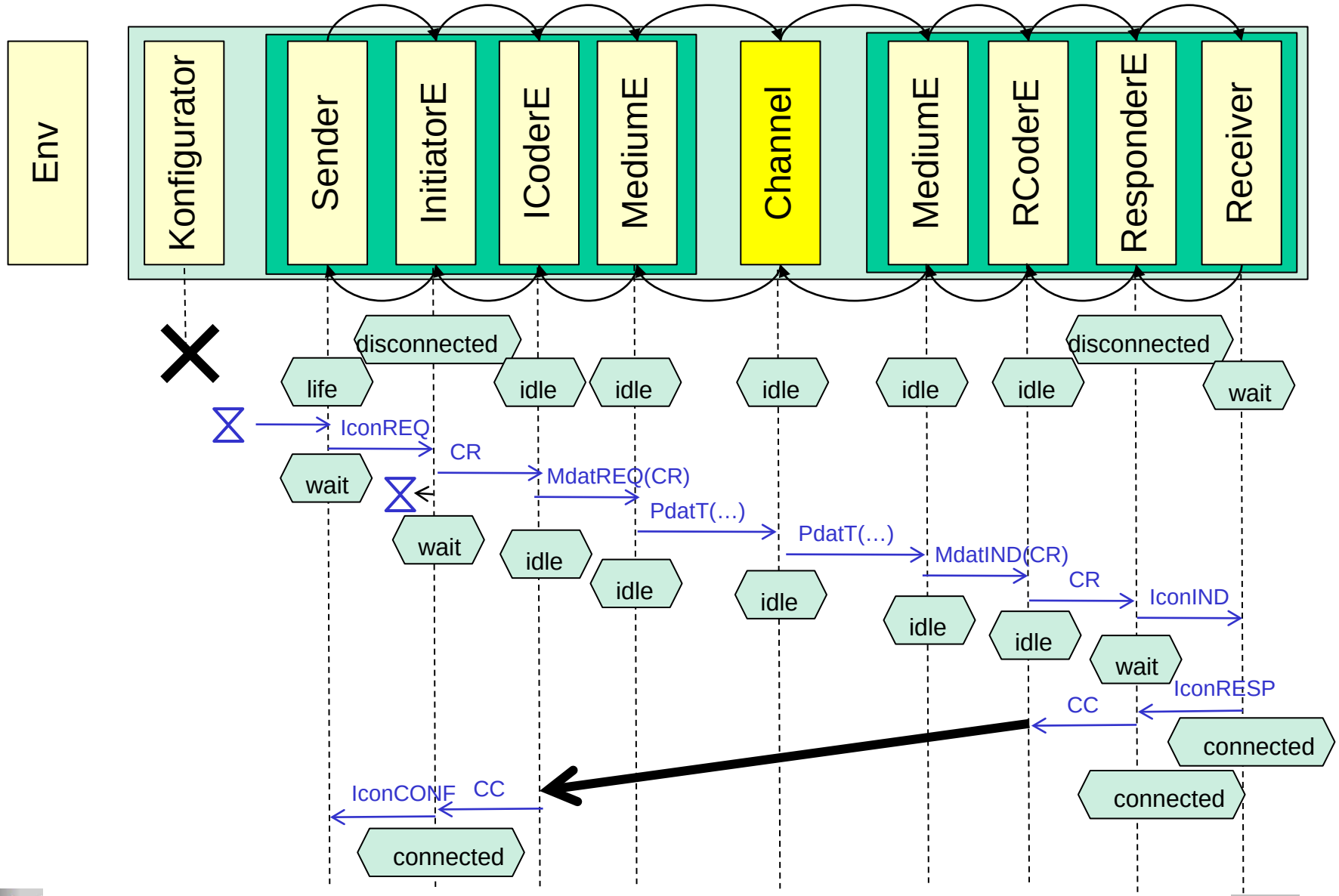


Medium-Entity: (Initiator- und Responder-seitig)

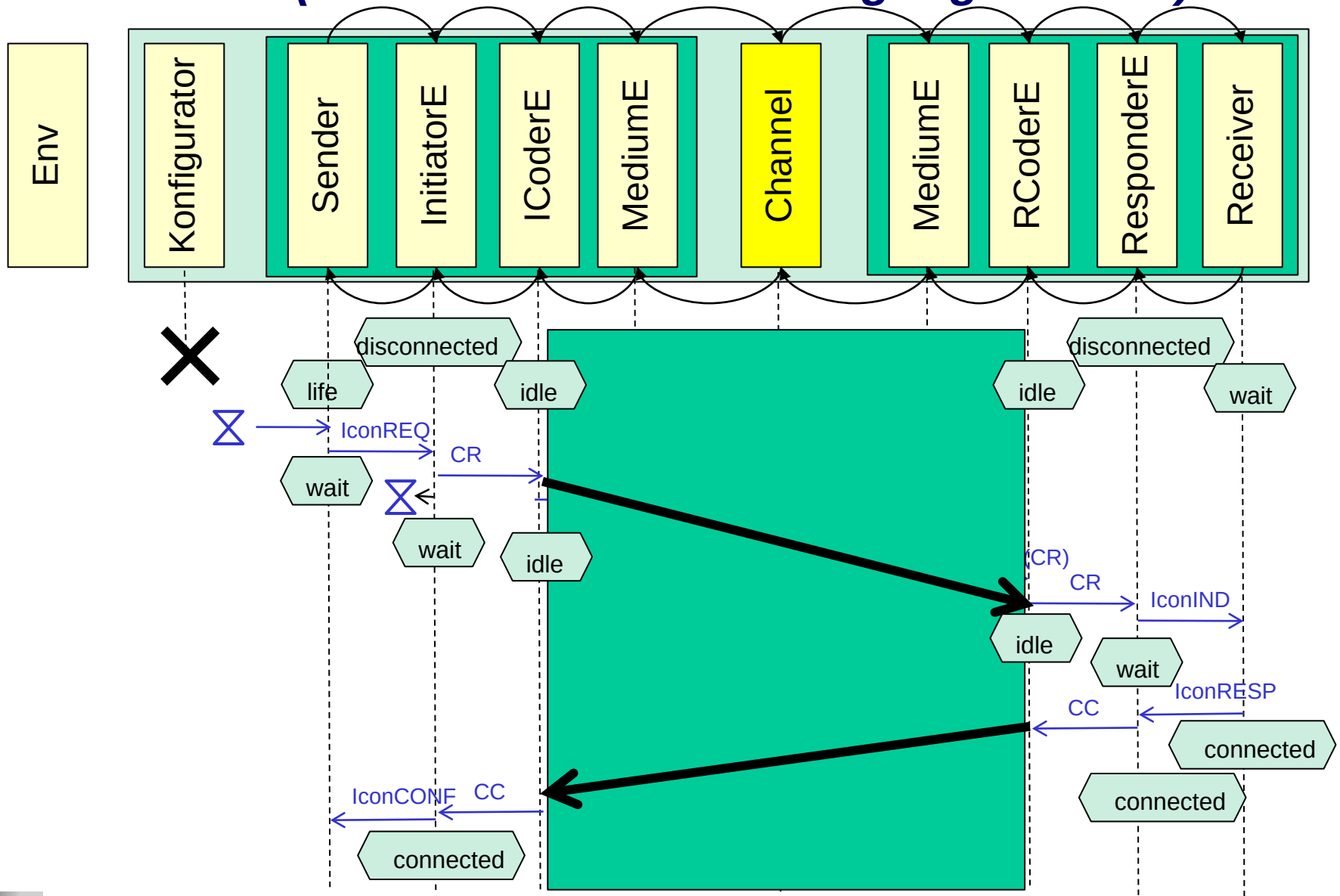
```
#ifndef TYPEDEFS_H_
#define TYPEDEFS_H_

typedef  char  ISduType;
typedef  int
SequenceNumber;

typedef  enum
{eCR, eCC, eDR, eDT,
eACK}  IPduType;

typedef  struct  MDatType {
    int id;
    SequenceNumber num;
    ISduType data ;
} MDatType;
#endif /*TYPEDEFS_H_*/
```



Ablauf


Env
Konfigurator
Sender
InitiatorE
ICoderE
MediumE
Channel
MediumE
RCoderE
ResponderE
Receiver
disconnected
disconnected
life
idle
idle
idle
idle
idle
wait
IconREQ
CR
wait
MdatREQ(CR)
PdatT(...)
wait
idle
PdatT(...)
MdatIND(CR)
idle
idle
CR
IconIND
idle
idle
wait
IconRESP
CC
connected
IconCONF
CC
connected
connected

Ablauf (Ausblenden der Übertragungsschicht)



8. Protokollentwicklung in SDL

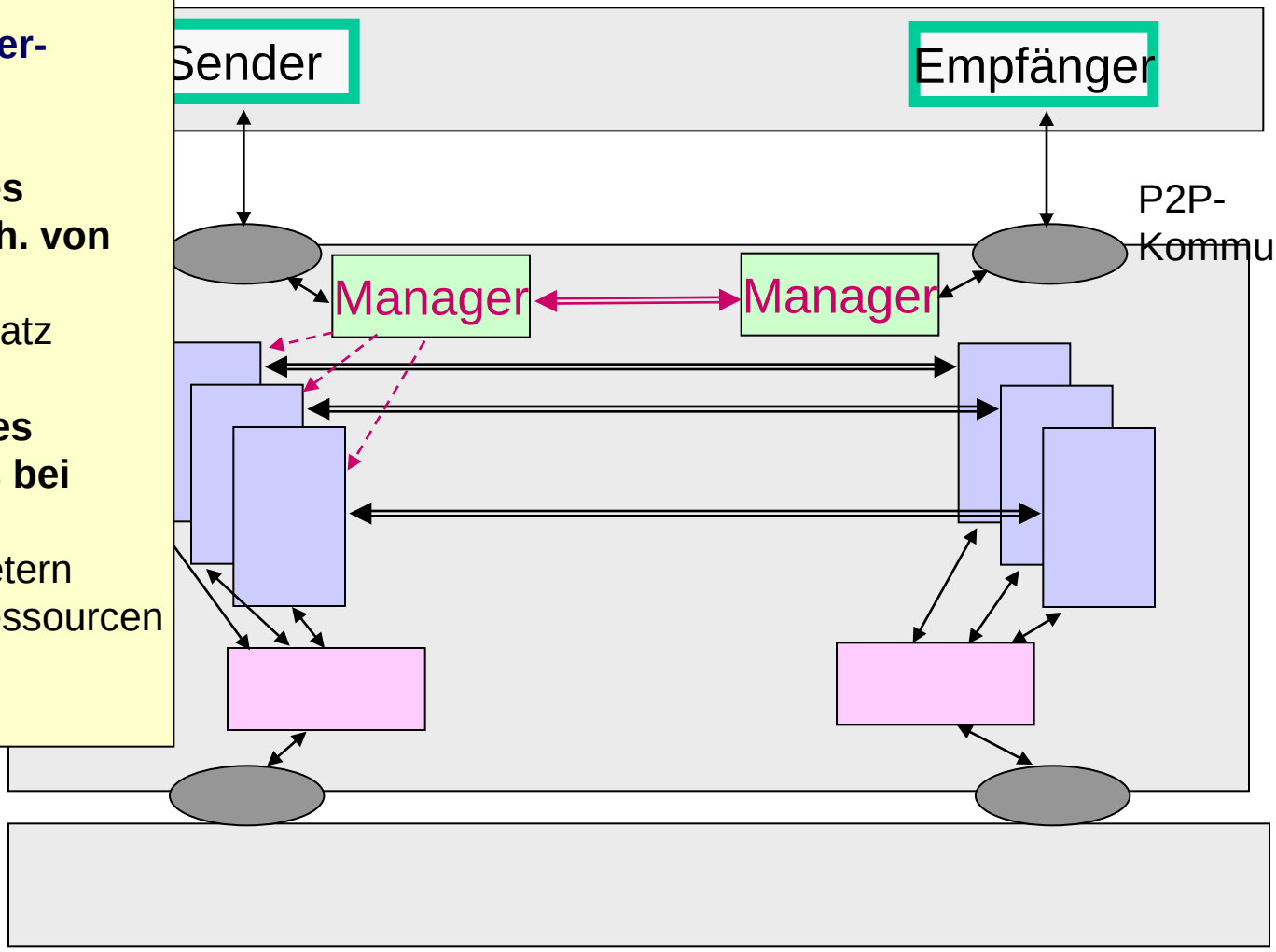
- OSI-Schichtenmodell (Konzept)
- InRes: allgemeine informale Beschreibung
- InRes-Dienstbeschreibung
- InRes-Protokollbeschreibung
- InRes-Semantik (Laufzeit-Betrachtung)
- Umsetzung in SDL/RT
- weiterer Ausbau

Allgemeine Protokollarchitektur

Typische Manager-Aufgaben:

Aushandlung des Protokolls in Abh. von
- Dienstgüte
- Ressourceneinsatz

Überwachung des Protokollablaufs bei
- Monitoring von Verkehrsparametern
- dynamischer Ressourcen zuteilung



N-1

(1) Verbesserungen

- Überprüfen der Typen lokaler Variablen, Nachrichtenparameter

- Erweiterung von Nachricht IerrIND um Parameter,
  der Abbruchursache mitteilt

- Erweiterung der Kanaleigenschaft um
  a) Verdopplungseffekte
  b) Verfälschungsaspekte (Prüfsumme als bool-Flag)
  c) Vertauschungsaspekte (bei Handshake unsinnig)

- Steuerung der Kanaleigenschaft von der Umgebung
  (Aufnahme weiterer Signale)
  a) Standard: korrekte Übertragung
  b) Verlustwahrscheinlichkeit
  c) Verdopplungswahrscheinlichkeit
  d) Verfälschungswahrscheinlichkeit

(2) Umgang mit dem RTDS

- Use-Case mit Sequenzdiagrammen

- Justierung der Timer (Simulator)

- Debugger-Funktionalität

- Einschränkung der MSC-Generierung

  Ausblendung der Übertragungsschicht

- MSC-Vergleich (Requirement – Simulation mit Fehlerfällen)