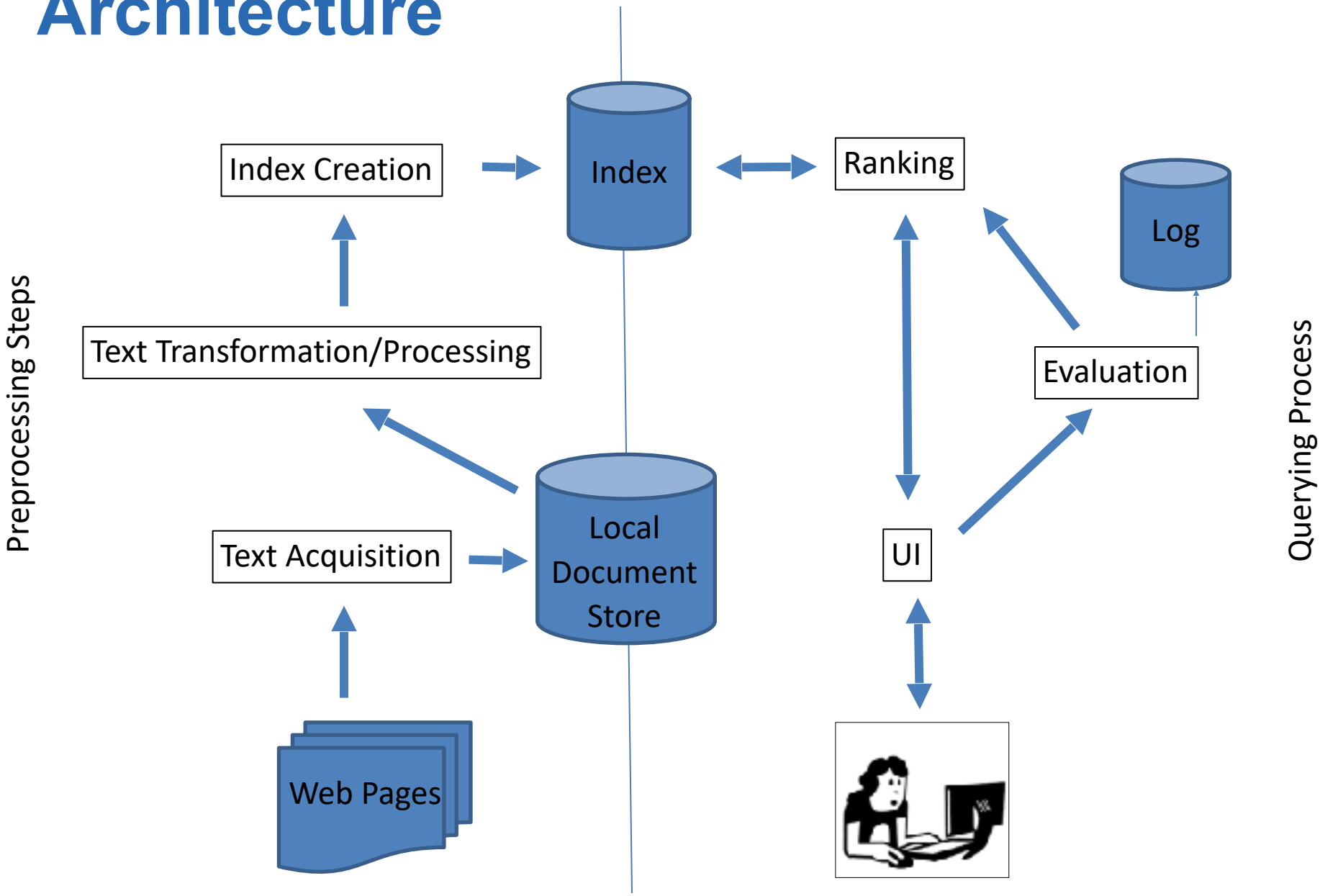
Informatics 141
Computer Science 121

Information Retrieval Lecture 2.2

Professor Iftekhar Ahmed
Department of Informatics
https://www.ics.uci.edu/~iftekha/

Architecture



History of Web Advertising

- Banner ads (1995-2001)
  - Initial form of web advertising
  - Popular websites charged *X*$ for every 1,000 "impressions" of the ad
    - Called "CPM" rate (Cost per thousand impressions)
    - Modeled similar to TV, magazine ads
  - From untargeted to demographically targeted
  - Low click-through rates
    - Low ROI for advertisers

CPM…cost per *mille*
Mille…thousand in Latin

Performance-based Advertising

- Introduced by Overture around 2000
  - Advertisers bid on search keywords
  - When someone searches for that keyword, the highest bidder's ad is shown
  - Advertiser is charged only if the ad is clicked on
- Similar model adopted by Google with some changes around 2002
  - Called Adwords

Ads on Search

- **Ad placement:**
  - **Fully automated**
  - **Balance auction price and relevance**
  - **Targeted advertising : focus on audience**

How companies pay for Ads on Search?

- **Cost Per Mil (CPM)**
  - Cost for showing the ad with 1,000-page shows
  - Important for branding campaigns
- **Cost Per Click (CPC)**
  - Cost for users clicking on the ad
  - Important for sales campaigns
- There are also other concepts
  - e.g. CPA: Cost per action/acquisition; if that click triggered some action; e.g. DuckDuckGo + Amazon.

Warning: Click Fraud

- **$7.2 billion USD were lost between 2016-2018 (Guardian)**
- **Google calls this “invalid clicks”**
- **Search engines have sophisticated tools to detect fraud**

Web 2.0

- **Interesting problem:**
  What ads to show for a given query?

Adwords Problem

- **Given:**
  - **1.** A set of bids by advertisers for search queries
  - **2.** A click-through rate for each advertiser-query pair
  - **3.** A budget for each advertiser (say for 1 month)
  - **4.** A limit on the number of ads to be displayed with each search query
- **Respond to each search query with a set of advertisers such that:**
  - **1.** The size of the set is no larger than the limit on the number of ads per query
  - **2.** Each advertiser has bid on the search query
  - **3.** Each advertiser has enough budget left to pay for the ad if it is clicked upon

Adwords Problem

- **A stream of queries arrives at the search engine: q_1, q_2, …**
- **Several advertisers bid on each query**
- **When query q_i arrives, search engine must pick a subset of advertisers whose ads are shown**

- **Goal: Maximize search engine's revenues**
  - **Simple solution:** Instead of raw bids, use the "**expected revenue per click**" (i.e., **Bid*CTR**)
- **Clearly we need an online algorithm!**

The Adwords Innovation

Advertiser	Bid	CTR	Bid * CTR
A	$1.00	1%	1 cent
B	$0.75	2%	1.5 cents
C	$0.50	2.5%	1.125 cents

Click through rate

Expected revenue

The Adwords Innovation

Advertiser	Bid	CTR	Bid * CTR
B	$0.75	2%	1.5 cents
C	$0.50	2.5%	1.125 cents
A	$1.00	1%	1 cent

Complications: Budget

- **Two complications:**
  - **Budget**
  - **CTR of an ad is unknown**

- **Each advertiser has a limited budget**
  - **Search engine guarantees that the advertiser will not be charged more than their daily budget**

Complications: CTR

- **CTR: Each ad has a different likelihood of being clicked**
  - **Advertiser 1** bids $2, click probability = 0.1
  - **Advertiser 2** bids $1, click probability = 0.5
  - **Clickthrough rate (CTR)** is measured **historically**
    - **Very hard problem: Exploration vs. exploitation**
      Exploit: Should we keep showing an ad for which we have good estimates of click-through rate
      or
      Explore: Shall we show a brand new ad to get a better sense of its click-through rate

Bad Scenario for Greedy

- **Two advertisers A and B**
  - ***A*** bids on query ***x***, ***B*** bids on ***x*** and ***y***
  - **Both have budgets of $4**
- **Query stream:** ***x x x x y y y y***
  - **Worst case greedy choice:** ***B B B B*** **_ _ _ _**
  - **Optimal: A A A A B B B B**

BALANCE Algorithm

- **BALANCE** Algorithm by Mehta, Saberi, Vazirani, and Vazirani
  - **For each query, pick the advertiser with the largest unspent budget**
    - **Break ties arbitrarily**

Example: BALANCE

- **Two advertisers A and B**
  - **A** bids on query ***x***, **B** bids on ***x*** and ***y***
  - **Both have budgets of $4**
- **Query stream: *x x x x y y y y***
- **BALANCE choice: A B A B _ _**
  -

Paid Ad Placement

- **It costs money**
- **So...**

Search Engine Optimization (SEO)

- **Tuning of a web page/site/app to rank highly** in search results for certain queries

- **Alternative to paying for ad placement**
  - **Bonus: visibility through rank = greater trust**

- **It's marketing**: getting your content to your audience

SEO

- Motives
  - Commercial
  - Political
  - Religious
  - Lobbying

- Who does this?
  - Internally: webmasters, writers
  - Commercially: companies, consultants
  - Hosting services

SEO

- **Essentially a marketing – technology interface area**

- **How to do it:**
  - **https://support.google.com/webmasters/answer/40349?hl=en**
  - **http://freetools.webmasterworld.com/category/seo-tools**
  - **Many books. E.g. :**

SEO

- **Ethical** and **unethical** ways of doing it

- **Legitimate approach:**
  - Good incoming links
  - Good content, well written, well organized, up to date
  - Good use of web standards/practices
  - Fast servers (quick response)

SEO

- **Unethical approaches (aka spam):**
  - **fake pages**
  - **fake sites that point to your site**
  - **fake comments/engagement (bots!)**
  - **in short: “alternative facts” aka lies**

- **Sometimes the line between legitimate and illegitimate practices is hard to find. There’s a large grey area**

SEO

- **Search results (also) depend on which data center receives the query:**
  - **google.com vs. google.fr vs google.pt will show different results for a single query: e.g. “Paris”**

- **Different SEO strategies can be considered per country**

Keyword Stuffing

- 1st- gen search engines relied heavily on textual content and frequency of words

- SEO moved to play around with keywords
  - Misleading meta-tags

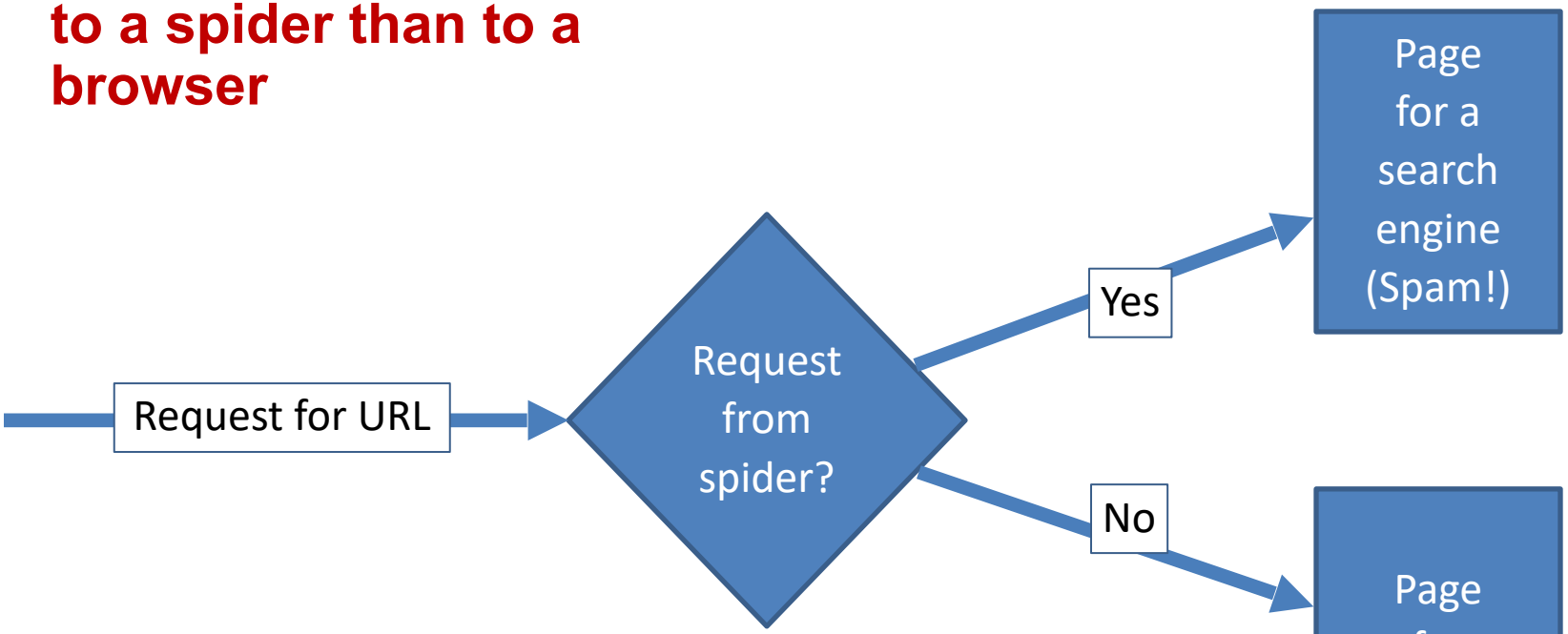
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.d
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<COMMENT TITLE="MONITOR"></COMMENT>
<meta http-equiv="Content-Language" content="en-us" />
<meta http-equiv="Content-type" content="text/html; charset=iso-8859-1"/>
<META NAME="ROBOTS" CONTENT="NOODP"><meta name="verify-v1" content="aeVxP6zTHeQzT620ipj5+ikXd/VXcdlKoYUJ/C6vVdY=" />
<META NAME="keywords" content="Expedia, Travel, Cheap Airfare, Car, Hotels, Vacations, Airfare, Car Rental, Cruises,

<META NAME="description" content="Purchase airline tickets, make hotel reservations, find vacation packages, car rent
```

  - Repeating words over and over
  - Playing games with colors (white on white)
    - visible to spiders/crawlers and indexers, but not users

Cloaking

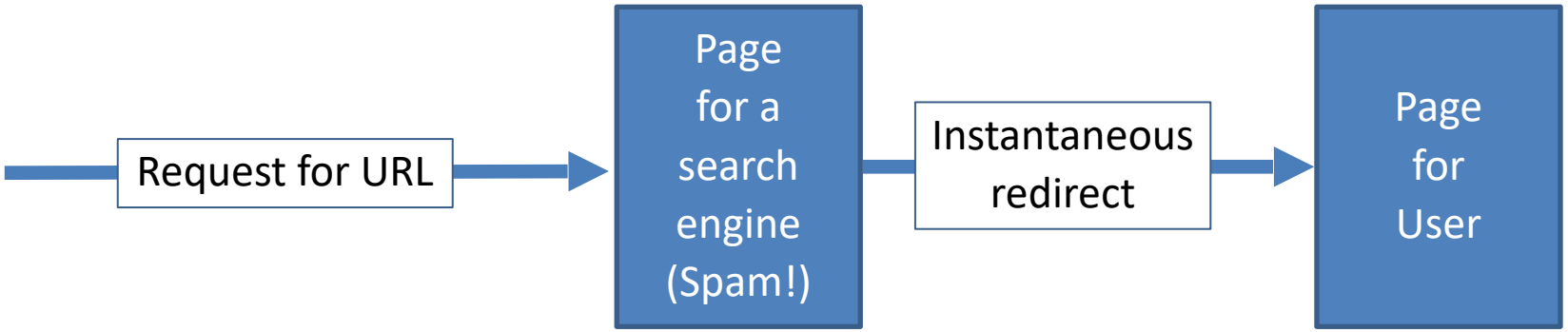
- **Serving different content to a spider than to a browser**



This architecture is the same as a dynamic content engine : *it can adapt to different search engines*

Doorway Pages

- **Like cloaking but using a redirect (302)**
  - **Code 302 : "The requested resource resides temporarily under a different URI. Since the redirection might be altered on occasion, the client SHOULD continue to use the Request-URI for future requests"**
  - **Initial page is optimized for spider, then redirect takes user to actual content; the user sees the final page.**



Link exchanges

- **I link to you, you link to me**
- **“Translations”**

Spam

Two major types

- **Link spamming**
  - Bots that search for blogs and leave comments with links
- **Clicker bots**
  - Bots that issue queries and “click” on targeted query results

Spam industry

The First Page of Google and How I Get my Clients There
www.malleeblue.com/1st-page-google-optimization-tips/
★★★★★ Rating: 10/10 · 276 reviews
Nov 8, 2016 - Can I get your website or page to rank on the first page of Google? ... If you want to get 1st page of Google search engine results listing, make ...

First Page of Google GUARANTEED !! | How To Get Your Business At ...
https://www.youtube.com/watch?v=qjASS_NsIL8
Dec 26, 2013 · Uploaded by Spotlight Ventures
Business on The First Page of Google GUARANTEED ! or Gotc... video, blog, or Business Listing on the ...

How To Get On The First Page of Google In 24 hours - YouTube
https://www.youtube.com/watch?v=20_pdMbJR3k
May 15, 2016 - Uploaded by Amazing Tricks World
how to get first page rank on google how to get first rank in google search how to rank first on google how ...

The war on spam

- **Quality indicators**
  - **Statistical analysis of links**
  - **Statistical analysis of votes**

The war on spam

- **Quality indicators**
  - **Statistical analysis of links**
  - **Statistical analysis of votes**
- **Usage indicators**
  - **Analytics**

The war on spam

- **Quality indicators**
  - **Statistical analysis of links**
  - **Statistical analysis of votes**
- **Usage indicators**
  - **Analytics**
- **Anti-bot mechanisms**
  - **Captchas**

The war on spam

- Limits on meta keywords
- Family-friendly filters
- Robust link analysis
  - Ignore statistically improbable links:
    - A page in Portuguese from Portugal links to a page in Chinese from China
  - Detect cycles
  - Use link analysis to detect spammers
    - Very simple rule : Guilt by association
- Editorial (human!) intervention
  - Black-listing and creation of training sets for supervised ML
- Spam recognition by machine learning

Webmaster Guidelines

- **Search engines have SEO policies**
  - **What is allowed and not allowed**

- **Must not be ignored**
  - **Once a site is blacklisted by a search engine, it will virtually disappear from the Web**

SEO “tricks” are volatile

- **Dependent on how the web search engines work**

- **The methods behind the engines change**
  - **Google: PageRank (1996 - ?), Panda (2011), Penguin (2012), Hummingbird (2013), Pigeon (2014), …**
    - **Words and links.**

SEO “tricks” are volatile

- **Dependent on how the web search engines work**

- **The methods behind the engines change**
  - **Google: PageRank (1996 - ?), Panda (2011), Penguin (2012), Hummingbird (2013), Pigeon (2014), …**
    - **Words and links.**
  - **Since 2016, it was using RankBrain: machine learning based.**
    - **Concepts.**

SEO “tricks” are volatile

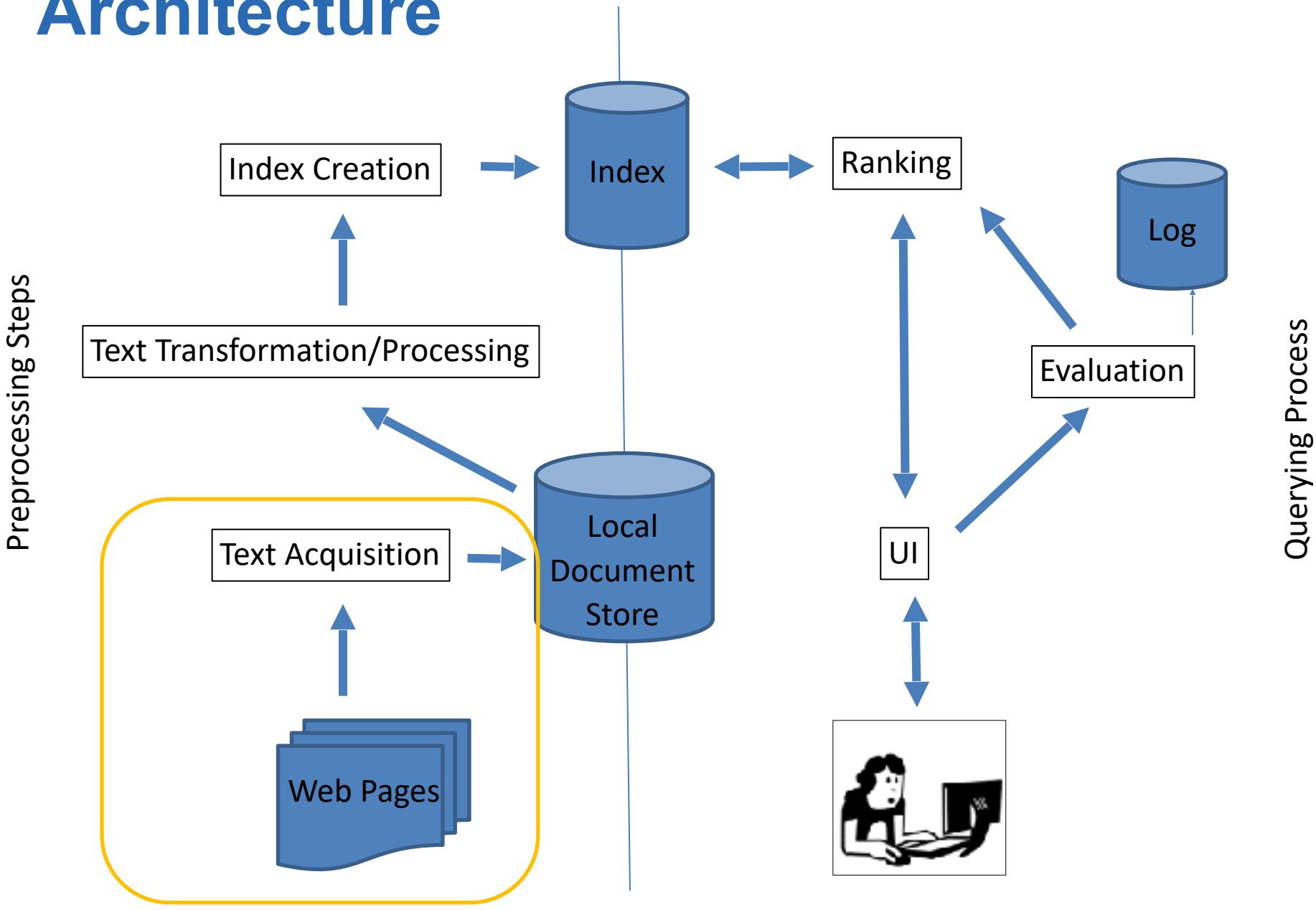
- Dependent on how the web search engines work
- The methods behind the engines change
  - Google: PageRank (1996 - ?), Panda (2011), Penguin (2012), Hummingbird (2013), Pigeon (2014), …
    - Words and links.
  - Since 2016, it was using RankBrain: machine learning based.
    - Concepts.
  - Since October 2019, Google is using BERT: NLP, machine learning.
  - Now Google Search uses : RankBrain + BERT + … > 200 methods&signals (!)
  - Can you optimize for BERT? Google says no (but too early to tell).

SEO “tricks” are volatile

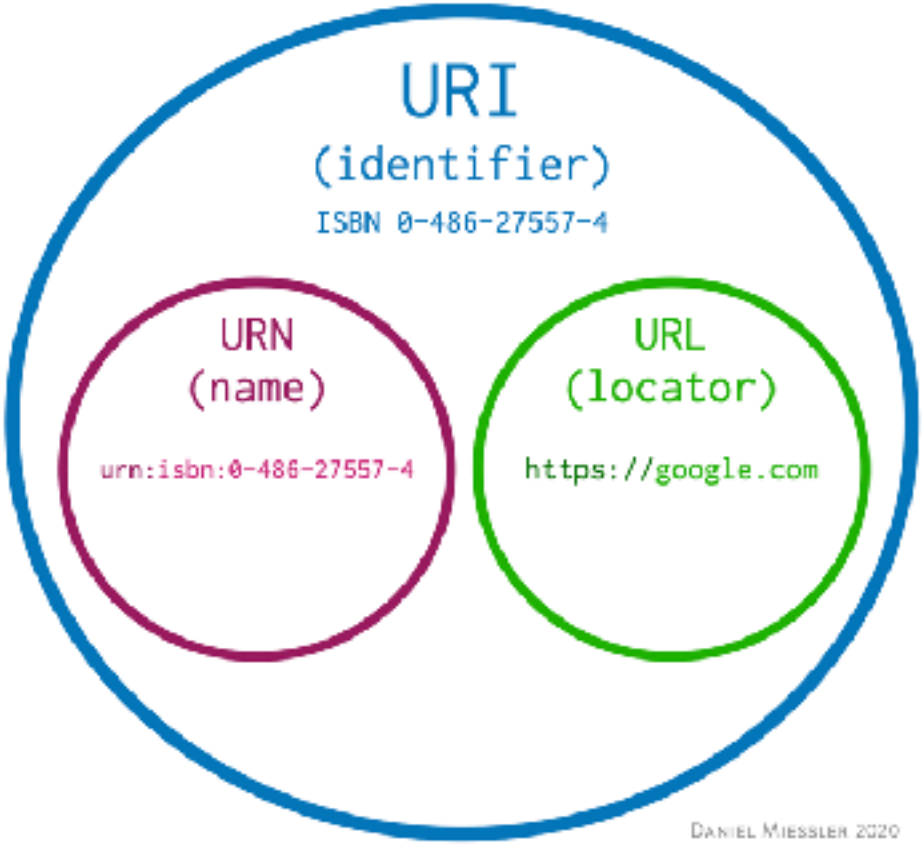
- **Solution :**

Optimizing for users will guarantee to optimize for good search engines

Architecture



Universal Resource Identifiers



All URLs are URIs, but not all URIs are URLs

“URL” is a specific type of URI that provides an access method/location

Anatomy of a URL

- **Syntax:**
  - **scheme://domain:port/path?query_string#fragment_id**

  - ***(the entire picture is slightly more complicated than this)***

- **Full spec:**
  - **http://www.w3.org/Addressing/URL/url-spec.txt**

Anatomy of a URL

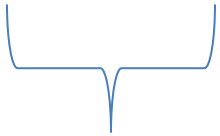
on a web server

no port! just domain

path

- **http://calendar.ics.uci.edu/calendar.php?type=month&calendar=1&category=&month=02&year=2013**

- **Domains and subdomains:**
  - **calendar.ics.uci.edu**



Domain name

Different Flavors of Web Data Collection

- **How to acquire data?**
  - **Data dumps**
  - **URL downloads**
  - **Web APIs**
  - **Web Crawling**

Data dumps

- **Sites may package their data periodically and provide it as a “dump”**
  - **Example: Wikipedia (it suggests you to Torrent)**
  - **arXiv Bulk Full-Text Access: https://arxiv.org/help/bulk_data_s3**

URL Downloads

- **Two step process:**
  1. **Find out the URLs of specific resources**
  2. **Run a downloader that takes that list and downloads the resources**

- **Example: "crawling" (!) sourceforge / github for source code**

- **Doesn't need to be source code; can be papers, pages, etc.**
  - **http://link.springer.com/chapter/10.1007/978-3-642-34213-4_1**
  - **http://link.springer.com/chapter/10.1007/978-3-642-34213-4_2**
  - **...**

- **Some sites use regular URLs. E.g. Google Code**
  - **http://code.google.com/p/python-for-android/downloads/list**
  - **...**

Web APIs

- Sites may provide (REST) interfaces for getting their data
  - Usually higher-level: avoids having to parse HTML
  - Usually restrictive: only part of the data

- Examples:
  - Facebook Graph API
    - My data in facebook api
    - More examples
  - Youtube API
  - Twitter API
  - arXiv API
  - ...

Web Crawling

- Like people, getting HTML pages and other documents and discovering new URLs as it goes
  - Good for changing collections
  - Good for unknown documents

- Web admins don't like crawlers
  - Crawlers consume resources that are meant for people
  - More on this...

Ways of Acquiring Web Data

- **Data dumps**
- **Web APIs**
- **Targeted downloads**

- **Web crawling** ← **last option**

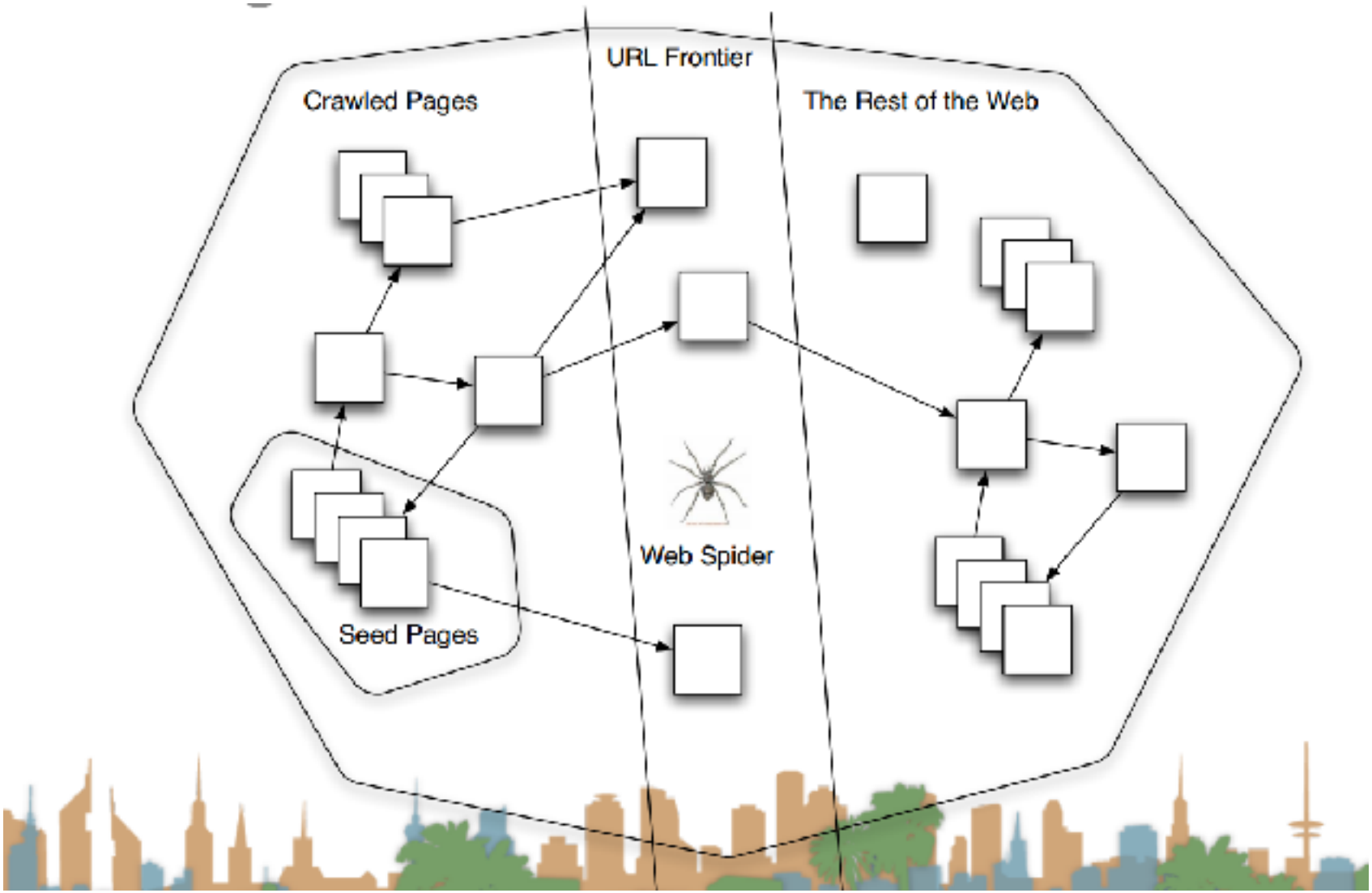
Crawling

- **Information Retrieval**

Basic Crawl Algorithm

- **Initialize a queue of URLs (seeds)**
- **Repeat until no more URLs in queue:**
  - **Get one URL from the queue**
  - **If the page can be crawled, fetch associated page**
  - **Store representation of page**
  - **Extract URLs from page and add them to the queue**
- **Queue = “frontier”**

Basic Crawl Algorithm



Pseudo Code

```
procedure CrawlerThread(frontier)
    while not frontier.done() do
        website ← frontier.nextSite()
        url ← website.nextURL()
        if website.permitsCrawl(url) then
            text ← retrieveURL(url)
            storeDocument(url, text)
            for each url in parse(text) do
                frontier.addURL(url)
            end for
        end if
        frontier.releaseSite(website)
    end while
end procedure
```

Pseudo Code

```
procedure CrawlerThread(frontier)
    while not frontier.done() do
        website ← frontier.nextSite()
        url ← website.nextURL()
        if website.permitsCrawl(url) then
            text ← retrieveURL(url)
            storeDocument(url, text)
            for each url in parse(text) do
                frontier.addURL(url)
            end for
        end if
        frontier.releaseSite(website)
    end while
end procedure
```

Pseudo Code

```
procedure CrawlerThread(frontier)
    while not frontier.done() do
        website ← frontier.nextSite()
        url ← website.nextURL()
        if website.permitsCrawl(url) then
            text ← retrieveURL(url)
            storeDocument(url, text)
            for each url in parse(text) do
                frontier.addURL(url)
            end for
        end if
        frontier.releaseSite(website)
    end while
end procedure
```

But how do you know if the website allows you to crawl?

Permission to crawl

Permission to crawl

- **Be polite: try to ask the website if you can crawl it first!**

- **Robots Exclusion Standard aka robots.txt**

  - **Sites may have that file at the root. Examples:**
    - **http://www.cnn.com/robots.txt**
    - **http://en.wikipedia.org/robots.txt**

  - **Very simple syntax (but no formal/official standard yet!):**
    - **http://www.robotstxt.org/robotstxt.html**

Permission to crawl

Exclude all

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

Allow all

```
user-agent: *
Disallow:
```

Exclude all from a part of the site

```
User-agent: *
Disallow: /cgi-bin/
Disallow: /tmp/
Disallow: /junk/
```

Exclude a single robot

```
User-agent: BadBot
Disallow: /
```

Allow a single robot

```
User-agent: Google
Disallow:

User-agent: *
Disallow: /
```

http://www.robotstxt.org/robotstxt.html

Question

```
User-agent: *
Disallow: /foo
Disallow: /bar

User-agent: Googlebot
Disallow: /baz/a
```

According to this, the Googlebot is?

A. Allowed to crawl /foo and /bar
B. Allowed to crawl /baz/a
C. Not allowed to crawl neither /foo nor /bar
D. Not allowed to crawl /baz/a

Permission to crawl

- **Robots Exclusion Standard aka robots.txt**
  - **Sites may have that file at the root. Examples:**
    - http://www.cnn.com/robots.txt
    - http://en.wikipedia.org/robots.txt
  - **Very simple syntax (but no formal/official standard yet!):**
    - http://www.robotstxt.org/robotstxt.html
  - **Honor basis!**
    - **It's not a security mechanism**

Information to crawlers

- **Sitemaps (introduced by Google)**
  - **Also listed in robots.txt**
  - **Allow web masters to send info to crawlers. E.g.**
    - **Location of pages that might not be linked**
    - **Relative importance**
    - **Update frequency**
- **Example:**
  - **http://www.cnn.com/robots.txt**

Sitemap Example

```
<?xml version="1.0" encoding="UTF-8"?>
<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">
  <url>
    <loc>http://www.company.com/</loc>
    <lastmod>2008-01-15</lastmod>
    <changefreq>monthly</changefreq>
    <priority>0.7</priority>
  </url>
  <url>
    <loc>http://www.company.com/items?item=truck</loc>
    <changefreq>weekly</changefreq>
  </url>
  <url>
    <loc>http://www.company.com/items?item=bicycle</loc>
    <changefreq>daily</changefreq>
  </url>
</urlset>
```

Pseudo Code

```
procedure CrawlerThread(frontier)
    while not frontier.done() do
        website ← frontier.nextSite()
        url ← website.nextURL()
        if website.permitsCrawl(url) then
            text ← retrieveURL(url)
            storeDocument(url, text)
            for each url in parse(text) do
                frontier.addURL(url)
            end for
        end if
        frontier.releaseSite(website)
    end while
end procedure
```

“Basic algorithm” is...

- **Theoretically correct**

“Basic algorithm” is...

- **Theoretically correct**
- **Seriously lacking to use in practice**

“Basic algorithm” is...

- **Theoretically correct**
- **Seriously lacking to use in practice**
  1. **Will upset web admins (impolite)**
     - **It’s abusing the web servers**

“Basic algorithm” is...

- **Theoretically correct**
- **Seriously lacking to use in practice**
  1. **Will upset web admins (impolite)**
     - **It’s abusing the web servers**
  2. **Very slow**
     - **1 page at a time**

“Basic algorithm” is...

- **Theoretically correct**
- **Seriously lacking to use in practice**
  1. **Will upset web admins (impolite)**
     - **It’s abusing the web servers**
  2. **Very slow**
     - **1 page at a time**
  3. **Will get caught in traps and infinite sequences**

“Basic algorithm” is...

- **Theoretically correct**
- **Seriously lacking to use in practice**
  1. **Will upset web admins (impolite)**
     - **It’s abusing the web servers**
  2. **Very slow**
     - **1 page at a time**
  3. **Will get caught in traps and infinite sequences**
  4. **Will fetch duplicates without noticing**

“Basic algorithm” is...

- **Theoretically correct**
- **Seriously lacking to use in practice**
  1. **Will upset web admins (impolite)**
     - **It’s abusing the web servers**
  2. **Very slow**
     - **1 page at a time**
  3. **Will get caught in traps and infinite sequences**
  4. **Will fetch duplicates without noticing**
  5. **Will bring in data noise**

“Basic algorithm” is...

- **Theoretically correct**
- **Seriously lacking to use in practice**
  1. **Will upset web admins (impolite)**
     - **It’s abusing the web servers**
  2. **Very slow**
     - **1 page at a time**
  3. **Will get caught in traps and infinite sequences**
  4. **Will fetch duplicates without noticing**
  5. **Will bring in data noise**
  6. **Will miss content due to client-side scripting**

Architecture