EDU/EVS

Entwurf Verteilter Systeme

Prof. Dr.-Ing. Martin Gaedke

Technische Universität Chemnitz

Fakultät für Informatik

Professur Verteilte und selbstorganisierende Rechnersysteme

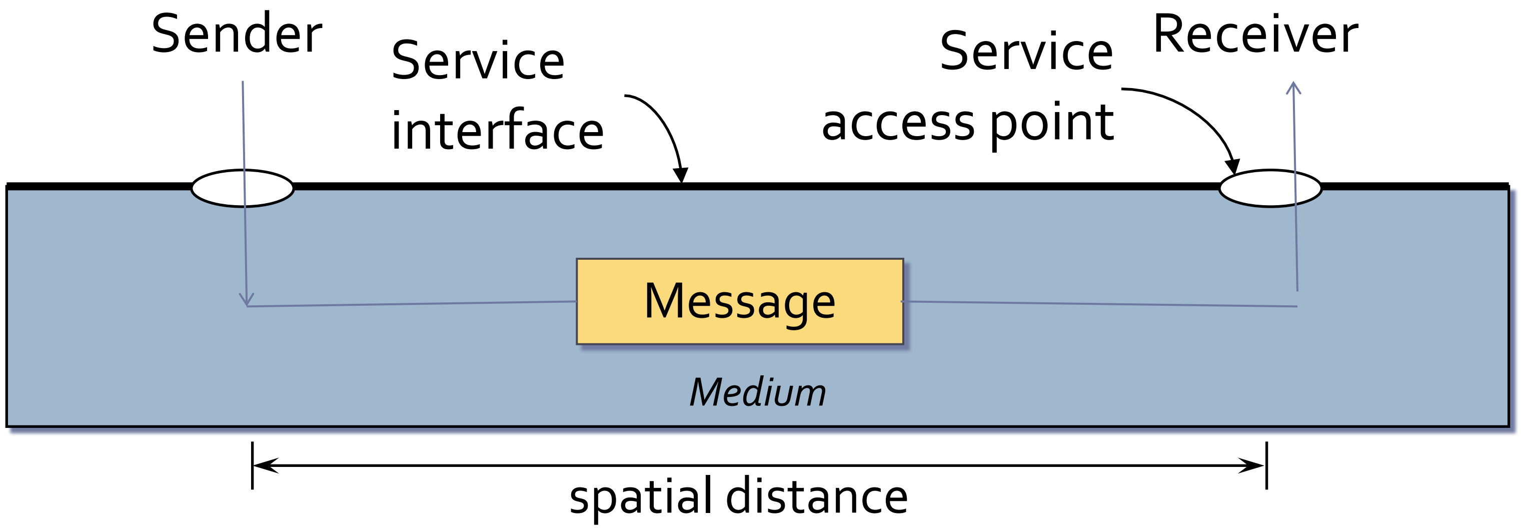
http://vsr.informatik.tu-chemnitz.de

WS 2017/18

Chapter 3

BASIC COMMUNICATION MODELS

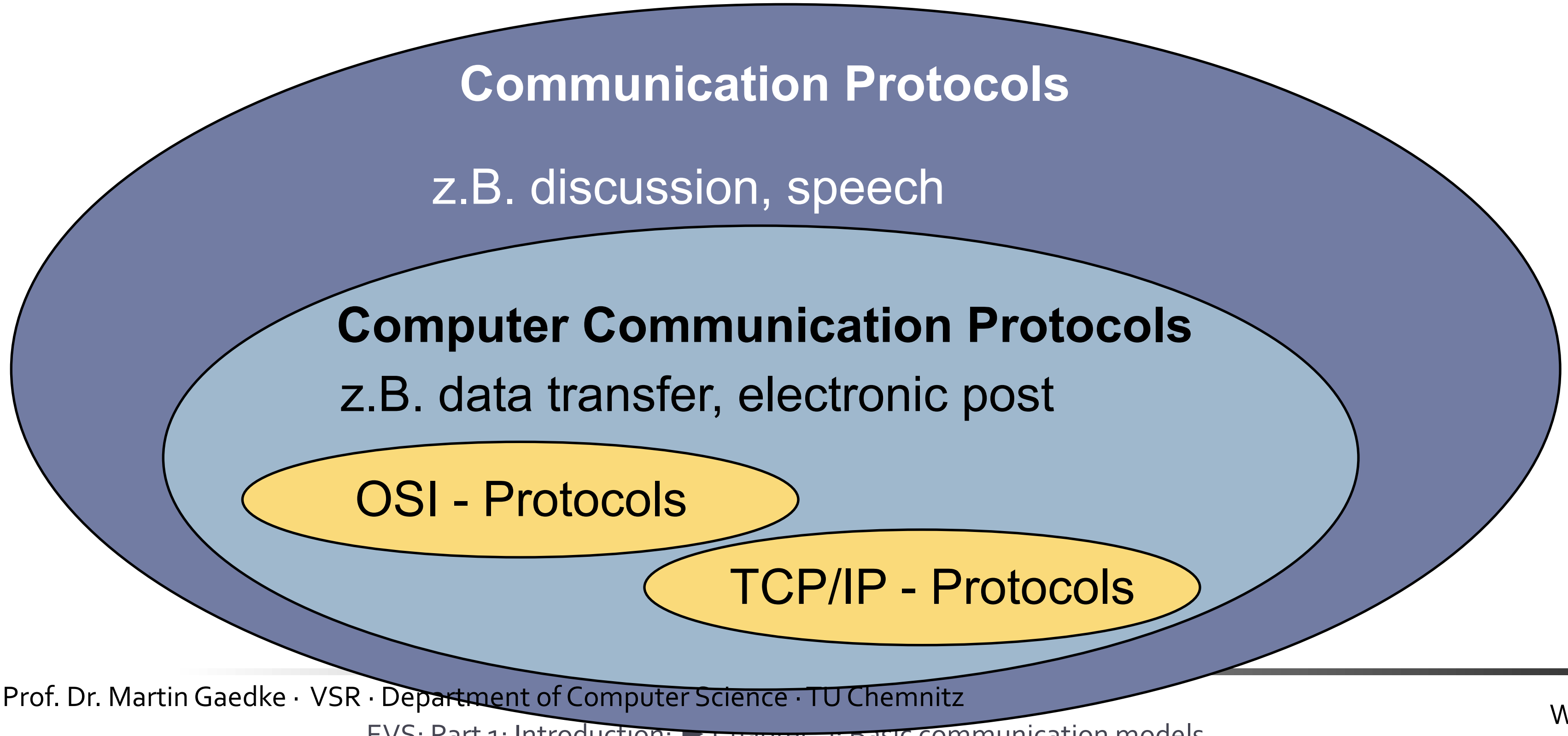
Basic Telecommunication Model



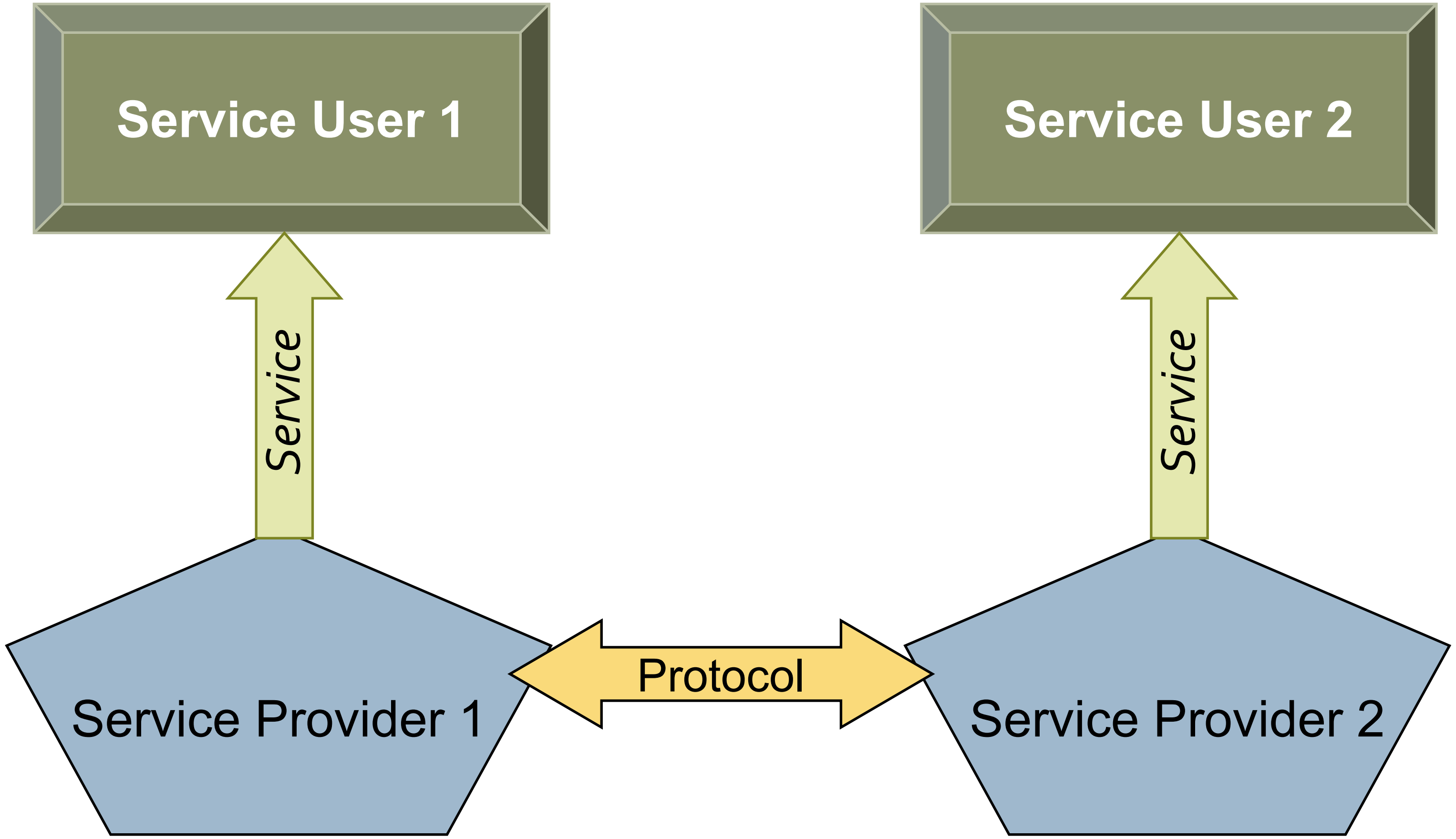
- Participants act as transmitters or receivers.
- Service use by participants is carried out on a special service interface using a service access point.
- Medium bridges over the spatial distance.

Protocol Characterization

- **Communication protocol** describes a set of rules according to which communication between two or more parties must be carried out.

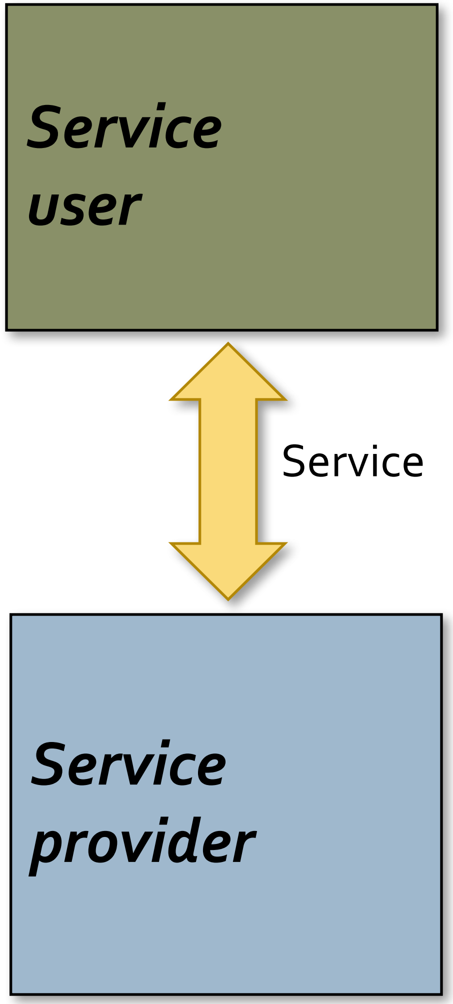


Service and Protocol

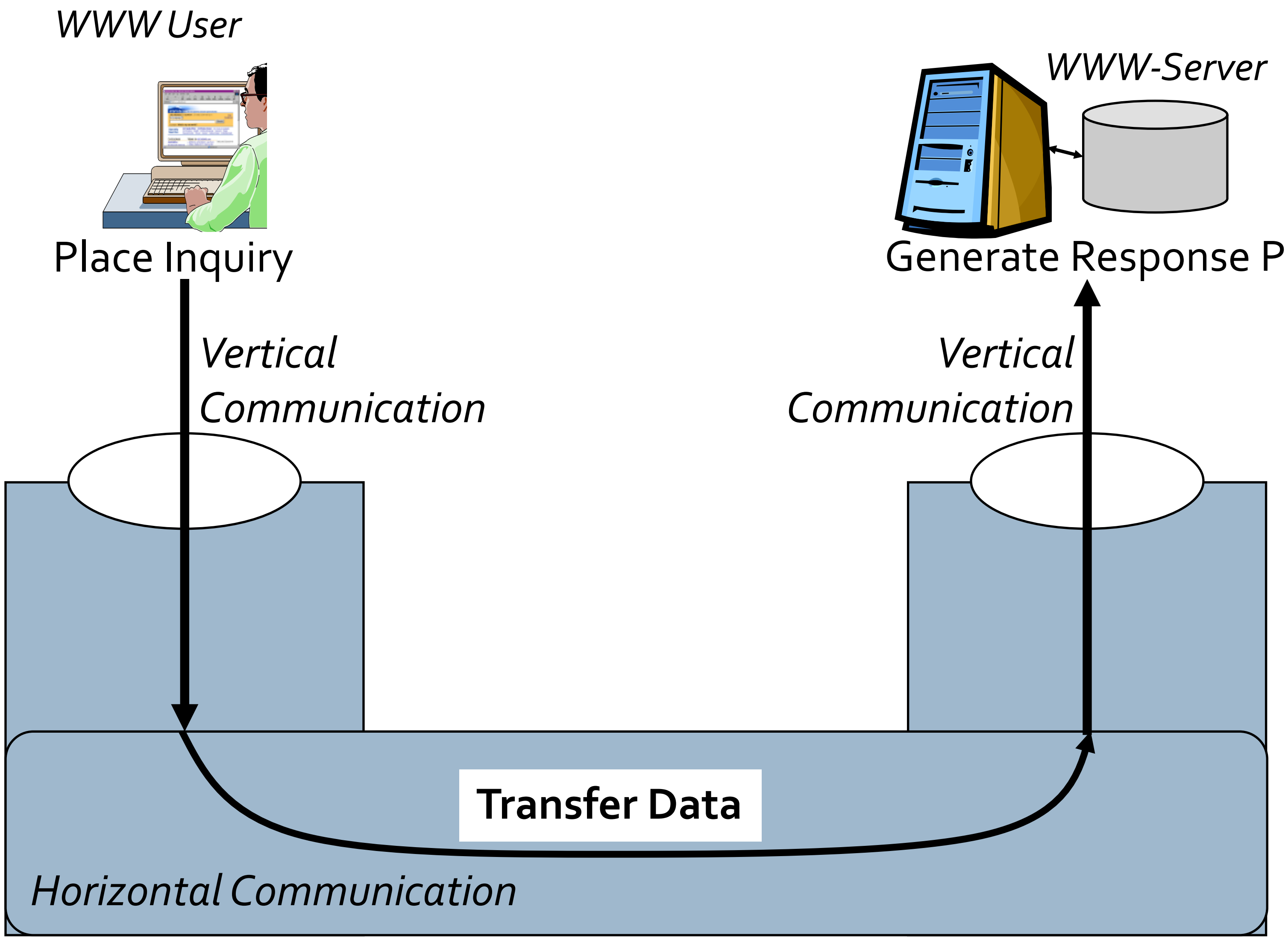


Service Characterization

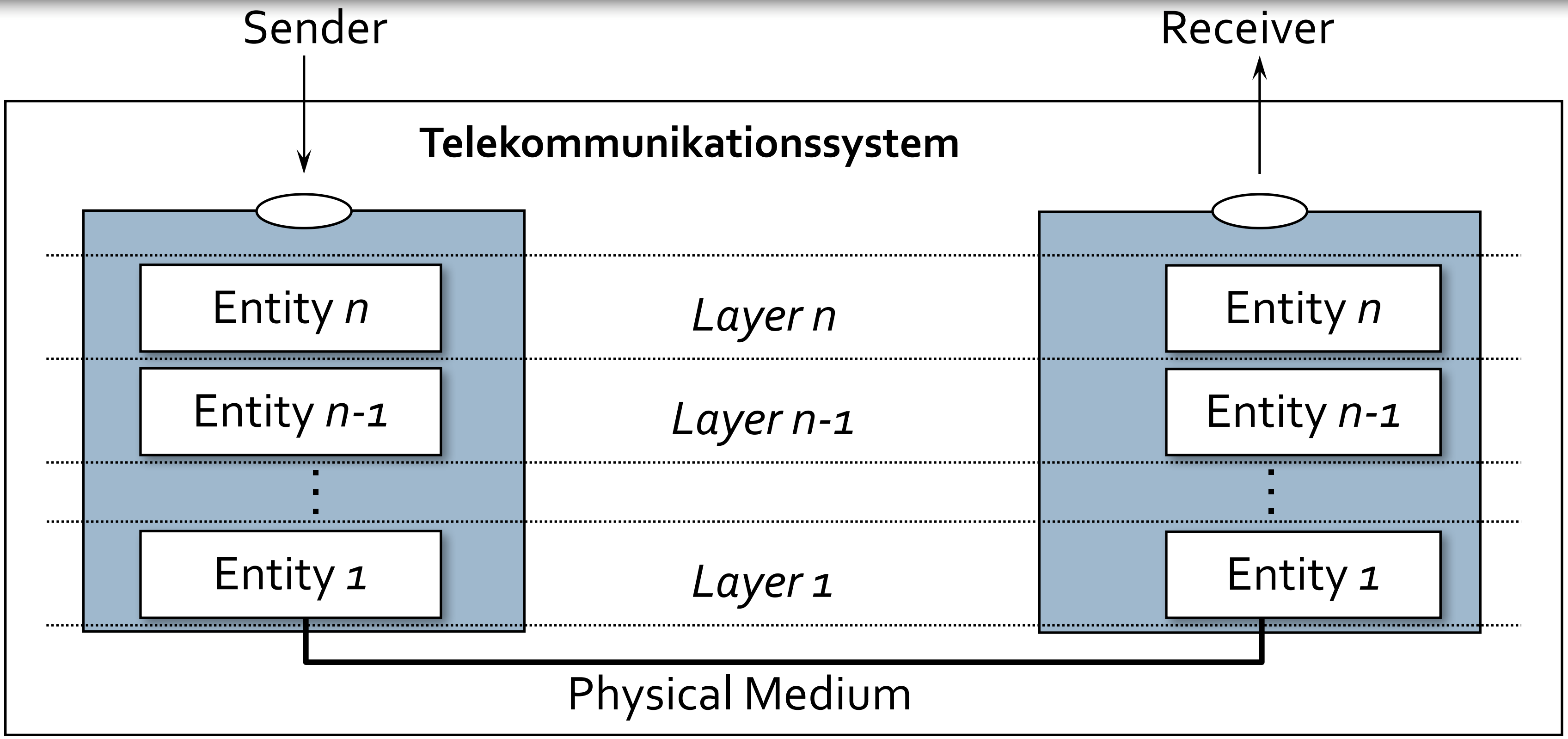
- *Service user* claims a service.
- Service is requested from a service provider.
- Service is provided in the context of service provision, which in turn has different properties (e.g. confirmed, unconfirmed).
- Service provision thus includes the fulfillment of an order, which is specified as part of the service.



Horizontal and Vertical Communication

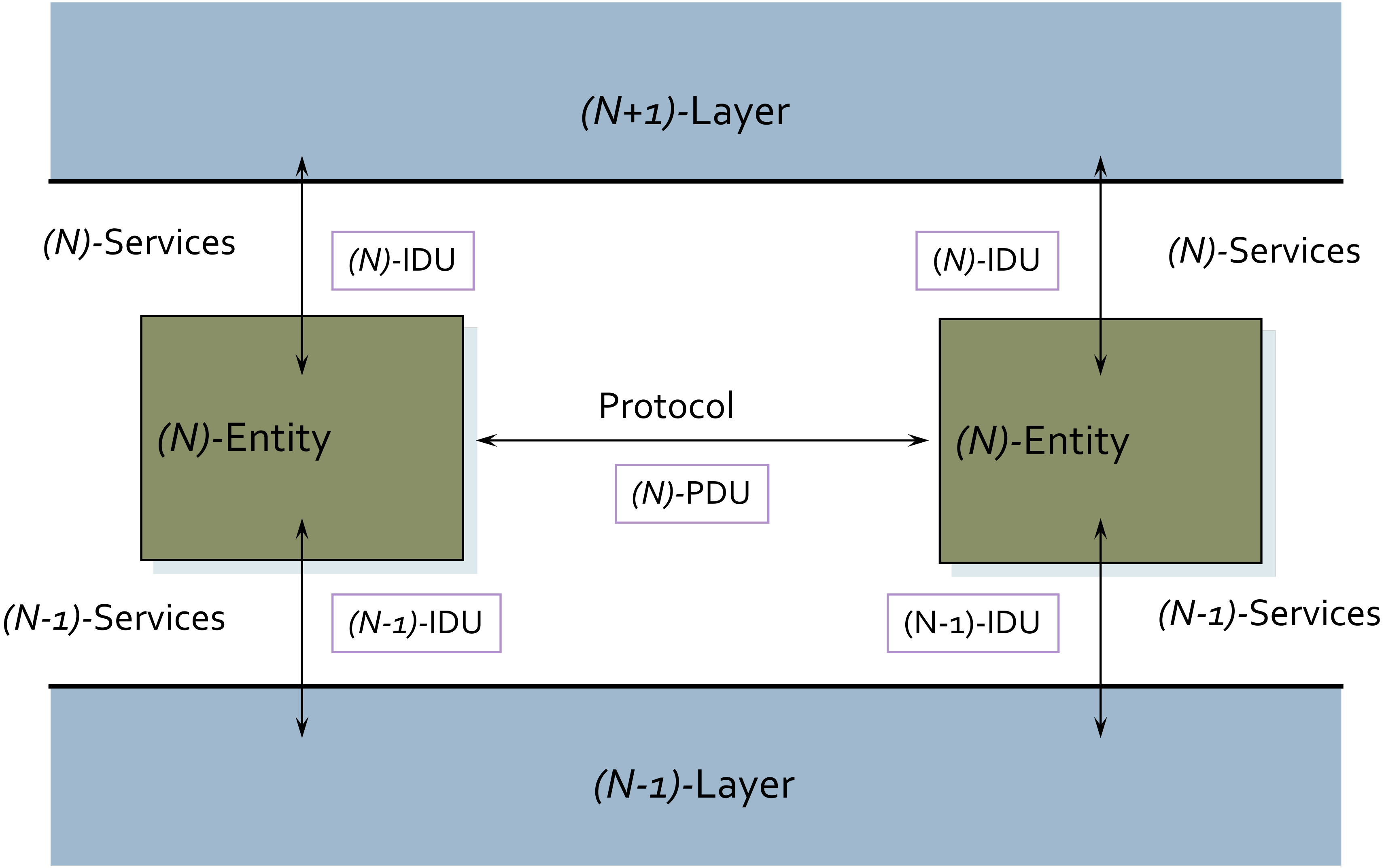


Telecommunication System

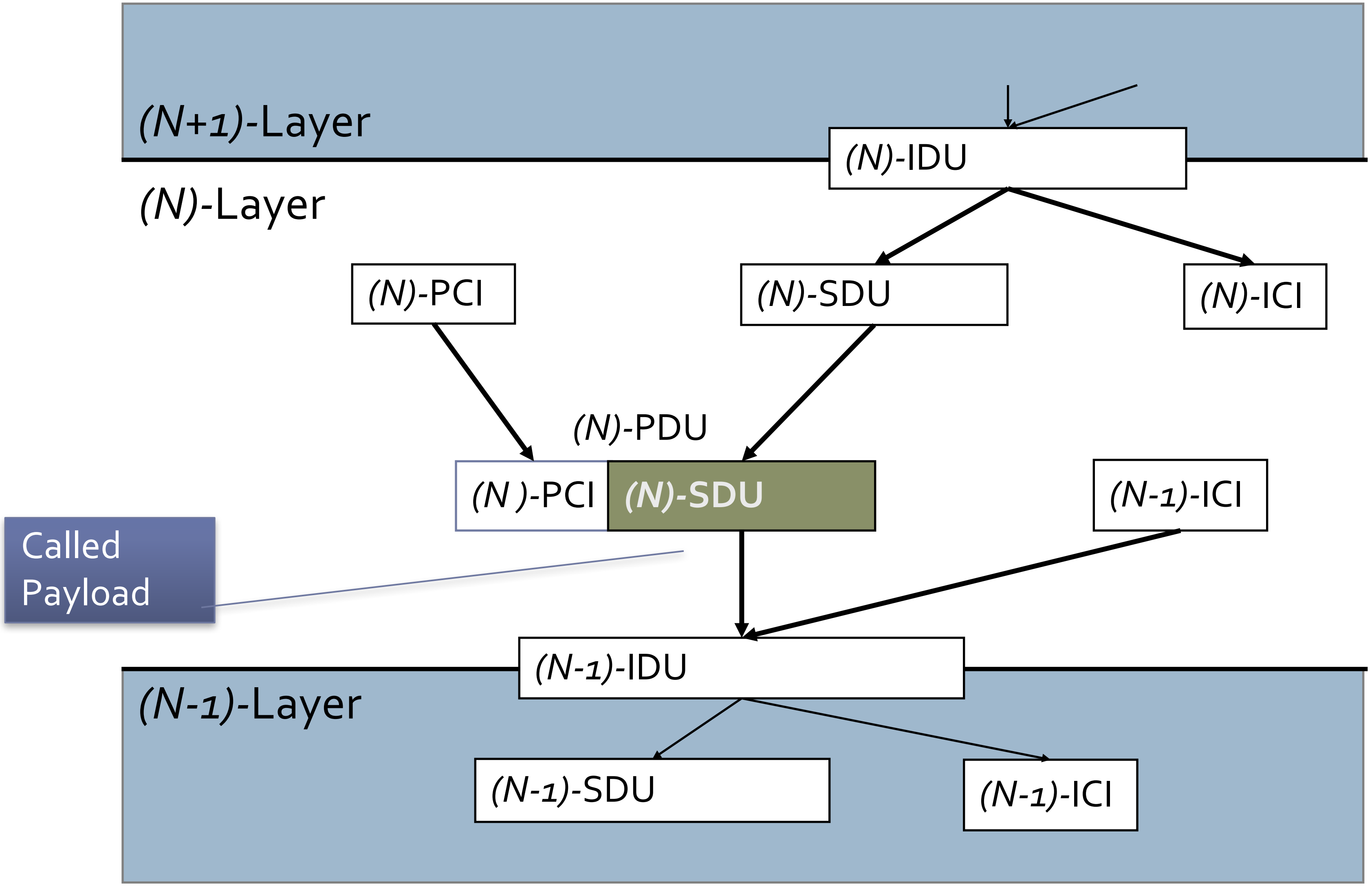


- Each layer provides a service to the overlaying one.
- A Service is provided by interaction of entities of the same layer according to a specified protocol.

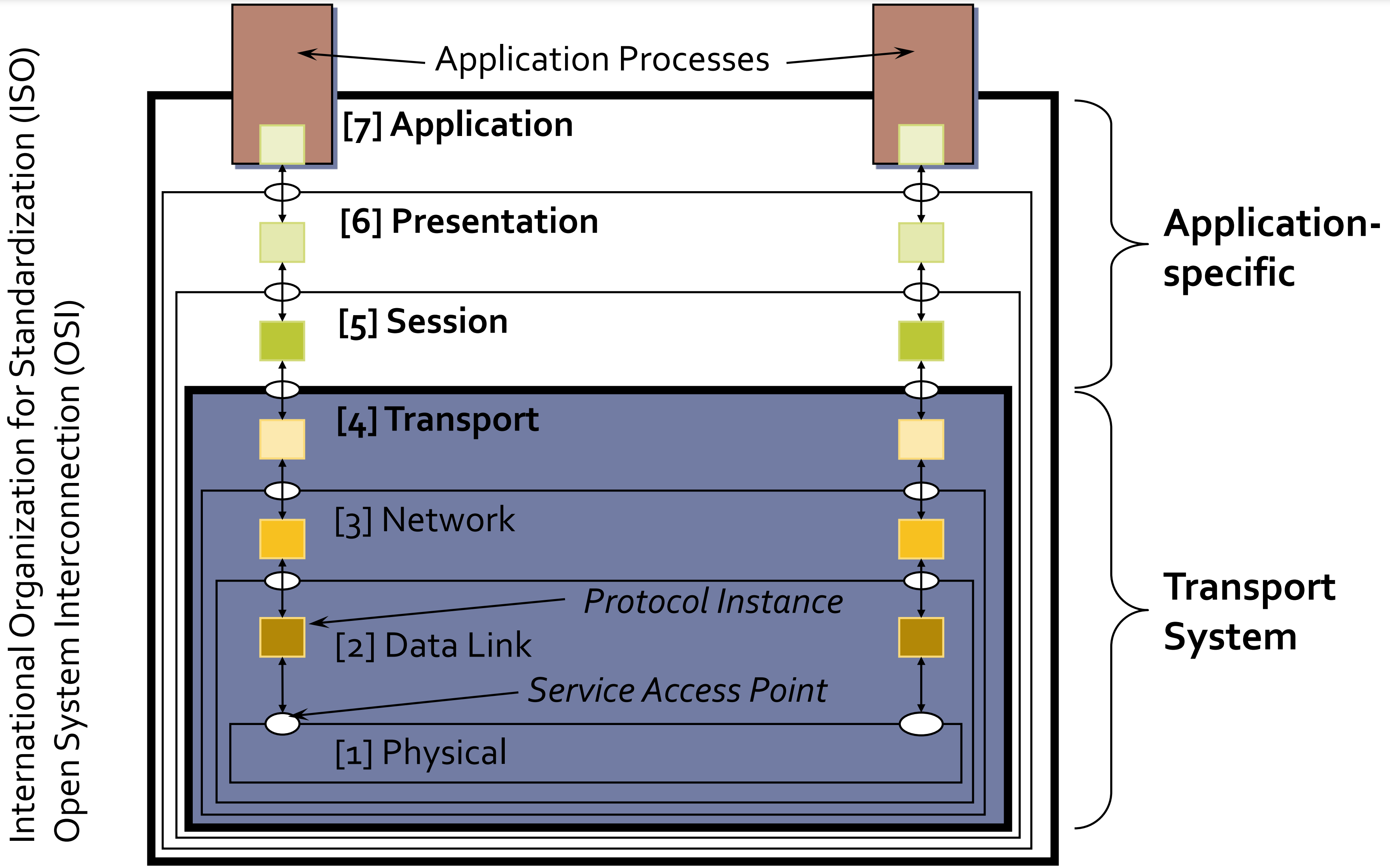
Comm. within/between Layers (ISO/OSI)



Layer Interaction (ISO/OSI)



ISO/OSI–Base Reference Model



Entities in the ISO/OSI Model

- **(N)-Layer**
  - All entities of the *(N)* hierarchy in all end systems
- **(N)-Entity**
  - Implementation of an *(N)*-Service in an end-system
  - Different types of *(N)*-Entities can exist implementing different protocols for a layer.
- **Peer-Entities**
  - Entities of the layer
  - Peer entities fulfill functions of a service by message exchange.
- **Data Units**
  - ***(N)*-IDU** (Interface Data Unit): Data unit passed from an (N+1)-Entity to an (N) one
  - ***(N)*-SDU** (Service Data Unit): Transparently transferred data between (N)-SAPs
  - ***(N)*-ICI** (Interface Control Information): Control parameters of an (N)-Service
  - ***(N)*-PDU** (Protocol Data Unit): Data units exchanged between (N)-Peer entities
  - ***(N)*-PCI** (Protocol Control Information): Data on protocol execution control exchanged between (N)-Peer Entities

Transport-Oriented Layers

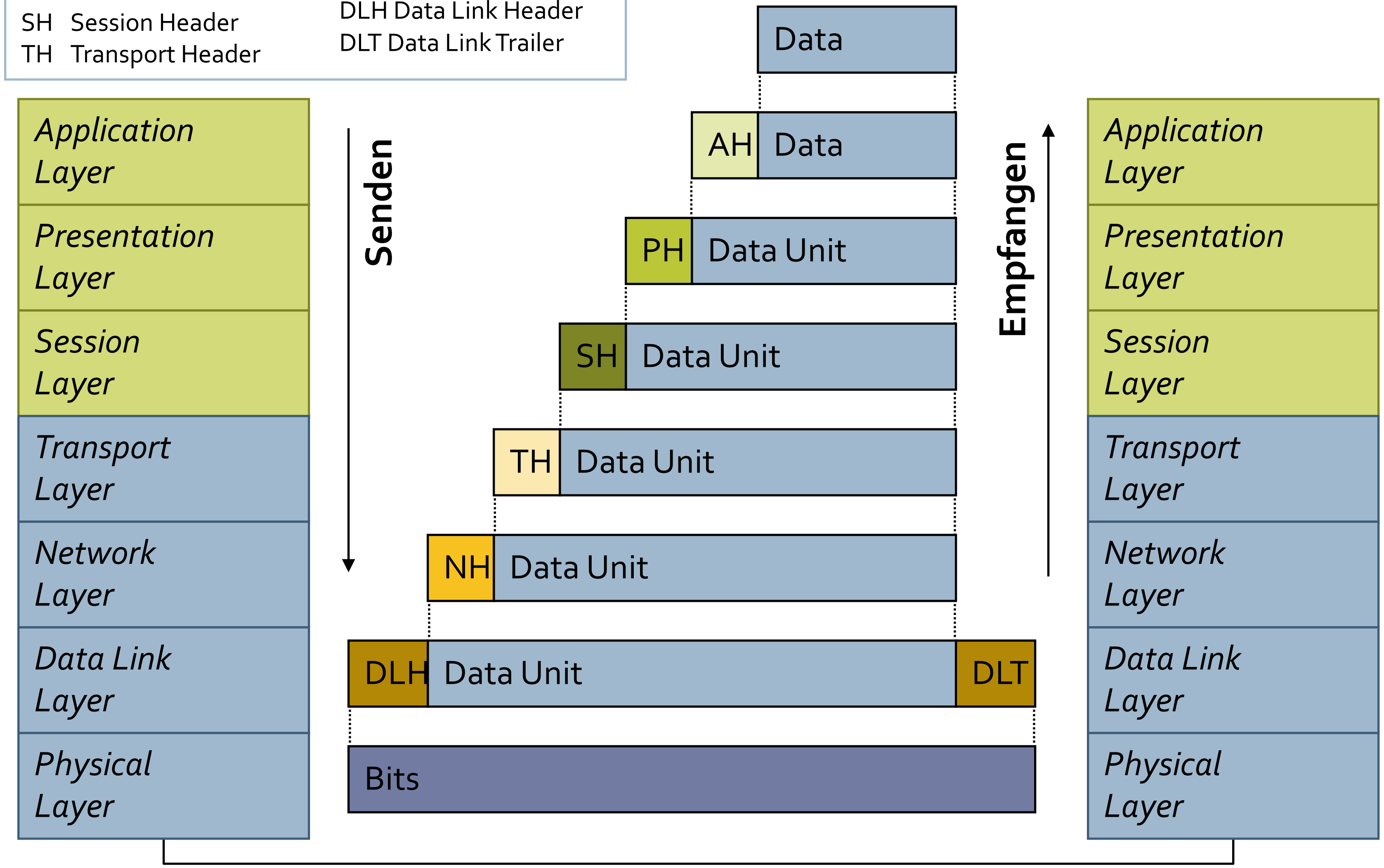
- **Physical Layer, Layer 1:**
  - possible transmission disruptions
  - fixed transmission quality/message length
  - no buffering
- **Data Link Layer, Layer 2:**
  - secure channels between directly connected service users
  - reduced transmission error probability
  - FIFO-buffering of packets
- **Network Layer, Layer 3:**
  - multiple connectivity
  - addressing: not only computers directly connected via a physical medium
  - quality is adjustable if there exist alternative route choices
- **Transport Layer, Layer 4:**
  - flexibly adjustable quality
  - variable message packet length
  - multiple use of connections
  - packet buffering

Application-Oriented Layers

- **Session Layer, Layer 5:**
  - Control of message exchange between two communication partners:
    - Rights handover
    - Transmission structuring
    - Reset agreement
- **Presentation Layer, Layer 6:**
  - Communication is made possible despite the different local data formats of the participants
  - Messages are composed of typed data
- **Application Layer, Layer 7:**
  - Exchange of messages with application-specific structure and purpose

Data Encapsulation (of iso-osi model)

AH	Application Header
PH	Presentation Header
SH	Session Header
TH	Transport Header
NH	Network Header
DLH	Data Link Header
DLT	Data Link Trailer



Application Layer

Presentation Layer

Session Layer

Transport Layer

Network Layer

Data Link Layer

Physical Layer

Senden

Data

AH Data

PH Data Unit

SH Data Unit

TH Data Unit

NH Data Unit

DLH Data Unit DLT

Bits

Empfangen

Application Layer

Presentation Layer

Session Layer

Transport Layer

Network Layer

Data Link Layer

Physical Layer

Binding and Endpoints

- **Binding:** *Description of a mechanism to be used to exchange data/messages between the user and the endpoint*
  - Binding defines a transport- and/or application protocol for data/message transfer
  - Binding defines rules of data/message exchange
  - Binding defines data/message encoding rules
- Important properties and types
  - Message exchange approach
  - Message exchange models
  - Transport- and application protocol
  - Integration with programming models

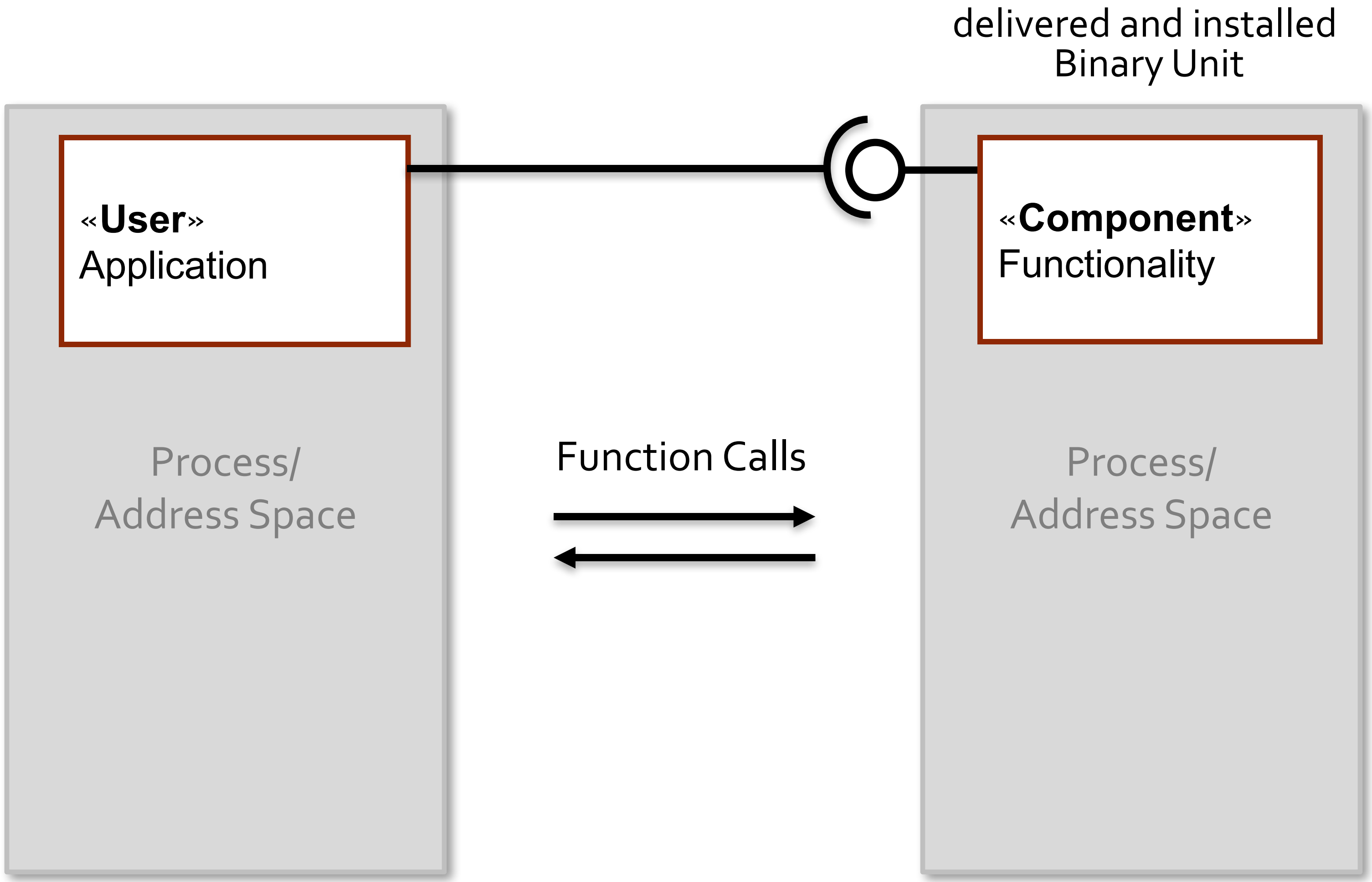
Chapter 4

COMMUNICATION IN PROGRAMMING

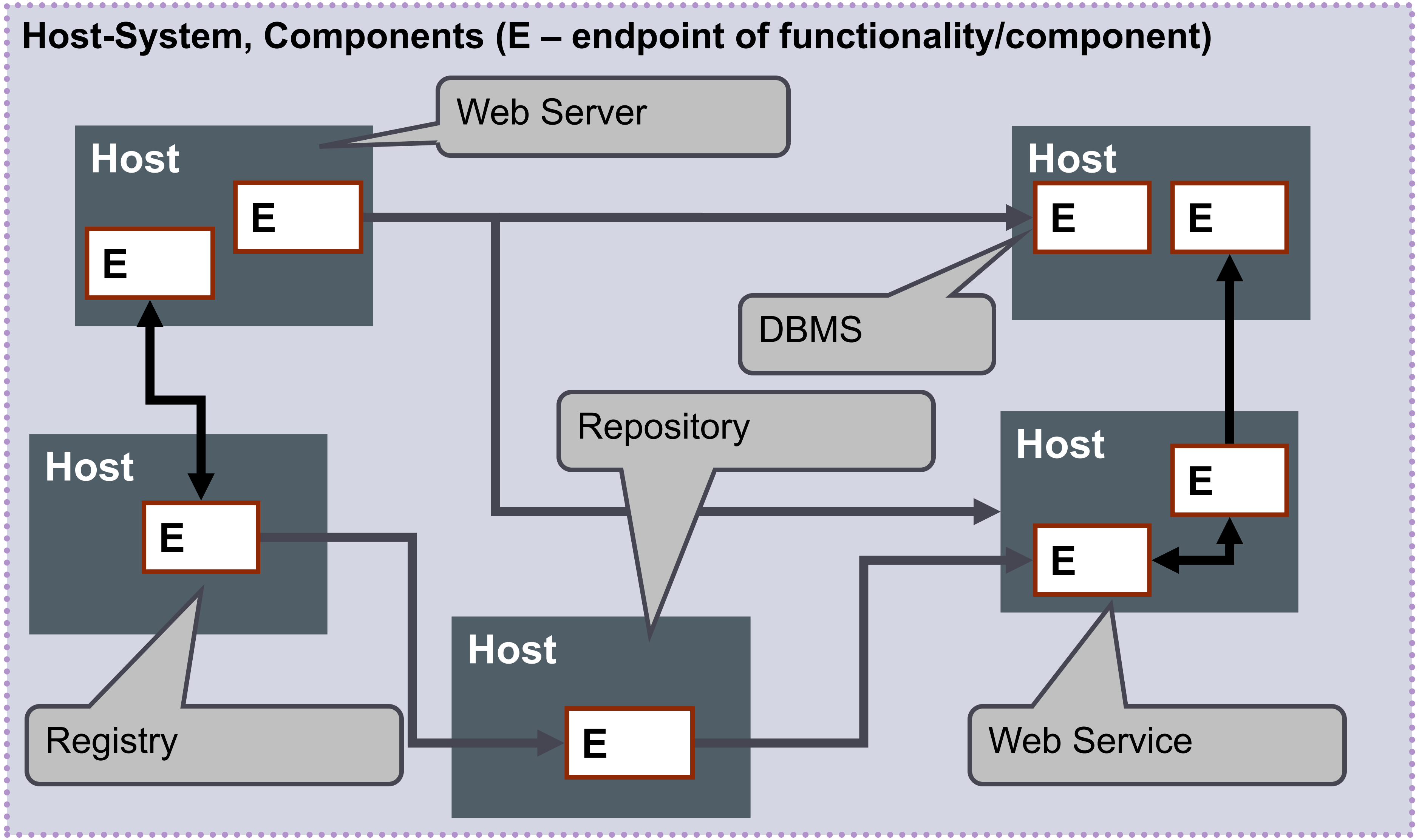
Classical Programming

- Key idea: Algorithm / Knowledge hidden in a reusable unit
- Programming approaches and their Reuse Unit
  - Procedural programming: Function
    - Aggregation of functions (step-by-step, conditional, loops etc.)
    - Knowledge abstraction low via function
    - Reuse via source code (same language)
  - Object-oriented programming: Object
    - Aggregation of data and functions (aka methods)
    - Knowledge abstraction high via objects, composition etc.
    - Reuse via source code (same OO-language)
  - Component-based Software Development: Component
    - Separation between interface and CBSD-knowledge
    - Knowledge abstraction packaged as binary unit
    - Reuse unit very high via distribution of binary

Components in use [UML-Diagram]



Shifting towards Distributed System



Reuse units in distributed systems?

- Different systems provide different abstractions (e.g. database, webserver, business logic, …)
  - From small-to-large scale
  - Independence of programming paradigm
  - Service – have more potential than components
- Focus on communication / message exchange
  - As means to providing functionality
  - Zero-Installation (provider did already)
  - Risks and challenges different from classical programming paradigm ones

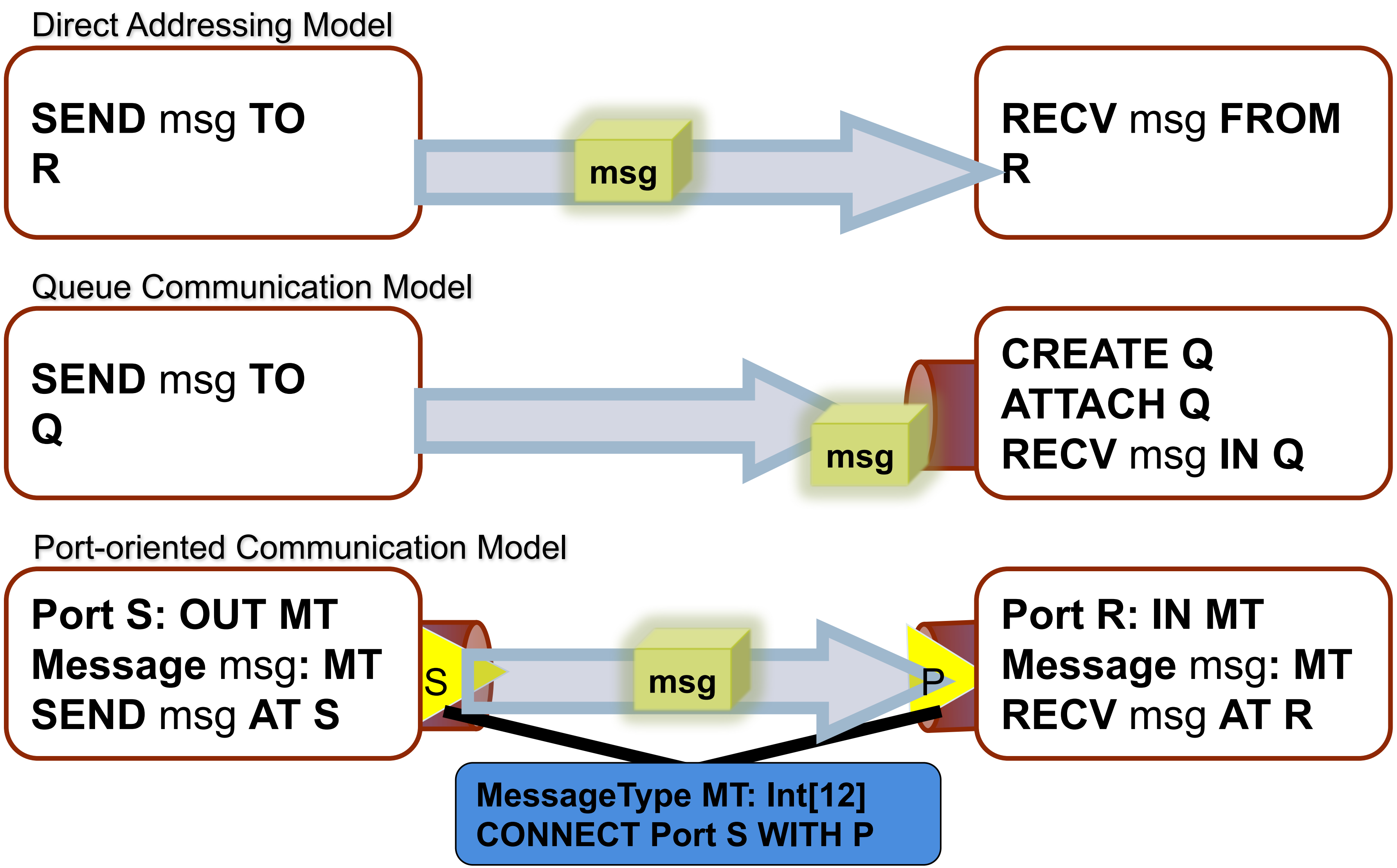
Communication

- **Communication** – Mechanism of data exchange between components that are executed on host systems
- Challenges
  - Message transport with regard to communication medium conditions
  - Interoperability and cooperation of components and host systems
  - System architecture support wrt. communication- or programming model
  - Much more: Quality aspects, security, trust

Message Exchange Approach

- Sender-Receiver paradigm
  - **Message**: (typed) Data is sent from a **Sender (S)** to a **Receiver (R)**
  - **Sender-Receiver relationship**
    - **Symmetric** (S and R know each other)
    - **Asymmetric** (only S knows R)
  - **Sender operation**
    - Sender operates **synchronously**
    - Sender operates **asynchronously**
- Message Exchange Models
  - Direct Addressing - Model
  - Queue Communication - Model
  - Port-oriented Communication - Model
  - Request/Response - Model
  - Pull/Push - Model

Message Exchange Models



Direct Addressing Model

SEND msg **TO R**

RECV msg **FROM R**

Queue Communication Model

SEND msg **TO Q**

CREATE Q
ATTACH Q
RECV msg **IN Q**

Port-oriented Communication Model

Port S: OUT MT
Message msg**: MT**
SEND msg **AT S**

Port R: IN MT
Message msg**: MT**
RECV msg **AT R**

MessageType MT: Int[12]
CONNECT Port S WITH P