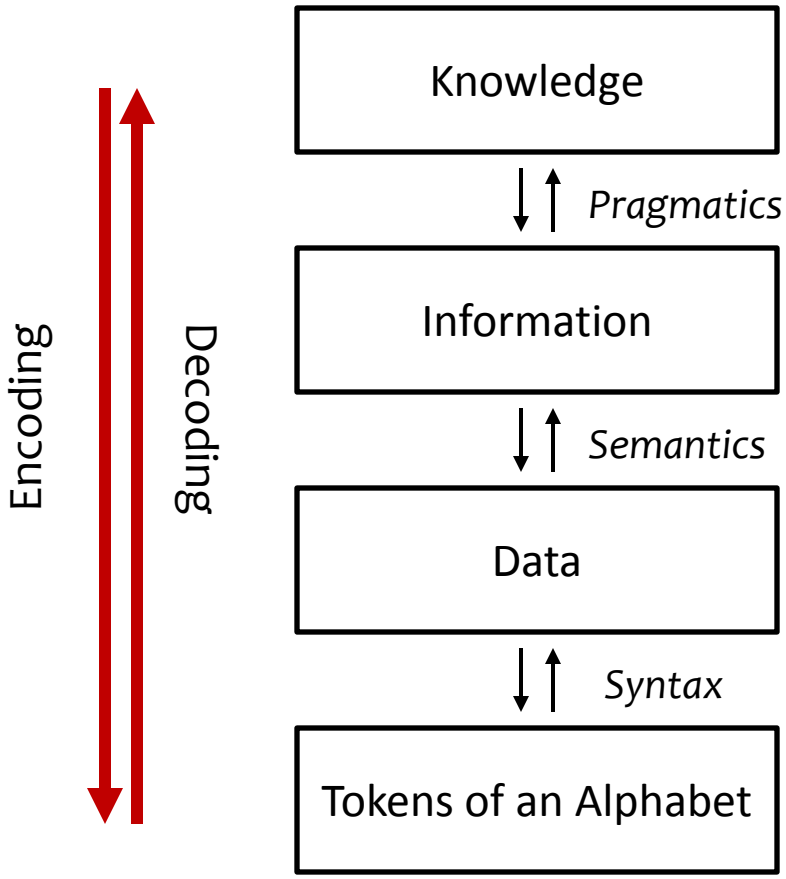
2. Background on Data Management

Aspects of Data Management and an Overview of Solutions used in Engineering Applications

Overview

- Basic Terms
  - What is data, information, data management, a data model, a schema, metadata, etc.?
- Aspects of Data Management
  - What is the most relevant functionality required for engineering data?
- Data Management Approaches
  - Files and File Systems
  - Database Management Systems

Terms: Data and Information



Information represent message or observation transmitted as some form of matter or energy.

Data represent encoded information with a fixed syntax (structure) and meaning (semantics) usable by computers.

Term: Data Management

- Term **Data** often used referring to static aspect of describing and storing facts
- Term **Information** more often refers to dynamic aspects such as transmission and communication
- Within the scope of this lecture focus on data and how it is managed

Data Management refers to activities and concepts to store, retrieve, manipulate, exchange, and data according to a fixed syntax and semantics.

Term: Data Model

A **data model** is a model that describes in an abstract way how data is represented in an information system or a database management system.

- A data model defines syntax and semantics, i.e.
  - How can data be structured (syntax)
  - What does this structure mean (semantics)
- Very generic term for many applications
  - Programming languages have their data models (e.g. C++ and Java have object-oriented data models)
  - Conceptual design methods (e.g. ER, UML) represent a data model
  - File formats either apply a data model (e.g. XML) or implement their own
  - Database management systems implement data(base) models

Term: Schema

A **database schema** is a map of concepts and their relationships for a specific universe of discourse. It describes the structure of a database.

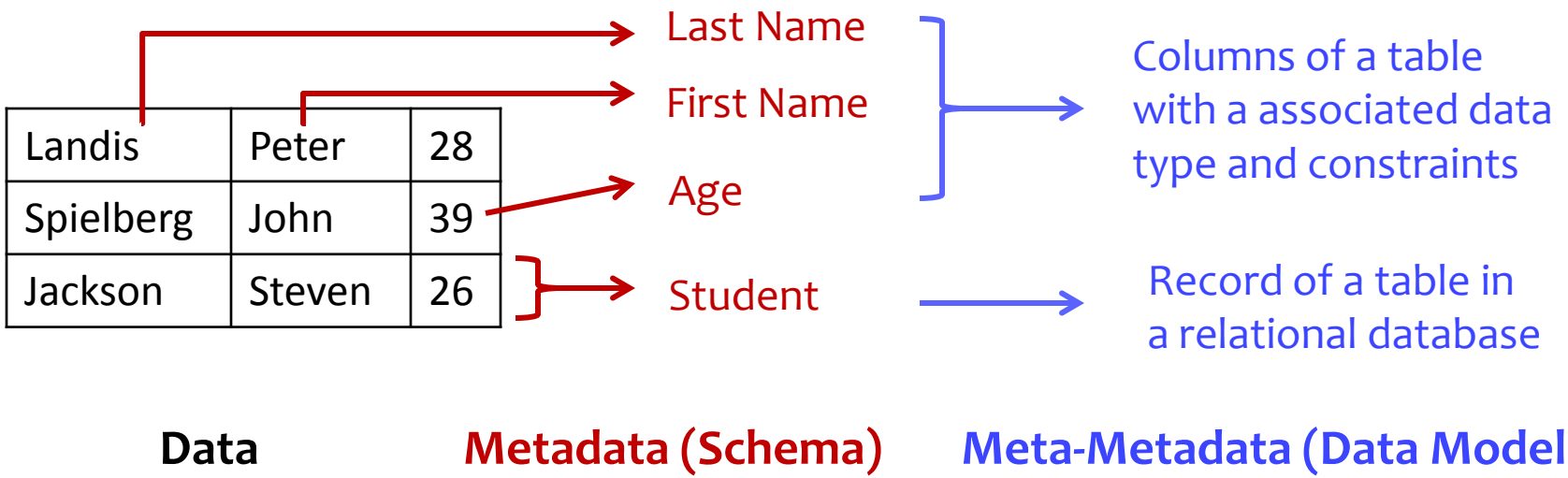
- Specific for a given application
- During information design design two main types
  - **Conceptual schema** represents information on an implementation-independent level, often using graphical notations
  - **Logical schema** represents information according to data model of implementation language or system
- Often erroneously confused with Data Model

Data Model vs. Schema

Data Model	Data Schema
Independent of application	Application specific
Describes concepts of the schema	Describes concepts of the application
“Language” used to describe data	Description of data
Meta-metadata	Metadata
Property of a programming language, modeling language, database management system, etc.	Property of a program, conceptual diagram, database , etc.

Terms: Data and Metadata

- "Data about data"
- In Engineering
  - Data describing products directly (geometry, product structure, simulation data, etc.)
  - Metadata describing how data is used (users, processes, relations, documents, etc.)
- May be organized in several levels of abstraction (data, metadata, meta-metadata, etc.)



Aspects of Data Management

- Focus on most important aspects for Engineering
  - Persistence
  - Data Integration
  - Independence
  - Security
  - Data Quality
  - Concurrency Control

Persistence

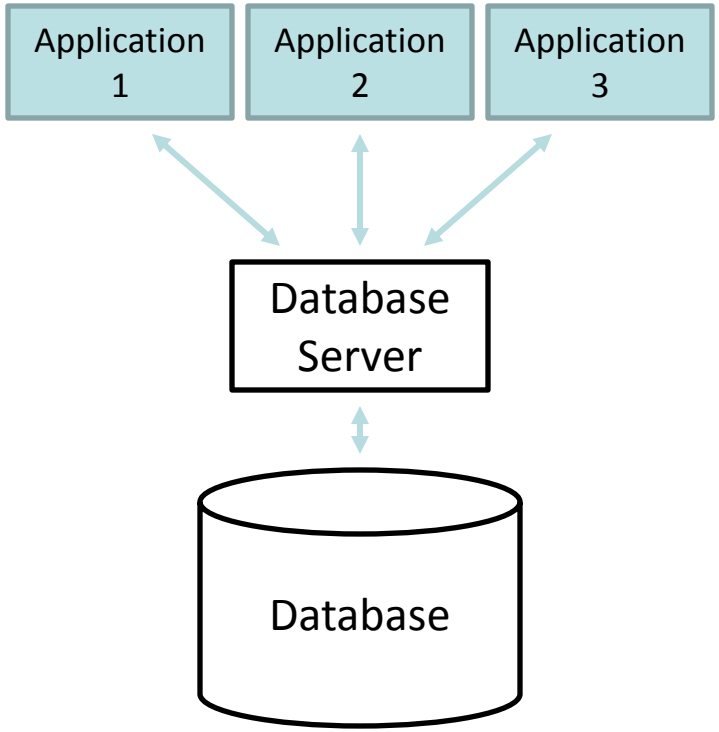
- Term Data Management primarily focuses on mid- and long-term **persistent storage** of data beyond process runtime or system up-time
  - In file systems or databases on secondary storage (hard disk)
  - As backups on tertiary storage (archives on tapes, discs)
- Contrary to **transient storage** in main memory (RAM) or CPU registers or cache
- Persistence mechanisms requires mapping main memory structures (freely accessible address space) to secondary storage structures (e.g. blocks, cylinders, etc. for hard disk)
- **Archiving** on tertiary media
  - To prevent data loss due to common secondary storage failures
  - To free secondary storage resources from data that is not frequently used
  - For long-time documentation purposes required due to legal requirements

Data Integration

- Ideal state of information system: each real-world object is represented in information system exactly once and accessible via a uniform interface
- Goal: avoid redundancy
  - Possible cause of inconsistencies
  - Storage consumption (minor problem)
- Problems
  - In most companies/organizations many information systems are used with overlapping functionality and data
  - Heterogeneity: data maybe represented in different ways (e.g. different data model, different schema, different interfaces)
  - Distribution: data is stored in possibly many departments, subsidiaries, etc.
  - Hard to control within complex information system infrastructures
  - Integration of data in files/file systems requires strong regulations

Data Integration /2

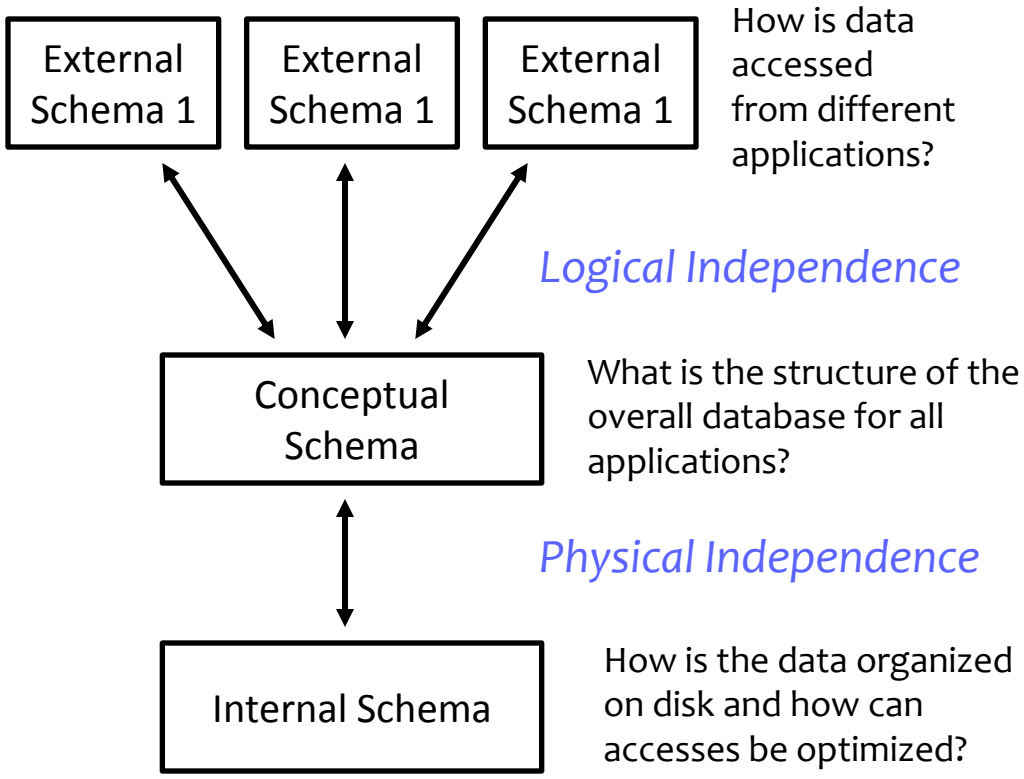
- **Materialized integration:** store integrated data physically exactly once and within one system
  - Databases sometimes used to enforce integration
  - E.g. EDM/PDM system, Warehouse Data
- **Virtual integration:** provide single point of access and reconciliation mechanism for data physically stored in several systems
  - Federated databases and mediators



Typical ideal of a Database System as an integration platform for several applications

Independence

- Application should be independent of how data is stored and accessed
- Decoupling allows flexible changes of storage mechanism (e.g. conceptual changes, optimizations, replacement of entire storage system)
- Achieved by using standards, e.g.
  - XML for content of files
  - Query language SQL fro databases



3-Level Schema Architecture of Database Systems providing independence from storage details

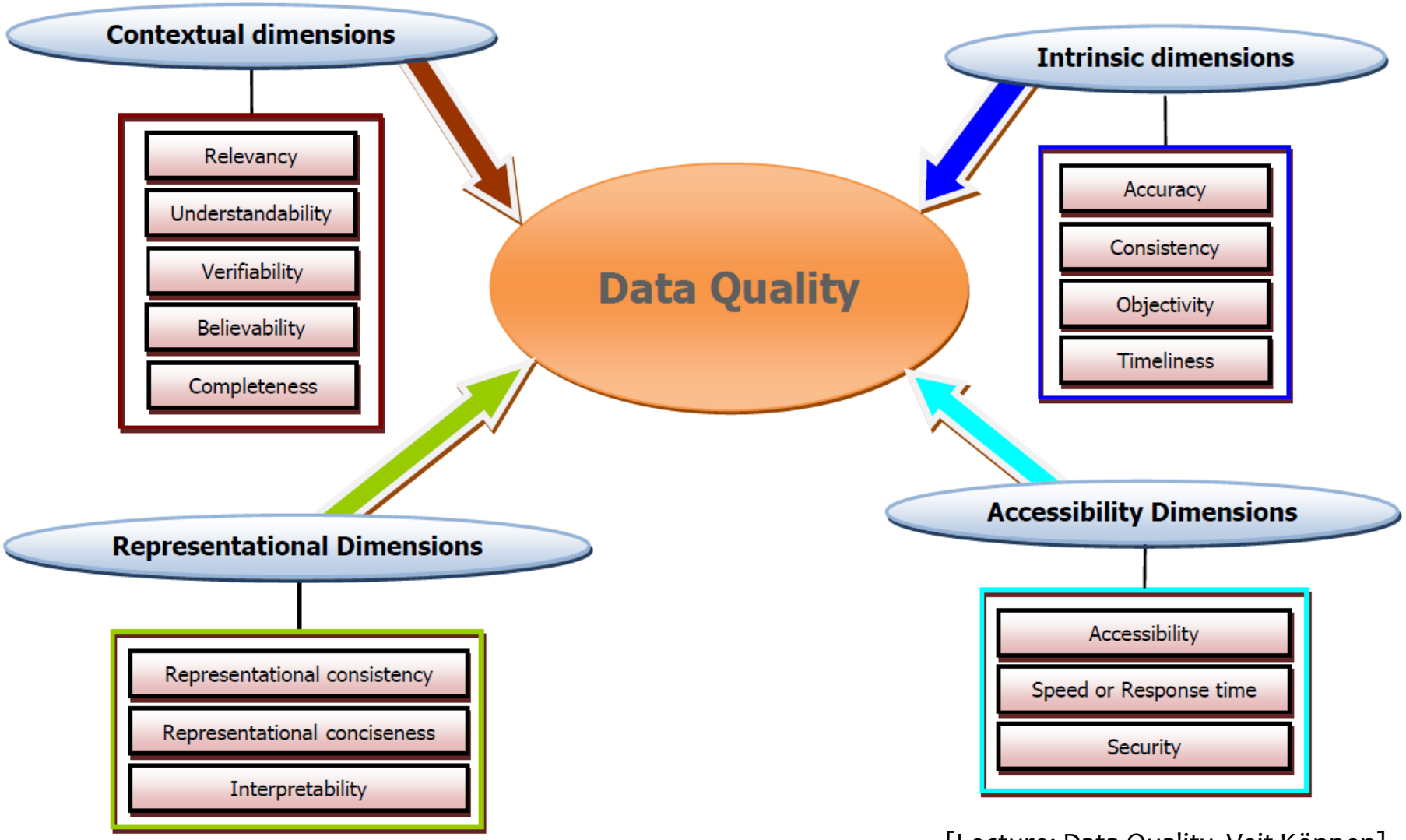
Security

- Data Management must provide measures to protect data from manipulation or disclosure of valuable data
- Access control mechanisms commonly used
  - Authentication: mechanism to identify users of a systems
  - Authorization: mechanism to grant or revoke specific access rights to users
  - Backup (see above): mechanisms to prevent data loss
- Typical threats to data
  - Data loss or integrity violations
  - Intrusion: un-authorized access by misused identity or compromised security measures
  - Data Leakage: disclosure by authorized users to third partys

Data Quality

- “Fitness for use” of data in a given application
- Depends on requirements of this one application
- Quality may refer to many properties classified as so-called “Dimensions” of Data Quality
- Data Quality can be
  - Measured
  - Improved
  - Managed

Data Quality: Dimensions



[Lecture: Data Quality, Veit Köppen]

Concurrency Control

- Data management must support concurrent access by multiple users
- Synchronization required to solve possible problems with consistency
  - **Pessimistic concurrency control** avoids problems by blocking operations leading to violations
  - **Optimistic concurrency control** checks for inconsistencies after operations were performed and tries to resolve possible issues
- Synchronization mechanisms include
  - Locking and exclusive accesses (pessimistic)
  - Timestamp-based synchronization (pessimistic)
  - Versions and variants (optimistic)
- Optimistic concurrency control often more suitable for engineering application due to long-running interactive sessions

Data Management Approaches

- Data Management in Files
  - File Systems
  - File Formats
- Data Management in Databases
  - Overview
  - Relational DBMS
  - Beyond Relational DBMS

File Systems

- Part of operating system to fulfill basic storage requirements
  - Store units of data belonging together in an application context in files
  - Semantics (format, interpretation) of file content is irrelevant to operating system
  - Content type often indicated by file type extension as part of the file name
  - File system provides overlay structure of (hierarchical) directories/folders to organize files and make them more easily accesible
  - Operating systems maps these logical structures to physical of storage media
  - Current operating systems support access control on in multi-user settings
- Distributed file systems used to connect physical resources (hard disks) across nodes in a network and
  - provide shared access typically within local network settings

Text vs. Binary Formats

- Text files
  - Sequence of encoded characters, typically ASCII or UTF
  - Advantages
    - Content may be human-readable
    - Standardized encoding

```
ISO-10303-21;
HEADER;
/* Exchange file generated using ST-DEVELOPER 1.6 */

FILE_NAME(
/* name */ '1797609in',
/* time_stamp */ '1999-02-11T15:33:57-05:00',
/* organization */ ('STEP Tools, Inc.'),
…

DATA;
#10=ORIENTED_EDGE('',*,*,#11,.T.);
#11=EDGE_CURVE('',#13,#13,#12,.T.);
#12=INTERSECTION_CURVE('',#2476,(#2340,#2471),.CURVE_3D.);
…
```

- Binary Files
  - Data encoded as variably used sequence of bytes
  - Advantages:
    - Efficient storage usage
    - Content protected/hidden to some degree

```
00281F50  5D C2 08 00 AC 2B 68 00  00 00 00 00 00 00 00 00  ]Â··¬+h·····
00281F60  00 00 00 00 E4 2B 68 00  00 00 00 00 00 00 00 00  ····ä+h·····
00281F70  00 00 00 00 C8 2B 68 00  34 00 00 00 DC DF 67 00  ····È+h·4···Üßg·
00281F80  F8 59 40 00 00 5A 40 00  1C 5C 40 00 14 5C 40 00  øY@··Z@··\@··\@·
00281F90  34 5C 40 00 38 5C 40 00  3C 5C 40 00 30 5C 40 00  4\@·8\@·<\@·0\@·
00281FA0  D4 58 40 00 F0 58 40 00  CC F3 67 00 3C 2D 68 00  ÔX@·ðX@·Ìóg·<-h·
00281FB0  C0 30 68 00 0C FB 67 00  F8 FA 67 00 A0 FA 67 00  À0h··ûg·øúg· úg·
00281FC0  78 2D 68 00 4C F1 67 00  16 54 53 79 6E 74 61 78  x-h·Lñg··TSyntax
00281FD0  43 6F 6C 6F 72 73 50 6F  77 65 72 47 52 45 50 90  ColorsPowerGREP
00281FE0  E4 2B 68 00 07 16 54 53  79 6E 74 61 78 43 6F 6C  ä+h···TSyntaxCol
00281FF0  6F 72 73 50 6F 77 65 72  47 52 45 50 AC 2B 68 00  orsPowerGREP¬+h·
00282000  70 E0 67 00 00 00 0F 50  6F 77 65 72 47 52 45 50  pàg····PowerGREP
00282010  43 6F 6C 6F 72 73 00 00  00 00 02 00 74 2C 68 00  Colors·····t,h·
00282020  00 00 00 00 00 00 00 00  00 00 00 00 B0 2C 68 00  ·····°,h·
00282030  00 00 00 00 00 00 00 00  00 00 00 00 90 2C 68 00  ····· ,h·
00282040  34 00 00 00 54 2B 68 00  F8 59 40 00 00 5A 40 00  4···T+h·øY@··Z@·
```

Standard vs. Proprietary Formats

- **Standardized file formats:**
  - Typically created or adopted by consortium or initiative including several companies and/or organizations
  - Examples in Engineering: STEP, IGES, JT, etc.
  - Advantages:
    - Allow easy data exchange between applications
    - Useful for rachiving
- **Proprietary formats:**
  - Formats developed for specific application by one developer
  - Examples in Engineering: DWG and DXF (both AutoCAD), BRD (Eagle)
  - “Industry standard”, if format used by several applications (e.g. for exchange) and specification is publically available
  - Advantages:
    - More efficient
    - Tailor-made for application (cover even specific functionality)

XML

- **eXtensible Markup Language**
  - Hierarchical structure of nested elements (tags)
  - Elements may have attributes
  - Actual data on the leave level
  - Mix of content (data) and description (schema, metadata)
- Developed based on SGML (document processing) to exchange any kind of data on the Web
- Inspired by HTML (also based on SGML), which is only useful to exchange
- Can be considered a neutral text format for files
- Application-specific schemas of valid documents can be defined by Document Type Definitions (DTD) or XML Shema (XSD)
- Standard software/libraries for XML processing publically available

XML Example: EAGLE .sch File

```
<schematic>
  <parts>
    <part name="SUPPLY1" deviceset="GND" device=""/>
    <part name="C1" deviceset="C-EU" device="050-024X044" value="22pF"/>
  </parts>
  <sheets>
    <sheet>
      <instances> <!-- Positions the parts on the board. E. g.: -->
        <instance part="SUPPLY1" gate="GND" x="132.08" y="187.96"/>
        <instance part="C1" x="-50.8" y="200.66" rot="R270"/>
      </instances>
      <nets>
        <net name="N$1" class="0">
         <segment>
           <wire x1="9.44" y1="19.04" x2="8.9" y2="19.04" width="0.15"/>
           <wire x1="8.9" y1="19.04" x2="8.9" y2="20.66" width="0.15"/>
           <wire x1="8.9" y1="20.66" x2="2.4" y2="20.66" width="0.15"/>
           <pinref part="C1" pin="5"/>
           <pinref part="SUPPLY1" pin="1"/>
        </segment>
       </net>
     </nets>
   </sheet>
  </sheets>
</schematic>
```

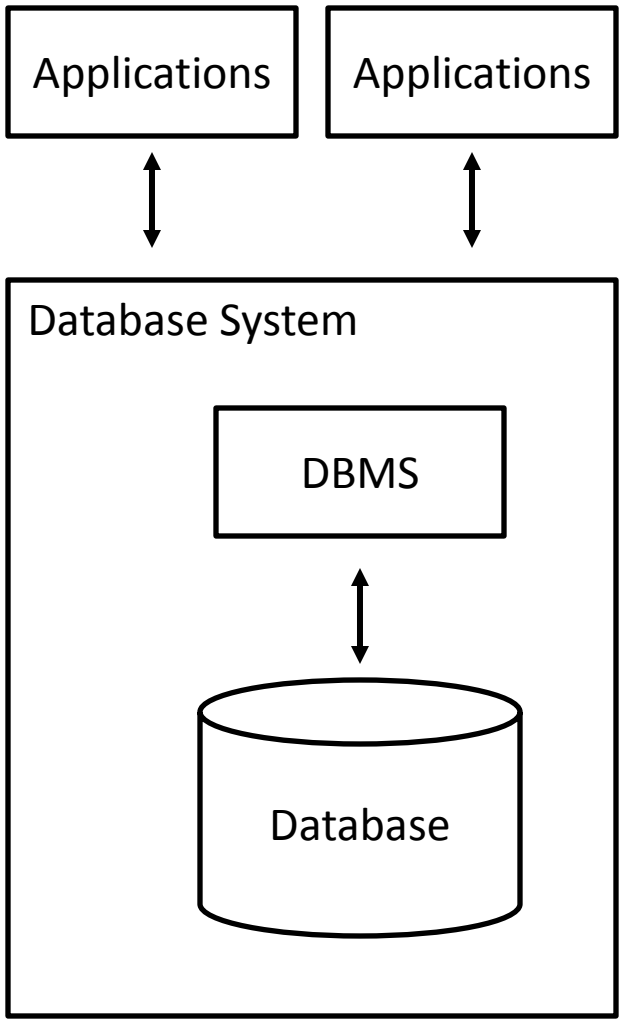
[Source: Philipp Ludwig]

XML in Engineering

- Many formats based on XML
- Especially intended for data exchange
- Some examples:
  - **Collada** for interactive 3D applications
  - **3DXML** for the exchange of geometrical data
  - **EAGLE** board (BRD) and schema (SCH) files for electronic circuits (see above)
  - **CAEX** general purpose language for the exchange of engineering data by European consortium
  - **AutomationML** for plant engineering
  - …

Database Systems

- Database Systems special solution for high data management requirements regarding
  - Efficiency
  - Consistency
  - Concurrent users
- **Database (DB):** collection of real world facts within a given universe of discourse persistently stored
- **Database Management System (DBMS):** specific software independent of universe of discourse
- **Database System (DBS):** entire system of database managed by DBMS



Database Management Systems

- Provide data to applications based on Client Server model within a network
- Current systems mostly relational or object-relational (see below)
  - Oracle (object-relational, commercial)
  - IBM DB2 (object-relational, commercial)
  - Microsoft SQL Server (object-relational, commercial)
  - PostgreSQL (object-relational, open source)
  - MySQL (relational, open source)

Relational Database Systems

- Developed since early 1970s based on mathematical theory of relations and operations performed on them (relational algebra)
  - Data is stored as records (tuples) in tables (relations) with values for each column (attribute)
  - Records can be identified by keys
  - Keys can be used to establish connections across dat in different tables (foreign keys)
  - Constraints can be specified to grant consistency
- **SQL** (Structured Query Language) as a strong standard to access relational databases

Relational Database Systems /2

PartID	**Name**	**Weight**	**SupplierID**
GT-876-140425	Plunger	143.5	1
FT-852-130707	Shaft	77.0	3
FT-855-140809	Bolt	15.7	1
TT-707-778	Case	22.8	2

SupplierID	**Name**	**Location**
1	Reed & Sons	New York
2	CaseStudio	Boston
3	ToolTime	Austin

Beyond RDBMS

- **Object-Oriented DBMS (OODBMS or ODBMS):** usage of object-oriented concepts from OO-programming to add semanticly rich modeling concepts
- **Object-Relational DBMS (ORDBMS):** integrate OO concepts with relational model
- **XML- and Document-Oriented DBMS:** using semi-structured, text-based formats for mass data storage
- **NoSQL DBMS:** flexible data models, mostly used within Cloud context