



Web-scale search engine toolkit

Today and tomorrow

Andrzej Białecki
ab@apache.org



Agenda

- About the project
- Web crawling in general
- Nutch architecture overview
- Nutch workflow:
 - Setup
 - Crawling
 - Searching
- Challenges (and some solutions)
- Nutch present and future
- Questions and answers

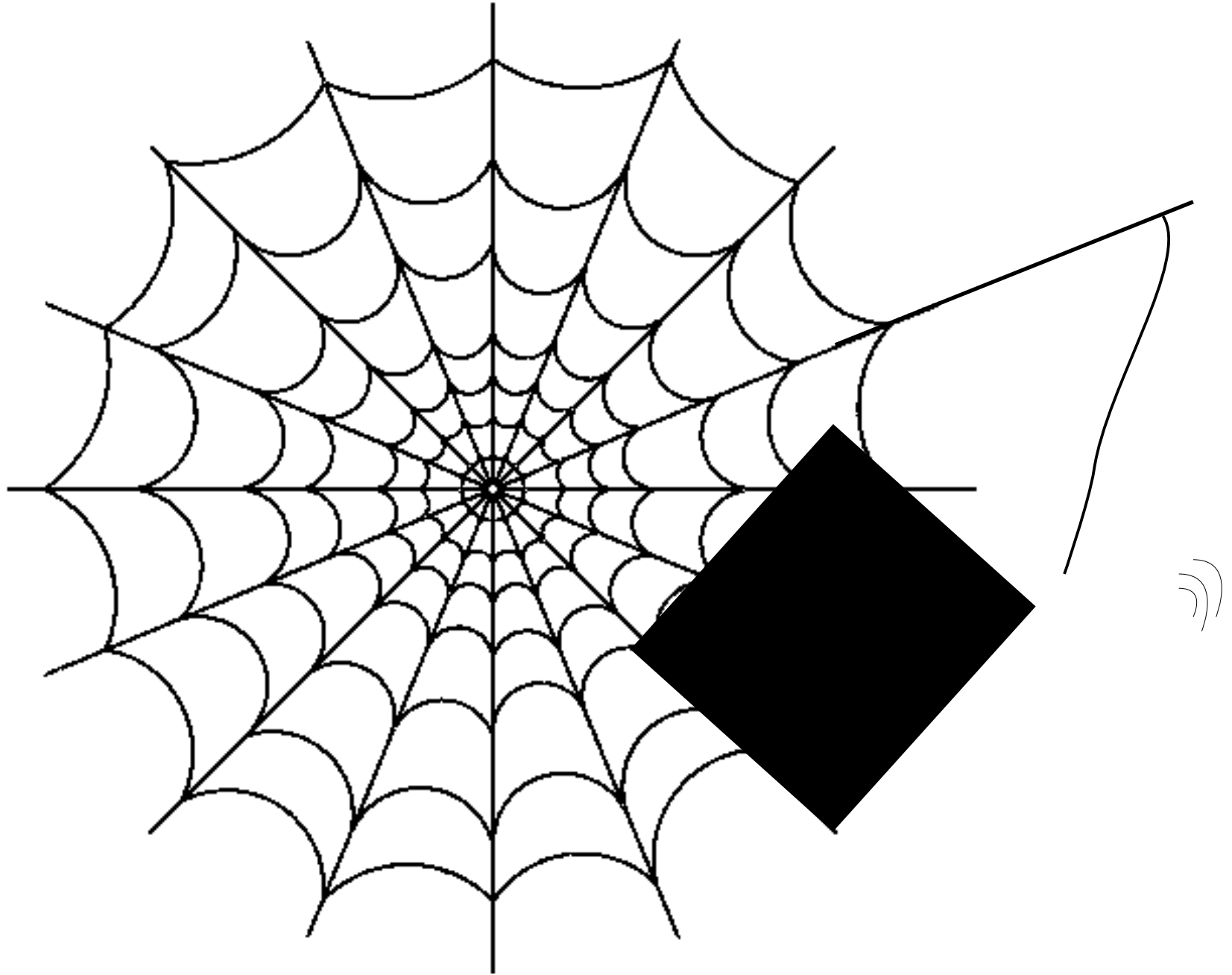


Apache Nutch project

- Founded in 2003 by Doug Cutting, the Lucene creator, and Mike Cafarella
- Apache project since 2004 (sub-project of Lucene)
- Spin-offs:
 - Map-Reduce and distributed FS → Hadoop
 - Content type detection and parsing → Tika
- Many installations in operation, mostly vertical search
- Collections typically 1 mln - 200 mln documents



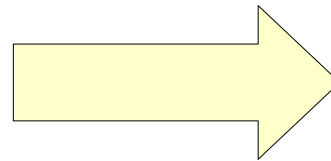
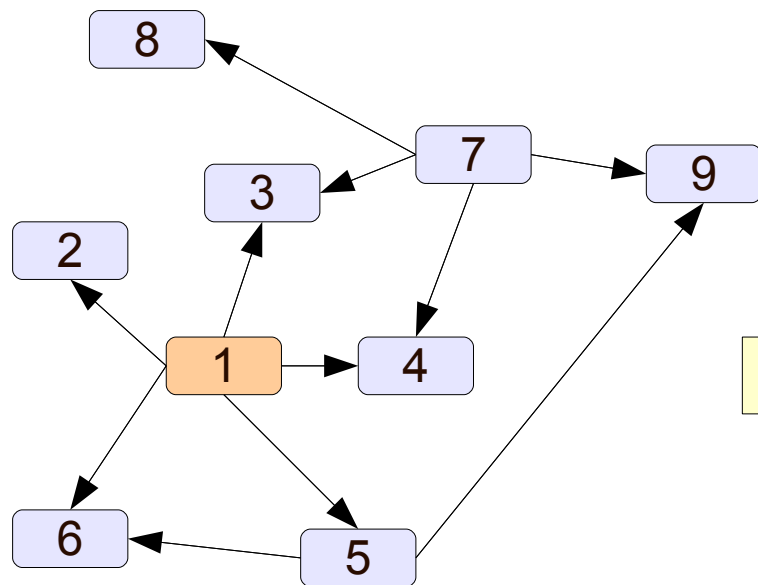
Concerning the web crawlers...





Web as a directed graph

- Nodes (vertices): URL-s as unique identifiers
- Edges (links): hyperlinks like ``
- Edge labels: `anchor text`
- Often represented as adjacency (neighbor) lists
- Traversal: follow the edges, breadth-first, depth-first, random

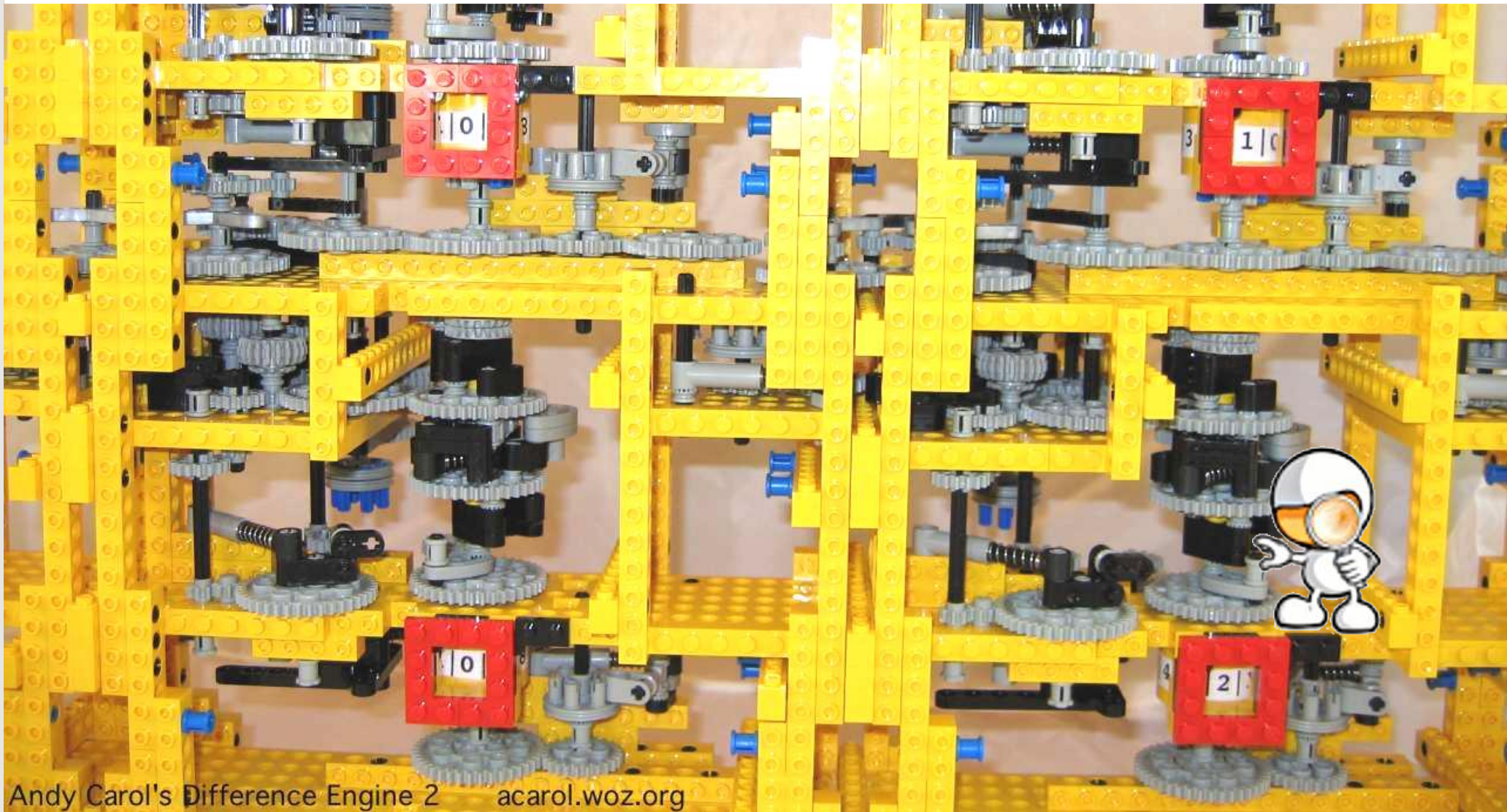


1 → 2, 3, 4, 5, 6
5 → 6, 9
7 → 3, 4, 8, 9



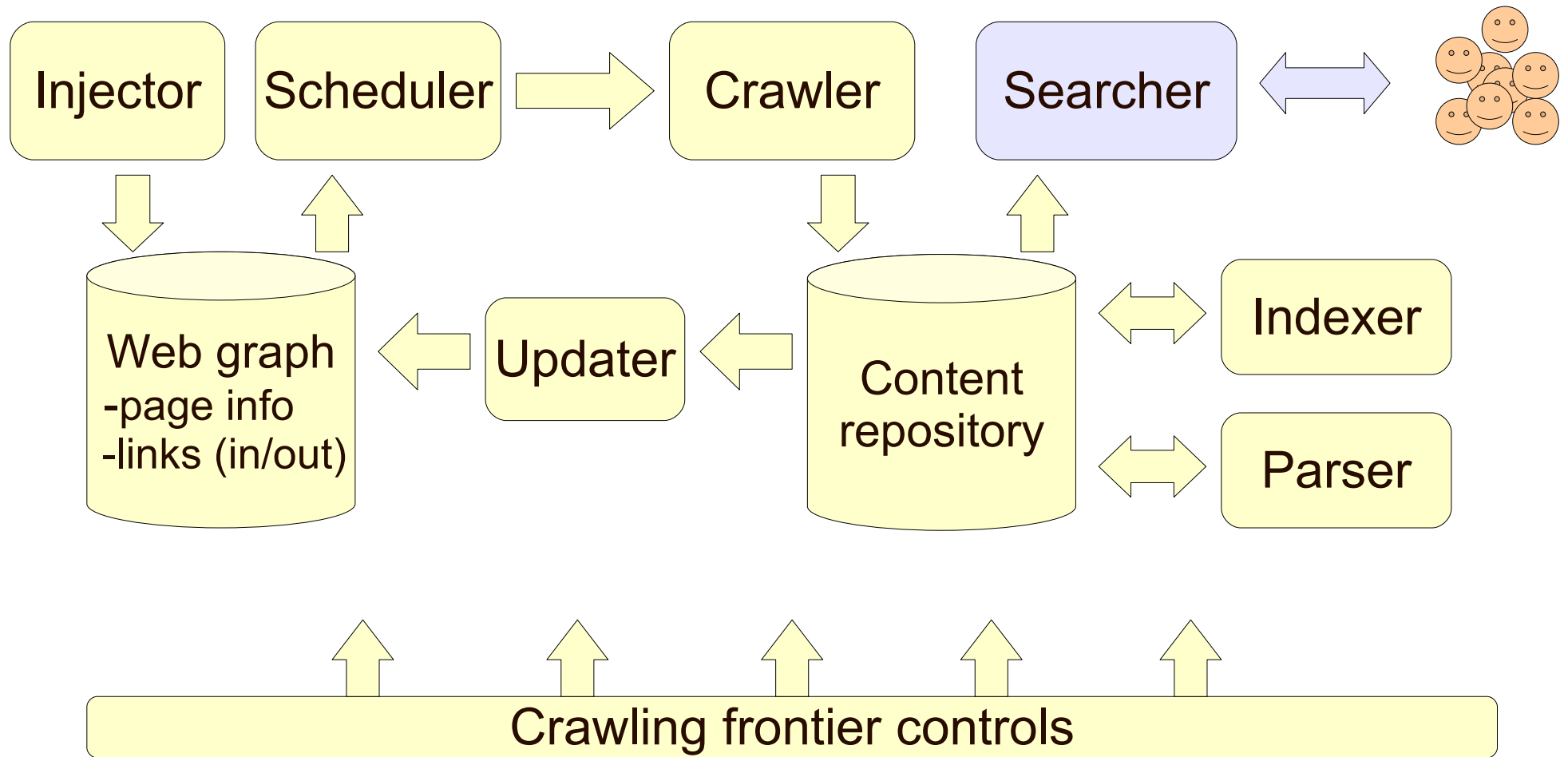
What's in a search engine?

... a few things that may surprise you! 😊





Search engine building blocks





Nutch features at a glance

- Page database and link database (web graph)
- Plugin-based, highly modular:
 - Most behavior can be changed via plugins
- Multi-protocol, multi-threaded, distributed crawler
- Plugin-based content processing (parsing, filtering)
- Robust crawling frontier controls
- Scalable data processing framework
 - Map-reduce processing
- Full-text indexer & search engine
 - Using Lucene or Solr
 - Support for distributed search
- Robust API and integration options

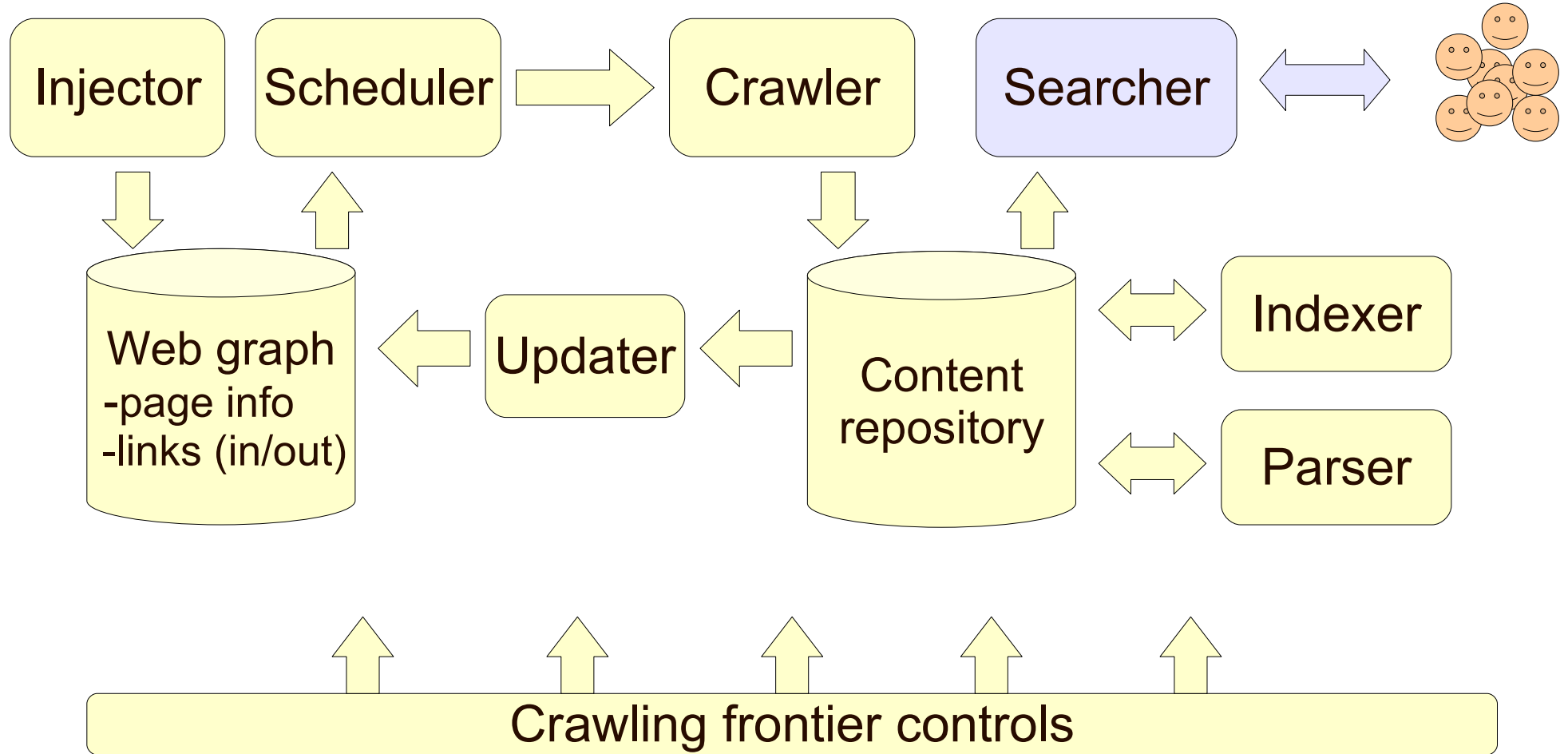


Hadoop foundation

- File system abstraction
 - Local FS, or
 - Distributed FS
 - also Amazon S3, Kosmos and other FS impl.
- Map-reduce processing
 - Currently central to Nutch algorithms
 - Processing tasks are executed as one or more map-reduce jobs
 - Data maintained as Hadoop MapFile-s / SequenceFile-s
 - Massive updates very efficient
 - Small updates costly
 - Hadoop data is immutable, once created
 - Updates are really merge & replace

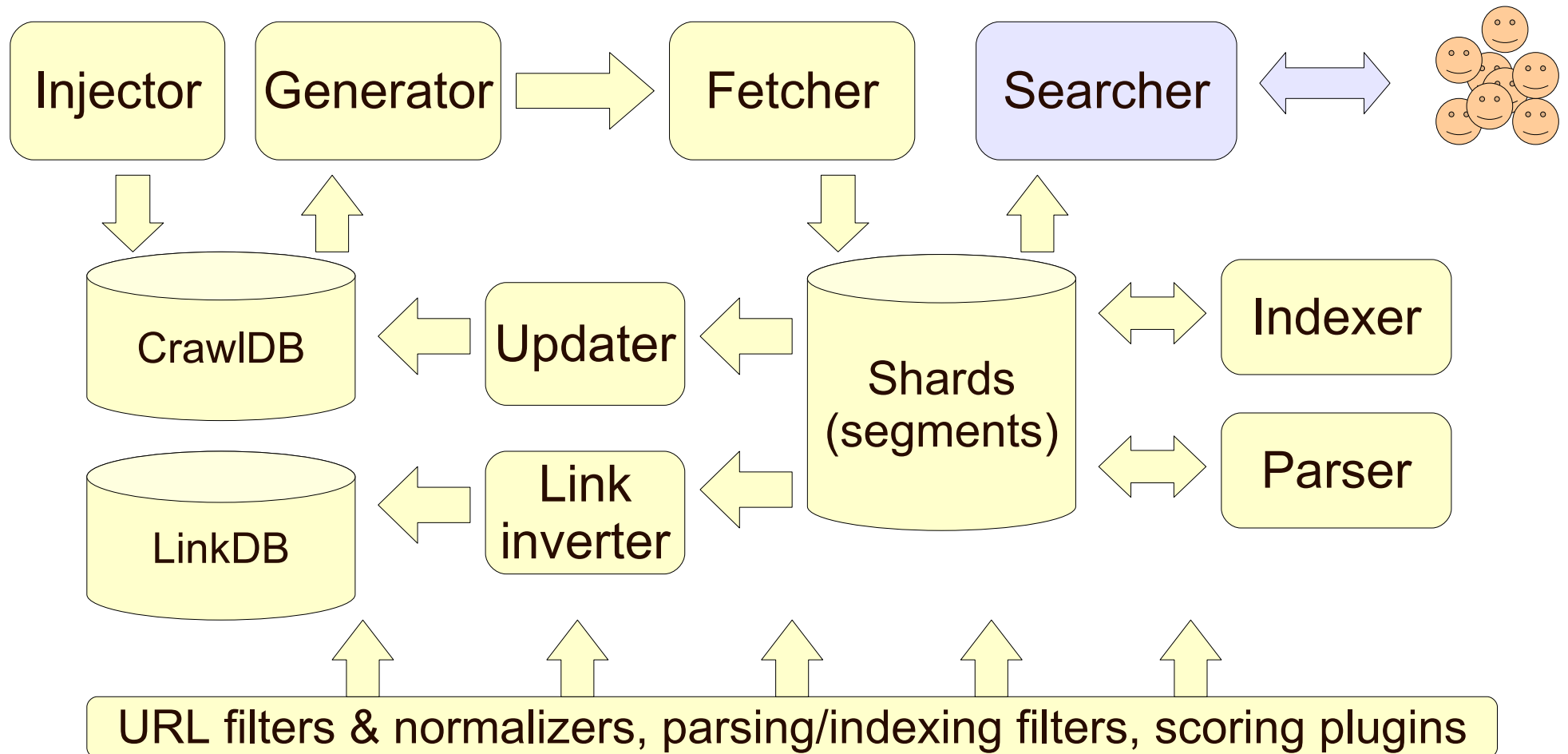


Search engine building blocks



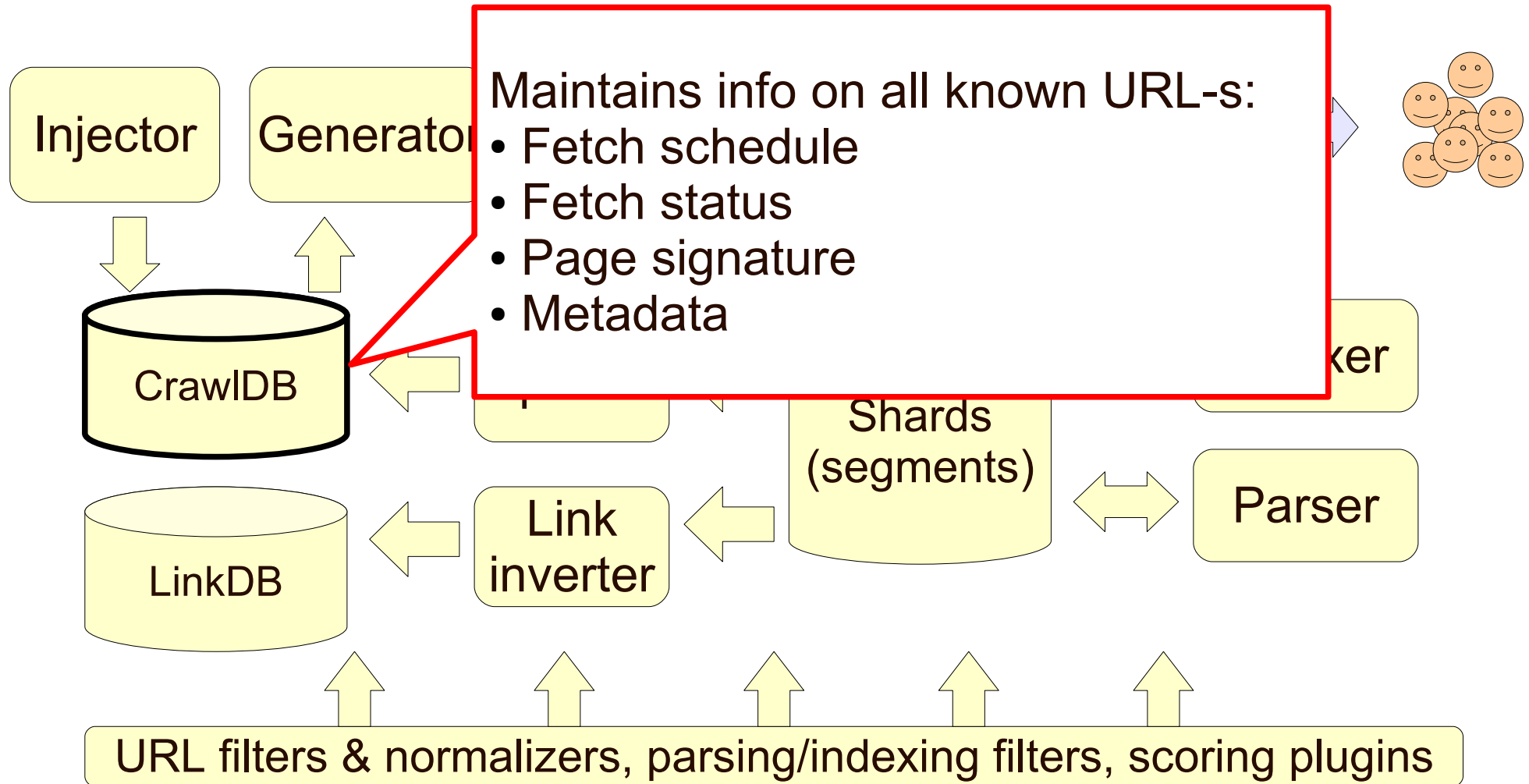


Nutch building blocks



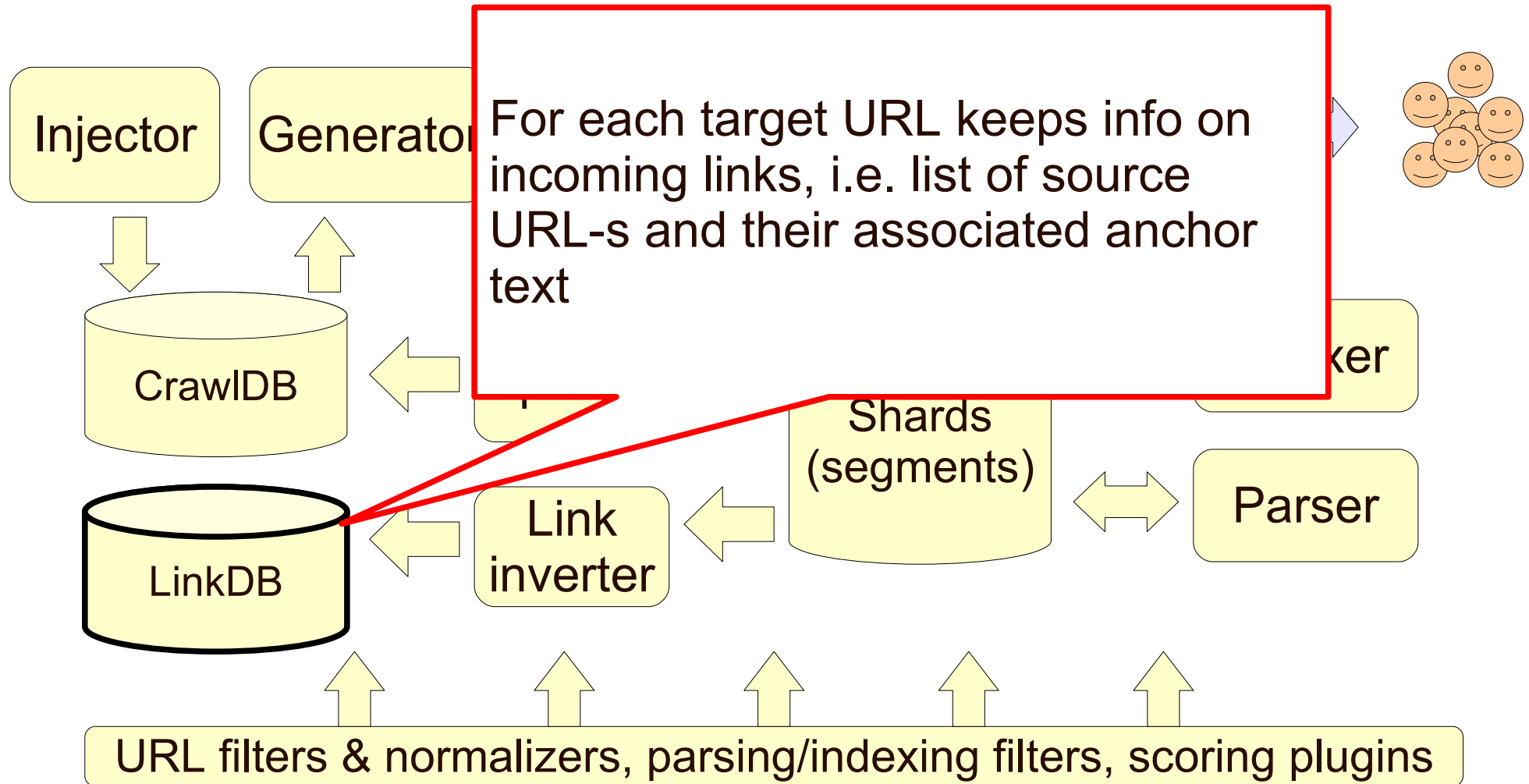


Nutch data





Nutch data

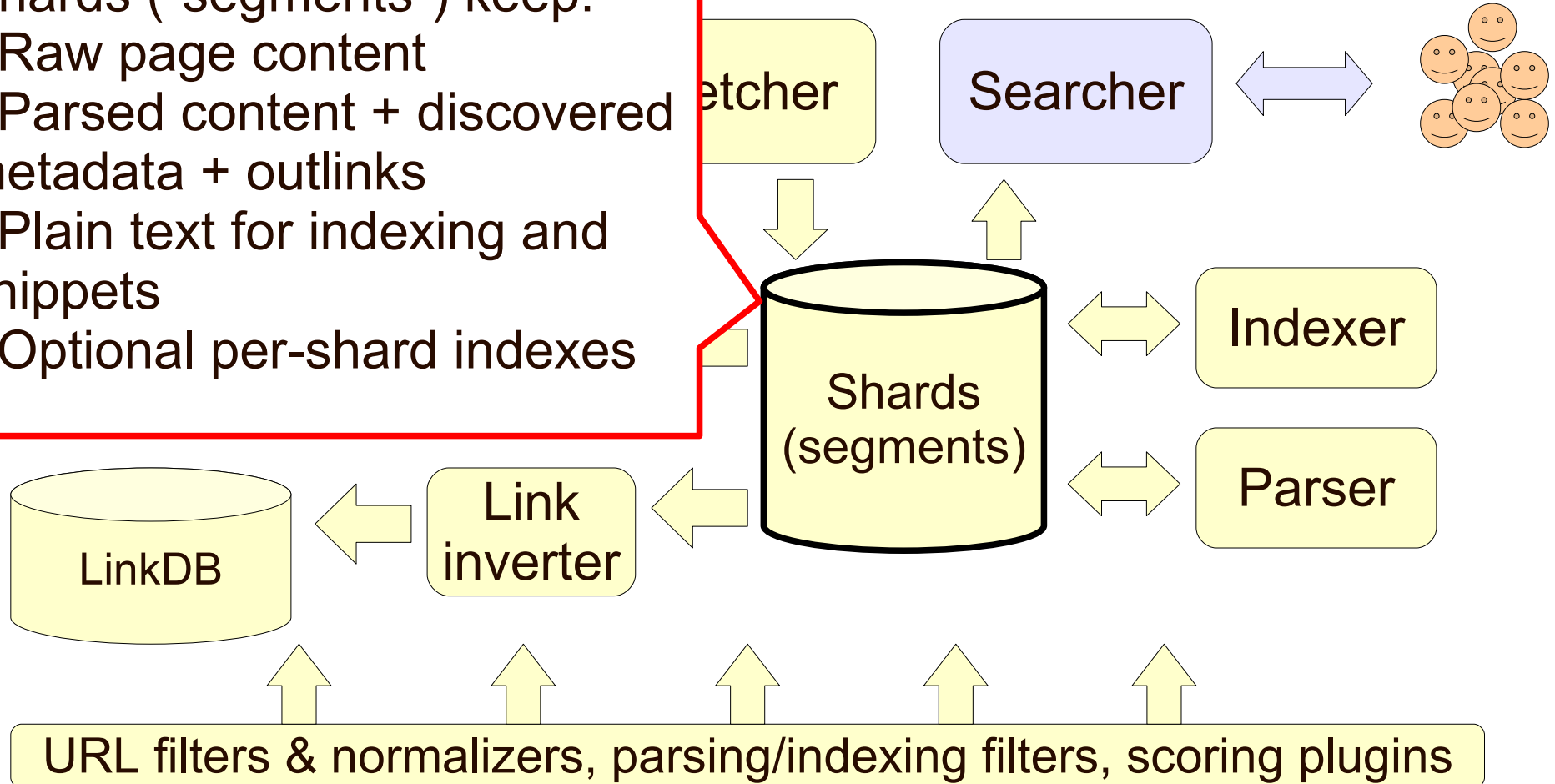




Nutch data

Shards (“segments”) keep:

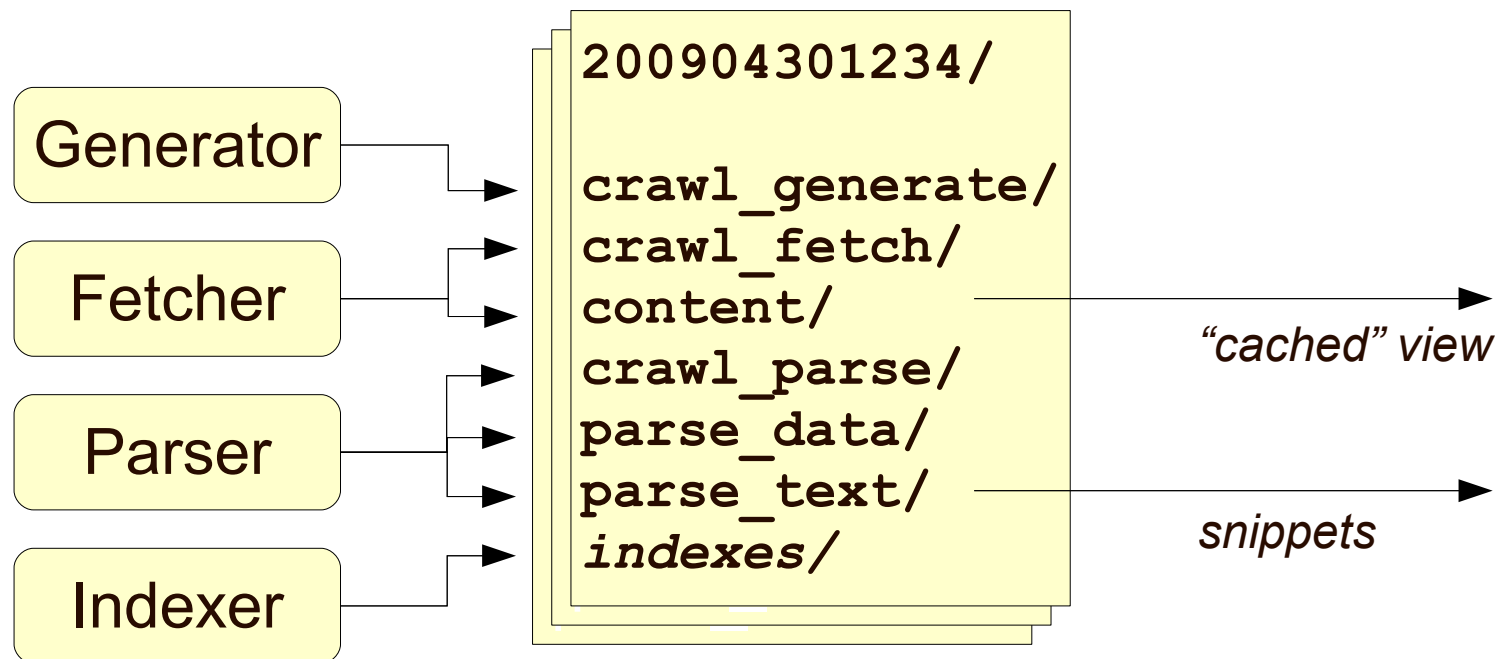
- Raw page content
- Parsed content + discovered metadata + outlinks
- Plain text for indexing and snippets
- Optional per-shard indexes





Nutch shards (a.k.a. “segments”)

- Unit of work (batch) – easier to process massive datasets
- Convenience placeholder, using predefined directory names
- Unit of deployment to the search infrastructure
- May include per-shard Lucene indexes
- Once completed they are basically unmodifiable
 - No in-place updates of content, or replacing of obsolete content
- Periodically phased-out

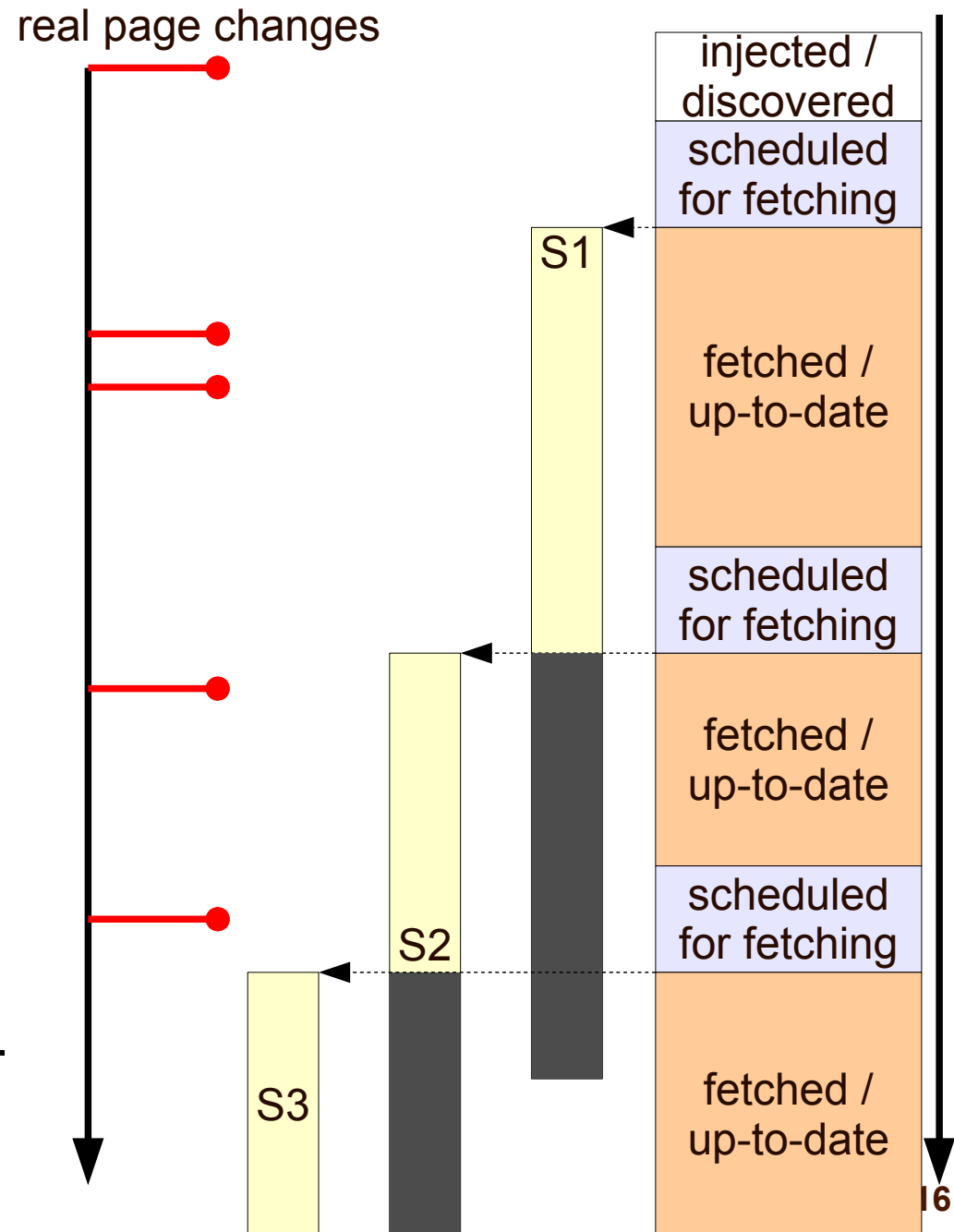




URL life-cycle and shards

observed changes time


- A time-lapse view of reality
- Goals:
 - Fresh index → sync with the real rate of changes
 - Minimize re-fetching → don't fetch unmodified pages
- Each page may need its own crawl schedule
- Shard management:
 - Page may be present in many shards, but only the most recent record is valid
 - Inconvenient to update in-place, just mark as deleted
 - Phase-out old shards and force re-fetch of the remaining pages





Crawling frontier

- No authoritative catalog of web pages
- Search engines discover their view of web universe
 - Start from “seed list”
 - Follow (walk) some (*useful? interesting?*) outlinks
- Many dangers of simply wandering around
 - explosion or collapse of the frontier
 - collecting unwanted content (spam, junk, offensive)



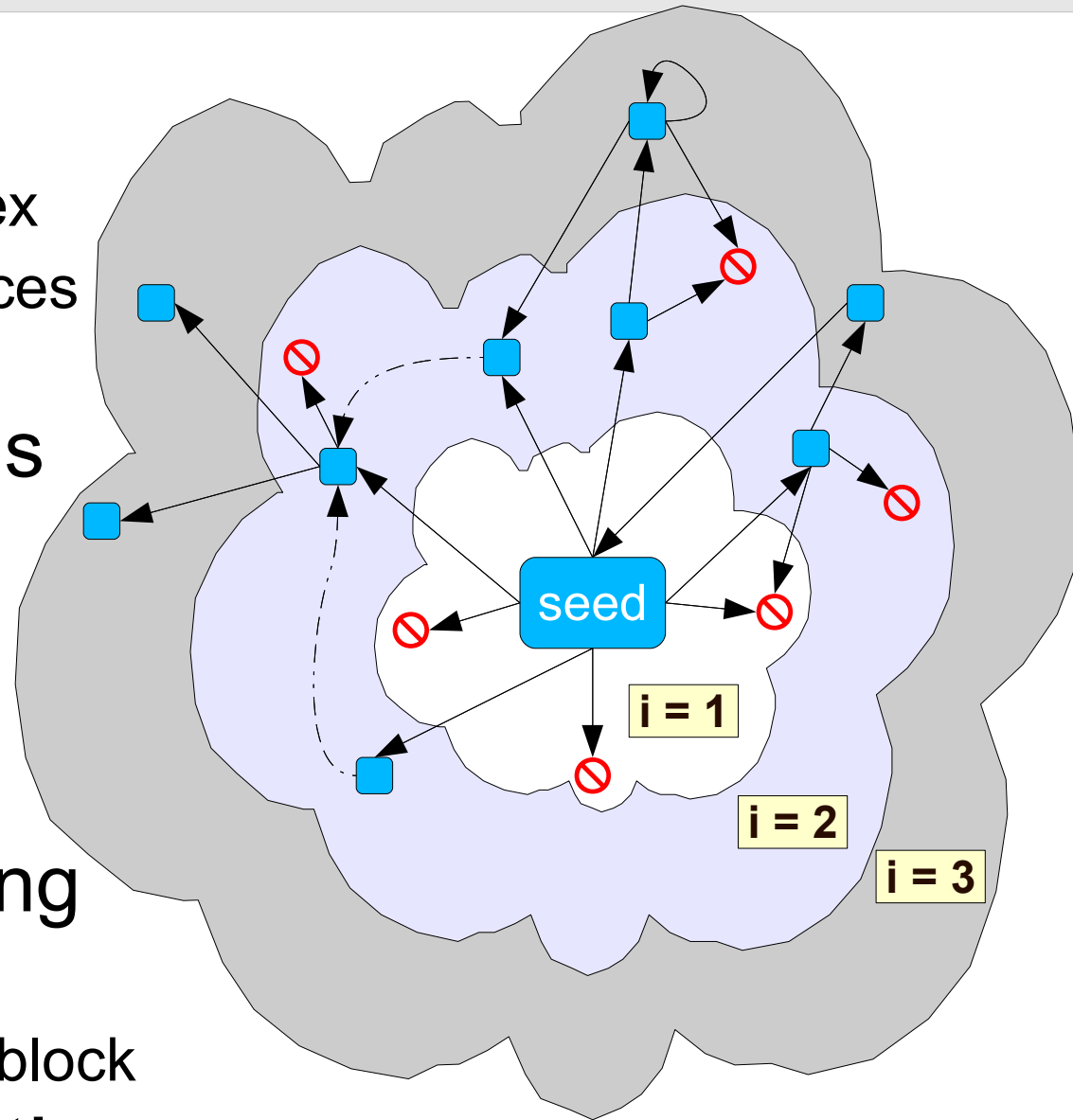
*I could use
some guidance*





Controlling the crawling frontier

- URL filter plugins
 - White-list, black-list, regex
 - May use external resources (DB-s, services ...)
- URL normalizer plugins
 - Resolving relative path elements
 - “Equivalent” URLs
- Additional controls using scoring plugins
 - priority, metadata select/block
- Crawler traps – difficult!
 - Domain / host / path-level stats





Wide vs. focused crawling

- Differences:
 - Little technical difference in configuration
 - Big difference in operations, maintenance and quality
- Wide crawling:
 - (Almost) Unlimited crawling frontier
 - High risk of spamming and junk content
 - **“Politeness” a very important limiting factor**
 - Bandwidth & DNS considerations
- Focused (vertical or enterprise) crawling:
 - Limited crawling frontier
 - Bandwidth or politeness is often not an issue
 - Low risk of spamming and junk content



Vertical & enterprise search

- Vertical search
 - Range of selected “reference” sites
 - Robust control of the crawling frontier
 - Extensive content post-processing
 - Business-driven decisions about ranking
- Enterprise search
 - Variety of data sources and data formats
 - Well-defined and limited crawling frontier
 - Integration with in-house data sources
 - Little danger of spam
 - PageRank-like scoring usually works poorly



Face to face with Nutch





Simple set-up and running

- You already have Java 5+ , right?
- Get a 1.0 release or a nightly build (pretty stable)
- Simple search setup uses Tomcat for search web app
- Command-line bash script: bin/nutch
 - Windows users: get Cygwin
 - Early version of a web-based UI console
 - Hadoop web-based monitoring



Configuration: files

- Edit configuration in `conf/nutch-site.xml`
 - Check `nutch-default.xml` for defaults and docs
 - You **MUST** at least fill the name of your agent
- Active plugins configuration
- Per-plugin properties
- External configuration files
 - `regex-urlfilter.xml`
 - `regex-normalize.xml`
 - `parse-plugins.xml`: mapping of MIME type to plugin



Nutch plugins

- Plugin-based extensions for:
 - Crawl scheduling
 - URL filtering and normalization
 - Protocol for getting the content
 - Content parsing
 - Text analysis (tokenization)
 - Page signature (to detect near-duplicates)
 - Indexing filters (index fields & metadata)
 - Snippet generation and highlighting
 - Scoring and ranking
 - Query translation and expansion (user → Lucene)



Main Nutch workflow

- **Inject**: initial creation of CrawlDB
 - Insert seed URLs
 - Initial LinkDB is empty

- **Generate** new shard's fetchlist
- **Fetch** raw content
- **Parse** content (discovers outlinks)
- **Update CrawlDB** from shards
- **Update LinkDB** from shards
- **Index** shards

(repeat)

Command-line:
bin/nutch

inject

generate

fetch

parse

updatedb

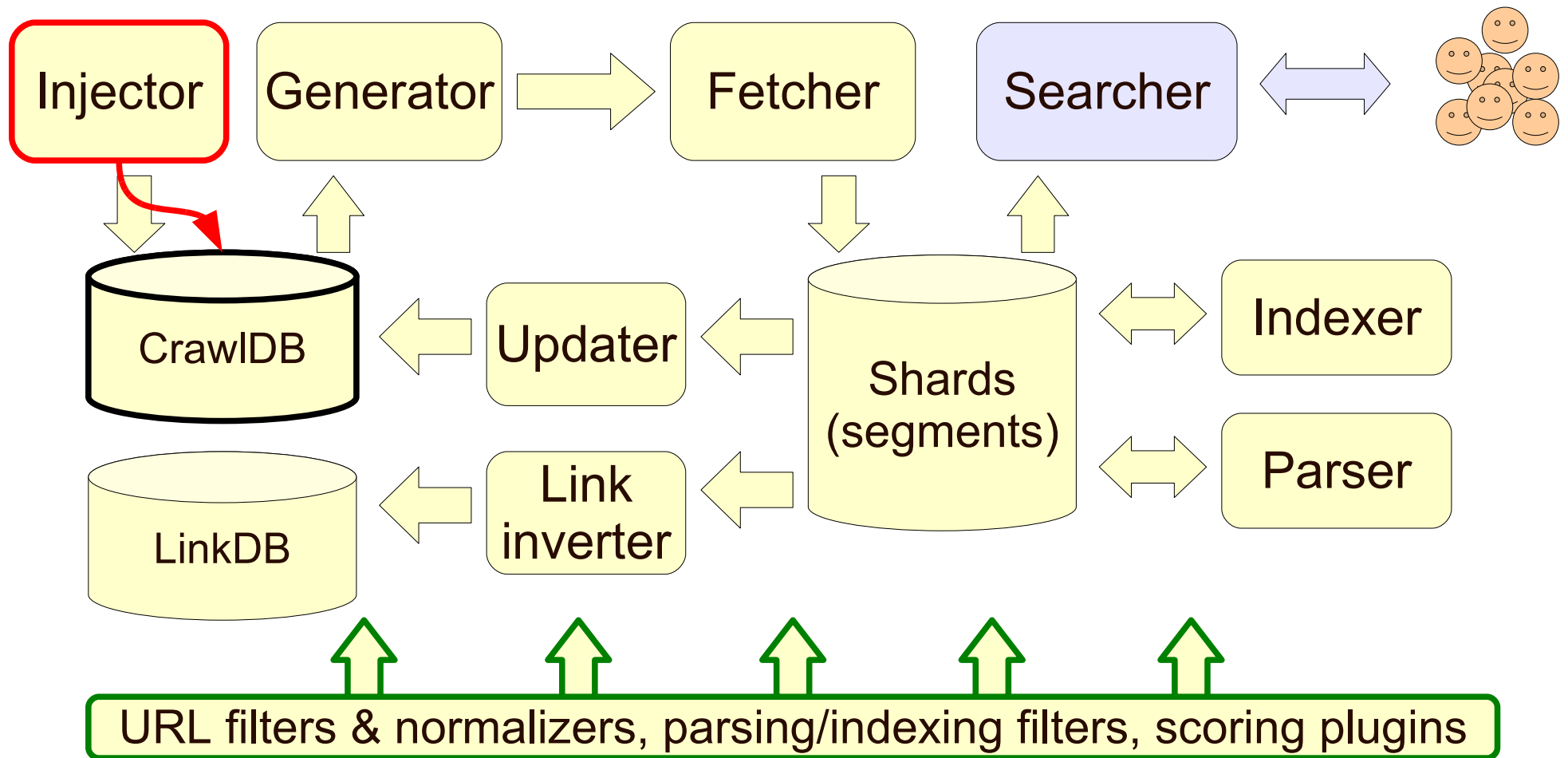
updatelinkdb

index /

solrindex

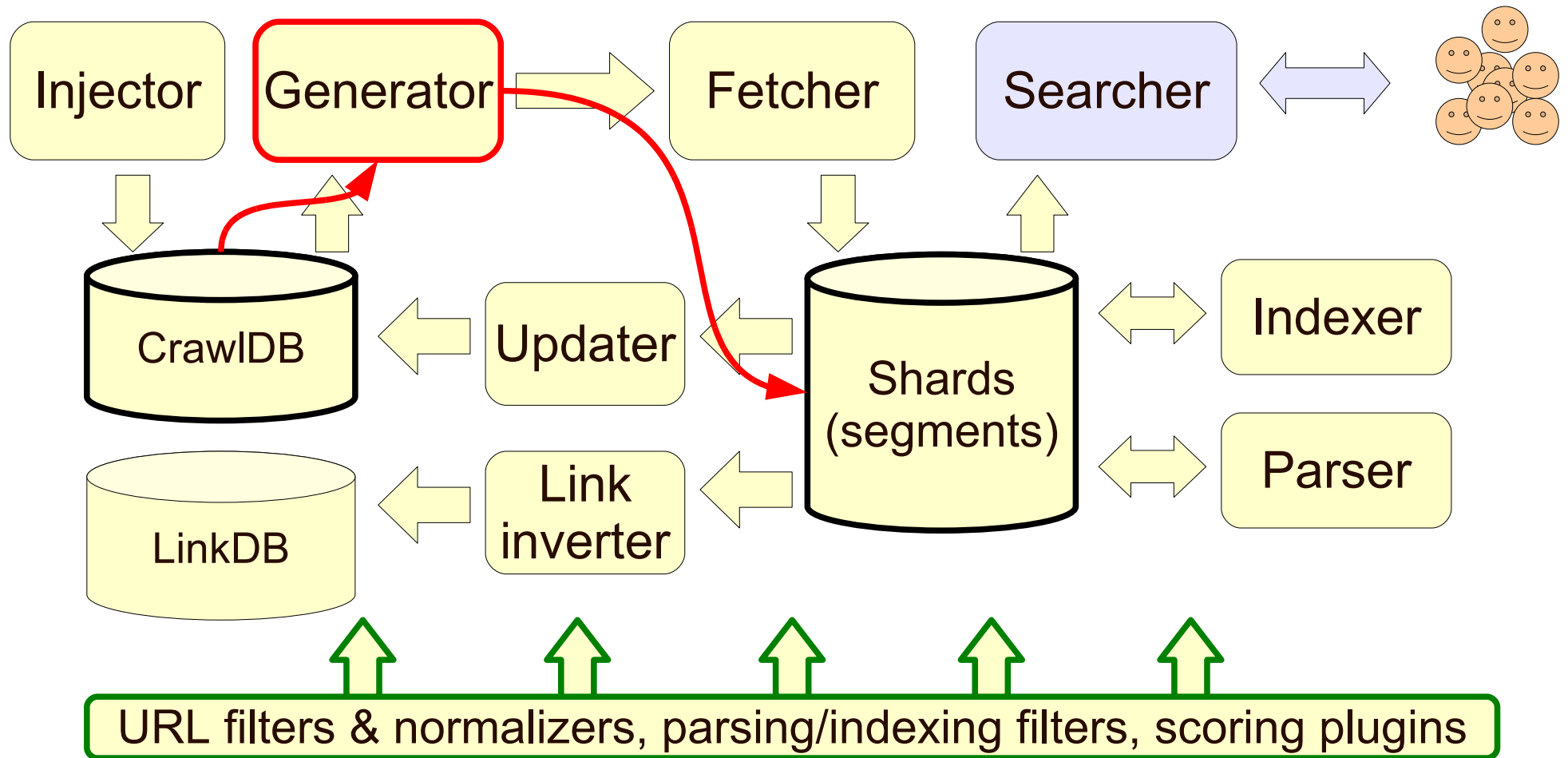


Injecting new URL-s





Generating fetchlists



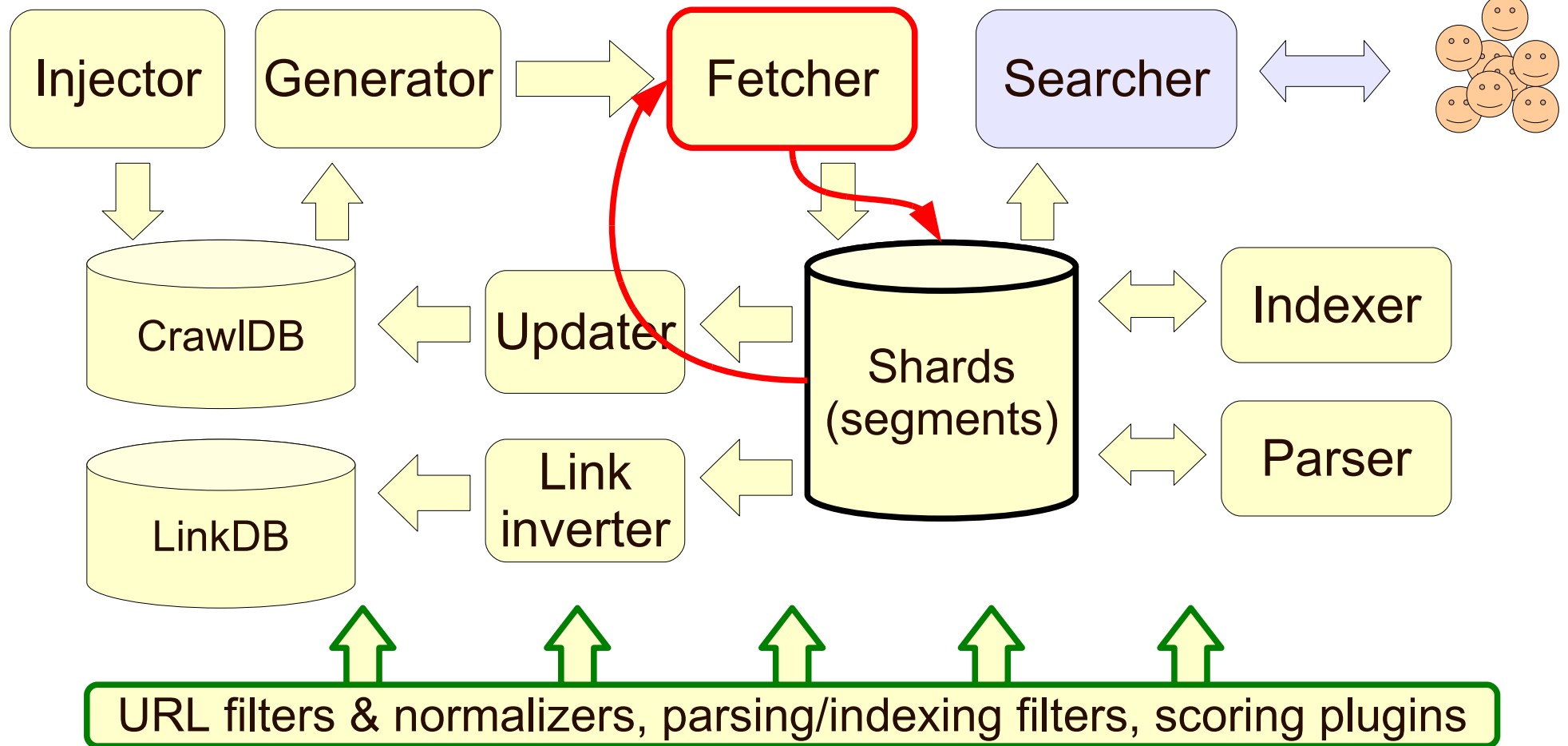


Workflow: generating fetchlists

- What to fetch next?
 - Breadth-first – important due to “politeness” limits
 - Expired (longer than `fetchTime` + `fetchInterval`)
 - Highly ranking (PageRank)
 - Newly added
- Fetchlist generation:
 - “topN” - select best candidates
 - Priority based on many factors, pluggable
- Adaptive fetch schedule
 - Detect rate of changes & time of change
 - Detect unmodified content
 - Hmm, and how to recognize this? → approximate page signatures (near-duplicate detection)



Fetching content



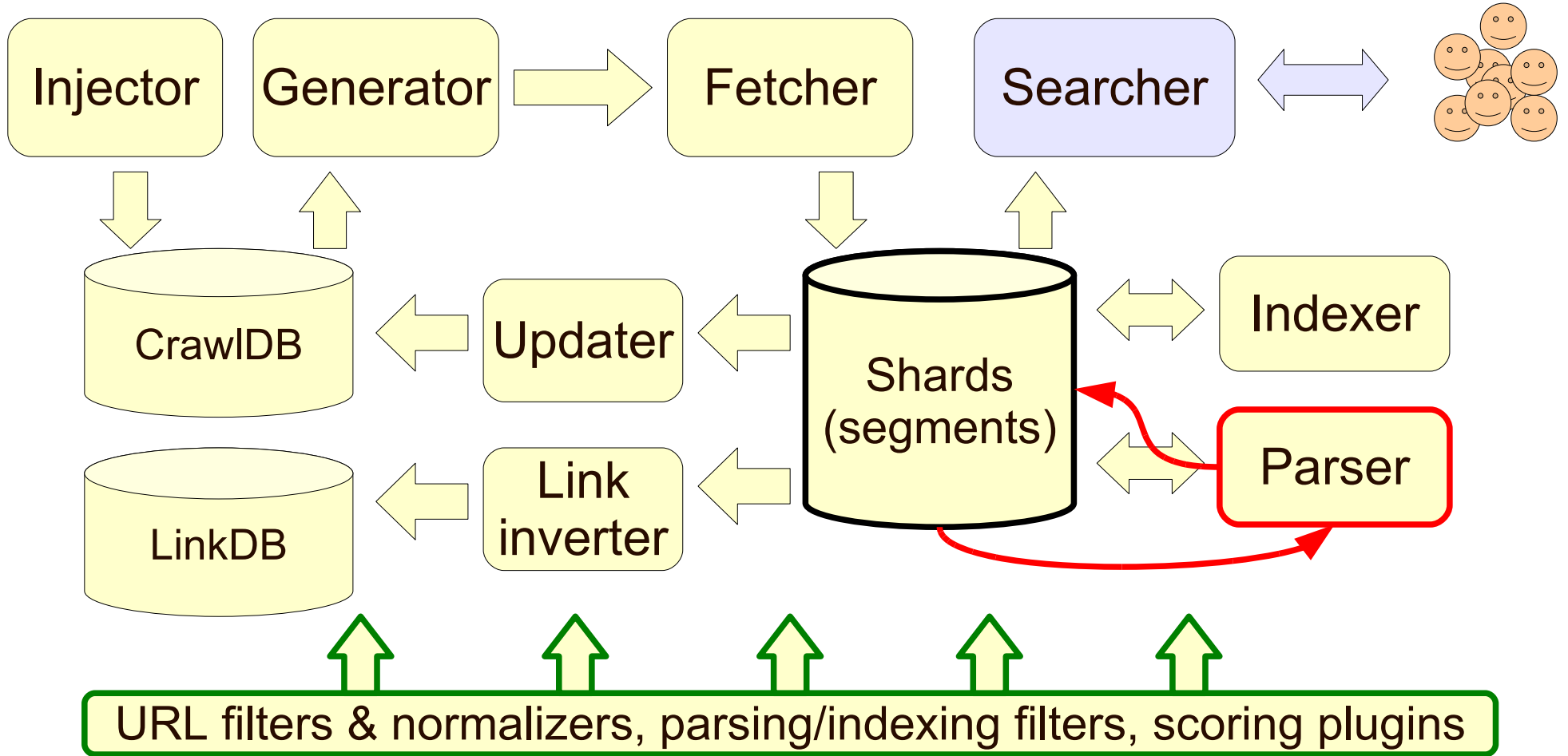


Workflow: fetching

- Multi-protocol: HTTP(s), FTP, [file://](#), etc ...
- Coordinates multiple threads accessing the same host
 - “politeness” issues vs. crawling efficiency
 - Host: an IP or DNS name?
 - Redirections: should follow immediately? What kind?
- Other netiquette issues:
 - robots.txt: disallowed paths, Crawl-Delay
- Preparation for parsing:
 - Content type detection issues
 - Parsing usually executed as a separate step (resource hungry and sometimes unstable)



Content processing



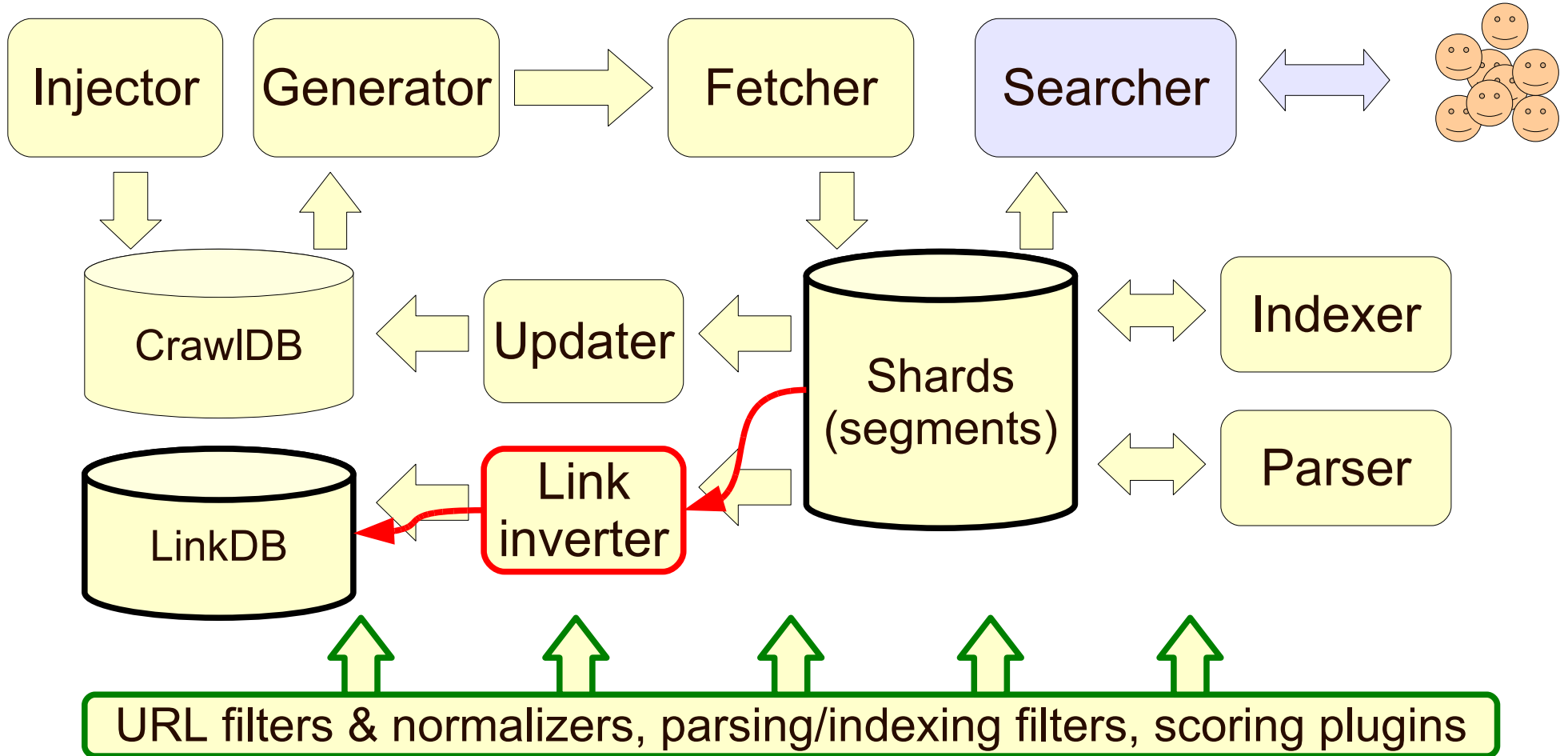


Workflow: content processing

- Protocol plugins retrieve content as plain bytes + protocol-level metadata (e.g. HTTP headers)
- Parse plugins
 - Content is parsed by MIME-type specific parsers
 - Content is parsed into parseData (title, outlinks, other metadata) and parseText (plain text content)
- Nutch supports many popular file formats



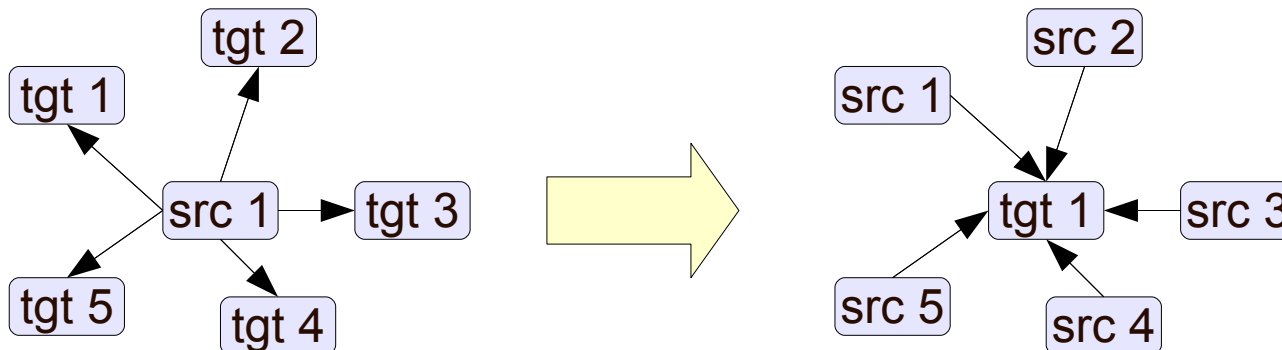
Link inversion





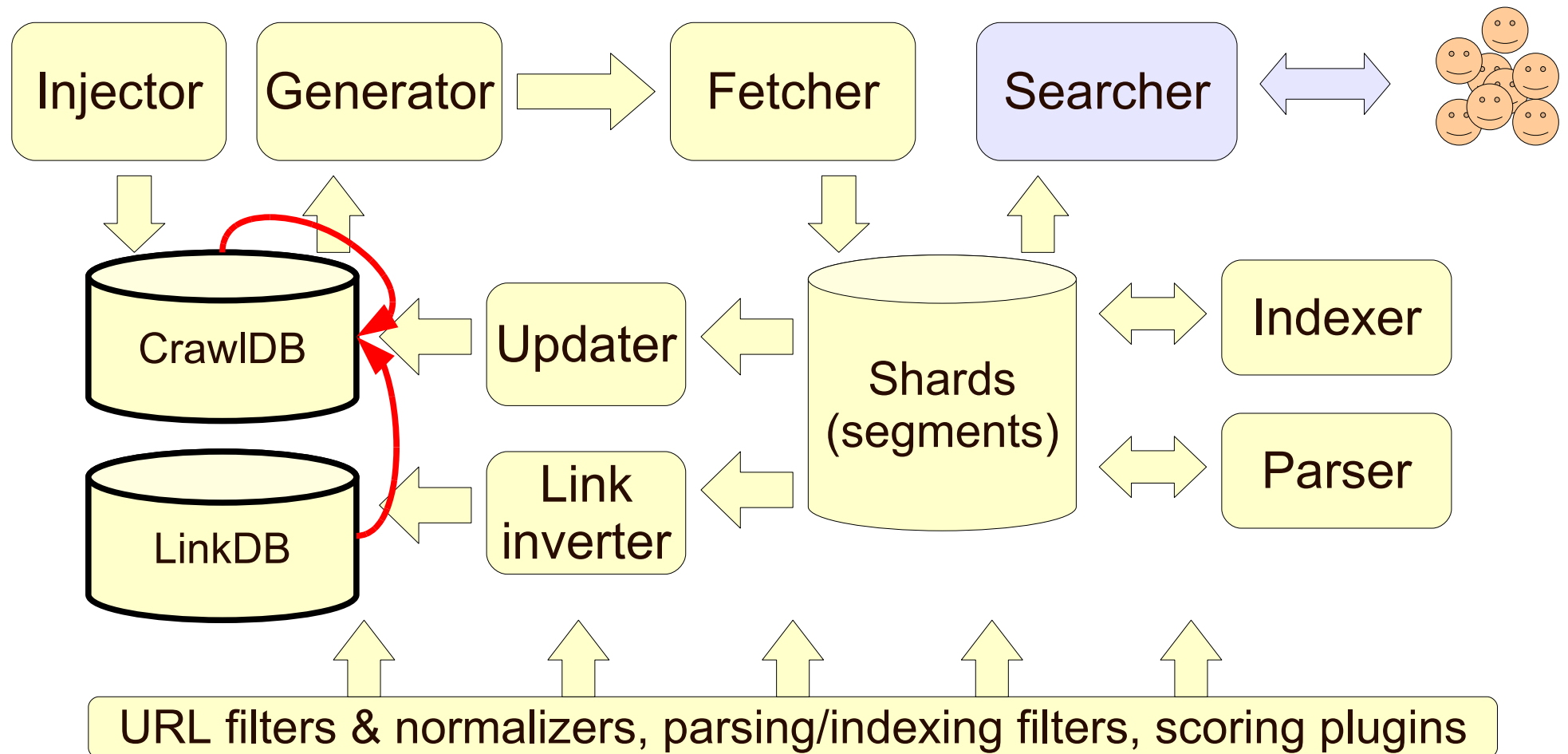
Workflow: link inversion

- Pages have outgoing links (outlinks)
 - ... I know where I'm pointing to
- Question: who points to me?
 - ... I don't know, there is no catalog of pages
 - ... NOBODY knows for sure either!
- In-degree indicates importance of the page
- Anchor text provides important semantic info
- Partial answer: invert the outlinks that I know about, and group by target





Page importance - scoring



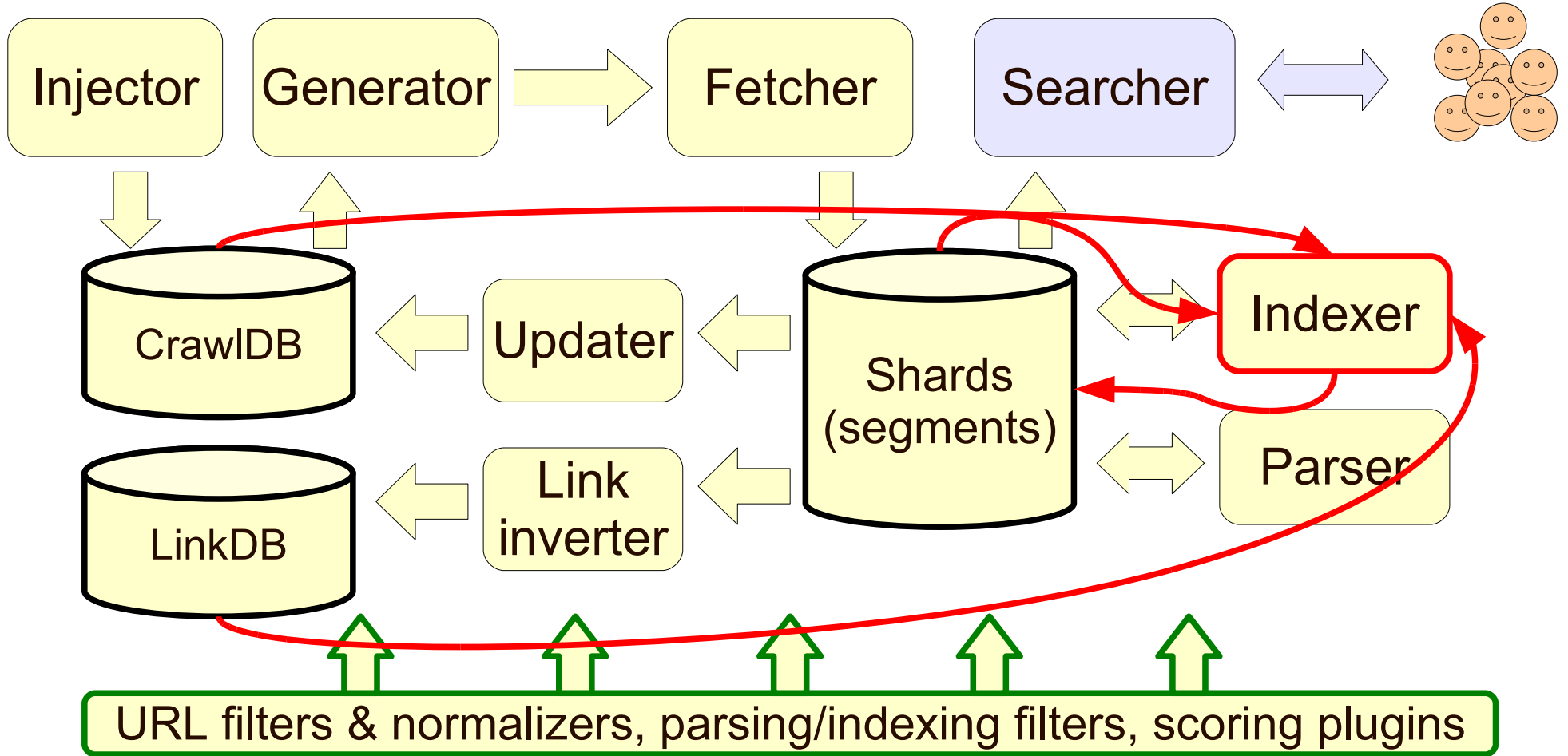


Workflow: page scoring

- Query-independent importance factor
 - Affects search ranking
 - Affects crawl prioritization
- **May include arbitrary decision of the operator**
- Currently two systems (plugins + tools)
- OPIC scoring
 - Doesn't require explicit steps - “online”
 - Difficult to stabilize
- PageRank scoring with loop detection
 - Periodically run to update CrawlDb scores
 - Computationally intensive, esp. loop detection



Indexing



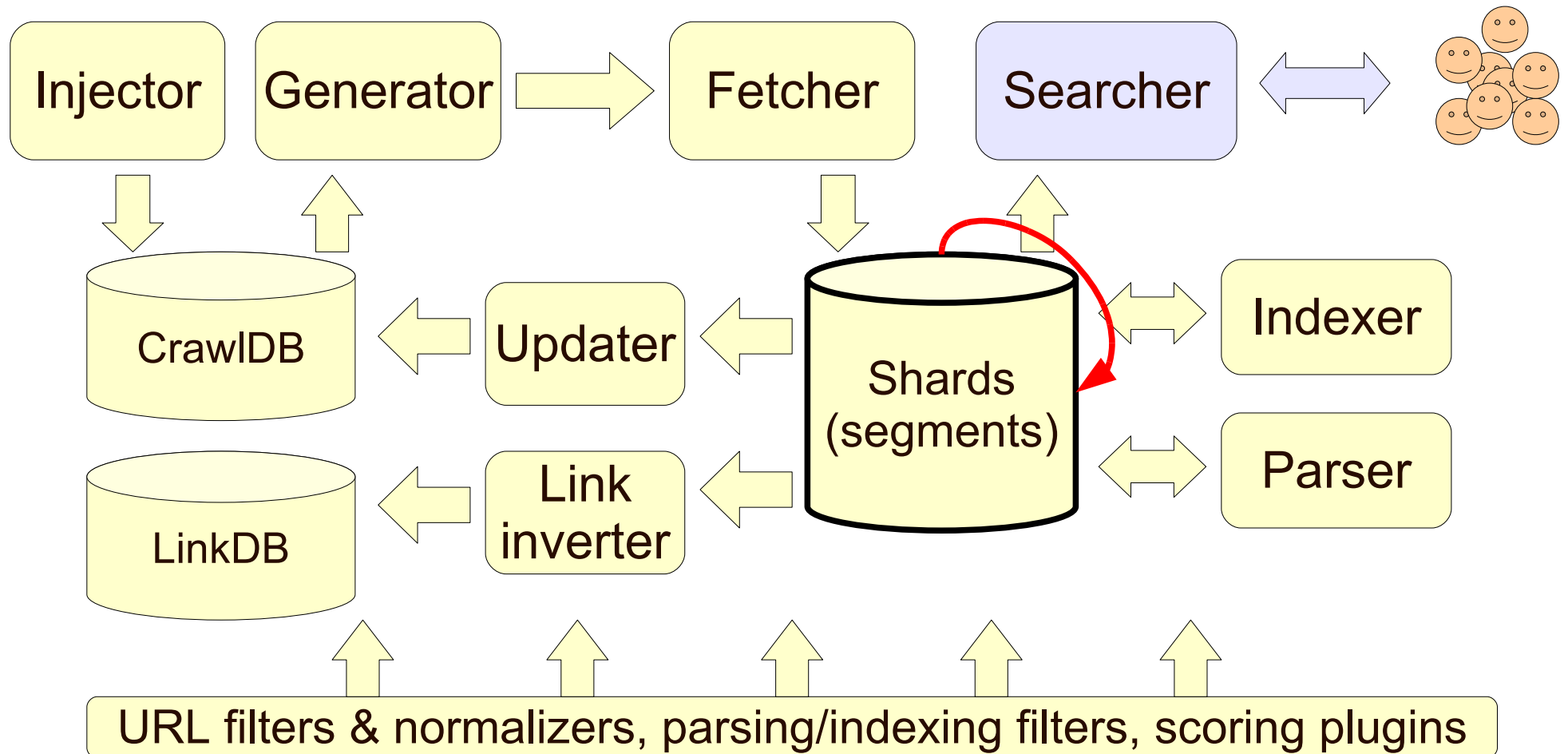


Workflow: indexing

- Indexing plugins
 - Create full-text search index from segment data
 - Apply adjustments to score per document
 - May further post-process the parsed content (e.g. language identification) to facilitate advanced search
- Indexers
 - Lucene indexer – builds indexes to be served by Nutch search servers
 - Solr indexer – submits documents to a Solr instance



De-duplication



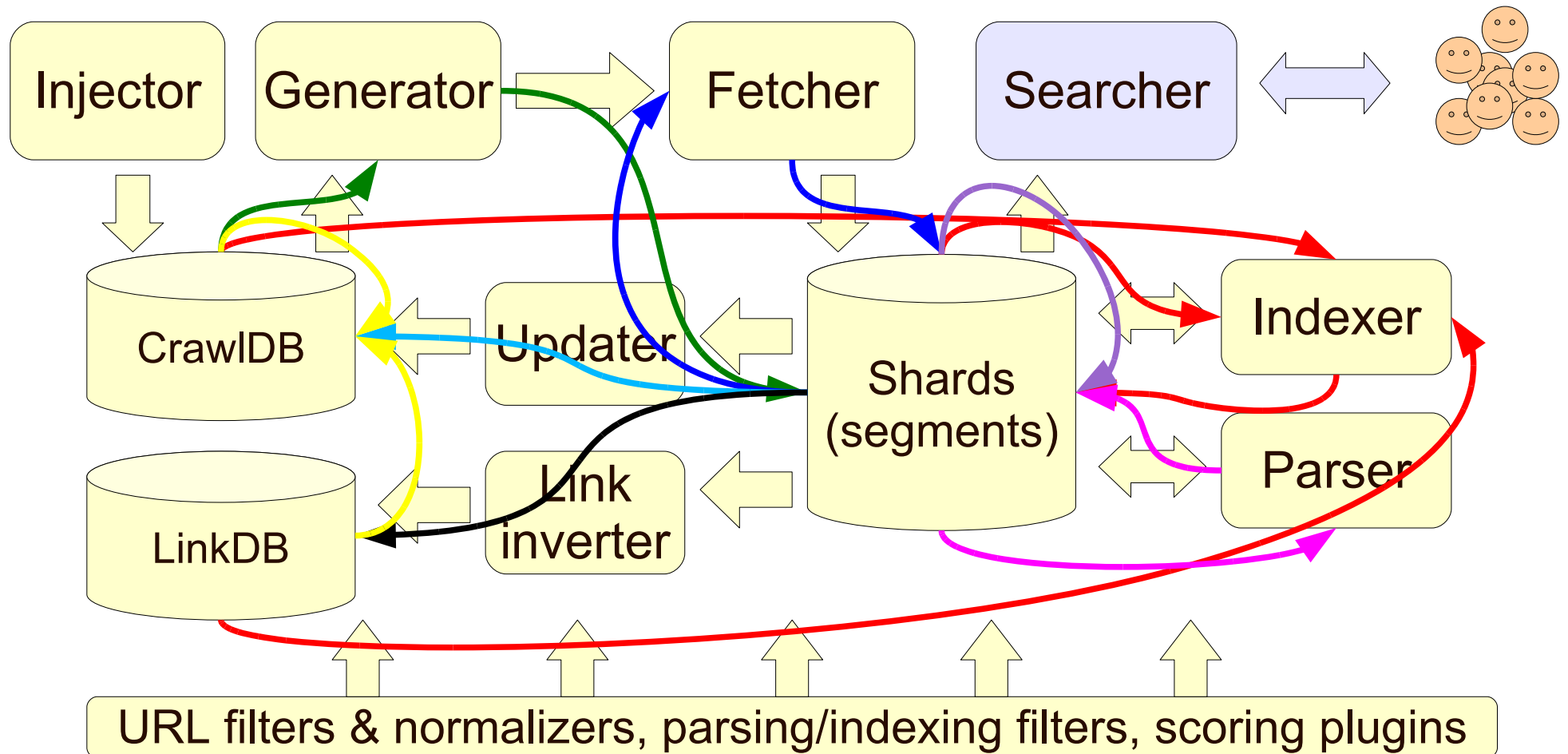


Workflow: de-duplication

- The same page may be present in many shards
 - Obsolete versions in older shards
 - Mirrored pages, or equivalent (a.com → www.a.com)
- Many other pages are almost identical
 - Template-related differences (banners, current date)
 - Font / layout changes, minor re-wording
- Hmm ... what is a significant change ???
 - Tricky issue! Hint: what is the page content?
- Near-duplicate detection and removal
 - Nutch uses approximate page signatures (fingerprints)
 - Duplicates are only marked as deleted

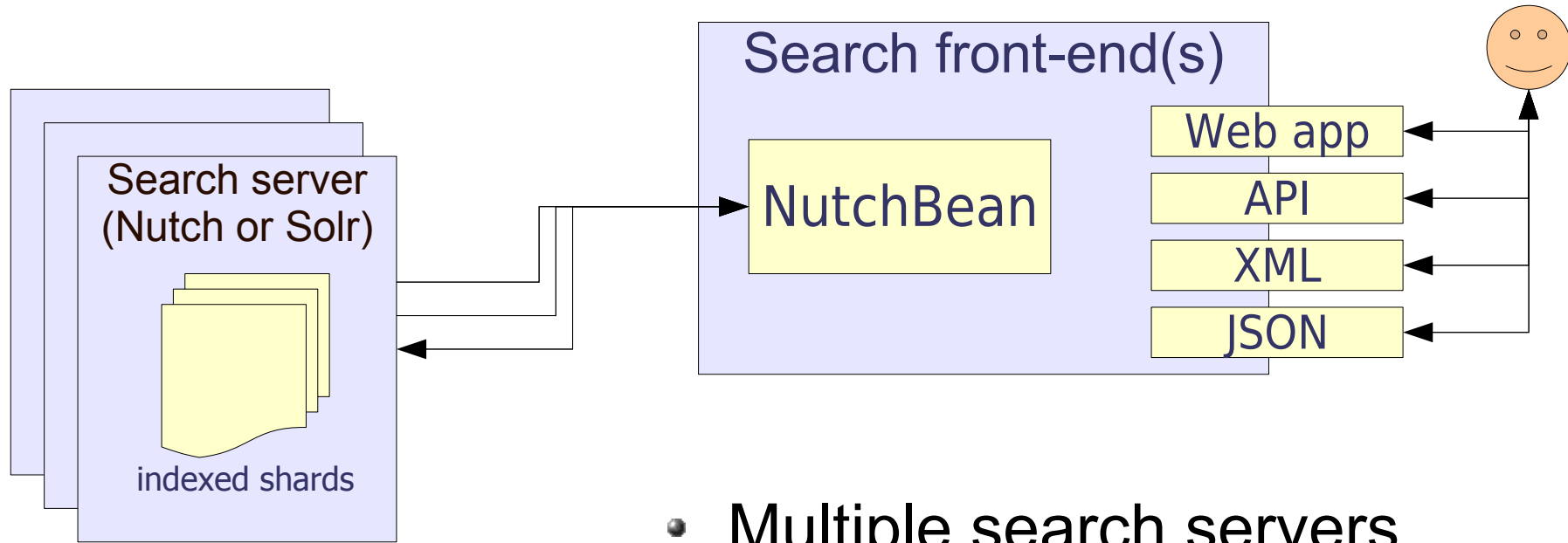


Cycles may overlap

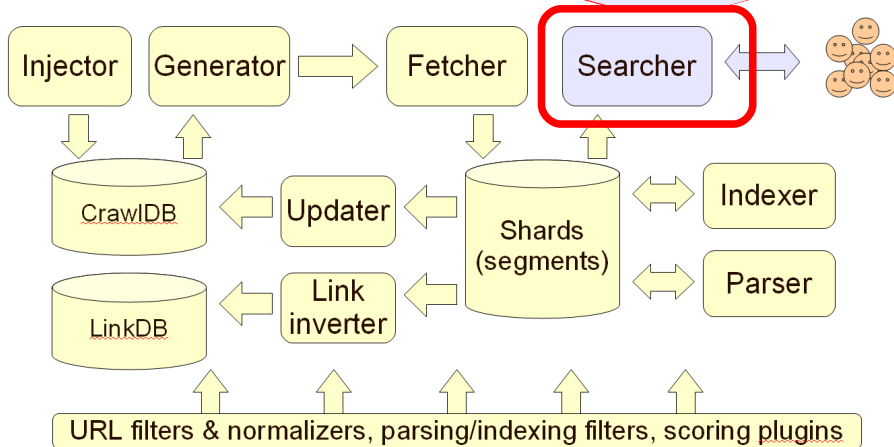




Nutch distributed search



- Multiple search servers
- Search front-end dispatches queries and collects search results
 - multiple access methods (API, OpenSearch XML, JSON)
- Page servers build snippets
- Fault-tolerant *



* with degraded quality



Search configuration

- Nutch syntax is limited – on purpose!
 - Some queries are costly, e.g. leading wildcard, very long
 - Some queries may need implicit (hidden) expansion
- Query plugins
 - From user query to Lucene/Solr query

User query: web search

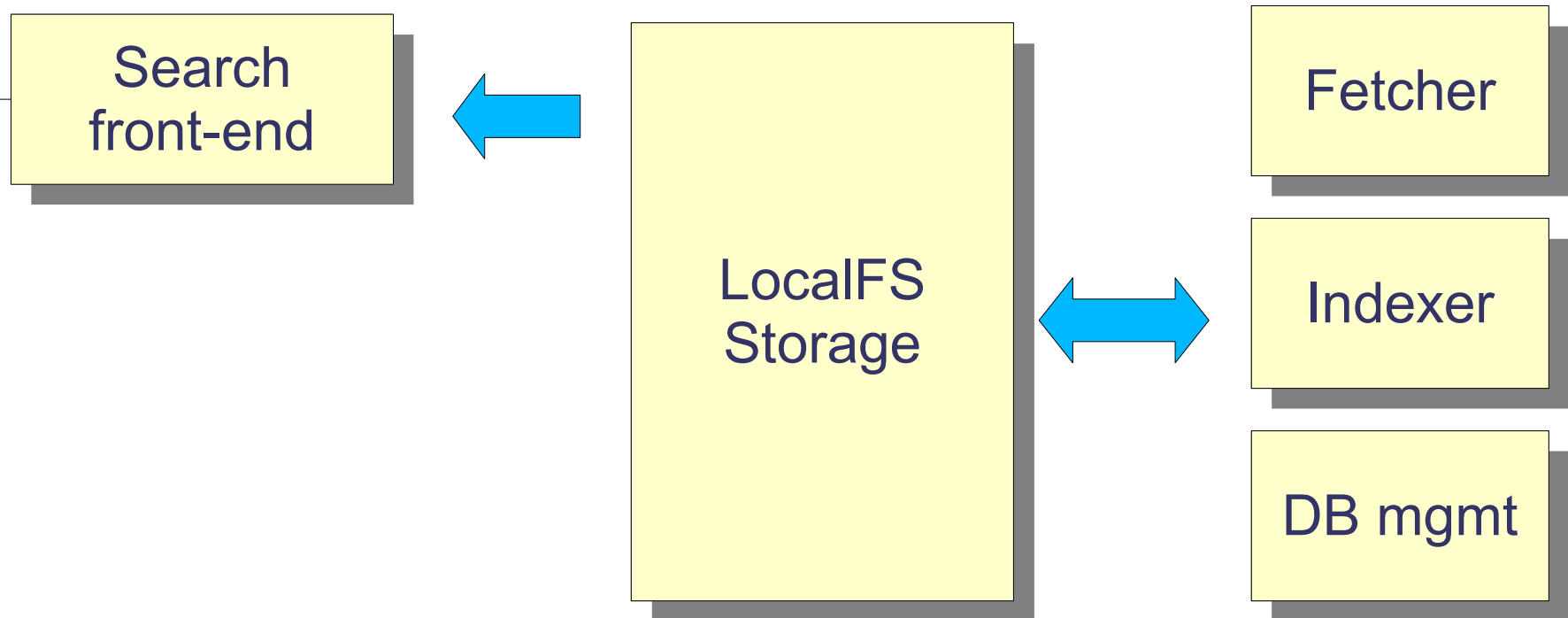
```
+(url:web^4.0 anchor:web^2.0 content:web title:web^1.5 host:web^2.0)  
+(url:search^4.0 anchor:search^2.0 content:search title:search^1.5  
host:search^2.0) url:"web search"~10^4.0 anchor:"web search"~4^2.0  
content:"web search"~10 title:"web search"~10^1.5  
host:"web search"~10^2.0
```

- Search server configuration
 - Single search vs. distributed
 - Using Nutch searcher, or Solr, or a mix
 - Currently no global IDF calculation



Deployment: single-server

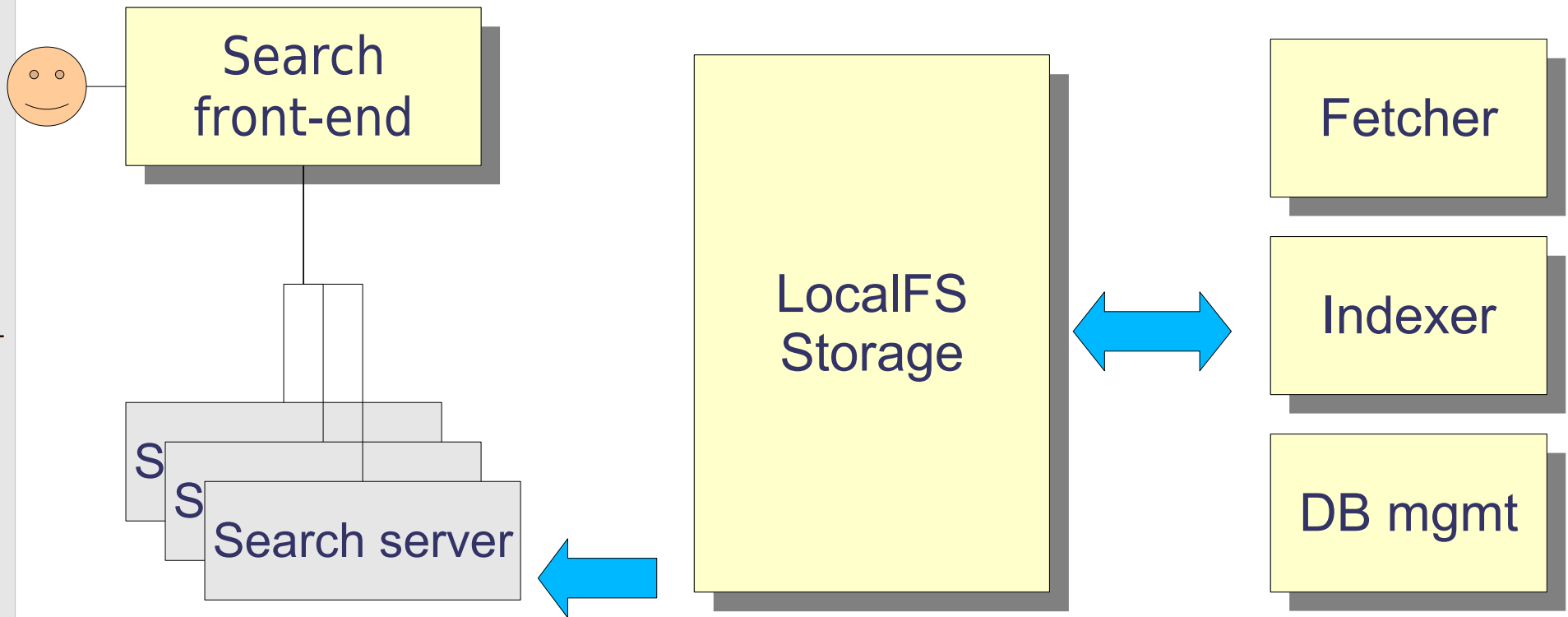
- The easiest but the most limited deployment
- Centralized storage, centralized processing
- Hadoop LocalFS and LocalJobTracker
- Drop nutch.war in Tomcat/webapps and point to shards





Deployment: distributed search

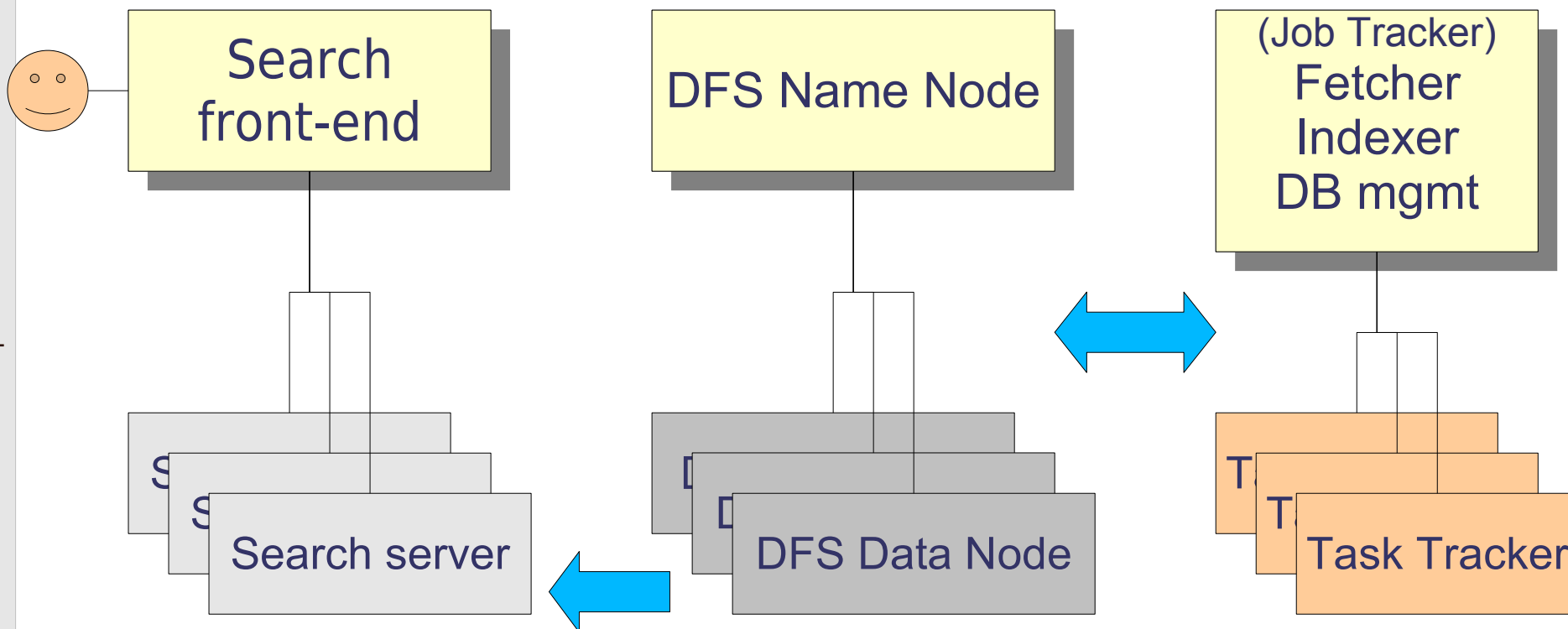
- Local storage on search servers preferred (perf.)





Deployment: map-reduce processing & distrib. search

- Fully distributed storage and processing
- Local storage on search servers preferred (perf.)





Nutch on a Hadoop cluster

- Assumes an up & running Hadoop cluster
 - ... which is covered elsewhere
- Build `nutch.job` jar and use it as usual:

```
bin/hadoop jar nutch.job <className> <args...>
```

- Note: Nutch configuration is inside `nutch.job`
 - When you change it you need to rebuild job jar
- Searching is not a map-reduce job – often on a separate group of machines

Hadoop Map/Reduce Administration

State: RUNNING

Started: Wed Apr 15 13:39:01 EDT 2009

Version: 0.19.1, r745977

Compiled: Fri Feb 20 00:16:34 UTC 2009 by ndaley

Identifier: 200904151339

Cluster Summary

Maps	Reduces	Total Submissions	Nodes	Map Task Capacity	Reduce Task Capacity	Avg. Tasks/Node
22	17	37	8	18	16	4.25

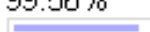
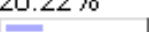
Scheduling Information

Queue Name	Scheduling Information
default	N/A

Filter (Jobid, Priority, User, Name)

Example: 'user:smith 3200' will filter by 'smith' only in the user field and '3200' in all fields

Running Jobs

Jobid	Priority	User	Name	Map % Complete	Map Total	Maps Completed	Reduce % Complete	Reduce Total	Reduces Completed	Job Scheduling Information
job_200904151339_0039	NORMAL	nutch	index-lucene /data/ /out/indexes	99.56% 	476	471	28.22% 	17	0	

Completed Jobs



Nutch index

Luke - Lucene Index Toolbox, v 0.9.9 (2009-09-30)

File Tools Settings Help

Overview Documents Search Files Plugins

Browse by document number:
Doc. #: 0 0

Browse by term:
(Hint: enter a substring and press Next to start at the nearest term).
First Term Term: boost
Decoded value:
Browse documents with this term (0 documents)
Document: ? of ?
Term freq in this doc: ?

Delete specified list of documents:

Example: 0,12,45-90,17,123,30-32

Doc #: 0 **Flags:** I - Indexed; T - Tokenized; S - Stored; V - Term Vector (o - offsets; p - positions)
O - Omit Norms; f - Omit TF; L - Lazy; B - Binary; C - Compressed

Field	ITSVopfOLBC	Norm	Value
boost	--S----O---	---	1.4142135
cache	--S----O---	---	content
digest	--S----O---	---	752e8663a0859a8a356741cefcb65da2
kw	ITS-----	0.625	currency stock business finance
segment	--S----O---	---	20090924115804
title	ITS-----	0.5	Business, financial, personal finance news - CNNMoney.com
tstamp	--S----O---	---	20090924100058609
url	ITS-----	0.21875	http://money.cnn.com/

Selected field: Copy text to Clipboard:

Index name: e:\work\nu...bin\test\indexes\part-00000



... and press

Search

[About](#) [FAQ](#)clustering [help](#)

Hits **1-10** (out of about 9,530 total matching pages):

[Live web site traffic analysis and hit counters](#)

... counters **web** traffic analysis by **Web-Stat Live web** site traffic analysis and ... connected to www. ...
<http://www.web-stat.com/> ([cached](#)) ([explain](#)) ([anchors](#))

[Yahoo!](#)

Yahoo! My Yahoo! My Mail Why miss out? To see all the new Yahoo! home ...
<http://www.yahoo.com/> ([cached](#)) ([explain](#)) ([anchors](#))

[Free Website Hit Counters, Web Page and Traffic Counter](#)

... Free Website Hit Counters, **Web** Page and Traffic ... and reliable website hit counters, **web** page counters and traffic ...
<http://www.website-hit-counters.com/> ([cached](#)) ([explain](#)) ([anchors](#))

[Opera browser: Home page](#)

... Mini, a tiny **web** browser for your ... that lets you surf any **Web** site. Download Opera Mobile Learn ...
<http://www.opera.com/> ([cached](#)) ([explain](#)) ([anchors](#))

[NUMBER THEORY WEB](#)

... NUMBER THEORY **WEB** Number Theory **Web** Aims New Listings Number theorists ... Search the Number Theory **Web** Pages The main ...
<http://www.numbertheory.org/ntw/> ([cached](#)) ([explain](#)) ([anchors](#))

[Website Design Chicago | Website Design Company | Illinois Web Design | Weblinx, Inc.](#)

... Design | Logo Design | Flash Animation | **Web** Hosting | Search Engine Optimization © 2008 ... **Web** Design - Website Design Chicago - Web ... Site Design - Website Design Chicago ...
<http://www.weblinxinc.com/> ([cached](#)) ([explain](#)) ([anchors](#))

[Walla Walla Web Weavers](#)

... Walla Walla **Web** Weavers Walla Walla **Web** Weavers is a full ... estimates! Copyright ©2004 Walla Walla
<http://www.wwwwebweavers.com/> ([cached](#)) ([explain](#)) ([anchors](#))

[::Oberlausitz-Web:: Die grösste Linkliste der Oberlausitz und Niederschlesien!](#)

Web Design

- [Website Design Chicago | Website De...](#)
- [Sacramento Web Design](#)
- [Online Web Design Course - A free 0...](#)

Web Hosting

- [Web Host Directory: The best free s...](#)
- [web hosting, web design and the sma...](#)
- [web-cp: a free web hosting control ...](#)

Promotion Software

- [Web Promotion Software "WebCEO" - S...](#)
- [Dating Software, Social Network Sof...](#)
- [Conception et Création de Sites Web...](#)

Internet Services

- [welcome@web-e-facts - internet serv...](#)
- [Solent Computer and Internet Servic...](#)
- [Web-Galleries \(Internet Specialists...](#)

Search Engine Marketing

- [Search engine marketing, web market...](#)
- [Web-Grafix Top Search Engine](#)



Conclusions

(This overview is a tip of the iceberg)

Nutch

- Implements all core search engine components
- Scales well
- Extremely configurable and modular
- It's a complete solution – and a toolkit



Future of Nutch

- Avoid code duplication
- Parsing → Tika
 - Almost total overlap
 - Still some missing functionality in Tika → contribute
- Plugins → OSGI
 - Home-grown plugin system has some deficiencies
 - Initial port available
- Indexing & Search → Solr, Zookeeper
 - Distributed and replicated search is difficult
 - Initial integration needs significant improvement
 - Shard management - Katta?
- Web-graph & page repository → HBase
 - Combine CrawlDB, LinkDB and shard storage
 - Avoid tedious shard management
 - Initial port available



Future of Nutch (2)

- What's left then?
 - Crawling frontier management, discovery
 - Re-crawl algorithms
 - Spider trap handling
 - Fetcher
 - Ranking: enterprise-specific, user-feedback
 - Duplicate detection, URL aliasing (mirror detection)
 - Template detection and cleanup, pagelet-level crawling
 - Spam & junk control
- Share code with other crawler projects → crawler-commons
- Vision: á la carte toolkit, scalable from 1-1000s nodes



Summary

- Q&A
- Further information:
 - <http://lucene.apache.org/nutch>