

2.6 OSI-Referenzmodell: Schicht 6 - Darstellung

Inhalt

- Aufgaben und Funktionen
- Dienstelemente
- ASN.1 (Abstract Syntax Notation One)

Die **Darstellungsschicht** sorgt für die korrekte Interpretation der übertragenen Informationen. Dazu wird die lokale Syntax (Codierung) der Information in Endsystemen in eine einheitliche Transfersyntax übersetzt. **Datenkompression und Verschlüsselung** gehören ebenfalls zu ihren Aufgaben.

Die Aufgaben **Verschlüsselung und Datenkompression** können nach der Systematik des OSI-Modells der Darstellungsschicht zugeordnet werden. Hier wurde jedoch eine Einordnung in die Themen Sicherheit bzw. Multimedia gewählt.

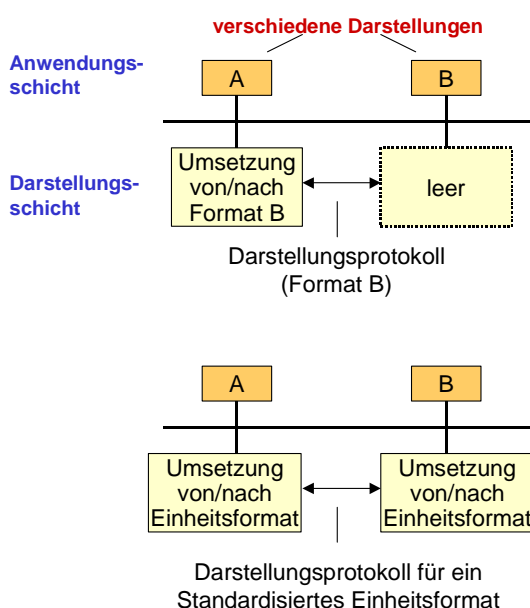


Bild: Formatkonvertierung in der Darstellungsschicht

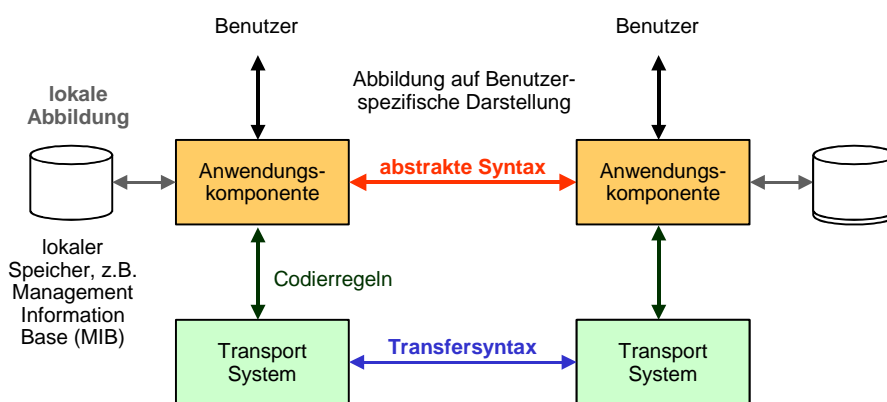


Bild: Abstrakte Syntax und Transfersyntax

Dienstelemente

Die Darstellungsschicht (Präsentationsschicht) enthält echte Dienste (die eine Funktion beinhalten) und Durchreichtedienste, die nur die Verbindung zwischen Anwendungs- und Sitzungsschicht herstellen.

Der Verbindungsaufbau wird durch die **P-Connect-Dienstelemente** veranlasst, der geordnete Verbindungsabbau durch **P-Release** und der abrupte Verbindungsabbau durch **P-Abort**. Die Dienstelemente des Context Management können einen Presentation Context aushandeln. Dazu besitzt jede Instanz eine Liste der ihr bekannten abstrakten Syntaxen und der zugehörigen Transfersyntaxen.

Die Anwendungsschicht teilt der Darstellungsschicht beim Verbindungsaufbau die gewünschte abstrakte Syntax mit. Die Darstellungsschicht ergänzt die verfügbaren Transfersyntaxen und sendet die Liste an die Partnerinstanz.

Diese wählt einen ihr bekannten Presentation Context aus der Liste aus. Er wird für die Übertragung verwendet. Falls keine Einigung auf einen Presentation Context möglich ist, kann die Verbindung nicht hergestellt werden.

Nach einem erfolgreichen Verbindungsaufbau übergibt die Anwendungsschicht ihre Datenpakete an die Darstellungsschicht unter Angabe der abstrakten Syntax und des enthaltenen syntaktischen Typs. Die Darstellungsschicht kann nun den Darstellungskontext auswählen und die Datenpakete in die Transfersyntax übersetzen. Diese Übersetzung wird beim Empfänger rückgängig gemacht.

(1) Syntax-Auswahl

Auswahl einer geeigneten Syntax für eine Übertragung und gegebenenfalls Änderung dieser Wahl

(2) Syntax-Transformation

Konvertierungen von Code oder Zeichensatz (z.B. ASCII -> EBCDIC), Modifikation im Datenlayout (z.B. auf Bildschirm, mobiles Gerät)

Eine **abstrakte Syntax** wird benötigt, um Datenstrukturen und Inhalte logisch zu beschreiben. Die abstrakte, logische Beschreibung der Datenstrukturen ist nach OSI in allen beteiligten Systemen dieselbe.

Bild: Dienste der Darstellungsschicht

Die systeminterne Darstellung wird als konkrete, lokale Syntax bezeichnet. Sie kann in Abhängigkeit vom Prozessor und vom Betriebssystem unterschiedlich sein. Für die Übertragung wird die lokale Syntax mit Hilfe der abstrakten Syntax in die (konkrete) Transfersyntax übersetzt.

Aufgaben der Darstellungsschicht

Hauptaufgabe der **Darstellungsschicht** (auch **Präsentationsschicht**, presentation layer) ist es, Dateneinheiten der Anwendungsschicht unter Wahrung ihres Informationsgehaltes zu übertragen.

(1) Anforderung des Aufbaus einer Sitzungsverbindung

(2) Datenübertragung

(3) Aushandeln der zu wählenden Transfersyntax

Transformationen der lokalen Syntax auf ausgehandelte Transfersyntax

(4) Syntax-Transformationen

- drei unterschiedliche Syntaxen für Daten möglich: Sender- und Empfängerseite, Transfersyntax
- Transformationen zwischen Syntaxen innerhalb der Darstellungsschicht:
 - > senderseitige Syntax
 - > Transfersyntax bzw. Transfersyntax
 - > empfängerseitige Syntax
- geeignete Formatierung (zur Darstellung auf dem Ausgabegerät), Codierung und Kompression von Daten

(5) Anforderung des Abbaus einer Sitzungsverbindung

Da in den Endsystemen unterschiedliche Codierungen (als **lokale Syntax** bezeichnet) vorliegen können, wird eine einheitliche, genormte Codierung (als **Transfersyntax** bezeichnet) für die Übertragung verwendet. Paare aus einer abstrakten Syntax und einer Transfersyntax werden als Darstellungskontext bezeichnet.

Der Darstellungskontext bestimmt die **Übersetzungsregeln** für die Codierung und Decodierung.

Die Darstellungsschicht hat insgesamt die folgenden Aufgaben zu erfüllen:

- Verbindungsauf- und -abbau,
- Kontext-Management,
- Datentransfer mit Datentransformation.

Bild: Funktionen der Darstellungsschicht

Anwendung	Elemente des Darstellungsprotokolls
einfacher Text	Zeichenalphabet, Steuerzeichen für Beginn, Ende, Zeilenende (LF), Seitenende (FF) usw.
Textverarbeitung	Zeichenalphabet, Steuerzeichen für Beginn, Ende, Zeilenende (LF), Seitenende (FF) usw., Schriftarten, Schriftgrößen, Absätze, Einzüge, Kopf- und Fußzeilen usw.
Zeichenprogramme (objektorientiert)	Codierung von Positionen, Größe und Objektarten wie Striche, Rechtecke, Kreise usw.
Schwarz/Weiß oder Graustufen-Festbilder (z.B. Telefax)	Codierung von Bildbeginn und Ende, Graustufen-Codierung der Bildelemente, Redundanzreduzierende Codierung
Sprache	Quantisierung und Codierung der Abtastwerte, Redundanzreduzierende Codierung

Einige Basiselemente in Darstellungsprotokollen.

Bild: Anwendung und Darstellung von Objekten

Anwendung	Darstellungsformat
Text	ASCII, EBCDIC
Bildübertragung	GIF, JPEG
Video	MPEG
Audio	PCM, H.261
WWW	HTML
E-Mail	MIME
SNMP	ASN.1/ BER
RPC	eXternal Data Representation (XDR)

Einige Beispiele für Darstellungsformate.

SNMP	Simple Network Management Protocol	GIF	Graphical Interchange Format
RPC	Remote Procedure Call	JPEG	Joint Photographic Expert Group
PCM	Pulse Coded Modulation	MPEG	Motion Picture Expert Group
ASN	Abstract Syntax Notation Number One	HTML	Hypertext Markup Language
BER	Basic Encoding Rules	MIME	Multipurpose Internet Mail Extension

ASCII	American National Standard Code for Information Interchange
EBCDIC	Extended Binary-Coded-Decimal Interchange Code

Bild: Anwendungen und ihre Darstellungsformate

Dienstelemente der Darstellungsschicht

Dienst	Dienstelemente	Parameter
Verbindungsaufbau	P-Connect.Request P-Connect.Indication P-Connect.Response P-Connect.Confirm	Presentation-Context- Definition-List, Quality-of-Service, Initial-Assignment-of-Tokens, Result-List, Quality-of-Service, Initial-Assignment-of-Tokens
Verbindungsabbau	P-Release.Request P-Release.Indication P-Release.Response P-Release.Confirm P-U-Abort.Request P-U-Abort.Indication P-P-Abort.Indication	User-Data, Reason
Kontext-Management	P-Define-Context.Request P-Define-Context.Indication P-Define-Context.Response P-Define-Context.Confirm P-Delete-Context.Request P-Delete-Context.Indication P-Delete-Context.Response P-Delete-Context.Confirm	Presentation-Context-1d Definition-List Result-List Presentation-Context-1d- List Result-List
Datentransfer	P-Data.Request P-Data.Indication P-Expedited-Data.Request P-Expedited-Data.Indication	User-Data: Presentation- Context-1d, Tag, Value

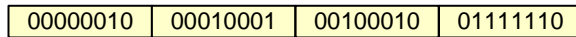
Die Durchreachedienste der Darstellungsschicht stellen dem Anwender die Funktionen der Sitzungsschicht für die Dialogsteuerung zur Verfügung.

Durchreachedienste der Darstellungsschicht (Req = Request, Ind = Indication, Resp = Response, Conf = Confirm)

P-Token-Give Req, .Ind	P-Activity-Start Req, .Ind
P-Token-Please Req, .Ind	P-Activity-Interrupt Req, .Ind, .Resp, .Conf
P-Sync-Minor Req, .Ind, .Resp, .Conf	P-Activity-Resume Req, .Ind
P-Sync-Major Req, .Ind, .Resp, .Conf	P-Activity-End Req, .Ind, .Resp, .Conf
P-Resynchronize Req, .Ind, .Resp, .Conf	P-Activity-Discard Req, .Ind, .Resp, .Conf

- Verschiedene Maschinen verwenden unterschiedliche Reihungen von Bytes
- Beispiel: Zahl 34677374:

- **Big Endian (z.B. Motorola)**



- **Little Endian (z.B. Intel)**

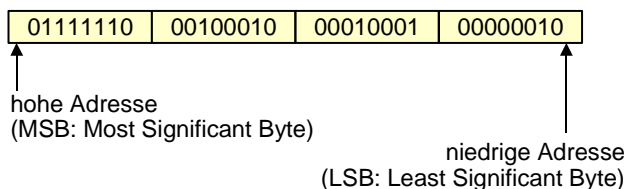


Bild: Darstellung von Integer-Zahlen

Abstrakte Syntax

- Menge von Typdefinitionen für Datenobjekte, z.B.
 - ASN.1
 - XDR language (eXternal Data Representation)
- beschreibt generische Datenstruktur unabhängig von der Art der Codierung

Transfersyntax

- konkrete Repräsentation der beschriebenen Daten bei der Übertragung
- definiert durch eine Menge von Codierregeln, z.B.
 - Basic Encoding Rules (BER) für ASN.1
 - XDR Regeln

Bild: Abstrakte Syntax und Transfersyntax

- Beschreibungssprache für Datentypen
 - Untermenge wird für SNMP Objekte verwendet (abstrakte Syntax)
- Basic Encoding Rules (BER)
 - beschreiben die Abbildung der Datentypen auf die übertragenen Bits (Transfersyntax)

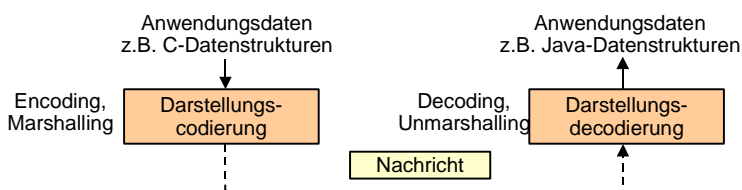


Bild: Abstrakte Syntax Notation One ASN.1

Bei der Übermittlung von Nachrichten zwischen zwei Endsystemen mit Betriebssystemen, die auf verschiedenen CPUs laufen, kann der Reihenfolge der Byte-Abspeicherung verschieden sein. In diesem Fall muss die Byte-Reihung der ankommenden Nachrichten abgeändert werden.

ASN.1 und BER

ASN.1 (Abstract Syntax Notation One) ist eine Sprache zur Spezifikation abstrakter Syntaxen. Mittels ASN.1 können Datentypen und ihre Werte beschrieben werden. BER (Basic Encoding Rules) beschreibt eine Transfersyntax, erlaubt also die eindeutige Beschreibung der Daten (Typ und Wert) auf der Bitebene.

ASN.1 und BER sind von ITU-T und ISO standardisiert.

ASN.1 wird für mehrere Zwecke eingesetzt:

- zur Definition abstrakter Syntaxen für Anwendungsdaten,
- zur Beschreibung von Anwendungs- und Darstellungs-PDUs,
- zur Beschreibung der MIB (Management Information Base) beim Systemmanagement sowohl nach OSI als auch nach SNMP.

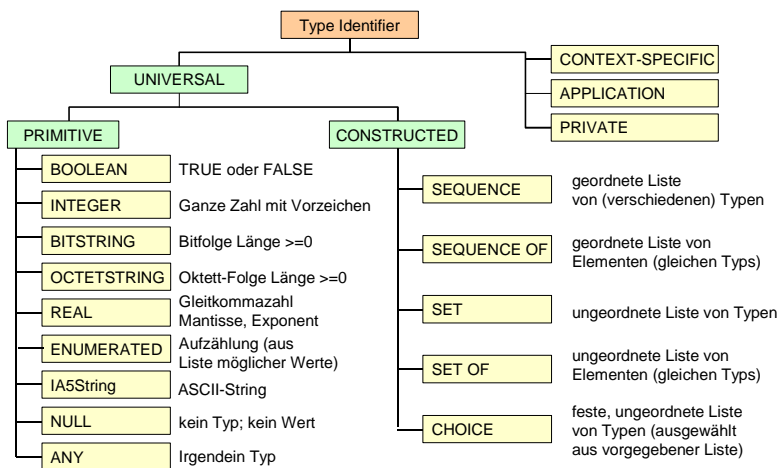


Bild: Datentypen in ASN.1

Ein **Datentyp** (kurz: Typ) kann als Menge (collection) von Werten aufgefasst werden. Ein Typ kann eine unendlich große Anzahl von Wert annehmen. Dies ist beim Typ INTEGER der Fall.

Typen können **einfach** (primitive) sein (diese sind atomar, bestehen also nicht aus Komponenten).

Zusammengesetzte Typen (structured oder constructed) bestehen aus Komponenten, denen jeweils wieder ein Typ zugeordnet ist. ASN.1 unterscheidet vier Klassen von Datentypen bzw. Tags.

Tags in ASN.1

Klasse des Datentyps	Aufbau des Typs	Definition	Gültigkeit
UNIVERSAL	elementar, zusammengesetzt	vordefiniert	Internationaler Standard
APPLICATION	zusammengesetzt	selbst definiert	Eine Anwendung, mehrere Organisationen
PRIVATE	zusammengesetzt	selbst definiert	Eine Organisation
CONTEXT- SPECIFIC	zusammengesetzt	selbstdefiniert	Ein strukturierter Typ innerhalb einer Anwendung

Jeder Datentyp (mit Ausnahme von ANY und CHOICE) wird durch ein Tag gekennzeichnet, das aus dem Klassennamen und einer Nummer besteht, z. B. UNIVERSAL 5 oder PRIVATE 22.

BER-Codierung eines Datenwertes umfasst

- **Typ** (Ein oder mehrere Oktett, welche codierung für Datentyp-Klasse und -Nummer enthalten)
- **Länge**
- **Wert** (Value, Codierung des eigentlichen Inhalts)

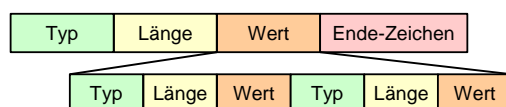
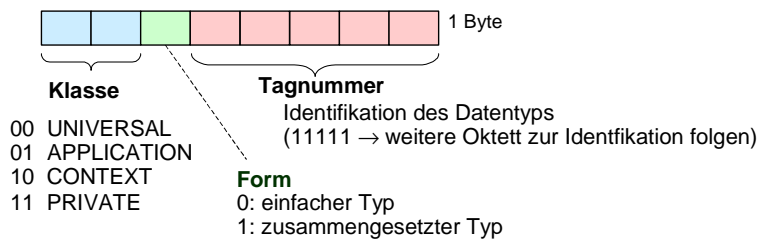


Bild: Basic Encoding Rules (BER)

Die **Transfersyntax** wird erhalten, indem die Basic Encoding Rules (BER) auf die abstrakte Syntax angewendet werden.

Dazu wird jeder (Wert eines Typs durch ein Datenelement dargestellt, das aus **drei Felder** besteht, die je ein oder mehreren Oktette enthalten können:

- Das **Identifizier-Feld** kennzeichnet den vorliegenden Typ gemäß ASN.1
- Das **Länge-Feld** enthält die Anzahl der Oktette im Feld Inhalt.
- Das **Feld Inhalt** enthält den eigentlichen Wert, bei strukturierten Typen können dies weitere Datenelemente sein.



Datenelemente dieser Form werden als TLV (Type Length Value) bezeichnet.

Das Typfeld muss den Typ eindeutig spezifizieren. Das Identifier-Feld besteht aus einem Byte oder - falls die Tagnummer größer als 30 ist - aus mehreren Bytes.

Bekanntes Längenfeld (Länge des Inhalts in Oktett)

- kurze Form: 1 Byte
 - erstes Bit 0, weitere 7 Bits codieren Länge als binäre Ganzzahl
- lange Form: n Bytes, n > 1
 - erstes Byte: erstes Bit 1, weitere Bits codieren Anzahl der folgenden Längenoktett
 - folgende Längenoktett codieren Länge als binäre Ganzzahl

Das Längenfeld besteht aus einem Byte (falls Länge < 128) oder aus mehreren Bytes. Für bestimmte Typen muss das Längenfeld nicht im Voraus bekannt sein. Dann wird das Längenfeld als ein Oktett mit dem Wert 8016 und das Ende des Datenelements durch zwei Oktette mit dem Wert 000016 dargestellt.

Unbekanntes Längenfeld

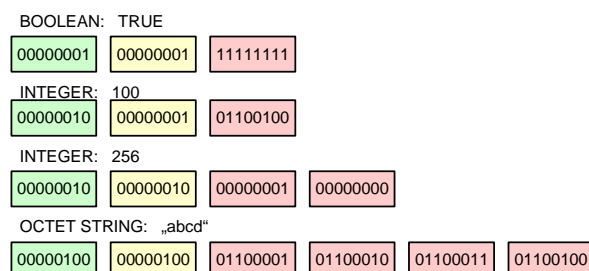
- 00000000₁₆ als spezielles Ende-Zeichen
- Kennung: 1 Byte
 - erstes Bit 1, alle anderen Bits 0

Bild: Typ- und Längenfeld

Tag: 0 ... 30 für Klasse UNIVERSAL

- 1 = Boole'scher Typ
- 2 = Ganzzahltyp
- 3 = Bitstring-Typ
- 4 = Oktettstring-Typ
- 5 = Null-Typ
- 9 = Real-Typ
- 10 = Aufzählungstyp
- 16 = Sequence/Sequence-of-Typ
- 17 = Set/Set-of-Typ
- 18-22 = Stringtypen mit alternativen Zeichensätzen
- 23-24 = Zeittypen
- 25-27 = Stringtypen mit alternativen Zeichensätzen
- > 30 = alle 5 Bit auf Eins, weiter im nächsten Oktett

Einfache Typen	Zusammengesetzte Typen	Andere Typen
- BOOLEAN - INTEGER - ENUMERATED - REAL - BIT STRING - OCTET STRING - NULL	- SET - SET OF - SEQUENCE - SEQUENCE OF - CHOICE - SELECTION - TAGGED - ANY	- NumericString - PrintableString - TeletexString - VideotexString - VisibleString - IA5String - CharacterString - GraphicString - GeneralString - UTCTime - ObjectDescriptor - ObjectIdentifier



Einige Beispiele einer BER-Codierung sind im Bild gegeben.

Bild: Beispiele für BER-Codierung