The GRACE Project

GRid enabled seArch and Categorization Engine

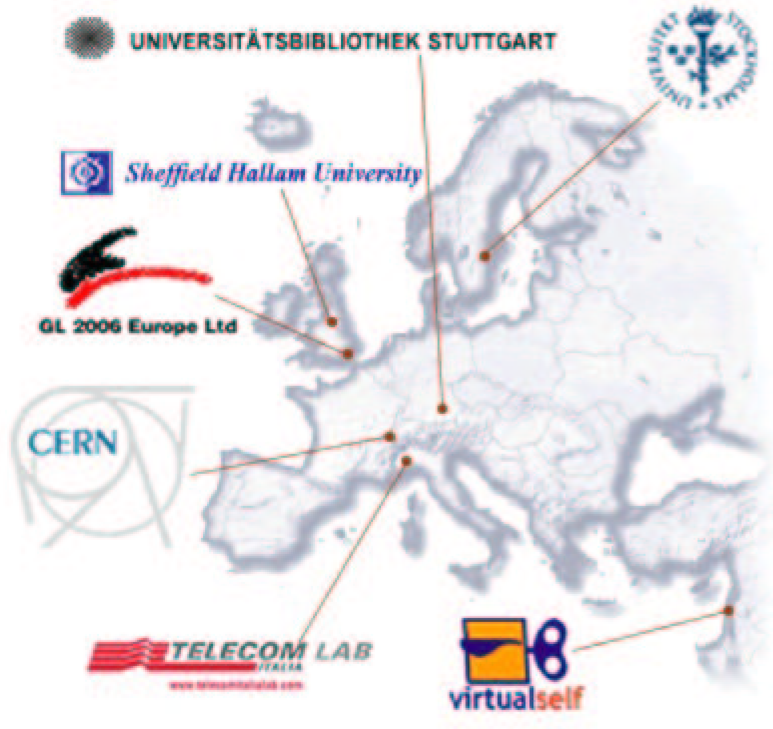
http://www.grace-ist.org/

The GRACE Project

GRid enabled seArch and Categorization Engine

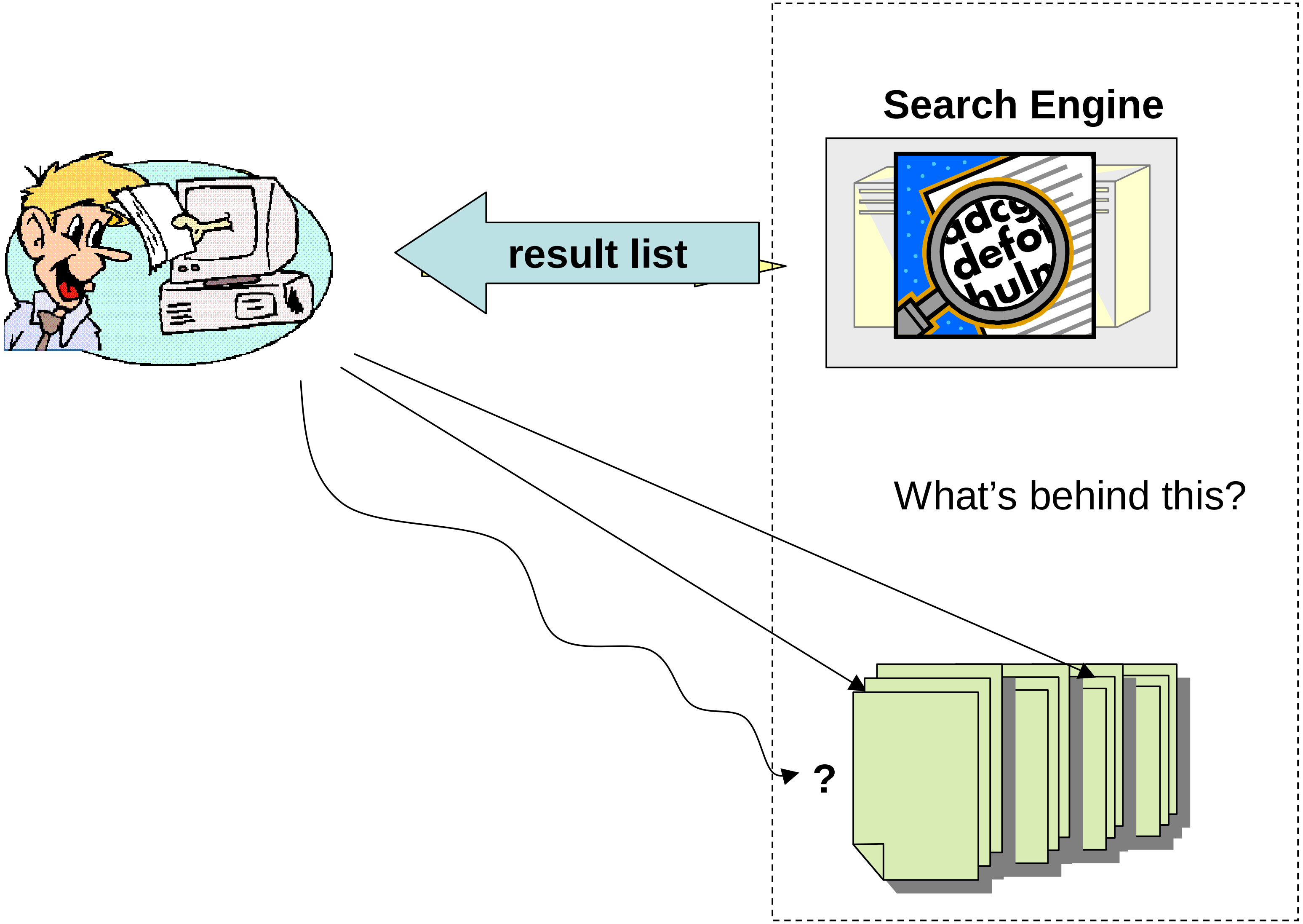
GRACE proposes the development of a distributed **search and categorization engine** that will enable just-in-time, flexible allocation of data and computational resources.

GRACE handles structured and unstructured **textual information** (text files, documents, Web pages, text stored in databases) in GRID environment.



Project's lifetime: September 2002-February 2005

Why a Search Engine ?



Search Engine

result list

What's behind this?

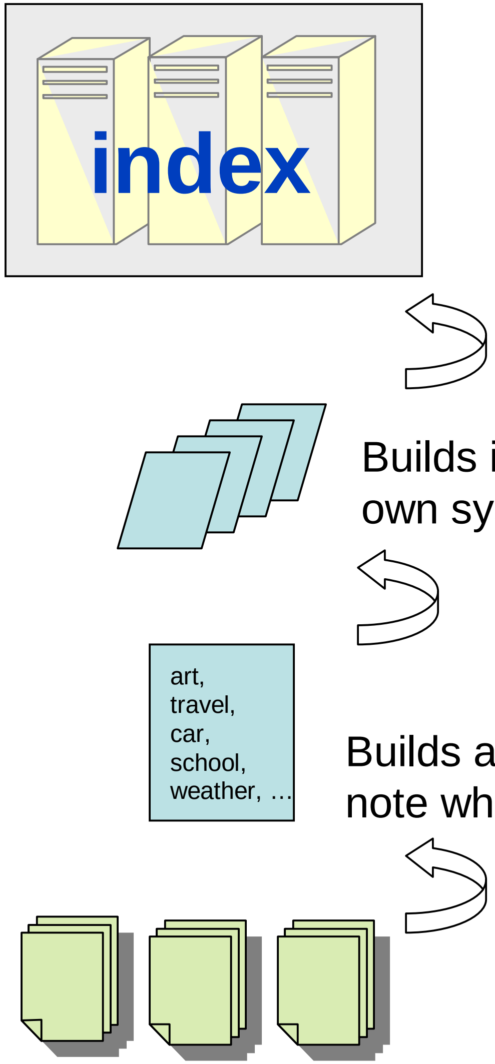
?

What is behind the Search Engine interface?

There are differences in the ways various search engines work, but they all perform three basic tasks:

- They periodically **search for information**.
- They **keep an index** of the words they find, and where they find them.
- They **allow users to look for words** or combinations of words found in that index.

Search Engine

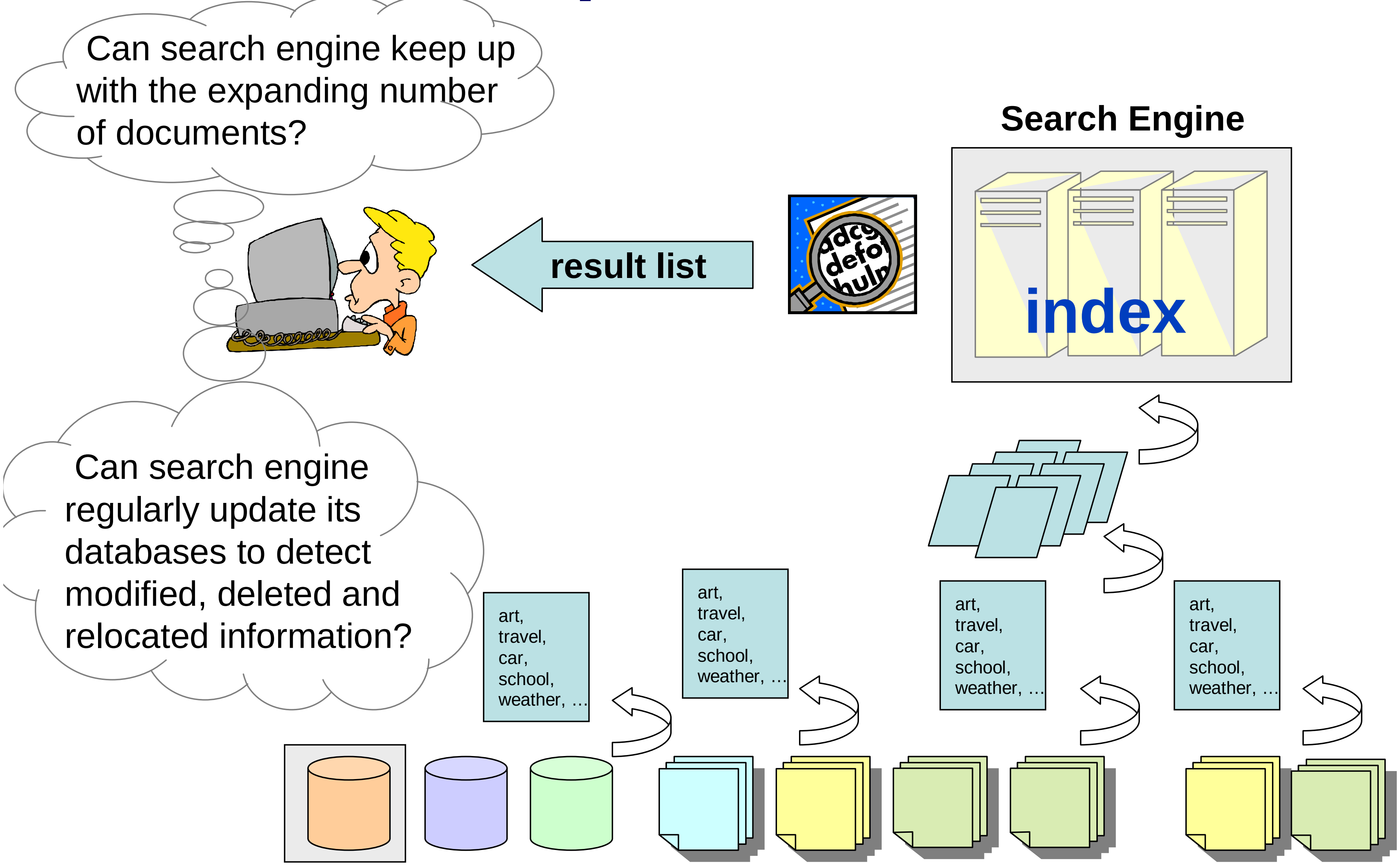


Encodes the data to save space and store them for user to access

Builds index based on its own system of weighting

Builds a list of words and note where they were found

Which are the problems ?



Can search engine keep up with the expanding number of documents?

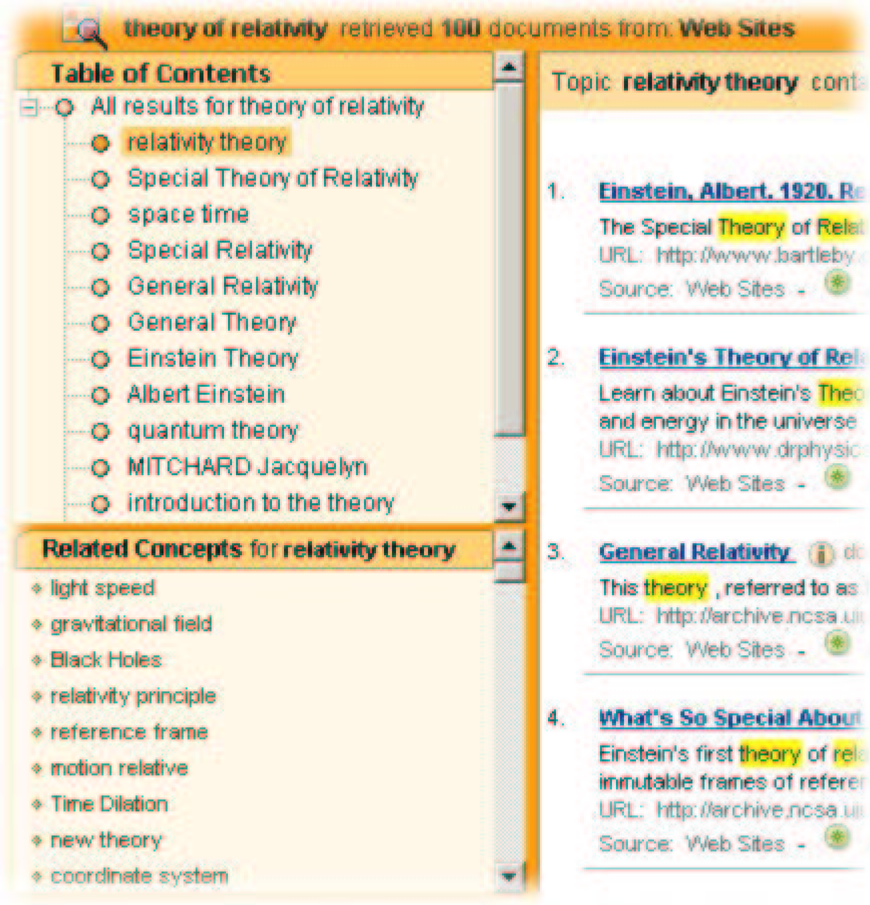
Can search engine regularly update its databases to detect modified, deleted and relocated information?

Project Objectives

GRACE specifically addresses the situations in which a centralized index is simply unfeasible, and a distributed search-and-retrieval is necessitated.

Centralised solution limitations	GRACE solution
Scalability: the amount of documents may overwhelm any centric search engine (limited by network bandwidth, disk storage, computational power)	**Decentralized Index and Processing:** documents are indexed locally in each Grid-node. The resulting index will be also stored locally and will allow querying on-demand from other nodes in the Grid
Frequency of Update: the rate of update cannot be too frequent, this may prove too slow for some content sources (dynamic data, “real time” information)	**Smaller indexes:** faster update
Access and integration: not all content sources are accessible to an external crawler (local search, authentication, heterogeneous databases…)	**Personalized Interfaces** according to user/organization profile, access rights, subscription, etc.
Accuracy: central indexing only approaches the least common denominator in documents, it cannot support accurate search	**Domain Specific Metadata Search:** greater accuracy

The Categorization Engine



theory of relativity retrieved **100** documents from: **Web Sites**

Table of Contents

- All results for theory of relativity
  - relativity theory
  - Special Theory of Relativity
  - space time
  - Special Relativity
  - General Relativity
  - General Theory
  - Einstein Theory
  - Albert Einstein
  - quantum theory
  - MITCHARD Jacquelyn
  - introduction to the theory

Related Concepts for **relativity theory**

- light speed
- gravitational field
- Black Holes
- relativity principle
- reference frame
- motion relative
- Time Dilation
- new theory
- coordinate system

Topic **relativity theory** cont

1. Einstein, Albert. 1920. Re
The Special Theory of Relat
URL: http://www.bartleby.
Source: Web Sites -

2. Einstein's Theory of Rel
Learn about Einstein's Theo
and energy in the universe
URL: http://www.drphysic
Source: Web Sites -

3. General Relativity
This theory , referred to as
URL: http://archive.ncsa.ui
Source: Web Sites -

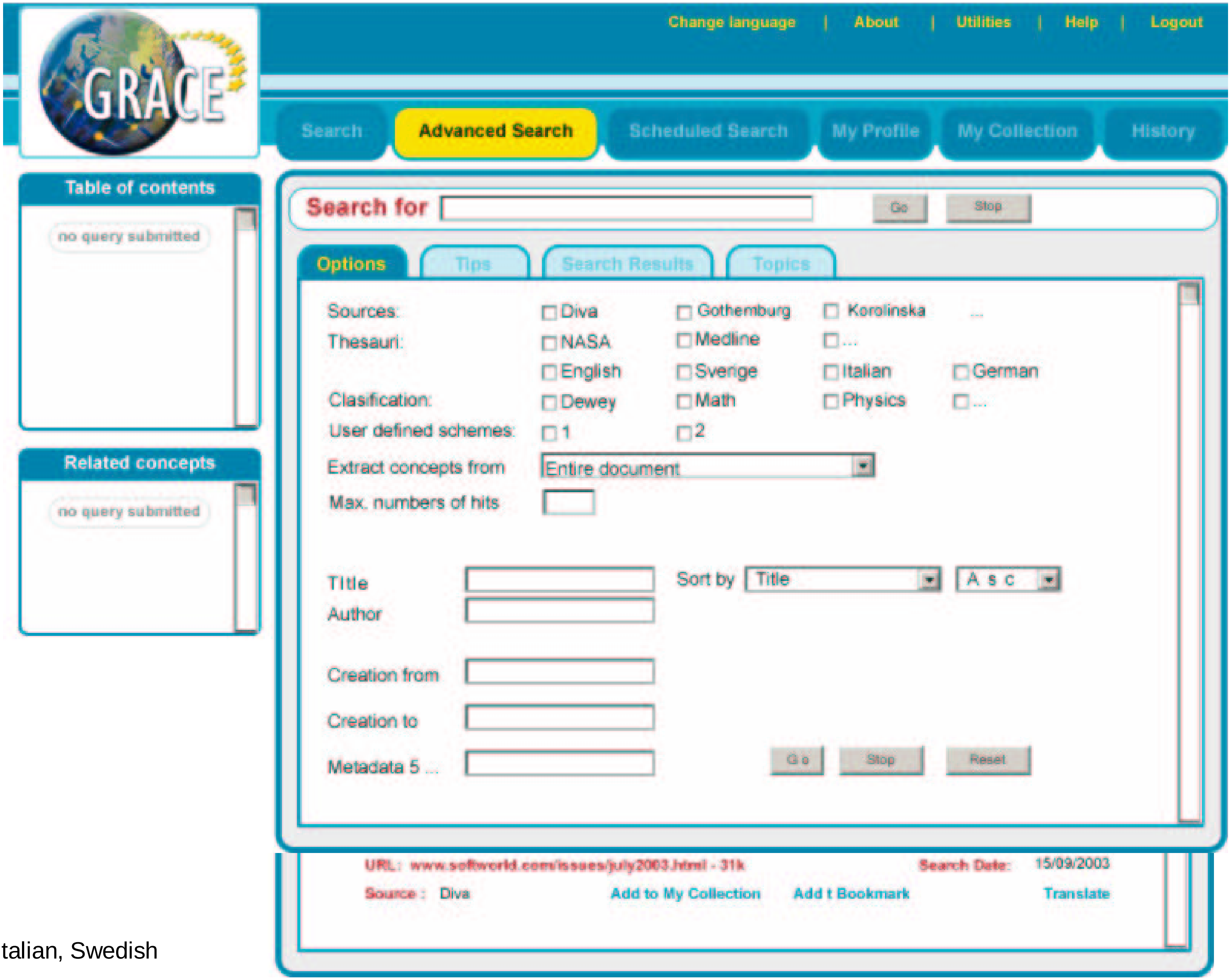
4. What's So Special About
Einstein's first theory of rel
immutable frames of referer
URL: http://archive.ncsa.ui
Source: Web Sites -

Virtual Self: http://www.vself.com/

- By analyzing all of the text in real-time, the Categorization Engine autonomously **infers the relevant key phrases** (idioms) in a document.
- Classes are built on-the-fly.
- The Categorization Engine **creates a "Real-Time" hierarchical Concept map** associate pages to each class, and orders them.

e.g.: Categories, automatically generated from an external search engine results for ”theory of relativity”.

The User Interface



Sources: □ Diva □ Gothemburg □ Korolinska ...

Thesauri: □ NASA □ Medline □ ...

□ English □ Sverige □ Italian □ German

Clasification: □ Dewey □ Math □ Physics □ ...

User defined schemes: □ 1 □ 2

Extract concepts from: Entire document

Max. numbers of hits

Title

Author

Sort by: Title; A s c

Creation from

Creation to

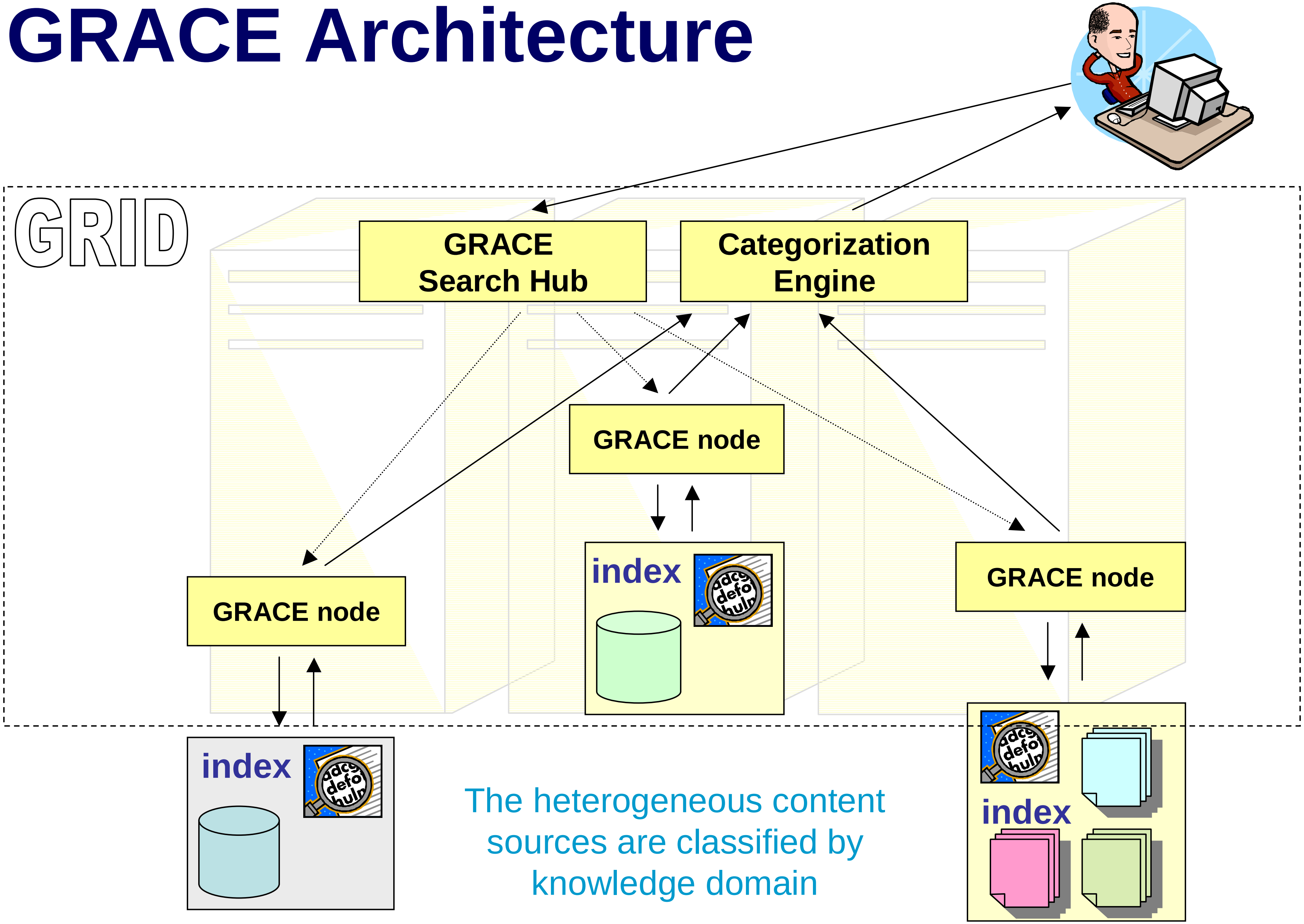
Metadata 5 ...

URL: www.softworld.com/issues/july2003.html - 31k Search Date: 15/09/2003

Source : Diva

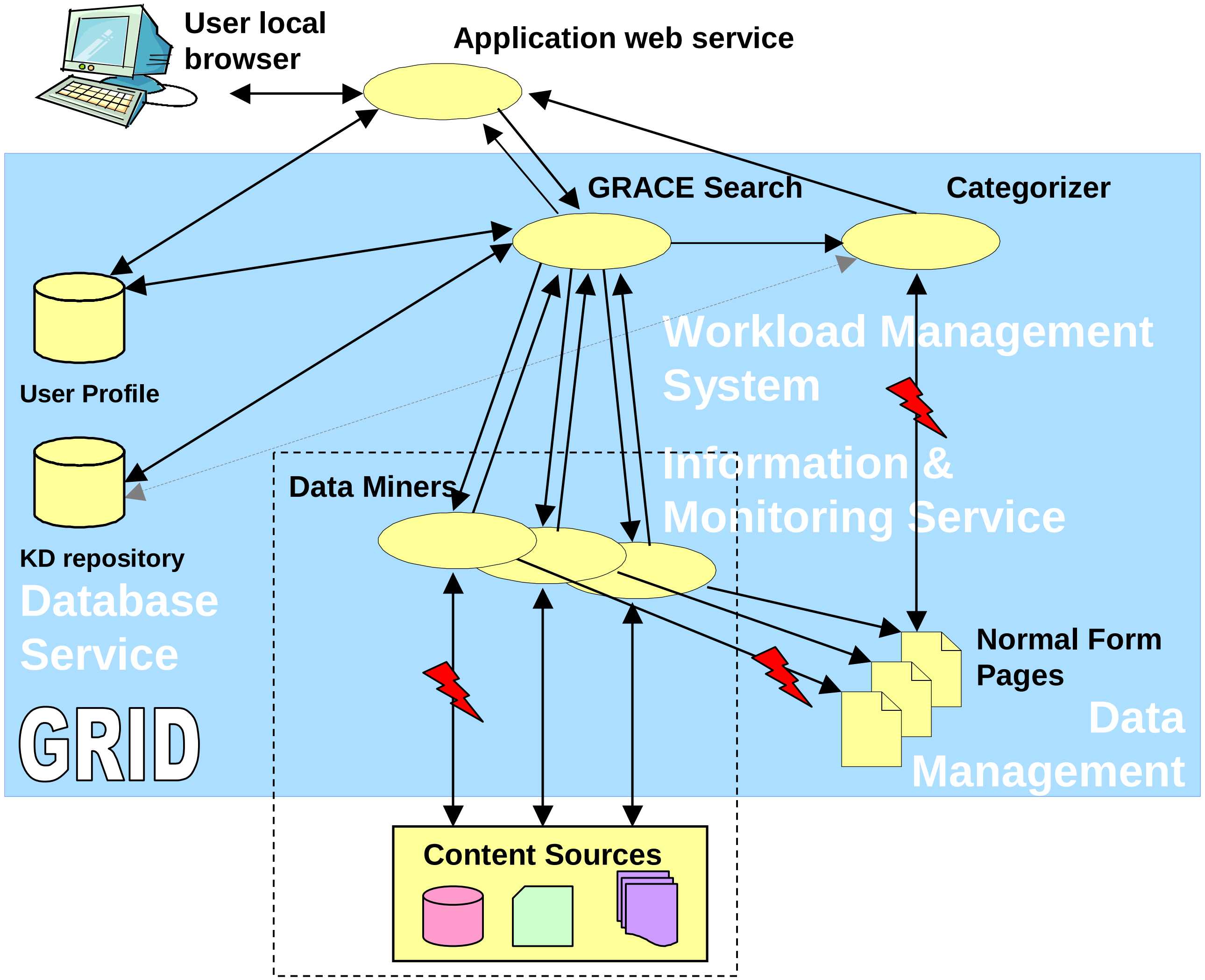
- API and Web interface
- Selection on Content Sources
- Thesauri-based categorization
- User defined classification schema
- Search limit
- Accurate research fields
- Personalization
- Search history
- Scheduled queries and alert
- “My Collection” of documents
- Interface support English, German, Italian, Swedish

GRACE Architecture



The heterogeneous content sources are classified by knowledge domain

GRACE Architecture



Login: get user settings/preferences

Query: start a search

Look for relevant Content Source

Start the Categorization Process

Get KD information: thesauri,…

Start the Data Miners

Query the Content Sources

Send the result list

NFPs preparation

Save search history information

Send back the result list

Run categorization and integration

Send concept map of the results

Send categorized list of results

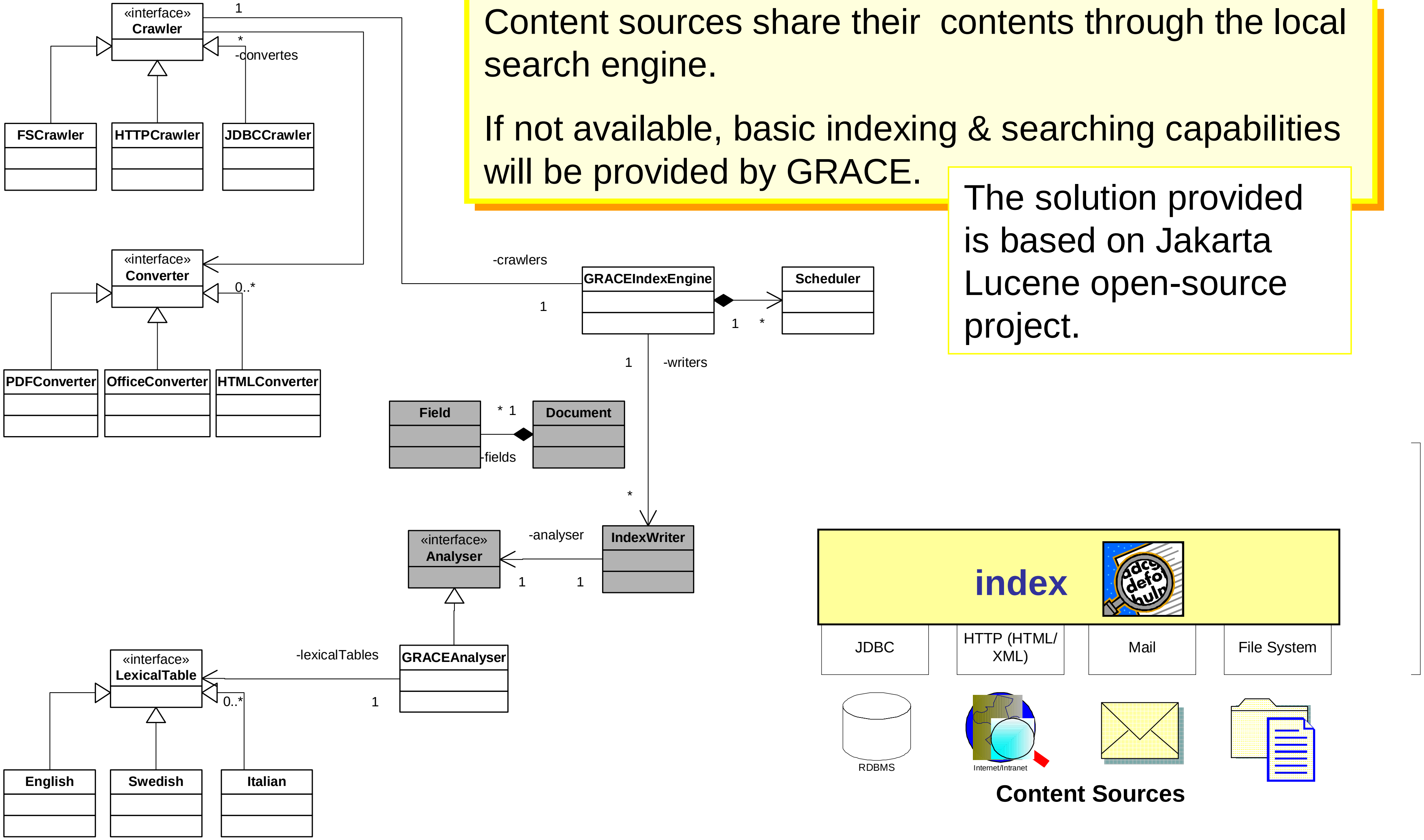
⚡ File transfer

Content Sources integration

Content sources share their contents through the local search engine.

If not available, basic indexing & searching capabilities will be provided by GRACE.

The solution provided is based on Jakarta Lucene open-source project.

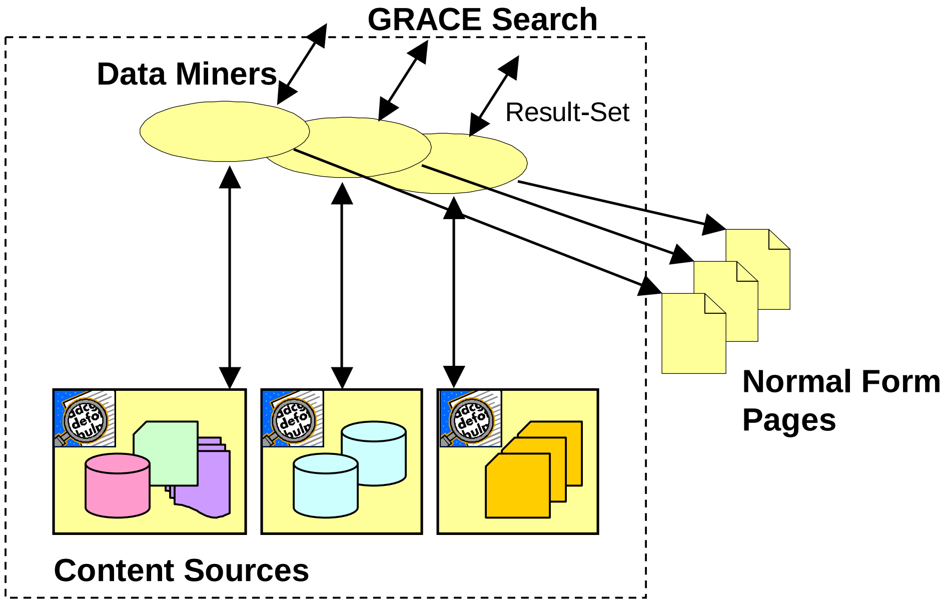


The Data Miner

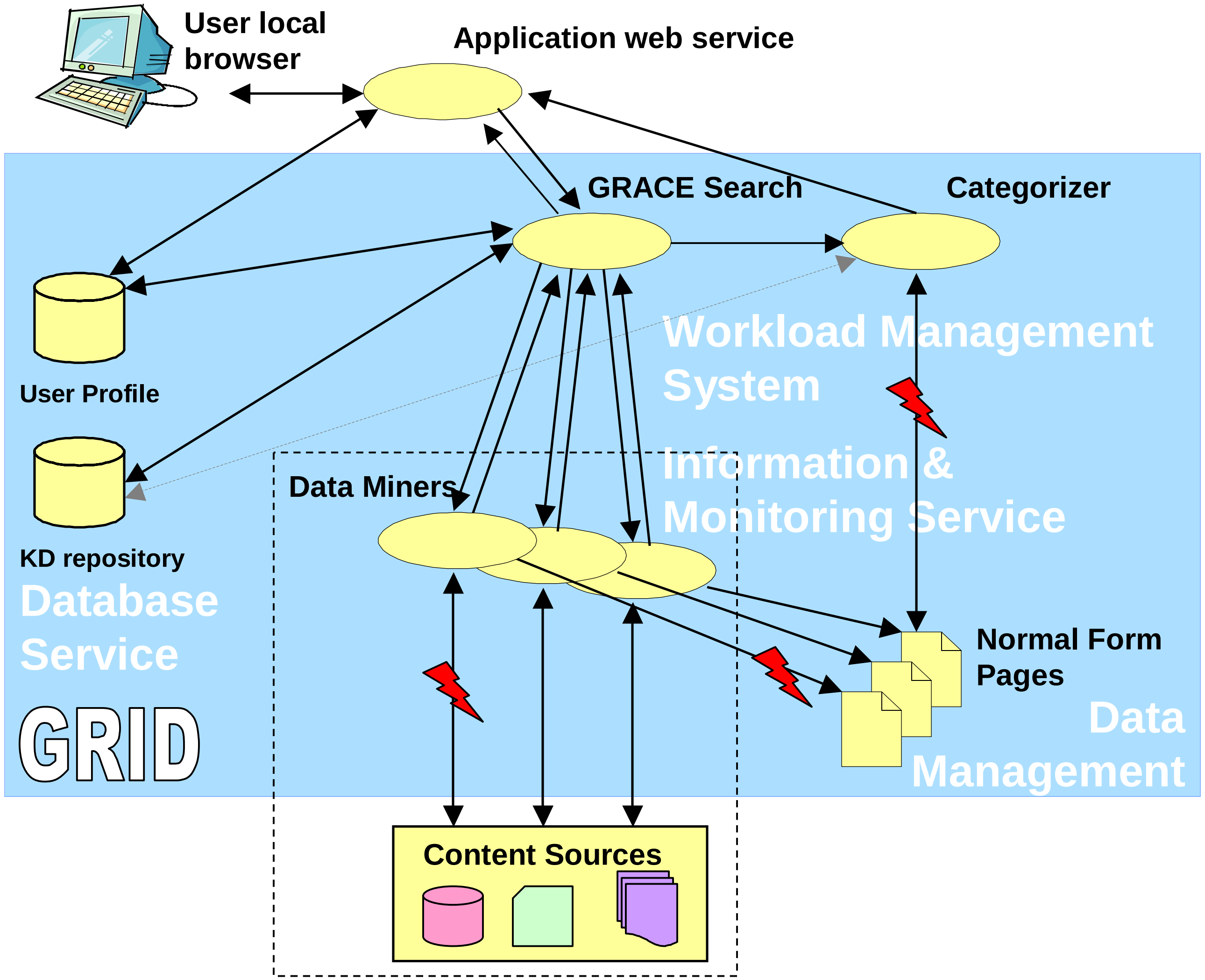
This component query a single content source, returning the source's **flat result** list as well as the documents in **digested form** for the categorization engine. It's main components are:

SEAL: The Search Engine Abstraction Layer sits on-top of any query-supporting module and translates the user query in API calls. Its output is a list of results. (Result-Set). Each result must contain a reference to the original document and a reference to the matching NFP file.

Document Processor: it processes the input documents (from the result list) and their meta-data. The output for each document is called NFP (Normal Form Page), which includes the digested document data, ready for the categorization algorithm.



Conclusions



Project activities:

- User Requirements
- General Architecture
- Local Search Engine
- Multilingual abilities
- User Interface
- Development testbed installation (Turin&Milan)
- Integration on Grid middleware
- Testing of the components
- Finalization of the GRACE toolkit and testing
- Installation and testing on an extended testbed
- Final Validation and Evaluation

Project's lifetime: September 2002-February 2005