



---

# Rechnernetze

## Eine (kurze) Einführung

Cluj, Wintersemester 2019/20

**Prof. Dr.-Ing. habil. Hartmut König**

---

## II.

# Prinzipielle Funktionsweise eines Rechnernetzes



---

# II.1

## Dienste



König 1.1  
Tanenbaum/Wetherall 1.3.3



---

# II.1.1

## Begriff des Dienstes



---

# Dienste

Rechnernetze stellen **Dienste** (*services*) bereit, mit deren Hilfe der Nutzer auf das Netz zugreifen kann.

- ↪ direkter Zugriff
- ↪ Integration in Anwendungen

● **Dienst:** Funktion oder Fähigkeit (*capability*) eines Rechnernetzes zur Unterstützung der Kommunikation und der Kooperation zwischen den Nutzern.

● Beispiele: - eMail, WWW, telnet, ftp, Skype, Facebook u.v.a.

☞ **Dienstbegriff ist grundlegend für Verständnis und Beschreibung von Kommunikationsabläufen in Rechnernetzen !!!**



---

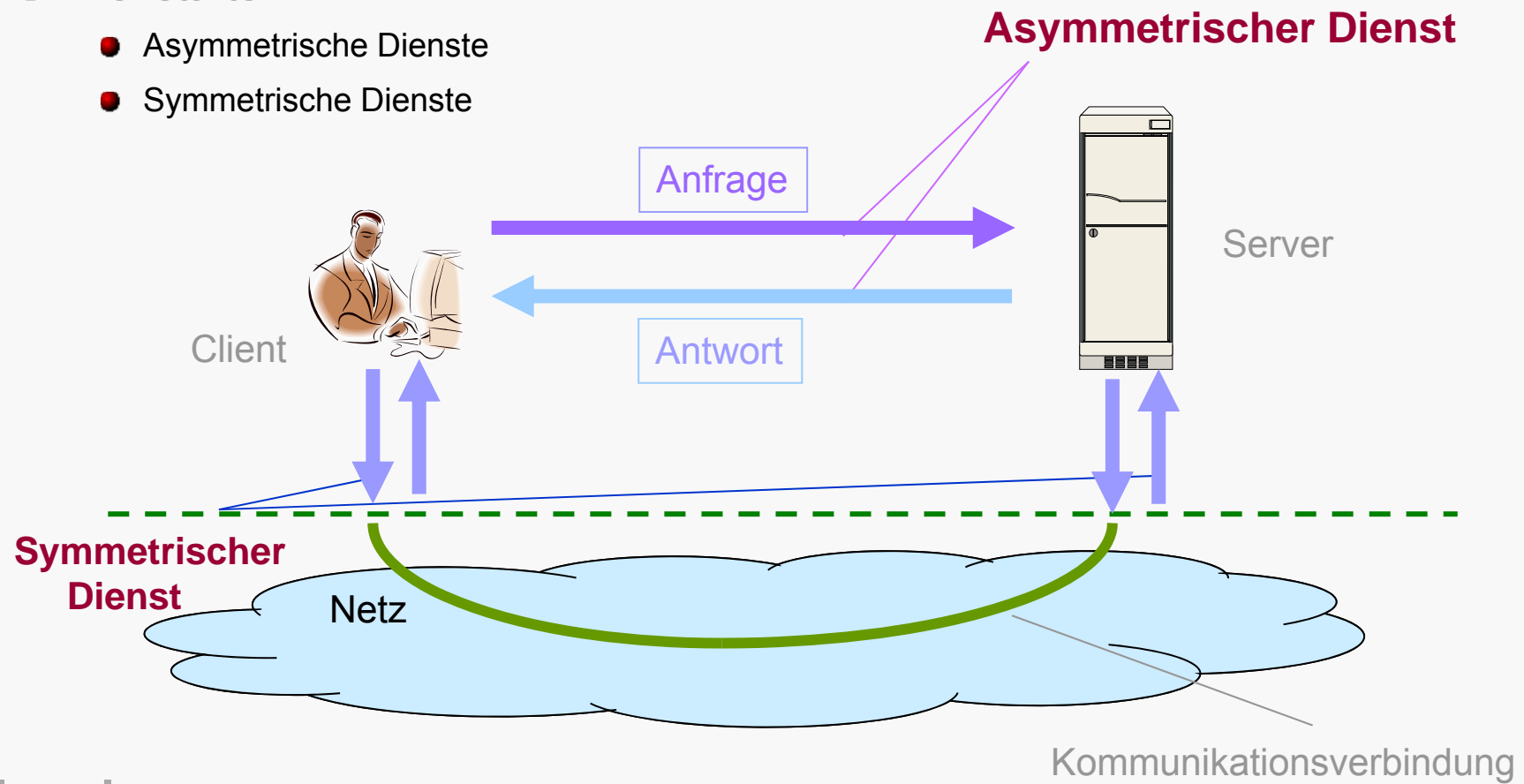
# A) Dienstarten



# Dienst-Arten

## ● Dienstarten

- Asymmetrische Dienste
- Symmetrische Dienste



---

# Asymmetrische Dienste

Dienste, die auf Anfrage eines Dienstnutzers, dem **Client**, von einem Diensterbringer im Netz, dem **Server**, erbracht.

↪ Netz- oder Anwendungsdienste

## ● Realisierung

- entfernter Prozeduraufruf (*remote procedure call*, RPC)

- Verteilungsplattformen

  - ↪ z.B. CORBA

  - ↪ Web Services

☞ Die Mehrzahl der Anwendungsdienste sind asymmetrische Dienste !!!

☞ **Asymmetrische Dienste sind an die Ausführung symmetrischer Dienste gebunden !**





---

# Symmetrische Dienste

Dienste, die die angebotene Dienstleistung an zwei oder mehreren Zugangspunkten gleichzeitig bereitstellen.

- Stellen eine Kommunikationsdienstleistung bereit

☞ **Kommunikationsdienste**

- Auch Netz- und Anwendungsdienste können symmetrischer Natur sein.

- meistens kommunikative Dienste

☞ Instant Messaging, Videokonferenzen

☞ **Im Weiteren betrachten wir vorrangig symmetrische Dienste !!!**



---

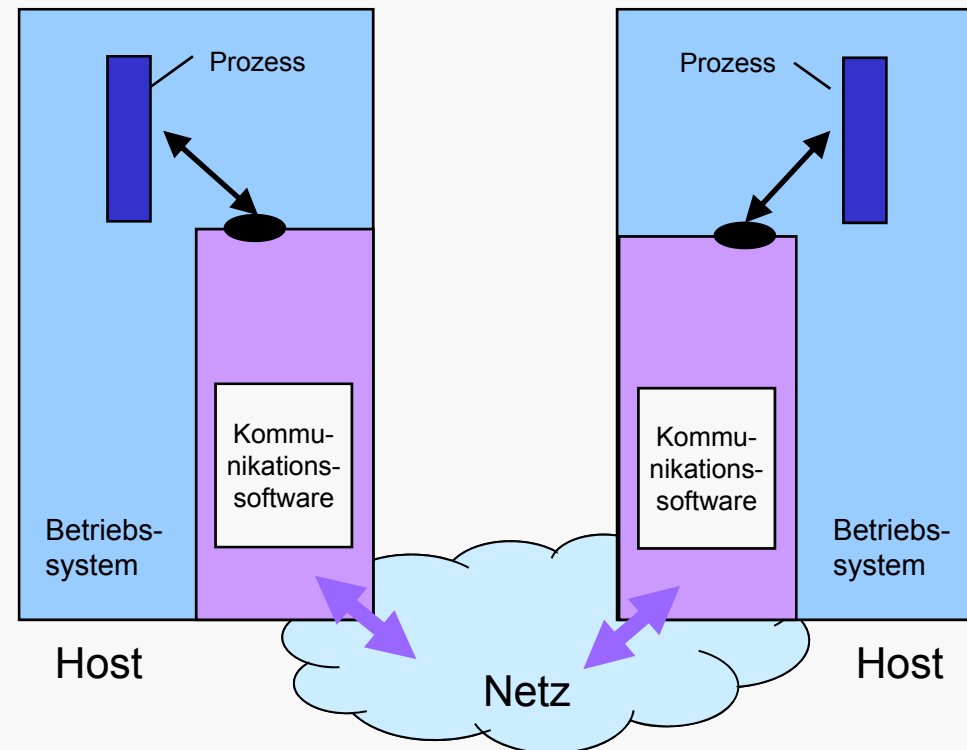
## B) Kommunikationssoftware



# Kommunikationssoftware

Kommunikationsdienste erfordern das Vorhandensein einer speziellen Softwareunterstützung in den Endsystemen (→ **Host**).

- Die Software, die diese in einem Rechner umsetzt, bezeichnet man als **Kommunikationssoftware**.
- Anwendungen greifen unter Verwendung der Adressen der Kommunikationspartner über Dienstschnittstellen auf die Kommunikationssoftware zu.



---

## II.1.2

# Modell eines Kommunikationsdienstes



---

# Modell eines Kommunikationsdienstes (vorläufig)

Dienstnutzer

Dienstnutzer

Dienstschnittstelle

Diensterbringer



---

# Modell eines Kommunikationsdienstes

Das Dienstmodell unterscheidet:

- **Dienstanutzer** (*service user*): Anwender des Dienstes
- **Diensterbringer** (*service provider*): Bereitsteller des Dienstes
- **Dienstschnittstelle**: Schnittstelle zwischen beiden Bereichen

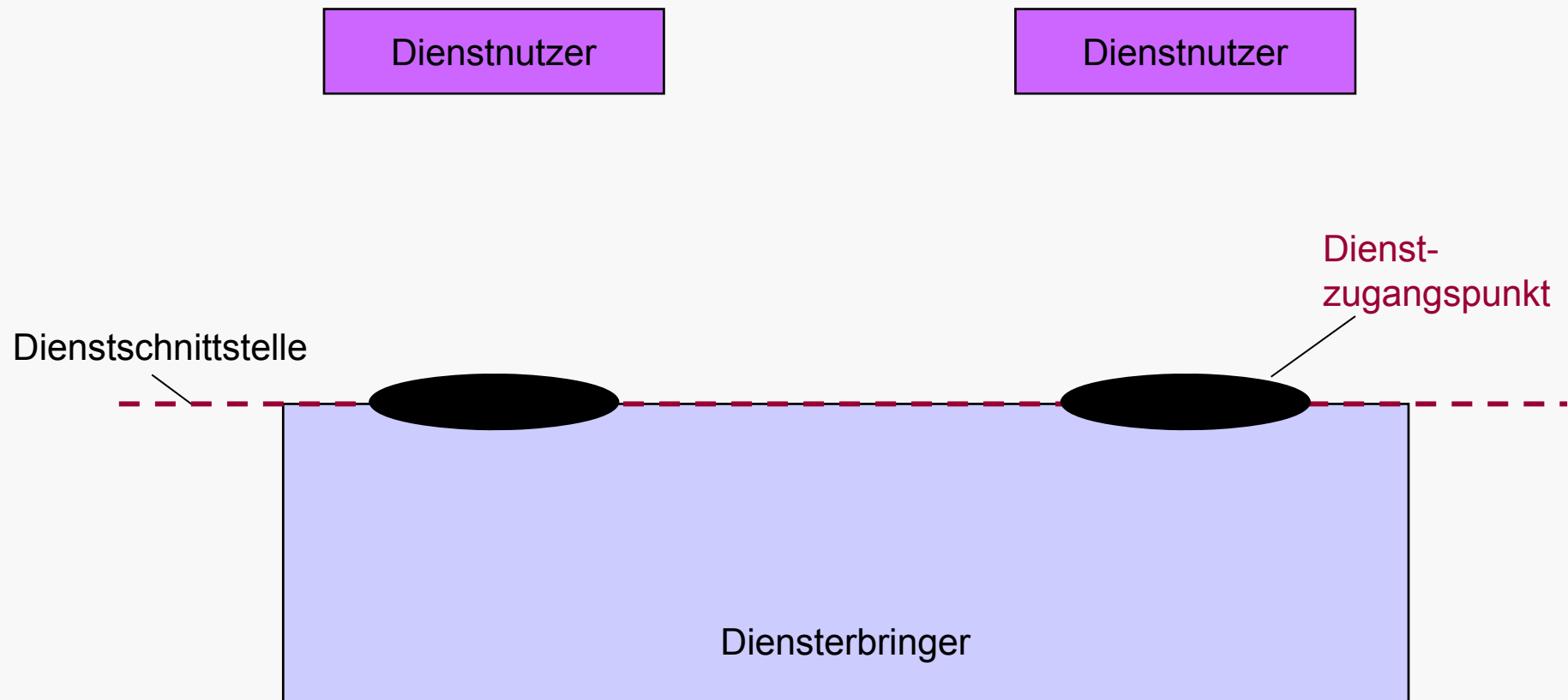
☞ **Es sind immer zwei Dienstanutzer erforderlich, während sich der Diensterbringer als Einheit präsentiert !**

- **Analogie: Telefondienst**
  - ☞ 2 Dienstanutzer (Gesprächspartner)
  - ☞ 1 Diensterbringer (Telefonnetz)



---

# Modell eines Kommunikationsdienstes (vorläufig)



---

# Dienstzugangspunkte

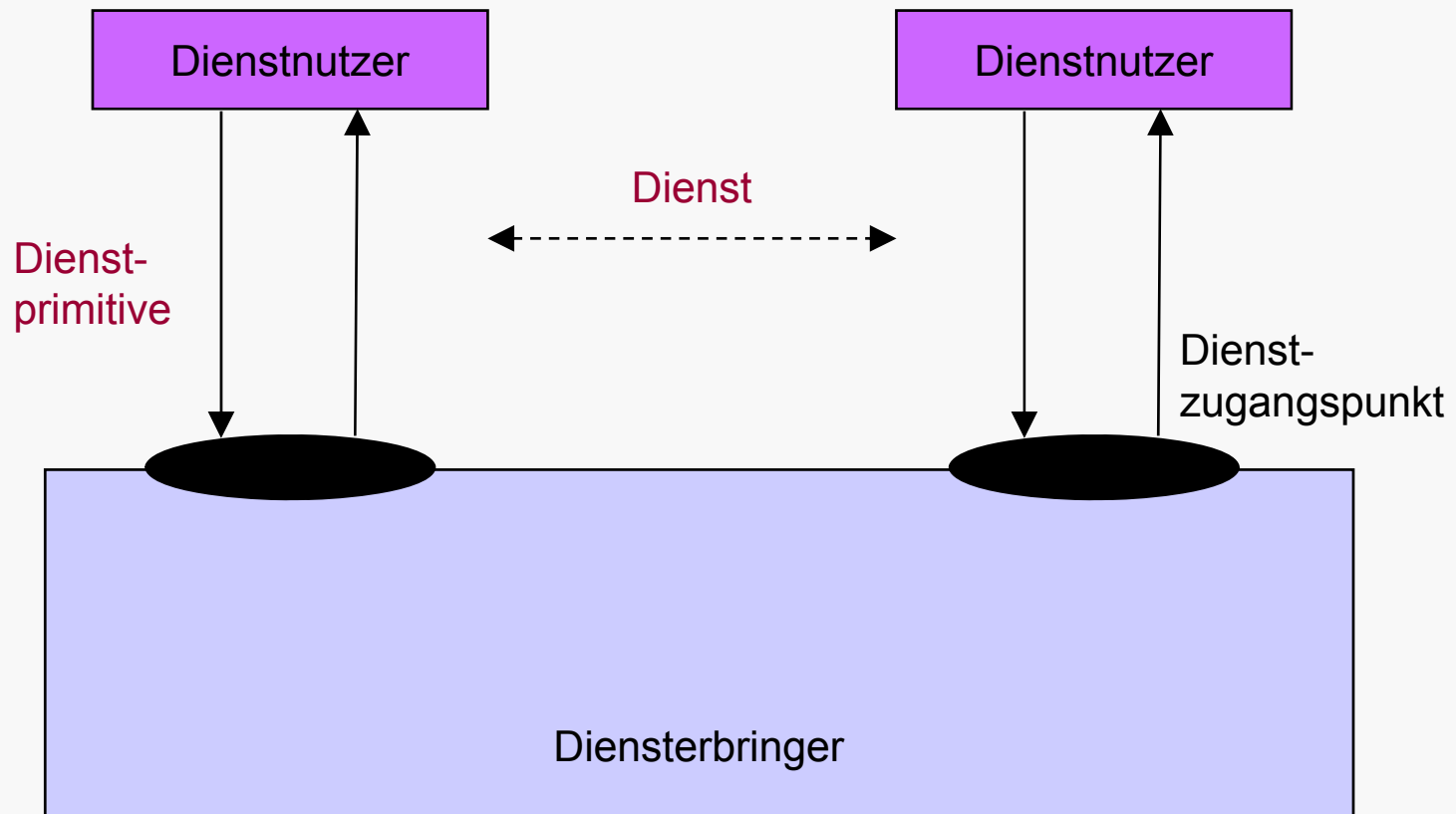
Dienste können vom Dienstinutzer nur an festgelegten **Dienstzugangspunkten** (*service access points, SAP*) in Anspruch genommen werden.

- **Analogie:** - Telefondienst: SAP  $\Rightarrow$  Telefon  
- Briefzustelldienst: SAP  $\Rightarrow$  Briefkasten
- Jeder SAP besitzt eine eindeutige Adresse.
- **Analogie:** Telefonnummer, Postleitzahl





# Modell eines Kommunikationsdienstes (vorläufig)



---

# Dienstprimitive

Die Nutzung eines Dienstes wird durch **Dienstprimitive** (*service primitives, SP*) beschrieben.

● **Dienstprimitiv:** Abstrakte Darstellung der Wechselwirkung zwischen Dienstanutzer und Dienstbringer.

- haben einen festen syntaktischen Aufbau
- Nutzung der Dienstprimitive ist nicht zufällig

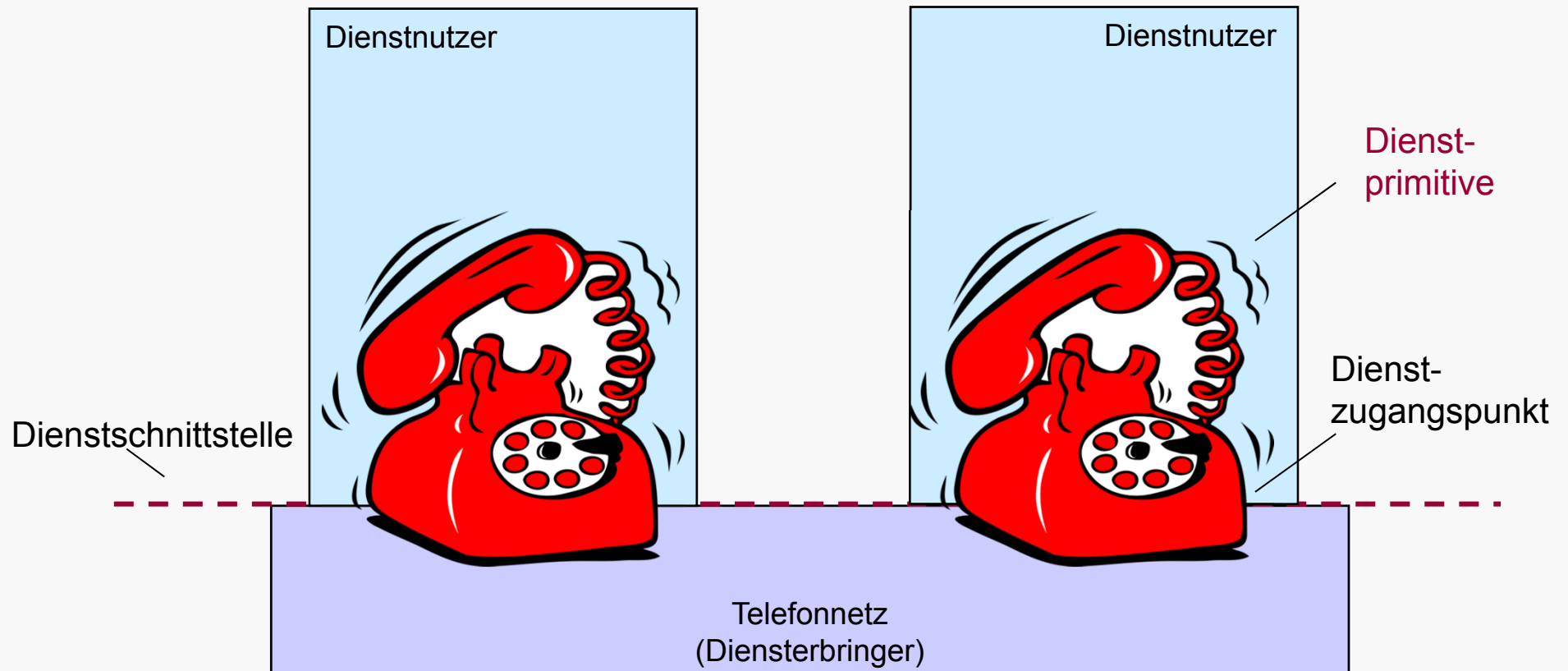
● **Analogie:** - Klingeln des Telefons  
- Frei- bzw. Besetztzeichen

☞ **Dienstprimitive sagen nichts über die konkrete Implementierung aus !**

☞ z.B. Prozeduraufruf,  
Prozessinteraktion



# Beispiel Telefondienst



---

# II.1.3

## Beschreibung von Diensten



---

# A) Notation



---

# Notation für Dienstprimitive (1)

Dienstprimitive werden durch 3 Elemente beschrieben:

***name typ(parameter)***

● **name:** gibt Auskunft über die Funktion des Dienstprimitivs, z. B.

CONNECT	- Verbindungsaufbau
DISCONNECT	- Verbindungsabbau
DATA	- Datentransfer
ABORT	- Verbindungsabbruch

● **typ:** kennzeichnet die Funktion des Dienstprimitivs, i.d.R.

<i>request</i>	- Anfordern eines Dienstes am SAP
<i>indication</i>	- Anzeige der Dienstanforderung am Partner-SAP
<i>response</i>	- Antwort auf ein <i>indication</i> -Primitiv am Partner-SAP
<i>confirm</i>	- Bestätigung der Dienstleistung am anfordernden SAP

● **parameter:** - Adressen, Daten, Qualitätsparameter



---

# Notation für Dienstprimitive (2)

- Beispiele:

- CONNECT request(gerufener SAP, rufender SAP, Dienstgüte, Daten)
- DATA indication(Daten)
- DISCONNECT

heutzutage meistens Abkürzungen, z. B.

- XDATrequ(...)
- XABORTind usw.



---

## B) Abhängigkeiten zwischen Dienstprimitiven





---

# Abhängigkeiten zwischen Dienstprimitiven

Die Verwendung der Dienstprimitive ist nicht zufällig, sondern an einen bestimmten zeitlichen Ablauf gebunden.

- **Analogie:** Folge von „Dienstprimitiven“ beim Telefongespräch

*Hörer abnehmen - Freizeichen erwarten - Nummer wählen –  
Ruf- oder Besetztzeichen erwarten*

- Für die Darstellung der zeitlichen Reihenfolge des Auftretens der Dienstprimitive werden **Zeitablaufdiagramme** (*time sequence diagrams*) verwendet.

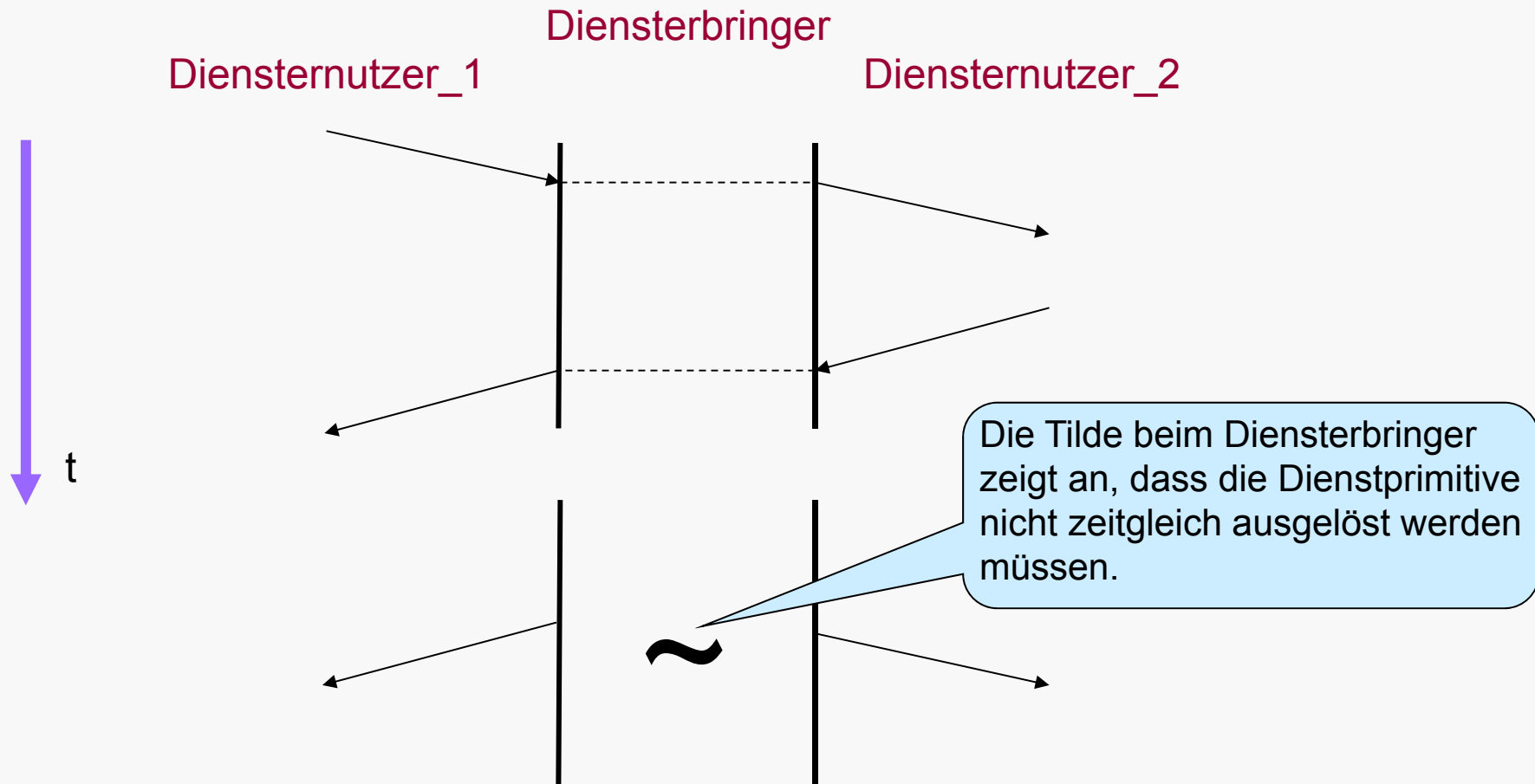


---

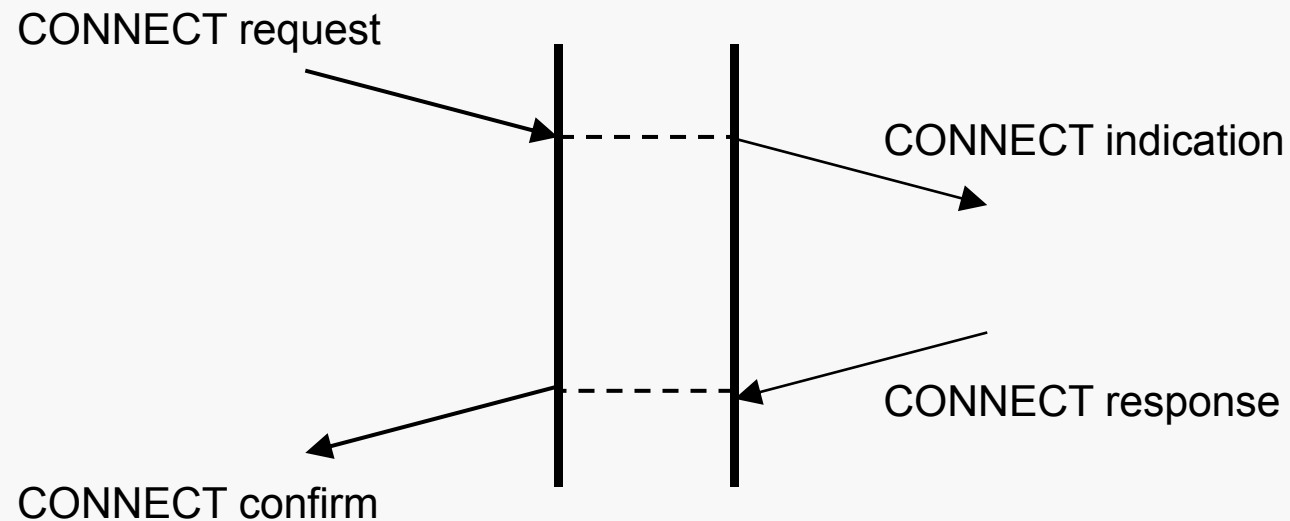
## C) Zeitablaufdiagramme



# Struktur von Zeitablaufdiagrammen

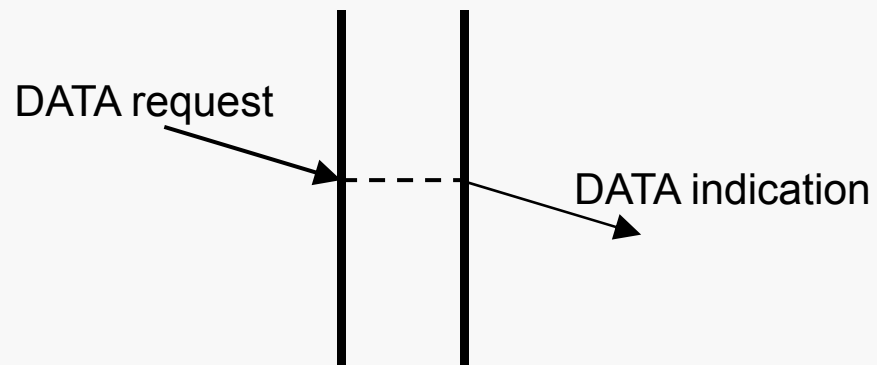


# Beispiele für Zeitablaufdiagramme (1)

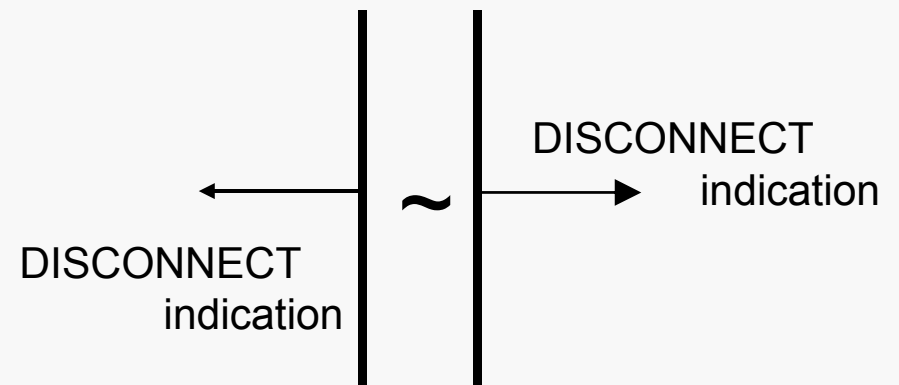


a) Bestätigter (Verbindungsaufbau-) Dienst

## Beispiele für Zeitablaufdiagramme (2)



b) Unbestätigter  
(Datenübertragungs-) Dienst



c) Vom Dienstbringer  
initiiertes Verbindungsabbau

---

# Grenzen der Zeitablaufdiagramme

 **Zeitablaufdiagramme sind nicht ausreichend, um alle Abhängigkeiten zwischen den Dienstprimitiven darzustellen !**

- Nur Beschreibung ausgewählter Abläufe !
- Um Reihenfolgen zwischen den Teilabläufen darzustellen, sind ergänzende Darstellungen erforderlich, z. B.
  - ↪ Matrizendarstellungen
  - ↪ Zustandsdiagramme
    - Beschreiben Zustand des Dienstbringers an dem jeweiligen SAP



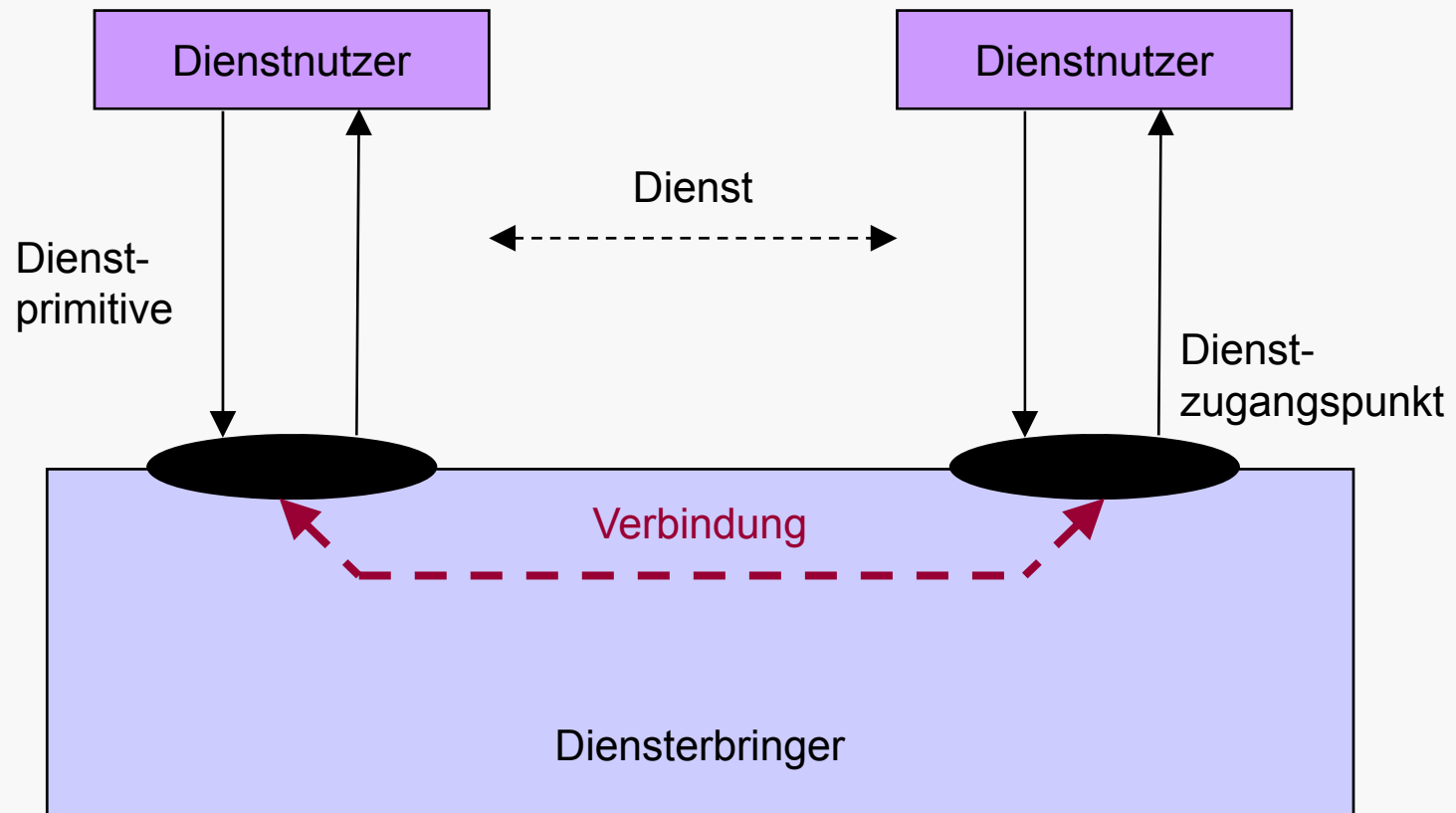
---

## II.1.4

# Verbindungsorientierte und verbindungslose Dienste



# Modell eines Kommunikationsdienstes





---

# Verbindung

Die Dienstanutzer können zwischen den Dienstzugangspunkten eine **Verbindung** (*connection*) aufbauen.

- **Verbindung:** Logische Beziehung zwischen den Kommunikationspartnern für die Dauer einer Kommunikation
- **Analogie:** - Telefonverbindung  
- „Heißer Draht“ USA - Russland
  - ↪ Verbindung fest geschaltet, kein wählen nötig
- Dienstformen:
  - verbindungsorientierte Dienste
  - verbindungslose Dienste



---

# A) Verbindungsorientierte Dienste



---

# Verbindungsorientierte Dienste (1)

**Verbindungsorientierte Dienste** (*connection-oriented services*) beruhen auf dem Aufbau einer Verbindung zwischen den Dienstnutzern an den Dienstzugangspunkten.

- **Eigenschaften:** - Zuverlässige Übertragung der Daten  
- Wahrung ihrer Reihenfolge

☞ **Diese Eigenschaften werden vom Diensterbringer sichergestellt !**

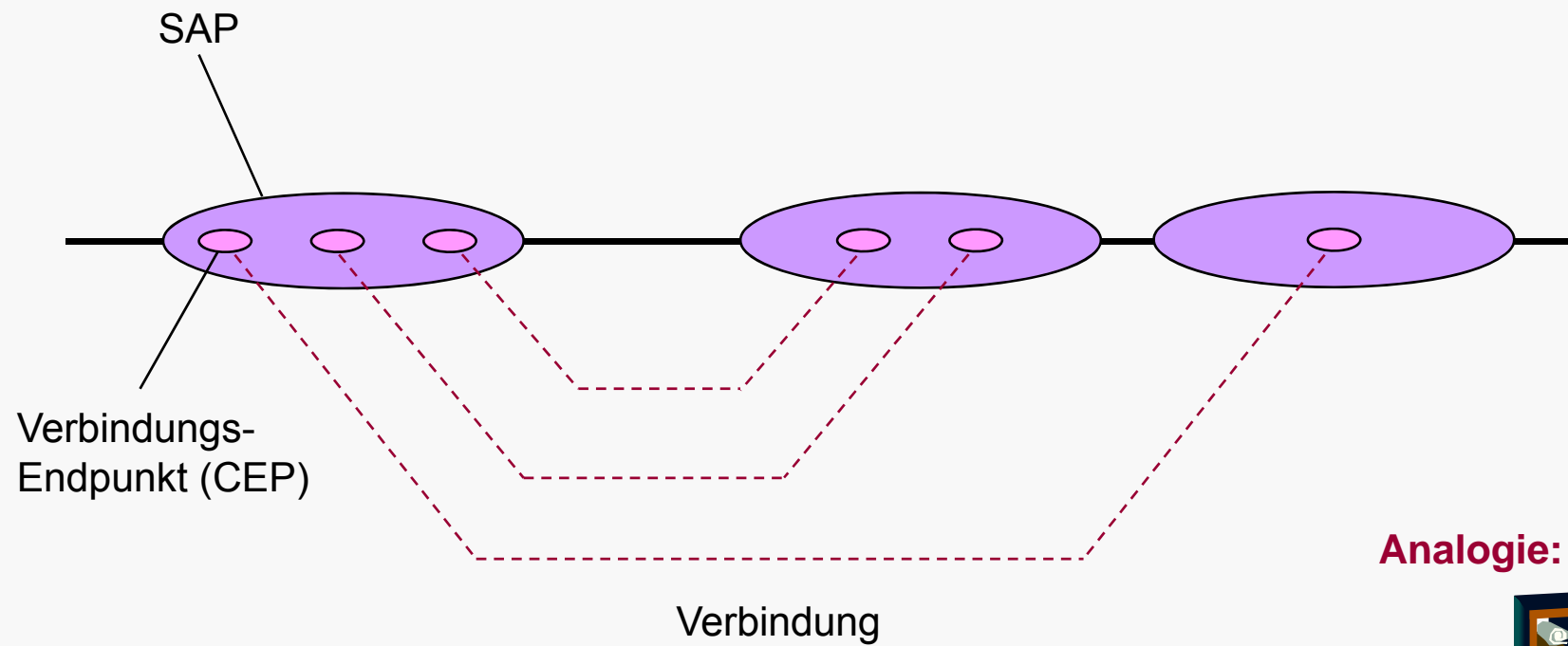
- Es können prinzipiell mehrere Verbindungen zwischen denselben oder zu verschiedenen Dienstzugangspunkten aufgebaut werden.

☞ **Verbindungsendpunkt-Identifikatoren** zur Unterscheidung der verschiedenen Verbindungen (nur OSI)  
→ CEP-Identifizier – connection endpoint identifier

☞ **Analogie:** Postfächer



# Verbindungsendpunkte



**Analogie:**



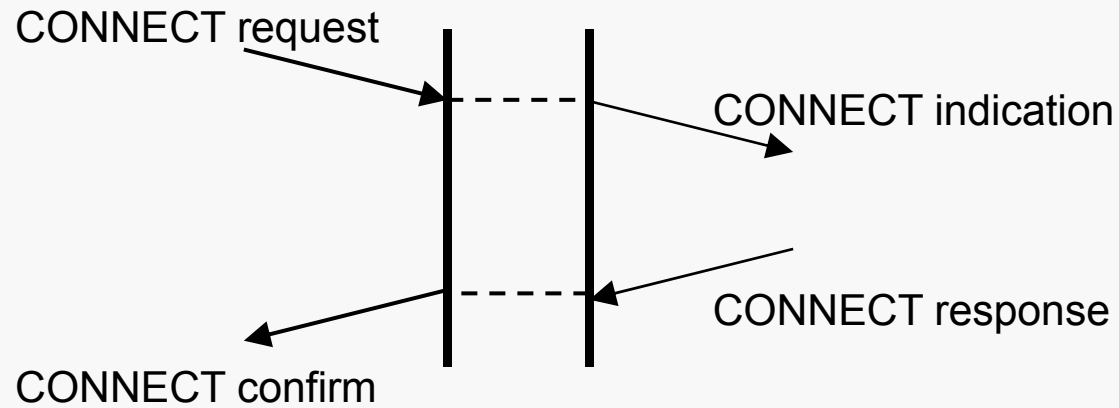
---

# Verbindungsorientierte Dienste (2)

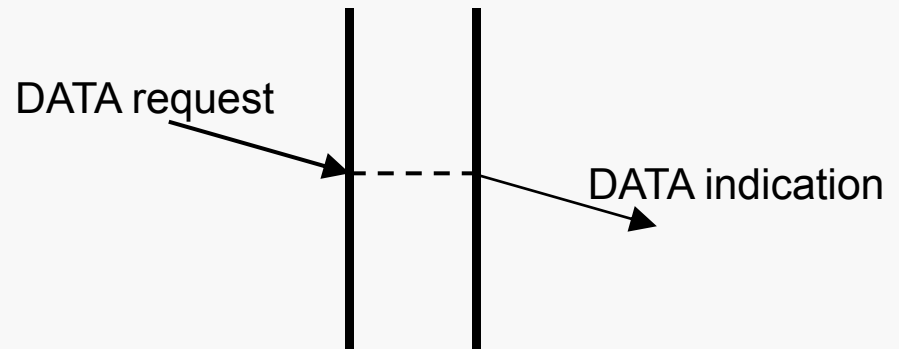
- Die **verbindungsorientierte Kommunikation** besteht aus 3 Phasen:
  - Verbindungsaufbau (*connection set up*)
    - ↪ bestätigter Dienst
  - Datenübertragung (*data transmission*)
    - ↪ unbestätigter Dienst
  - Verbindungsabbau (*connection release*)
    - ↪ eigentlich auch bestätigter Dienst (siehe Abschnitt 2.5)
  
- **Teildienste mit festen Abläufen an der Dienstschnittstelle**
  - ↪ können mit anderen Teildiensten zu einem anderen Dienst kombiniert werden



# Bestätigter vs. unbestätigter Dienst



a) Bestätigter Verbindungsaufbaudienst



b) Unbestätigter Datenübertragungsdienst

---

# Verbindungsorientierte Dienste

## ● Nachteile:

- Mehraufwand für Verbindungsaufbau/-abbau
- Mehraufwand Verbindungsverwaltung
  - ☞ Nicht benötigt bei seltenem Datenaustausch oder geringem Datenaufkommen !!!
- Zuverlässige Datenübertragung und Reihenfolgesicherung erfordern zusätzliche Maßnahmen
  - ☞ Bei einigen Anwendungen nicht erforderlich !!!
    - ↪ z.B. Videoübertragungen
  - ☞ Von einigen Anwendungen nicht tolerierbar !!!
    - ↪ z. B. Echtzeitkommunikation

☞ **Alternative: Verbindungslose Kommunikation**



---

## B) Verbindungslose Dienste





---

# Verbindungslose Dienste (1)

**Verbindungslose Dienste** (*connectionless services*) realisieren den Datenaustausch ohne den vorherigen Aufbau einer Verbindung, d. h., die Daten werden einfach an den Kommunikationspartner abgesandt.

↪ unbestätigte Dienste

● **Analogie:** - Briefzustelldienst  
- SMS

● **Eigenschaften:** - keine Einhaltung der Übertragungsreihenfolge  
- unzuverlässige Übertragung

☞ **Datenverluste müssen vom Dienstanutzer erkannt werden, um gegebenenfalls die Übertragung zu wiederholen !!!**



---

# Verbindungslose Dienste (2)

- **Einsatz:** - Übertragung geringer Datenmengen (z.B. Messdaten)
  - seltener Datenaustausch
    - ↳ z. B. Messwerterfassung
  - multimediale Daten
  - komplexe Datenströme

