mobile business

GOETHE UNIVERSITÄT FRANKFURT AM MAIN

Lecture 14

Business Informatics 2 (PWIN)
WS 2023/24

Questions & Answers

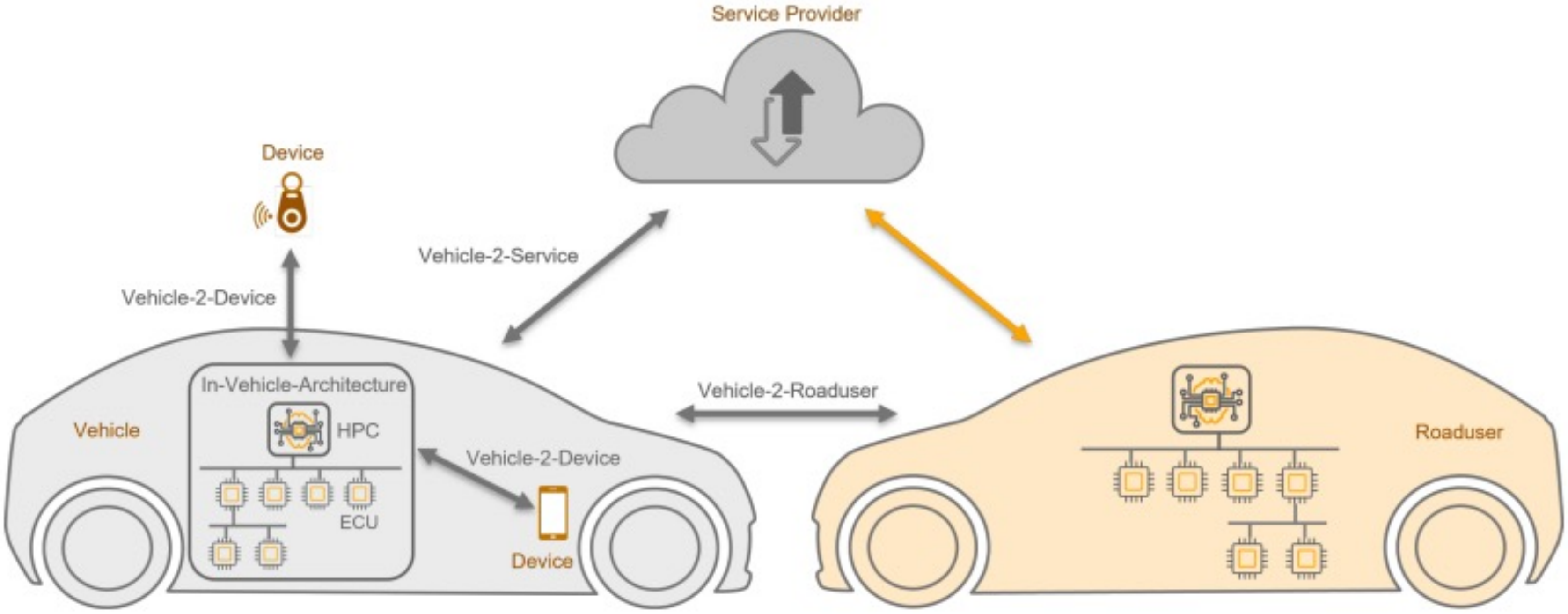
Prof. Dr. Kai Rannenberg

Chair of Mobile Business & Multilateral Security
Johann Wolfgang Goethe University Frankfurt a. M.

General Questions

- Wie wird die **Klausur aufgebaut** sein(eher offene Fragen oder Single-/Multiple-Choice)
- Wie sieht die **Relevanz der Gastvorträge** aus? Benötigt es nur Grundsätzliches Verständnis oder muss man die Einzelheiten komplett verstanden haben? Also zum Beispiel müssen wir genau wissen, wie das System Model: Entities & Core Model aussieht?
- Wenn wir auswendig lernen, reicht es, wenn wir uns die **Abkürzung merken**, oder müssen wir das Wort ausgeschrieben können? Zum Beispiel, was CSMA/CD bedeutet?

System Model
Entities & Core Model



Continental

PWIN II -- Guest Lecture

Sebastian Pape © Continental Automotive Technologies GmbH

Public

December 12th, 2023

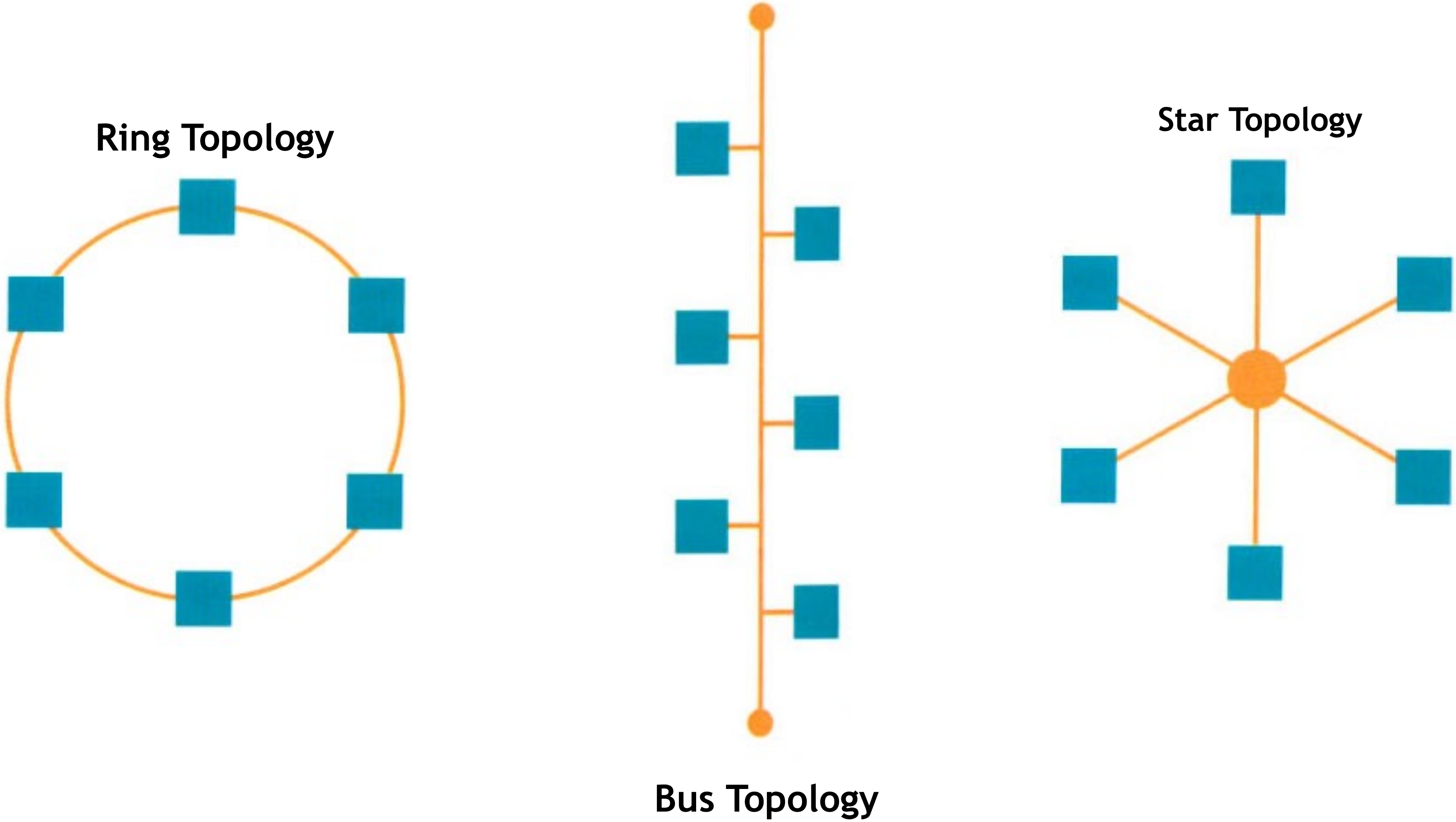
61

Ethernet

- Definition of wiring and signalling standards for local area networks (LAN)
- Developed in the beginning of the 1970s by Xerox and from 1980 continued by DEC, Intel and Xerox (DIX group)
- Implementation of an access method called Carrier Sense Multiple Access/Collision Detection (CSMA/CD), an enhancement of the Aloha concept, which was previously developed at the University of Hawaii
- Thus, Ethernet is a product name by Xerox, DEC and Intel, in which the product itself had not been compatible to the IEEE specification 802.3 for years.

Source: Tanenbaum, A. (2006)

Wired Network Topologies



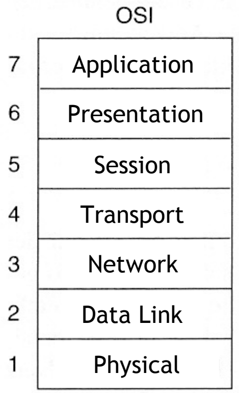
Source: Laudon, K.C., Laudon, J.P., Schoder, D. (2010), p. 346.

Questions VL3/4/5

1. Bei der **1. OSI-Schicht** woher weiß man ob Bits über Luft oder Glasfaser übertragen werden sollten?
2. Bei der **7. OSI-Schicht** woher weiß man ob man http oder dns nehmen soll?
3. Inwiefern erfüllt eine **MAC Adresse** die Aufgaben der zweiten Schicht (Fehlerkontrolle, flow control, Fehlererkennung und korrektur)?
4. Außerdem stelle ich mir die Frage, ob OSI Reference Model genauso funktioniert wie der **5-Layer model** (nur mit 7 Schichten)?
   Wenn die Datenübertragung nur von oberen auf die untere Schicht funktioniert, wie kann dann die Nachricht an den anderen Nutzer übertragen werden (da dass dann die Datenübertragung von unteren bis zum oberen Schicht wäre)? Für ein Beispiel wäre ich dankbar.

OSI Reference Model

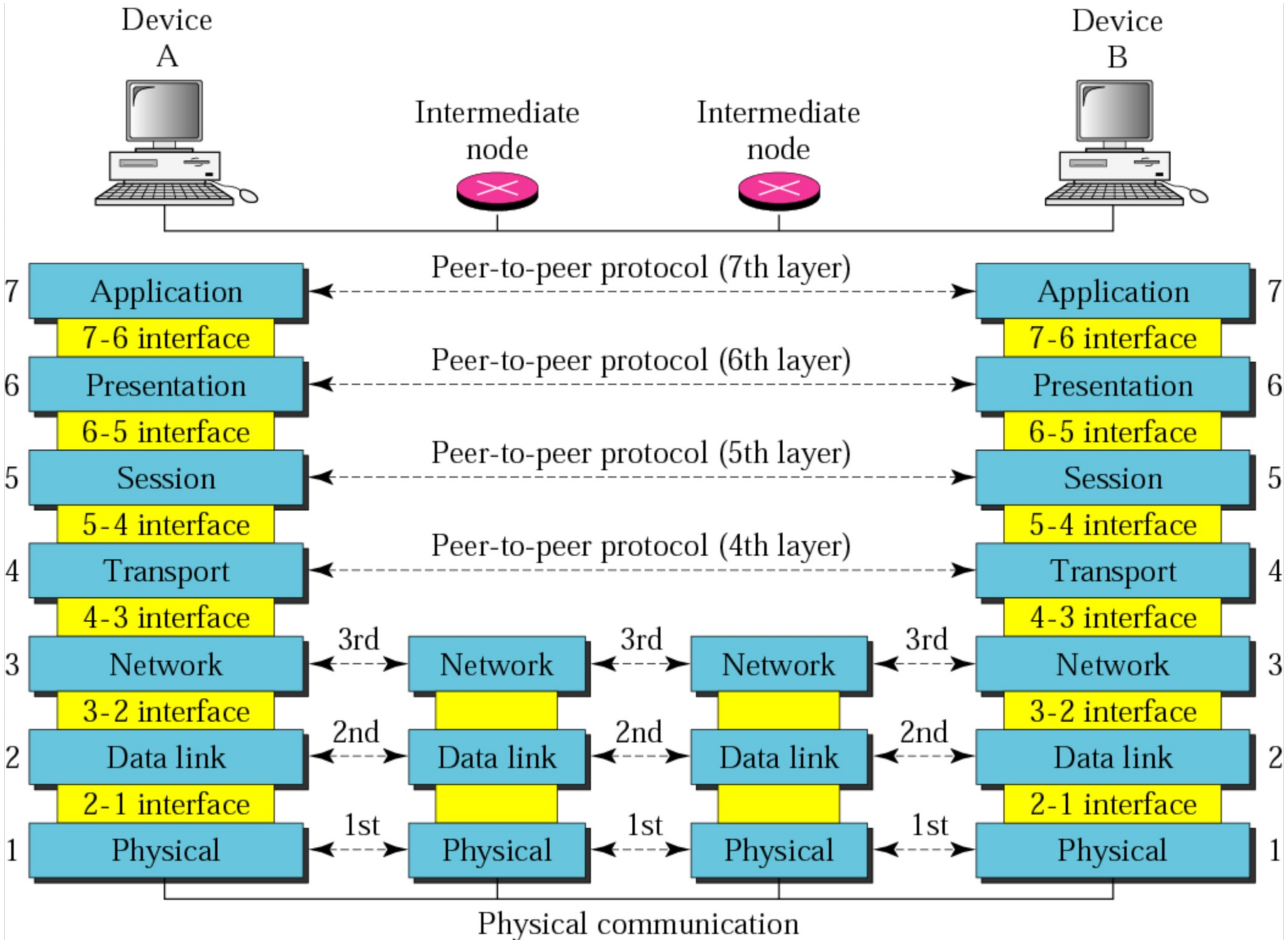
- The OSI reference model consists of **seven** layers:
  - **Layers 1-4** are **transport-oriented** layers as they focus on the transportation of bits.
  - **Layers 5-7** are **application-oriented** layers as they support the use of data transport applications.
  - The following basic ideas have lead to the 7 layers:
    - Whenever a new **level of abstraction** is necessary, a new layer has to be introduced
    - Each layer should deal with one well-defined task or function.



	OSI
7	Application
6	Presentation
5	Session
4	Transport
3	Network
2	Data Link
1	Physical

Source: Tanenbaum (2006), p. 54-58

OSI Reference Model



Device A

Device B

Intermediate node

Intermediate node

7 Application — 7-6 interface — 6 Presentation — 6-5 interface — 5 Session — 5-4 interface — 4 Transport — 4-3 interface — 3 Network — 3-2 interface — 2 Data link — 2-1 interface — 1 Physical

Peer-to-peer protocol (7th layer)

Peer-to-peer protocol (6th layer)

Peer-to-peer protocol (5th layer)

Peer-to-peer protocol (4th layer)

Network, Data link, Physical (intermediate nodes)

3rd, 2nd, 1st

Physical communication

Source: http://studyregular.blogspot.com/2016/10/computer-networks-topic-osi-model.html

Layer 1: Physical Layer

- The **physical layer** is the lowermost layer of the OSI model. It defines **mechanical**, **electrical** and **time-related** specifications for interfaces to the network. It deals with the **transmission of bits** over a communication channel.

  - Typical transmission-related questions are:
    - How to represent bits?
    - Definition and assignment of connectors and pins.
    - How to achieve full-duplex transmission (both directions simultaneously)?
    - Connection set-up and termination
    - Which are the available transmission media?

Source: Tanenbaum (2006), p.105

Layer 7: Application Layer Definition

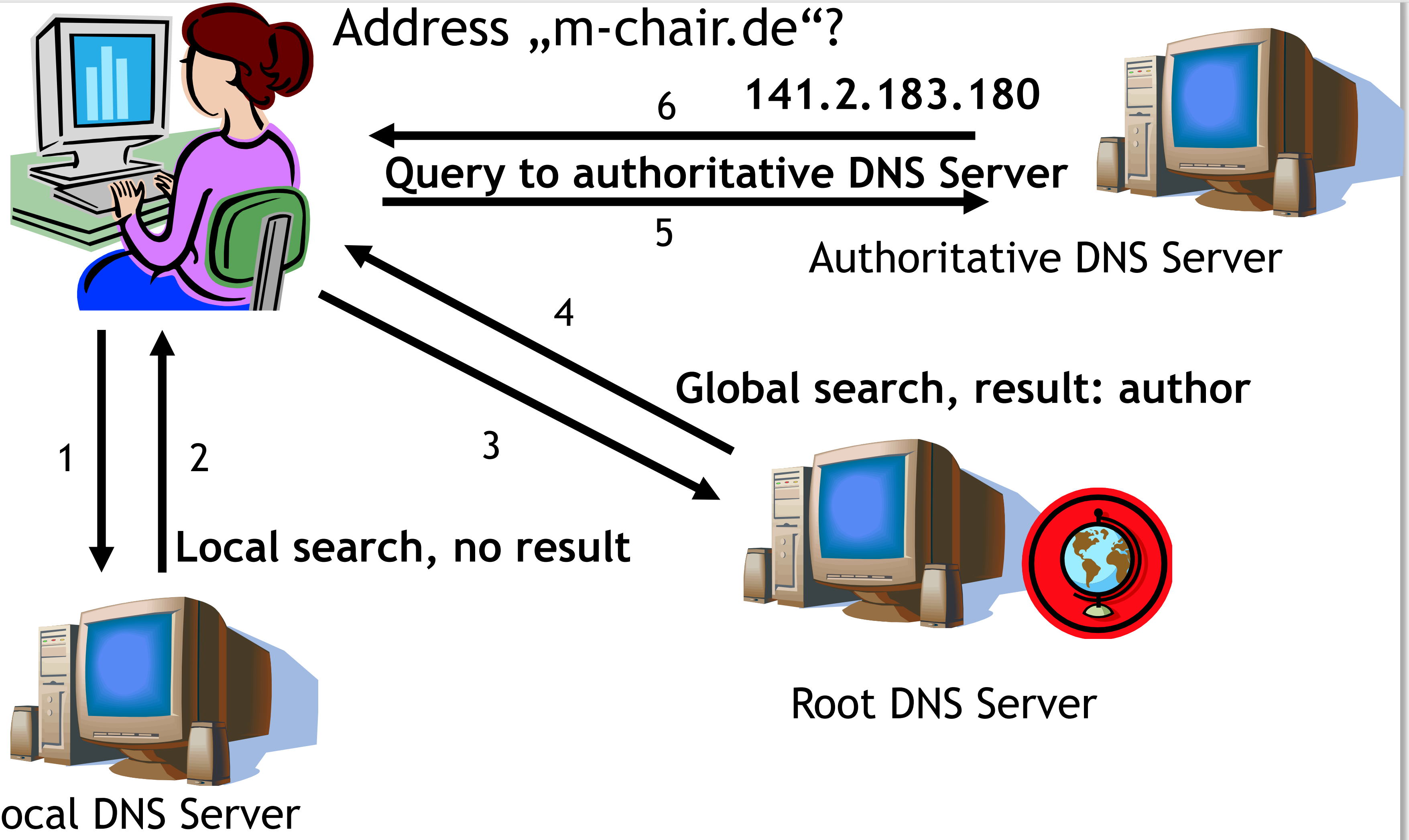
- All layers below the application layer serve the purpose of ensuring reliable transport of data, but they do not perform any specific tasks for the user.
- The application layer contains a multitude of protocols which are often needed by users in order for applications to work.
  - **DNS**
  - E-Mail
  - FTP
  - HTTP
  - Skype
  - …

Source: Tanenbaum (2006)

Layer 7: Application Layer Example: Domain Name System

- Introduction of a distributed database, the „Domain Name System“ (DNS).
- The DNS is located on the application layer.
- The DNS translates names into addresses, addresses into names or provides e.g. information about the IP address of a mail server for a domain (mail exchange or MX record).
- There are three types of DNS Servers:
  - Local: DNS server inside the organisation (of your own)
  - Root: Root DNS of a domain
  - Authoritative: Authoritative name servers are assigned to be responsible for particular domain names.

Layer 7: Application Layer
Example: Domain Name System



Address „m-chair.de“?

1

2

Local search, no result

3

4

Global search, result: author

5

Query to authoritative DNS Server

6

141.2.183.180

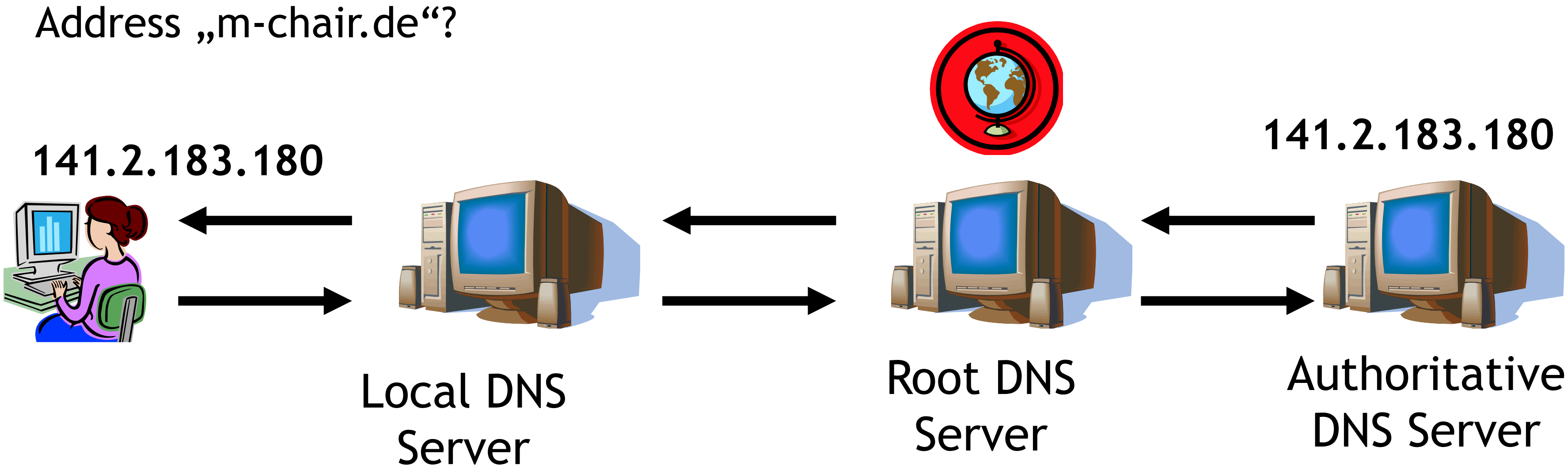
Local DNS Server

Root DNS Server

Authoritative DNS Server

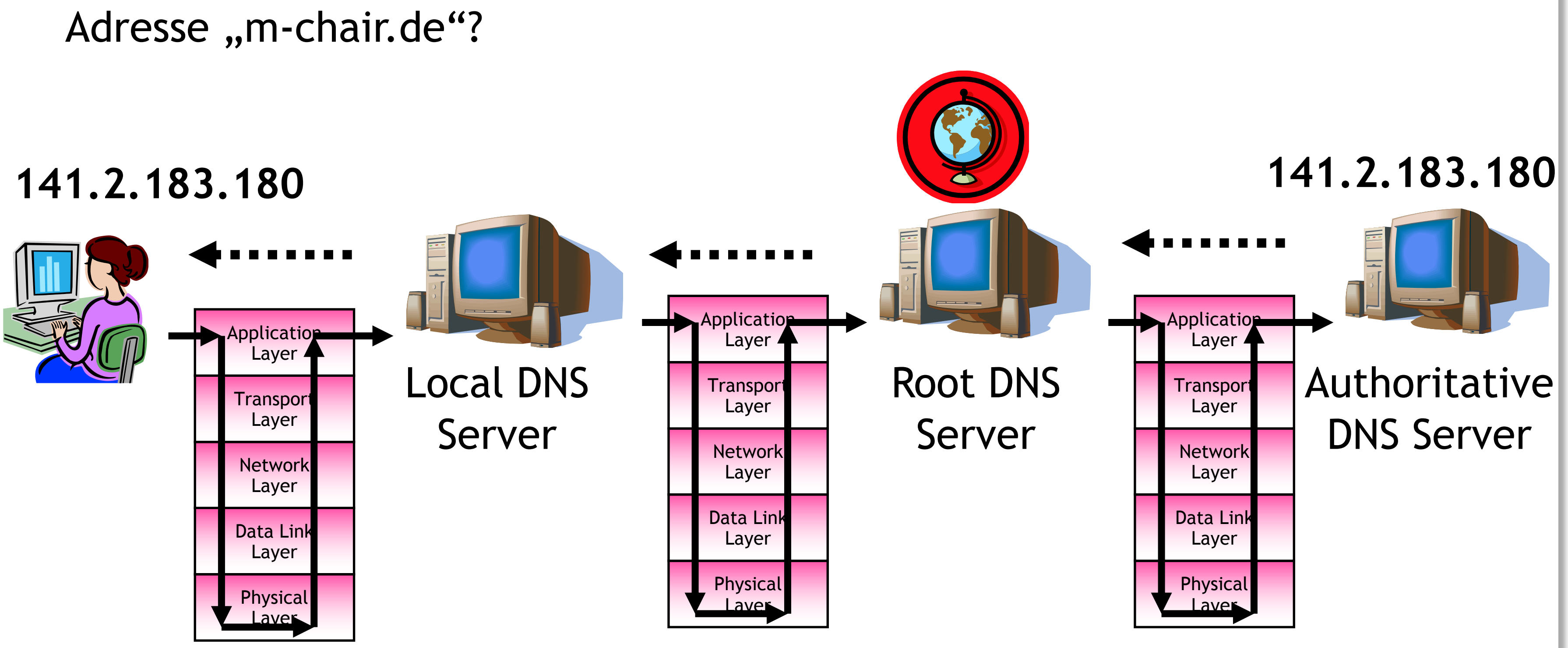
Layer 7: Application Layer
Example: DNS Forwarding

Aim: Load-balancing for DNS queries



Layer 7: Application Layer
Example: DNS and Layers

DNS queries involve all the layers of the network stack.

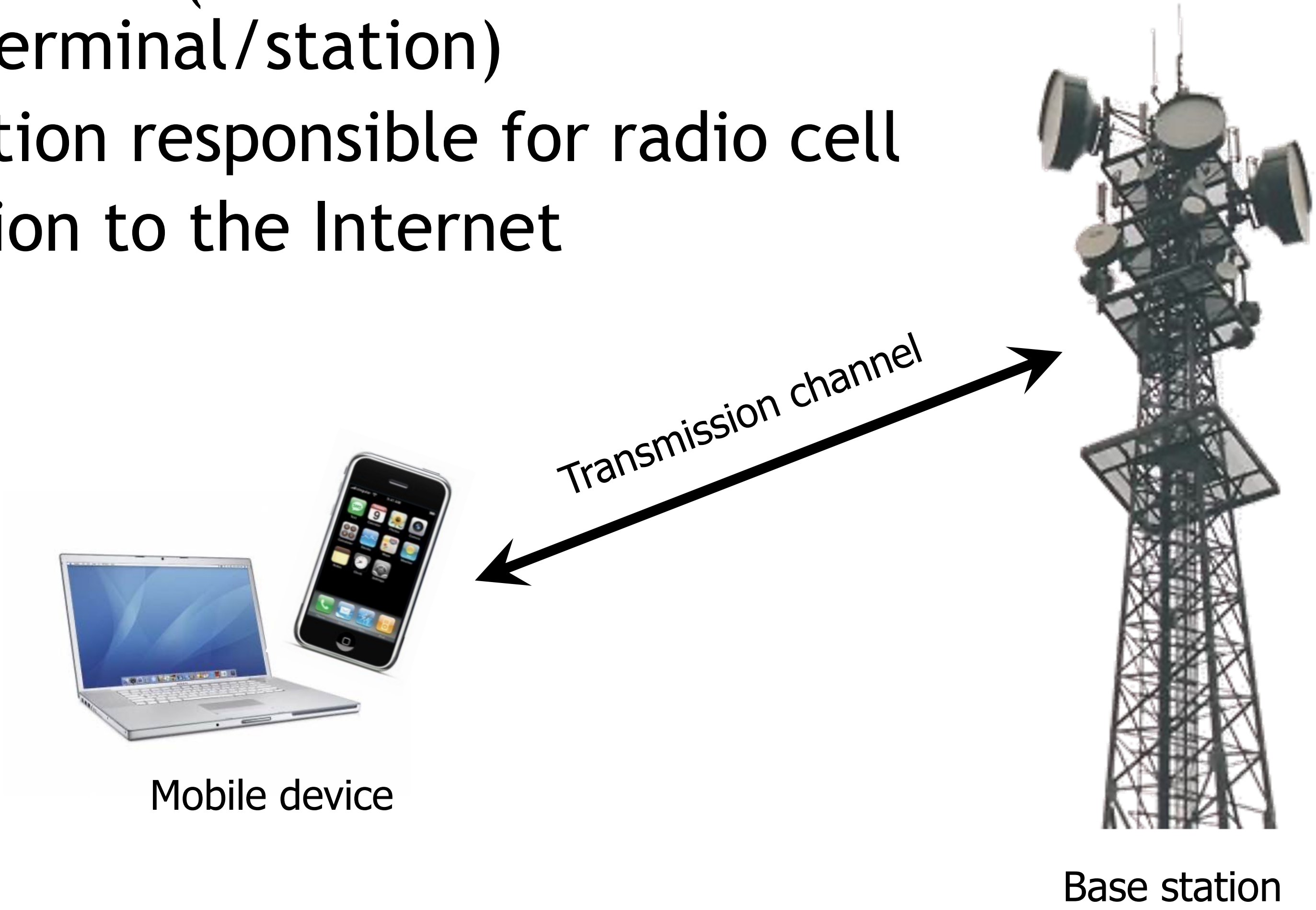


Questions GSM/UMTS

- Könnten Sie in der Q&A Session bitte nochmal die Systemarchitektur der GSM-Dienste anhand eines Beispiels erläutern? Also zum Beispiel wie die Verbindung des Handys durch die 3 Systeme (RSS, NSS, OSS) geschaltete/abgefragt wird.
- Können Sie bitte nochmal erklären wie beim **GSM (Global System for Mobile Communication)** ein Anruf zwischen zwei Leuten an unterschiedlichen Orten stattfindet (Folie 12 der vierten Vorlesung)
- Wie relevant ist die **UMTS Infrastruktur**. Genauso relevant für die Klausur, wie die GSM Infrastruktur?

Mobile Voice & Data Communication Services

- Mobile device (sometimes named mobile terminal/station)
- Base station responsible for radio cell
- Connection to the Internet



Transmission channel

Mobile device

Base station

Mobile Voice & Data Communication Services

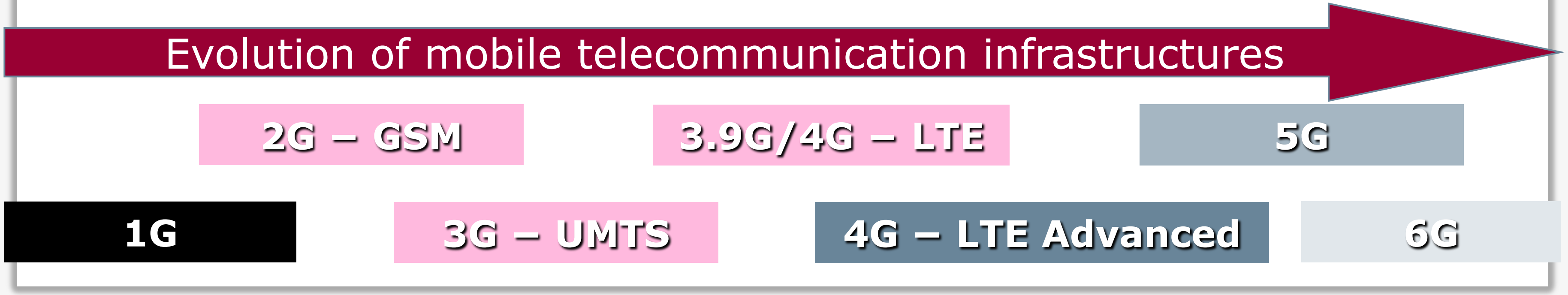
Fundamental mobile communication services

- Mobile voice
  - GSM, UMTS
- Mobile data
  - GPRS, EDGE, 3G networks (UMTS, HSDPA), 4G networks (LTE, LTE advanced), 5G networks (forthcoming)
- Mobile messaging
  - Text messaging (SMS), Multimedia messaging (MMS)
- Mobile Machine-To-Machine
  - GPRS, EDGE, 3G networks (UMTS, HSDPA), 4G networks (LTE, LTE advanced), 5G networks, 6G networks (forthcoming)

→ Lecture focuses on mobile data communications

Mobile Voice & Data Communication Services

- *1st Generation (1G) - Analogue networks*
- *2nd Generation (2G) - GSM networks*
  Global System for Mobile Communications
- *3rd Generation (3G/3.5G) - UMTS/HSPA/HSPA+*
  Universal Mobile Telecommunications System
  High Speed Packet Access / Evolved HSPA = HSPA+
- *3.9G or 4G - LTE*
  Long Term Evolution
- *4th Generation (4G) - LTE Advanced*
- *5th Generation (5G) - Mobile broadband*
- *6th Generation (6G) - Ubiquitous wireless intelligence [Latva19]*



Evolution of mobile telecommunication infrastructures

1G

2G – GSM

3G – UMTS

3.9G/4G – LTE

4G – LTE Advanced

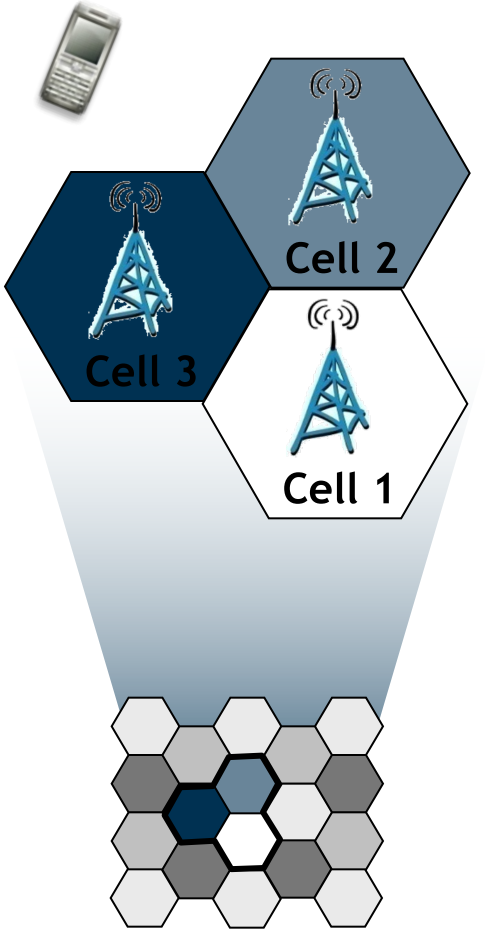
5G

6G

Cell-Based Communication (CBC)

What is a Cellular Network?

- Cellular networks are radio networks consisting of several transmitters.
- Each transmitter or base station, covers a certain area ➲ ***a cell.***
- Cell radii can vary from tens of meters to several kilometres.
- The shape of a cell is influenced by the environment (buildings, etc.) and usually neither hexagonal nor a perfect circle, even though this is the usual way of drawing them.



[Schiller2003]

mobile business

Cell-Based Communication (CBC)

Advantages of CBC (1)

- Cellular networks offer a number of advantages compared to centralised radio systems:
  - ***Higher capacity:*** Cells offer the possibility to "*reuse*" the transmission frequencies assigned to mobile devices (e.g. by multiplexing). In order to do so, the networks need a thorough planning of the position of base stations and their frequencies.
    - ➲ More users can use the infrastructure
  - ***Reduced transmission power:*** Reduced power usage for the mobile device, due to the fact that only a limited amount of transmission power is needed in a small cell, compared to a far away base station.
    - ➲ Reduced power consumption for mobile devices

Cell-Based Communication (CBC)

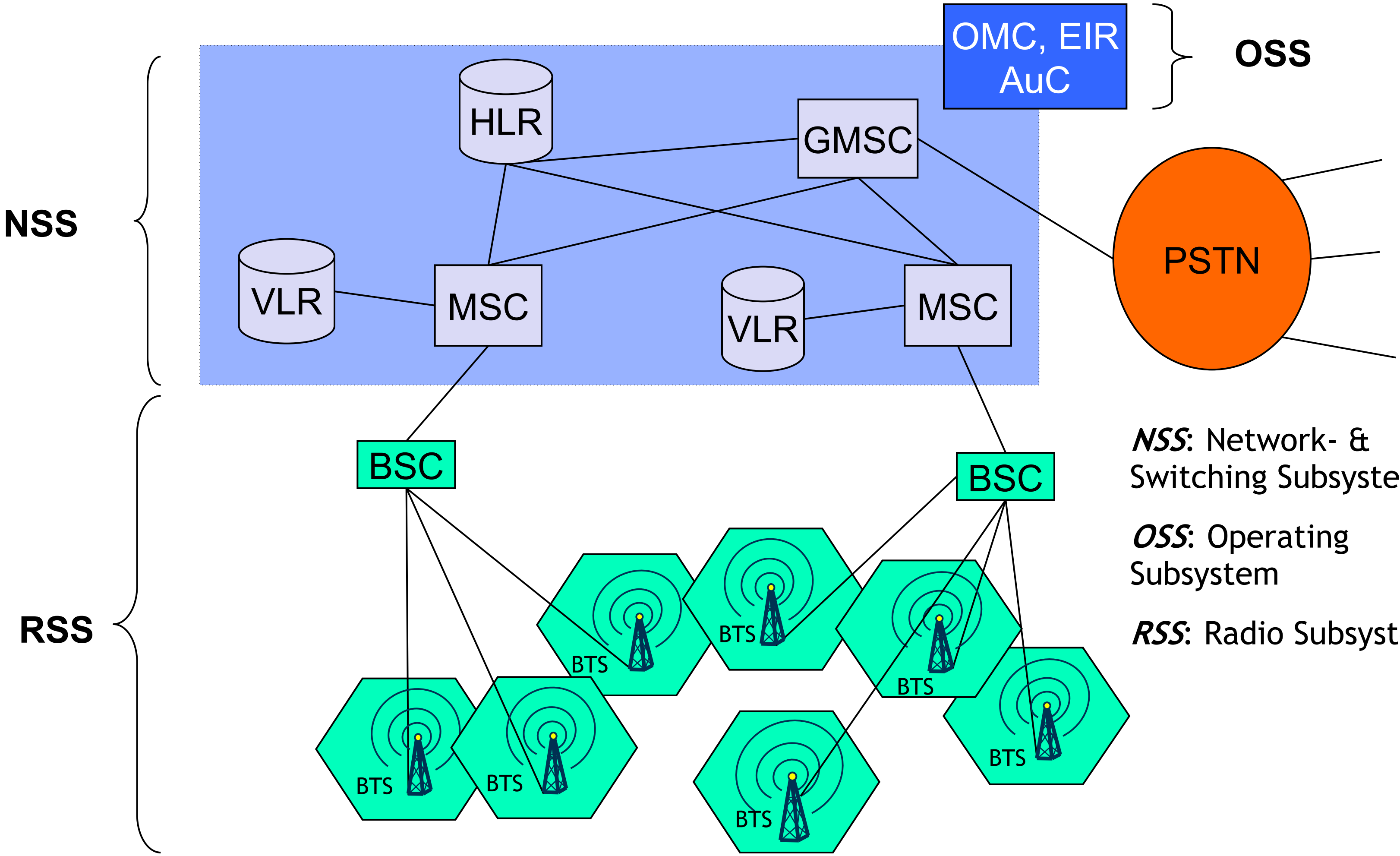
Advantages of CBC (2)

- Cellular networks offer a number of advantages compared to centralised radio systems:
  - ***Robustness:*** Cellular systems are decentralised with regard to their base stations. In the case that one antenna fails, only a small area gets affected.
    - ➲ Failure of one base station does not affect the complete infrastructure
  - ***Better coverage:*** Cells can be adapted to geographic conditions (mountains, buildings, etc.).
    - ➲ Better availability of the infrastructure

But: A complex and costly infrastructure is required, in order to link all base stations. This includes switches, antennas, location registers, etc.

GSM (2G)

System Architecture



NSS: Network- & Switching Subsystem

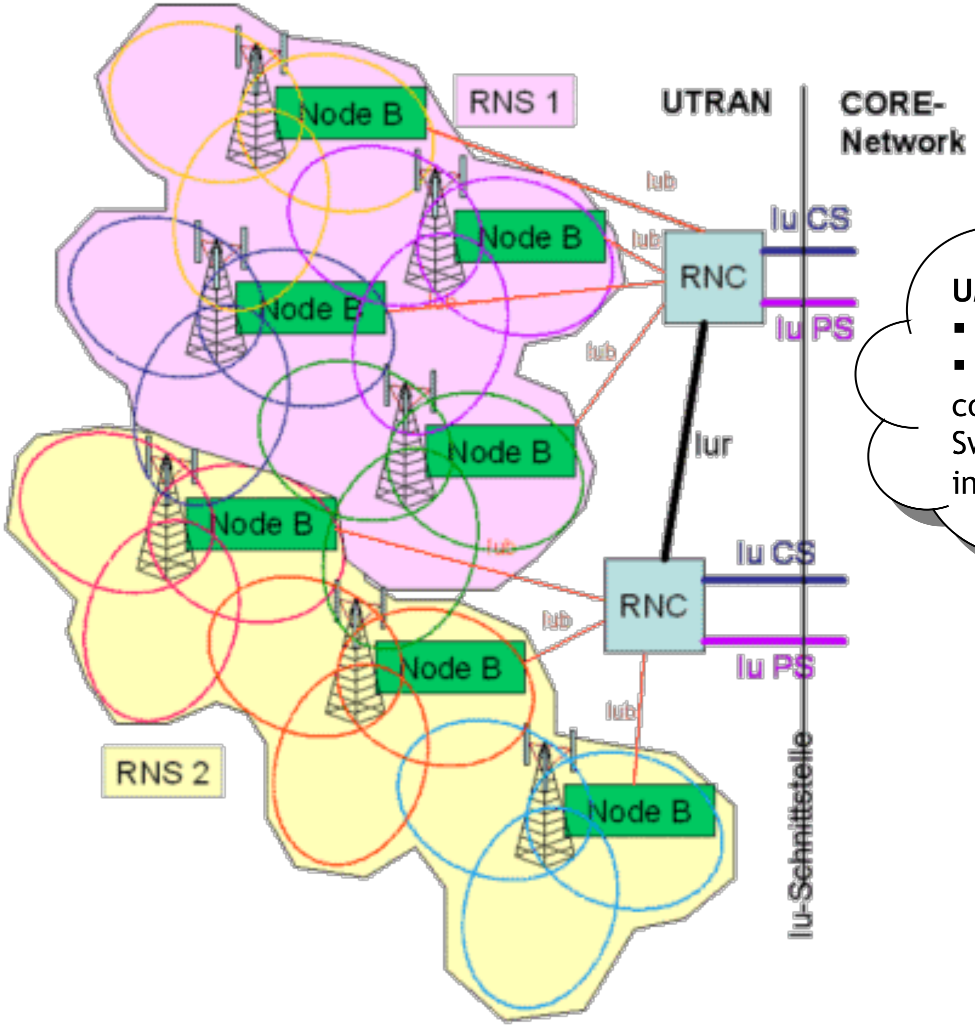
OSS: Operating Subsystem

RSS: Radio Subsystem

Based on [Schiller2003]

CBC using the example of UMTS (3G) System Architecture

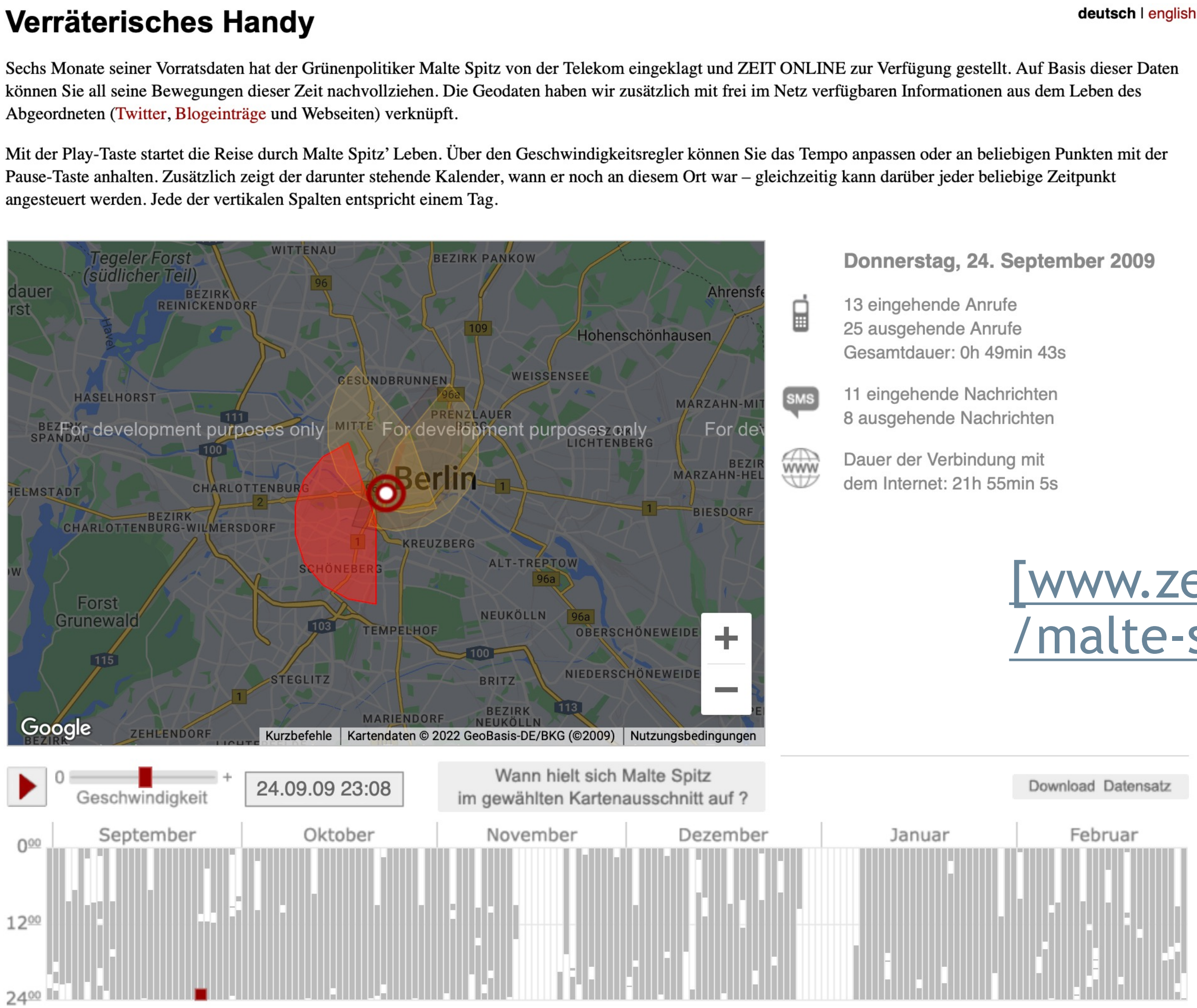
- *UTRAN:* UMTS Terrestrial Radio Access Network
- *RNS:* Radio Network Subsystem
- *RNC:* Radio Network Controller (controls the Node Bs)
- *Node B:* UMTS base stations (equivalent to base transceiver stations (BTS) in GSM



UMTS Core network
- is not shown here in detail
- UMTS Core network corresponds to Network- & Switching Subsystem (NSS) in GSM

Source: UMTSlink.at (2011)

Data retention of cell-based location data



Verräterisches Handy

deutsch | english

Sechs Monate seiner Vorratsdaten hat der Grünenpolitiker Malte Spitz von der Telekom eingeklagt und ZEIT ONLINE zur Verfügung gestellt. Auf Basis dieser Daten können Sie all seine Bewegungen dieser Zeit nachvollziehen. Die Geodaten haben wir zusätzlich mit frei im Netz verfügbaren Informationen aus dem Leben des Abgeordneten (Twitter, Blogeinträge und Webseiten) verknüpft.

Mit der Play-Taste startet die Reise durch Malte Spitz' Leben. Über den Geschwindigkeitsregler können Sie das Tempo anpassen oder an beliebigen Punkten mit der Pause-Taste anhalten. Zusätzlich zeigt der darunter stehende Kalender, wann er noch an diesem Ort war – gleichzeitig kann darüber jeder beliebige Zeitpunkt angesteuert werden. Jede der vertikalen Spalten entspricht einem Tag.

Donnerstag, 24. September 2009

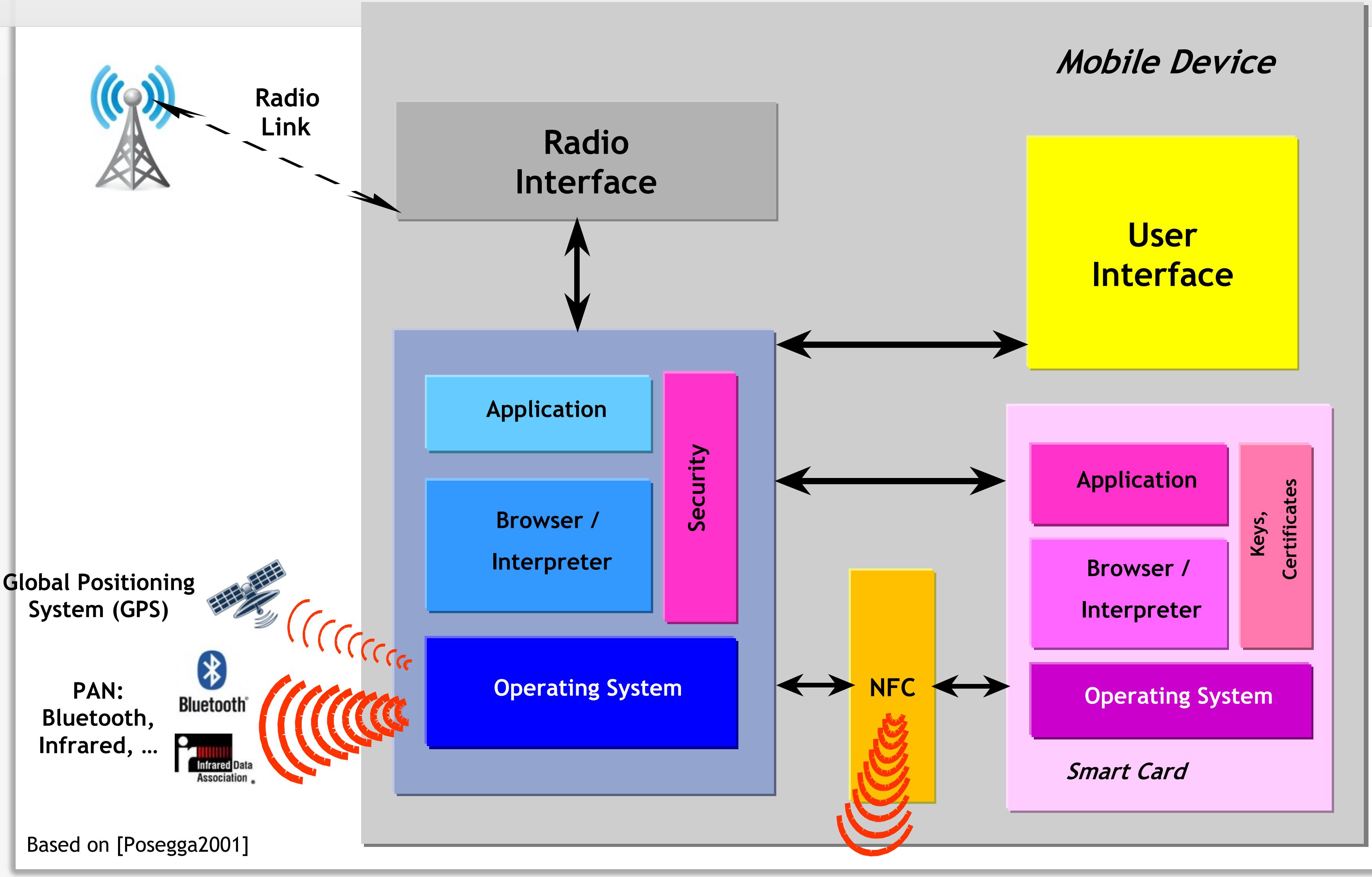
13 eingehende Anrufe
25 ausgehende Anrufe
Gesamtdauer: 0h 49min 43s

11 eingehende Nachrichten
8 ausgehende Nachrichten

Dauer der Verbindung mit
dem Internet: 21h 55min 5s

[www.zeit.de/datenschutz/malte-spitz-vorratsdaten]

Mobile Device & Operating Systems – Functional Architecture



Based on [Posegga2001]

Questions VL7 (ICT Projects)

- Kann in der letzten Vorlesung Stunde der Professor noch mal erklären wie Edge Computing genau funktioniert (kernnetzlast, Latenzlast) und mehrere Beispiele geben.

Questions VL7: ICT Projects

- Inwieweit müssen wir uns alle **fehlgeschlagenen ICT-Projekte** merken? Dienen diese Beispiele nur dem Verständnis?

Edge Computing Architecture

Pushing "intelligence" to the edge of the network

- Why edge computing?
  - Proliferation of IoT devices producing data to be processed
  - Limitations due to centralized nature of cloud architectures
  - Clouds' quality of service impacted by distance to data center
  - Steady decline in the cost of processing power & appearance of intelligent endpoint devices that sense and can make inferences
- What is edge computing?
  - Distributed approach to computing at/near network-endpoints
  - Heterogeneous nomenclature [edge (2004-), fog (2012-) and mist (2015-) computing] due to multiple interests and approaches
  - None of them synonymous with cloud computing

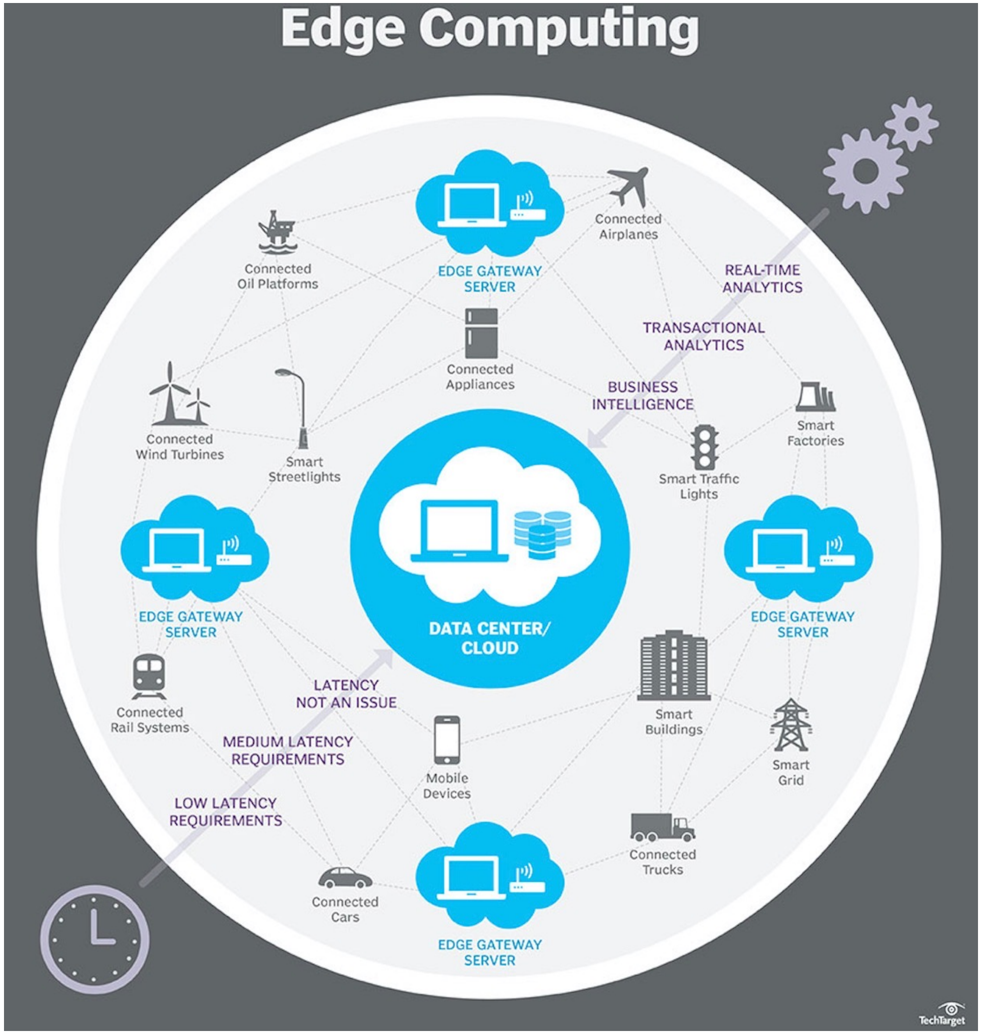
Edge Computing Architecture: Advantages and Applications

- Lower core network load and transmission costs since less / pre-processed data is transmitted

- Reduced load on cloud server / data centre and more efficient resource use possible

- Reduced latency key enabler of use-cases like autonomous driving

- New functionalities provided by intelligent endpoint devices



Source: Tech Target, 2017

Edge Computing Architecture: Disadvantages

- Security challenges
  - Distributed architecture increases number of attack vectors.
  - The more "intelligent" the device, the more vulnerable to infections and exploits (e.g. integrated webserver)
  - IP address spoofing, man-in-the-middle attacks
- Trust and authentication concerns
- Risk of a configuration drift when inferior device management solutions are implemented
- Fixed physical location and cost of hardware
- "Hidden" licensing costs of endpoint devices (base version vs. additional functionalities)
- Sometimes adding unnecessary overhead and complexity to the system, as not always needed for IoT applications

Questions VL9-12

1. **Könnten Sie in der Q&A Session bitte nochmal das Klassendiagramm erklären?**

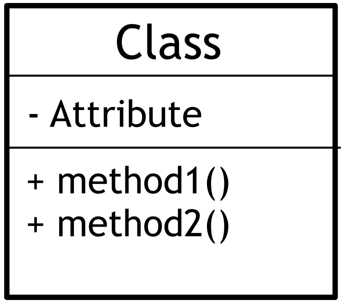
Structural Diagrams

- Class diagrams
  - Representation of the static structure of a software system
  - Description of logical relations between structural elements
  - No activity or control logic

- Object diagrams
  - Instances of a class diagram
  - „Snapshot“ of a system during runtime

UML Class

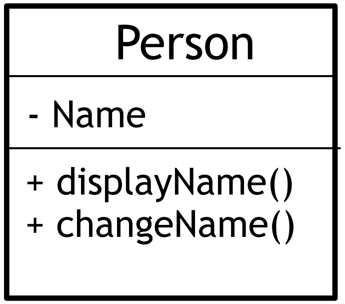
- Classes are represented by rectangles, which include the name of the class, its attributes and methods.
- The class name is in singular and starts with an upper case letter.
- Attributes and methods are separated by horizontal lines.
- „+/-“: Attribute/Method is public/private



Class

- Attribute

+ method1()
+ method2()



Person

- Name

+ displayName()
+ changeName()

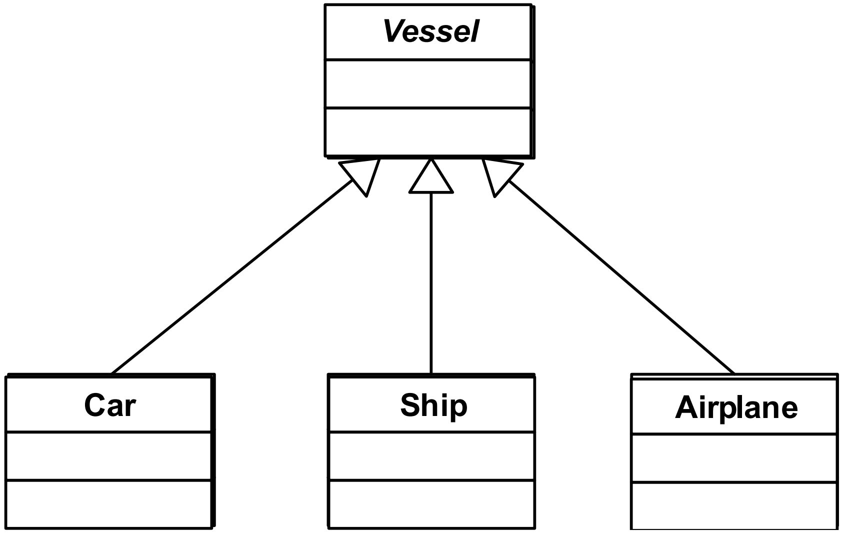
UML Class

- Class attributes
  - Class attributes belong to the class, not to the object.
  - Class attributes have the same value for all instances (objects). For instance, attribute „Number“ to count the number of created objects for a class.
  - Class attributes are underlined in the class diagram.

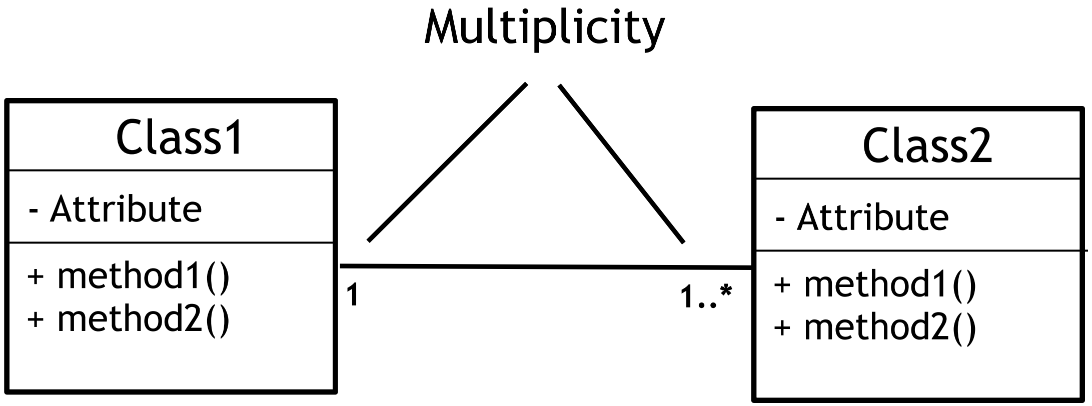
- Class methods
  - Class methods are executed within the class not on the object.
  - E.g. „count number of created objects of the class“
  - The class method is underlined in the class diagram.

Abstract Classes

- Definition / aggregation of common properties
- An abstract class does not allows objects to be instantiated.
- Template to create subclasses
- Abstract methods get “overwritten” by default
- The name of abstract classes is written in italic.



Associations

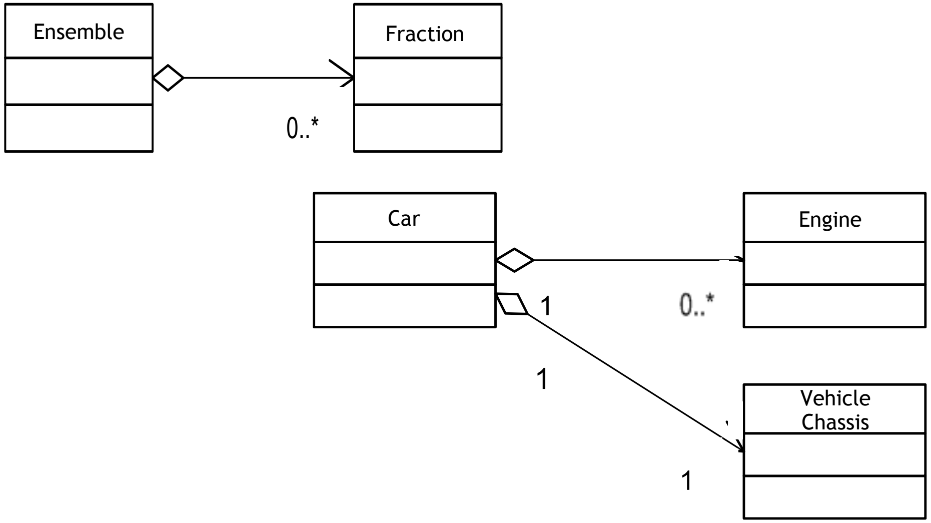


- Describes the relationship between two classes
- It is represented by a line connecting the two classes.
- The multiplicity min..max attached to the association defines the minimal or maximal number of associations between the objects of the two classes.

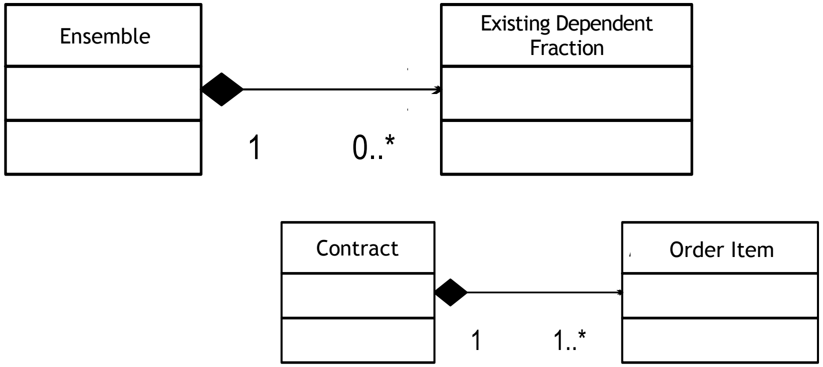
(*) denotes any number of objects.

Associations

- Aggregation
  - Denotes a „has a“ relationship

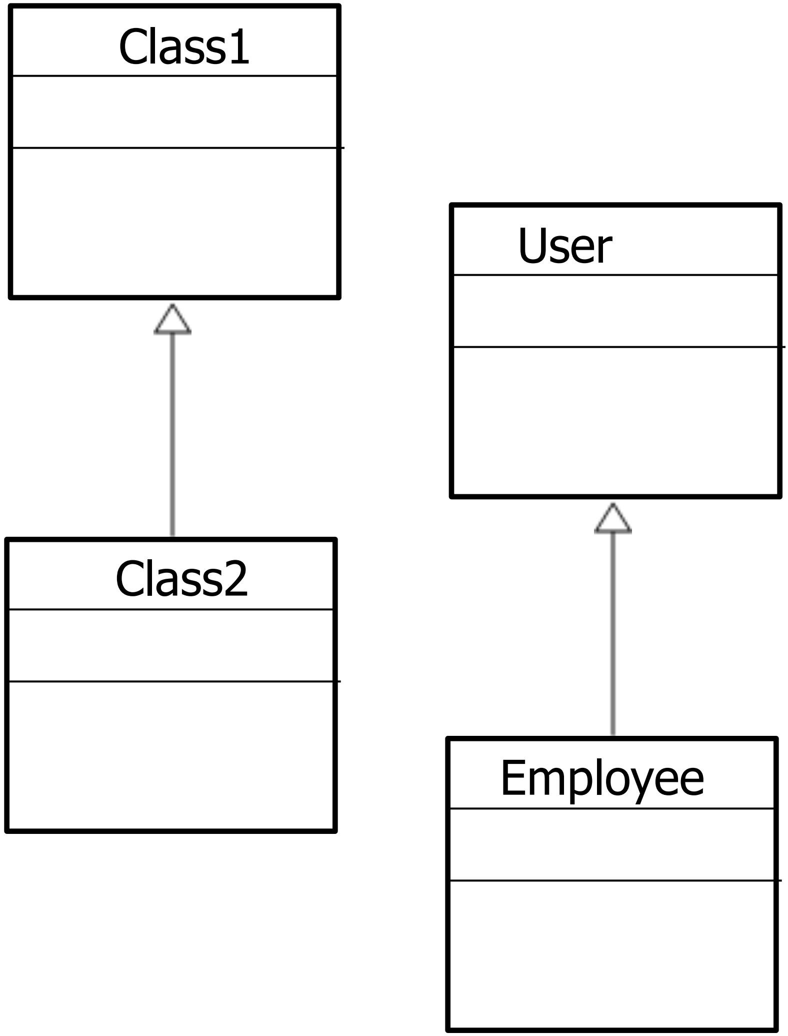


- Composition
  - Composition is a stronger variant of the aggregation
  - Denotes an “owns a” relationship



Inheritance

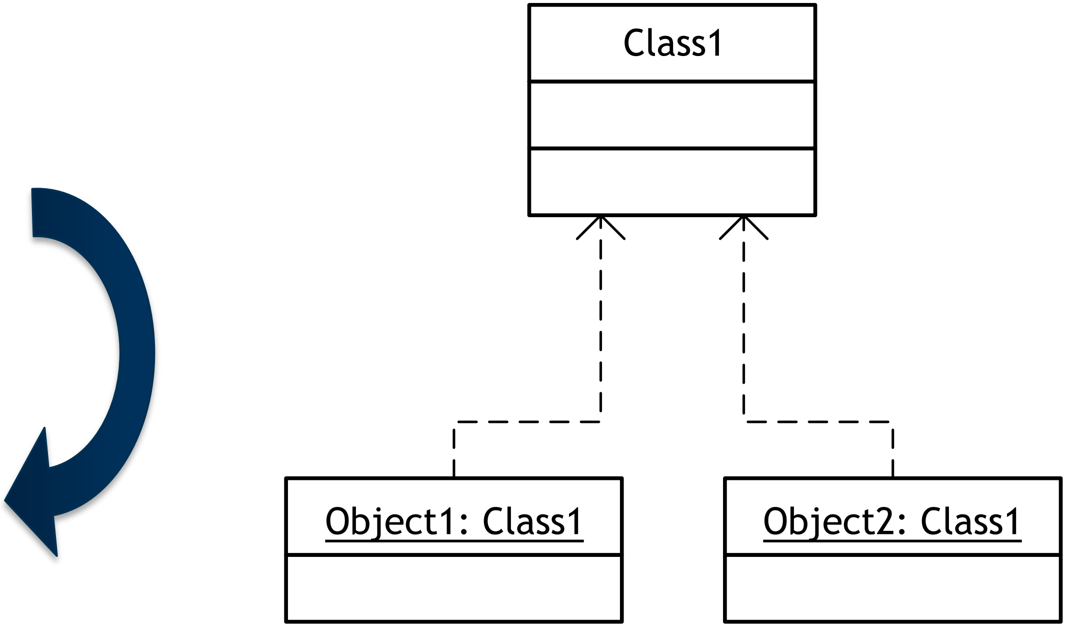
- Denotes an relation between parent class and subclass
- Is represented by a line with an empty arrow at the end, pointing towards the parent class
- Class2 inherits from Class1.
- Purpose:
  - Reuse code, by objects which can be based on previously created objects



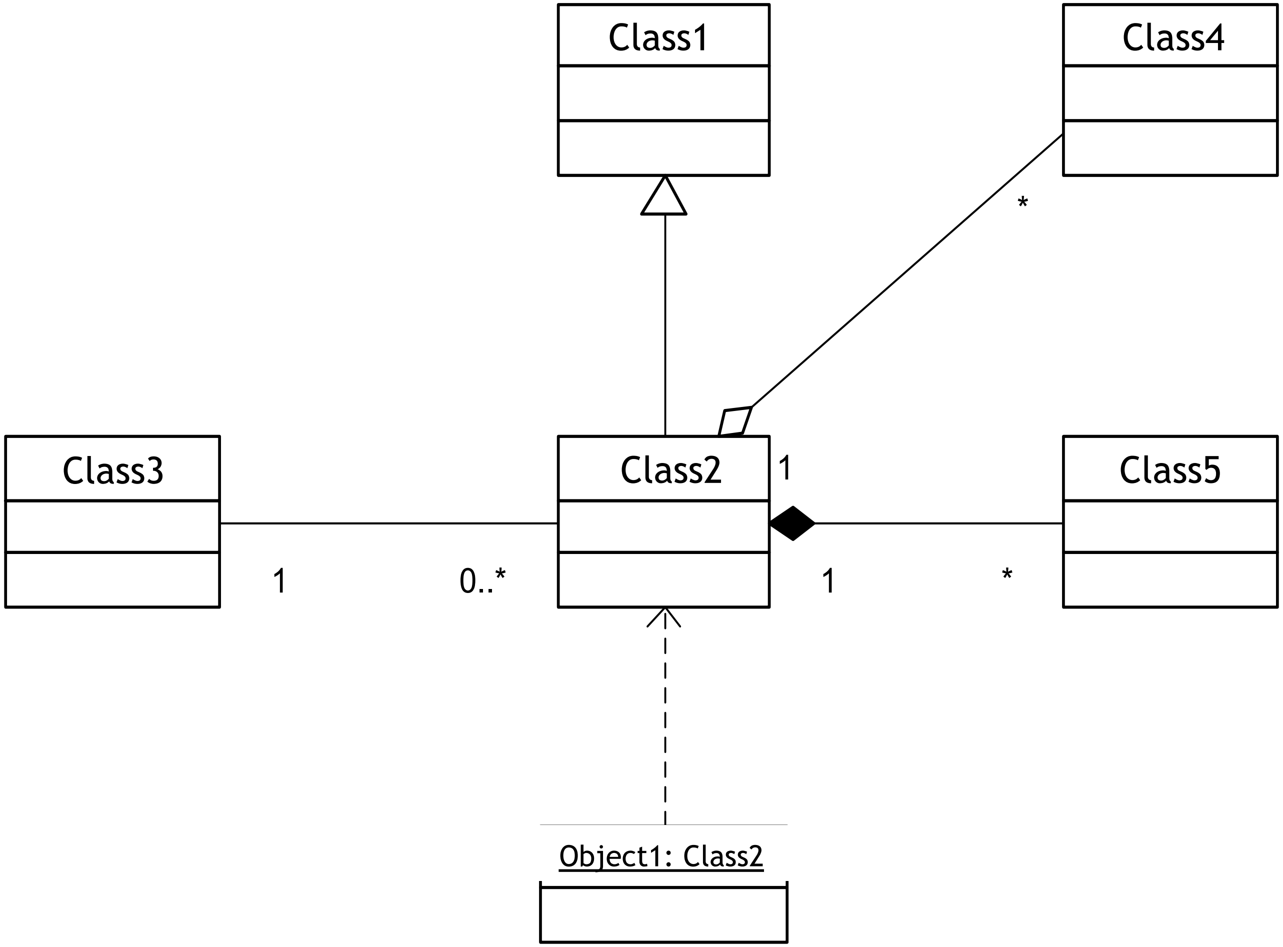
Instantiation

- Representation of the relation “class-object“
- An object is an instance of a class.

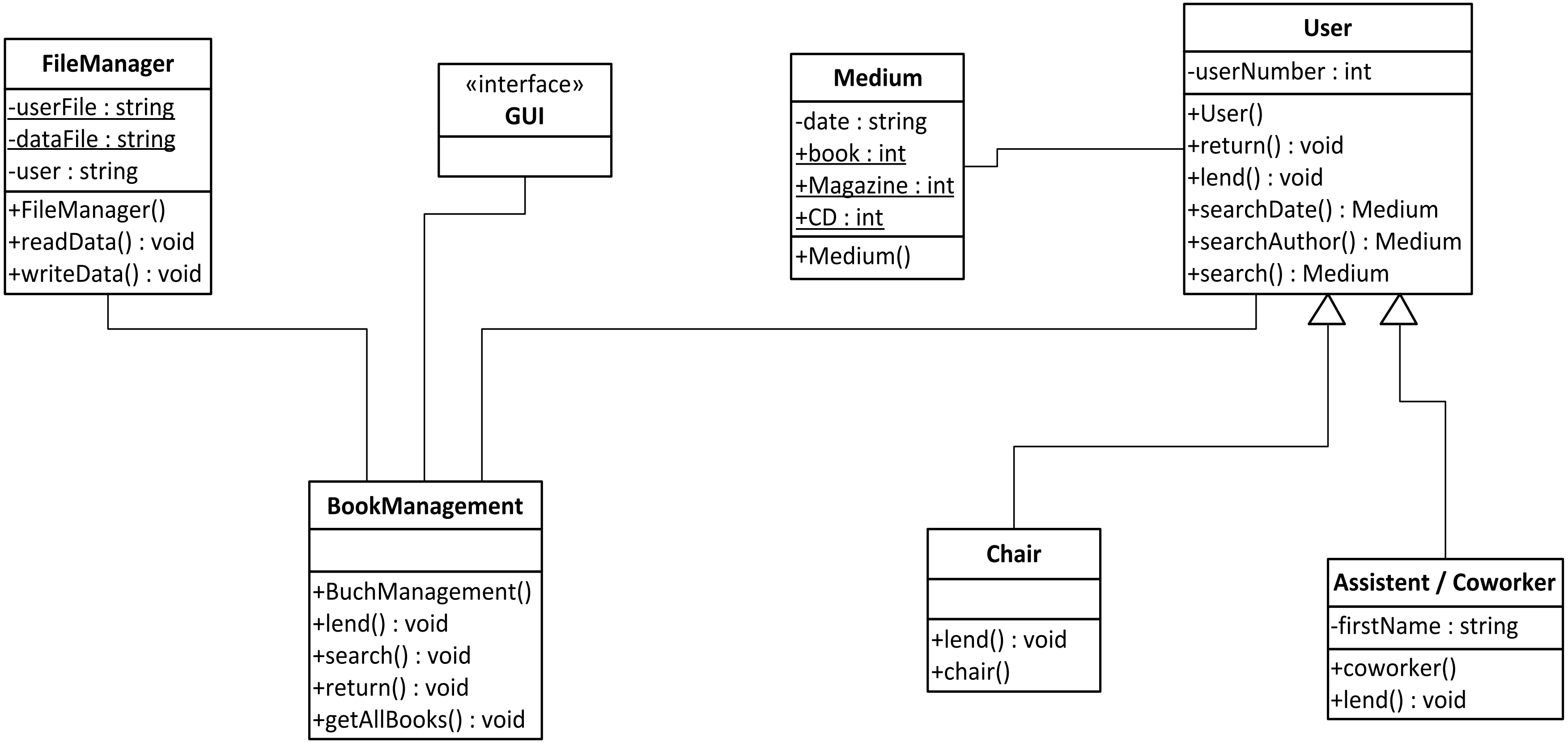
- Class
  - Attributes
  - Methods
- Object
  - Attribute values
  - Messages



Class Diagram



Class Diagram (Example)



FileManager

-userFile : string
-dataFile : string
-user : string

+FileManager()
+readData() : void
+writeData() : void

«interface»
GUI

Medium

-date : string
+book : int
+Magazine : int
+CD : int

+Medium()

User

-userNumber : int

+User()
+return() : void
+lend() : void
+searchDate() : Medium
+searchAuthor() : Medium
+search() : Medium

BookManagement

+BuchManagement()
+lend() : void
+search() : void
+return() : void
+getAllBooks() : void

Chair

+lend() : void
+chair()

Assistent / Coworker

-firstName : string

+coworker()
+lend() : void

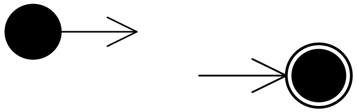
Questions VL9-12

1. **Können Sie ein Beispiel zu einem Split/Join im Aktivitätsdiagramm geben?**

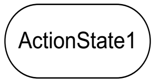
Activity Diagram Notation Elements

Notation elements

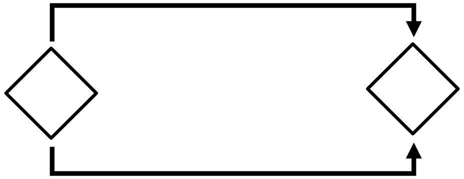
- Initial state/final state



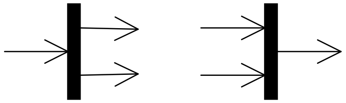
- Activity



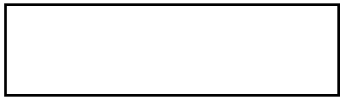
- Decision



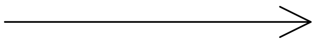
- Split/join



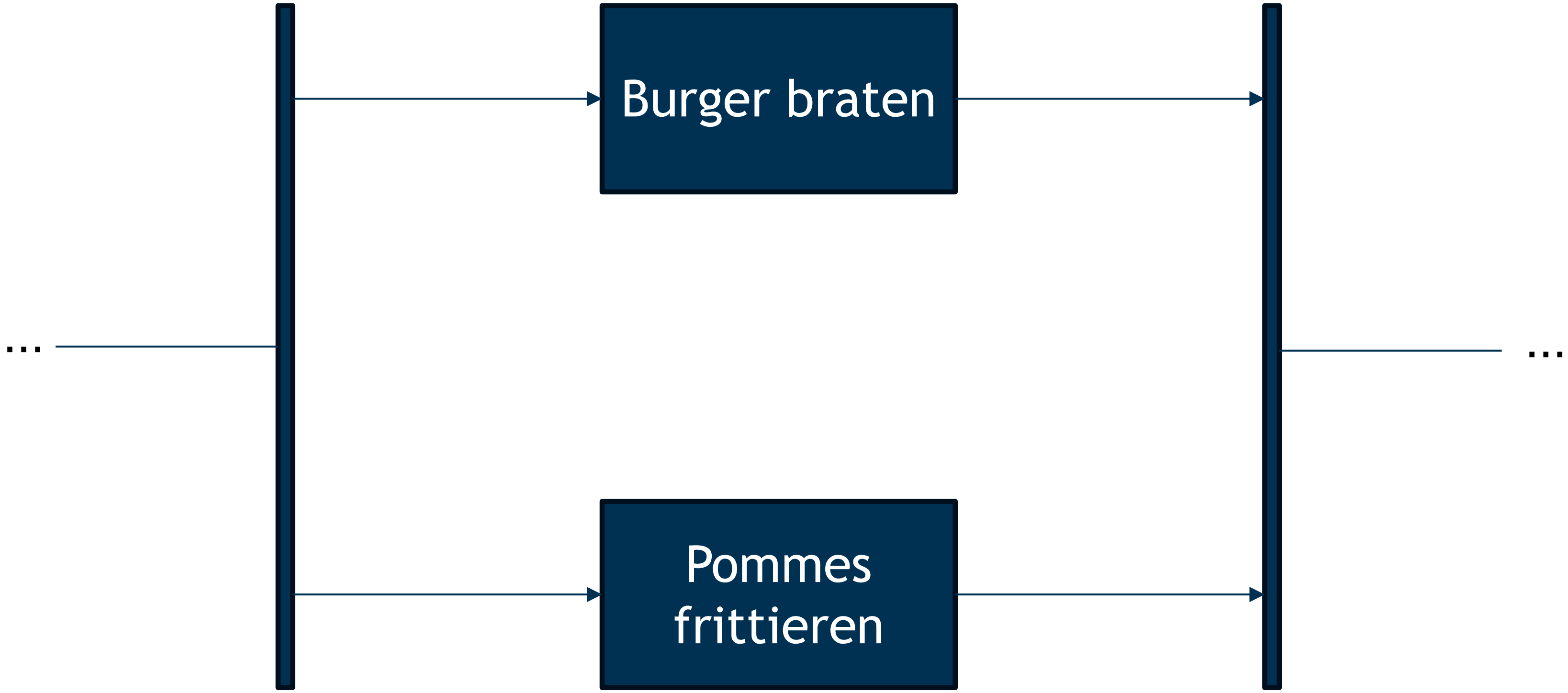
- Responsibility



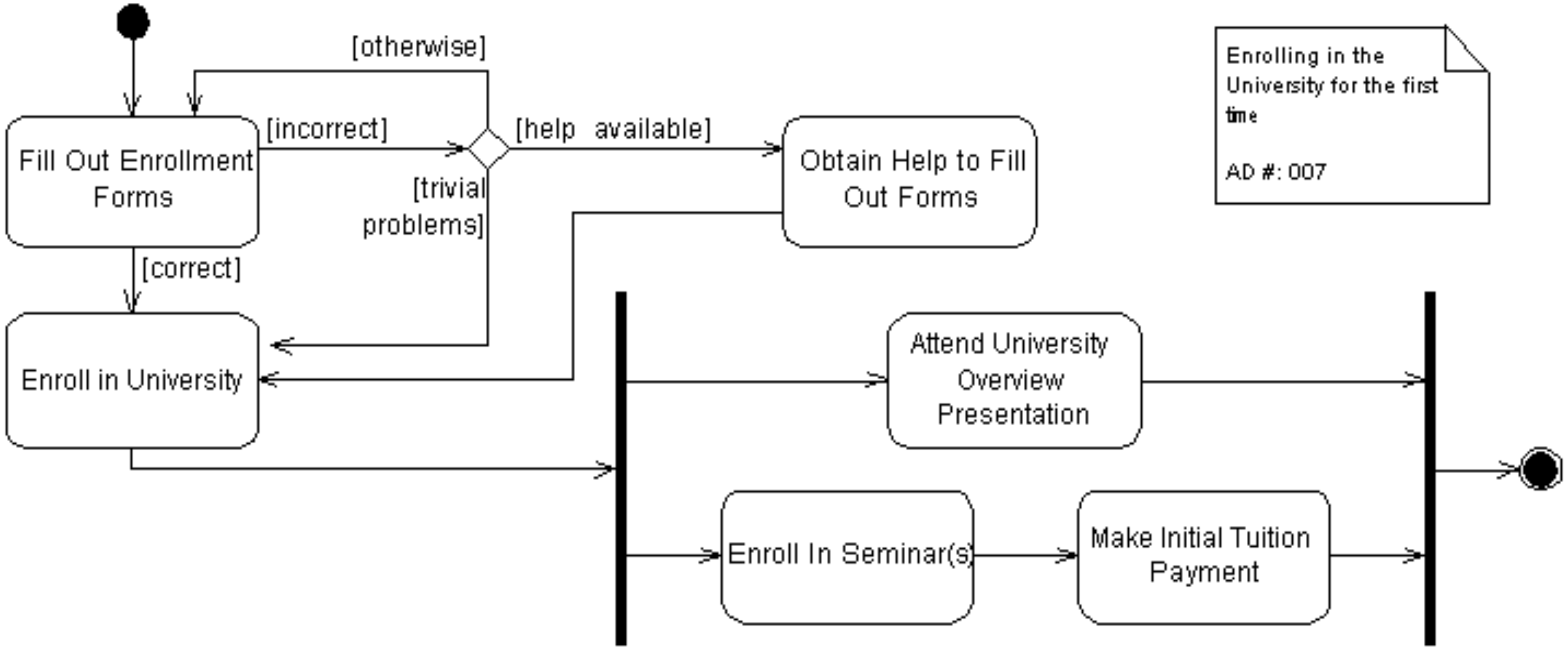
- Activity flow



Beispiel Aktivitätsdiagramm



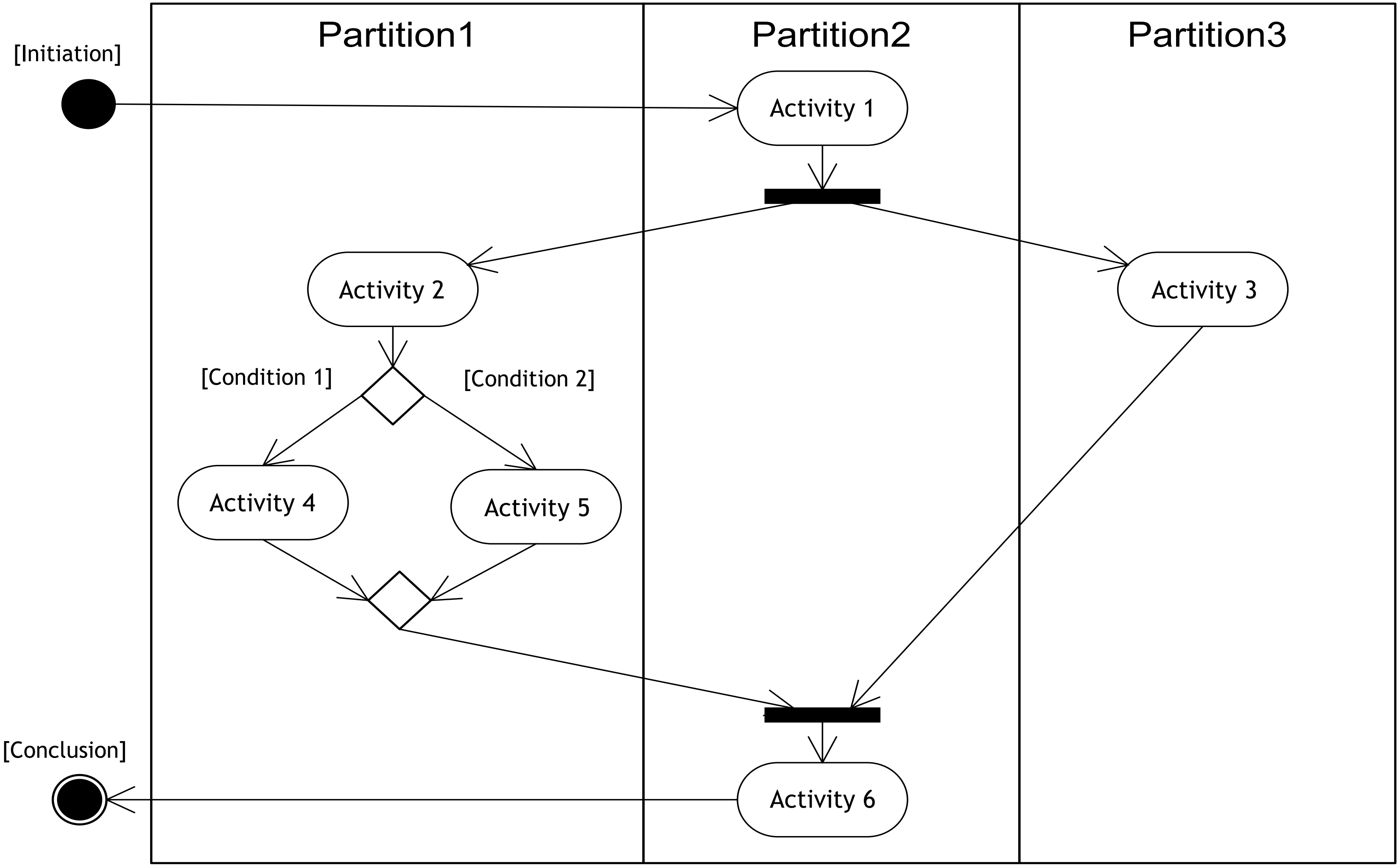
Activity Diagram (Example)


[otherwise]
[incorrect]
[help available]
Fill Out Enrollment Forms
Obtain Help to Fill Out Forms
[trivial problems]
[correct]
Enroll in University
Attend University Overview Presentation
Enroll In Seminar(s)
Make Initial Tuition Payment

Enrolling in the University for the first time

AD #: 007

Activity Diagram

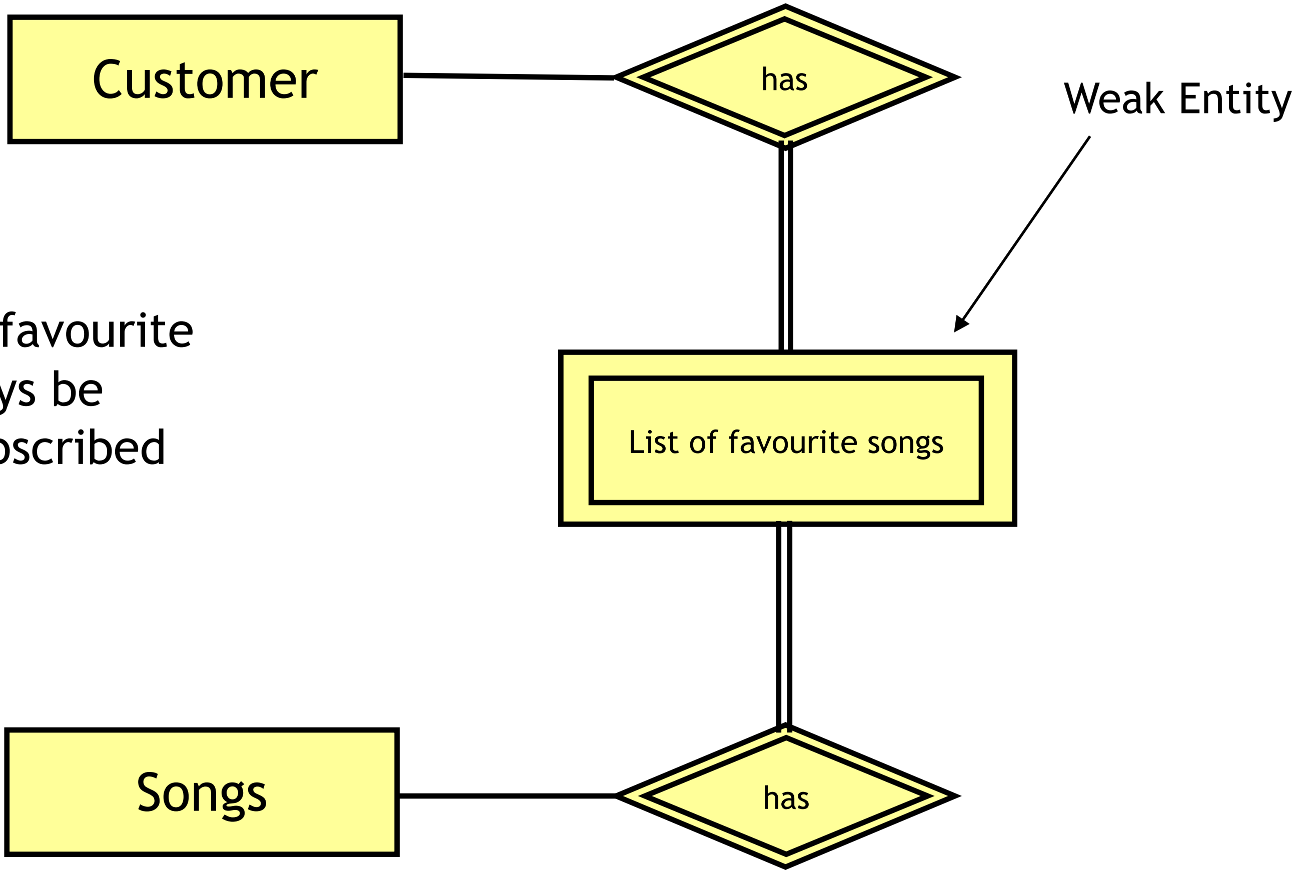


Questions VL9-12

1. **Wenn eine Weak Entity 2 Seiten hat, wie wird die “dominierende” [identifiziernde] Entität gekennzeichnet?**

Weak Entities

Weak entities depend on at least one entity and consequently cannot exist without them.



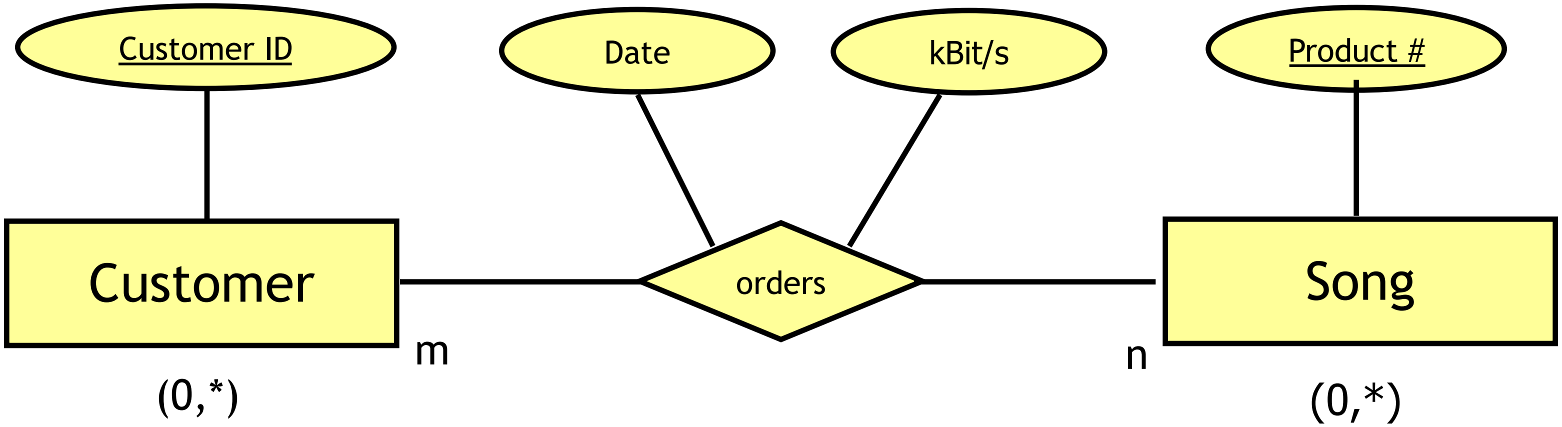
An active list of favourite songs must always be assigned to a subscribed customer.

Example Relationship with Attributes

- Many **Customers** can order many **songs**. **Customers are** uniquely identified by the attribute Customer ID. Each **song** has a unique identifier called **Product #.** The orders relationship has the attributes **date**, recording when the order was placed, and **kBit/s**, indicating the quality or speed of the song's digital file.

Process for the Development of an ER-Model/Diagram

- New example with m customers



Questions VL9-12

1. **Könnten wir noch einmal eine Beispielaufgabe zu DTD und XML durchsprechen?**

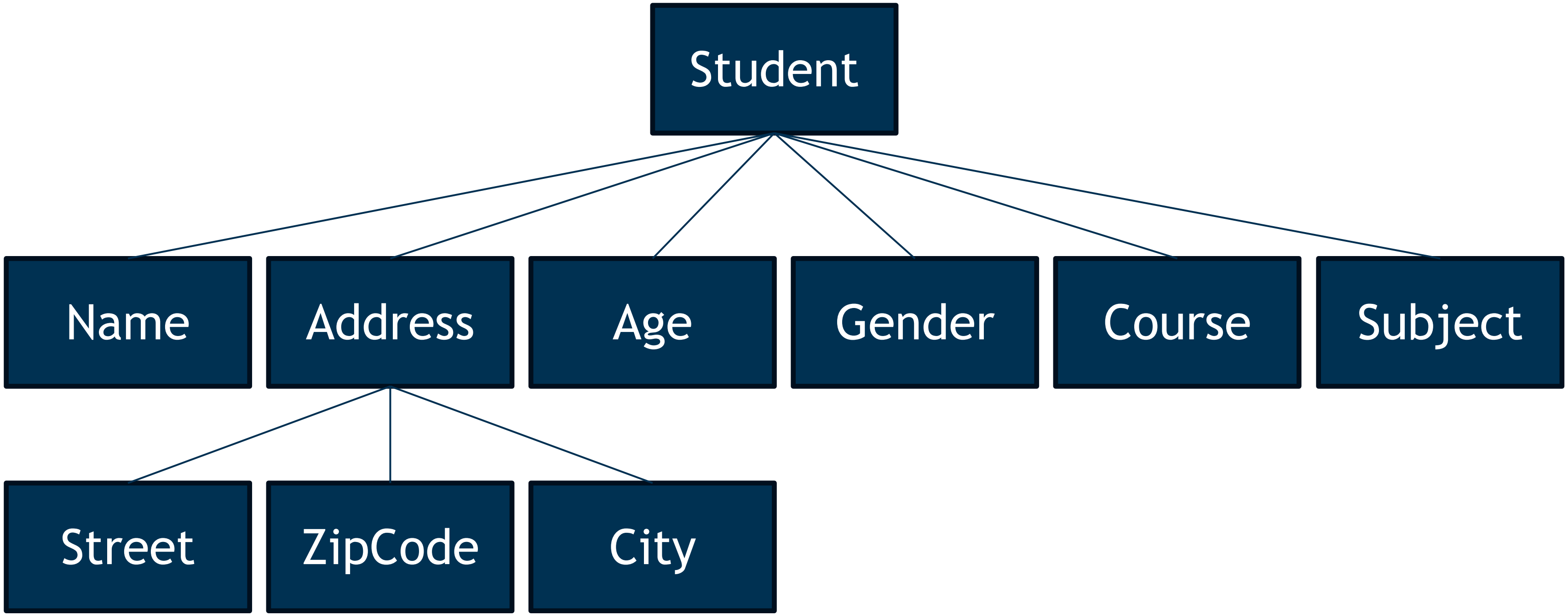
Document Type Definition

a.) Erstellen sie für das nachfolgende XML Dokument ein Baumdiagramm, welches die Hierarchie abbildet.

b.) Erstellen Sie für das rechts genannte XML-Dokument nun ein DTD. Neben Course kann auch ein Subject gewählt werden.

```
<?xml version="1.0"?>
<Student>
   <Name>Guenther M</Name>
   <Address>
       <Street>Cluestreet
       </Street>
       <ZipCode>63743
       </ZipCode>
     <City>Berlin</City>
   </Address>
   <Age>23</Age>
   <Gender>male</Gender>
   <Course>Finance</Course>
   <Course>Economics</Course>
</Student>
```

Hirachy


Student

Name

Address

Age

Gender

Course

Subject

Street

ZipCode

City

Document Type Definition

b.) Erstellen Sie für das rechts genannte XML-Dokument nun ein DTD. Neben Course kann auch ein Subject gewählt werden. Course und Subject können beliebig oft vorkommen.

```
<?xml version="1.0"?>
<Student>
   <Name>Guenther M</Name>
   <Address>
       <Street>Cluestreet
       </Street>
      <ZipCode>63743
      </ZipCode>
     <City>Berlin</City>
   </Address>
   <Age>23</Age>
   <Gender>male</Gender>
   <Course>Finance</Course>
   <Course>Economics</Course>
</Student>
```

Document Type Definition

```
<?xml version="1.0"?>
<Student>
   <Name>Guenther M</Name>
   <Address>
     <Street>Cluestreet
     </Street>
     <ZipCode>63743</ZipCode>
     <City>Berlin</City>
   </Address>
   <Age>23</Age>
   <Gender>male</Gender>
   <Course>Scientific
     Work</Course>
   <Course>Economics</Course>
</Student>
```

```
<!DOCTYPE Student [

]>
```

Document Type Definition

```
<?xml version="1.0"?>
<Student>
   <Name>Guenther M</Name>
   <Address>
     <Street>Cluestreet
     </Street>
     <ZipCode>63743</ZipCode>
     <City>Berlin</City>
   </Address>
   <Age>23</Age>
   <Gender>male</Gender>
   <Course>Scientific
     Work</Course>
   <Course>Economics</Course>
</Student>
```

```
<!DOCTYPE Student [
<!ELEMENT Student  (Name,
Address, Age, Gender,
Course*, Subject*)>











]>
```

Document Type Definition

```
<?xml version="1.0"?>
<Student>
   <Name>Guenther M</Name>
   <Address>
     <Street>Cluestreet
     </Street>
     <ZipCode>63743</ZipCode>
     <City>Berlin</City>
   </Address>
   <Age>23</Age>
   <Gender>male</Gender>
   <Course>Scientific
     Work</Course>
   <Course>Economics</Course>
</Student>
```

```
<!DOCTYPE Student [
<!ELEMENT Student  (Name,
Address, Age, Gender,
Course*, Subject*)>
<!ELEMENT Name (#PCDATA)>









]>
```

mobile business

Document Type Definition

```
<?xml version="1.0"?>
<Student>
   <Name>Guenther M</Name>
   <Address>
     <Street>Cluestreet
     </Street>
     <ZipCode>63743</ZipCode>
     <City>Berlin</City>
   </Address>
   <Age>23</Age>
   <Gender>male</Gender>
   <Course>Scientific
     Work</Course>
   <Course>Economics</Course>
</Student>
```

```
<!DOCTYPE Student [
<!ELEMENT Student  (Name,
Address, Age, Gender,
Course*, Subject*)>
<!ELEMENT Name (#PCDATA)>
<!ELEMENT Address (Street,
ZipCode, City)>







]>
```

Exercise 2 b): Document Type Definition

```
<?xml version="1.0"?>
<Student>
   <Name>Guenther M</Name>
   <Address>
     <Street>Cluestreet
     </Street>
     <ZipCode>63743</ZipCode>
     <City>Berlin</City>
   </Address>
   <Age>23</Age>
   <Gender>male</Gender>
   <Course>Scientific
     Work</Course>
   <Course>Economics</Course>
</Student>
```

```
<!DOCTYPE Student [
<!ELEMENT Student  (Name,
Address, Age, Gender,
Course*, Subject*)>
<!ELEMENT Name (#PCDATA)>
<!ELEMENT Address (Street,
ZipCode, City)>
<!ELEMENT Street (#PCDATA)>
<!ELEMENT ZipCode (#PCDATA)>
<!ELEMENT City (#PCDATA)>
<!ELEMENT Age (#PCDATA)>
<!ELEMENT Gender (#PCDATA)>
<!ELEMENT Course (#PCDATA)>
<!ELEMENT Subject (#PCDATA)>
]>
```

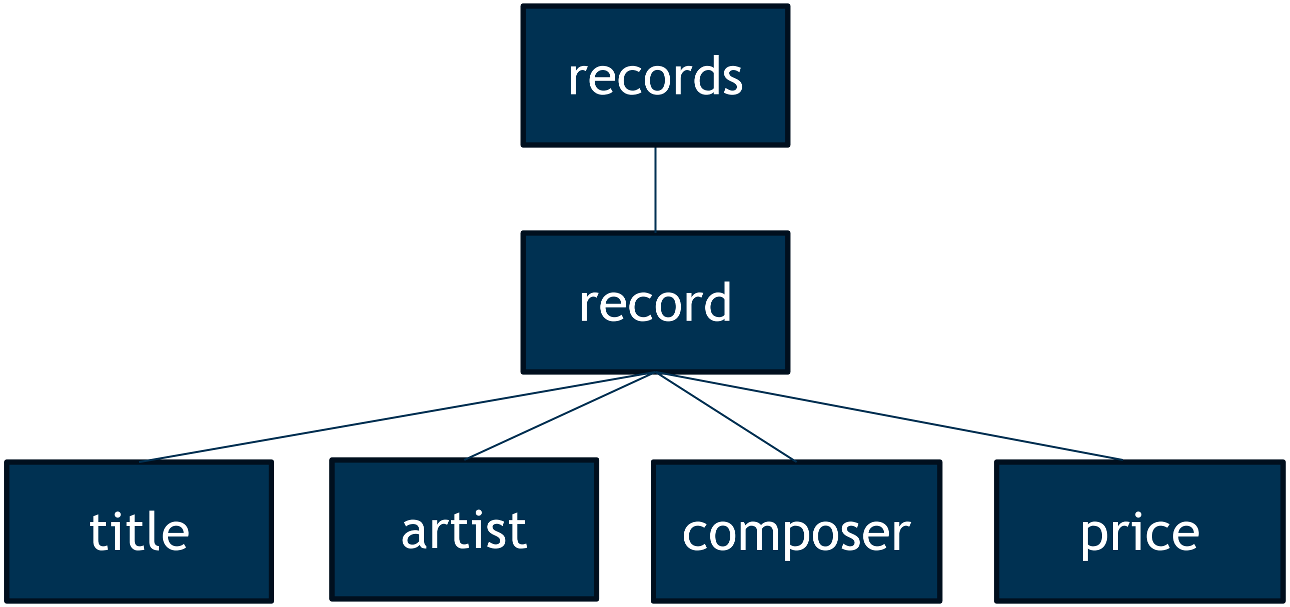
mobile business

DTD → XML

Erstellen Sie ein beispielhaftes XML-Dokument auf der Grundlage der folgenden DTD-Datei. Das <records>-Element darf nicht leer sein leer sein. Zeichnen sie zuerst die Hirachie des Dokumentes.

```
<!DOCTYPE records [
<!ELEMENT records (record*)>
<!ELEMENT record (title,
artist, composer?, price)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT artist (#PCDATA)>
<!ELEMENT composer (#PCDATA)>
<!ELEMENT price (#PCDATA)>
]>
```

Hirachy



records

record

title

artist

composer

price

DTD → XML

```
<?xml version="1.0"?>
```

```
<!DOCTYPE records [
<!ELEMENT records (record*)>
<!ELEMENT record (title,
artist, composer?, price)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT artist (#PCDATA)>
<!ELEMENT composer (#PCDATA)>
<!ELEMENT price (#PCDATA)>
]>
```

DTD → XML

```
<?xml version="1.0"?>
<records>












</records>
```

```
DOCTYPE records [
ELEMENT records (record*)>
ELEMENT record (title,
tist, composer?, price)>
ELEMENT title (#PCDATA)>
ELEMENT artist (#PCDATA)>
ELEMENT composer (#PCDATA)>
ELEMENT price (#PCDATA)>
```

DTD → XML

```
<?xml version="1.0"?>
<records>
   <record>




   </record>
   <record>







   </record>
</records>
```

```
<!DOCTYPE records [
<!ELEMENT records (record*)>
<!ELEMENT record (title,
artist, composer?, price)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT artist (#PCDATA)>
<!ELEMENT composer (#PCDATA)>
<!ELEMENT price (#PCDATA)>
]>
```

DTD → XML

```
<?xml version="1.0"?>
<records>
   <record>
      <title>Help!</title>


   </record>
   <record>
      <title>The very best
      of</title>




   </record>
</records>
```

```
<!DOCTYPE records [
<!ELEMENT records (record*)>
<!ELEMENT record (title,
artist, composer?, price)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT artist (#PCDATA)>
<!ELEMENT composer (#PCDATA)>
<!ELEMENT price (#PCDATA)>
]>
```

DTD → XML

```xml
<?xml version="1.0"?>
<records>
   <record>
      <title>Help!</title>
      <artist>The Beatles</artist>

   </record>
   <record>
      <title>The very best
      of</title>
      <artist>Lang Lang</artist>



   </record>
</records>
```

```dtd
<!DOCTYPE records [
<!ELEMENT records (record*)>
<!ELEMENT record (title,
artist, composer?, price)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT artist (#PCDATA)>
<!ELEMENT composer (#PCDATA)>
<!ELEMENT price (#PCDATA)>
]>
```

DTD → XML

```
<?xml version="1.0"?>
<records>
   <record>
      <title>Help!</title>
      <artist>The Beatles</artist>

   </record>
   <record>
      <title>The very best
      of</title>
      <artist>Lang Lang</artist>
      <composer>Bach</composer>

   </record>
</records>
```

```
<!DOCTYPE records [
<!ELEMENT records (record*)>
<!ELEMENT record (title,
artist, composer?, price)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT artist (#PCDATA)>
<!ELEMENT composer (#PCDATA)>
<!ELEMENT price (#PCDATA)>
]>
```

DTD → XML

```
<?xml version="1.0"?>
<records>
   <record>
      <title>Help!</title>
      <artist>The Beatles</artist>
      <price>9.95</price>
   </record>
   <record>
      <title>The very best
      of</title>
      <artist>Lang Lang</artist>
      <composer>Bach</composer>
      <price>19.90</price>
   </record>
</records>
```

```
<!DOCTYPE records [
<!ELEMENT records (record*)>
<!ELEMENT record (title,
artist, composer?, price)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT artist (#PCDATA)>
<!ELEMENT composer (#PCDATA)>
<!ELEMENT price (#PCDATA)>
]>
```

Fragen XML

- **Können Sie ein noch einmal ANY und EMPTY erklären?**
- Darf in der „selection list“ eine Kardinalität innerhalb und außerhalb der Klammern stehen?

XML DTD

- Content (in elements):

EMPTY	Empty element
ANY	Any content
|	Selection list
,	Sequence
()	Grouping
(#PCDATA)	*Parsed Character Data* (mixed data)

- Cardinalities (for elements):

	empty: exactly one value is necessary
+	At least one value
?	None or one value
*	None or multiple values

Beispiel ANY

- **ANY:** Dieses Element kann jeden beliebigen Inhalt haben kann, einschließlich Text, Zahlen, andere Elemente oder eine Mischung aus diesen. Elemente müssen im DTD definiert sein.

-
```
<!ELEMENT element-name ANY>
Example:

<!ELEMENT note ANY>
```

Beispiel ANY

- **EMPTY:** Ein Element dar keinen Inhalt haben. Dies bedeutet, dass das Element geschlossen sein muss also keine anderen Elemente oder Text enthalten kann.
-
  ```
  <!ELEMENT element-name EMPTY>

  XML example Zeilenumbruch:

  <br/>
  ```

Fragen XML

Können Sie ein noch einmal ANY und EMPTY erklären?

Fragen XML

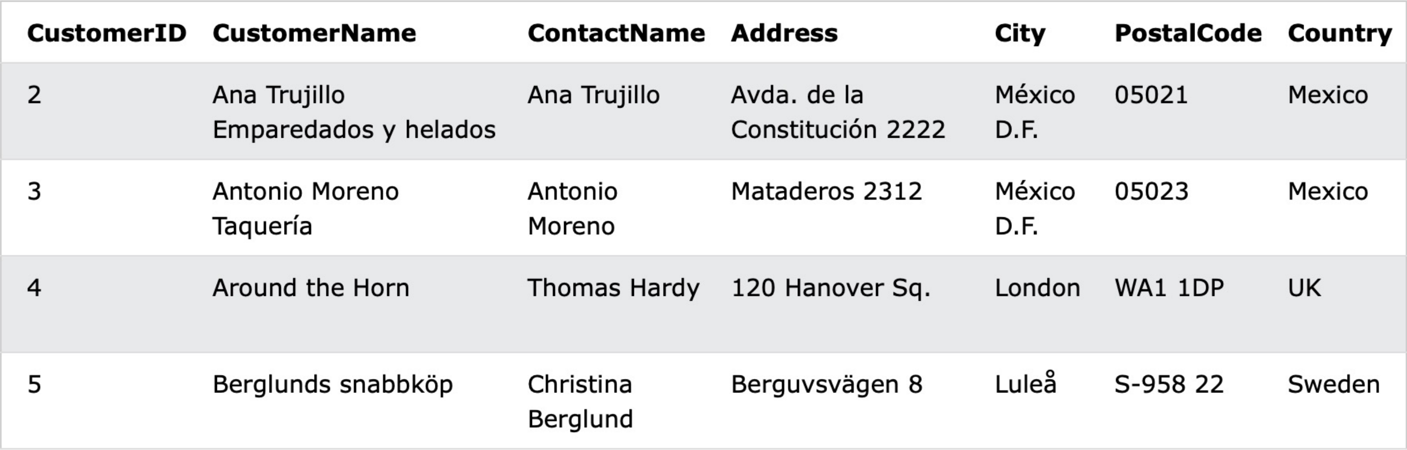
- Können Sie ein noch einmal ANY und EMPTY erklären?
- **Darf in der „selection list“ eine Kardinalität innerhalb und außerhalb der Klammern stehen?**

Exkurs Determinismus Regel:

- „It is required that content models in element type declarations be deterministic“
- <!ELEMENT test (a* | b | c)*>
- When an XML parser reads b, it cannot determine whether this is the end of a* or the beginning of a new b element without waiting for further information. This violates the rule of the deterministic content model.
- -> For PWIN: Choice inside XOR outside!

Questions VL9-12

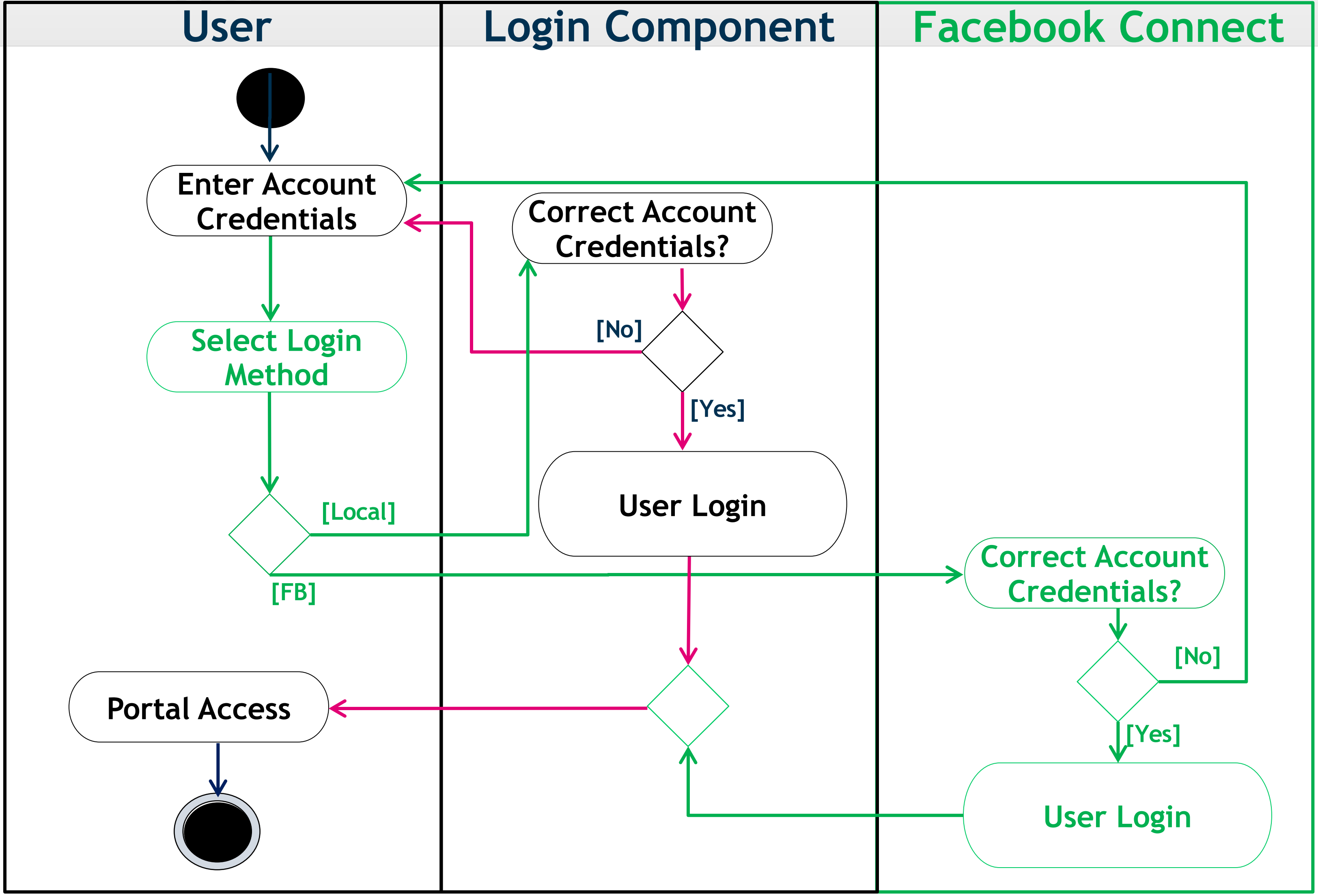
Wenn ich in SQL mehrere Customers löschen möchte benutze ich AND order ‘,’?



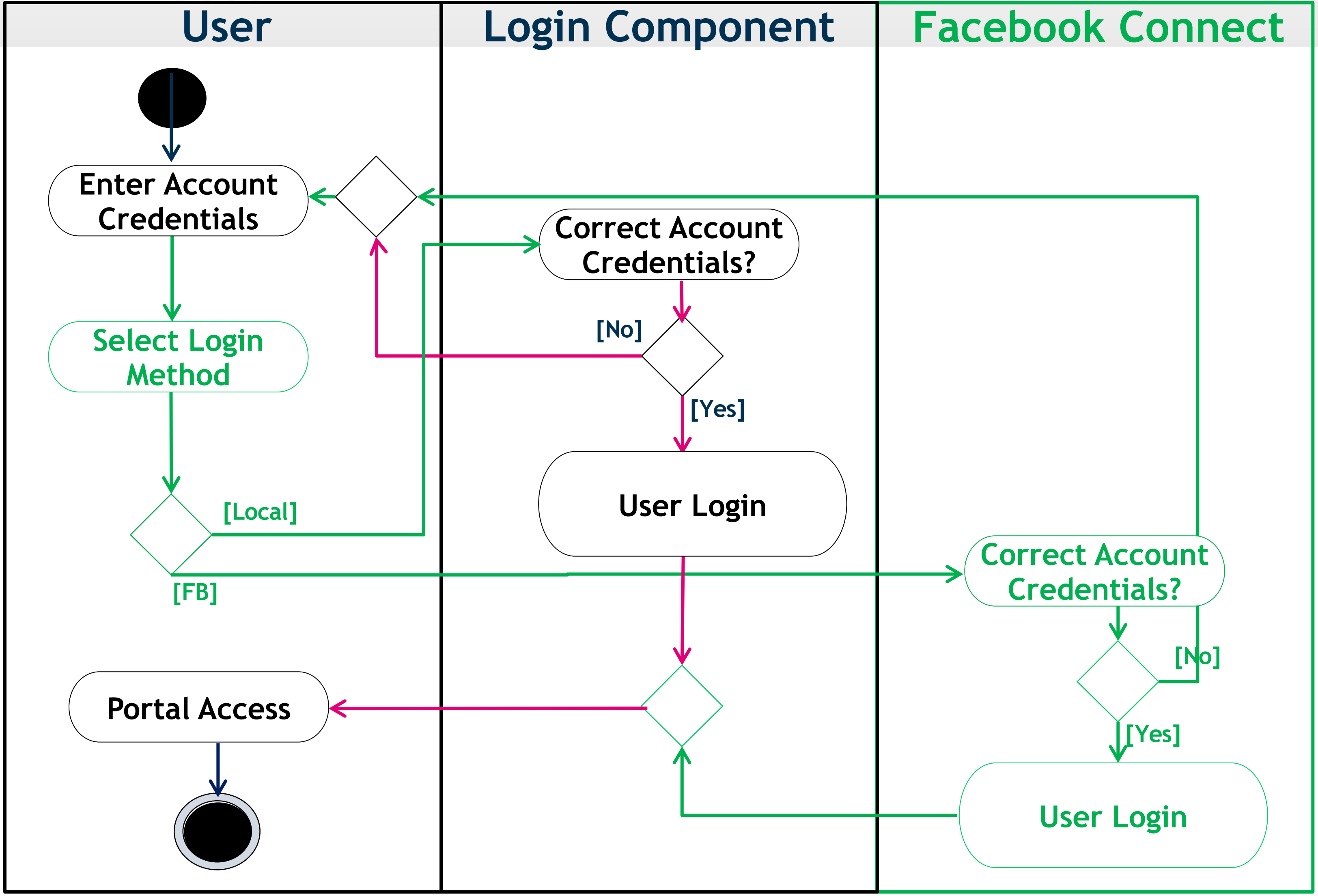
CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden

```
DELETE FROM Customers
WHERE CustomerID = 2 OR 3
```

Fehlt in M05 eine Raute?



Fehlt in M05 eine Raute?


User
Login Component
Facebook Connect
Enter Account Credentials
Select Login Method
[Local]
[FB]
Correct Account Credentials?
[No]
[Yes]
User Login
Correct Account Credentials?
[No]
[Yes]
User Login
Portal Access