DHBW Stuttgart

Industrielle Bussysteme : Modbus/TCP

Dr. Leonhard Stiegler
Automation

www.dhbw-stuttgart.de

Inhalt

DHBW Stuttgart

Modbus/TCP

- Grundsätze und Versionen
- Protokollbeschreibung
- Datenmodell und Datencodierung
- Adressierung und Transaktionen
- Function Codes

DHBW Stuttgart

Grundlegende Merkmale

- Modbus Messaging-Protokoll auf der Anwendungsschicht
- Modbus/TCP: Übertragung über TCP-IP-Ethernet
- Client-Server-Kommunikation
- Unterstützung unterschiedlicher Bus- und Netz-Topologien
- De-Facto-Standard in der industriellen Automation
- Einsatz seit 1979

DHBW Stuttgart

Erste Definition

- Protokoll für Modicon Programmable Logic Controller PLC
- Verfügbar seit 1979
- Entwicklung durch Schneider Electric

Weiterentwicklung und Pflege

- Modbus Organization seit 2004
- Vereinigung von Modbus-Nutzern und -Herstellern

Modbus : Übertragungsvarianten

DHBW Stuttgart

Modbus/TCP

- Übermittlung : TCP/IP über Ethernet

Asynchrone serielle Übertragung

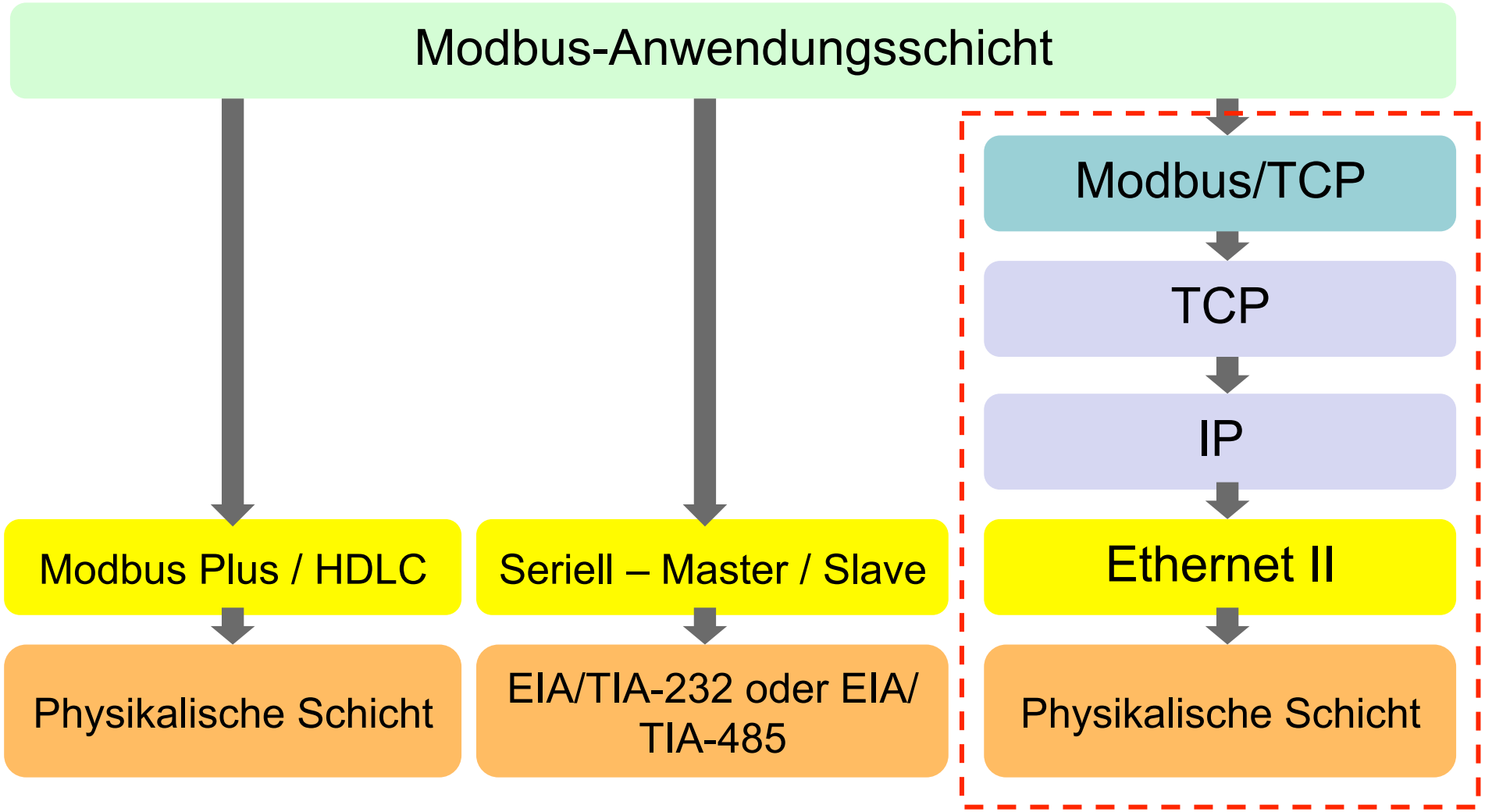
- Nutzung unterschiedlicher Medien
- drahtgebunden: EIA/TIA-232-E, EIA-422, EIA/TIA-485-A

Modbus Plus

- eigenständiges Protokoll für Token-Passing-Netze

Modbus : Variantenübersicht

DHBW Stuttgart



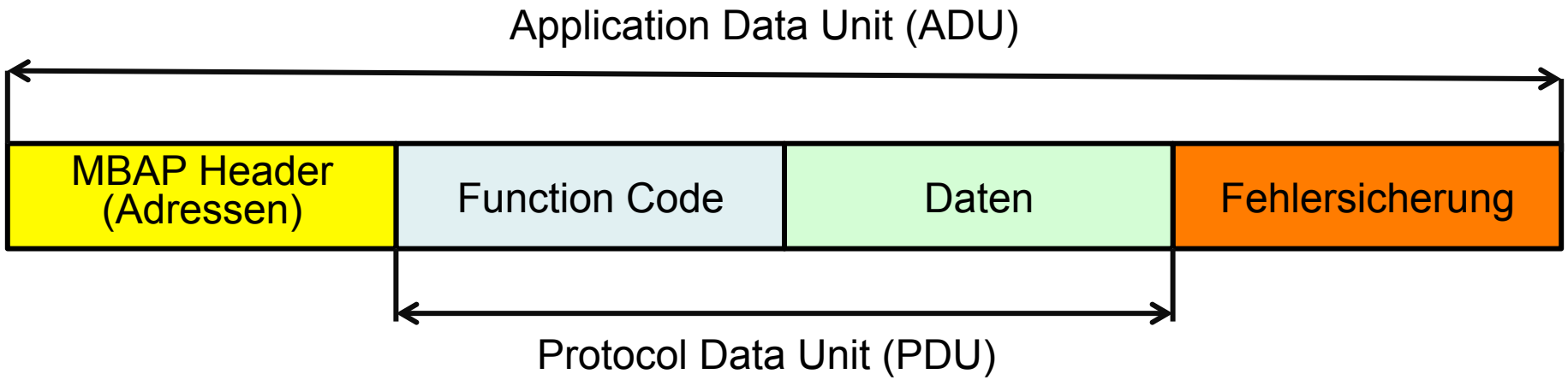
Inhalt

DHBW Stuttgart

Modbus/TCP

- Grundsätze und Versionen
- Protokollbeschreibung
- Datenmodell und Datencodierung
- Adressierung und Transaktionen
- Function Codes

Modbus : PDU und ADU



PDU: unabhängig von den genutzten Protokollschichten

ADU: Anpassung an die genutzten Protokollschichten

Function Code (1 Byte): Definition der auszuführenden Aktion

Modbus Rollen

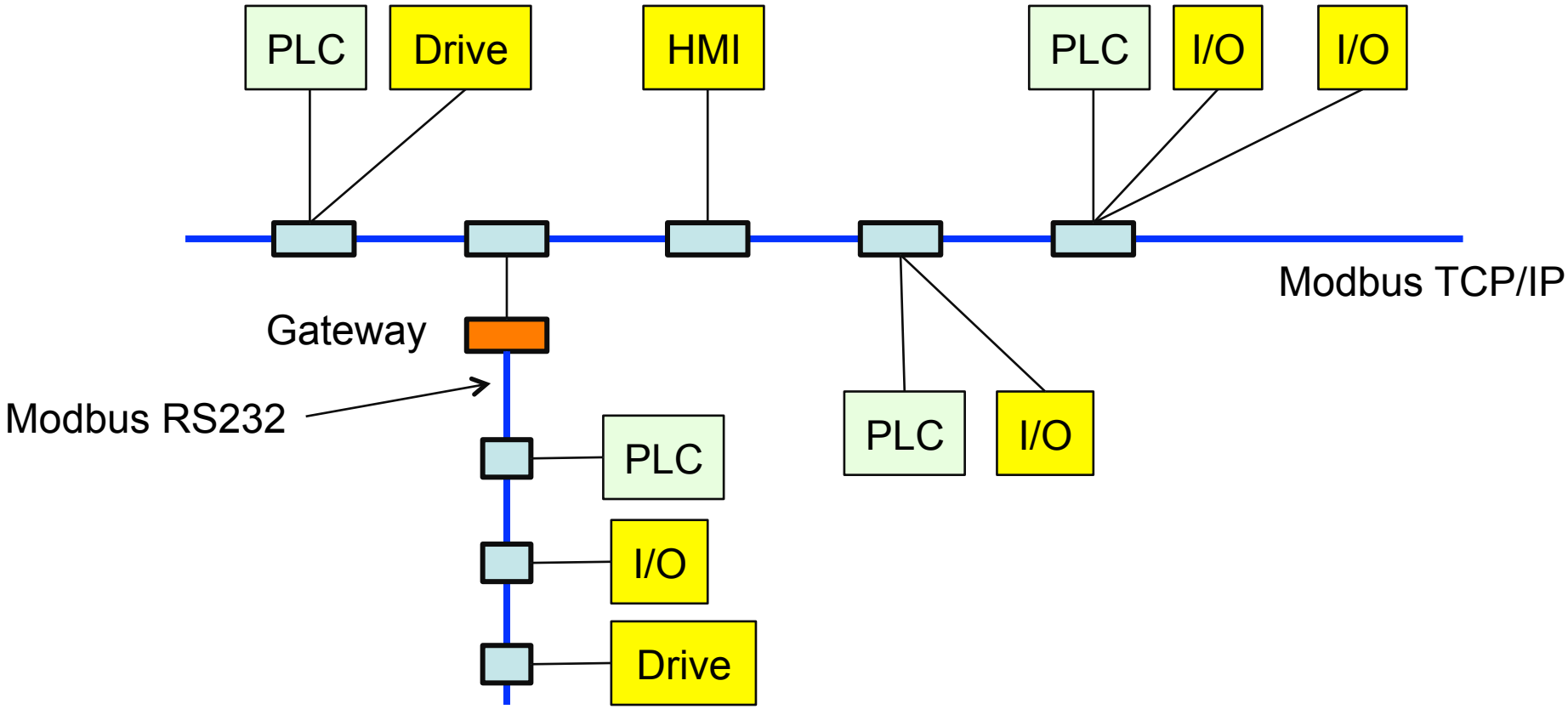
- **Modbus Client**
  - ermöglicht die Komunikation mit einem Remote Device
  - erzeugt Modbus Request-Nachrichten gesteuert durch die Anwendung
  - überträgt Modbus Request-Nachrichten an das Modbus Client Interface
- **Modbus Server**
  - Wartet auf Modbus Request-Nachrichten (TCP-Port 502)
  - Liest und verarbeitet Modbus Request-Nachrichten
  - erzeugt Modbus Response-Nachrichten
- **Modbus User Application (Backend) Interface**
  - Schnittstelle zwischen Server und User Application

Modbus AP Header

Application Protocol Header Parameter

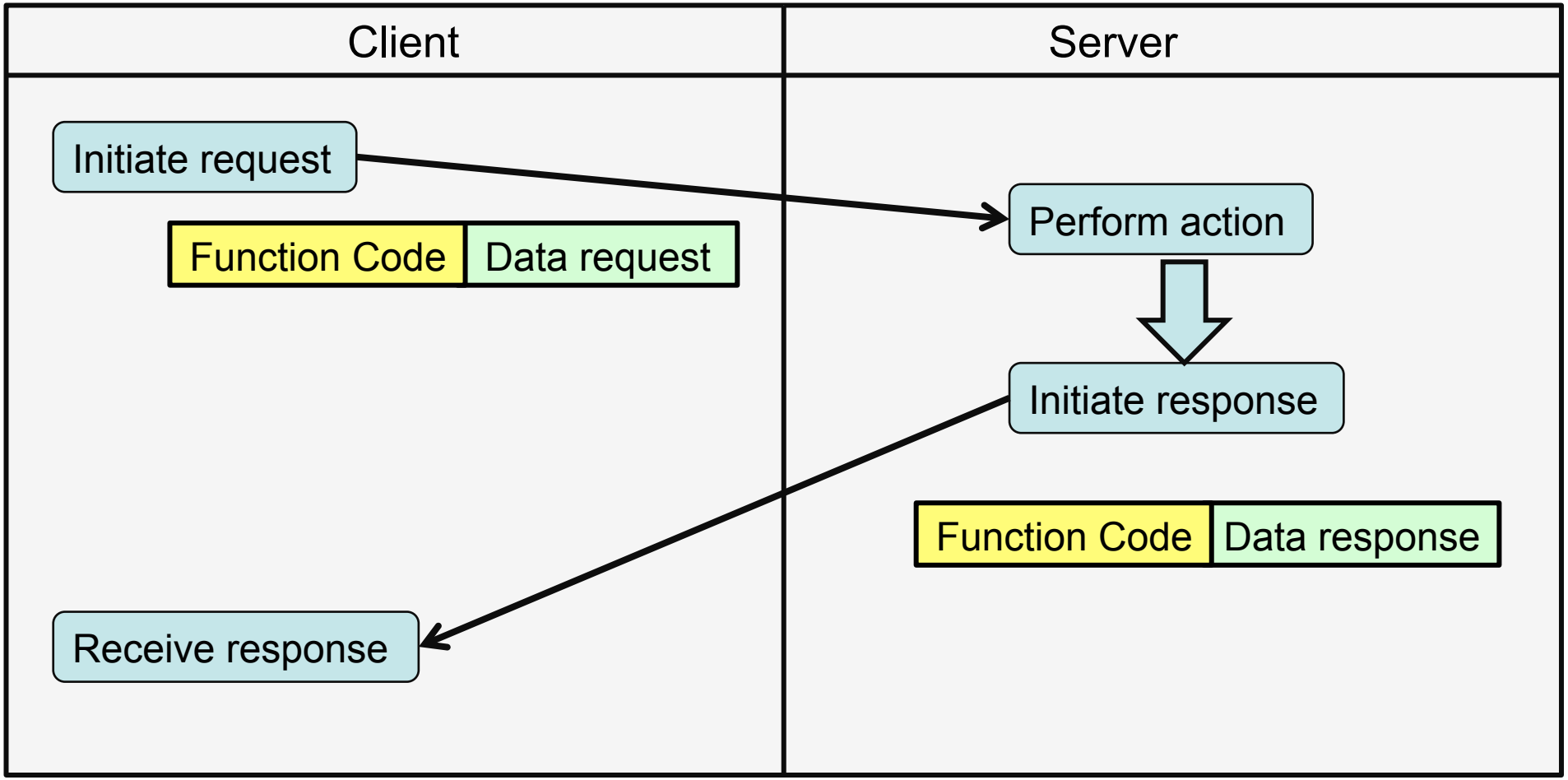
Fields	Length	Description -	Client	Server
Transaction Identifier	2 Bytes	Identification of a MODBUS Request / Response transaction.	Initialized by the client	Recopied by the server from the received request
Protocol Identifier	2 Bytes	0 = MODBUS protocol	Initialized by the client	Recopied by the server from the received request
Length	2 Bytes	Number of following bytes	Initialized by the client (request)	Initialized by the server (Response)
Unit Identifier	1 Byte	Identification of a remote slave connected on a serial line or on other buses.	Initialized by the client	Recopied by the server from the received request

Modbus Kommunikation



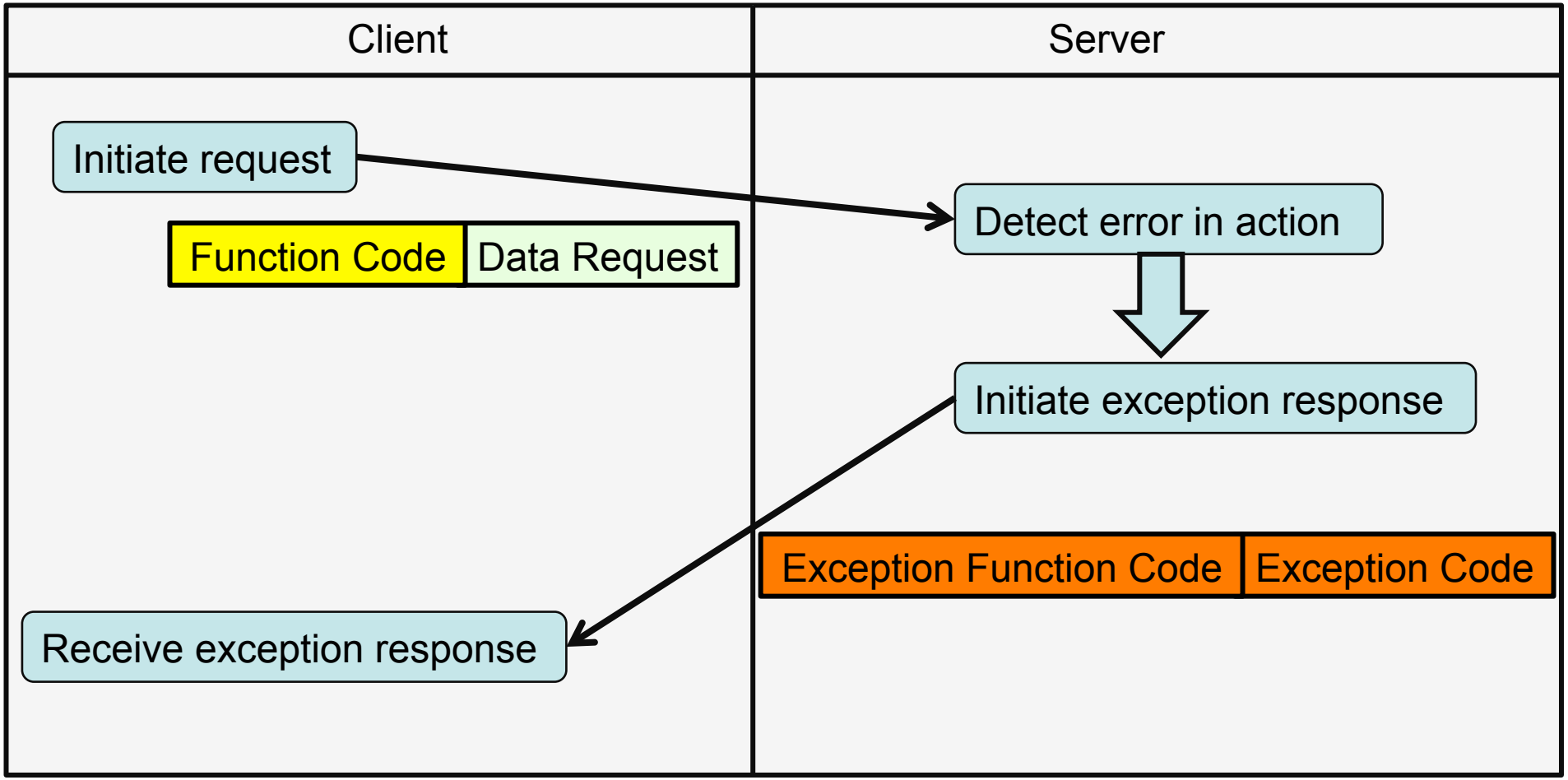
DHBW Stuttgart

Schema der fehlerfreien Client-Server-Kommunikation



DHBW Stuttgart

Schema der fehlerhaften Client-Server-Kommunikation



Modbus : Typen von PDUs

DHBW Stuttgart

Modbus Request PDU

- mb_req_pdu = {function_code, request_data}

Modbus Response PDU

- mb_rsp_pdu = {function_code, response_data}

Modbus Exception Response PDU

- mb_excep_rsp_pdu = {exception-function_code, request_data}

DHBW Stuttgart

Längenbegrenzung für Modbus ADUs und PDUs

- ADU : bestimmt durch erste RS-485-Implementierung
- RS-485-ADU : maximal 256 Byte
- damit Modbus-PDU : max. 253 Byte : 256 Byte – 1 Byte Server-Adresse – 2 Byte CRC-Fehlersicherung
- Modbus/TCP-PDU : max. 260 Byte : 253 Byte PDU + 6 Byte MBAP (ModBus Application Protocol)

Inhalt

DHBW Stuttgart

Modbus/TCP

- Grundsätze und Versionen
- Protokollbeschreibung
- Datenmodell und Datencodierung
- Adressierung und Transaktionen
- Function Codes

Primäre Tabellen

DHBW Stuttgart

Modbus Datenformate

Name	Größe	Zugangsart	veränderbar durch
Discrete Input	1 Bit	Read Only	Management
Coils	1 Bit	Read-Write	Application
Input Registers	16 Bit	Read Only	Management
Holding Registers	16 Bit	Read-Write	Application

Implementierung der Tabellen

DHBW Stuttgart

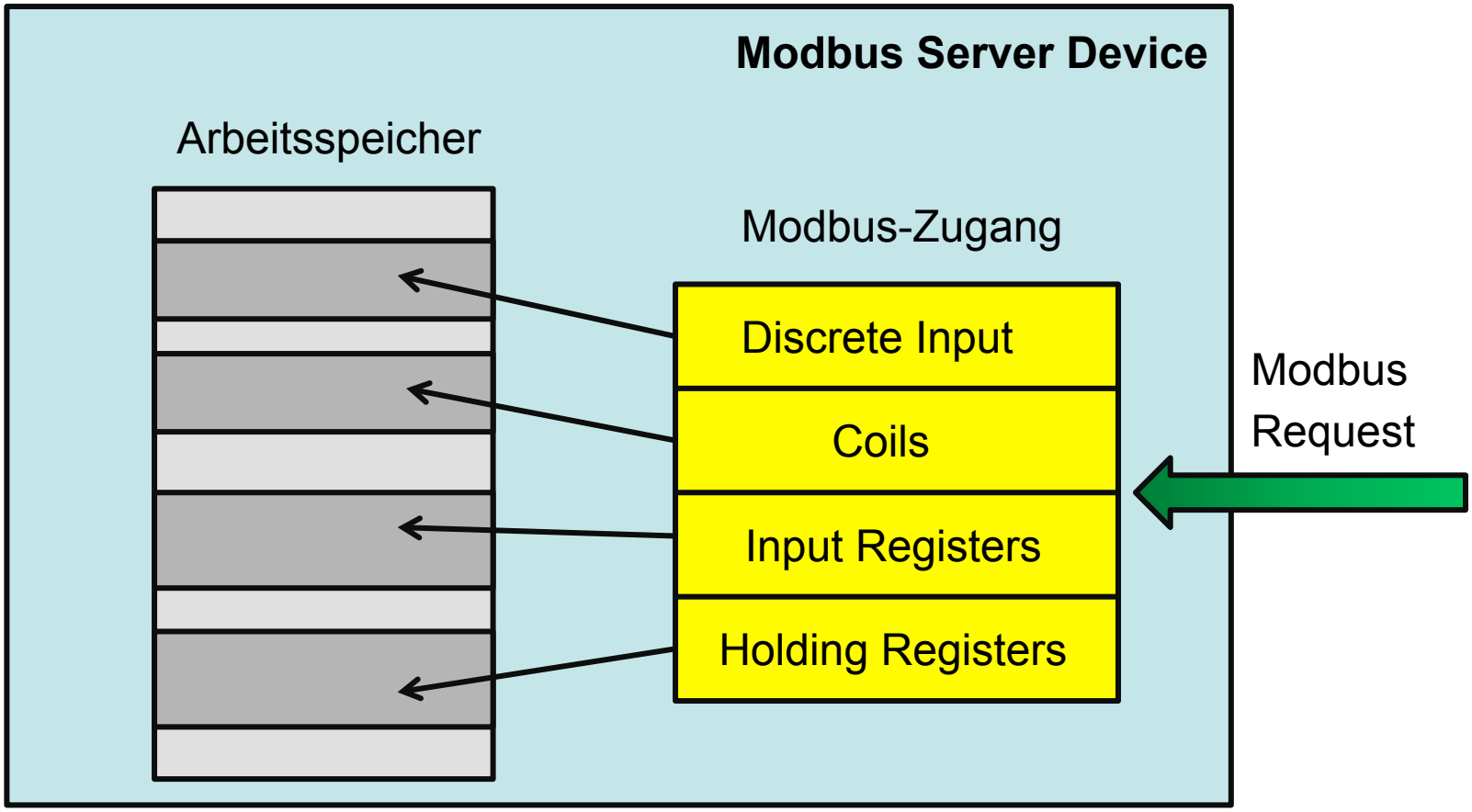
Speichern der Tabellen im Arbeitsspeicher der Modbus-Geräte

- Zugriff mit Umsetzung der logischen Modbus-Referenzen auf Adressen des Arbeitsspeichers
- Auswahl aus maximal 65 536 Daten pro Tabellenart

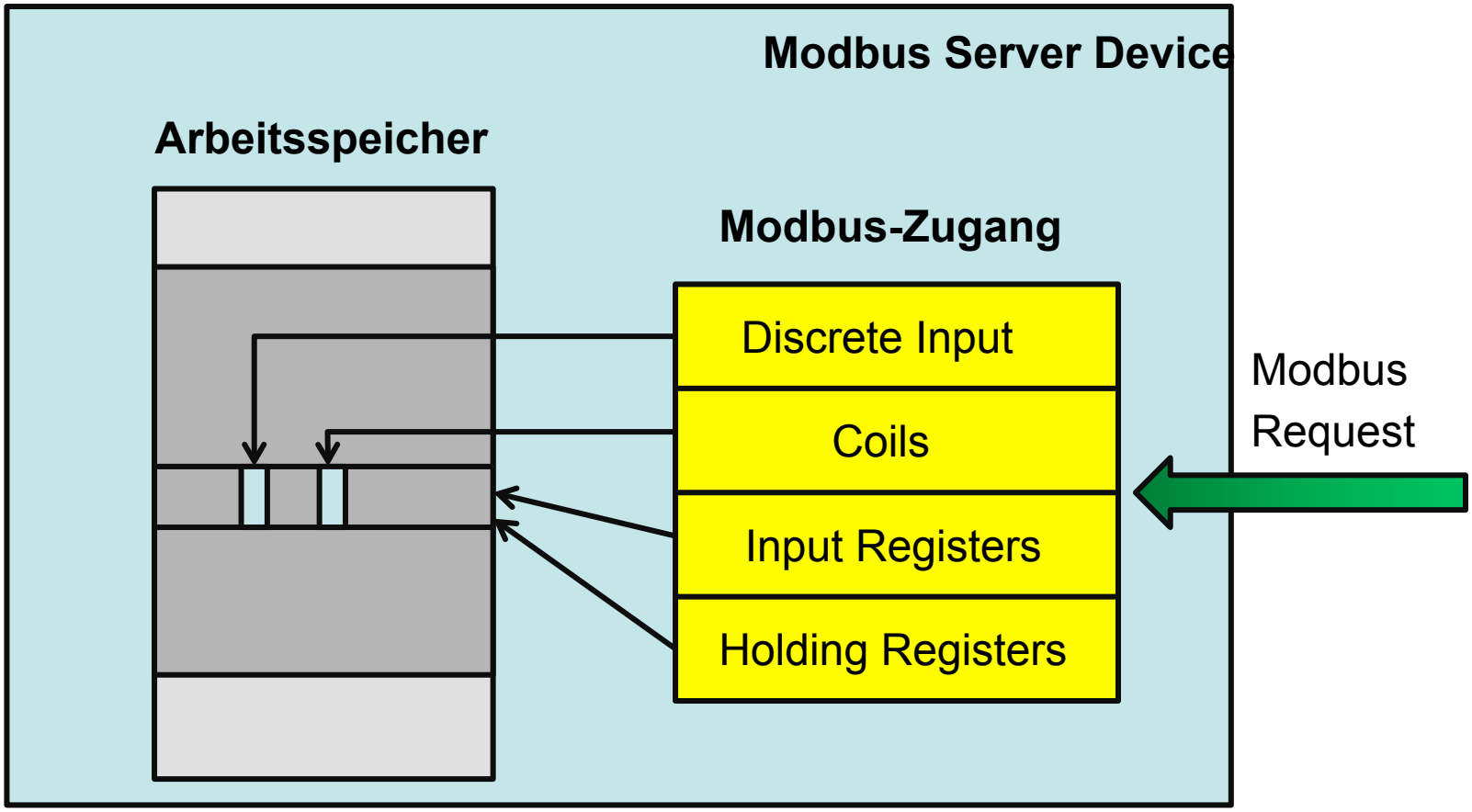
Bedarfsgerechte Speicherorganisation im Modbus-Gerät

- **Option** : getrennte Speicherbereiche pro Tabellenart
- **Option** : ein Speicherbereich mit überlappendem Zugriff

Getrennte Datenblöcke



Ein Datenblock mit Überlappung



DHBW Stuttgart

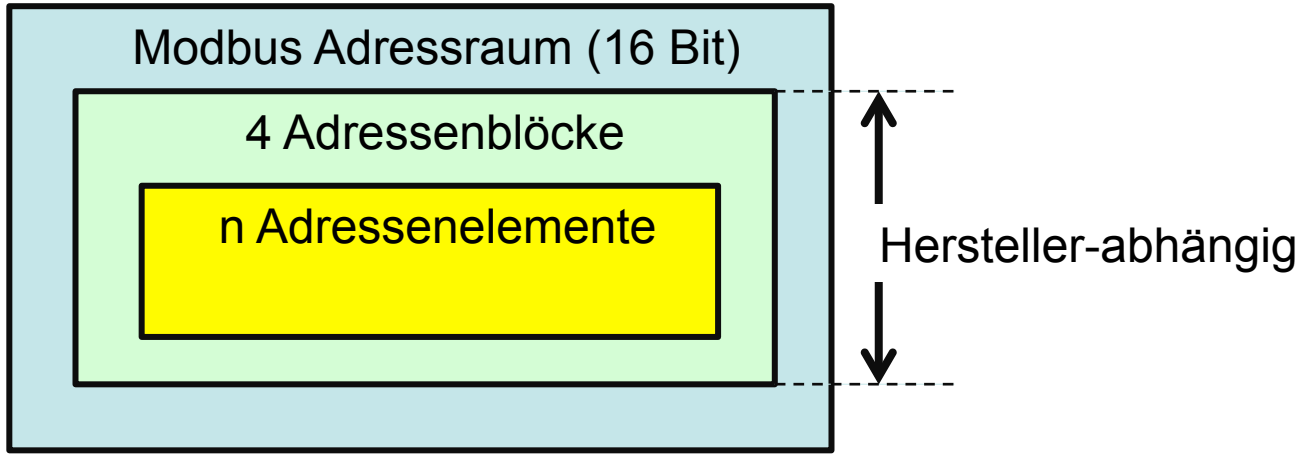
Modbus/TCP

- Grundsätze und Versionen
- Protokollbeschreibung
- Datenmodell und Datencodierung
- Adressierung und Transaktionen
- Function Codes

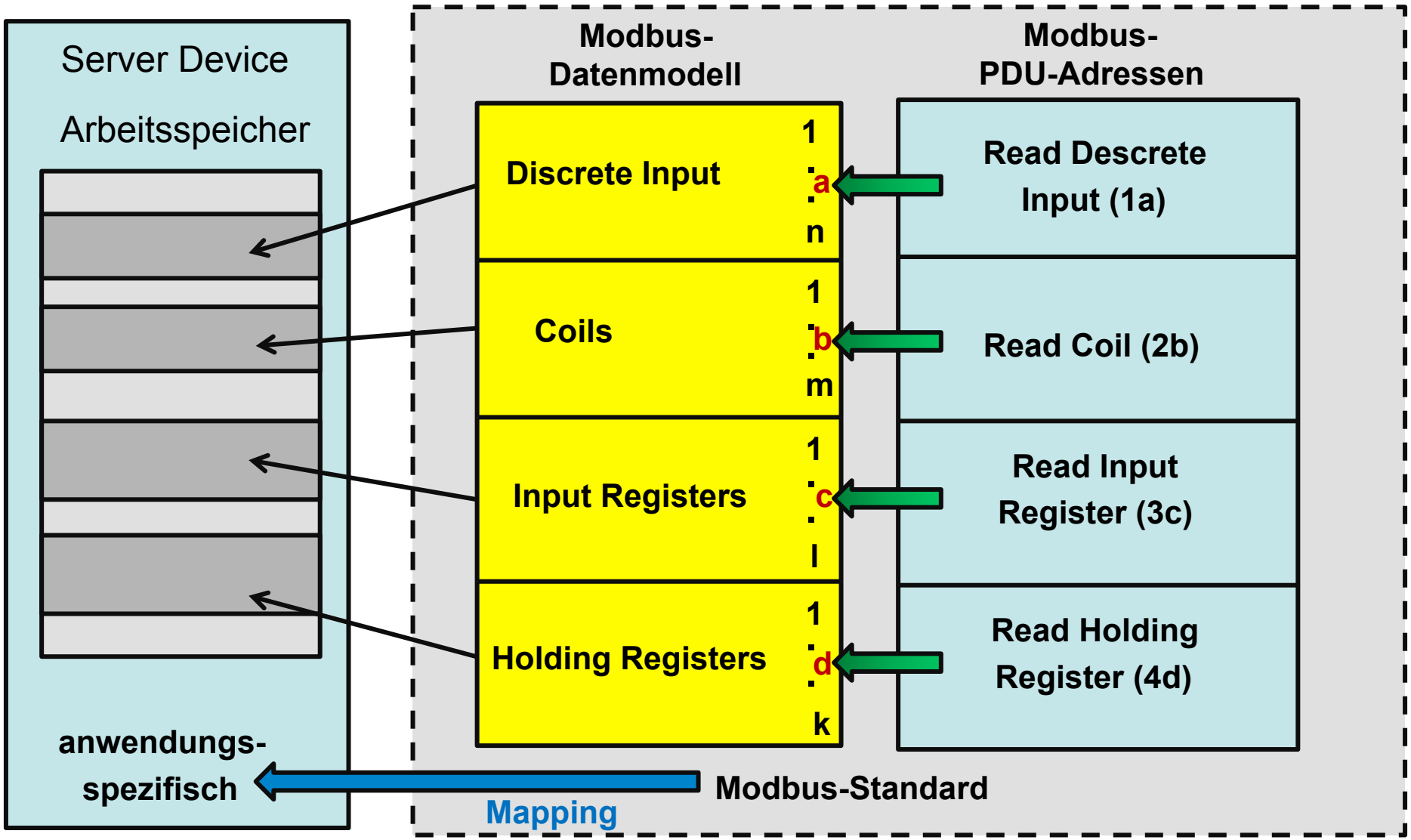
DHBW Stuttgart

Regeln der Adressierung in Modbus-PDUs

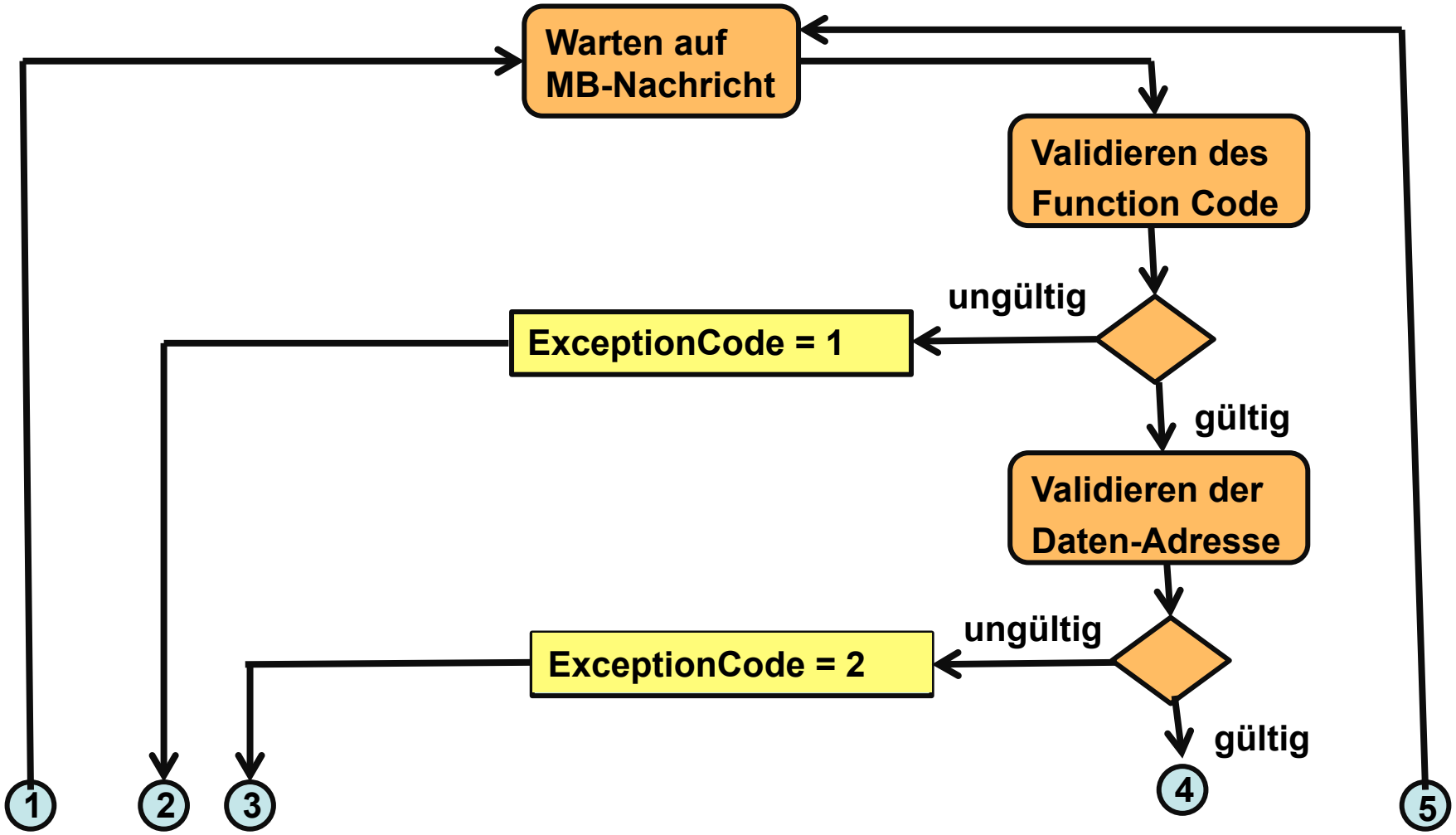
- Adressen für Daten in PDUs im Bereich 0 .. 65 535 (16 Bit)
- Vier getrennte Adressbereiche pro primärer Datentabelle
- Nummerierung der Datenelemente in den Datentabellen von 1 bis n
- Adresse (1a) adressiert Datenelement a im Datenmodell Bereich 1
- Abbildung (Mapping) des Datenmodells auf die Geräte-Implementierung (z.B. IEC-61 131-Objekt)



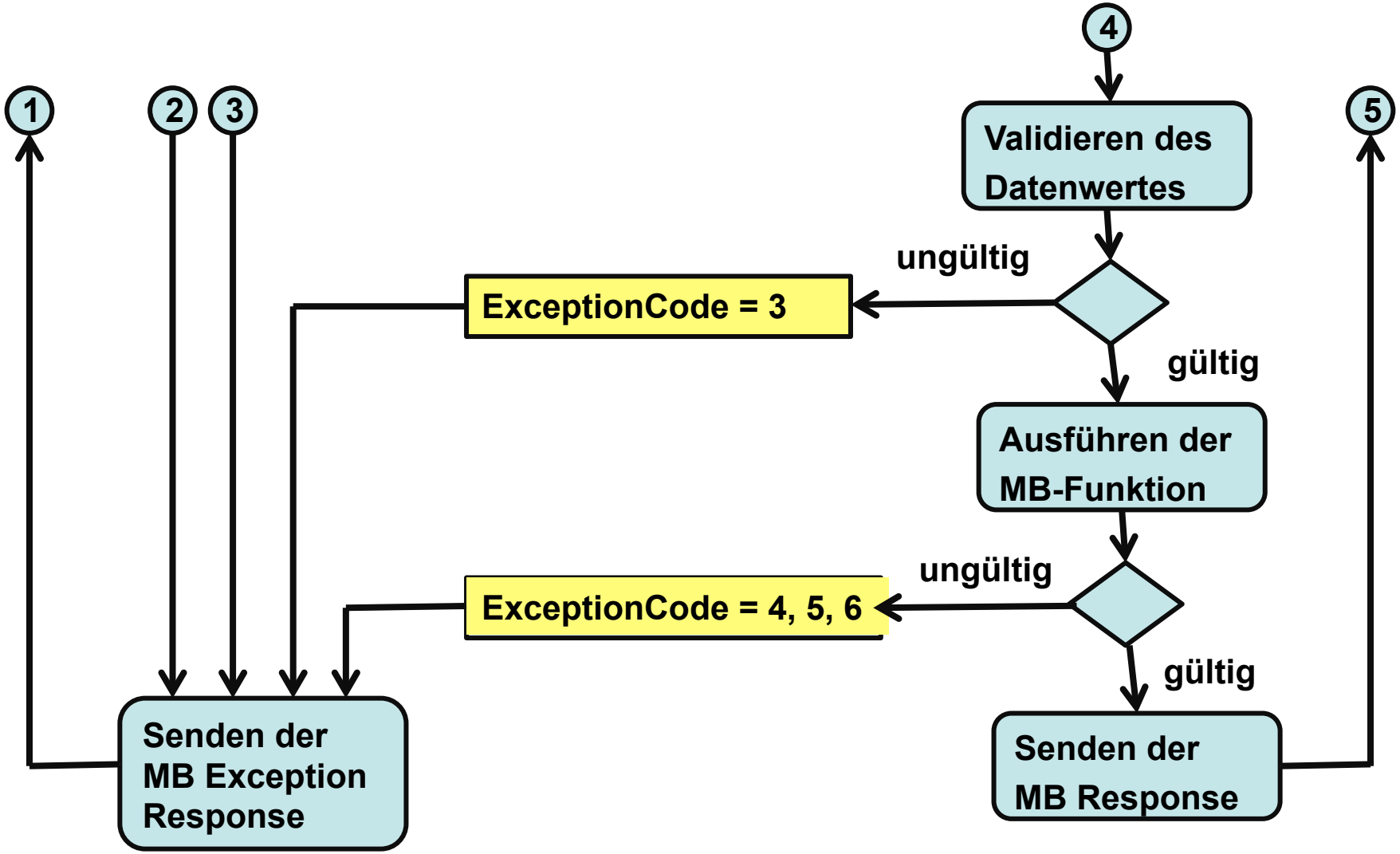
Adressmodell



Zustandsdiagramm und Transaktionen 1



Zustandsdiagramm und Transaktionen 2



Transaction Response

DHBW Stuttgart

Erfolgreiche Transaktion ➔ positive Modbus Response

- Response Function Code = Request Function Code

Fehler bei der Transaktion ➔ Modbus Exception Response

- Exception Function Code = Request Funktion Code + 0x80
- Information an den Client über die Art des Fehlers im zugefügten Exception Code
- Beispiel: “ExceptionCode = 01” ➔ Function Code nicht unterstützt

Inhalt

DHBW Stuttgart

Modbus/TCP

- Grundsätze und Versionen
- Protokollbeschreibung
- Datenmodell und Datencodierung
- Adressierung und Transaktionen
- Function Codes

DHBW Stuttgart

Grundlegende Merkmale der Function Codes

- Codierung in 1 Byte: gültige Werte 1 .. 255
- Bereich 128 .. 255 für Exceptions (Fehlercodes) genutzt
- Festlegung der auszuführenden Aktion durch den Function Code in Nachrichten vom Client zum Server
- Sub-Function Codes zur Definition von Mehrfach-Aktionen

DHBW Stuttgart

Grundlegende Merkmale des Datenfeldes

- Zusatzinformationen zur Aktion in Nachrichten vom Client zum Server
- Angeforderte Antwortdaten in Nachrichten vom Server zum Client bei fehlerfreier Ausführung
- Exception Code (Fehlercode) in Nachrichten vom Server zum Client bei Auftreten von Fehlern
- Beispiele: Adressen von Bitdaten und Registern, Anzahl von Objekten der Aktion, Länge des Datenfeldes
- Option: leeres Datenfeld (0 Byte) bei Nichtvorliegen von Daten

Kategorien von Function Codes

DHBW Stuttgart

Public Function Codes : standardisiert und veröffentlicht

- Definition und Validierung durch Modbus Organization
- Conformance Tests festgelegt

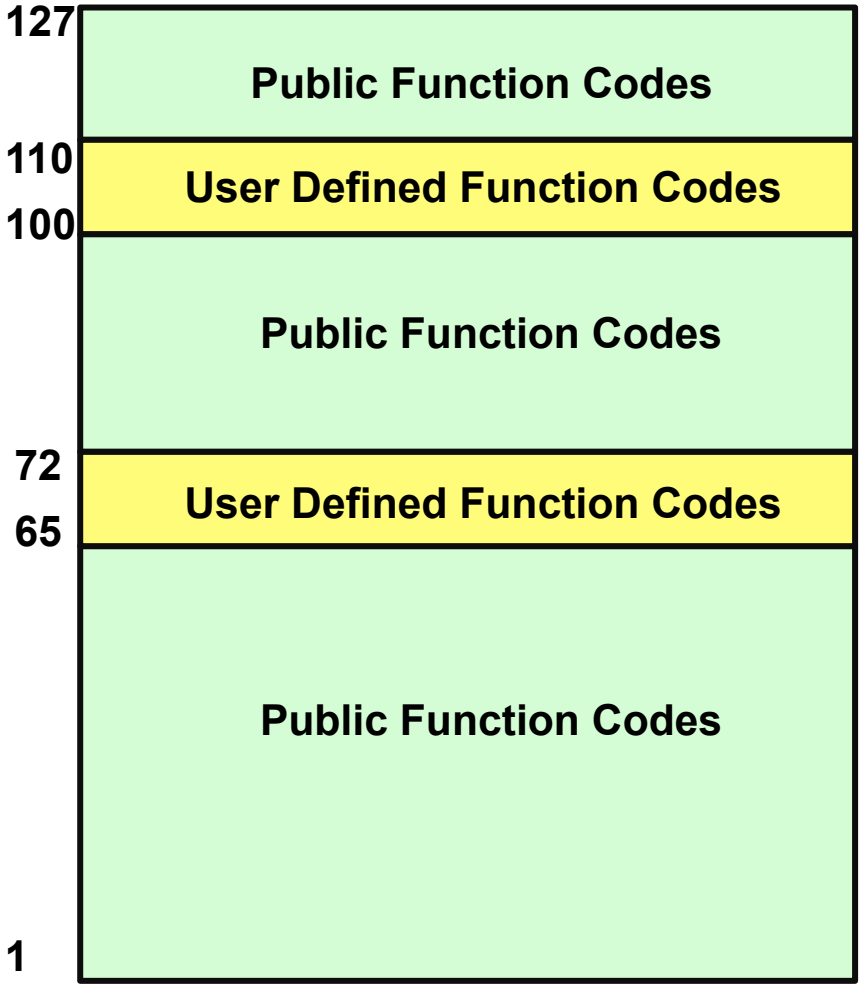
User-Defined Function Codes : nutzerspezifische Bereiche

- Auswahl und Implementierung durch den Nutzer
- Eindeutigkeit nicht gewährleistet

Reserved Function Codes (128..255)

- Nutzung teilweise durch Legacy-Anwendungen
- Nutzung auch für Fehlersignalisierung (Exceptions)

Bereiche der Function Codes



Bereich 128 .. 255: Reserved Function Codes

Function Code 0: nicht verwendet

Function Codes für Datenzugriff

DHBW Stuttgart

Datenelemente	Zugriff	Fu. Code
Physical Descrete Inputs	Read	02
Physical Coils oder Internal Bits	Read	01
	Write Single	05
	Write Multiple	15
Physical Input Registers	Read	04
Physical Holding Registers oder Internal Registers	Read	03
	Write Single	06
	Write Multiple	16
	Read/Write Multiple	23
	Mask Write	22
	Read FIFO	24
File Records	Read	20
	Write	21
Encapsulated Interface Transport		43
CANopen General Reference		43-13

DHBW Stuttgart

Beschreibungsbeispiel: Function Code 01 “Read Coils”

- Lesen von 1 bis 2000 zusammenhängenden Bitwerten
- Ermitteln des Status von Coils in abgesetzten Geräten
- Request PDU: Startadresse für das erste Element (Coil) und Anzahl der zusammenhängenden Elemente
- Response PDU: Übermitteln der gelesenen Bitwerte (1 = ein ; 0 = aus)

Beispiel 1: Function Code 01 (2)

DHBW Stuttgart

Request PDU

Function Code	1 Byte	0x01
Startadresse	2 Byte	0x0000 bis 0xFFFF
Anzahl der Bitwerte	2 Byte	1 bis 2000 (0x7D0)

Response PDU

Function Code	1 Byte	0x01
Anzahl folgender Bytes	1 Byte	z
Statuswerte (Coils)	n Byte	n = z oder z+1

n = z, wenn Anzahl der Bitwerte durch 8 teilbar, sonst z+1

Beispiel 1: Function Code 01 (3)

DHBW Stuttgart

Exception Response PDU

Exception Function Code	1 Byte	0x81
Exception Code	1 Byte	01, 02, 03 oder 04

Exception Code 01: ungültiger Function Code

Exception Code 02: ungültige Anzahl von Bitwerten

Exception Code 03: ungültige Startadresse

Exception Code 04: Fehler beim Lesen der Bitwerte

DHBW Stuttgart

Function Code 23 (0x17) “Read/Write Multiple Registers”

- Lese- und Schreib-Aktionen für Holding-Register in einer Transaktion
- Ausführen des Schreib-Aktionen vor dem Lese-Aktionen
- Request PDU: Startadresse für das erste zu lesende Holding-Register und Anzahl der danach zu lesenden Elemente
- Request PDU: Startadr. für das erste zu schreibende Holding-Register, Anzahl der danach zu schreibenden Elemente und Schreibdaten
- Response PDU: Übermitteln der Werte aus den gelesenen Holding-Registern

Beispiel 2: Function Code 23 (2)

DHBW Stuttgart

Request PDU

Function Code	1 Byte	0x17
Lese-Startadresse	2 Byte	0x0000 bis 0xFFFF
Anzahl der Lesewerte	2 Byte	L: 0x0001 bis 0x007D
Schreib-Startadresse	2 Byte	0x0000 bis 0xFFFF
Anzahl der Schreibwerte	2 Byte	S: 0x0001 bis 0x0079
Anzahl der Schreibbyte	1 Byte	2 x S
Register-Schreibwerte	S x 2 Byte	Daten

Response PDU

Function Code	1 Byte	0x17
Anzahl folgender Bytes	1 Byte	2 x L
Register-Lesewerte	L x 2 Byte	Daten

Beispiel 2: Function Code 23 (3)

DHBW Stuttgart

Exception Response PDU

Exception Function Code	1 Byte	0x97
Exception Code	1 Byte	01, 02, 03 oder 04

- **Exception Code 01**:
  Function Code ungültig
- **Exception Code 03**:
  Anzahl der Schreibwerte oder Anzahl der Lesewerte ungültig oder Anzahl der Schreibbyte ungleich der doppelten Anzahl der Schreibwerte
- **Exception Code 02**:
  Startadressen für das Lesen oder das Schreiben ungültig
- **Exception Code 04**:
  Fehler beim Lesen oder Schreiben der Registerwerte