Introduction to Information Retrieval

Präsentation für den Kurs

Wissensmanagement und Information Retrieval

im Master-Studiengang Strategisches Informationsmanagement
des Fachbereich 3: Wirtschaft & Recht der Frankfurt University of Applied Science

Kursleiter: Dipl. Inf Michael Christen, mc@yacy.net

Themen

- Grundlagen des Wissensmanagements
- Grundlagen des Information Retrievals
- Suchmaschinen Technologien (z.B. Crawling, Indexing, Ranking) und Architekturen
- Gesellschaftliche Aspekte und Anwendungen des Information Retrievals

Ablauf des Kurses

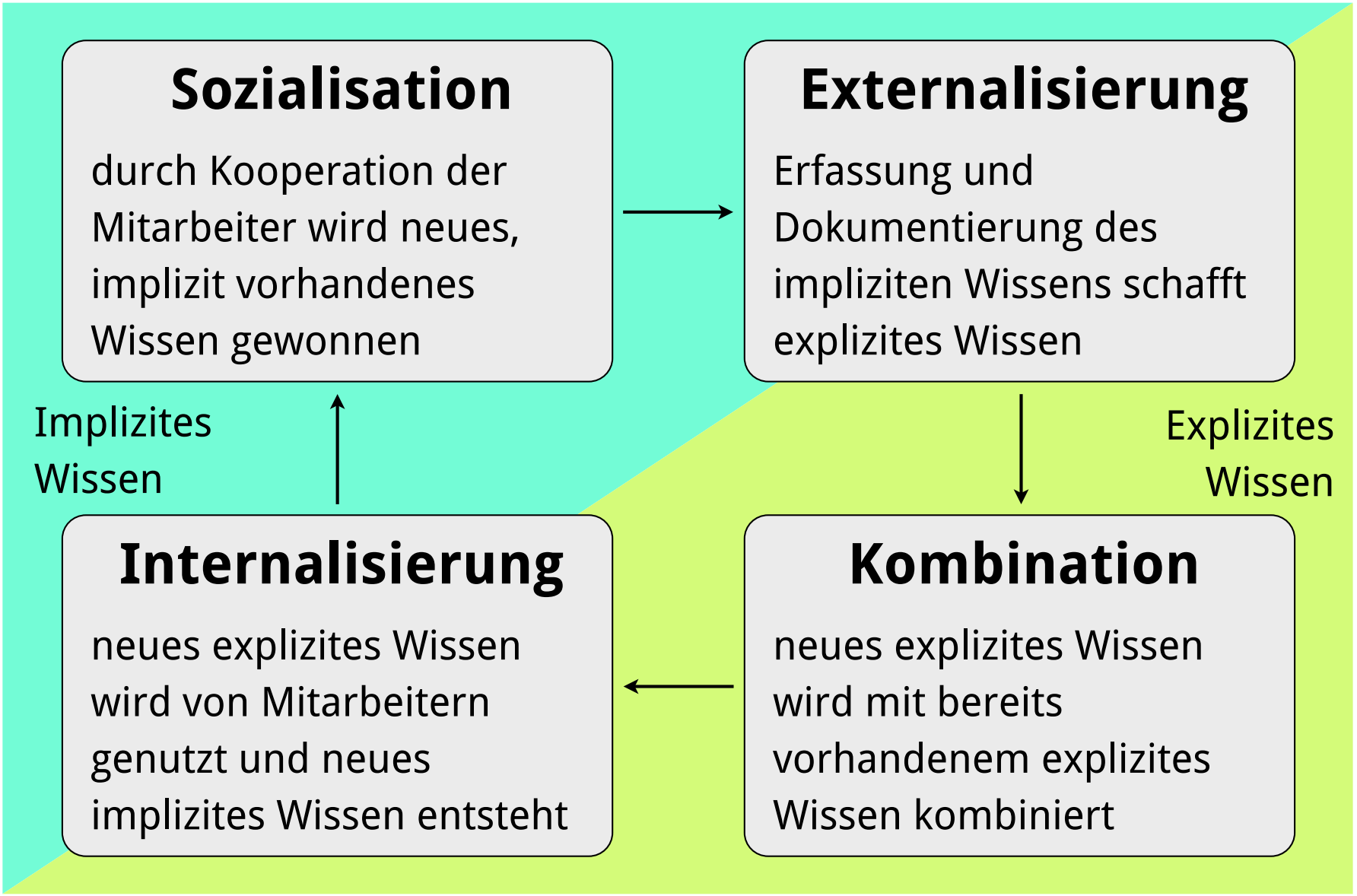
- Sie wählen heute ein Hausarbeitsthema: Vortrag zum Thema (ca. 30 Minuten + 15 Minuten Fragen+ Antworten), Ausarbeitung des Themas als Hausarbeit zum Semesterende.
- Einführung in Information Retrieval Themen, 4 Termine
- Sie stellen am 4. Termin die Gliederung der Hausarbeiten vor, je 5. Minuten
- Zeitplanung für die Vortragstermine vorrangig Themenorientiert bedingt; erste studentische Vorträge am 5. Termin (19. Mai)
- Bitte melden Sie sich in Moodle an: „Christen: Wissensmanagement und Information Retrieval - SS15„ https://elearning.frankfurt-university.de/course/view.php?id=5690

Literatur

- **Baeza-Yates, Ricardo; Ribeiro-Neto, Berthier: Modern Information Retrieval. ACM Press Series / Addison Wesley, New York, USA, 2011.**
- Ferber, Reginald: Information Retrieval. dpunkt-Verlag Heidelberg, Deutschland, 2003 (Online-Version frei im Internet).
- Krah, H.; Müller-Terpitz, R.: "Suchmaschinen"; Passauer Schriften zur interdisziplinären Medienforschung, Bd. 4, Passau, 2014.
- Katenkamp, Olaf, Implizites Wissen in Organisationen; Konzepte, Methoden und Ansätze im Wissensmanagement, Wiesbaden: VS-Verlag 2011[FH Frankfurt - elektronische Ressource].
- Balakrishnan, H; Kaashoek, F.; Karger, D.; Morris, R.; Stoica, I.: Looking Up Data in P2P Systems. In: Communications of the ACM, Vol. 46, No. 2, USA, 2003.
- Barbaro, Michael; Zeller, Tom Jr.: A Face Is Exposed for AOL Searcher No. 4417749. In: The New York Times, 9. August 2006.
- Frieder, O.; Grossmann, D.: Information Retrieval. Algorithms and Heuristics. Second Edition, Springer-Verlag, 2004.
- Manning, Christopher; Raghavan, Prabhakar; Schütze, Hinrich: Introduction to Information Retrieval. Cambridge University Press, 2008.
- Steinmetz, R.; Wehrle, K. (eds.): Peer-to-Peer Systems and Applications. Lecture Notes in Computer Science No. 3485, Springer-Verlag, Berlin-Heidelberg, 2005.
- Stock, Wolfgang: Information Retrieval. Oldenbourg Wissenschaftsverlag, München, 2007.

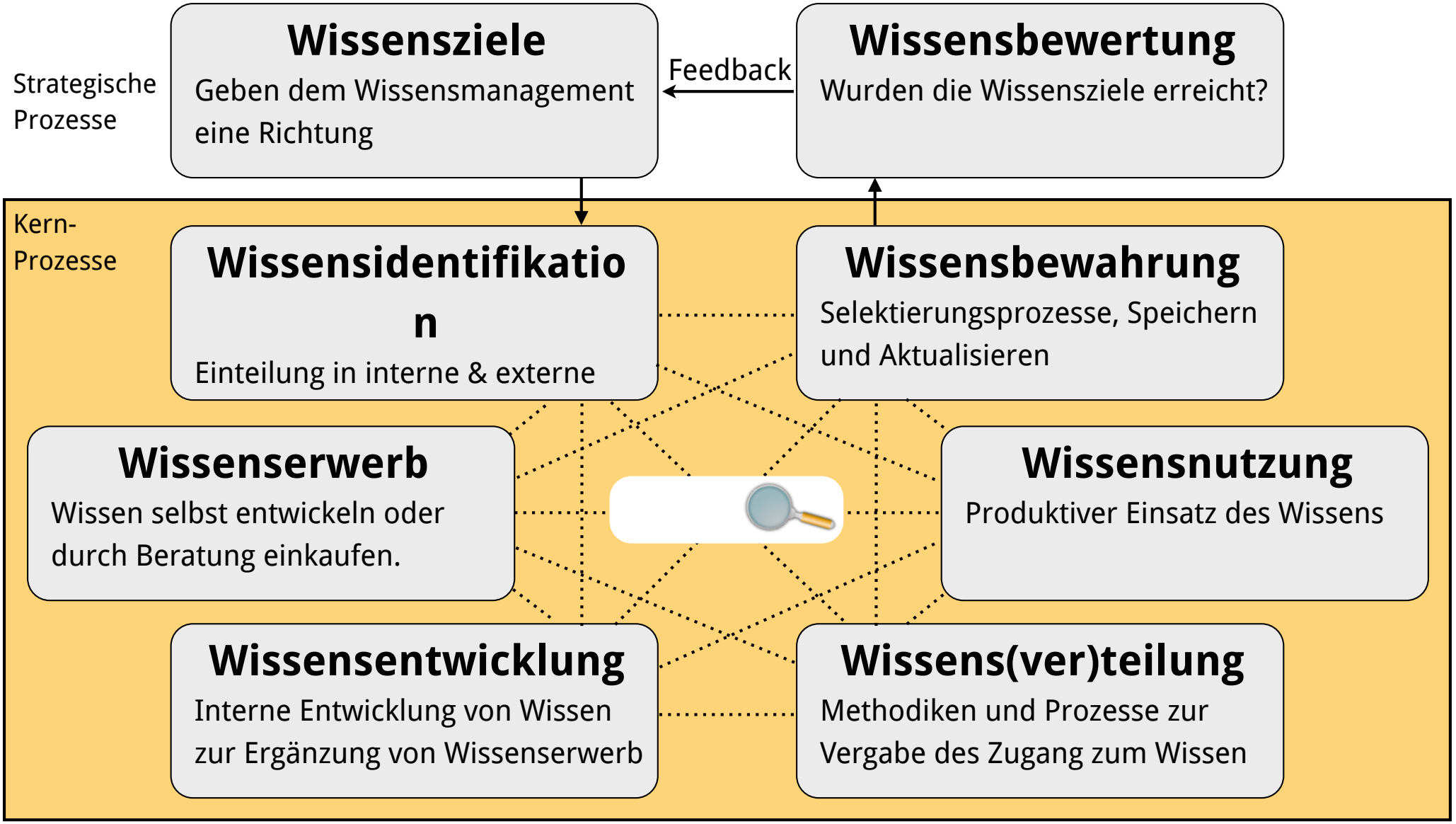
Wissensmanagement - SECI Model

(**S**ocialization, **E**xternalization, **C**ombination, **I**nternalization)



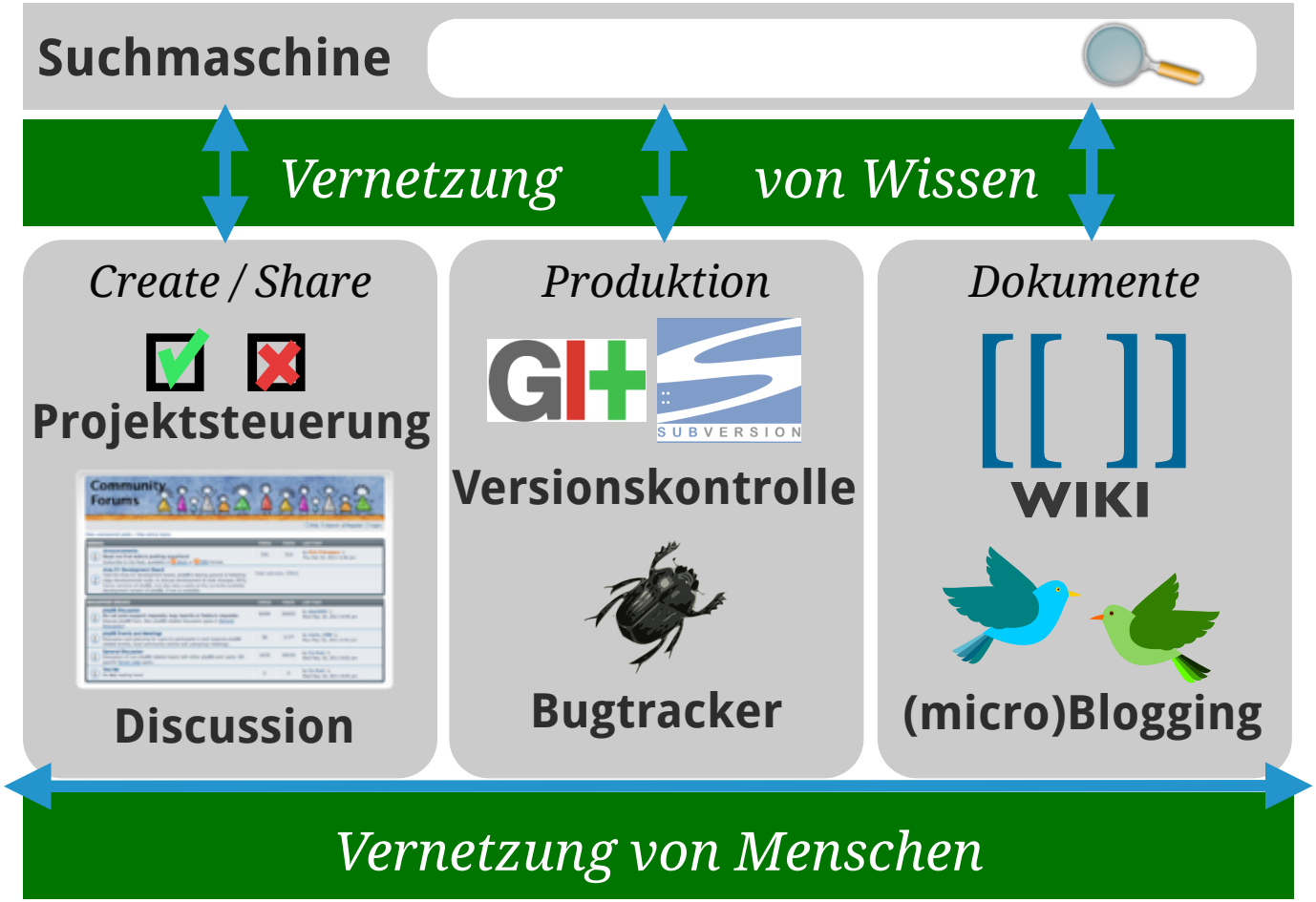
Nonaka, I./Takeuchi, H.: Die Organisation des Wissens. Campus Verlag. Frankfurt 1997

Wissensmanagement Merkmale nach Probst/Raub/Romhardt



Probst, G., Raub, St., Romhardt, K.: Wissen managen - Wie Unternehmen ihre wertvollste Ressource nutzen. Gabler. 2006. Seite 25 ff.

Wissensmanagement - Suchmaschine zum Wissenserwerb



Vorteile im Unternehmen:

- Information ist unabhängig vom Ablagesystem sichtbar
- Gemeinsame Navigation unterstützt Vernetzung
- Nutzer wählen das optimale System zur Ablage

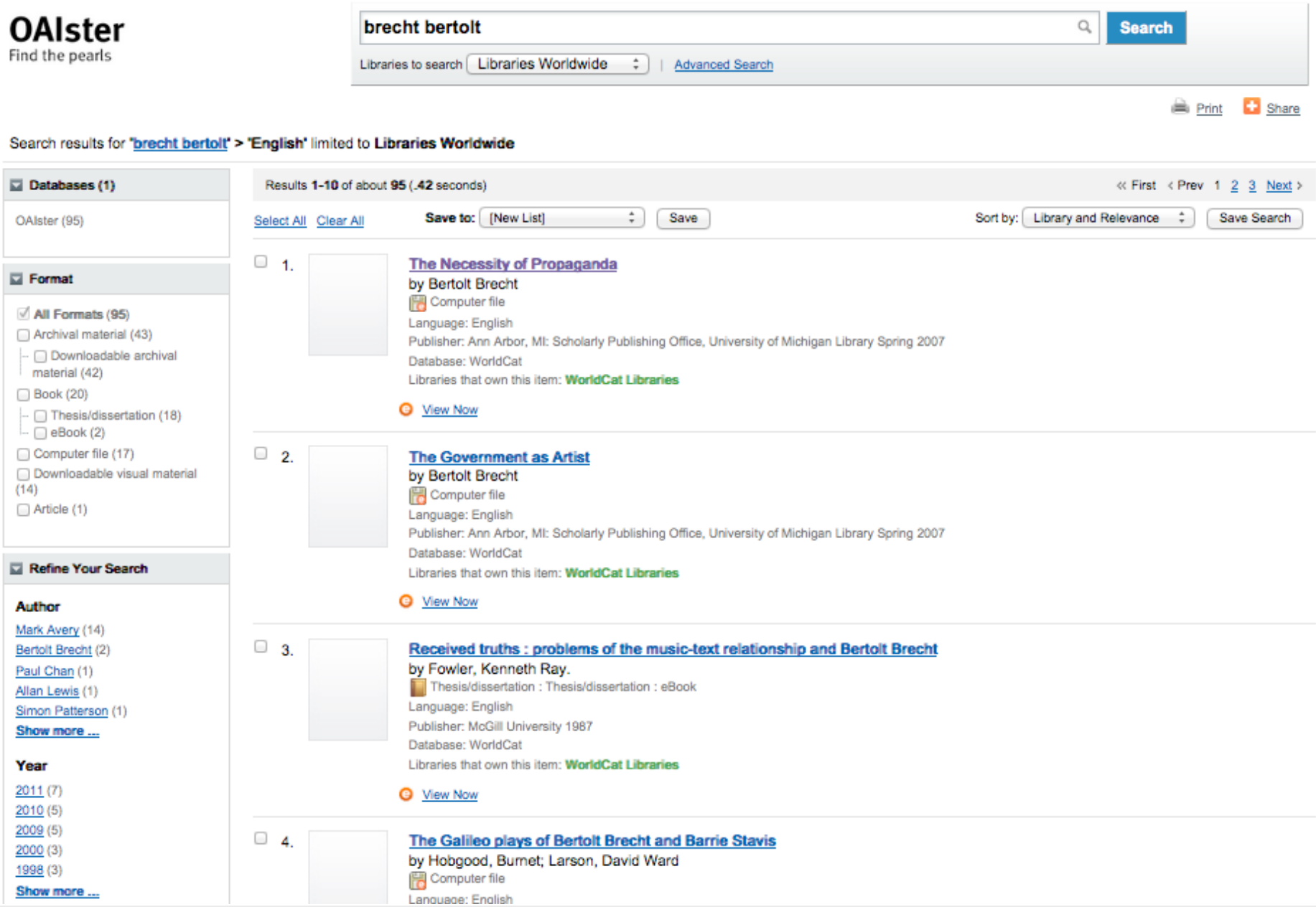
Technologische Vernetzung

„wie setze ich Technik ein um Wissen zu generieren?“

Soziotechnische Vernetzung

„wie gehen Menschen mit Technik um?“

What is Information Retrieval? - IR in Libraries



http://oaister.worldcat.org

What is Information Retrieval? - IR in Museums

LIDO

Lightweight Information Describing Objects

Descriptive and administrative elements of a LIDO record

-Object Classifications-

Object / Work Type *(mandatory)*

Classification

-Object Identifications-

Title / Name *(mandatory)*

Inscriptions

Repository / Location

State / Edition

Object Description

Measurements

-Events-

Event Set

-Relations-

Subject Set

Related Works

-Administrative Metadata-

Rights

Record *(mandatory)*

Resource

Events in LIDO

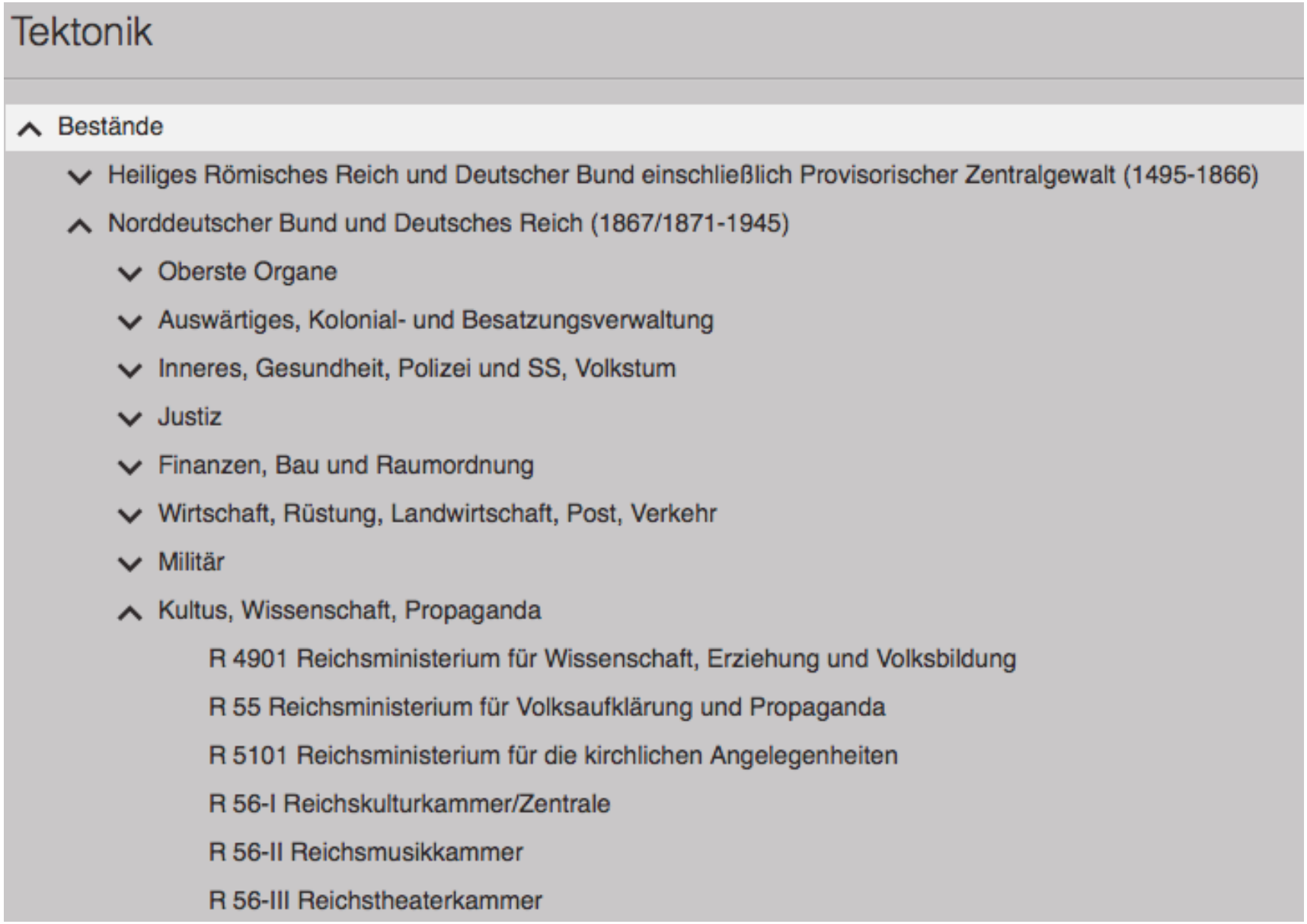
Event
- Event Identifier
- Event Type
- Role in Event
- Event Name
- Event Actor
- Culture
- Event Date
- Event Place
- Event Method
- Materials / Technique
- Thing Present
- Event Related
- Event Description

Content / Subject in LIDO

Subject
- Extent Subject
- Subject Concept
- Subject Actor
- Subject Date
- Subject Place
- Subject Event
- Subject Object

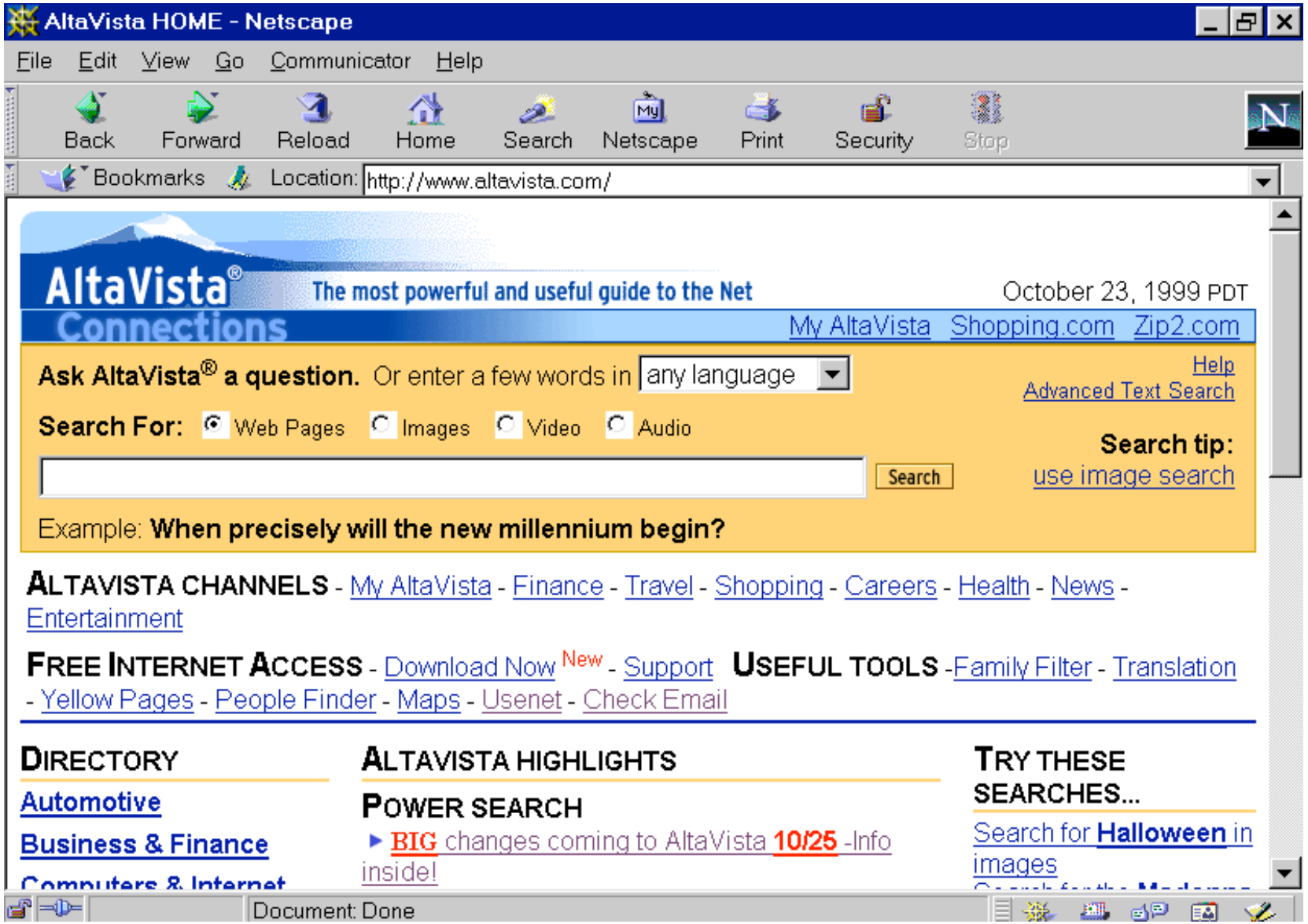
http://www.lido-schema.org/documents/LIDO-Handout.pdf

What is Information Retrieval? - IR in Archives



https://invenio.bundesarchiv.de/basys2-invenio/main.xhtml

What is Information Retrieval? - IR in the WWW

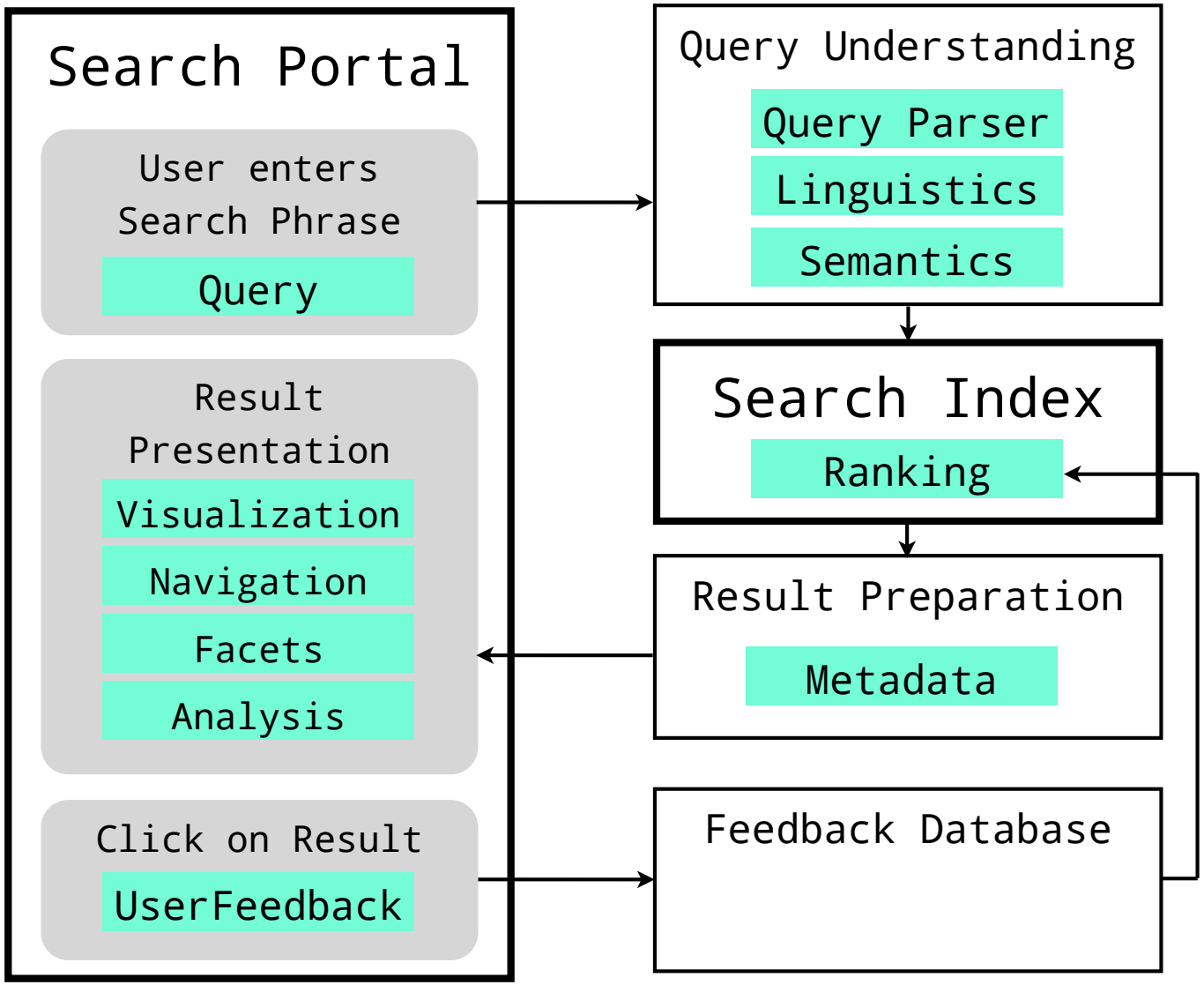


http://en.wikipedia.org/wiki/File:Altavista-1999.png

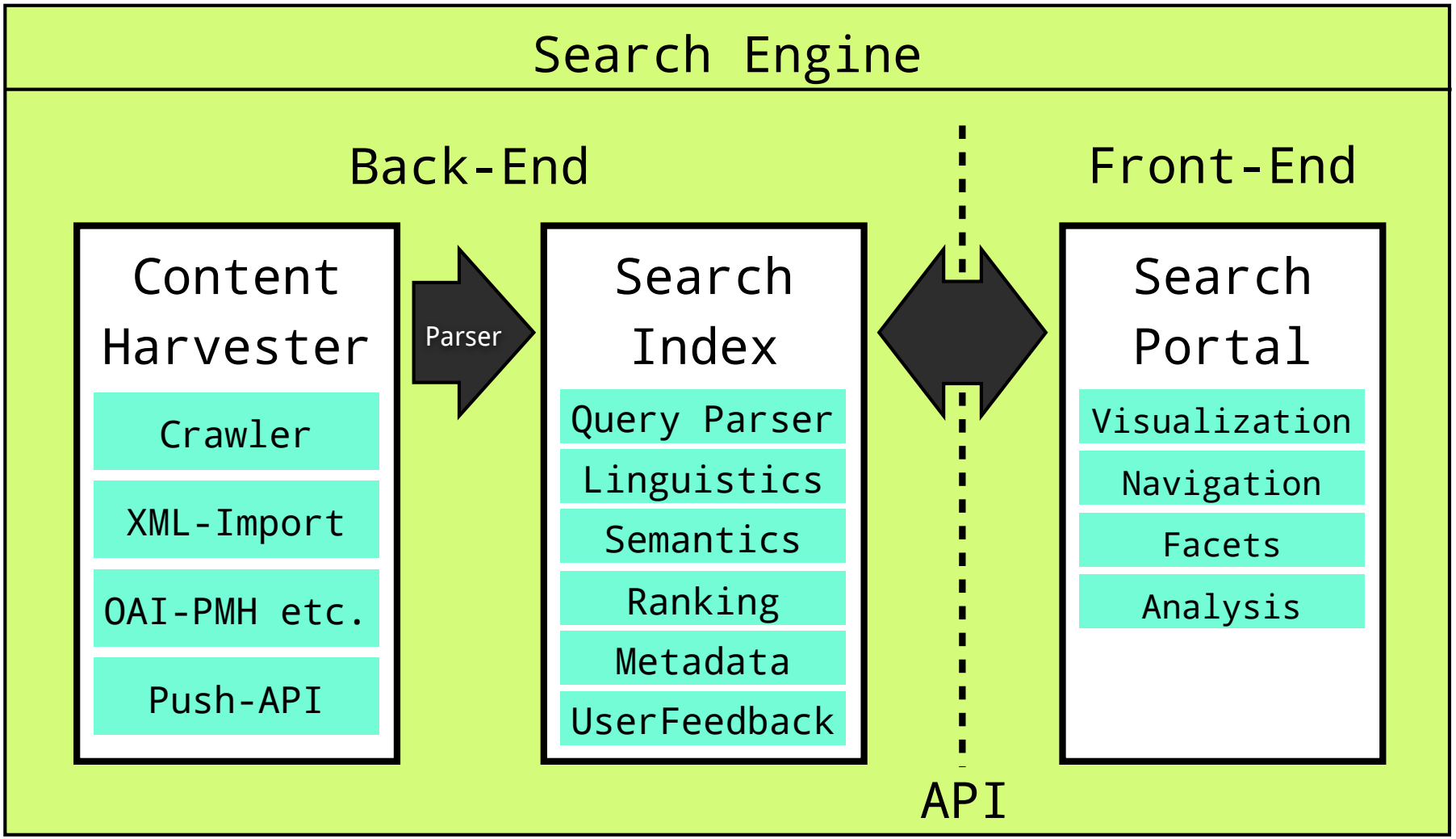
What is Information Retrieval? vs. Data Retrieval

	Information Retrieval i.e. Search Engines	**Data Retrieval** i.e. Databases
Purpose	retrieve information based on query keywords	retrieve records based on well-formed terms
Query Language	human language, limited formal syntax, i.e. search modifier	formal language, logic, regular expressions
Expectation	information about the content, fuzzyness accepted	exact evaluation, single error means failure
Order	a result ranking based on user expectation, implicit order	none or explicit order may be given in query
Users	Humans (sometimes Software)	Machines / Software
Semantics	enrichment about result domain using facets by default	result grouping possible on demand

Information Retrieval Software - Process

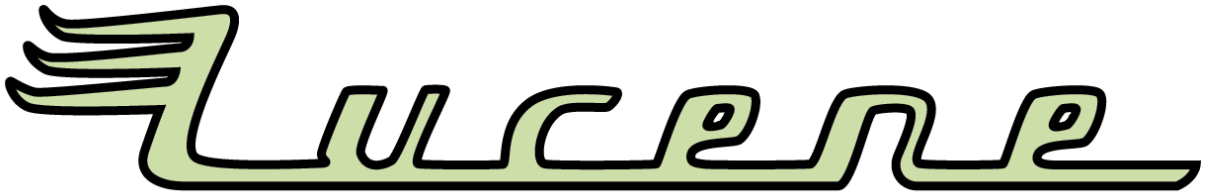


Information Retrieval Software - Concept



Information Retrieval Software - Free and Open Source

Library:



with XML/JSON API:

with API+ Harvester:

loklak.org

Information Retrieval Software - Free and Open Source

Christian Middleton,
Ricardo Baeza-Yates:

A Comparison of Open Source Search Engines

Search Engine	Update	Version	Observation
ASPSeek	2002	N/A	The project is paralyzed.
BBDBot	2002	N/A	Last update was on 2002, but since then it has not have any activity.
Datapark	13/03/2006	4.38	
ebhath	N/A	N/A	No existing website.
Eureka	N/A	N/A	Website is not working.
ht://Dig	16/06/2004	3.2.0b6	
Indri	01/2007	2.4	
ISearch	02/11/2000	1.75	According to the website, "the software is not actively maintained, although it is available for download".
IXE	2007	1.5	
Lucene	02/03/2006	1.9.1	
Managing Gigabytes	01/08/1999	1.2.1	
MG4J	03/10/2005	1.0.1	
mnoGoSearch	15/03/2006	3.2.38	
MPS Inform. Server	01/09/2000	6.0	
Namazu	12/03/2006	2.0.16	
Nutch	31/03/2006	0.7.2	Subproject of the Lucene project.
Omega	08/04/2006	0.9.5	Omega is an application that uses the Xapian library.
OmniFind IBM Yahoo!	2006/12/07	8.4.0	
OpenFTS	05/04/2005	0.39	
PLWeb	16/03/1999	3.0.4	On 2000, AOL Search published a letter stating that the code will no longer be available.
SWISH-E	17/12/2004	2.4.3	
SWISH++	14/03/2006	6.1.4	
Terrier	17/03/2005	1.0.2	
WAIS & freeWAIS	N/A	N/A	The software is outdated.
WebGlimpse	01/04/2006	4.18.5	Uses Glimpse as the indexer.
XML Query Engine	02/04/2005	0.69	It is an XML search engine.
Zebra	23/02/2006	1.3.34	It is an XML search engine.
Zettair	09/2006	0.93	

http://wrg.upf.edu/WRG/dctos/Middleton-Baeza.pdf

Information Retrieval Software - Free and Open Source

Christian Middleton,
Ricardo Baeza-Yates:

A Comparison of Open Source Search Engines

Search Engine	Storage[f]	Increm. Index	Results Excerpt	Results Template	Stop words	Filetype[e]	Stemming	Fuzzy Search	Sort[d]	Ranking	Search Type[c]	Indexer Lang.[b]	License[a]
Datapark	2	■	■	■	■	1,2,3	■	■	1,2	■	2	1	4
ht://Dig	1	■	■	■	■	1,2	■	■	1	■	2	1,2	4
Indri	1	■	■	■	■	1,2,3,4	■	■	1,2	■	1,2,3	2	3
IXE	1	■	■	■	■	1,2,3	□	■	1,2	■	1,2,3	2	8
Lucene	1	■	□	□	■	1,2,4	■	■	1	■	1,2,3	3	1
MG4J	1	■	■	■	■	1,2	■	□	1	■	1,2,3	3	6
mnoGoSearch	2	■	■	■	■	1,2	■	■	1	■	2	1	4
Namazu	1	■	■	■	□	1,2	□	□	1,2	■	1,2,3	1	4
Omega	1	■	□	■	■	1,2,4,5	■	□	1	■	1,2,3	2	4
OmniFind	1	■	■	■	■	1,2,3,4,5	■	■	1	■	1,2,3	3	5
OpenFTS	2	■	□	□	■	1,2	■	■	1	■	1,2	4	4
SWISH-E	1	■	□	□	■	1,2,3	■	■	1,2	■	1,2,3	1	4
SWISH++	1	■	□	□	■	1,2	■	□	1	■	1,2,3	2	4
Terrier	1	□	□	□	■	1,2,3,4,5	■	■	1	■	1,2,3	3	7
WebGlimpse	1	■	■[g]	■[g]	□	1,2	□	■	1[e]	■	1,2,3	1	8,9
XMLSearch	1	■	□	□	■	3	□	■	3	□	1,2,3	2	8
Zettair	1	■	■	□	■	1,2	■	□	1	■	1,2,3	1	2

[a] 1:Apache,2:BSD,3:CMU,4:GPL,5:IBM,6:LGPL,7:MPL,8:Comm,9:Free
[b] 1:C, 2:C++, 3:Java, 4:Perl, 5:PHP, 6:Tcl
[c] 1:phrase, 2:boolean, 3:wild card.
[d] 1:ranking, 2:date, 3:none.
[e] 1:HTML, 2:plain text, 3:XML, 4:PDF, 5:PS.
[f] 1:file, 2:database.
[g] Commercial version only.

■ Available
□ Not Available

http://wrg.upf.edu/WRG/dctos/Middleton-Baeza.pdf

Information Retrieval Software - Commercial

on Hardware:

Software:

~~fast ESP™~~

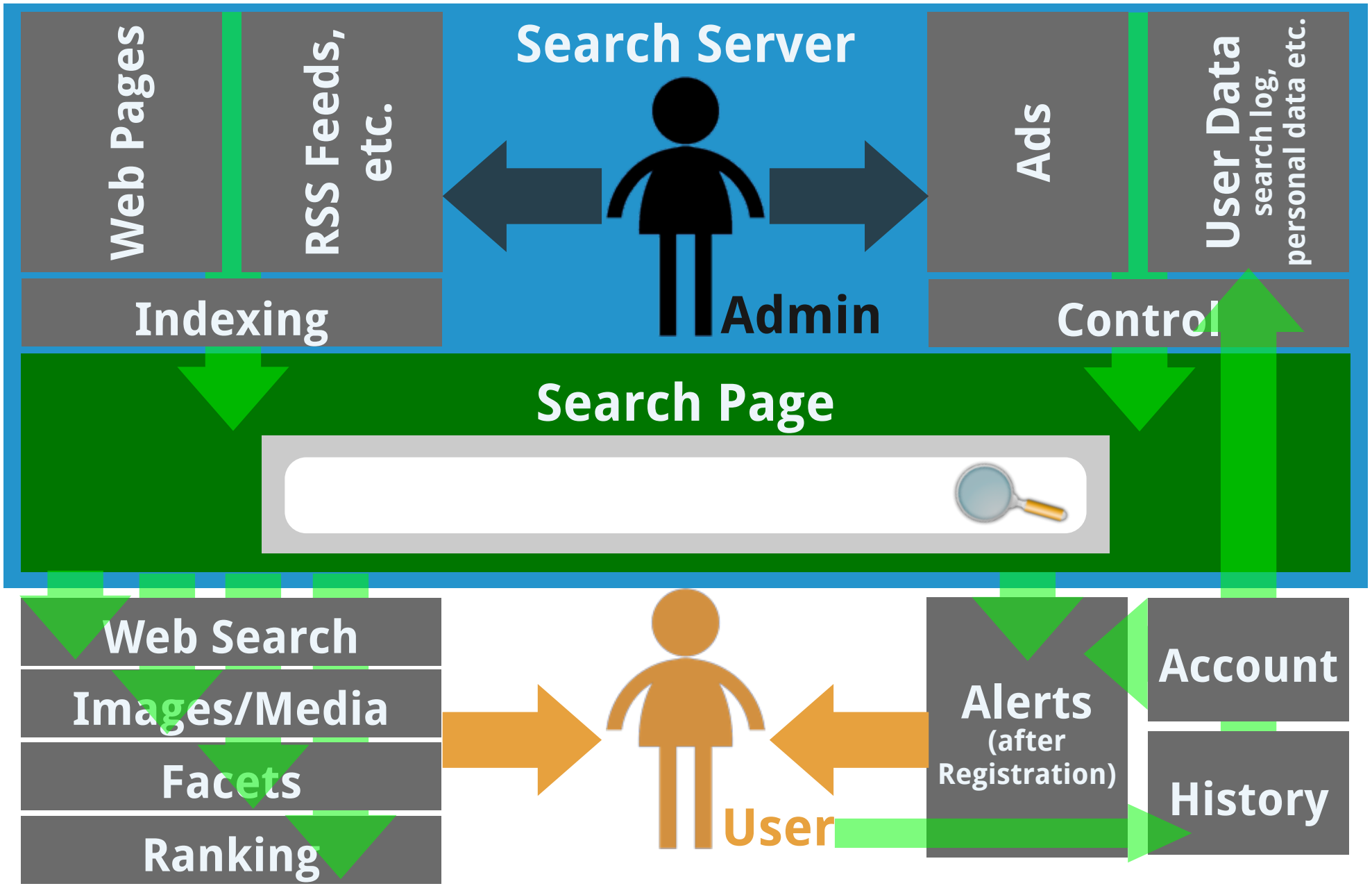
Aquired by Microsoft and removed from market

Software:

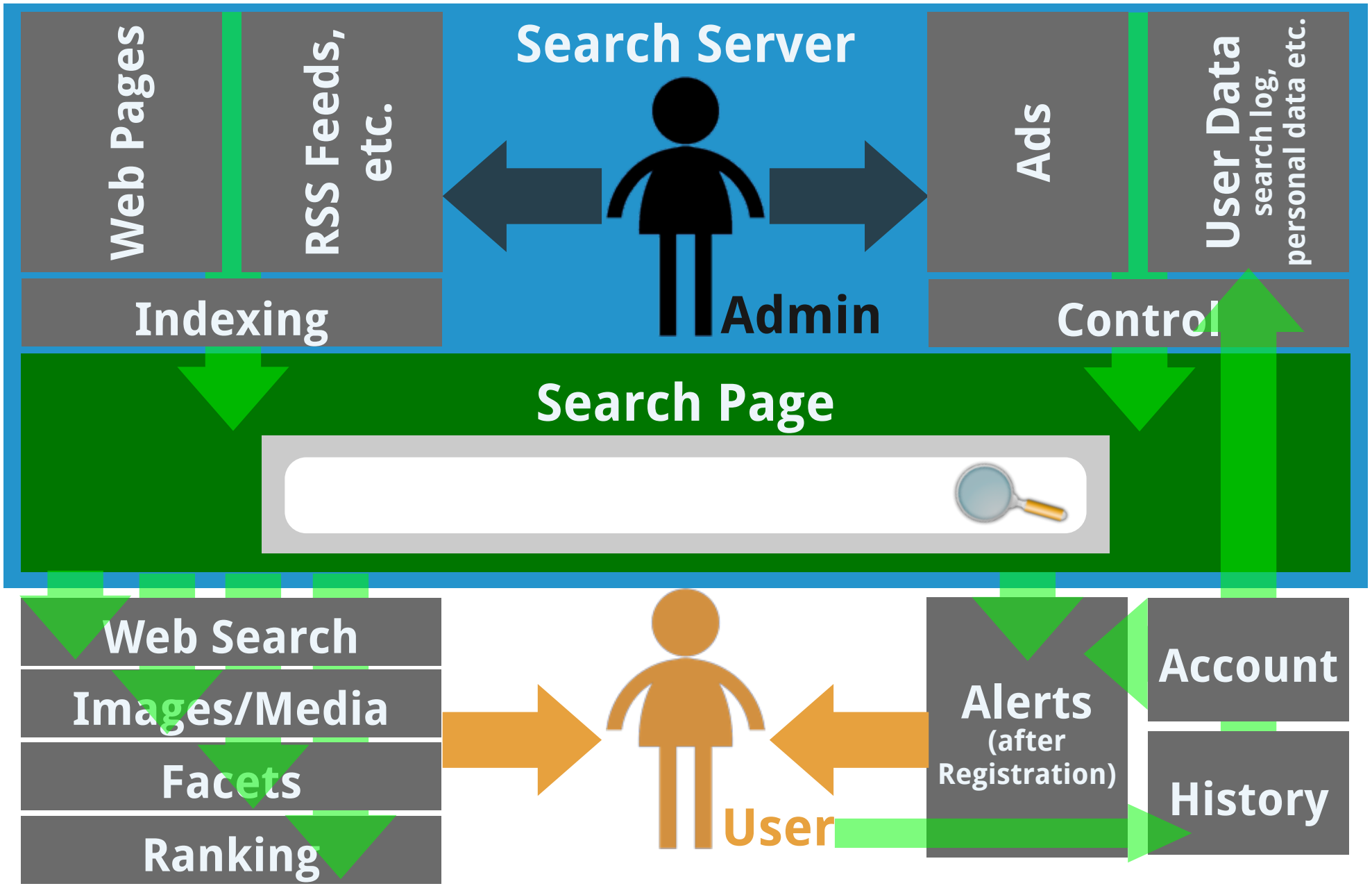
Autonomy®

an HP company

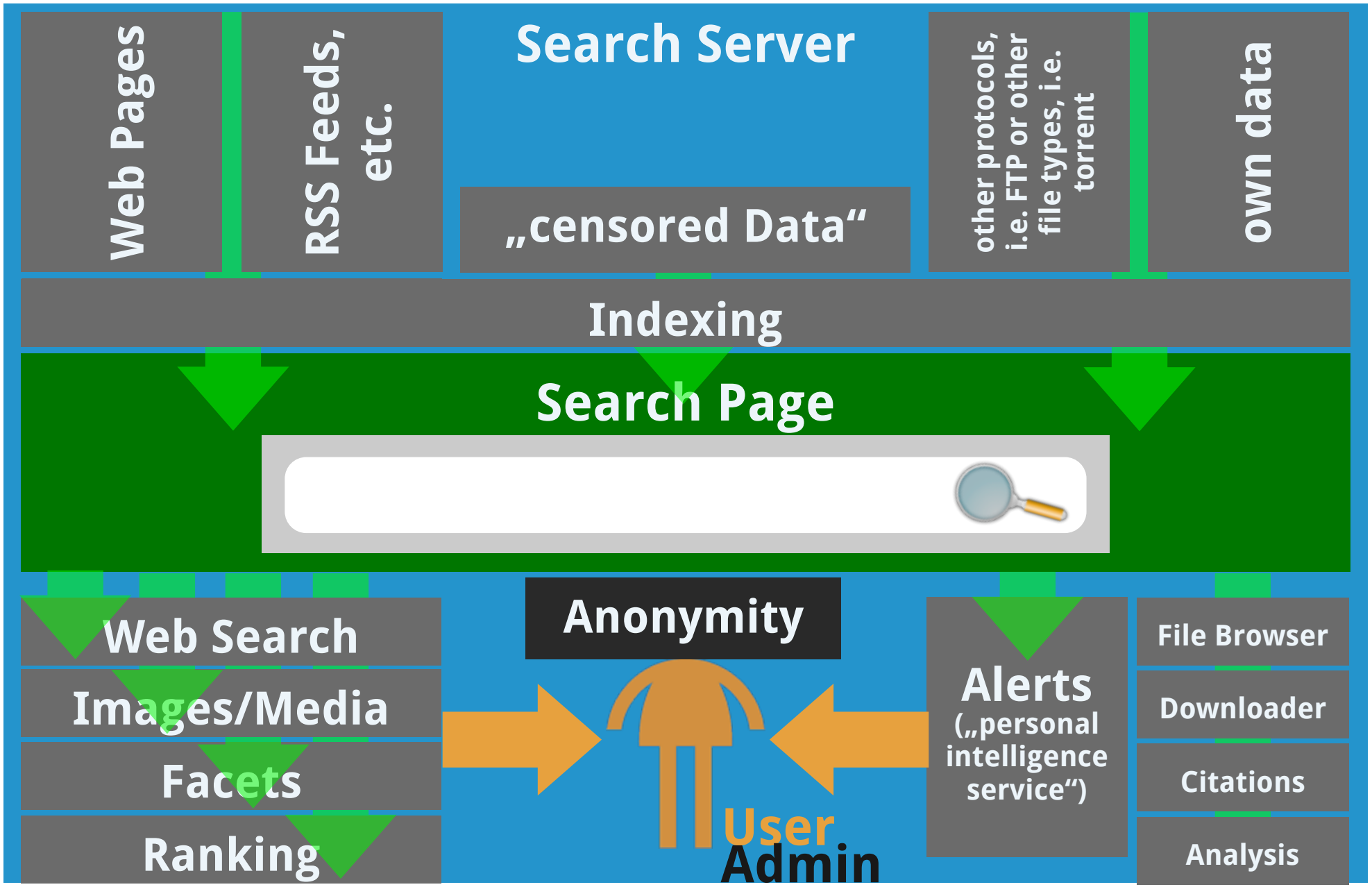
Search Portal Components - Services



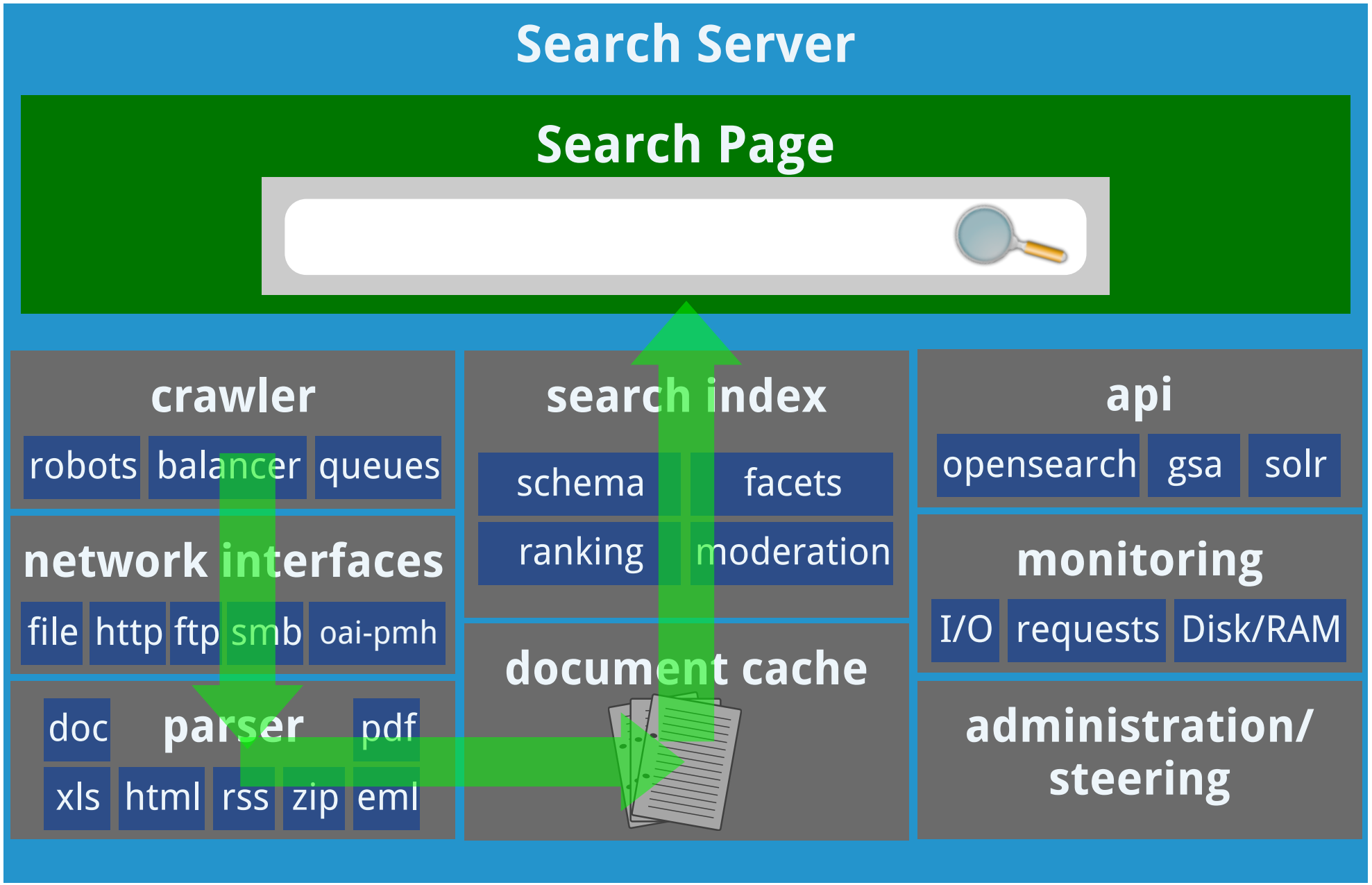
Search Portal Components - Services Re-Design



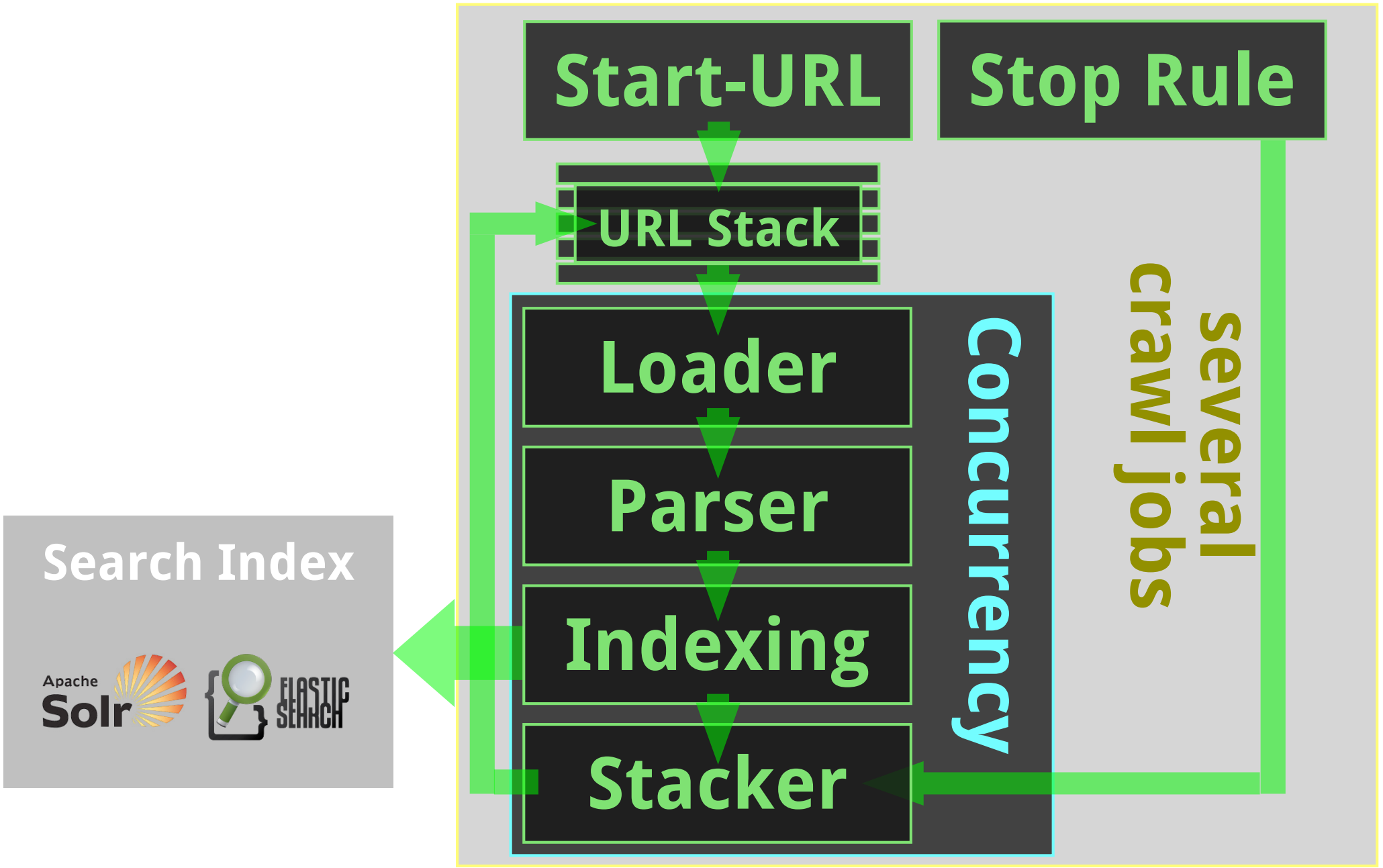
Search Portal Components - Services Re-Design



Search Portal Components - Modules



Crawler Architecture - Modules



Einteilung der Themen in Kategorien

- **Theorie**: Modelle
- **Methoden**: Algorithmen
- **Soziales**: Einflussnahme auf Menschen
- **Forschung**: Systematische Suche nach Erkenntnissen
- **Praxis**: Implementierung von Lösungen

Hausarbeitsthemen

Anforderungen

- **Referat**: Sie referieren über Ihr Thema; ca 20-30 Min. + 15 Min. Fragen + Diskussion
- **Hausarbeit**: schriftliche Ausformulierung Ihres Themas

allgemeine Qualitätsmerkmale

- **Umgang mit Quellen**: machen Sie keine reine Reproduktion; vertiefen Sie das Thema, stellen Sie Querbezüge zu anderen Autoren her, vergleichen Sie Methodiken und Ergebnisse: wenden Sie Wissensmanagementmethodiken an um Ihr eigenes Wissen zu vertiefen, um es mit externen Wissen zu vereinen.
- **Umgang mit veralteten Informationen**: versuchen Sie Alternativen zu nicht mehr existierenden Quellen zu finden (das kommt bei Weblinks häufig vor)

Gestaltung des Referats

- **Stil**: tragen Sie flüssig und verständlich vor (Üben hilft!), lesen Sie nicht alles ab!
- **Folien**: versuchen Sie Bulletpoints zu vermeiden, Grafiken sind gut
- **Inhalt der Folien**: sehen Sie hierzu die Hinweise zur Gestaltung der Hausarbeit
- **Moderation**: gehen Sie auf Fragen ein? Fordern Sie das Publikum zur Interaktion auf?
- **Korrektheit**: reden Sie nicht von Dingen die Sie selbst nicht verstanden haben.
- **Zeitmanagement**: bleiben Sie im Zeitrahmen und schöpfen Sie die Zeit voll aus!
- **Abgabe der Vortragsfolien**: elektronisch spätestens unmittelbar vor der Präsentation.

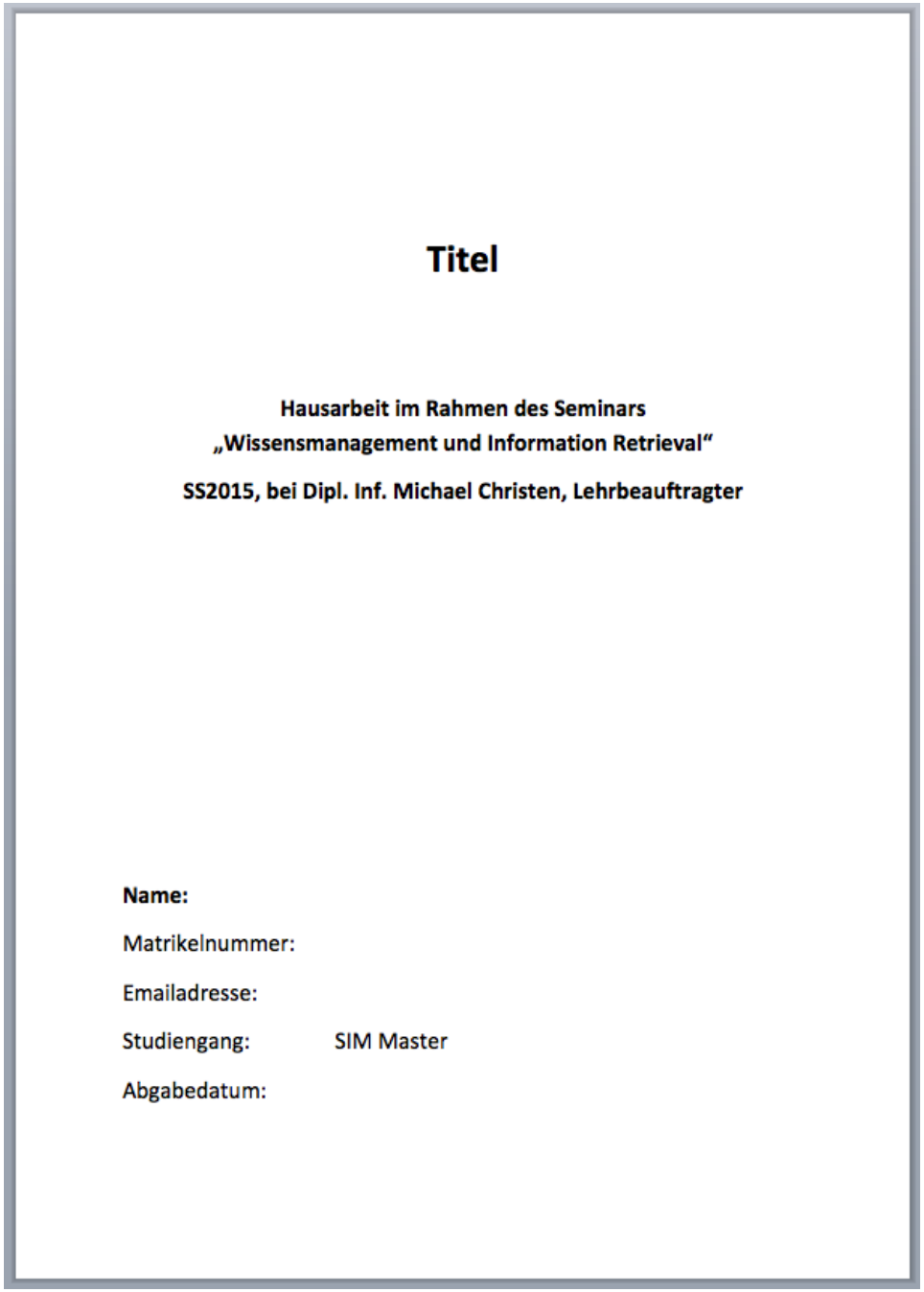
Gestaltung der Hausarbeit

- **Umfang und Vollständigkeit**: 8-10 Seiten plus Titelblatt, Inhalt, Literaturverzeichnis
- **Gliederung**: sinnvoll und plausibel, angemessenes Gewicht auf Schwerpunkte
- **Ausformulierung**: erreichen Sie eine inhaltliche Tiefe und bringen Sie eigene Leistungen ein. Machen Sie nicht einfach nur Copy-Paste aus den Folien. Wenn Sie in den Folien viele Grafiken und wenig Bulletpoints verwendet haben, sollten Sie selbstverständlich alle Grafik unverändert übernehmen.
- **Zitate**: verwenden Sie wissenschaftliche Quellen und nicht die Wikipedia. Machen Sie vollständige Literaturangaben (Name des Autors, Titel, Buch/Zeitschrift, Ort/Land, Verlag, Datum). Verwenden Sie Internetquellen wenn es sich um eine Primärquelle handelt und geben Sie die URL an.
- **Abgabe**: auf Papier mit Unterschrift zu einer Erklärung dass Sie die Arbeit selbst und nur mit angegebenen Hilfsmitteln angefertigt haben

Hausarbeitsthemen

Gestaltung der Hausarbeit

- Verwenden Sie ein Titelblatt-Template



Titel

Hausarbeit im Rahmen des Seminars
„Wissensmanagement und Information Retrieval“

SS2015, bei Dipl. Inf. Michael Christen, Lehrbeauftragter

Name:

Matrikelnummer:

Emailadresse:

Studiengang: SIM Master

Abgabedatum:

Information Retrieval Modelle

- Definition des IR-Modell Quadrupels
- Beispiele
- Bedeutung
- Literatur: **Baeza-Yates, Ricardo; Ribeiro-Neto, Berthier: Modern Information Retrieval. ACM Press Series / Addison Wesley, New York, USA, 2011.** Kapitel 3.1
- Ergänzen Sie mit weiteren Quellen

Das Vektorraum-Modell zur Relevanzberechnung von Dokumenten

- Baut auf boolesches Modell auf (Vortrag wird vorher gehalten)
- Abgrenzung zum Probabilistischem Modell
- Literatur: **Baeza-Yates, Ricardo; Ribeiro-Neto, Berthier: Modern Information Retrieval. ACM Press Series / Addison Wesley, New York, USA, 2011.** Kapitel 3.2.6
- Ergänzen Sie mit weiteren Quellen

Das Probalistische Modell zur Relevanzberechnung von Dokumenten

- Abgrenzung zum Vektorraum Modell
- Literatur: **Baeza-Yates, Ricardo; Ribeiro-Neto, Berthier: Modern Information Retrieval. ACM Press Series / Addison Wesley, New York, USA, 2011.** Kapitel 3.2.7
- Ergänzen Sie mit weiteren Quellen

Suche im Web: Das Pagerank-Verfahren von Google

- Theorie und Bedeutung des Verfahrens
- Was gab es vor Google und warum war das Pagerank-Verfahren so viel besser?
- Führen Sie eine Ranking-Berechnung vor
- Benutzen Sie die Original-Publikation „The PageRank Citation Ranking: Bringing Order to the Web“ von Larry Page, 29.1.1998, Stanford University; http://ilpubs.stanford.edu:8090/422/1/1999-66.pdf

Suche im Web: Das HITS-Verfahren von Kleinberg

- Theorie und Bedeutung des Verfahrens
- Abgrenzung zu Page Rank
- Führen Sie eine Ranking-Berechnung vor
- Literatur: **Baeza-Yates, Ricardo; Ribeiro-Neto, Berthier: Modern Information Retrieval. ACM Press Series / Addison Wesley, New York, USA, 2011.** Kapitel 11.5.2

Evaluierung von Suchmaschinen

- Erklären Sie die Begriffe Precision und Recall
- Führen Sie Beispiele vor
- Betrachten Sie diese Evaluierungsmethoden kritisch: wann sind sie sinnvoll anwendbar, wann nicht?
- Literatur: **Dirk Lewandowski, Handbuch der Internet-Suchmaschinen 2, ab Seite 203**

Anfragetypisierung nach Broder

- Welche Kategorien gibt es und was bedeuten sie?
- Literatur: Broder, Andrei, 2002, A taxonomy of web search, ACM Sigir forum, 36(2), 3-10; http://www.researchgate.net/publication/220466848_A_taxonomy_of_web_search
- Evaluieren Sie die Anfragetypen im Hinblick auf die Messbarkeit von Zufriedenheit der Suchmaschinenbenutzer
- Literatur: Krah, Müller-Terpitz, Suchmaschinen, Passauer Schriften zur interdisziplinären Medienforschung, 2013, Uni Passau; ab Seite 35: Dirk Lewandowski: Wie läßt sich die Zufriedenheit der Suchmaschinennutzer mit ihren Suchergebnisen erklären?

Informationsstatistische und informationslinguistische Verfahren für Information Retrieval

- Wie misst man Popularität?
- kann man Texte auf signifikante Inhalte reduzieren?
- was sagt Klickverhalten über die Popularität aus?
- Annäherung des Vokabulars der Texte an die Terminologie des Rechercheurs
- Stemming, Synonyme, Akronyme, Phrasen
- Rechtschreibfehler ausgleichen
- Literatur: http://www.bui.haw-hamburg.de/fileadmin/user_upload/lewandowski/doc/Web_Information_Retrieval_Buch.pdf

Angriffsmethoden auf Ranking-Verfahren (aka ‚Black-Hat SEO')

- Differenzieren Sie zwischen White-Hat und Black-Hat SEO
- Stellen Sie Ranking-Verfahren und Angriffsmethodik gegenüber (gehen Sie nicht detailliert auf das Ranking-Verfahren ein, nur auf den Angriff darauf)
- Wie hat Google in der Vergangenheit auf typische Angriffsmuster reagiert? Stichworte: Doorway-Pages, User-Agent, robots.txt, Link-Farm, Google Penalty / reconsideration, Google Quality Guidelines

Techniken zur Sicherung der Privatsphäre beim Suchen

- Informieren Sie sich über den AOL Leak
- User Identification: informieren Sie sich über die http-header Daten welche Sie bei jeder Anfrage an einen Webserver versenden. Verstehen Sie was Cookies und Session IDs sind.
- Spuren im Web: informieren Sie sich über die Funktionsweise von ‚referer' Referrern und die Bedeutung im Hinblick auf Werbung im Netz
- Sie sollten entdecken, dass die Aktivitäten eines Users leicht getrackt werden kann.
- Verbinden Sie alle Themen und schlagen Sie Sicherungsmaßnahmen im Hinblick auf die entdeckten Tracking-Methoden vor.

Enterprise Search - Suche in Unternehmen

- Skizzieren Sie die Einsatzmöglichkeiten einer unternehmens-Suchmaschine
- Betrachten Sie die bekannten Ranking-Methoden und nennen Sie vor- und Nachteile dieser Methoden in der Unternehmensinternen Suche
- Schlagen sie einen geeigneten Ranking-Algorithmus vor und begründen Sie das
- Nennen Sie Harvesting-Methoden im Intranet
- Sie sollten wissen was für Datenprotokolle in Intranets üblich sind (http, https, ftp, smb, file-Ablage ‚Laufwerk z:')
- Gehen Sie auf die Problematik von Lese- und Schreibrechten ein

Kritische Betrachtung zentraler Suchmaschinen

- Sie kennen bereits einige Funktionskomponenten von Suchmaschinen. Die Arbeitsweise dieser Komponenten läßt sich parametrisieren und modifizieren. Sie können bestimmte Ergebnisse der Suchmaschine dadurch beeinflussen. Was kann das Resultat dieser Beeinflussung sein?
- Stichworte: Ranking, Crawling, Aktualität, Zugänglichkeit der Quellen
- Betrachten Sie soziale und politische Effekte
- Betrachten Sie technische und organisatorische Effekte
- Erkennen Sie die Vor- und Nachteile bezüglich einer dezentralen Architektur und stellen Sie diese gegenüber.

Rechtliche Aspekte von Suchmaschinen

- Betrachte Sie das Grundrecht auf Informationsfreiheit (nicht das „Informationsfreiheitsgesetzt“!) im Hinblick auf rechtliche Einschränkungen wie Marken- Urheber- und Wettbewerbsrecht ab.
- Finden Sie Beispiele zu entsprechenden Urteilen zum Thema
- Persönlichkeitsrecht („Recht auf Vergessen“), Leistungsschutzrecht etc.
- Beantworten Sie selbstgestellte Fragen zu ‚darf die Suchmaschine das‘? (Hinweis: Snippets, Bildersuche mit Vorschau, Zugriff auf Crawl-Cache, Historisierung)

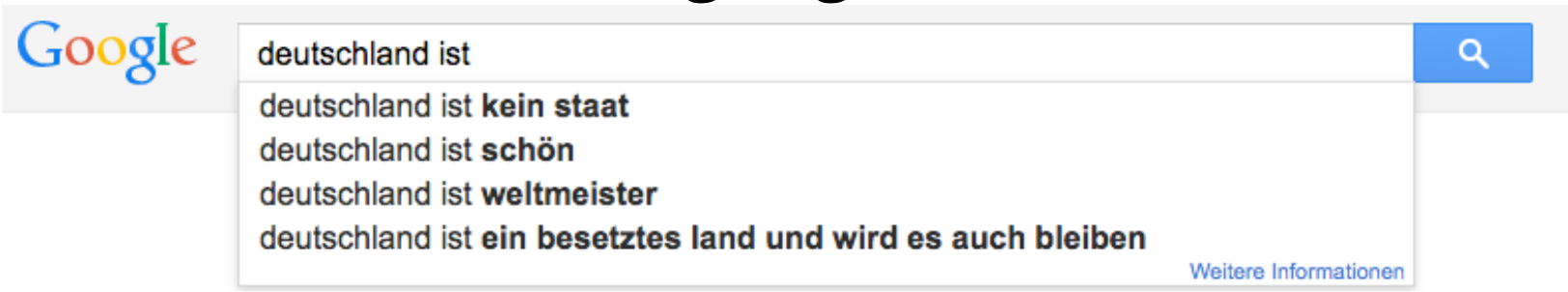
Gesellschaftliche und/oder Wirtschaftliche Auswirkungen von Suchmaschinen durch Vorgabe von Relevanz

- Arbeiten Sie dieses Thema nur für das gesellschaftliche oder wirtschaftliche Themengebiet aus, nicht für beide. Bonus: doch beide.
- Wir kennen verschiedene Relevanzmethodiken, jede dieser Methodiken sind parametrisierbar und an soziale und rechtliche Vorgaben anpassbar. Die Frage ist: wie können Gemeinschaften und Gesellschaften (bei öffentlicher Suche) oder Mitarbeiter (bei Enterprise Search) beeinflusst werden?
- Welches Ergebnis kann erzielt werden, welches könnte angesteuert werden.

Gefahren bei der Suche im Web und Techniken zur Sicherung der Privatsphäre beim Suchen

- Welche personenbezogenen Informationen können geharvestet werden (denken Sie auch an die Metadaten)
- Welche personenbezogenen Information können von der nutzenden Person (bei der Suche) preisgegeben werden, auch ohne Anmeldung
- Wie kann eine Person (ohne Anmeldung) identifiziert werden?
- Wie kann die gewonnene Personenidentifizierung vom Suchportal genutzt werden? Stichworte: Internet-Marketing, Such-Historie
- log-leaks?
- Evaluieren Sie ein bereitgestelltes Log (nur Keywords, keine Identifikation)

Methodiken zur Erzeugung von Suchwortvorschlägen



- evaluieren Sie die Verhaltensweise von verschiedenen Suchportalen inklusive Google, Facebook Social Graph, WolframAlpha, OPACs, Shoppingportale etc. Finden Sie möglichst viele verschiedenen Beispiele.
- Betrachten Sie Details (ein Wort, mehrere, letztes Zeichen ist ein Whitespace, letzter Term ist unvollständig etc.)
- Sozialer Aspekt: stellen Sie sich die Frage „was bringt der Vorschlag für die suchende Person, welche Effekte sind vorteilhaft für das Suchportal“ (ggf. keine?)
- definieren Sie welche Daten benötigt werden, um das beobachtete Verhalten erzeugen zu können mit Ergebnisaufstellung: wenn folgende Daten vorhanden ist welche Methodik verfügbar?
- Erfinden Sie eine Methodik für Suchwortvorschläge für eine ‚interessante‘ Suchfunktion im Stil von ‚Schenken ohne Denken‘.

Semantische Anreicherung durch Erkennung von Emotionen

- Emotionserkennung ist bei der Kategorisierung von Kurznachrichten sinnvoll; betrachten Sie die Aufgabe unter der Prämisse, dass sie solche Nachrichten auswerten würden.
- Die Erkennung von Texteigenschaften ist durch informationsstatistische Methodiken möglich; in diesem Fall untersuchen wir die Erkennung von Emotionen durch Abgleich mit Vokabularien
- Identifizieren Sie geeignete Vokabularien
- Können Sie eine Emotionsbegriff-Ontologie erstellen?
- Schlagen Sie eine Datenstruktur (Indexfelder) für die Ergebnisse der Emotionserkennung vor
- Wie sieht dann eine Suchfacette zur Selektion entsprechender Texte aus?

Mikroformate: Anwendung für Suchfacetten

- Machen Sie sich mit dem Begriff ‚Linked Open Data' (LOD) vertraut.
- Lernen Sie was Mikroformate (engl. microformats) sind bzw. was rdf und rdfa ist.
- Welche Anwendung haben solche semantic-Web Technologien in der Suchmaschinentechnik?
- Betrachten Sie schema.org (ein von Google vorgeschlagenes LOD Schema) und bewerten Sie dessen Anwendbarkeit für Suchmaschinen.
- Bewerten Sie die Nutzbarkeit: kann ein Black Hat SEO dies angreifen? Wie nutzt ein Suchportalbetreiber schema.org Daten zur Verbesserung des Sucherlebnisses?
- Sind mit schema.org neue Typen von Suchportalen möglich? Stichwort: Themenorientierung

Praxisarbeit: Erstellen Sie eine Suchmaschine für einen Wikipedia-XML-Dump

- Laden Sie einen XML-Dump von http://dumps.wikimedia.org/dewiki/
- Indexieren Sie die Daten des Dumps mit Solr oder elasticsearch (sie müssen ggf. einen XML Parser programmieren...)
- Erstellen Sie eine Suchseite für den Index (elasticsearch und Solr bieten entsprechende Werkzeuge an)
- Hausarbeit besteht aus der Dokumentation aller Schritte, muss durch copy-paste von Kommandos nachvollziehbar sein.
- Pluspunkt: freie Softwarelizenz

Dieses Thema ist ggf. etwas umfangreicher, spricht aber einen praxisnahen Studenten ggf. eher an.

Indexing - Definitions

- *Indexing* is the process to create an *inverted index* from a (set of) document(s).
- *Queries* are the strings which the user sends to a search interface

- A query can be characterized, i.e.:
  - *Single-term* or *multiple-term* queries
  - multiple-term queries may use *boolean operators*; they can have a *default boolean operator*
  - a query may use *search operators* (i.e. quotes, wildcards, number ranges, social tags @ and #, date constraints, site:, link:, linanchor:, allintext:, allintitle:, allinurl:, cache:, define:, filetype:, id:, inanchor:, info:, intext:, intitle:, inurl:, related: and much more)

```
# http://www.googleguide.com/advanced_operators_reference.html
```

Indexing - Ranking Definitions

- *Ranking* is the process to order the search result in a specific way.
- There are many different ranking methods!
- *Relevance* is a general term for different ranking methods; mostly used in the context of non-numeric (like by-date or by-size) ranking methods.

personal/professional experience with these words:

- If the word ‚relevance' is used in the context of the opinion of humans then the ranking of an IR system must order search results in such a way that the order corresponds to the expectation of the person, which means in this context:

  relevance : an expectation of a human, a personal view

  ranking : a technical process (which shall compute relevance)

- This often means that an IR system must apply a ranking which must satisfy many relevancy expectations of the users.

Indexing - Document Definitions

- A *Document* is a ‚bag of words', a set of *index terms* (terms to be indexed)
- an *index term* is a word of a document which represents semantic for the document
- let D = $\{d_0, \ldots d_{m-1}\}$ be the set of m *documents* (the *Corpus*)

  where each $d_{j,\ 0 \le j < m} \in D$ is a document $d_j = \{t_{j,0}, \ldots t_{j,n-1}\}$ and

  where d_j is a set of n *index terms* $t_{i,j\ 0 \le i < n} \in d_j$

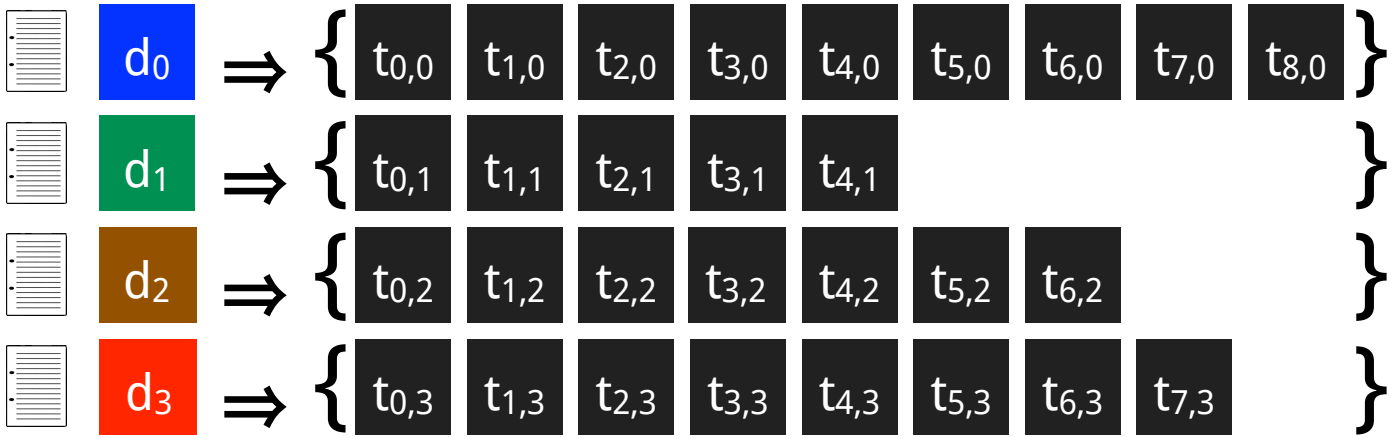
- $\exists\ p_1, p_2, q_1, q_2$ with $0 \le p1 < m,\ 0 \le q1 < n,\ 0 \le p2 < m,\ 0 \le q2 < n$

  in such a way that $t_{p1,q1} = t_{p2,q2}$
  (or $t_{p1,q1} \simeq t_{p2,q2}$ ... see stemming, synonyms, etc.)

- different *index terms* can have more or less *weight* when representing the semantic of the document; rules which describe that weight are described in *ranking methods*

Indexing - Example

Corpus: every document is a ‚bag of words‘

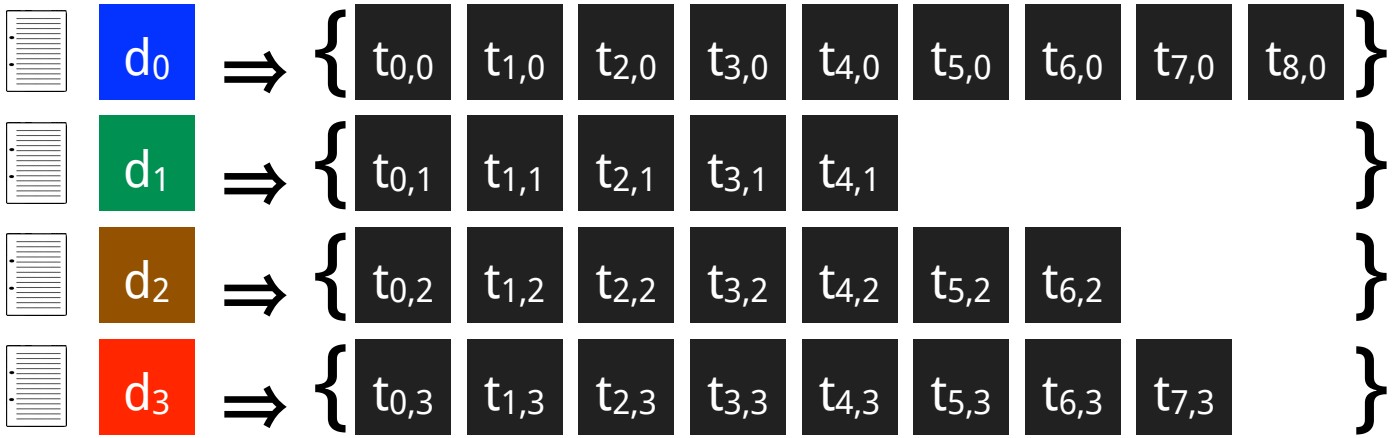


Index: all same words are referenced by a list of doc-ids

- *Indexing* is the process to create an *inverted index* from a (set of) document(s).

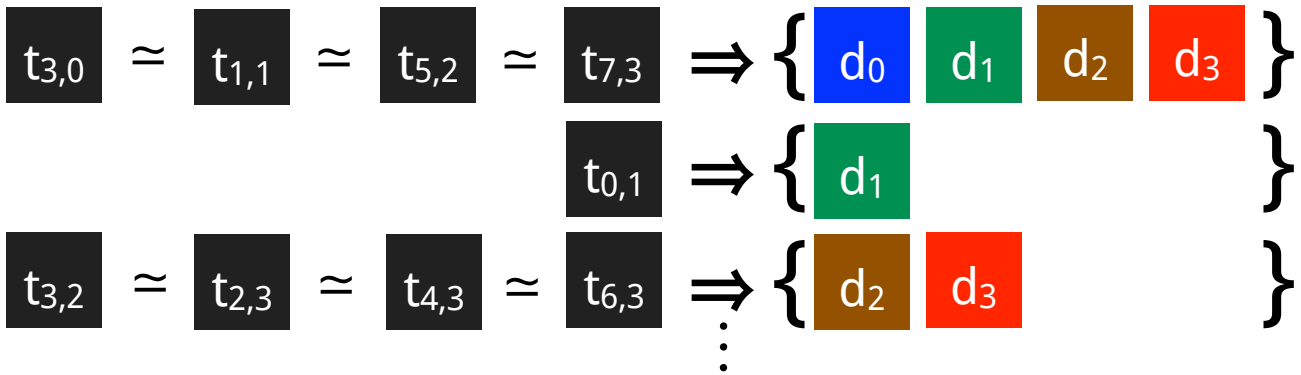
Indexing - Example

Corpus: every document is a ‚bag of words‘



Index: all same words are referenced by a list of doc-ids

- *Indexing* is the process to create an *inverted index* from a (set of) document(s).



Terms are equal ‚≃‘ in the context of stemming rules, synonyms, localization techniques, phonetic rewrites, partial match algorithms, stopword rules, and other techniques

- An *inverted index* consist of *inverted lists* sorted by *document identifier*.

Indexing - Index Compression

- Each *inverted lists* is sorted by the *document identifier*.

- The *document identifier* is either a cardinal number or a string. In both cases the binary representation is prefixed by same bytes for a large sequence of identifiers
- The prefix of same identifier can be noted only once, with a counter of apearances and bitlength
- The actual document identifier is composed with that prefix and a tail-string of the identifier which changes for each document
- The larger the inverted list, the better the compression works

Query Types - Definitions

- A *Single Word Query* retrieves a single inverted list; the document identifier in that list refers to the search results documents.
- A *Multiple Word Query* must be done in the context of an explicit or a default boolean operator:
  - are *all query terms conjunctive* (default AND): the search result is computed using an *intersection* (join) of the document identifiers from the individual inverted lists
  - are *all query terms disjunctive* (default OR): the search result is computed using a *combination* (merge) of the document identifiers from the individual inverted lists
- To express the meaning of the actual (implicit/explicit boolean operators applied) formal query, a *query normal form* is computed.
- One model to represent the logic of queries is the *Boolean Model*

Query Model - Boolean Query Model

- To express a query in the Boolean Model, the *Disjunctive Normal Form* (*DNF*) of the query expression is computed.
- A DNF is a normalization: a disjunction of conjunctive terms:

  $a \wedge (b \vee \neg c) \quad \Rrightarrow_{DNF} \quad (a \wedge b) \vee (a \wedge \neg c)$

- The *query disjunctive normal form* q_{DNF} of a query q is a rewrite as disjunction of *conjunctive components* c(q) which satisfies the conditions of the query:

  $a=1, b=0, c=0 \quad \Rrightarrow \quad c_0(q) = (1, 0, 0)$
  $a=1, b=1, c=0 \quad \Rrightarrow \quad c_1(q) = (1, 1, 0)$
  $a=1, b=1, c=1 \quad \Rrightarrow \quad c_2(q) = (1, 1, 1)$

- The q_{DNF} of $q = a \wedge (b \vee \neg c)$ is $q_{DNF} = (1, 0, 0) \vee (1, 1, 0) \vee (1, 1, 1)$
- The search result for q is a combination (merge) of
  - the inverted list for a
  - the intersected (joined) inverted lists for a and b
  - the intersected (joined) inverted lists for a, b and c

} *When they are merged, their ranking is added! → inverted list for a ranks higher than all other*

Baeza-Yates, Ricardo; Ribeiro-Neto, Berthier: Modern Information Retrieval. ACM Press Series / Addison Wesley, New York, USA, 2011. Chapter: 3.2.2

Ranking with the Term Frequency relevancy model

- Each term $t_{i,j}$ within a document d_j has a weight $w_{i,j} > 0$
- Consider p_1, p_2, q_1, q_2 with $0 \leq p1 < m$, $0 \leq q1 < n$, $0 \leq p2 < m$, $0 \leq q2 < n$ in such a way that $t_{p1,q1} = t_{p2,q2}$
- This does not mean that $w_{p1,q1} = w_{p2,q2}$ when $t_{p1,q1} = t_{p2,q2}$ (that depends also on the ranking model)
- The ‚raw' *Term Frequency* $f(t_{i,j}, d_j)$ is the number of occurrences of the term $t_{i,j}$ in d_j
- A simple way to write the term frequecy is, to just define $f_{i,j} = f(t_{i,j}, d_j)$
- There are different variants to define the Term Frequency $tf_{i,j}$, the most simple form just defines $tf_{i,j} = f_{i,j}$ (Luhn Assumption)
- A popular variant of the Term Frequency is the *log normalization*:

$$tf_{i,j} = \begin{cases} 1 + \log f_{i,j} & \text{if } f_{i,j} > 0 \\ 0 & \text{otherwise} \end{cases}$$

Baeza-Yates, Ricardo; Ribeiro-Neto, Berthier: Modern Information Retrieval. ACM Press Series / Addison Wesley, New York, USA, 2011. Chapter: 3.2.4

Ranking with the Term Frequency relevancy model

- The *Inverse Document Frequency* is defined using attributes which are different for each query:
  - let N be the number of documents in the corpus.
  - let n_i be the number of documents containing the search term $t_{i,j}$

$$IDF_i = \log \frac{N}{n_i}$$

- The weight $w_{i,j}$ for a term $t_{i,j}$ can be defined using the TF*IDF formula:

$$w_{i,j} = tf_{i,j} * IDF_i$$

$$= \begin{cases} (1 + \log f_{i,j}) * \log \frac{N}{n_i} & \text{if } f_{i,j} > 0 \\ 0 & \text{otherwise} \end{cases}$$

- It is popular to use $\log_2$ for log (can be computed using bit shift)

```
Baeza-Yates, Ricardo; Ribeiro-Neto, Berthier: Modern Information Retrieval. ACM
Press Series / Addison Wesley, New York, USA, 2011. Chapter: 3.2.4
```

Ranking Term Frequency in Solr / lucene

- The Ranking of a document for a query q is computed with a *score*
- The *lucene* scoring method uses other variants of tf and IDF:

$tf_{i,j} = f_{i,j}^{0.5}$

$IDF_i = 1 + \log \frac{N}{1 + n_i}$

$coord(q,d_j)$ = number of terms in q appear in d_j

$queryNorm(q,d_j)$ = normalizing factor between queries

$boost_i$ = a boost factor on the term t_i

$norm_{i,j}$ = combined: size of d_j, document- and field-boosts

$$score_j(q) = coord(q,d_j) * queryNorm(q,d_j) * \sum_{\forall t_i \simeq q_i \text{ in} \in q} (tf_{i,j} * (IDF_i)^2 * boost_i * norm_{i,j})$$

http://lucene.apache.org/core/5_1_0/core/org/apache/lucene/search/similarities/TFIDFSimilarity.html

Ranking configuration with Boosts

- Ranking models can be modified with *Field Boosts* (by the search engine operator) and/or with *Term Boosts* (by the search engine user)
- *Field Boosts* increase the score if a query term appears in a specific field
- *Term Boosts* increase the score for a specific query term
- *Boost Queries* are ranking rules which are statically attached to every query; they can increase the score for a static defined document attribute, like:
  - boost documents containing a specifc string (i.e. „2015“)
- *Boost Functions* are add-on terms to the weight functions which can change the ranking completely, like
  - boost documents independently from the query, based only on document metadata (i.e. the document date)

Ranking Boosts in YaCy (config for embedded Solr)

Boost Function

A Boost Function can combine numeric values from the result document to produce a number which is multiplied with the score value from the query result. To see all available fields, see the YaCy Solr Schema and look for numeric values (these are names with suffix '_i'). To find out which kind of operations are possible, see the Solr Function Query documentation. Example: to order by date, use "recip(ms(NOW,last_modified),3.16e-11,1,1)", to order by crawldepth, use "div(100,add(crawldepth_i,1))".

boost=

Set Boost Function | Re-Set to default

Boost Query

The Boost Query is attached to every query. Use this to statically boost specific content in the index. Example: "fuzzy_signature_unique_b:true^100000.0f" means that documents, identified as 'double' are ranked very bad and appended to the end of all results (because the unique are ranked high). To find appropriate fields for this query, see the YaCy Solr Schema and look for boolean values (with suffix '_b') or tags inside string fields (with suffix '_s' or '_sxt').

bq=crawldepth_i:0^0.8 crawldepth_i:1^0.4

Set Boost Query | Re-Set to default

Solr Boosts

This is the set of searchable fields (see YaCy Solr Schema). Entries without a boost value are not searched. Boost values make hits inside the corresponding field more important.

title	5.0	content of title tag
publisher_t		the name of the publisher of the document
author	1.0	content of author-tag
description_txt		content of description-tag(s)
keywords	2.0	content of keywords tag; words are separated by
text_t	1.0	all visible text
synonyms_sxt	0.5	additional synonyms to the words in the text
h1_txt	5.0	h1 header
h2_txt	3.0	h2 header
h3_txt		h3 header
url_paths_sxt	3.0	all path elements in the url hpath (see:
host_s	6.0	host of the url
host_organization_s		either the second level domain or, if a ccSLD is

Set Field Boosts | Re-Set to default

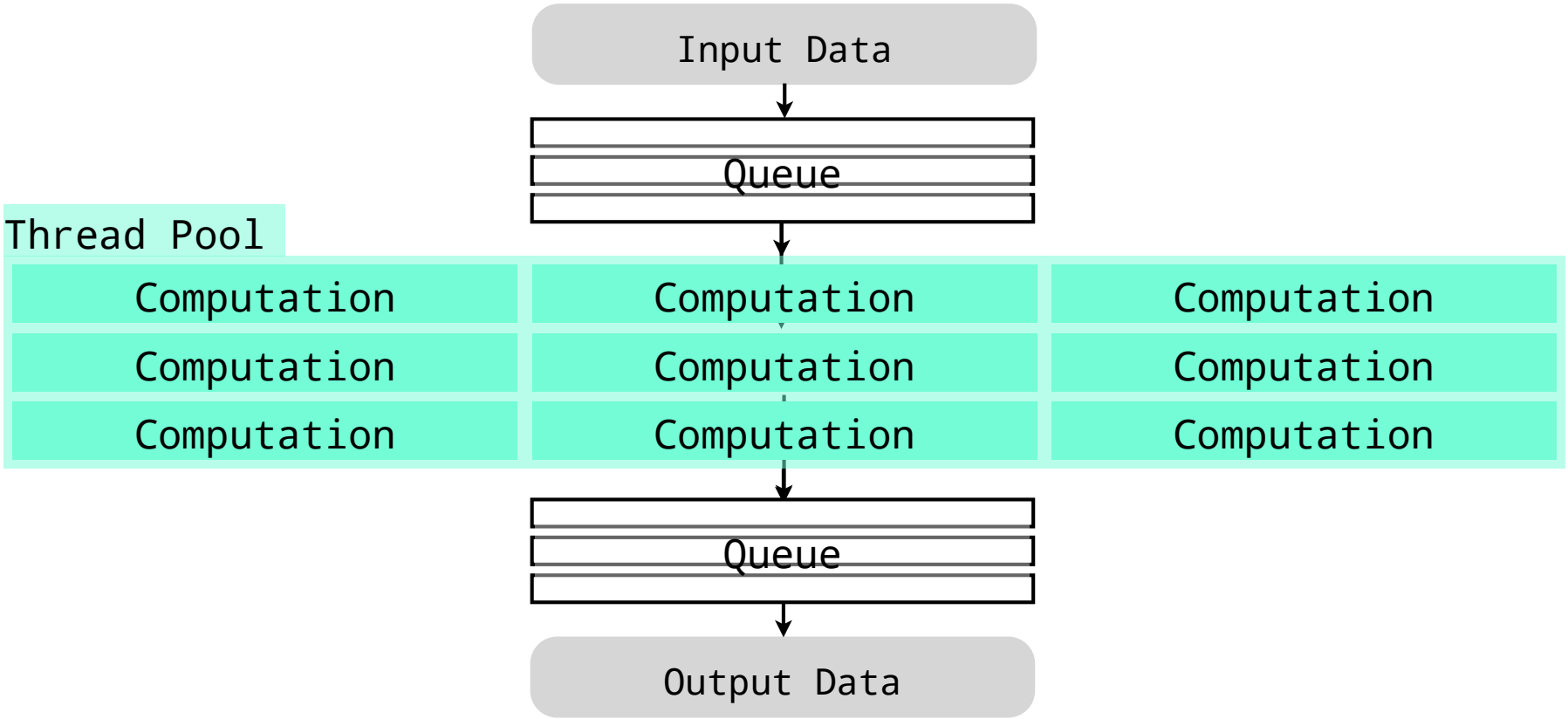
Introduction to Solr - easy test installation

Search Engine - ‘Hello World‘:

- curl -OL http://ftp.fau.de/apache/lucene/solr/5.1.0/solr-5.1.0.tgz
- tar xfz solr-5.1.0.tgz
- cd solr-5.1.0
- bin/solr -e schemaless
- open http://localhost:8983 (redirects to http://localhost:8983/solr/#/)
- curl 'http://localhost:8983/solr/gettingstarted/update/json?commit=true' -H 'Content-type:application/json' -d '{"add":{"doc":{"id":"data1", "title":"Hello World"}}}'
- bin/post -c gettingstarted docs/
- bin/post -c gettingstarted http://lucene.apache.org/solr -recursive 1 -delay 1
- curl 'http://localhost:8983/solr/gettingstarted/select/?q=*%3A*'
- http://localhost:8983/solr/gettingstarted/browse
- bin/solr stop -all ; rm -Rf example/schemaless/
- http://lucene.apache.org/solr/quickstart.html

Distributed IR Architecture - Parallel Computing

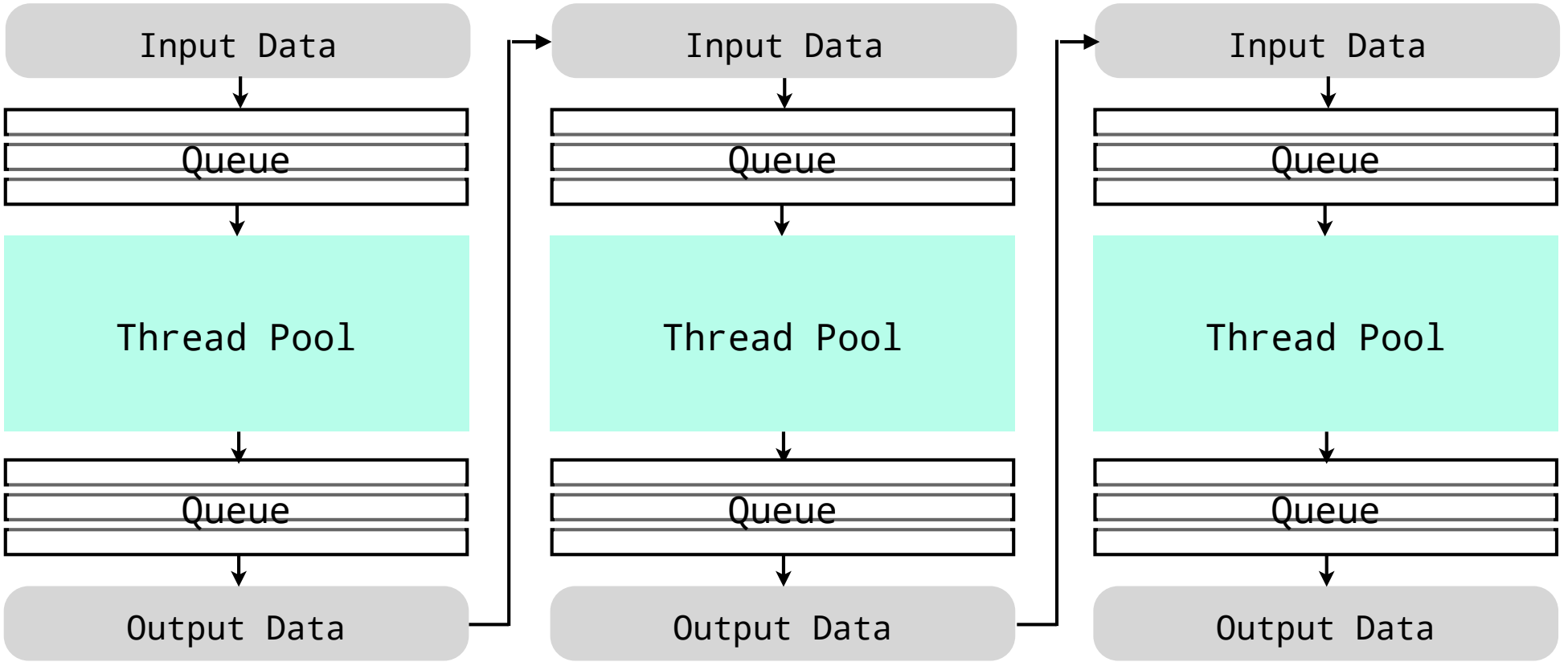
Parallel Computing Concept „MIMD“



Distributed IR Architecture - Parallel Computing

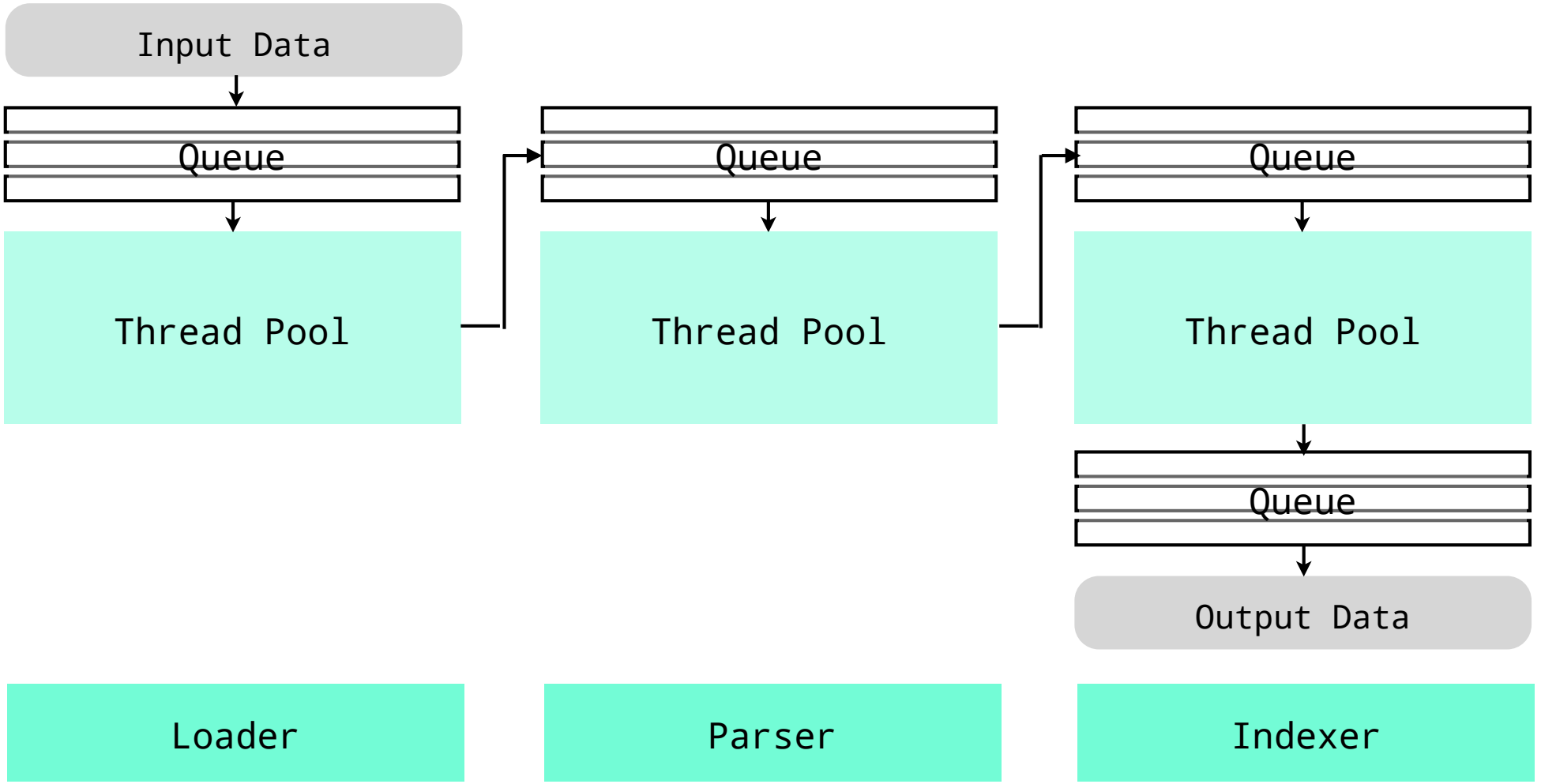
Parallel Computing Concept

„MIMD“



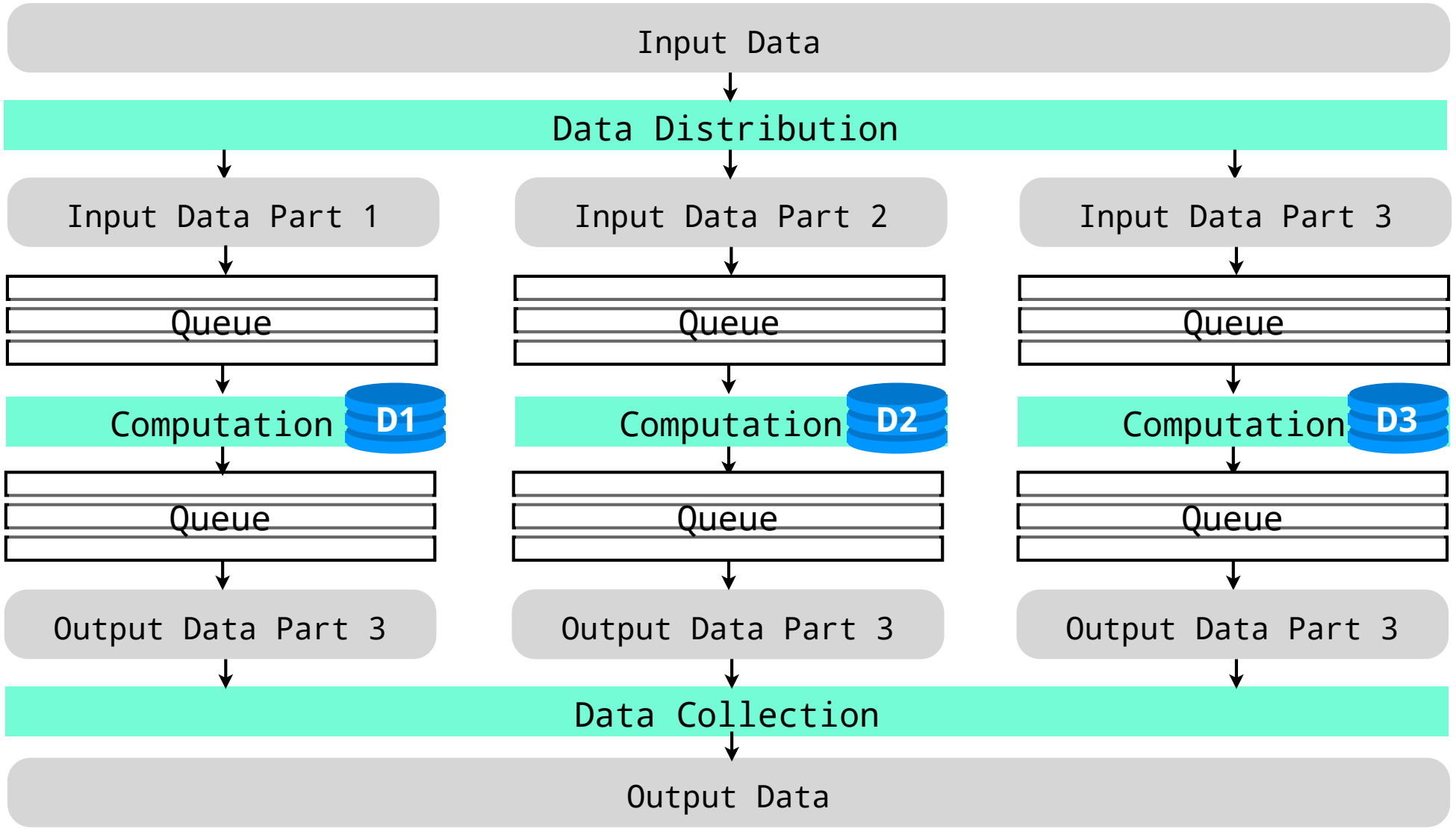
Distributed IR Architecture - Parallel Computing

Parallel Computing Concept „MIMD“



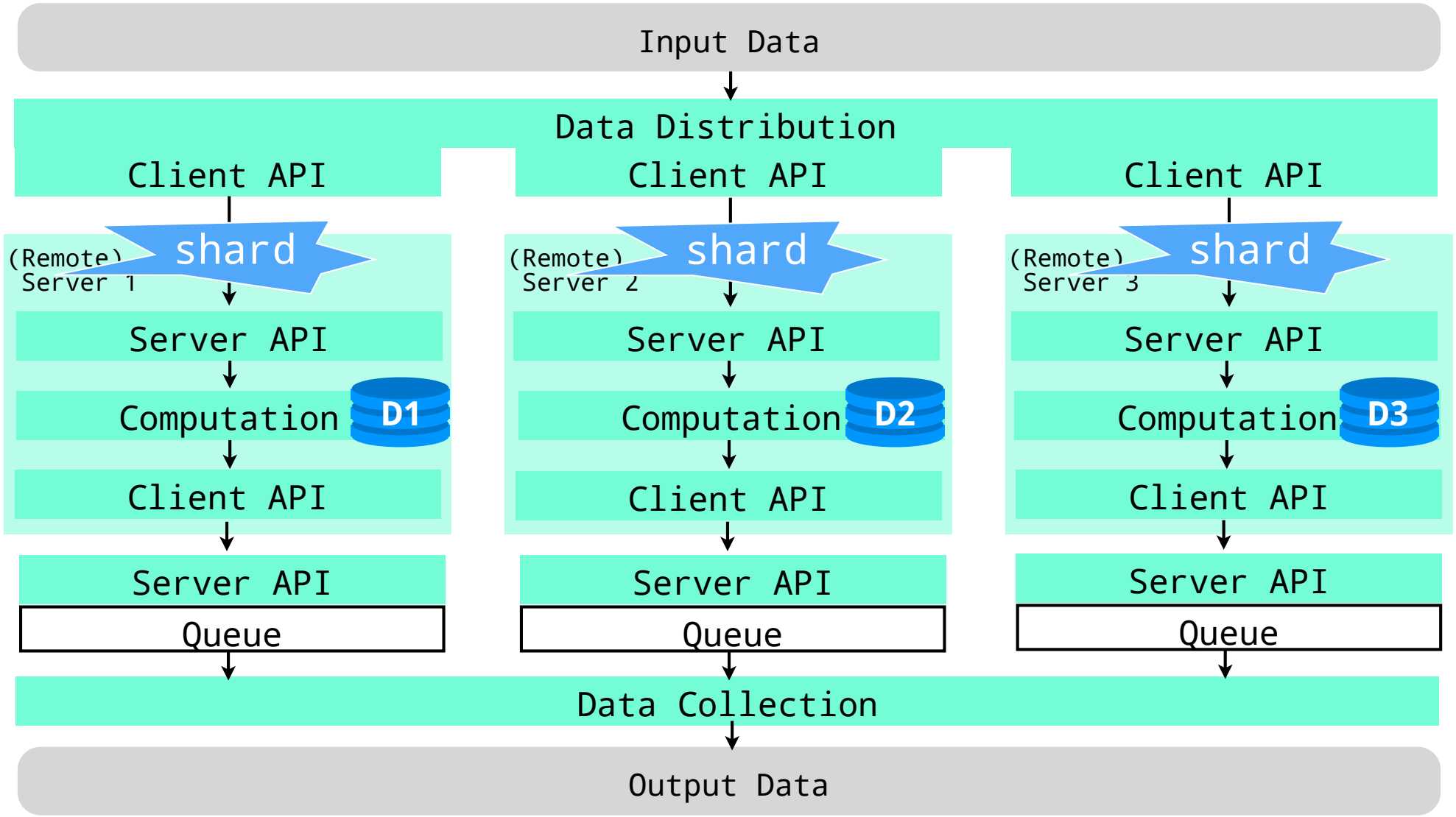
Distributed IR Architecture - Parallel Computing

Parallel Computing Concept „SIMD“



Distributed IR Architecture - Parallel Computing

Parallel Computing Concept „SIMD“



Distributed IR Architecture - Distributed Indexing

How to distribute to the shards ?

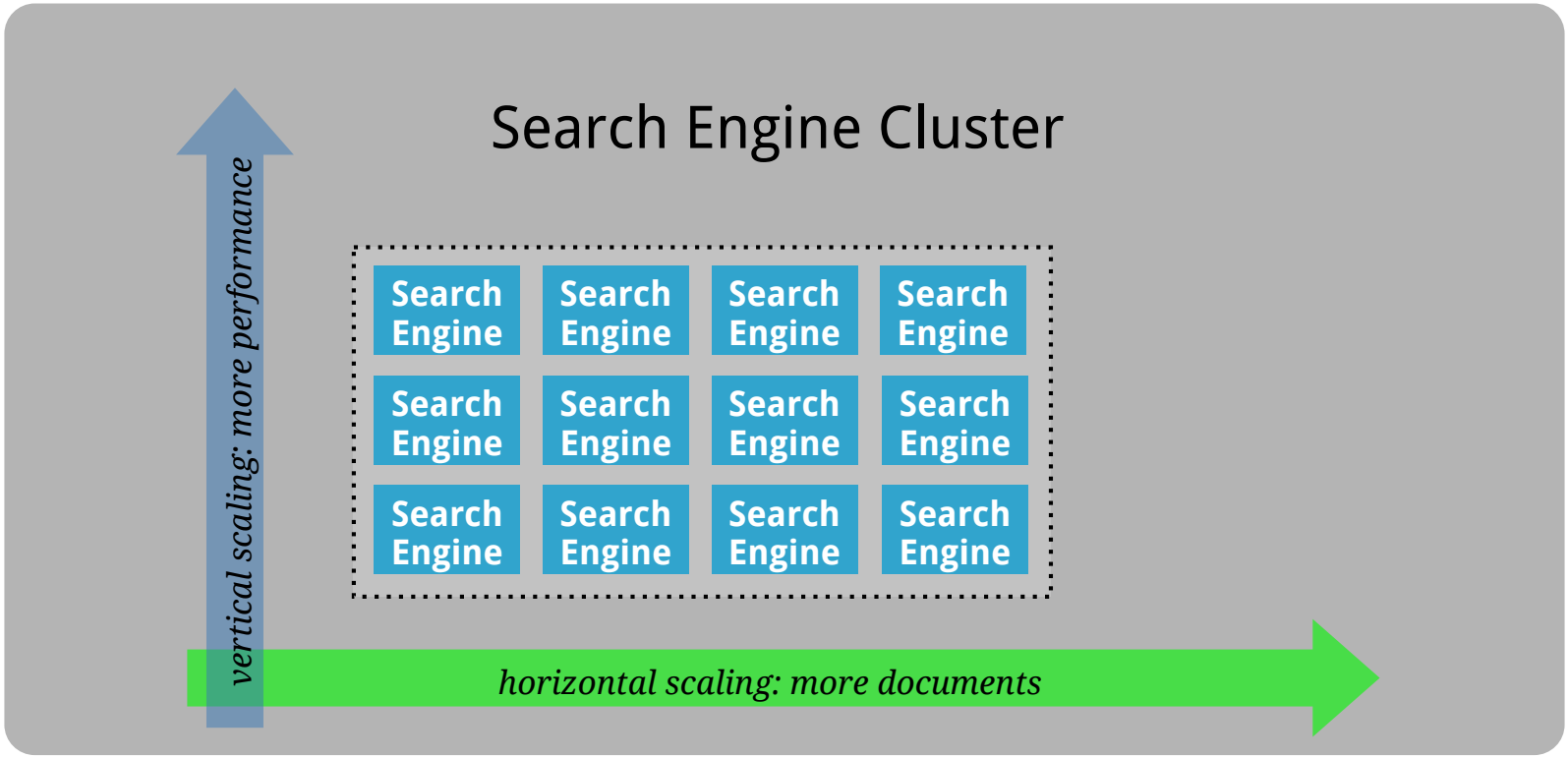
Data Distribution

- *round-robin:* iteration over all targets
- *by document:* hashing over ID of document
- *by term:* hashing over index terms

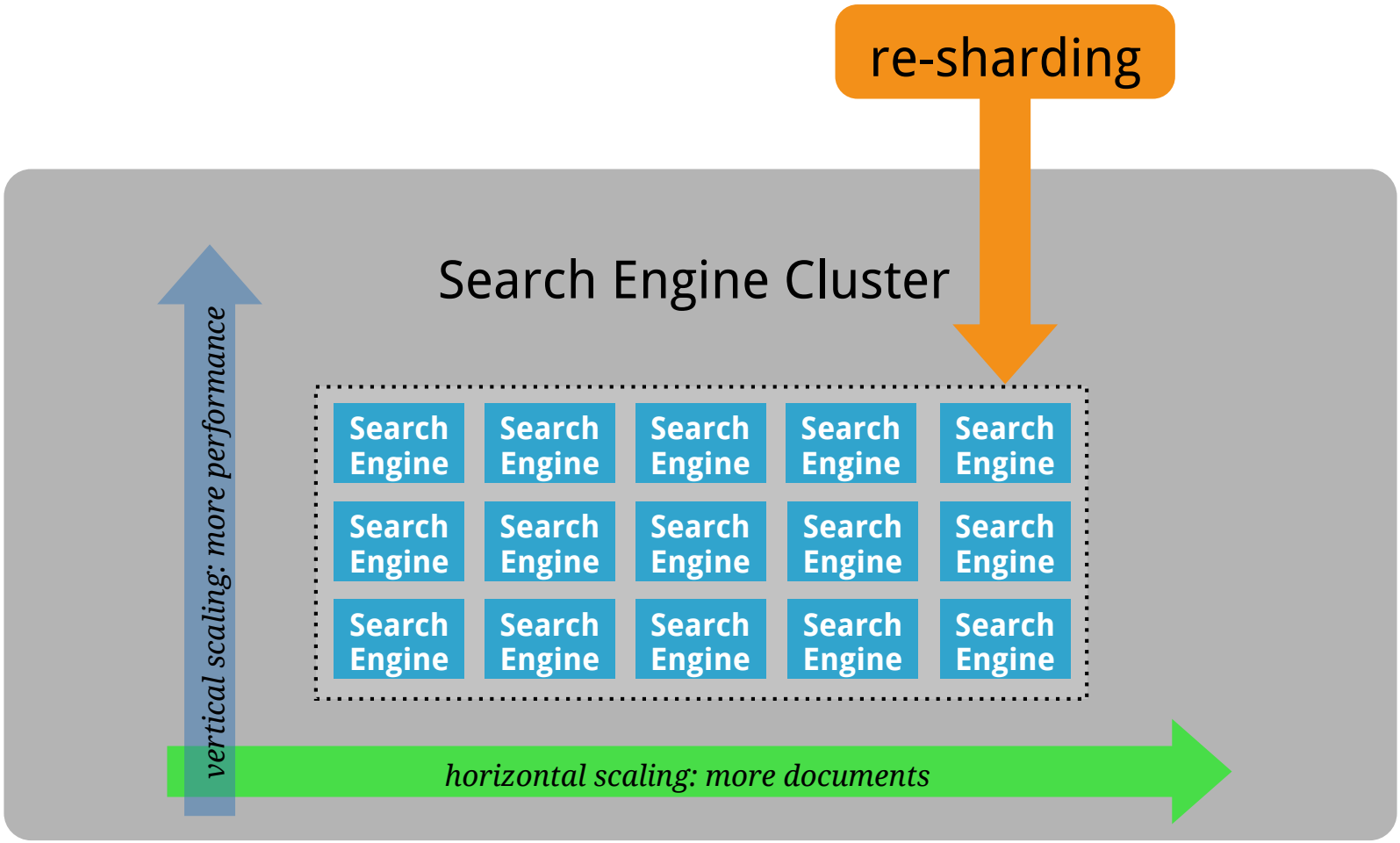
Data Collection

- *no hint where to search:* all shards must be queried
- *no hint where to search:* all shards must be queried
- *search term matches distribution term:* one query to only one shard

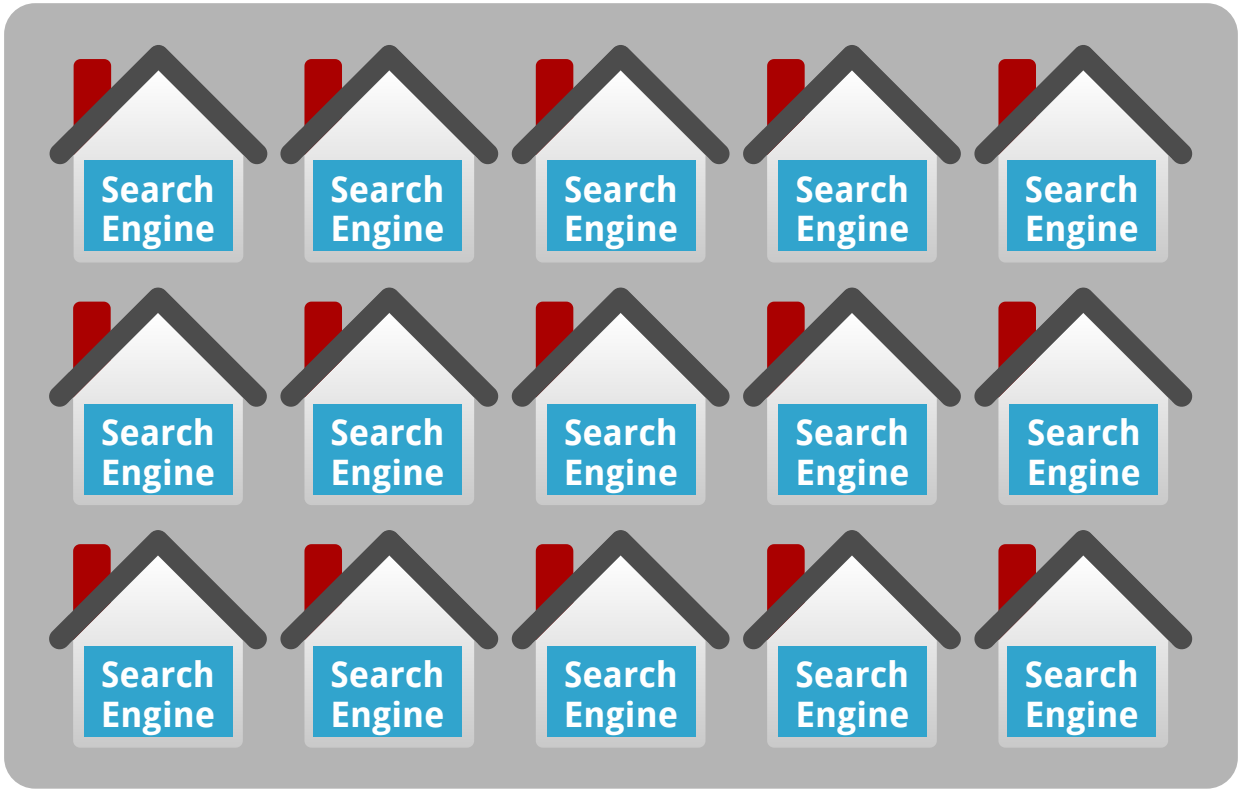
Distributed IR Architecture - Search Engine Scaling



Distributed IR Architecture - Search Engine Scaling



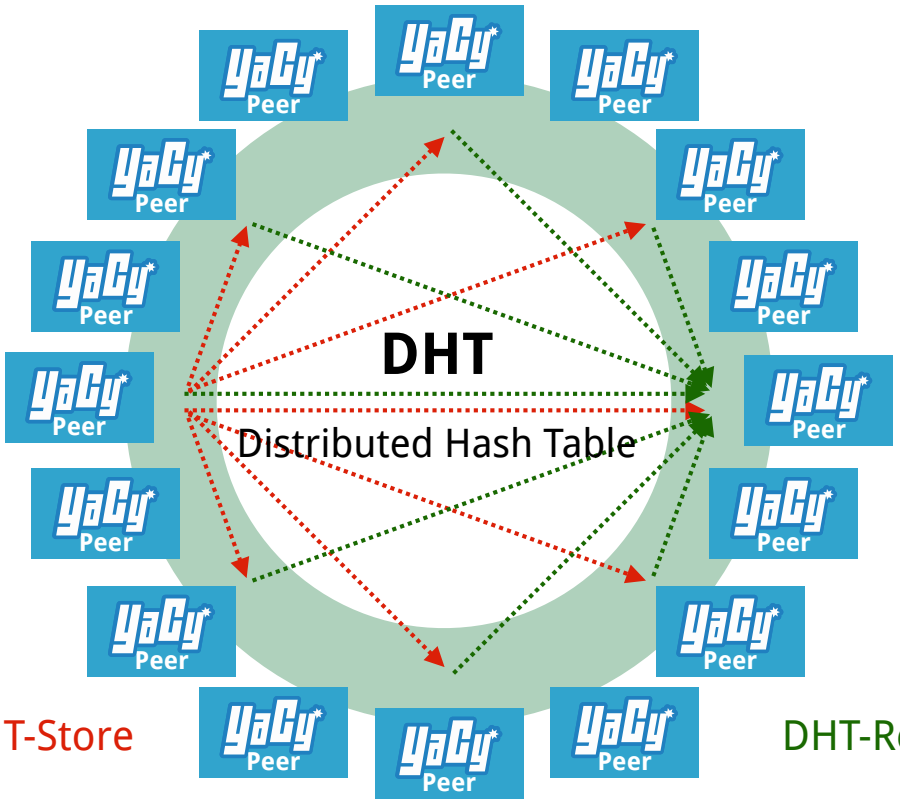
Distributed IR Architecture - Scaling with YaCy & Decentralized



Distributed IR Architecture - Scaling with YaCy & Decentralized

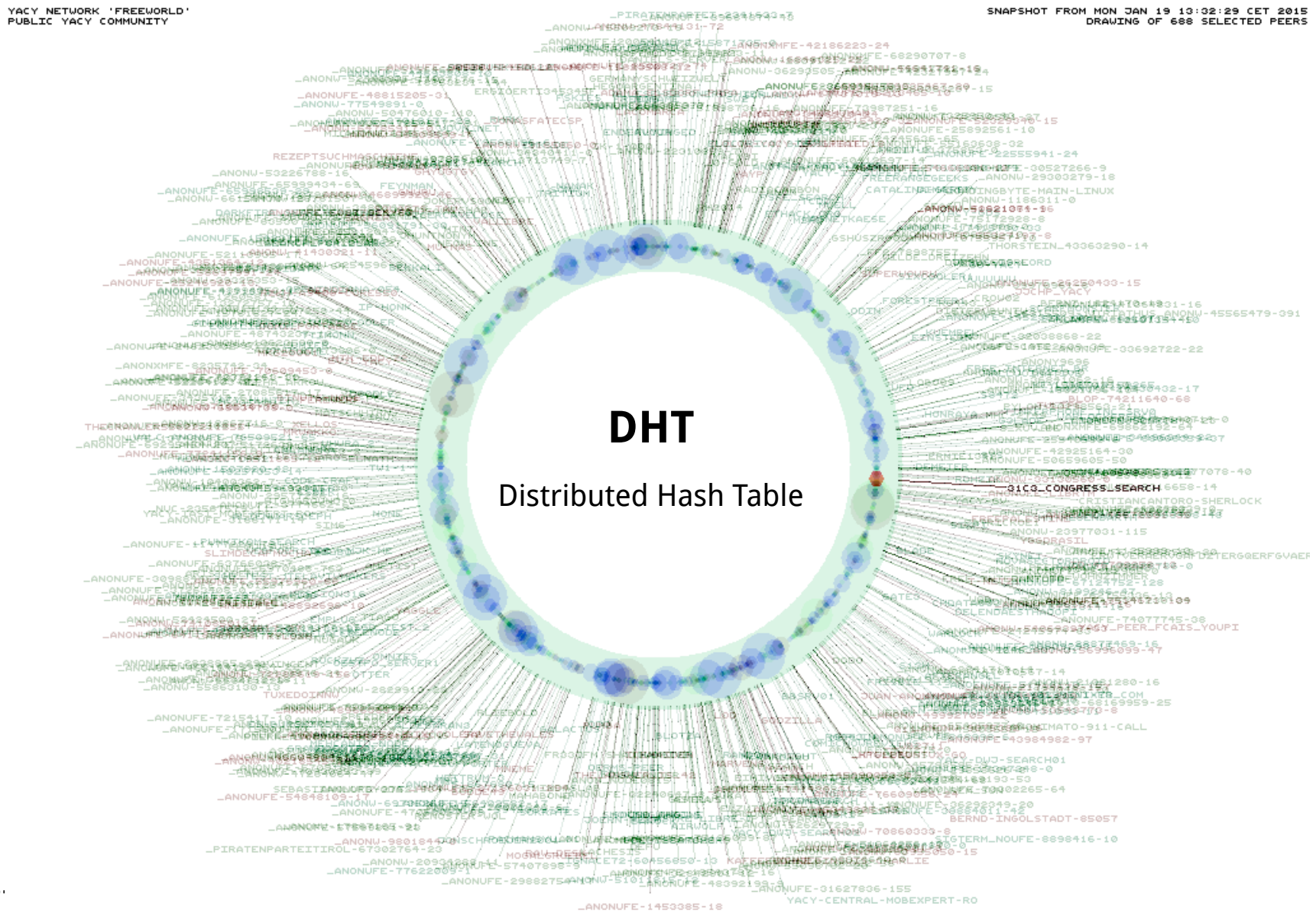
Distributed IR Architecture - Scaling with YaCy & Decentralized

Crawl the web, create a web index, distribute the index



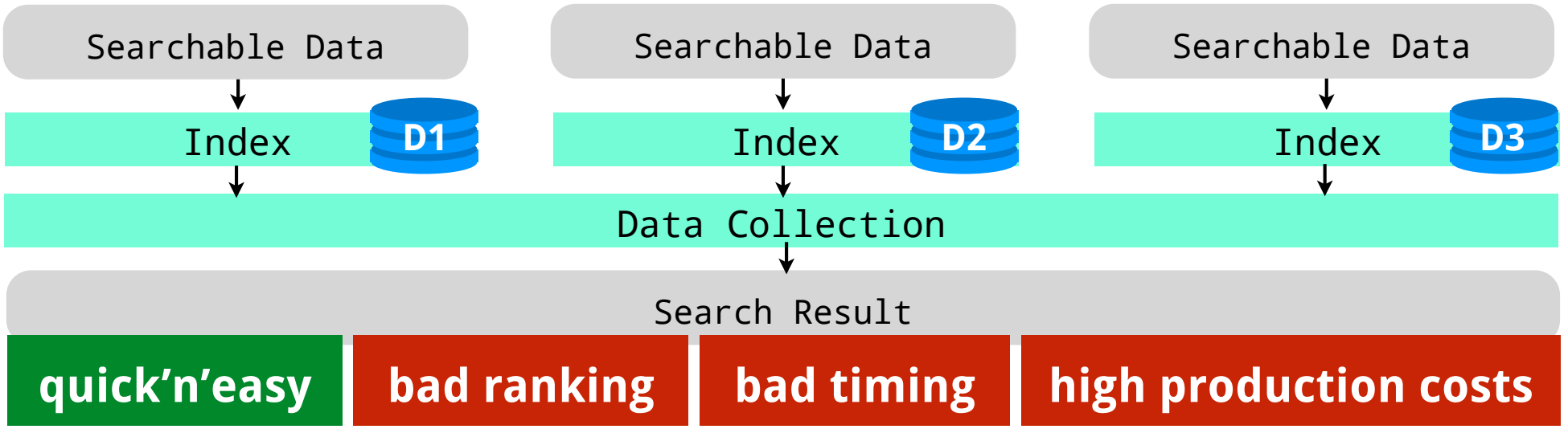
Search in a Distributed Hash Table

Distributed IR Architecture - Scaling with YaCy & Decentralized

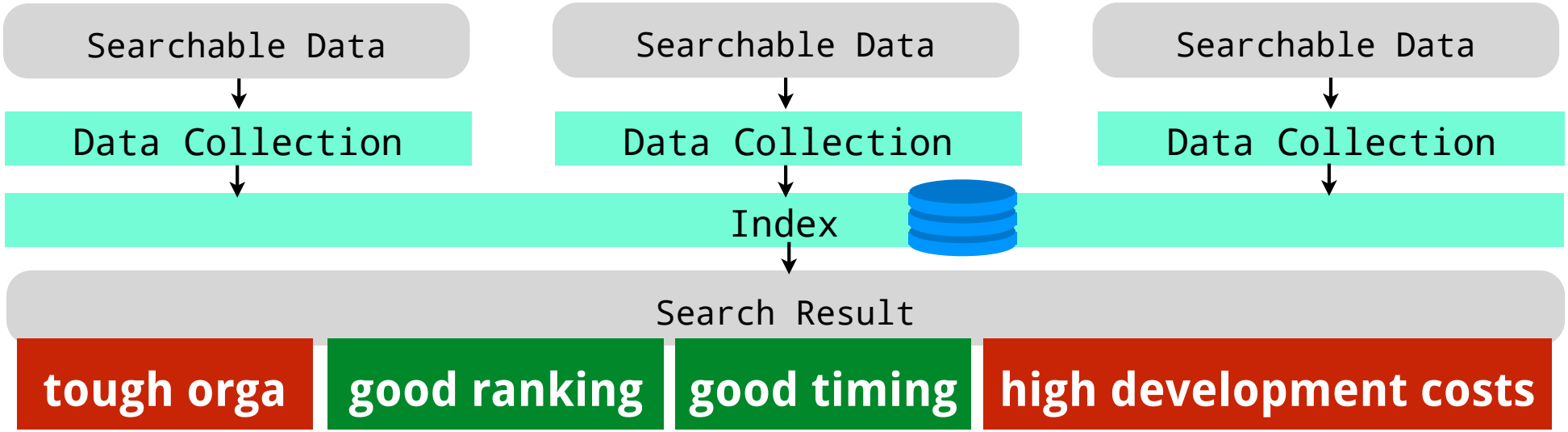


Federated Search Multiple Searchable Resources

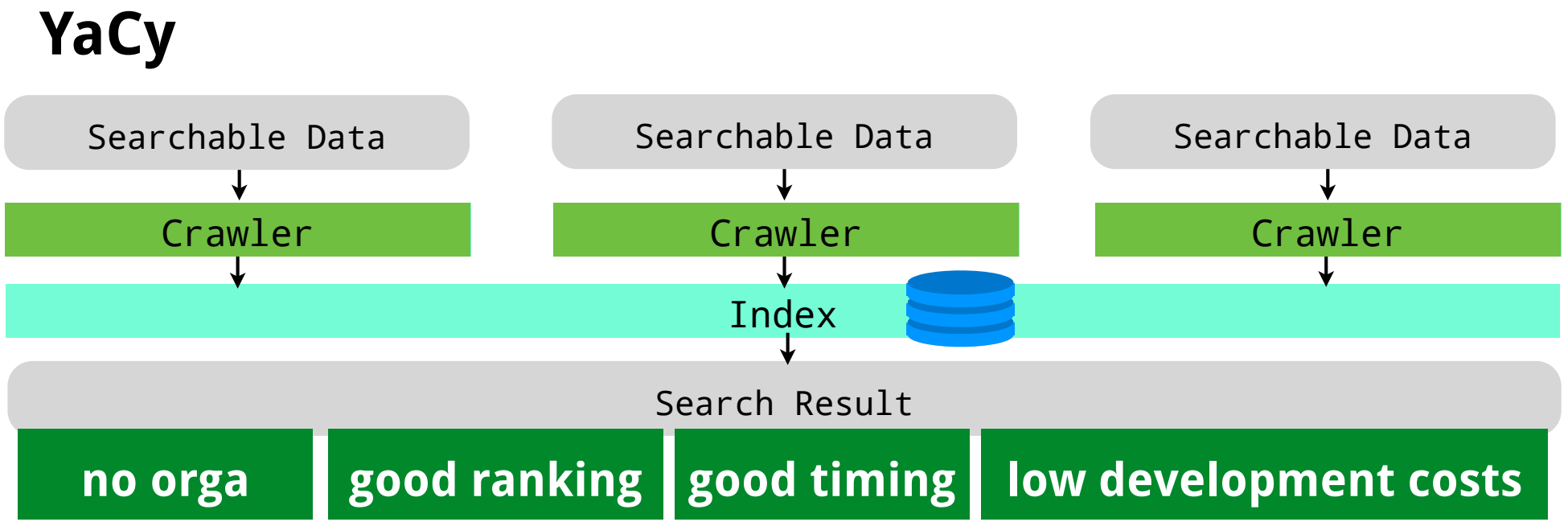
Schema 1: Federated Search Engines (aka Meta-Search)



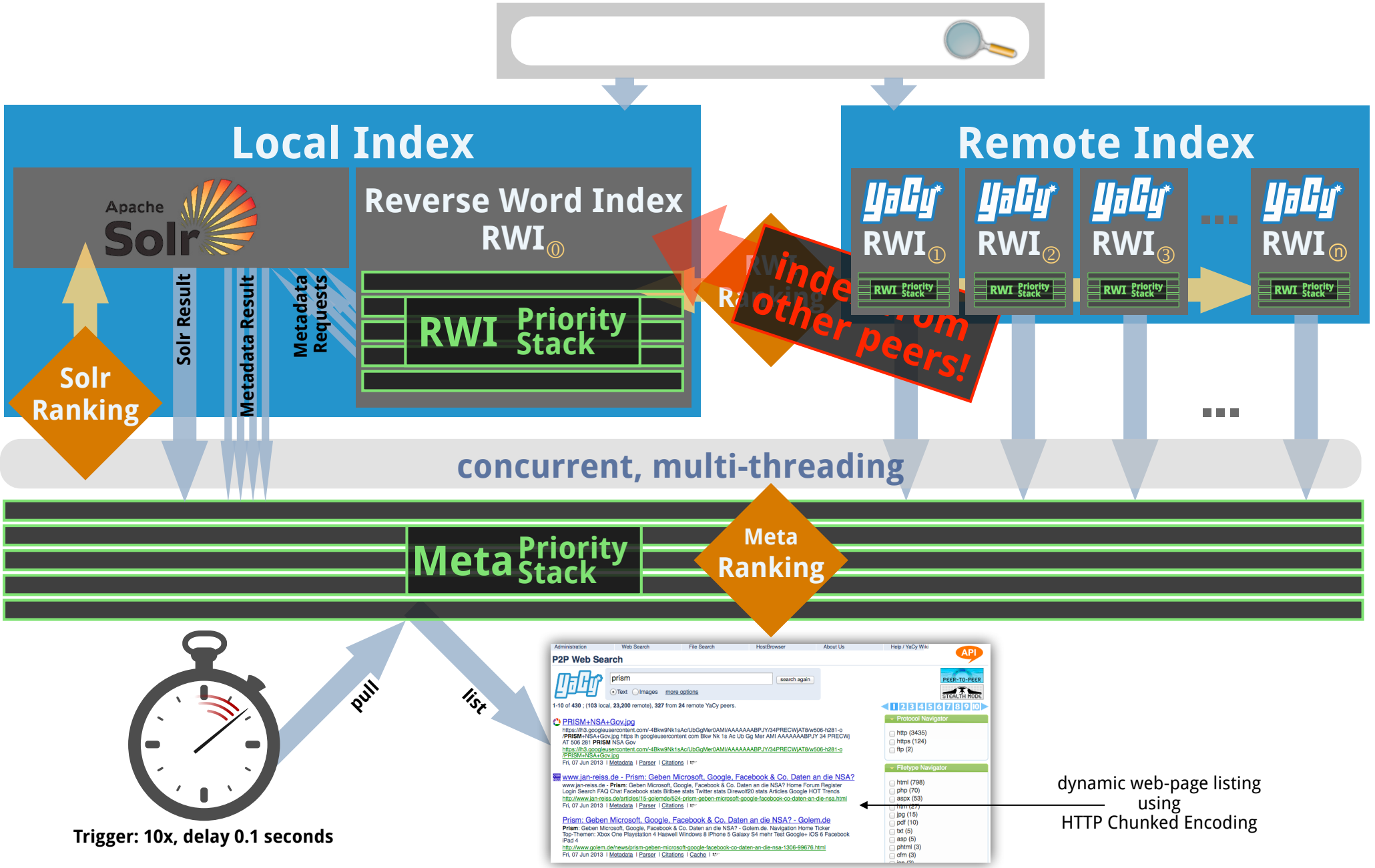
Schema 2: Federated Data



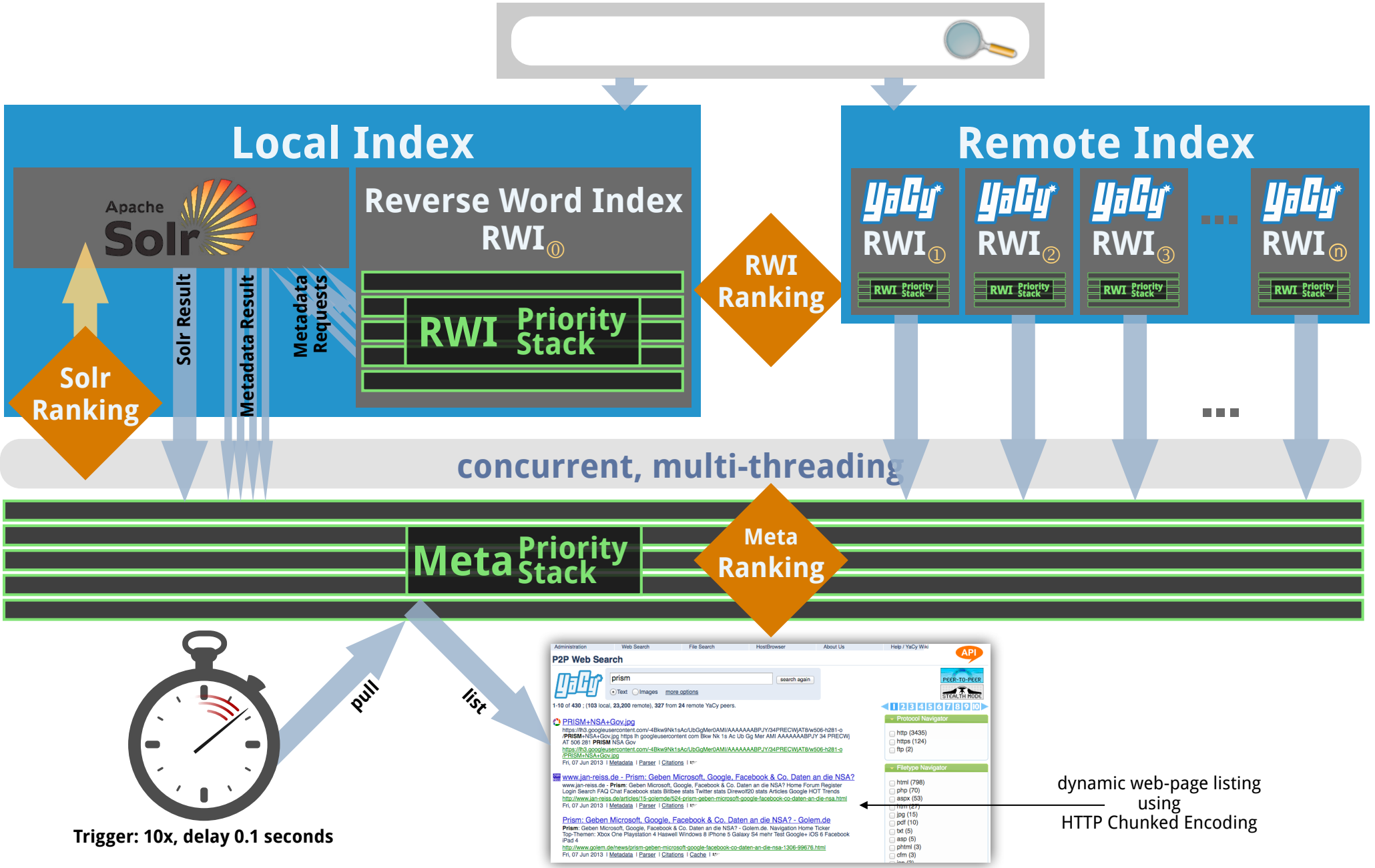
Federated Search Multiple Searchable Resources



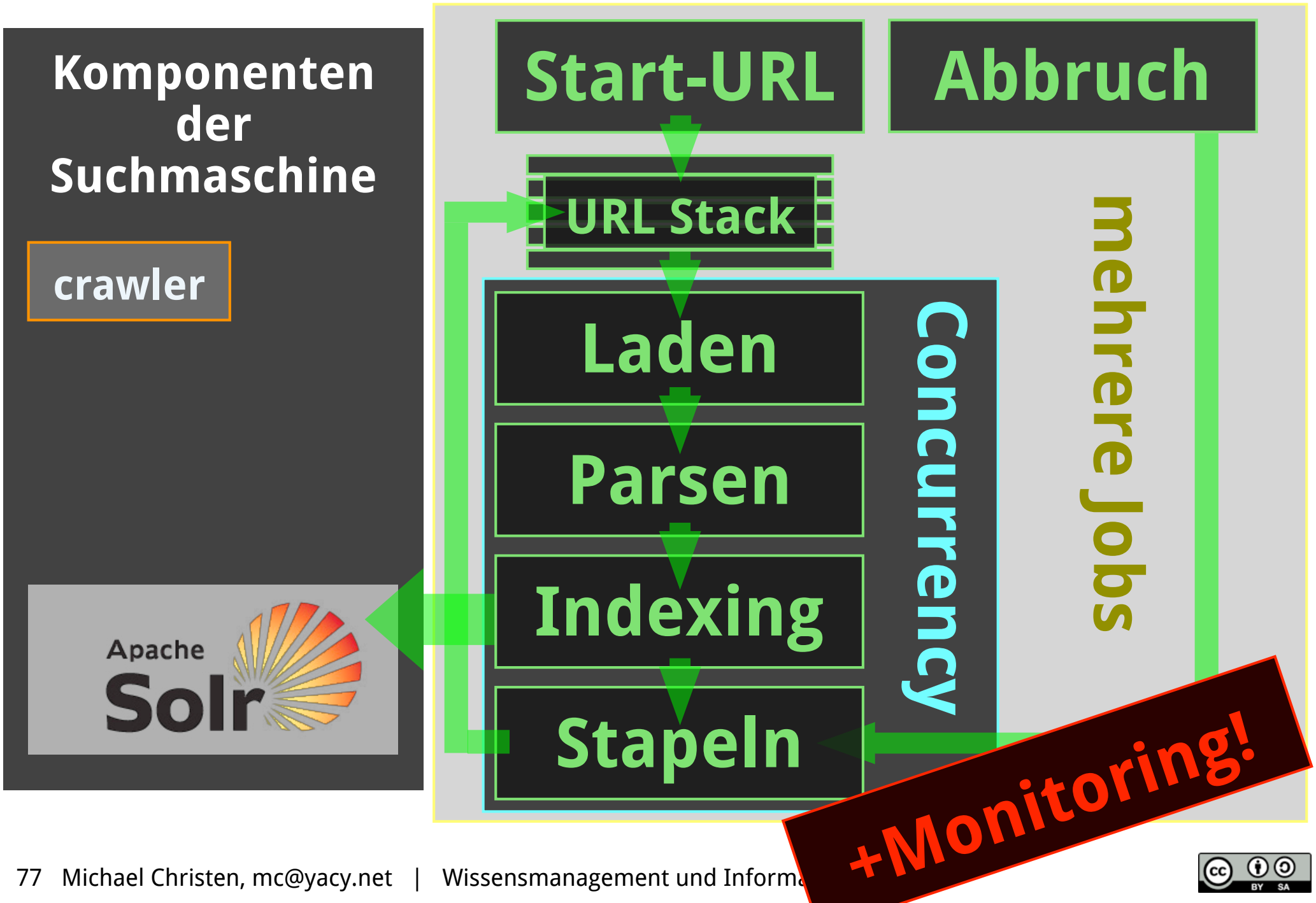
Federated Search in YaCy (peer-to-peer search)



‚Portal' Search in YaCy (without peer-to-peer)



Funktionskomponenten einer Suchmaschine

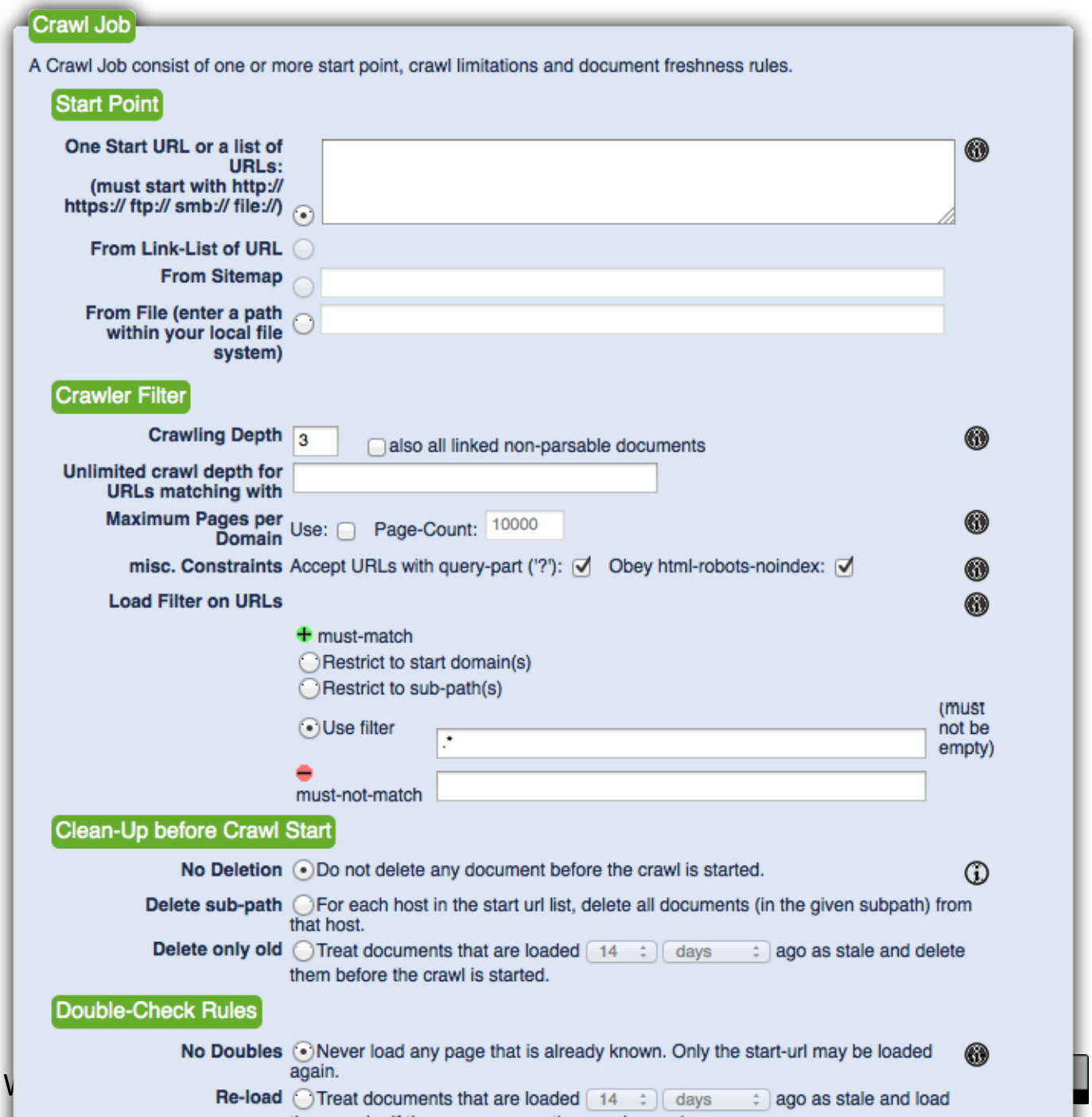


Funktionskomponenten einer Suchmaschine

Komponenten der Suchmaschine

crawler

Apache Solr



Funktionskomponenten einer Suchmaschine

Komponenten der Suchmaschine

crawler | parser

Apache Solr

Content Parser Settings

enable/disable	Extension
Microsoft Powerpoint Parser	
☑	pps
☑	ppt
GNU Zip Compressed Archive Parser	
☑	gz
☑	tgz
Adobe Flash Parser	
☑	swf
vCard Parser	
☑	vcf
Audio File Meta-Tag Parser	
☐	m4p
☐	m4a
☐	oga
☐	flac
☐	ogg
☐	mp3
☐	wma
Comma Separated Value Parser	
☐	csv
Microsoft Visio Parser	
☑	vdx
☑	vtx
☑	vss
☑	vsd
☑	vst
Generic Image Parser	
☑	bmp
☑	jpg
☑	jpeg
☑	png
☑	jpe
☑	gif
FreeMind Parser	
☑	mm
PostScript Document Parser	
☑	ps
Commodore 64 SID Audio File Parser	
☑	sid
Open Office XML Document Parser	
☑	xltx
☑	xlsx
☑	ppsx
☑	docx
☑	pptx
Torrent Metadata Parser	
☑	torrent
Word Document Parser	
☑	doc
OASIS OpenDocument V2 Text Document	
☑	odg
☑	odf
Bzip 2 UNIX Compressed File Parser	
☑	tbz
☑	tbz2
☑	bz2
Streaming HTML Parser	
☑	xhtml
☑	php4
☑	php5
☑	php3
☑	shtml
☑	html
☑	htm
Microsoft Excel Parser	
☑	xla
☑	xls
ZIP File Parser	
☑	zip
☑	jar
☑	apk
Acrobat Portable Document Parser	
☑	pdf
7zip Archive Parser	
☑	7z
RSS Parser	
☑	rss
☑	xml
Tape Archive File Parser	
☑	tar
Rich Text Format Parser	
☑	rtf
RDF Parser	
☑	rdf

Submit

Funktionskomponenten einer Suchmaschine

Komponenten der Suchmaschine

crawler | parser

Index Schema

Apache Solr

Active	Attribute	Comment	show active	show all available	show disabled
☑	id	primary key of document, the URL hash **mandatory field**			
☑	sku	url of document			
☑	last_modified	last-modified from http header			
☐	dates_in_content_sxt	if date expressions can be found in the content, these dates are listed here in order of the appearances			
☐	dates_in_content_count_i	the number of entries in dates_in_content_sxt			
☐	date_in_content_min_dt	if dates_in_content_sxt is filled, this contains the oldest date from the list of available dates			
☐	date_in_content_max_dt	if dates_in_content_sxt is filled, this contains the youngest date from the list of available dates, that may also be possibly in the future			
☑	content_type	mime-type of document			
☐	http_unique_b	unique-field which is true when an url appears the first time. If the same url which was http then appears as https (or vice versa) then the field is false			
☐	www_unique_b	unique-field which is true when an url appears the first time. If the same url within the subdomain www then appears without that subdomain (or vice versa) then the field is false			
☑	title	content of title tag			
☐	title_exact_signature_l	the 64 bit hash of the org.apache.solr.update.processor.Lookup3Signature of title, used to compute title_unique_b			
☐	title_unique_b	flag shows if title is unique within all indexable documents of the same host with status code 200; if yes and another document appears with same title, the unique-flag is set to false			
☑	host_id_s	id of the host, a 6-byte hash that is part of the document id			
☑	md5_s	the md5 of the raw source			
☑	exact_signature_l	the 64 bit hash of the org.apache.solr.update.processor.Lookup3Signature of text_t			
☑	exact_signature_unique_b	flag shows if exact_signature_l is unique at the time of document creation, used for double-check during search			
☐	exact_signature_copycount_i	counter for the number of documents which are not unique (== count of not-unique-flagged documents + 1)			
☑	fuzzy_signature_l	64 bit of the Lookup3Signature from EnhancedTextProfileSignature of text_t			
☐	fuzzy_signature_text_t	intermediate data produced in EnhancedTextProfileSignature: a list of word frequencies			
☑	fuzzy_signature_unique_b	flag shows if fuzzy_signature_l is unique at the time of document creation, used for double-check during search			
☐	fuzzy_signature_copycount_i	counter for the number of documents which are not unique (== count of not-unique-flagged documents + 1)			
☑	size_i	the size of the raw source			
☑	failreason_s	fail reason if a page was not loaded. if the page was loaded then this field is empty			
☑	failtype_s	fail type if a page was not loaded. This field is either empty, 'excl' or 'fail'			
☑	httpstatus_i	html status return code (i.e. "200" for ok), -1 if not loaded			
☐	httpstatus_redirect_s	redirect url if the error code is 299 < httpstatus_i < 310			
☑	references_i	number of unique http references, should be equal to references_internal_i + references_external_i			
☑	references_internal_i	number of unique http references from same host to referenced url			
☑	references_external_i	number of unique http references from external hosts			
☑	references_exthosts_i	number of external hosts which provide http references			
☑	crawldepth_i	crawl depth of web page according to the number of steps that the crawler did to get to this document; if the crawl was started at a root			

Funktionskompon

Komponenten der Suchmaschine

crawler | parser

Index Schema

Apache Solr

☐	httpstatus_redirect_s	redirect url if the error code is 299 < httpstatus_i < 310
☑	references_i	number of unique http references, should be equal to references_internal_i + references_external_i
☑	references_internal_i	number of unique http references from same host to referenced url
☑	references_external_i	number of unique http references from external hosts
☑	references_exthosts_i	number of external hosts which provide http references
☑	crawldepth_i	crawl depth of web page according to the number of steps that the crawler did to get to this document; if the crawl was started at a root document, then this is equal to the clickdepth
☑	process_sxt	needed (post-)processing steps on this metadata set
☑	harvestkey_s	key from a harvest process (i.e. the crawl profile hash key) which is needed for near-realtime postprocessing. This shall be deleted as soon as postprocessing has been terminated.
☑	load_date_dt	time when resource was loaded
☑	fresh_date_dt	date until resource shall be considered as fresh
☑	referrer_id_s	id of the referrer to this document, discovered during crawling
☑	publisher_t	the name of the publisher of the document
☑	language_s	the language used in the document
☑	audiolinkscount_i	number of links to audio resources
☑	videolinkscount_i	number of links to video resources
☑	applinkscount_i	number of links to application resources
☑	coordinate_p	point in degrees of latitude,longitude as declared in WSG84
☐	coordinate_p_0_coordinate	automatically created subfield, (latitude)
☐	coordinate_p_1_coordinate	automatically created subfield, (longitude)
☐	ip_s	ip of host of url (after DNS lookup)
☑	author	content of author-tag
☐	author_sxt	content of author-tag as copy-field from author. This is used for facet generation
☑	description_txt	content of description-tag(s)
☐	description_exact_signature_l	the 64 bit hash of the org.apache.solr.update.processor.Lookup3Signature of description, used to compute description_unique_b
☐	description_unique_b	flag shows if description is unique within all indexable documents of the same host with status code 200; if yes and another document appears with same description, the unique-flag is set to false
☑	keywords	content of keywords tag; words are separated by space
☑	charset_s	character encoding
☑	wordcount_i	number of words in visible area
☑	linkscount_i	number of all outgoing links; including linksnofollowcount_i
☑	linksnofollowcount_i	number of all outgoing inks with nofollow tag
☑	inboundlinkscount_i	number of outgoing inbound (to same domain) links; including inboundlinksnofollowcount_i
☑	inboundlinksnofollowcount_i	number of outgoing inbound (to same domain) links with nofollow tag
☑	outboundlinkscount_i	number of outgoing outbound (to other domain) links, including outboundlinksnofollowcount_i
☑	outboundlinksnofollowcount_i	number of outgoing outbound (to other domain) links with nofollow tag
☑	imagescount_i	number of images
☑	responsetime_i	response time of target server in milliseconds
☑	text_t	all visible text
☑	synonyms_sxt	additional synonyms to the words in the text

Funktionskompon

Komponenten der Suchmaschine

crawler | parser

Index Schema

Apache Solr

☑	outboundlinkscount_i	number of outgoing outbound (to other domain) links, including outboundlinksnofollowcount_i
☑	outboundlinksnofollowcount_i	number of outgoing outbound (to other domain) links with nofollow tag
☑	imagescount_i	number of images
☑	responsetime_i	response time of target server in milliseconds
☑	text_t	all visible text
☑	synonyms_sxt	additional synonyms to the words in the text
☑	h1_txt	h1 header
☑	h2_txt	h2 header
☑	h3_txt	h3 header
☑	h4_txt	h4 header
☑	h5_txt	h5 header
☑	h6_txt	h6 header
☑	collection_sxt	tags that are attached to crawls/index generation to separate the search result into user-defined subsets
☐	csscount_i	number of entries in css_tag_txt and css_url_txt
☐	css_tag_sxt	full css tag with normalized url
☐	css_url_sxt	normalized urls within a css tag
☐	scripts_sxt	normalized urls within a scripts tag
☐	scriptscount_i	number of entries in scripts_sxt
☐	robots_i	content of <meta name="robots" content=#content#> tag and the "X-Robots-Tag" HTTP property
☐	metagenerator_t	content of <meta name="generator" content=#content#> tag
☑	inboundlinks_protocol_sxt	internal links, only the protocol
☑	inboundlinks_urlstub_sxt	internal links, the url only without the protocol
☑	inboundlinks_anchortext_txt	internal links, the visible anchor text
☑	outboundlinks_protocol_sxt	external links, only the protocol
☑	outboundlinks_urlstub_sxt	external links, the url only without the protocol
☑	outboundlinks_anchortext_txt	external links, the visible anchor text
☑	images_text_t	all text/words appearing in image alt texts or the tokenized url
☑	images_urlstub_sxt	all image links without the protocol and '://'
☑	images_protocol_sxt	all image link protocols
☑	images_alt_sxt	all image link alt tag
☑	images_height_val	size of images:height
☑	images_width_val	size of images:width
☐	images_pixel_val	size of images as number of pixels (easier for a search restriction than with and height)
☐	images_withalt_i	number of image links with alt tag
☐	htags_i	binary pattern for the existance of h1..h6 headlines
☐	canonical_s	url inside the canonical link element
☐	canonical_equal_sku_b	flag shows if the url in canonical_t is equal to sku
☐	refresh_s	link from the url property inside the refresh link element
☐	li_txt	all texts in tags

Funktionskompon

Komponenten der Suchmaschine

crawler | parser

Index Schema

Apache Solr

☐	canonical_s	url inside the canonical link element
☐	canonical_equal_sku_b	flag shows if the url in canonical_t is equal to sku
☐	refresh_s	link from the url property inside the refresh link element
☐	li_txt	all texts in tags
☐	licount_i	number of tags
☑	bold_txt	all texts inside of <b> or tags. no doubles. listed in the order of number of occurrences in decreasing order
☐	boldcount_i	total number of occurrences of <b> or
☑	italic_txt	all texts inside of <i> tags. no doubles. listed in the order of number of occurrences in decreasing order
☐	italiccount_i	total number of occurrences of <i>
☑	underline_txt	all texts inside of <u> tags. no doubles. listed in the order of number of occurrences in decreasing order
☐	underlinecount_i	total number of occurrences of <u>
☐	flash_b	flag that shows if a swf file is linked
☐	frames_sxt	list of all links to frames
☐	framesscount_i	number of frames_txt
☐	iframes_sxt	list of all links to iframes
☐	iframesscount_i	number of iframes_txt
☐	hreflang_url_sxt	url of the hreflang link tag, see http://support.google.com/webmasters/bin/answer.py?hl=de&answer=189077
☐	hreflang_cc_sxt	country code of the hreflang link tag, see http://support.google.com/webmasters/bin/answer.py?hl=de&answer=189077
☐	navigation_url_sxt	page navigation url, see http://googlewebmastercentral.blogspot.de/2011/09/pagination-with-relnext-and-relprev.html
☐	navigation_type_sxt	page navigation rel property value, can contain one of {top,up,next,prev,first,last}
☐	publisher_url_s	publisher url as defined in http://support.google.com/plus/answer/1713826?hl=de
☑	url_protocol_s	the protocol of the url
☑	url_file_name_s	the file name (which is the string after the last '/' and before the query part from '?' on) without the file extension
☐	url_file_name_tokens_t	tokens generated from url_file_name_s which can be used for better matching and result boosting
☑	url_file_ext_s	the file name extension
☑	url_paths_count_i	number of all path elements in the url hpath (see: http://www.ietf.org/rfc/rfc1738.txt) without the file name
☑	url_paths_sxt	all path elements in the url hpath (see: http://www.ietf.org/rfc/rfc1738.txt) without the file name
☐	url_parameter_i	number of key-value pairs in search part of the url
☐	url_parameter_key_sxt	the keys from key-value pairs in the search part of the url
☐	url_parameter_value_sxt	the values from key-value pairs in the search part of the url
☑	url_chars_i	number of all characters in the url == length of sku field
☑	host_s	host of the url
☐	host_dnc_s	the Domain Class Name, either the TLD or a combination of ccSLD+TLD if a ccSLD is used.
☑	host_organization_s	either the second level domain or, if a ccSLD is used, the third level domain
☐	host_organizationdnc_s	the organization and dnc concatenated with '.'
☐	host_subdomain_s	the remaining part of the host without organizationdnc
☑	host_extent_i	number of documents from the same host; can be used to measure references_internal_i for likelihood computation
☐	title_count_i	number of titles (counting the 'title' field) in the document
☐	title_chars_val	number of characters for each title

Funktionskompon

Komponenten der Suchmaschine

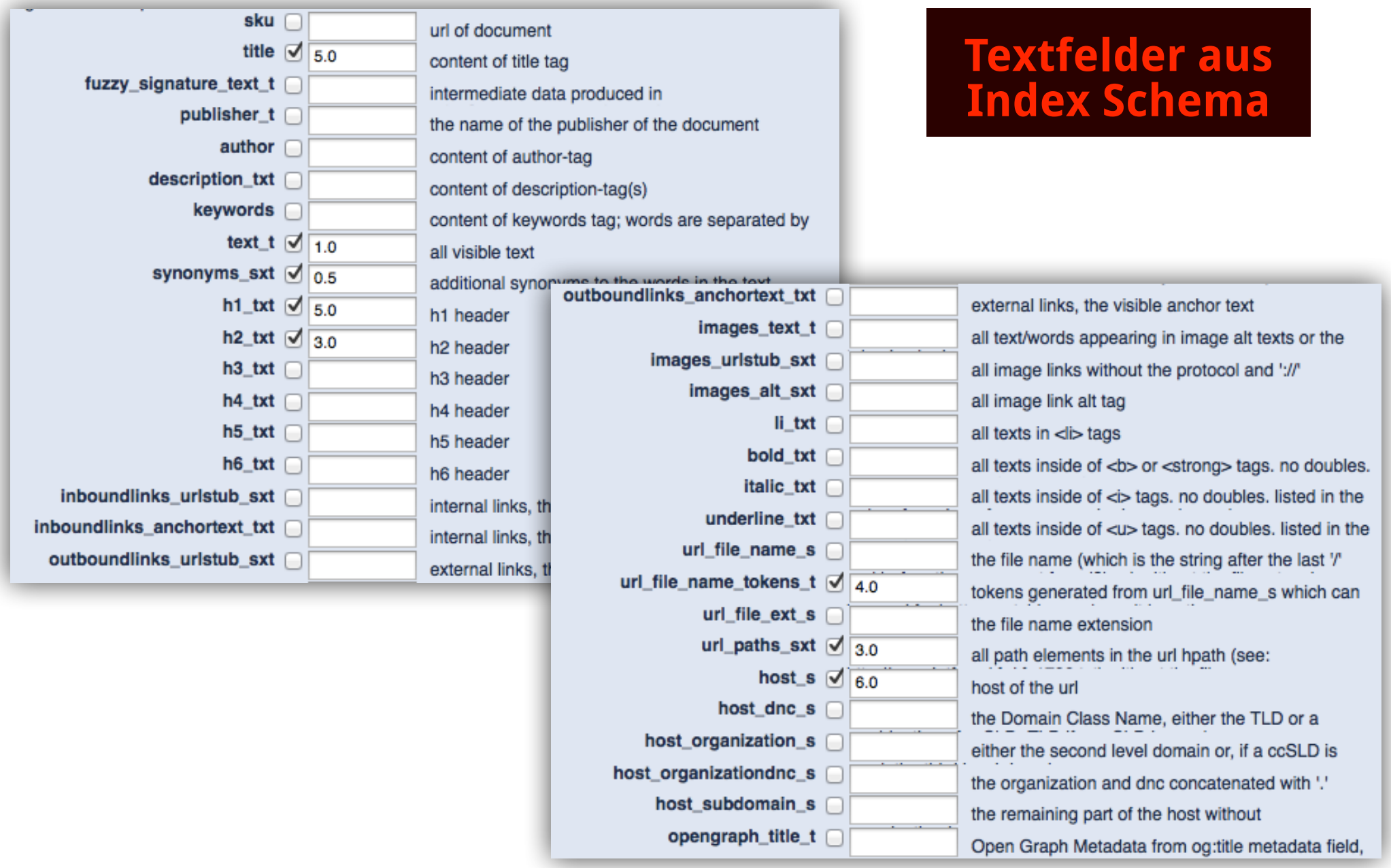
crawler | parser

Index Schema

Apache Solr

☐	title_chars_val	number of characters for each title
☐	title_words_val	number of words in each title
☐	description_count_i	number of descriptions in the document. Its not counting the 'description' field since there is only one. But it counts the number of descriptions that appear in the document (if any)
☐	description_chars_val	number of characters for each description
☐	description_words_val	number of words in each description
☐	h1_i	number of h1 header lines
☐	h2_i	number of h2 header lines
☐	h3_i	number of h3 header lines
☐	h4_i	number of h4 header lines
☐	h5_i	number of h5 header lines
☐	h6_i	number of h6 header lines
☐	schema_org_breadcrumb_i	number of itemprop="breadcrumb" appearances in div tags
☐	opengraph_title_t	Open Graph Metadata from og:title metadata field, see http://ogp.me/ns#
☐	opengraph_type_s	Open Graph Metadata from og:type metadata field, see http://ogp.me/ns#
☐	opengraph_url_s	Open Graph Metadata from og:url metadata field, see http://ogp.me/ns#
☐	opengraph_image_s	Open Graph Metadata from og:image metadata field, see http://ogp.me/ns#
☐	cr_host_count_i	the number of documents within a single host
☐	cr_host_chance_d	the chance to click on this page when randomly clicking on links within on one host
☐	cr_host_norm_i	normalization of chance: 0 for lower halve of cr_host_count_i urls, 1 for 1/2 of the remaining and so on. the maximum number is 10
☐	rating_i	custom rating; to be set with external rating information
☐	bold_val	number of occurrences of texts in bold_txt
☐	italic_val	number of occurrences of texts in italic_txt
☐	underline_val	number of occurrences of texts in underline_txt
☐	ext_cms_txt	names of cms attributes; if several are recognized then they are listen in decreasing order of number of matching criterias
☐	ext_cms_val	number of attributes that count for a specific cms in ext_cms_txt
☐	ext_ads_txt	names of ad-servers/ad-services
☐	ext_ads_val	number of attributes counts in ext_ads_txt
☐	ext_community_txt	names of recognized community functions
☐	ext_community_val	number of attribute counts in attr_community
☐	ext_maps_txt	names of map services
☐	ext_maps_val	number of attribute counts in ext_maps_txt
☐	ext_tracker_txt	names of tracker server
☐	ext_tracker_val	number of attribute counts in ext_tracker_txt
☐	ext_title_txt	names matching title expressions
☐	ext_title_val	number of matching title expressions
☑	vocabularies_sxt	collection of all vocabulary names that have a matcher in the document - use this to boost with vocabularies

Funktionskomponenten einer Suchmaschine



Textfelder aus Index Schema

Funktionskomponenten einer Suchmaschine

`http://localhost:8090/solr/select?q=text_t:ibm%20mainframe%20AND%20url_file_ext_s:pdf&fl=sku,author,publisher_t`

Entwicklung einer eigenen Suchmaschine

crawler | parser

search interface

Apache Solr

Web Search

yacy

ibm mainframe filetype:pdf

3 results from a total of 3 docs in index; search time: 217 milliseconds.

create a download script | all results

remove the filter 'filetype:pdf'

Count	Protocol	Host	Path	URL	Size	Date
1	https	www.dbsystel.de	/file/3275180 /data/	https://www.dbsystel.de/file/3275180 /data/bereitstellung_betrieb_von_ibm-mainframe-kapazitaeten_englisch.pdf	2 mbyte	Mon, 05 Aug 2013 08:14:04
2	https	www.dbsystel.de	/file/2239284 /data/	https://www.dbsystel.de/file/2239284 /data/bereitstellung_betrieb_von_ibm-mainframe-kapazitaeten_deutsch.pdf	2 mbyte	Wed, 16 Mar 2011 16:36:45
3	https	www.dbsystel.de	/file/2247366 /data/	https://www.dbsystel.de/file/2247366 /data/bereitstellung_betrieb_von_ibm-mainframe-kapazitaeten_englisch.pdf	2 mbyte	Tue, 04 Oct 2011 07:04:25

Based Trai [..]
http://ma-co.de/
Sat, 21 Sep 2013 | Metadata | Parser | Citations |

Management- und Personalberatung van der Zalm Hamburg
[..] Suche Direktansprache MANAGEMENT Potenzialanalyse Organisationsentwicklung **Personalentwicklung** Vergütungsmanagement TRAINING & COACHING. [..]
http://www.vanderzalm.de/index.php?kat=997
Sat, 21 Sep 2013 | Metadata | Parser | Citations |

Preise und Auszeichnungen für EF Englishtown
Jedes Jahr werden ausgewählte Personalabteilungen und Trainingsanbieter vom HRE für ihr herausragendes Personalmanagement und vorbildliche **Personalentwicklung** geehrt
http://www.englishtown.de/online/awards.aspx
Sat, 21 Sep 2013 | Metadata | Parser | Citations |

mpmEXPERTS - Wir machen MultiProjektmanagement. Einfach. Sicher.
Personalentwicklung, Talanx Service AG &100% professionell, hervorragende

</result>
</response>

- dbsystel.de (9)
- stadtbranche.de (5)
- mpm-experts.com (2)
- d-nb.info (1)
- berlin.city-map.de (1)
- englishtown.de (1)
- de.wikipedia.org (1)
- lebensmittel-verzeichnis.de (1)
- central.de (1)
- nammert24.de (1)
- ma-co.de (1)
- vanderzalm.de (1)

Studies Navigator

- Seminars (1)

Fu

Host Browser

Browse the index of 20,593,544 documents.

Host/URL: http://www.dbsystel.de/dbsystel/ Browse Host

Browser for http://www.dbsystel.de/dbsystel/

	Path	stored	linked	pending	excluded	failed
	..					
	http://www.dbsystel.de/dbsystel/ansprechpartner/	5	0	0	0	0
	http://www.dbsystel.de/dbsystel/jobs_karriere/	62	0	0	0	0
	http://www.dbsystel.de/dbsystel/kernkompetenzen/	6	0	0	0	0
	http://www.dbsystel.de/dbsystel/leistungsspektrum/	8	0	0	0	0
	http://www.dbsystel.de/dbsystel/start/	14	0	0	0	0

Browser for http://www.dbsystel.de/dbsystel/jobs_karriere/

	Path	stored	linked
	..		
	http://www.dbsystel.de/dbsystel/jobs_karriere/ihr_einstieg_und_aktuelle_jobs/	18	0
	http://www.dbsystel.de/dbsystel/jobs_karriere/messen_und_veranstaltungen/	4	0
	http://www.dbsystel.de/dbsystel/jobs_karriere/rund_um_ihre_bewerbung/	5	0
	http://www.dbsystel.de/dbsystel/jobs_karriere/standorte/	25	0
	http://www.dbsystel.de/dbsystel/jobs_karriere/warum_db_systel/	9	0
	http://www.dbsystel.de/dbsystel/jobs_karriere/	indexed	clickdepth: 999, cr=0.004, crn=8, refs: 0 internal referrer: ⓘⓘⓘⓘⓘⓘⓘⓘⓘⓘⓘⓘⓘⓘⓘⓘⓘⓘⓘⓘⓘⓘⓘⓘⓘ ⓘⓘⓘⓘⓘⓘⓘⓘⓘⓘⓘⓘⓘⓘⓘ

Browser for http://www.dbsystel.de/dbsystel/ansprechpartner/

documents stored for host: 5; documents stored for subpath: 5; unloaded documents detected in subpath: 0

	Path	stored	linked	pendi
	..			
	http://www.dbsystel.de/dbsystel/ansprechpartner/	indexed	clickdepth: 999, cr=0.004, crn=10, refs: 0 hosts, 0 ext, 96 in internal referrer: ⓘⓘⓘⓘⓘⓘⓘⓘⓘⓘⓘⓘⓘⓘⓘⓘⓘⓘⓘⓘⓘⓘⓘⓘⓘⓘⓘⓘⓘⓘⓘⓘⓘⓘⓘⓘⓘ ⓘⓘⓘⓘⓘⓘⓘⓘⓘⓘⓘⓘⓘⓘⓘ	
	http://www.dbsystel.de/dbsystel/ansprechpartner/kommunikation.html	indexed	clickdepth: 999, cr=0.001, crn=2, refs: 0 hosts, 0 ext, 7 int internal referrer: ⓘⓘⓘⓘⓘⓘⓘ	
	http://www.dbsystel.de/dbsystel/ansprechpartner/kundenservicecenter.html	indexed	clickdepth: 999, cr=0.001, crn=2, refs: 0 hosts, 0 ext, 6 int internal referrer: ⓘⓘⓘⓘⓘⓘ	
	http://www.dbsystel.de/dbsystel/ansprechpartner/vertrieb.html	indexed	clickdepth: 999, cr=0.001, crn=2, refs: 0 hosts, 0 ext, 7 int internal referrer: ⓘⓘⓘⓘⓘⓘⓘ	
	http://www.dbsystel.de/dbsystel/ansprechpartner/zentrale.html	indexed	clickdepth: 999, cr=0.001, crn=2, refs: 0 hosts, 0 ext, 6 int internal referrer: ⓘⓘⓘⓘⓘⓘ	

h load errors.

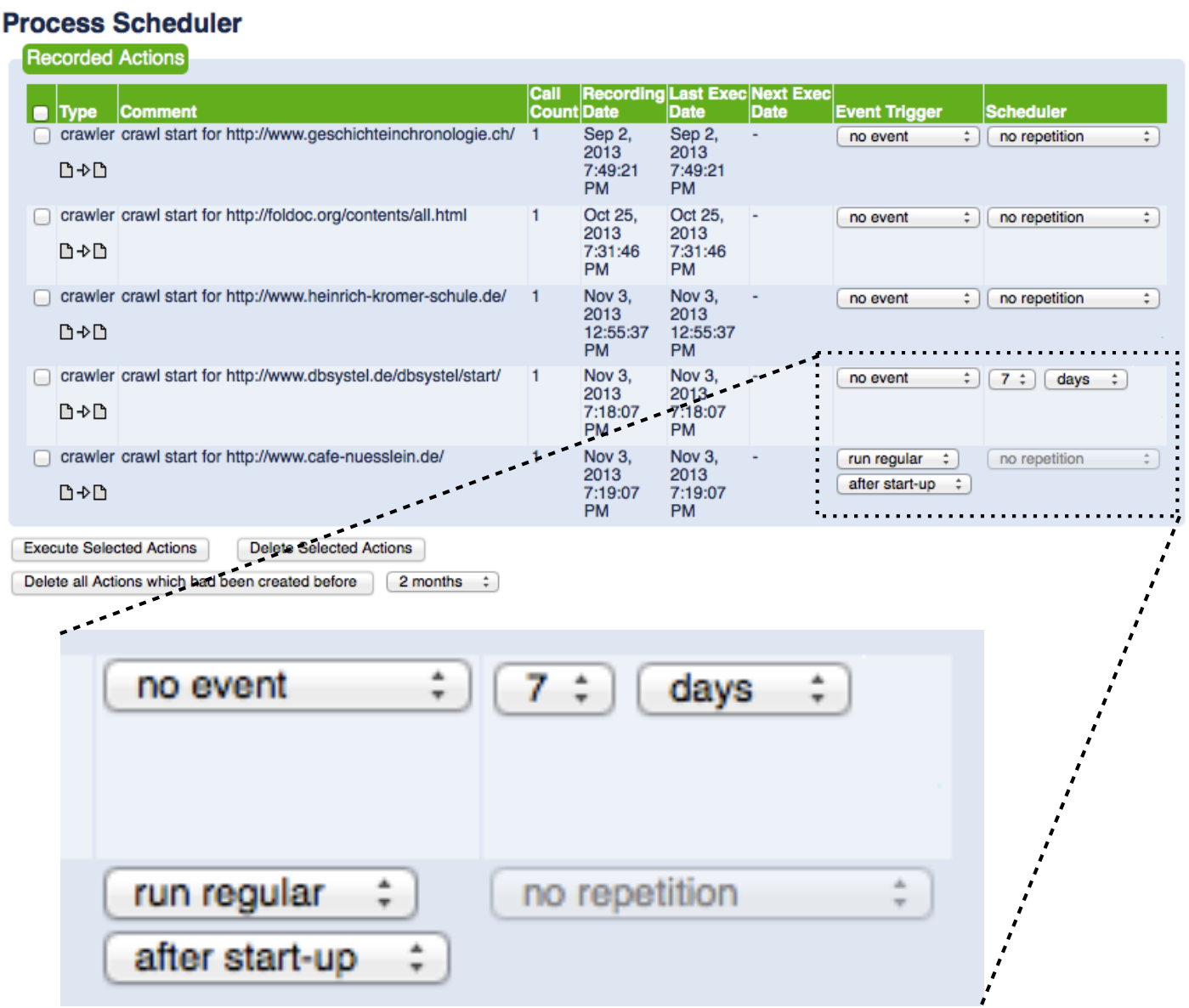
284577/0/25 URLs
199646/1/1 URLs
134104 URLs
109339 URLs
82411 URLs
80795 URLs
68936 URLs
66522 URLs
59525/0/5 URLs
58273 URLs
54271 URLs
46964/0/83 URLs
43381 URLs
42855 URLs
38587 URLs
36301/0/62 URLs
36250 URLs
33713 URLs
32681 URLs
32269 URLs
31837 URLs
28931/0/31 URLs
27546 URLs
26171/0/7 URLs
25343 URLs
23708 URLs
23370 URLs
22682 URLs
22242 URLs
21934 URLs
21447 URLs
21080 URLs
20857/0/56 URLs
20295 URLs
20035 URLs
19650 URLs
19324 URLs
18664 URLs
18202/0/17 URLs
17930/0/1 URLs
17369 URLs
17260 URLs
17195 URLs
17108/0/1 URLs
16854 URLs
16598 URLs
16154 URLs
15948 URLs
15399 URLs
15155 URLs
14878/0/5 URLs
14736 URLs
14598 URLs

K S C S

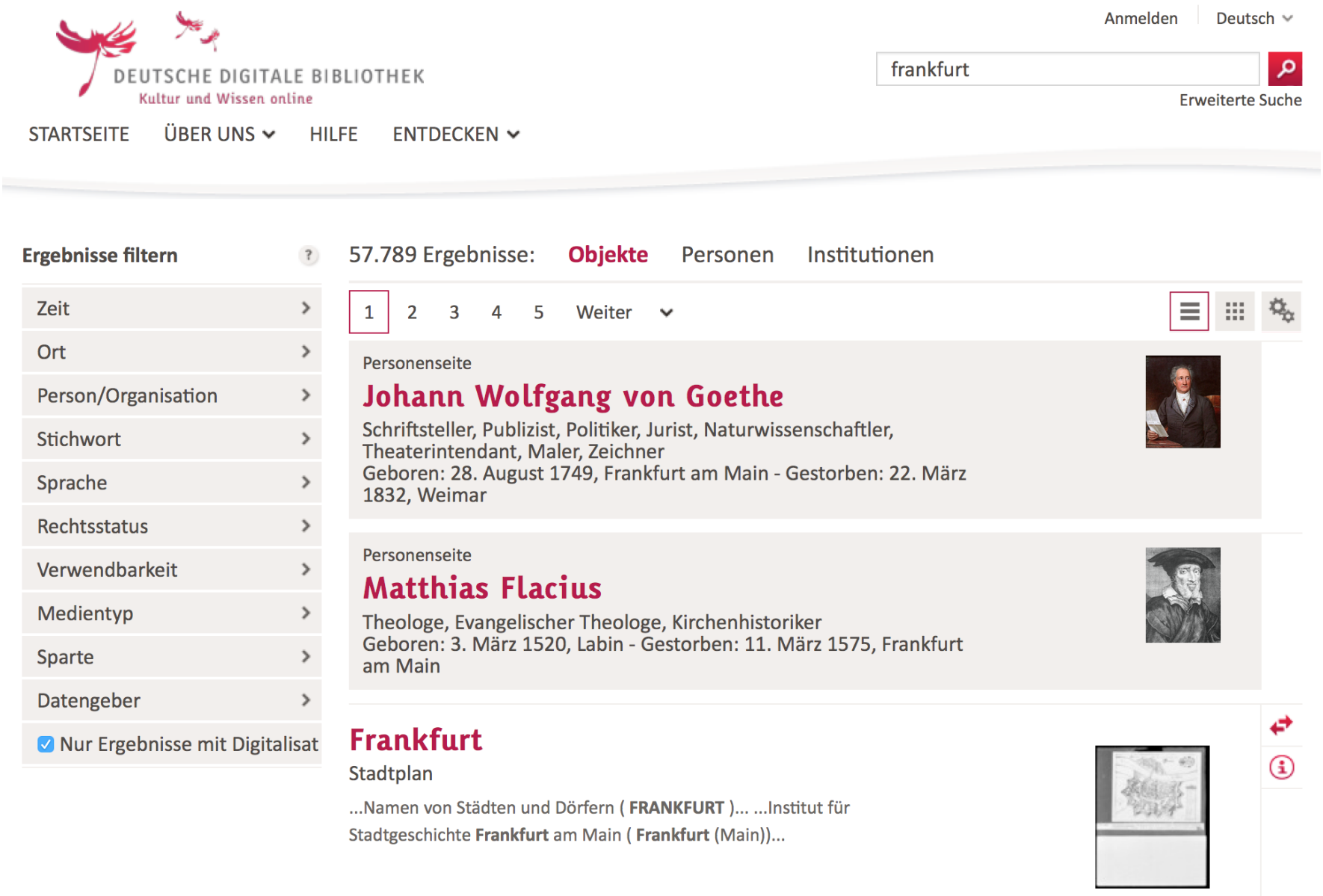
Funktionskomponenten einer Suchmaschine

Komponenten der Suchmaschine

- crawler
- parser
- search interface
- monitoring
- administration
- Apache Solr



Semantische Suchmaschinen - Deutsche Digitale Bibliothek



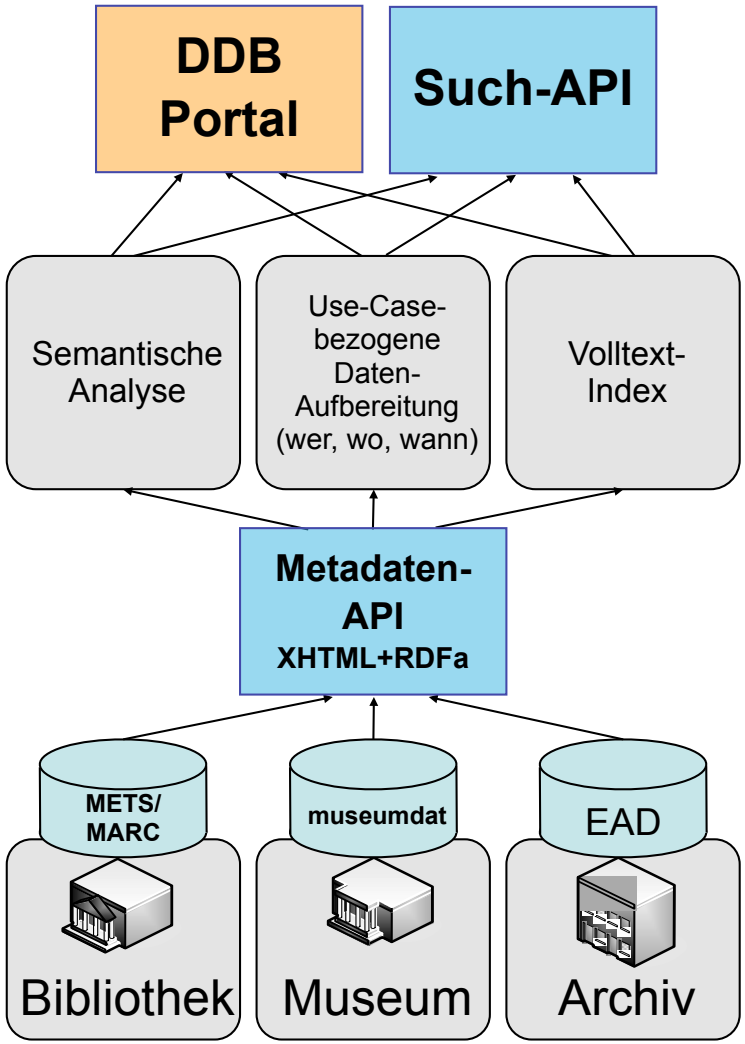
https://www.deutsche-digitale-bibliothek.de/

Semantische Suchmaschinen - Deutsche Digitale Bibliothek

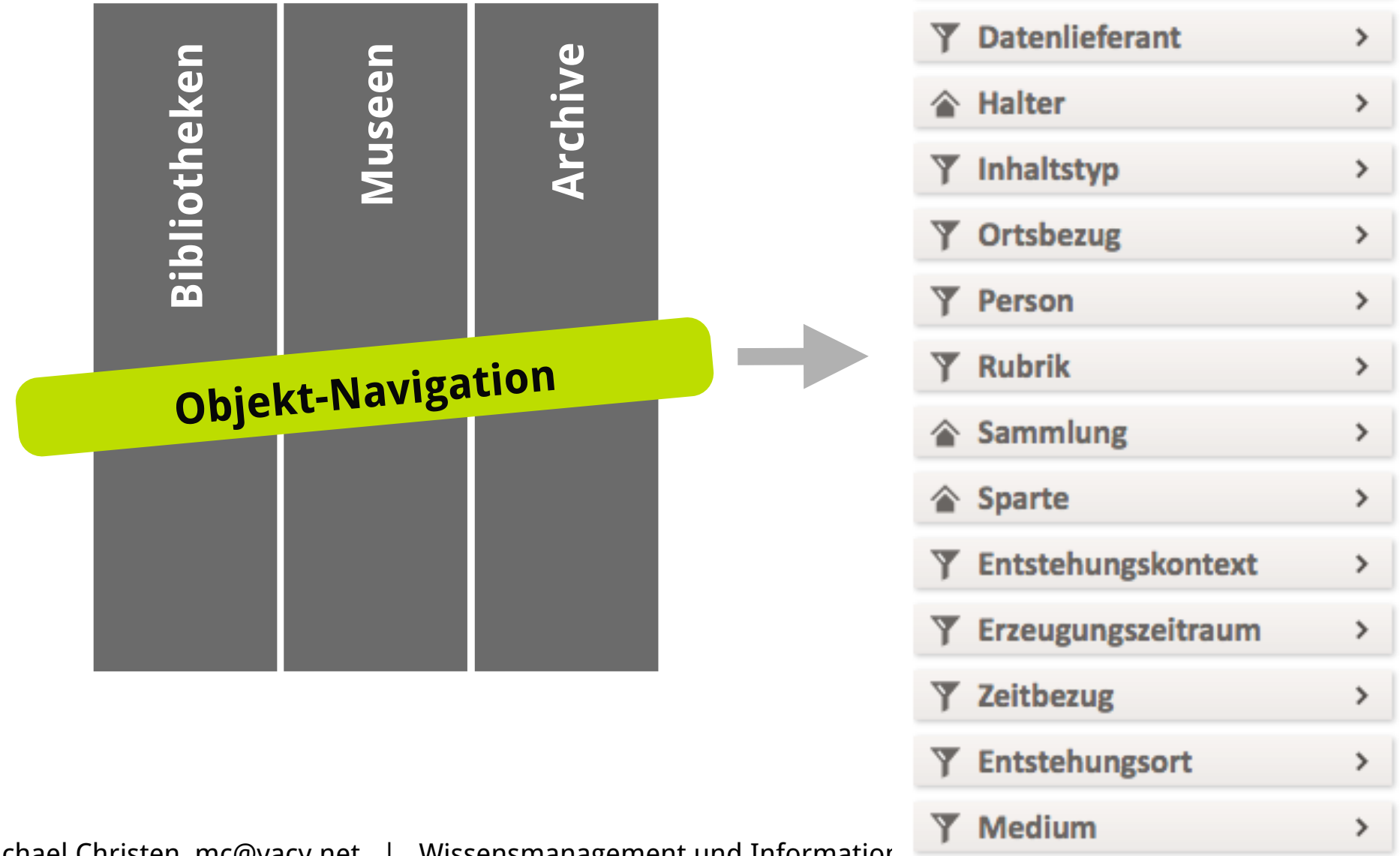
Präsentation und Benutzerinteraktion

Daten-Analyse und Suchfunktionen

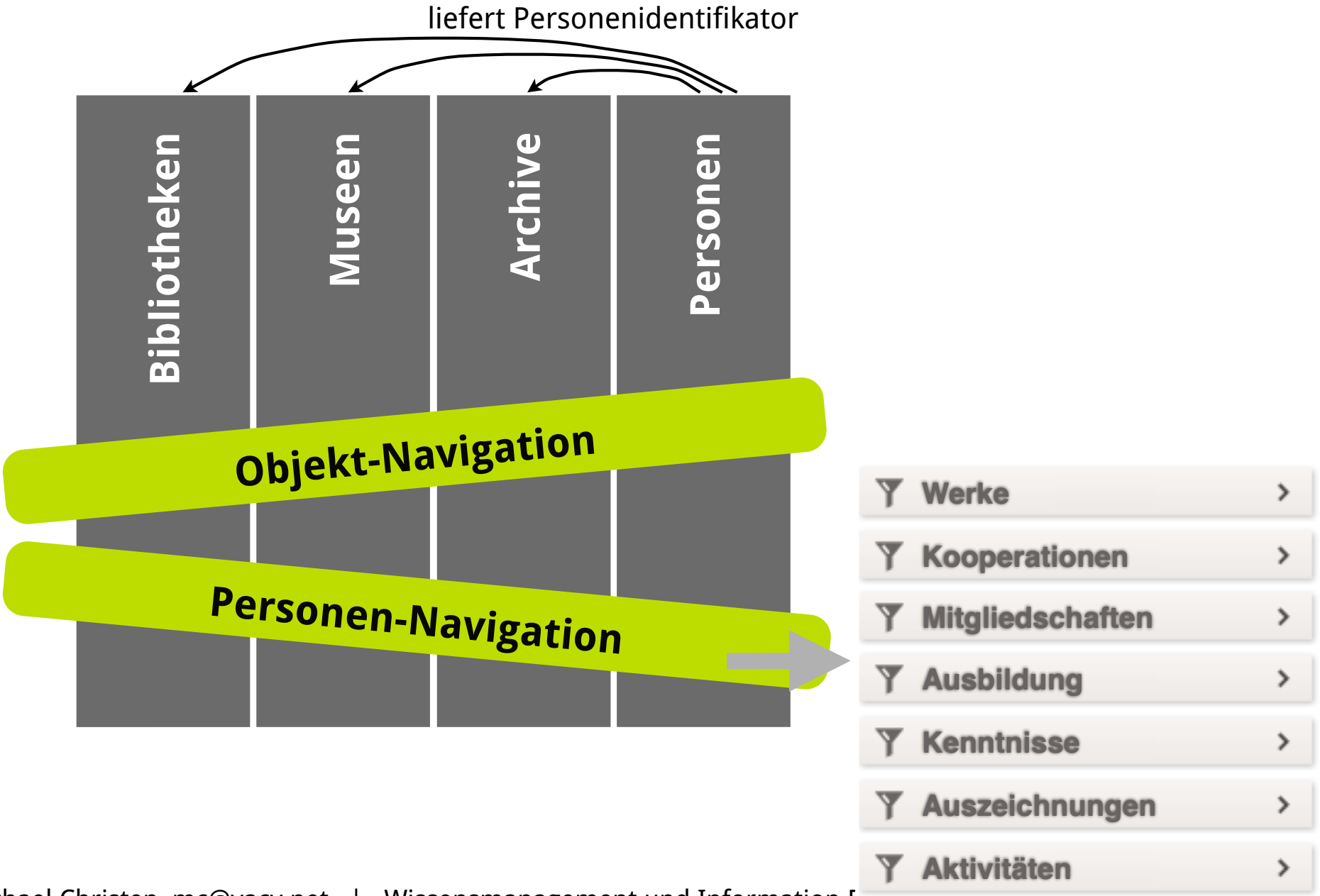
Harvesting, Speicherung und Konsolidierung



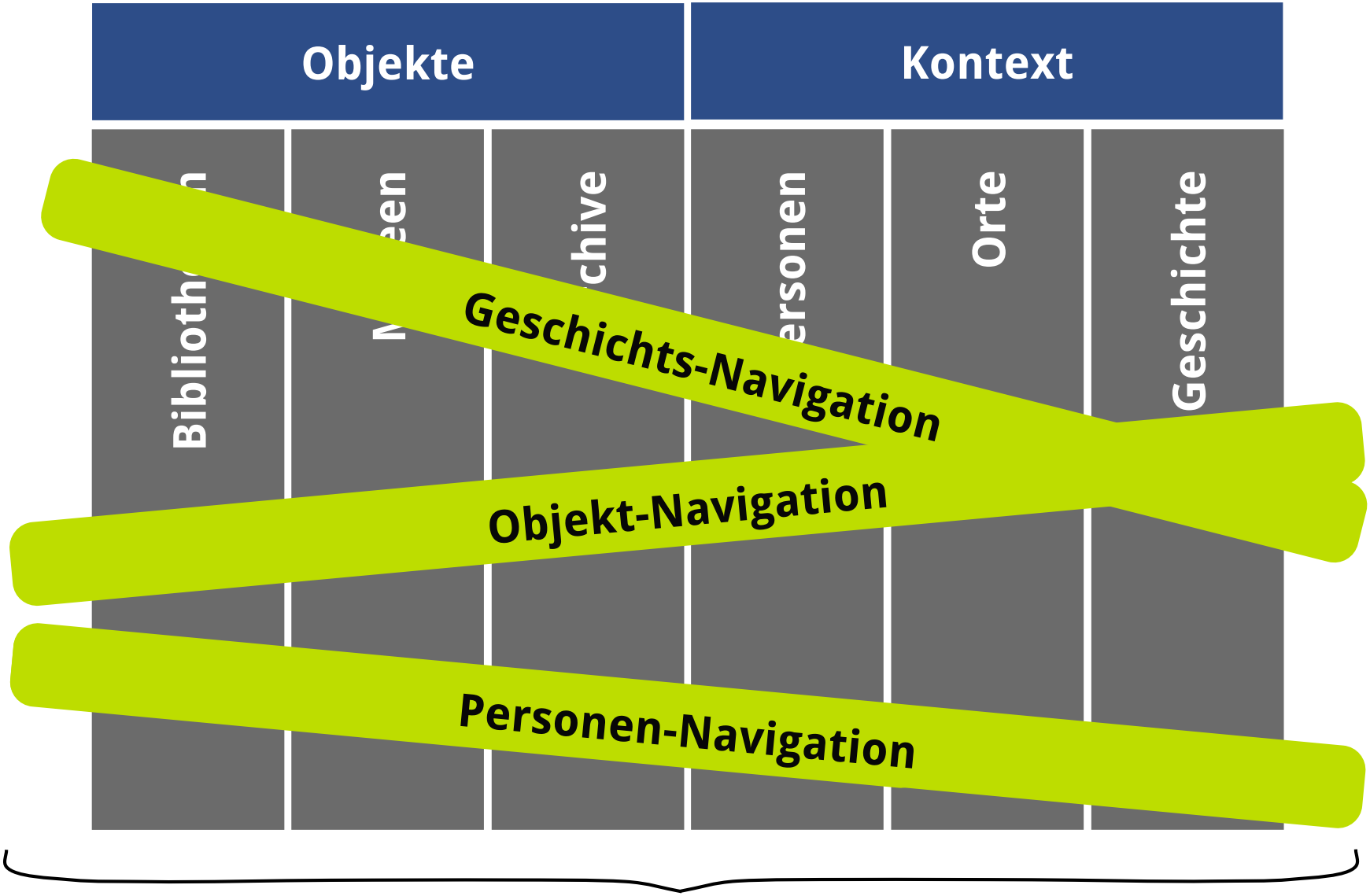
Semantische Suchmaschinen - Deutsche Digitale Bibliothek



Semantische Suchmaschinen - Deutsche Digitale Bibliothek

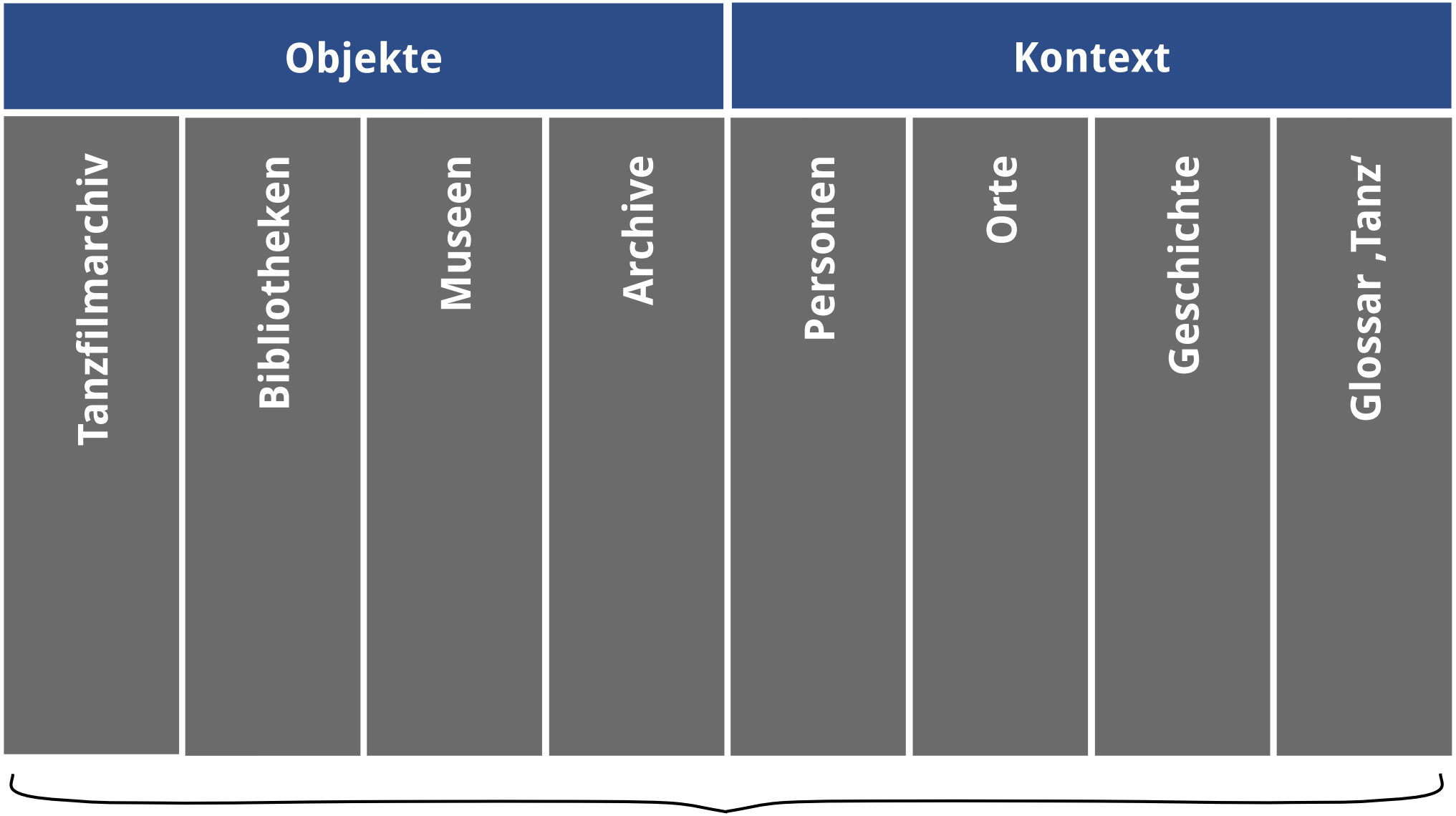


Semantische Suchmaschinen - Deutsche Digitale Bibliothek



durch eine Einbeziehung des Kontextes in die Navigation
erlangt ein Objekt eine Bedeutung in diesem Kontext

Semantische Suchmaschinen - Deutsche Digitale Bibliothek

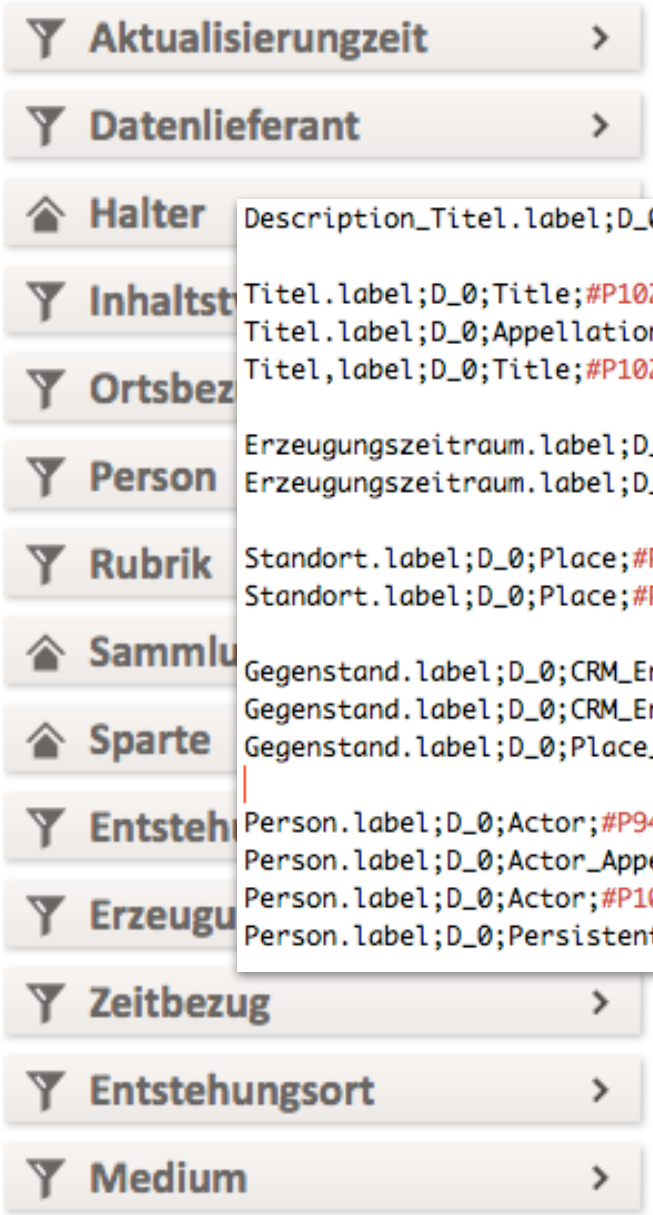


eine Erweiterung der Menge der Objekt-Typen kann durch Hinzunahme von Kontextwissen dem allgemeinen Verständnis der Objekte dienen

Semantische Suchmaschinen - Deutsche Digitale Bibliothek

DDB: Objekte

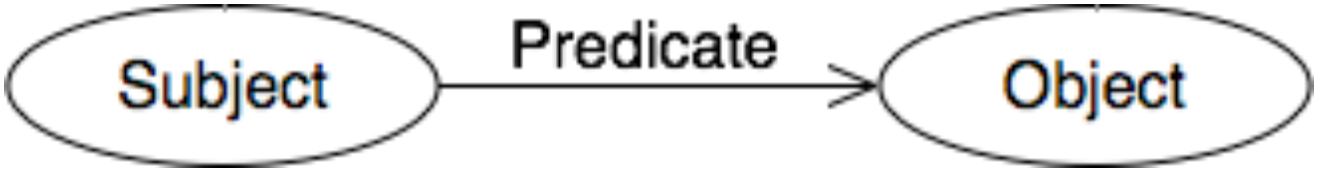
Objektidentifizierung: hash, z.B.
http://www.ddb.de/DXEGS4J62TWKR6VSPUPLA3S26UZOSTKL




```
Description_Titel.label;D_0;Title;#P3F.has_note,#P102F.has_title,label

Titel.label;D_0;Title;#P102F.has_title,label
Titel.label;D_0;Appellation;#P1F.is_identified_by,label
Titel,label;D_0;Title;#P102F.has_title,http://purl.org/dc/elements/1.1/title

Erzeugungszeitraum.label;D_0;Time-Span;#P108B.was_produced_by,#P4F.has_time-span,label
Erzeugungszeitraum.label;D_0;Time-Span;#P94B.was_created_by,#P4F.has_time-span,label

Standort.label;D_0;Place;#P53F.has_former_or_current_location,label
Standort.label;D_0;Place;#P52F.has_current_owner,#P74F.has_current_or_former_residence,label

Gegenstand.label;D_0;CRM_Entity;#P129F.is_about,label
Gegenstand.label;D_0;CRM_Entity;#P138F.represents,label
Gegenstand.label;D_0;Place_Appellation;#P138F.represents,label

Person.label;D_0;Actor;#P94B.was_created_by,#P14F.carried_out_by,label
Person.label;D_0;Actor_Appellation;#P94B.was_created_by,#P14F.carried_out_by,#P131F.is_identified_by,label
Person.label;D_0;Actor;#P108B.was_produced_by,#P14F.carried_out_by,label
Person.label;D_0;Persistent_Item;#P12F.occurred_in_the_presence_of, #P14F.carried_out_by,label
```

Semantische Suchmaschinen - RDF Datenstrukturen

- Framework um Information und Relationen von Objekten auszudrücken
- Darstellung als Triple



- Beispiel:

```
<http://sws.geonames.org/2925533/>
   <http://www.geonames.org/ontology#wikipediaArticle>
   <http://zu.wikipedia.org/wiki/Frankfurt> .

<http://sws.geonames.org/2925533/>
   <http://www.w3.org/2000/01/rdf-schema#seeAlso>
   <http://dbpedia.org/resource/Frankfurt_am_main> .

<http://sws.geonames.org/2925534/>
   <http://www.geonames.org/ontology#alternateName>
   "Kreisfreie Stadt Frankfurt" .

<http://sws.geonames.org/2925534/>
   <http://www.geonames.org/ontology#alternateName>
   "Stadtkreis Frankfurt" .
```

```
# http://www.w3.org/TR/rdf11-concepts/
```

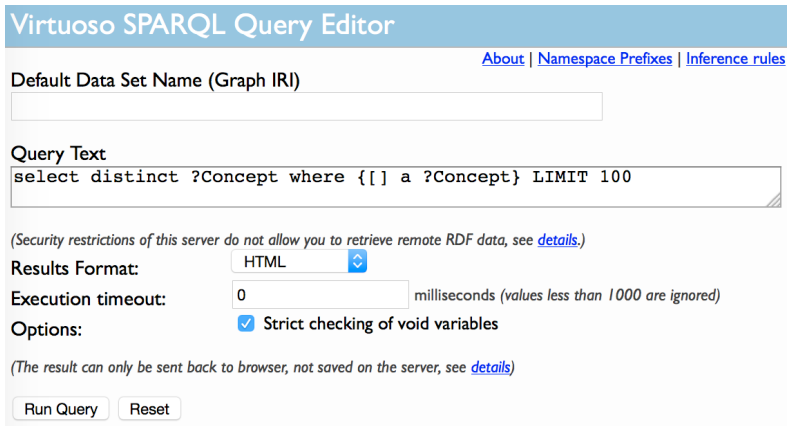
Semantische Suchmaschinen - SPARQL / RDF Queries

- Die Anfrage im folgenden Beispiel findet die Namen aller afrikanischen Hauptstädte und das Land, in dem sich die jeweilige Hauptstadt befindet.

```
PREFIX abc: <http://example.com/exampleOntology#>
SELECT ?capital ?country
WHERE {
  ?x abc:cityname ?capital ;
     abc:isCapitalOf ?y .
  ?y abc:countryname ?country ;
     abc:isInContinent abc:Africa .
}
```

```
# http://de.wikipedia.org/wiki/SPARQL
```

- Testen: http://de.dbpedia.org/sparql



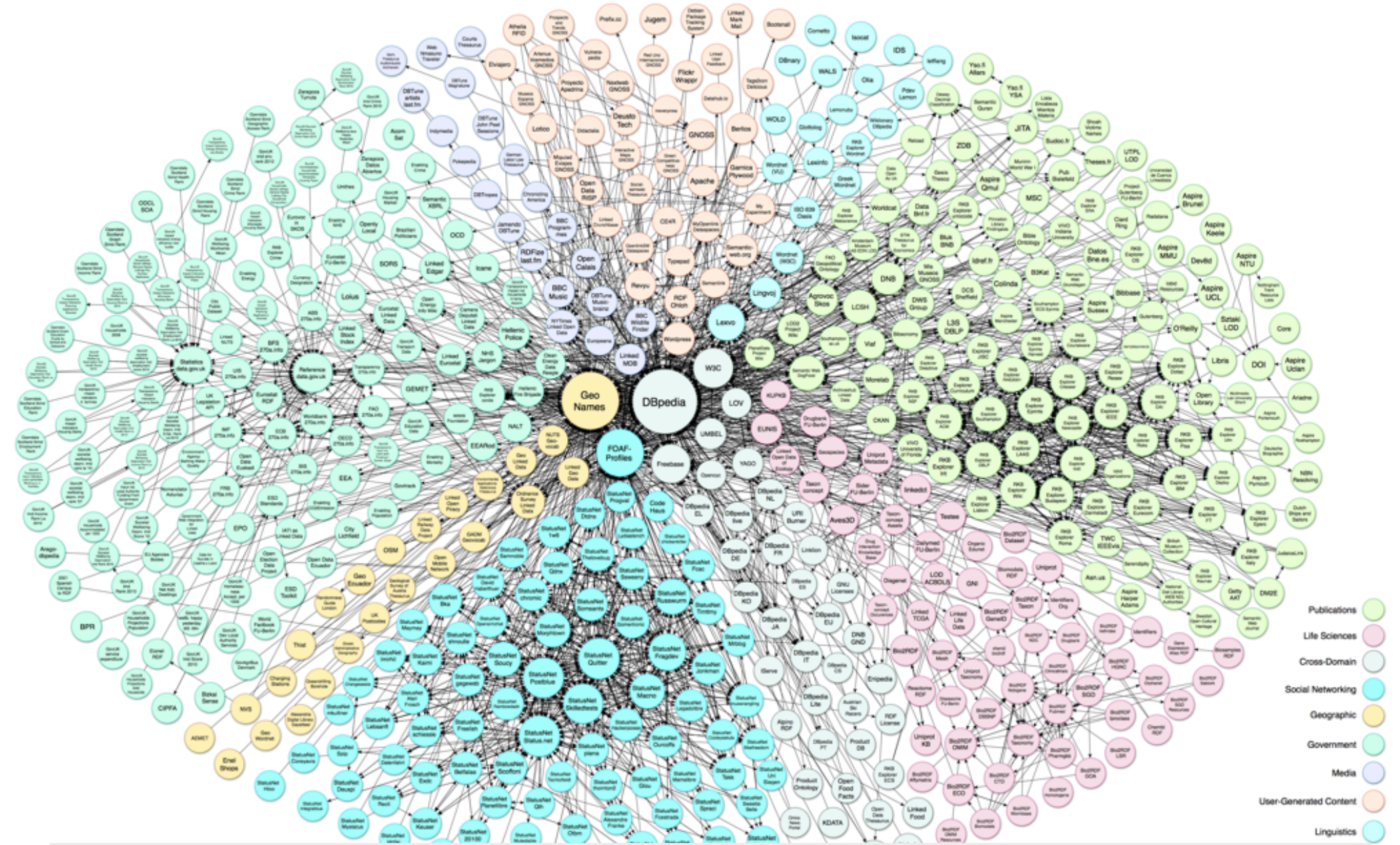
Semantische Suchmaschinen - RDF Datenstrukturen

- eine geparste Version der Wikipedia(s) gibt es als RDF Datenbanken

dbpedia-owl:administrativeDistrict	dbpedia:Darmstadt
dbpedia-owl:areaCode	069, 06109, 06101
dbpedia-owl:areaTotal	248310000.000000 (xsd:double)
dbpedia-owl:country	dbpedia:Germany
dbpedia-owl:elevation	112.000000 (xsd:double)
dbpedia-owl:federalState	dbpedia:Hesse
dbpedia-owl:foundingYear	0001-01-01 (xsd:date)
dbpedia-owl:leaderName	dbpedia:Peter_Feldmann
dbpedia-owl:leaderTitle	Lord Mayor
dbpedia-owl:populationAsOf	2011-09-30 (xsd:date)
dbpedia-owl:populationMetro	5600000 (xsd:integer)
dbpedia-owl:populationTotal	695624 (xsd:integer)
dbpedia-owl:populationUrban	2500000 (xsd:integer)
dbpedia-owl:postalCode	60001–60599, 65901–65936

```
# http://dbpedia.org/page/Frankfurt
```

Semantische Suchmaschinen - RDF Datenstrukturen



http://lod-cloud.net/

Semantische Suchmaschinen - schema.org

- feststehendes Schema für Prädikate
- Einbindung über Mikroformats

```
<div itemscope itemtype="http://schema.org/Event">
  <div itemprop="name">Spinal Tap</div>
  <span itemprop="description">One of the loudest bands ever
  reunites for an unforgettable two-day show.</span>
  Event date:
  <time itemprop="startDate" datetime="2011-05-08T19:30">May 8, 7:30pm</time>
</div>
```

- Beispiel: popula.de

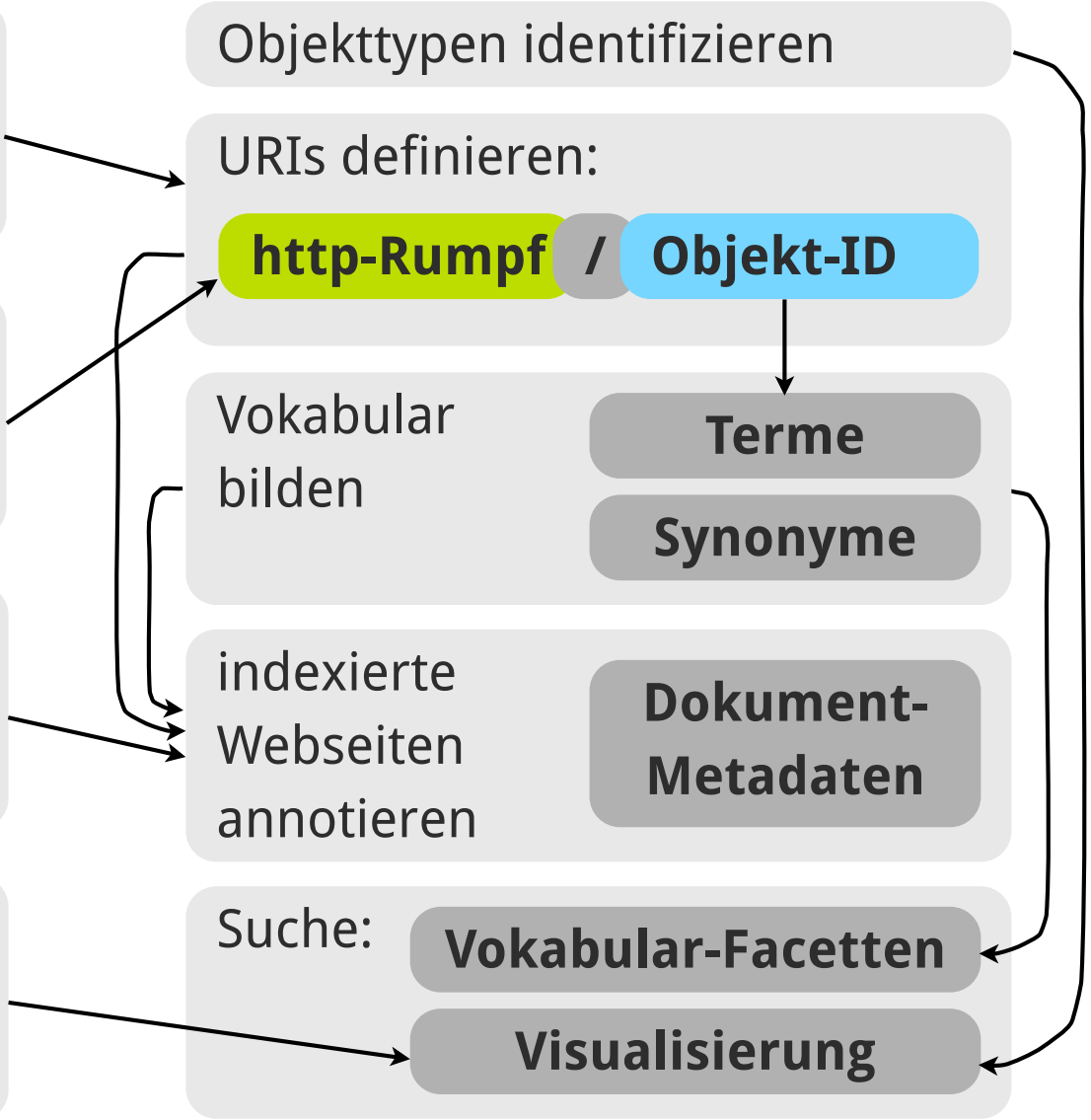
```
# http://schema.org/docs/gs.html
```

Linked Open Data -Konzept

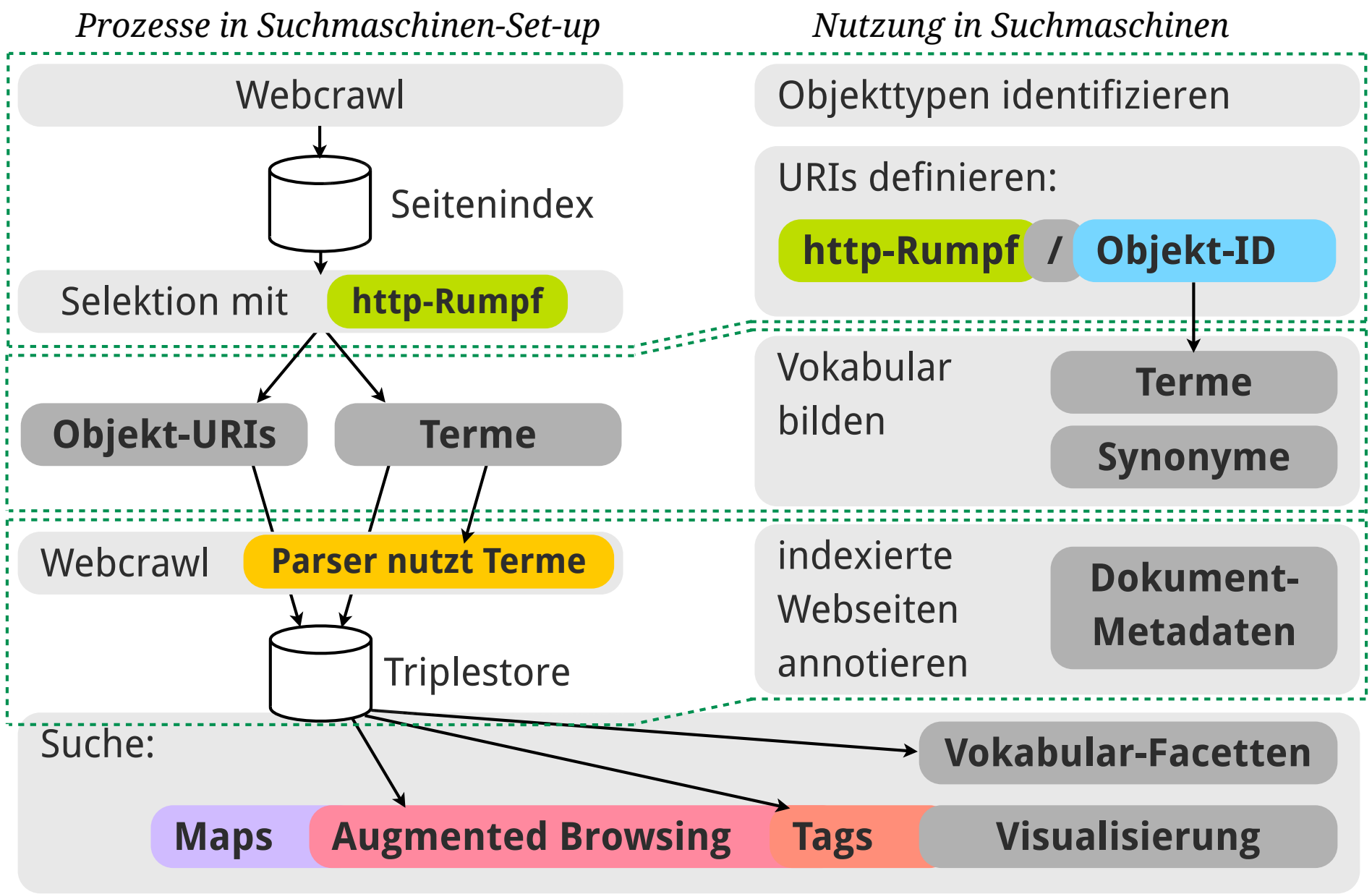
Linked Open Data Konzept

1. URIs als Namen für Dinge benutzen.
2. URIs sollen HTTP URIs sein. Dadurch sind Information leicht abrufbar.
3. Standards/Ontologien zur Ablage benutzen, z.B. RDF + Triplestores.
4. Referenzen zu weiteren URIs integrieren, um mehr zu entdecken.

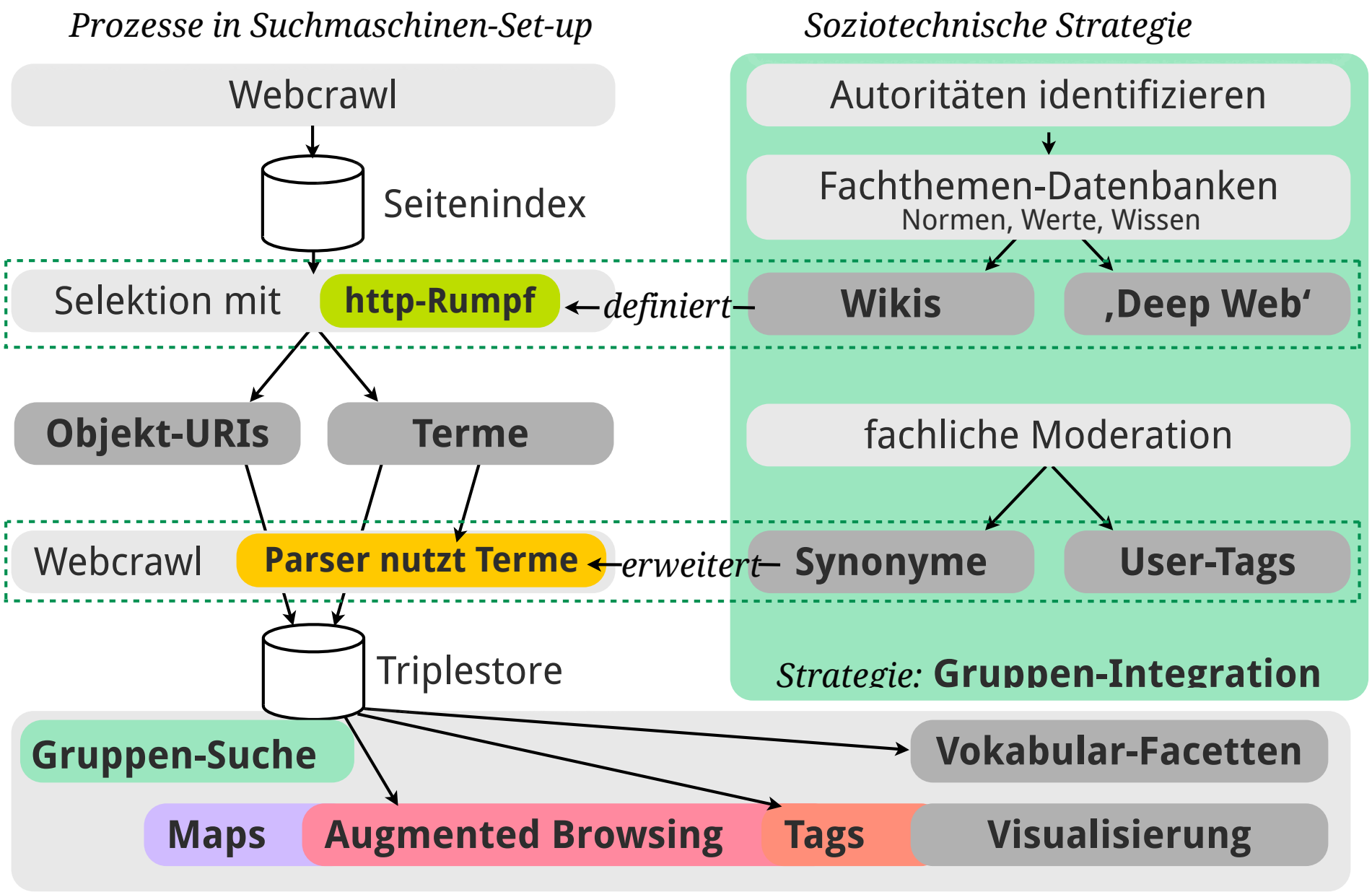
Nutzung in Suchmaschinen



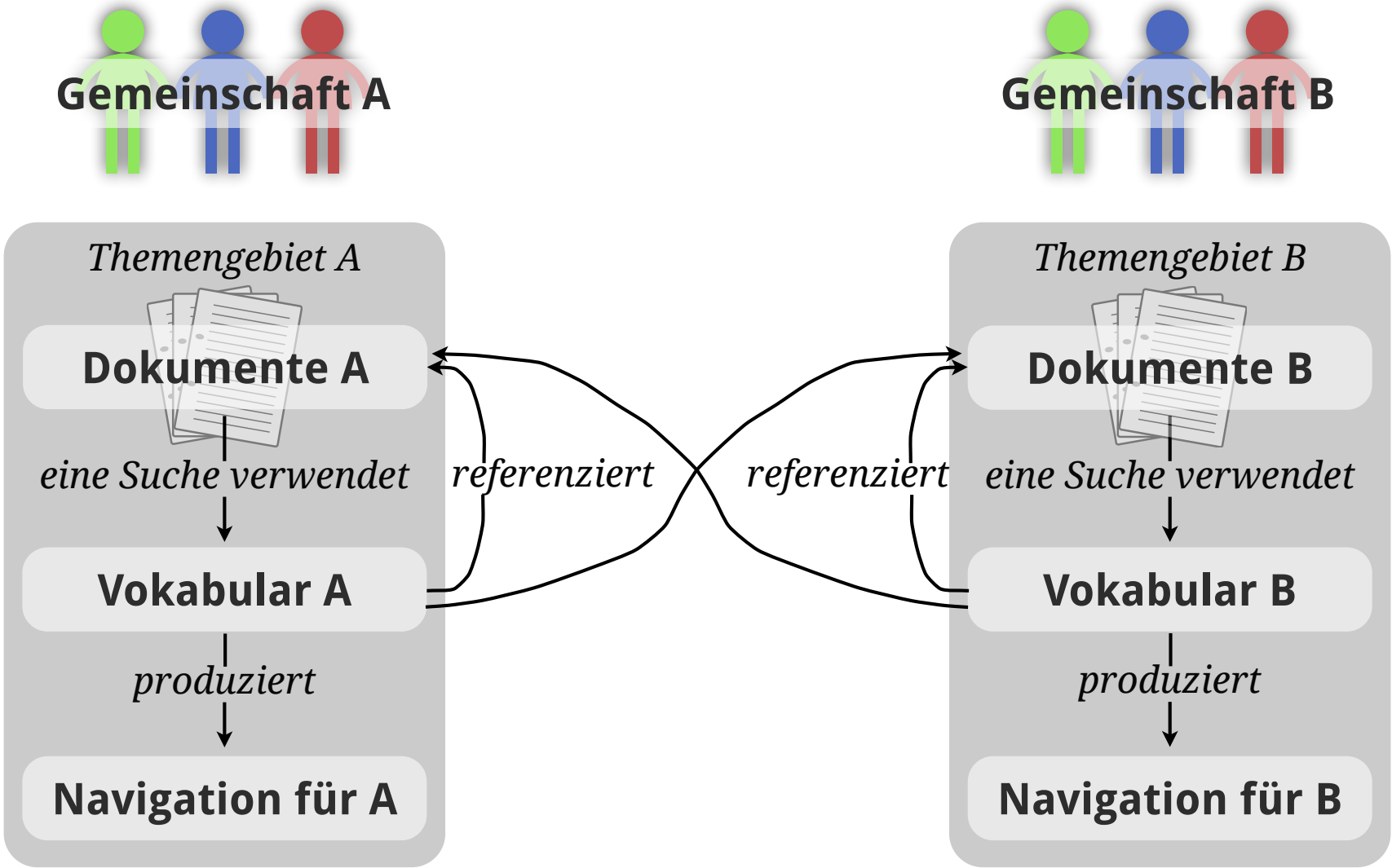
Linked Open Data - Automatisierung



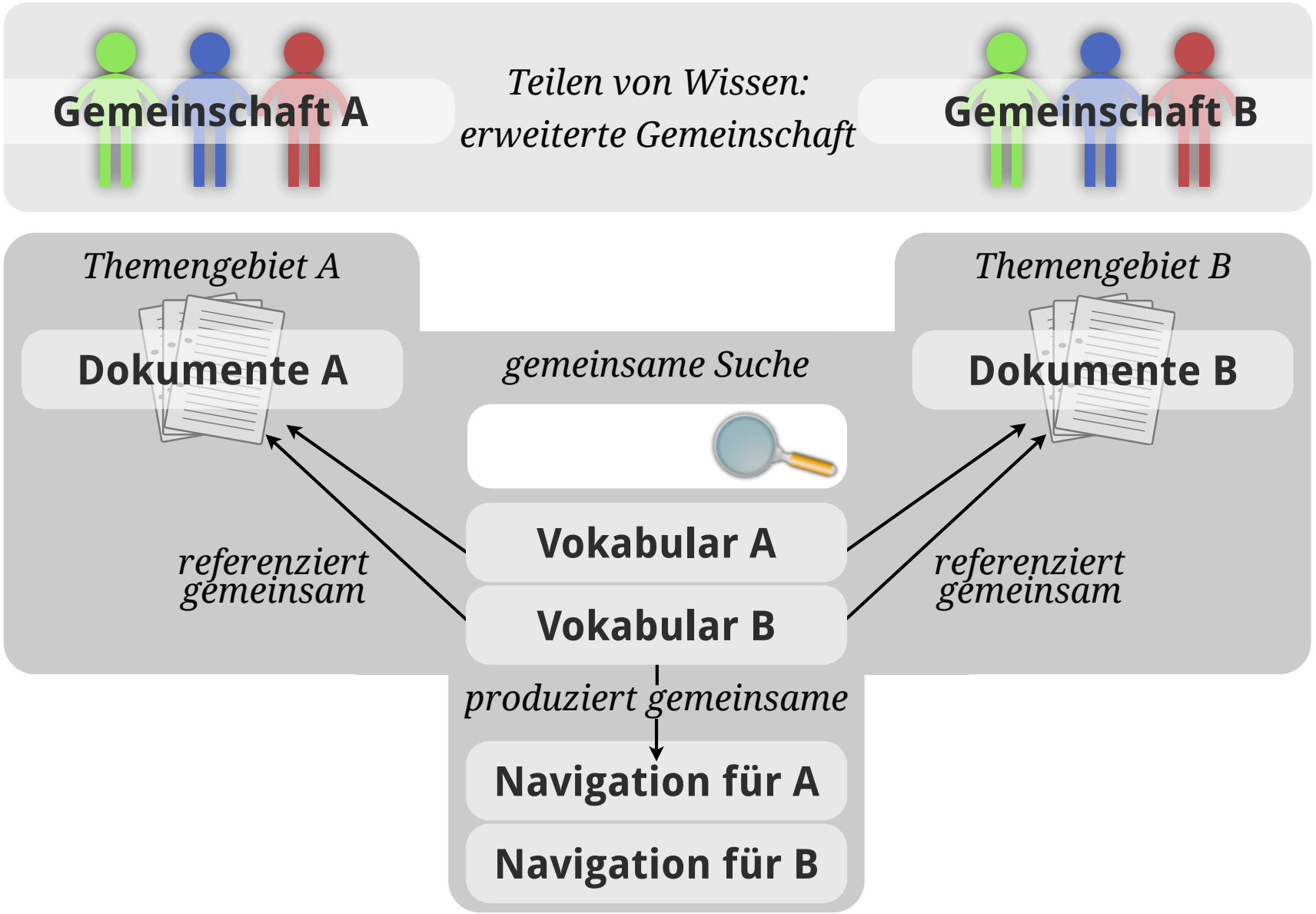
Linked Open Data - Automatisierung



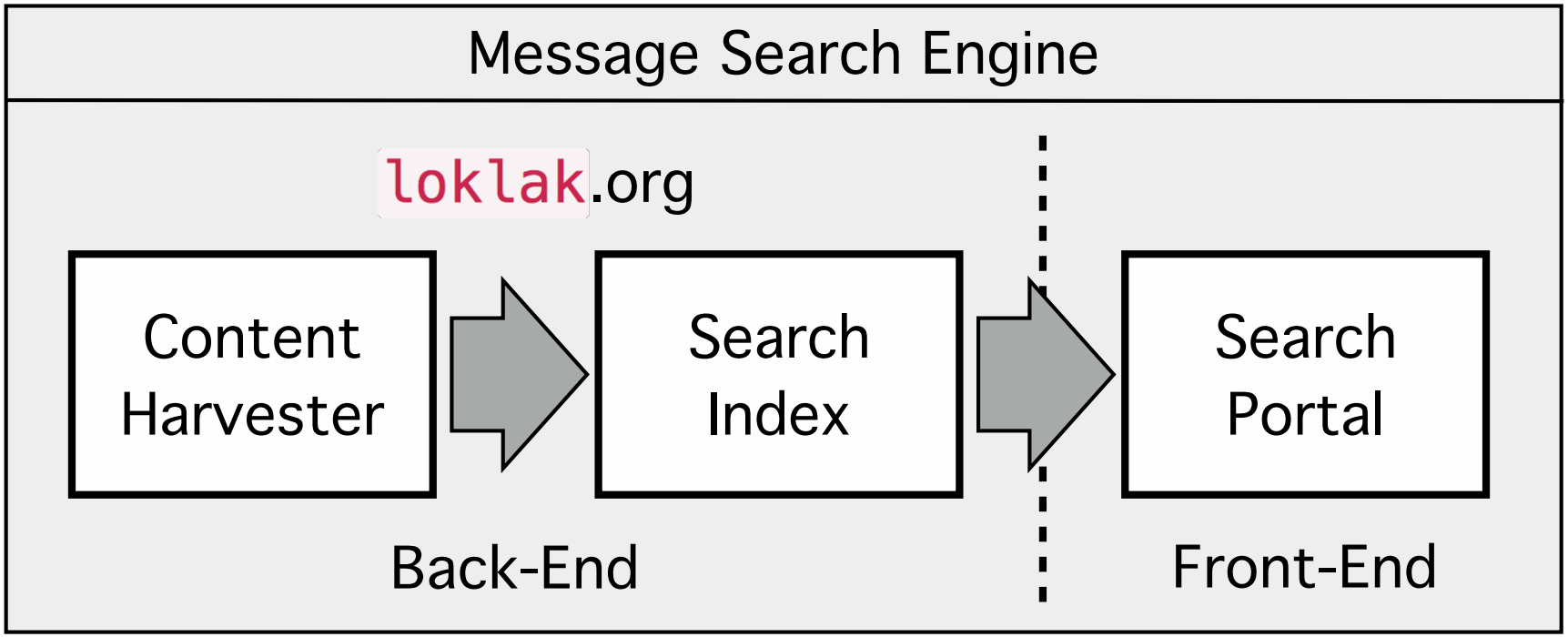
Linked Open Data - gemeinsame Vokabularien



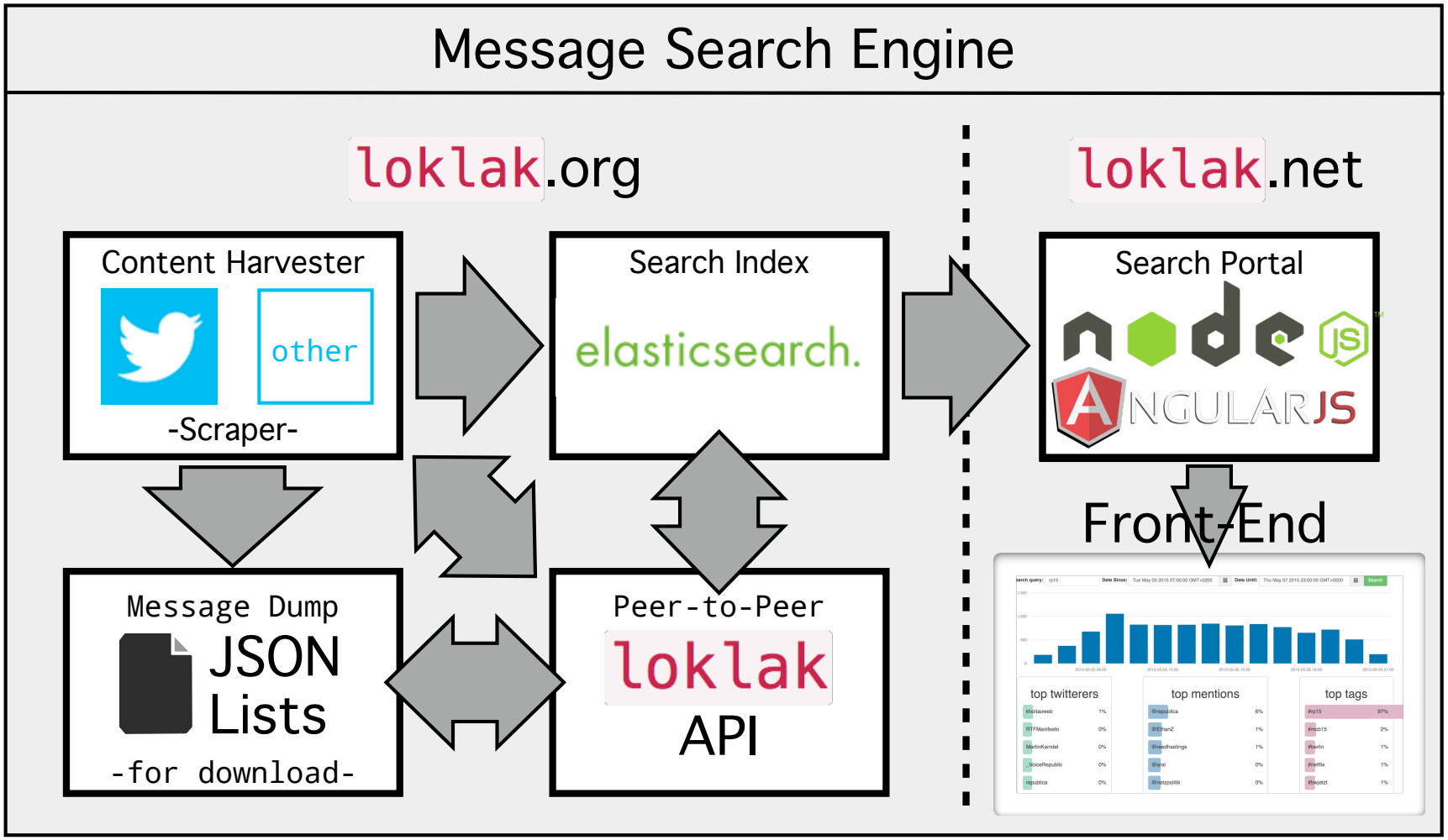
Linked Open Data - Vernetzung von Gemeinschaften



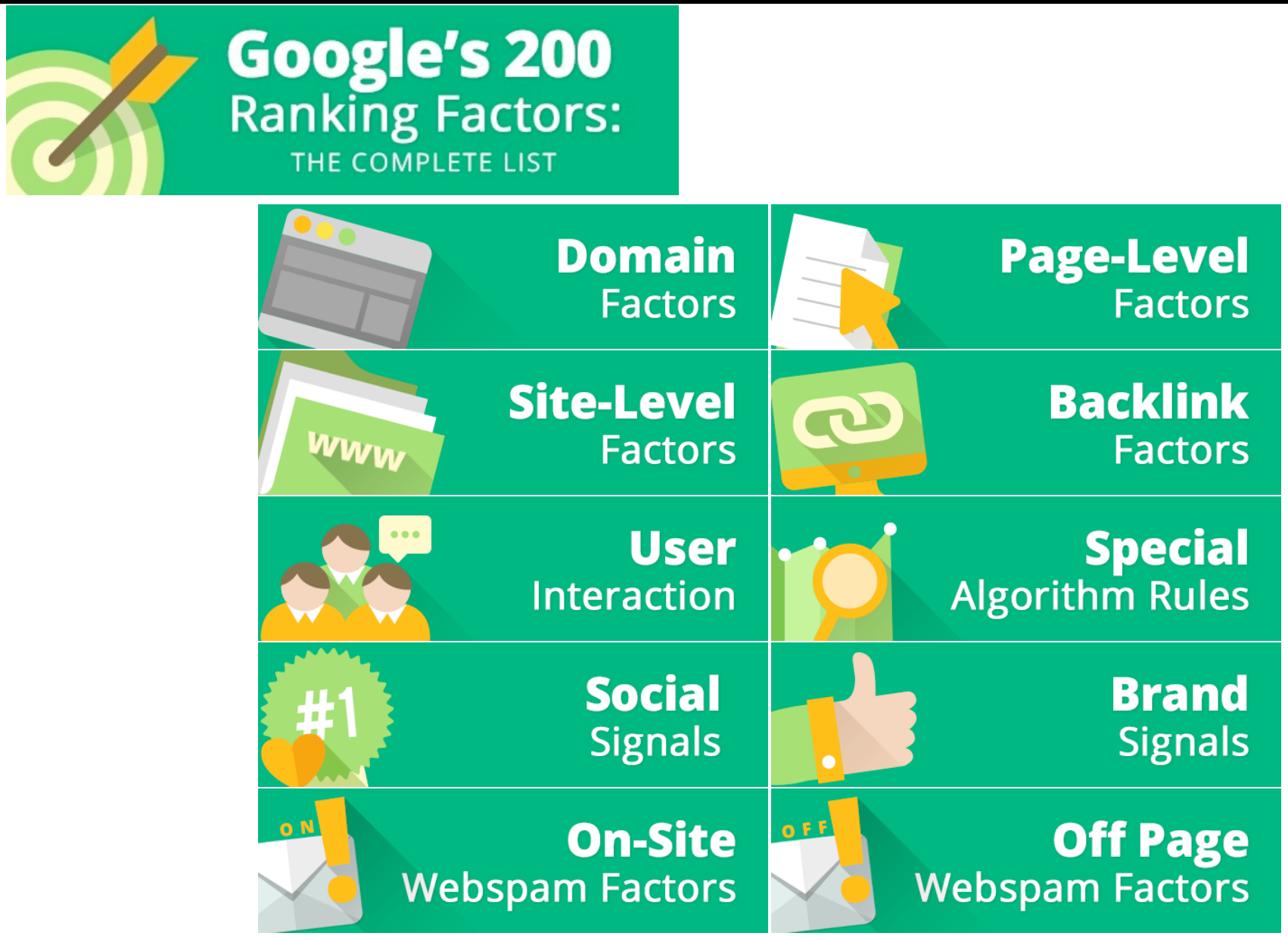
Suchmaschinenbeispiel - Twitter-Suche mit elasticsearch



Suchmaschinenbeispiel - Twitter-Suche mit elasticsearch



Ranking - Diverse Heuristiken



http://backlinko.com/google-ranking-factors

Vielen Dank fürs Zuhören

Dipl. Inf. Michael Christen
mc@yacy.net
@0rb1t3r

Links

YaCy Home Page
http://yacy.net
(Downloads, YaCy Forum + YaCy Wiki + Bugtracker dort verlinkt)