Architecture of Information Retrieval Systems

Notation

Documents file

Docs

File of catalog records

Catalog

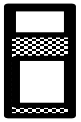
Searchable index

Index

User interface

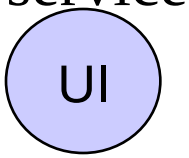
Human action

Automatic process



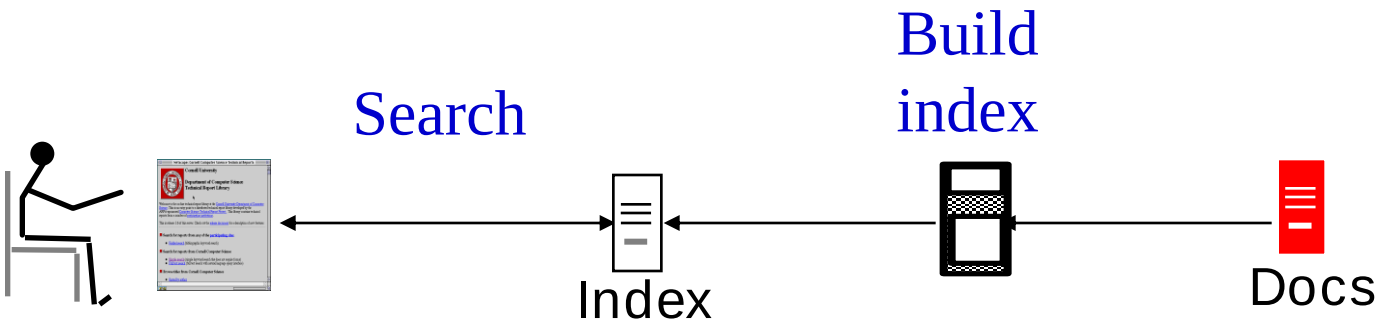
Physical objects

User interface service



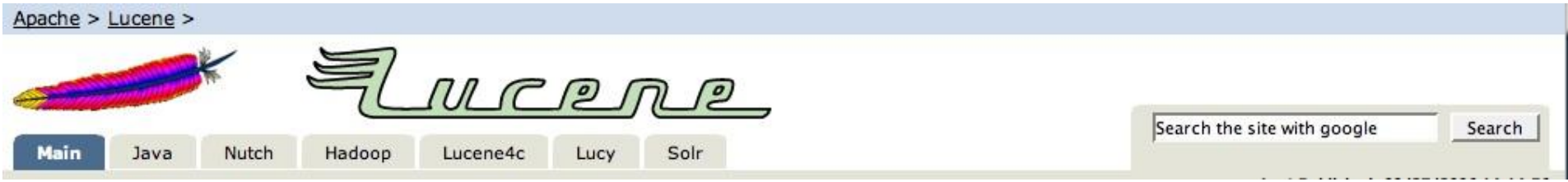
Single Homogeneous Collection: Full Text Indexing

- Documents and indexes are held on a single computer system (may be several computers).
- Information retrieval uses a full text index, which may be tuned to the specific corpus.



Examples: SMART, Lucene

Lucene



Apache Lucene

- High-performance, full-featured text search engine library.
- Written entirely in Java.
- Suitable for nearly any application that requires full-text search, especially cross-platform.
- An open source project available for free download.

Nutch

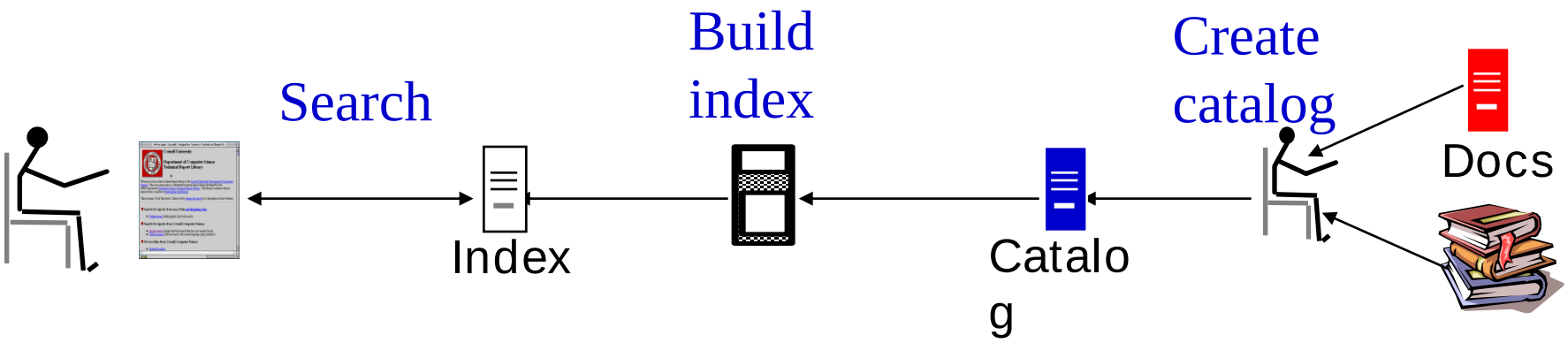
Nutch is open source web-search software for the Web, built on Lucene. It adds:

- A medium scale crawler
- A link-graph database
- Parsers for HTML and other document formats

Doug Cutting is the principal author of both Lucene and Nutch.

Single Homogeneous Collection: Use of Catalog Records

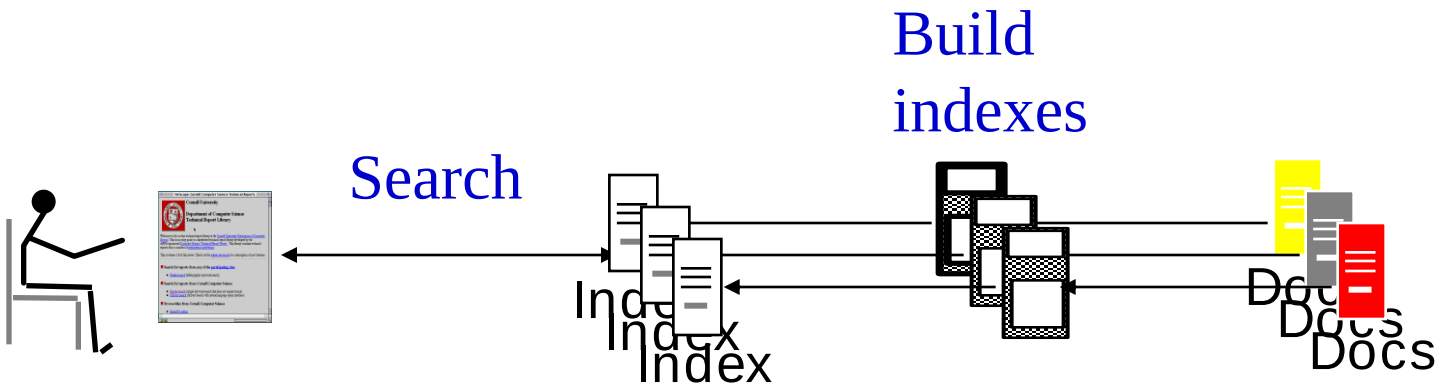
- Documents may be digital or physical objects, e.g., books.
- Documents are described by catalog records generated manually (or sometimes automatically).
- Information retrieval uses an index of catalog records



Example: Library catalog

Several Similar Collections: One Computer System

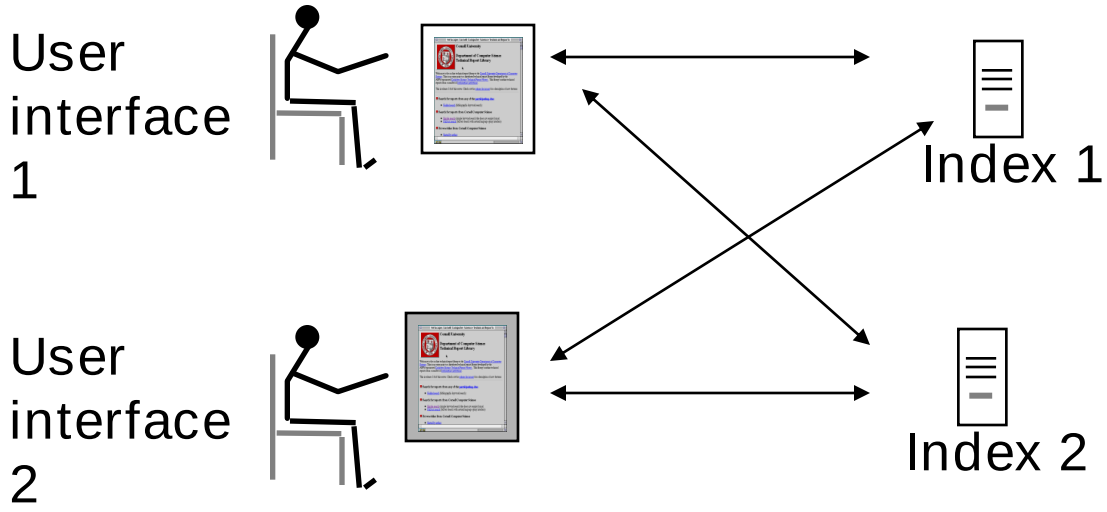
- Several more or less similar collections are held on a single computer system.
- Each collection is indexed separately using the same software, procedures, algorithms, etc. (but tuned for each collection, e.g., different stoplists).



Example: PubMed

Distributed Architecture: Standard Search Protocols

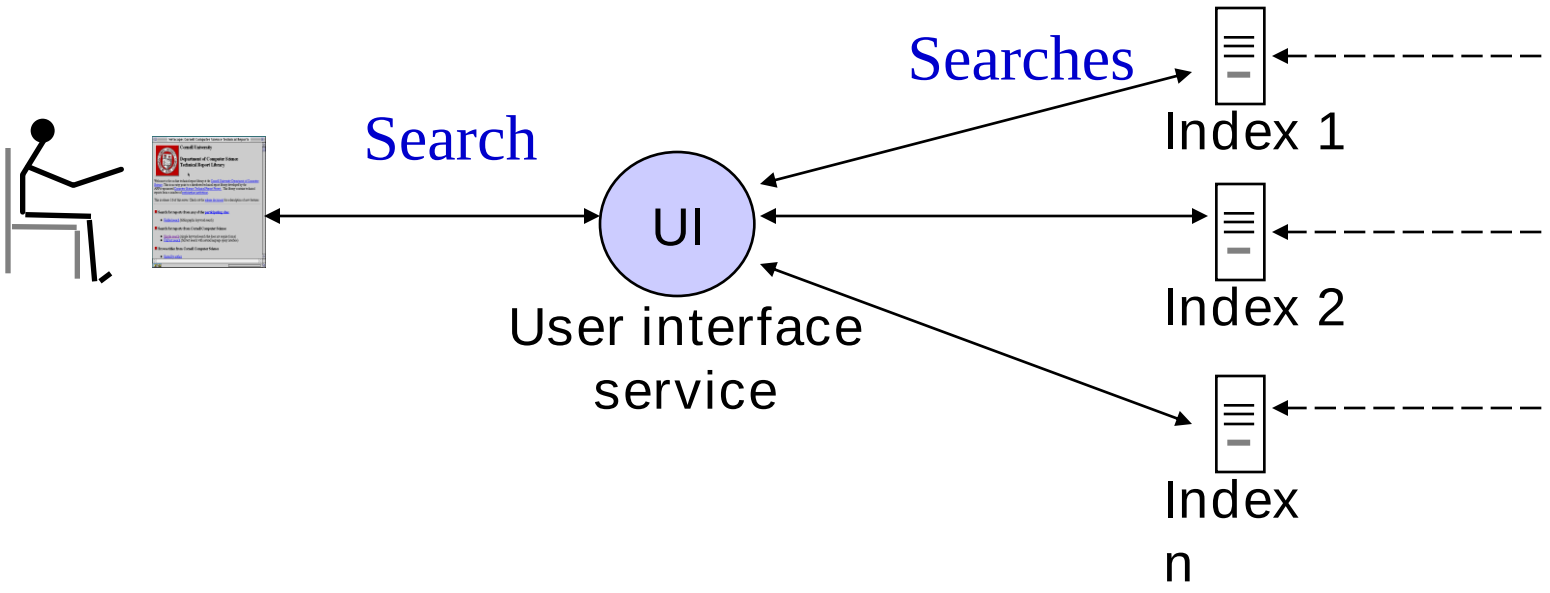
- A user interface is configured so that the user can search several different indexes, one at a time



Strict adherence to standards allows any user interface to search any conforming search service.

Distributed Architecture: Meta-search (Broadcast Search)

- A user interface service broadcasts a query to several indexes and merges the results.
- Can be used with full text or catalogs.



Example: Dienst

Distributed Architecture: Broadcast Search

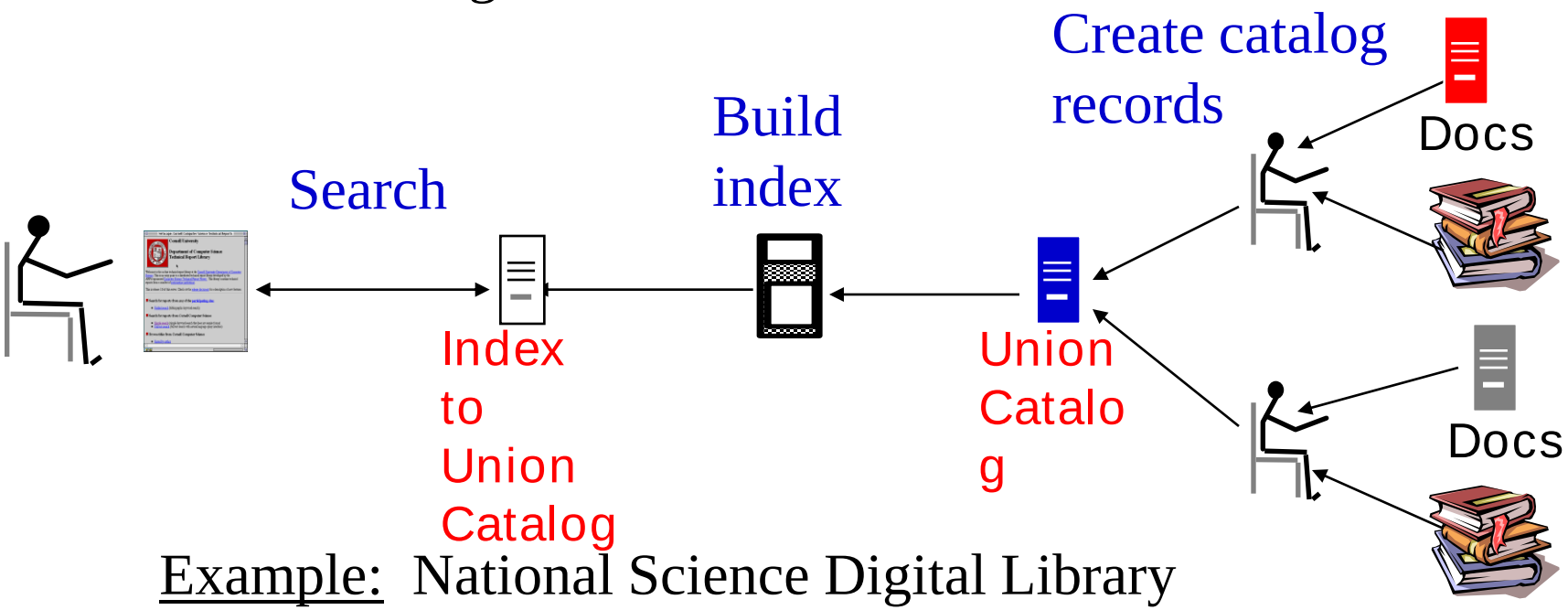
Problems with Broadcast Search

- Performance: If any collection does not respond, the Interface Server waits for a time out.
- Recall: If any collection does not respond, documents in that collection are not found.
- Ranking and duplicates: There are great difficulties in reconciling ranked lists from different collections.

Conclusion: Broadcast search does not scale beyond a small number of collections, even with strict standardization.

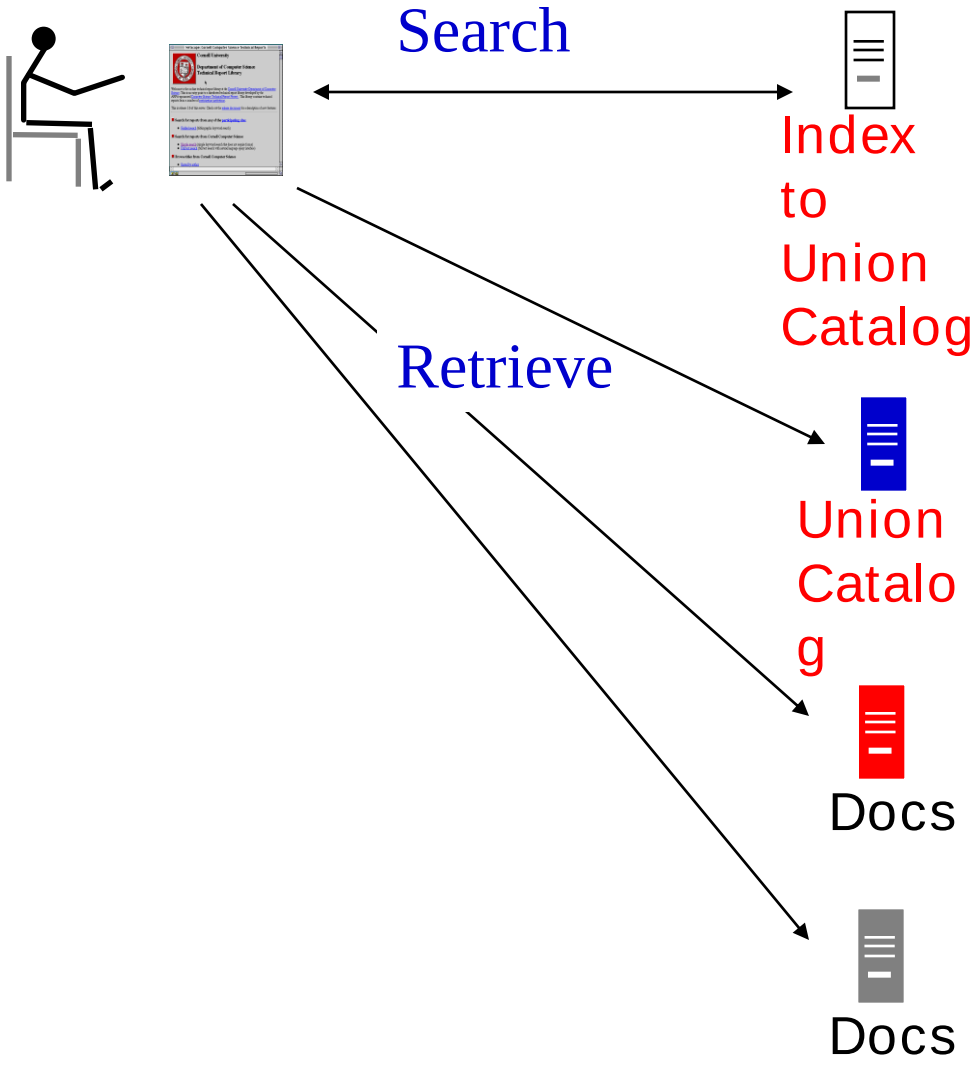
Union Catalog

- Catalog records from several libraries are merged into a single ***union catalog***
- Information retrieval uses an index of the records in the union catalog



Example: National Science Digital Library

Use of Union Catalogs

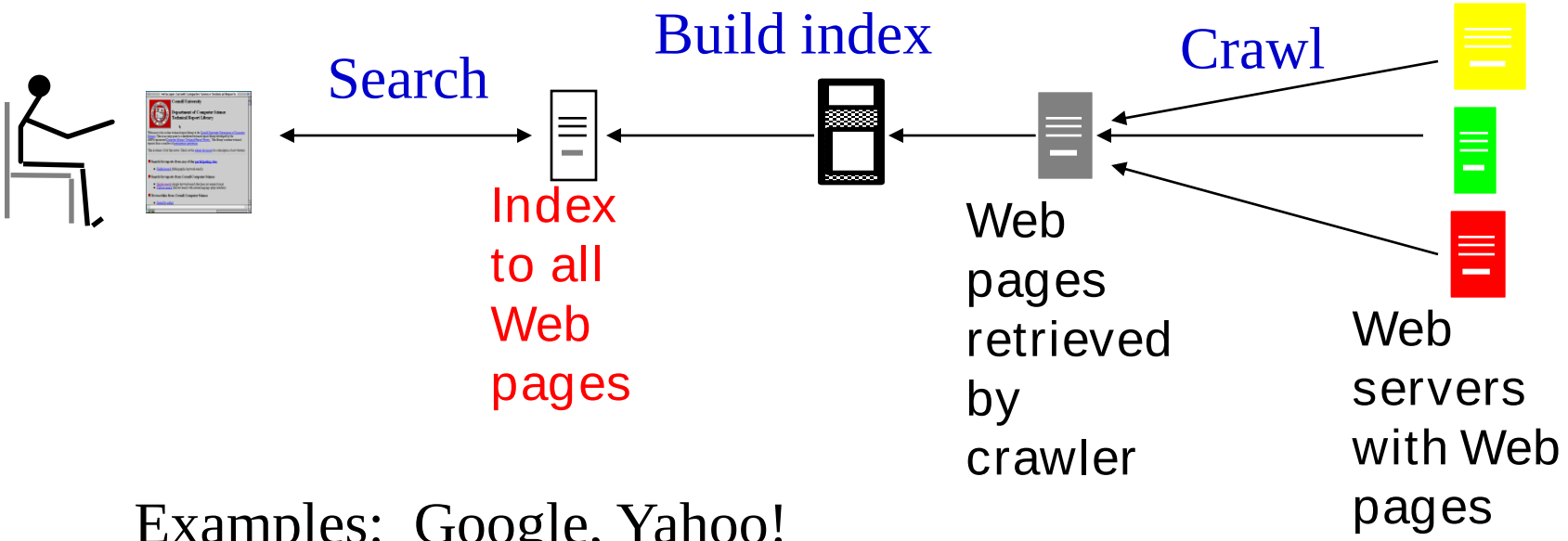


Batch indexing: Metadata about all items is accumulated in a central system.

Real-time searching: The user (a) searches the central index, (b) retrieves catalog records, (c) retrieves documents from collections.

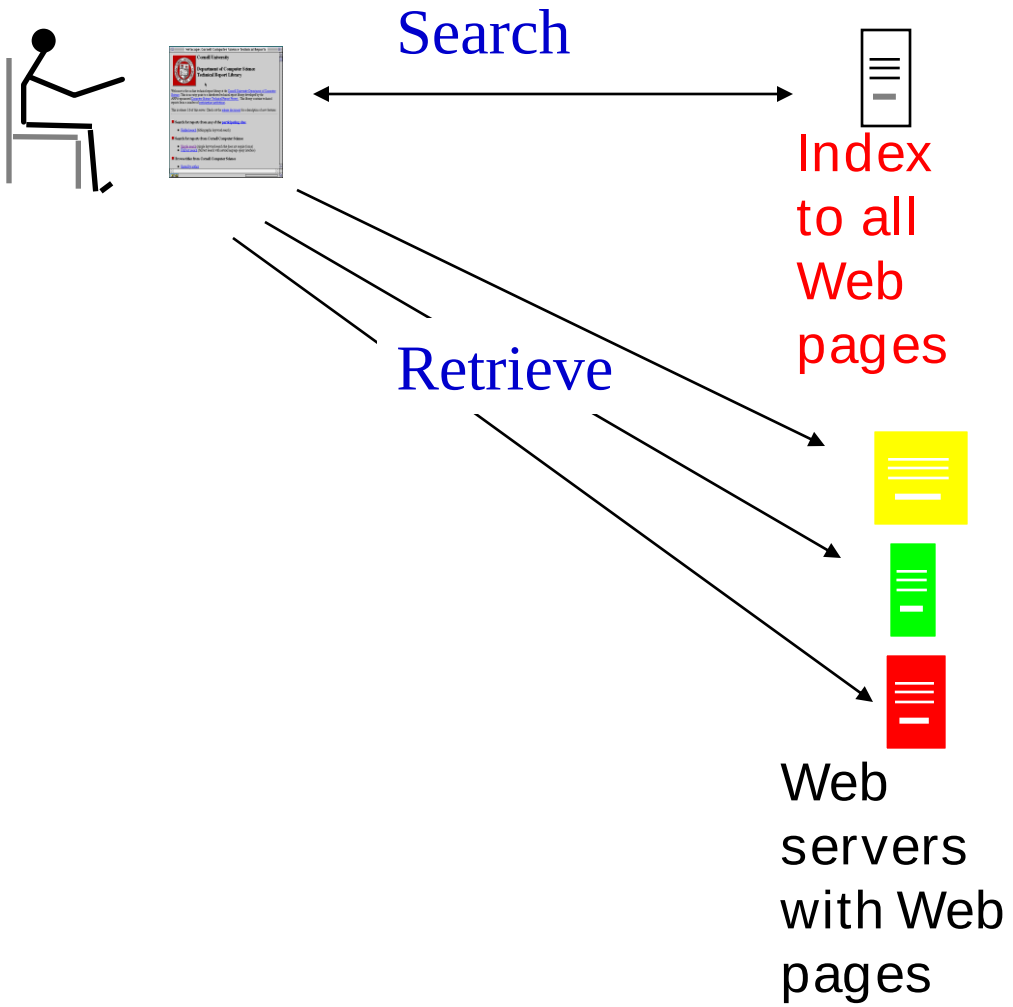
Web Searching: Architecture

- Documents stored on many Web servers are indexed in a single central index. (This is similar to a union catalog.)
- The central index is implemented as a single system on a very large number of computers



Examples: Google, Yahoo!

Use of Web Search Service



Batch indexing: Each Web page is brought to the central location and indexed.

Real-time searching: The user (a) searches the central index, (b) retrieves documents (Web pages) from original location.

Web Searching: Building the Index

Documents are Web pages

Each document is:

- **identified** by Web Crawling
- **copied** to a central location
- **indexed** and added to the central index

After indexing the documents may be discarded, but a copy may be retained, for use by the user interface.

Web Crawling

Advantages of Web crawling

- Entirely automatic, low cost. Highly efficient at gathering very large amounts of material.

but ...

- Can only gather openly accessible materials.
- Cannot gather material in databases unless explicit URLs are known.
- Cannot easily make use of metadata provided by collections.