



## *Lecture 14*

### Business Informatics 2 (PWIN) WS 2019/20

#### Questions and Answers

**Prof. Dr. Kai Rannenber**

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

- Questions on the Lecture
- Questions on Exercises and Mentoriums

- *„Im Layer 7 des OSI Models sind wir in der Vorlesung nur auf das Domain Name System genauer eingegangen. Jedoch gibt es ja noch zahlreiche weitere Protokolle wie FTP oder HTTP. Ist es von Relevanz diese auch genauer zu kennen bzw. erklären zu können?“*

# 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

## Layer 7: Application Layer

- The **application layer** contains a multitude of protocols often needed by users for their applications to work.
  - Examples:
    - DNS,
    - E-Mail,
    - HTTP,
    - FTP,
    - ...

# 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.

- *„Was genau besagt der 802.11i Standard bei der WPA2 Verschlüsselung? Warum ist es wichtig, dass die Encryption diesen Standard enthält?“*

# Wireless LAN (Wi-Fi)

## State-of-the-Art Encryption (2)

- **Wi-Fi Protected Access** was developed by the Wi-Fi Alliance Source: Wi-Fi (2010)

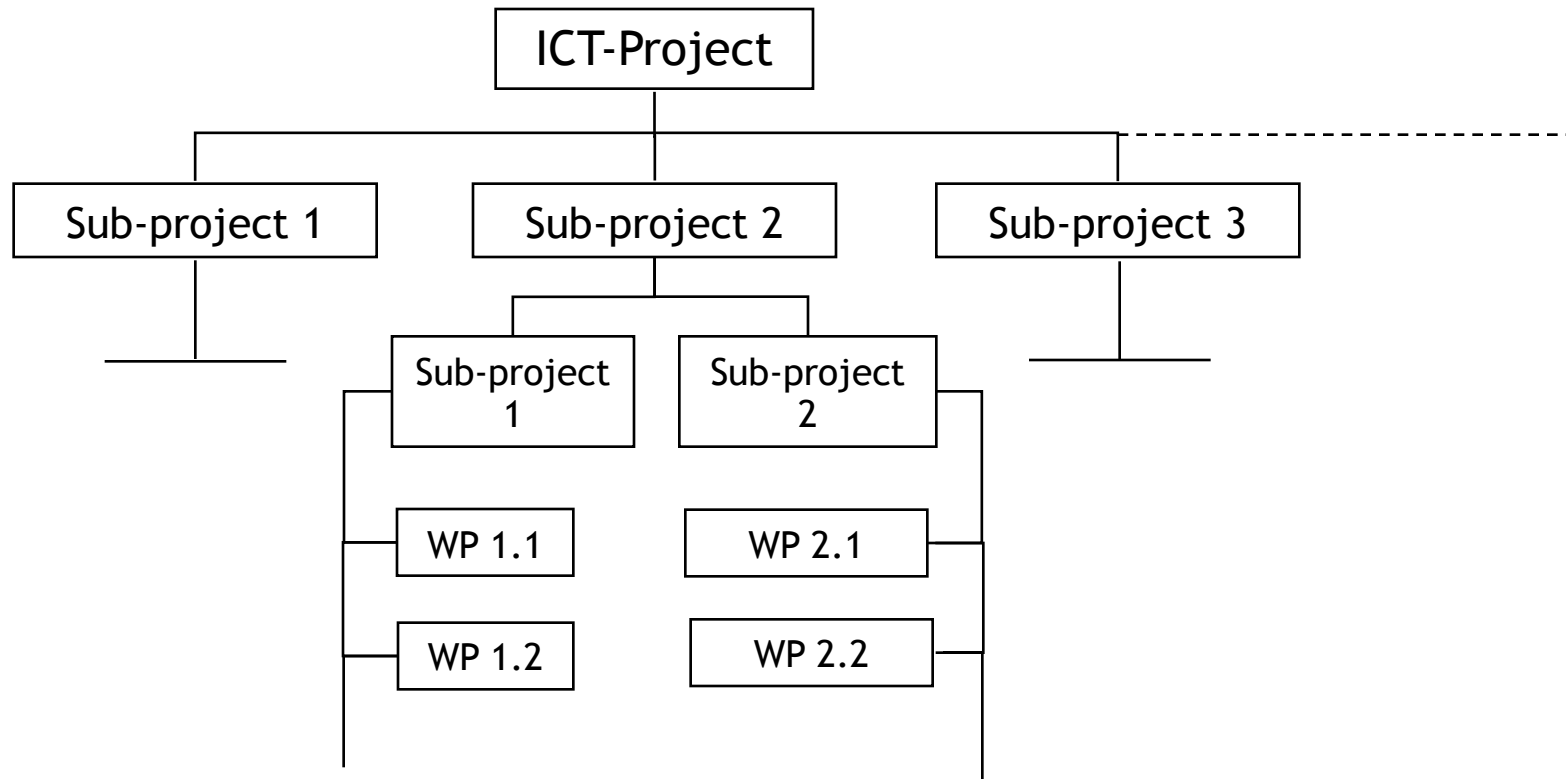


- There are two versions of Wi-Fi Protected Access, WPA and WPA2:
  - **WPA** includes most of the 802.11i standard.
  - **WPA2** includes **802.11i** to full extent and also the Advanced Encryption Standard (AES).



- *“Können Sie nochmal erläutern, wie man einen Projektstrukturplan aufstellt?”*

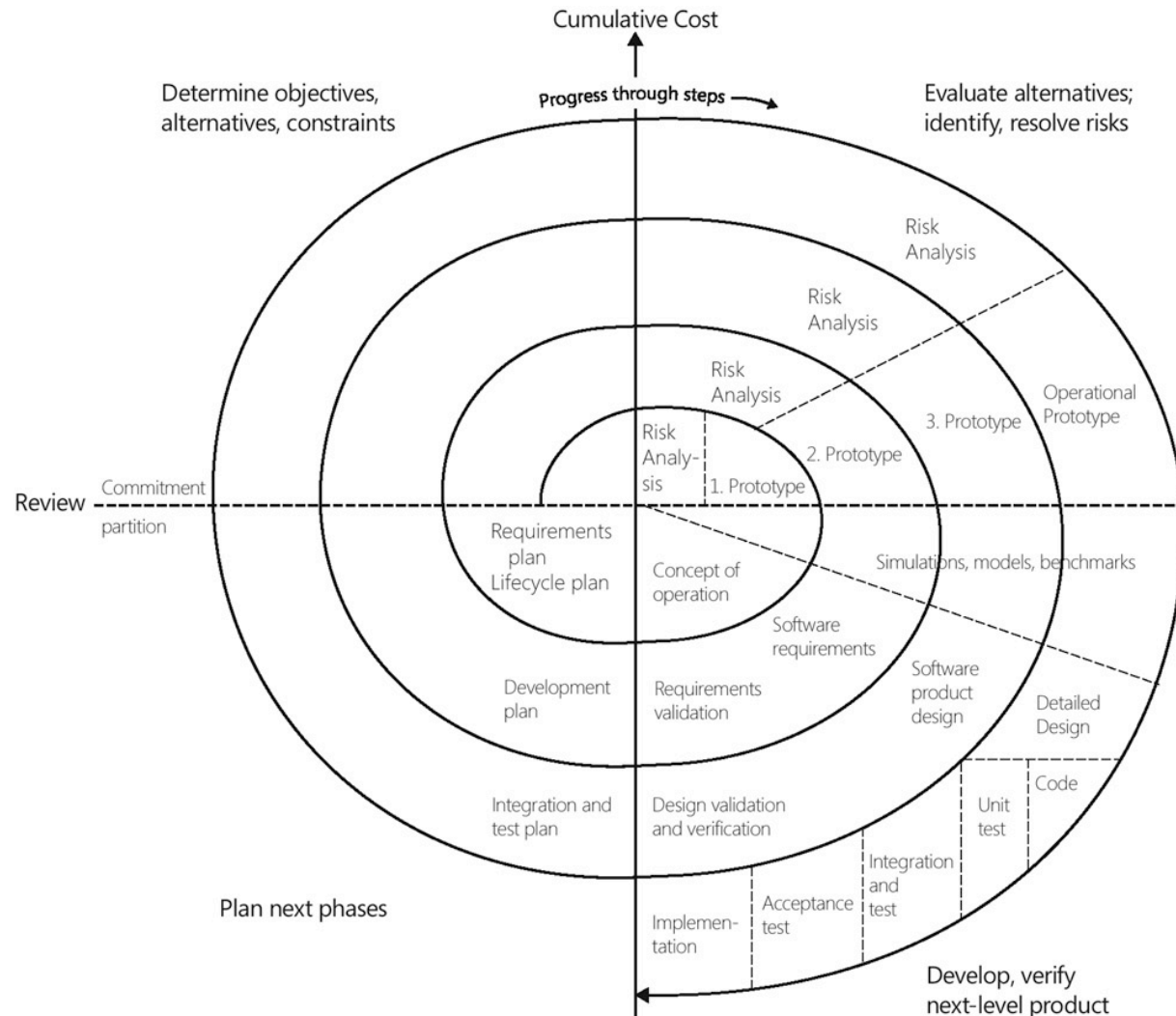
# Project Structure Plan



- Breakdown of the project in sub-projects and work packages (WP).
- If the resulting project structure plan contains the efforts and responsibilities, it is called a Work Breakdown Structure (WBS).

- *„Könnten Sie noch einmal genauer auf das Spiral Model eingehen? Mir ist unklar wo man genau beginnt und inwiefern in jeder neuen Iteration neue Prototypen entwickelt werden oder diese nur von Stufe zu Stufe weiterentwickelt werden?“*
- *„Was ist der Unterschied zwischen Exploratory und Throw-away Prototyping?“*
- *„Was wäre ein gutes Beispiel, um ein Spiral Modell anzuwenden?“*

# Evolutionary Model Example: Spiral Model



Source: Marciniak (2002)

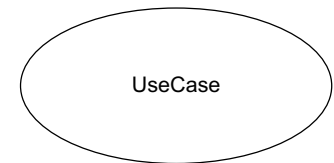
# Evolutionary Model Example: Spiral Model

- Basic Concept
  - Develop an initial implementation, demonstrate it to user, get feedback and refine it until an adequate system has been produced.
- Two types of evolution models:
  - *Exploratory*
  - *Throw-away prototyping*
- Advantages
  - Estimates for budget, schedule, etc. become more realistic as work progresses
- Disadvantages
  - Requires expertise in risk evaluation and mitigation
  - Appropriate only for large systems

- *“Können Sie bitte nochmal auf die wichtigsten Notationselemente des Use Case Diagramms eingehen und nochmal kurz das Beispiel aus der Vorlesung erläutern?”*

# Use Case Diagram Notation Elements

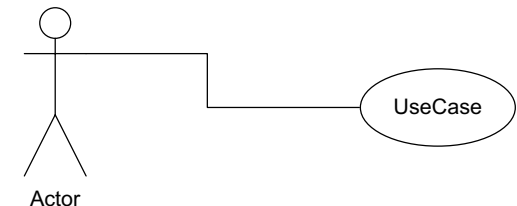
- Use Case
  - Representation of a sequence of actions that provides value to an actor.



- User of the system

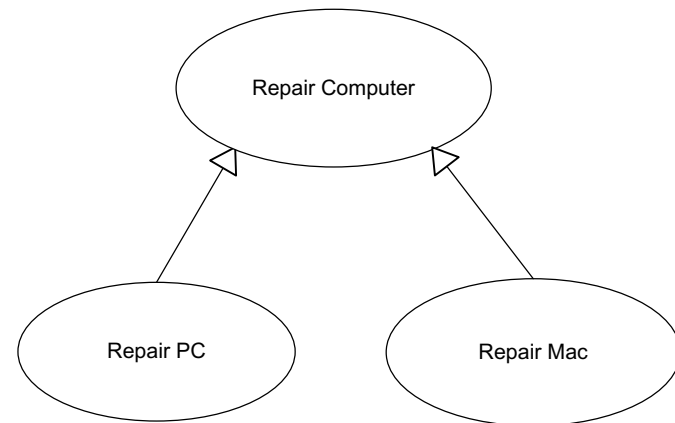
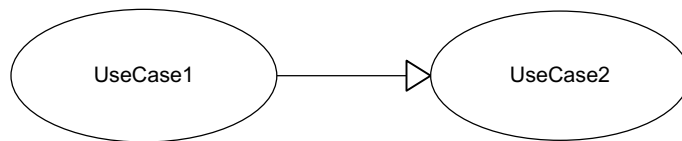


- Association
  - Interaction of an actor with a use case



# Use Case Diagram Notation Elements

- Generalisation
  - Generalisation of Use Cases
  - UseCase2 generalises the behaviour of UseCase1

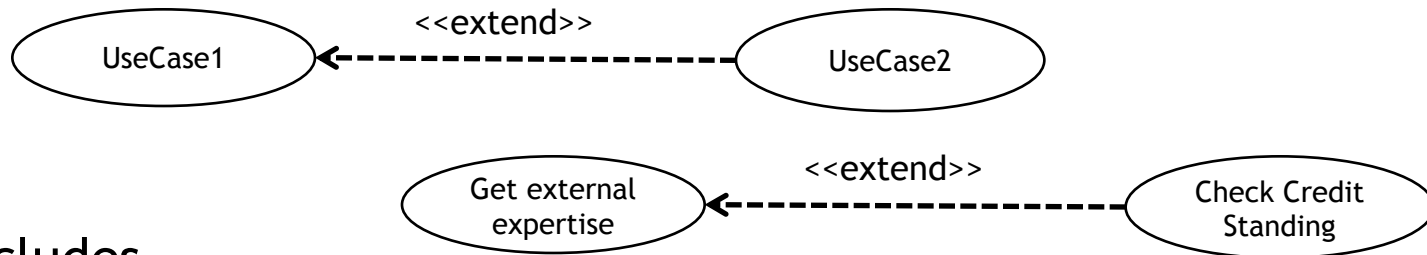




# Use Case Diagram Notation Elements

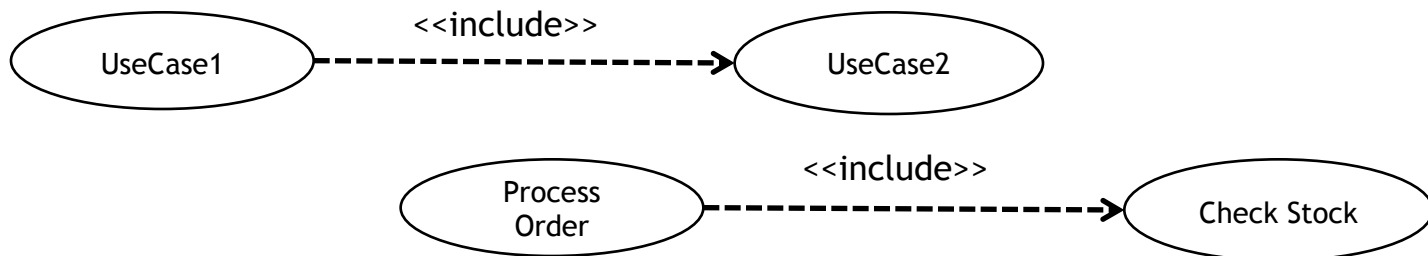
- Extends

- Extends a Use Case
- UseCase2 extends UseCase1

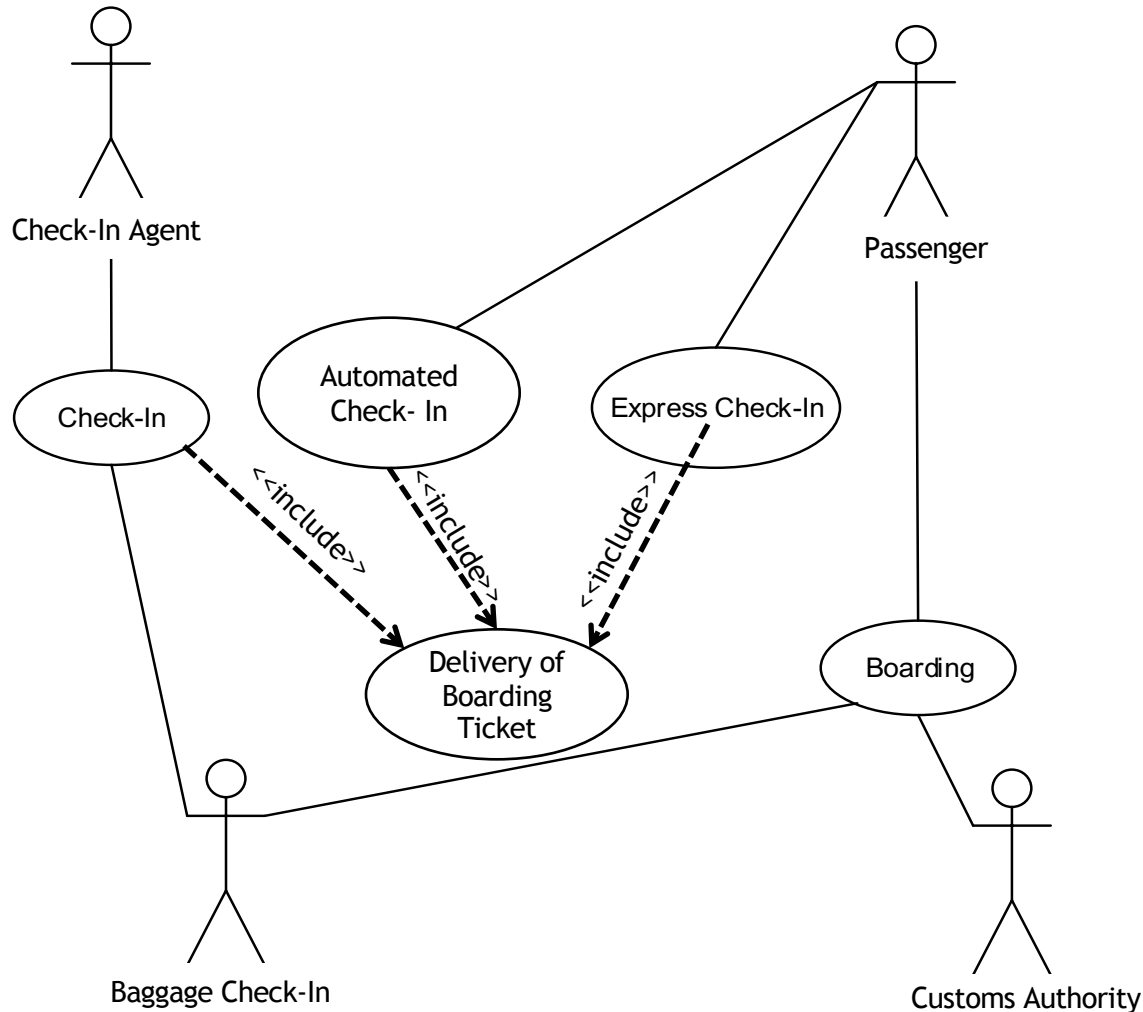


- Includes

- Inclusion of a Use Case
- UseCase1 includes the behaviour of UseCase2



# Use Case Diagram (Example)



- Questions on the Lecture
- Questions on Exercises and Mentoriums

- *„Bei der Aufzählung der Charakteristika der Mobilen Daten Kommunikation, ist mir unverständlich was es genau mit der „Limited I/O capabilities“ Eigenschaft auf sich hat.“*

# Unique Characteristics of Mobile Data Communications

- Time and location independence
- Instant on of mobile devices
- Limited I/O capabilities
- Location awareness
- Personal nature of the medium
- Identification of mobile user
- 1:1 communications
- High penetration in the population

# Unique Characteristics of Mobile Data Communications

- Time and location independence
- Instant on of mobile devices
- Limited I/O capabilities
  - Gemeint ist, dass ein mobiles Endgerät aufgrund des kleinen Bildschirms nur limitierte Input/Outputmöglichkeiten hat, was die Ein- und Ausgabe von Informationen erschwert
- Location awareness
- Personal nature of the medium
- Identification of mobile user
- 1:1 communications
- High penetration in the population

- *„In welchem Fall ist es besser eine Web App anstelle einer Native App zu benutzen?“*

# Mobile Web Apps vs. Mobile Apps

## Mobile app (“native App”)

## Mobile web app

Supports offline use

Needs constant internet connectivity (network coverage)

Can be found easily in app store(s)

Distribution via URL, e.g. QR-codes

Business model: Sold in app store(s)

Difficult to implement payment and authentication system

Can make use of all OS and device functions

Cannot access OS core functions (e.g. 3D graphic processing or access to locally protected storage)

Needs to be platform-specific (native code)

Using web browser of the device, hence manufacturer-independent multi-platform support possible; also porting to other devices/platforms is less expensive

Based on Objective-C, C#.Net, Java

Based on HTML5, CSS, Javascript

Updates/versioning through app stores

Easy updates as they are done on the server, not on every client device



# Mobile Web Apps vs. Mobile Apps

## Mobile app (“native App”)

Supports offline use

Can be found easily in app store(s)

Business model: Sold in app store(s)

Can make use of all OS and device functions

Needs to be platform-specific (native code)

Based on Objective-C, C#.Net, Java

Updates/versioning through app stores

## Mobile web app

Needs constant internet connectivity (network coverage)

Distribution via URL, e.g. QR-codes

Difficult to implement payment and authentication system

Cannot access OS core functions (e.g. 3D graphic processing or access to locally protected storage)

Using web browser of the device, hence manufacturer-independent multi-platform support possible; also porting to other devices/platforms is less expensive

Based on HTML5, CSS, Javascript

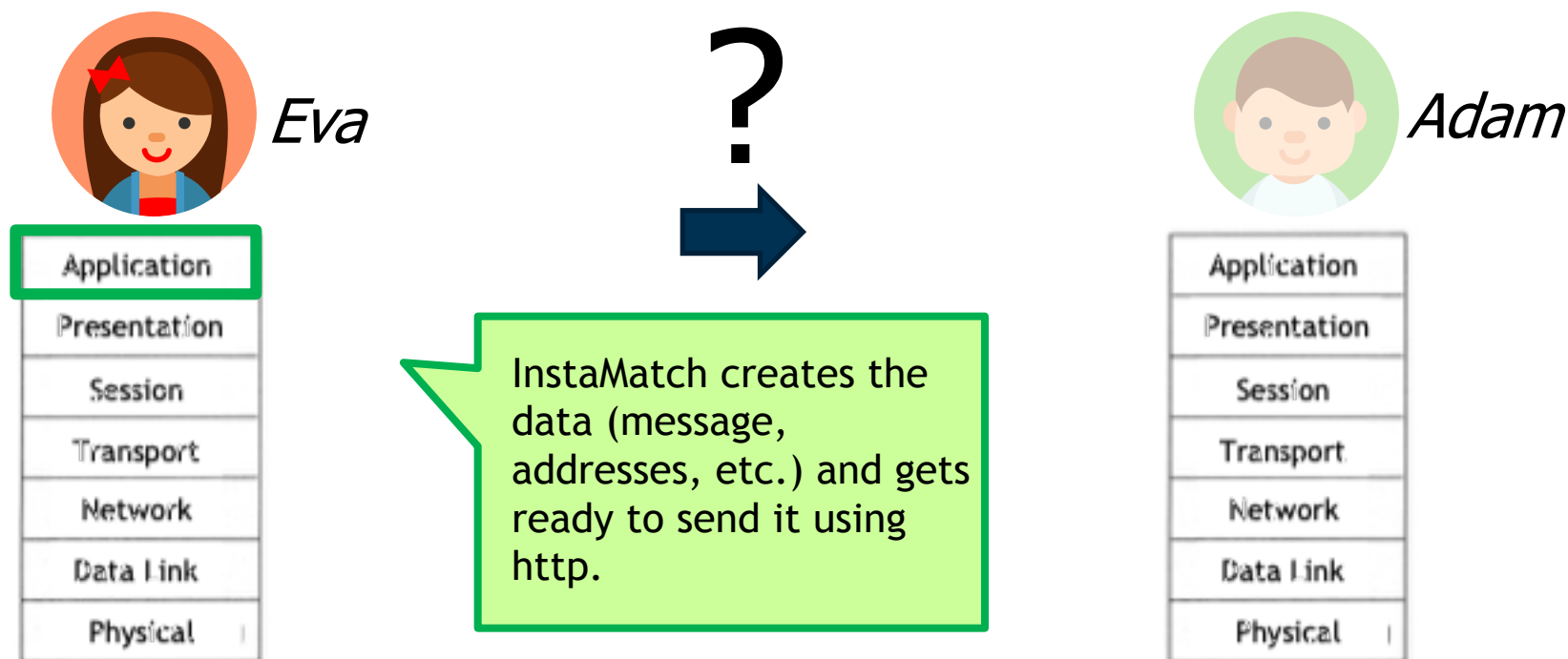
Easy updates as they are done on the server, not on every client device

- *„Wie granular muss man die einzelnen Komponenten des GSM erklären können?“*

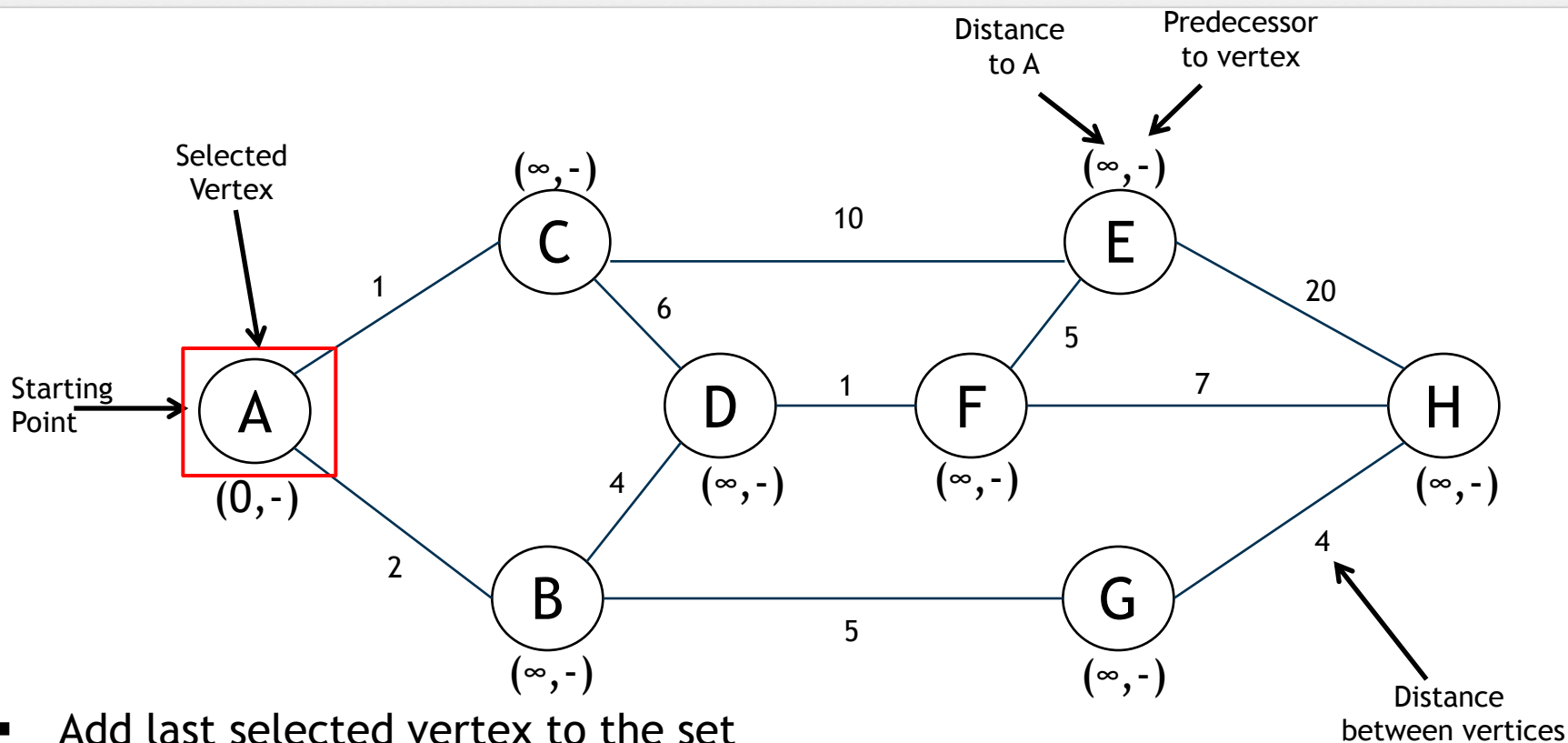
- *„Wie granular muss man die einzelnen Komponenten des GSM erklären können?“*
- You should know how cell-based communication works in general
- You should know what happens if you move in the network
- You do NOT need to know details about the individual components of GSM

**Exercise 3 - task 2:** *“Briefly explain the information flow in the OSI reference model when a user of InstaMatch sends a message to another user who is not in the same network.”*

- In principle no difference



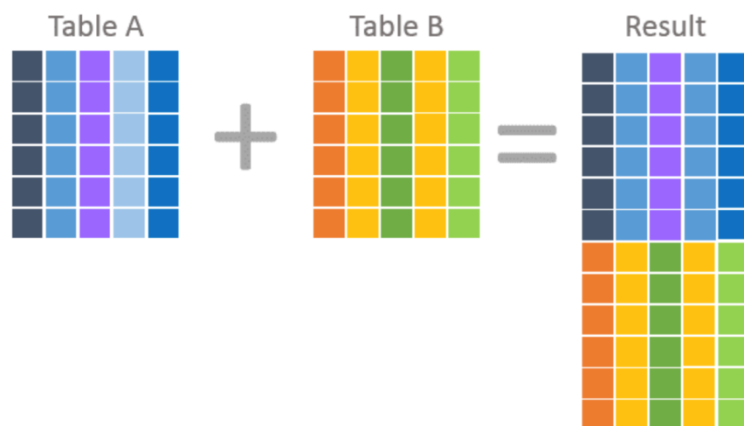
# Layer 3: Network Layer Using Dijkstra Algorithm



- Add last selected vertex to the set
- If shorter (longer), update distance and predecessor values of the neighbours of the last selected vertex
- Select the vertex, which is not in the set and has the minimum (maximum) value

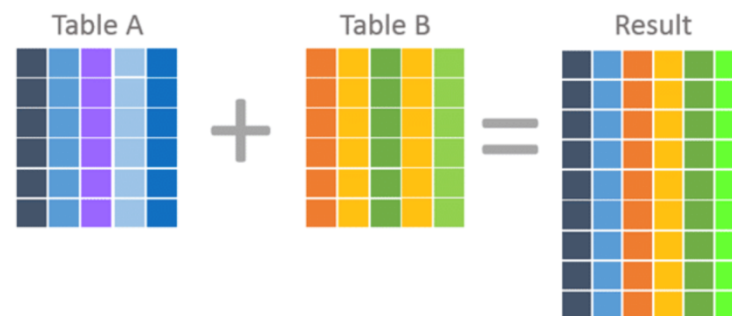
*"How relevant are UNION and CREATE PROCEDURE?"*

## ■ UNION



- combines rows from tables

## ■ JOIN



- combines columns from tables

*"How relevant are UNION and CREATE PROCEDURE?"*

```
CREATE PROCEDURE SelectAllCustomers  
AS  
SELECT * FROM Customers  
GO
```

```
EXEC SelectAllCustomers;
```

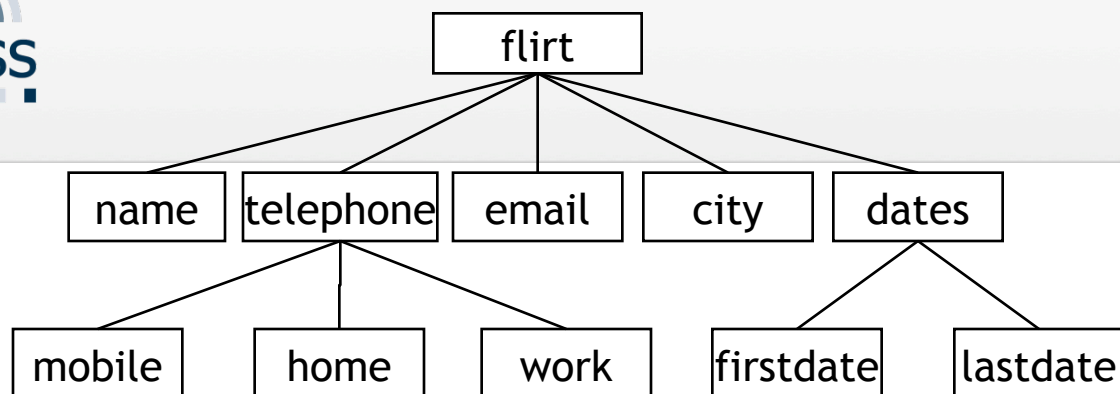
Number of Records: 91

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Frankfurt	12209	Germany
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

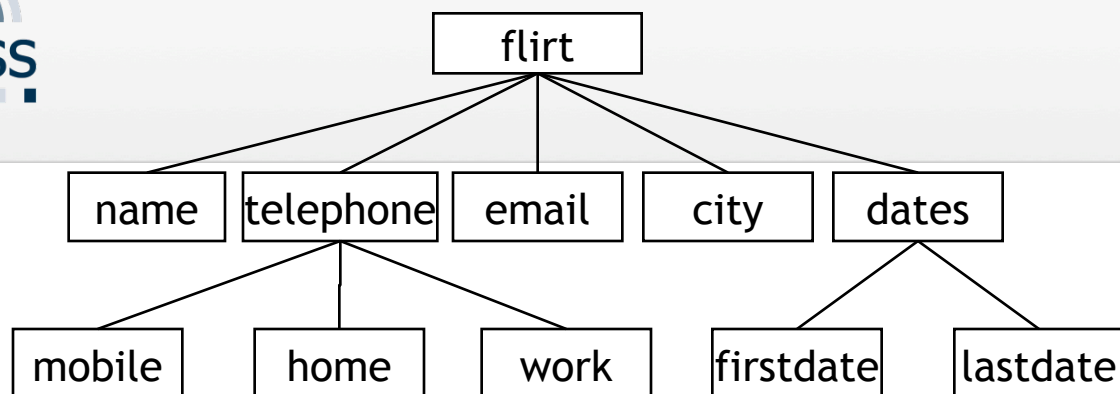
- *“Was ist der Unterschied zwischen ,(Komma) und | bei der DTD?”*



- Cardinalities (for elements):
  - empty: exactly one value is necessary
  - + At least one value
  - ? None or one value
  - \* None or multiple values
  
- Content (in elements):
  - EMPTY Empty element
  - ANY Any content
  - | Selection list
  - , Sequence
  - () Grouping
  - (#PCDATA) *Parsed Character Data* (mixed data)

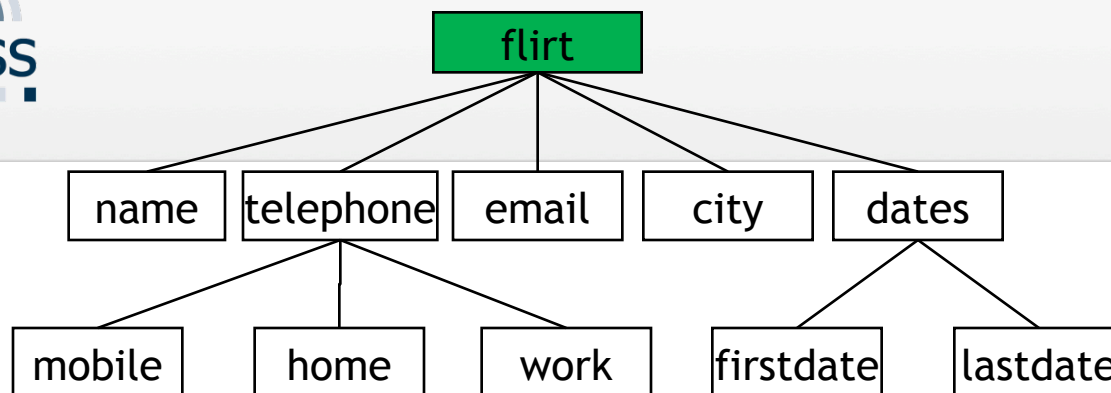


```
<flirt>
  <name>Daisy</name>
  <telephone>
    <mobile> 0177 / 1234567 </mobile>
  </telephone>
  <email> daisy@m-chair.de </email>
  <city>  Innsbruck </city>
  <dates>
    <firstdate>2019-01-01 </firstdate>
    <lastdate> 2019-01-10 </lastdate>
  </dates>
</flirt>
```



**Rule declaration for the elements in a DTD:**

```
<!DOCTYPE flirt [  
]>
```

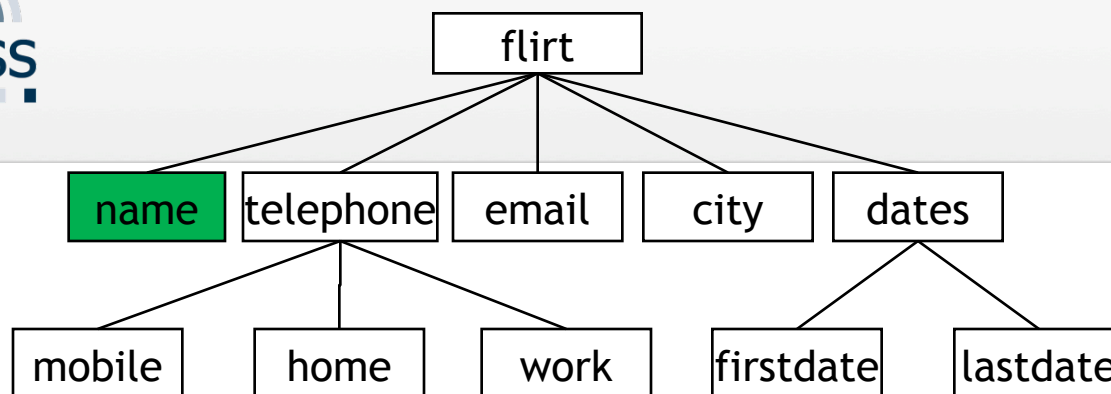


Rule declaration for the elements in a DTD:

```
<!DOCTYPE flirt [
```

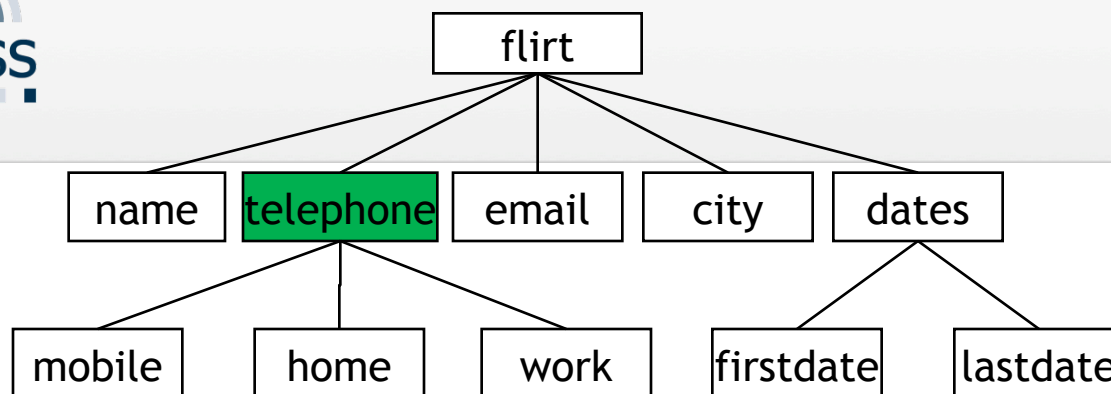
```
<!ELEMENT flirt (name, telephone, email, city, dates)>
```

```
]>
```



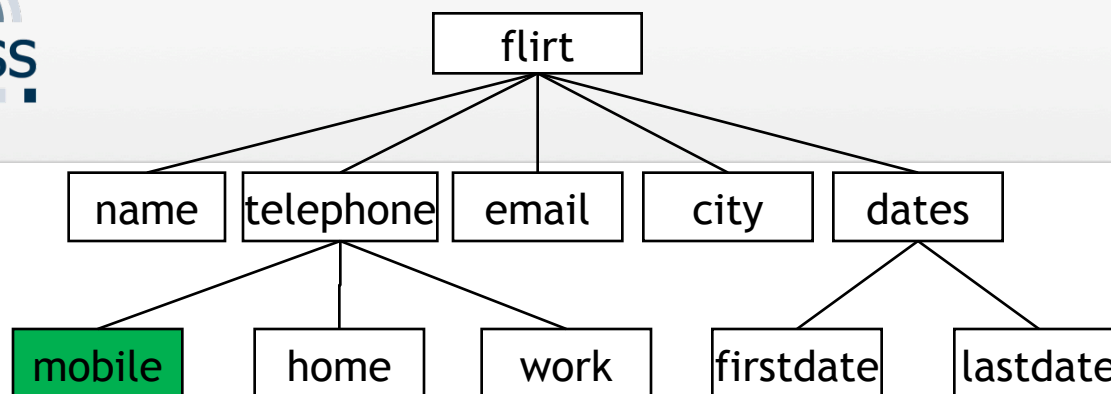
Rule declaration for the elements in a DTD:

```
<!DOCTYPE flirt [  
<!ELEMENT flirt (name, telephone, email, city, dates)>  
<!ELEMENT name      (#PCDATA)>  
>
```



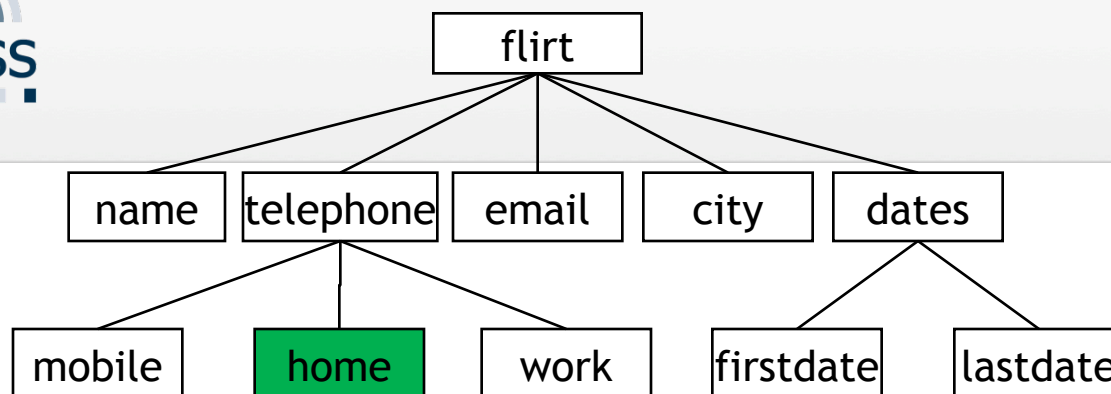
Rule declaration for the elements in a DTD:

```
<!DOCTYPE flirt [  
<!ELEMENT flirt (name, telephone, email, city, dates)>  
<!ELEMENT name      (#PCDATA)>  
<!ELEMENT telephone (mobile | home | work)>  
>
```



Rule declaration for the elements in a DTD:

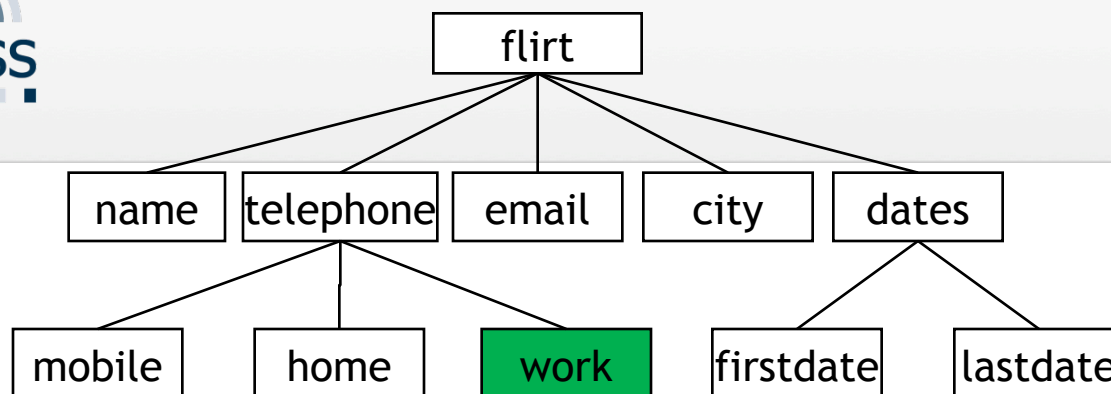
```
<!DOCTYPE flirt [  
<!ELEMENT flirt (name, telephone, email, city, dates)>  
<!ELEMENT name      (#PCDATA)>  
<!ELEMENT telephone (mobile | home | work)>  
<!ELEMENT mobile    (#PCDATA)>  
>
```



Rule declaration for the elements in a DTD:

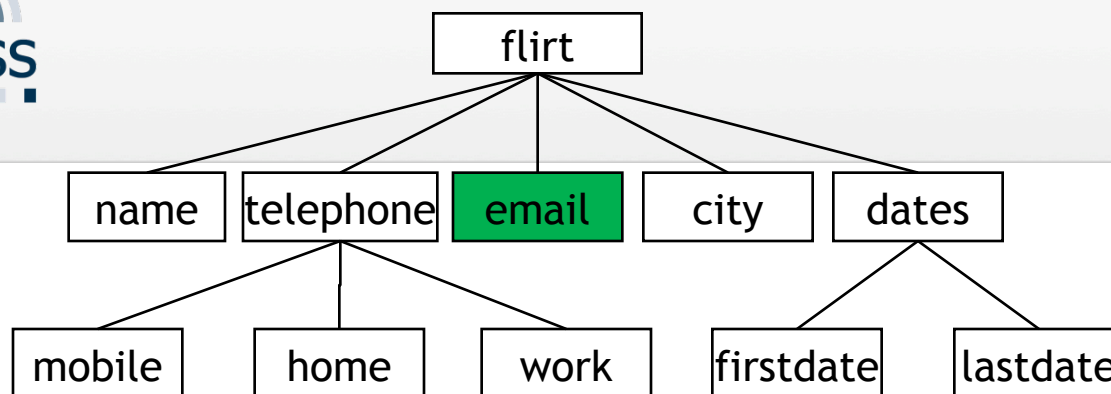
```
<!DOCTYPE flirt [  
<!ELEMENT flirt (name, telephone, email, city, dates)>  
<!ELEMENT name      (#PCDATA)>  
<!ELEMENT telephone (mobile | home | work)>  
<!ELEMENT mobile     (#PCDATA)>  
<!ELEMENT home       (#PCDATA)>  
>
```





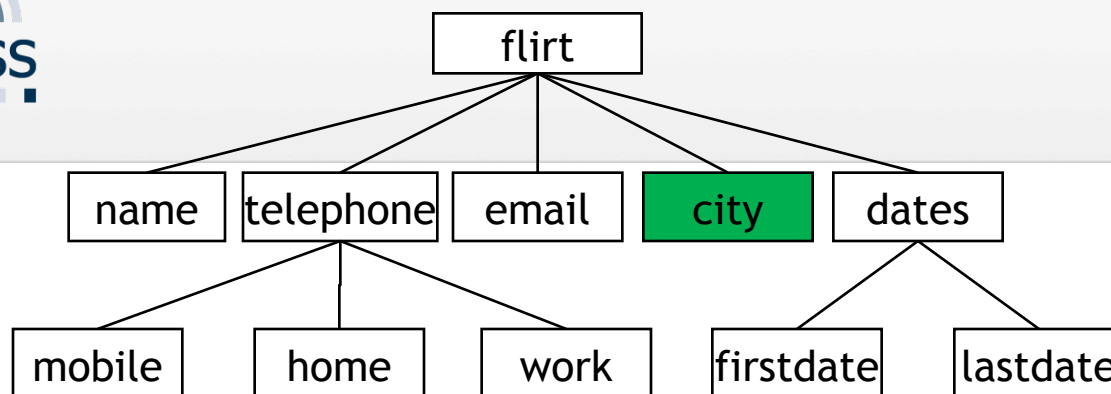
Rule declaration for the elements in a DTD:

```
<!DOCTYPE flirt [  
<!ELEMENT flirt (name, telephone, email, city, dates)>  
<!ELEMENT name      (#PCDATA)>  
<!ELEMENT telephone (mobile | home | work)>  
<!ELEMENT mobile     (#PCDATA)>  
<!ELEMENT home       (#PCDATA)>  
<!ELEMENT work       (#PCDATA)>  
>
```



Rule declaration for the elements in a DTD:

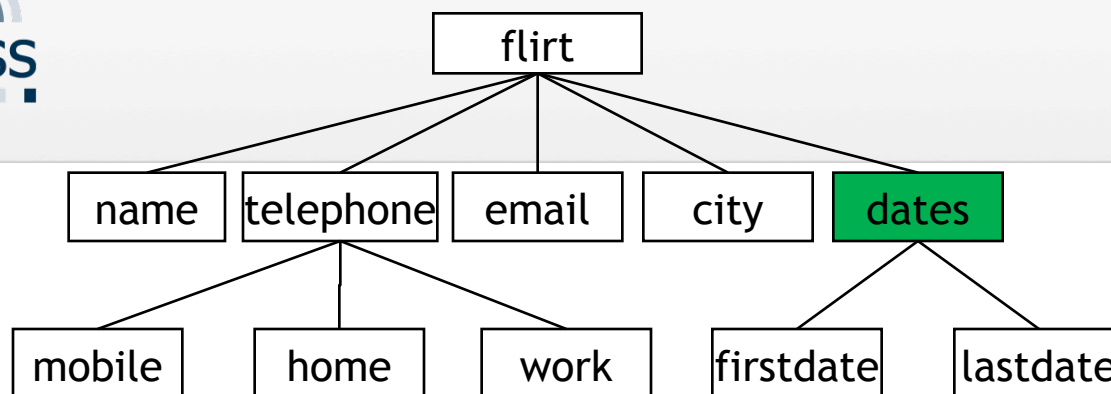
```
<!DOCTYPE flirt [  
<!ELEMENT flirt (name, telephone, email, city, dates)>  
<!ELEMENT name      (#PCDATA)>  
<!ELEMENT telephone (mobile | home | work)>  
<!ELEMENT mobile     (#PCDATA)>  
<!ELEMENT home       (#PCDATA)>  
<!ELEMENT work       (#PCDATA)>  
<!ELEMENT email      (#PCDATA)>  
>
```



Rule declaration for the elements in a DTD:

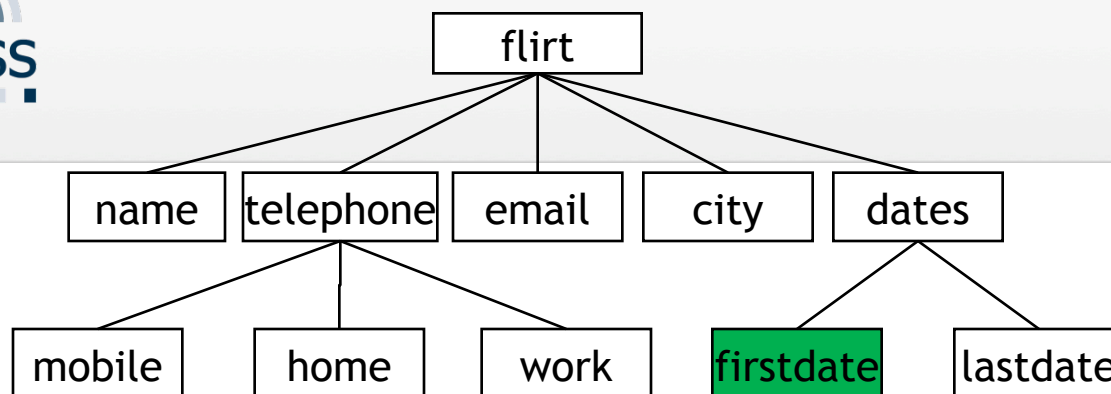
```

<!DOCTYPE flirt [
  <!ELEMENT flirt (name, telephone, email, city, dates)>
  <!ELEMENT name      (#PCDATA)>
  <!ELEMENT telephone (mobile | home | work)>
  <!ELEMENT mobile     (#PCDATA)>
  <!ELEMENT home       (#PCDATA)>
  <!ELEMENT work       (#PCDATA)>
  <!ELEMENT email      (#PCDATA)>
  <!ELEMENT city       (#PCDATA)>
]>
  
```



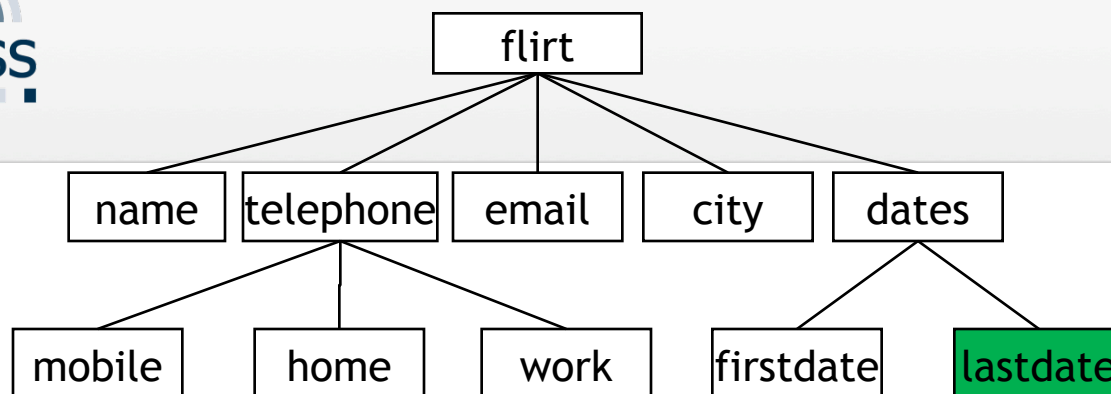
Rule declaration for the elements in a DTD:

```
<!DOCTYPE flirt [  
<!ELEMENT flirt (name, telephone, email, city, dates)>  
<!ELEMENT name      (#PCDATA)>  
<!ELEMENT telephone (mobile | home | work)>  
<!ELEMENT mobile     (#PCDATA)>  
<!ELEMENT home       (#PCDATA)>  
<!ELEMENT work       (#PCDATA)>  
<!ELEMENT email      (#PCDATA)>  
<!ELEMENT city       (#PCDATA)>  
<!ELEMENT dates      (firstdate, lastdate)>  
>
```



Rule declaration for the elements in a DTD:

```
<!DOCTYPE flirt [  
<!ELEMENT flirt (name, telephone, email, city, dates)>  
<!ELEMENT name      (#PCDATA)>  
<!ELEMENT telephone (mobile | home | work)>  
<!ELEMENT mobile     (#PCDATA)>  
<!ELEMENT home       (#PCDATA)>  
<!ELEMENT work       (#PCDATA)>  
<!ELEMENT email      (#PCDATA)>  
<!ELEMENT city       (#PCDATA)>  
<!ELEMENT dates      (firstdate, lastdate)>  
<!ELEMENT firstdate (#PCDATA)>  
>
```



Rule declaration for the elements in a DTD:

```
<!DOCTYPE flirt [  
<!ELEMENT flirt (name, telephone, email, city, dates)>  
<!ELEMENT name      (#PCDATA)>  
<!ELEMENT telephone (mobile | home | work)>  
<!ELEMENT mobile     (#PCDATA)>  
<!ELEMENT home       (#PCDATA)>  
<!ELEMENT work       (#PCDATA)>  
<!ELEMENT email      (#PCDATA)>  
<!ELEMENT city       (#PCDATA)>  
<!ELEMENT dates      (firstdate, lastdate)>  
<!ELEMENT firstdate  (#PCDATA)>  
<!ELEMENT lastdate   (#PCDATA)>  
>
```

- *“Das Wurzelement in Übung 5 Aufgabe 2a ist doch User, wieso wird in der XML Datei nicht erst <User> aufgeführt, und dann <User id="194"> ?”*

```
<?xml version="1.0"?>
<User id="194">
  <Pseudonym>
    Jenny23
  </Pseudonym>
  <Mobile_Operator>
    t-mobile
  </Mobile_Operator>
  <Registration>
    03.02.2020
  </Registration>
  <Lastlogin>
    29.04.2020
  </Lastlogin>
</User>
```

```
<!DOCTYPE User [
  <!ELEMENT User
    (Pseudonym,Mobile_Operator,
    Registration,Lastlogin)>

  <!ELEMENT Pseudonym (#PCDATA)>
  <!ELEMENT Mobile_Operator (#PCDATA)>
  <!ELEMENT Registration(#PCDATA)>
  <!ELEMENT Lastlogin(#PCDATA)>
]>
```