Academic Year 2023/2024

Social Media Analytics (SMA)

Internet and Web Technologies

Marco Viviani

University of Milano-Bicocca

Department of Informatics, Systems, and Communication

UNIVERSITÀ DEGLI STUDI DI MILANO BICOCCA

DIPARTIMENTO DI INFORMATICA, SISTEMISTICA E COMUNICAZIONE

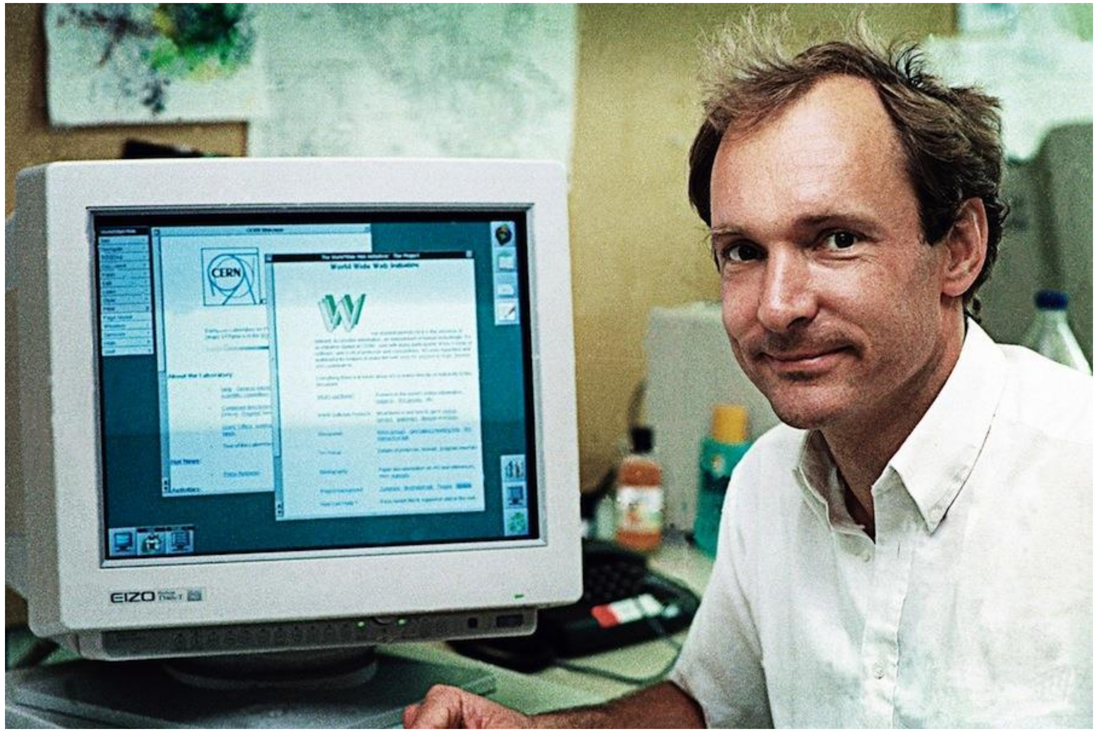
Internet

- **Internet** is the global system of interconnected computer networks that use the Transmission Control Protocol/Internet Protocol (TCP/IP) protocol suite to connect devices around the world (network of networks)
- **TCP/IP protocols** allow devices connected through the Internet to communicate with each other at a high level, regardless of the underlying hardware and software architecture, thus ensuring interoperability between different physical systems and sub-networks
- The Internet provides a wide range of resources and **services**, such as e-mail, file sharing, and the **World Wide Web**

The World Wide Web – 1

- The World Wide Web is one of the major **services** of the Internet

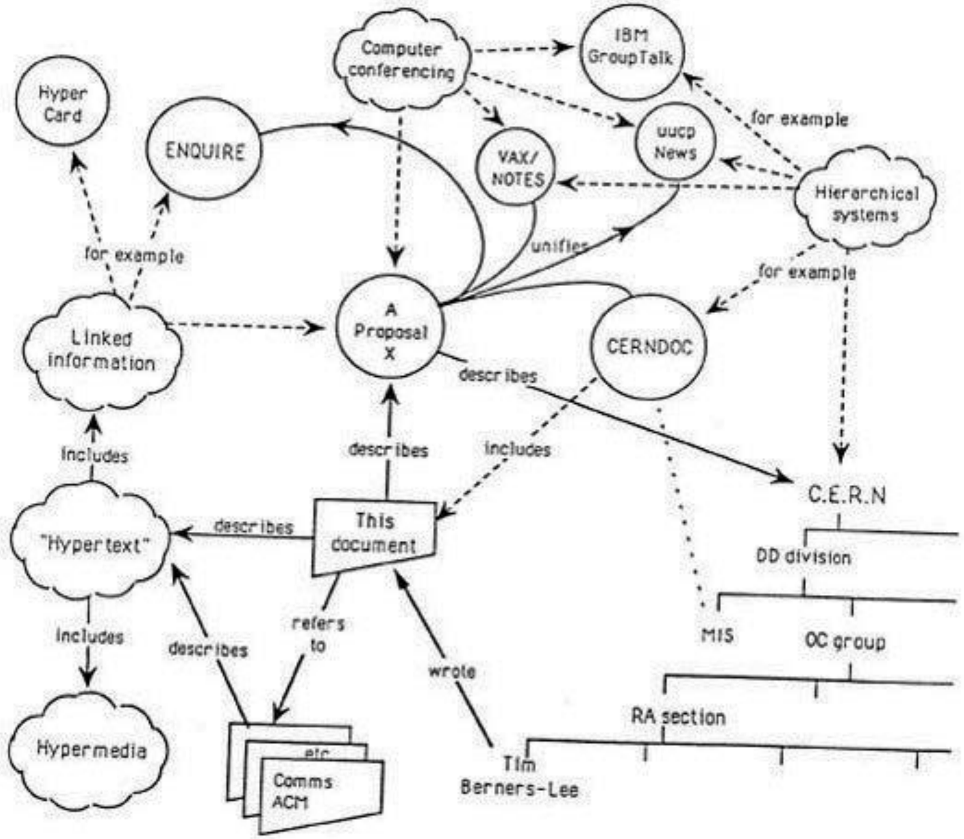
- The conception of the World Wide Web is due to **Tim Berners-Lee**, elaborating a previous project (with Robert Cailliau) of sharing scientific documentation in electronic format regardless of the platform



The World Wide Web – 2

- The proposal was designed to provide a more effective communication system within the European Organization for Nuclear Research (CERN: *Conseil Européen pour la Recherche Nucléaire*), but Berners-Lee soon realized the global potential of the idea

The historic WWW logo designed by Robert Cailliau in 1990



WWW architecture in Berners-Lee's original proposal

The World Wide Web – 3

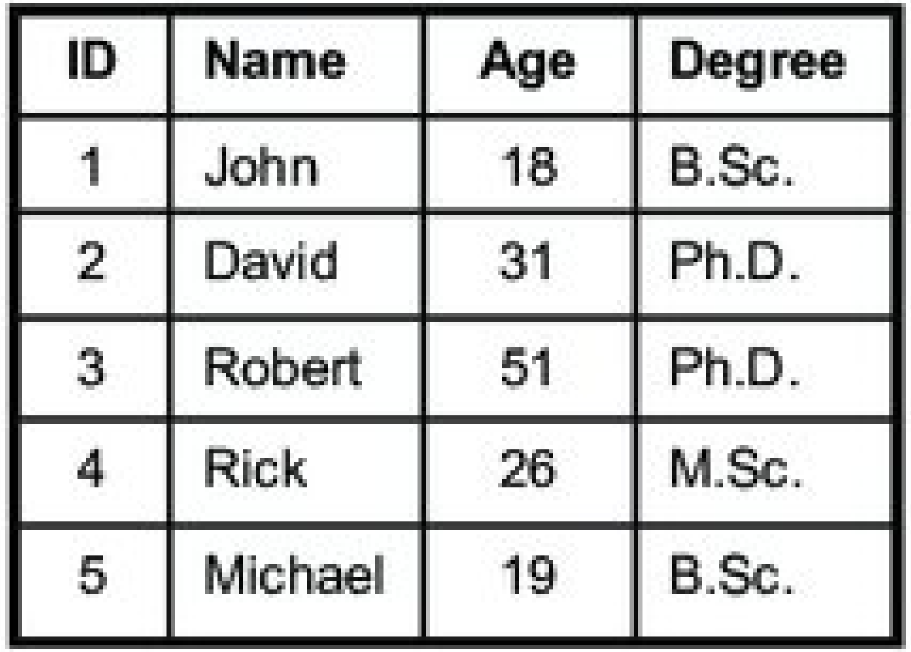
- The Web allows to browse and take advantage of a very vast set of amateur and professional (multimedia) **content** connected to each other and of further services accessible to all or a selected part of Internet users

- **Uniform Resource Identifier** (URI): sequence of characters that uniquely identifies a generic resource. Examples of URIs are: a Web address (Uniform Resource Locator, URL), a document, an image, a file, an e-mail address, etc.

- The fundamental concepts behind the Web are that of **hypertext** and **hyperlink**. Hypertext is semi-structured text that uses logical links between pages that contain text (and other multimedia content)

A parenthesis: Structured, unstructured, semi-structured data

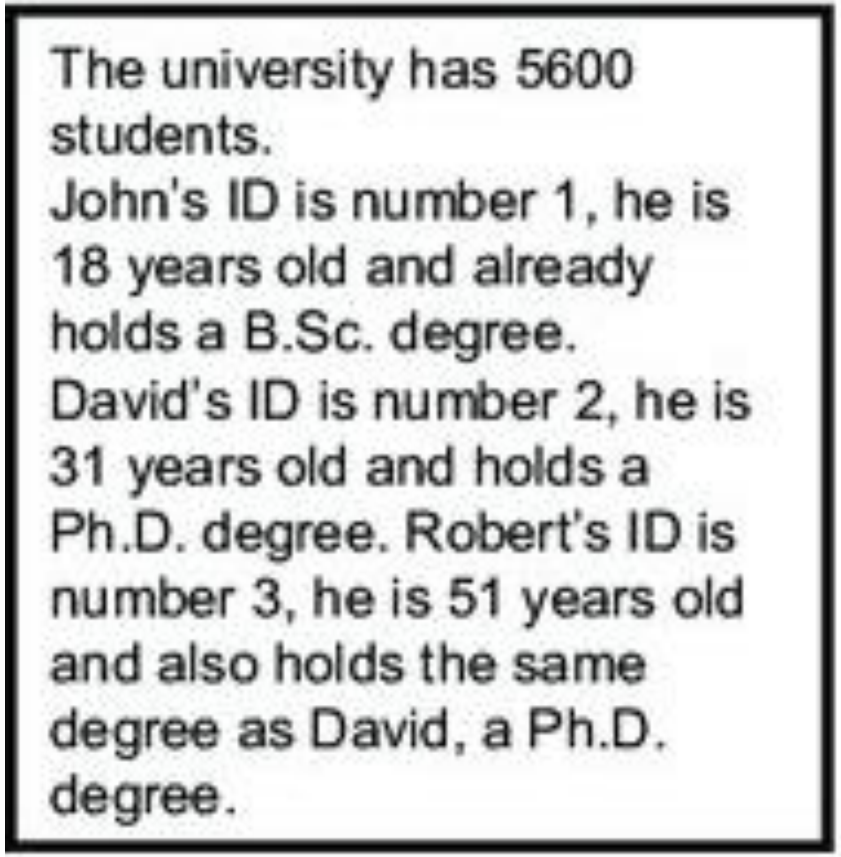
- **Structured data**
  - Data stored in **databases**, organized according to rigid schemes and tables
  - This is the most suitable type of data for **relational information management** models



ID	Name	Age	Degree
1	John	18	B.Sc.
2	David	31	Ph.D.
3	Robert	51	Ph.D.
4	Rick	26	M.Sc.
5	Michael	19	B.Sc.

A parenthesis: Structured, unstructured, semi-structured data

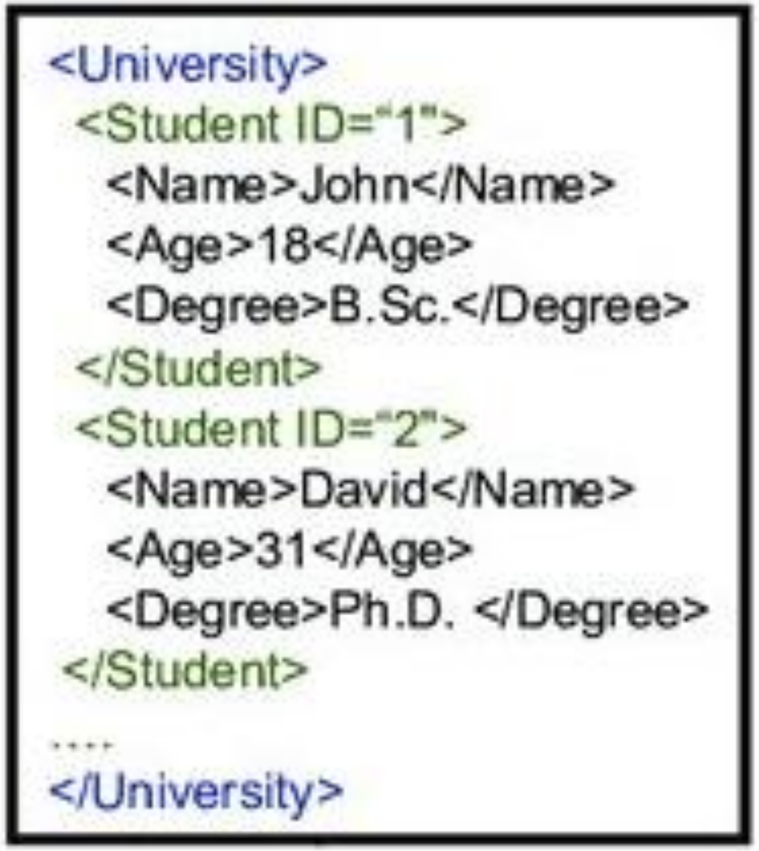
- **Unstructured data**
  - Data stored without any scheme. An example could be narrative texts produced by means of one of the most popular text editing software
  - In this case, the data management systems that can be used are those based on **Information Retriveval** models



The university has 5600 students.
John's ID is number 1, he is 18 years old and already holds a B.Sc. degree.
David's ID is number 2, he is 31 years old and holds a Ph.D. degree. Robert's ID is number 3, he is 51 years old and also holds the same degree as David, a Ph.D. degree.

A parenthesis: Structured, unstructured, semi-structured data

- **Semi-structured data**
  - A form of structured data that does not obey the tabular structure of data models associated with relational databases or other forms of data tables
  - Nonetheless, it contains **tags** or other **markers** to separate semantic elements and enforce hierarchies of records and fields within the data. Therefore, it is also known as self-describing structure.



```
<University>
 <Student ID="1">
   <Name>John</Name>
   <Age>18</Age>
   <Degree>B.Sc.</Degree>
 </Student>
 <Student ID="2">
   <Name>David</Name>
   <Age>31</Age>
   <Degree>Ph.D. </Degree>
</Student>
....
</University>
```

The World Wide Web – 4

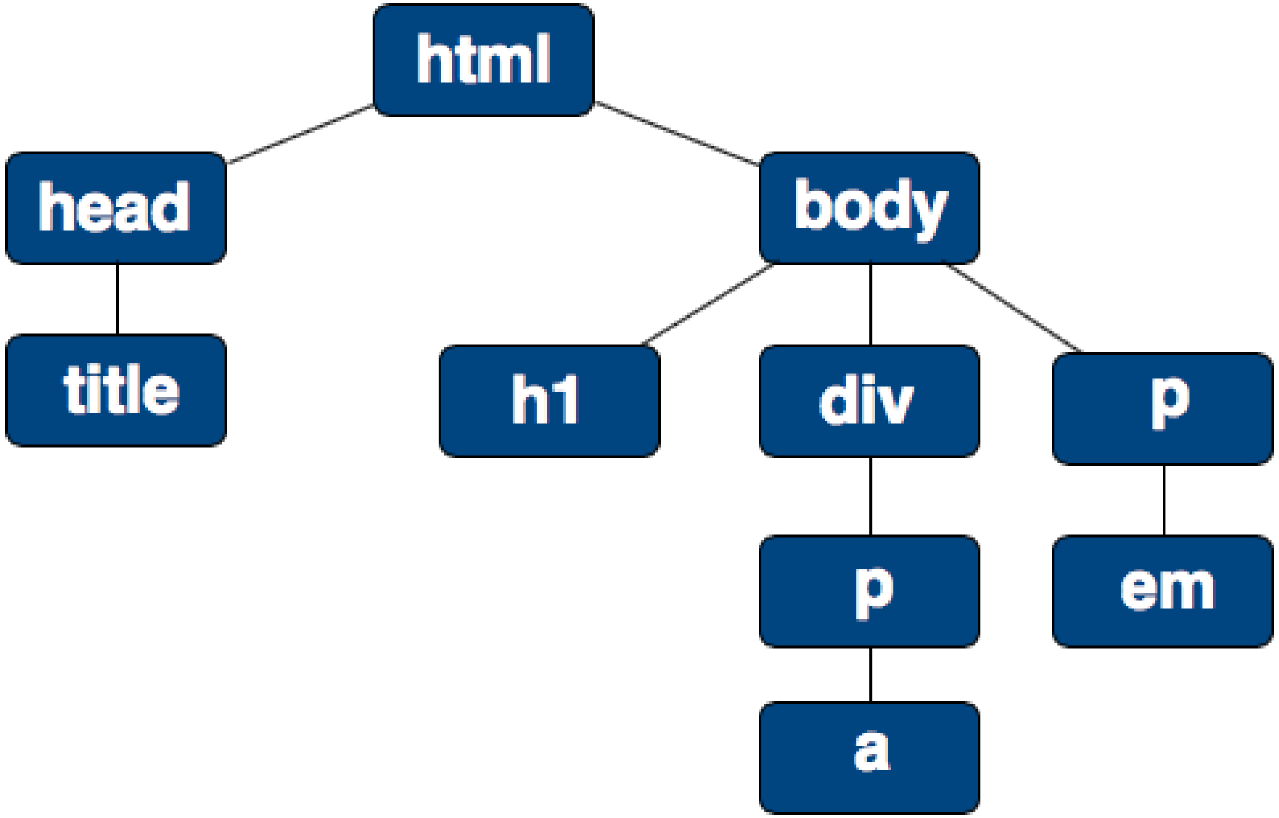
- Berners-Lee developed the **HyperText Markup Language** (HTML) publishing language, through which it is possible to indicate in the text the structure, the semantic meaning, or the presentation method
  - The language behind Web pages
  - A **markup language** is a computer language that uses tags to define elements within a document. It is human-readable, meaning markup files contain standard words, rather than typical programming syntax

- An HTML page is represented by a **text file**, which is a file that we can edit with programs such as Notepad and generally have a name that ends with the **.html** extension

The tree structure of an HTML document

```
<html>
  <head>
    <title>Structure of an HTML document</title>
  </head>
  <body>
    <h1>Title</h1>
    <div>
      <p>First <a href="pagina.htm">paragraph</a>.</p>
    </div>
    <p>Second <em>paragraph</em>.</p>
  </body>
</html>
```

The tree structure of an HTML document



Other semi-structured data formats

- **Extensible Markup Language** (XML)
  - XML is a markup language much like HTML
  - XML was designed to store and transport data
  - XML was designed to be self-descriptive

- **JavaScript Object Notation** (JSON)
  - JSON is a syntax for storing and exchanging data
  - JSON is text, written with JavaScript object notation

A brief intro to XML

- The basic unit of any XML document are **elements**

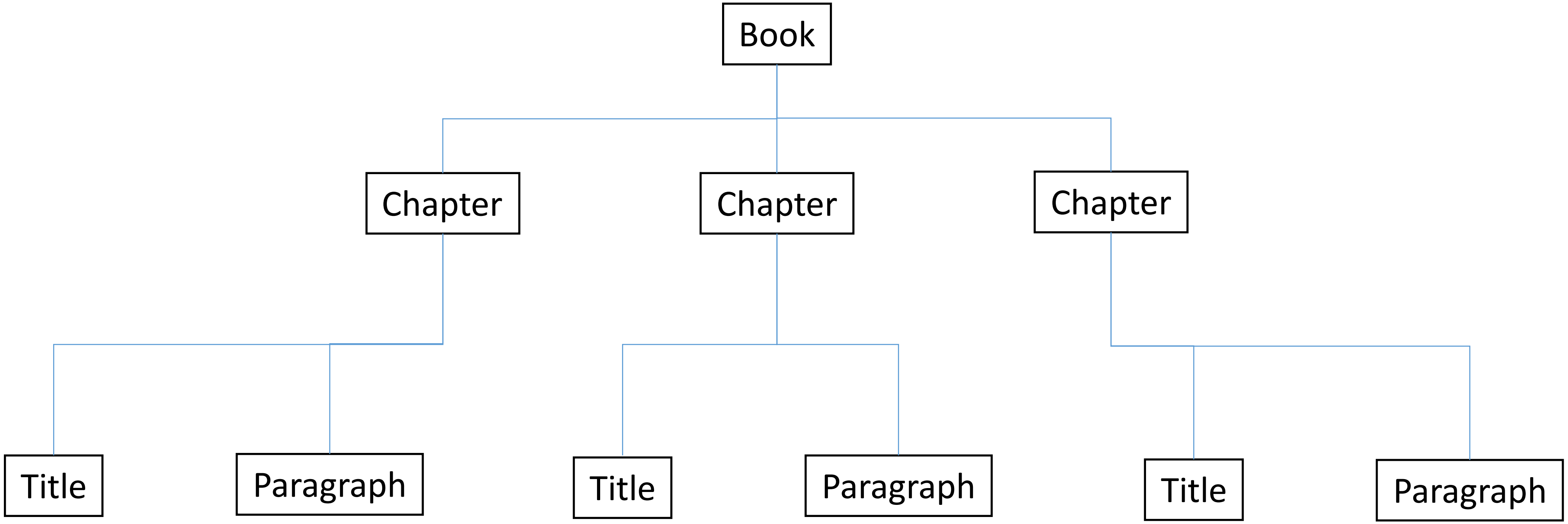
- An element corresponds to a specific block of text within a document, marked with a pair of **tags**
  - Start tag, End tag
  - <paragraph> Paragraph text </paragraph>

- An **element** therefore consists of a pair of tags and its content

Structure of an XML document

- An XML document is inherently characterized by a **hierarchical structure**
- The organization of the elements follows a hierarchical or arboreal order that includes a main element, called the **root element** or simply root
- The root contains all the other elements of the document. We can graphically represent the structure of an XML document through a **tree**, generally known as a document tree

An example of an XML tree structure



An example of an XML tree structure

```
<book>
       <chapter>
              <title>Title of the chapter</title>
              <paragraph>Text of the paragraph 1</paragraph>
              <paragraph>Text of the paragraph 2</paragraph>
       </chapter>
       <chapter>
              ...
       </chapter>
       <chapter>
              ...
       </chapter>
</book>
```

Document Type Definition

- The **Document Type Definition** (DTD) define the components allowed in the construction of an XML document
  - Defines the legal elements and attributes within the document. You cannot use any other elements except those defined. A kind of "vocabulary" for the files that will use it
  - It defines the structure of each element (e.g., what each element can contain, the order, the quantity of elements that can appear, etc.). A kind of "grammar"
  - Finally, it provides some mechanisms to simplify document management (e.g., the ability to import parts of other DTDs)

A simple example of DTD

- The following DTD:

```
<! ELEMENT person (name, surname)>
<! ELEMENT name (#PCDATA) >  (PCDATA stands for "Parsed Character Data")
<! ELEMENT surname (#PCDATA) >
```

defines an XML structure as follows:

```
<person>
      <name>Marco</name>
      <surname>Viviani</surname>
</person>
```

XML schema

- An **XML Schema** describes the structure of an XML document
  - The XML Schema language is also referred to as XML Schema Definition (XSD)
  - XML Schema is an XML-based (and more powerful) alternative to DTD.

- The **purpose of an XML Schema** is to define the legal building blocks of an XML document:
  - The elements and attributes that can appear in a document
  - The number of (and order of) child elements
  - Data types for elements and attributes
  - Default and fixed values for elements and attributes

An example of a XSD

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

<xs:element name="note">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="to" type="xs:string"/>
      <xs:element name="from" type="xs:string"/>
      <xs:element name="heading" type="xs:string"/>
      <xs:element name="body" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

</xs:schema>
```

A brief intro to JSON

- JavaScript Object Notation is a schema-less, text-based representation of structured data that is based on **name(key)-value pairs** and **ordered lists**

- **JSON Data**: A `name` and a `value`
  - A name/value pair consists of a field name (in double quotes), followed by a colon, followed by a value
  - `"firstName":"Marco"`

- **JSON Objects**
  - JSON objects are written inside curly braces
  - `{"firstName":"Marco", "lastName":"Viviani"}`

A brief intro to JSON

- **JSON Arrays**
  - JSON arrays are written inside square brackets

  -
    ```
    "teachers":[
      {"firstName":"Marco", "lastName":"Viviani"},
      {"firstName":"Davide", "lastName":"Mancino"}
    ]
    ```

XML vs JSON

XML

```xml
<teachers>
  <teacher>
    <firstName>Marco</firstName>
    <lastName>Viviani</lastName>
  </teacher>
  <teacher>
    <firstName>Davide</firstName>
    <lastName>Mancino</lastName>
  </teacher>
</teachers>
```

JSON

```json
"teachers":[
  {"firstName":"Marco",
   "lastName":"Viviani"},
  {"firstName":"Davide",
   "lastName":"Mancino"}
]
```

XML vs JSON

- Both XML and JSON can be used to **exchange data** on the Web

- JSON **is like** XML Because
  - Both JSON and XML are "self describing" (human-readable)
  - Both JSON and XML are hierarchical (values within values)
  - Both JSON and XML can be parsed and used by lots of programming languages

- JSON **is unlike** XML Because
  - JSON does not use the end tag
  - JSON is shorter (less verbose)
  - JSON is quicker to read and write
  - JSON can use arrays

XML, JSON and Databases

- Many **database management systems** (DBMS) use nowadays JSON to store data

- Most are from the **NoSQL** breed of DBMS
    - MongoDB actually stores the data in BSON format, which is an extension of JSON.
        - MongoDB Tutorial: https://www.quackit.com/mongodb/tutorial/

- Some **relational databases** use XML to store their data, but also have a certain level of support for JSON
    - MySQL, Oracle, PostgreSQL, and SQL Server now offer JSON support

The World Wide Web – 5

- **Transfer Protocol**: **HyperText Transfer Protocol** (HTTP) is an application layer protocol used as the primary system for transmitting information on the Web

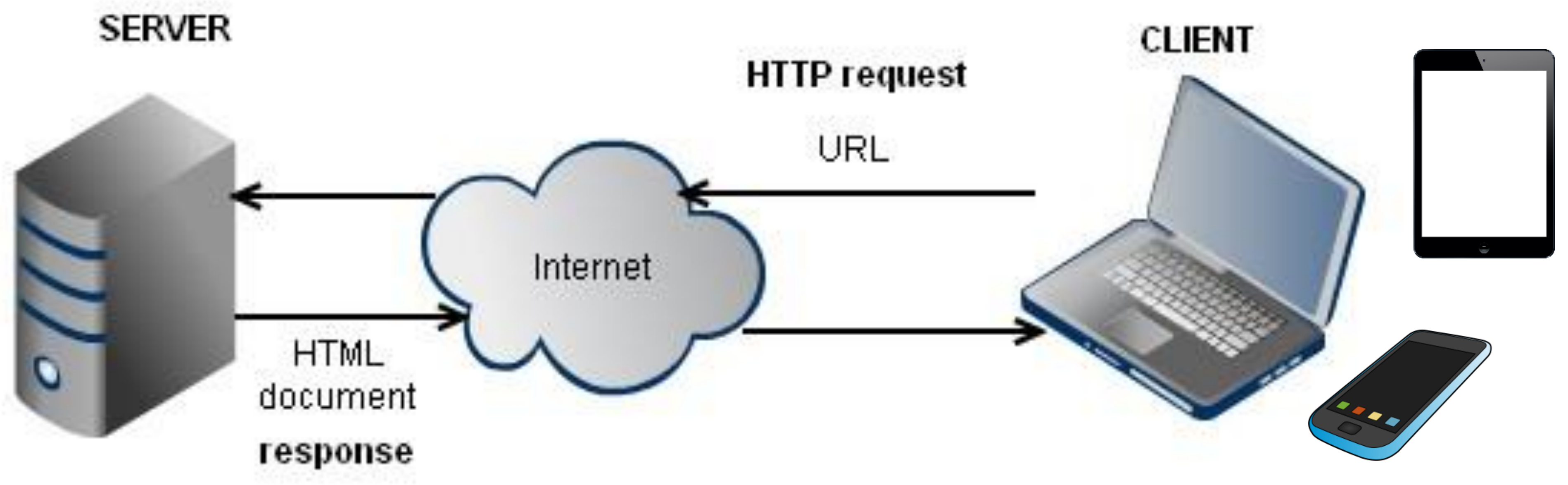
The Client/Server model

- HTTP is a protocol that works with a **Client/Server architecture**: the client executes a request, and the server returns the response

- In common use, the **client** is the browser and the **server** is the remote machine on which the Website resides.

- There are therefore **two types of HTTP messages**: request messages and response messages.

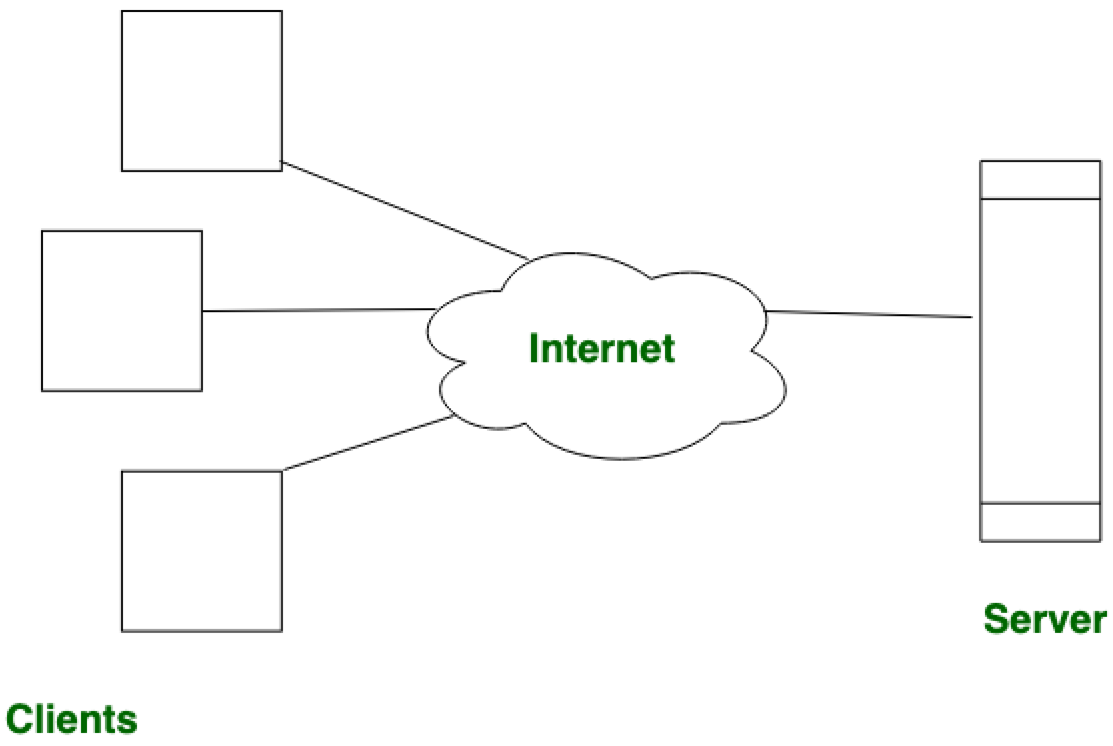
The Client/Server model



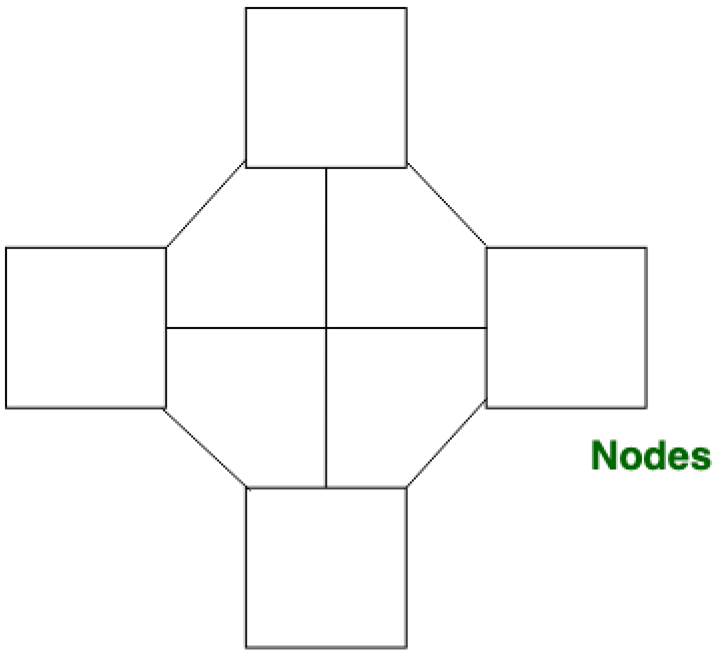
What about «peer-to-peer»?

- In a **peer-to-peer** (p2p) network, resources are shared among peers without any central coordination by a server.
- Peers act as both providers and consumers of resources. The idea behind peer-to-peer networks is to share resources economically.
- There is no centralized security scheme, and end users themselves can control access to resources, reducing security in peer-to-peer networks.
- Users can create any sharing point they want on their own computer, and security can only be provided by assigning a password when they create the sharing point.

Client-server VS peer-to-peer (1)



Client-Server Network Model



Peer-to-Peer Network Model

Client-server VS peer-to-peer (2)

Client-Server Networks	**Peer-to-Peer Networks**
In client-server networks, clients and servers are differentiated.	In peer-to-peer networks, clients and servers are not differentiated.
In client-server networks, centralized servers are used to store data.	In peer-to-peer networks, each peer has its own data.
In client-server networks, servers respond with the services which are requested by clients.	In peer-to-peer networks, each and every node can do both requests and respond for the services.
Client-server networks are costlier than peer-to-peer networks.	Peer-to-peer networks are more expensive than client-server network.
Client-server networks are more stable than peer-to-peer networks.	Peer-to-peer networks are less stable than client-server networks, especially when the number of peers increases.

Decentralized Web? (1)

- The infrastructure of the Web is “de facto” **centralized** in at least two major ways:
  - There are relatively few **major user-facing content distribution platforms** (Google, YouTube, Facebook, Twitter, TikTok, etc.) and they clearly have outsized power over people's ability to get their message amplified.
  - The easiest way to build any large-scale system—and almost the only economical way unless you are very well-funded—is to run it on one of a relatively small number of **infrastructure providers**, such as Amazon Web Services, Google Cloud Platform, Cloudflare, Fastly, etc., who already have highly scalable geographically distributed systems.

https://educatedguesswork.org/posts/challenges-web-decentralization/

Decentralized Web? (2)

"The Decentralized Web is like the World Wide Web we have today, filled with amazing, interactive content and information — the only difference is that the underlying architecture is decentralized, so that it becomes much harder for any one entity (whether through malicious censorship or accidental failure) to take down any single Web page, website, or service."

Jeremy Gillula
Staff Technologist, Electronic Frontier Foundation

https://ischool.syr.edu/decentralized-web-experts/

Decentralized Web? (3)

"A Decentralized Web is a network of resources in which no one player can control the conversation or spin it to [his or her] exclusive advantage."

Simon St. Laurent
Strategic Content Director, O'Reilly Media, Inc.

https://ischool.syr.edu/decentralized-web-experts/

Web Evolution and Classification

Static Web

- It consists of **static Web pages**. A static page is sent to the user exactly as stored on the server. It is built using HTML code and offers the same presentation and the same content, regardless of user identity or other factors

- When the user of a site visits a page, what happens is that the server on which the site resides sends the HTML file to the browser (Chrome, Mozilla, etc.)
  - The browser knows how to decode the file, and therefore shows the contents of the page on the user's screen

- Static web pages are suitable for content that **never or rarely needs updating**. Maintaining a large number of static pages can be impractical without automated tools, such as static (template-based) Website builders

Dynamic Web

- It consists of **dynamic Web pages**. Dynamic pages, instead of containing (only) HTML code, contain particular programs, known as **scripts**

- **Scripts**: programs written in various possible scripting languages such as JavaScript, PHP, ASP, typically inserted within the HTML Web page and which, at the user's particular request, are activated and processed on the client side via browser or server side, returning the content dynamic in the form of HTML code then interpreted by the browser and displayed to the user.

Client-side scripting

- Modifies a specific Web page **in response to mouse or keyboard actions** or at specified timing events

- In this case, the dynamic behavior occurs within the **presentation layer**

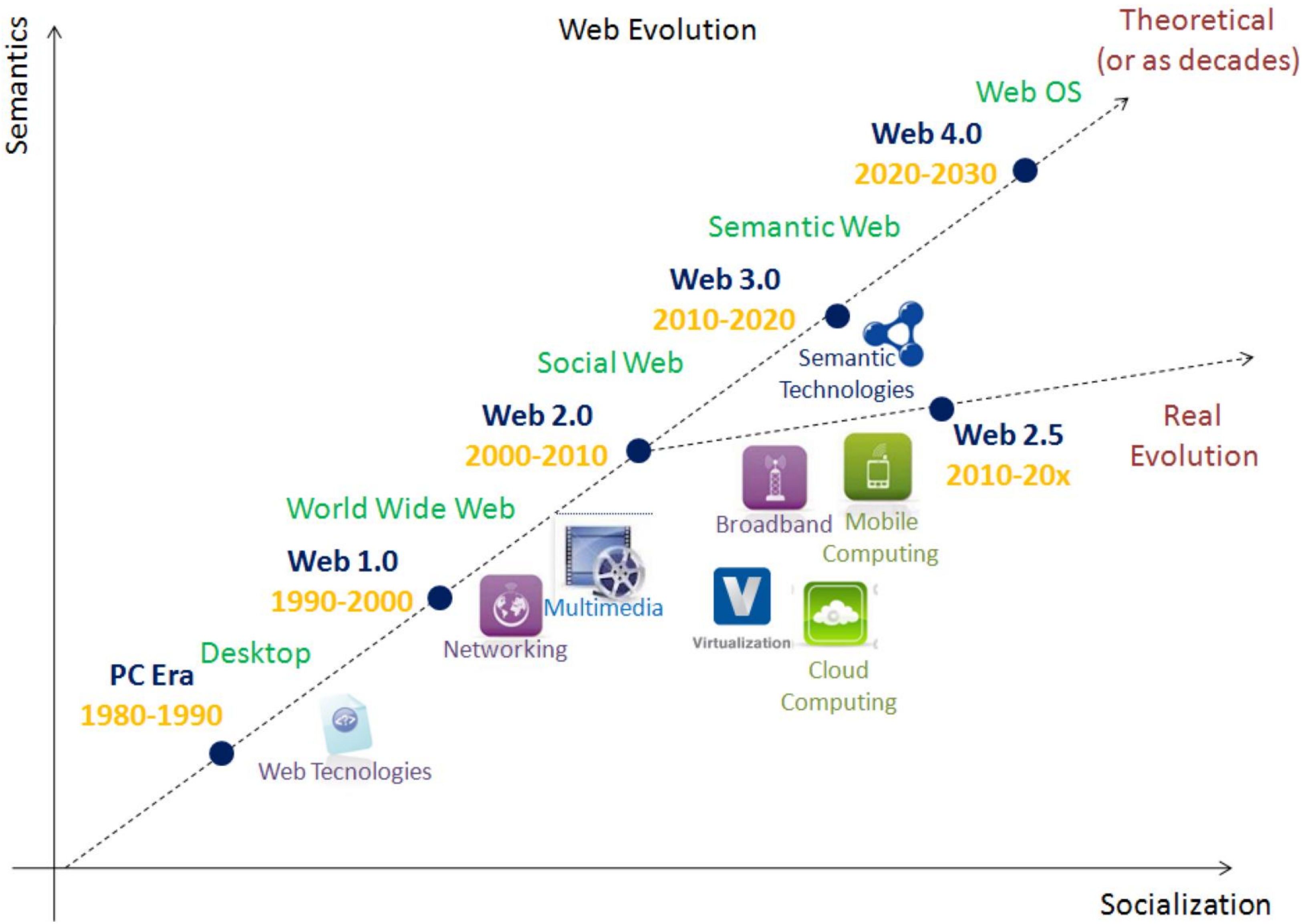
- **Client-side content** is generated on the user's local device. One of the most widely used client-side scripting languages is **JavaScript**

Server-side scripting

- A script running **on a Web server** (server-side scripting) is used to generate Web content on various Web pages, manage user sessions, and control workflow

- The most popular **server-side languages** are PHP, ASP and JSP (Java Server Page). The server-side scripting code is not made available to the user, who can only view the result of the script processing (typically HTML code)

The Web evolution



Web 1.0

*“Web 1.0 refers to the first stage in the World Wide Web, which was entirely made up of web pages connected by hyperlinks. Although the exact definition of Web 1.0 is a source of debate, it is generally believed to refer to the web when it was a set of static websites that were not yet providing interactive content. In Web 1.0, applications were also generally proprietary”.**

*https://www.techopedia.com/definition/27960/web-10

Web 1.0

- **Web 1.0** is a **retronym** that refers to the first stage of the evolution of the World Wide Web

- In Web 1.0 most users can enjoy the contents but only a small part (the experts) become content generators

- Web characterized by **one-way communication**

Web 1.0

- **Main characteristics**:
  - Content managed by the server's file system instead of a (relational) database management system, i.e., (R)DBMS
  - Static instead of dynamic pages: Generated on the server side using static interfaces instead of using Web applications written in a dynamic programming language
  - Menus created through the use of GIF images, pages structured in frames
  - The user can interact (through feedback) with the site through HTML forms sent by e-mail
  - Initially through the mailto protocol: a user fills out a form and, by clicking on the form submit button, her/his email client starts and tries to send an email containing the form details
    - The complications of the mailto protocol have led browser developers to embed email clients in their browsers

Web 2.0

- The term **Web 2.0** appears to have been first used by Darcy DiNucci in 1999[1] and popularized several years later by **Tim O'Reilly** and Dale Dougherty at the O'Reilly Media Web 2.0 Conference in late 2004[2]

- Web 2.0 does not refer to an update of any technical specification, but to changes in the way Web pages are designed and used

- The transition has been gradual and there is no precise date that can identify the transition between Web 1.0 and Web 2.0

[1] http://darcyd.com/fragmented_future.pdf
[2] http://www.paulgraham.com/web20.html

Web 2.0

- A Web 2.0 site can allow users to interact and collaborate with each other in **virtual communities**, generating and exchanging content, in contrast to the first generation of Web 1.0 sites where people were limited to passive viewing of the content

- Web 2.0 gives almost all users the same freedom to contribute

- The type of communication is **bidirectional**

Web 3.0

- The main evolutionary aspects of **Web 3.0** compared to Web 2.0 can be identified as follows:
  - Semantic Web and Artificial intelligence, associated with a more structured data storage
    - Possibility of developing search engines that allow querying through natural language and the retrieval of information according to approaches aimed at exploiting artificial intelligence to better identify the needs and tastes of users according to their behavior on the net → BING + GPT-4
  - Greater computing capacity and new algorithms aimed at the construction of truly usable 3D environments (evolution of what was the attempt of SecondLife)

Semantic Web

- Term coined by its creator, **Tim Berners-Lee**

- It is the transformation of the World Wide Web into an environment where documents (HTML pages, files, images, and so on) are associated with information and data (metadata) that specify their **semantic context** in a format suitable for querying and interpretation (e.g., through search engines) and, more generally, automatic processing

Semantic Web

- Based on languages that use **predicate logic**

- The information can be expressed with **assertions** consisting of triples formed by subject, predicate and value (subject, verb, object).

- Example:

	Assertion 1	**Assertion 2**
Subject	Marco Viviani	Marco Viviani
Predicate	lives in	teaches
Value	Milan	Social Media Analytics

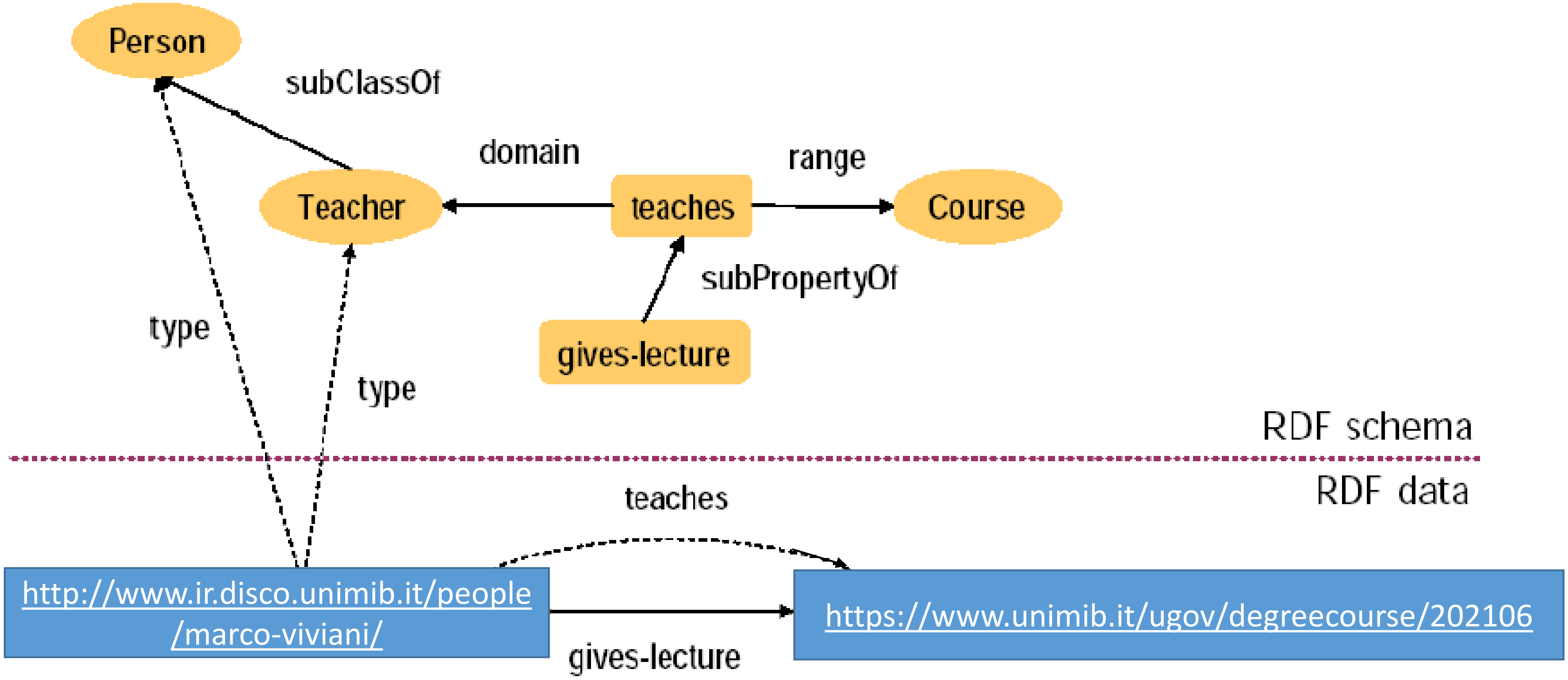
Resource Description Framework

- The **Resource Description Framework** (RDF) is the basic tool proposed by W3C for the encoding, exchange and reuse of structured metadata and allows semantic interoperability between applications that share information on the Web

- Through RDF it is possible to perform **reasoning**, that is to infer logical consequences from a set of axioms

https://www.w3.org/

Resource Description Framework



Web 4.0?

- Newly introduced definition, still "unstable"

- Web 4.0 **should be** characterized by:
  - Augmented reality: devices such as Google Glasses or smartwatches that allow you to interact in real time with the Web by superimposing the world around us with the network
  - Digital alter ego: documents that update and connect together, incorporating chips, with a technical infrastructure as support. As we populate the web with our personal contents, we will create a real virtual alter ego, which will allow us to make the two identities interact in real-time: the real one and the digital one
  - New interfaces: applied to electronic devices around us and the Internet. E.g., home automation;
  - Pervasive information: if the transition to an "enhanced" Web allows us to change society, by intervening on the information on the network, we will be able to change the reality that surrounds us → Metaverse

The Social Web

- The **Social Web** is made up of the set of social relationships that connect people through the World Wide Web
- The Social Web focuses on how Websites and software are designed and developed to support and foster social interaction
- Online social interactions form the basis of **many online activities**:
  - Online shopping (e-commerce)
  - Online education (e-learning)
  - Social networking
  - Websites with social components
  - …