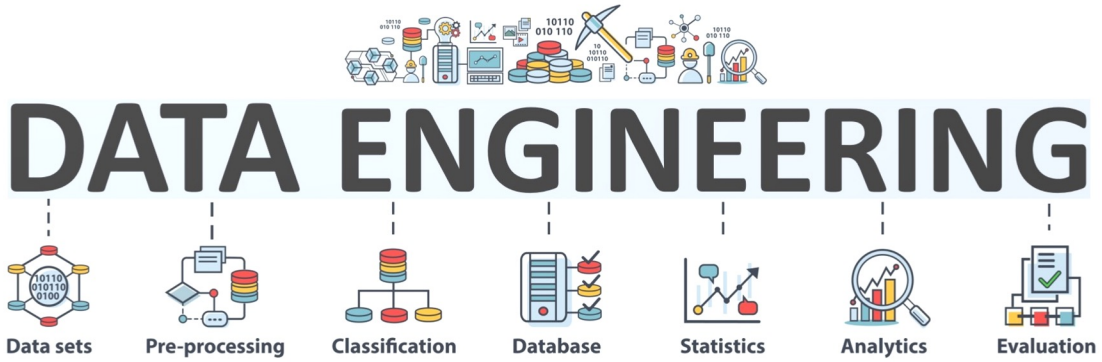


Lecture 2: Data Acquisition

CS5481 Data Engineering

Instructor: Linqi Song

Outline

- 1. Data sources
- 2. Web scraping
- 3. From web craping to web crawling

Sources of data

Relational databases

Flat files and XML datasets

APIs and web services

Relational databases

Typically, data stored in databases and data warehouses can be used as a source for analysis, organizations have internal applications to support them in managing:

- Day to day business activities
- Customer transactions
- Human resource activities
- Workflows

Using queries to extract data from relational databases

SQL, or **Structured Query Languages**, is a querying language used for extracting information form relational databases. Offers simple commands to specify:

- What is to be retrieved from the database.
- Table from which it needs to be extracted.
- Grouping records with matching values.
- Dictating the sequence in which the query results are displayed.
- Limiting the number of results that can be returned by the query.

Using queries to extract data from non-relational databases

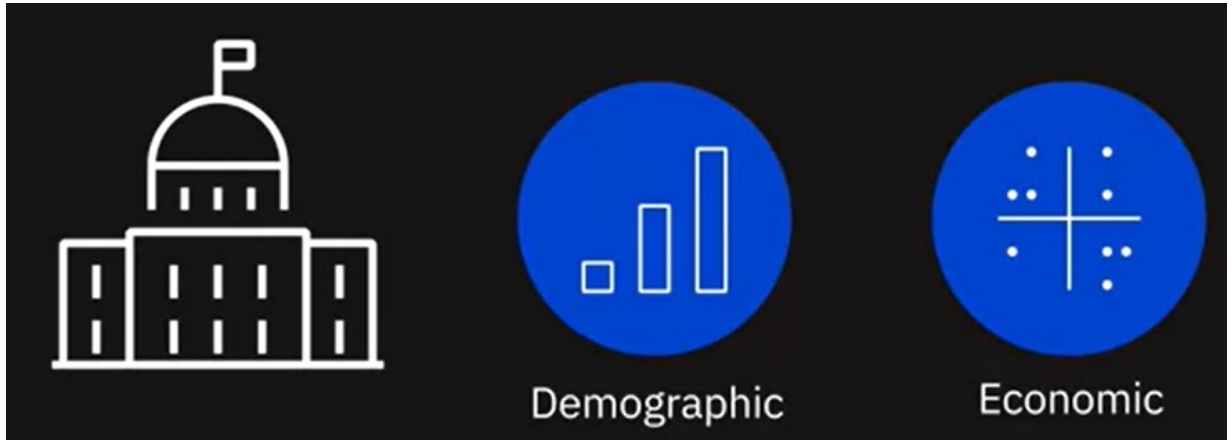
Non-relational databases can be queried using **SQL** or **SQL-like** query tools.

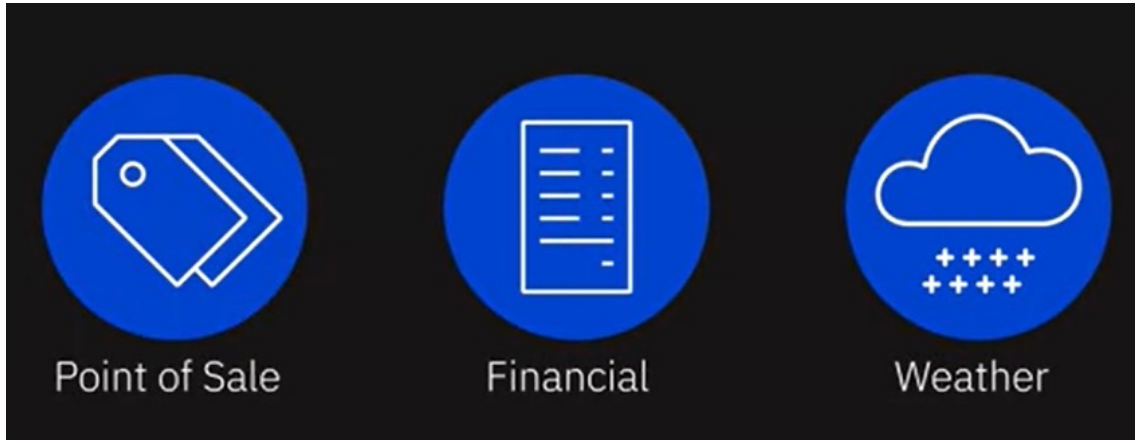
Some non-relational databases come with their own querying tools such as CQL for Cassandra and GraphQL for Neo4J.

Flat file and XML datasets

External to the organization, there are other publicly and privately available datasets.
Such data sets are typically made available as flat files, spreadsheet files, or XML documents.

```
<?xml version="1.0" encoding="UTF-8" ?>
- <customer_order number="004985" date="2004-06-24">
  - <lines>
    - <line no="1">
        <item>Disc CD</item>
        <quantity>30</quantity>
        <price>0.95</price>
      </line>
    - <line no="2">
        <item>Disc CD-RW</item>
        <quantity>20</quantity>
        <price>2.95</price>
      </line>
  </lines>
- <customer>
    <name>Technical University of Lublin</name>
    <street>Nadbystrzycka 38</street>
    <city>Lublin</city>
    <post_code>20-501</post_code>
  </customer>
- <payment>
    <card_issuer>Master Card</card_issuer>
    <card_number>1234 567890 12345</card_number>
    <expiration_date month="10" year="2005" />
  </payment>
</customer_order>
```





Flat files

- Store data in plain text format
- Each line, or row, is one record
- Each value is separated by a delimiter
- All of the data in a flat file maps to a single table
- Most common flat file format is .CSV

APIs and web services

APIs and Web Services typically listen for incoming requests, which can be in the form of **web requests** from users or **network requests** from applications, and return data in plain text, XML, HTML, JSON, or media files.

Web Requests

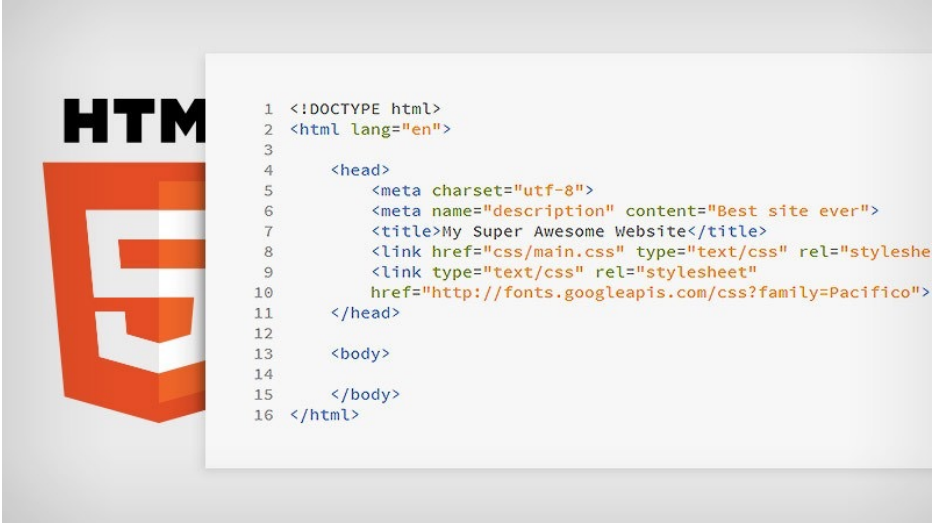
Network Requests

Sources for gathering data – web

Web is a source of publicly available data that is available to companies and individuals for free or commercial use.

- News websites, social networks
- Textbooks
- Government records
- Papers and articles for public consumption

Types of file formats (1)



- **Delimited text file formats, or .CSV**

  used to store data as text, each value is separated by a delimiter.

- **Microsoft Excel Open .XML Spreadsheet, or .XLSX**

- The **HyperText Markup Language (HTML)** is the standard markup language for documents designed to be displayed in a web browser**.**

- **Extensible Markup Language, or .XML**

  It is a markup language with set rules for encoding data.

```
<?xml version="1.0"?>
<quiz>
 <qanda seq="1">
  <question>
   Who was the forty-second
   president of the U.S.A.?
  </question>
  <answer>
   William Jefferson Clinton
  </answer>
 </qanda>
 <!-- Note: We need to add
  more questions later.-->
</quiz>
```

XML

Types of file formats (2)

- **Portable Document Format, or .PDF**

  Can be viewed the same way on any device.

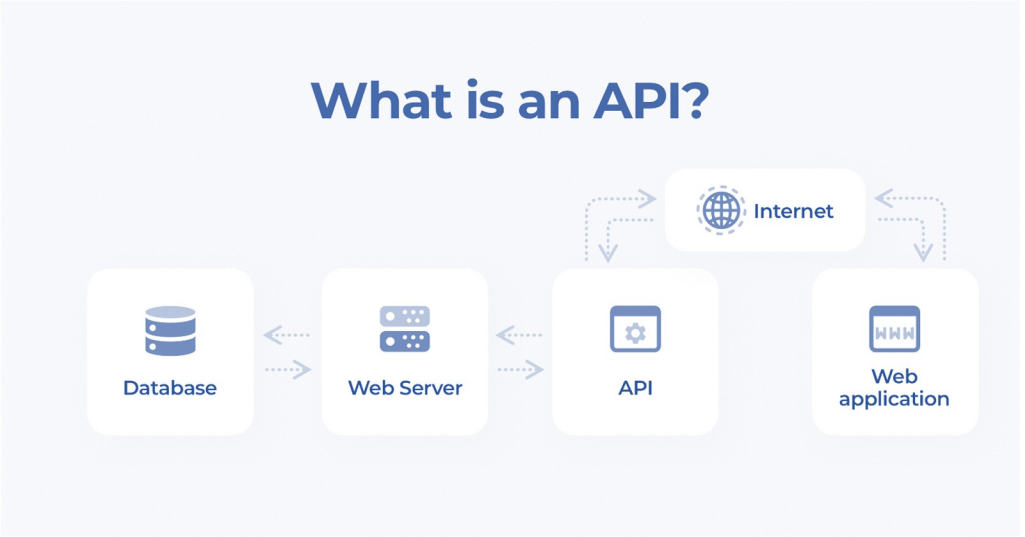
- **JavaScript Object Notation, or .JSON**

  A text-based open standard designed for transmitting structured data over the web.


```
{
    hey: "guy",
    anumber: 243,
  - anobject: {
        whoa: "nuts",
      - anarray: [
            1,
            2,
            "thr<h1>ee"
        ],
        more: "stuff"
    },
    awesome: true,
    bogus: false,
    meaning: null,
    japanese: "明日がある。",
    link: http://jsonview.com,
    notLink: "http://jsonview.com is great"
}
```

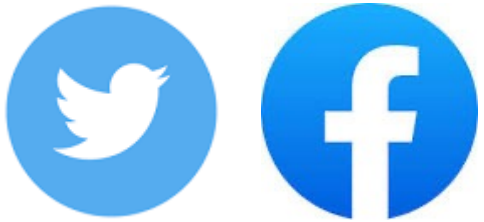
Application Programming Interfaces (or APIs)

- Popularly used for extracting data from a variety of data sources.
- Are invoked form applications that require the data and access an endpoint containing the data. Endpoints can include databases, web services, and data marketplaces.
- Also used for data validation.



API examples

Popular Examples of APIs



Twitter and Facebook APIs
For customer sentiment analysis

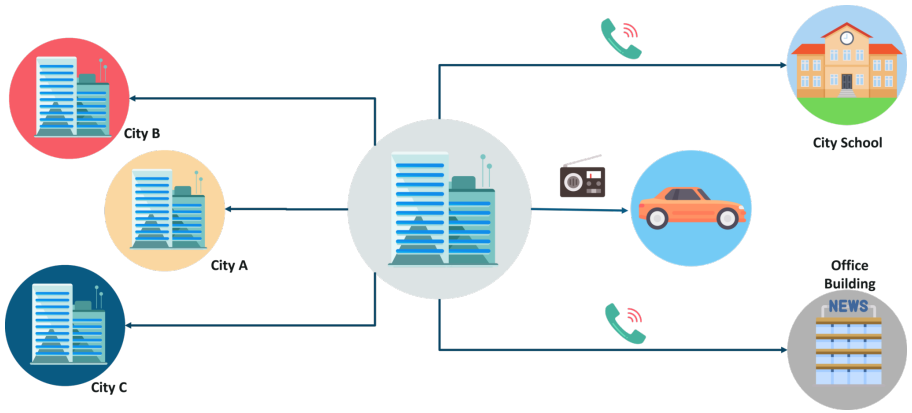
Stock market APIs
For trading and analysis

Data lookup and validation APIs
For cleaning and co-relating data

Data streams and feeds

Aggregating streams of data flowing from:

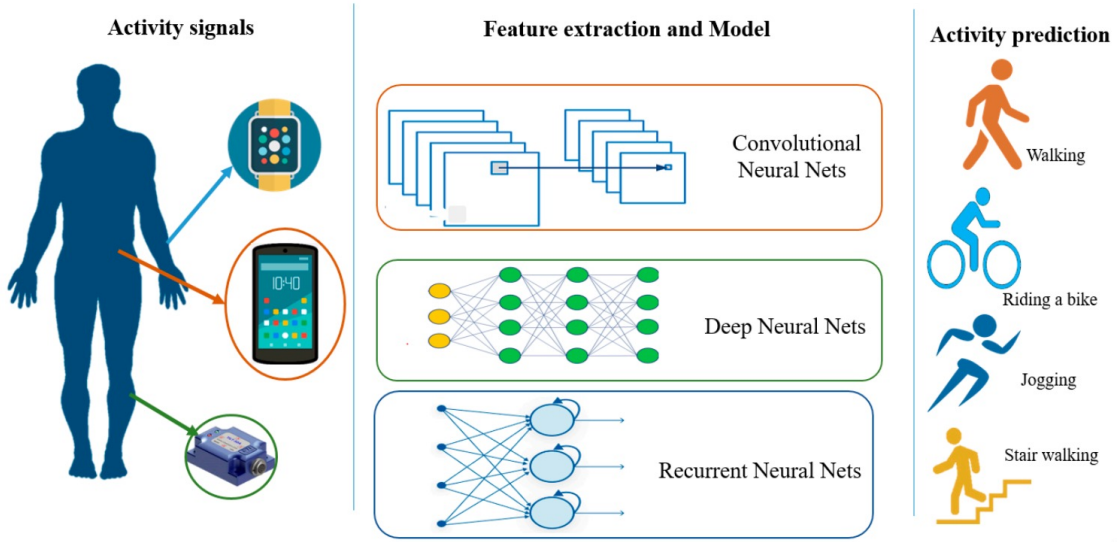
1. Instruments
2. IoT devices and applications
3. GPS data from cars
4. Computer programs
5. Websites
6. Social network.

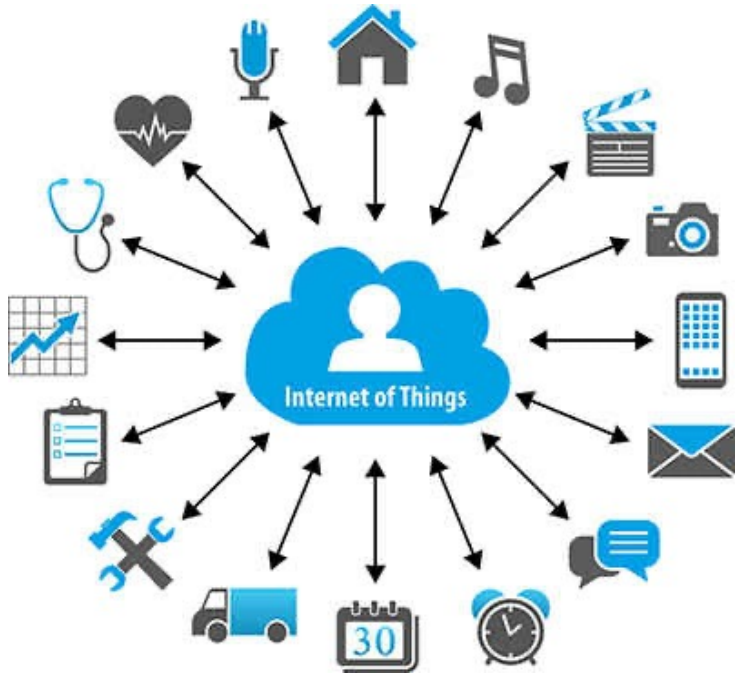


On the World Wide Web, a web feed is a data format used for providing users with frequently updated content.

Sources for gathering data – sensor data

Sensor data produced by wearable devices, smart buildings, smart cities, smartphones, medical devices, even household appliances, is a widely used source of data.





Data streams and feeds examples

Examples of uses:

- Surveillance and video feeds for **threat detection**.
- **Sensor data** feeds for monitoring industrial or framing machinery.
- **Social media** feeds for sentiment analysis.
- Stock and market tickers for **financial trading**.
- **Retail transaction** streams for predicting demand and supply chain management.
- **Web click** feeds for monitoring web performance and improving design.

Data streams and feeds tools

Popular technologies used to process data streams include:

- Kafka

Apache Kafka is a distributed event store and stream-processing platform. It is an open-source system developed by the Apache Software Foundation written in Java and Scala. The project aims to provide a unified, high-throughput, low-latency platform for handling real-time data feeds

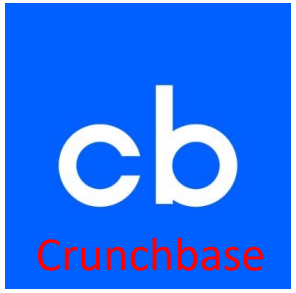
- Storm

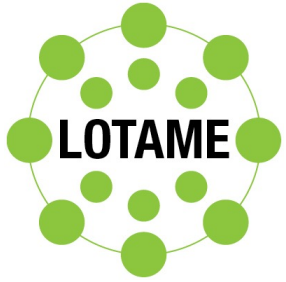
Apache Storm is a distributed stream processing computation framework written predominantly in the Clojure programming language.

Sources for gathering data – data exchange

Data exchange is a source of third-party data that involves the voluntary sharing of data between data providers and data consumers. Individuals, organizations, and governments could be both data providers and data consumers.

AWS DataExchange





Lotame

Snowflake

- Data from business applications
- Sensor devices
- Social media activity
- Location data
- Consumer behavior data

Other sources for gathering data

- **Surveys** – gather information through questionnaires distributed to a select group of people.
- **Census** – popularly used for gathering household data such as wealth and income of population.
- **Interviews** – a source for gathering qualitative data such as the participant's opinions and experiences. Interviews can be telephonic, over the web, or face-to-face.
- **Observation studies** – include monitoring participants in a specific environment or while performing a particular task.

Outline

- 1. Data sources
- 2. Web scraping
- 3. From web craping to web crawling

Web scraping

- The construction of an agent to download, parse, and organize data from the web in an automated manner
- Extract relevant data from unstructured sources on the Internet
- Also known as screen scraping, web harvesting, and web data extraction
- Can extract text, contact information, images, videos, product items, etc.

Web scraping examples

Popular examples of uses:

- Generating sales leads through public data sources; weather information to forecast, for example, soft drink sales.
- Collecting training and testing datasets for machine learning models.
- There might be an interesting table on a Wikipedia page (or pages) you want to retrieve to perform some statistical analysis.
- You might wish to get a listing of properties on a real-estate site to build an appealing geo-visualization.

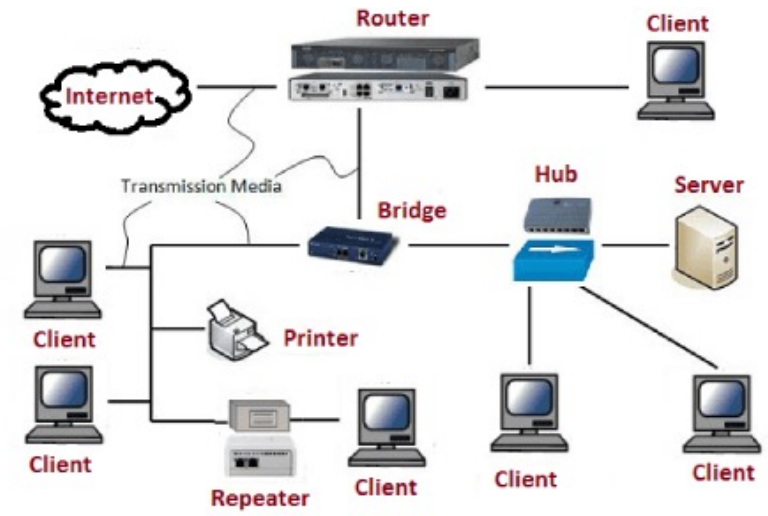
Web scraping tools

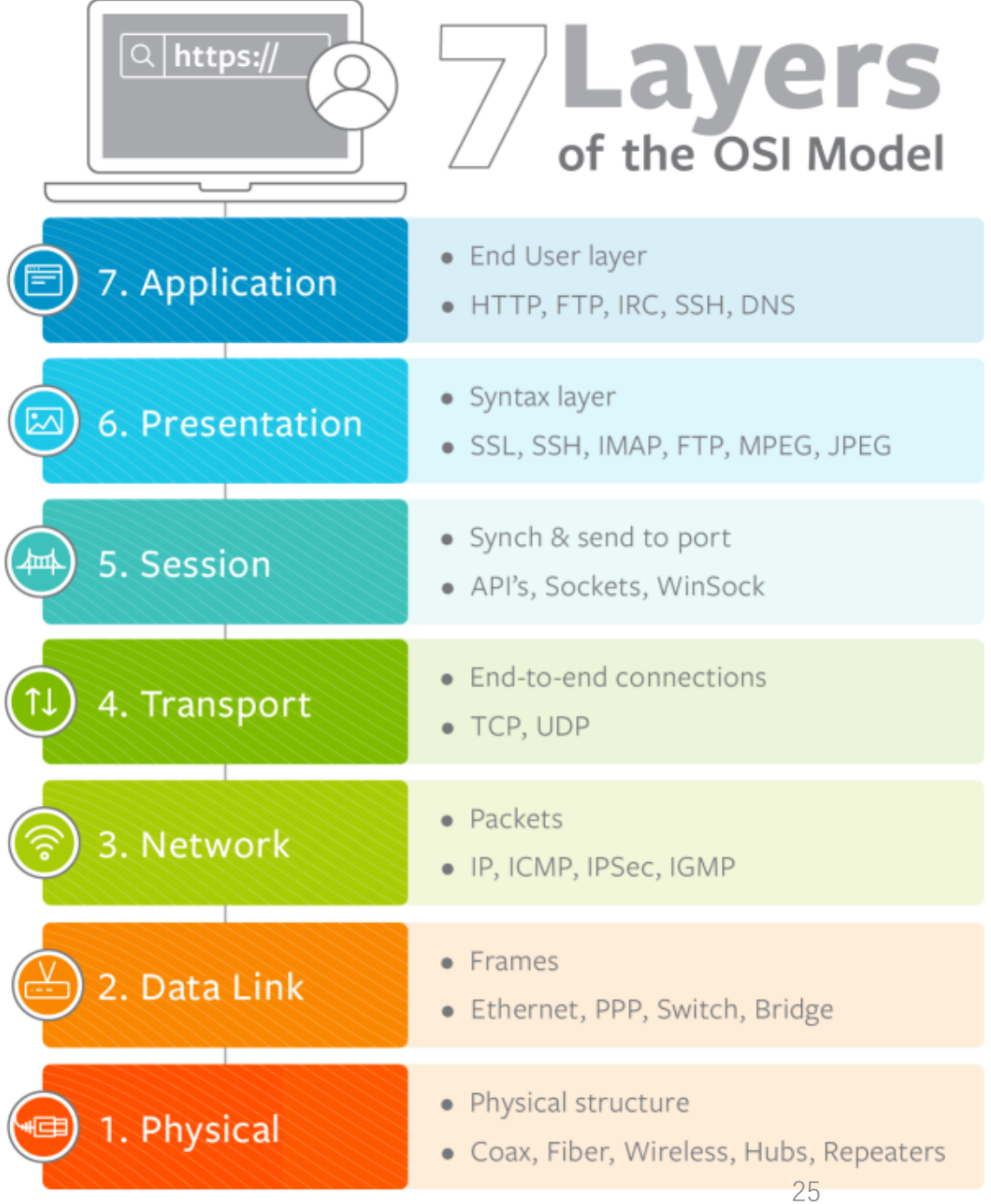
Popular Web Scraping tools:

- BeautifulSoup
- Scrapy
- Pandas
- Selenium

The web speaks HTTP

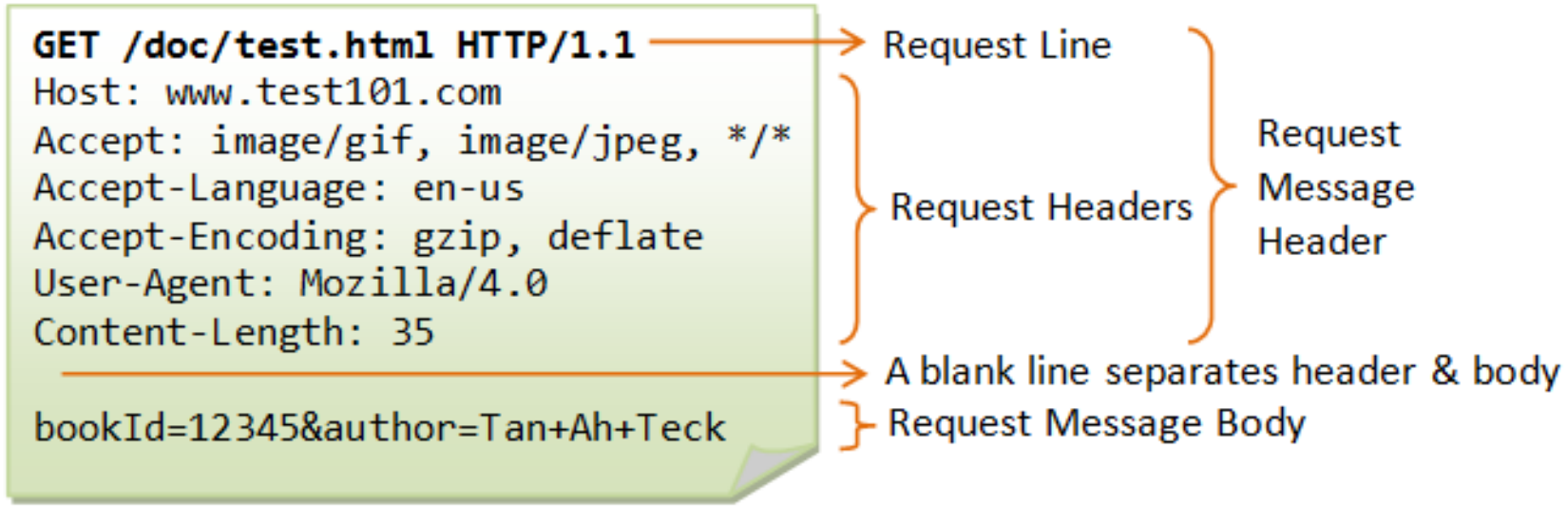
- 7 network layers of Open Systems Interconnection (OSI) model
- Web scraping mainly focus on the application layer: HTTP

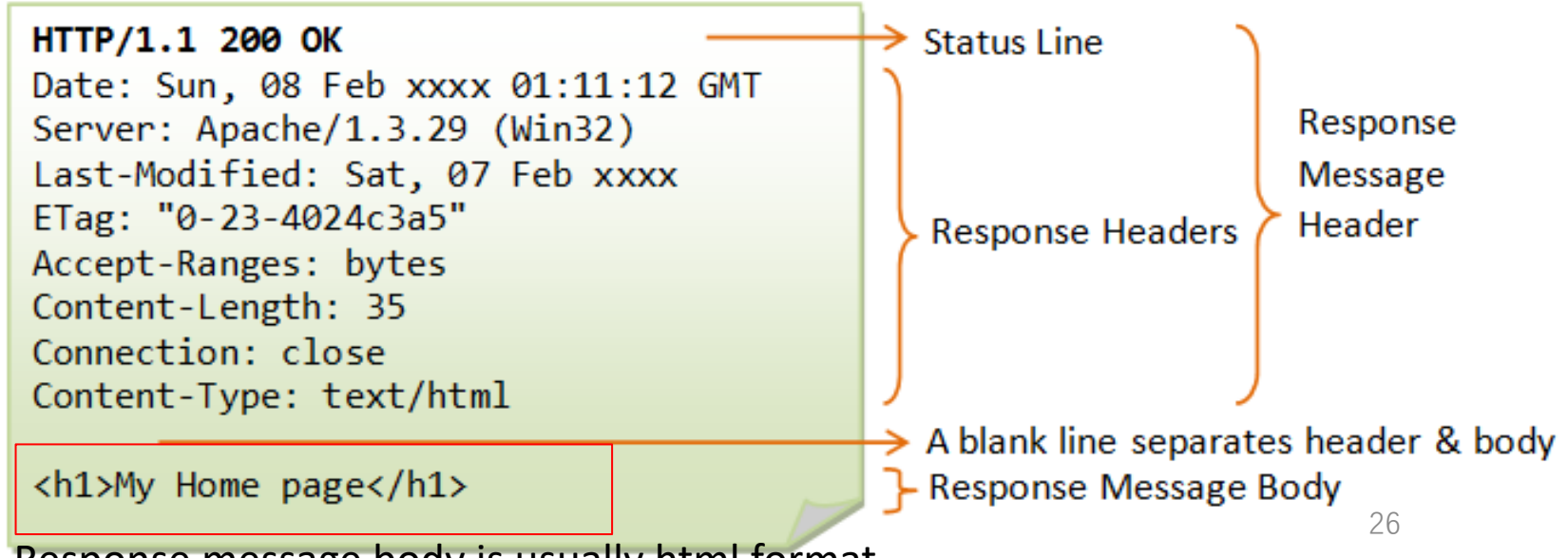




HTTP

The core component in the exchange of messages on WWW consists of a HyperText Transfer Protocol (HTTP) request message to a web server, followed by an HTTP response, which can be rendered by the browser.





Response message body is usually html format

Hypertext Markup Language (HTML)

- HTML is a standard markup language for creating web pages.
- HTML provides the building blocks to provide structure and formatting to documents.
- Python ‘requests’ library could get the html content from a webpage.

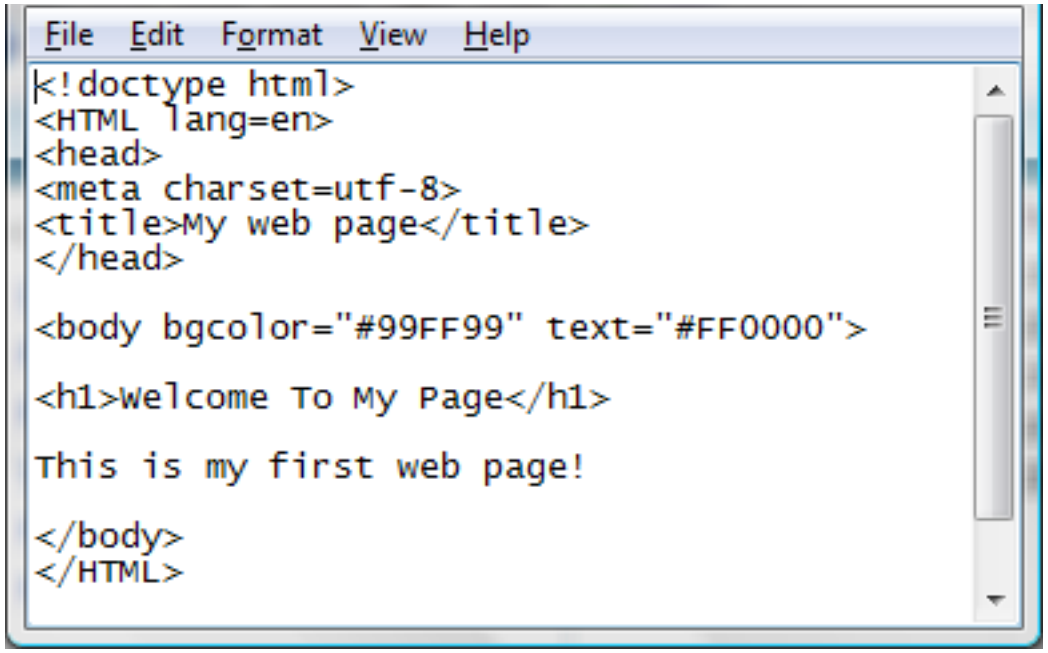
```
import requests

url = 'https://en.wikipedia.org/w/index.php' + \
      '?title=List_of_Game_of_Thrones_episodes&oldid=802553687'

r = requests.get(url)
print(r.text)
```

HTML format

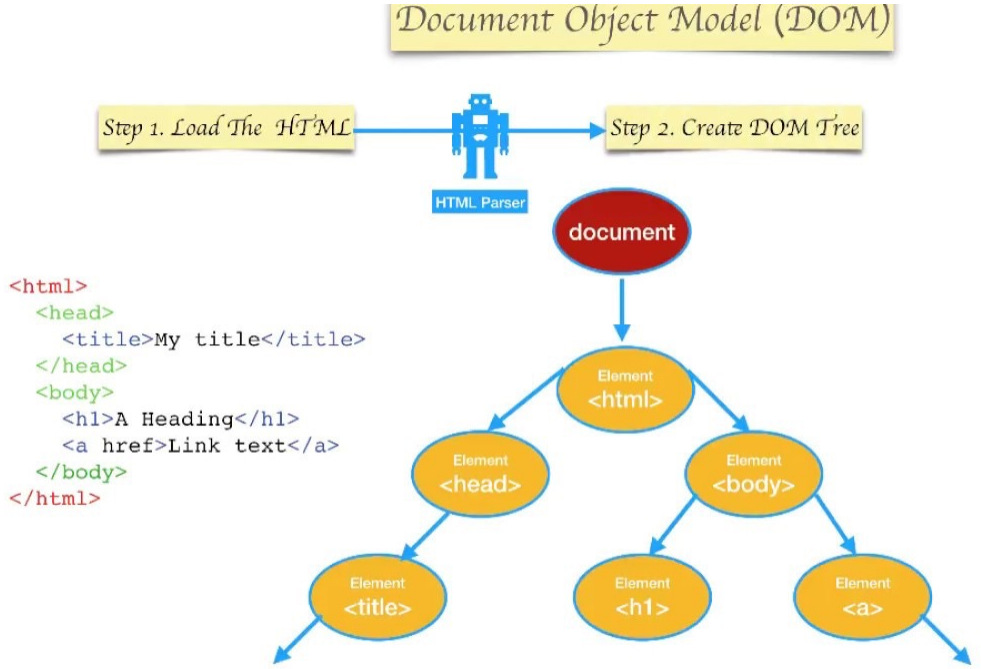
- HTML's building blocks are usually a series of tags that often come in pairs (but not always).
- Commonly used tags
  - ... to enclose a paragraph;
  -
 to set a line break;
  - <table>...</table> to start a table block, inside; <tr>...<tr/> is used for the rows; and <td>...</td> cells;
  - <img> for images;
  - <h1>...</h1> to <h6>...</h6> for headers;
  - ... to indicate a "division" in an HTML document, basically used to group a set of elements;
  - <a>... for hyperlinks;
  - ..., <ol>... for unordered and ordered lists respectively; inside of these, <li>... is used for each list item.



HTML parsing

Beautiful Soup

- HTML parsing involves tokenization and tree construction. HTML tokens include start and end tags, as well as attribute names and values. If the document is well-formed, parsing it is straightforward and faster. The parser parses tokenized input into the document, building up the document tree.



General web scraping procedure

- Identifying data for scraping
- Scraping the data
- Importing the data

Identifying data for scraping (1)

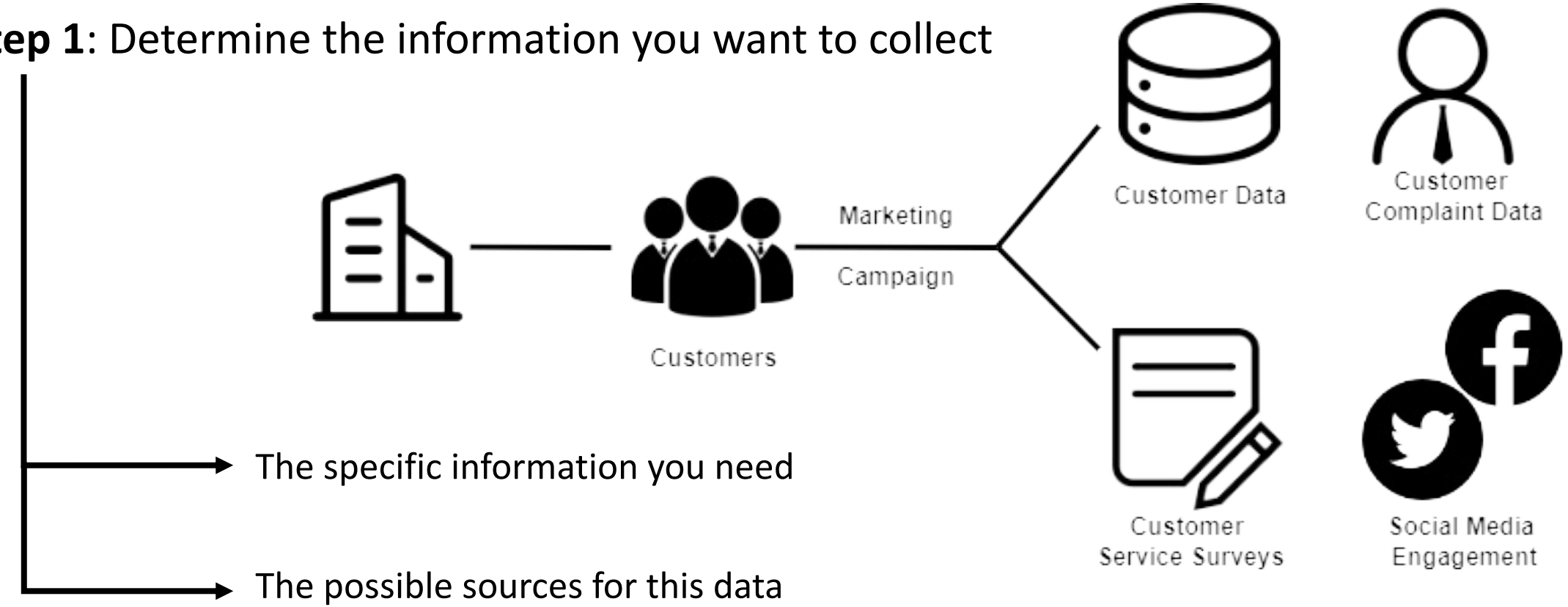
Importance of Identifying data for analysis:

- Identifying the right data is very important step of the data analysis process.
- Done right, it will ensure that your are able to look at a problem from multiple perspectives and your findings are credible and reliable.

Identifying data for scraping (2)

Process for identifying data

Step 1: Determine the information you want to collect



Identifying data for scraping (3)

Process for identifying data

Step 2: Define a plan for collecting data

Establish a timeframe for collecting data

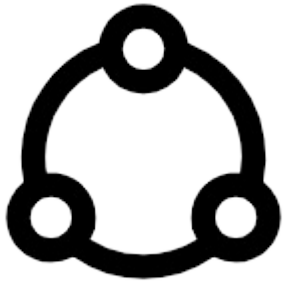
How much data is sufficient for a credible analysis

Define dependencies, risks, and mitigation plan

Identifying data for scraping (4)

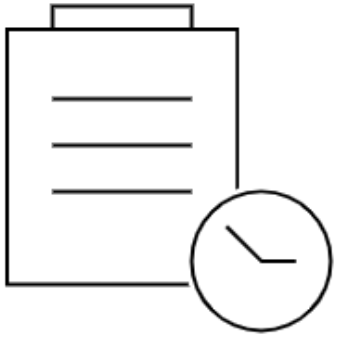
Process for identifying data

Step 3: Determine your data collection methods. The methods depend on:



Sources of Data

Type of Data

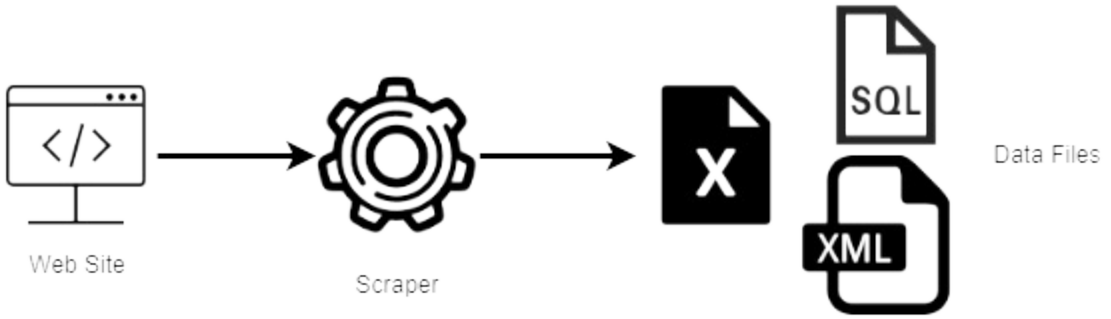


Timeframe over which you need the data

Volume of data

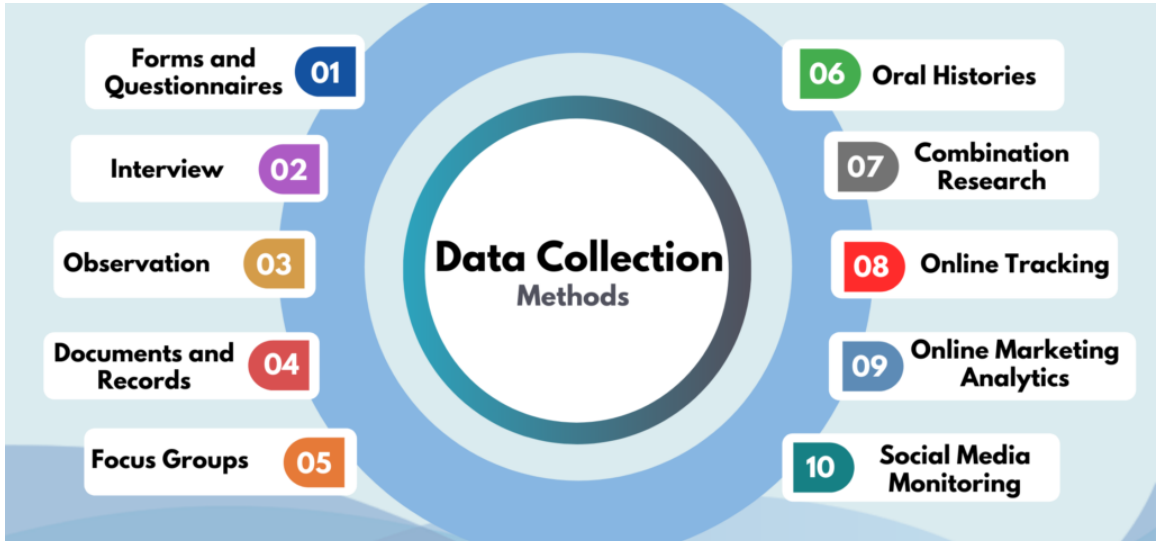
Web scraping

- **Web Scraping:** Extracting a large amount of specific data from online sources



Importing data into data repositories

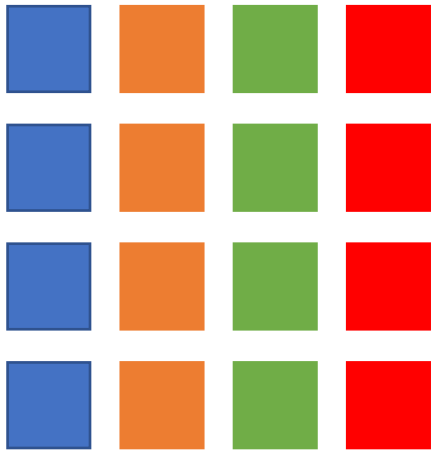
- **Gathering** data from data sources such as databases, the web, sensor data, data exchanges, and several other sources leveraged for specific data needs.
- **Importing** data into different types of data repositories.



Importing structured data

Importing data: data Identified and gathered -> data **repository**

Specific data **repositories** are optimized for certain types of data.

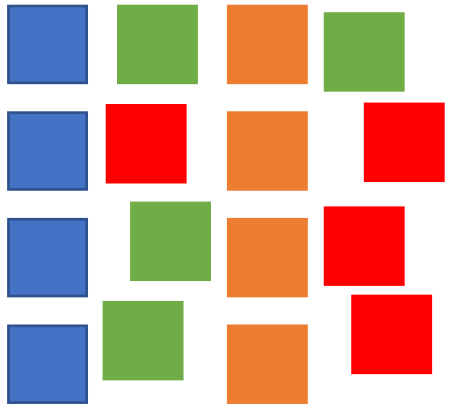


Structured data

- Relational databases store structured data with a well-defined schema
- Sources include data from OLTP systems, spreadsheets, online forms, sensors, network and web logs.
- Can be stored in NoSQL database.

Importing unstructured data

Specific data **repositories** are optimized for certain types of data.



Semi-structured Data

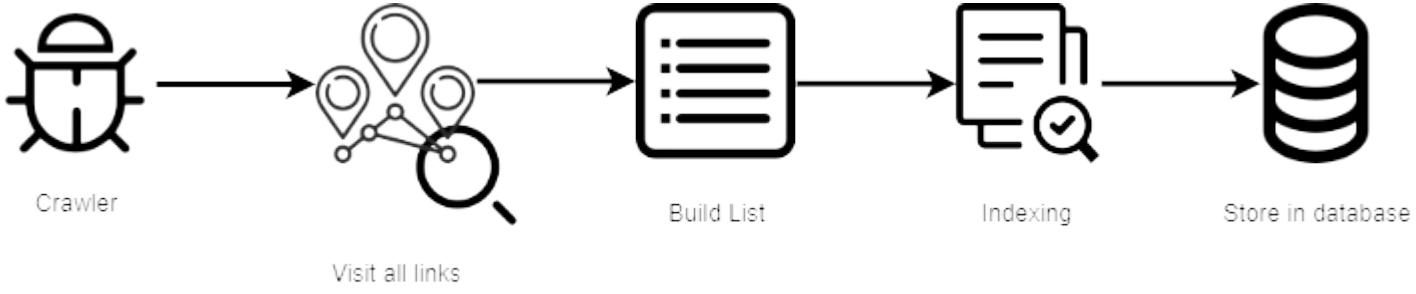
- Sources include emails, XML, zipped files, binary executables, and TCP/IP protocols.
- Can also be stored in NoSQL clusters.
- XML and JSON are commonly used for storing and exchanging semi-structured data.

Outline

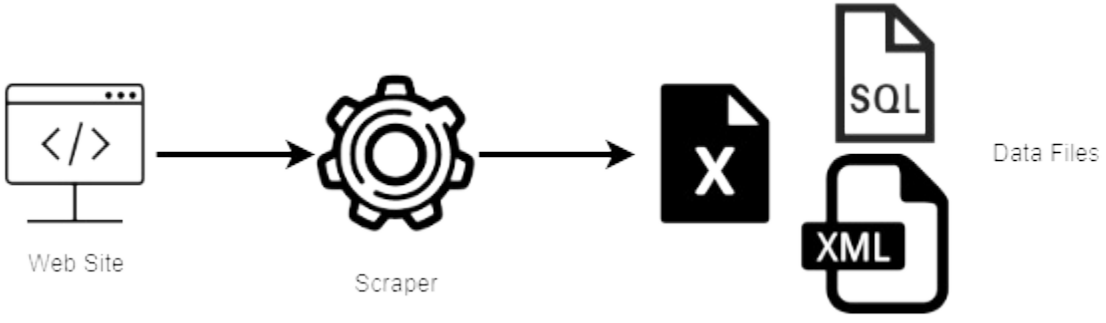
- 1. Data sources
- 2. Web scraping
- 3. From web craping to web crawling

From web scraping to web crawling

Web Crawling: Using tools to read, copy and store the content of the websites for archiving or indexing purposes. Crawling usually deals with a network of webpages



Web Scraping: Extracting a large amount of specific data usually from a single webpage or a single website



Different use cases

Web Crawling

- Generating search engine results.
- Monitoring SEO analytics.
- Performing website analysis.
- Performed only by large corporations.

Web Scrapers

- Comparing prices.
- Stock market analysis.
- Managing brand reputation.
- Academic and scientific research.
- Used by small and large businesses

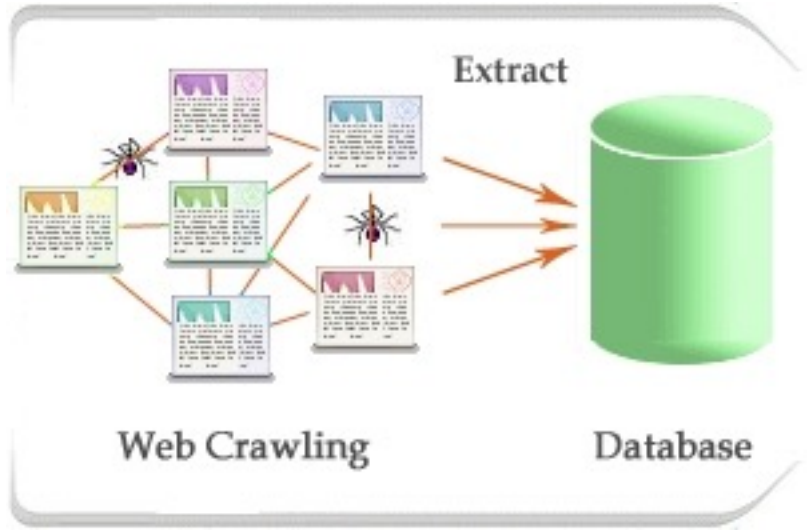
Tools

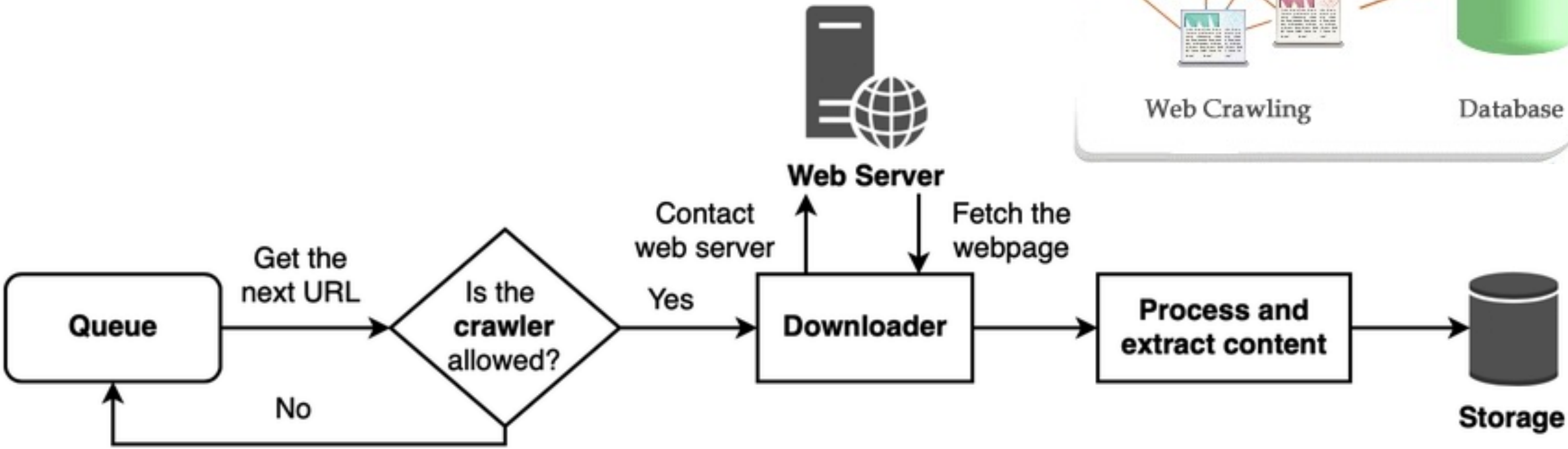
Differences between Web Crawling and **Web Scraping**

Web Crawlers

Web Scrapers

Web crawling process





A **web crawler** operates like a **graph** traversal/search algorithm.

What any crawler *MUST* do ?

- **Be robust**: Be immune to spider traps and other malicious behavior from web servers
- **Be polite**: Respect implicit and explicit politeness considerations.
  - **Explicit politeness:** specifications from webmaster on what portions of a site can be crawled – robots.txt
  - **Implicit politeness**: even with no specification, avoid hitting any site too often.

Robots.txt

- Protocol for giving spiders (“robots”) limited access to a website, originally from 1994.
    - www.robotstxt.org/robotstxt.html
- Website announces its request on what can(not) be crawled.
    - For a server, create a file /robots.txt.
    - This file specifies access restrictions.

Thanks for your attention!

Appendix

1.https://www.coursera.org/learn/introduction-to-data-analytics/home/week/3

2.https://www.ics.uci.edu/~lopes/teaching/cs221W15/slides/WebCrawling.pdf

3. https://link.springer.com/content/pdf/10.1007/978-1-4842-3582-9.pdf