

# Kapitel 18 Interbus

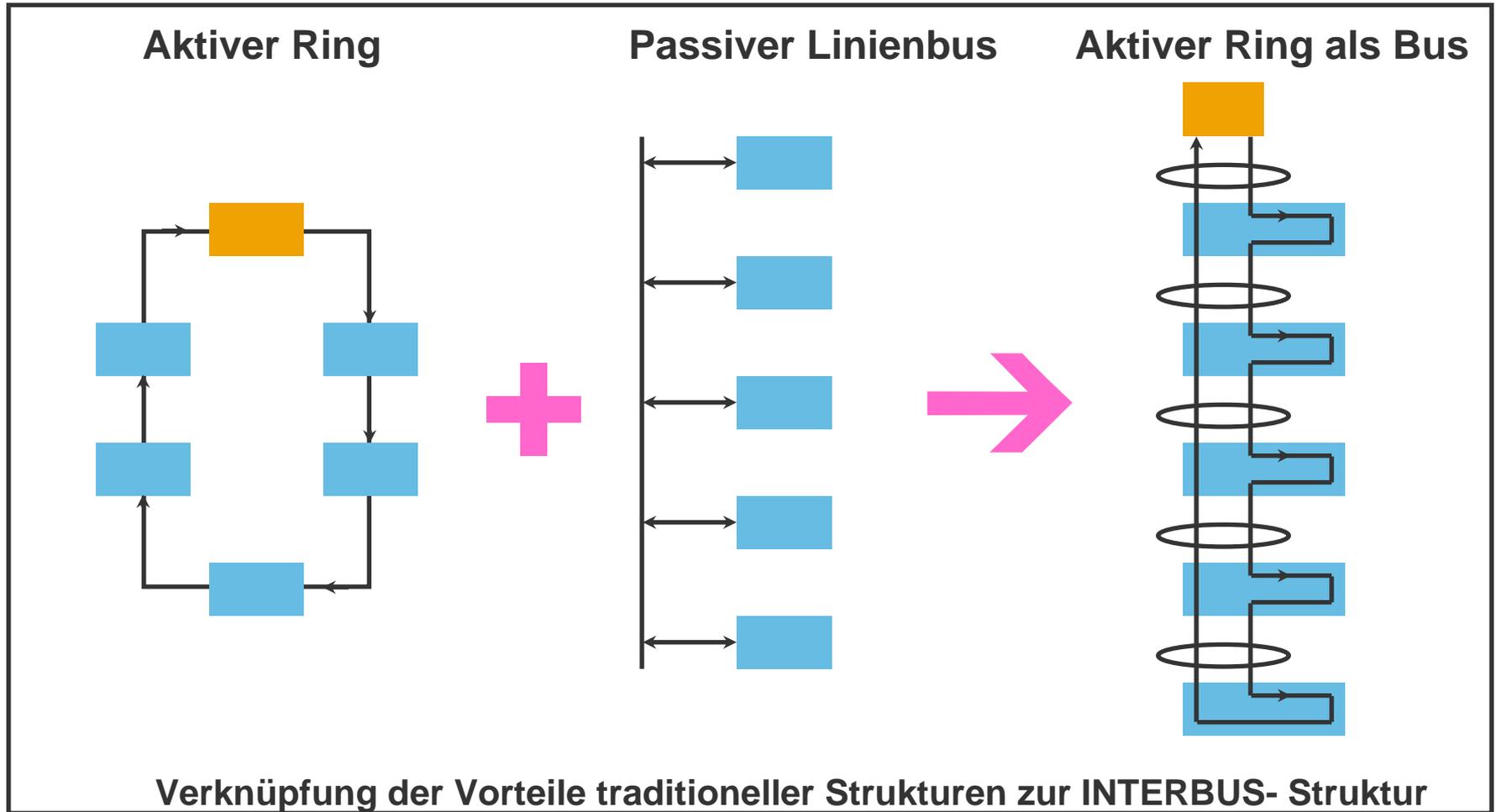
# Kapitel 18.1 Hersteller

# Firma Phoenix-Kontakt (Blomberg)

# Kapitel 18.2 Interbus-Topologie

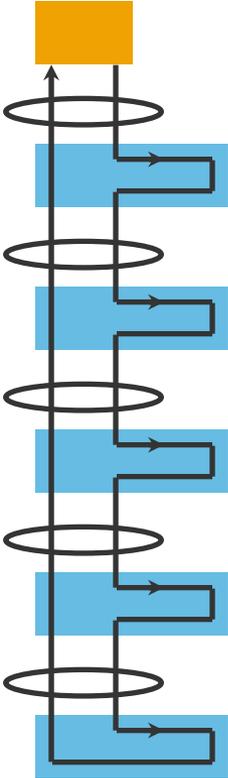
# Busstruktur

Bustopologie (1)

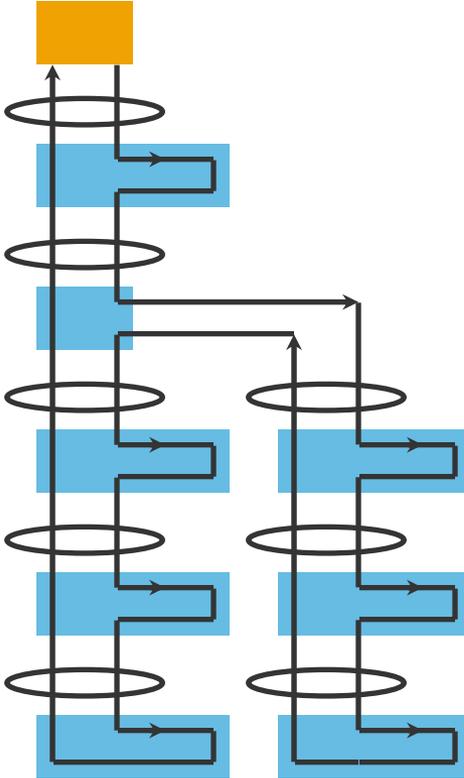


# Bustopologie (2)

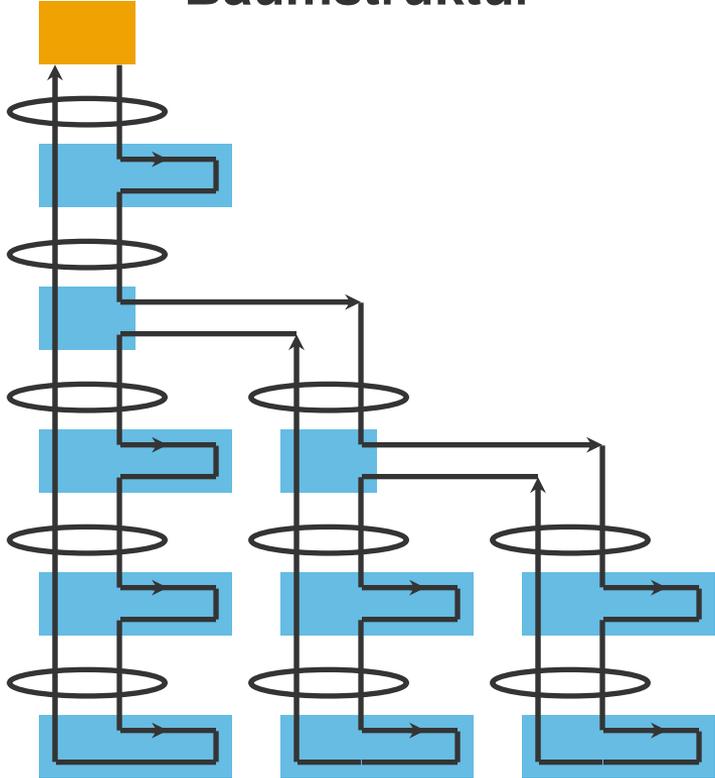
**Grundstruktur**



**Baumstruktur**



**Erweiterte Baumstruktur**



prinzipielle Strukturерweiterung

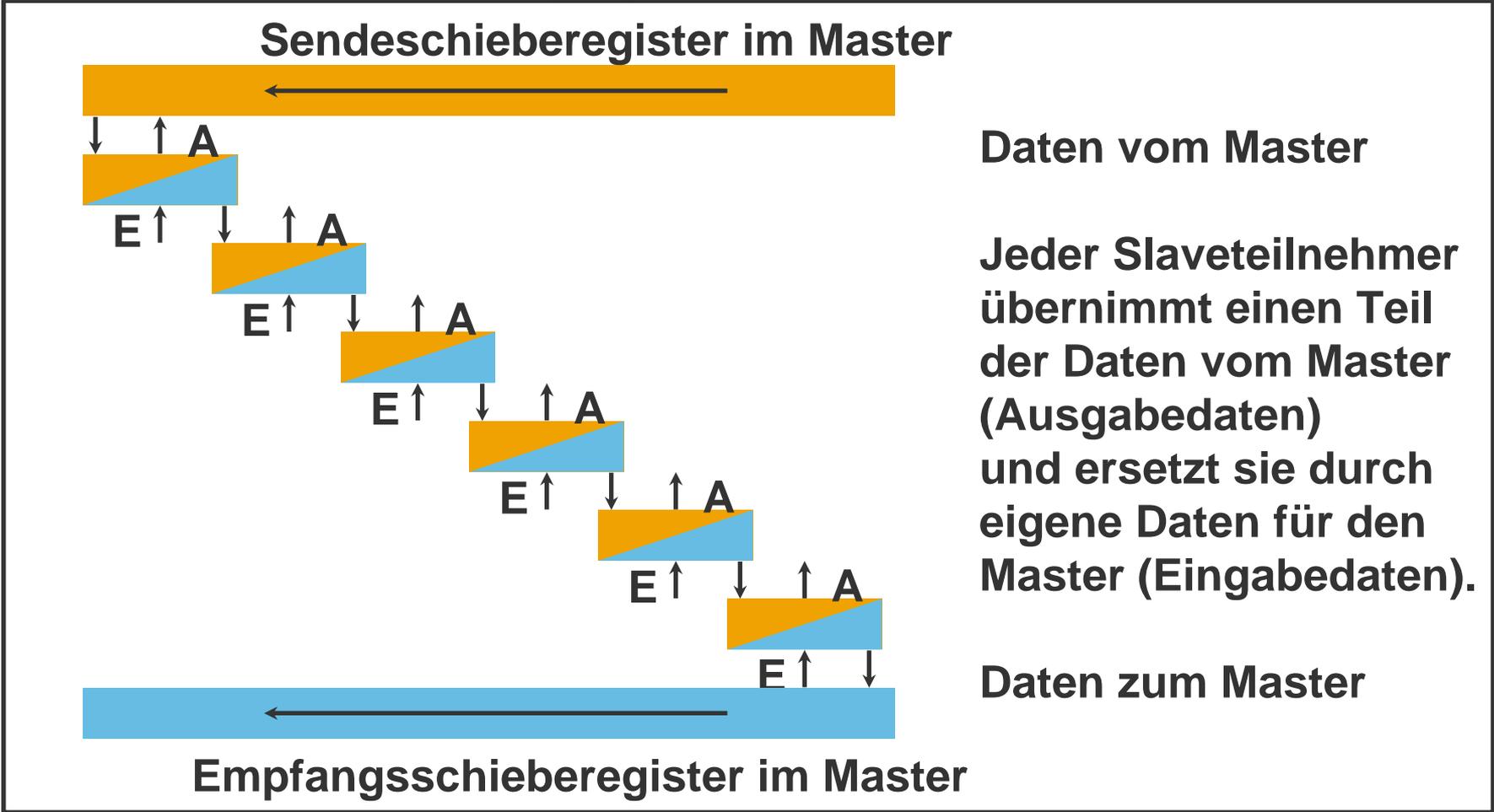
## Kapitel 18.3 Medien des Interbus

Zahlreiche unterschiedliche Kabeltypen  
mit unterschiedlichen Aderzahlen

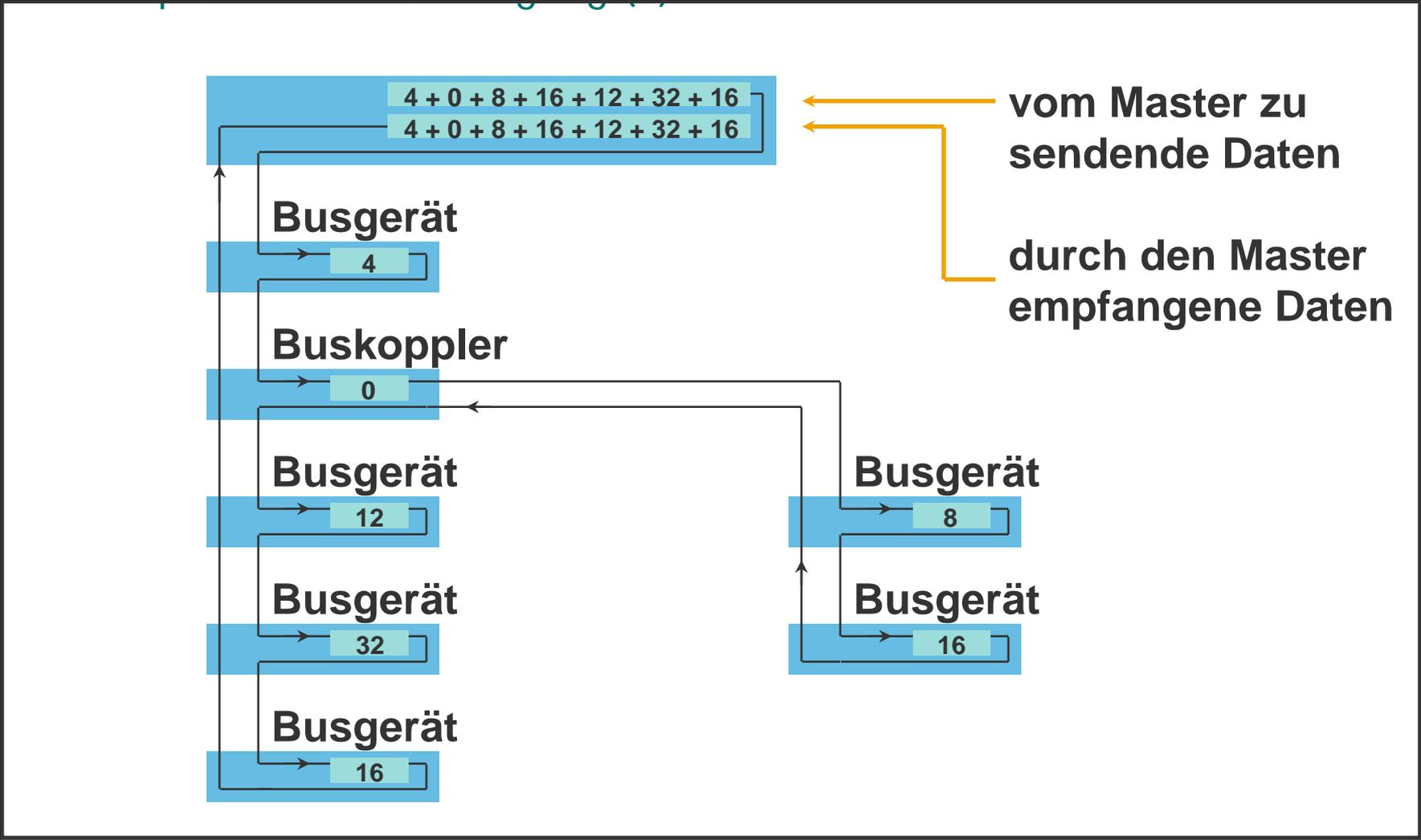
# Kapitel 18.4 Interbus-Adressierung

Datenübertragung auf dem INTERBUS

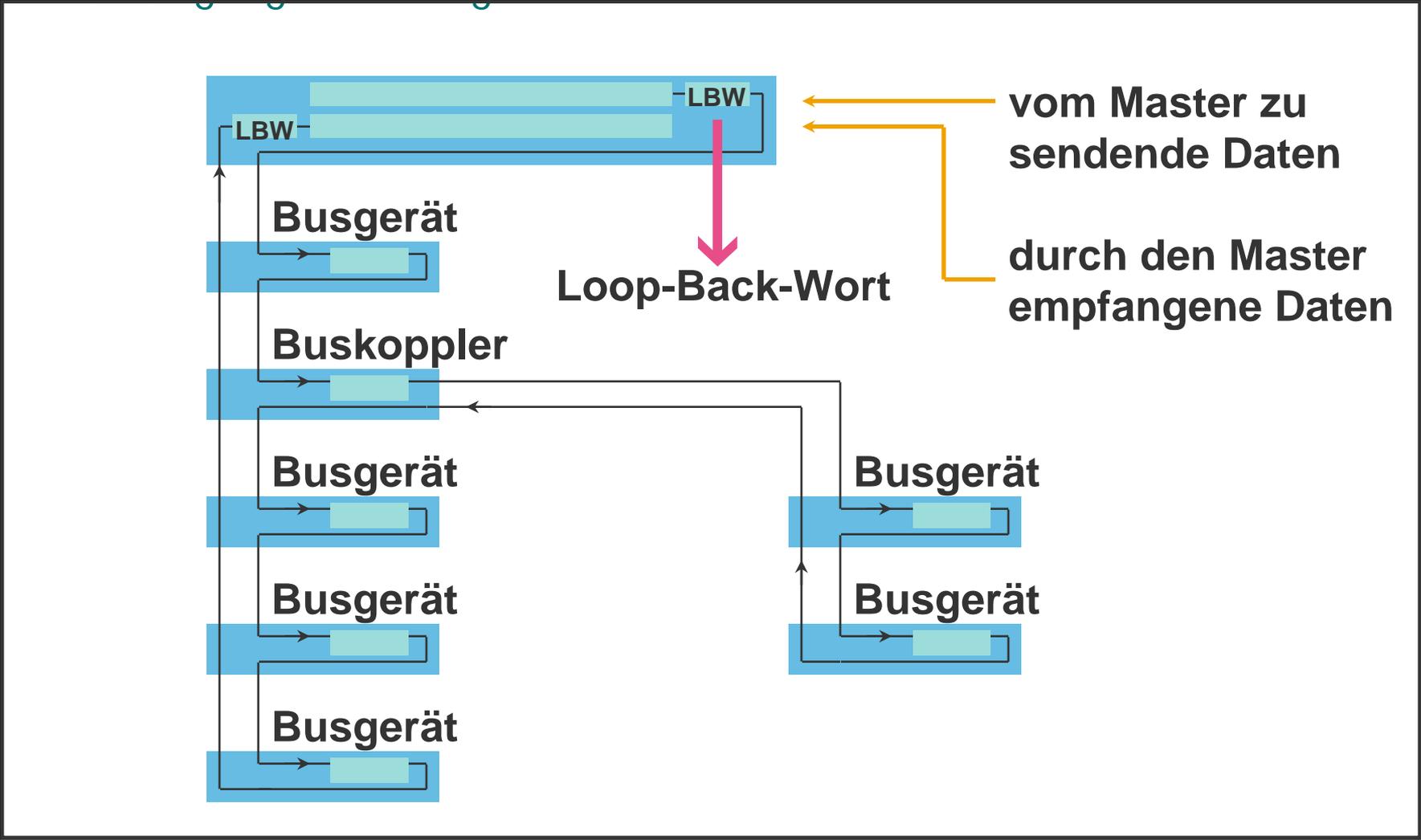
Prinzip der Datenübertragung (1)



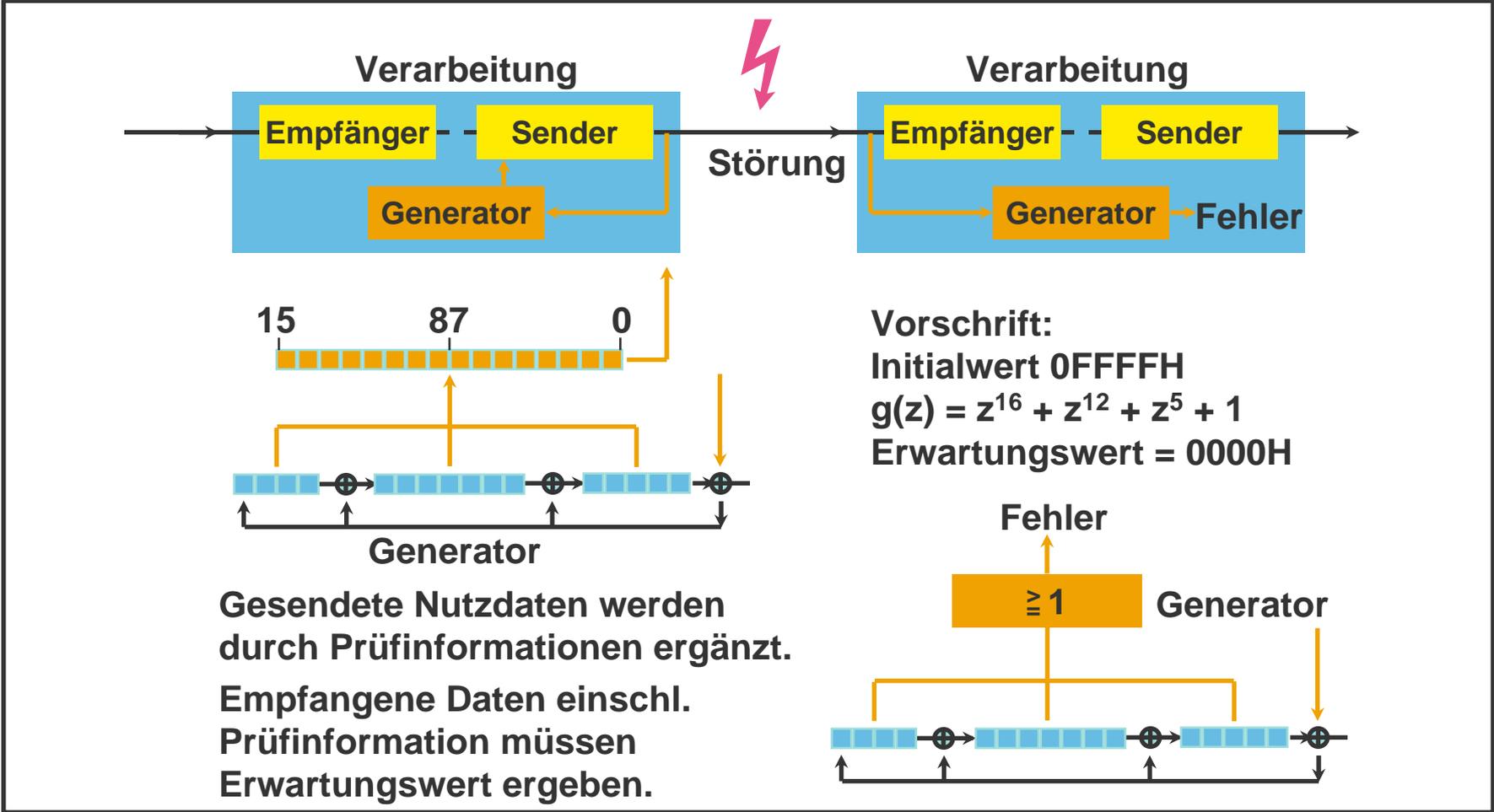
Prinzip der Datenübertragung (2)



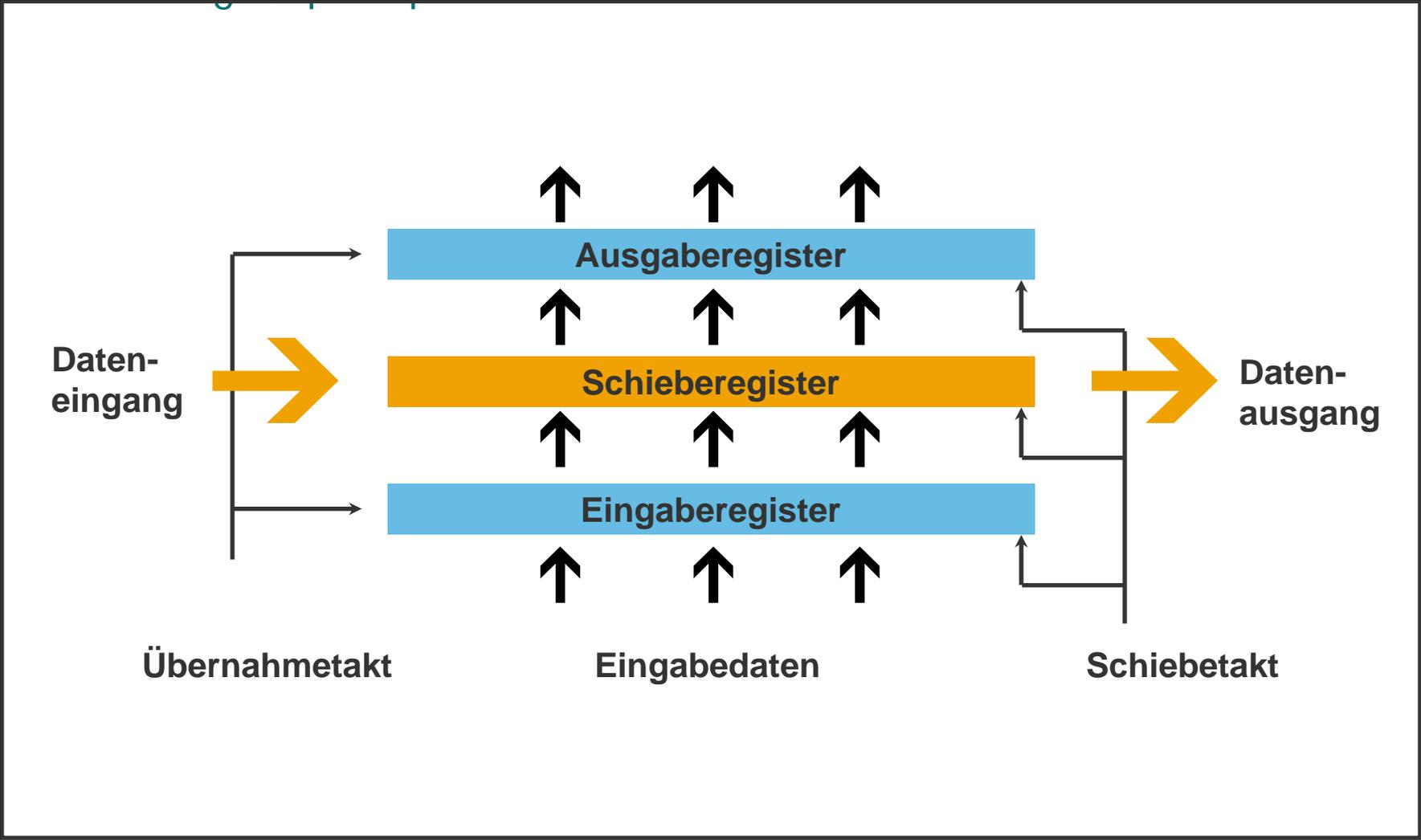
Übertragungssicherung



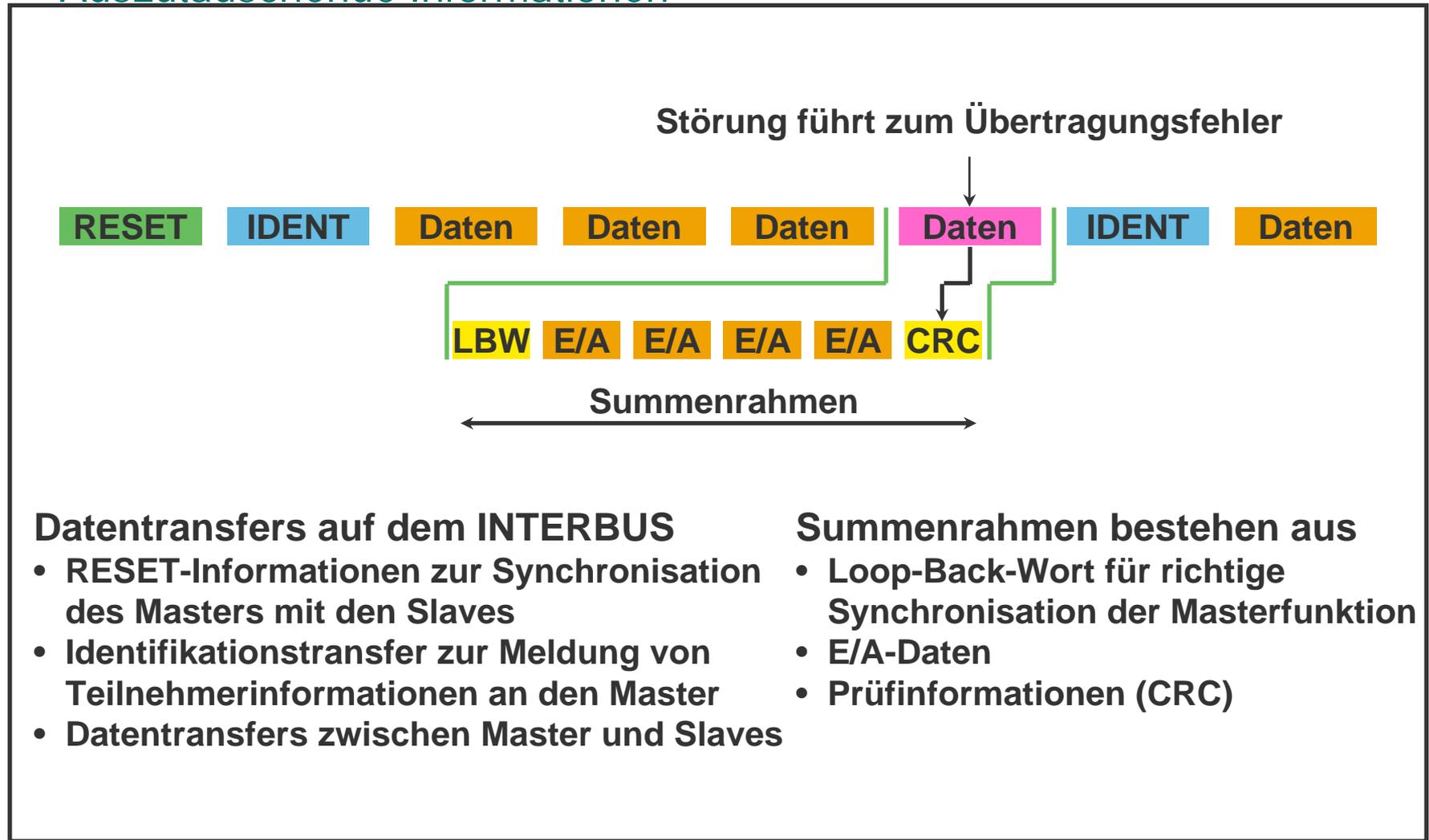
# CRC-Überwachung der Datenübertragung



Ein-/Ausgabeprinzip

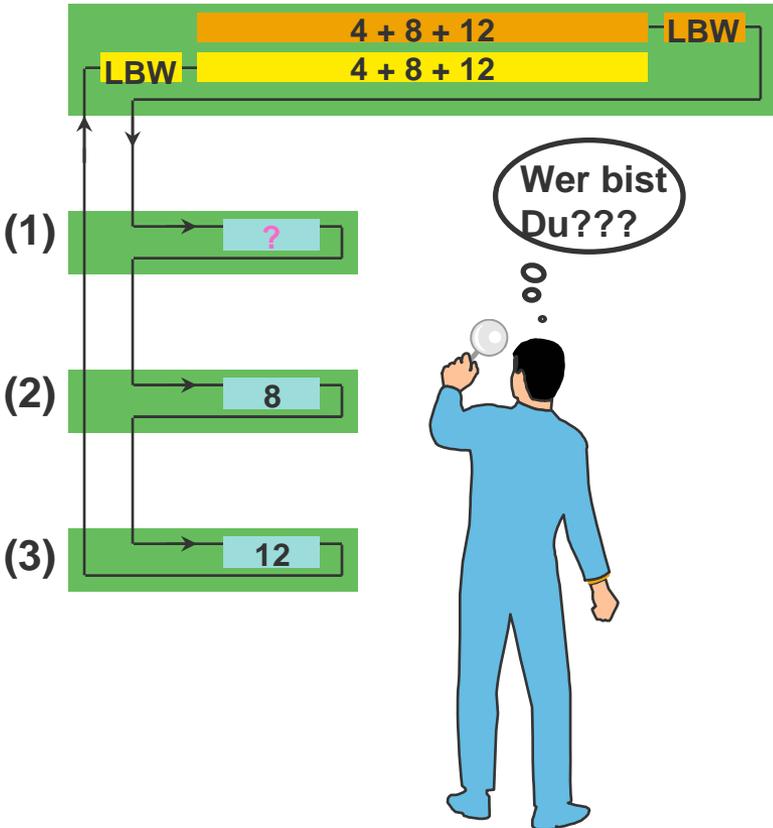


## Auszutauschende Informationen



Funktionsbeispiel

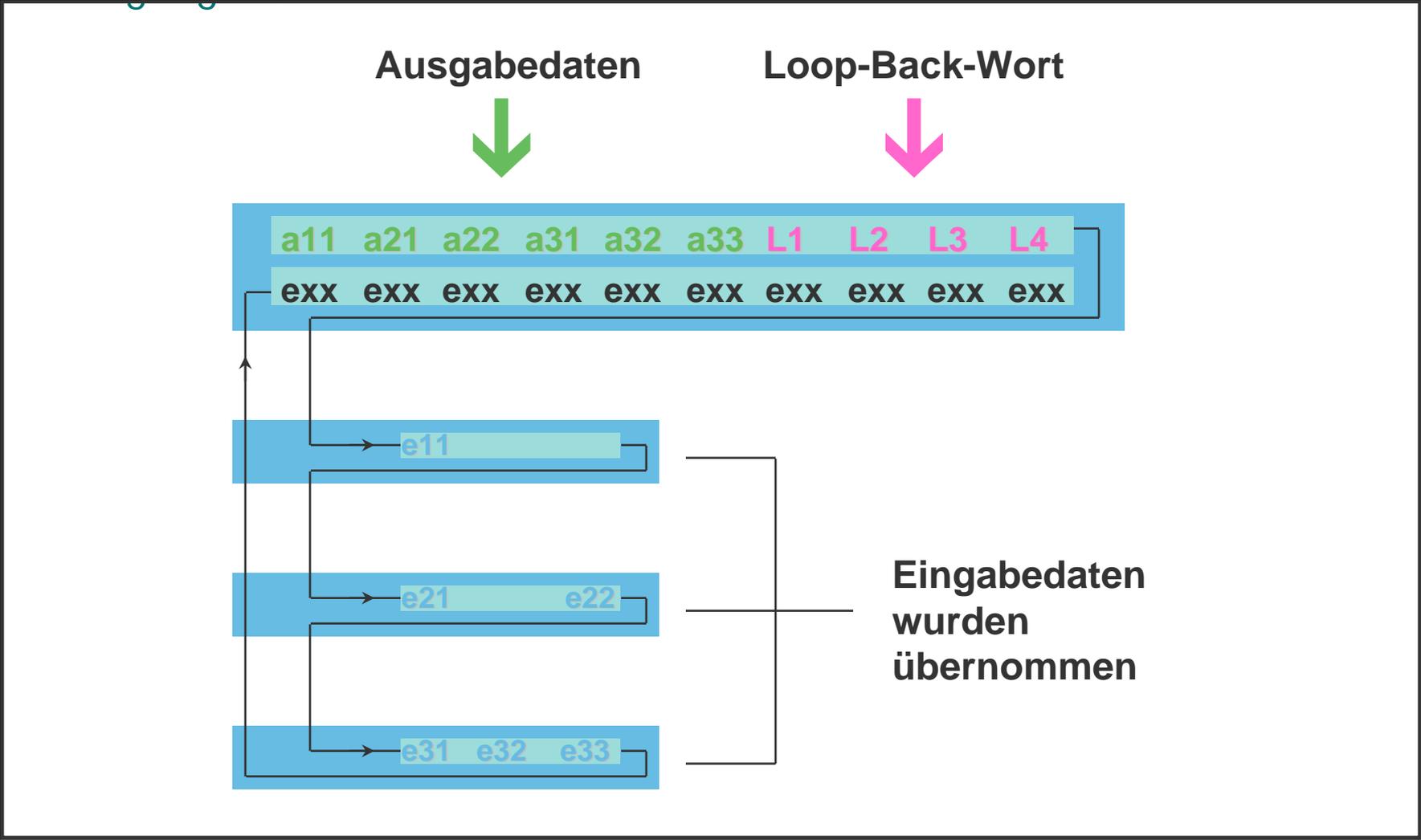
Ausgabewerte von SPS / IPC



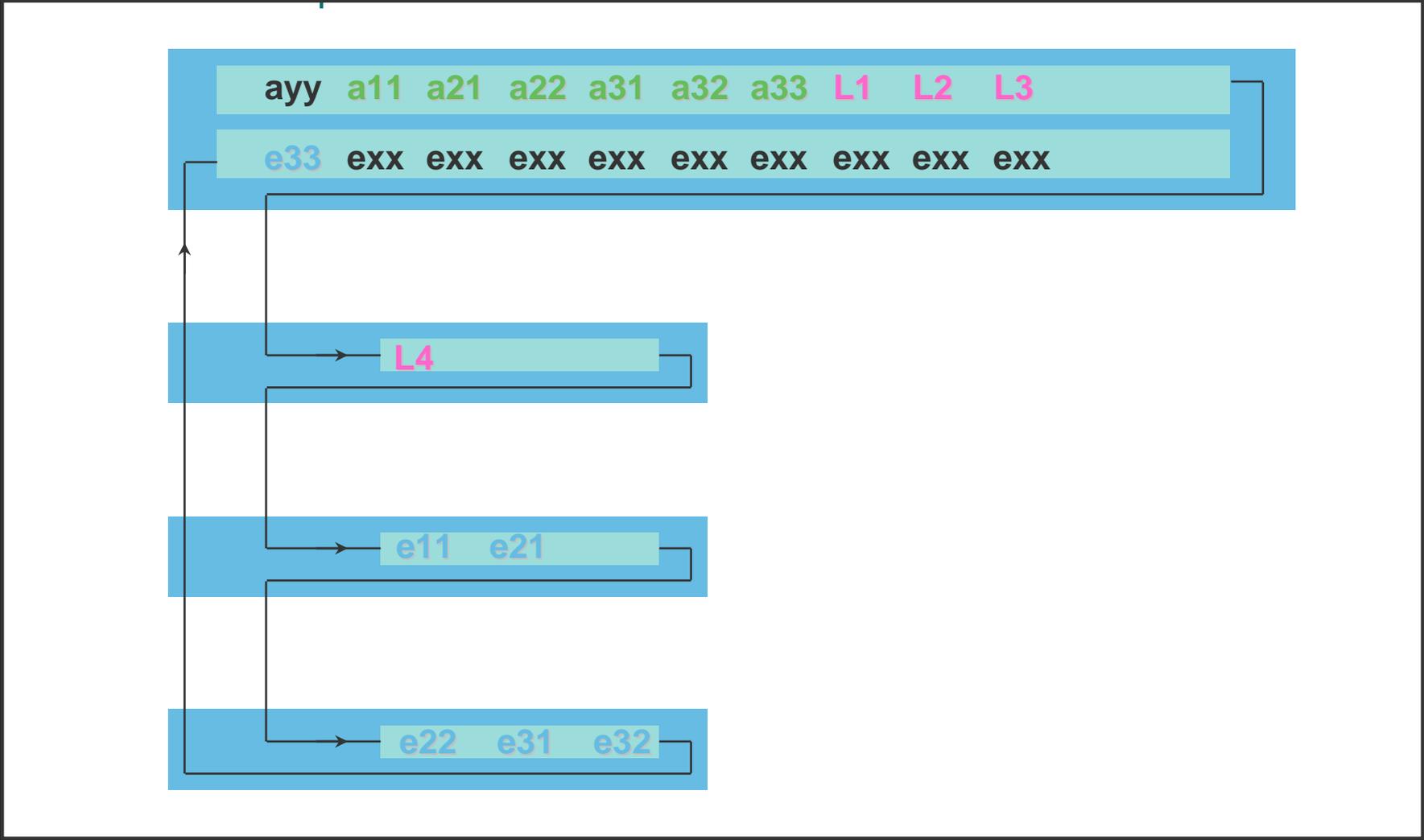
a11	a21	a22	a31	a32	a33	L1	L2	L
exx	ex							
e11								
e21		e22						
e31		e32		e33				

Eingabewerte der Busgeräte

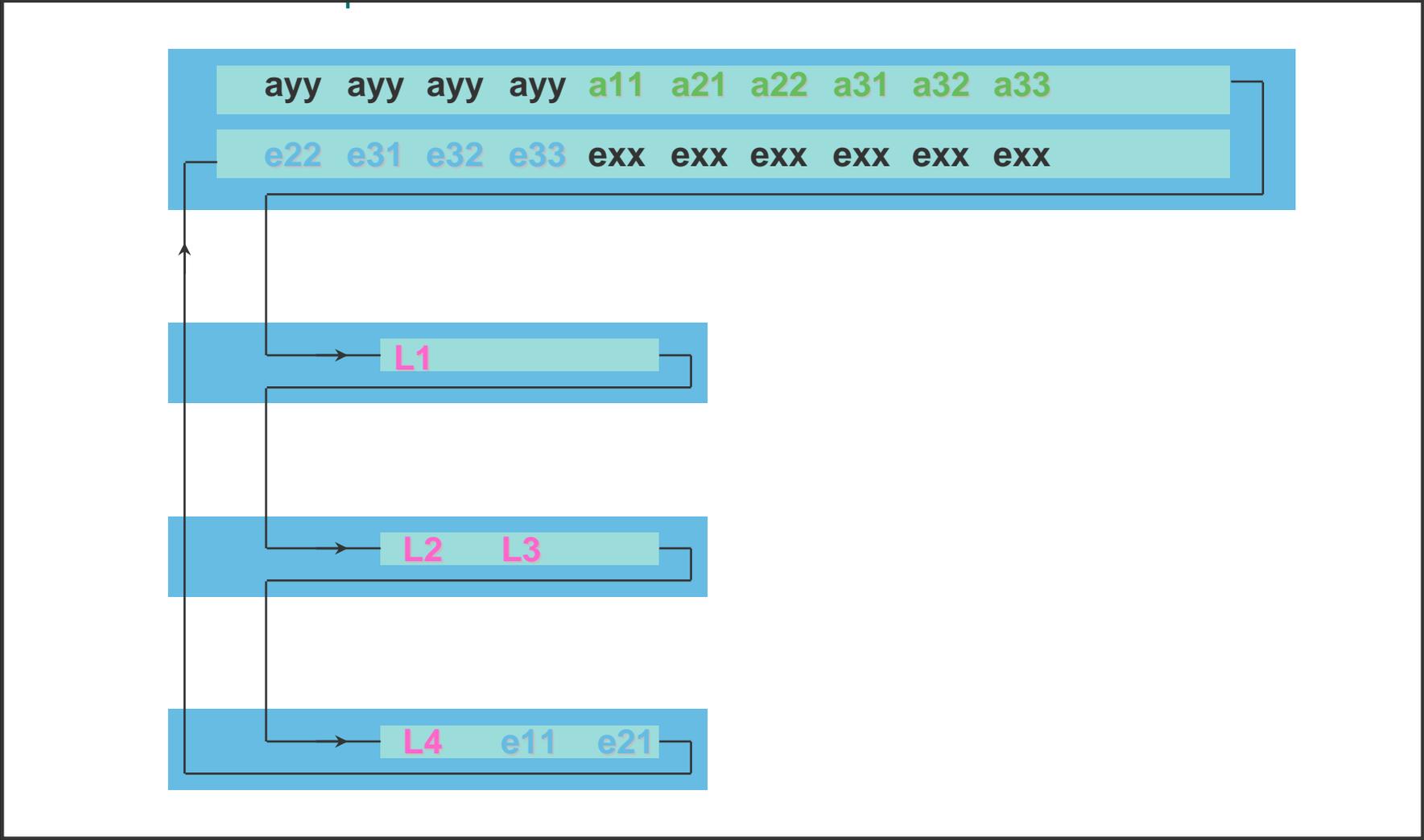
Ausgangssituation



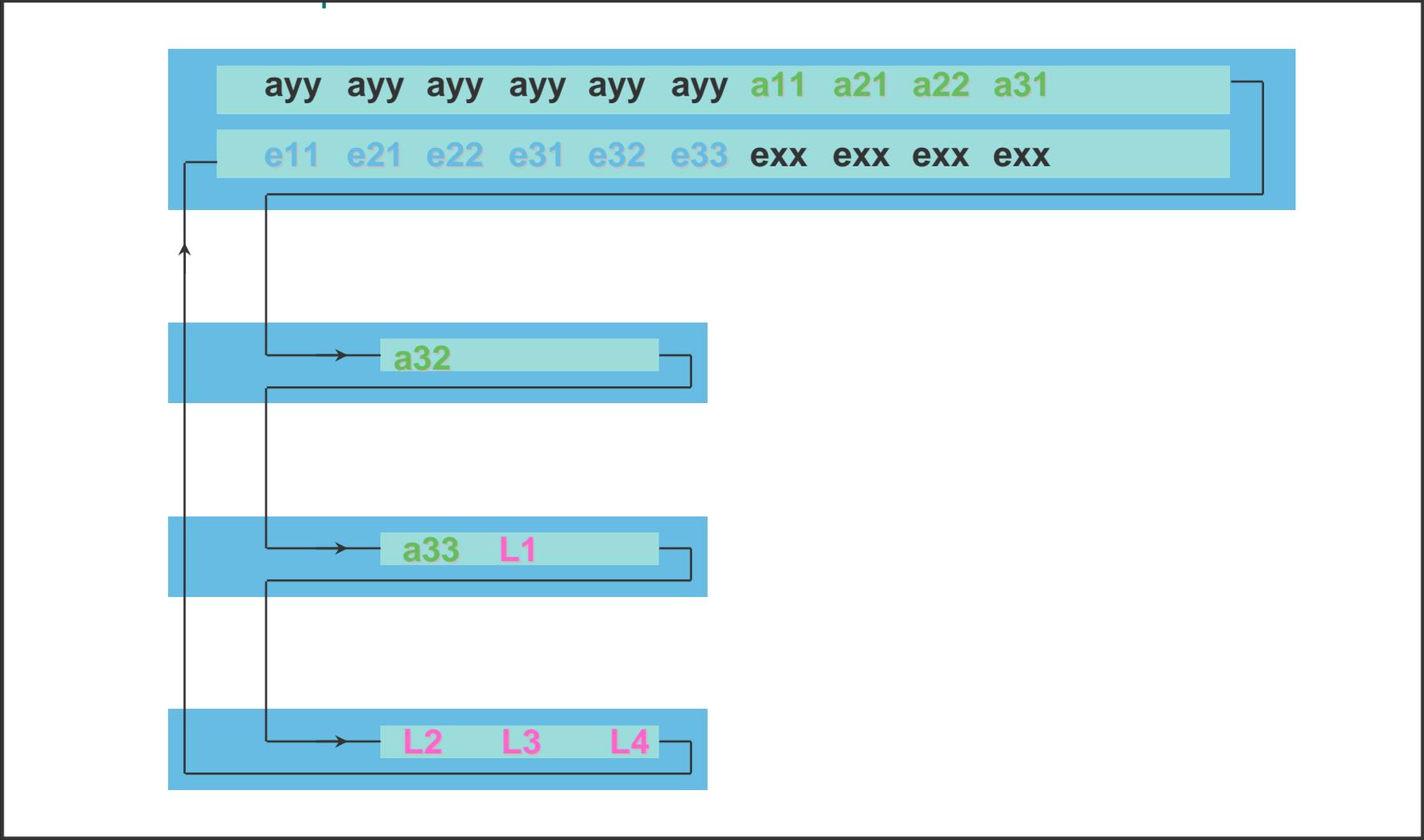
Erste Schiebephase



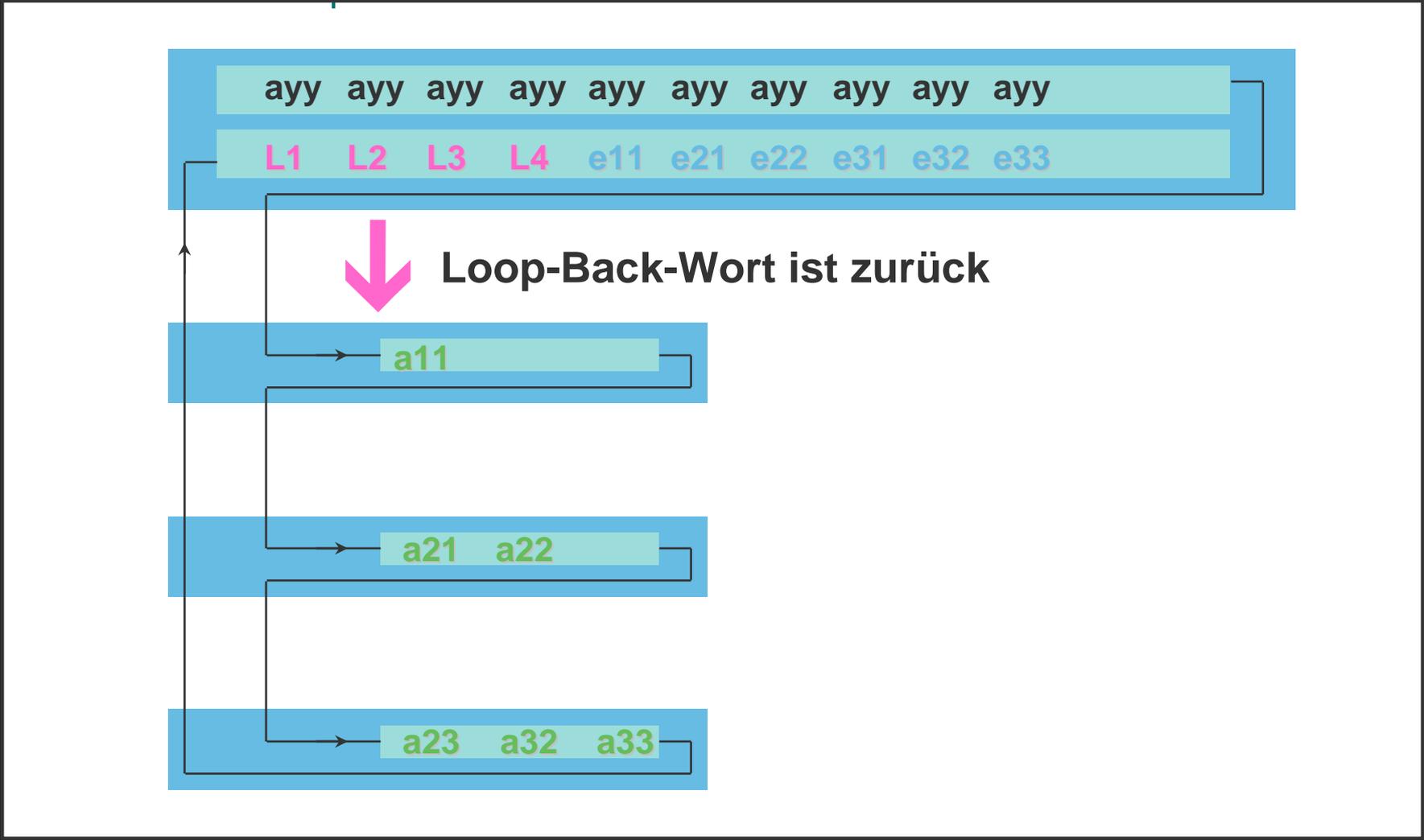
Zweite Schiebephase



Dritte Schiebephase



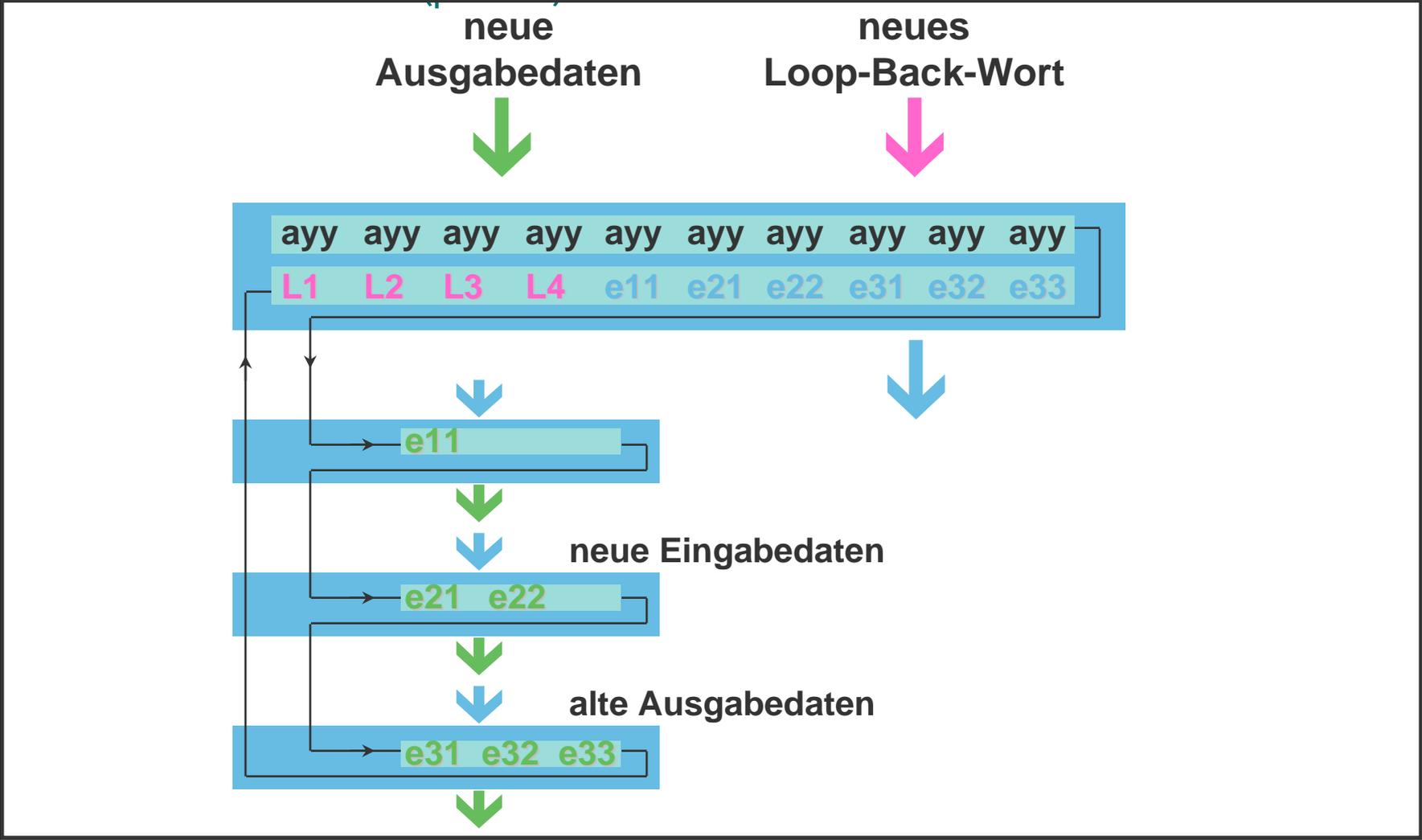
Vierte Schiebephase



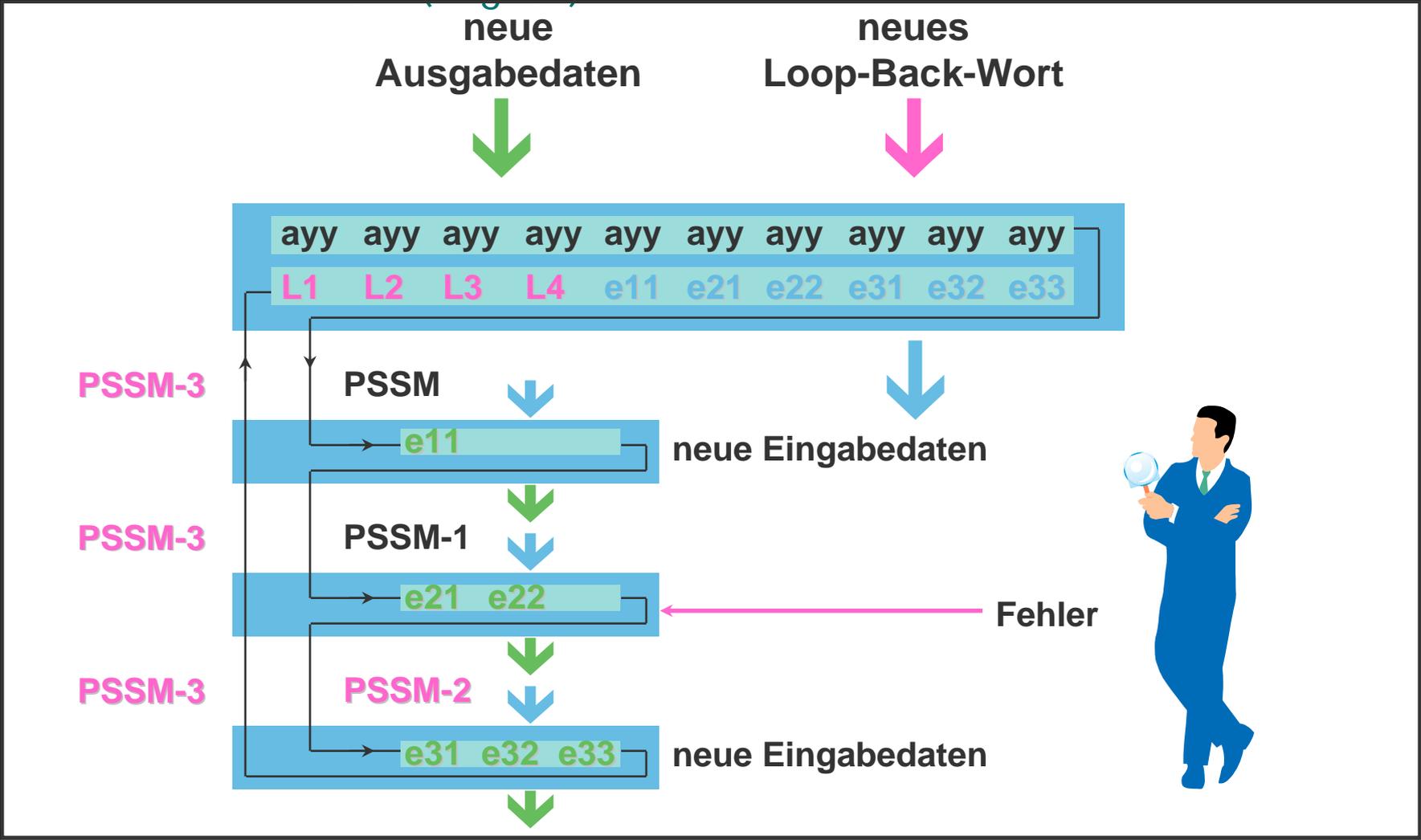
Prüfsumme



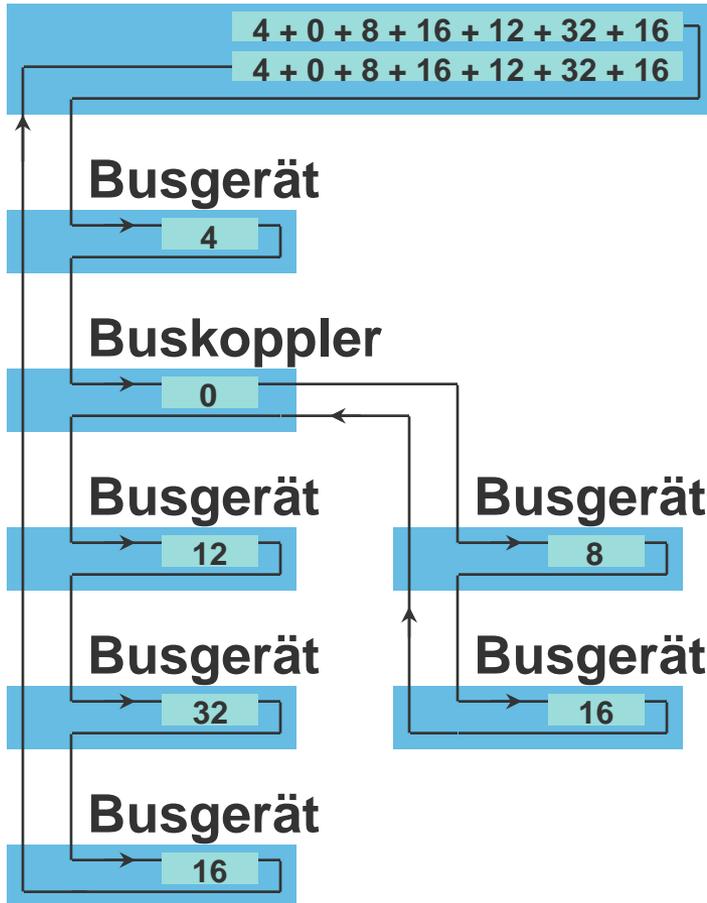
Prüfsummenstatus (positiv)



Prüfsummenstatus (negativ)



## Übertragungszeiten: Buszykluszeit



Die Buszykluszeit: Zeit vom Senden des ersten Bits bis zum Empfangen des letzten Bits des Datenzyklus. Sie hängt von der Buskonfiguration ab.

$$T_{BC} = t_{bit} * (13*6 + 13*n + 2*m)$$

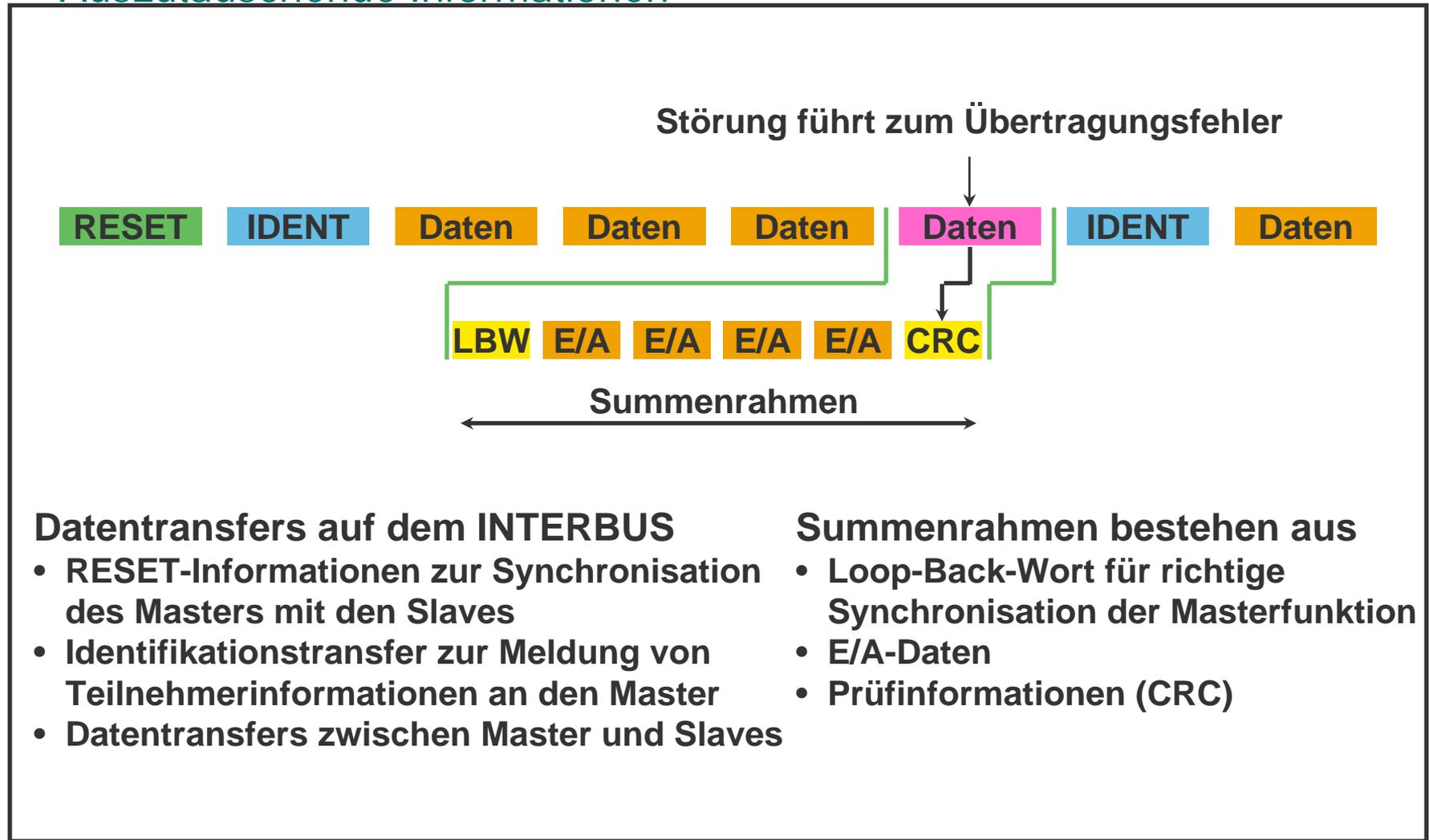
- 2 Bit Verzögerungszeit je Busteilnehmer
- 13 Bit Verzögerungszeit je Paket mit max. 8 bit Daten
- 6 Pakete mit je 13 Bit Verzögerungszeit / Buszyklus (2\*Loop-Back-Wort, 2\*Prüfsumme, 2\*Prüfsummenstatus)

Beispiel:

$t_{bit}$	= 1/500KHz = 2µs
n	= 11 Datenoktetts
m	= 7 Busteilnehmer
$T_{BC}$	= 0,47 ms

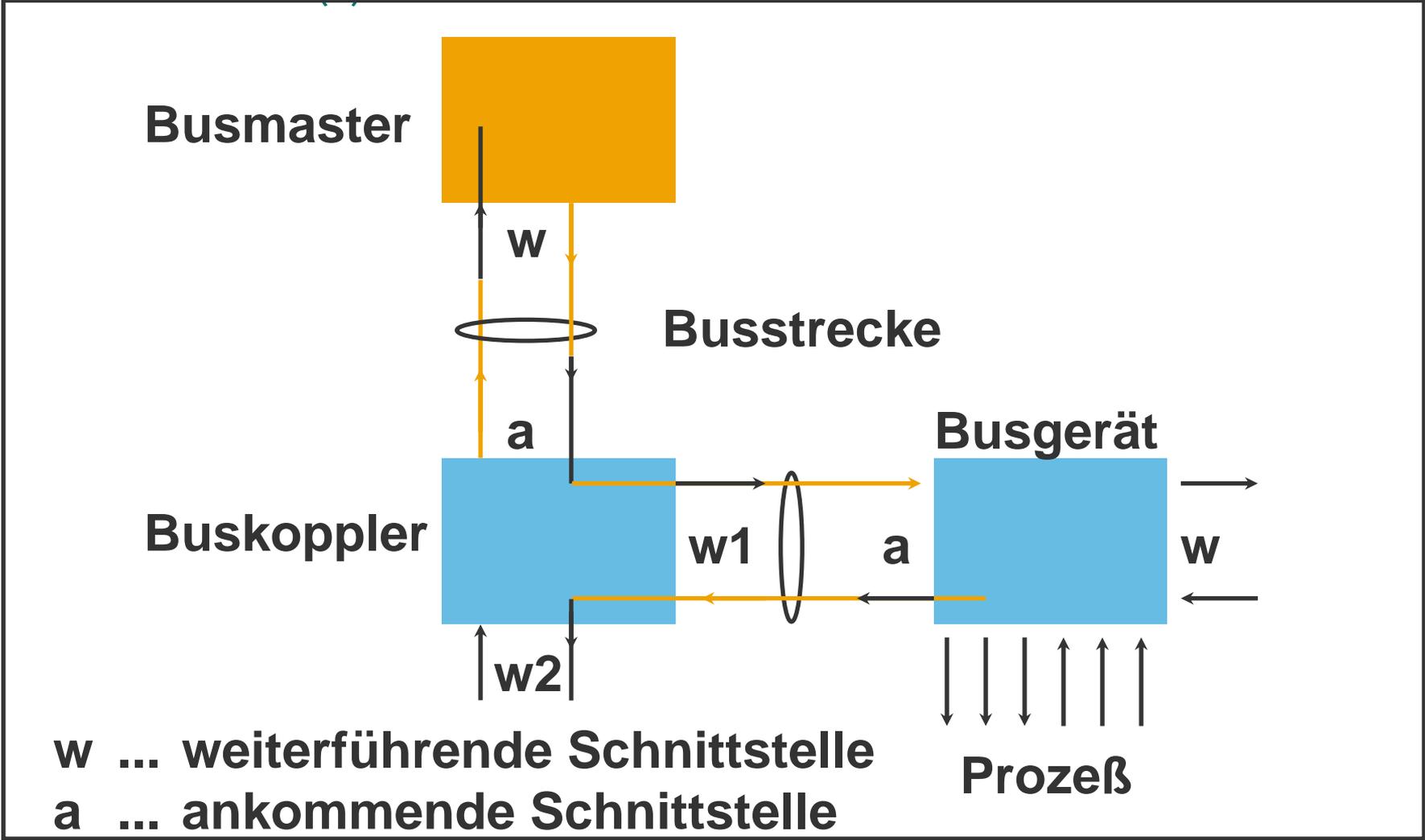
# Kapitel 18.5 Interbus-Datenrahmen

## Auszutauschende Informationen

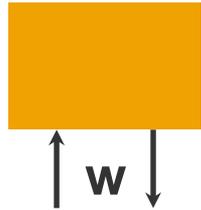


# Kapitel 18.6 Interbus-Systemkomponenten

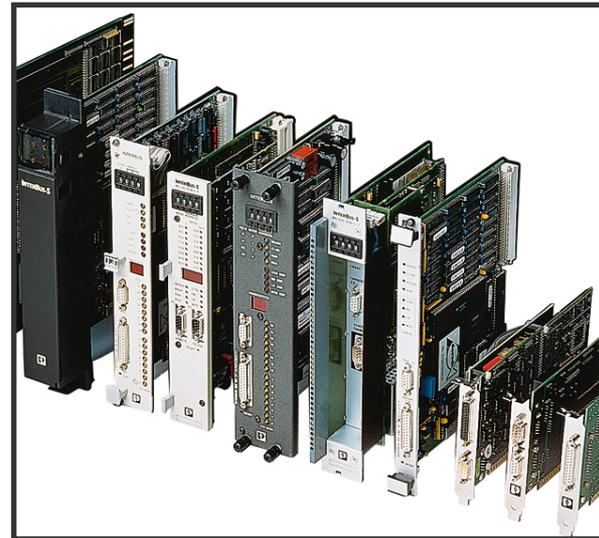
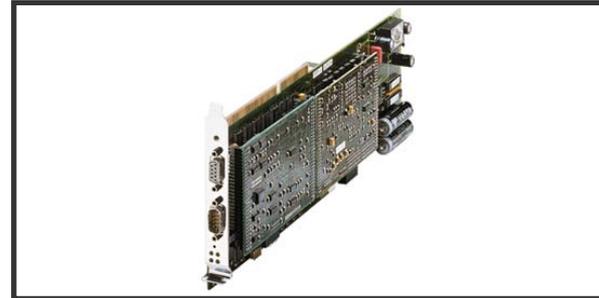
Buselemente (1)



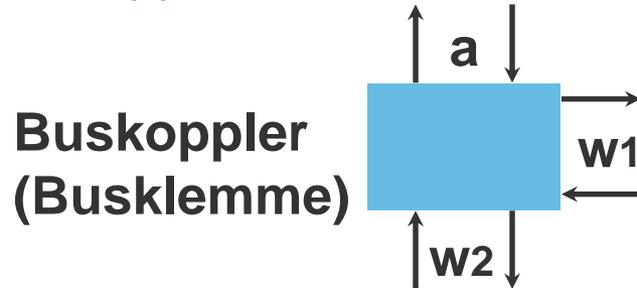
## Busmaster

**Busmaster**

- **Anschaltbaugruppe für SPS  
Industrie-PC**
- **verantwortlich für  
Datentransfer zwischen  
SPS / Industrie-PC und Busgerät  
Kommunikation zwischen  
Applikationen  
Busmanagement (Konfiguration,  
Fehlererkennung, Rekonfiguration)**



## Buskoppler (Busklemme)

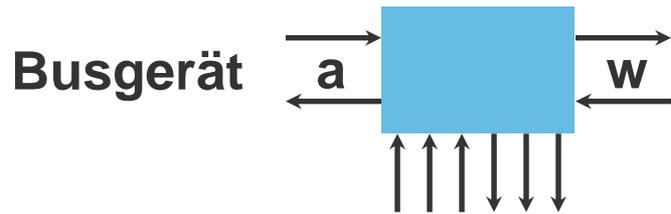


- **Busteilnehmer für**  
Verzweigung des Bussystems  
(Baumstruktur)  
Anschluß von Lokalbussen  
(E/A-Stationen an den Feldbus)
- **verantwortlich für**  
Konfigurationsmöglichkeit bei  
Initialisierung oder  
Fehlererkennung

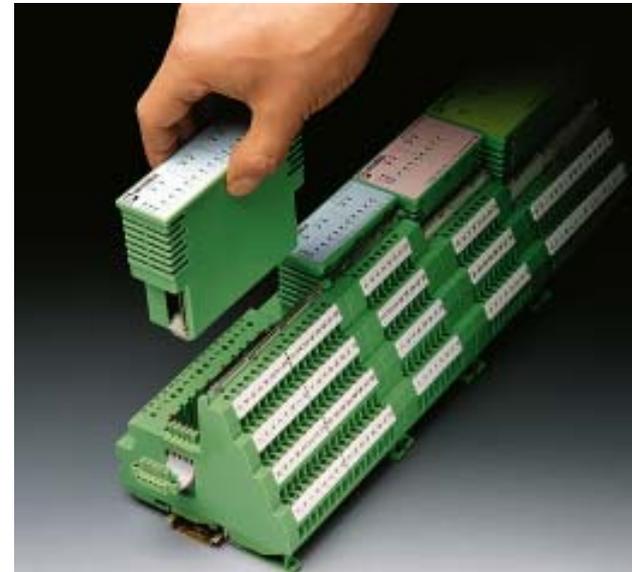


# Kapitel 18.7 Interbus-Geräte

## E/A-Geräte im Bus



- **Busteilnehmer für**  
Prozeßdatenübertragung für
  - analog / digital
  - Eingabe / AusgabeParametrierung einer Steuerung
- **verantwortlich für**  
Datentransport mit Signal-  
regenerierung und Fehlererkennung  
Ein-/Ausgliederung der teilnehmer-  
bezogenen Daten



# Kapitel 18.8 Interbus-Programmierung

Programmierung mit dem Tool PCworks

Codesys-Anwendung

The screenshot displays the PC-WORX software interface for configuring an Interbus network. The main window shows a project tree on the left with components like 'Steuerung/Rechner', 'Anschaltbaugruppe', and 'Phoenix Contact' modules. A central table lists participants with their addresses and connection points.

T.-Nr.	Anschlusspunkt	I/Q	Länge	Byte	Bit	Lage (Byte/Bit)	MZ	Variable	Funktionstext
1	1.5	16-Bit_Ausgang							
2	1.5	1.1.1							
3	1.5	1.2.1							
4	1.5	1.1.4							
5	1.5	1.2.4							
6	1.5	2.1.1							
7	1.5	2.2.1							
8	1.5	2.1.4							
9	1.5	2.2.4							
10	1.5	3.1.1							
11	1.5	3.2.1							
12	1.5	3.1.4							
13	1.5	3.2.4							
14	1.5	4.1.1							
15	1.5	4.2.1							
16	1.5	4.1.4							
17	1.5	4.2.4							

The right-hand pane shows a ladder logic diagram for 'Graphik: STEP2.TEST01.Test01'. It features two logic networks. The first network has two normally open contacts labeled 'Taster2' and 'Taster4' connected in series to an 'OR' gate. The output of the OR gate is connected to the 'clock' input of a 'SIMPFF\_2 SIMPFF' timer block, which has an 'output' connected to 'Lampe2'. The second network has two normally open contacts labeled 'Taster3' and 'Time1' connected in series to the 'IN' input of a 'TOF\_1 TOF' timer block. The 'PT' input is also connected to 'Time1'. The 'Q' output of the TOF block is connected to 'Lampe3'. The 'ET' input of the TOF block is also connected to 'Time1'.

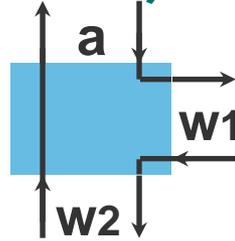
Programmierung und Parametrierung

eines Interbus-Netzwerks mit PC-Worx

# Kapitel 18.9 Weitere Informationen

# Signalwege durch den Buskoppler (Busklemme)

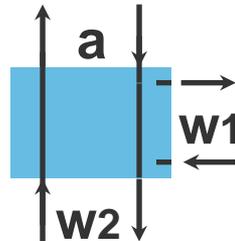
Buskoppler



**Zwei weiterführende Bussegmente**

- Bildung von “Baumstrukturen”
- E/A-Stationen an w1 gebildet

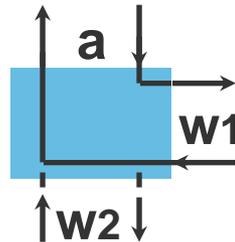
Buskoppler



**Kein weiterführendes Bussegment**

- Letztes Busgerät in der E/A-Station

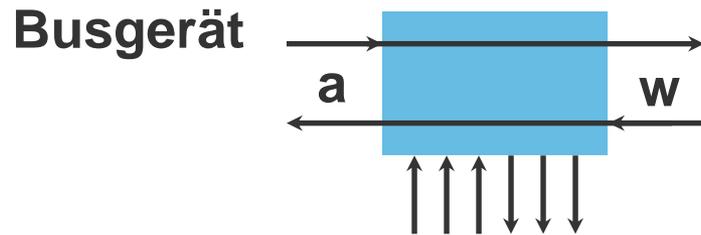
Buskoppler



**Ein weiterführendes Bussegment w2**

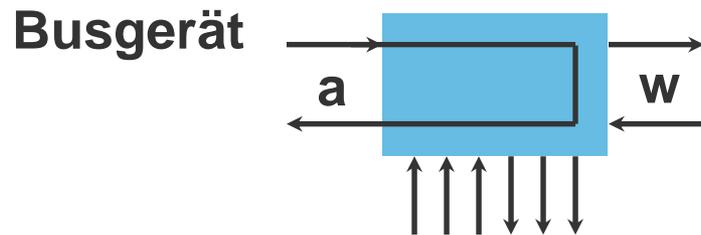
- Substruktur an w2 ausgefallen
- oder abgeschaltet
- w2 ist nicht beschaltet
- letzte E/A-Station im Fernbus

# Signalwege durch die Busgeräte



**Ein weiterführendes Bussegment**

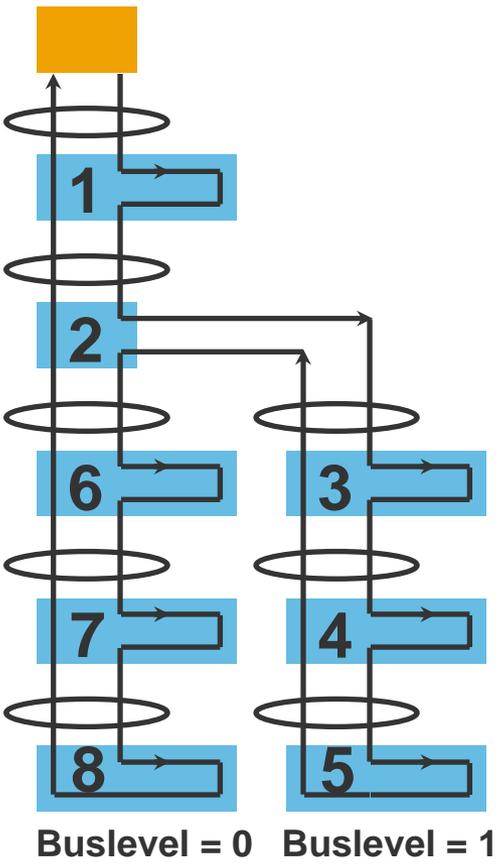
- Aufreihung



**Kein weiterführendes Bussegment**

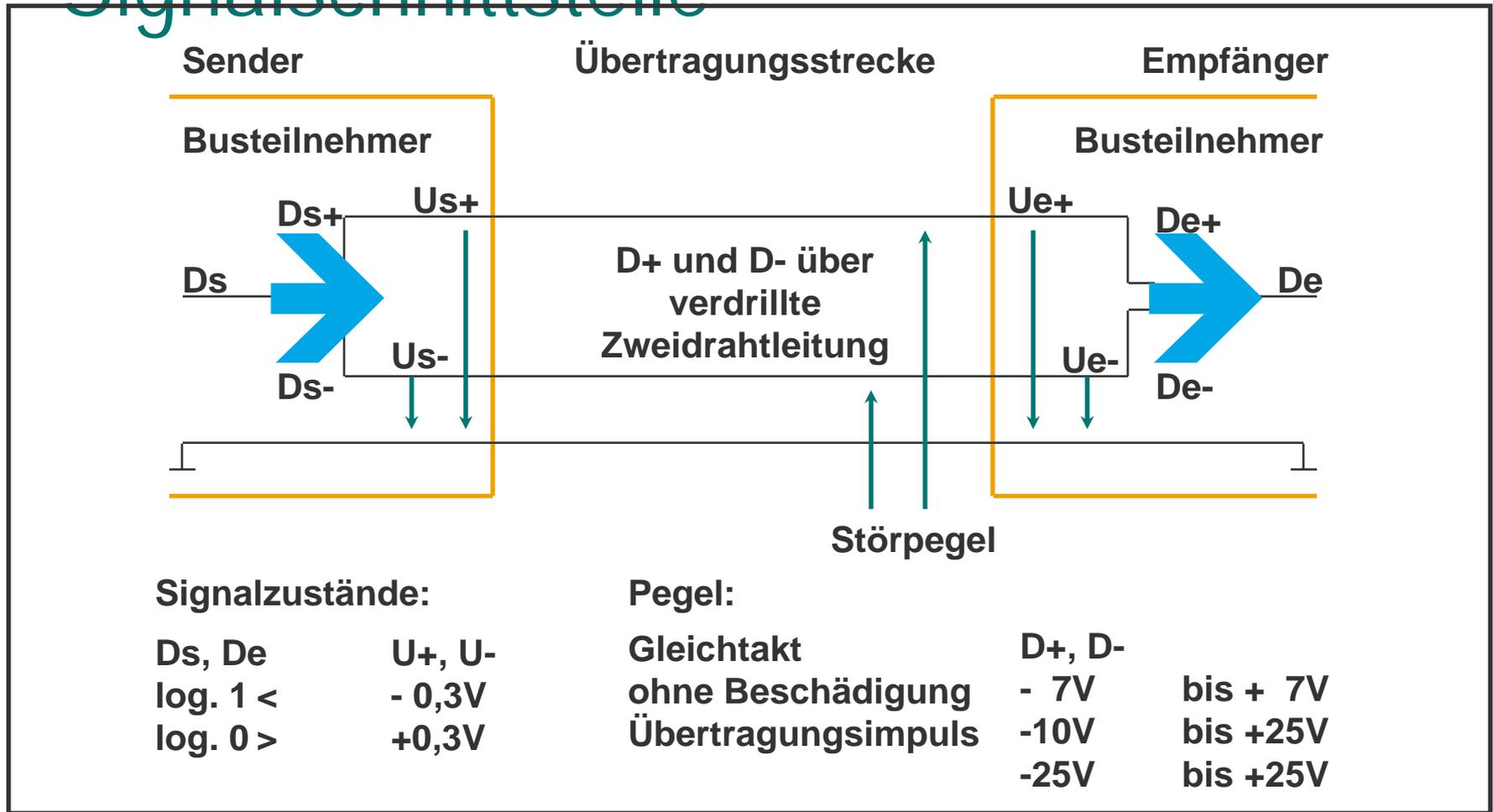
- Letztes Busgerät in der E/A-Station

# Netzkonfiguration

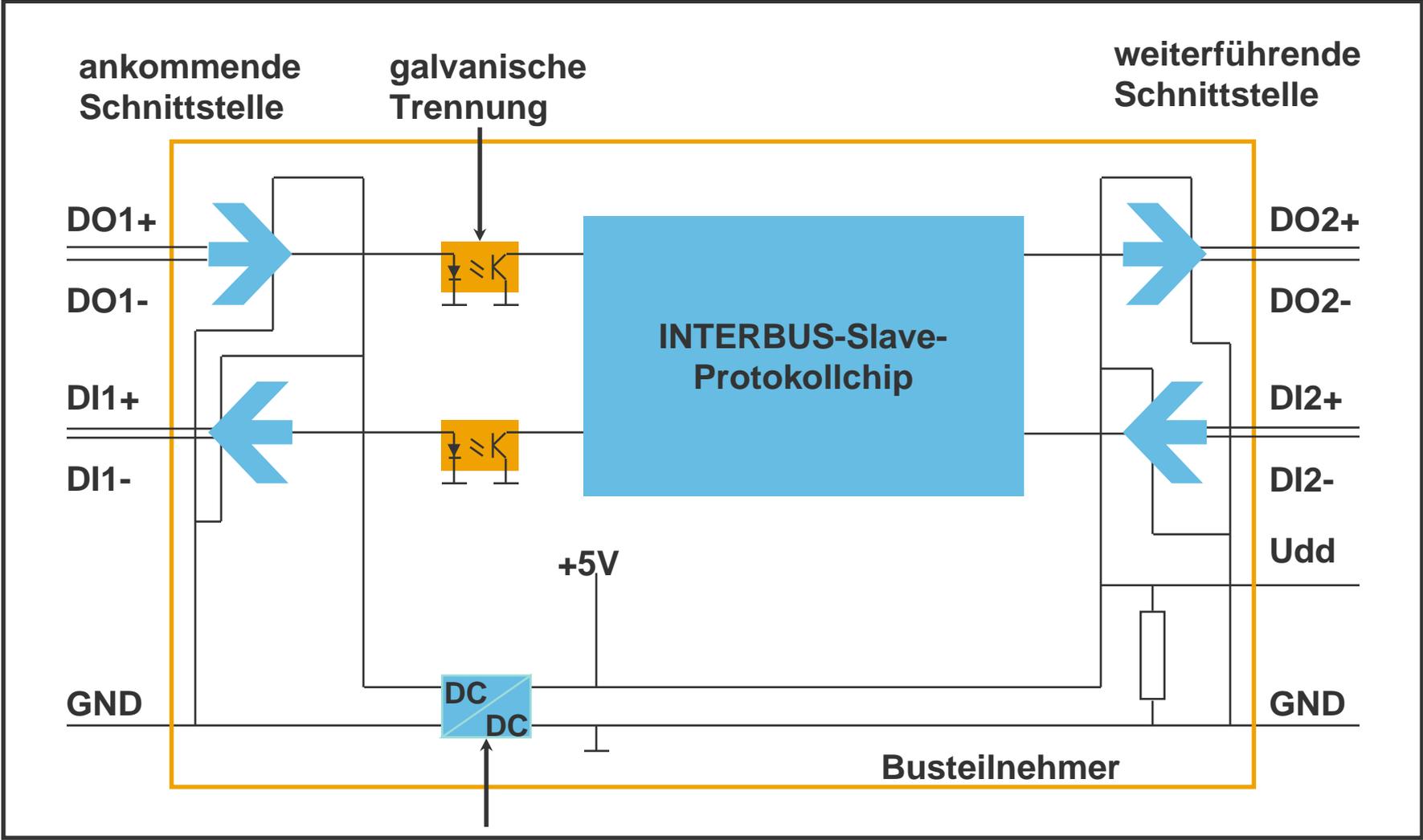


Ident.	Buslevel	Slave-Nr.
FB-Tin	0	1
BK	0	2
LB-Tin	1	3
LB-Tin	1	4
LB-Tin	1	5
LB-Tin	0	6
LB-Tin	0	7
LB-Tin	0	8

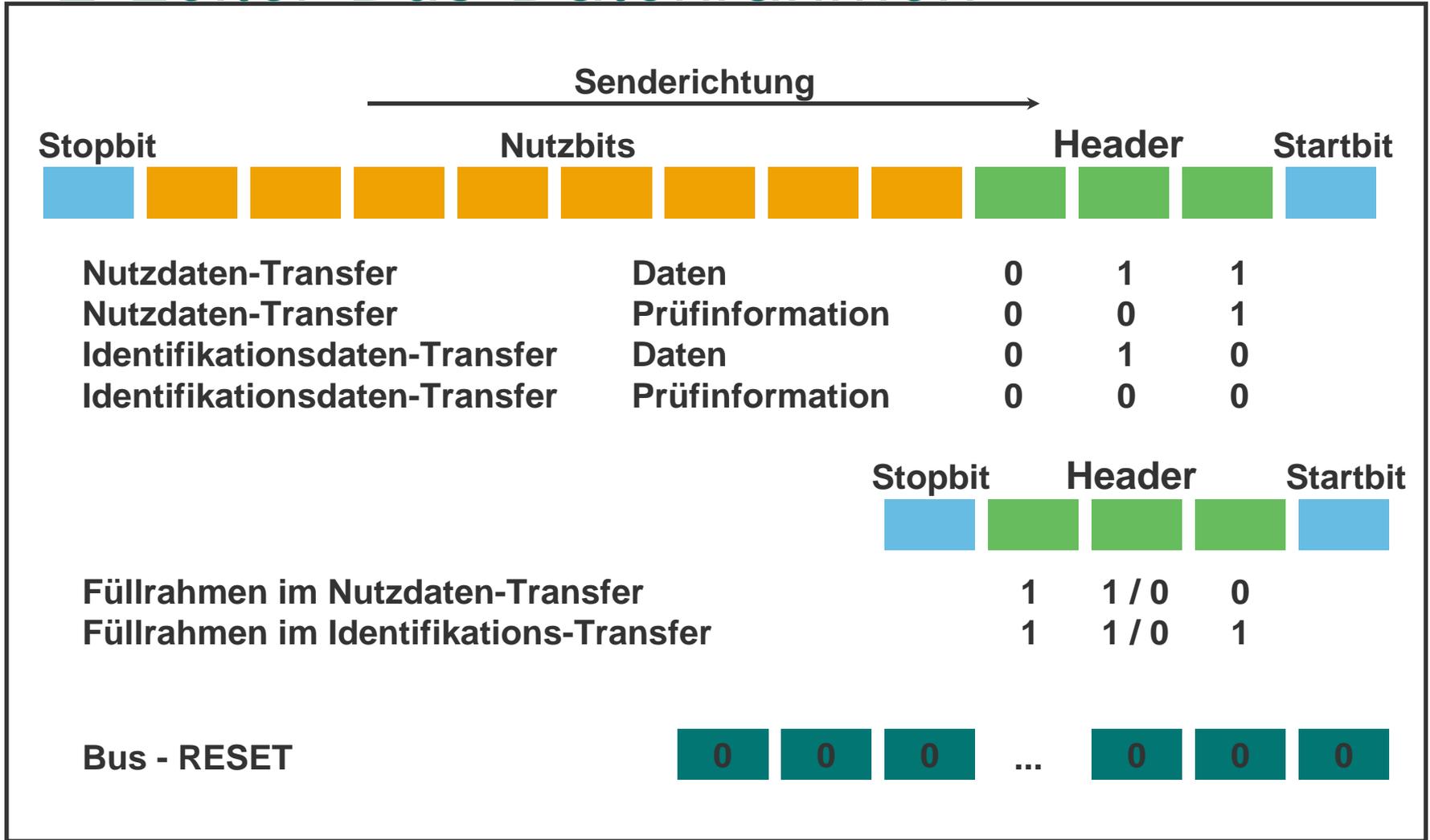
# Hardware des 2-Leiter-Fernbusses Signalschnittstelle



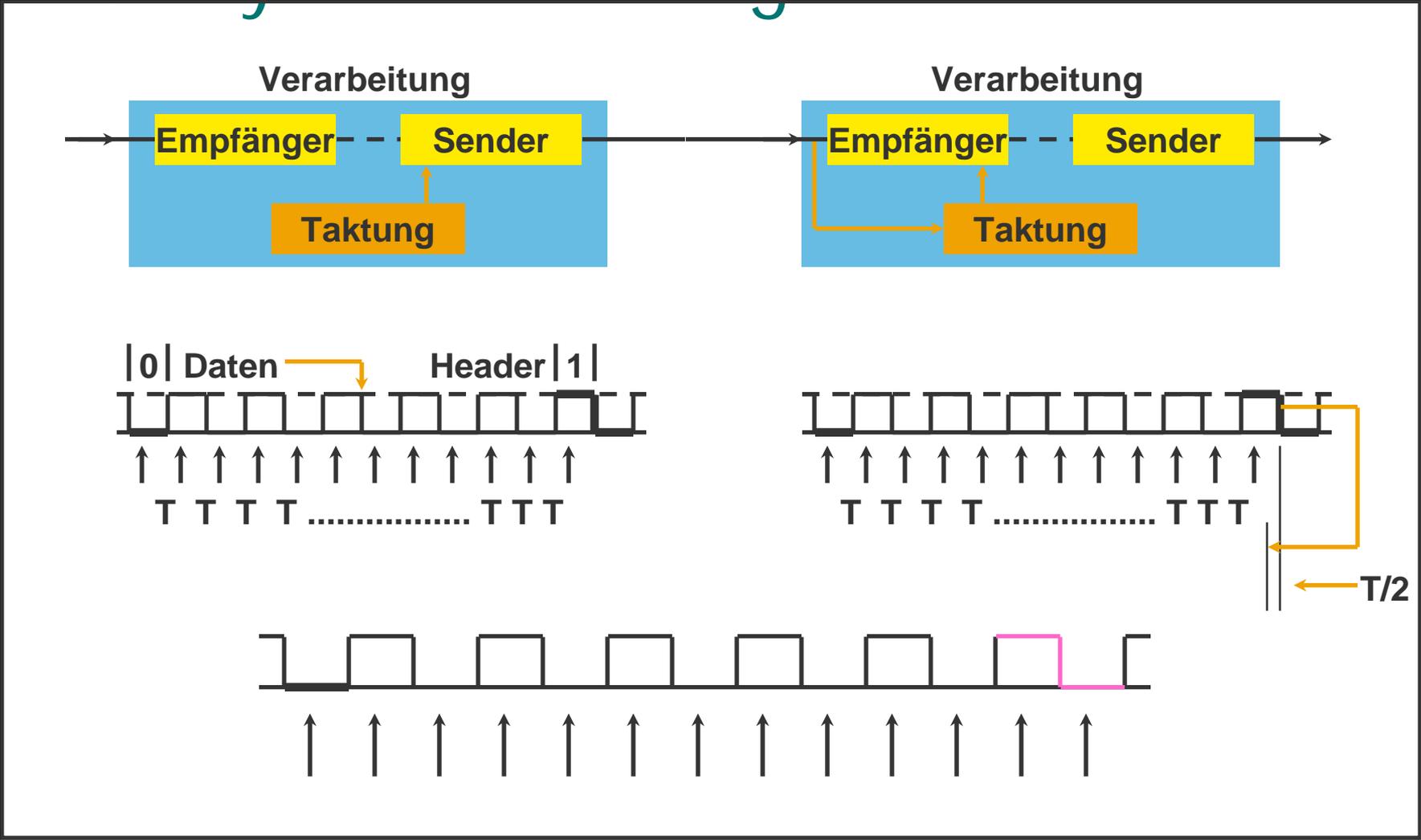
# Busschnittstelle



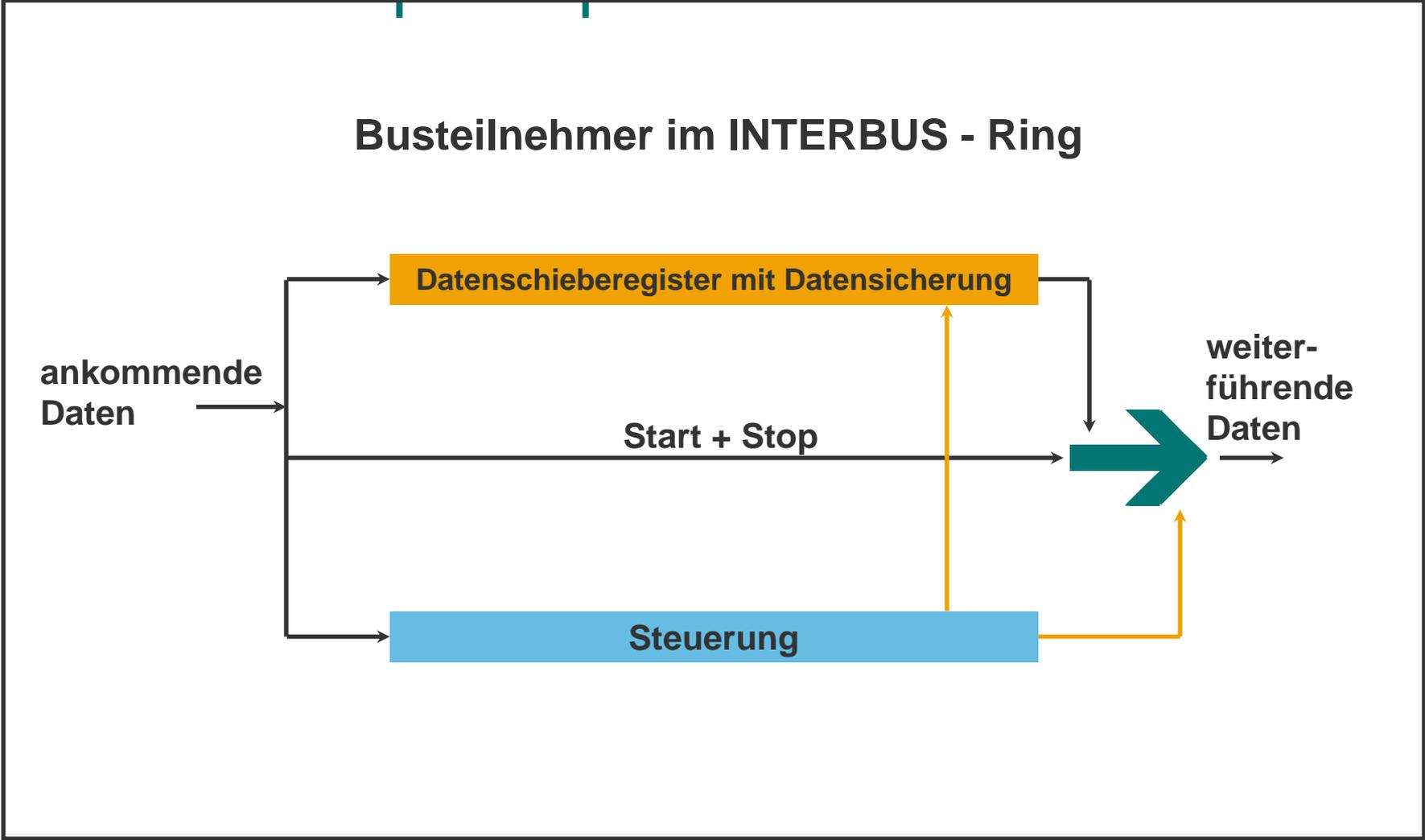
# 2-Leiter-Bus-Datenrahmen



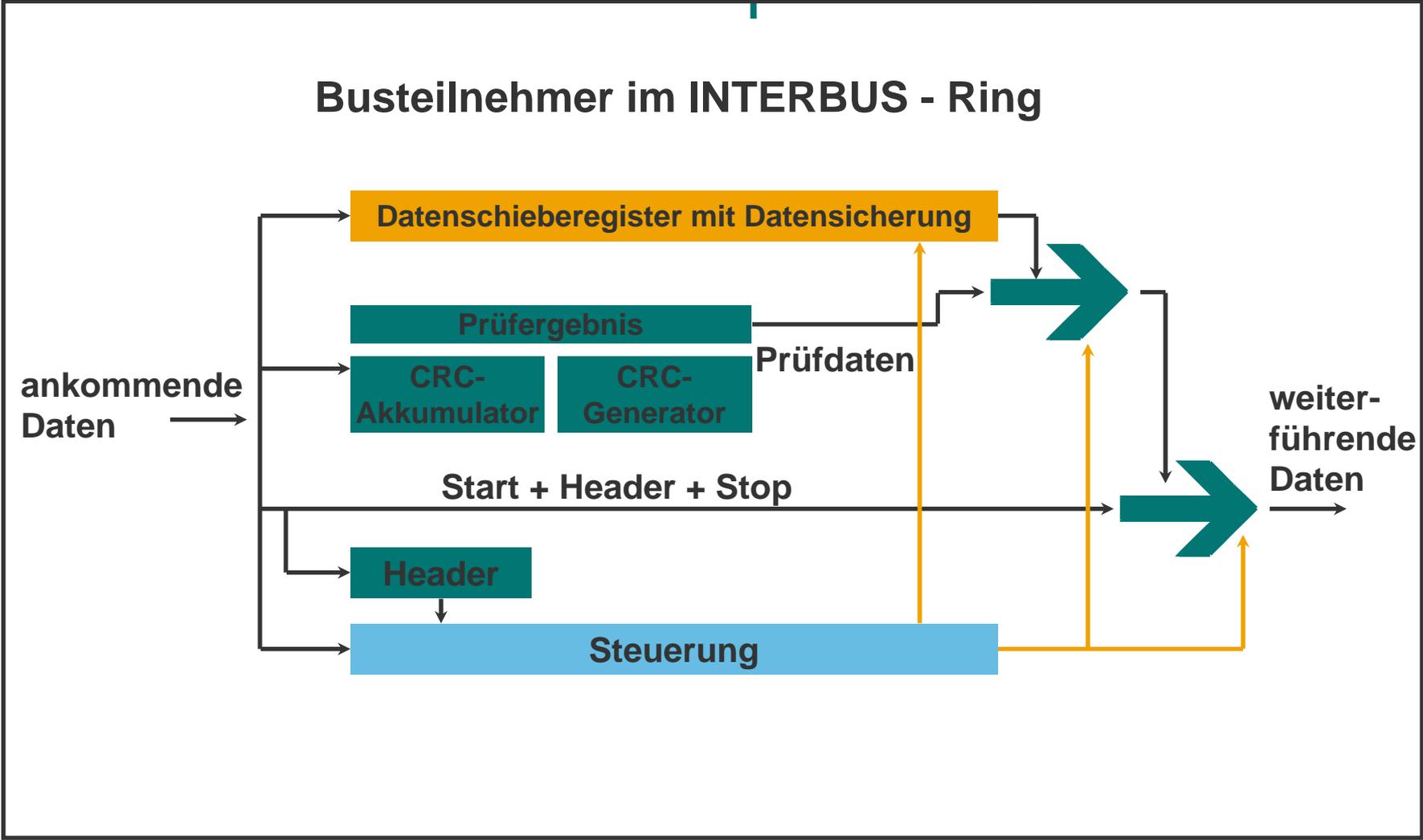
# Taktsynchronisierung



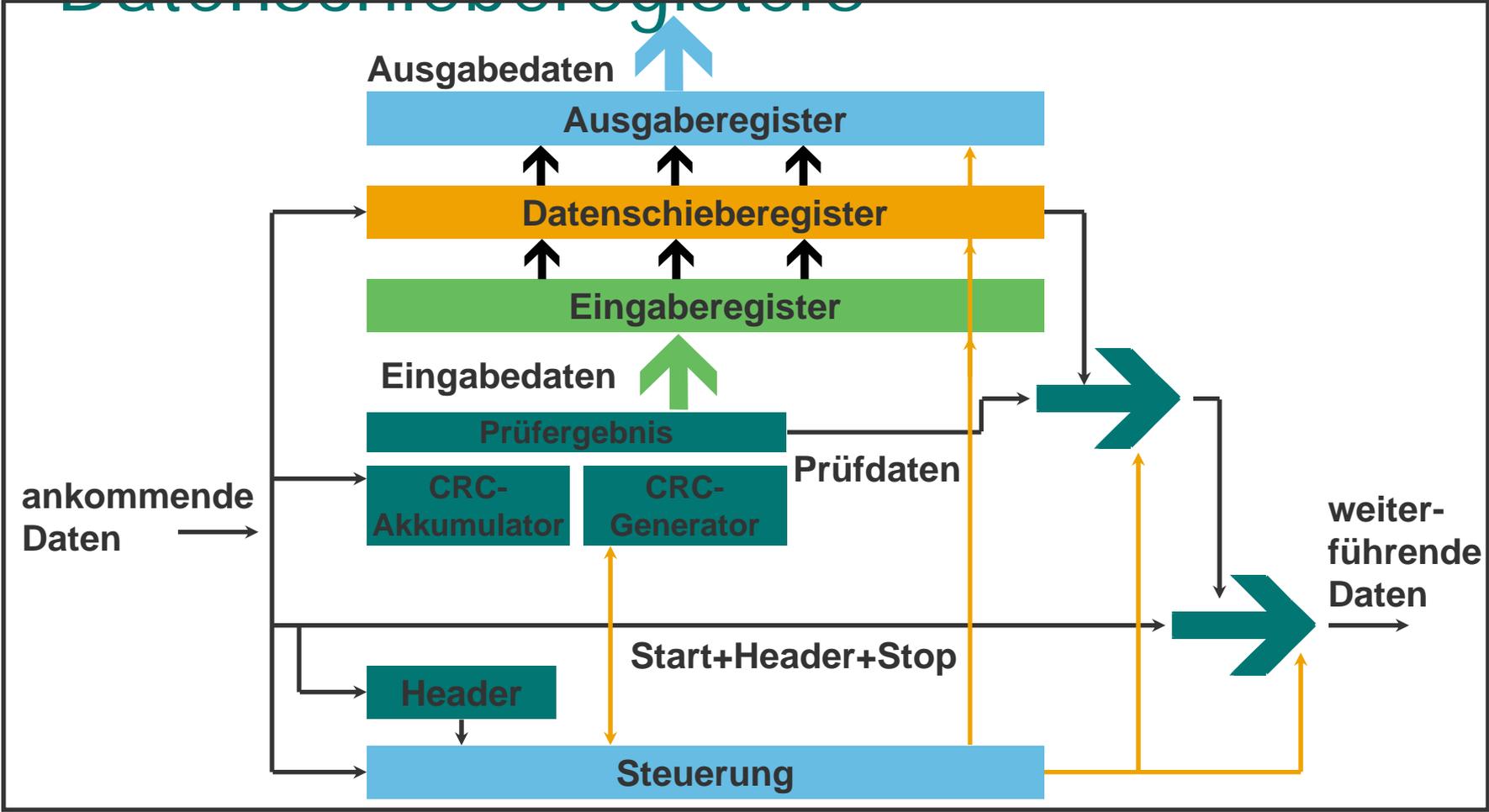
# Funktionsprinzip



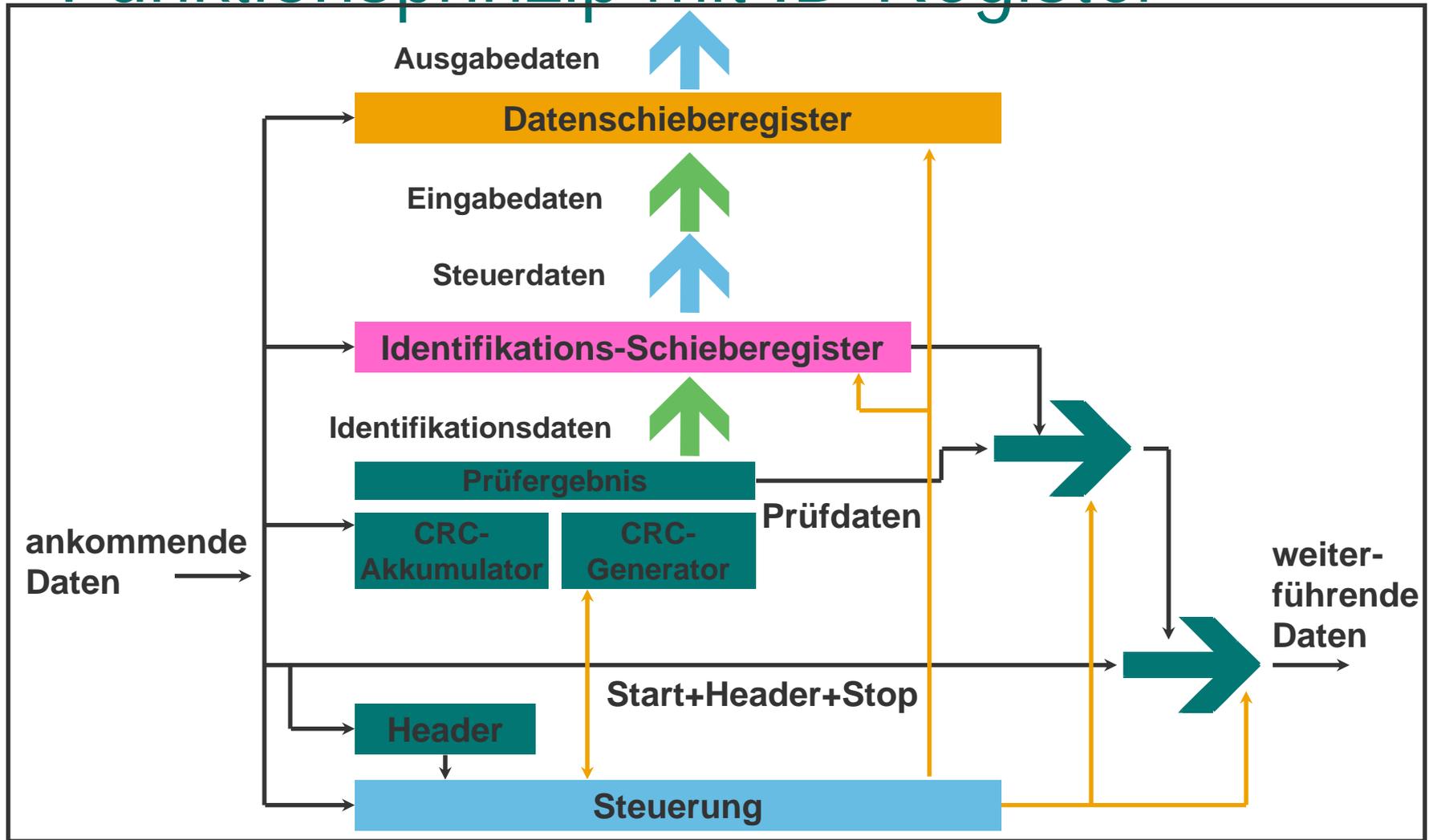
# Herausfiltern der Kopfinformation



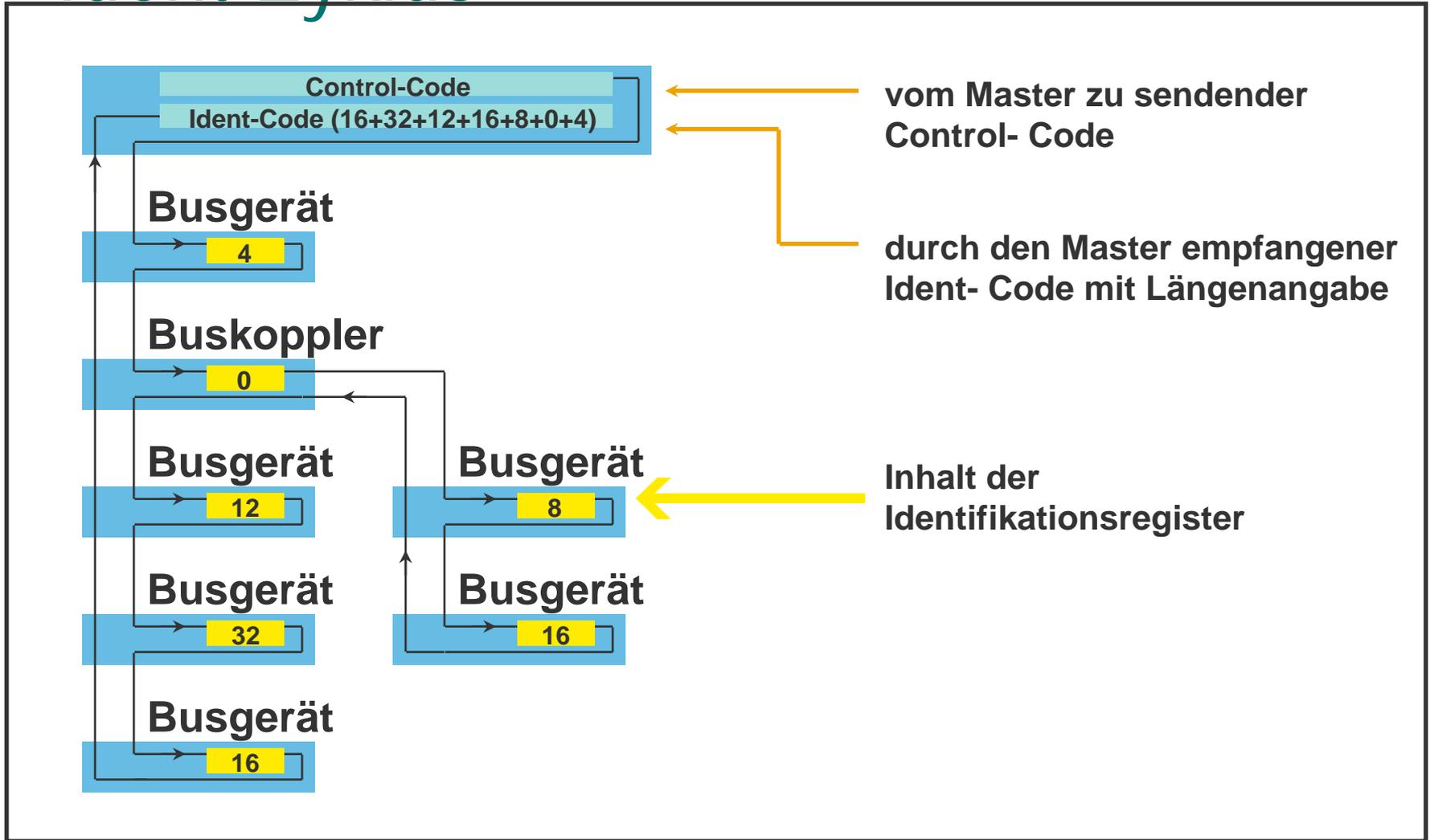
# Funktionsweise des Datenschieberegisters



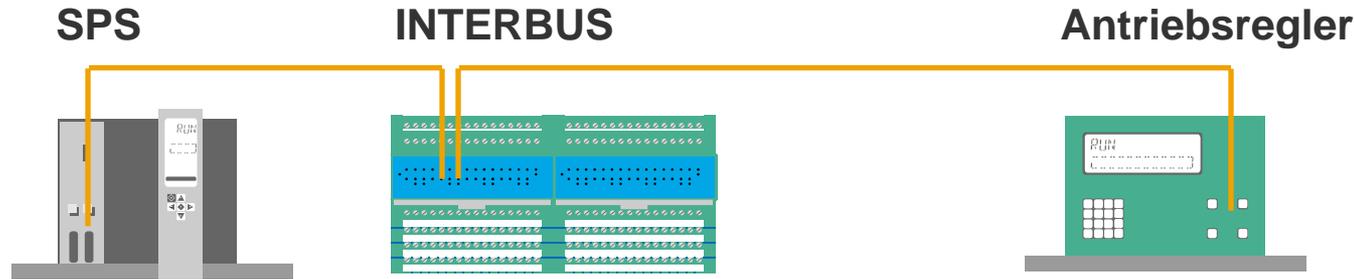
# Funktionsprinzip mit ID-Register



# Ident-Zyklus



# Parameterkanal



**Sensoren /  
Aktoren**

**Zustandsübertragung**

- Datenaustausch in jedem SPS-Zyklus
- Zyklisches Abtasten der Sensoren
- Zyklisches Aktualisieren der Aktoren
- Häufiger Austausch weniger Daten

**Nachrichtenübertragung**

- Übermittlung nur bei Bedarf (selten, aber dann ggf. viel)
- Parameterblöcke und Ereignisse (Start, Stop, ...)
- Zeitunkritisch, aber sicher

➔ **Prozeßdatenkanal**

➔ **Parameterkanal**

# Aufbau des Parametertelegramms

## Zusatzinformationen

**Dienst****Adresse****Index**

## Parameter / Programm

**Daten**

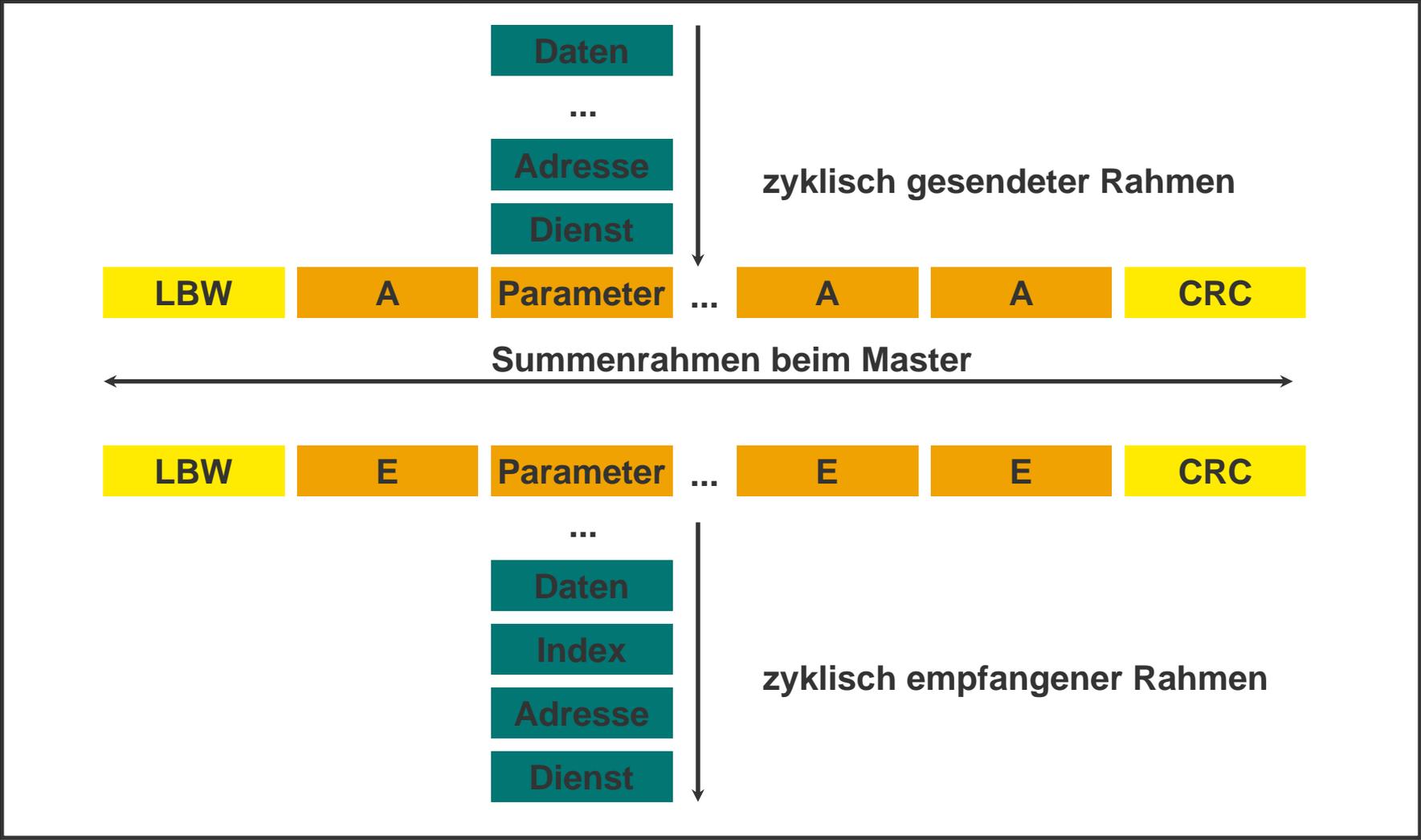
Wie werden die Daten übertragen?  
Wohin werden die Daten übertragen?  
Was bedeuten die Daten?

### Kommunikationsdienste

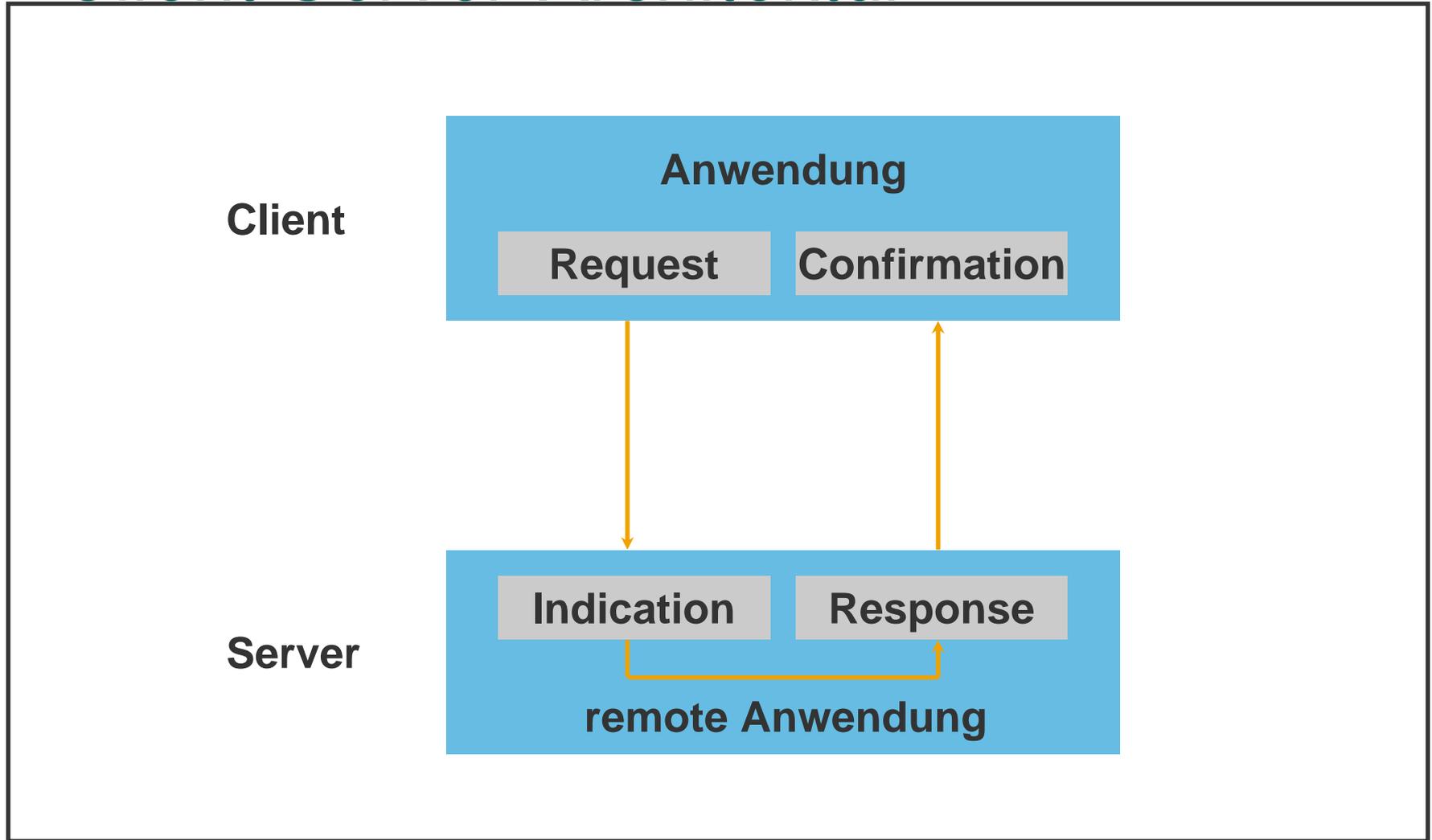
Initiate	Kommunikationsverbindung aufbauen
Aborte	Kommunikationsverbindung abbauen
Start	Programm starten
Stop	Programm stoppen
Read	Variable lesen
Write	Variable schreiben
Identify	Ident.-Information des Gerätes anfordern
Status	Status des Gerätes anfordern
Get-OV	Lesen aus Objektverzeichnis

...

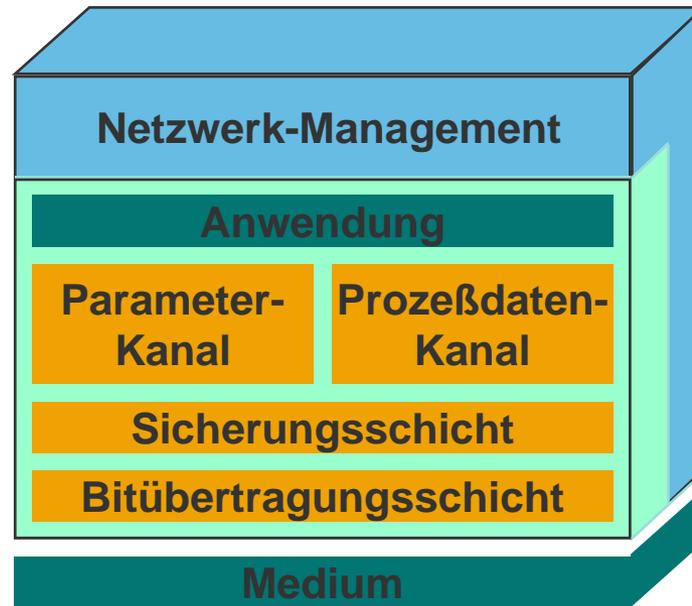
# Parameterkanal im Summenrahmen



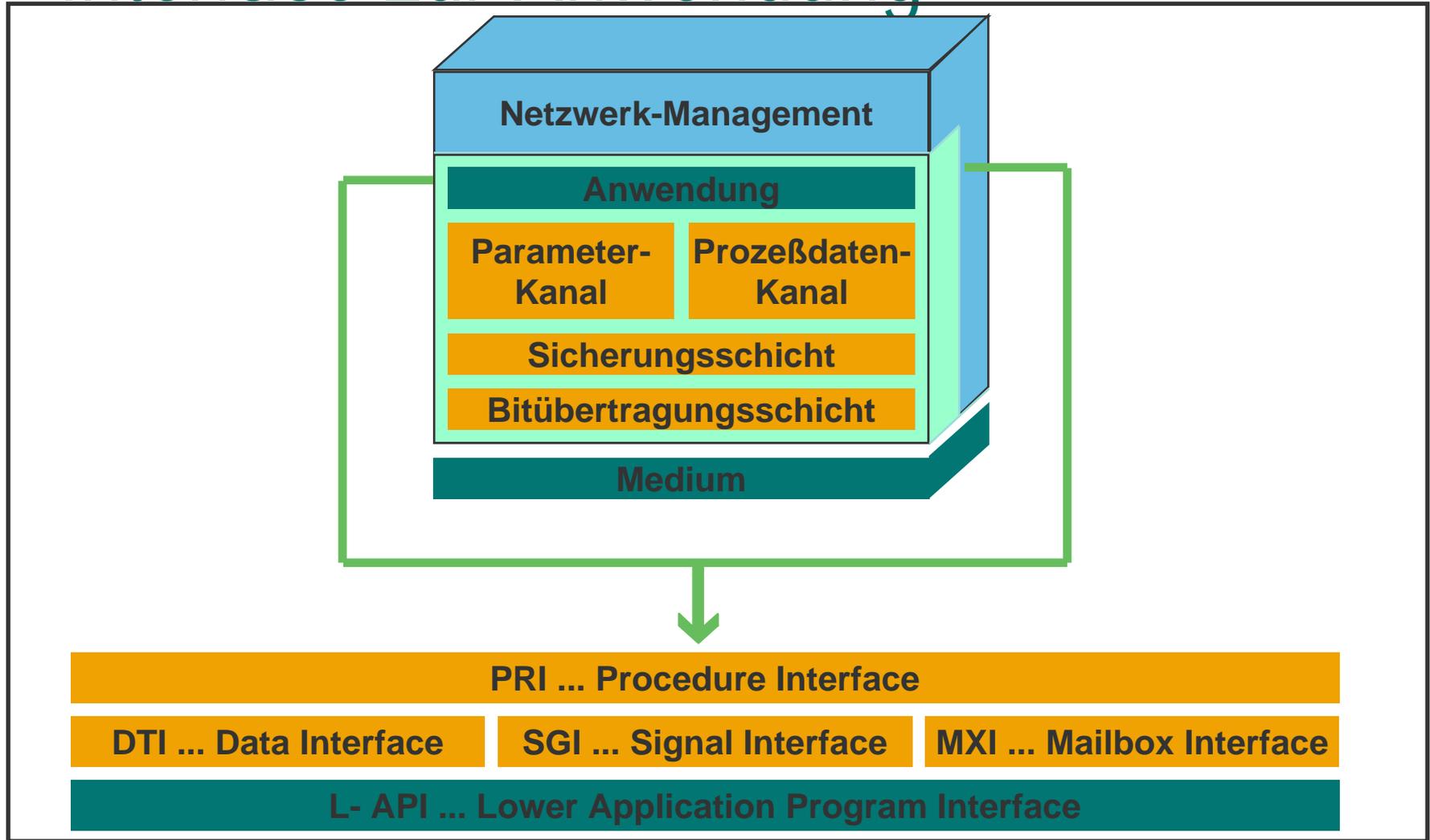
# Client-Server-Architektur



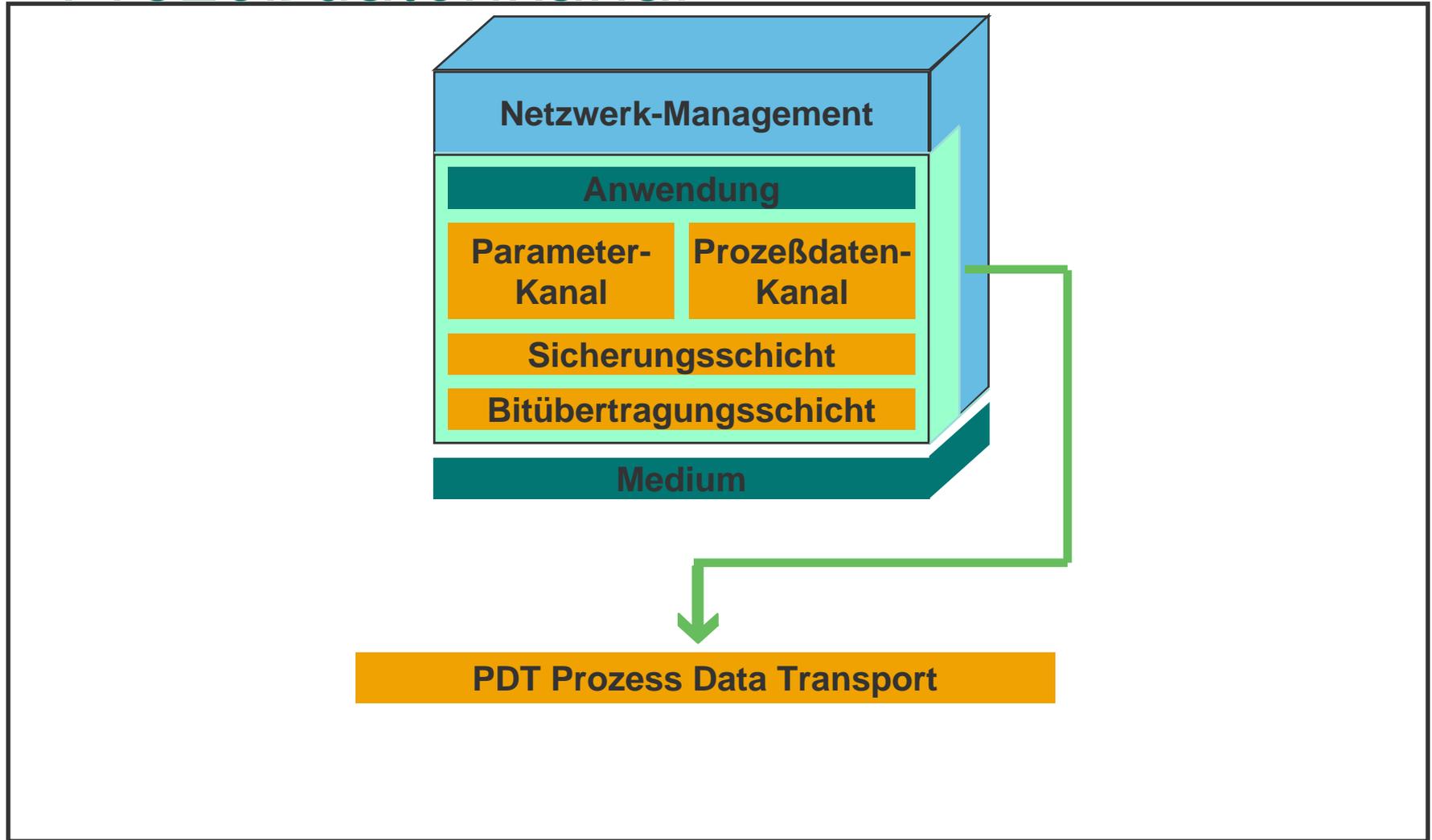
# INTERBUS-Protokoll



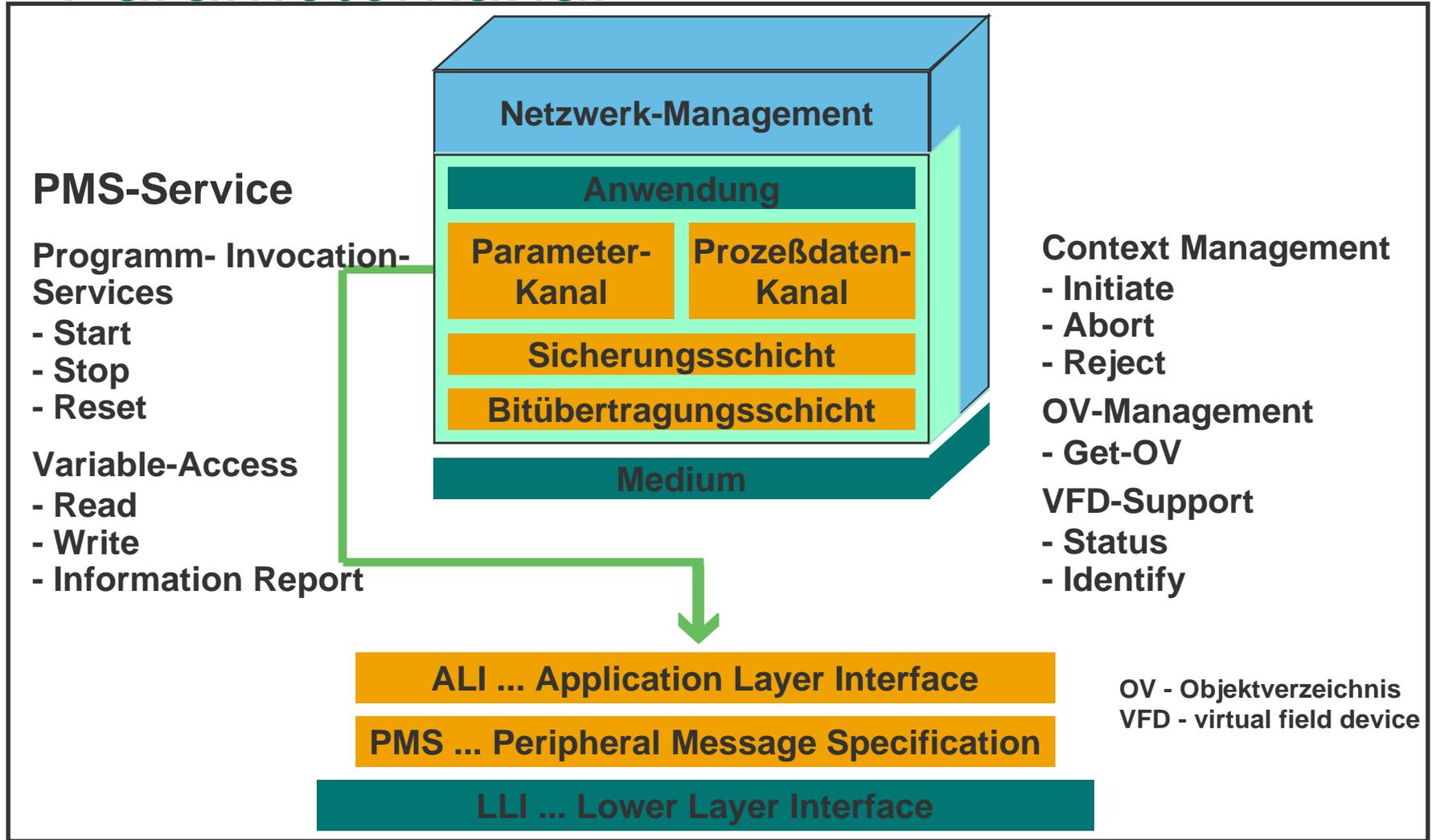
# Interface zur Anwendung



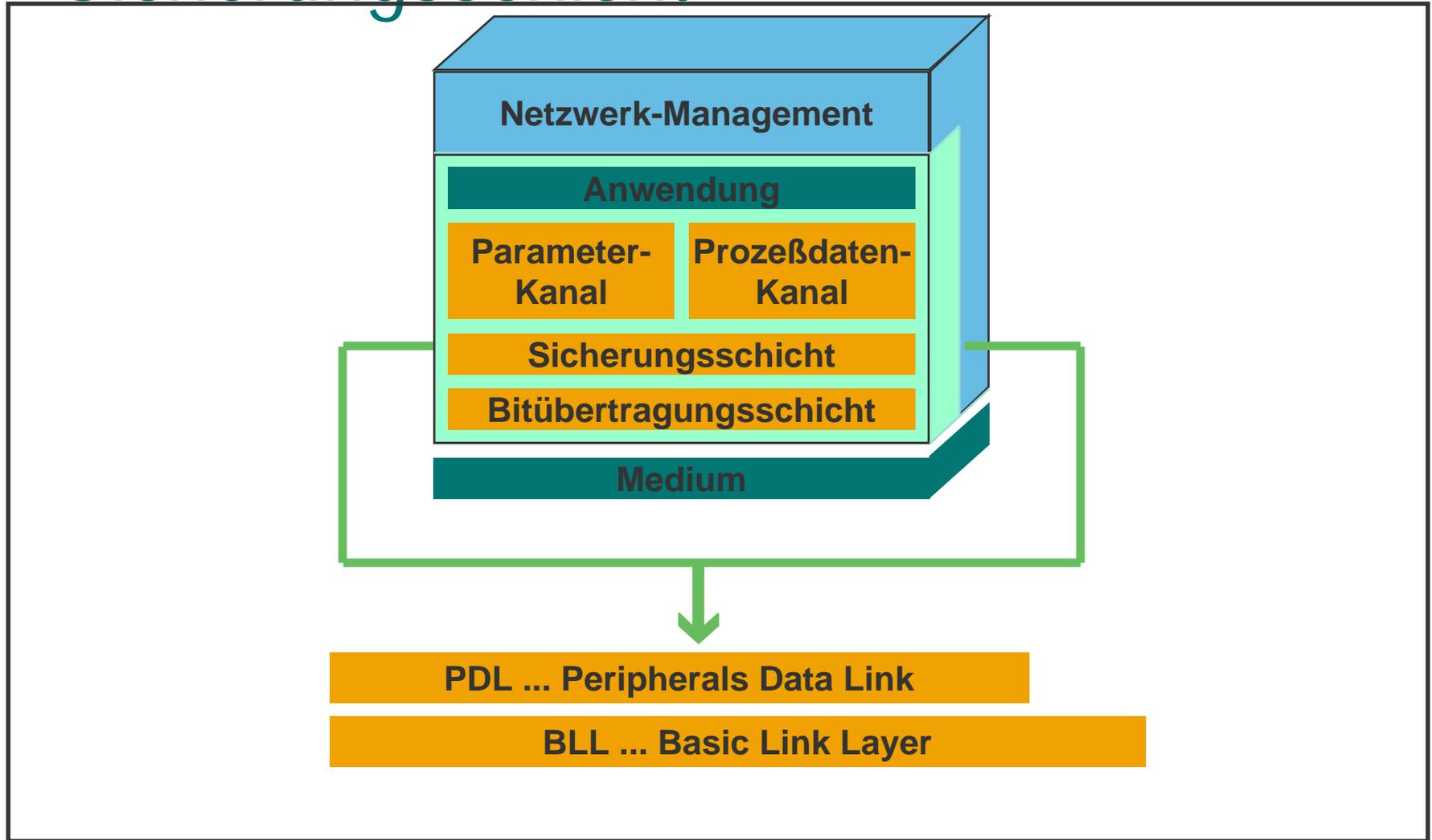
# Prozeßdatenkanal



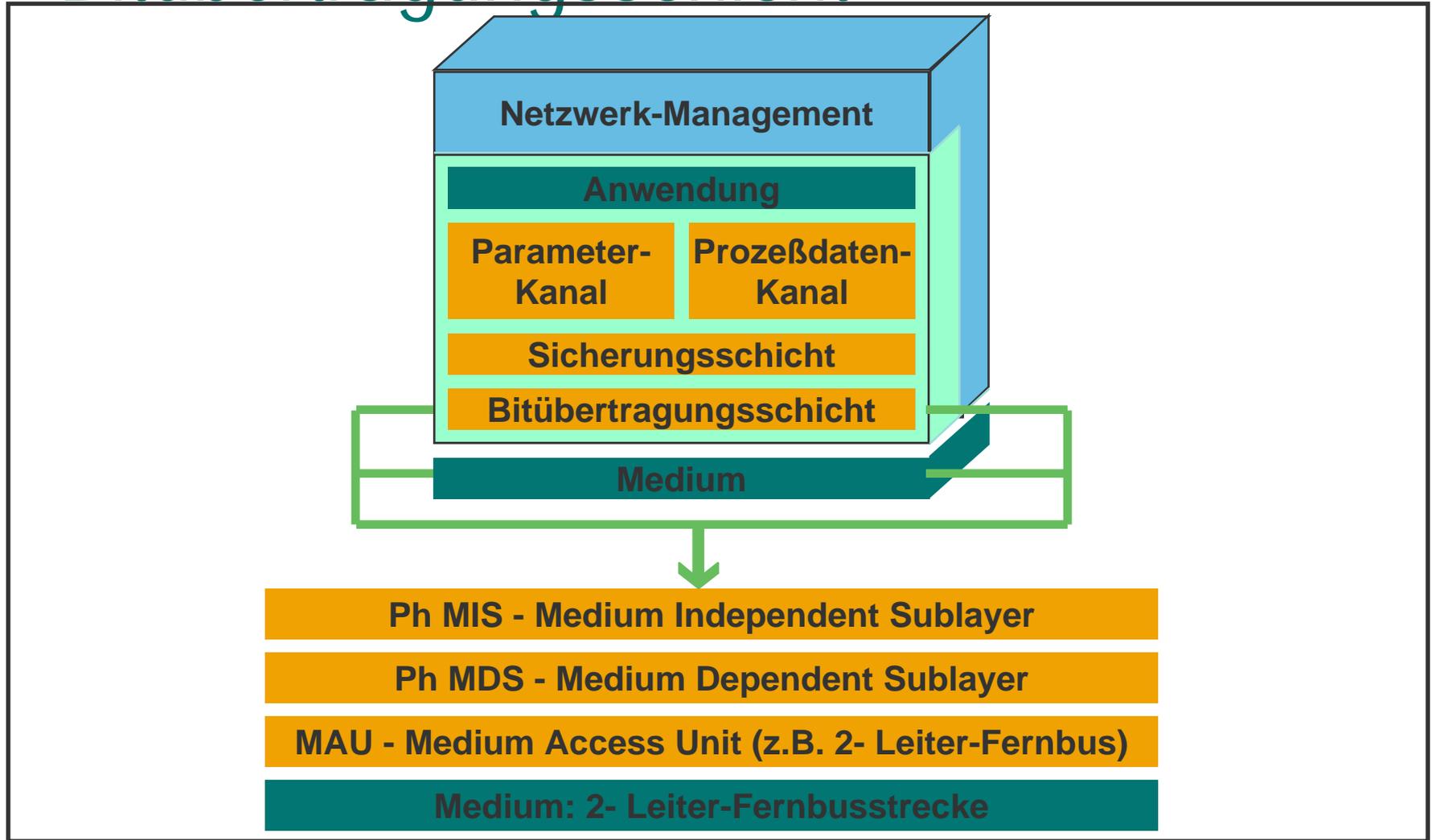
# Parameterkanal



# Sicherungsschicht



# Bitübertragungsschicht



# Netzwerkmanagement

