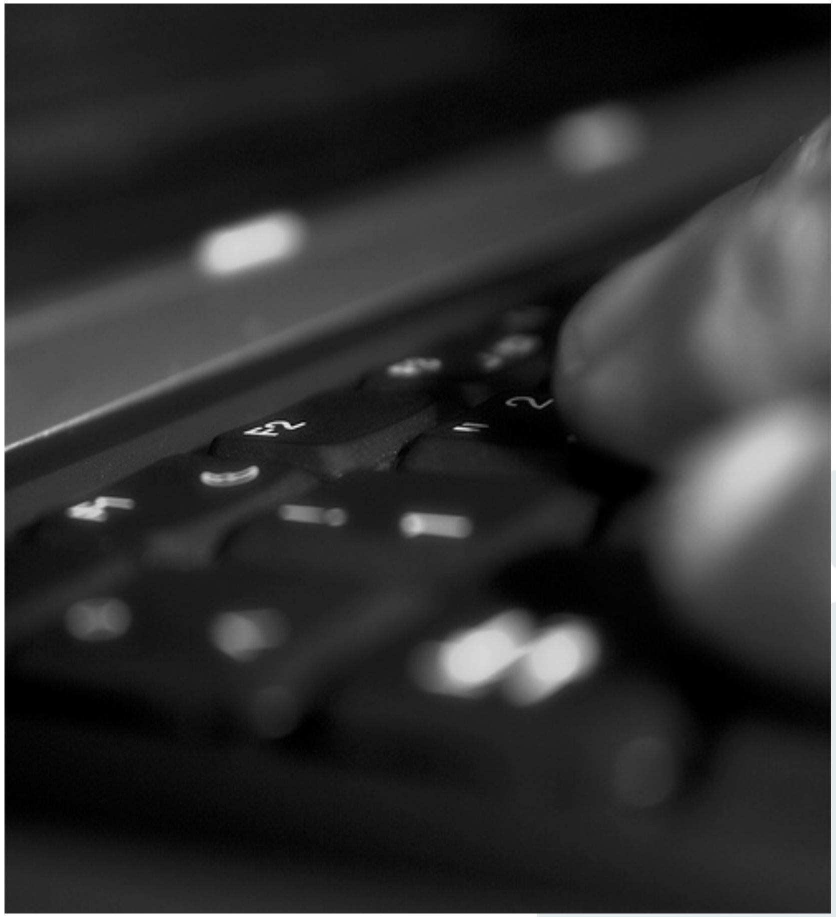
Lecture 14
Business Informatics 2 (PWIN)

Q&A

SS 2017

Prof. Dr. Kai Rannenberg
Akos Grosz, M.Sc.
Christopher Schmitz, M.Sc.
www.m-chair.de


Jenser (Flickr.com)

Welche Themen sind nicht klausurrelevant?

Ausgeschlossene Themen

Nicht klausurrelevante Themen:

- Vorlesung 13: Business Process Reengineering
- XML Example Applications in Vorlesung 10, S.40 ff.
- 1. Gastvorlesung der KfW Bankengruppe

Organisatorisches

- Wird die Klausur auf Deutsch oder Englisch gestellt? Und in welcher Sprache darf ich antworten?

Sprache

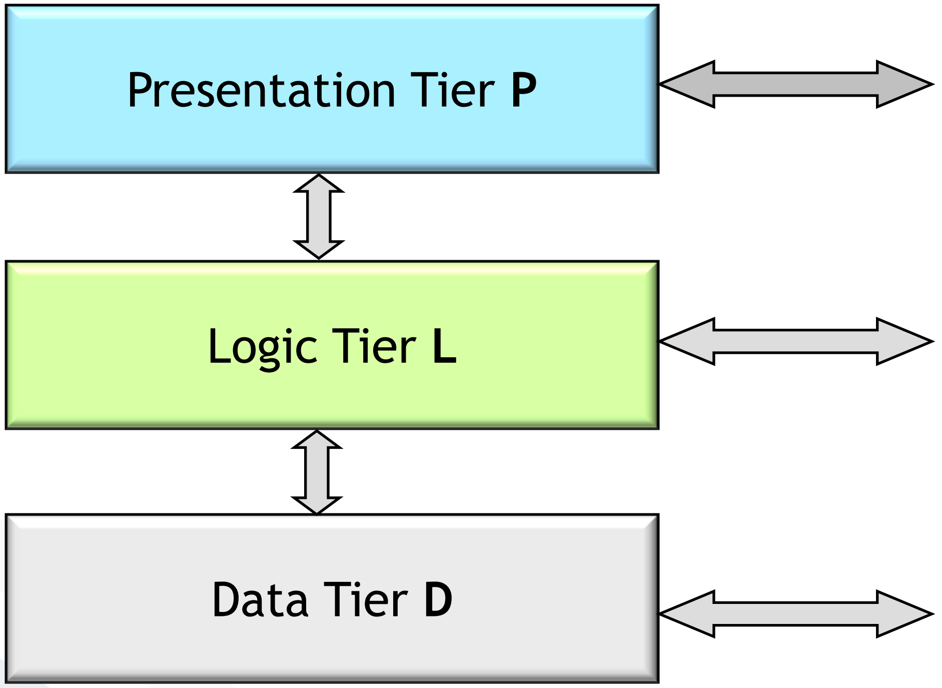
- Die Aufgaben werden auf Deutsch gestellt.
- Sie dürfen auf Deutsch und Englisch antworten.

IS Architekturkonzepte, VL 3

- Können Sie bitte noch einmal kurz auf die genauen Unterscheide zwischen dem Three-Tier- und dem MVC-Konzept eingehen?

Structuring IS Architectures: Three-Tier Concept

VL3F8



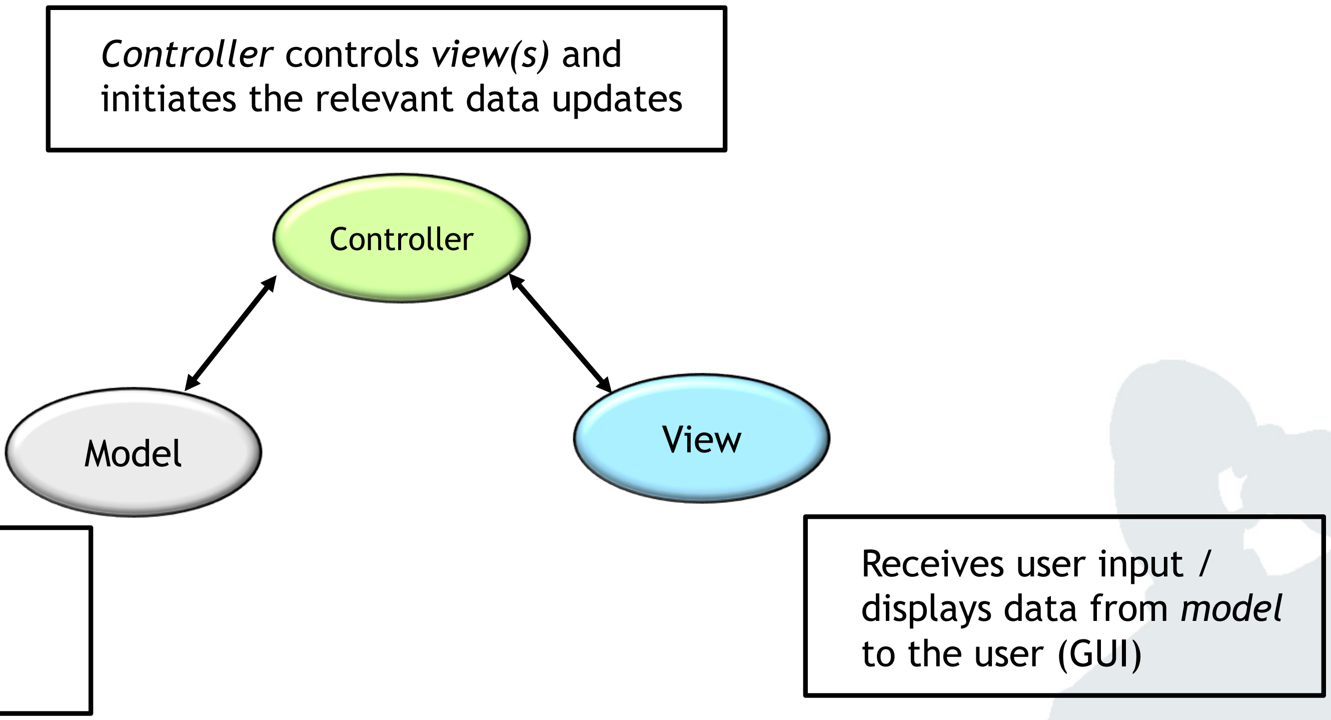
Receives user input / display of data to the user (GUI)

Contains all the program logic

Stores and manage the data

Structuring IS Architectures: Model-View-Controller Concept

VL3F11


Controller controls *view(s)* and initiates the relevant data updates
Controller
Model
View
Manages data and, if applicable, contains the program logic
Receives user input / displays data from *model* to the user (GUI)

Summary on Three-Tier and MVC Concept

VL3F13

- Similar concepts for structuring IS architectures
- Neither one of the concepts is universally defined or specified, e.g.
  - Two-tier concepts are also in existence (Tier Architecture)
  - Program logic resides sometimes in the *model* and other times in the *controller* (MVC Architecture)
- **In conclusion:**

  Independent of the underlying structural models for IS architectures, make sure to modularise certain categories of functionality in an IS.

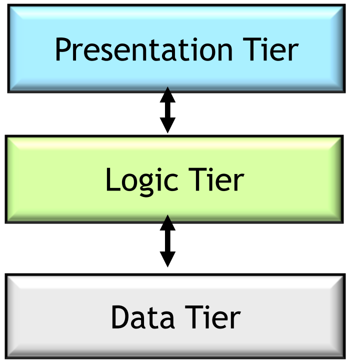
IS Architekturkonzepte, VL 3

▪Wie stehen die verschiedenen Architekturkonzepte zueinander? In welchem Verhältnis? (3-Schichten Architektur, MVC, Client Server, *Osi Modell* etc.)

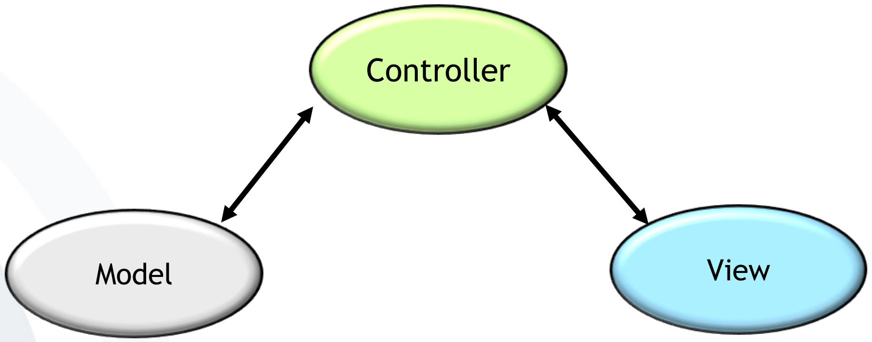
Two Common Structural Models for IS Architectures

VL3F7

- Three-Tier Concept



- Model-View-Controller (MVC) Concept



mobile business

Architecture Concepts of Networked IS

VL3F15

- **Central Server Architecture**
  Low-feature terminals (receiver of services) attached to a powerful central computing unit (provider of services)

- **Client / Server Architecture**
  Network of computers, which can take the role of a server (provider of services), a client (receiver of services) or both.

- **Cloud Computing Architecture**
  Network of computers in the role of a client (receiver of services) connected to a “cloud” of computers (provider of services), which act as a single central server

- **Peer-to-Peer Architecture**
  Network of computers holding equal rights (provider / receiver of services)

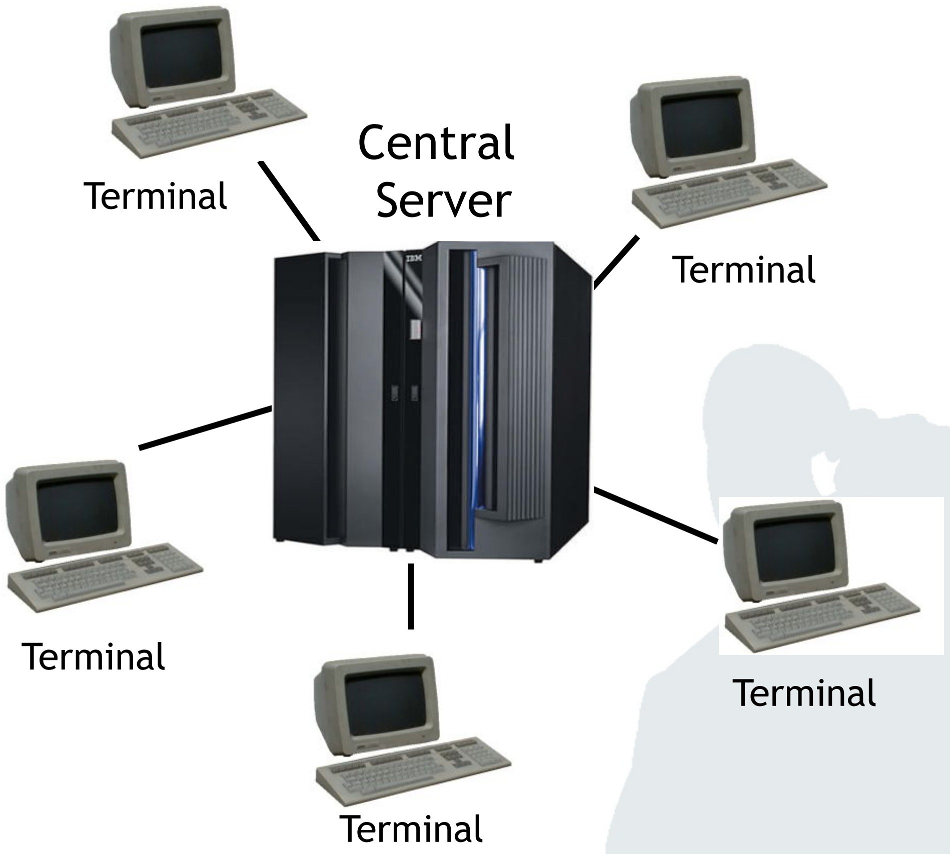
Zentralrechner-Konzept, VL 3

- Was genau versteht man unter Low-Feature-Terminals?

Central Server Architecture

VL3F16

- One powerful Central Computer
- **„Dumb“ low-feature terminals** (often even without hard drive)
- Terminals provide only the graphical user interface (GUI)
- Central Server in charge of processing applications
- Central Server takes care of database and its management



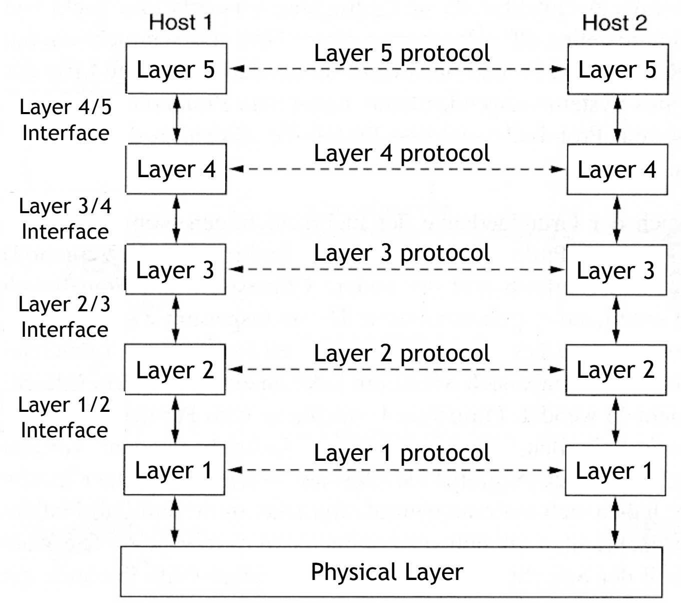
OSI Modell, VL 5

- Regeln Protokolle die Kommunikation innerhalb einer Schicht oder die Kommunikation zwischen den Schichten oder beides?

Protocols

E3F6

- Layers provide specific services to the layer above.
- **Communication inside one layer uses the respective protocol of a layer (i.e. rules and conventions, on which the communication is based).**
- No direct data communication from layer n of one host to the same layer n of another host
- Each layer sends data and control messages to the layer below until the lowermost layer is reached.
- Located below layer 1 is the physical transmission medium which is used for the communication.



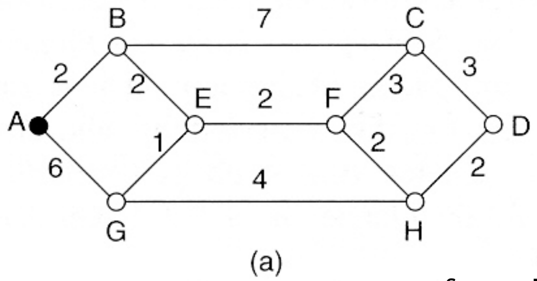
Dijkstra Algorithmus, VL 5

- Könnten Sie im Rahmen der Vorlesung noch mal den Dijkstra Algorithmus erklären?

Layer 3: Network Layer Routing – Dijkstra Algorithm

VL5F27

- The algorithm was developed 1959 by Edsger Wybe Dijkstra.
- It solves the problem of finding the shortest path between two vertices *(singular: vertex)* in a graph.
- For this concept, a graph is created in which every router is represented by a **vertex** and every transmission line by an **edge**.
- The algorithm computes the shortest path between a selected pair of (two) routers with the help of this graph.
- The labels of the **edges** can e.g. be distance, bandwidth, average traffic, transmission costs, average queue length, average transmission time measured or other factors.
- Every **weighted edge** has an impact on the shortest path.

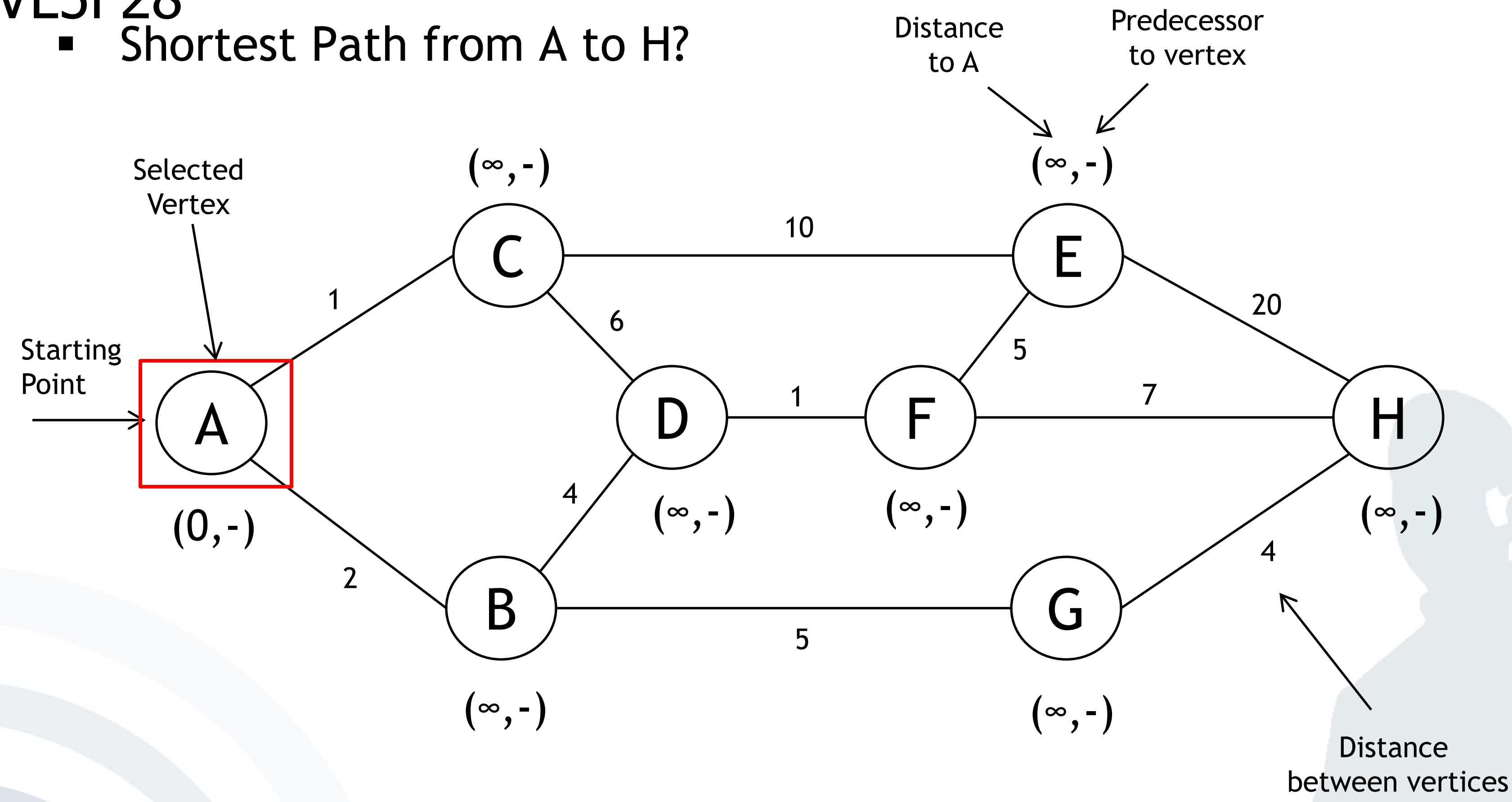


Source: Tanenbaum (2006), p. 391-393

Layer 3: Network Layer
Using Dijkstra Algorithm

VL5F28

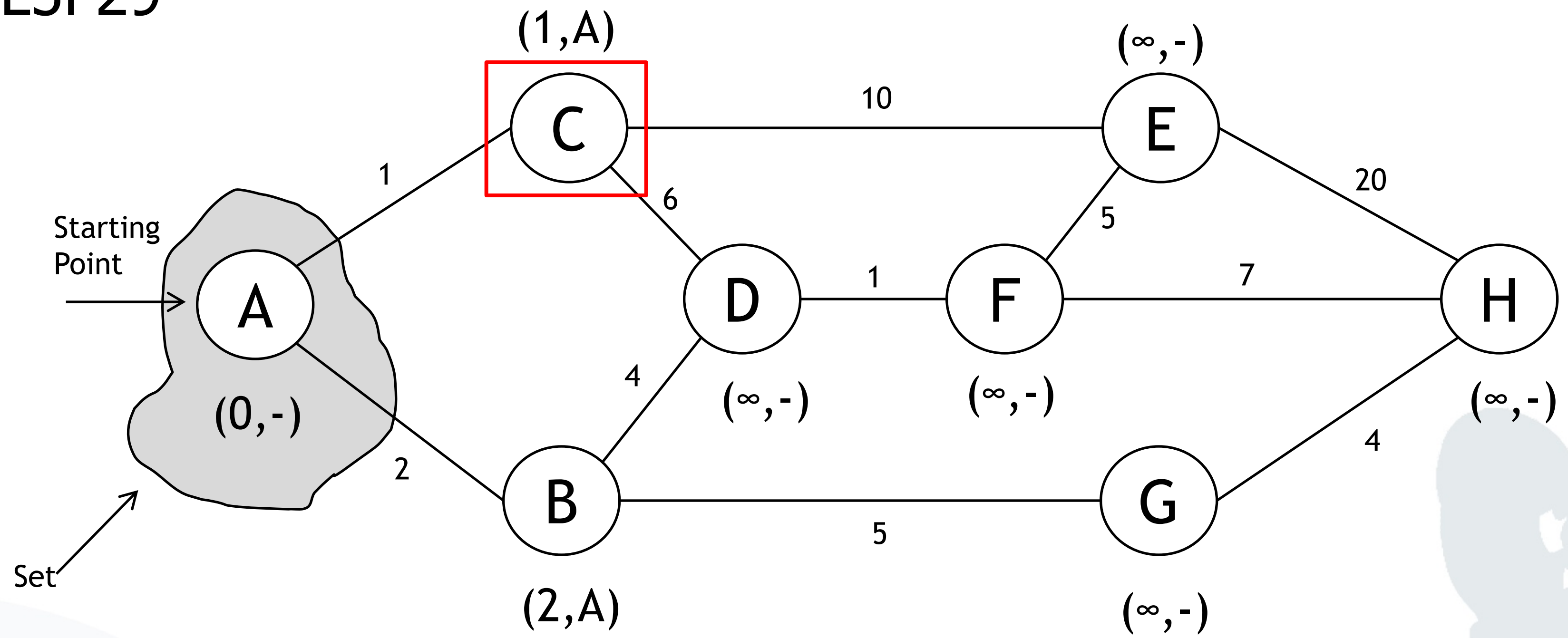
- Shortest Path from A to H?



- Initial State of Graph

Layer 3: Network Layer Using Dijkstra Algorithm (2)

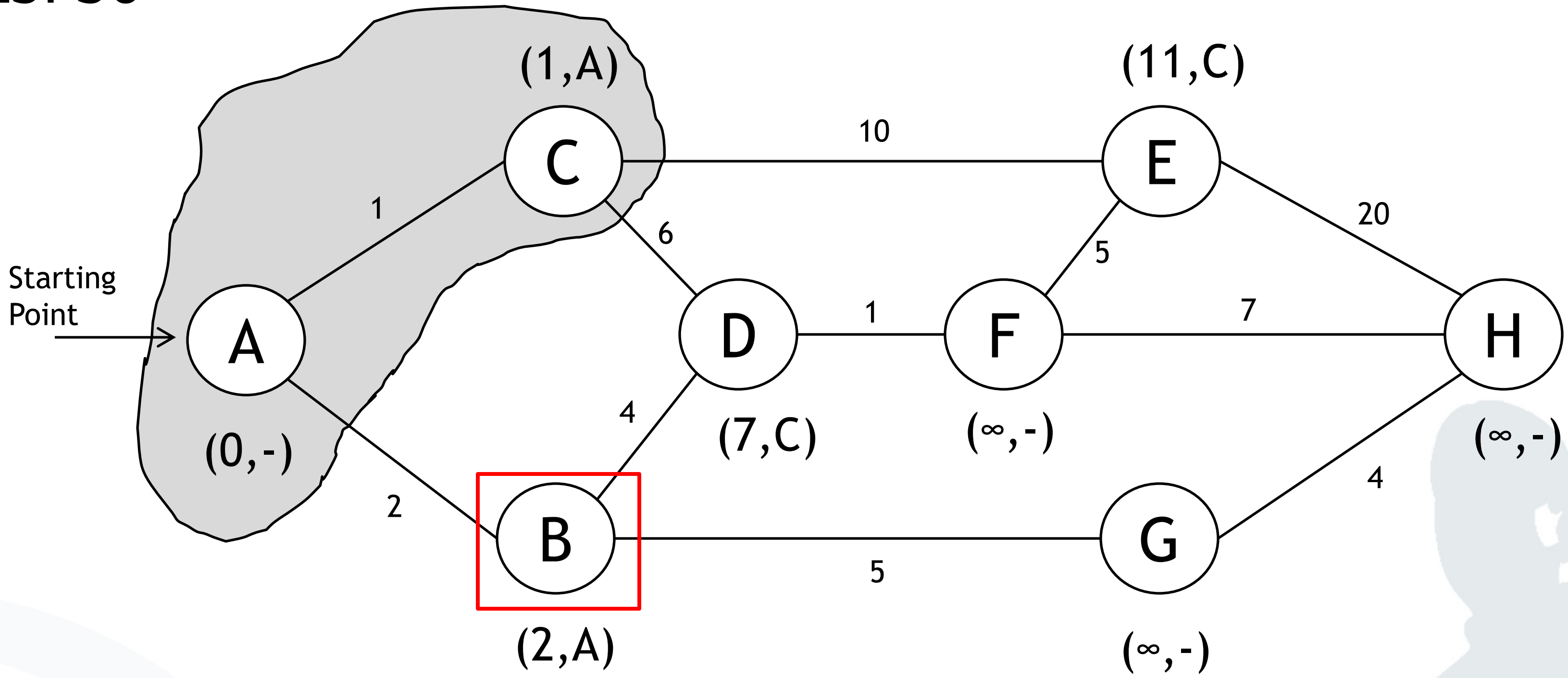
VL5F29



- Add last selected vertex to the set: A
- If shorter, update distance and predecessor values of the neighbours of the last selected vertex: B and C
- Select the vertex, which is not in the set and has the minimum value: C

Layer 3: Network Layer Using Dijkstra Algorithm (3)

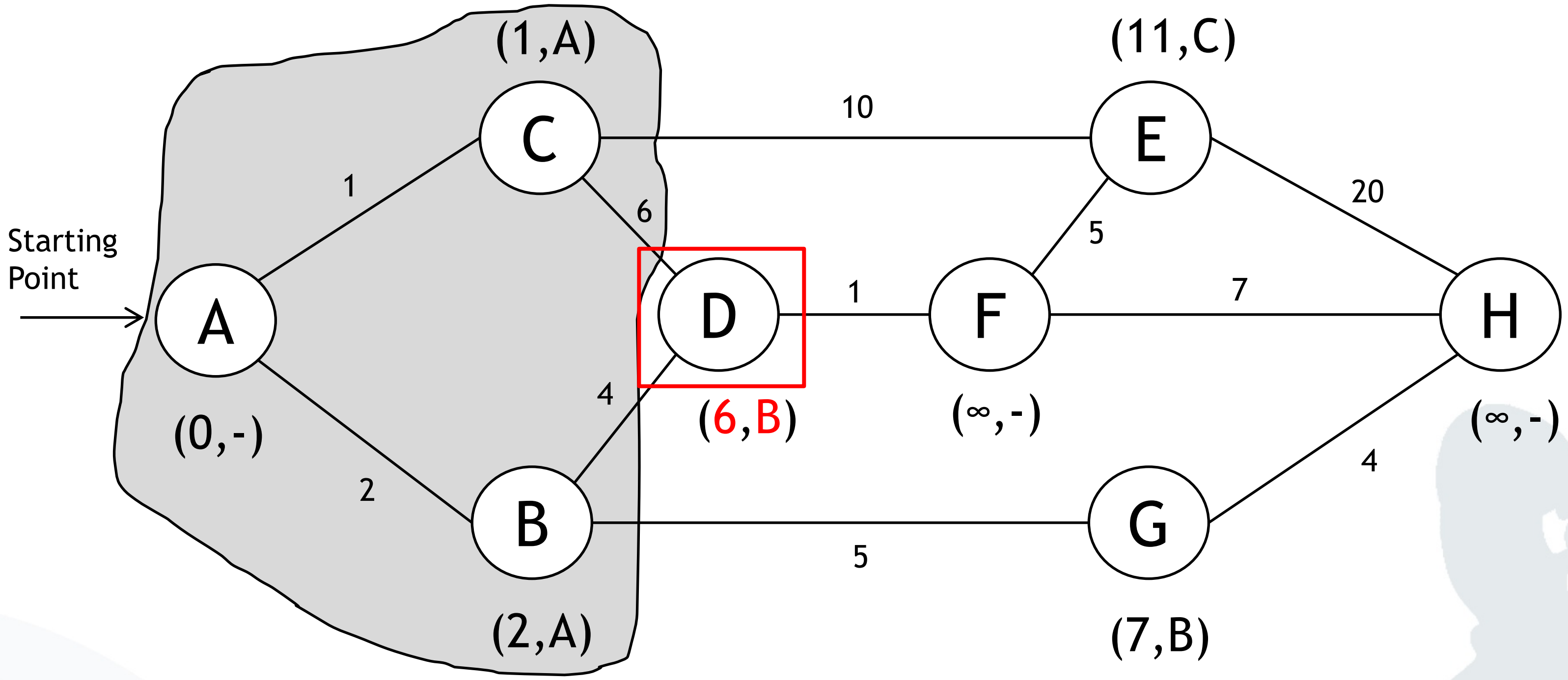
VL5F30



- Add last selected vertex to the set: C
- If shorter, update distance and predecessor values of the neighbours of the last selected vertex: D and E
- Select the vertex, which is not in the set and has the minimum value: B

Layer 3: Network Layer Using Dijkstra Algorithm (4)

VL5F31



- Add last selected vertex to the set: B
- If shorter, update distance and predecessor values of the neighbours of the last selected vertex: D and G
- Select the vertex, which is not in the set and has the minimum value: D

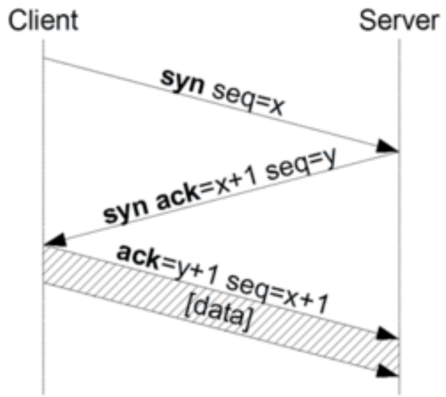
Transmission Control Protocol, VL5

- Bezuglich der Klausur wurde ich sehr dankbar sein, wenn Sie das Setup of a Transmission Control Protocol (Lecture5 Folie53) noch einmal erklaren wurden.

VL5F53

Setup of a TCP connection by 3-way handshake

- Computer (client) sends a SYN to the remote station (server). SYN packets („synchronise“) have a sequence number x.
- Sequence numbers are important in order to determine if the transmission was completed in the correct order and without duplicates.
- The remote station (server) receives the SYN packet.
  - In case the port is closed, it replies with **TCP-RST**.
  - In case the port is open, it sends a SYN ACK providing its own starting sequence number y. At the same time, the remote station acknowledges the receipt of the first SYN packet by increasing its sequence number by one and including it in the ACK part (“acknowledgment”) of the header.
- The computer (client) receives and acknowledges receipt of the SYN ACK packet by sending an ACK packet with the sequence number y+1 (this is also called a „forward acknowledgement“). Also, the client sends the sequence number x+1 to the server.
- This ACK segment contains information about the remote station and the ACK flag serves as a label.
- Connection has been successfully set up and the actual data transmission can start.



VL5F52

- Example from everyday life – making an appointment via correspondence

Prof. Rannenberg wants to make an appointment with Prof. König via correspondence.

1. Prof. Rannenberg sends a message to Prof. König to suggest an appointment date.
2. Prof. König confirms the appointment date by sending a message back to Prof. Rannenberg.
3. Prof. Rannenberg sends a message to Prof. König to let him know that he received the confirmation message.

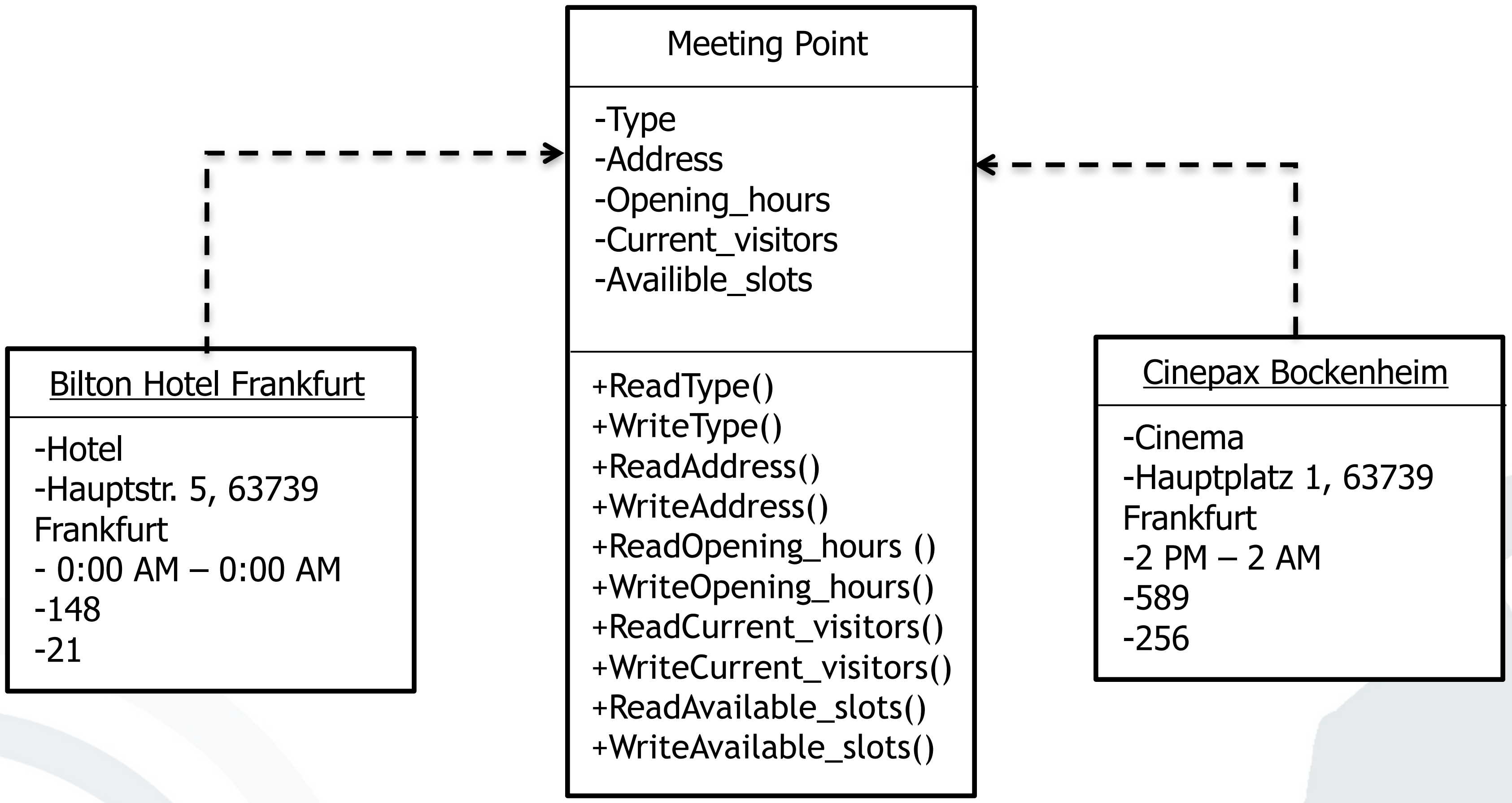
Step 3 is necessary in order for Prof. König to know that Prof. Rannenberg has received the confirmation. Message No. 2 could have gotten lost and then Prof. König would show up alone for the meeting.

Klassendiagramm, VL9

•Es wäre toll, wenn der Aufbau vom Klassen-Diagramm wiederholt werden würde.

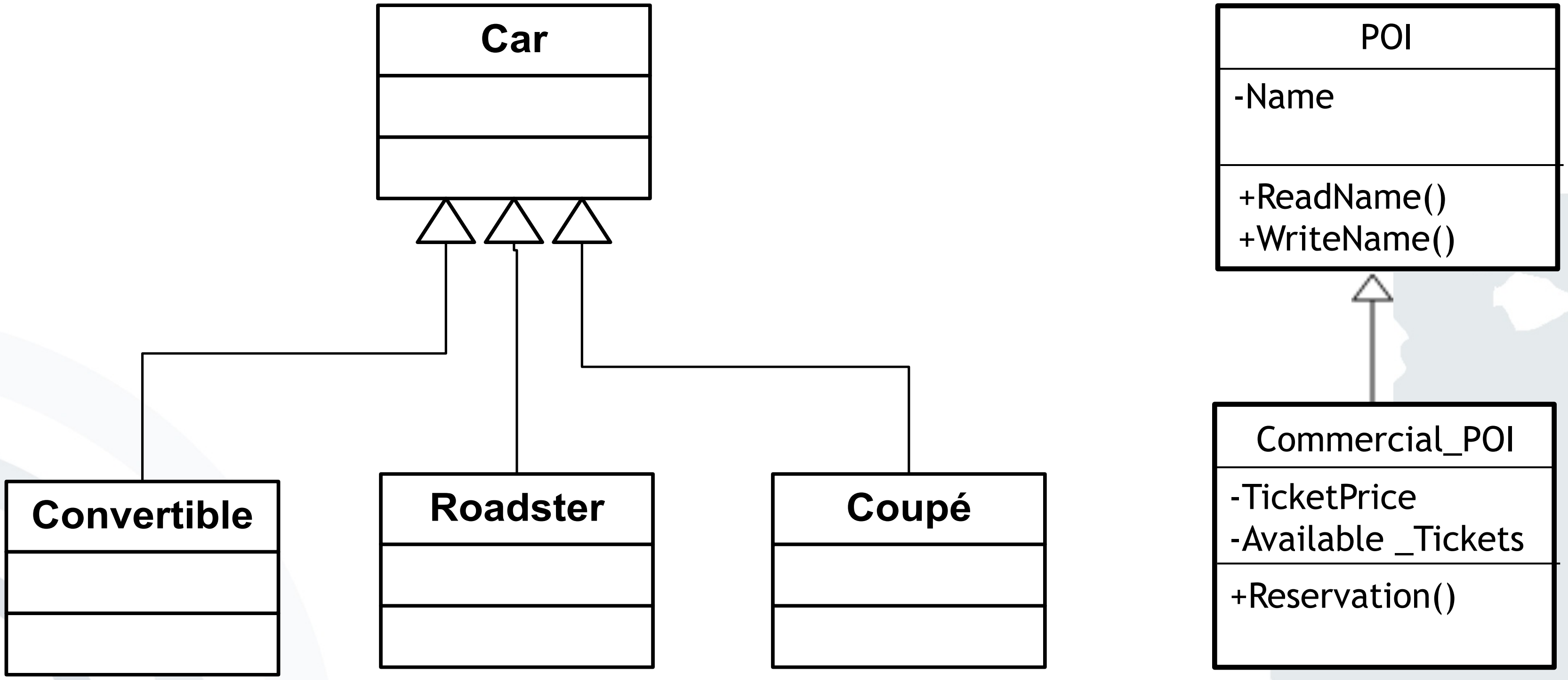
•Für was steht +, - beim Klassen-Diagramm? Werden + und - sowohl bei Attributen als auch bei Methoden verwendet?

Class Diagram



Class Diagram

- Inheritance
  - Classes can inherit attributes or methods to other classes. The inheriting class is called “super class” or “parent class”. The new class is called a “sub class”.

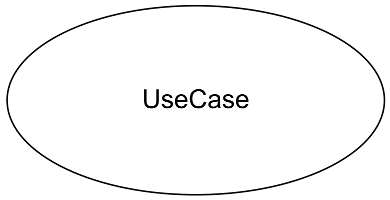


UML, VL9

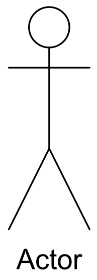
- Könnten Sie bitte UseCase-Diagramme nochmal erklären?

Use Case Diagram Notation Elements (1/3)

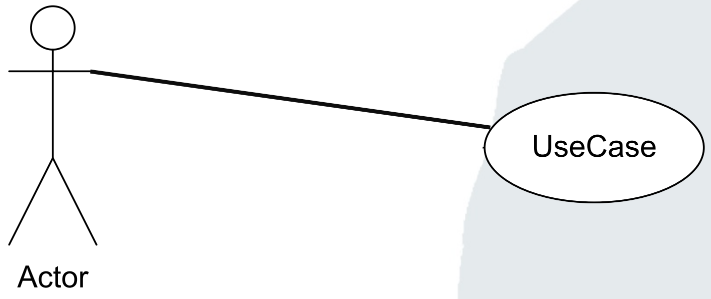
- **Use Case**
  - Representation of a sequence of actions that provides value to an actor.



- **User of the system**

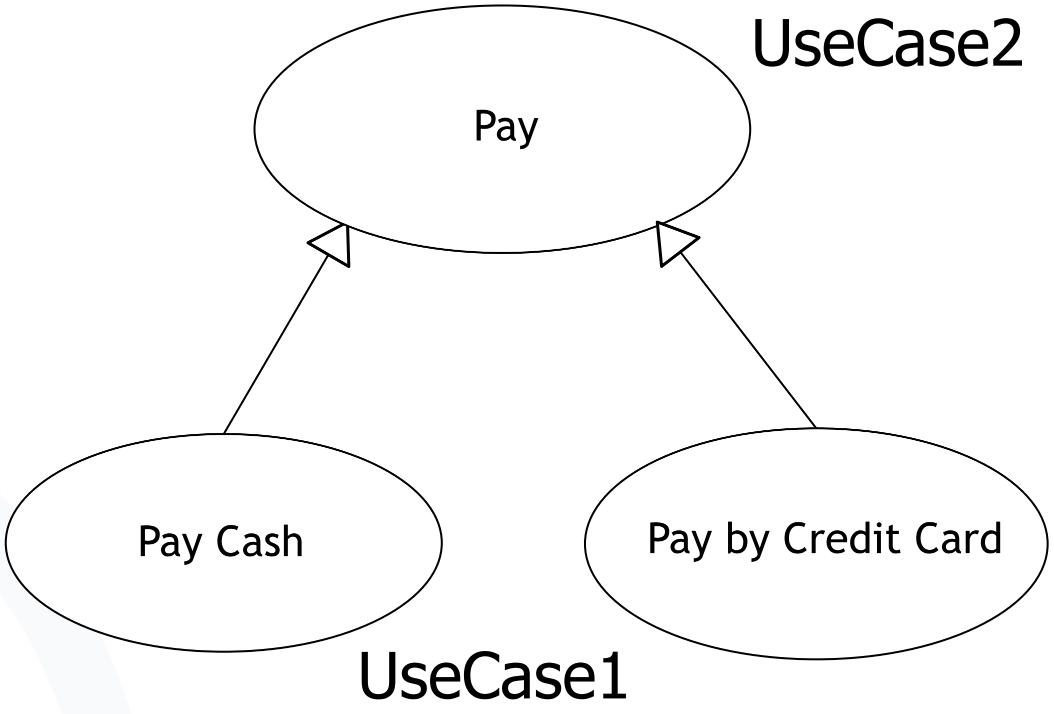


- **Association**
  - Interaction of an actor with a use case



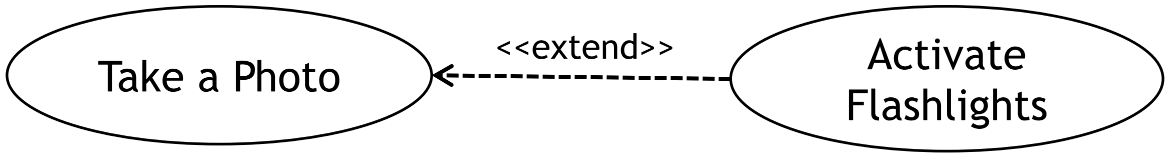
Use Case Diagram Notation Elements (2/3)

- **Generalisation**
  - Generalisation of use cases
  - UseCase2 generalises the behaviour of UseCase1

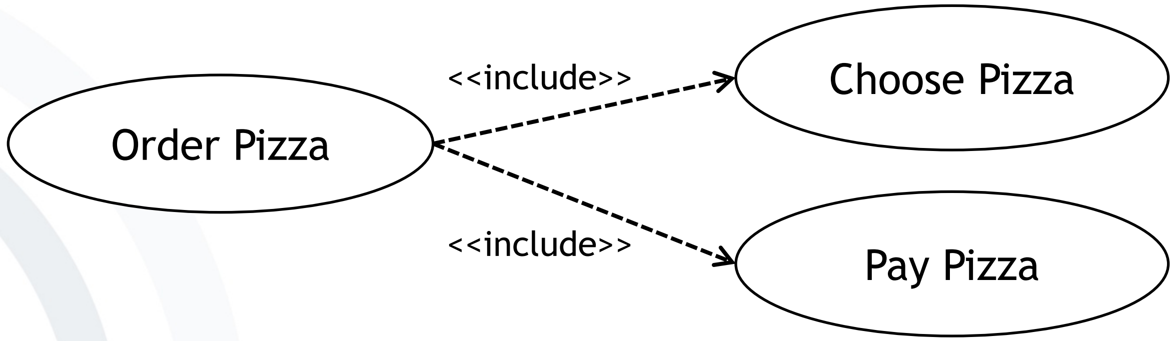


Use Case Diagram Notation Elements (3/3)

- **<<Extend>>**
  - Extends a use case
  - UseCase2 extends UseCase1



- **<<Include>>**
  - Inclusion of a use case
  - UseCase1 includes the behaviour of UseCase2

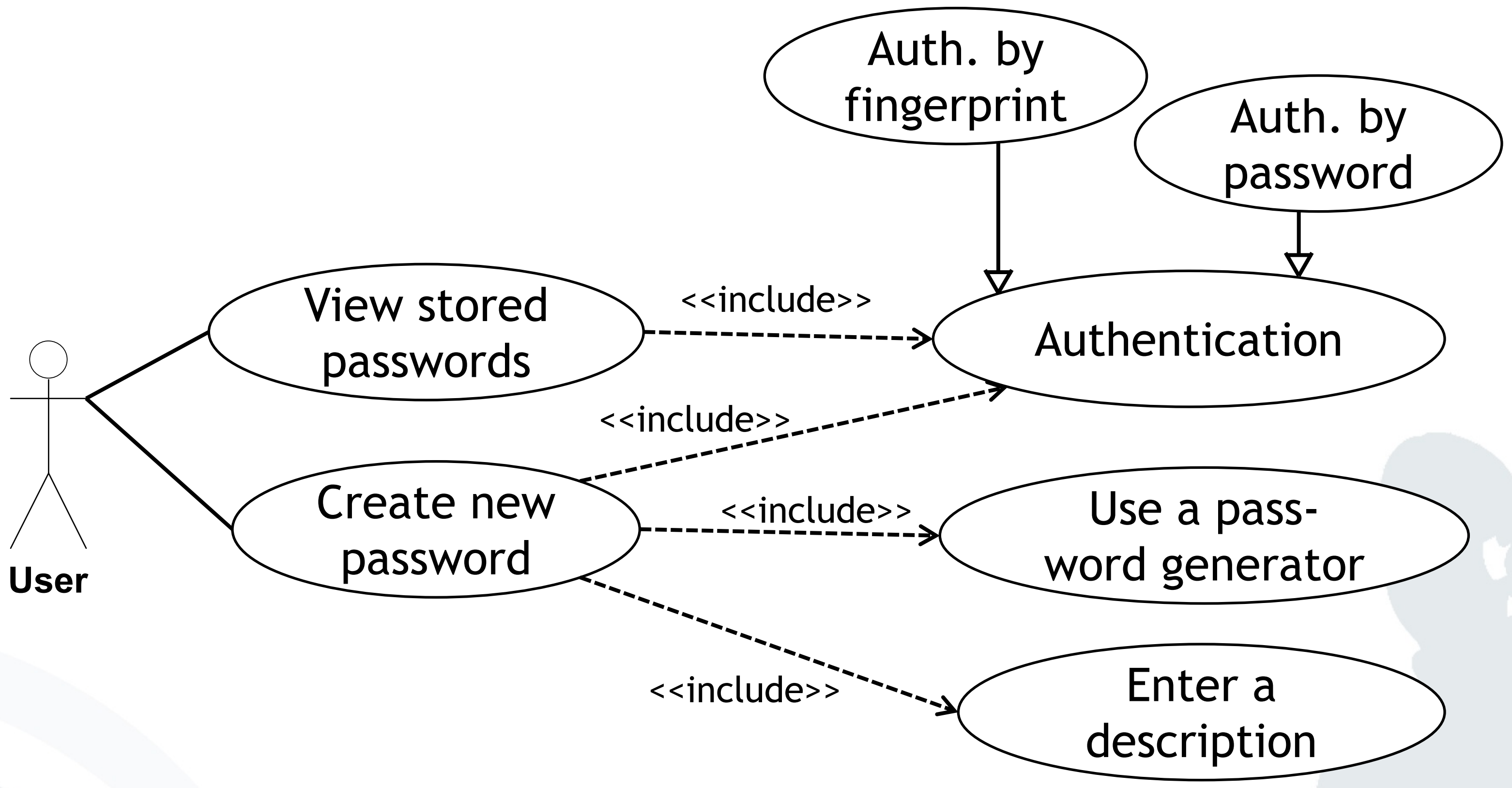


Use Case Diagram

Create a use case diagram for a password manager app:

A user can either **view stored passwords** or **create a new password**. Both use cases require an **authentication**. The app supports both user **authentication by password** and **authentication by fingerprint**. To **create a new password**, the user has to **use** an integrated **password generator**. Furthermore, he has to **enter a description** for his new password.

Use Case Diagram

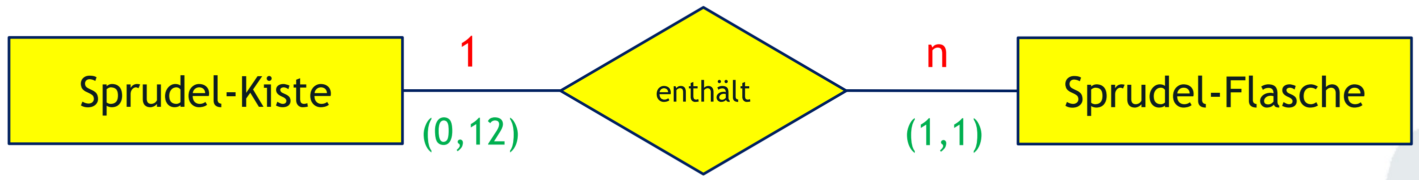


ERM, VL11

- Können Sie die Intervall-Notation bei ER-Modellen nochmal erklären?
- Könnten Sie bitte nochmal den Begriff "Schwache Entität" erklären?

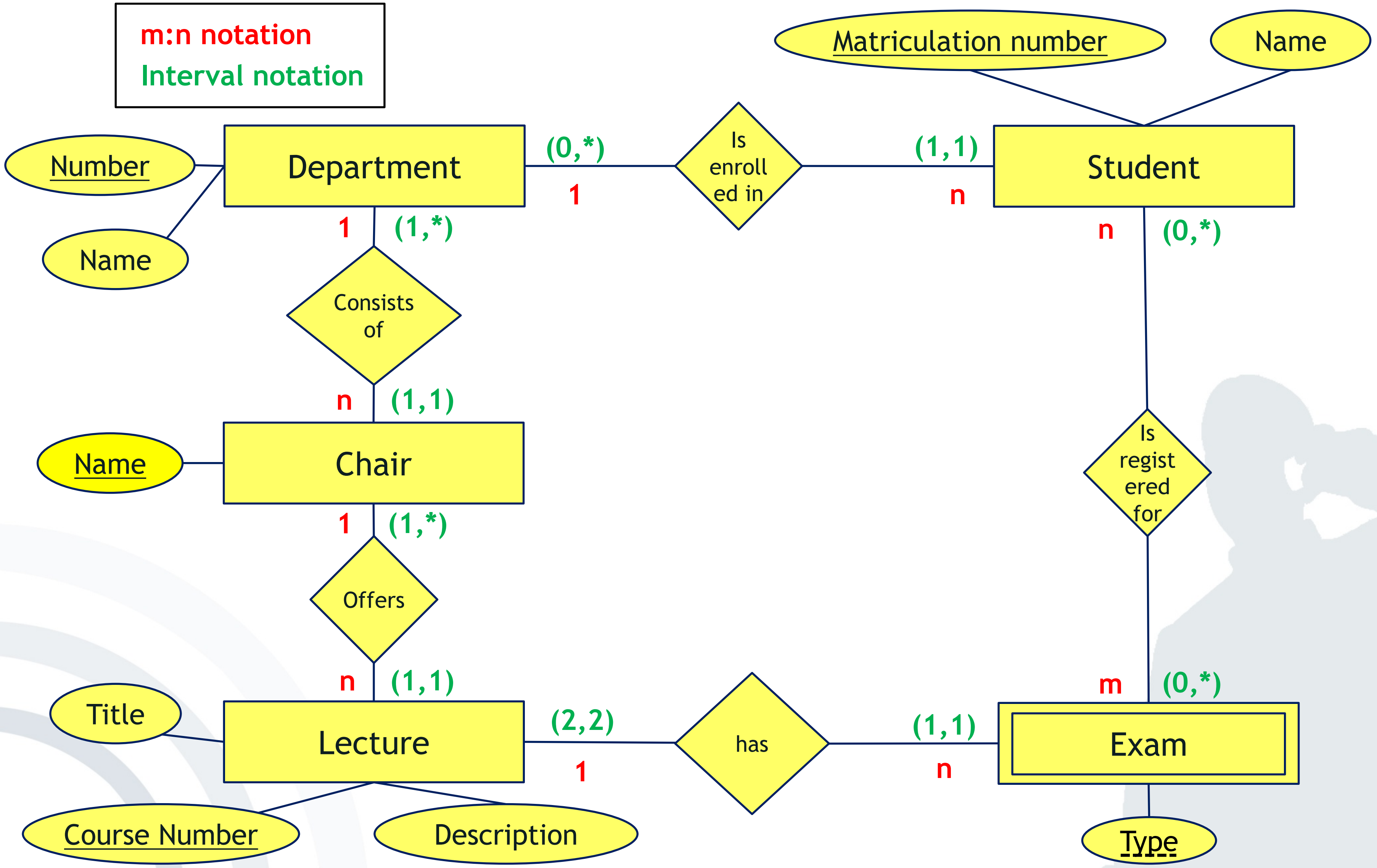
ERM: Cardinalities and Intervals

Cardinalites



Intervals (according to Ferstl/Sinz, 2001)

ERM: Cardinalities and Intervals, Weak Entities



Open Questions?