

Internetanwendungstechnik

TCP/IP- und OSI-Referenzmodell

Gero Mühl

Technische Universität Berlin

Fakultät IV – Elektrotechnik und Informatik

Kommunikations- und Betriebssysteme (KBS)

Einsteinufer 17, Sekr. EN6, 10587 Berlin

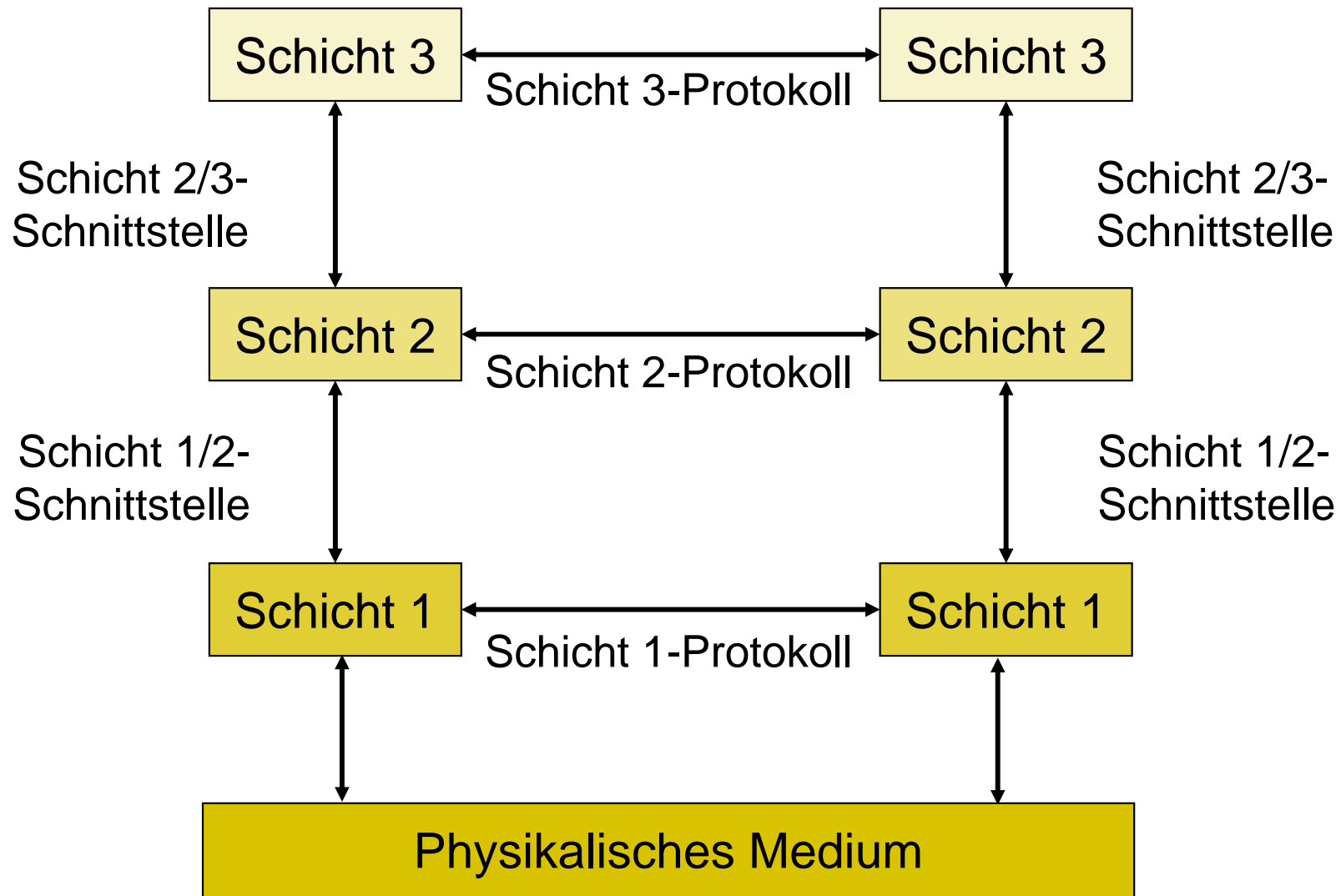
Schichtenmodelle

- > In hierarchischen Architekturen lassen sich Hierarchieebenen mit unterschiedlichen Funktionen identifizieren → **Schichten**

- > Jede Schicht
 - > hat bestimmten Zweck und Funktion
 - > bietet den jeweils höheren Schichten **Dienste** an
 - > besitzt **Schnittstelle** zum Zugriff auf angebotene Dienste
 - > abstrahiert von Details der Implementierung
 - > nutzt Dienste tieferer Schichten zur Funktionserbringung

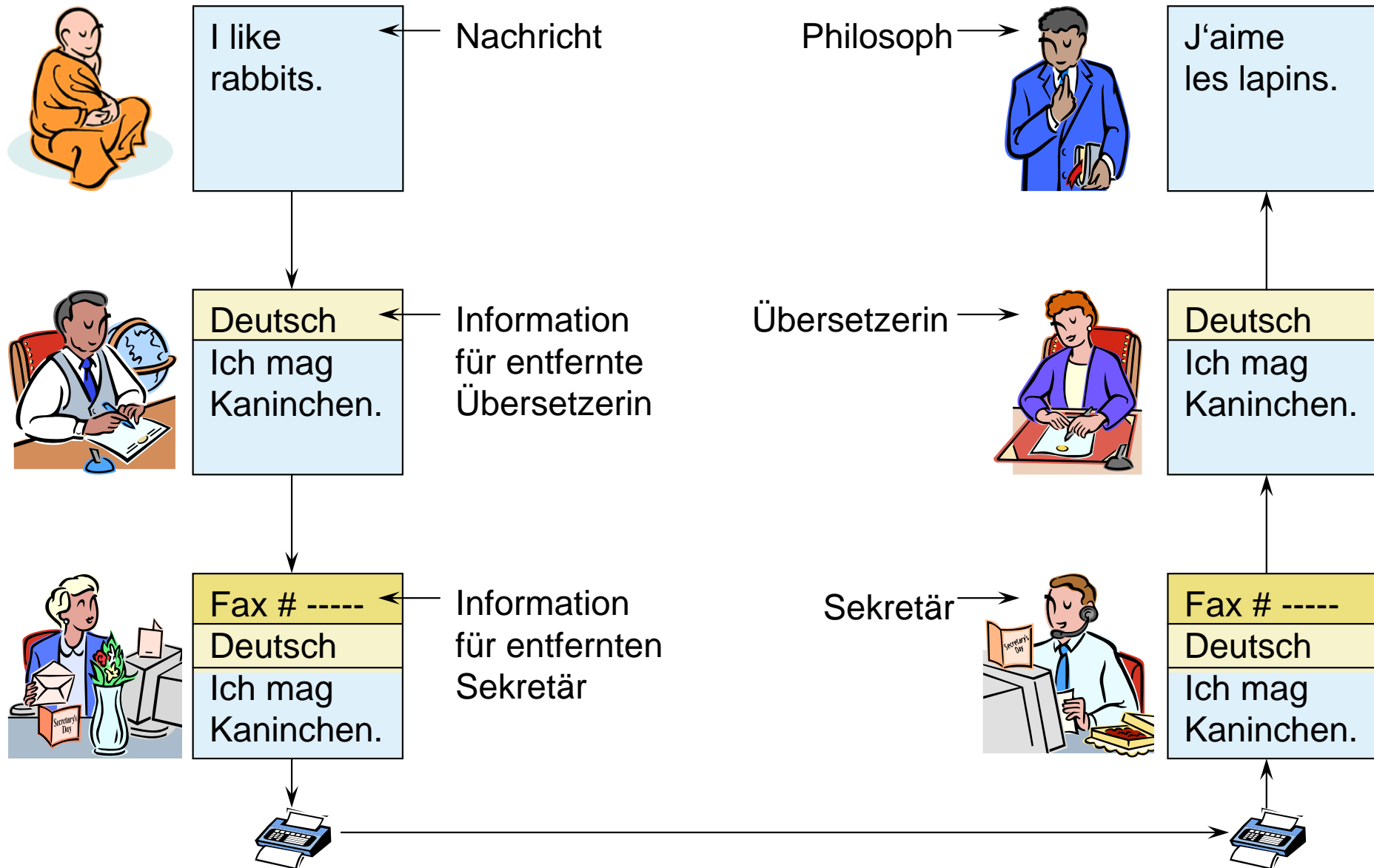
- > Ziel: Reduzierung der Komplexität

Protokollhierarchien



Architekturbeispiel

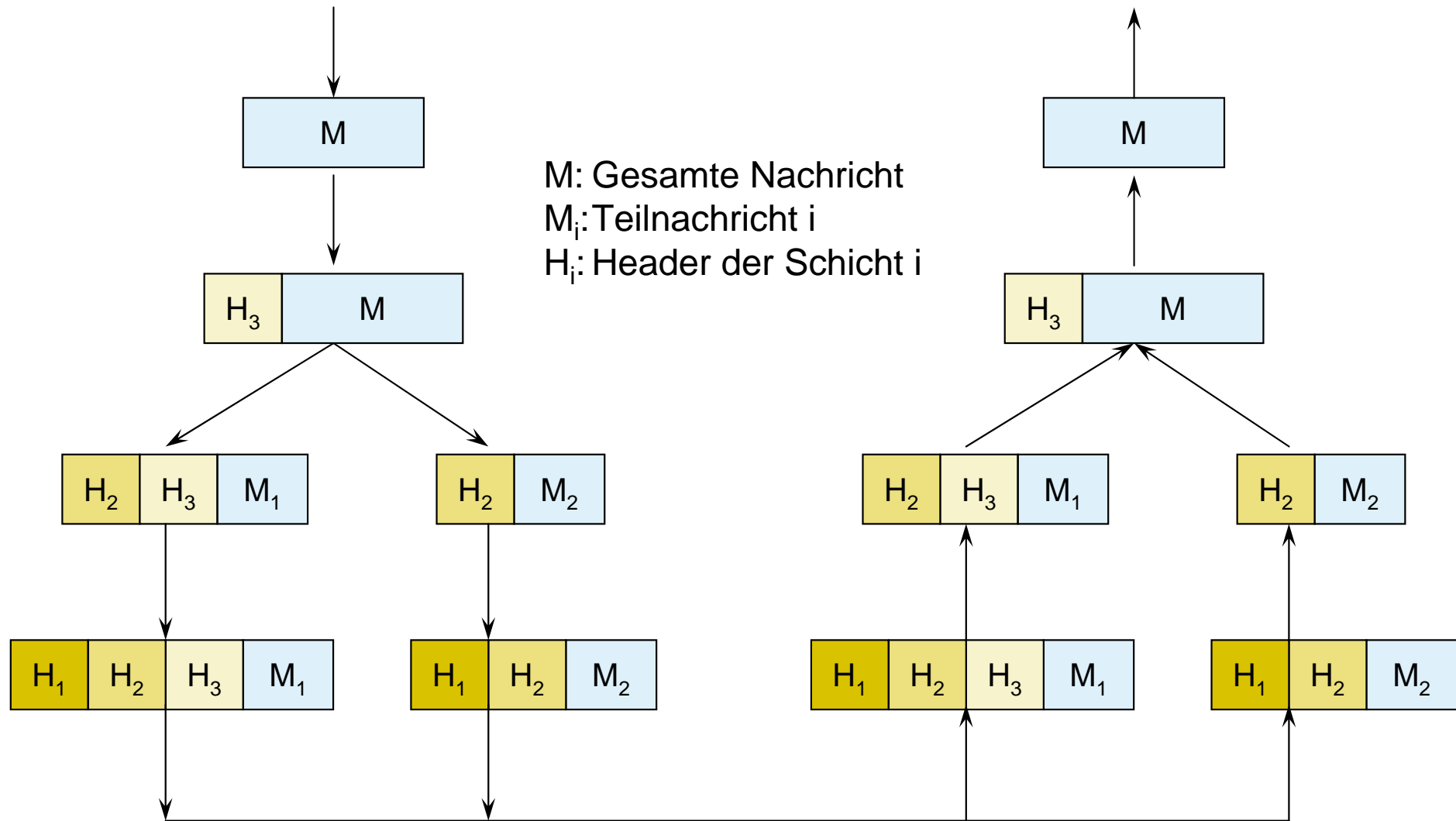
Philosoph/ÜbersetzerIn/
SekretärIn-Architektur



Protokoll

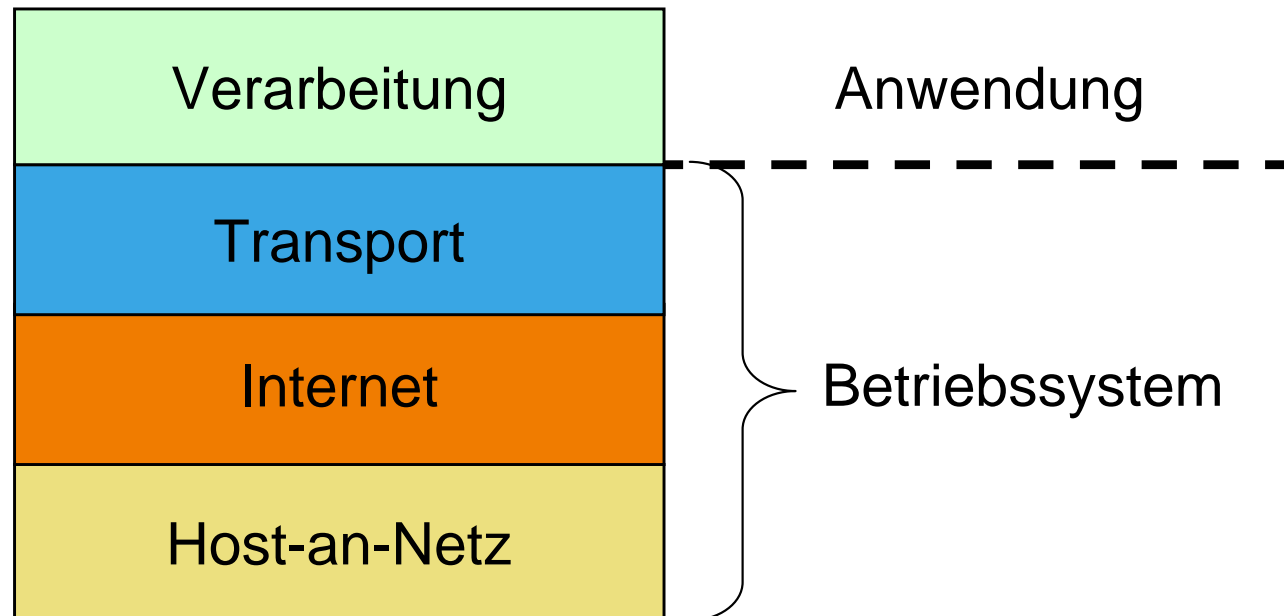
- > Ein **Protokoll** legt die Regeln und Konventionen fest, wie die Schicht n der Maschine A mit Schicht n der Maschine B kommuniziert
- > Beim Senden leitet jede Schicht die Daten an die unterliegende Schicht weiter, bis die physikalische Schicht erreicht ist
- > In jeder Schicht können die Daten transformiert (z.B. aufgeteilt) sowie zusätzliche Informationen (z.B. Header, Trailer) hinzugefügt werden
- > Der Empfang erfolgt entsprechend umgekehrt
- > Eine Protokollhierarchie nennt man auch **Protocol Stack**

Informationsfluss



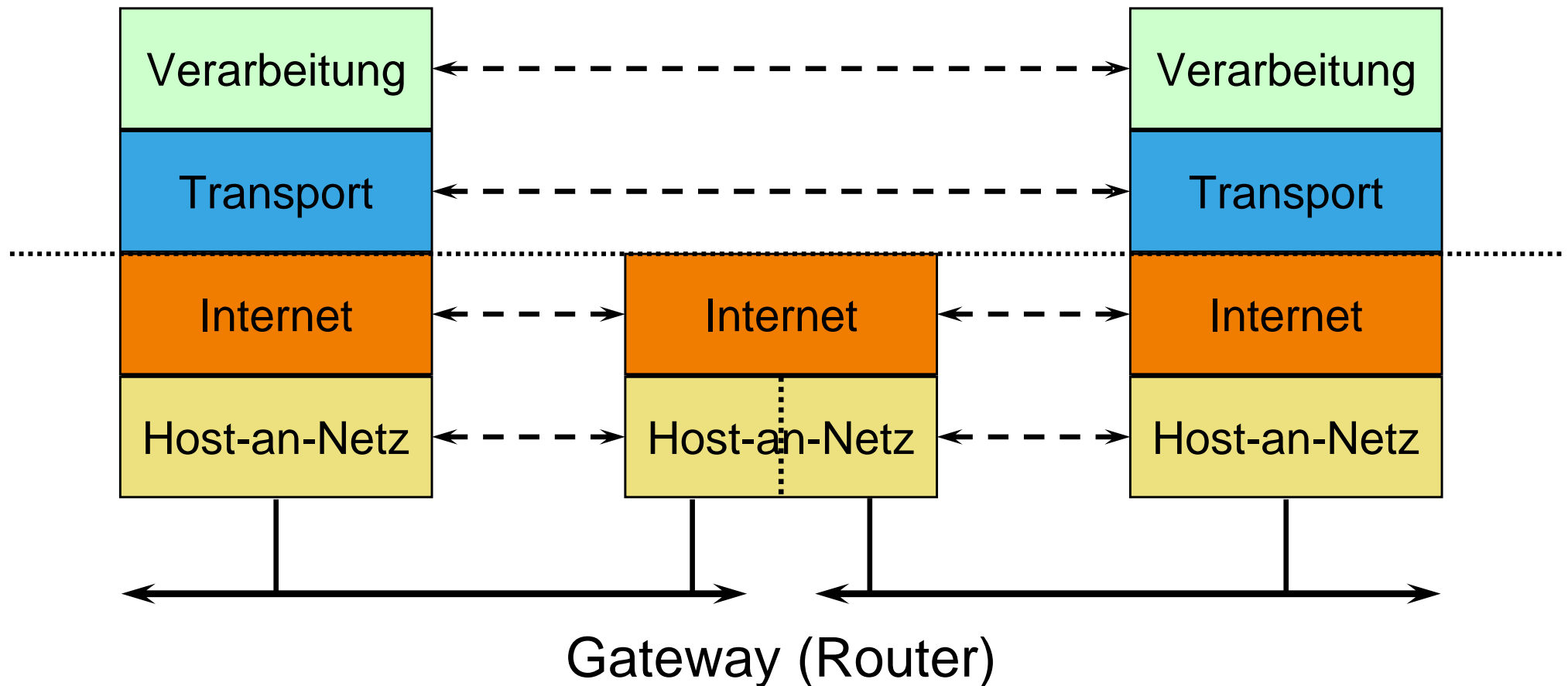
TCP/IP-Referenzmodell

- > Schichtenmodell mit vier Schichten



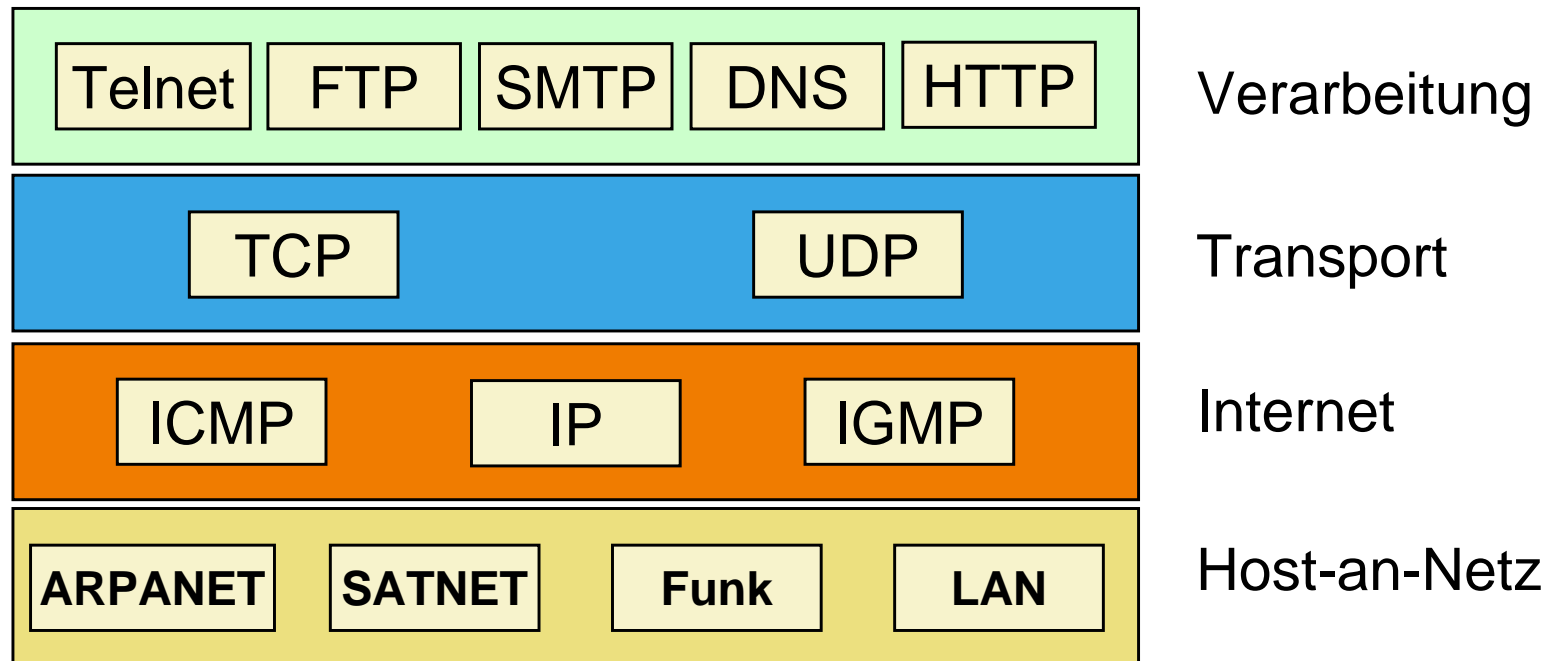
TCP/IP-Referenzmodell

- > Endsysteme und Router



TCP/IP-Referenzmodell

> Protokolle (Beispiele)

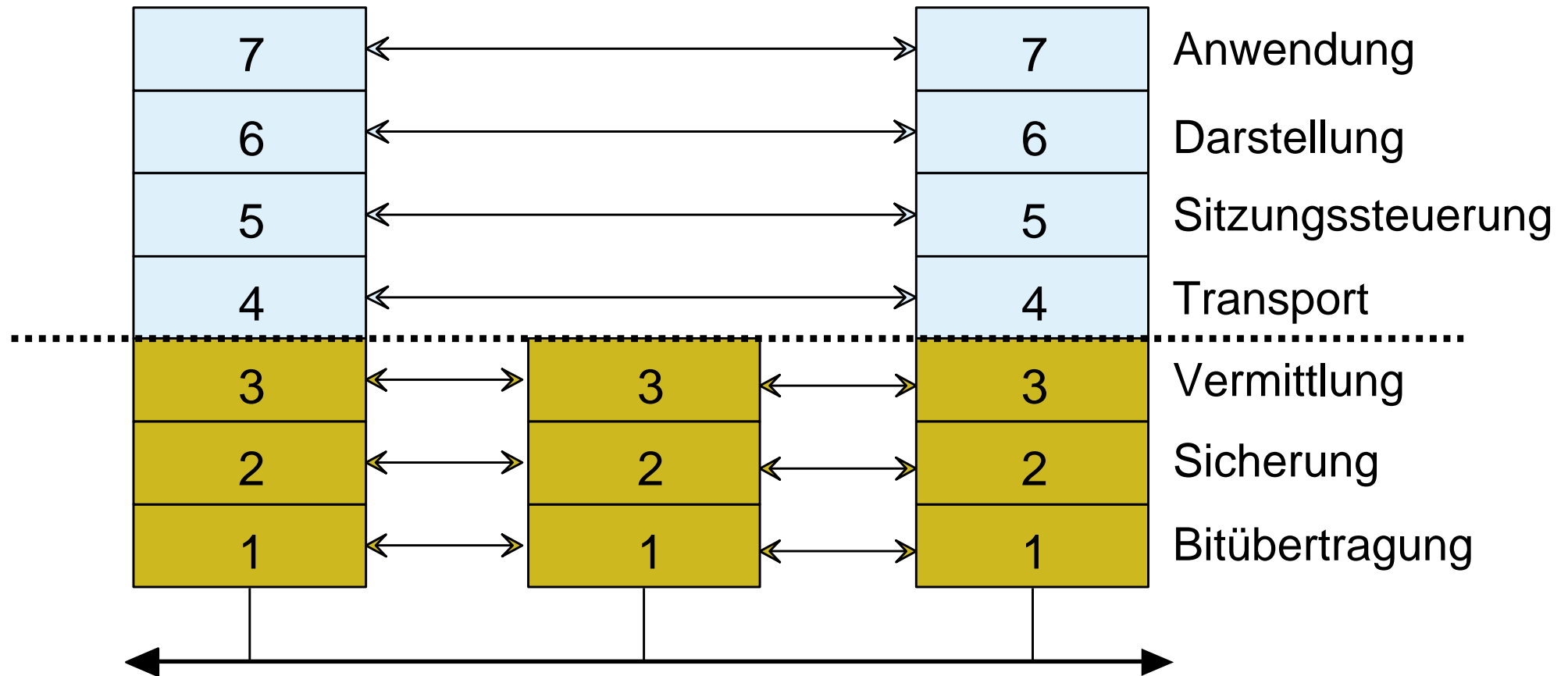


Die Schichten des TCP/IP-Referenzmodells

- > **Host-an-Netz**
 - > Senden, weiterleiten und empfangen einzelner Pakete zwischen direkt miteinander kommunizierenden Rechnern
- > **Internet**
 - > Ende-zu-Ende senden und empfangen einzelner Pakete zwischen miteinander kommunizierenden Rechnern
- > **Transport**
 - > Ermöglicht die Kommunikation zwischen Anwendungen
 - > Zwei bekannte Protokolle
 - > TCP ermöglicht zuverlässige, verbindungsorientierte Kommunikation auf Basis von Byteströmen
 - > UDP ermöglicht unzuverlässige, verbindungslose Kommunikation auf Basis von Nutzdatenpaketen
- > **Verarbeitung**
 - > Stellt Applikationsprotokolle wie HTTP, FTP oder SMTP zur Verfügung

OSI-Referenzmodell

- > OSI = Open Systems Interconnection
- > Schichtenmodell mit sieben Schichten



Die Schichten des OSI-Referenzmodells

- > **Bitübertragung** (engl.: *Physical*)
 - > Übertragung ungesicherter Bitströme über den Kommunikationskanal
- > **Sicherung** (engl.: *Data Link*)
 - > Sicherung der zu übertragenden Daten durch Aufteilung in Rahmen, Checksummen und Bestätigungen; Flusskontrolle
- > **Vermittlung** (engl.: *Network*)
 - > Annehmen und Zustellen einzelner Pakete
- > **Transport** (engl.: *Transport*)
 - > Entgegennehmen der Daten
 - > Aufteilung in Pakete für die Vermittlungsschicht
 - > Abliefern der vollständigen und korrekten Daten

Die Schichten des OSI-Referenzmodells

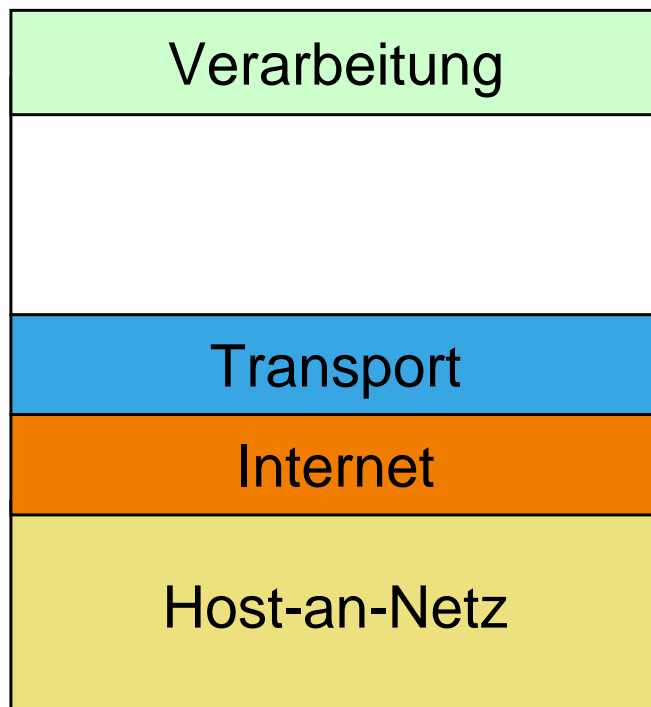
- > **Sitzungssteuerung** (engl.: *Session*)
 - > Synchronisation der Kommunikationspartner
 - > Checkpointing bei langen Übertragungen

- > **Darstellung** (engl.: *Presentation*)
 - > Legt die Syntax und die Semantik der zu übertragenden Daten fest (Datenrepräsentation)

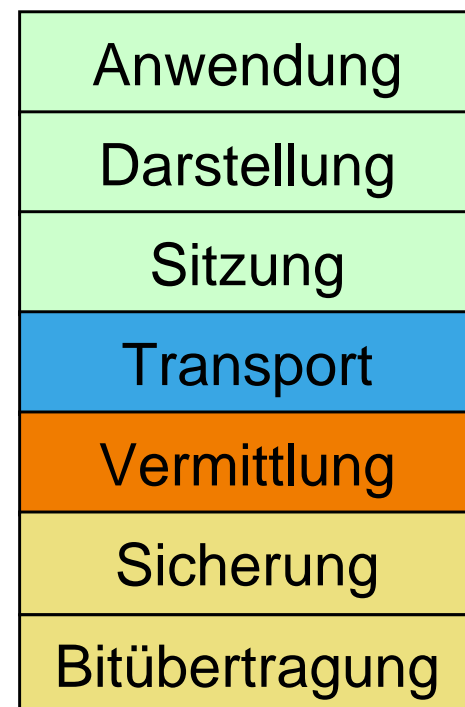
- > **Anwendung** (engl.: *Application*)
 - > Stellt Applikationsprotokolle wie HTTP, FTP oder SMTP zur Verfügung

Internet- vs. OSI-Referenzmodell

TCP/IP



OSI



Vergleich der Referenzmodelle

TCP/IP

- ☺ „Bottom-up“-Entwicklung getrieben aus der Praxis
- ☺ Einfachheit statt Universalität
- ☺ Nur verbindungsloses Vermittlungsprotokoll
- ☹ Schwächen im Modell und in der Ebenenstruktur
- ☹ Keine klare Trennung zwischen "Dienst", "Schnittstelle", "Protokoll"

Modell	☹
Protokolle	☺

OSI

- ☹ Zähe „top-down“-Entwicklung mit wenig Bezug zur Praxis
- ☺ Klar strukturiert und konzeptionell umfassend
- ☹ Ineffiziente Verarbeitung
- ☹ Aufwendige, komplexe Implementierung
- ☹ Redundante Funktionen
- ☹ Schlechtes Timing der Stand.

Modell	☺
Protokolle	☹

Fragen?

