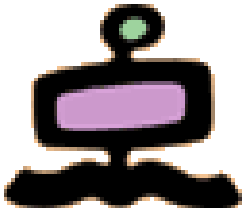
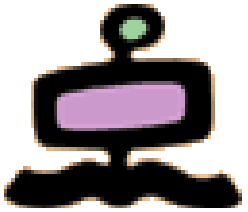
Corso di Biblioteche Digitali



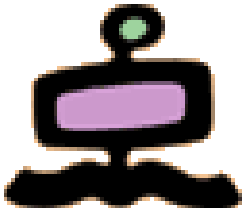
- Vittore Casarosa
  - Mail: casarosa@isti.cnr.it
  - Tel: 050 621 3115 (office) 348 397 2168 (mobile)
  - Skype: vittore1201
- “Ricevimento” at the end of the lessons or by appointment
- Final assessment
  - 70% oral examination
  - 30% project (development of a small digital library))
- Reference material:
  - Ian Witten, David Bainbridge, David Nichols, How to build a Digital Library, Morgan Kaufmann, 2010, ISBN 978-0-12-374857-7 (Second edition)
  - Material provided by the teacher
- **http://cloudone.isti.cnr.it/casarosa/BDG/**

Modules



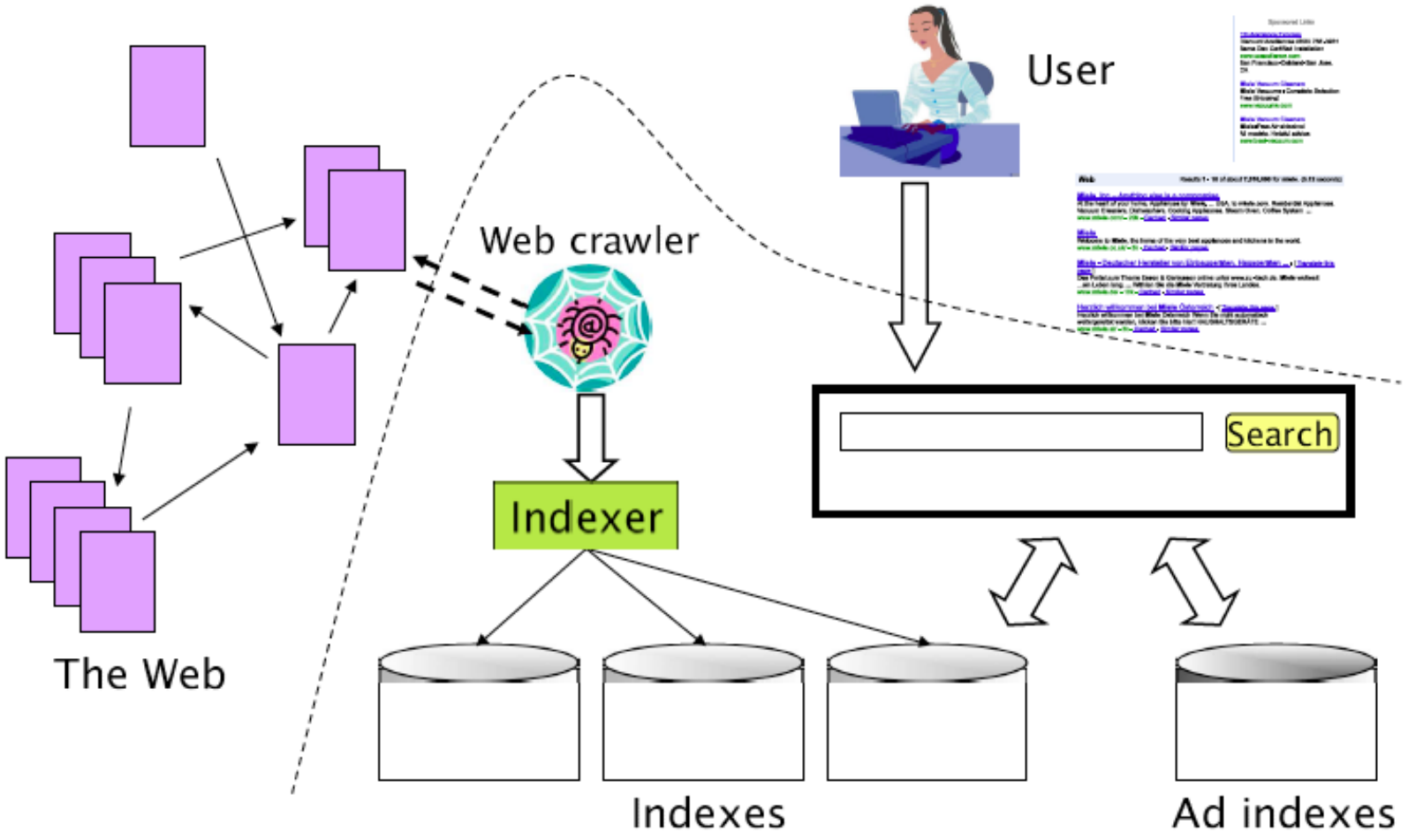
- Computer Fundamentals and Networking
- A conceptual model for Digital Libraries
- Bibliographic records and metadata
- Information Retrieval and Search Engines
- Knowledge representation
- Digital Libraries and the Web
- Hands-on laboratory: the Greenstone system

Information Retrieval

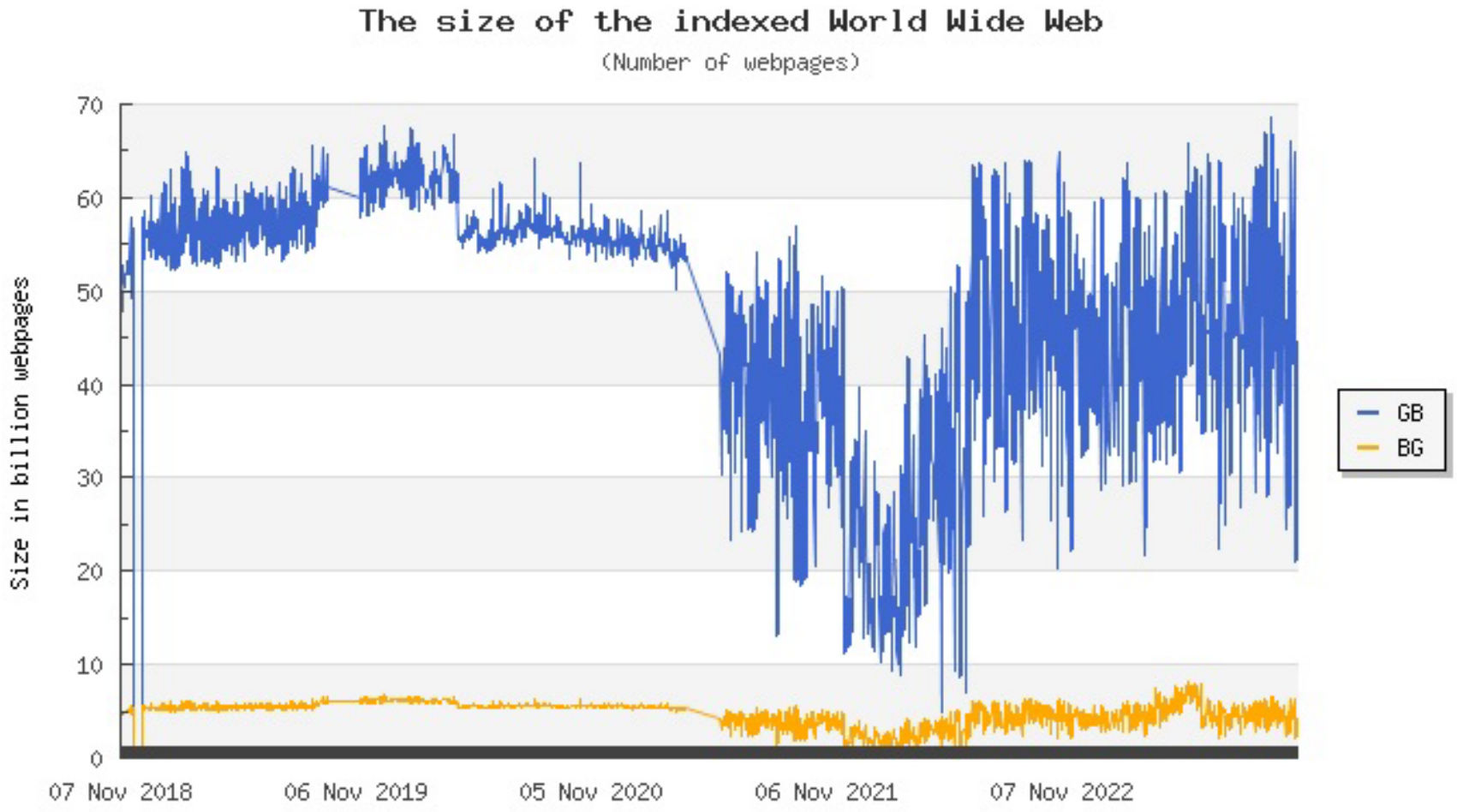


- Information Retrieval and Search Engines
  - Indexing a collection of documents
  - Ranking query results
  - Search engines in the Web
  - Ranking in Web search engines

Architecture of a Search Engine

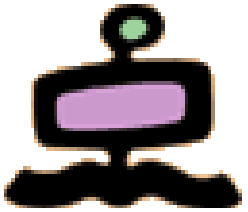


The size of the indexed Web



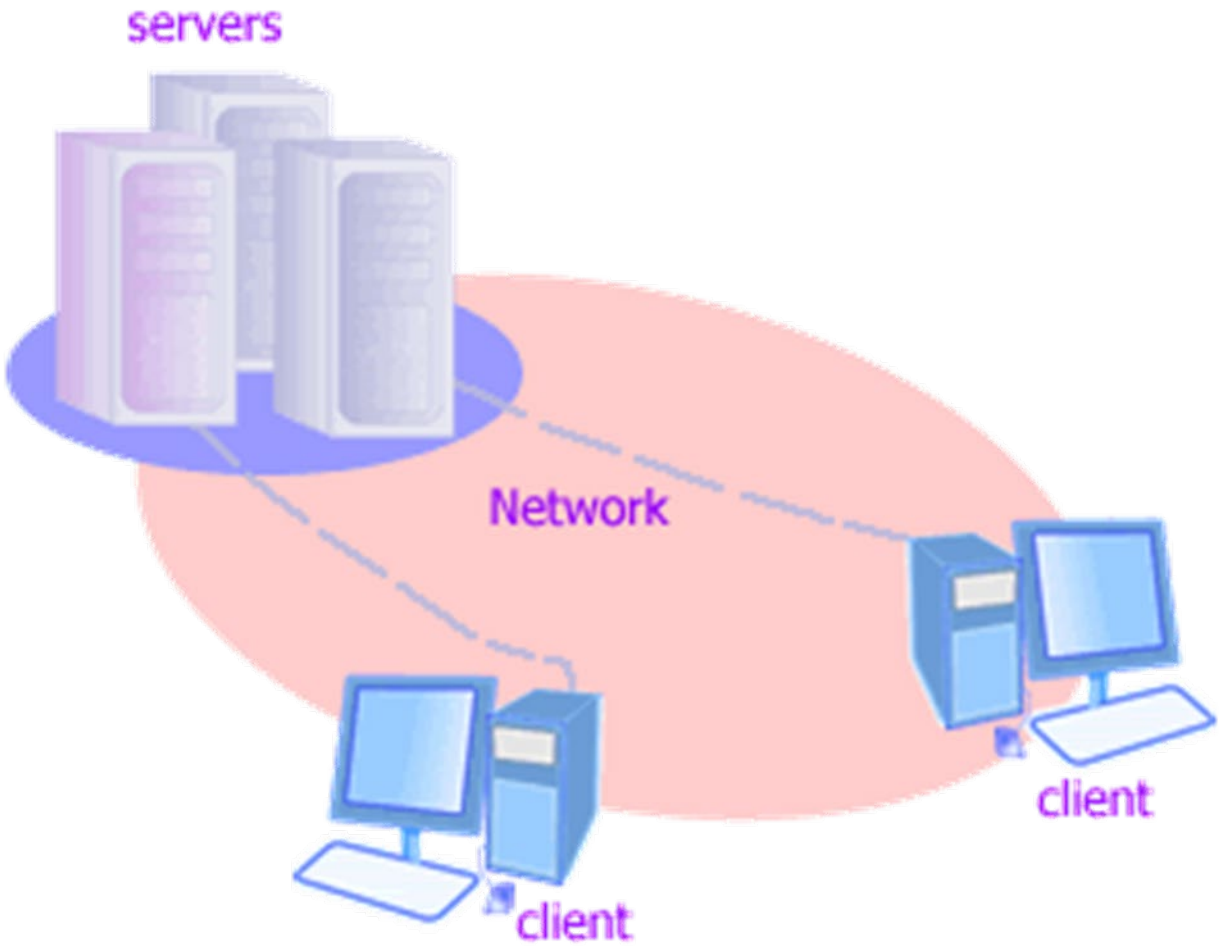
https://www.worldwidewebsize.com/

The Depth of the Web

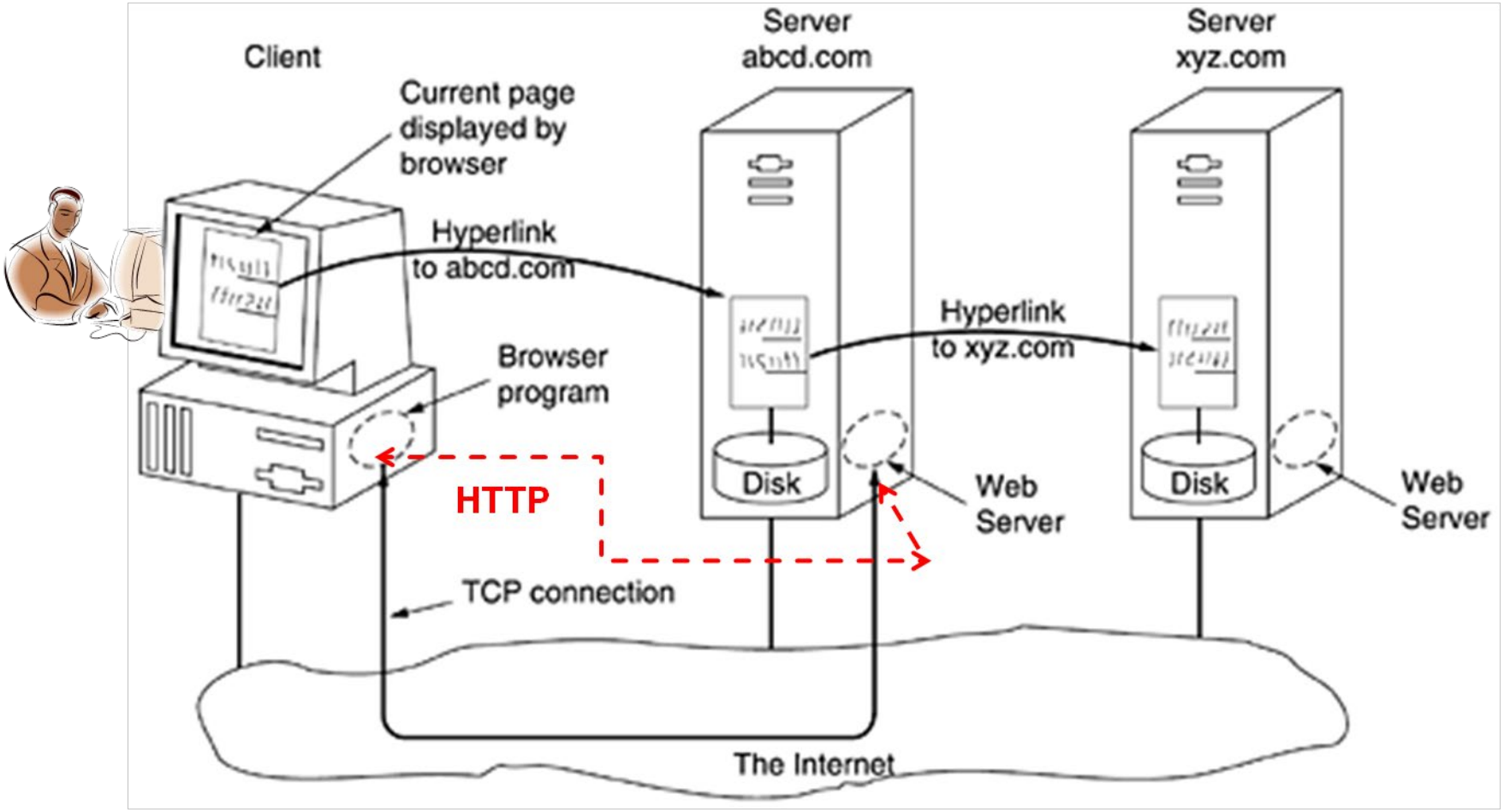


- A URL gives access to a web page.
- That page may have links to other pages (static pages). This is the surface web.
- Some pages (dynamic pages) are generated only when some information is provided to the web server.
- These pages cannot be discovered just by crawling. This is the deep web.
- The surface web is huge.
- The deep web is “unfathomable”.

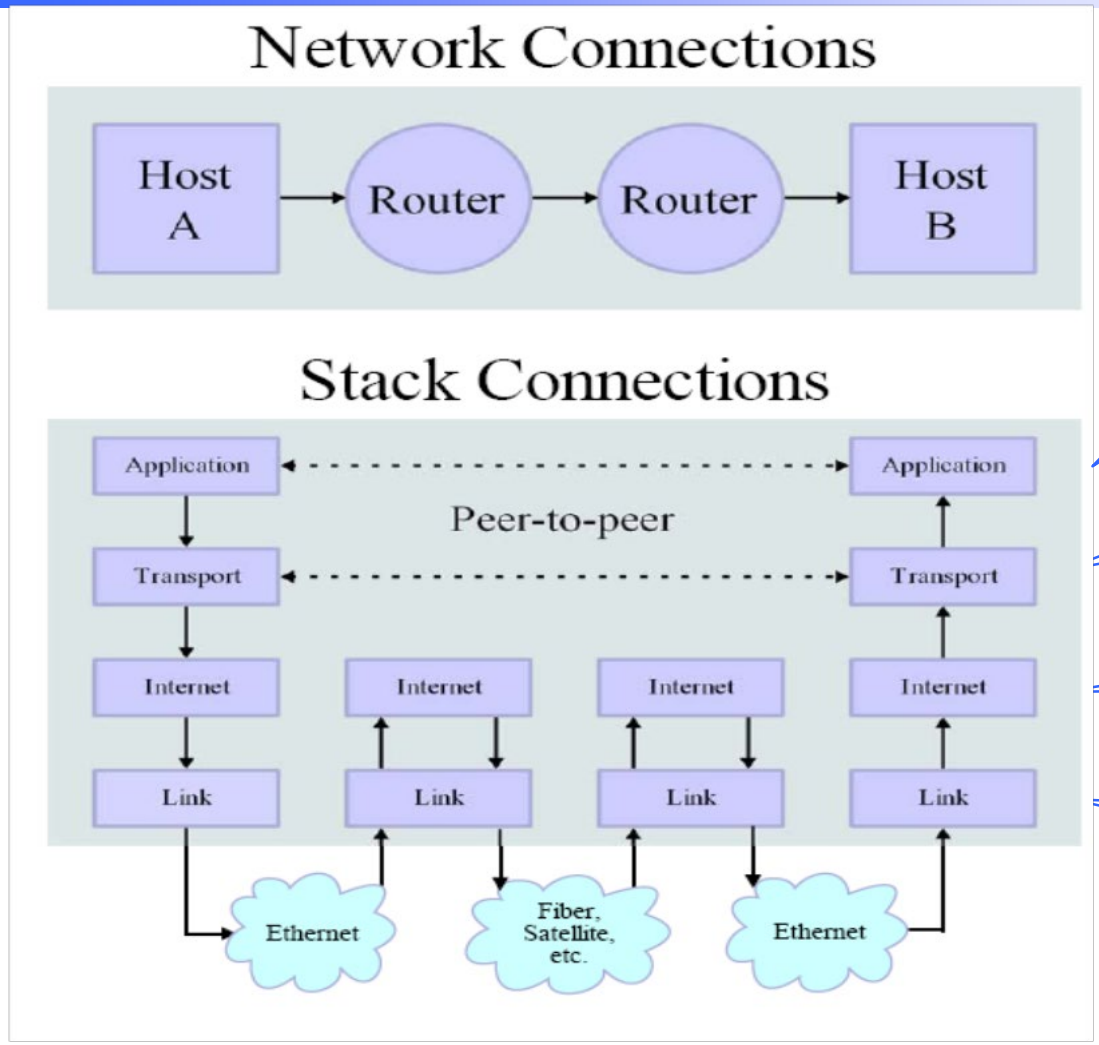
Client-server networks



The Web architecture



Internet protocols



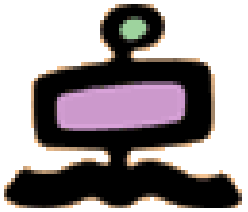
Application protocols e.g. HTTP

TCP/UDP

IP

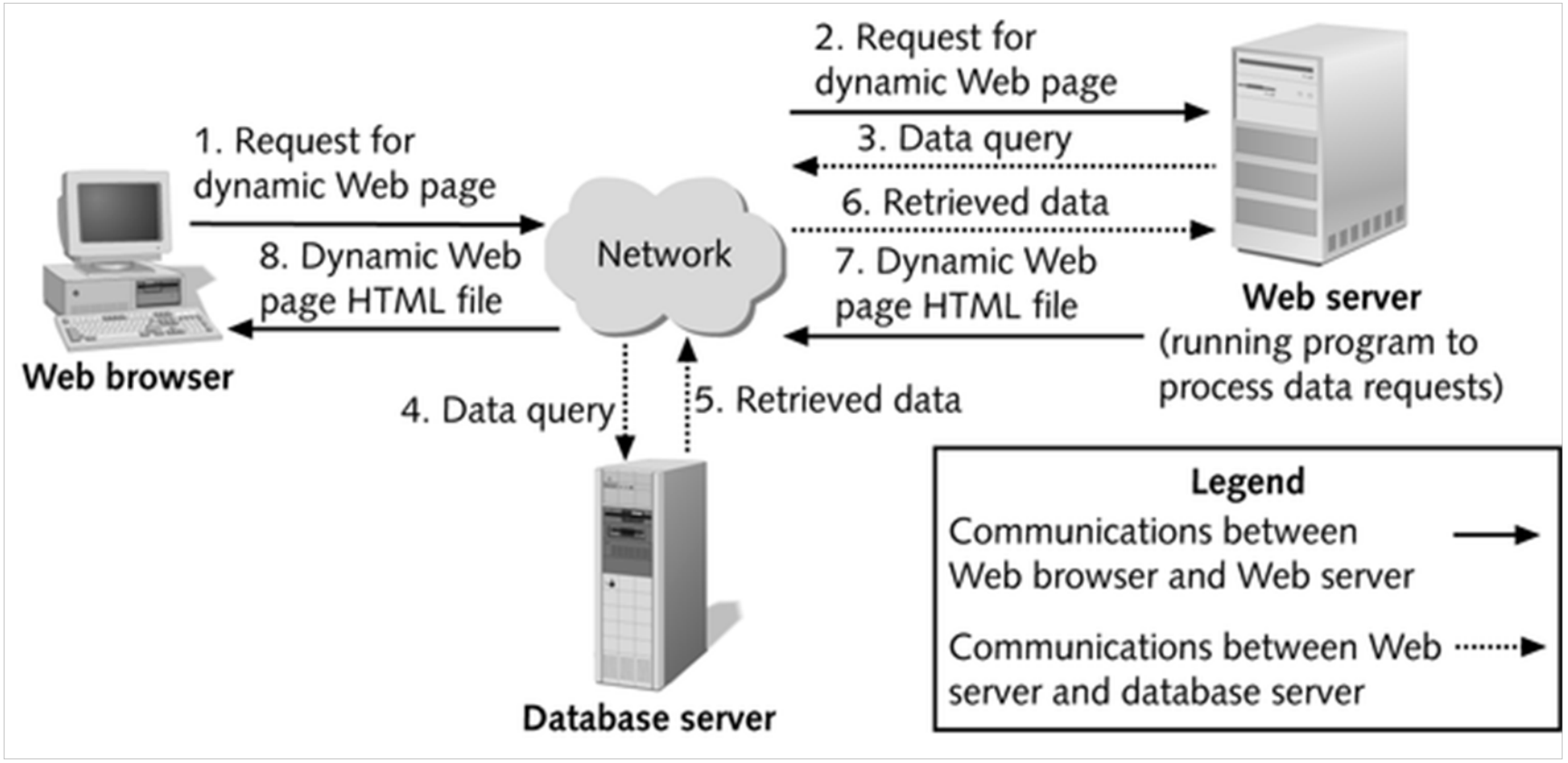
ETHERNET

The Depth of the Web

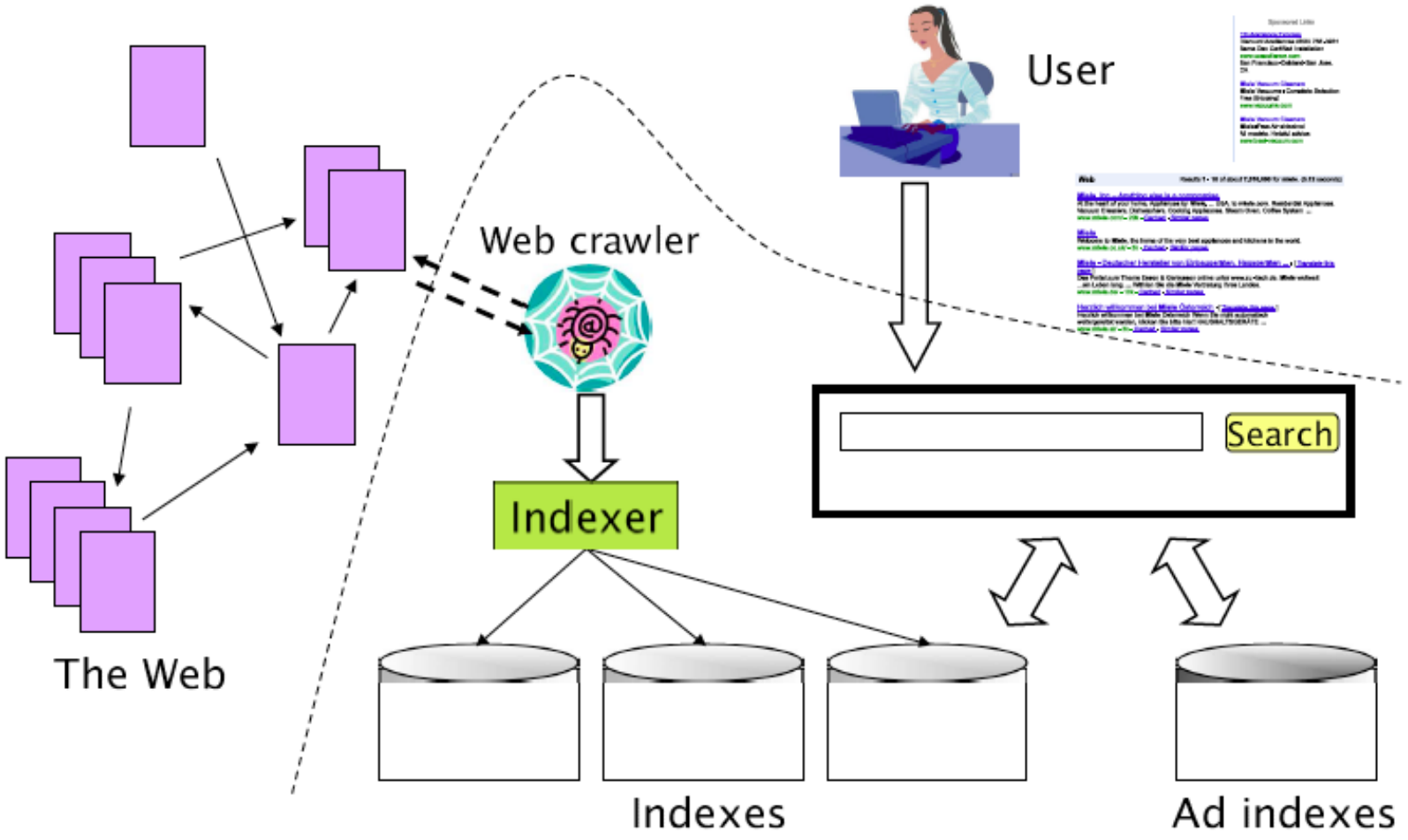


- A URL gives access to a web page.
- That page may have links to other pages (static pages). This is the surface web.
- Some pages (dynamic pages) are generated only when some information is provided to the web server.
- These pages cannot be discovered just by crawling. This is the deep web.
- The surface web is huge.
- The deep web is “unfathomable”.

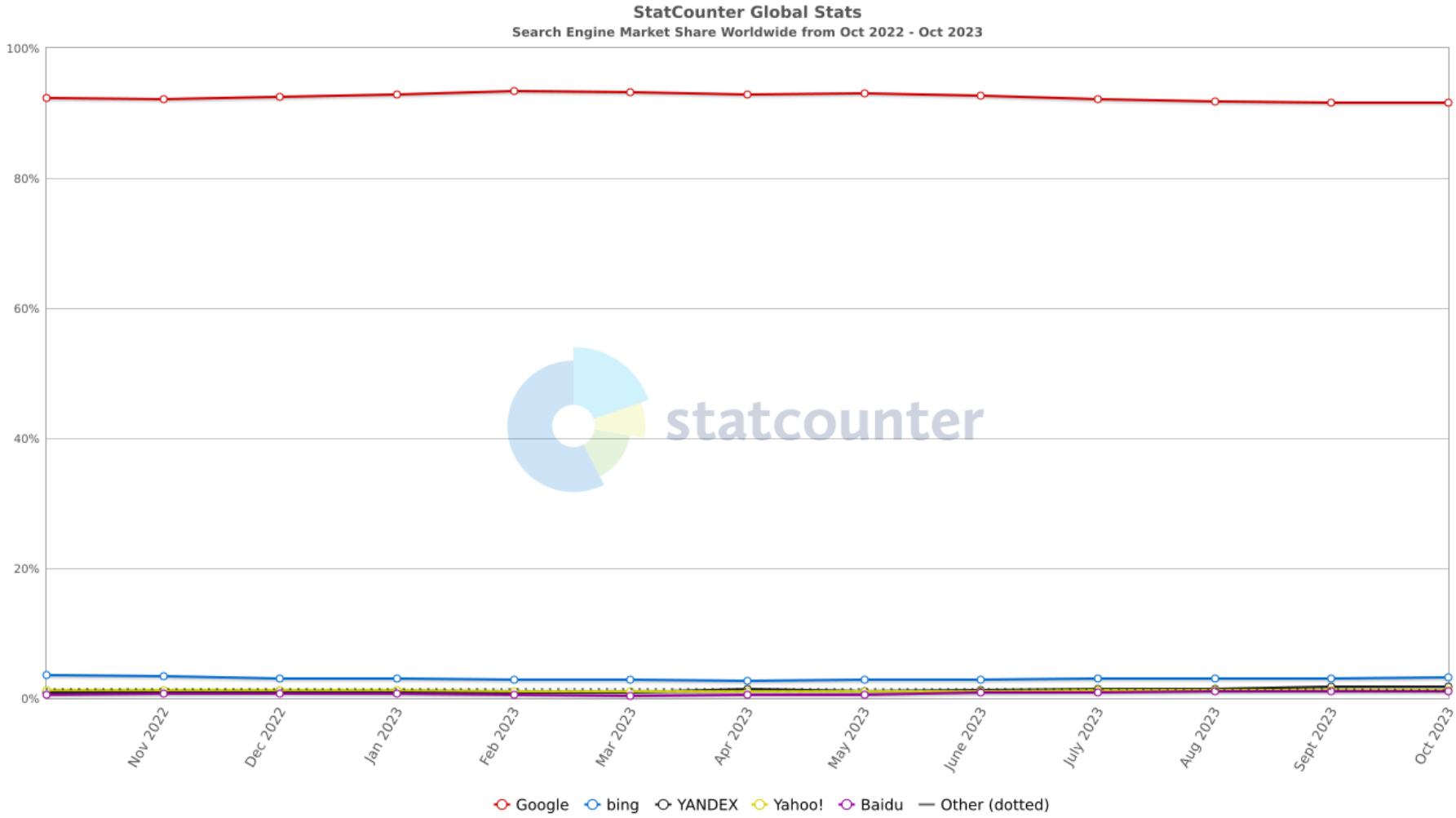
Dynamic web pages (data base driven)



Architecture of a Search Engine

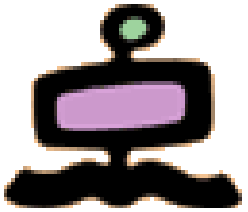


Worldwide queries to search engines (2023)



https://gs.statcounter.com/search-engine-market-share

Google searches



GENERAL GOOGLE SEARCH STATISTICS

Google is the most visited website

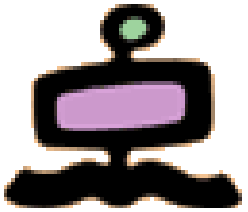
92% of all search volume globally is from Google

There are at least **8.5 billion** Google searches per day

The Google Index has up to **40 to 60 billion** active web pages

https://fitsmallbusiness.com/google-search-statistics/

Google searches



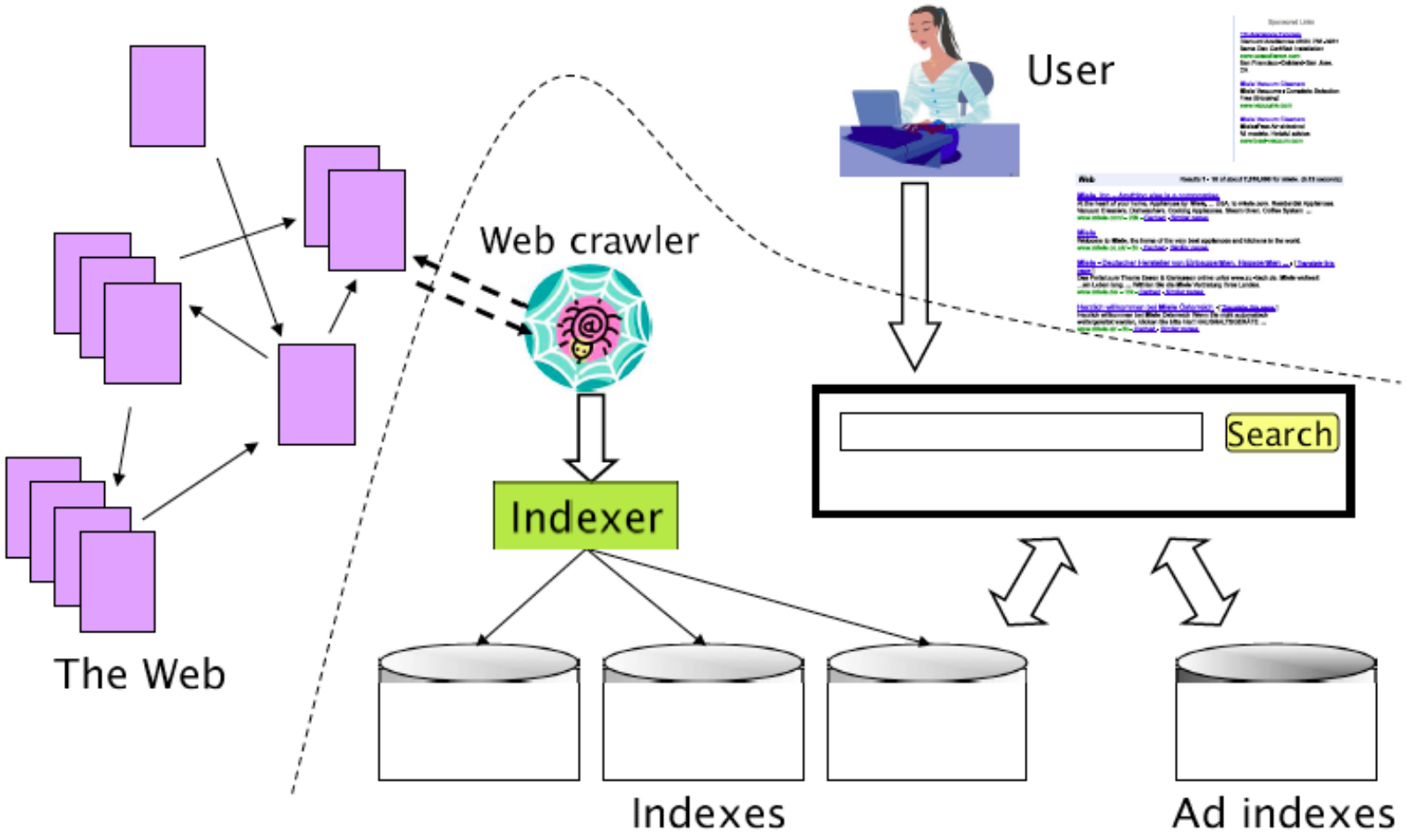
Year	Annual Number of Google Searches	Average Searches Per Day
2016	3,293,250,000,000	9,022,000,000
2015	2,834,650,000,000	7,766,000,000
2014	2,095,100,000,000	5,740,000,000
2013	2,161,530,000,000	5,922,000,000
2012	1,873,910,000,000	5,134,000,000
2011	1,722,071,000,000	4,717,000,000
2010	1,324,670,000,000	3,627,000,000
2009	953,700,000,000	2,610,000,000
2008	637,200,000,000	1,745,000,000
2007	438,000,000,000	1,200,000,000
2000	22,000,000,000	60,000,000
1998	3,600,000 *Googles official first year*	9,800

actual (2023) estimate is 8.5 billions queries per day

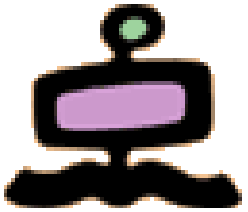
To google


© Original Artist
Reproduction rights obtainable from
www.CartoonStock.com
"Did you Google me before I was born to see if I was going to be a boy or a girl?"

Architecture of a Search Engine

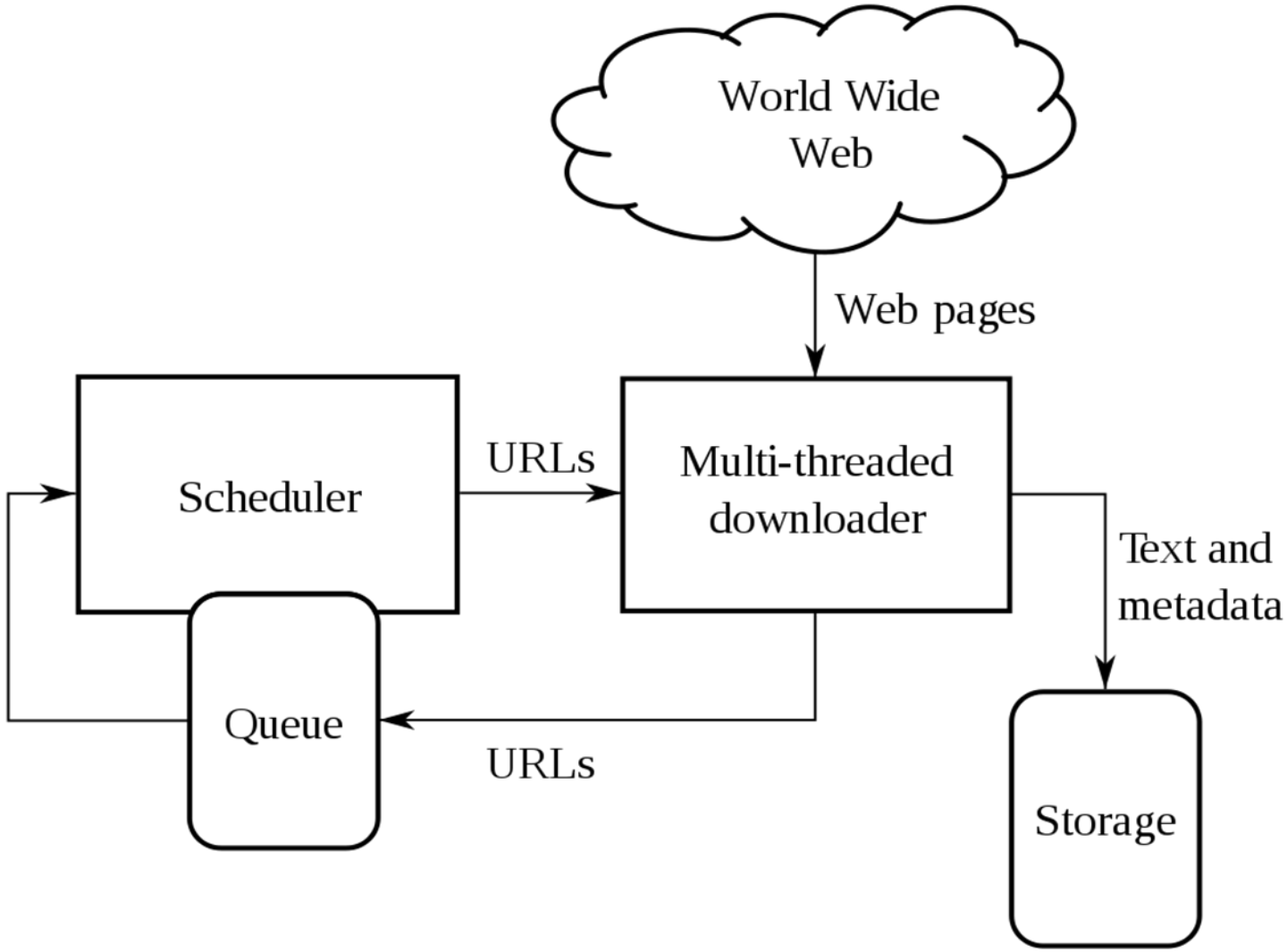


Main functions of a search engine

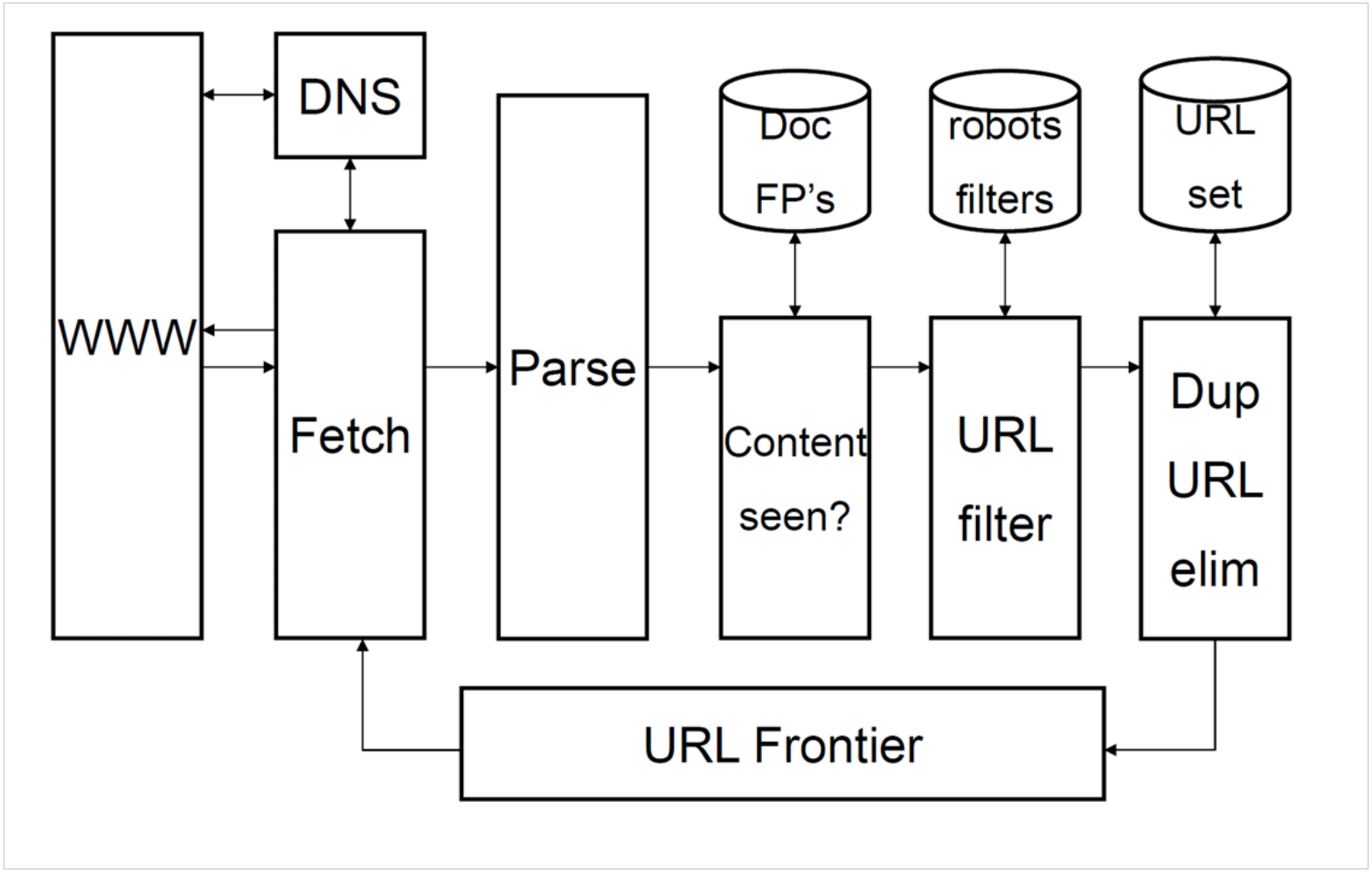


- Crawling 
- Indexing (in parallel with crawling)
- Ranking based on page content
- Ranking based on Web considerations
- Display of results

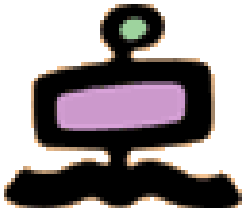
Basic architecture of a crawler (spider, bot)



Crawler architecture

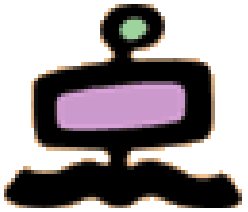


Crawling



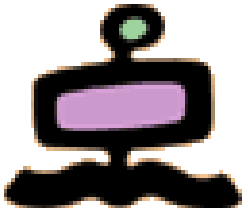
- A web crawler (aka a *spider* or a *bot*) is a program
  - Starts with one or more URL – the *seed*
    - Other URLs will be found in the pages pointed to by the seed URLs. They will be the starting point for further crawling
  - Uses the standard protocols (HTTP, FTP) for requesting a resource from a server
    - Requirements for respecting server policies
    - Politeness
  - Parses the resource obtained
    - Obtains additional URLs from the fetched page
    - Provides the fetched page to the indexer
  - Implements policies about content
    - Recognizes and eliminates duplicate or unwanted URLs
  - Adds the URLs found in the fetched page to the queue and continues requesting pages

What any crawler must do



- A crawler must be
  - Robust: Survive *spider traps*. Websites that fool a spider into fetching large or limitless numbers of pages within the domain.
    - Some deliberate; some errors in site design
  - Polite:: Crawlers can interfere with the normal operation of a web site. Servers have policies, both implicit and explicit, about the allowed frequency of visits by crawlers. Responsible crawlers obey these policies.

Politeness



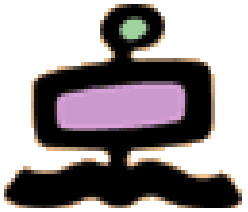
- Explicit
  - Specified by the web site owner
  - What portions of the site may be crawled and what portions may not be crawled
    - robots.txt file
- Implicit
  - If no restrictions are specified, still restrict how often you hit a single site.
  - You may have many URLs from the same site. Too much traffic can interfere with the site's operation. Crawler hits are much faster than ordinary traffic – could overtax the server. (Constitutes a denial of service attack) Good web crawlers do not fetch multiple pages from the same server at one time.

robots.txt example

- No robot should visit any URL starting with "/yoursite/temp/", except the robot called “searchengine":

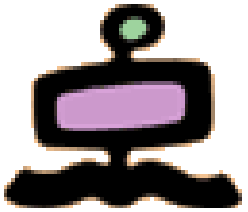
```
User-agent: *
Disallow: /yoursite/temp/

User-agent: searchengine
Disallow:
```

Scale of crawling



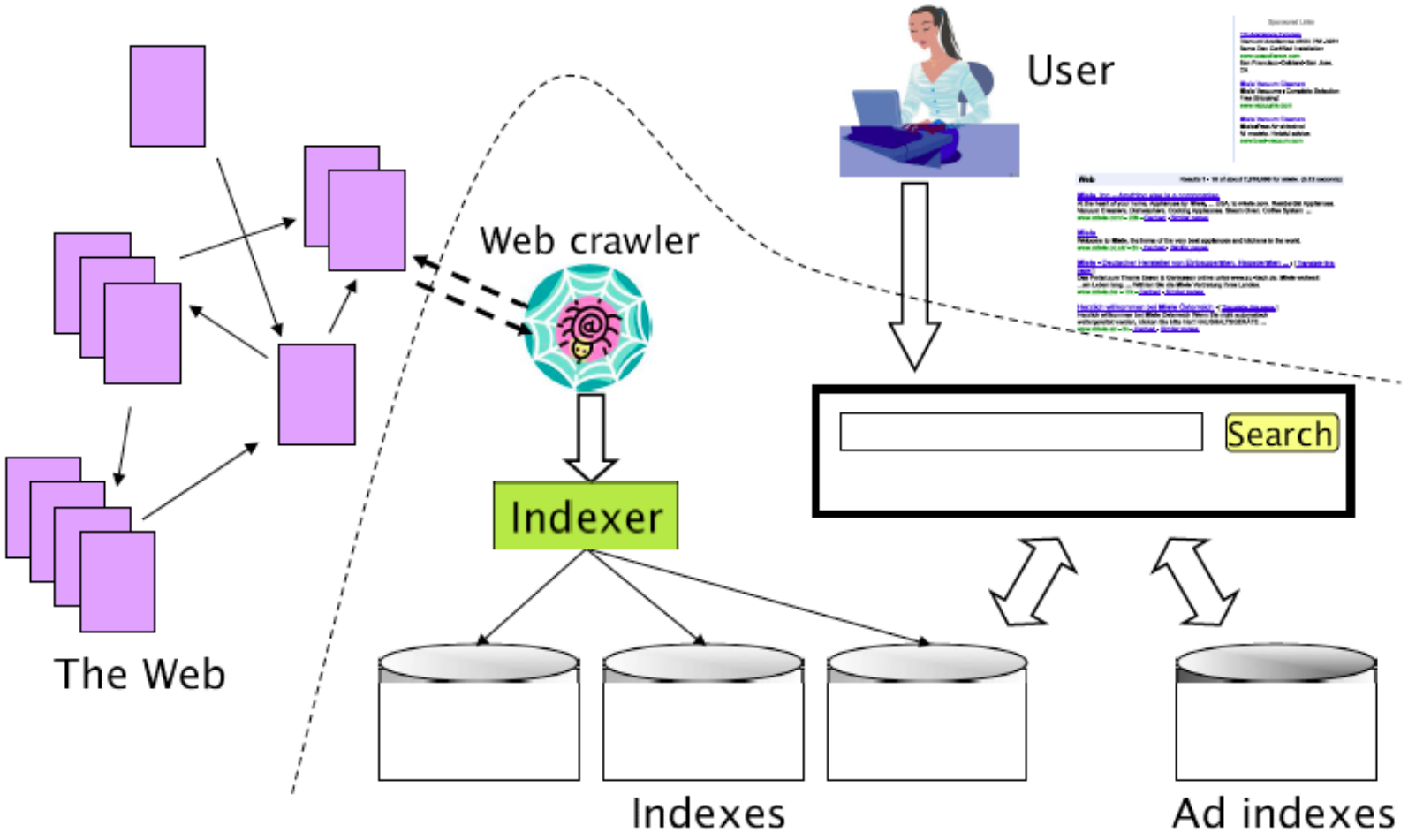
- A one month crawl of a billion pages requires fetching several hundred pages per second
- It is easy to lose sight of the numbers when dealing with data sources on the scale of the Web.
  - 30 days * 24 hours/day * 60 minutes/hour * 60 seconds/minute = 2,592,000 seconds
  - 1,000,000,000 pages/2,592,000 seconds **= 386** pages/second
- Note that those numbers assume that the crawling is continuous

Distributed crawler

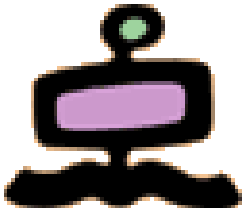


- For big crawls,
  - Many processes, each doing part of the job
    - Possibly on different nodes
    - Geographically distributed
  - How to distribute
    - Give each node a set of hosts to crawl
    - Use a hashing function to partition the set of hosts
  - How do these nodes communicate?
    - Need to have a common index

Architecture of a Search Engine

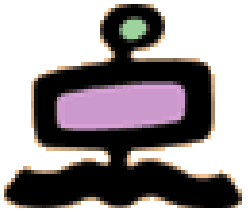


Main functions of a search engine



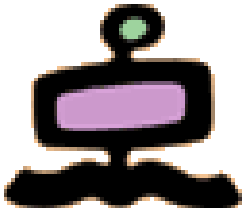
- Crawling
- Indexing (in parallel with crawling)
- Ranking based on page content
- Ranking based on Web considerations
- Display of results

Indexing



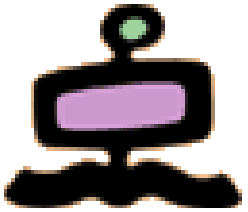
- Retrieved web page sent also to indexer to scan text (ignoring links)
- Build the index and the document (pages) representations (bag of words represented as vectors)
- Use of HTML information to improve the index and the “weight vectors”
- At query time, use the index and the weight vectors to get an initial ranking of relevant web pages, based on their content

Summary of retrieval and ranking



- Build a “term-document matrix”, assigning a weight to each term in a document (instead of just a binary value as in the simple approach)
  - Usually the weight is *tf.idf*, i.e. the product of the “term frequency” (number of occurrences of the term in the document) and the “inverse of the “term document frequency” (number of documents in which the term appears)
- Consider each document as a vector in n-space (n is the number of distinct terms, i.e. the size of the lexicon)
  - The non-zero components of the vector are the weights of the terms appearing in the document
  - Normalize each vector to “unit length” (divide each component by the modulus – the “length” – of the vector)
- Consider also the query as a vector in n-space
  - The non-zero components are just the terms appearing in the query (possibly with a weight)
  - Normalize also the query vector
- Define the similarity measure between the query and a document as the cosine of the “angle” beteen the two vectors
  - If both vectors are normalized, the computation is just the inner product of the two vectors

Final weight: tf x idf (or tf.idf)



- In conclusion, the weight of each term *i* in each document *d* ($w_{i,d}$) is usually given by the following formula (or very similar variations), called the *tf.idf* weight

$$w_{i,d} = tf_{i,d} \times \log(n / df_i)$$

$tf_{i,d} = \text{frequency of term } i \text{ in document } d$

$n = \text{total number of documents}$

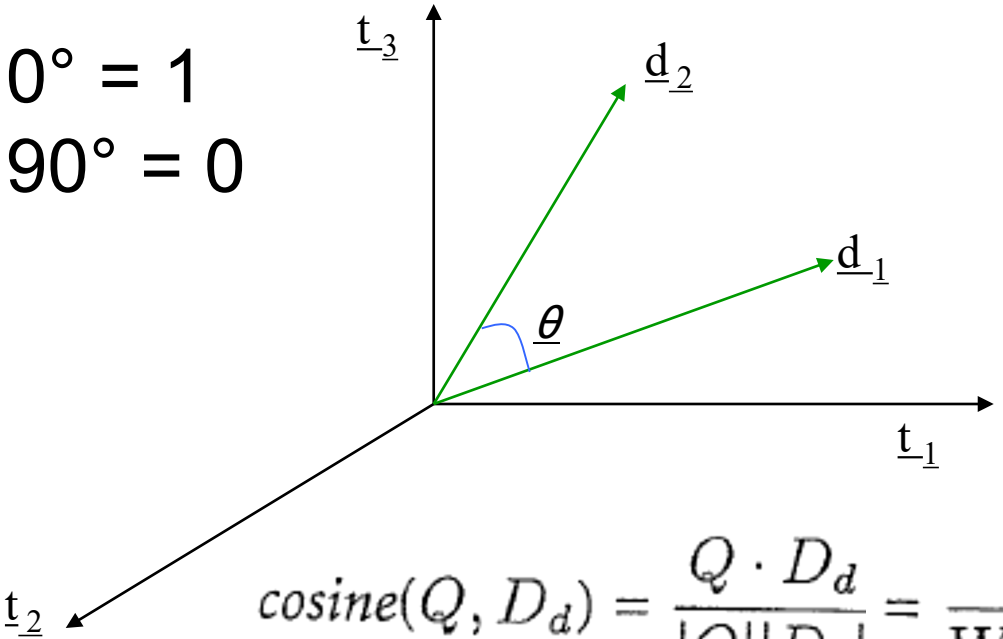
$df_i = \text{the number of documents that contain term } i$

- Increases with the number of occurrences *within* a doc
- Increases with the rarity of the term *across* the whole corpus

Similarity in vector space

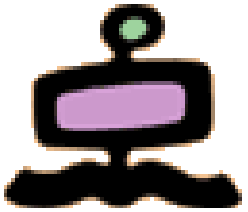
- Similarity between vectors d_1 and d_2 is *captured* by the **cosine** of the angle *x* between them.
- Note – this is *similarity*, not distance

cos 0° = 1
cos 90° = 0



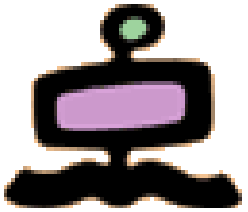
$$cosine(Q, D_d) = \frac{Q \cdot D_d}{|Q||D_d|} = \frac{1}{W_q W_d} \sum_{t=1}^{n} w_{q,t} \cdot w_{d,t}$$

“Boosting” of terms



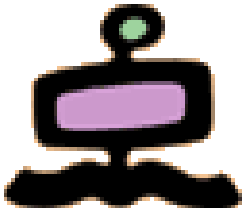
- A term coming from an HTML page is “more important” if it is:
  - In the title tag
  - In the page URL
  - In an HTML heading
  - In capital letters
  - Larger font
  - Early on in the page
  - In an HTML metatag
  - in the anchor text of a link pointing to that page
- A set of query terms is more important if they appear in the page:
  - Close together
  - In the right order
  - As a phrase

Main functions of a search engine



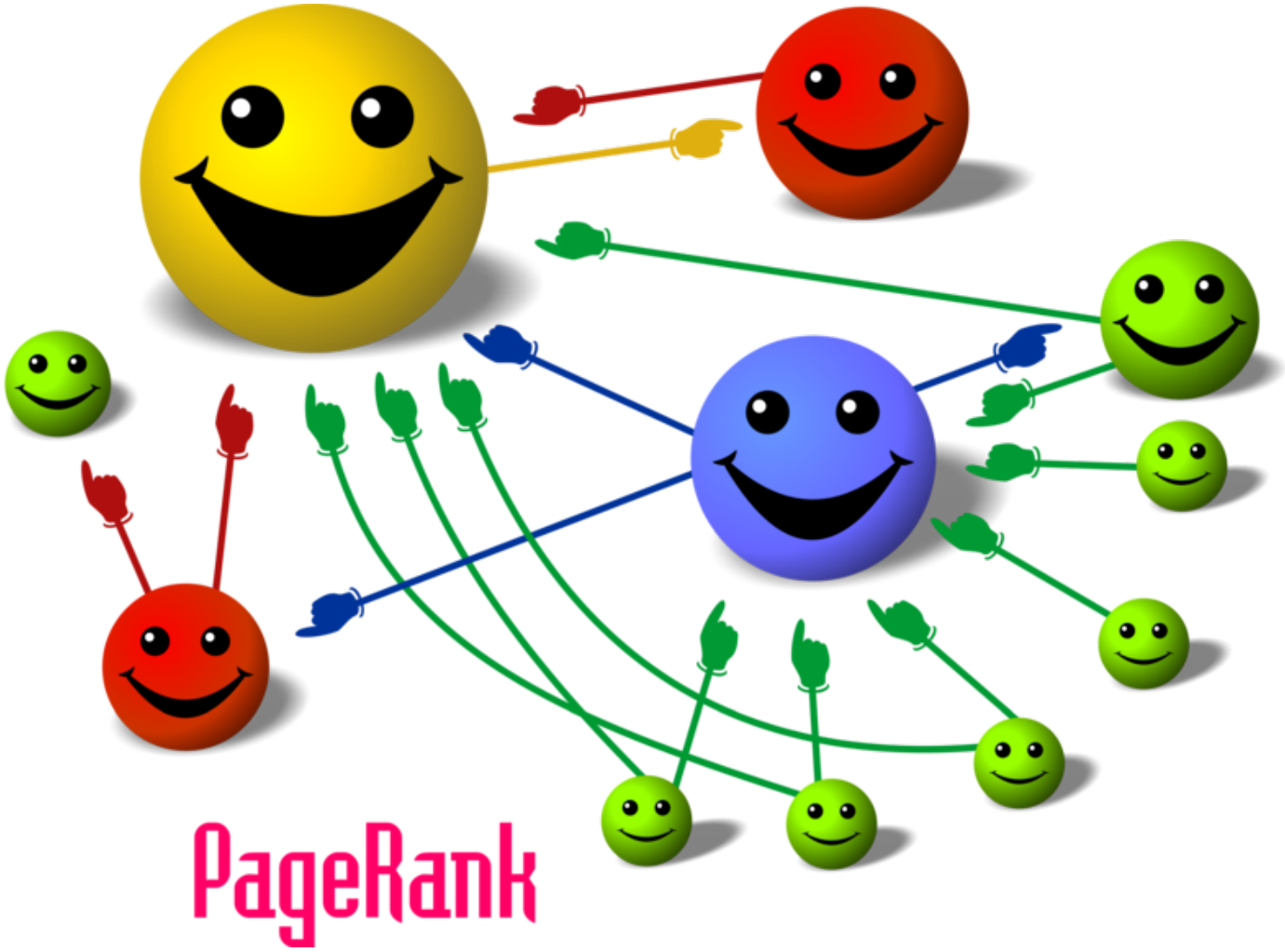
- Crawling
- Indexing (in parallel with crawling)
- Ranking based on page content
- Ranking based on Web considerations
- Display of results

Ranking in the Web

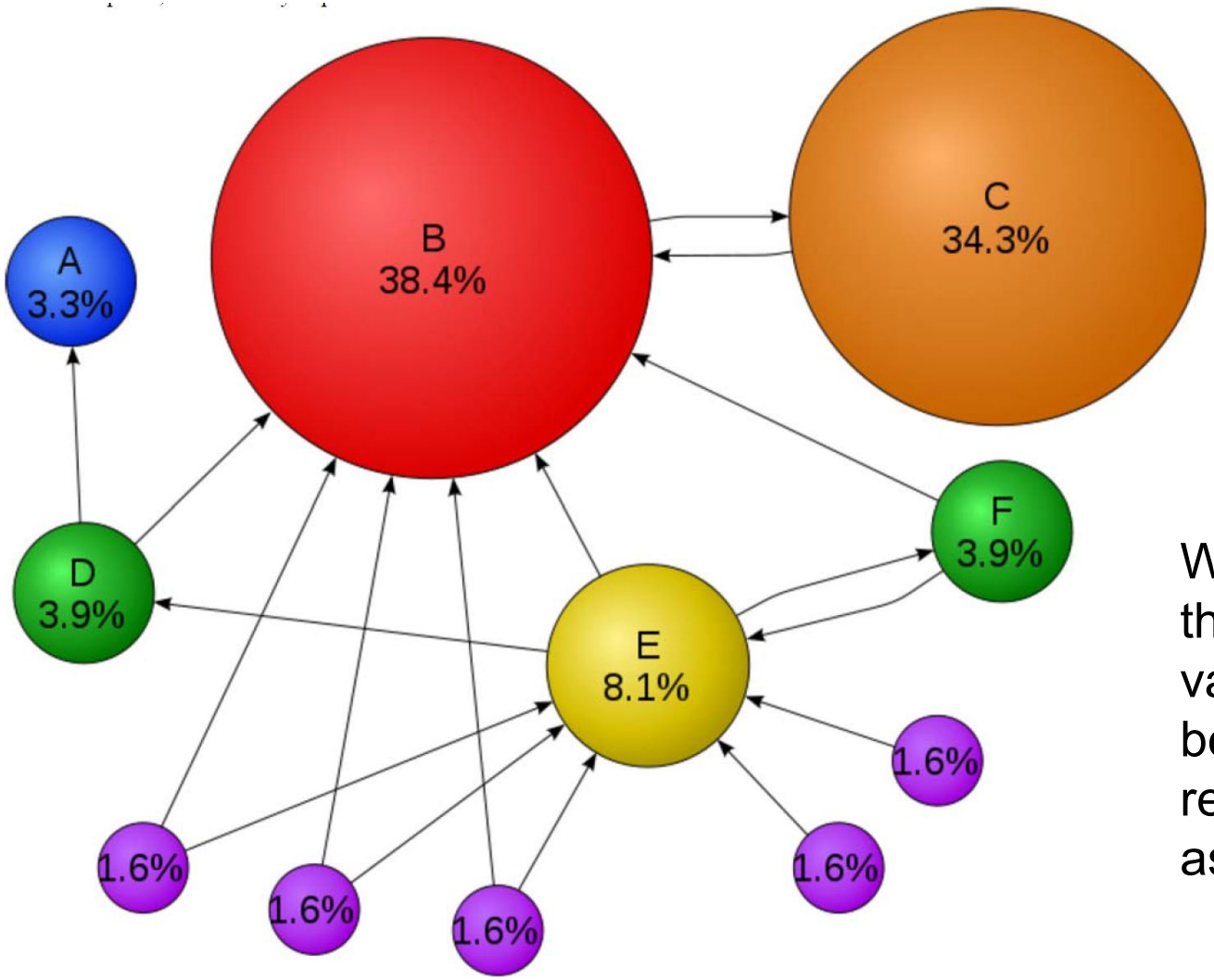


- Searching and ranking for “relevant documents” in a collection depends only on the content of the documents (free text search/retrieval, “bag of words” model)
- In the web, however, in addition to the page content there is the information provided by the hyperlinks from one web page to another
- The idea is therefore to rank the relevance of a web page based also on its “popularity” in the web, i.e. the number of links pointing to it from other web pages

The PageRank idea

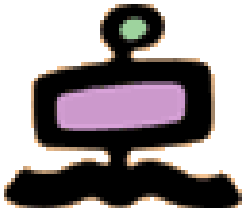


The PageRank values



We can consider the PageRank value as a number between 0 and 1, represented here as a percentage

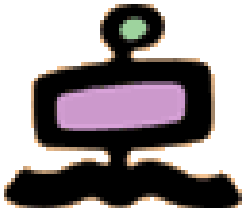
The PageRank algorithm



- The PageRank algorithm was published in 1996 by two students at Stanford University (Larry Page and Sergey Brin, the founders of Google)
  - the patent belongs to the University of Stanford and Google has the exclusive right to it
- The PageRank of a page is the sum of the values of the links pointing to it
- The value of an outgoing link is the PageRank of the page containing the link divided by the total number of outgoing links from that page
- Simple example for a “Web” of four pages, where pages B, C and D contain a link to page A:

$$PR(A) = \frac{PR(B)}{L(B)} + \frac{PR(C)}{L(C)} + \frac{PR(D)}{L(D)}.$$

The PageRank algorithm

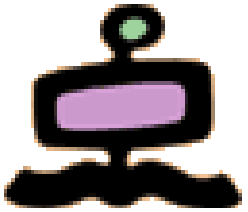


- More in general:

$$PR(u) = \sum_{v \in B_u} \frac{PR(v)}{L(v)}$$

- where B_u is the set of pages pointing to page *u* and *L(v)* is the number of outgoing links in page *v*
- In the mathematical model behind the PageRank algorithm, the rank of a page represents the probability that a **random surfer** sooner or later will land on that page
  - a random surfer starts navigation from random page of the web
  - clicks at random a link on that page
  - goes on “forever”
- The above is valid for a web completely connected
  - What if a page does not have outgoing links ?
  - What if a page does not have incoming links ?

Complete PageRank algorithm

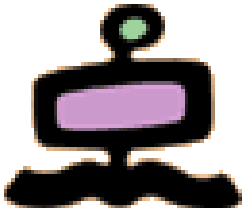


- To take into account “dangling pages”, the random surfer model is modified
  - At each page, the surfer can choose between clicking a link on that page, or jumping to a new page at random
  - The probability that the surfer clicks a link on that page is called the “damping factor”
- The final formula is (d is the damping factor, between 0 and 1, usually set at 0,85):

$$PR(p_i) = \frac{1-d}{N} + d \sum_{p_j \in M(p_i)} \frac{PR(p_j)}{L(p_j)}$$

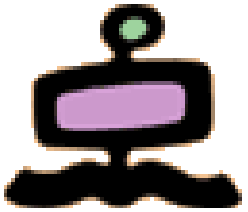
N is the total number of pages

Calculating the PageRank



- PageRank is a “normal” problem of linear algebra
  - a system of N equations in N unknowns
- For big (huge) systems, mathematicians have developed “iterative” ways to solve the system
  - all the pages are assigned an initial value (usually the same, 1/N)
  - the system is solved to get new values
  - the new values are assigned to the pages
  - the process is repeated until the difference with the previous step is negligible
- In the real Web, the number of iterations is in the order of 100, and the computation of the PageRank for all the pages may take several days

Search Engines considerations



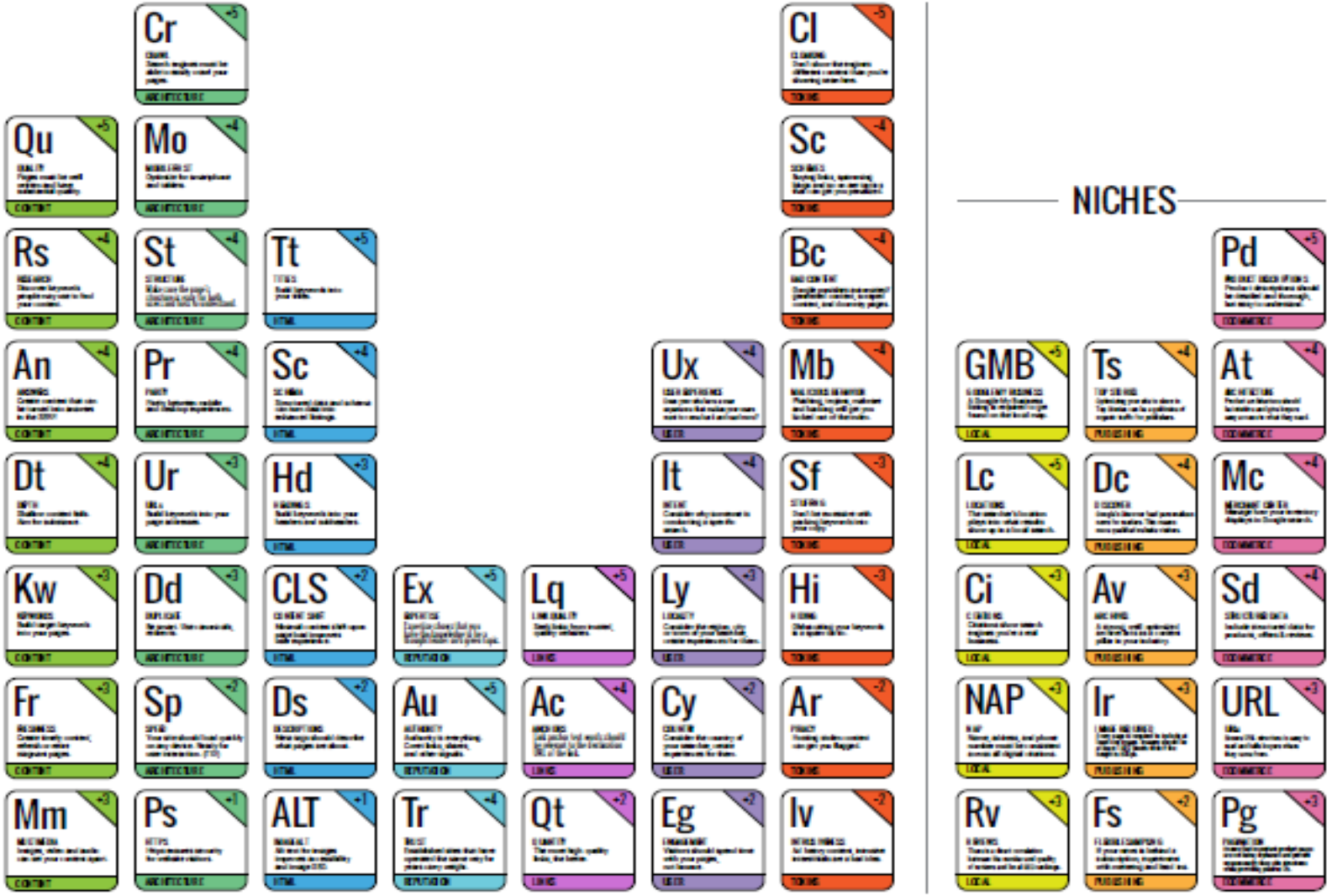
- Collection of query data (for statistics)
  - topics
  - time and location
  - number of clicks
- Search Engines Optimization (SEO)
  - increase the number of incoming links (link farms)
  - increase the PageRank of the pages pointing to it
  - divide a Web site into many pages
- Advertising on search engines
  - high volume of visitors
  - “knowledge” of web page content
  - targeted advertising

Just

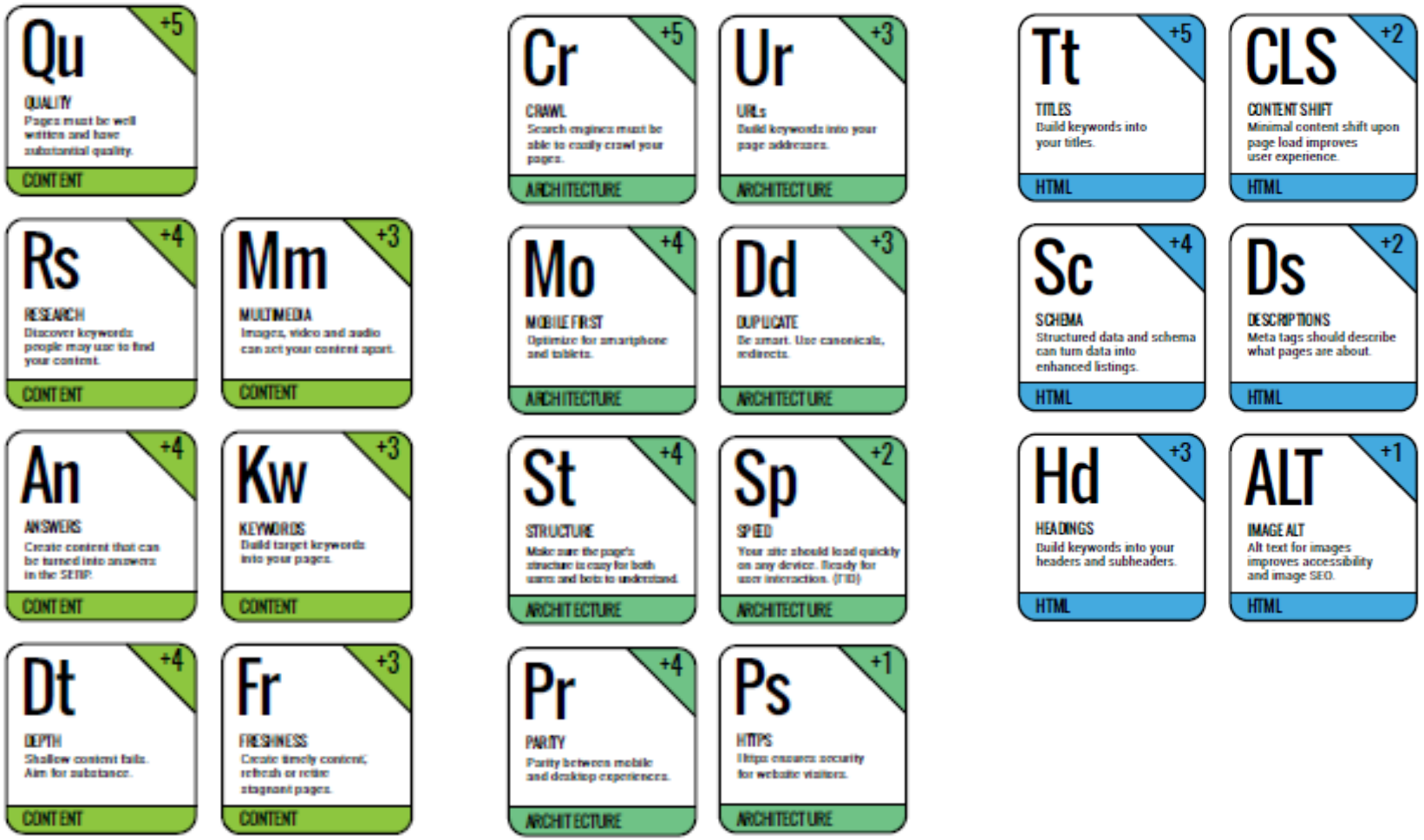
0.63%

of people click on the second page of Google search results.

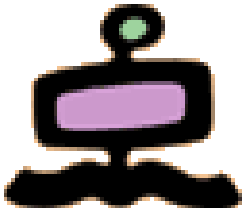
Table of SEO ranking factors



On-page SEO ranking factors



Off-page SEO ranking factors



Ex +5
EXPERTISE
Expertise shows that you have the knowledge to be a thought leader on a given topic.
REPUTATION

Lq +5
LINK QUALITY
Seek links from trusted, quality websites.
LINKS

Ux +4
USER EXPERIENCE
Does your site have a user experience that makes your users want to come back and read more?
USER

Au +5
AUTHORITY
Authority is everything. Covet links, shares, and other signals.
REPUTATION

Ac +4
ANCHORS
Link anchor text words should be relevant to the destination URL of the link.
LINKS

It +4
INTENT
Consider why someone is conducting a specific search.
USER

Cy +2
COUNTRY
Consider the country of your searcher, create experiences for them.
USER

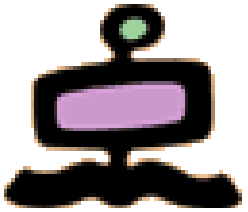
Tr +4
TRUST
Established sites that have operated the same way for years carry weight.
REPUTATION

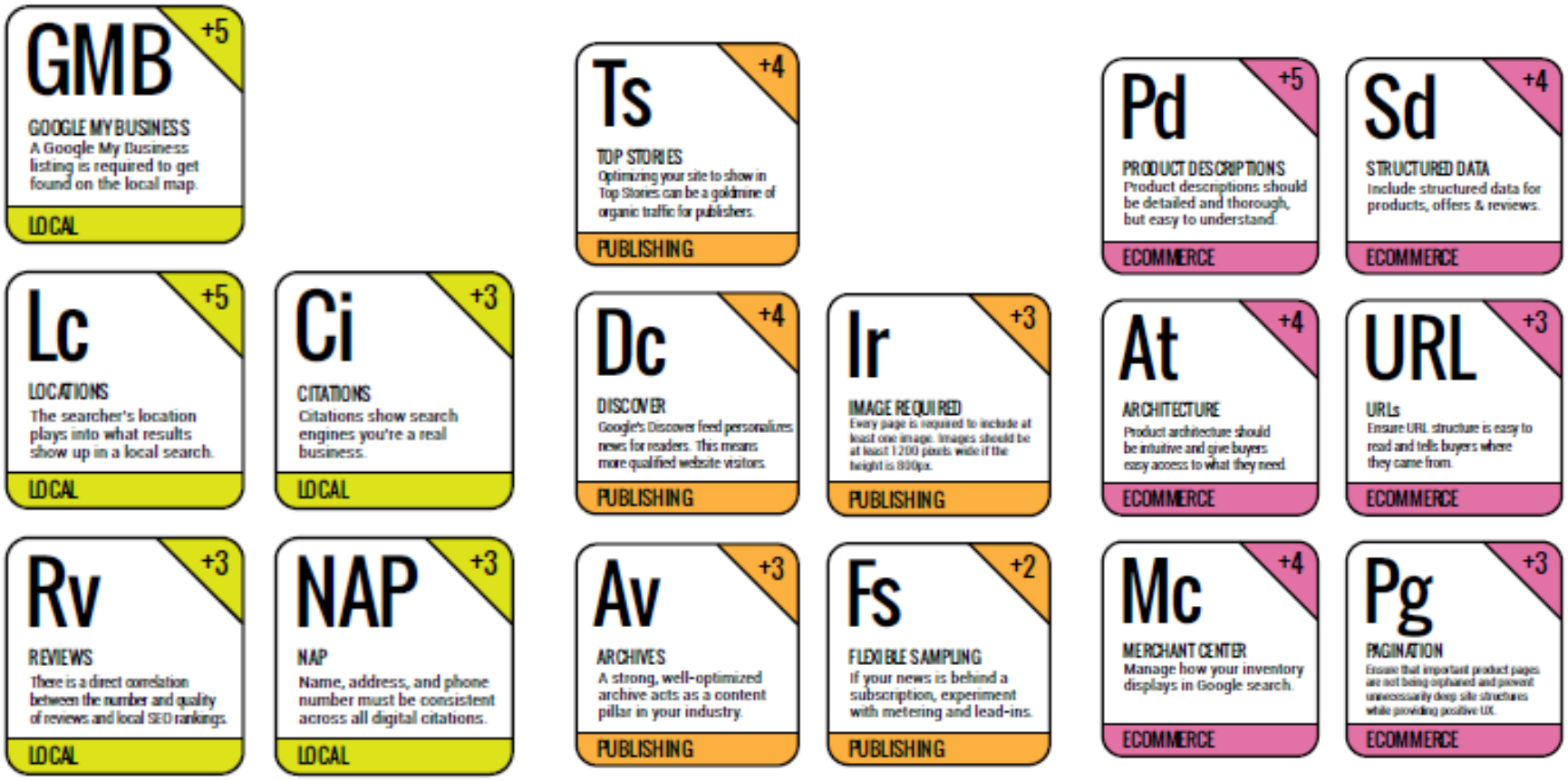
Qt +2
QUANTITY
The more high-quality links, the better.
LINKS

Ly +3
LOCALITY
Consider the region, city or town of your searcher, create experiences for them.
USER

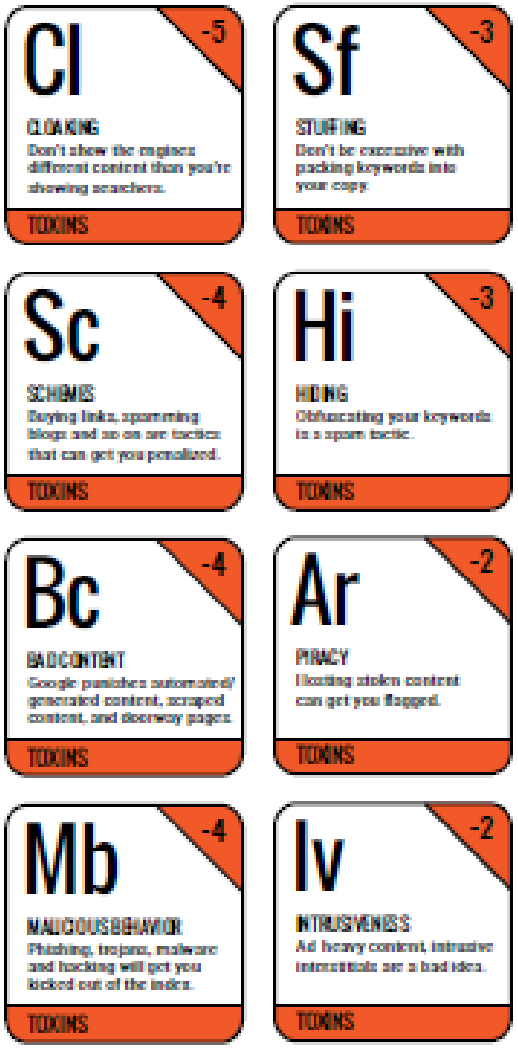
Eg +2
ENGAGEMENT
Visitors should spend time with your pages, not bounce.
USER

Additional SEO ranking factors

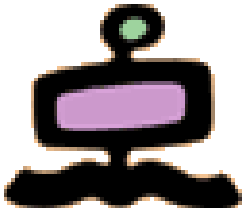




Negative SEO ranking factors

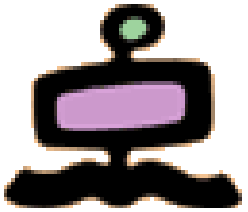


Violations



- Link spam
  - Link farms
  - Hidden links
  - Sybil attack
  - Page hijacking
  - Buying expired domains
  - Cookie stuffing
  - Using world-writable pages
  - Blog spam
  - Comment spam
  - Wiki spam
  - Referrer log spamming
- Content spam
  - Keyword stuffing
  - Hidden or invisible text
  - Meta-tag stuffing
  - Doorway pages
  - Scraper sites
  - Article spinning
- Other types
  - Mirror websites
  - URL redirection
  - Cloaking

Search Engines considerations



- Search Engines Optimization (SEO)
  - increase the number of incoming links (link farms)
  - increase the PageRank of the pages pointing to it
  - divide a Web site into many pages
- Collection of query data (for statistics)
  - topics
  - time and location
  - number of clicks
- Advertising on search engines
  - high volume of visitors
  - “knowledge” of web page content
  - targeted advertising

Advertizing on Google

- advertising is associated to “key words” (Google AdWords)
- ads are published on the result page of a query containing a keyword
- ads are paid “per click”
- ads may be published also on “partner sites” (Google AdSense)

Google advertising revenues