Data Management in Intelligent Systems

Data Representation and Serialization:

Formats, Markup Languages, etc.

Speaker: Dr. **Alexey Neznanov**, PhD
Senior Data Scientist at Schlumberger Russia
Associate professor at HSE

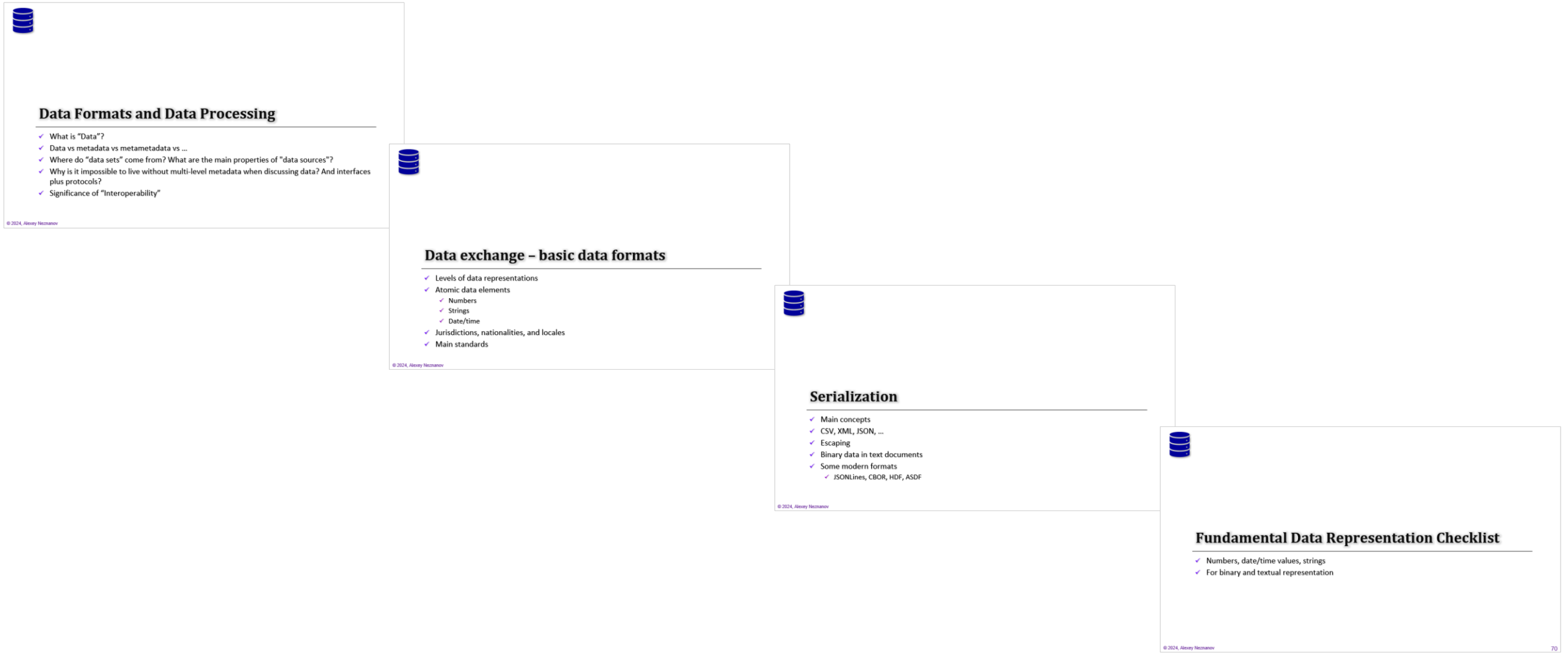
2024-09-01 (v.1.02)




About speaker

- Alexey Neznanov
  - PhD
  - IEEE member
  - Senior Data Scientist at Schlumberger Russia
  - Associate professor at HSE (http://www.hse.ru/staff/aneznanov)
  - «Small Guide to Big Data» consultant at PostScience portal (http://postnauka.ru/author/a_neznanov)
  - Author of textbook, instructional guidelines, 11 courses and more than 65 research papers

Contents


Data Formats and Data Processing
What is "Data"?
Data vs metadata vs metametadata vs ...
Where do "data sets" come from? What are the main properties of "data sources"?
Why is it impossible to live without multi-level metadata when discussing data? And interfaces plus protocols?
Significance of "Interoperability"
© 2024, Alexey Neznanov
Data exchange – basic data formats
Levels of data representations
Atomic data elements
Numbers
Strings
Date/time
Jurisdictions, nationalities, and locales
Main standards
© 2024, Alexey Neznanov
Serialization
Main concepts
CSV, XML, JSON, ...
Escaping
Binary data in text documents
Some modern formats
JSONLines, CBOR, HDF, ASDF
© 2024, Alexey Neznanov
Fundamental Data Representation Checklist
Numbers, date/time values, strings
For binary and textual representation
© 2024, Alexey Neznanov
70

Data Formats and Data Processing

- ✓ What is “Data”?
- ✓ Data vs metadata vs metametadata vs …
- ✓ Where do “data sets” come from? What are the main properties of "data sources"?
- ✓ Why is it impossible to live without multi-level metadata when discussing data? And interfaces plus protocols?
- ✓ Significance of “Interoperability”

Information and Data

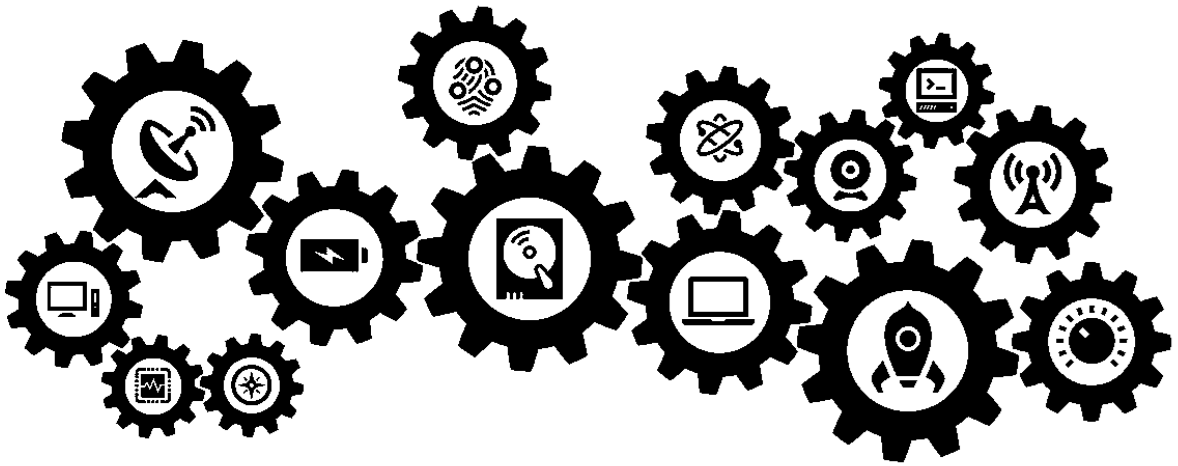
- **Information**
  - It is a basic concept, not defined constructively without recursion due to the problem of *hermeneutic circle*
  - “Resolution of uncertainty” (some common sense)
  - “Knowledge concerning objects, such as facts, events, things, processes, or ideas, including concepts, that within a certain context has a particular meaning”
    (ISO/IEC 2382-1:1993 Information technology – Vocabulary – Part 1: Fundamental terms)
  - My favorite: “**Information is information, not matter or energy!**”
    (Norbert Wiener, Cybernetics: Or Control and Communication in the Animal and the Machine, 1948)

- **Data** – a reinterpretable representation of information in a formalized manner suitable for communication, interpretation, or processing (ISO/IEC 2382)
  - **Data processing** as a main activity!

Main tasks of data processing

- Information system (IS) as a data processing facility
  - Relevant knowledge about data → adequate data processing
  - Workflows and pipelines (another lecture)

- Six main tasks:
  1. **Input/Output**
  2. **Format conversion** (sic!)
  3. **Telecommunication**
  4. **Storage**
  5. **Search (data retrieval)**
  6. **Manipulation**
     - Analysis!

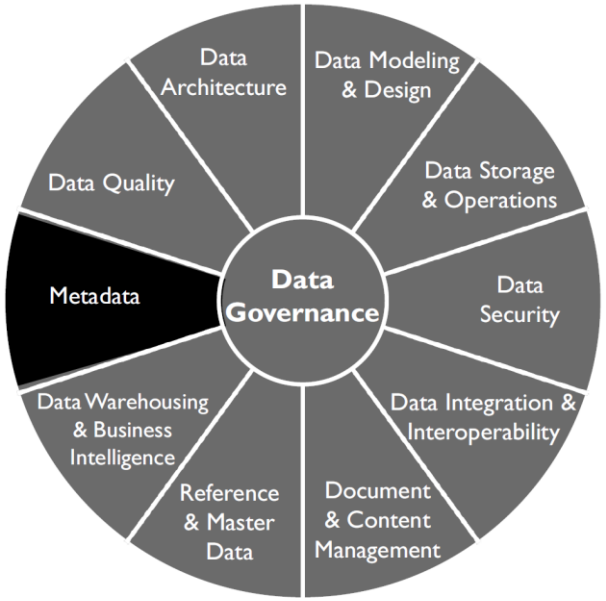


Data and metadata

- *The knowledge* appears at the **metadata** level
  - Data about data ☺
  - What is Metadata (with examples) (http://dataedo.com/kb/data-glossary/what-is-metadata)
  - Dataedo – Data vs Metadata (https://landing.dataedo.com/data-vs-metadata)
- Metadata management
  - Full chapter in DMBoK (Chapter 12: Metadata Management)!
- Metadata formalization and standardization
- Metadata about metadata (infinite hierarchy)
  - Where can we stop? Metadata, **metametadata**, **metametametadata**, ...
  - The problem of harmonization of *ontologies* and *ontics* (interpretation)
  - The problem of interoperability
- Metadata and meta-modelling: modelling, notations, architecture
  - Model, metamodel, metametamodel, … (see UML meta-model, BPMN metamodel)

From DMBoK Chapter 12: Metadata Management

- Common definition of Metadata, “data about data,” is misleadingly simple!
- Metadata includes information about technical and business processes, data rules and constraints, and logical and physical data structures
- It describes:
  - The data itself (e.g., databases, data elements, data models),
  - The concepts the data represents (e.g., business processes, application systems, software code, technology infrastructure),
  - The connections (relationships) between the data and concepts



DAMA-DMBOK2 Data Management Framework

Basic terminology (1)

- ISO 1087:2019 Terminology work and terminology science — Vocabulary

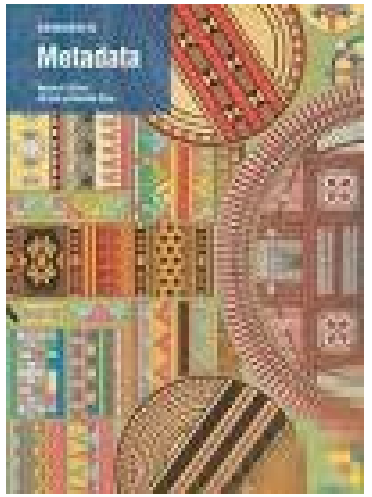
- Committee ISO/IEC JTC 1/SC 32 Data management and interchange (http://www.iso.org/committee/45342.html)
- Information technology — Metadata registries (MDR)
  - ISO/IEC 11179-1:2015 — Part 1: **Framework**
  - ISO/IEC 11179-3:2013 — Part 3: **Registry metamodel and basic attributes**
  - ISO/IEC 11179-4:2004 — Part 4: **Formulation of data definitions**
  - ISO/IEC 11179-5:2015 — Part 5: **Naming principles**
  - ISO/IEC 11179-6:2015 — Part 6: **Registration**
  - **…**

Basic terminology (2)

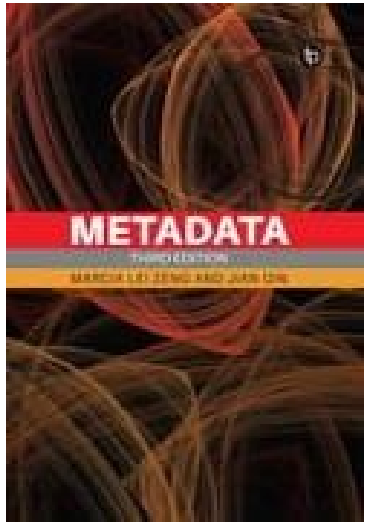
- Information technology — Metamodel framework for interoperability (MFI)
  - ISO/IEC 19763-1:2015 — Part 1: **Framework**
  - ISO/IEC 19763-3:2010 — Part 3: **Metamodel for ontology registration**
  - ISO/IEC 19763-5:2015 — Part 5: **Metamodel for process model registration**
  - ISO/IEC 19763-6:2015 — Part 6: **Registry Summary**
  - ISO/IEC 19763-7:2015 — Part 7: **Metamodel for service model registration**
  - ISO/IEC 19763-8:2015 — Part 8: **Metamodel for role and goal model registration**
  - ISO/IEC TR 19763-9:2015 — Part 9: **On demand model selection**
  - ISO/IEC 19763-10:2014 — Part 10: **Core model and basic mapping**
  - ISO/IEC 19763-12:2015 — Part 12: **Metamodel for information model registration**

Sources about metadata (1)

- *Introduction to Metadata. Edited by Baca M. 3rd ed.* 2016
  - https://www.getty.edu/publications/intrometadata/

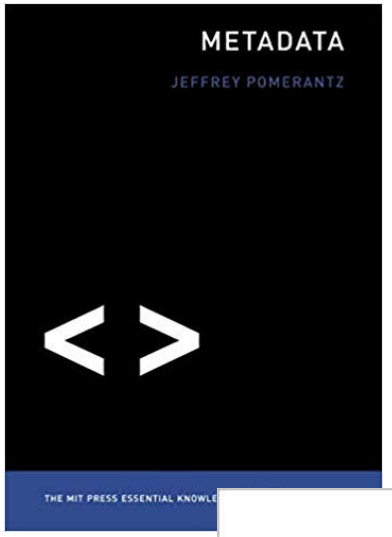


- *Zeng M.L., Qin J. Metadata. 3rd ed. Facet Publishing*, 2022
  - https://www.routledge.com/Metadata/Zeng-Qin/p/book/9781783305889

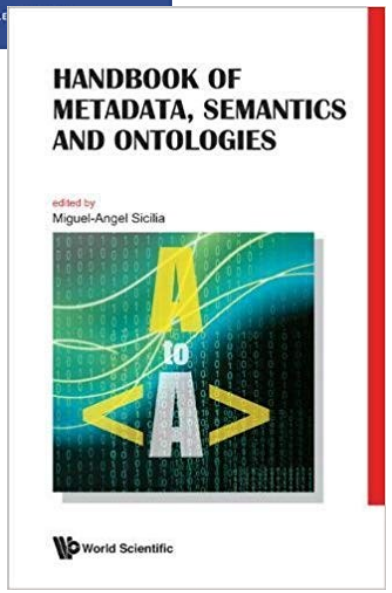


Sources about metadata (2)

- Jeffrey Pomerantz, 2015 – Metadata
  - http://www.amazon.com/Metadata-MIT-Press-Essential-Knowledge/dp/0262528517

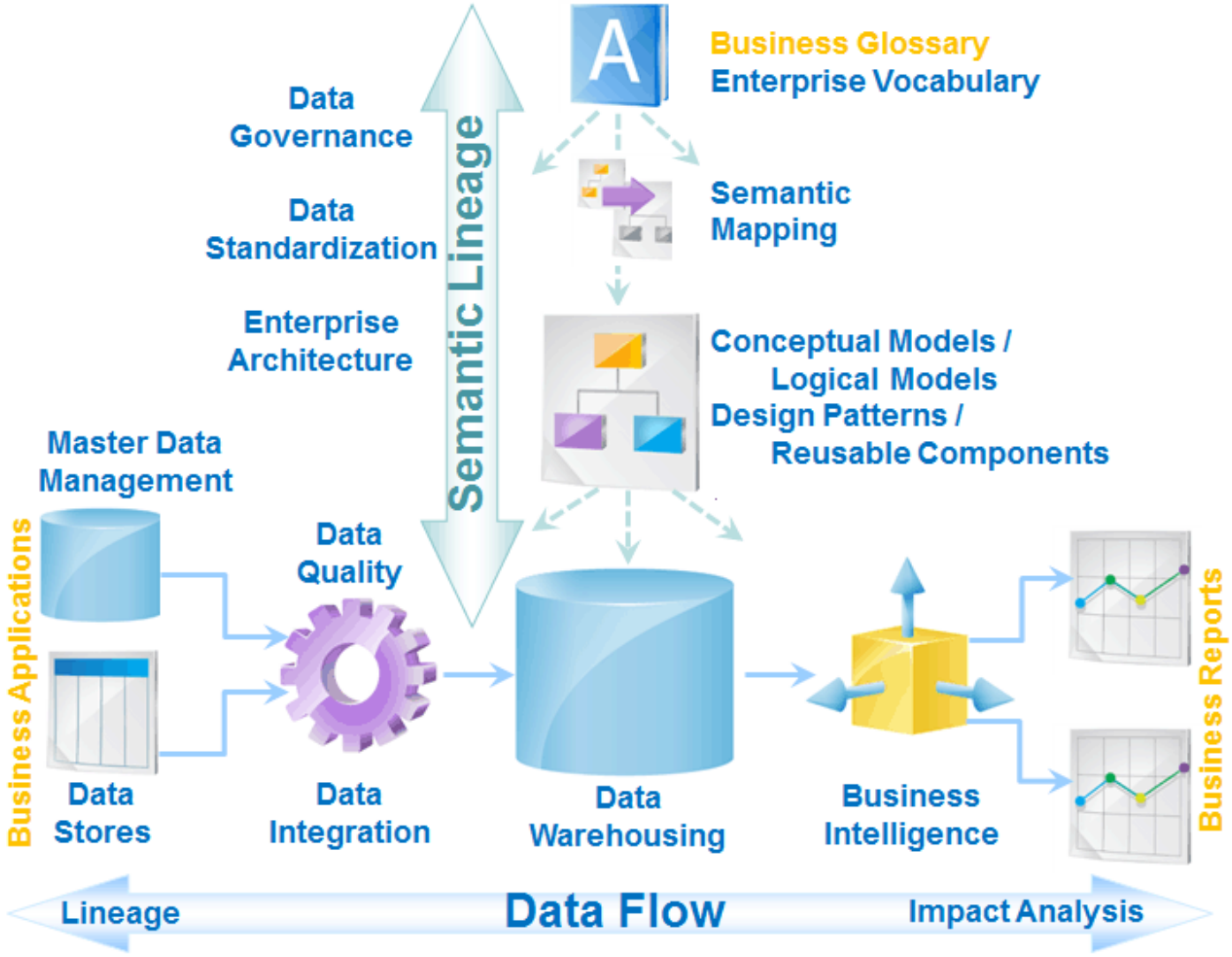


- Miguel-Angel Sicilia (Editor), 2013 – Handbook of Metadata, Semantics And Ontologies
  - http://www.amazon.com/HANDBOOK-METADATA-SEMANTICS-ONTOLOGIES-MIGUEL-ANGEL/dp/9812836292



Example of metadata management

- Meta Integration (http://www.metaintegration.net)
  - Metadata Integration
  - Metadata Harvesting
  - Metadata Management
  - Enterprise Architecture
  - Data Governance

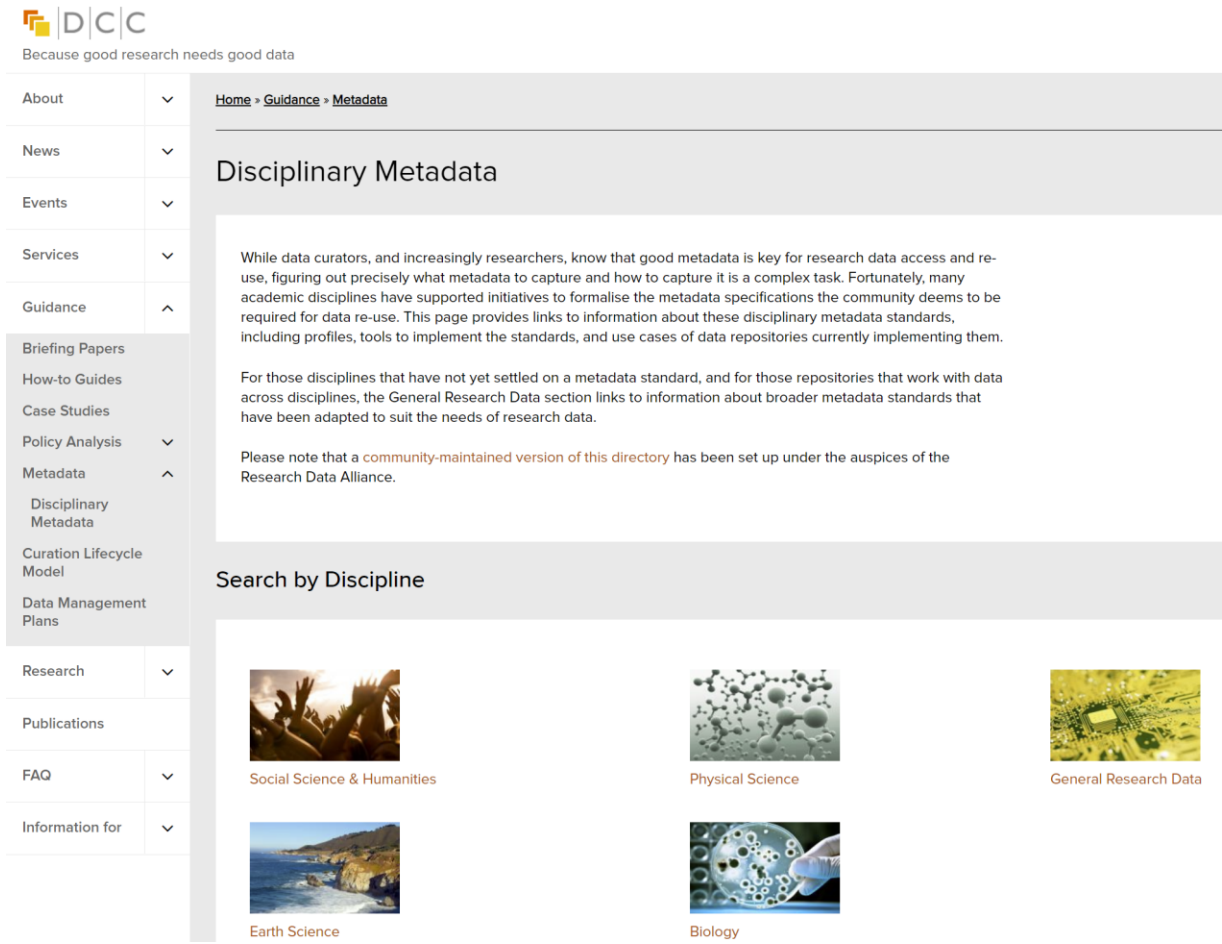


Some other metadata standards

- ISO **15836**-1:2017 Information and documentation — The **Dublin Core** metadata element set — Part 1: **Core elements** (http://www.iso.org/obp/ui/#iso:std:iso:15836:-1:ed-1:v1:en)
- ISO **15836**-2:2019 Information and documentation — The **Dublin Core** metadata element set — Part 2: **DCMI Properties and classes** (http://www.iso.org/obp/ui/#iso:std:iso:15836:-2:ed-1:v1:en)
  - Naden C. International Standard for Descriptive Metadata Just Update. 2020 (http://www.iso.org/news/ref2474.html)
- UNC – Metadata for Data Management: A Tutorial. Standards/Schema (http://guides.lib.unc.edu/metadata/standards)

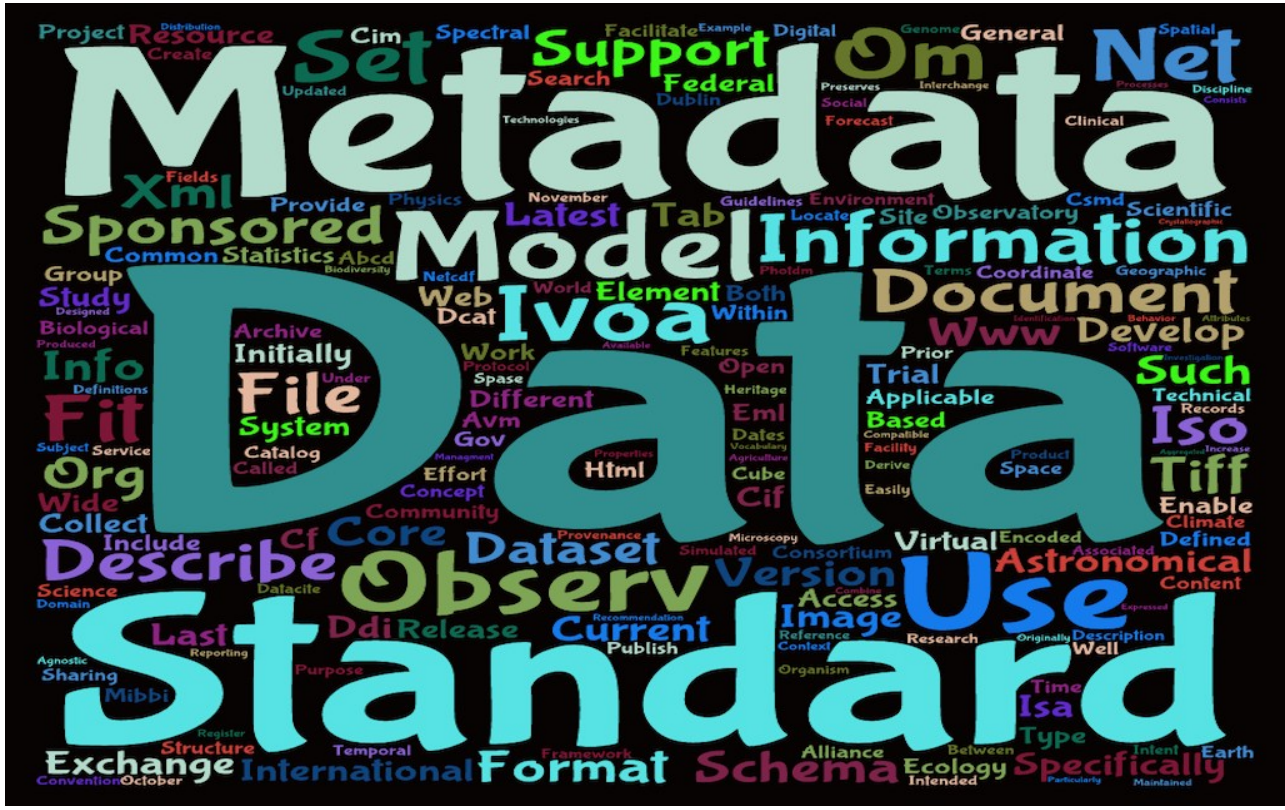
Collections of general metadata (1)

- Digital Curation Centre Disciplinary Metadata (http://www.dcc.ac.uk/guidance/standards/metadata)



Collections of general metadata (2)

- Metadata Standards Directory Working Group (http://rd-alliance.github.io/metadata-directory/)



Current formats – sources

- Where should we look for data/metadata formats that make sense to use in different situations?
  - Library of Congress – Recommended Formats Statement (http://www.loc.gov/preservation/resources/rfs/index.html)
  - UK Data Service – Recommended formats (http://ukdataservice.ac.uk/learning-hub/research-data-management/format-your-data/recommended-formats/)
  - …
- How do you look and check for adequacy?
  - Consistency of description
  - Clear statement of purpose, limitations, constraints
  - The relevance of the recommendations (we will seriously revisit this issue)
  - Completeness of multimedia options
  - Ease of serialization
  - Coherence of recommendations with internationalization requirements

Artifacts and documents

- **Artifact** – artificially created tangible or intangible (information) object with some purpose
- **Document** – artifact, which is a means of fixing the facts, events, phenomena of objective reality and thinking activity of a person on the material carrier in various ways
  - Documents:
    - **Primary** (fixing the facts)
    - **Secondary** (fixing the results of analysis of the facts)
  - Documents often have normative metadata
    - Document entry

Documents standardization

- ISO 5127:2017 Information and documentation — Foundation and vocabulary (http://www.iso.org/obp/ui#iso:std:iso:5127:en)
- **Definition 3.1.1.38 Document – recorded information** (3.1.8.26) or **material object** (3.1.1.60) which can be treated as a unit in a **documentation** (3.2.1.22) process
  - Note 1: See also ISO 25964-1:2011, definition 2.15; ISO 11005:2010, definition 3.1; ISO 15489-1:2016, definition 3.10; IEC 82045-1:2001, definition 3.2.3; ISO 9000:2015, definition 3.7.2.
  - Note 2: This definition refers not only to written and printed materials in paper or microform versions (for example, conventional books, journals, diagrams, maps), but also to non-printed media such as machine-readable and digitized records, Internet and intranet resources, films, sound recordings, people and organizations as knowledge resources, buildings, sites, monuments, three-dimensional objects or realia; and to collections of such items or parts of such items. (Note taken from ISO 25964-1:2011, definition 2.15.) Also, **software** (3.1.12.14), since **recorded** (3.1.8.26), can be considered a *document*.
  - Note 3: Documents often are the **manifestations** (3.2.1.09) of **works** (3.2.1.07).They can differ extensively in form and characteristics.
  - Note 4: In some professional usage, *documents* are sometimes referred to as “medium”, “title” (3.7.4.01, Note 1 to entry) or “item”.

Datasets and data sources

- **Dataset** – any collection of data fully available in a particular IS with a fully known structure and format (basic metadata!)
  - The requirement of "known structure and format" leads us to a definition of a so-called **raw dataset** when we do not know something important about dataset
  - But here we start to make some misinterpretations: sometimes raw dataset is a non-aggregated primary data of some IS
    - Remember Tim Berners-Lee: Raw data, now! (http://www.ted.com/talks/tim_berners_lee_the_next_web).
- **Data source** – the IS programming interface (API) that allows to obtain a dataset
  - Interface may refer to: file system API, data stream API, database API, knowledge base API, …
  - ☑ Openness is defined precisely at the level of data sources!
    - See later: open data

What should we know about the data source?

- Main characteristics of the data source:
  1. Source type (there is a variety of classifications)
  2. Address (binding point) and access protocol
  3. Data formats (basic metadata needed for interpretation of data in datasets)
  4. License (hereinafter disclosed when discussing open data)
  5. Data volume (if the volume grows, the data may become “big data”)
  6. Update frequency and changeset sizes

- And goals of stakeholders!

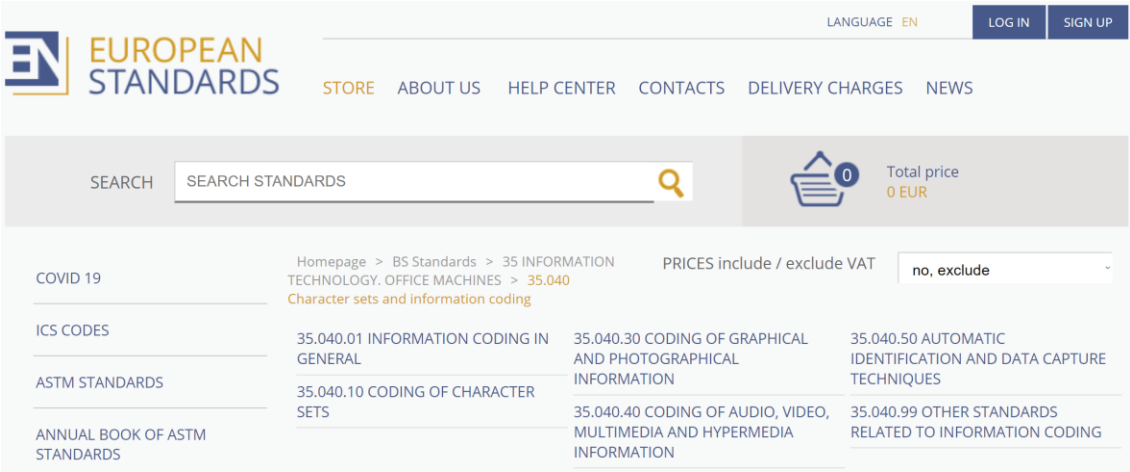
Basic dataset kinds

- **Unstructured text** – texts in natural language
  - Without additional syntactic/semantic markup
  - With minimal markup (wiki, Markdown, ...)
- **Tabular data** – matrices (and relational model)
  - Example: Data.gov: Datasets ordered by Popular (formats:CSV) (http://catalog.data.gov/dataset?res_format=CSV)
    - And in any other formats! XML, JSON, Microsoft Excel, …
- **Hierarchical data** – tree-like structures
  - JSON Example (http://json.org/example.html), XML, HDF5, …
- **Network data** – graph/hypergraph models
  - Stanford Large Network Dataset Collection (http://snap.stanford.edu/data/)

- Big Data as a characteristic of processing complexity (and not just volume!)

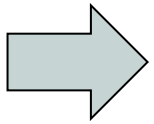
The concept of data format

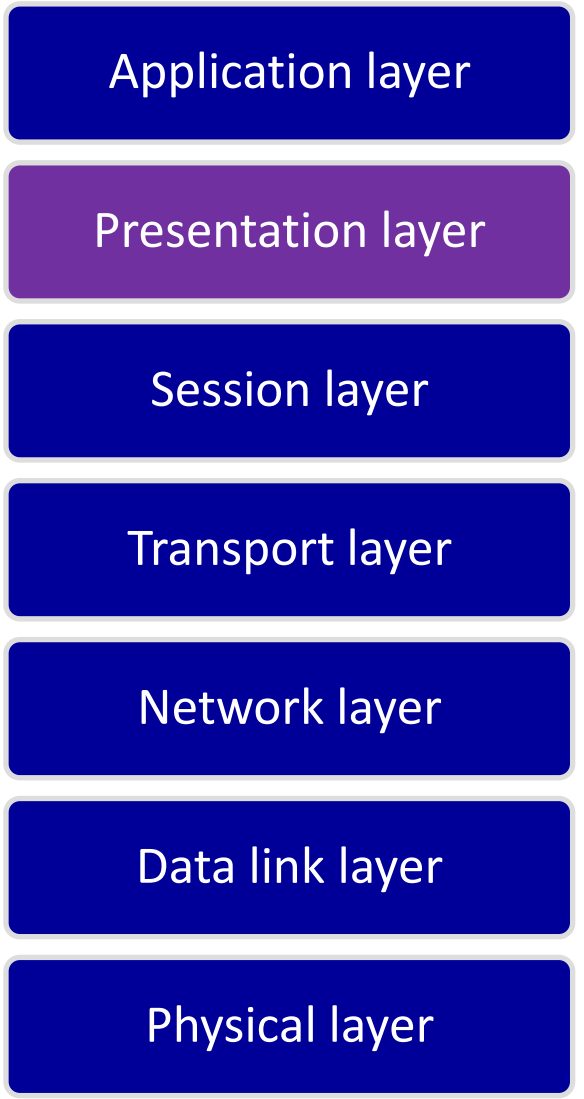
- **Data format** – a set of rules for representing and interpreting data in memory, on external media, in input/output operations, and in communication channels
  - The specific format defines the *way of encoding* data in some sign system and *placement* in media (in other words, format is a description of the physical structure of an information element)
  - Standard Group ISO **35.040** “Information coding” (http://www.iso.org/ics/35.040/x/)
    - Including coding of audio, picture, multimedia and hypermedia information, bar coding, etc.
    - More than 400 standards define the formats of text, audio and visual information, algorithms for coding and decoding, compression (archiving) formats, and methods of format conversion



Where do data formats live?

- They live anywhere data can be found!
  - But we are most interested in the penultimate layer of the OSI – the data representation layer
  - Here application data formats, data serialization formats and telecommunication formats meet
    - *Presentation Layer* (http://www.tech-faq.com/presentation-layer.html)





Interoperability – concept

- **Interoperability** – a polysemous concept, but we are interested in the following definition:
  - Characteristic of a set of systems whose interfaces are known (documented) and witch can interchangeably function as an element of a supersystem
    - It is broader than **compatibility**
    - It also provides a good distinction between **levels of abstraction** and complements the **open systems** model (OSI)
    - It allows us to talk more precisely about **platforms** as the place where it manifests itself in modern IT
    - Interfaces, protocols, and data formats are convenient to discuss from the perspective of interoperability
- ☑ The higher the level of interoperability –
  the faster the development of platform services

Interoperability – good example of description

- As of 2017, there is a very good description of interoperability practices by *NIFO – National Interoperability Framework Observatory* (http://joinup.ec.europa.eu/collection/nifo-national-interoperability-framework-observatory)
  - This is part of the JoinUp initiative with the following slogan "***Share and reuse Interoperability solutions for public administrations, businesses and citizens***"

- *European Interoperability Framework* (***EIF***) (http://ec.europa.eu/isa2/eif)
  - Обзор: *Sharma G. eIDAS and the new European Interoperability Framework – One step closer to the Single Market*. 2018 (http://www.cryptomathic.com/news-events/blog/eidas-and-the-new-european-interoperability-framework-one-step-closer-to-the-single-market)

Interoperability – principles and levels

1: Subsidiarity and proportionality
1 recommendation

2: Openness
3 recommendations

3: Transparency
1 recommendation

4: Reusability
2 recommendations

5: Technological neutrality and data portability
2 recommendations

6: User-centricity
4 recommendations

7: Inclusion and accessibility
1 recommendation

8: Security and privacy
1 recommendation

9: Multilingualism
1 recommendation

10: Administrative simplification
1 recommendation

11: Preservation of information
1 recommendation

12: Assessment of Effectiveness and Efficiency
1 recommendation

Interoperability governance
5 recommendations

Integrated public service governance
2 recommendations

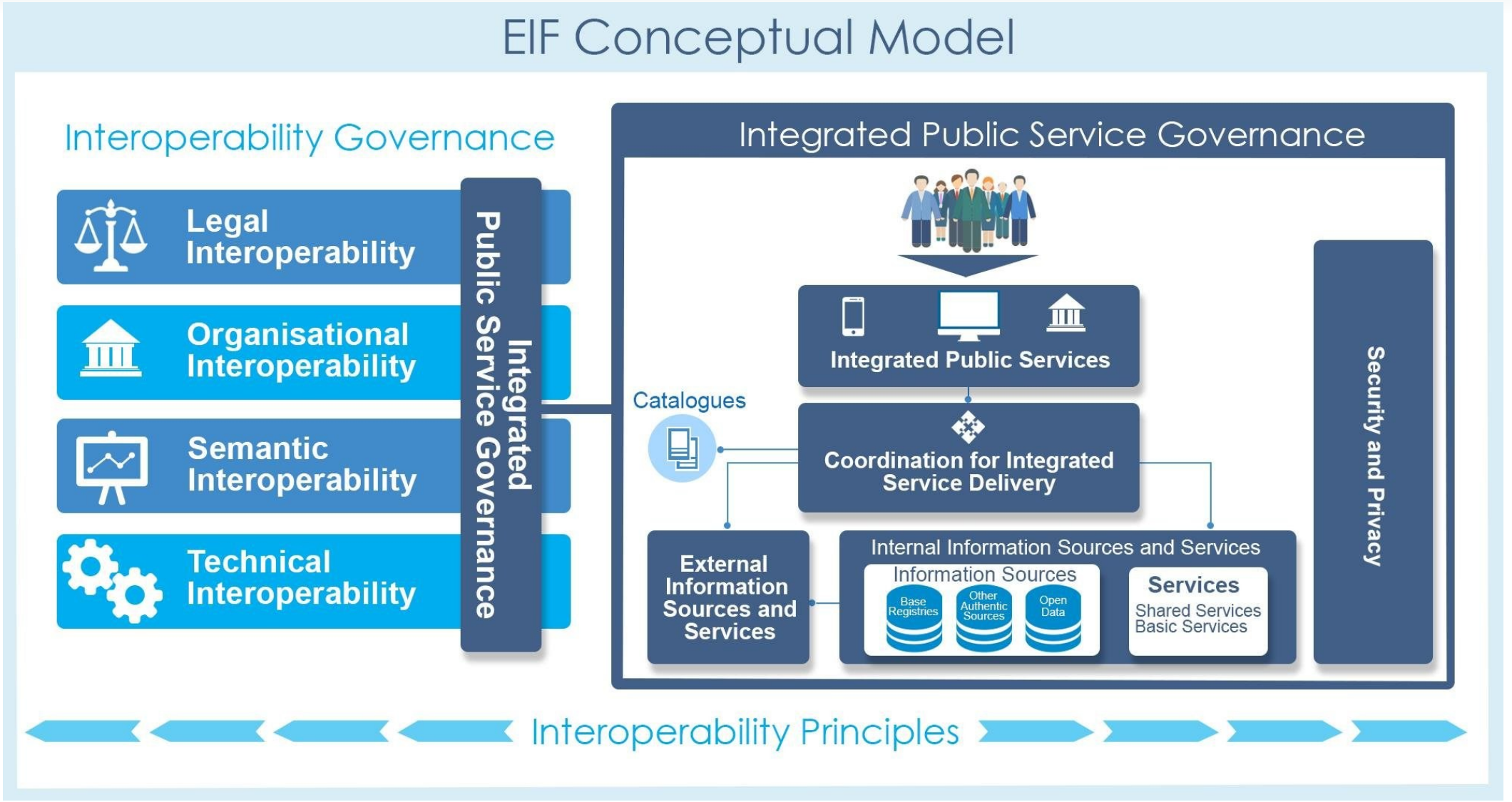
Legal interoperability
1 recommendation

Organisational interoperability
2 recommendations

Semantic interoperability
3 recommendations

Technical interoperability
1 recommendation

Interoperability – EIF conceptual model



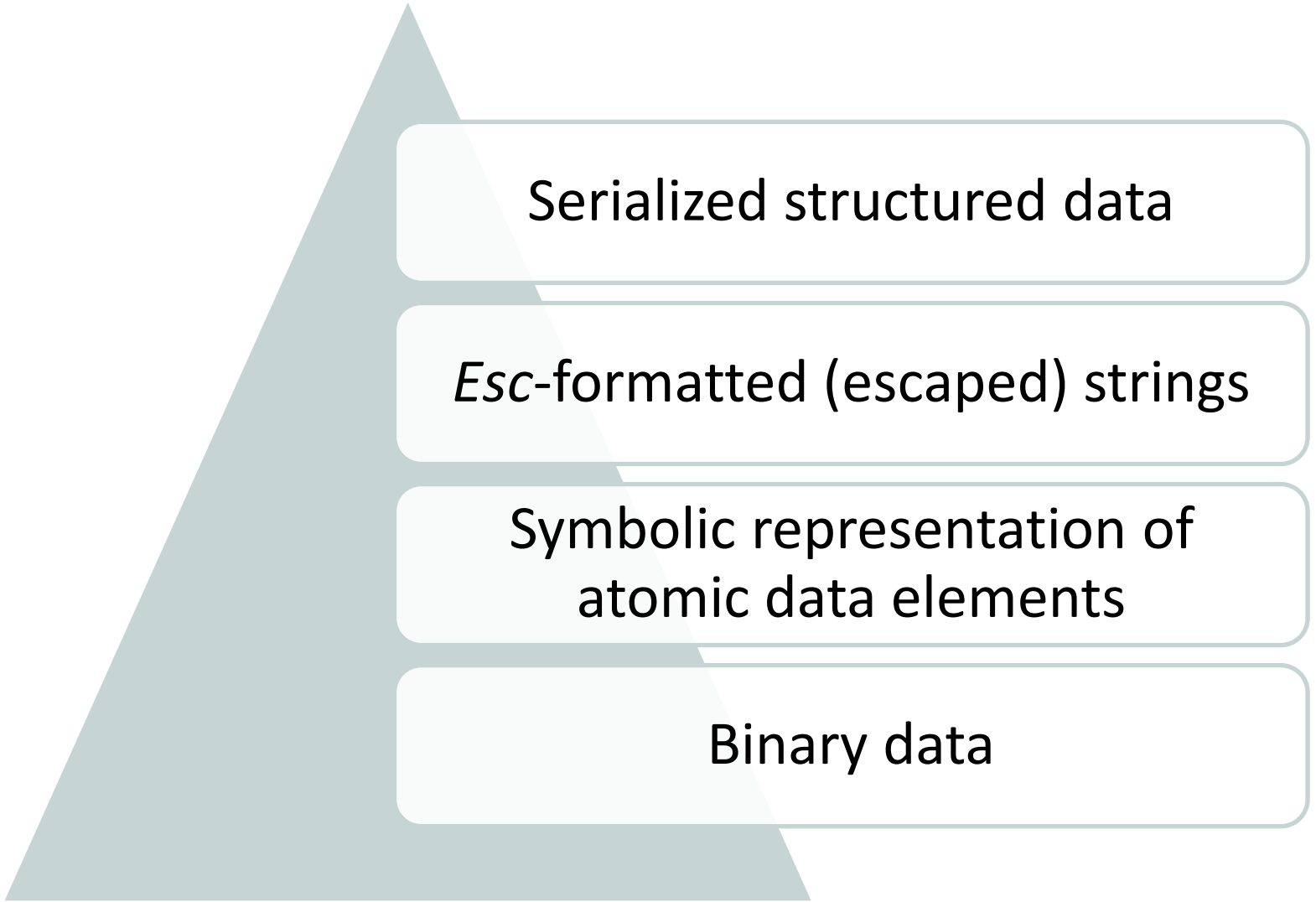
Data exchange – basic data formats

- ✓ Levels of data representations
- ✓ Atomic data elements
  - ✓ Numbers
  - ✓ Strings
  - ✓ Date/time
- ✓ Jurisdictions, nationalities, and locales
- ✓ Main standards

Data exchange – levels of “readability”

Converters

Data types



Main standards (numbers)

- Integer number
  - Let's not repeat...
- Real number
  - They have two basic forms of representation – fixed point and floating point
- Fixed Point Number
  - Separate formats for working with sums of money (currency)
- Floating Point Number
  - **IEEE 754-2019** "IEEE Standard for Floating-Point Arithmetic" (http://grouper.ieee.org/groups/754)
    - ISO/IEC **60559**:2020 (http://www.iso.org/standard/80985.html)
  - ☑ Precision problem and rounding problem!
    - Possible solution: **Interval Computations** (http://www.reliable-computing.org)

- A Tutorial on Data Representation Integers, Floating-point Numbers, and Characters (http://www3.ntu.edu.sg/home/ehchua/programming/java/DataRepresentation.html)

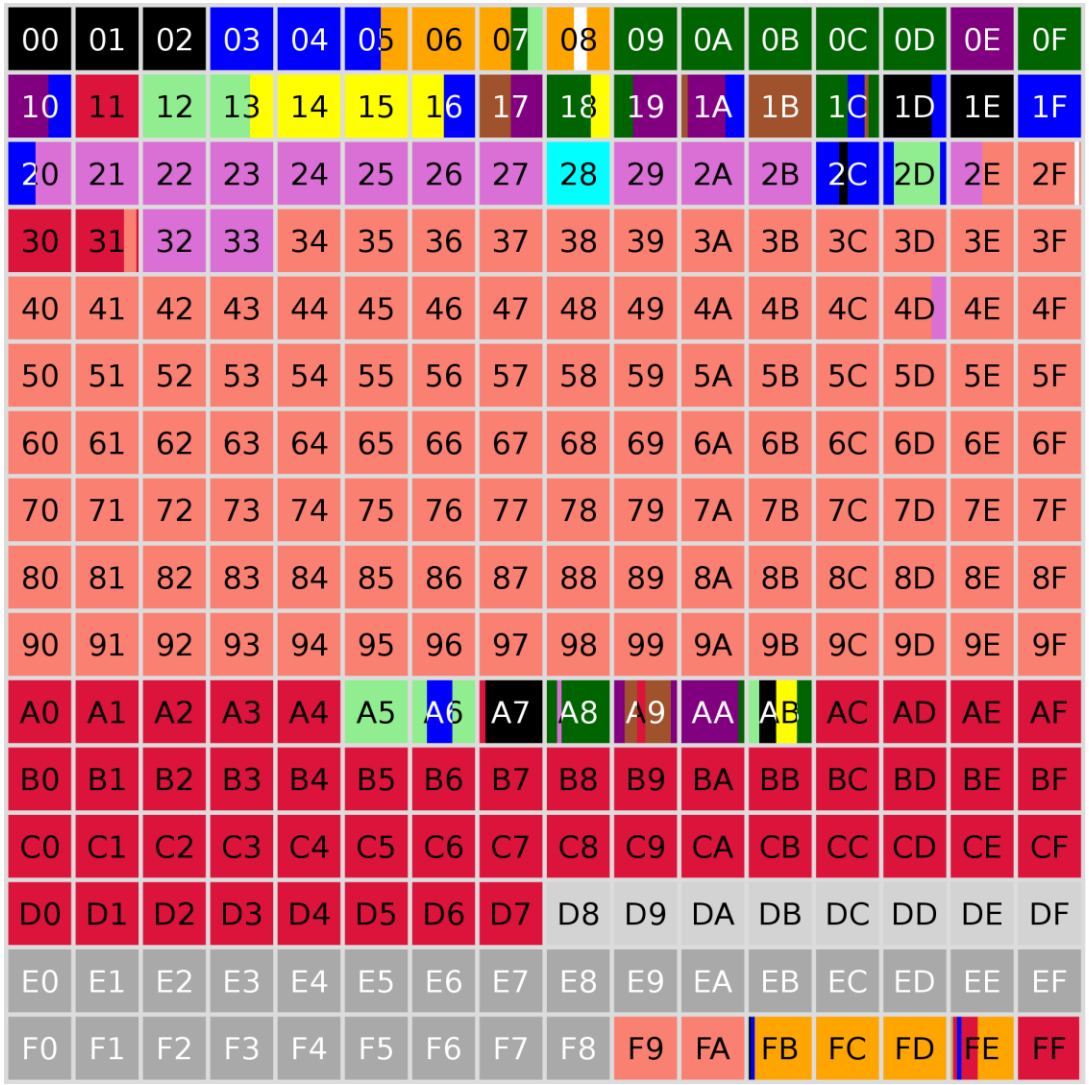
Main standards (strings)

- String (taking language into account!)
  - ASCII – American Standard Code aimed at Information Interchange (http://www.ascii.ca)
    - National codepages (in Russia: MS-DOS Codepage 866, Microsoft Windows Codepage 1251)
  - ISO/IEC 646:1991 “Information technology – ISO 7-bit coded character set for information interchange“ (http://www.iso.org/iso/catalogue_detail.htm?csnumber=4777)
  - ISO/IEC 10646:2014 “Information technology – Universal Coded Character Set” (http://www.iso.org/iso/home/store/catalogue_ics/catalogue_detail_ics.htm?csnumber=63182)
    - Specifies three encoding forms of the UCS: UTF-8, UCS2 (UTF-16), and UCS4 (UTF-32)
  - **UNICODE** (http://www.unicode.org)
    - ICU library – International Components for Unicode (http://site.icu-project.org)
  - ISO 4217:2015 “Codes for the representation of currencies and funds” (http://www.iso.org/iso/home/standards/currency_codes.htm)

Unicode – features (UCS и UTF)

- Be sure to distinguish:
  - **Universal character set** (UCS), that defines code points that formalize a clear correspondence between a **character** and a **code point**
    - Basic multilingual plane (BMP)! It's plane **0**
    - Code points from BMP are denoted by the prefix "U-" with the addition of hexadecimal value (four digits)
    - Next code point areas are highlighted for Cyrillic characters: from U+0400 to U+052F, from U+2DE0 to U+2DFF, and from U+A640 to U+A69F
  - Family of **encodings** – Unicode transformation formats (UTF), that define how a text string is represented as a bit sequence
- UTF-8 Everywhere Manifesto (http://utf8everywhere.org)
- Character search service "&what"; (http://www.amp-what.com/unicode/search/check)

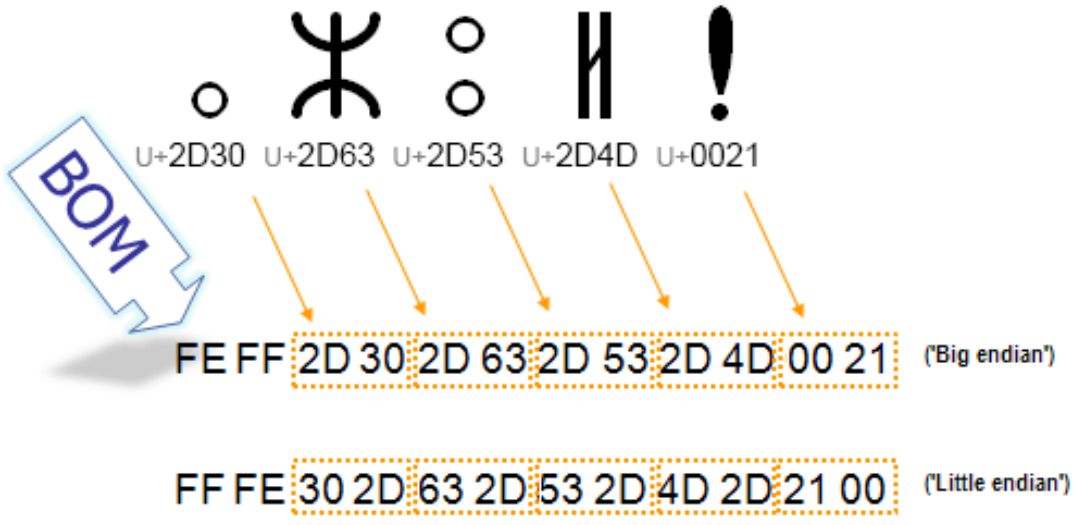
Unicode – BMP Structure



Unicode – features (BOM)

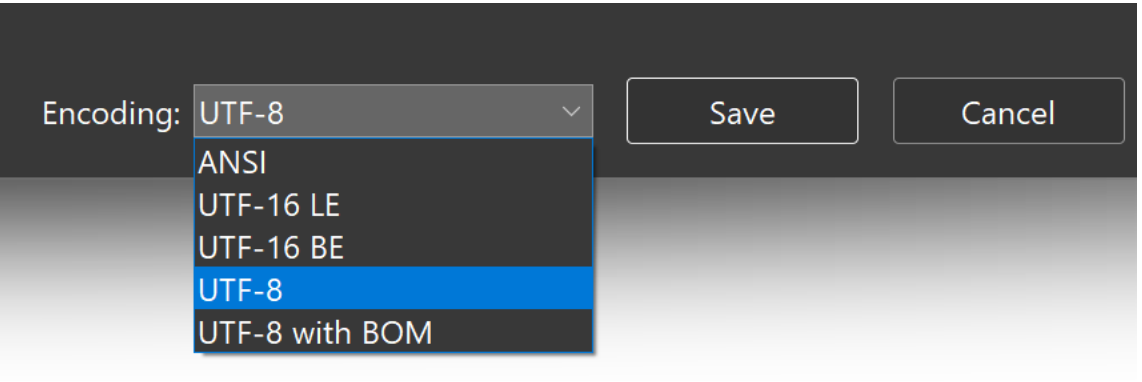
- **Byte order mark** (BOM) – U+FEFF symbol code at the beginning of the data stream to identify the byte order in the character representation and UTF encoding format
  - High level protocols may explicitly require or prohibit BOM in data streams
  - General questions, relating to UTF or Encoding Form (http://unicode.org/faq/utf_bom.html)

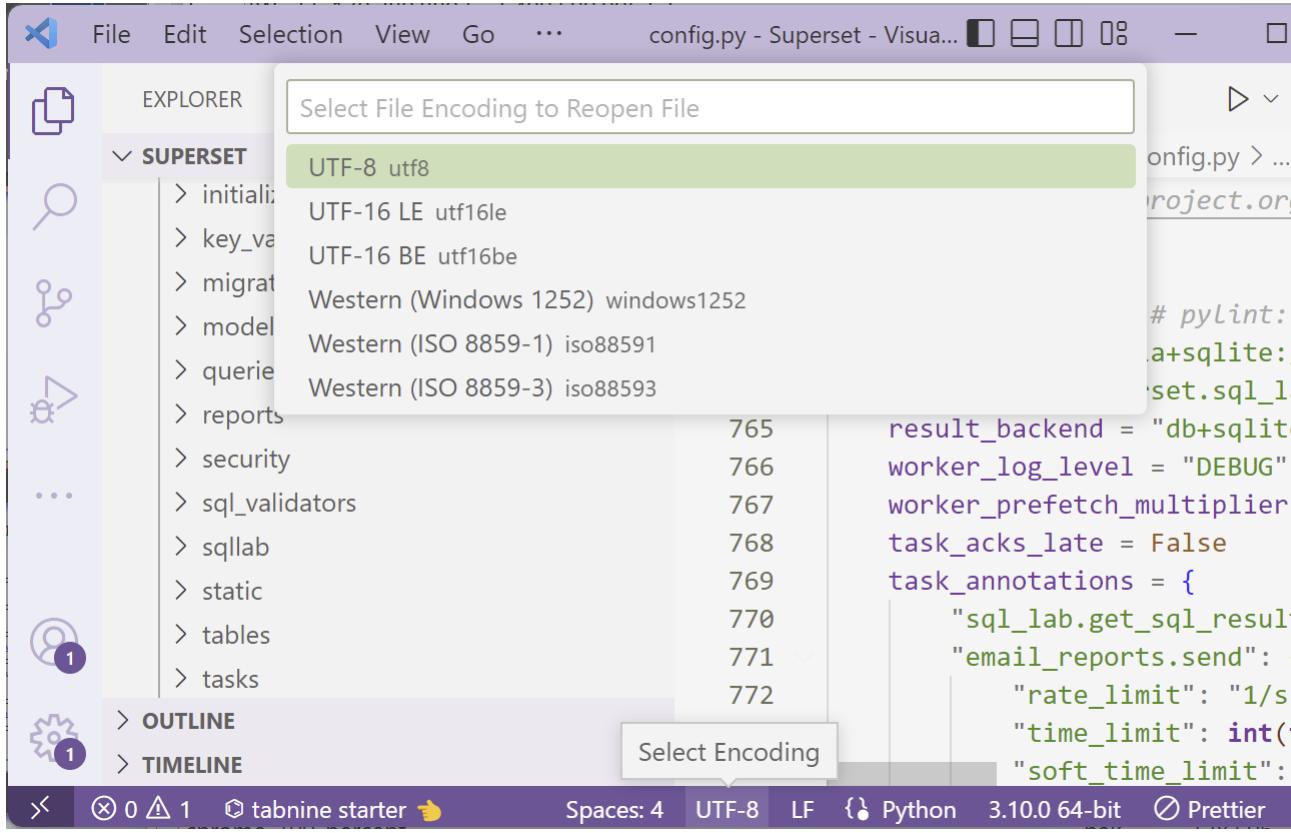
BOM	Stream format
00 00 FE FF	UTF-32, big-endian
FF FE 00 00	UTF-32, little-endian
FE FF	UTF-16, big-endian
FF FE	UTF-16, little-endian
EF BB BF	**UTF-8**



Note on Notepad, Terminal, VSCode, ...

- Understanding file encoding in VS Code and PowerShell (http://learn.microsoft.com/en-us/powershell/scripting/dev-cross-plat/vscode/understanding-file-encoding)





Strings: localization and internationalization (1)

- “**ISO 639** Language Codes” standards:
  - ISO 639-1:2002 Codes for the representation of names of languages — Part 1: **Alpha-2 code** (http://www.iso.org/standard/22109.html)
  - ISO 639-2:1998 Codes for the representation of names of languages — Part 2: **Alpha-3 code** (http://www.iso.org/standard/4767.html)
  - ISO 639-3:2007 Codes for the representation of names of languages — Part 3: Alpha-3 code for comprehensive coverage of languages (http://www.iso.org/standard/39534.html)
  - ISO 639-4:2010 Codes for the representation of names of languages — Part 4: General principles of coding of the representation of names of languages and related entities, and application guidelines (http://www.iso.org/standard/39535.html)
  - ISO 639-5:2008 Codes for the representation of names of languages — Part 5: Alpha-3 code for language families and groups (http://www.iso.org/standard/39536.html)
  - ISO 639-6:2009 Codes for the representation of names of languages — Part 6: **Alpha-4 code** for comprehensive coverage of language variants (http://www.iso.org/standard/43380.html)

Strings: localization and internationalization (2)

- “**ISO 3166** Country Codes“ standards:
  - ISO 3166-1:2013 Codes for the representation of names of countries and their subdivisions — Part 1: **Country codes** (http://www.iso.org/standard/63545.html)
  - ISO 3166-2:2013 Codes for the representation of names of countries and their subdivisions — Part 2: **Country subdivision code** (http://www.iso.org/standard/63546.html)
  - ISO 3166-3:2013 Codes for the representation of names of countries and their subdivisions — Part 3: Code for formerly used names of countries (http://www.iso.org/standard/63547.html)

- Use Online Browsing Platform (OBP) (http://www.iso.org/obp/ui/#search)
  - Search “**Language Codes**” and “**Country codes**”!

Strings: localization and internationalization (3)

- Internationalization techniques: Authoring HTML & CSS (http://www.w3.org/International/techniques/authoring-html)

- Tables:
  - Codes for the Representation of Names of Languages (http://www.loc.gov/standards/iso639-2/php/English_list.php)
  - HTML **Language Code** Reference (http://www.w3schools.com/tags/ref_language_codes.asp)
    - Alpha-2 code from ISO 639-1
  - ISO **Country Codes** (http://www.w3schools.com/tags/ref_country_codes.asp)
    - The result is the language code and the country code through a hyphen: html lang="en-US", html lang="ru-RU"
  - ISO **Language Codes** (639-1 and 693-2) and IETF Language Types (http://datahub.io/core/language-codes)

Main standards (date/time)

- ISO **8601**-1:2019 Date and time — Representations for information interchange — Part 1: **Basic rules** (http://www.iso.org/obp/ui/#iso:std:iso:8601:-1:ed-1:v1:en)
- ISO **8601**-2:2019(en) Date and time — Representations for information interchange — Part 2: **Extensions** (http://www.iso.org/obp/ui/#iso:std:iso:8601:-2:ed-1:v1:en)
  - ISO 8601-1, ISO 8601-2 both published **2019-03-26**
- Introduction to the new ISO 8601-1 and ISO 8601-2 (http://www.isotc154.org/posts/2019-08-27-introduction-to-the-new-8601/)
- The Mathematics of the ISO 8601 Calendar by R.H. van Gent
  - The 15 Possible ISO Calendars (http://webspace.science.uu.nl/~gent0113/calendar/isocalendar.htm)

Date/time: ISO 8601 evolving

- History of the standard:

  - ISO 8601 is now a family of standards
    - ISO 8601-1:2019 is the direct successor to ISO 8601:2004
    - ISO 8601-2:2019 provides extensions on top of ISO 8601-1:2019
  - Now normatively referenced by BIPM (creator of UTC)

ISO 8601 – Date/time: joint representation of calendar date and time

- Basic format
  - `YYYYMMDDThhmmss` – calendar date and time of day
  - `YYYYMMDDThhmmssZ` – calendar date and Coordinated Universal Time (UTC) of day
  - `YYYYMMDDThhmmss±hhmm` – calendar date and difference between local time and UTC of day (in hours and minutes)
  - `YYYYMMDDThhmmss±hh` – calendar date and difference between local time and UTC of day (in hours)

- Extended format
  - `YYYY-MM-DDThh:mm:ss`
  - `YYYY-MM-DDThh:mm:ssZ`
  - `YYYY-MM-DDThh:mm:ss±hh:mm`
  - `YYYY-MM-DDThh:mm:ss±hh`

ISO 8601 – Duration

- Format with designators:
  - **P**nnYnnMnnDTnnHnnMnnS
  - **P**nnW

  - Combination of components with designators. The designator [P] shall precede, without space, the remainder of the expression of duration
  - The number of years shall be followed by the designator [Y], the number of months by [M], the number of weeks by [W], and the number of days by [D]
  - The part including time components shall be preceded by the designator [T]; the number of hours shall be followed by [H], the number of minutes by [M] and the number of seconds by [S]
  - The designator [T] shall be absent if all of the time components are absent

ISO 8601 – Time interval

- We need to now start and end times of an interval
  - The **“/” solidus** character separates start and end times in the representation

- Time interval defined by start and end:
  - Basic format: YYYYMMDDThhmmss/YYYYMMDDThhmmss
  - Extended format: YYYY-MM-DDThh:mm:ss/YYYY-MM-DDThh:mm:ss

- Time interval defined by start and duration:
  - Basic format: YYYYMMDDThhmmss/**P**nnYnnMnnDTnnHnnMnnS or YYYYMMDDThhmmss/**P**YYYYMMDDThhmmss
  - Extended format: YYYY-MM-DDThh:mm:ss/**P**nnYnnMnnDTnnHnnMnnS or YYYY-MM-DDThh:mm:ss/**P**YYYY-MM-DDThh:mm:ss

ISO 8601 – Recurring time interval

- All representations start with the designator [R] followed, without spaces, by the number of recurrences, if present, followed, without spaces, by a solidus [/], followed, without spaces, by the expression of a time interval

- Basic format:
  - **R**n/YYYYMMDDThhmmss/YYYYMMDDThhmmss – a start and an end
  - **R**n/YYYYMMDDThhmmss/**P**nYnMnDTnHnMn – a start and a duration

- Extended format:
  - **R**n/YYYY-MM-DDThh:mm:ss/YYYY-MM-DDThh:mm:ss
  - **R**n/PYYYY-MM-DDThh:mm:ss/**P**nYnMnDTnHnMnS

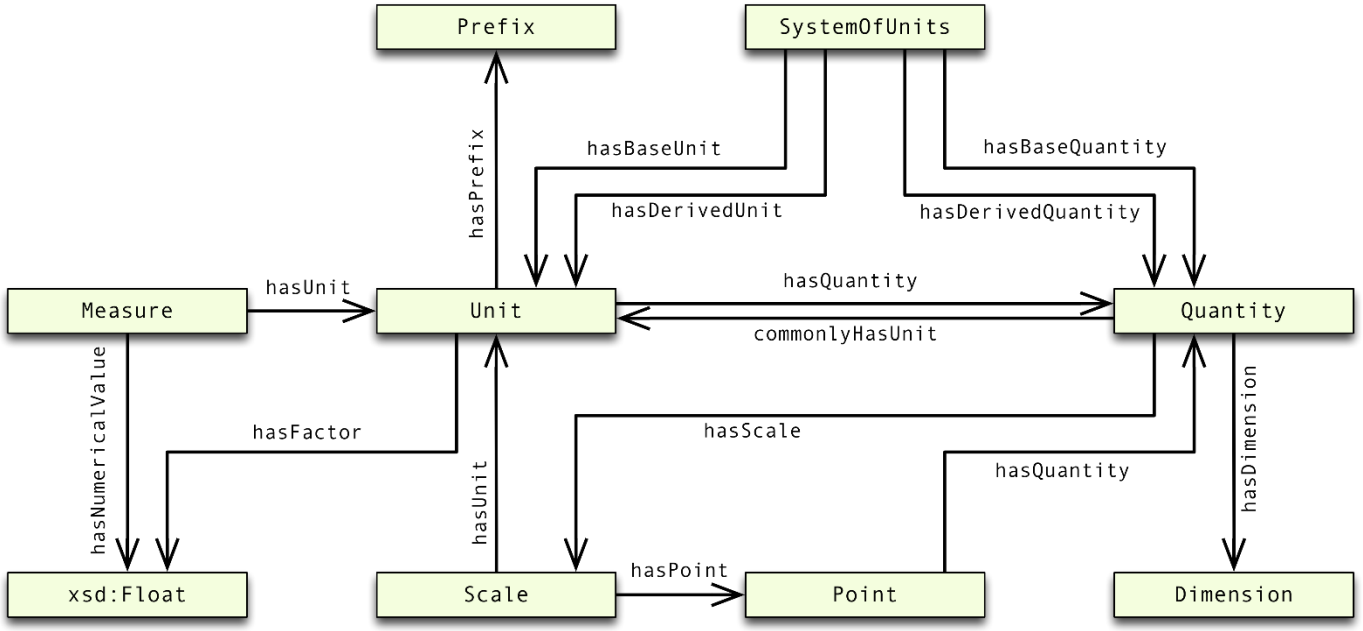
And what about physical quantities?

- Unit designations
  - **Units in the VO** (http://www.ivoa.net/documents/VOUnits/index.html)
    - IVOA Recommendation 23 May 2014
    - Very convenient standard, agreed with BIPM, ISO/IEC, IAU
  - Example:

m (metre)	**g** (gram)	**J** (joule)	**Wb** (weber)
s (second)	**rad** (radian)	**W** (watt)	**T** (tesla)
A (ampere)	**sr** (steradian)	**C** (coulomb)	**H** (henry)
K (kelvin)	**Hz** (hertz)	**V** (volt)	**lm** (lumen)
mol (mole)	**N** (newton)	**S** (siemens)	**lx** (lux)
cd (candela)	**Pa** (pascal)	**F** (farad)	**Ohm** (ohm)

Physical ontologies

- **QUDT** CATALOG – Quantities, Units, Dimensions and Data Types Ontologies (http://www.qudt.org/release2/qudt-catalog.html)
- **Ontology of units of Measure** (OM) 2.0 (http://www.ontology-of-units-of-measure.org/resource/om-2)
  - [#GitHub] HajoRijgersberg/OM (http://github.com/HajoRijgersberg/OM)



ISO standard for quantities and units

- ISO **80000**-1:2022 Quantities and units — Part 1: **General** (http://www.iso.org/standard/76921.html)
- ISO 80000-2:2019 Quantities and units — Part 2: **Mathematics** (http://www.iso.org/standard/64973.html)
- ISO 80000-3:2019 Quantities and units — Part 3: **Space and time** (http://www.iso.org/standard/64974.html)
- ISO 80000-4:2019 Quantities and units — Part 4: **Mechanics** (http://www.iso.org/standard/64975.html)
- …
- ISO 80000-12:2019 Quantities and units — Part 12: **Condensed matter physics** (http://www.iso.org/standard/63480.html)

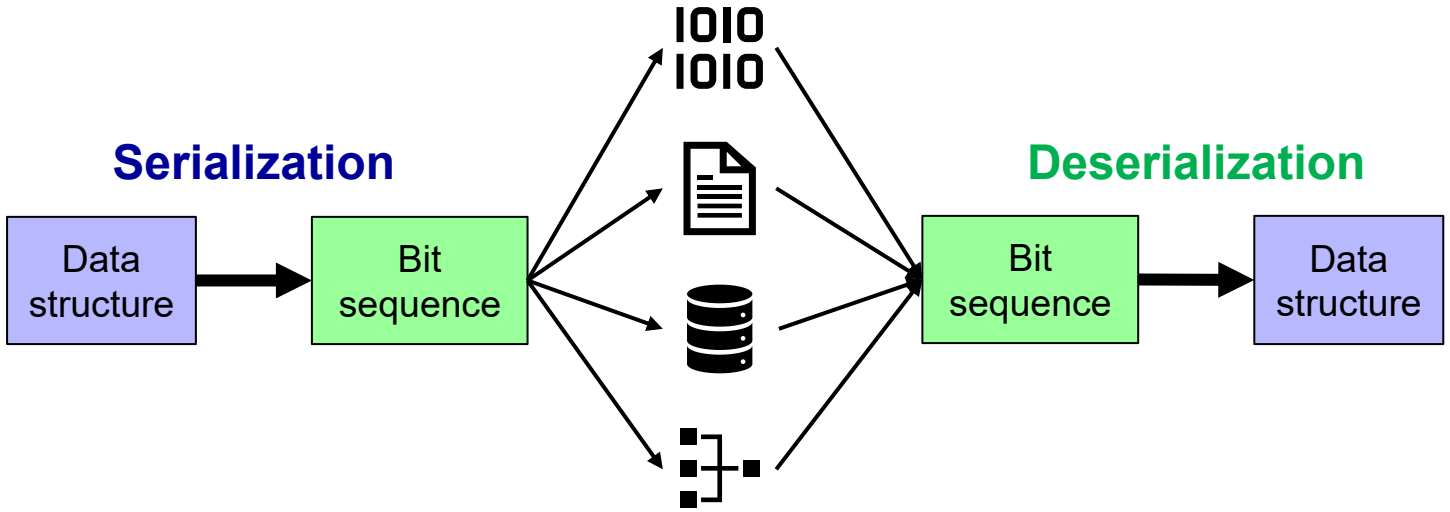
- IEC 80000-13:2008 Quantities and units — Part 13: **Information science and technology** (http://www.iso.org/standard/31898.html)

Serialization

- ✓ Main concepts
- ✓ CSV, XML, JSON, …
- ✓ Escaping
- ✓ Binary data in text documents
- ✓ Some modern formats
  - ✓ JSONLines, CBOR, HDF, ASDF

Serialization

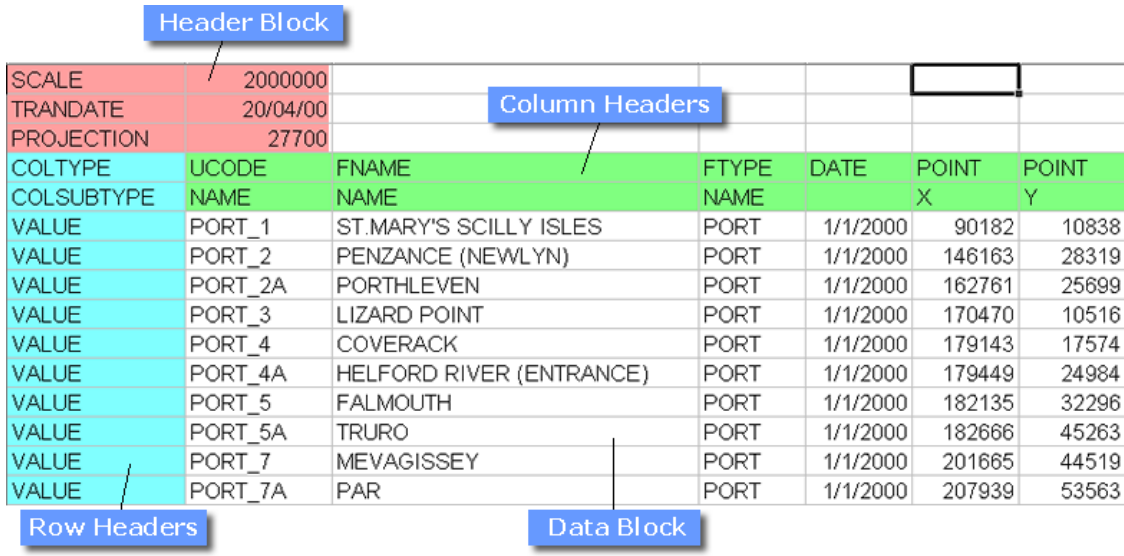
- **Serialization** – the transformation of a data structure into a bit sequence
- **Deserialization** – the restoration of a data structure from a bit sequence
  - It becomes necessary as soon as we lose the current execution context of program, computing node, …
  - Compare this to **direct copying** memory contents when we can use **absolute addresses** in memory
- Serialization – the heart of any high-level protocol!
  - Data format in a message



Comma-Separated Values (CSV)

- Informal description: CSV – Comma Separated Values (http://data.okfn.org/doc/csv)
- RFC **4180** – Common Format and MIME Type for Comma-Separated Values (CSV) Files (http://tools.ietf.org/html/rfc4180)
- RFC **7111** – URI Fragment Identifiers for the text/csv Media Type (http://tools.ietf.org/html/rfc7111)

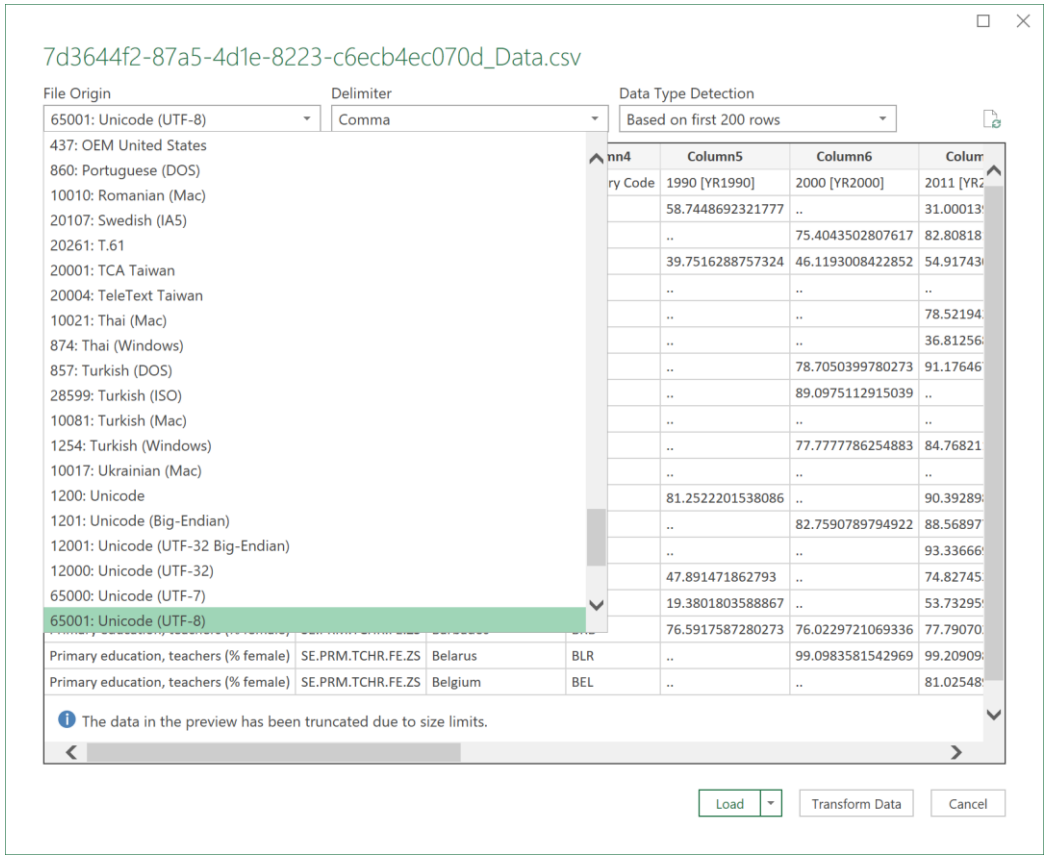
- Many extensions and specific CSV-based file formats!
  (Illustrative example: http://www.discoverysoftware.co.uk/SGHelp/dwLoader%20Format.htm)



SCALE	2000000					
TRANDATE	20/04/00					
PROJECTION	27700					
COLTYPE	UCODE	FNAME	FTYPE	DATE	POINT	POINT
COLSUBTYPE	NAME	NAME	NAME		X	Y
VALUE	PORT_1	ST.MARY'S SCILLY ISLES	PORT	1/1/2000	90182	10838
VALUE	PORT_2	PENZANCE (NEWLYN)	PORT	1/1/2000	146163	28319
VALUE	PORT_2A	PORTHLEVEN	PORT	1/1/2000	162761	25699
VALUE	PORT_3	LIZARD POINT	PORT	1/1/2000	170470	10516
VALUE	PORT_4	COVERACK	PORT	1/1/2000	179143	17574
VALUE	PORT_4A	HELFORD RIVER (ENTRANCE)	PORT	1/1/2000	179449	24984
VALUE	PORT_5	FALMOUTH	PORT	1/1/2000	182135	32296
VALUE	PORT_5A	TRURO	PORT	1/1/2000	182666	45263
VALUE	PORT_7	MEVAGISSEY	PORT	1/1/2000	201665	44519
VALUE	PORT_7A	PAR	PORT	1/1/2000	207939	53563

Note on character encoding

- A particularly scary issue in the recent past is character coding (string formats)
  - Legacy code pages → Unicode world
  - Things are much better now!
    - For example, in Microsoft Excel with the development of **Power Query**
    - Microsoft Power Query documentation – Text/CSV (http://learn.microsoft.com/en-us/power-query/connectors/text-csv)



EXtensible Markup Language (XML)

- Standard: **XML 1.0** (**5th** edition) W3C Recommendation 2008-11-26 (http://www.w3.org/TR/REC-xml)
  - **> 68** pages!
- W3Schools XML Tutorial (http://www.w3schools.com/xml)
- The Annotated XML 1.0 Specification 1998-02-10 (http://www.xml.com/axml/testaxml.htm)
- Namespaces in XML 1.0 (3rd edition) W3C Recommendation 2009-12-8 (http://www.w3.org/TR/REC-xml-names)
- Document Object Model (DOM) Level 2 Core Specification (Version 1.0) W3C Recommendation 2000-11-13 (http://www.w3.org/TR/DOM-Level-2-Core)
- Microsoft XML Standards Reference (https://msdn.microsoft.com/en-us/library/ms256177(v=vs.110).aspx)

Using XML: successes and challenges

- In the corporate sector, XML has won, but we should understand who!
  - All popular information buses from the 2000s
  - EDI (Electronic Data Interchange)
    - What is EDI (Electronic Data Interchange)? (http://www.edibasics.com/what-is-edi)
  - ebXML
    - ebXML Business Process Specification Schema и др. (https://www.oasis-open.org/standards#ebxmlbpv2.0.4)
    - EBXML | Online Community for Electronic Business Using XML (ebXML) Standards (http://ebxml.xml.org)
  - HL7 Version 3 Standard: XML Implementation Technology Specification (http://www.hl7.org/implement/standards/product_brief.cfm?product_id=163)
- Discussion:
  - One of the first adequate notes: XML Is Too Hard For Programmers (http://www.tbray.org/ongoing/When/200x/2003/03/16/XML-Prog)

Example of SVG-document

```
<?xml version="1.0" encoding="utf-8"?>
<!-- Generator: Adobe Illustrator 19.0.0, SVG Export Plug-In . SVG Version: 6.00 Build 0)  -->
<svg version="1.1" id="Layer_1" xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink"
x="0px" y="0px"  viewBox="0 0 558 558" style="enable-background:new 0 0 558 558;" xml:space="preserve">
<style type="text/css">
	.st0{fill:url(#XMLID_73_);}	.st1{fill:#FFFFFF;}
</style>
<g id="Layer_2">
	<linearGradient id="XMLID_73_" gradientUnits="userSpaceOnUse" x1="279" y1="558" x2="279" y2="-1.108856e-008">
		<stop  offset="0" style="stop-color:#2196F3"/>	<stop  offset="1" style="stop-color:#4FC3F7"/>
	</linearGradient>
	<path id="XMLID_5_" class="st0" d="M526,558H32c-17.6,0-32-14.4-32-32V32C0,14.4,14.4,0,32,  …  "/>
</g>
<g id="XMLID_9_">
	<path id="XMLID_10_" class="st1" d="M53.5,405h14.3V82.5H53.5v-18H118v18h-14.5V405H118v18H53.5V405z"/>
</g>
<g id="XMLID_4_">
	<path id="XMLID_18_" class="st1" d="M297,183.9c-11.1-13.8-24.1-20.8-39.2-20.8c-6.2,0-12.1, … "/>
</g>
<g id="XMLID_1_">
	<path id="XMLID_12_" class="st1" d="M234,102.9c-11.1-13.8-24.1-20.8-39.2-20.8c-6.2,0-12.1,  …  "/>
</g>
<g id="XMLID_14_">
	<path id="XMLID_15_" class="st1" d="M426,495h17.3L428,423.5h-88L324.5,495H342v18h-54.3v-18h21.5L382,154.5h26L480.8,495h18.5v18 H426V495z M343.5,405.5h81l-
40.5-189L343.5,405.5z"/>
</g>
<rect id="XMLID_6_" x="90" y="405" class="st1" width="48.7" height="18"/>
<rect id="XMLID_7_" x="297" y="477" class="st1" width="18" height="27"/>
<rect id="XMLID_8_" x="261" y="504" class="st1" width="27" height="9"/>
</svg>
```

XQuery – evil fate

- Powerful query language, Turing-complete!
  - One of the firs articles: **Kepser S.** A Proof of the Turing-completeness of XSLT and XQuery, Proceedings of Extreme Markup Languages, **2004**.
    - http://conferences.idealliance.org/extreme/html/2004/Kepser01/EML2004Kepser01.html
- Old standard: XQuery **1.0**: An XML Query Language (Second Edition)
  - W3C Recommendation 2010-12-14 (revised 2015-09-07) (http://www.w3.org/TR/xquery)
- Current standard: XQuery **3.1**: An XML Query Language
  - W3C Recommendation 2017-03-21 (http://www.w3.org/TR/xquery-3)
- Basics:
  - XQuery Tutorial (http://www.w3schools.com/xsl/xquery_intro.asp)
  - XQuery Language Reference в SQL Server (https://msdn.microsoft.com/ru-ru/library/ms189075(v=sql.120).aspx)

JavaScript Object Notation (JSON)

- Introducing JSON (http://json.org)
- ECMA-404 standard, 2013 (http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf)
  - New **2nd edition** (December 2017)
  - **7** pages!

- Features
  - Originally designed for **serialization**
  - Originally designed for **UTF8** only
  - Support **vectors** («**[**<>, …, <>,**]**»)

Example of JSON-document


```
{  "ObjsCount": 4,
   "ObjsNames":  [
     "o1", "o2", "o3", "o4"
   ],
   "AttrsCount": 5,
   "AttrsNames":  [
     "a1", "a2", "a3", "a4", "a5"
   ],
   "ContextType": "SPARSE",
   "ContextSparse": [
     { "ObjId": 0,
       "Attrs": [ 2, 4 ]
     },
     { "ObjId": 1,
       "Attrs": [ 0, 2, 3 ]
     },
     { "ObjId": 2,
       "Attrs": [ 1, 3 ]
     },
     { "ObjId": 3,
       "Attrs": [ 0, 3, 4 ]
     }
   ],
   "ContextDense": null
}
```

Will JSON be able to (fully) replace XML?

- There is a standard
  - JSON – **ECMA-404**
- There are binary representations for effective storage and processing
  - **BSON** (http://bsonspec.org), **UBJSON** (http://ubjson.org), etc.
- There are extensions and human-oriented versions
  - **JSON5** – "JSON for Humans" with some features from ECMA-262 (http://json5.org)
  - **Hjson** – a user interface for JSON (http://hjson.github.io)
  - **GeoJSON** (RFC 7946) for encoding a variety of geographic data structures (http://datatracker.ietf.org/doc/html/rfc7946)
- There are document schemas and validators
  - **JSON Schema** (http://json-schema.org, examples: http://json-schema.org/learn/json-schema-examples)
    - JSON Schema validator (http://www.jsonschemavalidator.net)
- There is Xpath analogue
  - **JSONPath** – XPath for JSON (http://goessner.net/articles/JsonPath/)
    - JSONPath Online Evaluator (http://jsonpath.com)
- There is XQuery analogue
  - **JSONiq** – The JSON Query language (http://www.jsoniq.org)

Escaping and escape sequences

- **Escape sequence** – a sequence of characters that does not represent itself when used inside a literal, but is translated into another character or a sequence of characters that may be difficult or impossible to represent directly

- ☑ Any formal language uses escaping of literals!

- W3C – Using character escapes in markup and CSS (http://www.w3.org/International/questions/qa-escapes)
- Microsoft – C Language Escape Sequences (http://docs.microsoft.com/en-us/cpp/c-language/escape-sequences)

Base64, Uuencode, Xxencode

- Binary data → Text. Why?
  - Insert into “text stream”
  - Invariability while transmission in case of recoding
- RFC 3548 – The Base16, Base32, and Base64 Data Encodings (https://tools.ietf.org/html/rfc4648)
- Base64 Decode and Encode Online (https://www.base64encode.org)
- The **MIME** (Multipurpose Internet Mail Extensions) specification suggests using Base64 as one of the main methods to represent binary data as text

Examples of binary data to text conversion

- 0000-0010: **12 e2 83 77-51 94 5b 56-d8 35 09 84-14 e2 de 77** (...wQ.[V .5.....w)
- *MIME Base64*:

```
MIME-Version: 1.0
Content-Type: application/octet-stream; name="MsSp7ru.dub"
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename="MsSp7ru.dub"

EuKDd1GUW1bYNQmEFOLedw==
```

- *UUE*:

```
begin 644 MsSp7ru.dub
0$N*#=U&46U;8-0F$%.+>=P``
`
end
sum -r/size 52992/16
```

- *XXE*:

```
begin 644 MsSp7ru.dub
E2i81Rp4IKpPMBEa23C9SRk++
+
end
sum -r/size 52992/16
```

XML CDATA

- XML language define CDATA section:
  - **<![CDATA[**Any_text**]]>**,
    where Any_text is treated as arbitrary text with **XML markup ignored**
    - Mnemonics and symbol codes are also ignored (for excample: `<` or ` `)
  - «CDATA» caption is formed by the abbreviation of “character data”
  - ☑ Text in the CDATA section may not contain “**]]>**”

- Sometimes this section is misinterpreted as binary data! You do not need to make such an error
- If you want to include binary data in an XML document, it is better to simply use Base-64 encoding in some “normal” tag

Automatic serialization

- OOP
  - Object persistence
  - Parameter marshalling at remote procedure calls
- Standard libraries of all modern programming languages support automatic serialization
  - But often additional libraries are faster and/or more functional
- Pitfalls
  - Data annotations
  - Custom serialization formats
- Microsoft .NET examples
  - JSON serialization and deserialization (marshalling and unmarshalling) (http://docs.microsoft.com/en-us/dotnet/standard/serialization/system-text-json-overview)
  - XML and SOAP serialization (http://docs.microsoft.com/en-us/dotnet/standard/serialization/introducing-xml-serialization)

JSON Lines – CSV with JSON records?

- **JSON Lines** text format, also called newline-delimited JSON (http://jsonlines.org)
  - UTF-8 Encoding, each Line is a Valid JSON Value, line terminator is "\n"
    - JSON Lines validator (http://jsonlines.org/validator)
  - Benefits:
    - Better than CSV
    - Self-describing data
      - JSON Lines + JSON Schema = ♥
    - Easy Nested Data
  - ISO uses it as standard download format (http://www.iso.org/open-data.html)

  - Stream compressors like **gzip** or **bzip2** are recommended for saving space, resulting in .jsonl.gz or .jsonl.bz2 files.
  - MIME type may be **application/jsonl**, but this is not yet standardized.

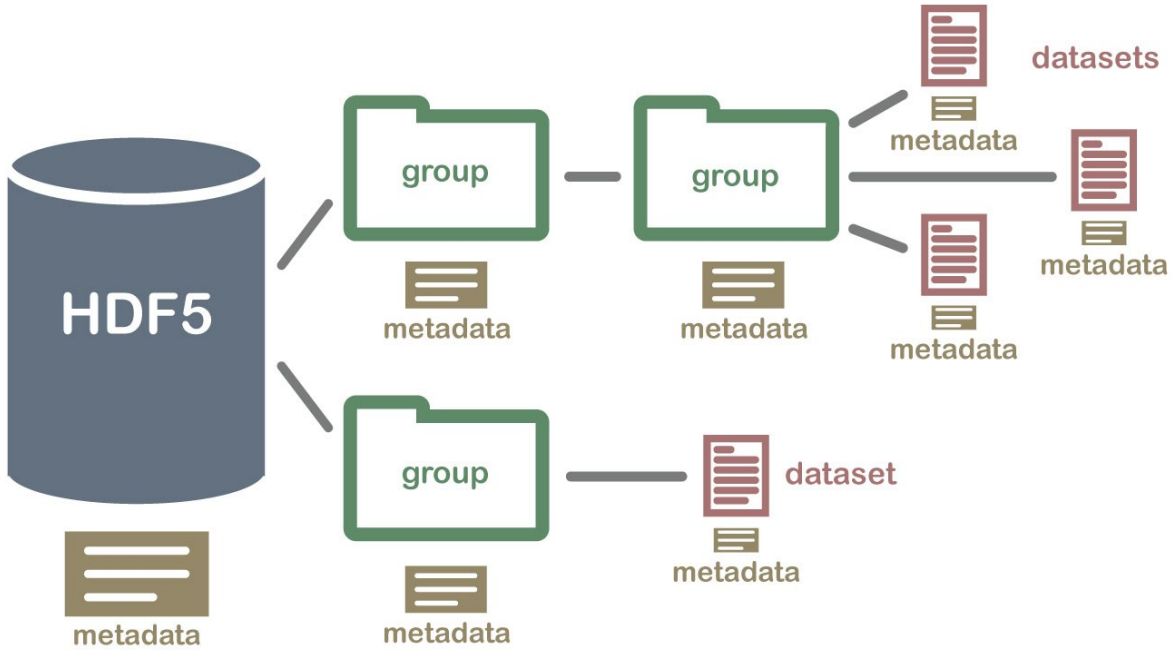
Other promising projects (1) – CBOR

- The Concise Binary Object Representation (**CBOR**) (http://cbor.io)
  - RFC 8949 (https://www.rfc-editor.org/rfc/rfc8949.html)
    - Obsoletes RFC 7049!
    - + RFC 8152: CBOR Object Signing and Encryption (**COSE**)
    - + RFC 8610: Concise Data Definition Language (**CDDL**)

Major Type	Meaning	Content
0	unsigned integer N	-
1	negative integer -1-N	-
2	byte string	N bytes
3	text string	N bytes (UTF-8 text)
4	array	N data items (elements)
5	map	2N data items (key/value pairs)
6	tag of number N	1 data item
7	simple/float	-

Other promising projects (2) – HDF

- **HDF5** (http://portal.hdfgroup.org/display/HDF5)
  - Developer: The HDF Group
  - Openness, convenient metadata, heterogeneity of datasets, automatic compression
  - Hierarchical Data Formats – What is HDF5? (http://www.neonscience.org/about-hdf5)
  - Pitfalls: Moving away from HDF5 (http://cyrille.rossant.net/moving-away-hdf5/)



Other promising projects (3) – ASDF

- Advanced Scientific Data Format (**ASDF**) (http://github.com/spacetelescope/asdf)
  - Developer: Space Telescope Science Institute
  - A next-generation interchange format for scientific data
  - In fact – one more JSON extension:
    - Reference implementation in Python
    - Python data types
    - Human-readable metadata description
  - Documentation: http://asdf-standard.readthedocs.io/

How to train your ~~dragon~~ serialization data format

- Wikipedia – Comparison of data-serialization formats (http://en.wikipedia.org/wiki/Comparison_of_data-serialization_formats)
- Veksler M. awesome-serialization (http://github.com/maximveksler/awesome-serialization)

- Mansfield S. Creating a Custom Serialization Format. 2017 (http://about.sourcegraph.com/go/creating-a-custom-serialization-format/)

- Exciting discussion (in Russian): http://habrahabr.ru/post/248147 (☺)

Fundamental Data Representation Checklist

- ✓ Numbers, date/time values, strings
- ✓ For binary and textual representation

Checklist for binary data representation

1. Integer numbers — 64 bits = 8 bytes according to **IEEE754**
2. Real numbers — double precision floating point according to **IEEE754**
   a. Comment: Sources: IEEE 754-2019 – IEEE Standard for Floating-Point Arithmetic (http://standards.ieee.org/content/ieee-standards/en/standard/754-2019.html)
3. Date/Time — support all capabilities of **ISO 8601** (conversion to UTC and time zones)
   a. Comment 1: Interesting choice – Temporenc (http://temporenc.org)
   b. Comment 2: If you select unix time stamp you should use 64 bit!!! See The Year 2038 problem and TimeStamp Converter (http://www.unixtimestamp.com)
4. Strings — **UTF-8**
   a. Comment 1: sources: Section 3.9 in **The Unicode® Standard** (http://www.unicode.org/versions/Unicode15.0.0/ch03.pdf#G7404) and former Appendix D in **ISO 10646**, which became Section 9.1 (http://unicode.org/L2/L2010/10038-fcd10646-main.pdf)
   b. Comment 2: see also RFC 3629 (http://tools.ietf.org/html/rfc3629)
   c. Comment 3: you need to consider the requirements for **BOM** (Byte Order Mark)!
5. Unique identifiers — specific unique 128-bit integer value: GUID (Globally Unique Identifier) or UUID (Universally Unique Identifier)
   a. Comment 1: Universally Unique IDentifiers (UUIDs) **RFC 9562** (http://datatracker.ietf.org/doc/rfc9562/) – obsoletes RFC 4122

Checklist for textual data representation (1)

1. Base encoding — **UTF-8**
   a. Comment 1: sources: Section 3.9 в **The Unicode® Standard** (http://www.unicode.org/versions/Unicode15.0.0/ch03.pdf#G7404) and former Appendix D in **ISO 10646**, which became Section 9.1 (http://unicode.org/L2/L2010/10038-fcd10646-main.pdf)
   b. Comment 2: see also RFC **3629** (http://tools.ietf.org/html/rfc3629)
   c. Comment 3: you need to consider the requirements for BOM (Byte Order Mark)!
2. Integer numbers — preferably of arbitrary length or at least 64 bits = 8 bytes
3. Real numbers — minimum double precision floating point according to **IEEE754**
   a. Comment 1: sources: RFC **6340** (http://tools.ietf.org/html/rfc6340)
   b. Comment 1: if you need **accurate conversion**, consider using Hexadecimal floating literals for C++ (http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2016/p0245r0.html)
   c. Comment 3: Check **country/region differences** in formatting: decimal separation, number grouping and separation, etc. (http://learn.microsoft.com/en-us/globalization/locale/number-formatting)
4. Date/Time — full format with separators according to **ISO 8601**
   a. Comment 1: sources: ISO 8601-1:2019 Date and time — Representations for information interchange — Part 1: Basic rules (http://www.iso.org/standard/70907.html) и ISO 8601-1:2019 Date and time — Representations for information interchange — Part 2: Extensions (http://www.iso.org/standard/70908.html)
   b. Comment 2: Russian GOST 7.0.64-2018 (ISO 8601:2004) «Система стандартов по информации, библиотечному и издательскому делу. Представление дат и времени. Общие требования» (http://docs.cntd.ru/document/1200159341)

Checklist for textual data representation (2)

5. Strings — with normal Unicode support and escaping
   a. Comment: sources: **The Unicode® Standard** (http://www.unicode.org)

6. String escaping — accurate formal description without “gray areas”
   a. Preferably **strict schema**
      i. Comment: this allows you to always distinguish between an empty string (for example: `""`) and an indeterminate (unspecified) value
   b. Preferably according to Section 7 **RFC 8259** (http://tools.ietf.org/html/rfc8259)
      i. Comment: RFC 4627 (http://www.ietf.org/rfc/rfc4627.txt) and RFC 7159 (http://tools.ietf.org/html/rfc7159) was deprecated! But RFC 7159 is a malfeasance :)

7. Indeterminate values — blank (really nothing)
   a. Comment: see sense of strict escaping

8. Unique identifiers — full format of GUID/UUID text string with separators without brackets
   a. Comment 1: sources: Section 4. **UUID Format** in Universally Unique IDentifiers (UUIDs) **RFC 9562** (http://datatracker.ietf.org/doc/rfc9562/) – obsoletes RFC 4122
   b. Comment 2: example: `"f81d4fae-7dec-11d0-a765-00a0c91e6bf6"`

Sources

1. **Petzold C.** Code: The Hidden Language of Computer Hardware and Software. 2nd ed. Microsoft Press, 2022. 480 p.
2. **Fenwick P.** Introduction to Computer Data Representation. Bentham Science Publishers, 2018. 270 p.
3. **Soukup J., Macháček P.** Serialization and Persistent Objects: Turning Data Structures into Efficient Databases. Springer, 2014. 263 p.
4. **Chappell D.** Enterprise Service Bus. O'Reilly, 2004. 352 p.
5. **European Commission**. New European Interoperability Framework. European Union, 2017. (http://ec.europa.eu/isa2/sites/default/files/eif_brochure_final.pdf)

Sources (Internet)

1. A Tutorial on Data Representation Integers, Floating-point Numbers, and Characters (https://www3.ntu.edu.sg/home/ehchua/programming/java/DataRepresentation.html)
2. Data Representation (http://www.csfieldguide.org.nz/en/chapters/data-representation.html)
3. UNICODE (http://www.unicode.org)
4. Introducing JSON (http://json.org)
5. W3School – XML (http://www.w3schools.com/xml)
6. Serializing Python Objects (http://www.diveintopython3.net/serializing.html)
7. Kraus A. The Definitive Serialization Performance Guide. 2018 (http://aloiskraus.wordpress.com/2017/04/23/the-definitive-serialization-performance-guide/)
8. Kraus A. .NET Serialization Benchmark 2019 Roundup. 2019 (http://aloiskraus.wordpress.com/2019/09/29/net-serialization-benchmark-2019-roundup/)

But it's not the end yet?!

Questions? Comments? Remarks?

- Contacts:
  - Alexey Neznanov, PhD
    - E-mail: aneznanov@hse.ru
    - Web-site: http://hse.ru/staff/aneznanov