Chair of Mobile Business & Multilateral Security

Lecture 10

Business Informatics 2 (PWIN) WS 2023/24

ICS Development III

Markup Languages

Prof. Dr. Kai Rannenberg

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

Agenda

- From HTML to XML
- XML Concepts
- Processing of XML Documents
- XML Example Applications

Overview of HTML

- HTML is a mark-up language for describing, structuring and presenting contents such as text, pictures, video, hyperlinks, etc.
- Originally developed by W3C, moved to a continuously updated and improved Living Standard maintained by the WHATWG in collaboration with W3C in 2019.
- In former times mainly used to deliver and present static contents of service providers (news providers, enterprises, government, personal websites, etc.)
- …

HTML Example

```
<html>
    <head>
        <title>M-Chair Website</title>
    </head>
    <body>
        <h1>Chair of Mobile Business & Multilateral Security</h1>
        <h2>Theodor-W.-Adorno Platz 4</h2>
        <h3>60623 Frankfurt am Main</h3>
    </body>
</html>
```

mobile business

Issues of HTML

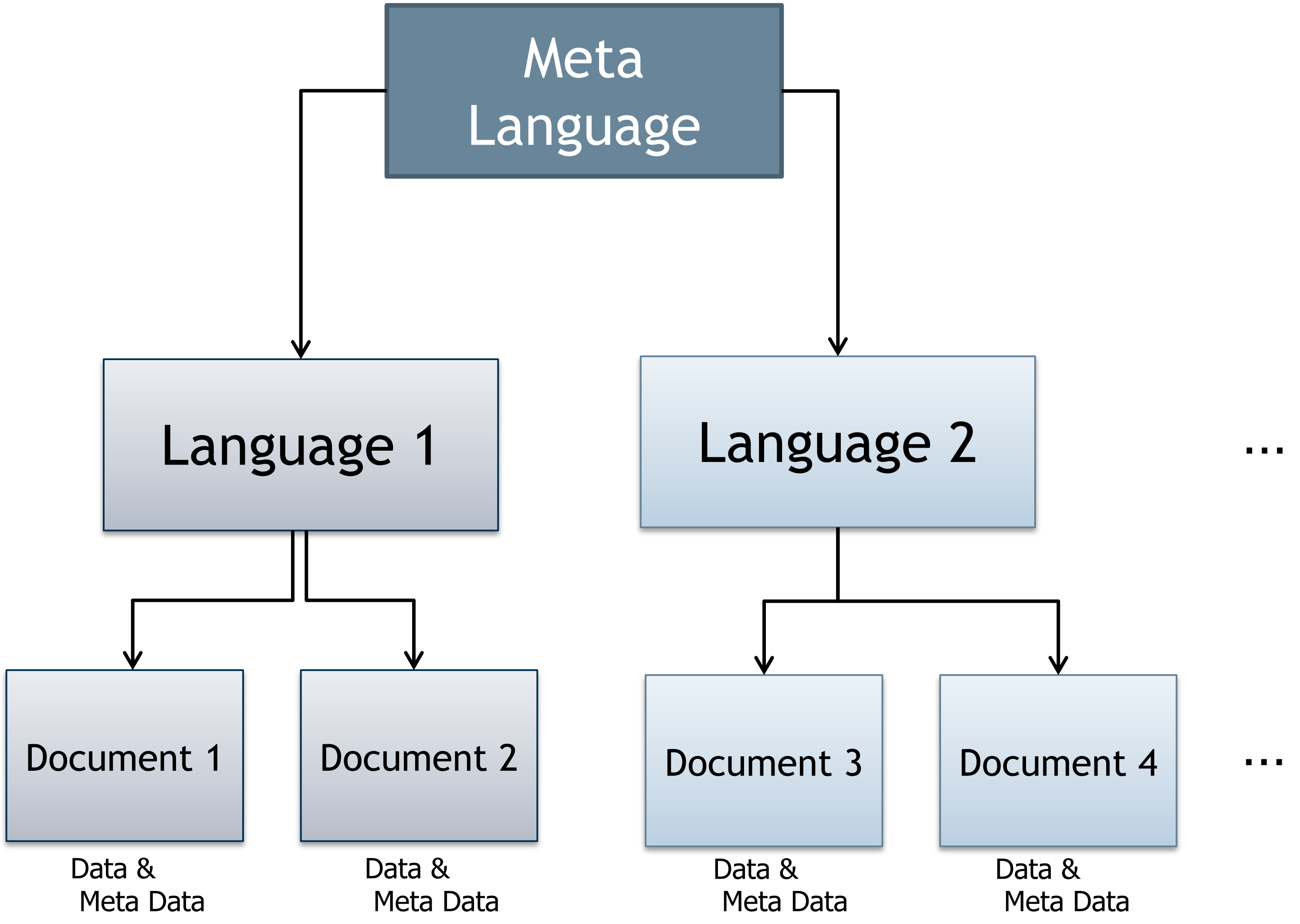
- The Hypertext Markup Language (HTML) is a very simple description language for contents:
  - Hardly any semantic descriptions for content
  - Mainly structural and layout information such as sections, headlines, lists, etc. exist.
- So, how can, for instance, a postal address in HTML be recognised and processed by a software system on a website?

```
<h1>Chair of Mobile Business & Multilateral Security</h1>
<h2>Theodor-W.-Adorno Platz 4</h2>
<h3>60623 Frankfurt am Main</h3>
```

Meta Languages

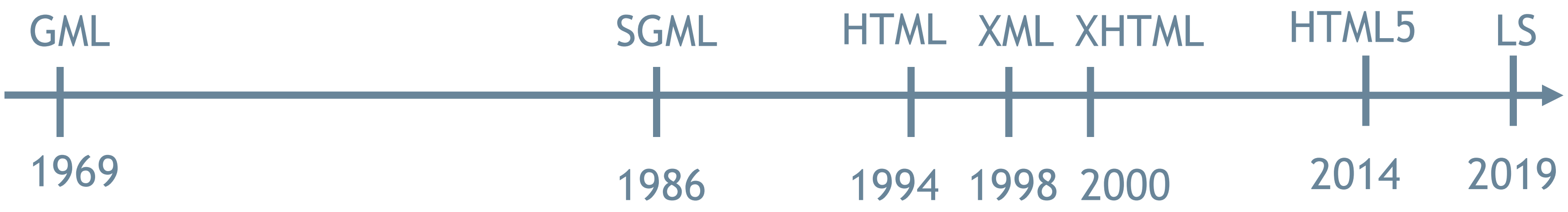
- Describing data requires a formal markup language (consisting of a vocabulary and grammar rules).
- HTML is a formal markup language but is targeted towards structuring and presenting data rather describing it.
- A language describing data always has to be domain specific (e.g. law vs. economics; business vs. private). Consequently, a meta (markup) language is required.
- A meta language provides a vocabulary and grammar rules for specifying application domain specific languages (without being a specific language on its own).

Meta Language



Markup Language History

Development of markup languages for data description



GML:	Generalized Markup Language by IBM
SGML:	Standard Generalized Markup Language as standard ISO 8879 for data exchange and storage
HTML:	Definition of version 2 as SGML dialect
XML:	Links HTML with the claim of SGML: Extensible Markup Language
XHTML:	HTML based on XML
HTML5:	Redefinition of HTML for browsing without plugins
LS:	Living Standard, new functionality to be added continuously

Standard Generalized Markup Languages

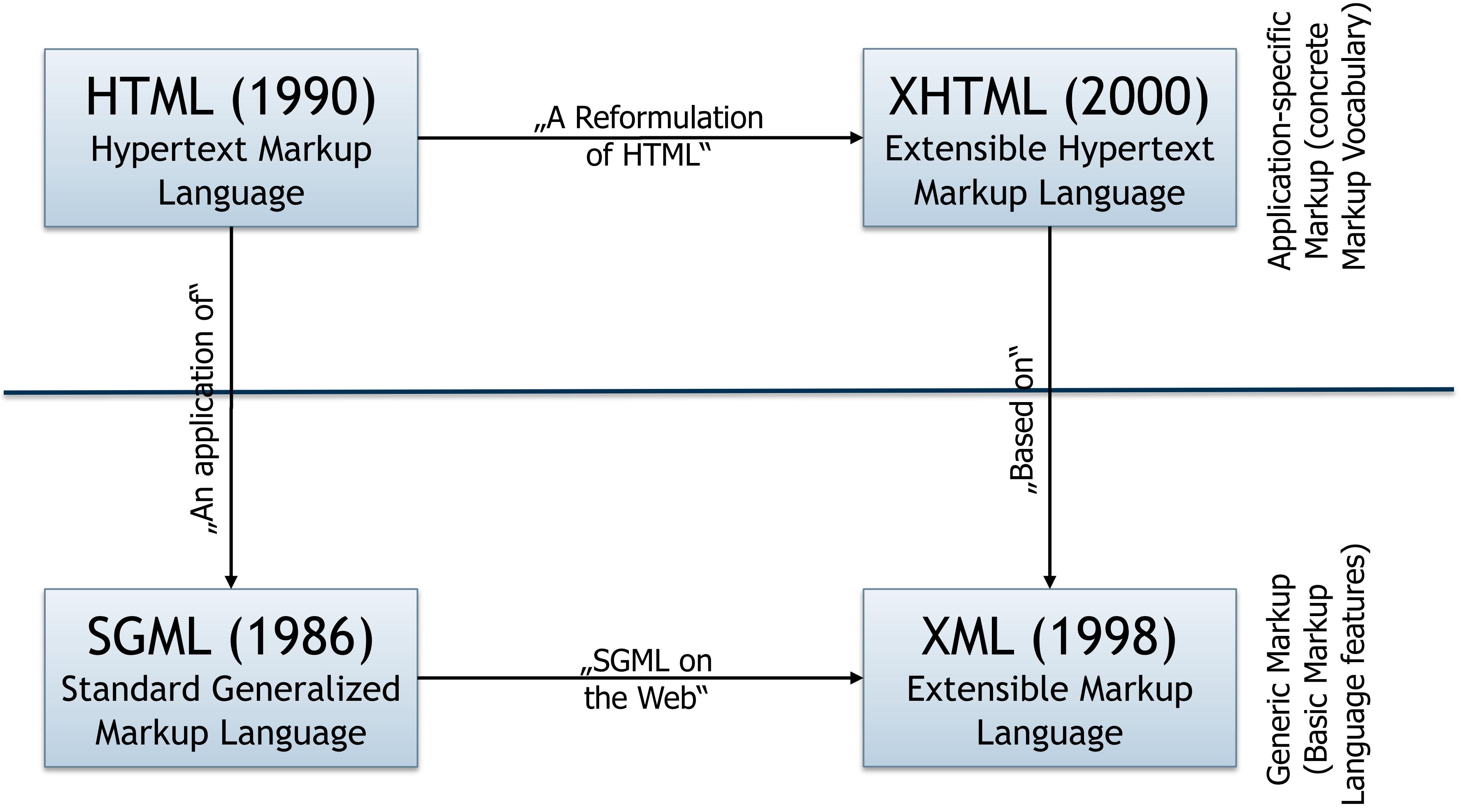
- Basic idea of all Standard Generalized Markup Languages (SGMLs)
  - Create processable documents by adding information about structure and content
  - Establish a system und manufacturer independent standard
  - Separate structure, content and presentation of a document
  - A meta language from which concrete languages (e.g. HTML) can be specified

- Popular SGML dialects
  - LaTeX
  - Postscript

Extensible Markup Language (XML)

- Light subset of SGML, carrying only the most relevant language features
- Standardised
- Self-describing thanks to included meta information
- Extendable with new elements -> creation of application specific models
- Suitable for data storage
- Simple and easy to read for humans (not binary)

SGML, XML, HTML and XHTML

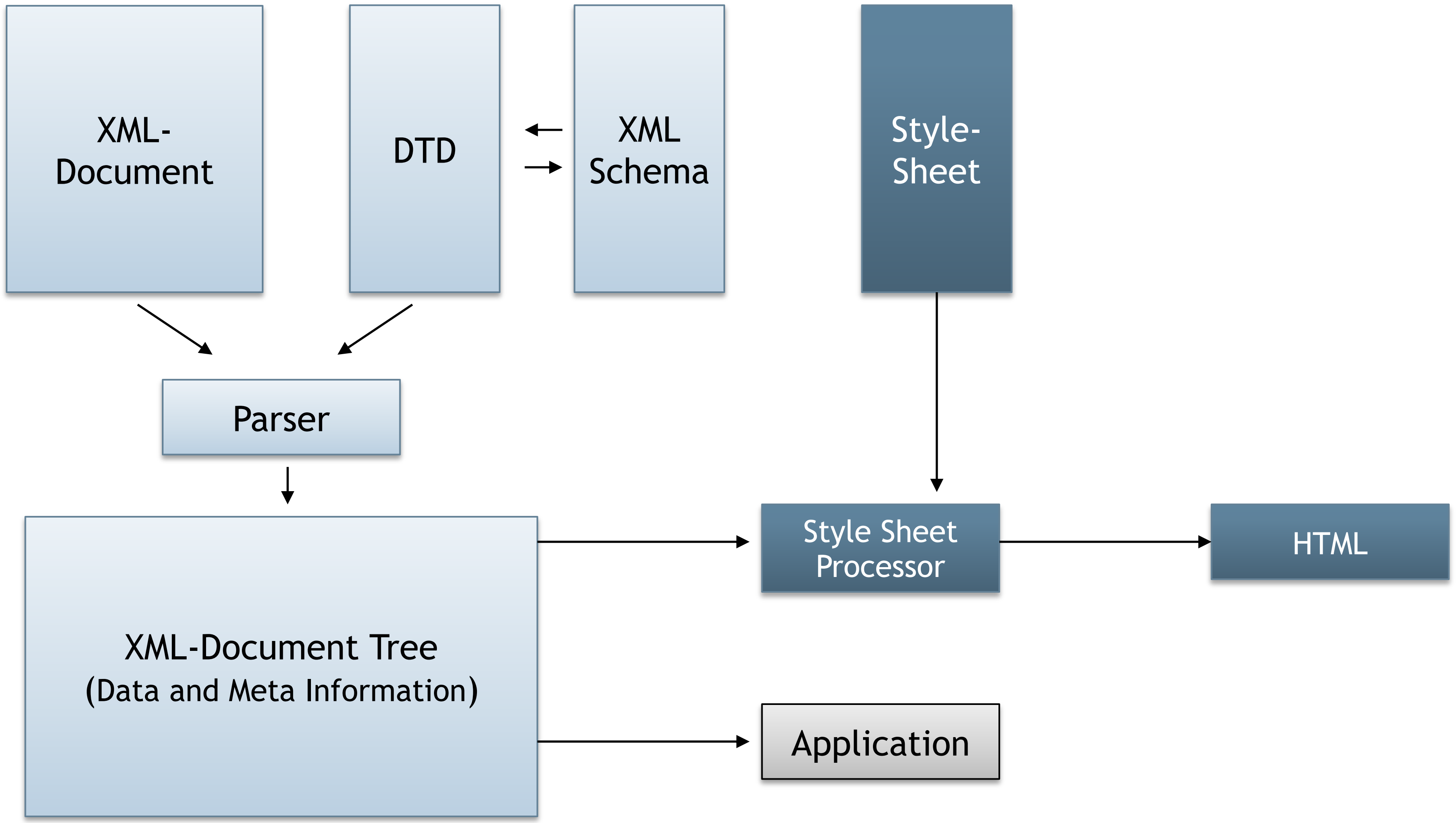

HTML (1990)
Hypertext Markup Language
„A Reformulation of HTML"
XHTML (2000)
Extensible Hypertext Markup Language
Application-specific Markup (concrete Markup Vocabulary)
„An application of"
„Based on"
SGML (1986)
Standard Generalized Markup Language
„SGML on the Web"
XML (1998)
Extensible Markup Language
Generic Markup (Basic Markup Language features)

Based on: Erik Wilde (2008), http://dret.net/lectures/web-spring11/html

Relevant XML Terms

- **DTD**
  Document Type Definition – describes the structure of an XML document and defines its *grammar*.
- **XML Schema**
  Alternative approach to DTD with additional features
- **Parser**
  Translates an XML document in a document tree while making is elements accessible for applications
- **Style Sheet**
  Layout information for rendering the XML documents contents
- **Style-Sheet-Processor**
  Implements the style information and generates the result pages

Processing and Usage of XML Documents



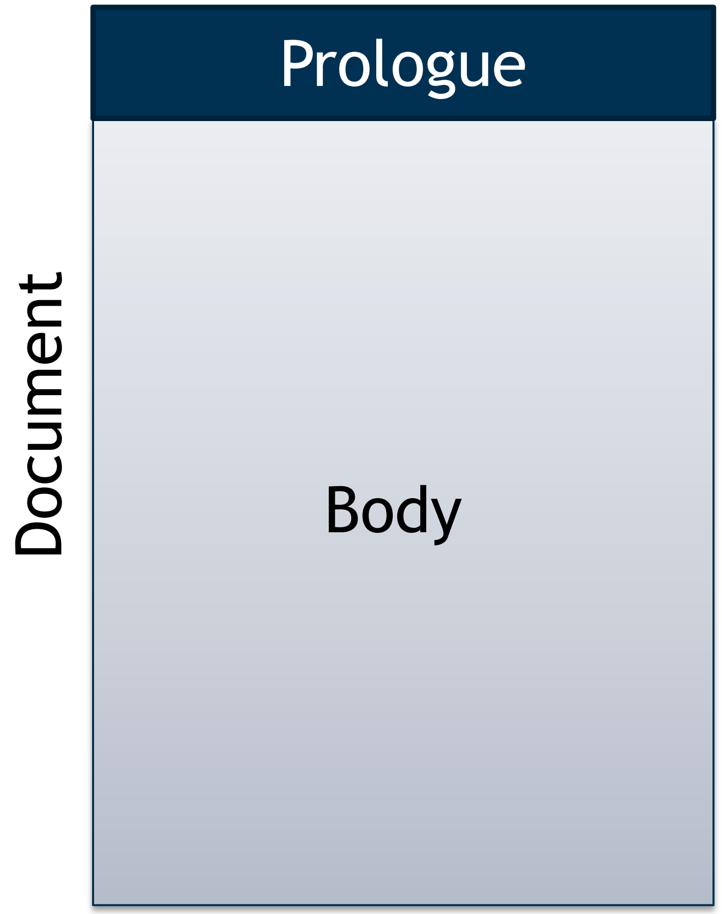
Some general XML Applications

- Sharing of data between different components of an application (e.g. Microsoft Excel / Access)
- Storage of application data in plain, non-binary text files (e.g. Microsoft Word Format)
- Advancing Electronic Data Exchange (EDI):
  - Transactions between banks
  - Producers and suppliers sharing product data
- User generated content (e.g. Google Maps layers)
- Access to services and applications via the Internet (e.g. Web Service APIs)

Agenda

- From HTML to XML
- XML Concepts
- Processing of XML Documents
- XML Example Applications

XML Document Structure



Prologue contains the XML version and information about the used character encoding.

Body contains data

Example: Manage Dates via XML

```
<?xml version=„1.0“ encoding=„ISO-8859-1“ ?>
```

Prologue

```
<flirt>
    <name>Daisy</name>
    <mobile>+436508469249</mobile>
    <email>daisy@m-chair.net</email>
    <city>Innsbruck</city>
    <first date>2013-01-23</first date>
    <last date>2013-05-01</last date>
    <birthday>1983-11-13</birthday>
    <vegetarian>no</vegetarian>
    <status>single</status>
</flirt>
```

Body

mobile business

XML Syntax

- XML expects closed elements!
  - <name> is a tag
  - Syntax: <StartTag>content</EndTag>
  - Start tags must correspond to end tags, and vice versa
  - <name>Daisy</name>

- Attributes are included in the start tag:
  - <city residence=„first“>Innsbruck</city>

XML Syntax

- An **element**: Everything between two tags; for instance
  - `<title>Complete Guide to DB2</title>`

- Elements may be **nested**; for instance

  -
    ```
    <book>
        <title>Complete Guide to DB2</title>
        <author>Chamberlin</author>
    </book>
    ```

- Empty element
  - `<red></red>`
  - abbreviated `<red/>`

- An XML document has a unique **root element.**

Well-formed XML Documents

- An XML document is **well-formed**, if
  - It only contains properly encoded legal Unicode characters.
  - None of the special syntax characters such as "<" and "&" appears “un-escaped” in the data.
  - The begin, end, and empty-element tags, which delimit the elements, are correctly nested, whereas none is missing or overlapping.
  - The element tags are case-sensitive; the beginning and end tags must match exactly.
  - There is a single "root" element which contains all the other elements.

mobile business

Wrong XML Syntax

- This type of nesting is not allowed:

```
<telephone>
    <mobile>+4916008154712
    <home>+4972138488551
    </mobile></home>
</telephone>
```

- It cannot be determined if a number belongs to <mobile> or <home>.

- The document is not well-formed. This can be automatically detected by a parser software.

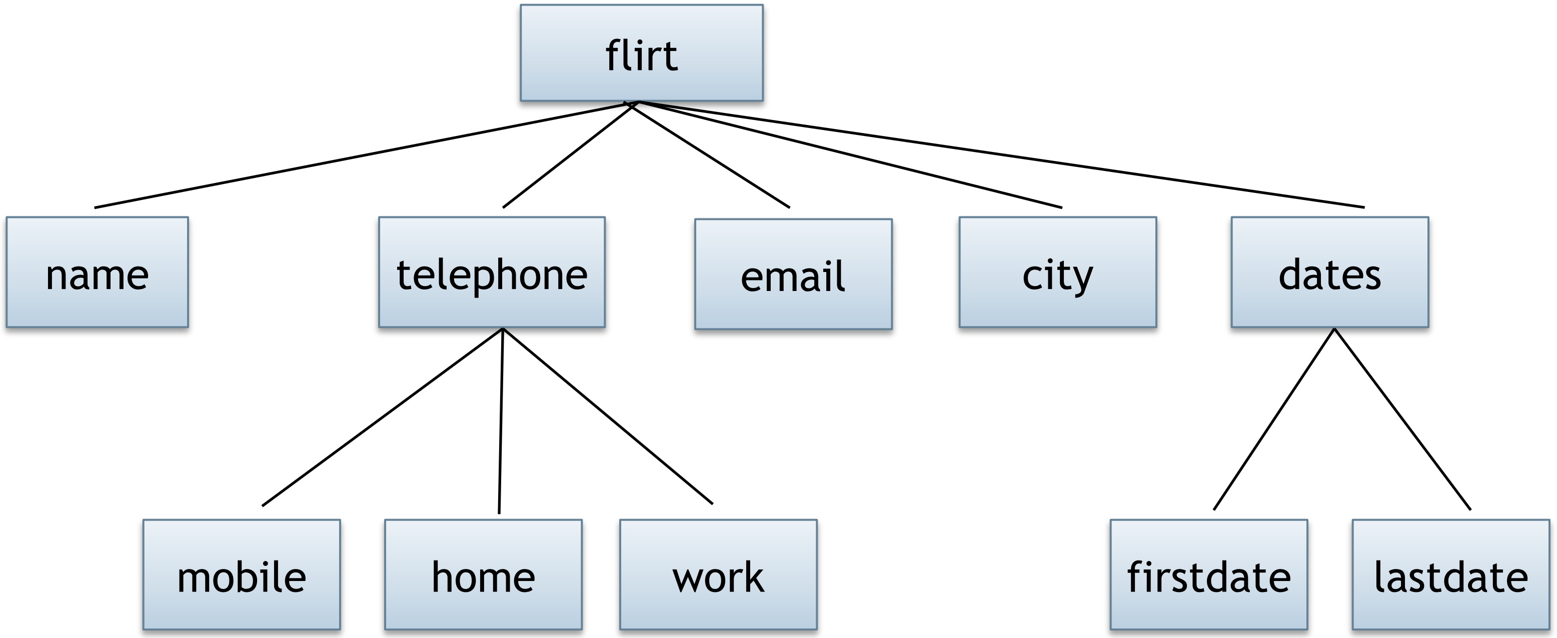
XML Syntax

- As in HTML, some characters are used for the syntax:

Character	notation
<	&lt;
>	&gt;
&	&amp;
'	&apos;
"	&quot;

XML Document Tree

- XML document tags can also be considered as objects in an object-oriented database or a tree (document tree):



- Because of the distinct, tree-like structure and similarity to object-oriented systems, computers are able to unambiguously recognise the data structure when reading an XML document.

XML
Document Type Definition (DTD)

- The Document Type Definition (DTD) describes the structure of a document and defines a *grammar* for the XML document.
- Comparable to a type or variable declaration in a programming language.
- The DTD defines which elements and references may appear in the document based on it.
- The DTD also declares entities that are allowed to be used in the XML document.

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

XML DTD

- Rule declaration for the elements in a DTD:

```
<!ELEMENT flirt         (name, telephone, email, city, dates)>
<!ELEMENT name          (#PCDATA)>                 Text
<!ELEMENT telephone     (mobile | home | work)+>
<!ELEMENT mobile        (#PCDATA)>
<!ELEMENT home          (#PCDATA)>
<!ELEMENT work          (#PCDATA)>
<!ELEMENT email         (#PCDATA)>
<!ELEMENT city          (#PCDATA)>
<!ELEMENT dates         (firstdate, lastdate)>     Selection list
<!ELEMENT firstdate     (#PCDATA)>
<!ELEMENT lastdate      (#PCDATA)>
```

Valid XML Document

- An XML document, which complies with a DTD is called „valid“.
- The validity of an XML document can be automatically determined by a parser software.
- This concept allows consumers of XML documents (e.g. a software application) to verify that the XML documents contents comply with their expected document format
  - Specified document structure
  - Allowed elements and data
  - …

XML Schema

- “XML Schema” is an alternative to the DTD.

- XML schema eliminates some of the DTD weaknesses by adding the following features:
  - Better content modelling for syntax check
  - Order and nesting are configurable.
  - Configurable value margins
  - Verification of element data types
  - Better definition of the cardinalities with Min. and Max.
  - Greater choice of data types in analogy to programming languages and databases (e.g. boolean, number, float, date time, ...)

XML Namespaces

- XML documents are especially beneficial if data is shared across applications, between users or even across independent enterprises.
- How can tag mix-ups be prevented, if data from different sources with identical tag names is merged?

```
<Book>
    <Title>Computer Networks</Title>
            …
</Book>

<Author>
    <Title>Professor</Title>
            …
</Author>
```

mobile business

XML Namespaces

- Idea: A Universal Resource Identifier (URI), which allows the introduction of a namespace defined by a globally unique path.
- For this, a prefix for an element is created.

```
<book
  xmlns:book="http://www.amazonen.de/namespaces/books"
  xmlns:aut="http://www.amazonen.de/namespaces/authors"
>
        <book:Title>Networks</book:Title>
</book>
```

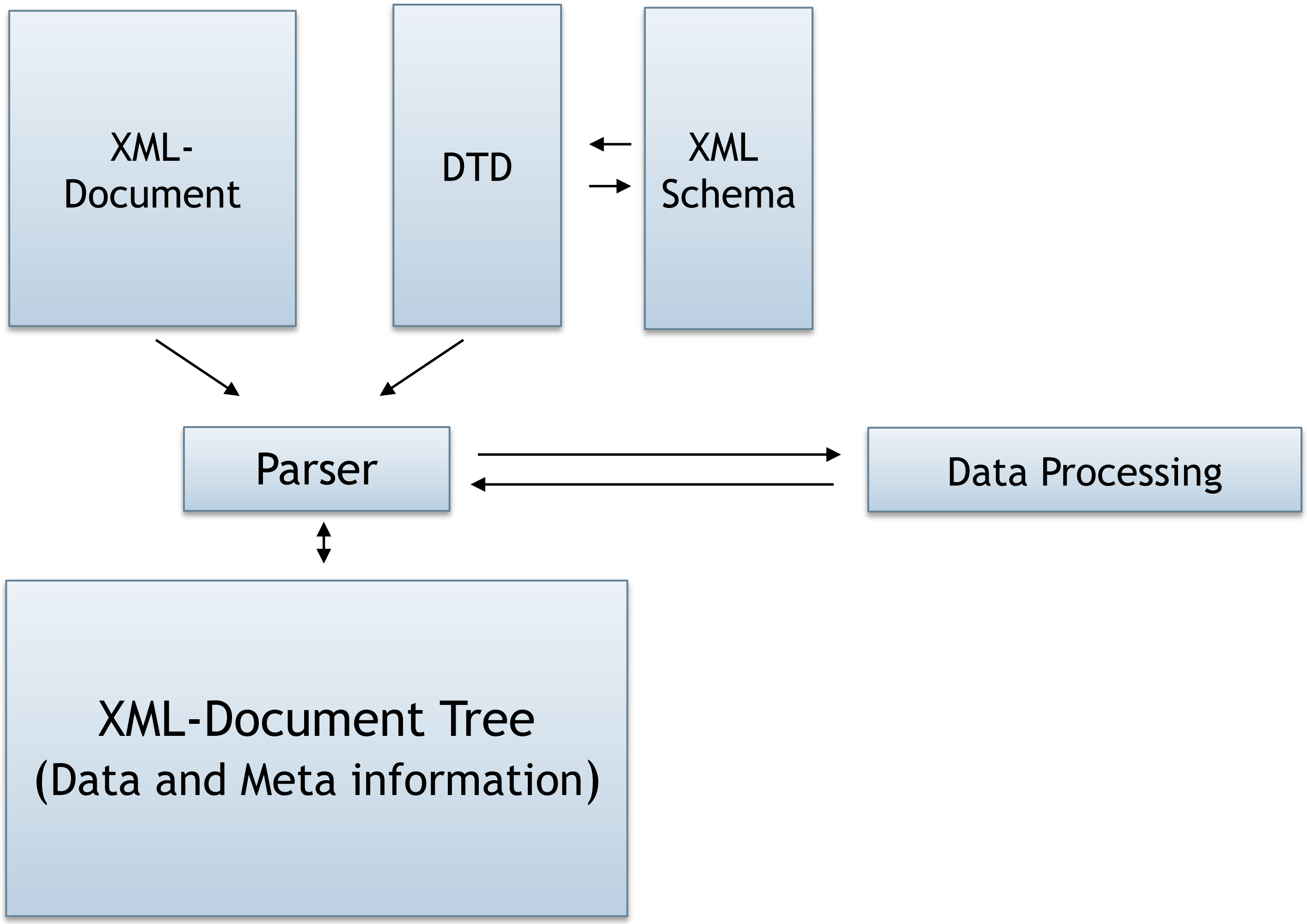
Agenda

- From HTML to XML
- XML Concepts
- Processing of XML Documents
- XML Example Applications

Processing XML Documents

- Processing an XML document requires a parser
- A parser is a software that reads DTDs, schemas and XML documents and enables an application to access all of the XML document elements.
- General parsing process
  1. An application (e.g. Microsoft Word) opens an XML document.
  2. The parser reads the XML document and the corresponding DTDs, schemas.
  3. The parser checks if the XML document is well-formed and valid.
  4. Parser offers an application interface with functions like „ListElements()“.
  5. The application accesses the elements of the XML document using the available interfaces, and processes the received data.
  6. The application saves the modified/updated XML document.

Processing of XML Documents



XML Document Parsers

- There are two types of parsers:
  - Document Object Model (DOM)
  - Simple API for XML (SAX)

- **DOM type parsers** load all elements in the memory and create a tree data structure, which can be then processed.

- **SAX type Parsers** navigate through a document offering only parts of its contents without loading it completely into memory.

XML Document Parsers

- Comparison of DOM and SAX type parsers
  - SAX is able to parse files of any size.
  - SAX is efficient, if only parts of the file are relevant.
  - SAX is easy to use.

  - DOM allows free access and changes to a document.
  - DOM creates a full image of the document in memory.

mobile business

XML Document Parsers

Typical application of DOM und SAX parsers

- **DOM parsers** are useful when editing entire documents at once. For instance, for editing a structured text in a word processor.

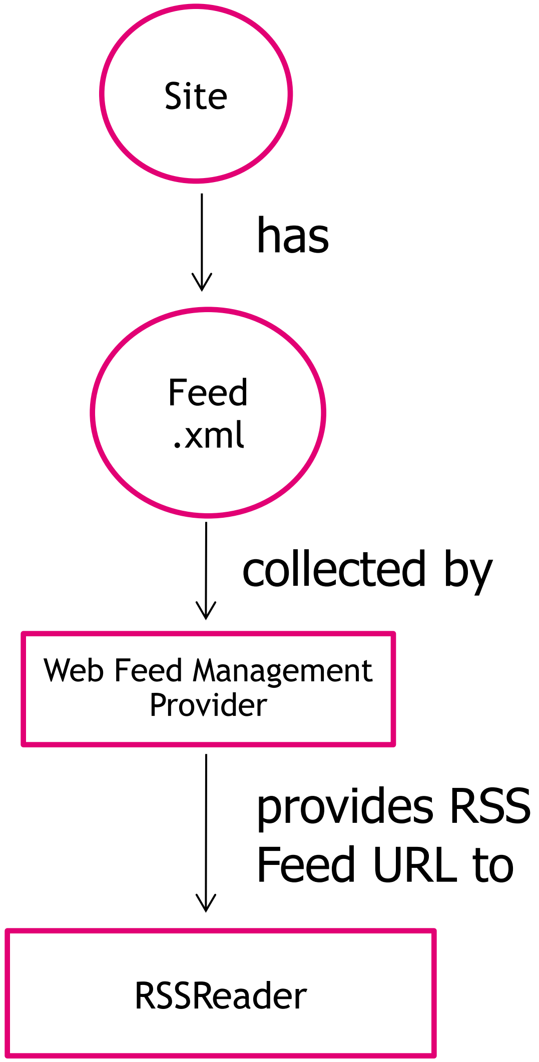
- **SAX parsers** are useful for quick retrieval of records, e.g. for accessing addresses in an XML-based customer database.

Agenda

- From HTML to XML
- XML Concepts
- Processing of XML Documents
- XML Example Applications

Really Simple Syndication (RSS)

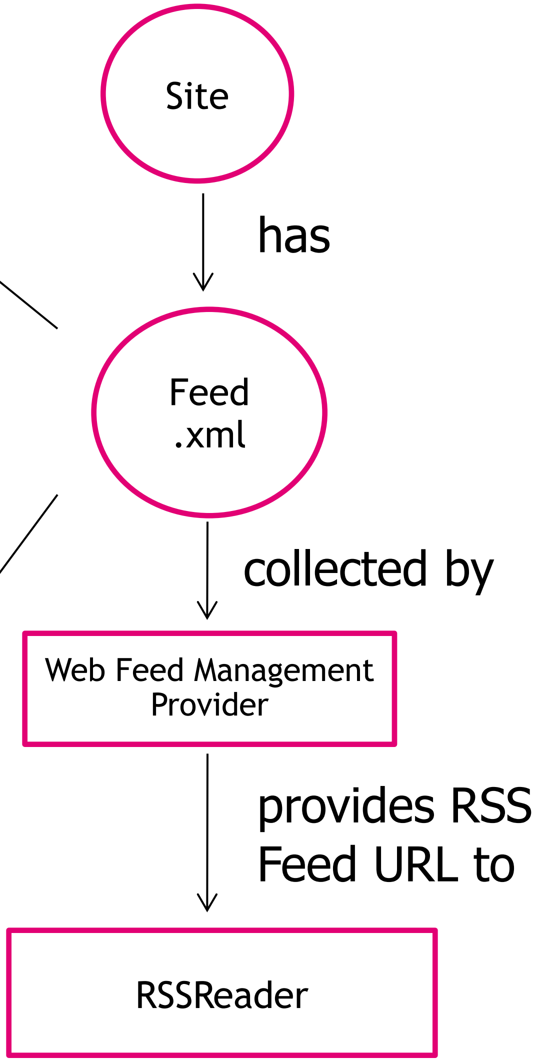
- RSS is a web content syndication format.
- RSS is a dialect of XML: All RSS files must conform to the XML 1.0 specification, as published on the World Wide Web Consortium (W3C) website.



Source: http://www.rssboard.org/rss-specification#whatIsRss

RSS Feed Example

```
<?xml version="1.0" encoding="utf-8"?>
<rss version="2.0">
 <channel>
      <title>Titel des Feeds</title>
      <link>URL der Webpräsenz</link>
      <description>Kurze Beschreibung des Feeds</description>
      <language>Sprache des Feeds (z. B. "de-de")</language>
      <copyright>Autor des Feeds</copyright>
      <pubDate>Erstellungsdatum("Tue, 8 Jul 2008 2:43:19")</pubDate>
      …
      <item>
              <title>Titel des Eintrags</title>
              <description>Kurze Zusammenfassung des Eintrags</description>
              <link>Link zum vollständigen Eintrag</link>
              <author>Autor des Artikels, E-Mail-Adresse</author>
              <guid>Eindeutige Identifikation des Eintrages</guid>
              <pubDate>Datum des Items</pubDate>
      </item>
 </channel>
</rss>
```



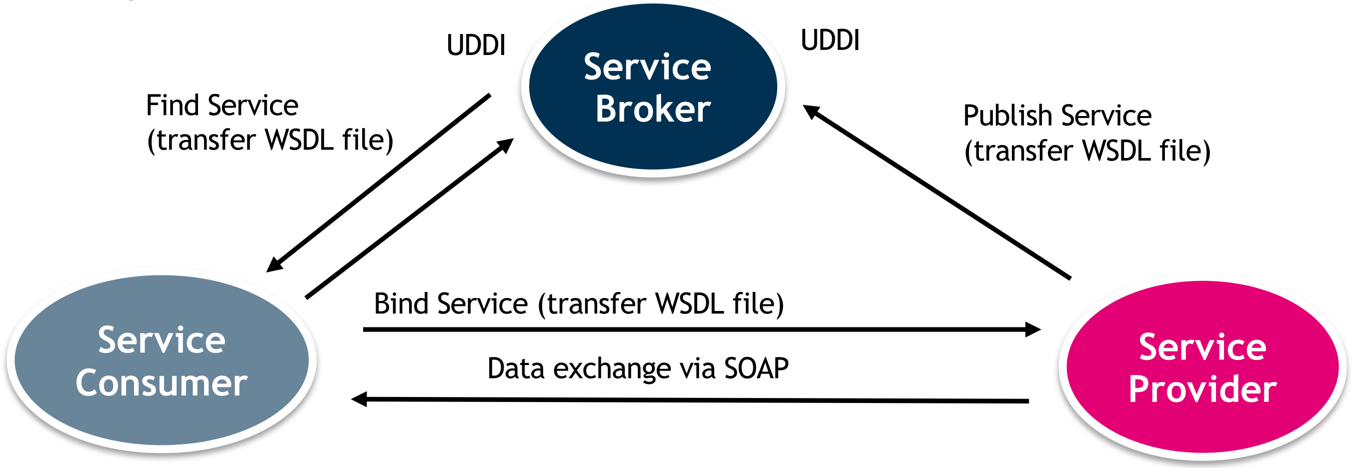
Source: http://de.wikipedia.org/wiki/RSS

XML Web Services

- The term *Web Service* describes a standardised way of integrating Web-based applications using XML, SOAP, WSDL, and UDDI over an Internet protocol backbone.

Source: www.webopedia.com/TERM/W/Web_services.html

  - XML is used to tag the data,
  - SOAP (Simple Object Access Protocol) is used to transfer the data.
  - WSDL (Web Services Description Language) is used for describing the services available.
  - UDDI (Universal Description, Discovery and Integration) is used for listing what services are available.



UDDI
Service Broker
UDDI
Find Service (transfer WSDL file)
Publish Service (transfer WSDL file)
Bind Service (transfer WSDL file)
Data exchange via SOAP
Service Consumer
Service Provider

Further XML Applications

- DOCX file format: MS Word file format
- ODF format: Open Document Format for office applications
- XML/EDIFACT: XML/EDIFACT is an Electronic Data Interchange (EDI) format which is used in business-to-business transactions.
- OFX: Open Financial Exchange for finance information (www.ifxforum.org)
- MathML: Mathematical formula description language (www.w3.org/Math)
- SAML: Security Assertion Markup Language for exchanging authentication and authorisation information (www.oasis-open.org)
- EPAL: Enterprise Privacy Authorisation Language is a formal language to specify fine-grained enterprise privacy policies (www.zurich.ibm.com/security/enterprise-privacy/epal/)
- ...

mobile business

Literature

- Tim Berners-Lee (2000), W3C Talk, Internet: www.w3.org/2000/Talks/1206-xml2k-tbl/slide10-0.html
- Webopedia, Internet: www.webopedia.com/TERM/W/Web_services.html
- Erik Wilde (2011), Web Architecture, Fall 2011 — INFO 153 (CCN 42509). http://dret.net/lectures/web-spring11/html