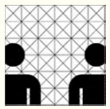
D Kommunikation & Rechnernetze

D1 Einführung und Motivation

D2 Technischer Überblick

D3 Lokale Rechnernetze

D4 Einige Gemeinsamkeiten von Betriebssystemen und Rechnernetzen: Architekturmodelle und Diensthierarchien



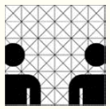
D “Kommunikation und Rechnernetze”

D4 Einige Gemeinsamkeiten von Betriebssystemen und Rechnernetzen

Bei Betriebssysteme und Rechnernetze haben (u.a.) gemeinsam:

- **Client-/Serverarchitekturen**
- **Interprozesskommunikation** (rechnerintern versus „remote“)
- **Betriebsmittelverwaltungsprobleme** (rechnerspezifische versus globale Betriebsmittel)
- **Schichtenstrukturen bzw. Diensthierarchien** → Fokus des vorliegenden Kapitels von GSS (4.2)
- **Sicherheitsanforderungen und -probleme** → vgl. GSS Teil II

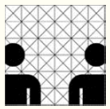
u.a.m.



D4.1 Client-/Serverarchitekturen, Interprozess-kommunikation & Betriebsmittelverwaltung

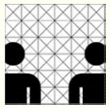
Client-/Serverarchitekturen

- ***geographisch zentralisiert (insbes. in zentralisierten Systemen):***
  - i.a. schneller, effizienter Zugriff ohne Kommunikationsengpass
  - vereinfachte Namensgebung und Adressierung
  - vereinfachte Behandlung von Ausfällen → *„rien ne va plus“*
  - evtl. Verarbeitungsengpass wegen der Nutzung gemeinsamer Ressourcen
- ***geographisch verteilt (insbes. in verteilten Systemen):***
  - evtl. aufwändiger Zugriff auf Server mit potentiellem Kommunikationsengpass bzw. Netzausfällen → *„Deadlock“*-Gefahr!
  - erschwerte Namensgebung und Adressierung (netzweit eindeutige Namen benötigt)
  - aufwändigere Behandlung von Ausfällen mit Möglichkeit der Realisierung von *„graceful degradation“*
  - vermehrte Verarbeitungsressourcen → möglicher Lastausgleich



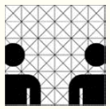
Interprozesskommunikation

- ***zentralisiert:***
  - i.a. schneller, effizienter Nachrichtenaustausch ohne Kommunikationsengpass; evtl. überdies mögliche Kommunikation über gemeinsamen Speicher (= „direkte Kopplung“ der Kommunikationspartner)
  - vereinfachte Namensgebung und Adressierung
  - vereinfachte Behandlung von Ausfällen → *„rien ne va plus“*
  - evtl. Verarbeitungsengpass wegen der Nutzung gemeinsamer Ressourcen
  - evtl. vereinfachte Synchronisation dank möglicher direkter Signalisierung zwischen Prozessen
- ***verteilt:***
  - Notwendigkeit der Behandlung von Nachrichtenverlusten/-verfälschungen → *„Deadlock“*-Gefahr !
  - erschwerte Namensgebung und Adressierung (netzweit eindeutige Namen, als Prozess-Id‘s, benötigt)
  - aufwändigere Behandlung von Rechnerausfällen mit ihren kommunizierenden Prozessen mit Möglichkeit der Realisierung von *„graceful degradation“*



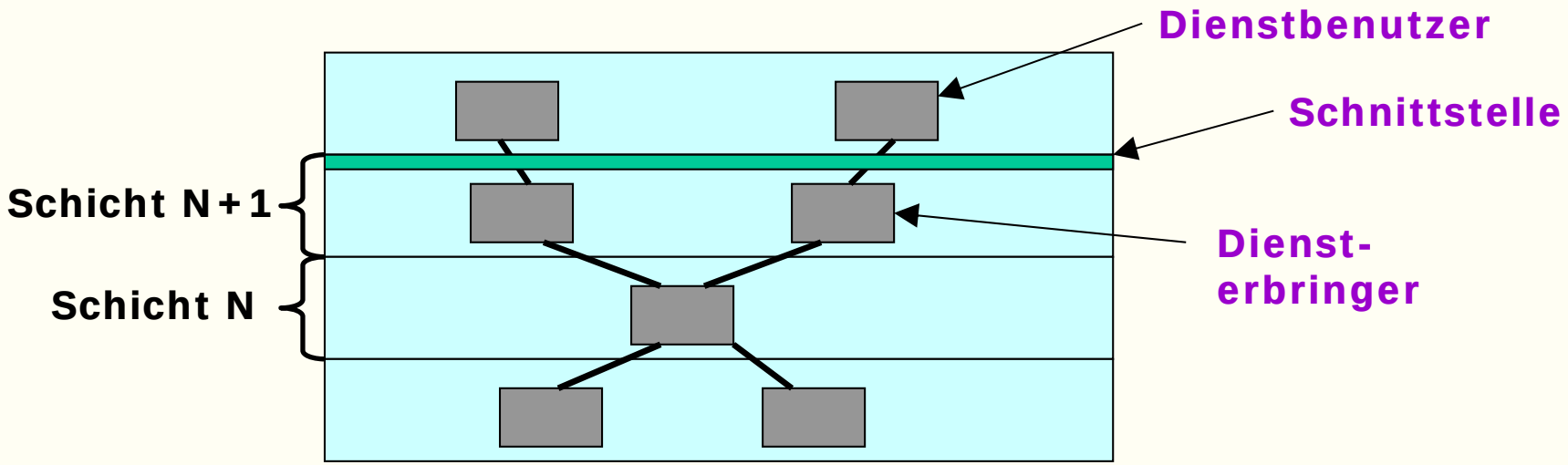
Betriebsmittelverwaltung

- ***zentralisiert:***
  - i.a. schneller, effizienter Betriebsmittel (BM)-Zugriff ohne Kommunikationsengpass
  - vereinfachte Namensgebung und Adressierung der BM
  - vereinfachte Behandlung von Ausfällen → *„rien ne va plus“*
  - i.a. deutlich exaktere Kenntnis der Betriebsmittelauslastung
- ***verteilt:***
  - Notwendigkeit der Behandlung von Nachrichtenverlusten/ -verfälschungen → *„Deadlock“*-Gefahr bei Allokation von Betriebsmitteln!
  - erschwerte Namensgebung und Adressierung (netzweit eindeutige Namen benötigt - zur Benennung und eindeutigen Adressierung)
  - aufwandsärmere Hinzunahme weiterer Ressourcen im Bedarfsfall
  - aufwändigere Behandlung von Rechnerausfällen mit ihren netzweit genutzten BM mit Möglichkeit der Realisierung von *„graceful degradation“*
  - i.a. deutlich unpräzisere Kenntnis der BM-Auslastung, da meist nicht aktuelle Sicht des Zustands der BM-Belegung und evtl. Verlust von Updates des Momentanzustands der BM-Belegung



D4.2 Rechnernetzarchitekturen: Modelle und Diensthierarchien

Merkmale von Schichtenmodelle und Diensthierarchien

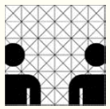


Dienst = Leistung einer Schicht, die „Benutzern“ einer höheren Schicht angeboten wird
(z.B. bei Rechnernetzen: *Übertragungsdienste*
bei Betriebssystemen: *Ein-/Ausgabedienste*)

Weshalb Schichtenmodelle und Diensthierarchien?

Vorteile:

- Vereinfachte Testbarkeit
- Standardisierbarkeit von Schnittstellen
- Details unterliegender Schichten (d.h. der Diensterbringung) können Benutzern verborgen werden
  → Virtualisierung, Transparenz
- Erhöhung der Sicherheit (z.B. Zugriffskontrolle an Schnittstellen)
- Austauschbarkeit von Schichten (bzw. ihrer Instanzen)
- Komplexitätsreduktion



Motivation und Ziele des OSI-Architektur-Modells

Def. „Offenes System“:

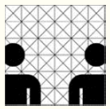
System von Rechnern, Software, Peripherie, Ü-Medien, dem *Standards für Informationsaustausch* mit anderen (offenen) Systemen zugrunde liegen.

→ herstellerübergreifend (heterogene Hard-/Software)

Die Standardisierung:

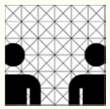
- Gremien
- Standardisierungsresultate
- Bedeutung von Standards ?

OSI : Open Systems Interconnection
(Interconnection ≅ Interaktion: Kommunikation und Kooperation)



Standardisierung bei (weltweiter) Kommunikation

- **Ziele** einer Standardisierung :
  - technische Kommunikation erfordert wohl definierte, *präzise* Regeln
  - wegen Heterogenität der Rechner: *herstellerübergreifende* Regeln
  - wegen Kopplung/Interkonnektion von Rechnernetzen: *netzübergreifende* Regeln
  - wegen weltweiter Liberalisierung der Telekommunikation: (internationale) *Absprachen* zwischen Netzbetreibern



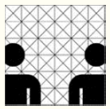
Standardisierung bei (weltweiter) Kommunikation

- **Vorteile** einer Standardisierung:
  - bei Beschränkung auf einen bzw. wenige Standards → große Stückzahlen (ergo: preisgünstige Komponenten)

    Bsp. Ethernetkarten, IP-Router, Switches, ...
  - sofern elementare Kommunikations- und Kooperationsdienste standardisiert → Schwerpunkt des Wettbewerbs auf höhere Ebenen, z.B. hin zu neuen Applikationen, verlagert
  - bei standardisierter „Kommunikationssteckdose" → große Flexibilität bei Produktion von Endsystemen (allerdings: Test auf Schnittstellenkompatibilität erforderlich)

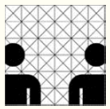
    [gewisse Analogie: Stromsteckdose]

- **Nachteile** einer Standardisierung:
  - Standards stellen Kompromisse dar
  - Weiterentwicklungen/Innovationen können behindert werden (nur „aufwärts-kompatible" Entwicklungen akzeptabel?!)
  - Standards evtl. komplex (zahlreiche Sonderfälle abzudecken) und daher z.T. ineffizient



Einige Standardisierungsgremien

- **ISO** (International Organization for Standardization)
  → u.a. OSI-Referenzmodell (Open Systems Interconnection, s.u.), ODP-Referenzmodell (Open Distributed Processing)
- **DIN** (in BRD), **ANSI** (in USA), **AFNOR** (in F), **BSI** (in GB) : nationale Entsprechungen zur ISO
- **CEN/CENELEC** (Comité Européen de Normalisation Electrotechnique)
  → Europäisches Komitee für Elektrotechnische Normung
- **DoD** (Dept. of Defence, USA) → militärische Standards
- **IEEE** (Institute of Electrical and Electronic Engineers) → u.a. LAN-Standards (wie Ethemet, Token-Ring, WLANs, ...)
- **CCITT** (Comité Consultatif International des Télégraphique et Téléphonique), später ersetzt durch **ITU** (Internat. Telecommunication Union) → u.a. Standards für „postalische“ (Vermittlungs-) Netze
- **ECMA** (European Computer Manufacturers Association) → Organisation europ. Rechnerhersteller (ergo : nicht nur Kommunikationsstandards)
- **ETSI** (European Telecommunications Standards Institute) → u.a. Standards für Mobilkommunikation
- **IETF** (Internet Engineering Task Force) → Standards des Internets



Zur Bedeutung von Standards

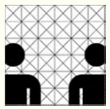
Mögliche Wege :

- *Standardisierungskomitee*
  a) auf Erfahrungen mit konkreten Realisierungen (Prototypen) basierend → z.B. Ethernet, X.25
  b) Standardisierung "am grünen Tisch" → z.B. einige OSI-Standards

- *de facto - Standards*
  a) Herstellerstandards → z.B. Rechnernetzarchitekturen wie SNA (IBM)
  b) herstellerunabhängige Realisierungen → z.B. Internet-Standards (insbes. TCP/IP)

Verbindlichkeit von Standards :

- in der Regel kein Zwang (Bsp. für Ausnahmen: Zugang zu öffentl. Netzen von Netzbetreibern wie Telekom, ISPs, US DoD etc.)
- unverzichtbar bei Kommunikation in heterogenen Systemumgebungen
- Offenheit wichtig, trotz Effizienzverlust
- adaptierbare Standards (z.B. durch Auswahlmöglichkeiten) flexibel, aber i.a. komplex und evtl. inkompatible Implementierungen



Allgemeines zur Strukturierung von Rechnernetzen

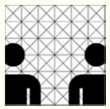
Modellierte ***Rechnernetzkomponenten*** u.a.:

- Endsysteme (Rechner mit kommunizierenden und kooperierenden Anwendungsprozessen)
- Transitsysteme (z.B. postalisches Vermittlungsnetz)
- Relais (z.B. postalischer Vermittlungsrechner)
- Übertragungsmedium (z.B. Koaxialkabel, Funkstrecke, Glasfaser).

Anwendungsprozesse (auf Endsystemen ablaufend) zerlegt in:

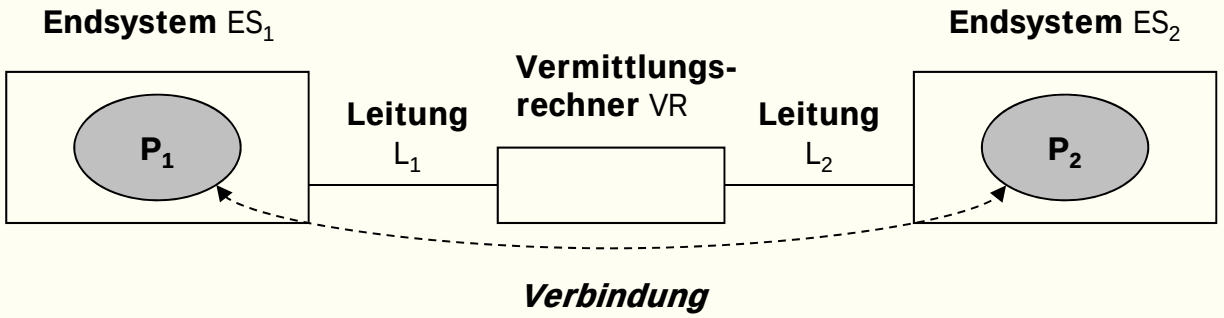
- kommunizierenden Teil
- lokalen Teil (→ lokale Verarbeitungsfunktionen).

Verbindungen im Sinne von Kommunikationsbeziehungen.

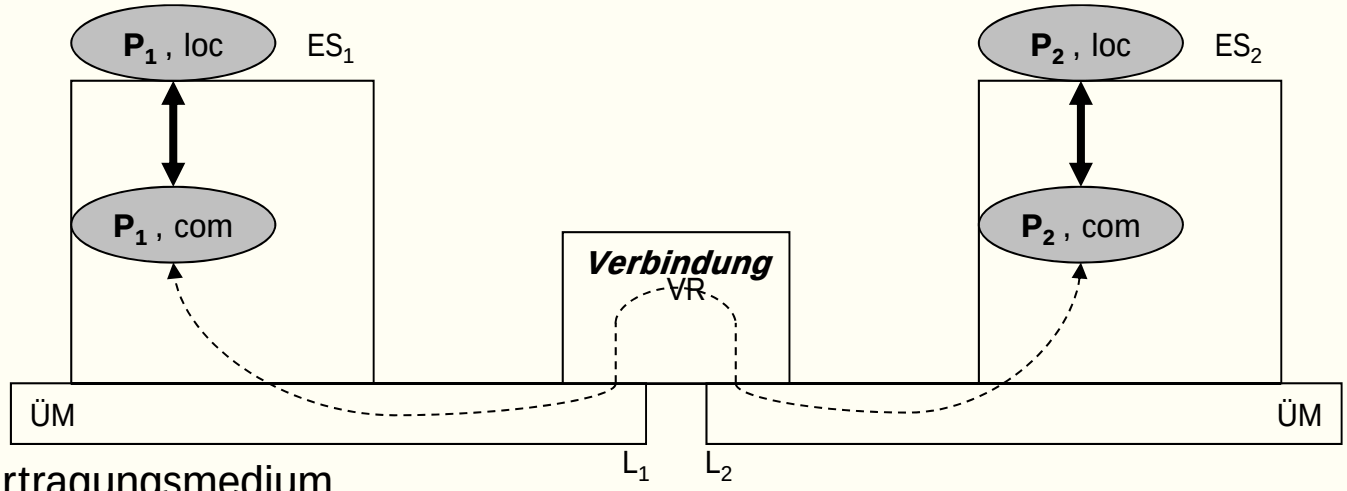


Verbindungsorientierte Kommunikation

Konfigurationsbeispiel:

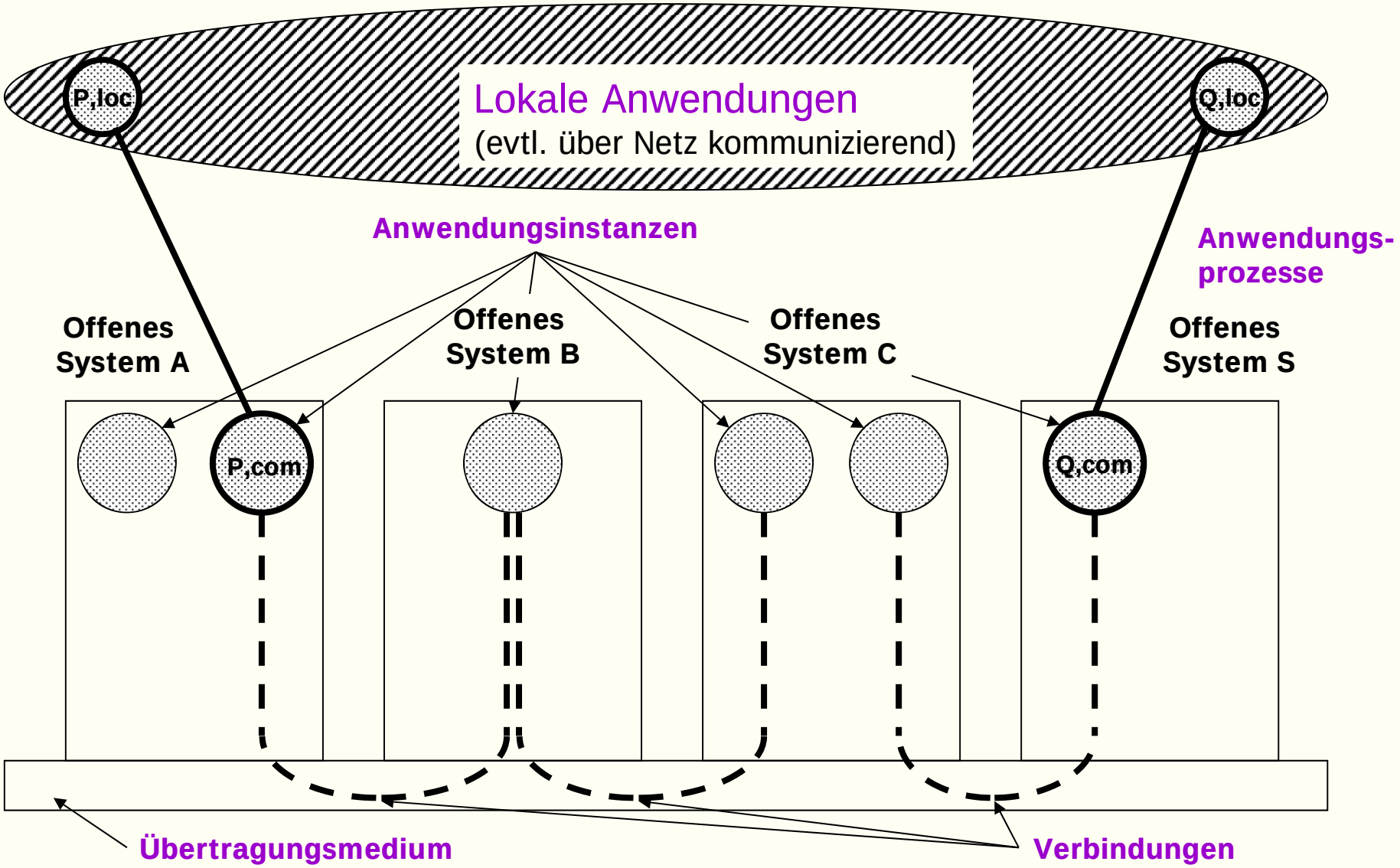


→ OSI-Schicht (grob):

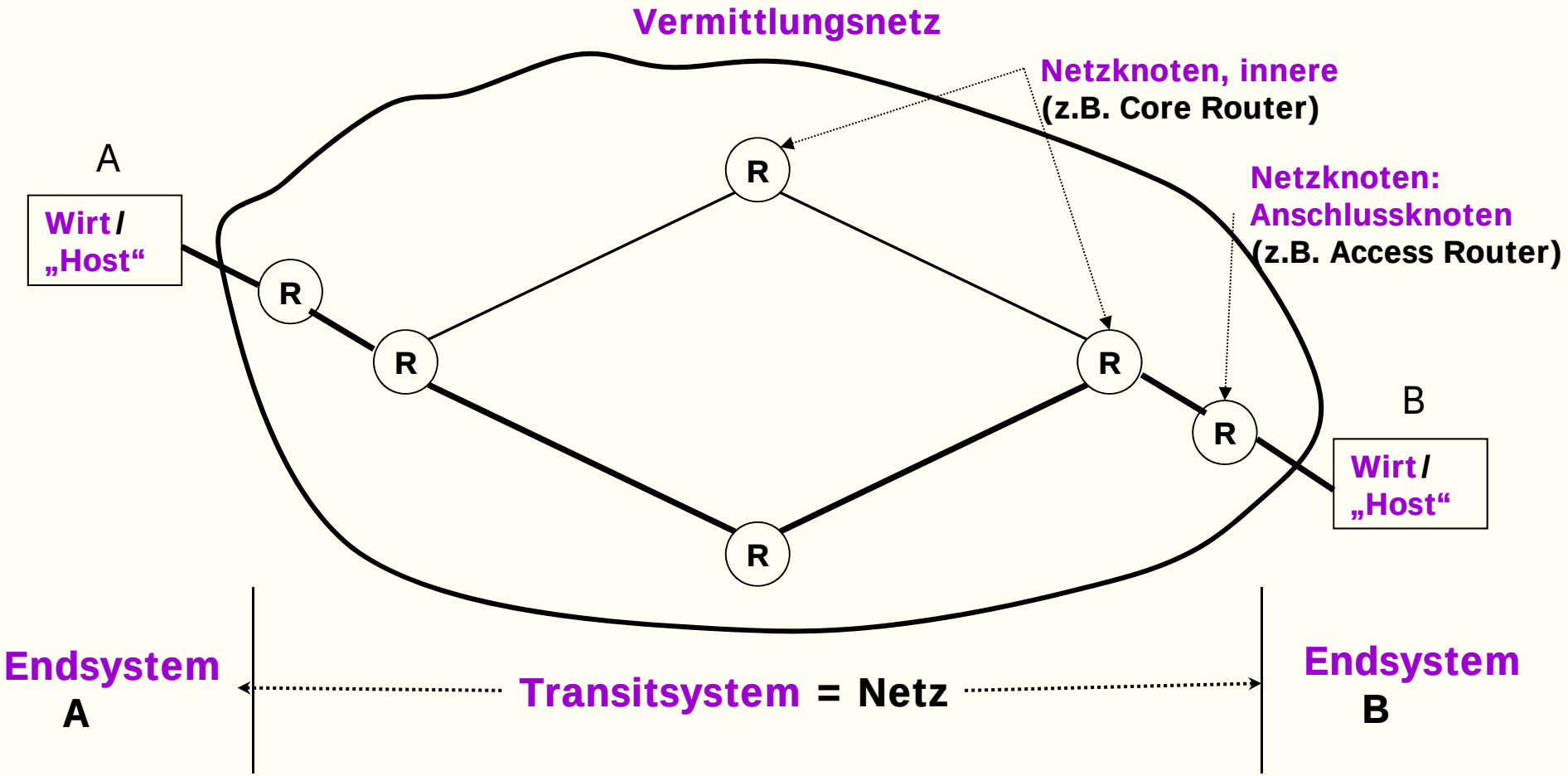


ÜM: Übertragungsmedium

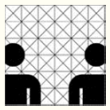
OSI-Basisbegriffe



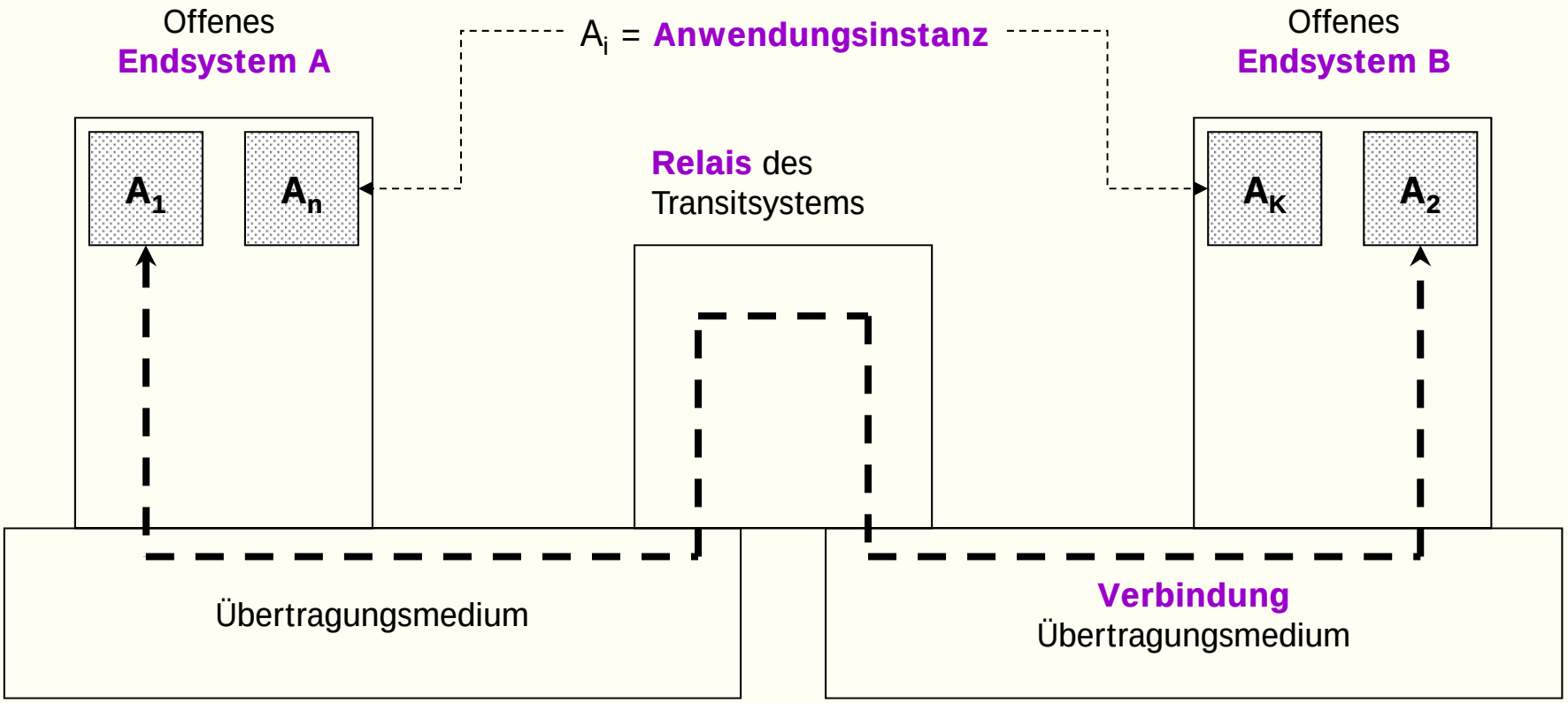
Schematische Darstellung einer Verbindung (stark gezeichnet) **im Netz**



R: Relais, auch: Vermittlungsrechner (z.B. Router im Internet)

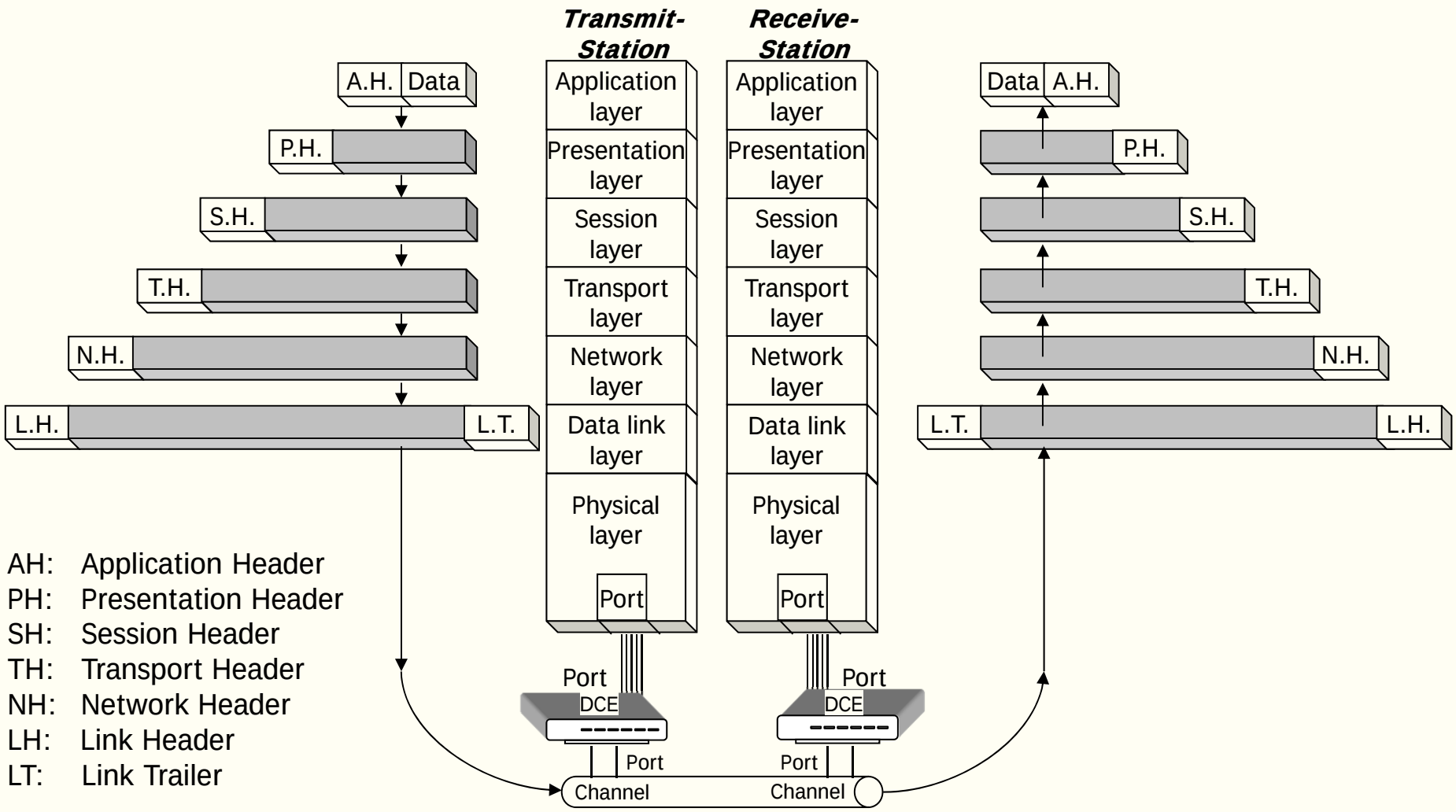


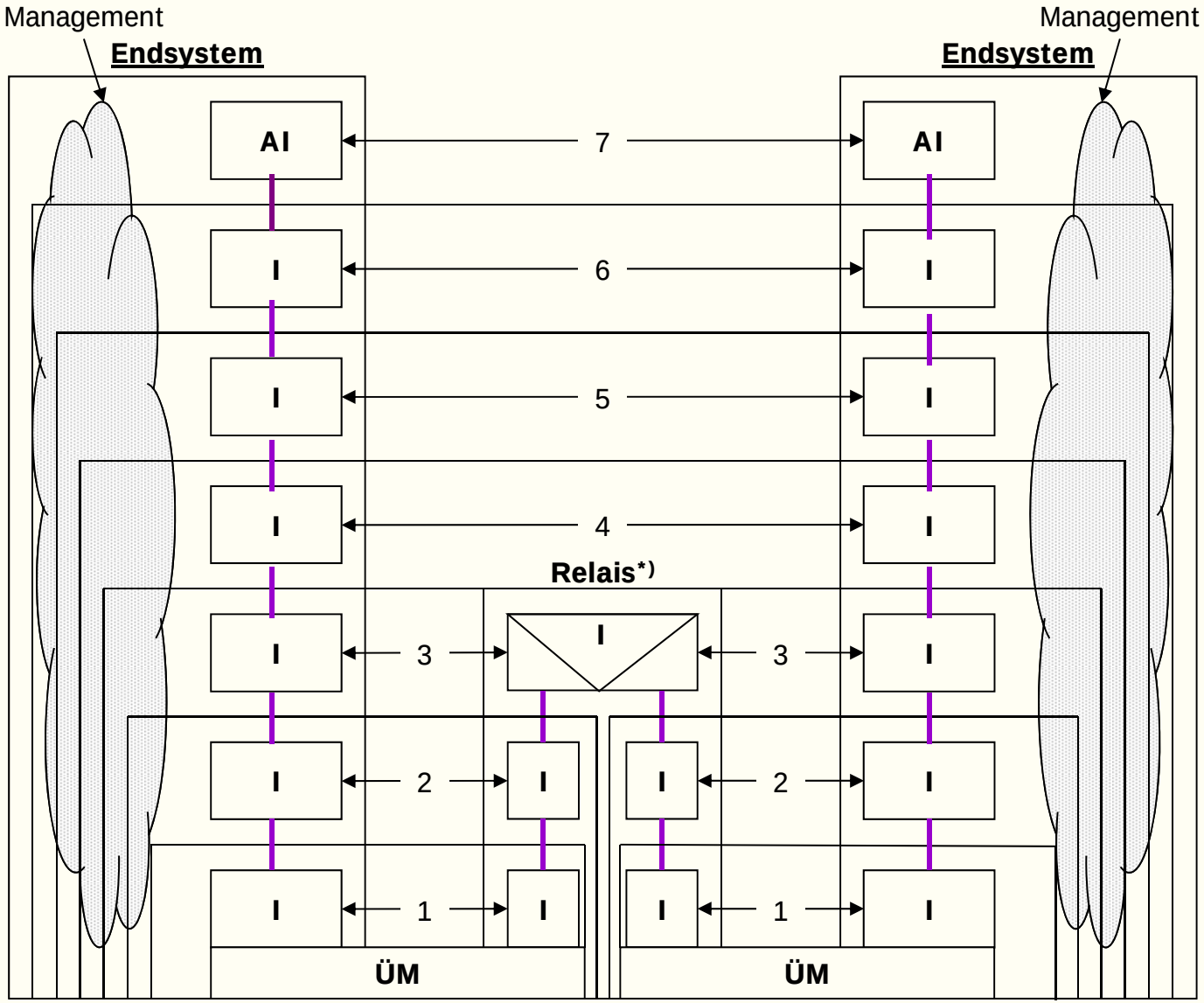
Schematische Darstellung einer Verbindung im Schnitt



A_1 / A_2: kommunizierende und kooperierende Anwendungsinstanzen

Schichtenstruktur in Rechnernetzen und daraus resultierender Datenfluss





*) *Endsysteme* können *direkt oder über mehrere Relais verbunden* sein.

AI ... Anwendungsinstanz
I ... Instanz

Schichten

Anwendungsschicht
S7: → z.B. Dateitransfer

Darstellungsschicht
S6: → z.B. Behandlung unterschiedl. Zahlendarstellungen

Kommunikationssteuerungsschicht
S5: → z.B. Dialogfunktionen

Transportschicht
S4: → z.B. Fehlerkontrolle

Vermittlungsschicht
S3: → z.B. Wegeermittlung durch Vermittlungsnetz

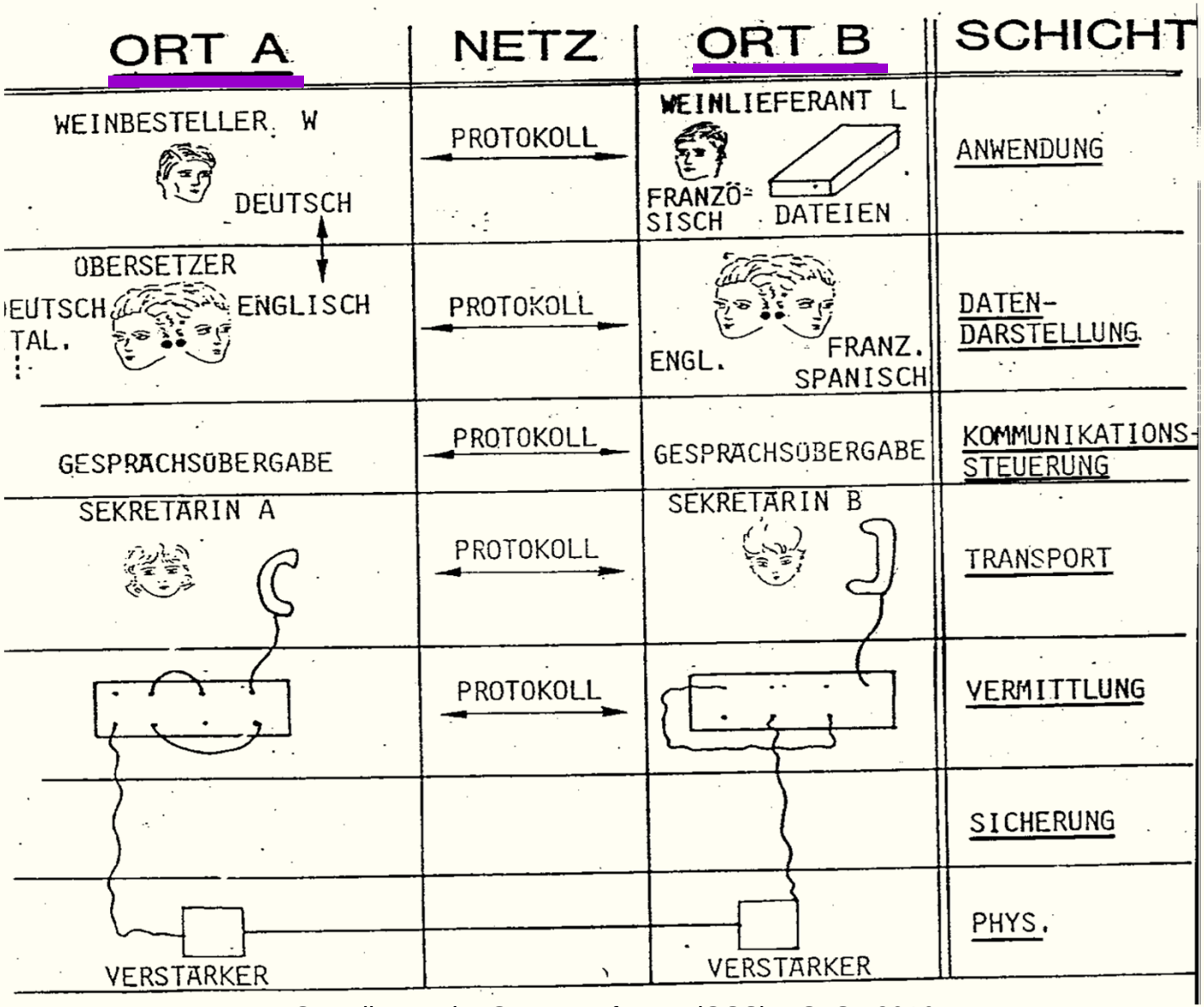
Sicherungsschicht
S2: → z.B. Fehlerkontrolle zwischen direkt verbundenen Rechnern

Bitübertragungsschicht
S1: → insbes. Austausch von Bitströmen

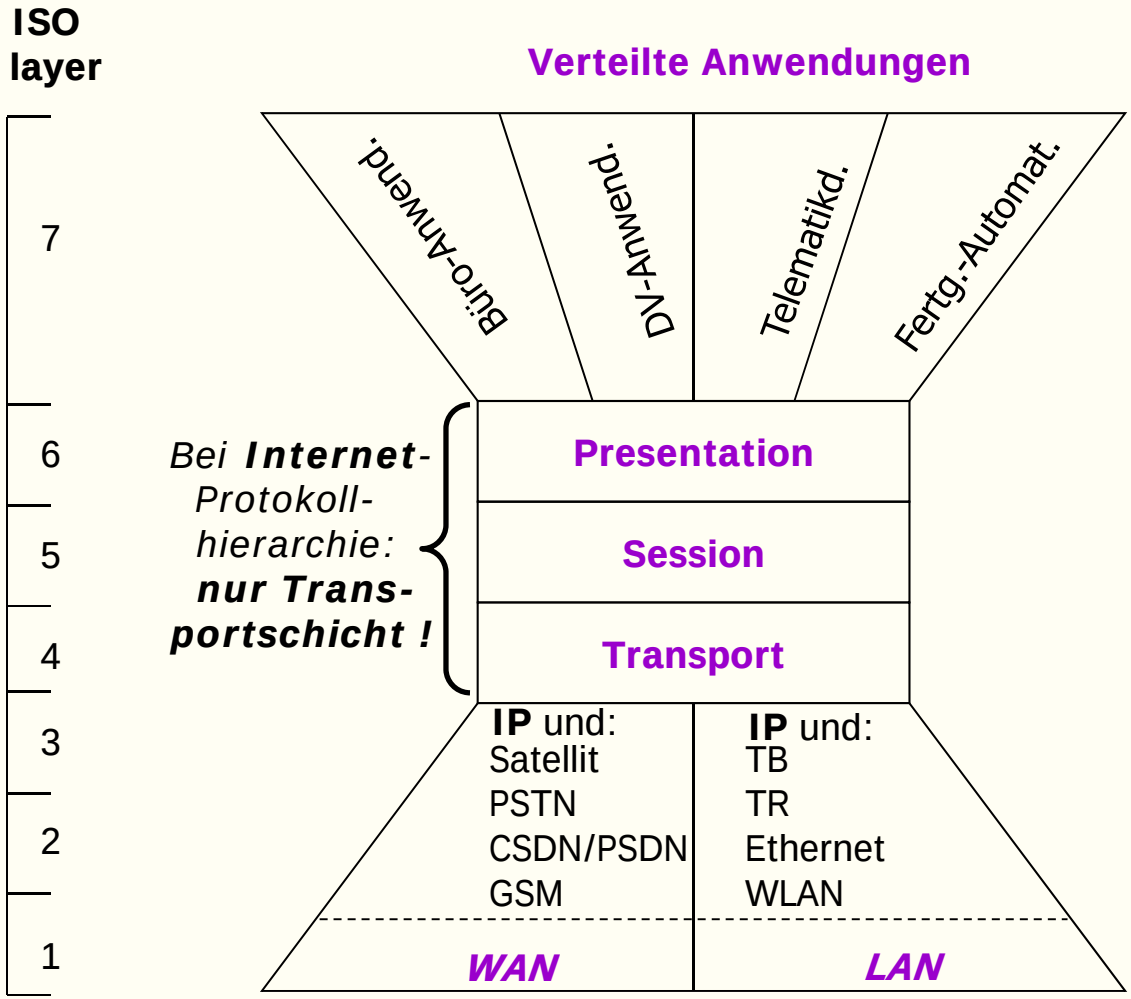
Übertragungsmedium
„S0“: → insbes. Austausch elektr./optischer Signale

Die Schichten und Protokolle des Referenzmodells

Analogie zum OSI-Architektur-Modell

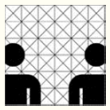


Real World Reference Model

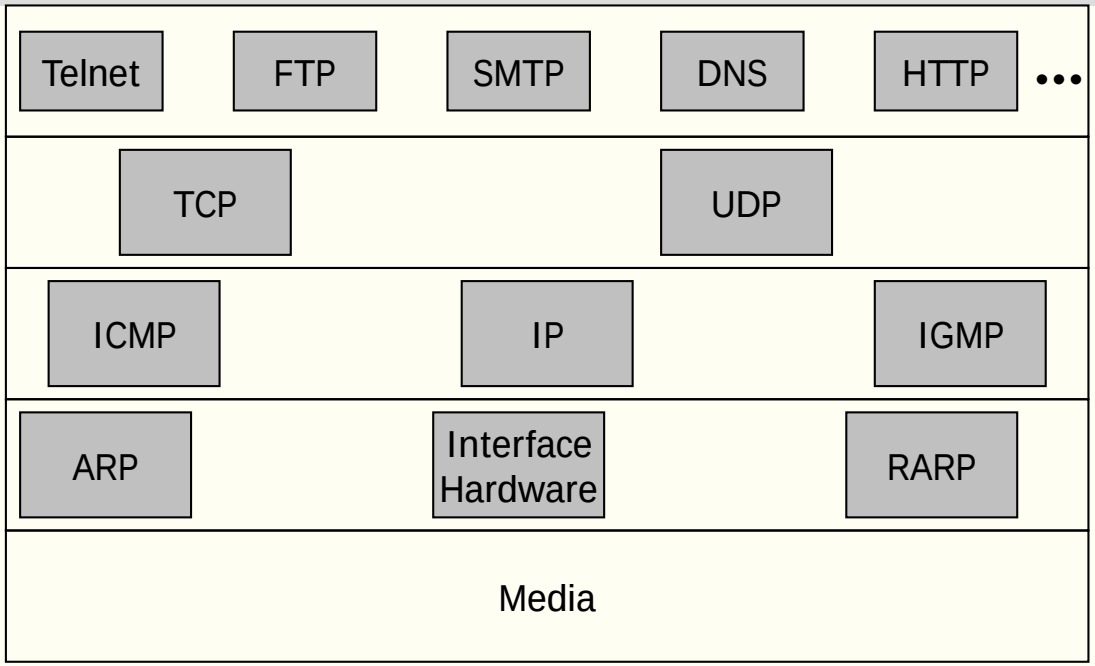


Abkürzungen:

- **LAN / WAN**:
  Local-/**W**ide-**A**rea **N**etwork
- **TB / TR**:
  Token **B**us/**T**oken **R**ing
- **WLAN**:
  Wireless **LAN**
- **GSM:**
  Global **S**ystem for **M**obile Communications
- **CSDN / PSDN**:
  Circuit/**P**acket **S**witched **D**ata **N**etwork (Leitungs-/Paketvermitt-lungsnetz)
- **PSTN**:
  Public **S**witched **T**ele-phone **N**etwork (öffentl. Fernsprechnetz)



Die Protokollhierarchie des Internet



Anwendungsschicht
Application Layer

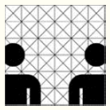
Transport ~
Transport L.

Vermittlungs ~
Network L.

Datensicherungs ~
Data Link L.

Physikalische ~
Physical L.

Telnet:	virtueller Terminaldienst
FTP:	**F**ile **T**ransfer **P**rotocol → Dateitransfer
SMTP:	**S**imple **M**ail **T**ransfer **P**rotocol → Electronic Mail
DNS:	**D**omain **N**ame **S**ystem → Konvertieren Hostnamen in Netzadressen
HTTP:	**H**yper **T**ext **T**ransfer **P**rotocol → WWW
TCP:	**T**ransmission **C**ontrol **P**rotocol → verbindungsorient. Transportdienst
UDP:	**U**ser **D**atagram **P**rotocol → verbindungsloser Transportdienst
IP:	**I**nternet **P**rotocol → datagrammorient. Vermittlung
ICMP/IGMP:	**I**nternet (**C**ontrol **M**essage) / (**G**roup **M**anagement) **P**rotocol
ARP/RARP:	(**R**everse) **A**ddress **R**esolution **P**rotocol



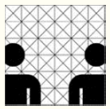
Phasen im Lebenslauf („Life Cycle“) eines Netzes

1. **Strategische Phase** → Netzeinrichtung
   Aufgaben:
   - Installation von
     - Ü-Medien
     - Kommunikationshardware/-software
   - Festlegung der Adressierung

   (lt. OSI: „administrative actions“)

2. **Taktische Phase** → Verbindungsaufbau/-abbau
   Aufgaben:
   - Herstellen des Kontakts zwischen Kommunikationspartnern (im allgemeinen: kommunizierenden Prozessen)
   - Reservierung von Betriebsmitteln

3. **Operative Phase** → Betriebsablauf
   Aufgabe:
   - Datenaustausch über etablierte Verbindungen (*Verbindung* ≅ Kommunikationsbeziehung)



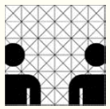
Phasen im Lebenslauf („Life Cycle“) **eines Netzes**, Forts.

→ aus o.g. Phasen resultierend:

- **statische Struktur** des Netzes (≅ vor Betrieb festgelegte Komponenten)
- **dynamische Struktur** des Netzes (≅ Ergebnis von Aktionen während taktischer/operativer Phase)

Bsp. Telefon:

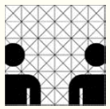
- *strategische* Phase → Einrichtung Anschluss
- *taktische* Phase → Anruf/Wählvorgang
- *operative* Phase → Gespräch



D4.3 Architekturmodelle für Rechnernetze: Die *statische Struktur* eines Netzes

Wesentliche Basiskonzepte für Rechnernetze (bzgl. statischer Struktur):

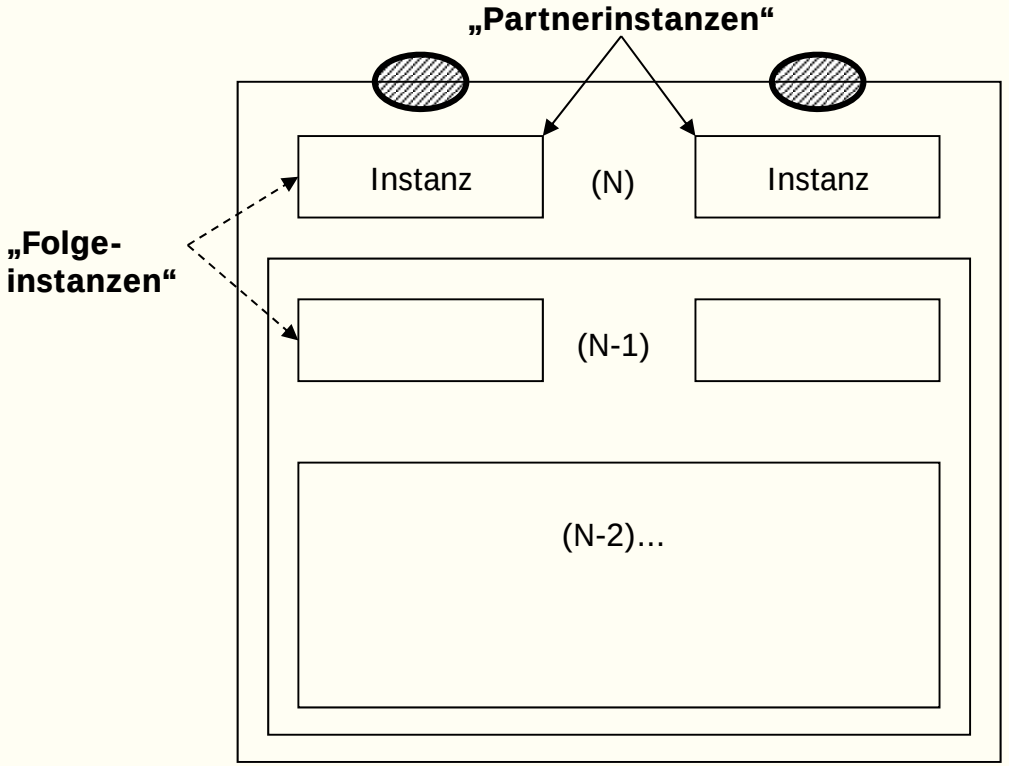
- **Dienst** (Telefon, DATEX-P, BTX, ISP, ...) – *„service“*
- **Protokoll** (Regeln der Benutzung) – *„protocol“*
- **Dienstzugangspunkt** (SAP) – *„service access point“*
- **Dienstelement** – *„service primitive“*
- **Protokolldateneinheit** (PDU) – *„protocol data unit“*
- **Instanz** – *„entity“*
- **Vorkommnis** – *„invocation“* (auch *„instance“*→ Vorsicht!)
- **Verbindungsendpunktkennung** (CEP-ID) – *„connection endpoint identifier“*



Der Begriff „Schicht“

Dienst = implementierungs*un*abhängige Beschreibung der Leistung einer Schicht

(N)-Dienst = Leistung der (N)-Schicht und der Schichten darunter

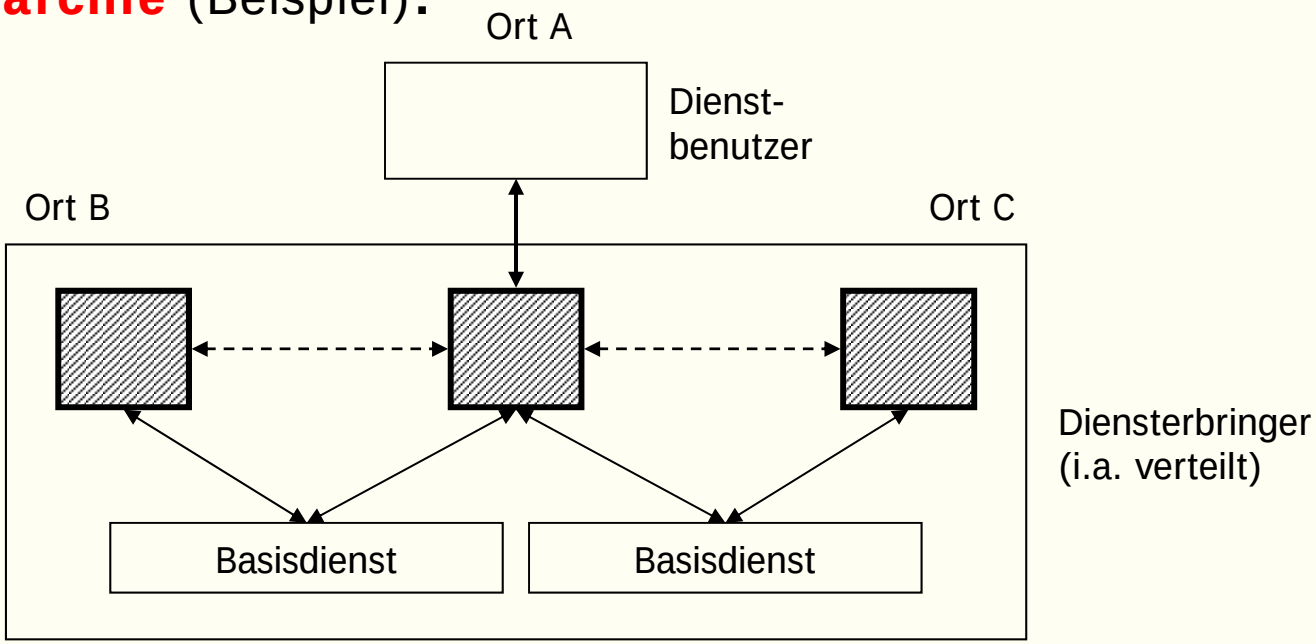


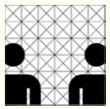
Das Basiskonzept „Dienst“

Dienst:

Leistung (Funktionsumfang) einer Schicht einer Rechnernetzarchitektur, die ein (in der Regel geographisch verteilter) Diensterbringer einem Dienst(be)nutzer – zumeist unter Rückgriff auf elementarere Basisdienste – bereitstellt.

Diensthierarchie (Beispiel):





Beispiele für Kommunikationsdienste

- **Datenkommunikation:**

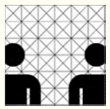
  Dateitransferdienst → Benutzer veranlasst z.B. Transfer einer Datei von Rechner R_1 nach R_2 (in LAN, in Internet, ...)

- **Sprachkommunikation:**

  Telefondienst → Fest- bzw. Mobilnetzbetreiber ermöglicht z.B. Ferngespräch zwischen Person P an Ort A und Person Q an Ort B = B(t), sofern Q ein Handy benutzt

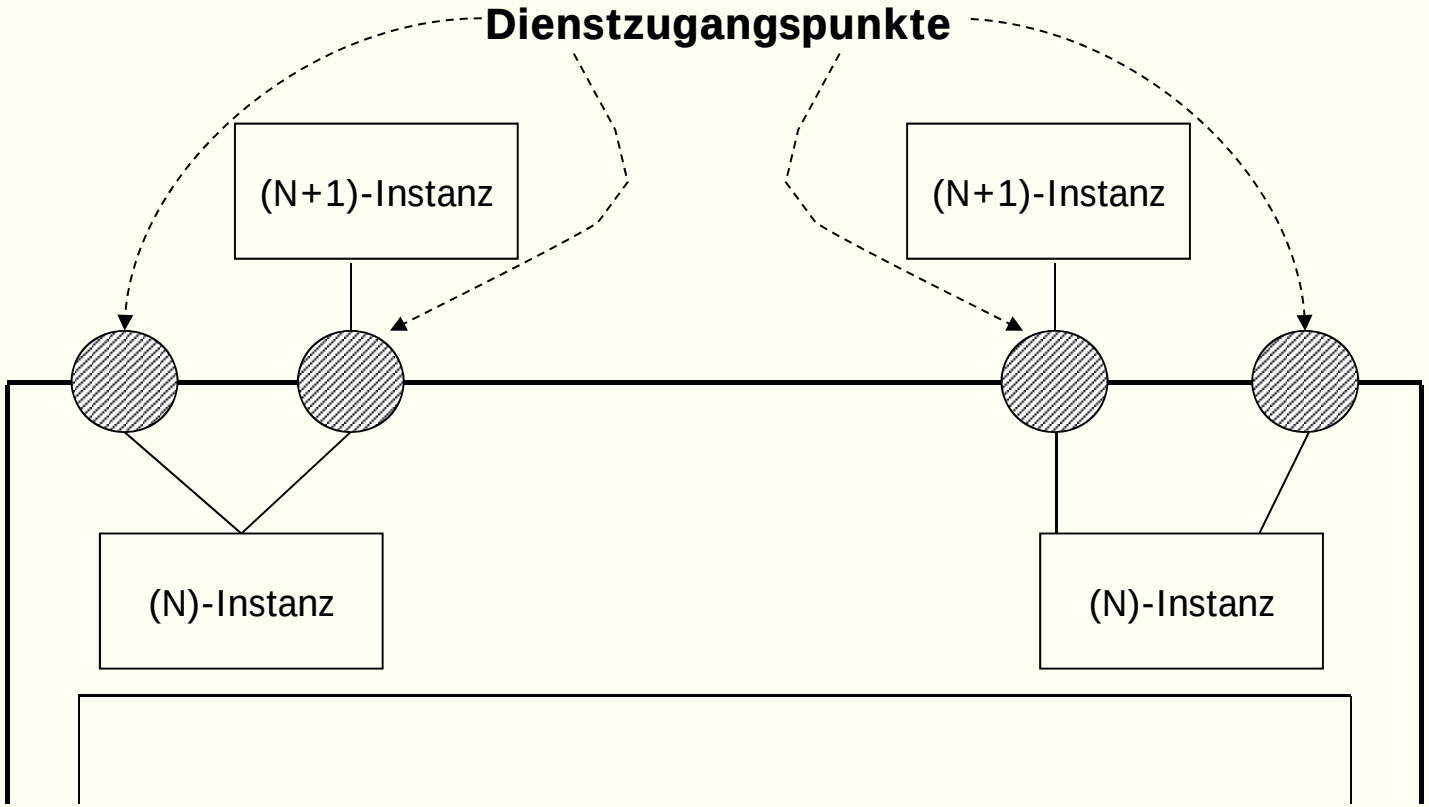
- **Audio-/Videokommunikation:**

  Video-on-Demand-Dienst → Echtzeitkommunikationsnetz ermöglicht Übertragung einer abgerufenen Videosequenz von einem Videoserver zum Bildschirm des abrufenden Benutzers



Dienstzugangspunkt („Service Access Point“, SAP)

- Jedem **Dienstzugangspunkt** ist eine **Adresse zugeordnet**
- **Nur eine Instanz** ist **oberhalb eines Dienstzugangspunktes** zugelassen (Adresse der darüberliegenden Instanz)



Resümee: Dienst, Dienstzugangspunkte, Dienstelemente

Dienst (Service)

Dienst (einer Schicht S_N) ≡ Leistung einer Schicht S_N inkl. Leistungen von Schichten S_j , $j<N$

Dienst der Schicht S_N wird S_{N+1} angeboten („Benutzer“ des Dienstes).

Bsp.: Dateitransferdienst, Datensicherungsdienst, …

Dienstzugangspunkt („Service Access Point“ – SAP):

Schnittstelle zwischen S_j und S_{j+1}, über die Dienste angefordert bzw. bereitgestellt werden.

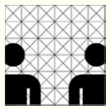
Bsp.: Implementierung von SAP über gemeinsamen Speicher, Prozeduraufruf, API-Schnittstelle (Application Programming Interface) o.ä.

Dienstelemente („Service Primitives“):

Kommandos zur Inanspruchnahme von Diensten ∪ Ergebnismeldungen.

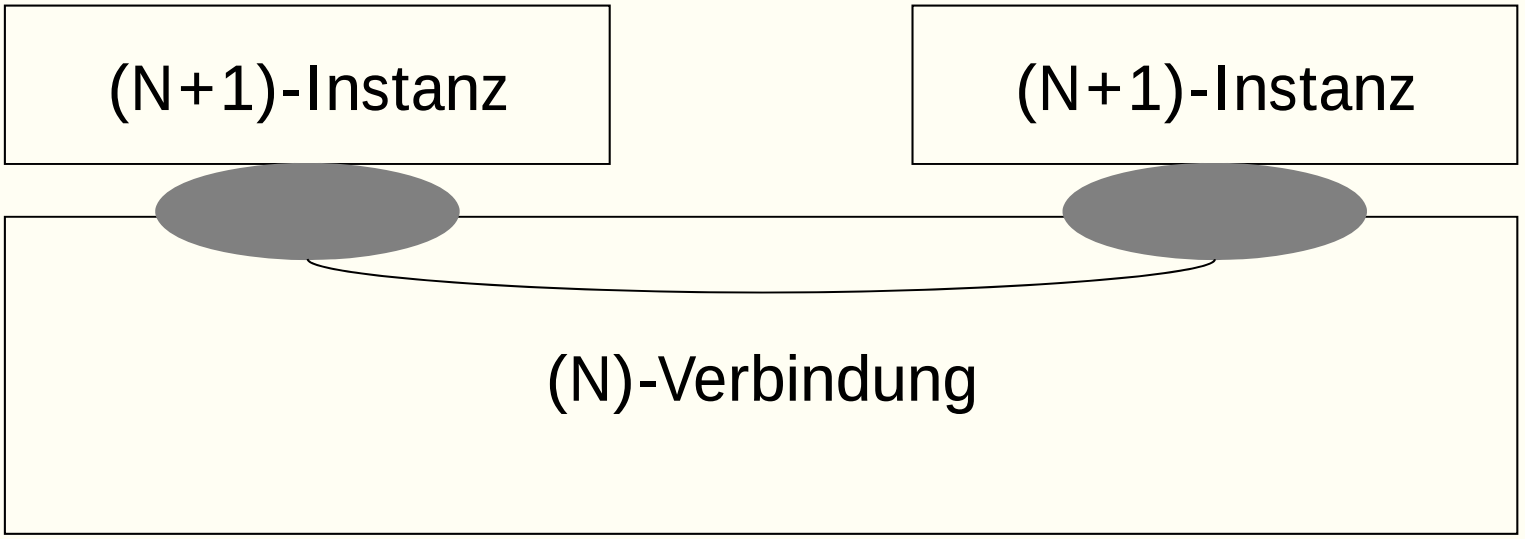
Bsp.: Dienstinanspruchnahme → CONNECT_REQUEST

Ergebnismeldung → CONNECT_CONFIRM (bei Verbindungsaufbau)

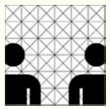


Operationsarten: *verbindungsorientierte* versus *verbindungslose* Kommunikation

a) verbindungsorientiert

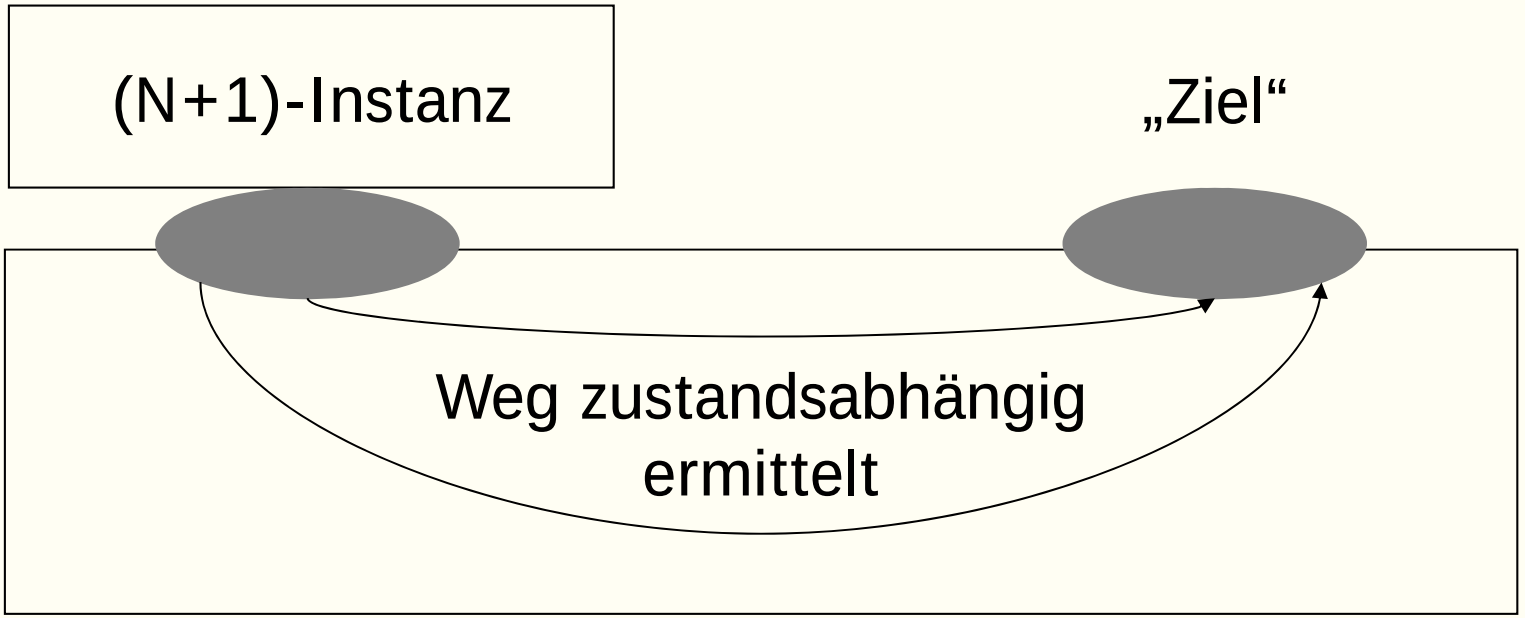


+ vereinfachte Adressierung nach Verbindungsaufbau
+ i.a. „höherwertiger“ Dienst (z.B. Reihenfolgeerhalt)
+ Möglichkeit der Nutzung von (verbindungsbezogener) Zustandsinformation

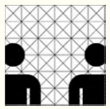


Operationsarten: *verbindungsorientierte* versus *verbindungslose* Kommunikation (Forts.)

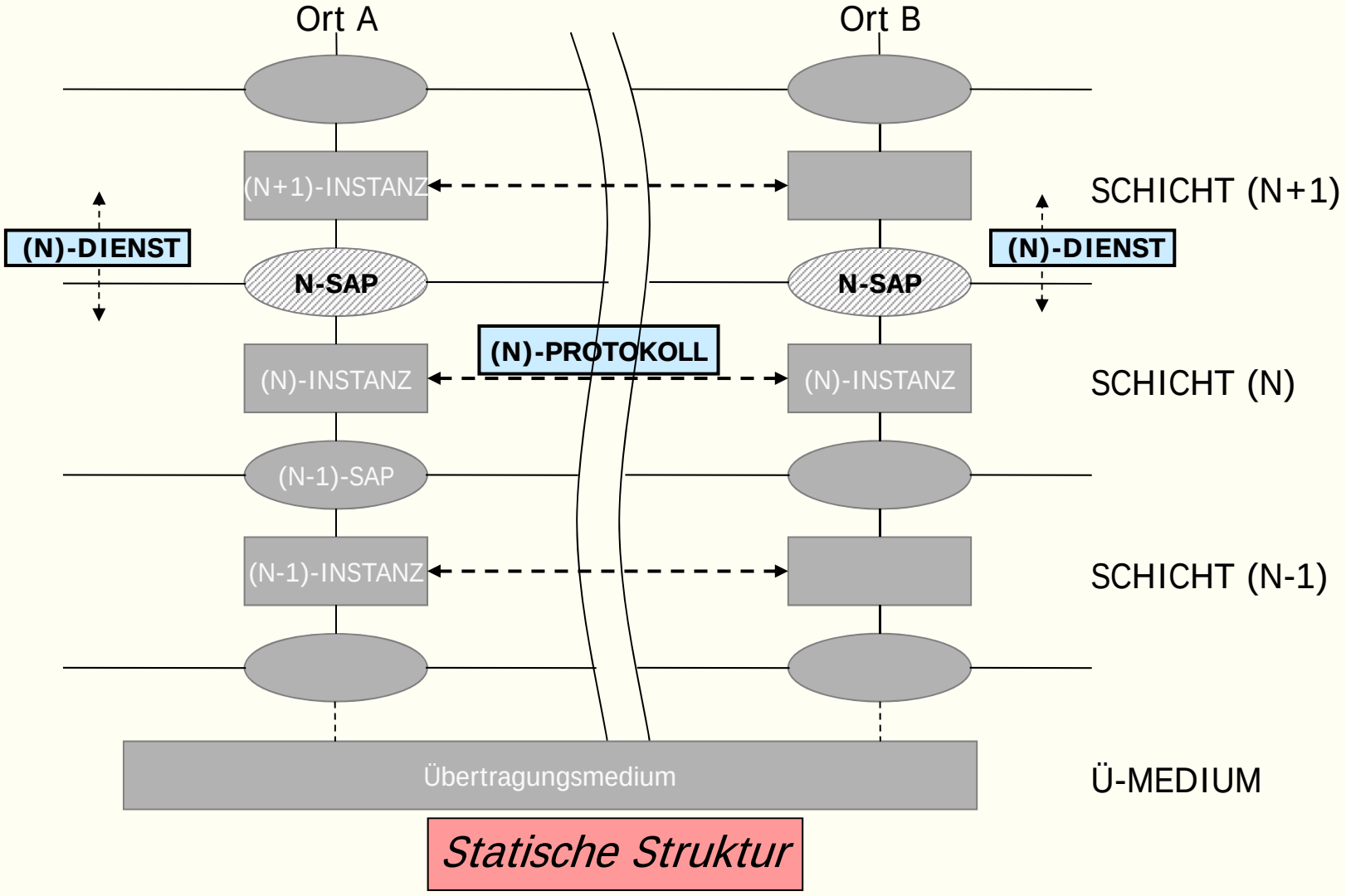
b) verbindungslos



\+ sofortiger Datenaustausch (ohne Verbindungsaufbau)



Dienst versus **Protokoll**



Das Basiskonzept „Protokoll“

Def. (Kommunikations-) Protokoll:

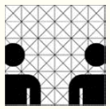
Regeln für die Kommunikation zwischen zwei oder mehreren Kommunikationspartnern (durch Austausch von Dateneinheiten).

Kommunikation zwischen

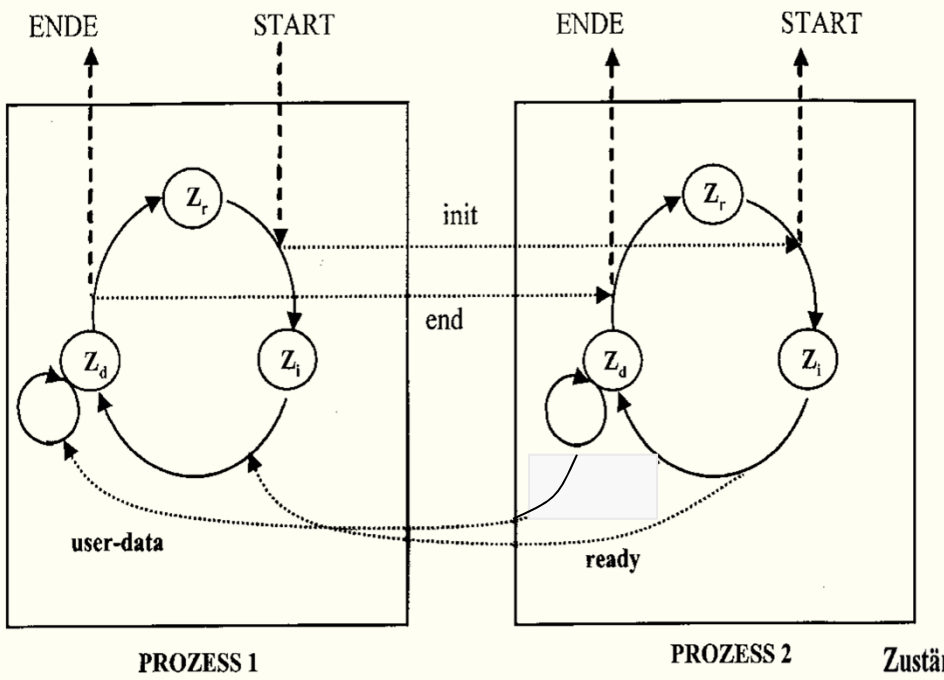
- 2 Partnern, z.B. bei Dateitransfer, Transportprotokoll wie UDP/TCP, ...
- ≥ 3 Partnern (im allg.), z.B. bei Videokonferenz, „Multicast“-Protokoll, ...

Wesentlich für Protokoll:

- **Syntax** der ausgetauschten (Protokoll-)Dateneinheiten
  → Struktur der Dateneinheiten, Formate, Codierung
- **Semantik** der ausgetauschten (Protokoll-)Dateneinheiten
  → Bedeutung der Dateneinheiten für Sender und Empfänger
- **„Timing“ / “(Ablauf-) Regeln“**
  → Festlegung der (grundsätzlich möglichen) zeitlichen Abläufe



Der „PROTOKOLL“-Begriff: Protokollinstanzen als *gekoppelte endliche Automaten*

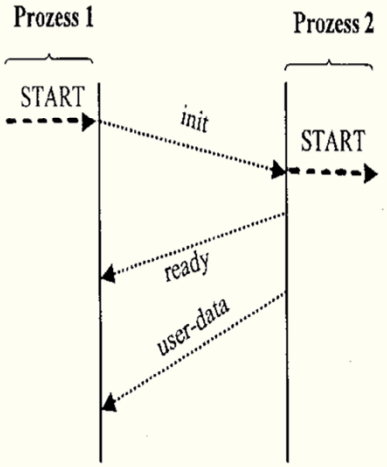


Dienstinteraktionen
→ vgl. Dienstelemente

Protokollinteraktionen
→ vgl. PDU-Austausch

Zustände:

- Z_r: Ruhezustand
- Z_i: Verbindungsaufbau initiiert
- Z_d: Datenaustauschphase

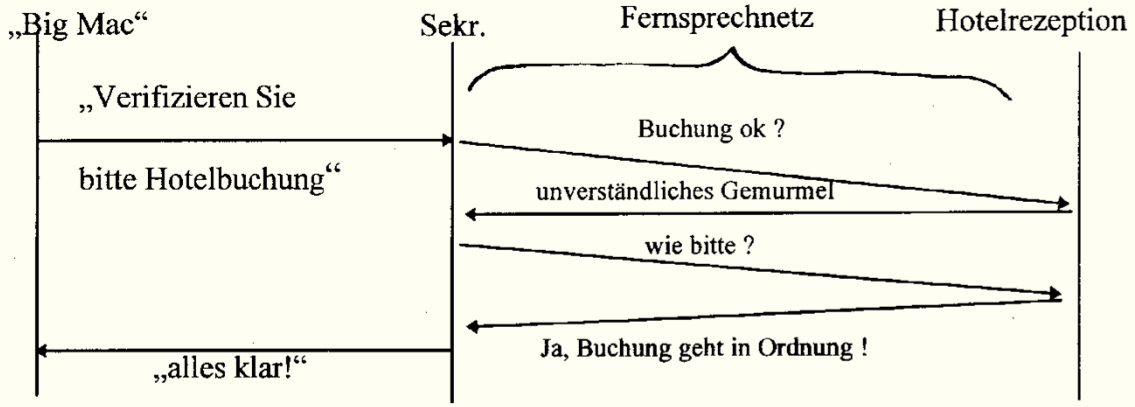


Deadlock-Gefahr in obiger automatenbasierter Protokollspezifikation?
(*nota bene:* Spezifikationen können „unvollständig“ sein !)

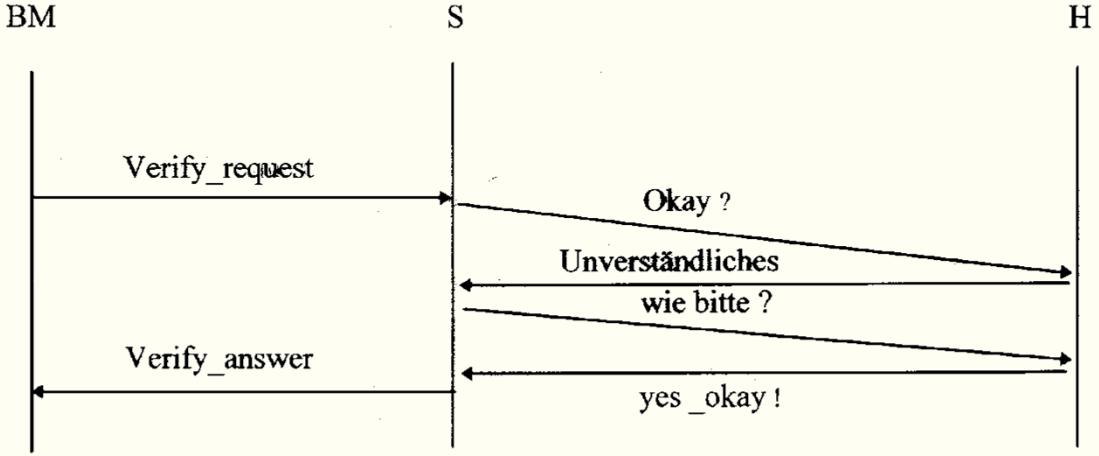
Formale Beschreibung einer Diensterbringung unter Berücksichtigung des verwendeten Protokolls

→ *Beispiel: Bestätigung einer Hotelbuchung*

- **Ort-Zeit-Diagramm**:



- **Ort-Zeit-Diagramm** (leicht formalisiert):

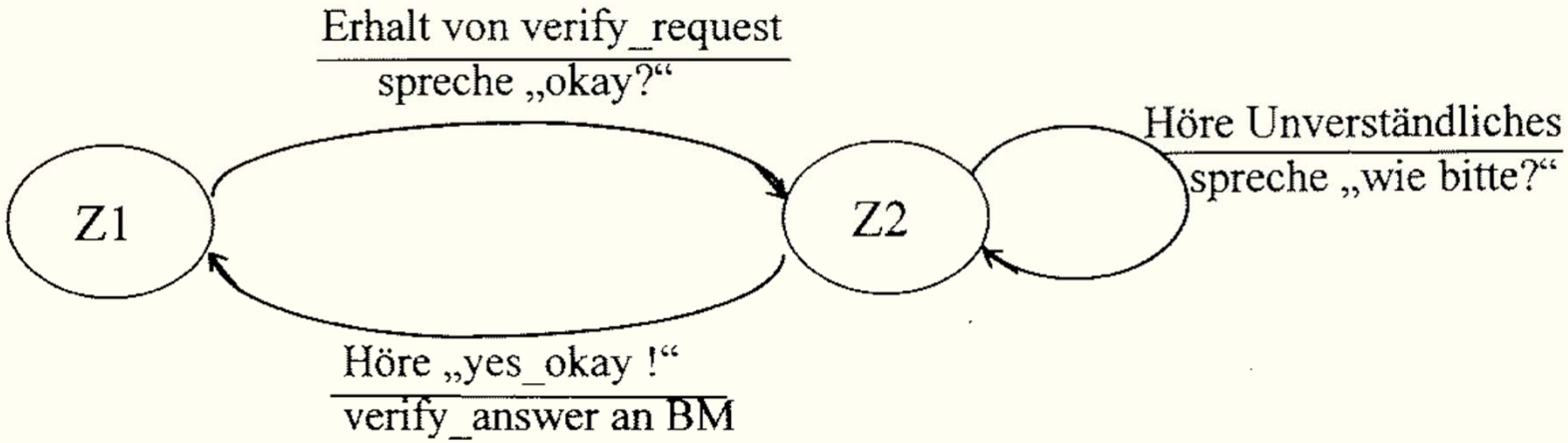


Formale Beschreibung einer Diensterbringung unter Berücksichtigung des verwendeten Protokolls (Forts.)

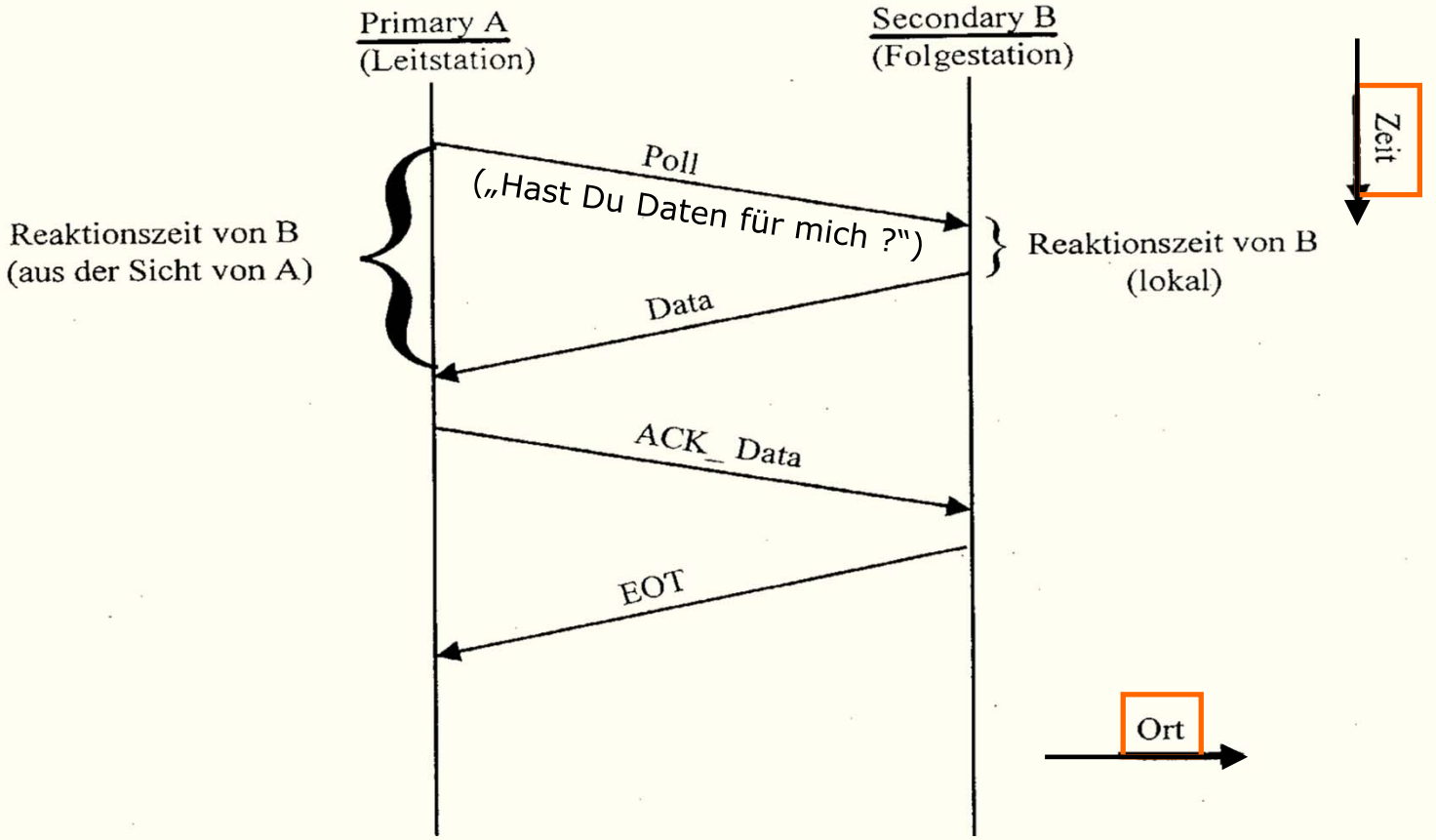
- **automatenbasierte, formale Beschreibung des Verhaltens** von Sekr.

  Zustände:

  - Z1: warten auf Auftrag des Chefs
  - Z2: warten auf Reaktion des Hotelrezeptionisten

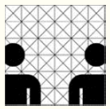


Semi-formale Beschreibung des Ablaufes von Protokollen mittels Ort-Zeit-Diagrammen



Vorteile: anschaulich, leicht interpretierbar

Nachteile: unvollständige Spezifikation → daher: ungeeignet als Grundlage für Protokollimplementierung, -verifikation

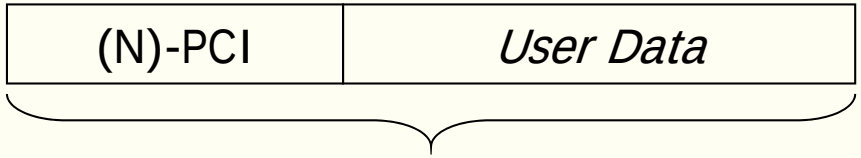


Informationsfluss zwischen angrenzenden Schichten

a) protokollrelevante Daten:

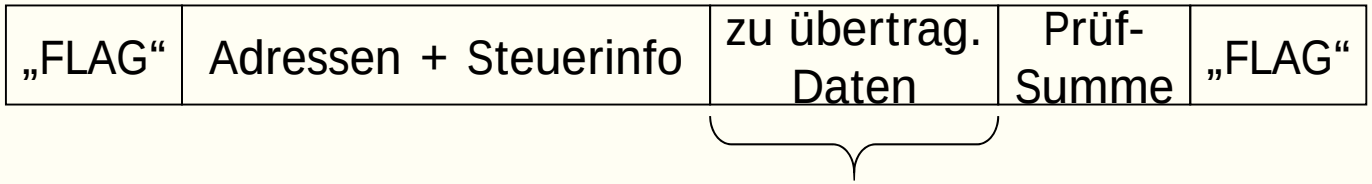
- **Protokolldateneinheit** (protocol data unit-**PDU**) der Schicht S_N: (N)-PDU
- **Benutzerdaten** (User Data) ≅ (N+1)-PDU aus Sicht von S_N
- **Protokoll-Steuerinformation** (Protocol Control Info.-**PCI**) zur Realisierung eines Protokolls

Modell:



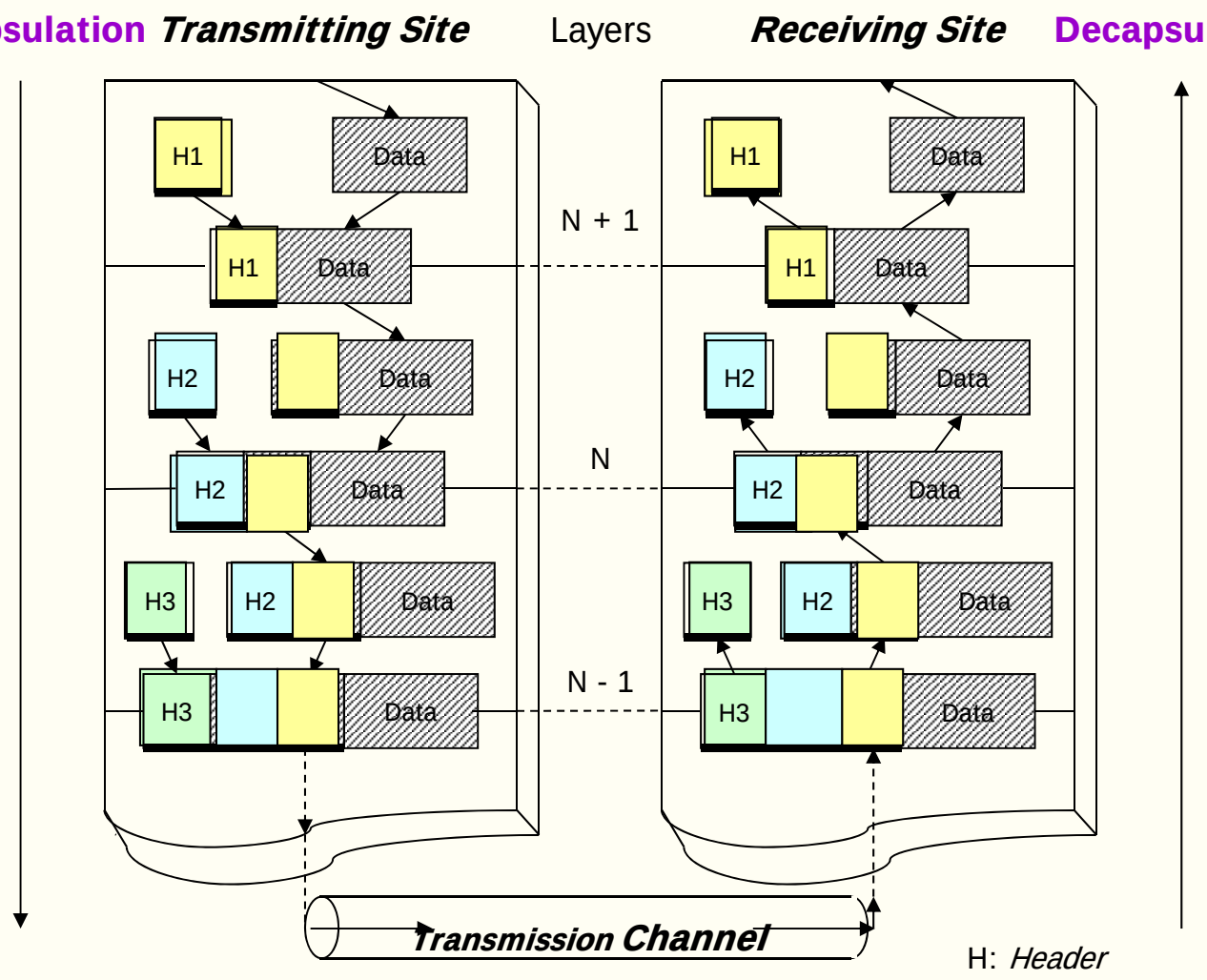
(N)-PDU

real: (HDLC)



user data

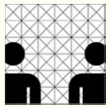
Datenfluss in Protokollhierarchie (vergröberte Sicht)



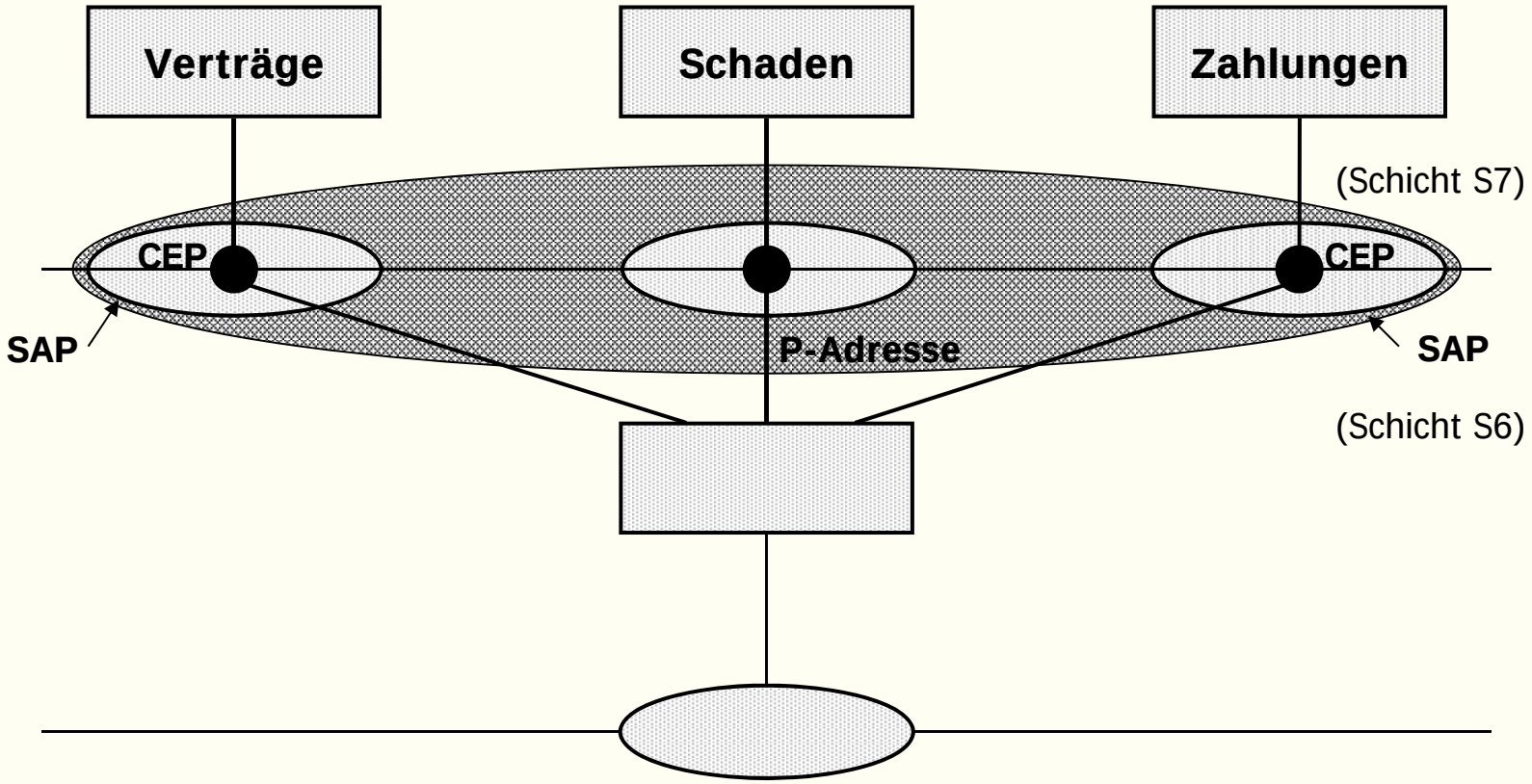
Analogie zu Encapsulation / Decapsulation bei konventioneller Briefpost

1. Brieftext in Briefumschlag (bei Unternehmenspost: z.B. ≥ 2 Briefe in größeren Umschlag zur Kostenreduktion; ident. Ziel)
2. Brief in Postsack (gemeinsam mit anderen Briefen eines gemeinsamen Post-Briefkastens)
3. Entpacken in Postamt (in HH) und Briefe von HH → M wieder in neuen Postsack P1 und andere Briefe von HH → S in weiteren, separaten Postsack P2
4. Postsäcke P1 und P2 in Transportflugzeug HH → F/M.
5. Transport Postsack P1 von F/M → M und Transport Postsack P2 von F/M → S
6. In München bzw. Stuttgart entpacken von P1 bzw. P2 und Verteilen der Briefe auf neue Postsäcke zur endgültigen Zustellung der Briefe an ihre Empfänger

... *dort:* Brieftexte aus Umschlag entnommen

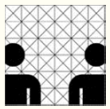


Zur Adressierung von Instanzen

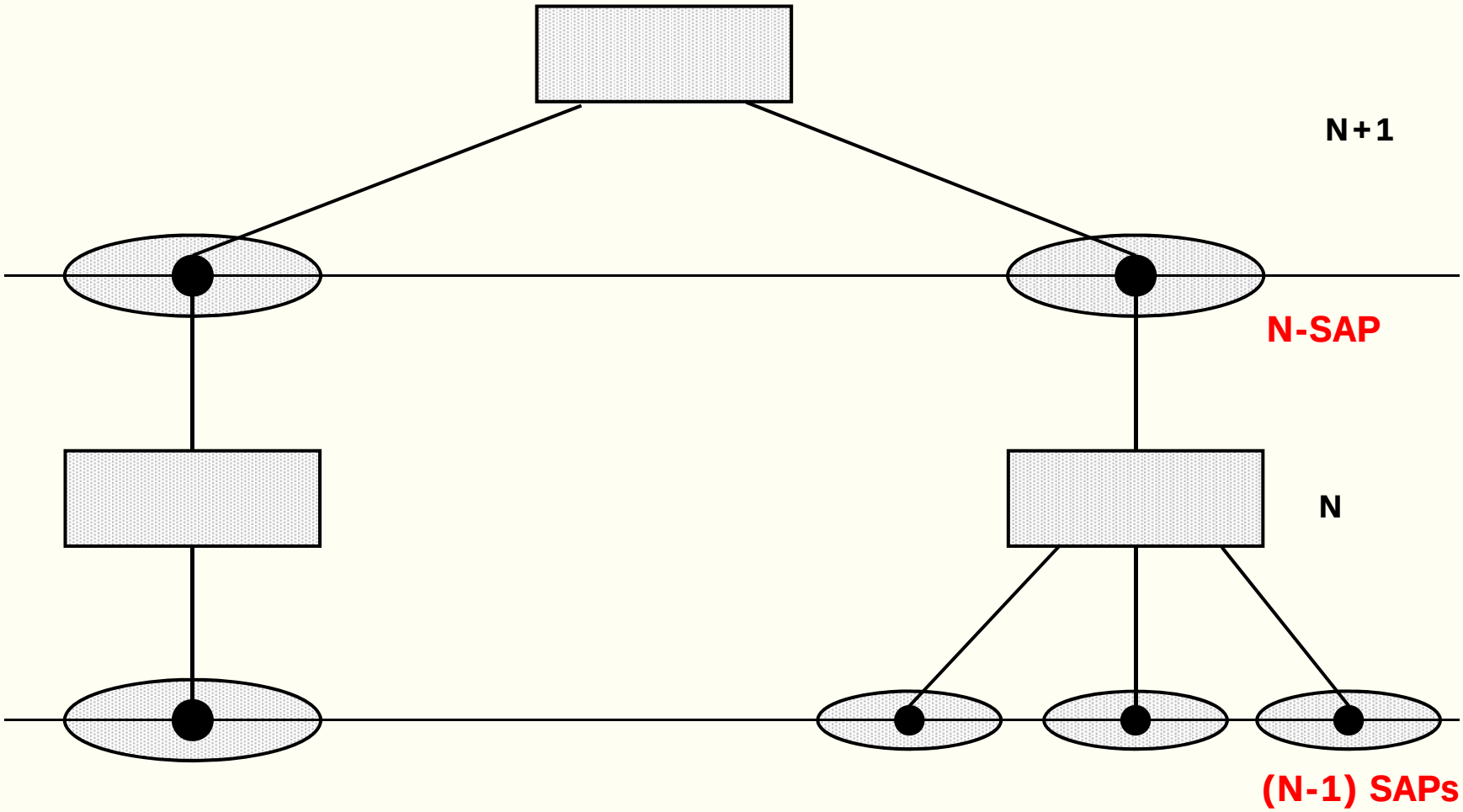


Verzweigungen **nach oben** innerhalb eines offenen Systems

[SAP ≡ Dienstzugangspunkt (s.o.); CEP ≡ Verbindungsendpunkt (s.u.)]

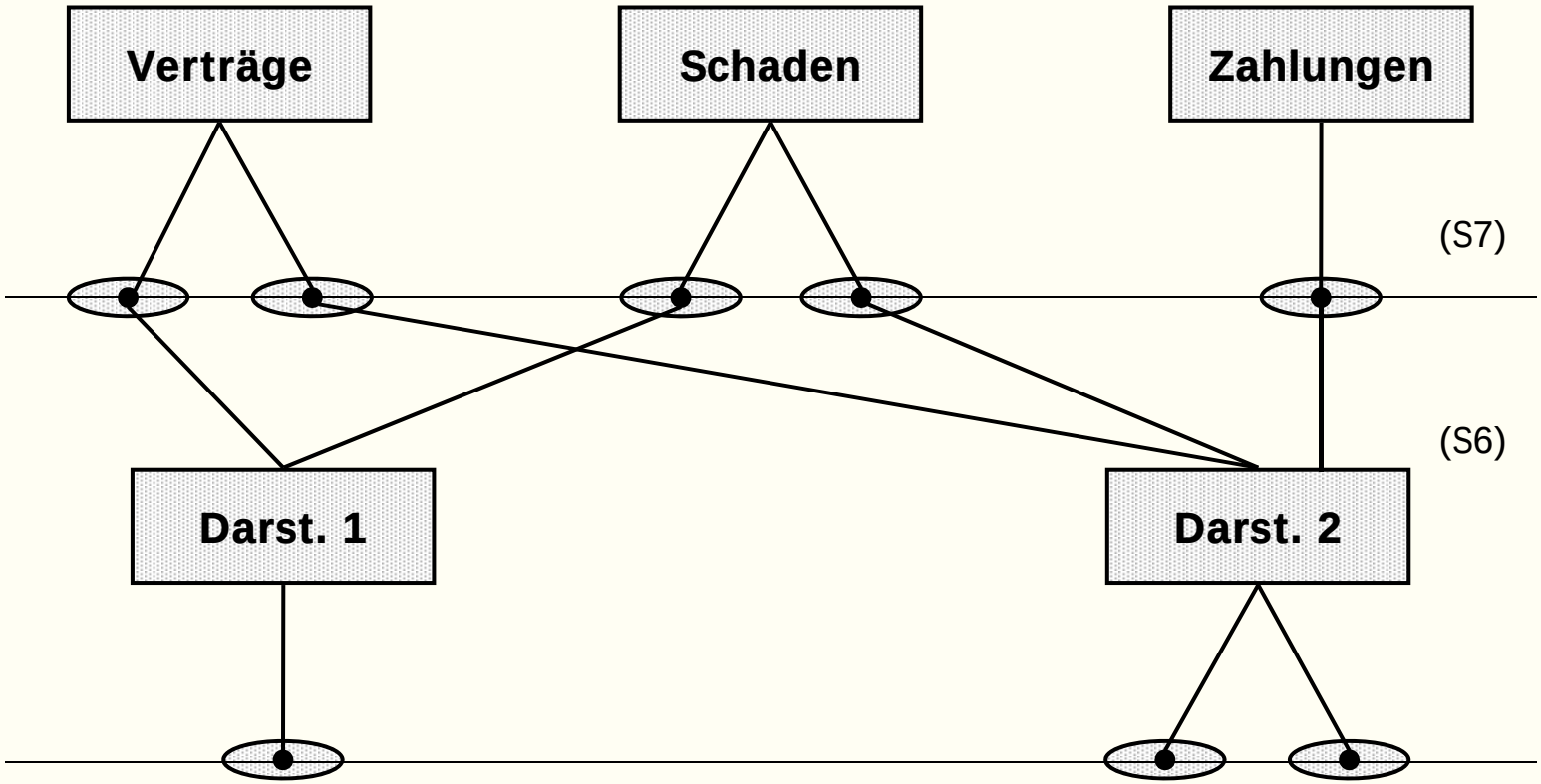


Adressierung von Instanzen (Forts.)



Verzweigungen **nach unten** innerhalb eines offenen Systems

Adressierung von Instanzen (Forts.)



Verzweigungen **nach oben und unten** innerhalb eines offenen Systems

D4.4 Architekturmodelle für Rechnernetze: Die *dynamische Struktur* eines Netzes

Zur Beschreibung von Diensten:

→ dynam. Inanspruchnahme und Bereitstellung von Diensten mittels

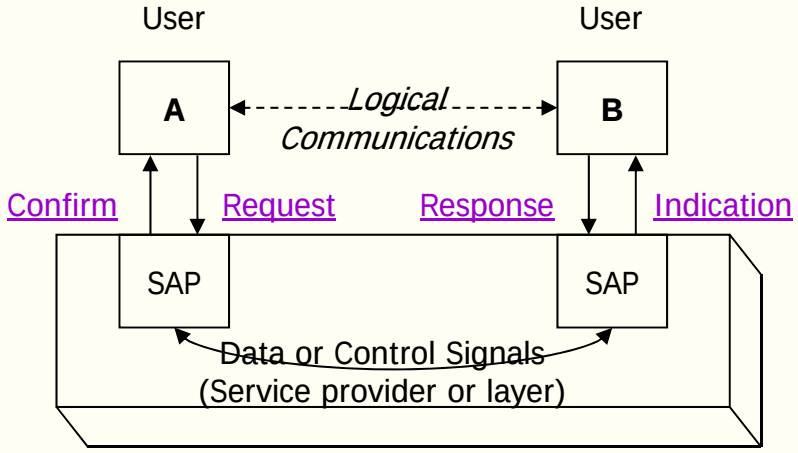
- **Request**
- **Response**

} „Benutzer“→ Diensterbringer

- **Confirm**
- **Indication**

} Diensterbringer → „Benutzer“

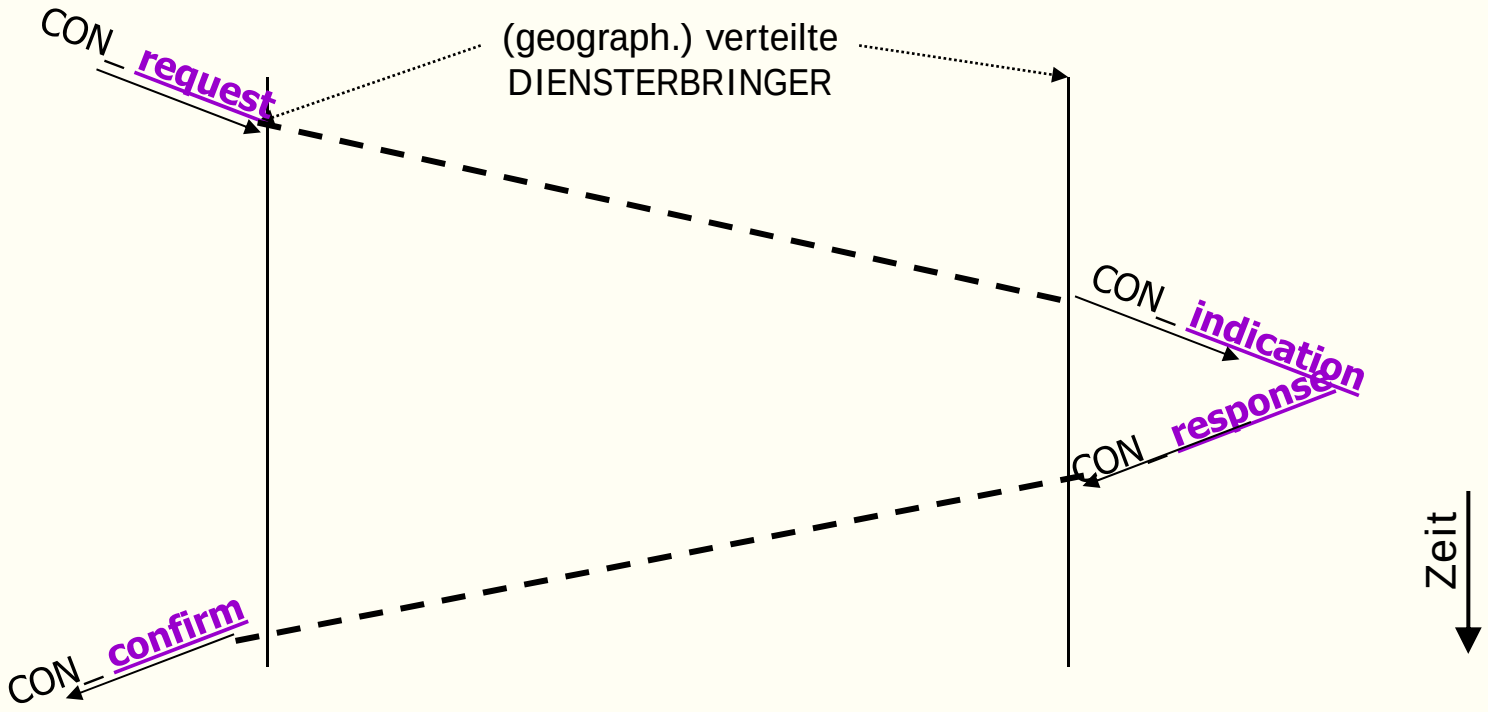
nota bene :
wohldefinierte Reihenfolge
(vgl. Beispiel s.u.)



SAP: **Service Access Point**

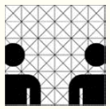
Beispiel bei Verbindungsaufbau:

„Two-way handshake“

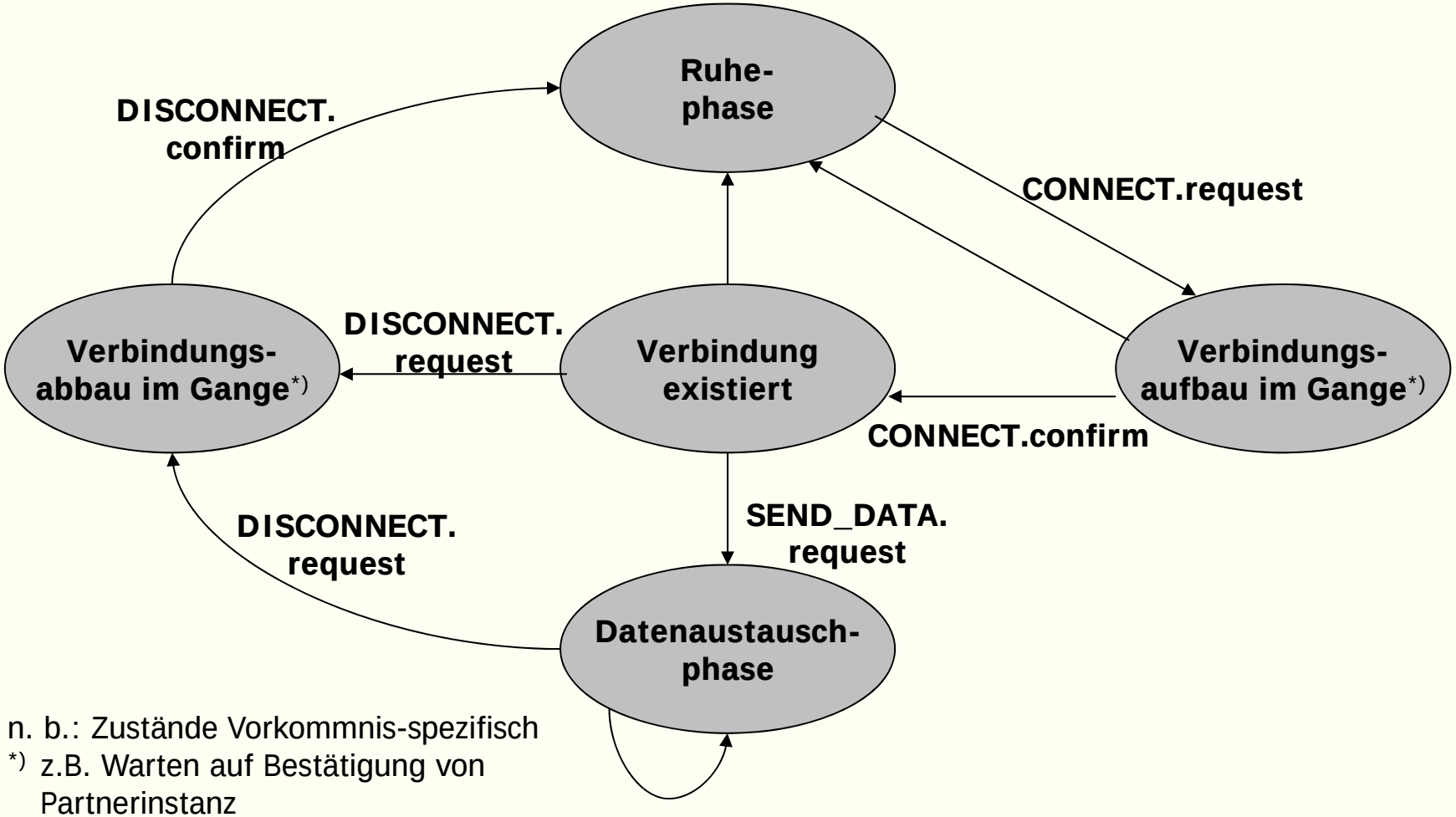


Bem.: Die (dynamisch) kommunizierenden Kommunikationspartner nennen wir „**Vorkommnisse**“ (→ Instanzen-Code spezifiziert Verhalten)

[vgl. Zusammenhang zwischen Programm (hier: Instanz) und Prozess (hier: Vorkommnis) !]



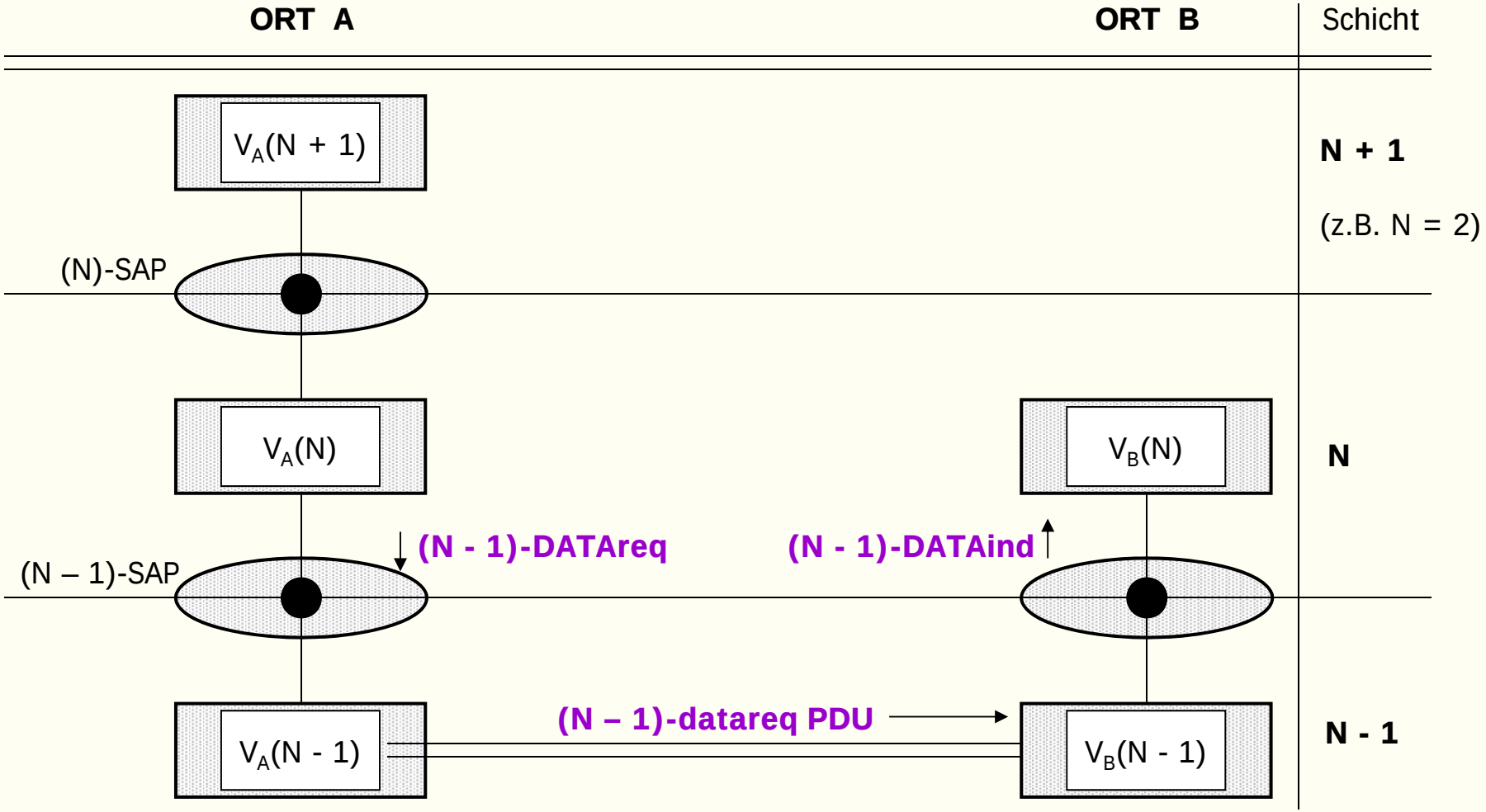
Zur formalisierten Beschreibung von Diensten
(hier: *verbindungsorientierter* Dienst)



n. b.: Zustände Vorkommnis-spezifisch

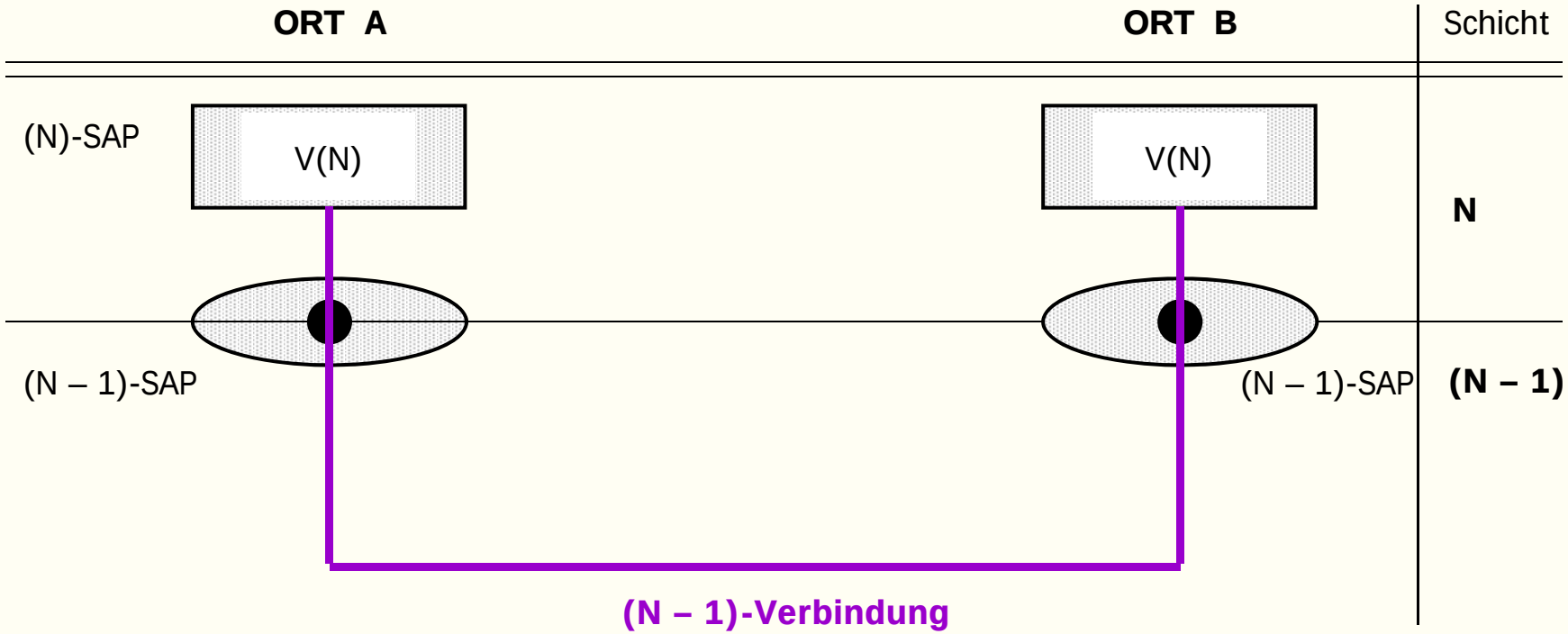
*) z.B. Warten auf Bestätigung von Partnerinstanz

Ziel: Aufbau einer (N)-Verbindung

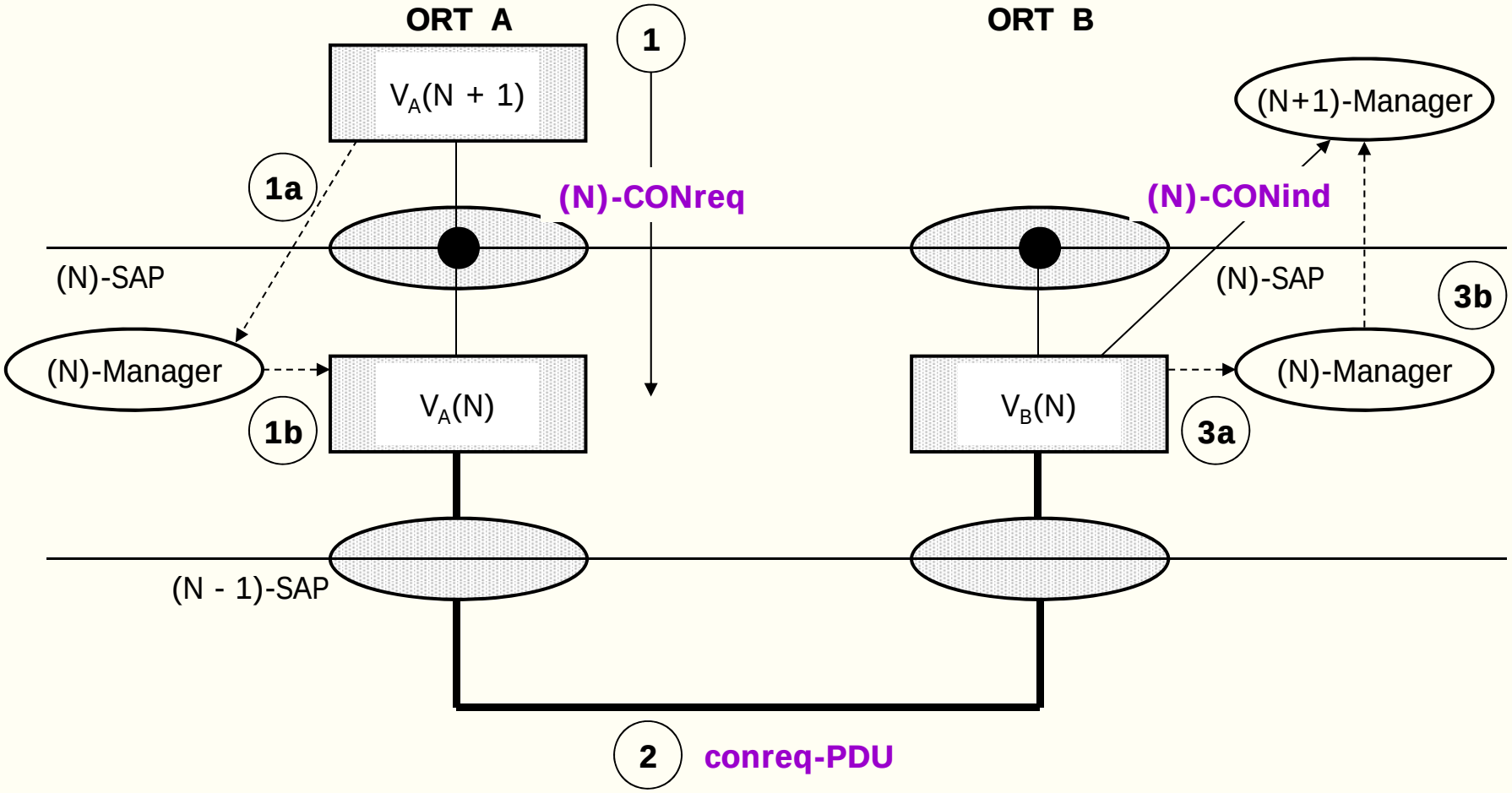


Ausgangssituation und Datenübertragung in (N-1)-Verbindung

Virtuelle (N-1)-Verbindung als Dienst gesehen

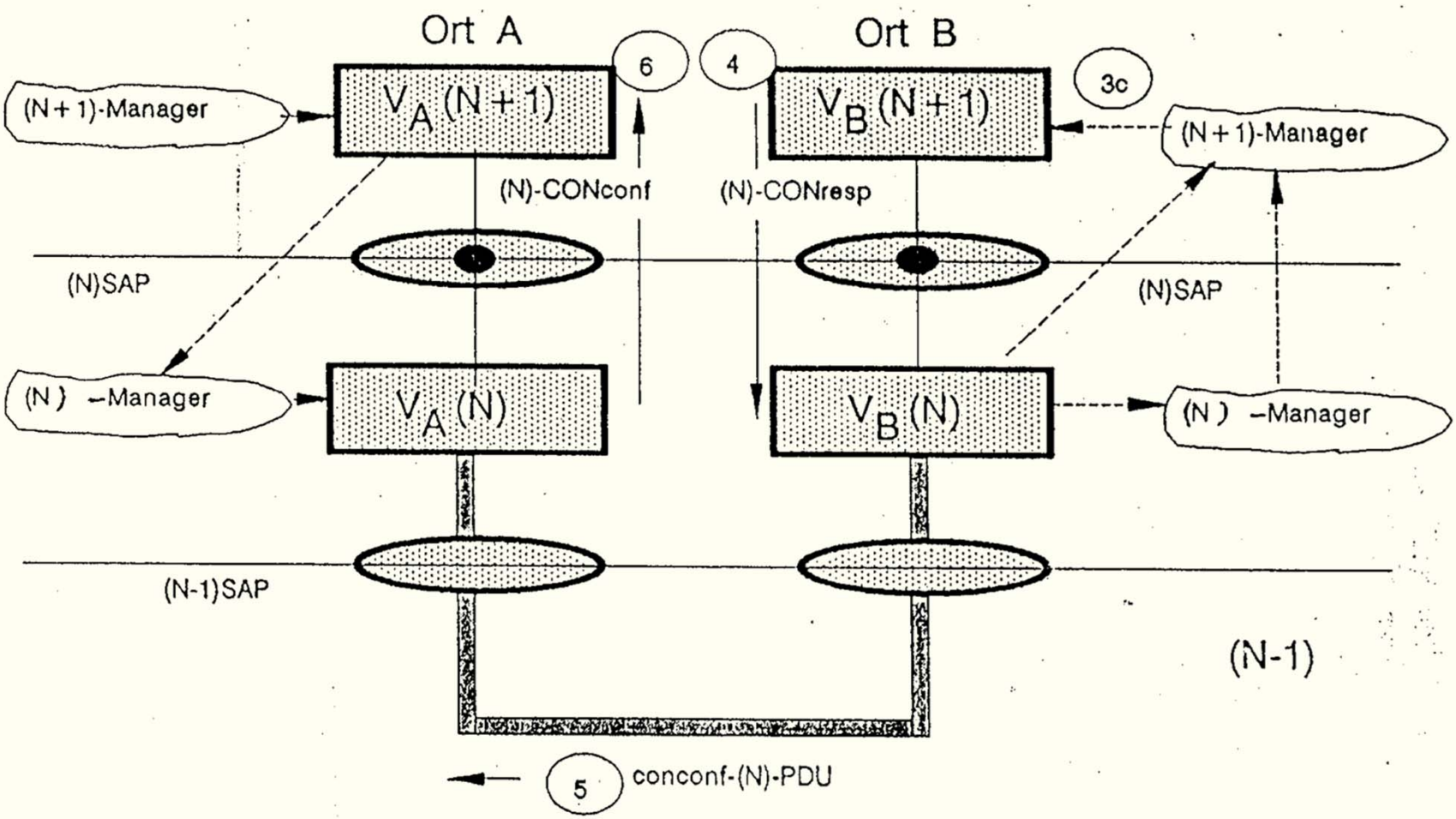


Phasen beim Aufbau einer (N)-Verbindung: TEIL 1



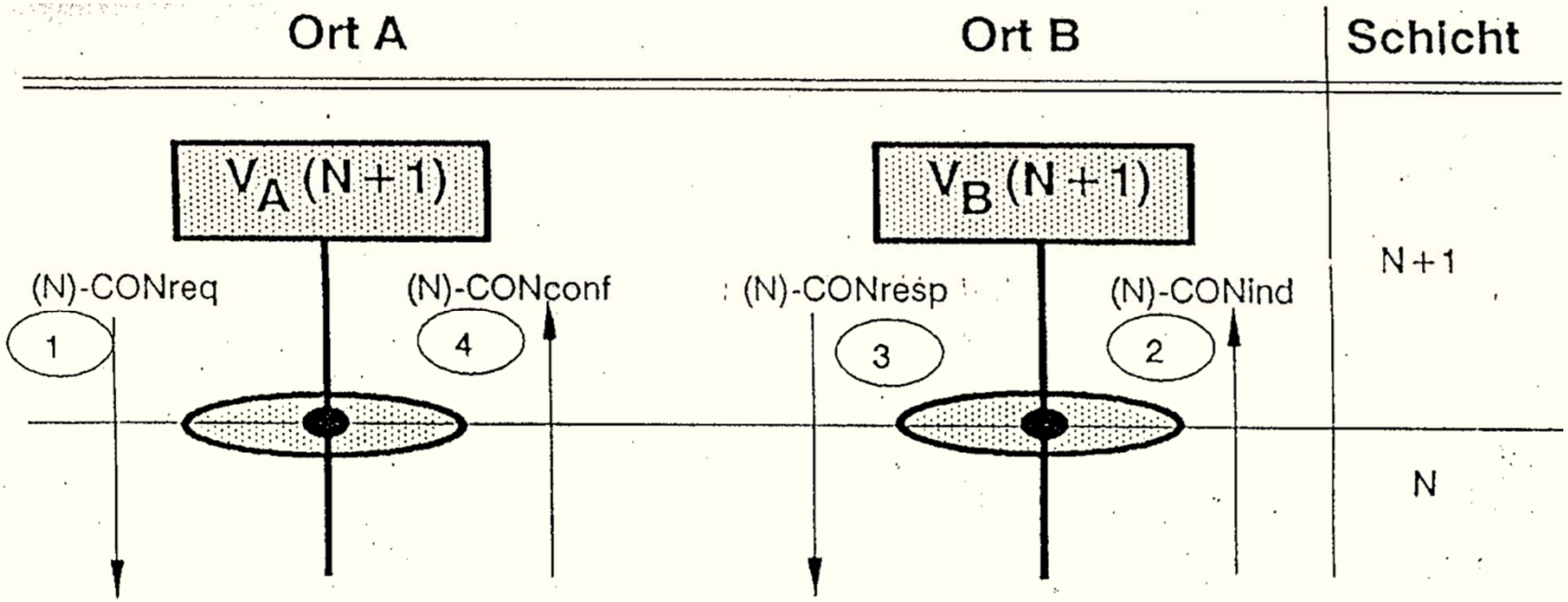
Aufbau einer (N)-Verbindung
Schritte **1**, **2**, **3**

Phasen beim Aufbau einer (N)-Verbindung: TEIL 2

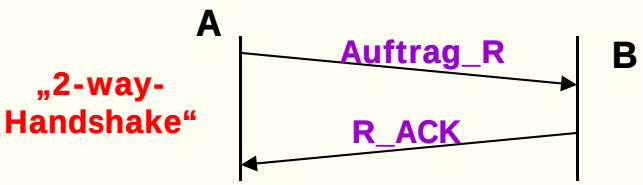


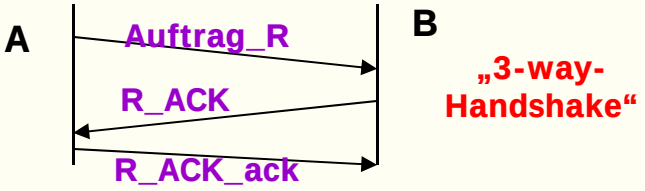
Aufbau einer (N)-Verbindung
Schritte **4**, **5**, **6**

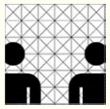
Resümee: „Handshake“ beim Aufbau einer Verbindung



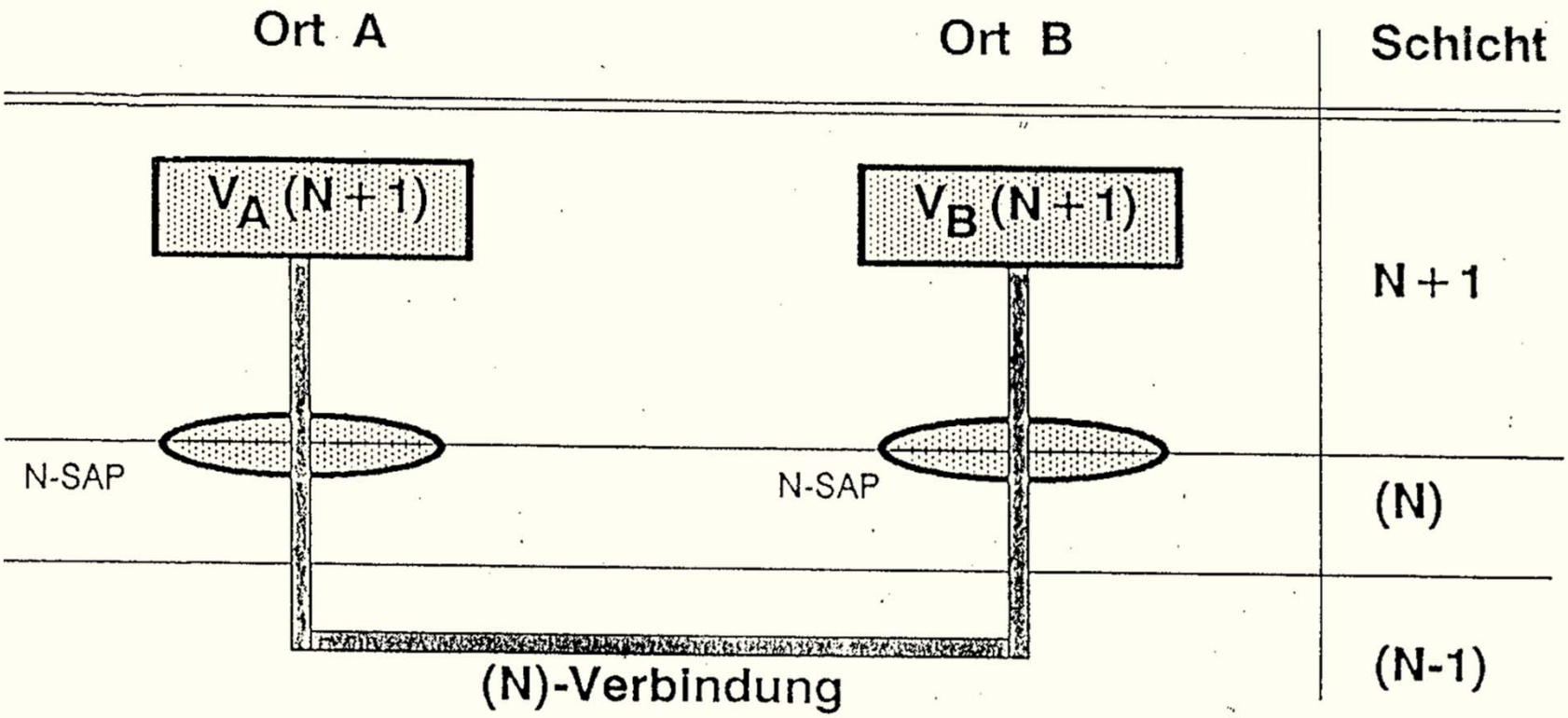
„Handshake“ (hier: „two-way handshake“)
→ Vorteil eines „three-way handshake“ ?







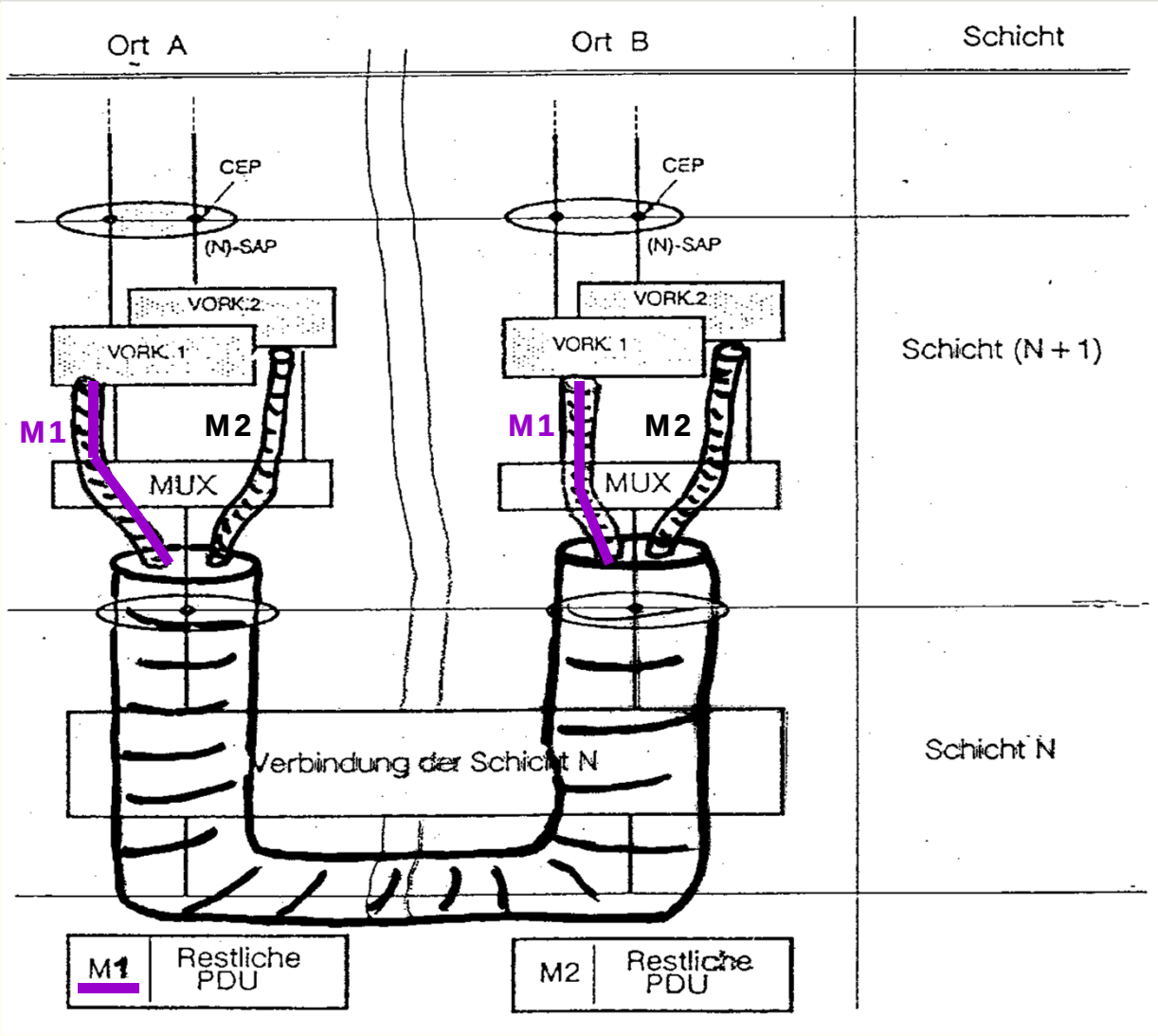
Ergebnis des Aufbaus einer (N)-Verbindung



Endresultat: virtuelle (N)-Verbindung

→ ergo: Ziel erreicht!

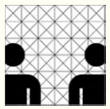
Multiplexen für Verbindungen



Transportsystem

Strukturiert in vier Abstraktionsebenen

- **Technologie-abhängige Schicht 1**:
  Bitübertragung:
  „Kabel" und Geräte-Anschluss; X.21; (IEEE 802.x) Medium Access Control (MAC): CSMA/CD, Token-Ring, WLAN, etc; „Repeaters" und „Bridges" (Schicht 2); auch: optische und drahtlose DÜ.
- **Geräteanschluss-abhängige Schicht 2**:
  Datensicherung:
  Fehlererkennung (CRC), -korrektur durch Wiederholung; Link Control (HDLC, LLC 1/2/3).
- **Netzwerk-abhängige Schicht 3**:
  Vermittlung:
  verbindungslose/-orientierte Vermittlung; Adressierung von Endsystem; (IP-) Router; (X.25-) PABX/Untervermittlung; Gateways zwischen unterschiedlichen Technologien.
- **Endsystem-abhängige Schicht 4**:
  Transport:
  verbindungsloser/-orientierter Transportdienst; unterschiedliche Transport-Protokolle; Anwendungs-Identifikation; Gateways zwischen unterschiedlichen Transport-Protokoll- „Säulen“.



Grobzerlegung von Kommunikationssystemen

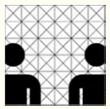
Ein Kommunikationssystem wird zerlegt in:

- ***Anwendungssystem***
  (OSI-Schichten 5 – 7 *bzw.* Internet-Protokollstack oberhalb TCP/UDP)
- ***Transportsystem***
  (OSI-Schichten 1 – 4 *bzw.* Internet-Protokollstack bis einschließlich TCP/UDP)

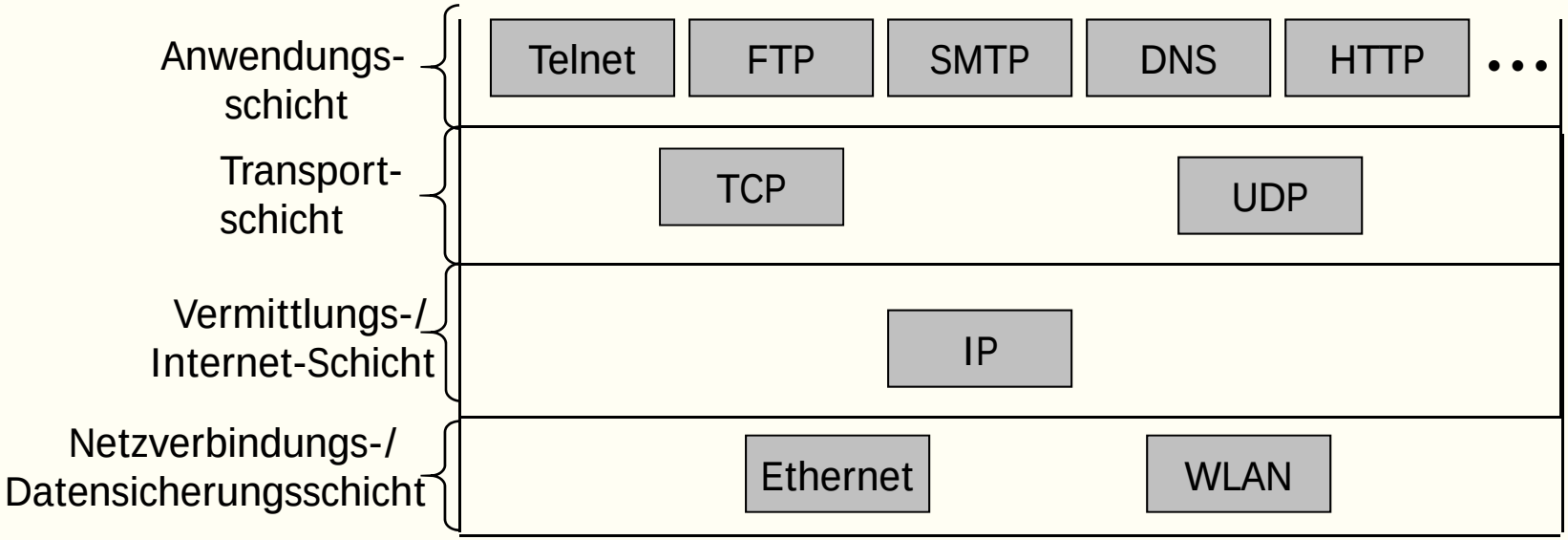
Jeweils eine Gruppe aufeinander aufbauender

- Transport-,
- Vermittlungs-,
- Datensicherungs- und
- Bitübertragungs-Protokolle

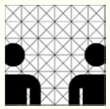
heißt **OSI-Transport-“Säule“** und stellt eine der möglichen Realisierungen des OSI-Transportsystems dar.



D4.5 Das Architekturmodell des Internet



- ***IP*** (Internet Protocol): datagrammorientierte Vermittlung
- ***TCP*** (Transmission Control Protocol): verbindungsorientiertes Transportprotokoll
- ***UDP*** (User Datagram Protocol): verbindungsloses Transportprotokoll
- ***Telnet*** : Virtueller Terminaldienst, vgl. Remote Login
- ***FTP*** (File Transfer Protocol): Dateitransfer zwischen verschiedenen Endsystemen
- ***SMTP*** (Simple Mail Transfer Protocol): Electronic Mail (elektronische Briefpost)
- ***DNS*** (Domain Name System): Konvertieren von Hostnamen in Netzadressen
- ***HTTP*** (HyperText Transfer Protocol): Holen von Seiten aus dem World Wide Web (WWW)



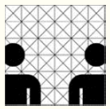
Internet- vs. ISO/OSI-Kommunik.architektur

Vorteile des ISO/OSI-Architekturmodells :

- Entwicklung von (relativ) wohldefinierten Basiskonzepten für Rechnernetze, u.a. Dienst, Protokoll, ...
- klare Schichtenstruktur für Protokollhierarchie (später etwas „aufgeweicht", u.a. „Sublayers")
- Architektur stärker unabhängig von konkreten Protokollen, relativ allgemeines Modell
- für viele Wissenschaftler für lange Zeit: „die reinere Lehre“!

Vorteile des Internet-Architekturmodells :

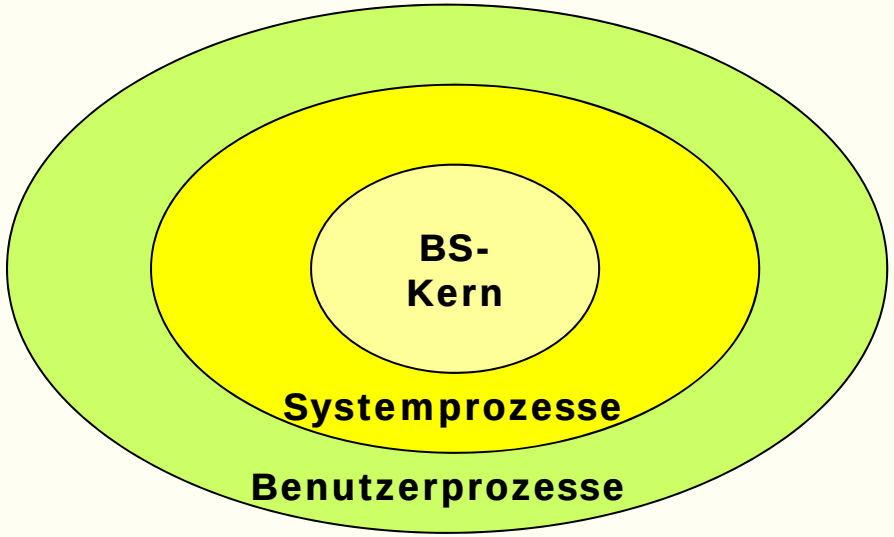
- relativ früh effiziente, allgemein zugängliche Implementierungen verfügbar → weltweite Akzeptanz
- Standardisierung von Protokollen erst nach deren erfolgreichem Einsatz
- kleinere Anzahl von Protokollschichten → wiederum effizienter!
- keine Scheu vor pragmatischen Lösungen
- Motto : „(relatively) simple is beautiful“
- Interkonnektion von Netzen dank IP-Protokoll relativ einfach wegen verbindungsloser Kommunikation bei IP (vgl. techn. Details von IP)
- „Killer Applications" für das Internet gefunden und relativ frühzeitig bereitgestellt, u.a. E-mail, WWW,...



Betriebssystem (BS) – *versus* **Rechnernetz** (RN)-**Diensthierarchie**

Unterschiede:

- Diensthierarchien in BS weniger streng (z.B. hinsichtlich Dienstbenutzer/ Diensterbringer-Relation)
- bei BS Dienste im allg. *zentralisiert*, bei Netzen i.d.R *verteilt*, erbracht
- bei BS auch von („Zwiebel-“) Schalenstruktur die Rede:



- **Hauptgründe für Diensthierarchien in BS** :
  - ***Schutz-/Sicherheitsmechanismen*** an Dienstschnittstellen integrierbar (z.B. bei Dienstaufrufen)
  - Wunsch nach ***Komplexitätsreduktion***

→ Standardisierungsargument weniger bedeutend als bei Netzen